E-GOLDvoice

GSM Voice Only Single Chip Solution

E-GOLDvoice (PMB7880), V1.xx

Target Specification

Rev. 1.05

Communication Solutions

Never stop thinking

E-GOLDvoice GSM Voice Only Single Chip Solution
CONFIDENTIAL
Revision History: 2006-12-06, Rev. 1.05

Previous Version: 2006-10-04, Rev. 1.04

Page	Subjects (major changes since last revision)
All Pages	Design Specification updates for Power Domains names & Electrical Characteristics (SMS00147105, SMS00429563, SMS00417279)
	Design Specification updates for RF macro (SMS00330373, SMS00254415, SMS00429579, SMS00323765)
	Design Specification updates for LDO (SMS00366822, SMS00414335, SMS00414336, SMS00415798)
	Design Specification updates for PMU macro (SMS00414331, SMS00418894, SMS00420316, SMS00425149, SMS00429427, SMS00429435, SMS00429437, SMS00425315)
	Design Specification updates for BootCode Utility routines (SMS00415783)
	Design Specification updates for E-Fuse Boxes (SMS00424075)
All Pages	Update Product Versions / RePrint of Book

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Overview of E-GOLDvoice

The E-GOLDvoice is a GSM baseband modem including RF transceiver covering the low bands GSM850 / GSM900 and high bands GSM1800 / GSM1900 bands. E-GOLDvoice is Dual Band, therefore, it supports by default a low / high pair of bands at the same time:

1. GSM850 / GSM1800
2. GSM850 / GSM1900
3. GSM900 / GSM1800
4. GSM900 / GSM1900.

The E-GOLDvoice is optimized for voice-centric Mobile Phone applications.

The E-GOLDvoice is designed as a single chip solution that integrates the digital, mixed-signal, RF functionality and a direct-to-battery Power Management Unit.

The transceiver consists of:

- Constant gain direct conversion receiver with an analog I/Q baseband interface
- Fully integrated Sigma/Delta-synthesizer capability
- Fully integrated two-band RF oscillator
- Two-band digital GMSK modulator with digital TX interface
- Digitally controlled crystal oscillator generating system clocks.

The E-GOLDvoice supports a direct battery connection, hence eliminating the need for an external Power Management Unit. The E-GOLDvoice has different power down modes and an integrated power up sequencer. The E-GOLDvoice is powered by the C166®S MCU and TEAKLite® DSP cores. The operating temperature range from -40°C to 85°C. It is manufactured using the 0.13 µm CMOS process.

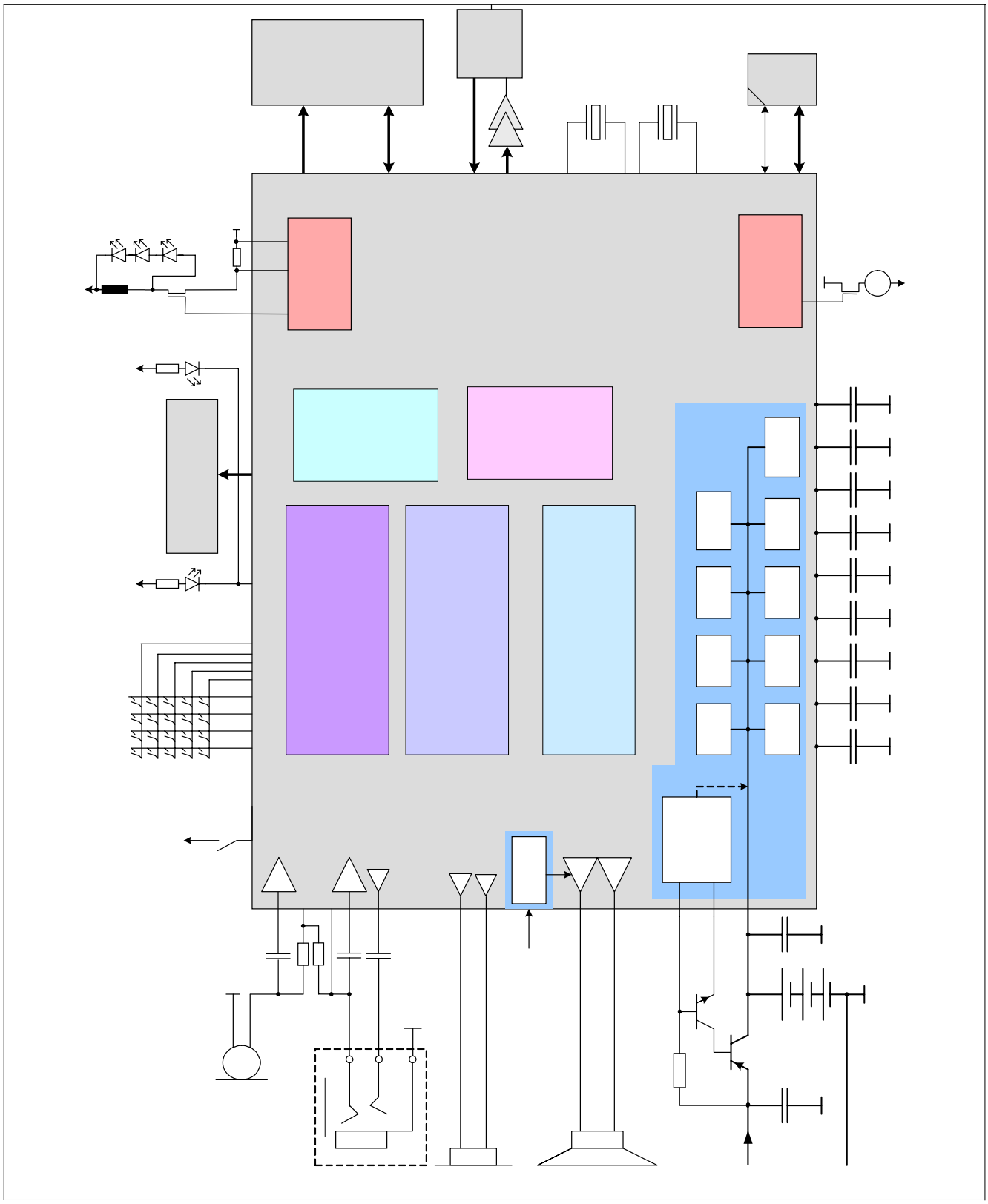


Figure 1 E-GOLDvoice Block Diagram

1.2 Features

Baseband

- High performance fixed-point TEAKlite DSP
- C166S high performance microcontroller
- There are several Interfaces:
  - I2S interface for DAI connections (for Tape Approval)
  - High Speed SSC Interface for connection of external peripherals
  - SIM Interface
  - Keypad Interface (6x4 or 5x5 keys)
  - EBU for external RAM/FLASH connection
  - Asynchronous serial interface (incl. IrDA support capability)
  - JTAG Interface
  - Black & white and color displays are supported
  - PWM source to drive vibrator
  - Keypad and display backlight supported.

Receiver

- Constant gain, direct conversion receiver with fully integrated blocking filter
- Two integrated LNAs
- No need of interstage and IF filter
- Highly linear RF quadrature demodulator
- Programmable DC output level
- Very low power budget.

Transmitter

- Digital Sigma-Delta modulator for GMSK modulation, typical -163.5 dBc/Hz @ 20 MHz
- Single ended outputs to PA, Pout = +3.5 dBm
- Very low power budget.

RF-Synthesizer

- ©–Synthesizer for multi-slot operation
- Fast lock-in times (< 150 ∝s)
- Integrated loop filter
- RF Oscillator
- Fully integrated RF VCO.

Crystal Oscillator

- Fully digital controlled crystal oscillator core with a highly linear tuning characteristic.

Mixed Signal and Power Management Unit

- DC/DC boost for voltages up to 15V for driving White or Blue LEDs
- 8-Ohm loud speaker driver (250/350mW)
- 16-Ohm earpiece driver
- 32-Ohm headset driver
- 4 measurement interfaces (PA temperature, battery voltage, battery temperature, and ambient temperature)
- Differential microphone input
- System start up circuitry
- Charger circuitry for NiCd, NiMh and LiIon cells
- Integrated regulators for direct connection to battery.

Package

E-GOLDvoice utilizes an LF2BGA-189-1 lead-free (green) package. The high degree of integration in E-GOLDvoice in conjunction with a sophisticated designed ball-out allows building a complete mobile phone complete with all its peripherals on a 4-layer PCB.

1.3 Application

E-GOLDvoice was developed for very low cost voice-only Dual Band GSM system solutions.

1.4 GSM System Description

Figure 2 shows a typical application circuit for the E-GOLDvoice.

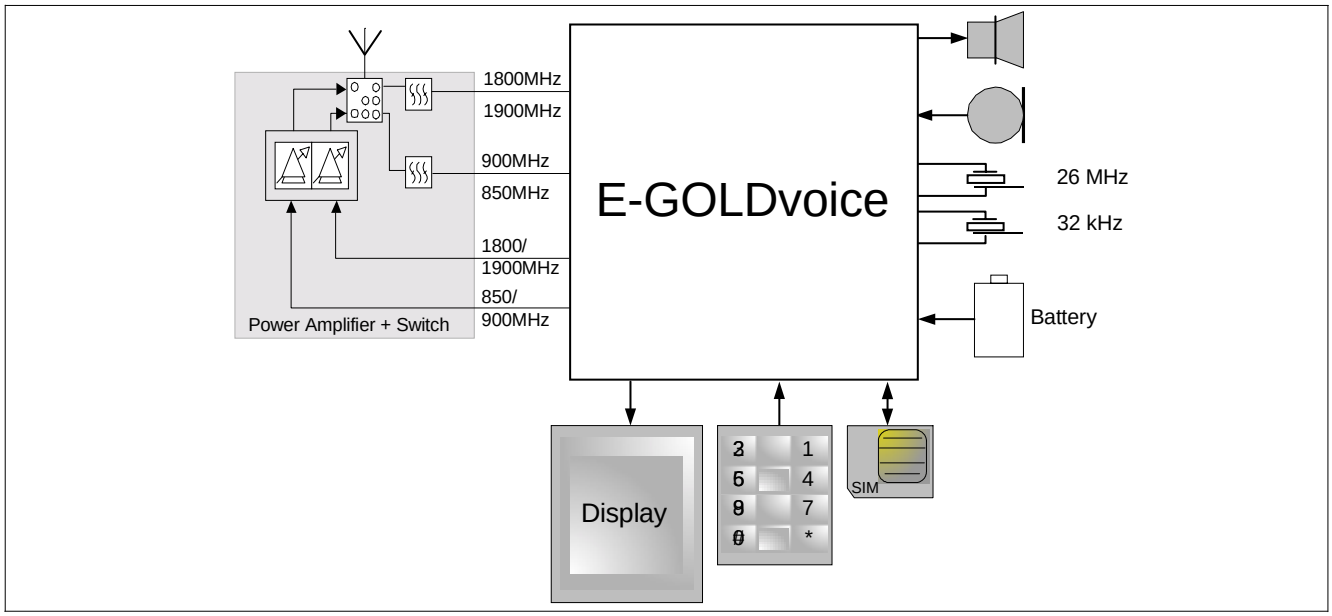


Figure 2 Typical E-GOLDvoice Application Circuit

The E-GOLDvoice is suited for mobile stations operating in the GSM850/900/1800/1900 bands.

In the receiver path the antenna input signal is converted to the baseband, filtered, and then amplified to target level by the RF transceiver chipset. Two A-to-D converters generate two 6.5 Mbit/s data streams. The decimation and narrowband channel filtering is done by a digital baseband filter in each path. The DSP performs:

1. The GMSK equalization of the received baseband signal (SAIC support available)
2. Viterbi channel decoding supported by an hardware accelerator.

The recovered digital speech data is fed into the speech decoder. The E-GOLDvoice supports fullrate, halfrate, enhanced fullrate and adaptive multirate speech CODEC algorithms.

The generated voice signal passes through a digital voiceband filter. The resulting 4 Mbit/s data stream is D-to-A converted by a multi-bit-oversampling converter, postfiltered, and then amplified by a programmable gain stage. The output buffer can drive a handset ear-piece or an external audio amplifier, an additional output driver for external loud speaker is implemented.

In the transmit direction the differential microphone signal is fed into a programmable gain amplifier. The prefiltered and A-to-D converted voice signal forms a 2 Mbit/s data stream. The oversampled voice signal passes a digital decimation filter.

The E-GOLDvoice performs speech and channel encoding (including voice activity detection (VAD) and discontinuous transmission (DTX)) and digital GMSK modulation.

In the RF transceiver part, the baseband signal modulates the RF carrier at the desired frequency in the 850 MHz, 900 MHz, 1.8 GHz, and 1.9 GHz bands using an I/Q modulator. The E-GOLDvoice supports dual band applications.

Finally, an RF power module amplifies the RF transmit signal at the required power level. Using software, the E-GOLDvoice controls the gain of the power amplifier by predefined ramping curves (16 words, 11 bits).
For baseband operation, the E-GOLDvoice supports:

- Making or receiving a voice call
- Sending or receiving an SMS.

1.5 PMU Details

The E-GOLDvoice includes battery charger support (various sensor connections for temperature, battery technology, voltage, etc.) and a ringer buffer.

E-GOLDvoice avoids the need for an external power management component because its internal power management unit contains:

- Voltage regulators for the On-chip and Off-chip functional blocks
- Charger circuitry for NiCd, NiMh and LiIon cells.

1.6 Bus Concept

The E-GOLDvoice has two cores (a microcontroller and a DSP), each with its own bus.

There is an interconnection between the TEAKlite bus and the C166S X-Bus.

1.6.1 C166S Buses

The C166S is connected to three buses:

1. Local Memory (LM) bus
2. X-Bus
3. PD-Bus.

1.6.2 TEAKLite Bus

The TEAKlite is connected to the TEAKlite bus.

GSM Cipher Unit

This unit on the TEAKlite bus calculates the GSM/EDGE encryption keystream and the GSM/EDGE decryption keystream. It implements the following algorithms: A5/1, A5/2, and A5/3.

1.6.3 Bus Interconnections

The interconnection between the X-Bus and the TEAKlite Bus uses:

- Multicore Synchronization
- Shared Memory.

1.7 Clock Concept

The E-GOLDvoice has a flexible clock control.

1.8 Interrupt Concept

The C166 MCU carries out the E-GOLDvoice interrupt system.

1.9 Debug Concept

The E-GOLDvoice includes a multi-core debug. The C166 and TEAKlite cores can be debugged in parallel with:

- A single JTAG port (that is, on a single host)
- Mutual breakpoint control.

1.9.1 C166 Debug Concept

The debugging of the C166 uses the OCDS and the Cerberus.

1.9.2 TEAKLite Debug Concept

TEAKlite debugging uses the OCEM and the SEIB.

1.10 Power Management

The E-GOLDvoice provides the power management unit (PMU) for the complete mobile phone application. The integrated PMU is directly connected to the battery and provides a set of linear voltage regulators (LDO's). These LDO's generate all required supply voltages and currents needed in a low feature mobile phone.

A charger control circuit charges NiCd, NiMH and LiIon batteries. The charger control supports hardware controlled pre-charging and software controlled charging. It offers a wide charger voltage range, making half-wave/full-wave charging with cheap transformers possible.

White/blue backlight generation is supported with a special driver for very a low external parts count.

Power consumption during operation phases is minimized due to flexible clock switching

In the Standby Mode most parts of the device are switched off, only a small part is running at 32kHz and the controller RAM is switched to a power saving mode. The TEAKLite ROM can be switched off during Standby via SW.

1.11 On-Chip Security Concept

Secure boot is based on a public/private key approach. Flash images that are not signed with the private key during phone manufacture cannot be loaded. Verification of the Flash code is done with the public key. The public key as well as hash and verify algorithms are stored in the ROM, which ensures a hardware secured boot procedure.

The following security features are supported:

- Prevention of illegal Flash programming
- Flash programming makes use of the E-GOLDvoice ID for personalization checks with IMEI and SIM-lock protection

The security features use the following mechanism:

- Boot ROM flow:
  - Controls the boot transition to external flash
  - Controls the flash update
- Flash tied to the individual chip via an ID using e-fuses, that is, each E-GOLDvoice chip has its own fused ID.

Further details on the E-GOLDvoice security concept are not publicly documented.

1.12 Asynchronous Operation Mode Concept

The E-GOLDvoice can operate in either:

- The traditional synchronous mode with the 26MHz system clock synchronized on the base station
- A special asynchronous mode (XO concept).
  In the asynchronous mode the 26MHz clock input is not synchronized with the base station; the residual frequency offset is compensated in the digital signal processing domain. This processing includes frequency and timing compensation of the baseband and voiceband signals.

1.13 Product Versions & Namings

Table 1 E-GOLDvoice Version Naming

Product Code and Version	FW Mask Code	Boot Code versions	Package Label[1)]	Description
PMB 7880 V1.1	M10	V1.9	7880 1.1 .xx Gwwyy	First Version
PMB 7880 V1.3G	M12	V2.2	7880 V1.3G xx . Gyyww	Final Version

1) See **Figure 3 Package Label** for a full describtion of the package label.

1.14 Package

The package is SG-LF2BGA-189-1 (a “Green flipchip” with 189 pins), see **Figure 3** (Character height = 1.0 mm).

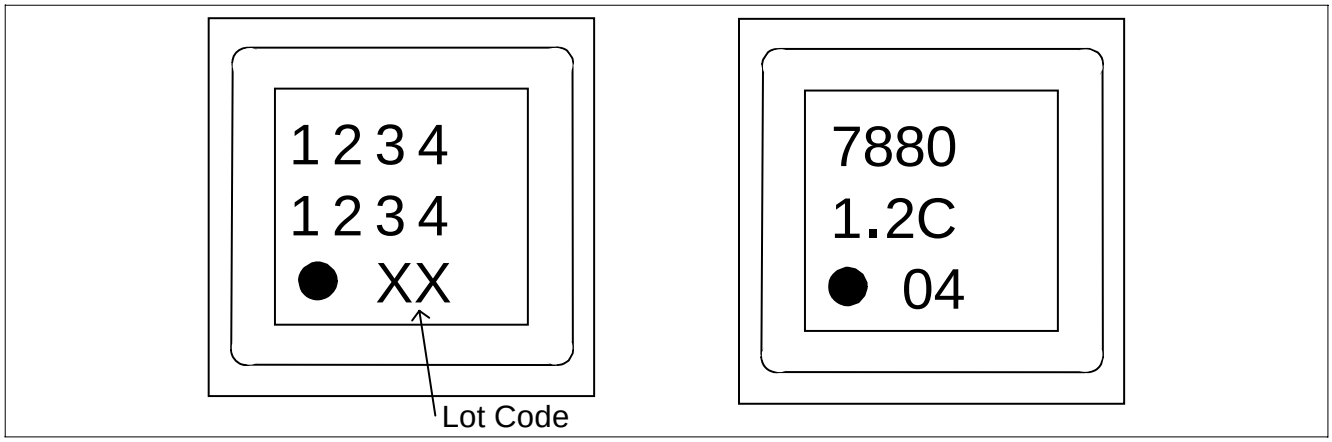


Figure 3 Package Label

Table 2 Legend for Figure 3

		e.g. E-GOLDVoice PMB7880 V1.2C Engineering Samples
1234	PMB number	7880
1234	Product version 1= Major full mask redesign 2= Separator (dot) 3= Firmware mask 4 = Minor redesign (metal bug fix)	 1 . 2 C
XX	XX is for Lot Code	04
GYYWW	G is for the green package	G
	YY for year number modulus 100. For Engineering Samples the first Y is replaced by the letter "E".	06 (for Productive Devices) E6 (for Engineering Samples)
	WW for week number in the year	38

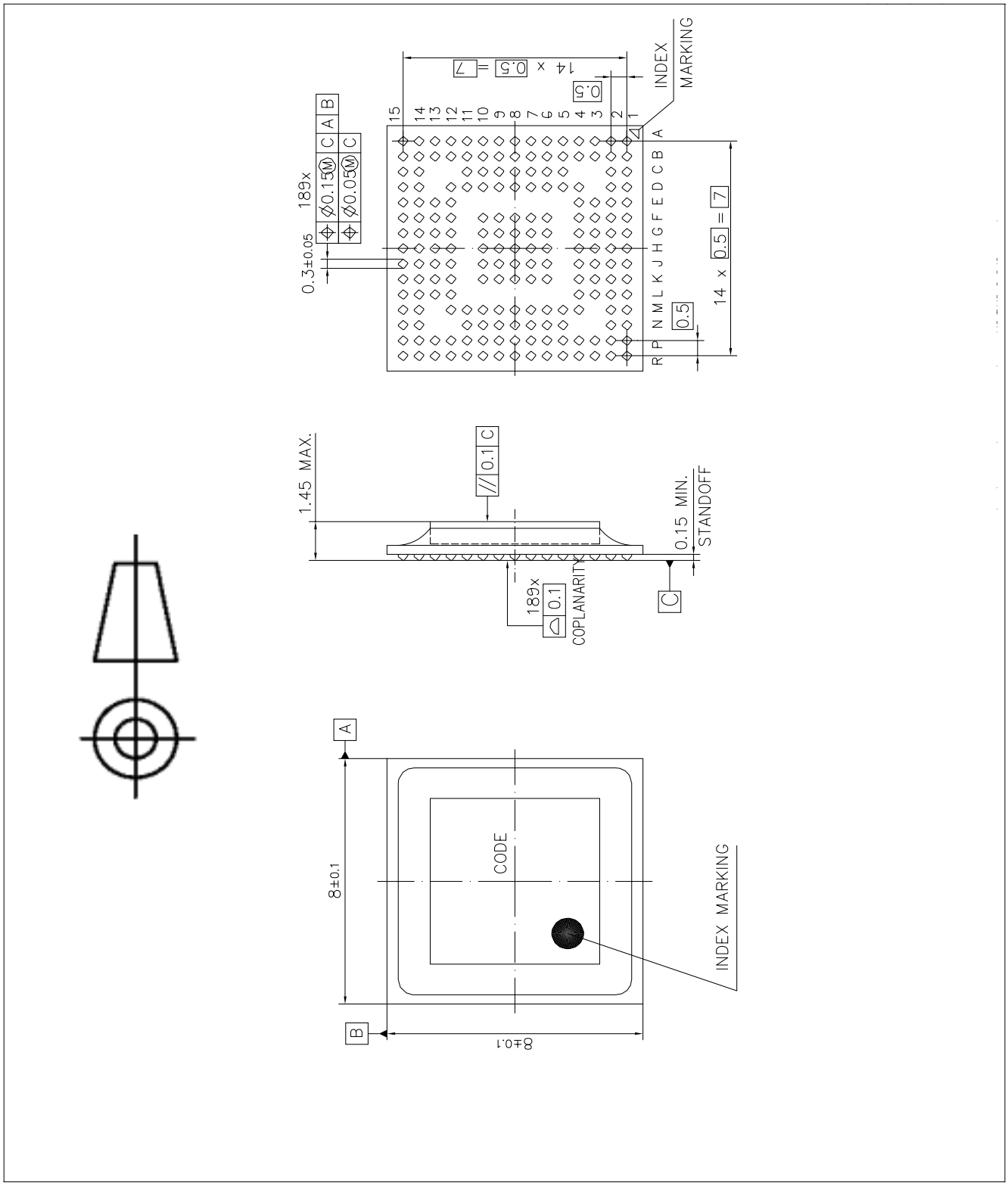


Figure 4 E-GOLDvoice: SG-LF2BGA-189-1 Package Details and Ball Labelling

2 Pin Diagram

2.1 E-GOLDvoice Ballout

Figure 6 shows the location of the balls on the E-GOLDvoice package and their associated signals and voltages. The ballout view in **Figure 6** is from above the chip, that is, the ball connections on the mobile motherboard.

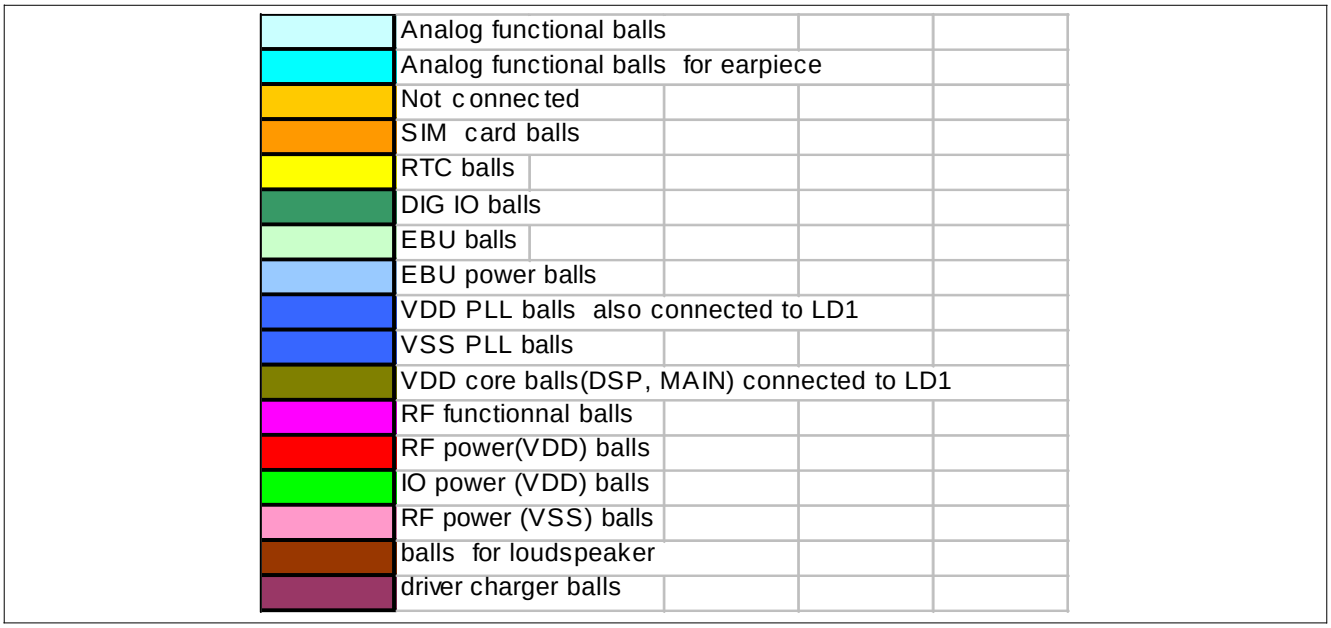


Figure 5 Legend for E-GOLDvoice Ballout

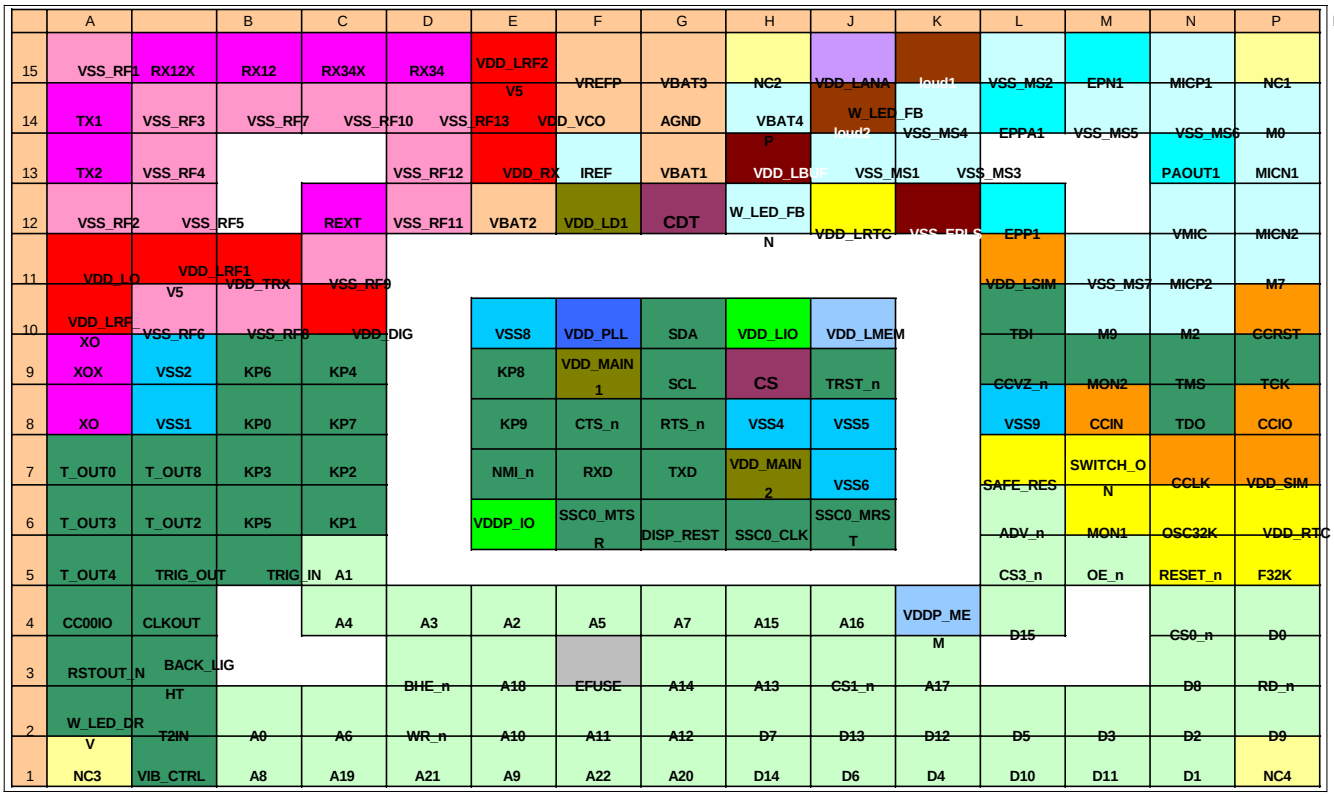


Figure 6 E-GOLDvoice Ballout Viewed from Above the Chip

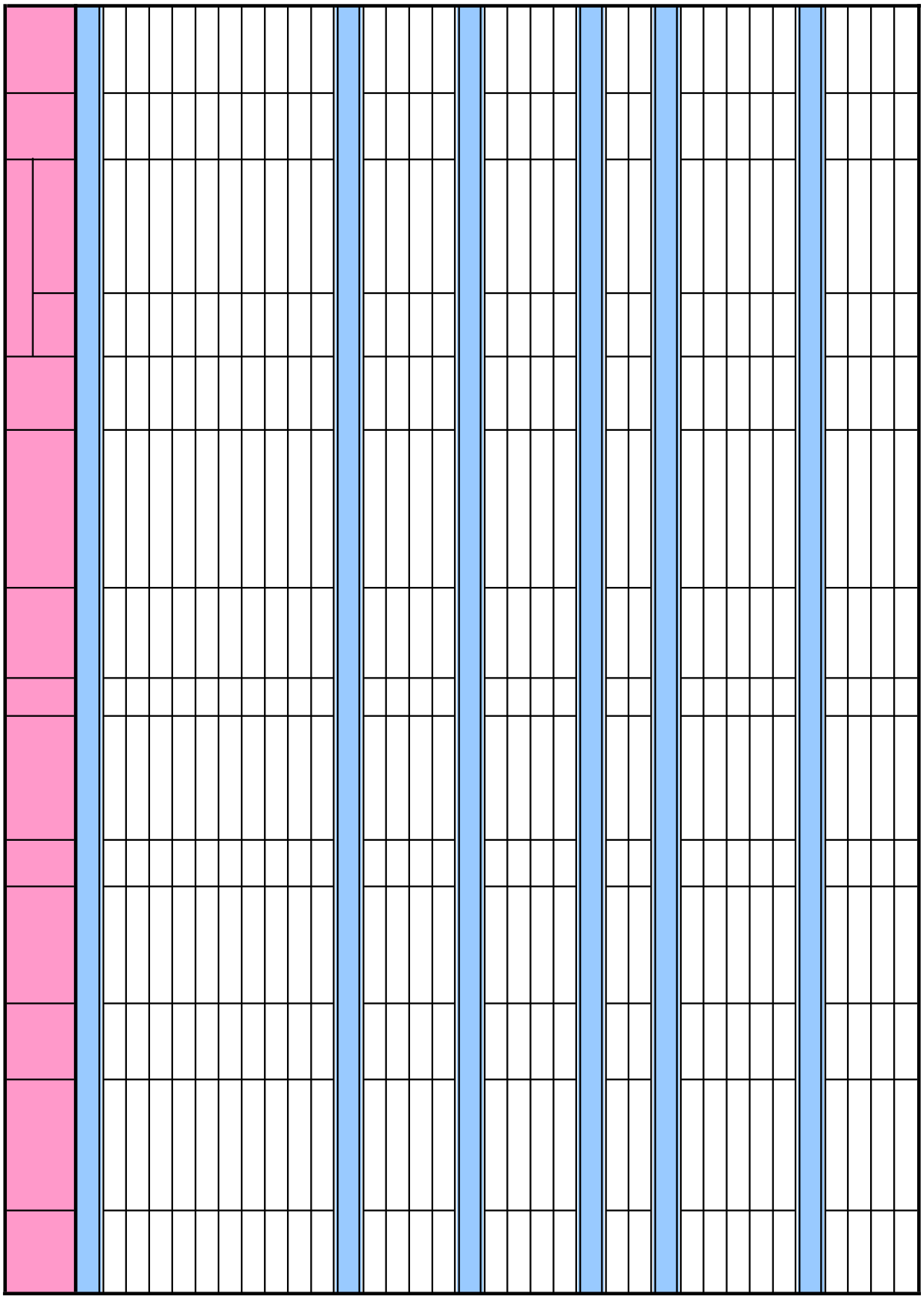
3 Pin Descriptions

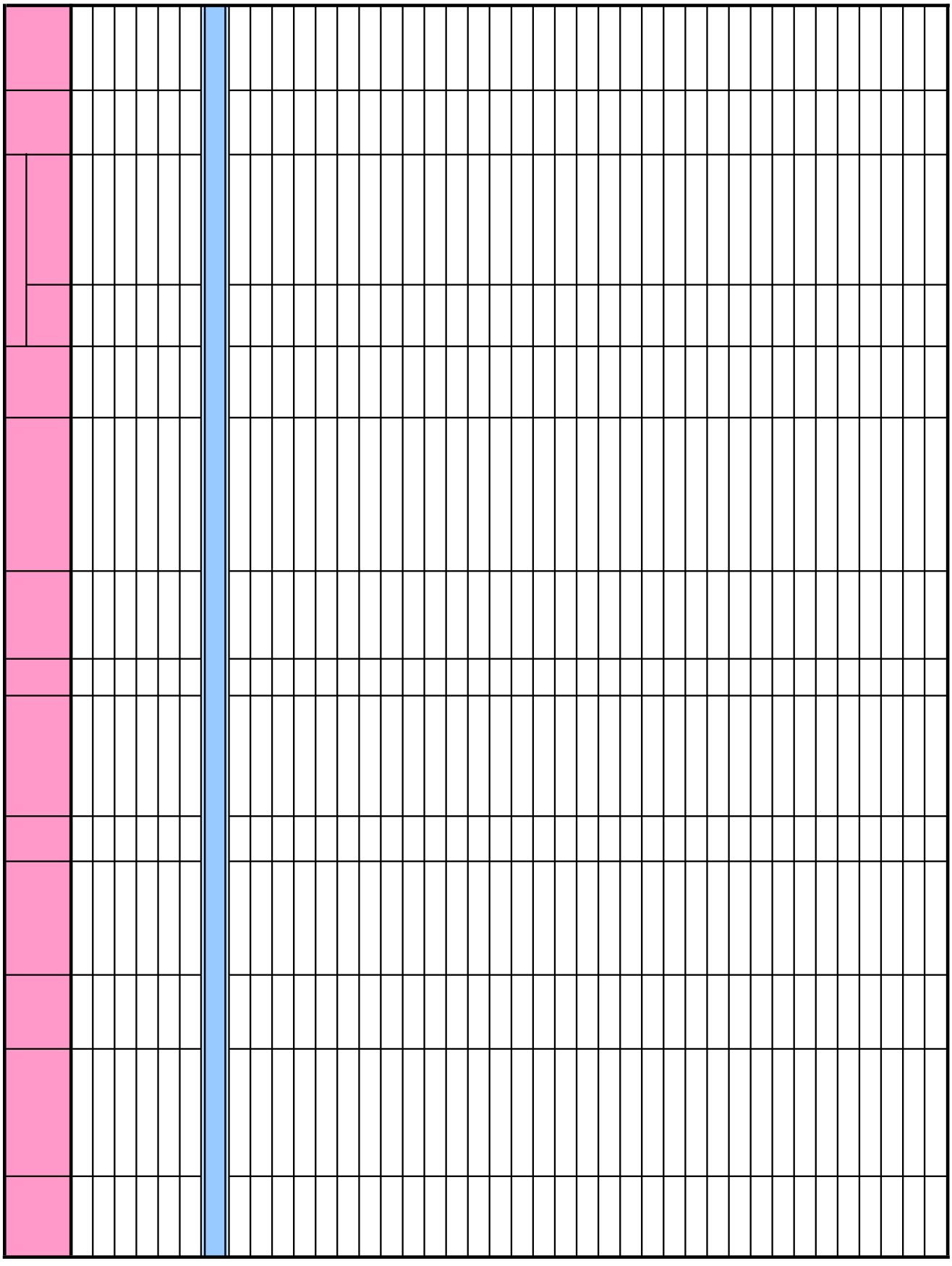
Table 3 describes the E-GOLDvoice ballout.

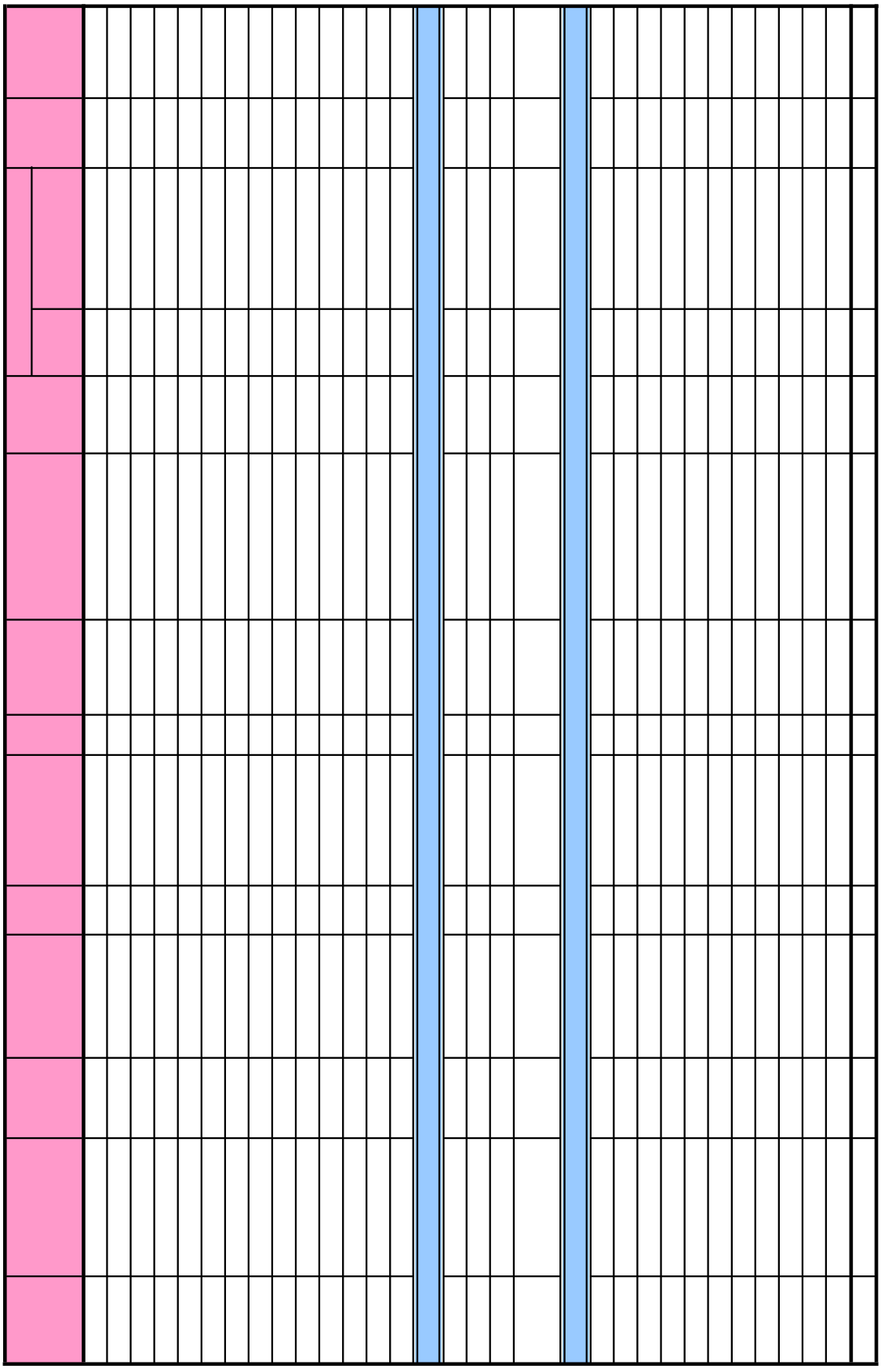
Legend for **Table 3**:

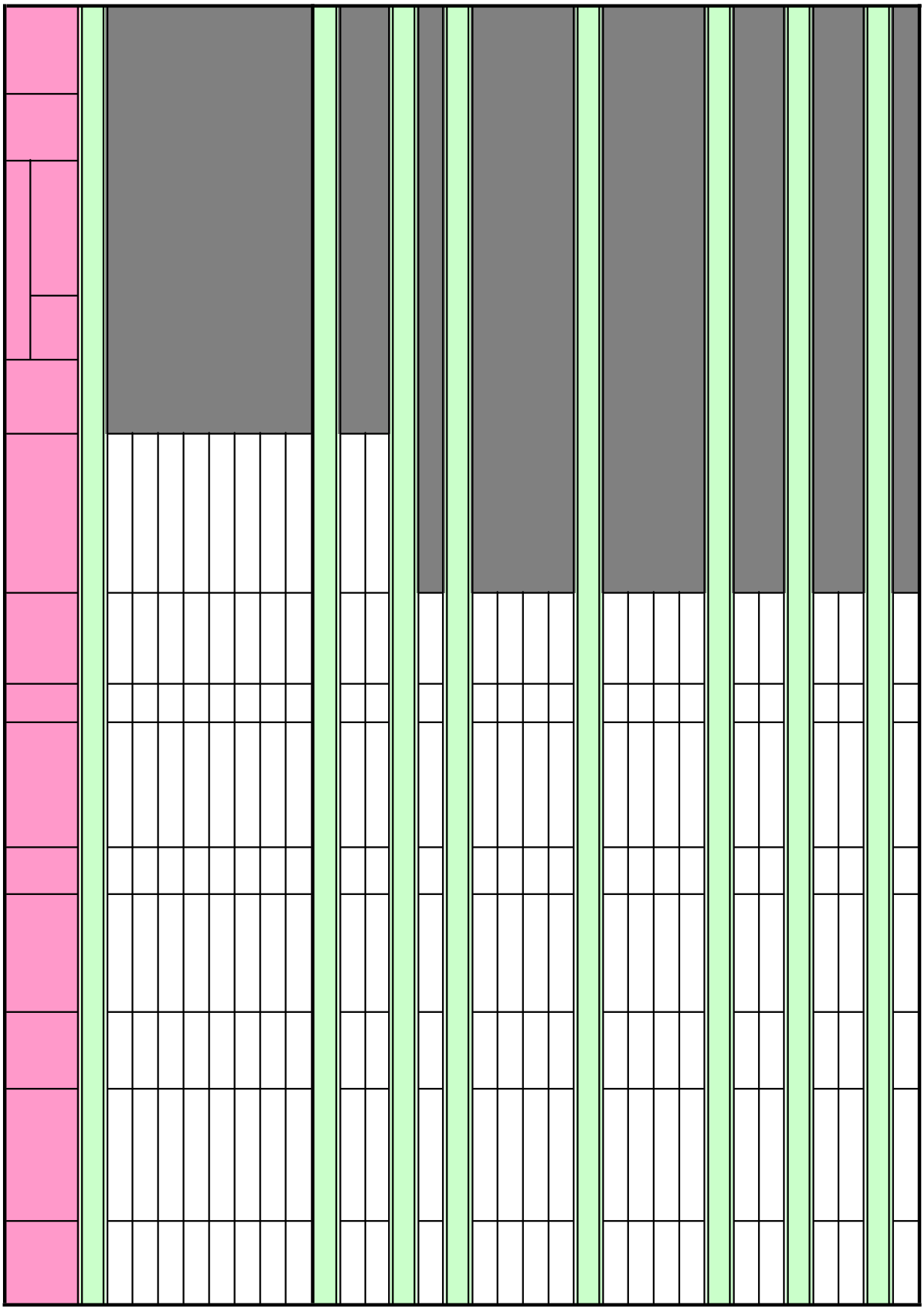
Name	Definitions
ST:	Schmitt Trigger (Input pads)
BU:	Buffer (Output pads)
BU_I2C:	I^2C buffer
PU+PD only:	Only Pull-Up and Pull-Down
Fixed PD:	Fixed at Pull-Down
Fixed PU:	Fixed at Pull-UP **Programmable:** The pin be programmed (yes/no)
T:	Tristate

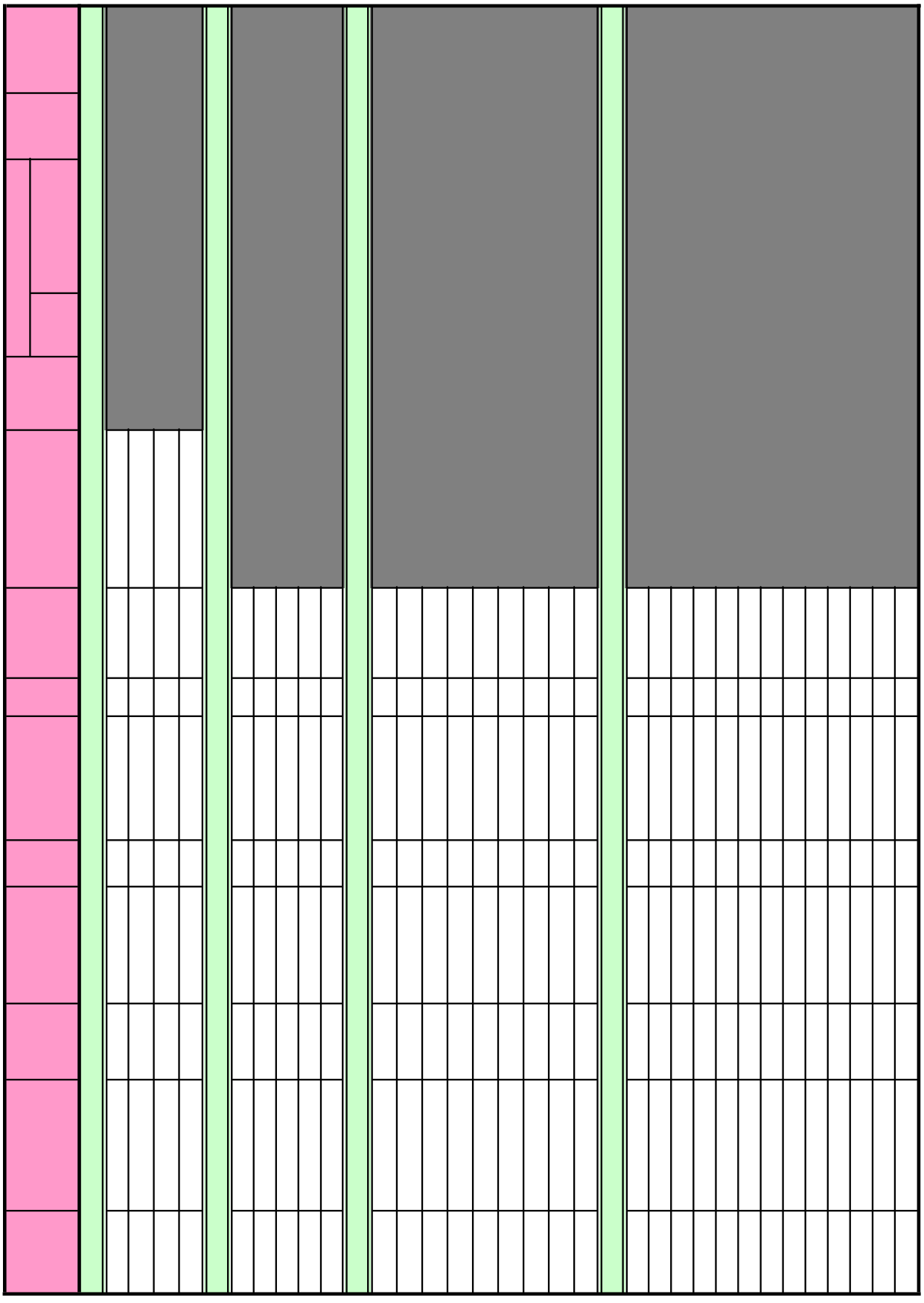
Table 3 Pin Descriptions

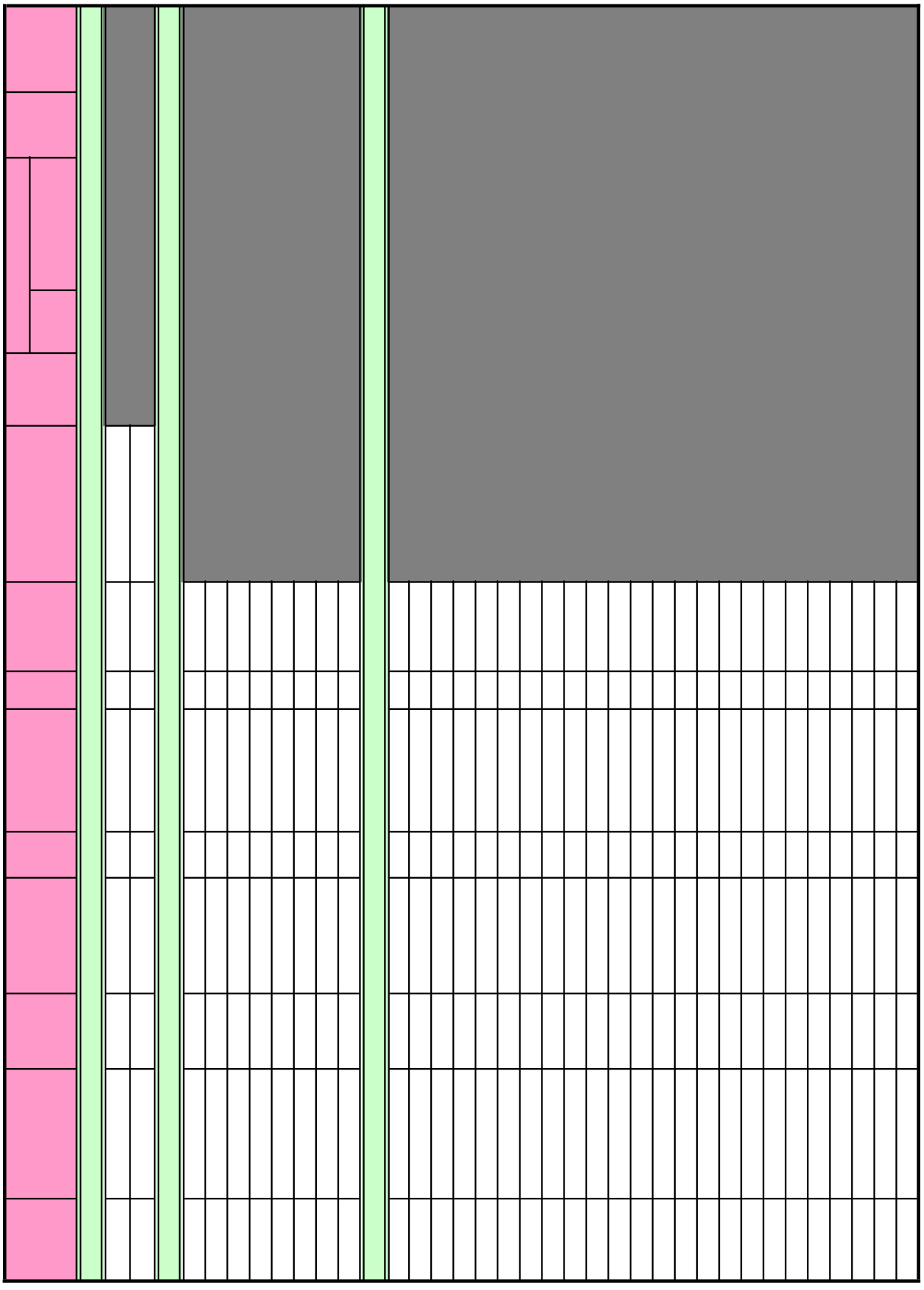












4 Bus Overview

Figure 7 gives an overvew of the E-GOLDvoice functional blocks and its buses.

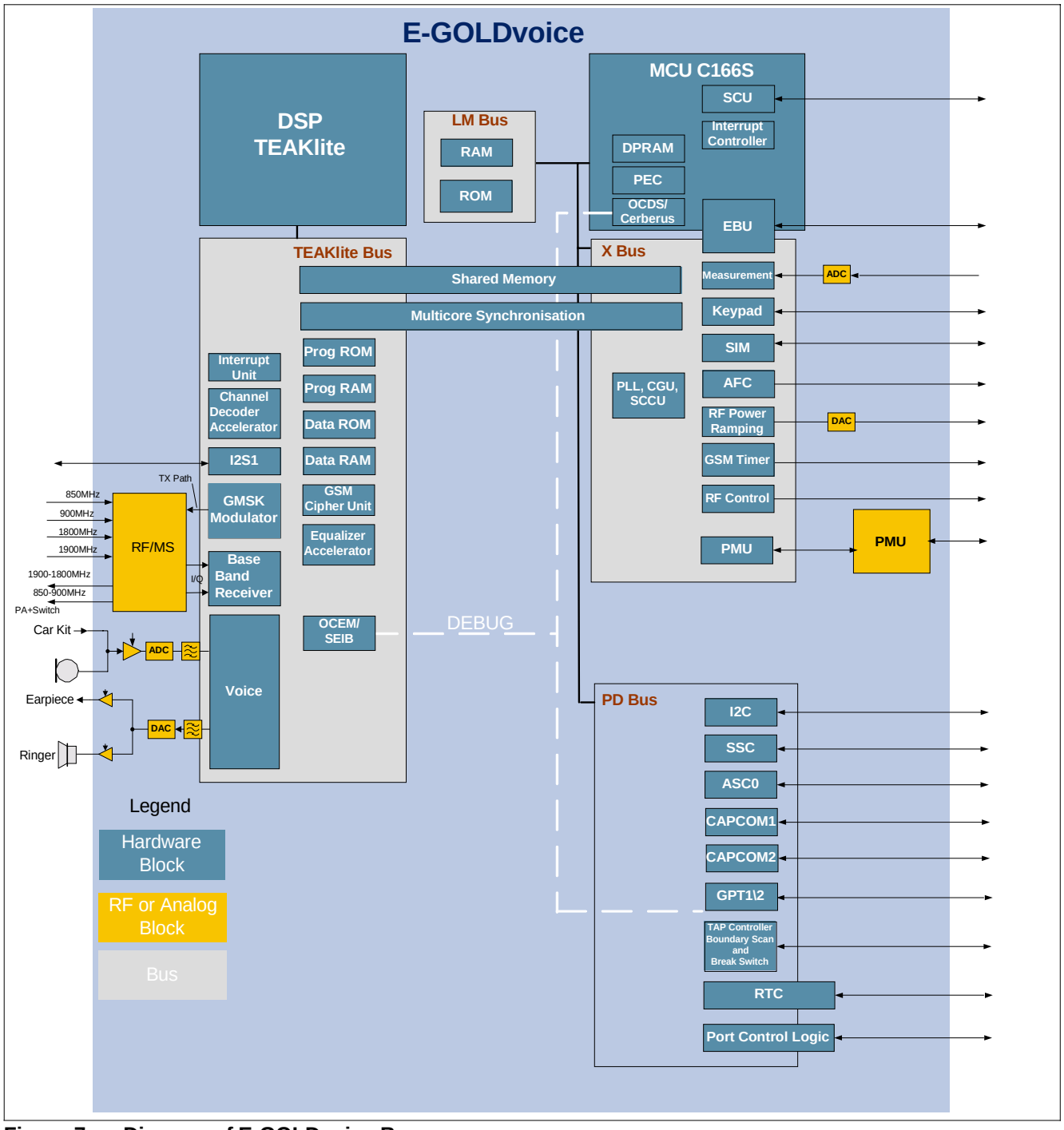


Figure 7 Diagram of E-GOLDvoice Buses

5 Digital and Mixed-Signal Subsystems

5.1 TEAKLite

The TEAKLite core has 16-bit data and 16-bit program memory accesses, a high performance fixed-point DSP core, and low power consumption.

The core consists of a high performance processing unit including a full featured Bit-Manipulation Unit, RAM and ROM addressing units, and program control logic. This core has an improved set of DSP and general microprocessor functions to meet application requirements. The programming model and instruction set are optimized for generation of efficient and compact code.

The Computation Unit consists of a 16 by 16 multiplier unit with a 32-bit product and a 36-bit ALU with two accumulator registers.

The Bit Manipulation Unit consists of a full 36-bit barrel shifter, an exponent unit, a bit-field operation unit, two 36-bit accumulator registers, and a shift value register.

The Data Address Arithmetic Unit performs all the address storage and calculations necessary for accessing the data and program memories. It also supports a software stack pointer, loop counter, and min/max operations.

The Program Control Unit performs instruction fetches and decoding, exception handling, and hardware loop control. This unit is provided with three independent input interrupt lines.

Key Features of the TEAKLite Core

- 16-bit fixed-point DSP core
- 16x16-bit 2’s complement parallel multiplier with a 32-bit product
- Single cycle multiply-accumulate instructions
- 36-bit ALU
- 36-bit left/right barrel shifter
- Four 36-bit accumulators
- Software stack residing in the data RAM
- User-defined registers off-core
- Three high-active interrupt input lines
- Automatic context switching by interrupts
- Up to 16-bit Bit Field Operations (BFO)
- Three modes for power saving features: Operational, Idle, and Sleep.

Clock

The maximum Frequency is 78 MHz. The TEAKlite core clock is scalable to lower frequencies.

5.2 C166S MCU

The C166S is a 16-bit CMOS microcontroller containing a Central Processing Unit core (the MCU) and a set of peripherals.

The architecture of the MCU combines both RISC and CISC architecture.

Key Features

- High Performance 16-Bit MCU with a four-stage pipeline:
  - Using a 26MHz clock:
    a)76.9 ns minimum instruction cycle time with most instructions executed in 1 cycle (2 clock ticks)
    b)256 ns multiplication (16-bit x 16-bit), 512 ns division (32-bit/16-bit)
  - Parallel use of multiple high bandwidth internal data buses
  - Register based design with multiple variable register banks

  - Single cycle context switching support
  - 16 MBytes linear address space for code and data (von Neumann architecture)
  - System stack cache support with automatic stack overflow/underflow detection.
- Control Oriented Instruction Set with High Efficiency:
  - Bit, byte, and word data types
  - Flexible and efficient addressing modes for high code density
  - Enhanced boolean bit manipulation with direct addressing of 6 Kbits for peripheral control and user defined flags
  - Hardware traps to identify exception conditions during runtime
  - HLL support for semaphore operations and efficient data access.
- External Bus Interface:
  - Demultiplexed bus configurations
  - Segmentation capability and chip select signal generation
  - 8-bit or 16-bit data bus
  - Bus cycle characteristics selectable for four programmable address areas.
- 16-Priority-Level Interrupt System:
  - Up to 112 interrupt nodes with separate interrupt vectors
  - 16 priority levels and 8 group levels.
- 16-Channel Peripheral Event Controller (PEC):
  - Interrupt driven single cycle data transfer
  - Transfer count option (using a standard MCU interrupt after programmable number of PEC transfers)
  - Long Transfer Counter
  - Channel Linking
  - Eliminates overhead for saving and restoring system state for interrupt requests.
- DPRAM:
  - Internal 16-bit dual port RAM with a 1K x 16-bit size.
- SCU (System Control Unit):
  - Handles the boot and sleep modes of the core
  - Provides a watchdog timer.

Debug

The On-Chip Debug Support (OCDS) and Cerberus provide facilities for the debugger to emulate resources and assist in application program debugging. Their main features are:

- Real time emulation
- Extended trigger capability, including: instruction pointer events, data events on address and/or value, external inputs, counters, chaining of events, timers, etc....
- Software break support
- Break and “break before make” (on IP events only)
- Simple monitor mode or JTAG based debugging through instruction injection.

The OCDS is connected to the Break Switch block and triggers the C166S core.

The Cerberus is connected to the external JTAG port through the TAP Controller/Boundary Scan block. It allows standard debug controls.

Clock

The maximum Frequency is 26 MHz.

5.3 TEAKlite Bus

The TEAKlite bus has 16-bit data and addresses; the address bus has a multi-page access mode. The TEAKlite bus has a 2-wait-state access for non-memory peripherals.

5.3.1 Interrupt Unit

The Interrupt Unit provides interrupt arbitration with different priority levels using a standard interrupt mechanism. The interrupt unit is connected to the TEAKLite DSP core with three high-active interrupt lines INT0, INT1, and INT2. INT0 has the highest priority.

The TEAKLite core also provides basic control of the three interrupt lines through internal TEAKlite registers.

5.3.2 Channel Decoder Accelerator

The Channel Decoder Hardware Accelerator provides very fast calculation of branch metrics, butterflies, states, and final metrics for trellis processing. The input of the Channel Decoding Hardware Accelerator consists of soft input values; the output consists of trace back values for each timestamp.

5.3.3 I2S 1 (DAI)

The bi-directional I^2S interface is used only for DAI mode testing and supports.

- Bi-directional transmission with independent transmitter and receiver
- I^2S mode
- DAI mode compatible with GSM Spec 11.10.

5.3.4 Baseband Receiver

This block contains the ADC converters, low pass filters, flicker noise suppression, a I/Q imbalance correction stage, a DC compensation stage, and the CORDIC processor.

This block supports the Zero IF received signal.

5.3.5 Voice

This block contains the audio front end (filters, sampling hardware, ...).

It uses an oversampling, multi-bit, digital-to-analog converter for good linearity and a high signal-to-noise ratio.

Key feature of Voice Blocks

- Three audio input sampling frequencies: 8kHz, 16kHz, and 32kHz
- Two microphone inputs with variable voltages
- Two speaker outputs
- One integrated loudspeaker driver
- Mono voice system (stereo not supported).

5.3.6 GSM Cipher Unit

This block performs GSM data encryption and decryption. It implements the A51, A52, and A53 algorithms.

The Ciphering Unit simultaneously computes:

- An encryption bit stream for an uplink burst
- A decryption bit stream for a downlink burst.

The output encryption/decryption bit stream are XOR combined with the uplink/downlink data bits by the DSP.

5.3.7 Equalizer Accelerator

The Equalization Hardware Accelerator provides very fast calculation of branch metrics, butterflies, states and final metrics for trellis processing, and soft outputs. The input of the Equalization Hardware Accelerator consists of the received values and channel coefficients; the output consists of soft output values.

5.3.8 Shared Memory

The size of the shared memory is 0.75k x 16-bits.

5.3.9 Program ROM

The size of the Program ROM (PROM) is 72 kwords:

The PROM is switched off during TEAKlite core idle gaps to reduce power consumption.

5.3.10 Program RAM

The size of the Program RAM (PRAM) is 1K x 16-bits.

5.3.11 Data ROM

The size of the Data ROM (DROM) is 42K x 16-bits:

The DROM is switched off during TEAKlite core idle gaps to reduce power consumption.

5.3.12 Data RAM

The size of the Data RAM (DRAM) is 22.5K x 16 bits (19.5K x 16 bits for the XRam and 3K x 16 bits for the YRAM).

5.3.13 Multicore Synchronization

This block includes 16 semaphores, 16 communication flags, and the Bus Interface Unit (BIU).

The semaphores and communication flags are accessible in read and write from the TEAKlite and C166S cores.

The BIU permits the TEAKlite bus to read and write the registers located in the controller memory space.

5.3.14 OCEM/SEIB

OCEM

The One Chip Emulator Module (OCEM) is a stand alone unit that provides emulation capability for an TEAKLite core and supports the following emulation functions:

- Program address break-point
- Data address break-point (for single or multiple locations)
- Data value break-point
- Combined data address and data value break-point
- External break-point
- Single step
- Instruction break-point
- Interrupt break-point
- BKREP loop break-point
- Program flow trace buffer.

The OCEM is connected to the Break Switch block (refer to **“Break Switch” on Page 51**), it triggers the TEAKlite core.

SEIB

The Serial Emulator Interface Block (SEIB) is connected to the external JTAG port through the TAP Controller and Boundary Scan and Break Switch block (refer to **“TAP Controller & Boundary Scan and Break Switch” on Page 51**), which allows standard debug control.

5.4 LM Bus

The Local Memory (LM) Bus is a 16-bit data read-write/32-bit program read bus, which allows 32 bits to be fetched in parallel from the program code in the internal local memory.

5.4.1 RAM

The RAM is a 1.75M-bit SRAM that contains the MCU data.

The LM bus access to this RAM has 0 wait states when using a 26MHz MCU clock.

5.4.2 ROM

The ROM contains the needed code to boot the MCU.

The ROM size is 10K x 16 bits.

The LM bus access to this RAM has 0 wait states when using a 26MHz MCU clock.

5.5 X-Bus

The X-Bus is compliant with the XBus+ specifications. This interface is a general purpose bus interface with a 24-bit address bus and a 16-bit data bus that allows peripherals and memories to be connected to the C166S.

X-Bus Key Features

- Up to 4 MBytes of address space (via a 24-bit address bus) shared with the external bus
- Up to 6 address ranges with a dedicated bus configuration
- Synchronous XBus+ protocol
- Minimum cycle time: 2 positive clock periods
- Split I/O data buses (8-/16-bit) supporting byte and word access
- Up to 15 programmable and dynamic wait state mechanisms
- Multi-master capabilities allowing another bus master to access the C166S peripherals connected on the XBus+ (in the XPER-SHARE mode).

5.5.1 Shared Memory

Refer to **“Shared Memory” on Page 42**.

5.5.2 Multicore Synchronization

Refer to **“Multicore Synchronization” on Page 42**.

5.5.3 PLL - CGU - SCCU

This block contains the Phase Locked Loop (PLL), Clock Control Unit (CGU), and the Standby Clock Control Unit (SCCU).

The PLL provides the masters clock used by the CGU.

The CGU provides a flexible clocking scheme with clock pre-scalers and clock control.

The SCCU performs the power consumption management that provides minimized power dissipation during the standby and sleep modes.

5.5.3.1 Clocks

The input frequency of 26MHz and 32.768kHz will be generated internally by connecting external crystal (see **Figure 8**).

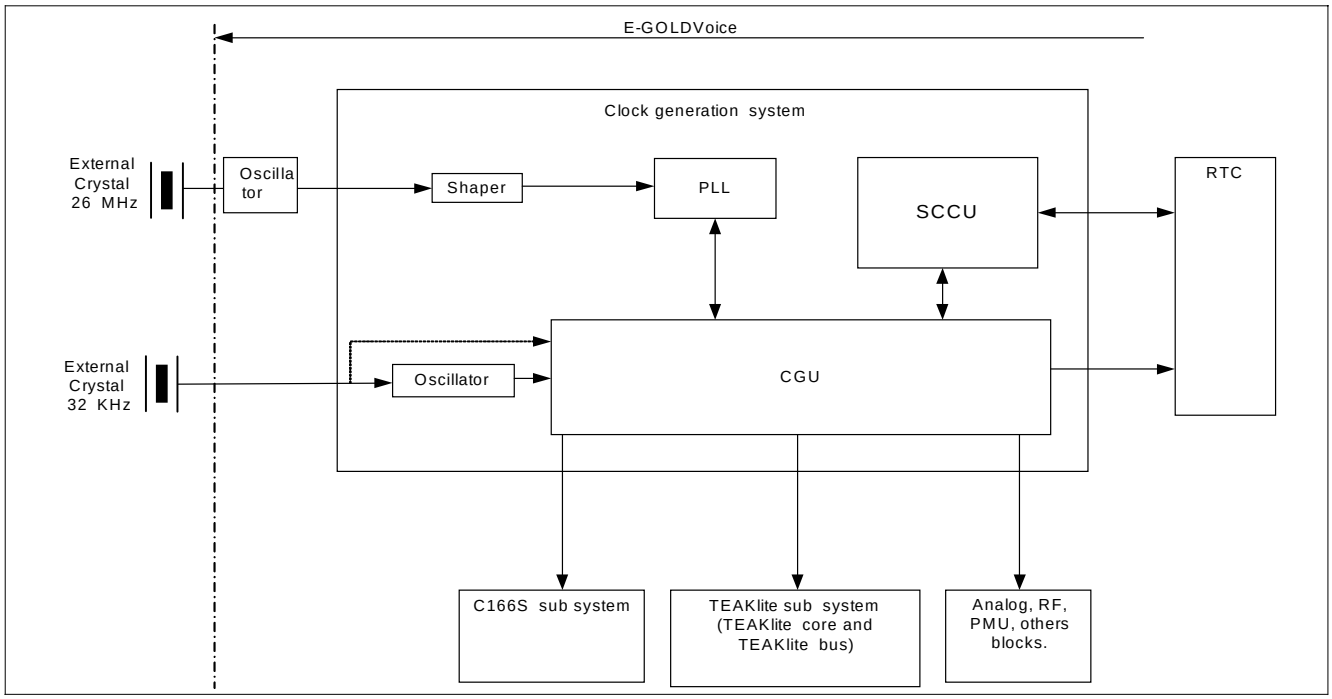


Figure 8 Clock Distribution

5.5.4 Measurement

The measurement block provides the following general purpose measurements:

- Battery voltage (VBAT)
- Battery temperature (TBAT)
- Environmental temperature (TENV)
- Power of power amplifier (PWPA)

The general purpose measurement inputs can be configured for voltage measurements with the following characteristics:

- Single-ended, voltage mode, referred to the internal AGND
- Single-ended, current mode, referred to the internal AGND
- Single-ended, voltage mode, referred to an external voltage
- Single-ended, current mode, referred to an external voltage drop (resistor)
- Fully differential, voltage mode.

This block also provides on chip measurements of:

- On chip temperature (TIC)
- Static offset of power ramp output (PAOUTOF1/2)
- Measurement offset self-calibration (OFS).

Note: A separate analog-to-digital converter (ADC) is used.

Various measurements with different characteristics can be combined. The measurement modes are adjusted via analog switches controlled by the Measurement Control Register, which provides information about the status of the measurement interface. Data is transferred via the Measurement Data Registers. All registers are part of the measurement interface X-Bus interface.

5.5.5 Keypad

The keypad interface is a peripheral that can be used for scanning keypads that have up to 5x5 or 6x4 matrix.

5.5.6 SIM Interface

The SIM Card Interface is a customized USART with additional features:

- For use as a 1.8 or 3 Volt Subscriber Identity Module (SIM) as specified by ISO 7816
- Support of speed enhancement as specified in GSM 11.11 Phase 2+ with baud rates up to 100 kBaud. When the SIM card is pulled, the chip card interface powers down automatically the electrical contacts to the SIM card.

The Character and T=0 modes are supported. The T=1 mode is not supported.

5.5.7 RF Power Ramping

The RF power control is carried out via the transmit power ramping path.

The shape of the power up and down ramping is digitally controlled by the GSM System Interface. The linear ramp shape in the active part of the burst is programmable.

The analog output is generated by the RF power ramping DAC with a 11-bit resolution and an analog post filter.

5.5.8 GSM Timer

The GSM Timer schedules periodic events within a TDMA frame. It provides high flexibility in scheduling and executing events using a programmable RAM table and is able to generate timing and trigger signals and interrupts for both the CPU and DSP, it will have only 5 outputs used as external.

5.5.9 EBU

The External Bus Unit provides an external RAM/ROM access to the MCU. Up to 8 MBytes of external RAM and/or ROM can be connected to the MCU via its external bus interface. NOR Flash memory is supported, but not NAND Flash memory.

The page mode is only supported for flash memories (NOR flash).

The integrated Bus Controller allows the access of external memory and/or peripheral resources in a very flexible manner. It can be programmed either to the Single Chip Mode when no external memory is required or to one of two different external memory access modes:

- 16-/18-/20-/22-bit Addresses, 16-bit Data, Demultiplexed
- 16-/18-/20-/22-bit Addresses, 8-bit Data, Demultiplexed.

Important timing characteristics of the external bus interface (Memory Cycle Time, Memory Tristate Time, Length of ALE (ALE is an internal signal), and Read/Write Delay CS and WR) have been made programmable to allow use of a wide range of different types of memories. In addition, different address ranges may be accessed with different bus characteristics.

Up to 3 external CS signals (and one additional CS as alternate pad function) can be generated to save external glue logic. Access to very slow memories is supported via a special ‘Ready’ function.

5.6 PD Bus

The Peripheral Bus (PDBus+) interface is the physical support of the SFR and ESFR address spaces. It connects peripherals that interface via software to a limited set of registers that require a fast and easy access. This is done by combining the various possibilities of the instruction set to access the SFRs and ESFRs and using the speed of the PDBus+.

This bus is used to connect the Infineon standard peripherals of the C166 family (timers, GPT1/2, ASC, SSC, CAPCOMs, etc.).

PD Bus Key Features

- Synchronous protocol, bus clock/CPU clock ratio programmable
- Minimum cycle time of 1 bus clock cycle, waitstates can be generated from the peripheral side by a ready signal
- 2 x 256-word address space mapped in the SFR and ESFR areas
- 16-bit data read and write
- Mask-protected byte write allowing individual bit access
- Split I/O data busses: 16-bit read data bus (to the C166S) and write data bus (from the C166S).

The PDBus+ protocol does not support the following features:

- Multi-master (the C166S core is the only master)
- Time-out monitoring (no time-out mechanism is implemented in the C166S core to prevent deadlock situations with the ready mechanism).

The PD bus receives one 26MHz clock.

5.6.1 I2C

The I^2C bus interface block can be used for communication with any device with an I^2C interface.

The I^2C bus is a master interface used with a single-master I^2C bus in the standard or fast mode. The transmit buffer is 64 bytes, the receive buffer is 1 byte. This permits transmission of up to 62 or 63 bytes depending upon the addressing mode.

The interface generates interrupts to indicate:

- A data transmitted completed
- A data received completed
- A failed data transfer.

The serial clock speed is programmable in the range from 51 kHz to 3.25 MHz.

5.6.2 SSC

The SSC supports duplex and half-duplex synchronous communication at up to 13 MBaud. The serial clock signal can be generated by the SSC itself (master mode) or be received from an external master (slave mode).

Data width, shift direction, clock polarity, and phase are programmable allowing communication with SPI-compatible devices. Transmission and reception of data is double-buffered.

A 16-bit baud rate generator provides the SSC with a separate serial clock signal.

This high-speed synchronous serial interface can be configured in several ways. It can be used with other synchronous serial interfaces, as a master or slave, using multi-master interconnections, or operations compatible with the SPI interface. Therefore, it can be used to communicate with shift registers (I/O expansion), peripherals (EEPROMs etc.), or other controllers (networking).

Data is transmitted or received on the MTSR (Master Transmit/Slave Receive) and MRST (Master Receive/Slave Transmit) pins. The clock signal is output or input on the SCLK pin. These pins are alternate functions of port pins. There are two FIFOs, one for the receiver and one for the transmitter. The size of each FIFO is 32 x 16-bit words.

SSC Key Features

- Master and slave mode operation
- Duplex or half-duplex operation
- Flexible data format with a programmable:
  - Number of data bits: 2 to 16 bits
  - Shift direction: LSB or MSB shift first
  - Clock polarity: idle low or high state for the shift clock
  - Clock/data phase: data shift with leading or trailing edge of SCLK
- Baudrate generation from 99.2 Baud up to 13 MBaud.
- Interrupt generation on a:
  - Transmitter empty condition
  - Receiver full condition
  - Error condition (receive, phase, baudrate, and transmit error).
- Three pin interface for a flexible pin configuration.
- 1 RX FIFO and 1 TX FIFO.

5.6.3 ASC0

The Asynchronous Serial Interface 0 (ASC0) provides serial communication between the host microcontroller and external peripherals.

The ASC0 supports Duplex Asynchronous and Half-Duplex Synchronous communication.

1. Duplex asynchronous communication
   In this operating mode the ASC0 features:
   - 8- or 9-bit data frames, LSB first
   - Parity bit generation and checking
   - One or two stop bits
   - Baudrates from 0.3 Baud to 1.62 MBaud (with a 26 MHz clock)
   - A multiprocessor mode for automatic address or data byte detection
   - Loop-back capability
   - Double buffered transmission and reception
   - Interrupt generation on:
     - A transmitter buffer empty condition
     - A transmit last bit of a frame condition
     - A receiver buffer full condition
     - An error condition (frame, parity, and overrun error)
     - The start and end of a autobaud detection.
   - FIFO functionality:
     - 8-byte receive FIFO (RXFIFO)
     - 8-byte transmit FIFO (TXFIFO)
     - Independent control of RXFIFO and TXFIFO
     - 9-Bit FIFO data width
     - Programmable Receive and Transmit Interrupt Trigger Level
     - Receive and transmit FIFO filling level indication
     - Overrun error generation
     - Timeout.
2. Half-duplex synchronous communication
   In this operating mode the ASC0 features:
   - 8-bit length
   - Baudrates from 26 Baud to 3.25 MBaud (with a 26MHz clock).

5.6.4 CAPCOM1/2

The CAPCOM blocks provides two Capture/Compare (CAPCOM) units that differ in the way they are connected to the IO pins. They provides 15 channels, which interact with 3 timers.

The CAPCOM units can **capture** the contents of a timer on specific internal or external events, or they can **compare** a timer content with given values and modify output signals if they match. With this mechanism they support generation and control of timing sequences on up to seven channels for CAPCOM1 and up to eight channels for CAPCOM2 with a minimum of software intervention. The CAPCOM units features are:

- Timer registers and comparator modules for high speed pulse and waveform generation or time measurement
- 16-bit timers with reload registers (one timer for CAPCOM1 and two timers for CAPCOM2)
- Registers that can be individually configured for the capture or compare function (seven for CAPCOM1 and eight for CAPCOM2)
- Eight interrupts for CAPCOM1 (seven capture compare interrupts and one timer interrupts) and ten for CAPCOM2 (eight capture compare interrupts and two timer interrupts)
- Up to seven software timers for CAPCOM1 and eight for CAPCOM2.
- A programmable clock with multiple sources
- 307.5 ns maximum resolution with a 26 MHz master clock if the staggered mode is enabled
- 38.5 ns maximum resolution with a 26 MHz master clock if the staggered mode is disabled
- A double register compare function
- A primary clock prescaler
- A single event mode.

5.6.5 RTC

The integrated Real Time Clock (RTC) provides programmable alarm functions and external interrupts. The RTC clock of 32.768kHz (32k) is generated by connecting external crystal to the PMB7880 OSC32K input pins. The RTC is, in the power domain, isolated from the rest of the blocks; it can be supplied independently in the standby mode. Due to the extremely low RTC power consumption during the standby mode, the E-GOLDvoice can be supplied from a small backup battery. For example, the 26 MHz reference oscillator can be switched off during system standby and the time reference can be kept via the 32k clock provided to the RTC.

The RTC can generate external interrupts while the main PMB7880 supply voltage is switched off.

5.6.6 GPT1/2

GPT1 and GPT2 contain five identical general purpose timers.

5.6.7 Port Control Logic

This block provides a flexible pad interconnection (alternate functions, ...), It contains the Security block (fuse 1, fuse 2, and the fuse box) and provides General Purpose Input Output (GPIO) functions.

Most digital pads may be configured either:

- For General Purpose IO (GPIO)
- Connected to one of several inputs
- Connected to one of several outputs from chip internal blocks.
- Optional monitoring of internal signals for analysis purposes.

5.6.8 TAP Controller & Boundary Scan and Break Switch

5.6.8.1 TAP Controller & Boundary Scan

The main part of the Test Access Port (TAP) controller is a finite state machine which controls the different operational modes of the boundary scan.

Both the TAP controller and boundary scan meet the requirements specified by the JTAG standard IEEE Std.1149.1.

The TAP controller is also used as an interface for multicore debugging.

Note: The external debugger uses this interface to debug the C166 and TEAKlite with a JTAG interface.

5.6.8.2 Break Switch

The Break Switch triggers the TEAKlite and C166S cores. It is connected to the C166S OCDS block and to the OCEM block (from the TEAKlite bus).

6 RF Subsystem

6.1 Pin Definition and Function

Table 4 Pin Definition and Function

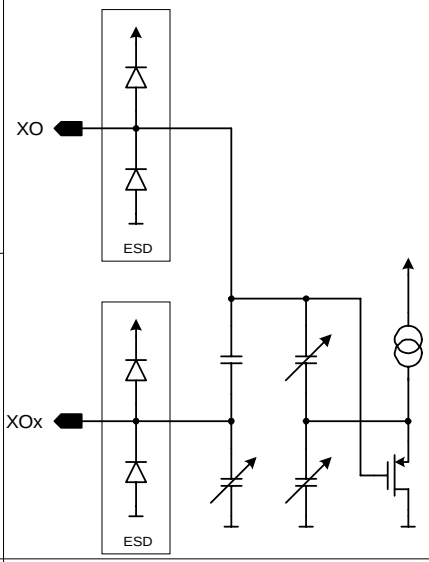
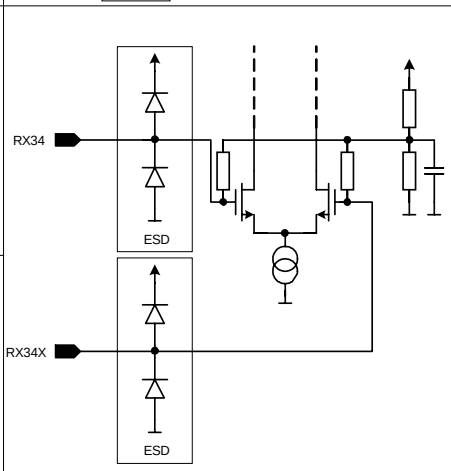
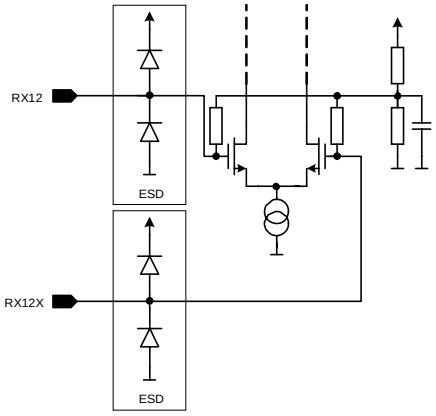
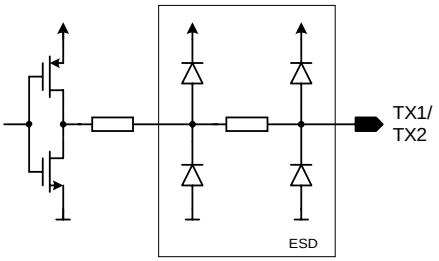
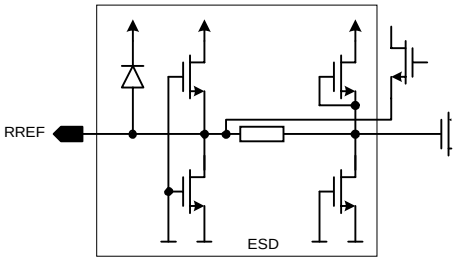
Symbol	Equivalent I/O-Schematic	Function
XO		26 MHz crystal
XOX		26 MHz crystal
RX34		GSM 1800/1900 Rx input
RX34X		GSM 1800/1900 Rx input

Table 4 Pin Definition and Function

Symbol	Equivalent I/O-Schematic	Function
RX12		GSM 850/900 Rx input
RX12X		GSM 850/900 Rx input
TX2		GSM 1800/1900 Tx output
TX1		GSM 850/900 Tx output
REXT		Reference resistor[1)]
VDD_LRFRX2V5		2.5V power supply
VDD_LRFRX2V5		2.5V power supply
VDD_LRFXO		2.5V power supply
VDD_LRFTRX1V5		1.5V power supply
VDD_LRFTRX1V5		1.5V power supply
VDD_LRFTRX1V5		1.5V power supply
VSSRF		RF ground

1) RREF = 220 Ohm. External resistor is not needed.

6.1.1 Receiver

E-GOLDvoice features a fully integrated constant-gain direct conversion receiver, i.e. there is no interstage filter needed and the baseband level at the analogue IQ-interface follows directly the RF input level. Depending on the

baseband ADC dynamic range, single- or multiple-step gain switching schemes are possible. An integrated, self-aligning, low-pass filter ensures the receivers to function under blocking and reference interference conditions and avoids aliasing by baseband sampling. An automatic DC-offset compensation is implemented and can be switched depending on the gain setting.

6.1.1.1 RF Front-End and Demodulator

The E-GOLDvoice RF front-end contains 2 integrated LNAs for the pair of Dual Bands with balanced inputs. The amplified RF signal is direct converted by a quadrature demodulator to the final output signals at the baseband frequency. The orthogonal LO signals are internally generated via a divider by four for the GSM850/900 bands and by two for the GSM1800/1900 bands.

6.1.1.2 Baseband Stage

The resulting in-phase and quadrature signals are fed into the baseband stage, which comprises low pass filtering, programmable gain steps, a programmable gain correction and automatic DC offset compensation circuitry. The fully integrated baseband filter provides sufficient suppression of blocking signals and adjacent channel interferences to match optimally with the baseband ADCs providing a 72dB dynamic range at full scales from 1 V_{pp} up to 4 V_{pp}. The ADCs anti-aliasing requirements are fulfilled for sampling rates from 6.5MHz on up. The low pass filter is self-aligning with a residual 3dB roll off frequency tolerance of ±7%.

6.1.1.3 Gain Correction

Process tolerances mainly in the receiver RF section may cause a deviation from nominal overall gain values (receiver gain + front-end insertion losses). To be centered within the ADC dynamic range over process and temperature a programmable gain correction with a range of +/-6 dB with 1 dB stepping is implemented. This avoids an exceeding noise contribution of the ADC in a minimum gain case or an ADC overdrive in a maximum gain case.

Gain Correction Procedure

Receive level measurement and reporting in GSM is needed for adjacent cell monitoring and part of the transmit power control. The specified level reporting accuracy requires a calibration of the receiver overall gain during the final assembly of the mobile phone. For PMB7880 it is recommended to perform this separately for RXGAIN0 = 1 (high/medium gain) and RXGAIN0 = 0 (low gain). The calibration information is saved in a look-up table and available to correct the receiver overall gain over frequency and temperature burst by burst. All gain steps RXGS0..RXGS3 have a high accuracy (+/-0.2dB) and, therefore, their consideration in the receiver calibration routine is not needed.

6.1.2 Transmitter

The digital transmitter architecture is based on a fractional-N sigma-delta synthesizer for constant envelope GMSK modulation. This configuration allows a very low power design with a reduced external component count.

The modulation is transferred between baseband- and RF-part of the PMB7880 via a digital interface signal into the digital modulator.

The following Gaussian filter shapes the digital data stream for the GMSK modulation. Additionally a pre-distortion filter compensates the attenuation of the PLL transfer function resulting in a very low distortion at the transmit output.

The filtered digital data stream is scaled appropriately and added to the channel word. This sum is fed into the MASH modulator. The output of the MASH modulator is a sequence of integer divider values representing the high resolution fractional input signal. This sequence controls the MMD (multi modulus divider) at a sample rate of 26MHz. Thus a tightly controlled frequency modulation of the VCO is achieved.

The output signal of the VCO is divided by four for GSM 850/900 or by two for GSM 1800/1900 respectively. Finally the divided signal is amplified by a single ended output driver with 50Ohm output impedance to allow for a direct connection to the PA.

The transmitter achieves a very low out-of-band noise, typically-163.5 dBc/Hz @ 20 MHz offset, and a very low rms phase error of 1.2 degree typically.

6.1.3 RF-Synthesizer

The PMB7880 contains a fractional-N sigma-delta synthesizer for the frequency synthesis in the RX and TX operation mode. The 26MHz reference signal is provided by the internal crystal oscillator. This frequency serves as comparison frequency of the phase detector and as clock frequency of all digital circuitry.

The N-counter of the synthesizer is carried out as a multi-modulus divider (MMD). The loop filter is fully integrated and the loop bandwidth is about 100 kHz to allow the transfer of the phase modulation.

The fully integrated quad-band VCO is designed for the four GSM bands (850, 900, 1800, 1900 MHz) and operates at the double or four times transmit or receive frequency. To cover the wide frequency range the VCO is automatically aligned by a binary automatic band selection (BABS) before each synthesizer startup.

6.1.4 Power Up Sequencer

The different transceiver states are defined by a central state machine. The mode transitions are achieved by sending one of the 3-wire-bus telegrams CHANNEL2 or RXTX. As prerequisite the 26 MHz crystal oscillator has to be active.

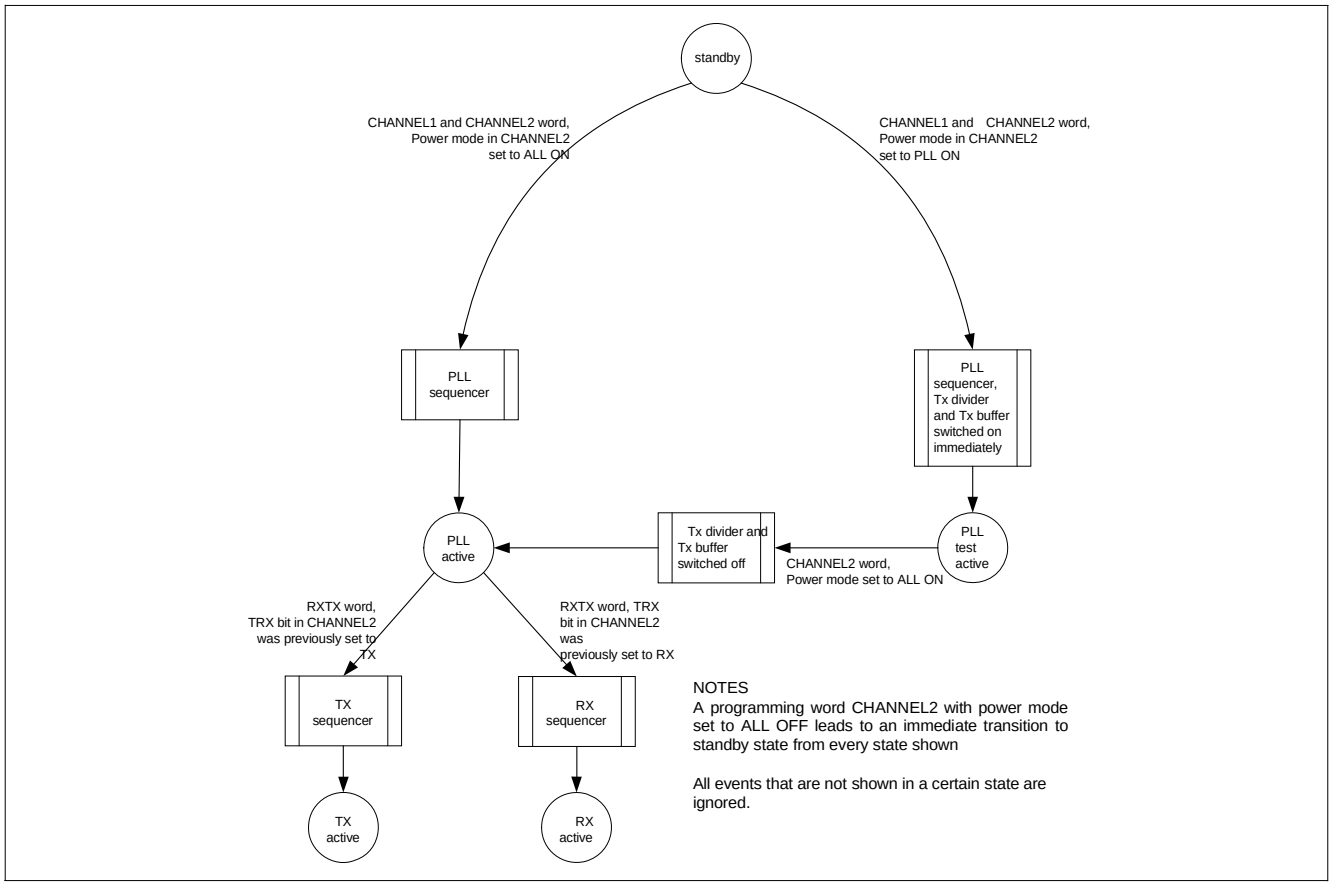


Figure 9 E-GOLDvoice Transceiver State Machine Modes

Most of the mode transitions cause the start of a power up sequencer. There are three sequencers implemented on the chip:

- PLL sequencer
- RX sequencer
- TX sequencer

The PLL sequencer controls the internal alignments and startup of the sigma delta synthesizer. There is no effect on any signal outputs.

The RX and TX sequencer timings are shown in **Figure 10** and **Figure 11**. According to the state diagram the PLL must be started to enable an activation of the RX or TX sequencer.

The TX sequencer also controls the timing of the transmit output buffer. 14.4µs after the end of the RXTX word the transmit output buffer is activated. The so caused rise of the output level should be time aligned with the 1st step of the power time template. This reduces the isolation requirements to the PA.

The “PLL test active” mode allows to monitor the PLL lock-in transition at the transmitter output since the transmit divider and buffer are activated immediately after the CHANNEL2 word. The lock-in behavior can deviate from the normal behavior (using ALL ON and a following RXTX word) since high startup currents and settling of the internal bias in the transmit chain may cause interference to the PLL startup steps.

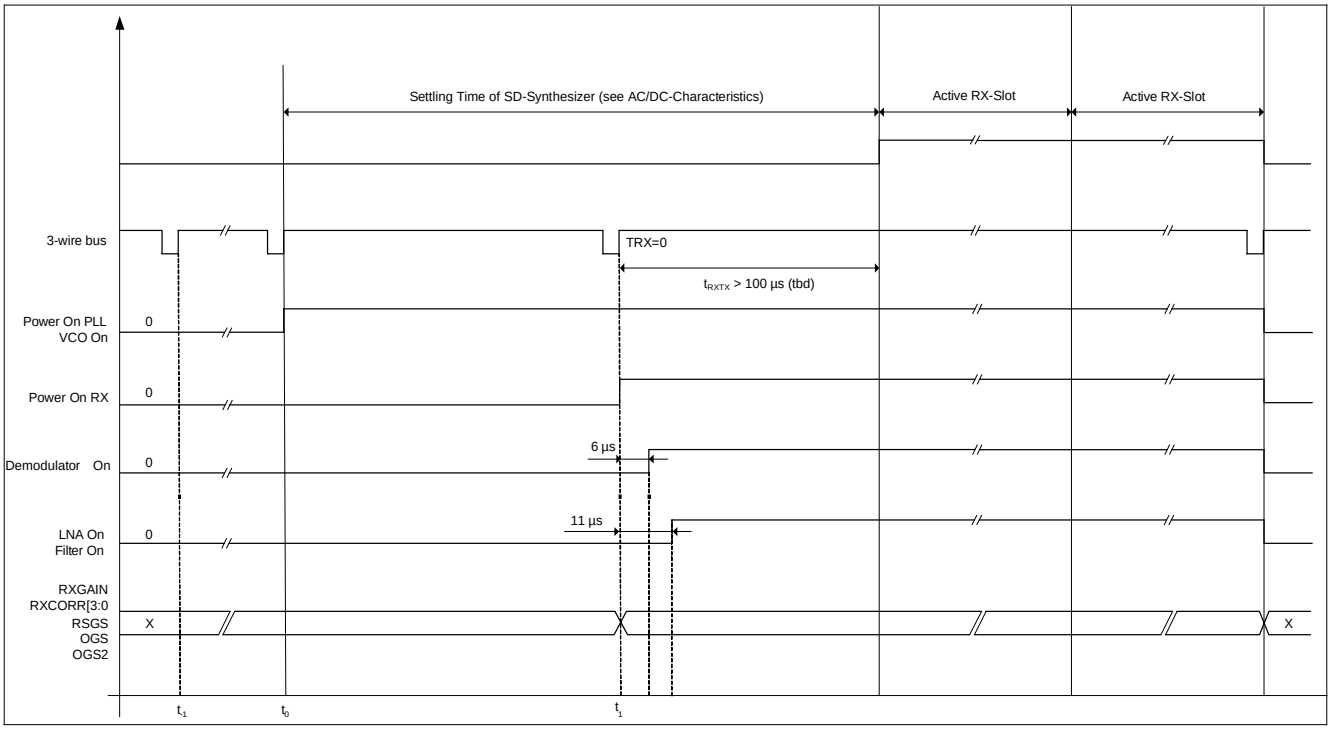


Figure 10 RX Sequencer Timing

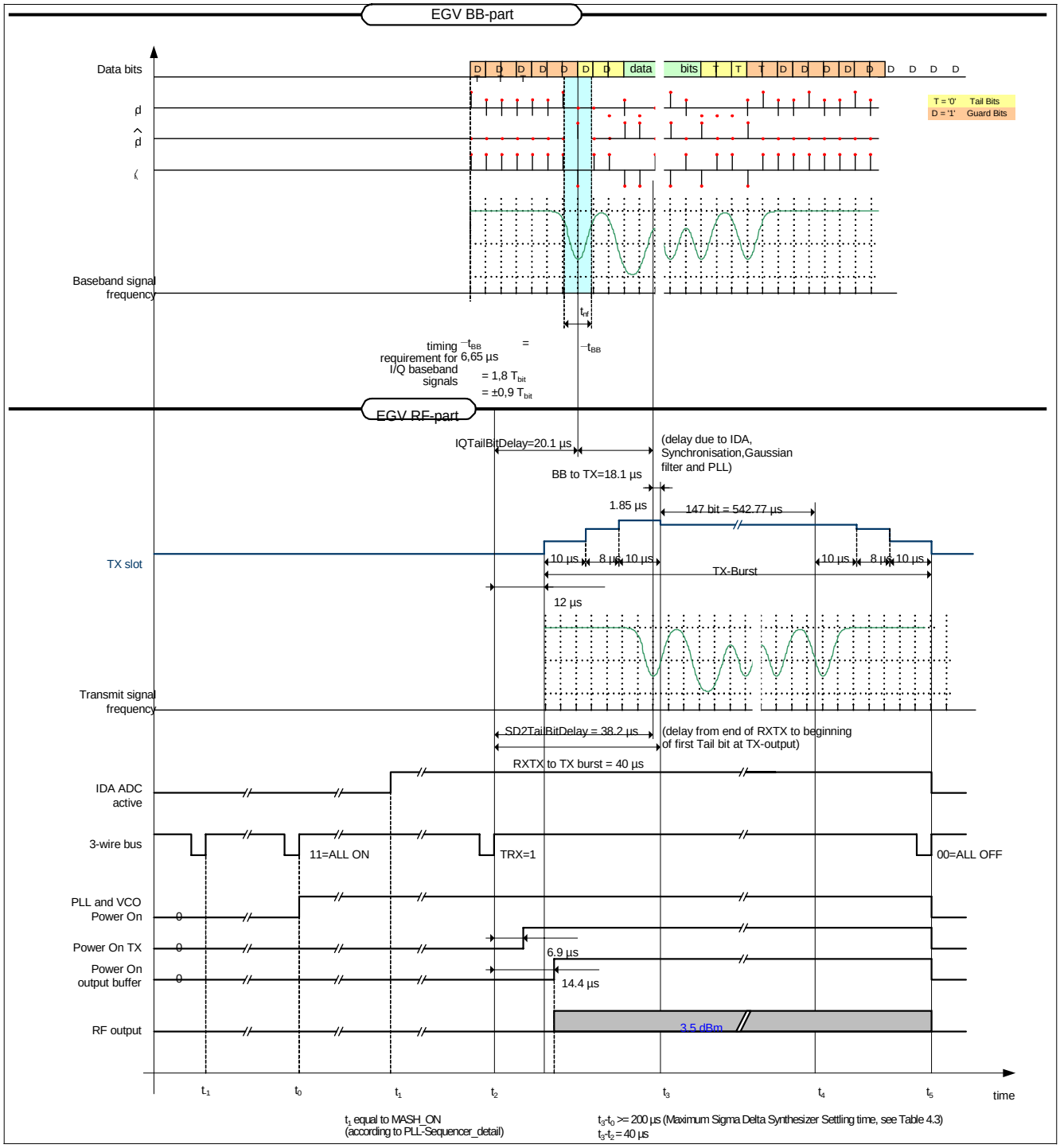


Figure 11 TX Sequencer Timing

6.1.5 3-Wire Serial Bus

The operational functionality as well as the power-on timing of the E-GOLDvoice is controlled by software programming via a internal the 3-wire bus (CLK, DA, EN). This interface is also accessible externally when in a certain test mode.

The required programming data is split into several 24 bit long registers. For operation three 24 bit telegrams before the active slot and 1 telegram for power down after the active slot are needed. Before programming the 3-

wire bus VDDXO (2.5V) and VDDDIG (1.5V) have to be applied. Three initiation telegrams are needed after the activation of the crystal oscillator to adjust frequency and operating mode.

6.1.6 DCXO

The PMB7880 contains a fully integrated 26MHz digitally controlled crystal oscillator (DCXO).The overall pulling range of the DCXO consists of 8 subranges. The sub-range closest to the 0ppm at the middle AFC-value is selected during the calibration process in production. The spacing between the sub-ranges is constant. Therefore, for a proper sub-range selection only 2 measurements are sufficient. Afterwards, the selected sub-range number will be stored in the software and used for the mobile´s lifetime. An additional measurement in calibration can be used to determine the AFC value, which corresponds to the 0 ppm along the selected sub-range. The pulling along the selected sub-range is used for correction of all kinds of frequency drift. **Figure 12 DCXO Sub-range Selection** shows the sub-range selection.

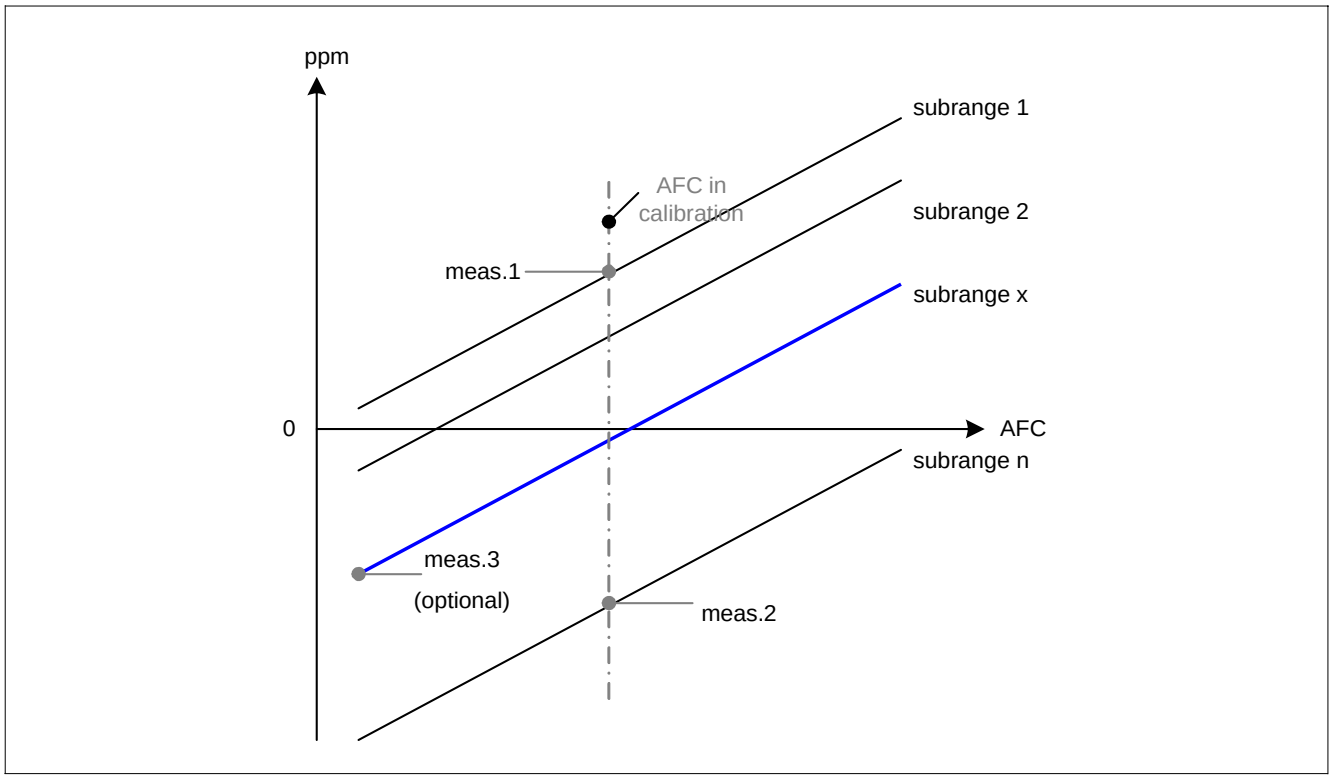


Figure 12 DCXO Sub-range Selection

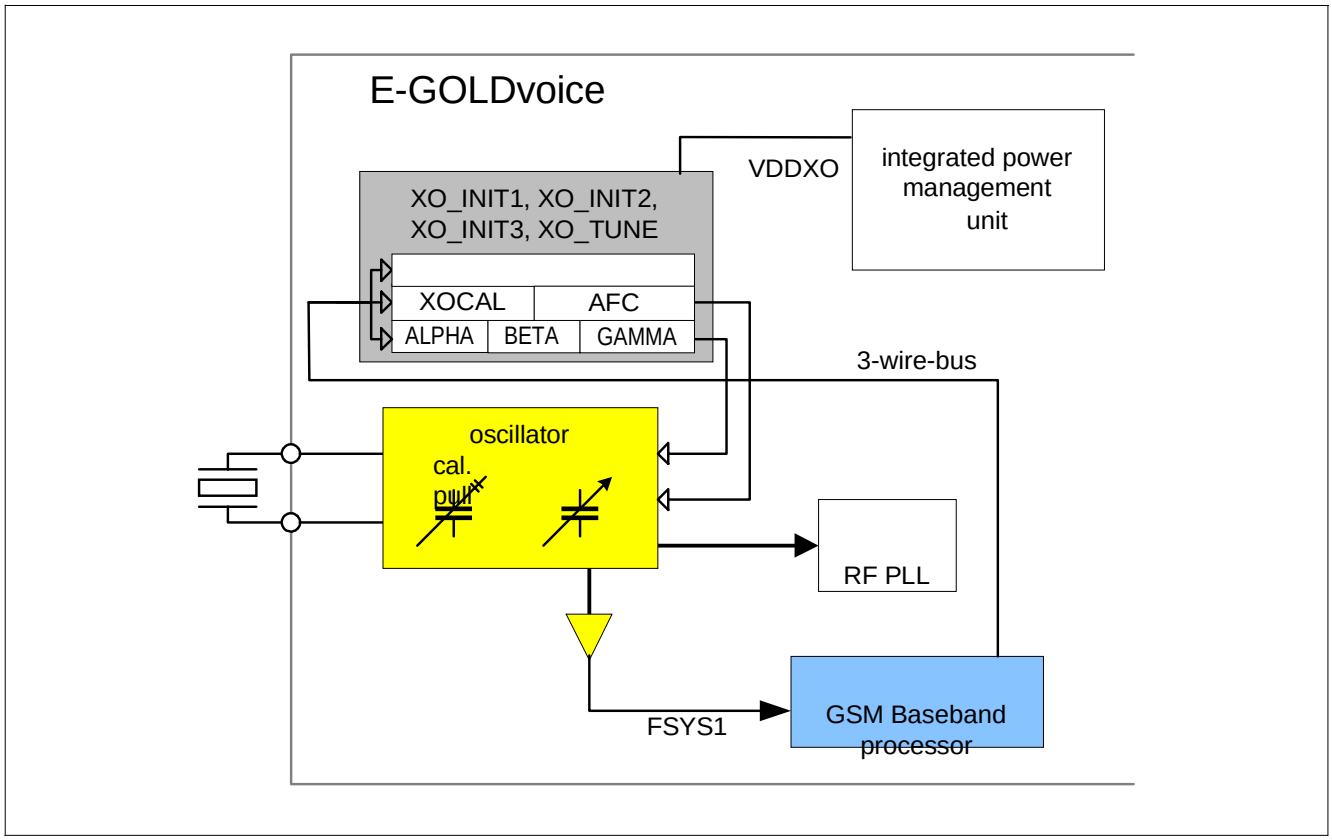


Figure 13 Crystal Oscillator Functional Overview

The XO_TUNE register holds the digital correction value for the crystal oscillator frequency. The XOMODE bits of XO_INIT1 register contain setup informations for the crystal oscillator (for example, current programming, etc.). See **Figure 13 Crystal Oscillator Functional Overview**.

The registers XO_INIT2 and XO_INIT3 contain the coefficients information for the linearization unit of crystal oscillator (LUXO) This linearization unit computes the required digital control word out of the programmed AFC bits in order to have a linear pulling curve ppm vs. AFC word. The resulting digital control word DIG is filtered by a digital lowpass filter, which can be scaled or deactivated using the bits DIGFILT0 and DIGFILT1 of the XO_INIT3 register.

The frequency correction splits into 2 parts:

1. The XOCAL bits in the XO_INIT1 register are used for the coarse frequency adjustment and are set once for a mobile lifetime (during production test)
2. The XO_TUNE register contains the information for frequency correction when the mobile is used (correction of temperature drift, crystal aging).

7 X-Bus

7.1 X-Bus Description

The X-Bus access to the X-peripherals is determined by the **XBCONx** and **XADRSx** registers. The C166S core supports only synchronous-ready operations on the X-Bus.

The description of the XBCON registers is in **Section 7.5.2.1.2 X-Bus Control Registers (on Page 207)**. The description of the XADRS registers is in **Section 7.5.2.1.1 X-Bus Address Selection Registers (on Page 206)**.

The Address Mapping of the X-Bus Peripherals is given in **Section 10.2 X-Bus Register Addresses**.

7.1.1 X-Bus Interface

All X-Bus peripherals are split into a peripheral kernel and a bus peripheral interface (BPI), see **Figure 14**. The BPI interface performs the adaptation to the X-Bus.

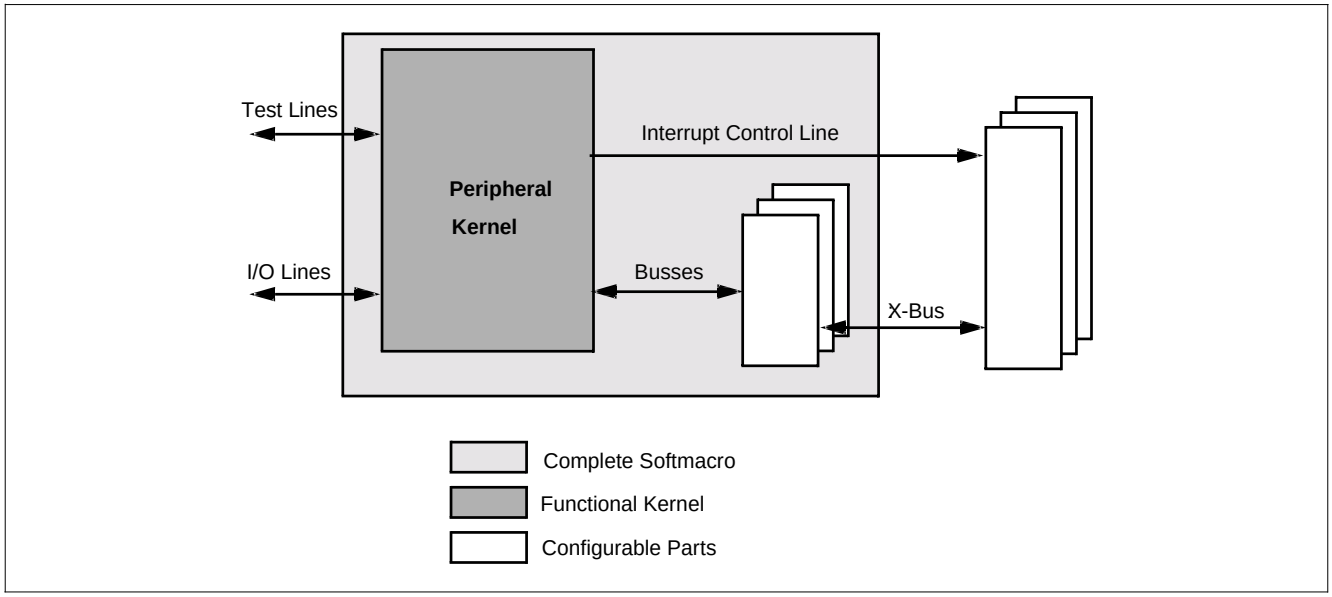


Figure 14 General Block Diagram of X-Bus Peripheral

As the registers of the peripherals have different addresses in different MCUs, also the addresses are decoded in the BPI block.

X-Bus Peripherals which include only a very low number of registers (<256byte) should be mapped within one address window of 256 bytes and use only one **XBCONx**/**XADRSx** register pair. Therefore all these peripherals must use the same bus cycles (MUX/DEMUX, wait states, etc.) and allow a flexible mapping of their addresses into the 256byte address space of one **XBCONx**/**XADRSx** register pair. These should be ensured by the X-Bus-BPI. Which 256 byte address segment of the whole address space is used is determined by the **XADRSx** register and so by the XCS signal of the peripheral.

7.1.2 8- and 16-bit Access to X-Bus Peripherals

All registers in X-Bus peripherals should support byte accesses to the upper and lower byte of 16-bit words. RAM accesses (GSM Timer Unit, Shared memory) can only be performed 16 bit wise.

7.1.3 Reset Behavior of X-Bus

Refer to **Section 7.7.4 Reset Control (on Page 228)** and **Figure 52**.

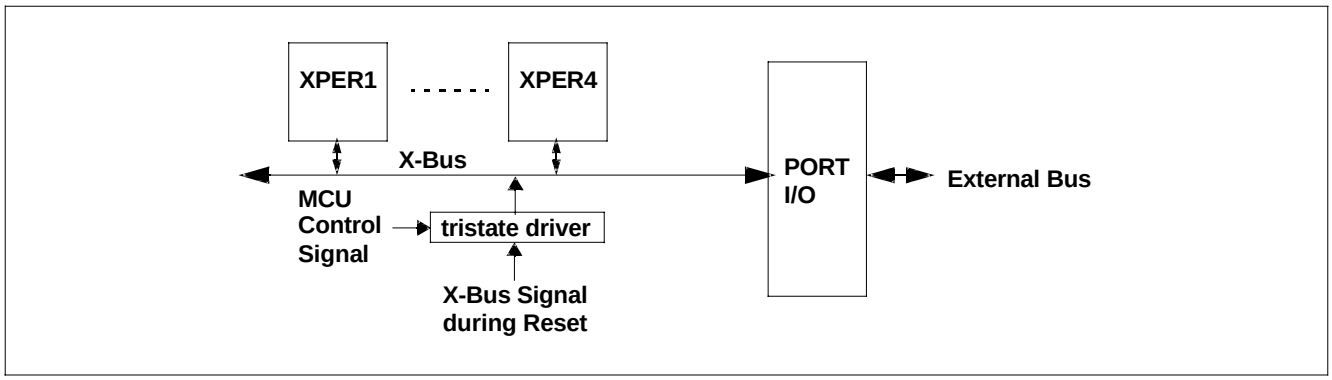


Figure 15 MCU Configuration during Reset via X-Bus

7.2 PLL - CGU - SCCU

System Integration

- Supply domain:
  - VDD_PLL for the PLL block
  - VDD_LD1 for the CGU and SCCU blocks
- Chip internal interfaces:
  - Clock domain: Refer to **Section 7.2.1.1 E-GOLDvoice Clock Concept Description (on Page 65)**
- Interrupt sources:
- Monitor Pins: Refer to **Section 9.7.10 Internal Signal Monitoring**

7.2.1 Clock Generation Unit

The E-GOLDvoice integrates a power full clocking scheme which facilitates flexibility during normal operation mode combined with minimized power dissipation during standby and sleep mode. Two separate clock inputs are provided, a 26 MHz low swing input and a 32.768 kHz square wave input. The 32.768 kHz clock can also be generated by an on chip oscillator.

The low swing input F26M (XO) passes a clock shaper first. This shaped clock or the 32.768 kHz may be used directly for clocking E-GOLDvoice in power saving mode. During normal operation a PLL with 1 phase shifter generates different clocks from the shaped 26 MHz clock. The principle relations between the frequencies of the input clock, the PLL clock and the phase shifter clock are:

$$F_{pll} = finputx \times \frac{m}{n} \quad (1)$$

$$F_{phaseshifter} = \frac{F_{pll} \cdot 12}{k} \quad (2)$$

Where m,n,K are programmable integer. For the E-GOLDvoice application, F_{pll} must always be set to 208 MHz (that is, clk_pll = 104 MHz).

The PLL (phased locked loop) macro includes 1 phase shifter. The PLL or phase shifter clocks as well as the input clock and the 32.768 kHz clock are distributed via synchronous multiplexers to the master clock independent for each master clock.

From the master clocks additional clocks for different peripherals and buses are derived. In addition, the peripherals allow frequency reduction and clock disabling.

Most, but not all, of the logic shown is implemented in the clock generation unit CGU. Exeption are the 32 kHz oscillator, which is implemented together with the RTC in the RTC voltage domain, Also the DSP peripherals clocks are generated locally in the DSP sub-system.

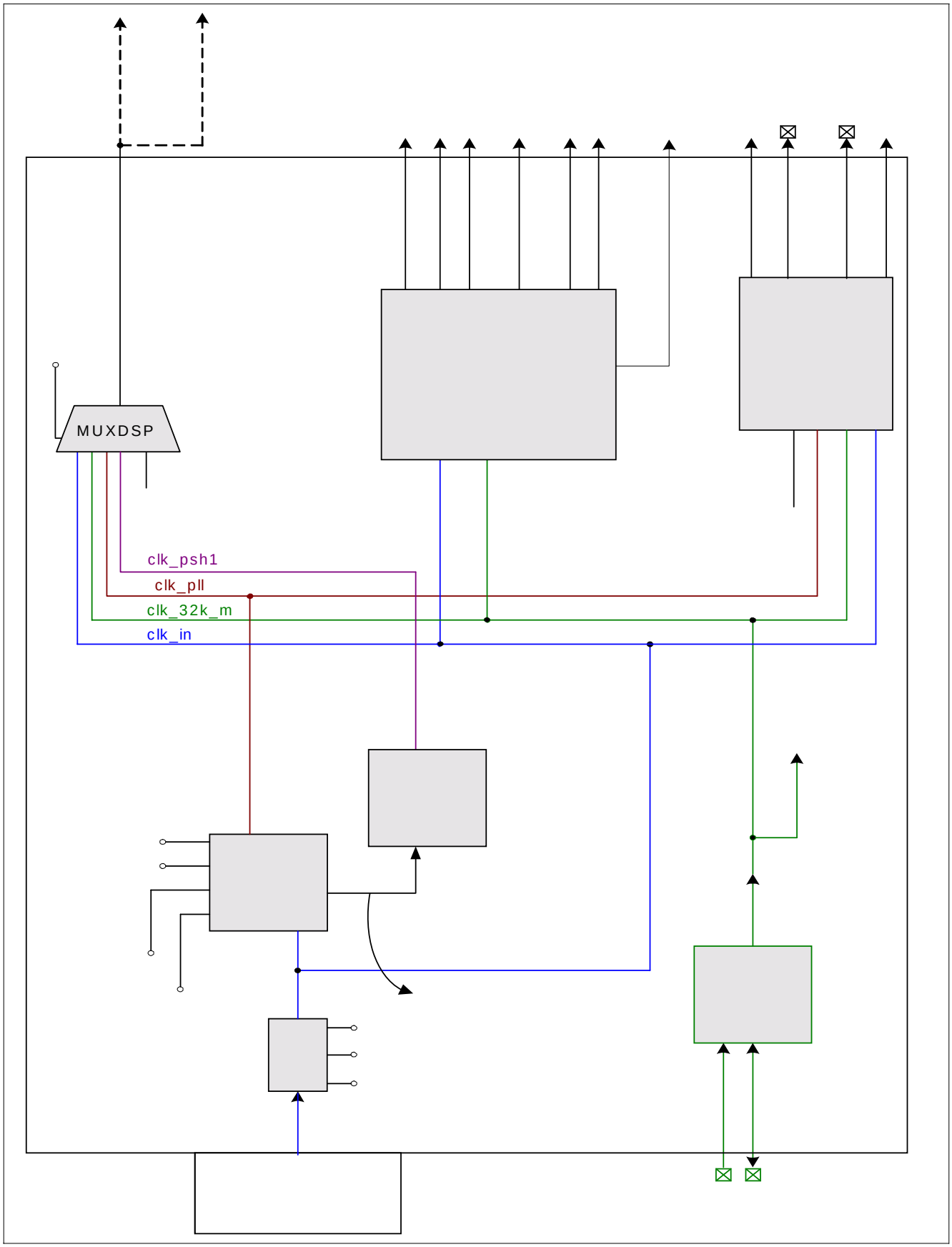


Figure 16 Clock Generation Unit Scheme

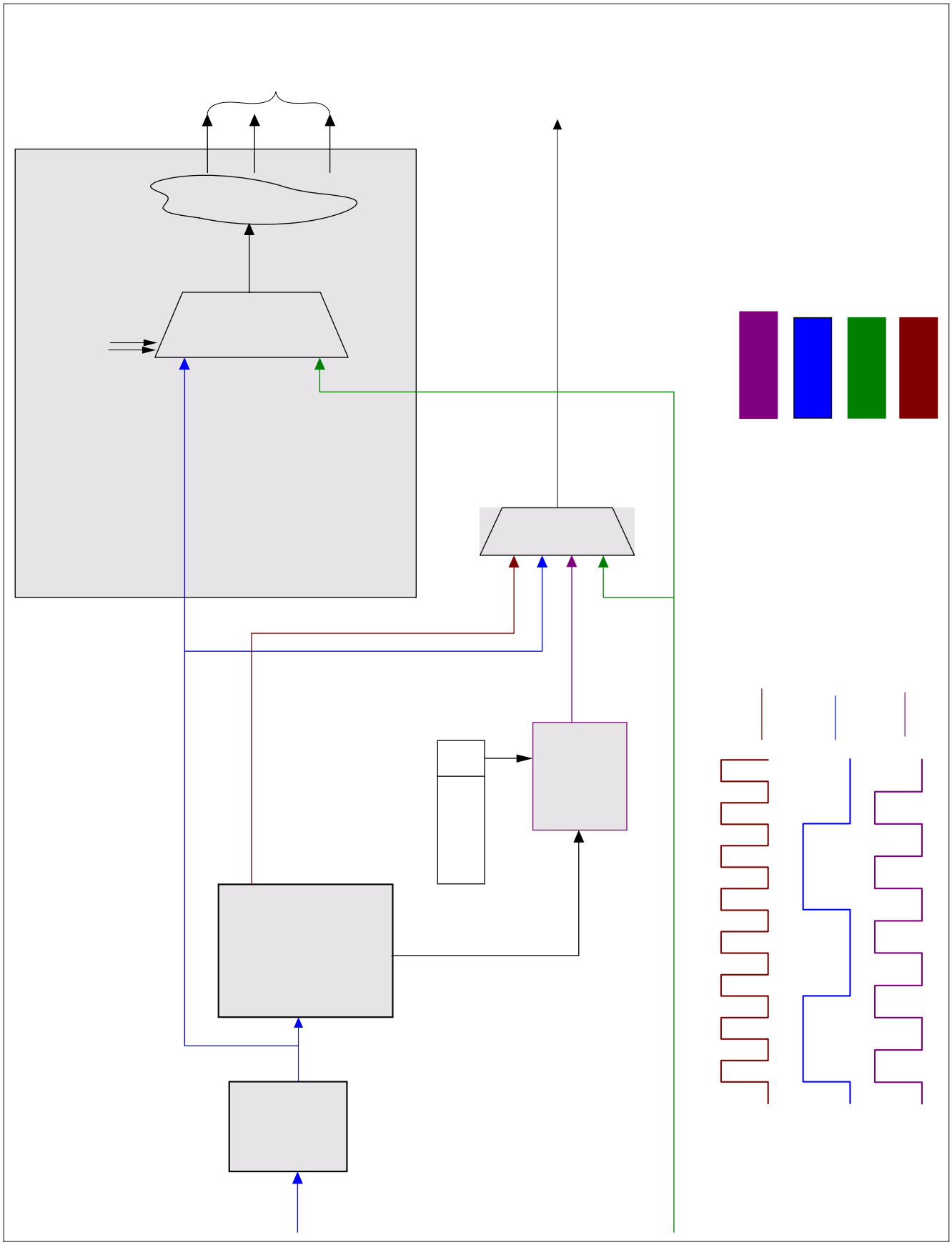


Figure 17 Overview of E-GOLDvoice CPU Clock Generation System

7.2.1.1 E-GOLDvoice Clock Concept Description

The E-GOLDvoice divides into several clock subsystems which can operate asynchronously to each other. Each clock sub-system is driven by a master clock. DSP clock sub-system may operate at a lower frequencies. In this case, the lower frequency clock is derived from the respective master clock by an integer divider or a fractional divider and is synchronous with the master clock.

The clock sub-systems are:

- PLL frequency clock system
  is driven by the PLL output clk_pll. The clk_pll runs at 104 MHz.
- The real time clock RTC
  the RTC clock clk_32K runs at 32 KHz.
- The controller system includes the C166s with its peripherals. The master clk_cpu_master runs at 26MHz in the normal mode (it is also 26 MHz in bypass mode but 32 KHz in oscillator mode).
  To save power we use the 32 KHz clock from the RTC in oscillator mode.
- The DSP sub-system
  includes the DSP core and its peripherals. The master clk_dsp is coming from a Phase shifter. The target frequency is 78 MHz. Lower frequency are configurable. However, due to GSM system requirements, system functions are not guaranteed at lower frequencies.
- The Analog macro
  receives its clocks from some peripherals, such as the DSP sub-system, the GSM interface, or the measurement unit.
- The interfaces between the subsystems support asynchronous clocks. These interfaces are:
  – Controller sub-system – DSP sub-system
    One asynchronous shared memory interfaces between the X-Bus and the data buses of the DSP
    Interrupt lines between the controller system and the DSP sub-system
  – Controller system – RTC, Keypad
    Asynchronous interface within the peripherals
  – DSP System – Analog part of GSM
    Asynchronous interfaces between the mixed signals blocks and the DSP buses

The following parts of E-GOLDvoice normally run at different frequencies:

- MCU sub system
- DSP sub system
- RTC
- TCU
- Analog macro.

Table 5 shows the target operating frequencies for the different parts of E-GOLDvoice.

Table 5 Target Operating Frequencies for E-GOLDvoice

BLOCK	TARGET FREQUENCY
MCU Sub-System	26 MHz
DSP	78 MHz
Shared Ram (on dsp side)	78 MHz
RTC	26 MHz
TCU	10 MHz
ANALOG	
104 MHz **PLL MACRO**	104 MHz

7.2.1.2 Operation Description

The internal RF oscillator provides the major 26MHz reference clock for E-GOLDvoice via the XO pad.

An external DCXO may be connected to the E-GOLDvoice via pins XO and XOX.

The input clock first passes though a shaper, which keeps a low swing input level on XO. The shaper output is clk_in. The shaper can be powered up with a delay related to the DCXO power up. This might slightly reduce the overall power consumption.

The following subsection describes different scenarios for starting or re-configuration of the clock system.

7.2.1.2.1 Startup on External Reset

Upon reset of the MCU, all internal clocks are supplied with the input clock clk_in, then:

1. The PLL is powered up automatically and starts generating clk_pll with 4 times the frequency of clk_in. That is, clk_pll runs at 104 MHz because clk_in is running at 26 MHz.
2. The phase shifter is switched off.
3. The SW must configure the PLL (if needed) and the phase shifter as desired for the application.
4. The clk_pll must always be configured for 104 MHz.
5. After setting the PLL parameters, the primary phase shifter will reach their target frequencies within a maximum of 100 ∝s.
6. When PLL has reached its target frequency, it activates an internal ‘lock’, which is latched afterwards as a register bit.

Thus, the SW has two options to find a suitable point of time for switching from clk_in to the PLL and phase shifter clocks. Perform either:

1. Use a 100 ∝s timer
2. Poll the bit **PLL_CTRL.Pll_locked**.

When the PLL is locked, the clk_pll and phase shifter clocks may be used as master clocks for the other clock subsystems of E-GOLDvoice by:

1. Deactivating the external INIT reset.
2. Setting M and N PLL factors via **PLL_CTRL.Pll_M** and **PLL_CTRL.Pll_N**. PLL_powerup is already activated.
3. If the phase shifter is used, set phase shifter factors X1 and Y1 via **PHX_CTRL.Phs_X** and **PHX_CTRL.Phs_Y**.
4. Setting the phase shifter power up, if needed, via **PLL_CTRL.Pll_powerup**.
5. Waiting for the PLL to lock-on.
6. When locked-on, the SW can set **PLL_CTRL.Pll_bypass_n** (un-bypass the PLL).
7. If the phase shifter is used, the SW can set **PHX_CTRL.Phs1_bypass_n** (un-bypass the Phase Shifter).
8. Setting the select and divider field as desired.

The clock multiplexers deliver the new clock 4 cycles in the old clock domain plus 4 cycles in the new clock domain after changing the select field.

7.2.1.2.2 Changing the Phase Shifter Frequency

Before changing the phase shifter frequency, it is mandatory to disconnect all clocking systems connected to that source via **MST_CLK_CTRL.DSP_SEL** = 100_B. The frequency of the phase shifter can only be changed only while the clock is not used by the systems.

To change the phase shifter frequency:

1. Set **PHX_CTRL.Phs1_powerup** and **PHX_CTRL.Phs1_bypass_n** to 0
2. Set **PHX_CTRL.Phs_X** and **PHX_CTRL.Phs_Y** to the new values
3. Set **PHX_CTRL.Phs1_powerup** and **PHX_CTRL.Phs1_bypass_n** to 1
4. Wait for 1 ∝s before using the clock in the system.

7.2.1.2.3 Changing the MCU Sub-System Frequencies During Operation

The controller clock source can always be changed during operation.

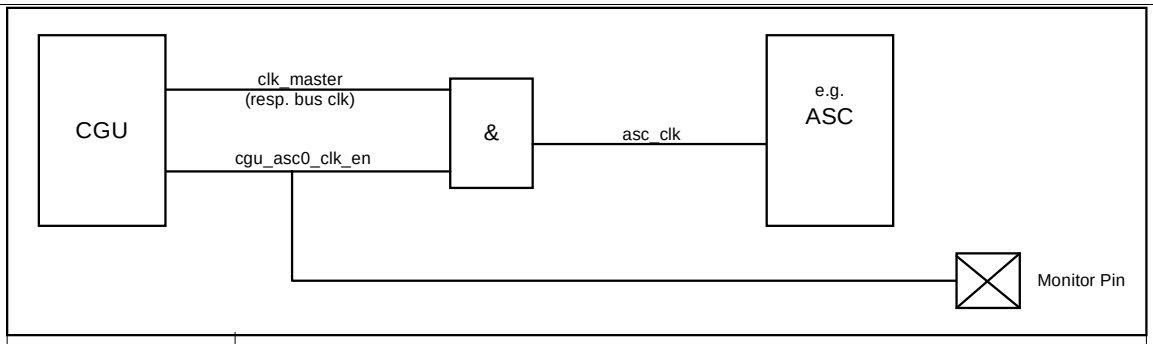
Note: The duty cycle of the CCLK (SIM card clock) signal is impacted when the following bits are updated:

- **MST_CLK_CTRL**.CPUH
- **MST_CLK_CTRL.CPUPRE**.

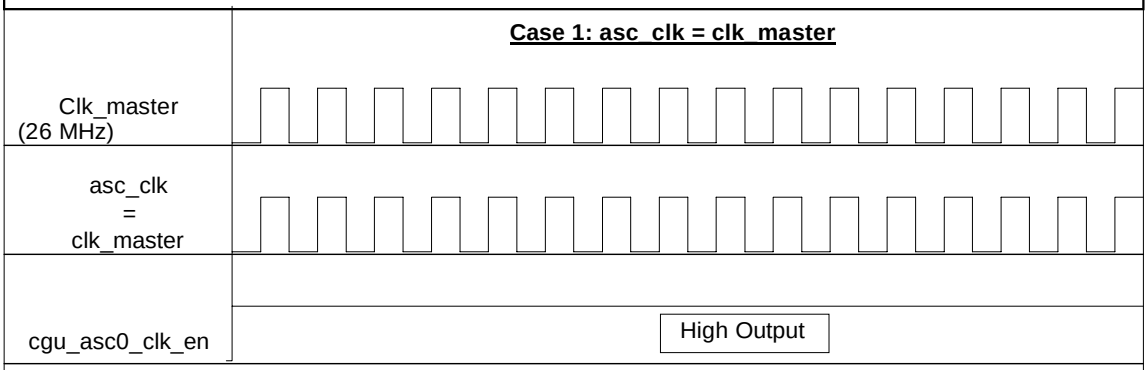
7.2.1.3 Sub-System Clocks and Enables

A sub-system includes a bus and its peripherals. The bus interface is always clocked with its respective bus clock. For the peripherals, it depends on which sub-system they are integrated. For the DSP sub-system the clock generation is made in the DSP part (refer to **Chapter 6 TEAKlite DSP (on Page 108)**). For the MCU sub-system the clocks and enables are generated internally of the clock generation unit CGU with the corresponding enable via the clock_cpu_master.

Figure 18 shows an example how the clocks enable works.

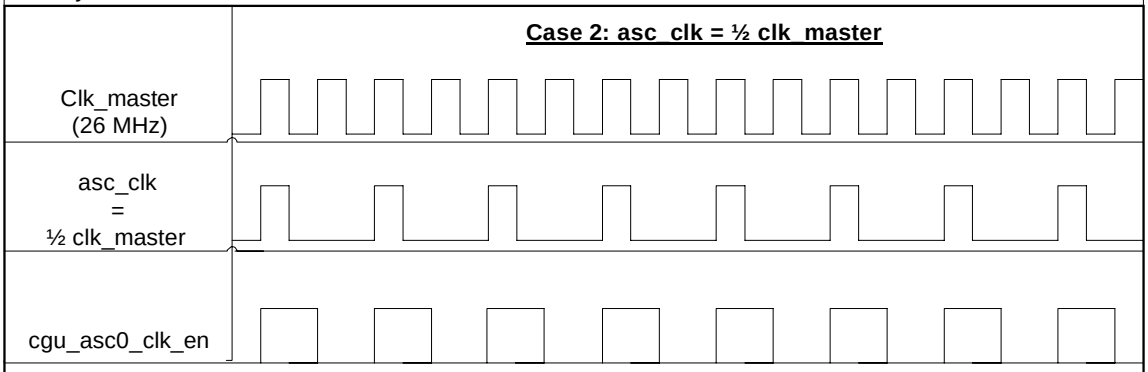


Case 1: asc_clk = clk_master



In this case, when asc_clk should be configured as equal as to the bus clock, the enable signal is set to permanently HIGH (visibable at Monitor Pin). This means, that the AND-Gate will provide the clk_master direktly to ASC.

Case 2: asc_clk = ½ clk_master



In this case, when asc_clk should be configured as half as to the master clock, the enable signal is set to HIGH (HIGH OUTPUT visible at Monitor Pin) for every **second** clk_master cycle. This means, that the AND-Gate will provide every second clock cycle of the clk_master to the ASC.
This will generate **(output at the MON Pin)**
in **Normal Mode** an enable signal **looks like a** 26MHz clock, but it is not the asc_clk.

For the case asc_clk = ¼ clk_master every 4th clk_master cycle will be provided to asc_clk.
For the case asc_clk = 1/8 clk_master every 8th clk_master cycle will be provided to asc_clk.

Figure 18 Clock Enable

7.2.1.3.1 MCU System Clocks

Table 6 shows configurations available for the MCU system .

Table 6 MCU Clock Master

Mode	MST_CLK_CTRL.CPUH & sccu_clk_cpu_en_del	Clk_master
Power saving	X0	32 kHz
Normal Mode and PLL bypass[1)]	01 [2)]	26 MHz

1) The clock source is clk_in taken after Shaper.

2) Reset Value .

Note:

- The Bypass mode 26 MHz clock source is taken after the shaper before the PLL Macro (in case PLL is switched off: PLL bypass).
- The oscillator mode 32 KHz clock source comes from the Pad oscillator.

Table 7 MCU Sub-System Clock

Mode	MST_CLK_CTRL. CPUH	Clk_master	MST_CLK_CTRL. CPUPRE	Cpu_clk_pos Coefficient[1)]	Cpu_clk_neg Coefficient[1)]
Reset	X	x	X	0	0
Power saving	X		32 kHz	X	1 0
Normal and Bypass 26 MHz[2)]	0	(from shaper)	26 MHz	000	1 1
			001	1/2	1/2
			010	1/4	1/4
			011	1/8	1/8
			100	1/16	1/16
			101	1/32	1/32
			110	1/64	1/64
			111	1/128	1/128

1) The real clock is calculated as follows: Clk_master x Clock Coefficient.

2) This mode has exactly the same behavior as Normal 26 MHz. The clock source is the same: clk_in taken after Shaper.

The fraction value indicates the division frequency applied on the clock master.
Normal mode “cpu_pre = 001” => cpu_clk_pos frequency

Table 8 MCU Sub-System Bus Clocks

Mode	MST_CLK_CTRL.C puh	Clk_master	SYSCON1. PDCLKDIV	PD-Bus clock Coefficient[1) 2)]	Xper clock Coefficient[1)]
Reset	x	x	X	0	0
Power saving	x	32 kHz	X	1	1
Normal and PLL bypass 26 MHz[3)]	0	(from Shaper)	26 MHz	000	1 1/2
			001	1/2	1/2
			010	1/4	1/2
			011	1/8	1/2
			100	1/16	1/2

1) The real clock is calculated as follows: Clk_master x Clock Coefficient.

2) This common pd_clk_en is not used in E-GOLDvoice: it is stuck at 1. So any Pd_bus peripheral has either his own clock division (SIFCLKS) as explained below or it has no clock division.

3) This mode has exactly the same behavior as Normal 26 MHz. The clock source is the same: clk_in taken after Shaper.

Table 9 MCU Sub-System Peripheral Clocks (ASC0, SSC, PCL, IIC)

Mode	MST_CLK_CTRL. CPUH	Clk_master	SIFCLKS.(perif)CL	Peripheral clock Coefficient[1]
Reset	x	x	X	0
Power saving	x	32 kHz	X	1
Normal and PLL bypass 26 MHz[2]	0	26 MHz	11	1
			10	1
			01	1/2
			00	1/4

1) The real clock is calculated as follows: Clk_master x Clock Coefficient.

2) This mode has exactly the same behavior as Normal 26 MHz. The clock source is the same: clk_in taken after Shaper.

Table 10 MCU Sub-System Peripheral Clocks (CAPCOM, GPT)

Mode	MST_CLK_CTRL.C PUH	Clk_master	SIFCLKS.(perif)CL	Peripheral clock Coefficient[1]
Reset	x	x	X	0
Power saving	x	32 kHz	x	1
Normal and PLL bypass[2]	0	26 MHz	11	1
			10	1/2
			01	1/4
			00	1/8

1) The real clock is calculated as follows: Clk_master x Clock Coefficient.

2) This mode has exactly the same behavior as Normal 26 MHz. The clock source is the same: clk_in taken after Shaper.

7.2.1.3.2 MCU System Clock

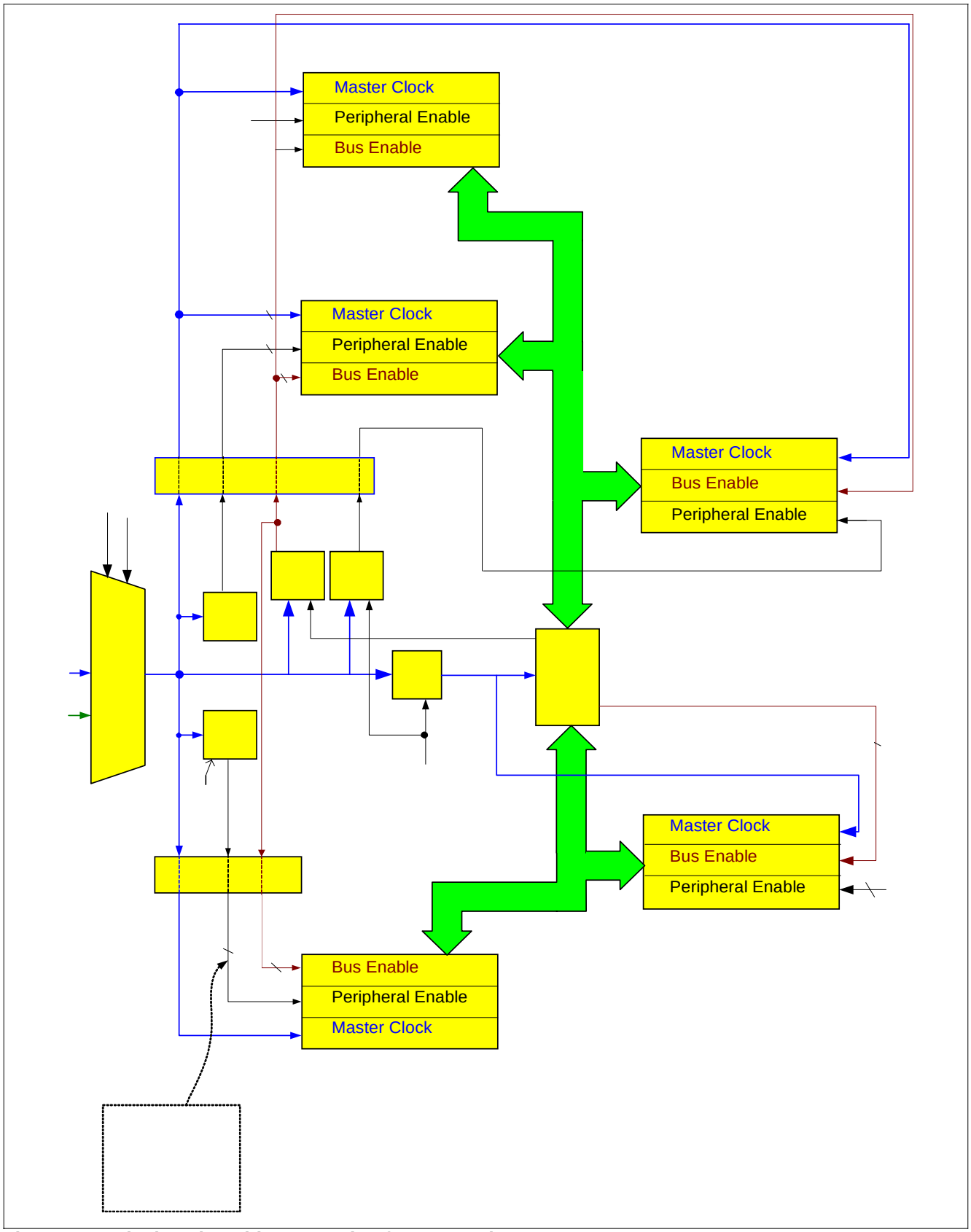


Figure 19 Clock and Enable Generation for MCU Sub-System

7.2.1.3.3 DSP Clock System

The DSP and its peripherals are clocked with the DSP master clock clk_dsp. The related mixed signals blocks are clocked with the clk_pll. All clock enabling and dividing is done locally in the peripherals. Details are given in the respective peripheral sections (see **Figure 20 Generation of Clocks for the DSP Section (on Page 73)**).

The controller may get information on the peripherals clock status via the shared memory and a dedicated command structure. The CGU provides different clocks source to the DSP via the clk_dsp signal. According the value of **MST_CLK_CTRL.DSP_SEL**, the CLOCK_DSP signal receives one of the clock sources shown in **Table 11**.

Table 11 DSP Clock Sources

DSP_SEL	CLK_DSP	Frequency
000	clk_in	26 MHz
001	clk_psh1	Programmable from 48 to 156 MHz (default: 78 MHz)
010	clk_pll	104 MHz
011	clk_32K	32 kHz
100	0	0
Other	clk_dsp clock fixed to 0 Hz	0

The test clock is provided when the device is in the test mode.

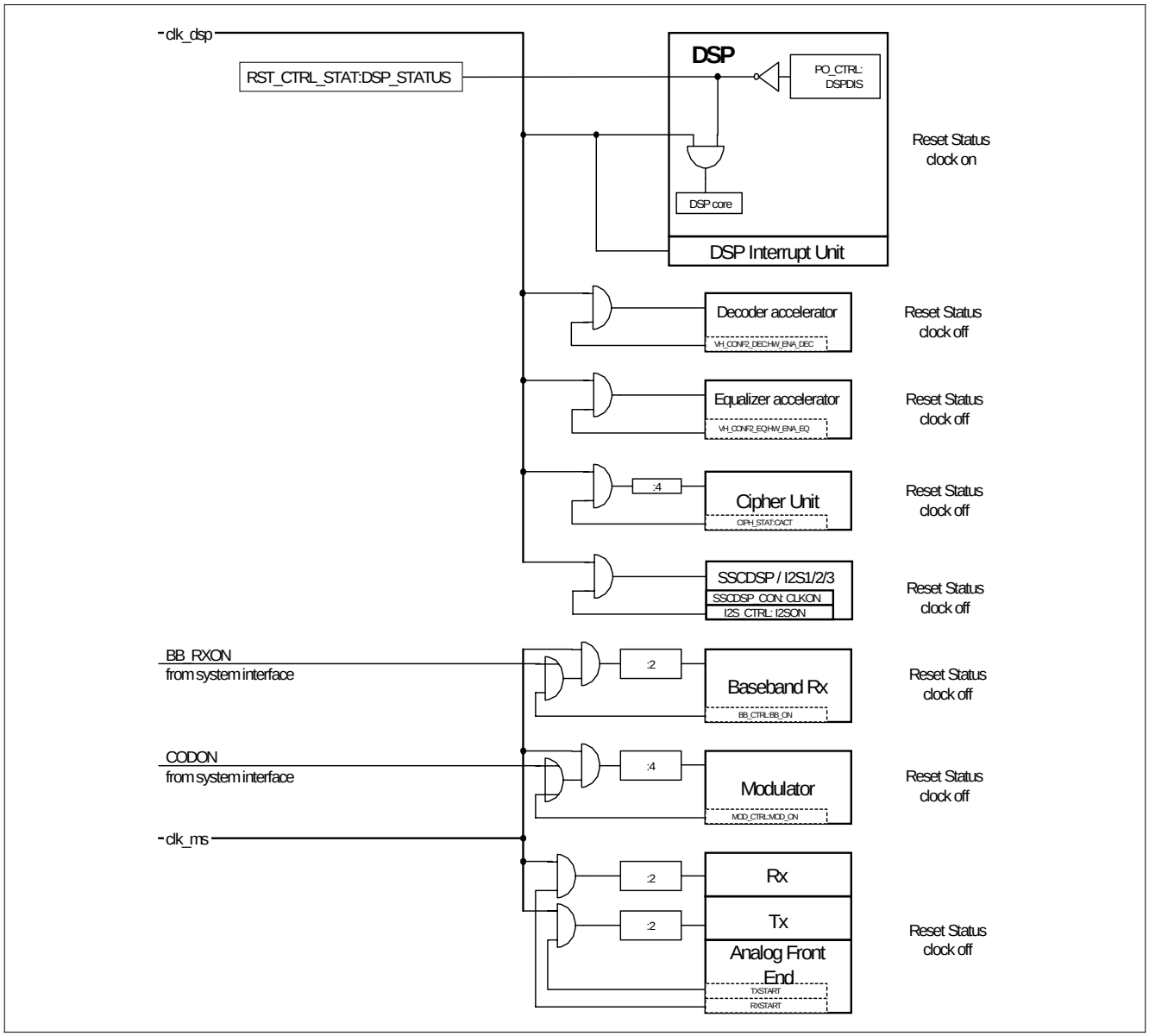


Figure 20 Generation of Clocks for the DSP Section

7.2.1.3.4 Analog Clock

The master analog clock is derived from the clk_pll_o which runs at 104 MHz. It is called clk_ms, for mixed signal. The Analog master clock can be turned on or off by enabling/disabling the clk_ms with **MST_CLK_CTRL.MSEN** bits. Clk_ms can also receive the clock after the shaper called CLK_IN.

Clk_ms does not go directly to the Analog macro, it goes through several peripherals to generate several sub-clocks.

Table 12 Analog Clock Generation

MST_CLK_CTRL. MSEN	**SCCU signal *sccu_clk_cpu_en_del*** (**see Page 90**)	**Clk_ms switched to:**
	0	Clock not enabled
00	1	Clk_in (26 MHz)
01	1	Clock not enabled
10	1	Clk_pll (104 MHz or 26 MHz, according the value of **PLL_CTRL.PLL_bypass_n**)
11	1	Clock not enabled

There are several analog clocks. They come from different peripherals as given in **Table 13**.

Table 13 The Analog Clocks

Analog Clock	**Frequency**
clk_vbrx	4 MHz (has to have a low jitter)
clk_vbtx	2 MHz
clk_vbtx_dith	2 MHz/10
clk_kernel	8 MHz
clk_meas	1 MHz
clk_par	6,5 MHz
clk_pa2	6,5 MHz
clk_pa1	6,5 MHz
clk_bbtx	13 MHz
clk_bbrx	now 26 MHz
clk_bbrx_dith	13 MHz/12 (1.08 MHz: 50% duty cycle)

7.2.1.3.5 Output Clock

The E-GOLDvoice provides three different clocks for external devices (see **Table 14**). These three independent outputs allow external access to:

- clk_in -> CLKOUT_26
- clk_32k -> CLKOUT_32K
- clk_out -> CLKOUT.

Table 14 Output Clock

MST_CLK_CTRL.CLK_OUTL_EN (Output Clock Enables)	Output Clocks
Bit 8 CLKOUT_26	0 Disables 1 Enables
Bit 9 CLKOUT_32K	0 Disables 1 Enables
Bits 11:10	00 No clocks applied on output CLK_OUT 01 Enables clk_pdbus 10 Enables clk_xbus 11 Enables clk_cpu_master

7.2.1.4 Standby Mode Support

The standby mode is controlled by the SCCU block. In the standby mode the shaper is switched off and some clocks are switched to clk_32k, controlled by signals from the SCCU. The other clocks have to be adjusted by SW before entering and after leaving the standby mode.

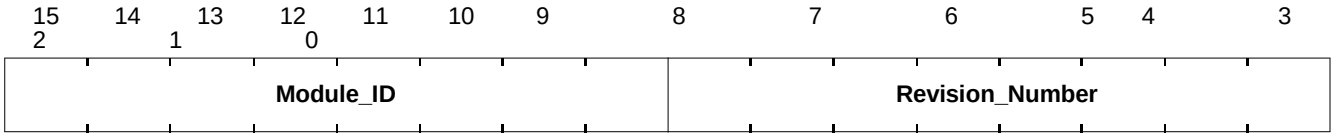
7.2.1.5 Clock Control Register Descriptions

The CGU is located on the X-bus. For the mapping of these registers refer to **Section 10.2 X-Bus Register Addresses**.

7.2.1.5.1 Clock Control Identification Register

CGUID

Clock Control Identification Register **Reset values: 4005_H**

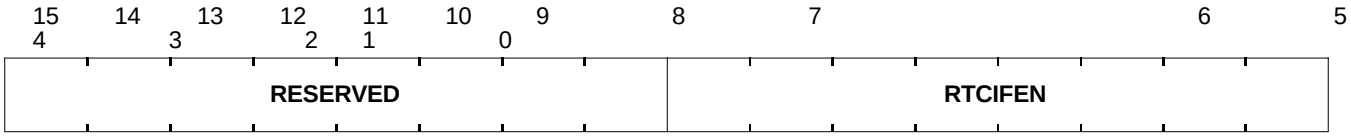


Field	Bits	Type	Description
Revision_Number	7:0	r	**CGU Revision Number** These hard-wired bits are used for the CGU revision numbering.
Module_ID	15:8	r	**CGU Identification Number** These hard-wired bits are used for CGU identification numbering.

7.2.1.5.2 Real Time Clock Interface Enable

RTCIF

Real Time Clock Interface Enable **Reset value: 0000_H**



Field	Bits	Type	Description
RTCIFEN	7:0	rw	**RTC Interface Enable Field** 10101010:Enables the RTC register interface access via the level shifter and clock enabling (required for register R/W operation) Any other:Disables the RTC register interface access (greatly reduces vdd_RTC current consumption during CPU_not_in_idle)
Reserved	15:8	r	Reserved, these bits must be left at their reset values.

7.2.1.5.3 Master Clock Control Register

MST_CLK_CTRL

Master Clock Control Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSP_SEL			CBUSH	CLK_OUTL_EN			reserved	MSEN		CPUPRE			CPUH		

Field	Bits	Type	Description
CPUH	0	rw	**26Mhz Operation of the MCU and Serial Interfaces** The maximum clock for the MCU and serial interfaces is: 0 26 MHz 1 reserved
CPUPRE	3:1	rw	**MCU Clock Prescaler Factor** Sets the divider factor of the MCU prescaler, **CPUPRE** = 2^n (n = 0..7)
MSEN	5:4	rw	Mixed Signal Clock Enable Clk_ms is switched to: 00 Clk_in (26 MHz) 01 Clock not enabled 10 Clk_pll (104 MHz or 26 Mhz, according PLL bypass value) 11 Clock not enabled
Reserved	7:6	rw	**Reserved**, these bits must be left at their reset values.
CLK_OUTL_EN	11:8	rw	Output Clocks Enables Bit 8 0 Disables 1 Enables CLKOUT_26 Bit 9 0 Disables 1 Enables CLKOUT_32K Bits 11:10 00 No clocks applied on output CLK_OUT 01 Enables clk_pdbus 10 Enables clk_xbus 11 Enables clk_cpu_master
CBUSH	12	rw	**Update GSM Timer** 0 No update 1 Update value
DSP_SEL	15:13	rw	**DSP Mode Select** Selects the clock source for clk_dsp: 000 Clk_in clock selected (26 MHz) 001 Clk_phs1 clock selected (refer to **Table 15 Phase Shifter Frequencies (on Page 79)**) 010 Clk_pll clock selected (104 MHz) 011 clk_32k clock selected (32KHz) 100 clk_dsp clock signal fixed to 0 otherclk_dsp clock fixed to 0 Hz. *Note: When in the Power Down Mode, clk_dsp is no longer fixed at 32KHz, but is controlled by this bit field.*

7.2.1.5.4 PLL Control Register

PLL_CTRL

PLL Control Register **Reset value: 0711_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pll_locked	**Pll_sh_invcl**	**PLL_N**						**PLL_M**		**Pll_sh_bypass_n**	**Pll_sh_powerup**	**Pll_bypass_n**	**Pll_powerup**		

Field	Bits	Type	Description	Reset
Pll_powerup	0	rw	**Powers Up the Analog Part of the PLL** 0: Disables 1: Enables	**1**
Pll_bypass_n	1	rw	**PLL Bypass Control** 0: PLL bypassed 1: PLL not bypassed *Note: The PLL bypass control is effective immediately and does not depend on bit **Pll_locked**.*	**0**
Pll_sh_powerup	2	rw	**Enables Shaper Power Up Model** 0: Powers down the Shaper (pll_sh_clk_o is 1 or 0 depending on the value of **Pll_sh_invcl**) 1: Powers up the Shaper *Note: The power down is effective only if the standby mode has been activated (32kHz activated)*	**0**
Pll_sh_bypass_n	3	rw	**Enables Bypass Mode of the Shaper** 0: Enable PLLM_sh_clk_0 = $\overline{PLL\_clkref\_i}$ Remark: no dependent of **Pll_sh_invel** 1: Disable	**0**
PLL_M	7:4	rw	**Input Divider Factor M in the PLL**	**1**
PLL_N	13:8	rw	**Feedback Divider Factor N in the PLL** PLL acts as a frequency synthesizer following the equation: N (t x $F_{pllclkref\_}$ F_{pll_clkout} —————————— M (t)	**7**
Pll_sh_invcl	14	rw	**Invertion of the Shaper Output** 0: Disables 1: Enables	**0**
Pll_locked	15	r	**PLL Status** 0: PLL not locked 1: PLL locked	**0**

7.2.1.5.5 Phase Shifter Control Register

PHX_CTRL

Phase Shifter Control Register **Reset value: 4200_H**

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0															
RESERVED	Phs_X			Phs_Y				RESERVED			RESERVED		Phs1_bypass_n	Phs1_powerup	RESERVED

Field	Bits	Type	Description	Reset
Phs1_powerup	1	rw	**Phase Shifter 1 Power Up Enable** 0: Disables Pll_phs1_clk_o 1: Enables Pll_phs1_clk_o	**0**
Phs1_bypass_n	2	rw	**Phase Shifter 1 Bypass Control** 0: Pll_phs1_clk_o bypassed 1: Pll_phs1_clk_o not bypassed *Note: The Pll_phs1_clk_o control is effective immediately and does not depend on bit **PLL_CTRL**.Pll_locked.*	**0**
Phs_Y	11:8	rw	**Y Factor of Phase Shifter 1** Refer to **Table 15**.	**2**
Phs_X	14:12	rw	**X Factor of Phase Shifter 1** Refer to **Table 15**.	**4**
Reserved	0,3:4 7:5, 15	r	Reserved, these bits must be left at their reset values.	

Table 15 Phase Shifter Frequencies

Frequency	Phs_X	Phs_Y
48 MHz	2	4
52 MHz	0	4
62.4 MHz	2	3
69.3 MHz	0	3
78 MHz	4	2
89.1 MHz	2	2
96 MHz	1	2
113.5 MHz	5	1
124.8 MHz	4	1
138.7 MHz	3	1
156 MHz	2	1

7.2.1.5.6 Serial Interface Clock Select Register

SIFCLKS

Serial Interface Clock Select Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCL		GPTCL		CAPCOMCL2		CAPCOMCL1		SSCCL		IICCL		reserved		ASC0CL	

Field	Bits	Type	Description For CPUH = 0
ASC0CL	1:0	rw	**ASC0 Clock Select** 00: 6.5 MHz 01: 13 MHz 10: 26 MHz 11: 26 MHz
IICCL	5:4	rw	**IIC Clock Select** 00: 6.5 MHz 01: 13 MHz 10: 26 MHz 11: 26 MHz
SSCCL	7:6	rw	**SSC Clock Select** 00: 6.5 MHz 01: 13 MHz 10: 26 MHz 11: 26 MHz
CAPCOMCL1	9:8	rw	**CAPCOM1 Clock Select** 00: 3.25 MHz 01: 6.5 MHz 10: 13 MHz 11: 26 MHz *Note: Maximum CAPCOM1 clock is clk_master/4.*
CAPCOMCL2	11:10	rw	**CAPCOM2 Clock Select** 00: 3.25 MHz 01: 6.5 MHz 10: 13 MHz 11: 26 MHz *Note: Maximum CAPCOM2 clock is clk_master/4.*
GPTCL	13:12	rw	**GPT Clock Select** 00: 3.25 MHz 01: 6.5 MHz 10: 13 MHz 11: 26 MHz *Note: Maximum GPT clock value is clk_master/4.*

Field	Bits	Type	Description For CPUH = 0
PCL	15:14	rw	**PCL Clock Select** 00: 6.5 MHz 01: 13 MHz 10: 26 MHz 11: 26 MHz
Reserved	3:2		**Reserved**

7.2.1.5.7 Reset Control and Status Register

RST_CTRL_STA

Reset Control And Status Register **Reset value: 0100_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSP STA	PLL_PHS1_CORR_RESY	PLL_CLKREF_FAIL_RES	RESERVED NC	RESERVED			RF_RESET	SCCUCL		RTCCL		reserved	DSP_RESET	SIM_RESET	RTC_RESET

Field	Bits	Type	Description
RTC_RESET	0	rw	**RTC Software Reset** 0: No action 1: Reset applied
SIM_RESET	1	rw	**SIM Software Reset** 0: No action 1: Reset applied
DSP_RESET	2	rw	**DSP Software Reset** 0: No action 1: Reset applied
Reserved	3	r	**Reserved**
RTCCL	5:4	rw	**RTC Clock Select** **For CPUH = 0** 00: 6.5 MHz 01: 13 MHz 10: 26 MHz 11: 26 MHz
SCCUCL	7:6	rw	**SCCU Clock Select** (Refer to **Section 7.2.2.4 Setup (on Page 95)**) **For CPUH = 0** 00: 6.5 MHz 01: 13 MHz 10: 26 MHz 11: 26 MHz
RF_RESET	8	rw	**Reset of RF Part** 0: Releases the reset

1: Activates the reset

Field	Bits	Type	Description
Reserved	12	r	**Reserved**
PLL_CLKREF_FAIL_RESYNC	13	r	**pll_clkref_fail_resync** 0: A reference clock is applied 1: No reference clock is applied
PLL_PHS1_CORR_RESYNC	14	r	**pll_phs1_corr_resync** 0: No error in the counter of the phase shifter 1 1: An error has occurred in the phase shifter 1
DSP_STATUS	15	r	**DSP Status signal** 0: DSP not clocked 1: DSP clocked
Reserved	11:9	r	Reserved, these bits must be left at their reset values.

7.2.2 Standby Clock Control Unit (SCCU)

7.2.2.1 Functional Overview

The SCCU provides all control signals for the transitions from the normal mode to the standby clock mode and back.

7.2.2.1.1 Clock Control Function

The SCCU controls the timing of the transitions from normal clock mode to standby clock mode and back to normal clock mode. For timing critical parts of the device like the GSM timer unit the SCCU guarantees an accurate time synchronization after return from standby clock mode.

In standby clock mode basic wake-up functions are still active permitting fast return to active mode whenever required, for example, when a key on the keypad has been pressed. Output signal DCXO_EN may be used to control the DCXO. It will be de-asserted whenever the DCXO is not needed.

The clock control functionality splits up into two parts, the GSM timer clock control and the system clock control.

7.2.2.1.2 System Clock Control

The system clock control function permits switching-over to the slow standby-clock for the major part of the system whenever no activity is required.

The system clock control function also controls the switching off of the external DCXO, which may be switched off if neither the system nor the GSM timer requires the DCXO any more.

Prior to return from the standby clock mode to normal clock operation the SCCU generates an external signal enabling the DCXO again. The pre-wakeup time interval between turn-on of the DCXO and return to normal clock operation is selectable in the range from 0 up to 508 RTC clock cycles. This corresponds to a maximum pre-wakeup time of 15.5 ms.

7.2.2.1.3 GSM Timer Clock Control

The counter of the GSM Timer Unit (refer to **Section 7.9 GSM Timer Unit**) runs off a 2.166 MHz clock derived from the DCXO reference clock. Since this clock is not available in standby clock mode the GSM Timer has to be stopped during standby clock mode. To permit accurate re-synchronization with the GSM network after the end of the standby clock phase, the GSM timer clock is stopped for an accurate integer multiple of the length of a GSM TDMA frame. Therefore, the GSM timer remains in synchronization with the network except for the TDMA frame number which has to be adjusted by software.

The duration of the GSM timer stop phase may be defined in two ways. It may be predefined by software. This is done by writing the number of TDMA frames to a control register. Alternatively it may be terminated by a hardware-controlled wake-up function triggered e.g. by a key being pressed on the keypad. Thereby generating an early termination of the sleep phase can be generated. Also with the early termination of the GSM timer stop phase the length of the stop phase is kept to an accurate multiple of a GSM TDMA frame.

To permit return from standby clock mode after an accurate integer multiple of TDMA frames the SCCU maintains an internal time scale based upon the RTC clock. Since the frequency tolerance of the RTC clock is much higher than that of the DCXO calibration of the RTC clock is required to achieve the targeted accuracy. This calibration has to be done before entering standby clock mode for the first time. It has to be repeated in regular intervals to account for a potential temperature drift of frequency between the RTC clock frequency.

7.2.2.1.4 Wake-Up Functions

Return from standby clock mode to normal operation (wake-up) may be triggered either by C166 software or by hardware functions.

Software Triggered Wakeup

By setting a control bit in the SCCU C166 may trigger a return from standby clock mode to normal operation (wake-up) if C166 is running off the standby clock. In this case C166 can trigger a return to normal clock mode whenever an interrupt occurs.

Hardware Triggered Wakeup

Besides that the SCCU permits hardware triggered wakeup from standby clock mode by some predefined sources. Wakeup may be triggered by one of the following interrupt sources:

1. Keypad event
2. SIM card interrupt
3. RTC interrupt
4. Interrupt on one of the external interrupts.
5. Event on the interrupt lines (IRQ or FIR) from the interrupt control unit (ICU) to C166.

Each of these sources can be enabled individually by setting a bit in the control registers **SCCUHWWAKEUPH** and **SCCUHWWAKEUPL**. The interrupt sources from the peripherals are not affected by the peripheral internal enabling mechanism.

Enabling the direct hardware wake-up functions 1 to 4 permit the fastest wake-up. Using the alternate functions an external or CAPCOM interrupt may be used to trigger a wake-up on a rising or falling edge on most pins. This also applies to the serial interfaces.

Using the interrupt from the ICU causes a delay of severalclock cycles due to the internal processing of the ICU. It permits wakeup from all interrupt sources which have been enabled.

Wakeup may also be triggered by the interrupt handling routine by C166. This introduces even more clock cycles of delay depending upon software.

7.2.2.1.5 Other Functions

Besides the clock control functions the SCCU also controls functions to reduce the leakage current of the device.

7.2.2.1.6 Power Control

The voltage regulators, located in the power management unit (PMU), can be turned off during standby clock mode. For this purpose, output signal VCXO_EN is available on the device, which is controlled by the internal signal *Vcxo_en* from the SCCU. This signal will be de-asserted in standby mode. The power management unit (PMU) should be programmable to turn off the analog supply voltage and the DCXO supply voltage when output signal VCXO_EN is de-asserted.

Reset Control

If VDD_ANA is switched off in standby power-down, the logic has to be reset upon power-up. For this purpose the SCCU has to be set up to generate the reset signal *SCCU_resa* for the analog supply domain. This signal is already asserted when the system is ready for standby power-down and is kept asserted until power (and clock) has been applied again. The corresponding bit **SCCUSPCR.APDN** has to be set for that purpose as shown in **Figure 21**. The controller will have to set this bit to its appropriate values prior to first entering standby power-down.

Table 16 Setup of register SCCUSPCR

VDD_ANA	Register SCCUSPCR
	APDN
on	0
off	1

The reset signals are routed to the controlled blocks via the Section 7.7 SCU (on Page 225).

Analog Supply Domain Interface Control

If the analog supply voltage is switched off in standby clock mode, all output signals from this domain to the other supply domains are tied to their reset values. Else floating levels on these outputs might cause short-circuit currents in the other parts of the circuit.

This is avoided by means of tie cells (T) as shown in **Figure 21**. When enabled, the tie cells tie their output signal to a predefined value, else they are transparent to the signal. The tie cells are controlled by signal *fr_ana_out.* This signal is activated one clock cycle before the reset is applied to the blocks to be powered down. The sequence is triggered by signal *SCCU_res* from the SCCU main state machine.

With a delay of one real-time clock cycle after the tie signals have been asserted, the reset signal for the analog part *SCCU_resa* is asserted.

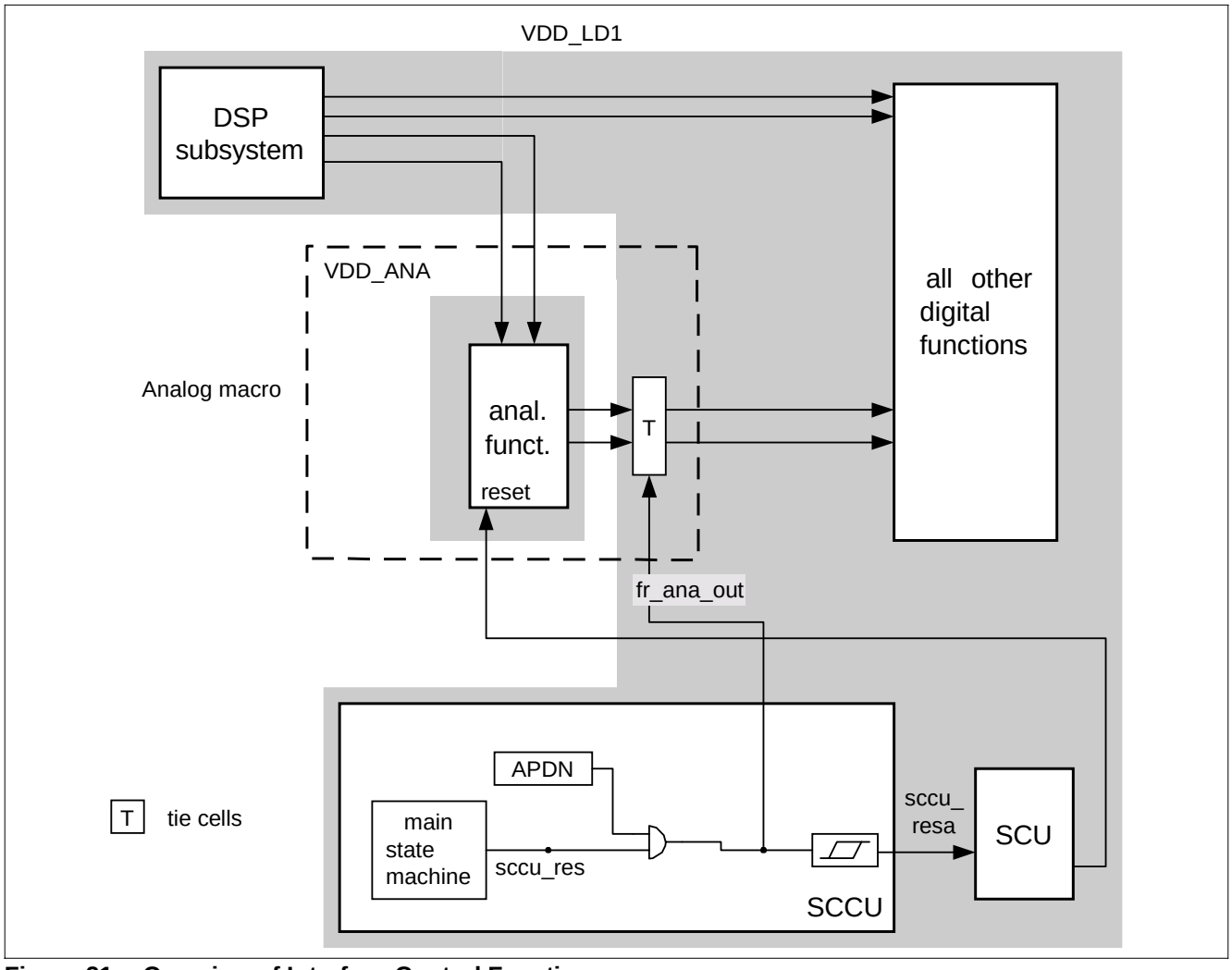


Figure 21 Overview of Interface Control Functions

If the analog supply domain is powered down during standby power-down (bit APDN set) the analog block has to be reset upon power-up. This is done by signal *SCCU_resa* sent to the SCU. Signal *SCCU_resa* is asserted one X-Bus bus clock cycle after signal *fr_ana_out*. It is de-asserted simultaneously with signal *fr_ana_out*.

Before the analog supply domain is powered down all analog input pins have to be at low level. They have to be kept at low level during standby power-down.

If the analog subsystem is selected to be switched off during the standby clock mode, a reset signal is generated during this phase. After termination of the standby clock mode phase, the analog subsystem will then leave reset and re-start.

7.2.2.2 System Clock Control

The hardware controlled sequence of standby power-down states is shown in **Figure 22**. The state machine is clocked with the standby clock from the RTC block. The function of the control signals from the GSM sleep timer is shown in **Figure 23**.

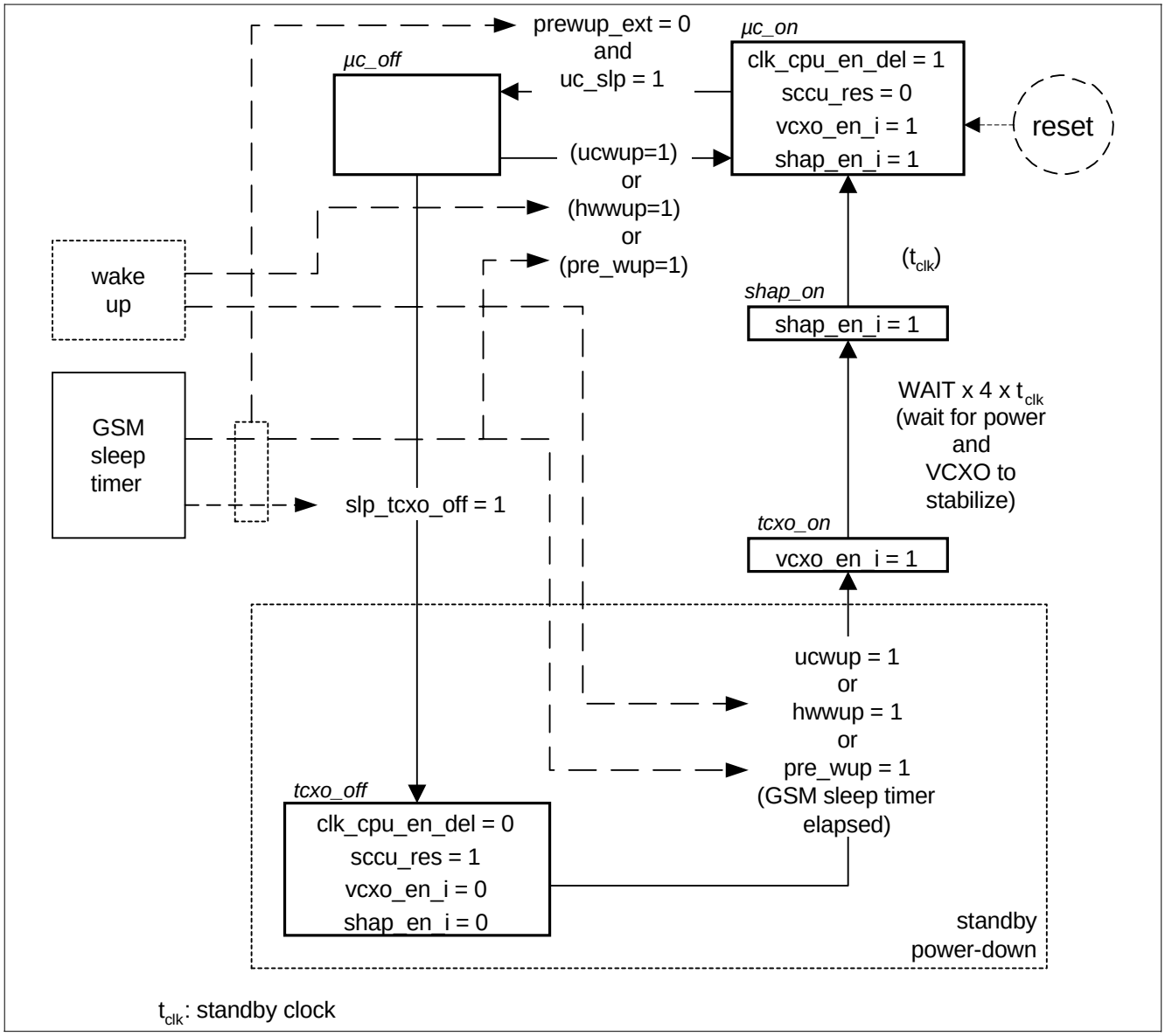


Figure 22 SCCU Main State Machine

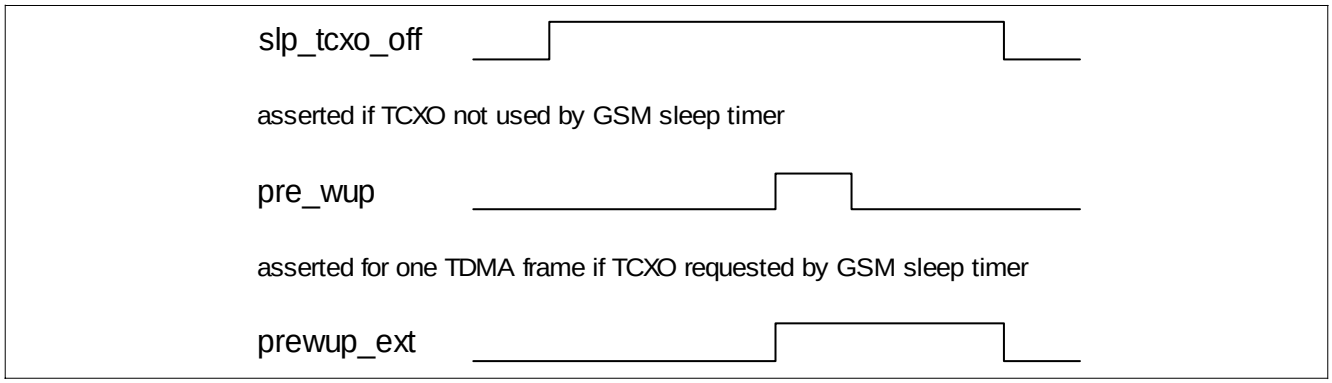


Figure 23 Function of Signals from GSM Sleep Timer

Entering Standby Power-Down (*µc_on -> µc_off)*

After reset the main state will be in state *µc_on*.

To enter standby power-down the controller will have to set bit **SCCUSCCTRL.UCSLP**. The SCCU main state machine enters state *µc_off* which signals that the reference clock is no longer required as system clock.

If the GSM sleep timer is already in the wake-up phase and a signal *prewup_ext* is asserted, the transition is inhibited.

Transition from state *µc_off* to state *µc_on*

As long as the SCCU main state machine is in state *µc_off,* the controller can abort the transition to standby power-down by setting bit **SCCUSCCTRL.UCWUP**, which triggers the transition to state *µc_on*.

The transition to state *µc_on* is also triggered if the pre-wakeup phase is started by the GSM sleep timer (signalled by assertion of signal *pre_wup*). This transition has priority over the transition to state *tcxo_off*.

Transition from state *µc_off* to state *tcxo_off*

This transition is enabled only by an output signal from the GSM sleep timer. When the GSM sleep timer has frozen the GSM timer and runs off the standby clock, it asserts signal *slp_tcxo_off*. This triggers the transition to state *tcxo_off* which triggers the transition to standby power-down mode as shown in **Figure 24**.

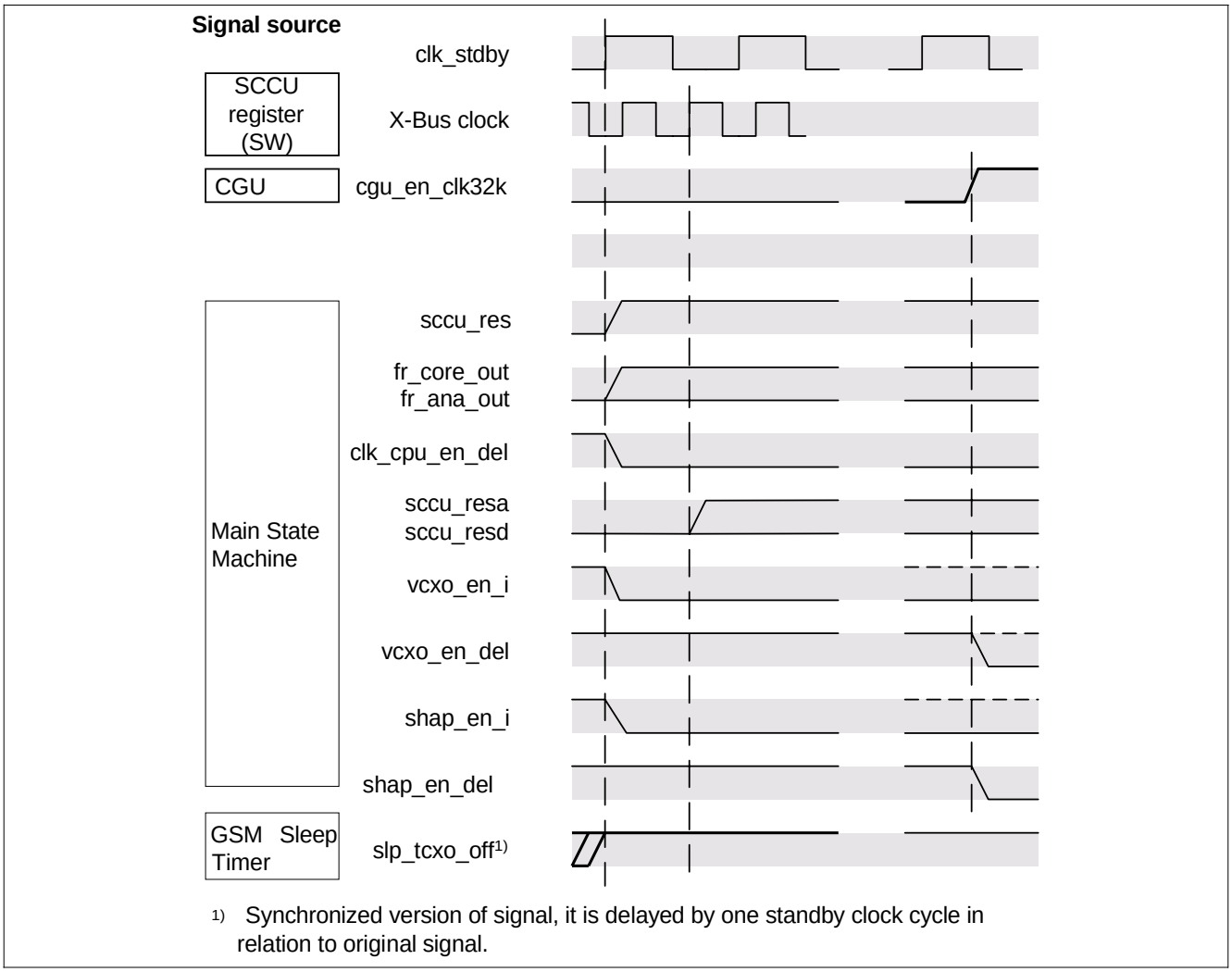


Figure 24 Timing Diagram for Transition µc_off -> tcxo_off

In state tcxo_off the switch-over of the system clock (*clk_tc_m* from the reference clock from VCXO) to the standby clock (from the RTC) is started by de-asserting signal *clk_cpu_en_del* to the clock generation unit (CGU). Also signals vcxo_en_i and shap_en_i are set to 0 thereby enabling switching off of the external VCXO and the shaper a few clock cycles later.

To prepare for a standby power-down (see **Figure 21Overview of Interface Control Functions (on Page 85)**):

- Set bit **SCCUSPCR.APDN** (signal *fr_ana_out* is activated).

One X-Bus bus clock cycle later signals *SCCU_resa* are asserted.

The switch-over from the reference clock to the standby clock requires approximately 6 standby clock cycles. The completion of the switch-over is signaled by signal *cgu_en_clk32k* from the CGU being asserted. Only after the switch-over has been completed are signals *vcxo_en* and *shap_en_del* (controlling the external power-management IC and the shaper in the CGU) set to 0. Depending upon the setup of the power management IC the power supply of the VCXO, the baseband DSP subsystem supply and the analog supply may be switched off by the de-assertion of output signal *vcxo_en*.

Transition from State *tcxo_off* to State *tcxo_on*

If the SCCU is in state *tcxo_off*, wakeup can be triggered by several sources: the GSM sleep timer, the hardware wake-up functions or the controller (see **Figure 25**). If the GSM sleep timer of the SCCU elapses, signal *pre_wup* is asserted. If the hardware wake-up function detects an event signal, *hwwup* is asserted if the corresponding bit in **SCCUHWWAKEUPH or SCCUHWWAKEUPL** is set. For example, for *hwwup* to be asserted both:

- A SIM interrupt must occur
- **SCCUHWWAKEUPL.SIM_EN** must be set.

Any interrupt can trigger the controller to set bit UCWUP thereby asserting signal *ucwup*.

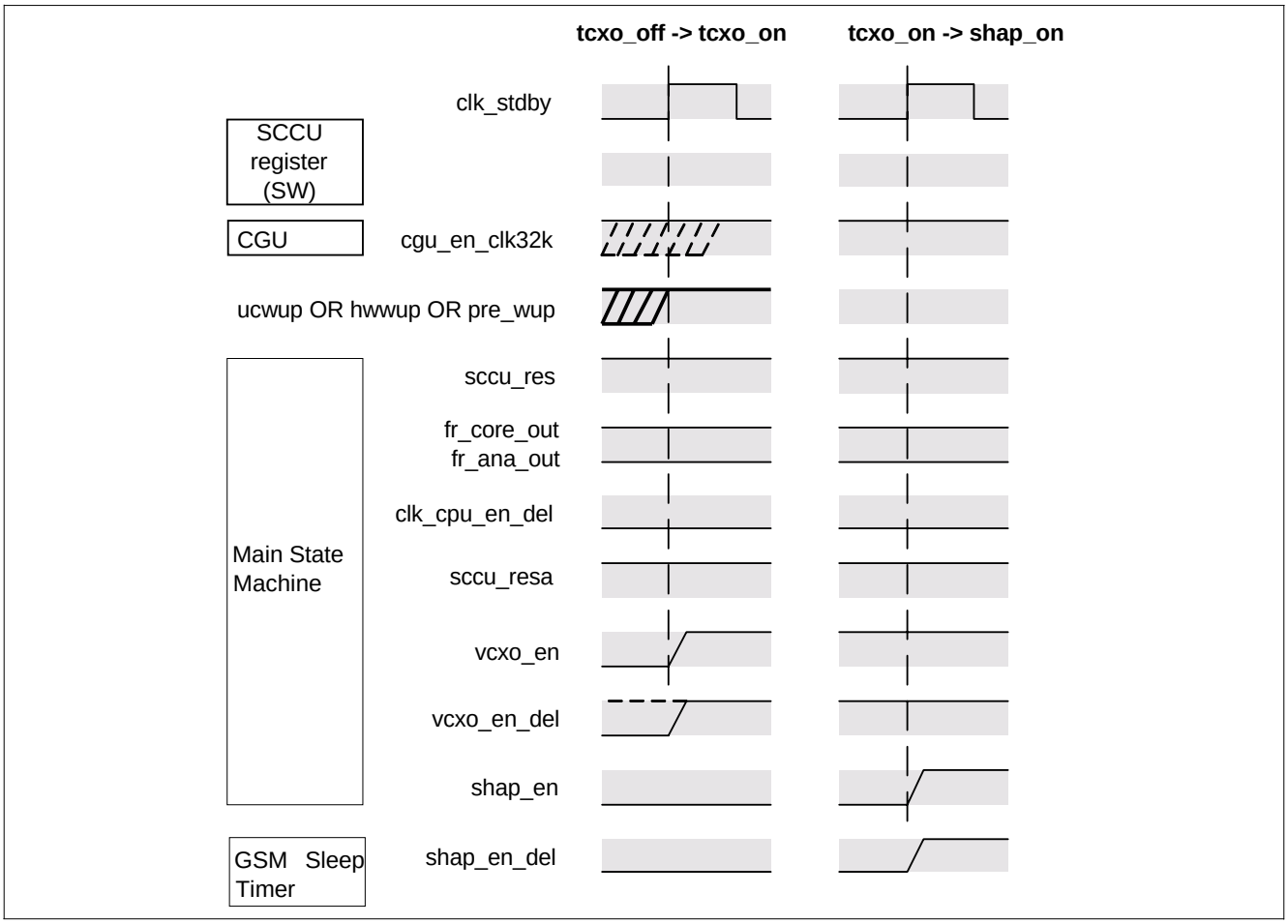


Figure 25 Timing Diagrams for Transitions tcxo_off -> tcxo_on -> shap_on

Transition from state *tcxo_on* to state *µc_on*

The transition from state *tcxo_on* to state *µc_on* occurs after a predefined time interval set by value in **SCCUWAITH.WAIT**. After 4 * **WAIT** + 1 cycles of the standby clock the system clock switches back to the VCXO clock as shown in **Figure 26**.

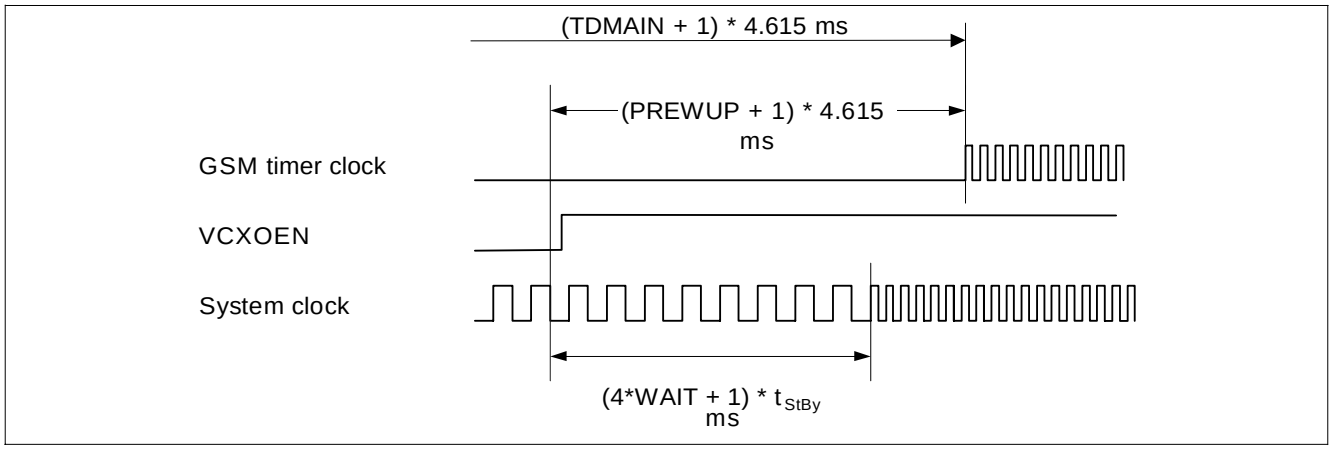


Figure 26 Timing of Wake-Up Phase

$$^{1)} > (4*WAIT + 1)*t_{StBy}{}^{2)}$$

*Note: Always make sure that (PREWUP + 1) * t_{Frame} because if WAIT is at its maximum (127), PREWUP must equal 3 to insure that the relation remains true. In this case, the system clock wakeup time is 15.5 ms, refer to the registers in* ***Section 7.2.2.5.11 High Pre-Wakeup and Wait Registers (on Page 100)*****.**

Table 17 SCCU Signal Descriptions

Signal Name	I/O	From/to Block	Value	Function
sccu_slp_tcxo_off	I	GSM sleep timer	1	VCXO clock no more required for GSM sleep timer - GSM timer frozen
			0	VCXO clock still required by GSM sleep timer
sccu_pre_wup	I		1	VCXO clock requested for preparation of end of GSM timer sleep period
			0	VCXO clock not requested
sccu_hwwup	I	Hardware		Wakeup signalling service request from hardware wakeup sources
sccu_cgu_en_clk32k	I	CGU		Enables power-down of VCXO
sccu_shap_en_del	O	Shaper		Enables shaper
sccu_clk_cpu_en_del	O		1 0	Normal 32 kHz standby	Selects system clock clk_tc_m (except GSM timer)
sccu_clk_gsm_en	O		1	Clock supplied to GSM timer
			0	No clock supplied to GSM timer
sccu_vcxo_en_del	O	Power Mgmt. IC		Main power control signal also connected to pin VCXO_EN for control of power management IC
sccu_fr_dsp_out	O	Tie cells		Ties the output level of all signals from the DSP supply domain

1) t_{Frame} = 4.615 ms

2) $t_{StBy} = \frac{1}{32768\ Hz}$ sec

Table 17 SCCU Signal Descriptions

Signal Name	I/O	From/to Block	Value	Function
sccu_fr_ana_out	O	Tie cells		Ties the output level of all signals from the analog supply domain
sccu_resa				Resets analog supply domain

7.2.2.3 GSM Timer Control

7.2.2.3.1 Synchronous Wake-Up Concept

To permit synchronous wake-up of the mobile after the end of the standby power-down phase, the GSM timer is frozen for a fixed time representing an accurate multiple n_{TDMA} of the TDMA frame duration t_{TDMA}. The duration of the freeze period is controlled by the GSM sleep timer of the SCCU. Since only the inaccurate standby clock is available during standby power-down this clock will have to be calibrated against the highly accurate clock of the 26 MHz system reference oscillator (VCXO) prior to entering standby power-down. A detailed description of the calibration can be found in **Section 7.2.2.1.3 GSM Timer Clock Control (on Page 82)**. After this calibration the GSM timer freeze time interval can be accurately expressed as sum of an integer multiple n_{stdby} of the standby clock period t_{stdby} and an integer multiple n_{ref} of the reference clock period t_{ref}:

$$n_{TDMA} \otimes t_{TDMA} = n_{stdby} \otimes t_{stdby} + n_{ref} \otimes t_{ref} \quad (3)$$

Therefore, an accurate wake-up after the freeze time interval is guaranteed, it is only limited by the stability of the standby clock frequency over the freeze interval.

7.2.2.3.2 GSM Sleep Timer

The GSM state machine freezes the GSM timer for a defined number of TDMA frames (5 to 8192 frames, that is, 0.023 to 37 s). After the end of the sleep period no correction to the GSM timer is required. Only the TDMA frame counters have to be updated.

To start the sleep function, the controller writes the number of TDMA frames diminished by 1 to register **SCCUTDMINL**.

The GSM timer sleep phase is activated by setting bit **SCCUSLPCTRL.SLPEN**. Depending upon the value of bit **SCCUSLPCTRL.HWACTDI** there are two modes:

1 The sleep phase starts when bit **SLPEN** is set. Thereby the start of the sleep phase is purely software controlled.

0 The start of the sleep phase may be controlled by the rising edge of the GSM timer signal SLPSTART.
- If signal SLPSTART is already high when bit **SLPEN** is set, the sleep phase starts immediately.
- If signal SLPSTART is low when bit **SLPEN** is set, the sleep phase starts with the next rising edge of signal SLPSTART (hardware controlled).

After the end of the sleep phase the stand-by clock control block resets bit **SLPEN** to 0. To monitor the duration of the sleep phase, poll **SLPEN**.

Before the end of the sleep period the GSM state machine requests the VCXO reference clock by asserting signal pre_wakeup (see **Figure 27**). If the VCXO has been disabled, the number of TDMA frames before the end of the sleep period is defined by the value in **SCCUWAITH.PREWUP**.

The SCCU can signal to the controller the end of the sleep phase by asserting interrupt signal ECO_INT(see **Figure 27**).

Notes

1. *The length of the TDMA frame used for the GSM sleep phase is always based upon the nominal TDMA frame length of 60 000 13 MHz clock cycles corresponding to a TDMA timer overflow value of 10 000. Any differing values stored in registers elsewhere (for example, in the GSM Timer Unit)are not taken into account.*
2. *During the sleep phase the number of TDMA frames slept is shown in **SCCUTDMOUTL.TDMAOUT** and may be read by the controller.*

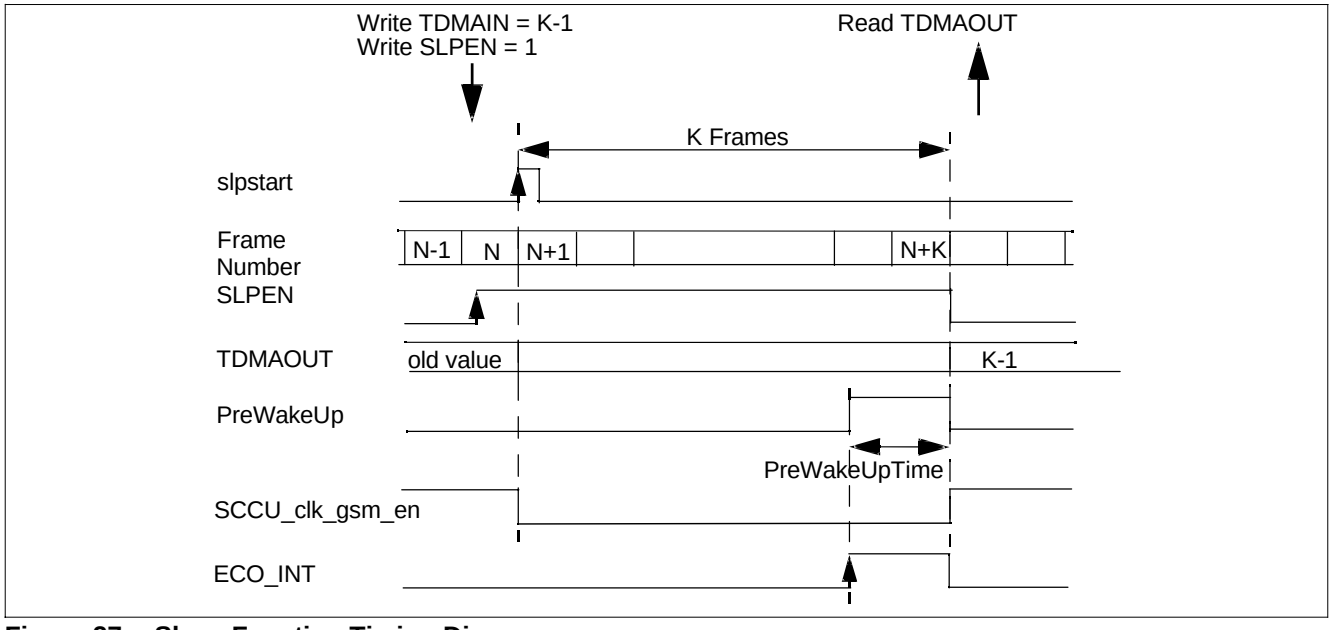


Figure 27 Sleep Function Timing Diagram

Early Termination of GSM Timer Sleep Phase

If the GSM timer functionality is required, the GSM timer sleep phase can be aborted by setting bit **SCCUSLPCTRL.SLPSTP** (see **Figure 28**).

The duration of the sleep period can be calculated from **SCCUTDMOUTL.TDMAOUT** after the sleep phase has been terminated. The number of TDMA frames equals **TDMAOUT** incremented by 1.

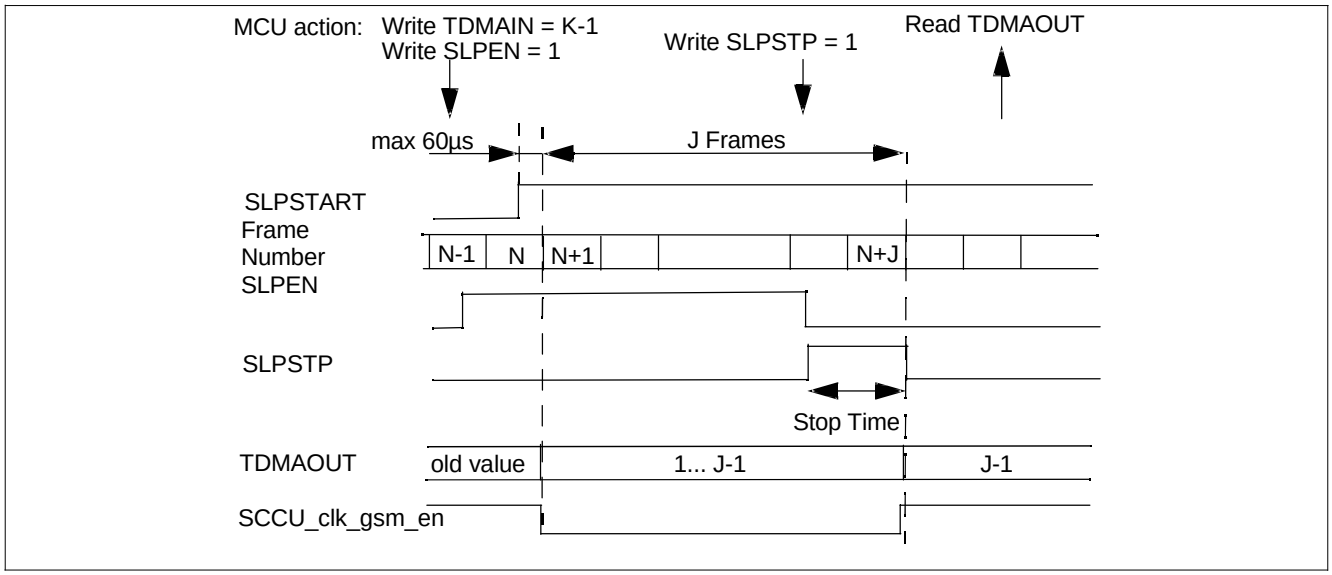


Figure 28 Early Termination of GSM Timer Sleep Phase

7.2.2.3.3 Stand-By Clock Calibration

Principle of Calibration

To be able to sleep for a GSM sleep timer period with a duration of an exact multiple of a TDMA frame, the duration of a TDMA frame t_{TDMA} has to be expressed as sum of a multiple NQTZ of a standby clock period t_{stdby} and a multiple n_{ref} of a reference clock period t_{ref} as shown in the equation below:

$$t_{TDMA} = NQTZ \cdot t_{stdby} + n_{ref} \cdot t_{ref} \quad (4)$$

During the calibration value n_{ref} has to be determined. For this purpose the required multiple of the standby clock period NQTZ * t_{stdby} has to be measured accurately in terms of reference clock periods t_{ref}.

To increase the accuracy of this calibration 16 TDMA frames are chosen for this calibration procedure instead of one given in **Equation (4)**:

$$16 \cdot t_{TDMA} = 16 \cdot NQTZ \cdot t_{stdby} + 16 \cdot n_{ref} \cdot t_{ref} \quad (5)$$

To measure value 16 * n_{ref} each TDMA frame is divided into 8 interval of equal length (corresponding in length to the TDMA time slots). In each of these intervals the REFOUT counter is running with the reference clock being decremented from 7500 to 0. In the last (8th) interval of the 16th TDMA frame the counter is stopped when the predefined number (16 * NQTZ) of standby clock cycles has elapsed as shown in **Figure 29**. The value of the REFOUT counter then shows exactly the 16-fold value of n_{ref}.

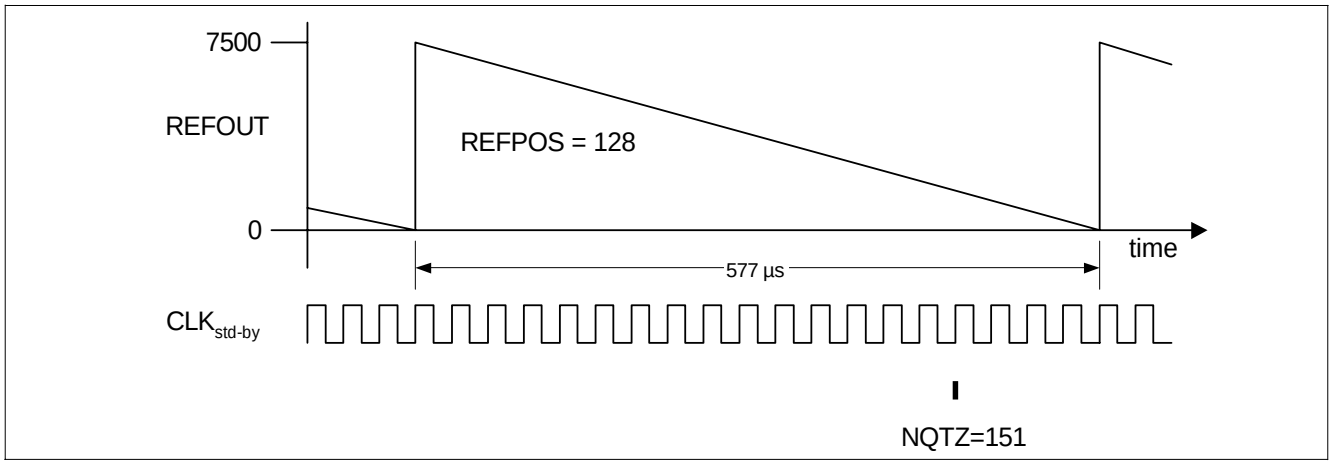


Figure 29 Stand-By Clock Calibration

The number of standby clock cycles per TDMA frame NQTZ has to be selected so that it is slightly below the TDMA frame duration.

For a nominal standby clock frequency of 32 768 Hz the ration of the TDMA frame length to the standby clock period is 151.2:

$$\frac{T_{TDMA\ frame}}{T_{slot}} = \frac{1}{8} \cdot \frac{75}{130} \cdot 10^{3} = 151.2 \quad (6)$$

Choosing NQTZ = 151 gives a value of n_{ref} of 94. The nominal value of 16 * n_{ref} for the 16 TDMA frame calibration period is 1504. This is the value of counter REFOUT for a standby clock frequency of exactly 32 768 Hz.

In general, the value of REFOUT for a stand-by clock frequency $f_{stand\text{-}by}$ can be calculated by

$$REFOUT = 128 * 7500 - NQTZ * 16 * f_{sccu}/f_{stby} \quad (7)$$

where $f_{sccu = 13\ MHz}$ (refer to **RST_CTRL_STA**.SCCUCL)

The optimum achievable resolution of the calibration can be calculated from the calibration time used. Since 16 TDMA frames are used for the calibration the total number of reference clock cycles used is 16 * 4.615 ms * 13 MHz = 960 000. The optimum achievable resolution, therefore, is about 1 ppm/LSB.

For a valid measurement the value of counter REFOUT must not be below 16. This limits the frequency tolerance to about ± 1500 ppm% which is far above the tolerance of standard crystals available on the market.

If a stand-by clock frequency other than 32 768 Hz is used, a similar calculation has to be done. The number of stand-by clock cycles NQTZ has to be adjusted so that values **SCCUREFH.REFPOS** and **SCCUREFH.REFOUT** meet the above criteria. The value NQTZ has to be chosen so that:

$$4.579 / kHz * f_{stand\text{-}by} < NQTZ < 4.615 / kHz * f_{stand\text{-}by}. \quad (8)$$

Operational Description

The calibration function is activated by setting bit **SCCUSLPCTRL.REFEN** to 1. As soon as the calibration has been completed the hardware resets bit **REFEN** to 0. By polling the value of this bit the end of the calibration can be determined. As soon as bit **REFEN** has been set, the values in registers **SCCUREFH** and **SCCUREFL** become invalid. When bit **REFEN** has been reset by the hardware and the calibration has been completed successfully, the new values in registers **SCCUREFH** and **SCCUREFL** are valid again.

*Note: The calibration is done completely independent from the system interface unit. All GSM timer functions, including modifications of registers **TCOR** or **TOVF**, can still be used without affecting the calibration process.*

A valid calibration has been performed if **SCCUREFH.REFPOS** equals 128 and value **SCCUREFH.REFOUT** is larger than 16. For a frequency of 32 768 Hz **REFOUT** is set to 1504.

An invalid calibration where the criteria mentioned above have been violated is signalled by bit **SCCUSLPCTRL.REFERR** being set to 1.

The result of the calibration contained in register **SCCUREFH.REFOUT** can be modified by the controller by writing a value to register **SCCUREFINL**. This new value will be copied to register **SCCUREFH** and will be used for the next sleep phase timing. Therefor, the controller can average the results over several calibration results.

7.2.2.4 Setup

All SCCU functions rely on the SCCU clock which is generated by frequency division in the BPI from the X-Bus clock. For proper operation of the SCCU the frequency of this clock has to be 13 MHz. Do not select other SCCU reference clock settings if the Stand-By Mode is entered (even though other values are available via **RST_CTRL_STA.SCCUCL**.

Before switching to standby-clock mode the PLL in the CGU should be by-passed and turned off to save energy. The X-Bus clock will then have to be set up in the CGU (refer to **Section 7.2.1 Clock Generation Unit (on Page 62)**) using the PLL by-pass.

After switch-back from standby-clock mode the SCCU clock must remain at 13 MHz until the switch-back is completed, that is, until the GSM timer clock has been switched back to the reference oscillator (see **Figure 26 Timing of Wake-Up Phase**). The state of the GSM sleep timer can be detected by evaluating bit **SCCUSLPCTRL.SLPEN**.

7.2.2.5 SCCU Registers

7.2.2.5.1 Overview

Table 18 SCCU Register List 1

Register Group	Register Name	Register Symbol
Identification	SCCU Identification	**SCCUID**
GSM Sleep Timer	Sleep Duration Value	**SCCUTDMINL**
	Sleep Duration Status	**SCCUTDMOUTL**
	Sleep and Calibration Control Register[1)]	**SCCUSLPCTRL**
	Pre-Wakeup and Wait Register[2)]	**SCCUWAITH**
Standby Clock Calibration	Reference Calibration Output Value High & Low	**SCCUREFH** & **SCCUREFL**
	Quartz Number Register	**SCCUNQTZ**
	Sleep and Calibration Control Register[3)]	**SCCUSLPCTRL**
	Reference Input for standby clock	**SCCUREFINL**
System Control	Switch Control Register	**SCCUSCCTRL**
	Pre-Wakeup Register Wait Register[4)]	**SCCUWAITH** **SCCUWAITL**
Other Registers	Standby Power Down Control	**SCCUSPCR**
	Hardware Wakeup Control Register High Hardware Wakeup Control Register Low	**SCCUHWWAKEUPH** **SCCUHWWAKEUPL**
	Clock Status Register	**SCCUCLKSTA**
	State Machine Status Register	**SCCUSMSTA**

1) Except bits REFEN and REFERR

2) Pre-Wakup bits only belong to GSM sleep timer

3) Bits REFEN and REFERR only
4) Wait bits are only for system control.

7.2.2.5.2 SCCU Identification Register

SCCUID

SCCU Identification Register **Reset values: 4004_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Module_ID								Revision_Number							

Field	Bits	Type	Description
Revision_Number	7:0	r	**SCCU Revision Number** These hard-wired bits are used for the SCCU revision numbering.
Module_ID	15:8	r	**SCCU Identification Number** These hard-wired bits are used for SCCU identification numbering.

7.2.2.5.3 Standby Power Control Register

SCCUSPCR

Standby Power Control Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DREN	DROFF	RESERVED			APDN	RESERVED	

Field	Bits	Type	Description
APDN	1	rw	0 No action 1 Analog supply domain reset in Standby Mode (refer to **Figure 21**)
DROFF	5	rw	0 DSP ROM is dependant on **DREM** 1 DSP ROM is off immediately
DREN	6	rw	0 DSP ROM is never off 1 DSP ROM is off when VCXO is off (**SCCUSMSTA.tcxo_off (S3)** = 1) and SHAPER POWER is down (**PLL_CTRL.Pll_sh_powerup** = 0)
Reserved	0, 2, 3, 4, 15:7	r	Reserved; these bits must be left at their reset values.

7.2.2.5.4 Sleep Duration Value Register

SCCUTDMINL

Sleep Duration Value **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				TDMAIN											

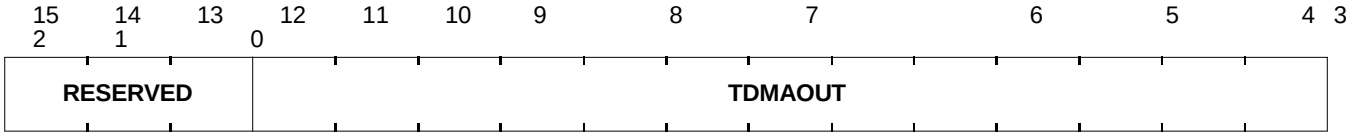
Field	Bits	Type	Description
TDMAIN	12:0	rw	Duration (-1) of the Sleep time (in TDMA frames) T_{sleep} = (TDMAIN+1)*4.615 ms

Field	Bits	Type	Description
Reserved	15:13	r	Reserved, these bits must be left at their reset values.

7.2.2.5.5 Sleep Duration Status Register

SCCUTDMOUTL

Sleep Duration Status **Reset value: 0000_H**

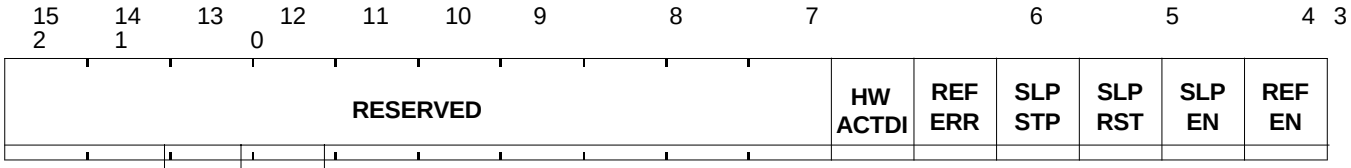


Field	Bits	Type	Description
TDMAOUT	12:0	rh	Number of frames -1 during which the GSM timer has remained frozen
Reserved	15:13	r	Reserved, these bits must be left at their reset values.

7.2.2.5.6 Sleep Control Register

SCCUSLPCTRL

Sleep Control Register **Reset value: 0000_H**



Field	Bits	Type	Description
REFEN[1]	0	rwh	**Reference Enable** If set by software, calibration of standby clock is started. Reset by hardware after the end of calibration of standby clock. Read values: 0 Calibration inactive 1 Calibration active
SLPEN[1]	1	rwh	**Sleep Enable** If set by software, activates the GSM sleep timer (sleep mode is started either immediately or after next rising edge of signal slpstart from GSM timer depending upon bit HWACTDI). Reset by hardware after GSM sleep timer has terminated. Read values: 0 GSM sleep timer inactive 1 GSM sleep timer running or activated
SLPRST[2]	2	rw	**Reset Sleep Counter** 0 Normal operation 1 Resets register **SCCUTDMOUTL**
SLPSTP[1]	3	rwh	**Sleep Stop** If set by software, synchronously stops the sleep counter. Reset by hardware when the sleep counter actually stops. Read values: 0 No sleep stop action pending (see **Figure 28 Early Termination of GSM Timer Sleep Phase (on Page 93)**) 1 Sleep stop action pending

Field	Bits	Type	Description
REFERR	4	rh	**Reference Error Flag** Set by HW during calibration. Reset by HW at the beginning of the calibration. 0 Calibration within range 1 Calibration out of range
HWACTDI	5	rw	**slpstart Activation Disable** 0 Sleep phase starts if SLPEN is set with rising edge of signal slpstart from GSM timer 1 Sleep phase starts when SLPEN is set
Reserved	15:6	r	Reserved, these bits must be left at their reset values.

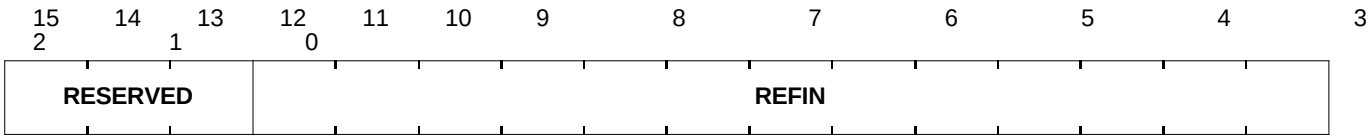
1) Writing 0 to this bit has no effect on hardware and does not reset this bit, it is only reset by hardware; when writing to this register always write 0 to this bit if the function is not activated.

2) Must not be released as long as **SCCUTDMOUTL.TDMAOUT** != 0.

7.2.2.5.7 Standby Clock Reference Input Register

SCCUREFINL

Standby Clock Reference Input Register **Reset value: 0000_H**



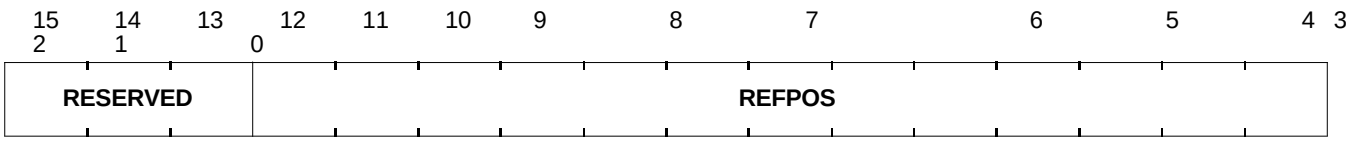
Field	Bits	Type	Description
REFIN[1)]	12:0	rw	Value of the reference controlled by the MCU (1 LSB = 1.043 ppm) optionally used for overriding value in register **SCCUREFL.REFOUT**
Reserved	15:13	r	Reserved, these bits must be left at their reset values.

1) **REFIN** is only taken into account after:
1. **SCCUREFL.REFOUT** has been modified by a calibration operation (**SCCUSLPCTRL.REFEN** = 1)
   => **SCCUREFL.REFOUT = REFOUT**$_{calibrated}$
2. **REFIN** is written to
   => **SCCUREFL.REFOUT** is not overwritten, **REFOUT** != **REFIN**
3. Do a Sleep Enable (**SCCUSLPCTRL.SLPEN** = 1)
4. Wakeup the GSM (**SCCUSLPCTRL.SLPEN** = 0)
   => Now **SCCUREFL.REFOUT** = **REFIN**.

7.2.2.5.8 Reference Calibration Output Values

SCCUREFH

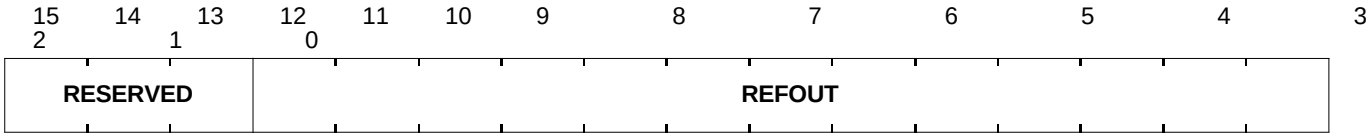
High Reference Calibration Output Value **Reset value: 0000_H**



Field	Bits	Type	Description
REFPOS	12:0	rh	Coarse measurement of the slow Xtal frequency (1 LSB = 1/8-th of TDMA frame). Nominal value is 128 (16 TDMA frames).
Reserved	15:13	r	Reserved, these bits must be left at their reset values.

SCCUREFL

Low Reference Calibration Output Value **Reset value: $05E0_H$**

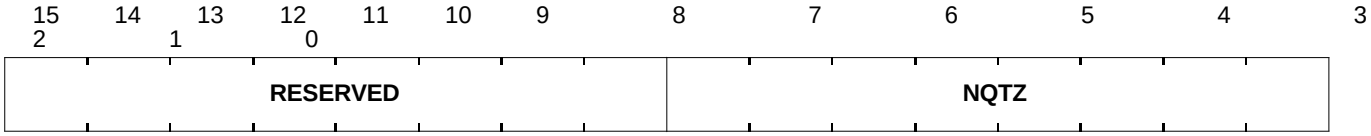


Field	Bits	Type	Description
REFOUT	12:0	rh	Fine measurement (1 LSB = 1.043 ppm)
Reserved	15:13	r	Reserved, these bits must be left at their reset values.

7.2.2.5.9 Oscilator Crystal Periods per Frame

SCCUNQTZ

Xtal Oscilator Number Register **Reset value: 0097_H**

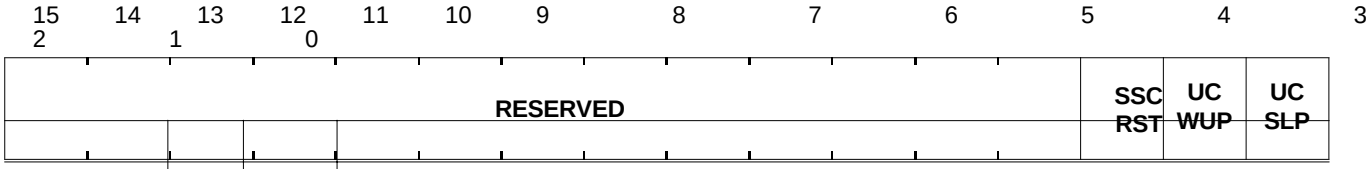


Field	Bits	Type	Description
NQTZ	7:0	rw	Number of slow Xtal oscillator periods in one TDMA frame (default 151)
Reserved	15:8	r	Reserved for future use; these bits must be left at their reset values.

7.2.2.5.10 Switch Control Register

SCCUSCCTRL

Switch Control Register **Reset value: 0000_H**



Field	Bits	Type	Description
UCSLP[1)]	0	wh	**Sleep Command** Set by software to start system sleep phase, then reset by hardware after 3 standby clock periods.
UCWUP[1)]	1	wh	**Wakeup Command** Set by software to start wakeup of system, then reset by hardware after 3 standby clock periods.
SSCRST[1)]	2	wh	**Reset Command** Can optionally be set by software to reset state machine. It is reset by hardware after 3 standby clock periods.
Reserved	15:3	r	Reserved for future use; these bits must be left at their reset values.

1) Bit can also be reset by software, but not required; resetting bit too early (before next rising edge of standby clock) may cancel function; bit will in any case be reset by hardware after 3 standby clock periods.
For example: set bit ucslp: write 0001_H, set bit ucwup: write 0002_H - potentially cancels function triggered by bit ucslp since 0 is written to bit ucslp; bit ucwup will in any case be reset after 3 standby clock periods and system will return to µc_on state.

7.2.2.5.11 High Pre-Wakeup and Wait Registers

Note: Refer to the note after ***Figure 26 Timing of Wake-Up Phase (on Page 90)****.*

SCCUWAITH

High Pre-Wakeup Register **Reset value: 0003_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															PREWUP

Field	Bits	Type	Description
PREWUP	1:0	rw	**Pre-Wakeup Time** Determines the number of TDMA frames signal *pre_wup* is asserted before the end of the sleep phase. The number of TDMA frames is given by **PREWUP** + 1, that is, it is selectable between 1 and 4 frames. Reset value is 4 frames.
Reserved	15:2	r	Reserved for future use; these bits must be left at their reset values.

SCCUWAITL

Wait Register **Reset value: $007F_H$**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								WAIT							

Field	Bits	Type	Description
WAIT	6:0	rw	**Used to Calculate the VCXO Wait Loop Duration** Range: 0 to 127 Resolution = $\frac{4}{32768\ \text{Hz}}$ sec The reset value (127) corresponds to the maximum loop value of 15.5 ms.
Reserved	15:7	r	Reserved for future use; these bits must be left at their reset values.

7.2.2.5.12 Hardware Wakeup Control Registers

SCCUHWWAKEUPH

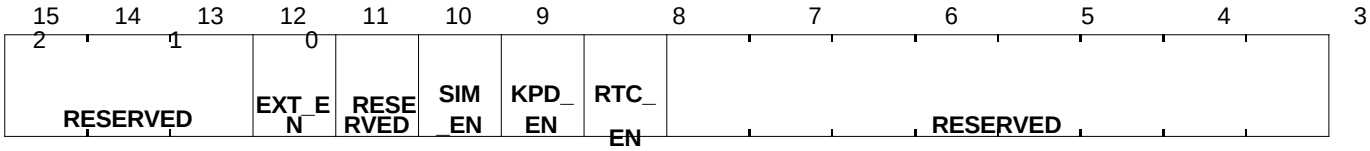
High Hardware Wakeup Control Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															ICU_EN

Field	Bits	Type	Description
ICU_EN	0	rw	Enables wake-up of SCCU by interrupt from ICU to MCU
Reserved	15:1	r	Reserved for future use; these bits must be left at their reset values.

SCCUHWWAKEUPL

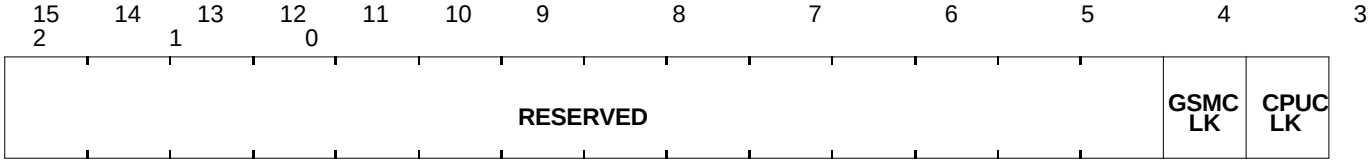
Low Hardware Wakeup Control Register **Reset value: 0000_H**



Field	Bits	Type	Description
RTC_EN	8	rw	Enables the SCCU sleep mode termination by RTC block
KPD_EN	9	rw	Enables the SCCU sleep mode termination by pressing a keypad key
SIM_EN	10	rw	Enables the SCCU sleep mode termination by SIM card insertion/removal
EXT_EN	12	rw	Enables wake-up from one of the CAPCOM or external interrupt pins
Reserved	15:13, 11, 7:0	r	Reserved; these bits must be left at their reset values.

SCCUCLKSTA

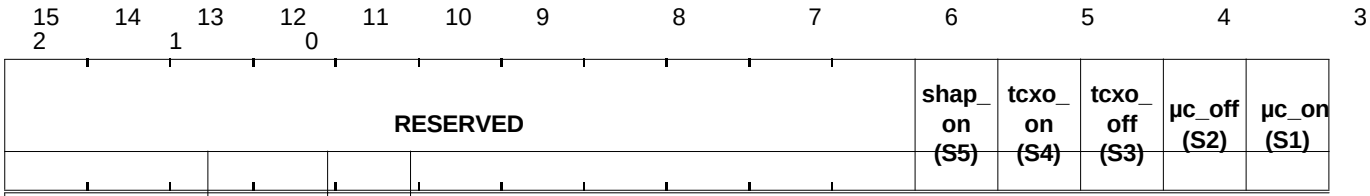
Clock Status Register **Reset value: 0003_H**



Field	Bits	Type	Description
CPUCLK	0	r	**Status of the MCU Clock** 0: Slow MCU clock 0 or 32kHz) 1: Fast MCU clock
GSMCLK	1	r	**Status of the System Interface Clock** 1: 13MHz 0: 0Hz
Reserved	15:2	r	Reserved; these bits must be left at their reset values.

SCCUSMSTA

State Machine Status Register **Reset value: 0001_H**



Field	Bits	Type	Description
µc_on (S1)	0	r	0: SCCU state machine is not in indicated state 1: SCCU state machine is in indicated state Refer to **Section 7.2.2.2 System Clock Control (on Page 86)**
µc_off (S2)	1	r	
tcxo_off (S3)	2	r	
tcxo_on (S4)	3	r	
shap_on (S5)	4	r	
Reserved	15:5	r	Reserved; these bits must be left at their reset values.

7.2.3 Efuse Management

7.2.3.1 Overview

The E-GOLDvoice uses electrical fuses for:

- HDFR Memory Redundancy (Redundant Address Registers) .
- CHIP-IDs (**ID_SNUM0**, **ID_SNUM1**, **ID_SNUM2**, **ID_SNUM3**, **ID_SNUM4**) .
- CUSTOMER_ID (**ID_SNUM6**)
- OCDS disable, External Boot disable, Efuse testmode disable (**ID_SNUM5**)
- Several chip production releated functions, e.g. RF tuning and PMU trimming.

The fuses of each module are grouped in a wrapper that controls all the Efuse functions locally.(see: **Section 7.2.3.2 Efuse Box Description and Modes**.) The two (lmu_ram and DSP DRAM) HDFR Efuse Wrappers are chained to each other. All the others are independent from each other.

All those wrappers have direct connections to Four modules:

- **Functional Block**:
  The Kernel will use the Efuse values as any kind of Register Value. (i.e.: RARs (redundant address registers for Hdfr)**CGU**:
  CGU controls the Efuse sensing during the Efuse Reset Procedure. All Efuses are automatically sensed during the reset sequence by an Efuse Reset procedure as described in **Section 7.2.3.5 CGU Efuse Initialization Procedure**).

- **Processor (C166s or TeakLite)**:
  Processor has access to those Efuse Register in Read-only or read-Write mode. Those access can be done by standard software but also via OCDS (Cerberus or OCEM).

Fuses will be usable after two operations. Efuse need first to be programmed once at IFX production line or customer Production line. The second required operation is a Efuse sensing which is done during each power-on sequence (refer to **Section 7.2.3.5 CGU Efuse Initialization Procedure**)

7.2.3.2 Efuse Box Description and Modes

7.2.3.2.1 Efuse Box Description

An EfuseBox is a N-bit non volatile data storage unit based on electrically programmable fuse elements (e-fuses). It consists of two functional modules:

1. A bank of e-fuse cells (**efuse_wrapper**)
   This Bank of Efuse cells is provided with a local control state machine (CSM) for fuse programming/sensing. This sub-unit allows permanent storage of data register states into e-fuses.
   E-fuses are either programmed (i.e. blown) and read one per each efb_clk cycle to avoid excess peak current; programming/sensing sequences are performed by the local control state machine (CSM) in combination with the pointer register P.
   The minimum number of efb_clk cycles required for fuse programming/sensing is equal to the total number of fuses (N) plus 3.
   During fuse programming, a low frequency clock signal (min. 500 ∝s cycle time) must be applied to the efb_clk input; to speed up the programming sequence, an optional blow acceleration logic block allows skipping those fuses which do not have to be blown.
2. A bank of registers (**Efuse Reg**)
   This Bank of Registers is used for temporary data storage; data can be loaded or read out either in parallel or via the serial interface.
   During data stream in, current register states are flushed out through the stream out output; after data stream out, data are automatically restored into the bank of registers.

7.2.3.2.2 Efuse Box Modes

Please find below the description of different efuse operating modes:

- **Fuse sensing:**
  fuses are sensed and their states prior to load stored states into the data registers; the sensing sequence is controlled by a local state machine. Fuse sensing mode is enabled by signal fuse_read;
- **Fuse Preset:**
  For Standard efuse boxes, the Efuse registers are preset (data_out = ‘1111”b) by efb_resn signal.
  For HDFR efuse boxes, the Efuse registers are preset (data_out = ‘1111”b) by the combination of efb_resn = ‘1’ and TM = ’1’.

Note: To allow the efuse wrapper state machine to setup, at least three standby efw clk cycles must be given in between each time two consecutive operations among those listed upon are executed.

7.2.3.3 CGU Detailed Pins and connections for Efuse Management

The CGU has to control additional pins specific to the Efuse Management.

But since we just need to do a Reset/Preset and Fuse_sensing only some of the Efuse_entity pins have to be driven by the CGU. The rest of the Efuse_entity pins are controlled by the TCU. for these pins the TCU sends some default values in functional mode or select their control via PADS in specific testmode.

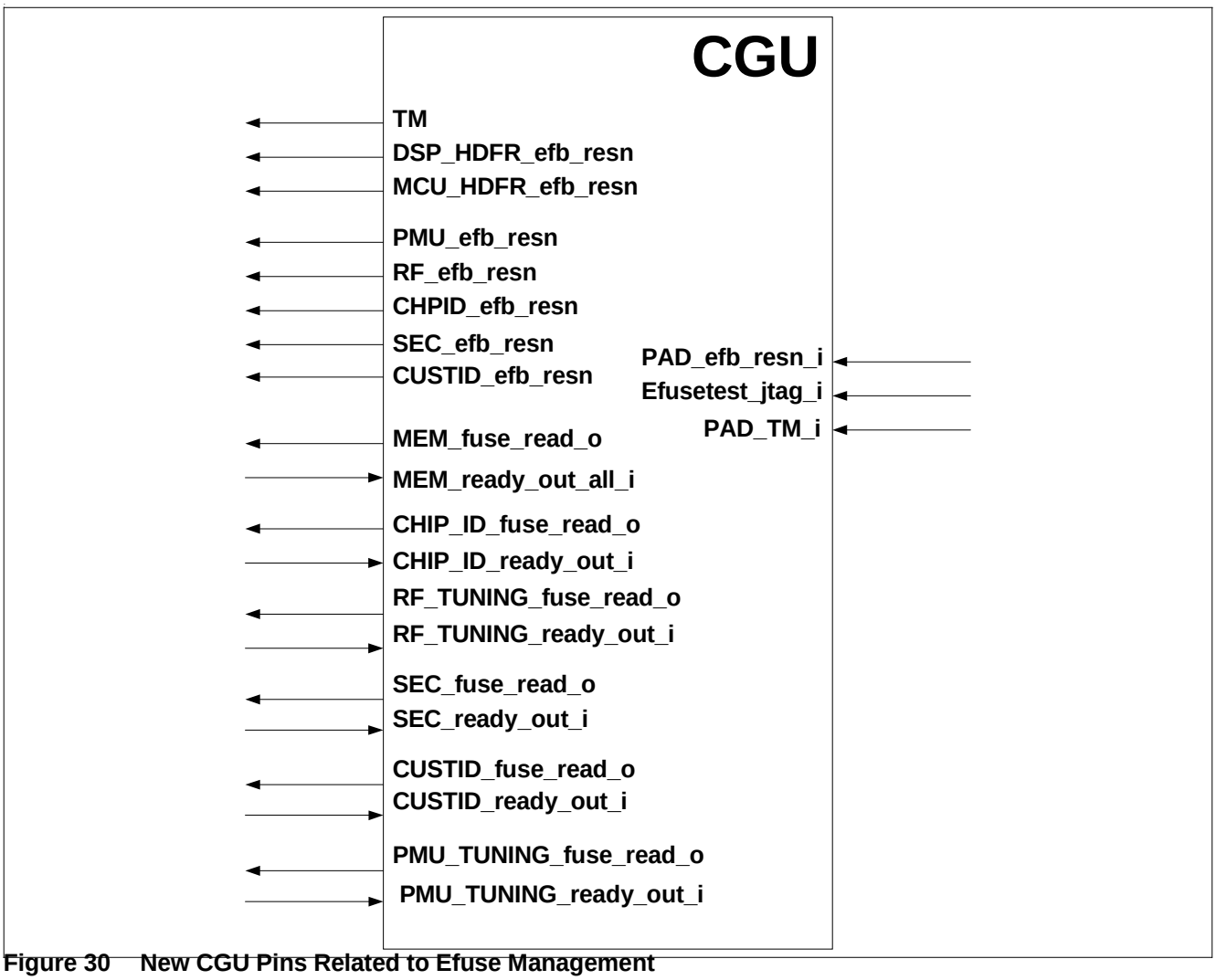


Figure 30 New CGU Pins Related to Efuse Management

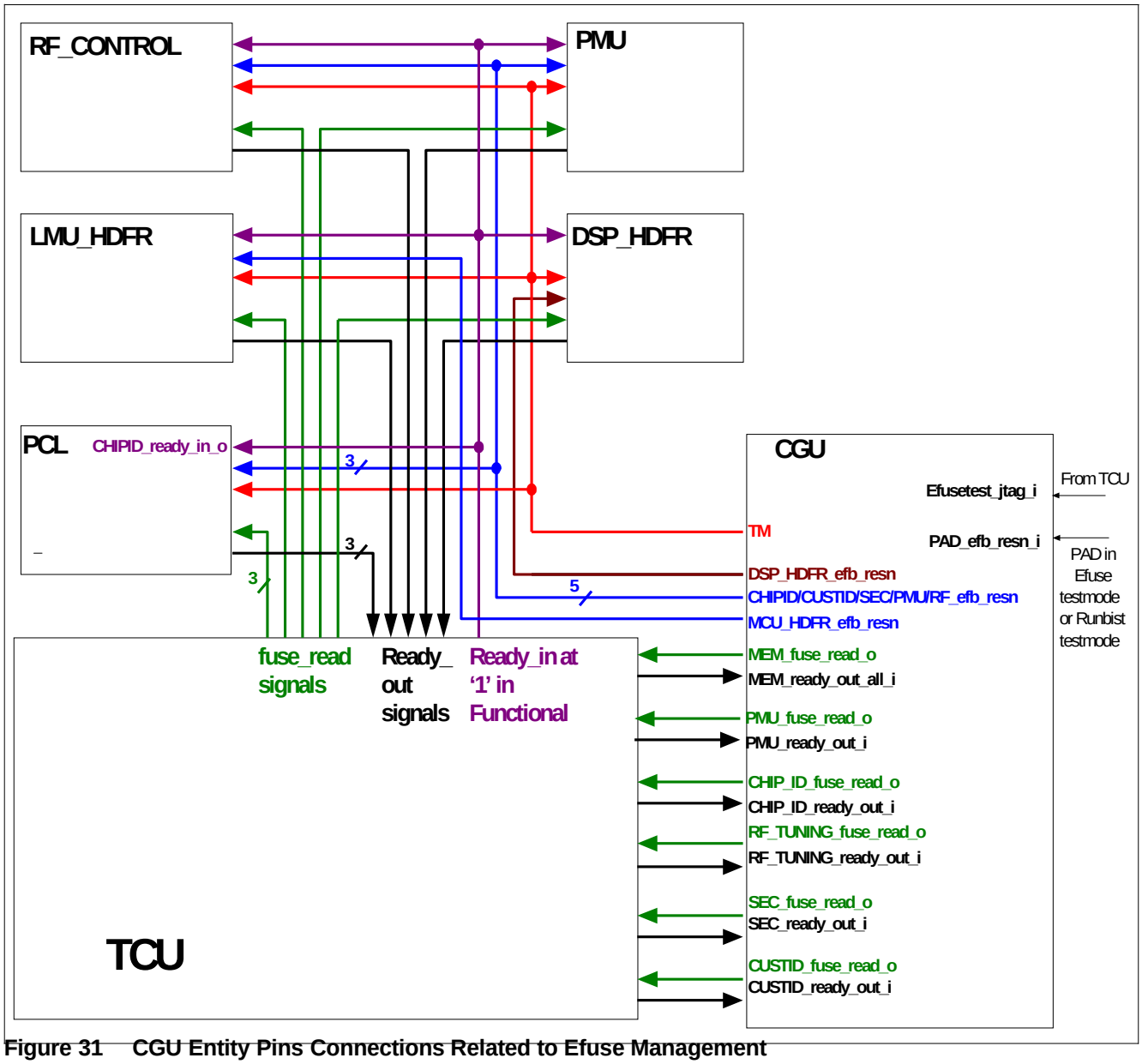


Figure 31 CGU Entity Pins Connections Related to Efuse Management

7.2.3.4 CGU Efuse Programming

Efuse are blown on the Infineon production line.

Attention: Once programming is done on an Efuse-Box, it cannot be re-programmed.

7.2.3.5 CGU Efuse Initialization Procedure

Each time the Chip is powered-on, the PMU releases the power_on_reset. This power_on_reset release is detected by the CGU which start an initialization procedure. During this procedure all the E-fuses are sensed. Sensing means that the Efuse registers receives the Efuse values.

- PMU Releases the power-on reset
- CGU Releases the HDFRs reset (MCU_HDFR_efb_resn / DSP_HDFR_efb_resn). While HDFR reset are at ‘0’, TM is at 1 to do an HDFR efuse preset.
- CGU starts a fuse sensing each efuse box one by one in the following order: All HDFRs(HDFRs are chained together) / CHIP-ID / SEC / CUSTOMER_ID . All the sensing procedure is done serially (doing sensing of two fuses in parallel is not allowed). Each Ready_out of each CHIP/Customer-ID / RF-tuning / PMU-Tuning / last HDFR is sent to the CGU which knows that it can start the next sensing.
- When the CGU receives the PMU-Tuning ready_out, then it waits during a defined time called “Band Gap Trimming Time”
- At the end of this “Band Gap Trimming Time”, the CGU will release the mcu_reset and dsp_reset.

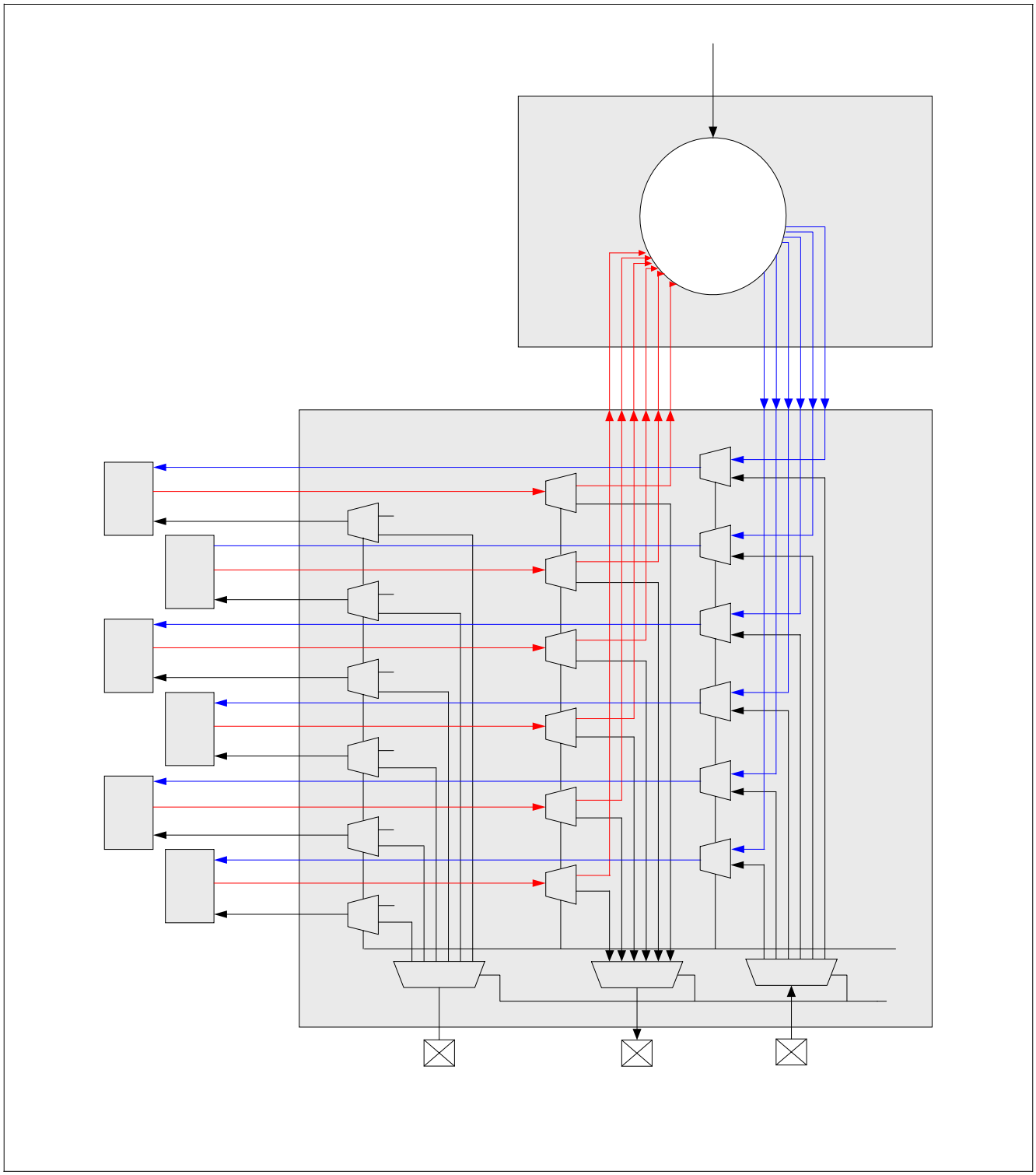


Figure 32 Top-level Connections Used for efuse Initialization Procedure

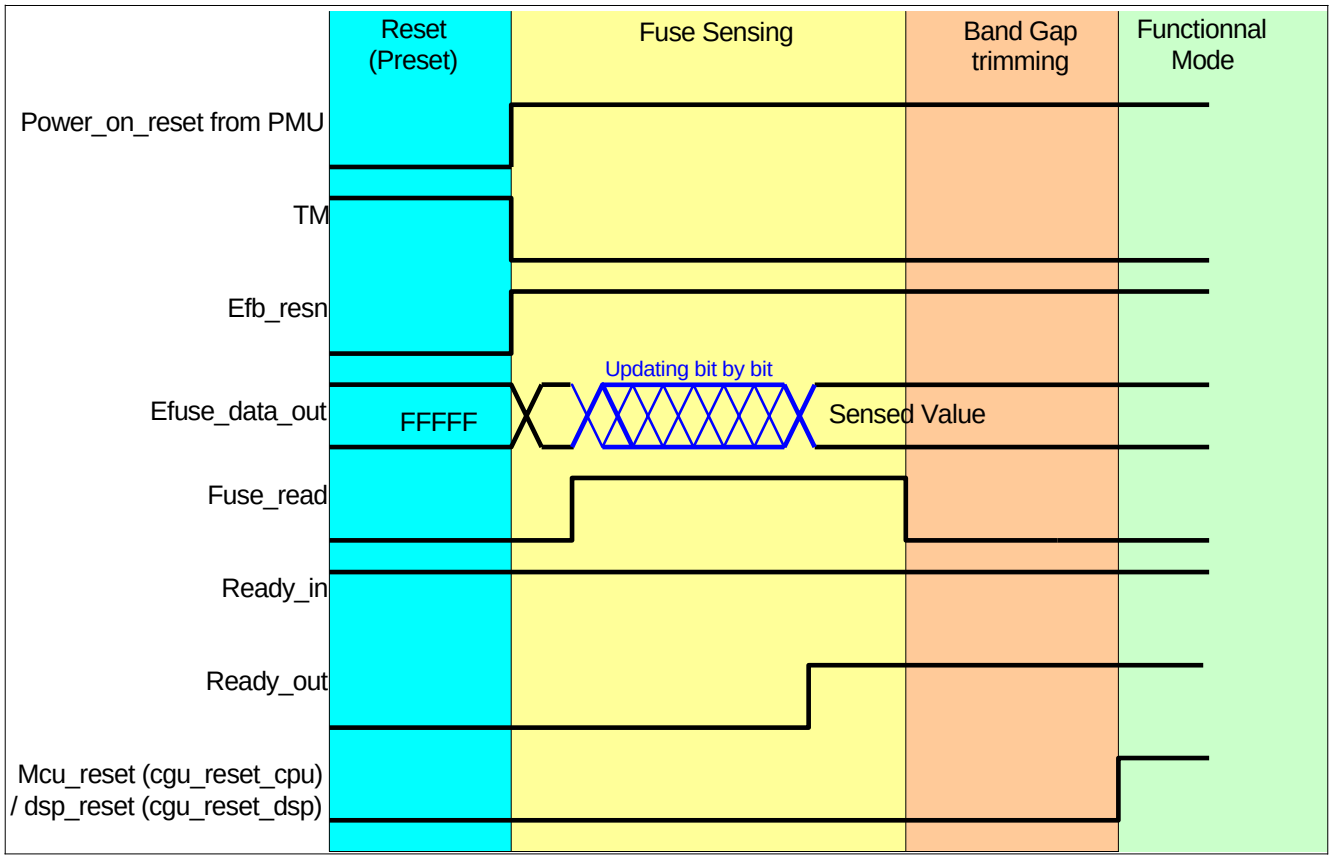


Figure 33 Functional Timing Diagram for Initialization Procedure at efuse Box Level

Notes

1. *TM pulse to do the Preset is only required by the HDFR efuse wrappers and not by all others Efuse-Box. For others Efuse Box, the TM is only used as testmode pin.*
2. *Having the Ready_in always stuck at '1' has no influence in functional mode. Then the fuse_sensing is only controlled by the Fuse_read pin.*

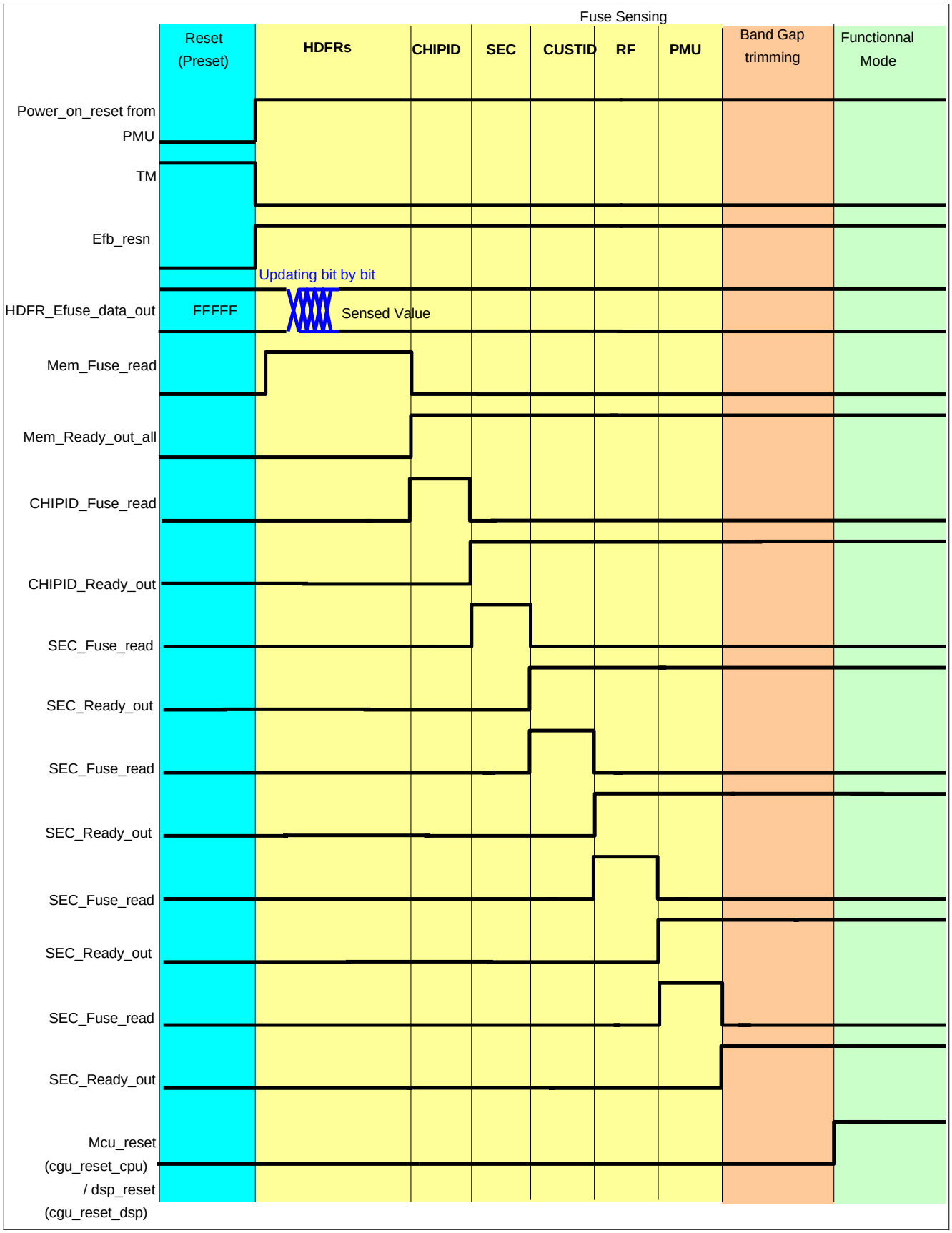


Figure 34 Functional Timing Diagram for Initialization Procedure at Top-Level

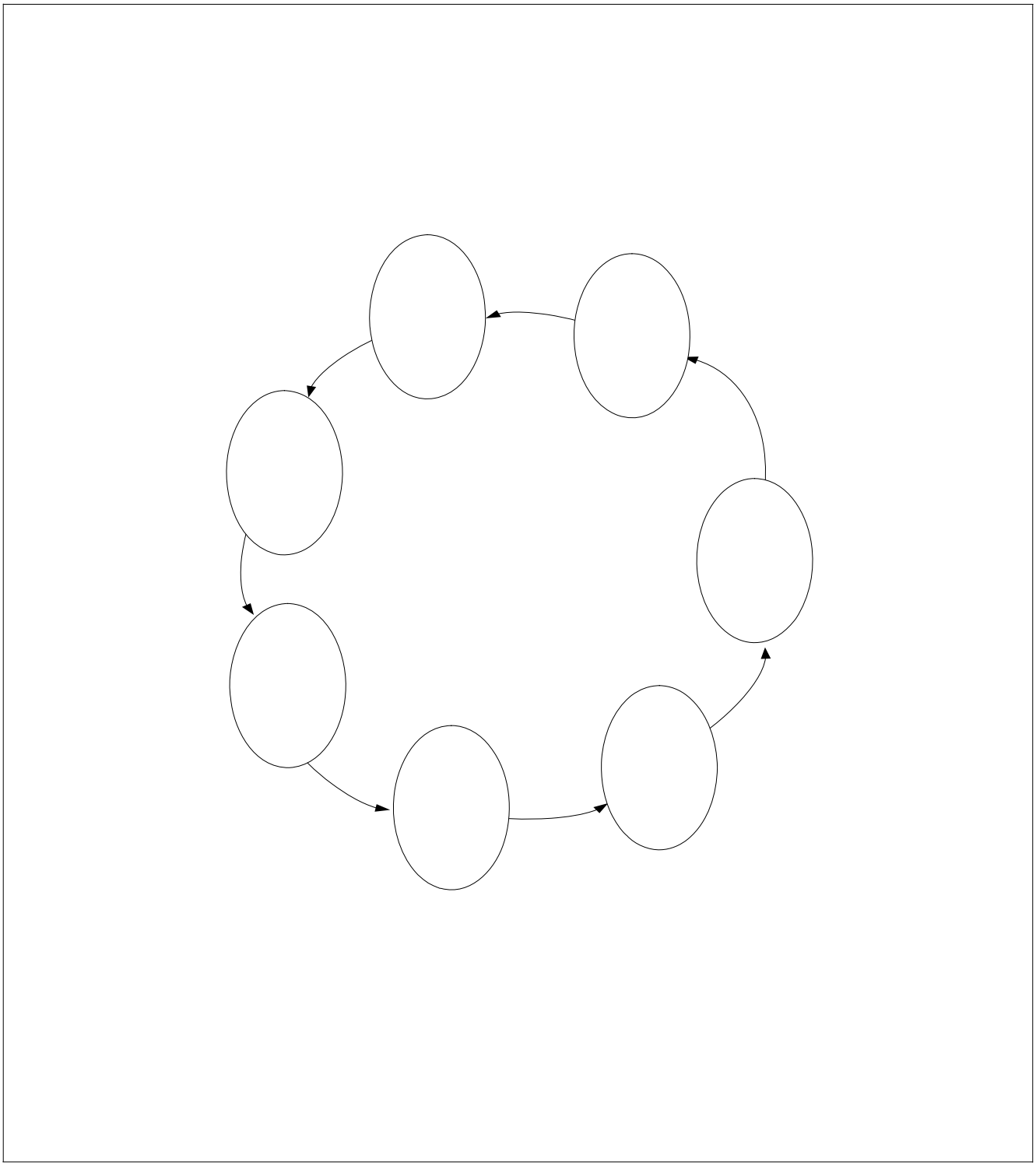


Figure 35 Initialization Efuse Sensing State Machine

Specific HDFR reset connection due to initialization Procedure:

The Preset phase of this Initialization procedure requires special reset connections on the HDFRs.

During this initialization procedure, the MBIST and functional reset have to be set active while the redundancy logics reset has to be released.

The HDFR-Efuse Box delivery is usually made of only one reset efb_resn. Then we have to separate the efb_resn going to HDFR redundancy logic from the MBIST and functional.

7.2.3.6 CGU Reset Scheme for Efuse Management

The reset behavior delivered by CGU has to be adapted depending on the following different modes:

- INITIALIZATION Procedure and Hardware reset.
- FUNCTIONNAL Mode (Software reset)

Further details regarding the power-on-reset are given in **Chapter 12 System Reset** and in **Section 8.1.2.11 Power-on and external Resets**.

Initialization Procedure and Hardware Reset Generation

Due to the Efuse initialization procedure, the global Hardware reset procedure has been modified.

Figure shows the global hardware reset sequence driven by the CGU:

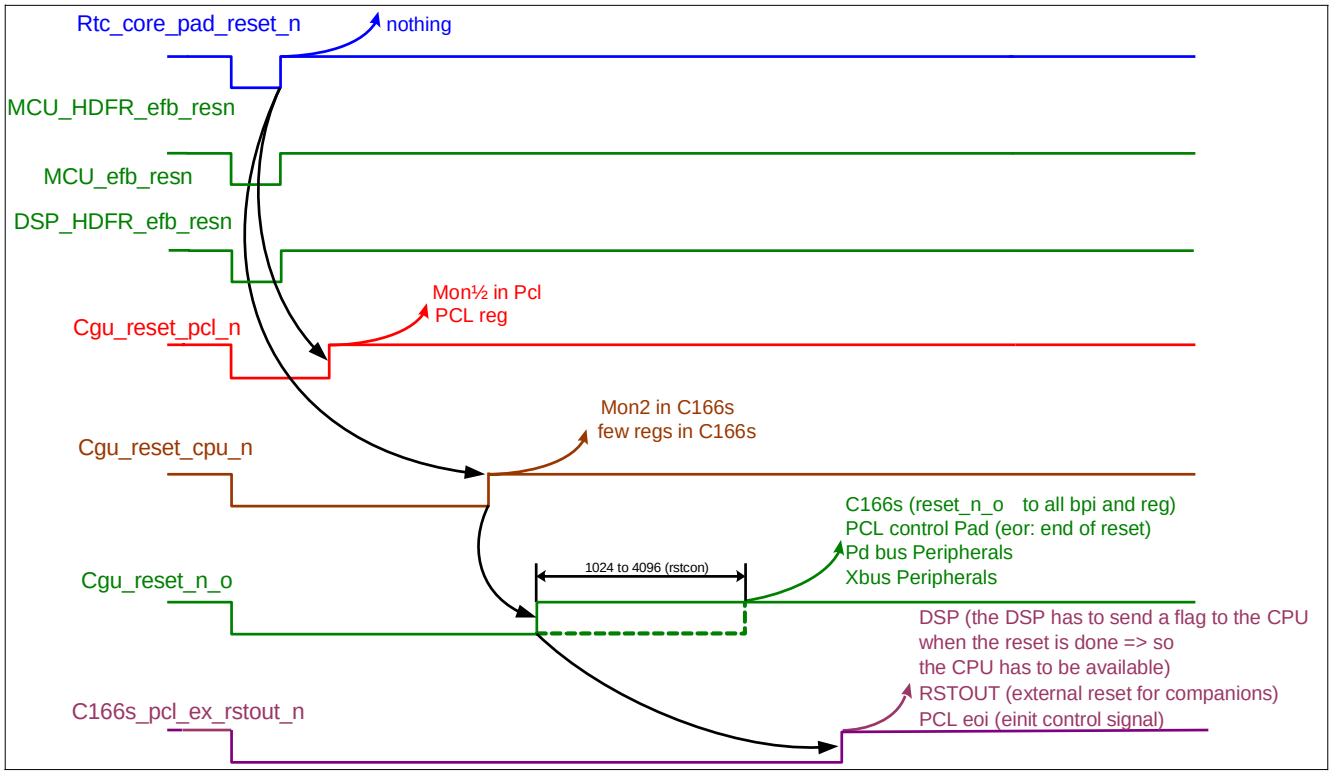


Figure 36 Global Hardware Reset Sequence

Software Reset

The CGU has also to drive specific values to efuse boxes in case of Software reset (see **Figure**).

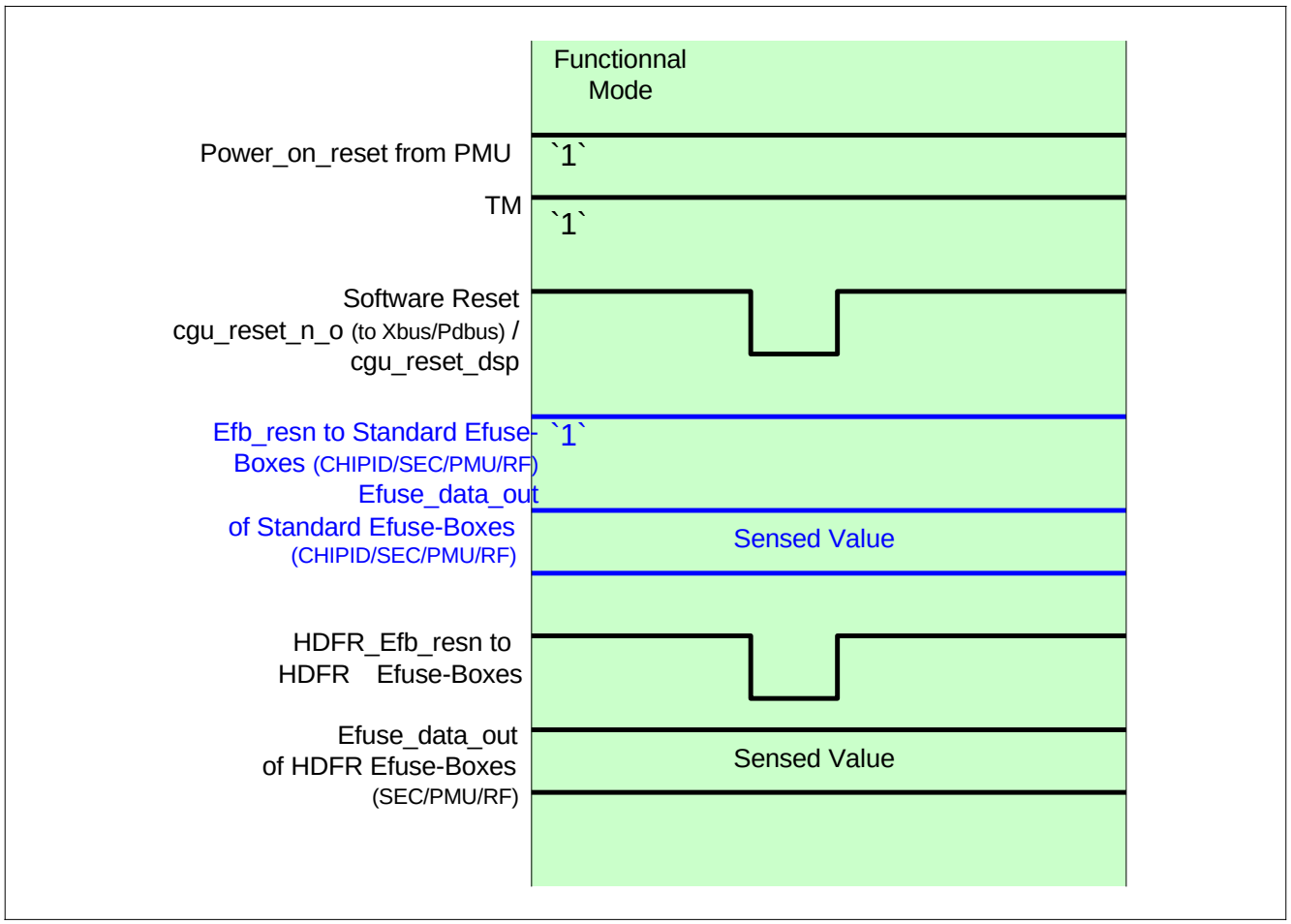


Figure 37 Software Reset on Efuses

During any software reset, the Sensed values are kept unchanged to avoid any additional fuse sensing. But The HDFR_resn signal going to HDFR has to be activated because the HDFR also contains some logic which needs to be reset. However, the Redundant register won't be reset since the preset is only done when activating TM and efb_resn at the same time (see:**Efuse Box Modes (on Page 103)**)

7.3 Measurement Interface

System Integration

- Supply domains
  - VDD_LD1 for the digital part of the measurement interface
  - VDD_LANA, for the analog part of the measurement interface
    AGND(analog domain)
- Chip internal interfaces:
  - Clock domain: Refer to **Section 7.2.1.3 Sub-System Clocks and Enables (on Page 67)** and see **Figure (on Page 71)**.
  - Bus domain: X-Bus
- Interrupt sources1 regular interrupt
- Other interfacesADCTRIG
- Chip external signals
  - M0, M2, M7,
    M8 (internally connected), M9, measurement input
  - VBB_LANA, VSSmsupply

7.3.1 Functional Overview

This peripheral is connected to the 16-bit X-Bus.

7.3.1.1 Features

Five general purpose measurement inputs are provided for measurement purposes. (In E-GOLDvoice four of those measurement inputs are mapped onto balls (M0, M2, M7, M9), 1 is connected internally for battery measurement (M8)). The remaining inputs are still in the design, but are unused. Therefore the description of the measurement interface design details remains unchanged, although only M0, M2, M7, M8, M9 are used in E-GOLDvoice for measurement purposes. M7 has an internal voltage divider by [scaled by default to 0.29]).

The measurement interface can measure the following parameter.

- Battery voltage (VBAT) is done internally by M8
- Battery technology (BTEC)
- Battery temperature (TBAT)
- Environmental temperature (TENV)
- Oscillator temperature (TVCO)
- Power of power amplifier (PWPA)
- Brightness sensor
- Accessory detector
- etc.

There are various functions for on-chip measurements:

- On chip temperature (TIC)
- Static offset of baseband transmit I and Q path (TXOF)
- Static offset of power ramp output (PAOUTOF1)
- Measurement offset self-calibration (OFS)

One separate analog-to-digital converter (ADC) is used.

The general purpose measurement inputs can be configured for voltage measurements with the following characteristics:

- Single-ended, voltage mode, referred to internal AGND
- Single-ended, current mode, referred to internal AGND
- Single-ended, voltage mode, referred to external voltage
- Single-ended, current mode, referred to external voltage drop (resistor)
- Fully differential, voltage mode.

Various measurements with different characteristics can be combined. The measurement modes are adjusted via analog switches as shown in the following sections. The switches are controlled by the Measurement Control Registers, **MEAS_CTRL1**/**MEAS_CTRL2**. The Measurement Status Register, **MEAS_STAT**, provides information about the status of the measurement interface. Data is transferred via the Measurement Data Registers, **MEAS_DATAx**. All registers are part of the measurement interface X-Bus interface (refer to **Section 7.3.9 (on page 124)**).

7.3.2 Register Overview

All registers are accessed as 32 bit registers. Unused or reserved bits have to be set to zero. The register addresses are defined in **Section 10.2 X-Bus Register Addresses (on Page 489)**.

Table 19 Measurement Interface Register List

Register	Register Name
MEAS_CTRL1	Low Measurement Control Register
MEAS_CTRL2	High Measurement Control Register
MEAS_STAT	Measurement Status Register
MEAS_DATA0	Measurement Data Register 0
MEAS_DATA1	Measurement Data Register 1
MEAS_DATA2	Measurement Data Register 2
MEAS_DATA3	Measurement Data Register 3
MEAS_DATA4	Measurement Data Register 4
MEAS_DATA5	Measurement Data Register 5
MEAS_DATA6	Measurement Data Register 6
MEAS_DATA7	Measurement Data Register 7
MEAS_CLK	Measurement Peripheral Clock Control Register
MEAS_ID	Module Identification Register

7.3.3 Measurement Circuit and Mode Setting

For the conversion of all measurement signals like battery voltage, battery technology code, temperature, etc. identical measurement inputs M0, M2, M9 can be used (M7 has the internal voltage divider, M8 is internally connected for battery voltage measurement). For self-calibration purposes several additional chip internal inputs are available. **Figure 38** shows the principle of the measurement circuits consisting of external sensing components (recommendation/example) and integrated analog multiplexers and switches. **Table 20 Measurement Switch Settings (on Page 116)** shows the corresponding switch settings.

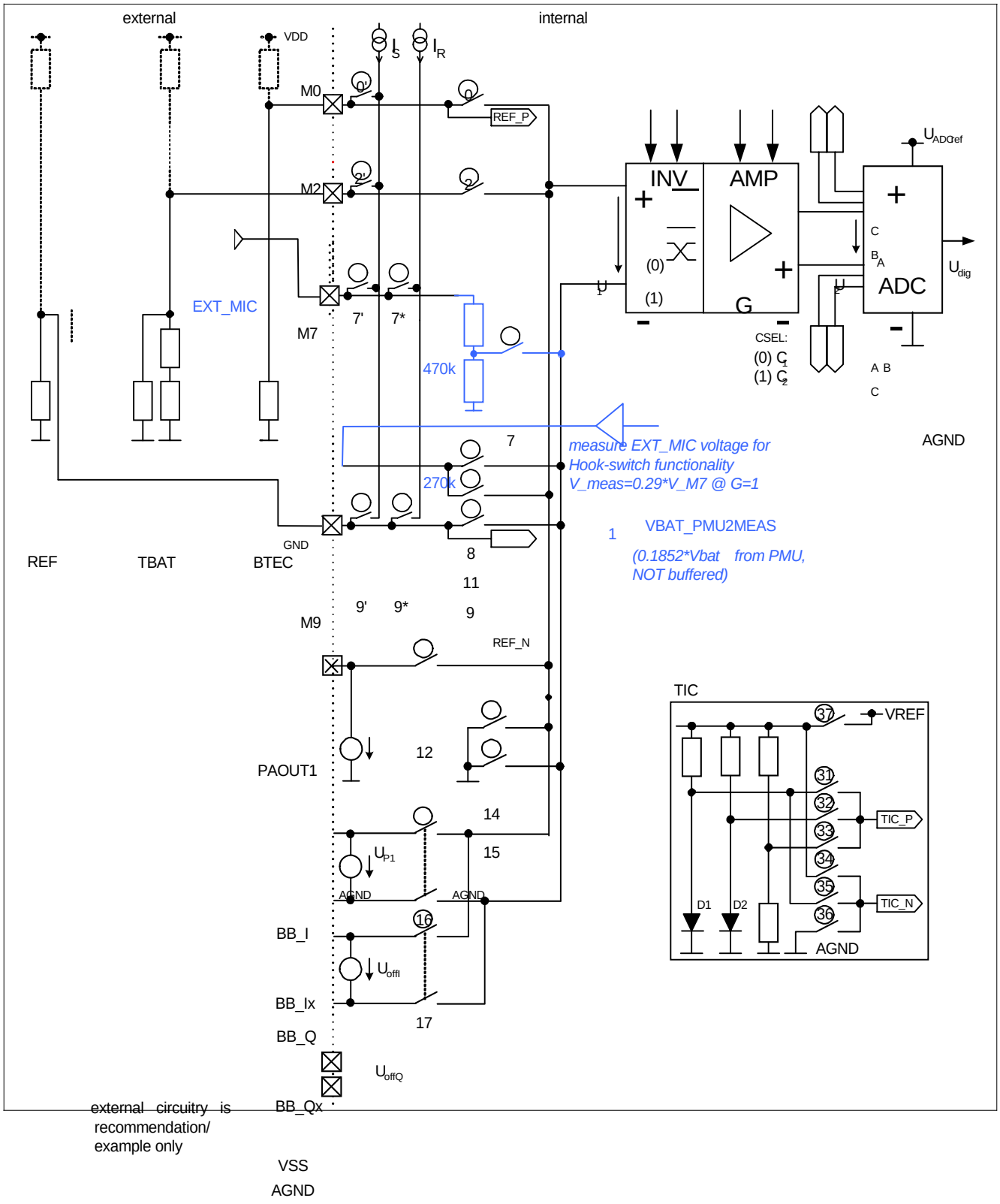


Figure 38 Measurement Circuit and Mode Setting

The measurement modes and switch settings defined in **Table 20** are controlled via the Measurement Control Registers (**MEAS_CTRL1** and **MEAS_CTRL2**). The dashed lines in **Figure 38** indicate the additional external components that have to be used if the internal current source is not used. On chip temperature measurement TIC is described later (refer to **Section 7.3.6 (on page 121)**).

Table 20 Measurement Switch Settings

Mode		Switches Closed[1)2)]			Description
M0A	01_H	0, 0', 15	0.5	A	GP[3)] referred to internal AGND
M2A	03_H	2, 2', 15	0.5	A	
M7B	08_H	7, 7', 14	1.0	A	
M8B	09_H	8, 8', 14	1.0	A	
M9B	$0A_H$	9, 9', 14	1.0	A	
M0M9A	$0C_H$	0, 0', 9, 9*	0.5	A	GP referred to external voltage or GP differential
M2M9A	$0E_H$	2, 2', 9, 9*	0.5	A	
M7M8A	13_H	7, 7', 11, 8*	0.5	A	
M2M7A	15_H	2, 2', 7, 7*	0.5	A	
M0M9B	18_H	0, 0', 9, 9*	1.0	A	
M2M9B	$1A_H$	2, 2', 9, 9*	1.0	A	
M7M8B	20_H	7, 7', 11, 8*	1.0	A	
M2M7B	21_H	2, 2', 7, 7*	1.0	A	
PAOUTOF1	24_H	12, 15	1.0	A	Offset of power ramping output PAOUT1
TXOFI	26_H	16	4.0	A	Offset TX path I component
TXOFQ	27_H	17	4.0	A	Offset TX path Q component
ADCCAL	$2A_H$	-	-	B	ADC gain calibration mode
TICD1	$2B_H$	31, 36, 37	-	C	On chip temperature: $U_{BE,D1}$
TICD2	$2C_H$	32, 36, 37	-	C	On chip temperature: $U_{BE,D2}$
TICR1	$2D_H$	31, 34, 37	-	C	On chip temperature: $-U_{R41}$
TICR2	$2E_H$	32, 34, 37	-	C	On chip temperature: $-U_{R42}$
TICDIFF	$2F_H$	32, 35, 37	-	C	On chip temperature: $U_{BE,D2} - U_{BE,D1}$
TICREF	30_H	33, 36, 37	-	C	On chip temperature calibration: VREF/2
OFF	00_H	-	-	-	Power save

1) A given value for the MX bit field sets the mode. This closes the switches and sets the gain and ADC inputs to the values listed on the same line as the mode.

2) Switches not mentioned are open. In the power save mode (OFF) all switches are open.

3) General purpose

Table 21 General Pre-Amplifier Settings

Bit Name	Value	Description
MEAS_CTRL2.CSEL	0	Slow settling of pre-amplifier (C1)[1)]
	1	Fast settling of pre-amplifier (C2)
MEAS_CTRL2.INV	0	No inversion of pre-amp. input voltage
	1	Inversion of pre-amplifier input voltage[2)]

1) It is recommended to set **CSEL** = 0 for all measurement modes, except PWPA, to ensure 12-bit resolution of the ADC.

2) Voltages measured with **INV** = 1 are marked with an asterisk (*) in this document section.

Table 22 Resistors for Measurement Circuits

Resistor/Capacitor	Value[1)] (typical)
R10	user defined
R11	user defined
R12	user defined
R13	user defined
R14	user defined
R15	user defined
R16	user defined
R17	user defined
R_{NTC}	user defined
R_{CODE}	user defined
R41	500k
R42	50k
R43	50k
R44	50k

1) Resistor values R41-R44 are typical values. These values may differ by R = $R_{typ.}$ * (1 ± 25% process tolerances ± 10% temperature coefficient).

7.3.4 Differential General Purpose Measurements (MxMy)

7.3.4.1 Equivalent Network for Differential Measurement Circuit

The electrical behavior of the differential measurement circuit for the modes MxMyA and MxMyB (that is, VR-VS) can be modelled by the equivalent network according to **Figure 39** containing an ideal amplifier with gain G. The parameters R_{cm}, R_x, U_{cmi} and G are individual for each measurement mode but do not depend on the external circuitry. The user dependent external circuitry is also replaced by equivalent voltage sources U_{ref} and U_{sig} with internal resistances R_{ref} and R_{sig} respectively. Refer to **Section 7.3.4.2 Equivalent Network for Reference Voltage Generation (on Page 118)** and **Section 7.3.4.3 Equivalent Network for Signal Input (on Page 119)**.

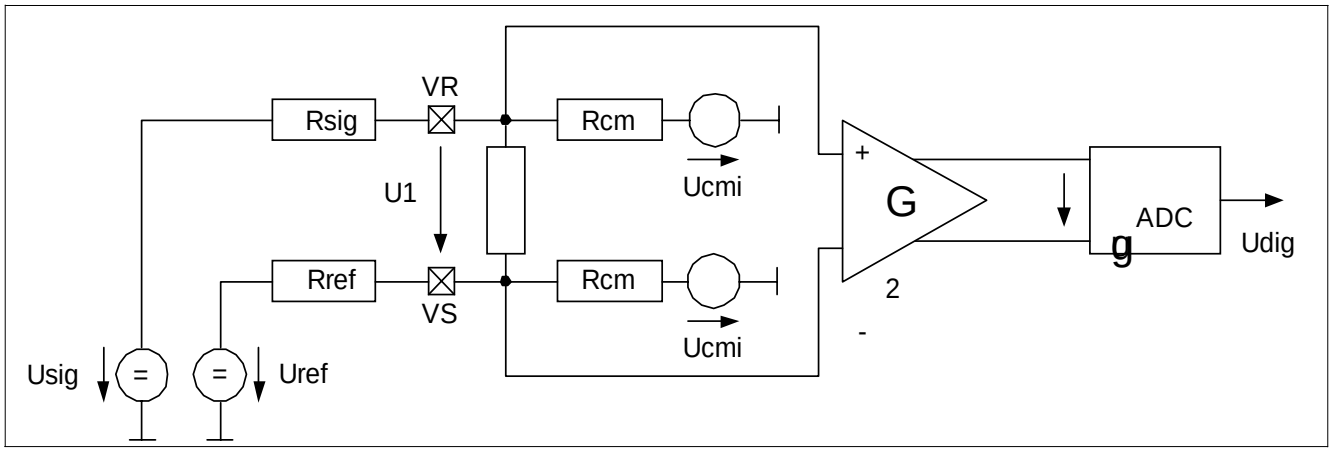


Figure 39 Equivalent Network for Differential Measurement Circuit

The input voltage U_2 of the analog-to-digital converter can be calculated according to

$$U_2 = G \cdot \left(U_{cmi} \cdot \left(\frac{1}{R_{sig}} - \frac{1}{R_{cm}} \right) + \frac{U_{sig}}{R_{sig} + R_{cm}} - \frac{U_{ref}}{R_{ref} + R_{cm}} \right) \Bigg/ \left(\frac{1}{R_{sig}} + \frac{1}{R_{ref}} + \frac{1}{R_{cm}} \right) \tag{9}$$

and the output of the ADC results in

$$U_{dig} = U_2 \cdot g_{ADC} + U_{dig,0}, \tag{10}$$

$$\text{with } g_{ADC} = 2048 \text{ LSB} \cdot \frac{1}{0.8333 \cdot V_{REF}} = 2048 \text{ LSB/V typ.}, \tag{11}$$

$$U_{dig,0} = 2048 \text{ LSB typ.} \tag{12}$$

$$\text{and } V_{REF} = 1.2 \text{ V typ.}. \tag{13}$$

Note: The reference voltage of the ADC ($0.8333 \cdot V_{REF}$) shows the same tolerance as VREF.

The measurement modes M0M9A/B, M1M8A/B and M2M7A/B are especially suited for measurements of true differential voltages. If the internal resistance of the differential source is negligible the input voltage of the ADC in these modes is given by

$$U_2 = G \cdot (U_{Mx} - U_{My}) \tag{14}$$

Note: In addition to the condition for the input voltage, the conditions for differential and common mode voltage have to be satisfied.

Note: For offset elimination of offsets caused by the measurement path itself the measurement can be repeated with **MEAS_CTRL1.*INV*** *= 1 and the evaluation method as described in* ***Section 7.3.8 (on page 124)*** *can be used.*

7.3.4.2 Equivalent Network for Reference Voltage Generation

According to the example for the external circuitry (REF) in **Figure 38 (on page 115)** the external reference voltage generation can be replaced by a voltage source as shown in **Figure 40**.

For the pins Mx and My an alternative measurement mode exists which can be enabled via the control bits **MEAS_CTRL1.TC**. In this case the internal current sources are switched on and the difference of the voltage which drops over the external resistors is measured at the differential input. With **TC** the current source IR = 60 µA is switched on and for IS different currents can be selected.

Note: IR and IS are proportional to VREF (IR = G_{IR} VREF, IS = G_{IS} VREF).

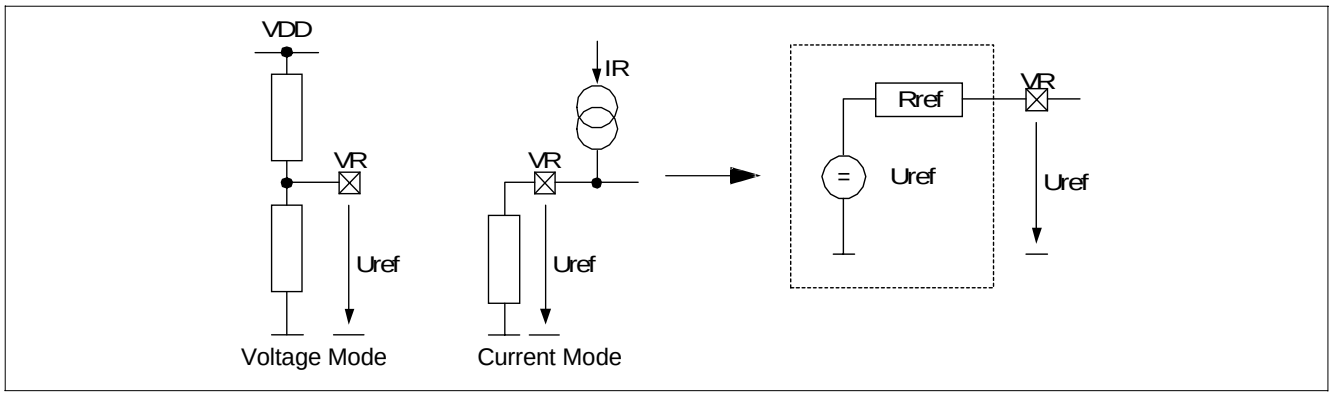


Figure 40 Equivalent Network for Reference Voltage Generation

The equivalent parameters in voltage mode are

$$U_{ref} = VDD \cdot \frac{R_2}{R_1 + R_2} \quad (15)$$

and $$R_{ref} = \frac{R_1 \cdot R_2}{R_1 + R_2} \quad (16)$$

and in current mode they are

$$U_{ref} = IR \cdot R_3 \quad \text{with} \quad IR = 60\ \mu A \quad \text{typ.} \quad (17)$$

and $$R_{ref} = R_3 \quad . \quad (18)$$

7.3.4.3 Equivalent Network for Signal Input

According to the example for the external circuitry (TVCO, TENV, TBAT) in **Figure 38 (on page 115)** the signal input can also be replaced by a voltage source as shown in **Figure 41**.

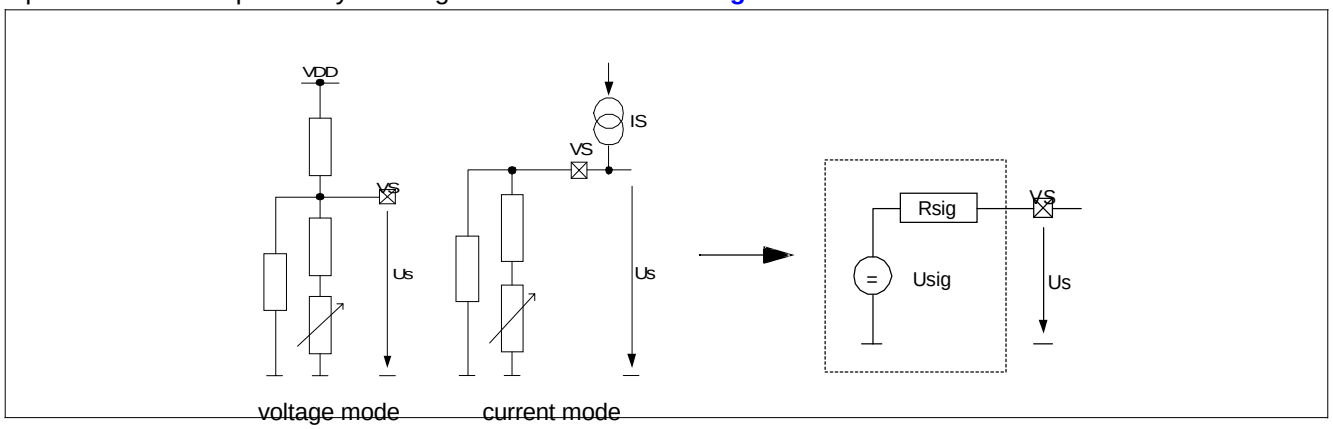


Figure 41 Equivalent Network for Measurement Signal Input

The equivalent parameters in voltage mode are

$$U_{sig} = \frac{VDD}{\left(\frac{1}{R_V} + \frac{1}{R_P}\right) \cdot R_V + R_S + R_{ntc}} \quad (19)$$

and

$$R_{sig} = \left(\frac{1}{R_V} + \frac{1}{R_P + R_S + R_{ntc}}\right)^{-1} \quad (20)$$

In current mode the following equations apply:

$$U_{sig} = \frac{IS}{\frac{1}{R_P} + \frac{1}{R_S + R_{ntc}}} \quad \text{with} \quad IS = I_{TC[2:0]} \text{ typ.} \quad (21)$$

and

$$R_{sig} = \left(\frac{1}{R_P} + \frac{1}{R_S + R_{ntc}}\right)^{-1} \quad (22)$$

7.3.5 Single-Ended General Purpose Measurements (Mx)

7.3.5.1 Equivalent Network for Measurement Circuit

The electrical behavior of the measurement circuit for the input modes Mx (i.e. VS) in voltage mode can be modelled by the equivalent network according to **Figure 42** containing a resistor, a voltage source and an ideal amplifier with gain G. The parameters R_{eq}, U_{eq} and G are individual for each measurement mode but not dependent on the external circuitry.

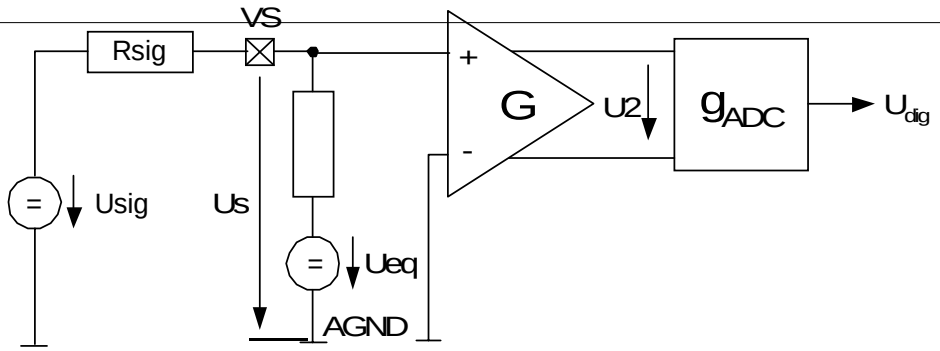


Figure 42 Equivalent Network for Single-Ended Measurement Mx

The input voltage U_2 of the analog-to-digital converter can be calculated according

$$U_2 = \frac{U_{sig} \cdot R_{sig} + U_{eq} \cdot R_{eq}}{R_{sig} + R_{eq}} \quad (23)$$

and the output of the ADC according **Equation (10) (on Page 118)**.

7.3.5.2 Equivalent Network for Signal Input

According to the example for the external circuitry (VBAT, BTEC, PWPA) in **Figure 38 (on page 115)** the signal input can also be replaced by a voltage source as shown in **Figure 43**. For the single-ended measurements an alternative measurement mode exists which can be also enabled via the control bits **MEAS_CTRL1.TC**. In this case an internal current source is switched on and the voltage drop over an external resistor is measured at the input Mx. With **TC** different IS currents can be selected.

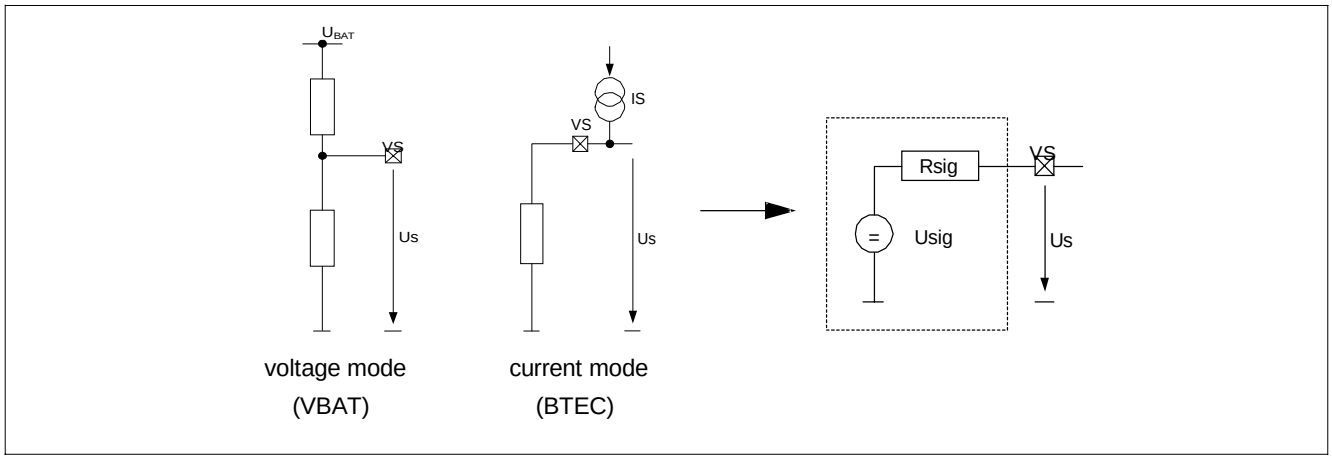


Figure 43 Equivalent Network for Measurement Signal Input

The equivalent parameters in voltage (for example, VBAT) mode are:

$$U_{sig} = \frac{U_{BAT}}{R_{13} + R_{14}} \quad (24)$$

$$\text{and } R_{sig} = \left(\frac{1}{R_{13}} + \frac{1}{R_{14}} \right)^{-1} \quad (25)$$

In current mode (for example, BTEC) the following equations apply:

$$U_{sig} = IS \cdot R_{CODE} \quad \text{with} \quad IS = I_{TC[2:0]} \quad \text{typ.} \quad (26)$$

$$\text{and } R_{sig} = R_{CODE}. \quad (27)$$

Note: IS is proportional to VREF (IS = G_{IS} VREF).

7.3.6 On-Chip Temperature Measurement

The temperature sensor for measurement of the on chip temperature (TIC) is placed in the mixed signal section of the chip in the neighborhood of the controller block. The temperature characteristic of a well-defined semiconductor junction at a defined current level is used for determining the on chip temperature.

For the measurement modes TICDx and TICRx the output of the ADC is given by

$$U_{dig,\,TICDx} = g_{ADC} \cdot \frac{R_{Dx}(I_{Dx})}{R_{Dx}(I_{Dx}) + R_{4x}} \cdot VREF + U_{dig,0} \quad (28)$$

$$U_{dig,\,TICRx} = -g_{ADC} \cdot \frac{R_{3x}}{R_{Dx}(I_{Dx}) + R_{4x}} \cdot VREF + U_{dig,0} \quad (29)$$

where $R_{Dx}(I_{Dx})$ is the actual temperature sensitive resistance of the diode Dx with current I_{Dx} and g_{ADC} is the gain of the ADC according to **Equation (44) (on Page 124)**.

To avoid expensive temperature calibration processes a four-step measurement method can be carried out. For the first two measurements the modes TICD1 and TICD2 are used and to determine the current ratio of the two diodes D1 and D2. The absolute temperature in degrees Kelvin can be calculated by

$$K = \frac{q}{k} \cdot \left[\ln\left[\frac{R_{41}}{R_{14}} \cdot \frac{U_{dig,\,TICR2} - U_{dig,0}}{U_{dig,\,TICD2} - U_{dig,\,TICD1}} \right] \right]^{-1}$$

(30)

$$42 \quad U_{dig,\,TICR1} - U_{dig,0} \qquad g_{ADC}$$

where $q = 1.602177 \cdot 10^{-19}$ As, $k =$ $1.380658 \cdot 10^{-23}$ J/K (31)

Note: The area ratio of the diodes is $A_{D1}/A_{D2} = 14$.

To achieve the specified accuracy in **Table 185 On Chip Temperature Measurement TIC (on Page 550)**, a factory calibration of the ADC gain g_{ADC} has to be carried out before temperature measurements using the measurement mode ADCCAL. Refer to **Section 7.3.8.1 Gain Calibration of Measurement ADC (on Page 124)** for further details.

7.3.7 Modulator Unit Offset Measurement TXOFI and TXOFQ

The input voltage of the ADC in mode TXOFI and TXOFQ is given by

$$U_2 = U_{offI/Q} \cdot G \quad (32)$$

If the bit INV is set HIGH (this is indicated by an asterisk) then the input voltage of the ADC in mode TXOFI* and TXOFQ* is given by

$$U_2^* = G \cdot U_{offI/Q} . \quad (33)$$

To eliminate effects caused by offsets in the measurement interface the offsets in the transmit paths can be calculated using

$$U_{offI/Q} = \frac{1}{G} \cdot \frac{1}{2} \cdot \left(U_2 - U_2^* \right) \quad (34)$$

and

$$U_{offI/Q} = -\frac{1}{G} \cdot \frac{1}{2} \cdot \left(U_{dig,TXOFI/Q} - U_{dig,TXOFI/Q}^* \right) \cdot g_{ADC} \quad (35)$$

Note: This offset elimination method can be used for all differential measurements.

7.3.7.1 PA Control Hardware Offset Measurement PAOUTOF1

For PAOUT1 outputs of the PA Control Hardware the offset can be determined. The output of the measurement ADC is

$$U_{dig,PAOUTOF} = G_{ADC} \cdot U_P + U_{dig,0} \cdot g \quad (36)$$

and the PAOUT voltage is

$$U_P = \frac{U_{dig,PAOUTOF} - U_{dig,0}}{g_{ADC} \cdot G} . \quad (37)$$

*Note: For measurement accuracy enhancement, a calibration of g_{ADC} using the mode ADCCAL according to **Equation (44) (on Page 124)** and an offset correction according **Section 7.3.7 Modulator Unit Offset Measurement TXOFI and TXOFQ (on Page 122)** is recommended.*

To determine the offset of the PAOUT signal two measurements at different PAOUT voltages U_{Pa} and U_{Pb} should be carried out according **Figure 44 (on page 123)**. Then, the offset voltage

$$U_{offP} = U_{Pa} - \frac{U_{Pb} - U_{Pa}}{PARx(16)_b - PARx(16)_a} \cdot \left(PARx(16)_a \right) \quad (38)$$

can be calculated by solving a linear equation.

*Note: The PAOUT voltage U_P can be adjusted in the GSM System Interface by programming a specific power ramp (ramp up sequence) with the desired final output voltage at the PAOUT pin defined by PARx(16) (refer to **Section 7.8.2 RF RAM (on Page 170)**). Set PAINCx to 000_H. Additionally, the fields*

***ANA_CTRL2*.PA_CAL1** *have to be set to 00_H prior to any PAOUTOF measurement.*

For determining U_{offP} set the ***ANA_CTRL1*.PA_OFF1** flag to 0. If the result shows $U_{offP} > U_{sat-}$ then determine U_{offP} a second time with **PA_OFF1** = 1. In this case it should be $U_{offP} \delta U_{sat-}$. For usual PAOUT operation use the **PA_OFF1** value where $U_{offP} \delta U_{sat-}$.

Finally, the calibration value is given by

$$PA\_CAL\ (PARx(16)_a) = \frac{(PARx(16)_b - PARx(16)_a) \cdot U_{drop-}}{U_{Pb} - U_{Pa}} + \frac{}{1.25\ mV/LSB} \quad (39)$$

The calibration field ***ANA_CTRL2*.PA_CAL1** can now be programmed with the value according to **Equation (39)** to avoid saturation of the output amplifier for the pin PAOUT and to avoid the need of device specific power ramp sequences (only concerning any offset effects).

Due to this offset calibration the output voltage at the pin PAOUT for PARx(16) = 00_H is defined and is $U_{PAOUT} = U_{drop-}$ which is within the linear operating range of the output buffer. This is the precondition for transient-free power ramps.

Note: During final offset measurement the **ANA_CTRL1*.PA_OFF1*** *flag has to be set in the same way as it is set during usual PAOUT operation to get correct measurement results and a valid* ***PA_CAL1*** *value.*

Note: Setting ***PA_OFF1*** *= 1 during normal operation without any digital offset correction causes saturation effects.*

Note: Due to quantization errors the accuracy of the ***PA_CAL1*** *value depends on the proper selection of $PARx(16)_1$ and $PARx(16)_2$.*

Note: It has to be ensured that the power amplifier does not output any power for approximately $U_{PAOUT} \delta U_{drop-} + 10\ mV$. For the design of the power ramp sequences the defined minimum output voltage of U_{drop-} has to be considered (refer to **Table 184 Specification of PAOUTOF1 (on Page 550)***).*

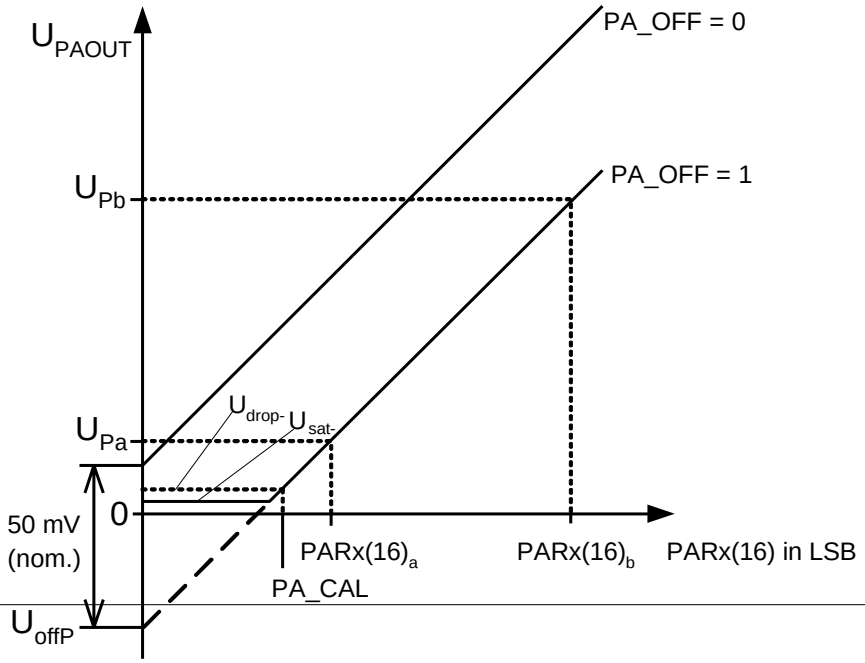


Figure 44 PAOUT Offset Measurement

7.3.8 Offset Calibration of Measurement Interface

For each measurement path the offset introduced by the pre-amplifier and the ADC itself can be determined and eliminated if the measurement is carried out twice, the first time with **MEAS_CTRL1.INV** = 0 and the second time with **INV** = 1. In this case a simple offset model with one offset source U_{off} comprising the offsets of the pre-amplifier and the ADC is assumed as shown in **Figure 45**. Measurements with **INV** = 1 are indicated by an asterisk (*).

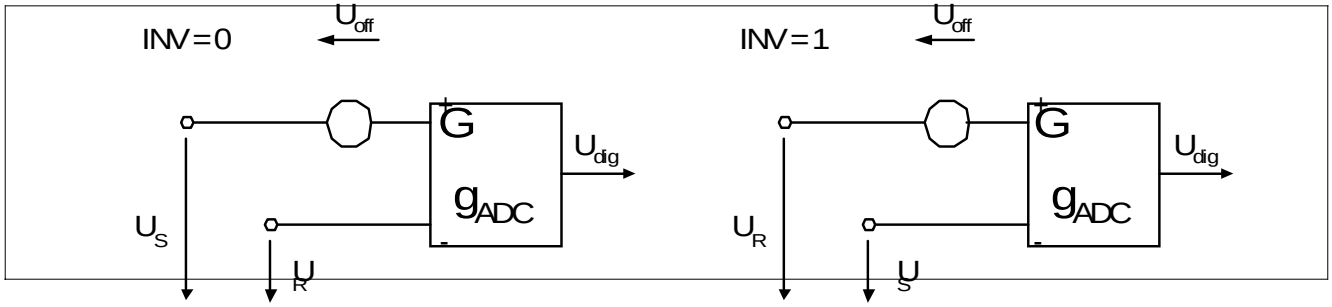


Figure 45 Equivalent Network for Offset Calculation

The output voltage of the ADC with **INV** = 0 is given by

$$U_{dig} = g_{ADC} \cdot G \cdot (U_S - U_R + U_{off}) \quad (40)$$

and with **INV** = 1 by

$$U^*_{dig} = g_{ADC} \cdot G \cdot (U_R - U_S + U_{off}) \quad (41)$$

U_{off} can easily be eliminated by calculating

$$U_{dig}' = U_{dig} - U^*_{dig} \quad (42)$$

and the digital correction value $U_{dig,off}$ which has to be subtracted from the measurement result can be calculated by

$$U_{dig,off} = (U^*_{dig} - U_{dig}) \quad (43)$$

7.3.8.1 Gain Calibration of Measurement ADC

In gain self-calibration mode ADCCAL a known voltage is applied to the pins M0 and M9 and the ADC gain can be calculated by

$$g_{ADC} = \frac{U_{dig,ADCCAL} - U_{dig,0}}{U_{M0} - U_{M9}} \cdot \quad (44)$$

If the pins M0 and M9 are shortened the actual offset $U_{dig,0}$ (refer to **Equation (10) (on Page 118)**) of the ADC can be determined, too.

Note: During measurement mode ADCCAL the electrical characteristics of the pins M0 and M1 are different to the characteristics during measurement mode M0 and M1!

Note: This calibration method can be used to enhance measurement accuracy of other measurement modes. However, with ADCCAL only the gain of the ADC can be calibrated and not the gain of the pre-amplifier.

7.3.9 BPI Bus Register Descriptions

All clocks necessary for the measurement interface are derived from the master clock clk_bus.

The register addresses are in **Section 136 Address Mapping of X-Bus Peripherals (on Page 489)**.

7.3.9.1 Measurement Control Registers

These registers control the measurement operating modes and the data transfer.

MEAS_CTRL1

Measurement Control Register 1 **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			TC			CSEL	RESERVED						INV	MX		

Field	Bits	Type	Description
MX	5:0	rw	Measurement mode 00_H Off (Default) Other valuesRefer to **Table 20 Measurement Switch Settings (on Page 116)**, do not use any combination not specified this table. *Note: Due to the pre-amplifier and multiplexer settling time, the measurement mode has to be set prior to setting the start flag if this settling time is longer than the acquisition delay (refer to **Table 182 ADC Characteristics (on Page 548)**).*
INV	6	rw	**Inversion of Pre-Amplifier Input** The input voltage of the pre-amplifier is: 0 Not inverted 1 Inverted If two measurements with and without inversion are carried out, the offset of the pre-amplifier and ADC can be determined and eliminated for a dedicated measurement mode. This is valid for all modes which use the ADC input A. **INV** only has an effect if the ADC input A is used.
CSEL	11	rw	**Fast Settling of Pre-Amplifier** 0 Fast settling disabled, capacitor C1 of the pre-amplifier is used 1 Fast settling enabled, capacitor C2 with a lower capacitance is used. For the PWPA Measurement Mode **CSEL** must = 1.
TC	14:12	rw	**Current Source for Current Measurement Mode** 000 $I_S = I_R = 0$: Voltage mode (Default) 001 $I_S = 30\ \mu A I_R$ =60 µA: Current mode 010 $I_S = 60\ \mu A I_R$ =60 µA: Current mode 011 $I_S = 90\ \mu A I_R$ =60 µA: Current mode 100 $I_S = 120\ \mu A I_R$ =60 µA: Current mode 101 $I_S = 150\ \mu A I_R$ =60 µA: Current mode 110 $I_S = 180\ \mu A I_R$ =60 µA: Current mode 111 $I_S = 210\ \mu A I_R$ =60 µA: Current mode Only use the Current Measurement Mode for the measurements defined in **Table 20 Measurement Switch Settings (on Page 116)**. Current sources are switched on as soon as **TC** is set to any value greater than 0_H. **TC** has to be set to 0_H to turn current sources off.
RESERVED	15, 10:7	r	Reserved; these bits must be left at their reset values.

MEAS_CTRL2

Measurement Control Register 2 **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START	RESERVED	ADCON	ENTRIG	ENSTOP	RESERVED	MXREF			BUFSIZE		FREQ				

Field	Bits	Type	Description
FREQ	2:0	rw	**ADC Sampling Rate Control** 000 Single shot mode (Default) 001 Repetitive mode with f_M*S/(32*640) (typical 0.39 kHz) 010 Repetitive mode with f_M*S/(32*320) (typical 0.78 kHz) 011 Repetitive mode with f_M*S/(32*160) (typical 1.56 kHz) 100 Repetitive mode with f_M*S/(32*80) (typical 3.13 kHz) 101 Repetitive mode with f_M*S/(32*40) (typical 6.25 kHz) 110 repetitive mode with f_M*S/(32*20) (typical 12.50 kHz) 111 Repetitive mode with f_M*S/(32*15) (typical 16.67 kHz) f_M is the peripheral master clock frequency clk_bus. f_M = 52 MHz (typical) and S = K/L, K = 02_H and L = $0D_H$ (typical). See **Section 7.3.12 (on page 136)**. If **ENSTOP** = 1, successive conversion is stopped after the number of measurements defined in **BUFSIZE** until it is started again by **START** or the ADCTRIG signal. The ADC performs measurements adjusted in **MEAS_CTRL1.MX** with frequency defined in **FREQ**.
BUFSIZE	5:3	rw	**Size of Output Buffer** 000 **MEAS_DATA0** (Default) 001 **MEAS_DATA0..MEAS_DATA1** 010 **MEAS_DATA0..MEAS_DATA2** 011 **MEAS_DATA0..MEAS_DATA3** 100 **MEAS_DATA0..MEAS_DATA4** 101 **MEAS_DATA0..MEAS_DATA5** 110 **MEAS_DATA0..MEAS_DATA6** 111 **MEAS_DATA0..MEAS_DATA7**

Field	Bits	Type	Description
MXREF	9:6	rw	**Reference Measurement Mode** 00_H No second measurement(Default) Other valuesRefer to **Table 20 Measurement Switch Settings (on Page 116)**, do not use any combination not specified this table. For **MXREF** > 0_H: 1. A measurement with the mode defined in **MEAS_CTRL1.MX** is carried out 2. A measurement defined by **MXREF** is carried out. The **MX** measurement result is always stored first, then the **MXREF** measurement result is stored in the **MEAS_DATAx** with the next higher index. The **MEAS_STAT.READY** flag is set to 1 and a RDYIRQ interrupt is generated depending on the value of **BUFSIZE**. **MEAS_STAT.WPTR** indicates where the last value was written to. For **MXREF** = 0_H no second measurement is carried out. For **MXREF** > 0_H it is recommend to set **BUFSIZE** to 1, 3, 5 or 7. If **MEAS_CTRL1.MX** = 0_H, **MXREF** is ignored.

Field	Bits	Type	Description
ENSTOP	11	rw	**Circular Output Buffer Disable** 0 The output buffer operates as a circular buffer with size defined in **BUFSIZE**. *Note:The results of a series of conversions are stored in increasing order in **MEAS_DATAx** beginning at **MEAS_DATA0** and ending at **MEAS_DATAx[BUFSIZE]**. The following result is stored in **MEAS_DATA0** again and circular operation is going on.* *After a change in any of the **MEAS_CTRL1** bit fields (except **START** and **ENSTOP**), the next measurement result is always written to register **MEAS_DATA0**.* *The **MEAS_STAT.WPTR** field indicates the index of the **MEAS_DATAx** where the latest value was written to.* *Whenever a new value is ready for reading in the with the index defined in **BUFSIZE**, the **MEAS_STAT.READY** flag is set to 1 and a RDYIRQ interrupt is generated.* 1 **In the Repetitive Mode**: The output buffer operates as a linear buffer with size defined in **BUFSIZE**. The results of a series of conversions are stored in increasing order in **MEAS_DATAx** beginning at **MEAS_DATA0** after successive conversion is started by setting **MEAS_CTRL1.START** or via the signal ADCTRIG. The last result is written to **MEAS_DATAx[BUFSIZE]** and successive conversion is stopped until it is started again. *Note:The **MEAS_STAT.WPTR** field, **MEAS_STAT.READY** flag, and RDYIRQ interrupt show the same behavior as they do when **ENTRIG** = 0.* **In the Single Shot Mode**: Only **MEAS_DATA0** is used and is independent from the buffer size defined in **BUFSIZE**. If **BUFSIZE** is greater than 0, no RDYIRQ interrupt is generated.
ENTRIG	12	rw	**Triggered Mode Enable** 0 Triggered Mode Disabled 1 Conversion is started when a rising edge occurs on the trigger signal ADCTRIG provided by the GSM timer unit. *Note:Setting **MEAS_CTRL1.START** has no effect when **ENTRIG** = 1.* **In the Repetitive Mode**: Successive conversions starts with a rising edge of **ADCTRIG** **In the Single Shot Mode**: A single conversion is carried out whenever a rising edge of ADCTRIG occurs..

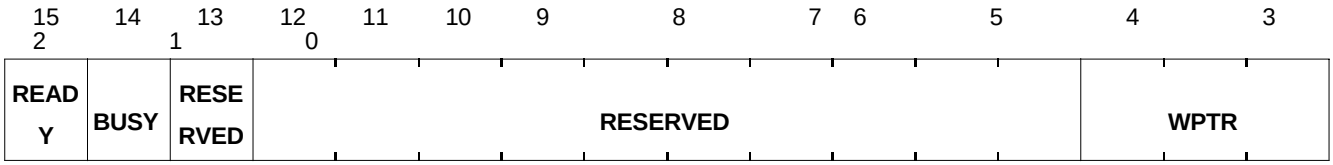
Field	Bits	Type	Description
ADCON	13	rw	**ADC Power Save Control** 0 The ADC and pre-amplifier are in the Power Save Mode and no conversion is performed. 1 The ADC and pre-amplifier are powered up and is ready for start of conversion. *Note:**ADCON** has to be set to 1 in advance to any measurement due to the wake-up time according to* **Table 182 ADC Characteristics (on Page 548)**.
START	15	rw	0 No action. **START** is always cleared when read. 1 Initiates conversion: • In the single shot mode a single conversion is performed when **START** is set to 1. • In repetitive mode successive conversion is started when **START** is set to 1. *Note:Setting **START** has no effect if **ENTRIG** is set to 1. If **START** is set to 1 during a conversion, it is ignored. The measurement mode (**MEAS_CTRL1.MX**, **MEAS_CTRL1.INV**, **MEAS_CTRL1.CSEL**, **MEAS_CTRL1.TC**, **MEAS_CTRL2.FREQ**, **MEAS_CTRL2.BUFSIZE**, etc.) must be set properly in advance to setting **START** to 1 to ensure settling and wake-up times.*
RESERVED	14, 10	r	Reserved; these bits must be left at their reset values.

7.3.9.2 Measurement Status Register

This register contains the READY flag indicating if a new value can be read and a BUSY flag indicating if a conversion is in progress.

MEAS_STAT

Measurement Status Register **Reset value: 4000_H**



Field	Bits	Type	Description
WPTR	2:0	rh	**Write Pointer** Indicates the **MEAS_DATAx** index where the latest valid result was written to. 000 **MEAS_DATA0** contains valid data 001 **MEAS_DATA0** and **MEAS_DATA1** contain data ... ... 111 **MEAS_DATA0** to **MEAS_DATA7** contain data **WPTR** is updated whenever valid data is written to **MEAS_DATA0** to **MEAS_DATA7**.
BUSY	14	rh	**Busy Flag** 0 ADC is free 1 An ADC operation is in progress or the ADC is in the Power Save Mode. *Note: No new ADC conversion should be started while **BUSY** is set.*
READY	15	rh	**Data Ready** 0 No data is available for reading. **READY** is cleared by hardware after any of the **MEAS_DATAx** registers are read. 1 Data is available for reading. **READY** flag is set and a RDYIRQ interrupt is generated whenever a new value is ready for reading in **MEAS_DATAx**. The index is defined in **MEAS_CTRL2.BUFSIZE**.
RESERVED	13:3	r	Reserved; these bits must be left at their reset values.

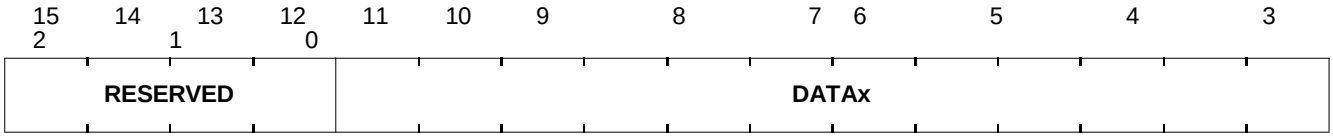
7.3.9.3 Measurement Data Registers 0-7

Data from the ADC is transferred via the eight Measurement Data Registers to the X-Bus bus.

MEAS_DATAx

MEAS_DATA0
MEAS_DATA1
MEAS_DATA2
MEAS_DATA3
MEAS_DATA4
MEAS_DATA5
MEAS_DATA6
MEAS_DATA7

Measurement Data Registers **Reset value: 0000_H**



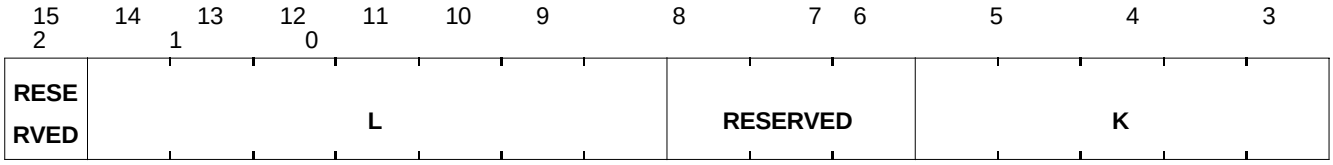
Field	Bits	Type	Description
DATAx (x = 0 to 7)	11:0	rh	**ADC Data 0..7** The number of registers used depends on **MEAS_CTRL2.BUFSIZE**.
RESERVED	15:12	r	Reserved; these bits must be left at their reset values.

7.3.9.4 Measurement Peripheral Clock Control Register

The internal peripheral clock clk_kernel_2 for the digital part of the measurement interface and of the reference voltage generation can be adjusted via a fractional clock divider with the rational factor K/L. The numerator K and the denominator L can be programmed in the **MEAS_CLK** register. This register is also part of the measurement X-Bus interface. In this way the clock clk_kernel_2 can be a constant, independent from the variable peripheral master clock clk_bus.

MEAS_CLK

Measurement Peripheral Clock Control Register **Reset value: 0100_H**



Field	Bits	Type	Description
K		0:4	rw **Numerator of Fractional Divider** $K = 0_D$ ... 31_D Setting K depends on the frequency of clk_bus according **Table 30 K and L Values for Various Bus Clock Frequencies (on Page 137)**. K must always be < L. K = L is not allowed, except when K = L = 0. If K = L = 0, clk_kernel_2 is equal to clk_bus. *Note: K has to be programmed prior to any action of the ADC or band-gap.*
L		14:8	rw **Denominator of Fractional Divider** $L = 0_D$ ... 127_D L is set in dependence of the frequency of clk_bus according **Table 30**. *Note: L has to be programmed prior to any action of the ADC or band-gap.*
RESERVED	7:5, 15	r	Reserved; these bits must be left at their reset values.

7.3.9.5 Interrupts

The following interrupt requests (service requests) are generated by the measurement BPI. Refer to **Section 8.1 TEAKLite Interrupt Unit (on Page 302)**.

Table 23 Measurement BPI Interrupt Requests (Service Requests)

Interrupt Request	Description
RDYIRQ	**Ready Interrupt Request** Indicates that data can be read from the **MEAS_DATAx** registers. When a new data value is ready for reading in the **MEAS_DATAx[MEAS_CTRL2.BUFSIZE]** registers. The **MEAS_STAT.READY** flag is set to 1 and a RDYIRQ interrupt is generated

7.3.9.6 System Registers

The measurement interface and the analog control block (refer to **ANA_CTRLx (on Page 138)**) share one BPI and they have the following system registers:

- Service Request Control
- Peripheral Clock Control

MEAS_CLK

Measurement Interface Clock Control Register (refer to **Page 132**).

- Module Identification

MEAS_ID

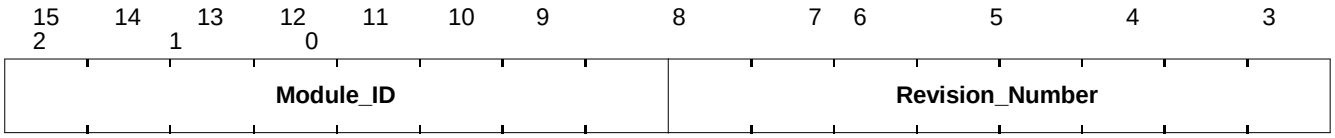
Measurement Interface Identification Register (refer to **Page 133**).

7.3.9.6.1 Measurement & Analog Identification Register

This is the ID for both the measurement interface and the analog control part

MEAS_ID

Measurement Identification Register **Reset value: 2413_H**



Field	Bits	Type	Description
Revision_Number	0:7	r	**Measurement Revision Number** These hard-wired bits are used for module revision numbering.
Module_ID	8:15	r	**Measurement Identification Number** These hard-wired bits are used for module identification numbering.

7.3.10 Special Operating Modes and Examples

7.3.10.1 Triggered and Repetitive Measurements

To measure at definite time, for example, during a burst, an A-to-D conversion can be started by the trigger signal ADCTRIG, which is generated by the programmable GSM Timer Unit. To enable this mode the bit **MEAS_CTRL2.ENTRIG** must be equal to 1.

To get a series of measurement values in a dedicated measurement mode, repetitive conversion can be enabled by setting the field **MEAS_CTRL2.FREQ** to one of its possible sampling rates. Repetitive conversion can be started either by:

- If **ENTRIG** = 0, setting the **MEAS_CTRL2.START** flag
- If **ENTRIG** = 1, the signal ADCTRIG.

MEAS_CTRL2.BUFSIZE controls whether the **MEAS_STAT.READY** flag is set after each conversion or after a series of up to 8 conversions (that are stored in **MEAS_DATAx**).

The bit **MEAS_CTRL2.ENSTOP** controls whether the output buffer operates as linear or circular buffer.

In the triggered mode (**ENTRIG** = 1) measurement is started when a rising edge on signal ADCTRIG occurs. Setting **START** has no effect while **ENTRIG** = 1. The user has to be sure that the GSM Timer Unit is programmed to set ADCTRIG at the desired points in time, for example, at the start of a burst taking in account any delays occurring in the system.

In the repetitive mode successive measurements with frequency defined in **FREQ** are carried out until **FREQ** is set to 0_H or **ENSTOP** is set to 1.

7.3.10.2 Measurement of VBAT During Transmission Bursts

To measure the battery voltage at several instances in time during a transmission burst on pin M8 consider the following example:

The battery voltage should be automatically measured at 6 equidistant instances in time during a dedicated burst. The six results should be available for reading after completion of measurement operation has been indicated via an interrupt request.

Use the configurations in **Table 24** and **Table 25**

Table 24 MEAS_CTRL1 = 0009

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			TC			CSEL	RESERVED						INV	MX		
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Table 25 MEAS_CTRL2 = 382E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START	RESERVED	ADC ON	EN TRIG	EN STOP	RESERVED	MXREF			BUFSIZE		FREQ				
0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0

The GSM Timer Unit has to be programmed so that the ADCTRIG signal provides a rising edge at the beginning of the burst. The falling edge of ADCTRIG is not evaluated.

When the RDYIRQ interrupt occurs, the results can be read from **MEAS_DATA0** ... **MEAS_DATA5**.

7.3.10.3 Measurement of PWPA During Tail Symbols of a Burst

To measure the output transmission power during the tail symbols at the beginning and end of one dedicated burst in every frame, use the configurations in **Table 26** and **Table 27**.

Table 26 MEAS_CTRL1 = 080B

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			TC			CSEL	RESERVED						INV	MX		
0	0	0	0	0	1	0	0	1	1	0	1	0	0	0	0

Table 27 MEAS_CTRL2 = 3008

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START	RESERVED	ADC ON	EN TRIG	EN STOP	RESERVED	MXREF			BUFSIZE		FREQ				
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0

The GSM Timer Unit has to be programmed that ADCTRIG signal provides rising edges at the tail symbols at the beginning and the end of the burst in every frame considering all relevant delays. The falling edge of ADCTRIG is not evaluated.

When the RDYIRQ interrupt occurs, the results can be read from **MEAS_DATA0** ... **MEAS_DATA1**. They will be

overwritten in the following frame.

To stop measurements, set **MEAS_CTRL2.ENTRIG** to 0 or change configuration of the GSM Timer Unit.

7.3.10.4 Measurement of Battery Technology

To carry out a single measurement of the battery technology in the current mode on pin M0, use the configurations in **Table 28** and **Table 29**.

Table 28 MEAS_CTRL1 = 3001

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		TC		CSEL	RESERVED			INV	MX						
0	0	0	1		1	0	0		0	0	0	0	0	0	1

Table 29 MEAS_CTRL2 = A000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START	RESERVED	ADCON	ENTRIG	ENSTOP	RESERVED	MXREF			BUFSIZE		FREQ				
1	0	0	1		0	0	0		0	0	0	0	0	0	0

If **MEAS_STAT.READY** is set to 1 or a RDYIRQ interrupt has occurred, the result can be read from **MEAS_DATA0**.

7.3.10.5 Power Save Features

The digital part of the measurement interface is switched off by disabling the clock of the bus interface (**MEAS_CLK.K** = 0).

Power control for the complete analog block of the measurement interface is managed via the bit **MEAS_CTRL2.ADCON**, which can switch the ADC and pre-amplifier to the power save mode. Additional power down features for the mixed signal section are explained in **Section 7.2.2 Standby Clock Control Unit (SCCU) (on Page 82)**.

7.3.11 System Block Diagram of Measurement Interface

The measurement interface consists of two parts, the analog part and the digital part. Both parts communicate via the signals shown in **Figure 46**. The measurement Interface is connected to the X-Bus via an X-Bus interface. The signal ADCTRIG is provided by the GSM Timer Unit, which is not part of the measurement interface. Usually, ADCTRIG is used to control measurement operation for battery voltage or transmission power (PWPA).

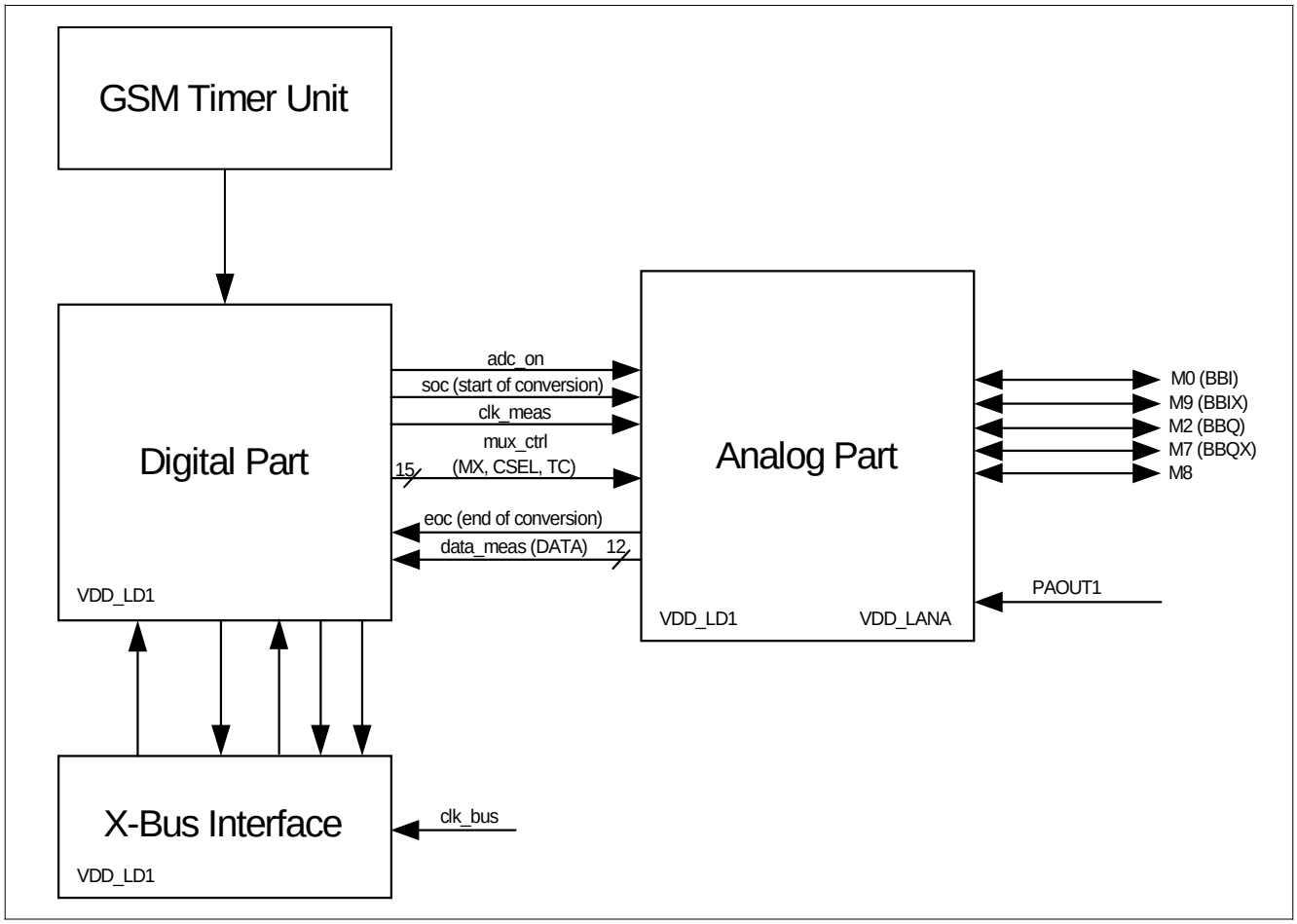


Figure 46 Logic Block Diagram of Measurement Interface

7.3.12 Clock Control of Measurement Interface

All clocks necessary for the measurement interface (and the reference voltage generation) are derived from the peripheral master clock clk_bususing clock dividers as it is shown in **Figure 47**.

This clk_bus is divided by a fractional divider with factor K/L giving clk_kernel_2.

This clk_kernel_2 is divided by M = 8 in the digital part and the output is clk_meas. clk_kernel_2 determines most of the timing characteristics of the ADC (refer to **Table 182 ADC Characteristics (on Page 548)**).The programming of the fractional divider for different frequencies of the clock clk_bus are listed in **Table 30 K and L Values for Various Bus Clock Frequencies (on Page 137)**. K and L are in the **MEAS_CLK** register.

Note: For normal operation, that is, clk_bus = 52 MHz, the fractional divider has to be programmed with K = 02_H, L = $0D_H$ to have a clk_kernel_2 of 8 MHz and a clk_meas of 1 MHz which is the usual operating condition for the ADC.

The programming of the measurement interface has to be done in the following order:

1. K and L in the **MEAS_CLK** register are programmed.
2. **MEAS_CTRL1**/**MEAS_CTRL1** are programmed
3. An A-to-D conversion can be started.

Attention: Never start a A-to-D conversion before programming the appropriate frequency of clk_kernel_2.

Note: The band_gap needs the clk_kernel_2 before it is powered up.

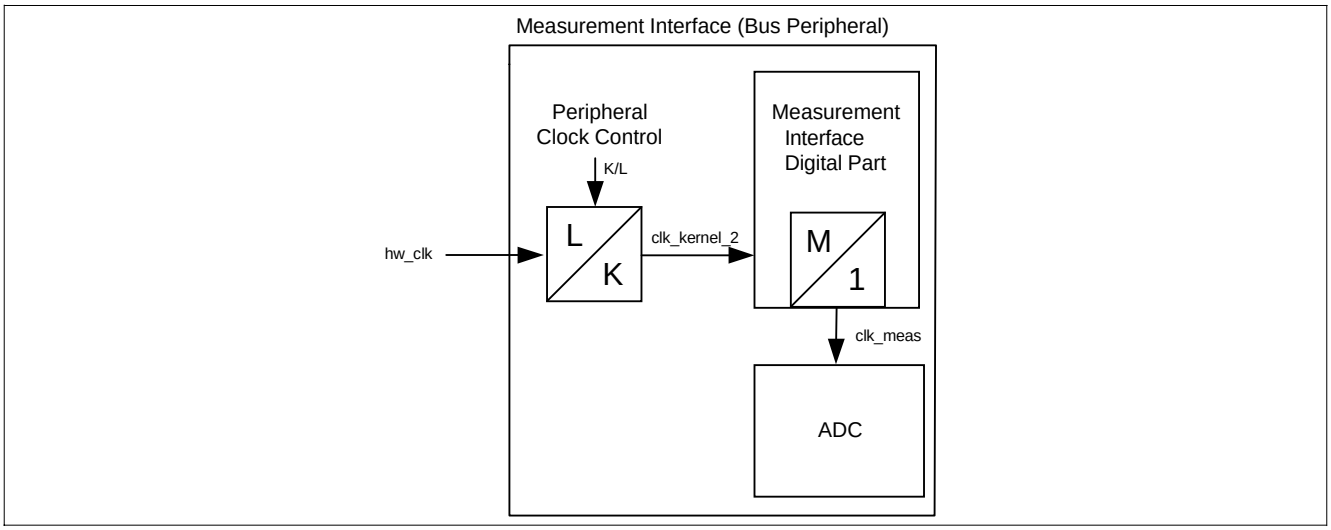


Figure 47 Clock Control for Measurement Interface

Refer to the Clock Domain in **System Integration (on Page 113)**.

Table 30 K and L Values for Various Bus Clock Frequencies

clk_bus in MHz	K	L
52.0	2_D	13_D
26.0	4_D	13_D
13.0	8_D	13_D
X		X X

7.4 Analog Control Registers

7.4.1 Functional Overview

The analog part contains several dedicated registers which are provided for specific control purposes of the analog blocks.

- Control of reference voltage generation
- Control of nominal common mode voltage of GSM TX path
- Control of offset voltage for power amplifier control
- Identification.

7.4.2 Analog Control Register 1

ANA_CTRLx

ANA_CTRL1

Analog Control Register **Reset value: 0000_H**

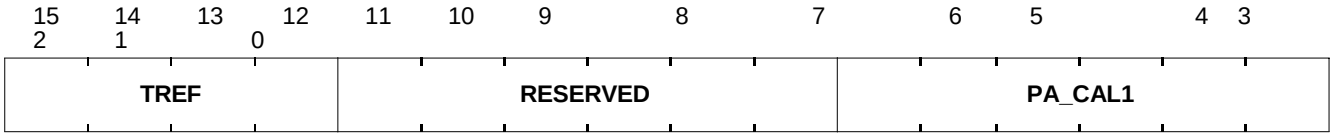
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PAOPM1	RESERVED	RESERVED				TXREF_PU	RESERVED	RESERVED	RESERVED	TX_DIS	RESERVED	PA_OFF1	BG_PWUP	RXREF_PU

Field	Bits	Type	Description
RXREF_PU	0	rw	**RX Local Reference Buffer Switch** 0 Local reference buffer in baseband receive path is off when signal RXON = 0. 1 Local reference buffer in baseband receive path is on when RXON = 0.
BG_PWUP	1	rw	**Band-Gap Power Control** 0 Power down of reference generation 1 Normal operation of reference generation. clk_kernel_2 must be running before the band-gap can be enabled via **BG_PWUP**.
PA_OFF1	2	rw	**PAOUT1 Analog Offset Correction** 0 No Analog offset correction 1 Analog offset correction with nominal -50 mV
TX_DIS	4	rw	**DIsable Analog Part of Modulator Unit** 0 Analog part is switched on and off depending on the signal TXON 1 Analog part is always off *Note: TX_DIS is set to 1 in those cases, where the analog part of the on chip modulator isn't used, for example, if an external modulator is connected to the E-GOLDvoice.* *Note:*
TXREF_PU	8	rw	**TX Local Reference Buffer Switch** 0 Local reference buffer in baseband transmit path is off when signal TXON = 0. 1 Local reference buffer in baseband transmit path is on when TXON = 0.
PAOPM1	14	rw	**PAOUT1 Output Buffer Operation Mode** 0 Output buffer in standard mode 1 Output buffer in enhanced transient performance mode
RESERVED	15, 13:9, 7:5, 3	r	Reserved; these bits must be left at their reset values.

7.4.3 Analog Control Register 2

ANA_CTRL2

Analog Control Register **Reset value: 0000_H**



Field	Bits	Type	Description
PA_CAL1	5:0	rw	**PAOUT1 Digital Offset Correction** 00_H No digital offset correction every DAC input value is 01_H Increased by 1 LSB 02_H Increased by 2 LSB ... $3F_H$ Increased by 63 LSB *Note: Refer to **Section 7.7.1.4 D-to-A Conversion and Post Filtering (on Page 166)** and **Section 7.3.7.1 PA Control Hardware Offset Measurement PAOUTOF1 (on Page 122)**.*
TREF	15:12	rw	**Common Mode Voltage for Baseband TX Path[1)]** 1111 0.900 V (typical) 0000 0.955 V (typical) (default) 0001 1.010 V (typical) 0010 1.065 V (typical) 0011 1.120 V (typical) 0100 1.175 V (typical) 0101 1.230 V (typical) 0110 1.285 V (typical) 0111 1.340 V (typical)
RESERVED	11:6	r	Reserved; these bits must be left at their reset values.

1) (BB_I + BB_Ix)/2 and (BB_Q + BB_Qx)/2 in TX case

7.5 Keypad

System Integration

- Supply domain: VDD_LD1
- Chip internal interfaces:
  - Clock domain: Refer to **Section 7.2.1.3 Sub-System Clocks and Enables (on Page 67)** and see **Figure (on Page 71)**.
  - Bus domain: X-Bus
- Interrupt sources:
- Monitor Pins: Refer to **Section 9.7.10 Internal Signal Monitoring (on Page 435)**.

7.5.1 Description

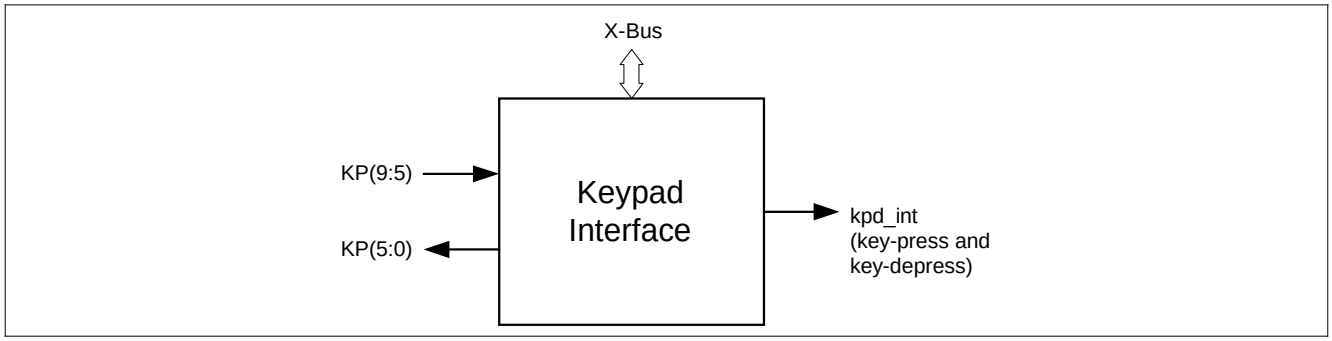


Figure 48 Block Symbol of the Keypad Interface

The keypad interface is connected to the X-Bus, together with the XBIU and the Shared Memory Register, using a single Bus Interface.

The keypad supports two scan modes:

- By default, the keypad is a 4x6 scan matrix (4 input and 6 output pins).
- To set the keypad to a 5x5 scan matrix (5 input and 5 output pins) set KPCTRL.mode_sel to 1.

The scan mode should be determined at the very beginning of the system start because changes are not allowed later.

There are four registers in keypad module.

7.5.2 Keypad Registers

7.5.2.1 Keypad Identification Register

KBID

Keypad Identification Register **Reset values: A803$_H$**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Module_ID								Revision_Number							

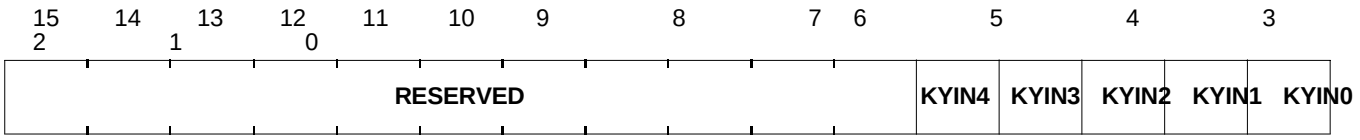
Field	Bits	Type	Description
Revision_Number	0:7	r	**Keypad Revision Number** These hard-wired bits are used for the module revision numbering.

Field	Bits	Type	Description
Module_ID	8:15	r	**Keypad Identification Number** These hard-wired bits are used for module identification numbering.

7.5.2.2 Keypad Input Register

KBDINP

Keypad Input Register **Reset value: 000X$_H$**

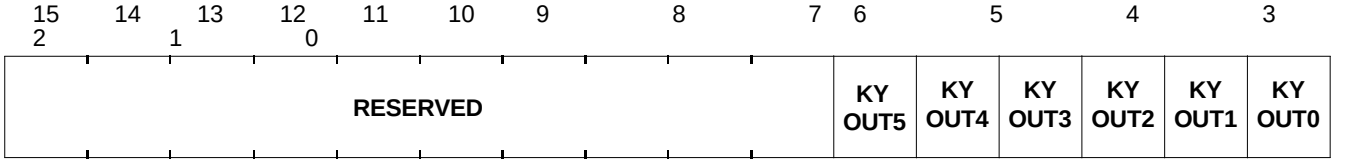


Field	Bits	Type	Description
KYINx (x = 0 to 4)	0:4	r	**Keypad Input x** The reset value for this register depends on the value of **KYINx**.
RESERVED	15:5	r	Reserved for future use; these bits must be left at their reset values.

7.5.2.3 Keypad Output Register

KBDOUT

Keypad Output Register **Reset value: 0000$_H$**

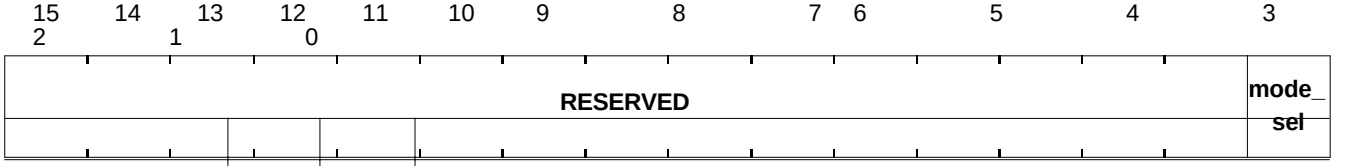


Field	Bits	Type	Description
KYOUTx (x = 0 to 5)	0:5	w	**Keypad Output x** Output pin values
RESERVED	15:6	r	Reserved for future use; these bits must be left at their reset values.

7.5.2.4 Keypad Control Register

KPCTRL

Keypad Control Register **Reset value: 0000$_H$**



Field	Bits	Type	Description
mode_sel	0	rw	**Keypad Scan Mode Selection** 0 4x6 scan mode 1 5x5 scan mode
RESERVED	15:1	r	Reserved for future use; these bits must be left at their reset values.

7.5.3 Keypad Port

The two scan modes of the keypad operation the same way. The only difference is the scan matrix structure (refer to **Description (on Page 140)**). The description below is based on the 4x6 structure, but the principle is the same for the 5x5 structure.

The levels at the input pins KP(9:4) loaded into **KBDINP** are obtained with a read access. The levels that the output pins KP(5:0) have to show have to be loaded into **KBDOUT** with a write access. Pressing a key generates an interrupt to the MCU. Releasing this key generates an additional interrupt. The length of a key press can be determined by measuring the time between these two interrupts.

Figure 49 shows a block diagram of the keypad interface.

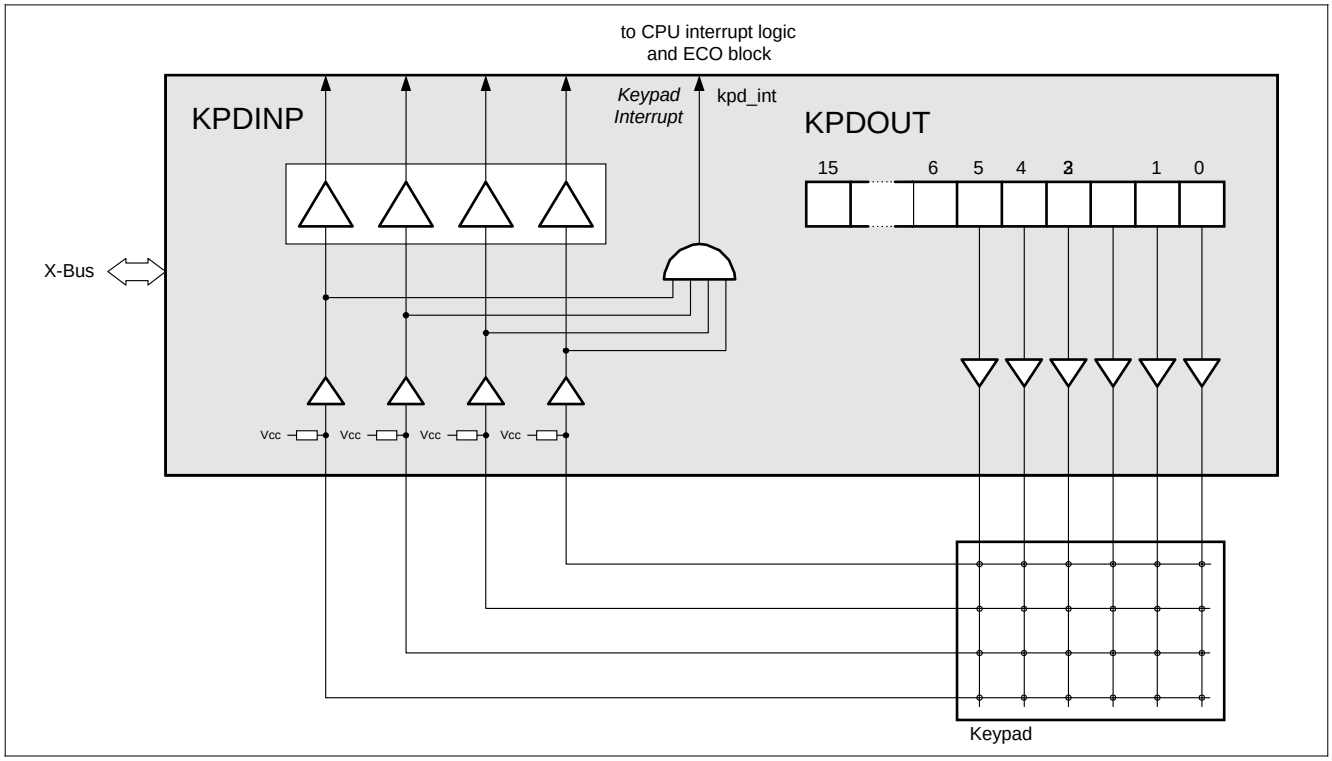


Figure 49 Block Diagram of the Internal Keypad Port Circuitry

Keypad Port Operation

As long as no input port pin (KP9...6) is connected to an output port pin (KP5...0) the input pins are pulled high internally if the pad is programmed with a pull-up in the port control logic. This causes a high level to appear on the keypad interrupt input of the controller.

If the controller wants to check the key status, it loads **KBDOUT** with 00_H to start the keypad scan. If a key is pressed, both the corresponding keypad row and column are connected to each other and the corresponding **KBDINP** register bit is set to LOW. Due to the LOW level the kpd_int interrupt (active high, duration 2 X-Bus clock cycles) is generated to the controller and to the SCCU block, which is then able to initiate an early wake-up procedure in case of the sleep mode. Knowing the correct row number, the controller scans the six bits of **KBDOUT** by switching them from 1 to 0 step-by-step to find the correct column number.

During normal operation the clock frequency of the keypad interface is 13 MHz, during sleep mode the 32 kHz sleep clock is provided.

Pins not to be used for the keypad can be configured for either GPIO port or alternate functionality as described in **Chapter 3 Pin Descriptions**.

After a key-press IRQ is generated and the control logic scans the keypad to detect the key release. When the release of the same key is detected, a second interrupt is generated.

- There must be an instruction or NOP between a write to **KBDOUT** and a read from **KBDINP** to avoid pipeline effects.
- The interrupt to the CPU is generated even when the X-Bus clock is switched off.
- If the keypad port is not used, the inputs are pulled high internally.
  KBDINP is not a real register, it is a set of four parallel drivers (with enables), which connect the input pins with the internal bus. A read access opens these drivers.
- Pressing two or more keys at the same time leads to a short circuit of two output pins and with this to a short circuit current. Reducing the short circuit current requires either external series resistors or a limitation of the short circuit time.
- The junction resistance of the keypad should be < 2 k& and the maximum leakage current is 5 ∝A.

7.6 SIM Interface

System Integration

- Supply domain: VDD_LD1
- Chip internal interfaces:
  - Clock domain: Refer to **Section 7.2.1.3 Sub-System Clocks and Enables (on Page 67)** and see **Figure (on Page 71)**.
  - Bus domain: X-Bus bus
- Interrupt sources: 3 regular controller interrupts: USIM_ERR_INT, USIM_IN_INT, USIM_OK_INT.
- Chip external signals related to this block (refer to **Chapter 3 Pin Descriptions** for pin configuration options): CCIO, CC_VZ_n, CC_CLK, and CC_RST.
- Monitor pins:

7.6.1 Functional Overview

The SIM interface is compatible with the ISO 7816-3 IC Card standard on the issues required by the GSM 11.11 Phase 2+ standard. Features included in the interface are:

- SIM Card is provided with a 3.25 MHz clock (derived from 13 MHz clock).
- Automatic parity error detection and error signalling in RX mode.
- Automatic character repetition on parity errors in TX mode.
- Automatic switching between RX and TX mode.
- Automatic work waiting timer supervision for use with T=0 protocol.
- Supports enhanced speed SIM's as specified in GSM 11.11 - Phase 2+ using the SIM Baud Rate Factor settings.
- Supports both SW and HW controlled T=0 protocol.
- Work Waiting for T=0 mode.
- Supports 1 MHz SIM clock in low power mode.
- Supports GSM Phase 2 clock stop modes.
- Automatic power down for immediate SIM deactivation.

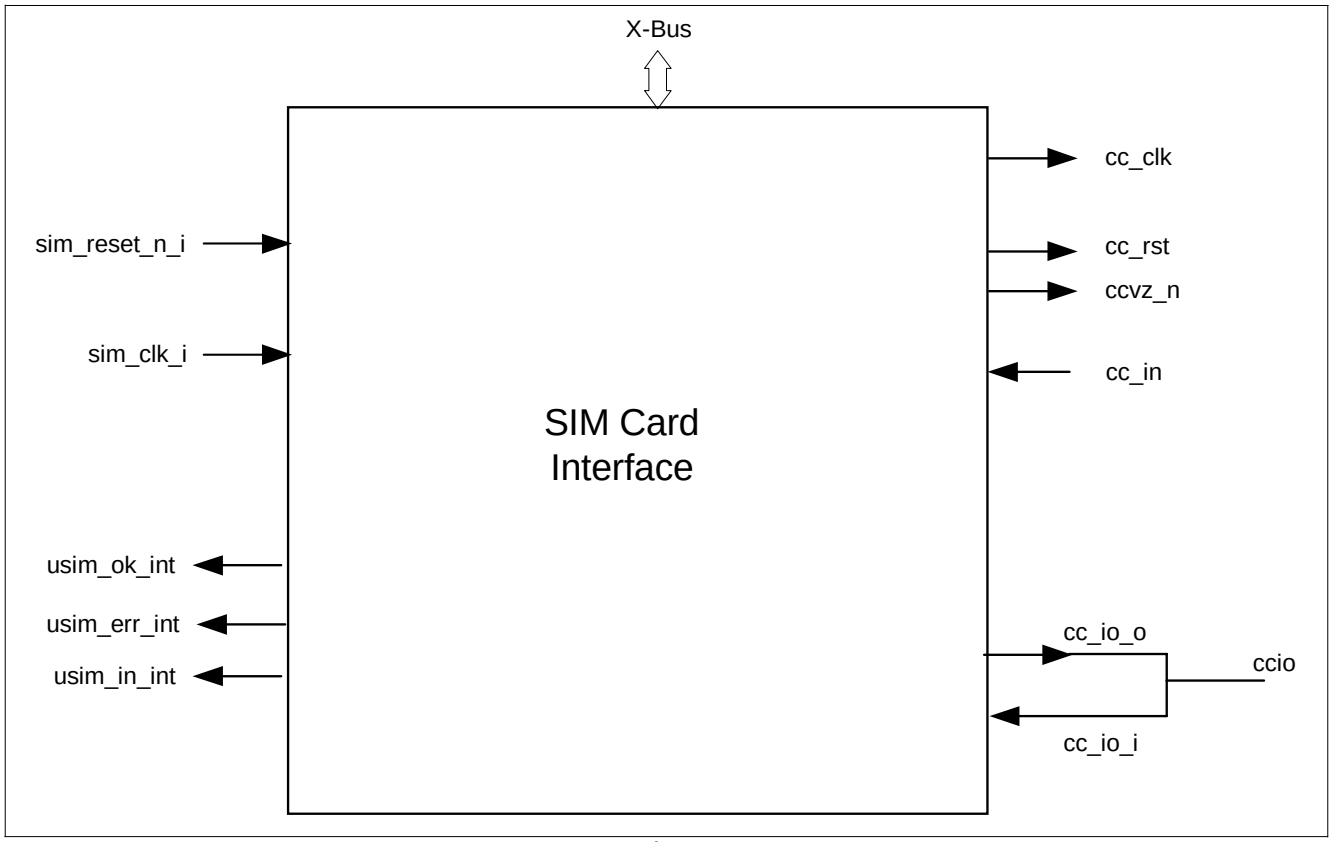


Figure 50 Block Symbol of the SIM Card Interface[1)]

Refer to **Figure 51**, the block diagram.

The SIM interface consists of 7 main modules:

- SIM UART
- T=0 Controller
- Clock generation unit.
- Register and state machine unit
- Character timer unit (Work Waiting Timer for T=0)

For an overview, see **Figure 51**.

1) Refer to Clock Domain in **System Integration** on page 144.

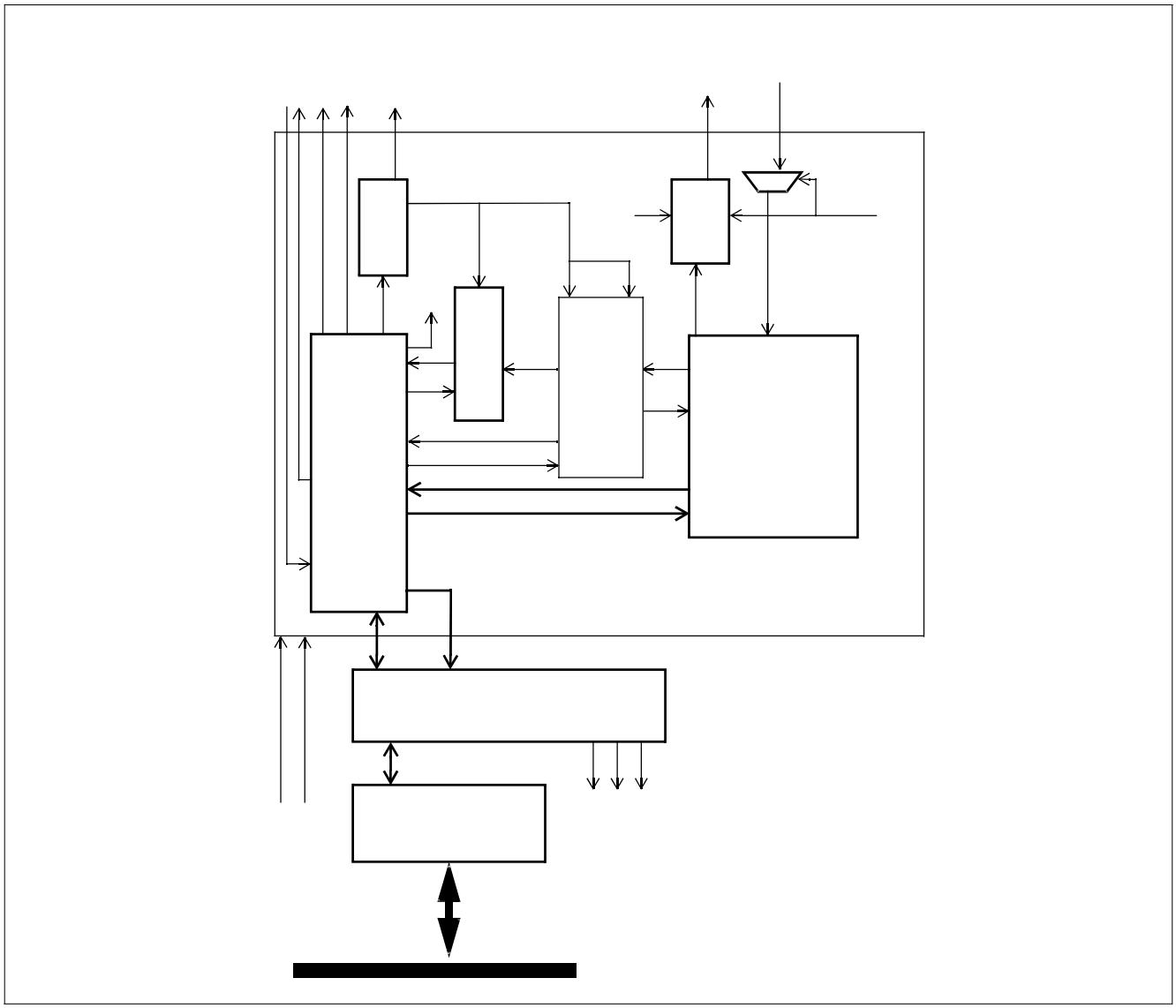


Figure 51 SIM Card Interface Block Diagram[1)]

7.6.2 SIM Register Overview

Table 31 SIM Register Overview

Register Group	Register Name	Register Symbol
System Register	SIM Identification Register	SIMID

1) Refer to Clock Domain in **System Integration** on page 144.

Table 31 SIM Register Overview

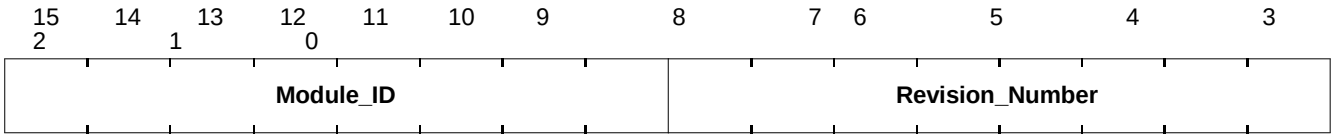
Register Group	Register Name	Register Symbol
Control and Status Registers	SIM Control Register	**SIMCTRL**
	SIM Baud Rate Factor Register	**SIMBRF**
	SIM Status Register	**SIMSTATUS**
	SIM Interrupt Enable Register	**SIMIRQEN**
	SIM RX Spacing Register	**SIMRXSPC**
	SIM TX Spacing Register	**SIMTXSPC**
	SIM Character Timer Registers	**SIMCHTIMERx**
Data Registers	SIM transmit data register SIM	**SIMTX**
	receive data register SIM	**SIMRX**
	instruction class register SIM	**SIMINS**
	parameter 3 register SIM	**SIMP3**
	Status Word 1 Register SIM	**SIMSW1**
	Status Word 2 Register	**SIMSW2**

7.6.3 Register Description

7.6.3.1 SIM Interface Identification Register

SIMID

SIM Interface Identification Register **Reset values: 0012_H**



Field	Bits	Type	Description
Revision_Number	0:7	r	**SIM Interface Revision Number** These hard-wired bits are used for the SIM revision numbering.
Module_ID	8:15	r	**SIM Interface Identification Number** These hard-wired bits are used for SIM identification numbering.

7.6.3.2 SIM Control Register

The **SIMCTRL** register contains information for controlling the operation mode of the SIM interface except the clock signals used.

SIMCTRL

SIM Control Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	**SMCIOSWACT**	**UARTON**	**CLKHIGH**	**CLKSEL**	**SIMON**	**SIMPDWN**	**APDWN**	**RPTOFF**	**ERROFF**	**SIMRST**	**SIMVCC**	**SIMEN**	**SIMIOL**	**SIMT0**	**INCON**

Field	Bits	Type	Description
INCON	1	rw	0 Direct convention. LSB transmitted/received first, logic 1 equals high level. 1 Inverse convention. MSB transmitted/received first, logic 1 equals low level.
SIMT0	2	rw	0 SIM runs in normal character based mode. This mode is used during ATR (Answer to Reset). 1 SIM runs in instruction mode. The T=0 protocol is handled in HW.
SIMIOL	2	rw	0 Sets SIM I/O to low level as long as this bit is reset. 1 SIM I/O line enabled.
SIMEN	3	rw	0 SIM interface disabled. 1 SIM interface enabled.
SIMVCC	4	rw	0 SIM supply voltage is removed ($\overline{CCVZ}$ at high level). 1 SIM supply voltage is applied ($\overline{CCVZ}$ at low level).
SIMRST	5	rw	0 SIMRST signal at low level. 1 SIMRST signal at high level.
ERROFF	6	rw	0 Enables error signalling during transmission and reception 1 Disables error signalling during transmission and reception.
RPTOFF	7	rw	0 Character retransmission is enabled. 1 Character retransmission is switched off.
APDWN	8	rw	0 Automatic power down is disabled. 1 Automatic power down of the SIM occurs on a falling edge of CCIN.
SIMPDWN	9	rw	0 The SIM signals can be controlled manually. 1 When set, the SIM is powered down and the SIM signals stay disabled until SIMPDWN is reset. This also causes SIMEN to be disabled after a delay of 5 of the 13 MHz clock cycles.
SIMON	10	rw	0 The 3.25 MHz SIM clock is switched off. Can be used if the SIM supports clock stop mode. 1 The 3.25 MHz SIM clock is switched on.
CLKSEL	11	rw	0 SIM clock is 3.25 MHz. 1 SIM clock is 1.08 MHz. Must only be used between commands.
CLKHIGH	12	rw	0 SIM clock stops at a low level. 1 SIM clock stops at a high level.
UARTON	13	rw	0 UART, T=0 clocks are off. Use this mode between instructions to save power. 1 UART and T=0 clock is on.

Field	Bits	Type	Description
SMCSWACT	14	rw	0 Smart card interface data output is connected to the SMC_IO pin. Output SMC_IOSW is high (if pin is not configured as GPIO) 1 Inverted Smart card interface data output is connected to SMC_IOSW. Pin SMC_IO is a tri-state input.
RESERVED	15	r	Reserved; these bits must be left at their reset values.

7.6.3.3 SIM Baud Rate Factor Register

This factor defines the baud rate used when communicating with the SIM card. On Reset the register is loaded with the value 93 ($5D_H$) corresponding with the default values for D and F (D = 1 and F = 372). The BRF is calculated using the following equation:

$$BRF \quad \frac{E}{4D} \tag{45}$$

Note: The SW must ensure that the BRF is an integer value and appropriate to the ISO 7816 and GSM specification, that is, BRF = 16 (10_h) for F = 512, D = 8 and BRF 8 (08_h) for F =512, D = 16.

SIMBRF

SIM Baud Rate Factor Register **Reset value: $005D_H$**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									BRF						

Field	Bits	Type	Description
BRF	6:0	rw	**SIM Baud-Rate Factor** Legal values are 2 to 127.
RESERVED	15:7	r	Reserved; these bits must be left at their reset values.

7.6.3.4 SIM Status Register

SIMSTATUS shows which event caused the interrupt and the status of pin CCIN.

*Note: The reset value of **SIMSTATUS** is 0000_H if the SIM card or Smart card is not connected and 0010_H if the SIM card or Smart card is connected.*

SIMSTATUS
SIM Status Register **Reset value: See Note**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							RESERVED	RESERVED	RESERVED	CHTIMEOUT	SIMDET	T0END	OVRRUN	PARINT	UARTOK

Field	Bits	Type	Description
UARTOK	0	rw	**UART Ready Interrupt** 0 No interrupt 1 One byte was received or transmitted without errors
PARINT	1	rw	**UART Parity Error Interrupt** 0 No error 1 Parity error occurred in receive or transmit direction
OVRRUN	2	rw	**UART Overrun Error** 0 No error 1 New byte written to **SIMTX** before previous byte has been transmitted or a new byte was received before **SIMRX** was read
T0END	3	rw	**T=0 Instruction Ended** 0 Command not executed 1 Command executed. Refer to **SIMSW1** and **SIMSW2** for status.
SIMDET	4	rw	**SIM Present Indication** This bit can always be read. 0 SIM card is not present 1 SIM card is present (pin SMC_IN = 1)
CHTIMEOUT	5	rw	**Character Timer Has Timed Out** 0 Has not timed out 1 The character time out as defined in the **SIMCHTIMERx** registers has expired. In T=0 mode this indicates that a WWT time out has occurred.
RESERVED	15:6	r	Reserved; these bits must be left at their reset values.

Notes:

1. *The reset value of **SIMSTATUS** is 0000_H if the SIM card or Smart card is not connected and 0010_H if the SIM card or Smart card is connected.*
2. *The values of **SIMSTATUS** can only be written by software with the value zero. Not all bits are writable by software:*

   *To reset the bits **PARINT**, **OVRRUN**, **T0END**, and **CHTIMEOUT** the value 0000_H must be written to the register. This does not effect **SIMDET** that reflects the level of the pin CCIN even if the interface has been disabled and does not effect the value of **UARTOK**.*
   *If **UARTOK** is set (due to successful data transmission) **PARTINT**, **OVRRUN**, and **CHTIMEOUT** are cleared.*

7.6.3.5 SIM Interrupt Enable Register

These bits enable the interrupts shown in **SIMSTATUS**.

SIMIRQEN

SIM Interrupt Enable Register **Reset value: 0000_H**

15 14 13 12 11 10 9 8 7 6	5	4	3	2	1	0
RESERVED	EN BWT TIMER	EN CH TIMER	EN T0 END	EN OVR	EN PAR	EN OK INT

Field	Bits	Type	Description
ENOKINT	0	rw	**UARTOK Interrupt Enable** 0 Disabled 1 Enabled
ENPAR	1	rw	**PARINT Interrupt Enable** 0 Disabled 1 Enabled
ENOVR	2	rw	**OVRRUN Interrupt Enable** 0 Disabled 1 Enabled
ENT0END	3	rw	**T0END Interrupt Enable** 0 Disabled 1 Enabled
ENCHTIMER	4	rw	**Character Timer Interrupt Enable** 0 Disabled 1 Enabled
ENBWTTIMER	5	rw	**BWT Timer Interrupt Enable** 0 Disabled 1 Enabled
RESERVED	15:6	r	Reserved; these bits must be left at their reset values.

7.6.3.6 SIM Transmit Register

SIMTX

SIM Transmit Register **Reset value: 0000_H**

15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	SIMTX

Field	Bits	Type	Description
SIMTX	7:0	rw	Data byte to be transmitted to the SIM card.
RESERVED	15:8	r	Reserved; these bits must be left at their reset values.

7.6.3.7 SIM Receive Register

SIMRX

SIM Receive Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SIMRX							

Field	Bits	Type	Description
SIMRX	7:0	rw	Data byte received from the SIM card.
RESERVED	15:8	r	Reserved; these bits must be left at their reset values.

7.6.3.8 SIM Instruction Class Register

This register contains the instruction code in the instruction class as described in the Protocol type T=0.

When this word is written the T=0 controller executes the instruction. Hence, this word must be written after **SIMP3**.

SIMINS

SIM Instruction Class Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							INS DIR	INS							

Field	Bits	Type	Description
INS	7:0	rw	Data byte received from the SIM card.
INSDIR	8	rw	**Instruction Direction** 0 The T=0 controller sends data to the SIM. 1 The T=0 controller receives data from the SIM.
RESERVED	15:9	r	Reserved; these bits must be left at their reset values.

7.6.3.9 SIM Parameter 3 Register

SIMP3

SIM Parameter 3 Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								P3							

Field	Bits	Type	Description
P3	7:0	rw	This value determines the number of bytes to receive or transmit depending on the bit **SIMINS.INSDIR**.
RESERVED	15:8	r	Reserved; these bits must be left at their reset values.

7.6.3.10 SIM Status Words

These registers contain the instruction status words SW1 and SW2 as described in the Protocol type T=0. These register can be read when a T0END interrupt has been received.

SIMSW1 always holds the last procedure byte sent by the SIM. This can be used for debugging purposes if the SW implements a SIM timeout.

These registers are reset when a new instruction starts.

SIMSW1

SIM Status Word 1 Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SW1							

Field	Bits	Type	Description
SW1	7:0	r	**Status Word 1**
RESERVED	15:8	r	Reserved; these bits must be left at their reset values.

SIMSW2

SIM Status Word 2 Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SW2							

Field	Bits	Type	Description
SW2	7:0	r	**Status Word 2**
RESERVED	15:8	r	Reserved; these bits must be left at their reset values.

7.6.3.11 SIM Receive Spacing Register

SIMRXSPC

SIM Receive Spacing Register **Reset values: T=0: 0028_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RXSPC							

Field	Bits	Type	Description
RXSPC	7:0	rw	This defines the spacing between a received command response character and the next transmitted character.
RESERVED	15:8	r	Reserved; these bits must be left at their reset values.

For T=0 Mode

The spacing is measured in 1/16 ETU, which means that the reset value of 0028_H is 2.5 ETU.

7.6.3.12 SIM Transmit Spacing Register

SIMTXSPC

SIM Transmit Spacing Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TXSPC							

Field	Bits	Type	Description
TXSPC	7:0	rw	This defines the extra spacing between successive transmitted characters for the character, T=0mode.
RESERVED	15:8	r	Reserved; these bits must be left at their reset values.

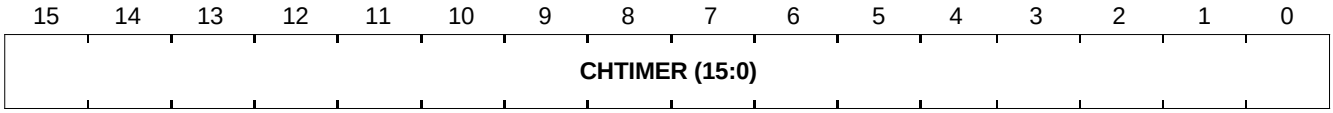
The spacing is given in ETU.

7.6.3.13 SIM Character Timer Registers

SIMCHTIMERx

SIMCHTIMER1

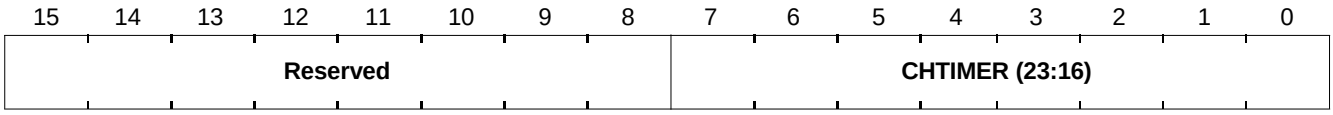
SIM Character Timer Register 1 **Reset value: 2580_H**



Field	Bits	Type	Description
CHTIMER (15:0)	15:0	rw	**SIM Character Timer**

SIMCHTIMER2

SIM Character Timer Register 2 **Reset value: 0000_H**



Field	Bits	Type	Description
CHTIMER (23:16)	7:0	rw	**SIM Character Timer**
RESERVED	15:8	r	Reserved; these bits must be left at their reset values.

The spacing is given in ETU. The reset value = 9600 ETU.

7.6.4 SIM Card Interface

The SIM interface can issue three interrupts:

1. SIMOKINT for the normal character interrupts
2. SIMERRINT for the parity, overrun, T=0 complete, and work waiting timer interrupts
3. SIMININT for the SIM presence interrupt (auto power down).

7.6.5 Clock Control

All SIM related clock signals[1)] are controlled via the **SIMCTRL** register. After system reset all clocks are off and must be enabled by the MCU during initialization of the SIM interface.

The SIM UART is initialized by enabling the SIM UART clock by setting **SIMCTRL.UARTON** and enabling the SIM card interface by setting **SIMCTRL.SIMEN**. By doing this, the UART enters RX mode and is ready for Answer To Reset (ATR). Initializing the UART with **SIMEN** does not affect the SIM clock, SIM V_{cc}, SIM RST or SIM I/O. Before entering low power mode:

1. Disable the SIM card (**SIMEN** = 0)
2. Turn off the UART clock .

1) Refer to **System Integration** Clock domain on **Page 144**.

The bit **SIMCTRL.UARTON** and the register **SIMBRF** control the clock used internally by the SIM UART. **UARTON** enables the clock controlling the UART and the T=0 controller and can be set to zero between commands to save power.

The value of **SIMBRF** (SIM Bit Rate Factor) directly controls the baud rate used by the UART. This factor can be used if the SIM during ATR lets the ME (mobile equipment) know that it supports enhanced speed mode – D = 8, F = 512. By changing the BRF, virtually any other combination of D and F can be used. After a successful Protocol and Parameter Selection the ME can switch to the enhanced speed and use this speed during the rest of the session.

The BRF is calculated by using the following equation:

$$BRF \quad \frac{E}{4D} \tag{46}$$

During ATR the SIM card uses the default values D = 1 and F = 372 which result in BRF = 93. This value is automatically loaded into **SIMBRF** when the peripheral is reset (refer to **RST_CTRL_STA** register), but **not** when the SIM interface is disabled by resetting **SIMCTRL.SIMEN**.

The value of the BRF must be an integer and appropriate to the ISO 7816 and GSM specification, that is, BRF = 16 (10_h) for F = 512, D = 8 and BRF 8 (08_h) for F =512, D = 16.

The clock signal feeding the SIM card can be switched on and off by the bit **SIMCTRL.SIMON**. This is used when starting an ATR sequence and between commands if the SIM supports the clock stop mode. Some Phase 2 SIMs may require the clock to stop at a high state. If this is the case, the bit **SIMCTRL.CLKHIGH** must be set prior to a clock stop.

*Note: If card deactivation is done in SW, **CLKHIGH** must be reset before stopping the clock. If card deactivation is done automatically, the setting of this bit is ignored.*

Because PMB7880 supports low power mode in which the 13 MHz reference oscillator can be switched off, the 3.25 MHz SIM clock may not be available.

For SIM cards, which do NOT support clock stop operation, the reference oscillator and the clock operating the SIM interface must stay switched on. In this case an 1.08 MHz clock generated from the 13 MHz clock can be used instead of the 3.25 MHz clock. This 'sleep' clock is selected by setting bit **SIMCTRL.CLKSEL**.

Note: The 'sleep' clock must not be used during communication with the SIM card.

7.6.6 SIM Activation and Deactivation

Prior to an ATR:

- The SIM interface must be enabled
- The normal speed must be selected.

Next, the UART ready interrupt and UART parity interrupt must be enabled by setting bit **SIMIRQEN.ENOKINT** and **SIMIRQEN.ENPAR**. UART overrun interrupts can be enabled when debugging the ATR state machine. During ATR, the HW controlled T=0 protocol must be disabled (default after reset).

The SIM interface is now ready to receive the initial character TS given the card has successfully been reset. The card is reset by doing the following steps:

1. Switch on SIM V_{cc} by setting bit **SIMCTRL.SIMVCC**. SIMVCC is inverted on the pin for control of an external PNP transistor. If the ME supports both 5 V and 3 V Smart cards, the proper voltage must be selected and controlled via one of the I/O ports.
2. Enable the SIM I/O line by setting **SIMCTRL.SIMIOL**. The I/O line is now tristated.
3. Switch on the SIM clock by setting **SIMCTRL.SIMON**. An internal reset card now answers within 40 000 clock cycles.

If the card is with active low reset, it does not answer until the RST signal goes high. This is controlled by setting bit **SIMCTRL.SIMRST**.

Note: According to ISO/IEC 7816-3 - Amendment 2 (1997), internal reset cards are no longer produced. However, old versions may still be in use.

By default, the interface expects the SIM to use direct convention (**SIMCTRL.INCON** = 0). If this is the case, the TS character causes an interrupt on SIMOKINT, and **UARTOK** is the only bit set in **SIMSTATUS**. If the SIM uses inverse convention instead, the TS character causes an interrupt on SIMERRINT and **SIMSTATUS.PARINT** is set.

When parity error detection signalling on received characters is disabled (**SIMCTRL.ERROFF** = 1), no error signal is generated on the I/O-line upon parity error detection. The received TS character can be read from **SIMRX** and should be used by MCU to determine the character coding convention used by SIM.

Enabling of parity error signalling on received characters (**SIMCTRL.ERROFF** = 0) and, if needed, selection of inverse convention (**SIMCTRL.INCON** = 1) must be done within two ETU after SIMOKINT or SIMERRINT interrupt.

If the UART overrun interrupt is enabled the MCU must always read the **SIMRX** register even if the data received is not needed. If not read, an overrun interrupt is generated instead of a UART ready interrupt.

SIM deactivation is done by executing the following steps:

1. If the clock is stopped at high level switch it back on (**SIMCTRL.SIMON** = 1) and set clock stop to low level (**SIMCTRL.CLKHIGH** = 0).
2. Set RST to low level (**SIMCTRL.SIMRST** = 0).
3. Stop the SIM clock (**SIMCTRL.SIMON** = 0).
4. Set SIM I/O to low level (**SIMCTRL.SIMIOL** = 0).
5. Remove SIM V_{cc} (**SIMCTRL.SIMVCC** = 0).

The interface can now be disabled.

Note: Forcing the I/O line to LOW while the SIM is transmitting may damage the SIM. It should always be done as above.

7.6.7 Initialization Sequence Overview

The following gives the programming sequence for initializing the SIM interface. This should be read in conjunction with the **SIMCTRL** register description.

1. Make sure that bit **SIMCTRL.SIMRST** is not set.
2. Configure CCIOSW corresponding to hardware (only required for 5V Smart Card) by programming bit **SIMCTRL.SMCIOSWACT**, and programming the port logic of SMCIOSW.
3. Enable SIM interface (**SIMCTRL.SIMEN** = 1).
4. Enable SIM UART (**SIMCTRL.UARTON** = 1).
5. Configure SIM interrupts for ATR procedure.
6. Start driving the external interface by programming the following:
   a) Voltage on (**SIMCTRL.SIMVCC** = 1)
   b) I/O line open drain (**SIMCTRL.SIMIOL** = 1)
   c) Enable SIM clock (**SIMCTRL.SIMCLK** = 1).

7. Wait 108 ETUs (~12,4ms).

8. Set CC_RST line high (**SIMCTRL.SIMRST** = 1).

9. Wait for ATR - copied by interrupt handler into some internal buffer; change convention if first character generates a parity error (**SIMCTRL.INCON** = 1)

10. Evaluate ATR and make a PTS procedure if high-speed mode is supported and desired: PTS is done sending in character mode (refer to **Section 7.6.9 SIM Character Mode (on Page 158)**) a PTS command and receiving the confirmation/rejection from the card. The usage of the T=0 controller is not possible for the PTS command. The bitrate factor may be changed in **SIMBRF**.

7.6.8 Automatic Power Down

Attention: The Automatic Power Down of the SIM card performed by the HW is not functional. A software work around has to be done. When the SIM card is removed, SW has to perform the power down.

The SIM interface has a feature that can be used in conjunction with mechanical detection of SIM presence. A mechanical switch can be connected to the SMC_IN pin. The SMC_IN pin can be used either by the Smart card interface (with some limitations) or the SIM interface.

SMC_IN can be used in several ways:

1. The status of pin SMC_IN can always be read on bit **SIMSTATUS.SIMDET**. Reading this bit does not affect any of the other bits that may have been set by UART and T=0 controller interrupts. This functions when connected to the Smart Card or the SIM card.
2. The SIM_IN_INT interrupt is enabled by setting bit **SIMIRQEN.EN**. The interrupt is issued each time the level of SMC_IN changes. The SIM card can now be disabled in three ways:
   a) The MCU can manually disable the SIM as described above. This method functions in conjunction with either the SIM card of the Smart card.
   b) The MCU can force an automatic power down by setting bit **SIMCTRL.SIMPDWN**. This also cause the **SIMCTRL.SIMEN** bit to be reset after an internal delay. This method only functions when one SIM card is being supported by the SIM card interface.
   c) If bit **SIMCTRL.APDWN** is set, any transition from HIGH to LOW on the pin SMC_IN causes an automatic power down. This method also only functions when one SIM card is being supported by the SIM card interface.

The automatic power down works even if PMB7880 is running on 32 kHz - although it will run considerably slower. **Figure 52** shows the power down sequence and timing (the values in brackets are for sleep mode, 32 kHz operation).

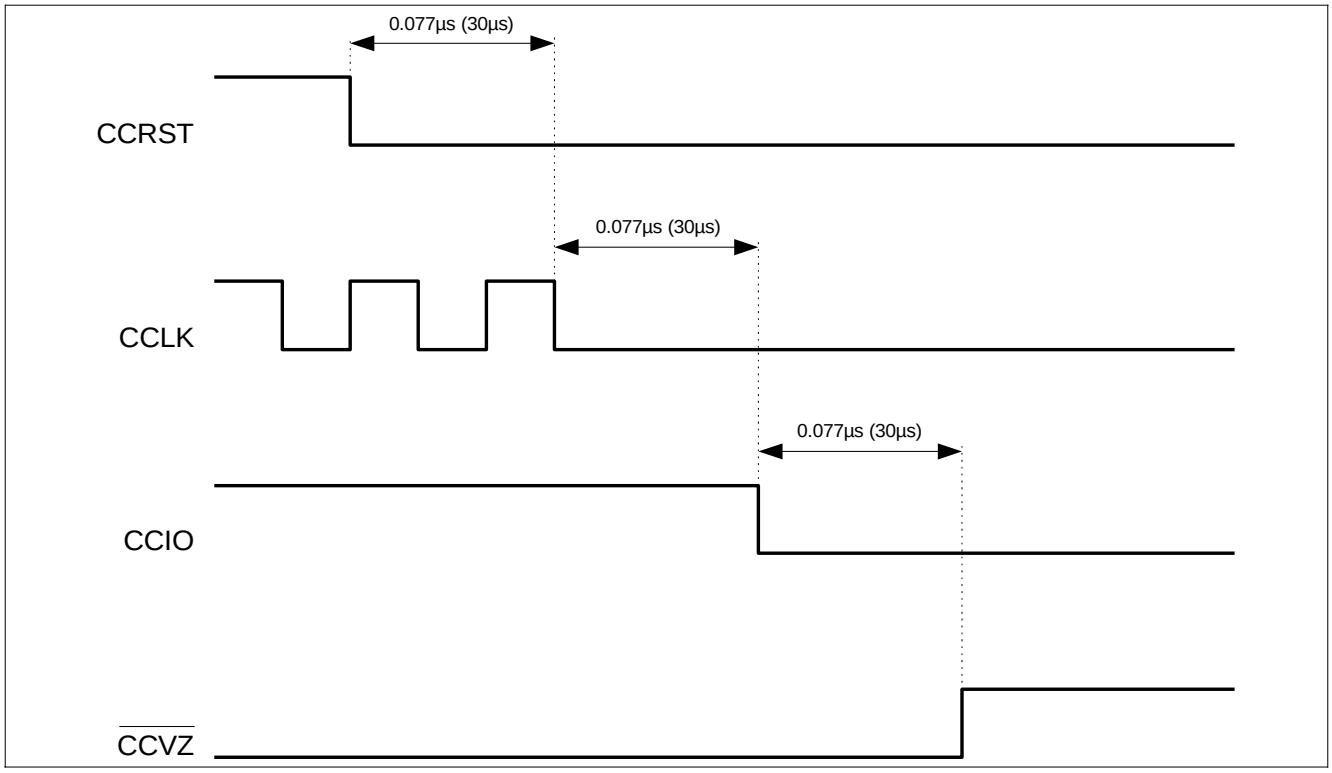


Figure 52 Timing of Automatic Power Down Sequence

Once triggered the SIM signals are held to LOW level until the MCU resets the SIM interface by writing 0 to **SIMCTRL.SIMEN**. Before setting **SIMEN**, the MCU must reset the **SIMCTRL** bits **SIMVCC**, **SIMON**, **SIMIOL** and **SIMRST**. This ensures LOW level on the SIM signals when auto power down mode is re-initialized.

Note: The SIMININT interrupt is generated, even if the SIM has been powered down. This can be used to detect if the SIM has been removed or inserted.

7.6.9 SIM Character Mode

The mode used during ATR is called SIM Character Mode (SCM). In this mode, the SIM interface only ensures an error free character transmission/reception. Any overlaying protocol - such as the T=0 protocol - must be implemented in SW especially spacing between characters and timeouts. The SIM interface provides all the signals and data needed to implement such protocols.

Figure 53 shows the basic structure of a SIM character. This format is also used in T=0 mode.

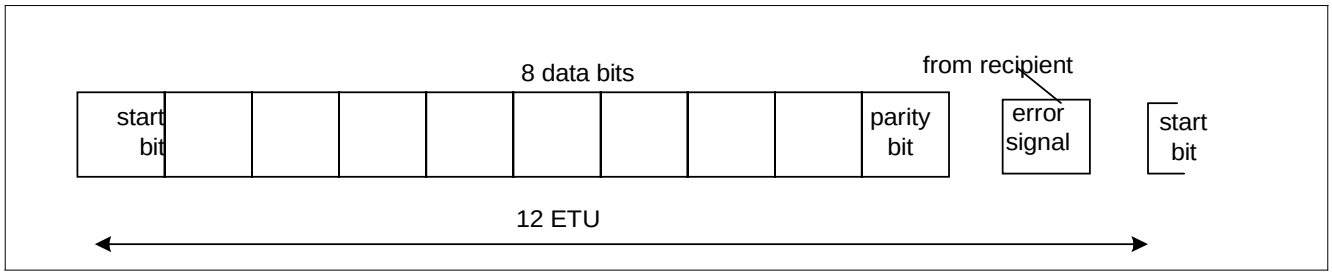


Figure 53 SIM Character Data Structure

7.6.9.1 Receiving Characters

The basic functionality of the SIM interface is described in **Section 7.6.1 Functional Overview (on Page 144)**. The following is more detailed description of the receive mode.

The SIM interface scans the I/O line for a start bit. When a start bit is detected, the following bits are shifted into an internal register and transferred to the **SIMRX** register when the last bit is received. The parity bit is checked after 9.5 ETU and an interrupt is generated. If no parity error is detected, the **SIMSTATUS.UARTOK** bit is set and a SIMOKINT is generated. In case of a parity error, the SIM interface generates an error condition lasting **one** eTu (from 10.5 ETU to 11.5 ETU) on the I/O line. The **SIMSTATUS.PARINT** bit is then set instead of the UARTOK bit and a SIMERRINT is generated. If a new byte is received before the previous one is read from the **SIMRX** register, the **SIMSTATUS.OVRRUN** bit is set overriding all other **SIMSTATUS** interrupt bits - again a SIMERRINT is generated. If the overrun interrupt is enabled the **SIMRX** register must always be read on an **UARTOK** interrupt. If an overrun error occurs, the SIM session cannot continue.

Error signalling on received characters can operate without enabling the **SIMSTATUS.PARINT** interrupt. By keeping bit **SIMCTRL.ERROFF** at 0, and by resetting **SIMIRQEN.ENPAR**, the UART detects and signals parity errors without interrupting the MCU. To disable error detection **ERROFF** must be set.

7.6.9.2 Sending Characters

Setting the SIM interface in TX mode is straight forward. A transmission is initiated by writing the byte to be transmitted to the **SIMTX** register. The data is shifted out with the LSB first at a bit rate of 8.73kbaud as a default. Inverse convention can be chosen by setting the **SIMCTRL.INCON** bit, thus inverting the polarity and bit order of the data byte.

The interface generates the parity bit on the I/O line at the end of the data byte, according to the **INCON** bit. After 11 ETUs the status of the I/O line is checked for the receiver status. If no parity error is reported by the SIM (the I/O line is HIGH), the **SIMSTATUS.UARTOK** bit is set indicating that the interface is ready to accept a new byte to be transferred. If the SIM reports a parity error (the I/O line is LOW), the **SIMSTATUS.PARINT** bit is set and the data byte is retransmitted until a successful transmission has been performed. A parity error interrupt is generated

before each retransmission. When a byte has been transmitted, the UART automatically enters RX mode. A new byte can be written to **SIMTX** right after the previous byte has been transmitted successfully, without considering the character spacing. The SIM UART ensures a character spacing of 12 ETUs when the MCU transmits characters continuously. This is however not the case when changing from RX- to TX-mode. After receiving a character, the MCU must wait for at least 2.5 ETU before writing to **SIMTX**.

If the spacing of 12 ETUs between two successive characters is too tight, extra spacing can be programmed in the **SIMTXSPC** register in steps of one ETU.

Similar to error detection and signalling in RX-mode, character repetition can be enabled (**SIMCTRL.RPTOFF** = 0) while the **PARINT** interrupt is switched off. Character repetition is disabled by setting **RPTOFF**.

If a new byte is written to **SIMTX** before the previous byte has been transmitted, the **SIMSTATUS.OVRRUN** bit is set.

7.6.10 SIM T=0 Protocol Mode

When setting **SIMBRF** to 16 (corresponds to D = 8 and F = 512), the MCU interrupt load from the SIM interface increases approximately 6 times. To lower the interrupt-load, the T=0 protocol has been implemented in HW. This unit is called T=0 controller.

7.6.10.1 Data Fetch Instructions

Data fetch instructions all receive data from the SIM card. The example below describes a READ BINARY instruction where the SIM sends 100 bytes. In receive mode, the header must be written by the MCU.

The max number of bytes that can be transferred in a data fetch command (refer to ISO 7816-3, section 8.2.1 and ISO 7816-3 - Amendment 2 - 2ed Edition, section 9.1) are 256 (P3 = 00).

7.6.10.2 Data Write Instructions

Data write instructions all transfer data to the SIM card. The example below describes an UPDATE BINARY instruction where the SIM receives 100 bytes.

The max number of bytes that can be transferred in a data fetch command (refer to ISO 7816-3, section 8.2.1 and ISO 7816-3 - Amendment 2 - 2ed Edition, section 9.1) are 255 (P3 = FF).

7.6.10.3 Work Waiting Timer (WWT)

As indicated above parity errors are handled by the T=0 controller during execution of an instruction. Additionally, the controller contains a timer unit to detected if the distance between received characters exceeds the Work Waiting Time (WWT) defined during ATR. WWT is always checked between the leading edges of any character sent by the card, and the previous character.

The timer is programmed through 24 bits in registers **SIMCHTIMERx** (SIM Character Timer) with a base resolution of 1 ETU.

The timer function is enabled by setting bit **SIMIRQEN.ENCHTIMER** meaning that the character timer interrupt is enabled when the SIM card interface block is operated in SIM T=0 protocol mode.

In T=0 mode, the character timer behaves as a WWT. The T=0 controller starts/restarts the character timer with the value defined in **SIMCHTIMERx**:

- When the timer is enabled with **ENCHTIMER**.
- Whenever a procedure byte (NULL, ACK or SW1) is received (also at the reception of retransmitted procedure bytes). If a character or a procedure byte has not been received before the expiration of the timer, a timeout interrupt is generated.
- Whenever a character is received (also at the reception of retransmitted characters) by the T=0 controller. If a character or a procedure byte has not been received before the expiration of the timer, a timeout interrupt is generated.
- Whenever a header or body character is transmitted (also at retransmission of characters) by the T=0 controller. If a procedure byte has not been received before the expiration of the timer, a timeout interrupt is generated.
- After the WWT value has been written to the **SIMCHTIMERx** registers.

The character timer is stopped without causing a timeout interrupt at the reception of the status byte SW2 (resulting in a command end interrupt) and at deactivation of the SIM – both with respect to MCU controlled deactivation but also when **SIMCTRL.SIMEN** is programmed with the value 0 (disabled following a automatic power down for example).

The character timer is also stopped when a character timeout interrupt is generated.

A character timeout results in a SIMERRINT interrupt causing the **SIMSTATUS.CHTIMEOUT** bit to be set. A character timeout does not change the behavior of the ongoing T=0 instruction handling and it is the responsibility of the MCU to determine whether further instruction operation can be accepted.

Note: A character timeout interrupt is also generated during transmission of subsequent characters, if the time between two characters exceeds the defined timeout limit. (This is not strictly required by ISO/IEC 7816-3, but can be used to increase the robustness of the system)

7.6.10.4 Character Spacing

When receiving characters from the SIM, proper spacing (minimum allowed) is handled by the card.

The spacing after receiving a character and before starting to transmit, is ensured to be 2.5 ETU. However, this can be changed through the **SIMRXSPC** register, where this spacing is calculated in 1/16 ETU, so the default value of 40 equals 40/16 = 2.5.

When transmitting characters to the SIM, the T=0 controller ensures 12 ETUs between two consecutive transmitted characters. This spacing can be increased by programming the additional spacing in ETU to the **SIMTXSPC** register.

7.6.11 Connection of 5 V, 3 V, and 1.8 V SIM Cards

Because the pad supply voltage for E-GOLDvoice is limited to 3.0 V, an external level shifter is required to connect 5V Smart cards.

The E-GOLDvoice can also support 1.8 V SIMs, but in this case, both the SIM interface, and the Smart Card interface has a voltage of 1.8 V. If in such a case, the Smart card needs a higher voltage (3.0 V for example), then a level shifter is also required.

7.6.12 SIM Card Pads

During powering up the mobile it may happen, that the pads (Vdd2.x) are supplied earlier than the core (Vdd1.x). To avoid spikes on the SIM-interface due to a undefined reset status, the SIM interface pads have a special reset behavior.

If the chip reset ($\overline{\text{RESET\_IN}}$ = 0) is active, the following fixed logic states are driven at the SIM Card pads:

- CC_IO = 0

- CC_RST = 0
- CC_CLK = 0
- CC_VZ = 1.

In this state boundary scan data, which is latched into the boundary scan cells, does not appear at the SIM card pins.

7.7 RF Power Ramping

System Integration:

- Supply domain: VDD_LD1
- Chip internal interfaces:
  - Clock domain: clk_gsm_if
  - Bus domain: X-Bus
- Interrupt sources:
- Monitor Pins: Refer to **Section 9.7.10 Internal Signal Monitoring (on Page 435)**.

7.7.1 Introduction

The RF power amplifier (PA) needs analog control voltages to set the output power of the mobile. The PA Control Hardware provides the possibility, to set the output power of the mobile according to the GSM requirements to fulfill the power time template specification and the adjacent channel power specification for switching transients. For the PA adjustments an analog control voltage is supplied by the PMB7880 at the output pin PAOUT1. As shown in **Figure 54** the PA Control Hardware is connected to GSM System Interface units and to the Analog Control Registers (refer to **Section 7.4 Analog Control Registers (on Page 137)**).

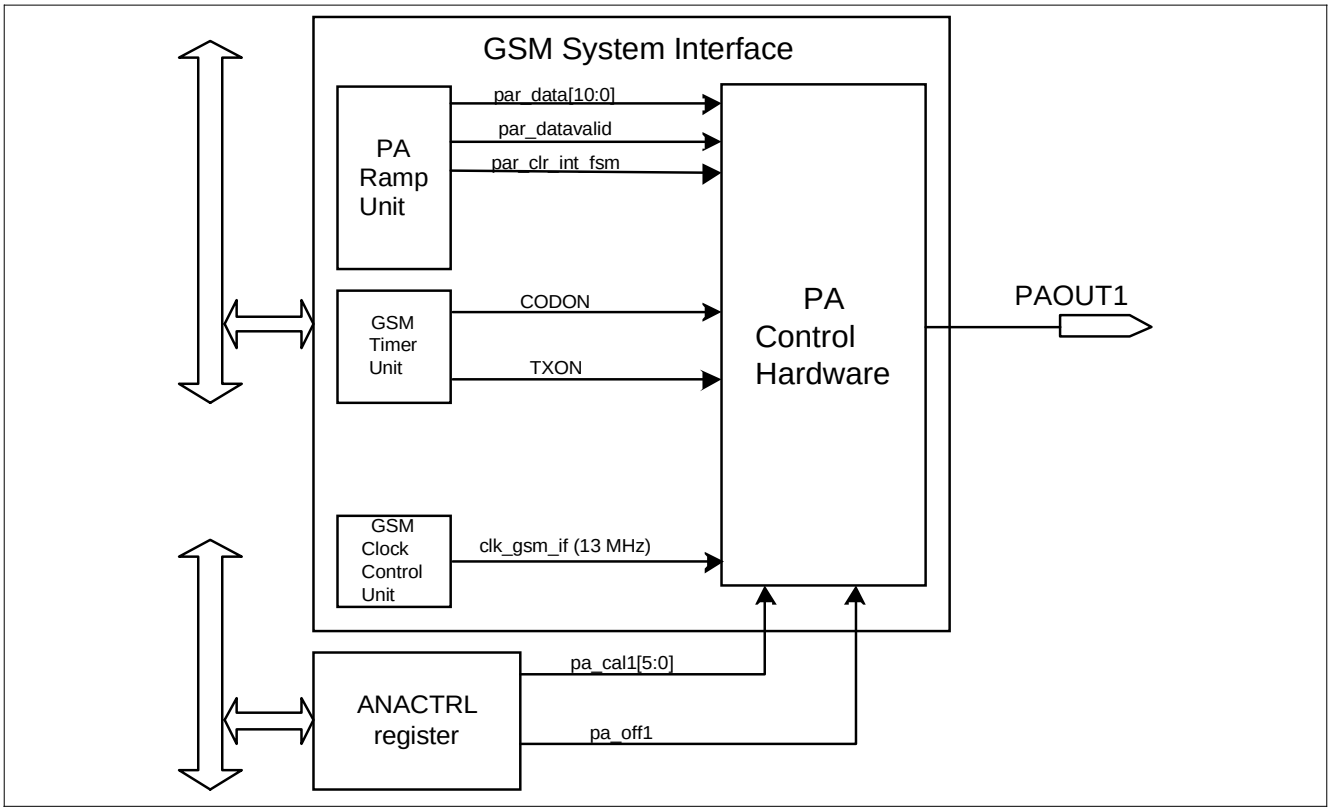


Figure 54 Power Amplifier Control Hardware System Overview

Note: The PA ramping data is stored by the controller within the RF RAM, refer to **Section 7.8.2 RF RAM**. The power ramping words are transferred from the PA Ramp Unit to the PA Control Hardware. *For information about the GSM CGU refer to **Section 7.2.1 Clock Generation Unit**.*

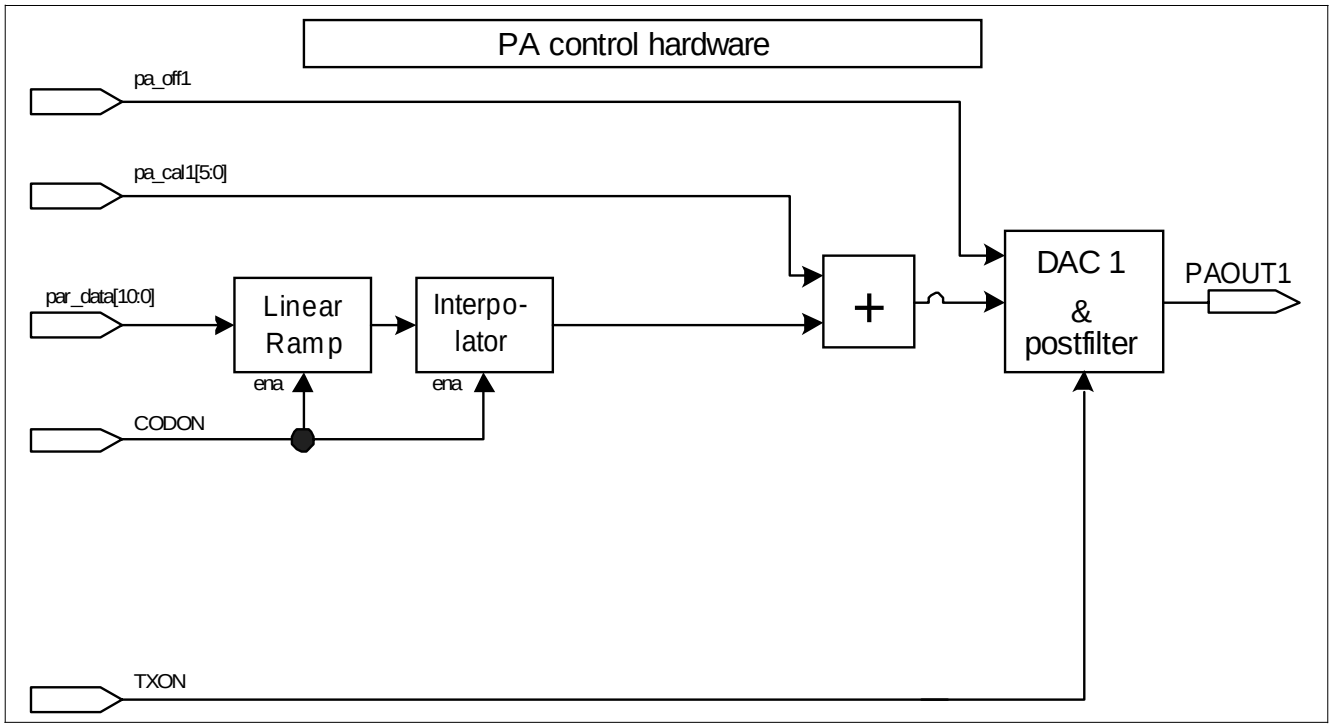


Figure 55 PA Control Hardware

An additional feature of the PA Control Hardware is the possibility to correct the PA power with a linear slope during the active part of the burst. The PA power ramping is basically divided into three periods: the power ramping at the start and end of a burst and the linear slope during the active part of the burst. Intermediate ramping between two consecutive burst is possible as well when the correct ramping shape is programmed by the controller. Internal details about the HW are shown in **Figure 55**. The units shown in the block schematic have the following functions:

- **Linear Ramp**Performs the linear ramping during the active part of the burst.
- **Interpolator**Interpolates the data rate to 6.5 MSamples/s.
- **Adders**Adds offset value for offset compensation. Refer to **Section 7.7.1.3.1 Digital Correction of Analog DC Offset (on Page 166)**.
- **DAC & filter**Digital-to-analog conversion and analog anti-aliasing filtering.

7.7.1.1 Power Ramping Shapes

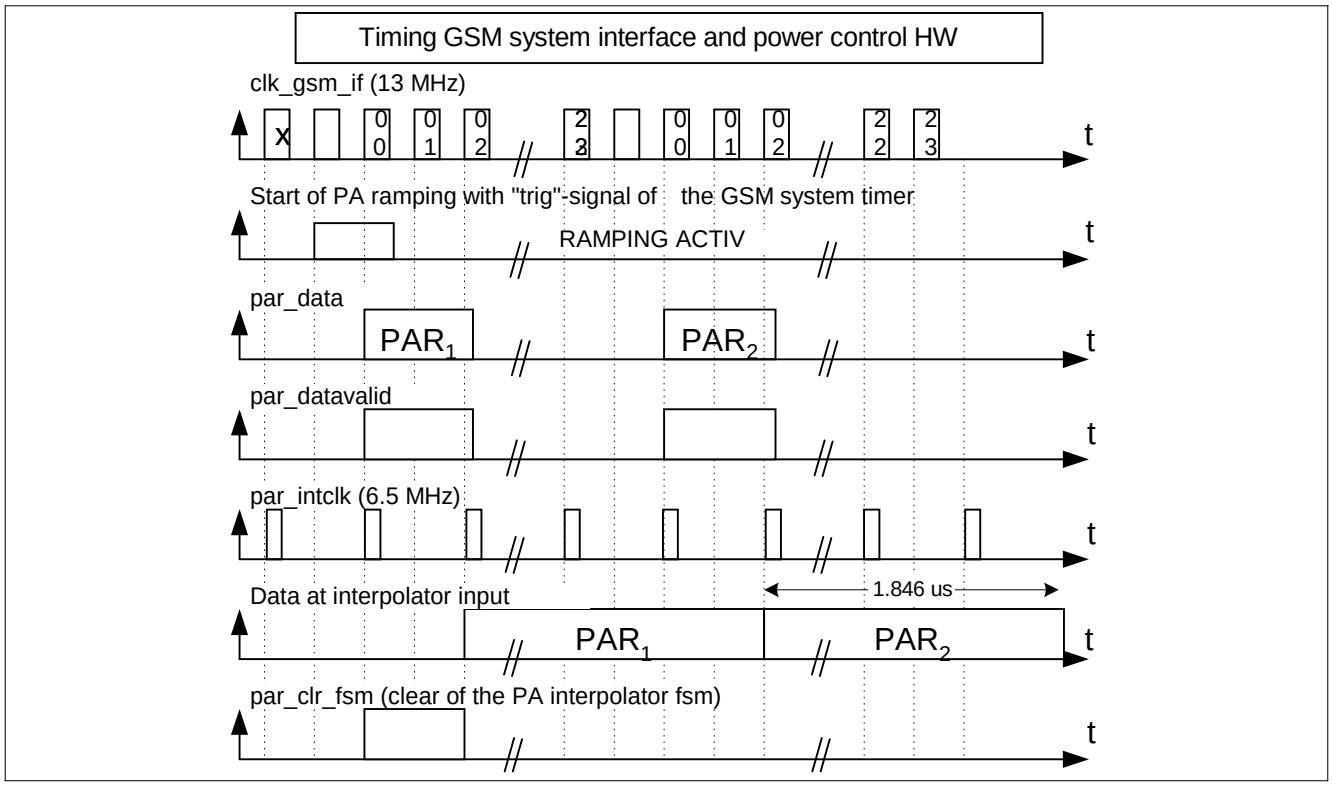


Figure 56 Power Ramping Shapes

Several types of power time templates are defined for the GSM system. The implementation of the rising and the falling part of a ramping curve is described in **Section 7.7.1.1.1 Up and Down Ramping (on Page 164)**.

One complete set of ramping data in the RF RAM of the RF Control Unit comprises 16 values for the ramping shape and one 11-bit value (word 17) for the step size of the linear power interpolation for the ramping during the active part of the burst. The structure of the data is described in the **Table 32 RF RAM Partitioning of the RF Control Unit (Type 1)**.

7.7.1.1.1 Up and Down Ramping

In the RF RAM 16 specific RAM locations PAR[1:16] have to be loaded with 11-bit values by the controller. This represent the shape of the desired ramping curve as digital ramp samples and the PAINC value as word 17 (refer to **Section 7.8 RF Control**). The data is transferred from the RF Ramping Unit to the PA Ramping Hardware with the rate of 6.5 MHz/12 = 541.67 kSamples/s.

With the trigger mechanism described in **Section 7.8** the ramping process is started and the 16 ramp samples and the PAINC value are transferred to the PA Ramping Hardware.

Note: The data path delay from the RF Ramping Unit to the interpolator is approximately 2 symbols (7.385 µs).

7.7.1.1.2 Linear Power Ramping during Active Part of Burst

The output power of a PA without amplitude feedback control usual is changing during the active part of the burst. To compensate this deviation the RF Control Unit has to be able to correct this effect. The shape of the active part of an GMSK burst can be corrected by the linear slope.

The 18th value of a ramping data set is the value PAINC (11 bits, 0...2047) and defines the amount of increment cycles ($f_{clk\_gsm\_if}$/24 = 541.67 kHz) during the active part of the burst after which the output value of the multiplexer in the PA Ramping Hardware is incremented by 1 as shown in **Figure 57**.

Note: With PAINC = 0 the linear ramping functionality is deactivated. One $f_{clk\_gsm\_if}$/24 cycle after the value PAINC is transferred to the PA Ramping Hardware the linear ramping is started if the value PAINC is any value except 0.

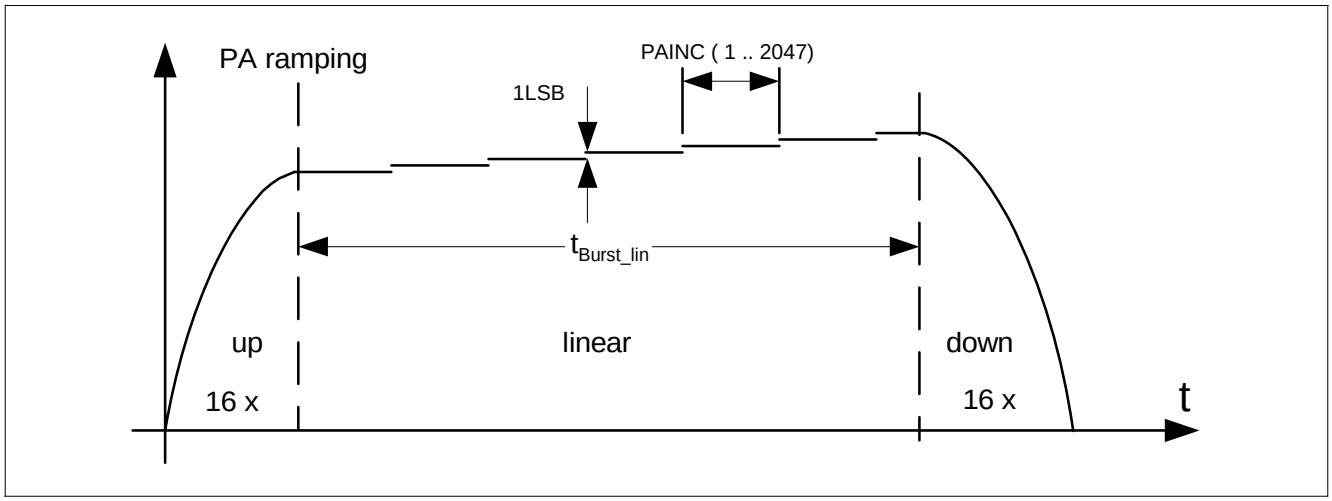


Figure 57 Linear Power Ramping during Active Part of the Burst

7.7.1.1.3 Consecutive Bursts

When the 16 ramp samples and the value PAINC have been transferred and the linear ramping is deactivated the latest voltage output on PAOUT remains constant until a new set of 16 samples representing the falling part of the ramping shape is to be processed.

If the linear ramping is activated the output value is changing in steps of one LSB. It is SW responsibility to start the following up/down ramping sequence with a correct start value to avoid a step in the power ramp.

Note: For the down ramping the value PAINC should be set to 0 in order to avoid unwanted changes of the PA output power when the PA power control is regarded as switched off.

The start value of the down ramping section can be calculated as follows:

$$y = \text{floor}\left\{\frac{t_{burstlin} \cdot 541.6\ \text{kHz} - 1}{\text{PAINC}}\right\} + x \qquad (47)$$

where:

x = end value of the up ramping sequence

y = start value of the down ramping sequence

$t_{burstlin}$ = length of the active part of the burst in seconds.

7.7.1.2 Interpolation

The interpolator increases the sampling frequency from 541.6 kHz to 6.5 MHz thus, providing a 11-bit sample every 154 ns to the D-to-A converter.

7.7.1.3 Programming Sequence for Power Ramping

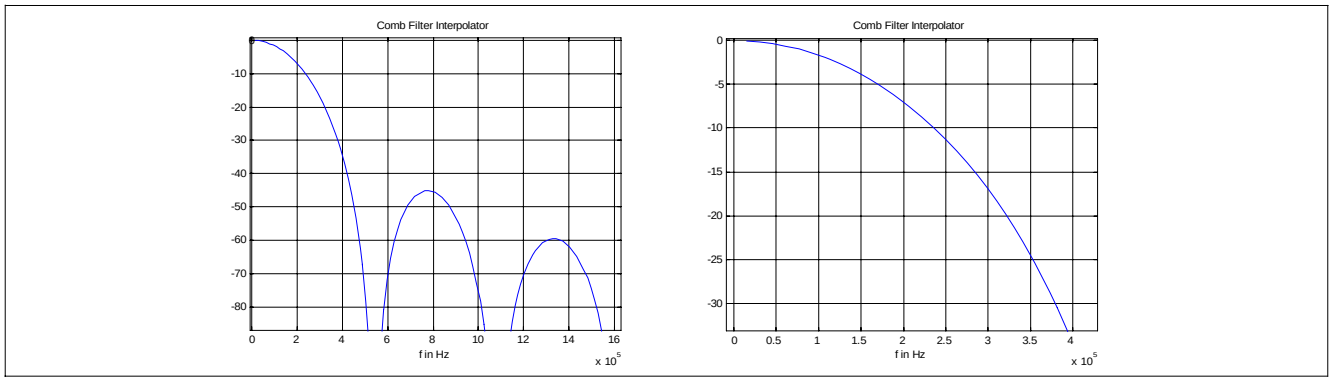


Figure 58 Programming Sequence for Power Ramping

The MCU procedure to get the PA ramping path working after a boot up of the PMB7880:

The entire control of the PA ramping is done by the GSM system interface.

1. Refer to **Section 7.8 RF Control** for programming the power ramping data into the RF control RAM.
2. Enable the signal CODON by programming the GSM timer unit in **Section 9.6 GPT 1 and 2** to switch on the digital power ramping HW.
3. Enable TXON by programming the GSM timer unit in **Section 9.6** to switch on the analog power ramping HW.
4. Start the transfer of the power ramping telegrams.
5. To disable the PA ramping path in a correct way, first switch off TXON, then switch off the CODON.

Note: The CODON signal enables the digital blocks; the TXON signals enables the analog blocks of the TX modulator as well.

7.7.1.3.1 Digital Correction of Analog DC Offset

The analog DC offset of the analog part of the PA Control Hardware is measured by the measurement interface. For a digital correction of the analog DC offsets digital adders are used. The advantage of separate adders compared to a software-only solution is that standard tables for each mobile can be used and only two calibration values, that is, the value of **ANA_CTRL2 (on Page 139).PA_CAL1** has to be measured and set.

The measurement procedure is described in **Section 7.3.7 Modulator Unit Offset Measurement TXOFI and TXOFQ (on Page 122)**. The correction values **PA_CAL1**

(= $0..63_D$) are added to the output of the PA Multiplexer Unit.

Note: The input value to the adder must not exceed the range of $0..1984_D$.
This is 2047_D - 63_D = 1984_D for a 6-bit wide correction word and leaves the necessary headroom for the DC correction.

7.7.1.4 D-to-A Conversion and Post Filtering

The digital-to-analog converters are R-string type. After interpolation the signal spectrum will be repeated around multiples of the sampling frequency (6.5 MHz). These frequency components have to be suppressed by a subsequent lowpass reconstruction filter. This post filters are active 1rd order filters with a cutoff frequency of 300 kHz. The resulting analog voltage is shown on the output pin PAOUT1.

The nominal linear output voltage range for PAOUT1 is U_{drop-} to VDD - U_{drop+}. (**Figure 59**, $I_{PAOUT} > 0$). More details for different currents are listed in **Table 184 Specification of PAOUTOF1 (on Page 550)**. $I_{PAOUT} > 0$ denotes a current flow out of the pin PAOUT1 and $I_{PAOUT} < 0$ vice versa. Therefore, the actual linear output range depends on I_{PAOUT}, VDD, and the dropout voltages as shown in **Figure 59**.

Note: For transient-free power ramps it is a precondition to stay within the linear operating range of the PAOUT1 pin during the HF power amplifier unit is sensitive for the PAOUT1 output voltage.

For **ANA_CTRL2 (on Page 139).PA_OFF1** = 0 there's zero (no) voltage bias. For **PA_OFF1** = 1 the output buffer is biased by nominal -50 mV for offset adjustment. As the PMB7880 can only output positive voltages, only digital values above up to 40 ensure an output voltage in the linear operating range in the case of **PA_OFF1** = 0.

The signal TXON is used to switch on the analog voltage supply of the DAC1 and the postfilter. The PAOUT1 output is forced to 0 V during TXON = 0.

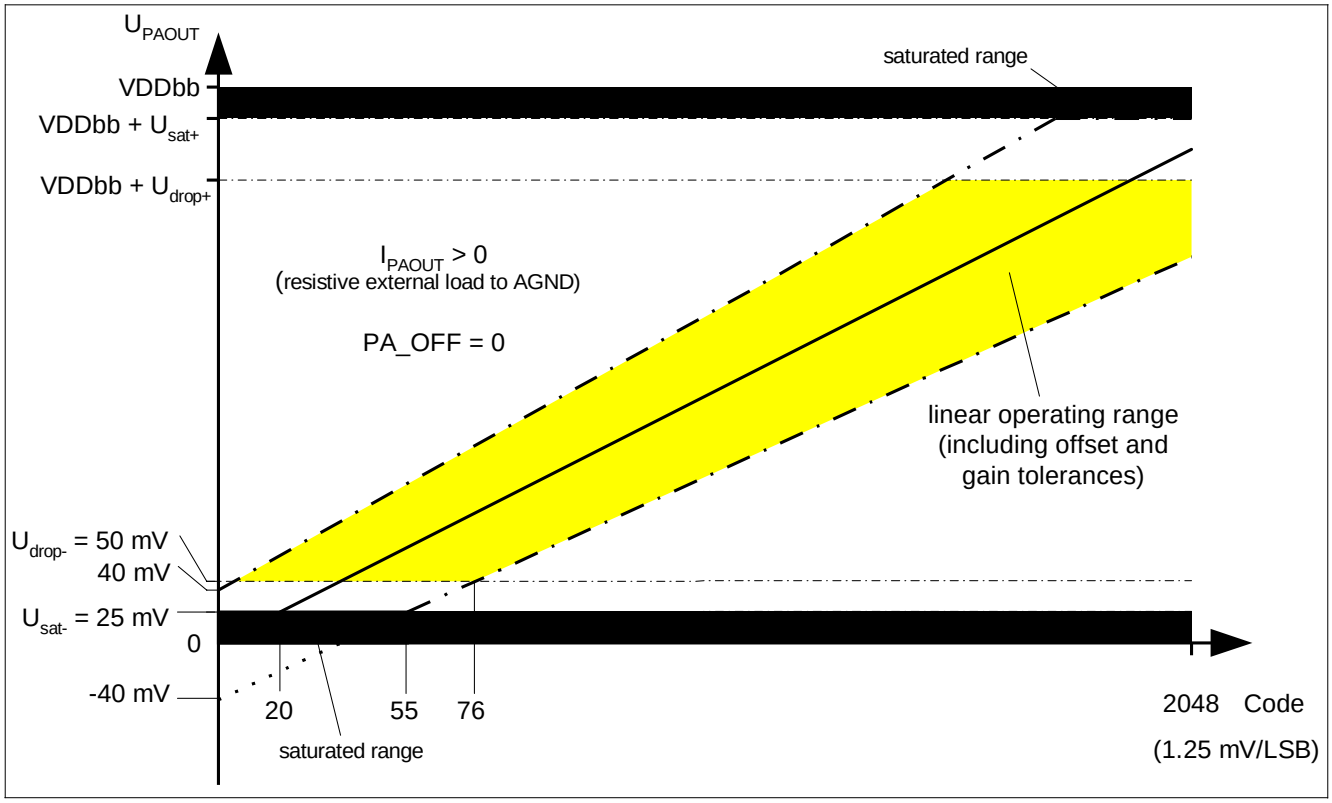


Figure 59 Operating Range of PAOUT1 Output after Offset Calibration

7.7.2 Application Notes

7.7.2.1 How to Program a Power Ramp Signal, after a POWER Up

1. Program the power ramping sequences (up and down ramping), the PAINC values into the RF RAM (refer to **Section 7.8.2 RF RAM (on Page 170)**).
2. Program the offset calibration information in **ANA_CTRL2 (on Page 139).PA_CAL1** and **ANA_CTRL1 (on Page 138).PA_OFF1**.
3. Program the GSM timer signals CODON, TXON (refer to **Section 7.9.2 GSM Clock Control Unit**).
4. Program the trigger signals for the PA ramping sequence telegrams (refer to **Section 7.8.2 RF RAM**).
5. After down ramping, switch TXON signal off.
6. Switch CODON signal off, the PAOUT1 pin is now in high impedance.

Notes

1. *TXON and CODON are enabling the TX modulator hardware as well (refer to **Section 8.4 GMSK Modulator**).*
2. *For offset self calibration of PAOUT1 refer to **Section 7.4.3 Analog Control Register 2 (on Page 139)**.*
3. *The CODON signal must be enabled before the TXON signal is switched active to get a valid analog output signal with the rising edge of TXON.*
4. *The ramping sequence must not be started within 2 symbols (7.385 µs) after CODON is set active.*
5. *The last word of the ramp down sequence is D-to-A converted until the falling edge of TXON. The last word of the ramp down sequence is again D-to-A converted starting with the rising edge of TXON until the first word of the ramp up sequence is received.*
6. *To reduce power consumption, the clock for the PA Ramping Hardware is only enabled when needed (during a valid burst by the CODON signal). For correct operation of the PA Ramping Hardware the signal CODON must be switched on in advance to the beginning of the power ramp period and must remain active until the end of the ramp down period.*
7. *The user must ensure that RF chips, such as the PA, are not powered on until a valid PAOUT1 signal is available.*

7.7.3 System Register

Module Identification

TID

GSM System Interface Identification Register (refer to **TID (on Page 185)**).

7.8 RF Control

System Integration

- Supply domain: VDD_LD1
- Chip internal interfaces:
  - Clock domain: Refer to **Section 7.2.1.3 Sub-System Clocks and Enables (on Page 67)** and see **Figure (on Page 71)**.
  - Bus domain: X-Bus
- Interrupt sources:
- Monitor Pins: Refer to **Section 9.7.10 Internal Signal Monitoring (on Page 435)**.

7.8.1 Introduction

The RF Control Unit generates the control information which has to be provided to the RF chip set and the analog part. This information is provided by telegrams of 8, 16, or 24 bits via the RF Interface. The block diagram of the RF Control Unit is shown in **Figure 60**.

*Note: For information about the GSM CGU refer to **Section 7.2.1 Clock Generation Unit**.*

By means of the signal 'trig_start' coming from the GSM Timer Unit the transfer of a telegram or power ramp stored in the RF RAM of the RF Control Unit is initiated. Which telegram will be transferred is coded by the 6 signals TRIG[5:0].

The transmission of telegrams can also directly be programmed via the X-Bus interface similar to a SSC interface. This feature is especially suited for the initialization of the RF devices.

Within the RF Control Unit the PA power ramping data is generated as well. The interface to the PA Control Hardware are two 11-bit wide buses with corresponding control signals. Via this interface the power ramping data and additional a PA bias control signal (DCPA) are provided.

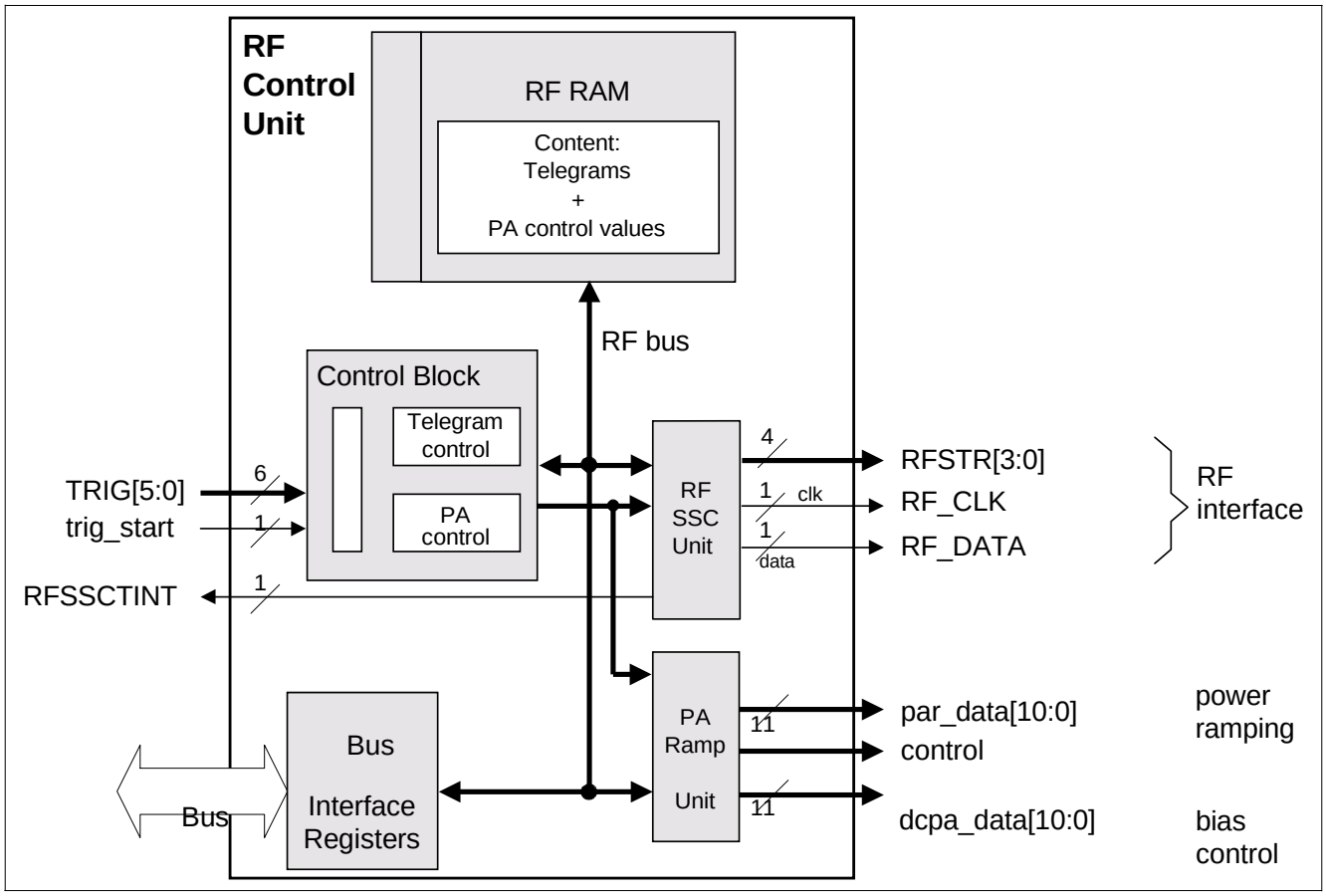


Figure 60 RF Control Unit

7.8.2 RF RAM

The RF Control Unit comprises a dual port RF RAM of 448 * 11 bits which is programmable via the X-Bus. The RAM address range is given in **Table 32** and **Table 33**. The GSM System Interface base address is defined in **Table 136 Address Mapping of X-Bus Peripherals**. The RF RAM can store:

- With RF RAM Partitioning Type 1 (refer to **Table 32**):
  - Up to 40 telegrams (refer to **Table 34 Structure of Telegrams (on Page 173)**), each consisting up to 24 bits of data and 9 bits of control information
  - 16 power ramp sequences, each consisting of 16 values of 11 bits, one 11 bit value for controlling the linear ramping during the active part of the burst, and one 10 bit value **DCPAdata (on Page 174)** for controlling the power amplifier's bias (refer also to **Section 7.7 RF Power Ramping (on Page 162)**)
- With RF RAM Partitioning Type 2 (refer to Table 33 on page 172):
  - Up to 112 telegrams (refer to **Table 34**) each consisting up to 24 bits of data and 9 bits of control information.

The partitioning type is selected in the RF Control Register 1 (**RFCON1 (on Page 174)**).

Note: RF RAM Partitioning Type 2 is especially suited for those cases where the internal power ramping and the output PAOUT1 is not needed, for example, if an external modulator with separate PA control is used. Usually, in this case more telegram capacity is needed to control the external modulator.

Table 32 RF RAM Partitioning of the RF Control Unit (Type 1)

Physical address[1)] (Hex)	Name	Function
	RAM entries for telegrams T1 to T40	
000	T1	Definition of telegram 1:Word 1: 000_HWord 2: 002_H (trigger value: 8_D)Word 3: 004_HWord 4: 006_H
008	T2	Definition of telegram 2 (trigger value: 9_D)
...	...	...
130	T39	Definition of telegram 39 (trigger value: 46_D)
138	T40	Definition of telegram 40 (trigger value: 47_D)
	RAM entries for power ramp sequence 1 start at 140_H (trigger value: 48_D)	
140	DCPAdata1	PA bias data for PA ramp 1 (refer to **DCPAdata (on Page 174)**)
142	PAR1(1)	1. power ramp value [10:0]
...	...	...
160	PAR1(16)	16. power ramp value [10:0]
162	PAINC1	1st incremental value for linear power ramping [10:0]
	RAM entries for power ramp sequence 2 start at 164_H (trigger value 49_D)	
164	DCPAdata2	PA bias data for PA ramp 2 (refer to Table DCPAdata)
166	PAR2(1)	1. power ramp value [10:0]
...	...	...
184	PAR2(16)	16. power ramp value [10:0]
186	PAINC2	2nd incremental value for linear power ramping [10:0]
	RAM entries for power ramp sequence 3 start at 188_H (tigger value 50_D)	
	RAM entries for power ramp sequence 4 start at $1AC_H$ (tigger value 51_D)	
	RAM entries for power ramp sequence 5 start at $1D0_H$ (tigger value 52_D)	
	RAM entries for power ramp sequence 6 start at $1F4_H$ (tigger value 53_D)	
	RAM entries for power ramp sequence 7 start at 218_H (tigger value 54_D)	
	RAM entries for power ramp sequence 8 start at $23C_H$ (tigger value 55_D)	
	RAM entries for power ramp sequence 9 start at 260_H (tigger value 56_D)	
	RAM entries for power ramp sequence 10 start at 284_H (tigger value 57_D)	
	RAM entries for power ramp sequence 11 start at $2A8_H$ (tigger value 58_D)	
	RAM entries for power ramp sequence 12 start at $2CC_H$ (tigger value 59_D)	
	RAM entries for power ramp sequence 13 start at $2F0_H$ (tigger value 60_D)	
	RAM entries for power ramp sequence 14 start at 314_H (tigger value 61_D)	
	RAM entries for power ramp sequence 15 start at 338_H (tigger value 62_D)	
	RAM entries for power ramp sequence 16 start at $35C_H$ (tigger value 63_D)	

1) plus RAM base address

Table 33 RF RAM Partitioning of the RF Control Unit (Type 2)

Physical address[1] (Hex)	Name	Function
	RAM entries for telegrams T1 to T112	
000	T1	Definition of telegram 1:Word 1: 000_HWord 2: 002_H (tigger value 1_D)Word 3: 004_HWord 4: 006_H
008	T2	Definition of telegram 2 (no trigger value)
010	T3	Definition of telegram 3 (trigger value 2_D)
...	...	...
288	T98	Definition of telegram 98 (no trigger value)
290	T99	Definition of telegram 99 (trigger value 50_D)
...	...	...
370	T111	Definition of telegram 111 (tigger value 62_D)
378	T112	Definition of telegram 112 (tigger value 63_D)

1) Plus RAM base address

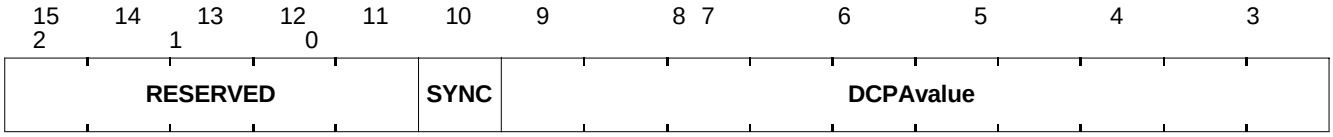
Table 34 Structure of Telegrams

Word	Bit	Function
1	0	**Multiple Telegram Transmission Control (Burst Mode)** 0 No following telegram is automatically transmitted after this telegram 1 The telegram i +1 is automatically transmitted after this telegram, that is, telegram i. Refer to **Section 7.8.4 Control Block (on Page 178)**.
	3:1	**RF SSC Unit Strobe Select Bits (SSCSB[2:0])** 000 RFSTR[0] is active during transmission of telegram i 001 RFSTR[1] is active during transmission of telegram i 010 RFSTR[2] is active during transmission of telegram i 011 RFSTR[3] is active during transmission of telegram i 100 Reserved 101 Reserved 110 Reserved 111 Reserved
	5:4	**RF SSC Unit Telegram Length Bits (TLB[1:0], Corresponds to SSCBM)** 00 8 bit long telegram 01 16 bit long telegram 10 24 bit long telegram 11 Not allowed
	6	**Reserved**[1]
	7	**RF SSC Unit Clock Frequency Bit (SSCFB)** 0 Telegram transfer with 6.50 MHz bit clock 1 Telegram transfer with 3.25 MHz bit clock
	8	**RF SSC Unit Clock Phase Select Bit (SSCPB)** 0 Shift transmit data on leading clock edge 1 Shift transmit data on trailing edge
	9	**RF SSC Unit Heading Control Bit (SSCHB)** 0 Transmit LSB first 1 Transmit MSB first
	10	**Reserved**[1]
2	7:0	**Telegram Data** IF (TLB[1:0] = 00) THEN bits [7:0] of telegram ELSIF (TLB[1:0] = 01) THEN bits [15:8] of telegram ELSIF (TLB[1:0] = 10) THEN bits [23:16] of telegram
	10:8	**Reserved**[1]
3	7:0	**Telegram Data** IF (TLB[1:0] = 01) THEN bits [7:0] of telegram ELSIF (TLB[1:0] = 10) THEN bits [15:8] of telegram
	10:8	**Reserved**[1]
4	7:0	**Telegram Data** IF (TLB[1:0] = 10) THEN bits [7:0] of telegram
	10:8	**Reserved**[1]

1) always set to ’0’.

7.8.2.1 DCPA Data Entry in RF Control Unit RAM

DCPAdata
DCPA Data Entry in RF Control Unit RAM



Field	Bits	Type	Description
DCPAvalue	9:0	rw	**DAC Value for DCPA Output** The DAC Value is equal to 2 times **DCPAvalue**. 000_H DAC Value = 0 ... ... $3FF_H$ DAC Value = $7FF_H$
SYNC	10	rw	**Synchronization of DAC with PA Ramp** 0 The DCPAvalue is getting valid with the beginning of the PA ramp it is associated with. 1 The DCPAvalue is getting valid after DELAY * 24 clk_gsm_if clock cycles after the beginning of the PA ramp it is associated with (refer to Table 32 on page 171). *Note: Setting SYNC = 1 and DELAY = 0 is identical to setting SYNC = 0 and DELAY to any arbitrary value.*
RESERVED	15:11	r	Reserved; these bits must be left at their reset values.

7.8.3 Synchronous Serial Interface (3-wire RF interface)

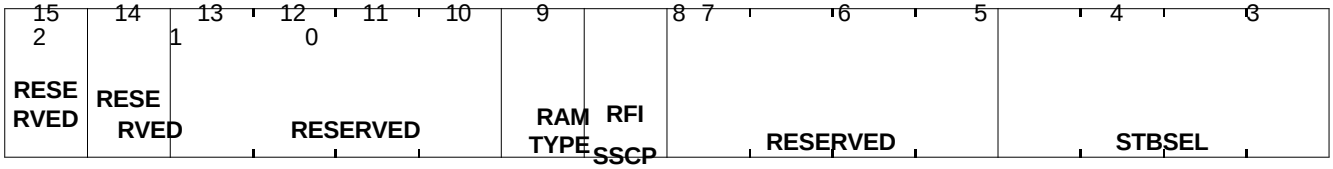
The Synchronous Serial Interface (RF SSC Unit) provides a high speed serial communication between the E-GOLDvoice and external peripherals like the RF ICs via a 3-wire interface. This interface is not mapped onto pads, only the internal connection between Base-band and RF inside E-GOLDvoice is used. For observing these signals monitoring signals are available.

The RF SSC Unit generates the strobe signals RFSTR[3:0], the clock signal RF_CLK and the serial data stream RF_DATA.

The format of the data stream which has to be transferred has to be provided in the RF control registers **RFCON1** and **RFCON2 (on Page 175)** of the RF Control Unit in advance to the data transmission. The register **RFCON1** stores the control information as far as these control information is identical for all transferred data streams. The RF control register 2 (**RFCON2 (on Page 175)**) stores the control information as far as these control information can vary from telegram to telegram. The data bits which are transferred by the RF SSC Unit are stored in the RF SSC Unit transmit buffer register (**RFSSCTB (on Page 177)**).

7.8.3.1 RF Control Register 1

RFCON1
RF Control Register 1 **Reset value: 0000_H**



Field	Bits	Type	Description
STBSEL	3:0	rw	**Strobe Select** For each bit **STBSEL**[i] the meaning is: 0 RFSTR[i] is low active 1 RFSTR[i] is high active
RFISSCP	8	rw	**SSC Clock Polarity of RF Interface** 0 Idle clock line is low, leading clock edge is low to high transition 1 Idle clock line is high, leading clock edge is high to low transition
RAMTYPE	9	rw	**RF Control Unit RAM Partitioning Type** 0 Type 1 is selected (refer to Table 32 on page 171) 1 Type 2 is selected (refer to Table 33 on page 172)
RESERVED	15:10, 7:4	r	Reserved; these bits must be left at their reset values.

7.8.3.2 RF Control Register 2

RFCON2

RF Control Register 2 **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSC EN	RESE RVED	SSC FB	RESERVED		SSC SB		RESERVED		SSC PB	SSC HB	SSCBM				

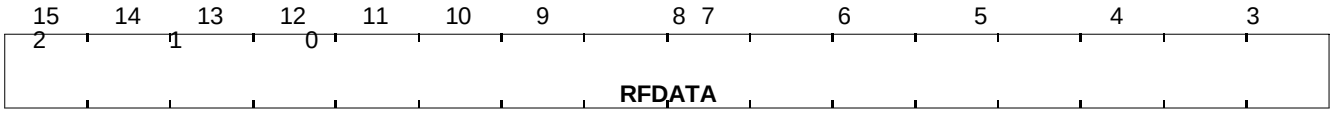
Field	Bits	Type	Description
SSCBM	3:0	rw	**RF SSC Unit Telegram Length Control** The telegram length is defined by **SSCBM** + 1. 0000 Do not use 0001 Telegram length is 2 bits 0010 Telegram length is 3 bits ... ... 1111 Telegram length is 16 bits
SSCHB	4	rw	**RF SSC Unit Heading Control** 0 Transmit LSB first 1 Transmit MSB first
SSCPB	5	rw	**RF SSC Unit Clock Phase Control** 0 Shift transmit data on leading clock edge 1 Shift transmit data on trailing edge
SSCSB	10:8	rw	**RF SSC Unit Strobe Select** 000 RFSTR[0] is active during transmission 001 RFSTR[1] is active during transmission 010 RFSTR[2] is active during transmission 011 RFSTR[3] is active during transmission 100 Reserved 101 Reserved 110 Reserved 111 Reserved

Field	Bits	Type	Description
SSCFB	13	rw	**RF SSC Unit Clock Frequency** 0 Telegram transfer with 6.50 MHz bit clock 1 Telegram transfer with 3.25 MHz bit clock
SSCEN	15	rw	**RF SSC Unit Enable** 0 Transmission is disabled 1 Transmission is enabled
RESERVED	7:6, 12:11, 14	r	Reserved; these bits must be left at their reset values.

7.8.3.3 RF SSC Transmit Buffer

RFSSCTB

RF SSC Transmit Buffer **Reset value: 0000_H**



Field	Bits	Type	Description
RFDATA	15:0	rw	**RF Data** Data bits which are transferred via the RF SSC Unit *Note: This register cannot be accessed byte-wide.*

Whenever new data is transferred to the register **RFSSCTB** and the previous transmission is finished (for example, the 16-bit shift register is empty) and the bit **RFSSCTB.SSCEN** is set to 1, the content of the register **RFSSCTB** is copied to the 16-bit wide shift register, an interrupt RFSSCTINT is generated and the transmission is started according to the control parameters of the registers **RFCON1** and **RFCON2**.

The RF SSC Unit is programmed either from the Control Block of the RF Control Unit or directly via the X-Bus.

7.8.3.4 Programming via the Control Block

If the transmission of a telegram is initialized by the GSM Timer Unit the programming of the RF SSC Unit is mainly done by the Control Block of the RF Control Unit. Only the register **RFCON1** has to be programmed by the controller. The registers **RFCON2** and **RFSSCTB** are automatically programmed by the Control Block according to the data and control information of the RF Control RAM. Nevertheless, the interrupt RFSSCTINT is generated and has therefore to be disabled at the interrupt control unit of the controller.

7.8.3.5 Programming Directly via the X-Bus

A telegram can also directly be programmed via the X-Bus interface. A transmission is started when the bit **RFCON2.SSCEN** is set to 1 and a write operation has been performed in the register **RFSSCTB**. It does not matter whether the register **RFSSCTB** or **RFCON2** is programmed first. The interrupt RFSSCTINT is generated by the RF Control Unit when the RFSSCTB data has been copied to the shift register. Therefore, the software can rewrite the register **RFSSCTB** again after the interrupt has occurred. If the register **RFSSCTB** is rewritten before the transmission of the previous telegram - the content of which is now in the shift register - is finished, a continuous transmission of both telegrams is performed. By such means a 24-bit telegram to the RF IC can be generated by programming two 12-bit wide words into the register **RFSSCTB**. When the transmission of the telegram is finished the bit **RFCON2.SSCEN** is cleared by hardware which enables the software to detect whether the transmission has been finished or not.

Note: There is no hardware protection mechanism which takes care that the RF SSC Unit registers are NOT written at the same time by the Control Block of the RF Control Unit and the controller via the X-Bus. This has to be guaranteed by software. Therefore, it is recommended to program the RF SSC Unit directly via the X-Bus only for the initialization procedure of the RF devices when the GSM Timer Unit does not initiate any telegram transfer.

7.8.3.6 Application Note

To transmit a telegram of 24 bits, it has to be split into two parts of 12 bits. After sending the first 12 bits, the second part has to be written immediately in the transmit buffer register **RFSSCTB** to avoid a gap in the telegram transmission. It is not necessary to set the bit **RFCON2.SSCEN** again if the second value is written fast enough. This pseudo-code sends two 12-bit telegram with a 6.5 MHz bit clock:

(...)
RFCON1 = 0000_H;
RFCON2 = $853B_H$; **RFSSCTB** = 0222_H; **RFSSCTB** = $0FFF_H$;
(...)

Note: The use of C code to send two 12-bit telegrams is not fast enough. If C code is used, only the first telegram is sent. Assembly code for the RFSSC must be used to be fast enough.

7.8.4 Control Block

7.8.4.1 Telegram Control Block

The telegram control block has to take care that the control information for each telegram and the telegram data are transferred from the RF RAM at the appropriate point in time to the RF SSC Unit. In addition it has to trigger the start of the telegram transmission.

The principle timing of an 8 bit telegram with a low active strobe signal (STBSEL[i] = 0) and telegram control bits set to:

RFCON2.SSCHB = 0

TLB = 00 (**RFCON2.SSCBM** = 7_D)

See **Figure 61**.

Single Telegram Transmission

Telegrams and power ramp data can be transmitted in parallel. When the signal 'trig_start' is activated by the GSM Timer Unit, a telegram is transferred.

1. The GSM timer Unit sends a trigger signal to the RF Control Block, which initializes the telegram control part of the Control Block and initiates the telegram transmission. This trigger signal 'trig_start' is provided by a timing event in the GSM Timer Unit.
2. The telegram data is transferred from the RF Control Unit RAM to the synchronous serial interface (RF SSC Unit).
3. In the RF SSC Unit each 8, 16 or 24 bit telegram is transmitted immediately via the 3-wire bus to the RF IC. When the telegram is transferred via the serial interface with a 6.5 MHz or 3.25 MHz bit block the serial transmission on RF_DATA starts after a fixed delay of 11 clk_gsm_if clock cycles after the CTDMA Counter has reached the corresponding timing compare value.

Note: For single telegram transmission the Multiple Telegram Transfer Control bit must always be set to 0.

Multiple Telegram Transmission

In multiple telegram transmission mode a series of telegrams can be transmitted automatically only needing one trigger event by the GSM Timer Unit. The minimum series length is 1 (that is, single telegram transmission is also included), the maximum series length is 40 or 112 (depending on the RAM partitioning type). The RF RAM address of the first telegram of a series is encoded in the bits TRIG[5:0] of the timing event in the GSM Timer Unit and every telegram contains a Multiple Telegram Transfer Control Bit.

When initiated by the trigger signal 'trig_start' the first telegram is read out of the RF RAM and the Multiple Telegram Transfer Control Bit is analyzed. If the Multiple Telegram Transfer Control Bit is set to 1, the next telegram with index i +1 (this is not the trigger value) will be automatically transmitted after the telegram with index i without the signal 'trig_start' signal being activated. In this way a series of consecutive telegrams with variable length can be transmitted with only one entry in the GSM Timer RAM. The last telegram of a series is indicated by setting the Multiple Telegram Transfer Control Bit to 0.

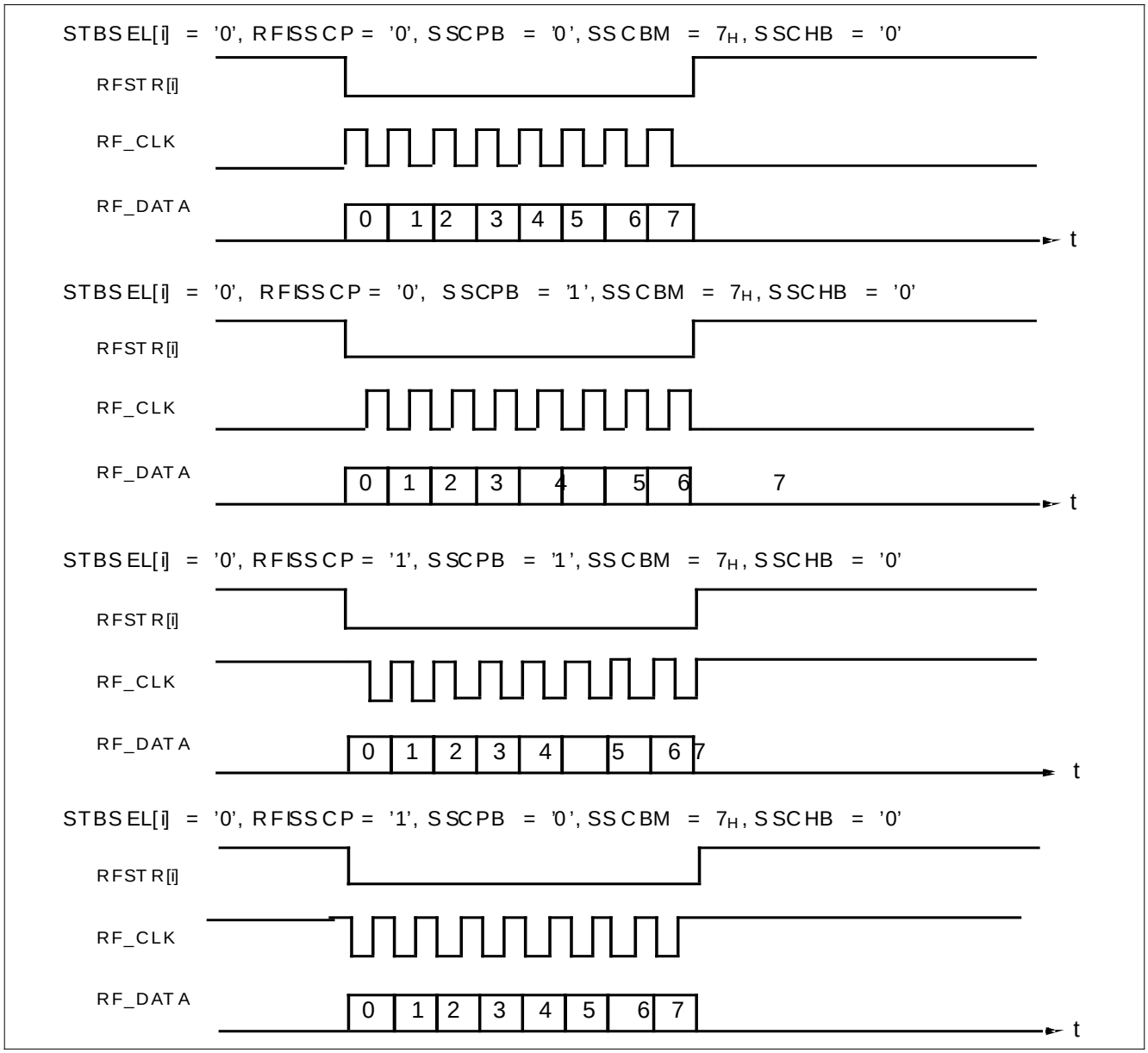


Figure 61 Principle Timings of 8 Bit Telegrams

*Note: In this implementation the number of telegrams to be transferred in multiple telegram transmission mode is **not** encoded in the first telegram to determine the end of transmission. By the use of the Multiple Telegram Transfer Control Bit the end of the series of consecutive telegrams is determined by the first telegram index where this bit is set to 0, thus, avoiding the implementation of a telegram counter.*

When the signal 'trig_start' is activated by the GSM Timer Unit, the first telegram of the series is transferred. If this telegram is transferred over the serial interface with 6.5 MHz or 3.25 MHz, the transmission starts after a fixed delay of 11 clk_gsm_if clock cycles after the CTDMA Counter has reached the corresponding timing compare value. The transmission of the telegram with index i +1 starts after a fixed delay of 5 clk_gsm_if clock cycles after the signal RFSTR of the telegram with index i goes inactive.

Notes

1. *If the Multiple Telegram Control Bit is set to 1 in the last telegram of the RF RAM memory space, that is, telegram i = 40 in RF RAM partitioning type 1 and i = 112 in RF RAM partitioning type 2, the series of telegrams is aborted because there is no next telegram available.*
2. *The RFSTR[i] signal is active during each telegram and inactive between the transmission of telegrams. It changes RFSTR(i) to RFSTR(j) from telegram to telegram to allow transmission of telegrams to different receivers during a series of telegrams.*
3. *During transmission of telegrams (single or multiple mode) the GSM Timer Unit can generate all timer events except telegram transmission. The user has be careful that the transmission of a series of telegrams is finished prior to any new telegram transmission event in the GSM Timer Unit. Otherwise, the new telegram transmission event is ignored.*

Attention: When the signal trig_start is activated and the previous telegram (or a series of telegrams) is still being transmitted, the signal trig_start is ignored. Therefore, the programmer of the GSM Timer Unit must ensure that the transmission of a telegram is never requested before the transmission of the last telegram is finished.

Note: Power ramp sequences cannot be transmitted during multiple telegram transmission.

Telegram Index Mapping

Which telegram is transferred is determined by the signal lines TRIG[5:0] with TRIG[5:0] = i.

For **RF RAM Partitioning Type 1** (refer to Table 32 on page 171) i = TRIG[5:0] is decoded as follows:

IF ($i < 8_D$) THEN
    don't transfer anything
ELSIF ($i < 48_D$) THEN
    transmit telegram ($i - 7_D$)
ELSIF
    transmit power ramp sequence ($i - 47_D$)
ENDIF

For **RF RAM Partitioning Type 2** (refer to Table 33 on page 172) i is decoded as follows:

IF (i = 0) THEN
    don't transfer anything
ELSIF ($i < 50_D$) THEN
    transmit telegram (2*i - 1)
ELSIF
    transmit telegram ($i + 49_D$)
ENDIF

Note: TRIG[5:0] indicates either the index of a single telegram or the index of the first telegram of a series of telegrams in multiple telegram transmission mode.

7.8.4.2 PA Control Block

The PA Control Block controls the data transfers and timing aspects within the RF Control Unit necessary to provide the PA Ramping Unit (refer to **Section 7.7 RF Power Ramping (on Page 162)**) with information used to generate the signals for the PA Control Hardware (refer to **Section 7.3.7.1 PA Control Hardware Offset Measurement PAOUTOF1 (on Page 122)**).

7.9 GSM Timer Unit

System Integration

- Supply domain: VDD_LD1
- Chip internal interfaces:
  - Clock domain: Refer to **Section 7.2.1.3 Sub-System Clocks and Enables (on Page 67)** and see **Figure (on Page 71)**.
  - Bus domain: X-Bus
- Interrupt sources:
- Monitor Pins: Refer to **Section 9.7.10 Internal Signal Monitoring (on Page 435)**.

7.9.1 Overview

The GSM Timer Unit provides all timing signals which are periodically repeated in TDMA frames and thus off-loads the controller from scheduling events. These timing signals are chip internal signals (for example, trigger signals for equalizer, RF control, etc.) as well as chip external signals for control of the HF ICs. By means of a user programmable event table each timing signal can be programmed very flexible. Thus, the timing signals of the GSM Timer Core are not application specific. They are all generated in an identical way and the meaning is only determined by the GSM Timer Decoder and the controller software or DSP firmware (see **Figure 62**).

*Note: In E-GOLDvoice only 5 of the 12 GSM T_OUT timer signals are mapped onto balls for system control (refer to **T_OUT[8,4,3,2,0]**). The remaining signals are not connected to balls.*

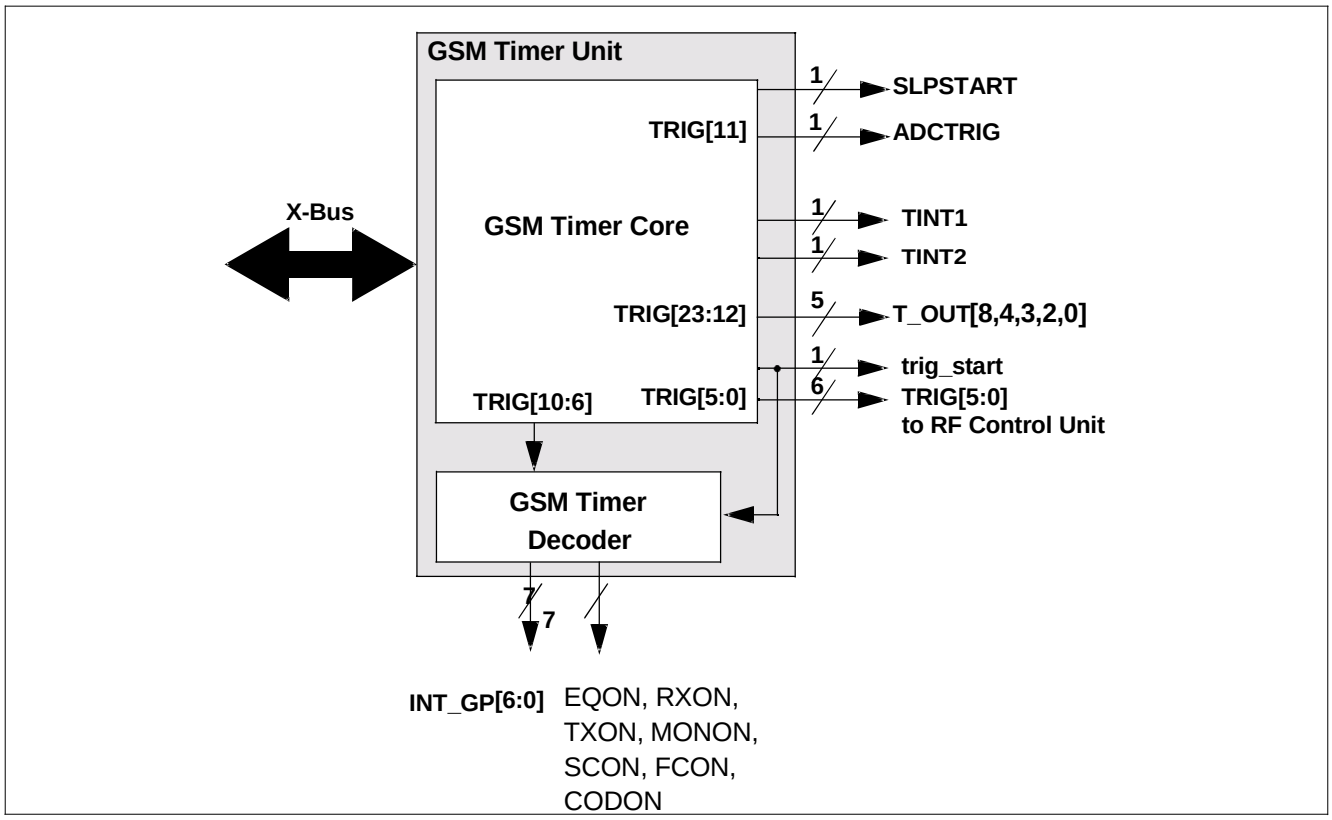


Figure 62 External Interfaces of the GSM Timer Unit

Table 35 GSM Timer Unit Signals

Signal Name	Derived from	Function
TRIG[5:0]	-	Selects the telegram which has to be transferred by the RF Control Unit
INT_GP[4:0]	TRIG[10:6][1)]	5 general purpose interrupt signals connected to the controller
INT_GP[6:5]		2 general purpose interrupt signals connected to DSP: INT_GP[5] = FRAME_INT INT_GP[6] = SYS_MCU_INT
EQON MONON SCON FCON CODON		Signals for the DSP to control data acquisition
TXON RXON[2)]		Signals to enable/disable the hardware blocks of the RX path (analog and digital) and TX path (analog)
ADCTRIG	TRIG[11]	Trigger signal for measurement interface
T_OUT[8,4,3,2,0]	TRIG[23:12]	12 signals are provided to switch on/off RF ICs periodically each TDMA frame. For E-GOLDvoice only 5 signals are mapped onto the pins T_OUT0, T_OUT2, T_OUT3, T_OUT4, and T_OUT8.
trig_start	-	Signal triggers transmission of telegrams or power ramps by the RF Control Unit and triggers the GSM Timer Decoder.
SLPSTART	-	Trigger signal for SCCU block
TINT1	-	Controller interrupts generated at a programmed timer value of the RTDMA Counter
TINT2	-	

1) The description of the decoding scheme of TRIG[10:6] is located in **Section 7.9.3 GSM Timer Decoder (on Page 200).**

2) The RXON/TXON signals are internally connected to the RX_ON, TX_ON input signals of the PMU core

The GSM Timer Unit reduces the controller load for scheduling events and, therefore, it supports full rate channels, half rate channels, and multislot configurations.

An easy reconfiguration is possible for:

- Timing advance adjustment in transmit time slots
- Fade-out of the TX and RX control signals every 26th TDMA frame
- Insertion or fade-out of monitoring time slots
- Battery measurements
- Hand-over in general.

7.9.1.1 Basic Operation

The reference counter of the GSM Timer Unit (RTDMA Counter) located in the TDMA Counter Unit of the TDMA Compare Unit. The RTDMA Counter is a programmable (usually) modulo 10000_D 15-bit wide counter operated by the clock clk_counter with a clock period of 0.461 µs and allows to measure the length of one TDMA frame (4.615 ms) with a resolution of one eight of a bit (refer to **Section 7.9.1.5 TDMA Counter Unit**).

The Timer RAM of the GSM Timer Unit contains the information about the point in time (within a TDMA frame, called the Timing Compare Value) when the 26 output signals TRIG[23:0] may get their new values or change/toggle their values. The record of one Timing Compare Value and the corresponding values for the new output signals TRIG[23:0] is called a Timing Event.

The Timer RAM configured like a FIFO. The TDMA Timer Comparator compares the last Timing Compare Value of the FIFO with the value of the TDMA Counter Unit. When both values are identical, the corresponding output values for TRIG[23:0] are read out of the Timer RAM and forwarded to the output signals TRIG[23:0]. After a match

the Timer RAM address is incremented by the address generator of the RAM Control Unit and the next Timing Compare Value is compared with the value of the TDMA Counter Unit. A simplified example with only 4 timer output signals TRIG[3:0] illustrates the basic operation of the GSM Timer Unit in **Figure 63**. (In this example the Group Enable Unit with their corresponding entries in the Timer RAM have not been taken into account.)

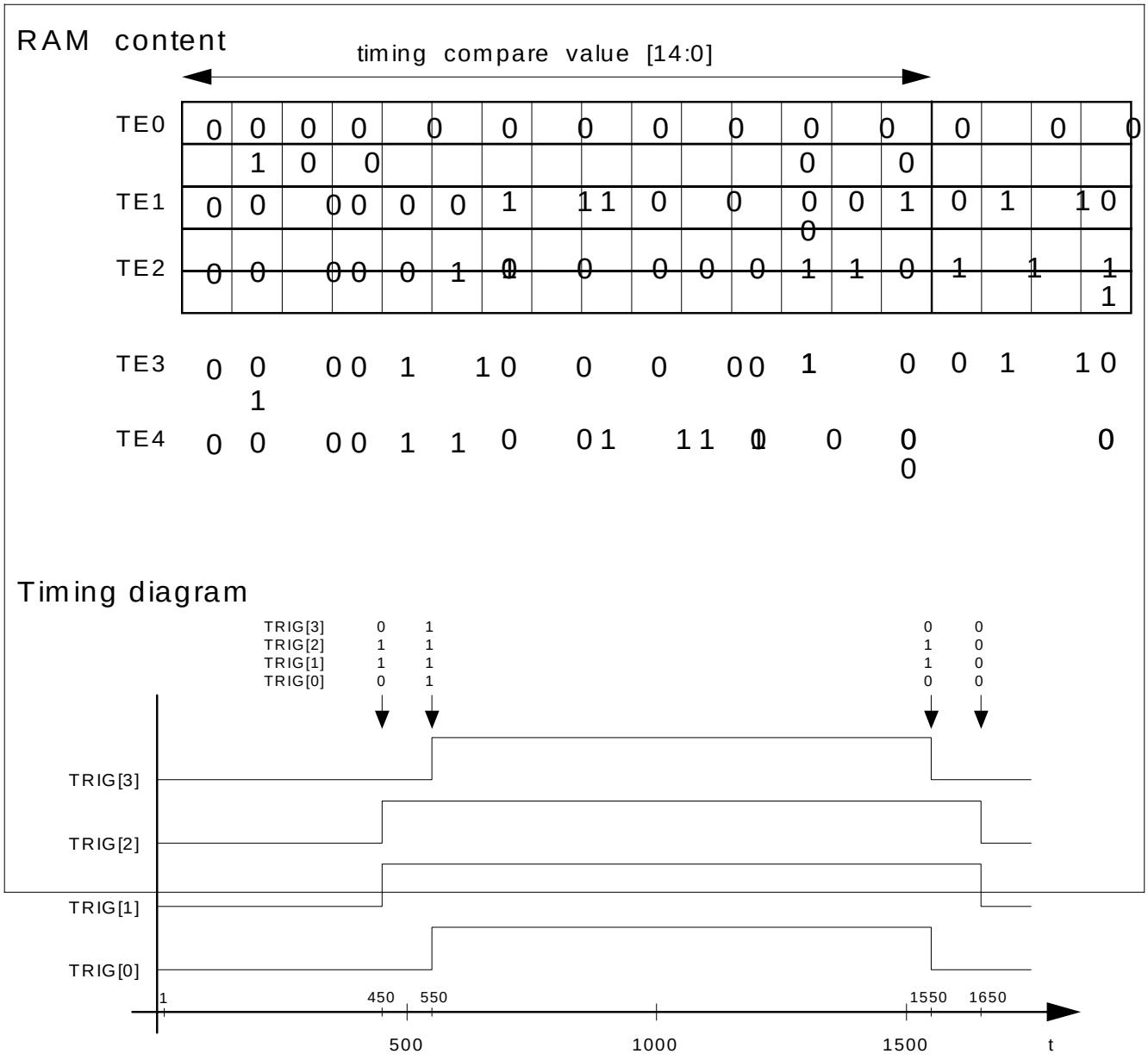


Figure 63 Basic Operation of the GSM Timer Unit

The Timer RAM in **Figure 63** stores 5 timing events TE0...TE4 each consisting of a 15-bit wide Timing Compare Value and the corresponding values for the 4 timer outputs. For example, the address generator points at the beginning to Timer RAM entry 0 with the Timing Compare Value of 0001. When the TDMA Counter Unit reaches this value the timer outputs TRIG[3:0] are changed according to the new output values stored in the Timer RAM. Afterwards the address generator is incremented and points to the second entry containing the Timing Compare Value 0450_D. When the TDMA Counter Unit reaches this value the timer outputs change again, etc. The timing event values have to be stored in incremental and consecutive order in the GSM Timer RAM.

When the address generator has reached the top address (defined by the pointer **TEAPT**) it jumps to the bottom address (defined by the pointer **TEAPB**). While an address region is executed a new set of timing events can be written into the Timer RAM region not in use to prepare the next frame. Thus, a new set of timing events can be activated by changing the bottom and top addresses of the address generator only.

7.9.1.2 Timer RAM

The GSM Timer Unit comprises a dual port Timer RAM of 512 * 16 bits. The RAM address range is defined in **Chapter 10 Register Lists and Mapping**. The Timer RAM contains timing event entries as already mentioned according to **Table 36 Structure of Timing Events (on Page 184)** and timing offset entries for timing advance operations according to **Table 37 Structure of Timing Advance (on Page 185)**.

Three different types of timing events are distinguished:

- Timing events in which all trigger signals TRIG[23:0] are identical to the bits TB[25:0] (refer to **Table 36**). This operation is explained in **Section 7.9.1.1 Basic Operation**.
- Two other types of timing events allow marking only the trigger signals that have to be set or reset in comparison to the previous trigger values. This feature is important for using the Group Assign Bits GAB[4:0] (refer to **Table 36**).

The Timer RAM can hold a maximum number of 170 Timing Events and/or Timing Offset Values.

The allowed sequence of Timing Events and Timing Offset Values in the Timer RAM is:

When the CTDMA Counter is reset the Timing Events or Offset Values are read out of the Timer RAM beginning at address **TEAPB** (refer to **Section 7.9.1.14 RAM Control Unit**). Therefore, the Timing Events and Timing Offset Values have to be stored in ascending order between the pointers **TEAPB** and **TEAPT** (the lowest timing compare value has to be stored at **TEAPB**, the highest timing compare value at entry [**TEAPT**-1]).

*Note: A timing event at the Timing Compare Value 0 is performed at the Timing Compare Value 1 and therefore it should **NOT** be programmed (this is a limitation of the chosen implementation).*

Within one 2.167 MHz clock cycle of the clock clk_counter (which is the frequency of the CTDMA Counter) up to 6 accesses of the dual port Timer RAM can be performed. To enable the timing events at the correct time and, therefore, to exclude delayed enables, the following conditions have to be considered:

- Not more than **one enabled** timing event is allowed at one timing compare value.
- Between two enabled timing events which timing compare values differ by n up to 3(n-1) disabled timing events and/or Timing Offset Values are allowed. The Disabled Timing Events may even have the same Timing Compare Value.
- Between the Timing Compare Value 0 and the first enabled timing event with the Timing Compare Value n (n>0) up to 3(n-1) Disabled Timing Events and/or timing offset values are allowed.

Table 36 Structure of Timing Events

Word	Bit	Function
1	8:0	Timer Bits TB[24:16] The new signals $TRIG_{new}$[24:16] are defined by TB[24:16] at the timing compare value TCV[14:0] specified in word 2.
	13:9	Group Assign Bits GAB[4:0]) The bits GAB[4:0] are processed in the Group Enable Unit (refer to **Section 7.9.1.17 Group Enable Unit (on Page 192)**).
	15:14	Entry Select Bits 00 This word and the following two RAM words represent a timing event entry with $TRIG_{new}$[25:0] = TB[25:0]. 01 This and the following two RAM words represent a timing event entry with $TRIG_{new}$[25, 11:0] = TB[25, 11:0] and $TRIG_{new}$[24:12] = $TRIG_{old}$[24:12] OR TB[24:12]. 10 This and the following two RAM words represent a timing event entry with $TRIG_{new}$[25, 11:0] = TB[25, 11:0] and $TRIG_{new}$[24:12] = $TRIG_{old}$[24:12] AND NOT(TB[24:12]). 11 Refer to **Table 37 Structure of Timing Advance (on Page 185)**, do not use for timing events

Table 36 Structure of Timing Events (cont'd)

Word	Bit	Function
2	14:0	Timing compare value TCV[14:0] Timing compare value which is compared with the CTDMA Counter value in the TDMA Timer Comparator.
	15	Timer Bit TB[25] The new signal $TRIG_{new}$[25] is defined by TB[25] at the timing compare value specified in TCV[14:0].
3	15:0	Timer Bits TB[15:0] The new signals $TRIG_{new}$[15:0] are defined by TB[15:0] at the timing compare value TCV[14:0] specified in word 2.

Table 37 Structure of Timing Advance

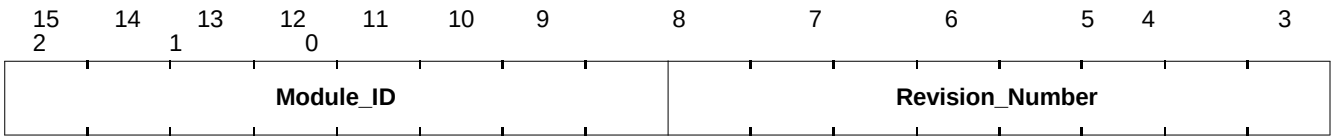
Word	Bit	Function
1	8:0	Reserved[1)]
	13:9	Group Assign Bits GAB[4:0]) The bits GAB[4:0] are processed in the Group Enable Unit (refer to **Section 7.9.1.17 Group Enable Unit (on Page 192)**).
	15:14	Entry Select Bits 00 Refer to **Table 36 Structure of Timing Events (on Page 184)**, do not use for timing offset values 01 Refer to **Table 36**, do not use for timing offset values 10 Refer to **Table 36**, do not use for timing offset values 11 This word and the following two RAM words represent a timing offset entry.
2	14:0	Timing offset value TOV[14:0] The timing offset value can be positive or negative. TOV[14] represents the sign. Negative values have to be expressed in two's complement notation.
	15	Reserved[1)]
3	15:0	Reserved[1)]

1) Unused or reserved bits should be set to 0.

7.9.1.3 GSM Timer Identification Register

TID

GSM Timer Identification Register **Reset values: 2103_H**



Field	Bits	Type	Description
Revision_Number	7:0	r	**GSM Timer Revision Number** These hard-wired bits are used for module revision numbering.
Module_ID	15:8	r	**GSM Timer Identification Number** These hard-wired bits are used for module identification numbering.

7.9.1.4 TDMA Compare Unit

The TDMA Compare Unit consists of:

- **TDMA Counter Unit** (**Section 7.9.1.5 TDMA Counter Unit (on Page 186)**) that contains a programmable modulo 10 000_D 15-bit counter operated by a 2.167 MHz clock and measures the length of one TDMA frame (4.615 ms).
- **Timing Advance Unit** (**Section 7.9.1.12 Timing Advance Unit (on Page 190)**) that stores the latest Timing Offset Value provided by the Timer RAM and adds this value to the Timing Compare Value.
- **TDMA Timer Comparator** (**Section 7.9.1.4 TDMA Compare Unit (on Page 185)**) which compares the value of the TDMA Counter Unit with the Timing Compare Value increased by the latest Timing Offset Value by the Timing Advance Unit.

7.9.1.5 TDMA Counter Unit

The block diagram of the TDMA Counter Unit is shown in **Figure 64**.

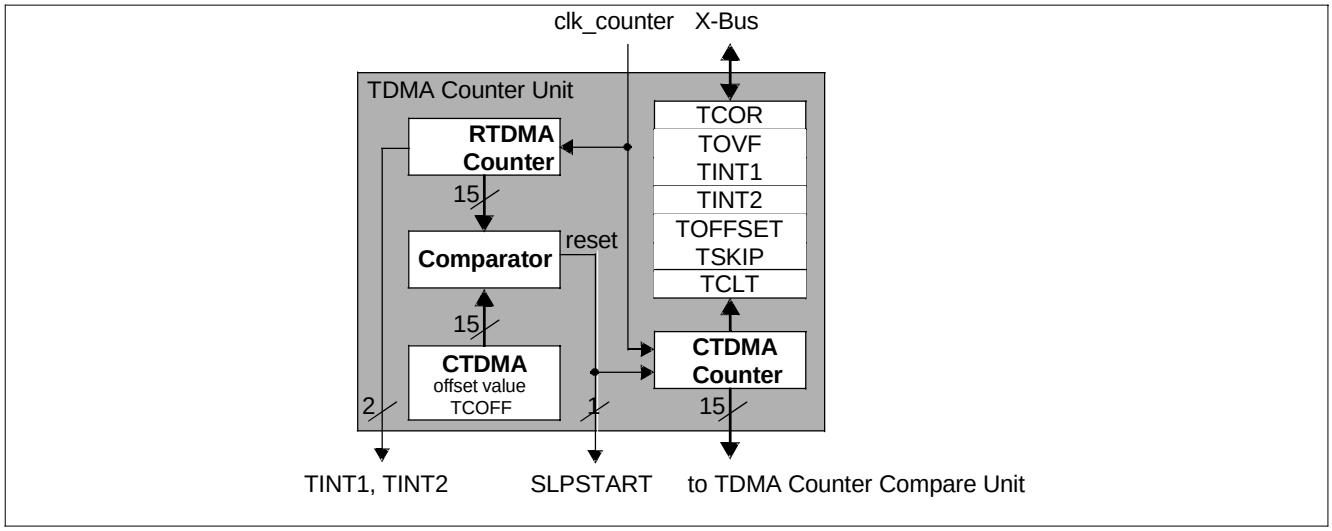


Figure 64 TDMA Counter Unit

The reference counter of the GSM Timer Unit is the RTDMA Counter of the TDMA Counter Unit. The RTDMA Counter is a modulo 10000_D 15 bit wide counter operated by a clock with 0.461 µs clock period and allows to measure the length of one TDMA frame (4.615 ms) with a resolution of one eights of a bit.

To synchronize the local TDMA frame timing of the mobile station with the global timing defined by the base station of the SCELL, the regular overflow value 9999_D ($270F_H$) of the RTDMA Counter can be overwritten temporarily by loading a 15-bit wide correction value into the RTDMA counter correction register **TCOR**. **See Figure 65**.

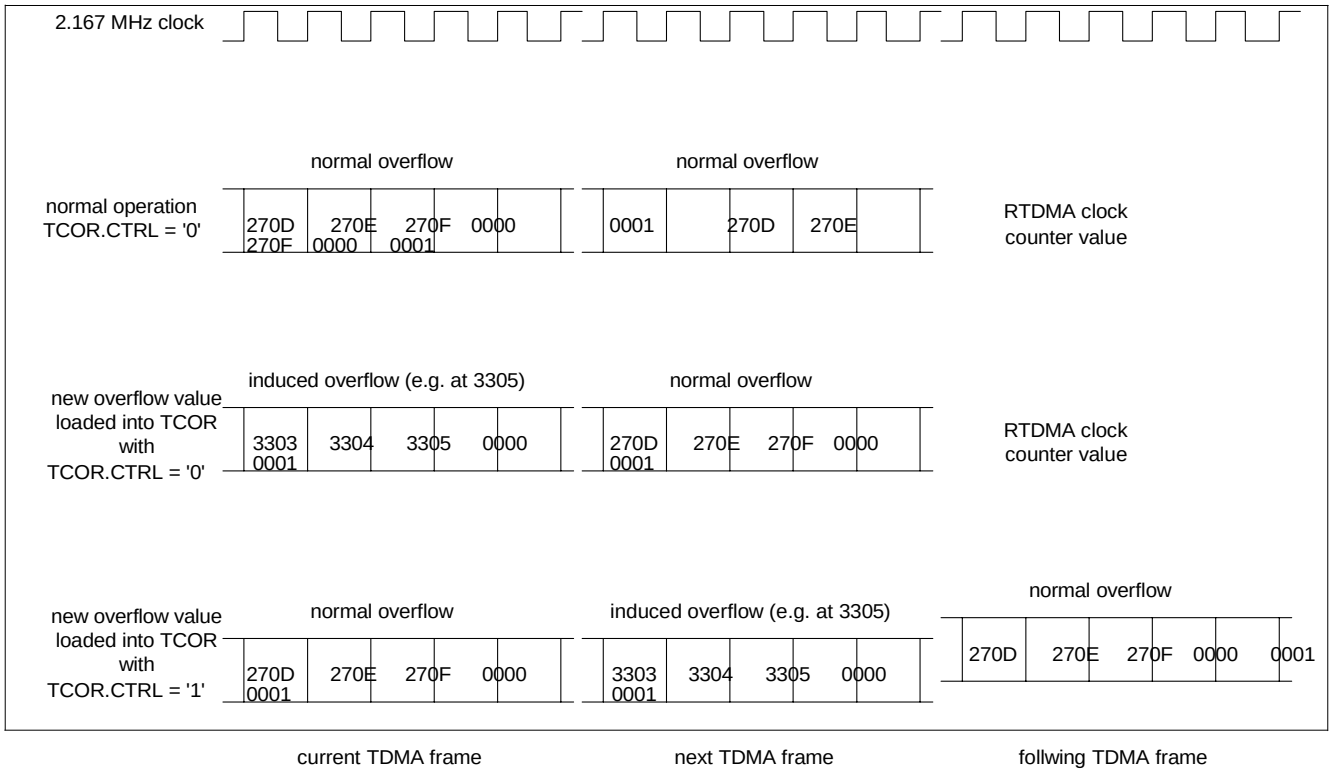


Figure 65 Overflow Behavior of RTDMA Counter

The correction value to be stored in the register **TCOR** can be found as follows:
A Frequency Correction Burst (FCB) search is started by the GSM Timer Unit (FCON goes high). When the DSP has found a FCB the number of transferred bits from the start of the FCB search up to the end of the detected FCB (end of time slot 0) is transferred to the shared memory. By means of this value the controller is able to evaluate the **TCOR** value.

9999_D ($270F_H$) is the overflow value of the RTDMA Counter being valid after a reset. If this value is modified (for example, for multiple monitoring during a TDMA frame), a new 15-bit wide overflow value (TOVF[14:0]) can be loaded into the counter overflow register **TOVF**.

*Note: For modulo 5000_D counting, the overflow value to be loaded into **TOVF** is 5000_D - 1 = 4999_D, that is, TOVF[14:0] = 1387_H.*

Related to the counter value of the RTDMA Counter two interrupt signals TINT1 and TINT2 are provided to the interrupt control unit of the controller. The TINT1 and TINT2 are toggle interrupts, that is, every time the level of TINT1 or TINT2 toggles an interrupt is generated. The RTDMA Counter value where the interrupt TINT1 and TINT2 is generated is determined by the timer interrupt registers **TINT1** and **TINT2**.

Besides the RTDMA Counter a second 15-bit wide counter, the CTDMA (current TDMA) counter exists. Using the CTDMA counter offset register **TOFFSET** an offset TCOFF between the RTDMA Counter and CTDMA Counter can be programmed. Therefore, the RTDMA Counter can always be synchronized to the SCELL and during monitoring frames. By changing the **TOFFSET** register value, the CTDMA Counter can easily be synchronized with the neighboring cells (NCELLs):

'CTDMA Counter value' = 'RTDMA Counter value' - 'TCOFF value'. (48)

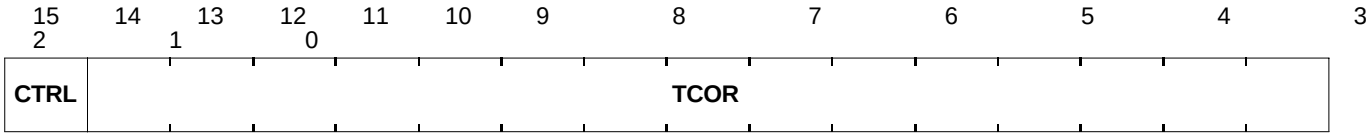
By using the Frame Skip Register **TFSKIP**, the reset of the CTDMA Counter can be canceled once. Afterwards, the bit **TFSKIP.SKIP** is automatically reset internally. The bit **SKIP** is especially suited for the handling of the idle frame in TCH/FR channels. For this mode the bit **SKIP** should be set to 1 in the CTDMA frame 24 to make the CTDMA Counter perform only 25 frame cycles, each of them with one RX, TX, and neighbor cell monitoring mode, which eases the programming of the Timer RAM.

The CTDMA Counter value can be monitored via the Counter Latch Register **TCLT**.

7.9.1.6 RTDMA Counter Correction Register

TCOR

RTDMA Counter Correction Register **Reset value: 0000_H**

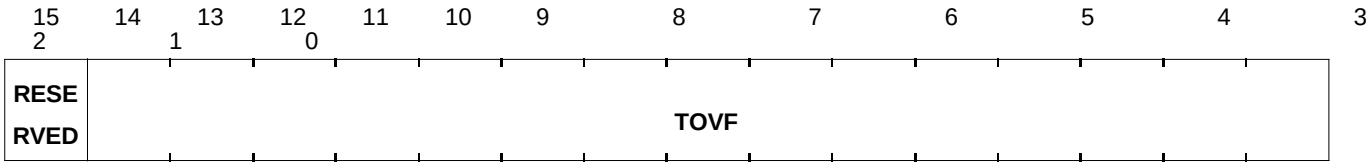


Field	Bits	Type	Description
TCOR	14:0	rw	**Counter Correction Value** Temporarily new overflow value
CTRL	15	rw	**Control** 0 The next RTDMA Counter overflow occurs at the counter value equal to the correction value **TCOR**. After that, the regular overflow value is valid again. *Note:If the correction value **TCOR** is lower than the current RTDMA Counter value the counter will overflow first at the regular overflow value (defined in register **TOVF**) and after that it overflows at the correction value. Note:**TCOR** cannot be written to while a correction is active. When activating a correction with TCOR:* *If **CTRL** = 0, the TCOR register cannot be written until the end of the current RTDMA frame* *If **CTRL** = 1 the TCOR register cannot be written until the end of the next RTDMA frame.* 1 The RTDMA Counter first overflows at the regular overflow value (defined in register **TOVF**) and then at the counter value equal to the correction value TCOR. After that, the regular overflow value is valid again.

7.9.1.7 RTDMA Counter Overflow Register

TOVF

RTDMA Counter Overflow Register **Reset value: $270F_H$**

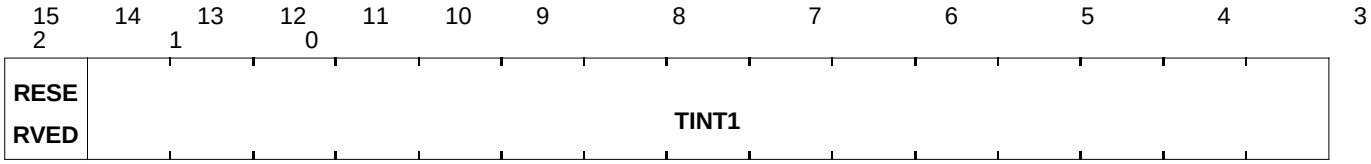


Field	Bits	Type	Description
TOVF	14:0	rw	**Counter Overflow Value** User defined overflow value of the RTDMA Counter. *Note: **TOVF** cannot be written to while a TCOR correction is active. Access to TOVF register must wait until the completion the TCOR correction.*
RESERVED	15	r	Reserved; these bits must be left at their reset values.

7.9.1.8 RTDMA Timer Interrupt Registers

TINT1

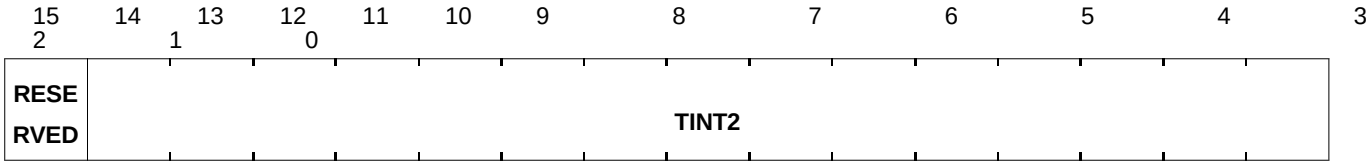
RTDMA Timer Interrupt Register 1 **Reset value: $7FFF_H$**



Field	Bits	Type	Description
TINT1	14:0	rw	**Timer Interrupt 1 Value** User defined RTDMA Counter value at which an interrupt TINT1 is generated.
RESERVED	15	r	Reserved; these bits must be left at their reset values.

TINT2

RTDMA Timer Interrupt Register 2 **Reset value: $7FFF_H$**

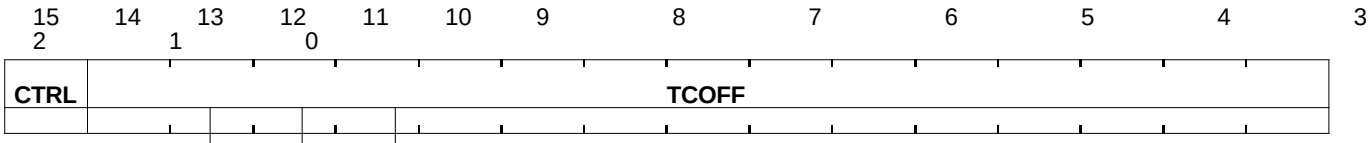


Field	Bits	Type	Description
TINT2	14:0	rw	**Timer Interrupt 2 Value** User defined RTDMA Counter value at which an interrupt TINT2 is generated.
RESERVED	15	r	Reserved; these bits must be left at their reset values.

7.9.1.9 CTDMA Counter Offset Register

TOFFSET

CTDMA Counter Offset Register **Reset value: 0000_H**



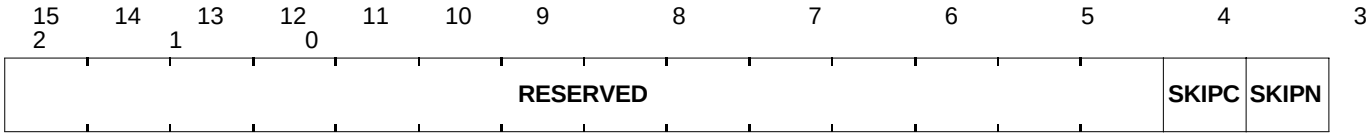
Field	Bits	Type	Description
TCOFF	14:0	rw	**CTDMA Counter Offset Value** RTDMA Counter value at which CTDMA Counter is reset.
CTRL	15	r	**Control** 0 New CTDMA offset value is directly validated *Note: If the CTDMA Counter offset value is lower than the current RTDMA Counter value the CTDMA Counter will not be reset before the next RTDMA frame.* 1 New CTDMA offset value validated the after next RTDMA counter overflow[1)]

1) If this register is read, the actual active value is read.

7.9.1.10 CTDMA Frame Skip Register

TFSKIP

CTDMA Frame Skip Register **Reset value: 0000_H**

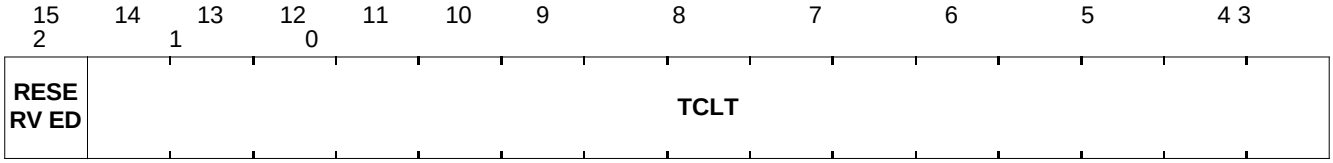


Field	Bits	Type	Description
SKIPN	0	rw	**Skip Next CTDMA Counter Reset** 0 No action 1 The next reset of the CTDMA Counter is skipped. This bit is validated after the next CTDMA Counter overflow. The bit is reset by the unit with the skipped reset pulse.
SKIPC	1	r	**Skip Current CTDMA Counter Reset** 0 No action 1 The current reset of the CTDMA Counter is skipped. This bit is directly validated. The bit is reset by the unit with the skipped reset pulse.
RESERVED	15:2	r	Reserved; these bits must be left at their reset values.

7.9.1.11 Counter Latch Register

TCLT

Counter Latch Register **Reset value: 0000_H**



Field	Bits	Type	Description
TCLT	14:0	r	**Timer counter latch value** At every READ access the current CTDMA Counter can be monitored via the **TCLT**.
RESERVED	15	r	Reserved; these bits must be left at their reset values.

7.9.1.12 Timing Advance Unit

When a new timing event or Timing Offset Value is read out of the Timer RAM, the Timing Advance Unit is activated:

- If a new Timing Offset Value is read out of the Timer RAM that **is not disabled** via the corresponding Group Enable Bit, this value is written into the Timing Advance Register of the Timing Advance Unit (this register can not be directly accessed via the X-bus).
- If a new Timing Offset Value is read out of the Timer RAM that **is disabled** via the corresponding Group Enable Bit, this value is skipped.
- If a new timing event is read out of the Timer RAM that **is not disabled** via the corresponding Group Enable Bit, the corresponding timing compare value is written into the Timing Compare Register of the Timing Advance Unit (this register can not be directly accessed via the X-Bus). The Timing Compare Register value and the Timing Advance Register value are added and forwarded to the TDMA Timing Compare Unit.

- If a new timing event is read out of the Timer RAM that **is disabled** via the corresponding Group Enable Bit, this event is skipped and the next entry is requested from the RAM Control Unit.

Notes

1. *When a timing event or Timing Offset Value was read out of the Timer RAM (disabled or not), the next entry in the Timer RAM is requested from the RAM Control Unit.*
2. *For power consumption reasons it is recommended to have always one valid entry in the Timer RAM. Otherwise, the Timer RAM is read out with the maximum possible frequency without any effect on the output signals TRIG[23:0] of the GSM Timer Unit.*

7.9.1.13 TDMA Timer Comparator

The TDMA Timer Comparator compares the value provided by the Timing Advance Unit with the CTDMA Counter value. If both values are equal, the next entry of the Timer RAM is requested from the RAM Control Unit.

7.9.1.14 RAM Control Unit

The TDMA Compare Unit requests a new entry from the RAM Control Unit when a:

- Timing offset value has been read by the timing advance unit
  (no matter if the Timing Offset Value was enabled or disabled),
- Timing event has been read that was disabled by the Group Enable Unit or
- Timing event that was enabled has reached the current CTDMA Counter value.

The current address pointer in the register **TCEAP** of the RAM Control Unit always points to the next address that has to be read out of the RAM. When a new entry is requested by the TDMA Compare Unit (starting from the current address pointer), the next three RAM addresses are read out of the Timer RAM and the current address pointer is correspondingly incremented. When the upper address specified by the pointer in the registers has been reached, the current address pointer is loaded with the lower address specified by the pointer in the register **TEAPB** and starts to increment again.

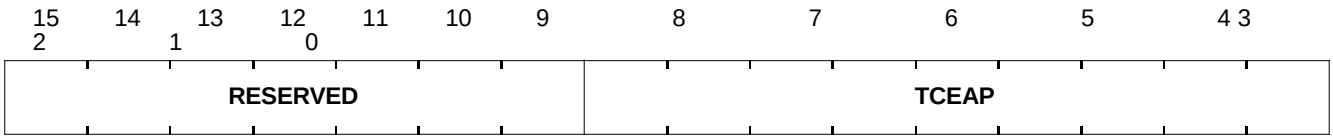
Notes

1. *When a new value is written into the register **TEAPT** or **TEAPB**, this value is valid after the next reset of the CTDMA Counter.*
2. *To prevent dead lock situations, the current address pointer is always reset to the **TEAPB** value when the CTDMA Counter is reset.*
   *When the GSM Timer Unit is initialized via the bit **TPARA.TINI** in the Output Control Unit, the current RAM pointer is set to the **TEAPB** value.*
3. *The RAM Control Unit works with only one timing event in the RAM.*

7.9.1.15 Current Timer Event Address Pointer Register

TCEAP

Current Timer Event Address Pointer Register **Reset value: 0000_H**



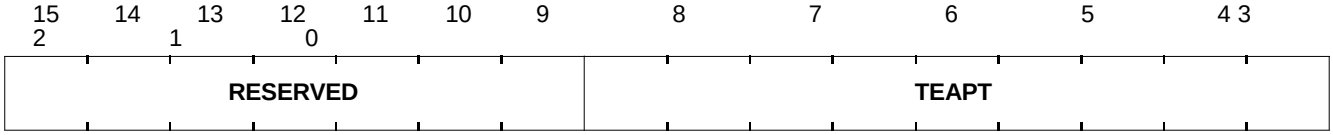
Field	Bits	Type	Description
TCEAP	8:0	r	**Current Timer Event Address Pointer** This is the address of the RAM where the next timing event will be read. *Note: This register should only be used for debug purposes since the register value can be wrong when, during the read access of the controller , the register content is changed by the RAM Control Unit!*
RESERVED	15:9	r	Reserved; these bits must be left at their reset values.

7.9.1.16 Timer Event Address Pointer Registers

TEAPx

TEAPT

Timer Event Top Address Pointer Register **Reset value: 0000_H**

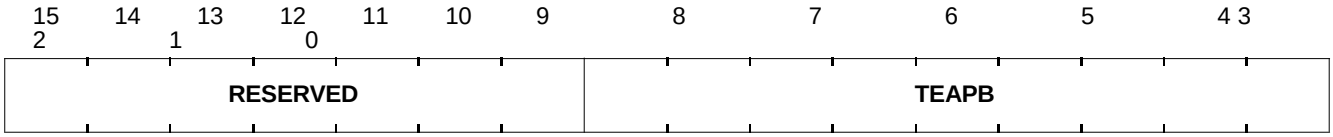


Field	Bits	Type	Description
TEAPT	8:0	rw	**Timer Event Top Address Pointer Value** This is the address where the current address pointer is reset to **TEAPB**. Allowed values are 3_D, 6_D, ... , 510_D. *Note: The entry at **TEAPT** is not executed.* *A new entry is active after the next reset of the CTDMA Counter.*[1)]
RESERVED	15:9	r	Reserved; these bits must be left at their reset values.

1) If this register is read, the actual active value is read.

TEAPB

Timer Event Bottom Address Pointer Register **Reset value: 0000_H**



Field	Bits	Type	Description
TEAPB	8:0	rw	**Timer Event Bottom Address Pointer Value** This is the low address of the Timer RAM entry to be executed. A new entry is active after the next reset of the CTDMA Counter[1)]. Allowed values are 0_D, 3_D, 6_D, ..., 507_D.
RESERVED	15:9	r	Reserved; these bits must be left at their reset values.

1) If this register is read the actual active value is read.

7.9.1.17 Group Enable Unit

Each timing event or Timing Offset Value can be assigned to one of 32 groups by setting the corresponding Group Assign Bits GAB[4:0] of the Timer RAM entry (see word 1 in **Table 36 Structure of Timing Events (on Page 184)** and in **Table 37 Structure of Timing Advance (on Page 185)**). A group of events can be enabled or disabled by

setting the corresponding Group Enable Bits GEB[31:0] in the Timer Group Enable Registers **TGERx**. 0 means that the event group is deactivated. 1 means that the event group is activated.

For example, all events necessary to be processed in TX bursts can be assigned to one group and disabled if needed. In the same way the events to be processed in RX bursts, monitoring bursts or to perform measurements can be grouped together and enabled or disabled whenever desired.

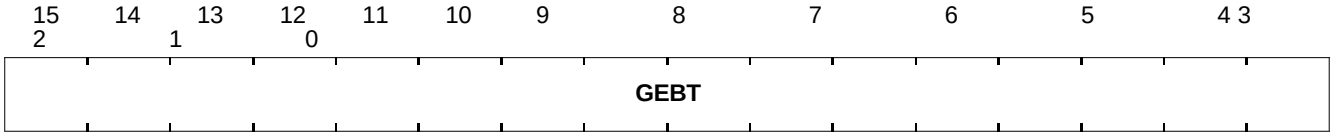
The timing event entries which mark the trigger signals that have to be set or reset, in addition to the previous timing events, are only necessary if, for example, a battery measurement is usually performed during a TX burst but also, sometimes, during a disabled TX burst.

7.9.1.18 Timer Group Enable Registers

TGERx

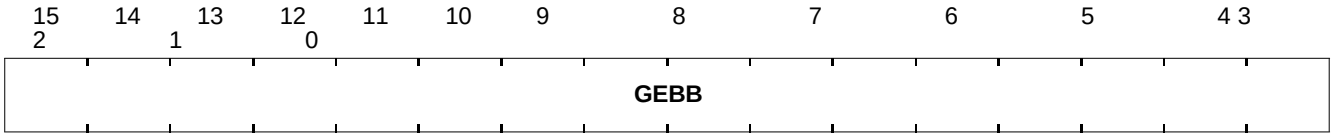
TGERT

Timer Group Enable Top Register **Reset value: 0000_H**



TGERB

Timer Group Enable Bottom Register **Reset value: 0000_H**



Field	Bits	Type	Description
GEBT GEBB	15:0	rw	**Group Enable Bits** For each group enable bit: 0 The corresponding timing events belonging to this group are masked out. That means that they do not affect the TRIG[23:0] output signals. 1 The corresponding timing events belonging to this group will be executed. *Note: A new value is active after the next reset of the CTDMA Counter.*[1)]

1) If this register is read, the actual active value is read.

7.9.1.19 Output Control Unit

The new values of the signals TRIG[23:0] are stored in the Select Unit of the Output Control Unit when both the:

- Output Control Unit is triggered by the RAM Control Unit
- Corresponding event is not masked out by the Group Enable Unit.

Valid entries are forwarded to the output signals TRIG[23:0] of the GSM Timer Unit if they are not faded out by the Fade Out Unit (**TPARA.TINI** = 1 and **TPARA.FDIS** = 1).

Using the Fade Out Unit the signal lines TRIG[23:0] of the GSM Timer Unit can be forced to the predefined values in **TFADEx.FTRIG**:

- If the Fade Out Unit of the Output Control Unit is enabled (**TPARA.FDIS** = 0) and the RTDMA Counter and the CTDMA Counter is running (**TPARA.TINI** = 1), then the output signals of the Fade Out Unit (instead of the output signals of the Select Unit) are forwarded to the GSM Timer Unit output signals TRIG[23:0].
  This means that TRIG[23:0] = **TFADEx.FTRIG**[23:0].
  The signal SLPSTART and the interrupts TINT1 and TINT2 are still generated.
- If the GSM Timer Unit is in the initialization state (**TPARA.TINI** = 0), the RTDMA Counter value is 1, the CTDMA Counter value is 0 and the current address pointer is equal to **TEAPB**, then all state machines are reset but the registers accessible via the controller keep their values.
  If **TPARA.TINI** = 0, the output signals of the Fade Out Unit determined by the Timer Fade Out register **TFADEx** are forwarded to the output signals TRIG[23:0]. The signal SLPSTART is 0.

The registers **TGERx** and keep their old values even if new values are written. The new values are not lost. They are stored in temporary registers. If the GSM is started (**TPARA.TINI** = 1), these registers are loaded with the new values if the RTDMA counter reaches the value of **TOFFSET** (for example, the CTDMA is reset).

If **TOFFSET** = 0, these registers get new values just after **TPARA.TINI** is set to 1.
If **TOFFSET** != 0, the old values are kept for as many timer cycles as the value of **TOFFSET**.

After initial programming of the GSM Timer Unit, the reset values of the registers **TGERx** and are zero. No timing event is executed until the RTDMA counter reaches the value of **TOFFSET** because, since both **TGERT** and **TGERB** are equal to 0, all timing events are disabled.

But, if the GSM Timer Unit is stopped and reprogrammed with new timing events and is restarted, the timing events are executed until the RTDMA counter reaches the value of **TOFFSET** (**TGERx** and still have their old values). To have the timer unit restart using immediately new timing event values, use the following procedure:

.../// previous GSM Timer Unit action

1. **TPARA**= 0// GSM Timer Unit is stopped
2. **TGERx**= 0// disable all groups
3. **TEAPT**= 0
4. **TEAPB**= 0
5. **TOFFSET**= 0// CTDMA counter reset just after restart
6. **TPARA**= 1// the registers **TGERx** are loaded with zero
7. **TPARA**= 0

.../// reprogramming of Timer RAM

8. **TGERx**= x// new values for the registers are set
9. **TOFFSET**= x
10. **TEAPT**= x
11. **TEAPB**= x
12. **TPARA**= 3

// restart GSM Timer Unit with new action

*Note: Power ramp sequences cannot be started while **TPARA.TINI** = 0.*

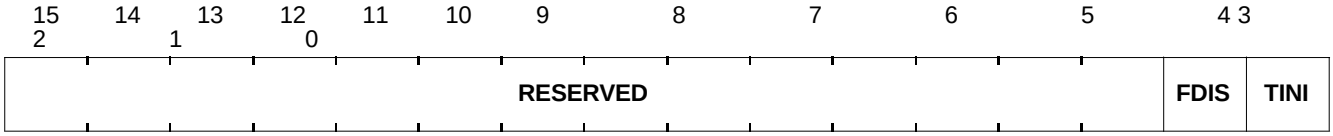
*Note: When **TPARA.FDIS** = 0 and registers **TFADEx** is written, the signal trig_start is generated.*

*Note: When **TPARA.FDIS** = 0, the **TFADEx** registers can be used to send telegrams or generate interrupts during the initialization phase.*

7.9.1.20 Timer Parameter Register

TPARA

Timer Parameter Register **Reset value: 0000_H**



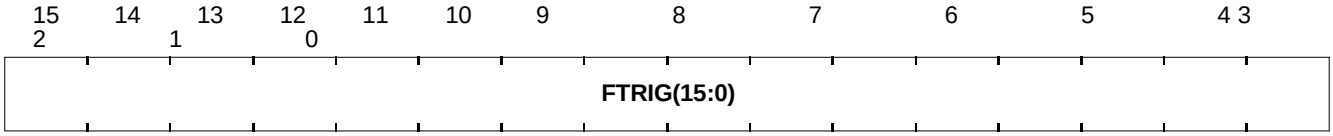
Field	Bits	Type	Description
TINI	0	rw	**Timer Init** 0 GSM Timer is in the initialization state 1 GSM Timer runs
FDIS	1	rw	**Fade Out Unit Disable** 0 Fade Out Unit enabled 1 Fade Out Unit disabled
RESERVED	15:2	r	Reserved; these bits must be left at their reset values.

7.9.1.21 Timer Fade Out Registers

TFADEx

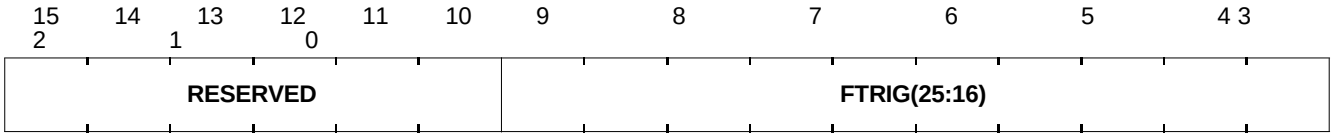
TFADE1

Timer Fade Out Register **Reset value: 0700_H**



TFADE2

Timer Fade Out Register **Reset value: 0000_H**



Field	Bits	Type	Description
FTRIG(25:16) FTRIG(15:0)	15:0	rw	**Fade Out Trigger Signal** If (***TPARA.FDIS*** = 0 OR ***TPARA*.TINI** = 0), then TRIG[23:0] := **FTRIG**[23:0].

7.9.1.22 SLPSTART Signal

The SLPSTART signal determines the exact point in time when the GSM System Interface starts or ends a sleep phase of the SCCU block. When the CTDMA Counter is reset, SLPSTART is set and when the CTDMA Counter reaches the value 1024, SLPSTART is reset.

7.9.2 GSM Clock Control Unit

The clock hw_clk[1)] is used to generate the clk_gsm_if with a frequency of 13 MHz for the GSM Timer Unit, the RF Control Unit, and the PA Control Hardware. This clock clk_gsm_if can be programmed by setting the corresponding control registers **GSMCLK1T**, **GSMCLK2T**, **GSMCLK3**.

Since the GSM System Interface, especially the GSM Timer Unit, is very sensitive to frequency errors of the system clock, either a voltage controlled quartz oscillator (VCXO) reference is required for the system or the frequency errors of a non-controlled quartz oscillator (XO) have to be compensated. Therefore, a programmable fractional divider with high resolution instead of a simple frequency divider with fixed ratio converts the clock hw_clk into the clock clk_gsm_if (**Figure 66**).Furthermore, the clock clk_gsm_if is divided by a fixed ratio of 1/6 to the clock clk_counter with 2.167 MHz used for the TDMA Counter Unit.

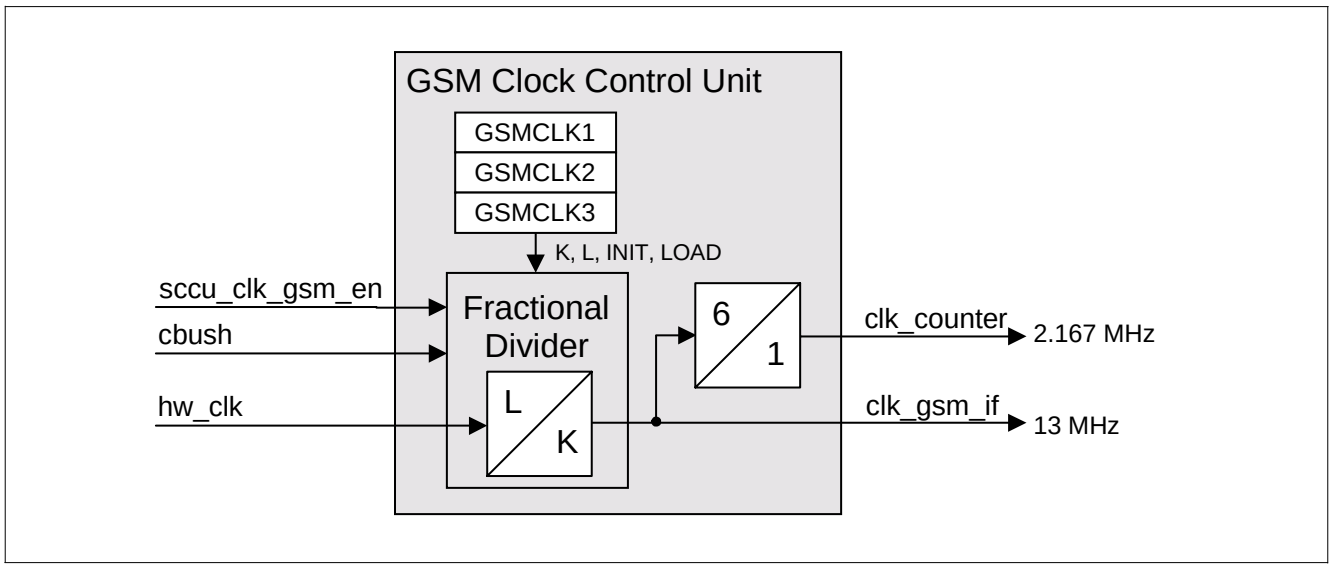


Figure 66 GSM Clock Control Unit

The fractional divider converts the frequency of hw_clk according to:

$$f_{clk\_gsm\_if} = K/L * f_{hw\_clk} \quad (49)$$

The numerator K, which is a 30-bit unsigned integer value, is set in the GSM System Interface Clock Control Register 1 (**GSMCLK1T**). The denominator L which is also a 30-bit unsigned integer value is set in the GSM System Interface Clock Control Register 2 (**GSMCLK2T**).

Reprogramming K and L has not an immediate effect, because this would lead to undesired intermediate states. Activation of new K and L values is done by setting appropriate bits in register **GSMCLK3** or by toggling the signal cbush which is provided by the **Section 7.2.1 Clock Generation Unit**.

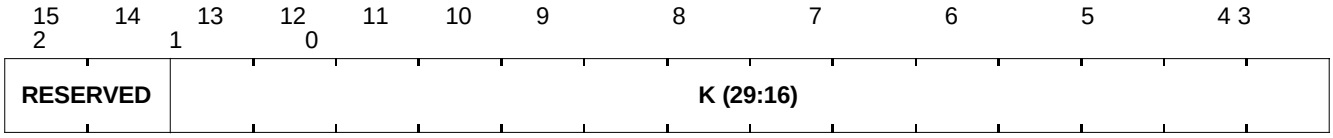
7.9.2.1 GSM System Interface Clock Control Registers

GSMCLK1x

1) Refer to the Clock Domain in **System Integration (on Page 181)**.

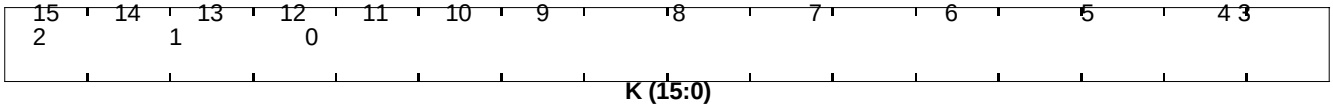
GSMCLK1T

GSM System Interface Clock Control Register 1 Top **Reset value: 0000_H**



GSMCLK1B

GSM System Interface Clock Control Register 1 Bottom **Reset value: 0001_H**

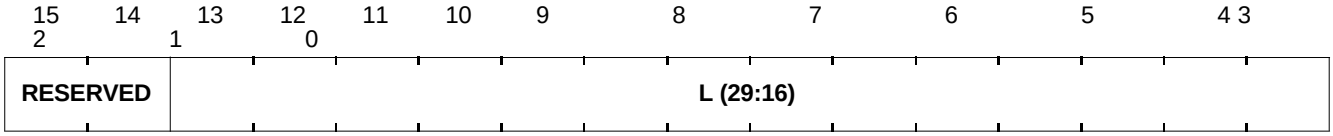


Field	Bits	Type	Description
K (29:16) K (15:0)	13:0 15:0	rw	**Numerator of the Fractional Divider** K is set according to **Table 38 K and L Values for Various Kernel Clock Frequency Errors (on Page 200)**. *Note: It always has to be **K < L** and **K > 0**.* *__K__ = __L__ is not allowed.* *Exception: __K__ = __L__ = 0 only if INIT (see register **GSMCLK3**) is used. For __K__ = __L__ = 0 clk_gsm_if is equal to clk_kernel.* *Note: __K__ has to be programmed prior to any action of the GSM System Interface.*
RESERVED	15:14 in GSM CLK1T	r	Reserved; these bits must be left at their reset values.

GSMCLK2x

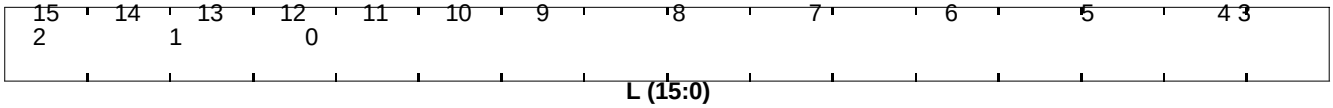
GSMCLK2T

GSM System Interface Clock Control Register 2 Top **Reset value: 0000_H**



GSMCLK2B

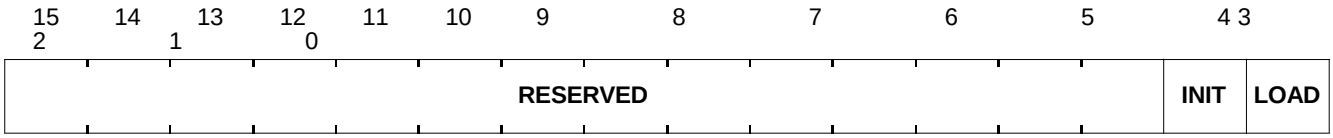
GSM System Interface Clock Control Register 2 Bottom **Reset value: 0002_H**



Field	Bits	Type	Description
L (29:16) L (15:0)	13:0 15:0	rw	**Denominator of the Fractional Divider** The value of **L** is set in depends on the frequency of the system oscillator according to **Table 38 K and L Values for Various Kernel Clock Frequency Errors (on Page 200)**. *Note: **L** has to be programmed prior to any action of the GSM System Interface.*
RESERVED	15:14 in GSM CLK2T	r	Reserved; these bits must be left at their reset values.

GSMCLK3

GSM System Interface Clock Control Register 3 **Reset value: 0000_H**



Field	Bits	Type	Description
LOAD	0	w	**Load Bit for K and L** 0 **LOAD** not used to load **K** and **L** *Note: **LOAD** always = 0 when read.* 1 **K** and **L** are loaded from **GSMCLK1x** and **GSMCLK2x** into the fractional divider and become valid. **LOAD** is used if both: • New values of **K** and **L** are needed because the ratio of the fractional divider has to be slightly adopted to a new deviation $\pm f_{hw\_clk}$ of the hw_clk frequency. • Operation of the clock can not be stopped. If **LOAD** is set to 1, The internal state of the fractional divider is not initialized, only the new values for **K** and **L** become valid. *Note: If **LOAD** is used the new values K_{new} and L_{new} have to fulfill the following conditions:* $L_{new} - K_{new} > K_{old}$, $K_{new} < L_{new}$, $K_{new} > 0$. *Note: If **LOAD** is used, it is recommended to only change either **L** or **K**. Otherwise, a gap in the clk_gsm_if can occur.* *Note: If the internal state of the fractional divider has to also be initialized, use **INIT** instead.*
INIT	1	w	**INIT Bit for K and L** 0 **INIT** not used to load **K** and **L** 1 **K** and **L** are loaded from **GSMCLK1x** and **GSMCLK2x** into the fractional divider and the internal state of the fractional divider is initialized according on the values of **K** and **L**. If the frequency of clk_kernel (and clk_bus) changes due to changes in the **Section 7.2.1 Clock Generation Unit**, there are two ways to initialize the fractional divider: **Asynchronous**: Configure the CGU, **GSMCLK1x**, and **GSMCLK2x**, use the **INIT** bit to load **K** and **L** and initialize the fractional divider. **Synchronous**: Configure the CGU, **GSMCLK1x**, and **GSMCLK2x**, and initialize the fractional divider via the cbush signal from the CGU. This changes synchronously the frequency of clk_kernel or clk_bus and the ratio of the fractional divider. *Note: **INIT** always = 1 when read.* *Note: If **INIT** = 1, **LOAD** = 'don't care' because **INIT** includes the load operation.*
RESERVED	15:2	r	Reserved; these bits must be left at their reset values.

If a VCXO reference is used, the fractional divider is used for simple integer division.

- For the 26 MHz clock clk_kernel:

$K = 0000\ 0001_H$ and $L = 0000\ 0002_H$ (50)

- For a 52 MHz clock clk_kernel:

$K = 0000\ 0001_H$ and $L = 0000\ 0004_H$ (51)

If an XO is used, the frequency error $^{TM}f_{clk\_kernel}$ must be compensated by appropriate choice of K and L.

- For a 52 MHz clock clk_kernel set:

$K = 1000'0000_H$ (52)

- For a 26 MHz clock clk_kernel set:

$K = 0800'0000_H$ (53)

In both cases the denominator is:

$L = round[2000'0000_H * (1 + {}^{TM}f_{clk\_kernel}/10^6\ ppm)]$. (54)

For an XO with 100 ppm accuracy the maximum error for clk_gsm_if is below 0.5 ppb with this setting.

A 13 MHz XO is not supported.

For a 13 MHz VCXO set:

$K = L = 0$. (55)

Table 38 shows several K and L values for a clock clk_kernel with and without the minimal and maximal frequency error.

Table 38 K and L Values for Various Kernel Clock Frequency Errors

clk_kernel in MHz	$^{TM}f_{clk\_kernel}$ in ppm	clk_gsm_if in MHz	K	L	System Concept
13	0	13.0	$0000\ 0000_H$	$0000\ 0000_H$	VCXO
26	0	13.0	$0000\ 0001_H$	$0000\ 0002_H$	VCXO
26	-100	13.0	$1000\ 0000_H$	$1FFF\ 2E49_H$	XO
26	100	13.0	$1000\ 0000_H$	$2000\ D1B7_H$	XO
52	0	13.0	$0000\ 0001_H$	$0000\ 0004_H$	VCXO
52	-100	13.0	$0800\ 0000_H$	$1FFF\ 2E49_H$	XO
52	100	13.0	$0800\ 0000_H$	$2000\ D1B7_H$	XO

7.9.3 GSM Timer Decoder

The pseudo code below describes how the signals TRIG[10:6] are decoded to generate the following internal signals: INT_GP[6:0], EQON, MONON, SCON, FCON, CODON, RXON, and TXON. The signals INT_GP[6:0] are toggle interrupts, that is, every time the level of INT_GP[i] toggles an interrupt is generated.

The function of the signals are described in **Table 35 GSM Timer Unit Signals**.

```
IF (signal trig_start is active AND TRIG[10:6] = 0)
       no action
ELSEIF (signal trig_start is active AND TRIG[10:6] = 1)
       signal EQON is set to 1
ELSEIF (signal trig_start is active AND TRIG[10:6] = 2)
       signal MONON is set to 1
ELSEIF (signal trig_start is active AND TRIG[10:6] = 3)
```

```
       signal SCON is set to 1
ELSEIF (signal trig_start is active AND TRIG[10:6] = 4)
       signal FCON is set to 1
ELSEIF (signal trig_start is active AND TRIG[10:6] = 5)
       signals EQON, MONON, SCON and FCON are reset to 0
ELSEIF (signal trig_start is active AND TRIG[10:6] = 6)
       signal RXON is set to 1
ELSEIF (signal trig_start is active AND TRIG[10:6] = 7)
       signal RXON is reset to 0
ELSEIF (signal trig_start is active AND TRIG[10:6] = 8)
       signal TXON is set to 1
ELSEIF (signal trig_start is active AND TRIG[10:6] = 9 )
       signal TXON is reset to 0
ELSEIF (signal trig_start is active AND TRIG[10:6] = 10)
       INT_GP[0] is toggled
ELSEIF (signal trig_start is active AND TRIG[10:6] = 11)
       INT_GP[1] is toggled
ELSEIF (signal trig_start is active AND TRIG[10:6] = 12)
       INT_GP[2] is toggled
ELSEIF (signal trig_start is active AND TRIG[10:6] = 13)
       INT_GP[3] is toggled
ELSEIF (signal trig_start is active AND TRIG[10:6] = 14)
       INT_GP[4] is toggled
ELSEIF (signal trig_start is active AND TRIG[10:6] = 15)
       INT_GP[5] is toggled
ELSEIF (signal trig_start is active AND TRIG[10:6] = 16)
       INT_GP[6] is toggled
ELSEIF (signal trig_start is active AND TRIG[10:6] = 17)
       signal CODON is set to '1'
ELSEIF (signal trig_start is active AND TRIG[10:6] = 18)
       signal CODON is reset to '0'
ELSEIF (signal trig_start is active AND TRIG[10:6] > 18)
       no action
ENDIF
```

7.10 External Bus Unit

System Integration

- Supply domain: VDD_LD1
- Chip internal interfaces:
  - Clock domain: cgu_c166_clk_pos1_o, cgu_c166_clk_pos2_o, and cgu_c166_clk_neg_o.
  - Bus domain: X-Bus
  - Interrupt sources:
- Monitor Pins: Refer to **Section 9.7.10 Internal Signal Monitoring (on Page 435)**.

7.10.1 Introduction

Although the E-GOLDvoice C166S subsystem supports a powerful set of on-chip peripherals and on-chip RAM and ROM areas, these internal units cover only a small fraction of the chip address space (up to 16 MBytes). The external bus interface allows access to external peripherals and additional volatile and non-volatile memory. The external bus interface has a number of possible configurations, so it can be tailored to fit perfectly into a given application system.

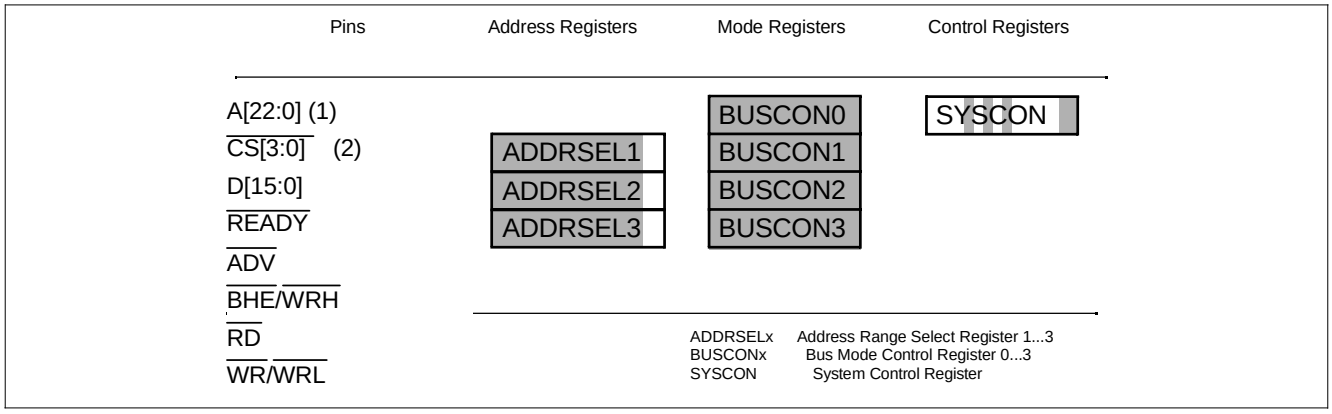


Figure 67 SFRs and Port Pins Associated with the External Bus Interface

Note: (2) for E-GOLDvoice the number of external chip-selects CS is restricted to CS[3:0] . So the number of associated ADDRSEL and BUSCON is also restricted to ADDRSEL[3:1] and BUSCON[3:0].

Accesses to external memory or peripherals are executed by the integrated External Bus Controller (EBC). The function of the EBC is controlled via the **SYSCON**, **BUSCONx**, and **ADDRSELx** registers. The **BUSCONx** registers specify the external bus cycles in terms of address (mux/demux), data width (16-bit/8-bit), chip selects, and length (waitstates/READY control/ALE/RW delay). These parameters are used for accesses within a specific address area that is defined via the corresponding register **ADDRSELx**.

The three pairs **BUSCON1**/**ADDRSEL1**...**BUSCON3**/**ADDRSEL3** make it possible to define three independent “address windows”, while all external accesses outside these windows are controlled via **BUSCON0**.

7.10.2 Single-Chip Mode

Single-chip mode is entered when the signal MON2 is Low during reset. In this case, **BUSCON0** is initialized with 0000_H, which resets bit **BUSCON0.BUSACT0**, so no external bus is enabled.
In this single-chip mode, the C166S operates only with internal resources. No external bus is configured and no external peripherals and/or memory can be accessed. No port lines are used for the bus interface. When running in single-chip mode, however, external access may be enabled by configuring an external bus under software control. Single-chip mode allows the C166S to start execution out of the internal program memory (Mask-ROM, SRAM).

Note: Any attempt to access a location in the external memory space in single-chip mode results in the hardware trap ILLBUS if no external bus has been enabled explicitly by software.

7.10.3 External Bus Modes

When the external bus interface is enabled (bit **BUSCONx.BUSACTx** = 1) and configured (bitfield **BUSCONx.BTYP**), E-GOLDvoice uses a subset of its port lines together with control lines to build the external bus.

Table 39 Summary of External Bus Modes

BTYP Encoding	External Data Bus Width	External Address Bus Mode
00	8-bit Data (1)	Demultiplexed Addresses
01	-	Reserved
10	16-bit Data	Demultiplexed Addresses
11	-	Reserved

Note: (1) this is not possible on $\overline{CS0}$.

The bus configuration (BTYP) for the address windows is selected via software in **BUSCON3**...**BUSCON1**, typically during the initialization of the system.

The external bus configuration at RESET for **BUSCON0** is described in **Chapter 12 System Reset (on Page 581)**. Further information on external bus signals is contained in **Section 7.10 Configuring External Bus and MCU Signals (on Page 291)**.

Otherwise, **BUSCON0** may be programmed via software just like the other BUSCON registers.

The 8-MByte address space of the C166S is divided into 128 segments of 64 kBytes each. The 23 address bits (A0-A22) have dedicated pins. In addition 4 chip select ($\overline{CS}$) lines may be used to select different memory banks or peripherals.

7.10.3.1 Multiplexed Bus Modes (Not Supported)

The multiplexed bus modes are not supported in the E-GOLDvoice.

7.10.3.2 Demultiplexed Bus Modes

In the demultiplexed bus modes no address latches are required.

The EBC initiates an external access by placing an address on the address bus. After a programmable period of time, the EBC activates the appropriate command signal ($\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$). Data is driven onto the data bus either by the EBC (for write cycles) or by the external memory/peripheral (for read cycles). After a period of time determined by the access time of the memory/peripheral, data becomes valid.

Read cycles: Input data is latched and the command signal is deactivated. This causes the accessed device to remove its data from the data bus which is then tri-stated again (refer to **Table 69 Demultiplexed Bus, Read Access (on Page 205)**).

Write cycles: The command signal is deactivated. If a subsequent external bus cycle is required, the EBC places the relevant address on the address bus. The data remain valid on the bus until the next external bus cycle is started (refer to **Table 68 Demultiplexed Bus, Write Access (on Page 204)**).

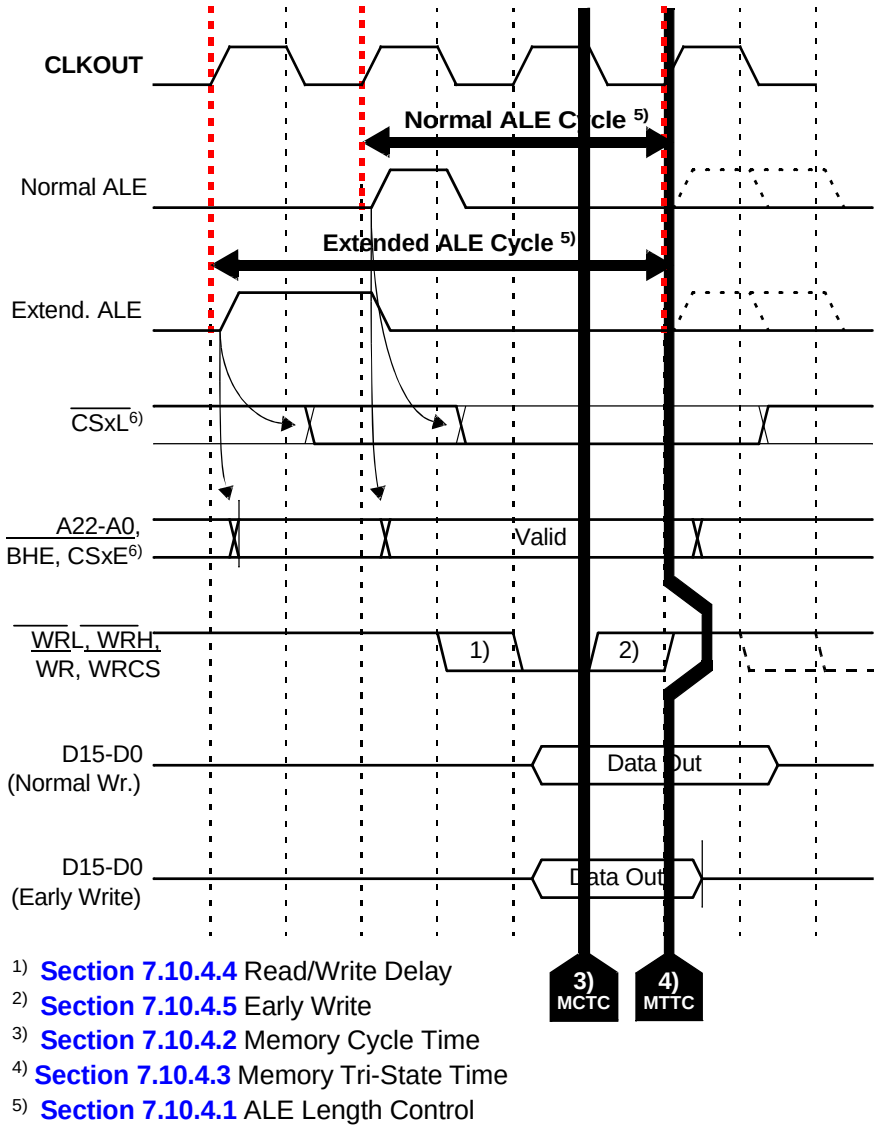
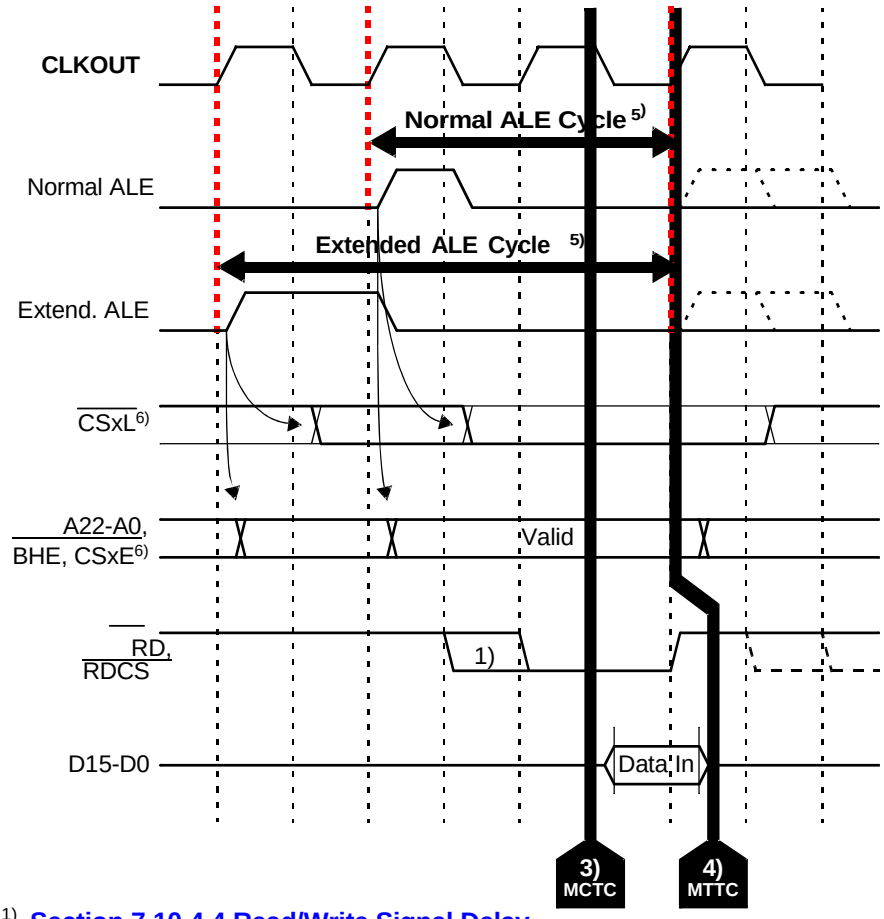


Figure 68 Demultiplexed Bus, Write Access



1) **Section 7.10.4.4 Read/Write Signal Delay**
3) **Section 7.10.4.2 Programmable Memory Cycle Time**
4) **Section 7.10.4.3 Programmable Memory Tri-State Time**
5) **Section 7.10.4.1 ALE Length Control**

Figure 69 Demultiplexed Bus, Read Access

7.10.3.3 Switching Among the Bus Modes

The EBC allows dynamic switching among different bus modes, that is, subsequent external bus cycles may be executed in different ways. Certain address areas can use predefined waitstates.

A change of the external bus characteristics can be initiated in two different ways:

- **Reprogramming the BUSCONx and/or ADDRSELx registers**
  This allows either:
  – The bus mode to be changed for a given address window
  – Changing the size of an address window that uses a certain bus mode.
- Reprogramming makes it possible to use a great number of different address windows (more than BUSCONs are available), although there is some overhead for changing the registers and keeping appropriate tables.
- **Switching between predefined address windows**
  This automatically selects the bus mode that is associated with the respective window. Predefined address windows allow the use of different bus modes without any overhead, but restrict the number of windows to the number of BUSCONs. However, as **BUSCON0** controls all address areas that are not covered by the other BUSCONs, there may be gaps between windows that use the bus mode of **BUSCON0**.

Note: Never change the configuration for an address area that currently supplies the instruction stream. Due to internal pipelining, it is very difficult to determine the first instruction fetch that will use the new configuration. Only change the configuration for address areas that are not currently accessed. This applies to **BUSCONx** *registers as well as to* **ADDRSELx** *registers.*

BUSCON/ADDRSEL register use is controlled via the addresses issued. When an access (code fetch or data) is initiated, the generated physical address determines whether the access is made internally, uses one of the address windows defined by **ADDRSEL3**...**ADDRSEL1** or uses the default configuration in **BUSCON0**. After initializing the active registers, they are selected and evaluated automatically by interpreting the physical address. No additional switching or selecting is necessary during run time, except when more than 3 address windows plus the default (**BUSCON0**) are to be used.

Switching between external resources (for example, for different peripherals) may create a problem if the previously-accessed resource needs too much time to switch off its output drivers (after a read), and if the resource to be accessed next switches on its output drivers very fast. In systems running on higher frequencies, this may lead to a bus conflict (switch-off delays normally are independent from the clock frequency).

In such a case, an additional waitstate can automatically be inserted when leaving a given address window, that is, when the next cycle accesses a different window. This is shown in **Figure 70**.

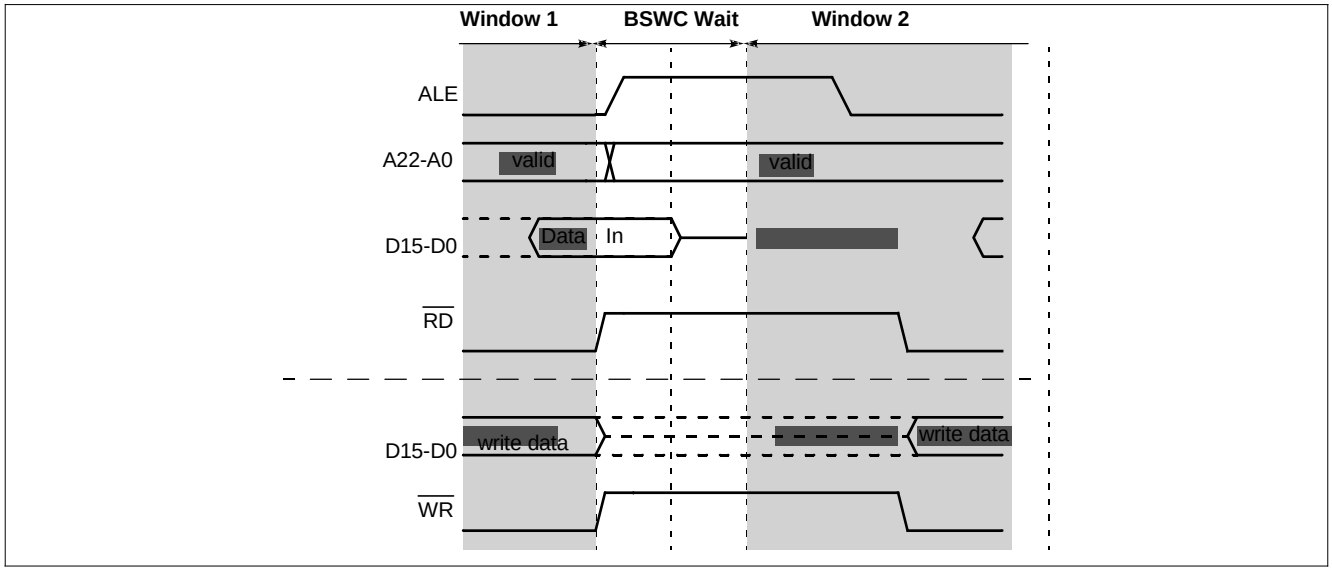


Figure 70 Optional BSWC Wait between BUSCON Windows

BUSCON switch waitstates are enabled via bits **BUSCONx.BSWCx**. By enabling the automatic BUSCON switch waitstate (**BSWCx** = 1), there is no impact on the system performance as long as the external bus cycles access the same address window. If the following cycle accesses a different window, a waitstate is inserted between the last access to the previous window and the first access to the new window.

After reset, no BUSCON switch waitstates are selected.

External Data Bus Width

The EBC can operate on a mixture of 8-bit- or 16-bit-wide external memory/peripherals. The 8-bit data accesses only use the lower data lines. If only 8-bit memories and peripherals are used, the upper lines can be reprogrammed to alternative functions. The EBC can control word accesses on an 8-bit data bus and byte accesses on a 16-bit data bus.

However Booting on an external 8bit device on $\overline{CS0}$ is not supported. Only 16bit devices are supported. The external bus configuration at RESET for **BUSCON0** is described in **Chapter 12 System Reset (on Page 581)**.

Word accesses on an 8-bit data bus are automatically split into two subsequent byte accesses in which the low byte is accessed first. The assembly of bytes to words and the disassembly of words into bytes is handled by the EBC and is transparent to the MCU and the programmer.

Byte accesses on a 16-bit data bus require that the upper and lower half of the memory can be accessed individually. In this case, the upper byte is selected with the Byte High Enable $\overline{BHE}$ signal, while the lower byte is selected with the A0 signal. The two bytes of the memory can be enabled either:

- Independently of each other
- Together when accessing words.

When writing bytes to an external 16-bit device that has a single $\overline{CS}$ input and two $\overline{WR}$ enable inputs (for the two bytes), the EBC can generate these two write control signals directly. This saves the external combination of the $\overline{WR}$ signal with A0 or $\overline{BHE}$. In this case, pin $\overline{WR}$ serves as $\overline{WRL}$ (WRite Low byte) and pin $\overline{BHE}$ serves as $\overline{WRH}$ (WRite High byte). **SYSCON.WRCFG** selects the operating mode for pins $\overline{WR}$ and $\overline{BHE}$. The respective byte is written on both data bus halves.

When reading bytes from an external 16-bit device, whole words may be read and the C166S automatically selects the byte to be input and discards the other. However, be careful when reading devices that change state when being read (such as FIFOs, interrupt status registers, etc.). In these cases, individual bytes must be selected using $\overline{BHE}$ and A0.

Disable/Enable Control for Pin $\overline{BHE}$ (BYTDIS)

SYSCON.BYTDIS is provided for controlling the active low Byte High Enable ($\overline{BHE}$) pin. The function of the $\overline{BHE}$ pin is enabled if the **BYTDIS** bit contains a 0. Otherwise, it is disabled and the pin can be used as a standard I/O pin. The $\overline{BHE}$ pin is used implicitly by the EBC to select one of two byte-organized memory chips, which are connected to the C166S via a word-wide external data bus. After reset, the $\overline{BHE}$ function is automatically enabled (**BYTDIS** = 0) if a 16-bit data bus is selected during reset; otherwise it is disabled (**BYTDIS** =1). It may be disabled if byte access to 16-bit memory is not required and if the $\overline{BHE}$ signal is not used.

Summary of Use of $\overline{WRL}$, $\overline{WRH}$, A0, and $\overline{BHE}$

A0 is never used as an address bit for 16-bit memory. The E-GOLDvoice address bit A1 is connected to memory A0.

A0, CSx#, and BHE# can be combined using external logic to generate CSL# and CSH# for memories with two chip select inputs or for two 8-bit memories.

It is better to use a 16-bit memory with WRL# and WRH# inputs. Setting bit **SYSCON.WRCFG** to 1 changes the function of BHE# to WRH# and WR# to WRL#. In this case A0 is not required. The write function is summarized in **Table 40 Use of WRL#, WRH#, A0, BHE for WRITE (on Page 207)** and the read function in **Table (on Page 208)**. For a byte read the controller reads a word from the memory and internally selects and aligns the required byte.

Memories that have the following input combination: CS#, LB# (low byte), UB# (upper byte), OE#, WE# must use the BHE# mode for a correct byte read and write. Refer to **Table 40** for a write and **Table** for a read.

Table 40 Use of WRL#, WRH#, A0, BHE for WRITE

SYSCON.WRCFG	A0	BHE#	WR#	Mode
0	0	0		0 Word
0	0	1	0	Low Byte on D(0:7)
0	1	0	0	High Byte on D(8:15)
0	x	x	1	No write Access
SYSCON.WRCFG	**A0**	**WRH#**	**WRL#**	**Mode**
1	0	0		0 Word
1	0	1	0	Low Byte on D(0:7)

Table 40 Use of WRL#, WRH#, A0, BHE for WRITE (cont'd)

SYSCON.WRCFG	A0	BHE#	WR#	Mode
1	1	0	1	High Byte on D(8:15)
1	x	1	1	No write Access

Table 41 Use of WRL#, WRH#, A0, BHE for READ

SYSCON.WRCFG	A0	BHE#	RD#	Mode
0	0	0		0 Word
0	0	1	0	Low Byte on D(0:7)
0	1	0	0	High Byte on D(8:15)
0	x	x	1	No read Access
SYSCON.WRCFG	**A0**	**WRH#, WRL#**	**RD#**	**Mode**
1		x	1	0 Word
1	x	1	0	Low Byte on D(0:7)
1	x	1	0	High Byte on D(8:15)
1	x	1	1	No read Access

Bus Mode Performance

Table 42 Bus Mode Versus Performance

Bus Mode	Transfer Rate (Speed factor for byte/word/dword access)	System Requirements	Free IO Lines
8-bit Demultipl.	Low(1/2/4)	Very low (no latch, byte bus)	D8:D15
16-bit Demultipl.	Very high(1/1/2)	Low (no latch, word bus)	---

Segment Address Generation

The number of address lines used internally is always 23. The **RP0H.SALSEL** bits are always set to full segment address A22...A16 at Reset.

Note: The total accessible address space may be increased by accessing several banks that are distinguished by individual chip select lines.

$\overline{\text{CS}}$ Signal Generation

CS0, CS1and CS3 have dedicated output pins. CS2 is an alternate function of the port ADV_n.

The $\overline{\text{CS}}$x signals identify accesses to different address ranges in the C16x memory map. The number of pins assigned to chip selects can be changed by configuring the pin logic by the SW after RESET.

During external accesses, the EBC can generate a (programmable) number of $\overline{\text{CS}}$ lines, which make it possible to select external peripherals or memory banks directly without requiring an external decoder. The number of $\overline{\text{CS}}$ lines is selected during reset in **RP0H.CSSEL** to the maximum of 4 (by default, there are only two pins connected immediately after a reset).

The $\overline{\text{CS}}$x outputs are associated with the **BUSCONx** registers, and they are driven active low for any access within the address area defined for the respective BUSCON register. For any access outside this defined address area, the respective $\overline{\text{CS}}$x signal will go inactive high. At the beginning of each external bus cycle, the corresponding valid $\overline{\text{CS}}$ signal is determined and activated. All other $\overline{\text{CS}}$ lines are deactivated (driven high) at the same time.

*Note: The $\overline{\text{CS}}$x signals is not updated for an access to any internal address area (that is, when no external bus cycle is started), even if this area is covered by the respective **ADDRSELx** register. An internal bus interface*

access deactivates all external $\overline{CS}$ signals.

On accesses to address windows without a selected $\overline{CS}$ line, all selected $\overline{CS}$ lines are deactivated.

The chip-select signals can be operated in four different modes (refer to **Table 43**) that are selected via bits **CSWENx** and **CSRENx** in the respective **BUSCONx** register.

Table 43 Chip-Select Generation Modes

CSWENx	CSRENx	Chip-Select Mode
0	0	Address chip select (default after reset)
0	1	Read chip select
1	0	Write chip select
1	1	Read/write chip select

Read or Write Chip-Select ($\overline{CS}$ is renamed $\overline{WRCS}$ or $\overline{RDCS}$ in the protocol diagrams) signals remain active only as long as the associated control signal ($\overline{RD}$ or $\overline{WR}$) is active. This also includes the programmable read/write delay. Read chip select is activated only for read cycles; write chip select is activated only for write cycles; and read/write chip select is activated for both read and write cycles (write cycles are assumed if either of the signals $\overline{WRH}$ or $\overline{WRL}$ goes active). These modes save external glue logic when accessing external devices such as latches or drivers that have only a single enable input.

Address Chip-Select signals remain active during the complete bus cycle. For address chip select signals, two generation modes can be selected via bit **SYSCON.CSCFG** (see **Figure 71 Latched and Early Chip Select (on Page 210)**):

- A **latched** address chip-select signal ($\overline{CS}$ is renamed in $\overline{CSxL}$ in the protocol diagrams) (**CSCFG** = 0) becomes active with the falling edge of ALE and becomes inactive at the beginning of an external bus cycle that accesses a different address window. No spikes are generated on the chip-select lines, and no changes occur as long as locations within the same address window or within internal memory (excluding internal bus interface) are accessed.
- An **early** address chip-select signal ($\overline{CS}$ is renamed in $\overline{CSxE}$ in the protocol diagrams) (**CSCFG** = 1) becomes active together with the address and BHE (if enabled) and remains active until the end of the current bus cycle. Early address chip-select signals are not latched internally and may toggle intermediately while the address is changing.

Note: $\overline{CS0}$ provides a latched address chip select directly after reset (except for single-chip mode) when the first instruction is fetched internally.

Internal pull-up devices are used to hold $\overline{CS0}$ and $\overline{CS1}$ lines high during reset.

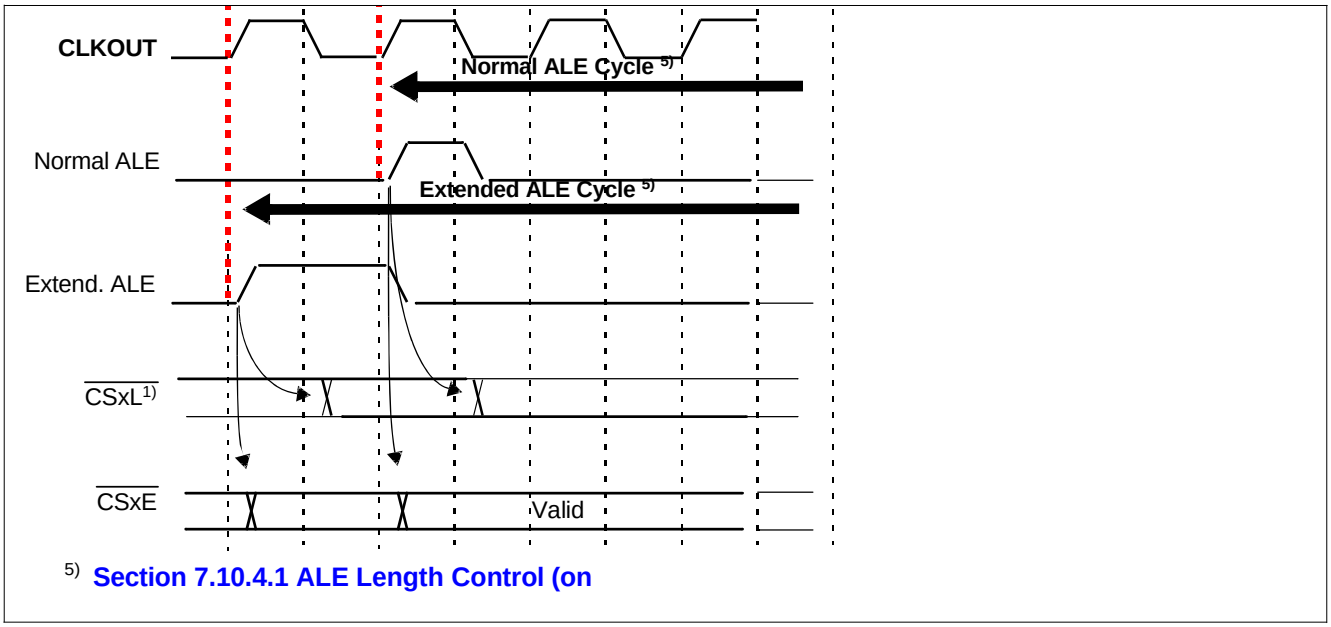


Figure 71 Latched and Early Chip Select

Segment Address Versus Chip Select

There are no limitations in the E-GOLDvoice on using all address lines and all chip selects.

The C166S can address a linear address space of 16 MByte. This allows implementation of a large sequential memory area and access to a great number of external devices using an external decoder. By increasing the number of $\overline{CS}$ lines, the C166S can access memory banks or peripherals without external glue logic. These two features may be combined to optimize the overall system performance.

7.10.4 Programmable Bus Characteristics

Important timing characteristics of the external bus interface have been made user-programmable to adapt it to a wide range of external bus and memory configurations with different types of memories and/or peripherals (see **Figure 72**).

The following parameters of an external bus cycle are programmable:

- **ALE Control**
  This defines the internal ALE signal length and the address hold time after its falling edge.
- **Memory Cycle Time** (extendable with 1-15 waitstates)
  This defines the allowable access time.
- **Memory Tri-State Time** (extendable with 1 waitstate)
  This defines the time for a data driver to float.
- **Read/Write Delay Time**
  This defines when a command is activated after the falling edge of ALE.

Note: External accesses use the slowest possible bus cycle after reset. The bus cycle timing can then be optimized by the initialization software.

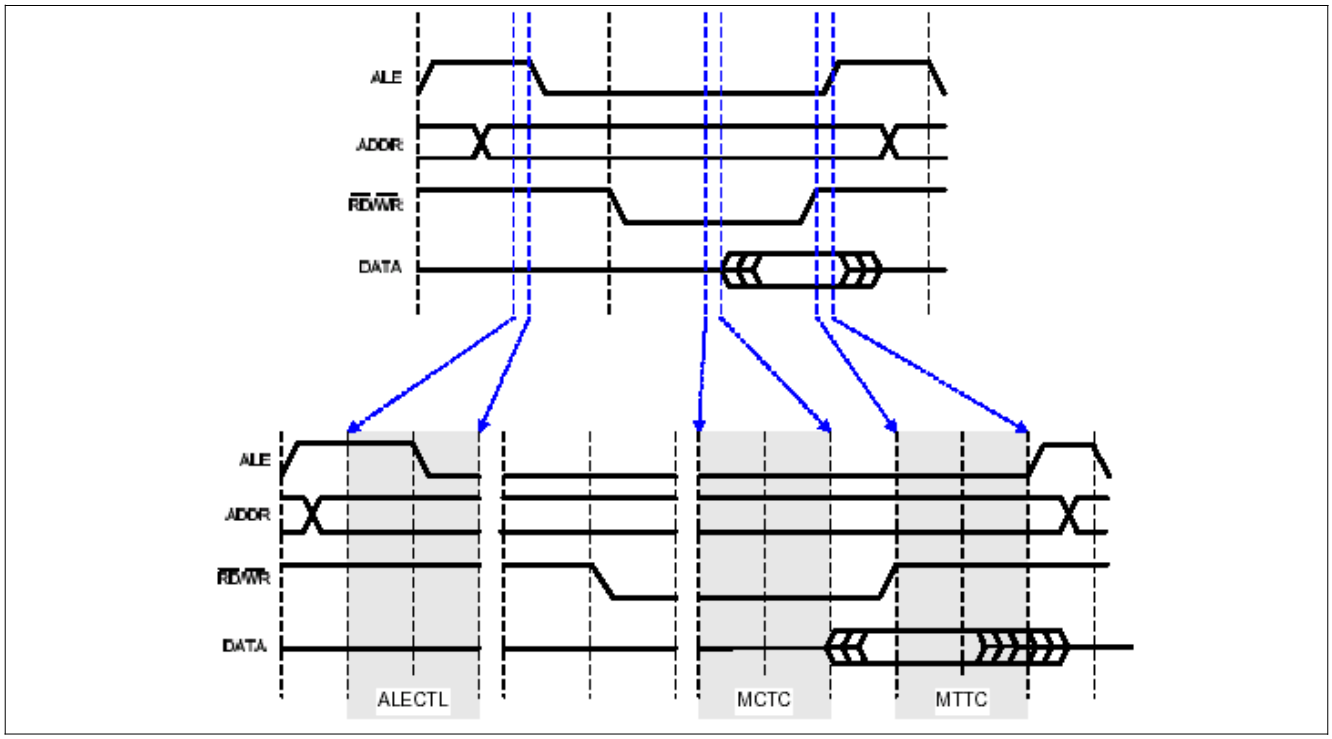


Figure 72 Programmable External Bus Cycle

7.10.4.1 ALE Length Control

Even though the ALE signal is not available externally this feature can be used for special timing adjustments. The length of the internal ALE signal and the address hold time after its falling edge are controlled by the **BUSCONx.ALECTLx** bits:

- When **ALECTLx** is set to 0, then the Normal ALE length is 1/2 of a bus cycle at 26Mhz
- When **ALECTLx** is set to 1, then the Extended ALE length is 1 bus cycle in all cases.

*Note: **BUSCON0.ALECTL0** is 1 after reset to select the slowest possible bus cycle, the other ALECTLx bits are 0 after reset.*

7.10.4.2 Programmable Memory Cycle Time

The C166S allows the user to adjust the controller external bus cycles to the access time of the respective memory or peripheral. This access time is the total time required to move the data to the destination. It represents the period of time during which the controller signals do not change.

The external bus cycles of the C166S can be extended by introducing wait states during access (see **Figure 72 Programmable External Bus Cycle (on Page 211)**) to compensate for a memory or peripheral that cannot keep pace with the controller maximum speed. During these memory cycle time wait states, the MCU is idle if this access is required for the execution of the current instruction.

The memory cycle time wait states can be programmed in increments of one MCU clock (2 TCLs) within a range from 0 to 15 (default after reset) via the Memory Cycle Time Control **BUSCONx.MCTC** fields. 15-<**MCTC**> wait states are inserted.

7.10.4.3 Programmable Memory Tri-State Time

The C166S allows the user to adjust the time between two subsequent external accesses to account for the tri-state time of the external device. The tristate time defines when the external device has released the bus after deactivation of the read command ($\overline{RD}$).

The output of the next address on the external bus can be delayed by introducing a wait state after the previous bus cycle to compensate for a memory or peripheral that needs more time to switch off its bus drivers (see **Figure 72 Programmable External Bus Cycle (on Page 211)**). During this memory tri-state time wait state the MCU is not idle, so MCU operations will be slowed down only if a subsequent external instruction or data fetch operation is required during the next instruction cycle.

The memory tristate time wait state requires one MCU clock (2 TCLs) and is controlled via is inserted if bit **BUSCONx.MTTCx** is 0 (default after reset).

7.10.4.4 Read/Write Signal Delay

The C166S allows the user to adjust the timing of the read and write commands to account for timing requirements of external peripherals. The read/write delay controls the time between the falling edge of ALE and the falling edge of the command. Without read/write delay, the falling edges of ALE and command(s) are concurrent (except for propagation delays). With the delay enabled, the command(s) become active half a MCU clock (1 TCL) after the falling edge of ALE. The read/write delay does not extend the memory cycle time, and does not slow down the controller.

The read/write delay is controlled via the Read Write Delay Control **BUSCONx.RWDCx** bits. The command(s) will be delayed if bit **RWDCx** is 0 (default after reset).

7.10.4.5 Early $\overline{WR}$

The duration of an external write access can be shortened by one TCL. The $\overline{WR}$ signal is activated (driven low) in the standard way, but can be deactivated (driven high) one TCL earlier than defined in the standard timing. In this case, the data output drivers will also be deactivated one TCL earlier.

This is especially useful in systems that operate on higher MCU clock frequencies and employ external modules (memories, peripherals, etc.) that switch on their own data drivers very rapidly in response to, for example, a chip select signal.

Conflicts between the C166S and external peripheral output drivers can be avoided by selecting early $\overline{WR}$ for the C166S.

Note: Make sure that the reduced $\overline{WR}$ low time still meets the requirements of the external peripheral or memory.
Early ~~WR~~ deactivation is controlled via the Early Write EnNable **BUSCONx.EWENx** bits. The ~~WR~~ signal is shortened if bit **EWENx** is 1 (default after reset is a standard ~~WR~~ signal, that is, **EWENx** = 0).

7.10.4.6 ~~READY~~ Controlled Bus Cycles

For situations in which the programmable wait states are not enough, or the response (access) time of a peripheral is not constant, the C166S has external bus cycles that are terminated via an asynchronous $\overline{READY}$ input signal. In this case the C166S first inserts a programmable number of wait states (0-7) and then monitors the $\overline{READY}$ line to determine the actual end of the current bus cycle. The external device drives $\overline{READY}$ low to indicate that either data have been latched (write cycle) or are available (read cycle).

The $\overline{READY}$ function is enabled via the ReaDY ENable **BUSCONx.RDYENx** bit. When this function is selected (**RDYENx** = 1), only the lower 3 bits of the respective **MCTC** bit field define the number of inserted waitstates (0-7), while the MSB of bit field **MCTC** is unused.

As shown in **Figure 74**, the asychronous $\overline{READY}$ requires additional wait states caused by the internal synchronization. The asynchronous $\overline{READY}$ is synchronized internally, and programmed wait states may be necessary to provide proper bus cycles (see notes on “normally-ready” peripherals on **Page 213**).

An asynchronous READY signal that has been activated by an external device may be deactivated in response to the trailing (rising) edge of the respective command (RD or WR).

Note: When the READY function is enabled for a specific address window, each bus cycle within this window must be terminated with an active READY signal, otherwise, the controller hangs until the next reset. A time-out function is provided by the watchdog timer.

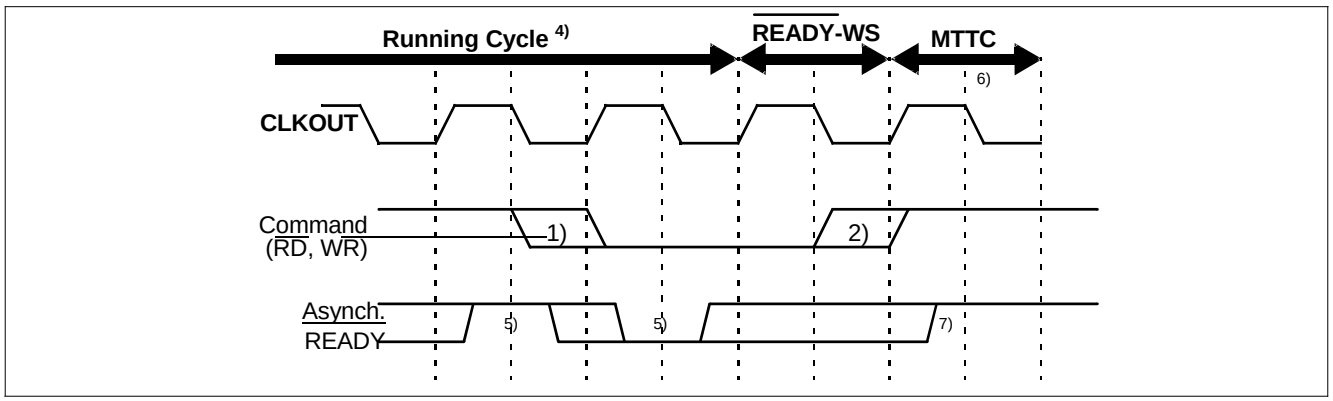


Figure 73 READY Controlled Bus Cycles a)

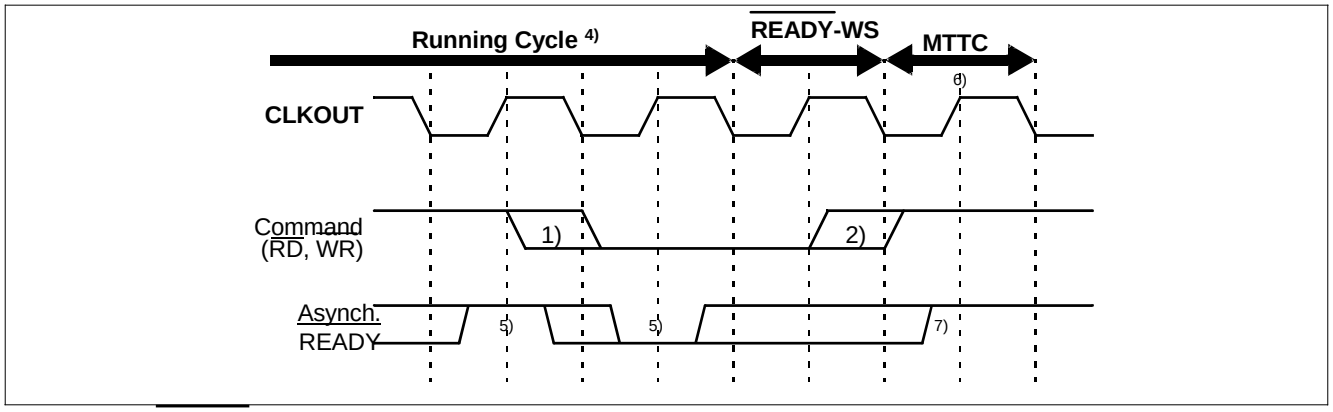


Figure 74 READY Controlled Bus Cycles b)

1) **Section 7.10.4.4 Read/Write Signal Delay (on Page 212)**

2) **Section 7.10.4.5 Early WR (on Page 212)**

4) Cycle as programmed, including **BUSCONx.MCTC** waitstates (Example shows 0 **MCTC** WS).

5) READY sampled HIGH at this sampling point generates a READY controlled waitstate,
READY sampled LOW at this sampling point terminates the currently running bus cycle.

6) For a demultiplexed bus **without MTTC** waitstate the delay here is zero.

7) If the next following bus cycle is READY controlled, an active READY signal must be disabled before the first valid sample point for the next bus cycle. This sample point depends on the **MTTC** waitstate of the current cycle, and on the **MCTC** waitstates and the ALE mode of the next cycle.

Combining the READY function with predefined wait states is advantageous in two cases:

1. Memory components with a fixed access time and peripherals operating with READY can be grouped into the same address window. The (external) waitstate control logic in this case would activate READY either upon the memory chip select or with the peripheral READY output. After the predefined number of wait states, the C166S checks its READY line to determine the end of the bus cycle. For a memory access it is already low; for a peripheral access it may be delayed. As memories tend to be faster than peripherals, there should be no impact on system performance.
2. When using the READY function with so-called “normally-ready” peripherals, erroneous bus cycles may occur if the READY line is sampled too early. These peripherals pull their READY output low while they are idle. When they are accessed, they deactivate READY until the bus cycle is complete, then drive it low again. If,

however, the peripheral deactivates $\overline{\text{READY}}$ after the first sample point of the C166S, the controller samples an active $\overline{\text{READY}}$ and terminates the current bus cycle too early. By inserting predefined waitstates, the first $\overline{\text{READY}}$ sample point can be shifted to an interval in which the peripheral has safely controlled the $\overline{\text{READY}}$ line.

7.10.5 Controlling the External Bus Controller

A set of registers controls the functions of the EBC. General features such as the usage of interface pins ($\overline{\text{WR}}$, $\overline{\text{BHE}}$), segmentation, and internal memory mapping are controlled via register **SYSCON**. The properties of a bus cycle, such as chip-select mode, length of ALE, external bus mode, read/write delay, and waitstates, are controlled via registers **BUSCON3**...**BUSCON0**. Four of these registers (**BUSCON3**...**BUSCON1**) have an address select register (**ADDRSEL3**...**ADDRSEL1**) associated with them, which makes it possible to specify up to four address areas and the individual bus characteristics within these areas. All accesses that are not covered by these four areas are controlled via **BUSCON0**. This allows the use of memory components or peripherals with different interfaces within the same system while optimizing accesses to each of them.

7.10.5.1 Bus Control Functions in the BUSCON Registers

Registers **BUSCON1**.. **BUSCON3**, which control the selected address windows, are completely under software control. Register **BUSCON0**, which is also used for the very first code access after reset, is partly controlled by hardware, that is, it is initialized via dedicated configuration signals during the reset sequence.

Attention: The bit descriptions are the same for registers BUSCON1.. BUSCON3 and are in the table below register BUSCON3.

BUSCONx

BUSCON0

Bus Control Register 0 **Reset value[1]: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSWEN0	CSREN0	RESERVED	RDYEN0	BSWC0	BUSACT0	ALECTL0	EWEN0	BTYP			MTTC0	RWDC0	MCTC		

1) The reset values of BUS ACT0 and ALE CTL0 may not be 0 as they are controlled by pin MON2 at reset.

BUSCON1

Bus Control Register 1 **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSWEN1	CSREN1	RESERVED	RDYEN1	BSWC1	BUSACT1	ALECTL1	EWEN1	BTYP			MTTC1	RWDC1	MCTC		

BUSCON2

Bus Control Register 2 **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSWEN2	CSREN2	RESERVED	RDYEN2	BSWC2	BUSACT2	ALECTL2	EWEN2	BTYP			MTTC2	RWDC2	MCTC		

7.10.5.2 Address Select Registers

ADDRSELx

BUSCON3

Bus Control Register 3 **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSW EN3	CSR EN3	RESE RVED	RDYE N3	BSW C3	BUS ACT3	ALE CTL3	EW EN3	BTYP		MTT C3	RWDC 3	MCTC			

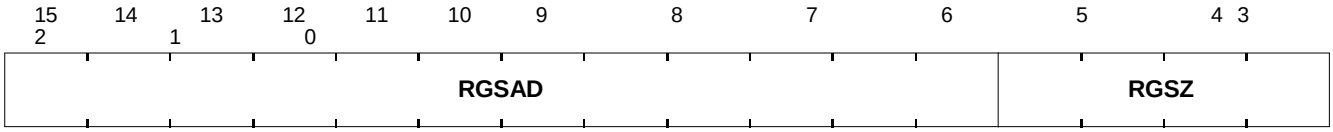
Field	Bits	Type	Description
MCTC	3:0	rw	**Memory Cycle Time Control** [1)] (Number of memory cycle time waitstates) 0000:15 waitstates . . . (Number = 15 - <**MCTC**>) 1111:No waitstates
RWDCx	4	rw	**Read/Write Delay Control for BUSCONx** 0: With rd/wr delay: activate command 1 TCL after falling edge of ALE 1: No RD/WR delay: activate command with falling edge of ALE
MTTCx	5	rw	**Memory Tristate Time Control** 0: 1 wait state 1: No wait states
BTYP	7:6	rw	**External Bus Configuration** 00: 8-bit Demultiplexed Bus – BUSCON0: 8-bit devices are not supported. By default at reset the $\overline{CS0}$ is configured for 16bit devices. – BUSCON1/2/3 : 8-bit devices are supported. 01: Reserved, do not use 10: 16-bit Demultiplexed Bus 11: Reserved, do not use
EWENx	8	rw	Early Write Enable 0: Normal $\overline{WR}$ signal 1: Early write: The $\overline{WR}$ signal is deactivated and write data is tri-stated one TCL earlier
ALECTLx	9	rw	ALE Lengthening Control 0: Normal ALE signal 1: Lengthened ALE signal
BUSACTx	10	rw	Bus Active Control 0: External bus disabled 1: External bus enabled within respective address window (**ADDRSELx**)
BSWCx	11	rw	BUSCON Switch Control 0: Address windows switched immediately 1: A tristate waitstate is inserted if the next bus cycle accesses a different window than the one controlled by this **BUSCONx** register[2)]
RDYENx	12	rw	READY Input Enable 0: External bus cycle is controlled by bit field **MCTC** only 1: External bus cycle is controlled by the $\overline{READY}$ input signal
CSRENx	14	rw	Read Chip Select Enable 0: The $\overline{CS}$ signal is independent of the read command ($\overline{RD}$) 1: The $\overline{CS}$ signal is generated for the duration of the read command
CSWENx	15	rw	Write Chip Select Enable 0: The $\overline{CS}$ signal is independent of the write cmd. ($\overline{WR}$,$\overline{WRL}$,$\overline{WRH}$) 1: The $\overline{CS}$ signal is generated for the duration of the write command
RESERVED	13	r	Reserved for future use; these bits must be left at their reset values.

1) When the READY function is selected (**BUSCONx.RDYENx** = 1), only the lower 3 bits of the respective MCTC bit field define the number of inserted wait states (0-7), while the MSB of bit field MCTC is unused

2) A BUSCON switch wait state is enabled by bit **BUSCONx.BSWCx** of the address window that is left.

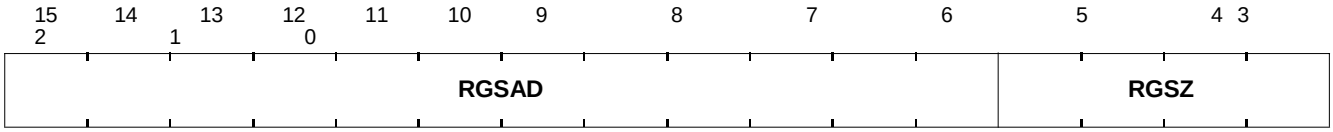
ADDRSEL1

Address Select Register 1 **Reset value: 0000_H**



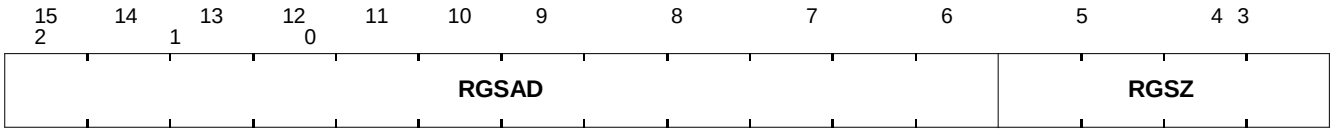
ADDRSEL2

Address Select Register 2 **Reset value: 0000_H**



ADDRSEL3

Address Select Register 4 **Reset value: 0000_H**



Field	Bits	Type	Description
RGSZ	0:3	rw	**Range Size Selection** Defines the size of the address area controlled by the respective **BUSCONx**/**ADDRSELx** register pair. Refer to **Table 44 Address Window Definition (on Page 217)**.
RGSAD	15:4	rw	**Range Start Address** Defines the upper bits of the start address of the respective address area. Refer to **Table 44**.

*Note: There is no register ADDRSEL0, as register **BUSCON0** controls all external accesses within the C166S address space but outside the four address windows of **BUSCON3**...**BUSCON1**.*

Definition of Address Areas

The three register pairs **BUSCON3**/**ADDRSEL3**...**BUSCON1**/**ADDRSEL1** allow four address areas to be defined within the address space of the C166S. Within each of these address areas, external accesses can be controlled by one of the four different bus modes, independent of each other and of the bus mode specified in register **BUSCON0**. Each **ADDRSELx** register has an address window, within which the **BUSCONx** parameters are used to control external accesses. The start address of an **ADDRSELx** window defines the upper address bits, which are not used within the address window, of the specified size (refer to **Table 44**). For a given window size, only those upper address bits of the start address (marked “R”) that are not implicitly used for addresses inside the window are used. The lower bits of the start address (marked “x”) are disregarded.

Table 44 Address Window Definition

Bit field RGSZ	Resulting Window Size	Relevant Bits (R) of Start Addr. (A12...)
0 0 0 0	4kByte	RRRRRRRRRRR
0 0 0 1	8kByte	RRRRRRRRRRx
0 0 1 0	16kByte	RRRRRRRRRxx
0 0 1 1	32kByte	RRRRRRRRxxx
0 1 0 0	64kByte	RRRRRRRxxxx
0 1 0 1	128kByte	RRRRRRxxxxx
0 1 1 0	256kByte	RRRRRxxxxxx
0 1 1 1	512kByte	RRRRxxxxxxx
1 0 0 0	1 MByte	RRRxxxxxxxx
1 0 0 1	2 MByte	RRxxxxxxxxx
1 0 1 0	4 MByte	Rxxxxxxxxxx
1 0 1 1	8 MByte	Rxxxxxxxxxx
1 1 x x	Reserved.	

Address Window Arbitration

The address windows that can be defined within the C166S address space may partly overlap each other. Thus small areas may be cut out of bigger windows, for example, to utilize external resources effectively, especially within segment 0.

For each access, the EBC compares the current address with all address-select registers (programmable **ADDRSELx** and hardwired/programmable **XADRSx**). This comparison is done in three levels. The **XADRSx** registers have the highest priority (priority I). The ADDRSEL registers have the second highest priority (priority II). If there is no match with any **XADRSx** or **ADDRSELx** register, the access to the external bus uses register **BUSCON0** (priority III):

- Priority 1:
  The **XADRSx** registers are evaluated first. A match with one of these registers directs the access to the respective X-Peripheral using the corresponding **XBCONx** register and ignoring all other **ADDRSELx** registers. Priority of the **XADRSx** registers:
  - **XADRS1** (priority I.1)
  - **XADRS2** (I.2)
  - **XADRS3** (I.3)
  - **XADRS4** (I.4)
  - **XADRS5** (I.5)
  - **XADRS6** (I.6).

- Priority 2:
  A match with one of the registers **ADDRSELx** directs the access to the respective external area using the corresponding **BUSCONx** register. Priority of the **ADDRSELx** registers:
  – **ADDRSEL2** (priority II.1)
  – **ADDRSEL1** (II.2)
  – **ADDRSEL3** (II.3).
- Priority 3:
  If there is no match with any **XADRSx** or **ADDRSELx** register, the access to the external bus uses **BUSCON0**.

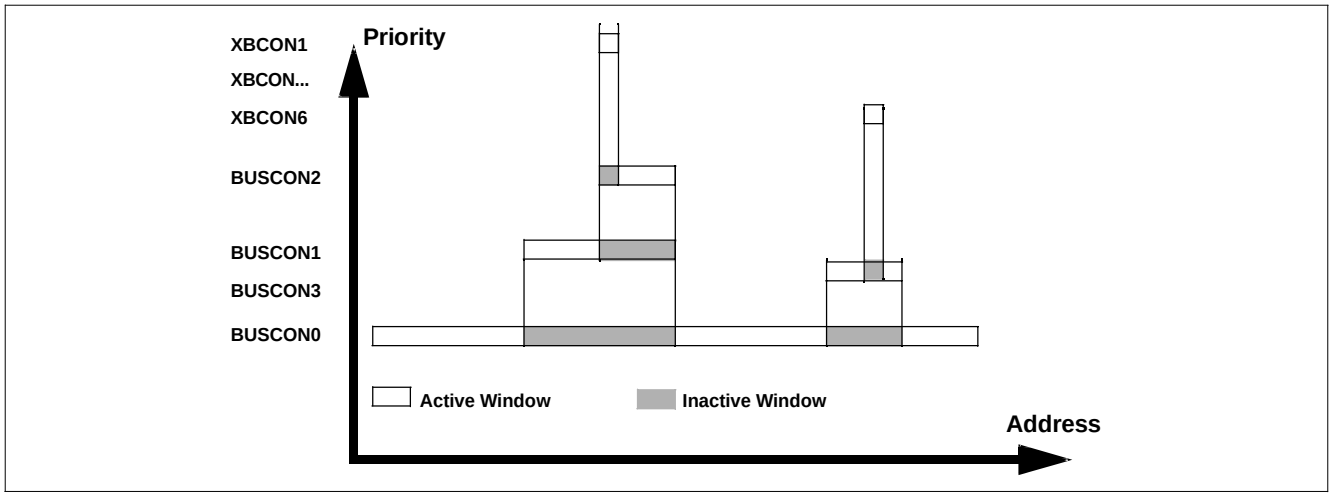


Figure 75 Address Window Arbitration Example

Precautions and Hints

The external bus interface is enabled as long as at least one of the BUSCON registers has its **BUSACT** bit set. The address windows defined via registers **ADDRSELx** may overlap internal address areas. In this case, internal accesses are executed.

For any access to an internal address area, the EBC remains inactive (refer to **Section 7.10.6 EBC Idle State**).

7.10.6 EBC Idle State

Refer to **Section 7.10 Configuring External Bus and MCU Signals (on Page 291)**.

7.10.7 Page Mode Flash Control Unit

System Integration

- Supply domain: VDD_LD1
  - Clock domain: Refer to **Section 7.2.1.3 Sub-System Clocks and Enables (on Page 67)** and see **Figure (on Page 71)**.
  - Interrupt sources:
- Monitor Pins: Refer to **Section 9.7.10 Internal Signal Monitoring (on Page 435)**.

7.10.7.1 Introduction to PMCU

Flash devices supporting page mode read accesses offer a significant reduction of access time for on-page accesses. When an access is on the same page as the previous access, the reduced on-page access time can be used.

Flash page sizes of 4 or 8 words are available.

In the address of a word:

- The upper bits are the page address
- The lowest 2 or 3 bits are the address on the page.

As long as the page address remains unchanged from the previous access, the reduced time on-page access can be used.

This Flash page mode access feature is supported on CS0 by PMCU module. In this case, the CS0_N, OE_N, ADV_N pins need to be connected to the page-mode Flash.

ADV_N and OE_N are also used for Flash devices on CS3. For Flash devices on CS3, page mode access is not suspported.

Note: During write accesses to the Flash device, the Page Mode Control Unit (PMCU) has to be disabled as write accesses are not supported while in the paging mode.

Page Mode Function

Page mode access is controlled by the PMCU shown in **Figure 76**. It uses the asynchronous READY_n_i function of the C166S core. It compares the new address coming from the C166S with the previous one and generates an asynchronous READY_n_o signal (for the C166S) depending on the access type.

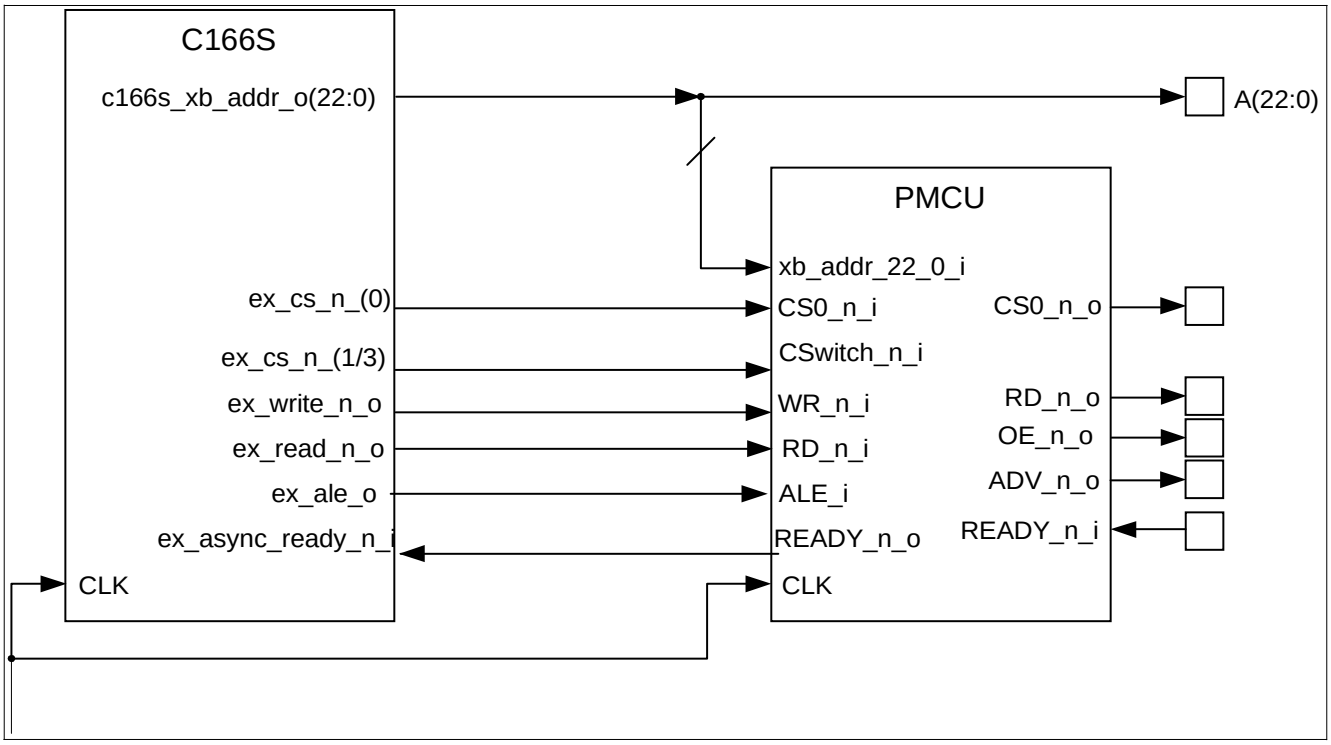


Figure 76 Page Mode Control Unit (PMCU) Block Diagram

Page Mode Switch Function

The on-page access is also used in the page mode switch case, which corresponds to the following situation:

1. The Flash is accessed in the page mode (on CS0) with the page mode timer enabled.
2. Another device is accessed for reading on CS1. In this case the signal CSwitch_n_i is activated. CS0_n_i is deactivated in parallel.
3. An on-page access is performed If the Flash is accessed again under these conditions:
   a) The new flash access on CS0 is on the same page as the last flash access on CS0 b) The page mode switch timer not yet timed-out.
   c) There was no external visible writing operation during the switched access to CS1.
   d) The CS1 should never be inactive during the switched access.

Example 1:

1. Code is fetched from the Flash on CS0 in the page mode
2. Variables are read from the external SRAM on CS1
3. It is not necessary to reopen the Flash page to continue fetching more code from the Flash (refer to the conditions in step 3 above)

Example 2:

1. Code is fetched from the Flash on CS0 in the page mode
2. Subroutines are called from the external SRAM on CS1
3. It is not always necessary to reopen the Flash page to continue fetching more code from the Flash (refer to the conditions in step 3 above).

Example 3:

1. Code is fetched from the Flash on CS0 in the page mode
2. Variables are written to the external SRAM on CS1
3. The Flash page must be reopened to continue fetching more code from the Flash.

Note: The page mode switch from CS0 to CS2 is not supported. because CS2_n_o signal is an alternative function of the OE_N pin who is necessary for flash control.

By using page mode function and page mode switch function, the controller performance can be increased up to 30% depending upon the structure of the executed code, compared to standard flash devices.

7.10.7.2 Page Mode Control Register

Page mode access is available for the address regions controlled by signals ex_cs_n_(0), see **Figure 76**. The function of the block is controlled by register **PMC0**.

The PMCU is clocked by the micro-controller clock. All access times are defined by wait states defined as multiples of a micro-controller clock cycle.

The PMCU can be configured to match the page mode flash memory specification. The PMCU allows the setting of:

- Page size
- Off-page wait states (off-page read access time)
- On-page wait states (on-page read access time).

The on-page access time is defined by the higher bit field value in either **PMC0.ONPWS** or **BUSCONx.MCTC.** If an on-page access is detected by the page mode control logic, signal READY_n_o is asserted **ONPWS** clock cycles after signal RD_n_i has been asserted.
The off-page access time is defined by the higher bit field value in either **PMC0.OFFPWS** or **BUSCONx.MCTC**. If an off-page access is detected, signal READY_n_o is asserted **OFFPWS** clock cycles after signal RD_n_i has been asserted

The C166S inserts a number of wait states (0-7) defined in **BUSCONx.MCTC** and then monitors the READY_n_o line to determine the actual end of the current bus cycle.

The transition to page mode has to be done in such a way that the PMCU is set up and enabled firstly by setting up bit **PMC0.PAEN** enabled while the memory timing is still controlled by bit field **BUSCON0.MCTC**. After register **PMC0.PAEN** has been set up, bit **BUSCON0.RDYEN** must be set and the value of wait states in bit field **BUSCON0.MCTC** can be reduced to 0 (set **MCTC** to F_H), therefore effectively profiting from the page mode feature.

To terminate the page mode, **BUSCON0.MCTC** has to be set to its original value. Then the PMCU can be disabled by resetting **PMC0.PAEN**.

To minimize power consumption, bit **PMC0.PAEN** must be kept deasserted if the page mode is not used. If bit **PMC0.PAEN** is deasserted, the PMCU is transparent, and it's a direct path from C166 EBU to chip pads. Otherwise, the C166 EBU signal ex_rd_n_o, ex_ale_o, ex_cs_n_(0) will be re-handled by PMCU, the flash output enable signal OE_N will be generated by PMCU. (See figure 10-77)

7.10.7.3 Page Mode Control Timer Registers

The page mode switch timer, **PMC_TIMER0**, controls the ACTIVE STATE LENGTH of CS0 during page mode switch. So that the flash suspending power consumption can be limited.

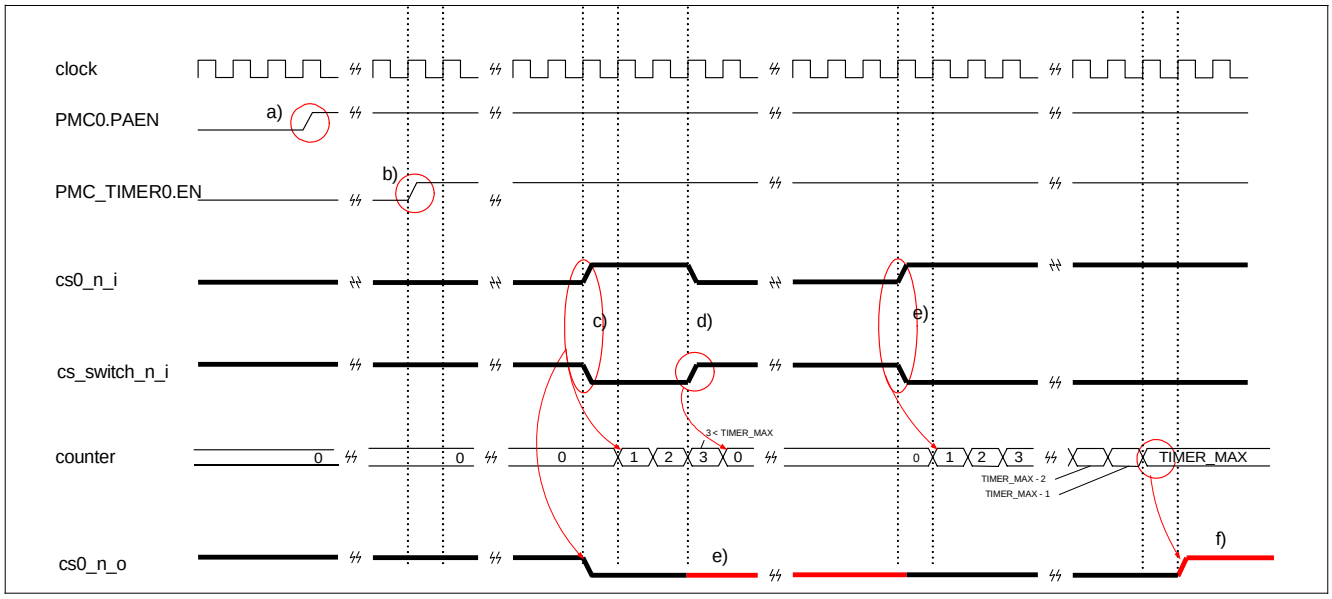


Figure 77 De-assertion of CS0_n_i Before/After Timeout

The counter of the timer starts from 1 after each falling edge of CSwitch_n_i (c,e).

The value of the counter is incremented by 1, at each MCU clock cycle.

The counter is reset to 0, when the value of the counter is equal to **PMC_TIMER0.TIMER_MAX** (f), or the page mode switch is finished (d) (CS1 deasserted), or a hardware reset is launched.

But the timeout signal of the timer is asserted only when the value of the counter is equal to **PMC_TIMER0.TIMER_MAX** (f).

Note: When **PMC_TIMER0.EN** *or* **PMC0.PAEN** *is set to 0 (disabled), the timer will not be started*

Note: Set ***PMC_TIMER0.EN*** *to:*

- 1 (enabled) after setting **PMC0.PAEN** to 1 (enabled) (a,b)
- 0 (disabled) before setting **PMC0.PAEN** to 0 (disabled).

If the page mode switch timer is used:

- The page mode switch function is enabled
- The power consumption of the flash memory device can be reduced.
- The MCU performance is improved when external Flash and RAM is accessed alternately because the page on the Flash device remains open while the RAM is accessed. Subsequent accesses by the MCU are faster due to the reduced page mode access times (refer to **Section 7.10.7.4 Page Mode Switch Details (on Page 222)**).

7.10.7.4 Page Mode Switch Details

Page Reads are initiated when either the CE# ("Chip Enable" pin of the Flash device) or the upper address bits of the Flash device change. A typical user suspends a long latency Flash operation to access another device such as SRAM. Due to the restrictions on holding the Flash CE# and address constant, it is not possible to suspend page access on regular Flash devices that only support asynchronous and page reads.

However, devices that support both Page and Burst modes allow the user to suspend page mode reads. This feature is beneficial on architectures that use many page mode reads. The ability to suspend an ongoing page read eliminates the initial access latency penalty upon resumption.

Note: The Pagemode switch feature does not work with ***SYSCON.CSCFG*** *= 0 (the normal chip-select). To use this feature, the early chip select has to be configured with* ***CSCFG*** *= 1.*

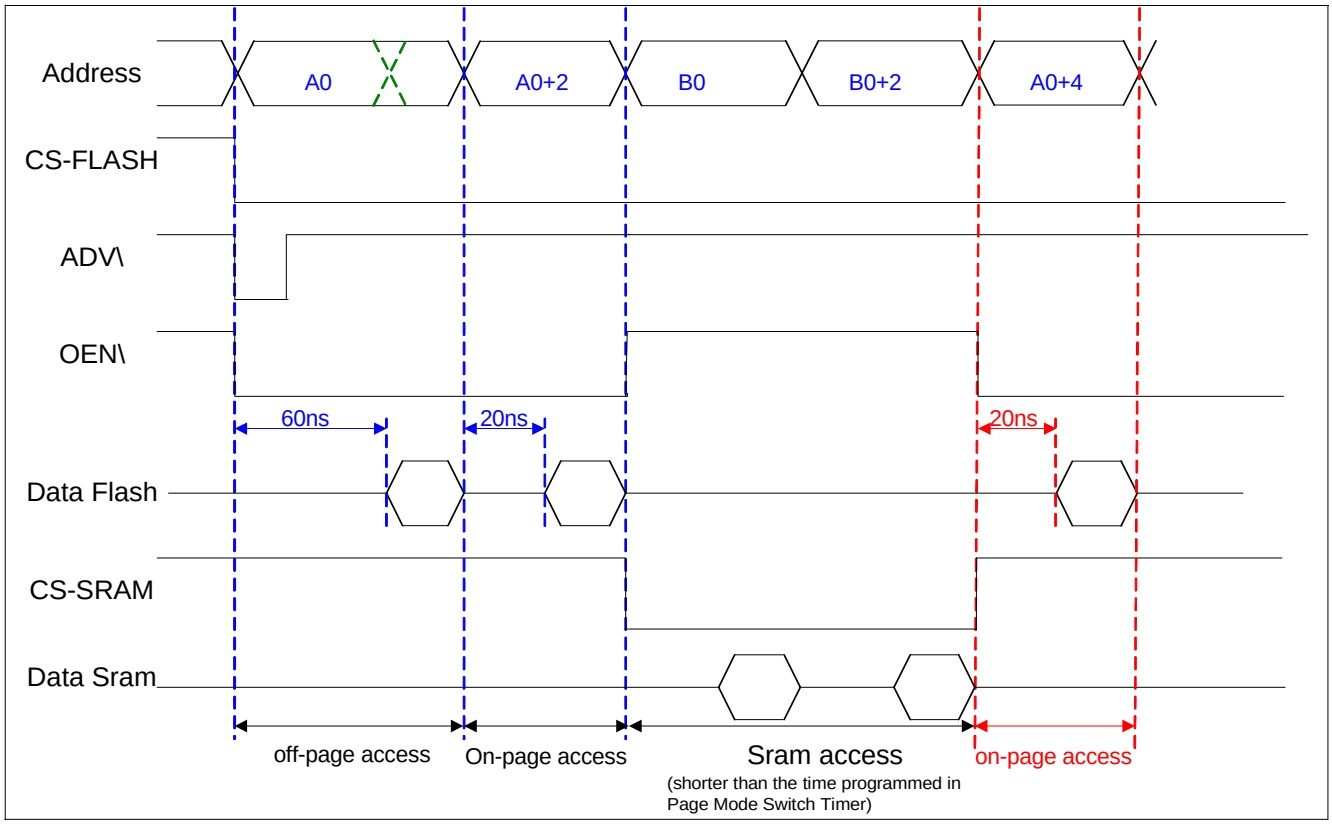


Figure 78 Page Mode Switch Function

To Suspend the Page Read

CE# must be left low to keep the Flash device active. Also OE# must be taken high to prevent the Flash device from driving the system data bus. Once this is done, the user can access another device for reading operations. The page read suspending can not apply to writing in another device. Because the write pin is shared by all external memory devices, the flash risks to be written by mistake. In the design, the page read suspending is stopped in case of any writing operations, by de-asserting the CE# of the flash during the writing.

In **Figure 79** the user enables the Flash device, drives the page read address on the address bus, drops ADV# for a specified period, and then raises it. This allows the flash to latch to the upper address bits internally. Then the user can proceed with the normal page read operation. There is an initial latency period for the first word and then a page latency period for subsequent words.

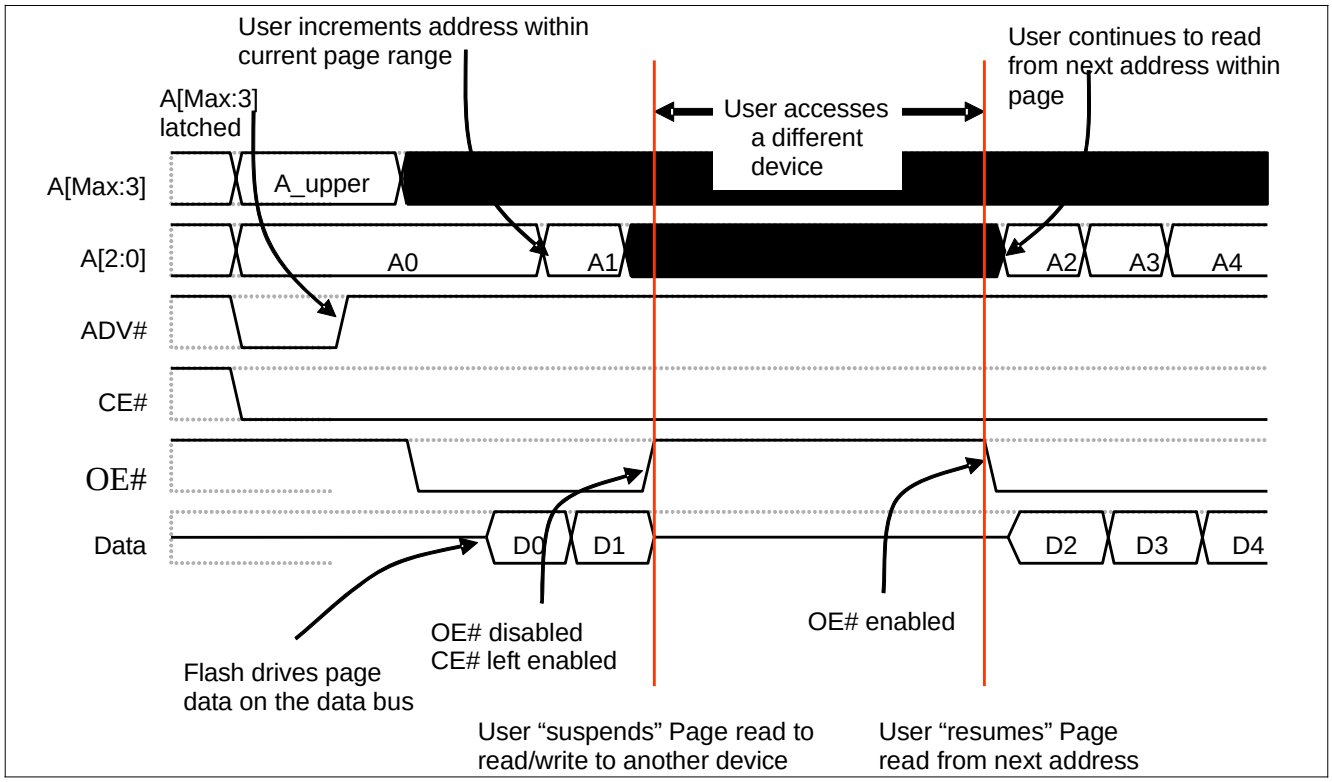


Figure 79 Page Suspend for an 8-Word Page Device

To Resume the Suspended Page Operation

OE# must be driven low again. Also the value for the resume address must be driven on the lower address bits. Upper address bits are don't care because they have already been latched internally. The user can then proceed to read through the remaining page as usual.

To Profit from the Page Mode Switch feature

The following pin connection are needed:

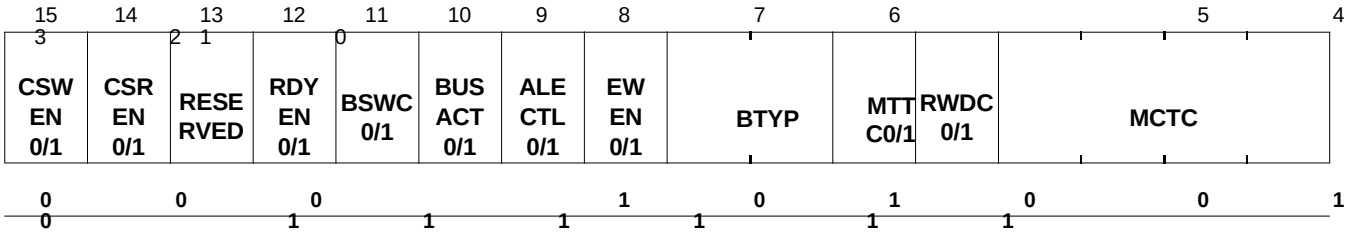
- The CE# pin of the Flash device must be connected to CS0_n_o output pin of the PMCU (via CS0_N chip pad).
- The OE# pin of the Flash device must be connected to OE_n_o output pin of the PMCU (via the OE_N chip pad).
- The ADV# pin of the Flash device must be connected to ADV_n_o output pin of the PMCU (via the ADV_N chip pad).
- The RD ("READ" pin of the RAM device) must be connected to RD_n_o output pin of the PMCU (via the RD_N chip pad).

To Get the Best Performance

For the best performance, set the Flash in the page mode with following configuration for the appropriate **BUSCONx (on Page 214)**,

A **BUSCONx** value of $14BF_H$ corresponds to the best access performance with 3 cycles.

BUSCONx
Bus Control Register 0



- **MCTC** => 0 wait states
- **RWDC** => No RD/WR delay: activate command with falling edge of ALE
- **MTTC** => Takes into account bit **MCTC**
- **EWEN** => Normal $\overline{WR}$ signal (No early write)
- **ALECTL** => Normal ALE signal
- **BSWC** => Address windows switched immediately (no tristate wait state is inserted when the window address is changed)
- **RDYEN** => External bus cycle is controlled by the $\overline{READY}$ input signal. **RDYEN** must be used to program the number of on-page and off-page wait states, which depends on the Flash specification. **RDYEN** requires an additional wait state in the C166S for internal synchronization (refer to the *C166S V1 User's Manual* (May 2002), Section 8.3.6 "$\overline{READY}$ Controlled Bus Cycles".

7.10.7.5 Timing Diagrams

Figure 80 and **Figure 81** show timing examples with an initial access, an on-page access, and an off-page access.

Notes

1. *The timing diagram in* **Figure 80** is *based on the performance to be achieved for E-GOLDvoice V3.0.*
2. *To view these figures with the PDF reader* Rotate Counterclockwise and then Zoom as needed.

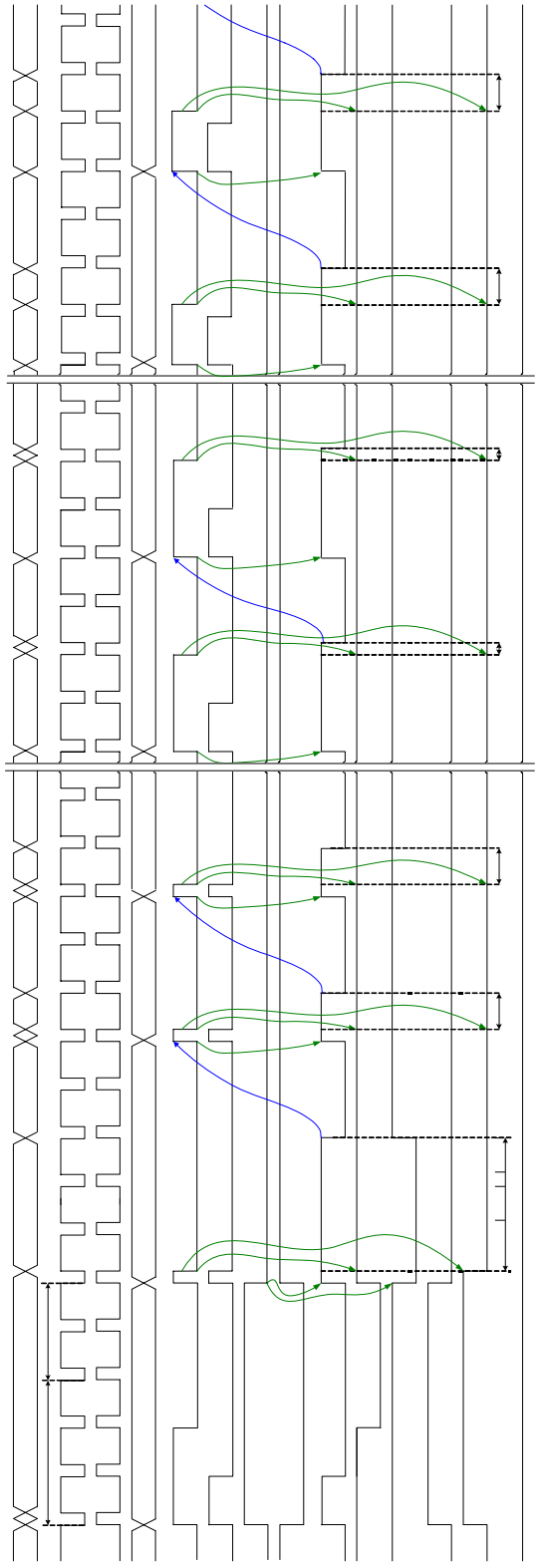


Figure 80 EBU - PMCU Configuration Effects

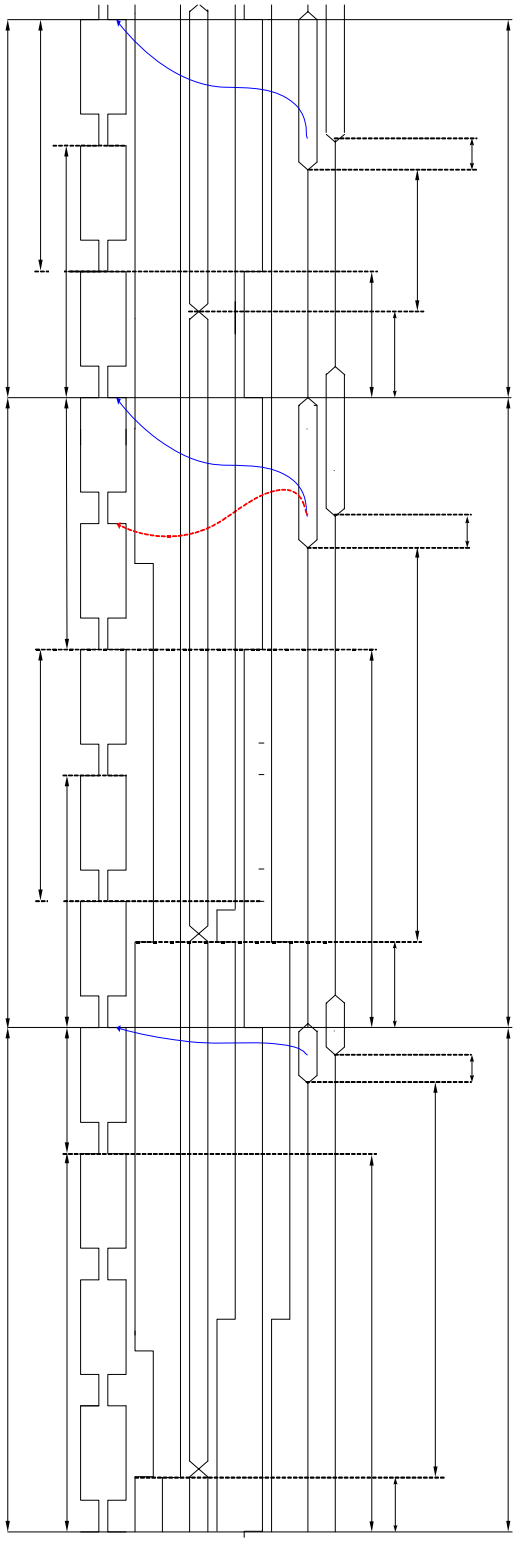


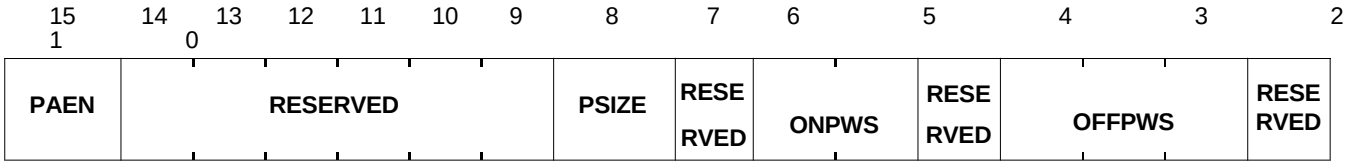
Figure 81 Example Read Access at 52 MHz

7.10.7.6 SFR Registers

7.10.7.6.1 Page Mode Control Register

PMC0

Page Mode Control 0 **Reset value: 0000_H**



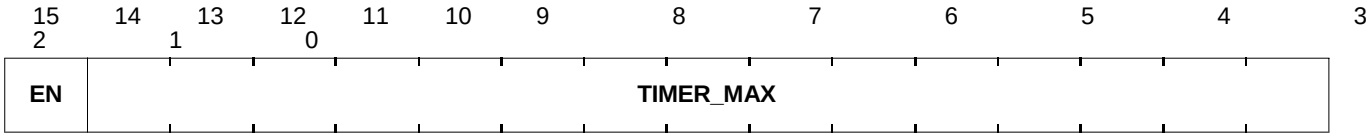
Field	Bits	Type	Description
OFFPWS	3:1	rw	**Off-Page Access Wait States** Number of wait states: 000 0 wait states 001 1 wait state 010 2 wait states 011 3 wait states 100 4 wait states 101 5 wait states 110 6 wait states 111 7 wait states
ONPWS	6:5	rw	**On-Page Access Wait States** Number of wait states: 00 0 wait states 01 1 wait state 10 2 wait states 11 3 wait states
PSIZE	8	rw	**Memory Page Size** 0 4 words 1 8 words
PAEN	15	rw	**Page Mode Enable** 0 Disabled 1 Enabled
RESERVED	0, 4, 14:9	r	Reserved for future use; these bits must be left at their reset values.

Note: The PMCU is only intended to be used for 16-bit wide, non-multiplexed, read accesses from external 16-bit wide devices.

7.10.7.6.2 Page Mode Control Timer Register

PMC_TIMER0

PMC Timer 0 **Reset value: 0000_H**



Field	Bits	Type	Description
TIMER_MAX	14:0	rw	**Timer 0 Duration** Its unit is one cycle of the PD Bus clock. The timer starts or re-starts on each falling edge of CS0_n.
EN	15	rw	**Page Mode Timer 0 Enable** 0 Disables and resets the timer to 0 1 Enables the timer (starts from 0)

Example 1

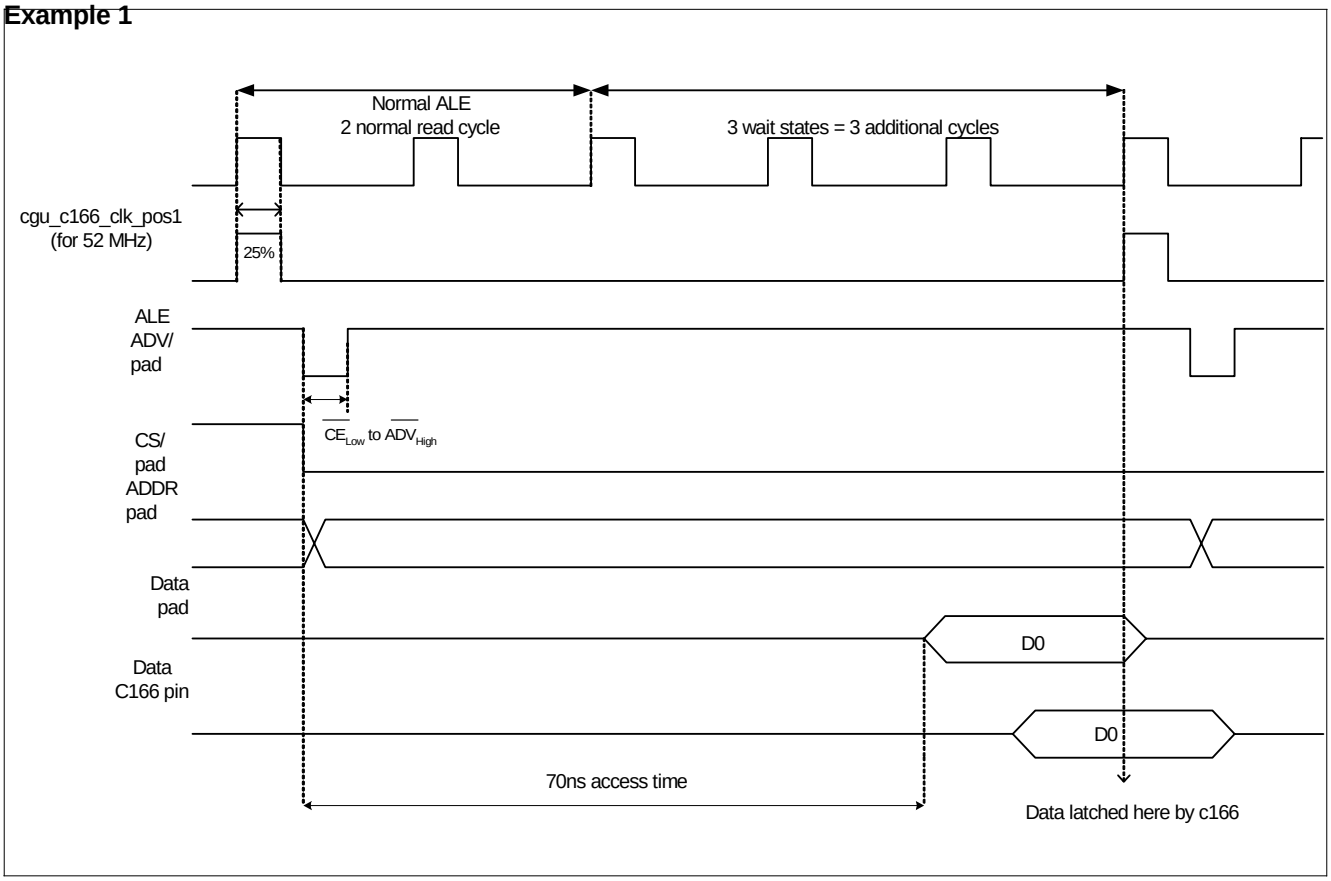


Figure 82 Random Read Access for External Flash with Normal ALE

Configuration: BUSCON = 04BC, PMC0.PAEN = 0

Note: CE/_Low_to_ADV/_High should be a minimum of 10ns (see Intel 28F320W18). This can not be guaranteed by using normal ALE and a clock frequency of 52MHz (1/52MHz / 4 = 4.8ns). In order to avoid a violation extended ALE has to be used. This configuration should be done in the boot code.

Example 2

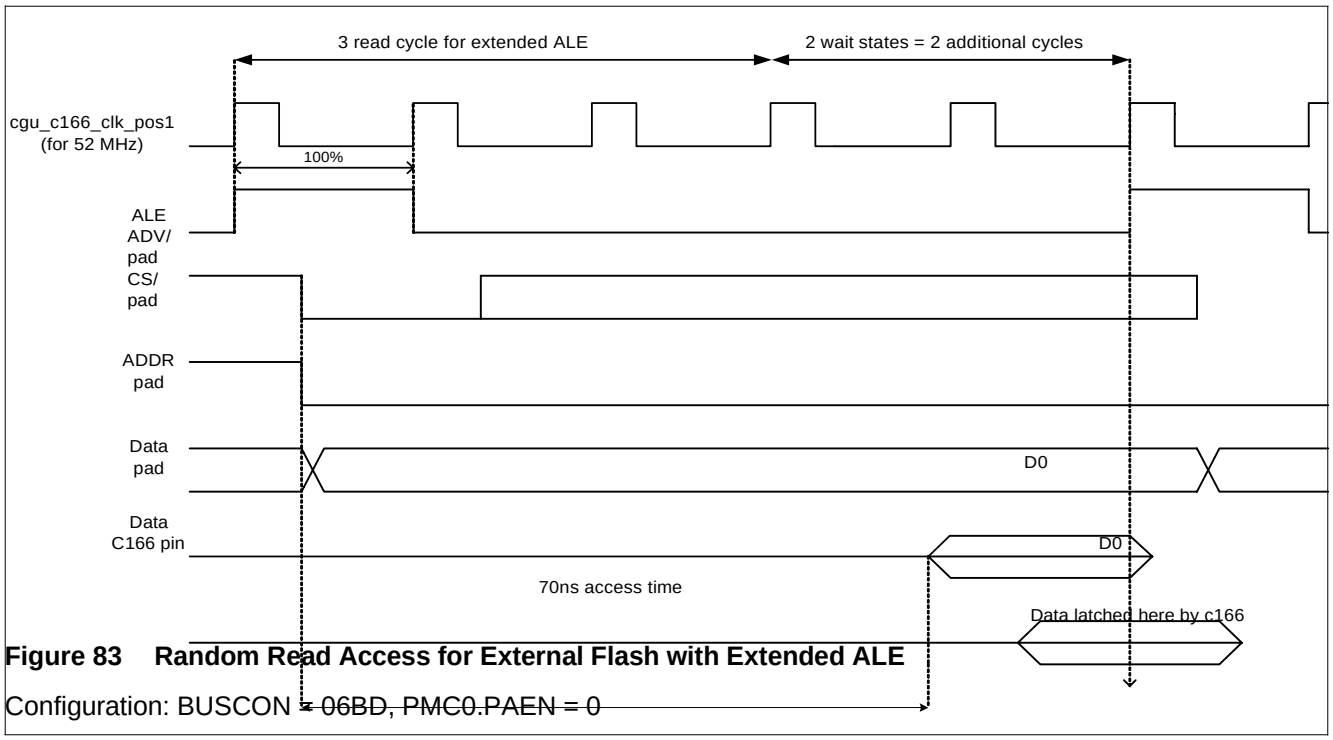


Figure 83 Random Read Access for External Flash with Extended ALE

Configuration: BUSCON = 06BD, PMC0.PAEN = 0

Example 3

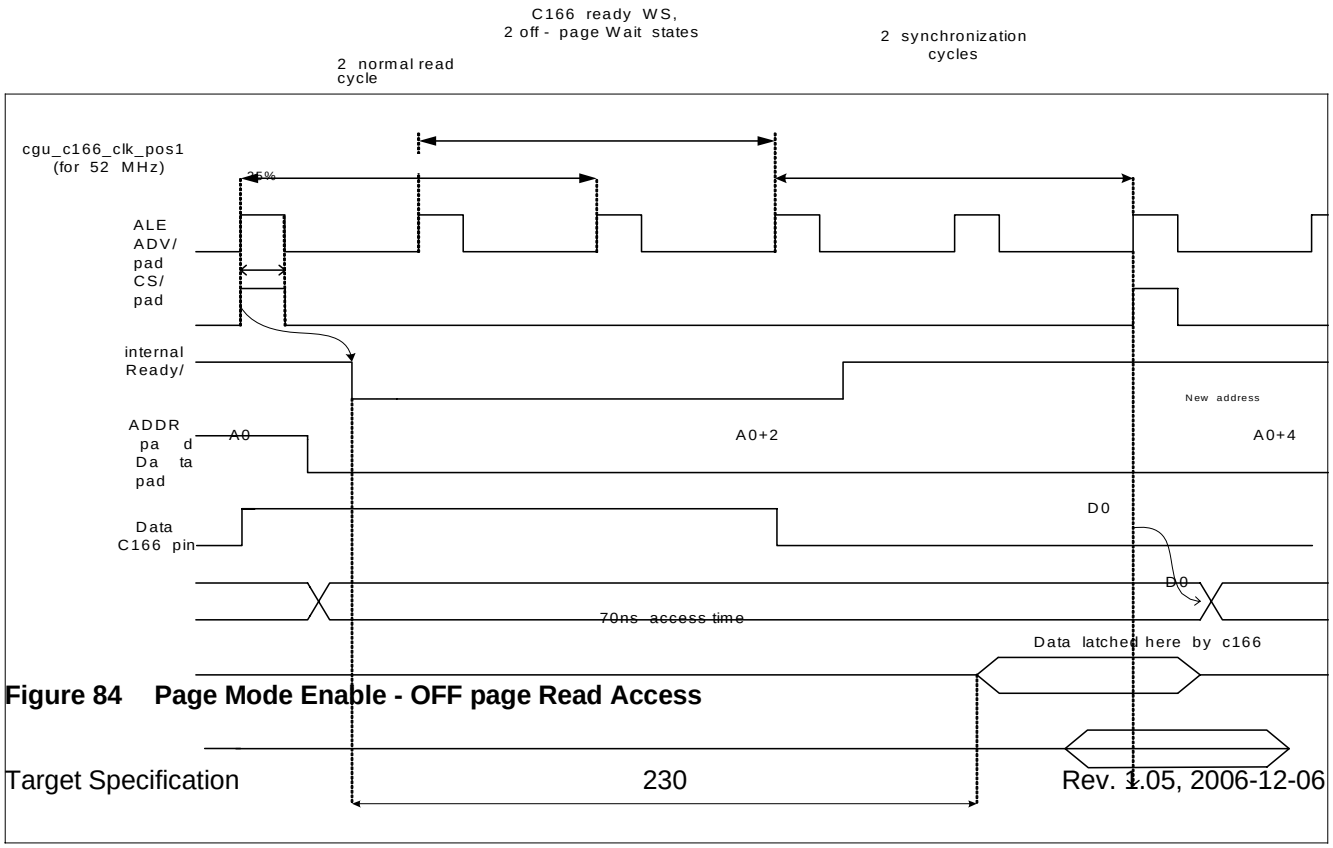


Figure 84 Page Mode Enable - OFF page Read Access

Configuration: BUSCON = 04BC, PMC0=8004

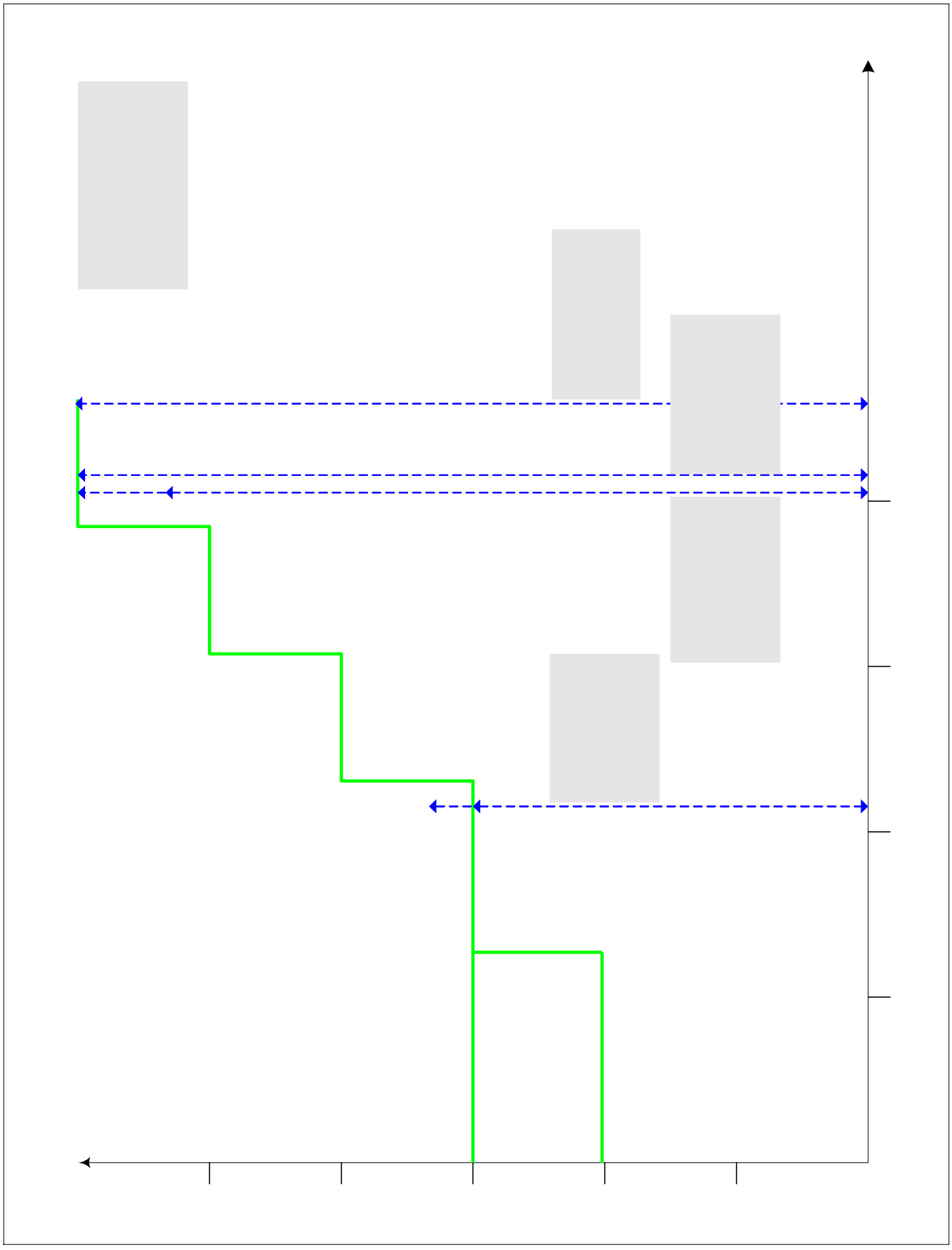


Figure 85 Wait State Computation and Performance

8 Power Management Unit

The E-GOLDvoice integrated power management unit (PMU) supports direct connection to battery (DCB), see **Figure 1 E-GOLDvoice Block Diagram**. That means all supply voltages needed are generated on-chip with integrated linear voltage regulators. The input of these linear voltage regulators is the battery voltage. The external memory and SIM card supply is provided by the on-chip voltage regulators. **Table 45** is an overview of the internal generated supply voltages.

Table 45 Internal Generated Supply Voltages by LDOs

Name	Output Voltage (V)	Output Current (mA)	Comment
LRTC	2.0	4	Used for the real time and digital PMU supply
LD1	1.2 / 1.5	150	Used for the core supplies (MCU and DSP via switch)
LIO	1.8 / 2.85	30	Used for the I/O pad supply and, for example, the display
LRFXO	2.5	10	Used for the crystal oscillator supply
LMEM	1.8 / 2.85	100	Used for the external memory supply, voltage can be configured during startup
LANA	2.5	100	Used for analog (audio and baseband processing) and headset driver
LSIM	1.8 / 2.85	30	Used of the SIM card supply
LBUF	2.6 / 2.8 / 3.0 / 3.2	300	Used for the loudspeaker and earpiece driver
LRFRX	2.5	100	Used for the RF RX part
LRFTRX	1.5	120	Used for the RF RX/TX part

The integrated power management also provides the control state machine for system start up, including start up with discharged batteries, trickle charging and system reset control.

After system start up several methods are implemented for active and idle power saving.

8.1 Features Overview

LDO output voltage selection

- LD1, LIO, LSIM, LBUF output voltage programmable by software.
- LMEM output voltage is selectable by pin configuration upon startup.

Active and idle power saving options:

- The flexible clock switching options allow minimizing the power consumption during the operation phases of the E-GOLDvoice.
- Current consumption during the standby mode is minimized by reducing the clock to 32 kHz and switching it off for most of the device. In addition, the power supply for the TEAKLite ROM is switched off and the controller RAM is switched to a power saving mode.

Start-up and Reset Control State Machine Features

- Power up upon battery insertion, push button, alarm, charger connection.
- Detection of battery exchange or re-insertion.
- Complete start-up sequence management.
- System turn-on, system turn-off operation management including emergency (under-voltage) and programmed shutdown functions.
- Internal reset of the baseband, including silent reset.
- Tristate function of the baseband module.
- Standby mode controlled by VCXO_EN provided by SCCU module.

Charger Features

- Switched charging (charge current/voltage adjusted in the charger unit)
- Charger detection
- Battery over-voltage detection, battery voltage monitoring
- Power-on reset
- IC over-voltage protection
- Pre-charging (ex: for deep discharged batteries)
- Software controlled charging.

The charger unit controls the charging of NiCd, NiMh, LiPolymer and LiON batteries. Only a few external parts are required to support half wave, full wave and electronic chargers. In addition, the charger generates the power-on reset after battery insertion or charger connection. The supported battery voltage range is 3.1 to 5.5 V for NiCd/NiMH and 3.1 to 4.6 V for LiPolymer and LiON batteries. The charger supports precharging, full-charging and trickle charging protected by timeouts. Charger idle voltages up to 20 V are supported. An integrated overvoltage protection protects batteries and system against overvoltage.

8.1.1 PMU Register Overview

This section specifies the E-GOLDvoice PMU control and status registers. Control registers provide read/write access; the status registers only provide read access. The PMU control and status registers are accessed via the C166S X-Bus controller.

8.1.1.1 PMU Registers

Table 46 PMU Registers

Reg. Name	Description
PMU_GENCTRL[1)]	General Control Register
PMU_PWRCTRL1[1)]	Power Control Register 1
PMU_PWRCTRL2[1)]	Power Control Register 2
PMU_LPDCTRL[1)]	LDOs PullDown Control Register
PMU_CHGCTRL[1)]	Charge Control Register
PMU_INTCTRL[1)]	Interrupts Control Register
PMU_ID[1)]	PMU Peripheral ID Register
PMU_STAT[1)]	Startup and Interrupts Status Register

1) SW user need to consider that those bits are written with a 26 MHz cycle (when code is executed), **but sampled with a 32/50 kHz period**. Hence setup (duration of signal) shall be long enough so that it can be captured by 32 kHz clock (period 31.2 us).

8.1.1.1.1 Control Registers

PMU_GENCTRL

General Control Register **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES PDN	ALL OFF	RESERVED													RES

Field	Bits	Type	Description
RES	0	w	**Internal Reset of the Baseband** 0 No reset (default) 1 Generate the internal reset to the baseband
ALLOFF	14	w	**Complete Shut Down** 0 No shutdown (default) 1 PMU shuts down completely (to system OFF state)
RESPDN	15	w	**Reset in Power Down** 0 Digital pads do NOT go to a reset state on power down (default) 1 Digital pads do go to a reset state on power down
RESERVED	13:1	r	Reserved; these bits must be left at their reset values.

RES =1 is a software programmed reset and will lead to a Baseband reset.

It is not a silent reset. A silent reset can be programmed on the MCU.

PMU_PWRCTRL1

Power Control Register 1 **Reset Value: $0C63_H$**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LRFRX MD		LRFTRX MD		LRFXO MD		LSIM V	LSIM MD		LMEM MD		LANA MD		RESE RVED	LD1 MD	

Field	Bits	Type	Description
LD1MD	1:0	rw	**LD1 Mode** 00 Off (test function only, shall not be used in real application) 01 In standby if VCXOEN is low 10 Reserved 11 Always on, no standby-function of LDO (default)
~~RESERVED~~	~~2~~	~~rw~~	~~Reserved, this bit has to be cleared always~~
~~LANAMD~~	~~4:3~~	~~rw~~	~~**LANA Mode**~~ 00 Off (default) 01 On if VCXOEN is high, Off if VCXOEN is low 10 On if VCXOEN is high, standby if VCXOEN is low 11 Always on, no standby-function of LDO
~~LMEMMD~~	~~6:5~~	~~rw~~	~~**LMEM Mode**~~ 00 Off 01 On if VCXOEN is high, Off if VCXOEN is low 10 On if VCXOEN is high, standby if VCXOEN is low 11 Always on, no standby-function of LDO (default)

Field	Bits	Type	Description
LSIMMD	8:7	rw	**LSIM Mode** 00 Off (default) 01 In standby if VCXOEN is low 10 Off 11 Always on, no standby-function of LDO
LSIMV	9	rw	**LSIM Output Voltage** 0 1.80 V (default) 1 2.85 V
LRFXOMD	11:10	rw	**LRFXO Mode** 00 Off 01 On if VCXOEN is high 10 Off 11 Always on, no standby-function of LDO (default)
LRFTRXMD	13:12	rw	**LRFTRX Mode** 00 Off (default) 01 On if VCXOEN is high 10 Off 11 Always on, no standby-function of LDO
LRFRXMD	15:14	rw	**LRFRX Mode** 00 Off (default) 01 On if VCXOEN is high 10 Off 11 Always on, no standby-function of LDO

PMU_PWRCTRL2

Power Control Register 2 **Reset Value: 0478_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSC PD	CLK SEL	RESE RVED	RESERVED		RESE RVED	RESE RVED RVED	RESE RVEDI	RESE	RESE RVED	LIO V	LIO MD		LBUF V		LBUF MD

Field	Bits	Type	Description
LBUFMD	0	rw	**LBUF Mode** 0 Off (default) 1 On
LBUFV	2:1	rw	**LBUF Output Voltage** 00 2.6 V (default) 01 2.8 V 10 3.0 V 11 3.2
V LIOMD	4:3	rw	**LIO Mode** 00 Off 01 On if VCXOEN is high, Off if VCXOEN is low 10 On if VCXOEN is high, standby if VCXOEN is low 11 Always on, no standby-function of LDO (default)

Field	Bits	Type	Description
LIOV	5	rw	**LIO Voltage** ES1: 0 2.85V 1 1.8V (default) ES1+: 0 1.8V 1 2.85V (default)
RESERVED	12:6	r	Reserved; these bits must be left at their reset values.
RESERVED	13	r	Reserved; this bit must be left at their reset values.
CLKSEL	14	rw	**PMU running clock selection** 0 From PMU local oscillator (default) 1 From RTC 32KHz clock
OSCPD	15	rw	**PMU oscillator power down** 0 Local oscillator is active (default) 1 Local oscillator is powered down

The **PMU_LPDCTRL** controls the pull down resistors in the output of the LDOs. If the pull down's are enabled, the pull down's are activated if an LDO is switched off. That means the output voltage will be faster discharged after switching off an LDO.

PMU_LPDCTRL

LDO Pull Down Control Register **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							LRF RX PD	LIO PD	LBUF PD	LRF TRX PD	LRF XO PD	LSIM PD	LMEM PD	LANA PD	LD1 PD

Field	Bits	Type	Description
LD1PD	0	rw	**LD1 Pull Down Enable** 0 Not enabled (default) 1 Enabled
LANAPD	1	rw	**LANA Pull Down Enable** 0 Not enabled (default) 1 Enabled
LMEMPD	2	rw	**LMEM Pull Down Enable** 0 Not enabled (default) 1 Enabled
LSIMPD	3	rw	**LSIM Pull Down Enable** 0 Not enabled (default) 1 Enabled
LRFXOPD	4	rw	**LRFXO Pull Down Enable** 0 Not enabled (default) 1 Enabled

Field	Bits	Type	Description
LRFTRXPD	5	rw	**LRFTRX Pull Down Enable** 0 Not enabled (default) 1 Enabled
LBUFPD	6	rw	**LBUF Pull Down Enable** 0 Not enabled (default) 1 Enabled
LIOPD	7	rw	**LIO Pull Down Enable** 0 Not enabled (default) 1 Enabled
LRFRXPD	8	rw	**LRFRX Pull Down Enable** 0 Not enabled (default) 1 Enabled
RESERVED	15:9	r	Reserved; these bits must be left at their reset values.

PMU_CHGCTRL

Charge Control Register **Reset Value: 0014_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						RESERVED	CHGONCLR	FCONCLR	CDTS	CHG	PCHG	LONS	OVEN	OVL	

Field	Bits	Type	Description
OVL	1:0	rw	**Charger over voltage threshold** 00 5.5 V (default) 01 4.57 V 10 4.47 V 11 Not used (5.5 V)
OVEN	2	rw	**Over voltage shutdown feature of the charger is** 0 Disabled for debug only, SW stop 1 Enabled (default)
LONS	3	r	**Status of ON-key** 0 On key not pressed at the moment (default) 1 On key currently pressed
PCHG	4	rw	**HW controlled pre-charging is** 0 Forces precharging off, enables software charging, stops SWCT supervision after first start up, clears SWCT overflow bit, stops and resets the 60min precharge time-out counter 1 On (default) Hardware precharging is enabled, bit CHG can not be used

Field	Bits	Type	Description
CHG	5	rw	**SW controlled charging** This bit can only be used if bit PCHG is cleared. 0 SW charging Off, charge switch is not conducting 0->1 A three second timer (SWCT) is started, generating for three seconds the 99% waveform for full charge current. For continuous charging the 0->1 level change has to be redone before a SWCT overflow. If there is no 0->1 level change during this three seconds SWCT run, charging is stopped after SWTC overflow for security reasons. 1 The 99% waveform is stopped after 3 seconds (SWCT overflow) 1->0 If occurs during the 3 second timer (SWCT): charging is stopped immediately
CDTS	6	r	**Status of CDT input** 0 Currently no charger unit detected (default) 1 Currently a charger unit is connected
FCONCLR	7	w	**FCON bit clear** 0 Do not clear (default) 1 Clear *Note: Always read as 0.*
CHGONCLR	8	w	**CHGON bit clear** 0 Do not clear (default) 1 Clear *Note: Always read as 0.*
RESERVED	9	rw	Reserved; these bits must be left at their reset values.
RESERVED	15:10	r	Reserved; these bits must be left at their reset values.

PMU_INTCTRL

Interrupt Control Register **Reset Value: 0000_H**

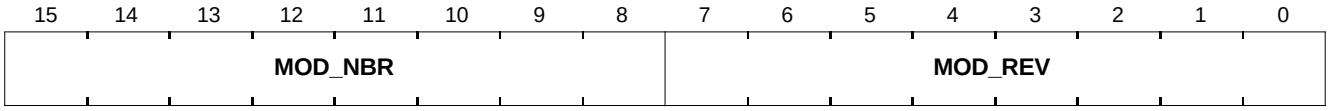
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCON CLR	RESERVED									LON CLR	OV CLR	CDT CLR	LON IE	OV IE	CDT IE

Field	Bits	Type	Description
CDTIE	0	rw	**CDT Interrupt Enable (charger unit was detected)** 0 Disabled 1 Enabled
OVIE	1	rw	**OV Interrupt Enable** 0 Disabled 1 Enabled
LONIE	2	rw	**LON Interrupt Enable (on key press)** 0 Disabled 1 Enabled
CDTCLR	3	w	**CDT Interrupt clear bit** 0 Do not clear interrupt 1 Clear interrupt *Note: Always read as 0.*

Field	Bits	Type	Description
OVCLR	4	w	**OV Interrupt clear bit** 0 Do not clear interrupt 1 Clear interrupt *Note: Always read as 0.*
LONCLR	5	w	**LON Interrupt clear bit** 0 Do not clear interrupt 1 Clear interrupt *Note: Always read as 0.*
RTCONCLR	15	w	**RTCON bit clear** 0 Do not clear (default) 1 Clear *Note: Always read as 0.*
RESERVED	14:6	r	Reserved; these bits must be left at their reset values.

PMU_ID

PMU peripheral ID Register **Reset Value: $B000_H$**

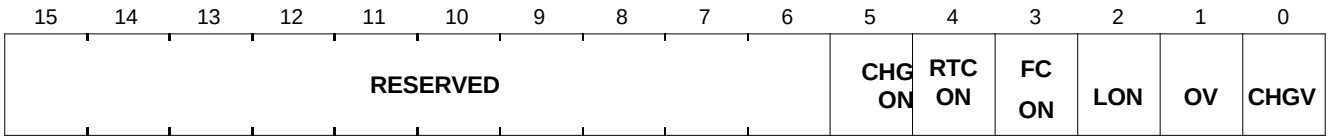


Field	Bits	Type	Description
MOD_REV	7:0	r	**Module Revision Number** Value = 00
MOD_NBR	15:8	r	**Module NUmber** Value = B0

8.1.1.1.2 Status Registers

PMU_STAT

Startup and Interrupts Status Register **Reset Value: 0000_H**



Field	Bits	Type	Description
CHGV	0	r	Change of charger input detection state *Note: **CHGV** is a source of interrupt if enabled.*
OV	1	r	**Over-voltage protection** 0 No charging interruption by an over-voltage condition 1 When charging has been interrupted by an over-voltage condition *Note: Example: When the battery voltage exceeds the limit programmed in the **PMU_CHGCTRL** register.* *Note: **OV** is a source of an interrupt if enabled.*
LON	2	r	**Change of Level On Pin ON** 0 No change occurred on level of pin ON (1 Change occurred on level of pin ON *Note: **LON** is a source of an interrupt if enabled.*
FCON	3	r	**Battery Insertion (First Connect) as Source of Startup** 0 Last start up was not caused by a battery insertion 1 Last start up was caused by a battery insertion
RTCON	4	r	**RTC Alarm as Source of Startup** 0 Last start up was not caused by a RTC alarm 1 Last start up was caused by a RTC alarm
CHGON	5	r	**Charger Detection as Source of Startup** 0 Last start up was not caused the connection of the charger 1 Last start up was caused by the connection of the charger
RESERVED	15:6	r	Reserved; these bits must be left at their reset values.

8.1.2 System Power On/Off and Reset Control Logic

When the battery voltage is below VBATon or the mobile phone is in shutdown mode, the integrated PMU in E-GOLDvoice has control over the whole system. Control is well defined in any condition by fixed state machine structures. Operation of the state machines are shown in **Figure 86** and **Figure 87**.

8.1.2.1 Tristate Function of Baseband Module

The power down tristate function isolates the outputs of the Baseband module from its environment, whenever no proper operation of the outputs can be guaranteed. It ensures that, when the battery is inserted into the handset, most output pins are locked in tri-state. The tri-state function ensures that the chip is isolated from the board but, depending on the pull-up or pull-down controls, it may not result necessarily in a Hi-Z state.

This functions is controlled by the BaseBand module input signals RESET_BB_N and PM_INT (refer to **Figure 47**)

Table 47 Tristate Function of the Baseband

RESET_BB_N	PM_INT Function	Value	Outputs State
0	**output tristate control**	**0**	**reset**
0	**output tristate control**	**1**	**tristate**
1	**PMU ext interrupt**	**X**	**X**

8.1.2.2 Insertion of Battery

When the battery is initially inserted into the handset a battery supervision circuit controls the subsequent activation of the power up state machines. A simple state machine determine if battery voltage is sufficent to start up the main PMU power-on state machine as shown below in **Figure 86**. A special state is introduced for IC testing. This state is entered based on the battery voltage 2.0V and on the state of several test signals.

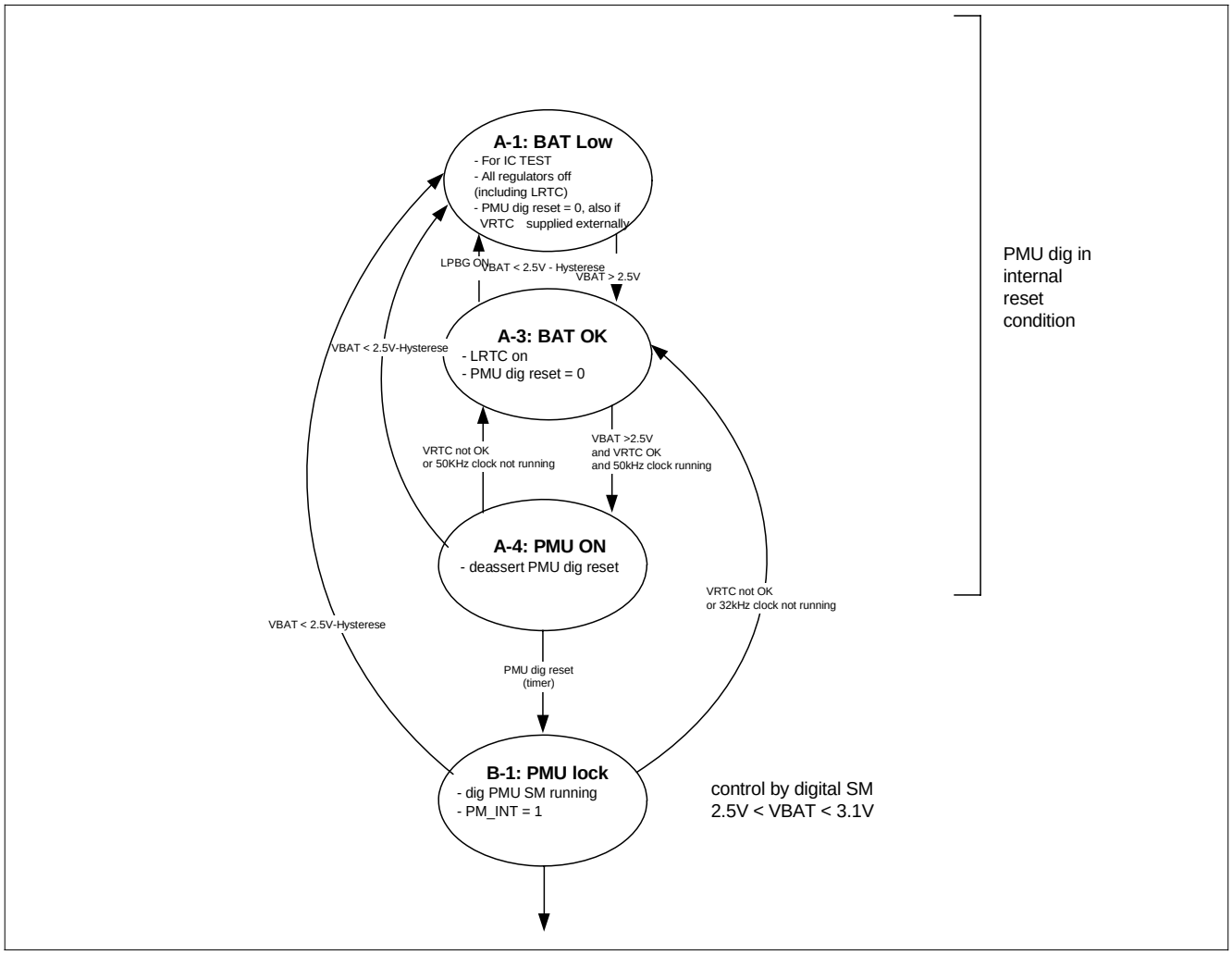


Figure 86 Battery Supervision State Diagram

When the battery voltage VBAT rises, regulator RTC is powered up, the PMU local oscillator starts running and control is transferred to the main PMU power-on control state machine:

1. Battery insertion
2. VBAT rises and power up RTC regulator
3. signals RESET_BB_N = 0 and PM_INT rises to 1 force the outputs of baseband in tristate mode.

The output of the local oscillator is used to generate the timing of the subsequent activation of the other regulators when the ON input is activated or a valid charger voltage has been detected. The control logic for the main PMU power-on state machine is located in the RTC power domain. The state transitions and conditions are shown in **Figure 87**

The main PMU power-on state machine forces a system power-on in case of battery insertion. This is done to allow the battery management software to detect when the battery has been exchanged or re-inserted. This information is used to reset timers used for battery capacity estimation in some estimation concepts, since these are no longer valid if a new battery is inserted. The state of this first connect feature can be read from the **PMU_STAT**.**FCON** bit. If this bit is set, then the system power on was caused by battery insertion.

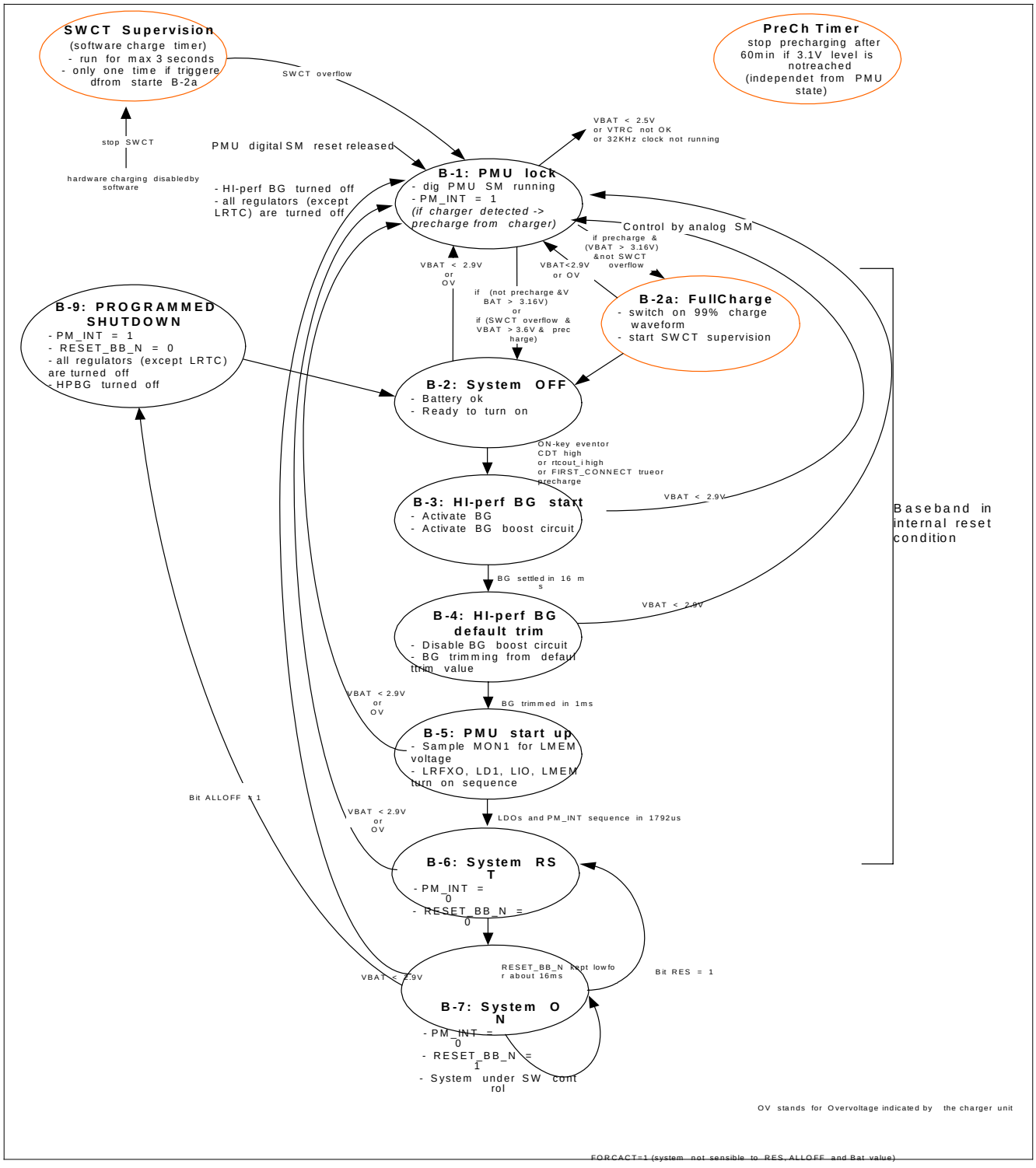


Figure 87 Main PMU Power On State Diagram

8.1.2.3 Precharge

When a battery with a voltage below the minimum operating voltage of the regulators is inserted, a special hardware controlled pre-charge mechanism ensures that the battery will be charged (when a charger is connected) to a voltage sufficient for system operation. The pre-charge circuit uses a pulsed charge scheme to generate a net charge current that is within the limits recommended by battery manufacturers for a safe pre-

charge. The pre-charge circuit contains its own voltage regulation based on a shunt regulator behind the CDT pin. The circuit also has its own oscillator for generation of the charge pulse frequency. The operation of the pre-charge circuit is controlled by an independent state machine to allow precharging (when a charger is connected) even when the battery voltage is at 0 V.

The PMU state-machine gets the information that pre-charge is running from a control precharge signal coming from the charger unit. This precharge signal is debounced in a small counter to have a stable signal. This is important, especially in half-wave charging where the charger detection is switching between charger detected/not detected according the AC supply frequency. The precharge signal is used in the state **B-1** to trigger the pre-charge functionality.

If the PMU starts the first time because of precharging, the state machine changes to state **B-2a** if the battery voltage exceeds the 3.16 V level. Now the PMU-state machine changes to full charge by applying the 99% duty cycle waveform. For security reasons it starts in parallel the SWCT (software charge timer). The SWCT supervision is switched on by state **B-2a**. The PMU-statemachine now goes to state **B-2** and the system starts. Bit PCHG in register PMU_CHCTRL is cleared - hardware charging disabled - the SWCT supervision is stopped. That means a SWCT overflow is only detected by the supervision if the software has not switched from a hardware control to software control within the first 3 seconds after startup. If a SWCT overflow is detected by the supervision, the PMU state-machine goes back to state B-1 and falls back to precharge mode (if the charger is still connected, indicated by the charger circuit). If supervision is active a SWCT overflow generates a state change to state B-1. Nevertheless the detected SWCT overflow is stored and now the pre-charge behavior changes compared to the behavior for first time startup. The PMU does not enter state B-2a any more but stays in state B-1 until the battery voltage is higher than the 3.6V level. Then it goes directly to state B-2 and the system starts. The stored SWCT overflow is cleared if bit PCHG in register PMU_CHCTRL is cleared.

An additional security feature for precharging is the 60min precharge timer. This timer is started always if the precharge condition is entered (indicated by the signal used for the CDTS bit, that means stable also if half-wave charging is used). This timer is running in parallel to the statemachine and stops pre-charging as long as bit PCHG in register PMU_CHCTRL is set (set to hardware charging) and a precharge timer overflow has been occured. The counter is reseted if bit PCHG is cleared. If PCHG is set once again the timer and pre-charging runs again for 60minutes.

8.1.2.4 Power-up Sequence

In order to avoid an excessive drop on the battery voltage caused by in-rush current during system power-on, possibly leading to system instability and “hick-ups”, a staggered turn-on approach for the regulators is implemented. The regulators are turned on in a well defined sequence, thus spreading the in-rush current transients over time.

The core of the E-GOLDvoice is held in reset state before enabling the I/Os. This avoids uncontrollable output signals during power-on. Inside E-GOLDvoice the reset logic is part of the RTC supply domain which is always powered up. This allows to power up the baseband core regulator and wait for the core to reach reset state before powering up the I/O supply regulators.

The power up sequence is initiated in one of four ways:

- A battery with a valid voltage has been inserted
- Input pin ON goes low to high caused by external push button
- Input RTCOUT goes low to high caused by a programmed alarm in the RTC macro
- A valid charger voltage is detected on pin CDT
- precharging is running and the battery voltage is above the 3.6V level

The root cause for powering up the system is recorded in the **PMU_STAT** register.

The startup time of the XO oscillator of 16ms in worst case has to be guaranteed between the power-up of the LRFXO and the release of the RESET_BB_N.

The complete Power-up sequence is shown in **Figure 88**.

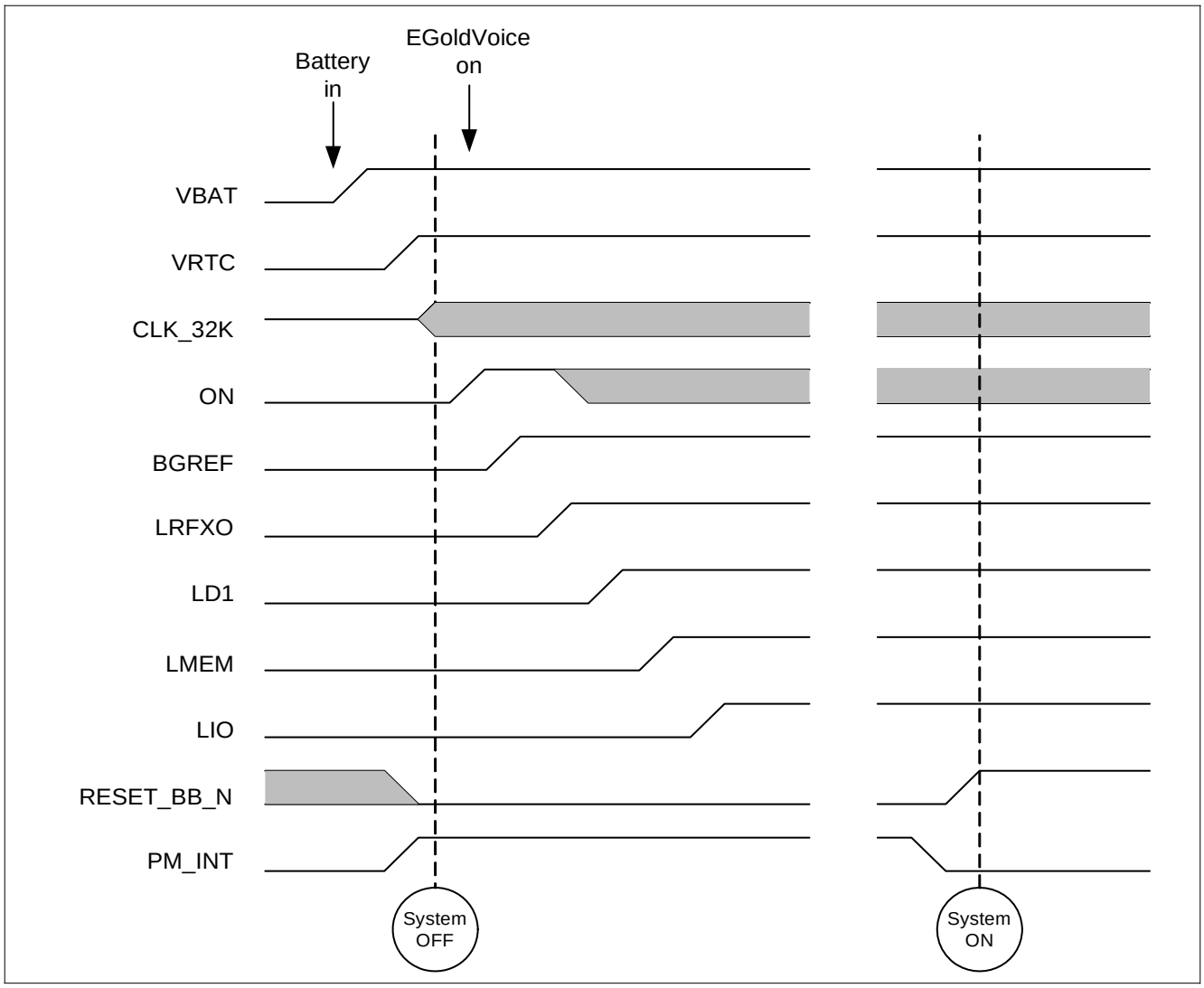


Figure 88 Start Up Sequence

Table shows the complete details of the power up timing including the delays between each step. Delay after sequence 2 is depending on a RC-constant, but after sequence 3, the reset counter starts operation (accuracy determined by the RTC oscillator). All regulators not listed in **Table** have to be turned on by software afterwards. The timings and number of cycles are based on the local 50 kHz osc plus 20% tolerance, ie with a cycle time of 16.66 us. The timings are the minimums, which have to be guaranteed.

Table 48 Turn-On sequence of Power Supply Functions

Sequence	Function	Ball	Typical time for activation of next sequence step
1	State “System OFF” LRTC active 50kHz oscillator running PM_INT high RESET_BB_N low	LRTC	-
2	ON input activated	ON	-

Table 48 Turn-On sequence of Power Supply Functions

Sequence	Function	Ball	Typical time for activation of next sequence step
3	Bandgap circuit activated		1024 cycles (min 17ms)
4	Bandgap circuit trimmed with default value		64 cycles (min 1ms)
5	LRFXO activated	LRFXO	8 cycles (min 133us)
6	LD1 activated	LD1	32 cycles (min 533us)
7	LMEM activated	LMEM	8cycles (min 133us)
8	LIO activated	LIO	64 cycles (min 1ms)
9	PM_INT low	PM_INT	1024 cycles (min 17ms)
10	RESET_BB_N released State “System ON”		

Table 49 Min/Typ/Max Startup time from VBAT rise to Baseband reset

Steps	Cycles	MIN 40 kHz osc cycle= 25 us	TYP 50 kHz osc cycle=20 us	MAX 60 kHz osc cycle=16.66 us
VBAT rise to HPBG start	7			
HPBG settling time	1024			
HPBG trimming	64			
LDOs activation	112			
RESET to SYSTEM ON	1024			
Total	2231	55775 us	44620 us	37168 us

8.1.2.5 Turn Off

A programmed system turn off is triggered by setting the **PMU_GENCTRL.ALLOFF** bit. After a turn-off event has been triggered the signal PM_INT is set high and after one clock cycle RESET_BB_N is set low. This forces the digital pins to tristate mode. After one further counter cycle all power supplies except LRTC are turned off.

The tri-state function ensures that the chip is isolated from the board but, depending on the pull-up or pull-down controls, it may not result necessarily in a Hi-Z state.

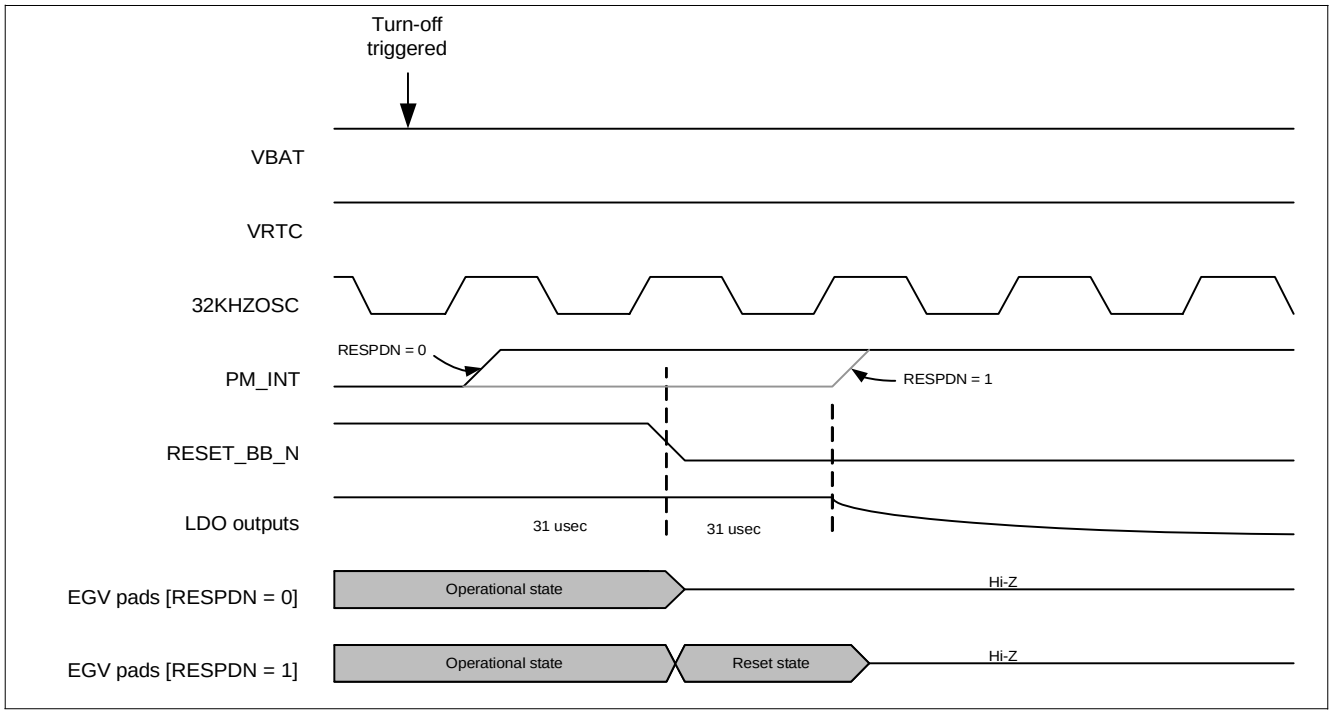


Figure 89 Power Off Timing

8.1.2.6 Undervoltage Shutdown

The normal way to shut down the system is triggered by the controller which senses the battery voltage and decides to turn off if the limit of operation is reached. If the battery is pulled out or looses contact during full operation, the capacitor in parallel to the battery which helps to avoid battery contact bouncing will be discharged in a couple of microseconds. The time for discharge depends on capacitor size, voltage level of the battery, consumers turned on etc.

In some cases the controller does not have sufficient time to program a shutdown. In this case PMU executes an undervoltage shutdown to the system because the undervoltage shutdown level is crossed. The undervoltage shutdown immediately pulls RESET_BB_N low to avoid execution of code with a supply voltage outside spec limits.

When an undervoltage or SWCT overflow condition occurs while in B-7 System ON state, RESET_BB_N is first pulled down and, after one cycle, PM_INT is set to high and LDOs are turned off (SWCT overflow is a turn-off condition if the SWCT has been started by hardware in state B-2a. However, the system will not go to power-down if SWCT overflow occurs with SWCT started by software).

8.1.2.7 Overvoltage

The overvoltage threshold can be configured via the field **PMU_CHGCTRL.OVL**. The different possible values are 5.5V, 4.57V, 4.47V. The overvoltage shutdown feature can be enabled or disabled via the bit **PMU_CHGCTRL.OVEN.**

If the overvoltage feature is enabled (default, bit OVEN=1): when the battery reaches the level defined by the bit field OVL, charging stops and an interrupt can be generated.

If the overvoltage feature is disabled (bit OVEN=0): the overvoltage interrupt is never generated and the charging is stopped when VBAT=5.5V, whatever the value of overvoltage threshold configured in the bit field OVL.

8.1.2.8 Software Reset

For support of the software reset functionality in the system, bit **PMU_GENCTRL.RES** is provided. Setting this bit to 1 performs a restart of the system by asserting the internal reset RESET_BB_N from PMU to BaseBand, in case of an unrecoverable system error. When set, bit **PMU_GENCTRL.RES** stays asserted until the MCU reset clears it. This is not a silent reset. A silent reset can be programmed on the MCU.

8.1.2.9 System Standby Mode and VCXO Control

When the system has reached System ON state (state B-7, refer to **Figure 87**), the PMU of E-GOLDvoice is under software control. The activation and deactivation of the regulators, which are not controlled directly by the power-up state machine, are controlled by the **PMU_PWRCTRL1** and **PMU_PWRCTRL2** registers.

In order to support the low-power system standby operation, a flexible clock switching option allows minimizing the power consumption during the operation phases of the E-GOLDvoice. The main task of the power management is to reduce current consumption during the standby mode, by switching the clock to 32 kHz and switching off most of the device.

Switching between system operating modes is controlled by the Standby Clock Control Unit SCCU. The SCCU provides the VCXOEN interface signal to the PMU. This signal indicates, to the PMU, the operating state of the E-GOLDvoice, and determines the operating mode of several of the regulators.

8.1.2.10 Control Input VCXOEN

VCXOEN is a control signal, which allows to switch the state of the power mangement functions of the PMU while in idle mode :

- VCXOEN = 0, standby mode
- VCXOEN = 1, paging mode.

The regulators LANA, LMEM, LRFXO, LRFTRX, LRFRX and LIO can be programmed to a configuration, which allows to control their on/off state with VCXOEN.

The regulators LD1, LANA, LMEM, LSIM and LIO can be programmed to a configuration, which allows to control their on/standby state with VCXOEN.

Since the LRFXO regulator supplies the 26MHz DCXO in the RF macro, VCXOEN indirectly controls also activation/deactivation of the 26MHz main system clock. During system standby, this clock is off and the system runs off the low power 32kHz clock.

In standby mode, the dynamic performance of LD1, LANA, LMEM, LSIM and LIO is reduced.

Table 50 shows the functions which can be controlled by VCXOEN and their required **PMU_PWRCTRL1** and **PMU_PWRCTRL2** registers programming.

Table 50 Control Function of VCXOEN

Regulator	Register setting to allow control by VCXOEN	Regulator State	
		VCXOEN = 0	**VCXOEN = 1**
LD1	**PMU_PWRCTRL1.LD1MD** = 01	standby	on
LANA	**PMU_PWRCTRL1.LANAMD** = 01 **PMU_PWRCTRL1.LANAMD** = 10	off standby	on on
LMEM	**PMU_PWRCTRL1.LMEMMD** = 01 **PMU_PWRCTRL1.LMEMMD** = 10	off standby	on on
LSIM	**PMU_PWRCTRL1.LSIMMD** = 01	standby	on
LRFXO	**PMU_PWRCTRL1.LRFXOMD** = 01	off	on
LRFTRX	**PMU_PWRCTRL1.LRFTRXMD** = 01	off	on

Table 50 Control Function of VCXOEN (cont'd)

LRFRX	**PMU_PWRCTRL1.LRFRXMD** = 01	off	on
LIO	**PMU_PWRCTRL2.LIOMD** = 01 **PMU_PWRCTRL2.LIOMD** = 10	off standby	on on

Table 51 Control Function of the Regulators

Regulator	Register setting to disable control by VCXOEN	Regulator State
LD1	**PMU_PWRCTRL1.LD1MD** = 00 **PMU_PWRCTRL1.LD1MD** = 11	off always on
LANA	**PMU_PWRCTRL1.LANAMD** = 00 **PMU_PWRCTRL1.LANAMD** = 11	off always on
LMEM	**PMU_PWRCTRL1.LMEMMD** = 00 **PMU_PWRCTRL1.LMEMMD** = 11	off always on
LSIM	**PMU_PWRCTRL1.LSIMMD** = 00 **PMU_PWRCTRL1.LSIMMD** = 10 **PMU_PWRCTRL1.LSIMMD** = 11	off off always on
LRFXO	**PMU_PWRCTRL1.LRFXOMD** = 00 **PMU_PWRCTRL1.LRFXOMD** = 10 **PMU_PWRCTRL1.LRFXOMD** = 11	off off always on
LRFTRX	**PMU_PWRCTRL1.LRFTRXMD** = 00 **PMU_PWRCTRL1.LRFTRXMD** = 10 **PMU_PWRCTRL1.LRFTRXMD** = 11	off off always on
LRFRX	**PMU_PWRCTRL1.LRFRXMD** = 00 **PMU_PWRCTRL1.LRFRXMD** = 10 **PMU_PWRCTRL1.LRFRXMD** = 11	off off always on
LIO	**PMU_PWRCTRL2.LIOMD** = 00 **PMU_PWRCTRL2.LIOMD** = 11	off always on

Table 52 Default setting of the LDOs

Regulator	Default Regulator Setting
LD1	always on, no standby function, 1.5V
LANA	off
LMEM	always on, no standby function
LSIM	off, 1.8V
LRFXO	always on, no standby function
LRFTRX	off
LRFRX	off
LIO	always on, no standby function, 1.8V
LBUF	off, 2.6V

8.1.2.11 Power-on and external Resets

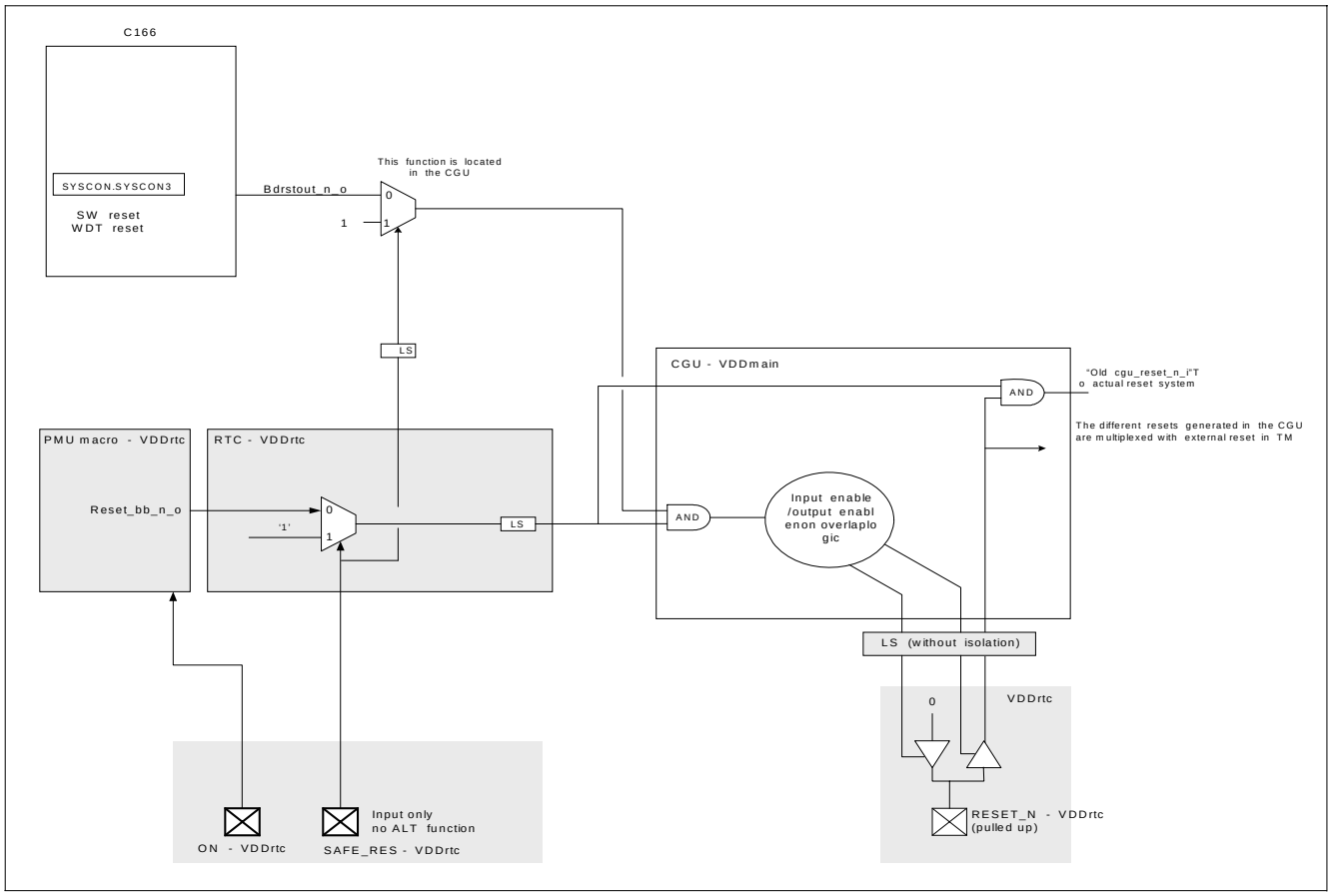


Figure 90 Power-on and External Reset Scheme

In normal functional mode, the PMU generates the power-on reset Reset_bb_n. It is sent to the Baseband, as well as onto the RESET_N pad in output mode.

RESET_N pad is bidirectional and is used :

1- in output mode, to reset external components, driven by the power-on reset from the PMU or the watchdog timer reset or the software reset.

2- in input mode, for application debug or test, to reset, from external, the Baseband independently from the PMU.

8.1.2.12 Interrupts

The EGoldVoice PMU asserts an interrupt on PM_INT line to the Baseband upon the following events :

- change of state on the CDT charger input
- charging has been interrupted by an overvoltage condition
- change of level on pin ON

Each interrupt source contributor to the PMU interrupt line PM_INT has to be enabled in the **PMU_INTCTRL** control register to be taken into account. In principle, PM_INT = IRQ1.EN1 + IRQ2.EN2 + IRQ3.EN3.
Each of these interrupt sources has:

- a status bit: **PMU_STAT**.**CHGV**, **PMU_STAT**.**OV**, **PMU_STAT**.**LON**
- an interrupt enable bit in the register **PMU_INTCTRL** : **PMU_INTCTRL**.**CDTIE**, **PMU_INTCTRL**.**OVIE**, **PMU_INTCTRL**.**LONIE**

- an interrupt clear bit in the register **PMU_INTCTRL** : **PMU_INTCTRL**.**CDTCLR** , **PMU_INTCTRL**.**OVCLR** , **PMU_INTCTRL**.**LONCLR**

While the change of state detection on CDT and ON pins is available on bits **PMU_STAT**.**CHGV** and **PMU_STAT**.**LON.** The current status of these pins is readable on bits **PMU_CHGCTRL**.**CDTS** and **PMU_CHGCTRL**.**LONS.**

When interrupt enable bit is reset (ie interrupt is disabled), the status register bits still show the source of the interrupt, but the PM_INT line is not triggered.

When the interrupt is cleared, both the status bit and interrupt contributor to PM_INT line are reset.

8.1.2.13 Clock Scheme

Upon the first battery insertion, the software is responsible to :

- de-isolate the RTC, by writing the correct value into **RTCIF** register

- apply a RTC software reset, writing **RST_CTRL_STA**.**RTC_RESET** (the reset is propagated only when the RTC isolation is disabled)

- enable the RTC oscillator (**RTC_CTRL**.**PU32K)** and the 32KHz clock (**RTC_CTRL**.**32KEN**)

- switch from the local PMU oscillator to the RTC 32KHz (**PMU_PWRCTRL2**.**CLKSEL**)

- for power savings, the software is also responsible to power down the PMU oscillator (**PMU_PWRCTRL2**.**OSCPD**), once it has switched to the RTC 32KHz.

Writing ‘1’ into **PMU_PWRCTRL2**.**CLKSEL** blocks the RTC software reset on the 2 bits **RTC_CTRL**.**PU32K and RTC_CTRL**.**32KEN.** The next RTC software reset will have, therefore, no effect on these 2 bits : the 32KHz clock cannot be switched off any longer.

8.1.3 Linear Voltage Regulator LDO (Low DropOut)

E-GOLDvoice provides efficient linear voltage regulators with very low dropout voltage and very low idle current. Several LDO's are availabe for all supply voltages needed in a cellular system. All LDO's input can be directly connected to a supply voltage up to 5.5 V that means direct battery connection is possible. Every LDO needs an output blocking capacitance of 1 µF for stability reasons.

LDO's needed for system startup are switched on automatically, the other LDO's can be switched on/off in the corresponding control register. The LDO's are splitted into four groups to allow different input voltages if available in the system.

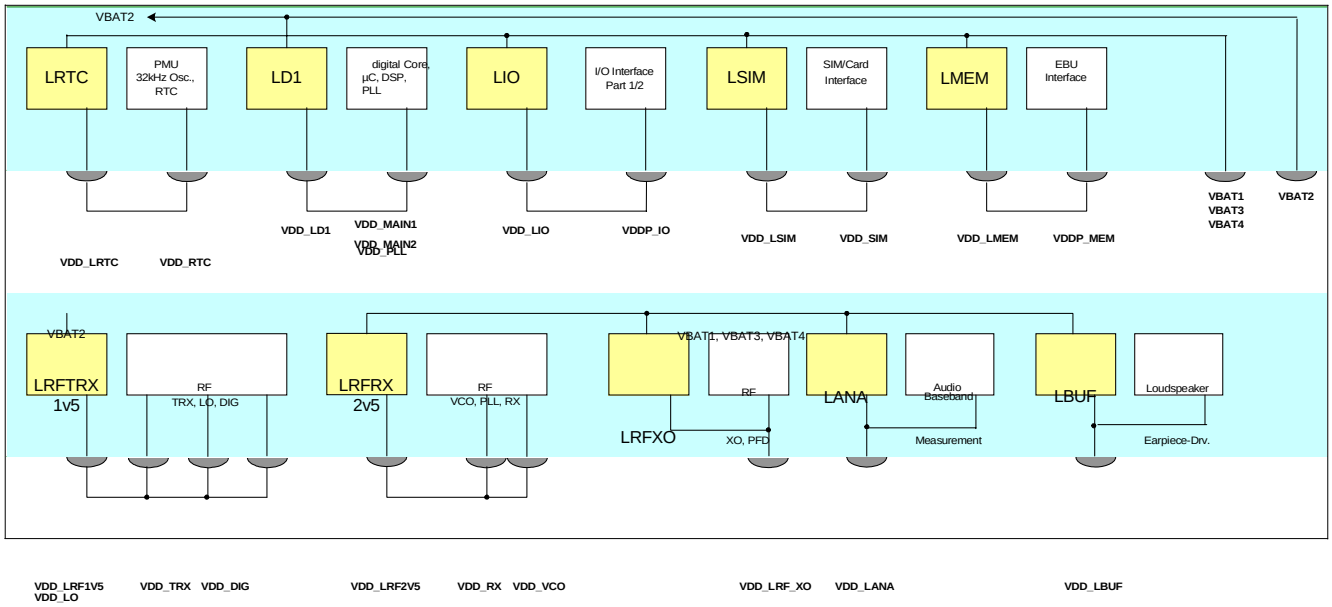


Figure 91 LDO Connections Overview

8.1.3.1 Thermal Design

If the input voltage of the LDO's is high, a huge part of the input power will be lost in the pass devices of the LDO's. It is not possible to use all LDO's with the maximum output current at the same time. The average power loss together with the thermal resistance of the package has to be concidered during system design. The die temperature shall never be higher than 125°C.

To calculate the power loss of the PMB7880 multiply the input voltage with the sum of the average currents of LRTC, LD1, LRFXO, LANA, LBUF, LRFRX, and LRFTRX. Then add the input/output voltage difference of LIO, LSIM, and LMEM multiplied with the average output current.

8.2 Battery Charger

The charger unit controls charging of NiCd, NiMH and LiON batteries. Only a few external parts are required to support half-wave, full-wave and electronic chargers. In addition the charger generates the power on reset after battery connection or charger connection. The supported battery voltage range is 3.1 V to 5.5 V for NiCd/NiMH and 3.1 V to 4.47 V for LiON batteries.

The charger supports the following features:

- switched charging, charge current/voltage adjusted in charger unit
- charger detection
- battery over-voltage detection, battery voltage monitoring
- power on reset
- IC over-voltage protection
- pre-charging (e.g. for deep discharged batteries)
- software controlled charging

8.2.1 Switched Charging

In principle the charger can be regarded as a switch between the charger unit and the batteries. In constant current charging the charge current is, for example, limited by a resistor placed before the charge switch or by a current control circuit in an electronic charger. It is also possible to use a constant voltage charger but not recommended because of the additional cost in the charging unit. The shortest charge times can be reached with constant current charger.

If a full- or a half-wave charger unit is used, the charge current is given by the voltage difference between charger voltage and battery voltage divided by the current limiting resistor (see **Figure 92**). If an electronic (switched) charger is used, the charge current is usually constant and defined by the electronics control in the charger.

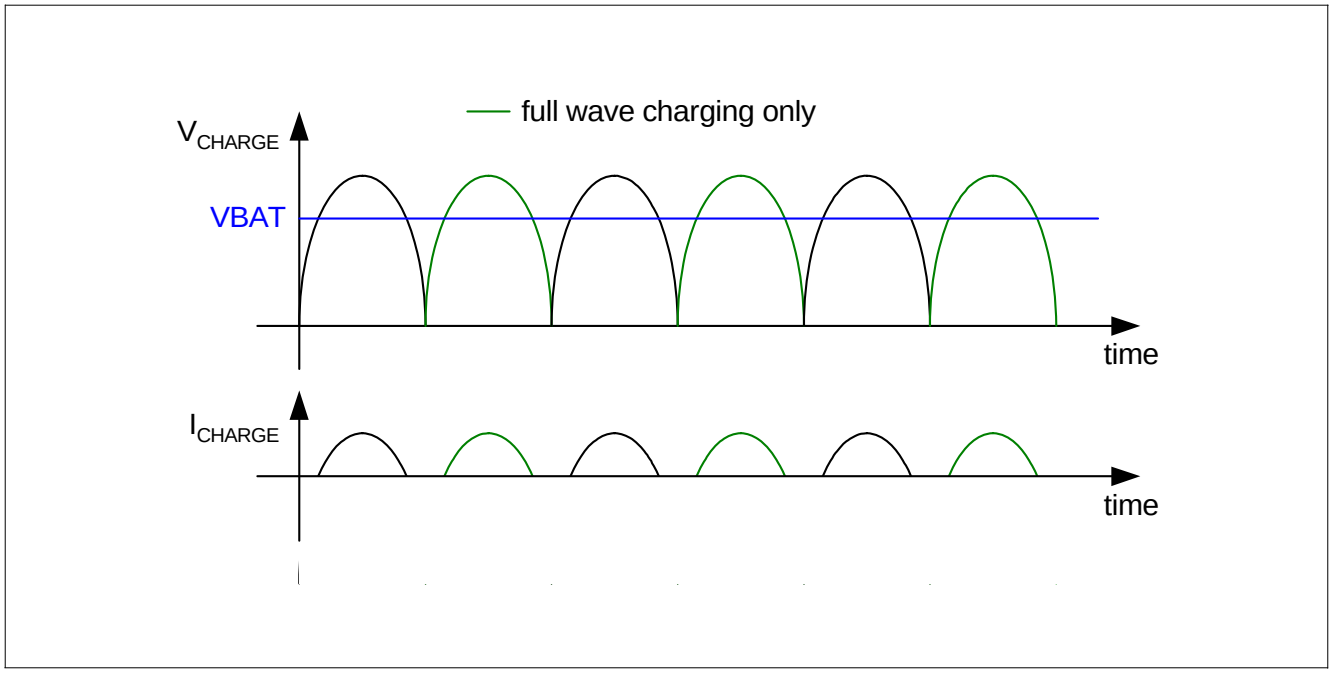


Figure 92 Charge Current in Half- and Full-wave Charging

The charger circuit can handle the normal AC supply frequency range from 50 to 60 Hz.

8.2.2 Charger Detection

A charger unit is detected by sampling current flowing into the CDT pin. If a sufficient current is detected, charge detect is indicated to the digital part of the IC, the corresponding register bit's are set and finally an interrupt is generated.

8.2.3 Battery Over-Voltage Detection and Battery Voltage Monitoring

The battery over-voltage detection is implemented for emergency switching off charging if e.g. the batteries are removed during charging or battery protection. The over-voltage level can be set by a register bit and is 5.5 V (default and start up) or 4.47 V / 4.57 V. The 4.47 V / 4.57 V level can be used for LiIon batteries for example. The battery voltage monitoring function is implemented for system start up and shut down. It delivers the input signals for the PMU state machine. The shut down feature is implemented as an emergency shut down. A controlled shut down should be done by software after measuring the battery voltage with the measurement unit.

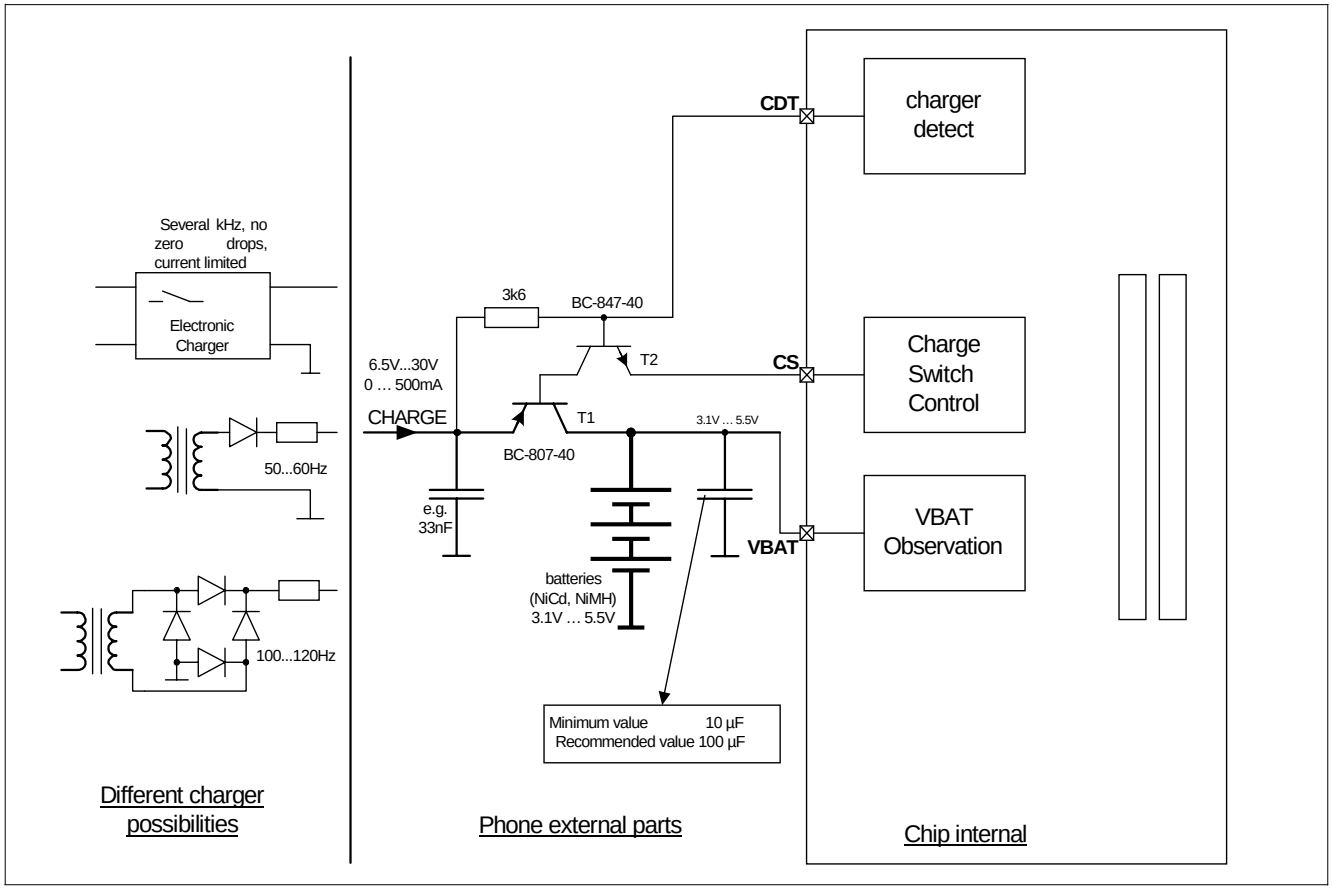


Figure 93 Block Figure of the Charger Circuit

8.2.4 Power ON Reset

The power on reset is released if the battery voltage exceeds typical 2.5 V. Because the power on detection is always running only a small current is allowed. That means the power on reset level accuracy is low. (2.25 V... 2.85 V). The power on reset starts the LRTC regulator. The system is started by the PMU state machine. Comparator Hysteresis is 40mV.

8.2.5 IC Over-Voltage Protection

The CDT / CS and VBAT pin needs to be protected against over-voltage from the charger unit.

The CDT pin is protected by use of an integrated shunt regulator together with the external resistor. This regulator will limit the voltage at the CDT pad to a value below 2.5 V.

The CS pin is automatically protected together with the CDT pin because of the emitter base diode of the external transistor T2. That means the voltage at pin CS is either defined by the internal circuit or one diode forward voltage below the voltage at pin CDT.

The VBAT pin is protected via an over-voltage detection in the charge switch circuit. That means the charge switch is automatically opened (charging stopped) if the battery voltage exceed the maximum value (4.47 V/5.5 V).

8.2.6 Pre-Charging

If the batteries are deeply discharged (that means battery voltage is between 0 V and 3.1 V) and the device is off (or software has not disabled pre-charging) the charger circuits starts pre-charging if a charger unit is connected. In pre-charging the charge switch is pulsed with 100 Hz and a duty cycle of 12.5%. That means the average charge current is reduced to avoid overheating of the charger parts and to gentle charge the deeply discharged batteries. Pre-charging is hardware controlled and continued as long as the software switches off pre-charging.

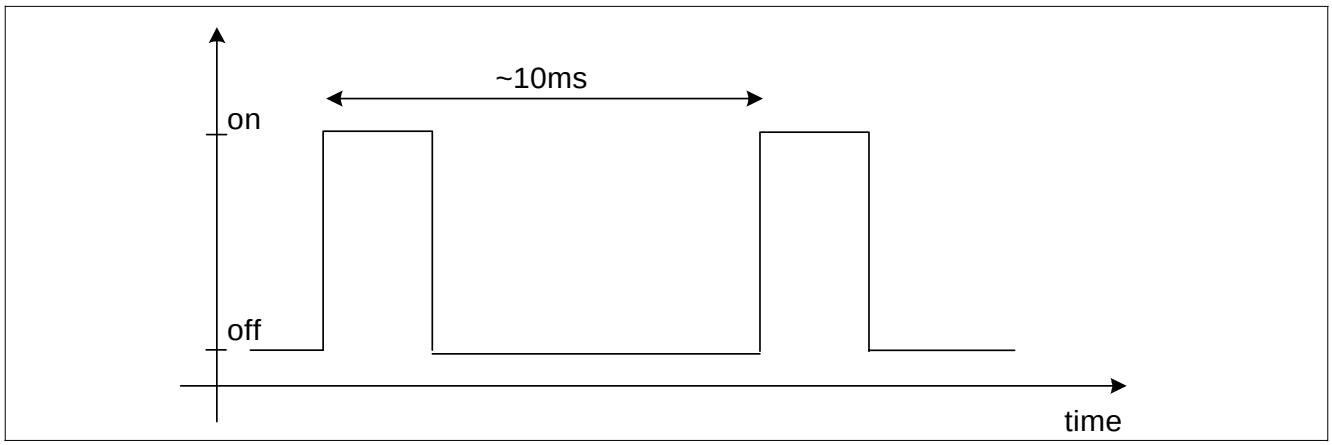


Figure 94 Charge Switch Timing in Pre-Charge Mode

8.2.7 Software Controlled Charging

If software is running, it can switch off the pre-charging function. That means the hardware will not start any charging after a charger unit was connected. Instead the software can read out the CDT bit (or enable the CDT interrupt) to get informed about a connected charger. Now the charge switch can be controlled by software according the software charge algorithm.

If the software closes the charge switch (transistor T1 is conducting) it is not closed for 100% of the time but still pulsed with a 100 Hz clock. The on time is >99% of a period. The remaining off time is used to check if the charger unit is still connected.

If the charger unit was disconnected, the hardware clears the CDT bit but do not interact with the charge switch. Software has to switch off the charge switch.

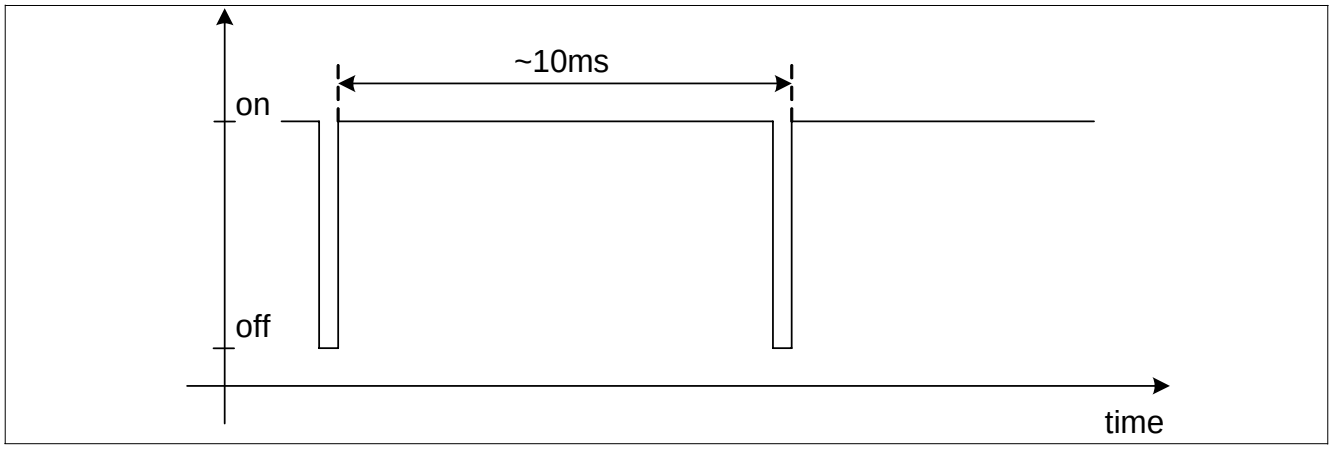


Figure 95 Charge Switch Timing in Software Charge Mode if Charging is ON

8.2.8 Thermal Design

The thermal design of the external charge circuit becomes critical if a high charge current is used. The thermal resistance of the BC807 is less than 280°C/W[1)]. Regarding an max. ambient temperature of 45°C (depends on the battery specification) for charging and a max. junction temperature 150°C for the BC807, the allowed power loss of the transistor is (150-45)/280=0.375W.

Table 53 Power Loss Examples for the BC807 in a SOT-23 Package[1)]

Charge Current	Max. CE-Drop	Power Loss
500 mA	700 mV	350 mW
400 mA	600 mV	240 mW
300 mA	500 mV	150 mW
200 mA	350 mV	70 mW

1) Data sheet Infineon BC807

That means for constant current charging (e.g. electronic charger) the max. charge current can be at 500mA (that is the max. DC collector current of the BC807). For full-wave charging the peak current can be higher (power loss limit 384mW), for half-wave charging it can be about 1 A, the maximum peak current allowed for the BC807.

8.2.9 Charge Current larger than 500 mA

If a higher charge current is required, the BC807 has to be replaced by a more powerfull PNP transistor. It should be a high gain transistor to be driven with the 20mA base current delivered from pin CS. For low gain transistors the CS drive capability can be changed to 45mA with the drawback of additional power loss.

8.3 LED Current Generation

This unit can be used to generate supply current for display LEDs or keypad backlight LEDs. The current is generated via an inductor and, therefore, the supply voltage and topology of the backlight LEDs is very flexible. In a first time period, T1, the inductor L is charged. During this time the external n-channel switch is closed. In the second time period, T2, the inductor is discharged via the parallel connected LEDs (see **Figure 96**).

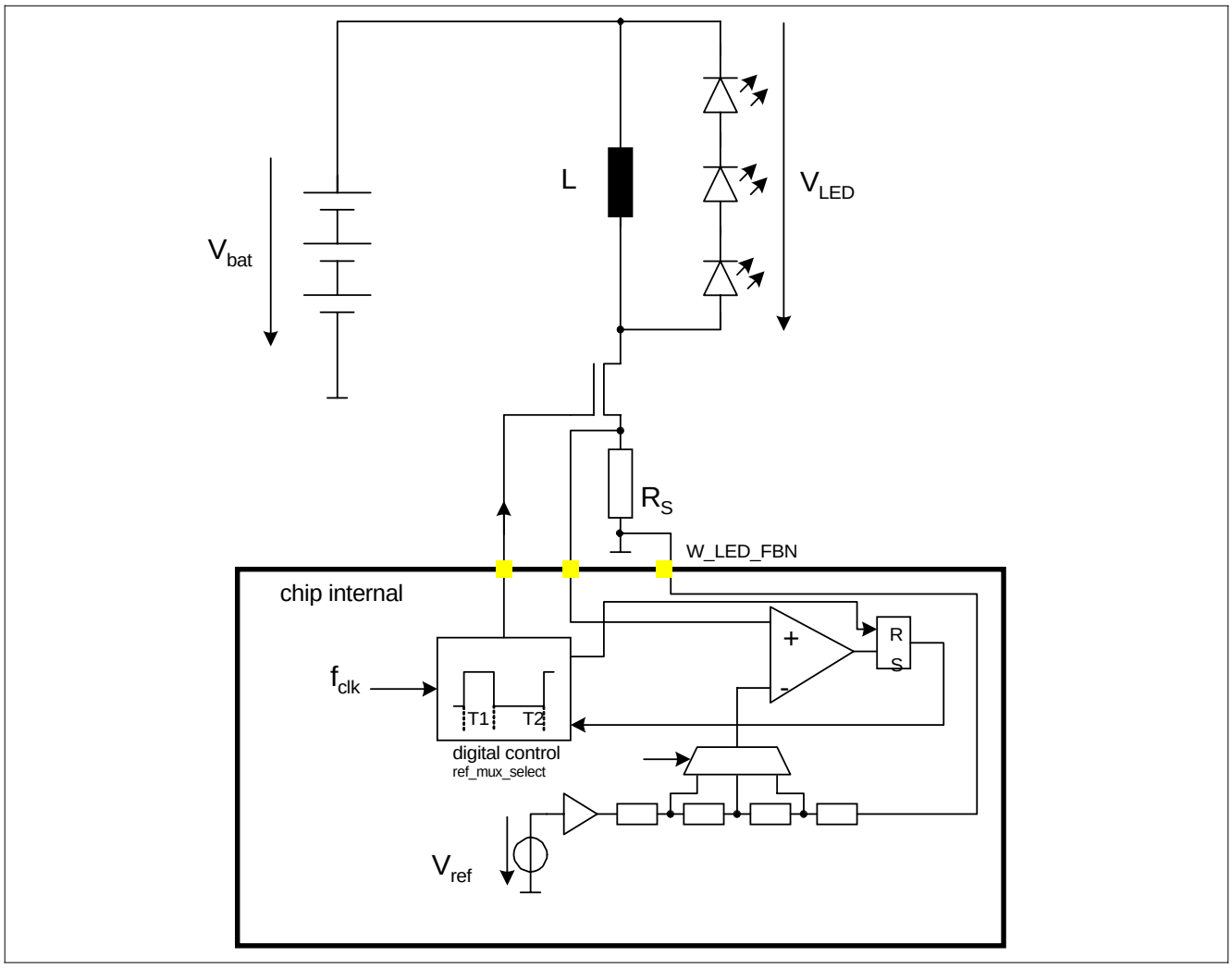


Figure 96 Block Figure of Backlight Current Generation

There are two possibilities for generating the driver waveform:

1. Pure digital (refer to **Section 8.3.1 “Digital Waveform Generation” on Page 259**)
2. Mixed analog/digital (refer to **Section 8.3.2 “Mixed Analog/Digital Waveform Generation” on Page 259**).

8.3.1 Digital Waveform Generation

The first is a pure digital generation, that means the times T1 and T2 are programmable. In this mode the resistor R_S and the W_LED_FBx pins are not needed.

8.3.2 Mixed Analog/Digital Waveform Generation

At the start of the mixed analog/digital waveform generation an analog comparator controls T1.

The sequence is started by a high level on W_LED_DRV.

The comparator stops the first phase if the voltage at the external shunt resistor R_S exceeds the selected reference voltage at the comparator.

The time T1 is measured and can be read back from software. The length of phase T2 is programmable. During T2 the inductor is discharged via the LEDs. Using T2 to select if the circuit is working in the discontinuous conduction mode (DCM) or in the continuous conduction mode (CCM). This choice effects the mean current through the LED. Changing T2 allows dimming the light intensity.

Table 54 Peak Current through the LED's for different reference voltage settings and shunt resistances

R_S	300 mV	200 mV	150 mV
4R7		42 mA	30 mA
2R4		80 mA	60 mA
1R8		110 mA	80 mA

8.3.2.1 LED Backlight Registers

Table 55 PMU Registers

Reg. Name	Description
LED_k1	counter value k1 for time T1
LED_k1max	ma x. limit for k1
LED_k2	counter value k2 for time T2
LED_k2min	min. value for k2
LED_k2max	max. limit for k2
LED_cip	coefficient for backlight control
LED_cv	coefficient for backlight control
LED_ciavt	coefficient for backlight control
LED_cpi	coefficient for backlight control
LED_CTRL[1)]	control register for backlight generation

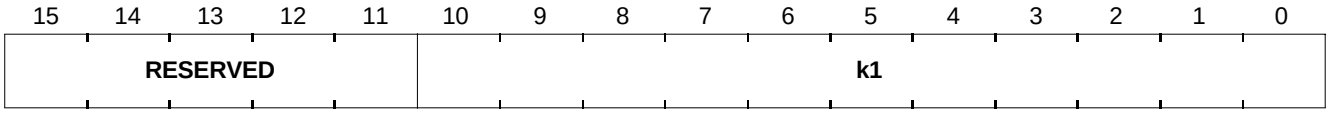
1) SW user need to consider that those bits are written with a 26MHz cycle (when code is executed), **but sampled with a 32/50KHz period**. Hence setup (duration of signal) shall be long enough so that it can be captured by 32KHz clock (period 31.2us).

8.3.2.1.1 LED Control Registers

Note: K1 and K2 parameters have to be initialised by SW for Digital control mode of PMU leds

LED_k1

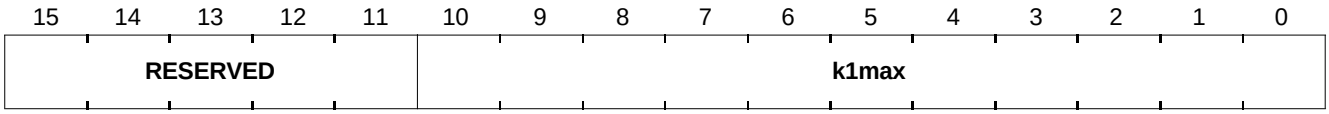
LED Backlight Coefficient Register **Reset Value: 0000_H**



Field	Bits	Type	Description
k1	10:0	rw	**Value of counter k1 for time T1** Meaning depends on LED Backlight mode: • Free Running Mode This counter value is used to determine the inductor charge time T1 • Analog Feedback Mode & Digital Control Mode The software can read out the counter value for the charge time T1 stopped by the analog comparator
RESERVED	15:11	r	Reserved; these bits are not used, always read 0

LED_k1max

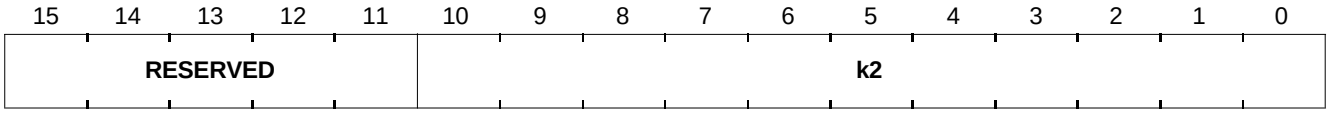
LED Backlight Coefficient Register **Reset Value: 0000_H**



Field	Bits	Type	Description
k1max	10:0	rw	**Maximum value of counter k1 for time T1** Only used in Digital Control Mode. Determines the maximum value for k1 the digital control algorithm is working. If k1 is higher than this value k2 is not calculated.
RESERVED	15:11	r	Reserved; these bits are not used

LED_k2

LED Backlight Coefficient Register **Reset Value: 0000_H**



Field	Bits	Type	Description
k2	10:0	rw	**Value of counter k2 for time T2** Meaning depends on LED Backlight mode: • Free Running Mode & Analog Feedback Mode This counter value is used to determine the inductor discharge time T2. • Digital Control Mode The software can read out the counter value for the inductor discharge time T2 calculated by the control algorithm
RESERVED	15:11	r	Reserved; these bits are not used, always read 0

LED_k2min

LED Backlight Coefficient Register **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					k2min										

Field	Bits	Type	Description
k2min	10:0	rw	**Minimum value of counter k2 for time T2** Only used in Digital Control Mode. The k2 value calculated by the control algorithm is limited to that value. That means k2 is always equal or greater this value
RESERVED	15:11	r	Reserved; these bits are not used

LED_k2max

LED Backlight Coefficient Register **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					k2max										

Field	Bits	Type	Description
k2max	10:0	rw	**Maximum value of counter k2 for time T2** Only used in Digital Control Mode. Determines the maximum value for the new calculated k2 value if the digital control algorithm is working. If the calculated k2 is larger than this value, it is set to k2max.
RESERVED	15:11	r	Reserved; these bits are not used

LED_cip

LED Backlight Coefficient Register **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cip															

Field	Bits	Type	Description
cip	15:0	rw	**Coefficient cip for the control algorithm**

LED_cv

LED Backlight Coefficient Register **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cv															

Field	Bits	Type	Description
cv	15:0	rw	**Coefficient cv for the control algorithm**

LED_clavt

LED Backlight Coefficient Register **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
clavt															

Field	Bits	Type	Description
clavt	15:0	rw	**Coefficient clavt for the control algorithm**

LED_cpi

LED Backlight Coefficient Register **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								cpi							

Field	Bits	Type	Description
cpi	7:0	rw	**Coefficient cpi for the control algorithm**
RESERVED	15:8	r	Reserved; these bits are not used

LED_CTRL

LED Backlight Control Register **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ref_sel	TX_msk	RX_msk	ref_mux_sel		cp_en	mode	

Field	Bits	Type	Description
mode	1:0	rw	**Select mode of backlight generation** 00 LED backlight not running 01 Free running mode 10 Analog feedback mode 11 Digital control mode
cp_en	2	rw	**Enable analog comparator and reference** 0 Comparator not running (no current consumption) 1 Comparator running
ref_mux_sel	4:3	rw	**select reference for comparator** 00 300 mV 01 200 mV 10 150 mV 11 Not used (300 mV)
RX_msk	5	rw	**stops backlight during RX slot** 0 Backlight running continuously 1 Backlight switched off during RF-RX slot
TX_msk	6	rw	**stops backlight during TX slot** 0 Backlight running continuously 1 Backlight switched off during RF-TX slot

Field	Bits	Type	Description
ref_sel	7	rw	**Signal to analog part, currently not used**
RESERVED	15:8	r	Reserved; these bits are not used

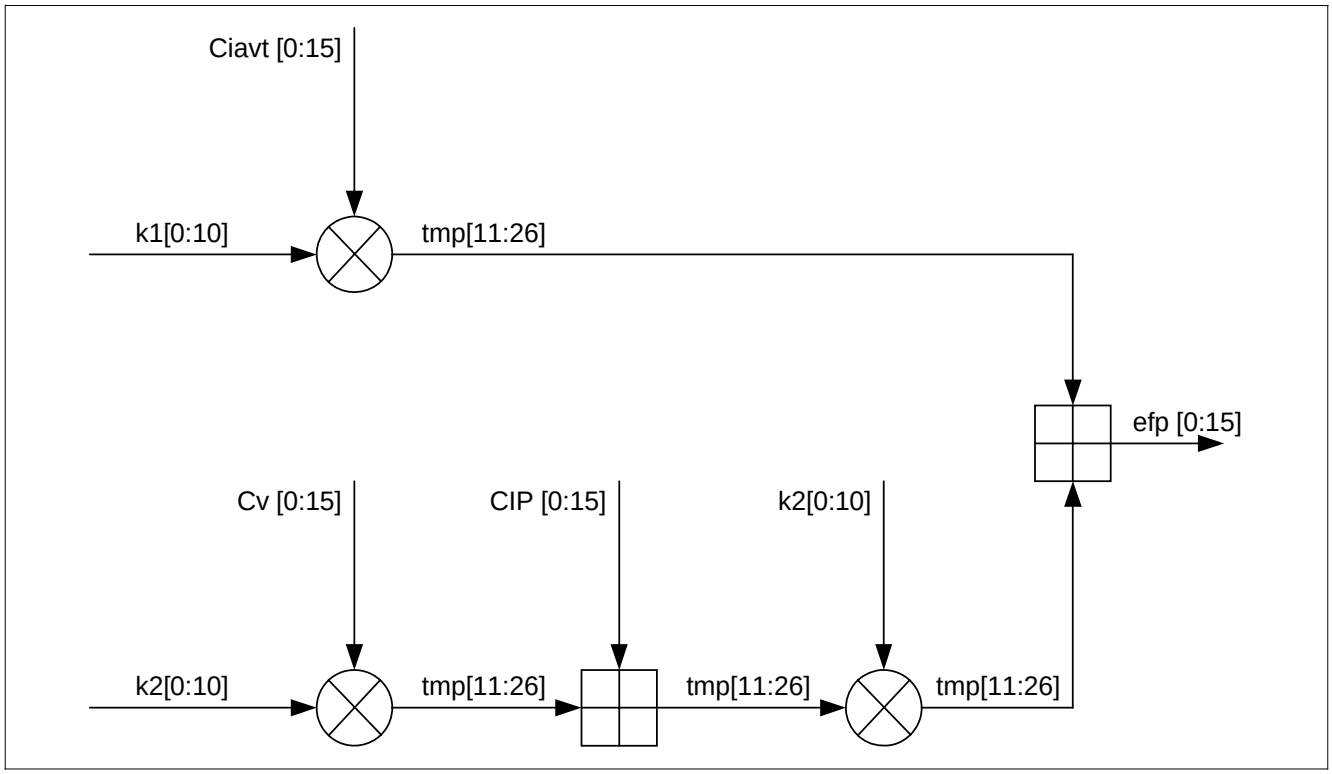


Figure 97 Arithmetic Flow for Observer and Control Error

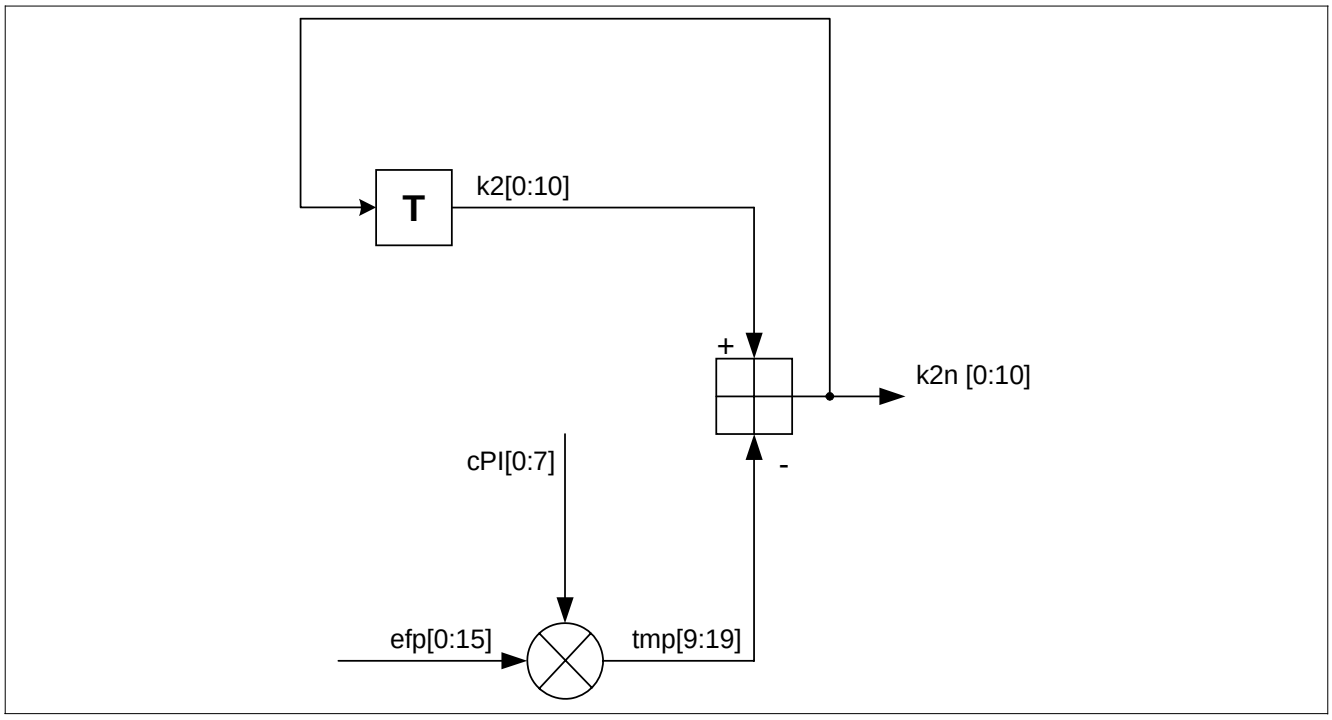


Figure 98 Arithmetic Flow for the Control Part

9 PD-Bus

9.1 I2C Bus Interface

System Integration

- Supply domain: VDD_LD1
- Chip internal interfaces:
  - Clock domain: Refer to **Section 7.2.1.3 Sub-System Clocks and Enables (on Page 67)** and see **Figure 18 Clock Enable (on Page 68)**.
  - Bus domain: PD-Bus
- Interrupt sources:
- Monitor Pins: refer to **Section 9.7.10 Internal Signal Monitoring (on Page 435)**.

9.1.1 Introduction

IIC supports a protocol that allows devices to communicate directly with each other via two wires. One line is responsible for clock transfer and synchronization (SCL), the other is responsible for the data transfer (SDA). The on-chip IIC Bus module connects the platform buses to other external controllers and/or peripherals via the two-line serial IIC interface. The IIC Bus module provides communication at data rates of up to 400kbit/s and features 7-bit addressing as well as 10-bit addressing. This module is fully compatible to the IIC bus protocol. The module can operate in three different modes:

- **Master Mode**, the IIC controls the bus transactions and provides the clock signal.
- **Slave Mode**, an external master controls the bus transactions and provides the clock signal.
- **Multimaster Mode**, several masters can be connected to the bus, that is, the IIC can be master or slave.

The on-chip IIC bus module allows efficient communication via the common IIC bus. The module unloads the CPU of low level tasks such as:

- (De)Serialization of bus data.
- Generation of start and stop conditions.
- Monitoring the bus lines in slave mode.
- Evaluation of the device address in slave mode.
- Bus access arbitration in multimaster mode.

Features

- Extended buffer allows up to 4 send/receive data bytes to be stored.
- Selectable baud rate generation.
- Support of standard 100kBaud and extended 400kBaud data rates.
- Operation in 7-bit addressing mode or 10-bit addressing mode.
- Flexible control via interrupt service routines or by polling.
- Dynamic access to up to 4 physical IIC buses.

Applications

- EEPROMs
- 7-Segment Displays
- Keyboard Controllers
- On-Screen Display
- Audio Processors.

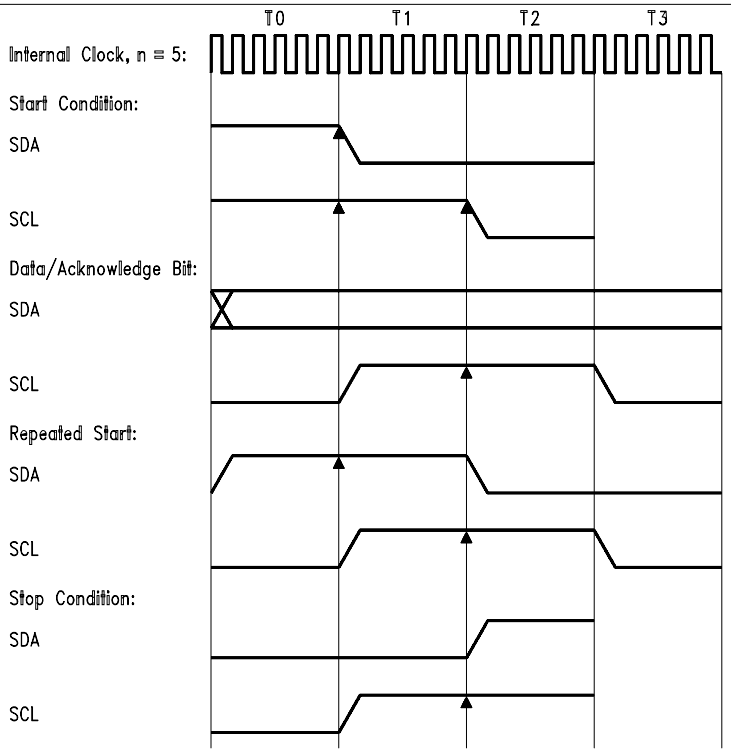
9.1.2 Operational Overview

Data is transferred by the 2-line IIC bus (SDA, SCL) using a protocol that ensures reliable and efficient transfers. This protocol clearly distinguishes regular data transfers from defined control signals which control the data transfers.

The following bus conditions are defined:

- Bus Idle:SDA and SCL remain high. The IIC bus is currently not used.
- Data Valid:SDA stable during the high phase of SCL. SDA then represents the transferred bit. There is one clock pulse for each transferred bit of data.
  During data transfers, the SDA may only change while SCL is low. If SCL is high, the transfer stops (refer to Stop Transfer below).
- Start Transfer:A falling edge on SDA (↘) while SCL is high indicates a start condition.
  This start condition initiates a data transfer over the IIC bus.
- Stop Transfer:A rising edge on SDA (↗) while SCL is high indicates a stop condition. This stop condition terminates a data transfer. Between a start condition and a stop condition, an arbitrary number of bytes may be transferred.

Figure 99 gives examples for these bus conditions.



↑ The high level of the respective signal is verified.
If the signal is low, the previous state (Ti) is repeated.

Figure 99 Bus Conditions

9.1.2.1 Physical IIC-Bus Interface

Communication via the IIC Bus uses two bidirectional lines, the serial data line SDA and the serial clock line SCL (see **Figure 100**). These two interface lines are connected to two I/O ports.

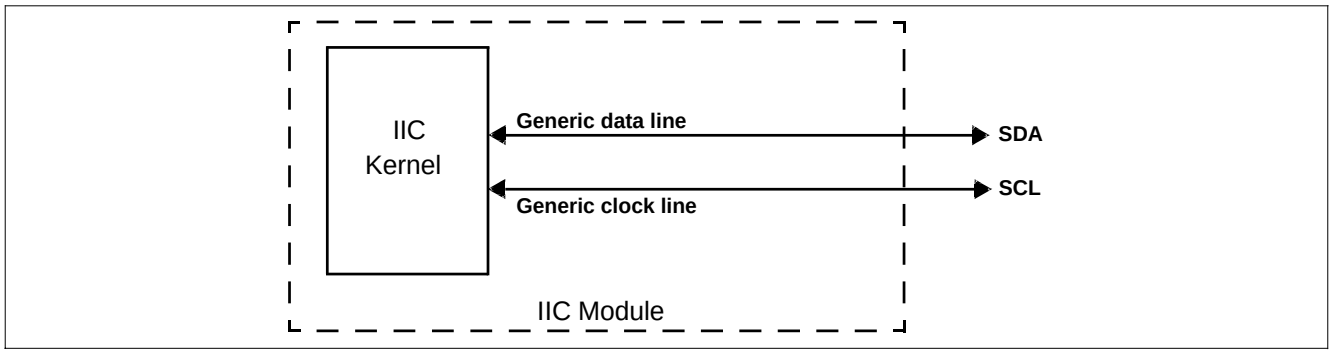


Figure 100 IIC Bus Line Connections

Register **IIC_CFG (on Page 279)** selects the bus baud rate as well the activation of SDA and SCL lines. So an external IIC channel can be established (baud rate and physical lines) with one single register access.

*Note: Baud rate and physical channels must never be changed (via **IIC_CFG**) during a transfer.*

9.1.2.2 Output Pin Configuration

The pin drivers that are assigned to the IIC channel(s) provide open drain outputs (that is, no upper transistor). This ensures that the IIC module does not put any load on the IIC bus lines while the C166S is not powered. The IIC bus lines, therefore, require pull-up resistors. Values for these resistors depend upon the capacitive load on the I2C lines. Rules for the selection of the values are given in the I2C bus specification.

All pins of the CB-Core that are to be used for IIC bus communication must be switched to output and their alternate function must be enabled (by setting the respective port output latch to 1) before any communication can be established.

If not driven by the IIC module (that is, the corresponding enable bit in register **IIC_CFG** is 0), they then switch off their drivers (that is, driving it to an open drain output). Due to the external pull-up devices, the respective bus levels are then 1, which is idle.

The IIC module features digital input filters to improve the rejection of noise from the external bus lines.

9.1.3 Functional Overview

For information about the registers in this section refer to **Section 9.1.4 Registers (on Page 268)**.

9.1.3.1 Operation in Master Mode

If the on-chip IIC module controls the IIC bus (that is, as a bus master), the master mode must be selected via bit field **IIC_CON.MOD**. The physical channel is configured by a control word written to register **IIC_CFG**, which activates the interface pins and the baud rate used. The address of the remote slave that is to be accessed is written to either **RTB_LO.RTB(0,1)** or **RTB_HI.RTB(2,3)**. The bus is claimed by setting bit **IIC_CON.BUM**. This generates a start condition on the bus and automatically starts the transmission of the address in **RTB_LO.RTB0**. Bit **IIC_CON.TRX** defines the transfer direction (TRX = 1, that is, transmit, for the slave address). A repeated start condition is generated by setting bit **IIC_CON.RSC**, which automatically starts the transmission of the address previously written to **RTB_LO.RTB0**. This may be used to change the transfer direction. **IIC_CON.RSC** is cleared automatically after the repeated start condition has been generated.

The bus is released by clearing bit **IIC_CON.BUM**. This generates a stop condition on the bus.

9.1.3.2 Operation in Multimaster Mode

If multimaster mode is selected via bit field **IIC_CON.MOD** the on-chip IIC module can operate concurrently as a bus master or as a slave. The descriptions of these modes apply accordingly.

Multimaster mode implies that several masters are connected to the same bus. As more than one master may try to claim the bus at a given time an arbitration is done on the SDA line. When a master device detects a mismatch between the data bit to be sent and the actual level on the SDA (bus) line it looses the arbitration and automatically switches to slave mode (leaving the other device as the remaining master). This loss of arbitration is indicated by bit **IIC_ST.AL**, which must be checked by the driver software when operating in multimaster mode. Lost arbitration is also indicated when the software tries to claim the bus (by setting bit **IIC_CON.BUM**) while the IIC bus is active (indicated by bit **IIC_ST.BB** = 1). Bit **IIC_ST.AL** must be cleared via software.

9.1.3.3 Operation in Slave Mode

If the on-chip IIC module shall be controlled via the IIC bus by a remote master (that is, be a bus slave) slave mode must be selected via bit field **IIC_CON.MOD**. The physical channel is configured by a control word written to register **IIC_ST**, defining the active interface pins and the baud rate used. It is recommended to have only one SDA and SCL line active at a time when operating in slave mode. The address for the slave module that can be selected is written to register **IIC_ADR**.

The IIC module is selected by another master when it receives (after a start condition) either its own device address (stored in **IIC_ADR**) or the general call address (00_H). In this case an interrupt is generated and bit **IIC_ST.SLA** is set indicating the valid selection. The desired transfer mode is then selected via bit **IIC_CON.TRX** (**TRX** = 0 for reception, **TRX** = 1 for transmission).

For a transmission the respective data byte is placed into either buffer **RTB_LO.RTB(0,1)** or **RTB_HI.RTB(2,3)** (which automatically sets bit **IIC_CON.TRX**) and the acknowledge behavior is selected via bit **IIC_CON.ACKDIS**. **For a reception** the respective data byte is fetched from either buffer **RTB_LO.RTB(0,1)** or **RTB_HI.RTB(2,3)** after **IIC_ST.IRQD** has been activated.

In both cases the data transfer itself is enabled by clearing bits **IIC_ST.IRQD**, **IIC_ST.IRQP,** and **IIC_ST.IRQE** which releases the SCL line.

When a stop condition is detected, bit **IIC_CON.SLA** is cleared.

The IIC Bus Configuration Register **IIC_CFG** selects the bus baud rate and activation of SDA and SCL lines. So an external IIC channel can be established (baud rate and physical lines) with one single register access.

*Note: Refer to **Section 9.1.2.1 Physical IIC-Bus Interface (on Page 267)**.*

9.1.4 Registers

For information about the IIC PD-Bus Register Mapping refer to **Section 10.1 PD-Bus Register Addresses (on Page 481)**.

All available module registers are summarized in **Table 56**.

Table 56 I2C Register List

Name	Clock	Access Condition	Description
IIC_ID	cfg_clk[1)]	not bit addressable	I2C Identification Register
IIC_PISEL	cfg_clk[1)]	bit addressable	Input port selection register.
IIC_CON	hw_clk[1)]	bit addressable	System Control Register
IIC_ST	hw_clk[1)]	bit addressable	System Status Register
IIC_ADR	hw_clk[1)]	bit addressable	Bus Address Register
IIC_CFG	cfg_clk[1)]	bit addressable	Bus Configuration Register

Table 56 I2C Register List (cont'd)

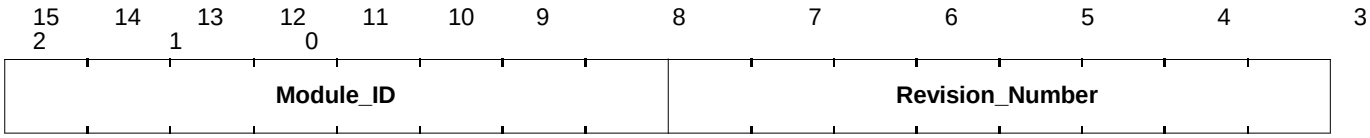
Name	Clock	Access Condition	Description
RTB_HI	hw_clk[1)]	none	High Receive/Transmit Buffer
RTB_LO	hw_clk[1)]	none	Low Receive/Transmit Buffer

1) Refer to the Clock Domain in the **System Integration (on Page 265)**.

9.1.4.1 I2C Identification Register

IIC_ID

IIC Identification Register **Reset value: 4604_H**



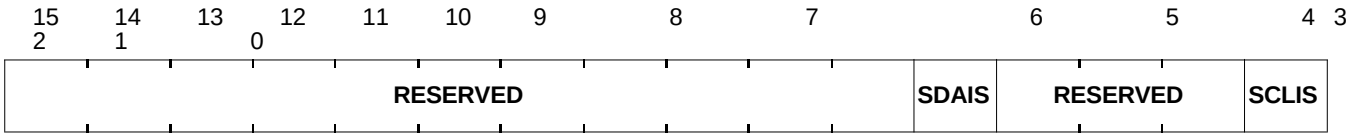
Field	Bits	Type	Description
Revision_Number	0:7	r	**I2C Revision Number** These hard-wired bits are used for module revision numbering.
Module_ID	8:15	r	**I2C Identification Number** These hard-wired bits are used for module identification numbering.

9.1.4.2 IIC Port Input Selection Register

This register does not need to be programmed. Just leave it at its reset value.

IIC_PISEL

Port Input Select Register **Reset values: 0001_H**



Field	Bits	Type	Description
SCLIS	0	rw	**Select Input for Clock Signal** 0 Signal on pin SCL 1 Not used
SDAIS	4	rw	**Select Input for Data Signal** 0 Signal on pin SDA 1 Not used
RESERVED	15:5, 3:1	r	Reserved, these bits must be left at their reset values.

9.1.4.3 System Control Registers

The operating mode of the IIC is controlled by the system control register **IIC_CON** and system status register **IIC_ST**. These registers contain control bits for mode and error check selection, and status flags for error identification.

Depending on bits **WMEN** and **RMEN**, either the write mirror (WM) or receive mirror (RM) is enabled.

For IIC_ST.RMEN = 1

IIC_CON

System Control Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RM							TRX	INT	ACK DIS	BUM	MOD		RSC	M10	

Field	Bits	Type	Description
M10	0	rw	**Address Mode** 0 7-bit addressing using **IIC_ADR.ICA(7..1)** 1 10-bit addressing using **IIC_ADR.ICA(9..0)**.
RSC	1	rwh	**Repeated Start Condition** 0 No operation. 1 Generate a repeated start condition in (multi) master mode. **RSC** can not be set in slave mode. *Note: **RSC** is cleared automatically after the repeated start condition has been sent.*
MOD	2:3	rw	**Basic Operating Mode** 00 IIC module is disabled and initialized (Init-Mode). Transmissions under execution was aborted. 01 Slave mode. 10 Master mode. 11 Multi-Master mode.
BUM	4	rwh	**Busy Master** 0 Clearing bit **BUM** (↘) generates a stop condition immediately. 1 Setting bit **BUM** (↗) generates a start condition in (multi)master mode. *Note: Setting bit **BUM** (↗) while BB = 1 generates an arbitration lost situation. In this case **BUM** is cleared and bit **AL** is set. BUM can not be set in slave mode.*
ACKDIS	5	rwh	**Acknowledge Pulse Disable** 0 An acknowledge pulse is generated for each received frame. 1 No acknowledge pulse is generated. *Note: **ACKDIS** is automatically cleared by a stop condition.*
INT	6	rw	**Interrupt Delete Select** 0 Interrupt flag **IRQD** is deleted by a read/write to **RTB_LO.RTB(0,1) or RTB_HI.RTB(2,3)**. 1 Interrupt flag **IRQD** is not deleted by a read/ write to **RTB_LO.RTB(0,1) or RTB_HI.RTB(2,3)**.

Field	Bits	Type	Description
TRX	7	rwh	**Transmit Select** 0 No data is transmitted to the IIC bus. 1 Data is transmitted to the IIC bus. *Note: **TRX** is set automatically when writing to the transmit buffer. It is not allowed to delete this bit in the same buscycle. It is automatically cleared after last byte as slave transmitter.*
IGE	8	rw	**Ignore IRQE** Ignore **IRQE** (End of transmission) interrupt. 0 The IIC is stopped at **IRQE** interrupt. 1 The IIC ignores the **IRQE** interrupt. *Note: If **IIC_ST.RMEN** is set, **RM** is mirrored here.*
RM	15:8	rh	**Read Mirror** If **IIC_ST.RMEN** is set, **RTB_LO.RTB0** may be read here. Writing to **RM** has no effect in this mode.

For IIC_ST.RMEN = 0

IIC_CON

System Control Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WMEN	RESERVED			CI		STP	IGE	TRX	INT	ACKDIS	BUM	MOD		RSC	M10

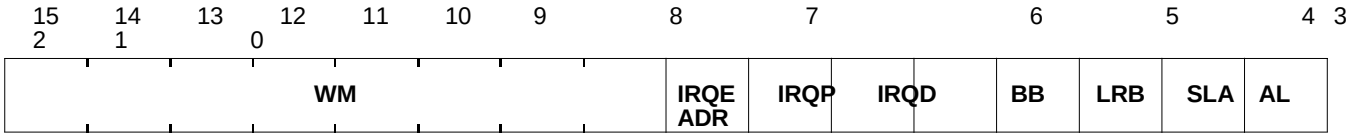
Field	Bits	Type	Description
M10	0	rw	**Address Mode** 0 7-bit addressing using **IIC_ADR.ICA(7..1)** 1 10-bit addressing using **IIC_ADR.ICA(9..0)**.
RSC	1	rwh	**Repeated Start Condition** 0 No operation. 1 Generate a repeated start condition in (multi) master mode. **RSC** can not be set in slave mode. *Note: **RSC** is cleared automatically after the repeated start condition has been sent.*
MOD	2:3	rw	**Basic Operating Mode** 00 IIC module is disabled and initialized (Init-Mode). Transmissions under execution was aborted. 01 Slave mode. 10 Master mode. 11 Multi-Master mode.
BUM	4	rwh	**Busy Master** 0 Clearing bit **BUM** (↘) generates a stop condition immediately. 1 Setting bit **BUM** (↗) generates a start condition in (multi)master mode. *Note: Setting bit **BUM** (↗) while BB = 1 generates an arbitration lost situation. In this case **BUM** is cleared and bit **AL** is set. BUM can not be set in slave mode.*
ACKDIS	5	rwh	**Acknowledge Pulse Disable** 0 An acknowledge pulse is generated for each received frame. 1 No acknowledge pulse is generated. *Note: **ACKDIS** is automatically cleared by a stop condition.*
INT	6	rw	**Interrupt Delete Select** 0 Interrupt flag **IRQD** is deleted by a read/write to **RTB_LO.RTB(0,1) or RTB_HI.RTB(2,3)**. 1 Interrupt flag **IRQD** is not deleted by a read/ write to **RTB_LO.RTB(0,1) or RTB_HI.RTB(2,3)**.
TRX	7	rwh	**Transmit Select** 0 No data is transmitted to the IIC bus. 1 Data is transmitted to the IIC bus. *Note: **TRX** is set automatically when writing to the transmit buffer. It is not allowed to delete this bit in the same buscycle. It is automatically cleared after last byte as slave transmitter.*

Field	Bits	Type	Description
IGE	8	rw	**Ignore IRQE** Ignore **IRQE** (End of transmission) interrupt. 0 The IIC is stopped at **IRQE** interrupt. 1 The IIC ignores the **IRQE** interrupt. *Note: If **IIC_ST.RMEN** is set, **RM** is mirrored here.*
STP	9	rwh	**Stop Master** 0 Clearing bit **STP** generates no stop condition. 1 Setting bit **STP** generates a stop condition after next transmission. **BUM** is set to zero. **ACKDIS** is set to one. *Note: **STP** is automatically cleared by a stop condition.* *If **IIC_ST.RMEN** is set, **RM** is mirrored here.*
CI	11:10	rw	**Length of the Transmit Buffer** 00 1 Byte 01 2 Bytes 10 3 Bytes 11 4 Bytes *Note: If **IIC_ST.RMEN** is set, **RM** is mirrored here.*
WMEN	15	rwh	**Write Mirror Enable** 0 Write mirror is not active 1 Write mirror is active *If **IIC_ST.RMEN** is set, **WMEN** cannot be set and remains 0.* *If **WMEN** and **IIC_ST.RMEN** are simultaneously set to 1, both remain what they are, only one of them can be set to 1.*
RESERVED	14:12	r	If **IIC_ST.RMEN** is cleared, then this bitfield is reserved, these bits must be left at their reset values.

For IIC_CON.WMEN = 1

IIC_ST

System Status Register **Reset value: 0000_H**



Field	Bits	Type	Description
ADR	0	rh	**Address** Bit **ADR** is set after a start condition in slave mode until the address has been received (1 byte in 7-bit address mode, 2 bytes in 10-bit address mode).
AL	1	rwh	**Arbitration Lost** Bit **AL** is set when the IIC module has tried to become master on the bus but has lost the arbitration. Operation is continued until the 9th clock pulse. If multimaster mode is selected the IIC module temporarily switches to slave mode after a lost arbitration. Bit **IRQP** is set along with bit **AL**. **AL** must be cleared via software.
SLA	2	rh	**Slave** 0 The IIC module is not selected as a slave, or the module is in master mode. 1 The IIC module has been selected as a slave (device address received).
LRB	3	rh	**Last Received Bit** Bit **LRB** represents the last bit (for example, the acknowledge bit) of the last transferred frame. It is automatically set to zero by a write or read access to buffers **RTB_LO.RTB(0,1)** and **RTB_HI.RTB(2,3)**. *Note: If LRB is high (no acknowledge) in slave mode, **TRX** bit is set automatically.*
BB	4	rh	**Bus Busy** 0 The IIC bus is idle, that is, a stop condition has occurred. 1 The IIC bus is active, that is, a start condition has occurred. *Note: Bit **BB** is always 0 while the IIC module is disabled.*
IRQD	5	rwh	**IIC Interrupt Request Bit for Data Transfer Events** [1)] 0 No interrupt request pending. 1 A data transfer event interrupt request is pending. **IRQD** is set after the acknowledge bit of the last byte has been received or transmitted, and is cleared automatically upon a complete read or write access to the buffers **RTB_LO.RTB(0,1)** and **RTB_HI.RTB(2,3**. New data transfers will start immediately after clearing **IRQD**. Do not access any register until next interrupt. If a multi byte write could not be finished in slave mode because of missing acknowledge, then the data interrupt is followed by an end of transmission interrupt. The number of bytes sent can be read from **CO**. The data interrupt must have higher priority than **IRQE**.

Field	Bits	Type	Description
IRQP	6	rwh	**IIC Interrupt Request Bit for Protocol Events** [1)] 0 No interrupt request pending. 1 A protocol event interrupt request is pending. **IRQP** is set when bit **SLA** or bit **AL** is set (↓), and must be cleared via software. If the IIC has been selected by an other master, the software must look up the required transmission direction by reading the received address and direction bit, stored in **RTB_LO.RTB0**. The **TRX** bit must be set by software correspondingly.
IRQE	7	rwh	**IIC Interrupt Request Bit for Data Transmission End** [1)] 0 No interrupt request pending. 1 A receive end event interrupt request is pending (a stop is detected). **IRQE** is automatically cleared upon a start condition. **IRQE** is not activated in init-mode. IRQE must always be deleted to continue transmission. *Note: In slave mode **IRQE** is set after the transmission is finished. This can also be after a stop or RSC condition. In this case the slave is not selected any more. This bit is also set, if a transmission is stopped by a missing acknowledge. In this case the bit must be cleared by software.*
WM	15:8	wh	**Write Mirror** If **IIC_CON.WMEN** is set, **RTB_LO.RTB0** may be written here. Reading **WM** will result in zero.

For IIC_CON.WMEN = 0

IIC_ST

System Status Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RMEN	RESERVED				CO			IRQE	IRQP	IRQD	BB	LRB	SLA	AL	ADR

Field	Bits	Type	Description
ADR	0	rh	**Address** Bit **ADR** is set after a start condition in slave mode until the address has been received (1 byte in 7-bit address mode, 2 bytes in 10-bit address mode).
AL	1	rwh	**Arbitration Lost** Bit **AL** is set when the IIC module has tried to become master on the bus but has lost the arbitration. Operation is continued until the 9th clock pulse. If multimaster mode is selected the IIC module temporarily switches to slave mode after a lost arbitration. Bit **IRQP** is set along with bit **AL**. **AL** must be cleared via software.
SLA	2	rh	**Slave** 0 The IIC module is not selected as a slave, or the module is in master mode. 1 The IIC module has been selected as a slave (device address received).
LRB	3	rh	**Last Received Bit** Bit **LRB** represents the last bit (for example, the acknowledge bit) of the last transferred frame. It is automatically set to zero by a write or read access to buffers **RTB_LO.RTB(0,1)** and **RTB_HI.RTB(2,3)**. *Note: If LRB is high (no acknowledge) in slave mode, **TRX** bit is set automatically.*
BB	4	rh	**Bus Busy** 0 The IIC bus is idle, that is, a stop condition has occurred. 1 The IIC bus is active, that is, a start condition has occurred. *Note: Bit **BB** is always 0 while the IIC module is disabled.*
IRQD	5	rwh	**IIC Interrupt Request Bit for Data Transfer Events** [1)] 0 No interrupt request pending. 1 A data transfer event interrupt request is pending. **IRQD** is set after the acknowledge bit of the last byte has been received or transmitted, and is cleared automatically upon a complete read or write access to the buffers **RTB_LO.RTB(0,1)** and **RTB_HI.RTB(2,3**. New data transfers will start immediately after clearing **IRQD**. Do not access any register until next interrupt. If a multi byte write could not be finished in slave mode because of missing acknowledge, then the data interrupt is followed by an end of transmission interrupt. The number of bytes sent can be read from **CO**. The data interrupt must have higher priority than **IRQE**.

Field	Bits	Type	Description
IRQP	6	rwh	**IIC Interrupt Request Bit for Protocol Events** [1)] 0 No interrupt request pending. 1 A protocol event interrupt request is pending. **IRQP** is set when bit **SLA** or bit **AL** is set (↓), and must be cleared via software. If the IIC has been selected by an other master, the software must look up the required transmission direction by reading the received address and direction bit, stored in **RTB_LO.RTB0**. The **TRX** bit must be set by software correspondingly.
IRQE	7	rwh	**IIC Interrupt Request Bit for Data Transmission End** [1)] 0 No interrupt request pending. 1 A receive end event interrupt request is pending (a stop is detected). **IRQE** is automatically cleared upon a start condition. **IRQE** is not activated in init-mode. IRQE must always be deleted to continue transmission. *Note: In slave mode **IRQE** is set after the transmission is finished. This can also be after a stop or RSC condition. In this case the slave is not selected any more. This bit is also set, if a transmission is stopped by a missing acknowledge. In this case the bit must be cleared by software.*
CO	10:8	rh	**Counter of Transmitted Bytes Since Last Data Interrupt.** If a multi byte transmission could not be finished because of missing acknowledge, the number of correctly transferred bytes can be read from **CO**. It is automatically set to zero by the correct number (defined by **CI**) of write/read accesses to the Receive Transmit Buffers **RTB_HI** and **RTB_LO**. 000 No Bytes 001 1 Byte 010 2 Bytes 011 3 Bytes 100 4 Bytes The number of legal bytes depends on the data buffer size (CI). Writing to this bitfield does not affect its content. If **IIC_CON.WMEN** is set, **WM** is mirrored here.
RMEN	15	rwh	**Read Mirror Enable** 0 Read mirror is not active 1 Read mirror is active *Note: If **IIC_CON.WMEN** is set **RMEN** can not be set and will remain zero. If **RMEN** and **IIC_CON.WMEN** are set simultaneously to 1, both will remain what they are, only one of them can be set to 1.*
RESERVED	14:11	r	If **IIC_CON.WMEN** is cleared, then this bitfield is reserved, these bits must be left at their reset values.

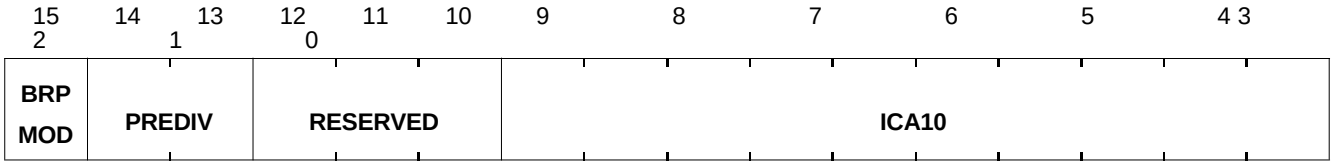
[1)] While either **IRQD**, **IRQP or IRQE** is set and the IIC module is in master mode or has been selected as a slave, the IIC clock line is held low which prevents further transfers on the IIC bus.
The clock line of the IIC bus is released when **IRQD**, **IRQE and IRQP** are cleared. Only in this case the next IIC bus action can take place.
Interrupt request bits may be set or cleared via software, for example, to control the IIC bus.

9.1.4.4 IIC Bus Control Registers

For IIC_CON.M10 = 1

IIC_ADR

Bus Address Register **Reset value: 0000_H**



Field	Bits	Type	Description
ICA10	9:0	rw	**Node Address in 10-Bit Mode** *Note: Access is only possible in the 10-bit mode (**IIC_CON.M10** = 1).*
PREDIV	14:13	rw	**Pre Divider for Baud Rate Generation** 00 Pre-divider is disabled 01 Pre-divider factor 8 is enabled 10 Pre-divider factor 64 is enabled 11 Reserved, do not use *Note: Refer to Table 57 on page 280.*
BRPMOD	15	rw	Baud Rate Prescaler Mode 0 Mode 0 is enabled (by default) 1 Mode 1 is enabled. *Note: Refer to **Table 57 IIC-Bus Baud Rate Selection for BRPMOD = 0 (on Page 280)**.*
RESERVED	12:10, 1	r	Reserved, these bits must be left at their reset values.

For IIC_CON.M10 = 0

IIC_ADR

Bus Address Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRPMOD	PREDIV		RESERVED				ICA7						RESERVED		

Field	Bits	Type	Description
ICA7	7:1	rw	Node Address in 7-Bit Mode *Note: Access is limited to this bitfield in the 7-bit mode (**IIC_CON.M10** = 0).*
PREDIV	14:13	rw	**Pre Divider for Baud Rate Generation** 00 Pre-divider is disabled 01 Pre-divider factor 8 is enabled 10 Pre-divider factor 64 is enabled 11 Reserved, do not use *Note: See **Table 57 IIC-Bus Baud Rate Selection for BRPMOD = 0 (on Page 280)**.*
BRPMOD	15	rw	Baud Rate Prescaler Mode 0 Mode 0 is enabled (by default) 1 Mode 1 is enabled. *Note: See **Table 57**.*
RESERVED	12:8, 1	r	Reserved, these bits must be left at their reset values.

IIC_CFG

Bus Configuration Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRP							RESERVED		SCLN	RESERVED					SDAN

Field	Bits	Type	Description
SDAEN	0	rw	**Enable Input for Data Pin** These bits determine if the SDA pin for the IIC data line is connected. 0 SDA pin is disconnected. 1 SDA pin is connected to IIC data line.

Field	Bits	Type	Description
SCLEN	4	rw	**Enable Input for Clock Pin** These bits determine if the SCL pin for the IIC clock line is connected. 0 SCL pin is disconnected. 1 SCL pin is connected to IIC clock line.
BRP	15:8	rw	**Baud Rate Prescaler** Determines the baud rate for the IIC channel(s). The prescaler may operate in two modes. Bit **IIC_CON.BRPMOD** selects the actual mode. Bit field **IIC_CON.PREDIV** selects an additional predivider. *Note: Refer to **Table 57 IIC-Bus Baud Rate Selection for BRPMOD = 0 (on Page 280)**.*
RESERVED	7:5, 3:1	r	Reserved, these bits must be left at their reset values.

9.1.4.4.1 Baud Rate Selection

To give the user high flexibility in selection of CPU frequency and baud rate without constraints to baud rate accuracy, a flexible baud rate generator has been implemented. It uses two different modes (**IIC_ADR.BRPMOD**) and an additional predivider (**IIC_ADR.PREDIV**). Low baud rates may be configured at high precision in mode 0 which is compatible with older versions. High baud rates may be configured precisely in mode 1.

Mode 0: Reciprocal Divider

The resulting baud rate is (refer to **IIC_CFG.BRP**):

If PREDIV = 0:

$$B_{IIC} = \frac{f_{IIC}}{4 \cdot (BRP + 1)} \qquad BRP = \frac{f_{IIC}}{4 \cdot B_{IIC}} - 1 \tag{56}$$

If PREDIV != 0:

$$B_{IIC} = \frac{1}{4 \cdot (BRP + 1) \cdot PREDIV} \cdot f_{IIC} \qquad BRP = \frac{f_{IIC}}{4 \cdot B_{IIC} \cdot PREDIV} - 1 \tag{57}$$

Mode 1: Fractional Divider

The resulting baud rate is:

$$B_{IIC} = \frac{f_{IIC} \cdot BRP}{1024} \cdot \frac{1}{PREDIV} \qquad BRP = \frac{1024 \cdot B_{IIC} \cdot PREDIV}{f_{cpu}} \tag{58}$$

Table 57 gives Baud Rate Selection.

Table 57 IIC-Bus Baud Rate Selection for BRPMOD = 0

BRPMOD = 0	BRP @ 100kBaud		BRP @ 400kBaud	
f_{cpu} [MHz]	PREDIV = 00_B	PREDIV = 01_B	PREDIV = 00_B	PREDIV = 01_B
100	$F9_H$	$1E_H$	$3E_H$	07_H
50	$7C_H$	$0F_H$	$1E_H$	03_H
24	$3B_H$	06_H	$0E_H$	-
20	31_H	05_H	$0C_H$	-
16	27_H	04_H	09_H	-

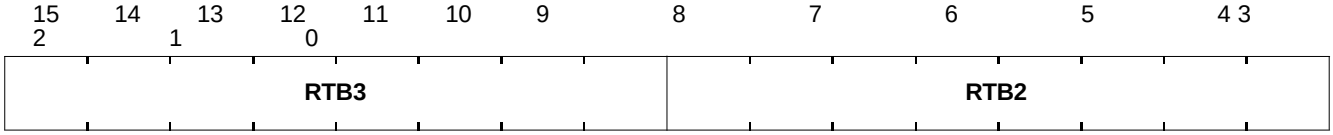
Table 58 IIC-Bus Baud Rate Selection for BRPMOD = 1

BRPMOD = 1	BRP @ 100kBaud			BRP @ 400kBaud		
f_{cpu} [MHz]	PREDIV=00_B	PREDIV=01_B	PREDIV=10_B	PREDIV=00_B	PREDIV=01_B	PREDIV=10_B
100	-	-	42_H	-	21_H	-
50	-	-	83_H	-	42_H	-
24	-	22_H	-	-	89_H	-
20	-	29_H	-	-	$A4_H$	-
16	-	33_H	-	-	CD_H	-

9.1.4.5 IIC Receive Transmit Buffers

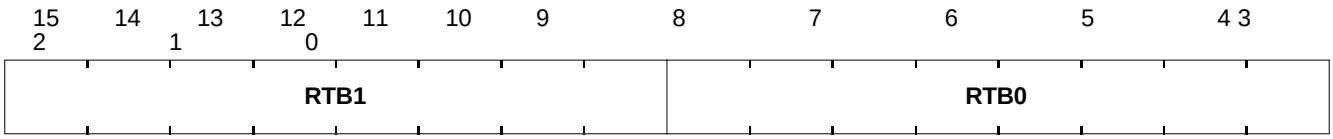
RTB_HI

High Receive Transmit Buffer **Reset value: 0000_H**



RTB_LO

Low Receive Transmit Buffer **Reset value: 0000_H**



Field	Bits	Type	Description
RTBx (x = 0 to 3)	31:0	rwh	**Receive/Transmit Buffer** [1] The buffers contain the data to be sent/received. The buffer size can be set in bitfield **IIC_CON.CI** (from 1 up to 4 bytes). **RTB0** is sent/received first.

1) If bit **IIC_CON.INT** is set to zero and all bytes (specified in **IIC_CON.CI**) of **RTB0..3** are read/written (depending on bit **IIC_CON.TRX**), **IIC_ST.IRQD** is cleared by hardware after completion of this access.

9.1.5 Reset Behavior

All resets are handled asynchronously. All registers are reset to their reset values.

9.1.6 Interrupts

Table 59 Interrupt Sources

Interrupt	Signal	Description
Data	IC_INT_D_O	Interrupt is requested after the acknowledge bit of the last byte has been received or transmitted.
Data Error	IC_INT_D_O	Interrupt is requested if a multi byte write could not be finished in slave mode because of missing acknowledge, then the data interrupt is followed by an end of transmission interrupt.
Protocol: Arbitration Lost	IIC_INT_P_O	Interrupt is requested when the IIC module has tried to become master on the bus but has lost the arbitration.
Protocol: Slave Mode after Lost Arbitration	IIC_INT_P_O	Interrupt is requested if multimaster mode is selected and the IIC module temporarily switches to slave mode after a lost arbitration.

Table 59 Interrupt Sources (cont'd)

Interrupt	Signal	Description
Protocol: Slave Mode after Device Address	IIC_INT_P_O	Interrupt is requested if multimaster mode is selected and the IIC module temporarily switches to slave mode after a lost arbitration.
Data Transmission End after Stop Condition	IIC_INT_E_O	Interrupt is requested after transmission is finished by a stop condition.
Data Transmission End after RSC Condition	IIC_INT_E_O	Interrupt is requested after transmission is finished by a repeated start condition (RSC).
Data Transmission End after missing Acknowledge	IIC_INT_E_O	Interrupt is also requested if a transmission is stopped by a missing acknowledge.

9.1.7 Synchronization

In the Master mode, the SCL line is controlled by the IIC Module. Sent and received data is only valid if SCL is high. With SCL going down, all modules are starting to count down their low period. During the low period all connected modules are allowed to hold SCL low. As the physical bus connection is wired-AND, SCL remains low until the device with the longest low period enters the high state. Then the device with the shortest high period pulls SCL low again.

9.1.8 Programming

It is strictly recommended not to write to the IIC registers (except for interrupt handling) when the IIC is working. This is indicated by the **IIC_CON.BUM** bit (in the master mode) and the interrupt flags. In the initial mode all registers can be written. In the master mode the IIC is working as long as bit **IIC_CON.BUM** is set, in the slave mode the IIC is working from receiving a start condition until receiving the next stop condition. Change of transmit direction is possible only after a protocol interrupt (IRQP) or in the initialization mode (MOD = 00_b).

9.1.8.1 Initialization

Before data can be sent or received, the data buffer size must be set in the bit field **IIC_CON.CI** (this is only necessary if buffer greater than one byte is available). To decide if the slave/master or multimaster mode is required, the **IIC_CON.MOD** bits must be programmed.

9.1.8.2 Repeated Start Condition

IIC_CON.RSC must be set to one.

9.1.8.3 Start Condition

To generate a start condition, the IIC must be in the master mode. If **IIC_CON.BUM** is set, a start condition is sent and the transmission is started. The slave returns the acknowledge bit that is indicated by **IIC_ST.LRB**.

9.1.8.4 Sending Data Bytes

To sent bytes it is only necessary to write data bytes to the transmit buffer every time a data interrupt (**IIC_ST.IRQD**) occurs.

9.1.8.5 Stop Condition

To stop the transmission, the **IIC_CON.BUM** bit must be set to zero, or the **IIC_CON.STP** bit must be set to one.

9.1.8.6 Receiving Data Bytes

To receive bytes it is necessary to set the **IIC_CON.TRX** bit to zero. The bytes can be read after every data interrupt (**IIC_ST.IRQD**). After a stop condition (protocol interrupt **IIC_ST.IRQE**), the count bit field **IIC_ST.CO** must be read (in case the buffer size, as defined in **IIC_CON.CI**, is greater than one byte to decide which bytes in the receive buffer were received in the last transmission cycle.

9.2 Synchronous Serial Interface

System Integration

- Supply domain: VDD_LD1
- Chip internal interfaces:
  - Clock domain: gclk_dsp_per drives the internal module clock ssc1_clk
  - Bus domain: TEAKLite Z-Bus
  - Interrupt sources: SSC_TX_INT, SSC_RX_INT, SSC_ERR_INT
- Monitor Pins: Refer to **Section 9.7.10 Internal Signal Monitoring (on Page 435)**.

9.2.1 Introduction

The Synchronous Serial Interface (SSC) supports both duplex and half-duplex serial synchronous communication up to 13 Mbps (@ 26 MHz module clock). The serial clock signal can be generated by the SSC itself (Master Mode) or can be received from an external master (Slave Mode). Data width, shift direction, clock polarity, and phase are programmable. This allows communication with SPI-compatible devices. Transmission and reception of data is double-buffered. A 16-bit baudrate generator provides the SSC with a separate serial clock signal.

Features

- Master and Slave Mode operation
  - Duplex or half-duplex operation
- Flexible data format
  - Programmable number of data bits: 2 to 16 bits
  - Programmable shift direction: LSB or MSB shift first
  - Programmable clock polarity: idle low or high state for the shift clock
  - Programmable clock/data phase: data shift with leading or trailing edge of the shift clock
- Baudrate generation from 396.72 up to 13 Mbps (@ 26 MHz module clock).
- Interrupt generation
  - On a transmitter empty condition
  - On a receiver full condition
  - On an error condition (receive, phase, baudrate, transmit error)
- FIFO
  - Up to 32 stage receive FIFO (RXFIFO)
  - Up to 32 stage transmit FIFO (TXFIFO)
  - Independent control of RXFIFO and TXFIFO
  - 16-Bit FIFO data width
  - Programmable Receive/Transmit Interrupt Trigger Level
  - Receive and transmit FIFO filling level indication
  - Overrun error generation
  - Underflow error generation

Figure 101 shows all functional relevant interfaces associated with the SSC Kernel.

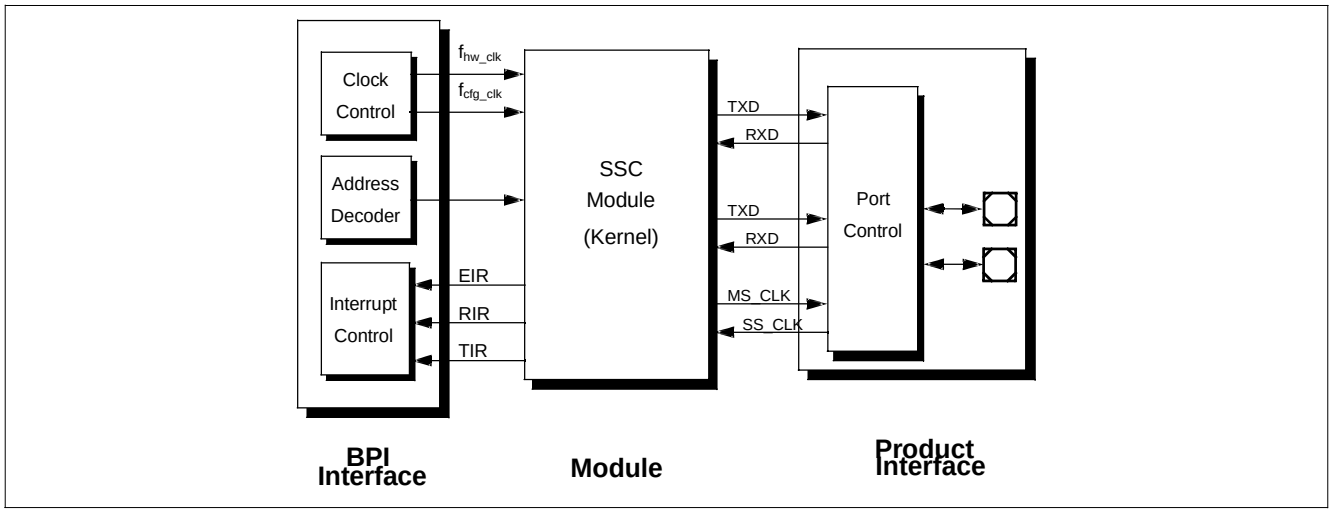


Figure 101 SSC Interface Diagram[1)]

9.2.2 General Operation

The SSC supports duplex and half-duplex synchronous communication up to 13 Mbps (@ 26 MHz module clock). The serial clock signal can be generated by the SSC itself (Master Mode) or can be received from an external master (Slave Mode). Data width, shift direction, clock polarity, and phase are programmable. This allows communication with SPI-compatible devices. Transmission and reception of data is double-buffered. A 16-bit baudrate generator provides the SSC with a separate serial clock signal.

The high-speed synchronous serial interface can be configured in a very flexible way, so it can be used with other synchronous serial interfaces, can serve for master/slave or multimaster interconnections or can operate compatible with the popular SPI interface. Thus, the SSC can be used to communicate with shift registers (IO expansion), peripherals (EEPROMs, etc.) or other controllers (networking). The SSC supports half-duplex and duplex communication. Data is transmitted or received on lines TXD and RXD, normally connected with pins MTSR (Master Transmit/Slave Receive) and MRST (Master Receive/Slave Transmit). The clock signal is output via line MS_CLK (Master Serial Shift Clock) or input via line SS_CLK (Slave Serial Shift Clock). Both lines are normally connoted to pin SCLK. These pins are alternate functions of port pins.

1) For information about hw_clk, refer to **Table 61 SSC Register Summary (on Page 297)**.

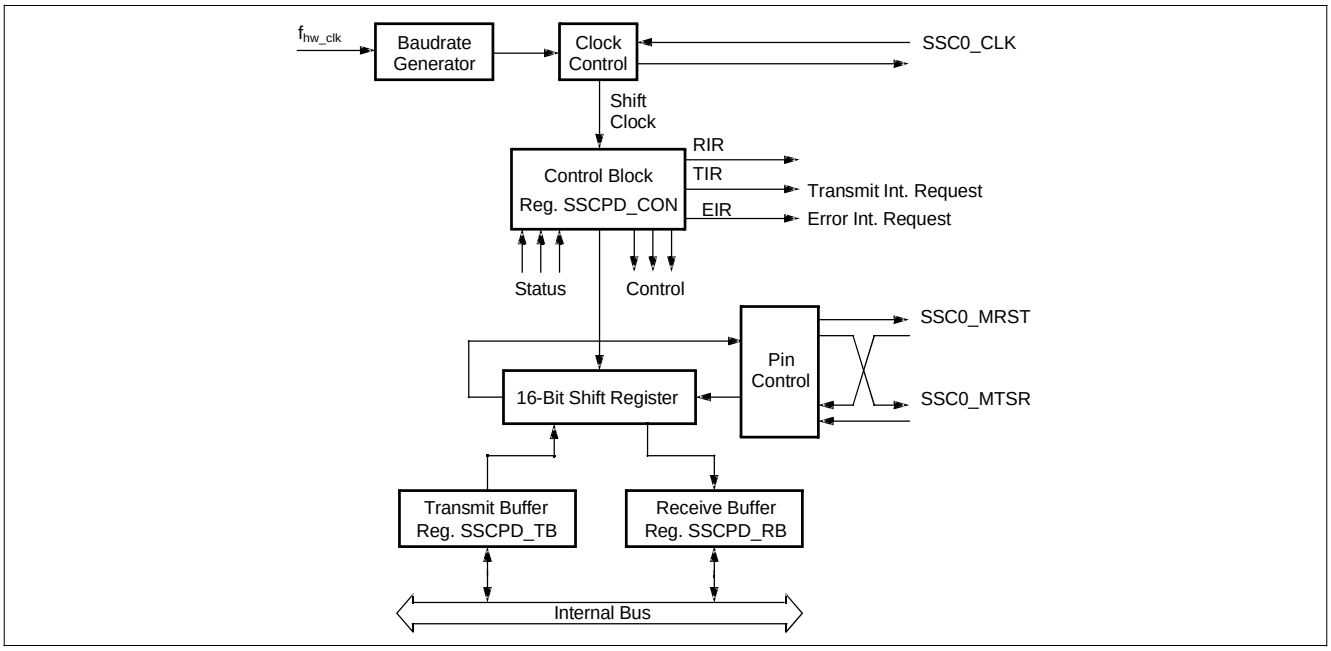


Figure 102 Synchronous Serial Channel SSC Block Diagram

9.2.2.1 Operating Mode Selection

The operating mode of the serial channel SSC is controlled by its control register **SSCPD_CON**. This register serves two purposes:

- During programming (SSC disabled by **SSCPD_CON.EN** = 0), it provides access to a set of control bits
- During operation (SSC enabled by **SSCPD_CON.EN** = 1), it provides access to a set of status flags.

The shift register of the SSC is connected to both the transmit lines and the receive lines via the pin control logic (see block diagram in **Figure 102**). Transmission and reception of serial data are synchronized and take place at the same time, That is, the same number of transmitted bits is also received. Transmit data is written into the Transmit Buffer Register (**SSCPD_TB**). It is moved to the shift register as soon as it is empty. A SSC master (**SSCPD_CON.MS** = 1) immediately begins transmitting, while an SSC slave (**MS** = 0) will wait for an active shift clock. When the transfer starts, the busy flag (**SSCPD_CON.BSY**) is set and the Transmit Interrupt Request line TIR is activated to indicate that register **SSCPD_TB** may be reloaded again.

When the programmed number of bits (2...16) has been transferred, the content of the shift register is moved to the Receive Buffer Register, (**SSCPD_RB**), and the Receive Interrupt Request line RIR is activated. If no further transfer takes place (**SSCPD_TB** is empty), **BSY** is cleared at the same time. Software should not modify **BSY** as it is hardware controlled.

Note: Only one SSC can be master at a given time.

The transfer of serial data bits can be programmed in many ways:

- Data width can be specified from 2 bits to 16 bits
- Transfer may start with either the LSB or the MSB
- Shift clock may be idle low or idle high
- Data bits may be shifted with the leading edge or the trailing edge of the shift clock signal
- Baudrate may be set from 396.72 Baud up to 13 Mbps (@ 26 MHz module clock)
- Shift clock can be generated (MS_CLK) or can be received (SS_CLK)

These features allow the adaptation of the SSC to a wide range of applications in which serial data transfer is required.

Data Width Selection supports the transfer of frames of any data length, from 2-bit “characters” up to 16-bit “characters”. Starting with the LSB (**SSCPD_CON.HB** = 0) allows communication with SSC devices in

Synchronous Mode or with 8051 like serial interfaces for example. Starting with the MSB (**HB** = 1) allows operation compatible with the SPI interface.

Regardless of the data width selected and whether the MSB or the LSB is transmitted first, the transfer data is always right-aligned in registers **SSCPD_TB** and **SSCPD_RB**, with the LSB of the transfer data in bit 0 of these registers. The data bits are rearranged for transfer by the internal shift register logic. The unselected bits of **SSCPD_TB** are ignored; the unselected bits of **SSCPD_RB** will not be valid and should be ignored by the receiver service routine.

The Clock Control allows the adaptation of transmit and receive behavior of the SSC to a variety of serial interfaces. A specific shift clock edge (rising or falling) is used to shift out transmit data, while the other shift clock edge is used to latch in receive data. Bit **SSCPD_CON.PH** selects the leading edge or the trailing edge for each function. Bit **SSCPD_CON.PO** selects the level of the shift clock line in the idle state. Thus, for an idle-high clock, the leading edge is a falling one, a 1-to-0 transition (see **Figure 103**).

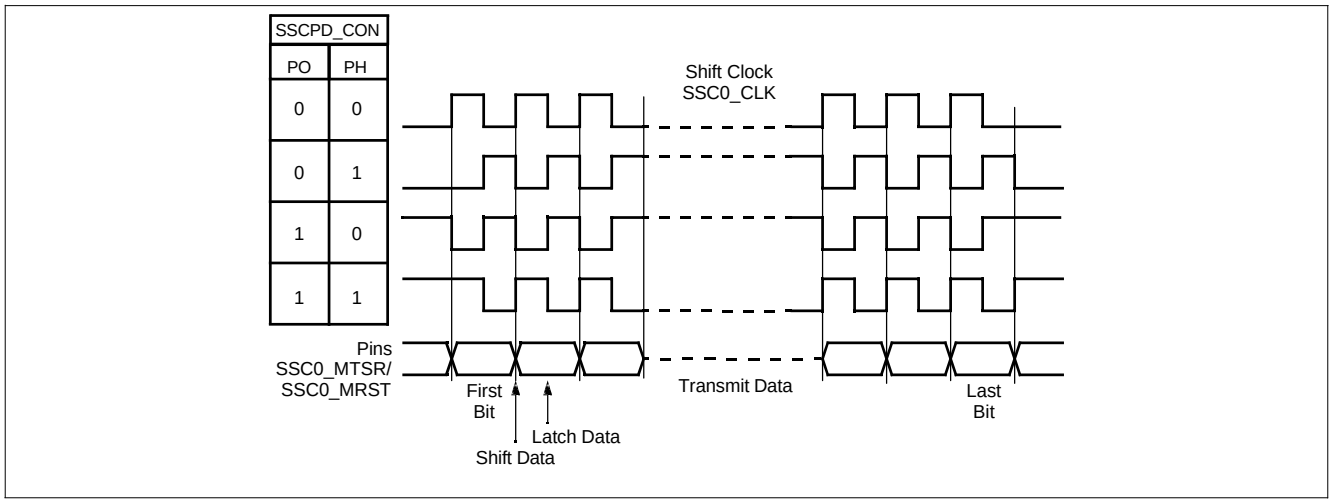


Figure 103 Serial Clock Phase and Polarity Options

9.2.2.2 Duplex Operation

The various devices are connected through three lines. The definition of these lines is always determined by the master: The line connected to the master's data output line TXD is the transmit line; the receive line is connected to its data input line RXD; the shift clock line is either MS_CLK or SS_CLK. Only the device selected for master operation generates and outputs the shift clock on line MS_CLK. All slaves receive this clock; therefore, their pin SCLK must be switched to input mode. The output of the master's shift register is connected to the external transmit line, which in turn is connected to the slaves' shift register input. The output of the slaves' shift register is connected to the external receive line in order to enable the master to receive the data shifted out of the slave. The external connections are hard-wired, the function and direction of these pins is determined by the master or slave operation of the individual device.

Note: The shift direction shown in the figure applies for MSB-first operation as well as for LSB-first operation.
When initializing the devices in this configuration, one device must be selected for master operation while all other devices must be programmed for slave operation. Initialization includes the operating mode of the device SSC and also the function of the respective port lines.

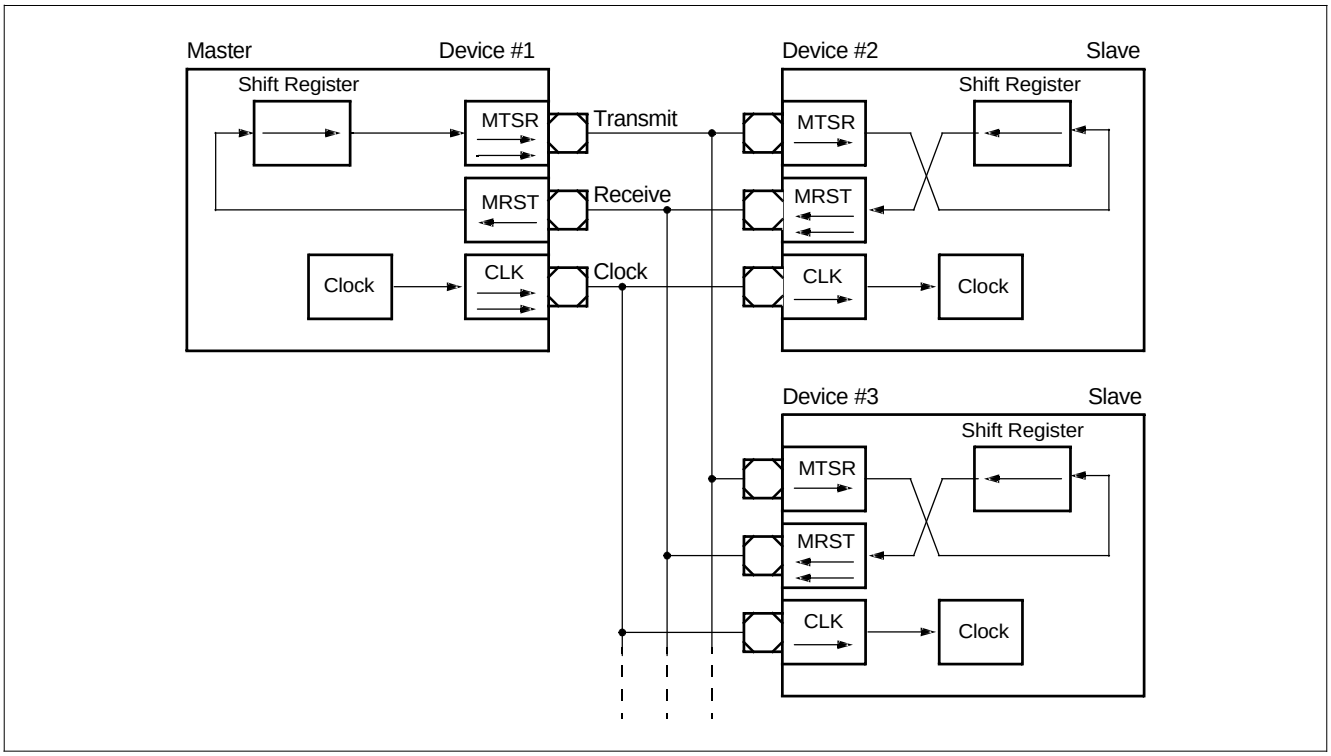


Figure 104 SSC Duplex Configuration

The data output pins MRST of all slave devices are connected together onto the one receive line in the configuration shown in **Figure 104**. During a transfer each slave shifts out data from its shift register. There are two ways to avoid collisions on the receive line due to different slave data:

- Only one slave drives the line, that is, enables the driver of its MRST pin. All the other slaves must have their MRST pins programmed as input so only one slave can put its data onto the master's receive line. Only receiving data from the master is possible. The master selects the slave device from which it expects data either by separate select lines, or by sending a special command to this slave. The selected slave then switches its MRST line to output until it gets a de-selection signal or command.
- The slaves use open drain output on MRST. This forms a wired-AND connection. The receive line needs an external pull-up in this case. Corruption of the data on the receive line sent by the selected slave is avoided when all slaves not selected for transmission to the master only send ones (1s). Because this high level is not actively driven onto the line, but only held through the pull-up device, the selected slave can pull this line actively to a low level when transmitting a zero bit. The master selects the slave device from which it expects data either by separate select lines or by sending a special command to this slave.

After performing the necessary initialization of the SSC, the serial interfaces can be enabled. For a master device, the alternate clock line will now go to its programmed polarity. The alternate data line will go to either 0 or 1 until the first transfer will start. After a transfer, the alternate data line will always remain at the logic level of the last transmitted data bit.

When the serial interfaces are enabled, the master device can initiate the first data transfer by writing the transmit data into register **SSCPD_TB**. This value is copied into the shift register (assumed to be empty at this time), and the selected first bit of the transmit data will be placed onto the TXD line on the next clock from the baudrate generator (transmission starts only if **SSCPD_CON.EN** = 1). Depending on the selected clock phase, a clock pulse will also be generated on the MS_CLK line. At the same time, with the opposite clock edge, the master latches and shifts in the data detected at its input line RXD. This “exchanges” the transmit data with the receive data. Because the clock line is connected to all slaves, their shift registers will be shifted synchronously with the master's shift register—shifting out the data contained in the registers, and shifting in the data detected at the input line. After the preprogrammed number of clock pulses (via the data width selection), the data transmitted by the master

is contained in all the slaves' shift registers, while the master's shift register holds the data of the selected slave. In the master and all slaves, the content of the shift register are copied into the receive buffer RB and the receive interrupt line RIR is activated.

A slave device will immediately output the selected first bit (MSB or LSB of the transfer data) at line RXD when the contents of the transmit buffer are copied into the slave's shift register. Bit **SSCPD_CON.BSY** is not set until the first clock edge at SS_CLK appears. The slave device will not wait for the next clock from the baudrate generator, as the master does. The reason for this is that, depending on the selected clock phase, the first clock edge generated by the master may already be used to clock in the first data bit. Thus, the slave's first data bit must already be valid at this time.

*Note: On the SSC, a transmission **and** a reception takes place at the same time, regardless of whether valid data has been transmitted or received.*

9.2.2.3 Half-Duplex Operation

In a Half-Duplex Mode, only one data line is necessary for both receiving **and** transmitting of data. The data exchange line is connected to both the MTSR and MRST pins of each device, the shift clock line is connected to the SCLK pin.

The master device controls the data transfer by generating the shift clock, while the slave devices receive it. Due to the fact that all transmit and receive pins are connected to the one data exchange line, serial data may be moved between arbitrary stations.

Similar to the Duplex Mode, there are two ways to avoid collisions on the data exchange line:

- only the transmitting device may enable its transmit pin driver
- the non-transmitting devices use open drain output and send only ones.

Because the data inputs and outputs are connected together, a transmitting device will clock in its own data at the input pin (MRST for a master device, MTSR for a slave). By this method, any corruptions on the common data exchange line are detected if the received data is not equal to the transmitted data.

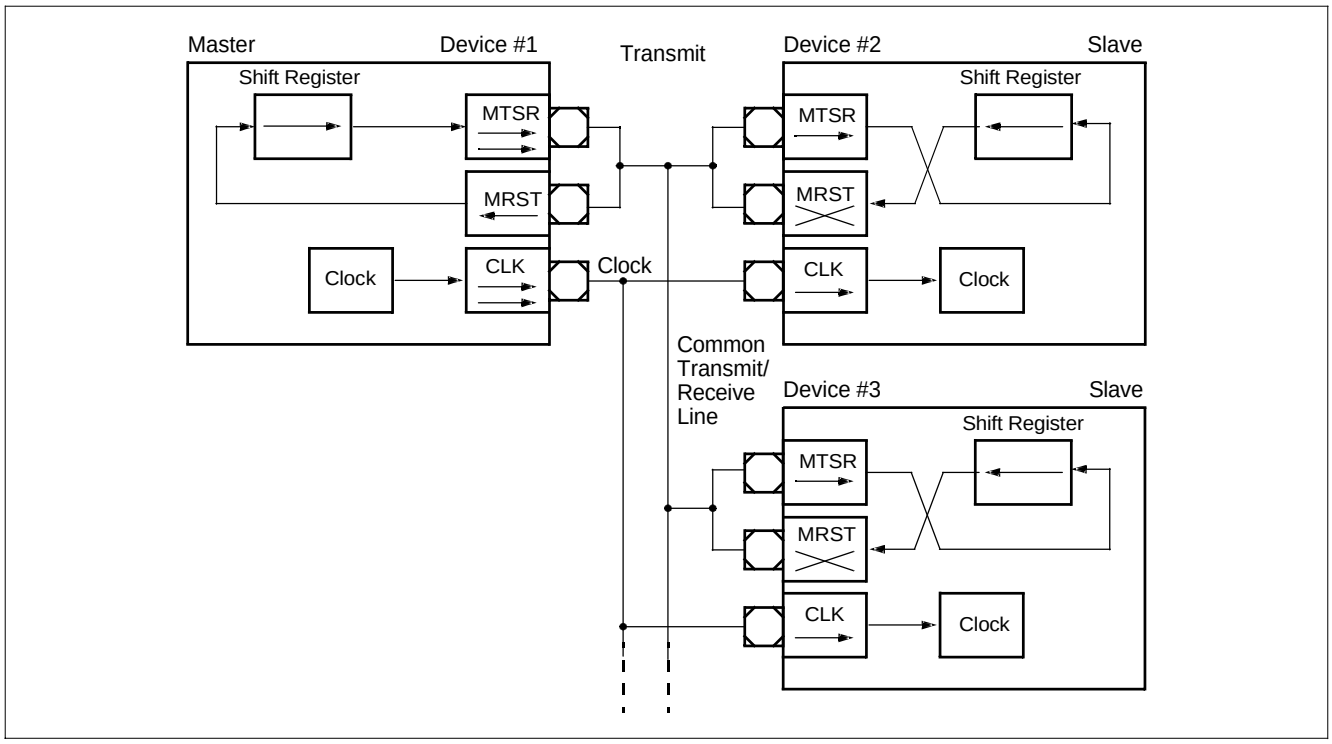


Figure 105 SSC Half-Duplex Configuration

9.2.2.4 Continuous Transfers

When the transmit interrupt request flag is set, it indicates that the transmit buffer **SSCPD_TB** is empty and ready to be loaded with the next transmit data. If **SSCPD_TB** has been reloaded by the time the current transmission is finished, the data is immediately transferred to the shift register and the next transmission will start without any additional delay. On the data line, there is no gap between the two successive frames. For example, two byte transfers would look the same as one word transfer. This feature can be used to interface with devices that can operate with or require more than 16 data bits per transfer. It is just a matter of software, how long a total data frame length can be. This option can also be used to interface to byte-wide and word-wide devices on the same serial bus, for instance.

Note: Of course, this can only happen in multiples of the selected basic data width, because it would require disabling/enabling of the SSC to reprogram the basic data width on-the-fly.

9.2.2.5 FIFO Operations

9.2.2.5.1 Transmit FIFO

The transmit FIFO (TXFIFO) provides the following functionality:

- Enable/disable control
- Programmable filling level for transmit interrupt generation
- Filling level indication
- FIFO clear (flush) operation
- FIFO overflow error generation

The 32 stage transmit FIFO is controlled by the **SSCPD_TXFCON** control register. When bit **SSCPD_TXFCON.TXFEN** is set, the transmit FIFO is enabled. The interrupt trigger level defined by **SSCPD_TXFCON.TXFITL** defines the filling level of the TXFIFO at which a transmit interrupt TIR is generated. These interrupt is always generated when the filling level of the transmit FIFO is equal to or less than the value stored in **SSCPD_TXFCON.TXFITL**.

Bit field **SSCPD_FSTAT.TXFFL** indicates the number of entries that are actually written (valid) in the TXFIFO. Therefore, the software can verify, in the interrupt service routine, for instance, how many bytes can be still written into the transmit FIFO via register **SSCPD_TB** without getting an overrun error.

The transmit FIFO cannot be accessed directly. All data write operations into the TXFIFO are executed by writing into the **SSCPD_TB** register.

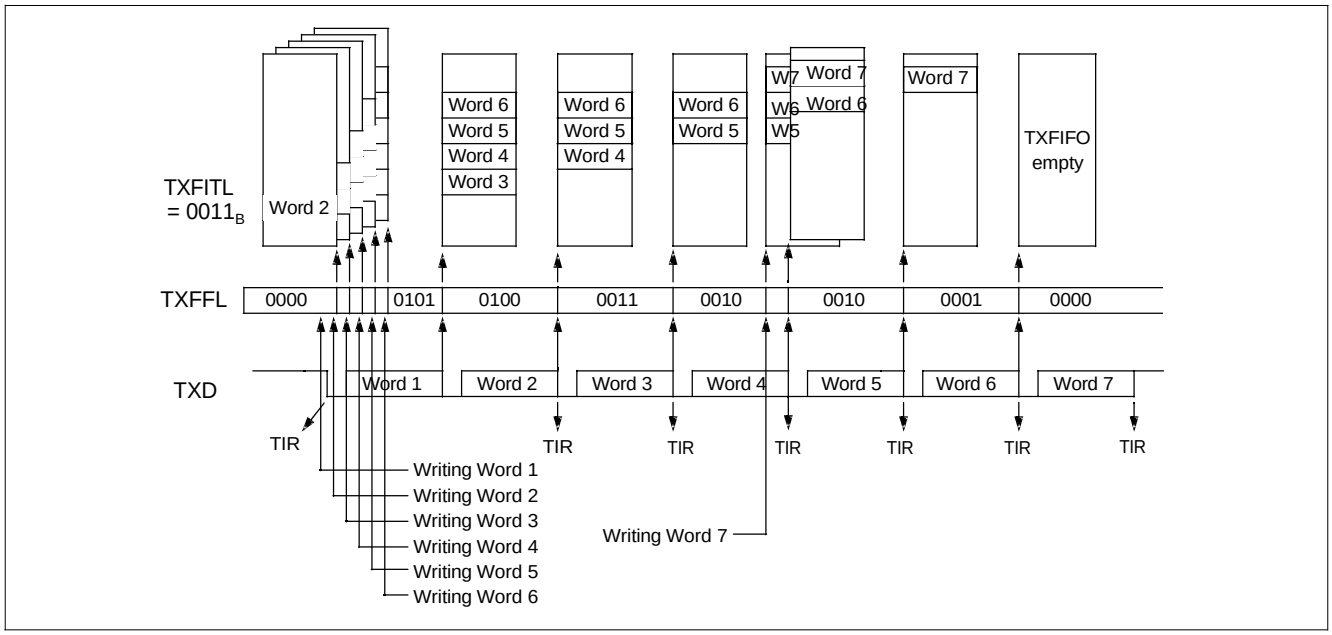


Figure 106 Transmit FIFO Operation Example

The example in **Figure 106** shows a typical transmit FIFO operation. In this example seven messages are transmitted via the TXD output line. The transmit FIFO interrupt trigger level **SSCPD_TXFCON.TXFITL** is set to 0011_B. The first message data written into the empty TXFIFO via **SSCPD_TB** is directly transferred into the transmit shift register and is not written into the FIFO. After message 1, messages 2 to 6 are written into the transmit FIFO.

When message 3 is transferred from TXFIFO into the SSC shift register, 3 other messages remain in the TXFIFO. **SSCPD_TXFCON.TXFITL** is then reached (= 3) and a transmit interrupt (TIR) is generated - at the beginning of the serial transmission. During the serial transmission of message 4, another message (message 7) is written into the TXFIFO (**SSCPD_TB** write operation). Finally, after the start of the serial transmission of message 7, the TXFIFO is again empty (and a TIR is raised).

If the TXFIFO is full and additional bytes are written into **SSCPD_TB**, the Error interrupt [EIR] will be generated with bit **SSCPD_CON.TE** set if bit **SSCPD_CON.TEN** was set. In this case, the data that was last written into the transmit FIFO is overwritten and the transmit FIFO filling level **SSCPD_FSTAT.TXFFL** is set to maximum.

The TXFIFO can be flushed or cleared by setting bit **SSCPD_TXFCON.TXFFLU**. After this TXFIFO flush operation, the TXFIFO is empty and the transmit FIFO filling level **SSCPD_FSTAT.TXFFL** is set to 000000_B. A running serial transmission is not aborted by a receive FIFO flush operation

*Note: The TXFIFO is flushed automatically with a reset operation of the SSC module and if the TXFIFO becomes disabled (resetting bit **SSCPD_TXFCON.TXFEN**) after it was previously enabled.*

9.2.2.5.2 Receive FIFO Operation

The receive FIFO (RXFIFO) provides the following functionality:

- Enable/disable control
- Programmable filling level for receive interrupt generation
- Filling level indication
- FIFO clear (flush) operation
- FIFO overflow error generation

The 32 stage receive FIFO is controlled by the **SSCPD_RXFCON** control register. When bit **SSCPD_RXFCON.RXFEN** is set, the receive FIFO is enabled. The interrupt trigger level defined by **SSCPD_RXFCON.RXFITL** defines the filling level of RXFIFO at which a receive interrupt RIR is generated. RIR

is always generated when the filling level of the receive FIFO is equal to or greater than the value stored in **SSCPD_RXFCON.RXFITL**.

Bit field **SSCPD_FSTAT.RXFFL** indicates the number of bytes that have been actually written into the FIFO and can be read out of the FIFO by a user program.

The receive FIFO cannot be accessed directly. All data read operations from the RXFIFO are executed by reading the **SSCPD_RB** register.

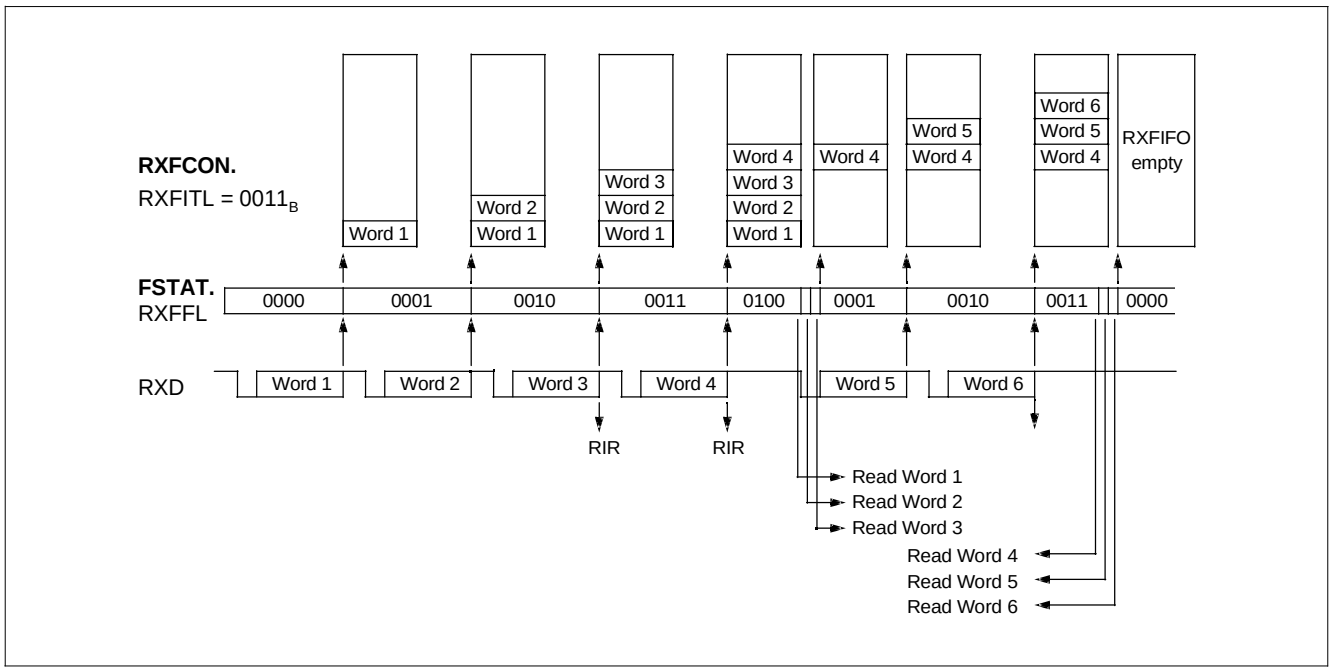


Figure 107 Receive FIFO Operation Example

The example in **Figure 107** shows a typical receive FIFO operation. In this example, six messages are received via the RXD input line. The receive FIFO interrupt trigger level **SSCPD_RXFCON.RXFITL** is set to 000011_B. Therefore, the first receive interrupt RIR is generated after the reception of message 3 (RXFIFO is filled with three messages).

After the reception of message 4, three messages are read out of the receive FIFO. After this read operation, the RXFIFO still contains one message. RIR becomes again active after two more messages (messages 5 and 6) have been received (RXFIFO filled again with 3 messages). Finally, the FIFO is cleared after three read operations.

If the RXFIFO is full and additional data are received, the Error interrupt [EIR] is generated and bit **SSCPD_CON.RE** is set if **SSCPD_CON.REN** is not cleared. In this case, the data byte last written into the receive FIFO is overwritten. With the overrun condition, the receive FIFO filling level **SSCPD_FSTAT.RXFFL** is set to maximum. If a **SSCPD_RB** read operation is executed with the RXFIFO enabled but empty, a receive interrupt RIR will be generated. In this case, the receive FIFO filling level **SSCPD_FSTAT.RXFFL** is set to 000000_B.

If the RXFIFO is available but disabled (**SSCPD_RXFCON.RXFEN** = 0) the receive operation is functionally equivalent to the receive operation of the SSC module without FIFO.

The RXFIFO can be flushed or cleared by setting bit **SSCPD_RXFCON.RXFFLU**. After this RXFIFO flush operation, the RXFIFO is empty and the receive FIFO filling level **SSCPD_FSTAT.RXFFL** is set to 000000_B. The RXFIFO is flushed automatically with a reset operation of the SSC module and if the RXFIFO becomes disabled (resetting bit **SSCPD_RXFCON.RXFEN**) after it was previously enabled. Resetting bit **SSCPD_CON.REN** without resetting **SSCPD_RXFCON.RXFEN** does not affect (reset) the RXFIFO state. This means that the receive operation of the SSC is stopped, in this case, without changing the content of the RXFIFO. After setting **SSCPD_CON.REN** again, the RXFIFO with its content is again available.

9.2.2.5.3 FIFO Transparent Mode

In Transparent Mode, a specific interrupt generation mechanism is used for receive and transmit interrupts. In general, in Transparent Mode, receive interrupts are always generated if data bytes are available in the RXFIFO. The relevant conditions for interrupt generation in Transparent Mode are:

- FIFO filling levels
- Read operations on the **SSCPD_RB** data register

Attention: There is no interrupt when writing to SSCPD_TB, only when the data is put on the Transmit line. Interrupt generation for the receive FIFO depends on the RXFIFO filling level and the execution of read operations of register **SSCPD_RB** (see **Figure 108**). Transparent Mode for the RXFIFO is enabled when bits **SSCPD_RXFCON.RXTMEN** and **SSCPD_RXFCON.RXFEN** are set.

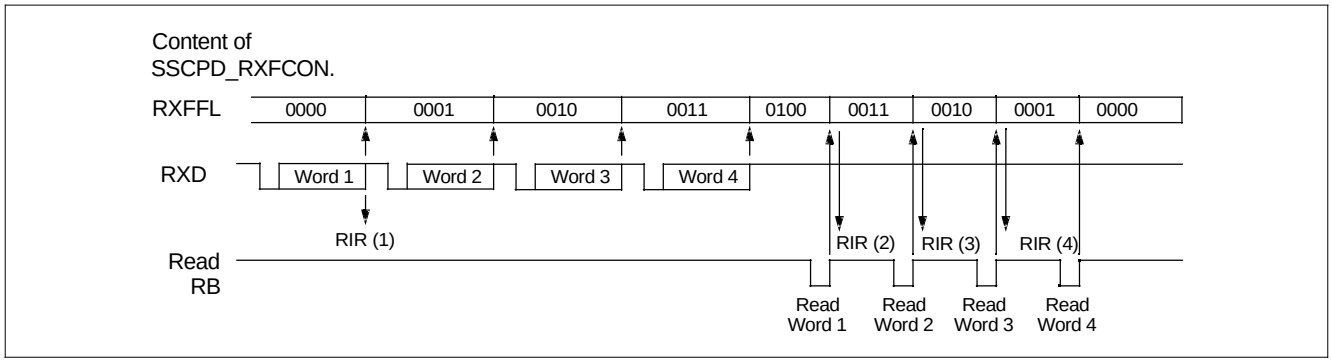


Figure 108 Transparent Mode Receive FIFO Operation

If the RXFIFO is empty, a receive interrupt RIR is always generated when the first message is written into an empty RXFIFO (**SSCPD_FSTAT.RXFFL** changes from 000000_B to 000001_B). If the RXFIFO is filled with at least one message, the occurrence of further receive interrupts depends on the read operations of register **SSCPD_RB**. The receive interrupt RIR will always be activated after a **SSCPD_RB** read operation if the RXFIFO still contains data (**SSCPD_FSTAT.RXFFL** is not equal to 000000_B). If the RXFIFO is empty after a **SSCPD_RB** read operation, no further receive interrupt will be generated.

If the RXFIFO is full (**SSCPD_TXFCON.TXTMEN** and **SSCPD_TXFCON.TXFEN** = maximum) and additional messages are received, a Error interrupt [EIR] is generated. In this case, the message last written into the receive FIFO is overwritten. If a **SSCPD_RB** read operation is executed with the RXFIFO enabled but empty (underflow condition), an Error interrupt EIR is generated in response to bit **SSCPD_CON.RE** set (if **SSCPD_CON.REN** has been previously set).

If the RXFIFO is flushed in Transparent Mode, the software must take care that a previous pending receive interrupt is ignored.

*Note: The Receive FIFO Interrupt Trigger Level bitfield **SSCPD_RXFCON.RXFITL** is a don't care in Transparent Mode.*

Interrupt generation for the transmit FIFO depends on the TXFIFO filling level and the execution of write operations to the register TB. Transparent Mode for the TXFIFO is enabled when bits **SSCPD_TXFCON.TXTMEN** and **SSCPD_TXFCON.TXFEN**.

TIR is also activated after a TXFIFO flush operation or when the TXFIFO becomes enabled (**SSCPD_TXFCON.TXTMEN** and **SSCPD_TXFCON.TXFEN** set) when it was previously disabled. In these cases, the TXFIFO is empty and ready to be filled with data.

If the TXFIFO is full (**SSCPD_TXFCON.TXTMEN** and **SSCPD_TXFCON.TXFFL** = maximum) and an additional message is written into **SSCPD_TB**, an Error interrupt [EIR] is generated after the **SSCPD_TB** write operation. In this case the data byte last written into the transmit FIFO is overwritten and an Error interrupt (EIR) is generated in response to bit **SSCPD_CON.TE** set (if **SSCPD_CON.TEN** has been previously set).

*Note: The Transmit FIFO Interrupt Trigger Level bitfield **SSCPD_TXFCON.TXFITL** is a don't care in Transparent Mode.*

9.2.2.6 Baudrate Generation

The serial channel SSC has its own dedicated 16-bit baudrate generator with 16-bit reload capability, allowing baudrate generation independent of the timers. **Figure 102 (on page 286)** shows the baudrate generator. **Figure 109** shows the baudrate generator of the SSC in more detail.

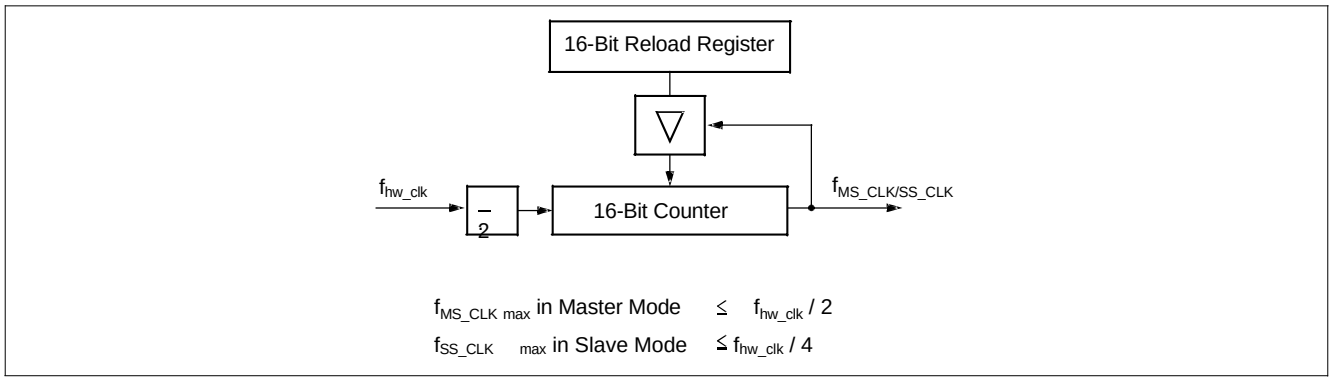


Figure 109 SSC Baudrate Generator

The baudrate generator is clocked with the module clock f_{hw_clk}. The timer counts downwards. Register **SSCPD_BR** is the dual-function Baudrate Generator/Reload register. Reading **SSCPD_BR**, while the SSC is enabled, returns the content of the timer. Reading **SSCPD_BR**, while the SSC is disabled, returns the programmed reload value. In this mode the desired reload value can be written to **SSCPD_BR**.

*Note: Never write to **SSCPD_BR** while the SSC is enabled (**SSCPD_CON.EN** must =0 to write to **SSCPD_BR**).*

The formulas below calculate either the resulting baudrate for a given reload value, or the required reload value for a given baudrate:

the required reload value for a given baudrate:

$$\text{Baudrate} = \frac{f_{hw\_clk}}{2 * (<SSCBR> + 1)} \qquad BR = \frac{f_{hw\_clk}}{2 * \text{Baudrate}} - 1 \tag{59}$$

<SSCBR> represents the contents of the reload register, taken as unsigned 16-bit integer; while Baudrate is equal to f_{MS_CLK/SS_CLK} as shown in **Figure 109**.

The maximum baudrate that can be achieved when using a module clock of 52 MHz is 26 MBaud in Master Mode (with <SSCBR> = 0000_H) or 13 MBaud in Slave Mode (with <SSCBR> = 0001_H).

Table 60 lists some possible baudrates together with the required reload values and the resulting bit times, assuming a module clock of 52 MHz.

Table 60 Typical Baudrates of the SSC (f_{hw_clk} = 52 MHz)

Reload Value	Baudrate (= f_{MS_CLK/SS_CLK})	Deviation
0000_H	26 MBaud (only in Master Mode)	0.0%
0001_H	13 MBaud	0.0%
$FFFF_H$	396.72 Baud	0.0%

9.2.2.7 Error Detection Mechanisms

The SSC is able to detect four different error conditions:

- Receive Error and Phase Error are detected in all modes
- Transmit Error and Baudrate Error are detected only in Slave Mode.

When an error is detected, the respective error flag is set and an error interrupt request is generated (see **Figure 110**): EIR IRQ line is activated during 2 CPU clocks only and only if the respective EN bit (**TEN**, **REN**, **PEN**, **BEN**) in **SSCPD_CON** has been set. The error flag is automatically reset by the Low to High IRQ pulse. No indication but a control by **SSCPD_CON** programming mode indicates the IRQ cause.

Note: The error interrupt handler must clear the associated (enabled) error flag(s) to prevent repeated interrupt requests.

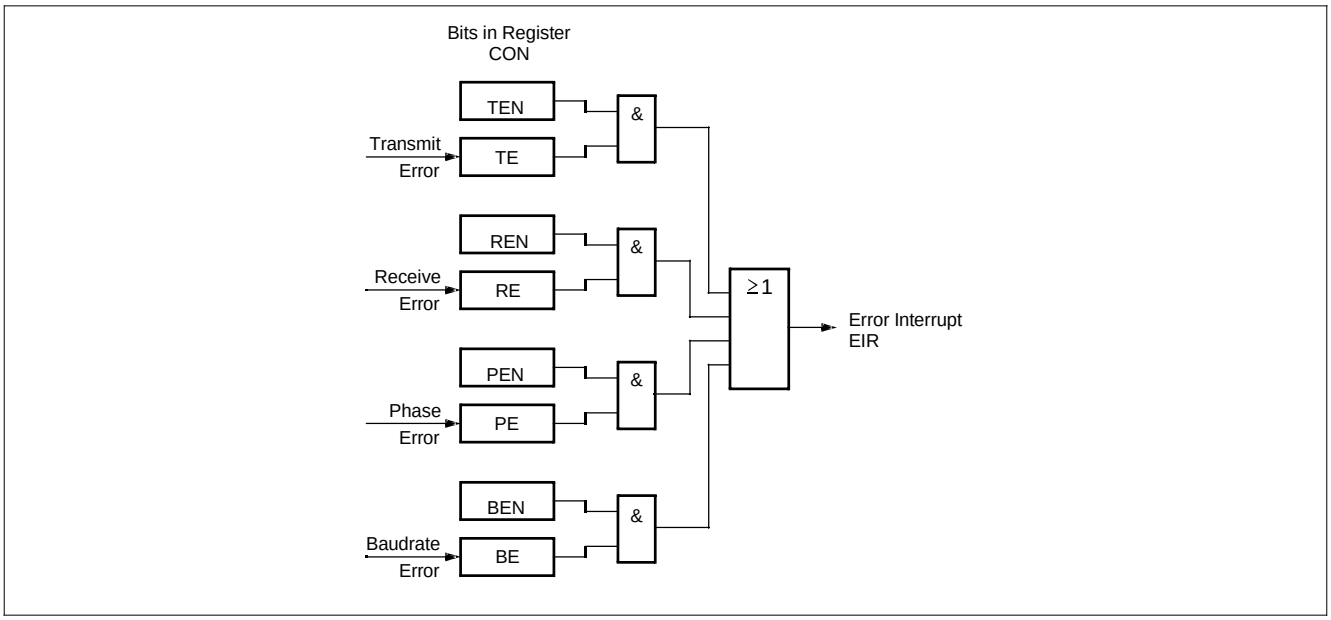


Figure 110 SSC Error Interrupt Control

A **Receive Error** (Master or Slave Mode) is detected when a new data frame is completely received but the previous data was not read out of the receive buffer register RB. This condition sets the error flag **SSCPD_CON.RE** and, when enabled via **SSCPD_CON.REN**, the error interrupt request line EIR. The old data in the receive buffer **SSCPD_RB** will be overwritten with the new value and is irretrievably lost (also when FIFO overflows).

A **Phase Error** (Master or Slave Mode) is detected when the incoming data at pin MRST (Master Mode) or MTSR (Slave Mode), sampled with the same frequency as the module clock, changes between one cycle before and two cycles after the latching edge of the shift clock signal SCLK. This condition sets the error flag **SSCPD_CON.PE** and, when enabled via **SSCPD_CON.PEN**, the error interrupt request line EIR.

A **Baudrate Error** (Slave Mode) is detected when the incoming clock signal deviates from the programmed baudrate by more than 100%, that is, it is either more than twice or less than the half of the expected baudrate. This condition sets the error flag **SSCPD_CON.BE** and, when enabled via **SSCPD_CON.BEN**, the error interrupt request line EIR. Using this error detection capability requires that the slave's baudrate generator is programmed to the same baudrate as the master device. This feature detects false additional, or missing pulses on the clock line (within a certain frame).

*Note: If this error condition occurs and bit **SSCPD_CON.AREN** = 1, an automatic reset of the SSC will be performed in case of this error. This is done to re-initialize the SSC if too few or too many clock pulses have been detected. In that case, no data are loaded from shift register to **SSCPD_RB**.*

A **Transmit Error** (Slave Mode) is detected when a transfer was initiated by the master (SSC_CLK gets active) but the transmit buffer **SSCPD_TB** of the slave was not updated since the last transfer. This condition sets the error flag **SSCPD_CON.TE** and, when enabled via **SSCPD_CON.TEN**, the error interrupt request line EIR. If a transfer starts while the transmit buffer is not updated, the slave will shift out the 'old' contents of the shift register,

which normally is the data received during the last transfer. This may lead to corruption of the data on the transmit/receive line in half-duplex mode (open drain configuration) if this slave is not selected for transmission. This mode requires that slaves not selected for transmission only shift out ones; that is, their transmit buffers must be loaded with $FFFF_H$ prior to any transfer.

Note: A slave with push/pull output drivers not selected for transmission, will normally have its output drivers switched. However, in order to avoid possible conflicts or misinterpretations, it is recommended to always load the slave's transmit buffer prior to any transfer.

A **Transmit Error** (Master or Slave Mode) is generated when a new data frame is written but the previous data was not sent. This condition sets the error flag **SSCPD_CON.TE** and, when enabled via **SSCPD_CON.TEN**, the Error interrupt request line EIR. The old data in the transmit buffer **SSCPD_TB** is overwritten with the new value and is irretrievably lost.

9.2.3 SSC Kernel Registers

Refer to **Section 10.1 PD-Bus Register Addresses** for the addresses of the SSC registers.

Table 61 SSC Register Summary

Name	Clock	Description	Access Condition
SSCPD_PISEL	hw_clk[1)]	Peripheral Input Select Register	Bit addressable
SSCPD_ID	hw_clk[1)]	Peripheral Input Select Register	Bit addressable
SSCPD_CON	hw_clk[1)]	Control Register	Bit addressable
SSCPD_BR	cfg_clk[1)]	Baudrate Timer Reload Register	Bit addressable
SSCPD_TB	cfg_clk[1)]	Transmit Buffer Register	Bit addressable
SSCPD_RB	hw_clk[1)]	Receive Buffer Register	Bit addressable
SSCPD_RXFCON	hw_clk[1)]	Receive FIFO Control Register	Bit addressable
SSCPD_TXFCON	hw_clk[1)]	Transmit FIFO Control Register	Bit addressable
SSCPD_FSTAT	hw_clk[1)]	FIFO Status Register	Bit addressable

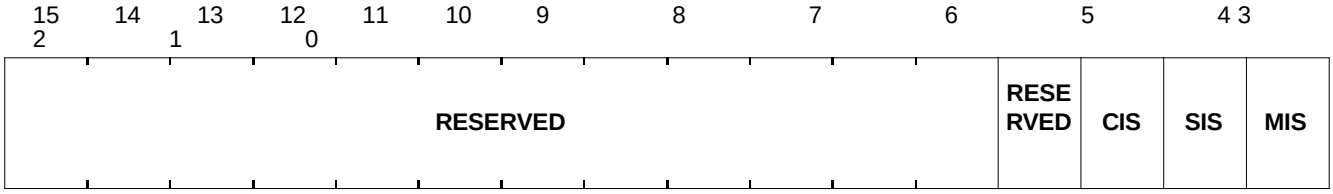
1) Refer to Clock Domain in **System Integration (on Page 284)**.

9.2.3.1 Port Input Select Register

Normally, the **SSCPD_PISEL** register controls the receiver input selection of the SSC via an input multiplexer. However, the E-GOLDvoice uses only one of the input lines of each input type, the other input is tied to logical level 0. Therefore, all bits in this register must remain at their reset default values of 0 to select the lines connected the input pads

SSCPD_PISEL

Port Input Select Register **Reset value: 0000_H**

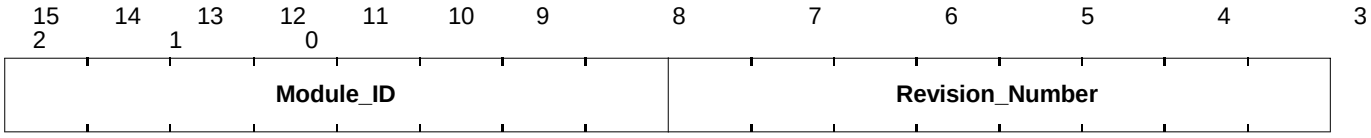


Field	Bits	Type	Description
MIS	0	rw	**Master Mode Receiver Input Select** 0 SSC0_MRST0 selected (Default) 1 SSC0_MRST1 selected, there is no input over this line
SIS	1	rw	**Slave Mode Receiver Input Select** 0 SSC0_MTST0 selected (Default) 1 SSC0_MTST1 selected, there is no input over this line
CIS	2	rw	**Slave Mode Clock Input Select** 0 SSC0_CLK0 selected (Default) 1 SSC0_CLK1 selected, there is no input over this line
RESERVED	15:3	r	Reserved; these bits must be left at their reset values.

9.2.3.1.1 SSC Identification Register

SSCPD_ID

SSC Identification Register **Reset value: 4526_H**



Field	Bits	Type	Description
Revision_Number	0:7	r	**SSC Revision Number** These hard-wired bits are used for module revision numbering.
Module_ID	8:15	r	**SSC Identification Number** These hard-wired bits are used for module identification numbering.

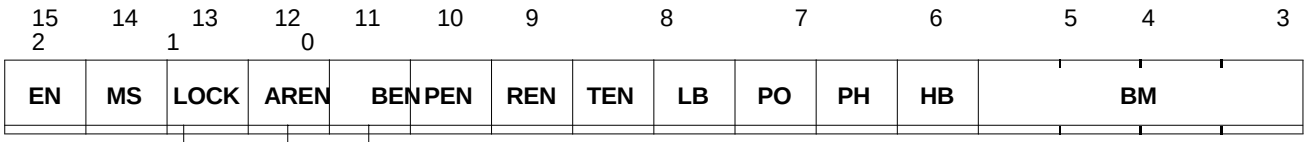
9.2.3.1.2 Configuration Register

The operating mode of the serial channel SSC is controlled by the control register **SSCPD_CON**. This register contains control bits for mode and error check selection, and status flags for error identification. Depending on bit **EN**, either control functions or status flags and master/slave control is enabled.

SSCPD_CON.EN = 0: Programming Mode

SSCPD_CON

Control Register **Reset Value: 0000_H**



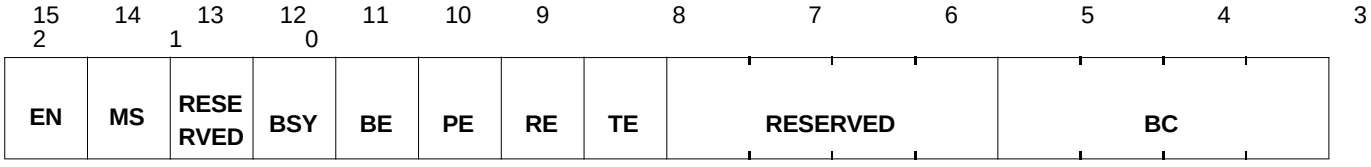
Field	Bits	Type	Description
BM	3:0	rw	Data Width Selection 0000 Reserved. Do not use this combination. 0001 - 1111 Transfer Data Width is 2...16 bit (<BM>+1)
HB	4	rw	Heading Control 0 Transmit/Receive LSB First 1 Transmit/Receive MSB First
PH	5	rw	Clock Phase Control 0 Shift transmit data on the leading clock edge, latch on trailing edge 1 Latch receive data on leading clock edge, shift on trailing edge
PO	6	rw	Clock Polarity Control 0 Idle clock line is low, leading clock edge is low-to-high transition 1 Idle clock line is high, leading clock edge is high-to-low transition
LB	7	rw	Loop Back Control 0 Normal output 1 Receive input is connected with transmit output (half-duplex mode if connecting by wiring MTSR ST). **SSCPD_TB** => **SSCPD_RB** with TIR & RIR “normal” generation. This could be useful for SSC design testing (alone).

Field	Bits	Type	Description
TEN	8	rw	Transmit Error Enable 0 Ignore transmit errors 1 Check transmit errors
REN	9	rw	Receive Error Enable 0 Ignore receive errors 1 Check receive errors
PEN	10	rw	Phase Error Enable 0 Ignore phase errors 1 Check phase errors
BEN	11	rw	Baudrate Error Enable 0 Ignore baudrate errors 1 Check baudrate errors
AREN	12	rw	Automatic Reset Enable 0 No additional action upon a baudrate error 1 The SSC is automatically reset upon a baudrate error (**SSCPD_RB** is not loaded with data from received line)
LOCK	13	r	Lock bit for the 8 MSB bits of the Transmist data register 0 Write in the 16 bits of the **SSCPD_TB** rgister 1 Lock of the 8 MSB: write possible only in the 8 LSBs of **SSCPD_TB** register
MS	14	rw	Master Select 0 Slave Mode. Operate on shift clock received via SCLK. 1 Master Mode. Generate shift clock and output it via SCLK.
EN	15	rw	Enable Bit = 0 Transmission and reception disabled. Access to control bits.

SSCPD_CON.EN = 1: Operating Mode

SSCPD_CON

Control Register **Reset Value: 0000_H**



Field	Bits	Type	Description
BC	3:0	rh	Bit Count Field 0001 to Shift counter is updated with every shifted bit. 1111 ***Attention: Do not write to this bit field.***
TE	8	rwh	Transmit Error Flag 0 No error 1 Transfer starts with the slave's transmit buffer not being updated or a transmit buffer overflow

Field	Bits	Type	Description
RE	9	rwh	Receive Error Flag 0 No error 1 Reception completed before the receive buffer was read
PE	10	rwh	Phase Error Flag 0 No error 1 Received data changes around sampling clock edge
BE	11	rwh	Baudrate Error Flag 0 No error 1 More than factor 2 or 0.5 between slave's actual and expected baudrate
BSY	12	rh	Busy Flag Set while a transfer is in progress. ***Attention: Do not write to this bit.***
MS	14	rw	Master Select Bit 0 Slave Mode. Operate on shift clock received via SCLK. 1 Master Mode. Generate shift clock and output it via SCLK.
EN	15	rw	Enable Bit = 1 Transmission and reception enabled. Access to status flags and M/S control
RESERVED	7:4, 13	r	Reserved for future use; these bits must be left at their reset values.

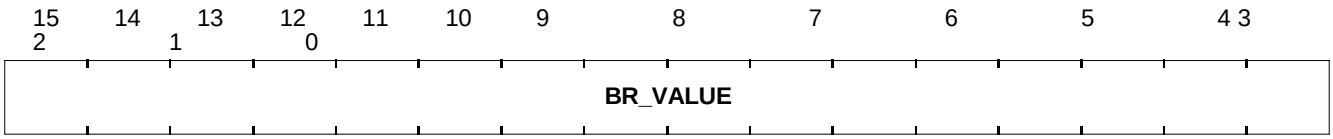
Note: The target of an access to **SSCPD_CON** *(control bits or flags) is determined by the state of* **SSCPD_CON.*EN*** *prior to the access; that is, writing $C057_H$ to* **SSCPD_CON** *in programming mode (**EN** = 0) initializes the SSC (**EN** was 0) and then turn it on (**EN** = 1). When writing to* **SSCPD_CON**, *ensure that reserved locations receive zeros.*

9.2.3.1.3 Baudrate Timer Reload Register

The SSC baudrate timer reload register **SSCPD_BR** contains the 16-bit reload value for the baudrate timer.

SSCPD_BR

Baudrate Timer Reload Register **Reset Value: 0000_H**



Field	Bits	Type	Description
BR_VALUE	15:0	rw	Baudrate Timer/Reload Register Value Reading **SSCPD_BR** returns the 16-bit content of the baudrate timer. Writing **SSCPD_BR** loads the baudrate timer reload register with **BR_VALUE** (write is only possible when **SSCPD_CON.EN** = 0).

9.2.3.1.4 Transmitter Buffer Register

The SSC transmitter buffer register **SSCPD_TB** contains the transmit data value.

SSCPD_TB

Transmitter Buffer Register **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TB_VALUE															

Field	Bits	Type	Description
TB_VALUE	15:0	rw	Transmit Data Register Value **TB_VALUE** is the data value to be transmitted. Unselected bits of **SSCPD_TB** are ignored during transmission.

9.2.3.1.5 Receiver Buffer Register

The SSC receiver buffer register **SSCPD_RB** contains the receive data value.

SSCPD_RB

Receiver Buffer Register **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB_VALUE															

Field	Bits	Type	Description
RB_VALUE	15:0	rh	Receive Data Register Value **SSCPD_RB** contains the received data value **RB_VALUE**. Unselected bits of **SSCPD_RB** will be not valid and should be ignored.

9.2.3.1.6 Receive FIFO Control Register

SSCPD_RXFCON

Receive FIFO Control Register **Reset value: 0100_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			RXFITL					RESERVED					RXTMEN	RXFFLU	RXFEN

Field	Bits	Type	Description
RXFEN	0	rw	Receive FIFO Enable 0 Receive FIFO is disabled 1 Receive FIFO is enabled *Note: Resetting **RXFEN** automatically flushes the receive FIFO.*
RXFLU	1	rw	Receive FIFO Flush 0 No operation 1 Receive FIFO is flushed *Note: Setting **RXFFLU** clears bitfield **SSCPD_FSTAT.RXFFL**. **RXFFLU** is always read as 0.*

Field	Bits	Type	Description
RXTMEN	2	rw	Receive FIFO Transparent Mode Enable 0 Receive FIFO Transparent Mode is disabled 1 Receive FIFO Transparent Mode is enabled *Note: This bit is don't care if the receive FIFO is disabled (**RXFEN** = 0).*
RXFITL	13:8	rw	Receive FIFO Interrupt Trigger Level Defines a receive FIFO interrupt trigger level. A receive interrupt request (RIR) is always generated after the reception of a byte when the filling level of the receive FIFO is equal to or greater **RXFITL** 000000Reserved. Do not use this combination 000001Interrupt trigger level is set to one 000010Interrupt trigger level is set to two ... 100000Interrupt trigger level is set to thirty two 100001Reserved Do not use this combination. ... Reserved Do not use this combination. 111111Reserved Do not use this combination. *Note: In Transparent Mode this bitfield is don't care.* *Note: Combinations defining a interrupt trigger level greater then the configured FIFO size should not be used* ***Attention: 32 Bytes is the maximum authorized.***
RESERVED	7:3, 15:14	r	Reserved; these bits must be left at their reset values.

9.2.3.1.7 Transmit FIFO Control Register

SSCPD_TXFCON

Transmit FIFO Control Register **Reset value: 0100_H**

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
RESERVED	**TXFITL**	**RESERVED**	**TX TMEN** / **TXFFLU** / **TXFEN**

Field	Bits	Type	Description
TXFEN	0	rw	Transmit FIFO Enable 0 Transmit FIFO is disabled 1 Transmit FIFO is enabled *Note: Resetting **TXFEN** automatically flushes the transmit FIFO.*
TXFLU	1	rw	Transmit FIFO Flush 0 No operation 1 Transmit FIFO is flushed *Note: Setting **TXFFLU** clears bitfield **TXFFL** in register **SSCPD_FSTAT.TXFFLU** is always read as 0.*

Field	Bits	Type	Description
TXTMEN	2	rw	Transmit FIFO Transparent Mode Enable 0 Transmit FIFO Transparent Mode is disabled 1 Transmit FIFO Transparent Mode is enabled *Note: This bit is don't care if the receive FIFO is disabled (**TXFEN** = 0).*
TXFITL	13:8	rw	Transmit FIFO Interrupt Trigger Level Defines a transmit FIFO interrupt trigger level. A transmit interrupt request (TIR) is always generated after the transfer of a byte when the filling level of the transmit FIFO is equal to or less than **TXFITL.** 000000Reserved. Do not use this combination. 000001Interrupt trigger level is set to one 000010Interrupt trigger level is set to two ... 100000Interrupt trigger level is set to thirty two 100001Reserved Do not use this combination. ... Reserved Do not use this combination. 111111Reserved Do not use this combination. *Note: In Transparent Mode this bitfield is don't care.* *Note: Combinations defining an interrupt trigger level greater then the configured FIFO size should not be used.* ***Attention: 32 bytes is the maximum authorized.***
RESERVED	7:3, 15:14	r	Reserved; these bits must be left at their reset values.

9.2.3.1.8 FIFO Status Register

SSCPD_FSTAT

FIFO Status Register **Reset value: 0000_H**

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RESERVED | TXFFL | RESERVED | RXFFL

Field	Bits	Type	Description
RXFFL	5:0	rh	**Receive FIFO Filling Level** 000000Receive FIFO is filled with zero bytes 000001Receive FIFO is filled with one byte ... 100000Receive FIFO is filled with thirty two bytes 111110Reserved. Reserved Do not use this combination. 111111Reserved. Reserved Do not use this combination. *Note: **RXFFL** is cleared after a receive FIFO flush operation.* *Note: Combinations defining a interrupt trigger level greater then the configured FIFO size should not be used.* ***Attention: 32 bytes is the maximum read.***

Field	Bits	Type	Description
TXFFL	13:8	rh	**Transmit FIFO Filling Level** 000000Receive FIFO is filled with zero bytes 000001Receive FIFO is filled with one byte ... 100000Receive FIFO is filled with thirty two bytes 111110Reserved. Reserved Do not use this combination. 111111Reserved. Reserved Do not use this combination. *Note: **TXFFL** is cleared after a receive FIFO flush operation.* *Note: Combinations defining a interrupt trigger level greater then the configured FIFO size should not be used.* ***Attention: 32 bytes is the maximum transmitted.***
RESERVED	7:6, 15:14	r	Reserved; these bits must be left at their reset values.

9.2.4 Interrupts

For a detailed description of the various interrupts refer to **Section 9.2.2 General Operation (on Page 285)**. An overview is given in **Table 62**

Table 62 SSC Interrupt Sources

Interrupt	Signal	Description
Transmission starts (master only)	TIR	Indicates that the transmit buffer can be reloaded with new data. If a FIFO is configured for the SSC and **SSCPD_TXFCON.TXTMEN** is cleared **SSCPD_TXFCON.TXFIFL** defines when the interrupt is generated depending on the FIFO filling state (when Trigger level set, TIR occurs when FIFO = or <).
Transmission ends (master only)	RIR	The configured number of bits have been transmitted and shifted to the receive buffer. If a FIFO is configured for the SSC and **SSCPD_RXFCON.RXTMEN** is cleared **SSCPD_RXFCON.RXFIFL** defines when the interrupt is generated depending on the FIFO filling state (when Trigger level set, RIR occurs when FIFO = or >).
Transparent Read Operation	RIR	In Transparent Mode a receive interrupt is always generated on a read operation from the MCU to the receive FIFO if the FIFO is not empty after this operation.
Receive Overflow	EIR	If an additional frame is received when the FIFO is completely full, an overflow error occurred. The interrupt is generated and the previously received frame is overwritten and therefore lost in the FIFO.
Transmit Overflow	EIR	If an additional frame is written to the transmit FIFO when it is completely full an overflow error occurred. The interrupt is generated and the previously written frame is overwritten and therefore lost in the FIFO.
Read to empty FIFO	RIR	A read operation from the MCU to an empty receive FIFO generated this interrupt.
Transparent first receive	RIR	In Transparent Mode the first received message in an empty receive FIFO generates this interrupt.
Flush Action	TIR	If **SSCPD_TXFCON.TXTMEN** is set, a transmit interrupt is generated when the transmit FIFO is flushed.
FIFO Enable	TIR	A transmit interrupt is generated when the transmit FIFO (**SSCPD_TXFCON.TXTMEN**) is enabled and **SSCPD_TXFCON.TXFEN** is set from 0 to 1.

Table 62 SSC Interrupt Sources

Interrupt	Signal	Description
Receive Error	EIR	This interrupt occurs if a new data frame is completely received and the last data in the receive buffer wasn't read yet. If a FIFO is configured for the SSC and **SSCPD_RXFCON.RXTMEN** is cleared **SSCPD_RXFCON.**RXFIFL defines when the interrupt is generated depending on the FIFO filling state. If a FIFO is configured for the SSC and **SSCPD_RXFCON.RXTMEN** is set the non FIFO interrupt generation in this case happened.
Phase Error	EIR	This interrupt is generated if the incoming data changes between one cycle before and two cycles after the latching edge of the shift clock signal SCLK.
Baudrate Error	EIR	This interrupt is generated when the incoming clock signal deviates from the programmed baudrate by more than 100%.
Transmit Error (Slave Mode only)	EIR	This interrupt is generated when **SSCPD_TB** was not updated since the last transfer if a transfer is initiated by a master.

Note: The interrupt request signals are connected to the central interrupt control unit; the control registers are also located there (XXXIC).

9.3 ASC0

System Integration

- Supply domain: VDD_LD1
- Chip internal interfaces:
  - Clock domain: Refer to **Section 7.2.1.3 Sub-System Clocks and Enables (on Page 67)** and see **Figure 18 Clock Enable (on Page 68)**.
  - Bus domain: PD-Bus
- Interrupt sources:
- Monitor Pins: Refer to **Section 9.7.10 Internal Signal Monitoring (on Page 435)**.

Attention: The ASC0 is connected to the following pins:

- RXD
- TXD
- ASC_RTS_n
- ASC_CTS_n

9.3.1 ASC0 Description

ASC0 is the acronym for Asynchronous Serial Communication interface. The ASC0 Subsystem consists of two blocks: one ASC0 controller itself and the AUTOSTART block. The ASC0 controller is a standard µC peripheral. The ASC0 subsystem also includes an echo mode (RxD0 -> TxD0) which retransmits received data. The autostart block is watching for an edge at the RxD0 data line to wake-up the required circuitry. The AUTOSTART should only be used in the asynchronous operating mode.

9.3.2 Features of the ASC0 Controller

- Duplex asynchronous operating modes:
  - 8- or 9-bit data frames, LSB first
  - Parity bit generation/checking
  - One or two stop bits
  - Baudrate from 3.27 MBaud to 0.77 Baud (with a 52MHz module clock f_{hw_clk},for information about hw_clk, refer to **Table 74 ASC0 Register Summary (on Page 334)**)
- Multiprocessor Mode for automatic address/data byte detection
- Loopback capability
- Support for IrDA data transmission up to 115.2 KBaud maximum
- Half-duplex 8-bit synchronous operating mode:
  - Baudrate from 6.5 MBaud to 662.5 Baud (with a 52 MHz module clock f_{hw_clk})
- Double buffered transmitter/receiver

- Interrupt generation:
  - On a transmitter buffer empty condition
  - On a transmit last bit of a frame condition
  - On a receiver buffer full condition
  - On an error condition (frame, parity, overrun error)
- Autobaud detection unit for asynchronous operating modes:
  - Detection of standard baudrates
    1200, 2400, 4800, 9600, 19 200, 38 400, 57 600, 115 200, and 230 400 Baud
  - Detection of non-standard baudrates
  - Detection of Asynchronous Modes
  - 7 bit, even parity; 7 bit, odd parity;
    8 bit, even parity; 8 bit, odd parity; 8 bit, no parity
  - Automatic initialization of control bits and baudrate generator after detection
  - Detection of a serial two-byte ASCII character frame
- FIFOs:
  - 8-stage receive FIFO (RXFIFO)
  - 8-stage transmit FIFO (TXFIFO)
  - Independent control of RXFIFO and TXFIFO
  - 9-Bit FIFO data width
  - Programmable Receive/Transmit Interrupt Trigger Level
  - Receive and transmit FIFO filling level indication
  - Overrun error generation
  - Underflow error generation
- HW Flow Control:
  - RTS/CTS handshaking
  - CTS interrupt
  - Programmable RTS de-assertion RXFIFO Trigger Level
- RX timeout

9.3.3 Functional Description of the ASC0

Figure 111 shows all functional relevant interfaces associated with the ASC0 Kernel.

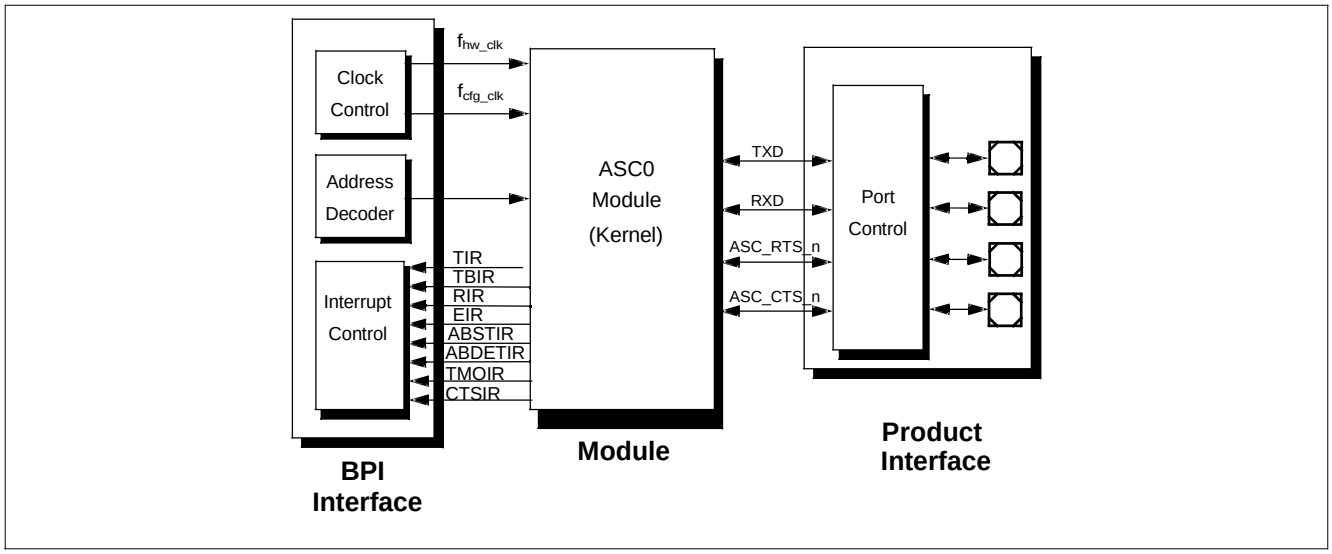


Figure 111 ASC0 Interface Diagram

9.3.4 Operational Overview

Figure 112 shows a block diagram of the ASC0 with its operating modes (Asynchronous and Synchronous Mode).

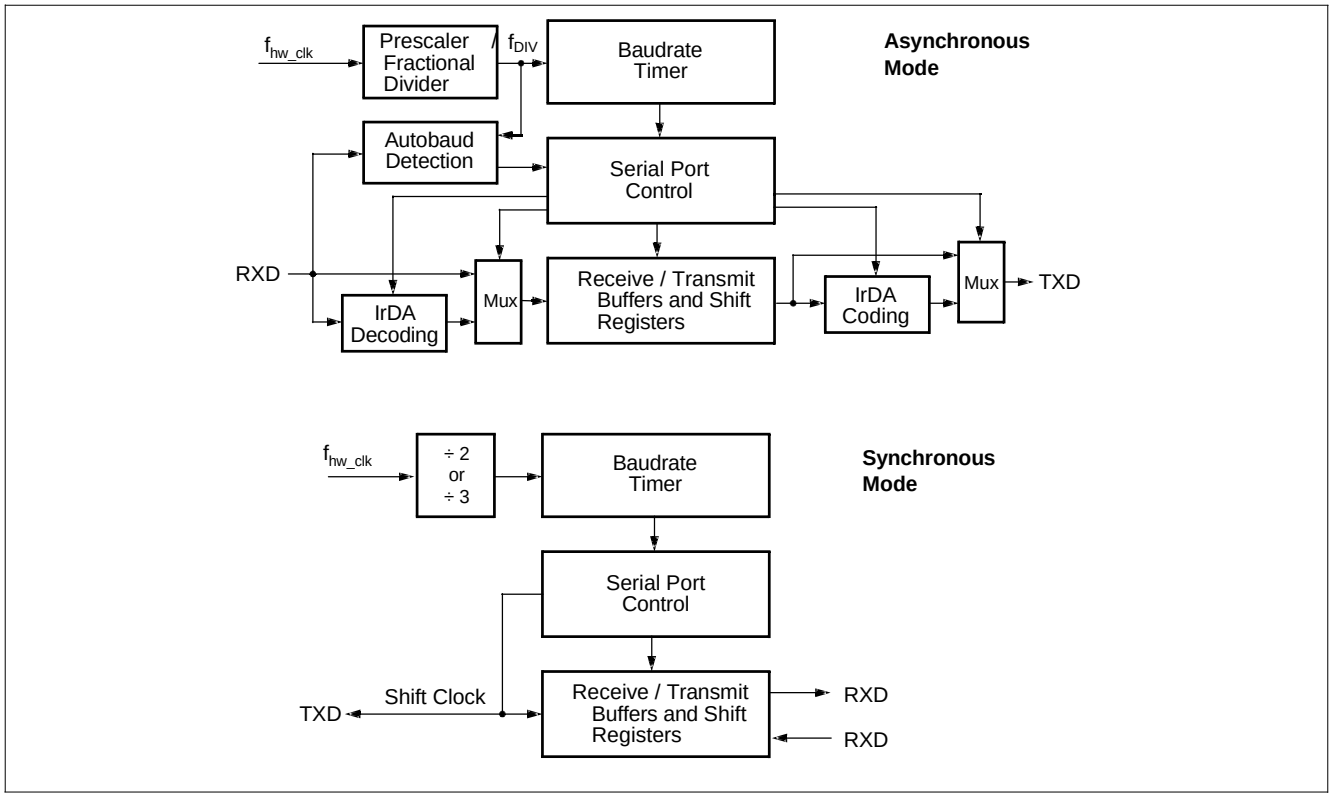


Figure 112 Block Diagram of the ASC0

9.3.5 General Operation

In Synchronous Mode, data are transmitted or received synchronous to a shift clock that is generated by the microcontroller. In Asynchronous Mode, either 8- or 9-bit data transfer, parity generation, and the number of stop bits can be selected. Parity, framing, and overrun error detection is provided to increase the reliability of data transfers. Transmission and reception of data is double-buffered. For multiprocessor communication, a mechanism is provided to distinguish address bytes from data bytes. Testing is supported by a loop-back option. A 13-bit baudrate timer with a versatile input clock divider circuitry provides the serial clock signal. In a Special Asynchronous Mode, the ASC0 supports IrDA data transmission up to 115.2 KBaud with fixed or programmable IrDA pulse width. Autobaud Detection allows to detect asynchronous data frames with its baudrate and mode with automatic initialization of the baudrate generator and the mode control bits.

A transmission is started by writing to the transmit buffer register **S0TBUF**. The selected operating mode determines the number of data bits that will actually be transmitted, so that, bits written to positions 9 through 15 of register **S0TBUF** are always insignificant. Data transmission is double-buffered, so a new character may be written to the transmit buffer register before the transmission of the previous character is complete. This allows the transmission of characters back-to-back without gaps.

Data reception is enabled by the Receiver Enable Bit **S0CON.REN**. After reception of a character has been completed, the received data can be read from the (read-only) receive buffer register **S0RBUF**; the received parity bit can also be read if provided by the selected operating mode. Bits in the upper half of **S0RBUF** that are not valid in the selected operating mode will be read as zeros.
Data reception is double-buffered, so that reception of a second character may already begin before the previously received character has been read out of the receive buffer register. In all modes, receive overrun error detection can be selected through bit **S0CON.OEN**. When enabled, the overrun error status flag **S0CON.OE** and the error

interrupt request line EIR will be activated when the receive buffer register has not been read by the time reception of a ninth character is complete. The previously received character in the receive buffer is overwritten.

The Loopback Mode (selected by bit **S0CON.LB**) allows the data currently being transmitted to be received simultaneously in the receive buffer. This may be used to test serial communication routines at an early stage without having to provide an external network.

Notes

1. *In Loopback Mode, the alternate input/output functions of the associated port pins are not necessary.*
2. *Serial data transmission or reception is only possible when the Baudrate Generator Run bit **S0CON.R** is set. Otherwise, the serial interface is idle.*
3. *Do not program the Mode Control bit field **S0CON.M** to one of the reserved combinations to avoid unpredictable behavior of the serial interface.*

The operating mode of the serial channel ASC0 is controlled by its control register **S0CON**. This register contains control bits for mode and error check selection, and status flags for error identification.

9.3.5.1 Asynchronous Operation

Asynchronous Mode supports full-duplex communication in which both transmitter and receiver use the same data frame format and the same baudrate. Data is transmitted on line TXD and received on line RXD. IrDA data transmission/reception is supported up to 115.2 KBit/s. **Figure 113** shows the block diagram of the ASC0 when operating in Asynchronous Mode.

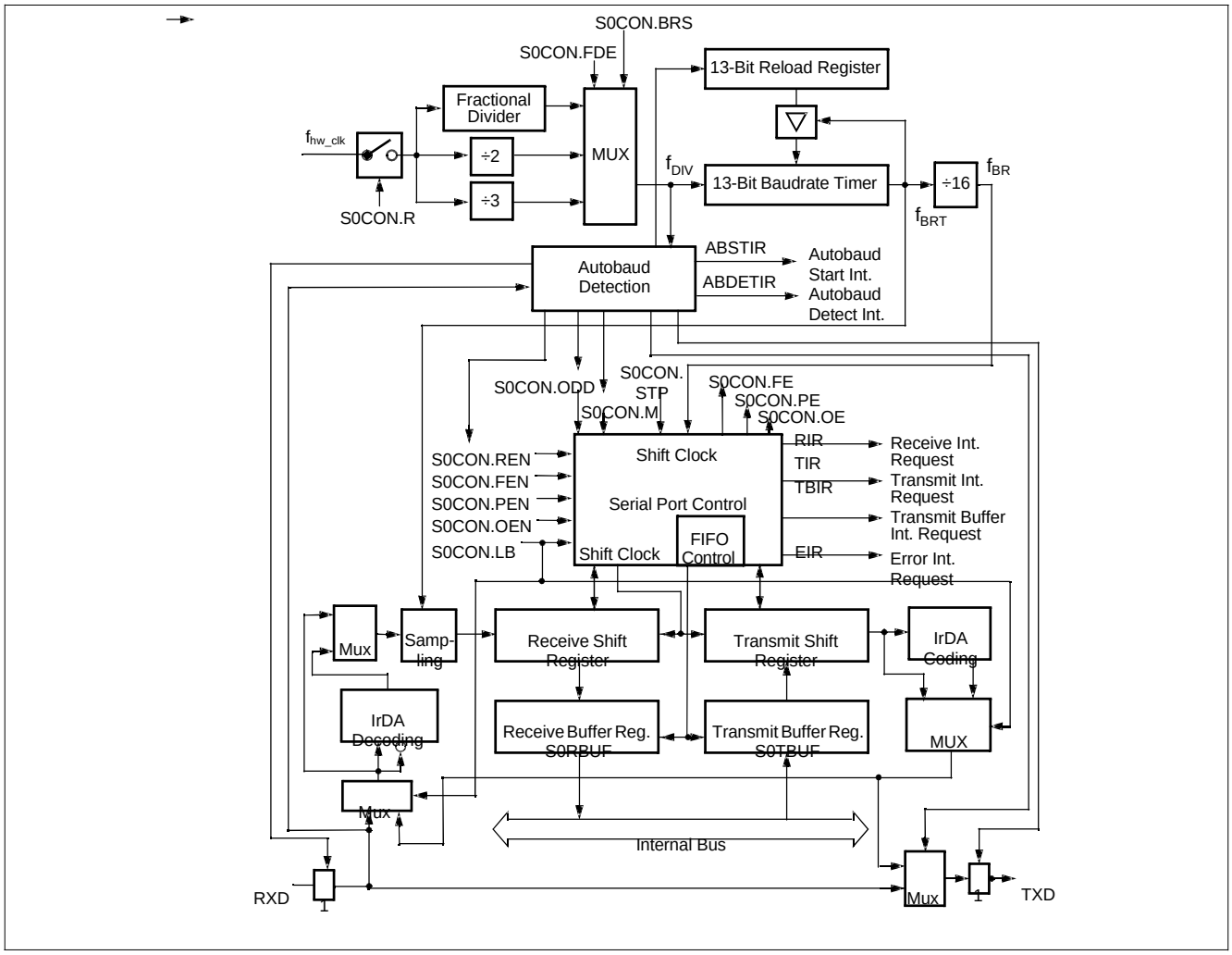


Figure 113 Asynchronous Mode of Serial Channel ASC0

9.3.5.2 Asynchronous Data Frames

8-Bit Data Frames

8-bit data frames consist of either eight data bits D7...D0 (**S0CON.M** = 001_B), or seven data bits D6...D0 plus an automatically generated parity bit (**S0CON.M** = 011_B). Parity may be odd or even, depending on bit **S0CON.ODD**. An even parity bit will be set if the modulo-2-sum of the 7 data bits is 1. An odd parity bit will be cleared in this case. Parity checking is enabled via bit **S0CON.PEN** (always OFF in 8-bit data mode). The parity error flag **S0CON.PE** will be set, along with the error interrupt request flag, if a wrong parity bit is received. The parity bit itself will be stored in bit **S0RBUF.**7.

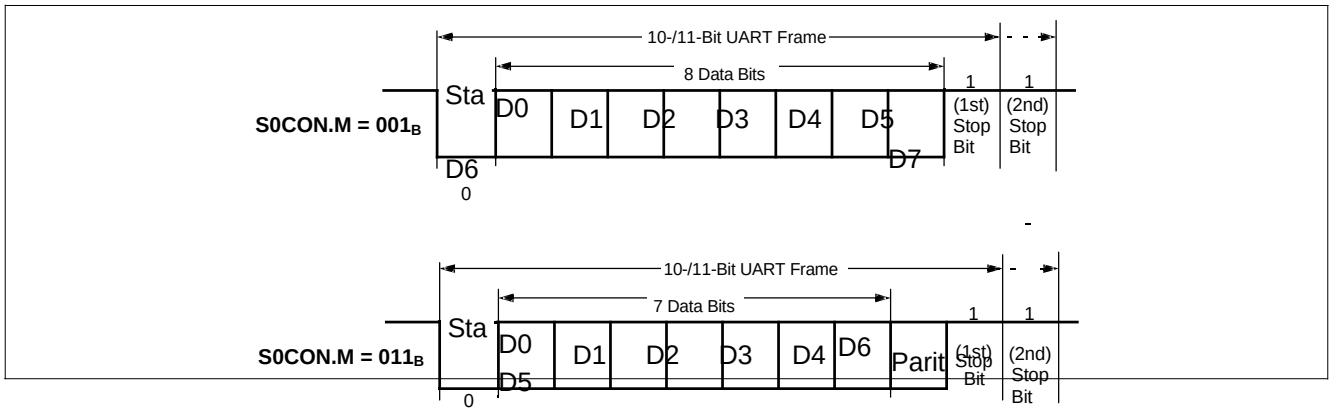


Figure 114 Asynchronous 8-Bit Frames

9-Bit Data Frames

9-bit data frames consist of either nine data bits D8...D0 (**S0CON.M** = 001_B), eight data bits D7...D0 plus an automatically generated parity bit (**S0CON.M** = 111_B), or eight data bits D7...D0 plus wake-up bit (**S0CON.M** = 101_B). Parity may be odd or even, depending on bit **S0CON.ODD**. An even parity bit will be set if the modulo-2-sum of the 8 data bits is 1. An odd parity bit will be cleared in this case. Parity checking is enabled via bit **S0CON.PEN** (always OFF in 9-bit data and wake-up mode). The parity error flag **S0CON.PE** will be set, along with the error interrupt request flag, if a wrong parity bit is received. The parity bit itself will be stored in bit **S0RBUF.**8.

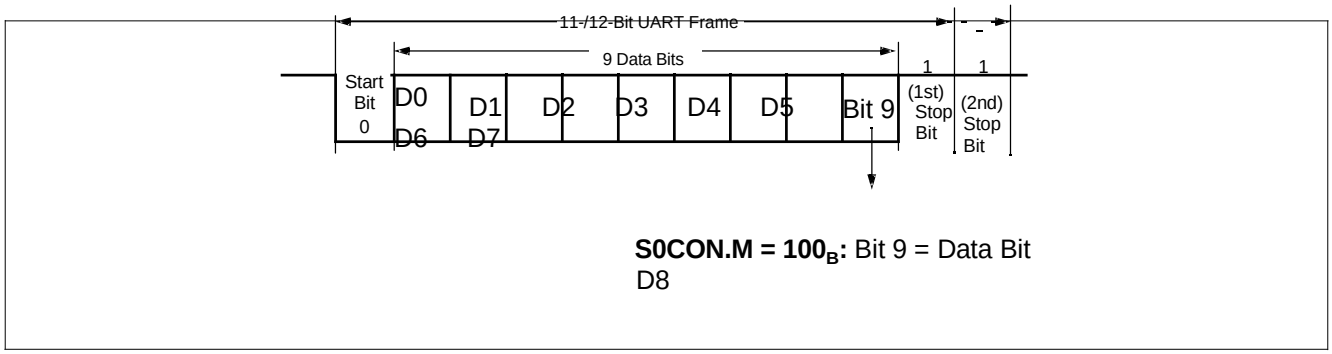


Figure 115 Asynchronous 9-Bit Frames

In wake-up mode, received frames are transferred to the receive buffer register only if the 9th bit (the wake-up bit) is 1. If this bit is 0, no receive interrupt request will be activated and no data will be transferred.
This feature may be used to control communication in a multi-processor system:
When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte to identify the target slave. An address byte differs from a data byte in that the additional 9th bit is a 1 for an address byte, but is a 0 for a data byte; so, no slave will be interrupted by a data 'byte'. An address 'byte' will interrupt all slaves (operating in 8-bit data + wake-up bit mode), so each slave can examine the eight LSBs of the received character (the address). The addressed slave will switch to 9-bit data mode (such as by clearing bit **S0CON.M**[0]), to enable it to also receive the data bytes that will be coming (having the wake-up bit cleared). The slaves not being addressed remain in 8-bit data + wake-up bit mode, ignoring the following data bytes.

IrDA Frames

The modulation schemes of IrDA are based on standard asynchronous data transmission frames. The asynchronous data format in IrDA Mode (**S0CON.M** = 010_B) is defined as follows:

1 start bit/8 data bits/1 stop bit

The coding/decoding of/to the asynchronous data frames is shown in **Figure 116**. In general, during IrDA transmissions, UART frames are encoded into IR frames and vice versa. A low level on the IR frame indicates an “LED off” state. A high level on the IR frame indicates an “LED on” state.

For a 0 bit in the UART frame, a high pulse is generated. For a 1 bit in the UART frame, no pulse is generated. The high pulse starts in the middle of a bit cell and has a fixed width of 3/16 of the bit time. The ASC0 also allows the length of the IrDA high pulse to be programmed. Further, the polarity of the received IrDA pulse can be inverted in IrDA Mode. **Figure 116** shows the non-inverted IrDA pulse scheme.

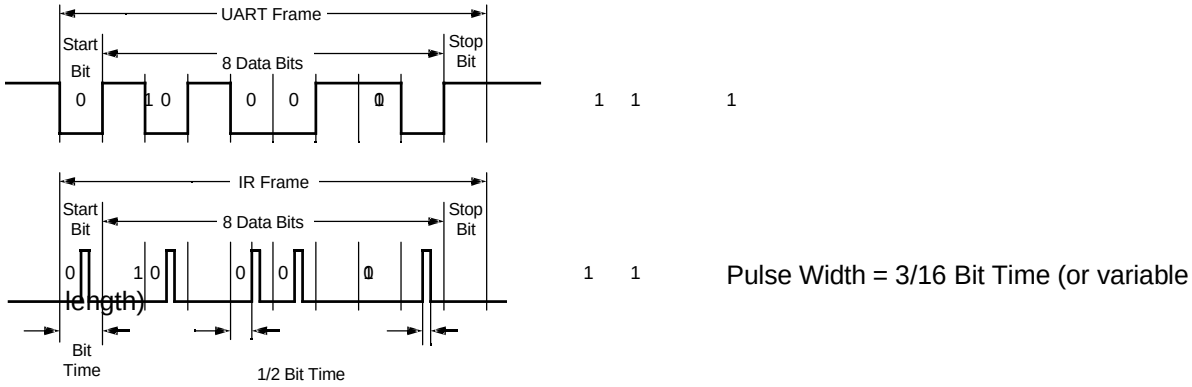


Figure 116 IrDA Frame Encoding/Decoding

The ASC0 IrDA pulse mode/width register PMW contains the 8-bit IrDA pulse width value and the IrDA pulse width mode select bit. This register is required in the IrDA operating mode only.

9.3.5.3 Asynchronous Transmission

Asynchronous transmission begins at the next overflow of the divide-by-16 baudrate timer (transition of the baudrate clock f_{BR}), if bit **S0CON.R** is set and data has been loaded into **S0TBUF**. The transmitted data frame consists of three basic elements:

- Start bit
- Data field (eight or nine bits, LSB first, including a parity bit, if selected)
- Delimiter (one or two stop bits)

Data transmission is double-buffered. When the transmitter is idle, the transmit data loaded in the transmit buffer register is immediately moved to the transmit shift register, thus freeing the transmit buffer for the next data to be sent. This is indicated by the transmit buffer interrupt request line TBIR being activated. **S0TBUF** may now be loaded with the next data, while transmission of the previous data continues.

The transmit interrupt request line TIR will be activated before the last bit of a frame is transmitted, that is, before the first or the second stop bit is shifted out of the transmit shift register.

Note: The transmitter output pin TXD must be configured for alternate data output.

9.3.5.4 Transmit FIFO Operation

The transmit FIFO (TXFIFO) provides the following functionality:

- Enable/disable control
- Programmable filling level for transmit interrupt generation
- Filling level indication
- FIFO clear (flush) operation
- FIFO overflow error generation.

The 8 stage transmit FIFO is controlled by the **S0TXFCON** control register. When bit **S0TXFCON.TXFEN** is set, the transmit FIFO is enabled. The interrupt trigger level defined by **S0TXFCON.TXFITL** defines the filling level of the TXFIFO at which a transmit buffer interrupt TBIR or a transmit interrupt TIR is generated. These interrupts are always generated when the filling level of the transmit FIFO is equal to or less than the value stored in **S0TXFCON.TXFITL**.

Bit field **S0FCSTAT.TXFFL** indicates the number of entries that are actually written (valid) in the TXFIFO. Therefore, the software can verify, for example, in the interrupt service routine, how many bytes can be still written into the transmit FIFO via register **S0TBUF** without getting an overrun error.

The transmit FIFO cannot be accessed directly. All data write operations into the TXFIFO are executed by writing into the **S0TBUF** register.

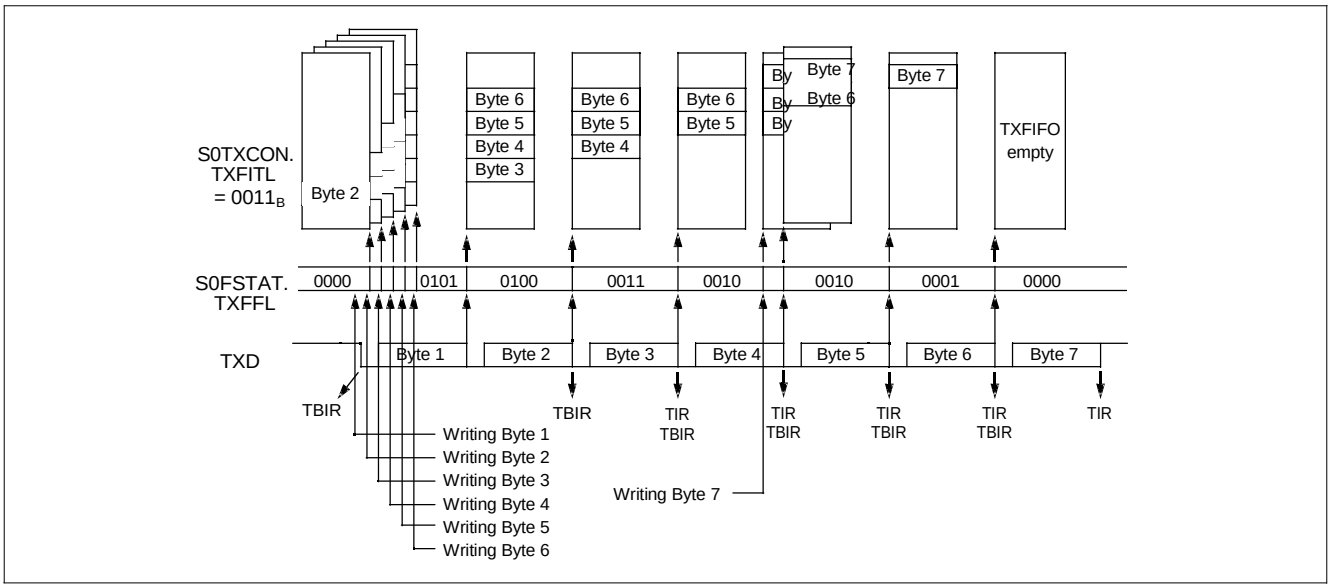


Figure 117 Transmit FIFO Operation Example

The example in **Figure 117** shows a typical 8 stage transmit FIFO operation. In this example seven bytes are transmitted via the TXD output line. The transmit FIFO interrupt trigger level **S0TXFCON.TXFITL** is set to 000011_B. The first byte written into the empty TXFIFO via **S0TBUF** is directly transferred into the transmit shift register and is not written into the FIFO. A transmit buffer interrupt will be generated in this case. After byte 1, bytes 2 to 6 are written into the transmit FIFO.

After the transfer of byte 3 from the TXFIFO into the transmit shift register of the ASC0, 3 bytes remain in the TXFIFO. Therefore, the value of **S0TXFCON.TXFITL** is reached and a transmit buffer interrupt will be generated at the beginning and a transmit interrupt at the end of the byte 3 serial transmission. During the serial transmission of byte 4, another byte (byte 7) is written into the TXFIFO (**S0TBUF** write operation). Finally, after the start of the serial transmission of byte 7, the TXFIFO is again empty.

If the TXFIFO is full and additional bytes are written into TBUF, the error interrupt will be generated with bit **S0CON.OE** set. In this case, the data byte that was last written into the transmit FIFO is overwritten and the transmit FIFO filling level **S0FCSTAT.TXFFL** is set to maximum.

The TXFIFO can be flushed or cleared by setting bit **S0TXFCON.TXFFLU.** After this TXFIFO flush operation, the TXFIFO is empty and the transmit FIFO filling level **S0FCSTAT.TXFFL** is set to 000000_B. A running serial transmission is not aborted by a receive FIFO flush operation

*Note: The TXFIFO is flushed automatically with a reset operation of the ASC0 module and if the TXFIFO becomes disabled (resetting bit **S0TXFCON.TXFEN**) after it was previously enabled.*

9.3.5.5 Asynchronous Reception

Asynchronous reception is initiated by a falling edge (1-to-0 transition) on line RXD, provided that bits **S0CON.R** and **S0CON.REN** are set. The receive data input line RXD is sampled at 16 times the rate of the selected baudrate. A majority decision of the 7th, 8th, and 9th sample determines the effective bit value. This avoids erroneous results that may be caused by noise.

If the detected value is not a 0 when the start bit is sampled, the receive circuit is reset and waits for the next 1-to-0 transition at line RXD. If the start bit proves valid, the receive circuit continues sampling and shifts the incoming data frame into the receive shift register.

When the last stop bit has been received, the content of the receive shift register are transferred to the receive data buffer register **S0RBUF**. Simultaneously, the receive interrupt request line RIR is activated after the 9th sample in the last stop bit time slot (as programmed), regardless of whether valid stop bits have been received or not. The receive circuit then waits for the next start bit (1-to-0 transition) at the receive data input line.

Note: The receiver input pin RXD must be configured for input.

Asynchronous reception is stopped by clearing bit **S0CON.REN**. A currently received frame is completed including the generation of the receive interrupt request and an error interrupt request, if appropriate. Start bits that follow this frame will not be recognized.

Note: In wake-up mode, received frames are transferred to the receive buffer register only if the 9th bit (the wake-up bit) is 1. If this bit is 0, no receive interrupt request will be activated and no data will be transferred.

9.3.5.6 Receive FIFO Operation

The receive FIFO (RXFIFO) provides the following functionality:

- Enable/disable control
- Programmable filling level for receive interrupt generation
- Filling level indication
- FIFO clear (flush) operation
- FIFO overflow error generation

The 8 stage receive FIFO is controlled by the **S0RXFCON** control register. When bit **S0RXFCON.RXFEN** is set, the receive FIFO is enabled. The interrupt trigger level defined by **S0RXFCON.RXFITL** defines the filling level of RXFIFO at which a receive interrupt RIR is generated. RIR is always generated when the filling level of the receive FIFO is equal to or greater than the value stored in **S0RXFCON.RXFITL**.

Bit field **S0FSTAT.RXFFL** in the FIFO status register FSTAT indicates the number of bytes that have been actually written into the FIFO and can be read out of the FIFO by a user program.

The receive FIFO cannot be accessed directly. All data read operations from the RXFIFO are executed by reading the **S0RBUF** register.

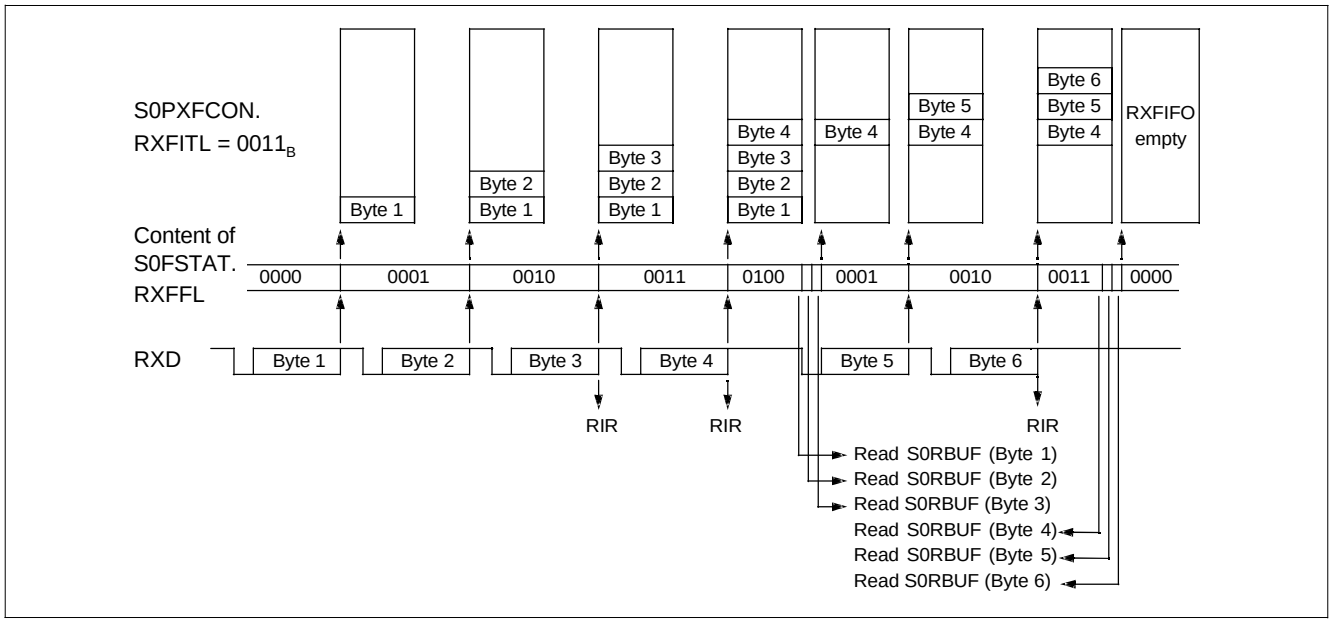


Figure 118 Receive FIFO Operation Example

The example in **Figure 118** shows a typical 8 stage receive FIFO operation. In this example, six bytes are received via the RXD input line. The receive FIFO interrupt trigger level **S0RXFCON.RXFITL** is set to 000011_B. Therefore, the first receive interrupt RIR is generated after the reception of byte 3 (RXFIFO is filled with three bytes).

After the reception of byte 4, three bytes are read out of the receive FIFO. After this read operation, the RXFIFO still contains one byte. RIR becomes again active after two more bytes (byte 5 and 6) have been received (RXFIFO filled again with 3 bytes). Finally, the FIFO is cleared after three read operation.

If the RXFIFO is full and additional bytes are received, the receive interrupt RIR and the error interrupt EIR will be generated with bit **S0CON.OE** set. In this case, the data byte last written into the receive FIFO is overwritten. With the overrun condition, the receive FIFO filling level **S0FSTAT.RXFFL** is set to maximum. If a **S0RBUF** read operation is executed with the RXFIFO enabled but empty, an error interrupt EIR will be generated, and with bit **S0CON.OE** set. In this case, the receive FIFO filling level **S0FSTAT.RXFFL** is set to 000000_B.

If the RXFIFO is available but disabled (**S0RXFCON.RXFEN** = 0) and the receive operation is enabled (**S0CON.REN** = 1), the asynchronous receive operation is functionally equivalent to the asynchronous receive operation of the ASC0 module.

The RXFIFO can be flushed or cleared by setting bit **S0RXFCON.RXFFLU**. After this RXFIFO flush operation, the RXFIFO is empty and the receive FIFO filling level **S0FSTAT.RXFFL** is set to 000000_B.
The RXFIFO is flushed automatically with a reset operation of the ASC0 module and if the RXFIFO becomes disabled (resetting bit **S0RXFCON.RXFEN**) after it was previously enabled. Resetting bit **S0CON.REN** without resetting **S0RXFCON.RXFEN** does not affect (reset) the RXFIFO state. This means that the receive operation of the ASC0 is stopped, in this case, without changing the content of the RXFIFO. After setting **S0CON.REN** again, the RXFIFO with its content is again available.

Note: After a successful autobaud detection sequence (if implemented), the RXFIFO should be flushed before data is received.

9.3.5.7 FIFO Transparent Mode

In Transparent Mode, a specific interrupt generation mechanism is used for receive and transmit buffer interrupts. In general, in Transparent Mode, receive interrupts are always generated if data bytes are available in the RXFIFO. Transmit buffer interrupts are always generated if the TXFIFO is not full. The relevant conditions for interrupt generation in Transparent Mode are:

- FIFO filling levels
- Read or write operations on the **S0RBUF** or **S0TBUF** data register.

Interrupt generation for the receive FIFO depends on the RXFIFO filling level and the execution of read operations of register **S0RBUF** (see **Figure 119**). Transparent Mode for the RXFIFO is enabled when bits **S0RXFCON.RXTMEN** and **S0RXFCON.RXFEN** are set.

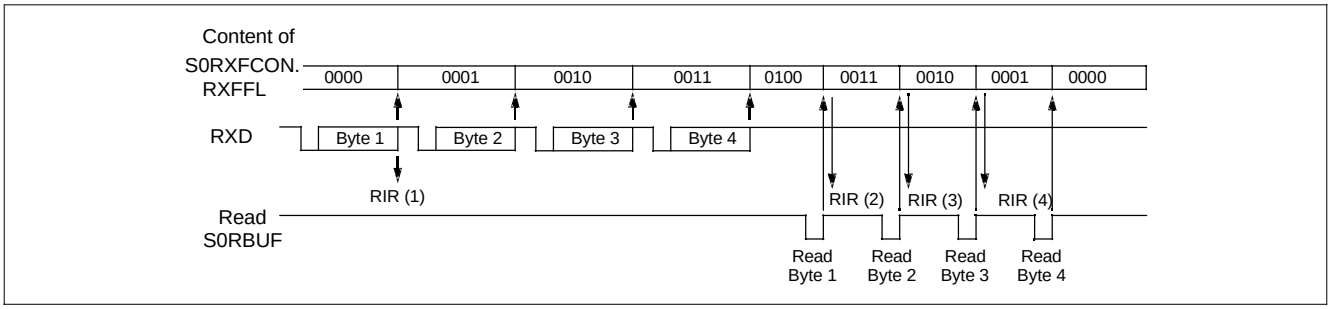


Figure 119 Transparent Mode Receive FIFO Operation

If the RXFIFO is empty, a receive interrupt RIR is always generated when the first byte is written into an empty RXFIFO (**S0FSTAT.RXFFL** changes from 000000_B to 000001_B). If the RXFIFO is filled with at least one byte, the occurrence of further receive interrupts depends on the read operations of register **S0RBUF**. The receive interrupt RIR will always be activated after a **S0RBUF** read operation if the RXFIFO still contains data (**S0FSTAT.RXFFL** is not equal to 000000_B). If the RXFIFO is empty after a **S0RBUF** read operation, no further receive interrupt will be generated.

If the RXFIFO is full (**S0FSTAT.RXFFL** = maximum) and additional bytes are received, an error interrupt EIR will be generated with bit **S0CON.OE** set. In this case, the data byte last written into the receive FIFO is overwritten. If a **S0RBUF** read operation is executed with the RXFIFO enabled but empty (underflow condition), an error interrupt EIR will be generated as well, with bit **S0CON.OE** set.

If the RXFIFO is flushed in Transparent Mode, the software must take care that a previous pending receive interrupt is ignored.

*Note: The Receive FIFO Interrupt Trigger Level bit field **S0RXFCON.RXFITL** is a don't care in Transparent Mode.*

Interrupt generation for the transmit FIFO depends on the TXFIFO filling level and the execution of write operations to the register **S0TBUF**. Transparent Mode for the TXFIFO is enabled when bits **S0TXFCON.TXTMEN** and **S0TXFCON.TXFEN** are set.

A transmit buffer interrupt TBIR is always generated when the TXFIFO is not full (**S0FSTAT.TXFFL** not equal to maximum) after a byte has been written into register **S0TBUF**. TBIR is also activated after a TXFIFO flush operation or when the TXFIFO becomes enabled (**S0TXFCON.TXTMEN** and **S0TXFCON.TXFEN** set) when it was previously disabled. In these cases, the TXFIFO is empty and ready to be filled with data.

If the TXFIFO is full (**S0FSTAT.TXFFL** = maximum) and an additional byte is written into **S0TBUF**, no further transmit buffer interrupt will be generated after the **S0TBUF** write operation. In this case the data byte last written into the transmit FIFO is overwritten and an overrun error interrupt (EIR) will be generated with bit **S0CON.OE** set.

*Note: The Transmit FIFO Interrupt Trigger Level bit field **S0TXFCON.TXFITL** is don't care in Transparent Mode.*

9.3.5.8 IrDA Mode

The duration of the IrDA pulse is normally 3/16 of a bit period. The IrDA standard also allows the pulse duration to be independent of the baudrate or bit period. In this case, the width of the transmitted pulse always corresponds to the 3/16 pulse width at 115.2 kBaud, which is 1.627 µs. Either fixed or bit-period-dependent IrDA pulse width generation can be selected. The IrDA pulse width mode is selected by bit **S0PWM.IRPW**.

In case of fixed IrDA pulse width generation, the lower eight bits in register PMW are used to adapt the IrDA pulse width to a fixed value such as 1.627 µs. The fixed IrDA pulse width is generated by a programmable timer as shown in **Figure 120**.

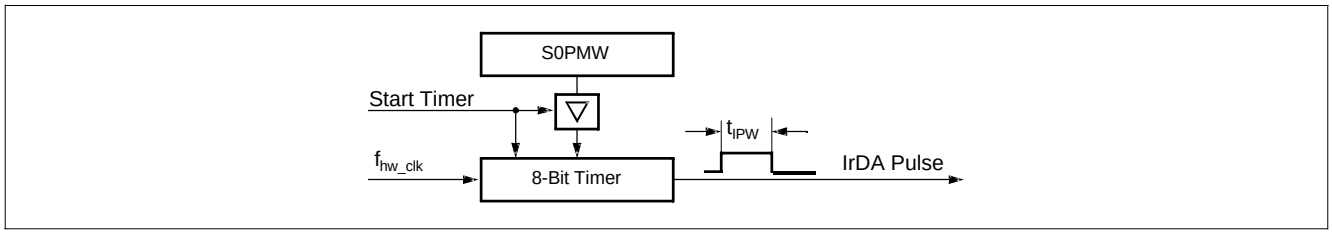


Figure 120 Fixed IrDA Pulse Generation

The IrDA pulse width can be calculated according the formulas given in **Table 63**.

Table 63 Formulas for IrDA Pulse Width Calculation

S0PWM.PW_VALUE		Formulas	
1 ... 255	0	$t_{IPW} = \frac{3}{16 \times Baudrate}$	$t_{IPW\ min} = \frac{(PMW >> 1)}{f_{hw\_clk}}$
	1	$t_{IPW} = \frac{PMW}{f_{hw\_clk}}$	

*Note: The name PMW in the formulas of **Table 63** represents the contents of the pulse width/mode register bits **S0PWM.PW_VALUE** taken as an unsigned 8-bit integer.*

The contents of **S0PWM.PW_VALUE** further define the minimum IrDA pulse width ($t_{IPW\ min}$) that is still recognized as a valid IrDA pulse during a receive operation. This function is independent of the selected IrDA pulse width mode (fixed or variable) which is defined by bit **S0PWM.IRPW**. The minimum IrDA pulse width is calculated by a shift right operation of ***S0PWM***[7:0] by one bit divided by the module clock f_{hw_clk}.

*Note: If **S0PWM.IRPW** is cleared (fixed IrDA pulse width), **S0PWM.PW_VALUE** must be a value which assures that $t_{IPW} > t_{IPW\ min}$.*

Table 64 gives three examples for typical frequencies of f_{hw_clk}.

Table 64 IrDA Pulse Width Adaption to 1.627 ∝s

f_{hw_clk}	PMW	t_{IPW}	Error	$t_{IPW\ min}$
13 MHz	21	1.615 ∝s	- 0.12 %	0.77 ∝s
26 MHz	42	1.615 ∝s	- 0.12 %	0.81 ∝s
52 MHz	84	1.615 ∝s	- 0.12 %	0.81 ∝s

9.3.5.9 RXD/TXD Data Path Selection in Asynchronous Modes

The data paths for the serial input and output data in Asynchronous Mode are affected by several control bits in the registers **S0CON** and **S0ABCON** as shown in **Figure 121**. The Synchronous Mode operation is not affected by these data path selection capabilities.

The input signal from RXD passes an inverter which is controlled by bit **S0ABCON.RXINV**. The output signal of this inverter is used for the Autobaud Detection and may bypass the logic in the Echo Mode (controlled by bit **S0ABCON.ABEM**). Further, two multiplexers are in the RXD input signal path for providing the Loopback Mode capability (controlled by bit **S0CON.LB**) and the IrDA receive pulse inversion capability (controlled by bit **S0CON.RXDI**).

Depending on the Asynchronous Mode (controlled by bit field **S0CON.M**), output signal or the RXD input signal in Echo Mode (controlled by bit **S0ABCON.ABEM**) is switched to the TXD output via an inverter (controlled by bit **S0ABCON.TXINV**).

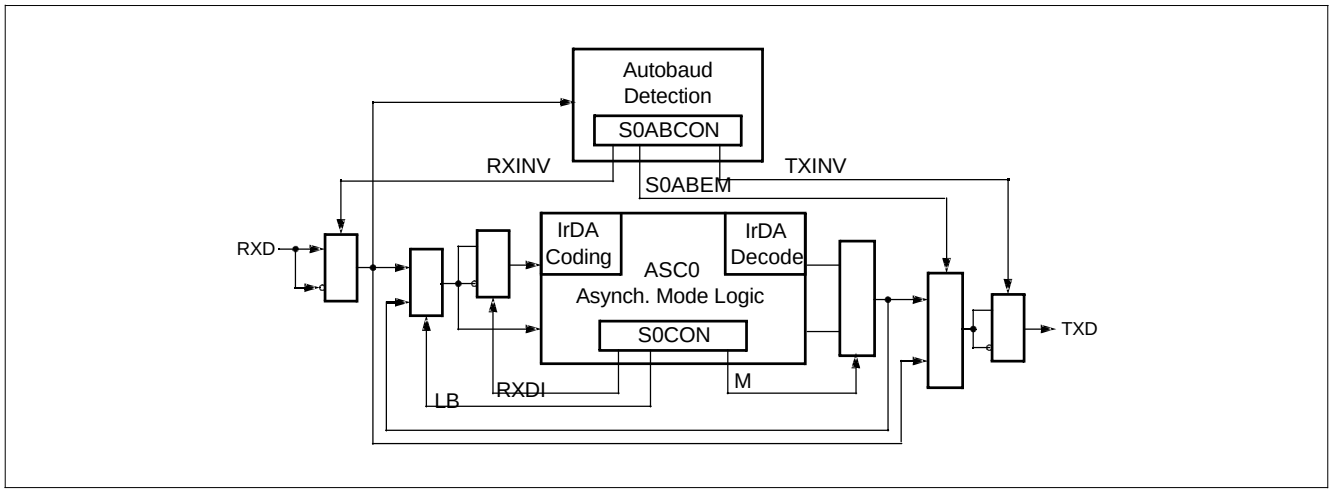


Figure 121 RXD/TXD Data Path in Asynchronous Modes

*Note: In Echo Mode the transmit output signal is blocked by the Echo Mode output multiplexer. **Figure 121** shows that it is not possible to use an IrDA coded receiver input signal for Autobaud Detection.*

9.3.5.10 Synchronous Operation

Synchronous Mode supports half-duplex communication, basically for simple I/O expansion via shift registers. Data is transmitted and received via line RXD while line TXD outputs the shift clock.

Synchronous Mode is selected with **S0CON.M** = 000_B.

Eight data bits are transmitted or received synchronous to a shift clock generated by the internal baudrate generator. The shift clock is active only as long as data bits are transmitted or received.

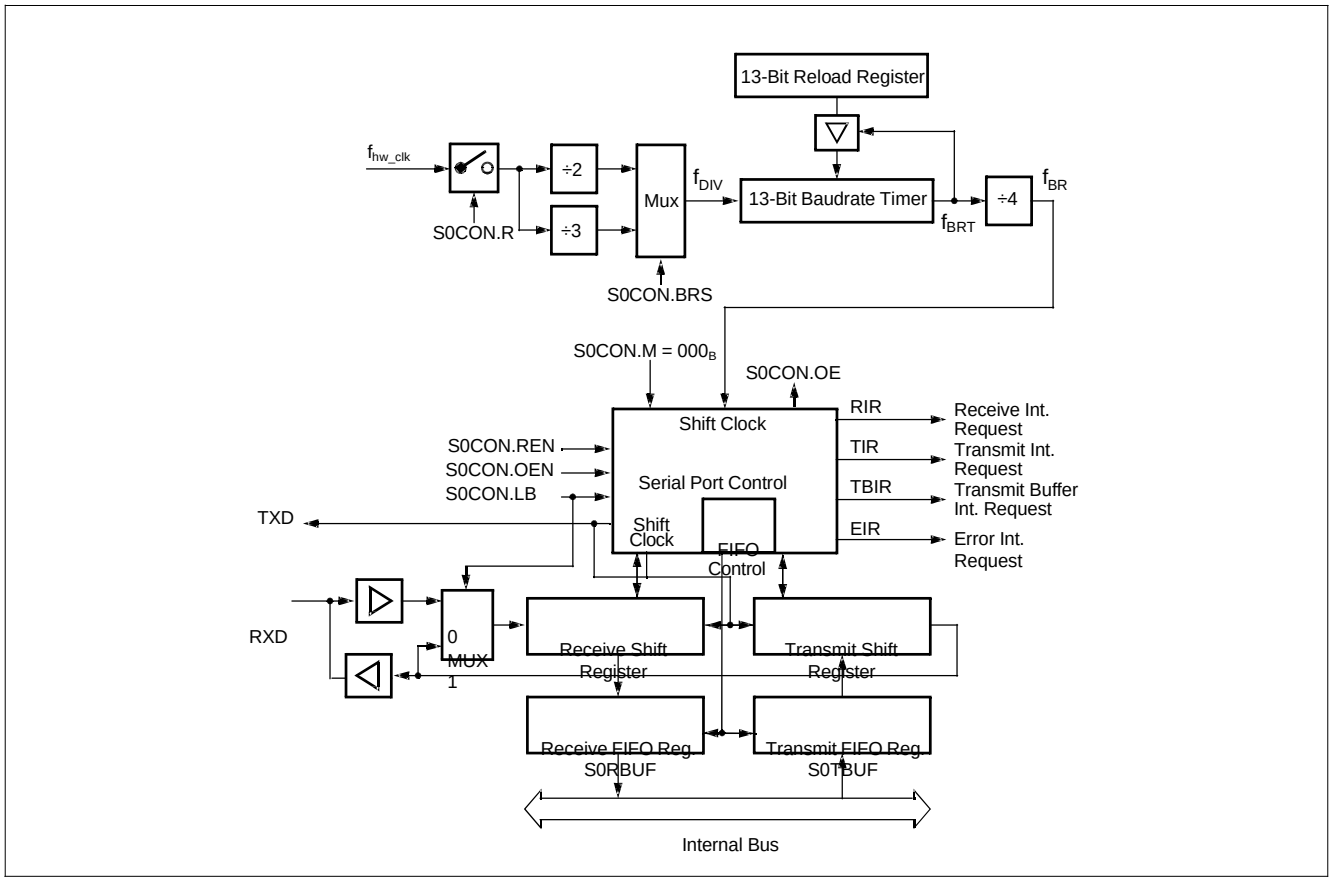


Figure 122 Synchronous Mode of Serial Channel ASC0

9.3.5.11 Synchronous Transmission

Synchronous transmission begins within four state times after data has been loaded into **S0TBUF**, provided that **S0CON.R** is set and **S0CON.REN** is cleared (half-duplex, no reception). Exception: in Loopback Mode (bit **S0CON.LB** set), **S0CON.REN** must be set for reception of the transmitted byte. Data transmission is double-buffered. When the transmitter is idle, the transmit data loaded into **S0TBUF** is immediately moved to the transmit shift register, thus freeing **S0TBUF** for more data. This is indicated by the transmit buffer interrupt request line TBIR being activated. **S0TBUF** may now be loaded with the next data, while transmission of the previous continuous. The data bits are transmitted synchronous with the shift clock. After the bit time for the eighth data bit, both the TXD and RXD lines will go high, the transmit interrupt request line TIR is activated, and serial data transmission stops.

Note: Pin TXD must be configured for alternate data output in order to provide the shift clock. Pin RXD must also be configured for output during transmission.

9.3.5.12 Synchronous Reception

Synchronous reception is initiated by setting bit **S0CON.REN**. If bit **S0CON.R** is set, the data applied at RXD is clocked into the receive shift register synchronous to the clock that is output at TXD. After the eighth bit has been shifted in, the contents of the receive shift register are transferred to the receive data buffer **S0RBUF**, the receive interrupt request line RIR is activated, the receiver enable bit **S0CON.REN** is reset, and serial data reception stops.

Note: Pin TXD must be configured for alternate data output in order to provide the shift clock. Pin RXD must be configured as alternate data input.

Synchronous reception is stopped by clearing bit **S0CON.REN**. A currently received byte is completed, including the generation of the receive interrupt request and an error interrupt request, if appropriate. Writing to the transmit buffer register while a reception is in progress has no effect on reception and will not start a transmission.

If a previously received byte has not been read out of a full receive buffer at the time the reception of the next byte is complete, both the error interrupt request line EIR and the overrun error status flag **S0CON.OE** will be activated/set, provided the overrun check has been enabled by bit **S0CON.OEN**.

9.3.5.13 Synchronous Timing

Figure 123 shows timing diagrams of the ASC0 Synchronous Mode data reception and data transmission. In idle state, the shift clock level is high. With the beginning of a synchronous transmission of a data byte, the data is shifted out at RXD with the falling edge of the shift clock. If a data byte is received through RXD, data is latched with the rising edge of the shift clock.

Between two consecutive receive or transmit data bytes, one shift clock cycle (f_{BR}) delay is inserted.

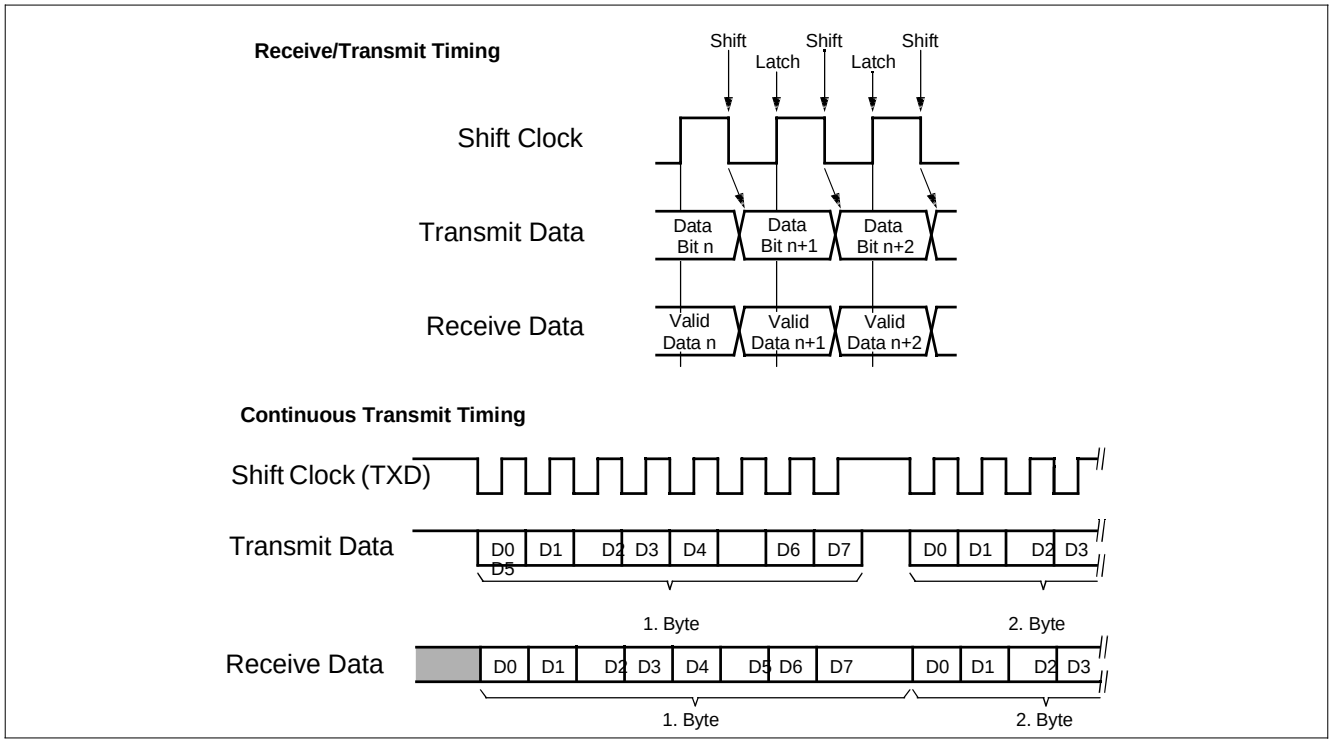


Figure 123 ASC0 Synchronous Mode Waveforms

9.3.5.14 Baudrate Generation

The serial channel ASC0 has its own dedicated 13-bit baudrate generator with reload capability, allowing baudrate generation independent of other timers.

The baudrate generator is clocked with a clock (f_{DIV}) derived via a prescaler from the ASC0 input clock f_{hw_clk}. The baudrate timer counts downwards and can be started or stopped through the baudrate generator run bit **S0CON.R**. Each underflow of the timer provides one clock pulse to the serial channel. The timer is reloaded with the value stored in its 13-bit reload register each time it underflow. The resulting clock f_{BRT} is again divided by a factor for the baudrate clock (16 in Asynchronous Modes and 4 in Synchronous Mode). The prescaler is selected by the bits **S0CON.BRS** and **S0CON.FDE**. In addition to the two fixed dividers, a fractional divider prescaler unit is available in the Asynchronous Modes that allows selection of prescaler divider ratios of n/512 with n = 0...511. Therefore, the baudrate of ASC0 is determined by the module clock, the content of **S0FDV**, the reload value of **S0BG**, and the operating mode (asynchronous or synchronous).

Register **S0BG** is the dual-function Baudrate Generator/Reload register. Reading **S0BG** returns the contents of the timer **S0BG.BR_VALUE** (bits 15...13 return zero), while writing to **S0BG** always updates the reload register (bits 15...13 are insignificant).

An autoreload of the timer with the contents of the reload register is performed each time **S0CON.BG** is written to. However, if **S0CON.R** is cleared at the time a write operation to **S0CON.BG** is performed, the timer will not be reloaded until the first instruction cycle after **S0CON.R** was set. For a clean baudrate initialization, **S0CON.BG** should be written only if **S0CON.R** is reset. If **S0CON.BG** is written while **S0CON.R** is set, unpredictable behavior of the ASC0 may occur during running transmit or receive operations.

The ASC0 baudrate timer reload register **S0BG** contains the 13-bit reload value for the baudrate timer in Asynchronous and Synchronous modes.

Note: IrDA Baudrate re-configuration causes unwanted pulses on the TXD PAD under following conditions:

1.ASC0 is in IrDA Mode
*2.**S0CON.R** is reset to change the Baudrate according the specification*
3.ASC0 is configured
*4.**S0CON.R** is set.*

During the configuration phase (point 3.) a high pulse on TXD is generated, which disturbs the IrDAProtocol. This problems occurs, because of the different Stop-bit active values:

- Asynchronous Mode: HIGH-active
- IrDA Mode: LOW-active.

*When the **S0CON**.R is reset during the reconfiguration of the Baudrate, the ASC changes to the Asynchronous Mode (--> HIGH Stop-Bit) and after reconfiguration the **S0CON**.R is set again (--> LOW Stop-Bit).*
To avoid this Problem the ASC0 has to be decoupled during the configuration phase of the IrDA from the PAD, by using the GPIO functionality of the TXD Pin.
--> Configure the according GPIO as OUTPUT and set the value to '0'.

9.3.5.15 Baudrate in Asynchronous Mode

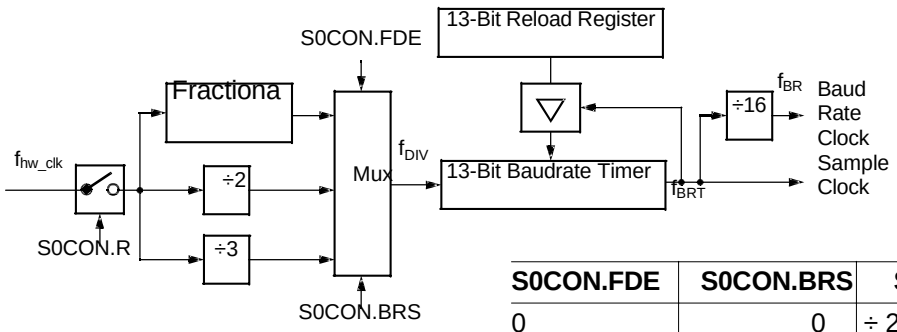
For Asynchronous Mode, the baudrate generator provides a clock f_{BRT} with sixteen times the rate of the established baudrate. Every received bit is sampled at the 7th, 8th, and 9th cycle of this clock. The clock divider circuitry, which generates the input clock for the 13-bit baudrate timer, is extended by a fractional divider circuitry that allows adjustment for more accurate baudrate and the extension of the baudrate range.

The baudrate of the baudrate generator depends on the following bits and register values:

- Input clock f_{hw_clk}
- Selection of the baudrate timer input clock f_{DIV} by bits **S0CON.FDE** and **S0CON.BRS**
- If bit **S0CON.FDE** is set (fractional divider): value of register **S0FDV**
- Value of the 13-bit reload register **S0BG.**

The output clock of the baudrate timer with the reload register is the sample clock in the Asynchronous Modes of the ASC0. For baudrate calculations, this baudrate clock f_{BR} is derived from the sample clock f_{DIV} by a division by 16.

The ASC0 fractional divider register **S0FDV** contains the 9-bit divider value for the fractional divider (Asynchronous Mode only). It is also used for reference clock generation of the autobaud detection unit.



S0CON.FDE	S0CON.BRS	Selected Divider
0	0	÷ 2
0	1	÷ 3
1	X	Fractional Divider

Figure 124 ASC0 Baudrate Generator Circuitry in Asynchronous Modes

Using the fixed Input Clock Divider

The baudrate for asynchronous operation of serial channel ASC0 when using the fixed input clock divider ratios (**S0CON.FDE** = 0) and the required reload value for a given baudrate can be determined by the following formulas:

Table 65 Asynchronous Baudrate Formulas using Fixed Input Clock Dividers

S0CON.FDE	S0CON.BRS	BG	Formula
0	0	0 ... 8191	$Baudrate = \frac{f_{hw\_clk}}{32 \times (BG+1)}$ $BG = \frac{f_{hw\_clk}}{32 \times Baudrate} - 1$
	1		$Baudrate = \frac{f_{hw\_clk}}{48 \times (BG+1)}$ $BG = \frac{f_{hw\_clk}}{48 \times Baudrate} - 1$

*Note: BG represents the contents of the reload register bits **S0BG.BR_VALUE**, taken as unsigned 13-bit integer.*
Table 66 lists various commonly used baudrates together with the required reload values and the deviation errors compared to the intended baudrate.

Table 66 Typical Asynchronous Baudrates Using Fixed Input Clock Dividers

Baudrate	S0CON.BRS = 0		S0CON.BRS = 1	
	Deviation Error	Reload Value	Deviation Error	Reload Value
	f_{hw_clk} = 26 MHz			
812.5 KBaud	---	0000_H	---	---
541.7 KBaud	---	---	---	0000_H
19.2 Baud	+0.8 %	0029_H	+0.8 %	$001B_H$
9600 Baud	-0.4 %	0054_H	+0.8 %	0037_H
4800 Baud	+0.2 %	$00A8_H$	-0.1 %	0070_H
2400 Baud	+0.1 %	0152_H	-0.1 %	$00E1_H$
1200 Baud	+0.0 %	$02A4_H$	+0.1 %	$01C2_H$
110 Baud	+0.0 %	$1CD9_H$	+0.0 %	$133B_H$
	f_{hw_clk} = 52 MHz			
1625 KBaud	---	0000_H	---	---
1083.3 KBaud	---	---	---	0000_H
19.2 Baud	-0.4 %	0054_H	+0.8	0037_H
9600 Baud	+0.2 %	$00A8_H$	-0.1 %	0070_H
4800 Baud	+0.1 %	0152_H	-0.1 %	$00E1_H$
2400 Baud	+0.0 %	$02A4_H$	-0.1 %	$01C2_H$
1200 Baud	+0.0 %	0549_H	+0.0 %	0386_H
110 Baud	+0.0 %	$39B4_H$	+0.0 %	2677_H

*Note: **S0CON.FDE** must be 0 to achieve the baudrates in **Table 66**. The deviation errors given in **Table 66** are rounded. Using a baudrate crystal provides correct baudrates without deviation errors.*

Using the Fractional Divider

When the fractional divider is selected, the input clock f_{DIV} for the baudrate timer is derived from the module clock f_{hw_clk} by a programmable divider. If **S0CON.FDE** is set, the fractional divider is activated. It divides f_{hw_clk} by a fraction of n/512 for any value of n from 0 to 511. If n = 0, the divider ratio is 1, which means that $f_{DIV} = f_{hw\_clk}$. In general, the fractional divider allows the baudrate to be programmed with much more accuracy than with the two fixed prescaler divider stages.

Table 67 Asynchronous Baudrate Formulas using the Fractional Input Clock Divider

S0CON.FDE	S0CON.BRS	S0BG	S0FDV	Formula
1	-	0 ... 8191	1 ... 511	$\text{Baudrate} = \frac{\text{FDV}}{512} \times \frac{f_{hw\_clk}}{16 \times (BG+1)}$
			0	$\text{Baudrate} = \frac{f_{hw\_clk}}{16 \times (BG+1)}$

*Note: BG represents the contents of the reload register **S0BG.BR_VALUE**, taken as an unsigned 13-bit integer. FDV represents the contents of the fractional divider **S0FDV.FD_VALUE** taken as an unsigned 9-bit integer.*

Table 68 Typical Asynchronous Baudrates using the Fractional Input Clock Divider

f_{hw_clk}	Desired Baudrate	S0BG	S0FDV	Resulting Baudrate	Deviation
52 MHz	115.2 kBaud	$0D_H$	$0FE_H$	115.164 kBaud	0.03 %
	57.6 kBaud	$0D_H$	$07F_H$	57.582 kBaud	0.03 %
	38.4 kBaud	19_H	079_H	38.403 kBaud	0.01 %
	19.2 kBaud	28_H	$07C_H$	19.175 kBaud	0.01 %

9.3.5.16 Baudrate in Synchronous Mode

For synchronous operation, the baudrate generator provides a clock with four times the rate of the established baudrate (see **Figure 125**).

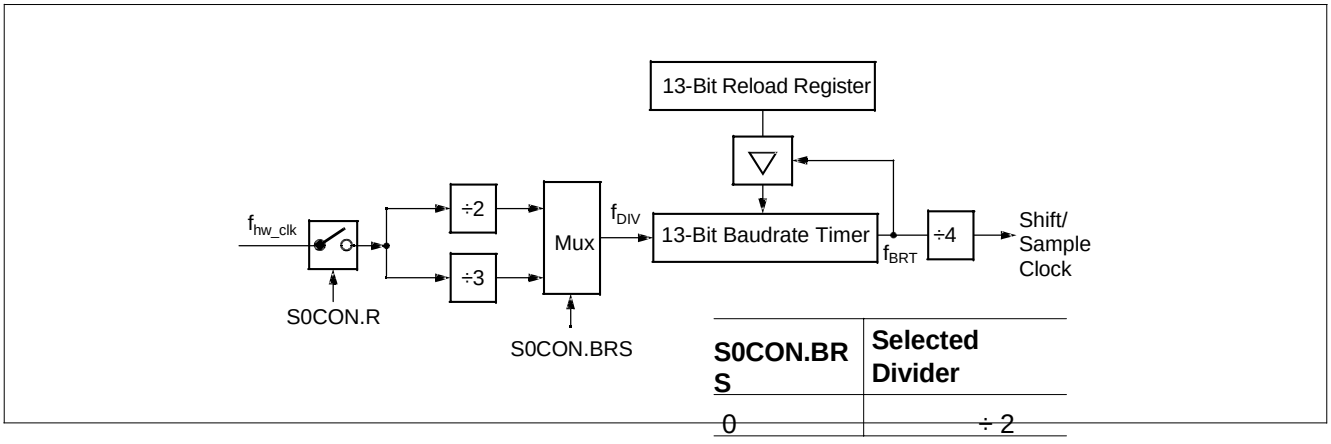


Figure 125 ASC0 Baudrate Generator Circuitry in Synchronous Mode

The baudrate for synchronous operation of serial channel ASC0 can be determined by the formulas as shown in **Table 69**.

Table 69 Synchronous Baudrate Formulas

S0CON.BRS	S0BG	Formula
0	0 ... 8191	$Baudrate = \frac{f_{hw\_clk}}{8 \times (BG+1)}$ $BG = \frac{f_{hw\_clk}}{8 \times Baudrate} - 1$
1		$Baudrate = \frac{f_{hw\_clk}}{12 \times (BG+1)}$ $BG = \frac{f_{hw\_clk}}{12 \times Baudrate} - 1$

*Note: BG represents the contents of the reload register **S0BG.BR_VALUE**, taken as an unsigned 13-bit integers.*

9.3.5.17 Autobaud Detection

9.3.5.18 General Operation

Autobaud Detection provides a capability to recognize the mode and the baudrate of an asynchronous input signal at RXD. Generally, the baudrates to be recognized must be known by the application. With this knowledge always a set of nine baudrates can be detected. The Autobaud Detection is not designed to calculate a baudrate of an unknown asynchronous frame.

Figure 126 shows how the Autobaud Detection is integrated into its Asynchronous Mode configuration. The RXD data line is an input to the autobaud detection unit. The clock f_{DIV}, generated by the fractional divider, is used by the autobaud detection unit as time base. After successful recognition of baudrate and Asynchronous Mode of the RXD data input signal, bits in the **S0CON** register and the value of the **S0BG** register in the baudrate timer are set to the appropriate values, and the ASC0 can start immediately with the reception of serial input data.

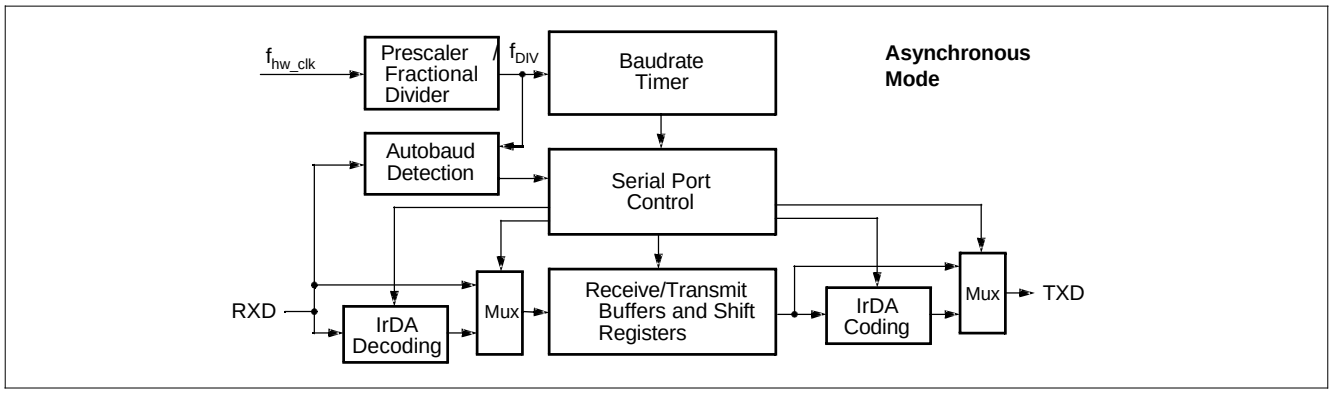


Figure 126 Asynchronous Mode Block Diagram

Note: Autobaud detection is not available in Synchronous Mode

The following sequence must be generally executed to start the autobaud detection unit for operation:

- Definition of the baudrates to be detected: standard or non-standard baudrates
- Programming of the prescaler/fractional divider to select a specific value of f_{DIV}
- Starting the prescaler/fractional divider (setting **S0CON.R**)
- Preparing the interrupt system
- Enabling the autobaud detection (setting **S0ABCON.EN** and the interrupt enable bits in **S0ABCON** for interrupt generation, if required)
- Polling interrupt request flag or waiting for the autobaud detection interrupt

9.3.5.19 Serial Frames for Autobaud Detection

The Autobaud Detection is based on the serial reception of a specific two-byte serial frame. This serial frame is build up by the two ASCII bytes “at” or “AT” (“aT” or “At” are not allowed). Both byte combinations can be detected

in five types of asynchronous frames. **Figure 127** and **Figure 128** show the serial frames which are detected at least.

Note: Some other two-byte combinations will be defined too.

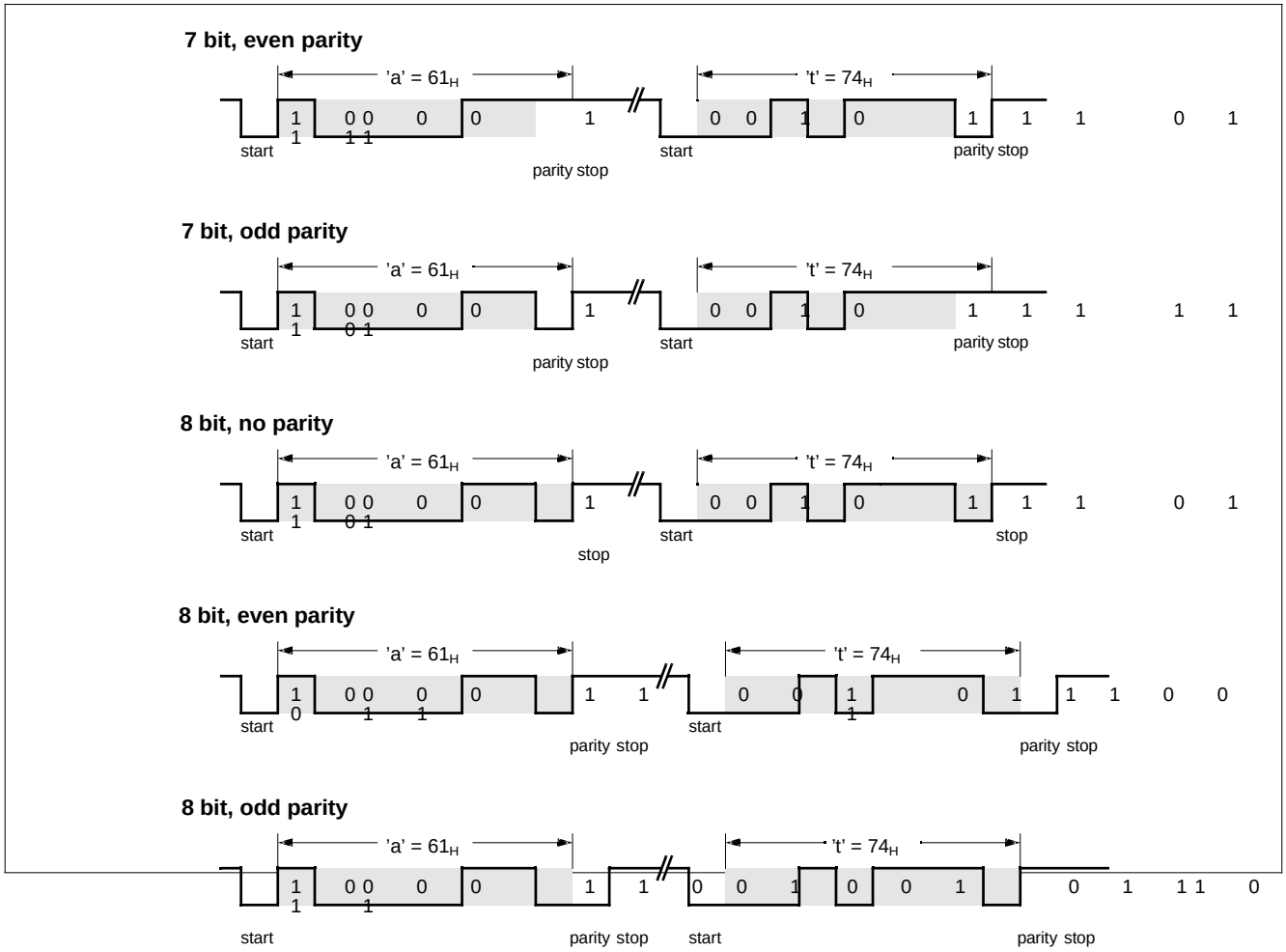


Figure 127 Two-Byte Serial Frames with ASCII 'at'

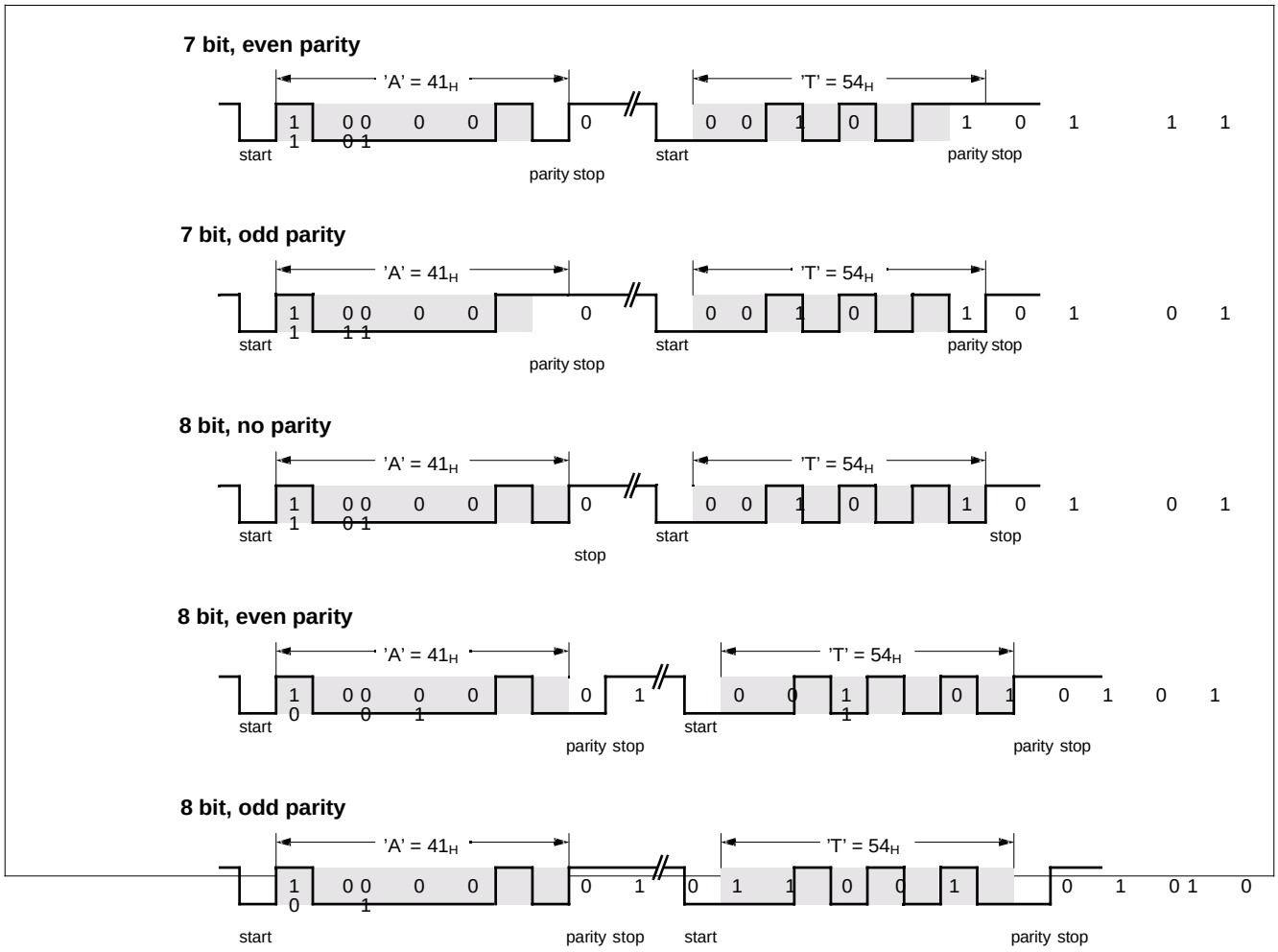


Figure 128 Two-Byte Serial Frames with ASCII 'AT'

9.3.5.20 Baudrate Selection and Calculation

Autobaud Detection requires some calculations concerning the programming of the baudrate generator and the baudrates to be detected. Two steps must be considered:

- Defining the baudrate(s) to be detected
- Programming of the baudrate timer prescaler - setup of the clock rate of f_{DIV}.

In general, the baudrate generator in Asynchronous Mode is build up by two parts (see also **Figure 124**):

- The clock prescaler part which derives f_{DIV} from f_{hw_clk}
- The baudrate timer part which generates the sample clock f_{BRT} and the baudrate clock f_{BR}.

Prior to an Autobaud Detection the prescaler part has to be setup by the CPU while the baudrate timer is initialized with a 13-bit value (**S0BG.BR_VALUE**) automatically after a successful autobaud detection. For the following calculations, the fractional divider is used (**S0CON.FDE** = 1).

Note: It is also possible to use the fixed divide-by-2 or divide-by-3 prescaler. But the fractional divider allows to adapt f_{DIV} much more precise to the required value.

Standard Baudrates

For standard baudrate detection the baudrates as shown in **Table 70** can be e.g. detected. Therefore, the output frequency f_{DIV} of the baudrate generator must be set to a frequency derived from the module clock f_{hw_clk} in a way

that it is equal to 11.0592 MHz. The value to be written into register **S0FDV** is the nearest integer value which is calculated according the following formula:

$$FDV = \frac{512 \times 11.0592\ MHz}{f_{MOD}} \quad (60)$$

Table 70 defines the nine standard baudrates (Br0 - Br8) which can be detected for f_{DIV} = 11.0592 MHz.

Table 70 Autobaud Detection using Standard Baudrates (f_{DIV} = 11.0592 MHz)

Baudrate Numbering	Detectable Standard Baudrate	Divide Factor d_f	BG is loaded after detection with value
Br0	230.400 kBaud	48	$2 = 002_H$
Br1	115.200 kBaud	96	$5 = 005_H$
Br2	57 600 kBaud	192	$11 = 00B_H$
Br3	38 400 kBaud	288	$17 = 011_H$
Br4	19 200 kBaud	576	$35 = 023_H$
Br5	9600 Baud	1152	$71 = 047_H$
Br6	4800 Baud	2304	$143 = 08F_H$
Br7	2400 Baud	4608	$287 = 11F_H$
Br8	1200 Baud	9216	$575 = 23F_H$

According **Table 70** a baudrate of 9600 Baud is achieved when register BG is loaded with a value of 047_H, assuming that f_{DIV} has been set to 11.0592 MHz.
Table 70 also lists a divide factor d_f which is defined with the following formula:

$$Baudrate = \frac{f_{DIV}}{d_f} \quad (61)$$

This divide factor d_f defines a fixed relationship between the prescaler output frequency f_{DIV} and the baudrate to be detected during the Autobaud Detection operation. This means, changing f_{DIV} results in a totally different baudrate table in means of baudrate values. For the baudrates to be detected, the following relations are always valid:

Br0 = $f_{DIV} / 48_D$, Br1 = $f_{DIV}/96_D$, ... up to Br8 = $f_{DIV}/9216_D$,

A requirement for detecting standard baudrates up to 230 400 kBaud is the f_{DIV} minimum value of 11.0592 MHz. With the value **S0FDV.FD_VALUE** the fractional divider f_{DIV} is adapted to the module clock frequency f_{MOD}. **Table 71** defines the deviation of the standard baudrates when using autobaud detection depending on the module clock f_{hw_clk}.

Table 71 Standard Baudrates - Deviations and Errors for Autobaud Detection

f_{hw_clk}	S0FDV	Error in f_{DIV}
10 MHz		not possible
12 MHz	472	+ 0.03 %
13 MHz	436	+ 0.1 %
16 MHz	354	+ 0.03 %
18 MHz	315	+ 0.14 %
18.432 MHz	307	- 0.07 %
20 MHz	283	- 0.04 %
24 MHz	236	+ 0.03 %
25 MHz	226	- 0.22 %

Table 71 Standard Baudrates - Deviations and Errors for Autobaud Detection (cont'd)

f_{hw_clk}	S0FDV	Error in f_{DIV}
10 MHz		not possible
12 MHz	472	+ 0.03 %
13 MHz	436	+ 0.1 %
16 MHz	354	+ 0.03 %
18 MHz	315	+ 0.14 %
18.432 MHz	307	- 0.07 %
20 MHz	283	- 0.04 %
26 MHz	218	+ 0.10 %
30 MHz	189	+ 0.14 %
33 MHz	172	+ 0.24 %
40 MHz	142	+ 0.31 %
50 MHz	113	- 0.23 %
52 MHz	109	- 0.10 %

Note: If the deviation of the baudrate after autobaud detection is to high, the baudrate generator (fractional divider **S0FDV** *and reload register* **S0BG***) can be reprogrammed if required to get a more precise baudrate with less error.*

Non-Standard Baudrates

Due to the relationship between Br0 to Br8 in **Table 70** concerning the divide factor d_f other baudrates than the standard baudrates can be also selected. For example, if a baudrate of 50 kBaud has to be detected, Br2 is defined as baudrate for the 50 kBaud selection. This further results in:

f_{DIV} = 50 kBaud x d_f@Br2 = 50 kBaud x 192 = 9.6 MHz

Therefore, depending on the module clock frequency f_{hw_clk}, the value of the fractional divider (register **S0FDV** must be set in this example according the formula:

$$FDV = \frac{512 \times f_{DIV}}{f_{hw\_clk}} \quad \text{with } f_{DIV} = 9.6 \text{ MHz} \tag{62}$$

Using this selection (f_{DIV} = 9.6 MHz), the detectable baudrates start at 200 kBaud (Br0) down to 1042 Baud (Br8). **Table 72** shows the baudrate table for this example.

Table 72 Autobaud Detection using Non-Standard Baudrates (f_{DIV} = 9.6 MHz)

Baudrate Numbering	Detectable Non-Standard Baudrates	Divide Factor d_f	BG is loaded after detection with value
Br0	200.000 kBaud	48	2 = 002_H
Br1	100.000 kBaud	96	5 = 005_H
Br2		50 kBaud	19211 =
$00B_H$ Br3		33.333 kBaud	28817 =
011_H Br4		16.667 kBaud	57635 =
023_H Br5		8333 Baud	1152
	71 = 047_H Br6		4167 Baud
	2304	143 = $08F_H$ Br7	2083
Baud	4608	287 = $11F_H$ Br8	1047
Baud	9216	575 = $23F_H$	

9.3.5.21 Overwriting Registers on Successful Autobaud Detection

With a successful Autobaud Detection some bits in register **S0CON** and **S0BG** are automatically set to a value which corresponds to the mode and baudrate of the detected serial frame conditions (see **Table 73**). In control register **S0CON** the mode control bits **S0CON.M** and the parity select bit **S0CON.ODD** are overwritten. Register **S0BG** is loaded with the 13-bit reload value for the baudrate timer.

Table 73 Autobaud Detection Overwrite Values for the CON Register

Detected Parameters		S0CON.M	S0CON.ODD	BG.BR_VALUE
Operating Mode	7 bit, even parity	0 1 1	0	-
	7 bit, odd parity	0 1 1	1	
	8 bit, even parity	1 1 1	0	
	8 bit, odd parity	1 1 1	1	
	8 bit, no parity	0 0 1	0	
Baudrate	Br0	-	-	2 = 002_H
	Br1			5= 005_H
	Br2			11= $00B_H$
	Br3			17= 011_H
	Br4			35= 023_H
	Br5			71= 047_H
	Br6			143= $08F_H$
	Br7			287= $11F_H$
	Br8			575= $23F_H$

Note: The autobaud detection interrupts are described in ***Section 9.3.5.25 Interrupts (on Page 331)****.*

9.3.5.22 Hardware Flow Control

The ASC0 supports both software- and hardware-controlled flow control with Request to Send (RTS)/Clear to Send (CTS) handshaking. Flow control is only available in asynchronous mode. In software mode (**S0FCCON.RTSEN** = 0), the handshake line RTS_N is controlled by the bit **S0FCCON.RTS**. In hardware mode (**S0FCCON.RTSEN** = 1), RTS is controlled depending on whether the receive FIFO is enabled (**S0RXFCON.RXFEN** = 1) or not. If the receive FIFO is enabled, RTS is active (RTS_N = 0) as long as the receive FIFO level is below the programmable RTS trigger level **S0FCCON.RTSTL**.

RTS is deactivated (RTS_N = 1), if the receive FIFO level is equal to or greater than the RTS trigger level. The RXFIFO is described in **Section 9.3.5.6 Receive FIFO Operation (on Page 314)**. If the receive FIFO is disabled, RTS is active as long as the receive buffer is empty and inactive as soon as a frame is in the receive buffer, which has not been read yet. If automatic CTS recognition is enabled (**S0FCCON.CTSEN** = 1), data frames are transmitted only as long as the handshake line CTS is active. If CTS is deactivated while a frame is being transmitted, the transmission of this frame will still be completed. Independent of the CTS enable bit **S0FCCON.CTSEN**, the status of the CTS line can be monitored by reading the CTS status bit **S0FSTAT.CTS**. Also, the interrupt CTS_INT indicates a change on the handshake line CTS_N (see **Section 9.3.5.25 (on page 331)**. In loop back mode (**S0CON.LB** = 1), RTS_N is connected to CTS_N internally.

9.3.5.23 Hardware Error Detection Capabilities

To improve the safety of serial data exchange, the serial channel ASC0 provides an error interrupt request flag to indicate the presence of an error, and three (selectable) error status flags in register **S0CON** to indicate which error has been detected during reception. Upon completion of a reception, the error interrupt request line EIR will be activated simultaneously with the receive interrupt request line RIR, if one or more of the following conditions are met:

- If the framing error detection enable bit **S0CON.FEN** is set and any of the expected stop bits is not high, the framing error flag **S0CON.FE** is set, indicating that the error interrupt request is due to a framing error (Asynchronous Mode only).
- If the parity error detection enable bit **S0CON**.**PEN** is set in the modes where a parity bit is received, and the parity check on the received data bits proves false, the parity error flag **S0CON.PE** is set, indicating that the error interrupt request is due to a parity error (Asynchronous Mode only).
- If the overrun error detection enable bit **S0CON.OEN** is set and the last character received was not read out of the receive buffer by software at the time the reception of a new frame is complete, the overrun error flag **S0CON.OE** is set indicating that the error interrupt request is due to an overrun error (Asynchronous and Synchronous Mode).

9.3.5.24 Receive Timeout Detection

A receive timeout detection functionality is provided to detect timeout conditions while the ASC0 is operating in reception mode. The submodule consists of a 16-bit timer and the timeout control register **S0TMO**. If the timeout control register is set to zero, the timeout detection is disabled. If the timeout detection is enabled, the timer is loaded with the contents of the timeout control register after the reception of the first character. Then the timer is decremented with each shift clock cycle generated by the baud rate generator. The timer is reloaded after the reception of each character. If the timer reaches zero, a timeout interrupt is generated. Thereby the timeout module has an inter-character-gap timeout functionality.

9.3.5.25 Interrupts

Eight interrupts sources are provided for serial channel ASC0:

- TIR indicates a Transmit Interrupt
- TBIR indicates a Transmit Buffer Interrupt
- RIR indicates a Receive Interrupt
- EIR indicates an Error Interrupt of the serial channel
- CTSIR indicates a change on the handshake line CTS_N
- TMOIR indicates a receive timeout condition.

The autobaud detection unit provides two additional interrupts, the ABSTIR start of autobaud operation interrupt and the ABDETIR autobaud detected interrupt.

The interrupt output lines TBIR, TIR, RIR, EIR, CTSIR, TMOIR, ABSTIR, and ABDETIR are activated (active state) for two periods of the module clock f_{hw_clk}.

The cause of an error interrupt request (framing, parity, overrun error) can be identified by the error status flags **S0CON.FE**, **S0CON.PE**, and **S0CON.OE**. For the two autobaud detection interrupts register **S0ABSTAT** provides status information.

Note: Unlike the error interrupt request line EIR, the error status flags **S0CON.FE**, **S0CON.PE**, and **S0CON.OE** *are not reset automatically but must be cleared by software.*

For normal operation (that is, besides the error interrupt) the ASC0 provides three interrupt requests to control data exchange via this serial channel:

- TBIR is activated when data is moved from **S0TBUF** to the transmit shift register.
- TIR is activated before the last bit of an asynchronous frame is transmitted, or after the last bit of a synchronous frame has been transmitted.
- RIR is activated when the received frame is moved to **S0RBUF**.

Note: While the receive task is handled by a single interrupt handler, the transmitter is serviced by two interrupt handlers. This provides advantages for the servicing software.

For single transfers it is sufficient to use the transmitter interrupt (TIR), which indicates that the previously loaded data has been transmitted, except for the last bit of an asynchronous frame.

For multiple back-to-back transfers it is necessary to load the following piece of data at last until the time the last bit of the previous frame has been transmitted. In Asynchronous Mode this leaves just one bit-time for the handler to respond to the transmitter interrupt request, in Synchronous Mode it is impossible at all.

Using the transmit buffer interrupt (TBIR) to reload transmit data gives the time to transmit a complete frame for the service routine, as **S0TBUF** may be reloaded while the previous data is still being transmitted.

The start of autobaud operation interrupt ABSTIR is generated whenever the autobaud detection unit is enabled (**S0ABCON.ABEN** and **S0ABCON.ABDETEN** and **S0ABCON.ABSTEN** are set), and a start bit has been detected at RXD. In this case ABSTIR is generated during Autobaud Detection whenever a start bit is detected. The autobaud detected interrupt ABDETIR is always generated after recognition of the second character of the two-byte frame, this means after a successful Autobaud Detection. If **S0ABCON.FCDETEN** is set the autobaud detected interrupt ABDETIR is also generated after the recognition of the first character of the two-byte frame.

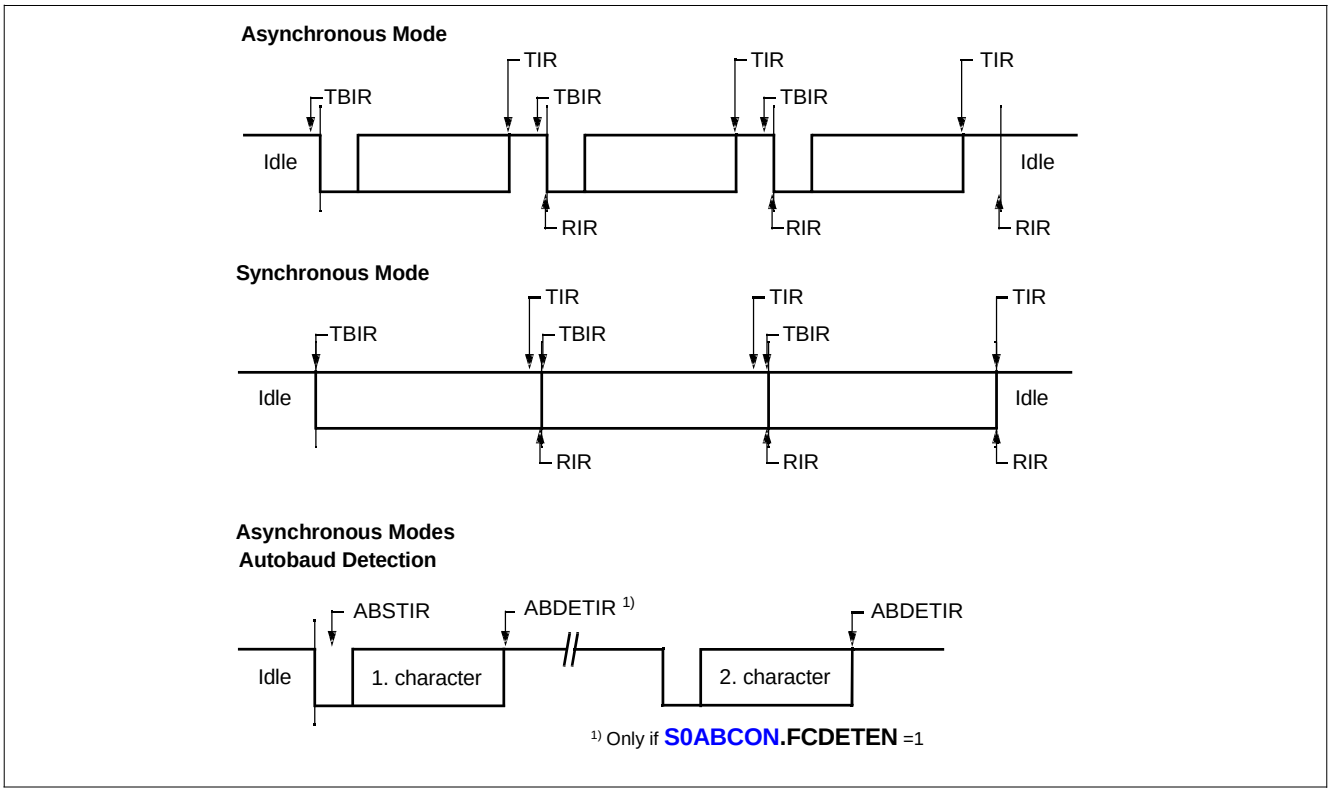


Figure 129 ASC0 Interrupt Generation

As shown in **Figure 129**, TBIR is an early trigger for the reload routine, while TIR indicates the completed transmission. Therefore, software using handshake should rely on TIR at the end of a data block to ensure that all data has actually been transmitted.

9.3.6 Registers

For the register addresses refer to **Section 10.1 PD-Bus Register Addresses (on Page 481)**.

Table 74 ASC0 Register Summary

Name	Clock	Access Condition	Description
S0PISEL	hw_clk[1)]	bit addressable	Peripheral Input Select Register
S0PERID	hw_clk[1)]	non bit addressable	Peripheral ID Number Register
S0CON	hw_clk[1)]	bit addressable	Control Register
S0BG	hw_clk[1)]	non bit addressable	Baudrate Timer Reload Register
S0FDV	hw_clk[1)]	non bit addressable	Fractional Divider Register
S0PWM	hw_clk[1)]	non bit addressable	IrDA Pulse Mode and Width Register
S0TBUF	hw_clk[1)]	non bit addressable	Transmit Buffer Register
S0RBUF	hw_clk[1)]	non bit addressable	Receive Buffer Register
S0ABCON	hw_clk[1)]	bit addressable	Autobaud Control Register
S0ABSTAT	hw_clk[1)]	non bit addressable	Autobaud Status Register
S0RXFCON	hw_clk[1)]	non bit addressable	Receive FIFO Control Register
S0TXFCON	hw_clk[1)]	non bit addressable	Transmit FIFO Control Register
S0FSTAT	hw_clk[1)]	non bit addressable	FIFO Status Register
S0FCCON	hw_clk[1)]	bit addressable	Flowcontrol Control Register
S0FCSTAT	hw_clk[1)]	bit addressable	Flowcontrol Status Register
S0TMO	hw_clk[1)]	non bit addressable	RX Timeout Control Register

1) Refer to the Clock Domain in the **System Integration (on Page 306)**.

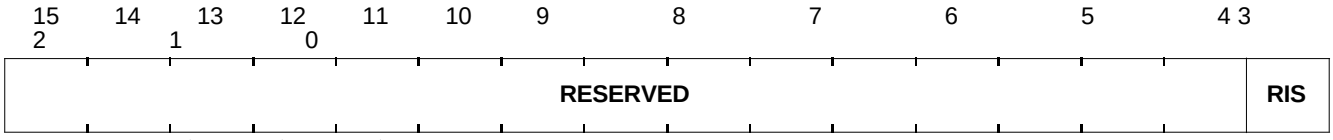
9.3.6.1 Port Input Select Register

The **S0PISEL** register selects the source to receiver data (RXD) of the ASC0.

S0PISEL.RIS must remain at the reset default value of 0 to select RXD. This ensures correct operation in Synchronous and Asynchronous modes

S0PISEL

Port Input Select Register **Reset value: 0000_H**



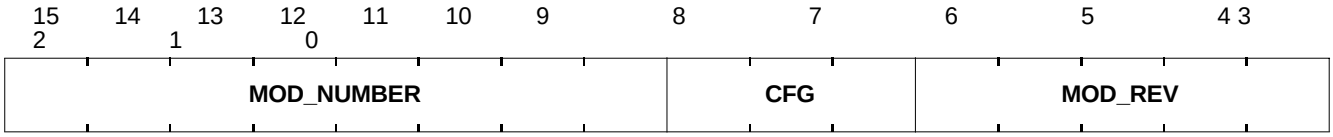
Field	Bits	Type	Description
RIS	0	rw	**Receiver Input Select** 0 RXD pin selected (default) 1 Tied to logical 0, no input received
RESERVED	15:1	r	Reserved; these bits must be left at their reset values.

9.3.6.2 ASC0 Identification Register

The **S0PERID** register contains the ASC0 Module Number, Configuration Number, and Revision Number.

S0PERID

Peripheral Identification Register **Reset values: $44E4_H$**



Field	Bits	Type	Description
MOD_REV	4:0	r	**ASC0 Revision Number** Value = 04_H
CFG	7:5	r	**Configuration Number** Bit 7 = 1 because Autobaud mode is implemented Bit 6 = 1 because Irda mode is implemented Bit 5 = 1 because FIFO mode is implemented Value = 7_H
MOD_NUMBER	15:8	r	**ASC0 Number** For ASC0, it is 44_H

9.3.6.3 Control Register

The operating mode of the serial channel ASC0 is controlled by its control register. This register contains control bits for mode and error check selection, and status flags for error identification.

S0CON

Control Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4 3 2 1 0
R	**LB**	**BRS**	**ODD**	**FDE**	**OE**	**FE**	**PE**	**OEN**	**FEN**	**PEN/ RXDI**	**REN**	**STP**	**M**

Field	Bits	Type	Description
M	2:0	rw	**Mode Control** 000 8-bit-data for synchronous operation 001 8-bit-data for asynchronous operation 010 8-bit-data IrDA Mode for asynchronous operation 011 7-bit-data and parity for asynchronous operation 100 9-bit-data for asynchronous operation 101 8-bit-data and wake up bit for asynchronous operation 110 Reserved. Do not use this combination 111 8-bit-data and parity for asynchronous operation
STP	3	rw	**Number of Stop Bits Selection** 0 One stop bit 1 Two stop bits
REN	4	rwh	**Receiver Enable Bit** 0 Receiver disabled 1 Receiver enabled *Note: Bit is cleared by hardware after reception of a byte in the Synchronous Mode*
PEN	5	rw	Any modes without the IrDA Mode **Parity Check Enable** 0 Ignore parity 1 Check parity
RXDI			Only in IrDA Mode **RXDI Invert in IrDA Mode** 0 RXD input is not inverted 1 RXD input is inverted
FEN	6	rw	**Framing Check Enable** (Asynchronous Mode only) 0 Ignore framing errors 1 Check framing errors
OEN	7	rw	**Overrun Check Enable** 0 Ignore overrun errors 1 Check overrun errors
PE	8	rwh	**Parity Error Flag** Set by hardware on a parity error (**PEN** = 1. Must be cleared by software.
FE	9	rwh	**Framing Error Flag** Set by hardware on a framing error (**FEN** = 1). Must be cleared by software.

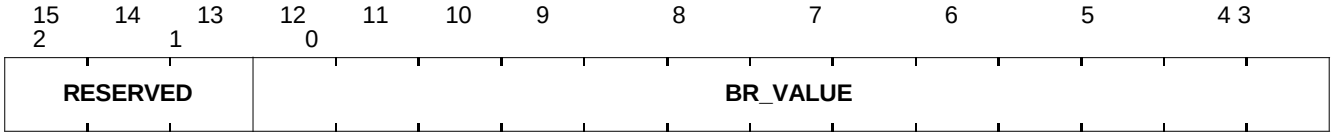
Field	Bits	Type	Description
OE	10	rwh	**Overrun Error Flag** Set by hardware on an overrun/underflow error (**OEN** = 1). Must be cleared by software.
FDE	11	rw	**Fractional Divider Enable** 0 Fractional divider disabled 1 Fractional divider enabled and used as perscaler for baudrate generator (bit BRS is 'don't care')
ODD	12	rw	**Parity Selection** 0 Even parity selected (parity bit of 1 is included in data stream on odd number of 1 and parity bit of 0 is included in data stream on even number of 1) 1 Odd parity selected (parity bit of 1 is included in data stream on even number of 1 and parity bit of 0 is included in data stream on odd number of 1)
BRS	13	rw	**Baudrate Selection** 0 Baud rate timer prescaler divide-by-2 selected 1 Baud rate timer prescaler divide-by-3 selected *Note: **BRS** is 'don't care' if **FDE** = 1 (fractional divider selected)*
LB	14	rw	**Loopback Mode Enabled** 0 Loopback Mode disabled. Standard transmit/receive Mode 1 Loopback Mode enabled
R	15	rw	**Baudrate Generator Run Control Bit** 0 Baudrate generator disabled (ASC0 inactive) 1 Baudrate generator enabled *Note: **S0BG.BR_VALUE** should only be written if **R** = 0.*

9.3.6.4 Baudrate Register

The ASC0 baudrate timer reload register contains the 13-bit reload value for the baudrate timer in Asynchronous and Synchronous Mode.

S0BG

Baudrate Timer/Reload Register **Reset value: 0000_H**



Field	Bits	Type	Description
BR_VALUE	12:0	rw	**Baudrate Timer/Reload Value** *Note: **BR_VALUE** should only be written if **S0CON.R** = 0.*
RESERVED	15:13	r	Reserved; these bits must be left at their reset values.

9.3.6.5 Fractional Divider Register

The ASC0 fractional divider register contains the 9-bit divider value for the fractional divider (Asynchronous Mode only). It is also used for reference clock generation of the autobaud detection unit.

S0FDV

Fractional Divider Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							FD_VALUE								

Field	Bits	Type	Description
FD_VALUE	8:0	rw	**Fractional Divider Register Value** **FD_VALUE** contains the 9-bit value of the fractional divider which defines the fractional divider ratio n/512 (n = 0-511). With n = 0, the fractional divider is switched off (input = output frequency, $f_{DIV} = f_{hw\_clk}$, see **Figure 124 ASC0 Baudrate Generator Circuitry in Asynchronous Modes (on Page 322)**).
RESERVED	15:9	r	Reserved; these bits must be left at their reset values.

9.3.6.6 IrDA Pulse Mode/Width Register

The ASC0 IrDA pulse mode and width register contains the 8-bit IrDA pulse width value and the IrDA pulse width mode select bit. This register is only required in the IrDA operating mode.

S0PWM

IrDA Pulse Mode/Width Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							IRPW	PW_VALUE							

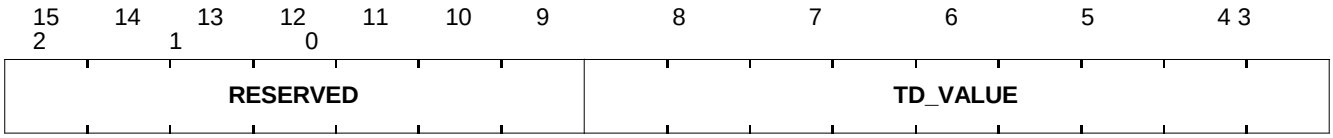
Field	Bits	Type	Description
PW_VALUE	7:0	rw	**IrDA Pulse Width Value** **PW_VALUE** is the 8-bit value n, which defines the variable pulse width of an IrDA pulse. Depending on the ASC0 input frequency f_{hw_clk}, this value can be used to adjust the IrDA pulse width to value which is not equal 3/16 bit time (e.g. 1.6 ms).
IRPW	8	rw	**IrDA Pulse Width Selection** 0 IrDA pulse width is 3/16 bit time 1 IrDA pulse width is defined by **PW_VALUE**
RESERVED	15:9	r	Reserved; these bits must be left at their reset values.

9.3.6.7 Transmitter Buffer Register

The ASC0 transmitter buffer register contains the transmit data value in Asynchronous and Synchronous Mode.

S0TBUF

Transmit Buffer Register **Reset value: 0000_H**



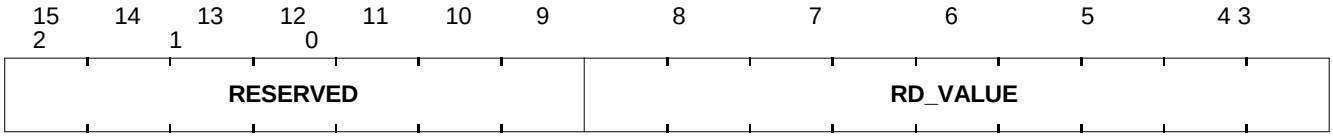
Field	Bits	Type	Description
TD_VALUE	8:0	rw	Transmit Data Register Value **TD_VALUE** contains the data to be transmitted in asynchronous and synchronous operating mode of the ASC0. Data transmission is double buffered, Therefore, a new value can be written to **TD_VALUE** before the transmission of the previous value is complete.
RESERVED	15:9	r	Reserved; these bits must be left at their reset values.

9.3.6.8 Receiver Buffer Register

The ASC0 Receiver buffer register contains the transmit data value in Asynchronous and Synchronous Modes.

S0RBUF

Receive Buffer Register **Reset value: 0000_H**



Field	Bits	Type	Description
RD_VALUE	8:0	rw	Receive Data Register Value **RD_VALUE** contains the received data bits and, depending on the selected mode, the parity bit in asynchronous and synchronous operating mode of the ASC0. In asynchronous operating mode if **S0CON.M** = 011 (7-bit data + parity), the received parity bit is written into **RD_VALUE[7]**. In asynchronous operating mode if **S0CON.M** = 111 (8-bit data + parity), the received parity bit is written into **RD_VALUE[8]**.
RESERVED	15:9	r	Reserved; these bits must be left at their reset values.

9.3.6.9 Autobaud Control Register

The autobaud control register of the ASC_P module is used to control the autobaud detection operation. It contains its general enable bit, the interrupt enable control bits, and data path control bits.

S0ABCON

Autobaud Control Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				RXINV	TXINV	ABEM		RESERVED	FC DET EN	AB DET EN	ABST EN		AURE N		ABEN

Field	Bits	Type	Description
ABEN	0	rwh	Autobaud Detection Enable 0 Autobaud detection is disabled 1 Autobaud detection is enabled *Note: **ABEN** is reset by hardware after a successful Autobaud Detection; (with the stop bit detection of the second character). Resetting **ABEN** by software if it was set aborts the Autobaud Detection.*
AUREN	1	rw	Automatic Autobaud Control of **S0CON**.REN 0 **S0CON.REN** is not affected during autobaud detection 1 **S0CON.REN** is cleared (receiver disabled) when **ABEN** and **AUREN** are set together. **S0CON.REN** is set (receiver enabled) after a successful Autobaud Detection (with the stop bit detection of the second character)
ABSTEN	2	rw	Start of Autobaud Detection Interrupt Enable 0 Start of Autobaud Detection interrupt disabled 1 Start of Autobaud Detection interrupt enabled
ABDETEN	3	rw	Autobaud Detection Interrupt Enable 0 Autobaud Detection interrupt disabled 1 Autobaud Detection interrupt enabled
FCDETEN	4	rw	First Character of Two-Byte Frame Detected Enable 0 Autobaud Detection interrupt ABDETIR becomes active after the two-byte frame recognition 1 Autobaud Detection interrupt ABDETIR becomes active after detection of the first and second byte of the two-byte frame
ABEM	9:8	rw	Autobaud Echo Mode Enable In Echo Mode the serial data at RXD is switched to TXD output. 00 Echo Mode disabled 01 Echo Mode is enabled during Autobaud Detection 10 Echo Mode is always enabled 11 Reserved; do not use this combination
TXINV	10	rw	Transmit Inverter Enable 0 Transmit inverter disabled 1 Transmit inverter enabled

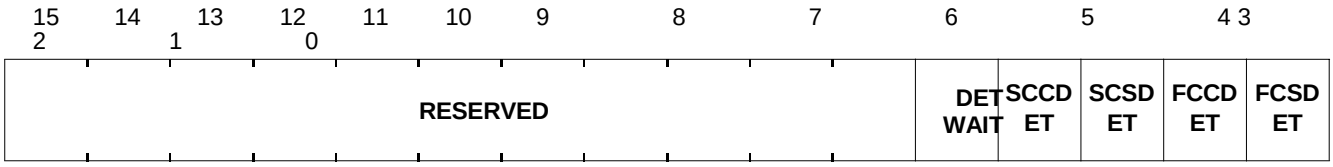
Field	Bits	Type	Description
RXINV	11	rw	Receive Inverter Enable 0 Receive inverter disabled 1 Receive inverter enabled
RESERVED	7:5, 15:12	r	Reserved; these bits must be left at their reset values.

9.3.6.10 Autobaud Status Register

The autobaud status register **S0ABSTAT** of the ASC_P module indicates the status of the autobaud detection operation.

S0ABSTAT

Autobaud Status Register **Reset value: 0000_H**



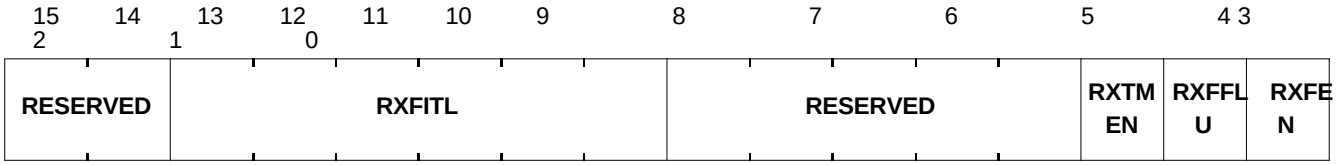
Field	Bits	Type	Description
FCSDET	0	rwh	First Character with Small Letter Detected 0 No small 'a' character detected 1 Small 'a' character detected Bit is cleared by hardware when **S0ABCON.ABEN** is set or if **FCCDET** or **SCSDET** or **SCCDET** is set. Bit can be also cleared by software.
FCCDET	1	rwh	First Character with Capital Letter Detected 0 No capital 'A' character detected 1 Capital 'A' character detected Bit is cleared by hardware when **S0ABCON.ABEN** is set or if **FCSDET** or **SCSDET** or **SCCDET** is set. Bit can be also cleared by software.
SCSDET	2	rwh	Second Character with Small Letter Detected 0 No small 't' character detected 1 Small 't' character detected Bit is cleared by hardware when **S0ABCON.ABEN** is set or if **FCSDET** or **FCCDET** or **SCCDET** is set. Bit can be also cleared by software.
SCCDET	3	rwh	Second Character with Capital Letter Detected 0 No capital 'T' character detected 1 Capital 'T' character detected Bit is cleared by hardware when **S0ABCON.ABEN** is set or if **FCSDET** or **FCCDET** or **SCSDET** is set. Bit can be also cleared by software.
DETWAIT	4	rwh	Autobaud Detection is Waiting 0 Either character 'a', 'A', 't', or 'T' has been detected. 1 The autobaud detection unit waits for the first 'a' or 'A' Bit is cleared when either **FCSDET** or **FCCDET** is set ('a' or 'A' detected). Bit can be also cleared by software. **DETWAIT** is set by hardware when **S0ABCON.ABEN** is set.
RESERVED	15:5	r	Reserved for future use; these bits must be left at their reset values.

*Note: **SCSDET** or **SCCDET** are set when the second character has been recognized. **S0ABCON.ABEN** is reset and **ABDETIR** set after **SCSDET** or **SCCDET** have seen set.*

9.3.6.11 Receive FIFO Control Register

S0RXFCON

Receive FIFO Control Register **Reset value: 0100_H**



Field	Bits	Type	Description
RXFEN	0	rw	Receive FIFO Enable 0 Receive FIFO is disabled 1 Receive FIFO is enabled *Note: Resetting **RXFEN** automatically flushes the receive FIFO.*
RXFFLU	1	rw	Receive FIFO Flush 0 No operation 1 Receive FIFO is flushed *Note: Setting **RXFFLU** clears bit field **S0FSTAT**.RXFFL.* ***RXFFLU** is always read as 0.*
RXTMEN	2	rw	Receive FIFO Transparent Mode Enable 0 Receive FIFO Transparent Mode is disabled 1 Receive FIFO Transparent Mode is enabled *Note: This bit is don't care if the receive FIFO is disabled (**RXFEN** = 0).*
RXFITL	13:8	rw	Receive FIFO Interrupt Trigger Level Defines a receive FIFO interrupt trigger level. A receive interrupt request (RIR) is always generated after the reception of a byte when the filling level of the receive FIFO is equal to or greater **RXFITL** 000000Reserved. Do not use this combination 000001Interrupt trigger level is set to one 000010Interrupt trigger level is set to two ... 011111Interrupt trigger level is set to thirty one 100000Interrupt trigger level is set to thirty two *Note: In Transparent Mode this bit field is don't care.* *Note: Combinations defining an interrupt trigger level greater then the configured FIFO size should not be used*
RESERVED	7:3, 15:14	r	Reserved; these bits must be left at their reset values.

Notes

1. *After a successful autobaud detection sequence (if implemented), the RXFIFO must be flushed before data is received.*
2. *For smaller FIFO implementations than 64 stages also less bits in bit field **S0RXFCON.RXFITL** are implemented. The implemented width of bit field **RXFITL** depends on the size of the receive FIFO RXFIFO.*

*Number of FIFO states = 2^x, x = number of bit in bit field **S0RXFCON.RXFITL**. Therefore, the bit field is located at [**S0RXFCON**.8+x:**S0RXFCON**.8].*

9.3.6.11.1 Transmit FIFO Control Register

S0TXFCON

Transmit FIFO Control Register **Reset value: 0100_H**

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0			
RESERVED	TXFITL	RESERVED	TXTMEN / TXFFLU / TXFEN

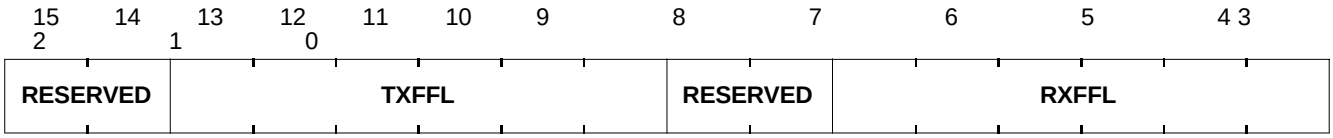
Field	Bits	Type	Description
TXFEN	0	rw	Transmit FIFO Enable 0 Transmit FIFO is disabled 1 Transmit FIFO is enabled *Note: Resetting **TXFEN** automatically flushes the transmit FIFO.*
TXFFLU	1	rw	Transmit FIFO Flush 0 No operation 1 Transmit FIFO is flashed *Note: Setting **TXFFLU** clears bit field **S0FSTAT.TXFFL**.* ***TXFFLU** is always read as 0.*
TXTMEN	2	rw	Transmit FIFO Transparent Mode Enable 0 Transmit FIFO Transparent Mode is disabled 1 Transmit FIFO Transparent Mode is enabled *Note: This bit is don't care if the receive FIFO is disabled (**TXFEN** = 0).*
TXFITL	13:8	rw	Transmit FIFO Interrupt Trigger Level Defines a transmit FIFO interrupt trigger level. A transmit interrupt request (TIR) is always generated after the transfer of a byte when the filling level of the transmit FIFO is equal to or lower **TXFITL** 000000Reserved. Do not use this combination 000001Interrupt trigger level is set to one 000010Interrupt trigger level is set to two ... 011111Interrupt trigger level is set to thirty one 100000Interrupt trigger level is set to thirty two *Note: In Transparent Mode this bit field is don't care.* *Note: Combinations defining an interrupt trigger level greater then the configured FIFO size should not be used*
RESERVED	7:3, 15:14	r	Reserved; these bits must be left at their reset values.

*Note: For smaller FIFO implementations than 64 stages also less bits in bit field **S0TXFCON.TXFITL** are implemented. The implemented width of bit field **TXFITL** depends on the size of the transmit FIFO TXFIFO. Number of FIFO states = 2^x, x = number of bit in bit field **TXFITL**. Therefore, the bit field is located at [**S0TXFCON**.8+x:**S0TXFCON**.8].*

9.3.6.11.2 FIFO Status Register

S0FSTAT

FIFO Status Register **Reset value: 0000_H**



Field	Bits	Type	Description
RXFFL	5:0	rh	Receive FIFO Filling Level 000 000:Receive FIFO is filled with zero bytes 000 001:Receive FIFO is filled with one byte ... 011 111:Receive FIFO is filled with thirty one bytes 100 000:Receive FIFO is filled with thirty two bytes *Note: **RXFFL** is cleared after a receive FIFO flush operation.*
TXFFL	13:8	rh	Transmit FIFO Filling Level 000 000: Transmit FIFO is filled with zero bytes 000 001: Transmit FIFO is filled with one byte ... 011 111: Transmit FIFO is filled with thirty one bytes 100 000: Transmit FIFO is filled with thirty two bytes *Note: **TXFFL** is cleared after a receive FIFO flush operation.*
RESERVED	7:6, 15:14	r	Reserved; these bits must be left at their reset values.

Notes

1. *For smaller FIFO implementations than 64 stages also less bits in bit field **S0FSTAT.TXFFL** are implemented. The implemented width of bit field **TXFFL** depends on the size of the transmit FIFO TXFIFO. Number of FIFO states = 2^x, x = number of bit in bit field **TXFFL**. Therefore, the bit field is located at [**S0FSTAT.8**+x:**S0FSTAT.8**].*
2. *For smaller FIFO implementations than 64 stages also less bits in bit field **S0FSTAT.RXFFL** are implemented. The implemented width of bit field **RXFFL** depends on the size of the transmit FIFO RXFIFO. Number of FIFO states = 2^x, x = number of bit in bit field **RXFFL**. Therefore, the bit field is located at [**S0FSTAT.x**:**S0FSTAT.0**].*

9.3.6.11.3 Flowcontrol Control Register

S0FCCON

Flowcontrol Control Register **Reset value: 0000_H**

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0						
RESERVED	RTSTL	RESERVED	RTS	RESERVED	CTSEN	RTSEN

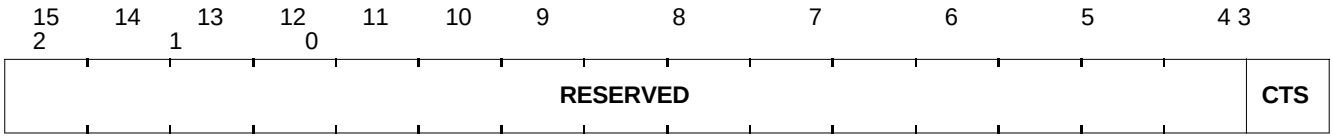
Field	Bits	Type	Description
RTSEN	0	rw	RTS Enable 0: Disable RTS HW Flowcontrol 1: Enable RTS HW Flowcontrol
CTSEN	1	rw	CTS Enable 0: Disable CTS HW Flowcontrol 1: Enable CTS HW Flowcontrol
RTS	4	rw	Request To Send Control Bit 0: RTS is inactive (RTS_N = '1') 1: RTS is active (RTS_N = '0') The RTS_N pin is controlled by this bit only if hardware flow control is disabled (**RTSEN** = 0).
RTSTL	13:8	rw	RTS Receive FIFO Trigger Level 000 000: Receive FIFO is filled with zero byte. 000 001: Receive FIFO is filled with one byte. ... 011 111: Receive FIFO is filled with thirty one bytes. 100 000: Receive FIFO is filled with thirty two bytes. *Note: **RTSTL** is cleared after a receive FIFO flush operation.* *Note: Combinations defining a RTS trigger level greater than the configured FIFO size should not be used.*
RESERVED	3:2, 7:5, 15:14	r	Reserved; these bits must be left at their reset values.

*Note: For smaller FIFO implementations than 64 stages also less bits in bit field **S0FCCON.RTSTL** are implemented. The implemented width of bit field **RTSTL** depends on the size of the receive FIFO RXFIFO. Number of FIFO states = 2^x, x = number of bit in bit field **RTSTL**. Therefor the bit field is located at [**S0FCCON**.8+x:**S0FCCON**.8].*

9.3.6.11.4 Flowcontrol Status Register

S0FCSTAT

Flowcontrol Status Register **Reset value: 0000_H**



Field	Bits	Type	Description
CTS	0	rh	CTS Status 0 CTS inactive (CTS_N = 1) 1 CTS active (CTS_N = 0)
RESERVED	15:1	r	Reserved; these bits must be left at their reset values.

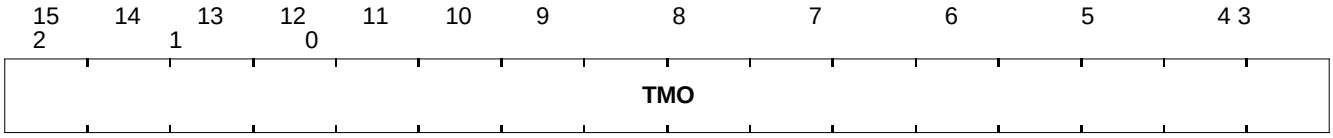
Note: The reset value depends on the input CTS_N.

9.3.6.11.5 RX Timeout Register

The timeout control register contains the 16-bit reload value for the timeout detection timer.

S0TMO

RX Timeout Register **Reset value: 0000_H**



Field	Bits	Type	Description
TMO	15:0	rw	Timeout Detection Timer Reload Value 0: timeout detection disabled 1_H..$FFFF_H$: Timeout after <**TMO**> shift clock cycles

9.4 CAPCOM 1 and 2

System Integration

- Supply domain: VDD_LD1
- Chip internal interfaces:
  - Clock domain: Refer to **Section 7.2.1.3 Sub-System Clocks and Enables (on Page 67)** and see **Figure 18 Clock Enable (on Page 68)**.
  - Bus domain: PD-Bus
- Interrupt sources:
  - Monitor Pins: refer to **Section 9.7.10 Internal Signal Monitoring (on Page 435)**.

9.4.1 Introduction

E-GOLDvoice provides two CAPCOM units, which can either:

- Capture the contents of a timer on specific internal or external events
- Compare a timer content with given values and modify output signals if there is a match.

With this mechanism it supports generation and control of timing sequences on up to 7 or 8 channels with a minimum of software intervention.

Features of both CAPCOM Units

- Module of timers registers and comparators for high speed pulse and waveform generation or time measurement
- Programmable clock with multiple sources
- 154 ns maximum resolution @ 52 MHz master clock (hw_clk) if staggered mode enabled
- 19.2 ns maximum resolution @ 52 MHz master clock (hw_clk) if staggered mode disabled
- Double register compare function
- Primary clock prescaler
- Additional output register
- Single event mode.

9.4.1.1 Features of the individual E-GOLDvoice CAPCOM Units

Features for CAPCOM1 Unit

- One 16-bit timer with reload registers
- 7 registers individually configurable for capture or compare function
- 8 interrupts: 7 capture compare interrupts and one timer interrupt
- Up to 7 software timers.

Features for CAPCOM2 Unit

- Two 16-bit timers with reload registers
- 8 registers individually configurable for capture or compare function
- 10 interrupts: 8 capture compare interrupts and two timer interrupts
- Up to 8 software timers.

Differences between CAPCOM1 and CAPCOM2

The functions of the standard CAPCOM are described in **Section 9.4.2 Operational Overview** and subsequent sections.

In E-GOLDvoice some CAPCOM features are restricted:

- **CAPCOM2** has the standard CAPCOM functions described in the rest of this section.
- **CAPCOM1** is the same except that:
  - Only one 16-bit timer with reload registers is supported.
    T0 is not available so Timer0 of CAPCOM1 is not usable (Respective Timer Mode and Counter Mode associated to T0 are not available).
  - Only 7 channels are supported.
    CC3IO pin (Channel 3) of CAPCOM1 is not available. So associated Capture Register CC3 is not usable and the Double Register Compare mode CC3-CC7 is also not usable.

CAPCOM restrictions are handled at the pin level. Both CAPCOM1/2 blocks are internally identical, but only the pins mentioned below are available at package level (see **Chapter 3 Pin Descriptions**):

• Pins available for CAPCOM1:

- CC0IO /CC0IOb / CC1IOa / CC2IO / CC4IO / CC5IO / CC6IOa / CC6IOb / CC7IO

• Pins available for CAPCOM2::

- CC16IOa /CC16IOb / CC17IO / CC18IOa / CC18IOb / CC19IO / CC20IOb / CC21IO / CC22IOa / CC22IOb / CC23IO / T7IN

*Note: Details about the available CAPCOM1/2 package level pins and how they are selected are described in **Table 79 CAPCOM 1 Input Signal Selection** and **Table 80 CAPCOM 2 Input Signal Selection**.*

9.4.2 Operational Overview

Note: in this section's figures the x labels are used for timers and the y and z labels for channels.

From the programmer's point of view, the term 'CAPCOM Unit' refers to a set of registers which are associated with this peripheral, including the port pins which may be used for alternate input/output functions including their direction control bits (refer to **Table 77 CAPCOM Register Summary (on Page 362)**).

Figure 131 and **Figure 131** show the interface diagrams of the CAPCOM Units.

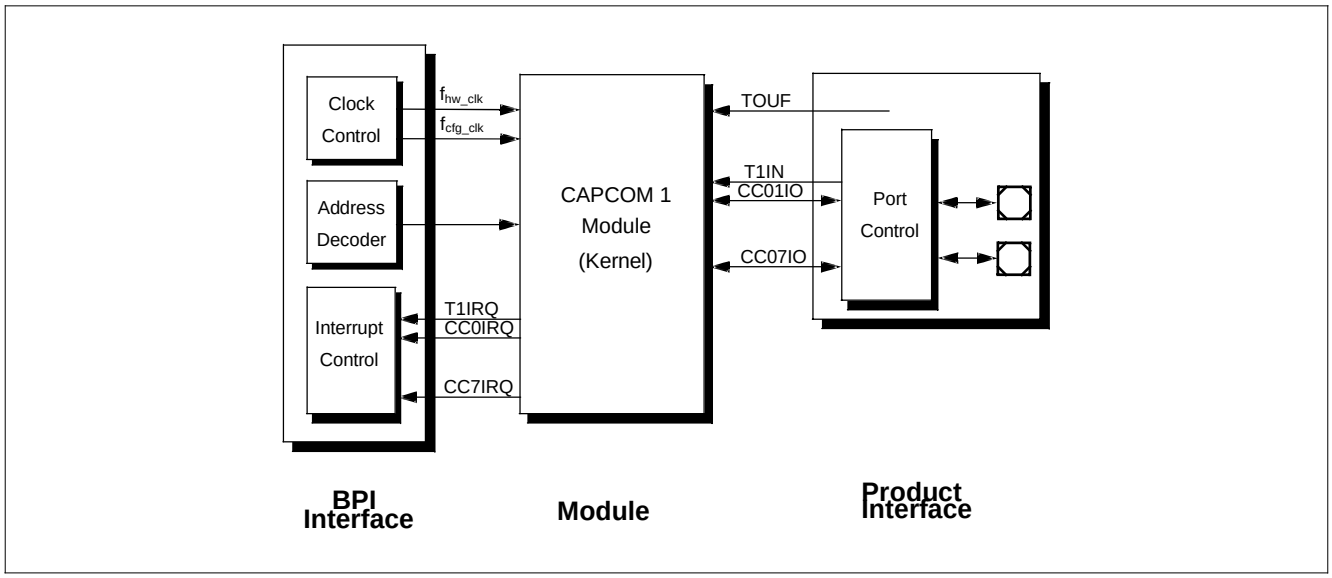


Figure 130 CAPCOM 1 Interface Diagram

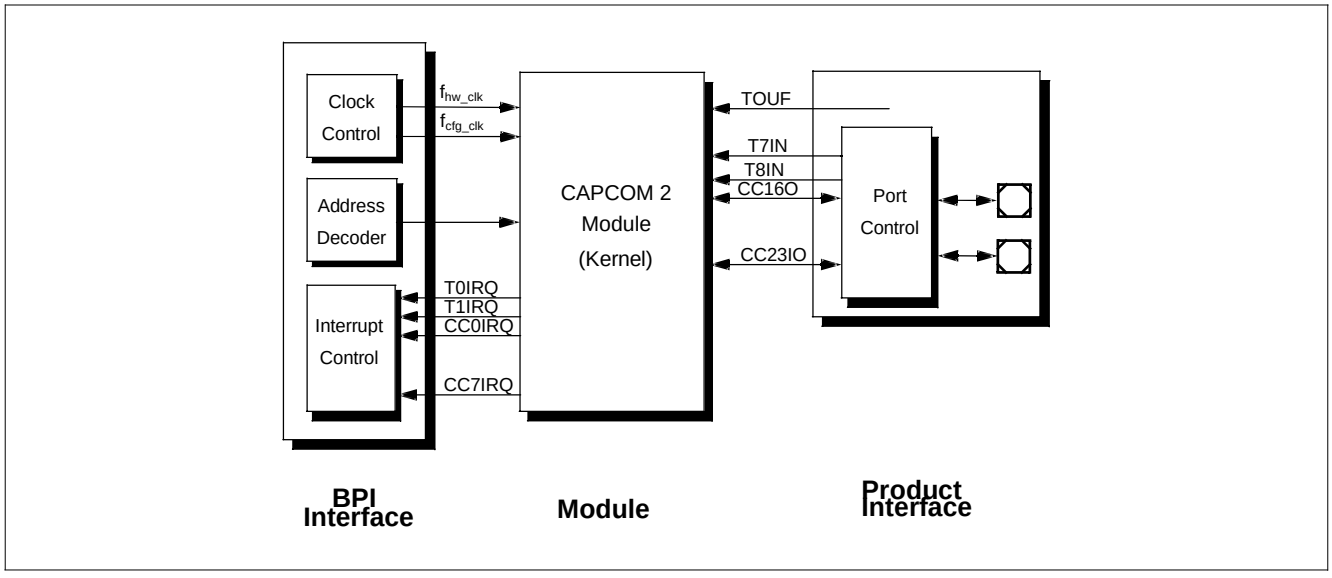


Figure 131 CAPCOM 2 Interface Diagram

Note: If required, it is also possible to connect CCxIO (x = 0 to 7 or 16 to 23) channels directly with the ports, without going though a port control module.

9.4.3 Functional Overview

A CAPCOM unit is typically used to handle high speed IO tasks such as pulse and waveform generation, pulse width modulation, or recording of the time when a specific event occurs. It also allows the implementation of up to eight software timers. The highest resolution of a CAPCOM Unit is 154 ns @ 52 MHz master clock (hw_clk) if staggered mode is enabled or 19.2 ns maximum resolution @ 52 MHz master clock (hw_clk) if staggered mode is disabled.

Each CAPCOM unit consists of two 16-bit timers, each with its own reload register and dual purpose 16-bit capture/compare registers.

The input clock for the CAPCOM timers is programmable to several prescaled values of the master clock (hw_clk), or it can be derived from an overflow/underflow (TOUF) of an external timer. **T0** and **T7** may also operate in counter mode (from an external input) where they can be clocked by external events (TxIN).

Note: The external timer is usually the Timer 6 of the General Purpose Timer unit.

Each capture/compare register may be programmed individually for capture or compare function, and each register may be allocated to either timer of the associated unit. Each capture/compare register has one signal associated with it which serves as an input signal for the capture function or as an output signal for the compare function. The capture function causes the current timer contents to be latched into the respective capture/compare register triggered by an event (transition) on its associated signal. The compare function may cause an output signal transition on that signal whose associated capture/compare register matches the current timer contents. Specific interrupt requests are generated upon each capture/compare event or upon timer overflow.

Figure 132 shows the basic structure of CAPCOM unit 1 (CAPCOM unit 1 is identical).

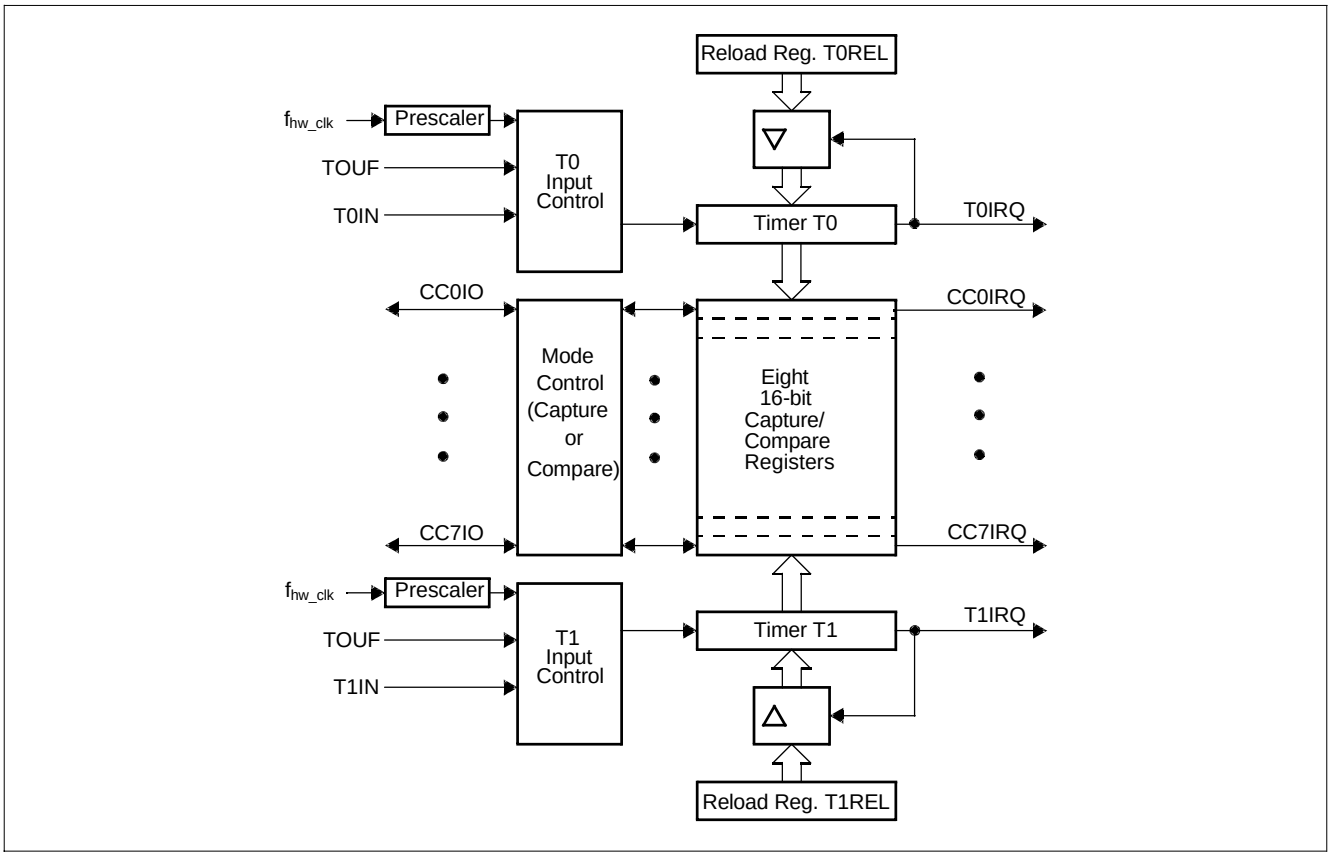


Figure 132 CAPCOM Unit 1 Block Diagram

9.4.3.1 The Timers

The primary use of the timers **T0** and **T1** (**T7** and **T8**) is to provide two independent time bases with 154 ns maximum resolution @ 52 MHz master clock (hw_clk) and staggered mode enabled or 19.2 ns maximum resolution @ 52 MHz master clock (hw_clk) and staggered mode disabled for the capture/compare registers of a unit, but they may also be used independent of the capture/compare registers.

The basic structure of the two timers is identical (see **Figure 133**).

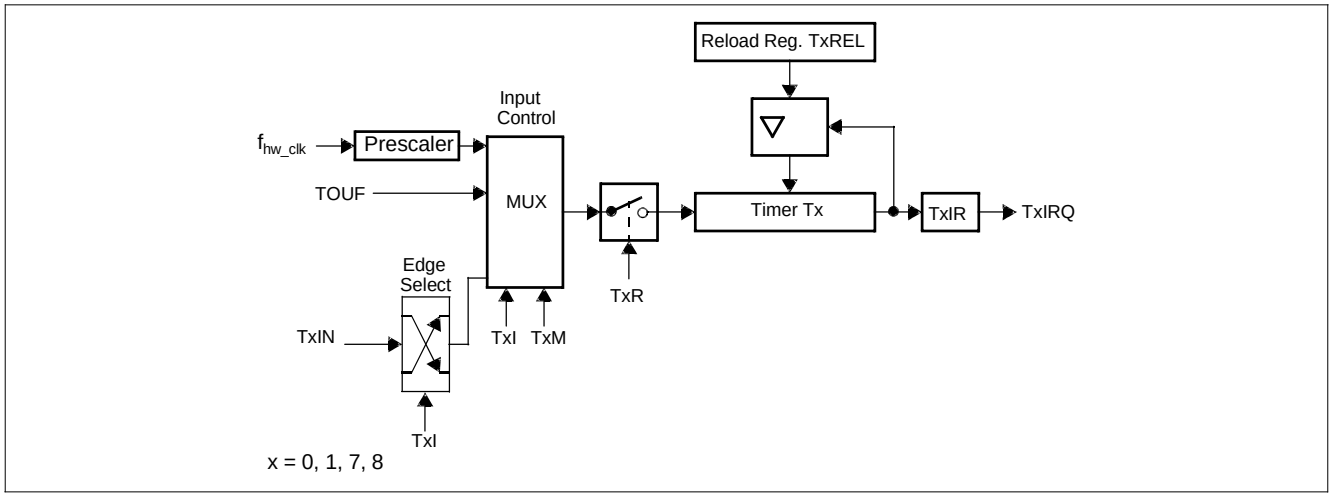


Figure 133 Block Diagram of Timers

The functions of the CAPCOM timers are controlled via the bit addressable control registers **T01CON** (for CAPCOM 1) and **T78CON** (for CAPCOM 2). The high-byte of **TxxCON** controls **T1** or **T7**, the low-byte of **TxxCON** controls **T0** or **T8**. The control options are identical for all timers.

The timer run flags **TxxCON.T0R** and **TxxCON.T1R** allow the starting and stopping of the timers. The following description of the timer modes and operation always applies to the enabled state of the timers, that is, the respective run flag is assumed to be set.

In all modes, the timers always count upward. The current timer values are accessible form the MCU in the timer registers Tx, which are non bit addressable registers. When the MCU writes to a register Tx in the state immediately before the respective timer increment or reload is to be performed, the MCU write operation has priority and the increment or reload is disabled to guarantee correct timer operation.

Timer Mode

The bits **TxxCON.TxM** select between the timer or counter mode for the respective timer. In timer mode (**TxxCON.TxM** set), the input clock for a timer is derived from the internal hw_clk clock divided by a programmable prescaler. The different options for the prescaler are selected separately for each timer by the bit field **TxxCON.TxI**.

The input frequencies f_{Tx} for Tx are determined as a function of the hw_clk clock as follows, where <TxI> represents the contents of the bit field **TxxCON.TxI**:

$$f_{Tx} = \frac{f_{hw\_clk}}{2^{(<TxI>+3)}} \quad \text{Staggered Mode enabled}$$

$$f_{Tx} = \frac{f_{hw\_clk}}{2^{<TxI>}} \quad \text{Staggered Mode disabled}$$

(63)

When a timer overflows from $FFFF_H$ to 0000_H it is reloaded with the value stored in its respective reload register TxREL. The reload value determines the period P_{Tx} between two consecutive overflows of Tx as follows

$$P_{Tx} = \frac{(2^{16} - \langle TxREL \rangle) * 2^{(\langle TxI \rangle + 3)}}{f_{hw\_clk}}$$

Staggered Mode enabled

$$P_{Tx} = \frac{(2^{16} - \langle TxREL \rangle) * 2^{\langle TxI \rangle}}{f_{hw\_clk}}$$

Staggered Mode disabled

Examples for timer input frequencies, resolution, and periods which result from the selected prescaler option in **TxI** when using a 52 MHz hw_clk clock are listed in **Table 75**. The numbers for the timer periods are based on a reload value of 0000_H.

Note: Some numbers may be rounded.

Table 75 Timing Examples

f_{hw_clk} = 52 MHz Staggered Mode enabled	Timer Input Selection TxI							
	000_B	**001_B**	**010_B**	**011_B**	**100_B**	**101_B**	**110_B**	**111_B**
Prescaler for f_{hw_clk}	8	16	32	64	128	256	512	1024
Input Frequency	6.5 MHz	3.25 MHz	1.625 MHz	812.5 kHz	406.25 kHz	203.125 kHz	101.562 kHz	50.781 kHz
Resolution	154 ns	308 ns	615 ns	1.23 µs	2.46 µs	4.92 µs	9.85 µs	19.7 µs
Period	10.08 ms	20.164 ms	40.3 ms	80.6 ms	161.3 ms	322.6 ms	645.27 ms	1.29 s
f_{hw_clk} = 52 MHz Staggered Mode disabled	Timer Input Selection TxI							
	000_B	001_B	010_B	011_B	100_B	101_B	110_B	111_B
Prescaler for f_{hw_clk}	1	2	4	8	16	32	64	128
Input Frequency	52 MHz	26 MHz	13 MHz	6.50 MHz	3.25 MHz	1.625 MHz	0.8125 MHz	0.406 MHz
Resolution	19.2 ns	38.5 ns	76.9 ns	154 ns	308 ns	615 ns	1.23 µs	2.46 µs
Period	1.26 ms	2.52 ms	5.04 ms	10.08 ms	20.16 ms	40.32 ms	80.65 ms	161.3 ms

After a timer has been started by setting its run flag (**TxxCON.TxR**), the first increment occurs within the time interval which is defined by the selected timer resolution. All further increments occur exactly after the time defined by the timer resolution.
When both timers of a CAPCOM unit are to be incremented or reloaded at the same time **T0** or **T7** is always serviced one hw_clk clock before **T1** or **T8**.

Counter Mode

The bits **TxxCON.TxM** select between timer or counter mode for the respective timer. In Counter mode (**TxM** = 1) the input clock for a timer can be derived from the overflows/underflows of timer_in.

In addition, the timers can be clocked by external events if a signal is connected to that module input. Either a positive, a negative, or both a positive and a negative transition can be selected to cause an increment of **T0** or **T7**.

*Note: When **T1** or **T8** is programmed to run in counter mode, bit field **TxxCON.T[1,8]I** is used to enable the overflows/underflows of the GPT Timer T6 as the count source. This is the only option for **T1** or **T8**, and it is selected by the combination **T[1,8]I** = $X00_B$. When bit field **T[1,8]I** is programmed to any other combination, the respective timer (**T1** or **T8**) stops.*

When **T0** or **T7** is programmed to run in counter mode, bit field **TxxCON.T*[0,7]*I** is used to select the count source and transition (if the source is the input pin) which should cause a count trigger (refer to **TxxCON** for the possible selections).

Note: To use pin T0IN as external count input pin, the respective port pin must be configured as input, that is, the corresponding direction control bit (DP3.0) must be cleared (0).
If the respective port pin is configured as output, the associated timer may be clocked by modifying the port output latch P3.0 via software, for example, for testing purposes.

The maximum external input frequency to **T0** in counter mode is $f_{hw\_clk}/16$ (3.25 MHz @ 52 MHz f_{hw_clk}). To ensure that a signal transition is properly recognized at the timer input, an external count input signal should be held for at least eight hw_clk clock cycles before it changes its level again. The incremented count value appears in **T0** within eight hw_clk clock cycles after the signal transition at T0IN.

Reload

A reload of a timer with the 16-bit value stored in its associated reload register in both modes is performed each time a timer would overflow from $FFFF_H$ to 0000_H. In this case the timer does not wrap around to 0000_H, but rather is reloaded with the contents of the respective reload register TxREL. The timer then resumes incrementing starting from the reloaded value.

The reload registers TxREL are not bit addressable.

9.4.3.2 Timer Interrupt

Upon a timer overflow the corresponding timer interrupt request flag TxIR for the respective timer will be set. This flag can be used to generate an interrupt or trigger a interrupt controller service request, when enabled by the respective interrupt enable bit TxIE.

Each timer has its own bit addressable interrupt control register (TxIC) and its own interrupt vector (TxIRQ). The organization of the interrupt control registers TxIC is identical with the other interrupt control registers.

9.4.3.3 Capture/Compare Registers

The 16-bit capture/compare registers **CCy** are used as data registers for capture or compare operations with respect to two timers in each CAPCOM unit. The capture/ compare registers are not bit addressable.

Each of the registers **CCy** may be individually programmed for capture mode or one of 4 different compare modes, and may be allocated individually to one of the two timers of the respective CAPCOM unit (**CC0** through **CC7** for CAPCOM 1; **CC16** through **CC23** for CAPCOM 2). A special combination of compare modes additionally allows the implementation of a 'double-register' compare mode. When capture or compare operation is disabled for one of the **CCy** registers, it may be used for general purpose variable storage.

The functions of the capture/compare registers are controlled by four bit-addressable mode control registers **CCMz** (**CCM0** and **CCM1** for CAPCOM 1; **CCM4** and **CCM5** for CAPCOM 2). Each register contains bits for mode selection and timer allocation of four capture/compare registers.

9.4.3.4 Capture Mode

In response to an external event the content of the associated timer (**T0**, depending on the state of the allocation control bit **CCMz.ACCy**) is latched into the respective capture register **CCy**. The external event causing a capture can be programmed to be either a positive, a negative, or both a positive or a negative transition at the respective external input pin CCyIO.

The triggering transition is selected by the mode bits **CCMz.MODy** in the respective mode control register. In any case, the event causing a capture will also set the respective interrupt request flag CCyIR, which can cause an interrupt or an interrupt controller service request when enabled.

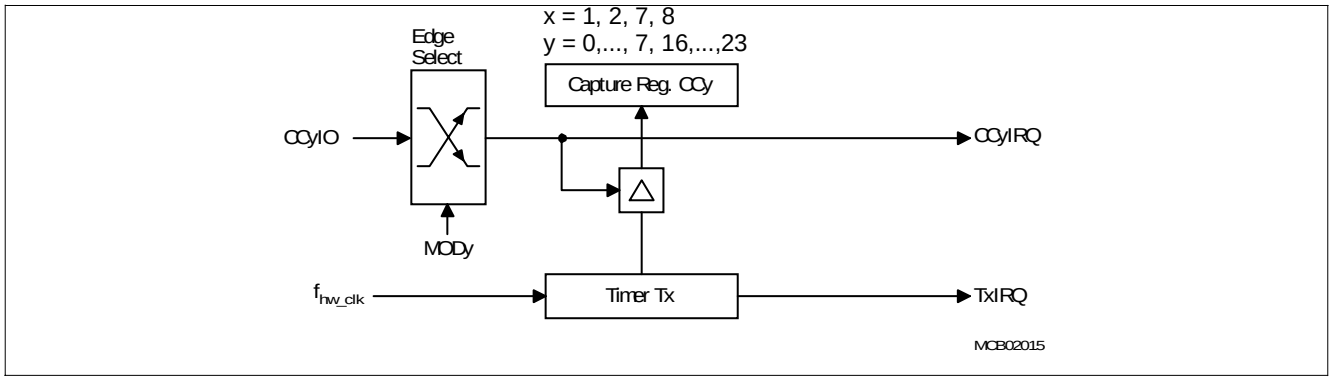


Figure 134 Capture Mode Block Diagram

*Note: In order to use the respective port pin as external capture input pin CCyIO for capture register **CCy**, this port pin must be configured as input, that is, the corresponding direction control bit must be cleared.*

To ensure that a signal transition is properly recognized, an external capture input signal should be held for at least eight hw_clk clock cycles before it changes its level.

During these eight hw_clk clock cycles the capture input signals are scanned sequentially. When a timer is modified or incremented during this process, the new timer contents will already be captured for the remaining capture registers within the current scanning sequence.

Note: If pin CCyIO is configured as output, the capture function may be triggered by modifying the corresponding port output latch via software, for example, for testing purposes.

9.4.3.5 Compare Modes

The compare modes allow triggering of events (interrupts and/or output signal transitions) with minimum software overhead. In all compare modes, the 16-bit value stored in compare register **CCy** (in the following also referred to as 'compare value') is continuously compared with the contents of the allocated timer (Tx). If the current timer contents match the compare value, an appropriate output signal, which is based on the selected compare mode, can be generated at the corresponding output pin CCyIO and the associated interrupt request flag CCyIR is set, which can generate an interrupt request (if enabled).

As for capture mode, the compare registers are also processed sequentially during compare mode. When any two compare registers are programmed to the same compare value, their corresponding interrupt request flags will be set and the selected output signals will be generated within eight hw_clk clock cycles after the allocated timer is incremented to the compare value. Further compare events on the same compare value are disabled until the timer is incremented again or written to by software. After a reset, compare events for register **CCy** only becomes enabled if the allocated timer has been incremented or written to by software and one of the compare modes described in the following has been selected for this register.

The different compare modes which can be programmed for a given compare register **CCy** are selected by the mode control field **CCMz.MODy** in the associated capture/compare mode control register. In the following, each of the compare modes, including the special 'double-register' mode, is discussed in detail.

Compare Mode 0

This is an interrupt-only mode which can be used for software timing purposes. Compare Mode 0 is selected for a given compare register **CCy** by setting bit field **CCMz.MODy** of the corresponding mode control register to 100_B. In this mode, the interrupt request flag CCyIR is set each time a match is detected between the content of compare register **CCy** and the allocated timer. Several of these compare events are possible within a single timer period, when the compare value in register **CCy** is updated during the timer period. The corresponding port signal CCyIO is not affected by compare events in this mode and can be used as general purpose IO.

*Note: If compare mode 0 is programmed for one of the registers CC...CC, the double-register compare mode may become enabled for this register if the corresponding bank 1 register is programmed to compare mode 1 (refer to **Section 9.4.3.6 Double-Register Compare Mode (on Page 357)**).*

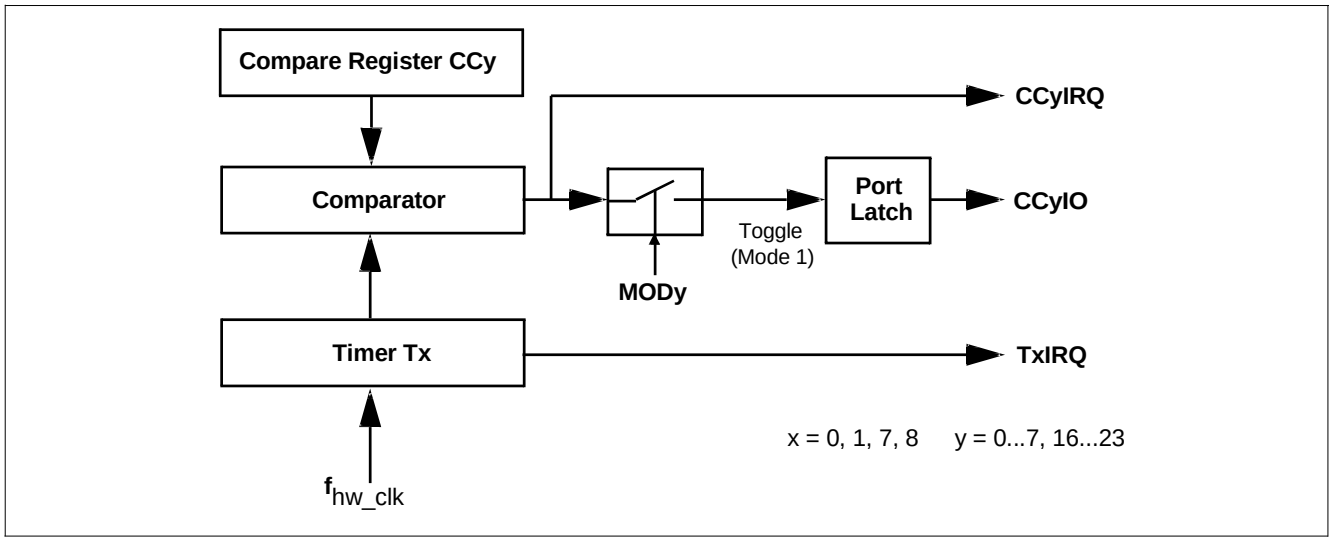
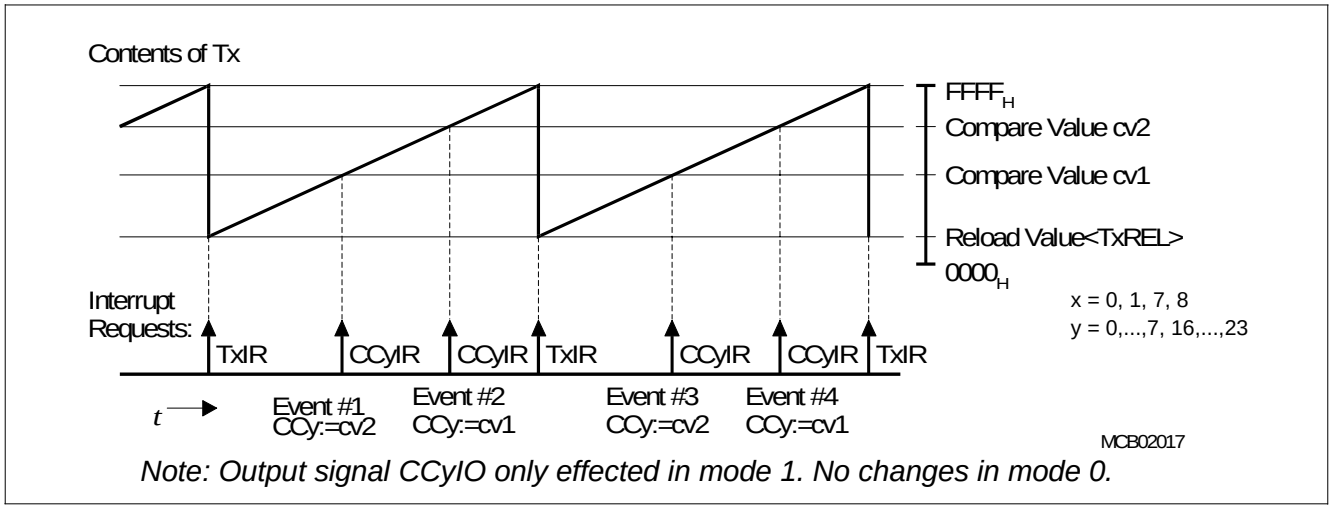


Figure 135 Compare Mode 0 and 1 Block Diagram

Note: The OUT latch, the port latch, and the pin remain unaffected in compare mode 0.

In the **Figure 136** example the compare value in register **CCy** is modified from cv1 to cv2 after compare events #1 and #3, and from cv2 to cv1 after events #2 and #4, etc. This results in periodic interrupt requests from timer Tx, and in interrupt requests from register **CCy** which occur at the time specified by the user through cv1 and cv2.



Note: Output signal CCyIO only effected in mode 1. No changes in mode 0.

Figure 136 Timing Example for Compare Modes 0 and 1

Compare Mode 1

Compare Mode 1 is selected for register **CCy** by setting bit field **CCMz.MODy** of the corresponding mode control register to 101_B.

When a match between the content of the allocated timer and the compare value in register **CCy** is detected in this mode, interrupt request flag CCyIR is set, and in addition the corresponding output signal CCyIO (alternate port output function) is toggled. For this purpose, the state of the respective port output latch (not the signal) is read, inverted, and then written back to the output latch.

Compare Mode 1 allows several compare events within a single timer period. An overflow of the allocated timer has no effect on the output signal nor does it disable or enable further compare events.

To use the respective port signal as compare signal output CCyIO for compare register **CCy** in Compare Mode 1, this port signal must be configured as output, that is, the corresponding direction control bit must be set. With this configuration, the initial state of the output signal can be programmed or its state can be modified at any time by writing to the port output latch.

In Compare Mode 1 the port latch is toggled upon each compare event (see **Figure 136** above).

Notes

1. *If Compare Mode 1 is programmed for one of the registers **CC0**...**CC3** (**CC16**...**CC23**) or the double-register compare mode may become enabled for this register if the corresponding bank 1 register is programmed to Compare Mode 0 (refer to **Section 9.4.3.6 Double-Register Compare Mode (on Page 357)**).*
2. *If the port output latch is written to by software at the same time it would be altered by a compare event, the software write has priority. In this case the hardware-triggered change does not become effective.*

Compare Mode 2

Compare Mode 2 is an interrupt-only mode similar to Compare Mode 0, but only one interrupt request per timer period is generated. Compare Mode 2 is selected for register **CCy** by setting bit field **CCMz.MODy** of the corresponding mode control register to 110_B.

When a match is detected in Compare Mode 2 for the first time within a timer period, the interrupt request flag CCyIR is set. The corresponding port 2 signal is not affected and can be used for general purpose IO. However, after the first match has been detected in this mode all further compare events within the same timer period are disabled for compare register **CCy** until the allocated timer overflows. This means, that after the first match, even when the compare register is reloaded with a value higher than the current timer value, no compare event occurs until the next timer period.

In the **Figure 138** example the compare value in register **CCy** is modified from cv1 to cv2 after compare event #1. Compare event #2, however, does not occur until the next period of timer Tx.

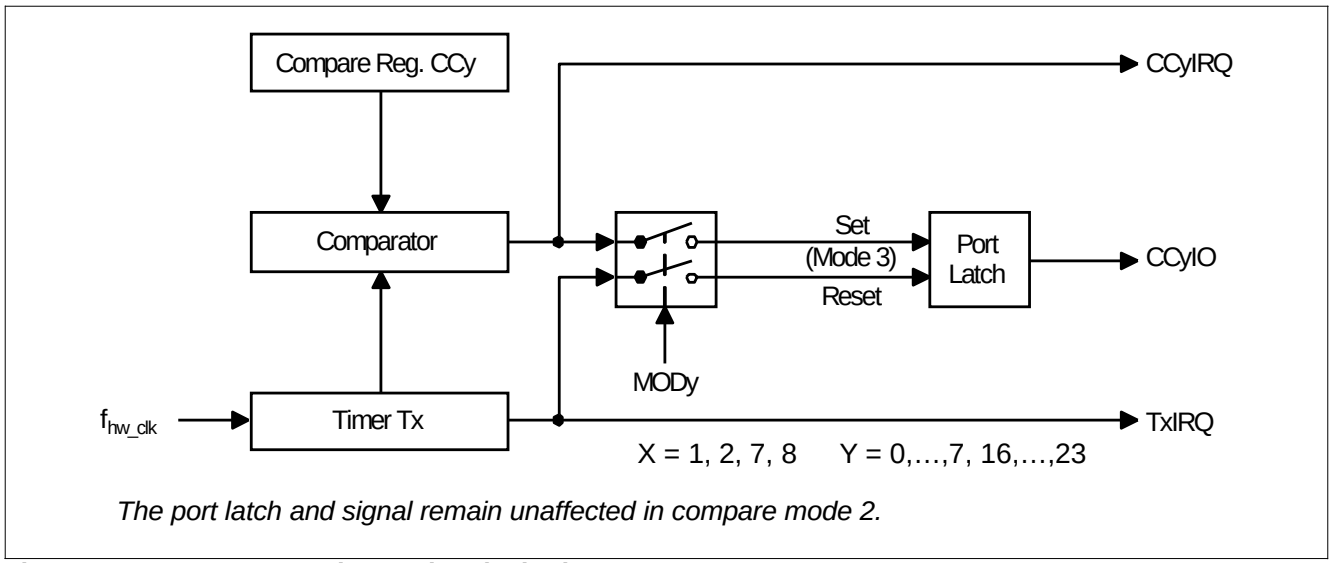
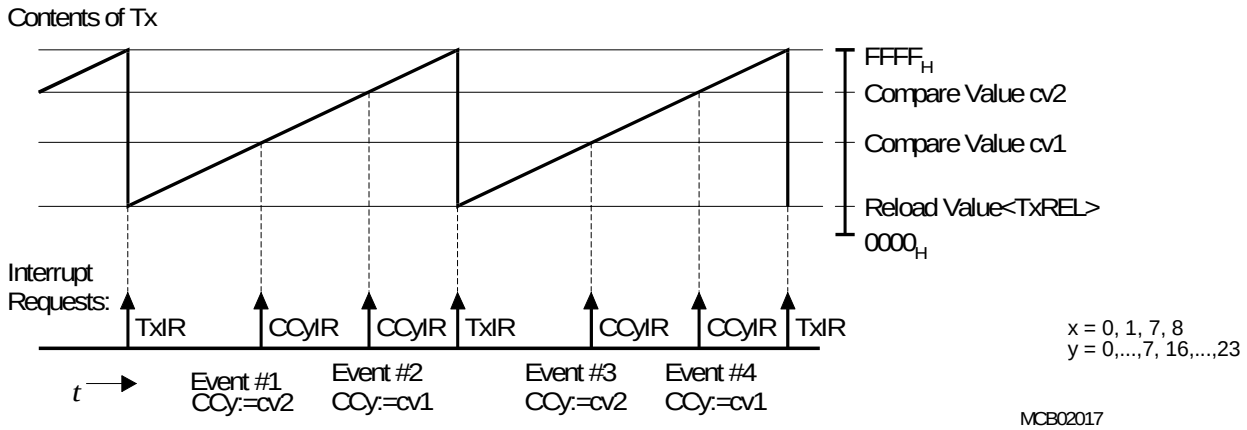


Figure 137 Compare Mode 2 and 3 Block Diagram



Note: Only output signal CCyIO is effected in mode 3. There are no changes in mode 2.

Figure 138 Timing Example for Compare Modes 2 and 3

Compare Mode 3

Compare Mode 3 is selected for register **CCy** by setting bit field **CCMz.MODy** of the corresponding mode control register to 111_B. In Compare Mode 3 only one compare event is generated per timer period.

When the first match within the timer period is detected the interrupt request flag CCyIR is set and the output pin CCyIO (alternate port function) is set. The signal is cleared when the allocated timer overflows.

If a match was found for register **CCy** in this mode, all further compare events during the current timer period are disabled for **CCy** until the corresponding timer overflows. If, after a match was detected, the compare register is reloaded with a new value, this value will not become effective until the next timer period.

To use the respective port signal as compare signal output signal CCyIO for compare register **CCy** in Compare Mode 3 this port signal must be configured as output, that is, the corresponding direction control bit must be set. With this configuration, the initial state of the output signal can be programmed or its state can be modified at any time by writing to the port output latch.

In Compare Mode 3 the port latch is set upon a compare event and cleared upon a timer overflow (see **Figure 138**).

However, when compare value and reload value for a channel are equal the respective interrupt requests are generated, only the output signal is not changed (set and clear would coincide in this case).

Note: If the port output latch is written to by software at the same time it would be altered by a compare event, the software write has priority. In this case the hardware-triggered change does not become effective.

9.4.3.6 Double-Register Compare Mode

In double-register compare mode two compare registers work together to control one output. This mode is selected by a special combination of modes for these two registers or via the **CCxDRM** register.

For double-register mode the eight capture/compare registers of a CAPCOM unit are regarded as two banks of 4 registers each. Registers **CC0**...**CC3** (**CC16**...**CC19**) form bank 1 while registers CC4...CC7 (**CC20**...**CC23**) form bank 2. For double-register mode a bank 1 register and a bank 2 register form a register pair. Both registers of this register pair operate on the pin associated with the bank 1 register (pins CC0IO...CC3IO).

The relationship between the bank 1 and bank 2 register of a pair and the effected output pins for double-register compare mode is listed in **Table 76**.

Table 76 Register Pairs for Double-Register Compare Mode

CAPCOM Unit 1		
Register Pair		**Associated Output Pin**
Bank 1	**Bank 2**	
CC0	CC4	CC0IO
CC1	CC5	CC1IO
CC2	CC6	CC2IO
CC3	CC7	CC3IO
CAPCOM Unit 2		
Register Pair		**Associated Output Pin**
Bank 1	**Bank 2**	
CC16	CC20	
CC17	CC21	
CC18	CC22	
CC19	CC23	

The double-register compare mode can be programmed individually for each register pair. To enable double-register mode the respective bank 1 register (see **Table 76**) must be programmed to Compare Mode 1 and the corresponding bank 2 register must be programmed to Compare Mode 0 or via the **CCxDRM** register.

If the respective bits in **CCxDRM** are set to 00 and if the respective bank 1 compare register is disabled or programmed for a mode other than mode 1 the corresponding bank 2 register will operate in Compare Mode 0 (interrupt-only mode).

in the following, a bank 2 register (programmed to Compare Mode 0) is referred to as CCz while the corresponding bank 1 register (programmed to Compare Mode 1) will be referred to as CCy.

When a match is detected for one of the two registers in a register pair (CCy or CCz) the associated interrupt request flag (CCyIR or CCzIR) is set and pin CCyIO corresponding to bank 1 register CCy is toggled. The generated interrupt always corresponds to the register that caused the match.

Note: If a match occurs simultaneously for both register CCy and register CCz of the register pair pin CCyIO are toggled only once but two separate compare interrupt requests are generated, one for vector CCyIRQ and one for vector CCzIRQ.

To use the respective port signal as compare signal output signal CCyIO for compare register CCy in double-register compare mode, this port signal must be configured as output, that is, the corresponding direction control bit must be set. With this configuration, the output signal has the same characteristics as in Compare Mode 1.

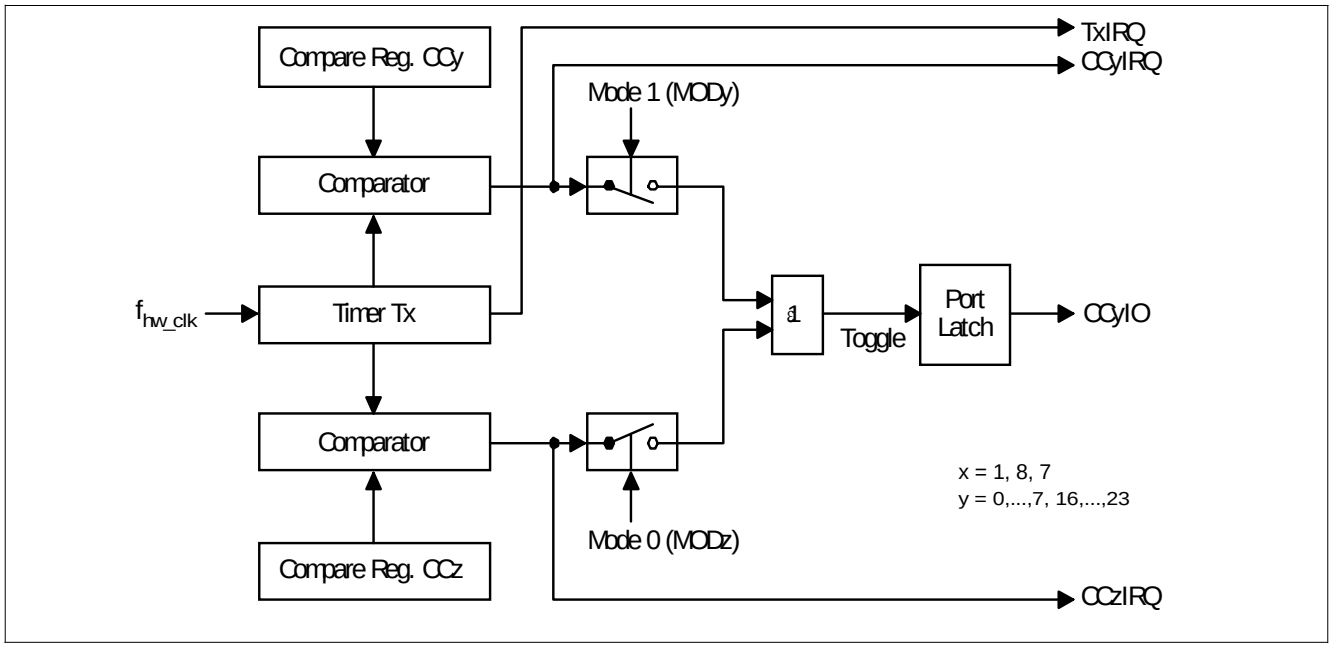


Figure 139 Double-Register Compare Mode Block Diagram

In this configuration example, the same timer allocation was chosen for both compare registers, but each register may also be individually allocated to one of the two timers of the respective CAPCOM unit. In the timing example for this compare mode (see **Figure 140**) the compare values in registers CCy and CCz are not modified.

Note: The signals CCzIO (which do not serve for double-register compare mode) may be used for general purpose IO.

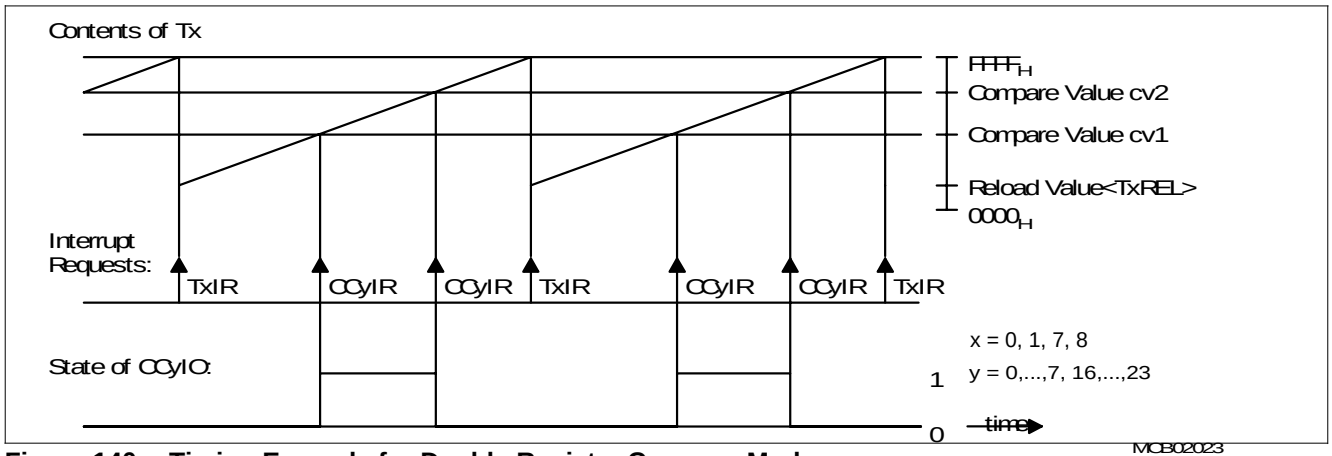


Figure 140 Timing Example for Double-Register Compare Mode

9.4.3.7 Disabled Capture and Compare Mode

Both Capture and Compare modes are disabled by setting bit field **CCMz.MODy** of the corresponding mode control register to 000_B.

The respective **CCy** registers then can serve for general variable storage. Also corresponding CCIOy pins can be used as General-Purpose Inputs/Outputs (GPIOs) and accessed directly through the **CCxOUT** register (refer to **Section 9.4.3.8.1 Alternative Implementation with Direct IO (on Page 360)**).

9.4.3.8 I/O Control Register

Via the CAPCOM's I/O control register, the enhanced features of the CAPCOM I/Os are selected such as general variable storage and others.

Bit **CCyIOC.ORSEL** determines whether the port register or the **CCxOUT** register is visible at the pins (refer to **Chapter 9.4.3.8.1**). For compatibility reasons the port register is default output if no other module has a higher priority at the port (EBC).

Bit **CCyIOC.PL** disables changes of the port output register by the CAPCOM. This feature is only to be used when the port register is connected to the pins (**CCyIOC.ORSEL** = 0).

Bit **CCyIOC.STAG** enables the I/O stagger mechanism. The staggered mode is only used when the **CCxOUT** register is connected to the pins (**CCyIOC.ORSEL** = 1).

The maximum resolution of the CAPCOM is reduced to 1/8 if this mode was enabled and the timers are running with 1/8 of the hw_clk clock. In this case the capture and compare functions are performed one after the other. Bit **CCyIOC.PDS** selects the port direction when the CAPCOM is neither in Capture nor Compare mode (**CCMz.MODy** = 000_B), and when the **CCxOUT** register is connected to the pins (**CCyIOC.ORSEL** = 1). In such a case the Alternative Implementation is true.

9.4.3.8.1 Alternative Implementation with Direct IO

In addition to the default implementation described above (see **Figure 131**), the CAPCOM can also be implemented in a way that the CAPCOM registers are connected to the ports without going through the product interface's port control
(**CCyIOC.ORSEL** = 1). Those bits of OUT register, currently in no capture/compare mode and hence serving as GPIOs, can be directly read/written from/to respective CCIOy pins. All such GPIOs are configured together by bit **CCyIOC.PDS**: either as outputs (**CCyIOC.PDS** = 0) or inputs (**CCyIOC.PDS** = 1).

Figure 141 shows the alternative interface diagram of the CAPCOM Unit 1 (CAPCOM Unit 2 is identical).

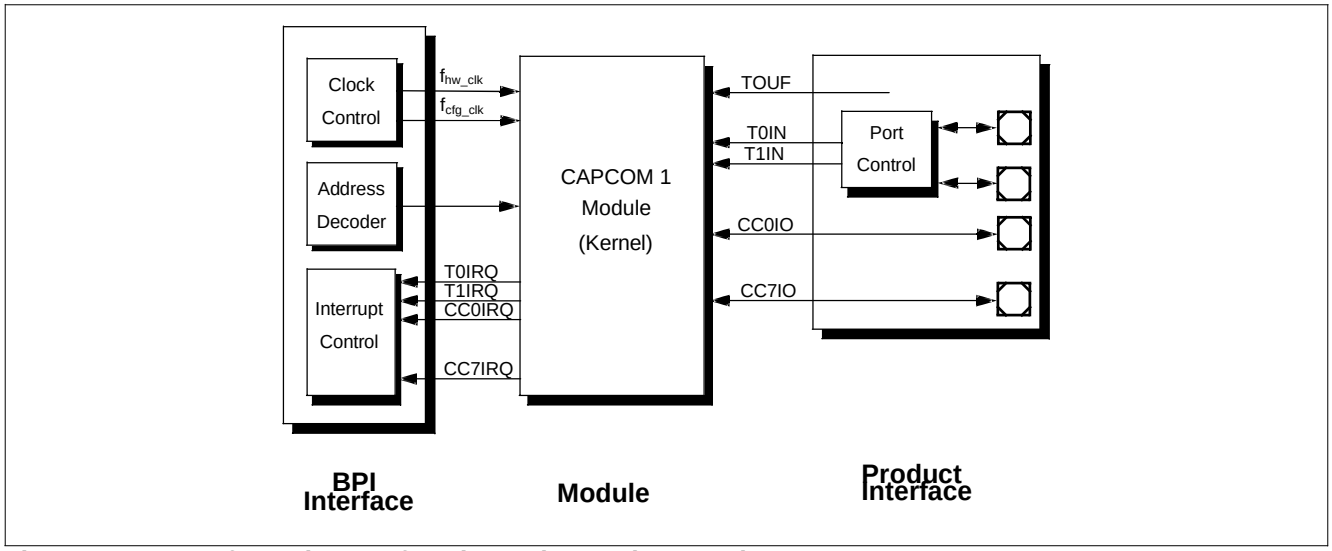


Figure 141 Interface Diagram for Alternative Implementation

9.4.3.9 Interrupt Node

Upon a capture or compare event, the interrupt request flag CCyIR for the respective capture/compare register CCyIC is set. This flag can be used to generate an interrupt or trigger an interrupt service request when enabled by the interrupt enable bit CCyIE.

Capture interrupts can be regarded as external interrupt requests with the additional feature of recording the time at which the triggering event occurred (refer to **Section 9.4.5 Interrupts (on Page 375)**).

Each of the 8 capture/compare registers (CC0...CC) has its own bit addressable interrupt control register (CC0IC...CCIC) and its own interrupt vector (CC0IRQ...CCIRQ). These registers are organized the same way as all other interrupt control registers.

9.4.3.10 Single Event Mode

The Single Event Mode is used to generate single edges or pulses and interrupts. It can be used with all compare modes.

For generating a single pulse the compare mode has to be set up and the **CCxSEM.SEMy** and **CCxSEE.SEEy** bits have to be set.

The **CCxSEE.SEEy** bit is cleared by HW after the event has occurred. By reading the register, the occurrence of the event can be controlled. Writing to this bit can be used to set and to clear it. Writing to this bit must be performed using bit protection.

9.4.4 Registers

The CAPCOM register mapping is in .

Table 77 CAPCOM Register Summary

Name	Access Condition	Clock	Description
T01CON **T78CON**	Bitaddressable	cfg_clk[1)]	CAPCOM 1 **T0** & **T1** control register CAPCOM 2 **T0** & **T1** control register
CCM0 & **CCM1** **CCM4** & **CCM5**	Bitaddressable	cfg_clk[1)]	CAPCOM 1 Mode Control registers CAPCOM 2 Mode Control registers
CC1ID **CC2ID**	Bitaddressable	cfg_clk[1)]	CAPCOM 1 Identification register CAPCOM 2 Identification register
CC1PISEL **CC2PISEL**	Bitaddressable	cfg_clk[1)]	CAPCOM 1 Port Input Select register CAPCOM 2 Port Input Select register
CC0 to **CC7** **CC16** to **CC23**	None	hw_clk[1)]	CAPCOM 1 Capture/Compare registers CAPCOM 2 Capture/Compare registers
CC1SEE **CC2SEE**	Bitaddressable, bit protected	hw_clk[1)]	CAPCOM 1 Single Event Enable register CAPCOM 2 Single Event Enable register
CC1SEM **CC2SEM**	Bitaddressable	cfg_clk[1]	CAPCOM 1 Single Event Mode register CAPCOM 2 Single Event Mode register
CC1DRM **CC2DRM**	Bitaddressable	cfg_clk[1)]	CAPCOM 1 Double Register Mode register CAPCOM 2 Double Register Mode register
CC1OUT **CC2OUT**	Bitaddressable, bit protected	hw_clk[1)]	CAPCOM 1 Output register CAPCOM 2 Output register
T0 **T7**	None	hw_clk[1)]	CAPCOM 1 Timer 0 registers CAPCOM 2 Timer 0 registers
T0REL **T7REL**	None	cfg_clk[1)]	CAPCOM 1 Timer 0 Reload registers CAPCOM 2 Timer 0 Reload registers
T1 **T8**	None	hw_clk[1)]	CAPCOM 1 Timer 1 registers CAPCOM 2 Timer 1 registers
T1REL **T8REL**	None	cfg_clk[1)]	CAPCOM 1 Timer 1 Reload registers CAPCOM 2 Timer 2 Reload registers
CC1IOC **CC2IOC**	Bitaddressable	cfg_clk[1)]	CAPCOM 1 I/O Control register CAPCOM 2 I/O Control register

1) Refer to Clock Domain in **System Integration (on Page 347)**

9.4.4.1 Timer Control Registers

TxxCON

T01CON

CAPCOM 1 Timer Control Registers **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	T1R	RESERVED		T1M	T1I		RESERVED	T0R	RESERVED	T0M	T0I				

T78CON

CAPCOM 2 Timer Control Registers **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	T8R	RESERVED		T8M	T8I		RESERVED	T7R	RESERVED	T7M	T7I				

Field	Bits	Type	Description
TxI	2:0, 10:8	rw	**Timer/Counter x Input Selection** • Timer Mode (**TxM** = 0): Input Frequency = $f_{hw\_clk}/2^{(<TxI>+3)}$. *Note: Refer to **Table 75 Timing Examples (on Page 352)** for examples.* • Counter Mode (**TxM** = 1): 000 Overflow/Underflow of Timer_in 001 Positive (rising) edge on TxIN 010 Negative (falling) edge on TxIN 011 Any edge (rising and falling) on TxIN 1XX Reserved. Do not use these combinations.
TxM	11, 3	rw	Timer/Counter x Mode Selection 0 Timer Mode (input derived from internal clock) 1 Counter Mode (input from external Input or T6)
TxR	14, 6	rw	Timer/Counter x Run Control 0 Timer/Counter x is disabled 1 Timer/Counter x is enabled
RESERVED	15, 13:12, 7, 5:4	r	Reserved; these bits must be left at their reset values.

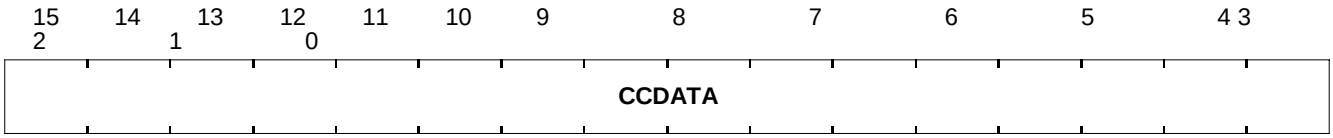
9.4.4.2 Capture/Compare Registers

CCy

CC0 through **CC7** are the data registers for the capture or compare operations for CAPCOM1 Timers **T0** and **T1**

CC0
CC1
CC2
CC3
CC4
CC5
CC6
CC7

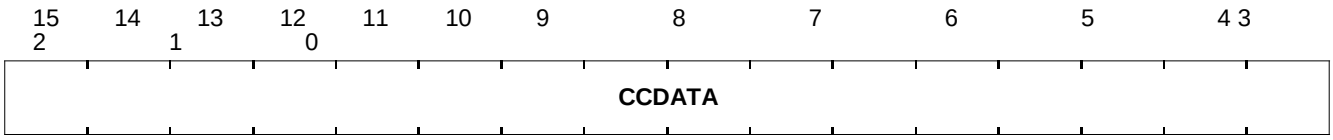
Capture/Compare Registers for the CAPCOM 1 **Reset value: 0000_H**



CC16 through **CC23** are the data registers for the capture or compare operations for CAPCOM 2 Timers **T7** and **T8**.

CC16
CC17
CC18
CC19
CC20
CC21
CC22
CC23

Capture/Compare Registers for the CAPCOM 2 **Reset value: 0000_H**



Field	Bits	Type	Description
CCDATA	15:0	rw	Capture/Compare Data Register

9.4.4.3 Capture/Compare Mode Registers

CCMz

CCM0

Capture/Compare Mode Registers for the CAPCOM 1 (CC0...CC3) **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC3	MOD3			ACC2	MOD2			ACC1	MOD1			ACC0	MOD0		

CCM1

Capture/Compare Mode Registers for the CAPCOM 1 (CC4...CC7) **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC7	MOD7			ACC6	MOD6			ACC5	MOD5			ACC4	MOD4		

CCM4

Capture/Compare Mode Registers for the CAPCOM 2 (CC16...CC19) **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC19	MOD19			ACC19	MOD18			ACC19	MOD17			ACC19	MOD16		

CCM5

Capture/Compare Mode Registers for the CAPCOM 2 (CC20...CC23) **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC23	MOD23			ACC22	MOD22			ACC21	MOD21			ACC20	MOD20		

Field	Bits	Type	Description
MODy (y = 0 to 7 or 16 to 23)	2:0, 6:4,10:8, 14:12	rw	Mode Selection for Capture/Compare Register **CCy** The available capture/compare modes are listed in **Table 78**.
ACCy (y = 0 to 7 or 16 to 23)	15, 11, 7, 3	rw	Allocation of Capture/Compare Register **CCy** 0 CCy allocated to Timer **T0**. 1 CCy allocated to Timer **T1**.

Table 78 Selection of Capture Modes and Compare Modes

MODy	Selected Operating Mode
000	Disable Capture and Compare Modes. The respective CAPCOM registers may be used for general variable storage.
001	Capture on Positive Transition (Rising Edge) at Pin CCyIO.
010	Capture on Negative Transition (Falling Edge) at Pin CCyIO.
011	Capture on Positive and Negative Transition (Both Edges) at Pin CCyIO.

Table 78 Selection of Capture Modes and Compare Modes (cont'd)

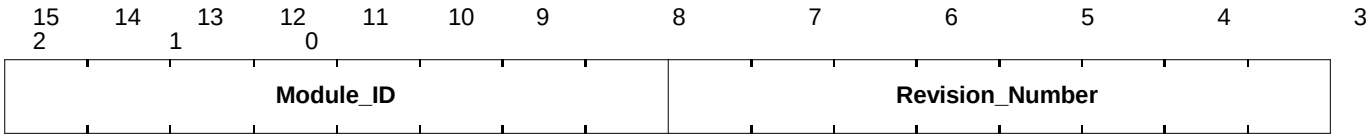
MODy	Selected Operating Mode
100	Compare Mode 0:Interrupt Only Several interrupts per timer period; Enables double-register compare mode for registers CC4...CC7.
101	Compare Mode 1:Toggle Output Pin on each Match. Several compare events per timer period; This mode is required for double-register compare mode for registers CC0...CC3.
110	Compare Mode 2:Interrupt Only Only one interrupt per timer period.
111	Compare Mode 3:Set Output Pin on each Match. Reset output pin on each timer overflow; Only one interrupt per timer period.

9.4.4.3.1 CAPCOM 1 & 2 Identification Registers

CCxID

CC1ID
CC2ID

CAPCOM Identification Register **Reset value: 5002_H**



Field	Bits	Type	Description
Revision_Number	0:7	r	**CAPCOM Revision Number** These hard-wired bits are used for module revision numbering.
Module_ID	8:15	r	**CAPCOM Identification Number** These hard-wired bits are used for module identification numbering.

9.4.4.4 CAPCOM 1 & 2 Compare Output Registers

The CAPCOM's compare output serves two registers in parallel, the port output register for binary compatibility and a separate one for enhanced functionality. The CAPCOM compare output and the port output latch is muxed in the port logic.

*Note: Compare output is visible at the pin if Compare Mode 1 or 3 is programmed in **CCMz.MODy**.*

CCxOUT

CC1OUT
CC2OUT

Compare Output Registers **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CC 7/23 IO	CC 6/22 IO	CC 5/21 IO	CC 4/20 IO	CC 3/19 IO	CC 2/18 IO	CC 1/17 IO	CC 0/16 IO	

Field	Bits	Type	Description
CCyIO (y = 0 to 7 or 16 to 23)	7:0	rwh	Compare Output for Channel y Alternative port output for port y. *Note: Refer to **CCyIOC**.*
RESERVED	15:8	r	Reserved; these bits must be left at their reset values.

9.4.4.5 Port Input Selection Registers

For switching between different port input sources the CAPCOM unit provides an input multiplexer. This multiplexer allows the selection between two input sources. The **CCxPISEL** registers control the switching of either each input channel or groups of input channels.

CCxPISEL

CC1PISEL

Port Input Select Register 1 **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								T1 INIS	T0 INIS	C7C6 IS	C5C4 IS	C3C2 IS	C1C0 IS		

Field	Bits	Type	Description
C1C0IS C3C2IS C5C4IS C7C6IS	0 1 2 3	rw	**Select Source for Capture Input Channels** 0 Default input port 1 Alternate input port Refer to **Figure 142** and **Table 79**.
TxINIS	4	rw	**Select Source for External Input of Timer 0 Clock** 0 T0IN pin selected (default) 1 Tied to logical 0, no input received
TxINIS	5	rw	**Select Source for External Input of Timer 1 Clock** 0 t_int1 signal from GSM Timer 1 Tied to logical 0, no input received
RESERVED	15:6	r	Reserved; these bits must be left at their reset values.

CC2PISEL

Port Input Select Register 2 **Reset value: 0000_H**

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0															
RESERVED									T8INIS	T7 INIS	C23 C22 IS	C21 C20 IS	C19 C18 IS	C17 C16 IS	

Field	Bits	Type	Description
C17C16IS C19C18IS C21C20IS C23C22IS	0 1 2 3	rw	**Select Source for Capture Input Channels** 0 Default input port. 1 Alternate input port. Refer to **Figure 142** and **Table 80**.
T0INIS	4	rw	**Select Source for External Input of Timer 7 Clock** 0 T7IN pin selected (default) 1 T7IN pin selected (default)
T1INIS	5	rw	**Select Source for External Input of Timer 8 Clock** 0 t_int2 signal from GSM Timer 1 Tied to logical 0, no input received
RESERVED	15:6	r	Reserved; these bits must be left at their reset values.

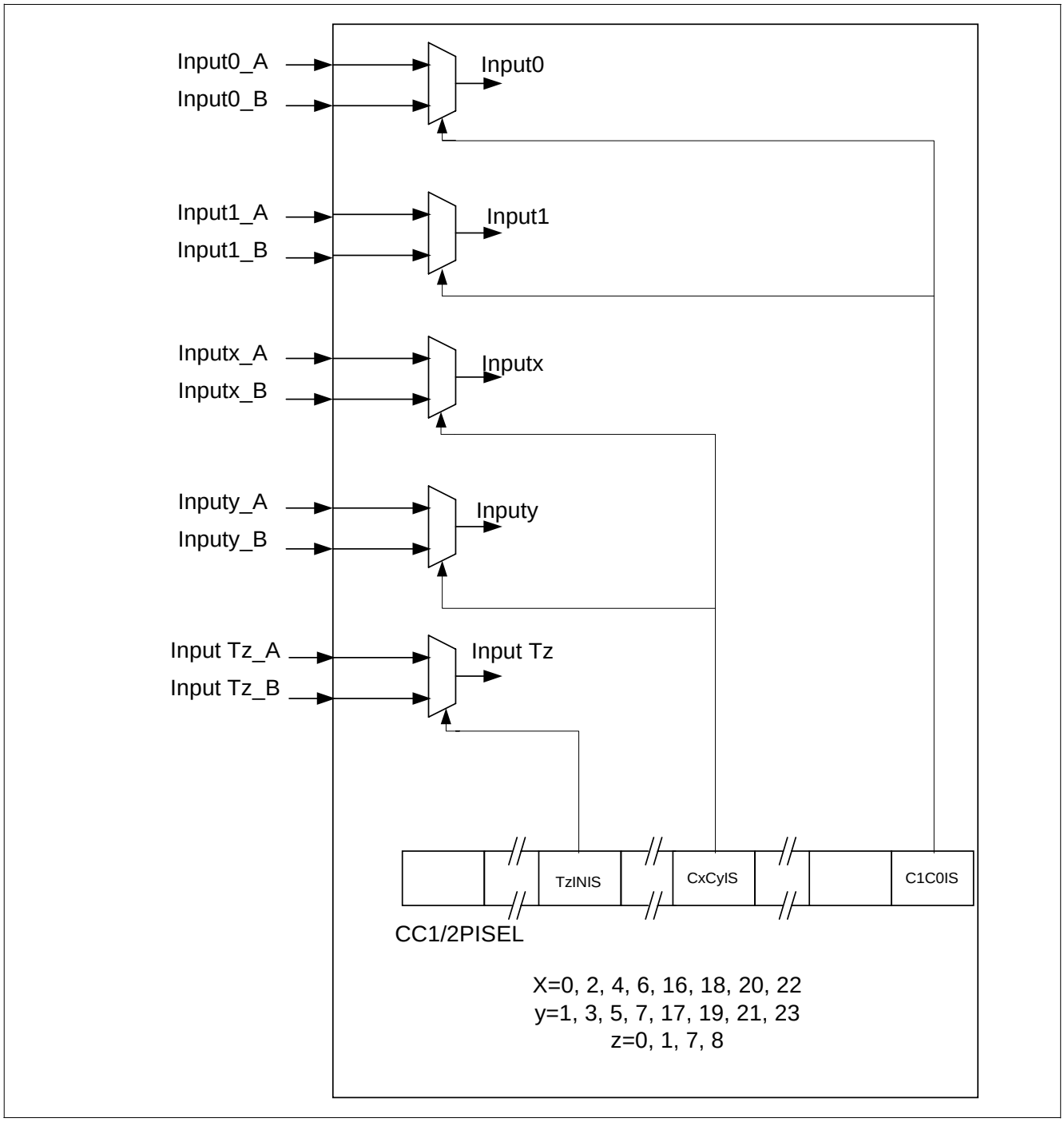


Figure 142 Port Input Selection for Capture Channels

Table 79 CAPCOM 1 Input Signal Selection

Pin Select Input (See Figure 142)	Selected with Bit in CC1PISEL	Ball Name[1)]	ALT Mode[2)]	Connected to Alternate Function
Input0_A	C1C0IS = 0	CC00IO SSC0_CLK	ALT 0 ALT 2	CC00IO CC00IO
Input0_B	C1C0IS = 1	CTS_n	ALT 1	CC00IOb
Input1_A	C1C0IS = 0	KP4	ALT 1	CC01IOa
Input2_A	C3C2IS = 0	KP5	ALT 1	CC02IO

Table 79 CAPCOM 1 Input Signal Selection (cont'd)

Pin Select Input (See Figure 142)	Selected with Bit in CC1PISEL	Ball Name[1)]	ALT Mode[2)]	Connected to Alternate Function
Input2_B	C3C2IS = 1	KP5	ALT 1	CC02IO
Input4_A	C5C4IS = 0	TXD	ALT 1	CC04IO
Input4_B	C5C4IS = 1	TXD	ALT 1	CC04IO
Input5_A	C5C4IS = 0	T2IN	ALT 1	CC05IO
Input5_B	C5C4IS = 1	T2IN	ALT 1	CC05IO
Input6_A	C7C6IS = 0	SSC0_MRST	ALT 2	CC06IOa
Input6_B	C7C6IS = 1	DISP_REST	ALT 2	CC06IOb
Input7_A	C7C6IS = 0	T_OUT2	ALT 1	CC07IO
Input7_B	C7C6IS = 1	T_OUT2	ALT 1	CC07IO
InputT0_A	T0INIS = 0	Not connected		
InputT0_B	T0INIS = 1	Not connected		
InputT1_A	T1INIS = 0	Connected to GSM interrupt TINT1		
InputT1_B	T1INIS = 1	Not connected		

1) Refer to **Chapter 3 Pin Descriptions**.

2) If there are two possible pads, choose one of the two. The pads are mutually exclusive.

Table 80 CAPCOM 2 Input Signal Selection

Pin Select Input (See Figure 142)	Selected with Bit in CC2PISEL	Ball Name[1)]	ALT Mode[2)]	Connected to Alternate Function
Input0_A	C17C16IS = 0	CCVZ_n	ALT1	CC16IOa
Input0_B	C17C16IS = 1	KP3	ALT1	CC16IOb
Input1_A	C17C16IS = 0	SSC0_MTSR	ALT 2	CC17IO
Input1_B	C17C16IS = 1	SSC0_MTSR	ALT 2	CC17IO
Input2_A	C19C18IS = 0	KP9	ALT 1	CC18IOa
Input2_B	C19C18IS = 1	RTS_n	ALT 1	CC18IOb
Input3_A	C19C18IS = 0	VIB_CONTROL	ALT 1	CC19IO
Input3_B	C19C18IS = 1	VIB_CONTROL	ALT 1	CC19IO
Input4_B	C21C20IS = 1	KP2	ALT 1	CC20IOb
Input5_A	C21C20IS = 0	OE_n	ALT 1	CC21IO
Input5_B	C21C20IS = 1	OE_n	ALT 1	CC21IO
Input6_A	C23C22IS = 0	KP8	ALT 1	CC22IOa
Input6_B	C23C22IS = 1	BACK_LIGHT	ALT 1	CC22IOb
Input7_A	C23C22IS = 0	W_LED_DRV	ALT 1	CC23IO
Input7_B	C23C22IS = 1	W_LED_DRV	ALT 1	CC23IO
InputT0_A	T7INIS = 0	KP7	ALT 1	T7IN
InputT0_B	T7INIS = 1	KP7	ALT 1	T7IN
InputT1_A	T8INIS = 0	Connected to GSM interrupt TINT2		
InputT1_B	T8INIS = 1	Not connected		

1) Refer to **Chapter 3 Pin Descriptions**.

2) If there are two possible pads, choose one of the two. The pads are mutually exclusive.

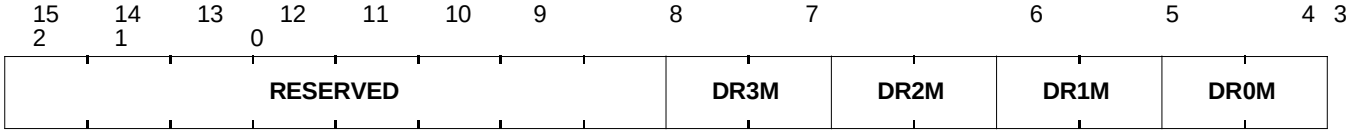
9.4.4.6 Double-Register Compare Mode Register

CCxDRM

CC1DRM

CC2DRM

Double-Register Compare Mode Register **Reset value: 0000_H**



Field	Bits	Type	Description
DRxM x = 0 to 3	1:0, 3:2, 5:4, 7:6	rw	**Double Register x compare Mode selection** 00 **DRxM** is controlled via compare modes 1 and 0. 01 **DRxM** disabled regardless of compare modes. 10 **DRxM** enabled regardless of compare modes. 11 Reserved *Note: **x**: 0 for registers 0 and 4, 1 for registers 1 and 5, etc.*
RESERVED	15:8	r	Reserved; these bits must be left at their reset values.

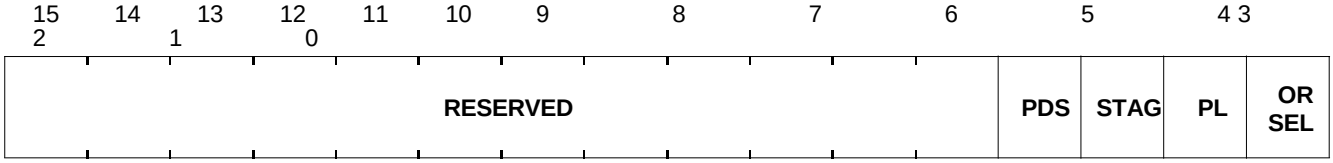
9.4.4.7 IOC Registers

CCyIOC

CC1IOC

CC2IOC

I/O Control Register **Reset value: 0001_H**



Field	Bits	Type	Description
ORSEL	0	r	**Output Register Select** 0 The contents of the port register Py is visible at pad y (default). 1 The contents of the **CCxOUT** register is visible at the pad y. *Note: This bit is hardwired and it indicates which implementation is used for the CAPCOM (see **Section 9.4.2 Section 9.4.2 (on page 349)** and **Section 9.4.3.8.1 Section 9.4.3.8.1 (on page 360)**).*
PL	1	rw	**Port Lock** 0 The contents of the port register is changed by the CAPCOM unit (default). 1 The contents of the port register is not changed by the CAPCOM unit. *Note: Value of **PL** will be ignored when **ORSEL** = 1.*
STAG	2	rw	**Stagger** 0 Staggered mode enabled (default). 1 Staggered mode disabled. *Note: **STAG** = 1 is reserved when **ORSEL** = 1.*
PDS	3	rw	**Port Direction Select** 0 Port direction is Output (default). 1 Port direction is Input. *Note: The **PDS** value is only relevant the following is true:* • When CAPCOM is in neither Capture nor Compare Mode (**CCMz.MODy** = 000_B) • **CCxOUT** registers are connected to the pads (**CCxOUT.ORSEL** = 1).
RESERVED	15:4	r	Reserved; these bits must be left at their reset values.

9.4.4.8 CAPCOM 1 & 2 Single Event Mode Register

CCxSEM

CC1SEM
CC2SEM

Single Event Mode Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							SEM 7	SEM 6	SEM 5	SEM 4	SEM 3	SEM 2	SEM 1	SEM 0	

Field	Bits	Type	Description
SEMy (y = 0 to 7 or 16 to 23)	7:0	rw	**Single Event Mode selection** 0 Mode disabled for channel y 1 Mode enabled for channel y
RESERVED	15:8	r	Reserved; these bits must be left at their reset values.

9.4.4.9 CAPCOM 1 & 2 Single Event Enable Register

CCxSEE

CC1SEE
CC2SEE

Single Event Enable Registers **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							SEE 7	SEE 6	SEE 5	SEE 4	SEE 3	SEE 2	SEE 1	SEE 0	

Field	Bits	Type	Description
SEEy (y = 0 to 7 or 16 to 23)	7:0	rwh	Single Event Enable 0 Event disabled for channel y. 1 Event enabled for channel y. This bit is cleared after the event has occurred by the HW.
RESERVED	15:8	r	Reserved; these bits must be left at their reset values.

9.4.4.10 Timer and Timer Reload Registers

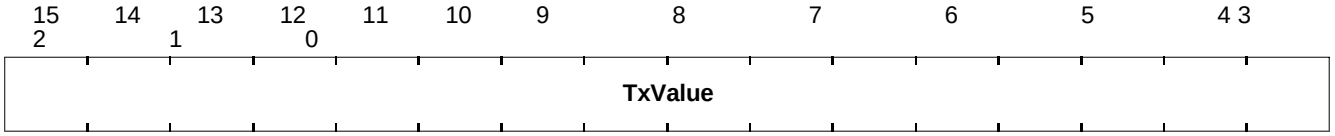
Timers

T0 and **T1** (for CAPCOM 1), **T7** and **T8** (for CAPCOM 2) are the timer registers that store timer values.

Tx

T0
T1
T7
T8

Timer Registers **Reset value: 0000_H**



Field	Bits	Type	Description
TxValue x = 0,1,7, or 8	15:0	rw	Timer value

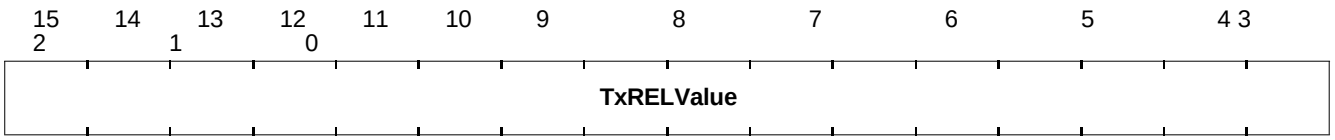
Timer Reloads

T0REL and **T1REL** (for CAPCOM 1), **T7REL** and **T8REL** (for CAPCOM 2) are the timer reload registers that store the timer reload values.

TxREL

T0REL
T1REL
T7REL
T8REL

Timer Reload Registers **Reset value: 0000_H**



Field	Bits	Type	Description
TxRELValue x = 0,1,7, or 8	15:0	rw	Timer reload value

9.4.5 Interrupts

Table 81 is an overview of the interrupts. For a detailed description refer to Section 9.4.4 Registers (on Page 362).

Table 81 CAPCOM Interrupt Sources

Interrupt	Signal	SRC Register	Description
Timer 0 overflow	int_T0_o		Interrupt is requested on overflow of Timer 0.
Timer 1 overflow	int_T1_o		Interrupt is requested on overflow of Timer 1.
Rising edge on Pin CCxIO	int_ CCx_o		Interrupt is requested on positive transition at pin CCxIO in Capture Mode **TxxCON.MODy** = 001_B.
Falling edge on Pin CCxIO	int_ CCx_o		Interrupt is requested on negative transition at pin CCxIO in Capture Mode **TxxCON.MODy** = 010_B.
Rising or falling edge on Pin CCxIO	int_ CCx_o		Interrupt is requested on positive or negative transition at pin CCxIO in Capture Mode **TxxCON.MODy** = 011_B.
Match CCx and Ty	int_ CCx_o		Interrupt is requested on detected match between content of compare register CCx and allocated Timer Ty in Compare Mode 0 (**TxxCON.MODy** = 100_B).
Match CCx and Ty	int_ CCx_o		Interrupt is requested on detected match between content of compare register CCx and allocated Timer Ty in Compare Mode 1 (**TxxCON.MODy** = 101_B).
Match CCx and Ty	int_ CCx_o		Interrupt is requested on detected match between content of compare register CCx and allocated Timer Ty in Compare Mode 2 (**TxxCON.MODy** = 110_B).
Match CCx and Ty	int_ CCx_o		Interrupt is requested on detected match between content of compare register CCx and allocated Timer Ty in Compare Mode 3 (**TxxCON.MODy** = 111_B).
Match CCx and CCy	int_ CCx_o or int_ CCy_o		Interrupt is requested on detected match between contents of register pair CCx and CCy in Double Register Compare Mode.

9.5 RTC

System Integration

- Supply domain:
  - Register **RTCIF** & Bus Interface: VDD_LD1
  - All other parts: VDD_RTC
- Chip internal interfaces:
  - Clock domains:
    - *pll_clk_i* for bus interface logic
    - software register controls selection of *pll_clk_i* or *rtc_ref_clk_i* (32KHz) for internal counter logic
  - Bus domain: PD Bus
  - Interrupt sources: 2
  - Other interfaces: Padframe.
- Chip external signals related to this block :
  F32K, OSC32K, RTC_OUT, RESET.

9.5.1 Introduction

The integrated Real Time Clock (RTC) is able to provide programmable alarm functions and external interrupts. Due to its extreme low power consumption the RTC can be supplied from a small backup battery. This allows the generation of external interrupts, even when the main PMB7880 supply voltage is switched off. For this purpose the RTC is powered by own voltage supply pins VDD_RTC and VSS_RTC.

The RTC shall be driven by a 32.768 kHz (32k) clock which needs to be applied via the PMB7880 F32K and OSC32K pins. The clock can be fed from either an external clock source or use the on chip 32 KHz oscillator module.

The low clock frequency and the optimized low power design give the possibility to run the chip with a minimum of power dissipation. For example, for this specific application the 26 MHz reference oscillator can be switched off during system standby and a low-power time reference can be kept when the 32k clock is provided to the RTC. The RTC consists of an PMB7880 specific RTC shell (**p3_rtc_core**), containing the RTC macro (**p3_rtc_kernel**), as well as the 32 kHz oscillator, as described in the following sections. The module **p3_rtc_pre** solely performs level translation of the 32KHz clock to the VDD_LD1 power supply domain, and is not functionally associated with the RTC.

In addition, another system function, the pad tristate control, is implemented in the RTC block. This is because this function needs the RTC power supply. Due to its strong relation with the system power management, the pad tristate control is described in **Chapter 12 System Reset (on Page 581)**.

9.5.2 RTC Register Overview

For the register addresses refer to **Section 10.1 PD-Bus Register Addresses (on Page 481)**

Table 82 RTC Register List

Register Group	Register Name	Register Symbol
System Register	RTC Identification	**RTC_ID**

Table 82 RTC Register List (cont'd)

Register Group	Register Name	Register Symbol
Control and Status Registers	RTC Shell Control	**RTC_CTRL**
	RTC Control Register	**RTC_CON**
	RTC Interface (Isolation)	**RTCIF**
Data Registers	Timer T14 Count	**RTC_T14_CNT**
	Timer T14 Reload	**RTC_T14_REL**
	RTC Count	**RTC_CNT_LO** & **RTC_CNT_HI**
	RTC Reload	**RTC_REL_LO** & **RTC_REL_HI**
	RTC Alarm	**RTC_ALARM_LO** & **RTC_ALARM_HI**
	RTC Interrupt Sub Node Control.	**RTC_ISNC**

9.5.3 RTC Shell

9.5.3.1 Register Descriptions

Only the registers of the RTC shell are described here. The main RTC register descriptions can be found in the RTC macro section.

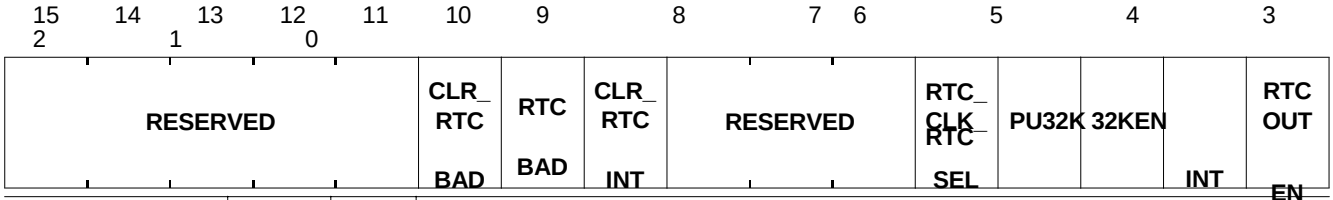
*Note: The register **RTC_CTRL** (along with all other RTC registers) can only be accessed by software when the register **RTCIF** is set so that RTC isolation is disabled.*

9.5.3.1.1 RTC Shell Control Register

RTC_CTRL

RTC Shell Control Register

Reset value: 0200_H
on RTC SW reset only



Field	Bits	Type	Description
RTCOUTEN	0	rw	**RTC External Interrupt Output Enable** 0: Disabled, RTC_OUT pin shows low level 1: Enabled
RTCINT	1	rh	**RTC Interrupt Status** 0: No RTC_INT interrupt occurred 1: RTC_INT interrupt occurred *Note: **RTCINT** needs to be cleared by the CPU by setting bit **CLR_RTCINT** in this register*
32KEN	2	rw	**32k Clock Enable** (See **Figure 145 (on page 381)** and **Figure 146**) 0: Disabled 1: Enabled
PU32K	3	rw	**32 kHz Oscillator Power Up** (See **Figure 145** and **Figure 146**) 0: Disabled 1: Enabled

Field	Bits	Type	Description
RTC_CLK_SEL	4	rw	**RTC Logic Clock Select** 0: 32 kHz clock operation mode (Asynchronous to microcontroller clock, low power, read only) 1: Bus clock operation mode (Synchronous to microcontroller clock, required for register write operation for some registers, refer to **Table 83 Register Assignment to Clock and Reset Domains (on Page 389)**) Upon changing this bit from 32 kHz to BPI clock, RTC register access can only begin when **RTC_CON.ACCPOS** is active. *Note: The clock must not be switched when the Bus clock frequency is less than or equal to 32 kHz.*
CLR_RTCINT	8	w	**Clears RTC_INT** 0 No action 1 Clears **RTCINT** Reading always returns 0.
RTCBAD	9	rh	**RTC Content Inconsistent Due to Power Supply Drop Down** 0: RTC content consistent 1: A error in the RTC counter has been detected. This bit is set by hardware. *Note: To clear this bit software must perform write operations to the **RTC_CNT_LO** & **RTC_CNT_HI** registers, followed by software asserting **CLR_RTCBAD** in this register. This must be done while the RTC operates in the synchronous mode.*
CLR_RTCBAD	10	w	**Clear RTCBAD** 0 No action 1 Clears **RTCBAD** Also refer to **RTCBAD** for details on clearing this bit. Reading always returns 0.
RESERVED	7:5, 15:11	r	Reserved; these bits must be left at their reset values.

*Note: The 2 Bits PU32K and 32KEN are locked by the **PMU_PWRCTRL2**.CLKSEL bit. As soon as this bit is set (this means PMU has switched to the RTC 32kHz clock), those bits can't be reset with an RTC SW reset.*

9.5.3.1.2 RTC Interface Register

RTCIF

RTC Interface Register **Reset value: 0000_H**

The **RTCIF** value must be set to the Enabled value specified to allow access to the RTC registers.

*Note: The register **RTCIF** is not in the RTC voltage supply domain but in the same supply domain as the CGU. It is also reset with the reset for the CGU.*

9.5.3.2 Internal Interrupts

The RTC may generate 2 internal interrupt signals *rtc_int* and *rtc_t14int* from 6 interrupt sources. The interrupt sources are ALARM, RTC0..3 and T14.

- *rtc_t14int* is driven directly by the T14 interrupt
- *rtc_int* is driven by combining the enabled 6 interrupt sources. Each of the 6 interrupt sources may be enabled independently, as described in **Section 9.5.5.4.11 (on page 388)**.

9.5.3.3 External Interrupt RTC_OUT

The interrupt output rtc_int of the RTC macro sets bit **RTC_CTRL.RTCINT**. Once an interrupt has been detected, bit **RTCINT** remains at the logic HIGH level until it is reset by the CPU. The status of bit **RTCINT** is directed to the RTC_OUT pin, if bit **RTC_CTRL.RTCOUTEN** is set and the **RTC_CON.RTCBAD** bit is not set (see **Figure 143**).

Note: RTC_OUT (as well as rtc_t14int and rtc_int) are undefined before the RTC is reset. The overall system has to handle this state appropriately.

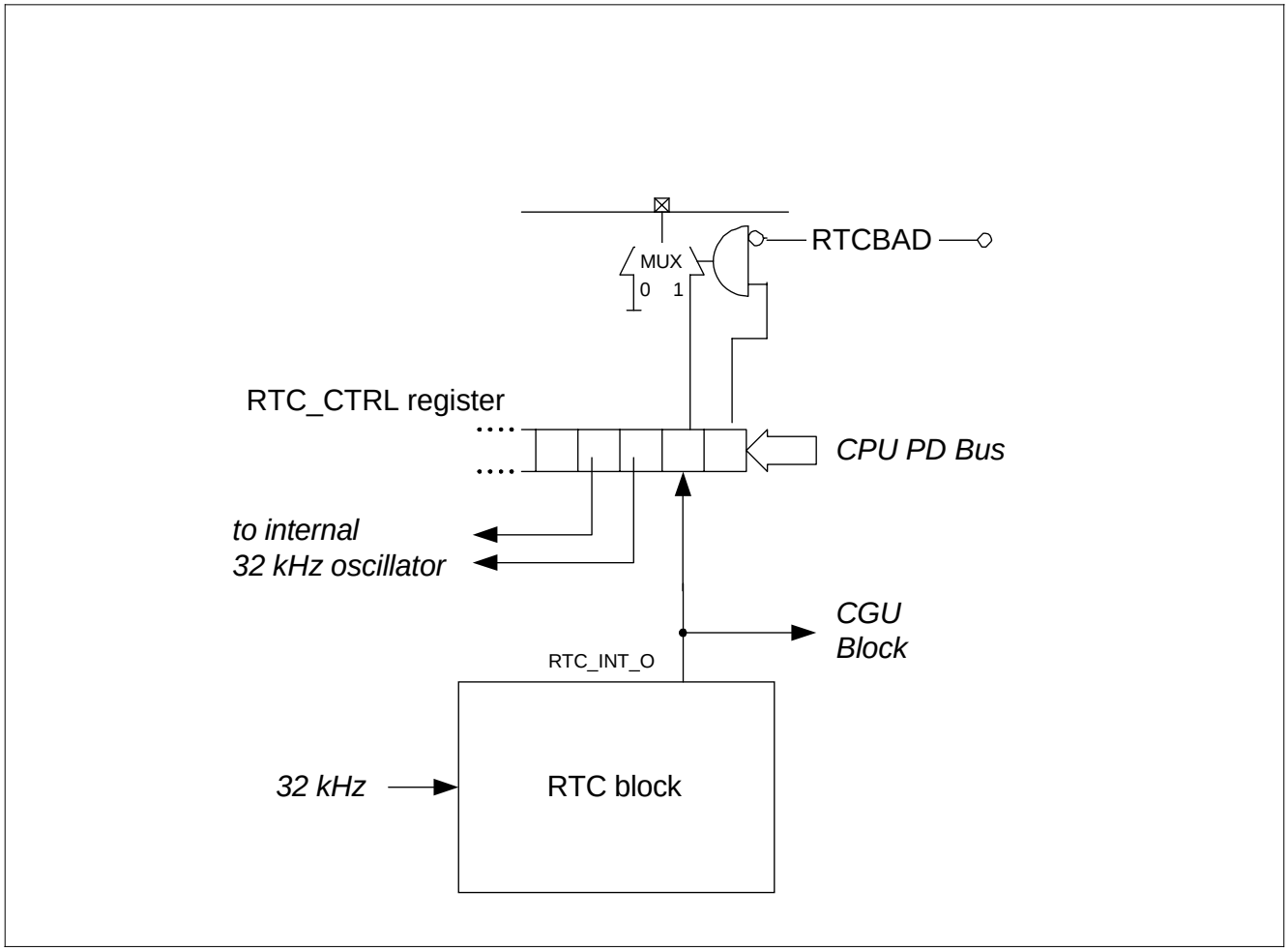


Figure 143 RTC Interrupt Generation

9.5.3.4 Content Validity Check

A feature of the RTC block in the PMB7880 is the data content validity check capability. There are two registers capturing a part of the RTC counter in inverted and non-inverted form (6-bit part of the **RTC_CNT_HI (on Page 386)** register field and it's shadow register). Both registers are hardware-incremented/decremented according to the RTC operation. In the case of a RTC battery voltage drop-out or replacement, the content of those two registers will no longer be complementary. This error is latched into the **RTC_CON.RTCBAD** register bit and can be detected by the software which may force the user to update the time and date information. The **RTC_CON.RTCBAD** register bit then has to be cleared by software.

9.5.3.5 *There is a VDD voltage range in which registers can not be incremented any more but they keep their value. When the VDD drops into this range, the registers stay complementary but they are not up-to-date any more. If the VDD voltage is stored externally in large capacitor, this is even likely to happen.*

*Thus the content validity check may not be useful in some applications.***RTC Software Reset**

The RTC can be reset by software-reset only. By this means the RTC keeps its value during the PMB7880 hardware-reset. Refer to **Section 7.2.1.5.7 Reset Control and Status Register** for details. It becomes effective only if the **RTCIF** register is set so that RTC Isolation is disabled.

The RTC is in an undefined state when VDD_RTC has dropped below the minimum specified value (first unit initialization or power-up after the RTC-backup cell died). During the time between the VDD power-up and controller software-reset, the RTC is in an undefined state.

Writing '1' into **PMU_PWRCTRL2**.**CLKSEL** blocks the RTC software reset on the 2 bits **RTC_CTRL**.**PU32K and RTC_CTRL**.**32KEN.** The next RTC software reset will have, therefore, no effect on these 2 bits : the 32KHz clock cannot be switched off any longer.Refer to **Section 8.1.2.13 Clock Scheme (on Page 253)** for details

9.5.3.6 RTC Isolation

The RTC unit together with its related pads may be isolated electrically from the other parts of the E-GOLDvoice and work completely autonomously in an asynchronous mode. This has to be done before the chip is put in the powerdown mode. The isolation helps also to save power on the RTC power supply because the BPI clock domain registers are not then clocked. Therefore the isolation may be used every time when the connection to the RTC is not required by the controller.

The isolation mode is controlled by the register **RTCIF**, which is in the VDD_LD1 supply domain. **RTCIF** is actually implemented in the CGU. Isolation mode is enabled upon power on reset and can be removed only by writing "10101010" to the register.If any other value is written to **RTCIF**, the RTC is isolated. It is strongly recommended to write "00000000" to **RTCIF** for isolation, because a 1 may cause some current flow during the drop down phase of VDD_LD1.

In the isolated state the RTC runs in the Asynchronous mode. Although the hardware forces this mode automatically when the isolated state is reached, the software should switch to Asynchronous mode before isolating the RTC in order to avoid undefined states during the transition from Not-Isolated to Isolated.

Isolation mode of the RTC is primarily intended to allow the RTC to operate when the supply voltage to the core has been removed. Before this occurs, the software must put the RTC in Asynchronous mode (i.e. select entire RTC operation from the 32KHz clock rather than the bus clock). The 8-bits from the **RTCIF** register are passed to permanently enabled level shifters located in the padframe, which have the characteristic of storing the input **RTCIF** value (on their outputs) before the inputs float down, as in the case when the core power is removed. By writing a value to **RTCIF** which selects isolation (preferably all zeros) prior to core power removal, all input level shifters within the RTC are disabled, which means that all input signals to the RTC will remain in a defined state during the time core power drops off, and allows the RTC to then operate autonomously when the core power has been removed.

9.5.4 32k Oscillator

The 32k oscillator is used primarily for the RTC clock generation. It also supports the CGU block (**Section 7.2.1.5.7 Reset Control and Status Register**) with a reference clock. Additionally the 32k clock can be used for controller operation in a power-saving mode. The major oscillator parameters are low power consumption and a wide operating voltage range. The oscillator is in the same supply domain as the RTC.

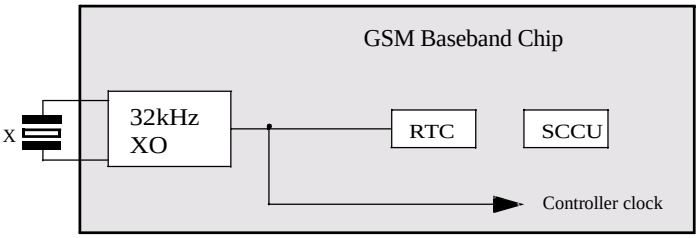


Figure 144 MS Clock Generation with On-Chip Oscillators

The user can use the on-chip oscillator or an external 32kHz circuitry (e.g. in a power management ASIC). If used, the 32k oscillator requires 2 pins for the external quartz.

The 32k oscillator block has a register interface to the controller via the PD Bus. The RTC block itself can operate in two modes: in an Asynchronous mode and a Synchronous mode. In the Asynchronous mode, the RTC block runs with the 32k clock. The register access is limited to those indicated as being in the Bus Clock domain in **Table 83**. To enable the complete register access the RTC has to be (temporarily) switched into the Synchronous mode. In the Synchronous mode, the block logic and all registers run with the bus clock.

The figure below shows the basic circuitry required by the internal 32k oscillator. Externally connected are the crystal, the load capacitances C_1 and C_2 and the resistor R_1 which is used to limit the drive level. Recommended values are given in **Section 11.2.2.9.1 Timing of RTC**.

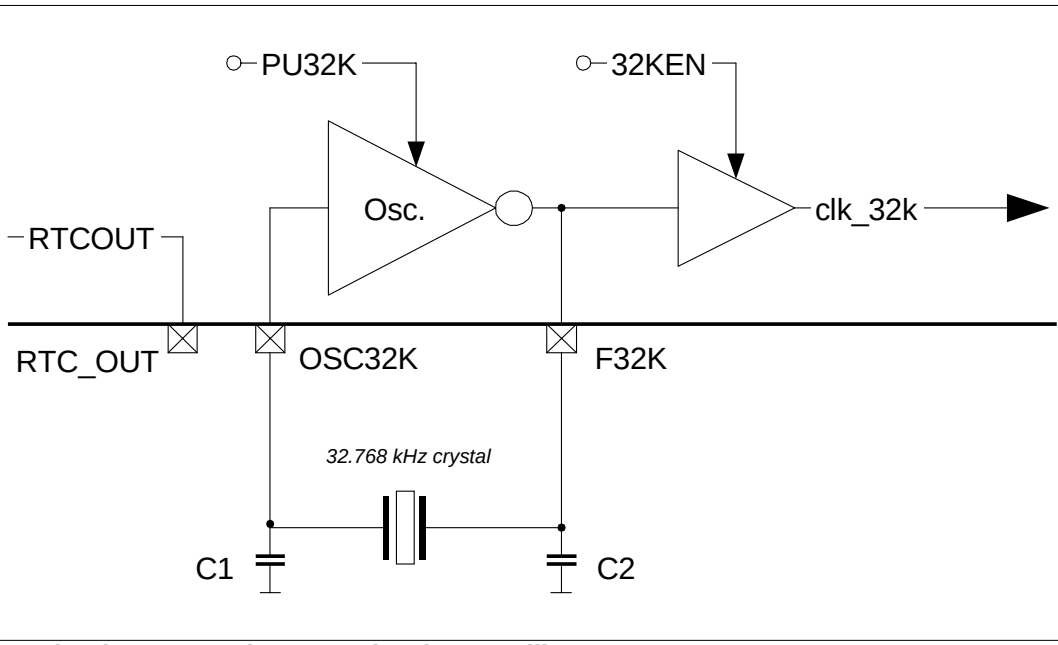


Figure 145 Circuitry to Use the Internal 32 kHz Oscillator

The Oscillator Block can be enabled by setting bit **RTC_CTRL.PU32K** of the RTC block. Bit **RTC_CTRL.32KEN** enables the clock clk_32k.

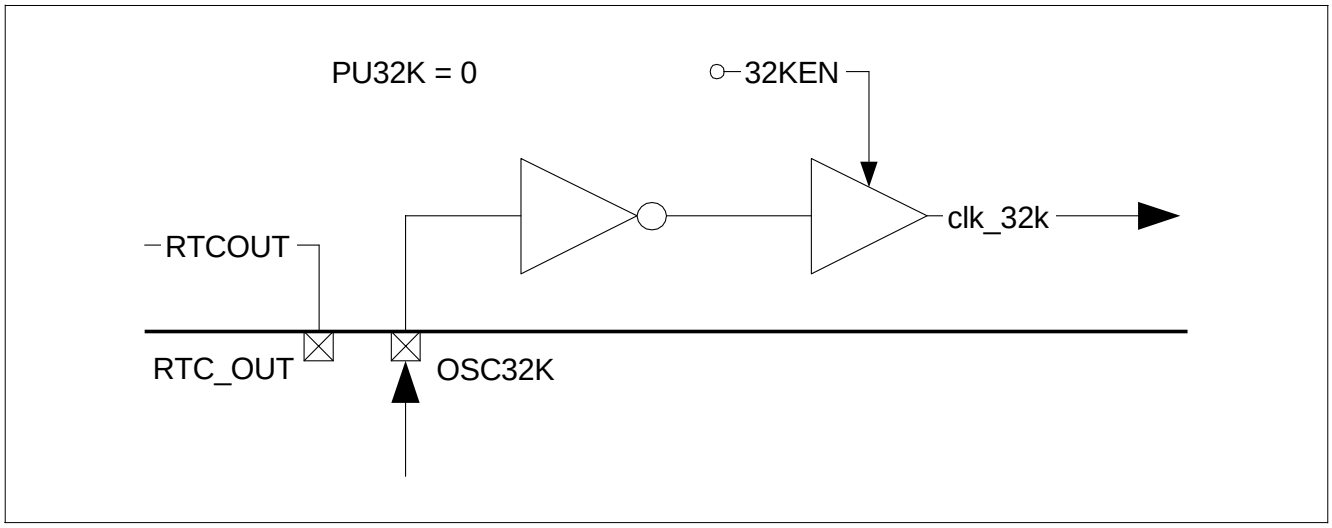


Figure 146 Circuitry to Use the External 32 kHz Oscillator

If the 32kHz signal is fed in via the OSC32K input pad, the signal levels correspond to those specified in **Section 11.2.1.2 Voltages**.

Writing '1' into **PMU_PWRCTRL2**.**CLKSEL** blocks the RTC software reset on the 2 bits **RTC_CTRL**.**PU32K and RTC_CTRL**.**32KEN.** The next RTC software reset will have, therefore, no effect on these 2 bits : the 32KHz clock cannot be switched off any longer.Refer to **Section 8.1.2.13 Clock Scheme** for details.

9.5.5 RTC Macro

The Real Time Clock (RTC) module is basically an independent timer chain and counts clock ticks. The base frequency of the RTC can be programmed via a reload counter. The RTC can work fully asynchronous to the system frequency and is optimized for low power consumption.

Features

- Two reloadable timers, T14 (16-bit) and CNT (32-bit)
- Timers can operate in Synchronous or Asynchronous Mode.

9.5.5.1 Operational Overview

The real time clock module provides three different types of registers:

- Control registers for controlling RTC functionality
- Data registers for setting the clock divider for RTC base frequency programming and for flexible interrupt generation
- Counter registers, which contain the actual time and date.

The interrupts are programmed via one interrupt sub node register (refer to **Section 9.5.3.2 Internal Interrupts (on Page 378)**).

9.5.5.2 Register Overview

Refer to **Table 82 RTC Register List (on Page 376)**.

9.5.5.3 Functional Overview

The RTC module consists of a chain of 2 divider blocks, the reloadable 16-bit timer T14 and the 32-bit RTC timer. Both timers count up. Each timer contains reload registers (**RTC_T14_REL** and **RTC_REL_LO**/**RTC_REL_HI**) and count registers (**RTC_T14_CNT** and **RTC_CNT_LO**/**RTC_CNT_HI**). **RTC_T14_CNT** is reloaded with the

value of **RTC_T14_REL** on every **RTC_T14_CNT** overflow. The 32-bit RTC timer (**RTC_CNT_LO**/**RTC_CNT_HI**) is split into four smaller related sections (10-bit/6-bit/6-bit/10-bit). The reload parts of these sections are grouped to register **RTC_REL_LO**/**RTC_REL_HI** and the count parts to register **RTC_CNT_LO**/**RTC_CNT_HI**.

The count input of the RTC module can be optional divided by a prescaler with factor 8.

The RTC module operates in two different modes, an Asynchronous and a Synchronous Mode. In Synchronous Mode the RTC module is clocked with a synchronous clock derived from the bus clock. In Asynchronous Mode the RTC module is clocked by the 32K clock. The Asynchronous Mode is necessary to allow a lower frequency or disabled Bus Clock (for example, Power Down Mode). The accessibility of the RTC registers Asynchronous Mode is described in **Section 9.5.5.5.1 Clock Operation (on Page 389)**.

9.5.5.3.1 RTC Diagrams

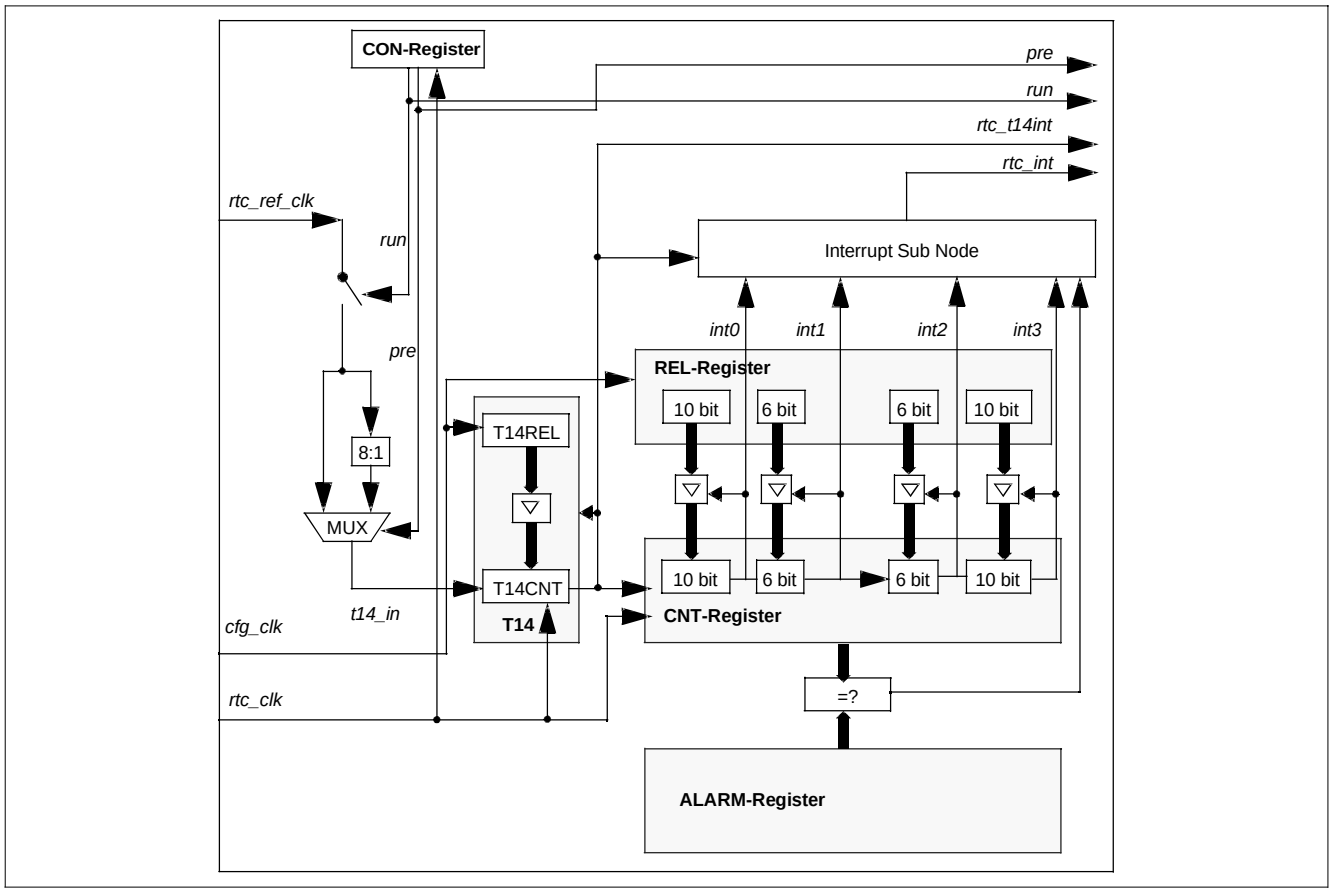


Figure 147 RTC Kernel Block Diagram

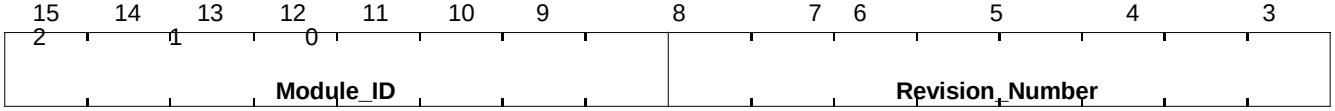
9.5.5.4 Registers

9.5.5.4.1 RTC Identification Register

RTC Identification Register

RTC_ID

RTC Identification Register **Reset value: 4903$_H$**

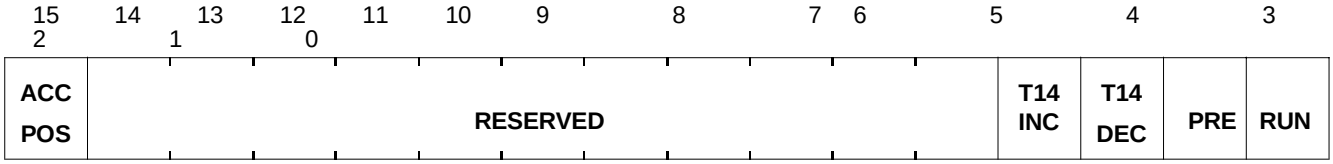


Field	Bits	Type	Description
Revision_Number	0:7	r	**Revision Number** These hard-wired bits are used for module revision numbering.
Module_ID	8:15	r	**Module Identification Number** These hard-wired bits are used for module identification numbering.

9.5.5.4.2 RTC Control Register

RTC_CON

RTC Control Register **Reset Value: 0003_H**



Field	Bits	Type	Description
RUN	0	rw	RTC Enable 0 Stopped 1 Runs
PRE	1	rw	RTC Input Source Pre-Scaler Enable 0 Disabled 1 Enabled
T14DEC	2	wh	Decrement T14 Timer Value 0 No action 1 T14 decremented The bit is automatically cleared by hardware after decrementing T14. Reading returns a zero.
T14INC	3	wh	Increment T14 Timer Value 0 No action 1 T14 incremented The bit is automatically cleared by hardware after incrementating T14. Reading returns a zero.
ACCPOS	15	rh	RTC Register Access Possible This bit indicates that synchronous read / write access to RTC registers is possible. Note that the Asynchronous/Synchronous mode select bit **RTC_CTRL.RTC_CLK_SEL** can always be accessed in either mode (see **Figure 148**). 0 No write access is possible, only asynchronous reads 1 Read and Write accesses are possible
RESERVED	14:4	r	Reserved, these bits must be left at their reset values.

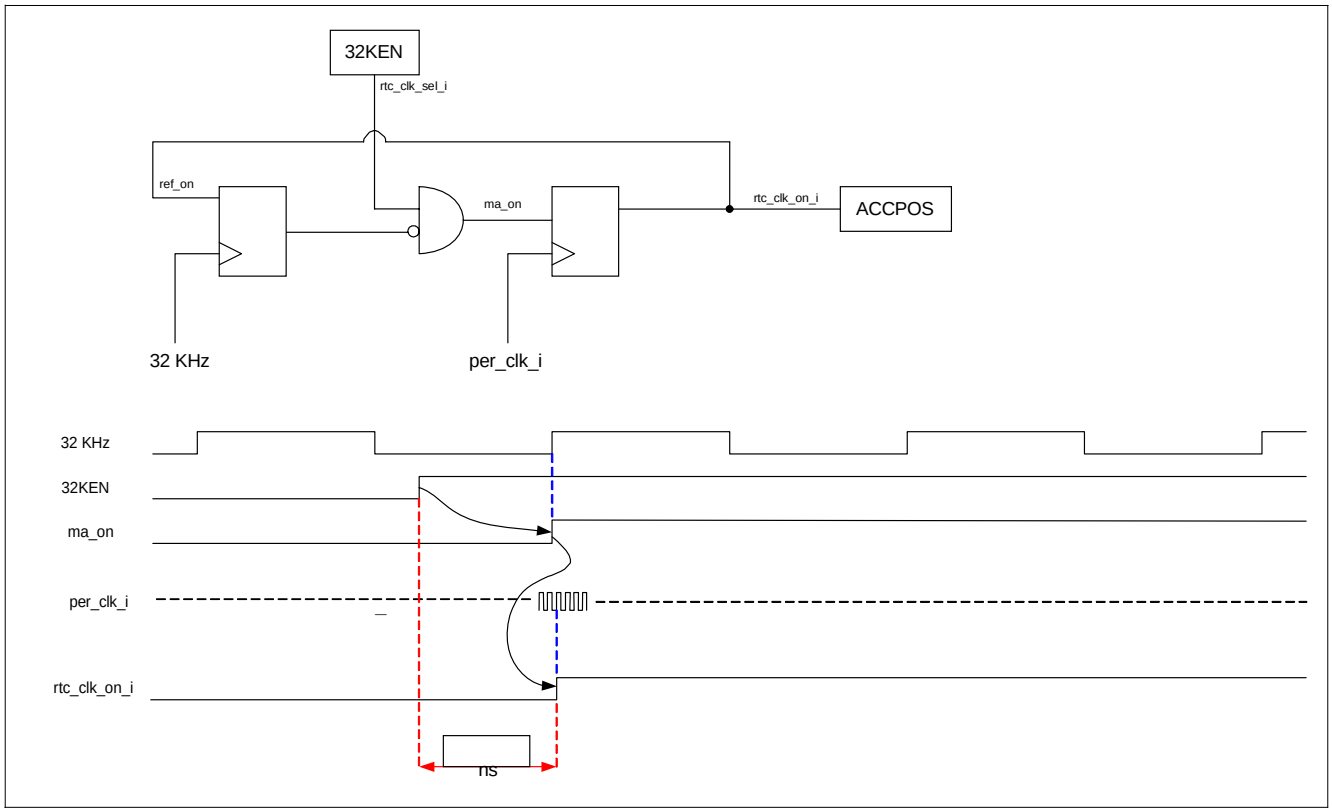


Figure 148 ACCPOS Bit Generation

Note: – ns is required to get the ACCPOS bit set to one when switching from the asynchronous to synchronous mode.

9.5.5.4.3 Prescaler Timer T14 Count Register

*Note: **RTC_T14_CNT** and **RTC_T14_REL** are physically located inside one 32-bit register in the RTC. The 32-bit register is updated when register **RTC_T14_CNT** is written to by software. Software must write the required data to the **RTC_T14_REL** address immediately before this to ensure that both 16-bit values are (simultaneously) written to the 32-bit register.*

RTC_T14_CNT

Timer T14 Count Register **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T14CNT															

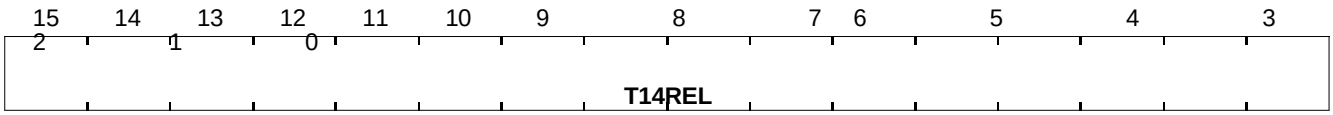
Field	Bits	Type	Description
T14CNT	15:0	rwh	T14 Counter

9.5.5.4.4 Prescaler Timer T14 Reload Register

*Note: Refer to **RTC_T14_CNT** for instructions on how to update this register.*

RTC_T14_REL

Timer T14 Reload Register **Reset Value: 0000_H**



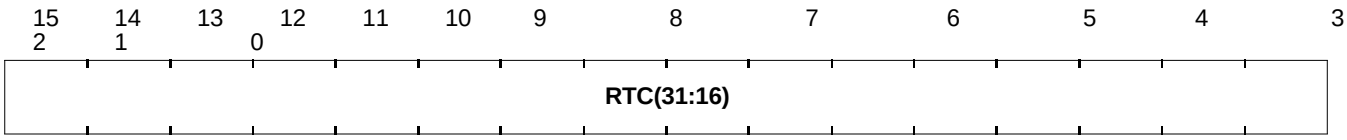
Field	Bits	Type	Description
T14REL	15:0	rw	T14 Reload Value

9.5.5.4.5 RTC Count Register (High Word)

*Note: **RTC_CNT_HI** and **RTC_CNT_LO** are physically located inside one 32-bit register in the RTC. The 32-bit register is updated when register **RTC_CNT_HI** is written to by software. Software must write the required data to the **RTC_CNT_LO** address immediately before this to ensure that both 16-bit values are (simultaneously) written to the 32-bit register.*

RTC_CNT_HI

RTC Count Register - High **Reset Value: 0000_H**



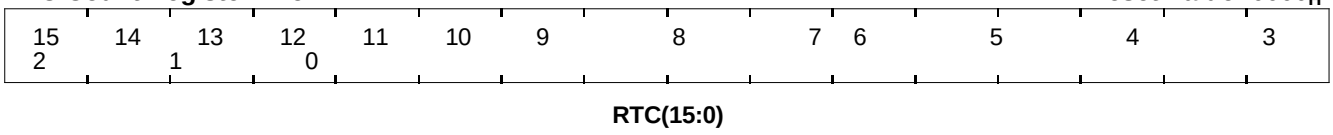
Field	Bits	Type	Description
RTC(31:16)	15:0	rwh	RTC Counter Bits 31:16

9.5.5.4.6 RTC Count Register (Low Word)

*Note: Refer to **RTC_CNT_HI** for instructions on how to update this register.*

RTC_CNT_LO

RTC Count Register - Low **Reset Value: 0000_H**



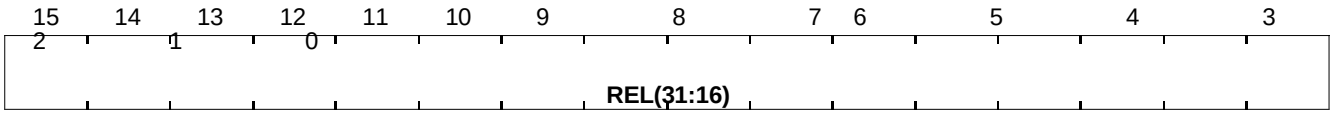
Field	Bits	Type	Description
RTC(15:0)	15:0	rwh	RTC Counter Bits 15:0

9.5.5.4.7 RTC Reload Register (High Word)

*Note: **RTC_REL_HI** and **RTC_REL_LO** are physically located inside one 32-bit register in the RTC. The 32-bit register is updated when register **RTC_REL_HI** is written to by software. Software must write the required data to the **RTC_REL_LO** address immediately before this to ensure that both 16-bit values are (simultaneously) written to the 32-bit register.*

RTC_REL_HI

RTC Reload Register - High **Reset Value: 0000_H**



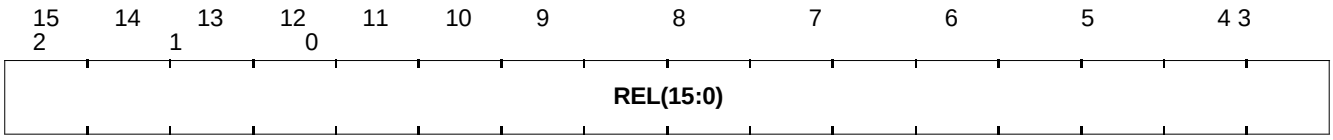
Field	Bits	Type	Description
REL(31:16)	15:0	rw	RTC Reload Value Bits 31:16

9.5.5.4.8 RTC Reload Register (Low Word)

Refer to **RTC_REL_HI** for instructions on how to update this register.

RTC_REL_LO

RTC Reload Register - Low **Reset Value: 0000_H**



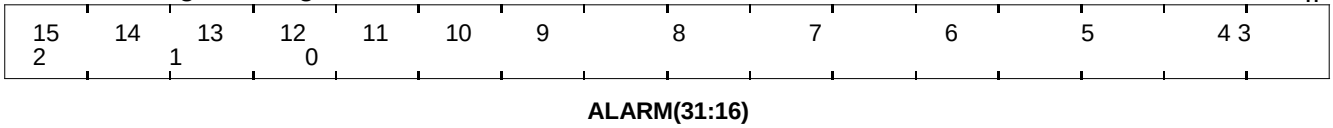
Field	Bits	Type	Description
REL(15:0)	15:0	rw	RTC Reload Value Bits 15:0

9.5.5.4.9 RTC Alarm Register (High Word)

Refer to **RTC_ALARM_HI** and **RTC_ALARM_LO** are physically located inside one 32-bit register in the RTC. The 32-bit register is updated when register **RTC_ALARM_HI** is written to by software. Software must write the required data to the **RTC_ALARM_LO** address immediately before this to ensure that both 16-bit values are (simultaneously) written to the 32-bit register.

RTC_ALARM_HI

RTC Alarm Register - High Word **Reset Value: $FFFF_H$**

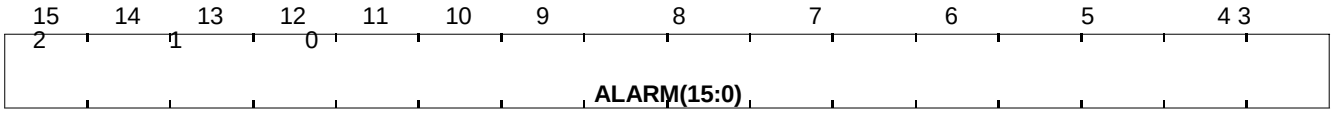


Field	Bits	Type	Description
ALARM(31:16)	15:0	rw	Alarm Time Bits 31:16

9.5.5.4.10 RTC Alarm Register (Low Word)

RTC_ALARM_LO

RTC Alarm Register - Low Word **Reset Value: $FFFF_H$**



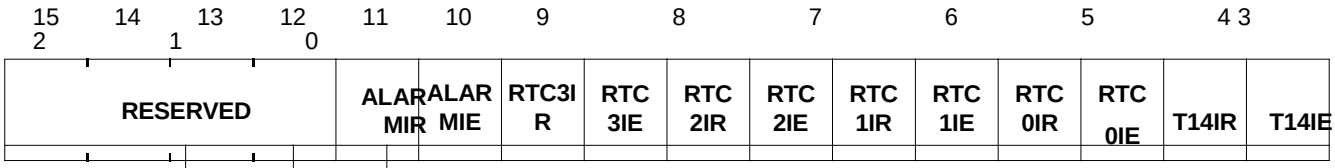
Field	Bits	Type	Description
ALARM(15:0)	15:0	rw	Alarm Time Bits 15:0

9.5.5.4.11 RTC Interrupt Sub Node Control

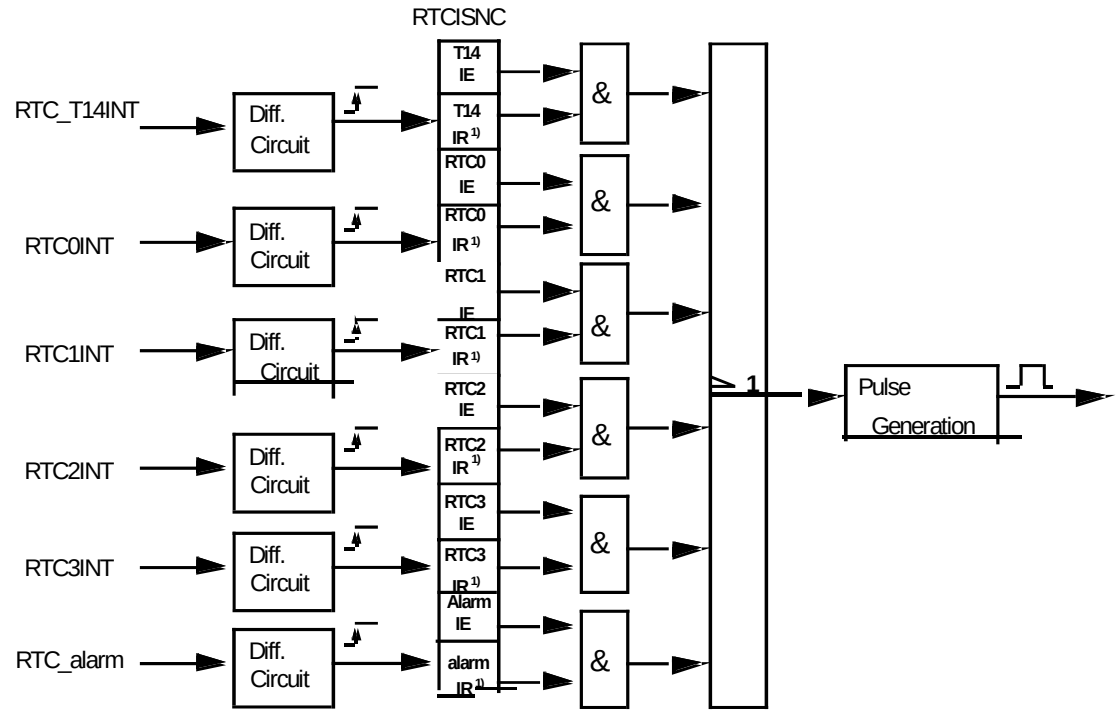
See **Figure 149 Differentiating Circuits for RTC interrupt**.

RTC_ISNC

RTC Interrupt Sub Node Control **Reset Value: 0000_H**



Field	Bits	Type	Description
T14IE	0	rw	T14 Overflow Interrupt Enable Control 0 Disables interrupt request 1 Enables interrupt request
T14IR	1	rw	T14 Overflow Interrupt Request 0 No request pending 1 This source has raised an interrupt request Write 0 to clear this interrupt bit.
RTCxIE	8,6,4,2	rw	Interrupt Enable Control x 0 Disables interrupt request 1 Enables interrupt request
RTCxIR	9,7,5,3	rw	Interrupt Request 0 No request pending 1 This source has raised an interrupt request Write 0 to clear this interrupt bit.
ALARMIE	10	rw	Alarm Interrupt Enable Control 0 Disables interrupt request 1 Enables interrupt request
ALARMIR	11	rw	Alarm Interrupt Request (bit protected) 0 No request pending 1 This source has raised an interrupt request Write 0 to clear this interrupt bit.
RESERVED	15:12	r	Reserved, these bits must be left at their reset values.



1) *The ISR (Interrupt Service Routine) must delete the IR flag in register RTCISNC manually. Otherwise no further interrupts can be detected.*

Figure 149 Differentiating Circuits for RTC interrupt

9.5.5.5 Functions

9.5.5.5.1 Clock Operation

The following clocking issues have to be considered when the RTC module runs in Synchronous Mode (**RTC_CTRL.RTC_CLK_SEL** = 1).

The module derives its Kernel Clock (*rtc_clk_s*) and the Bus Clock (*bpi_clk_s*) from the *per_clk_i* input by Clock Gating with the input signals *rtc_gating_en_inv_i* and *bpi_clk_en_i*. The following relation between the Kernel and Bus Clocks is required:

- The Kernel Clock has the same speed, or is faster than, the Bus Clock.
- When a Bus Clock pulse is active then a Kernel Clock pulse must be also active.

In Asynchronous Mode (**RTC_CLK_SEL** = 0) the Kernel Clock is driven by *clk_32k* and no write accesses to the registers in this clock domain are possible. Write access is possible to those which are indicated as being in the Bus Clock domain in **Table 83**. A read access is possible, but correct read data can not be guaranteed due to the possibility of an asynchronous data change during the read access.

Table 83 Register Assignment to Clock and Reset Domains

Register	Kernel Clock	Bus Clock	CFG Clock	RTC reset	Bus Reset
RTC_ID	-		-	-	- -
RTC_CON	X		- -	X	-
RTC_T14_CNT	X		- -	X	-
RTC_T14_REL	-			X	X -
RTC_CNT_LO/RTC_CNT_HI	X		- -	X	-

Table 83 Register Assignment to Clock and Reset Domains (cont'd)

Register	Kernel Clock	Bus Clock	CFG Clock	RTC reset	Bus Reset
RTC_REL_LO/RTC_REL_HI	-			X	X -
RTC_ISNC	X	-	-	X	-
RTC_CTRL	-		X		X -
RTC_ALARM_LO/RTC_ALARM_HI	X	-	-	X	-

9.5.5.5.2 RTC Control

The operating behavior of the RTC module is controlled by the **RTC_CON (on Page 384)** register. The RTC starts counting when **RTC_CON.RUN** is set. After a software reset the run bit is set and the RTC automatically starts operation (note that the RTC module is only reset by software, not by the power on reset). The bit **RTC_CON.RTCPRE** selects a prescaler which divides the counting clock by factor 8. Activating the prescaler reduces the resolution of the reload counter T14. If the prescaler is not activated, the RTC may lose counting clocks on switching from asynchronous to synchronous mode and back. This effect can be avoided by activating the prescaler.

Setting the bits **RTC_CON.T14DEC** or **RTC_CON.T14INC** decrements or increments the T14 timer with the next count event. If at the next count event a reload has to be executed, then an increment operation is delayed until the next count event occurs. The in/decrement function can only be used if register T14REL is not equal to $FFFF_H$. The in/decrement bits are cleared by hardware after the decrement/increment operation.

9.5.5.5.3 System Clock Operation

A real time system clock can be maintained that represents the current time and date. If the RTC module is not effected by a system reset, it keeps running also during idle mode and power down mode.

The maximum resolution (minimum step width) for this clock information is determined by timer T14's input clock. The maximum usable time span is achieved when **RTC_T14_REL.T14REL** is loaded with 0000_H and so T14 is divided by 2^{16}.

9.5.5.5.4 Cyclic Interrupt Generation

The RTC module can generate an interrupt request RTC_T14INT whenever timer T14 overflows and is reloaded. For example, this interrupt request may be used to provide a system time tick independent of the Kernel Clock frequency without loading the general purpose timers, or to wake up regularly from idle mode. The T14 overflow interrupt (RTC_T14INT) cycle time can be adjusted via the timer T14 reload register **RTC_T14_REL.T14REL**. This interrupt request is also OR'ed with all other interrupts of the RTC via the RTC interrupt sub node **RTC_ISNC (on Page 388)**.

9.5.5.5.5 Alarm Interrupt Generation

The RTC module can also provide an alarm interrupt. The alarm time is written to the registers **RTC_ALARM_LO** and **RTC_ALARM_HI (on Page 387)**. One **RTC_CNT_LO (on Page 386)** cycle after the value in registers **RTC_CNT_HI** and **RTC_CNT_LO** equals the value programmed in **RTC_ALARM_HI** and **RTC_ALARM_LO**, an internal interrupt request is generated. It is OR'ed in the interrupt sub node register **RTC_ISNC (on Page 388)** with the other RTC interrupts to generate one interrupt request RTC_INT.

9.5.5.5.6 48-bit Timer Operation

The concatenation of the 16-bit reload timer T14 and the 32-bit RTC timer can be regarded as a 48-bit timer which counts with the RTC count input frequency, (RTC_REF_CLK) divided by the fixed prescaler, if the prescaler is selected. The reload registers **RTC_T14_REL**, **RTC_REL_LO**, and **RTC_REL_HI** should be cleared to get the maximum usable time span of 2^{48} ($H10^{14}$) T14 input clocks.

9.5.5.5.7 Hardware Dependent RTC Accuracy

The RTC has different counting accuracies, depending on the operating mode (with or without prescaler). There is only an impact on the counting accuracy when switching the RTC from synchronous mode to asynchronous mode and back.

Table 84 Impact on Counting Accuracy

Operating Mode	Inaccuracy in T14 Counting Ticks
Without prescaler	+0 / -0.5
With prescaler	+0 / -0

9.5.5.5.8 RTC Disable Functionality

The Peripheral Kernel of the RTC can be disabled, if the RTC functionality is not used. In this case only the bus interface, interrupt logic and pad tristate control is enabled. Disabling the RTC module reduces the power consumption and the generated noise of the complete system.

9.5.6 RTC Power Supply Concept

The RTC module logic and the internal 32k oscillator are supplied by VDD_RTC. The VDD_RTC operating range is specified in the chapter Operating Conditions. Generally there are two ranges:

1. RTC and oscillator operating range **for RTC stand-alone operation** (for example, mobile station off, only RTC is running).
2. RTC and oscillator operating range **for communication with MCU**.

The voltages are given in**Section 11.2.1.2 Voltages**.

9.6 GPT 1 and 2

System Integration

- Supply domain: VDD_LD1
- Chip internal interfaces:
  - Clock domain: Refer to **Section 7.2.1.3 Sub-System Clocks and Enables (on Page 67)** and see **Figure 18 Clock Enable (on Page 68)**.
  - Bus domain: PD-Bus
- Interrupt sources:
- Monitor Pins: Refer to **Section 9.7.10 Internal Signal Monitoring (on Page 435)**.

9.6.1 E-GOLDvoice Specific Restrictions

The functionalities of the standard GPT12 described below are restricted in E-GOLDvoice.

Only the pins mentioned below are available in the pinlist at package level:

- Timer Block 1:
  - pins available: T2EUD / T4EUD / T2IN
- Timer Block 2:
  - pins available: T6OUT

9.6.2 Introduction

The General Purpose Timer Unit (GPT12) provides very flexible multifunctional timer structures that may be used for timing, event counting, pulse width measurement, pulse generation, frequency multiplication, and other purposes. The GPT12 incorporates five 16-bit timers grouped into the two timer blocks: Block1 (GPT1) and Block2 (GPT2). Each timer in each block can operate independently in a number of different modes such as Gated Timer Mode or Counter Mode; or, each timer can be concatenated with another timer of the same block.

Block 1 contains three timers/counters with a maximum resolution of $f_{hw\_clk}/4$ (for information about hw_clk, refer to **Table 99 GPT12 Register Summary (on Page 414)**). The auxiliary Timers of GPT1 may optionally be configured as reload or capture registers for the core Timer.

Block 2 contains 2 timers/counters with a maximum resolution of $f_{hw\_clk}/2$. An additional **CAPREL register** supports capture and reload operation with extended functionality.

Notes

1. *The Core Timer T6 may be concatenated with timers of other on-chip peripherals such as a CAPCOM unit.*
2. *The GPT12 can receive two clocks: f_{hw_clk} (the normal operation clock) and f_{cfg_clk} (only used for the configuration registers).*

The following summary identifies all features to be supported by the GPT12:

- Timer Block 1:
  - Maximum resolution of $f_{hw\_clk}/4$
  - Three independent timers/counters.
  - Concatenation of timers/counters can be done.
  - Four operating modes (Timer Mode, Gated Timer Mode, Counter Mode, Incremental Interface Mode).
  - Separate interrupt nodes.
- Timer Block 2:
  - Maximum resolution of $f_{hw\_clk}/2$
  - Two independent timers/counters.
  - Concatenation of timers/counters can be done.
  - Three operating modes (Timer Mode, Gated Timer Mode, Counter Mode).
  - Extended capture/reload functions via 16-bit Capture/Reload register **CAPREL**.
  - Separate interrupt nodes.

9.6.3 Overview

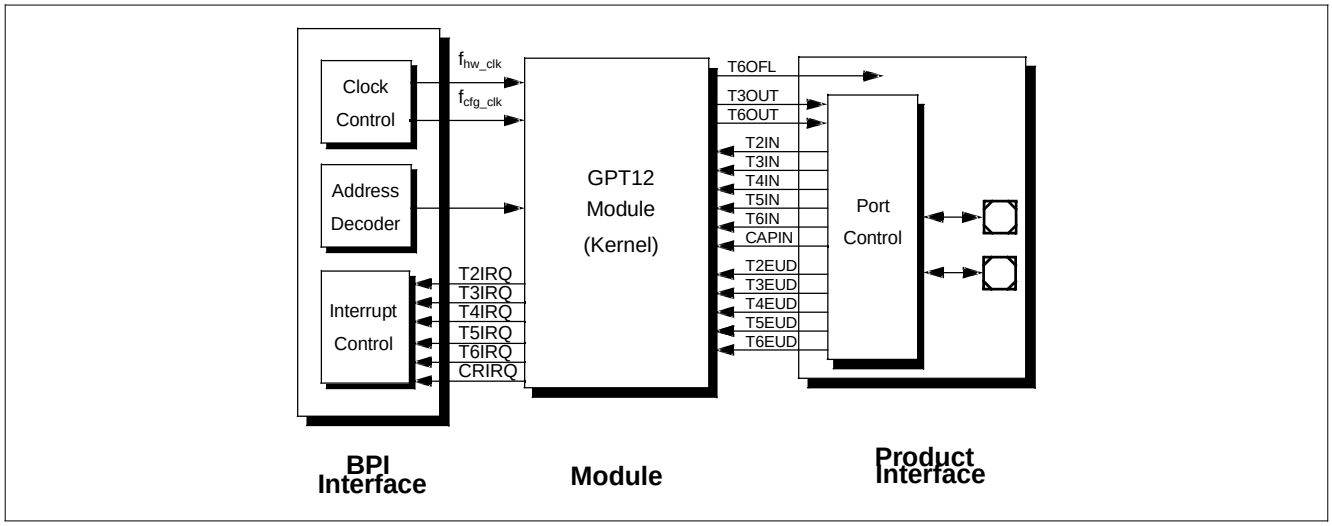


Figure 150 GPT12 Interface Diagram

Note: Bus Peripheral Interface (BPI) is the connection to the on-chip bus system.

9.6.4 Kernel Description

9.6.4.1 Functional Description of Timer Block 1

All three timers of Block 1 (T2, T3, T4) can run in 4 basic modes: Timer Mode, Gated Timer Mode, Counter Mode and Incremental Interface Mode. All timers can count up or down. Each timer of Block 1 is controlled by a separate control register **TxCON**.

Each timer has an input line, TxIN, associated with it which serves as the gate control in Gated Timer Mode, or as the count input in Counter Mode. The count direction (up/down) may be programmed via software or may be dynamically altered by a signal at an external control input line, External Up/Down Control Input TxEUD. An overflow/underflow of core Timer T3 is indicated by the Output Toggle Latch T3OTL whose state may be output on related line T3OUT . Additionally, the auxiliary Timers T2 and T4 may be concatenated with core Timer T3 or may be used as capture or reload registers for core Timer T3. Concatenation of T3 with other timers is provided through line T3OTL.

The current contents of each timer can be read or modified by the MCU by accessing the corresponding timer registers T2, T3, or T4, located in the non-bitaddressable Special Function Register (SFR) space. When any of the timer registers is written to by the MCU in the state immediately before a timer increment, decrement, reload, or capture is to be performed, the MCU write operation has priority in order to guarantee correct results.

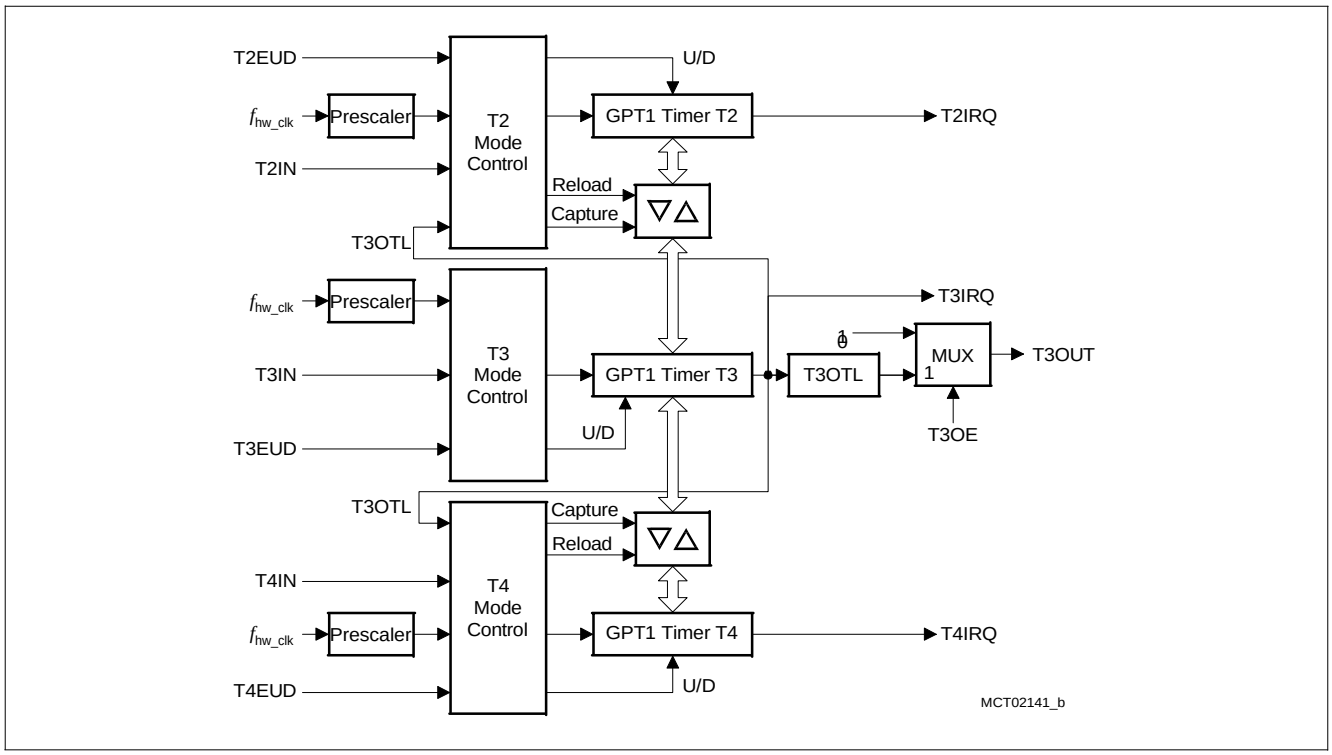


Figure 151 Structure of Timer Block 1

9.6.4.1.1 Core Timer T3

The operation of core Timer T3 is controlled by its bitaddressable control register **T3CON**.

Run Control

The timer can be started or stopped by software through bit **T3CON.T3R**. Setting bit **T3CON.T3R** will start the timer; clearing **T3CON.T3R** stops the timer.

In Gated Timer Mode, the timer will run only if **T3CON.T3R** is set and the gate is active (high or low, as programmed).

*Note: When bit **T2CON.T2RC** or **T4CON.T4RC** is set, **T3CON.T3R** will also control (start and stop) auxiliary Timer T2/T4.*

Count Direction Control

The count direction of the core Timer T3 can be controlled either by software or by the external input line, T3EUD. These options are selected by bits **T3CON.T3UD** and **T3CON.T3UDE**. When the up/down control is set by software (bit **T3CON.T3UDE** is cleared), the count direction can be altered by setting or clearing bit **T3CON.T3UD**. When **T3CON.T3UDE** is set, line T3EUD is selected to be the controlling source of the count direction. However, bit **T3CON.T3UD** can still be used to reverse the actual count direction, as shown in **Table 85**. If **T3CON.T3UD** is cleared and line T3EUD shows a low level, the timer is counting up. With a high level at T3UD the timer is counting down. If **T3CON.T3UD** is set, a high level at line T3EUD specifies counting up, and a low level specifies counting down. The count direction can be changed whether or not the timer is running or not.

Note: When line T3EUD is used as external count direction control input, its associated port pin must be configured as input.

Table 85 Core Timer T3 Count Direction Control

Line T3EUD	Bit T3CON.T3UDE	Bit T3CON.T3UD	Count Direction
X	0	0	Counts Up
X	0	1	Counts Down
0	1	0	Counts Up
1	1	0	Counts Down
0	1	1	Counts Down
1	1	1	Counts Up

Note: The direction control works in same way for core Timer T3 and for auxiliary Timers T2 and T4.

Timer 3 Overflow/Underflow Monitoring

An overflow or underflow of Timer T3 will clock bit **T3CON.T3OTL**. **T3CON.T3OTL** can also be set or reset by software. Bit **T3CON.T3OE** (overflow/underflow output enable) enables the state of **T3CON.T3OTL** to be monitored via an external line, T3OUT. If this line is linked to an external port pin (configured as output), T3OUT can be used to control external hardware.

Additionally, **T3CON.T3OTL** can be used in conjunction with auxiliary Timers T2 and T4. In this case **T3CON.T3OTL** serves as input for the counter function or as trigger source for the reload function of T2 and T4. T3OTL is internally connected for this functionality and it is not necessary to enable overflow/underflow output on T3OUT for this purpose.

Timer 3 in Timer Mode

Timer Mode for the core Timer T3 is selected by setting bit field **T3CON.T3M** to 000_B.

A block diagram of T3 in Timer Mode is shown in **Figure 152**.

In this mode, T3 is clocked with the module clock f_{hw_clk} divided by a programmable prescaler block, controlled by bit field **T3CON.T3I** and **T3CON.T3BPS1**. The input frequency f_{T3} for Timer T3 and its resolution r_{T3} are scaled linearly with lower module clock frequencies, as can be seen from the following formula:

$$f_{T3} = \frac{f_{hw\_clk}}{BPS1 * 2^{<T3I>}} \qquad r_{T3}\,[ms] = \frac{BPS1 * 2^{<T3I>}}{f_{hw\_clk}\,[MHz]} \tag{65}$$

*Note: <BPS1> represents the prescaler value of the prescaler part controlled by bit field **T3CON.T3BPS1**. For the values, see the bit description in register **T3CON**.*

Table 86 Timer 3 Input Parameter Selection: Timer Mode and Gated Timer Mode (all bits in T3CON)

T3I	Prescaler for f_{hw_clk} (T3BPS1 = 00)	Prescaler for f_{hw_clk} (T3BPS1 = 01)	Prescaler for f_{hw_clk} (T3BPS1 = 10)	Prescaler for f_{hw_clk} (T3BPS1 = 11)
000	8	4		32 16
001	16	8	64	32
010	32	16	128	64
011	64	32	256	128
100	128	64	512	256
101	256	128	1024	512

Table 86 Timer 3 Input Parameter Selection: Timer Mode and Gated Timer Mode (all bits in T3CON)

T3I	Prescaler for f_{hw_clk} (T3BPS1 = 00)	Prescaler for f_{hw_clk} (T3BPS1 = 01)	Prescaler for f_{hw_clk} (T3BPS1 = 10)	Prescaler for f_{hw_clk} (T3BPS1 = 11)
110	512	256	2048	1024
111	1024	512	4096	2048

Table 87 Example for Timer 3 Frequencies and Resolutions

f_{hw_clk} [MHz]	T3CON.T3I	T3CON.T3BPS1	f_{T3} [KHz]	r_{T3} [µs]
40	7	10	9.77	102.4
40	0	01	10000	0.1
50	0	00	6250	0.16
50	4	11	195.31	5.12
50	7	10	12.20	81.97

This formula also applies to the Gated Timer Mode of T3 and to the auxiliary Timers T2 and T4 in Timer Mode and Gated Timer Mode.

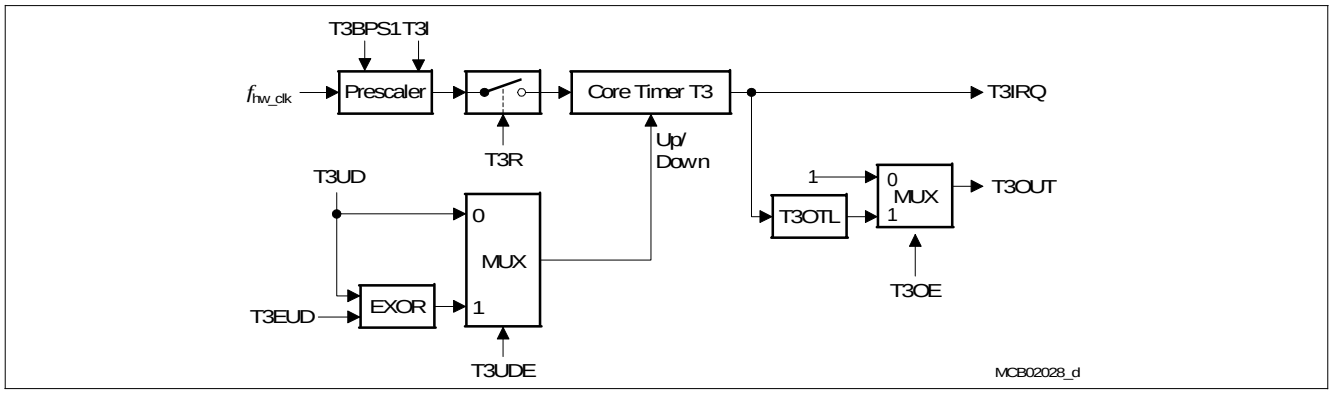


Figure 152 Block Diagram of Core Timer T3 in Timer Mode

Timer 3 in Gated Timer Mode

Gated Timer Mode for the core Timer T3 is selected by setting bit field **T3CON.T3M** to 010_B or 011_B.
Bit **T3CON.T3M(0)** (**T3CON(3)**) selects the active level of the gate input. The same options for the input frequency are available in Gated Timer Mode as for the Timer Mode. However, the input clock to the timer in this mode is gated by the external input line T3IN (Timer T3 External Input).

Note: An associated port pin should be configured as input.

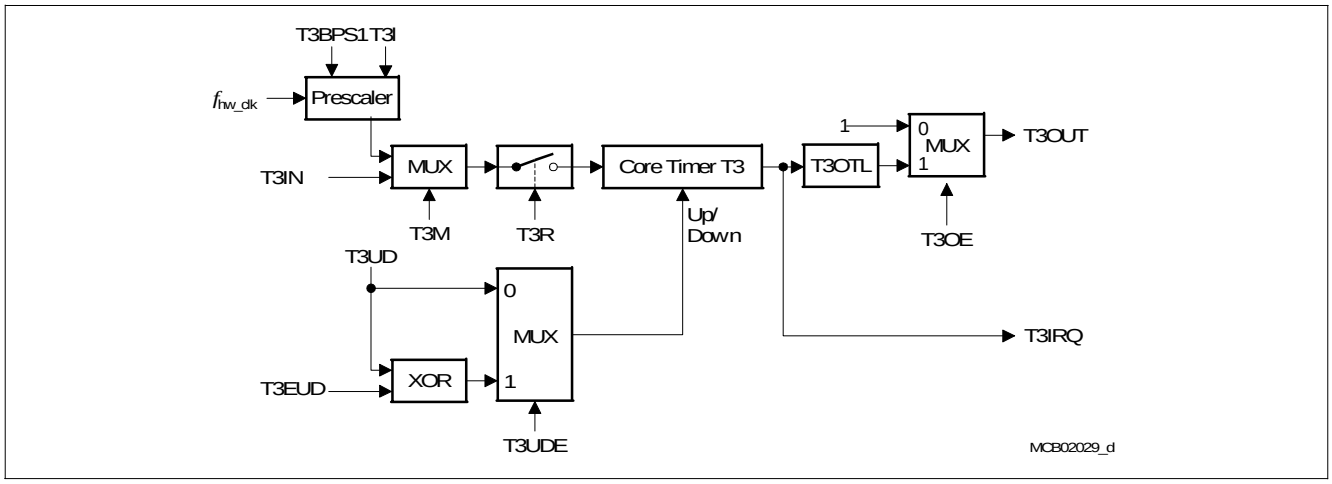


Figure 153 Block Diagram of Core Timer T3 in Gated Timer Mode

If **T3CON.T3M** = 010_B, the timer is enabled when T3IN shows a low level. A high level at this line stops the timer. If **T3CON.T3M** = 011_B, line T3IN must have a high level to enable the timer. Additionally, the timer can be turned on or off by software using bit **T3CON.T3R**. The timer will run only if **T3CON.T3R** is set and the gate is active. It will stop if either T3R is cleared or the gate is inactive.

Note: A transition of the gate signal at line T3IN does not cause an interrupt request.

Timer 3 in Counter Mode

Counter Mode for core Timer T3 is selected by setting bit field **T3CON.T3M** to 001_B. In the Counter Mode, Timer T3 is clocked by a transition at the external input line T3IN. The event causing an increment or decrement of the timer can be a positive, a negative, or both a positive and a negative transition at this line. bit field **T3CON.T3I** selects the triggering transition (see **Table 88**).

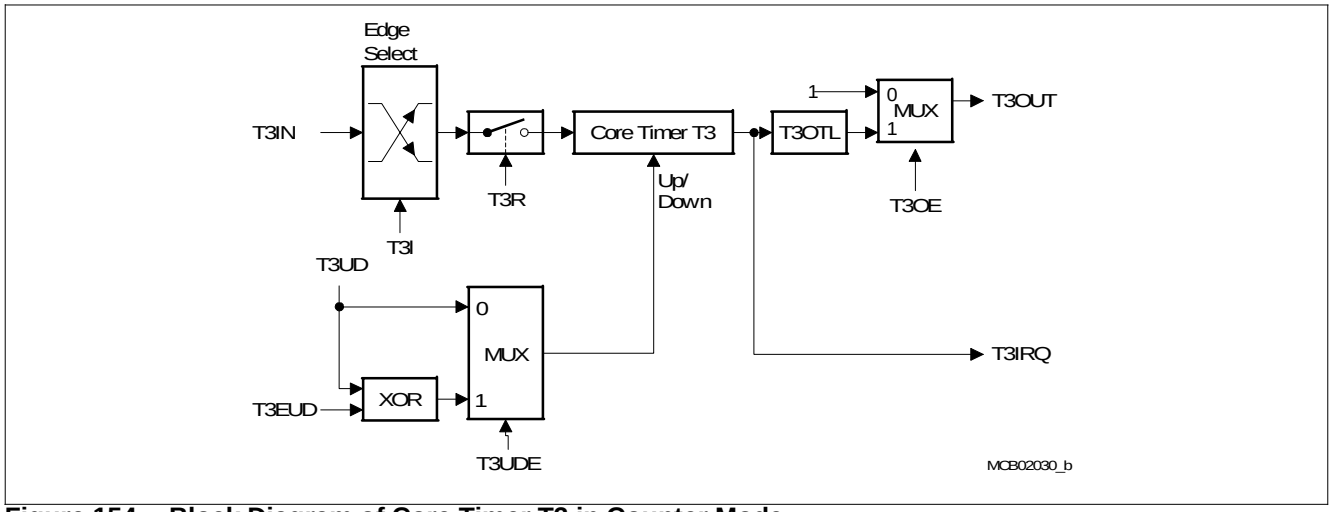


Figure 154 Block Diagram of Core Timer T3 in Counter Mode

Table 88 Core Timer T3 (Counter Mode) Input Edge Selection

T3CON.T3I	Triggering Edge for Counter Increment / Decrement
0 0 0	None. Counter T3 is disabled
0 0 1	Positive transition (rising edge) on T3IN
0 1 0	Negative transition (falling edge) on T3IN

Table 88 Core Timer T3 (Counter Mode) Input Edge Selection (cont'd)

T3CON.T3I	Triggering Edge for Counter Increment / Decrement
0 1 1	Any transition (rising or falling edge) onT3IN
1 X X	Reserved. Do not use these combinations.

For Counter Mode operation, a port pin associated with line T3IN must be configured as input. The maximum input frequency allowed in Counter Mode is $f_{hw\_clk}/8$ (**T3CON.T3BPS1** = 01). To ensure that a transition of the count input signal applied to T3IN is correctly recognized, its level should be held high or low for at least 4 f_{hw_clk} cycles (**T3CON.T3BPS1** = '01') before it changes.

Timer 3 in Incremental Interface Mode

Incremental Interface Mode for the core Timer T3 is selected by setting bit field **T3CON.T3M** to 110_B or 111_B. In Incremental Interface Mode the two inputs associated with Timer T3 (T3IN, T3EUD) are used to interface to an external incremental encoder. T3 is clocked by each transition on one or both of the external input lines which gives 2-fold or 4-fold resolution of the encoder input.

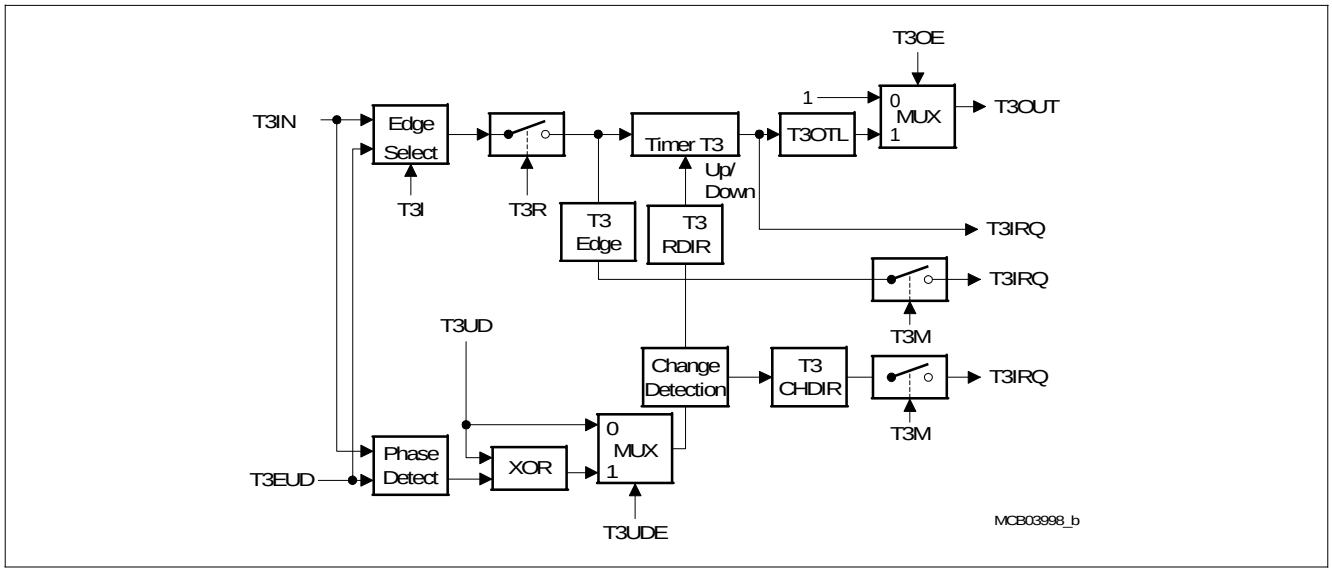


Figure 155 Block Diagram of Core Timer T3 in Incremental Interface Mode

bit field **T3CON.T3I** selects the triggering transitions (see **Table 89**). The sequence of the transitions of the two input signals is evaluated and generates count pulses as well as the direction signal. Depending on whether Rotation Detection Mode

(**T3CON.T3M** = 110_B) or Edge Detection Mode (**T3CON.T3M** = 111_B) is chosen, an interrupt request on T3IRQ is generated. For the Rotation Detection Mode, an interrupt is generated each time the count direction of Timer T3 changes. For the Edge Detection Mode an interrupt is generated each time a count action for Timer T3 occurs. Count direction, changes in the count direction and count requests are monitored by status bits **T3CON.T3RDIR**, **T3CON.T3CHDIR**, and **T3CON.T3EDGE**. T3 is modified automatically according to the speed and direction of the incremental encoder. Therefore, the contents of Timer T3 always represents the encoder's current position.

Table 89 Core Timer T3 (Incremental Interface Mode) Input Edge Selection

T3CON.T3I	Triggering Edge for Counter Increment / Decrement
0 0 0	None. Counter T3 stops.
0 0 1	Any transition (rising or falling edge) on T3IN.
0 1 0	Any transition (rising or falling edge) on T3EUD.

Table 89 Core Timer T3 (Incremental Interface Mode) Input Edge Selection (cont'd)

T3CON.T3I	Triggering Edge for Counter Increment / Decrement
0 1 1	Any transition (rising or falling edge) on any T3 input (T3IN or T3EUD).
1 X X	Reserved. Do not use these combinations.

The incremental encoder can be connected directly to the microcontroller without external interface logic. In a standard system, however, comparators will be employed to convert the encoder's differential outputs (such as A, $\overline{A}$) to digital signals (such as A in **Figure 156**). This greatly increases noise immunity.

Note: The third encoder output T0, which indicates the mechanical zero position, may be connected to an external interrupt input and trigger a reset of Timer T3.

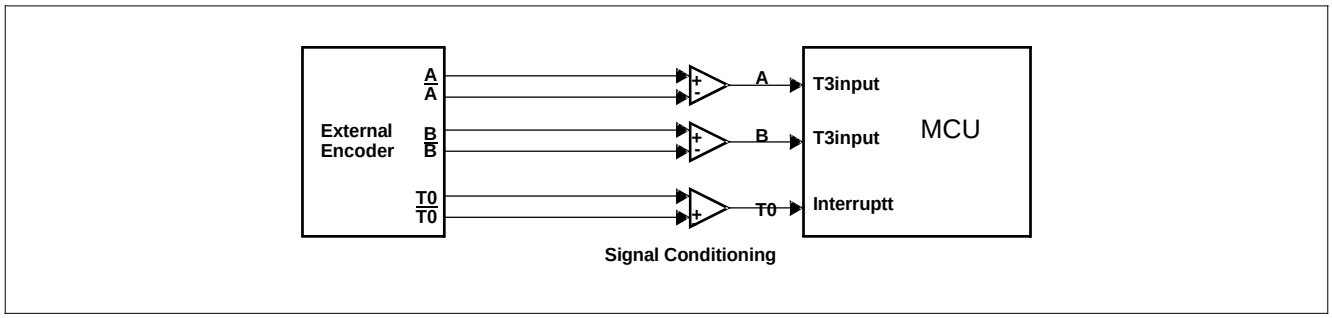


Figure 156 Interfacing the Encoder to the Microcontroller

The following conditions must be met for Incremental Interface Mode operation:

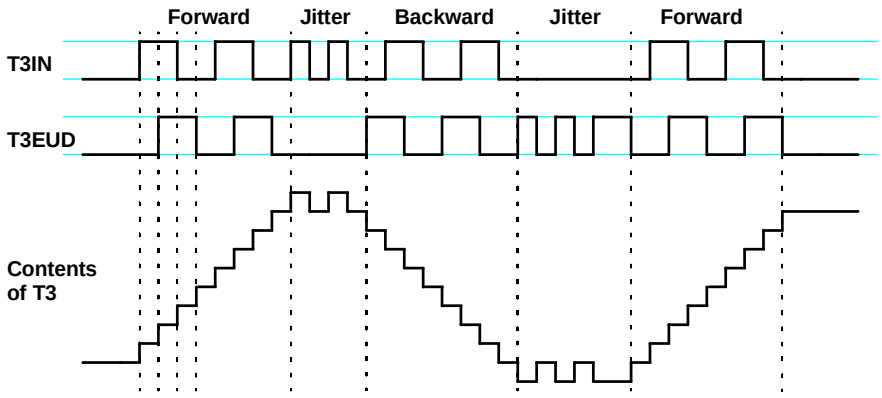
- bit field **T3CON.T3M** must be 110_B or 111_B.
- Pins associated to lines T3IN and T3EUD must be configured as input.
- Bit **T3CON.T3UDE** must be set to enable external direction control.

The maximum input frequency allowed in Incremental Interface Mode is $f_{hw\_clk}/8$ (**T3CON.T3BPS** = 01_B). To ensure that a transition of any input signal is correctly recognized, its level should be held high or low for at least 4 f_{hw_clk} cycles (**T3CON.T3BPS** = 01_B) before it changes.
In Incremental Interface Mode the count direction is automatically derived from the sequence in which the input signals change, which corresponds to the rotation direction of the connected sensor. **Table 90** summarizes the possible combinations.

Table 90 Core Timer T3 (Incremental Interface Mode) Count Direction

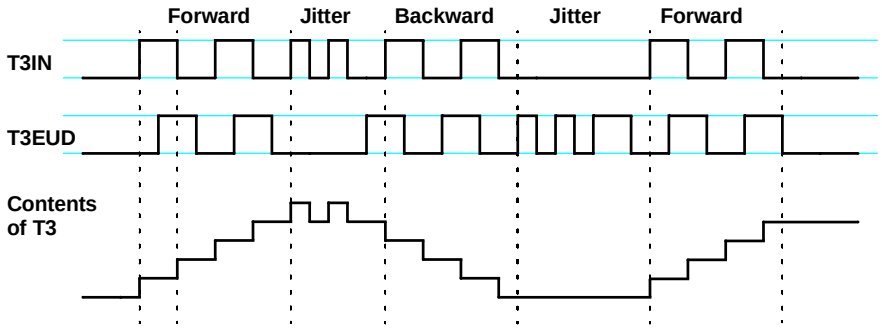
Level on respective other input	T3IN Input		T3EUD Input	
	Rising ↗	↘ Falling	Rising ↗	↘ Falling
High	Down	Up	Up	
Low	Up	Down	Down	Up

Figure 157 and **Figure 158** give examples of the operation of T3 to illustrate count signal generation and direction control. Each example also shows how input jitter, which might occur if the sensor rests near to one of its switching points, is compensated.



Note: This example shows the timer behavior assuming that T3 counts upon

Figure 157 Evaluation of the Incremental Encoder Signals



Note: This example shows the timer behavior assuming that T3 counts upon

Figure 158 Evaluation of the Incremental Encoder Signals

Note: Timer T3 operating in Incremental Interface Mode automatically provides information on the sensor's current position. Dynamic information (speed, acceleration, deceleration) may be obtained by measuring the incoming signal periods.

9.6.4.1.2 Auxiliary Timers T2 and T4

Both auxiliary Timers T2 and T4 have exactly the same functionality. They can be configured for Timer Mode, Gated Timer Mode, Counter Mode, or Incremental Interface Mode with the same options for the timer frequencies and the count signal as the core Timer T3. In addition to these 4 counting modes, the auxiliary timers can be concatenated with the core Timer T3, or they may be used as reload or capture registers in conjunction with the core Timer T3.

The individual configuration for Timers T2 and T4 are determined by their bitaddressable control registers **T2CON** and **T4CON**, which are both organized identically. Note that functions which are present in all 3 timers of Timer Block 1 are controlled in the same bit positions and manner in each of the specific control registers.

Run control for auxiliary Timers T2 and T4 can be handled by the associated run control bit **T2CON.T2R** and **T4CON.T4R**. Alternatively, a remote control option (**T2CON.T2RC**, **T4CON.T4RC** set) may be enabled to start and stop T2/T4 via the run bit **T3CON.T3R** of core Timer T3.

Timers T2 and T4 in Timer Mode or Gated Timer Mode

When the auxiliary Timers T2 and T4 are programmed to Timer Mode or Gated Timer Mode, their operation is the same as described for the core Timer T3. The descriptions, figures and tables apply accordingly with two exceptions:

- There is no TxOUT output line for T2 and T4
- Overflow/underflow monitoring is not supported (no bit TxOTL in registers TxCON)

Table 91 Timer 2,4 Input Parameter Selection: Timer Mode and Gated Timer Mode

T2CON.T2I or T4CON.T4I	Prescaler for f_{hw_clk} (T3CON.T3BPS1 = 00)	Prescaler for f_{hw_clk} (T3CON.T3BPS1 = 01)	Prescaler for f_{hw_clk} (T3CON.T3BPS1 = 10)	Prescaler for f_{hw_clk} (T3CON.T3BPS1 = 11)
000	8	4	32	16
001	16	8	64	32
010	32	16	128	64
011	64	32	256	128
100	128	64	512	256
101	256	128	1024	512
110	512	256	2048	1024
111	1024	512	4096	2048

Timers T2 and T4 in Counter Mode

In Counter Mode, Timers T2 and T4 can be clocked either by a transition at the respective external input line TxIN, or by a transition of T3OTL.

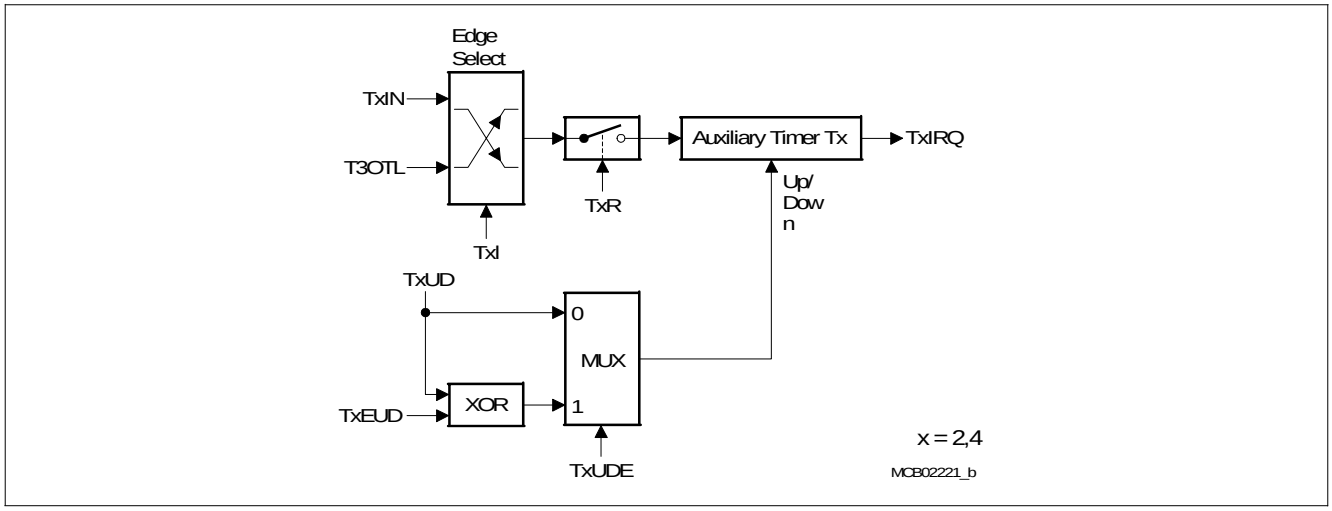


Figure 159 Block Diagram of an Auxiliary Timer in Counter Mode

The event causing an increment or decrement of a timer can be a positive, a negative, or both a positive and a negative transition at either the respective input line, or at the output toggle latch T3OTL.

Bit fields **T2CON.T2I** and **T4CON.T4I** select the triggering transition (see **Table 92**).

Table 92 Auxiliary Timer (Counter Mode) Input Edge Selection

T2I/T4I	Triggering Edge for Counter Increment / Decrement
X 0 0	None. Counter T2 or T4 is disabled
0 0 1	Positive transition (rising edge) on T2IN or T4IN
0 1 0	Negative transition (falling edge) on T2IN or T4IN
0 1 1	Any transition (rising or falling edge) on T2IN or T4IN
1 0 1	Positive transition (rising edge) of T3OTL
1 1 0	Negative transition (falling edge) of T3OTL
1 1 1	Any transition (rising or falling edge) of T3OTL

Note: Only state transitions of T3OTL caused by the overflow/underflow of T3 will trigger the counter function of T2/T4. Modifications of T3OTL via software will NOT trigger the counter function of T2/T4.

For counter operation, an external pin associated to line TxIN must be configured as input. The maximum input frequency allowed in Counter Mode is $f_{hw\_clk}/8$ (**T3CON.T3BPS1** = 01). To ensure that a transition of the count input signal which is applied to TxIN is correctly recognized, its level should be held for at least 4 f_{hw_clk} cycles (**T3CON.T3BPS1** = 01) before it changes.

9.6.4.1.3 Timer Concatenation

Using T3OTL as a clock source for an auxiliary timer of Block 1 in Counter Mode concatenates the core Timer T3 with the respective auxiliary timer. Depending on which transition of T3OTL is selected to clock the auxiliary timer, this concatenation forms a 32-bit or a 33-bit timer/counter:

- **32-bit Timer/Counter**: If both a positive and a negative transition of T3OTL are used to clock the auxiliary Timer, this timer is clocked on every overflow/underflow of the core Timer T3. Thus, the two timers form a 32-bit timer.
- **33-bit Timer/Counter**: If either a positive or a negative transition of T3OTL is selected to clock the auxiliary Timer, this timer is clocked on every second overflow/underflow of the core Timer T3. This configuration forms a 33-bit timer (16-bit core Timer+T3OTL+16-bit auxiliary Timer).

The count directions is not required to be the same in the two concatenated timers. This offers a wide variety of different configurations. T3 can operate in Timer Mode, Gated Timer Mode or Counter Mode in this case.

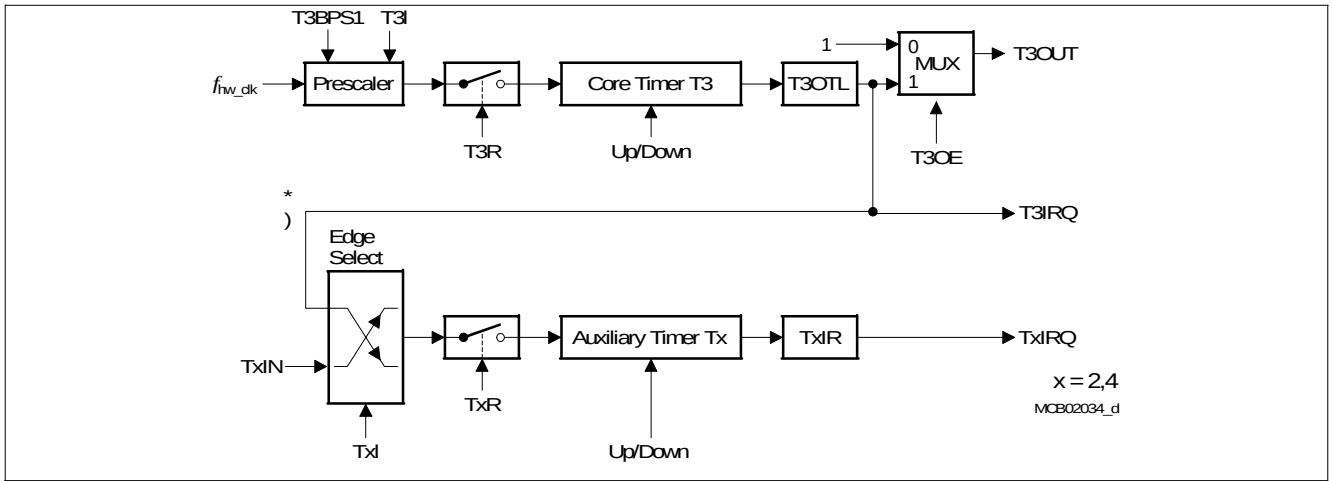


Figure 160 Concatenation of Core Timer T3 and an Auxiliary Timer

Note: Line '' only affected by over/underflows of T3, but NOT by software modifications of T3OTL.*

Auxiliary Timer in Reload Mode

Reload Mode for the auxiliary Timers T2 or T4 is selected by setting bit field **T2CON.T2I** or **T4CON.T4I** to 100_B. In Reload Mode the core Timer T3 is reloaded with the contents of an auxiliary timer register, triggered by one of two different signals. The trigger signal is selected the same way as the clock source for Counter Mode (see **Table 92**). That is, a transition of the auxiliary Timer's input or the output toggle latch T3OTL may trigger the reload.

Note: When programmed for Reload Mode, the respective auxiliary Timer T2 or T4 stops independent of its run flag T2R or T4R.

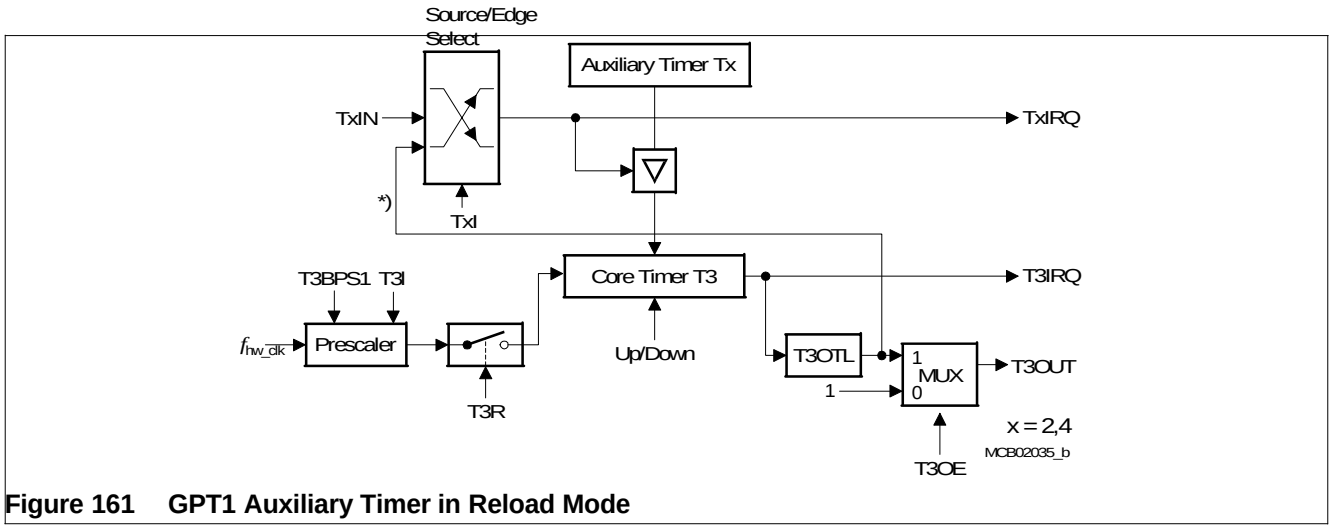


Figure 161 GPT1 Auxiliary Timer in Reload Mode

Note: Line '' only affected by over/underflows of T3, but NOT by software modifications of T3OTL.*

Upon a trigger signal, T3 is loaded with the contents of the respective timer register (T2 or T4) and T2IRQ or T4IRQ is set.

Note: When a T3OTL transition is selected for the trigger signal, the interrupt request flag T3IRQ will be set upon a trigger, indicating T3's overflow or underflow.
Modifications of T3OTL via software will NOT trigger the counter function of T2/T4.

The Reload Mode triggered by T3OTL can be used in a number of different configurations. Depending on the selected active transition, the following functions can be performed:

- If both a positive and a negative transition of T3OTL are selected to trigger a reload, the core Timer will be reloaded with the contents of the auxiliary Timer each time it overflows or underflows. This is the standard Reload Mode (reload on overflow/underflow).
- If either a positive or a negative transition of T3OTL is selected to trigger a reload, the core Timer will be reloaded with the contents of the auxiliary Timer on every second overflow or underflow.
- Using this "single-transition" mode for both auxiliary Timers allows to perform very flexible pulse width modulation (PWM). One of the auxiliary Timers is programmed to reload the core Timer on a positive transition of T3OTL, the other is programmed for a reload on a negative transition of T3OTL. With this combination the core Timer is alternately reloaded from the two auxiliary Timers.

The **Figure 162** shows an example for the generation of a PWM signal using the alternate reload mechanism. T2 defines the high time of the PWM signal (reloaded on positive transitions) and T4 defines the low time of the PWM signal (reloaded on negative transitions). The PWM signal can be output on line T3OUT if the control bit **T3CON.T3OE** is set. With this method, the high and low time of the PWM signal can be varied over a wide range.

Notes

1. *T3OTL is accessible via software and may be changed, if required, to modify the PWM signal. However, this will NOT trigger the reloading of T3.*
2. *An associated port pin linked to line T3OUT should be configured as output.*

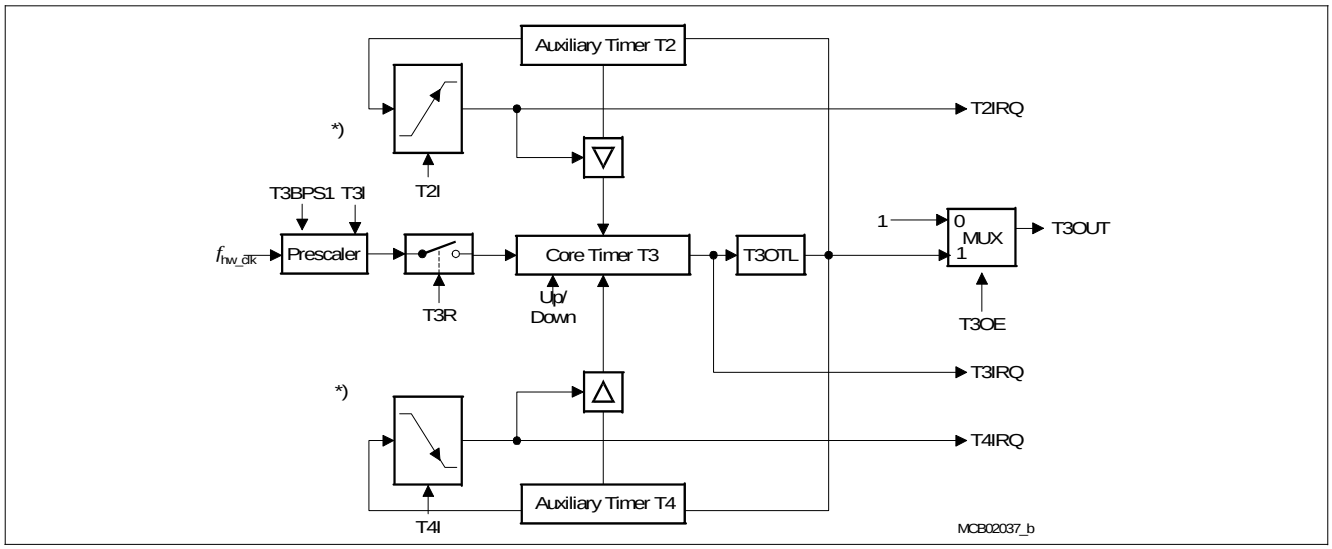


Figure 162 GPT1 Timer Reload Configuration for PWM Generation

Notes

1. *Lines '*' only affected by over/underflows of T3, but NOT by software modifications of T3OTL.*
2. *It should be avoided to select the same reload trigger event for both auxiliary Timers. In this case both reload registers would try to load the core Timer at the same time. If this combination is selected, T2 is disregarded and the contents of T4 is reloaded.*

Auxiliary Timer in Capture Mode

Capture Mode for the auxiliary Timers T2 or T4 is selected by setting bit field **T2CON.T2M** or **T4CON.T4M** to 101_B. In Capture Mode the contents of the core Timer are latched into an auxiliary timer register in response to a signal transition at the respective auxiliary Timer's external input line TxIN. The capture trigger signal can be a positive, a negative, or both a positive and a negative transition.

The two least significant bits of bit field **T2CON.T2I** or **T4CON.T4I** are used to select the active transition (see **Table 92**), while the most significant bit **T2CON.T2I(2)** or **T4CON.T4I(2)** is irrelevant for Capture Mode. It is recommended to keep this bit cleared.

*Note: When programmed for Capture Mode, the respective auxiliary Timer (T2 or T4) stops independent of its run flag **T2CON.T2R** or **T4CON.T4R**.*

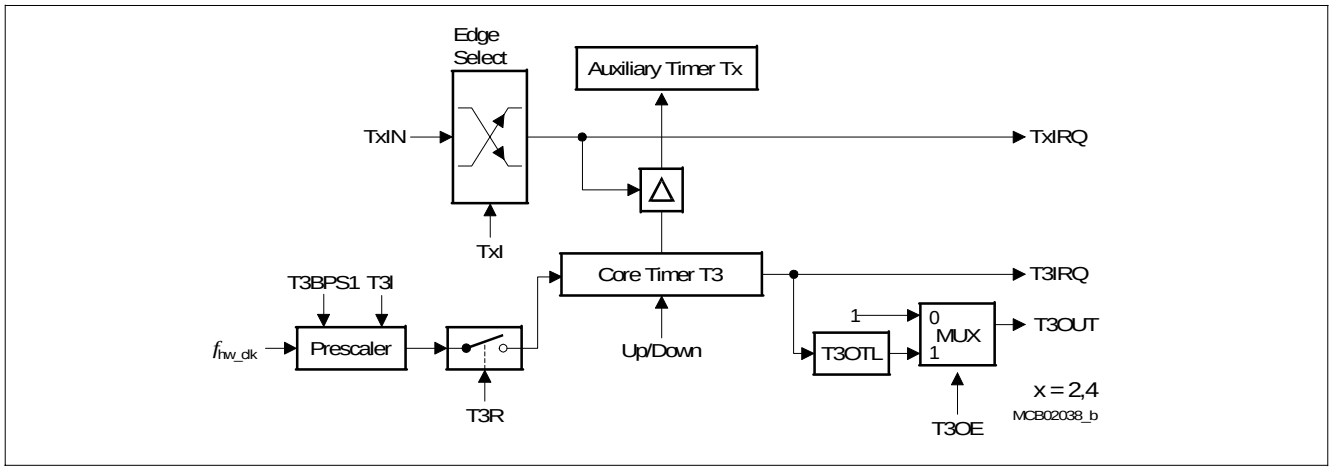


Figure 163 Auxiliary Timer of Timer Block 1 in Capture Mode

Upon a trigger (selected transition) at the corresponding input line TxIN, the contents of the core Timer are loaded into the auxiliary timer register and the associated interrupt request flag TxIRQ will be driven high.

*Note: Port pins associated with T2IN and T4IN must be configured to Input, and the level of the capture trigger signal should be held high or low for at least 4 f_{clk} (**T3CON.T3BPS1** = 01) cycles before it changes to ensure correct edge detection.*

Auxiliary in Incremental Interface Mode

When auxiliary Timers T2 and T4 are programmed to Incremental Interface Mode, their operation is the same as described for core Timer T3. The descriptions, figures, and tables apply accordingly with two exceptions:

- There is no TxOUT output line for T2 and T4.
- Overflow/underflow monitoring is not supported (no bit TxOTL).

Table 93 Timer x Input Parameter Selection for Incremental Interface Mode

T2CON.T2I T4CON.T4I	Triggering Edge for Counter Update
000	None. Counter Tx stops
001	Any transition (rising or falling edge) on TxIN
010	Any transition (rising or falling edge) on TxEUD
011	Any transition (rising or falling edge) on TxIN or TxEUD
1XX	***Reserved***. Do not use this combination!

9.6.4.2 Functional Description of Timer Block 2

Timer Block 2 includes the two Timers T5 (referred to as the auxiliary Timer) and T6 (referred to as the core Timer), and the 16-bit capture/reload register CAPREL. Each Timer of Block 2 is controlled by a separate control register **T5CON** or **T6CON**.

Each timer has an input line (TxIN) associated with it which serves as the gate control in Gated Timer Mode, or as the count input in Counter Mode. The count direction (up/down) may be programmed via software or may be dynamically altered by a signal at an external control input line. An overflow/underflow of core Timer T6 is indicated by bit T6OTL whose state may be output on related line T6OUT and on line T6OFL. The core Timer T6 may be reloaded with the contents of CAPREL.

The toggle bit also supports the concatenation of T6 with auxiliary Timer T5, while concatenation of T6 with other timers is provided through line T6OFL. Triggered by an external signal, the contents of T5 can be captured into register CAPREL and T5 may optionally be cleared. Both Timer (T6 and T5) can count up or down, and the current timer value can be read or modified by the MCU in the non-bitaddressable SFRs T5 and T6.

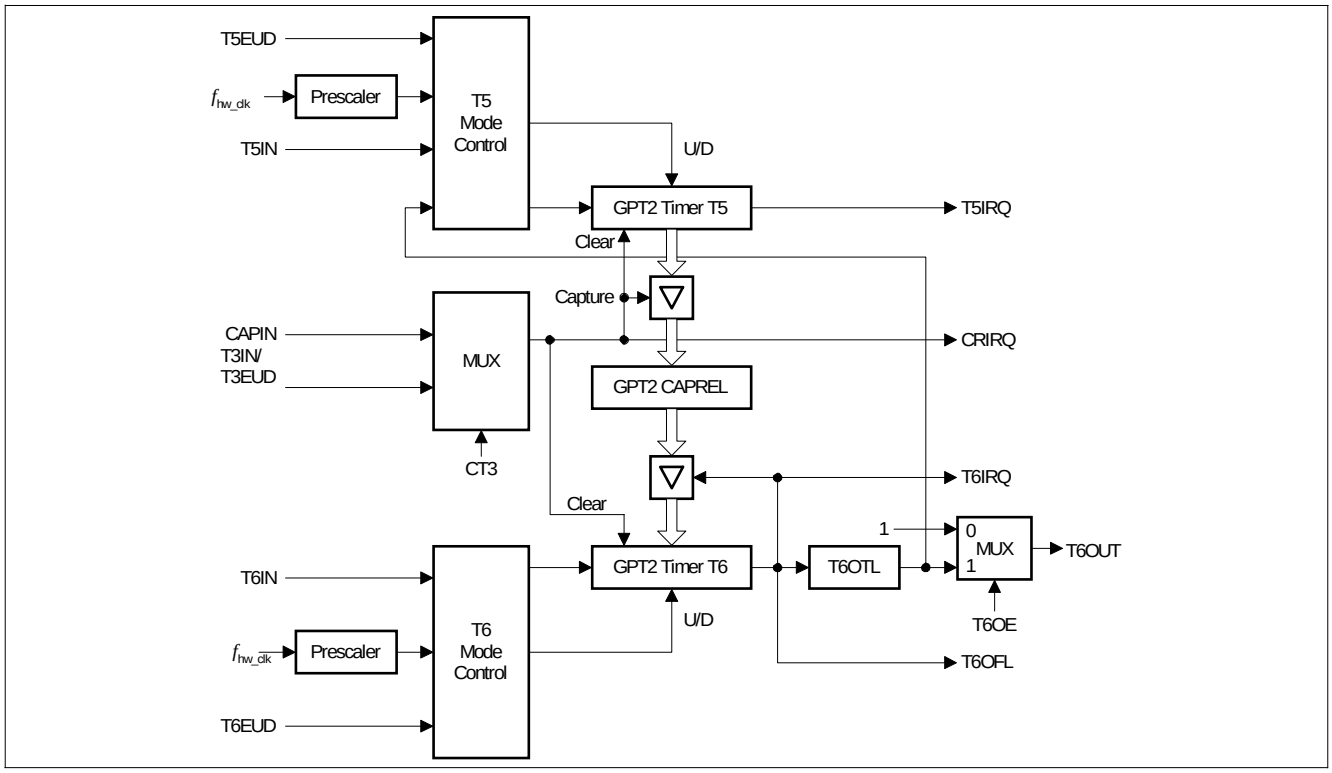


Figure 164 Structure of Timer Block 2

9.6.4.2.1 Core Timer T6

The operation of the core Timer T6 is controlled by its bitaddressable control register **T6CON**.

Timer 6 Run Bit

The timer can be started or stopped by software through bit **T6CON.T6R** (Timer T6 Run Bit). Setting bit **T6CON.T6R** will start the timer; clearing **T6CON.T6R** stops the timer.

In Gated Timer Mode, the timer will only run if **T6CON.T6R** is set and the gate is active (high or low, as programmed).

*Note: When bit **T5CON.T5RC** is set bit **T6CON.T6R** will also control (start and stop) auxiliary Timer T5.*

Count Direction Control

The count direction of the core Timer can be controlled either by software or by the external up/down input line T6EUD). These options are selected by bits **T6CON.T6UD** and **T6CON.T6UDE**. When the up/down control is done by software (bit **T6CON.T6UDE** is cleared), the count direction can be altered by setting or clearing bit **T6CON.T6UD**. When **T6CON.T6UDE** is set, line T6EUD is selected to be the controlling source of the count direction. However, bit **T6CON.T6UD** can still be used to reverse the actual count direction, as shown in **Table 94**. If **T6CON.T6UD** is cleared and line T6EUD shows a low level, the timer is counting up. With a high level at T6EUD the timer is counting down. If **T6CON.T6UD** is set, a high level at line T6EUD specifies counting up, and a low level specifies counting down. The count direction can be changed regardless of whether the timer is running or not.

Table 94 Core Timer T6 Count Direction Control (Tx = T5 or T6)

TxEUD	TxUDE	TxUD	Count Direction
X	0	0	Count Up
X	0	1	Count Down

Table 94 Core Timer T6 Count Direction Control (Tx = T5 or T6) (cont'd)

TxEUD	TxUDE	TxUD	Count Direction
0	1	0	Count Up
1	1	0	Count Down
0	1	1	Count Down
1	1	1	Count Up

Note: The direction control works the same for core Timer T6 and for auxiliary Timer T5.

Timer 6 Overflow/Underflow Monitoring

An overflow or underflow of Timer T6 will toggle **T6CON.T6OTL**. T6OTL can also be set or reset by software. Bit **T6CON.T6OE** enables the state of T6OTL to be monitored via the external output line T6OUT. An associated port pin must be configured as output.

Additionally, T6OTL can be used in conjunction with the timer over/underflow as an input for the counter function of auxiliary Timer T5. For this purpose, the state of T6OTL does not have to be available at line T6OUT, because an internal connection is provided for this option.

An overflow or underflow of Timer T6 can also be used to clock other timers. For this purpose, there is the special output line T6OFL.

Timer 6 in Timer Mode

Timer Mode for the core Timer T6 is selected by setting bit field **T6CON.T6M** to 000_B. In this mode, T6 is clocked with the module clock divided by a programmable prescaler, as selected by bit field T6I. The input frequency f_{T6} for Timer T6 and its resolution r_{T6} are scaled linearly with lower clock frequencies f_{hw_clk}, as can be seen from the following formula:

$$f_{T6} = \frac{f_{hw\_clk}}{BPS2 * 2^{<T6I>}} \qquad r_{T6}\,[ms] = \frac{BPS2 * 2^{<T6I>}}{f_{hw\_clk}\,[MHz]} \tag{66}$$

*Note: <BPS2> represents the prescaler value of the prescaler part controlled by bit field **T6CON.BPS2**.*

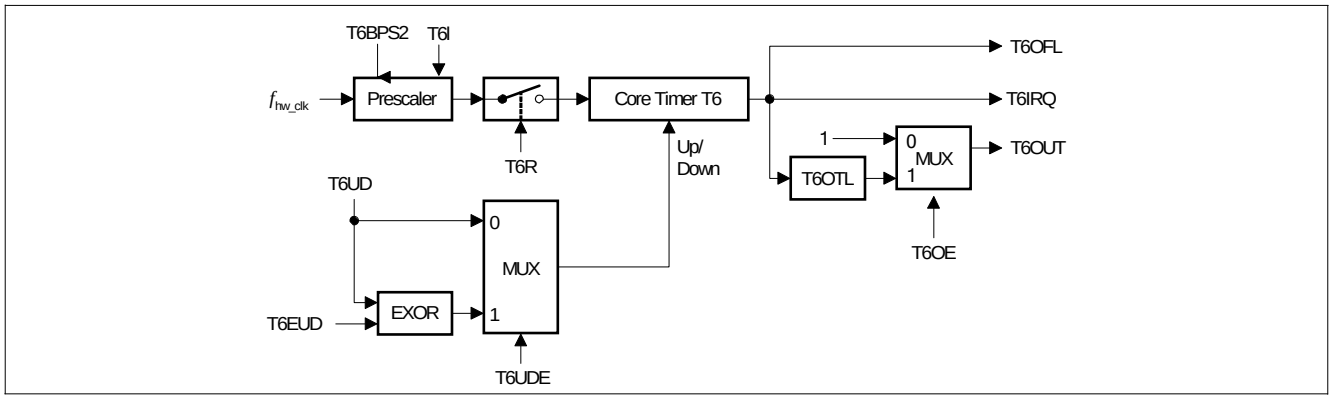


Figure 165 Block Diagram of Core Timer T6 in Timer Mode

Table 95 Timer 6 Input Parameter Selection: Timer Mode and Gated Timer Mode

T6CON.T6I	Prescaler for f_{hw_clk} (T6CON.BPS2 = 00)	Prescaler for f_{hw_clk} (T6CON.BPS2 = 01)	Prescaler for f_{hw_clk} (T6CON.BPS2 = 10)	Prescaler for f_{hw_clk} (T6CON.BPS2 = 11)
000	4	2	16	8
001	8	4	32	16
010	16	8	64	32

Table 95 Timer 6 Input Parameter Selection: Timer Mode and Gated Timer Mode (cont'd)

T6CON.T6I	Prescaler for f_{hw_clk} (T6CON.BPS2 = 00)	Prescaler for f_{hw_clk} (T6CON.BPS2 = 01)	Prescaler for f_{hw_clk} (T6CON.BPS2 = 10)	Prescaler for f_{hw_clk} (T6CON.BPS2 = 11)
011	32	16		128 64
100	64	32		256 128
101	128	64	512	256
110	256	128	1024	512
111	512	256	2048	1024

Timer 6 in Gated Timer Mode

Gated Timer Mode for the core Timer T6 is selected by setting bit field **T6CON.T6M** to 010_B or 011_B. Bit **T6CON.T6M(0)** (**T6CON(3)**) selects the active level of the gate input. In Gated Timer Mode the same options for the input frequency as for the Timer Mode are available. However, the input clock to the timer in this mode is gated by the external input line T6IN.

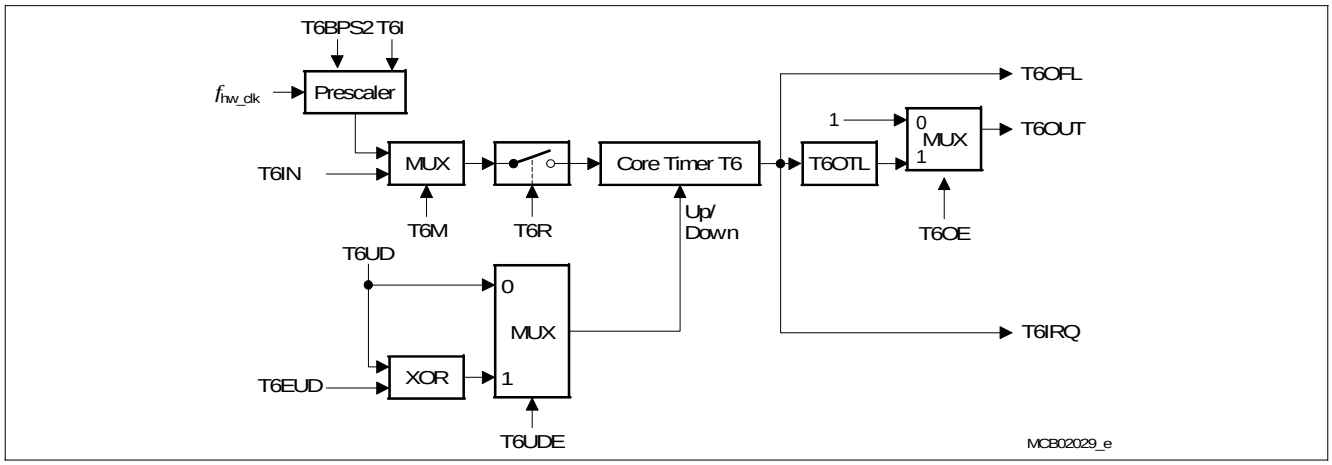


Figure 166 Block Diagram of Core Timer T6 in Gated Timer Mode

If **T6CON.T6M(0)** is cleared, the timer is enabled when T6IN shows a low level. A high level at this line stops the timer. If **T6CON.T6M(0)** is set, line T6IN must have a high level to enable the timer. Additionally, the timer can be turned on or off by software using bit T6R. The timer will only run, if **T6CON.T6R** is set and the gate is active. It will stop, if either **T6CON.T6R** is cleared or the gate is inactive.

Note: A transition of the gate signal at line T6IN does not cause an interrupt request.

Timer 6 in Counter Mode

Counter Mode for the core Timer T6 is selected by setting bit field **T6CON.T6M** to 001_B. In Counter Mode, Timer T6 is clocked by a transition at the external input line T6IN. The event causing an increment or decrement of the timer can be a positive, a negative, or both a positive and a negative transition at this line. bit field **T6CON.T6I** selects the triggering transition (see **Table 96**).

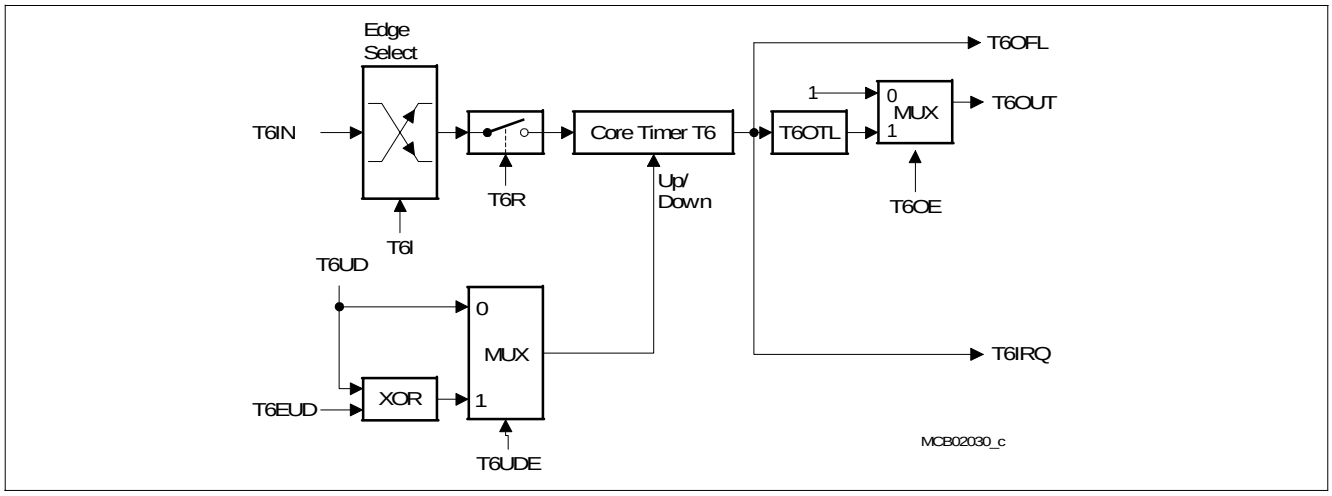


Figure 167 Block Diagram of Core Timer T6 in Counter Mode

Table 96 Core Timer T6 (Counter Mode) Input Edge Selection

T6CON.T6I	Triggering Edge for Counter Increment/Decrement
0 0 0	None. Counter T6 is disabled
0 0 1	Positive transition (rising edge) on T6IN
0 1 0	Negative transition (falling edge) on T6IN
0 1 1	Any transition (rising or falling edge) on T6IN
1 X X	Reserved. Do not use this combination

The maximum input frequency allowed in Counter Mode is $f_{hw\_clk}/4$ (**T6CON.**BPS2 = 01). To ensure that a transition of the count input signal which is applied to T6IN is correctly recognized, its level should be held high or low for at least 2 f_{hw_clk} cycles (**T6CON.**BPS2 = 01) before it changes.

9.6.4.2.2 Auxiliary Timer T5

The auxiliary Timer T5 can be configured for Timer Mode, Gated Timer Mode, or Counter Mode with the same options for the timer frequencies and the count signal as core Timer T6. In addition to these 3 counting modes, the auxiliary Timer T5 can be concatenated with the core Timer T6.

The individual configuration for Timer T5 is determined by its bitaddressable control register **T5CON**. The functions present in both timers of Timer Block 2 are controlled in the same bit positions and in the same manner in each of the specific control registers.

Run control for auxiliary Timer T5 can be handled by the associated Run Control Bit **T5CON.T5R**. Alternatively, a remote control option (**T5CON.T5RC** is set) may be enabled to start and stop T5 via the run bit **T5CON.T5R** of core Timer T6.

Note: The auxiliary Timer has no T5OTL. Therefore, an output line for overflow/underflow monitoring is not provided.

Count Direction Control for Auxiliary Timer

The count direction of the auxiliary Timer can be controlled in the same way as for the core Timer T6. The description and the table apply accordingly.

Timer T5 in Timer Mode or Gated Timer Mode

When auxiliary Timer T5 is programmed to Timer Mode or Gated Timer Mode, its operation is the same as described for the core Timer T6. The descriptions, figures and tables apply accordingly with two exceptions:

- There is no TxOUT line for T5.
- Overflow/underflow monitoring is not supported (no bit T5OTL).

Table 97 Timer 5 Input Parameter Selection: Timer Mode and Gated Timer Mode

T5CON.T5I	Prescaler for f_{hw_clk} (T6CON.BPS2 = 00)	Prescaler for f_{hw_clk} (T6CON.BPS2 = 01)	Prescaler for f_{hw_clk} (T6CON.BPS2 = 10)	Prescaler for f_{hw_clk} (T6CON.BPS2 = 11)
000	4	2		16 8
001	8	4		32 16
010	16	8	64	32
011	32	16	128	64
100	64	32	256	128
101	128	64	512	256
110	256	128	1024	512
111	512	256	2048	1024

Timer T5 in Counter Mode

Counter Mode for auxiliary Timer T5 is selected by setting bit field **T5CON.T5M** to 001_B. In Counter Mode, Timer T5 can be clocked either by a transition at the external input line T5IN, or by a transition of Timer T6's output toggle latch T6OTL.

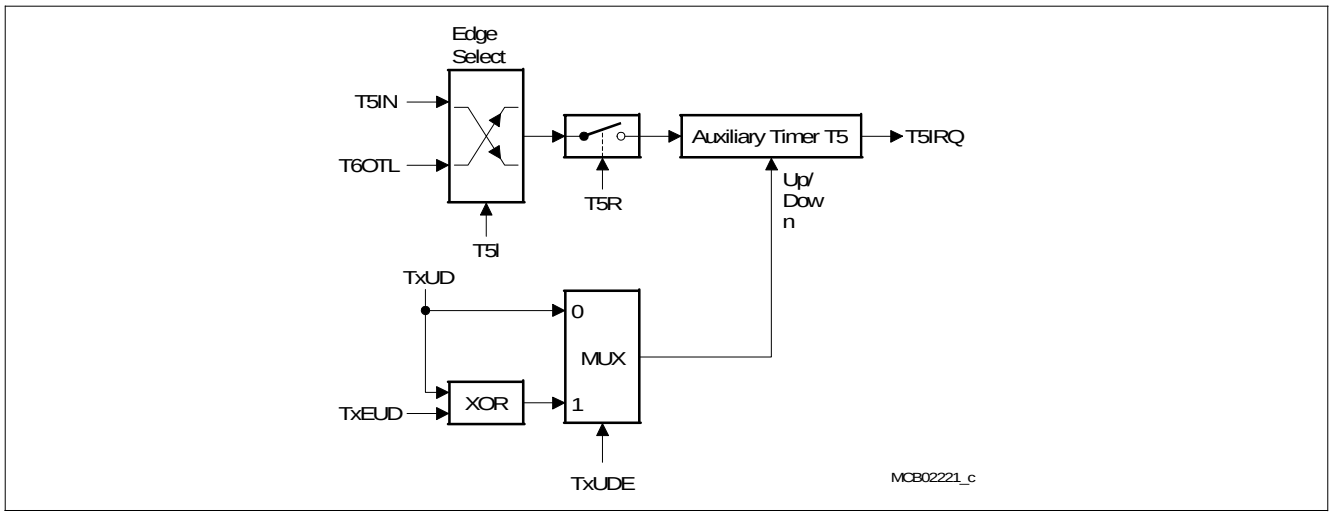


Figure 168 Block Diagram of Auxiliary Timer T5 in Counter Mode

The event causing an increment or decrement of the timer can be a positive, a negative, or both a positive and a negative transition at either the input line T5IN or at the toggle latch T6OTL.

Bit field **T5CON.T5I** selects the triggering transition (see **Table 98**).

Table 98 Auxiliary Timer (Counter Mode) Input Edge Selection

T5CON.T5I	Triggering Edge for Counter Increment/Decrement
X 0 0	None. Counter T5 is disabled
0 0 1	Positive transition (rising edge) on T5IN
0 1 0	Negative transition (falling edge) on T5IN
0 1 1	Any transition (rising or falling edge) on T5IN
1 0 1	Positive transition (rising edge) of T6OTL
1 1 0	Negative transition (falling edge) of T6OTL
1 1 1	Any transition (rising or falling edge) of T6OTL

*Note: Only state transitions of T6OTL caused by the overflow/underflow of T6 will trigger the counter function of T5. Modifications of T6OTL via software will **NOT** trigger the counter function of T5.*

The maximum input frequency which is allowed in Counter Mode is $f_{hw\_clk}/4$ (**T6CON.BPS2** = 01). To ensure that a transition of the count input signal which is applied to T5IN is correctly recognized, its level should be held high or low for at least 2 f_{hw_clk} cycles (**T6CON.BPS2** = 01) before it changes.

9.6.4.2.3 Timer Concatenation

Using the toggle bit **T6CON.T6OTL** as a clock source for the auxiliary Timer of Block 2 in Counter Mode concatenates core Timer T6 with auxiliary Timer T5. Depending on which transition of **T6CON.T6OTL** is selected to clock auxiliary Timer T5, this concatenation forms a 32-bit or a 33-bit timer/counter.

- 32-bit Timer/Counter: If both a positive and a negative transition of **T6CON.T6OTL** is used to clock the auxiliary Timer T5, this timer is clocked on every overflow/underflow of the core Timer T6. Thus, the two timers form a 32-bit timer.
- 33-bit Timer/Counter: If either a positive or a negative transition of **T6CON.T6OTL** is selected to clock the auxiliary Timer T5, this timer is clocked on every second overflow/underflow of the core Timer T6. This configuration forms a 33-bit timer (16-bit core Timer+**T6CON.T6OTL**+16-bit auxiliary Timer).The count directions of the two concatenated timers are not required to be the same. This offers a wide variety configurations. T6 can operate in Timer Mode, Gated Timer Mode or Counter Mode in this case.

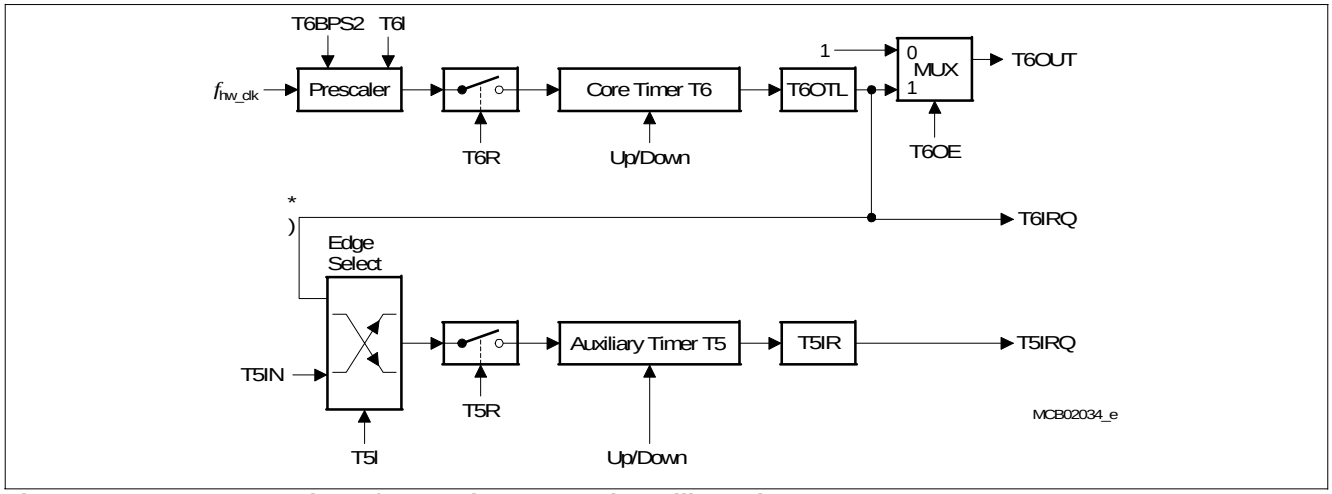


Figure 169 Concatenation of Core Timer T6 and Auxiliary Timer T5

Note: Line '' only affected by over/underflows of T6, but NOT by software modifications of **T6CON.T6OTL**.*

Capture/Reload Register CAPREL in Capture Mode

This 16-bit register can be used as a capture register for auxiliary Timer T5. This mode is selected by setting bit **T5CON.T5SC**. Bit **T5CON.CT3** selects the external input line (CAPIN) or the input lines (T3IN and/or T3EUD) of Timer T3 as the source for a capture trigger. Either a positive, a negative, or both a positive and a negative transition at line CAPIN can be selected to trigger the capture function, or transitions on input T3IN or input T3EUD or both inputs T3IN and T3EUD. The active edge is controlled by bit field CI in register **T5CON**.

The maximum input frequency for the capture trigger signal at CAPIN is $f_{hw\_clk}/2$ (**T6CON.BPS2** = 01). To ensure that a transition of the capture trigger signal is correctly recognized, its level should be held for at least 2 f_{hw_clk} cycles (**T6CON.BPS2** = 01) before it changes.

When the Timer T3 capture trigger is enabled (**T5CON.CT3** is set) register CAPREL captures the contents of T5 upon transitions of the selected input(s). These values can be used to measure input signals of T3. This is useful, for example, when T3 operates in Incremental Interface Mode, to derive dynamic information (speed acceleration) from the input signals.

When a selected transition at the external input line CAPIN is detected, the contents of auxiliary Timer T5 are latched into register CAPREL, and interrupt request line CRIRQ is driven at high level. With the same event, Timer T5 can be cleared to 0000_H. This option is controlled by bit **T5CON.T5CLR:**

- If **T5CLR** is cleared, the contents of Timer T5 is not affected by a capture.
- If **T5CLR** is set, Timer T5 is cleared after the current timer value has been latched into register CAPREL.

*Note: Bit **T5CON.T5SC** only controls whether a capture is performed or not. If **T5CON.T5SC** is cleared the input line CAPIN can still be used to clear Timer T5 or as an external interrupt input. This interrupt is controlled by the CAPREL interrupt control register CRIC.*

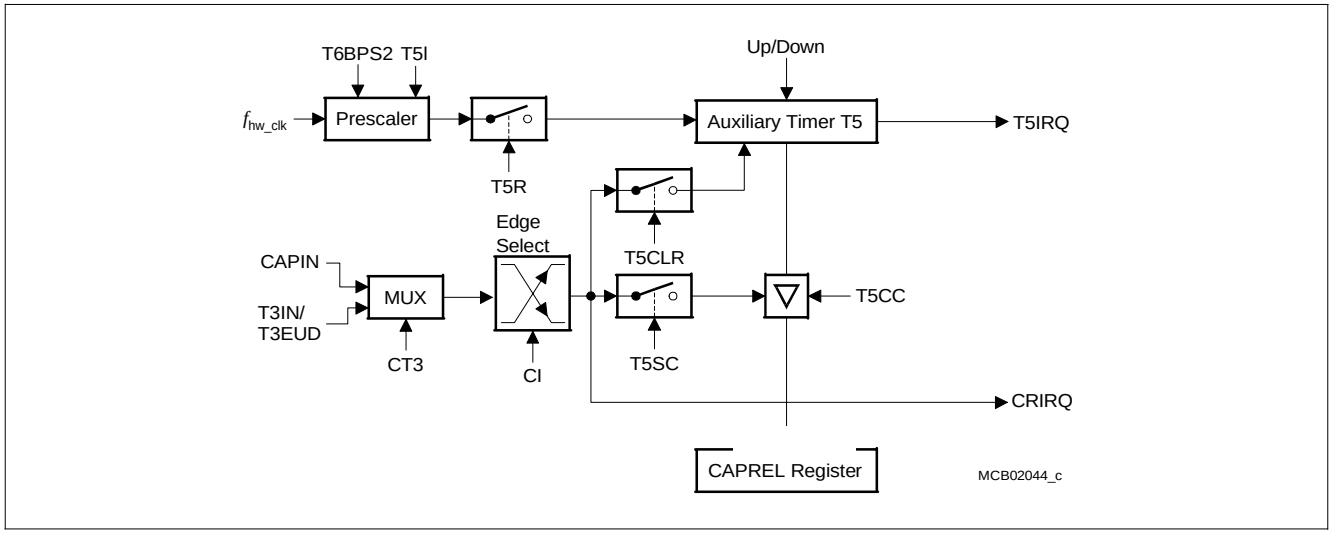


Figure 170 Timer Block 2 Register CAPREL in Capture Mode

Timer Block 2 Capture/Reload Register CAPREL in Reload Mode

This 16-bit register can be used as a reload register for core Timer T6. This mode is selected by setting bit **T6CON.T6SR**. The event causing a reload in this mode is an overflow or underflow of the core Timer T6. When Timer T6 overflows from $FFFF_H$ to 0000_H (when counting up) or underflows from 0000_H to $FFFF_H$ (when counting down), the value stored in register CAPREL is loaded into Timer T6. This will not drive the interrupt request line CRIRQ associated with the CAPREL register. However, interrupt request line T6IRQ will be driven at high level to indicate the overflow/underflow of T6.

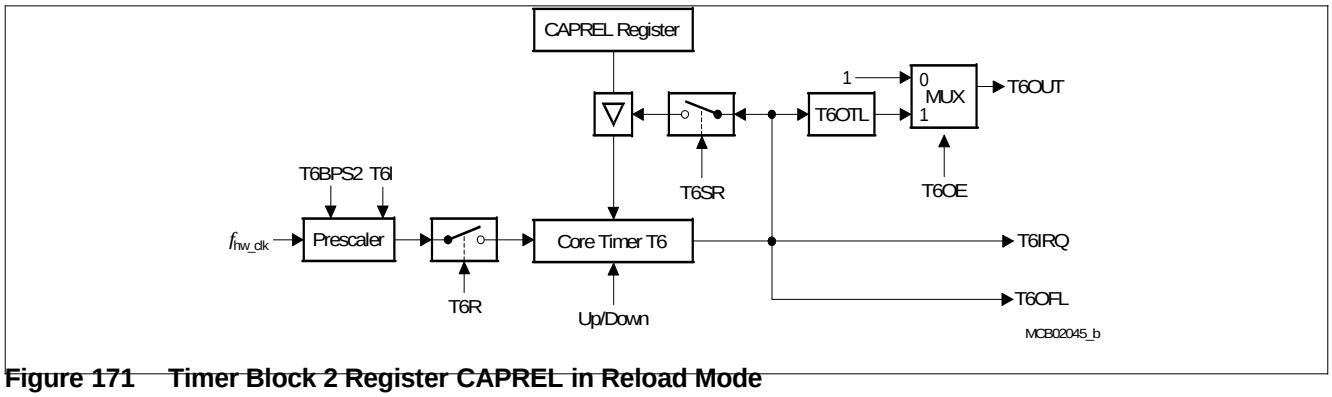


Figure 171 Timer Block 2 Register CAPREL in Reload Mode

Timer Block 2 Capture/Reload Register CAPREL in Capture-And-Reload Mode

Since the reload and capture functions of register CAPREL can be enabled individually by bits **T5CON.T5SC** and **T6CON.T6SR**, the two functions can be enabled simultaneously by setting both bits. This feature can be used to generate an output frequency that is a multiple of the input frequency.

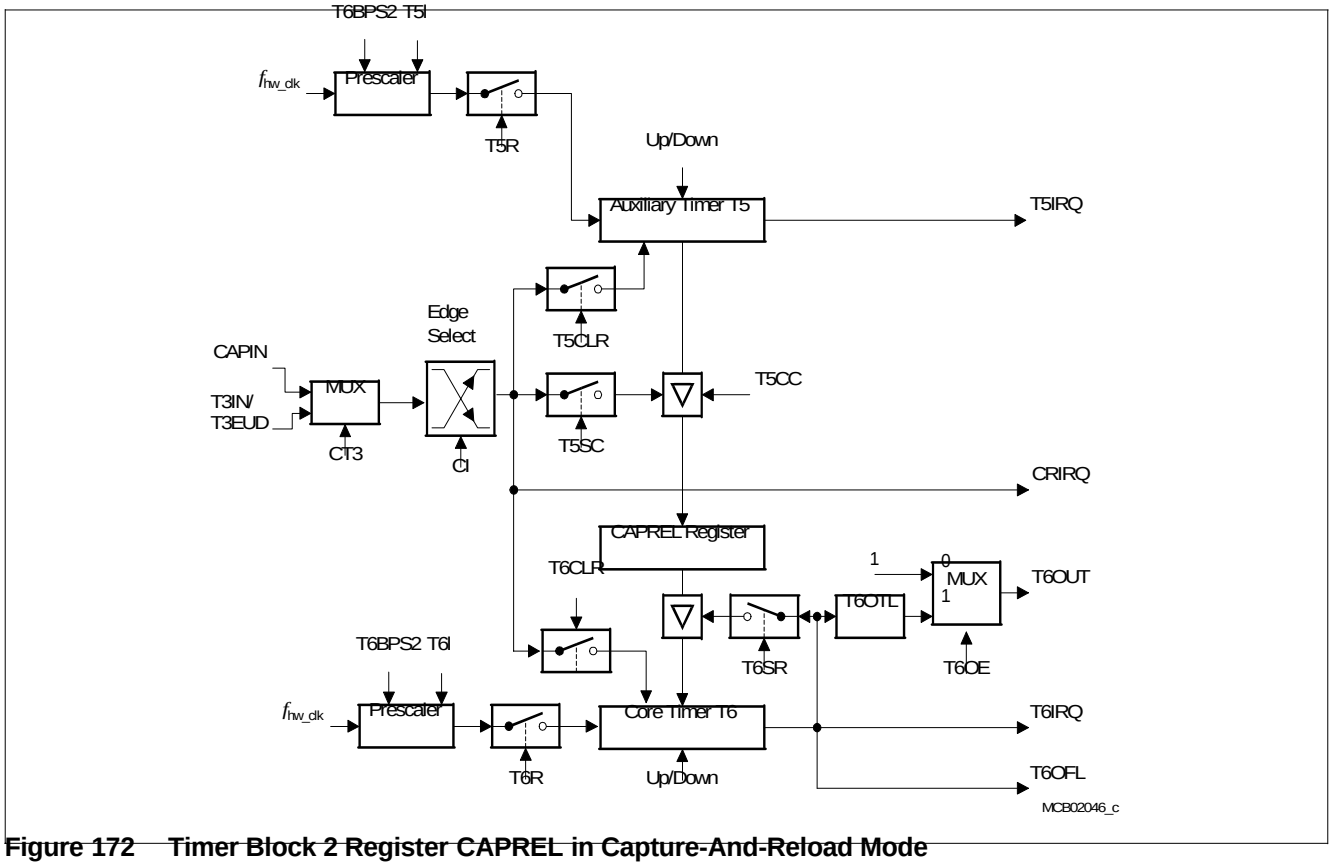


Figure 172 Timer Block 2 Register CAPREL in Capture-And-Reload Mode

This combined mode can be used to detect consecutive external events which may occur aperiodically, but where require a finer resolution (more 'ticks' within the time between two external events).

For this purpose, the time between the external events is measured using Timer T5 and the CAPREL register. Timer T5 runs in Timer Mode counting up with a frequency of $f_{hw\_clk}/32$, for example. The external events are applied to line CAPIN. When an external event occurs, the Timer T5 contents are latched into register CAPREL, and Timer T5 is cleared (**T5CON.T5CLR** cleared). Thus, register CAPREL always contains the correct time between two events, measured in Timer T5 increments. Timer T6, which runs in Timer Mode counting down with a frequency of $f_{hw\_clk}/4$, for example, uses the value in register CAPREL to perform a reload on underflow. This means, the value in register CAPREL represents the time between two underflows of Timer T6, now measured in Timer T6 increments. Since Timer T6 runs eight times faster than Timer T5, it will underflow eight times within the time between two external events. Thus, the underflow signal of Timer T6 generates eight 'ticks'. Upon each underflow, the interrupt request line T6IRQ will be driven at high level and bit **T6CON.T6OTL** will be toggled. The state of **T6CON.T6OTL** may be output on line T6OUT. This signal has eight times more transitions than the signal which is applied to line CAPIN.

A certain deviation of the output frequency is generated by the fact that Timer T5 will count actual time units (e.g. T5 running at 1 MHz will capture the value $64_H/100_D$ for a 10 KHz input signal) while **T6CON.T6OTL** will only toggle upon an underflow of T6 (that is, the transition from 0000_H to $FFFF_H$). In the above mentioned example T6 would count down from 64_H so the underflow would occur after 101 T6 timing ticks. The actual output frequency then is 79.2 KHz instead of the expected 80 KHz.
This can be solved by activating the Capture Correction (T5CC is set). If capture correction is active the content of T5 is decremented by 1 before being captured. The described deviation is eliminated (in the example, T5 would now capture $63_H/99_D$ and the output frequency would be 80 KHz).

Note: The underflow signal of Timer T6 can furthermore be used to clock one ore more of the timers of the CAPCOM units. This makes it possible to set compare events based on a finer resolution than that of the external events. This connection is accomplished via signal T6OFL.

9.6.4.3 GPT12 Kernel Registers

All available kernel registers are summarized in the overview table below.

Table 99 GPT12 Register Summary

Name	Clock	Access Condition	Description
GPTID	cfg_clk[1)]	None	Identification Register
T2CON	hw_clk[1)]	bitaddressable	Timer 2 Control Register
T3CON	hw_clk[1)]	bitaddressable	Timer 3 Control Register
T4CON	hw_clk[1)]	bitaddressable	Timer 4 Control Register
T5CON	cfg_clk[1)]	bitaddressable	Timer 5 Control Register
T6CON	hw_clk[1)]	bitaddressable	Timer 6 Control Register
CAPREL	hw_clk[1)]	None	Capture/Reload Register
T2	hw_clk[1)]	None	Timer 2 Register
T3	hw_clk[1)]	None	Timer 3 Register
T4	hw_clk[1)]	None	Timer 4 Register
T5	hw_clk[1)]	None	Timer 5 Register
T6	hw_clk[1)]	None	Timer 6 Register
GPTPISEL	cfg_clk[1)]	None	Port Input Select Register

1) Refer to Clock Domain in **System Integration (on Page 392)**

9.6.4.3.1 GPT Identification Register

GPTID

GPT Identification Register **Reset value: 5803_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Module_ID								Revision_Number							

Field	Bits	Type	Description
Revision_Number	0:7	r	**GPT Revision Number** These hard-wired bits are used for module revision numbering.
Module_ID	8:15	r	**GPT Identification Number** These hard-wired bits are used for module identification numbering.

9.6.4.3.2 GPT12 Port Input Selection Register

Normally the **GPTPISEL** register switches between different port input sources the GPT12 unit via an input multiplexer.

But in the E-GOLDvoice both values (0 or 1) for the **GPTPISEL.TxINIS** and **GPTPISEL.TxEUDIS** bits are tied to the same pin, therefore, the values chosen do not matter.

However, the **GPTPISEL.CAPINIS** bit is different. Because 0 selects a chip input pin, this register should be left at its reset value.

GPTPISEL

Input Select Register **Reset Value 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					T6 EUDIS	T5 EUDIS	CAP INIS	T6 INIS	T5 INIS	T4 EUDIS	T3 UDIS	T2 EUDIS	T4 INIS	T3 INIS	T2 INIS

Field	Bits	Type	Description
TxINIS (x = 2,3,4,5,6)	0, 1, 2, 6, 7	rw	**Select Source for Timer Input** 0 Pin TxIN selected 1 Pin TxIN selected
TxEUDIS (x = 2,3,4,5,6)	3, 4, 5, 9,	rw	**Select Source for Timer External Up/Down** 0 Pin TxEUD selected 1 Pin TxEUD selected
CAPINIS	10 8	rw	**Select Source for Timer Capture Input** 0 Pin CAPIN selected 1 No input, tied to logical 0
RESERVED	15:11	r	Reserved, these bits must be left at their reset values.

9.6.4.4 Function Control Registers

TxCON

9.6.4.4.1 Timer 3 Control Register

The operating mode of the core Timer T3 is configured and controlled via its bitaddressable control register **T3CON**.

T3CON

Timer 3 Control Register **Reset Value 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T3 RDIR	T3 CHDIR	T3 EDGE	T3 BPS1		T3 OTL	T3 OE	T3 UDE	T3 UD	T3R	T3M			T3I		

Field	Bits	Type	Description
T3I	2:0	rw	**Timer 3 Input Parameter Selection** • Timer Mode, see **Table 100 on page 417** for encoding • Gated Timer Mode, see **Table 100 on page 417** for encoding • Counter Mode, see Table 101 on page 417 for encoding • Incremental Interface Mode, see Table 102 on page 417 for encoding
T3M	5:3	rw	**Timer 3 Mode Control** 000 Timer Mode 001 Counter Mode 010 Gated Timer Mode with Gate active low 011 Gated Timer Mode with Gate active high 100 Reserved. Do not use this combination. 101 Reserved. Do not use this combination. 110 Incremental Interface Mode (Rotation detection) 111 Incremental Interface Mode (Edge detection)
T3R	6	rw	**Timer 3 Run Bit** 0 Timer/counter 3 stops 1 Timer/counter 3 runs
T3UD	7	rw	**Timer 3 Up/Down Control** (when T3UDE is cleared) 0 Counting up 1 Counting down
T3UDE	8	rw	**Timer 3 External Up/Down Enable** 0 Counting direction is internally controlled by SW 1 Counting direction is externally controlled by line T3EUD
T3OE	9	rw	**Overflow/Underflow Output Enable** 0 T3 overflow/underflow can not be externally monitored 1 T3 overflow/underflow may be externally monitored via T3OUT
T3OTL	10	rwh	**Timer 3 Output Toggle Latch** Toggles on each overflow/underflow of T3. Can be set or reset by software.

Field	Bits	Type	Description
BPS1	12:11	rw	**Timer Block Prescaler 1** The maximum input frequency 00 For Timer 2/3/4 is f_{hw_clk} /8 01 For Timer 2/3/4 is f_{hw_clk} /4 10 For Timer 2/3/4 is f_{hw_clk} /32 11 For Timer 2/3/4 is f_{hw_clk} /16
T3EDGE	13	rwh	**Timer 3 Edge Detection** The bit is set on each successful edge detection. The bit has to be reset by SW. 0 No count edge was detected 1 A count edge was detected
T3CHDIR	14	rwh	**Timer 3 Count Direction Change** The bit is set on a change of the count direction of timer 3. The bit has to be reset by SW. 0 No change in count direction was detected 1 A change in count direction was detected
T3RDIR	15	rh	**Timer 3 Rotation Direction** 0 Timer 3 counts up 1 Timer 3 counts down

Table 100 Timer 3 Input Parameter Selection for Timer Mode and Gated Timer Mode

T3I	Prescaler for f_{hw_clk} (T3CON.BPS1 = 00)	Prescaler for f_{hw_clk} (T3CON.BPS1 = 01)	Prescaler for f_{hw_clk} (T3CON.BPS1 = 10)	Prescaler for f_{hw_clk} (T3CON.BPS1 = 11)
000	8	4		32 16
001	16	8	64	32
010	32	16	128	64
011	64	32	256	128
100	128	64	512	256
101	256	128	1024	512
110	512	256	2048	1024
111	1024	512	4096	2048

Table 101 Timer 3 Input Parameter Selection for Counter Mode

T3CON.T3I	Triggering Edge for Counter Update
000	None. Counter T3 is disabled
001	Positive transition (raising edge) on T3IN
010	Negative transition (falling edge) on T3IN
011	Any transition (raising or falling edge) on T3IN
1XX	Reserved. Do not use this combination!

Table 102 Timer 3 Input Parameter Selection for Incremental Interface Mode

T3CON.T3I	Triggering Edge for Counter Update
000	None. Counter T3 stops
001	Any transition (raising or falling edge) on T3IN
010	Any transition (raising or falling edge) on T3EUD

Table 102 Timer 3 Input Parameter Selection for Incremental Interface Mode (cont'd)

T3CON.T3I	Triggering Edge for Counter Update
011	Any transition (raising or falling edge) on T3IN or T3EUD
1XX	***Reserved***. Do not use this combination!

9.6.4.4.2 Timers 2 & 4 Control Register

T2CON
T4CON

Timer 2/4 Control Register **Reset Value 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxRDIR	**TxCHDIR**	**TxEDGE**	**TxIRDIS**	**RESERVED**		**TxRC**	**TxUDE**	**TxUD**	**TxR**	**TxM**			**TxI**		

Field	Bits	Type	Description
TxI	2:0	rw	**Timer x Input Parameter Selection** • Timer Mode, refer to Table 103 on page 419 for encoding • Gated Timer Mode, refer to Table 103 on page 419 for encoding • Counter Mode, refer to Table 104 on page 419 for encoding • Incremental Interface Mode, refer to Table 105 on page 419 for encoding
TxM	5:3	rw	**Timer x Mode Control (Basic Operating Mode)** 000 Timer Mode 001 Counter Mode 010 Gated Timer Mode with Gate active low 011 Gated Timer Mode with Gate active high 100 Reload Mode 101 Capture Mode 110 Incremental Interface Mode (Rotation detection) 111 Incremental Interface Mode (Edge detection)
TxR	6	rw	**Timer x Run Bit** 0 Timer/counter x stops 1 Timer/counter x runs
TxUD	7	rw	**Timer x Up/Down Control** (when TxUDE is cleared) 0 Counting up 1 Counting down
TxUDE	8	rw	**Timer x External Up/Down Enable** 0 Counting direction is internally controlled by SW 1 Counting direction is externally controlled by line TxEUD
TxRC	9	rw	**Timer x Remote Control** 0 Timer/counter x is controlled by its own run bit TxR 1 Timer/counter x is controlled by the run bit of core Timer 3
TxIRDIS	12	rw	**Timer x Interrupt Disable** 0 Interrupt generation for TxCHDIR and TxEDGE interrupts in Incremental Interface Mode is enabled 1 Interrupt generation for TxCHDIR and TxEDGE interrupts in Incremental Interface Mode is disabled

Field	Bits	Type	Description
TxEDGE	13	rwh	**Timer x Edge Detection** The bit is set on each successful edge detection. The bit has to be reset by SW. 0 No count edge was detected 1 A count edge was detected
TxCHDIR	14	rwh	**Timer x Count Direction Change** The bit is set on a change of the count direction of timer x. The bit has to be reset by SW. 0 No change in count direction was detected 1 A change in count direction was detected
TxRDIR	15	rh	**Timer x Rotation Direction** 0 Timer x counts up 1 Timer x counts down
RESERVED	11:10	r	Reserved, these bits must be left at their reset values.

Table 103 Timer 2,4 Input Parameter Selection for Timer Mode and Gated Timer Mode

T2CON.T2I or T4CON.T4I	Prescaler for f_{hw_clk} (T3CON.BPS1 = 00)	Prescaler for f_{hw_clk} (T3CON.BPS1 = 01)	Prescaler for f_{hw_clk} (T3CON.BPS1 = 10)	Prescaler for f_{hw_clk} (T3CON.BPS1 = 11)
000	8	4	32	16
001	16	8	64	32
010	32	16	128	64
011	64	32	256	128
100	128	64	512	256
101	256	128	1024	512
110	512	256	2048	1024
111	1024	512	4096	2048

Table 104 Timer 2,4 Input Parameter Selection for Counter Mode

T[2,4]I	Triggering Edge for Counter Update
X00	None. Counter Tx is disabled
001	Positive transition (raising edge) on TxIN
010	Negative transition (falling edge) on TxIN
011	Any transition (raising or falling edge) on TxIN
101	Positive transition (rising edge) of T3OTL
110	Negative transition (falling edge) of T3OTL
111	Any transition (rising or falling edge) of T3OTL

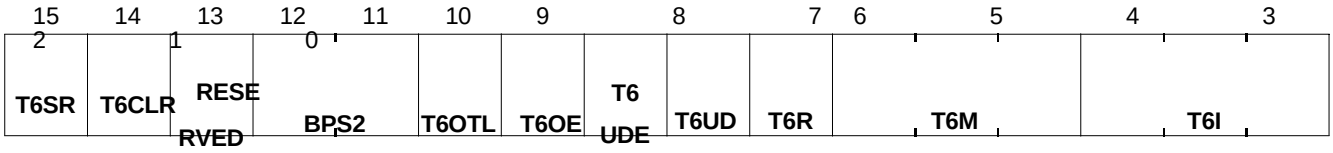
Table 105 Timer 2,4 Input Parameter Selection for Incremental Interface Mode

T[2,4]I	Triggering Edge for Counter Update
000	None. Counter Tx stops
001	Any transition (raising or falling edge) on TxIN
010	Any transition (raising or falling edge) on TxEUD
011	Any transition (raising or falling edge) on TxIN or TxEUD
1XX	Reserved. Do not use this combination!

9.6.4.4.3 Timer 6 Control Register

T6CON

Timer 6 Control Register **Reset Value 0000_H**



Field	Bits	Type	Description
T6I	2:0	rw	**Timer 6 Input Parameter Selection** • Timer Mode, see Table 106 on page 421 for encoding • Gated Timer Mode, see Table 106 on page 421 for encoding • Counter Mode, see Table 107 on page 421 for encoding
T6M	5:3	rw	**Timer 6 Mode Control (Basic Operating Mode)** 000 Timer Mode 001 Counter Mode 010 Gated Timer Mode with Gate active low 011 Gated Timer Mode with Gate active high 1XX Reserved. Do not use this combination!
T6R	6	rw	**Timer 6 Run Bit** 0 Timer/counter 6 stops 1 Timer/counter 6 runs
T6UD	7	rw	**Timer 6 Up/Down Control** (when T6UDE is cleared) 0 Counting up 1 Counting down
T6UDE	8	rw	**Timer 6 External Up/Down Enable** 0 Counting direction is internally controlled by SW 1 Counting direction is externally controlled by line TxEUD
T6OE	9	rw	**Overflow/Underflow Output Enable** 0 T6 overflow/underflow can not be externally monitored 1 T6 overflow/underflow may be externally monitored via T6OUT
T6OTL	10	rwh	**Timer 6 Output Toggle Latch** Toggles on each overflow/underflow of T6. Can be set or reset by software.
BPS2	12:11	rw	**Timer Block Prescaler 2** The maximum input frequency 00 For Timer 5/6 is f_{hw_clk} / 4 01 For Timer 5/6 is f_{hw_clk} / 2 10 For Timer 5/6 is f_{hw_clk} / 16 11 For Timer 5/6 is f_{hw_clk} / 8
T6CLR	14	rw	**Timer 6 Clear Bit** 0 Timer 6 is not cleared on a capture event 1 Timer 6 is cleared on a capture event
T6SR	15	rw	**Timer 6 Reload Mode Enable** 0 Reload from register CAPREL disabled 1 Reload from register CAPREL enabled
RESERVED	13	r	Reserved, these bits must be left at their reset values.

Table 106 Timer 6 Input Parameter Selection for Timer Mode and Gated Timer Mode

T6CON.T6I	Prescaler for f_{hw_clk} (T6CON.BPS2 = 00)	Prescaler for f_{hw_clk} (T6CON.BPS2 = 01)	Prescaler for f_{hw_clk} (T6CON.BPS2 = 10)	Prescaler for f_{hw_clk} (T6CON.BPS2 = 11)
000	4	2	16	8
001	8	4	32	16
010	16	8	64	32
011	32	16	128	64
100	64	32	256	128
101	128	64	512	256
110	256	128	1024	512
111	512	256	2048	1024

Table 107 Timer 6 Input Parameter Selection for Counter Mode

T6CON.T6I	Triggering Edge for Counter Update
000	None. Counter T6 is disabled
001	Positive transition (raising edge) on T6IN
010	Negative transition (falling edge) on T6IN
011	Any transition (raising or falling edge) on T6IN
1XX	Reserved. Do not use this combination!

T5CON

Timer 5 Control Register **Reset Value 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T5SR	T5CLR	CI		T5CC	CT3	T5RC	T5 UDE	T5UD	T5R	T5M		T5I			

Field	Bits	Type	Description
T5I	2:0	rw	**Timer 5 Input Parameter Selection** • Timer Mode, refer to Table 108 on page 422 for encoding • Gated Timer Mode, refer to Table 108 on page 422 for encoding • Counter Mode, refer to Table 109 on page 422 for encoding
T5M	5:3	rw	**Timer 5 Mode Control (Basic Operating Mode)** 000 Timer Mode 001 Counter Mode 010 Gated Timer Mode with Gate active low 011 Gated Timer Mode with Gate active high 1XX Reserved. Do not use this combination!
T5R	6	rw	**Timer 5 Run Bit** 0 Timer/counter 5 stops 1 Timer/counter 5 runs
T5UD	7	rw	**Timer 5 Up/Down Control** (when T5UDE is cleared) 0 Counting up 1 Counting down

Field	Bits	Type	Description
T5UDE	8	rw	**Timer 5 External Up/Down Enable** 0 Counting direction is internally controlled by SW 1 Counting direction is externally controlled by line TxEUD
T5RC	9	rw	**Timer 5 Remote Control** 0 Timer/counter x is controlled by its own run bit T5R 1 Timer/counter 5 is controlled by the run bit of core Timer 6
CT3	10	rw	**Timer 3 Capture Trigger Enable** 0 Capture trigger from input line CAPIN 1 Capture trigger from T3 input lines
T5CC	11	rw	**Timer 5 Capture Correction** 0 T5 is just captured without any correction 1 T5 is decremented by 1 before being captured
CI	13:12	rw	**Register CAPREL Capture Trigger Selection** (depending in bit CT3) 00 Capture disabled 01 Positive transition (rising edge) on CAPIN or any transition on T3IN 10 Negative transition (falling edge) on CAPIN or any transition on T3EUD 11 Any transition (rising or falling edge) on CAPIN or any transition on T3IN or T3EUD
T5CLR	14	rw	**Timer 5 Clear Bit** 0 Timer 5 is not cleared on a capture event 1 Timer 5 is cleared on a capture event
T5SC	15	rw	**Timer 5 Capture Mode Enable** 0 Capture into register CAPREL disabled 1 Capture into register CAPREL enabled

Table 108 Timer 5 Input Parameter Selection for Timer Mode and Gated Timer Mode

T5CON.T5I	Prescaler for f_{hw_clk} (T6CON.BPS2 = 00)	Prescaler for f_{hw_clk} (T6CON.BPS2 = 01)	Prescaler for f_{hw_clk} (T6CON.BPS2 = 10)	Prescaler for f_{hw_clk} (T6CON.BPS2 = 11)
000	4	2	16	8
001	8	4	32	16
010	16	8	64	32
011	32	16		128 64
100	64	32		256 128
101	128	64	512	256
110	256	128	1024	512
111	512	256	2048	1024

Table 109 Timer 5 Input Parameter Selection for Counter Mode

T5CON.T5I	Triggering Edge for Counter Update
X00	None. Counter T5 is disabled
001	Positive transition (rising edge) on T5IN
010	Negative transition (falling edge) on T5IN
011	Any transition (rising or falling edge) on T5IN
101	Positive transition (rising edge) of T3OTL

Table 109 Timer 5 Input Parameter Selection for Counter Mode (cont'd)

T5CON.T5I	Triggering Edge for Counter Update
110	Negative transition (falling edge) of T3OTL
111	Any transition (rising or falling edge) of T3OTL

9.6.4.5 Register Mapping

For information about the GPT12 PD-Bus Register Mapping refer to **Section 10.1 PD-Bus Register Addresses (on Page 481)**.

9.6.5 Interrupts

For a detailed description of the various interrupts refer to Section 9.6.4 Kernel Description (on Page 393). An overview is given in Table 110

Table 110 GPT1_2 Interrupt Sources

Interrupt	Signal	Description
Timer 2 Overflow	int_t2_o	Interrupt is requested on overflow of Timer 2 if counting up.
Timer 2 Underflow	int_t2_o	Interrupt is requested on underflow of Timer 2 if counting down.
Timer 3 Overflow	int_t3_o	Interrupt is requested on overflow of Timer 3 if counting up.
Timer 3 Underflow	int_t3_o	Interrupt is requested on underflow of Timer 3 if counting down.
Timer 4 Overflow	int_t4_o	Interrupt is requested on overflow of Timer 4 if counting up.
Timer 4 Underflow	int_t4_o	Interrupt is requested on underflow of Timer 4 if counting down.
Timer 5 Overflow	int_t5_o	Interrupt is requested on overflow of Timer 5 if counting up.
Timer 5 Underflow	int_t5_o	Interrupt is requested on underflow of Timer 5 if counting down.
Timer 6 Overflow	int_t6_o	Interrupt is requested on overflow of Timer 6 if counting up.
Timer 6 Underflow	int_t6_o	Interrupt is requested on underflow of Timer 6 if counting down.
Rotation Direction Change Timer 2	int_t2_o	Interrupt is requested on a change of the count direction in the Incremental Interface Mode (**T2CON.T2I** = 110).
Edge Detection Timer 2	int_t2_o	Interrupt is requested on a successful detected edge resulting in a timer count action (**T2CON.T2I** = 111).
Rotation Direction Change Timer 3	int_t3_o	Interrupt is requested on a change of the count direction in the Incremental Interface Mode (**T3CON.T3I** = 110).
Edge Detection Timer 3	int_t3_o	Interrupt is requested on a successful detected edge resulting in a timer count action (**T3CON.T3I** = 111).
Rotation Direction Change Timer 4	int_t4_o	Interrupt is requested on a change of the count direction in the Incremental Interface Mode (**T4CON.T4I** = 110).
Edge Detection Timer 4	int_t4_o	Interrupt is requested on a successful detected edge resulting in a timer count action (**T4CON.T4I** = 111).
Reload Action Timer 2	int_t2_o	Interrupt is requested on a trigger signal for reloading Timer 3 in Reload Mode (**T2CON.T2I** = 100).
Reload Action Timer 4	int_t4_o	Interrupt is requested on a trigger signal for reloading Timer 3 in Reload Mode (**T4CON.T4I** = 100).
Capture Action Timer 2	int_t2_o	Interrupt is requested on a trigger signal for a capture action to capture Timer 3 in Timer 2 Capture Mode (**T2CON.T2I** = 101).
Capture Action Timer 4	int_t4_o	Interrupt is requested on a trigger signal for a capture action to capture Timer 3 in Timer 4 Capture Mode (**T4CON.T4I** = 101).
Capture Action Timer Block 2	int_cr_o	Interrupt is requested on a trigger signal for a capture action of Timer 5 to register CAPREL in Capture Mode (**T5CON.T5SC** = 1).

9.7 Port Control Logic

System Integration

- Supply domain: VDD_LD1
- Chip internal interfaces:
  - Clock domain: Refer to **Section 7.2.1.3 Sub-System Clocks and Enables (on Page 67)** and see **Figure 18 Clock Enable (on Page 68)**.
  - Bus domain: PD-Bus
    Bus interface common with internal signal monitoring
  - Interrupt sources: no
  - Other interface:
- Chip external signals related to this block (refer to **Chapter 3 Pin Descriptions** for pin configuration options): all configurable pins.

9.7.1 Functional Overview

Most digital pads may be configured to be a general purpose IO port (GPIO), be connected to one of several inputs, or to one of several outputs from internal chip blocks.

The connection and the GPIO function are realized individually for each pad <pad> with the blocks PCL_<pad> and GPIO_<pad> (denoted PCL and GPIO in the rest of this section). Both blocks are accessible for the MCU via the bus.

The principal structure of PCL and GPIO is shown in **Figure 173**.

The main functional blocks of the PCL are the pad control register **PCL_<pad>**, the input demultiplexer IDMX, the output multiplexer OMX and the direction control block DIRCTL.

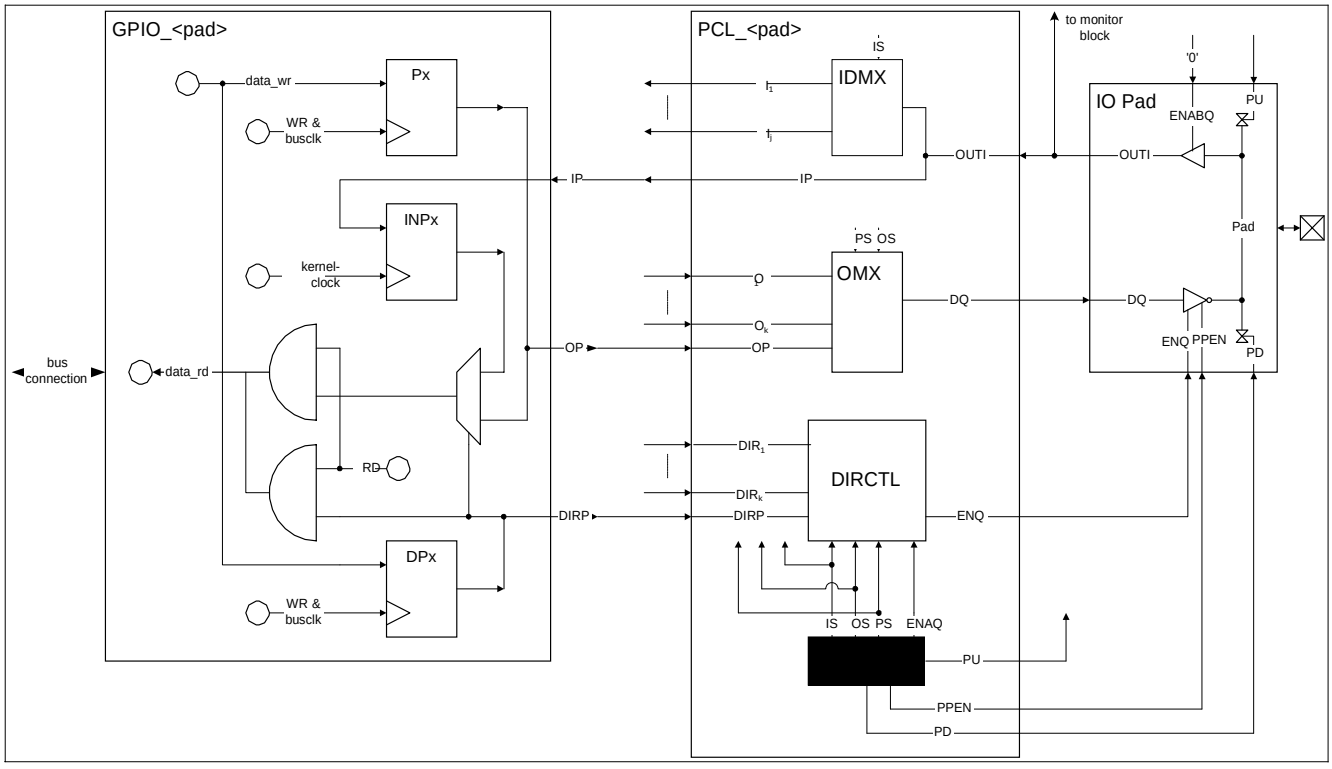


Figure 173 Architecture of Pad Control and Port Logic

Figure 174 shows a typical connection of several PCL blocks and chip internal blocks.

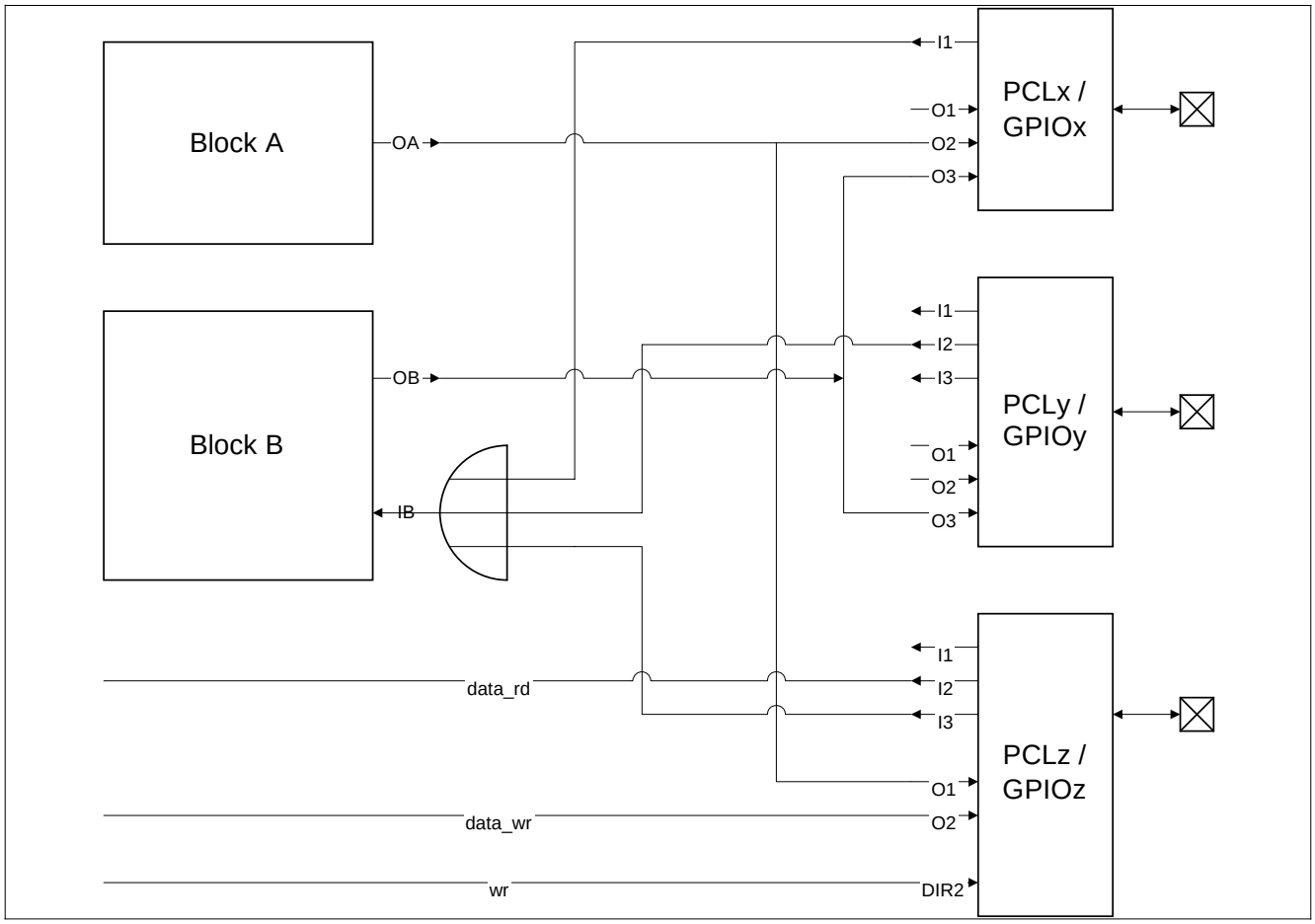


Figure 174 Example for Connecting PCL Blocks and Chip Internal Blocks

Note: Only the principal circuitry for pads with GPIO and additional functions is described here. The control circuitry for other pads like analog pads or digital pads with dedicated non-multiplexed functions is described in the related module sections (for example, RTC, Measurement, ...).

9.7.2 Register Description PCL_<pad>

The **PCL_<pad>** registers select the alternate functions of the GPIO chip pads. For example, **PCL_00** is the register that selects Alt0, Alt1, or Alt2 for the GPIO_00 pad (ball name KP0), refer to **Chapter 3 Pin Descriptions**. Certain pads do not use all the bits (they do not have alternate functions).

PCL_<pad>
Control and GPIO Register for pad <pad> **Reset value: refer to Table 111**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENAQ	PDPU		PPEN	RESERVED	DPx	Px	PS	RESERVED	OS		RESERVED		IS		

Field	Bits	Type	Description
IS	1:0	rw	**Input Selection** Connects the pad input to one of up to 3 chip internal inputs (see **Figure 173 (on page 425)**): 00 No input selected 01 Alt0 10 Alt1 11 Alt2 ***Attention: If an alternate is selected for a pad that does not have the alternate function, the pad behaves as if 00 (no input) were selected.***
OS	5:4	rw	**Output Selection** Connects one of up to 3 chip internal outputs to the pad output (see **Figure 173**): 00 No output selected 01 Alt0 02 Alt1 03 Alt2 ***Attention: If an alternate is selected for a pad that does not have the alternate function, the pad behaves as if 00 (no output) were selected.***
PS	8	rw	**Port Selection** 0: The internal signal connected to the pad cell and the pad direction are determined by the bit fields **IS** and **OS** and related internal direction signals. 1: The pad is connected to its associated GPIO cell. The direction is determined by the corresponding DP register bit.
Px	9	rwh	**Data of GPIOx** The written value and the hardware (pad) determined value are stored internally in different registers. Thus no conflict is possible when the register is written. The read delivers either of these values, depending on DPx, as explained in **Section 9.7.6 General Purpose I/O (on Page 430)**.
DPx	10	rw	**Direction Control for GPIOx** 0 GPIOx is input. The value applied on the input pad updates the Px field value. 1 GPIOx is output. The value set in the Px field drives the pad output value.

Field	Bits	Type	Description
PPEN	12	rw	**Push/Pull Enable** 0 Push/pull function of the pad is activated 1 Open drain function of the pad is activated
PDPU	14:13	rw	**Pullup/Pulldown Selection** 00 Pullup and pulldown resistors of the pad are deactivated 01 Pullup resistor of the pad is activated 10 Pulldown resistor of the pad is activated 11 Reserved
ENAQ	15	rw	0 Output function of the pad as determined by the other bits of this register 1 Output of the pad is set to tristate
RESERVED	3:2, 7:6, 11, 31:16	r	Reserved; these bits must be left at their reset values.

Table 111 GPIO Pad Reset Values

GPIO Pad Number	Reset Value (Hex)	GPIO Pad Number	Reset Value (Hex)	GPIO Pad Number	Reset Value (Hex)
00	8900[1)]	20	0010[2)]	40	0010[2)]
01	8900[1)]	21	4001[3)]		
02	8900[1)]	22	0010[2)]		
03	8900[1)]	23	0010[2)]		
04	8900[1)]	24	0010[2)]		
05	8900[1)]	25	0010[2)]		
06	8900[1)]	26	C900[4)]		
07	8900[1)]	27	C900[4)]		
08	8900[1)]	28	A900[5)]		
09	8900[1)]	29	C900[4)]		
10	8900[1)]	30	C900[4)]		
11	8900[1)]	31	C900[4)]		
12	8900[1)]	32	C900[4)]		
13	A900[5)]	33	C900[4)]		
14	8900[1)]	34	C900[4)]		
15	8900[1)]	35	C900[4)]		
16	8900[1)]	36	C900[4)]		
17	8900[1)]	37	C900[4)]		
18	9900[6)]	38	A900[5)]		
19	9900[6)]	39	0010[2)]		

1) Tristate.
2) Pad configured as Alt 0 Output.
3) Pad configured as Alt 0 Input with Pull Down active.
4) Tristate and Pull Down.
5) Tristate and Pull UP.
6) Tristate and Open Drain.

9.7.3 IDMX

The input demultiplexer IDMX connects the internal pad output OUTI to an input of a chip internal block input via one of the lines Ix. The maximal number of Ix lines depends on the actual size of the input select word **PCL_<pad>.IS**. The size may be 1 or 2 bits, allowing 1 or 3 different input lines. If all **IS** bits are 0, no input is selected. Others return 0 when read. The same is true for output select word **PCL_<pad>.OS**.

The respective port input is always connected to OUTI. Thus a read access to the port shows always the value of the respective input line, even if the port function of the pad is not activated.

The function of IDMX is described in **Table 112**.

Table 112 Input Programming

IS	I1 ... I3
00	Depending on the connected block input: inactive state of this input.
others	I_x for x = **PCL_<pad>.IS**: OUTI I_x for x /= **PCL_<pad>.IS**: Inactive state of connected block input

If different alternative input lines shall be connected to one input of a block, these lines are "ORed" or 'ANDed' before entering the block, depending on the inactive state of the block input. No other hardware is used outside of PCL. The actual input pad is selected by programming its respective input line Ix to be connected to OUTI and not selecting the respective Ix lines of all other related pads. For example, in case of inactive input state 0, these Ix lines are set to 0 and do not influence the output of the OR gate.

9.7.4 OMX

The output multiplexer OMX connects the GPIO output OP or one of several outputs of chip internal blocks to the pad data input DQ. The maximal number of block outputs lines Ox depends on the actual size of the output select word OS. OS is programmed in the register **PCL_<pad>**. The size may be 1 or 2 bits, allowing 1 or 3 different output lines in addition to OP. If all OS bits are 0, no output is selected.

The function of OMX is described in **Table 113**.

Table 113 Output Programming

PS	OS	DQ
1	xx	OP
0	00	0
0	others	DQ = O_{OS}

Each block output may be connected to an arbitrary number of pads without additional hardware. The actual output pad is selected by programming its respective output line Ox to be connected to DQ and deselecting the respective Ox lines of all other related pads.

During switching of OS, spikes may occur on the internal DQ line. If this is a problem, the output driver can be disabled temporarily as described in the following DIRCTL section.

Logical combinations of outputs: some logical combinations of block outputs are possible. If required, refer to **Chapter 3 Pin Descriptions**. They are selected like other output signals.

9.7.5 DIRCTL

The direction control block DIRCTL determines the pad direction out of the PS, IS and OS selection and the related DIRx input and sets ENQ correspondingly, as described hereafter.

If neither the GPIO nor an input or an output are selected, that is, PS = IS = OS = 0, then the pad is set to tristate, that is, ENQ is deactivated.

If GPIO is selected, that is, PS = 1, the direction is determined by DIRP.

If no GPIO, no input and one output are selected, that is, PS = IS = 0 and OS > 0, then the pad is used as an output, that is, ENQ is activated.

If no GPIO, one input and no output are selected, that is, PS = 0, IS > 0 and OS = 0, then the pad is used as an input, that is, ENQ is deactivated.

If no GPIO, one input and one output are selected, that is, PS = 0, IS > 0 and OS > 0, then the pad is used as an IO pad with the direction control signal DIR(OS). If the pad shall be used as a true bidirectional pad, the direction control signal will be delivered by the respective peripheral. If the pad shall be used as an output only, the direction control signal has a fixed value, always enabling the output driver. In this case setting IS > 0 allows to feed back the output signal to an internal E-GOLDvoice signal.

Table 114 Pad Function Depending on Programming of PCL_<pad>

ENAQ	PS	IS	OS	ENQ	Pad function	
					Input	**Output**
1	x	x		x	1 Refer to **Table 112 Input Programming** **(on Page 429)**	Tristate
0		1		x	not DIRP GPIO, direction depending on DIRP	
x	0	0		0	1	Not active Tristate
0	0	0		> 0	0	Not active DQ (**Table 112**)
x		0		>0	1	Refer to **Table 112** Tristate
0	0	>0	>0	not DIR_{OS}	Bidirectional, direction depending on DIR_{OS}	

ENAQ allows setting the pad to tristate and programming the desired value for OS at the same time. Thus the spikes which may occur on the OMX output DQ during switching will not be visible on the chip output. The output level can be asserted by setting PU or PD. In a following cycle, the output may be activated by de-asserting ENAQ and PU or PD while OS remains unchanged.

9.7.6 General Purpose I/O

In order to accept or generate external control signals, the PMB7880 provides a number of general purpose IO (GPIO) lines. Each GPIO is related to one pad, as shown in **Chapter 3 Pin Descriptions**. The pad may be configured to show the GPIO function or some other function, as shown in the subsections above and in **Chapter 3** in detail.

A general purpose IO cell x associated to a pad contains the data direction definition bit DPx and the data bit Px (see **Figure 173**).

It is always possible to write to or read from Px and DPx, even if the pad is not configured as port.

If DPx is set to 0, the data direction of the GPIO x is input. In this case reading of bit Px delivers the pad value.

If DPx is set to 1, the data direction of the GPIO x is output. In this case reading of bit Px delivers the value written to Px.

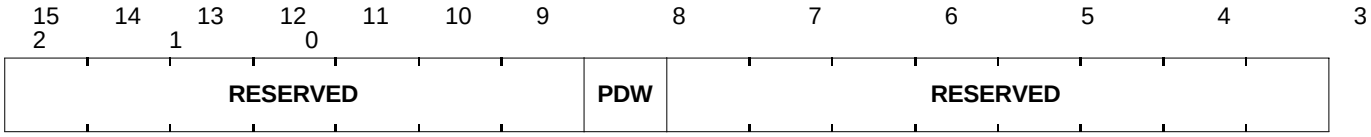
The EBU control block in the PCL controls the EBU Pads:

- A0 .. A22
- D0 .. D15
- CS0_N, CS1_N, CS3_N
- ADV_N
- WR_N
- RD_N
- OE_N
- BHE_N.
- RSTOUT_N

9.7.6.1 EBU Pull Down Control

EBU_PDC

EBU Pull Down Control Register **Reset value: 0000_H**



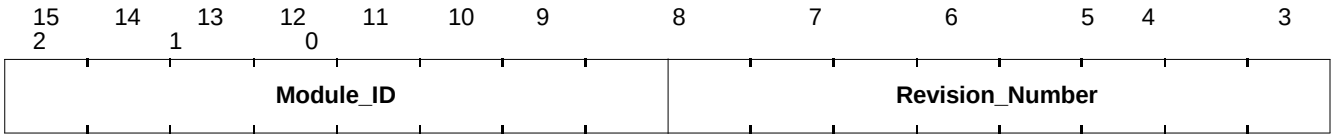
Field	Bits	Type	Description
PDW	8	rw	**Pull Down Weak** **PDW** is used in the Minimum Power Consumption Optimization and must be used the MCU IDLE command. 0 No input selected 1 A Pull-Down is set on the Address (A0..A23) and Data (D0..D15) EBU pads Refer to **Section 7.7.8.1 Idle Mode (on Page 251)**.
RESERVED	15:9, 7:0	r	Reserved; these bits must be left at their reset values.

9.7.7 Control Registers

9.7.7.1 Port Control Logic Identification Register

PCL_ID

Port Control Logic Identification Register **Reset value: 2305_H**



Field	Bits	Type	Description
Revision_Number	0:7	r	**Port Control Logic Revision Number** These hard-wired bits are used for module revision numbering.
Module_ID	8:15	r	**Port Control Logic Identification Number** These hard-wired bits are used for module identification numbering.

9.7.8 Flash Overwrite Protection

9.7.8.1 Functional Description

The external memory device is normally expected to be a flash device. Therefore, the names used in this section are focused on the flash device. It is however possible to utilize other type of external memory. All external memory is treated in the same manner.

The external flash memory device must be protected again an accidental overwrite. An overwrite is especially dangerous in the write-suspend mode when the flash on-chip write protection mechanism is already deactivated. The overwrite protection mechanism suppresses the activation of the PMB7880 external WRITE pin **$\overline{WR}$** during accesses to the CS0 or CS1 memory ranges. The WRITE pin **$\overline{WR}$** can only be activated when the core output write signal is set and a dedicated write-enable register for the dressed memory page contains a special pattern.

As this protection controls the **$\overline{WR}$** signal directly, it does not require a flash memory with an extra protection pin.

Note: The extra protection pin can be controlled by software driving a general purpose output pin.

Figure 175 shows the block schematic of the flash overwrite protection.

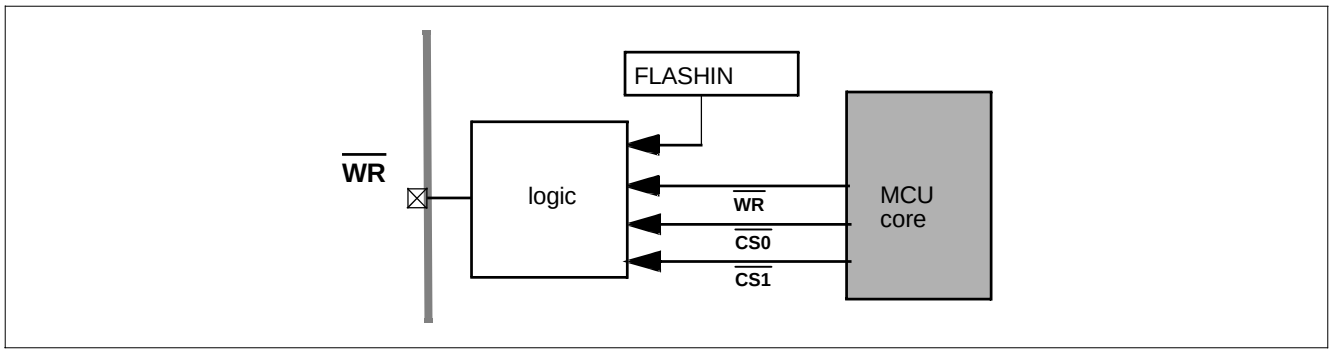


Figure 175 Flash Write Protection Functional Diagram

9.7.8.2 Flash Write Enable Register

FLASHIN

Flash Write Enable Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								WRITE_ENABLE_CS1				WRITE_ENABLE_CS0			

Field	Bits	Type	Description
WRITE_ENABLE_CS0	3:0	rw	6_H: Write on CS0 enabled Any other value:Write disabled
WRITE_ENABLE_CS1	7:4	rw	6_H: Write on CS1 enabled Any other value:Write disabled
RESERVED	15:8	rw	Reserved; these bits must be left at their reset values.

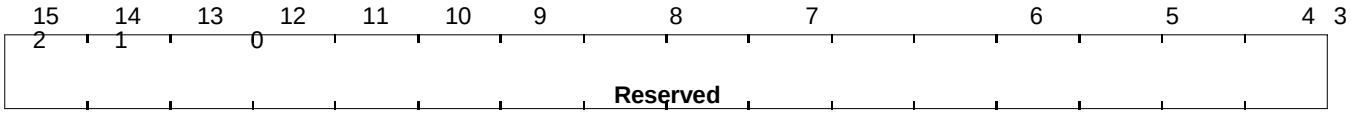
Note: Writing 0066_H to this register enables writing to both CS0 and CS1.

9.7.9 PCL E-Fuse Boxes and Corresponding Registers

The PMB7880 provides E-fuse boxes available via registers. These 6 Registers total 96 bits . Their values are programmed at IFX production line for each individual chip, and provide, for each chip, a unique identifier numberduring fabrication.

ID_SNUM0

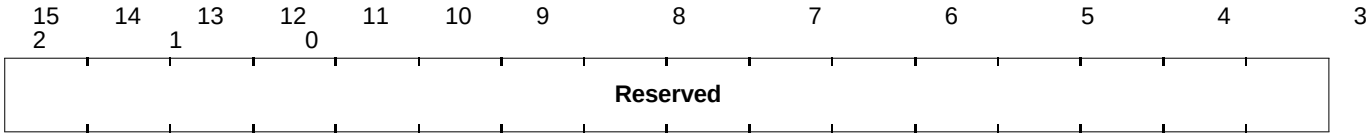
Serial Number Register 0 **Reset value: xxxx$_H$**



Field	Bits	Type	Description
Reserved	15:0	r	Reserved, these bits must be left at their manufacturing values.

ID_SNUM1

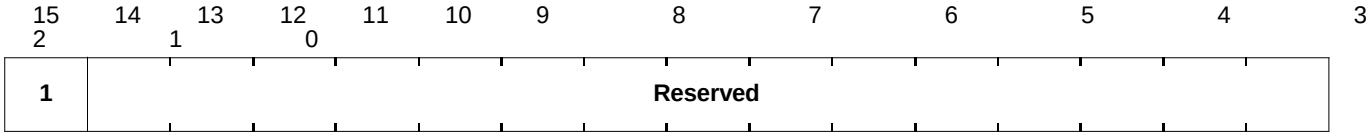
Serial Number Register 1 **Reset value: xxxx$_H$**



Field	Bits	Type	Description
Reserved	15:0	r	Reserved, these bits must be left at their manufacturing values.

ID_SNUM2

Serial Number Register 2 **Reset value: xxxx$_H$**



Field	Bits	Type	Description
1	15	r	Always fused to 1
Reserved	14:0	r	Reserved, these bits must be left at their manufacturing values.

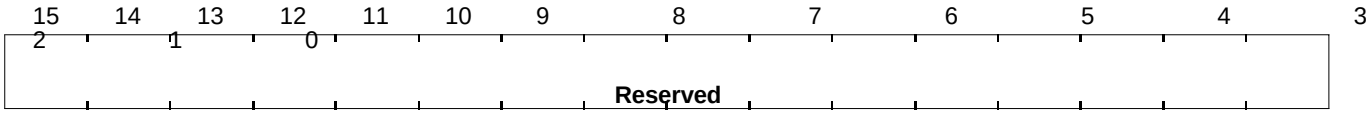
ID_SNUM3

Serial Number Register 3 **Reset value: xxxx$_H$**

Field	Bits	Type	Description
Reserved	15:0	r	Reserved, these bits must be left at their manufacturing values.

ID_SNUM4

Serial Number Register 4 **Reset value: xxxx$_H$**

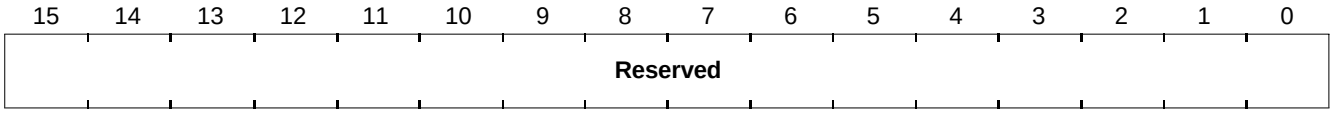


Field	Bits	Type	Description
Reserved	15:0	r	Reserved; these bits must be left at their manufacturing values.

ID_SNUM5

ID_SNUM5

Reset value: xxxx$_H$



Field	Bits	Type	Description
Reserved	15:0	r	Reserved; these bits must be left at their reset values.

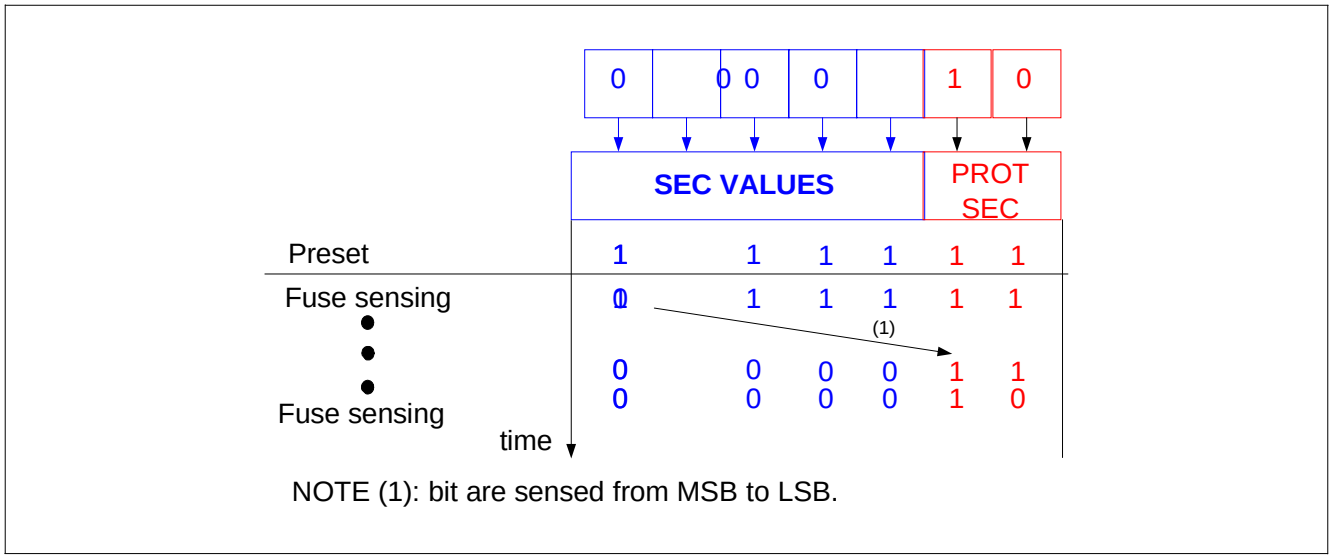
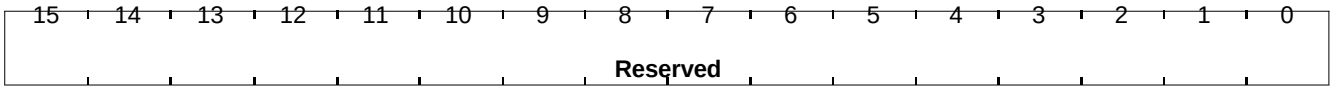


Figure 176 Fuse sensing of ID_SNUM5

ID_SNUM6

ID_SNUM6

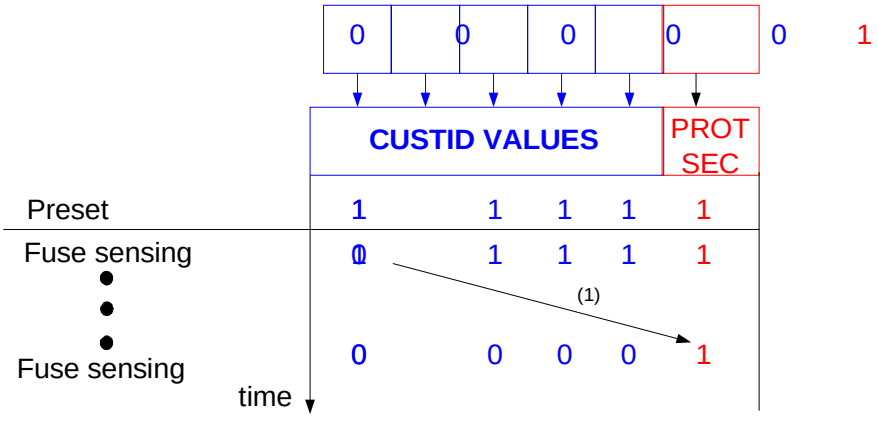
Reset value: xxxx$_H$



Field	Bits	Type	Description
Reserved	15:0	r	Reserved

Register connections

Figure 177 Fuse Sensing of ID_SDNUM6



NOTE (1): bit are sensed from MSB to LSB.

9.7.10 Internal Signal Monitoring

9.7.10.1 Functionality

This block is used to monitor internal signals.

The selected internal signals are redirected to two external signals: MON1 and MON2. The selection is done in two steps:

1. The selection of the block done with **MON_CRx.BLOCK_SELECT**.
2. The selection of the signal from the block done with **MON_CRx.SIGNAL_SELECT**.

Notes

1. *The signals MON1 and MON2 can be configured as an external input signal, in this case the MON1 and MON2 signals are connected to the **TEAKLite Interrupt Unit (on Page 302)**. The configuration of the MON1 signal and MON2 signal is independent; it is done by writing 1 into the **MON_CR1.ENAQ** for the MON1 signal and by writing 1 into the **MON_CR2.ENAQ** for the MON2 signal.*
2. *The pads MON1 and MON2 have a different usage during the system reset, refer to **Chapter 12 System Reset**.*

Figure 178 (on page 436) shows the signals path and control.

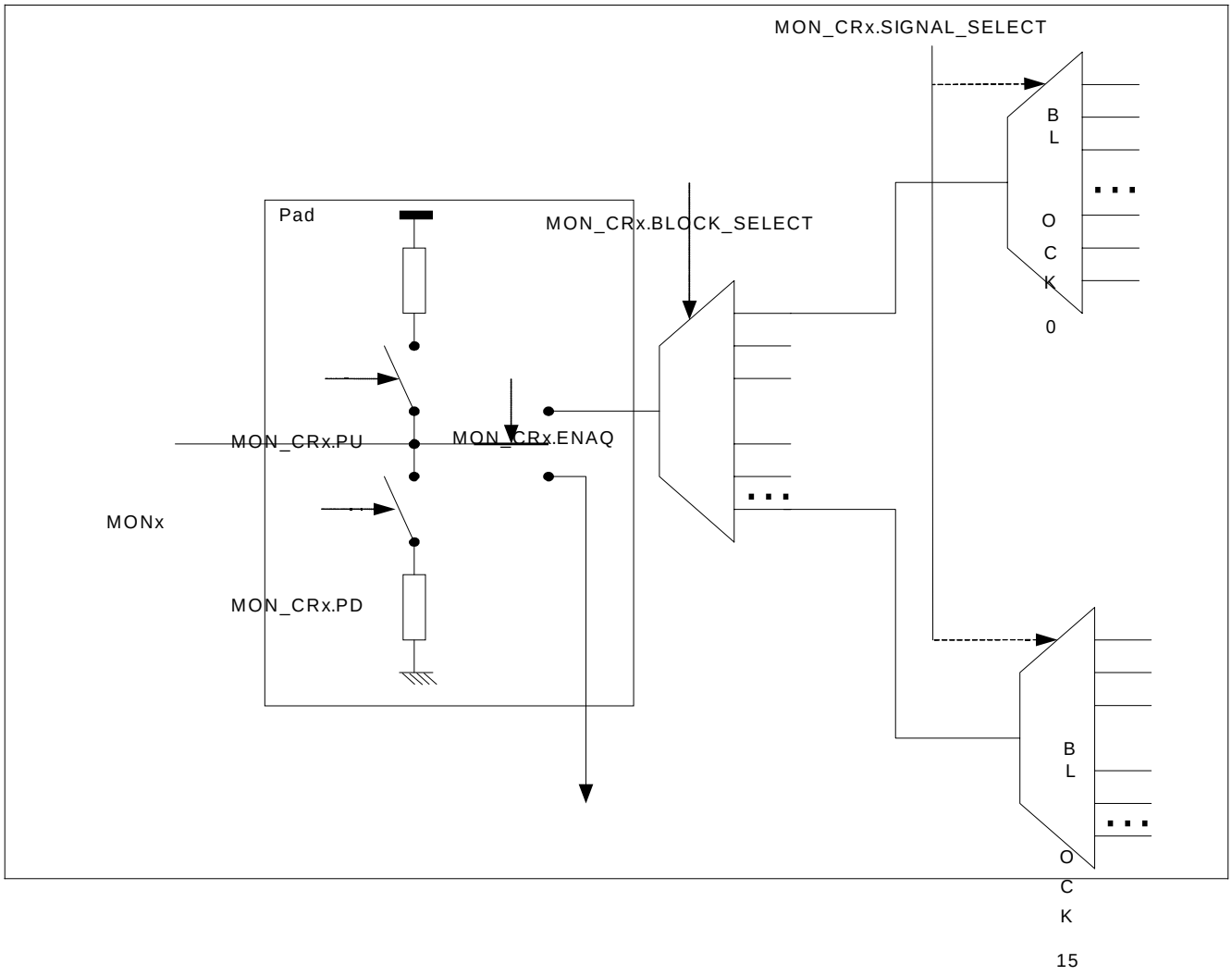


Figure 178 Monitoring of Internal Signals at the Monitor Pads

9.7.10.2 Control Registers

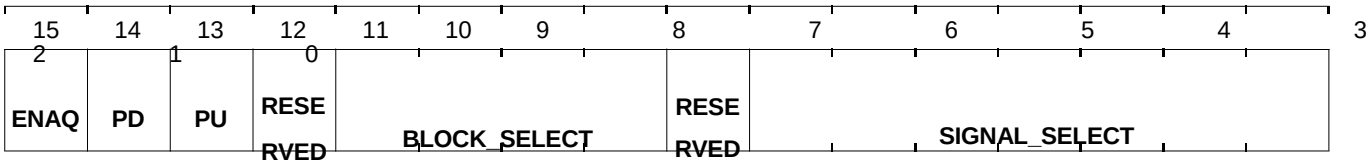
MON_CR1 controls the signal MON1.

MON_CR2 controls the signal MON2.

MON_CRx

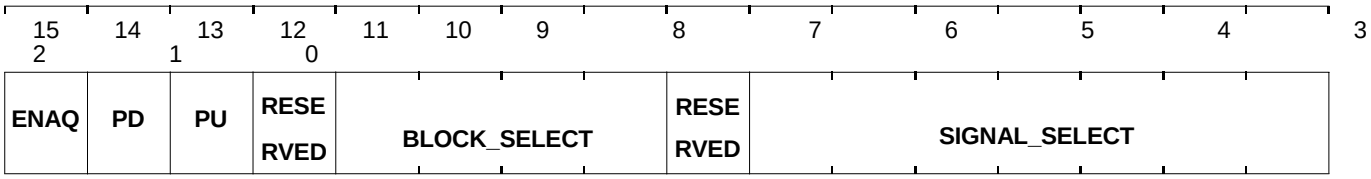
MON_CR1

Control Register 1 **Reset value: C900$_H$**



MON_CR2

Control Register 2 **Reset value: C900$_H$**



Field	Bits	Type	Description
SIGNAL_SELECT	6:0	rw	**Signal Selection**
BLOCK_SELECT	11:8	rw	**Block Selection** Refer to **Table 115**.
PD, PU	14:13	rw	**Pullup and Pulldown Enable** These bits activate pullup and pulldown resistors of the associated pad. 00 Pullup and pulldown resistors are deactivated 01 Pullup resistor of pad is activated 10 Pulldown resistor of pad is activated 11 Reserved
ENAQ	15	rw	**Pad Mode** 0 Pad in Output mode. 1 Pad in Input mode.
RESERVED	7, 12	r	Reserved; these bits must be left at their reset values.

Table 115 Block List

BLOCK_SELECT	Block Name
0000	**DSP_INT**
0001	**DSP_ANA**
0010	**Cipher A53**
0011	**CGU**
0100	**PADS**
0101	**INTR_EXT**
0110	**PCL_PER**
0111	Reserved
1000	**MCU_PER**
1001	**P3_PMU** (Baseband PMU digital)
1010	**PMU_SUPER** (PMU Macro)

Table 115 Block List (cont'd)

BLOCK_SELECT	Block Name
1011	TCU_SUPER
1100	Reserved
1101	Reserved
1110	Reserved
1111	Reserved

9.7.10.2.1 DSP_INT

Table 116 Signals from the DSP (Mainly Interrupts)

Signal Name	Select Signal (Hex)	Remarks
mcu0_int	01	DSP Interrupt A0
mcu1_int	02	DSP Interrupt A0
mcu2_int	03	DSP Interrupt A0
mcu3_int	04	DSP Interrupt A0
frame_int	05	DSP Interrupt A0
-	06	reserved
-	07	reserved
-	08	reserved
-	09	reserved
-	0A	reserved
codon_int	0B	DSP Interrupt A0
modu_int	0C	DSP Interrupt A0
chadec_int	0D	DSP Interrupt A0
eq_int	0E	DSP Interrupt A0
eqon_int	0F	DSP Interrupt A0
fcon_int	10	DSP Interrupt A0
monon_int	11	DSP Interrupt A0
scon_int	12	DSP Interrupt A0
bb_full_int	13	DSP Interrupt A0
i2s1rx_int	14	DSP Interrupt B0
-	15	reserved
-	16	reserved
-	17	reserved
-	18	reserved
-	19	reserved
vbtx_int	1A	DSP Interrupt B0
vbrx_int	1B	DSP Interrupt B0
-	1C	reserved
-	1D	reserved
-	1E	reserved
-	1F	reserved
-	20	reserved

Table 116 Signals from the DSP (Mainly Interrupts) (cont'd)

Signal Name	Select Signal (Hex)	Remarks
-	21	reserved
-	22	reserved
sysmcu_int	23	DSP Interrupt B0
ciph_int	24	DSP Interrupt 1
dtmr10_int	25	DSP Interrupt 1
dtmr11_int	26	DSP Interrupt 1
dtmr2_int	27	DSP Interrupt 1
dspin0_int	28	DSP Interrupt 1
dspin1_int	29	DSP Interrupt 1
monin1_int	2A	DSP Interrupt 1
monin2_int	2B	DSP Interrupt 1
monin3_int	2C	DSP Interrupt 1
monin4_int	2D	DSP Interrupt 1
intfw0	2E	DSP Interrupt 2
intfw1	2F	DSP Interrupt 2
intfw2	30	DSP Interrupt 2
intfw3	31	DSP Interrupt 2
intfw4	32	DSP Interrupt 2
intfw5	33	DSP Interrupt 2
intfw6	34	DSP Interrupt 2
intfw7	35	DSP Interrupt 2
intfw8	36	DSP Interrupt 2
intfw9	37	DSP Interrupt 2
intfw10	38	DSP Interrupt 2
intfw11	39	DSP Interrupt 2
intfw12	3A	DSP Interrupt 2
intfw13	3B	DSP Interrupt 2
intfw14	3C	DSP Interrupt 2
intfw15	3D	DSP Interrupt 2
int0	3E	DSP Interrupt Line 0
int1	3F	DSP Interrupt Line 1
int2	40	DSP Interrupt Line 2
tomcu0_int	41	Interrupt to MCU
tomcu1_int	42	Interrupt to MCU
tomcu2_int	43	Interrupt to MCU
tomcu3_int	44	Interrupt to MCU
-	45 to 7F	reserved

9.7.10.2.2 DSP_ANA

Table 117 Signals from DSP

Signal Name	Select Signal (Hex)	Remarks
mon_rxclear	01	Audio-FE
mon_txclear	02	Audio-FE
mon_data_rrq_rx	03	Audio-FE
mon_data_wrq_tx	04	Audio-FE
mon_powerup_int_asmd_tx	05	Audio-FE
mon_powerup_int_asmd_rx	06	Audio-FE
mon_zero_detect	07	Audio-FE
mon_data_vbrx_left	08	Audio-FE
mon_data_vbrx_lsb4(0)	09	Audio-FE
mon_data_vbrx_lsb4(1)	0A	Audio-FE
mon_data_vbrx_lsb4(2)	0B	Audio-FE
mon_data_vbrx_lsb4(3)	0C	Audio-FE
mon_clk_vbrx	0D	Audio-FE
mon_data_vbtx	0E	Audio-FE
mon_clk_vbtx_dith	0F	Audio-FE
mon_clk_vbtx	10	Audio-FE
mon_gclock_pll	11	BB-Receive
mon_clk_dither	12	BB-Receive
mon_gclock_mix	13	BB-Receive
mon_pdm1q	14	BB-Receive
mon_pdm2q	15	BB-Receive
mon_pdm1i	16	BB-Receive
mon_pdm2i	17	BB-Receive
mon_clk_bbrx	18	BB-Receive
mon_ciphering_ready	19	GSM-Cipher
mon_start_ciphering	1A	GSM-Cipher
mon_ecry_dcry_data	1B	GSM-Cipher
mon_dcry_valid	1C	GSM-Cipher
mon_ecry_valid	1D	GSM-Cipher
mon_gclock_dsp	1E	GSM-Cipher
mon_gclock_ram	1F	GSM-Cipher
mon_gclock_cipher	20	GSM-Cipher
mon_rx_state(0)	21	I2S1 (DSP)
mon_rx_state(1)	22	I2S1 (DSP)
mon_rx_state(2)	23	I2S1 (DSP)
mon_rx_state(3)	24	I2S1 (DSP)
mon_tx_state(0)	25	I2S1 (DSP)
mon_tx_state(1)	26	I2S1 (DSP)
mon_tx_state(2)	27	I2S1 (DSP)
mon_tx_state(3)	28	I2S1 (DSP)
mon_cl2_state(0)	29	I2S1 (DSP)

Table 117 Signals from DSP (cont'd)

Signal Name	Select Signal (Hex)	Remarks
mon_cl2_state(1)	2A	I2S1 (DSP)
mon_cl2_state(2)	2B	I2S1 (DSP)
mon_cl1_state(0)	2C	I2S1 (DSP)
mon_cl1_state(1)	2D	I2S1 (DSP)
mon_cl1_state(2)	2E	I2S1 (DSP)
mon_i2s_data_wrq	2F	I2S1 (DSP)
mon_i2s_data_rrq	30	I2S1 (DSP)
-	31	reserved
-	32	reserved
-	33	reserved
-	34	reserved
-	35	reserved
-	36	reserved
-	37	reserved
-	38	reserved
-	39	reserved
-	3A	reserved
-	3B	reserved
-	3C	reserved
-	3D	reserved
-	3E	reserved
-	3F	reserved
-	40	reserved
-	41	reserved
-	42	reserved
-	43	reserved
-	44	reserved
-	45	reserved
-	46	reserved
-	47	reserved
-	48	reserved
mon_a53_ciphering_ready	49	CIPHER_A53
mon_a53_start_ciphering	4A	CIPHER_A53
mon_a53_ecry_dcry_data	4B	CIPHER_A53
mon_a53_dcry_valid	4C	CIPHER_A53
mon_a53_ecry_valid	4D	CIPHER_A53
mon_a53_gclock_dsp	4E	CIPHER_A53
mon_a53_gclock_ram	4F	CIPHER_A53
mon_a53_gclock_cipher	50	CIPHER_A53
mon_dpram_data(0)	51	BB-Transmit
mon_dpram_req	52	BB-Transmit

Table 117 Signals from DSP (cont'd)

Signal Name	Select Signal (Hex)	Remarks
mon_gclk_dsp_per_mod	53	BB-Transmit
mon_clk_bbtx	54	BB-Transmit
mon_vh_status	55	Viterbi-Channel-Dec.
mon_dec_busy	56	Viterbi-Channel-Dec.
mon_dec_active	57	Viterbi-Channel-Dec.
mon_pointer_dec_new	58	Viterbi-Channel-Dec.
mon_res_dec	59	Viterbi-Channel-Dec.
mon_res_dsp_int	5A	Viterbi-Channel-Dec.
mon_vh_status	5B	Viterbi-Equalizer
mon_eq_busy	5C	Viterbi-Equalizer
mon_pointer_right_hs_new	5D	Viterbi-Equalizer
mon_pointer_left_hs_new	5E	Viterbi-Equalizer
mon_res_eq	5F	Viterbi-Equalizer
mon_res_dsp_int	60	Viterbi-Equalizer
mon_en_reg_read	61	DSP
mon_ppap_page	62	DSP
mon_dxap_page	63	DSP
mon_dspdis	64	DSP
pstatusp(0)	65	OCEM (DSP)
pstatusp(1)	66	OCEM (DSP)
pstatusp(2)	67	OCEM (DSP)
pstatusp(3)	68	OCEM (DSP)
psftp	69	OCEM (DSP)
ppwp	6A	OCEM (DSP)
pprp	6B	OCEM (DSP)
Reserved	6C	
Reserved	6D	
pdummyp	6E	OCEM (DSP)
ddtvmp	6F	OCEM (DSP)
odebugp	70	OCEM (DSP)
obootp	71	OCEM (DSP)
oabortn	72	OCEM (DSP)
btrapreqp	73	OCEM (DSP)
mon_rst_mix_n	74	DSP
mon_rst_dsp_n	75	DSP
mon_rst_dsp_per_n	76	DSP
mon_rst_dsp_mem_n	77	DSP
seib_j_tdo_p	78	SEIB (DSP)
seib_j_abort_iut_n	79	SEIB (DSP)
waitp	7A	SEIB (DSP)
enocemn	7B	SEIB (DSP)

Table 117 Signals from DSP (cont'd)

Signal Name	Select Signal (Hex)	Remarks
state_kgcore(0)	7C	CIPHER_A53
state_kgcore(1)	7D	CIPHER_A53
state_kgcore(2)	7E	CIPHER_A53
state_kgcore(3)	7F	CIPHER_A53

9.7.10.2.3 Cipher A53

Table 118 Signals from A53 Peripherals

Signal Name	Select Signal (Hex)	Remarks
mon_init_kgcore	01	Cipher_A53
mon_clear_init	02	Cipher_A53
mon_state_ctrl(0)	03	Cipher_A53
mon_state_ctrl(1)	04	Cipher_A53
mon_state_ctrl(2)	05	Cipher_A53
mon_state_kgcore(0)	06	Cipher_A53
mon_state_kgcore(1)	07	Cipher_A53
mon_state_kgcore(2)	08	Cipher_A53
mon_state_kgcore(3)	09	Cipher_A53
mon_block_count(0)	0A	Cipher_A53
mon_block_count(1)	0B	Cipher_A53
mon_block_count(2)	0C	Cipher_A53
mon_block_count(3)	0D	Cipher_A53
mon_register_A_value(0)	0E	Cipher_A53
mon_register_A_value(1)	0F	Cipher_A53
mon_register_A_value(2)	10	Cipher_A53
mon_register_A_value(3)	11	Cipher_A53
mon_register_A_value(4)	12	Cipher_A53
mon_register_A_value(5)	13	Cipher_A53
mon_register_A_value(6)	14	Cipher_A53
mon_register_A_value(7)	15	Cipher_A53
mon_register_A_value(8)	16	Cipher_A53
mon_register_A_value(9)	17	Cipher_A53
mon_register_A_value(10)	18	Cipher_A53
mon_register_A_value(11)	19	Cipher_A53
mon_register_A_value(12)	1A	Cipher_A53
mon_register_A_value(13)	1B	Cipher_A53
mon_register_A_value(14)	1C	Cipher_A53
mon_register_A_value(15)	1D	Cipher_A53
mon_register_A_value(16)	1E	Cipher_A53
mon_register_A_value(17)	1F	Cipher_A53
mon_register_A_value(18)	20	Cipher_A53
mon_register_A_value(19)	21	Cipher_A53

Table 118 Signals from A53 Peripherals (cont'd)

Signal Name	Select Signal (Hex)	Remarks
mon_register_A_value(20)	22	Cipher_A53
mon_register_A_value(21)	23	Cipher_A53
mon_register_A_value(22)	24	Cipher_A53
mon_register_A_value(23)	25	Cipher_A53
mon_register_A_value(24)	26	Cipher_A53
mon_register_A_value(25)	27	Cipher_A53
mon_register_A_value(26)	28	Cipher_A53
mon_register_A_value(27)	29	Cipher_A53
mon_register_A_value(28)	2A	Cipher_A53
mon_register_A_value(29)	2B	Cipher_A53
mon_register_A_value(30)	2C	Cipher_A53
mon_register_A_value(31)	2D	Cipher_A53
mon_register_A_value(32)	2E	Cipher_A53
mon_register_A_value(33)	2F	Cipher_A53
mon_register_A_value(34)	30	Cipher_A53
mon_register_A_value(35)	31	Cipher_A53
mon_register_A_value(36)	32	Cipher_A53
mon_register_A_value(37)	33	Cipher_A53
mon_register_A_value(38)	34	Cipher_A53
mon_register_A_value(39)	35	Cipher_A53
mon_register_A_value(40)	36	Cipher_A53
mon_register_A_value(41)	37	Cipher_A53
mon_register_A_value(42)	38	Cipher_A53
mon_register_A_value(43)	39	Cipher_A53
mon_register_A_value(44)	3A	Cipher_A53
mon_register_A_value(45)	3B	Cipher_A53
mon_register_A_value(46)	3C	Cipher_A53
mon_register_A_value(47)	3D	Cipher_A53
mon_register_A_value(48)	3E	Cipher_A53
mon_register_A_value(49)	3F	Cipher_A53
mon_register_A_value(50)	40	Cipher_A53
mon_register_A_value(51)	41	Cipher_A53
mon_register_A_value(52)	42	Cipher_A53
mon_register_A_value(53)	43	Cipher_A53
mon_register_A_value(54)	44	Cipher_A53
mon_register_A_value(55)	45	Cipher_A53
mon_register_A_value(56)	46	Cipher_A53
mon_register_A_value(57)	47	Cipher_A53
mon_register_A_value(58)	48	Cipher_A53
mon_register_A_value(59)	49	Cipher_A53
mon_register_A_value(60)	4A	Cipher_A53

Table 118 Signals from A53 Peripherals (cont'd)

Signal Name	Select Signal (Hex)	Remarks
mon_register_A_value(61)	4B	Cipher_A53
mon_register_A_value(62)	4C	Cipher_A53
mon_register_A_value(63)	4D	Cipher_A53
mon_kgcore_key_stored(0)	4E	Cipher_A53
mon_kgcore_key_stored(1)	4F	Cipher_A53
mon_kgcore_key_stored(2)	50	Cipher_A53
mon_kgcore_key_stored(3)	51	Cipher_A53
mon_kgcore_key_stored(4)	52	Cipher_A53
mon_kgcore_key_stored(5)	53	Cipher_A53
mon_kgcore_key_stored(6)	54	Cipher_A53
mon_kgcore_key_stored(7)	55	Cipher_A53
mon_kgcore_key_stored(8)	56	Cipher_A53
mon_kgcore_key_stored(9)	57	Cipher_A53
mon_kgcore_key_stored(10)	58	Cipher_A53
mon_kgcore_key_stored(11)	59	Cipher_A53
mon_kgcore_key_stored(12)	5A	Cipher_A53
mon_kgcore_key_stored(13)	5B	Cipher_A53
mon_kgcore_key_stored(14)	5C	Cipher_A53
mon_kgcore_key_stored(15)	5D	Cipher_A53
mon_kgcore_key_stored(16)	5E	Cipher_A53
mon_kgcore_key_stored(17)	5F	Cipher_A53
mon_kgcore_key_stored(18)	60	Cipher_A53
mon_kgcore_key_stored(19)	61	Cipher_A53
mon_kgcore_key_stored(20)	62	Cipher_A53
mon_kgcore_key_stored(21)	63	Cipher_A53
mon_kgcore_key_stored(22)	64	Cipher_A53
mon_kgcore_key_stored(23)	65	Cipher_A53
mon_kgcore_key_stored(24)	66	Cipher_A53
mon_kgcore_key_stored(25)	67	Cipher_A53
mon_kgcore_key_stored(26)	68	Cipher_A53
mon_kgcore_key_stored(27)	69	Cipher_A53
mon_kgcore_key_stored(28)	6A	Cipher_A53
mon_kgcore_key_stored(29)	6B	Cipher_A53
mon_kgcore_key_stored(30)	6C	Cipher_A53
mon_kgcore_key_stored(31)	6D	Cipher_A53
mon_kgcore_key_stored(32)	6E	Cipher_A53
mon_kgcore_key_stored(33)	6F	Cipher_A53
mon_kgcore_key_stored(34)	70	Cipher_A53
mon_kgcore_key_stored(35)	71	Cipher_A53
mon_kgcore_key_stored(36)	72	Cipher_A53
mon_kgcore_key_stored(37)	73	Cipher_A53

Table 118 Signals from A53 Peripherals (cont'd)

Signal Name	Select Signal (Hex)	Remarks
mon_kgcore_key_stored(38)	74	Cipher_A53
mon_kgcore_key_stored(39)	75	Cipher_A53
mon_kgcore_key_stored(40)	76	Cipher_A53
mon_kgcore_key_stored(41)	77	Cipher_A53
mon_kgcore_key_stored(42)	78	Cipher_A53
mon_kgcore_key_stored(43)	79	Cipher_A53
mon_kgcore_key_stored(44)	7A	Cipher_A53
mon_kgcore_key_stored(45)	7B	Cipher_A53
mon_kgcore_key_stored(46)	7C	Cipher_A53
mon_kgcore_key_stored(47)	7D	Cipher_A53
mon_ecry_dcry_add	7E	Cipher_A53
-	7F	Reserved

9.7.10.2.4 CGU

Table 119 Signals from the CGU

Signal Name	Select Signal (Hex)	Remarks
cgu_rtc_clk_en	01	RTC clock enable[1)]
cgu_capcom2_clk_en	02	CAPCOM2 clock enable[1)]
cgu_capcom1_clk_en	03	CAPCOM1 clock enable[1)]
cgu_ssc_clk_en	04	SSC clock enable[1)]
cgu_asc1_clk_en	05	ASC1 clock enable[1)]
cgu_asc0_clk_en	06	ASC0 clock enable[1)]
cgu_master_access	07	Master Access clock enable[1)]
cgu_xper_clk_en	08	X-Bus Peripheral clock enable[1)]
cgu_xbus_clk_en	09	X-Bus clock enable[1)]
cgu_pdbus_clk_en	0A	PD-Bus clock enable[1)]
cgu_clk_ms	0B	Mixed Signal Clock. The output frequency is divided by 2.
cgu_clk_dsp	0C	DSP Clock. The output frequency is divided by 2. cgu_clk_afc
	0D	AFC Clock. The output frequency is divided by 2.
cgu_clk_mem	0E	Local Memory Bus Clock. The output frequency is divided by 2.
cgu_clk_x	0F	X-Bus Clock. The output frequency is divided by 2.
cgu_clk_pd	10	PD-Bus Clock. The output frequency is divided by 2.
cgu_c166_clk_neg	11	C166S Negative Clock. The output frequency is divided by 2.
cgu_c166_clk_pos2	12	C166S Positive 2 Clock. The output frequency is divided by 2.
cgu_c166_clk_pos1	13	C166S Positive 1 Clock. The output frequency is divided by 2.
cgu_clk_master	14	Master Access Clock. The output frequency is divided by 2.
pll_clk_phs1	15	Phase Shifter 1 Clock. The output frequency is divided by 2.
pll_clk_pll	16	PLL Clock. The output frequency is divided by 2.
pll_clk_in	17	Shaper Output clock. The output frequency is divided by 2.
rtc_clk_32k	18	32kHz clock from RTC. The output frequency is divided by 2.

Table 119 Signals from the CGU (cont'd)

Signal Name	Select Signal (Hex)	Remarks
sccu_resa	1A	SCCU
slp_vcxo_off	1B	SCCU
pre_wakeup	1C	SCCU
hwwup	1D	SCCU
sccu_vcxo_en_del	1E	SCCU
fr_dsp_out	1F	SCCU
fr_ana_out	20	SCCU
-	21	Reserved
slprst	22	Sleep counter reset
reset_32	23	Power on reset for clk_32
reset_13	24	Power on reset for clk_13
-	25	Reserved
-	26	Reserved
res_ucslp	27	SCCU
res_ucwup	28	SCCU
clk_cpu_en_del	29	SCCU
clk_gsm_on	2A	SCCU
sccu_shap_en_del	2B	SCCU
-	2C	Reserved
tststate13m(0)	2D	13mHz state machine for sleep function
tststate13m(1)	2E	13mHz state machine for sleep function
tststate32k(0)	2F	32 kHz state machine for sleep function
tststate32k(1)	30	32 kHz state machine for sleep function
status(0)	31	Main state machine status
status(1)	32	Main state machine status
status(2)	33	Main state machine status
status(3)	34	Main state machine status
status(4)	35	Main state machine status
prewup_int_o	36	Pre-wakeup interrupt
functionnal_mode_i	37	'1' if functionnal mode set
hdfr_lmu_reset_s	38	HDFR reset for LMU RAM
hdfr_dsp_reset_s	39	HDFR reset for the DSP RAM
efuse_sense_done_s	3A	Main sensing done when goes high
ready_out_test_s	3B	'1' if all the efuse readyouts at '1'
release_cpu_dsp_reset_s	3C	Set to '1' at the end of the Band Gap trimming to release cpu reset then.

1) The output clock is the result of the logical AND combination from the source clock in the clock enable signal. With a enable clock equal to 1 (logical value) the output clock will be equal to the source clock, that is, the visible enable signal values:
- If output clk is equal to source clk, monitor signal is set to constant '1'
- If output clk is half of source clk, monitor signal is set to '1' every second source clk, else set to '0'
- etc.

9.7.10.2.5 PADS

Table 120 Pad Signals

Signal Name	Select Signal (Hex)	Remarks
KP0	01	Keypad
KP1	02	Keypad
KP2	03	Keypad
KP3	04	Keypad
KP4	05	Keypad
KP5	06	Keypad
KP6	07	Keypad
KP7	08	Keypad
KP8	09	Keypad
KP9	0A	Keypad
RXD	0B	ASC0
TXD	0C	ASC0
ASC_RTS_n	0D	ASC0
ASC_CTS_n	0E	ASC0
SSC0_CLK	0F	PD SSC
SSC0_MRST	10	PD SSC
SSC0_MTSR	11	PD SSC
DISP_REST	12	Display Reset
SCL	13	I2C
SDA	14	I2C
CCVZ_n	15	SIM Card I/F
CCIN	16	SIM Card I/F
CS1_n	17	External Bus I/F
BHE_n	18	External Bus I/F
ADV_n	19	External Bus I/F
OE_N	1A	External Bus I/F
T_OUT2	1B	Gsm TDMA Timer
T_OUT3	1C	Gsm TDMA Timer
T_OUT4	1D	Gsm TDMA Timer
T_OUT8	1E	Gsm TDMA Timer
CLKOUT	1F	Other Functional Pins: Clocks and control
TRIG_IN	20	Other Functional Pins: Clocks and control
RSTOUT_n	21	Other Functional Pins: Clocks and control
CC00IO	22	Other Functional Pins: Clocks and control
W_LED_DRV	23	LED drv
BACKLIGHT	24	LED drc
VIB_CONTROL	25	Other Functional Pins: Clocks and control
T2IN	26	GPT
NMI_I	27	NMI

Table 120 Pad Signals (cont'd)

Signal Name	Select Signal (Hex)	Remarks
CS3_n	28	External Bus I/F
A22	29	External Bus I/F
CCIO	2A	SIM Card I/F
CCLK	2B	SIM Card I/F
CCRST	2C	SIM Card I/F
T_OUT0	2D	GSM TDMA Timer
D0	2E	External Bus I/F
D1	2F	External Bus I/F
D2	30	External Bus I/F
D3	31	External Bus I/F
D4	32	External Bus I/F
D5	33	External Bus I/F
D6	34	External Bus I/F
D7	35	External Bus I/F
D8	36	External Bus I/F
D9	37	External Bus I/F
D10	38	External Bus I/F
D11	39	External Bus I/F
D12	3A	External Bus I/F
D13	3B	External Bus I/F
D14	3C	External Bus I/F
D15	3D	External Bus I/F
A0	3E	External Bus I/F
A1	3F	External Bus I/F
A2	40	External Bus I/F
A3	41	External Bus I/F
A4	42	External Bus I/F
A5	43	External Bus I/F
A6	44	External Bus I/F
A7	45	External Bus I/F
A8	46	External Bus I/F
A9	47	External Bus I/F
A10	48	External Bus I/F
A11	49	External Bus I/F
A12	4A	External Bus I/F
A13	4B	External Bus I/F
A14	4C	External Bus I/F
A15	4D	External Bus I/F
A16	4E	External Bus I/F
A17	4F	External Bus I/F
A18	50	External Bus I/F

Table 120 Pad Signals (cont'd)

Signal Name	Select Signal (Hex)	Remarks
A19	51	External Bus I/F
A20	52	External Bus I/F
A21	53	External Bus I/F
CS0_n	54	External Bus I/F
RD_n	55	External Bus I/F
WR_n	56	External Bus I/F

9.7.10.2.6 INTR_EXT

Table 121 INTR_EXT Signals

Signal Name	Select Signal (Hex)	Remarks
CC0_INT	01	CAPCOM-Register-0
CC1_INT	02	CAPCOM-Register-1
CC2_INT	03	CAPCOM-Register-2
CC3_INT	04	CAPCOM-Register-3
CC4_INT	05	CAPCOM-Register-4
CC5_INT	06	CAPCOM-Register-5
CC6_INT	07	CAPCOM-Register-6
CC7_INT	08	CAPCOM-Register-7
CC16_INT	09	CAPCOM-register-16
CC17_INT	0A	CAPCOM-register-17
CC18_INT	0B	CAPCOM-register-18
CC19_INT	0C	CAPCOM-register-19
CC20_INT	0D	CAPCOM-register-20
CC21_INT	0E	CAPCOM-register-21
CC22_INT	0F	CAPCOM-register-22
CC23_INT	10	CAPCOM-register-23
T0_INT	11	CAPCOM-Timer-0
T1_INT	12	CAPCOM-Timer-1
T2_INT	13	GPT1-Timer-2
T3_INT	14	GPT1-Timer-3
T4_INT	15	GPT1-Timer-4
T5_INT	16	GPT2-Timer-5
T6_INT	17	GPT2-Timer-6
T7_INT	18	CAPCOM-Timer-7
T8_INT	19	CAPCOM-Timer-8
CR_INT	1A	GPT2-CAPREL-Register
RTC_INT	1B	from RTC
RTC_T14_INT	1C	from RTC
S0T_INT	1D	ASC0-Transmit
S0R_INT	1E	ASC0-Receive
S0E_INT	1F	ASC0-Error

Table 121 INTR_EXT Signals (cont'd)

Signal Name	Select Signal (Hex)	Remarks
S0TB_INT	20	ASC0-Transmit-Buffer
S0ABST_INT	21	ASC0-Autobaud-Start
S0ABDET_INT	22	ASC0-Autobaud-Detect
S0CTS_INT	23	ASC0-Cts
S0TM0_INT	24	ASC0-Tm0
SSC0T_INT	25	SSC-Transmit
SSC0R_INT	26	SSC-Receive
SSC0E_INT	27	SSC-Error
RES1_INT	28	Reserved
PM_INT	29	Reserved
RES3_INT	2A	Reserved
KPD_INT	2B	Keypad-interrupt
ECO_INT	2C	Power-Management
S0AS_INT	2D	uC-int from autostart
T_INT1	2E	T_INT1-of-GSM-Timer
T_INT2	2F	T_INT2-of-GSM-Timer
TIM0_INT	30	INT_GP(0)-of-GSM-Timer
TIM1_INT	31	INT_GP(1)-of-GSM-Timer
TIM2_INT	32	INT_GP(2)-of-GSM-Timer
TIM3_INT	33	INT_GP(3)-of-GSM-Timer
TIM4_INT	34	INT_GP(4)-of-GSM-Timer
MEAS0_INT	35	Measurement interrupt
MEAS1_INT	36	Measurement interrupt
MEAS0_TOGGLE_INT	37	Measurement int. toggled
MEAS1_TOGGLE_INT	38	Measurement int. toggled
RFSSOT_INT	39	RFSSSTINT-of-GSM-Timer
IIC_D_INT	3A	I2C data
IIC_P_INT	3B	I2C protocol
IIC_E_INT	3C	I2C error
SIMOK_INT	3D	SIM Card-Interface-OK
SIMERR_INT	3E	SIM Card-Interface-BAD
SIMIN_INT	3F	SIM Card-in-(pad CCIN => from SIMCARD)
EX0_INT	40	External-Interrupt-0 and/or s0rint
EX1_INT	41	External-Interrupt-1 and/or siminint
EX2_INT	42	External-Int-2 and/or kpdint
EX3_INT	43	External-Interrupt-3 and/or rtc_int
EX4_INT	44	External-Interrupt-4 and/or ex4bint
EX5_INT	45	External-Interrupt-5 and/or ex5bint
EX6_INT	46	External-Int-6 and/or tim2int
EX7_INT	47	External-Int-7 and/or t3int
FROM_DSP_TO_MCU0_INT	48	DSP-Interrupt-0

Table 121 INTR_EXT Signals (cont'd)

Signal Name	Select Signal (Hex)	Remarks
FROM_DSP_TO_MCU1_INT	49	DSP-Interrupt-1
FROM_DSP_TO_MCU2_INT	4A	DSP-Interrupt-2
FROM_DSP_TO_MCU3_INT	4B	DSP-Interrupt-3
PECLIN_INT	4C	PEC-link
EX4B_INT	4D	EX4B
EX5B_INT	4E	EX5B
GPRS_INT	4F	GPRS Interrupt

9.7.10.2.7 PCL_PER

Table 122 PCL_PER Signals

Signal Name	Select Signal (Hex)	Description
chipid_efb_00	01	
chipid_efb_01	02	
chipid_efb_02	03	
chipid_efb_03	04	
chipid_efb_04	05	
chipid_efb_05	06	
chipid_efb_06	07	
chipid_efb_07	08	
chipid_efb_08	09	
chipid_efb_09	0A	
chipid_efb_10	0B	
chipid_efb_11	0C	
chipid_efb_12	0D	
chipid_efb_13	0E	
chipid_efb_14	0F	
chipid_efb_15	10	
chipid_efb_16	11	
chipid_efb_17	12	
chipid_efb_18	13	
chipid_efb_19	14	
chipid_efb_20	15	
chipid_efb_21	16	
chipid_efb_22	17	
chipid_efb_23	18	
chipid_efb_24	19	
chipid_efb_25	1A	
chipid_efb_26	1B	
chipid_efb_27	1C	
chipid_efb_28	1D	
chipid_efb_29	1E	

Table 122 PCL_PER Signals (cont'd)

Signal Name	Select Signal (Hex)	Description
chipid_efb_30	1F	
chipid_efb_31	20	
chipid_efb_32	21	
chipid_efb_33	22	
chipid_efb_34	23	
chipid_efb_35	24	
chipid_efb_36	25	
chipid_efb_37	26	
chipid_efb_38	27	
chipid_efb_39	28	
chipid_efb_40	29	
chipid_efb_41	2A	
chipid_efb_42	2B	
chipid_efb_43	2C	
chipid_efb_44	2D	
chipid_efb_45	2E	
chipid_efb_46	2F	
chipid_efb_47	30	
chipid_efb_48	31	
chipid_efb_49	32	
chipid_efb_50	33	
chipid_efb_51	34	
chipid_efb_52	35	
chipid_efb_53	36	
chipid_efb_54	37	
chipid_efb_55	38	
chipid_efb_56	39	
chipid_efb_57	3A	
chipid_efb_58	3B	
chipid_efb_59	3C	
chipid_efb_60	3D	
chipid_efb_61	3E	
chipid_efb_62	3F	
chipid_efb_63	40	
chipid_efb_64	41	
chipid_efb_65	42	
chipid_efb_66	43	
chipid_efb_67	44	
chipid_efb_68	45	
chipid_efb_69	46	
chipid_efb_70	47	

Table 122 PCL_PER Signals (cont'd)

Signal Name	Select Signal (Hex)	Description
chipid_efb_71	48	
chipid_efb_72	49	
chipid_efb_73	4A	
chipid_efb_74	4B	
chipid_efb_75	4C	
chipid_efb_76	4D	
chipid_efb_77	4E	
chipid_efb_78	4F	
security_efb_0	50	
security_efb_1	51	
security_efb_2	52	
security_efb_3	53	
security_efb_4	54	
security_efb_5	55	
security_efb_6	56	
security_efb_7	57	
security_efb_8	58	
security_efb_9	59	
security_efb_10	5A	
cusid_efb_0	5B	
cusid_efb_1	5C	
cusid_efb_2	5D	
cusid_efb_3	5E	
cusid_efb_4	5F	
cusid_ready_out	60	
security_ready_out	61	
chipid_ready_out	62	
ready_n	63	
rd_n	64	
oe_n	65	
cs0_n	66	
paen	67	
adv_n	68	
rtc_clk	69	
rtcbad	6A	
rtcalarm	6B	
rtc3	6C	
rtc2	6D	
rtc1	6E	
rtc0	6F	

9.7.10.2.8 MCU_PER

Table 123 MCU_PER Signals

Signal Name	Select Signal (Hex)	Remarks
gsm_dsp_int_gp(0)	01	to DSP
gsm_dsp_int_gp(1)	02	to DSP
gsm_mon_pasw (Not used in E-GOLDvoice)	03	GSM PORTS
gsm_mon_iqramp	04	GSM PORTS
gsm_mon_par_clr_int_fsm	05	GSM PORTS
gsm_mon_par_dac_clk_n	06	GSM PORTS
gsm_mon_par_dac(0)	07	GSM PORTS
gsm_mon_par_dac(1)	08	GSM PORTS
gsm_mon_par_dac(2)	09	GSM PORTS
gsm_mon_par_dac(3)	0A	GSM PORTS
gsm_mon_par_dac(4)	0B	GSM PORTS
gsm_mon_par_dac(5)	0C	GSM PORTS
gsm_mon_par_dac(6)	0D	GSM PORTS
gsm_mon_par_dac(7)	0E	GSM PORTS
gsm_mon_par_dac(8)	0F	GSM PORTS
gsm_mon_par_dac(9)	10	GSM PORTS
gsm_mon_par_dac(10)	11	GSM PORTS
gsm_mon_dcpa_datavalid	12	GSM PORTS
gsm_mon_dcpa_data(0)	13	GSM PORTS
gsm_mon_dcpa_data(1)	14	GSM PORTS
gsm_mon_dcpa_data(2)	15	GSM PORTS
gsm_mon_dcpa_data(3)	16	GSM PORTS
gsm_mon_dcpa_data(4)	17	GSM PORTS
gsm_mon_dcpa_data(5)	18	GSM PORTS
gsm_mon_dcpa_data(6)	19	GSM PORTS
gsm_mon_dcpa_data(7)	1A	GSM PORTS
gsm_mon_dcpa_data(8)	1B	GSM PORTS
gsm_mon_dcpa_data(9)	1C	GSM PORTS
gsm_mon_dcpa_data(10)	1D	GSM PORTS
gsm_mon_adctrig	1E	GSM PORTS
gsm_dsp_eqon	1F	GSM PORTS
gsm_dsp_monon	20	GSM PORTS
gsm_dsp_scon	21	GSM PORTS
gsm_dsp_fcon	22	GSM PORTS
gsm_dsp_rxon	23	GSM PORTS
gsm_dsp-ana_txon	24	GSM PORTS
gsm_dsp_codon	25	GSM PORTS
gsm_eco_slpstart	26	GSM PORTS
meas_ana-pcb_clk_meas	27	MEAS PORTS

Table 123 MCU_PER Signals (cont'd)

Signal Name	Select Signal (Hex)	Remarks
meas_ana-pcb_soc	28	MEAS PORTS
ana_meas-pcl_eoc	29	MEAS PORTS
ana_meas-pcb_penirq_n	2A	MEAS PORTS
com_mon(0)	2B	SM MCU Set Comm. Flag, Bit 2
com_mon(1)	2C	SM MCU Reset Comm. Flag, Bit 2
com_mon(2)	2D	SM DSP Set Comm. Flag, Bit 2
com_mon(3)	2E	SM DSP Reset Comm. Flag, Bit 2
com_mon(4)	2F	SM MCU Comm. Flag Status, Bit 2
sem_mon(0)	30	SM DSP Semaphore Register, Bit 4
sem_mon(1)	31	SM MCU Semaphore Register, Bit 4
sem_mon(2)	32	SM DSP Semaphore Read Data, Bit 4
sem_mon(3)	33	SM MCU Semaphore Read Data, Bit 4
biu_mon(0)	34	BIU
biu_mon(1)	35	BIU biu_mon(2)
	36	BIU biu_mon(3)
	37	BIU biu_mon(4)
	38	BIU biu_mon(5)
	39	BIU biu_mon(6)
	3A	BIU biu_mon(7)
	3B	BIU
pcl_sim_ccin	3C	SIM PORTS
sim_pcl_ccrst	3D	SIM PORTS
sim_pcl_ccclk	3E	SIM PORTS
sim_pcl_ccvz_n	3F	SIM PORTS
sim_pcl_cciosw	40	SIM PORTS
ssc_ms_out	41	SSC SIGNALS
ssc_sh_clk	42	SSC SIGNALS
ssc_sl_out	43	SSC SIGNALS
asc0_sel_n	44	ASC0 SIGNALS
asc0_ex_hw_disreq	45	ASC0 SIGNALS
asc1_cgu_sel_n	46	ASC1 SIGNALS
ram_mon(0)	47	SHARED RAM
ram_mon(1)	48	SHARED RAM
ram_mon(2)	49	SHARED RAM
ram_mon(3)	4A	SHARED RAM
ram_mon(4)	4B	SHARED RAM
ram_mon(5)	4C	SHARED RAM
ram_mon(6)	4D	SHARED RAM
ram_mon(7)	4E	SHARED RAM
ram_mon(8)	4F	SHARED RAM
ram_mon(9)	50	SHARED RAM

Table 123 MCU_PER Signals (cont'd)

Signal Name	Select Signal (Hex)	Remarks
ram_mon(10)	51	SHARED RAM
ram_mon(11)	52	SHARED RAM
ram_mon(12)	53	SHARED RAM
ram_mon(13)	54	SHARED RAM
ram_mon(14)	55	SHARED RAM
ram_mon(15)	56	SHARED RAM
ram_mon(16)	57	SHARED RAM
ram_mon(17)	58	SHARED RAM
ram_mon(18)	59	SHARED RAM
ram_mon(19)	5A	SHARED RAM
ram_mon(20)	5B	SHARED RAM
ram_mon(21)	5C	SHARED RAM
ram_mon(22)	5D	SHARED RAM
ram_mon(23)	5E	SHARED RAM
ram_mon(24)	5F	SHARED RAM
ram_mon(25)	60	SHARED RAM
ram_mon(26)	61	SHARED RAM
ram_mon(27)	62	SHARED RAM
ram_mon(28)	63	SHARED RAM
ram_mon(29)	64	SHARED RAM
ram_mon(30)	65	SHARED RAM
ram_mon(31)	66	SHARED RAM
ram_mon(32)	67	SHARED RAM
ram_mon(33)	68	SHARED RAM
ram_mon(34)	69	SHARED RAM
ram_mon(35)	6A	SHARED RAM
rf_str0	6B	GSM PORT
rf_data	6C	GSM PORT
rf_clk	6D	GSM PORT
fsys2	6E	RF PORT
fsys3	6F	RF PORT
vcxo_en	70	SCCU PORT

9.7.10.2.9 P3_PMU

Table 124 P3_PMU Signals

Signal Name	Select Signal (Hex)	Remarks
ledctrl(0)	01	
ledctrl(1)	02	
ledctrl(2)	03	
ledctrl(3)	04	
ledctrl(4)	05	

Table 124 P3_PMU Signals (cont'd)

Signal Name	Select Signal (Hex)	Remarks
ledctrl(5)	06	
ledctrl(6)	07	
ledctrl(7)	08	
bgtrm(0)	09	
bgtrm(1)	0A	
intctrl(0)	0B	
intctrl(1)	0C	
intctrl(2)	0D	
intctrl(3)	0E	
intctrl(4)	0F	
intctrl(5)	10	
intctrl(15)	11	
chgctrl(0)	12	
chgctrl(1)	13	
chgctrl(2)	14	
chgctrl(4)	15	
chgctrl(5)	16	
chgctrl(7)	17	
chgctrl(8)	18	
chgctrl(9)	19	
lpdctrl(0)	1A	
lpdctrl(1)	1B	
lpdctrl(2)	1C	
lpdctrl(3)	1D	
lpdctrl(4)	1E	
lpdctrl(5)	1F	
lpdctrl(6)	20	
lpdctrl(7)	21	
lpdctrl(8)	22	
pwrctrl2(0)	23	
pwrctrl2(1)	24	
pwrctrl2(2)	25	
pwrctrl2(3)	26	
pwrctrl2(4)	27	
pwrctrl2(5)	28	
pwrctrl2(11)	29	
pwrctrl2(12)	2A	
pwrctrl2(13)	2B	
pwrctrl2(14)	2C	
pwrctrl2(15)	2D	
pwrctrl1(0)	2E	

Table 124 P3_PMU Signals (cont'd)

Signal Name	Select Signal (Hex)	Remarks
pwrctrl1(1)	2F	
pwrctrl1(2)	30	
pwrctrl1(3)	31	
pwrctrl1(4)	32	
pwrctrl1(5)	33	
pwrctrl1(6)	34	
pwrctrl1(7)	35	
pwrctrl1(8)	36	
pwrctrl1(9)	37	
pwrctrl1(10)	38	
pwrctrl1(11)	39	
pwrctrl1(12)	3A	
pwrctrl1(13)	3B	
pwrctrl1(14)	3C	
pwrctrl1(15)	3D	
genctrl(0)	3E	
genctrl(14)	3F	
genctrl(15)	40	

9.7.10.2.10 PMU_SUPER

Table 125 PMU_SUPER Signals

MON1			MON2		
sel	**signal**	**description**	**sel**	**signal**	**description**
0x00	bg_timer (1)	hpbg seq timer	0x00	bg_timer (0)	hpbg seq timer
0x01	bg_timer (3)		0x01	bg_timer (2)	
0x02	bg_timer (5)		0x02	bg_timer (4)	
0x03	bg_timer (7)		0x03	bg_timer (6)	
0x04	bg_timer (9)		0x04	bg_timer (8)	
0x05	system_rst_timer (1)	sys reset timer	0x05	system_rst_timer (0)	sys reset timer
0x06	system_rst_timer (3)		0x06	system_rst_timer (2)	
0x07	system_rst_timer (5)		0x07	system_rst_timer (4)	
0x08	system_rst_timer (7)		0x08	system_rst_timer (6)	
0x09	system_rst_timer (9)		0x09	system_rst_timer (8)	
0x0A	main_fsm_state (1)	main fsm state	0x0A	main_fsm_state (0)	main fsm state
0x0B	main_fsm_state(3)		0x0B	main_fsm_state (2)	
0x0C	batmes_vsel(1)	vbat comp voltage selector	0x0C	batmes_vsel(0)	vbat comp voltage selector

Table 125 PMU_SUPER Signals (cont'd)

MON1			MON2		
sel	**signal**	**description**	**sel**	**signal**	**description**
0x0D	hpbg_trim_value (1)	hpbg trim value	0x0D	hpbg_trim_value (0)	hpbg trim value
0x0E	hpbg_trim_value (3)		0x0E	hpbg_trim_value (2)	
0x0F	hpbg_trim_value (5)		0x0F	hpbg_trim_value (4)	
0x10	hpbg_trim_value (7)		0x10	hpbg_trim_value (6)	
0x11	hpbg_trim_value (9)		0x11	hpbg_trim_value (8)	
0x12	cdt_counter(1)	charger removed cntr	0x12	cdt_counter(0)	charger removed cntr
0x13	batmes_hp_not_lp	hpbg/lpbg selector	0x13	cdt_counter(2)	
0x14	it_counter(1)	1 sec long timer cntr	0x14	it_counter(0)	1 sec long timer cntr
0x15	it_counter(3)		0x15	it_counter(2)	
0x16	it_counter(5)		0x16	it_counter(4)	
0x17	it_counter(7)		0x17	it_counter(6)	
0x18	it_counter(9)		0x18	it_counter(8)	
0x19	it_counter(11)		0x19	it_counter(10)	
0x1A	it_counter(13)		0x1A	it_counter(12)	
0x1B	reset_bb_n	baseband reset	0x1B	it_counter(14)	
0x1C	chw_counter(1)	sw chargewave cntr	0x1C	chw_counter(0)	sw chargewave cntr
0x1D	chw_counter(3)		0x1D	chw_counter(2)	
0x1E	chw_counter(5)		0x1E	chw_counter(4)	
0x1F	chw_counter(7)		0x1F	chw_counter(6)	
0x20	su_fcon	first connect startup source indic	0x20	chw_counter(8)	
0x21	ldo_timer(1)	ldo seq timer	0x21	ldo_timer(0)	ldo seq timer
0x22	ldo_timer(3)		0x22	ldo_timer(2)	
0x23	ldo_timer(5)		0x23	ldo_timer(4)	
0x24	ldofsm(1)	ldo startup fsm	0x24	ldofsm(0)	ldo startup fsm
0x25	sw_charge	sw charger clk	0x25	ldofsm(2)	
0x26	bgfsm(1)	hpbg charger fsm	0x26	bgfsm(0)	hpbg charger fsm

Table 125 PMU_SUPER Signals (cont'd)

MON1			MON2		
sel	**signal**	**description**	**sel**	**signal**	**description**
0x27	pch_cnt(1)	precharge 60mn cnter	0x27	pch_cnt(0)	precharge 60mn cnter
0x28	pch_cnt(3)		0x28	pch_cnt(2)	
0x29	pch_cnt(5)		0x29	pch_cnt(4)	
0x2A	pch_cnt(7)		0x2A	pch_cnt(6)	
0x2B	pch_cnt(9)		0x2B	pch_cnt(8)	
0x2C	pch_cnt(11)		0x2C	pch_cnt(10)	
0x2D	pch_cnt(13)		0x2D	pch_cnt(12)	
0x2E	pch_cnt(15)		0x2E	pch_cnt(14)	
0x2F	pch_cnt(17)		0x2F	pch_cnt(16)	
0x30	pch_cnt(19)		0x30	pch_cnt(18)	
0x31	pch_cnt(21)		0x31	pch_cnt(20)	
0x32	pch_cnt(23)		0x32	pch_cnt(22)	
0x33	pch_cnt(25)		0x33	pch_cnt(24)	
0x34	pch_cnt(27)		0x34	pch_cnt(26)	
0x35	vcxo_en	vcxoen	0x35	full_charge	Full charge startup performed
0x36	sw_cnt(1)	sw charge timer	0x36	sw_cnt(0)	sw charge timer
0x37	sw_cnt(3)		0x37	sw_cnt(2)	
0x38	sw_cnt(5)		0x38	sw_cnt(4)	
0x39	sw_cnt(7)		0x39	sw_cnt(6)	
0x3A	sw_cnt(9)		0x3A	sw_cnt(8)	
0x3B	sw_cnt(11)		0x3B	sw_cnt(10)	
0x3C	sw_cnt(13)		0x3C	sw_cnt(12)	
0x3D	sw_cnt(15)		0x3D	sw_cnt(14)	
0x3E	sw_cnt(17)		0x3E	sw_cnt(16)	
0x3F	ld1_en	LD1 enable	0x3F	swct_supervision	SWCT supervision active
0x40	lbuf_vsel(1)	LBUF voltage select	0x40	lbuf_vsel(0)	LBUF voltage select
0x41	lio_en	LIO enable	0x41	lio_en	LIO enable
0x42	lrfxo_en	LRFXO enable	0x42	lrfxo_en	LRFXO enable
0x43	lrfrx_en	LRFRX enable	0x43	lrfrx_en	LRFRX enable
0x44	lrftrx_en	LRFTRX enable	0x44	lrftrx_en	LRFTRX enable
0x45	lmem_en	LMEM enable	0x45	lmem_en	LMEM enable
0x46	lana_en	LANA enable	0x46	lana_en	LANA enable
0x47	lsim_en	LSIM enable	0x47	lsim_en	LSIM enable
0x48	lbuf_en	LBUF enable	0x48	lbuf_en	LBUF enable
0x49	lmem_idle	LMEM stdby	0x49	lmem_idle	LMEM stdby
0x4A	lana_idle	LANA stdby	0x4A	lana_idle	LANA stdby
0x4B	lsim_idle	LSIM stdby	0x4B	lsim_idle	LSIM stdby
0x4C	lio_idle	LIO stdby	0x4C	lio_idle	LIO stdby

Table 125 PMU_SUPER Signals (cont'd)

MON1			MON2		
sel	**signal**	**description**	**sel**	**signal**	**description**
0x4D	ld1_idle	LD1 stdby	0x4D	ld1_idle	LD1 stdby
0x4E	lmem_2v85	LMEM voltage select	0x4E	lmem_2v85	LMEM voltage select
0x4F	lsim_1v8	LSIM voltage select	0x4F	lsim_1v8	LSIM voltage select
0x50	ld1_1v2	LD1 voltage select	0x50	ld1_1v2	LD1 voltage select
0x51	lrtc_en	LRTC enable	0x51	lrtc_en	LRTC enable
0x52	lio_1v8	LIO voltage select	0x52	lio_1v8	LIO voltage select
0x53	ld1_pd	LD1 pull down enable	0x53	ld1_pd	LD1 pull down enable
0x54	lio_pd	LIO pull down enable	0x54	lio_pd	LIO pull down enable
0x55	lrfxo_pd	LRFXO pull down en.	0x55	lrfxo_pd	LRFXO pull down en.
0x56	lrfrx_pd	LRFRX pull down en.	0x56	lrfrx_pd	LRFRX pull down en.
0x57	lrftrx_pd	LRFTRX pull down en.	0x57	lrftrx_pd	LRFTRX pull down en.
0x58	lmem_pd	LMEM pull down en.	0x58	lmem_pd	LMEM pull down en.
0x59	lana_pd	LANA pull down enable	0x59	lana_pd	LANA pull down enable
0x5A	lsim_pd	LSIM pull down en.	0x5A	lsim_pd	LSIM pull down en.
0x5B	lbuf_pd	LBUF pull down en.	0x5B	lbuf_pd	LBUF pull down en.
0x5C	cdt_rise_edge	Rising edge on Charger detected	0x5C	cdt_rise_edge	Rising edge on Charger detected
0x5D	on_rise_edge	Rising edge on ON detected	0x5D	on_rise_edge	Rising edge on ON detected
0x5E	ov_rise_edge	Rising edge on overvoltage event detected	0x5E	ov_rise_edge	Rising edge on overvoltage event detected
0x5F	swct_overflow	SWCT overflow occured	0x5F	swct_overflow	SWCT overflow occured
0x60	ldo_stat	LDO start seq completed	0x60	ldo_stat	LDO start seq completed
0x61	hpbg_stat	hpbg start seq completed	0x61	hpbg_stat	hpbg start seq completed
0x62	dig_rstn	PMU dig reset	0x62	dig_rstn	PMU dig reset
0x63	hw_charge_off	precharge disable	0x63	hw_charge_off	precharge disable
0x64	cdt	charge detect state	0x64	cdt	charge detect state
0x65	ldo_timer_en	LDO seq timer enable	0x65	ldo_timer_en	LDO seq timer enable
0x66	interrupt	internal irq signal (before mux)	0x66	interrupt	internal irq signal (before mux)
0x67	clk_sel	PMU digital clk selector	0x67	clk_sel	PMU digital clk selector
0x68	clk_pd	PMU digital clock power down	0x68	clk_pd	PMU digital clock power down
0x69	ldo_start	LDO start seq enable	0x69	ldo_start	LDO start seq enable
0x6A	hpbg_start	hpbg start seq enable	0x6A	hpbg_start	hpbg start seq enable

Table 125 PMU_SUPER Signals (cont'd)

MON1			MON2		
sel	**signal**	**description**	**sel**	**signal**	**description**
0x6B	ls_en	VLD1 to VRTC levelshifter enable	0x6B	ls_en	VLD1 to VRTC levelshifter enable
0x6C	batmes_en	VBTA comparator enable	0x6C	batmes_en	VBTA comparator enable
0x6D	su_rtcon	RTC alarm startup source indicator	0x6D	su_rtcon	RTC alarm startup source indicator
0x6E	hpbg_pd	hpbg power down	0x6E	hpbg_pd	hpbg power down
0x6F	int_mux	startup to system on interrupt handover	0x6F	int_mux	startup to system on interrupt handover
0x70	system_rst_timer_en	system reset timer en.	0x70	system_rst_timer_en	system reset timer en.
0x71	bg_timer_en	hpbg seq timer enable	0x71	bg_timer_en	hpbg seq timer enable
0x72	pm_int	PMU interrupt line	0x72	pm_int	PMU interrupt line
0x73	on_i	ON pad signal	0x73	on_i	ON pad signal
0x74	mon1	MON1 pad signal	0x74	mon1	MON1 pad signal
0x75	rtcout	RTC alarm signal	0x75	rtcout	RTC alarm signal
0x76	bat_overvoltage_5v5	VBAT 5v5	0x76	bat_overvoltage_5v5	VBAT 5v5
0x77	batmes_comp	VBAT comparator	0x77	batmes_comp	VBAT comparator
0x78	charge_detect	charge detect signal	0x78	charge_detect	charge detect signal
0x79	it_ov	OV for 1sec detected	0x79	it_ov	OV for 1sec detected
0x7A	vbat_2v9	VBAT 2v9 register	0x7A	vbat_2v9	VBAT 2v9 register
0x7B	vbat_3v1	VBAT 3v1 register	0x7B	vbat_3v1	VBAT 3v1 register
0x7C	vbat_3v6	VBAT 3v2 register	0x7C	vbat_3v6	VBAT 3v2 register
0x7D	vbat_4vx7	VBAT 4v47/4v57 register	0x7D	vbat_4vx7	VBAT 4v47/4v57 register
0x7E	chg_risedge	rising edge of CHG bit	0x7E	vcxo_en	vcxo_en
0x7F	su_cdt	CDT startup source indicator	0x7F	ld1_en	LD1 enable

Table 126 Main FSM States

State Coding	State
0000	PMU_LOCK
0001	SYSTEM_OFF
0010	HPBG START
0011	HPBG TRIM
0100	PMU STARTUP
0101	SYSTEM RESET
0110	PROGRAMMED SHUTDOWN
0111	SYSTEM ON
1000	FULL CHARGE

Table 127 LDOs FSM States

State Coding	State
000	LDO OFF
001	LRFXO START
010	LD1 START
011	LMEM START
100	LIO START
101	unused
110	unused
111	LDOs up

Table 128 Bandgap FSM States

State Coding	State
00	BG OFF
01	BG START
10	unused
11	BG UP

9.7.10.2.11 TCU_SUPER

Table 129 TCU_SUPER Signals

Signal Name	Select Signal (Hex)	Remarks
tcu_security_fuse_ready_in_o	01	
tcu_custid_fuse_ready_in_o	02	
tcu_chipid_fuse_ready_in_o	03	
tcu_rf_fuse_ready_in_o	04	
tcu_pmu_fuse_ready_in_o	05	
tcu_security_fuse_read_o	06	
tcu_custid_fuse_read_o	07	
tcu_chipid_fuse_read_o	08	
tcu_rf_fuse_read_o	09	
tcu_pmu_fuse_read_o	0A	
tcu_security_stream_out_en_o	0B	
tcu_custid_stream_out_en_o	0C	
tcu_chipid_stream_out_en_o	0D	
tcu_rf_stream_out_en_o	0E	
tcu_pmu_stream_out_en_o	0F	
tcu_security_fuse_stream_in_en_o	10	
tcu_custid_fuse_stream_in_en_o	11	
tcu_chipid_fuse_stream_in_en_o	12	
tcu_rf_fuse_stream_in_en_o	13	
tcu_pmu_fuse_stream_in_en_o	14	
tcu_custid_fuse_prgm_o	15	
tcu_security_fuse_prgm_o	16	

Table 129 TCU_SUPER Signals (cont'd)

Signal Name	Select Signal (Hex)	Remarks
tcu_chipid_fuse_prgm_o	17	
tcu_rf_fuse_prgm_o	18	
tcu_pmu_fuse_prgm_o	19	
cgu_security_fuse_ready_out_o	1A	
cgu_custid_fuse_ready_out_o	1B	
cgu_chipid_fuse_ready_out_o	1C	
cgu_rf_fuse_ready_out_o	1D	
cgu_pmu_fuse_ready_out_o	1E	
tcu_testmargin_o	1F	
tcu_efuse_vdd_pad_en_o	20	
tcu_mem_fuse_read_o	21	
tcu_mem_stream_out_en_o	22	
tcu_mem_fuse_scan_in_en_o	23	
mem_fuse_prgm	24	
iopath_int_sc		
25 efusetest_int_sc	26	
sigmon_int_sc		
27 dsptest_int_sc		
28 dctest_int_sc	29	
testpll_int_sc	2A	
pmu_t4_int_sc	2B	
pmu_t3_int_sc	2C	
pmu_t2_int_sc	2D	
pmu_t1_int_sc	2E	
rf_pmu_scan_int_sc	2F	
rf_t3_int_sc		
30 rf_t2_int_sc	31	
jm_cconf_cc_s(1)		
32 jm_cconf_cc_s(0)	33	
analog_t3_int_sc	34	
analog_t2_int_sc	35	
analog_t1_int_sc	36	
runbist_int_sc	37	
transio_int_sc	38	
testoak_int_sc	39	
scanfast_int_sc	3A	
scanlong_int_sc	3B	

9.8 TAP Controller and Break Switch

System Integration

- Supply domain: VDD_LD1
- Chip internal interfaces:
  - Break Switch clock domain: Refer to **Section 7.2.1.3 Sub-System Clocks and Enables (on Page 67)** and see **Figure 18 Clock Enable (on Page 68)**.
  - JTAG clock domain: tck from the external port
    cgu_clk_x_o and clk_dsp_o for scan use
  - Bus domain: PD-Bus, JTAG
    PD-Bus interface for break_switch module
  - Interrupt sources: No
- Chip external signals:
  TCK, TMS, TDI, TDO, TRST_n, TRIG_IN, TRIG_OUT
  RF_CLK, RF_DATA and RF_STR[0:3]
- Monitor Pins: Refer to **Section 9.7.10 Internal Signal Monitoring (on Page 435)**.

9.8.1 Introduction

The TCU Test Control Unit includes the Primary Tap controller, the **Break Switch** for multicore debug, and some **JTAG** data registers for test and debug purposes.

9.8.2 Break Switch

The Break Switch is used to perform the multicore debug; it is linked to the **OCEM/SEIB (on Page 512)** and to the **OCDS (on Page 262)**.
The break switch:

Receives break source signals from the OCEM, OCDS, and the external pad

Sends break target signals to the OCEM, OCDS, and the external pad (see **Figure 179**).

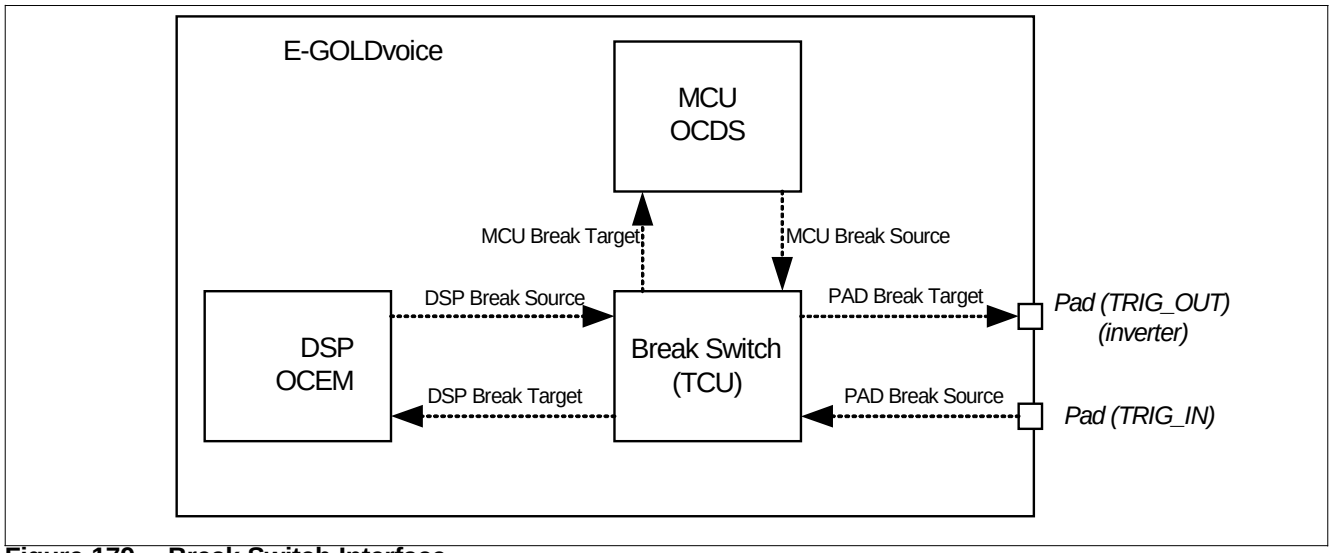


Figure 179 Break Switch Interface

Features

- Break sources: MCU, DSP, external (TRIG IN)
- Break targets: MCU, DSP, external (TRIG OUT)

Depending on the configuration of the register **MCD_BBS**, any active break source signal generates an active break target signal. The break target signal(s) remains active as long as the break condition is true.

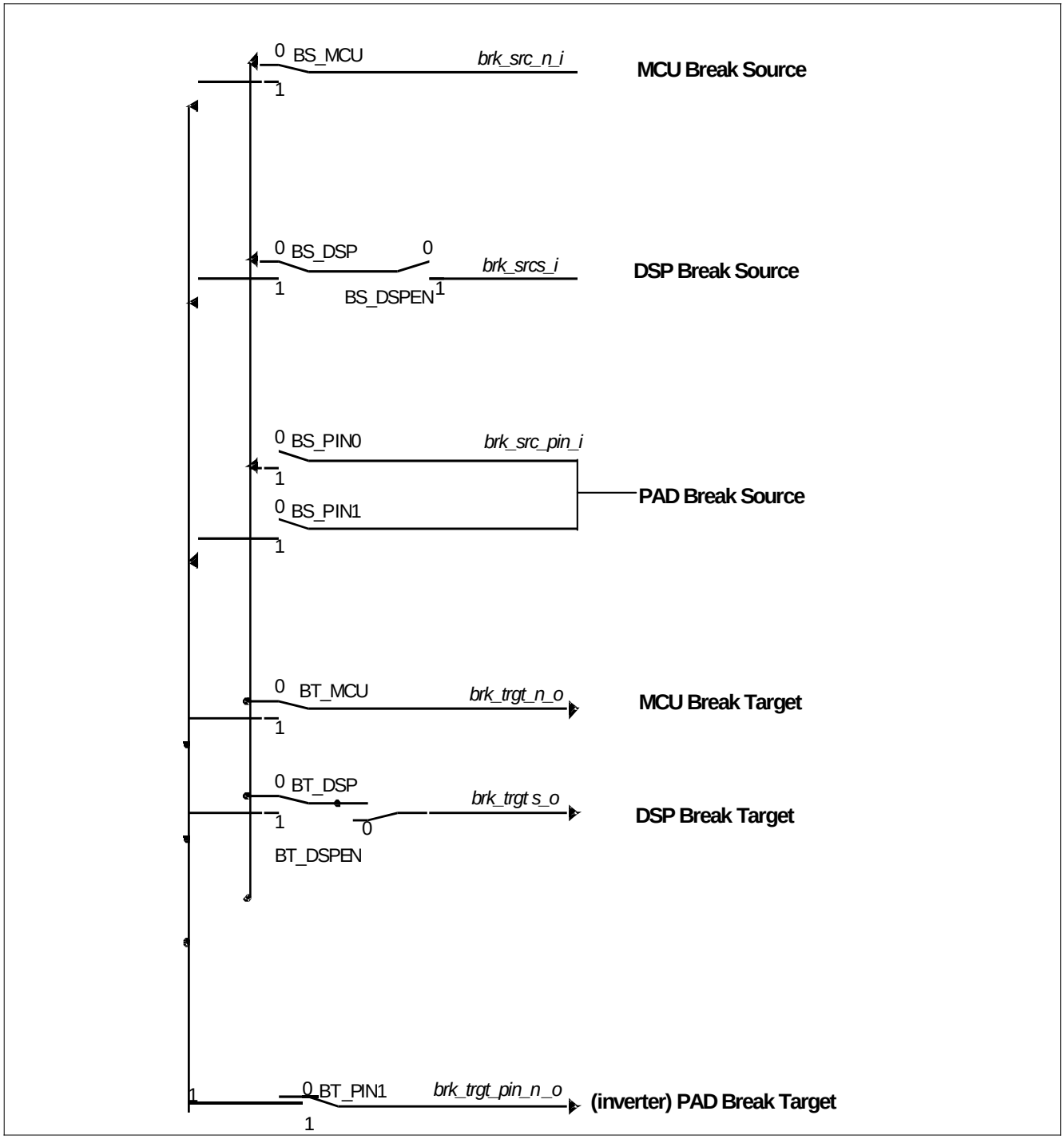


Figure 180 Break Bus Switch

9.8.2.1 Registers

There is one register (**MCD_BBS**) to configure the break switch. It is a control and status register.

MCD_BBS

Break Bus Switch Status and Control **Reset value: $F000_H$**

15	14	13	12	11	10	9	8	7	6	5	4	3
2	1	0										
	BSS1	BSS0	BSS_DSP	BSS_MCU	RESERVED		BT_PIN1	BS_PIN1	BS_PIN0	BT_DSPEN	BT_DSP	BT_MCU	BS_DSPEN	BS_DSP	BS_MCU

Field	Bits	Type	Description
BS_MCU	0	rw	**MCU Break Source Switch (Figure 180)** 0 Break source connected to break bus 0 1 Break source connected to break bus 1
BS_DSP	1	rw	**DSP Break Source Switch (Figure 180)** 0 Break source connected to break bus 0 1 Break source connected to break bus 1
BS_DSPEN	2	rw	DSP Break Source Switch Enable 0 Break source DSP disabled 1 Break source DSP enabled
BT_MCU	3	rw	**MCU Break Target Switch (Figure 180)** 0 Break target connected to break bus 0 1 Break target connected to break bus 1
BT_DSP	4	rw	**DSP Break Target Switch (Figure 180)** 0 Break target connected to break bus 0 1 Break target connected to break bus 1
BT_DSPEN	5	rw	DSP Break Source Target Enable 0 Break target DSP disabled 1 Break target DSP enabled
BS_PIN0	6	rw	**Pad Break Source TRIG_IN Enable for Bus 0** 0 Break source disabled. 1 Break source enabled.
BS_PIN1	7	rw	**Pad Break Source TRIG_IN Enable for Bus 1** 0 Break source disabled. 1 Break source enabled.
BT_PIN1	8	rw	**Pad Break Target TRIG_OUT Enable (for Break Bus 1)** 0 Break target disabled. 1 Break target enabled.
~~BSS_MCU~~	~~12~~	~~rh~~	~~**Status of Break Source MCU**~~
~~BSS_DSP~~	~~13~~	~~rh~~	~~**Status of Break Source DSP**~~
~~BBS0~~	~~14~~	~~rh~~	~~**Status of Break Bus 0**~~
~~BBS1~~	~~15~~	~~rh~~	~~**Status of Break Bus 1**~~
~~RESERVED~~	~~11:9~~	~~r~~	~~Reserved, these bits must be left at their reset values.~~

9.8.2.1.1 Application Examples

Here are three configuration examples:

1. All break sources are connected to bus 1
   All break targets are connected to bus 0.

The break source TRIG_IN and the break target TRIG_OUT are disabled.
The value to be written in **MCD_BBS** register is 0027_H.

2. The MCU break source is connected to bus 1.
   The external break target TRIG_OUT is connected to bus 1.
   The external break source TRIG_IN is connected to bus 0.
   The other break sources and break targets are disabled. The value to be written in **MCD_BBS** register is 0141_H.
3. The MCU break source is connected to bus 1.
   The external break target TRIG_OUT is connected to bus 1.
   DSP break source and external break source TRIG_IN are connected to bus 0.
   MCU break target is connected to bus 0.
   The value to be written in **MCD_BBS** register is 0165_H
   (binary value: xxxx 000**1** 0**110 01**0**1**. All bit positions that enable break sources and break targets are shown in bold).

9.8.3 JTAG

9.8.3.1 Overview

This specification defines the JTAG module and the protocol for a special JTAG mode to be used for On Chip Debug Support (OCDS) purposes, for Boundary Scan, and for Tests.

This specification refers to IEEE JTAG Standard (IEEE Std. 1149, October 21, 1993).

Features

- Based on the IEEE 1149 JTAG standard.
- 8-bit wide JTAG instruction register.
- Supports access to multiple CPUs (JTAG IO clients) on the same chip.
- Optional error protection for all IO mode data transfers.
- Generic memory access functionality.

9.8.3.1.1 Block Diagram

Figure 181 shows a block diagram with all IO mode related signals of the JTAG module.

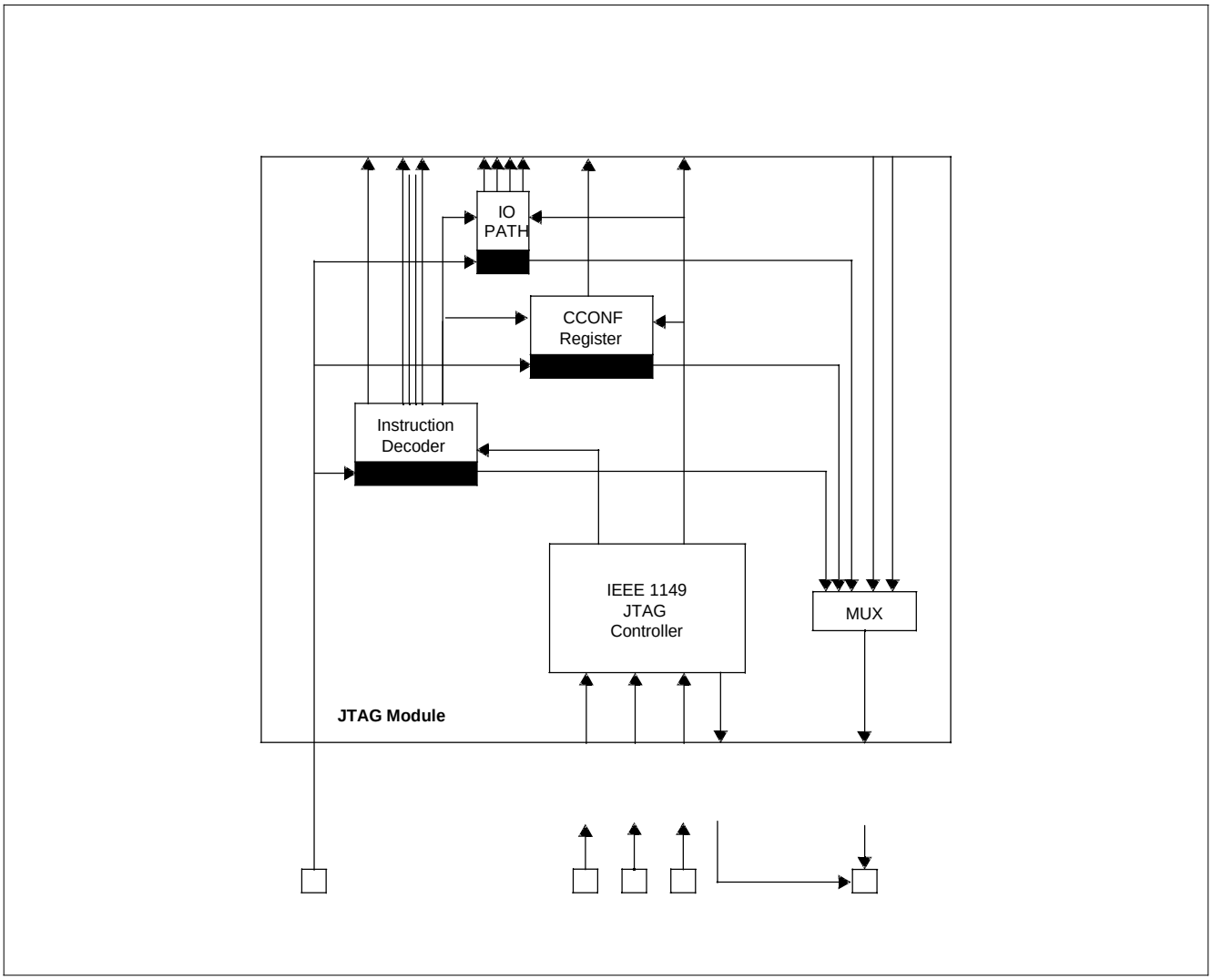


Figure 181 JTAG Module Block Diagram

9.8.3.1.2 TAP Controller

In E-GOLDvoice a Test Access Port (TAP) controller is integrated providing the standard functionality described in the IEEE 1149.1 standard.

a) To control the TAP controller five pins are necessary: TRST_N, TMS, TCK, TDI, TDO. The TAP controller can be reset independently from the chip reset by an external TRST pin. The device works according to the typical tap controller state diagram (see **Figure 182**). State transitions are performed according to the inputs TMS and TCK. Test data on TDI will be loaded with a clock less than or equal to the 4-MHz clock signal on TCK. 1 or 0 on TMS will cause a transition from one controller state to an other; a constant 1 on TMS guarantees normal operation of the chip. If no boundary scan testing is desired, TMS and TDI do not need to be connected since internal pull-up transistors ensure high input levels in this case. A reset forces the TAP controller to the Test Logic Reset state.

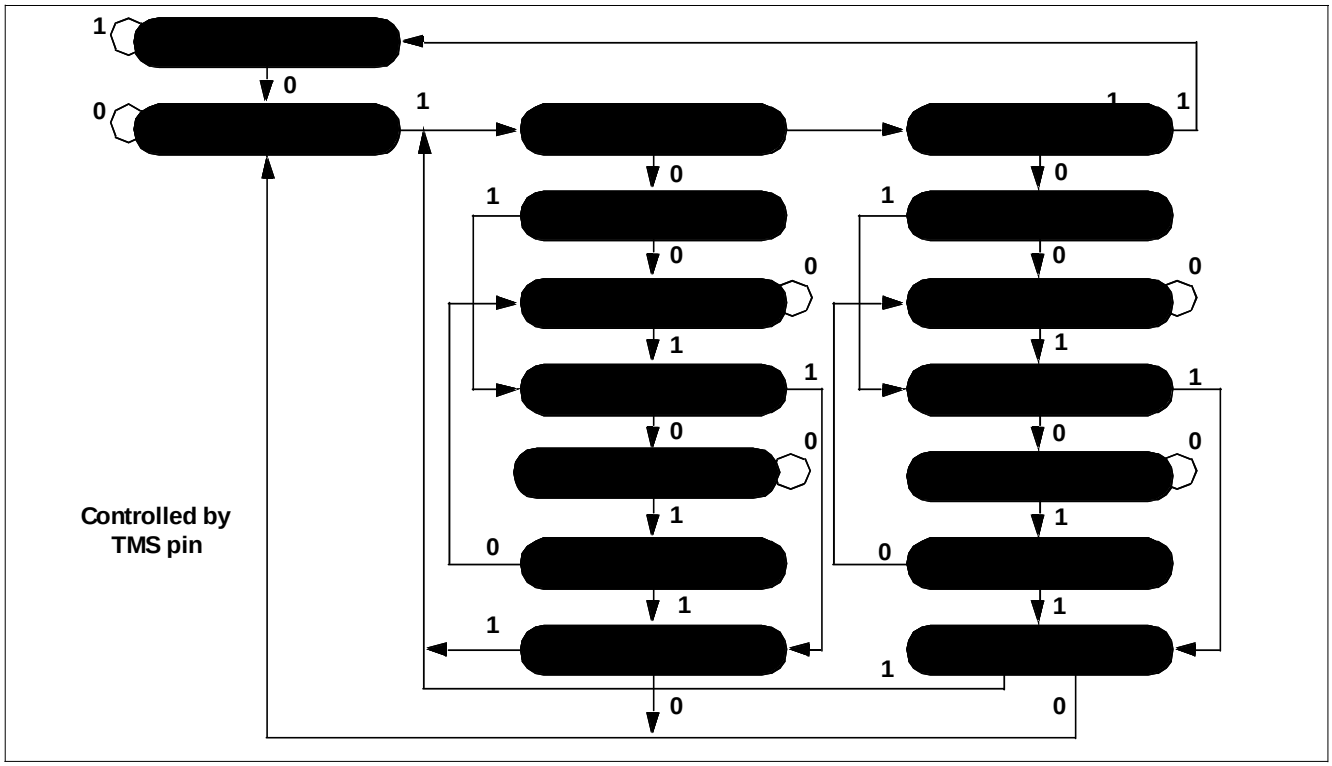


Figure 182 TAP Controller State Diagram

9.8.3.1.3 JTAG Instructions

The instruction register (IR) contains the opcode which are transferred during a JTAG IR-scan. In the following table there is a list of the JTAG instructions.

Table 130 JTAG Instructions

Opcode	Range	Type	Instruction	Select signal
0000 0000	00_H - 07_H 8 instr.	IEEE1149	EXTEST	
0000 0001			INTEST	
0000 0010			SAMPLE/PRELOAD - not implemented	
0000 0011			RUNBIST- not implemented	
0000 0100			IDCODE	
0000 0101			USERCODE - not implemented	
0000 0110			CLAMP - not implemented	
0000 0111			HIGHZ - not implemented	
0000 1000 ...	08H - $0F_H$ 8 instr.	Reserved		
0001 0000	10_H	Chip config.	CCONF_SET	
0001 0001	11_H	Reserved		
0001 0010 ...	12_H - $1F_H$ 14 instr.	Reserved		
0010 0000 ...	20_H - $3F_H$ 32 instr.	Reserved	Reserved	

Table 130 JTAG Instructions (cont'd)

Opcode	Range	Type	Instruction	Select signal
0100 0000 ...	40_H - $5F_H$ 32 instr.	Reserved		
0110 0000 ...	60_H - $7F_H$ 32 instr.	Reserved		
1000 0000 ...	80_H - $9F_H$ 32 instr.	Reserved		
1010 0000 ...	$A0_H$ - BF_H 32 instr.	Reserved		
1100 0000	$C0_H$ $C1_H$ - CF_H	JTAG IO mode	JTAG_IO_SELECT_PATH	jm_sel_ioX[1]
1100 0001 ...			JTAG_IO_INSTRUCTION1[2] ... JTAG_IO_INSTRUCTION15	
1101 0000 ...	$D0_H$ - FE_H 47 instr.	Reserved		
1111 1111	FF_H	IEEE1149	BYPASS	

1) Defined by the content of the **IOPATH Register**

2) Instructions $C1_H$ to CF_H "isolate" the selected client, that is, tdi/tdo will be dedicated to that client (with its own **I/O Instructions**, for example, CERBERUS or OCEM).

9.8.3.1.4 JTAG Control Pins

Table 131 defines how the JTAG signals shall be connected to port pins.

Table 131 JTAG Module Port Pins

Pin	Signal	I/O	Internal Pull Up/Down		Remark
			IEEE1149	Recommended	
TDI	tdi	I	Up	Up	
TMS	tms	I	Up	Up	
TCK	tck	I	-	Down	Keep in defined state.
TDO	tdo	O	-	-	
TRST_N	trstn	I	Up	Down	The pull-down avoids, that an external pull-down is required during normal operation. This reduces cost and power consumption.

The JTAG standard defines an internal pull up for the TRST_N reset pin. In normal operation, when the JTAG interface is not used, this pin has to be connected externally to ground. This means, that the pin permanently draws current. To avoid this, an internal pull down is recommended (Table 131). This has the benefit of reduced power consumption and the JTAG is kept in a defined state by default.

There are no compatibility problems with this non-standard solution. This means it will work with standard JTAG compliant tools, since those actively drive the TRST_N pin. Thus the pull-up/down is irrelevant.

Note: The JTAG standard requires dedicated pins.

9.8.3.1.5 General JTAG Registers

The JTAG module contains the standard **JTAG Instructions** (8 bits) and **BYPASS Register** and the IO mode specific **IOPATH Register**

Table 132 JTAG Module Register Overview

Register	Width	Reset (*trstn*)	Description
BYPASS	1 bit	X	JTAG standard bypass register. If selected (BYPASS instruction) the *tdo* output is equal to *tdi*, delayed by one tck cycle.
CCONF	16 bit	0000_H	Chip Configuration Register.
ID	32 bit	-	Optional JTAG standard chip ID register. The ID is shifted out, when INSTRUCTION contains the IDCODE instruction. This register is not a part of the JTAG module
INSTRUCTION	8 bit	04_H	JTAG standard instruction register. In difference to all other registers it is set with an IR scan. The reset value is the IDCODE instruction.
IOPATH	2 bit	0_H	Selects the IO client.

Note: All JTAG registers are shifted in and out with the LSB first.

BYPASS Register

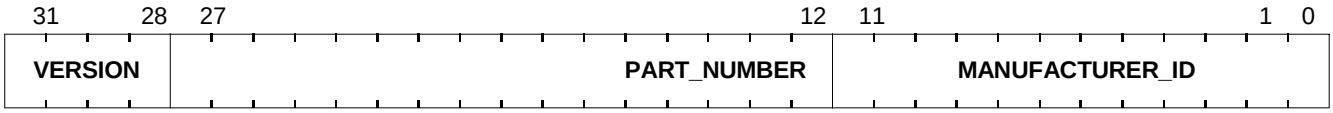
This is a mandatory JTAG register. If selected, the *tdo* output is the *tdi* input delayed by one *tck* cycle.

ID Register

The **ID** register is not part of the JTAG module, its implementation is a product specific decision. This allows to maintain one central version and part number register, which can be accessed across JTAG with the IDCODE instruction. According to the JTAG standard, the IDCODE instruction has to have the following structure:

ID

JTAG ID Register **Reset value: 1013 3083$_H$**



Field	Bits	Type	Description
VERSION	31:28	r	Version of the chip.
PART_NUMBER	27:12	r	Part number of the chip.
MANUFACTURER_ID	11:0	r	Infineon's Manufacturer ID. The value is based on the JEDEC standard (JEP-106-G) manufacturer ID and the IEEE JTAG Standard.

IOPATH Register

The **IOPATH** register is a modified JTAG scan register to provide error protection. For IOPATH *tdo* represents not the output of the **IOPATH** register but the input
(**Figure 183**). This allows to detect transmission bit errors.

The *tdi/tdo* behavior is exactly like for a JTAG BYPASS instruction except, that the first bit output (state Capture-DR) is 1 and not 0. This difference is important in the case, that there was a bit error when the JTAG instruction was shifted in. In the most probable case, that this faulty JTAG instruction is not implemented, the JTAG module would set the BYPASS mode which could otherwise not be distinguished from the JTAG_IO_SELECT_PATH instruction.

The **IOPATH** register is used to select the IO client (**Figure 185**). If the JTAG instruction is in the IO address range and not C0$_H$ (**Table 130**) the associated select signal is active. **IOPATH** register is 2 bits wide and set like a regular JTAG scan chain register with the JTAG_IO_SELECT_PATH instruction.

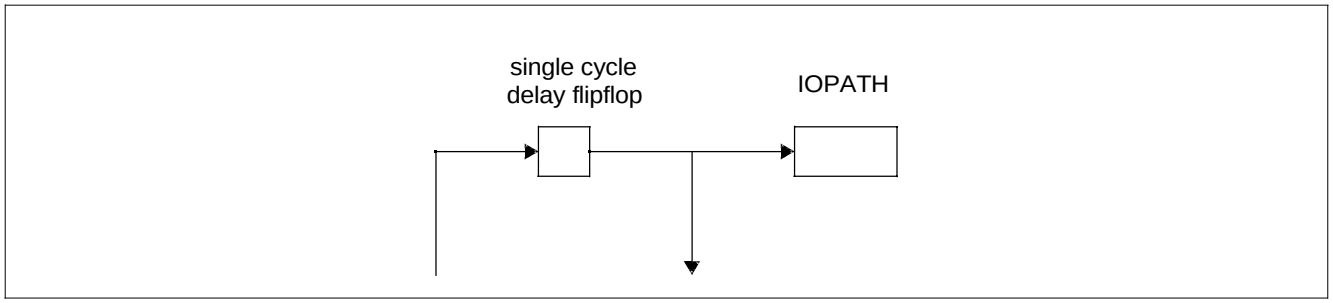


Figure 183 IOPATH Register

Table 133 IOPATH Register

Value (Binary)	Select Signal	Assignment Recommendation
00	jm_sel_io0	Reserved.
01	jm_sel_io1	Teaklite DSP (SEIB)
10	jm_sel_io2	C166S MCU (CERBERUS)
11	jm_sel_io3	Reserved

CCONF Register

The **CCONF** register is provided to configure special chip states. It can be considered as an alternate mechanism to reset configurations. All configuration bits have associated protection bits. This allows a very straightforward access by different tools to their dedicated bits, sharing the JTAG interface.

The **CCONF** register is set with the CCONF_SET JTAG instruction, refer to **Table 130**, with the same behavior as the **IOPATH Register** (see **Figure 183**).

Two bits (**RST_HLT**, **NET_ENABLE**) have dedicated meanings, all others are reserved.

CCONF

Chip Configuration Register **Reset value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED	RESERVED	RESERVED	RESERVED	NET EN.P	NET EN.	RST HLTP	RST HLT

Field	Bits	Type	Description
RST_HLT	0	w	**Halt After Reset** 0 No effect 1 Halt mode after reset
RST_HLT_P	1	w	0 Bit protection: **CCONF.RST_HLT** unchanged 1 **RST_HLT** will be changed
NET_ENABLE	2	w	**Enable NET Interface** 0 Disabled 1 Enabled
NET_ENABLE_P	3	w	0 Bit protection: **CCONF.NET_ENABLE** unchanged 1 **NET_ENABLE** will be changed
RESERVED	15:4	r	Reserved, these bits must be left at their reset values.

DEBUG_DSP

Debug control for DSP **Reset value: 000_H**

10	9	8	7	6	5	4	3	2	1	0
RESERVED			START	RESERVED	DSP_OCDS_EN	RESERVED	OCEM_ABORT	RESERVED	DSP_RST_JTAG	

Field	Bits	Type	Description
DSP_RST_JTAG	0	rw	**Reset DSP** This bit is OR'ed with the system reset and goes to the DSP.
OCEM_ABORT	2	rw	**Send DSP to Monitor Program (start debug)** This bit is OR'ed with brk_trgts_o (from TCU) and goes to POCEM.oabort.
DSP_OCDS_EN	4	rw	Activates **DSP_DEBUG.OCEM** bit; active high
START	6	rw	**Restart DSP Synchronously from Break** 0 Do not restart 1 Restart
RESERVED	1, 3, 5, 10:7	r	Reserved, these bits must be left at their reset values.

9.8.3.2 Core Debug (JTAG I/O)

An important functionality of JTAG is the possibility of access to DSP and MCU module for debugging purpose. The JTAG block is connected to the **OCDS (on Page 262)** C166 Cerberus and to the **OCEM/SEIB (on Page 512)** TEAKlite.

.

The selection between the TEAKlite OCEM (Named OCEM IO-client) and the C166 Cerberus (Named Cerberus IO-client) is done with the **IOPATH Register**.

The selected IO-client is addressed by the external JTAG port. There is only one IO-client selected.

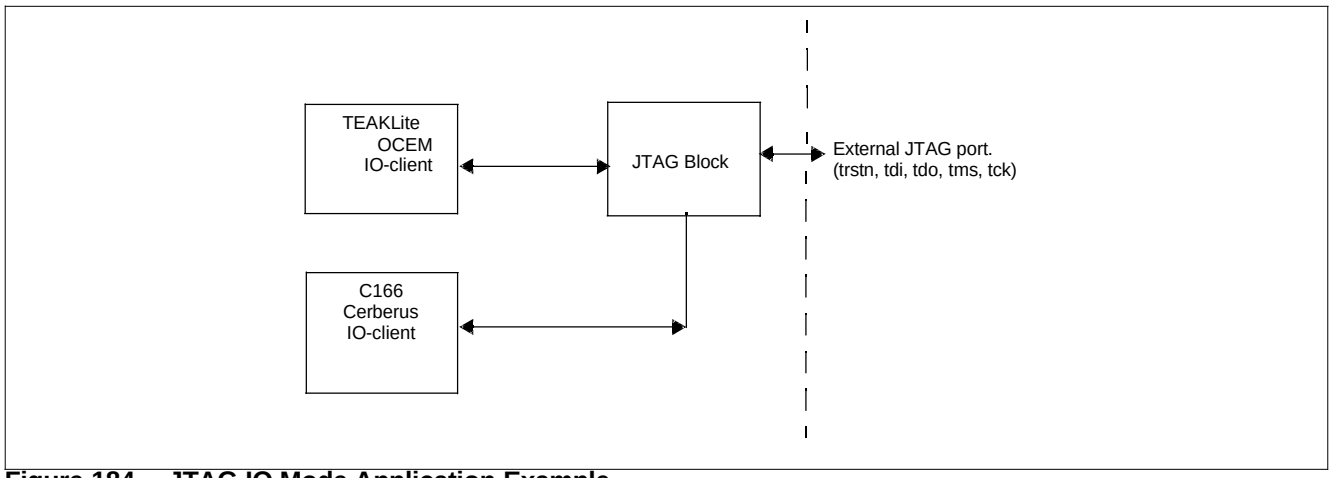


Figure 184 JTAG IO Mode Application Example

The IO mode is implemented in two parts (**Figure 185**):

- The extension of the JTAG module: IO mode JTAG instructions and IOPATH register to select the IO module. The extension is described in **Block Diagram**.
- The IO client modules (one or multiple). These IO clients are the generic IO mode part of, for example, the OCDS module in **Figure 184**.The selected client (IOPATH) is controlled with default chip internal JTAG signals by the JTAG module. Depending on the JTAG instruction, the configuration register IOCONF or the address register for the access IOADDR can be written, or the data register IODATA can be read or written.

Figure 185 shows only the JTAG IO mode related signals.

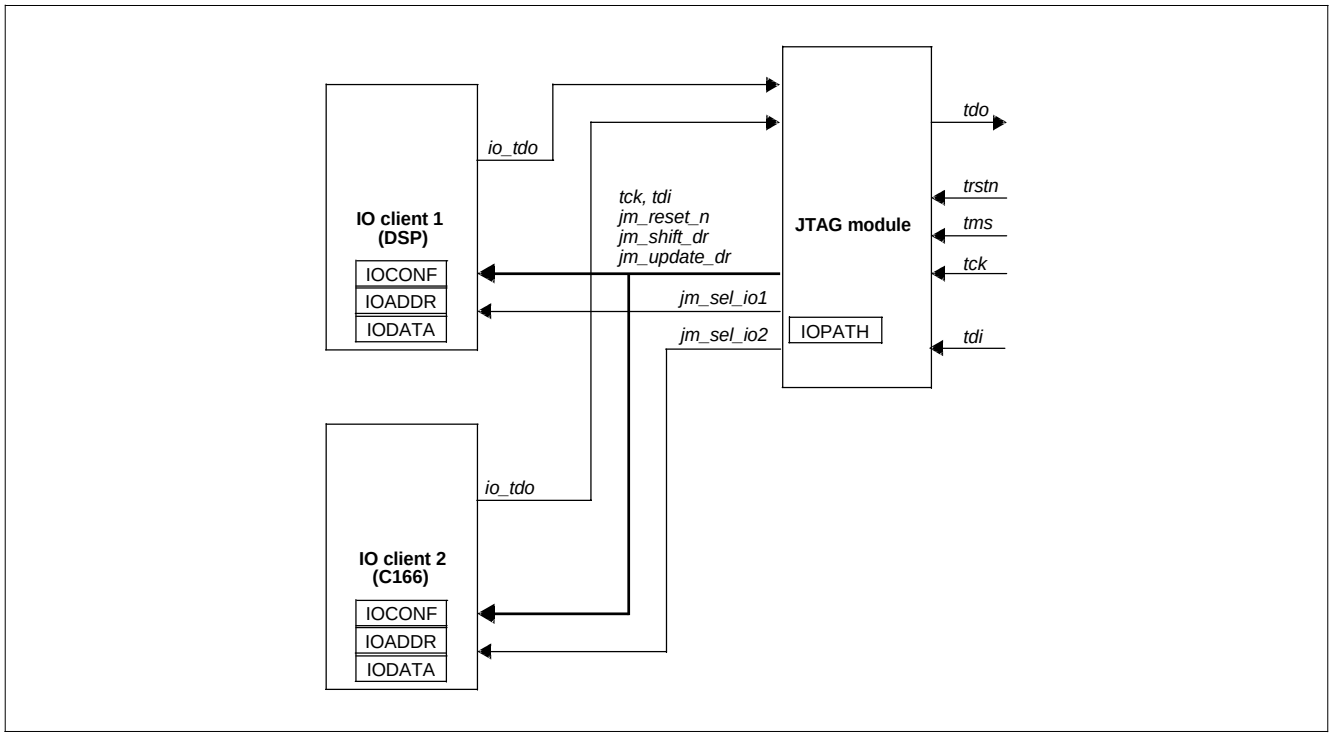


Figure 185 IO Mode Basic Architecture

Notes

1. *IO client 2 is now CERBERUS port C166 core.*
2. *You will see further that IO client 1 (DSP) is implemented into TCU block (TCU test mode block) whereas IO client 2 is inside C166 (CERBERUS port)*

9.8.3.2.1 Application Example for C166 Cerberus Debug

In order to connect the C166 Cerberus directly to the external JTAG port, do the following procedure:

1. Load the **JTAG IO mode** instruction: $C0_H$
2. Load the corresponding **IOPATH Register** with the data 10_B.
3. Load the **JTAG IO mode** instruction: $C1_H$ (to lead Cerberus tdo visible at external JTAG tdo port)
4. Deal with C166 Cerberus **I/O Instructions (on Page 278)** - IO_CONFIG to IO_CLIENT_ID

Figure 186 shows IO-client selection and setting.

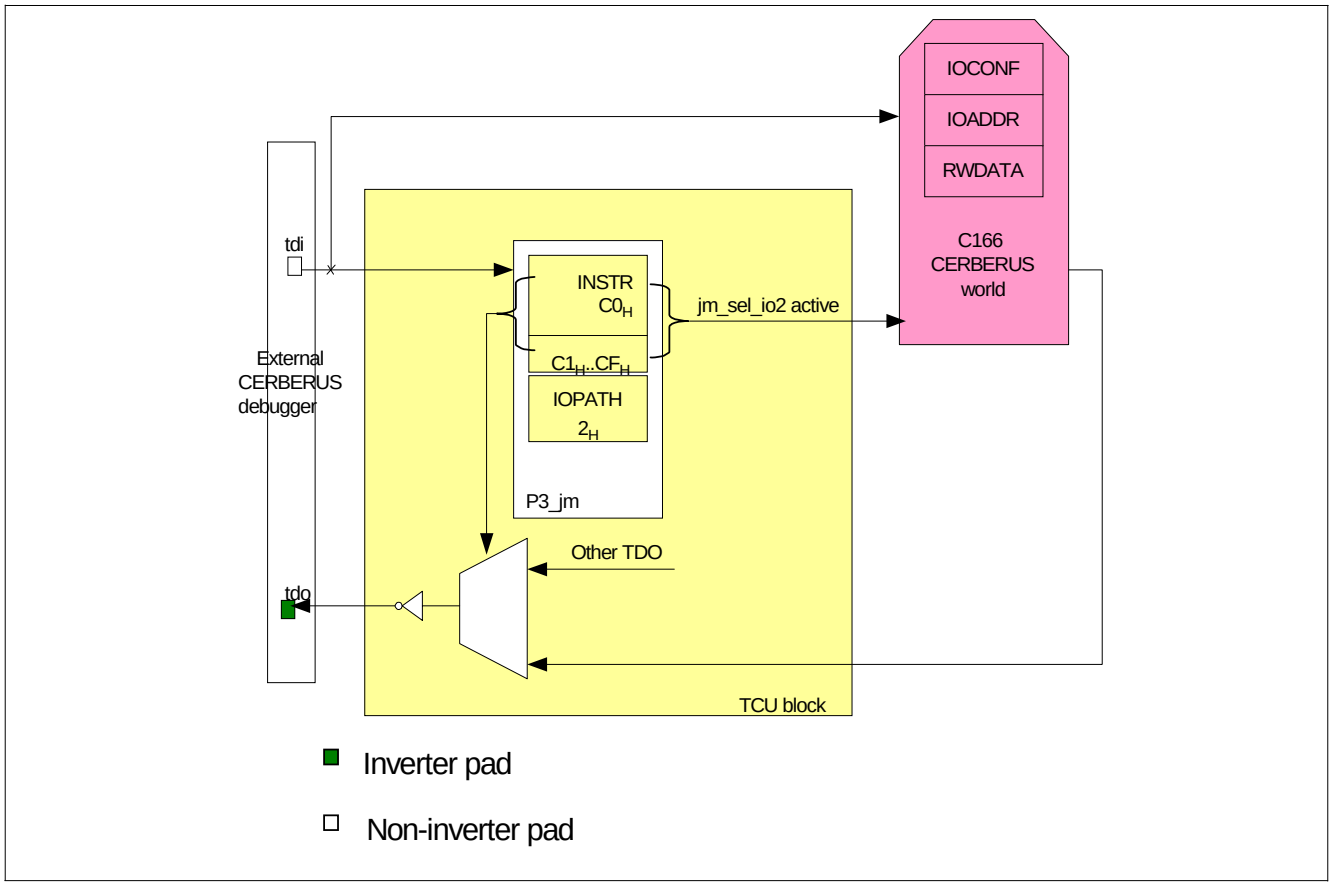


Figure 186 C166 CERBERUS IO-Client Selection

9.8.3.2.2 Application Example for the TEAKLITE debug

To connect the TEAKLITE OCEM directly to the external JTAG port, do the following:

1. Load instruction 21_H into **DEBUG_DSP** register to set **DEBUG_DSP.dsp_ocds_en** and **DEBUG_DSP.START** bits
2. Load the **JTAG IO mode** instruction: $C0_H$
3. Load the corresponding **IOPATH Register** with data **01**.
4. Load the **JTAG IO mode** instruction: $C1_H$
5. Load instruction 21_H into **DEBUG_DSP** register to set **DEBUG_DSP.ocem_abort** (DSP goes into monitor program)
6. Leave the trap by loading instruction 21_H into **DEBUG_DSP** register to set **START** bit.

9.8.3.2.3 Multi Core Debug Session

There are two debug session choices:

- MCU
- MCU + DSP

Debug Session
MCU

To enter MCU into debug:

- HW means => "***jm_sel_io2***"signal (= **IOPATH Register** value) enables MCU OCDS.

A wait loop is defined into the monitor polling any debugger command. A low level of**PSW.USR0** makes the MCU leave the monitor.

Start a MCU debug session

1. Plug JTAG debugger
2. Power on the chip
3. Give a debug system reset via JTAG trstn
4. Deactivate DSP debug system by resetting **DEBUG_DSP.dsp_ocds_en** and setting **DEBUG_DSP.START** *(*Data associated with JTAG instruction 21_H*)*
5. Activate MCU debug system by loading JTAG instruction $C0_H$ (communication mode) with its associated DATA_DR for C166 IO-client (2_H)

MCU is now in monitor program waiting for any debugger command, polling for **PSW.USR0** bit activity.

Interrupt the MCU debug session

Assume the debugger is "idle" waiting for a command (breakpoint...), MCU is in its monitor program => **PSW.USR0** is set.

1. Load JTAG instruction $C1_H$ to enter Cerberus debug commands (**I/O Instructions (on Page 278)**)
2. Execute the debug command (debugger is running) => **PSW.USR0** is reset. As a consequence, MCU leaves the monitor program to process the command. When debugger stops into the breakpoint, MCU has returned to monitor program => **PSW.USR0** is set. Into the debugger, the user can access different registers and carry on with an other debugger commands.

MCU + DSP

To enable the DSP OCEM block, refer to **Figure 224 OCEM Enabling Block (on Page 525)**: "***jm_sel_io1***"signal (= **IOPATH Register** value) and **DEBUG_DSP.dsp_ocds_en** bit (both high active) allows the DSP debug.

When system reset is released, the DSP boot ROM scans for **STATUS1**.**DBG** bit (OCEM register) and for **STATUS1**.**DBG** bit to know if DSP must enter monitor program or not.

Start a MCU + DSP Debug Session

1. Plug JTAG debugger
2. Power on the chip
3. Give a debug system reset via JTAG trstn
4. Activate DSP debug system by setting **DEBUG_DSP.dsp_ocds_en** and resetting **DEBUG_DSP.START** (Data associated with JTAG instruction 21_H) and by loading JTAG instruction $C0_H$ (communication mode) with associated **IOPATH Register** for Teaklite IO-client (1_H)
5. Activate MCU debug system by loading JTAG instruction $C0_H$ (communication mode) with its associated **IOPATH Register** for C166 IO-client (2_H)
6. Give a system reset via RESET_N
7. Load JTAG instruction 21_H
8. Set **DEBUG_DSP.DSP_OCDS_EN** to 1
9. Perform a DSP RST by loading JTAG instruction 21_H to set **DSP_RST_JTAG** to 1 => Monitor program is entered as **STATUS1**.**DBG** bit and **STATUS1**.**DBG** are both active (mailbox memory buffer will be used to dump all internal/OCEM registers)
10. Load instruction $C1_H$ => mailbox will poll for any debugger command

In a DSP Debug Session

There are two HW interrupts sources and one internal interrupt source for DSP core:

1. **DEBUG_DSP.ocem_abort** *activated (= 1)* as a consequence of JTAG instruction
2. *dsp_ocem_breakinp_i* signal activated in MCD
3. An address/data match (refer to **Section 8.11 OCEM/SEIB (on Page 512)**)

In any cases, monitor program is ended as soon as **DEBUG_DSP.START** is active (= 1) => OCEM/internal registers are reloaded from the mailbox.

Start a MCU debug session

1. Power on the chip
2. Plug JTAG debugger
3. Perform a HW JTAG trstn
4. Load JTAG instruction $C0_H$ (communication mode)
5. Load **IOPATH Register** with C166 IO-client (2_H)

10 Register Lists and Mapping

10.1 PD-Bus Register Addresses

10.1.1 Register Addresses

The following list of SFRs and extended SFRs (ESFR) shows PD-Bus SFRs and their addresses and differentiates between bitaddressable SFR/ESFRs and non-bitaddressable SFR/ESFRs.

10.1.1.1 SFR Description

Table 134 Bit and Non-Bit Addressable SFR Areas

Bitaddressable SFR Area				Non Bitaddressable SFR Area			
Long Address (Hex)	**Short Address (Hex)**	**Register Name**	**Where**	**Long Address (Hex)**	**Short Address (Hex)**	**Register Name**	**Where**
none	FF	Reserved	core				
none	FE	Reserved	core				
none	FD	Reserved	core				
none	FC	Reserved	core				
none	FB	Reserved	core				
none	FA	Reserved	core				
none	F9	Reserved	core				
none	F8	Reserved	core				
none	F7	Reserved	core				
none	F6	Reserved	core				
none	F5	Reserved	core				
none	F4	Reserved	core				
none	F3	Reserved	core				
none	F2	Reserved	core				
none	F1	Reserved	core				
none	F0	Reserved	core				
FFFE	none	Reserved		FEFE	7F	Reserved	Core
FFFC	none	Reserved		FEFC	7E	Reserved	Core
FFFA	none	Reserved		FEFA	7D	Reserved	Core
FFF8	none	Reserved		FEF8	7C	Reserved	Core
FFF6	none	Reserved		FEF6	7B	Reserved	Core
FFF4	none	Reserved		FEF4	7A	Reserved	Core
FFF2	none	Reserved		FEF2	79	Reserved	Core
FFF0	none	Reserved		FEF0	78	Reserved	Core
FFEE	none	Reserved		FEEE	77	Reserved	Core
FFEC	none	Reserved		FEEC	76	Reserved	Core
FFEA	none	Reserved		FEEA	75	Reserved	Core
FFE8	none	Reserved		FEE8	74	Reserved	Core
FFE6	none	Reserved		FEE6	73	Reserved	Core

Table 134 Bit and Non-Bit Addressable SFR Areas (cont'd)

Bitaddressable SFR Area				Non Bitaddressable SFR Area			
Long Address (Hex)	**Short Address (Hex)**	**Register Name**	**Where**	**Long Address (Hex)**	**Short Address (Hex)**	**Register Name**	**Where**
FFE4	none	Reserved		FEE4	72	Reserved	Core
FFE2	none	Reserved		FEE2	71	Reserved	Core
FFE0	none	Reserved		FEE0	70	Reserved	Core
FFDE	EF	**RTB_HI**	IIC	FEDE	6F	Reserved	Core
FFDC	EE	**RTB_LO**	IIC	FEDC	6E	Reserved	Core
FFDA	ED	**IIC_CON**	IIC	FEDA	6D	Reserved	Core
FFD8	EC	**IIC_CFG**	IIC	FED8	6C	Reserved	Core
FFD6	EB	**IIC_ADR**	IIC	FED6	6B	Reserved	Core
FFD4	EA	**IIC_ST**	IIC	FED4	6A	Reserved	Core
FFD2	E9	Reserved	IIC	FED2	69	Reserved	Core
FFD0	E8	**RTC_ISNC**	RTC	FED0	68	Reserved	Core
FFCE	E7	**CC2OUT**	CC2	FECE	67	Reserved	Core
FFCC	E6	**CC2DRM**	CC2	FECC	66	Reserved	Core
FFCA	E5	**CCM5**	CC2	FECA	65	Reserved	Core
FFC8	E4	**CCM4**	CC2	FEC8	64	Reserved	Core
FFC6	E3	**T78CON**	CC2	FEC6	63	Reserved	Core
FFC4	E2	**CC1OUT**	CC1	FEC4	62	Reserved	Core
FFC2	E1	**CC1DRM**	CC1	FEC2	61	Reserved	Core
FFC0	E0	**CCM1**	CC1	FEC0	60	Reserved	Core
FFBE	DF	**CCM0**	CC1	FEBE	5F	Reserved	Core
FFBC	DE	**T01CON**	CC1	FEBC	5E	Reserved	Core
FFBA	DD	Reserved	Core	FEBA	5D	Reserved	Core
FFB8	DC	**T6CON**	GPT12	FEB8	5C	Reserved	Core
FFB6	DB	**T5CON**	GPT12	FEB6	5B	Reserved	Core
FFB4	DA	**T4CON**	GPT12	FEB4	5A	Reserved	Core
FFB2	D9	**T3CON**	GPT12	FEB2	59	Reserved	Core
FFB0	D8	**T2CON**	GPT12	FEB0	58	Reserved	Core
FFAE	D7	Reserved	Core	FEAE	57	Reserved	Core
FFAC	D6	Reserved	Core	FEAC	56		
FFAA	D5	Reserved	Core	FEAA	55		
FFA8	D4	Reserved	Core	FEA8	54		
FFA6	D3	PCL_15(**PCL_<pad>**)	PCL	FEA6	53		
FFA4	D2			FEA4	52		
FFA2	D1	**PMC0**	PMCU behind PCL BPI	FEA2	51		
FFA0	D0	**MCU_DSP_INT_ACK**	intr_ext	FEA0	50		
FF9E	CF	T1IC	Int	FE9E	4F		
FF9C	CE	T0IC	Int	FE9C	4E		

Table 134 Bit and Non-Bit Addressable SFR Areas (cont'd)

Bitaddressable SFR Area				Non Bitaddressable SFR Area			
Long Address (Hex)	**Short Address (Hex)**	**Register Name**	**Where**	**Long Address (Hex)**	**Short Address (Hex)**	**Register Name**	**Where**
FF9A	CD	CRIC	Int	FE9A	4D		
FF98	CC	T8IC	Int	FE98	4C		
FF96	CB	CC23IC	Int	FE96	4B		
FF94	CA	CC22IC	Int	FE94	4A		
FF92	C9	CC21IC	Int	FE92	49		
FF90	C8	CC20IC	Int	FE90	48		
FF8E	C7	CC19IC	Int	FE8E	47		
FF8C	C6	CC18IC	Int	FE8C	46		
FF8A	C5	CC17IC	Int	FE8A	45		
FF88	C4	CC16IC	Int	FE88	44		
FF86	C3	CC7IC	Int	FE86	43		
FF84	C2	CC6IC	Int	FE84	42		
FF82	C1	CC5IC	Int	FE82	41	Reserved	
FF80	C0	CC4IC	Int	FE80	40	Reserved	
FF7E	BF	CC3IC	Int	FE7E	3F	Reserved	
FF7C	BE	CC2IC	Int	FE7C	3E	Reserved	
FF7A	BD	CC1IC	Int	FE7A	3D	Reserved	
FF78	BC	CC0IC	Int	FE78	3C	Reserved	
FF76	BB	S0TB_INT	Int	FE76	3B	Reserved	
FF74	BA	S0E_INT	Int	FE74	3A	Reserved	
FF72	B9	S0R_INT	Int	FE72	39	Reserved	
FF70	B8	S0T_INT	Int	FE70	38	Reserved	
FF6E	B7	RTC_T14_INT	Int	FE6E	37	Reserved	
FF6C	B6	RTC_INT	Int	FE6C	36	Reserved	
FF6A	B5	T7_INT	Int	FE6A	35	Reserved	
FF68	B4	T6_INT	Int	FE68	34	Reserved	
FF66	B3	T5_INT	Int	FE66	33	Reserved	
FF64	B2	T4IC	Int	FE64	32	**IIC_PISEL**	IIC
FF62	B1	T3IC	Int	FE62	31	reserved	IIC
FF60	B0	T2IC	Int	FE60	30	**IIC_ID**	IIC
FF5E	AF	Reserved				FE5E	2F
FF5C	AE	reserved					
FE5C	2E FF5A	AD	Reserved				
ASC0	FE5A	2D FF58	AC	Reserved			
ASC0	FE58	2C FF56	AB	Reserved			
ASC0	FE56	2B FF54	AA	**S0CON**			
	ASC0	FE54	2A				
FF52	A9	**S0PISEL**	ASC0	FE52	29	**RTC_ALARM_HI**	RTC
FF50	A8	**SSCPD_FSTAT**	SSC	FE50	28	**RTC_ALARM_LO**	RTC

Table 134 Bit and Non-Bit Addressable SFR Areas (cont'd)

Bitaddressable SFR Area				Non Bitaddressable SFR Area			
Long Address (Hex)	**Short Address (Hex)**	**Register Name**	**Where**	**Long Address (Hex)**	**Short Address (Hex)**	**Register Name**	**Where**
FF4E	A7	**SSCPD_TXFCON**	SSC	FE4E	27	Reserved	RTC
FF4C	A6	**SSCPD_RXFCON**	SSC	FE4C	26	**RTC_CTRL**	RTC
FF4A	A5	**SSCPD_CON**	SSC	FE4A	25	Reserved	RTC
FF48	A4	**MON_CR2**	PCL	FE48	24	Reserved	RTC
FF46	A3	**MON_CR1**	PCL	FE46	23	Reserved	RTC
FF44	A2	Reserved	PCL	FE44	22	Reserved	RTC
FF42	A1	Reserved	PCL	FE42	21	**RTC_REL_HI**	RTC
FF40	A0	Reserved	PCL	FE40	20	**RTC_REL_LO**	RTC
FF3E	9F	Reserved	PCL	FE3E	1F	**RTC_CNT_HI**	RTC
FF3C	9E	Reserved	PCL	FE3C	1E	**RTC_CNT_LO**	RTC
FF3A	9D	Reserved	PCL	FE3A	1D	**RTC_T14_CNT**	RTC
FF38	9C	Reserved	PCL	FE38	1C	**RTC_T14_REL**	RTC
FF36	9B	Reserved	PCL	FE36	1B	Reserved	RTC
FF34	9A	Reserved	PCL	FE34	1A	**RTC_CON**	RTC
FF32	99	Reserved	PCL	FE32	19	Reserved	RTC
FF30	98	Reserved	PCL	FE30	18	Reserved	RTC
FF2E	97	Reserved	PCL	FE2E	17		
FF2C	96	Reserved	PCL	FE2C	16		
FF2A	95	Reserved	PCL	FE2A	15		
FF28	94	Reserved	PCL	FE28	14		
FF26	93	Reserved	PCL	FE26	13		
FF24	92	**EBU_PDC**	PCL	FE24	12		
FF22	91	PCL_40(**PCL_<pad>**)	PCL	FE22	11		
FF20	90	PCL_39(**PCL_<pad>**)	PCL	FE20	10		
FF1E	8F	Reserved	Core	FE1E	0F	Reserved	
EBU FF1C	8E	Reserved	Core	FE1C	0E	**ADDRSEL3**	EBU
FF1A	8D	Reserved		EBU	FE1A	0D	
	ADDRSEL2		EBU FF18	8C	**BUSCON3**		EBU
	FE18	0C	**ADDRSEL1**	EBU FF16	8B		
	BUSCON2	EBU	FE16	0B	Reserved	Core FF14	8A
	BUSCON1	EBU	FE14	0A	Reserved	Core FF12	89
	Reserved	Core	FE12	09	Reserved	Core FF10	88
	Reserved	Core	FE10	08	Reserved	Core FF0E	87
	Reserved	Core	FE0E	07	Reserved	Core FF0C	86
	BUSCON0	EBU	FE0C	06	Reserved	Core FF0A	85
	Reserved		FE0A	05	Reserved	Core FF08	84
	Reserved		FE08	04	Reserved	Core FF06	83
	Reserved	Core	FE06	03	Reserved	Core FF04	82
	Reserved	Core	FE04	02	Reserved	Core	

Table 134 Bit and Non-Bit Addressable SFR Areas (cont'd)

Bitaddressable SFR Area				Non Bitaddressable SFR Area			
Long Address (Hex)	**Short Address (Hex)**	**Register Name**	**Where**	**Long Address (Hex)**	**Short Address (Hex)**	**Register Name**	**Where**
FF02	81	Reserved	Core	FE02	01	Reserved	Core
FF00	80	Reserved	Core	FE00	00	Reserved	Core

10.1.1.2 ESFR Description

Table 135 Bit and Non-Bit Addressable ESFR Areas

Bitaddressable ESFR Area				Non-Bitaddressable ESFR Area			
Long Address (Hex)	**Short Address (Hex)**	**Register Name**		**Long Address (Hex)**	**Short Address (Hex)**	**Register Name**	
none	FF	R15					
none	FE	R14					
none	FD	R13					
none	FC	R12					
none	FB	R11					
none	FA	R10					
none	F9	R9					
none	F8	R8					
none	F7	R7					
none	F6	R6					
none	F5	R5					
none	F4	R4					
none	F3	R3					
none	F2	R2					
none	F1	R1					
none	F0	R0					
F1FE	none			F0FE	7F	Reserved	
F1FC	none			F0FC	7E	Reserved	Core
F1FA	none			F0FA	7D	Reserved	Core
F1F8	none			F0F8	7C	Reserved	Core
F1F6	none			F0F6	7B	Reserved	
F1F4	none			F0F4	7A	Reserved	Core
F1F2	none			F0F2	79	Reserved	Core
F1F0	none			F0F0	78	Reserved	Core
F1EE	none			F0EE	77	Reserved	Core
F1EC	none			F0EC	76	Reserved	Core
F1EA	none			F0EA	75		
F1E8	none			F0E8	74	**SSCPD_ID**	PD SSC
F1E6	none	Reserved	Core	F0E6	73	**SSCPD_TB**	PD SSC

Table 135 Bit and Non-Bit Addressable ESFR Areas (cont'd)

Bitaddressable ESFR Area				Non-Bitaddressable ESFR Area			
Long Address (Hex)	**Short Address (Hex)**	**Register Name**		**Long Address (Hex)**	**Short Address (Hex)**	**Register Name**	
F1E4	none			F0E4	72	**SSCPD_RB**	PD SSC
F1E2	none			F0E2	71	**SSCPD_BR**	PD SSC
F1E0	none	Reserved	Core	F0E0	70	**SSCPD_PISEL**	PD SSC
F1DE	EF	Reserved	Core	F0DE	6F		
F1DC	EE	Reserved	Core	F0DC	6E		
F1DA	ED	Reserved	Core	F0DA	6D		
F1D8	EC	Reserved		F0D8	6C	Reserved	Core
F1D6	EB	Reserved		F0D6	6B	**CC2PISEL**	CC2
F1D4	EA	Reserved	Core	F0D4	6A	**CC2IOC**	CC2
F1D2	E9	PCL_14 (**PCL_<pad>**)	PCL	F0D2	69	**GPTPISEL**	GPT12
F1D0	E8	Reserved	Core	F0D0	68	CAPREL	GPT12
F1CE	E7	PCL_38 (**PCL_<pad>**)	PCL	F0CE	67	T6	GPT12
F1CC	E6	PCL_37 (**PCL_<pad>**)	PCL	F0CC	66	T5	GPT12
F1CA	E5	PCL_36 (**PCL_<pad>**)	PCL	F0CA	65	T4	GPT12
F1C8	E4	PCL_35 (**PCL_<pad>**)	PCL	F0C8	64	T3	GPT12
F1C6	E3	PCL_34 (**PCL_<pad>**)	PCL	F0C6	63	T2	GPT12
F1C4	E2	PCL_33 (**PCL_<pad>**)	PCL	F0C4	62	**GPTID**	GPT12
F1C2	E1	PCL_32 (**PCL_<pad>**)	PCL	F0C2	61	Reserved	Core
F1C0	E0	Reserved	Core	F0C0	60	Reserved	Core
F1BE	DF	PCL_31 (**PCL_<pad>**)	PCL	F0BE	5F	**CC2SEM**	CC2
F1BC	DE	PCL_30 (**PCL_<pad>**)	PCL	F0BC	5E	**CC2SEE**	CC2
F1BA	DD	PCL_29 (**PCL_<pad>**)	PCL	F0BA	5D	**T8**	CC2
F1B8	DC	PCL_28 (**PCL_<pad>**)	PCL	F0B8	5C	**T7**	CC2
F1B6	DB	PCL_27 (**PCL_<pad>**)	PCL	F0B6	5B	**T8REL**	CC2
F1B4	DA	PCL_26 (**PCL_<pad>**)	PCL	F0B4	5A	**T7REL**	CC2
F1B2	D9	PCL_25 (**PCL_<pad>**)	PCL	F0B2	59	**CC23**	CC2
F1B0	D8	PCL_24 (**PCL_<pad>**)	PCL	F0B0	58	**CC22**	CC2
F1AE	D7	PCL_23 (**PCL_<pad>**)	PCL	F0AE	57	**CC21**	CC2
F1AC	D6	PCL_22 (**PCL_<pad>**)	PCL	F0AC	56	**CC20**	CC2
F1AA	D5	PCL_21 (**PCL_<pad>**)	PCL	F0AA	55	**CC19**	CC2
F1A8	D4	PCL_20 (**PCL_<pad>**)	PCL	F0A8	54	**CC18**	CC2
F1A6	D3	PCL_19 (**PCL_<pad>**)	PCL	F0A6	53	**CC17**	CC2
F1A4	D2	PCL_18 (**PCL_<pad>**)	PCL	F0A4	52	**CC16**	CC2
F1A2	D1	PCL_17 (**PCL_<pad>**)	PCL	F0A2	51	**CC2ID**	CC2
F1A0	D0	PCL_16 (**PCL_<pad>**)	PCL	F0A0	50	**CC1PISEL**	CC1
F19E	CF	TIM0IC	Int	F09E	4F	**CC1IOC**	CC1
F19C	CE	TIM4IC	Int	F09C	4E	**CC1SEM**	CC1
F19A	CD	MEAS1_TOGGLEIC	Int	F09A	4D	**CC1SEE**	CC1

Table 135 Bit and Non-Bit Addressable ESFR Areas (cont'd)

Bitaddressable ESFR Area				Non-Bitaddressable ESFR Area			
Long Address (Hex)	**Short Address (Hex)**	**Register Name**		**Long Address (Hex)**	**Short Address (Hex)**	**Register Name**	
F198	CC	IIC_PIC	Int	F098	4C	**T1**	CC1
F196	CB	T_INT2IC	Int	F096	4B	**T0**	CC1
F194	CA	TIM3IC	Int	F094	4A	**T1REL**	CC1
F192	C9	MEAS0_TOGGLEIC	Int	F092	49	**T0REL**	CC1
F190	C8	IIC_DIC	Int	F090	48	**CC7**	CC1
F18E	C7	T_INT1IC	Int	F08E	47	**CC6**	CC1
F18C	C6	TIM2IC	Int	F08C	46	**CC5**	CC1
F18A	C5	MEAS1IC	Int	F08A	45	**CC4**	CC1
F188	C4	IIC_EIC	int	F088	44	**CC3**	CC1
F186	C3	S0ASIC	Int	F086	43	**CC2**	CC1
F184	C2	TIM1IC	Int	F084	42	**CC1**	CC1
F182	C1	MEAS0IC	Int	F082	41	**CC0**	CC1
F180	C0	RFSSOTIC	Int	F080	40	**CC1ID**	CC1
F17E	BF	ECOIC	Int	F07E	3F	**IDMANUF**	ID
F17C	BE	KPDIC	Int	F07C	3E	**IDCHIP**	ID
F17A	BD	Reserved		Int	F07A	3D	
	IDMEM	ID F178	BC	PM_INT	Int	F078	
	3C	**IDPROG**	ID F176	BB	Reserved		
Int	F076	3B	**IDMEM2**	ID F174	BA	SSC0EIC	Int
	F074	3A					
F172	B9	SSC0RIC	Int	F072	39		
F170	B8	SSC0TIC	Int	F070	38	Reserved	Core
F16E	B7	Reserved		Int	F06E	37 Reserved	
F16C	B6	Reserved	Int	F06C	36	Reserved	Core
F16A	B5	Reserved		Int	F06A	35	
	Reserved	Core F168	B4	Reserved			Int
	F068	34	Reserved		Core F166	B3	
	S0TM0IC	Int	F066	33			
F164	B2	S0CTSIC	Int	F064	32		
F162	B1	S0ABDETIC	Int	F062	31		
F160	B0	S0ABSTIC	Int	F060	30		
F15E	AF	EOPIC	Int	F05E	2F		
F15C	AE	From_DSP_to_MCU3IC	Int	F05C	2E	**ID_SNUM6**	PCL
F15A	AD	From_DSP_to_MCU2IC	Int	F05A	2D	**ID_SNUM5**	PCL
F158	AC	From_DSP_to_MCU1IC	Int	F058	2C	**ID_SNUM4**	PCL
F156	AB	From_DSP_to_MCU0IC	Int	F056	2B	**ID_SNUM3**	PCL
F154	AA	FEX7IC	Int	F054	2A	**ID_SNUM2**	PCL
F152	A9	FEX6IC	Int	F052	29	**ID_SNUM1**	PCL
F150	A8	FEX5IC	Int	F050	28	**ID_SNUM0**	PCL
F14E	A7	FEX4IC	Int	F04E	27		

Table 135 Bit and Non-Bit Addressable ESFR Areas (cont'd)

Bitaddressable ESFR Area				Non-Bitaddressable ESFR Area			
Long Address (Hex)	**Short Address (Hex)**	**Register Name**		**Long Address (Hex)**	**Short Address (Hex)**	**Register Name**	
F14C	A6	FEX3IC	Int	F04C	26		
F14A	A5	FEX2IC	Int	F04A	25	**S0TMO**	ASC0
F148	A4	FEX1IC	Int	F048	24	Reserved	ASC0
F146	A3	FEX0IC	Int	F046	23	**S0TXFCON**	ASC0
F144	A2	SIMINIC	Int	F044	22	**S0RXFCON**	ASC0
F142	A1	SIMERRIC	Int	F042	21	Reserved	ASC0
F140	A0	SIMOKIC	Int	F040	20	**S0RBUF**	ASC0
F13E	9F	**FLASHIN**	PCL	F03E	1F	**S0TBUF**	ASC0
F13C	9E	Reserved	Core	F03C	1E	**S0PWM**	ASC0
F13A	9D	PCL_13 (**PCL_<pad>**)	PCL	F03A	1D	**S0FDV**	ASC0
F138	9C	PCL_12 (**PCL_<pad>**)	PCL	F038	1C	**S0BG**	ASC0
F136	9B	PCL_11 (**PCL_<pad>**)	PCL	F036	1B	**S0PERID**	ASC0
F134	9A	PCL_10 (**PCL_<pad>**)	PCL	F034	1A		
F132	99	PCL_09 (**PCL_<pad>**)	PCL	F032	19		
F130	98	PCL_08 (**PCL_<pad>**)	PCL	F030	18		
F12E	97	PCL_07 (**PCL_<pad>**)	PCL	F02E	17		
F12C	96	PCL_06 (**PCL_<pad>**)	PCL	F02C	16		
F12A	95	PCL_05 (**PCL_<pad>**)	PCL	F02A	15		PMCU behind PCL BPI
F128	94	PCL_04 (**PCL_<pad>**)	PCL	F028	14	**PMC_TIMER0**	
F126	93	PCL_03 (**PCL_<pad>**)	PCL	F026	13	**MCD_BBS**	Break switch behind intr_ext BPI
F124	92	PCL_02 (**PCL_<pad>**)	PCL	F024	12	Reserved	Core
F122	91	PCL_01 (**PCL_<pad>**)	PCL	F022	11		
F120	90	PCL_00 (**PCL_<pad>**)	PCL	F020	10	**PCL_ID**	PCL
F11E	8F	Reserved	Core	F01E	0F	Reserved	Core
F11C	8E	Reserved	Core	F01C	0E	Reserved	Core
F11A	8D	Reserved	Core	F01A	0D	Reserved	Core
F118	8C	Reserved	Core	F018	0C	Reserved	Core
F116	8B	Reserved	Core	F016	0B	Reserved	Core
F114	8A	Reserved	Core	F014	0A	Reserved	Core
F112	89	Reserved	Core	F012	09	Reserved	
F110	88	Reserved		F010	08	Reserved	
F10E	87	Reserved		F00E	07	Reserved	Core
F10C	86	Reserved		F00C	06	Reserved	Core
F10A	85	Reserved	Core	F00A	05	Reserved	

Table 135 Bit and Non-Bit Addressable ESFR Areas (cont'd)

Bitaddressable ESFR Area				Non-Bitaddressable ESFR Area			
Long Address (Hex)	**Short Address (Hex)**	**Register Name**		**Long Address (Hex)**	**Short Address (Hex)**	**Register Name**	
F108	84	Reserved	Core	F008	04	Reserved	
F106	83	Reserved	Core	F006	03	Reserved	
F104	82	Reserved	Core	F004	02	Reserved	
F102	81	Reserved	Core	F002	01	Reserved	
F100	80	Reserved	Core	F000	00	Reserved	

10.2 X-Bus Register Addresses

Table 136 Address Mapping of X-Bus Peripherals

The Xbus peripherals space area is 8Kbytes from 00 $D000_H$ to $EFFF_H$

X-Bus Peripheral	Physical Address	Number of Bytes	Register Access Clock
	XADRS1: 256bytes (RGSAD: freely programmable, in this case we select: $0EF_H$ RGSZ: 0000 (256 Byte)		
Chip Card Interface	00 $EFFF_H$- 00 $EFC0_H$	64 bytes (connected to one BPI)	X-Bus
SCCU Block	00 $EFBF_H$ - 00 $EF80_H$	64 bytes (connected to one BPI)	X-Bus
Measurement	00 $EF7F_H$- 00 $EF40_H$	64 bytes (connected to one BPI)	X-Bus
Reserved	00 $EF3F_H$ - 00 $EF00_H$	64 bytes (connected to one BPI)	X-Bus
Reserved			
Keypad Block			
	Not Mapped		
Not Mapped	00 $EEFF_H$ 00 $EE00_H$	256byte	Free
	XADRS4: 512bytes RGSAD: freely programmable, in this case we select: $0EC_H$ RGSZ: 0001 (512 Byte)		
Not Used	00 $EDFF_H$ 00 $EDAA_H$	86 bytes free	Free
LED_backlight	00 $EDA8_H$ - 00 $ED96_H$	20 bytes	X-Bus

Table 136 Address Mapping of X-Bus Peripherals (cont'd)

The Xbus peripherals space area is 8Kbytes from 00 $D000_H$ to $EFFF_H$

X-Bus Peripheral	Physical Address	Number of Bytes	Register Access Clock
PMU	00 $ED94_H$ - 00 $ED80_H$	22 bytes (2 reserved in EE94)	X-Bus
Reserved	00 $ED7F_H$ - 00 $ED40_H$	64 bytes	X-Bus
Clock Generation Unit	00 $ED3F_H$ - 00 $ED00_H$	64 (connected to one BPI)	13 MHz
Reserved	00 $ECFF_H$ - 00 $EC80_H$	128 bytes	X-Bus
Not Used	00 $EC7F_H$ 00 $EC40_H$	64 bytes free.	Free
Reserved	00 $EC3F_H$ 00 $EC00_H$	64 bytes	X-Bus
	Not Mapped		
Not Mapped	00 $EBFF_H$ 00 $E800_H$	1Kbyte free	Free
	XADRS2: 2Kbytes RGSAD: freely programmable, in this case we select: $0E0_H$ RGSZ: 0011 (2 Kbytes)		
RF-Control	00 $E7FF_H$ - 00 $E000_H$	1920 bytes for RAM,128 bytes for registers	13 MHz
	XADRS3: 4Kbytes RGSAD: freely programmable, in this case we select: $0D0_H$ RGSZ: 0100(4 Kbytes)		
Not Used	00 DFFF 00 D600	2.5kbytes free	Free

Table 137 Register Overview for XADRS1

Name	Address Offset (Hex)	Functional Block
Reserved	08	
Reserved	0A	
unused	0C	
unused	0E	
unused	10	
unused	12	

Table 137 Register Overview for XADRS1 (cont'd)

Name	Address Offset (Hex)	Functional Block
unused	14	
unused	16	
KBID	18	Keypad
KBDINP	1A	Keypad
KBDOUT	1C	Keypad
KPCTRL	1E	
unused	20	
unused	22	
unused	24	
unused	26	
Reserved	28	XBIU
Reserved	2A	XBIU
Reserved	2A	XBIU
Reserved	2C	ComRAM
Reserved	2E	ComRAM
Reserved	30	ComRAM
Reserved	32	ComRAM
Reserved	34	ComRAM
Reserved	36	ComRAM
Reserved	38	ComRAM
Reserved	3A	ComRAM
unused	3C	
unused	3E	
MEAS_ID	40	Measurement
unused	42	Measurement
Reserved	44	Measurement
unused	46	Measurement
ANA_CTRL1	48	Measurement
ANA_CTRL2	4A	Measurement
MEAS_CTRL1	4C	Measurement
MEAS_CTRL2	4E	Measurement
MEAS_STAT	50	Measurement
unused	52	Measurement
MEAS_DATA0	54	Measurement
unused	56	Measurement
MEAS_DATA1	58	Measurement
unused	5A	Measurement
MEAS_DATA2	5C	Measurement
unused	5E	Measurement
MEAS_DATA3	60	Measurement
unused	62	Measurement

Table 137 Register Overview for XADRS1 (cont'd)

Name	Address Offset (Hex)	Functional Block
MEAS_DATA4	64	Measurement
unused	66	Measurement
MEAS_DATA5	68	Measurement
unused	6A	Measurement
MEAS_DATA6	6C	Measurement
unused	6E	Measurement
MEAS_DATA7	70	Measurement
unused	72	Measurement
MEAS_CLK	74	Measurement
unused	76	Measurement
unused	78	Measurement
unused	7A	Measurement
unused	7C	Measurement
unused	7E	Measurement
SCCUID	80	SCCU Block
unused	82	SCCU Block
SCCUTDMINL	84	SCCU Block
Not Used	86	SCCU Block
SCCUSLPCTRL	88	SCCU Block
Not Used	8A	SCCU Block
SCCUSCCTRL	8C	SCCU Block
Not Used	8E	SCCU Block
SCCUREFINL	90	SCCU Block
Not Used	92	
SCCU Block **SCCUNQTZ**	94	SCCU Block
Not Used	96	SCCU Block
SCCUWAITL	98	SCCU Block
SCCUWAITH	9A	SCCU Block
SCCUHWWAKEUPL	9C	SCCU Block
SCCUHWWAKEUPH	9E	SCCU Block
SCCUTDMOUTL	A0	SCCU Block
Not Used	A2 Not used	SCCU Block
SCCUREFL	A4	SCCU Block
SCCUREFH	A6	SCCU Block
SCCUCLKSTA	A8	SCCU Block
unused	AA	SCCU Block
SCCUSMSTA	AC	SCCU Block
unused	AE	SCCU Block
Reserved	B0	SCCU Block
unused	B2	SCCU Block
Reserved	B4	SCCU Block

Table 137 Register Overview for XADRS1 (cont'd)

Name	Address Offset (Hex)	Functional Block
unused	B6	SCCU Block
Reserved	B8	SCCU Block
unused	BA	SCCU Block
SCCUSPCR	BC	SCCU Block
unused	BE	SCCU Block
SIMID	C0	Chip Card
unused	C2	Chip Card
SIMCTRL	C4	Chip Card
unused	C6	Chip Card
SIMBRF	C8	Chip Card
unused	CA	Chip Card
SIMSTATUS	CC	Chip Card
unused	CE	Chip Card
SIMIRQEN	D0	Chip Card
unused	D2	Chip Card
SIMRXSPC	D4	Chip Card
unused	D6	Chip Card
SIMTXSPC	D8	Chip Card
unused	DA	Chip Card
SIMCHTIMER1	DC	Chip Card
SIMCHTIMER2	DE	Chip Card
Reserved	E0	Chip Card
unused	E2	Chip Card
Reserved	E4	Chip Card
unused	E6	Chip Card
SIMTX	E8	Chip Card
unused	EA	Chip Card
SIMRX	EC	Chip Card
unused	EE	Chip Card
SIMINS	F0	Chip Card
unused	F2	Chip Card
SIMP3	F4	Chip Card
unused	F6	Chip Card
SIMSW1	F8	Chip Card
unused	FA	Chip Card
SIMSW2	FC	Chip Card
unused	FE	Chip Card

Table 138 Register Overview for XADRS2

Name	Address Offset (Hex)	Functional Block
TID	008	GSM IF
RFCON1	010	GSM IF
unused	012	GSM IF
RFCON2	014	GSM IF
unused	016	GSM IF
RFSSCTB	018	GSM IF
unused	01A	GSM IF
TCOR	01C	GSM IF
unused	01E	GSM IF
TTOVF	020	GSM IF
unused	022	GSM IF
TINT1	024	GSM IF
unused	026	GSM IF
TINT2	028	GSM IF
unused	02A	GSM IF
TOFFSET	02C	GSM IF
unused	02E	GSM IF
TFSKIP	030	GSM IF
unused	032	GSM IF
TCLT	034	GSM IF
unused	036	GSM IF
TCEAP	038	GSM IF
unused	03A	GSM IF
TEAPT	03C	GSM IF
unused	03E	GSM IF
TEAPB	040	GSM IF
unused	042	GSM IF
TGERB	044	GSM IF
TGERT	046	GSM IF
TPARATGERT	048	GSM IF
unused	04A	GSM IF
TFADE1	04C	GSM IF
TFADE2	04E	GSM IF
Not Used	050	GSM IF
Not Used	052	GSM IF
GSMCLK1B	054	GSM IF
GSMCLK1T	056	GSM IF
GSMCLK2B	058	GSM IF
unused		GSM IF
GSMCLK2T	05A	GSM IF
GSMCLK3	05C	GSM IF

Table 138 Register Overview for XADRS2

Name	Address Offset (Hex)	Functional Block
Free	05E-5F	GSM IF
Not Used	060	GSM IF
Free	062-7F	GSM IF
RF Control RAM (448 words)	080-3FF	GSM IF
GSM Timer RAM (512 words)	400-7FF	GSM IF

Table 139 Register Overview for XADRS4

Name	Address Offset (Hex)	Functional Block
Reserved	008	
Reserved	010	
Reserved	012	
Reserved	080-0FF	
CGUID	100	CGU Block
RTCIF	102	CGU Block
MST_CLK_CTRL	104	CGU Block
PLL_CTRL	106	CGU Block
PHX_CTRL	108	CGU Block
SIFCLKS	10A	CGU Block
RST_CTRL_STA	10C	CGU Block
Reserved	10E	CGU Block
Free	110-13E	
Reserved	140-146	
Reserved	148	
Reserved	150	
Reserved	152	
Reserved	154	
Reserved	156	
Reserved	158	
Reserved	15A	
Reserved	15C	
Reserved	15E	
Reserved	160	
Reserved	162	
Reserved	164	
Reserved	166	
Reserved	168	
Reserved	16A	
Reserved	16C	
	16E	

Table 139 Register Overview for XADRS4 (cont'd)

Name	Address Offset (Hex)	Functional Block
	170	
	172	
	174	
	176	
	178	
	17A	
	17C	
	17E	
Reserved	148	
Reserved	150	
Reserved	152	
Reserved	154	
Reserved	156	
Reserved	158	
Reserved	15A	
Reserved	15C	
Reserved	15E	
Reserved	160	
Reserved	162	
Reserved	164	
Reserved	166	
Reserved	168	
Reserved	16A	
Reserved	16C	
Reserved	16E	
Reserved	170	
Reserved	172	
Reserved	174	
Reserved	176	
Reserved	178	
Reserved	17A	
Reserved	17C	
Free	17E	
PMU_GENCTRL	180	PMU
PMU_PWRCTRL1	182	PMU
PMU_PWRCTRL2	184	PMU
PMU_LPDCTRL	186	PMU
PMU_CHGCTRL	188	PMU
PMU_INTCTRL	18A	PMU

Table 139 Register Overview for XADRS4 (cont'd)

Name	Address Offset (Hex)	Functional Block
PMU_ID	18E	PMU
PMU_STAT	190	PMU
LED_k1	196	LED Backlight
LED_k1max	198	LED Backlight
LED_k2	19A	LED Backlight
LED_k2min	19C	LED Backlight
LED_k2max	19E	LED Backlight
LED_cip	1A0	LED Backlight
LED_cv	1A2	LED Backlight
LED_clavt	1A4	LED Backlight
LED_cpi	1A6	LED Backlight
LED_CTRL	1A8	LED Backlight
Free	1AA-1FF	

10.3 RF Register Set

For proper initialization, several internal registers have to be programmed before the RF part is activated:

The following settings (telegrams) should be sent to RF part (refer to **Section 7.8 "RF Control" (on page 169)**):

- 0x03000F
- 0x24901F
- 0x04142F
- 0x0087CF.

Additionally bit 23 of **XO_TUNE Register** has to be set to "1" (Mandatory / See on **Page 502**

Table 140 CHANNEL1 Register

Bit	FIX	BITNAME	Function
0	0	ADD0	Register Address CHANNEL1
1	0	ADD1	
2	0	ADD2	
3	0	ADD3	
4		F0	Fractional Channelword Part1 Power On Default = 0
5		F1	
6		F2	
7		F3	
8		F4	
9		F5	
10		F6	
11		F7	
12		F8	
13		F9	
14		F10	
15		F11	
16		F12	
17		F13	
18		F14	
19		F15	
20		F16	
21		F17	
22		F18	
23		F19	

Table 141 CHANNEL2 Register

Bit	FIX	BITNAME	Function
0	0	ADD0	Register Address CHANNEL2
1	1	ADD1	
2	0	ADD2	
3	0	ADD3	
4		F20	Fractional Channelword Part2
5		F2	Power On Default ∓ 0
6		F22	

Table 141 CHANNEL2 Register (cont'd)

7		CH0	Integer Channelword
8		CH	132_D
Power On Default =			P
9		CH2	
10		CH3	
11		CH4	
12		CH5	
13		CH6	
14		CH7	
15		ON0	Power Mode, If ON[1:0] =
16		ON1	00 ALL OFF 01 Not used 10 PLL test ON 11 ALL ON
			Power On Default = 00
17	0		Mandatory
18		TRX	Transmit Receive Switch 0 = RX 1 = TX
			Power On Default = 0
19		BSW0	Band Switch, if BSW[1:0] =
20		BSW1	00850 MHz 01900 MHz 101800 MHz 111900 MHz
			Power On Default = 00
21			Mandatory
22			Mandatory
23			Mandatory

Table 142 RXTX Register

Bit	FIX	BITNAME	Function
0	0	ADD0	Register Address
1	0	ADD1	RXTX
2	1	ADD2	
3	0	ADD3	
4		RXGAIN0	RX Gain (Low/High Frequency Bands), if RXGAIN[1:0] =
5		RXGAIN1	11High (57/56dB) 01Medium (43/42dB) 00Low (22dB/22dB) 10Not supported
			Power On Default = 00
6		RXCORR0	RX Gain Correction
7		RX	
Power On Default =		CO	9 RXCORR3
8		RXCORR2	

lt = 0100(0dB correction)

Table 142 RXTX Register (cont'd)

10	0		Mandatory
11		RXCM0	Mandatory
12		RXCM1	
13		RXGS0	RX Gain Step #0 0 = OFF (0dB) 1 = ON (+12dB) Power On Default = 0
14		RXGS1	RX Gain Step #1 0 = OFF (0dB) 1 = ON (+6dB) Power On Default = 0
15		RXGS2	RX Gain Step #2 0 = OFF (0dB) 1 = ON (-6dB) Power On Default = 0
16		RXGS3	RX Gain Step #3 0 = OFF (0dB) 1 = ON (-3dB) Power On Default = 0
17	0	OFC	DC Offset Compensation 0 = OFF 1 = ON Power On Default = 0
18	0		Mandatory
19	0		Mandatory
20	0		Mandatory
21			Mandatory
22			Mandatory
23			Mandatory

Table 143 XO_INIT1 Register

Bit	FIX	BITNAME	Function
0	0	ADD0	Register Address XO_INIT1
1	1	ADD1	
2	1	ADD2	
3	0	ADD3	
4		XOMODE0	External XO Mode 0 = Internal XO 1 = External XO Signal Applied to XO Pin Power On Default = 0

Table 143 XO_INIT1 Register (cont'd)

5		XOMODE1	Mandatory
6	0	XOMODE2	
7		XOMODE3	Mandatory
8	0	XOMODE4	
9	0	XOMODE5	Mandatory
10	0	XOMODE6	Mandatory
11	0	XOMODE7	Mandatory
12	0	XOMODE8	Mandatory
13	0	XOMODE9	Mandatory
14	0	XOMODE10	
15	0	XOMODE11	
16	0		Capacitance adaptation for crystal core XOSETUP0: increase cap C2 by 1pF XOSETUP1: increase cap C2 by 2pF XOSETUP2: increase cap Cx by 0.5pF XOSETUP3: increase cap Cx by 0.9pF Cv is the capacitance from XOX to ground; Cx is the capacitance between XO and XOX Power On Default XOSETUP[3:0] = 0000
17	0		
18	0		
19	0		
20		XOCAL0	Crystal Subrange Selection Power On Default XOCAL[2:0] = 100
21		XOCAL1	
22		XOCAL2	
23		XOCAL3	Crystal Size Selection 0 = Load Capacitance 10pF 1 = Load Capacitance 8pF Power On Default = 1

Table 144 XO_INIT2 Register

Bit	FIX	BITNAME	Function
0	0	ADD0	Register Address XO_INIT2
1	1	ADD1	
2	0	ADD2	
3	1	ADD3	
4		ALPHA0	ALPHA Coefficient for LUXO
5		AL	P PH fault = 0d wer
On De		A1	
6		ALPHA2	
7		ALPHA3	
8		ALPHA4	
9		ALPHA5	
10		ALPHA6	
11		ALPHA7	
12		ALPHA8	
13		ALPHA9	

Table 144 XO_INIT2 Register (cont'd)

14		GAMMA0	GAMMA Coefficient for LUXO Power On Default = 0d
15		GAMMA1	
16		GAMMA2	
17		GAMMA3	
18		GAMMA4	
19		GAMMA5	
20		GAMMA6	
21		GAMMA7	
22		GAMMA8	
23		GAMMA9	

Table 145 XO_TUNE Register

Bit	FIX	BITNAME	Function
0	0	ADD0	Register Address XO_TUNE
1	0	ADD1	
2	1	ADD2	
3	1	ADD3	
4		AFC0	Frequency Correction Value Power On Default AFC = 4096_D for starting at middle AFC (AFC0=LSB/AFC12=MSB)
5		AFC1	
6		AFC2	
7		AFC3	
8		AFC4	
9		AFC5	
10		AFC6	
11		AFC7	
12		AFC8	
13		AFC9	
14		AFC10	
15		AFC11	
16		AFC12	
17		AFC13	Mandatory = 0000 for ENLUXO = 1
18		AFC14	
19		AFC15	
20		AFC16	
21	0		Mandatory
22	0		
23	0		Mandatory, has to be set to 1.

11 Electrical and Temperature Characteristics

Attention: All values mentioned in this chapter are target values and have to be confirmed.

11.1 Maximum Values (Destruction limits)

This section contains the temperature, voltage, and ESD limit values.

WARNING

The maximum ratings may not be exceeded under any circumstances, not even momentarily or individually, as permanent damage to the device will result.

11.1.1 Maximum ESD

Table 146 Maximum ESD

Parameter	Symbol	Limit Values		Unit
		Minimum	Maximum	
ESD		-	1000	V
ESD exception for certain pins			500	V

11.1.2 Maximum Temperature

Table 147 Maximum Temperature

Parameter	Symbol	Limit Values		Unit
		Minimum	Maximum	
Temperature		-55	150	° C

11.1.3 Maximum Voltages (Digital, Analog and PMU)

Table 148 Maximum Power Supply Voltages

Parameter	Symbol	Limit Values			Unit
		Minimum	Maximum	AC Max	
Digital Power Supply	VDD_LD1	-0.15	1.7		V
	VDD_RTC	-0.15	2.5		
	VDDP_SIM	-0.3	3.6		
	VDDP_MEM	-0.15	3.6		
	VDDP_IO	-0.3	3.6		
	VDD_PLL	-0.15	1.7		
Analog Power Supply	VDD_ANA	-0.15	3		
	VDD_BUF	-0.15	3		
Battery Supply	VBAT1, VBAT3, VBAT4	-0.15	6.0	0.35	Vpp
	VBAT2	-0.15	6.0		

11.1.4 Maximum Current

Shorts at the output components of standard digital drivers may cause sink or source currents up to 100 mA per pin. Exposure to these currents may destroy power buses or pad cells inside the PMB7880.

The sum of the sink or source currents at all pads belonging to the same digital I/O supply domain (V_{DDP_IO}, V_{DDP_EB}, V_{DDP_SIM}) must not exceed [TBD: 20mA] at any time.

The sum of sinked or sourced currents in the connection between the bumps and the balls must not exceed (TBD: 20 mA @ 2.7 V). This corresponds to a maximum thermal dissipation inside the chip of approximately. (TBD: 55 mW).

11.1.5 Absolute Maximum Ratings

Table 149 Absolute Maximum Ratings, T_{AMB} = -30°C .. + 85°C

#	Parameter	Symbol	Limit Values		Unit	Remarks
			Minimum	Maximum		
1	Supply Voltages 1.5 V	V_{15}	-0.3	1.6	V	
2	Supply Voltages 2.5 V	V_{25}	-0.3	3.0	V	
3	Digital Input Voltage	V_{I15}	-0.3	3.0	V	CLK, DA, EN
4	Analog Input Voltage	V_{A25}	-0.3	3.0	V	A, AX, B, BX
5	Total Power Dissipation	P_{tot}		1100	mW	at ambient temp. less than 80°C
6	Junction Temperature	T_j		125	°C	
7	Storage Temperature	T_S	-55	125	°C	
8	Thermal Resistance (junction to ambient)	R_{thJA}		40	K/W	Soldered diepad
9	ESD-Integrity[1)]	V_{ESD}	1000		V	
10	ESD-Integrity	V_{ESD2}	500		V	RX1, RX1X, RX2, RX2X, RX3, RX3X, RX4, RX4X, TX1, TX2, XO, XOX
11	RX1/RX2 Receiver Input Level	$Pin\_{RX1/RX2}$		+5	dBm	Single Ended, GSM850/E-GSM900
12	RX3/RX4 Receiver Input Level	$Pin\_{RX3/RX4}$		+8	dBm	Single Ended, DCS1800/GSM1900
13	Supply voltage for fuse programming	V_{DD_EFUSE}	-0.1	4.1	V	

1) HBM according MIL-Std 883D, method 3015.7, and EOS/ESD assn. Standard S5.1-1993- only CMOS input/output pins.

11.2 Normal Operation Values

11.2.1 Static (DC Characteristics)

11.2.1.1 Temperature

Table 150 contains the operating temperature range

Table 150 Operating Temperature

Parameter	Symbol	Limit Values		Unit
		Minimum	Maximum	
Temperature		-30	+85	° C

11.2.1.2 Voltages

Due to the integrated PMU and the internal power supply connection, the normal operating voltage ranges correspond to the output voltage of the integrated regulators (refer to **Figure 91 LDO Connections Overview**). The blocks in the RTC domain are functional down to VDD_RTC=1.0V (in case of VDD_LD1=0V), in order to allow battery change whilst buffering with an external capacitor at VDD_RTC

Table 151 contain the operating voltage range for fuse programming.

Table 151 Operating Voltage for Fuse programming

Parameter	Ball Name	Limit Values			Unit
		Minimum	Typical	Maximum	
FUSE supply voltage for Fuse programming	VDD_EFUSE	0.0	0.0	0.0	V

11.2.1.3 Currents

Table 152 and **Table 153** contain the input Power Supply currents.

Table 152 Digital Power Supply Currents

Parameter	Ball Name	Limit Values			Unit
		Minimum	Typical	Maximum	
With CPU Master Clock running at 26 MHz and DSP Master Clock running at 104 MHz	VDD_LD1		87[1) 2)]		mA
With CPU Master Clock running at 32 KHz			0.16[1) 2)]		mA
With CPU Master Clock running at 26MHz and the RTC reference clock 32 kHz	VDD_RTC		0.5	0.5	mA
With CPU Master Clock running at 32 kHz and the RTC reference clock 32 kHz		0.003			mA
Without CPU Master Clock, VDD_RLD1=0V but with RTC reference clock at 32 kHz	VDD_RTC VDDP_SIM		0.004		mA mA
	VDD_PLL				mA
	VDDP_MEM				mA
	VDDP_IO				mA

Table 153 Analog Power Supply Currents (IDD measured with VDD_ANA = 2.5 V)

Ball Name Comments	Limit Values			Unit	
	Minimum	**Typical**	**Maximum**		
VDD_ANA (Receiver active, Standard mode)	5.1	6.4	7.7	mA	DC analog input clk_bbrx = 13 MHz RXON = 1
VDD_ANA (Receiver active, Enhanced mode)	13	16.5	20.5	mA	**BB_CTRL.BB_ADCMODE** = 0 DC analog input clk_bbrx = 26 MHz RXON = 1
VDD_ANA (Power Amplifier)	0.3	0.75	1.1	mA	**BB_CTRL.BB_ADCMODE** = 1 TXON = 1 clk_pa1 = 6.5 MHz clk_par = 6.5 MHz
VDD_ANA (Only Voiceband Receive DAC on)	2		2.5 3	mA	DC digital input = half of full scale **AFE_VRXCTRL2.RXDAC** = 1 digital input = 0
VDD_ANA (DAC on + Highpower Output Amplifier on)	1.8	3.4	4.2	mA	F26M (XO) = 52 MHz **AFE_VRXCTRL2.VEPPA** = 1 **AFE_VRXCTRL2.RXDAC** =1 digital input = 0
VDD_ANA (DAC on + Highpower Output Amplifier on)	1.6	3	4.2	mA	F26M (XO) = 52 MHz **AFE_VRXCTRL1.EPSAV** = 1
VDD_ANA (DAC on + Low Power Output Amplifier on)	2		2.5 3	mA	**AFE_VRXCTRL2.RXDAC** = 1 digital input pattern
VDD_ANA (Only Voiceband Receive ADC on)	0.8	1.6	2	mA	F26M (XO) = 52 MHz **AFE_VTXCTRL.TXMODE** = 01
VDD_ANA (DAC on + Highpower Output Amplifier on)	5.4	6.8	8.2	mA	**AFE_VRXCTRL2.VEPPA** = 1 **AFE_VRXCTRL2.RXDAC** = 1 digital input pattern
VDD_ANA (DAC on + Highpower Output Amplifier on)	1.6	3.2	4.2	mA	F26M (XO) = 52 MHz **AFE_VRXCTRL1.EPSAV** = 1
VDD_ANA (DAC + Highpower Output Amplifier)	0.2	0.5	1.2	mA	**AFE_VRXCTRL2.VEPPA** = 1 **AFE_VRXCTRL2.RXDAC** = 1
VDD_ANA	0	0.06	0.2	mA	
VDD_ANA	0.5	1	1.5	mA	**MEAS_CTRL2.ADCON** = 1

The mapping between the signals and the pad is in **Chapter 3 Pin Descriptions**.

11.2.1.4 Capacitances and Resistors

Table 154 Input Capacitances and Resistors

Parameter	Symbol	Limit Values			Unit
		Minimum	Typical	Maximum	
Digital I/O; DC value	C_{I/O_DIG}			10	pF
Oscillator (F32K); Input driven	C_{F32k}			25	pF

*Note: These values are **not** measured by production tests or characterization procedures. They are derived from simulations.*

Table 155 External Load Capacitances and Resistors

Parameter	Symbol	Limit Values		Unit
		Minimum	Maximum	
I²C Signal / Clock Pins			50	pF
Oscillator (F32K, OSC32K) Values have to be calculated depending on the external quartz	C_{32k}	-	-	pF

Further Requirements and Notes

1. **EBU**: For external pull up resistances and PCB integration guidelines on EBU pins (see **Section 7.10 External Bus Unit** and **Section 11.2.2.3 EBU**).
2. **Mixed Signal and Analog Interfaces**: The description of external load conditions of the mixed signal interfaces and the measurement interface are contained in:
   - **Section 11.3.1.1 Audio Receive Part**
   - **Section 11.3.2 Measurement Interface**, etc.
3. **PLL**: For V_{DD_PLL} an external filter must be applied. The blocking capacitor has to be located as close as possible to the pins V_{DD_PLL}, V_{SS_PLL} (see **Figure 187**).

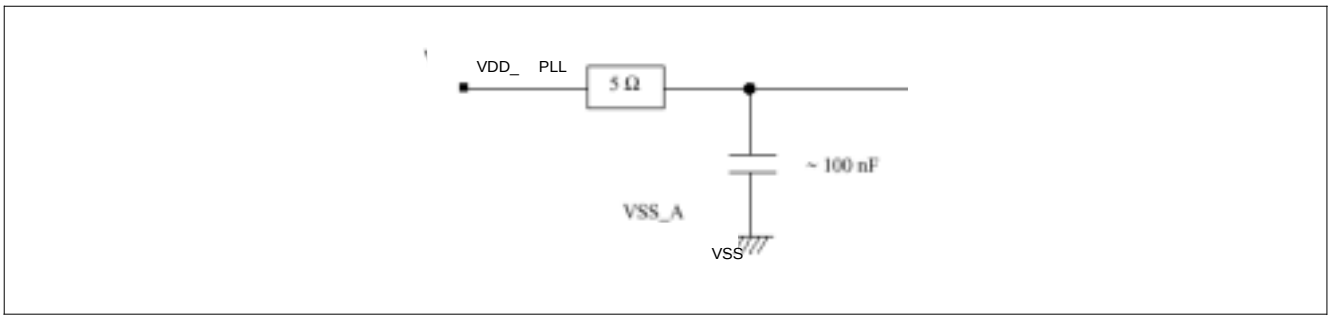


Figure 187 External Filter at PLL Supply

4. **I²C**: The dedicated I²C pads (pins SCL and SDA) have open drain functionality. Pull up resistors have to be implemented externally. Note that those signals also appear as alternative functions at standard I/O pads. Those pads can be configured by the port control logic to show OD functionality or active driver behavior. But those pads do not fulfill the I²C standard.
5. **Keypads**: The total sum of possible source currents from the keypad outputs to V_{ss} should be restricted to maximum 20 mA Dynamic.

11.2.1.5 Pad Characteristics

Table 156 Pad Output Current

Table legend:
P) Production test program
C) Characterization test program

Parameter	Symbol	Limit Values		Unit	Test Condition
		Minimum	Maximum		
Digital Pins in the I/O supply domain of the EBU					
EBU: L-input voltage for memory Interface	V_{IL_PEBU}	- 0.2	0.3	V	
EBU: H-input voltage for memory Interface	V_{IH_PEBU}	V_{DDP_EBU} - 0.3	V_{DDP_EBU} + 0.2	V	
EBU: L-output voltage for pad classes B & C of Memory Interface at a load current of 100 µA	V_{OL_PEBU}		V_{SSP_EBU} + 0.1	V	I_{OH_PEBU} = +100 ∝A C) 1)
EBU: H-output voltage for pad classes B & C of memory interface at a load current of 100 ∝A	V_{OH_PEBU}	V_{DDP_EBU} - 0.1		V	I_{OH_PEBU} = -100 ∝A C) 1)
Input/Output leakage current	I_{IZ}		0,7	∝A	0.2 V < V_{IN} < V_{IHmax}
Digital Pins in the I/O supply domain of the SIM Interface					
L-input voltage for SIM Interface	V_{IL_SIM}	0	0.2 *	V_{DDP_SIM} V	
L-input for SIM Interface	V_{IL_SIM}		0.37	V	V_{DDP_SIM} = 1.76 V
L-Input for SIM Interface	V_{IL_SIM}		0.60	V	V_{DDP_SIM} = 2.96 V
H-input voltage for SIM Interface	V_{IH_SIM}	0.7 x V_{DDP_SIM}	3.3	V	
H-input for SIM Interface	V_{IH_SIM}	1.22		V	V_{DDP_SIM} = 1.84 V
H-input voltage for SIM Interface	V_{IH_SIM}	1.95		V	V_{DDP_SIM} = 2.96 V
Input/Output leakage current	I_{IZ}		0,7	∝A	0.2 V < V_{IN} < V_{IHmax}
Full Swing Input of RTC					
L-input voltage for F32K input (PU32KEN = 0)	V_{IL_32KFS}	-0.25 V	+0.2 V	V	
H-input voltage for F32K input (PU32KEN = 0)	V_{IH_32KFS}	V_{DD_RTC} - 0.2	V_{DD_RTC} + 0.25	V	
I²C Interface (only for special OD drivers, Not valid for alternative functions)					
L-input voltage for inputs I2C_SCL and I2C_SDA (V_{DD}-related levels)	V_{ILI2C}	-0.3	0.3 * V_{DDP_IO}	V	
H-input voltage for inputs I2C_SCL and I2C_SDA (V_{DDP}-related levels)	V_{IHI2C}	0.7 x V_{DDP_IO}	max[V_{DDP_IO} + 0.5, 3.3 V]	V	
Hysteresis for inputs I2C_SCL and I2C_SDA	V_{HI2C}	0.05 x V_{DDP_IO}			In accordance with I2C bus specification
L-output voltage for outputs I2C_SCL and I2C_SDA	V_{OLI2C}		0.4 V		I_{OL} = +3.0 mA
Input/Output leakage current	I_Z			∝ A	0.2 V < V_{IN} < V_{IHmax}

Table 156 Pad Output Current (cont'd)

Table legend:
P) Production test program
C) Characterization test program

Parameter	Symbol	Limit Values		Unit	Test Condition
		Minimum	Maximum		
156.1Standard Digital I/Os in I/O Domains VDDP_IO ; standard digital I/Os in domain VDD_RTC (only valid in the range of 1,8 V <VDD_RTC < 2,25 V) **Not valid for special OD pads of I²C interface and pins F32k, OSC32k.**					
L-input voltage for general digital pads (except OD pads for I²C)	V_{IL}	- 0.2	0.2 * V_{DDP_DIG}	V	
H-input voltage for general digital pads (except OD pads I2C_SCL, I2C_SDA)	V_{IH}	0.7 x V_{DDP_DIG}	3.3	V	
L-output voltage for pad class B	V_{OLB}		0.2	V	I_{OL} = +2.5 mA P)
			0.35	V	I_{OL} = +5.0 mA C)
L-output voltage for pad class C	V_{OLC}		0.2	V	I_{OL} = +2.0 mA P)
			0.35	V	I_{OL} = +4.0 mA C)
L-output voltage for pad class D	V_{OLD}		0.2	V	I_{OL} = +1.0 mA P)
			0.35	V	I_{OL} = +2.0 mA C)
L-output voltage for pad class E/F	V_{OLE}		0.2	V	I_{OL} = +1.0 mA P)
			0.35	V	I_{OL} = +1.5 mA C)
H-output voltage for pad class B slow	$V_{OHBslow}$				I_{OH} = -15 mA V_{DDP_IO} = 1.7 V C)
H-output voltage for pad class B slow	$V_{OHBslow}$				I_{OH} = -10 mA V_{DDP_IO} = 1.7 V C)
H-output voltage for pad class B	V_{OHB}	V_{DDP_DIG} - 0.35		V	I_{OH} = -5.0 mA C)
		V_{DDP_DIG} - 0.2			I_{OH} = -2.5 mA P)
H-output voltage for pad class C	V_{OHC}	V_{DDP_DIG} - 0.35		V	I_{OH} = -4.0 mA C)
		V_{DDP_DIG} - 0.2	V		I_{OH} = -2.0 mA P)
H-output voltage for pad class D	V_{OHD}	V_{DDP_DIG} - 0.35		V	I_{OH} = -2.0 mA C)
		V_{DDP_DIG} - 0.2		V	I_{OH} = -1.0 mA P)
H-output voltage for pad class E/F	V_{OHE}	V_{DDP_DIG} - 0.35		V	I_{OH} = -1.5 mA C)
		V_{DDP_DIG} - 0.2		V	I_{OH} = -1.0 mA P)
Input/Output leakage current	I_{IZ}	I	± 0.7	∝A	0.2 V < V_{IN} < V_{IHmax}

1) This value only applies to the DC characteristics. AC characteristics, such as ripples and switching noise, are not included in this definition.

I/O Supply Domains

All digital I/O supply domains of the E-GOLDvoice (including the RTC) have to be operated under operating conditions. It is not possible to power down pad domains separately.

*Note: All values in **Table 156** and **Table 157** are derived from simulations.*

Table 157 Pull-Up/Pull-Down Currents

Parameter	Symbol	Limit Values		Unit	Test Condition V_{S_d}, V_{DD_PAD}
		Minimum	Maximum		
Pull-up input current for pull class A (1)	I_{PUa}		-450 -150	∝A	0 V, V_{DD_PAD} = 2.94 V C 0 V, V_{DD_PAD} = 1.90 V P
		-40 -15		∝A	1.90 V, V_{DD_PAD} = 2.75 V C 1.20 V, V_{DD_PAD} = 1.70 V P
Pull-up input current for pull class B (2)	I_{PUb}		-100 -35	∝A	0 V, V_{DD_PAD} = 2.94 V C) 0 V, V_{DD_PAD} = 1.90 V P)
		-5 -1		∝A	1.90 V, V_{DD_PAD} = 2.75 V C) 1.20 V, V_{DD_PAD} = 1.70 V P)
Pull-up input current for pull class C (3)	I_{PUc}		-30 -10	∝A	0 V, V_{DD_PAD} = 2.94 V C) 0 V, V_{DD_PAD} = 1.90 V P)
		-0.5 -0.0		∝A	1.90 V, V_{DD_PAD} = 2.75 V C) 1.20 V, V_{DD_PAD} = 1.70 V P)
Pull-down input current for pull class A (1)	I_{PDa}		+450 +200	∝A	2.94 V, V_{DD_PAD} = 2.94 V C) 1.90 V, V_{DD_PAD} = 1.90 V P)
		+7 +2		∝A	0.2 V, V_{DD_PAD} = 2.75 V C) 0.2 V, V_{DD_PAD} = 1.70 V P)
Pull-down input current for pull class B (2)	I_{PDb}		+100 +35	∝A	2.94 V, V_{DD_PAD} = 2.94 V C) 1.90 V, V_{DD_PAD} = 1.90 V P)
		+1.0 +0.5		∝A	0.2 V, V_{DD_PAD} = 2.75 V C) 0.2 V, V_{DD_PAD} = 1.70 V P)
Pull-down input current for pull class C (3)	I_{PDc}		+30 +10	∝A	2.94 V, V_{DD_PAD} = 2.94 V C) 1.90 V, V_{DD_PAD} = 1.90 V P)
		+0.5 +0.0		∝A	0.2 V, V_{DD_PAD} = 2.75 V C) 0.2 V, V_{DD_PAD} = 1.70 V P)

11.2.2 Dynamic (AC Characteristics)

Attention: F26M now refers to the XO pad. The major 26 MHz reference clock is now generated by the internal RF oscillator.

11.2.2.1 Rise and Fall Time of Signals

The level reached at a certain time is basically determined by following components:

- Supply voltage drop from ideal voltage at pad supply pin to pad due to:
  - Noise on power supply (determined in the application)
  - Voltage drop over pad supply rail: max. 30 mV
- Pad resistance Ri and external load C (RC loading curve with timing constant Ri * C).

Table 158 shows the resistances Ri for the different driver classes.

Table 158 Pad Resistances (1,75 V to 2,75 V)

Parameter 2,25-2,75 V	Symbol	Limit Values			Unit
		Minimum	Typical	Maximum	
Pad resistance at 2.5 mA - 5 mA load / rising edge for Pad Class B	RiB_r			50	&
Pad resistance at 2.5 mA - 5 mA load/falling edge for Pad Class B	RiB_f			50	&
Pad resistance at 2.5 mA - 5 mA load/rising edge for Pad Class Bslow	RiBslow_r			70	&
Pad resistance at 2.5 mA - 5 mA load/falling edge for Pad Class Bslow	RiBslow_f			70	&
Pad resistance at 2 mA - 4 mA load/rising edge for Pad Class C	RiC_r			70	&
Pad resistance at 2 mA - 4 mA load/falling edge for Pad Class C	RiC_f			70	&
Pad resistance at 1 mA - 2 mA load/rising edge for Pad Class D	RiD_r			115	&
Pad resistance at 1 mA - 2 mA load/falling edge for Pad Class D	RiD_f			115	&
Pad resistance at 1 mA - 1.5 mA load/rising edge for Pad Class E	RiE_r			130	&
Pad resistance at 1 mA - 1.5 mA load/falling edge for Pad Class E	RiE_f			120	&
Pad resistance at 1 mA - 1.5 mA load/rising edge for Pad Class F	RiF_r			180	&
Pad resistance at 1 mA - 1.5 mA load/falling edge for Pad Class F	RiF_f			180	&

11.2.2.2 Testing Waveforms

11.2.2.2.1 Testing Driven Values for All Digital I/Os Except EBU_AHB I/Os

Input signals during AC testing are driven to V_{DD} - 0.2 V for a logical 1 and to 0.2 V for a logical 0. Output signals during AC testing are measured at V_{DD} - 0.3 V for a logical 1 and at 0.3 V for a logical 0. The AC testing input/output waveforms are shown in **Table 188**.

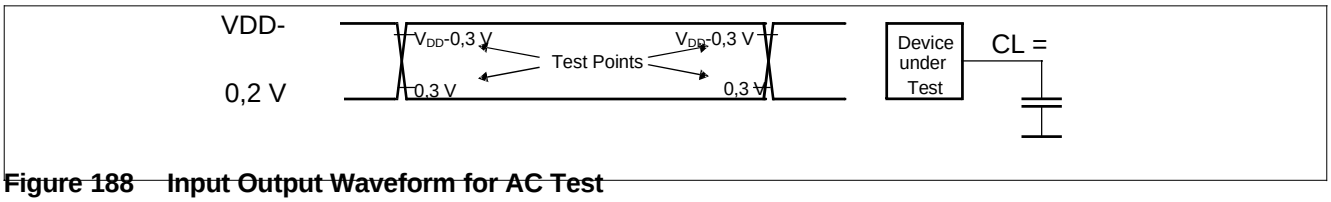


Figure 188 Input Output Waveform for AC Test

11.2.2.2.2 Testing Float Waveforms

A load voltage V_{LOAD} and a load current (I_{OH}= -5 mA, I_{OL} = 5 mA) are applied to the pin during the tristate phases (see **Figure 189**).

The end of floating is detected at the latest when a 100 mV change from V_{LOAD} occurs.

The begin of floating is detected at the latest when a 100 mV change from the loaded V_{OH}/V_{OL}-level occurs.

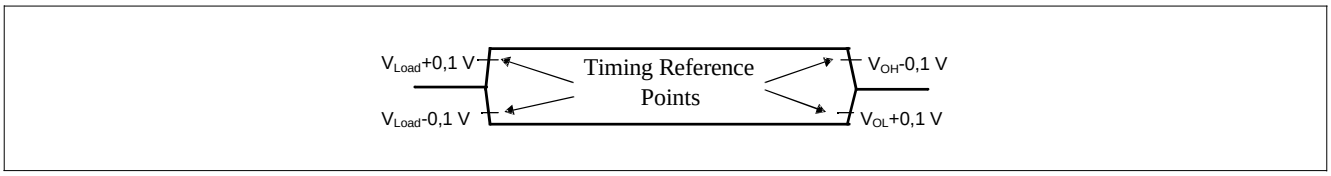


Figure 189 Float Waveforms

11.2.2.3 EBU

WARNING

The values in Table 159 and Table 160 are from E-GOLDradio V2.0. They will be validated and updated when the E-GOLDvoice characterization is finished.

11.2.2.3.1 Measured Parameters for VDDP_EBU = 1.8 V and 2.85 V

In **Table 159** BUSCON refers only to **BUSCON0** and **BUSCON1**. In this table the limits are the same for both VDDP_EBU voltages

The characterization for EBU and PMCU is done at VDDP_EBU = 1.8 V and 2.85 V. These results give the same limits for both voltages.

Table 159 EBU and PMCU Measured Parameters

Parameter Symbol	Timing Diagram	Limit Values 1.8 V			Limit Values 2.85 V			Unit
		Min	Typ	Max	Min	Typ	Max	
	Symbol Description							
$ts_{1\_adv\_f\_pmc}$	Page 526	tbd	tbd	tbd	tbd	tbd	tbd	ns
	ADV_n low after F26M (XO) falling edge **in pagemode during Nth (with N = 1) offpage access**: Without Read-Write Delay (BUSCON.RWD = 1) Without extended ALE (BUSCON.ALE = 0)							
$ts_{1\_adv\_r\_pmc}$	Page 526	tbd	tbd	tbd	tbd	tbd	tbd	ns
	ADV_n high after F26M (XO) falling edge **in pagemode during Nth (with N = 1) offpage access**: Without Read-Write Delay (BUSCON.RWD = 1) Without extended ALE (BUSCON.ALE = 0)							
$ts_{2\_adv\_f\_pmc}$	Page 528	tbd	tbd	tbd	tbd	tbd	tbd	ns
	ADV_n low after F26M (XO) falling edge **in pagemode during Nth (with N = 1) offpage access:** With Read-Write Delay (BUSCON.RWD = 0) Without extended ALE (BUSCON.ALE = 0)							
$ts_{2\_adv\_r\_pmc}$	Page 528	tbd	tbd	tbd	tbd	tbd	tbd	ns
	ADV_n high after F26M (XO) rising edge **in pagemode during Nth (with N = 1) offpage access:** With Read-Write Delay (BUSCON.RWD = 0) Without extended ALE (BUSCON.ALE = 0)							
$ts_{3\_adv\_f\_pmc}$	Page 529	tbd	tbd	tbd	tbd	tbd	tbd	ns
	ADV_n low after F26M (XO) falling edge **in pagemode during Nth (with N = 1) offpage access:** Without Read-Write Delay (BUSCON.RWD = 1) With extended ALE (BUSCON.ALE = 1)							
$ts_{3\_adv\_r\_pmc}$	Page 529	tbd	tbd	tbd	tbd	tbd	tbd	ns
	ADV_n high after F26M (XO) falling edge **in pagemode during Nth (with N = 1) offpage access:** Without Read-Write Delay (BUSCON.RWD = 1) With extended ALE (BUSCON.ALE = 1)							
$ts_{4\_adv\_f\_pmc}$	Page 530	tbd	tbd	tbd	tbd	tbd	tbd	ns
	ADV_n low after F26M (XO) falling edge **in pagemode during Nth (with N = 1) offpage access:** With Read-Write Delay BUSCON.RWD = 0) With extended ALE (BUSCON.ALE = 1)							
$ts_{4\_adv\_r\_pmc}$	Page 530	tbd	tbd	tbd	tbd	tbd	tbd	ns
	ADV_n high after F26M (XO) rising edge **in pagemode during Nth (with N = 1) offpage access:** With Read-Write Delay (BUSCON.RWD = 0) With extended ALE (BUSCON.ALE = 1)							

Table 159 EBU and PMCU Measured Parameters (cont'd)

Parameter Symbol	Timing Diagram	Limit Values 1.8 V			Limit Values 2.85 V			Unit
		Min	Typ	Max	Min	Typ	Max	
	Symbol Description							
$ts_{4\_data\_fl\_e}$	**Page 521**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	Write Data valid after F26M (XO) falling edge: First switching data bit from Z to valid) Without or with Early Write (BUSCON.EW = 0) With or without Read-Write Delay (BUSCON.RWD = 0 or 1)							
ts_{5_data}	**Page 521**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	Write Data valid after F26M (XO) falling edge: Last switching data bit from Z to valid Without or with Early Write (BUSCON.EW = 0 or 1) With or without Read-Write Delay (BUSCON.RWD = 0 or 1)							
ts_{5_addr}	**Page 521**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	Address valid after F26M (XO) falling edge: Last switching address bit from invalid to valid							
ts_{5_csx} (with x= [0,1,3])	**Page 521**	tbd (Except for pad ADV_N)	tbd	tbd	tbd (Except for pad ADV_N)	tbd	tbd	ns
	CSx_n valid after F26M (XO) falling edge:							
ts_{5_advn} (= ts_{5_csx} with x=2)	Early CS (**SYSCON.CSCFG** = 1)							
	Page 521	tbd	tbd	tbd	tbd	tbd	tbd	ns
	CS2_n (ADV_n) valid after F26M (XO) falling edge:							
$ts_{5\_cs0\_pmc}$	Early CS (**SYSCON.CSCFG** = 1)							
	Page 526	tbd	tbd	tbd	tbd	tbd	tbd	ns
	Flash CS0_n valid after F26M (XO) falling edge **in pagemode:** Flash is on CS0 Early CS (**SYSCON.CSCFG** = 1, **PMC_TIMER0.EN** = 0, **PMC0.PAEN** = 1, **PMC0.MEMCS** =1)							
ts_{6_csx} (with x= [0,1,3])	**Page 523**	tbd (Except for pad ADV_N)	tbd	tbd	tbd (Except for pad ADV_N)	tbd	tbd	ns
	CSx_n valid after F26M (XO) rising edge:							
ts_{6_advn} (= ts_{6_csx} with x=2)	Normal CS (**SYSCON.CSCFG** = 0)							
	Page 523	tbd	tbd	tbd	tbd	tbd	tbd	ns
	CS2_n (ADV_n) valid after F26M (XO) rising edge:							
$ts_{7\_wr\_f}$	Early CS (**SYSCON.CSCFG** = 1)							
	Page 522	tbd	tbd	tbd	tbd	tbd	tbd	ns
	WR_n valid after F26M (XO) rising edge:							
$ts_{7\_wr\_r}$	Without Read-Write Delay (BUSCON.RWD = 1)							
	Page 521	tbd	tbd	tbd	tbd	tbd	tbd	ns
	WR_n invalid after F26M (XO) rising edge: With Early Write (BUSCON.EW = 1)							

Table 159 EBU and PMCU Measured Parameters (cont'd)

Parameter Symbol	Timing Diagram	Limit Values 1.8 V			Limit Values 2.85 V			Unit
		Min	Typ	Max	Min	Typ	Max	
	Symbol Description							
$ts_{7\_rd\_f}$	**Page 522**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	RD_n valid after F26M (XO) rising edge: With Early Write (BUSCON.EW = 1)							
$ts_{7\_oe\_f\_pmc}$	**Page 526**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	OE_n valid after F26M (XO) rising edge **in pagemode:** Without Read-Write Delay (BUSCON.RWD = 1) Without extended ALE (BUSCON.ALE = 0)							
$ts_{7\_oe\_r\_pmc}$	**Page 526**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	OE_n invalid after F26M (XO) falling edge **in pagemode:** With or without Read-Write Delay With or without extended ALE							
$ts_{8\_rd\_f}$	**Page 521**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	RD_n valid after F26M (XO) falling edge: With Read-Write Delay (BUSCON.RWD = 0)							
$ts_{8\_rd\_r}$	**Page 521**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	RD_n invalid after F26M (XO) falling edge: With or without Read-Write Delay (BUSCON.RWD = 0 or 1)							
$ts_{8\_wr\_f}$	**Page 521**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	WR_n valid after F26M (XO) falling edge: With Read-Write Delay (BUSCON.RWD = 0)							
$ts_{8\_oe\_f\_pmc}$	**Page 528**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	OE_n valid after F26M (XO) falling edge **in pagemode:** With Read-Write Delay (BUSCON.RWD = 0) Without extended ALE (BUSCON.ALE = 0)							
$ts_{9\_oe\_f\_pmc}$	**Page 529**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	OE_n valid after F26M (XO) rising edge **in pagemode:** Without Read-Write Delay (BUSCON.RWD = 1) With extended ALE (BUSCON.ALE = 1)							
ts_{9_data}	**Page 524**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	Read Data hold to F26M (XO) falling edge							
$ts_{10\_oe\_f\_pmc}$	**Page 530**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	OE_n valid after F26M (XO) falling edge **in pagemode:** With Read-Write Delay (BUSCON.RWD = 0) With extended ALE (BUSCON.ALE = 1)							
ts_{10_data}	**Page 524**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	Read Data setup to F26M (XO) falling edge							
ts_{11_addr}	**Page 521**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	Address invalid after F26M (XO) falling edge: First switching address bit from valid to invalid							

Table 159 EBU and PMCU Measured Parameters (cont'd)

Parameter Symbol	Timing Diagram	Limit Values 1.8 V			Limit Values 2.85 V			Unit
		Min	Typ	Max	Min	Typ	Max	
	Symbol Description							
ts_{11_csx} (with x= [0,1,3])	**Page 523**	tbd	tbd	tbd	tbd	tbd	tbd	ns
		(Except for pad ADV_N)			(Except for pad ADV_N)			
	CSx_n invalid after F26M (XO) rising edge: Normal CS (**SYSCON.CSCFG** = 0)							
ts_{11_advn} (= ts_{11_csx} with x=2)	**Page 523**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	CS2_n (ADV_n) valid after F26M (XO) falling edge: Early CS (**SYSCON.CSCFG** = 1)							
$ts_{11\_2\_csx}$ (with x= [0,1,3])	**Page 521**	tbd	tbd	tbd	tbd	tbd	tbd	ns
		(Except for pad ADV_N)			(Except for pad ADV_N)			
	CSx_n invalid after F26M (XO) falling edge: Early CS (**SYSCON.CSCFG** = 1)							
$ts_{11\_2\_advn}$ (= $ts_{11\_2\_csx}$ with x=2)	**Page 521**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	CS2_n (ADV_n) valid after F26M (XO) falling edge: Early CS (**SYSCON.CSCFG** = 1)							
$ts_{11\_cs0\_pmc}$	**Page 526**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	Flash CS0_n invalid after F26M (XO) falling edge **in pagemode:** Flash is on CS0 Early CS (**SYSCON.CSCFG** = 1, **PMC_TIMER0.EN** = 0, **PMC0.PAEN** = 1, **PMC0.MEMCS** =1)							
$ts_{12\_data\_f}$	**Page 521**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	Write Data invalid after F26M (XO) falling edge: First switching data bit from valid to Z With Early Write (BUSCON.EW = 1)							
$ts_{12\_data\_r}$	**Page 524**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	Write Data invalid after F26M (XO) rising edge: First switching data bit from valid to Z Without Early Write (BUSCON.EW = 0)							
$ts_{13\_bus\_tris}$	**Page 521**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	Write Data invalid after F26M (XO) falling edge: Last switching data bit from valid to Z With Early Write (BUSCON.EW = 1)							
ts_{18_wr}	**Page 524**	tbd	tbd	tbd	tbd	tbd	tbd	ns
	WR_n invalid after F26M (XO) falling edge: Without Early Write (BUSCON.EW = 0)							

Table 159 EBU and PMCU Measured Parameters (cont'd)

Parameter Symbol	Timing Diagram	Limit Values 1.8 V			Limit Values 2.85 V			Unit
		Min	Typ	Max	Min	Typ	Max	
	Symbol Description							
$ts_{19\_cs0\_pmc}$		tbd	tbd	tbd	tbd	tbd	tbd	ns
	Flash CS0_n invalid after F26M (XO) falling edge **in pagemode:** The Flash is on CS0 Early CS (**SYSCON.CSCFG** = 1, **PMC_TIMER0.EN** = 1, **PMC0.PAEN** = 1)							

If **SYSCON**.WRCFG = 1, pin WR acts as $\overline{WRL}$ and pin $\overline{BHE}$ acts as $\overline{WRH}$, else pin $\overline{WR}$ and $\overline{BHE}$ retain their normal function.

If **BUSCONx.CSWENx** = 1, pin $\overline{CSx}$ acts as $\overline{WRCSx}$, else pin $\overline{CSx}$ retains its normal function.

11.2.2.3.2 Derived Parameters for VDDP_EBU = 1.8 V and 2.85 V

The timing parameters in **Table 160** are not characterized directly, but calculated from **Table 159 EBU and PMCU Measured Parameters**.

Table 160 EBU and PMCU Derived Parameters

Symbol Descriptions	Parameter Symbol	Limit Values 1.8 V			Limit Values 2.85 V			Unit
		Min	Typ	Max	Min	Typ	Max	
ADV plus data set-up ($ts_{1\_adv\_f\_pmc}$ + ts_{10_data})	ts_{20}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Early CS plus data set-up (ts_{5_csx} + ts_{10_data})	ts_{21}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Normal $\overline{CS}$ plus data set-up (ts_{6_csx} + ts_{10_data})	ts_{22}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Address plus data set-up (ts_{5_addr} + ts_{10_data})	ts_{23}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Early write timing (ts_{5_data} - $ts_{7\_wr\_r}$)	ts_{24}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Data valid to WR high: normal WR ($ts_{5\_data\_max}$ - $ts_{18\_wr\_min}$)	ts_{26}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Read enable timing with RW delay ($ts_{8\_rd\_f}$ + ts_{10_data})	ts_{28}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Data read write spacing ($ts_{8\_rd\_f\_max}$ - $ts_{12\_data\_r\_max}$)	ts_{29}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Address hold after write (ts_{11_addr} - ts_{18_wr}) [1]	ts_{30}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Data from $\overline{WR}$ / $\overline{CS}$ ($ts_{7\_wr\_r}$ - $ts_{12\_data\_f}$)	ts_{35}	tbd	tbd	tbd	tbd	tbd	tbd	ns
$\overline{WR}$ high pulse width ($ts_{7\_wr\_r}$ - $ts_{8\_wr\_f}$)	ts_{50}	tbd	tbd	tbd	tbd	tbd	tbd	ns
$\overline{RD}$ high pulse width ($ts_{8\_rd\_r\_max}$ - $ts_{8\_rd\_f\_min}$)	ts_{60}	tbd	tbd	tbd	tbd	tbd	tbd	ns

Table 160 EBU and PMCU Derived Parameters (cont'd)

Symbol Descriptions	Parameter Symbol	Limit Values 1.8 V			Limit Values 2.85 V			Unit
		Min	Typ	Max	Min	Typ	Max	
Read enable timing without RW delay ($ts_{7\_rd\_f\_max} + ts_{10\_data}$)	ts_{65}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Avoids data conflict when going from WR to RD: Normal WR ($ts_{13\_bus\_tri\_max} - ts_{7\_rd\_f\_min}$)	ts_{66}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Write low timing: Early WR ($ts_{7\_wr\_f\_max} - ts_{7\_wr\_r\_min}$)	ts_{67}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Read enable timing without RW delay ($ts_{7\_rd\_f\_max} - ts_{9\_data}$)	ts_{71}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Address hold time WR cycle: Normal write ($ts_{11\_address\_min} - ts_{18\_wr\_r\_max}$)	ts_{72}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Address hold time WR cycle: Early write ($ts_{11\_address\_min} - ts_{7\_wr\_r\_max}$)	ts_{73}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Avoid data conflict when going from WR to RD: Normal WR and RW delay ($ts_{13\_bus\_tri\_max} - ts_{8\_rd\_r\_min}$)	ts_{74}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Avoid data conflict when going from WR to RD: Normal WR and RW delay ($ts_{8\_rd\_r\_max} - ts_{7\_rd\_f\_min}$)	ts_{75}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Write high timing: Normal WR ($ts_{18\_wr\_max} - ts_{7\_wr\_f\_min}$)	ts_{76}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Write high timing: Normal WR and RW delay ($ts_{18\_wr\_max} - ts_{8\_wr\_f\_min}$)	ts_{77}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Write high timing: Early WR ($ts_{7\_wr\_r\_max} - ts_{7\_wr\_f\_min}$)	ts_{78}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Write low timing: Early WR and RW delay ($ts_{8\_wr\_f} - ts_{7\_wr\_f}$)	ts_{79}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Write low timing: Normal WR ($ts_{7\_wr\_f\_max} - ts_{18\_wr\_min}$)	ts_{80}	tbd	tbd	tbd	tbd	tbd	tbd	ns
Write low timing: Normal WR and RW delay ($ts_{18\_wr\_max} - ts_{8\_wr\_f\_min}$)	ts_{81}	tbd	tbd	tbd	tbd	tbd	tbd	ns

Notes

1. *ts_{20}, ts_{21}, and ts_{23} can be used to calculate the access time required for an external memory.*
2. *ts_{24}, ts_{25}, and ts_{26} can be used to calculate the data set up time for external memory.*
3. *ts_{28} can be used to calculate the time that the external memory has to enable its output for a read.*

4. ts_{29} can be used to calculate the time the data bus is tri-stated between write and read.

IF **BUSCONx.CSRENx** = 1, pin $\overline{CSx}$ acts as $\overline{RDCSx}$, else pin $\overline{CSx}$ retains its normal function.

Muxed Bus is not supported in E-GOLDvoice and is not included within these timings.

Read data are latched with the same clock edge that triggers the address change and the rising $\overline{RD}$ edge. Therefore, address changing before the end of $\overline{RD}$ have no impact on read cycles.

Attention: The X-Bus clock (XCLK) is an internal clock signal which is measured via pin CLKOUT (the address and data bus measurement is done with XCLK = 26 MHz).
The X-Bus clock is identical to the MCU clock.
When the MCU clock is derived directly from the shaper clock (refer to Section 7.2.1 Clock Generation Unit), XCLK can be asymmetrical (asymmetrical duty cycle) which will degrade the performance of the address and data bus.
Only when the X-Bus clock is derived from the PLL can a symmetrical XCLK can be guaranteed.

11.2.2.3.3 Timing Diagrams for EBU Parameter Characterization

BUSCON in this section refers to **BUSCON0** and **BUSCON1**.

SYSCON in this section refers to **SYSCON**.

Figure 190 shows timing for the EBU parameters.

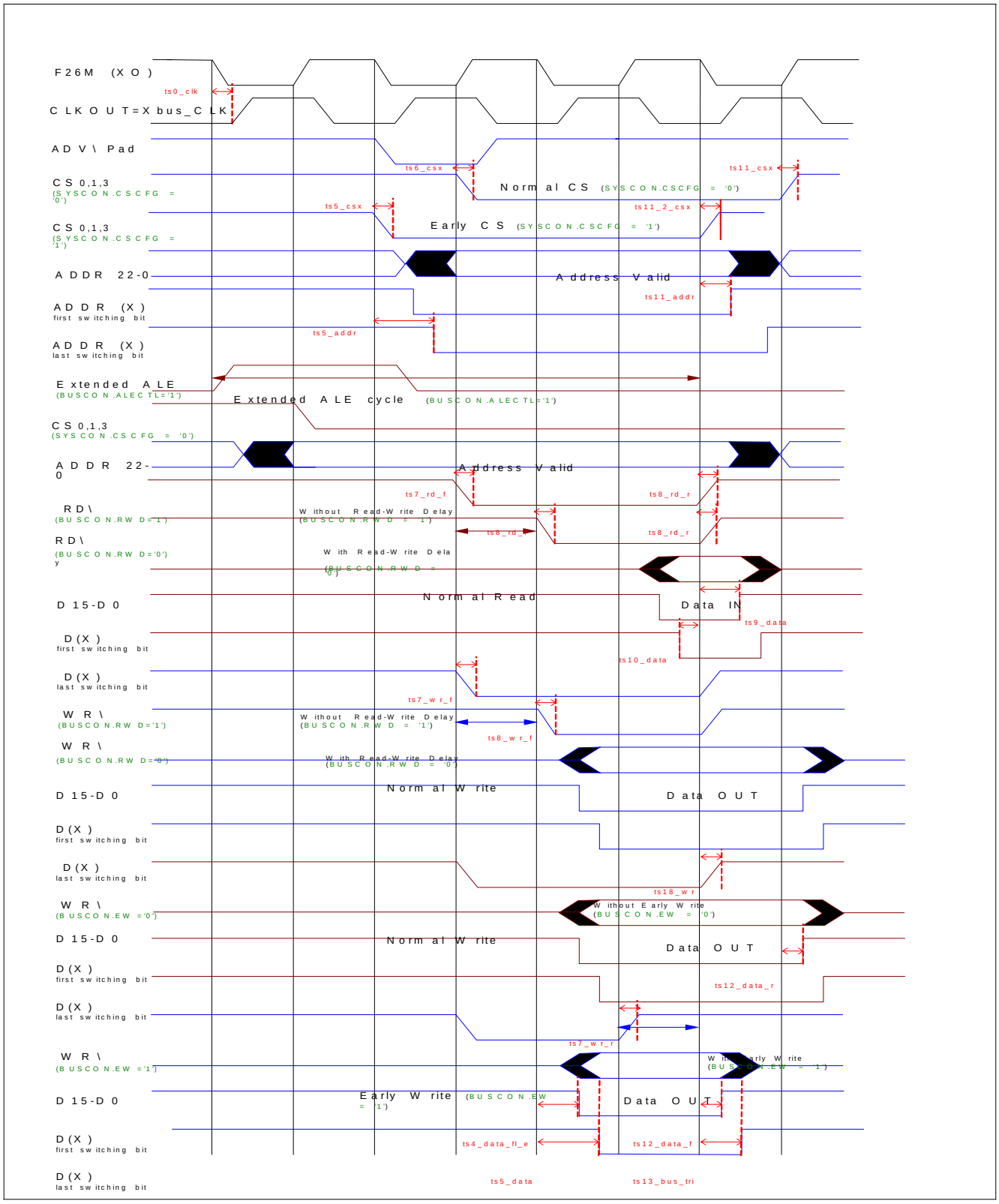


Figure 190 Overview of EBU Parameter Characterizations

Parameters

ts_{0_clk} $ts_{4\_data\_fl\_e}$ ts_{5_addr} ts_{5_csx} t_{s5_data} $ts_{7\_wr\_r}$ $ts_{8\_rd\_f}$ $ts_{8\_wr\_f}$ $ts_{11\_2\_csx}$ ts_{11_addr} $ts_{12\_data\_f}$ $ts_{13\_bus\_tris}$

are characterized under the following conditions (see **Figure 191**):

- **BUSCON** = $05AF_H$:
  - **BUSCON.RWD** = 0 => with Read Write Delay
  - **BUSCON.ALE** = 0 => without Extended ALE
  - **BUSCON.BSW** = 0 => without Tristate inserted when address window changes
- **SYSCON** = 7444_H:
  - **SYSCON.CSCFG** = 1 => with Early Chip Select

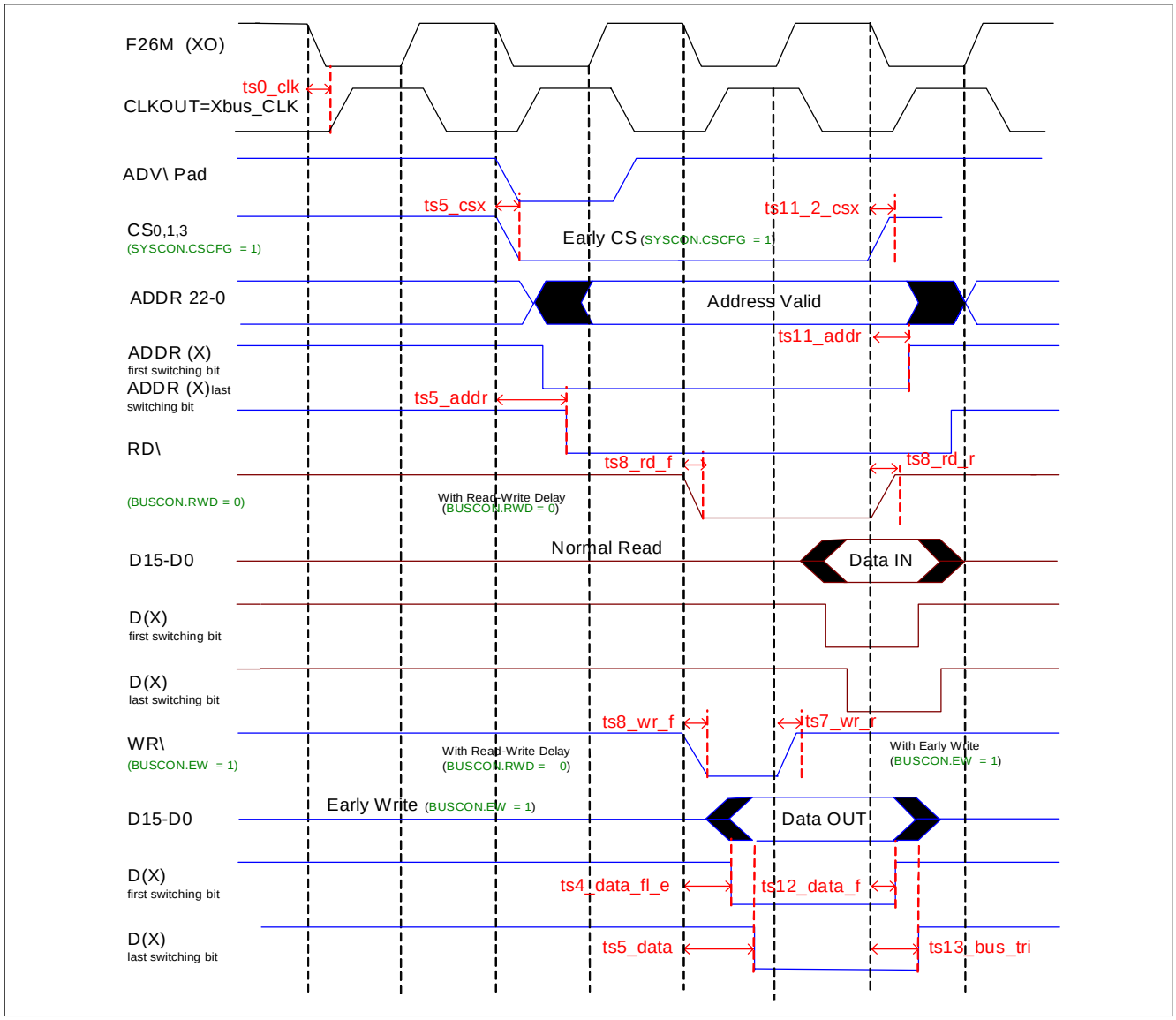


Figure 191 Timing Diagram of Parameters

Parameters of $ts_{7\_rd\_f}$ and $ts_{7\_wr\_f}$ are characterized under the following conditions (see **Figure 192**):

- **BUSCON** = $05BF_H$:
  - **BUSCON.RWD** = 1 => without Read Write Delay
  - **BUSCON.EW** = 1 => with Early Write
  - **BUSCON.ALE** = 0 => without Extended ALE
  - **BUSCON.BSW** = 0 => without Tristate inserted when address window changes
- **SYSCON** = 7444_H:
  - **SYSCON.CSCFG** = 1 => with Early Chip Select

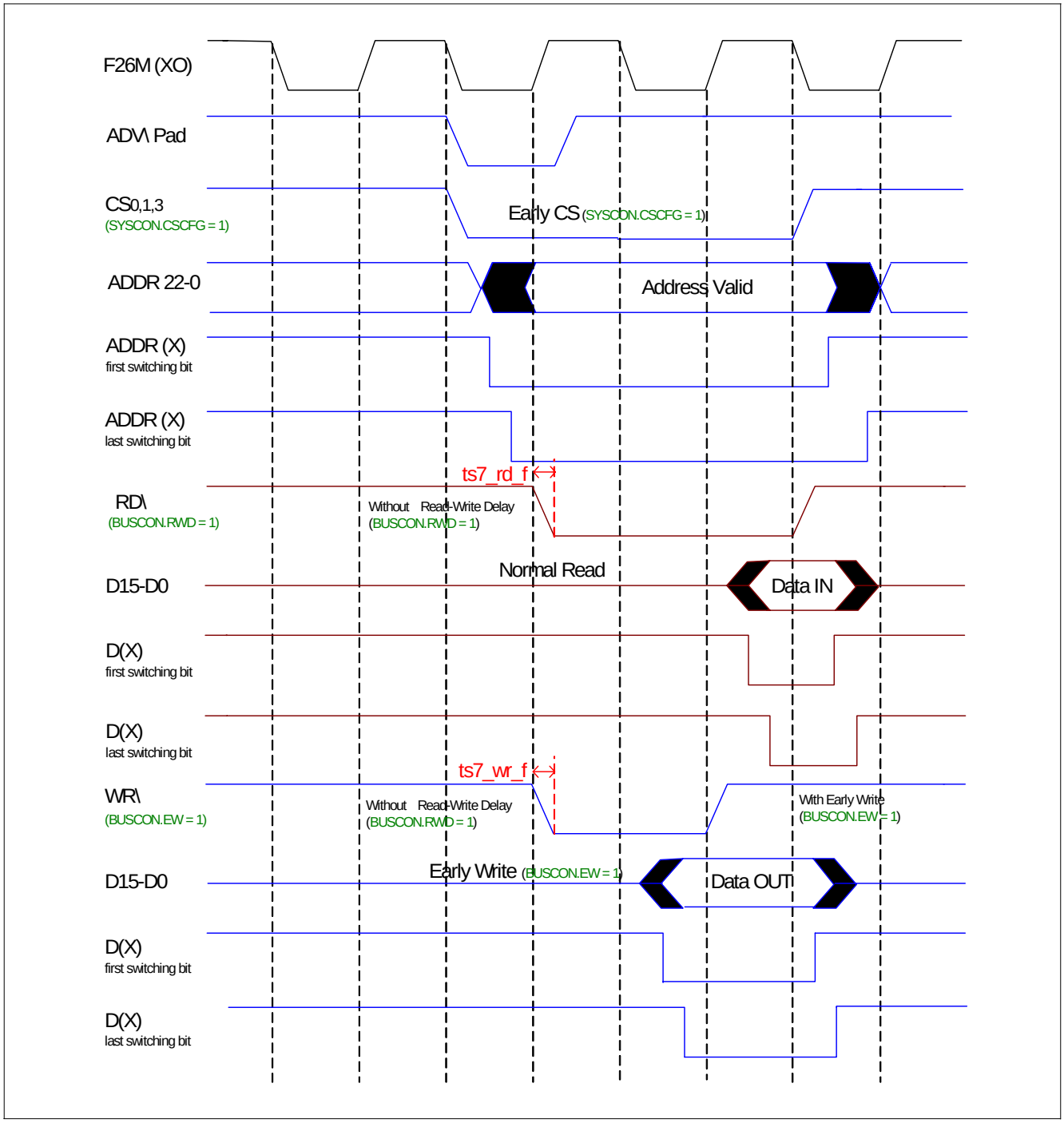


Figure 192 Timing Diagram for Parameters

Parameters ts_{6_csx} and ts_{11_csx} are characterized under the following conditions (see **Figure 193**):

- **BUSCON** = $05BF_H$:
  - **BUSCON.RWD** = 1 => without Read Write Delay
  - **BUSCON.EW** = 1 => with Early Write
  - **BUSCON.ALE** = 0 => without Extended ALE
  - **BUSCON.BSW** = 0 => without Tristate inserted when address window changes
- **SYSCON** = 7404_H:
  - **SYSCON.CSCFG** = 0 => without Early Chip Select

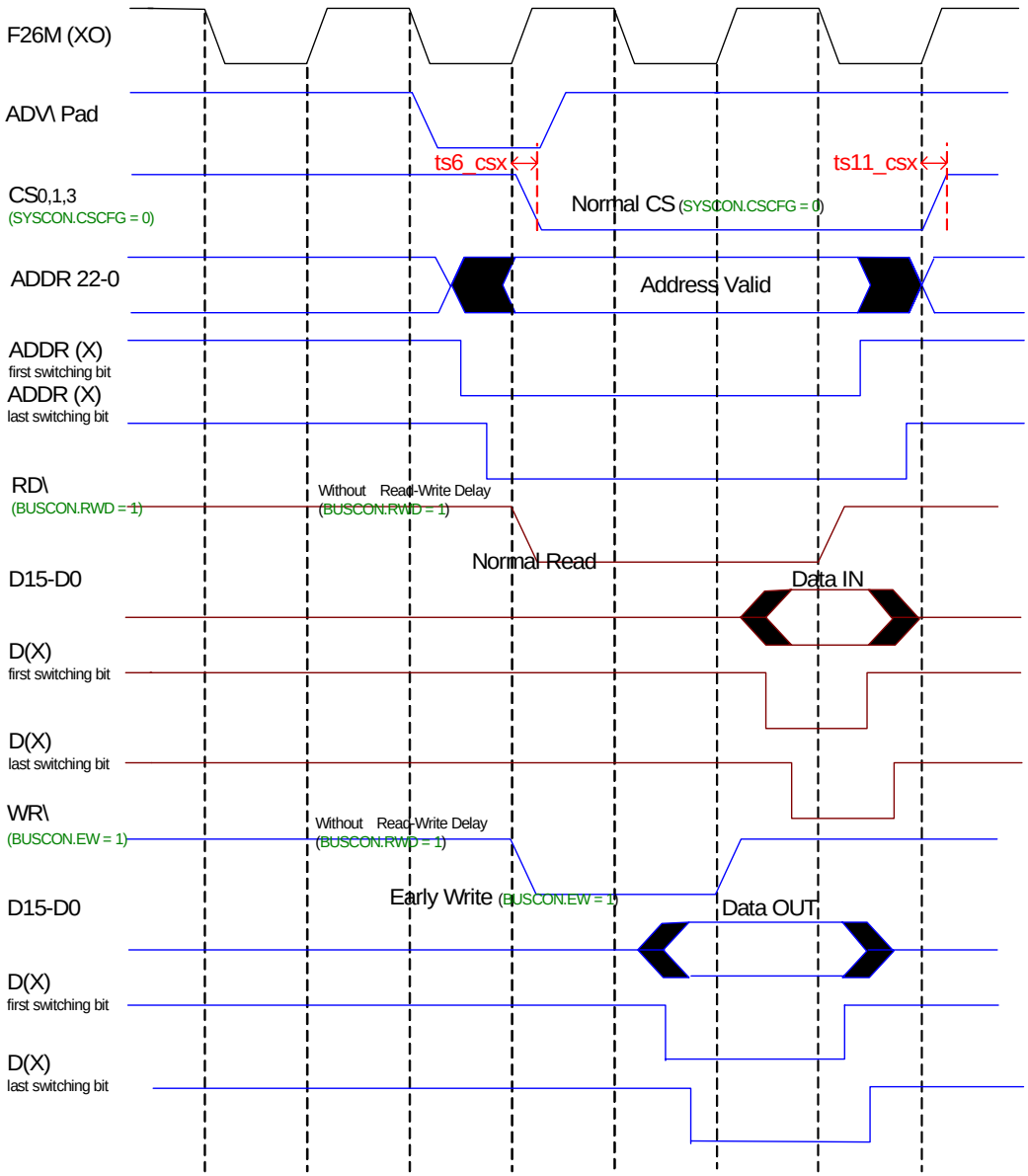


Figure 193 Timing Diagram for Parameters

Parameters of $ts_{12\_data\_r}$, ts_{18_wr}, ts_{9_data}, and ts_{10_data} are characterized under the following conditions (see **Figure 194**):

- **BUSCON** = $04BF_H$:
  - **BUSCON.RWD** = 1 => without Read Write Delay
  - **BUSCON.EW** = 0 => without Early Write
  - **BUSCON.ALE** = 0 => without Extended ALE
  - **BUSCON.BSW** = 0 => without Tristate inserted when address window changes
- **SYSCON** = 7404_H:
  - **SYSCON.CSCFG** = 0 => without Early Chip Select

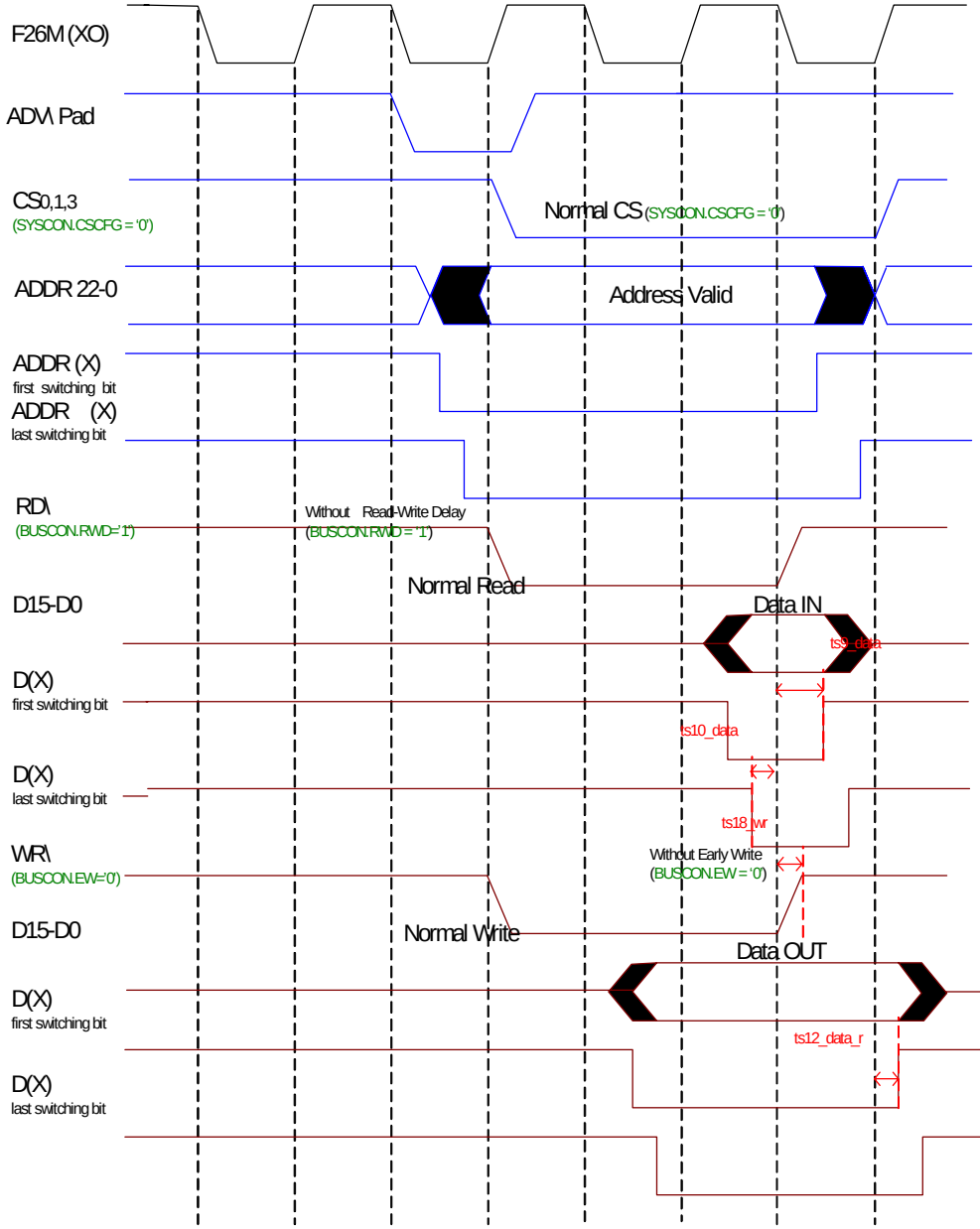


Figure 194 Timing Diagram for Parameters

11.2.2.3.4 Timing Diagrams for PMCU Parameter Characterization

In the following PMCU timing diagrams the F26M (XO) signal is 52 MHz with a 25% - 75% duty cycle.

BUSCON in this section refers to **BUSCON0** and **BUSCON1**.

SYSCON in this section refers to **SYSCON**.

Figure 195 shows timing for the PMCU parameters.

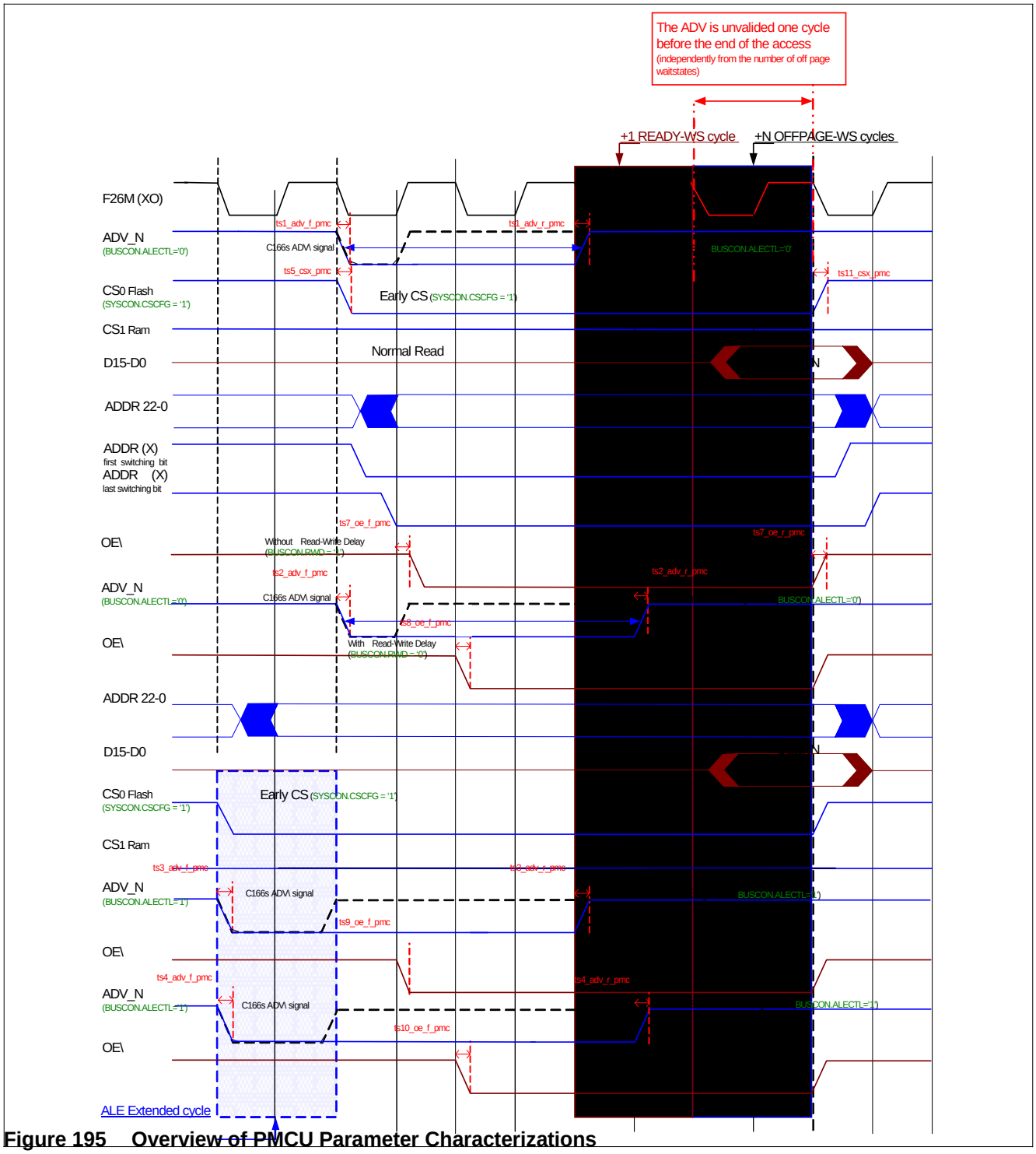


Figure 195 Overview of PMCU Parameter Characterizations

Parameters:

$ts_{1\_adv\_f\_pmc}$ $ts_{1\_adv\_r\_pmc}$ $ts_{5\_cs0\_pmc}$ $ts_{7\_oe\_f\_pmc}$ $ts_{7\_oe\_r\_pmc}$ $ts_{11\_cs0\_pmc}$ are characterized in **Figure** under the following conditions:

- **BUSCON** = $14BF_H$:
  - **BUSCON.RWD** = 1 => without Read Write Delay
  - **BUSCON.EW** = 0 => without Early Write
  - **BUSCON.ALE** = 0 => without Extended ALE
  - **BUSCON.BSW** = 0 => without Tristate inserted when address window changes
  - **BUSCON.RDYEN** = 1 => external bus cycles controlled by Ready signal generated by the PMCU
- **SYSCON** = 7444_H:
  - **SYSCON.CSCFG** = 1 => with Early Chip Select

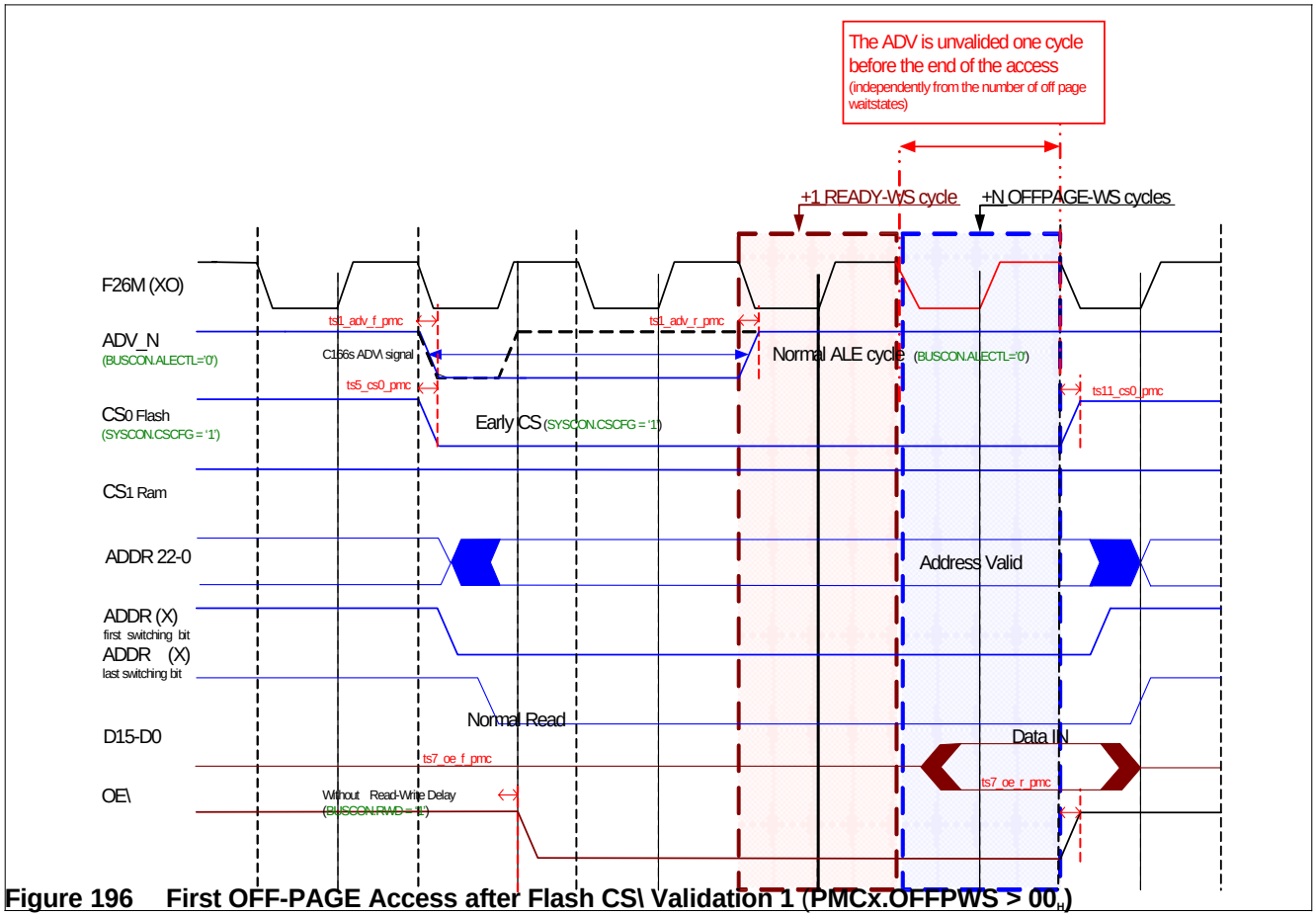


Figure 196 First OFF-PAGE Access after Flash CS\ Validation 1 (PMCx.OFFPWS > 00_H)

Parameters $ts_{19\_1\_cs0\_pmc}$ is characterized in **Figure 197** under the following conditions:

- **BUSCON** = $14BF_H$:
  - **BUSCON.RWD** = 1 => without Read Write Delay
  - **BUSCON.EW** = 0 => without Early Write
  - **BUSCON.ALE** = 0 => without Extended ALE
  - **BUSCON.BSW** = 0 => without Tristate inserted when address window changes
  - **BUSCON.RDYEN** = 1 => external bus cycles controlled by Ready signal generated by the PMCU
- **SYSCON** = 7444_H:
  - **SYSCON.CSCFG** = 1 => with Early Chip Select

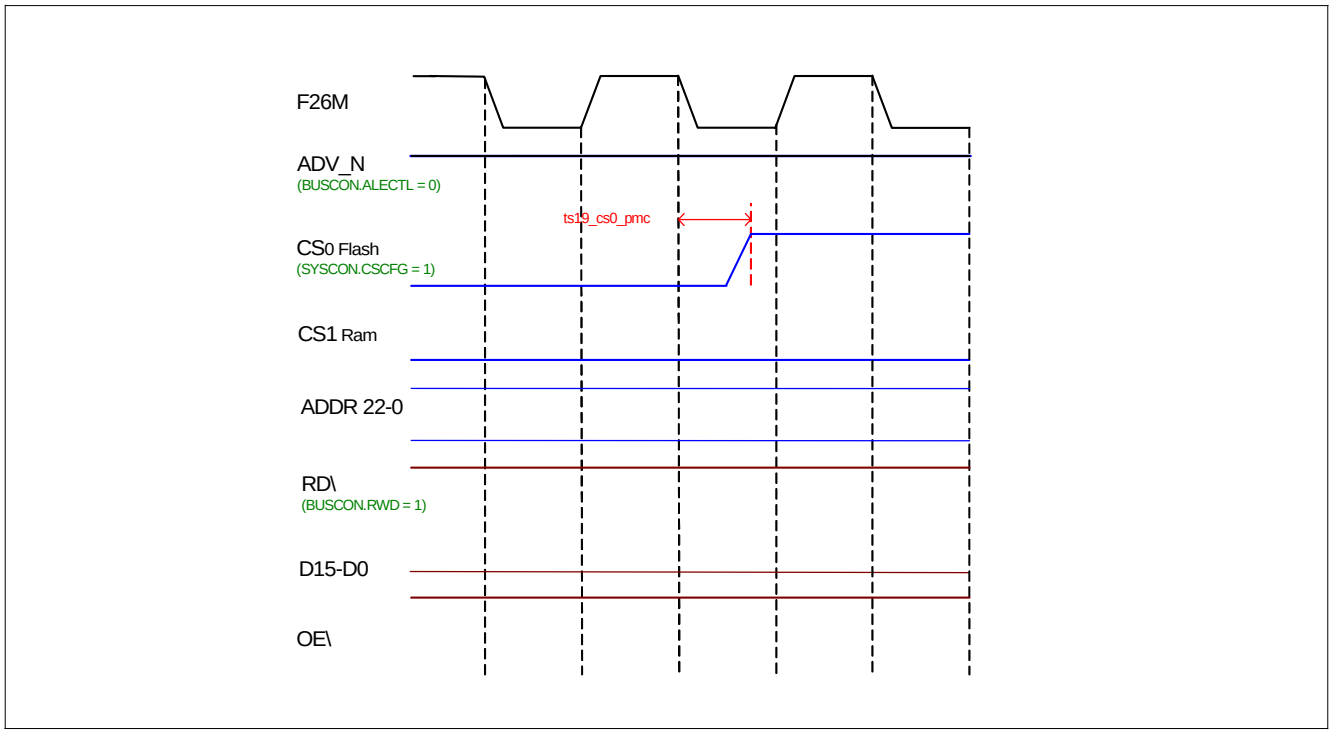


Figure 197 Switch of CS\ from Valid to Invalid (When Timeout of PMC_TIMER is Reached)

Parameters $ts_{2\_adv\_f\_pmc}$ $ts_{2\_adv\_r\_pmc}$ $ts_{8\_oe\_f\_pmc}$ are characterized in **Figure 198** under the following conditions:

- **BUSCON** = $14AF_H$:
  - **BUSCON.RWD** = 0 => with Read Write Delay
  - **BUSCON.EW** = 0 => without Early Write
  - **BUSCON.ALE** = 0 => without Extended ALE
  - **BUSCON.BSW** = 0 => without Tristate inserted when address window changes
  - **BUSCON.RDYEN** = 1 => external bus cycles controlled by Ready signal generated by the PMCU
- **SYSCON** = 7444_H:
  - **SYSCON.CSCFG** = 1 => with Early Chip Select

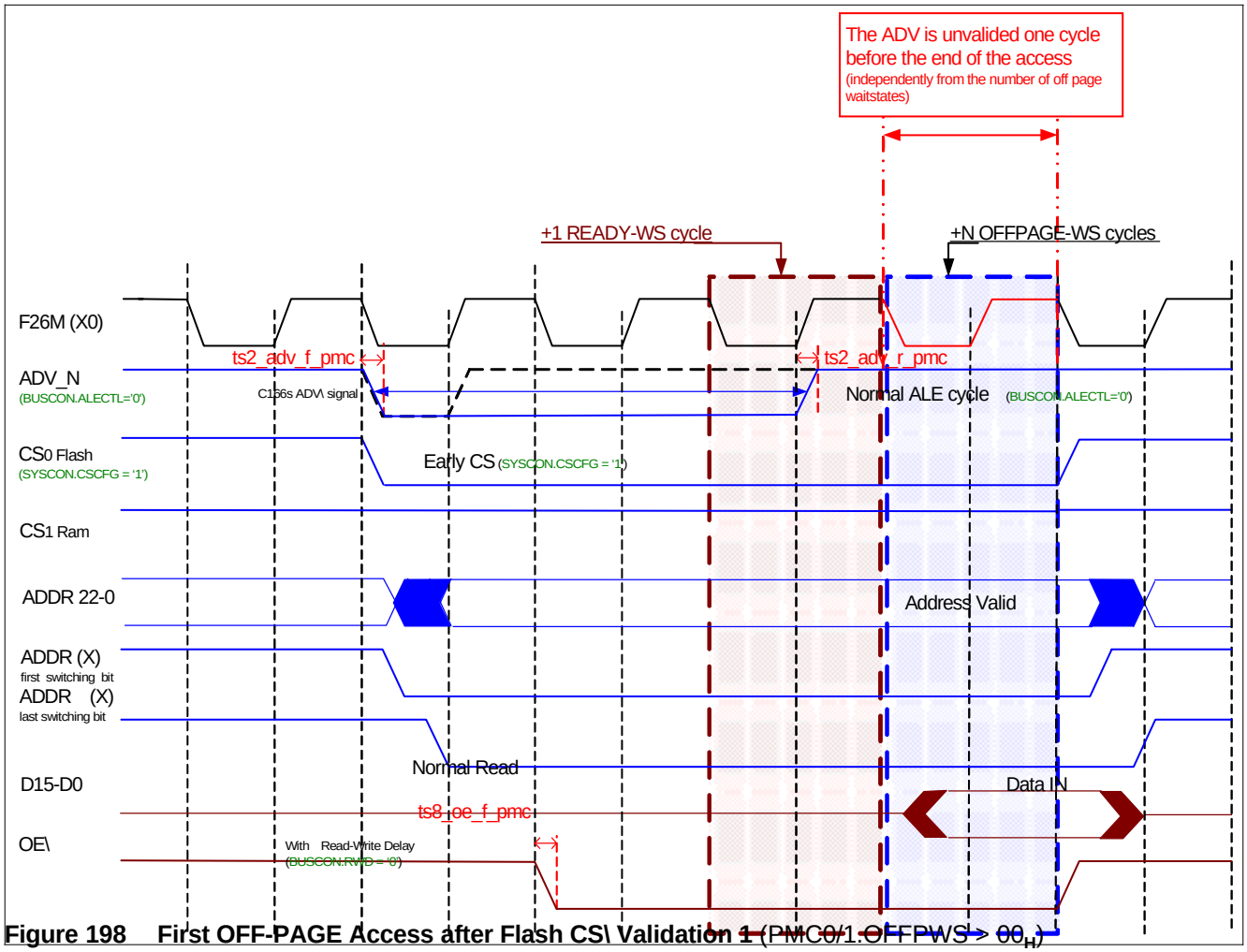


Figure 198 First OFF-PAGE Access after Flash CS\ Validation 1 (PMC0/1.OFFPWS > 00_H)

Parameters:

ts3_adv_f_pmc ts3_adv_r_pmc ts9_oe_f_pmc

are characterized in **Figure 199** under the following conditions:

- **BUSCON** = $16BF_H$:
  - **BUSCON.RWD** = 1 => with Read Write Delay
  - **BUSCON.EW** = 0 => without Early Write
  - ***BUSCON.ALE*** = *1 => with Extended ALE*
  - **BUSCON.BSW** = 0 => without Tristate inserted when address window changes
  - ***BUSCON.RDYEN*** = *1 => external bus cycles controlled by Ready signal generated by the PMCU*
- **SYSCON** = 7444_H:
  - **SYSCON.CSCFG** = 1 => with Early Chip Select

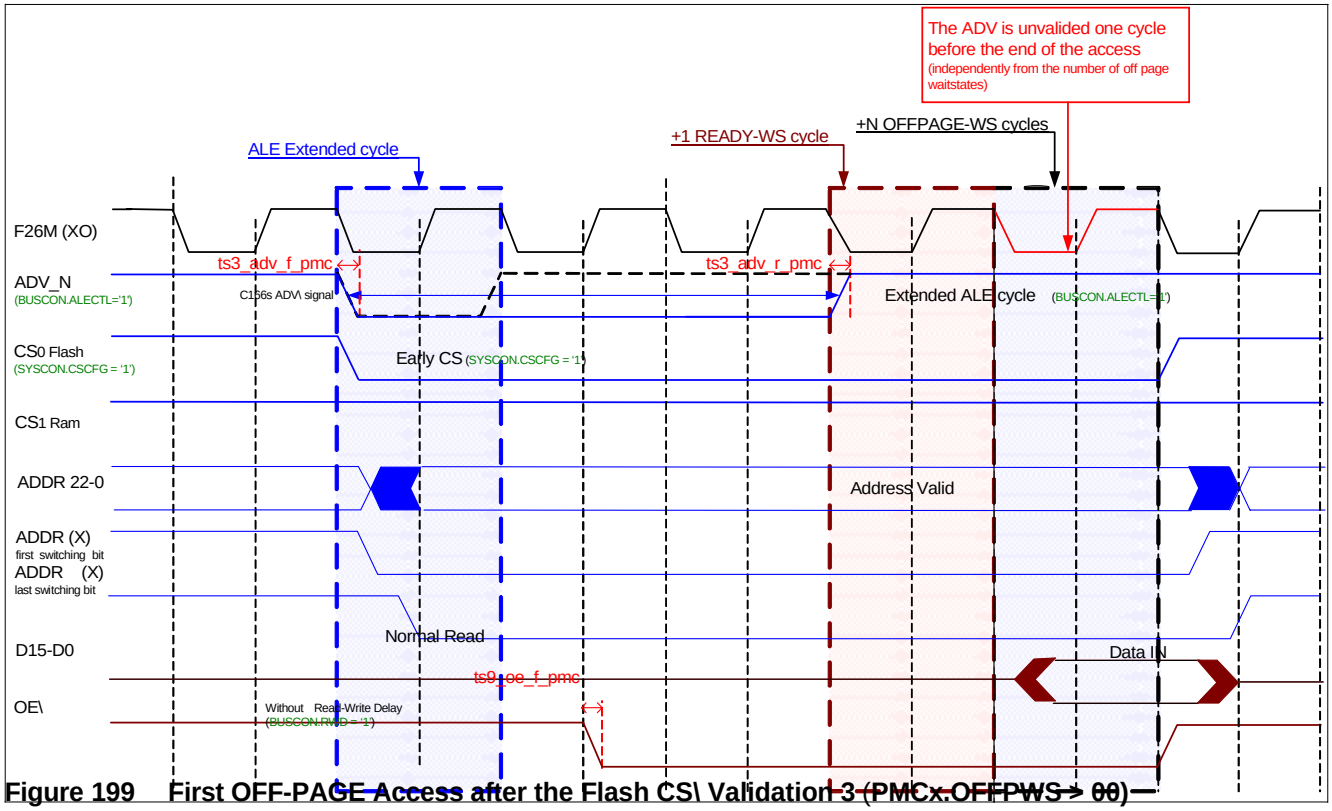


Figure 199 First OFF-PAGE Access after the Flash CS\ Validation 3 (PMCx.OFFPWS > 00)

Parameters:

ts4_adv_f_pmc ts4_adv_r_pmc ts10_oe_f_pmc

are characterized in **Figure 200** under the following conditions:

- **BUSCON** = $16AF_H$:
  - BUSCON.RWD = 0 => without Read Write Delay
  - **BUSCON.EW** = 0 => without Early Write
  - ***BUSCON.ALE*** = *1 => with Extended ALE*
  - **BUSCON.BSW** = 0 => without Tristate inserted when address window changes
  - ***BUSCON.RDYEN*** = *1 => external bus cycles controlled by Ready signal generated by the PMCU*
- **SYSCON** = 7444_H:
  - **SYSCON.CSCFG** = 1 => with Early Chip Select

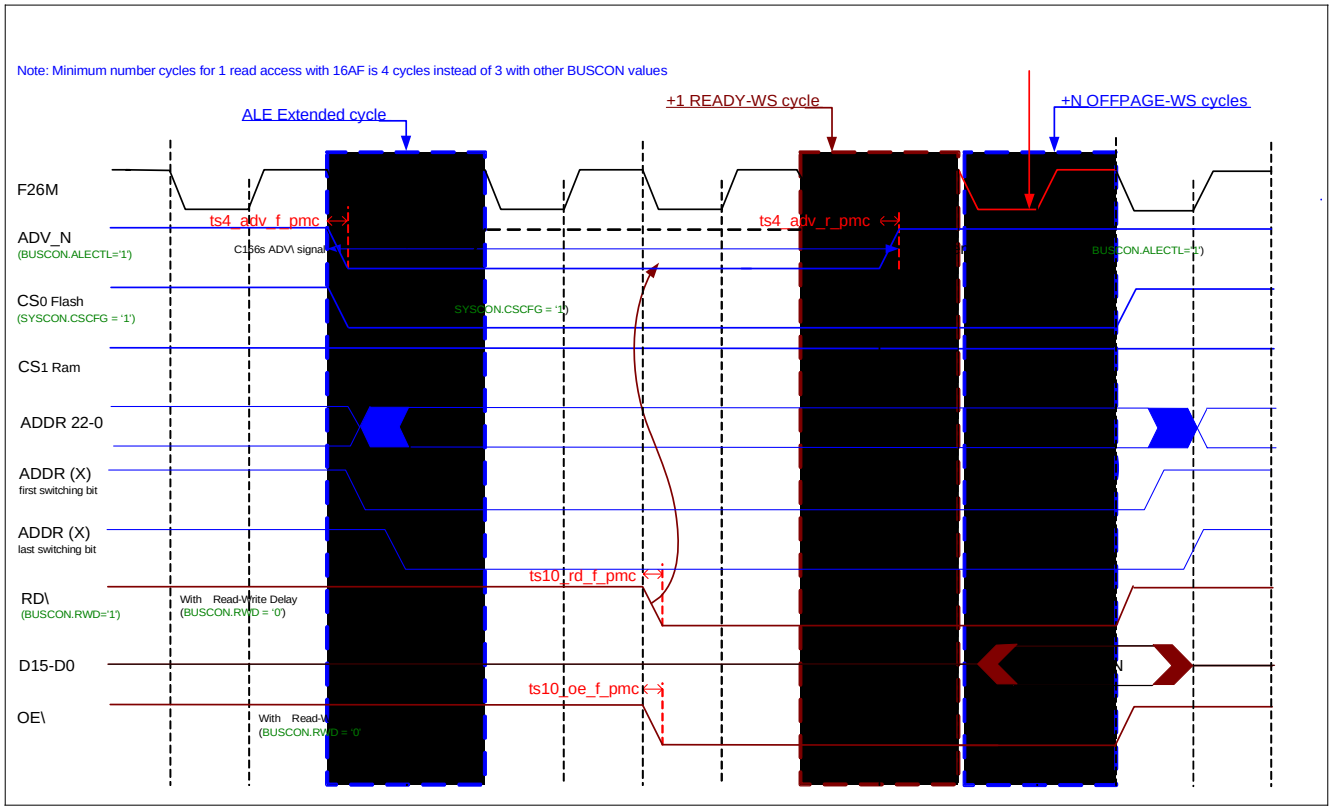


Figure 200 First OFF-PAGE Access after the Flash CS\ Validation 5 (PMCx.OFFPWS > 00_H)

11.2.2.4 SIM

Table 161 SIM Timing

Parameter	Symbol	Limit Values			Unit
		Minimum	Typical	Maximum	
CCLK clock period (output)	t_{i1}		307		ns
CCLK high time	t_{i2}	110		184	ns
CCLK low time	t_{i3}	110		184	ns
CCIO (output) valid after CCLK low end	t_{i4}			200	ns
CCIO (output) still stable after CCLK low end	t_{i5}	30			ns
CCIO (input, 50%) setup before CCLK high end	t_{i6}	45			ns
CCIO (input, 50%) hold after CCLK low begin	t_{i7}	45			ns

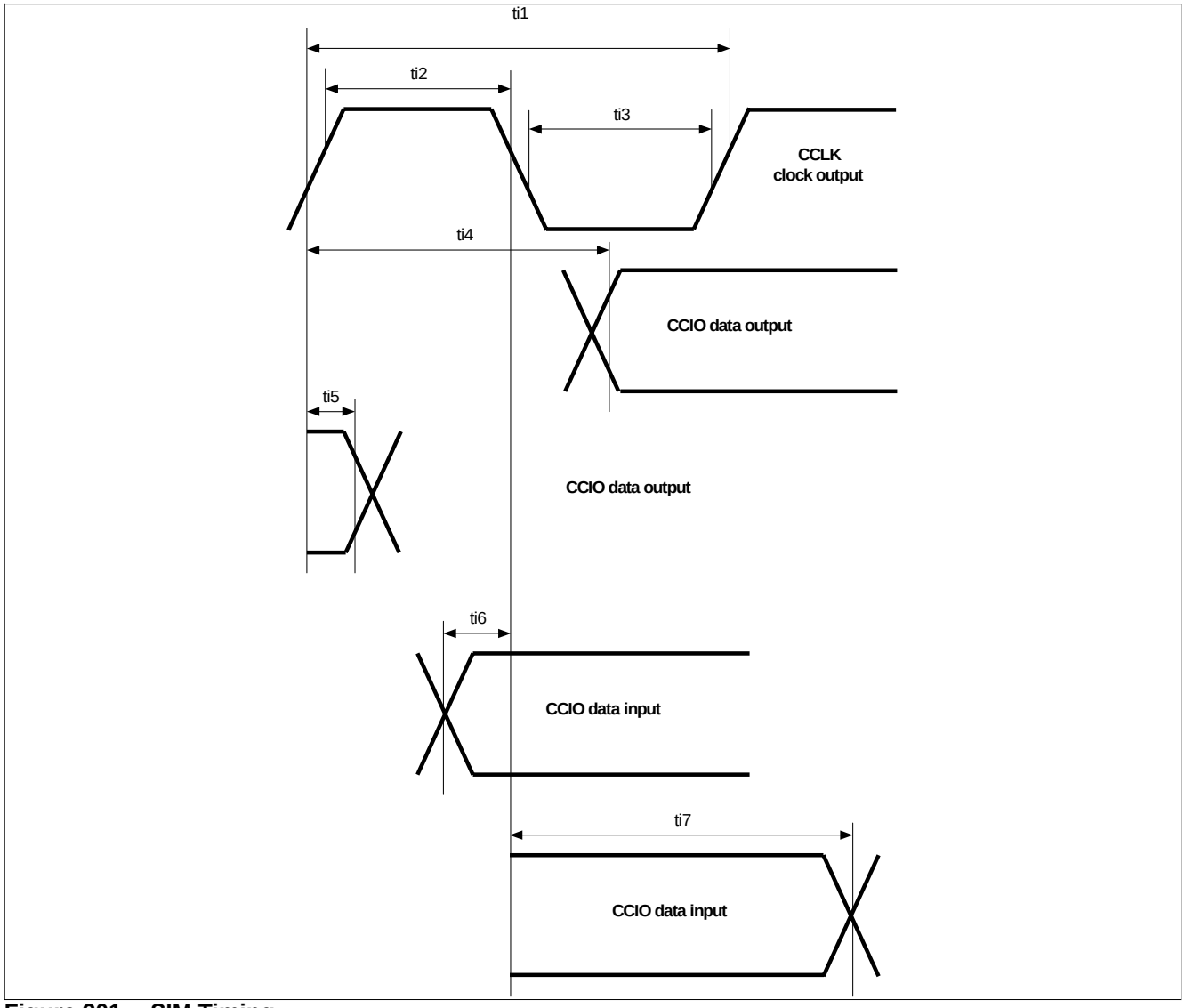


Figure 201 SIM Timing

Note: For information about changing the CCLK frequency, refer to ***Section 7.2.1.2.3 Changing the MCU Sub-System Frequencies During Operation****.*

11.2.2.5 I2C

The I²C bus interface meets the requirements of the STANDARD-MODE and FAST-MODE as described in the I²C-bus specification Version 2.1 as published by Phillips Semiconductors available on the World-Wide-Web. The timing definition of **Figure 202** is equal to the timing definition in Fig. 31 of the Phillips I²C-bus specification.

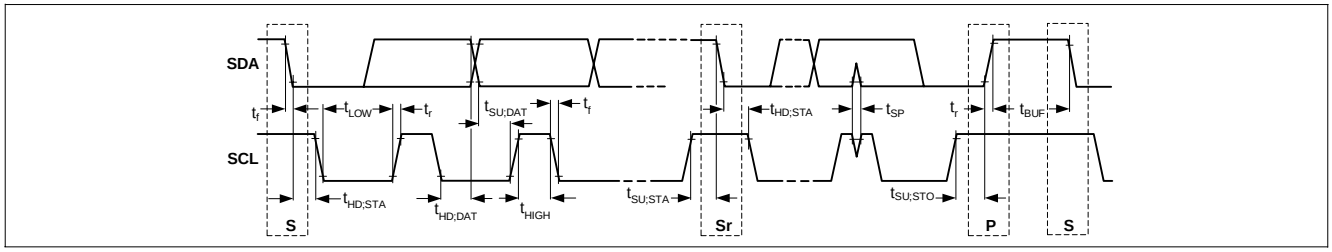


Figure 202 Definition of Timing for F/S-Mode Devices on I²C-Bus

*Note: The timings given in **Table 162** apply only to the special I²C Open Drain pins (Function1 in pin list). They do not apply to pins, where the I²C functionality is used as alternative function (Function 1, 2,3,.... in pin list).*

Table 162 Timing Characteristic of I²C-Bus Interface

Parameter	Symbol	Standard-Mode[1]		Fast-Mode[2]		Unit
		Min.	Max.	Min.	Max.	
SCL clock frequency	f_{SCL}	0	100	0	400	kHz
Hold time (repeated) START condition. After this period, the first clock pulse is generated	$t_{HD;STA}$	4.0	-	0.6	-	∝s
LOW period of the SCL clock	t_{LOW}	4.7	-	1.3[3]	-	∝s
HIGH period of the SCL clock	t_{HIGH}	4.0	-	0.6	-	∝s
Set-up time for a repeated START condition	$t_{SU;STA}$	4.7	-	0.6	-	∝s
Data hold time	$t_{HD;DAT}$	0	3.45	0		
	0.9	∝s Data setup time			$t_{SU;DAT}$	
	250		100		ns	
Rise time of both SDA and SCL signals	t_r	-	1000		150[5]	ns
	27[4]					
Fall time of both SDA and SCL signals	t_f	-	300	27	150	ns
Set-up time for STOP condition	$t_{SU;STO}$	4.0	-	0.6	-	∝s
Bus free time between a STOP and START condition	t_{BUF}	4.7	-	1.3	-	∝s
Capacitive load for each bus line	C_b		100		100	pF

1) Settings of register **IIC_CON**: BRP = 81_H, PREDIV = 01_H, BRPMOD = 0
2) Settings of register **IIC_CON**: BRP = $3F_H$, PREDIV = 08_H, BRPMOD = 1
3) 1.3 ∝s is the Phillips I2C-bus specification, but the E-GOLDvoice value is 1 ∝s.
4) Capacitive load C_b = 70 pF and pull-up resistor R_p = 1 kOhm
5) Capacitive load C_b = 100 pF and pull-up resistor R_p = 9 kOhm

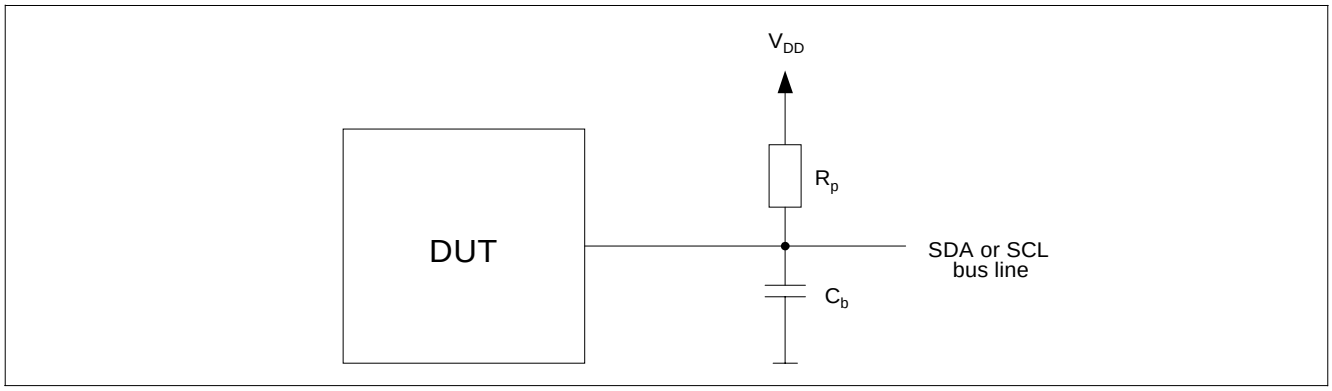


Figure 203 Test circuitry of I2C-bus

11.2.2.6 I2S

This timing characterization is valid for both bi-directional and unidirectional I²S interfaces.

The description of pin alternatives are in **Chapter 3 Pin Descriptions**.

11.2.2.6.1 Normal Mode

Master Mode

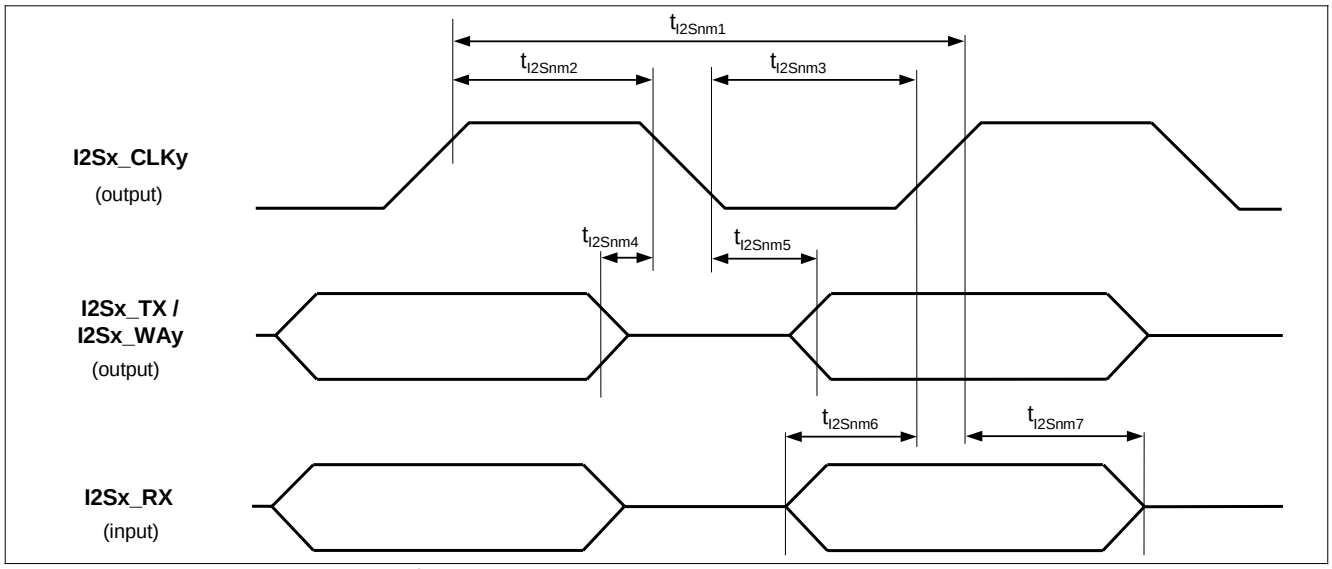


Figure 204 Timing Diagram of I²S Signals in Normal Mode - Master Mode

Table 163 Timing Characteristic of I²S in Normal Mode - Master Mode

Parameter	Symbol	Limit Values			Unit
		Minimum	Typical	Maximum	
I2Sx_CLKy clock period	t_{I2Snm1}	153[1)]			ns
I2Sx_CLKy high time	t_{I2Snm2}	61			ns
I2S_CLKy low time	t_{I2Snm3}	61			ns
I2Sx_TX invalid before I2S_CLK0 high end	t_{I2Snm4}			24	ns
I2Sx_TX valid after I2S_CLK0 low begin	t_{I2Snm5}			24	ns

Table 163 Timing Characteristic of I²S in Normal Mode - Master Mode (cont'd)

I2Sx_RX setup time before I2S_CLK1 low end	t_{I2Snm6}	27			ns
I2Sx_RX hold time after I2S_CLK1 high begin	t_{I2Snm7}	0			

ns

1) Values are characterized for a maximum clock rate of 6.5 MHz

Slave Mode

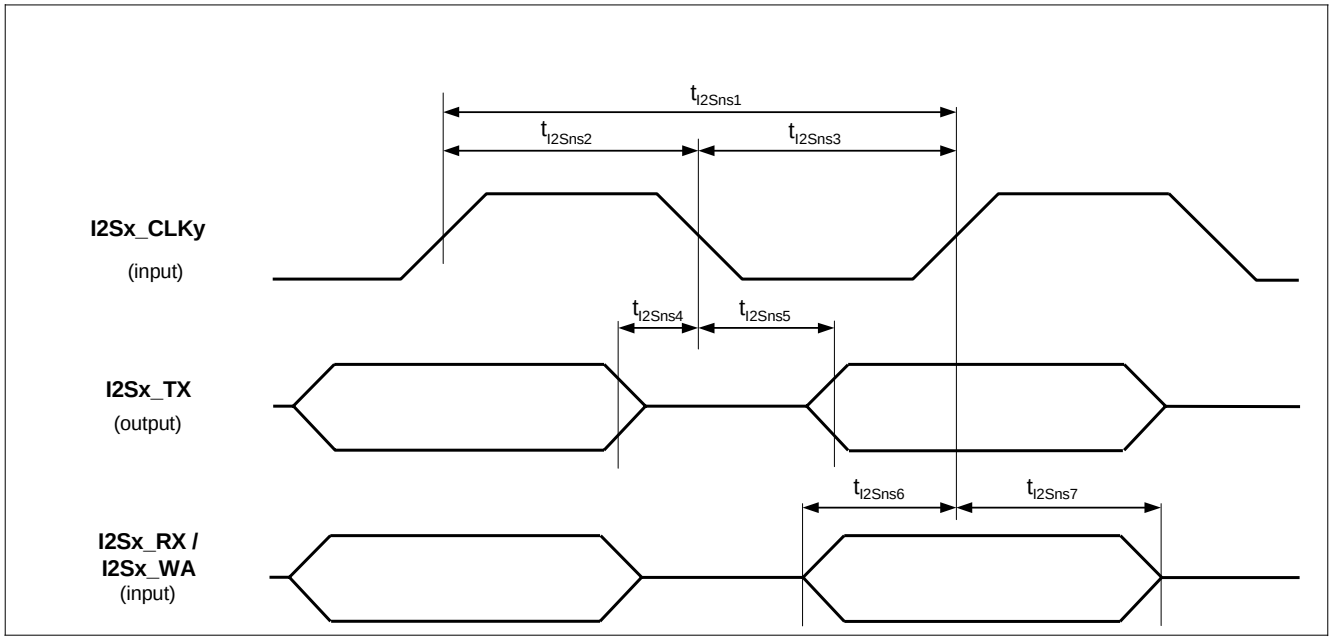


Figure 205 Timing Diagram of I²S Signals in Normal Mode - Slave Mode

Table 164 Timing Characteristic of I²S in normal mode - slave mode

Parameter	Symbol	Limit Values			Unit
		Minimum	Typical	Maximum	
I2Sx_CLKy clock period	t_{I2Sns1}	306[1)]			ns
I2Sx_CLKy high time	t_{I2Sns2}	n.a.			ns
I2S_CLKy low time	t_{I2Sns3}	n.a.			ns
I2Sx_TX invalid before I2S_CLKy falling edge	t_{I2Sns4}			0	ns
I2Sx_TX valid after I2S_CLKy falling edge	t_{I2Sns5}			80	ns
I2Sx_RX setup time before I2S_CLK1 rising edge	t_{I2Sns6}	24			ns
I2Sx_RX hold time after I2S_CLK1 rising edge	t_{I2Sns7}	24			ns

1) Values are characterized for a maximum clock rate of 3.25 MHz

11.2.2.6.2 Burst Mode

The values in **Figure 206** and **Figure 207** are valid for the I²S interface in the burst mode.

Master Mode

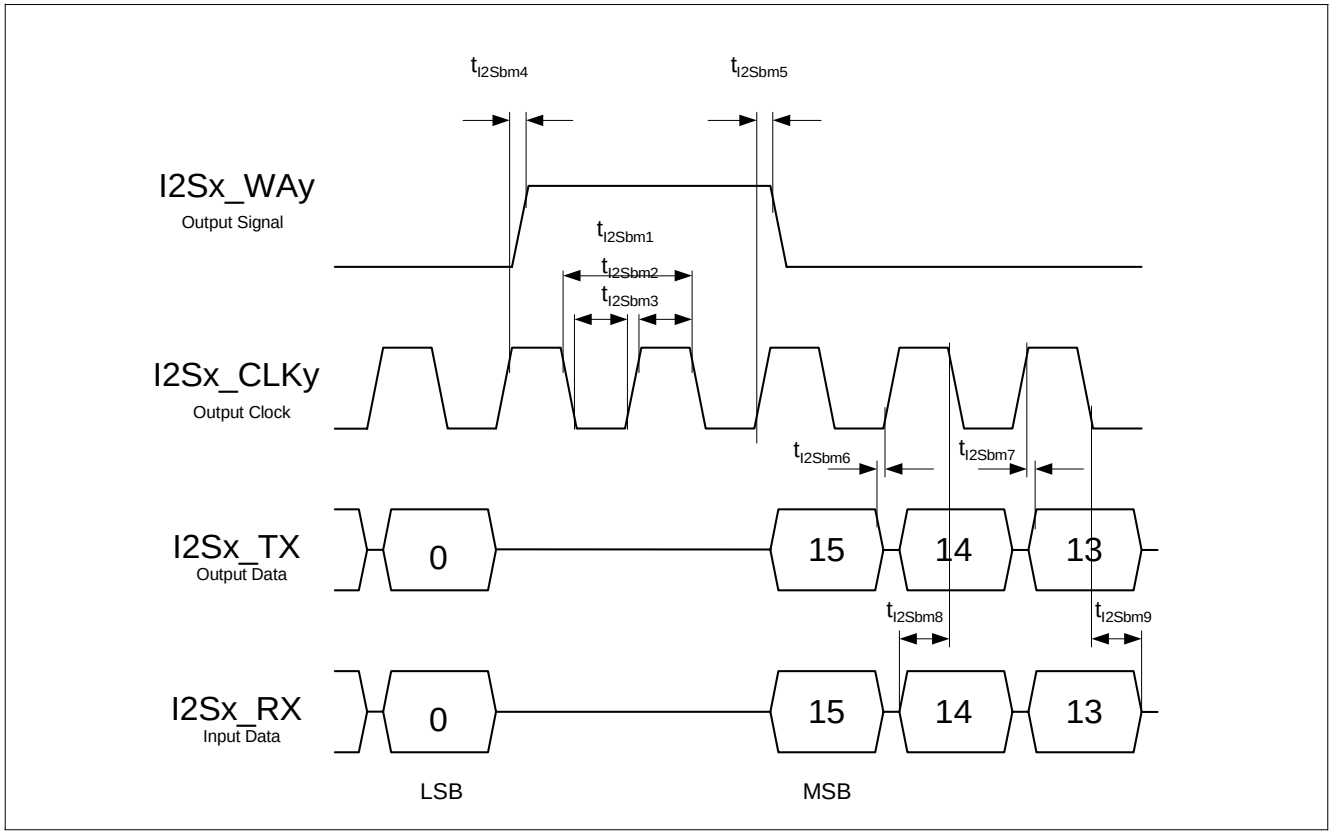


Figure 206 Timing Diagram of I²S Signals in Burst Mode - Master Mode

Table 165 Timing Characteristics of I²S in Burst Mode - Master Mode

Parameter	Symbol	Limit Values			Unit
		Minimum	Typical	Maximum	
I2Sx_CLKy clock period	t_{I2Sbm1}	153[1)]			ns
I2Sx_CLKy low time	t_{I2Sbm2}	61			ns
I2S_CLKy high time	t_{I2Sbm3}	61			ns
I2Sx_CLKy high begin to I2Sx_WAy high begin	t_{I2Sbm4}	-24		24	ns
I2Sx_CLKy low end to I2Sx_WAy high end	t_{2Sbm5}	-24		24	ns
I2Sx_TX invalid before I2S_CLK0 low end	t_{I2Sbm6}			24	ns
I2Sx_TX valid after I2S_CLK0 high begin	t_{I2Sbm7}			24	ns
I2Sx_RX setup time before I2S_CLK1 high end	t_{I2Sbm8}	27			ns
I2Sx_RX hold time after I2S_CLK1 low begin	t_{I2Sbm9}	0			

ns

1) Values are characterized for a maximum clock rate of 6.5 MHz

Slave Mode

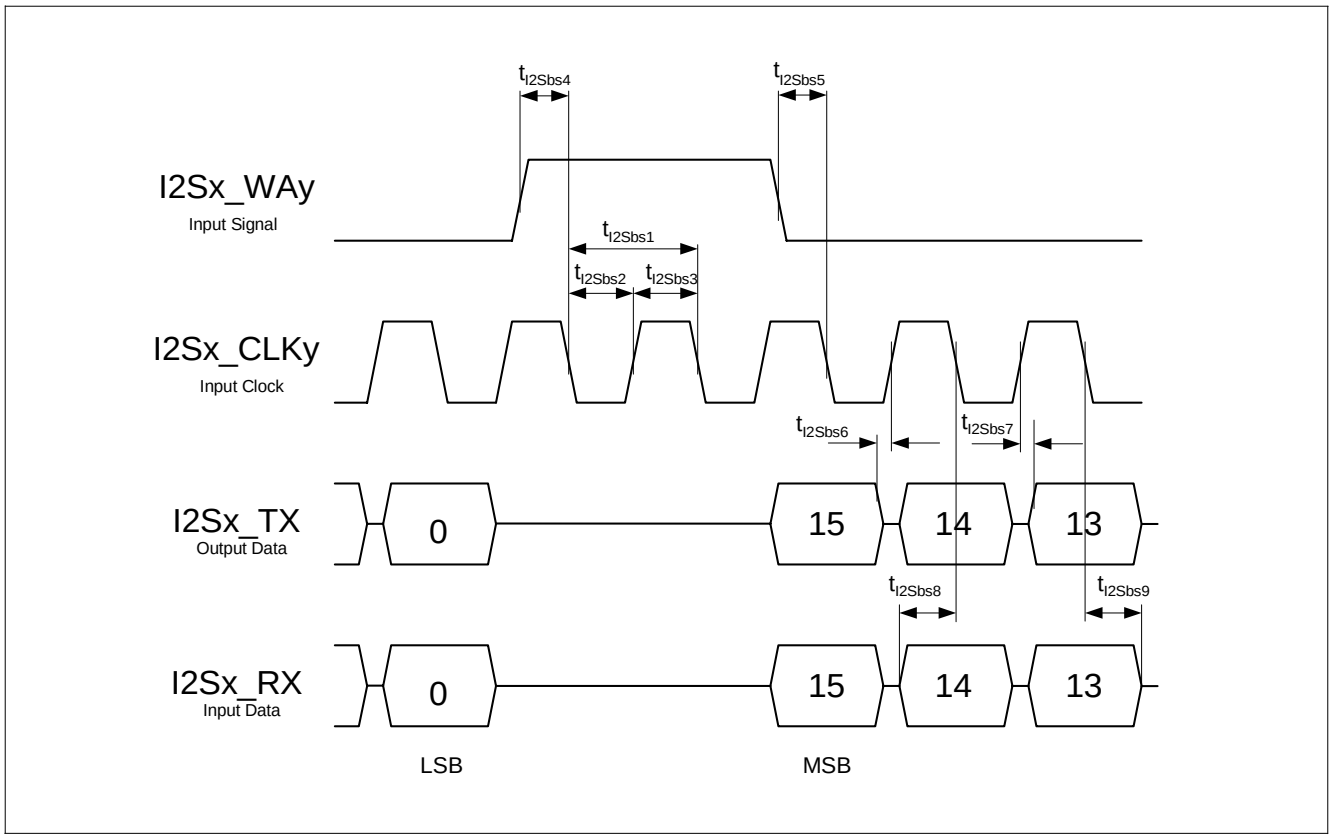


Figure 207 Timing Diagram of I²S Interface Signals in Burst Mode - Slave Mode

Table 166 Timing Characteristics of I²S in Burst Mode - Slave Mode

Parameter	Symbol	Limit Values			Unit
		Minimum	Typical	Maximum	
I2Sx_CLKy clock period	t_{I2Sbs1}	306[1)]			ns
I2Sx_CLKy low time	t_{I2Sbs2}	n.a.			ns
I2S_CLKy high time	tt_{I2Sbs3}	n.a.			ns
I2Sx_CLKy low begin to I2Sx_WAy high begin	t_{I2Sbs4}	36			ns
I2Sx_WAy low begin to I2Sx_CLKy low begin	t_{I2Sbs5}	36			ns
I2Sx_TX invalid before I2S_CLKy rising edge	t_{I2Sbs6}			0	ns
I2Sx_TX valid after I2S_CLKy rising edge	t_{I2Sbs7}			80	ns
I2Sx_RX setup time before I2S_CLK1 falling edge	t_{I2Sbs8}	24			ns
I2Sx_RX hold time after I2S_CLK1 falling edge	t_{I2Sbs9}	24			ns

1) Values are characterized for a maximum clock rate of 3.25 MHz

11.2.2.6.3 DAI Mode

Figure 208 and Table 167 shows the timing requirements for a DAI interface according to the GSM standard. These requirements are met by design. Thus no electrical characterization is necessary.

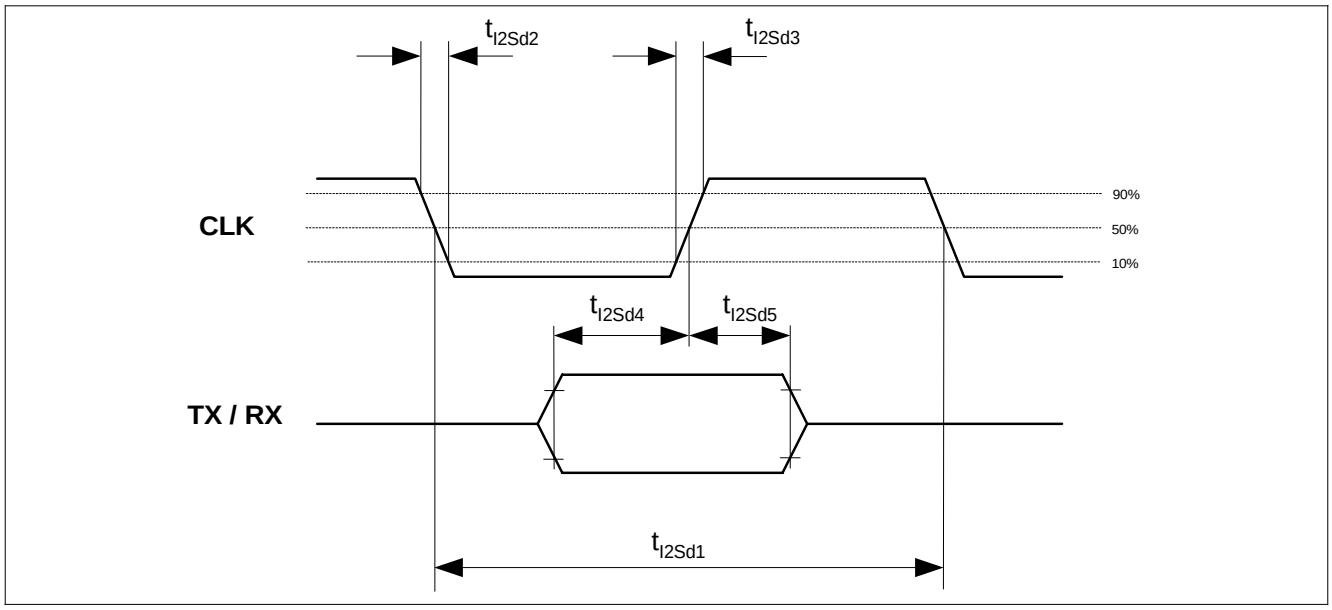


Figure 208 Timing Diagram of I²S Interface Signals in DAI Mode

Table 167 Timing Characteristics of I²S in DAI Mode

Parameter	Symbol	Limit Values			Unit
		Minimum	Typical	Maximum	
Clock period time[1)]	t_{I2Sd1}	9.61520	9.61538	9.61557	us
Clock falling edge time	t_{I2Sd2}			1	us
Clock rising edge time	t_{I2Sd3}			1	us
Data setup time before rising edge of CLK	t_{I2Sd4}	3			us
Data hold time after rising edge of CLK	t_{I2Sd5}	1			us

1) The clock frequency needs to be 104 kHz +/- 20 ppm

11.2.2.7 ASC

The following timing characteristic is valid for the ASC0, they apply only in the synchronous mode.

Note: In the asynchronous mode there is no timing given, since there is only one asynchronous transmission data line and one asynchronous reception data line. For the timing of the TX signal, the rise and fall times of the driver used must be considered.

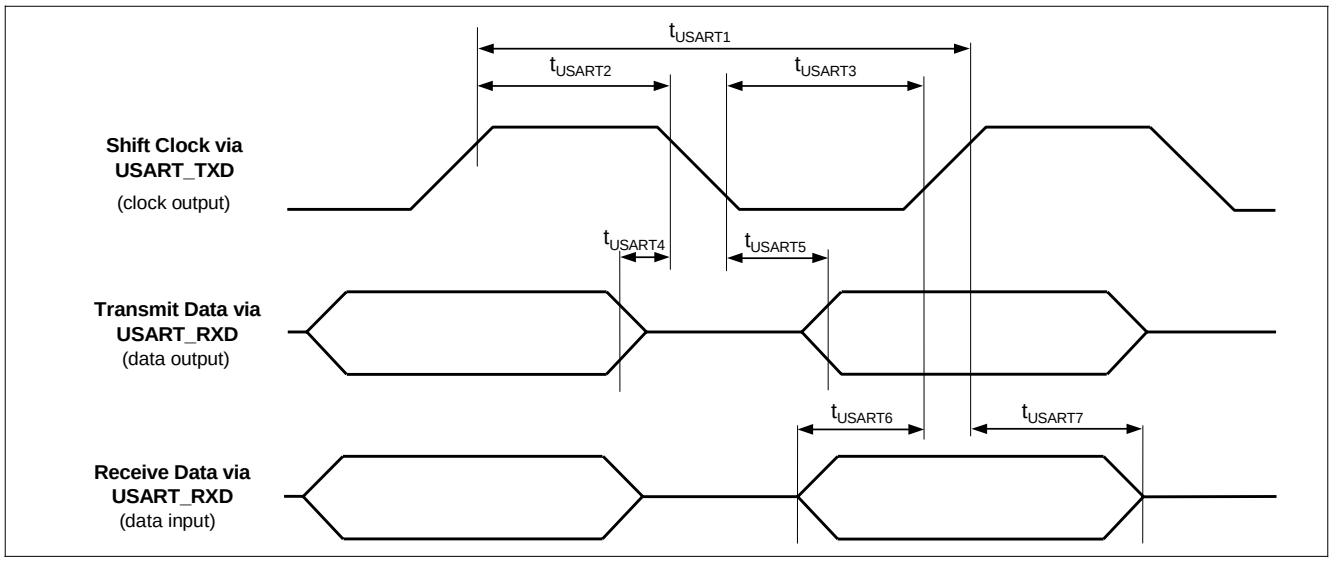


Figure 209 Timing of ASC in Synchronous Mode

Table 168 Timing Characteristic of ASC with D Drivers in Synchronous Mode

Parameter	Symbol	Limit Values			Unit
		Minimum	Typical	Maximum	
Shift clock period	t_{USART1}	150[1)]			ns
Shift high time	t_{USART2}	50			ns
Shift clock low time	t_{USART3}	50			ns
Transmit data invalid before shift clock high end	t_{USART4}			10	ns
Transmit data valid after shift clock low begin	t_{USART5}			10	ns
Receive data setup time before shift clock low end	t_{USART6}	50			ns
Receive data hold time after shift clock high begin	t_{USART7}	0			ns

1) Maximum characterized baud rate of ASC with D drivers in synchronous mode is 6.5 Mbps

Note: The USARTs in alternative functions with E drivers should only be used in asynchronous mode. Timings for synchronous modes are not guaranteed in this case.

11.2.2.8 SSC

11.2.2.8.1 Master Mode

In the master mode the maximum programmable baud rates of the MCU module is 26 MBaud and 44.55 MBaud, respectively. But the actual achievable baud rates are limited by the following timing characteristics.

Note: The settings for this specification are:

- SSC operates in master mode
- Idle clock line is high, the leading clock edge is high-to-low => **SSCPD_CON.PO** = 1
- Shift transmit data on the leading edge, latch on trailing edge => **SSCPD_CON.PH** = 0.

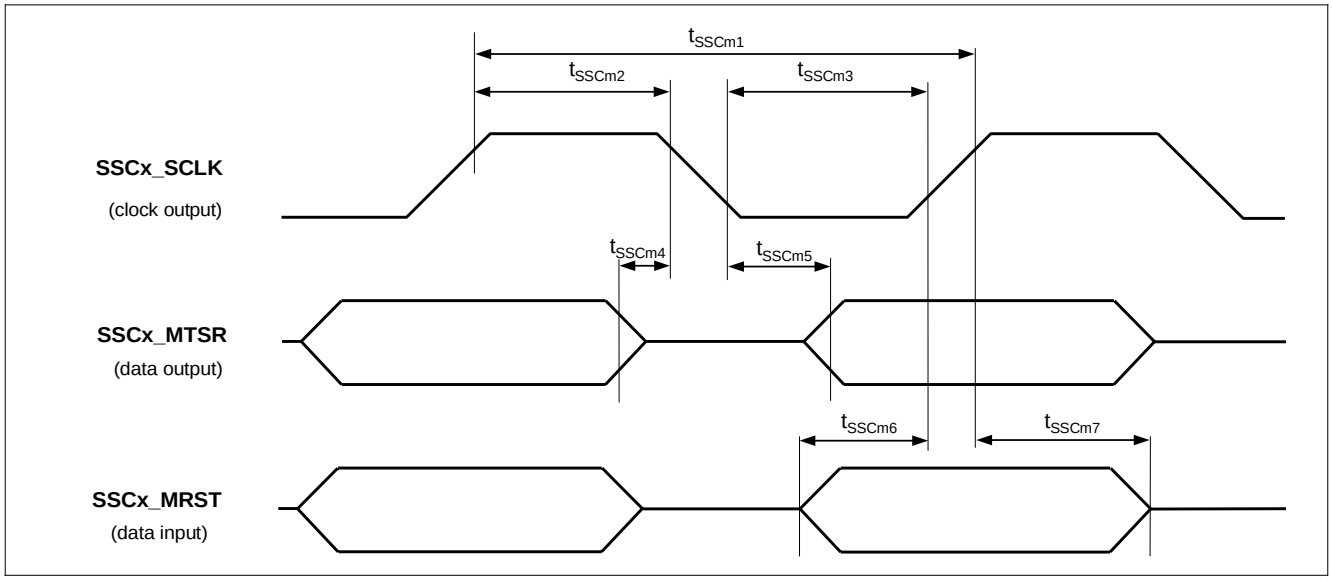


Figure 210 SSC Interface Timing in Master Mode

Table 169 Timing Characteristic of SSC in Master Mode

Parameter	Symbol	Limit Values			Unit
		Minimum	Typical	Maximum	
SCLK clock period	t_{SSCm1}	38			ns
SCLK high time	t_{SSCm2}	n.a.[1)]			ns
SCLK low time	t_{SSCm3}	n.a.			ns
MTSR invalid before SCLK high end	t_{SSCm4}			5	ns
MTSR valid after SCLK low begin	t_{SSCm5}			7	ns
MRST setup time before SCLK low end for 6.5MHz internal clock	t_{SSCm6} for MCU	165			ns
MRST setup time before SCLK low end for 13MHz internal clock		85			ns
MRST setup time before SCLK low end for 26MHz internal clock		45			ns
MRST setup time before SCLK low end for 52MHz internal clock		25			ns
MRST hold time after SCLK high begin	t_{SSCm7}	0			ns

1) The duty cycle of SCLK is 50%. Thus the resulting high and low times are by design one half of the clock period minus the slope. The slope depends on the driver class of the used pad (see **Chapter 3 Pin Descriptions**).

11.2.2.8.2 Slave Mode

In slave mode the maximum programmable baud rates of the MCU module is 13 MBaud and 22.25 MBaud, respectively. But the actual achievable baud rates are limited by the following timing characteristics.

Note: The settings for this specification are:

- SSC operates in slave mode
- Idle clock line is high, the leading clock edge is high-to-low => **SSCPD_CON.PO** = 1
- Shift transmit data on the leading edge, latch on trailing edge => **SSCPD_CON.PH** = 0.

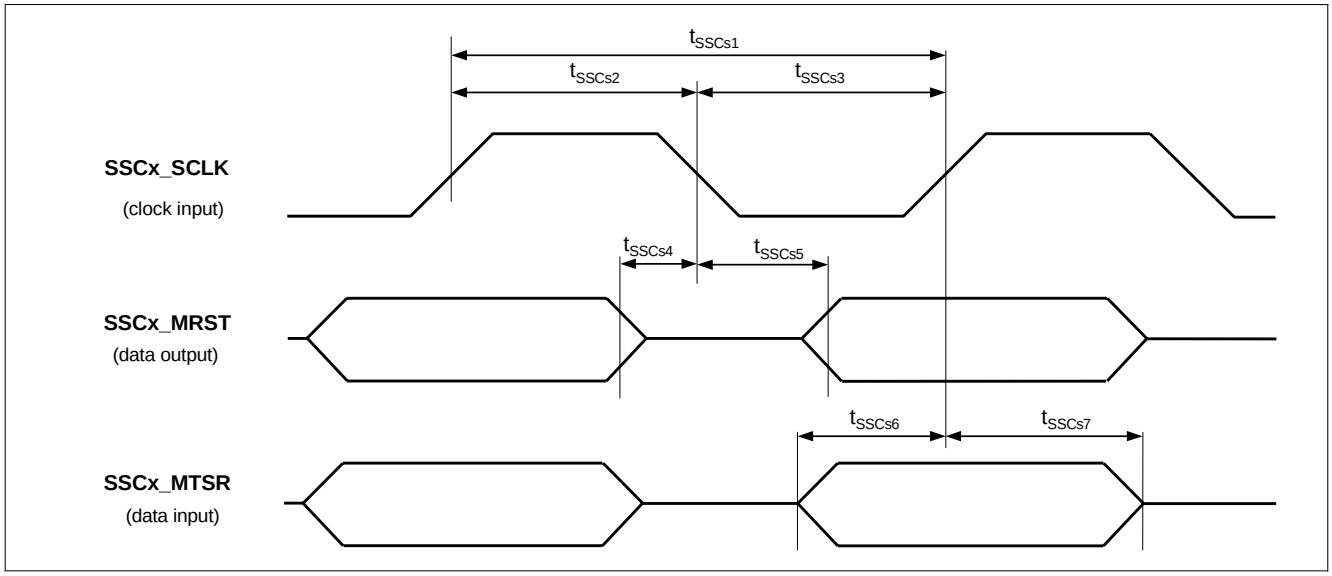


Figure 211 SSC Interface Timing in Slave Mode

Table 170 Timing Characteristic of SSC in Slave Mode

Parameter	Symbol	Limit Values			Unit
		Minimum	Typical	Maximum	
SCLK clock period	t_{SSCs1}	76.9			ns
SCLK high time	t_{SSCs2}	n.a.[1)]			ns
SCLK low time	t_{SSCs3}	n.a.			ns
MRST invalid before SCLK falling edge	t_{SSCs4}			0	ns
MRST valid after SCLK falling edge (with C/D/E driver)	t_{SSCs5}			32/42/45	ns
MTSR setup time before SCLK low end for 6.5 MHz internal clock	t_{SSCS6} for **MCU**	165			ns
MTSR setup time before SCLK low end for 13 MHz internal clock		85			ns
MTSR setup time before SCLK low end for 26 MHz internal clock		45			ns
MTSR setup time before SCLK low end for 52 MHz internal clock		25			ns
MTSR hold time after SCLK high begin	t_{SSCs7}	5			ns

1) The duty cycle should be approximately 50%. The minimum values of t_{SSCs2} and t_{SSCs3} are basically constrained by the values of $t_{SSCs2...7}$.

11.2.2.9 RTC and 32 kHz Pad Oscillator

11.2.2.9.1 Timing of RTC

Table 171 gives the timing for the optional standby clock that is used with the PMB7880.

Table 171 Standby Clock Timings

Parameter	Symbol	Limit Values			Unit
		Minimum	Typical	Maximum	
Input clock period f32k	f_{32k}	18	32	55	kHz
Input clock high time f32k	t_{H32k}	8			∝s
Input clock low time f32k	t_{L32k}	8			∝s

The duty cycle must be between 45%:55% and 55%:45%

11.2.2.9.2 External Circuitry and Recommended Crystal

For using the internal oscillator of the PMB7880 together with an external crystal the crystal parameters load capacitance C_L, static capacitance C_0 and internal serial resistance R_s, the parameters C_1 and C_2 of **Figure 145 Circuitry to Use the Internal 32 kHz Oscillator (on Page 381)** and the board capacitances must be related as follows.

The real part of the amplitude impedance (= negative resistive component of input impedance between F32k and OSC32K) of the 32k on chip oscillator is R_{int} = -240 kOhm (that is, $|R_{int}|$ ε 240 kOhm) . It is strongly recommended to use a crystal with a maximal serial resistance of:

$$R_{s\_max} < (-R_{int} / 3) = 80 \text{ kOhm} \quad (67)$$

Further it is recommended to choose the typical value of R_{s_typ}:

$$R_{s\_typ} < (-R_{int}/5) \quad (68)$$

Values of R_{s_max} above 80 kOhm have to be verified in the application and are not guaranteed.

The parameter C_L must have the following range:

$$9 \text{ pF} < C_L < 12.5 \text{ pF} \quad (69)$$

The external load capacities must be kept in the range of:

$$17 \text{ pF} < C_1 + C_{1b} < 20.5 \text{ pF} \quad (70)$$

$$17 \text{ pF} < C_2 + C_{2b} < 20.5 \text{ pF} \quad (71)$$

The relation between these values is given by:

$C_{1t} * C_{2t} / (C_{1t} + C_{2t}) + C_{io} = C$ (72)

Where:

- $C_{1t} = C_1 + C_{1b} + C_{1p}$; 1pF < C_{1p} < 2.5 pF
  - C_{1p} = capacitance from pin OSC32k to ground added by the package and internal circuitry;
  - C_{1b} = capacitance from pin OSC32K to ground added by the board-design
- $C_{2t} = C_2 + C_{2b} + C_{2p}$; 1pF < C_{2p} < 2.5 pF
  - C_{2b} = capacitance from pin F32K to ground added by the package and internal circuitry;
  - C_{2p} = capacitance from pin F32K to ground added by the board-design
- $C_{1t} = C_{2t}$
- $C_{io} = C_0 + C_{xb} + C_{xp}$; 0,05 pF < C_{xp} 0,5 pF
  - C_{xp} = capacitance parallel to the crystal added by the package and internal circuitry;
  - C_{xb} = capacitance parallel to the crystal added by the board-design

Note: The value C_{io} (and thus C_{xb}) should be as small as possible to reduce the startup time. That is, the external crystal should be connected as close as possible to the PMB7880.

Note: A 32 kHz jitter problem has been demonstrated mainly related to the PCB layout. ***To avoid this 32 kHz jitter, the external crystal must be protected by shielding.***

11.2.2.10 Output Clock CLKOUT

Table 172 Output Clock CLKOUT

Parameter	Symbol	Limit Values			Unit
		Minimum	Typical	Maximum	
CLKOUT period at 13 MHz	t_{r1}		77		ns
CLKOUT high time at 13 MHz	t_{r2}	26			ns
CLKOUT low time at 13 MHz	t_{r3}	26			ns

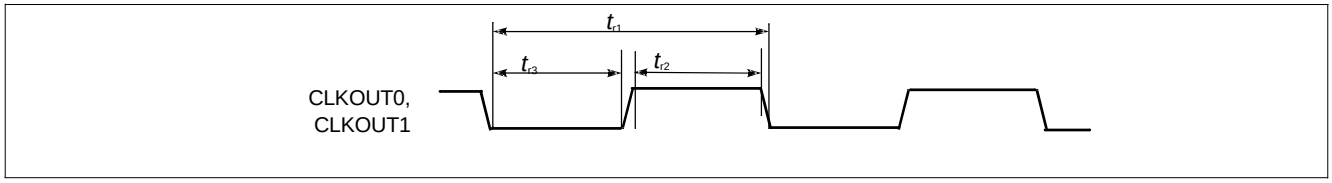


Figure 212 Output Clocks: CLKOUT Timing

11.3 Mixed Signals and Other Values

11.3.1 Audio

11.3.1.1 Audio Receive Part

Note: If not specified otherwise, all parameters are measured with a bandwidth of 20 Hz,...,20 kHz and gain setting gs = 0 dB.

Table 173 General Electrical Characteristics of Audio Receive Path

Parameter	Symbol	Limit Values			Unit	Remark
		Minimum	Typical	Maximum		
Maximal differential output voltage		3.3	3.7	4.1	Vpp	Full scale differential open circuit voltage. Valid for EPp1/EPn1
Maximal single-ended output voltage		1.65	1.85	2.05	Vpp	Full scale single-ended open circuit voltage. Valid for EPpa1
Current consumption of audio receive path				12	mA	without any external load
Output differential DC offset				+/- 50	mV	
Output load resistance for EPPa1 (Headset)			16		W	
Output load resistance for EPp1/EPn1 (Earpiece)			8			W
Signal to noise	S/N	70	80		dB	for R_L=16 &, gs=0 dB for headset driver, gs=-12 dB for earpiece driver, with inputsignal 0dBFS, code 0, A-weighted
	S/N	70	80		dB	for R_L=8 &, gs=-18 dB for earpiece driver, with input signal 0dBFS, code 0, A-weighted
Idle channel noise			-90	-80	dBFS	code 0, A-weighted (for Loudspeaker / Earpiece)
Signal to distortion[1)]	THD	60	70		dB	for R_L=16 &, gs=0 dB for headset driver, gs=-12 dB for earpiece driver, with input signal 0dBFS
	THD	60	70		dB	for R_L=8 &, gs=-18 dB for earpiece driver, with input signal 0dBFS
Signal to distortion	THD	60	70		dB	for R_L=16 &, gs=0dB for headset driver, gs=-12 dB for earpiece driver, with input signall -1dBFS
	THD	60	70		dB	for R_L=8 &, gs=-18 dB for earpiece driver, with input signal -1dBFS

Table 173 General Electrical Characteristics of Audio Receive Path (cont'd)

Parameter	Symbol	Limit Values			Unit	Remark
		Minimum	Typical	Maximum		
Signal to distortion	THD	60			dB	for R_L=16 &, gs=0 dB for headset driver, gs=-12 dB for earpiece driver, with input signal -6dBFS
	THD	60			dB	for R_L=8 &, gs=-18 dB for earpiece driver, with input signal -6dBFS
Signal to distortion	THD		60		dB	for R_L=16 &, gs=0 dB for headset driver, gs=-12 dB for earpiece driver, with input signal 0dBFS
	THD		60		dB	for R_L=8 &, gs=-18 dB for earpiece driver, with input signal 0dBFS
Signal to distortion	THD	50			dB	For loudspeaker driver 350 mW at RL=8 &
Power supply rejection - LOUD1/LOUD2	PSR	60	66		dB	$U_{VDDv}(t)$ = 2.5 V + 0.15 V sin(2□•1 kHz•t) and gs = 0 dB
Power supply rejection - EPPA1	PSR	60	66		dB	$U_{VDDv}(t)$ = 2.5 V + 0.15 V sin(2□•1 kHz•t) and gs = 0 dB
Power supply rejection - EPP1/EPN1	PSR	50	56		dB	$U_{VDDv}(t)$ = 2.5 V + 0.15 V sin(2□•1 kHz•t) and gs = 0 dB
Cross talk (between receive and transmit channel)				-65	dB	$U_{TX}(t)$ = 1.075 V + 0.15 V sin(2□•1 kHz•t) $U_{RX}(t)$ = 0.15 V sin(2□•33 kHz•t)
Passband ripple				0.5	dB	f < 0.45fs
Stopband attenuation		50			dB	f > 0.55fs
Absolute gain drift				+/- 2	%	Variation due to change in VDD, temperature and life time

1) THD is min. 50 dB for all drivers (i.e. Headset, Earpiece and Loudspeaker) on at the same time

Table 174 Characteristics of Audio Rx Path for Headset Driver EP_{pa1}

Parameter	Symbol	Limit Values			Unit	Remark
		Minimum	Typical	Maximum		
Maximal single-ended output voltage		1.65	1.85	2.05	Vpp	Full scale single-ended open circuit voltage
Maximal output current				200	mA	
Internal output resistance			1.7	4	W	
Single-ended output load capacitance				10	nF	
Single-ended output load capacitance					∝F	Between output pins and GND with & series resistance

Table 175 Characteristics of Audio Rx Path for differential Earpiece Drivers EP_{p1} and EP_{n1}

Parameter	Symbol	Limit Values			Unit	Remark
		Minimum	Typical	Maximum		
Maximal differential output voltage		3.3		3.7 4.1		Vppdif Full scale differential open circuit voltage
Differential output voltage			925		mVpp dif	at load resistance R_L = 16 & and -12dBFS
Differential output voltage			462.5		mVpp dif	at load resistance R_L = 8 & and -12dBFS
Maximal output current				64	mA	
Internal output resistance			4		W	
Single-ended output load capacitance				250	pF	Between output pins and GND
Single-ended output load capacitance					∝F	Between output pins and GND with & series resistance

Table 176 Characteristics of Audio Rx Path for Loudspeaker LOUD1, LOUD2

Parameter	Symbol	Limit Values			Unit	Remark
		Minimum	Typical	Maximum		
Maximal single-ended output voltage					Vpp	Full scale differential open circuit voltage
Maximal output current					mA	
Internal output resistance					&	
Single-ended output load capacitance				10	nF	
Inductive load				400	∝H	Between output pins and GND with & series resistance

Table 177 Electrical characteristics of Ringer Support

Parameter	Symbol	Limit Values			Unit	Test Condition/ Remark
		Minimum	Typical	Maximum		
Differential output voltage in **RING** Mode	V_r		2 Vdd		V	

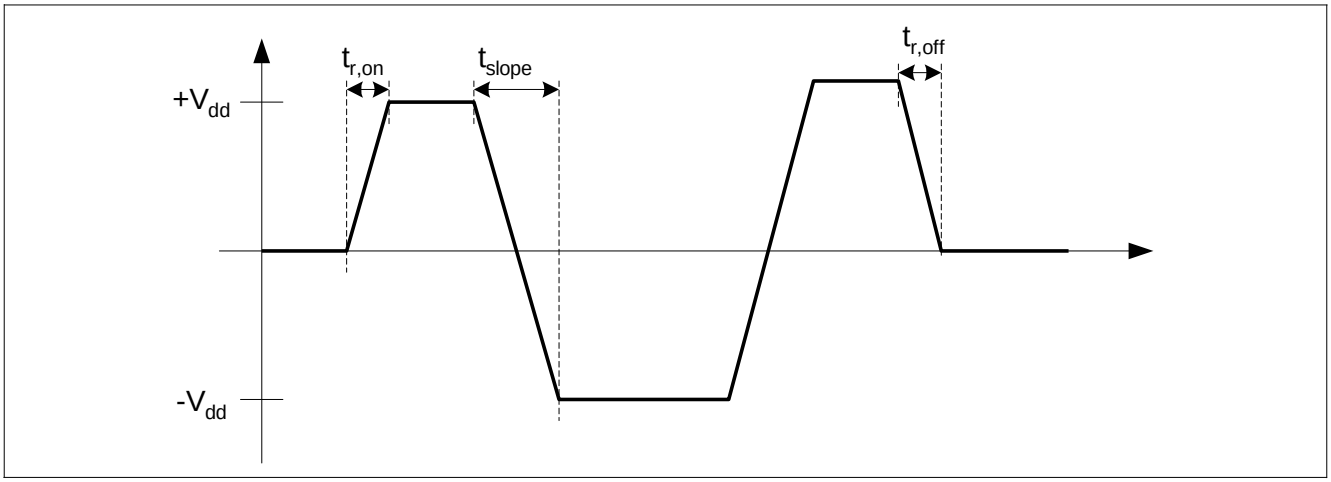


Figure 213 Timing Diagram for Analog Ringer Mode

Table 178 Timing Parameters of Ringer Output

Parameter	Symbol	Limit Values		
		Minimum	Typical	Maximum
Rise time for first slope		11.25		∝s
Rise and fall times for slopes during ringer operation		22.5		∝s
Fall time for last slope (transition to **RINGHOLD** mode)		11.25		∝s

Note: Rise and fall times of the ringer output signal depend on the actual supply voltage, whereas the slew rates are constant. For minimal supply voltage the typical rise and fall times are some 10% smaller than for nominal supply voltage. For maximal supply voltage the typical rise and fall times are some 10% greater than in the nominal case. Values given here are preliminary. Final values will be available as soon as measurements will be available.

11.3.1.2 Audio Transmit Path

Table 179 Electrical Characteristics of Audio Transmit Path

Parameter	Symbol	Limit Values			Unit	Remark
		Minimum	Typical	Maximum		
Differential input voltage				1.03	Vpp	
Differential input resistance			50		k&	
Input capacitance			5	10	pF	
Offset					mV	
S/D		65			dB	
Signal-to-noise ratio	S/N	75			dB	gs = +12 dB, bandwidth 300-3900 Hz (GSM mode)

Table 179 Electrical Characteristics of Audio Transmit Path (cont'd)

Parameter	Symbol	Limit Values			Unit	Remark
		Minimum	Typical	Maximum		
Signal-to-noise ratio	S/N	72			dB	gs = +12 dB, bandwidth 300 Hz-7.0 kHz (WAMR mode,)
Power supply rejection		66 62 45	85		dB dB dB	*gs* = 24 dB *gs* = 18 dB *gs* = 0 dB, $U_{VDDv}(t)$ = 2.5 V + 0.15V sin(2□•1 kHz•*t*)
Cross talk (between receive and transmit channel)				– 65		dB $U_{TX}(t)$ = 1.075 V + $U_{RX}(t)$ = 0.775 V sin(2□•1 kHz•*t*)
Cut-off frequency of anti-alias filter		16			kHz	
Gain of ADC					%/V	Definition see general remark
LSB step size					∝V	
Absolute gain drift				+/- 2	%	Variation due to change in VDD, temperature and life time

11.3.1.3 Microphone Supply

Table 180 Electrical Characteristics of Microphone Supply

Parameter	Symbol	Limit Values			Unit	Remark
		Minimum	Typical	Maximum		
Output voltage of pin VMICH			1.8 2.0 2.2		V	VDD (typ.) = 2.25 V ... 2.75 V VDD (typ.) = 2.4 V ... 2.75 V VDD (typ.) = 2.5 V ... 2.75 V
Output voltage of pin VMICL			0		V	
Microphone supply current				2.0	mA	
Load capacitance		1		2	nF	
Load resistance		1			k&	
Power supply rejection of microphone supply			75		dB	1)

1) UVDDbg(t) = 2.6 V + 0.10 V sin(2□•1 kHz•t) and gs = 0 dB in crosstalk free conditions at board level

Note: EN_L2N must be disabled before entering the standby powerdown mode because the PLL is powered down and, thus, de-asserting the LOCKED signal.

Table 181 Baseband Transmit Path (Signal Outputs I/IX, Q/QX) for TXON=0

Parameter	Symbol	Limit Values			Unit	Test Condition/ Remark
		Minimum	Typical	Maximum		
Differential output resistance		96.0	120.0	144.0	k&	TXON=0
Leakage current of ESD structure				10.0	uA	TXON=0

11.3.2 Measurement Interface

Table 182 ADC Characteristics

Parameter	Symbol	Limit Values			Unit	Remark
		Minimum	Typical	Maximum		
Resolution			12		Bits	
Differential linearity error	DNL			±0.5	LSB	
Integral linearity error	INL			±4	LSB	
Offset error	$U_{dig,0}$			±10	LSB	ADC input U_2 = 0 V
Gain	g_{ADC}	1966	2048	2130	LSB/V	1)
Absolute gain drift				±2	%	2)
Conversion time			60		cycle	clk_meas3) cycles
Throughput rate				f_M*S/480	Hz	S = K/L4)
Acquisition time			32		cycle	clk_kernel_24) cycles
Acquisition delay			39		cycle	clk_kernel_25) cycles
Acquisition jitter		0		1	cycle	clk_kernel_26) cycles
Wake-up time from power save			50		∝s	7)

1) Variation due to process tolerances and change in VDD_LANA, temperature and life time.

2) Variation due to change in VDD_LANA, temperature and life time.

3) f_M = clk_bus (52 MHz) and K = 02_H, L = $0D_H$ (typical). Refer to **Section 7.3.12 Clock Control of Measurement Interface (on Page 136)**.

4) During acquisition time the track and hold circuit is settling and tracking the signal (aperture). At the end of the acquisition period the track and hold circuit changes from track to hold and settles to its final value.

5) The acquisition delay is defined as the period in time from the occurrence of the rising edge of ADCTRIG (or setting **MEAS_CTRL1.START**) to the beginning of the acquisition period. Acquisition is finished after the acquisition time is over. Especially, for PWPA measurement care has to be taken that the timing advance of ADCTRIG matches with the desired end of the acquisition period.

6) The acquisition jitter is defined as the uncertainty in time where the acquisition period begins (ends) in comparison to the ideal point in time defined by the signal ADCTRIG.

7) Before ADC operation, the reference voltage has to be turned on (refer to **Section 7.4 Analog Control Registers (on Page 137)**).
Wake-up time of reference voltage (band-gap) has to be considered separately.

Table 183 Specification of pins M0 to M2 and M7 to M10[1)]

Parameter	Symbol	Limit Values			Unit	Remarks
		Minimum	Typical	Maximum		
Characteristics						
Input resistance		1			M&	2)3)

Table 183 Specification of pins M0 to M2 and M7 to M10[1)] (cont'd)

Input leakage current				0.1	∝A	
Equivalent network cross resistance	R_x	1440	2400	3360	k&	Modes MxMyA[4)]
		960	1600	2240		Modes MxMyB[4)]
Common mode input resistance	R_{cm}	360	600	840	k&	Modes MxMyA[4)]
		480	800	1120		Modes MxMyB[4)]
Internal common mode voltage	U_{cmi}	0.58	0.60	0.62	V	Modes MxMyA[2)4)]
		0.58	0.60	0.62		Modes MxMyB[2)4)]
Input resistance in measurement mode	R_{eq}	288	480	672	k&	Modes MxA[2)4)]
		320	533	747		Modes MxB[2)4)]
		70	117	163		Mode M10[2)4)]
Internal voltage	U_{eq}	0.46	0.48	0.50	V	Modes MxA[2)4)]
		0.38	0.40	0.42		Modes MxB[2)4)]
		0.48	0.50	0.52		Mode M10[2)4)]
Pre-amplifier gain	G		0.5			Modes MxA, MxMyA
			1.0			Modes MxB, MxMyB
			0.4			Mode M10
ADC gain	g_{ADC}	1966	2048	2130	LSB/V	[4)]
Absolute gain drift				±2		% [5)]
Current tolerance	IR, IS			±4	%	TC[2:0] > 0_H[4)]
[5)]Absolute current drift				±2	%	TC[2:0] > 0_H
[4)6)]Current ratio error				±1	%	TC[2:0] > 0_H
Pre-amplifier and multiplexer settling time	CSEL = LOW[7)]			18.8	∝s	Modes MxA, MxMyA
				37.5		Modes MxB, MxMyB
				3.8		Mode M10
	CSEL = HIGH[7)]			8.4		Modes MxA, MxMyA
				16.7		Modes MxB, MxMyB
				1.7		Mode M10
	-			1.0		Mode ADCCAL[7)]
Conditions						
Input voltage		0		VDD_LANA	V	Modes MxMyA/B, ADCCAL[2)8)]
		0		1.92		Modes MxA[8)9)]
		0		0.96		Modes MxB[8)9)]
		0		2.40		Mode M10[8)9)]
		0		VDD_LANA - 0.6 V		Also for all modes if TC[2:0] > 0[8)]
Differential input voltage		-1.92		1.92	V	Also for modes MxMyA[10)]
		-0.96		0.96		Also for modes MxMyB[10)]
		-0.96		0.96		Also for mode ADCCAL[10)]

Table 183 Specification of pins M0 to M2 and M7 to M10[1)] (cont'd)

Common mode input voltage		0	0.96	1.40	V	Also for modes MxMyA[11)]
		0	0.48	1.40		Also for modes MxMyB[11)]
		0.5		1.8		Also for mode ADCCAL[11)]

1) For measurement modes MxA, MxB and M10 as well as MxMyA, MxMyB and ADCCAL.
2) With respect to AGND.
3) If mode OFF is selected.
4) Variation due to process tolerances and change in VDD, temperature, and life time.
5) Variation due to change in VDD_LANA, temperature, and life time.
6) Error of ratio ITC1/ITC2 with TC[2:0]1 not equal to TC[2:0]2.
7) MX[5:0] has to be set in advance to the beginning of the acquisition time in order to guarantee that the external circuitry, the pre-amplifier, and the multiplexer are settled when the acquisition period begins. It is recommended to set CSEL = LOW to ensure 12 bit resolution of the ADC.
8) UMx/y for all measurement modes.
9) Pin may be left unconnected but measurement result is not specified.
10) UMx - UMy
11) (UMx + UMy)/2

Table 184 Specification of PAOUTOF1

Parameter	Symbol	Limit Values			Unit	Remark
		Minimum	Typical	Maximum		
Characteristics						
Input resistance		1			M&	1)2)
Overall gain		1966	2048	2130	LSB/V	U_{PAOUTx} 3)4)
Absolute gain drift				±2		% 3)5)
LSB step size			488		mV	
Pre-amplifier and multiplexer settling time				37.5	µs	**MEAS_CTRL1.CSEL** = 0[6)]
				16.7		**CSEL** = 1[6)]
Conditions						
Input voltage		0		0.95	V	2)

1) If mode OFF (**MEAS_CTRL1.MX** = 0_H) is selected.
2) With respect to AGND
3) including amplifier and ADC, without external load
4) Variation due to process tolerances and change in VDD_LANA, temperature and life time.
5) Variation due to change in VDD_LANA, temperature and life time.
6) **MEAS_CTRL1.MX** has to be set before the start of the acquisition time to guarantee that the external circuitry, the pre-amplifier, and the multiplexer are settled when the acquisition period begins.

Table 185 On Chip Temperature Measurement TIC

Parameter	Symbol	Limit Values			Unit	Remark
		Minimum	Typical	Maximum		
Temperature range		-30		125	°C	
Relative accuracy			±1	±2		°C[1)]

Table 185 On Chip Temperature Measurement TIC (cont'd)

Absolute accuracy				±5		°C[1)]
Multiplexer settling time				1	µs	[2)]

1) guaranteed for at least to two years after calibration according **Section 7.3.6 On-Chip Temperature Measurement (on Page 121)**.

2) **MEAS_CTRL1.MX** has to be set before the start of the acquisition time to guarantee that the external circuitry, the pre-amplifier, and the multiplexer are settled when the acquisition period begins.

11.3.3 RF Power Ramping

Table 186 Specification of Pin PAOUT1 (TXON = 1)

Parameter	Symbol	Limit Values			Unit	Test Condition/ Remark
		Minimum	Typical	Maximum		
Characteristics						
Gain		1.18	1.25	1.32	mV/LSB	1)
Gain drift				±1	%	2)
Offset		-20 -70	0 -50	40 -10	mV	PA_OFF = 0 1) PA_OFF = 1
Offset drift				±5	mV	2)
Output resistance				10	W	Within linear operating range
Saturation voltage	U_{sat+}	-25 -50 -270			mV	I_{PAOUT} = +0.05 mA I_{PAOUT} = +0.50 mA I_{PAOUT} = +5.00 mA U_{sat+} is referred to VDDbb 3)
	U_{sat-}			25	mV	U_{sat-} is referred to AGND 3)
Integral nonlinearity	INL			±10	LSB	
Differential nonlinearity	DNL			±1	LSB	
Power supply rejection		50			dB	$U_{VDDr}(t)$ = 2.5 V + 0.1V sin(2□•1 kHz•t) 4)
– 3 dB freq. of postfilter		180	300	450	kHz	1st order filter
Conditions						
Load capacitance				40	pF	
Load current	I_{PAOUT}	0		5	mA	Tested @ I_{PAUOT}=1 mA
Dropout voltage	U_{drop+}	-50 -100 -330			mV	I_{PAOUT} = +0.05 mA I_{PAOUT} = +0.50 mA I_{PAOUT} = +5.00 mA
						$U_{drop+ is}$ referred to VDDbb 5)
	U_{drop-}			50	mV	U_{drop-} is referred to AGND 5)

1) Variation due to process tolerances and change in VDDbb, temperature and lifetime.

2) Variation due to change in VDDbb, temperature and life time.

3) Saturation voltage is the output voltage if the output is completely saturated. Nonlinear settling to be expected.
Saturation voltages are given with respect to AGND and VDDbb.
Saturation voltage is independent on the offset voltage.
Voltage drops on wire resistances on the chip and within the package are included.
Saturation voltage has to be considered with respect to the actual output voltage, i.e. offset and gain errors as well as tolerances of VDDbb have to be considered.

4) Valid within the linear operating range between AGND+U_{drop-} and VDDbb+U_{drop+}.

5) Dropout voltage is a condition for the maximum/minimum output voltage for which a linear signal processing is achieved.
Dropout voltages are given with respect to AGND and VDDbb.
Dropout voltage is independent on the offset voltage.
Voltage drops on wire resistances on the chip and within the package are included.
Dropout voltage has to be considered with respect to the actual output voltage, i.e. offset and gain errors as well as tolerances of VDDbb have to be considered.

Table 187 Specification of Pins PAOUT1 (TXON = 0)

Parameter	Symbol	Limit Values			Unit	Test Condition/ Remark
		Minimum	Typical	Maximum		
Leakage current of ESD structure				10	∝A	
Output resistance				500	W	

11.3.4 AC/DC Characteristics of RF Subsystem

AC/DC characteristics involve the spread of values valid within the specified supply voltage and ambient temperature, refer to **Table 188**. Typical characteristics are the median of the production. All typical values are valid for 1.5 V and 2.5 V supply voltage at T_A= 25°C.

Table 188 AC/DC Characteristics

#	Parameter	Symbol	Limit Values			Unit	Test Conditions	Item
			Min	Typ	Max			
1. Receiver Supply Current								
	2.5V supply	VDD_LRFRX2V5		21	25.5	mA		
	2.5V supply, DCXO	VDD_LRFXO		1.6	2.3	mA	Z_L = 12 pF//30 k&, XOcal = 4,	
	2.5V supply, DCXO, Phase detector	VDD_LRFXO		6.5	7.5			
	1.5V supply	VDD_LRFTRX1V5		1	2	mA		
	1.5V supply	VDD_LRFTRX1V5		12.5	15	mA		
	1.5V supply	VDD_LRFTRX1V5		43	60	mA		
	2.5V supply	VDD_LRFRX2V5		41.5	50	mA		
	Subtotal 1.5V			56.5	77	mA	1)	
	Subtotal 2.5V			69	83	mA	2)	
	Total			257	339	mW		
2. Transmitter Supply Current								
	2.5V supply	VDD_LRFXO		2.3	3	mA		
	2.5V supply, DCXO	VDD_LRFXO		1.6	2.0	mA	Z_L = 12 pF//30 k&, XOcal = 4,	
	2.5V supply, DCXO, Phase detector	VDD_LRFXO		6.5	7.5			
	1.5V supply	VDD_LRFTRX1V5		1	2	mA		
	1.5V supply	VDD_LRFTRX1V5		40.5	49	mA		
	1.5V supply	VDD_LRFTRX1V5		14	18	mA		
	2.5V supply	VDD_LRFRX2V5		41.5	50	mA		
	Subtotal 1.5V			55.5	69	mA	1	
	Subtotal 2.5V			50.3	60.5	mA	2	
	Total			209	268	mW		

Table 188 AC/DC Characteristics (cont'd)

#	Parameter	Symbol	Limit Values			Unit	Test Conditions	Item
			Min	Typ	Max			
3. Standby Current								
	Standby Mode Current Consumption	I_{sby}			100	∝A	Sum of Standby Currents on all 1.5 V Supply Lines, T_A=25°C	
4. Overall Receiver Characteristics GSM850/EGSM900								
RX1/RX1X; f_{RX1}=869 MHz..894 MHz (GSM850) RX2/RX2X; f_{RX2}=925 MHz..960 MHz (EGSM900) All values valid for RXGAIN[1:0] = 11, RXGS[3:0] = 0000, RXCORR[3:0] = 0100 (default = 0 dB) unless otherwise stated.								
	Adjusted Overall Gain	$G_{RX1/2\_H\_adj}$	56.1	57.0	57.9	dB	RXCORR[3:0] applied	3)
	Overall High Gain	$G_{RX1/2\_H}$	51.5	57.0	62.5	dB		
	Overall Medium Gain	$G_{RX1/2\_M}$	37.5	43.0	48.5	dB	RXGAIN[1:0] = 01	
	Overall Low Gain	$G_{RX1/2\_L}$	16.5	23.0	27.5	dB	RXGAIN[1:0] = 00	
	Noise Figure	$NF_{RX1/2}$		2.4	3.7	dB	Spot NF @80 kHz[4]	3)
	Noise Figure	$NF_{RX1/2\_M}$			8.5	dB	Spot NF[4] @80 kHz, RXGAIN[1:0] = 01	3)
	Noise Figure	$NF_{RX1/2\_L}$			30	dB	Spot NF[4] @80 kHz, RXGAIN[1:0] = 00	3)
	Input Referred Integral Noise Power	$N_{intRX1/2}$		-123.6	-120.8	dBm	1 kHz..30 kHz[5]	
	Desensitization NF (incl. LO Phase Noise) NF_{tot} @ $P_{Blocker}$	$NFDES_{RX1/2\_0M6}$		3.5	8.0	dB	–f=600 kHz Pin=-47.5 dBm[6]	3)
	Desensitization NF (incl. LO Phase Noise) NF_{tot} @ $P_{Blocker}$	$NFDES_{RX1/2\_1M6}$		4.0	8.0	dB	–f=1.6 MHz Pin=-37.5 dBm[6]	3)
	Desensitization NF (incl. LO Phase Noise) NF_{tot} @ $P_{Blocker}$	$NFDES_{RX1/2\_3M}$		5.0	7.5	dB	–f=3 MHz, Pin=-27.5 dBm[6]	3)
	Desensitization NF (incl. LO Phase Noise) NF_{tot} @ $P_{Blocker}$	$NFDES_{RX1/2\_10M}$		3.5	7.5	dB	–f=10 MHz, Pin=-24 dBm[6]	
	Desensitization NF (incl. LO Phase Noise) NF_{tot} @ $P_{Blocker}$	$NFDES_{RX1/2\_20M}$		3.5	7.5	dB	–f=20 MHz, Pin=-24 dBm[6]	3)
	3rd Order Input Intercept Point	$IIP3_{RX1/2}$	-15	-12		dBm	[8]	3)
	AM Suppression	$AM_{RX1/2}$	82	90		dB	–f>6 MHz, T_A = 25°C[10]	
	Harmonic Suppression @ n x 26 MHz	$a_{n26\_RX1/2}$		70		dB	Harmonics within RX Band[10]	3)

Table 188 AC/DC Characteristics (cont'd)

#	Parameter	Symbol	Limit Values			Unit	Test Conditions	Item
			Min	Typ	Max			
	Harmonic Suppression 0.5*f_{LO}	$a_{05LO\_RX1/2}$		55		dB	T_A = 25°C, $P_{in,IC}$ = -50 dBm[10)]	[3)]
	Harmonic Suppression 2*f_{LO}	$a_{2LO\_RX1/2}$		55		dB	T_A = 25°C[10)]	[3)]
	Harmonic Suppression 3*f_{LO}	$a_{3LO\_RX1/2}$		20		dB	Differential Mode Blocker, T_A = 25°C[10)]	[3)]
	Harmonic Suppression 3*f_{LO}	$a_{3LO\_CM\_RX1/2}$		40		dB	Common Mode Blocker, T_A = 25°C[10)]	[3)]
	Harmonic suppression 4*f_{LO}	$a_{4LO\_RX1/2}$		50		dB	T_A = 25°C[10)]	[3)]
	Harmonic Suppression 5*f_{LO}	$a_{5LO\_RX1/2}$		30		dB	Differential Mode Blocker, T_A = 25°C[10)]	[3)]
	Harmonic Suppression 5*f_{LO}	$a_{5LO\_CM\_RX1/2}$		50		dB	Common Mode Blocker, T_A = 25°C[10)]	[3)]
	Spurious Emission at RF Port RX1/RX2 (single ended)			-43	-33	dBm	@VCO-Frequency[10)]	[3)]
	Spurious Emission at RF Port RX1/RX2 (single ended)			N/A	-57	dBm	Spurious <= 1 GHz[10)]	[3)]
	Spurious Emission at RF Port RX1/RX2 (single ended)			-50	-40	dBm	Spurious > 1 GHz [10)]	[3)]
	RX1 Input Impedance (differential)	Z_{RX1}		50-j204		&	f = 882 MHz[10)]	[3)]
	RX2 Input Impedance (differential)	Z_{RX2}		57-j220		&	f = 942 MHz[10)]	[3)]
	1dB-Compression Point	$P_{1dB\_RX1/2\ L}$	-22	-16		dB	–f =30 kHz, RXGAIN[1:0] = 00	

5. Overall Receiver Characteristics GSM1800/GSM1900

RX3/RX3X; f_{RX3} = 1805 MHz..1880 MHz (GSM1800)
RX4/RX4X; f_{RX4} = 1930 MHz..1990 MHz (GSM1900)
All values valid for RXGAIN[1:0] = 11, RXGS[3:0] = 0000, RXCORR[3:0] = 0100 (default = 0 dB) unless otherwise stated.

	Adjusted Overall Gain	$G_{RX3/4\_H\_adj}$	55.1	56	56.9	dB	RXCORR[3:0] applied	[3)]
	Overall High Gain	$G_{RX3/4\_H}$	50.5	56.0	61.5	dB		
	Overall Medium Gain	$G_{RX3/4\_M}$	36.5	42.0	47.5	dB	RXGAIN[1:0] = 01	
	Overall Low Gain	$G_{RX3/4\_L}$	16.5	22.0	27.5	dB	RXGAIN[1:0] = 00	
	Noise Figure	$NF_{RX3/4}$		2.7	4.3	dB	Spot NF @80 kHz[4)]	[3)]

Table 188 AC/DC Characteristics (cont'd)

#	Parameter	Symbol	Limit Values			Unit	Test Conditions	Item
			Min	Typ	Max			
	Noise Figure	$NF_{RX3/4\_M}$			10	dB	Spot NF @80 kHz[4], RXGAIN[1:0]=01	[3]
	Noise Figure	$NF_{RX3/4\_L}$			30	dB	Spot NF @80 kHz[4], RXGAIN[1:0] = 00	[3]
	Input Referred Integral Noise Power	$N_{intRX3/4}$		-122.8	-120.0	dBm	1 kHz..30 kHz[7]	
	Desensitization NF (incl. LO Phase Noise) NF_{tot} @ $P_{Blocker}$	$NFDES_{RX3/4\_0M6}$		6	8	dB	T_A=25°C, ⎓f=600 kHz, Pin=-47.5 dBm[6]	[3]
	Desensitization NF (incl. LO Phase Noise) NF_{tot} @ $P_{Blocker}$	$NFDES_{RX3/4\_1M6}$		6	8	dB	T_A=25°C, ⎓f=1.6 MHz, Pin=-37.5 dBm[6]	[3]
	Desensitization NF (incl. LO Phase Noise) NF_{tot} @ $P_{Blocker}$	$NFDES_{RX3/4\_3M}$		6	8	dB	T_A=25°C, ⎓f=3 MHz, Pin=-30.5 dBm[6]	[3]
	Desensitization NF (incl. LO Phase Noise) NF_{tot} @ $P_{Blocker}$	$NFDES_{RX3/4\_20M}$		4 MHz,	7.5	[3] dB	Pin=-24 dBm[6]	⎓f=20
	3rd Order Input Intercept Point	$IIP3_{RX3/4}$	-15	-12		dBm	[8]	[3]
	AM Suppression	$AM_{RX3/4}$	82	90		dB	⎓f>6 MHz, T_A=25°C[9]	
	Harmonic Suppression @n x 26 MHz	$a_{n26\_RX3/4}$		70		dB	Harmonics within RX Band[10]	[3]
	Harmonic Suppression 0.5*f_{LO}	$a_{05LO\_RX3/4}$		55		dB	T_A = 25°C, $P_{in,IC}$ = -56 dBm[10]	[3]
	Harmonic Suppression 2*f_{LO}	$a_{2LO\_RX3/4}$		55		dB	T_A = 25°C[10]	[3]
	Harmonic Suppression 3*f_{LO}	$a_{3LO\_RX3/4}$		25		dB	Differential Mode Blocker, T_A = 25°C[10]	[3]
	Harmonic Suppression 3*f_{LO}	$a_{3LO\_CM\_RX3/4}$		45		dB	Common Mode Blocker, T_A = 25°C[10]	[3]
	Harmonic Suppression 4*f_{LO}	$a_{4LO\_RX3/4}$		55		dB	T_A = 25°C[10]	[3]
	Harmonic Suppression 5*f_{LO}	$a_{5LO\_RX3/4}$		40		dB	Differential Mode Blocker, T_A = 25°C[10]	[3]
	Harmonic Suppression 5*f_{LO}	$a_{5LO\_CM\_RX3/4}$		60		dB	Common Mode Blocker, T_A = 25°C[10]	[3]
	Spurious Emission at RF Port RX3/RX4			-46	-33	dBm	@VCO-Frequency[10]	[3]
				N/A	-57	dBm	Spurious <= 1 GHz[10]	[3]
				-63	-47	dBm	Spurious > 1 GHz[10]	[3]

Table 188 AC/DC Characteristics (cont'd)

#	Parameter	Symbol	Limit Values			Unit	Test Conditions	Item
			Min	Typ	Max			
	RX3 Input Impedance (differential)	Z_{RX3}		22-j150		&	f = 1842 MHz[10)]	3)
	RX4 Input Impedance (differential)	Z_{RX4}		20-j138		&	f = 1960 MHz[10)]	3)
	1dB-Compression Point	$P_{1dB\_RX3/4\ L}$	-24	-14		dB	–f =30 kHz, RXGAIN[1:0] = 00	
6. Baseband Filter Response								
	3 dB Roll Off Frequency	f_{3dB}	132	142	152	kHz		3)
	Group Delay Ripple	$-\lvert_g$		530	620	ns	0..80 kHz	3)
	Passband Ripple	$-a_p$		0.2	0.3	dB	0..80 kHz	3)
	Stopband Attenuation (Unmodulated Source)	a_{s_200k}	7.0	8.5		dB	at 200 kHz	3)
		a_{s_400k}	22.0	24.0		dB	at 400 kHz	
		a_{s_600k}	32.0	34.0		dB	at 600 kHz	3)
		a_{s_800k}	39.0	41.0		dB	at 800 kHz	
		$a_{s\_1.6M}$	57.0	59.0		dB	at 1.6 MHz	3)
		a_{s_3M}	73.0	75.0		dB	at 3 MHz	3)
		a_{s_13M}	96.0	120		dB	at 12.9 MHz	3)
	Stopband Attenuation (GSM Modulated Source)	a_{sGSM_200k}	5.5	6.5		dB	at 200 kHz	3)
		a_{sGSM_400k}	21.0	22.5		dB	at 400 kHz	3)
		a_{sGSM_600k}	31.0	33.0		dB	at 600 kHz	3)
7. PGC, Output Signal A/AX and B/BX								
	Gain Correction Range	$-G_{GCR}$	-6		+6	dB		
	Gain Correction Step	$-G_{GCS}$		1			dB	
	Gain Step 0	$-G_{RXGS0}$	11.8	12	12.2	dB		
	Gain Step 1	$-G_{RXGS1}$	5.8	6	6.2	dB		
	Gain Step 2	$-G_{RXGS2}$	-5.8	-6	-6.2	dB		
	Gain Step 3	$-G_{RXGS3}$	-2.8	-3	-3.2	dB		
	DC Output level (Common Mode)	V_{DC_CM}		0.95		V	RXCM[1:0] = 00	
				1.25		V	RXCM[1:0] = 01	
				1.35		V	RXCM[1:0] = 10	
				1.425		V	RXCM[1:0] = 11	
	DC Offset (Differential Mode)	V_{DC_DM}		< 0.02	0.15	V	RXGAIN[1:0] = 11 RXCORR[3:0] = 0100 RXGS[3:0] = 0010 OFC = 0	
	DC Offset (Differential Mode) after Compensation	$V_{DC\_DM\_adj}$		< 0.02	0.2	V	RXGAIN[1:0] = 11 RXCORR[3:0] = 0100 RXGS[3:0] = 0011 OFC = 1	

Table 188 AC/DC Characteristics (cont'd)

#	Parameter	Symbol	Limit Values			Unit	Test Conditions	Item
			Min	Typ	Max			
	I/Q Phase Error	−∏I/Q		< 0.01	3	deg	11) RXGAIN[1:0] = 11	
	I/Q Amplitude Mismatch	−VI/Q		< 0.1	0.5	dB	11) RXGAIN[1:0] = 11	
	Short Term Offset Drift (Differential Mode)	V_{Drift}		< 100	400	∝V	12) RXGAIN[1:0] = 11 RXCORR[3:0] = 0100 RXGS[3:0] = 0010 OFC = 0	3)
	Output Resistance			200	500	&		
8. Overall Transmitter Characteristics								
GSM850/900 TX1; f_{TX}=824 MHz...849 MHz; f_{TX}=880 MHz...915 MHz								
	Output Level	P_{TX1}	1.5	3.5	5.5	dBm		
	Output Impedance	Z_{TX1}		50		&		3)
	Output Return Loss	RL_{TX1}	15	22		dB		3)
	Output Noise Floor	N_{TX1_20M}		-165.5	-163	dBc/Hz	|−f| = 20 MHz13) RBW=100 kHz T_A = 25°C	3)
	RMS Phase Error	E_{RMS_TX1}		0.7	2.5	° rms		3)
	Peak Phase Error	E_{peak_TX1}		2	7.5			°peak 3)
	PRBS Modulation Spectrum14) @200 kHz offset @250 kHz offset @400 kHz offset @600 kHz offset @1800 kHz offset @3000 kHz offset @6000 kHz offset			 -37 -41 -69 -75 -86 -87 -90	 -33 -36 -62-67 -73 -75 -80	dB	 RBW=30 kHz RBW=30 kHz RBW=30 kHz3) RBW=30 kHz3) RBW=100 kHz3) RBW=100 kHz3) RBW=100 kHz 3)	
9. Overall Transmitter characteristics								
GSM1800/1900 TX2; f_{TX}=1710 MHz...1785 MHz; f_{TX}=1850 MHz...1910 MHz								
	Output Level	P_{TX2}	1.5	3.5	5.5	dBm		
	Output Impedance	Z_{TX2}		50		&		
	Output Return Loss	RL_{TX2}	15	21		dB		
	Output Noise Floor	N_{TX2_20M}		-159	-154	dBc/Hz	|−f| = 20 MHz**13** RBW = 100 KkHzHz T_A = 25°C	
	RMS Phase Error	E_{RMS_TX2}		1.2	2.8	° rms		
	Peak Phase Error	E_{peak_TX2}		4	7.5	°peak		

Table 188 AC/DC Characteristics (cont'd)

#	Parameter	Symbol	Limit Values			Unit	Test Conditions	Item
			Min	Typ	Max			
	PRBS Modulation Spectrum[15)] @200 kHz offset @250 kHz offset @400 kHz offset @600 kHz offset @1800 kHz offset @6000 kHz offset			 -37 -41 -67 -73 -83 -85	 -33 -36 -62-65 -73 -80	dB	 RBW=30 kHz RBW=30 kHz RBW=30 kHz[3)] RBW=30 kHz[3)] RBW=100 kHz[3)] RBW=100 kHz[3)]	
10. Transmitter Baseband Inputs A/AX, B/BX								
	Input Resistance	R_{inTX}		100		k&	TX Mode only Single Ended	
	Input Capacitance	C_{inTX}		5		pF	TX Mode only Single Ended	
11. Sigma-Delta Synthesizer								
	Settling time (5° residual phase error)	t_{settle}		150	190	µs	5° refer to f_{TX} = 1990 MHz	
	SSB Inband Phase Noise of PLL @ 1 kHz	PN_{PLL_1K}		-77	-75	dBc/Hz	Measured at TX2	
	SSB Inband Phase Noise of PLL @ 40 kHz	PN_{PLL_40K}		-92	-88.5	dBc/Hz	Measured at TX2	
	Peak Phase Error due to Drop of Supply Voltages	$\|E_{peak\_drop}\|$		5	18	°peak	Measured at TX2 either 1.5 or 2.5 V or both Supply Domains Drop Linearly by 10 mV within 2 ∝s	
12. Crystal Oscillator 26MHz (DCXO)								
Digitally Controlled Crystal Oscillator Core								
	Startup Time	t_{start}			5	ms	|−f| δ 1.5ppm 90% Amplitude	
	Tuning Sensitivity	k_{XO}		0.01	0.025	ppm/LSB		
	Negative Resistance	R_{N_1f}		-200	-160	W	Fundamental, Measured at -40 dBm in Subrange 1	
		R_{N_3f}	(R_{N_1f}) /6				3rd Harmonic	
	Tuning Nonlinearity			3.5	12	%	20 Subranges compared with Full Range	
	Crystal Drive Level	P_{XTAL}		20	60	∝W		

1) Including PLL supply current I_{PLLtyp} = 14 mA / I_{PLLmax} = 17 mA

2) Including PLL supply current I_{PLLtyp} = 42 mA / I_{PLLmax} = 51 mA

3) Not subject to production test - verified by characterization and design.
4) Including worst case flicker noise contribution at 80 kHz.
5) According to f_{c2kT0} = 11.5 kHz/NF_{eqth} = 3.5 dB/〈 = 1, the mean value of channel A and B.
6) Desensitization measured with notch filter.
7) According to f_{c2kT0} = 15 kHz/NF_{eqth} = 4.0 dB/〈 = 1, mean value of channel A and B.
8) P1, P2 = -49 dBm, f1 = 800 kHz, f2 = 1.6 MHz.
9) Single tone P_B = -33 dBm, P_W = -101 dBm.
10) Measured on IFX S-parameter board.
11) P_{in} = -70 dBm; f_{CW} = f_{LO} + 20 kHz, measured from RF input to A,AX; B,BX.
12) Measurement starts 100 ∝s after sending RXTX word within 577 ∝s.
13) $P_{carrier}$ and P_{noise} measured with “normal dBm marker” of spectrum analyzer
N = P_{noise} - $P_{carrier}$ - 10 log (RBW/Hz).
14) Refer to GSM spec. 51.010.
15) Refer to GSM spec. 45.005.

11.4 AC/DC Characteristics of Linear Voltage Regulators

11.4.1 LIO Linear Voltage Regulator

Table 189 LIO Performance Specification

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
External capacitor	C_{ext}	-55%	1	+35%	∝F	External capacitor to be connected during test
Capacitor ESR	R_{ESR}			100	Ohm	100 Hz
				0.2	Ohm	100 kHz…10 MHz
Output voltage	V_{REG}	-3%	2.85	+3%	V	V_{DD} = 3.1 ... 5.5 V I_{REG} = 0 ... 30 mA
Output voltage	V_{REG}	-3%	1.80	+3%	V	V_{DD} = 3.1 ... 5.5 V I_{REG} = 0 ... 30 mA
Output current	I_{REG}			30	mA	
Current limitation	I_{MAX}	100		180	mA	Current pulled down from LDO to GND until LDO voltage is 50% of nominal value
Maximum Overshooting	V_{OS}			15	mV	
Discharge time constant	T_{OFF}			1	ms	90 -> 5%
Static Parameter						C = 1 ∝F
Line regulation (V_{REG} = 2.85 V)	DV_{REG}			7	mV	V_{DD} = 3.1...5.5 V I_{REG} = 30 mA
Load regulation (V_{REG} = 2.85 V)	DV_{REG}			5	mV	V_{DD} = 3.1 V I_{REG} = 1...30 mA
Dropout voltage	V_{DROP}			150	mV	I_{REG} = 30 mA
Dynamic Parameter						
Line transient response	V_{PEAK}			5	mV	V_{DD} = 3.1...3.6 V I_{REG} = 30 mA t_r = 10 ∝s, t_f = 10 ∝s
Load transient response	V_{PEAK}			20	mV	V_{DD} = 3.1 V I_{REG} = 1...30 mA t_r = 1 ∝s, t_f = 1∝s
Power supply rejection ratio (V_{REG} = 2.85 V)	PSRR	35			dB	f = 50 Hz .. 20 kHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp} I_{REG} = 0...30 mA

11.4.2 LANA Linear Voltage Regulator

Table 190 LANA Performance Specification

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
External capacitor	C_{ext}	-55%	1	+35%	∝F	External capacitor to be connected during test
Capacitor ESR	R_{ESR}			100	Ohm	100 Hz
				0.2	Ohm	100 kHz…10 MHz
Output voltage	V_{REG}	-3%	2.5	+3%	V	V_{DD} = 3.1...5.5 V I_{REG} = 0...100 mA
Output current	I_{REG}			100	mA	
Current Consumption	I_{DDREG}			125	∝A	I_{REG} = 0
active mode				125	∝A	I_{REG} = 1 mA
				550	∝A	I_{REG} =
100 mA Current Consumption	I_{DDREG}			50	∝A	I_{REG} = 0
idle mode				60	∝A	I_{REG} = 1 mA
power down				4	∝A	regulator switched off
Current limitation	I_{MAX}	165		265	mA	Current pulled down from LDO to GND until LDO voltage is 50% of nominal value
Maximum Overshooting	V_{OS}			20	mV	
Startup time	$t_{startup}$			50	∝s	0 -> 90% Iload = 10 mA
Discharge time constant	T_{OFF}			1	ms	90 -> 5% C = 1.0 ∝F
Static Parameter						
Line regulation	$-V_{REG}$			2.5	mV	V_{DD} = 3.1 ... 5.5 V I_{REG} = 100 mA
Load regulation	$-V_{REG}$			9	mV	V_{DD} = 3.1 V I_{REG} = 1 ... 100 mA
Dropout voltage	V_{DROP}			450	mV	I_{REG} = 100 mA
Dynamic Parameter						
Line transient response	V_{PEAK}			2	mV	V_{DD} = 3.1 … 3.6 V I_{REG} = 100 mA t_r = 10 ∝s, t_f = 10∝s
Load transient response	V_{PEAK}			15	mV	V_{DD} = 3.1 V I_{REG} = 0...10 mA t_r = 1 ∝s, t_f = 1 ∝s
	V_{PEAK}			33	mV	V_{DD} = 3.1 V I_{REG} = 1...100 mA t_r = 1 ∝s, t_f = 1 ∝s

Table 190 LANA Performance Specification (cont'd)

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
	t_{TRU}			30	∝s	For both conditions above; transients decayed to +/- 1 mV
	t_{TRD}			20	∝s	
Power supply rejection	PSRR	57			dB	f = 10 Hz...1 kHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp} I_{REG} = 0 ... 100 mA
		54			dB	f = 1 ... 100 kHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp} I_{REG} = 0 ... 100 mA
		34			dB	f = 100 kHz...1 MHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp} I_{REG} = 0 ... 100 mA
Output noise voltage	V_{RMS}			48	∝V_{RMS}	10 Hz to 100 kHz V_{DD} = 3.1...5.5 V I_{REG} = 0 ... 100 mA

11.4.3 LRFRX Linear Voltage Regulator

Table 191 LRFRX Performance Specification

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
External capacitor	C_{ext}	-55%	1	+35%	∝F	External capacitor to be connected during test
Capacitor ESR	R_{ESR}			100	Ohm	100 Hz
				0.2	Ohm	100 kHz…10 MHz
Output voltage	V_{REG}	-3%	2.5	+3%	V	V_{DD} = 3.1...5.5 V I_{REG} = 0...100 mA
Output current	I_{REG}			100	mA	
Current Consumption	I_{DDREG}			125	∝A	I_{REG} = 0
active mode				125	∝A	I_{REG} = 1 mA
				550	∝A	I_{REG} =
100 mA Current Consumption	I_{DDREG}			50	∝A	I_{REG} = 0
idle mode				60	∝A	I_{REG} = 1 mA
power down				4	∝A	regulator switched off
Current limitation	I_{MAX}	165		265	mA	Current pulled down from LDO to GND until LDO voltage is 50% of nominal value
Maximum Overshooting	V_{OS}			20	mV	
Startup time	$t_{startup}$			50	∝s	0 -> 90% Iload = 10 mA
Discharge time constant	T_{OFF}			1	ms	90 -> 5% C = 1.0 ∝F
Static Parameter						
Line regulation	$-V_{REG}$			2.5	mV	V_{DD} = 3.1 ... 5.5 V I_{REG} = 100 mA
Load regulation	$-V_{REG}$			9	mV	V_{DD} = 3.1 V I_{REG} = 1 ... 100 mA
Dropout voltage	V_{DROP}			450	mV	I_{REG} = 100 mA
Dynamic Parameter						
Line transient response	V_{PEAK}			2	mV	V_{DD} = 3.1 … 3.6 V I_{REG} = 100 mA t_r = 10 ∝s, t_f = 10 ∝s
Load transient response	V_{PEAK}			15	mV	V_{DD} = 3.1 V I_{REG} = 0...10 mA t_r = 1 ∝s, t_f = 1 ∝s
	V_{PEAK}			33	mV	V_{DD} = 3.1 V I_{REG} = 1...100 mA t_r = 1 ∝s, t_f = 1 ∝s

Table 191 LRFRX Performance Specification (cont'd)

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
	t_{TRU}			30	µs	For both conditions above; transients decayed to +/- 1 mV
	t_{TRD}			20	µs	
Power supply rejection	PSRR	57			dB	f = 10 Hz...1 kHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp} I_{REG} = 0 ... 100 mA
		54			dB	f = 1 ... 100 kHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp} I_{REG} = 0 ... 100 mA
		34			dB	f = 100 kHz...1 MHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp} I_{REG} = 0 ... 100 mA
Output noise voltage	V_{RMS}			48	µV_{RMS}	10 Hz to 100 kHz V_{DD} = 3.1...5.5 V I_{REG} = 0 ... 100 mA

11.4.4 LRFTRX Linear Voltage Regulator

Table 192 LRFTRX Performance Specification

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
External capacitor	C_{ext}	-35%	1	+35%	µF	External capacitor to be connected during test
Capacitor ESR	R_{ESR}			100	Ohm	100 Hz
				0.2	Ohm	100 kHz…10 MHz
Output voltage	V_{REG}	-3%	1.50	+3%	V	V_{DD} = 3.1...5.5 V I_{REG} = 0...120 mA
Output current				120	mA	
Current limitation	I_{MAX}	160		800	mA	Current pulled down from LDO to GND until LDO voltage is 50% of nominal value
Maximum Overshooting	V_{OS}			30	mV	
Startup time	$t_{startup}$			200	µs	0 -> 90% Iload = 10 mA
Discharge time constant	T_{OFF}			1	ms	90 -> 5% C = 1.0 µF

Table 192 LRFTRX Performance Specification (cont'd)

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
Static Parameter						
Line regulation	DV_{REG}			5	mV	V_{DD} = 3.1...5.5 V I_{REG} = 120 mA
Load regulation	DV_{REG}			9	mV	V_{DD} = 3.1 V I_{REG} = 1... 120 mA
Dropout voltage	V_{DROP}			800	mV	I = 120 mA
Dynamic Parameter						
Line transient response	V_{PEAK}			10	mV	V_{DD} = 3.1...3.6 V I_{REG} = 120 mA
						t_r = 10 ∝s, t_f = 10 ∝s
	t_{TRU}			20	∝s	for definition see Figure 215
	t_{TRD}			20	∝s	
Load transient response	V_{PEAK}			25	mV	V_{DD} = 3.1 V I_{REG} = 0...10 mA
	V_{PEAK}			30	mV	t_r = 1 ∝s, t_f = 1 ∝s V_{DD} = 3.1 V I_{REG} = 1...100 mA
	t_{TRU}			64	∝s	t_r = 1 ∝s, t_f = 1 ∝s For both conditions above; transients decayed to +/- 1 mV
	t_{TRD}			68	∝s	
Power supply rejection	PSRR	55			dB	f = 10 Hz...1 kHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp}
		38			dB	I_{REG} = 0 ... 120 mA f = 1 ... 100 kHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp}
		38			dB	I_{REG} = 0 ... 120 mA f = 100 kHz...1 MHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp}
Output noise voltage	V_{RMS}			40	∝V_{RMS}	I_{REG} = 0 ... 120 mA 10 Hz to 100 kHz V_{DD} = 3.1...5.5 V I_{REG} = 0 ... 120 mA

11.4.5 LRFXO Linear Voltage Regulator

Table 193 LRFXO Performance Specification

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
External capacitor	C_{ext}	-55%	1	+35%	∝F	External capacitor to be connected during test
Capacitor ESR	R_{ESR}			100	Ohm	100 Hz
				0.2	Ohm	100 kHz…10 MHz
Output voltage	V_{REG}	-3%	2.5	+3%	V	V_{DD} = 3.1...5.5 V I_{REG} = 0...10 mA
Output current	I_{REG}			10	mA	
Current Consumption	I_{DDREG}			110	∝A	I_{REG} = 0
				130		I_{REG} = 1 mA
				250		I_{REG} = 1 0mA
Power Down Consumption				5		I_{REG} = 0 mA
Current limitation	I_{MAX}	50		75	mA	Current pulled down from LDO to GND until LDO voltage is 50% of nominal value
Maximum Overshoot	V_{OS}			10	mV	
Startup time	$t_{startup}$			124	∝s	0 -> 90% Iload = 3 mA
Discharge time constant	T_{OFF}			1	ms	90 -> 5% C = 1 ∝F
Static Parameter						
Line regulation	DV_{REG}			3	mV	V_{DD} = 3.1...5.5 V I_{REG} = 10 mA
Load regulation	DV_{REG}			3	mV	V_{DD} = 3.1 V I_{REG} = 50 ∝A ... 10 mA
Dropout voltage	V_{DROP}			150	mV	I_{REG} = 10 mA
Dynamic Parameter						
Line transient response	V_{PEAK}			2	mV	V_{DD} = 3.1…3.6 V I_{REG} = 10 mA t_r = 10 ∝s, t_f = 10 ∝s
Load transient response	V_{PEAK}			10	mV	V_{DD} = 3.1 V I_{REG} = 50 ∝A ... 10 mA t_r = 1 ∝s, t_f∝= 1 ∝s
Power supply rejection ratio	PSRR	55			dB	f = 50 Hz...20 kHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp} I_{REG} = 0...10 mA

11.4.6 LRTC Linear Voltage Regulator

(Backward current save, that means in times without battery, no current from output cap into the chip.

Table 194 LRTC Performance Specification

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
External capacitor	C_{ext}	-35%	100	+35%	∝F	External capacitor to be connected during test
Capacitor ESR	R_{ESR}			500	Ohm	100 Hz
				1	Ohm	100 kHz ... 10 MHz
	C_{ext}	-35%	1.0	+35%	F	in case of buffering
	R_{ESR}		tbd		Ohm	100 Hz
			tbd		Ohm	100 kHz…10 MHz
	C_{ext}	-35%	220	+35%	nF	always required
	R_{ESR}			500	Ohm	100 Hz
				1	Ohm	100 kHz…10 MHz
Output voltage	V_{REG}	1.86	2.0	2.14	V	V_{DD} = 2.7... 5.5 V I_{REG} = 0 ... 4 mA
Dropout voltage in follower mode	V_{DROP}			0.8	V	in power off state (10 ∝A load) it is possible that VRTC drop's down to 1.86 V with 2.3 V input voltage
Current consumption	I_{DDREG}		8 2	12 5	∝A ∝A	I_{REG} = 2 mA at output in power down (leakage current)
Current limitation		7		18	mA	at all battery voltages between 0 V and 5.5 V Current pulled down from LDO to GND until LDO voltage is 50% of nominal value
Power supply rejection ratio	PSRR	30			dB	f = 50 Hz ... 20 kHz V_{DD} = 2.7 ... 5.5 V I_{REG} = 0...2 mA
Static line regulation	DV_{REG}			10	mV	V_{DD} = 3.1 ... 5.5 V I_{REG} = 2 mA

11.4.7 LD1 Linear Voltage Regulator

Table 195 LD1 Performance Specification

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
External capacitor	C_{ext}	-55%	1.0	+35%	∝F	External capacitor to be connected during test
Capacitor ESR	R_{ESR}			100	Ohm	100 Hz
				0.2	Ohm	100 kHz…10 MHz
Output voltage (includes static parameter)	V_{REG}	-3%	1.5	+3%	V	V_{DD} = 3.1...5.5 V I_{REG} = 0...150 mA
Output voltage (includes static parameter)	V_{REG}	-3%	1.2	+3%	V	V_{DD} = 3.1...5.5 V I_{REG} = 0...150 mA
Output current	I_{REG}			150	mA	
Current Consumption active mode	I_{DDREG}			110	∝A	I_{REG} = 0
				120	∝A	I_{REG} = 1 mA
				750	∝A	I_{REG} = 150 mA
Current Consumption idle mode	I_{DDREG}			55	∝A	I_{REG} = 0
				60	∝A	I_{REG} = 1 mA
				5	∝A	regulator off
Current limitation	I_{MAX}	300		450	mA	Current pulled down from LDO to GND until LDO voltage is 50% of nominal value
Maximum Overshooting	V_{OS}			30	mV	
Startup time	$t_{startup}$			360	∝s	0 -> 90% Iload = 10 mA
Discharge time constant	T_{OFF}		0.25	1	ms	90 -> 5% C = 1.0 uF
Static Parameter						
Line regulation	DV_{REG}			4	mV	V_{DD} = 3.1 ... 5.5 V I_{REG} = 150 mA
Load regulation	DV_{REG}			12	mV	V_{DD} = 3.1 V I_{REG} = 1 ... 150 mA
Dropout voltage	V_{DROP}			880	mV	I_{REG} = 150 mA
Dynamic Parameter						
Line transient response (V_{REG} = 1.5 V)	V_{PEAK}			3	mV	V_{DD} = 3.1 … 3.6 V I_{REG} = 150 mA t_r = 10 ∝s, t_f = 10 ∝s C_{ext} >= 1.0 ∝F-35%

Table 195 LD1 Performance Specification (cont'd)

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
Load transient response (V_{REG} = 1.5 V)	V_{PEAK}			30	mV	V_{DD} = 3.1 V I_{REG} = 1..150 mA t_r = 1 ∝s, t_f = 1 ∝s C_{ext} >= 2.2 ∝F-35%
Power supply rejection ratio (V_{REG} = 1.5 V)	PSRR	30			dB	f = 50 Hz .. 20 kHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp} I_{REG} = 0...150 mA

11.4.8 LSIM Linear Voltage Regulator

Table 196 LSIM Performance Specification

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
External capacitor	C_{ext}	-55%	1	+35%	∝F	External capacitor to be connected during test
Capacitor ESR	R_{ESR}			100	Ohm	100 Hz
				0.2	Ohm	100 kHz…10 MHz
Output voltage	V_{REG}	-3%	2.85	+3%	V	V_{DD} = 3.1 ... 5.5 V I_{REG} = 0 ... 30 mA
Output voltage	V_{REG}	-3%	1.80	+3%	V	V_{DD} = 3.1 ... 5.5 V I_{REG} = 0 ... 30 mA
Output current	I_{REG}			30	mA	
Current limitation	I_{MAX}	100		180	mA	Current pulled down from LDO to GND until LDO voltage is 50% of nominal value
Maximum Overshooting	V_{OS}			15	mV	
Discharge time constant	T_{OFF}			1	ms	90 -> 5% C = 1 ∝F
Static Parameter						
Line regulation (V_{REG} = 2.85 V)	DV_{REG}			7	mV	V_{DD} = 3.1...5.5 V I_{REG} = 30 mA
Load regulation (V_{REG} = 2.85 V)	DV_{REG}			5	mV	V_{DD} = 3.1 V I_{REG} = 1...30 mA
Dropout voltage	V_{DROP}			150	mV	I_{REG} = 30 mA
Dynamic Parameter						
Line transient response	V_{PEAK}			5	mV	V_{DD} = 3.1...3.6 V I_{REG} = 30 mA t_r = 10 ∝s, t_f = 10 ∝s

Table 196 LSIM Performance Specification (cont'd)

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
Load transient response	V_{PEAK}			20	mV	V_{DD} = 3.1 V I_{REG} = 1...30 mA t_r = 1 ∝s, t_f = 1∝s
Power supply rejection ratio (V_{REG} = 2.85 V)	PSRR	35			dB	f = 50 Hz .. 20 kHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp} I_{REG} = 0...30 mA

11.4.9 LMEM Linear Voltage Regulator

Table 197 LMEM Performance Specification

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
External capacitor	C_{ext}	-55%	1	+35%	∝F	External capacitor to be connected during test
Capacitor ESR	R_{ESR}			100	Ohm	100 Hz
				0.2	Ohm	100 kHz…10 MHz
Output voltage	V_{REG}	-3%	2.85	+3%	V	V_{DD} = 3.1...5.5 V I_{REG} = 0...100 mA
		-3%	1.8	+3%	V	
Current Consumption active mode	I_{DDREG}			160	∝A	I_{REG} = 0
				160	∝A	I_{REG} = 1 mA
				700	∝A	I_{REG} = 100 mA
Current Consumption idle mode	I_{DDREG}			100	∝A	I_{REG} = 0
				80	∝A	I_{REG} = 1 mA
				5	∝A	regulator off
Current limitation	I_{MAX}	280		420	mA	Current pulled down from LDO to GND until LDO voltage is 50% of nominal value
Maximum Overshooting	V_{OS}			30	mV	
Startup time	$t_{startup}$			200	∝s	0 -> 90% Iload = 10 mA
Discharge time constant	T_{OFF}			1	ms	90 ... 5 % C = 1.0 ∝F
Static Parameter						
Line regulation	DV_{REG}			5	mV	V_{DD} = 3.1...5.5 V I_{REG} = 100 mA
Load regulation	DV_{REG}			8	mV	V_{DD} = 3.1 V I_{REG} = 1...100 mA
Dropout voltage	V_{DROP}			150	mV	V_{REG} = 2.85 V; I_{REG} = 100 mA
Dynamic Parameter						
Line transient response	V_{PEAK}			5	mV	VDD = 3.1...3.6 V I_{REG} = 100 mA t_r = 10 ∝s, t_f = 10 ∝s
Load transient response	V_{PEAK}			30	mV	V_{DD} = 3.1V I_{REG} = 1..100 mA t_r = 1 ∝s, t_f = 1 ∝s

Table 197 LMEM Performance Specification (cont'd)

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
Power supply rejection ratio (V_{REG} = 1.80V)	PSRR	40	60		dB	f = 50 Hz .. 20 kHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp} I_{REG} = 0...100 mA
Power supply rejection ratio (V_{REG} = 2.85V)	PSRR	40	52		dB	f = 50 Hz .. 20 kHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp} I_{REG} = 0...100 mA

11.4.10 LBUF Linear Voltage Regulator

Table 198 LBUF Performance Specification

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
External capacitor	C_{ext}	-35%	2.2	+35%	µF	External capacitor to be connected during test
Capacitor ESR	R_{ESR}			100	Ohm	100 Hz
				0.2	Ohm	100 kHz…10 MHz
Output voltage	V_{REG}	-3%	2.6	+3%	V	V_{DD} = 3.1...5.5 V I_{REG} = 0...300 mA
Output voltage	V_{REG}	-3%	2.8	+3%	V	V_{DD} = 3.2...5.5 V I_{REG} = 0...300 mA
Output voltage	V_{REG}	-3%	3.0	+3%	V	V_{DD} = 3.4...5.5 V I_{REG} = 0...300 mA
Output voltage	V_{REG}	-3%	3.2	+3%	V	V_{DD} = 3.6...5.5 V I_{REG} = 0...300 mA
Output current	I_{REG}			300	mA	315 mA under typical conditions
Current Consumption	I_{DDREG}		100	270	µA	I_{REG} = 0
				150	µA	I_{REG} = 1 mA
				1000	µA	I_{REG} = 300 mA
Current limitation	I_{MAX}	450		5 600	µA mA	regulator switched off Current pulled down from LDO to GND until LDO voltage is 50% of nominal value
Maximum Overshooting	V_{OS}			30	mV	
Startup time	$t_{startup}$			100	µs	0 -> 90% Iload = 10 mA
Discharge time constant	T_{OFF}			1	ms	90 -> 5% C = 2.2 µF
Static Parameter						

Table 198 LBUF Performance Specification (cont'd)

Parameter	Symbol	Limit Values			Unit	Test Condition/Remark
		Min	Typ	Max		
Line regulation	$-V_{REG}$			5	mV	V_{DD} = 3.1 ... 5.5 V I_{REG} = 300 mA
Load regulation	$-V_{REG}$			5	mV	V_{DD} = 3.1 V I_{REG} = 1 ... 250 mA
Dropout voltage (V_{REG} = 2.6 V)	V_{DROP}			350	mV	I_{REG} = 300 mA
Dropout voltage (V_{REG} = 3.2 V)	V_{DROP}			300	mV	I_{REG} = 300 mA
Dynamic Parameter						
Line transient response	V_{PEAK}			5	mV	V_{DD} = 3.1 … 3.6 V I_{REG} = 300 mA t_r = 10 ∝s, t_f = 10 ∝s
Load transient response (V_{REG} = 2.6 V)	V_{PEAK}			40	mV	V_{DD} = 3.1 V I_{REG} = 1 ... 300 mA t_r = 1 ∝s, t_f = 1 ∝s
Load transient response (V_{REG} = 3.2 V)	V_{PEAK}			40	mV	V_{DD} = 3.1 V I_{REG} = 1 ... 300 mA t_r = 1 ∝s, t_f = 1 ∝s
Power supply rejection ratio	PSRR	45	55		dB	f = 50 Hz ... 20 kHz $V_{DD,DC}$ = 3.35 V $V_{DD,AC}$ = 500 mV_{pp} I_{REG} = 0 ... 300 mA

11.5 Regulator Parameter Definition

11.5.1 Line Regulation and Transient Line Regulation

The line regulation is a static variable that indicates the change in the output voltage of the voltage controller . VREG (at constant load) when there is a change.VDD at the input voltage DVDD. By contrast the line transient response represents dynamic peak value Vpeak to be observed during the change in input voltage. Thermal effects due to changes in the junction temperature are circumvented with pulsed voltage during test and are to be taken into account separately.

The **Figure 214** shows the boundary conditions for tf, tr and VDD to be taken as the basis of the measurement of the line transient response without additional decoupling of the supply voltage by a buffer capacity CBAT. The values defined in the specification apply, however, only in the case of decoupling of the supply voltage with such a capacity CBAT, as a result of which the values for tr and tf are influenced to some extent (on this see information in the relevant tables).

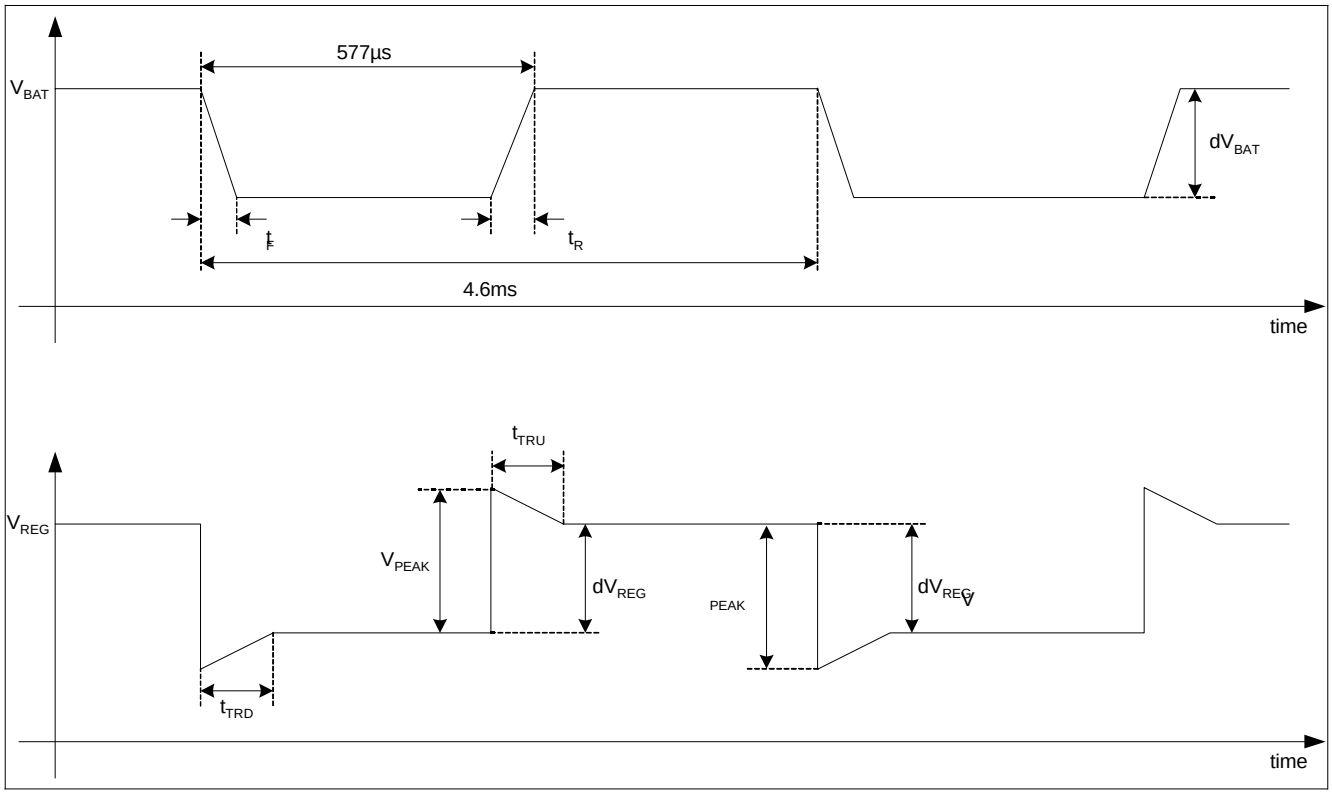


Figure 214 Voltage Pattern for Line Regulation and Transient Line Regulation

11.5.2 Load Regulation and Transient Load Regulation

The load regulation is a static variable which indicates the change in the output voltage of the voltage controller VREG (at constant input voltage) in the event of a change in the load current IREG. By contrast the load transient response represents the dynamic peak value Vpeak to be observed during load variation. Thermal effects due to changes in the junction temperature are circumvented by testing with pulsed load and are to be taken into account separately. The next figure shows the boundary conditions for tf, tr and IREG to be taken as the basis for the measurement of the load transient response.

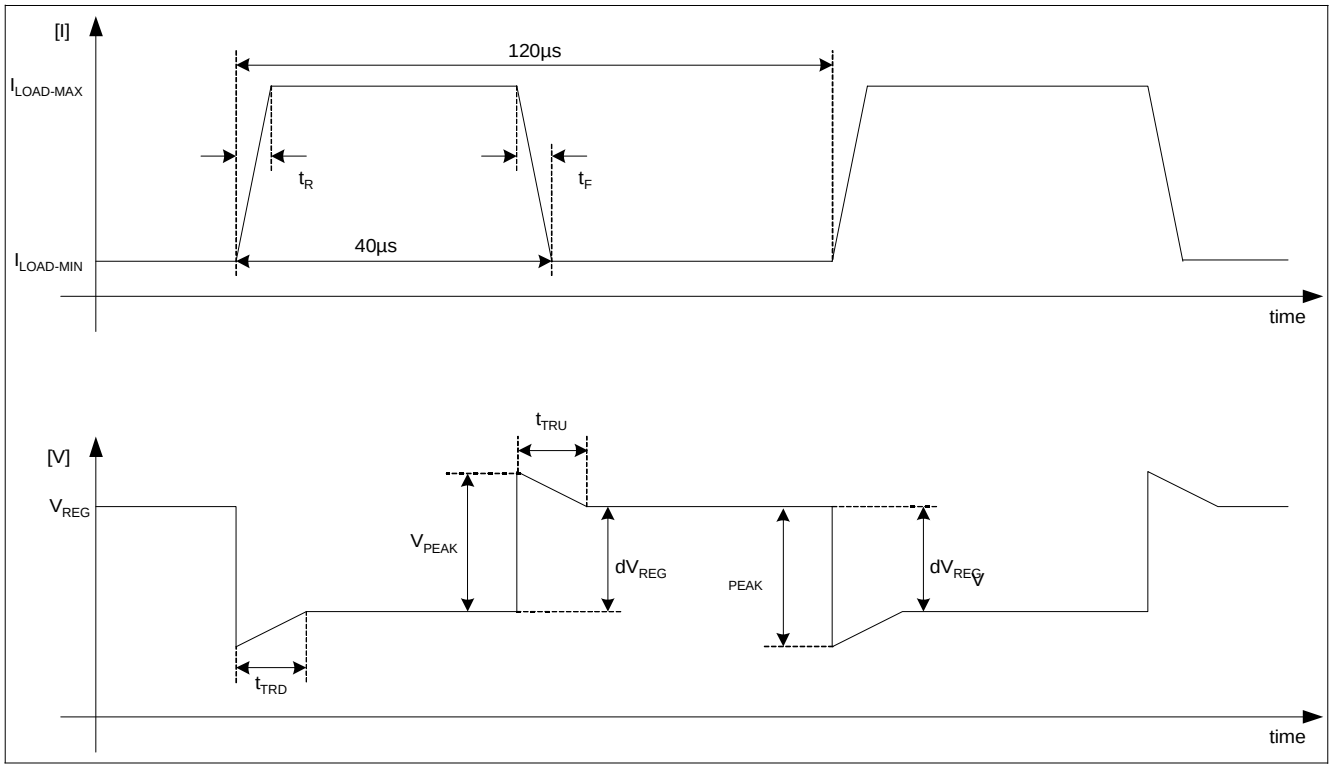


Figure 215 Current Pattern for Load Regulation and Transient Load Regulation

11.5.3 Dropout Definition

The dropout voltage is measured by decreasing the input voltage till the output voltage will drop by 5 mV compared to the output voltage at VBAT = 3.1 V for the LDOs supplied by VBAT. Worst case for dropout is maximum die temperature and maximum current load. This is done statical.

11.5.4 PSRR Definition

PSRR is simulated and measured in large signal over the frequency range: Sinus waveform with 500 mV_{pp} amplitude on a DC level 250 mV above the $VDD_{,min}$ value.

11.6 AC/DC Characteristics of Charger Components

11.6.1 Shunt Regulator

Table 199 Charge Detect Power On

Parameter	Limit Values			Unit	Test Condition/Remark
	Min	Typ	Max		
Supply Voltage / Output Voltage	1.8		2.5	V	
Temperature Range	-30		85	°C	
Shunt Current	10			mA	

11.6.2 Over-Voltage Detection

Table 200 Over-Voltage Detection

Parameter	Limit Values			Unit	Test Condition/Remark
	Min	Typ	Max		
Supply Voltage	0		6.0	V	
Temperature Range	-30		125	°C	
Current Consumption from VBAT		1	5	µA	
Over-voltage Detection Level	-3%	5.5	+3%	V	
Detection Time Constant		2		µs	(spikes below this time are ignored)

Table 201 Charge Switch Driver

Parameter	Limit Values			Unit	Test Condition/Remark
	Min	Typ	Max		
Supply Voltage	0		2.5	V	CDT
Temperature Range	-30		85	°C	
Current Consumption from CDT			100	µA	
Current Consumption from VBAT			1	µA	(pull up current not counted)
Pull-Up Tr. Current		1		mA	@CDT, t.b.d. in system simulation
Sink Current in CS	20			mA	

11.6.3 VBAT Power On

Table 202 VBAT Power On

Parameter	Limit Values			Unit	Test Condition/Remark
	Min	Typ	Max		
Supply Voltage	0		5.5	V	
Temperature Range	-30		125	°C	
Current Consumption from VBAT		1	3	µA	
Power On Detection Level	-8%	2.5	+8%	V	
Hysteresis		40		mV	

Table 202 VBAT Power On (cont'd)

Parameter	Limit Values			Unit	Test Condition/Remark
	Min	Typ	Max		
Start Up Time			1	ms	

11.6.4 Battery Voltage Supervision Unit

Table 203 Battery Voltage Measurement

Parameter	Limit Values			Unit	Test Condition/Remark
	Min	Typ	Max		
Supply Voltage	-7%	2	+7%	V	VRTC
Temperature Range	-30		125	°C	
Current Consumption from VBAT			30	∝A	
Detection Level		1.2		V	
Hysteresis Detection Level		20		mV	
Voltage Divider Level	-2% -7%	2.9 3.1 3.6 4.47 4.6 5.5	+2% +7%	V	2% tolerance with HP Bandgap 7% tolerance with LP Bandgap
Start up time			30	∝s	

11.6.5 Backlight Comparator

Table 204 Backlight Comparator

Parameter	Limit Values			Unit	Test Condition/Remark
	Min	Typ	Max		
Supply Voltage	-3%	2.5V	+3%	V	
Temperature Range	-30		125	°C	
Current Consumption			500	µA	(including voltage divider)
Precision			5	%	(buffer 2%, comp. 2%, r-string 1%, ext. R 1%, Vref 1%)
Hystereses		-	-		no Hysteresis
Standby Current			100	nA	
Delay Time			38	ns	
Detection Level	-2%	150	+2%	mV	
	-2%	200	+2%	mV	
	-2%	300	+2%	mV	

12 System Reset

12.1 Introduction

For the general system reset refer to **Section 8.1.2 “System Power On/Off and Reset Control Logic” (on page 243)** : the PMU generates the reset for the C166S.

This internal system reset function initializes the C166S into a defined state, and can be invoked in the following ways:

- By asserting a hardware reset signal RESET_N (Hardware Reset input)
- On the execution of the SRST instruction (refer to **Section 12.1.2 Software Reset**)
- By an overflow of the watchdog timer (refer to **Section 12.1.3 Watchdog Timer Reset**).

12.1.1 PMU Reset

When the power-on startup sequence is finished, the PMU releases the Baseband reset (refer to **Section 8.1.2.4 Power-up Sequence (on Page 246)** and **Section 8.1.2.11 Power-on and external Resets (on Page 252)**). This power-on reset is visible externally on the RESET_N pad (in output mode).

12.1.2 Software Reset

The reset sequence can be triggered at any time via the protected instruction SRST. The instruction can be either:

- Executed deliberately within a program
- In a hardware trap routine that reveals a system failure.

The software reset performs the same reset functionality as the hardware reset except that the SIM Card block (and its corresponding pins) is not reset by the software reset. The SIM card interface can be reset via the CGU register **RST_CTRL_STA**. This software reset is visible externally on the RESET_N pad (in output mode).

12.1.3 Watchdog Timer Reset

When the watchdog timer is not disabled during the initialization or serviced regularity during program execution, it will overflow and trigger the reset sequence. The watchdog timer resets the same E-GOLDvoice blocks as the software reset and is also visible externally on the RESET_N pad (in output mode).

12.1.4 Reset of RTC, SIM Cards, DSP, and Analog

The Real Time Clock, SIM card, DSP can separately be reset via the CGU register **RST_CTRL_STA** of the XBUS

The Analog part can be reset via the **SCCUSPCR** (Standby Power Control register)

12.1.5 CGU Reset Block

The Hardware reset from signal RESET_IN_n enters the CGU reset block. It is then delayed and re-synchronized on the external oscillator clock (13 MHz or 26 MHz). This signal “reset_in sync_s” is used internally to reset PLL and clock generator.

A delay line of 16 clock cycles is applied to make the signal long enough for the CPU. The delay line is synchronized on the external oscillator. This signal is the CPU C166S hardware reset. It enters the C166 SCU, which generates the reset signal “c166_reset_n”. It has three sources:

1. HWD for hardware reset
2. SW for software reset
3. WDT for watchdog reset.

The reset sequence default duration is 4096 clock cycles. For the SW and WDT resets, the sequence duration can be adjusted to 1024, 2048, or 4096 clock cycles by programming the **RSTCON (on Page 232)** register. The SW reset is generated in the C166S core. The WDT reset is sent to the core when a timer overflow occurs.

The signal “c166_reset_n” is then switched back to the CGU and is used to reset PCL and peripherals on the XBus and PDbus. Therefore, all these devices are still in reset state after the CPU has finished its reset.

Once the reset sequence is finished, the C166S starts its init sequence. At the end of the init sequence (EOI). The signal ex_rstout is switched back to the CGU and is used as a source for the DSP reset. At the end of the init sequence (EOI) of the MCU, the signal ex_rstout is deactivated.

For the DSP there are four different reset sources:

1. Hardware
2. General software (external reset from C166S)
3. Specific software (from a register)
4. A programmable bit from the SCCU (Standby Clock Control Unit).

The reset is then re-synchronized inside DSP subsystem. A flag/interrupt handshake occurs between DSP and CPU to wait for DSP boot (see **Figure 216 (on page 583)**). The DSP is responsible for resetting its peripherals including shared memories.

The Analog part has four different reset sources: one hardware, two software (SW and WDT from C166 SCU) and a programmable bit from the SCCU .

A reset request lines is routed from the SCCU to the CGU. This signal corresponds to bits **SCCUSPCR.ADPN**.

The RTC can only be reset by a specific software reset (**RST_CTRL_STA.RTC_RESET**).

The SIM card is reset using either hardware reset or a specific software reset (**RST_CTRL_STA.SIM_RESET**).

Note: Other devices (such as Timer, ASC, …) do not have specific software resets.

The Pad frame is reset by an external reset except for SIM card, which is reset by either a specific external signal or RTC (Power Isolation).

MON1 and MON2 on PCL are latched on the same reset (cgu_reset_pcl) before the BandGap trimming and the fuse sensing. So those values are Latched in the PCL before being latched respectively in the PMU and in the The PCL has two control signal: EOI (End of Init, the C166S external reset) and EOR (End of Reset, the C166S reset).

12.2 Reset

Table 205 Block Reset Actions

Block	MCU Output	External Reset or PMU power-on reset or PMU SW reset (PMU_GENCTRL.RES)	SW or WDT reset	DSP Standby power	Analog Standby Power	Module Reset
SCCU	Cgu_reset (active low)	R Not affected		Not affected	Not affected	Not affected
DSP subsystem	Cgu_reset_dsp (active low)	R	R	R		Not R
Analog block	Cgu_reset_ana (active low)	R Not affected		R	Not affected R	
interfaces Xbus and PDbus peripherals	Cgu_reset (active low)	R		R Not affected	Not affected	Not affected
RTC	Cgu_reset_rtc (active low)	Not affected	Not affected	Not affected	Not affected	R
SIM card	Cgu_reset_sim (active low)	R Not affected		Not affected	Not affected	R

Table 205 Block Reset Actions (cont'd)

SSC	Cgu_reset	R	R	Not affected	Not affected	Not affected
Timer Unit	Cgu_reset	R	R	Not affected	Not affected	Not affected
GSM unit	Cgu_reset	R	R	Not affected	Not affected	Not affected
Voice	Cgu_reset	R	R	Not affected	Not affected	Not affected
Equalizer Accelerator	Cgu_reset	R	R	Not affected	Not affected	Not affected
TAP	Cgu_reset	R	R	Not affected	Not affected	Not affected
SEIB	Cgu_reset	R	R	Not affected	Not affected	Not affected
Shared memory	From DSP	R	R	R	Not affected	Not affected

12.2.1 Reset Timing Diagrams

Figure 216 represents the reset sequence between the C166 MCU and TEAKlite DSP.

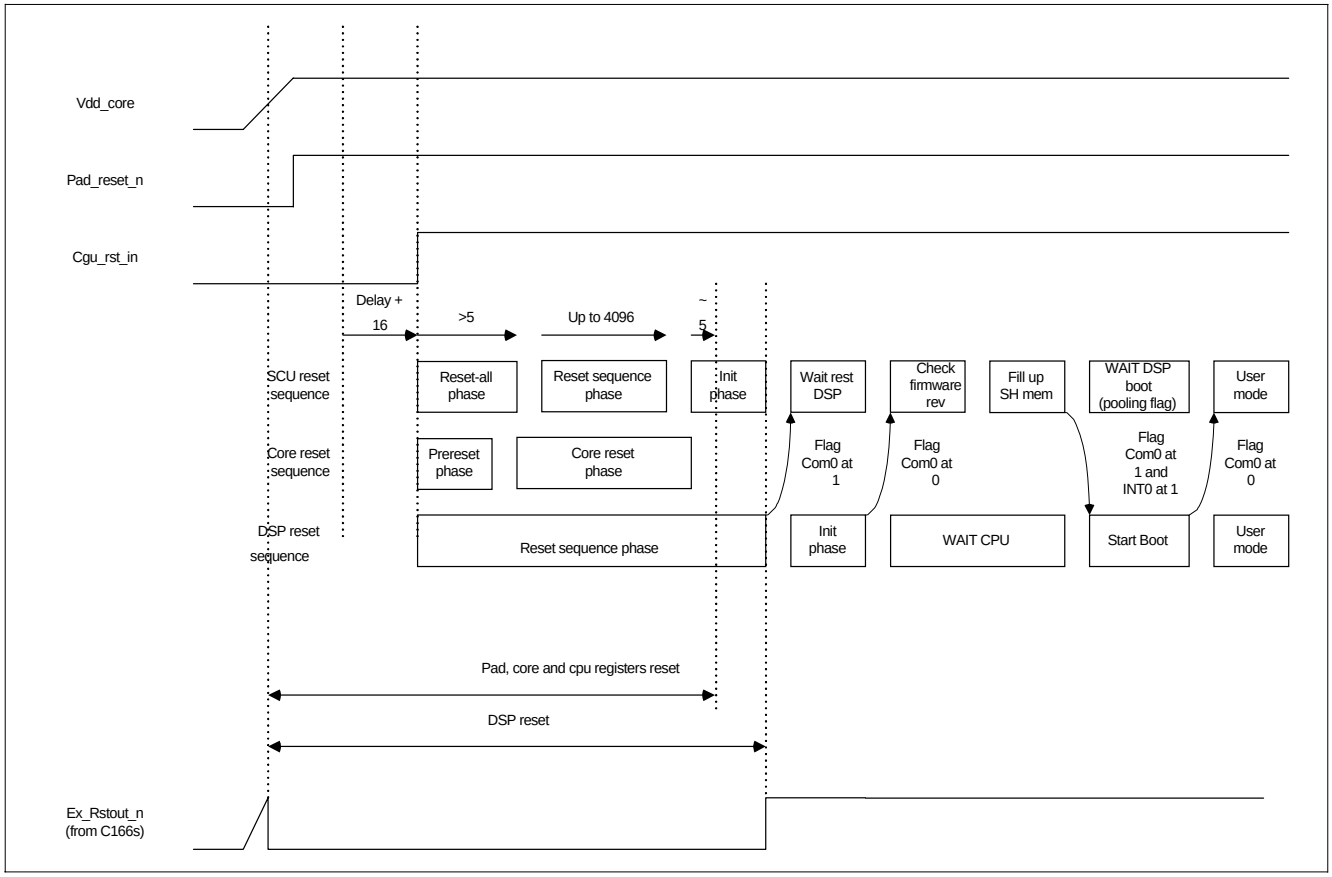


Figure 216 Reset Timing Diagram

After DSP reset deactivation, the software performs a handshake exchange between DSP and CPU using communication flag 0 and interrupt 0. This guaranties that DSP has booted properly before any further treatment takes place.

Figure 217 is an example of a reset interaction timing diagram.

The reset from C166S has a minimum length of 1024 clock cycles. The length can be adjusted using register **RSTCON (on Page 232)** and has a default value of 4096 clock cycles. At the end of the init phase (EOI), the signal RSTOUT is deactivated.

The SW reset length of the SIM, RTC and DSP modules is controlled by software. Each reset signal is activated as soon as the corresponding bit in register

RST_CTRL_STA is set to one. When the bit is reset to zero, the module exits.

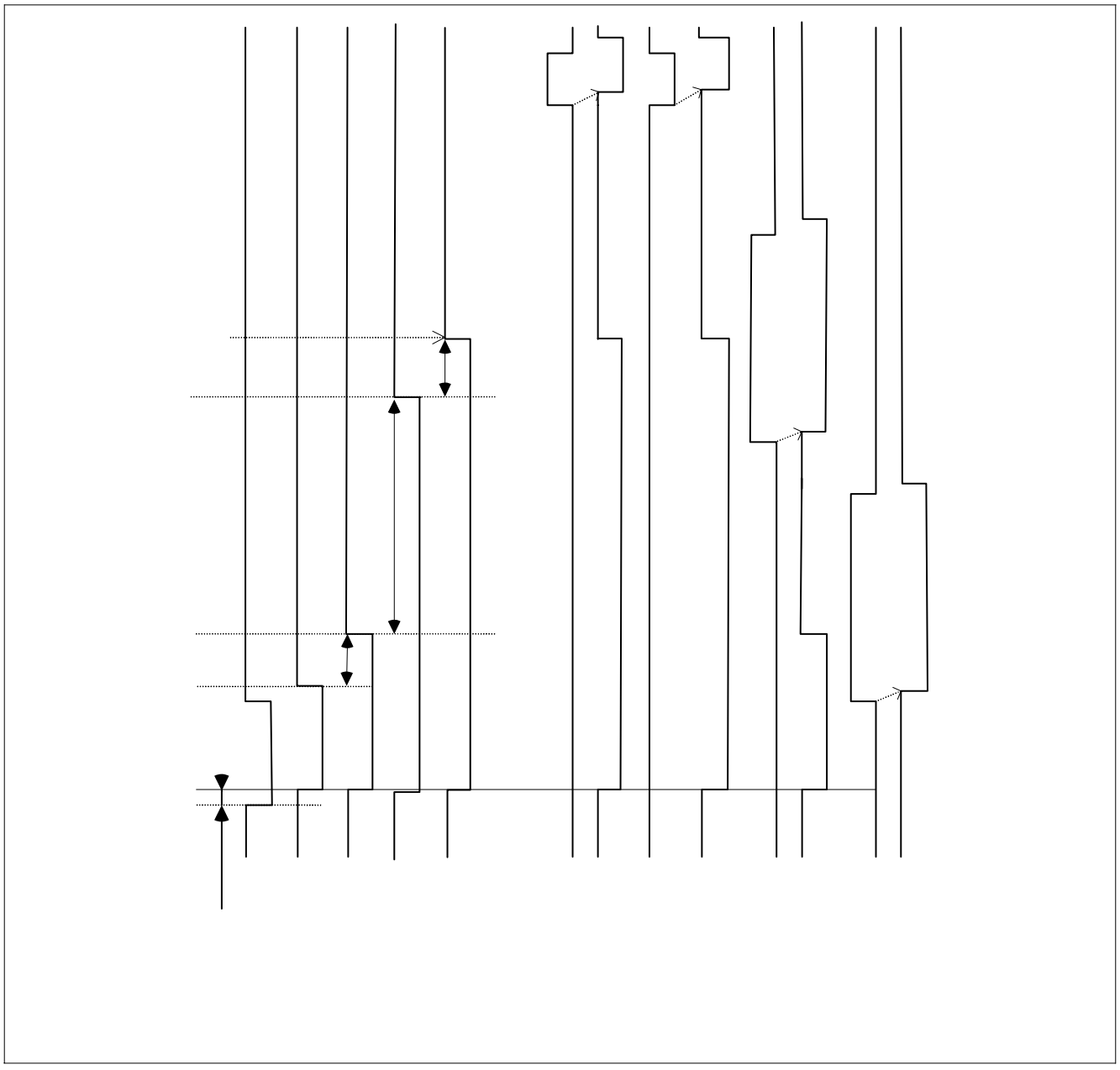


Figure 217 Reset Interaction Timing Diagram Example

12.2.2 Power On/Off Sequences

For the major part of the outputs, the power-off tristate function permits the deactivation of the outputs when proper operation cannot be guaranteed.

This function makes sure that the controlled PMB7880 outputs will never drive an undefined level even when the PMB7880 is switched off and the supply rail of an output pad is powered by an outside power source[1]. It might pull up an PMB7880 output pad to a high level; Current could flow across the ESD protection circuitry of this pad and pull up the voltage of the pad supply rail. Then other pads supplied by the same rail could possibly drive a high output level. By switching all outputs into tristate, this problem can be avoided.

The power-off pad tristate function does not apply to:

- SIM interface pins since special circuitry is provided there
- RTCOUT pin since this has to activate the power management IC
- TDO pin since it is a test pin that has to be available for testing under all conditions and is not used for control functions on the application board.

The power-off tristate function is controlled by the internal signals RESET_BB_n and PM_INT at the PMU/Baseband interface. These signals are supplied by the RTC supply voltage. They retain full functionality even if (the mobile and) PMB7880 is powered off. To put the output pins of PMB7880 in tristate in power-off, these pins have to be controlled appropriately by the power management IC.

Pin PM_INT retains its normal function (PMU interrupt) used by the PMU, as long as pin RESET_BB_n is deasserted. When pin RESET_BB_n is asserted, pin PM_INT can be used to set all outputs to tristate as shown in **Table 206**. As long as input RESET_BB_n is deasserted, the outputs are enabled.

The state of the outputs is controlled by the internal control signal PRG1 which is asserted whenever the outputs are tristated.

Table 206 Power-Off Output Tristate Control

input RESET_BB_n	input PM_INT		Output State	Internal Control Signal PRG1
	Function	Value		
0	Output tristate control	0	Enabled	0
		1	Tristate	1
1	Interrupt	X	Enabled	0

The appropriate control sequence for signals PM_INT and RESET_BB_n is described in **Section 12.2.2.3 Power-Up (on Page 586)** for power-up transitions and **Section 12.2.2.1 Power-Off (on Page 585)** for power-off transitions.

PM_INT, RESET_BB_n, and RTC_OUT have the specified IO functionality only if both:

- VDD_RTC is in the Operating Range 1.8 V < VDD_RTC < 2.25 V
- The other VDDs (such as VDD_LD1) are also in operating range.

Note: If VDD_RTC < 1.8 V and VDD_LD1 and VDD_LD1 are turned off, the functionality of these pins is not specified.

12.2.2.1 Power-Off

Power-off sequence and timings are described in the PMU chapter **Section 8.1.2.5 “Turn Off” (on page 248)**.

1) This power source might be an accessory attached to the powered down handset.

12.2.2.2 Insertion of Battery

Battery insertion and power-on sequences and timings are described in the PMU chapter **Section 8.1.2.2 “Insertion of Battery” (on page 243)** and **Section 8.1.2.4 Power-up Sequence (on Page 246)**.

When the battery is inserted into the handset, the PMU has to power-up the RTC regulator and drive input signal PM_INT high and input signal RESET_BB_n low to keep the outputs with the tristate function enabled.

12.2.2.3 Power-Up

Power-on sequence and timings are described in the PMU chapter **Section 8.1.2.4 Power-up Sequence (on Page 246)**.

12.2.3 CPU Boot Configuration

The basic PMB7880 CPU configuration setting (boot from internal ROM) is defined during **HW** RESET by the input levels on the pin MON2 (see **Figure 218**). If the pin is not connected, then internal pull-downs ensure that the system starts with the default configuration, internal boot. This can be overridden by pulling-up the MON2 pin during the power-up sequence (for example, via external pull-up resistors), MON2 = 1 selects boot from external ROM.

The External memory voltage (if MON1=0 then LMEM output voltage is 1.8v /if MON1=1 then LMEM output voltage is 2.85v is defined during **HW** RESET by the input levels on the pin MON1 (see **Figure 218**).

If the pin is not connected, then internal pull-downs ensure that the system starts with LMEM output voltage at 1.8v.

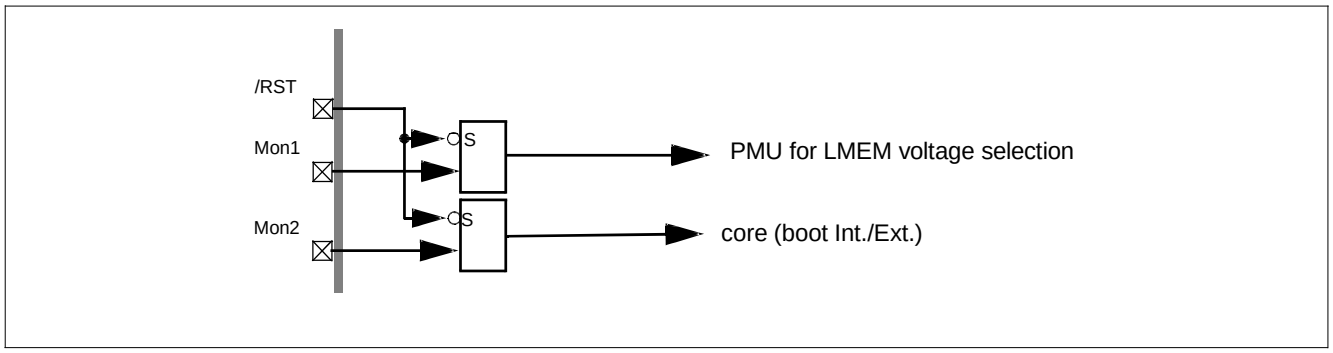


Figure 218 Boot Configuration Latch Register

13 Debug

This chapter contains links to the different debug features used in the E-GOLDvoice:

- Section 9.8.2 Break Switch (on Page 466)

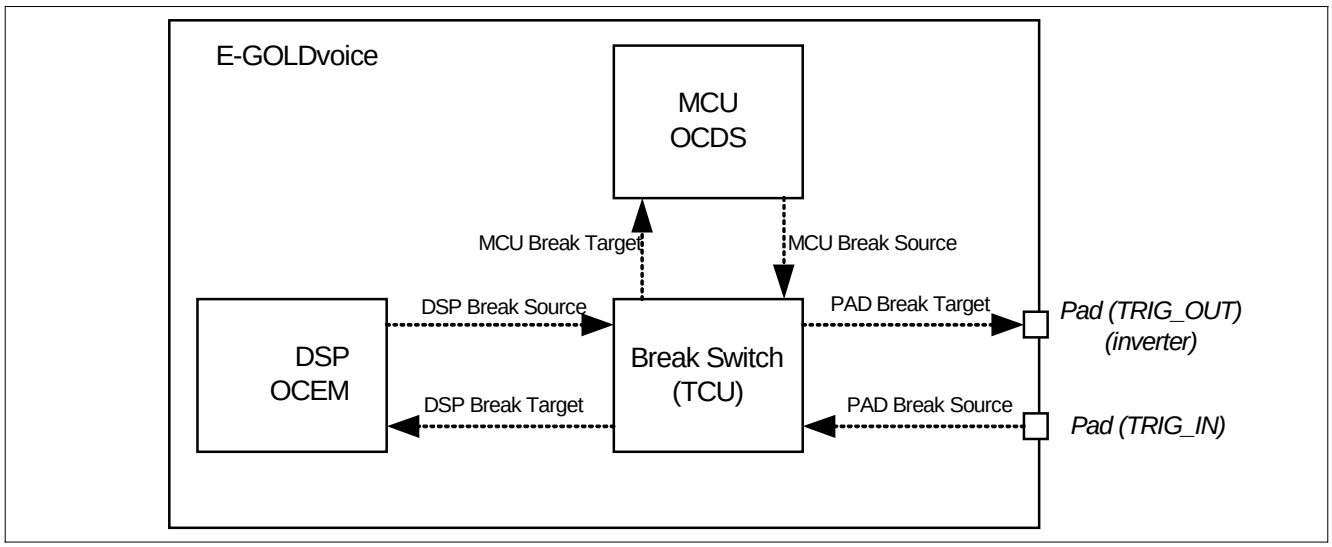


Figure 219 Break Switch Interface

- Section 9.8.3 JTAG (on Page 469)

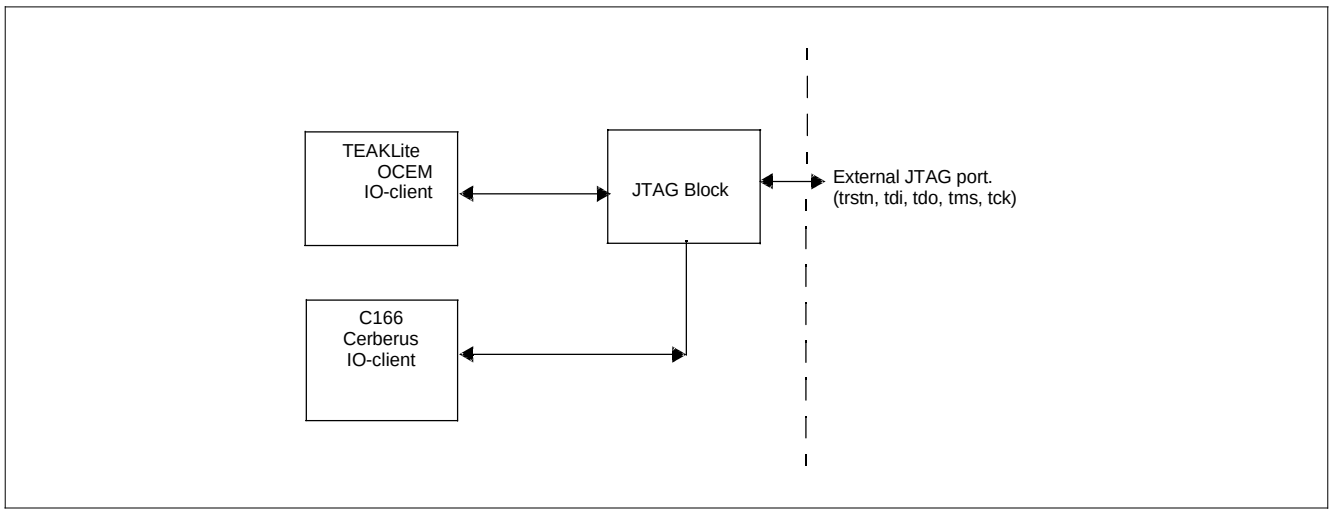


Figure 220 JTAG IO Mode Application Example

- Section 7.8 OCDS (on Page 262)
- Section 8.11 OCEM/SEIB (on Page 512)
- Section 7.9 Cerberus (on Page 276)
- Section 9.7.10 Internal Signal Monitoring (on Page 435)
- Section 6.7 DSP Debug Register (on Page 116)
- Section 6.8 Pad Access Register (on Page 118).

Appendix A Glossary

Glossary of Terms

Term	Description
4PPM	Four Pulse Position Modulation
AFC	Adaptive Frequency Correction Automatic Frequency Correction
ALU	Arithmetic Logic Unit
AMR	Adaptive Multi-Rate
ASC	Asynchronous Serial Interface Controller
BFO	Bit Field Operations
CAPCOM	Capture Compare
CGU	Clock Generation Unit
CPS	CPU Slave
CPU	Control Processing Unit
CRC	Cyclic Redundancy Check
DAC	Digital Analog Converter
DAI	Digital Audio Interface
DSP	Digital Signal Processor
DTX	Discontinues Transmission
EBU	External Bus Interface Unit
EDGE	Enhanced Data Rates for GSM Evolution
EFR	Enhanced Full-Rate
EGPRS	Enhanced GPRS
FCS	Frame Check Sequence
FIR	Fast Infrared Mode
FR	Full Rate
GEA	GPRS Encryption Algorithm
GMSK	Gauss Minimum Shift Keying
GPIO	General Purpose IO cell
GPRS	General Packet Radio Service
GPTU	General Purpose Timer Unit
GSM	Global System for Mobile communication
HR	Half Rate
HSCSD	High Speed Circuit Switched Data
I2C	Inter IC
I2S	Inter IC Sound
IDMX	Input demultiplex
IrDA	Infrared Data Association
IrLAP	Infrared Link Access Protocol
ISA	Instruction Set Architecture
JTAG	Joint Test Action Group
LLC	Logical Link Control
LMB	Local Memory Bus

Glossary of Terms

Term	Description
LNA	Low Noise Amplifier
MAC	Multiply Accumulate Module
MCS	Modulation and Coding Scheme
MCU	Micro Controller Unit
MIR	Medium Infrared Mode
MMU	Memory Management Unit
MPU	Micro Processor Unit
TEAKLite	DSP Core
OCDS	On Chip Debug Support
OCEM	On Chip Emulation Mode
PCL	Port Control Logic
PCP	Peripheral Control Processor
PEC	Peripheral Event Controller
PSK	Phase Shift Keying
RTC	Real Time Clock
RTOS	Real-Time Operating System
RZI	Return to Zero Inverted SAPP
	Sound APPlication
SCCU	Standby Clock Control Unit
SCU	System Control Unit
SEIB	System Emulation In Board
SIM	Subscriber Identity Module
SIR	Serial Infrared Mode
SPI	Serial Peripheral Interface
SRR	Service Request
SSC	Serial Synchronous Interface Controller
STM	System Timer Module
TAP	Test Access Port
TCL	Half of a MCU clock period
TDMA	Time Division Multiple Access
TOS	Type of Service
UART	Universal Asynchronous Receiver Transmitter
USART	Universal Synchronous Asynchronous Receiver Transmitter
USB	Universal Serial Bus
USF	Uplink Status Flag
VAD	Voice Activity Detection
XAB	X Address Bus
YAB	Y Address Bus
ZAB	Z Address Bus
XDB	X Data Bus
YDB	Y Data Bus

Glossary of Terms

Term	Description
ZDB	Z Data Bus
PAB	Program Address Bus
PDB	Program Data Bus

www.infineon.com

Published by Infineon Technologies AG