



Welcome to [E-XFL.COM](#)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Obsolete
Core Processor	C166SV2
Core Size	16-Bit
Speed	20MHz
Connectivity	CANbus, SPI, UART/USART
Peripherals	PWM, WDT
Number of I/O	47
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 2.7V
Data Converters	A/D 14x8/10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	PG-LQFP-64-4
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/saf-xc164cm-16f20f-ba

2 General Device Information

The XC164CM derivatives are high-performance members of the Infineon XC166 Family of full featured single-chip CMOS microcontrollers. These devices extend the functionality and performance of the C166 Family in terms of instructions (MAC unit), peripherals, and speed. They combine high CPU performance (up to 40 million instructions per second) with high peripheral functionality and enhanced IO-capabilities. They also provide clock generation via PLL and various on-chip memory modules such as program Flash, program RAM, and data RAM.

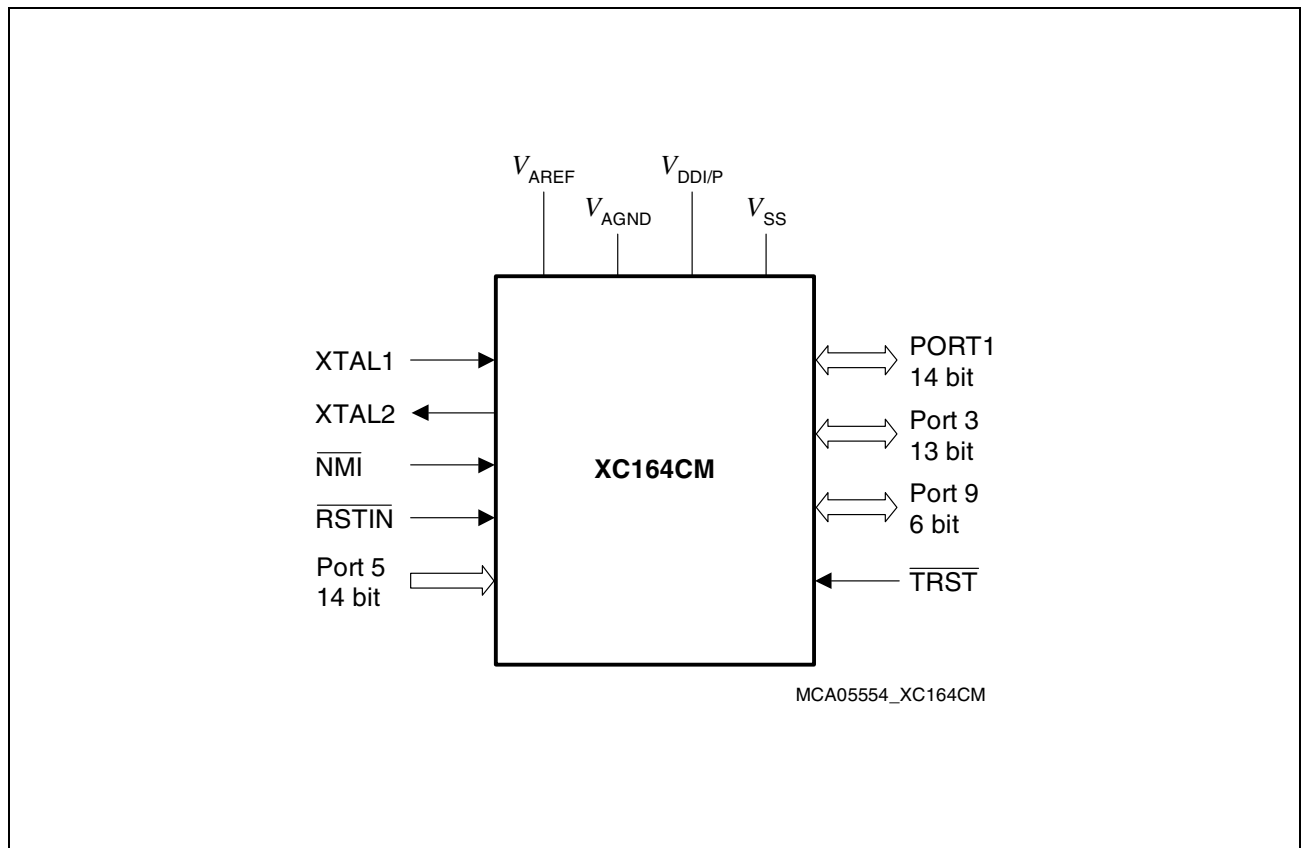


Figure 1 Logic Symbol

General Device Information
Table 2 Pin Definitions and Functions (cont'd)

Sym- bol	Pin Num.	Input Outp.	Function
Port 3	28-39, 42	IO	Port 3 is a 13-bit bidirectional I/O port. Each pin can be programmed for input (output driver in high-impedance state) or output (configurable as push/pull or open drain driver). The input threshold of Port 3 is selectable (standard or special). The following Port 3 pins also serve for alternate functions:
P3.1	28	O I/O I I	T6OUT: [GPT2] Timer T6 Toggle Latch Output, RxD1: [ASC1] Data Input (Async.) or Inp./Outp. (Sync.), EX1IN: [Fast External Interrupt 1] Input (alternate pin A), TCK: [Debug System] JTAG Clock Input
P3.2	29	I I	CAPIN: [GPT2] Register CAPREL Capture Input, TDI: [Debug System] JTAG Data In
P3.3	30	O O	T3OUT: [GPT1] Timer T3 Toggle Latch Output, TDO: [Debug System] JTAG Data Out
P3.4	31	I I	T3EUD: [GPT1] Timer T3 External Up/Down Control Input, TMS: [Debug System] JTAG Test Mode Selection
P3.5	32	I O O	T4IN: [GPT1] Timer T4 Count/Gate/Reload/Capture Inp. TxD1: [ASC0] Clock/Data Output (Async./Sync.), BRKOUT: [Debug System] Break Out
P3.6	33	I	T3IN: [GPT1] Timer T3 Count/Gate Input
P3.7	34	I I	T2IN: [GPT1] Timer T2 Count/Gate/Reload/Capture Inp. BRKIN: [Debug System] Break In
P3.8	35	I/O	MRST0: [SSC0] Master-Receive/Slave-Transmit In/Out.
P3.9	36	I/O	MTSR0: [SSC0] Master-Transmit/Slave-Receive Out/In.
P3.10	37	O I	TxD0: [ASC0] Clock/Data Output (Async./Sync.), EX2IN: [Fast External Interrupt 2] Input (alternate pin B)
P3.11	38	I/O I	RxD0: [ASC0] Data Input (Async.) or Inp./Outp. (Sync.), EX2IN: [Fast External Interrupt 2] Input (alternate pin A)
P3.13	39	I/O I	SCLK0: [SSC0] Master Clock Output / Slave Clock Input., EX3IN: [Fast External Interrupt 3] Input (alternate pin A)
P3.15	42	O O	CLKOUT: System Clock Output (= CPU Clock), FOUT: Programmable Frequency Output

General Device Information
Table 2 Pin Definitions and Functions (cont'd)

Sym- bol	Pin Num.	Input Outp.	Function
XTAL2 XTAL1	61 60	O I	<p>XTAL2: Output of the oscillator amplifier circuit XTAL1: Input to the oscillator amplifier and input to the internal clock generator To clock the device from an external source, drive XTAL1, while leaving XTAL2 unconnected. Minimum and maximum high/low and rise/fall times specified in the AC Characteristics must be observed.</p> <p><i>Note: Input pin XTAL1 belongs to the core voltage domain. Therefore, input voltages must be within the range defined for V_{DDI}.</i></p>
V_{AREF}	19	–	Reference voltage for the A/D converter
V_{AGND}	20	–	Reference ground for the A/D converter
V_{DDI}	26, 58	–	<p>Digital Core Supply Voltage (On-Chip Modules): +2.5 V during normal operation and idle mode. Please refer to the Operating Condition Parameters</p>
V_{DDP}	8, 27, 40, 57	–	<p>Digital Pad Supply Voltage (Pin Output Drivers): +5 V during normal operation and idle mode. Please refer to the Operating Condition Parameters</p>
V_{SS}	7, 25, 41, 59	–	<p>Digital Ground Connect decoupling capacitors to adjacent V_{DD}/V_{SS} pin pairs as close as possible to the pins. All V_{SS} pins must be connected to the ground-line or ground-plane.</p>

1) The CAN interface lines are assigned to port P9 under software control.

Functional Description

2 Kbytes of on-chip Dual-Port RAM (DPRAM) are provided as a storage for user defined variables, for the system stack, general purpose register banks. A register bank can consist of up to 16 word wide (R0 to R15) and/or byte wide (RL0, RH0, ..., RL7, RH7) so-called General Purpose Registers (GPRs).

The upper 256 bytes of the DPRAM are directly bit addressable. When used by a GPR, any location in the DPRAM is bit addressable.

1024 bytes (2 × 512 bytes) of the address space are reserved for the Special Function Register areas (SFR space and ESFR space). SFRs are word wide registers which are used for controlling and monitoring functions of the different on-chip units. Unused SFR addresses are reserved for future members of the XC166 Family. Therefore, they should either not be accessed, or written with zeros, to ensure upward compatibility.

Table 3 XC164CM Memory Map

Address Area	Start Loc.	End Loc.	Area Size ¹⁾	Notes
Flash register space	FF'F000 _H	FF'FFFF _H	4 Kbytes	2)
Reserved (Acc. trap)	F8'0000 _H	FF'FFFF _H	508 Kbytes	–
Reserved for PSRAM	E0'0800 _H	F7'FFFF _H	< 1.5 Mbytes	Minus PSRAM
Program SRAM	E0'0000 _H	E0'07FF _H	2 Kbytes	–
Reserved for pr. mem.	C2'0000 _H	DF'FFFF _H	< 2 Mbytes	Minus Flash
Program Flash	C0'0000 _H	C1'FFFF _H	128 Kbytes	XC164CM-16F
	C0'0000 _H	C0'FFFF _H	64 Kbytes	XC164CM-8F
	C0'0000 _H	C0'7FFF _H	32 Kbytes	XC164CM-4F
Reserved	20'0800 _H	BF'FFFF _H	< 10 Mbytes	Minus TwinCAN
TwinCAN registers	20'0000 _H	20'07FF _H	2 Kbytes	Accessed via EBC
Reserved	01'0000 _H	1F'FFFF _H	< 2 Mbytes	Minus segment 0
SFR area	00'FE00 _H	00'FFFF _H	0.5 Kbyte	–
Dual-Port RAM	00'F600 _H	00'FDFF _H	2 Kbytes	–
Reserved for DPRAM	00'F200 _H	00'F5FF _H	1 Kbyte	–
ESFR area	00'F000 _H	00'F1FF _H	0.5 Kbyte	–
XSFR area	00'E000 _H	00'FFFF _H	4 Kbytes	–
Reserved	00'D000 _H	00'DFFF _H	6 Kbytes	–
Data SRAM	00'C000 _H	00'CFFF _H	4 Kbytes	3)
Reserved for DSRAM	00'8000 _H	00'BFFF _H	16 Kbytes	–
Reserved	00'0000 _H	00'7FFF _H	32 Kbytes	–

1) The areas marked with "<" are slightly smaller than indicated, see column "Notes".

Functional Description

- 2) Not defined register locations return a trap code (1E9B_H).
- 3) Depends on the respective derivative. See [Table 1 “XC164CM Derivative Synopsis” on Page 6](#).

3.2 Central Processing Unit (CPU)

The main core of the CPU consists of a 5-stage execution pipeline with a 2-stage instruction-fetch pipeline, a 16-bit arithmetic and logic unit (ALU), a 32-bit/40-bit multiply and accumulate unit (MAC), a register-file providing three register banks, and dedicated SFRs. The ALU features a multiply and divide unit, a bit-mask generator, and a barrel shifter.

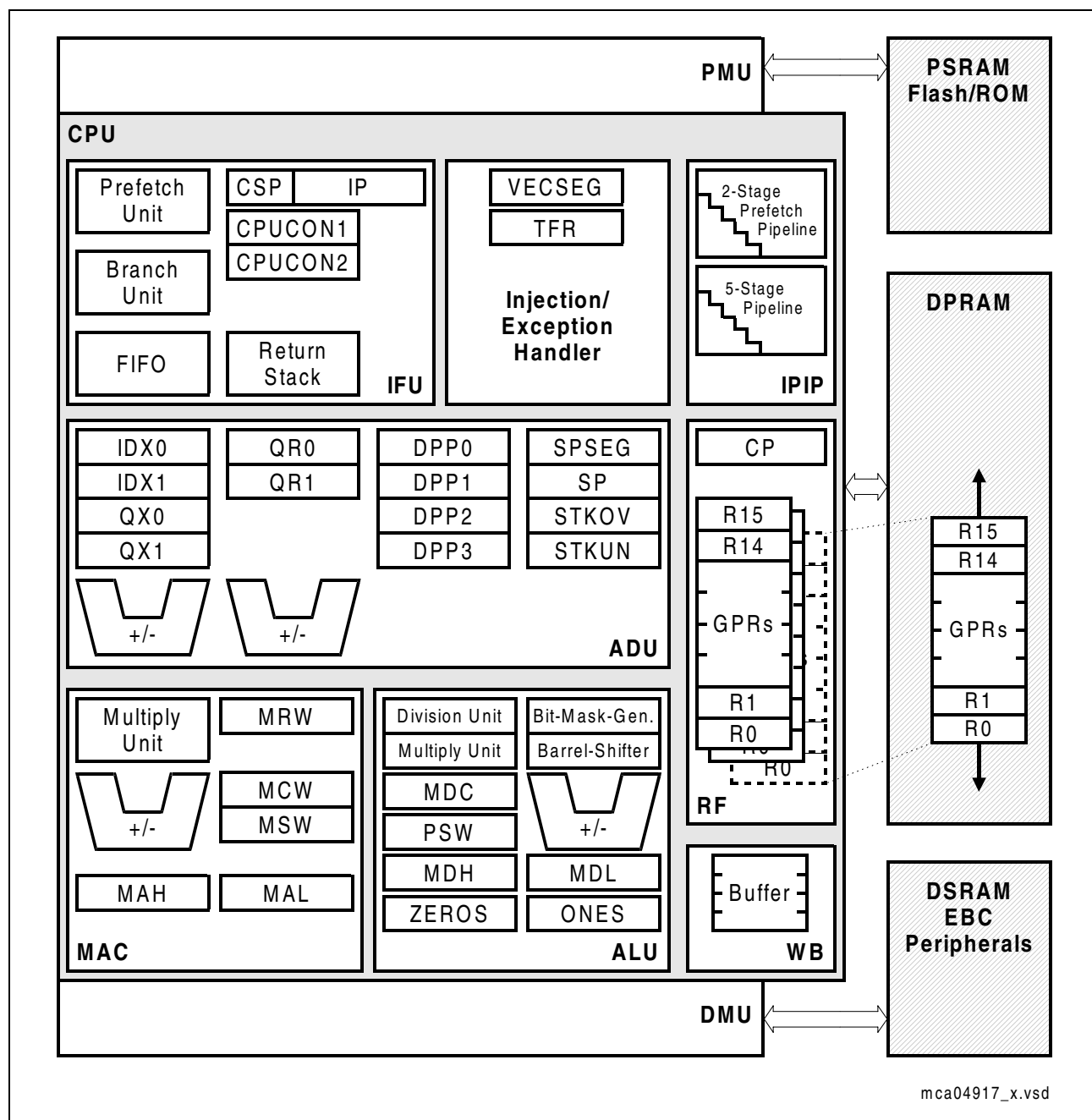


Figure 4 CPU Block Diagram

Based on these hardware provisions, most of the XC164CM's instructions can be executed in just one machine cycle which requires 25 ns at 40 MHz CPU clock. For

Functional Description

example, shift and rotate instructions are always processed during one machine cycle independent of the number of bits to be shifted. Also multiplication and most MAC instructions execute in one single cycle. All multiple-cycle instructions have been optimized so that they can be executed very fast as well: for example, a 32-/16-bit division is started within 4 cycles, while the remaining 15 cycles are executed in the background. Another pipeline optimization, the branch target prediction, allows eliminating the execution time of branch instructions if the prediction was correct.

The CPU has a register context consisting of up to three register banks with 16 word wide GPRs each at its disposal. One of these register banks is physically allocated within the on-chip DPRAM area. A Context Pointer (CP) register determines the base address of the active register bank to be accessed by the CPU at any time. The number of register banks is only restricted by the available internal RAM space. For easy parameter passing, a register bank may overlap others.

A system stack of up to 32 Kwords is provided as a storage for temporary data. The system stack can be allocated to any location within the address space (preferably in the on-chip RAM area), and it is accessed by the CPU via the stack pointer (SP) register. Two separate SFRs, STKOV and STKUN, are implicitly compared against the stack pointer value upon each stack access for the detection of a stack overflow or underflow.

The high performance offered by the hardware implementation of the CPU can efficiently be utilized by a programmer via the highly efficient XC164CM instruction set which includes the following instruction classes:

- Standard Arithmetic Instructions
- DSP-Oriented Arithmetic Instructions
- Logical Instructions
- Boolean Bit Manipulation Instructions
- Compare and Loop Control Instructions
- Shift and Rotate Instructions
- Prioritize Instruction
- Data Movement Instructions
- System Stack Instructions
- Jump and Call Instructions
- Return Instructions
- System Control Instructions
- Miscellaneous Instructions

The basic instruction length is either 2 or 4 bytes. Possible operand types are bits, bytes and words. A variety of direct, indirect or immediate addressing modes are provided to specify the required operands.

3.3 Interrupt System

With an interrupt response time of typically 8 CPU clocks (in case of internal program execution), the XC164CM is capable of reacting very fast to the occurrence of non-deterministic events.

The architecture of the XC164CM supports several mechanisms for fast and flexible response to service requests that can be generated from various sources internal or external to the microcontroller. Any of these interrupt requests can be programmed to being serviced by the Interrupt Controller or by the Peripheral Event Controller (PEC).

In contrast to a standard interrupt service where the current program execution is suspended and a branch to the interrupt vector table is performed, just one cycle is 'stolen' from the current CPU activity to perform a PEC service. A PEC service implies a single byte or word data transfer between any two memory locations with an additional increment of either the PEC source, or the destination pointer, or both. An individual PEC transfer counter is implicitly decremented for each PEC service except when performing in the continuous transfer mode. When this counter reaches zero, a standard interrupt is performed to the corresponding source related vector location. PEC services are very well suited, for example, for supporting the transmission or reception of blocks of data. The XC164CM has 8 PEC channels each of which offers such fast interrupt-driven data transfer capabilities.

A separate control register which contains an interrupt request flag, an interrupt enable flag and an interrupt priority bit field exists for each of the possible interrupt nodes. Via its related register, each node can be programmed to one of sixteen interrupt priority levels. Once having been accepted by the CPU, an interrupt service can only be interrupted by a higher prioritized service request. For the standard interrupt processing, each of the possible interrupt nodes has a dedicated vector location.

Fast external interrupt inputs are provided to service external interrupts with high precision requirements. These fast interrupt inputs feature programmable edge detection (rising edge, falling edge, or both edges).

Software interrupts are supported by means of the 'TRAP' instruction in combination with an individual trap (interrupt) number.

Table 4 shows all of the possible XC164CM interrupt sources and the corresponding hardware-related interrupt flags, vectors, vector locations and trap (interrupt) numbers.

Note: Interrupt nodes which are not assigned to peripherals (unassigned nodes), may be used to generate software controlled interrupt requests by setting the respective interrupt request bit (xIR).

Functional Description

register in response to an external event at the port pin which is associated with this register. In addition, a specific interrupt request for this capture/compare register is generated. Either a positive, a negative, or both a positive and a negative external signal transition at the pin can be selected as the triggering event.

The contents of all registers which have been selected for one of the five compare modes are continuously compared with the contents of the allocated timers.

When a match occurs between the timer value and the value in a capture/compare register, specific actions will be taken based on the selected compare mode.

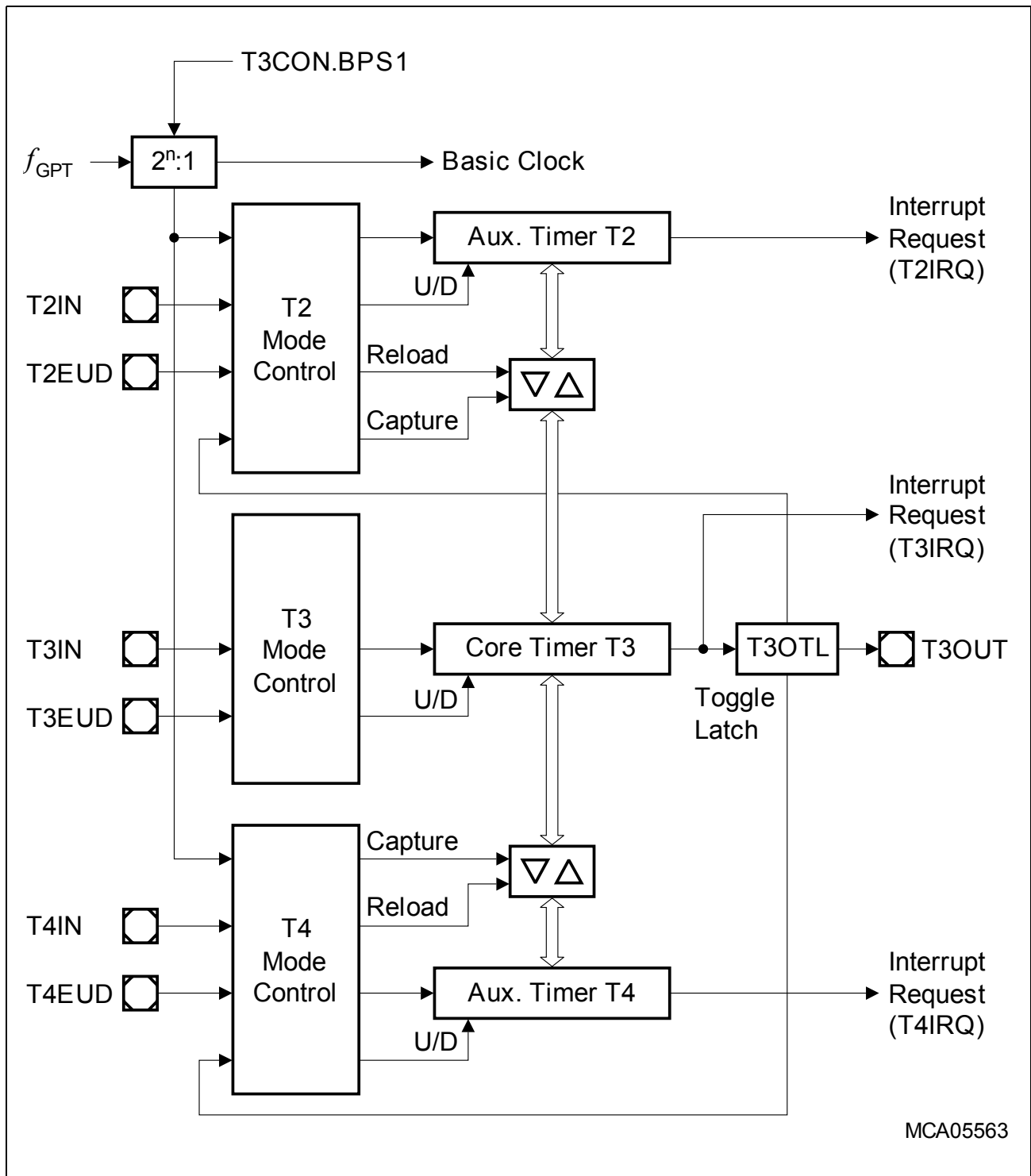


Figure 7 Block Diagram of GPT1

With its maximum resolution of 2 system clock cycles, the **GPT2 module** provides precise event control and time measurement. It includes two timers (T5, T6) and a capture/reload register (CAPREL). Both timers can be clocked with an input clock which is derived from the CPU clock via a programmable prescaler or with external signals. The

Functional Description

count direction (up/down) for each timer is programmable by software or may additionally be altered dynamically by an external signal on a port pin (TxEUD). Concatenation of the timers is supported via the output toggle latch (T6OTL) of timer T6, which changes its state on each timer overflow/underflow.

The state of this latch may be used to clock timer T5, and/or it may be output on pin T6OUT. The overflows/underflows of timer T6 can additionally be used to clock the CAPCOM2 timers, and to cause a reload from the CAPREL register.

The CAPREL register may capture the contents of timer T5 based on an external signal transition on the corresponding port pin (CAPIN), and timer T5 may optionally be cleared after the capture procedure. This allows the XC164CM to measure absolute time differences or to perform pulse multiplication without software overhead.

The capture trigger (timer T5 to CAPREL) may also be generated upon transitions of GPT1 timer T3's inputs T3IN and/or T3EUD. This is especially advantageous when T3 operates in Incremental Interface Mode.

3.8 Real Time Clock

The Real Time Clock (RTC) module of the XC164CM is directly clocked via a separate clock driver with the prescaled on-chip main oscillator frequency ($f_{\text{RTC}} = f_{\text{OSCM}}/32$). It is therefore independent from the selected clock generation mode of the XC164CM.

The RTC basically consists of a chain of divider blocks:

- A selectable 8:1 divider (on - off)
- The reloadable 16-bit timer T14
- The 32-bit RTC timer block (accessible via registers RTCH and RTCL), made of:
 - a reloadable 10-bit timer
 - a reloadable 6-bit timer
 - a reloadable 6-bit timer
 - a reloadable 10-bit timer

All timers count up. Each timer can generate an interrupt request. All requests are combined to a common node request.

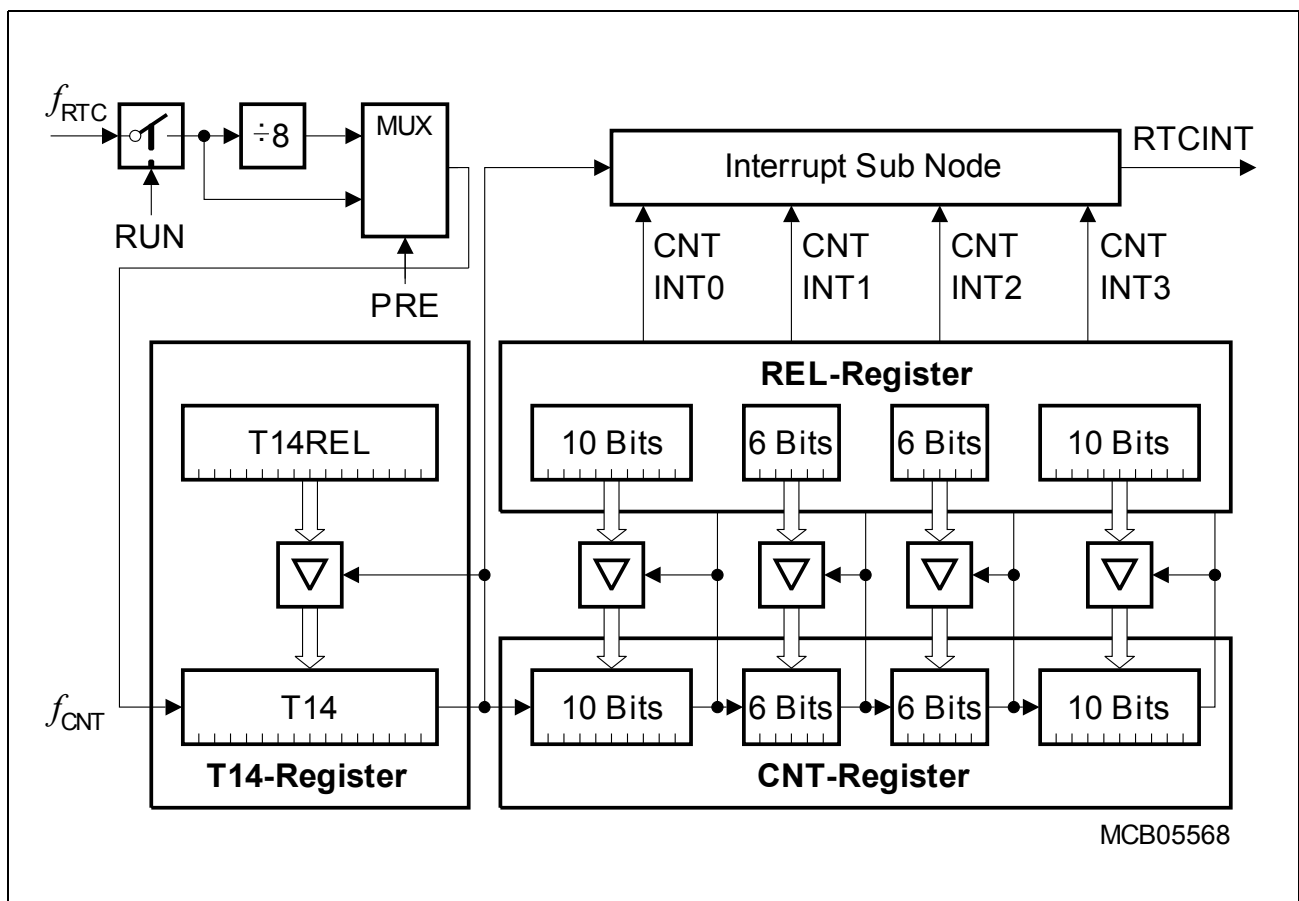


Figure 9 RTC Block Diagram

Note: The registers associated with the RTC are not affected by a reset in order to maintain the correct system time even when intermediate resets are executed.

3.12 TwinCAN Module

The integrated TwinCAN module handles the completely autonomous transmission and reception of CAN frames in accordance with the CAN specification V2.0 part B (active), i.e. the on-chip TwinCAN module can receive and transmit standard frames with 11-bit identifiers as well as extended frames with 29-bit identifiers.

Two Full-CAN nodes share the TwinCAN module's resources to optimize the CAN bus traffic handling and to minimize the CPU load. The module provides up to 32 message objects, which can be assigned to one of the CAN nodes and can be combined to FIFO-structures. Each object provides separate masks for acceptance filtering.

The flexible combination of Full-CAN functionality and FIFO architecture reduces the efforts to fulfill the real-time requirements of complex embedded control applications. Improved CAN bus monitoring functionality as well as the number of message objects permit precise and comfortable CAN bus traffic handling.

Gateway functionality allows automatic data exchange between two separate CAN bus systems, which reduces CPU load and improves the real time behavior of the entire system.

The bit timing for both CAN nodes is derived from the master clock and is programmable up to a data rate of 1 Mbit/s. Each CAN node uses two pins of Port 9 to interface to an external bus transceiver. The interface pins are assigned via software.

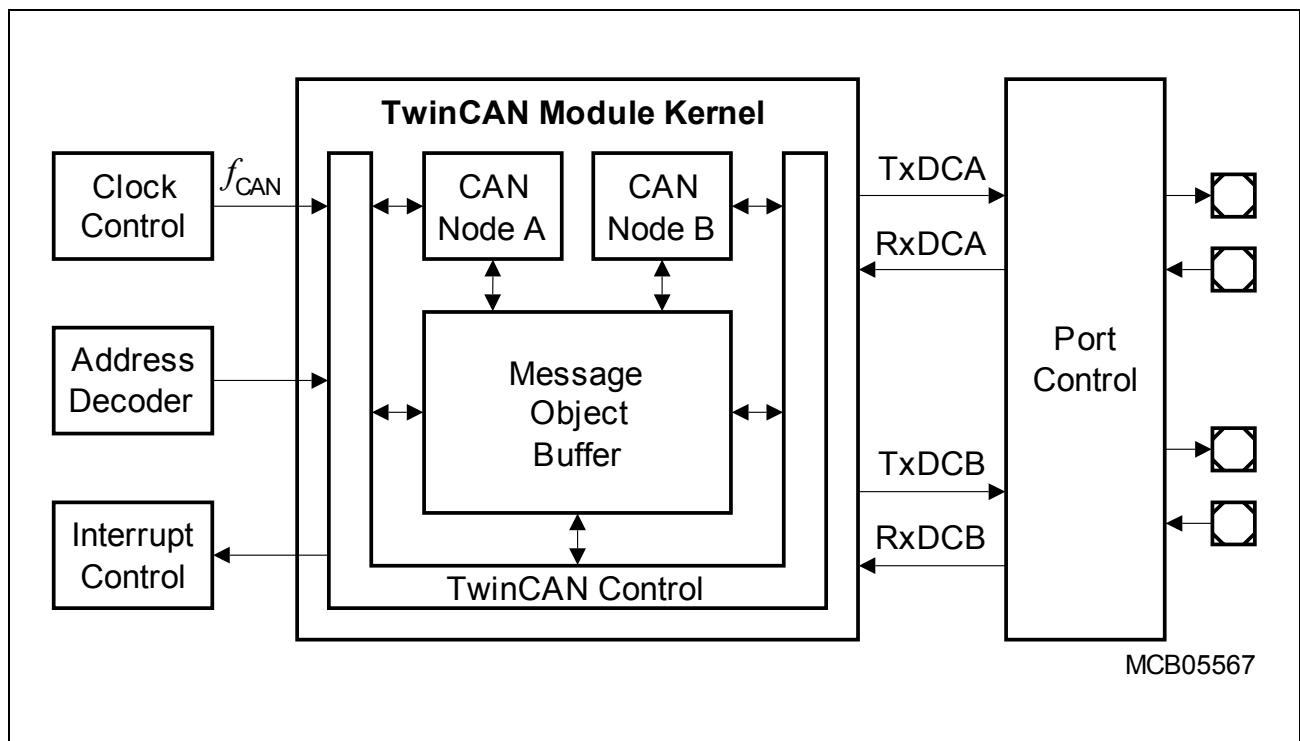


Figure 10 TwinCAN Module Block Diagram

3.14 Watchdog Timer

The Watchdog Timer represents one of the fail-safe mechanisms which have been implemented to prevent the controller from malfunctioning for longer periods of time.

The Watchdog Timer is always enabled after a reset of the chip, and can be disabled until the EINIT instruction has been executed (compatible mode), or it can be disabled and enabled at any time by executing instructions DISWDT and ENWDT (enhanced mode). Thus, the chip's start-up procedure is always monitored. The software has to be designed to restart the Watchdog Timer before it overflows. If, due to hardware or software related failures, the software fails to do so, the Watchdog Timer overflows and generates an internal hardware reset.

The Watchdog Timer is a 16-bit timer, clocked with the system clock divided by 2/4/128/256. The high byte of the Watchdog Timer register can be set to a prespecified reload value (stored in WDTREL) in order to allow further variation of the monitored time interval. Each time it is serviced by the application software, the high byte of the Watchdog Timer is reloaded and the low byte is cleared. Thus, time intervals between 13 μ s and 419 ms can be monitored (@ 40 MHz).

The default Watchdog Timer interval after reset is 3.28 ms (@ 40 MHz).

3.16 Parallel Ports

The XC164CM provides up to 47 I/O lines which are organized into three input/output ports and one input port. All port lines are bit-addressable, and all input/output lines are individually (bit-wise) programmable as inputs or outputs via direction registers. The I/O ports are true bidirectional ports which are switched to high impedance state when configured as inputs. The output drivers of some I/O ports can be configured (pin by pin) for push/pull operation or open-drain operation via control registers. During the internal reset, all port pins are configured as inputs.

The edge characteristics (shape) and driver characteristics (output current) of the port drivers can be selected via registers POCONx.

The input threshold of some ports is selectable (TTL or CMOS like), where the special CMOS like input threshold reduces noise sensitivity due to the input hysteresis. The input threshold may be selected individually for each byte of the respective ports.

All port lines have programmable alternate input or output functions associated with them. All port lines that are not used for these alternate functions may be used as general purpose IO lines.

Table 7 Summary of the XC164CM's Parallel Ports

Port	Control	Alternate Functions
PORT1	Pad drivers	Capture inputs or compare outputs, Serial interface lines
Port 3	Pad drivers, Open drain, Input threshold	Timer control signals, serial interface lines, System clock output CLKOUT (or FOUT)
Port 5	–	Analog input channels to the A/D converter, Timer control signals
Port 9	Pad drivers, Open drain, Input threshold	Capture inputs or compare outputs
		CAN interface lines ¹⁾

1) Can be assigned by software.

3.18 Instruction Set Summary

Table 8 lists the instructions of the XC164CM in a condensed way.

The various addressing modes that can be used with a specific instruction, the operation of the instructions, parameters for conditional execution of instructions, and the opcodes for each instruction can be found in the “**Instruction Set Manual**”.

This document also provides a detailed description of each instruction.

Table 8 Instruction Set Summary

Mnemonic	Description	Bytes
ADD(B)	Add word (byte) operands	2 / 4
ADDC(B)	Add word (byte) operands with Carry	2 / 4
SUB(B)	Subtract word (byte) operands	2 / 4
SUBC(B)	Subtract word (byte) operands with Carry	2 / 4
MUL(U)	(Un)Signed multiply direct GPR by direct GPR (16- × 16-bit)	2
DIV(U)	(Un)Signed divide register MDL by direct GPR (16-/16-bit)	2
DIVL(U)	(Un)Signed long divide reg. MD by direct GPR (32-/16-bit)	2
CPL(B)	Complement direct word (byte) GPR	2
NEG(B)	Negate direct word (byte) GPR	2
AND(B)	Bitwise AND, (word/byte operands)	2 / 4
OR(B)	Bitwise OR, (word/byte operands)	2 / 4
XOR(B)	Bitwise exclusive OR, (word/byte operands)	2 / 4
BCLR/BSET	Clear/Set direct bit	2
BMOV(N)	Move (negated) direct bit to direct bit	4
BAND/BOR/BXOR	AND/OR/XOR direct bit with direct bit	4
BCMP	Compare direct bit to direct bit	4
BFLDH/BFLDL	Bitwise modify masked high/low byte of bit-addressable direct word memory with immediate data	4
CMP(B)	Compare word (byte) operands	2 / 4
CMPD1/2	Compare word data to GPR and decrement GPR by 1/2	2 / 4
CMPI1/2	Compare word data to GPR and increment GPR by 1/2	2 / 4
PRIOR	Determine number of shift cycles to normalize direct word GPR and store result in direct word GPR	2
SHL/SHR	Shift left/right direct word GPR	2

Functional Description
Table 8 Instruction Set Summary (cont'd)

Mnemonic	Description	Bytes
NOP	Null operation	2
CoMUL/CoMAC	Multiply (and accumulate)	4
CoADD/CoSUB	Add/Subtract	4
Co(A)SHR	(Arithmetic) Shift right	4
CoSHL	Shift left	4
CoLOAD/STORE	Load accumulator/Store MAC register	4
CoCMP	Compare	4
CoMAX/MIN	Maximum/Minimum	4
CoABS/CoRND	Absolute value/Round accumulator	4
CoMOV	Data move	4
CoNEG/NOP	Negate accumulator/Null operation	4

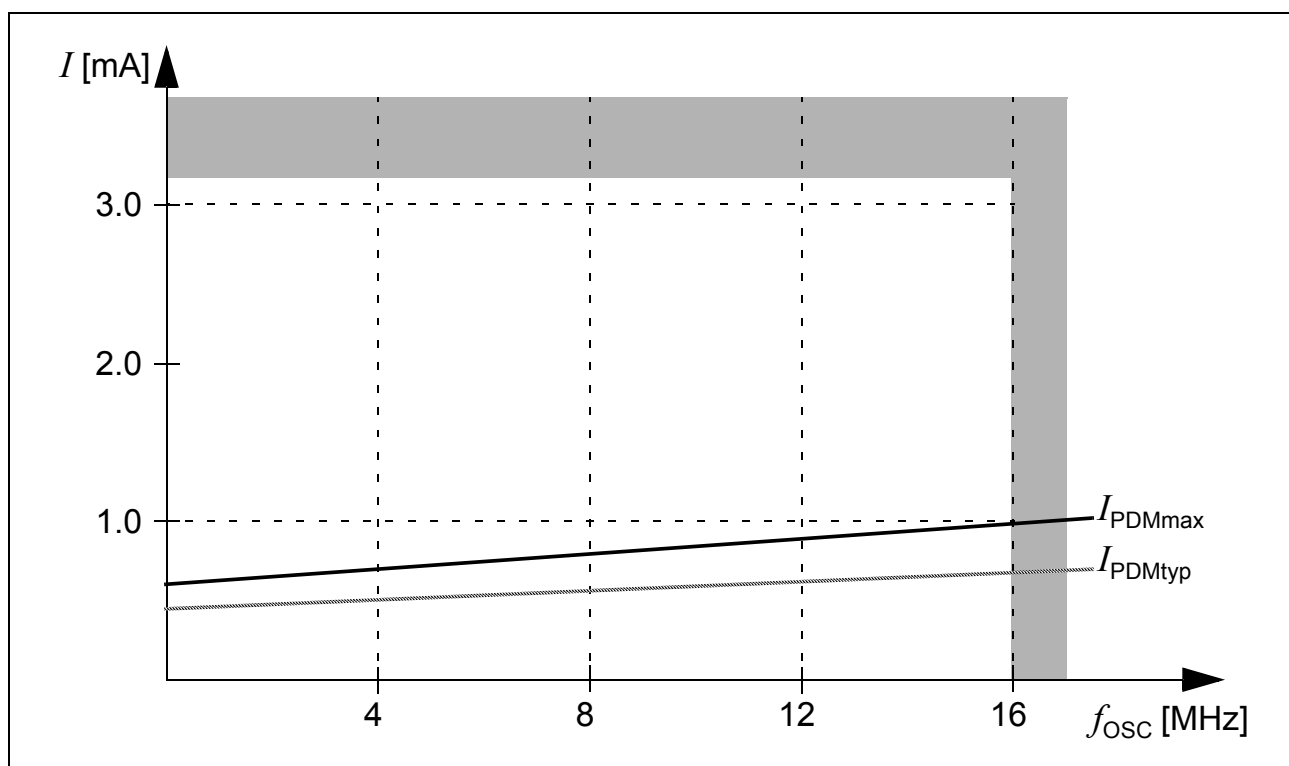


Figure 12 Sleep and Power Down Supply Current due to RTC and Oscillator Running, as a Function of Oscillator Frequency

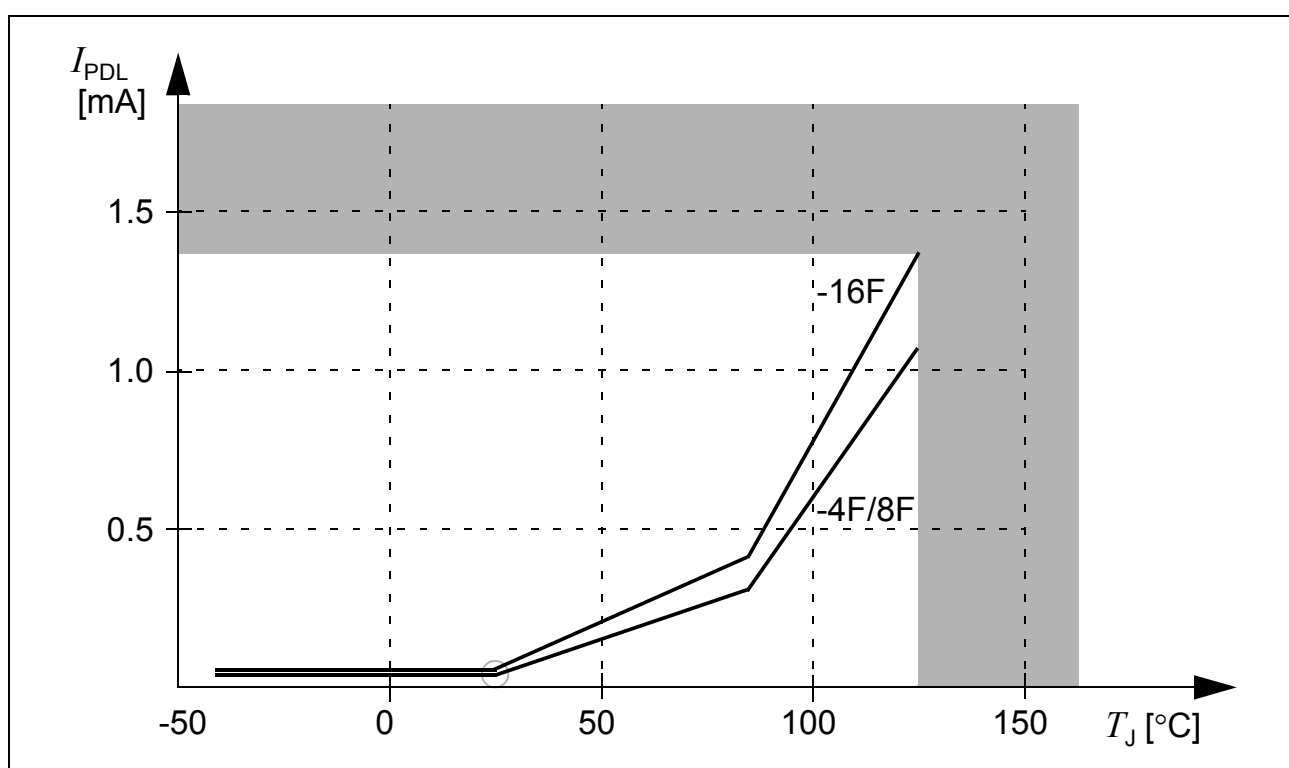


Figure 13 Sleep and Power Down Leakage Supply Current as a Function of Temperature

Electrical Parameters

Sample time and conversion time of the XC164CM's A/D Converter are programmable. In compatibility mode, the above timing can be calculated using [Table 15](#). The limit values for f_{BC} must not be exceeded when selecting ADCTC.

Table 15 A/D Converter Computation Table¹⁾

ADCON.15 14 (ADCTC)	A/D Converter Basic Clock f_{BC}	ADCON.13 12 (ADSTC)	Sample Time t_s
00	$f_{SYS} / 4$	00	$t_{BC} \times 8$
01	$f_{SYS} / 2$	01	$t_{BC} \times 16$
10	$f_{SYS} / 16$	10	$t_{BC} \times 32$
11	$f_{SYS} / 8$	11	$t_{BC} \times 64$

1) These selections are available in compatibility mode. An improved mechanism to control the ADC input clock can be selected.

Converter Timing Example:

Assumptions: $f_{SYS} = 40$ MHz (i.e. $t_{SYS} = 25$ ns), ADCTC = '01', ADSTC = '00'

Basic clock $f_{BC} = f_{SYS} / 2 = 20$ MHz, i.e. $t_{BC} = 50$ ns

Sample time $t_s = t_{BC} \times 8 = 400$ ns

Conversion 10-bit:

With post-calibr. $t_{C10P} = 52 \times t_{BC} + t_s + 6 \times t_{SYS} = (2600 + 400 + 150)$ ns = 3.15 μ s

Post-calibr. off $t_{C10} = 40 \times t_{BC} + t_s + 6 \times t_{SYS} = (2000 + 400 + 150)$ ns = 2.55 μ s

Conversion 8-bit:

With post-calibr. $t_{C8P} = 44 \times t_{BC} + t_s + 6 \times t_{SYS} = (2200 + 400 + 150)$ ns = 2.75 μ s

Post-calibr. off $t_{C8} = 32 \times t_{BC} + t_s + 6 \times t_{SYS} = (1600 + 400 + 150)$ ns = 2.15 μ s

Electrical Parameters

The used mechanism to generate the master clock is selected by register PLLCON.

CPU and EBC are clocked with the CPU clock signal f_{CPU} . The CPU clock can have the same frequency as the master clock ($f_{\text{CPU}} = f_{\text{MC}}$) or can be the master clock divided by two: $f_{\text{CPU}} = f_{\text{MC}} / 2$. This factor is selected by bit CPSYS in register SYSCON1.

The specification of the external timing (AC Characteristics) depends on the period of the CPU clock, called "TCP".

The other peripherals are supplied with the system clock signal f_{SYS} which has the same frequency as the CPU clock signal f_{CPU} .

Bypass Operation

When bypass operation is configured (PLLCTRL = 0x_B) the master clock is derived from the internal oscillator (input clock signal XTAL1) through the input- and output-prescalers:

$$f_{\text{MC}} = f_{\text{OSC}} / ((\text{PLLIDIV} + 1) \times (\text{PLLODIV} + 1)).$$

If both divider factors are selected as '1' (PLLIDIV = PLLODIV = '0') the frequency of f_{MC} directly follows the frequency of f_{OSC} so the high and low time of f_{MC} is defined by the duty cycle of the input clock f_{OSC} .

The lowest master clock frequency is achieved by selecting the maximum values for both divider factors:

$$f_{\text{MC}} = f_{\text{OSC}} / ((3 + 1) \times (14 + 1)) = f_{\text{OSC}} / 60.$$

Phase Locked Loop (PLL)

When PLL operation is configured (PLLCTRL = 11_B) the on-chip phase locked loop is enabled and provides the master clock. The PLL multiplies the input frequency by the factor **F** ($f_{\text{MC}} = f_{\text{OSC}} \times \mathbf{F}$) which results from the input divider, the multiplication factor, and the output divider ($\mathbf{F} = \text{PLLMUL} + 1 / (\text{PLLIDIV} + 1 \times \text{PLLODIV} + 1)$). The PLL circuit synchronizes the master clock to the input clock. This synchronization is done smoothly, i.e. the master clock frequency does not change abruptly.

Due to this adaptation to the input clock the frequency of f_{MC} is constantly adjusted so it is locked to f_{OSC} . The slight variation causes a jitter of f_{MC} which also affects the duration of individual TCMs.

The timing listed in the AC Characteristics refers to TCPs. Because f_{CPU} is derived from f_{MC} , the timing must be calculated using the minimum TCP possible under the respective circumstances.

The actual minimum value for TCP depends on the jitter of the PLL. As the PLL is constantly adjusting its output frequency so it corresponds to the applied input frequency (crystal or oscillator) the relative deviation for periods of more than one TCP is lower than for one single TCP (see formula and [Figure 16](#)).