Never stop thinking

8-Bit

XC864

8-Bit Single-Chip Microcontroller

Data Sheet

V1.1 2009-03

Microcontrollers

Edition 2009-03

Published by
Infineon Technologies AG
81726 Munich, Germany

Legal Disclaimer

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation, warranties of non-infringement of intellectual property rights of any third party.

Information

For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements, components may contain dangerous substances. For information on the types in question, please contact the nearest Infineon Technologies Office.

Infineon Technologies components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Never stop thinking

8-Bit

XC864

8-Bit Single-Chip Microcontroller

Data Sheet

V1.1 2009-03

Microcontrollers

XC864 Data Sheet
Revision History: 2009-03 V 1.1

Previous Version: V1.0

Page	Subjects (major changes since last revision)
Changes from V1.0 2008-08 to V1.1 2009-03	
3	Modified the paragraph to remove the Automotive quality profile

We Listen to Your Comments

Any information within this document that you feel is wrong, unclear or missing at all? Your feedback will help us to continuously improve the quality of this document. Please send your proposal (including a reference to this document) to:

mcdocu.comments@infineon.com

1 Summary of Features

- High-performance XC800 Core
  - compatible with standard 8051 processor
  - two clocks per machine cycle architecture (for memory access without wait state)
  - two data pointers
- On-chip memory
  - 8 Kbytes of Boot ROM
  - 256 bytes of RAM
  - 512 bytes of XRAM
  - 4 Kbytes of Flash for code (and data) (includes memory protection strategy)
- I/O port supply at 3.3 V/5.0 V and core logic supply at 2.5 V (generated by embedded voltage regulator)

(further features are on next page)

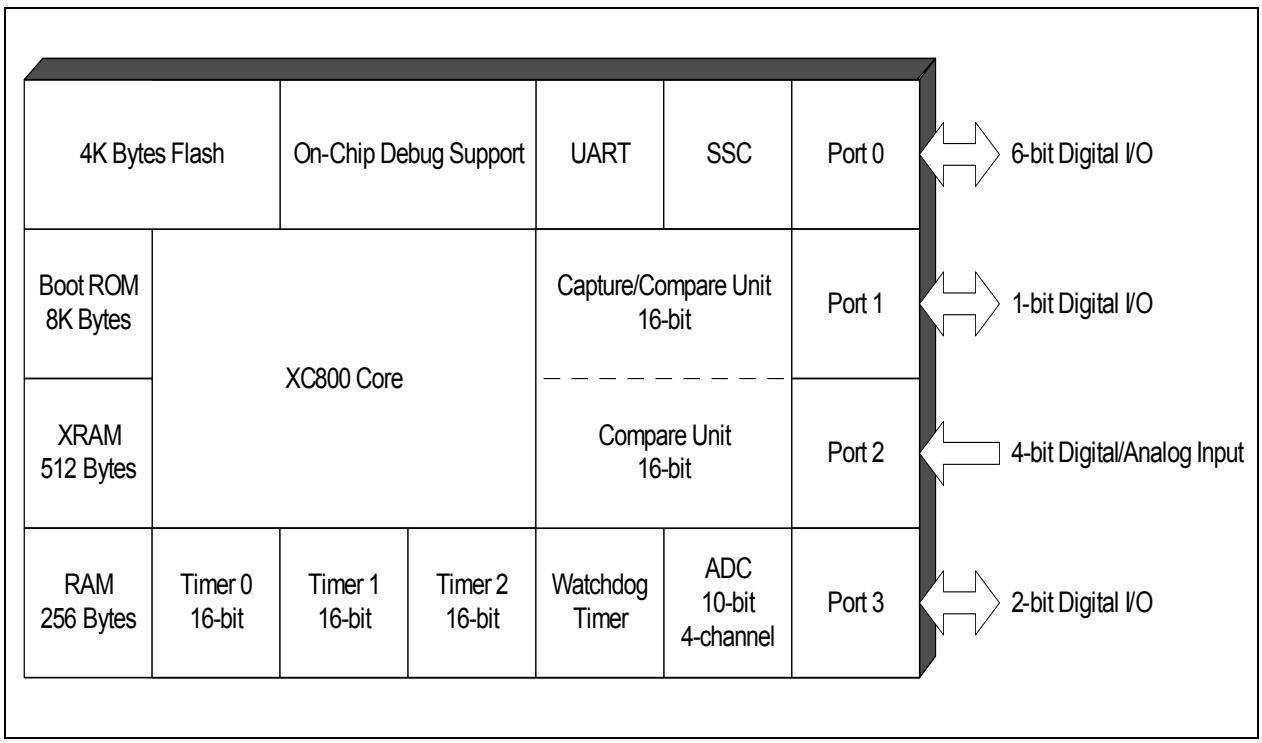


Figure 1 XC864 Functional Units

Features (continued):

- Reset generation
  - Power-On reset
  - Hardware reset
  - Brownout reset for core logic supply
  - Watchdog timer reset
  - Power-Down Wake-up reset
- On-chip OSC and PLL for clock generation
  - PLL loss-of-lock detection
- Power saving modes
  - slow-down mode
  - idle mode
  - power-down mode with wake-up capability via RXD or EXINT0
  - clock gating control to each peripheral
- Programmable 16-bit Watchdog Timer (WDT)
- Four ports
  - 9 pins as digital I/O
  - 4 pins as digital/analog input
- 4-channel, 8-bit ADC
- Three 16-bit timers
  - Timer 0 and Timer 1 (T0 and T1)
  - Timer 2
- Capture/compare unit for PWM signal generation (CCU6)
- Full-duplex serial interface (UART)
- Synchronous serial channel (SSC)
- On-chip debug support
  - 1 Kbyte of monitor ROM (part of the 8-Kbyte Boot ROM)
  - 64 bytes of monitor RAM
- PG-TSSOP-20 pin package
- Ambient temperature range T_A:
  - SAF (-40 to 85 °C)
  - SAK (-40 to 125 °C)

XC864 Variant Devices

The XC864 product family features devices with different power supply range and temperature, offering cost-effective solution for different application requirements. The package type available is TSSOP-20.

Table 1-1 summarizes the list of XC864 devices.

Table 1-1 Device Profile

Sales Type	Device Type	Program Memory (Kbytes)	Power Supply (V)	Temp-erature Profile (°C)	Quality Profile
SAK-XC864L-1FRI 5V	Flash	4	5.0	-40 to 125	Industrial
SAK-XC864L-1FRI 3V3	Flash	4	3.3	-40 to 125	Industrial
SAF-XC864L-1FRI 5V	Flash	4	5.0	-40 to 85	Industrial
SAF-XC864L-1FRI 3V3	Flash	4	3.3	-40 to 85	Industrial

Ordering Information

The ordering code for Infineon Technologies microcontrollers provides an exact reference to the required product. This ordering code identifies:

- The derivative itself, i.e. its function set, the temperature range, and the supply voltage
- the package and the type of delivery

For the available ordering codes for the XC864, please refer to your responsible sales representative or your local distributor.

As this document refers to all the derivatives, some descriptions may not apply to a specific product. For simplicity all versions are referred to by the term XC864 throughout this document.

2 General Device Information

2.1 Block Diagram

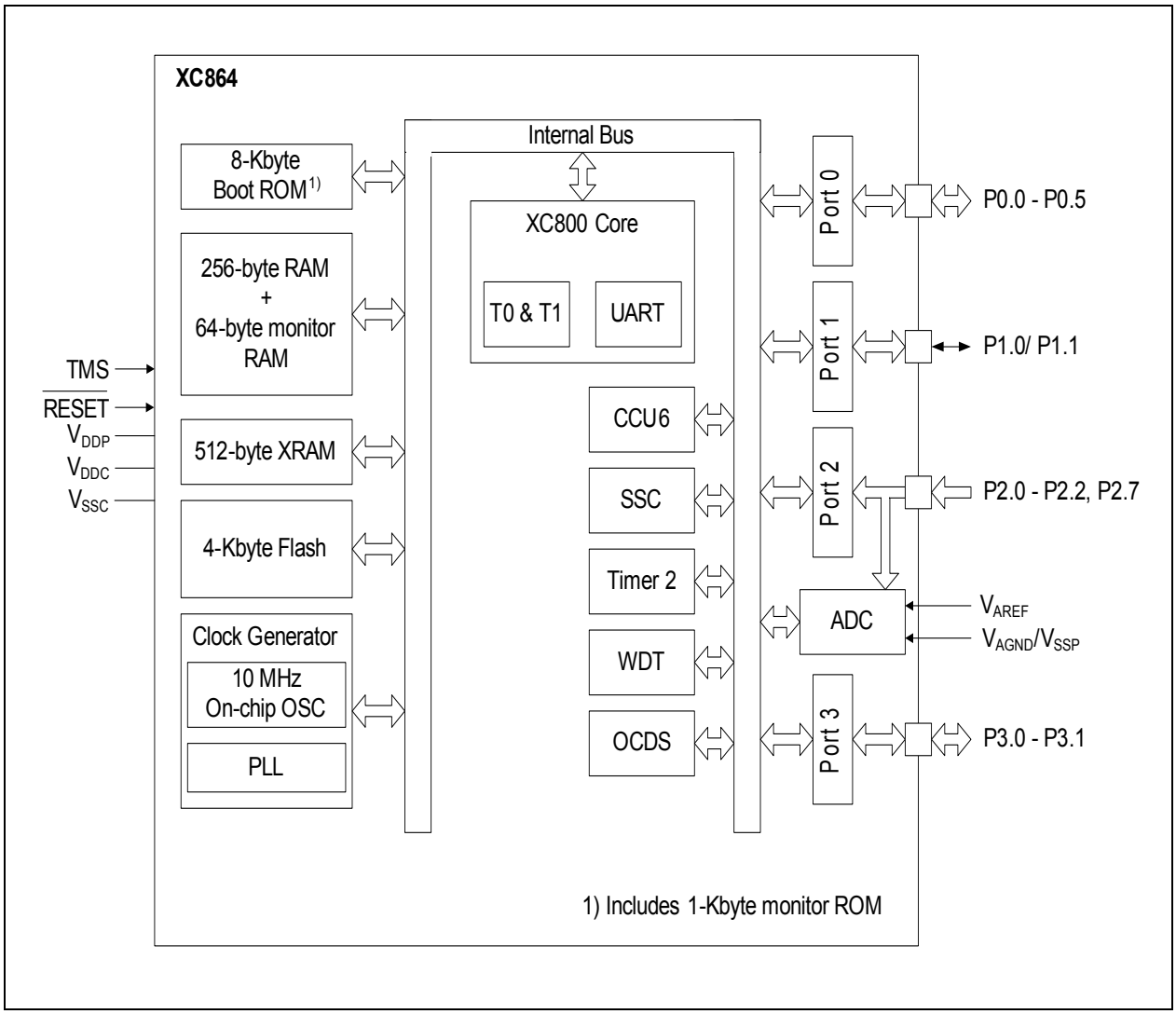


Figure 2 XC864 Block Diagram

2.2 Logic Symbol

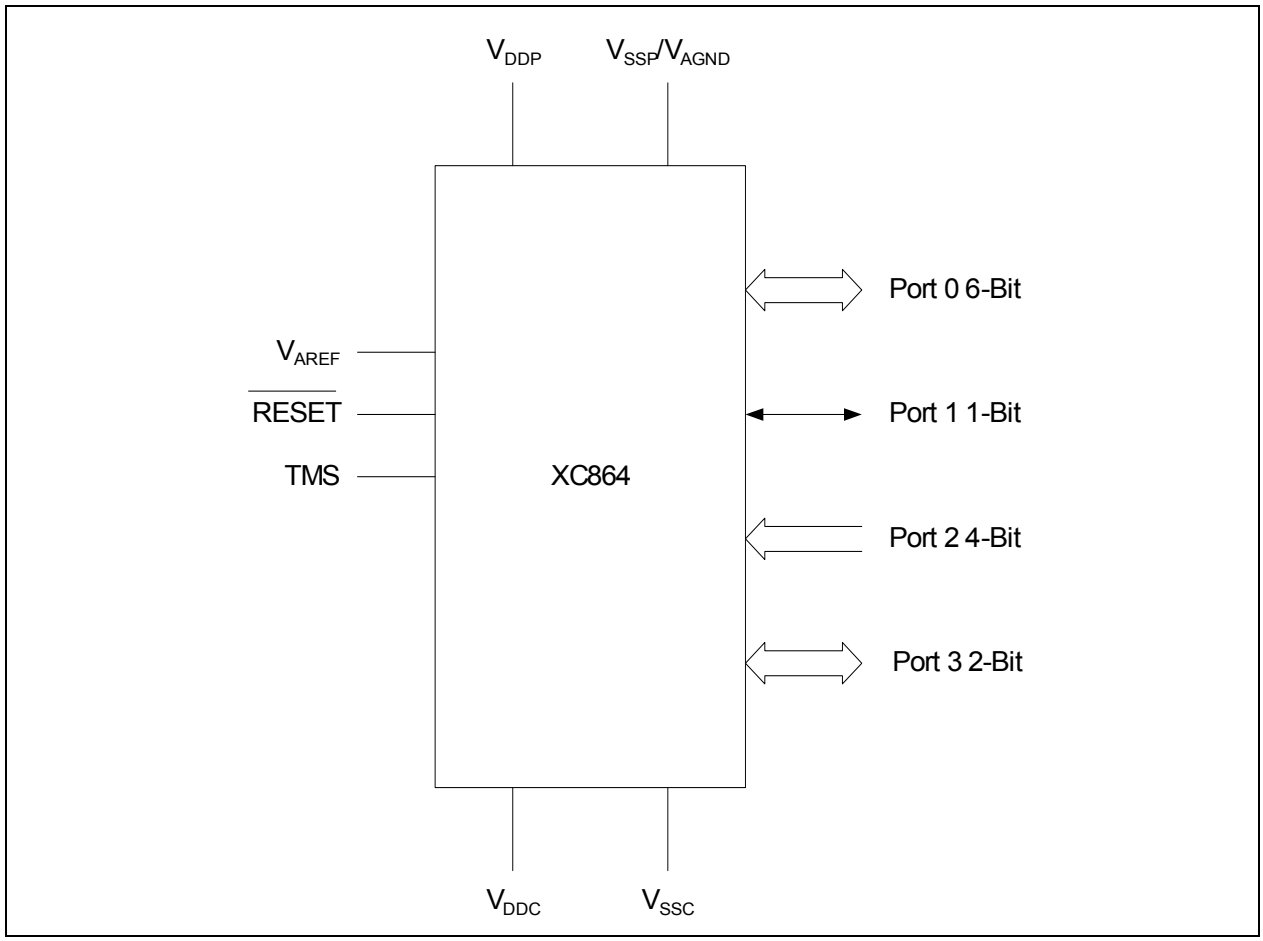


Figure 3 XC864 Logic Symbol

2.3 Pin Configuration

The pin configuration of the XC864, which is based on the PG-TSSOP-20 package, is shown in **Figure 4**. Every package pin is bonded to an input port pin or a bidirectional port pin except Pin 15. It is bonded to 2 bidirectional port pins namely, P1.0 and P1.1. Configurations of both port pins to output direction concurrently must be avoided to prevent permanent damage to the chip[1)].

In addition, open drain output mode with pull-up device enabled is recommended for P1.1 as TXD function and input mode for P1.0 as RXD function in single wire UART communication.

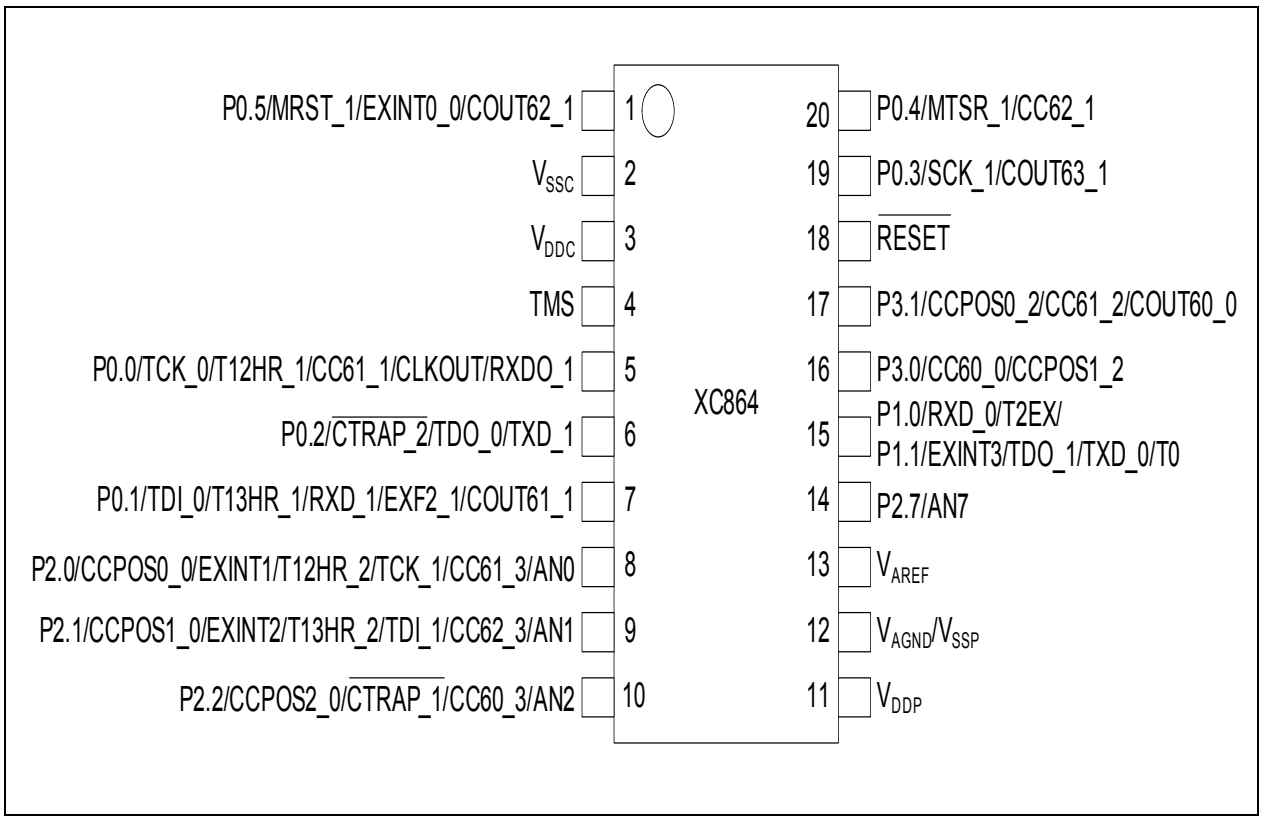


Figure 4 XC864 Pin Configuration, PG-TSSOP-20 Package (top view)

1) Protection against improper usage of P1.0 and P1.1 is not available in XC864.

2.4 Pin Definitions and Functions

Table 1 Pin Definitions and Functions

Symbol	Pin Number	Type	Reset State	Function	
P0		I/O		**Port 0** Port 0 is an 8-bit bidirectional general purpose I/O port. It can be used as alternate functions for the JTAG, CCU6, UART, Timer 2 and SSC.	
P0.0	5		Hi-Z	TCK_0 T12HR_1 CC61_1 CLKOUT_0 RXDO_1	JTAG Clock Input CCU6 Timer 12 Hardware Run Input Input/Output of Capture/ Compare channel 1 Clock Output UART Transmit Data Output
P0.1	7		Hi-Z	TDI_0 T13HR_1 RXD_1 COUT61_1 EXF2_1	JTAG Serial Data Input CCU6 Timer 13 Hardware Run Input UART Receive Data Input Output of Capture/Compare channel 1 Timer 2 External Flag Output
P0.2	6		PU	$\overline{\text{CTRAP}}$_2 TDO_0 TXD_1	CCU6 Trap Input JTAG Serial Data Output UART Transmit Data Output/ Clock Output
P0.3	19		Hi-Z	SCK_1 COUT63_1	SSC Clock Input/Output Output of Capture/Compare channel 3
P0.4	20		Hi-Z	MTSR_1 CC62_1	SSC Master Transmit Output/ Slave Receive Input Input/Output of Capture/ Compare channel 2
P0.5	1		Hi-Z	MRST_1 EXINT0_0 COUT62_1	SSC Master Receive Input/Slave Transmit Output External Interrupt Input 0 Output of Capture/Compare channel 2

Table 1 Pin Definitions and Functions (cont'd)

Symbol	Pin Number	Type	Reset State	Function
P1		I/O		**Port 1** Port 1 is an 8-bit bidirectional general purpose I/O port. It can be used as alternate functions for the JTAG, CCU6, UART, Timer 0, Timer 2 and SSC.
P1.0/ P1.1	15		PU	RXD_0 UART Receive Data Input T2EX Timer 2 External Trigger Input EXINT3 External Interrupt Input 3 T0 Timer 0 Input TDO_1 JTAG Serial Data Output TXD_0 UART Transmit Data Output/ Clock Output *Note: Pin 15 is bonded to both P1.0 and P1.1 port pins. See **Section 2.3** on the types of port pin configuration to be avoided to prevent permanent damage.*

Table 1 Pin Definitions and Functions (cont'd)

Symbol	Pin Number	Type	Reset State	Function
P2		I		**Port 2** Port 2 is an 8-bit general purpose input-only port. It can be used as alternate functions for the digital inputs of the JTAG and CCU6. It is also used as the analog inputs for the ADC.
P2.0	8		Hi-Z	CCPOS0_0 CCU6 Hall Input 0 EXINT1_0 External Interrupt Input 1 T12HR_2 CCU6 Timer 12 Hardware Run Input TCK_1 JTAG Clock Input CC61_3 Input of Capture/Compare channel 1 AN0 Analog Input 0
P2.1	9		Hi-Z	CCPOS1_0 CCU6 Hall Input 1 EXINT2_0 External Interrupt Input 2 T13HR_2 CCU6 Timer 13 Hardware Run Input TDI_1 JTAG Serial Data Input CC62_3 Input of Capture/Compare channel 2 AN1 Analog Input 1
P2.2	10		Hi-Z	CCPOS2_0 CCU6 Hall Input 2 CTRAP_1 CCU6 Trap Input CC60_3 Input of Capture/Compare channel 0 AN2 Analog Input 2
P2.7	14		Hi-Z	AN7 Analog Input 7

Table 1 Pin Definitions and Functions (cont'd)

Symbol	Pin Number	Type	Reset State	Function
P3		I/O		**Port 3** Port 3 is an 8-bit bidirectional general purpose I/O port. It can be used as alternate functions for CCU6.
P3.0	16		Hi-Z	CCPOS1_2 CCU6 Hall Input 1 CC60_0 Input/Output of Capture/ Compare channel 0
P3.1	17		Hi-Z	CCPOS0_2 CCU6 Hall Input 0 CC61_2 Input/Output of Capture/ Compare channel 1 COUT60_0 Output of Capture/Compare channel 0
V_{DDP}	11	–	–	**I/O Port Supply (3.3 or 5.0 V)** Also used by EVR and analog modules. All pins must be connected.
V_{DDC}	3	–	–	**Core Supply Monitor (2.5 V)**
V_{SSC}	2	–	–	**Core Supply Ground**
V_{AREF}	13	–	–	**ADC Reference Voltage**
V_{AGND}/ V_{SSP}	12	–	–	**ADC Reference Ground/ I/O Ground** All pins must be connected.
TMS	4	I	PD	**Test Mode Select**
RESET	18	I	PU	**Reset Input**

3 Functional Description

3.1 Processor Architecture

The XC864 is based on a high-performance 8-bit Central Processing Unit (CPU) that is compatible with the standard 8051 processor. While the standard 8051 processor is designed around a 12-clock machine cycle, the XC864 CPU uses a 2-clock machine cycle. This allows fast access to ROM or RAM memories without wait state. Access to the Flash memory, however, requires an additional wait state (one machine cycle). The instruction set consists of 45% one-byte, 41% two-byte and 14% three-byte instructions.

The XC864 CPU provides a range of debugging features, including basic stop/start, single-step execution, breakpoint support and read/write access to the data memory, program memory and SFRs.

Figure 5 shows the CPU functional blocks.

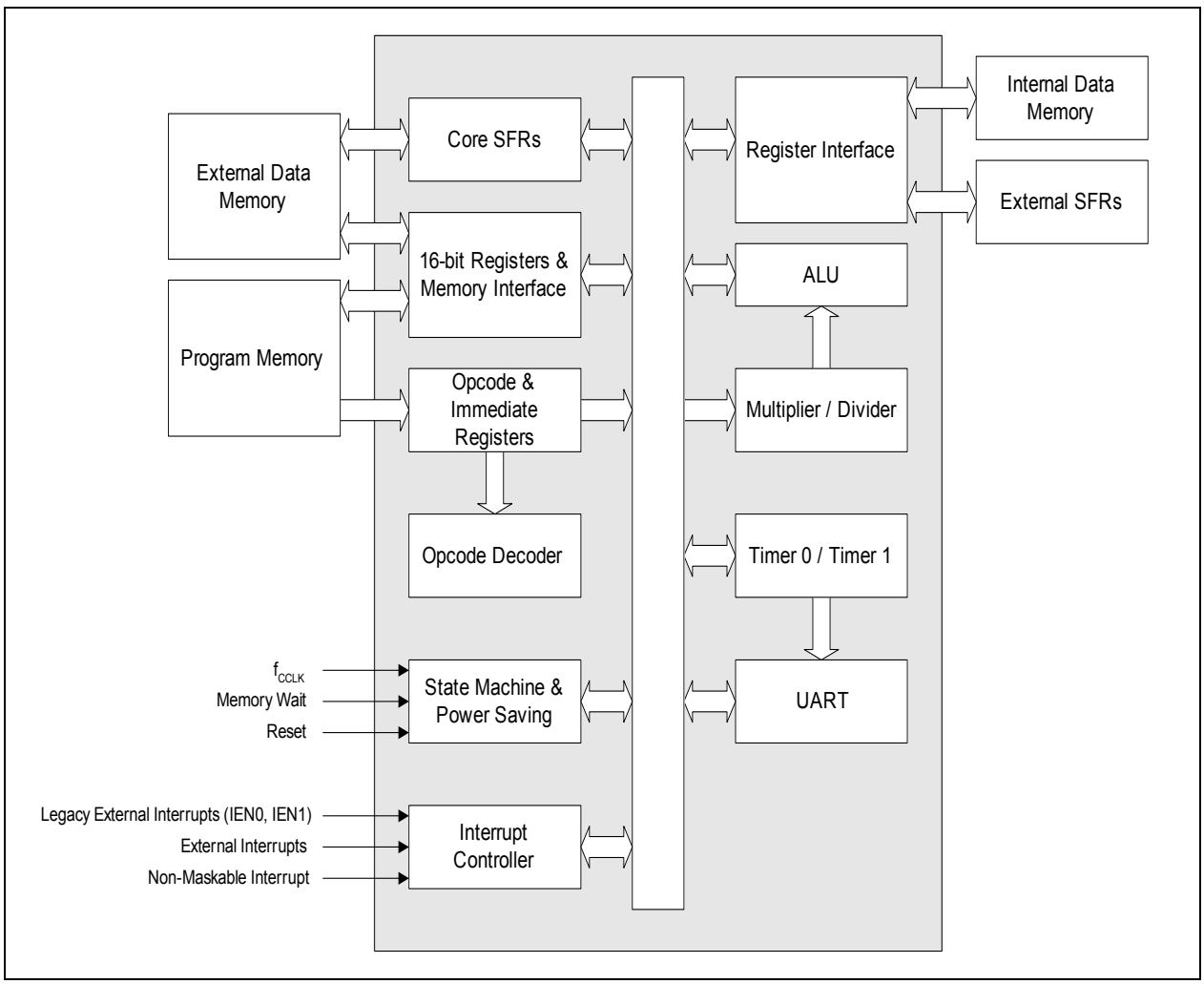


Figure 5 CPU Block Diagram

3.2 Memory Organization

The XC864 consists of four types of memory:

- 8 Kbytes of Boot ROM program memory
- 256 bytes of internal RAM data memory
- 512 bytes of XRAM memory
  (XRAM can be read/written as program memory or external data memory)
- 128 Special Function Register
- 4 Kbytes of Flash for code (and data)

Figure 6 illustrates the memory map of the address spaces of the XC864-1FR device.

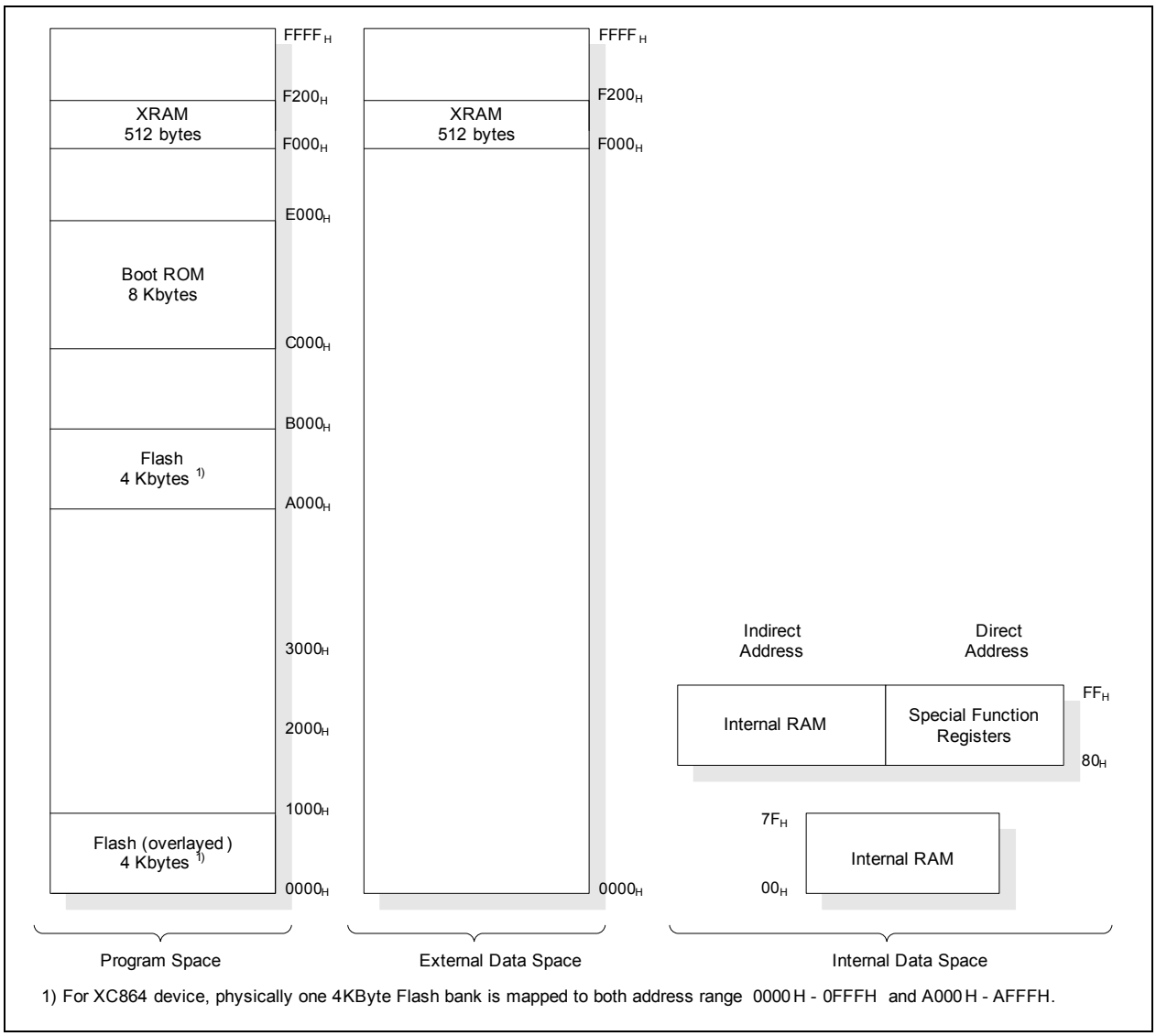


Figure 6 Memory Map of XC864

3.2.1 Memory Protection Strategy

The XC864 memory protection strategy includes:

- Read-out protection: The user is able to protect the contents in the Flash memory from being read
- Flash program and erase protection: The Flash memory in all devices can be enabled for program and erase protection
- Block external access and allow only boot in User Mode: Disable BSL and OCDS modes.

Flash memory protection modes provided are:

- Mode 0: Protect against accidental erase and block external access.
- Mode 1: Read, program and erase protection are enabled, and block external access.

Flash protection is enabled by installing the user password via BSL mode 6. The user setting of password for selection of each protection mode and the restrictions imposed are summarized in **Table 2**. Flash protection mode 1 is meaningful only if the Flash is used for code only. Otherwise if the Flash is used partially for code and partially for data, then only Flash protection mode 0 is meaningful.

Note: In XC864, the type of Flash protection scheme will affect the entering of BSL Mode once User Mode is entered.

Table 2 Flash Protection Modes

Mode	0	1
Selection	MSB of password = 0	MSB of password = 1
Flash contents can be read by	Read instructions in any program memory	Read instructions in Flash
Flash program	Possible	Not possible

Table 2 Flash Protection Modes (cont'd)

Flash erase	Possible, on condition that bit DFLASHEN in register MISC_CON is set to 1 prior to each erase operation	Not possible
Additional Protection	Block external access (can only start in User Mode)	Block external access (can only start in User Mode)
Subsequent entering of BSL mode with LSB of password is 1	Possible[1]; For detailed descriptions, see **"User Mode Entry 2" on Page 59**	Possible; For detailed descriptions, see **"User Mode Entry 2" on Page 59**
|Subsequent entering of BSL mode with LSB of password is 0	Not possible[1]	Not possible

[1] With MSB of password = 0, Flash content can be upgraded using a predefined routine in the user code via In-Application Programming(IAP). Programming via BSL mode is not needed. See **"User Mode Entry 3" on Page 60**.

BSL mode 6, which is used for enabling Flash protection, can also be used for disabling Flash protection. Here, the programmed password must be provided by the user. A password match triggers an automatic erase of the read-protected Flash contents (sector(s) to erase is defined by password, see **Table 3**), and the programmed password is erased. The Flash protection is then disabled upon the next reset.

Table 3 Password Definition

Password	To Enable Protection: Type of Hardware Protection[1)]	To Remove Protection: Sectors to Erase[2)] before Remove Hardware Protection
$1XXXXXXX_B$	Read/Program/Erase	All Sectors
$00001XXX_B$	Erase	Sector 0
$00010XXX_B$	Erase	Sector 0 and 1
$00011XXX_B$	Erase	Sector 0 to 2
$00100XXX_B$	Erase	Sector 0 to 3
$00101XXX_B$	Erase	Sector 0 to 4
$00110XXX_B$	Erase	Sector 0 to 5
$00111XXX_B$	Erase	Sector 0 to 6
$01000XXX_B$	Erase	Sector 0 to 7
$01001XXX_B$	Erase	Sector 0 to 8
$01010XXX_B$	Erase	All Sector
Others	Erase	None

1) On the whole Flash. This hardware protection is complimented by the ‘block external access’ feature (see **Table 2**).

2) Controlled automatically by BSL mode 6 routine in Boot ROM, based on the password previously installed by the user when enabling Flash protection.

Although no protection scheme can be considered infallible, the XC864 memory protection strategy provides a very high level of protection for a general purpose microcontroller.

3.2.2 Special Function Register

The Special Function Registers (SFRs) occupy direct internal data memory space in the range 80_H to FF_H. All registers, except the program counter, reside in the SFR area. The SFRs include pointers and registers that provide an interface between the CPU and the on-chip peripherals. As the 128-SFR range is less than the total number of registers required, address extension mechanisms are required to increase the number of addressable SFRs. The address extension mechanisms include:

- Mapping
- Paging

3.2.2.1 Address Extension by Mapping

Address extension is performed at the system level by mapping. The SFR area is extended into two portions: the standard (non-mapped) SFR area and the mapped SFR area. Each portion supports the same address range 80_H to FF_H, bringing the number of addressable SFRs to 256. The extended address range is not directly controlled by the CPU instruction itself, but is derived from bit RMAP in the system control register SYSCON0 at address $8F_H$. To access SFRs in the mapped area, bit RMAP in SFR SYSCON0 must be set. Alternatively, the SFRs in the standard area can be accessed by clearing bit RMAP. The SFR area can be selected as shown in **Figure 7**.

SYSCON0
System Control Register 0 **Reset Value: 04_H**

7	6	5	4	3	2	1	0
0					**1**	**0**	**RMAP**
r					rw	r	rw

Field	Bits	Type	Description
RMAP	0	rw	**Special Function Register Map Control** 0 The access to the standard SFR area is enabled. 1 The access to the mapped SFR area is enabled.
1	2	rw	**Reserved** Returns the last value if read; should be written with 1.
0	1,[7:3]	r	**Reserved** Returns 0 if read; should be written with 0.

Note: The RMAP bit must be cleared/set by ANL or ORL instructions.

As long as bit RMAP is set, the mapped SFR area can be accessed. This bit is not cleared automatically by hardware. Thus, before standard/mapped registers are accessed, bit RMAP must be cleared/set, respectively, by software.

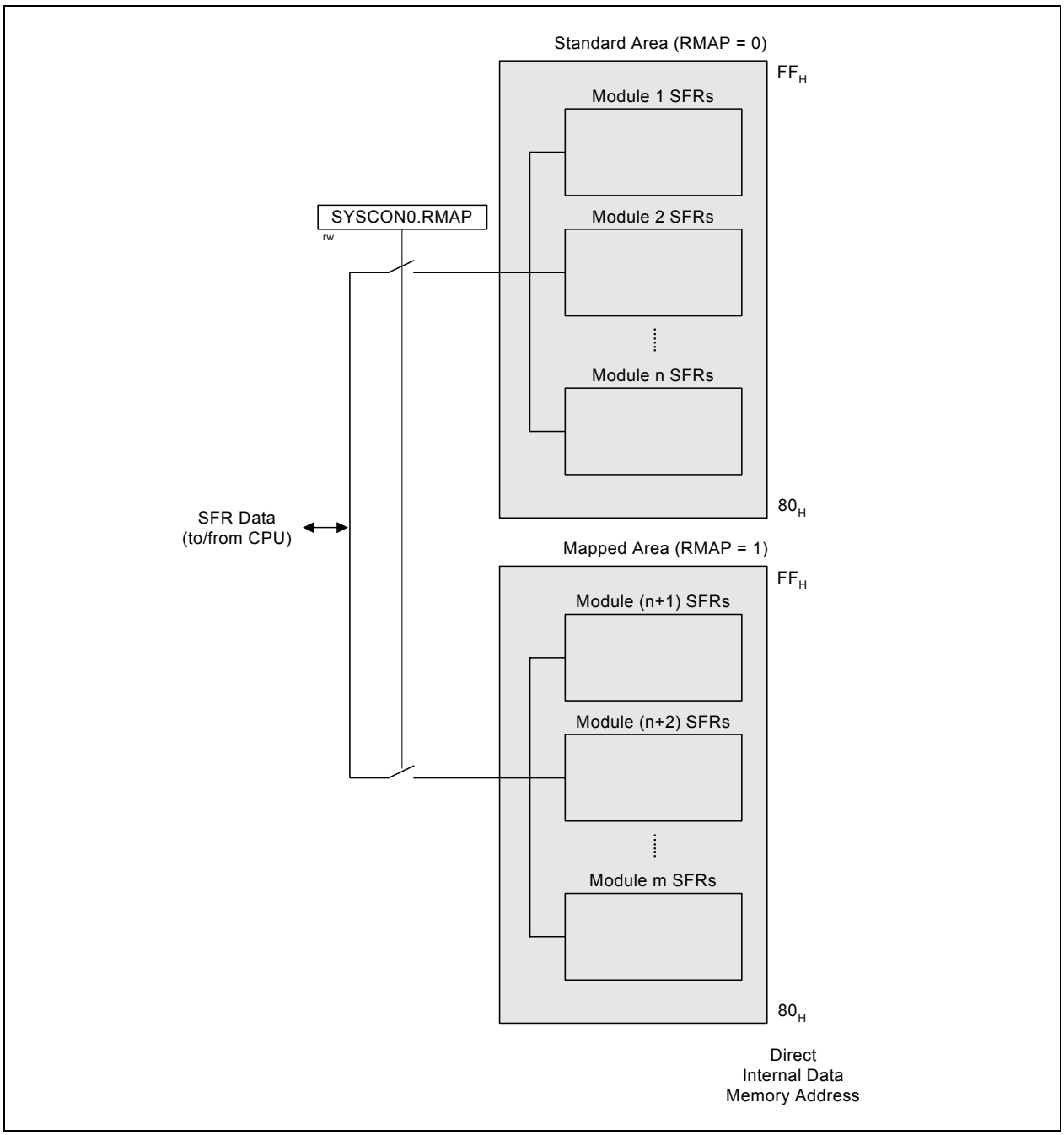


Figure 7 Address Extension by Mapping

3.2.2.2 Address Extension by Paging

Address extension is further performed at the module level by paging. With the address extension by mapping, the XC864 has a 256-SFR address range. However, this is still less than the total number of SFRs needed by the on-chip peripherals. To meet this requirement, some peripherals have a built-in local address extension mechanism for increasing the number of addressable SFRs. The extended address range is not directly controlled by the CPU instruction itself, but is derived from bit field PAGE in the module page register MOD_PAGE. Hence, the bit field PAGE must be programmed before accessing the SFR of the target module. Each module may contain a different number of pages and a different number of SFRs per page, depending on the specific requirement. Besides setting the correct RMAP bit value to select the SFR area, the user must also ensure that a valid PAGE is selected to target the desired SFR. A page inside the extended address range can be selected as shown in **Figure 8**.

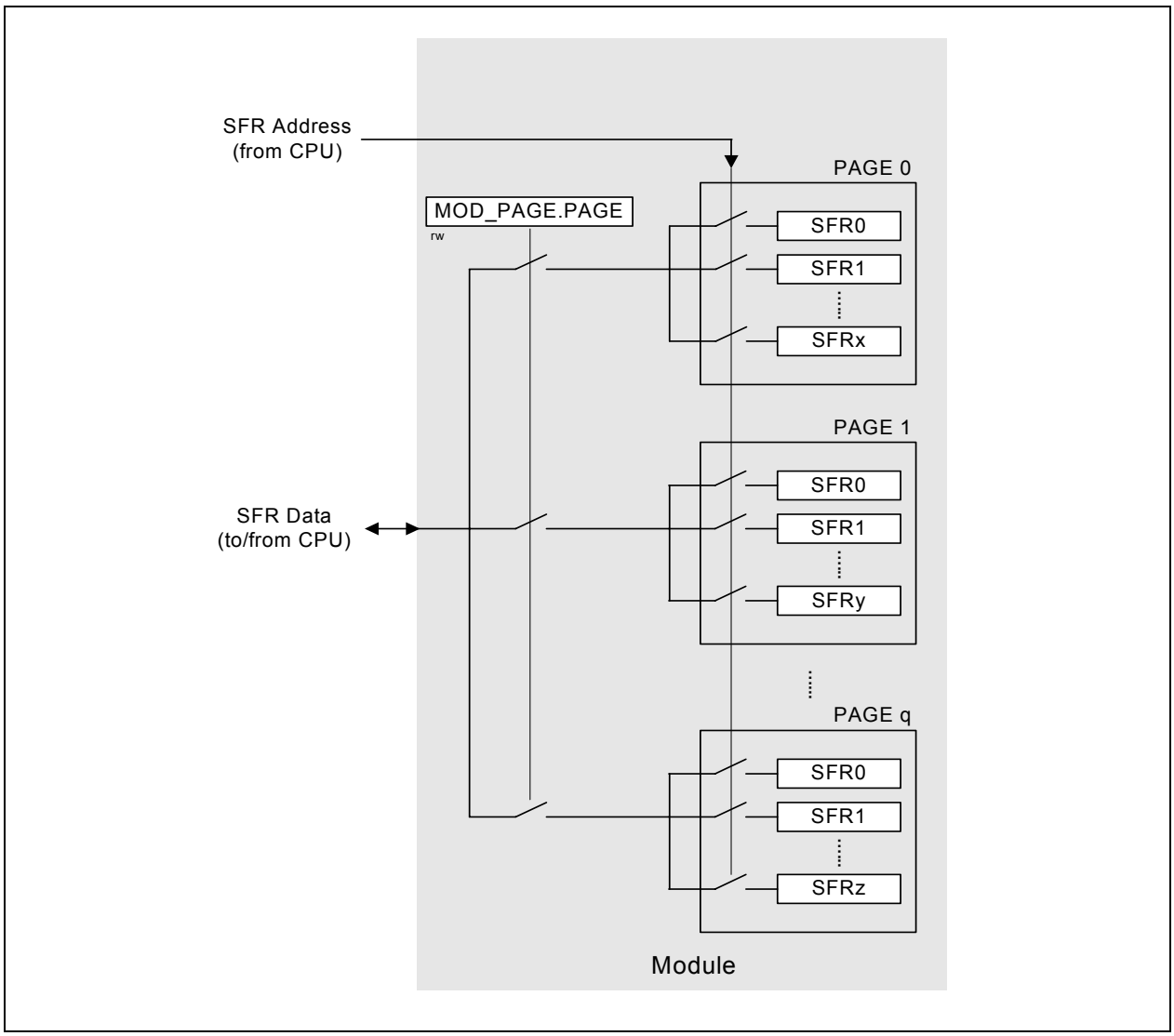


Figure 8 Address Extension by Paging

In order to access a register located in a page different from the actual one, the current page must be left. This is done by reprogramming the bit field PAGE in the page register. Only then can the desired access be performed.

If an interrupt routine is initiated between the page register access and the module register access, and the interrupt needs to access a register located in another page, the current page setting can be saved, the new one programmed and finally, the old page setting restored. This is possible with the storage fields STx (x = 0 - 3) for the save and restore action of the current page setting. By indicating which storage bit field should be used in parallel with the new page value, a single write operation can:

- Save the contents of PAGE in STx before overwriting with the new value
  (this is done in the beginning of the interrupt routine to save the current page setting and program the new page number); or
- Overwrite the contents of PAGE with the contents of STx, ignoring the value written to the bit positions of PAGE
  (this is done at the end of the interrupt routine to restore the previous page setting before the interrupt occurred)

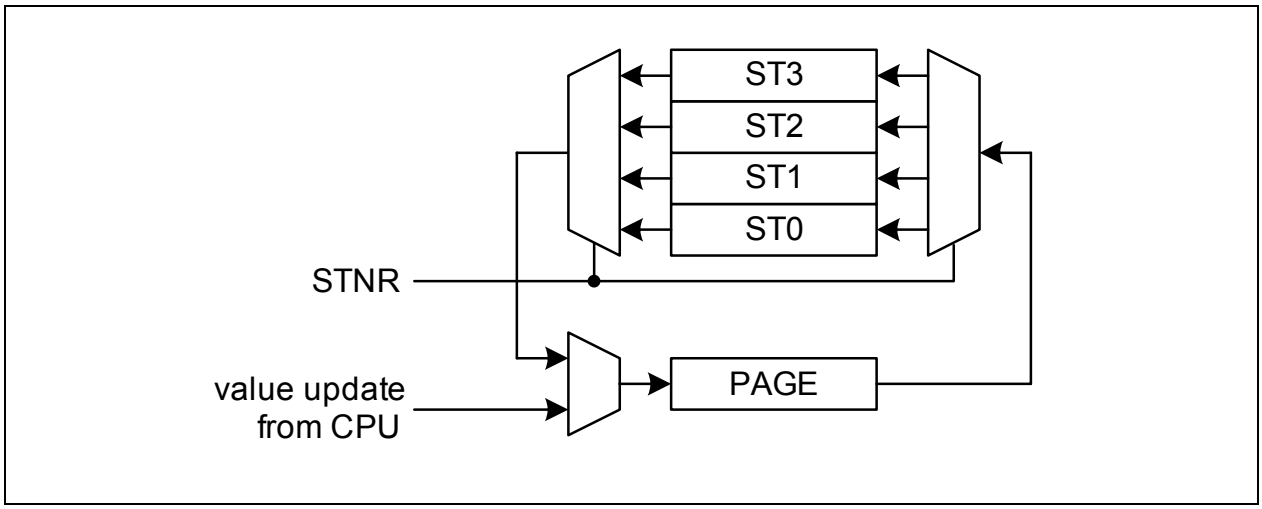


Figure 9 Storage Elements for Paging

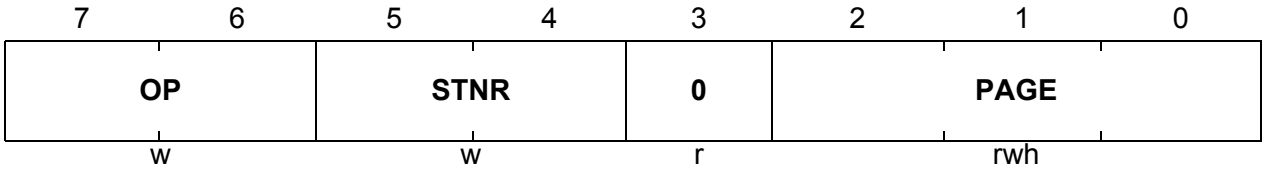
With this mechanism, a certain number of interrupt routines (or other routines) can perform page changes without reading and storing the previously used page information. The use of only write operations makes the system simpler and faster. Consequently, this mechanism significantly improves the performance of short interrupt routines.

The XC864 supports local address extension for:

- Parallel Ports
- Analog-to-Digital Converter (ADC)
- Capture/Compare Unit 6 (CCU6)
- System Control Registers

The page register has the following definition:

MOD_PAGE
Page Register for module MOD **Reset Value: 00_H**



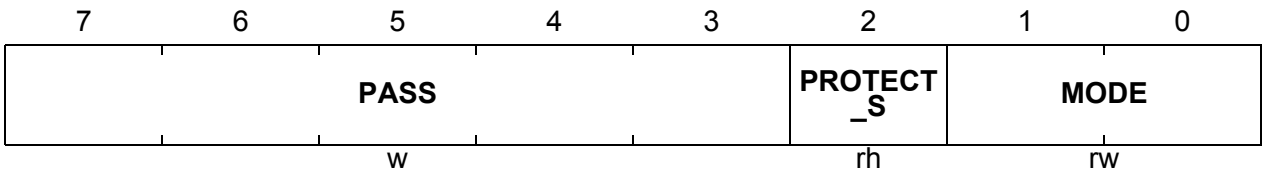
Field	Bits	Type	Description
PAGE	[2:0]	rwh	**Page Bits** When written, the value indicates the new page. When read, the value indicates the currently active page.
STNR	[5:4]	w	**Storage Number** This number indicates which storage bit field is the target of the operation defined by bit field OP. If OP = 10_B, the contents of PAGE are saved in STx before being overwritten with the new value. If OP = 11_B, the contents of PAGE are overwritten by the contents of STx. The value written to the bit positions of PAGE is ignored. 00 ST0 is selected. 01 ST1 is selected. 10 ST2 is selected. 11 ST3 is selected.
OP	[7:6]	w	**Operation** 0X Manual page mode. The value of STNR is ignored and PAGE is directly written. 10 New page programming with automatic page saving. The value written to the bit positions of PAGE is stored. In parallel, the previous contents of PAGE are saved in the storage bit field STx indicated by STNR. 11 Automatic restore page action. The value written to the bit positions PAGE is ignored and instead, PAGE is overwritten by the contents of the storage bit field STx indicated by STNR.

Field	Bits	Type	Description
0	3	r	**Reserved** Returns 0 if read; should be written with 0.

3.2.3 Bit Protection Scheme

The bit protection scheme prevents direct software writing of selected bits (i.e., protected bits) using the PASSWD register. When the bit field MODE is 11_B, writing 10011_B to the bit field PASS opens access to writing of all protected bits, and writing 10101_B to the bit field PASS closes access to writing of all protected bits. Note that access is opened for maximum 32 CCLKs if the “close access” password is not written. If “open access” password is written again before the end of 32 CCLK cycles, there will be a recount of 32 CCLK cycles. The protected bits include NDIV, WDTEN, PD, and SD.

PASSWD
Password Register **Reset Value: 07_H**



Field	Bits	Type	Description
MODE	[1:0]	rw	**Bit Protection Scheme Control bits** 00 Scheme Disabled 11 Scheme Enabled (default) Others: Scheme Enabled These two bits cannot be written directly. To change the value between 11_B and 00_B, the bit field PASS must be written with 11000_B; only then, will the MODE[1:0] be registered.
PROTECT_S	2	rh	**Bit Protection Signal Status bit** This bit shows the status of the protection. 0 Software is able to write to all protected bits. 1 Software is unable to write to any protected bits.
PASS	[7:3]	w	**Password bits** The Bit Protection Scheme only recognizes three patterns. 11000_B Enables writing of the bit field MODE. 10011_B Opens access to writing of all protected bits. 10101_B Closes access to writing of all protected bits.

3.2.4 XC864 Register Overview

The SFRs of the XC864 are organized into groups according to their functional units. The contents (bits) of the SFRs are summarized in **Table 4** to **Table 12**, with the addresses of the bitaddressable SFRs appearing in bold typeface.

Note: Bits marked as 0 or 1 must be initialized per se, the functionality of the device with the other setting is not guaranteed.

The CPU SFRs can be accessed in both the standard and mapped memory areas (RMAP = 0 or 1).

Table 4 CPU Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 0 or 1										
81_H	**SP** **Reset: 07_H** Stack Pointer Register	Bit Field	SP							
		Type	rw							
82_H	**DPL** **Reset: 00_H** Data Pointer Register Low	Bit Field	DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
83_H	**DPH** **Reset: 00_H** Data Pointer Register High	Bit Field	DPH7	DPH6	DPH5	DPH4	DPH3	DPH2	DPH1	DPH0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
87_H	**PCON** **Reset: 00_H** Power Control Register	Bit Field	SMOD	0			GF1	GF0	0	IDLE
		Type	rw	r			rw	rw	r	rw
88_H	**TCON** **Reset: 00_H** Timer Control Register	Bit Field	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
		Type	rwh	rw	rwh	rw	rwh	rw	rwh	rw
89_H	**TMOD** **Reset: 00_H** Timer Mode Register	Bit Field	GATE1	0	T1M		GATE0	T0S	T0M	
		Type	rw	r	rw		rw	rw	rw	
$8A_H$	**TL0** **Reset: 00_H** Timer 0 Register Low	Bit Field	VAL							
		Type	rwh							
$8B_H$	**TL1** **Reset: 00_H** Timer 1 Register Low	Bit Field	VAL							
		Type	rwh							
$8C_H$	**TH0** **Reset: 00_H** Timer 0 Register High	Bit Field	VAL							
		Type	rwh							
$8D_H$	**TH1** **Reset: 00_H** Timer 1 Register High	Bit Field	VAL							
		Type	rwh							
98_H	**SCON** **Reset: 00_H** Serial Channel Control Register	Bit Field	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
		Type	rw	rw	rw	rw	rw	rwh	rwh	rwh
99_H	**SBUF** **Reset: 00_H** Serial Data Buffer Register	Bit Field	VAL							
		Type	rwh							
$A2_H$	**EO** **Reset: 00_H** Extended Operation Register	Bit Field	0			TRAP_EN	0			DPSEL0
		Type	r			rw	r			rw
$A8_H$	**IEN0** **Reset: 00_H** Interrupt Enable Register 0	Bit Field	EA	0	ET2	ES	ET1	EX1	ET0	EX0
		Type	rw	r	rw	rw	rw	rw	rw	rw
$B8_H$	**IP** **Reset: 00_H** Interrupt Priority Register	Bit Field	0		PT2	PS	PT1	PX1	PT0	PX0
		Type	r		rw	rw	rw	rw	rw	rw
$B9_H$	**IPH** **Reset: 00_H** Interrupt Priority Register High	Bit Field	0		PT2H	PSH	PT1H	PX1H	PT0H	PX0H
		Type	r		rw	rw	rw	rw	rw	rw
$D0_H$	**PSW** **Reset: 00_H** Program Status Word Register	Bit Field	CY	AC	F0	RS1	RS0	OV	F1	P
		Type	rwh	rwh	rw	rw	rw	rwh	rw	rh

Table 4 CPU Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
$E0_H$	**ACC** **Reset: 00_H** Accumulator Register	Bit Field	ACC7	ACC6	ACC5	ACC4	ACC3	ACC2	ACC1	ACC0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
$E8_H$	**IEN1** **Reset: 00_H** Interrupt Enable Register 1	Bit Field	ECCIP 3	ECCIP 2	ECCIP 1	ECCIP 0	EXM	EX2	ESSC	EADC
		Type	rw	rw	rw	rw	rw	rw	rw	rw
$F0_H$	**B** **Reset: 00_H** B Register	Bit Field	B7	B6	B5	B4	B3	B2	B1	B0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
$F8_H$	**IP1** **Reset: 00_H** Interrupt Priority Register 1	Bit Field	PCCIP 3	PCCIP 2	PCCIP 1	PCCIP 0	PXM	PX2	PSSC	PADC
		Type	rw	rw	rw	rw	rw	rw	rw	rw
$F9_H$	**IPH1** **Reset: 00_H** Interrupt Priority Register 1 High	Bit Field	PCCIP 3H	PCCIP 2H	PCCIP 1H	PCCIP 0H	PXMH	PX2H	PSSCH	PADC H
		Type	rw	rw	rw	rw	rw	rw	rw	rw

The system control SFRs can be accessed in the standard memory area (RMAP = 0). A special case is SYSCON0 which can be accessed in both the standard and mapped memory areas (RMAP = 0 or 1).

Table 5 SCU Register Summary

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 0 or 1										
$8F_H$	**SYSCON0** **Reset: 04_H** System Control Register 0	Bit Field	0					1	0	RMAP
		Type	r					rw	r	rw
RMAP = 0										
BF_H	**SCU_PAGE** **Reset: 00_H** Page Register	Bit Field	OP		STNR		0	PAGE		
		Type	w		w		r	rwh		
RMAP = 0, PAGE 0										
$B3_H$	**MODPISEL** **Reset: 00_H** Peripheral Input Select Register	Bit Field	0		JTAGT DIS	JTAGT CKS	0		EXINT 0IS	URRIS
		Type	r		rw	rw	r		rw	rw
$B4_H$	**IRCON0** **Reset: 00_H** Interrupt Request Register 0	Bit Field	0				EXINT 3	EXINT 2	EXINT 1	EXINT 0
		Type	r				rwh	rwh	rwh	rwh
$B5_H$	**IRCON1** **Reset: 00_H** Interrupt Request Register 1	Bit Field	0			ADCS RC1	ADCS RC0	RIR	TIR	EIR
		Type	r			rwh	rwh	rwh	rwh	rwh
$B7_H$	**EXICON0** **Reset: 00_H** External Interrupt Control Register 0	Bit Field	EXINT3		EXINT2		EXINT1		EXINT0	
		Type	rw		rw		rw		rw	
BB_H	**NMICON** **Reset: 00_H** NMI Control Register	Bit Field	0	NMI ECC	NMI VDDP	NMI VDD	NMI OCDS	NMI FLASH	NMI PLL	NMI WDT
		Type	r	rw	rw	rw	rw	rw	rw	rw
BC_H	**NMISR** **Reset: 00_H** NMI Status Register	Bit Field	0	FNMI ECC	FNMI VDDP	FNMI VDD	FNMI OCDS	FNMI FLASH	FNMI PLL	FNMI WDT
		Type	r	rwh	rwh	rwh	rwh	rwh	rwh	rwh
BD_H	**BCON** **Reset: 00_H** Baud Rate Control Register	Bit Field	BGSEL		0	BRDIS	BRPRE			R
		Type	rw		r	rw	rw			rw
BE_H	**BG** **Reset: 00_H** Baud Rate Timer/Reload Register	Bit Field	BR_VALUE							
		Type	rwh							

Table 5 SCU Register Summary (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
$E9_H$	**FDCON** **Reset: 00_H** Fractional Divider Control Register	Bit Field	BGS	SYNEN	ERRSYN	EOFSYN	BRK0	NDOV	FDM	FDEN
		Type	rw	rw	rwh	rwh	rwh	rwh	rw	rw
EA_H	**FDSTEP** **Reset: 00_H** Fractional Divider Reload Register	Bit Field	STEP							
		Type	rw							
EB_H	**FDRES** **Reset: 00_H** Fractional Divider Result Register	Bit Field	RESULT							
		Type	rh							
RMAP = 0, PAGE 1										
$B3_H$	**ID** **Reset: $1B_H$** Identity Register	Bit Field	PRODID					VERID		
		Type	r					rw		
$B4_H$	**PMCON0** **Reset: 00_H** Power Mode Control Register 0	Bit Field	0	WDTRST	WKRS	WKSEL	SD	PD	WS	
		Type	r	rwh	rwh	rw	rw	rwh	rw	
$B5_H$	**PMCON1** **Reset: 00_H** Power Mode Control Register 1	Bit Field	0				T2_DIS	CCU_DIS	SSC_DIS	ADC_DIS
		Type	r				rw	rw	rw	rw
$B7_H$	**PLL_CON** **Reset: 20_H** PLL Control Register	Bit Field	NDIV				VCOBYP	OSCDISC	RESLD	LOCK
		Type	rw				rw	rw	rwh	rh
BA_H	**CMCON** **Reset: 00_H** Clock Control Register	Bit Field	VCOSEL	0			CLKREL			
		Type	rw	r			rw			
BB_H	**PASSWD** **Reset: 07_H** Password Register	Bit Field	PASS					PROTECT_S	MODE	
		Type	wh					rh	rw	
BC_H	**FEAL** **Reset: 00_H** Flash Error Address Register Low	Bit Field	ECCERRADDR[7:0]							
		Type	rh							
BD_H	**FEAH** **Reset: 00_H** Flash Error Address Register High	Bit Field	ECCERRADDR[15:8]							
		Type	rh							
BE_H	**COCON** **Reset: 00_H** Clock Output Control Register	Bit Field	0		TLEN	COUTS	COREL			
		Type	r		rw	rw	rw			
$E9_H$	**MISC_CON** **Reset: 00_H** Miscellaneous Control Register	Bit Field	0							DFLASHEN
		Type	r							rwh
RMAP = 0, PAGE 3										
$B3_H$	**XADDRH** **Reset: $F0_H$** On-chip XRAM Address Higher Order	Bit Field	ADDRH							
		Type	rw							
$B4_H$	**IRCON3** **Reset: 00_H** Interrupt Request Register 3	Bit Field	0			CCU6SR1	0			CCU6SR0
		Type	r			rwh	r			rwh
$B5_H$	**IRCON4** **Reset: 00_H** Interrupt Request Register 4	Bit Field	0			CCU6SR3	0			CCU6SR2
		Type	r			rwh	r			rwh
BD_H	**MODSUSP** **Reset: 01_H** Module Suspend Control Register	Bit Field	0				T2SUSP	T13USP	T12USP	WDTSUSP
		Type	r				rw	rw	rw	rw

The WDT SFRs can be accessed in the mapped memory area (RMAP = 1).

Table 6 WDT Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 1										
BB_H	**WDTCON** **Reset: 00_H** Watchdog Timer Control Register	Bit Field	0		WINB EN	WDT PR	0	WDT EN	WDT RS	WDT IN
		Type	r		rw	rh	r	rw	rwh	rw
BC_H	**WDTREL** **Reset: 00_H** Watchdog Timer Reload Register	Bit Field	WDTREL							
		Type	rw							
BD_H	**WDTWINB** **Reset: 00_H** Watchdog Window-Boundary Count Register	Bit Field	WDTWINB							
		Type	rw							
BE_H	**WDTL** **Reset: 00_H** Watchdog Timer Register Low	Bit Field	WDT[7:0]							
		Type	rh							
BF_H	**WDTH** **Reset: 00_H** Watchdog Timer Register High	Bit Field	WDT[15:8]							
		Type	rh							

The Port SFRs can be accessed in the standard memory area (RMAP = 0).

Table 7 Port Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 0										
$B2_H$	**PORT_PAGE** **Reset: 00_H** Page Register for PORT	Bit Field	OP		STNR		0	PAGE		
		Type	w		w		r	rwh		
RMAP = 0, Page 0										
80_H	**P0_DATA** **Reset: 00_H** P0 Data Register	Bit Field	0		P5	P4	P3	P2	P1	P0
		Type	r		rwh	rwh	rwh	rwh	rwh	rwh
86_H	**P0_DIR** **Reset: 00_H** P0 Direction Register	Bit Field	0		P5	P4	P3	P2	P1	P0
		Type	r		rw	rw	rw	rw	rw	rw
90_H	**P1_DATA** **Reset: 00_H** P1 Data Register	Bit Field	0						P1	P0
		Type	rwh						rwh	rwh
91_H	**P1_DIR** **Reset: 00_H** P1 Direction Register	Bit Field	0						P1	P0
		Type	rw						rw	rw
$A0_H$	**P2_DATA** **Reset: 00_H** P2 Data Register	Bit Field	P7	0				P2	P1	P0
		Type	rwh	rwh				rwh	rwh	rwh
$A1_H$	**P2_DIR** **Reset: 00_H** P2 Direction Register	Bit Field	P7	0				P2	P1	P0
		Type	rw	rw				rw	rw	rw
$B0_H$	**P3_DATA** **Reset: 00_H** P3 Data Register	Bit Field	0						P1	P0
		Type	rwh						rwh	rwh
$B1_H$	**P3_DIR** **Reset: 00_H** P3 Direction Register	Bit Field	0						P1	P0
		Type	rw						rw	rw
RMAP = 0, Page 1										
80_H	**P0_PUDSEL** **Reset: FF_H** P0 Pull-Up/Pull-Down Select Register	Bit Field	1		P5	P4	P3	P2	P1	P0
		Type	r		rw	rw	rw	rw	rw	rw
86_H	**P0_PUDEN** **Reset: $C4_H$** P0 Pull-Up/Pull-Down Enable Register	Bit Field	1		P5	P4	P3	P2	P1	P0
		Type	r		rw	rw	rw	rw	rw	rw
90_H	**P1_PUDSEL** **Reset: FF_H** P1 Pull-Up/Pull-Down Select Register	Bit Field	1						P1	P0
		Type	rw						rw	rw
91_H	**P1_PUDEN** **Reset: FF_H** P1 Pull-Up/Pull-Down Enable Register	Bit Field	1						P1	P0
		Type	rw						rw	rw

Table 7 Port Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
$A0_H$	**P2_PUDSEL** **Reset: FF_H** P2 Pull-Up/Pull-Down Select Register	Bit Field	P7	1				P2	P1	P0
		Type	rw	rw				rw	rw	rw
$A1_H$	**P2_PUDEN** **Reset: 00_H** P2 Pull-Up/Pull-Down Enable Register	Bit Field	P7	0				P2	P1	P0
		Type	rw	rw				rw	rw	rw
$B0_H$	**P3_PUDSEL** **Reset: BF_H** P3 Pull-Up/Pull-Down Select Register	Bit Field	1	0	1				P1	P0
		Type	rw	rw	rw				rw	rw
$B1_H$	**P3_PUDEN** **Reset: 40_H** P3 Pull-Up/Pull-Down Enable Register	Bit Field	0	1	0				P1	P0
		Type	rw	rw	rw				rw	rw
RMAP = 0, Page 2										
80_H	**P0_ALTSEL0** **Reset: 00_H** P0 Alternate Select 0 Register	Bit Field	0		P5	P4	P3	P2	P1	P0
		Type	r		rw	rw	rw	rw	rw	rw
86_H	**P0_ALTSEL1** **Reset: 00_H** P0 Alternate Select 1 Register	Bit Field	0		P5	P4	P3	P2	P1	P0
		Type	r		rw	rw	rw	rw	rw	rw
90_H	**P1_ALTSEL0** **Reset: 00_H** P1 Alternate Select 0 Register	Bit Field	0						P1	P0
		Type	rw						rw	rw
91_H	**P1_ALTSEL1** **Reset: 00_H** P1 Alternate Select 1 Register	Bit Field	0						P1	P0
		Type	rw						rw	rw
$B0_H$	**P3_ALTSEL0** **Reset: 00_H** P3 Alternate Select 0 Register	Bit Field	0						P1	P0
		Type	rw						rw	rw
$B1_H$	**P3_ALTSEL1** **Reset: 00_H** P3 Alternate Select 1 Register	Bit Field	0						P1	P0
		Type	rw						rw	rw
RMAP = 0, Page 3										
80_H	**P0_OD** **Reset: 00_H** P0 Open Drain Control Register	Bit Field	0		P5	P4	P3	P2	P1	P0
		Type	r		rw	rw	rw	rw	rw	rw
90_H	**P1_OD** **Reset: 00_H** P1 Open Drain Control Register	Bit Field	0						P1	P0
		Type	rw						rw	rw
$B0_H$	**P3_OD** **Reset: 00_H** P3 Open Drain Control Register	Bit Field	0						P1	P0
		Type	rw						rw	rw

The ADC SFRs can be accessed in the standard memory area (RMAP = 0).

Table 8 ADC Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 0										
$D1_H$	**ADC_PAGE** **Reset: 00_H** Page Register for ADC	Bit Field	OP		STNR		0	PAGE		
		Type	w		w		r	rwh		
RMAP = 0, Page 0										
CA_H	**ADC_GLOBCTR** **Reset: 30_H** Global Control Register	Bit Field	ANON	DW	CTC		0			
		Type	rw	rw	rw		r			
CB_H	**ADC_GLOBSTR** **Reset: 00_H** Global Status Register	Bit Field	0		CHNR			0	SAM PLE	BUSY
		Type	r		rh			r	rh	rh
CC_H	**ADC_PRAR** **Reset: 00_H** Priority and Arbitration Register	Bit Field	ASEN1	ASEN0	0	ARBM	CSM1	PRIO1	CSM0	PRIO0
		Type	rw	rw	r	rw	rw	rw	rw	rw
CD_H	**ADC_LCBR** **Reset: $B7_H$** Limit Check Boundary Register	Bit Field	BOUND1				BOUND0			
		Type	rw				rw			
CE_H	**ADC_INPCR0** **Reset: 00_H** Input Class Register 0	Bit Field	STC							
		Type	rw							

Table 8 ADC Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
CF_H	**ADC_ETRCR** **Reset: 00_H** External Trigger Control Register	Bit Field	SYNEN1	SYNEN0	ETRSEL1			ETRSEL0		
		Type	rw	rw	rw			rw		
RMAP = 0, Page 1										
CA_H	**ADC_CHCTR0** **Reset: 00_H** Channel Control Register 0	Bit Field	0	LCC			0	RESRSEL		
		Type	r	rw			r	rw		
CB_H	**ADC_CHCTR1** **Reset: 00_H** Channel Control Register 1	Bit Field	0	LCC			0	RESRSEL		
		Type	r	rw			r	rw		
CC_H	**ADC_CHCTR2** **Reset: 00_H** Channel Control Register 2	Bit Field	0	LCC			0	RESRSEL		
		Type	r	rw			r	rw		
$D3_H$	**ADC_CHCTR7** **Reset: 00_H** Channel Control Register 7	Bit Field	0	LCC			0	RESRSEL		
		Type	r	rw			r	rw		
RMAP = 0, Page 2										
CA_H	**ADC_RESR0L** **Reset: 00_H** Result Register 0 Low	Bit Field	RESULT[1:0]		0	VF	DRC	CHNR		
		Type	rh		r	rh	rh	rh		
CB_H	**ADC_RESR0H** **Reset: 00_H** Result Register 0 High	Bit Field	RESULT[9:2]							
		Type	rh							
CC_H	**ADC_RESR1L** **Reset: 00_H** Result Register 1 Low	Bit Field	RESULT[1:0]		0	VF	DRC	CHNR		
		Type	rh		r	rh	rh	rh		
CD_H	**ADC_RESR1H** **Reset: 00_H** Result Register 1 High	Bit Field	RESULT[9:2]							
		Type	rh							
CE_H	**ADC_RESR2L** **Reset: 00_H** Result Register 2 Low	Bit Field	RESULT[1:0]		0	VF	DRC	CHNR		
		Type	rh		r	rh	rh	rh		
CF_H	**ADC_RESR2H** **Reset: 00_H** Result Register 2 High	Bit Field	RESULT[9:2]							
		Type	rh							
$D2_H$	**ADC_RESR3L** **Reset: 00_H** Result Register 3 Low	Bit Field	RESULT[1:0]		0	VF	DRC	CHNR		
		Type	rh		r	rh	rh	rh		
$D3_H$	**ADC_RESR3H** **Reset: 00_H** Result Register 3 High	Bit Field	RESULT[9:2]							
		Type	rh							
RMAP = 0, Page 3										
CA_H	**ADC_RESRA0L** **Reset: 00_H** Result Register 0, View A Low	Bit Field	RESULT[2:0]			VF	DRC	CHNR		
		Type	rh			rh	rh	rh		
CB_H	**ADC_RESRA0H** **Reset: 00_H** Result Register 0, View A High	Bit Field	RESULT[10:3]							
		Type	rh							
CC_H	**ADC_RESRA1L** **Reset: 00_H** Result Register 1, View A Low	Bit Field	RESULT[2:0]			VF	DRC	CHNR		
		Type	rh			rh	rh	rh		
CD_H	**ADC_RESRA1H** **Reset: 00_H** Result Register 1, View A High	Bit Field	RESULT[10:3]							
		Type	rh							
CE_H	**ADC_RESRA2L** **Reset: 00_H** Result Register 2, View A Low	Bit Field	RESULT[2:0]			VF	DRC	CHNR		
		Type	rh			rh	rh	rh		
CF_H	**ADC_RESRA2H** **Reset: 00_H** Result Register 2, View A High	Bit Field	RESULT[10:3]							
		Type	rh							
$D2_H$	**ADC_RESRA3L** **Reset: 00_H** Result Register 3, View A Low	Bit Field	RESULT[2:0]			VF	DRC	CHNR		
		Type	rh			rh	rh	rh		
$D3_H$	**ADC_RESRA3H** **Reset: 00_H** Result Register 3, View A High	Bit Field	RESULT[10:3]							
		Type	rh							
RMAP = 0, Page 4										
CA_H	**ADC_RCR0** **Reset: 00_H** Result Control Register 0	Bit Field	VFCTR	WFR	0	IEN	0			DRCTR
		Type	rw	rw	r	rw	r			rw

Table 8 ADC Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
CB_H	**ADC_RCR1** **Reset: 00_H** Result Control Register 1	Bit Field	VFCTR	WFR	0	IEN	0			DRCTR
		Type	rw	rw	r	rw	r			rw
CC_H	**ADC_RCR2** **Reset: 00_H** Result Control Register 2	Bit Field	VFCTR	WFR	0	IEN	0			DRCTR
		Type	rw	rw	r	rw	r			rw
CD_H	**ADC_RCR3** **Reset: 00_H** Result Control Register 3	Bit Field	VFCTR	WFR	0	IEN	0			DRCTR
		Type	rw	rw	r	rw	r			rw
CE_H	**ADC_VFCR** **Reset: 00_H** Valid Flag Clear Register	Bit Field	0				VFC3	VFC2	VFC1	VFC0
		Type	r				w	w	w	w
RMAP = 0, Page 5										
CA_H	**ADC_CHINFR** **Reset: 00_H** Channel Interrupt Flag Register	Bit Field	CHINF7	0				CHINF2	CHINF1	CHINF0
		Type	rh	rh				rh	rh	rh
CB_H	**ADC_CHINCR** **Reset: 00_H** Channel Interrupt Clear Register	Bit Field	CHINC7	0				CHINC2	CHINC1	CHINC0
		Type	w	w				w	w	w
CC_H	**ADC_CHINSR** **Reset: 00_H** Channel Interrupt Set Register	Bit Field	CHINS7	0				CHINS2	CHINS1	CHINS0
		Type	w	w				w	w	w
CD_H	**ADC_CHINPR** **Reset: 00_H** Channel Interrupt Node Pointer Register	Bit Field	CHINP7	0				CHINP2	CHINP1	CHINP0
		Type	rw	rw				rw	rw	rw
CE_H	**ADC_EVINFR** **Reset: 00_H** Event Interrupt Flag Register	Bit Field	EVINF7	EVINF6	EVINF5	EVINF4	0		EVINF1	EVINF0
		Type	rh	rh	rh	rh	r		rh	rh
CF_H	**ADC_EVINCR** **Reset: 00_H** Event Interrupt Clear Flag Register	Bit Field	EVINC7	EVINC6	EVINC5	EVINC4	0		EVINC1	EVINC0
		Type	w	w	w	w	r		w	w
$D2_H$	**ADC_EVINSR** **Reset: 00_H** Event Interrupt Set Flag Register	Bit Field	EVINS7	EVINS6	EVINS5	EVINS4	0		EVINS1	EVINS0
		Type	w	w	w	w	r		w	w
$D3_H$	**ADC_EVINPR** **Reset: 00_H** Event Interrupt Node Pointer Register	Bit Field	EVINP7	EVINP6	EVINP5	EVINP4	0		EVINP1	EVINP0
		Type	rw	rw	rw	rw	r		rw	rw
RMAP = 0, Page 6										
CA_H	**ADC_CRCR1** **Reset: 00_H** Conversion Request Control Register 1	Bit Field	CH7	0			0			
		Type	rwh	rwh			r			
CB_H	**ADC_CRPR1** **Reset: 00_H** Conversion Request Pending Register 1	Bit Field	CHP7	0			0			
		Type	rwh	rwh			r			
CC_H	**ADC_CRMR1** **Reset: 00_H** Conversion Request Mode Register 1	Bit Field	Rsv	LDEV	CLR PND	SCAN	ENSI	ENTR	0	ENGT
		Type	r	w	w	rw	rw	rw	r	rw
CD_H	**ADC_QMR0** **Reset: 00_H** Queue Mode Register 0	Bit Field	CEV	TREV	FLUSH	CLRV	TRMD	ENTR	0	ENGT
		Type	w	w	w	w	rw	rw	r	rw
CE_H	**ADC_QSR0** **Reset: 20_H** Queue Status Register 0	Bit Field	Rsv	0	EMPTY	EV	0			
		Type	r	r	rh	rh	r			
CF_H	**ADC_Q0R0** **Reset: 00_H** Queue 0 Register 0	Bit Field	EXTR	ENSI	RF	V	0	REQCHNR		
		Type	rh	rh	rh	rh	r	rh		
$D2_H$	**ADC_QBUR0** **Reset: 00_H** Queue Backup Register 0	Bit Field	EXTR	ENSI	RF	V	0	REQCHNR		
		Type	rh	rh	rh	rh	r	rh		

Table 8 ADC Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
$D2_H$	**ADC_QINR0** **Reset: 00_H** Queue Input Register 0	Bit Field	EXTR	ENSI	RF	0		REQCHNR		
		Type	w	w	w	r		w		

The Timer 2 SFRs can be accessed in the standard memory area (RMAP = 0).

Table 9 Timer 2 Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
$C0_H$	**T2_T2CON** **Reset: 00_H** Timer 2 Control Register	Bit Field	TF2	EXF2	0		EXEN2	TR2	0	$\overline{CP}$/RL2
		Type	rwh	rwh	r		rw	rwh	r	rw
$C1_H$	**T2_T2MOD** **Reset: 00_H** Timer 2 Mode Register	Bit Field	T2 REGS	T2 RHEN	EDGE SEL	PREN	T2PRE			DCEN
		Type	rw	rw	rw	rw	rw			rw
$C2_H$	**T2_RC2L** **Reset: 00_H** Timer 2 Reload/Capture Register Low	Bit Field	RC2[7:0]							
		Type	rwh							
$C3_H$	**T2_RC2H** **Reset: 00_H** Timer 2 Reload/Capture Register High	Bit Field	RC2[15:8]							
		Type	rwh							
$C4_H$	**T2_T2L** **Reset: 00_H** Timer 2 Register Low	Bit Field	THL2[7:0]							
		Type	rwh							
$C5_H$	**T2_T2H** **Reset: 00_H** Timer 2 Register High	Bit Field	THL2[15:8]							
		Type	rwh							

The CCU6 SFRs can be accessed in the standard memory area (RMAP = 0).

Table 10 CCU6 Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 0										
$A3_H$	**CCU6_PAGE** **Reset: 00_H** Page Register for CCU6	Bit Field	OP		STNR		0	PAGE		
		Type	w		w		r	rwh		
RMAP = 0, Page 0										
$9A_H$	**CCU6_CC63SRL** **Reset: 00_H** Capture/Compare Shadow Register for Channel CC63 Low	Bit Field	CC63SL							
		Type	rw							
$9B_H$	**CCU6_CC63SRH** **Reset: 00_H** Capture/Compare Shadow Register for Channel CC63 High	Bit Field	CC63SH							
		Type	rw							
$9C_H$	**CCU6_TCTR4L** **Reset: 00_H** Timer Control Register 4 Low	Bit Field	T12 STD	T12 STR	0		DTRES	T12 RES	T12RS	T12RR
		Type	w	w	r		w	w	w	w
$9D_H$	**CCU6_TCTR4H** **Reset: 00_H** Timer Control Register 4 High	Bit Field	T13 STD	T13 STR	0			T13 RES	T13RS	T13RR
		Type	w	w	r			w	w	w
$9E_H$	**CCU6_MCMOUTSL** **Reset: 00_H** Multi-Channel Mode Output Shadow Register Low	Bit Field	STRM CM	0	MCMPS					
		Type	w	r	rw					
$9F_H$	**CCU6_MCMOUTSH** **Reset: 00_H** Multi-Channel Mode Output Shadow Register High	Bit Field	STRHP	0	CURHS			EXPHS		
		Type	w	r	rw			rw		

Table 10 CCU6 Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
$A4_H$	**CCU6_ISRL Reset: 00_H** Capture/Compare Interrupt Status Reset Register Low	Bit Field	RT12PM	RT12OM	RCC62F	RCC62R	RCC61F	RCC61R	RCC60F	RCC60R
		Type	w	w	w	w	w	w	w	w
$A5_H$	**CCU6_ISRH Reset: 00_H** Capture/Compare Interrupt Status Reset Register High	Bit Field	RSTR	RIDLE	RWHE	RCHE	0	RTRPF	RT13PM	RT13CM
		Type	w	w	w	w	r	w	w	w
$A6_H$	**CCU6_CMPMODIFL Reset: 00_H** Compare State Modification Register Low	Bit Field	0	MCC63S	0			MCC62S	MCC61S	MCC60S
		Type	r	w	r			w	w	w
$A7_H$	**CCU6_CMPMODIFH Reset: 00_H** Compare State Modification Register High	Bit Field	0	MCC63R	0			MCC62R	MCC61R	MCC60R
		Type	r	w	r			w	w	w
FA_H	**CCU6_CC60SRL Reset: 00_H** Capture/Compare Shadow Register for Channel CC60 Low	Bit Field	CC60SL							
		Type	rwh							
FB_H	**CCU6_CC60SRH Reset: 00_H** Capture/Compare Shadow Register for Channel CC60 High	Bit Field	CC60SH							
		Type	rwh							
FC_H	**CCU6_CC61SRL Reset: 00_H** Capture/Compare Shadow Register for Channel CC61 Low	Bit Field	CC61SL							
		Type	rwh							
FD_H	**CCU6_CC61SRH Reset: 00_H** Capture/Compare Shadow Register for Channel CC61 High	Bit Field	CC61SH							
		Type	rwh							
FE_H	**CCU6_CC62SRL Reset: 00_H** Capture/Compare Shadow Register for Channel CC62 Low	Bit Field	CC62SL							
		Type	rwh							
FF_H	**CCU6_CC62SRH Reset: 00_H** Capture/Compare Shadow Register for Channel CC62 High	Bit Field	CC62SH							
		Type	rwh							
RMAP = 0, Page 1										
$9A_H$	**CCU6_CC63RL Reset: 00_H** Capture/Compare Register for Channel CC63 Low	Bit Field	CC63VL							
		Type	rh							
$9B_H$	**CCU6_CC63RH Reset: 00_H** Capture/Compare Register for Channel CC63 High	Bit Field	CC63VH							
		Type	rh							
$9C_H$	**CCU6_T12PRL Reset: 00_H** Timer T12 Period Register Low	Bit Field	T12PVL							
		Type	rwh							
$9D_H$	**CCU6_T12PRH Reset: 00_H** Timer T12 Period Register High	Bit Field	T12PVH							
		Type	rwh							
$9E_H$	**CCU6_T13PRL Reset: 00_H** Timer T13 Period Register Low	Bit Field	T13PVL							
		Type	rwh							
$9F_H$	**CCU6_T13PRH Reset: 00_H** Timer T13 Period Register High	Bit Field	T13PVH							
		Type	rwh							
$A4_H$	**CCU6_T12DTCL Reset: 00_H** Dead-Time Control Register for Timer T12 Low	Bit Field	DTM							
		Type	rw							
$A5_H$	**CCU6_T12DTCH Reset: 00_H** Dead-Time Control Register for Timer T12 High	Bit Field	0	DTR2	DTR1	DTR0	0	DTE2	DTE1	DTE0
		Type	r	rh	rh	rh	r	rw	rw	rw

Table 10 CCU6 Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
$A6_H$	**CCU6_TCTR0L** **Reset: 00_H** Timer Control Register 0 Low	Bit Field	CTM	CDIR	STE12	T12R	T12 PRE	T12CLK		
		Type	rw	rh	rh	rh	rw	rw		
$A7_H$	**CCU6_TCTR0H** **Reset: 00_H** Timer Control Register 0 High	Bit Field	0		STE13	T13R	T13 PRE	T13CLK		
		Type	r		rh	rh	rw	rw		
FA_H	**CCU6_CC60RL** **Reset: 00_H** Capture/Compare Register for Channel CC60 Low	Bit Field	CC60VL							
		Type	rh							
FB_H	**CCU6_CC60RH** **Reset: 00_H** Capture/Compare Register for Channel CC60 High	Bit Field	CC60VH							
		Type	rh							
FC_H	**CCU6_CC61RL** **Reset: 00_H** Capture/Compare Register for Channel CC61 Low	Bit Field	CC61VL							
		Type	rh							
FD_H	**CCU6_CC61RH** **Reset: 00_H** Capture/Compare Register for Channel CC61 High	Bit Field	CC61VH							
		Type	rh							
FE_H	**CCU6_CC62RL** **Reset: 00_H** Capture/Compare Register for Channel CC62 Low	Bit Field	CC62VL							
		Type	rh							
FF_H	**CCU6_CC62RH** **Reset: 00_H** Capture/Compare Register for Channel CC62 High	Bit Field	CC62VH							
		Type	rh							
RMAP = 0, Page 2										
$9A_H$	**CCU6_T12MSELL** **Reset: 00_H** T12 Capture/Compare Mode Select Register Low	Bit Field	MSEL61				MSEL60			
		Type	rw				rw			
$9B_H$	**CCU6_T12MSELH** **Reset: 00_H** T12 Capture/Compare Mode Select Register High	Bit Field	DBYP	HSYNC			MSEL62			
		Type	rw	rw			rw			
$9C_H$	**CCU6_IENL** **Reset: 00_H** Capture/Compare Interrupt Enable Register Low	Bit Field	ENT12 PM	ENT12 OM	ENCC 62F	ENCC 62R	ENCC 61F	ENCC 61R	ENCC 60F	ENCC 60R
		Type	rw	rw	rw	rw	rw	rw	rw	rw
$9D_H$	**CCU6_IENH** **Reset: 00_H** Capture/Compare Interrupt Enable Register High	Bit Field	ENSTR	EN IDLE	EN WHE	EN CHE	0	EN TRPF	ENT13 PM	ENT13 CM
		Type	rw	rw	rw	rw	r	rw	rw	rw
$9E_H$	**CCU6_INPL** **Reset: 40_H** Capture/Compare Interrupt Node Pointer Register Low	Bit Field	INPCHE		INPCC62		INPCC61		INPCC60	
		Type	rw		rw		rw		rw	
$9F_H$	**CCU6_INPH** **Reset: 39_H** Capture/Compare Interrupt Node Pointer Register High	Bit Field	0		INPT13		INPT12		INPERR	
		Type	r		rw		rw		rw	
$A4_H$	**CCU6_ISSL** **Reset: 00_H** Capture/Compare Interrupt Status Set Register Low	Bit Field	ST12PM	ST12OM	SCC62F	SCC62R	SCC61F	SCC61R	SCC60F	SCC60R
		Type	w	w	w	w	w	w	w	w
$A5_H$	**CCU6_ISSH** **Reset: 00_H** Capture/Compare Interrupt Status Set Register High	Bit Field	SSTR	SIDLE	SWHE	SCHE	SWHC	STRPF	ST13 PM	ST13 CM
		Type	w	w	w	w	w	w	w	w
$A6_H$	**CCU6_PSLR** **Reset: 00_H** Passive State Level Register	Bit Field	PSL63	0	PSL					
		Type	rwh	r	rwh					
$A7_H$	**CCU6_MCMCTR** **Reset: 00_H** Multi-Channel Mode Control Register	Bit Field	0		SWSYN		0	SWSEL		
		Type	r		rw		r	rw		

Table 10 CCU6 Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
FA_H	**CCU6_TCTR2L** **Reset: 00_H** Timer Control Register 2 Low	Bit Field	0	T13TED		T13TEC			T13 SSC	T12 SSC
		Type	r	rw		rw			rw	rw
FB_H	**CCU6_TCTR2H** **Reset: 00_H** Timer Control Register 2 High	Bit Field	0				T13RSEL		T12RSEL	
		Type	r				rw		rw	
FC_H	**CCU6_MODCTRL** **Reset: 00_H** Modulation Control Register Low	Bit Field	MC MEN	0	T12MODEN					
		Type	rw	r	rw					
FD_H	**CCU6_MODCTRH** **Reset: 00_H** Modulation Control Register High	Bit Field	ECT13O	0	T13MODEN					
		Type	rw	r	rw					
FE_H	**CCU6_TRPCTRL** **Reset: 00_H** Trap Control Register Low	Bit Field	0					TRPM2	TRPM1	TRPM0
		Type	r					rw	rw	rw
FF_H	**CCU6_TRPCTRH** **Reset: 00_H** Trap Control Register High	Bit Field	TRPPEN	TRPEN13	TRPEN					
		Type	rw	rw	rw					
RMAP = 0, Page 3										
$9A_H$	**CCU6_MCMOUTL** **Reset: 00_H** Multi-Channel Mode Output Register Low	Bit Field	0	R	MCMP					
		Type	r	rh	rh					
$9B_H$	**CCU6_MCMOUTH** **Reset: 00_H** Multi-Channel Mode Output Register High	Bit Field	0		CURH			EXPH		
		Type	r		rh			rh		
$9C_H$	**CCU6_ISL** **Reset: 00_H** Capture/Compare Interrupt Status Register Low	Bit Field	T12PM	T12OM	ICC62F	ICC62R	ICC61F	ICC61R	ICC60F	ICC60R
		Type	rh	rh	rh	rh	rh	rh	rh	rh
$9D_H$	**CCU6_ISH** **Reset: 00_H** Capture/Compare Interrupt Status Register High	Bit Field	STR	IDLE	WHE	CHE	TRPS	TRPF	T13PM	T13CM
		Type	rh	rh	rh	rh	rh	rh	rh	rh
$9E_H$	**CCU6_PISEL0L** **Reset: 00_H** Port Input Select Register 0 Low	Bit Field	ISTRP		ISCC62		ISCC61		ISCC60	
		Type	rw		rw		rw		rw	
$9F_H$	**CCU6_PISEL0H** **Reset: 00_H** Port Input Select Register 0 High	Bit Field	IST12HR		ISPOS2		ISPOS1		ISPOS0	
		Type	rw		rw		rw		rw	
$A4_H$	**CCU6_PISEL2** **Reset: 00_H** Port Input Select Register 2	Bit Field	0						IST13HR	
		Type	r						rw	
FA_H	**CCU6_T12L** **Reset: 00_H** Timer T12 Counter Register Low	Bit Field	T12CVL							
		Type	rwh							
FB_H	**CCU6_T12H** **Reset: 00_H** Timer T12 Counter Register High	Bit Field	T12CVH							
		Type	rwh							
FC_H	**CCU6_T13L** **Reset: 00_H** Timer T13 Counter Register Low	Bit Field	T13CVL							
		Type	rwh							
FD_H	**CCU6_T13H** **Reset: 00_H** Timer T13 Counter Register High	Bit Field	T13CVH							
		Type	rwh							
FE_H	**CCU6_CMPSTATL** **Reset: 00_H** Compare State Register Low	Bit Field	0	CC63ST	CCPOS2	CCPOS1	CCPOS0	CC62ST	CC61ST	CC60ST
		Type	r	rh	rh	rh	rh	rh	rh	rh
FF_H	**CCU6_CMPSTATH** **Reset: 00_H** Compare State Register High	Bit Field	T13IM	COUT63PS	COUT62PS	CC62PS	COUT61PS	CC61PS	COUT60PS	CC60PS
		Type	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

The SSC SFRs can be accessed in the standard memory area (RMAP = 0).

Table 11 SSC Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 0										
$A9_H$	**SSC_PISEL** **Reset: 00_H** Port Input Select Register	Bit Field	0					CIS	SIS	MIS
		Type	r					rw	rw	rw
AA_H	**SSC_CONL** **Reset: 00_H** Control Register Low *Programming Mode*	Bit Field	LB	PO	PH	HB	BM			
		Type	rw	rw	rw	rw	rw			
AA_H	**SSC_CONL** **Reset: 00_H** Control Register Low *Operating Mode*	Bit Field	0				BC			
		Type	r				rh			
AB_H	**SSC_CONH** **Reset: 00_H** Control Register High *Programming Mode*	Bit Field	EN	MS	0	AREN	BEN	PEN	REN	TEN
		Type	rw	rw	r	rw	rw	rw	rw	rw
AB_H	**SSC_CONH** **Reset: 00_H** Control Register High *Operating Mode*	Bit Field	EN	MS	0	BSY	BE	PE	RE	TE
		Type	rw	rw	r	rh	rwh	rwh	rwh	rwh
AC_H	**SSC_TBL** **Reset: 00_H** Transmitter Buffer Register Low	Bit Field	TB_VALUE							
		Type	rw							
AD_H	**SSC_RBL** **Reset: 00_H** Receiver Buffer Register Low	Bit Field	RB_VALUE							
		Type	rh							
AE_H	**SSC_BRL** **Reset: 00_H** Baudrate Timer Reload Register Low	Bit Field	BR_VALUE[7:0]							
		Type	rw							
AF_H	**SSC_BRH** **Reset: 00_H** Baudrate Timer Reload Register High	Bit Field	BR_VALUE[15:8]							
		Type	rw							

The OCDS SFRs can be accessed in the mapped memory area (RMAP = 1). Certain bits (marked with asterisk) are writable by Monitor program only, in Monitor Mode. In general, user code shall not access these SFRs.

Table 12 OCDS Register Summary

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 1										
$E9_H$	**MMCR2** **Reset: 1010 000X$_B$** Monitor Mode Control 2 Register	Bit Field	STMODE	EXBC	DSUSP	MBCON	0	MMEP	MMODE	JENA
		Type	r	rw	rw	rwh	rw	rwh	rh	rh
EB_H	**MMWR1** **Reset: 00_H** Monitor Work Register 1	Bit Field	MMWR1							
		Type	rw							
EC_H	**MMWR2** **Reset: 00_H** Monitor Work Register 2	Bit Field	MMWR2							
		Type	rw							
$F1_H$	**MMCR** **Reset: 00_H** Monitor Mode Control Register	Bit Field	MEXIT_P	MEXIT	0	MSTEP	MRAMS_P	MRAMS	TRF	RRF
		Type	w	rwh	r	rw	w	rwh	rh	rh
$F2_H$	**MMSR** **Reset: 00_H** Monitor Mode Status Register	Bit Field	MBCAM	MBCIN	EXBF	SWBF	HWB3F	HWB2F	HWB1F	HWB0F
		Type	rw	rwh	rwh	rwh	rwh	rwh	rwh	rwh
$F3_H$	**MMBPCR** **Reset: 00_H** BreakPoints Control Register	Bit Field	SWBC	HWB3C		HWB2C		HWB1C	HWB0C	
		Type	rw	rw		rw		rw	rw	

Table 12 OCDS Register Summary (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
$F4_H$	**MMICR** **Reset: 00_H** Monitor Mode Interrupt Control Register	Bit Field	DVECT	DRETR	COMRST	MSTSEL	MMUIE_P	MMUIE	RRIE_P	RRIE
		Type	rwh	rwh	rwh	rh	w	rw	w	rw
$F5_H$	**MMDR** **Reset: 00_H** Monitor Mode Data Transfer Register *Receive*	Bit Field	MMRR							
		Type	rh							
$F6_H$	**HWBPSR** **Reset: 00_H** Hardware Breakpoints Select Register	Bit Field	0			BPSEL_P	BPSEL			
		Type	r			w	rw			
$F7_H$	**HWBPDR** **Reset: 00_H** Hardware Breakpoints Data Register	Bit Field	HWBPxx							
		Type	rw							

3.3 Flash Memory

The Flash memory provides an embedded user-programmable non-volatile memory, allowing fast and reliable storage of user code and data. It is operated from a single 2.5 V supply from the Embedded Voltage Regulator (EVR) and does not require additional programming or erasing voltage. The sectorization of the Flash memory allows each sector to be erased independently.

Features:

- In-System Programming (ISP) via UART
- In-Application Programming (IAP)
- Error Correction Code (ECC) for dynamic correction of single-bit errors
- Background program and erase operations for CPU load minimization
- Support for aborting erase operation
- 32-byte minimum program width[1)]
- 1-sector minimum erase width
- 1-byte read access
- Operating supply voltage: 2.5 V ± 7.5 %
- Read access time: 3 × t_{CCLK} = 112.5 ns[2)]
- Program time: 209440 / f_{SYS} = 2.6 ms[3)]
- Erase time: 8175360 / f_{SYS} = 102 ms[3)]

Table 13 shows the Flash data retention and endurance targets.

Table 13 Flash Data Retention and Endurance (Operating Conditions apply)

Retention	Endurance[1)]	Size
20 years	1,000 cycles	up to 4 Kbytes
5 years	10,000 cycles	1 Kbyte
2 years	70,000 cycles	512 bytes
2 years	100,000 cycles	128 bytes

1) One cycle refers to the programming of all wordlines in a sector and erasing of sector. The Flash endurance data specified in **Table 13** is valid only if the following conditions are fulfilled:
- the maximum number of erase cycles per Flash sector must not exceed 100,000 cycles.
- the maximum number of erase cycles per Flash bank must not exceed 300,000 cycles.
- the maximum number of program cycles per Flash bank must not exceed 2,500,000 cycles.

1) 32-byte wordline can be programmed twice, i.e., two gate disturbs allowed.
2) Values shown here are typical values. f_{sys} = 80 MHz ± 7.5% (f_{CCLK} = 26.7 MHz ± 7.5 %) is the maximum frequency range for Flash read access.
3) Values shown here are typical values. f_{sys} = 80 MHz ± 7.5% is the only frequency range for Flash programming and erasing. f_{sysmin} is used for obtaining the worst case timing.

3.3.1 Flash Bank Sectorization

The XC864 has 4 Kbytes of embedded Flash memory. The Flash bank sectorization is shown in **Figure 10**.

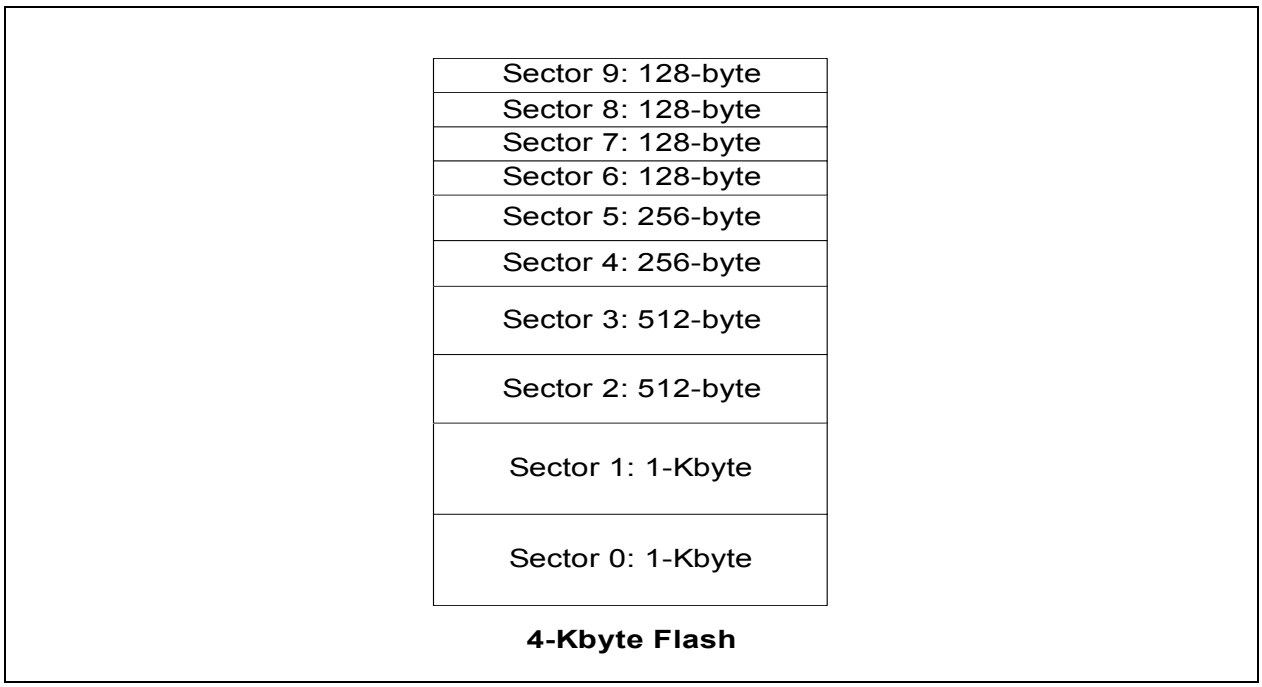


Figure 10 Flash Bank Sectorization

The minimum erase width is always a complete sector, and sectors can be erased separately or in parallel. Contrary to standard EPROMs, erased Flash memory cells contain 0s.

Flash memory can be used for code and data storage. The Flash bank is divided into several physical sectors for extended erasing and reprogramming capability; even numbers for each sector size are provided to allow greater flexibility and the ability to adapt to a wide range of application requirements.

It must be noted that the Flash is double mapped to two address range in the code space: 0000_H – $0FFF_H$ and $A000_H$ – $AFFF_H$. Accessing 0000_H or $A000_H$ is physically accessing the same Flash location, and likewise for corresponding addresses within each range.

3.3.2 Flash Programming Without Erase

The same WL can be programmed twice before erasing is required as the Flash cells are able to withstand two gate disturbs. This means if the number of data bytes that needs to be written is smaller than the 32-byte minimum programming width, the user can opt to program this number of data bytes (x; where x can be any integer from 1 to 31) first and program the remaining bytes (32 - x) later. Hence, it is possible to program the same WL, for example, with 16 bytes of data in two times (see **Figure 11**).

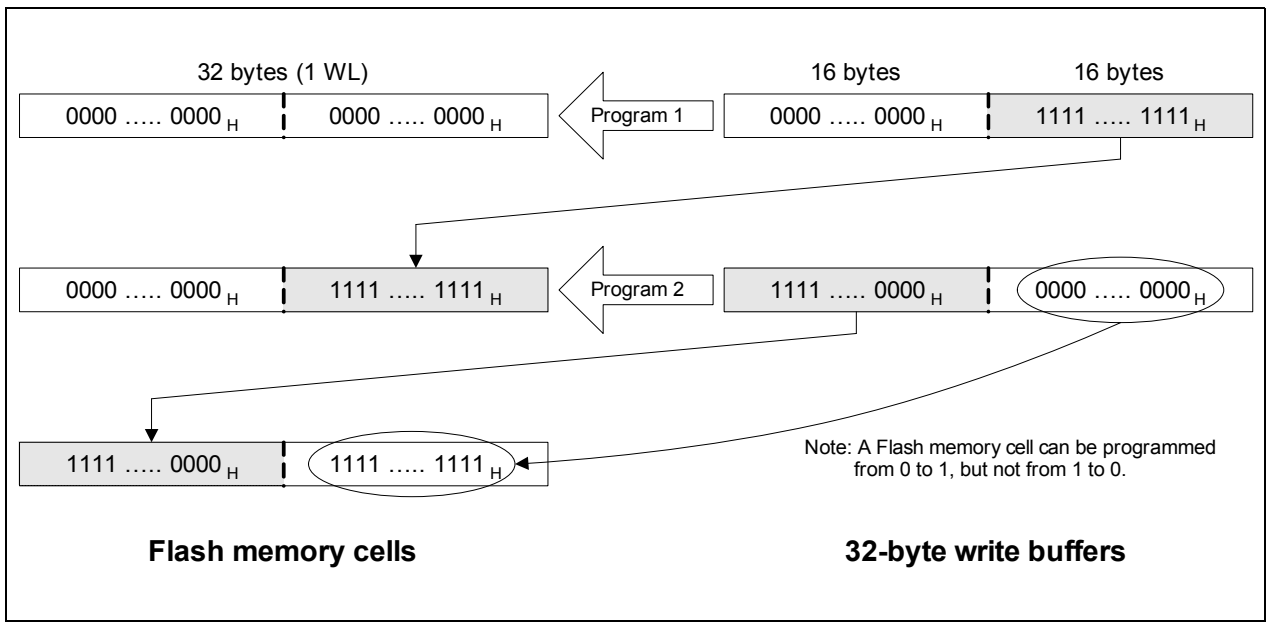


Figure 11 Flash Programming Without Erase

Note: When programming a WL the second time, the previously programmed Flash memory cells (whether 0s or 1s) should be reprogrammed with 0s to retain its original contents and to prevent “over-programming”.

3.3.3 In-Application Programming

In some applications, the Flash contents may need to be modified during program execution. In-Application Programming (IAP) is supported so that users can program or erase the Flash memory from their Flash user program by calling some special subroutines. The Flash subroutines will first perform some checks and an initialization sequence before starting the program or erase operation. A manual check on the Flash data is necessary to determine if the programming or erasing was successful via using the 'MOVC' instruction to read out the Flash contents. Other special subroutines include aborting the Flash erase operation and checking the Flash bank ready-to-read status.

3.3.3.1 Flash Programming

Each call of the Flash program subroutine allows the programming of 32 bytes of data into the selected wordline (WL) of the Flash bank. Before calling the Flash program subroutine, the user must ensure that required inputs (**Table 14** and **Table 15**) are provided.

Flash Program Subroutine Type 1

If valid inputs have been set up, calling the subroutine begins flash programming. The subroutine exits and returns to the user code, while the target Flash bank is still in program mode, and is not accessible by user code.

The user code continues execution until the Flash NMI event is generated; bit FNMIFLASH in register NMISR is set, and if enabled via NMIFLASH, an NMI to the CPU is triggered to enter the Flash NMI service routine. At this point, the Flash bank is in ready-to-read mode.

Table 14 Flash Program Subroutin Type 1

Subroutine	$DFF6_H$: FSM_PROG
Input	DPTR (DPH, DPL[1]): Flash WL address
	R0 of Register Bank 3 (IRAM address 18_H): IRAM start address for 32-byte Flash data
	32-byte Flash data
	Flash NMI (NMICON.NMIFLASH) is enabled (1) or disabled (0)
Output	PSW.CY: 0 = Flash programming is in progress 1 = Flash programming is not started Flag FNMIFLASH will be set when Flash programming has successfully completed.
	DPTR is incremented by 20_H[2]

Table 14 Flash Program Subroutin Type 1 (cont'd)

Stack size required	12
Resource used/ destroyed	ACC, B, SCU_PAGE
	R0 – R7 of Register Bank 3 (IRAM address 18_H – $1F_H$) (8 bytes) IRAM address 36_H – $3D_H$ (8 bytes)

1) The last 5 LSB of the DPL is 0 for an aligned WL address, for e.g. 00_H, 20_H, 40_H, 60_H, 80_H, $A0_H$, $C0_H$ and $E0_H$..

2) DPTR is only incremented by 20_H when PSW.CY is 0.

Flash Program Subroutine Type 2

This routine will wait until Flash programming is completed before the user code can continue its execution. Therefore, background programming is not supported. This type of routine can be used to program the Flash bank where the user code is in execution. The Flash cannot be in both program mode and read mode at the same time. It can also be used for programming the Flash bank where the interrupt vectors are defined as interrupts cannot be handled when the Flash is in program mode.

Note: For the Flash programming of XC864 device, Flash Program Subroutine Type 2 is allowed. The users can also use Flash Program Subroutine Type 1 if it is called from XRAM.

Table 15 Flash Program Subroutine Type 2

Subroutine	$DFDB_H$: FSM_PROG_NO_BG
Input	DPTR (DPH, DPL[1]): Flash WL address
	R0 of Register Bank 3 (IRAM address 18_H): IRAM start address for 32-byte Flash data
	32-byte Flash data
	All interrupts including NMI must be disabled (0) Set SFR NMISR = 00_H
Output	PSW.CY: 0 = Flash programming is successful 1 = Flash programming is not successful due to: Flash Protection Mode 1 is enabled, or NMI has occurred Flag FNMIFLASH is cleared by this routine before return to user code.
	DPTR is incremented by 20_H[2]
Stack size required	15

Table 15 Flash Program Subroutine Type 2 (cont'd)

Resource used/ destroyed	ACC, B, SCU_PAGE
	R0 – R7 of Register Bank 3 (IRAM address 18_H – $1F_H$) (8 bytes) IRAM address 36_H – $3D_H$ (8 bytes)

1) The last 5 LSB of the DPL is 0 for an aligned WL address, for e.g. 00_H, 20_H, 40_H, 60_H, 80_H, $A0_H$, $C0_H$ and $E0_H$..

2) DPTR is only incremented by 20_H when PSW.CY is 0.

3.3.3.2 Flash Erasing

Each call of the Flash erase subroutine allows the user to select one sector or a combination of several sectors for erase. Before calling the Flash erase subroutine, the user must ensure that required inputs (**Table 16** and **Table 17**) are provided. Also, protected Flash banks should not be targeted for erase.

Flash Erase Subroutine Type 1

If valid inputs have been set up, calling the subroutine begins flash erasing. The subroutine exits and returns to the user code, while the target Flash bank is still in erase mode, and is not accessible by user code.

Table 16 Flash Erase Subroutine Type 1

Subroutine	$DFF9_H$: FLASH_ERASE
Input[1)]	R3 of Register Bank 3 (IRAM address $1B_H$): Select sector(s) to be erased. LSB represents sector 0, MSB represents sector 7.
	R4 of Register Bank 3 (IRAM address $1C_H$): Select sector(s) to be erased. LSB represents sector 8, bit 1 represents sector 9.
	Flash NMI (NMICON.NMIFLASH) is enabled (1) or disabled (0)
	MISC_CON.DFLASHEN[2)] bit = 1
Output	PSW.CY: 0 = Flash erasing is in progress 1 = Flash erasing is not started Flag FNMIFLASH will be set when Flash erasing has successfully completed.
Stack size required	10
Resource used/ destroyed	ACC, B, SCU_PAGE
	R0 – R7 of Register Bank 3 (IRAM address 18_H – $1F_H$) (8 bytes)
	IRAM address 36_H – $3D_H$ (8 bytes)

1) The inputs should be set as 0 if the sector(s) of the bank(s) is/are not to be selected for erasing.

2) When Flash Protection Mode 0 is enabled, in order to erase Flash bank, DFLASHEN bit needs to be set.

Flash Erase Subroutine Type 2

This routine will wait until Flash erasing is completed before the user code can continue its execution. Therefore, background erasing is not supported. This type of routine can be used to erase the Flash bank where the user code is in execution. The Flash cannot be in both erase mode and read mode at the same time. It can also be used for erasing the Flash bank where the interrupt vectors are defined as interrupts cannot be handled when the Flash is in erase mode.

This routine will be aborted if the FNMIVDDP, FNMIVDD or FNMIPLL flag is set while they are being polled for error by the routine.

Note: For the Flash erasing of XC864 device, Flash Erase Subroutine Type 2 is allowed. The users can also use Flash Erase Subroutine Type 1 if it is called from XRAM.

Table 17 Flash Erase Subroutine Type 2

Subroutine	$DFDE_H$: FLASH_ERASE_NO_BG
Input[1)]	R3 of Register Bank 3 (IRAM address $1B_H$): Select sector(s) to be erased for the Flash bank. LSB represents sector 0, MSB represents sector 7.
	R4 of Register Bank 3 (IRAM address $1C_H$): Select sector(s) to be erased for the Flash bank. LSB represents sector 8, bit 1 represents sector 9.
	All interrupts including NMI must be disabled (0) SET SFR NMISR = 00_H.
	MISC_CON.DFLASHEN[2)] bit = 1
Output	PSW.CY: 0 = Flash erasing is successful 1 = Flash erasing is not successful due to: MISC_CON.DFLASHEN bit is not set when Flash Protection Mode 0 is enabled, or Flash Protection Mode 1 is enabled, or NMI has occurred[3)] Flag FNMIFLASH will be set when Flash erasing has successfully completed.
Stack size required	13

Table 17 Flash Erase Subroutine (cont'd)**Type 2**

Resource used/ destroyed	ACC, B, SCU_PAGE
	R0 – R7 of Register Bank 3 (IRAM address 18_H – $1F_H$) (8 bytes)
	IRAM address 36_H – $3D_H$ (8 bytes)

1) The inputs should be set as 0 if the sector(s) of the bank(s) is/are not to be selected for erasing.

2) When Flash Protection Mode 0 is enabled, in order to erase Flash bank, DFLASHEN bit needs to be set.If DFLASHEN is not set, PSW.CY will be set to 1.

3) NMISR is checked for critical NMI events, namely NMIVDDP, NMIVDD, and NMIPLL.

3.4 Interrupt System

The XC800 Core supports one non-maskable interrupt (NMI) and 14 maskable interrupt nodes. In addition to the standard interrupt functions supported by the core, e.g., configurable interrupt priority and interrupt masking, the interrupt system provides extended interrupt support capabilities such as mapping interrupt events to interrupt nodes to increase the number of interrupt sources supported, and additional status registers for detecting and determining the interrupt source.

3.4.1 Interrupt Source

Figure 12 to **Figure 16** give a general overview of the interrupt sources and illustrates the request and control flags.

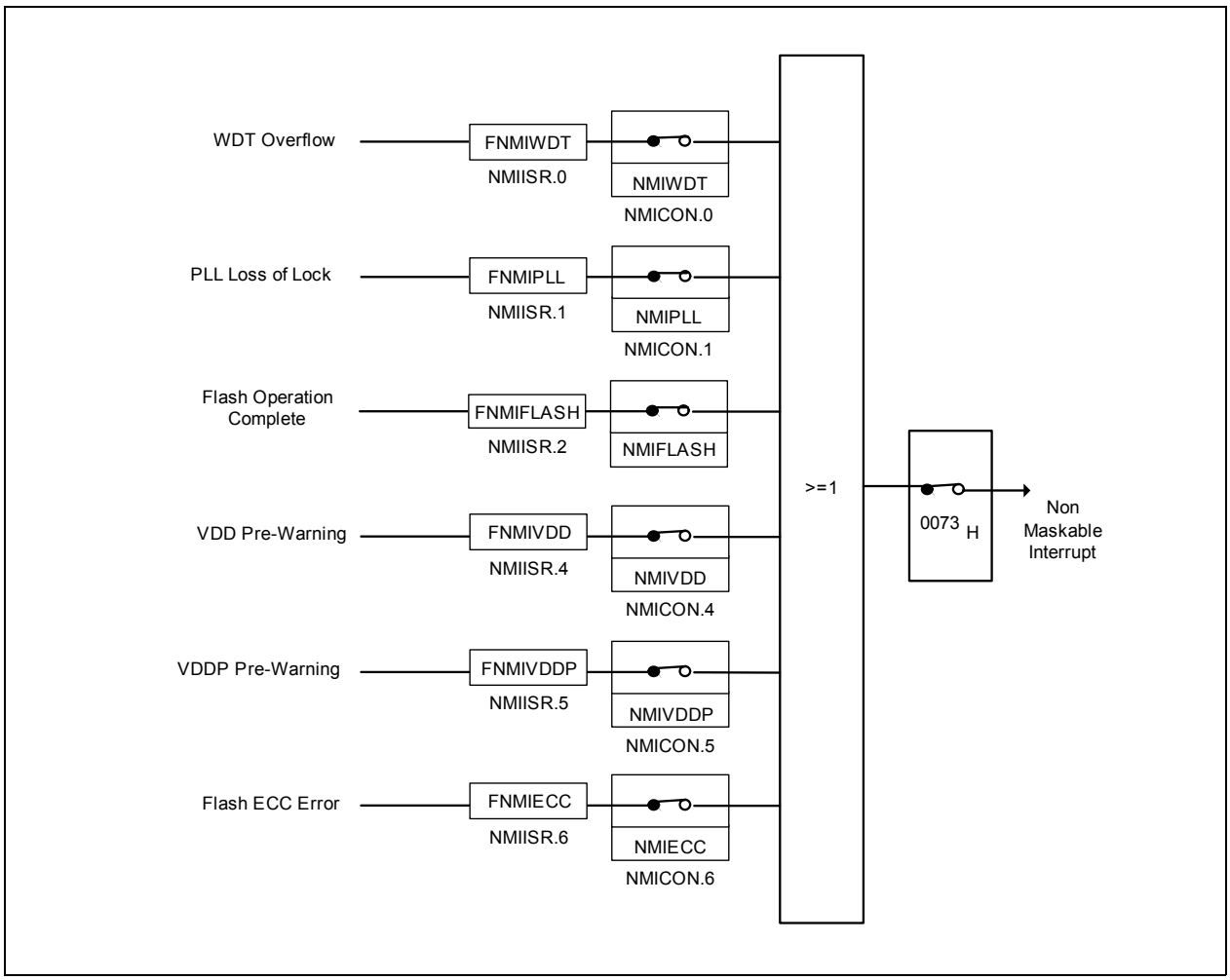


Figure 12 Non-Maskable Interrupt Request Sources

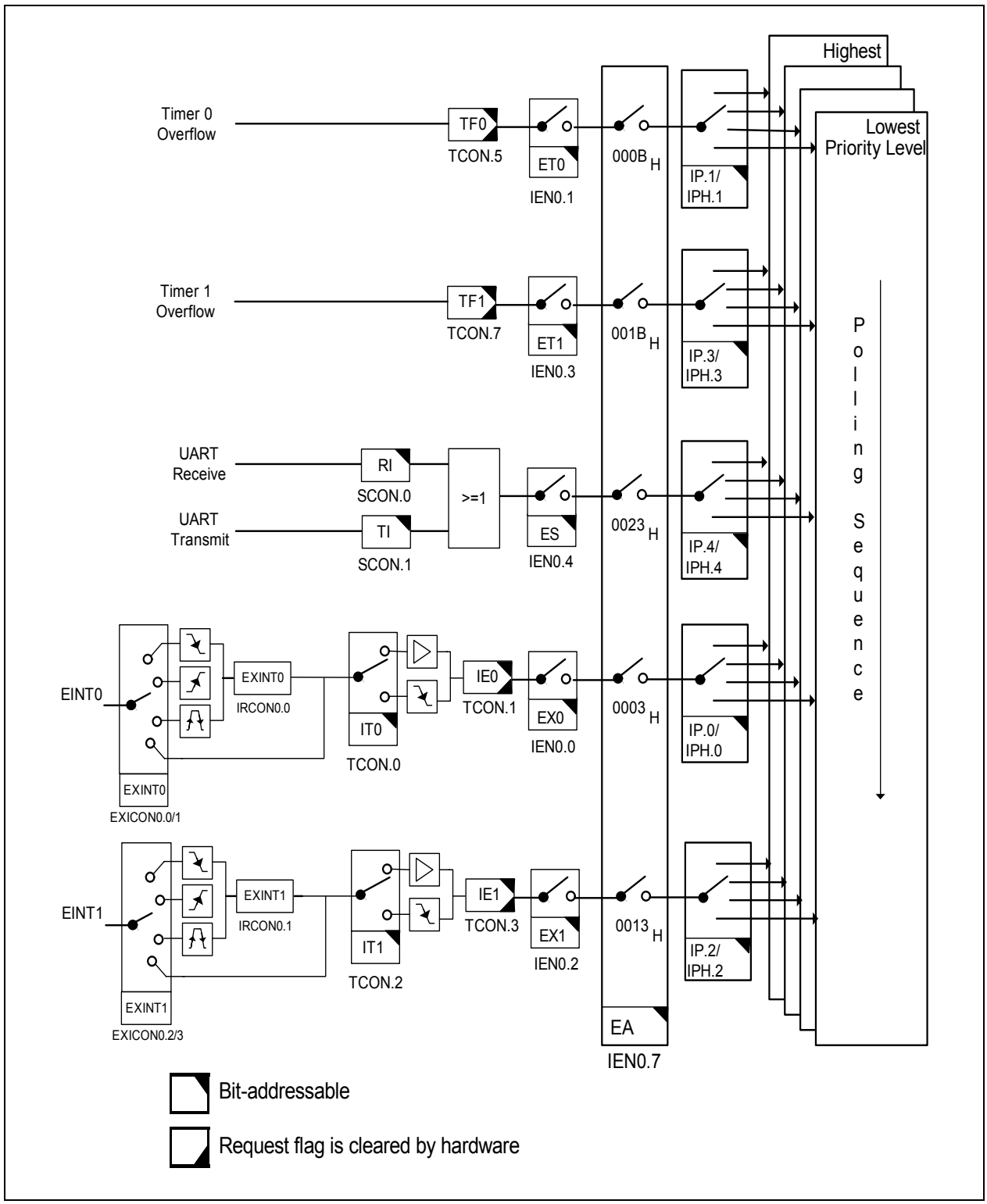


Figure 13 Interrupt Request Sources (Part 1)

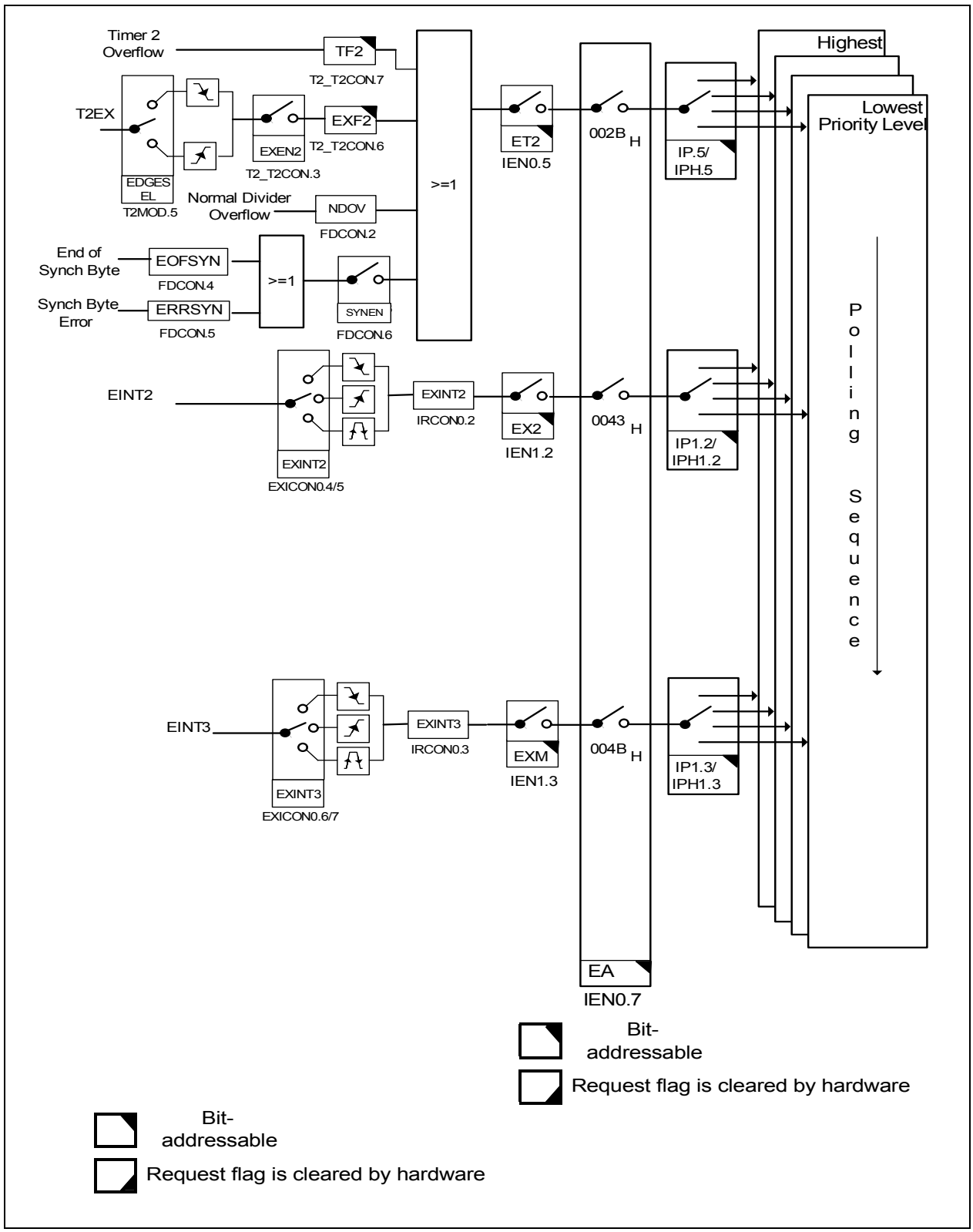


Figure 14 Interrupt Request Sources (Part 2)

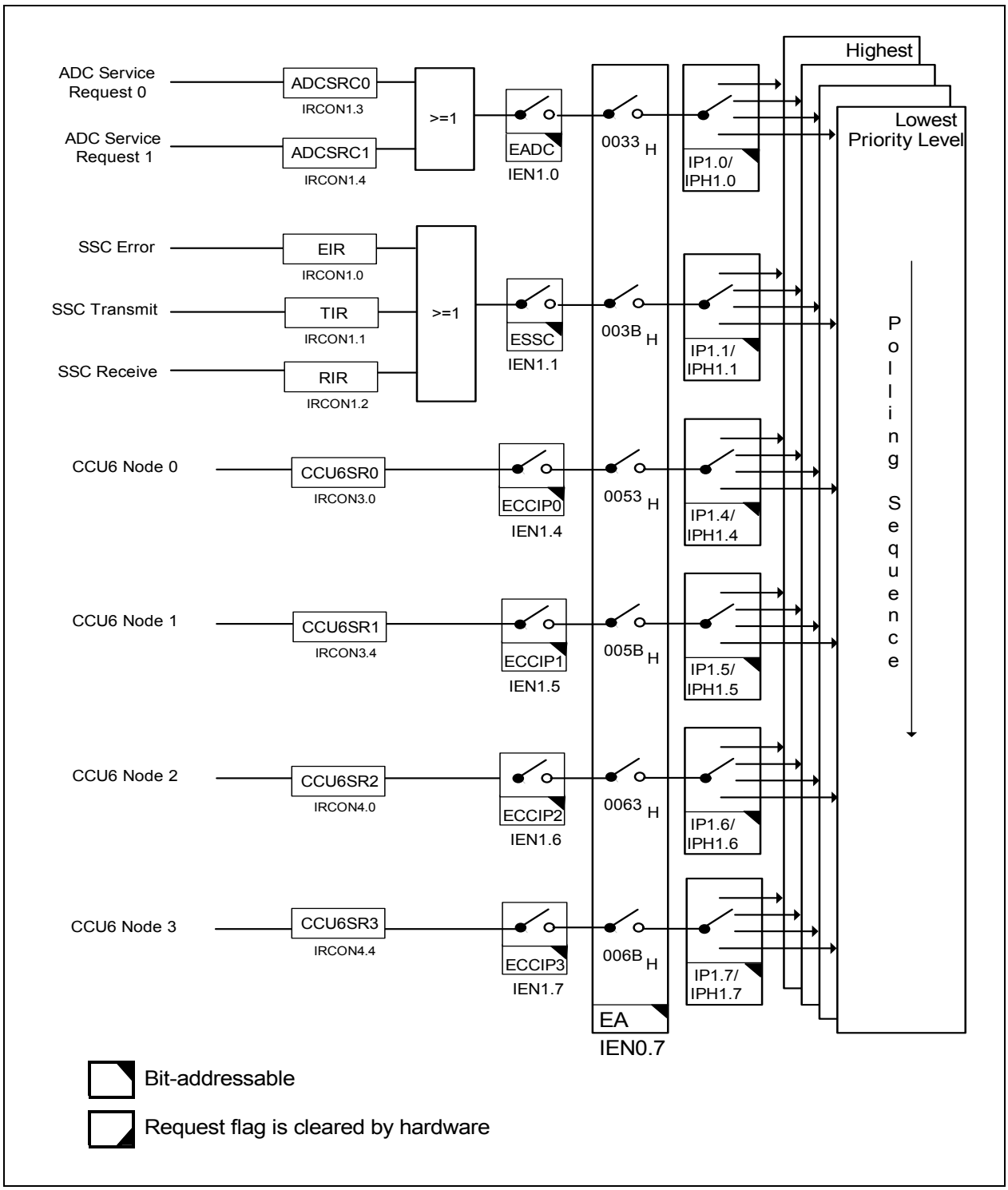


Figure 15 Interrupt Request Sources (Part 3)

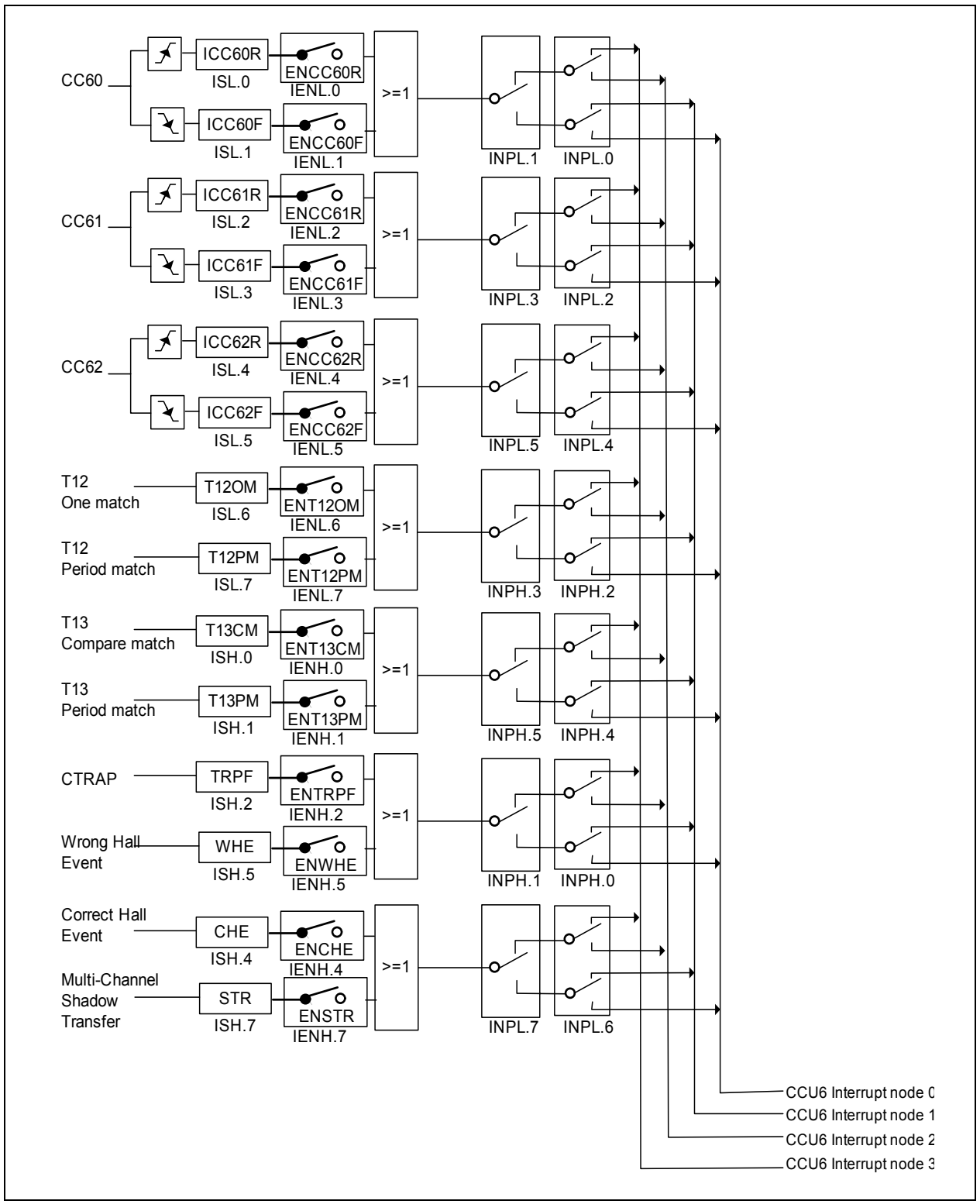


Figure 16 Interrupt Request Sources (Part 4)

3.4.2 Interrupt Source and Vector

Each interrupt source has an associated interrupt vector address. This vector is accessed to service the corresponding interrupt source request. The interrupt service of each interrupt source can be individually enabled or disabled via an enable bit. The assignment of the XC864 interrupt sources to the interrupt vector addresses and the corresponding interrupt source enable bits are summarized in **Table 18**.

Table 18 Interrupt Vector Addresses

Interrupt Source	Vector Address	Assignment for XC864	Enable Bit	SFR
NMI	0073_H	Watchdog Timer NMI	NMIWDT	NMICON
		PLL NMI	NMIPLL	
		Flash NMI	NMIFLASH	
		VDDC Prewarning NMI	NMIVDD	
		VDDP Prewarning NMI	NMIVDDP	
		Flash ECC NMI	NMIECC	
XINTR0	0003_H	External Interrupt 0	EX0	IEN0
XINTR1	$000B_H$	Timer 0	ET0	
XINTR2	0013_H	External Interrupt 1	EX1	
XINTR3	$001B_H$	Timer 1	ET1	
XINTR4	0023_H	UART	ES	
XINTR5	$002B_H$	T2	ET2	
		Fractional Divider (Normal Divider Overflow)		
		LIN		
XINTR6	0033_H	ADC	EADC	IEN1
XINTR7	$003B_H$	SSC	ESSC	
XINTR8	0043_H	External Interrupt 2	EX2	
XINTR9	$004B_H$	External Interrupt 3	EXM	
XINTR10	0053_H	CCU6 INP0	ECCIP0	
XINTR11	$005B_H$	CCU6 INP1	ECCIP1	
XINTR12	0063_H	CCU6 INP2	ECCIP2	
XINTR13	$006B_H$	CCU6 INP3	ECCIP3	

3.4.3 Interrupt Priority

Each interrupt source, except for NMI, can be individually programmed to one of the four possible priority levels. The NMI has the highest priority and supersedes all other interrupts. Two pairs of interrupt priority registers (IP and IPH, IP1 and IPH1) are available to program the priority level of each non-NMI interrupt vector.

A low-priority interrupt can be interrupted by a high-priority interrupt, but not by another interrupt of the same or lower priority. Further, an interrupt of the highest priority cannot be interrupted by any other interrupt source.

If two or more requests of different priority levels are received simultaneously, the request of the highest priority is serviced first. If requests of the same priority are received simultaneously, then an internal polling sequence determines which request is serviced first. Thus, within each priority level, there is a second priority structure determined by the polling sequence shown in **Table 19**.

Table 19 Priority Structure within Interrupt Level

Source	Level
Non-Maskable Interrupt (NMI)	(highest)
External Interrupt 0	1
Timer 0 Interrupt	2
External Interrupt 1	3
Timer 1 Interrupt	4
UART Interrupt	5
Timer 2, UART Normal Divider Overflow, LIN	6
ADC Interrupt	7
SSC Interrupt	8
External Interrupt 2	9
External Interrupt 3	10
CCU6 Interrupt Node Pointer 0	11
CCU6 Interrupt Node Pointer 1	12
CCU6 Interrupt Node Pointer 2	13
CCU6 Interrupt Node Pointer 3	14

3.5 Parallel Ports

The XC864 has 14 port pins organized into 4 parallel ports, Port 0 (P0) to Port 3 (P3). Each pin has a pair of internal pull-up and pull-down devices that can be individually enabled or disabled. Ports P0, P1 and P3 are bidirectional and can be used as general purpose input/output (GPIO) or to perform alternate input/output functions for the on-chip peripherals. When configured as an output, the open drain mode can be selected. Port P2 is an input-only port, providing general purpose input functions, alternate input functions for the on-chip peripherals, and also analog inputs for the Analog-to-Digital Converter (ADC).

*Note: P1.0 and P1.1 are bonded to the same package pin. See **Section 2.3** for the limitation of using these port pins.*

Bidirectional Port Features:

- Configurable pin direction
- Configurable pull-up/pull-down devices
- Configurable open drain mode
- Transfer of data through digital inputs and outputs (general purpose I/O)
- Alternate input/output for on-chip peripherals

Input Port Features:

- Configurable input driver
- Configurable pull-up/pull-down devices
- Receive of data through digital input (general purpose input)
- Alternate input for on-chip peripherals
- Analog input for ADC module

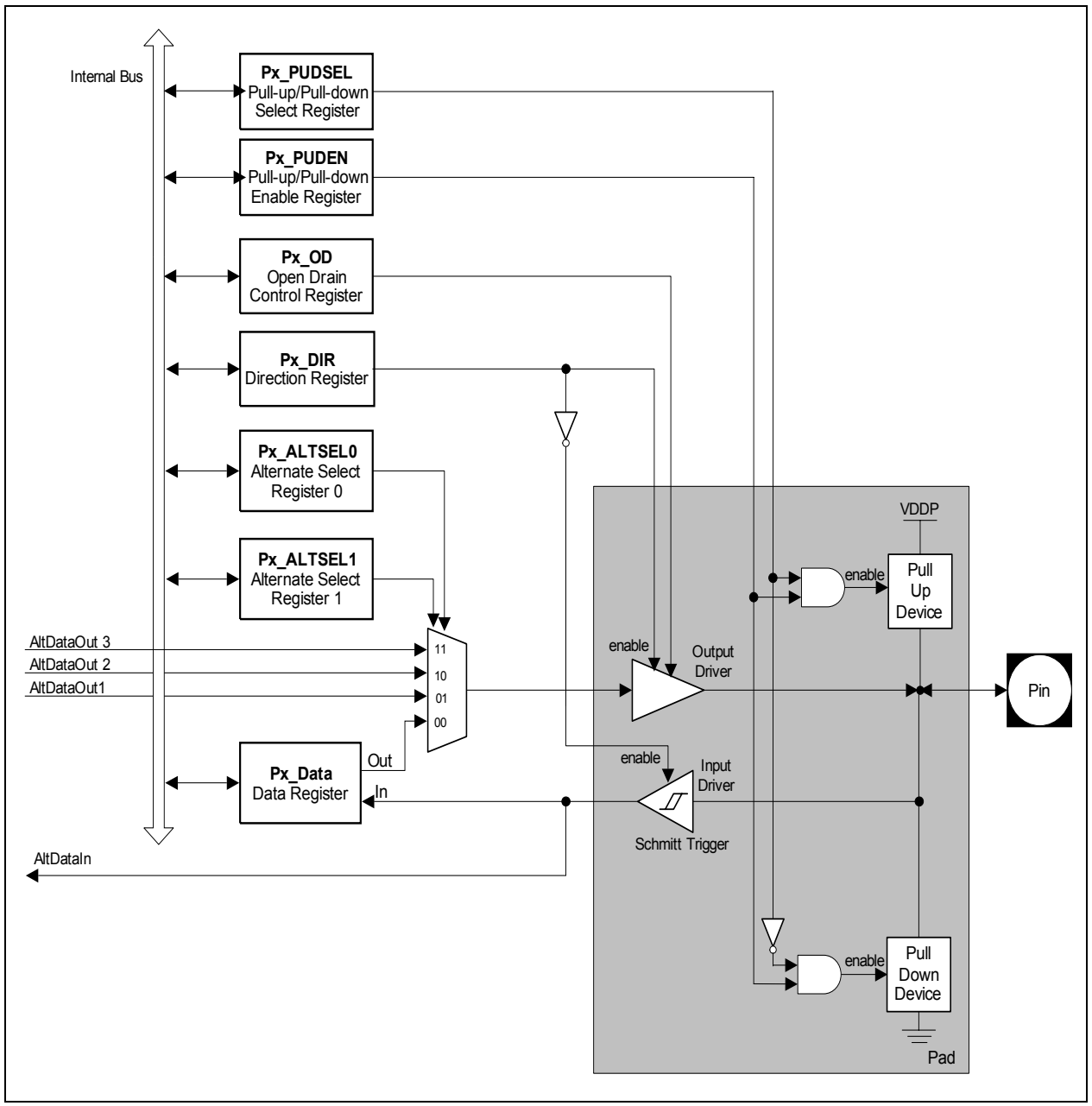


Figure 17 General Structure of Bidirectional Port

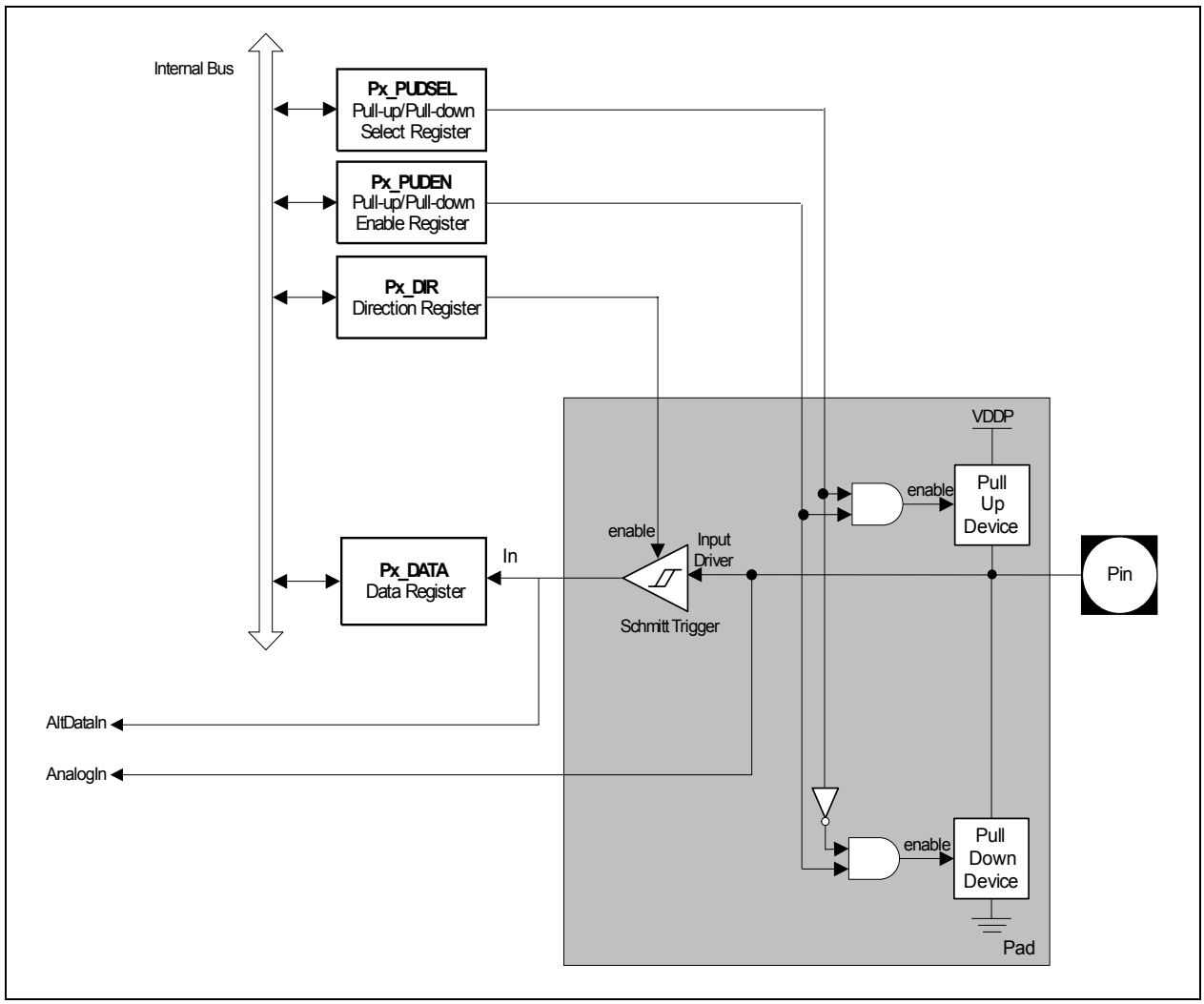


Figure 18 General Structure of Input Port

3.6 Power Supply System with Embedded Voltage Regulator

The XC864 microcontroller requires two different levels of power supply:

- 5.0 V for the Embedded Voltage Regulator (EVR) and Ports
- 2.5 V for the core, memory, on-chip oscillator, and peripherals

Figure 19 shows the XC864 power supply system. A power supply of 5.0 V must be provided from the external power supply pin. The 2.5 V power supply for the logic is generated by the EVR. The EVR helps to reduce the power consumption of the whole chip and the complexity of the application board design.

The EVR consists of a main voltage regulator and a low power voltage regulator. In active mode, both voltage regulators are enabled. In power-down mode, the main voltage regulator is switched off, while the low power voltage regulator continues to function and provide power supply to the system with low power consumption.

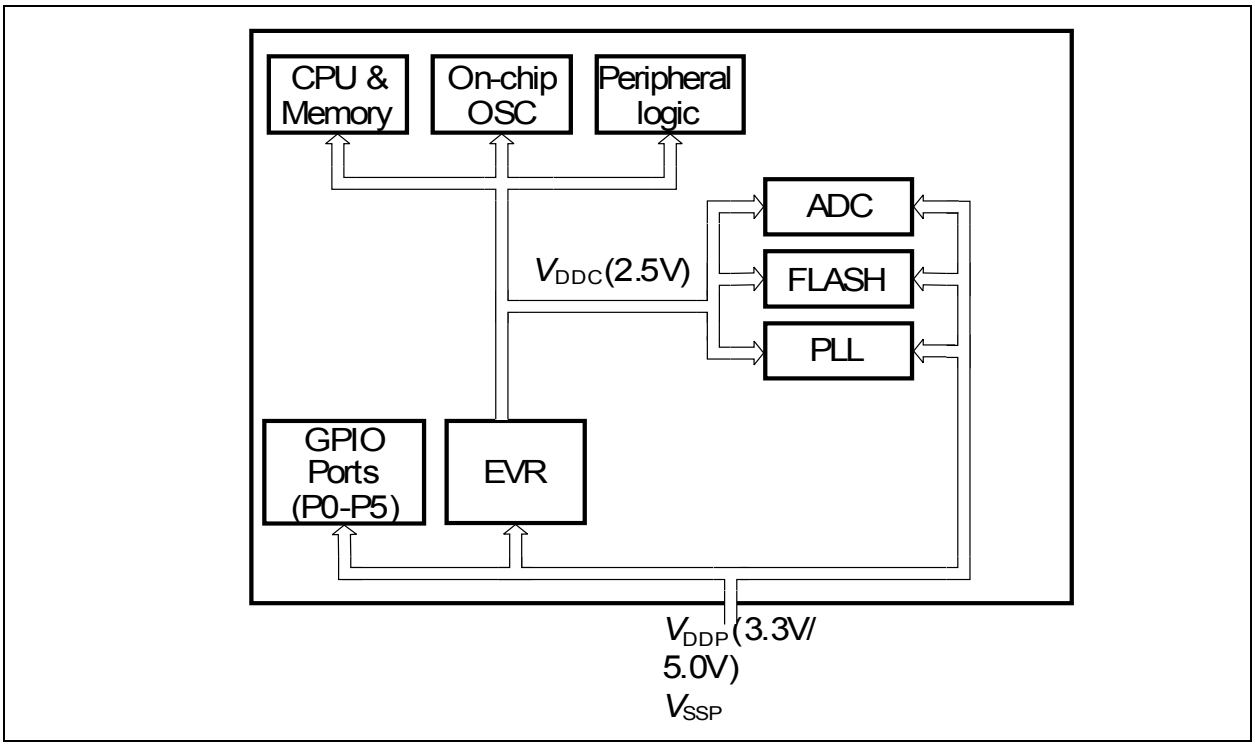


Figure 19 XC864 Power Supply System

EVR Features:

- Input voltage (V_{DDP}): 5.0 V
- Output voltage (V_{DDC}): 2.5 V ± 7.5%
- Low power voltage regulator provided in power-down mode
- V_{DDC} and V_{DDP} prewarning detection
- V_{DDC} brownout detection

3.7 Reset Control

The XC864 has five types of reset: power-on reset, hardware reset, watchdog timer reset, power-down wake-up reset, and brownout reset.

When the XC864 is first powered up, the status of certain pins (see **Table 21**) must be defined to ensure proper start operation of the device. At the end of a reset sequence, the sampled values are latched to select the desired boot option, which cannot be modified until the next power-on reset or hardware reset. This guarantees stable conditions during the normal operation of the device.

In order to power up the system properly, the external reset pin $\overline{\text{RESET}}$ must be asserted until V_{DDC} reaches 0.9*V_{DDC}. The delay of external reset can be realized by an external capacitor at $\overline{\text{RESET}}$ pin. This capacitor value must be selected so that V_{RESET} reaches 0.4 V, but not before V_{DDC} reaches 0.9* V_{DDC}.

A typical application example is shown in **Figure 20**. V_{DDP} capacitor value is 300 nF. V_{DDC} capacitor value is 220 nF. The capacitor connected to $\overline{\text{RESET}}$ pin is 100 nF.

Typically, the time taken for V_{DDC} to reach 0.9*V_{DDC} is less than 50 µs once V_{DDP} reaches 2.3V. Hence, based on the condition that 10% to 90% V_{DDP} (slew rate) is less than 500 µs, the $\overline{\text{RESET}}$ pin should be held low for 500 µs typically. See **Figure 21**.

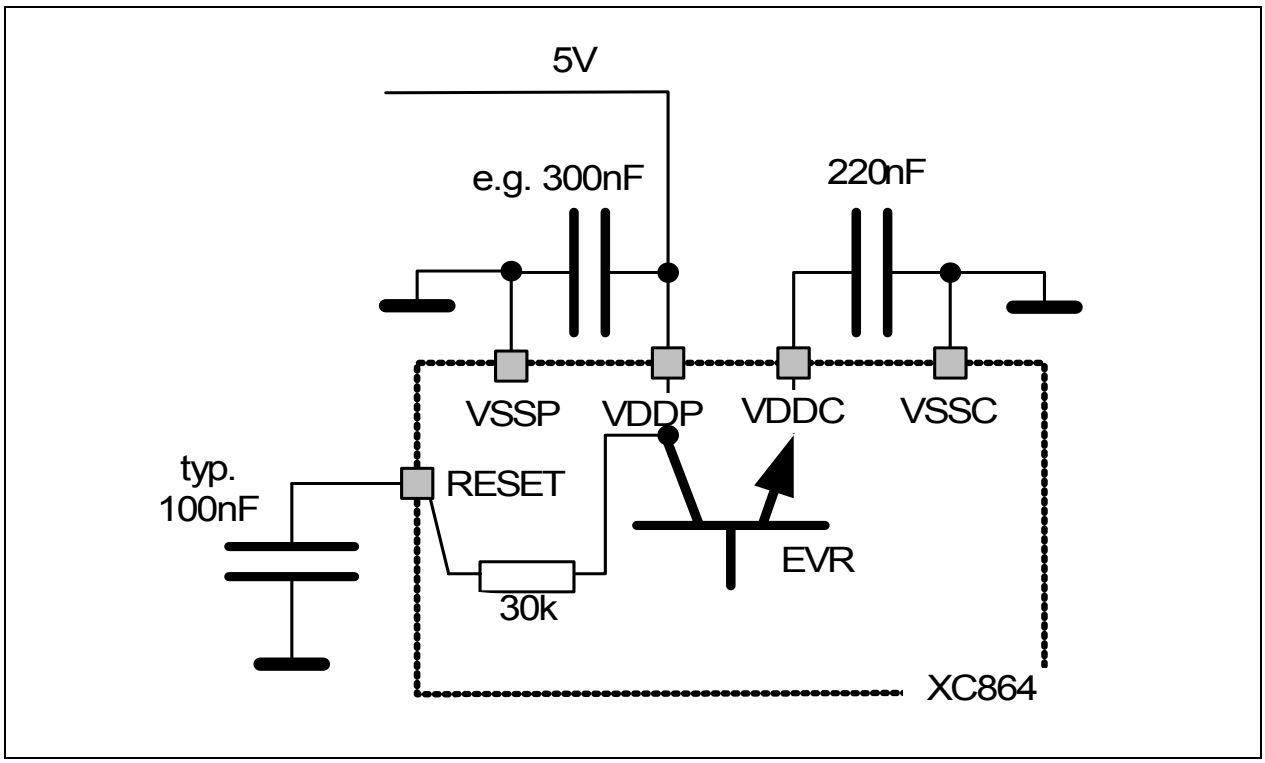


Figure 20 Reset Circuitry

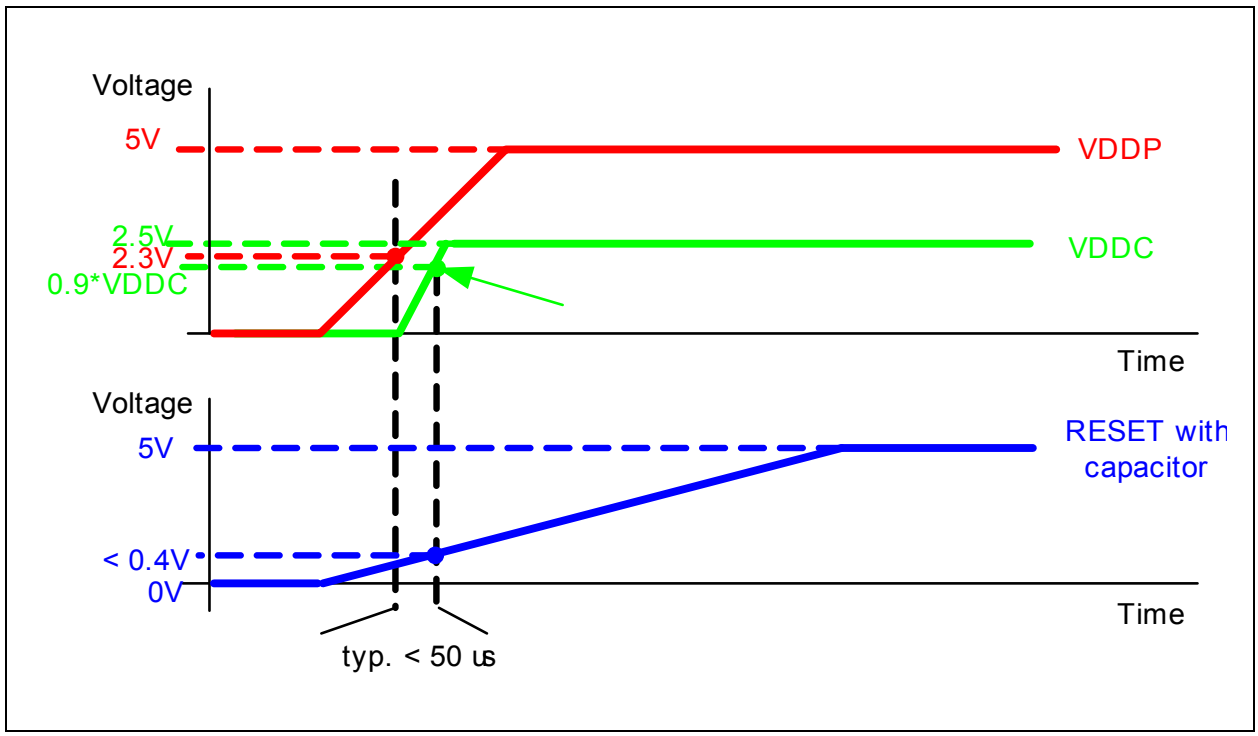


Figure 21 V_{DDP}, V_{DDC} and V_{RESET} during Power-on Reset

The second type of reset is the hardware reset. This reset function can be used during normal operation or when the chip is in power-down mode. A reset input pin $\overline{\text{RESET}}$ is provided for the hardware reset. To ensure the recognition of the hardware reset, pin $\overline{\text{RESET}}$ must be held low for at least 100 ns.

The Watchdog Timer (WDT) module is also capable of resetting the device if it detects a malfunction in the system.

Another type of reset that needs to be detected is a reset while the device is in power-down mode (wake-up reset). While the contents of the static RAM are undefined after a power-on reset, they are well defined after a wake-up reset from power-down mode.

3.7.1 Module Reset Behavior

Table 20 shows how the functions of the XC864 are affected by the various reset types. A "■" means that this function is reset to its default state.

Table 20 Effect of Reset on Device Functions

Module/ Function	Wake-Up Reset	Watchdog Reset	Hardware Reset	Power-On Reset	Brownout Reset
CPU Core	■	■	■	■	■
Peripherals	■	■	■	■	■
On-Chip Static RAM	Not affected, reliable	Not affected, reliable	Not affected, reliable	Affected, un-reliable	Affected, un-reliable
Oscillator, PLL	■	Not affected	■	■	■
Port Pins	■	■	■	■	■
EVR	The voltage regulator is switched on	Not affected	■	■	■
FLASH	■	■	■	■	■
NMI	Disabled	Disabled	■	■	■

3.7.2 Booting Scheme

When the XC864 is reset, it must identify the type of configuration with which to start the different modes once the reset sequence is complete. Thus, boot configuration information that is required for activation of special modes and conditions needs to be applied by the external world through input pins. After power-on reset or hardware reset, the pins TMS and P0.0 collectively select the different boot options. **Table 21** shows the available boot options in the XC864.

Note: The boot options are valid only with the default set of UART and JTAG pins.

Table 21 XC864 Boot Selection

TMS	P0.0	Type of Mode	PC Start Value
0	x	BSL Mode(User Mode)[1]; on-chip OSC/PLL non-bypassed	0000_H
1	0	OCDS Mode; on-chip OSC/PLL non-bypassed	0000_H

1) User Mode is enterd via BSL Mode depends on the user-parameter No_Activity_Count(NAC) and the Flash protection.

3.7.2.1 User Mode Entry in BSL Mode

In XC864, User Mode is entered through the BSL Mode. The entry also depends on the type of Flash protection[1] and the NAC (No_Activity_Count) values. NAC is a user defined parameter as described in each type of user mode entry.

There are three types of User Mode entry. Each entry was designed to be used under different situations.

User Mode Entry 1

- TMS = 0 during power-on reset or hardware reset
- Flash is not protected (PASSWORD[7:0][2] = 00_H)
- Flash address 0000_H is non-zero value
- NAC is valid

Once the chip is in BSL mode with Flash memory not protected and a non-zero at Flash address 0000_H, User Mode can be entered with or without delay depending on the NAC values. Delays are calculated based on the equation of [(NAC - 1) * 5 ms] where NAC value ranges from 01_H - $0C_H$. **Table 22** summarises different type of actions related to the NAC value. In order to ensure the validity of the NAC, the inverted values ($\overline{NAC}$) are needed to programmed togerther with the actual values.

1) Flash protection has to be taken and use with proper care as it will directly impact the usage of BSL mode and entry to User Mode. Refer to the 3 types of User Mode entry for detail descriptions.

2) Flash protection can be enabled or disabled by installing the user PASSWORD via BSL mode 6.

Table 22 Type of Actions related to the NAC value

NAC Value	Action
01_H	0 ms delay. Jump to User Mode immediately
02_H	5 ms delay before jumping to User Mode
03_H	10 ms delay before jumping to User Mode
04_H	15 ms delay before jumping to User Mode
05_H	20 ms delay before jumping to User Mode
06_H	25 ms delay before jumping to User Mode
07_H	30 ms delay before jumping to User Mode
08_H	35 ms delay before jumping to User Mode
09_H	40 ms delay before jumping to User Mode
$0A_H$	45 ms delay before jumping to User Mode
$0B_H$	50 ms delay before jumping to User Mode
$0C_H$	55 ms delay before jumping to User Mode
$0D_H$ - $0FF_H$, 00_H	Enter BSL Mode (Invalid NAC)

Once NAC and $\overline{NAC}$ is programmed within the valid range, entry to User Mode is always possible. If a LIN frame is received within the delay period(NAC = 02_H to $0C_H$), it will be processed as in the BSL mode and User mode will not be entered. Alternatively, user can erase the NAC values (and/or program an invalid NAC) to enter BSL mode. This can be done by having a Flash erase(/program) user-routine in the Flash memory.

User Mode Entry 2

- TMS = 0 during power-on reset or hardware reset
- Flash is protected (PASSWORD[0][1] = 1_B)
- NAC is valid (01_H - $0C_H$)

Once the chip is in BSL mode and Flash memory is protected with LSB of PASSWORD set to 1, User Mode can be entered with or without delay depending on the NAC values. The concept of using NAC as delays are similiar to **User Mode Entry 1** except for the definition of NAC parameter when flash is protected.

Once NAC is valid and programmed with the valid range, entry to User Mode is always possible. If a LIN frame is received within the delay period (NAC = 02_H to $0C_H$) as specified in **Table 22**, it will be processed as in the BSL mode and User mode will not be entered. Alternatively, user can erase the NAC value (and/or program an invalid NAC) to enter BSL mode. This can be done by having a user-routine in Flash to erase the

1) Flash protection can be enabled or disabled by installing the user PASSWORD via BSL mode 6.

existing NAC values and program an invalid NAC located in address($0FF8_H$) if flash protection mode 0(MSB of PASSWORD is 0) is selected. When Flash protetcion mode 1(MSB of PASSWORD = 1) is selected, the only way to enter BSL mode is to send a LIN frame within the delay period.

Note: Entering of BSL Mode is not possible if MSB of PASSWORD is 1 and NAC is 01_H.

User Mode Entry 3

- TMS = 0 during power-on reset or hardware reset
- Flash is protected (PASSWORD[0][1] = 0_B)

Once the chip is in BSL mode and Flash memory is protected with LSB of PASSWORD set to 0, User Mode will be entered immediately. Entering of BSL Mode is not possible in this type of User mode entry. Hence, changing of Flash code, XRAM code or flash protection scheme is not allowed. If there is an intention to upgrade Flash content, a pre-defined routine in the user code via In-Application Programming can be used. But it is possible only if flash protection mode 0(MSB of PASSWORD to 0) is selected. This option can be applied to all the user entry mode to change the flash content.

1) Flash protection can be enabled or disabled by installing the user PASSWORD via BSL mode 6.

3.8 Clock Generation Unit

The Clock Generation Unit (CGU) allows great flexibility in the clock generation for the XC864. The power consumption is indirectly proportional to the frequency, whereas the performance of the microcontroller is directly proportional to the frequency. During user program execution, the frequency can be programmed for an optimal ratio between performance and power consumption. Therefore the power consumption can be adapted to the actual application state.

Features:

- Phase-Locked Loop (PLL) for multiplying clock source by different factors
- PLL Base Mode
- Prescaler Mode
- PLL Mode
- Power-down mode support

The CGU consists of an oscillator circuit and a PLL.In the XC864, the oscillator is the on-chip oscillator (10 MHz). In addition, the PLL provides a fail-safe logic to perform oscillator run and loss-of-lock detection. This allows emergency routines to be executed for system recovery or to perform system shut down.

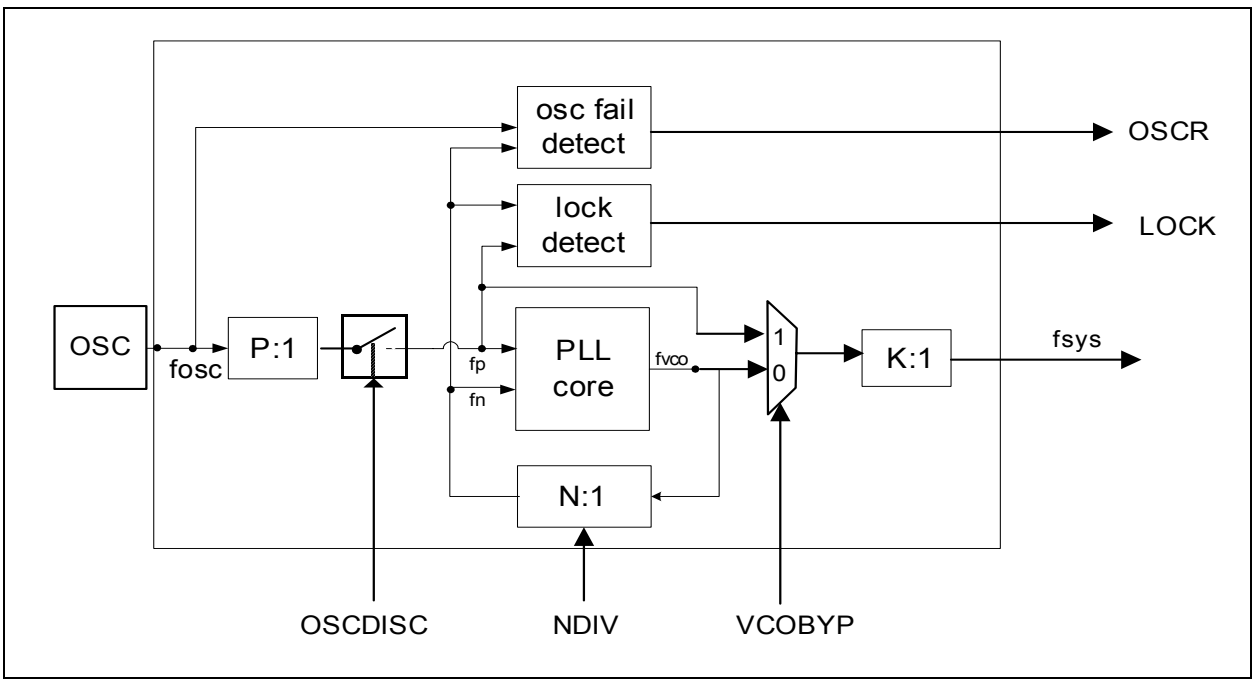


Figure 22 CGU Block Diagram

The clock system provides three ways to generate the system clock:

PLL Base Mode

The system clock is derived from the VCO base (free running) frequency clock divided by the K factor.

[1]

$$f_{SYS} = f_{VCObase} \times \frac{1}{K}$$

Prescaler Mode (VCO Bypass Operation)

In VCO bypass operation, the system clock is derived from the oscillator clock, divided by the P and K factors.

[2]

$$f_{SYS} = f_{OSC} \times \frac{1}{P \times K}$$

PLL Mode

The system clock is derived from the oscillator clock, multiplied by the N factor, and divided by the P and K factors. VCO bypass must be inactive for this PLL mode.

. [3]

$$f_{SYS} = f_{OSC} \times \frac{N}{P \times K}$$

Table 3-1 shows the settings of bits OSCDISC and VCOBYP for different clock mode selection.

Table 3-1 Clock Mode Selection

OSCDISC	VCOBYP	Clock Working Modes
0	0	PLL Mode
0	1	Prescaler Mode
1	0	PLL Base Mode
1	1	PLL Base Mode

Note: When oscillator clock is disconnected from PLL, the clock mode is PLL Base mode regardless of the setting of VCOBYP bit.

In normal running mode, the system works in the PLL mode.

For the XC864, the value of P and K are fixed to 1 and 2 respectively. In order to obtain the required f_{sys} at 80 MHz with a fixed oscillator frequency of 10 MHz, the N factor must be set to 16 by programming the NDIV bits to “0010”. In XC864, the output frequency needs to be at 80 MHz.

For f_{sys} = 80 MHz and K = 2, $f_{vco} = f_{sys}$ *2 = 160 MHz, VCOSEL bit in CMCON register must be set to 0 to select the VCO range of 150 MHz - 200 MHz.

3.8.1 Clock Management

The CGU generates all clock signals required within the microcontroller from a single clock, f_{sys}. During normal system operation, the typical frequencies of the different modules are as follow:

- CPU clock: CCLK, SCLK = 26.7 MHz
- CCU6 clock: FCLK = 26.7 MHz
- Other peripherals: PCLK = 26.7 MHz
- Flash Interface clock: CCLK3 = 80 MHz, CCLKn = 80 MHz and CCLK = 26.7 MHz

In addition, different clock frequency can output to pin CLKOUT(P0.0). The clock output frequency can further be divided by 2 using toggle latch (bit TLEN is set to 1), the resulting output frequency has 50% duty cycle. **Figure 23** shows the clock distribution of the XC864.

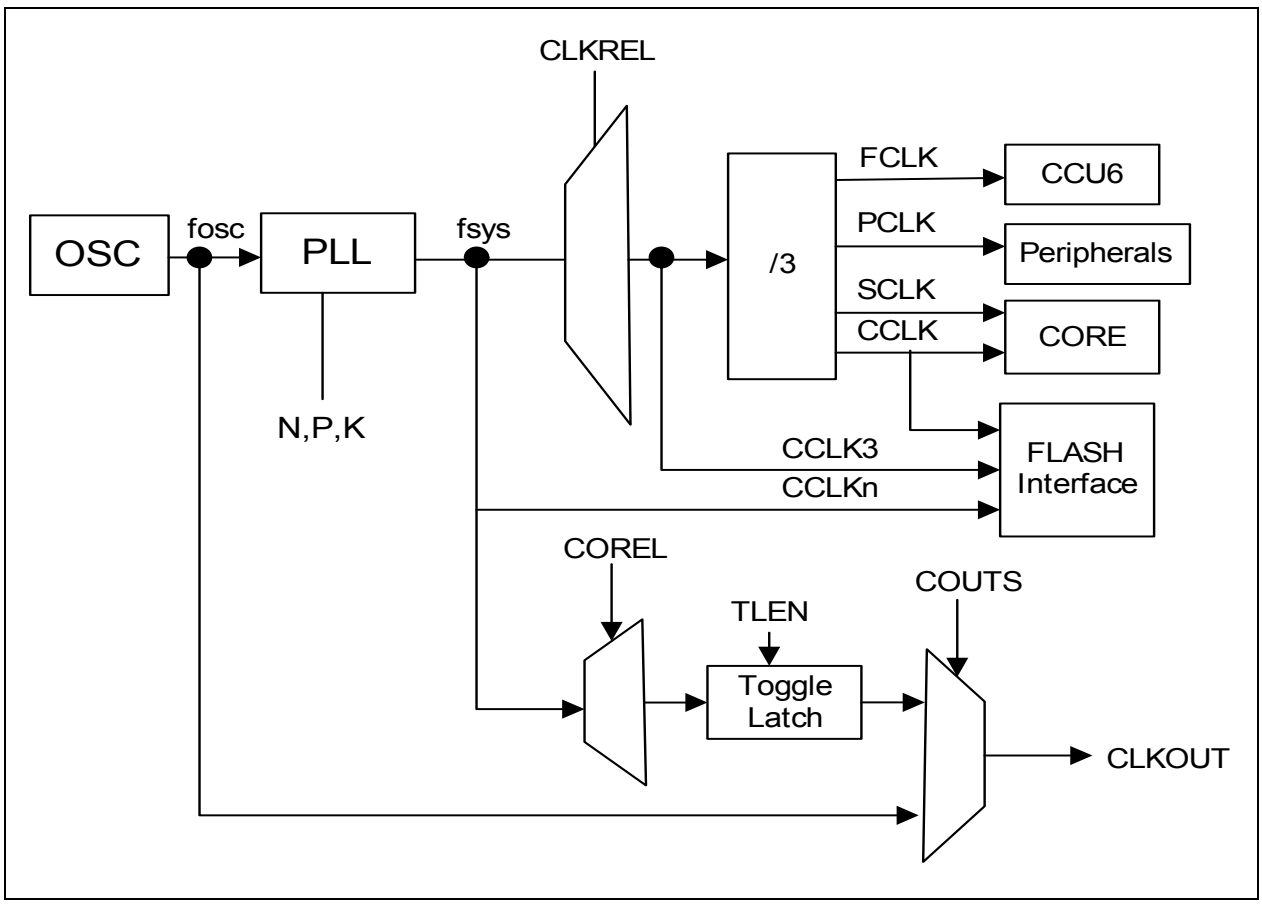


Figure 23 Clock Generation from f_{sys}

For power saving purposes, the clocks may be disabled or slowed down according to Table 23.

Table 23 System frequency (f_{sys} = 80 MHz)

Power Saving Mode	Action
Idle	Clock to the CPU is disabled.
Slow-down	Clocks to the CPU and all the peripherals, including CCU6, are divided by a common programmable factor defined by bit field CMCON.CLKREL.
Power-down	Oscillator and PLL are switched off.

Note: Flash programming and erasing can only be performed at f_{sys} = 80 MHz. However, Flash read access can be performed as long as $f_{sys} \leq 80$ MHz.

3.9 Power Saving Modes

The power saving modes of the XC864 provide flexible power consumption through a combination of techniques, including:

- Stopping the CPU clock
- Stopping the clocks of individual system components
- Reducing clock speed of some peripheral components
- Power-down of the entire system with fast restart capability

After a reset, the active mode (normal operating mode) is selected by default (see **Figure 24**) and the system runs in the main system clock frequency. From active mode, different power saving modes can be selected by software. They are:

- Idle mode
- Slow-down mode
- Power-down mode

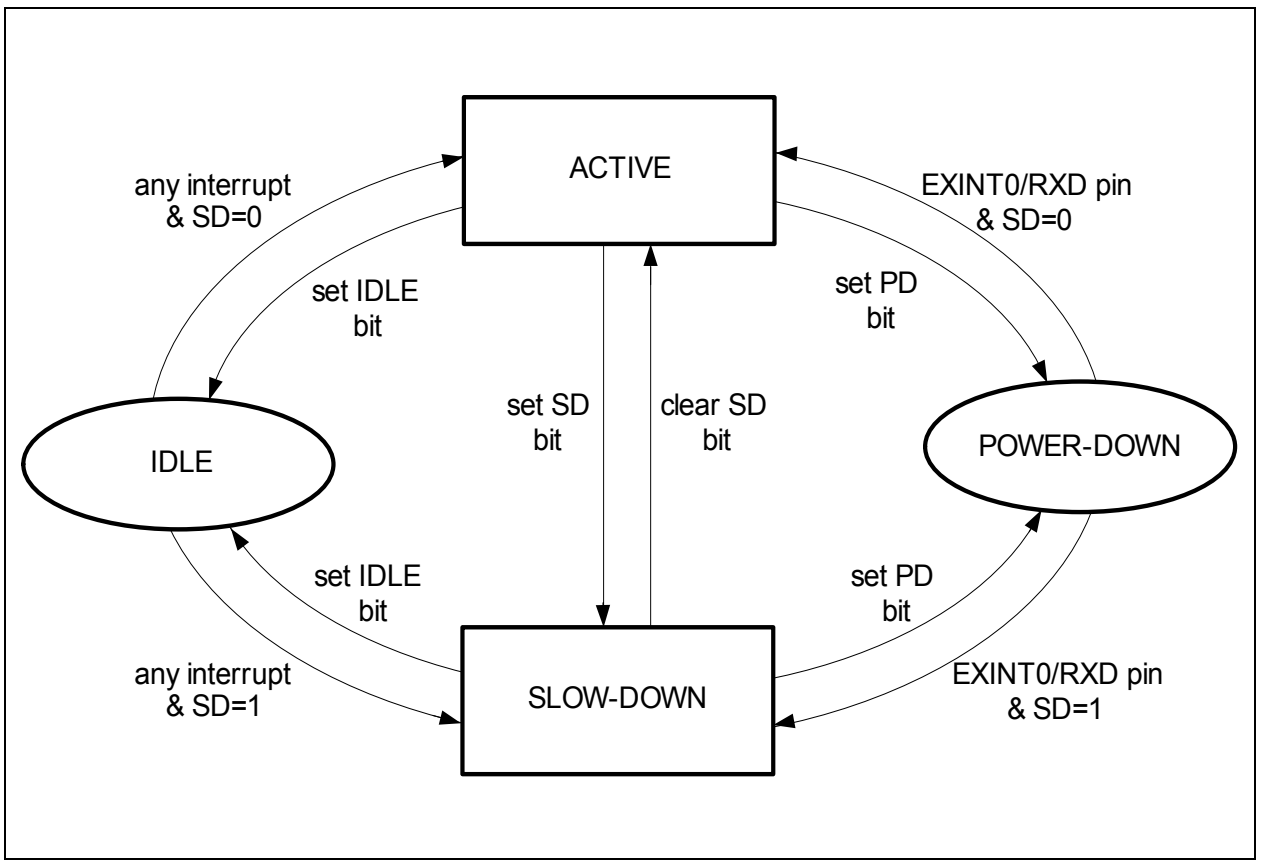


Figure 24 Transition between Power Saving Modes

3.10 Watchdog Timer

The Watchdog Timer (WDT) provides a highly reliable and secure way to detect and recover from software or hardware failures. The WDT is reset at a regular interval that is predefined by the user. The CPU must service the WDT within this interval to prevent the WDT from causing an XC864 system reset. Hence, routine service of the WDT confirms that the system is functioning properly. This ensures that an accidental malfunction of the XC864 will be aborted in a user-specified time period. In debug mode, the WDT is suspended and stops counting. Therefore, there is no need to refresh the WDT during debugging.

Features:

- 16-bit Watchdog Timer
- Programmable reload value for upper 8 bits of timer
- Programmable window boundary
- Selectable input frequency of $f_{PCLK}/2$ or $f_{PCLK}/128$
- Time-out detection with NMI generation and reset prewarning activation (after which a system reset will be performed)

The WDT is a 16-bit timer incremented by a count rate of $f_{PCLK}/2$ or $f_{PCLK}/128$. This 16-bit timer is realized as two concatenated 8-bit timers. The upper 8 bits of the WDT can be preset to a user-programmable value via a watchdog service access in order to modify the watchdog expire time period. The lower 8 bits are reset on each service access. **Figure 25** shows the block diagram of the WDT unit.

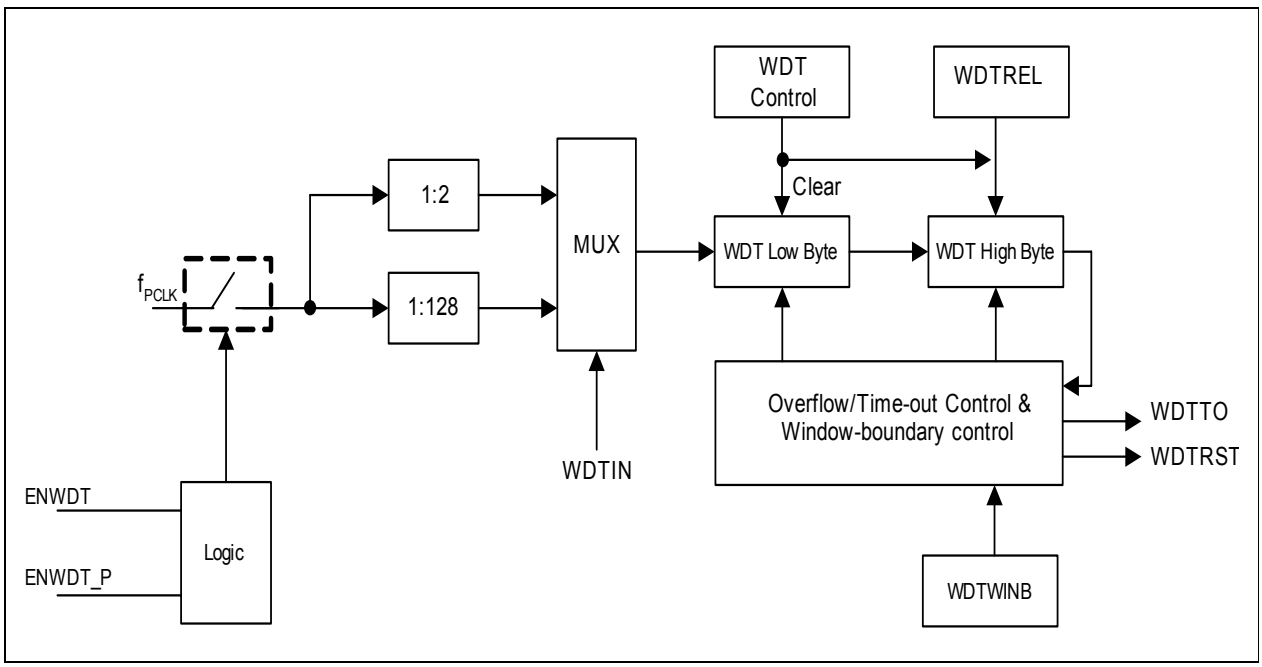


Figure 25 WDT Block Diagram

If the WDT is not serviced before the timer overflow, a system malfunction is assumed. As a result, the WDT NMI is triggered (assert WDTTO) and the reset prewarning is entered. The prewarning period lasts for 30_H count, after which the system is reset (assert WDTRST).

The WDT has a “programmable window boundary” which disallows any refresh during the WDT’s count-up. A refresh during this window boundary constitutes an invalid access to the WDT, causing the reset prewarning to be entered but without triggering the WDT NMI. The system will still be reset after the prewarning period is over. The window boundary is from 0000_H to the value obtained from the concatenation of WDTWINB and 00_H.

After being serviced, the WDT continues counting up from the value (<WDTREL> * 2^8). The time period for an overflow of the WDT is programmable in two ways:

- the input frequency to the WDT can be selected to be either $f_{PCLK}/2$ or $f_{PCLK}/128$
- the reload value WDTREL for the high byte of WDT can be programmed in register WDTREL

The period, P_{WDT}, between servicing the WDT and the next overflow can be determined by the following formula:

[4]

$$P_{WDT} = \frac{2^{(1 + WDTIN \times 6)} \times (2^{16} - WDTREL \times 2^8)}{f_{PCLK}}$$

If the Window-Boundary Refresh feature of the WDT is enabled, the period P_{WDT} between servicing the WDT and the next overflow is shortened if WDTWINB is greater than WDTREL, see **Figure 26**. This period can be calculated using the same formula by replacing WDTREL with WDTWINB. For this feature to be useful, WDTWINB should not be smaller than WDTREL.

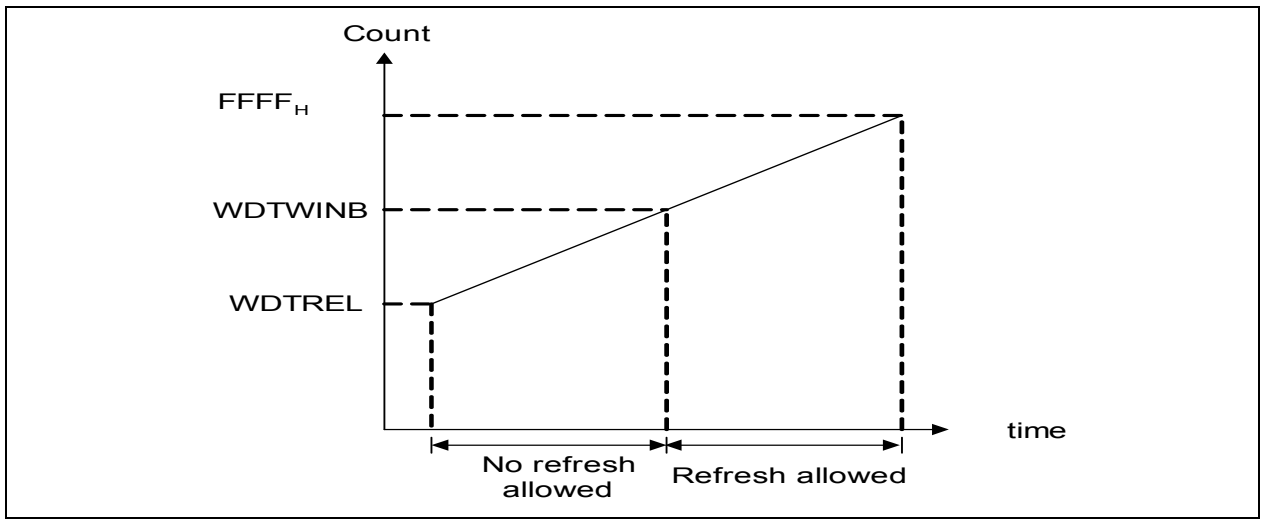


Figure 26 WDT Timing Diagram

Table 24 lists the possible watchdog time range that can be achieved for different module clock frequencies . Some numbers are rounded to 3 significant digits.

Table 24 Watchdog Time Ranges

Reload value in WDTREL	Prescaler for f_{PCLK}	
	2 (WDTIN = 0)	**128 (WDTIN = 1)**
	26.7 MHz	**26.7 MHz**
FF_H	19.2 µs	1.23 ms
$7F_H$	2.48 ms	159 ms
00_H	4.92 ms	315 ms

3.11 Universal Asynchronous Receiver/Transmitter

The Universal Asynchronous Receiver/Transmitter (UART) provides a full-duplex asynchronous receiver/transmitter, i.e., it can transmit and receive simultaneously. It is also receive-buffered, i.e., it can commence reception of a second byte before a previously received byte has been read from the receive register. However, if the first byte still has not been read by the time reception of the second byte is complete, one of the bytes will be lost.

Beside the standard dual pin configuration for UART, single pin communication is also available in XC864. It is supported by the primary UART pin.

Features:

- Full-duplex asynchronous modes
  - 8-bit or 9-bit data frames, LSB first
  - fixed or variable baud rate
- Receive buffered
- Multiprocessor communication
- Interrupt generation on the completion of a data transmission or reception

The UART can operate in four asynchronous modes as shown in **Table 25**. Data is transmitted on TXD and received on RXD.

Table 25 UART Modes

Operating Mode	Baud Rate
Mode 0: 8-bit shift register	$f_{PCLK}/2$
Mode 1: 8-bit shift UART	Variable
Mode 2: 9-bit shift UART	$f_{PCLK}/32$ or $f_{PCLK}/64$
Mode 3: 9-bit shift UART	Variable

There are several ways to generate the baud rate clock for the serial port, depending on the mode in which it is operating. In mode 0, the baud rate for the transfer is fixed at $f_{PCLK}/2$. In mode 2, the baud rate is generated internally based on the UART input clock and can be configured to either $f_{PCLK}/32$ or $f_{PCLK}/64$. The variable baud rate is set by either the underflow rate on the dedicated baud-rate generator, or by the overflow rate on Timer 1.

3.11.1 Baud-Rate Generator

The baud-rate generator is based on a programmable 8-bit reload value, and includes divider stages (i.e., prescaler and fractional divider) for generating a wide range of baud rates based on its input clock f_{PCLK}, see **Figure 27**.

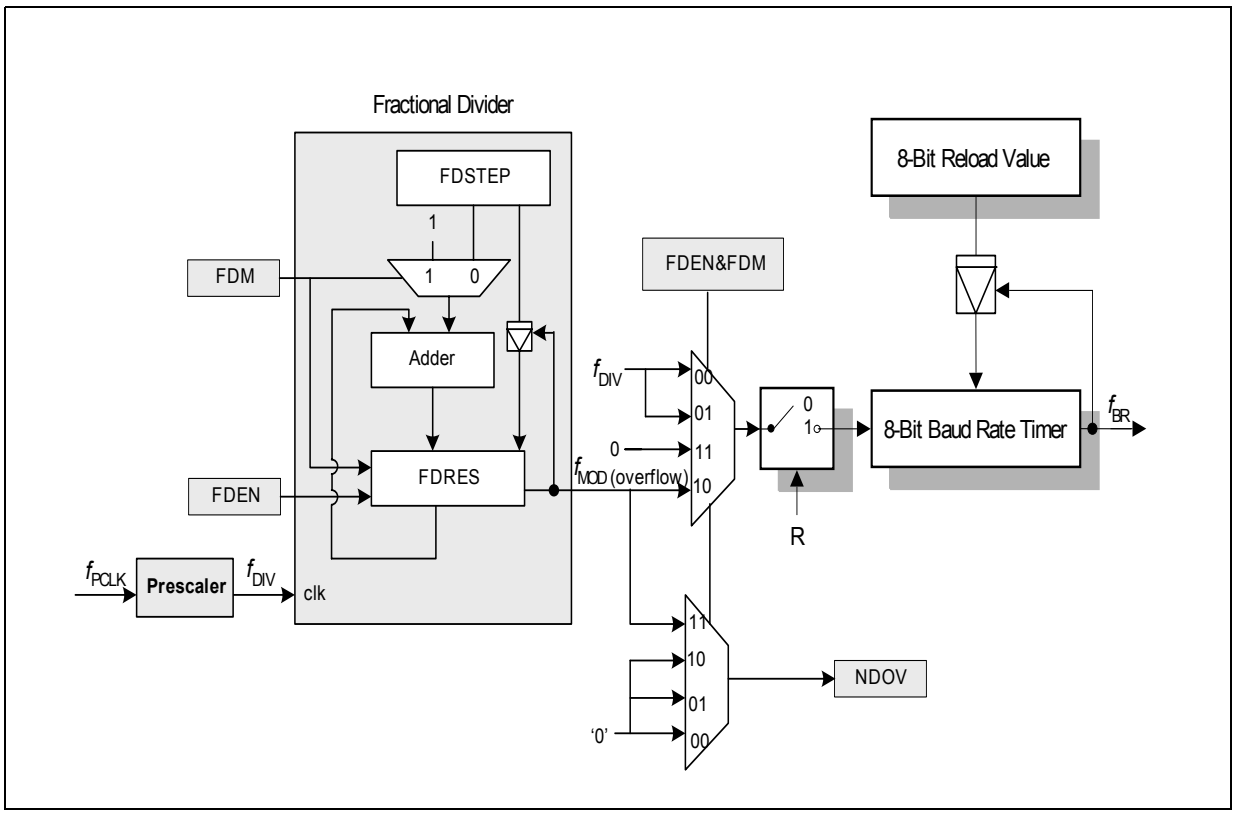


Figure 27 Baud-rate Generator Circuitry

The baud rate timer is a count-down timer and is clocked by either the output of the fractional divider (f_{MOD}) if the fractional divider is enabled (FDCON.FDEN = 1), or the output of the prescaler (f_{DIV}) if the fractional divider is disabled (FDEN = 0). For baud rate generation, the fractional divider must be configured to fractional divider mode (FDCON.FDM = 0). This allows the baud rate control run bit BCON.R to be used to start or stop the baud rate timer. At each timer underflow, the timer is reloaded with the 8-bit reload value in register BG and one clock pulse is generated for the serial channel.

Enabling the fractional divider in normal divider mode (FDEN = 1 and FDM = 1) stops the baud rate timer and nullifies the effect of bit BCON.R. See **Section 3.12**.

The baud rate (f_{BR}) value is dependent on the following parameters:

- Input clock f_{PCLK}
- Prescaling factor (2^{BRPRE}) defined by bit field BRPRE in register BCON
- Fractional divider (STEP/256) defined by register FDSTEP
  (to be considered only if fractional divider is enabled and operating in fractional divider mode)

- 8-bit reload value (BR_VALUE) for the baud rate timer defined by register BG

The following formulas calculate the final baud rate without and with the fractional divider respectively:

[5]

$$\text{baud rate} = \frac{f_{PCLK}}{16 \times 2^{BRPRE} \times (BR\_VALUE + 1)} \quad \text{where} \quad 2^{BRPRE} \times (BR\_VALUE + 1) > 1$$

[6]

$$\text{baud rate} = \frac{f_{PCLK}}{16 \times 2^{BRPRE} \times (BR\_VALUE + 1)} \times \frac{STEP}{256}$$

The maximum baud rate that can be generated is limited to $f_{PCLK}/32$. Hence, for a module clock of 26.7 MHz, the maximum achievable baud rate is 0.83 MBaud.

Standard LIN protocal can support a maximum baud rate of 20kHz, the baud rate accuracy is not critical and the fractional divider can be disabled. Only the prescaler is used for auto baud rate calculation. For LIN fast mode, which supports the baud rate of 20kHz to 115.2kHz, the higher baud rates require the use of the fractional divider for greater accuracy.

Table 26 lists the various commonly used baud rates with their corresponding parameter settings and deviation errors. The fractional divider is disabled and a module clock of 26.7 MHz is used.

Table 26 Typical Baud rates for UART with Fractional Divider disabled

Baud rate	Prescaling Factor (2^{BRPRE})	Reload Value (BR_VALUE + 1)	Deviation Error
19.2 kBaud	1 (BRPRE=000_B)	87 (57_H)	-0.22 %
9600 Baud	1 (BRPRE=000_B)	174 (AE_H)	-0.22 %
4800 Baud	2 (BRPRE=001_B)	174 (AE_H)	-0.22 %
2400 Baud	4 (BRPRE=010_B)	174 (AE_H)	-0.22 %

The fractional divider allows baud rates of higher accuracy (lower deviation error) to be generated. **Table 27** lists the resulting deviation errors from generating a baud rate of

115.2 kHz, using different module clock frequencies. The fractional divider is enabled (fractional divider mode) and the corresponding parameter settings are shown.

Table 27 Deviation Error for UART with Fractional Divider enabled

f_{PCLK}	Prescaling Factor (2^{BRPRE})	Reload Value (BR_VALUE + 1)	STEP	Deviation Error
26.67 MHz	1	10 (A_H)	177 ($B1_H$)	+0.03 %
13.33 MHz	1	7 (7_H)	248 ($F8_H$)	+0.11 %
6.67 MHz	1	3 (3_H)	212 ($D4_H$)	-0.16 %

3.11.2 Baud Rate Generation using Timer 1

In UART modes 1 and 3, Timer 1 can be used for generating the variable baud rates. In theory, this timer could be used in any of its modes. But in practice, it should be set into auto-reload mode (Timer 1 mode 2), with its high byte set to the appropriate value for the required baud rate. The baud rate is determined by the Timer 1 overflow rate and the value of SMOD as follows:

[7]

$$\text{Mode 1, 3 baud rate} = \frac{2^{SMOD} \times f_{PCLK}}{32 \times 2 \times (256 - TH1)}$$

3.12 Normal Divider Mode (8-bit Auto-reload Timer)

Setting bit FDM in register FDCON to 1 configures the fractional divider to normal divider mode, while at the same time disables baud rate generation (see **Figure 27**). Once the fractional divider is enabled (FDEN = 1), it functions as an 8-bit auto-reload timer (with no relation to baud rate generation) and counts up from the reload value with each input clock pulse. Bit field RESULT in register FDRES represents the timer value, while bit field STEP in register FDSTEP defines the reload value. At each timer overflow, an overflow flag (FDCON.NDOV) will be set and an interrupt request generated. This gives an output clock f_{MOD} that is 1/n of the input clock f_{DIV}, where n is defined by 256 - STEP.

The output frequency in normal divider mode is derived as follows:

[8]

$$f_{MOD} = f_{DIV} \times \frac{1}{256 - STEP}$$

3.13 LIN Protocol

The UART can be used to support the Local Interconnect Network (LIN) protocol for both master and slave operations. The LIN baud rate detection feature provides the capability to detect the baud rate within LIN protocol using Timer 2. This allows the UART to be synchronized to the LIN baud rate for data transmission and reception.

LIN is a holistic communication concept for local interconnected networks in vehicles. The communication is based on the SCI (UART) data format, a single-master/multiple-slave concept, a clock synchronization for nodes without stabilized time base. An attractive feature of LIN is self-synchronization of the slave nodes without a crystal or ceramic resonator, which significantly reduces the cost of hardware platform. Hence, the baud rate must be calculated and returned with every message frame.

The structure of a LIN frame is shown in **Figure 28**. The frame consists of the:

- header, which comprises a Break (13-bit time low), Synch Byte (55_H), and ID field
- response time
- data bytes (according to UART protocol)
- checksum

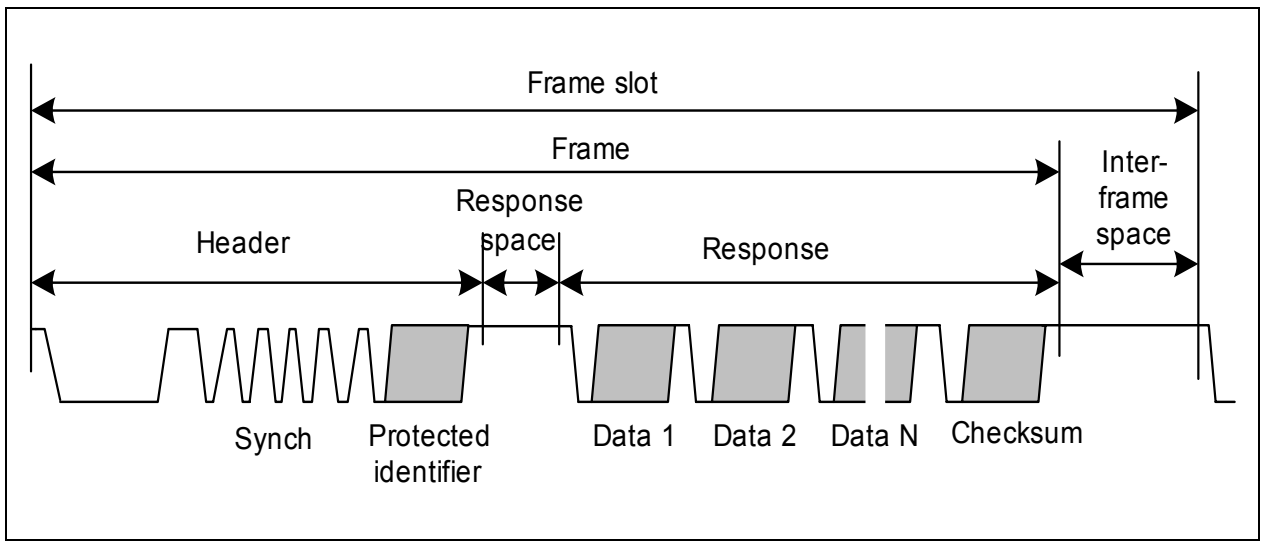


Figure 28 Structure of LIN Frame

3.13.1 LIN Header Transmission

LIN header transmission is only applicable in master mode. In the LIN communication, a master task decides when and which frame is to be transferred on the bus. It also identifies a slave task to provide the data transported by each frame. The information needed for the handshaking between the master and slave tasks is provided by the master task through the header portion of the frame.

The header consists of a break and synch pattern followed by an identifier. Among these three fields, only the break pattern cannot be transmitted as a normal 8-bit UART data.

The break must contain a dominant value of 13 bits or more to ensure proper synchronization of slave nodes.

In the LIN communication, a slave task is required to be synchronized at the beginning of the protected identifier field of frame. For this purpose, every frame starts with a sequence consisting of a break field followed by a synch byte field. This sequence is unique and provides enough information for any slave task to detect the beginning of a new frame and be synchronized at the start of the identifier field.

Upon entering LIN communication, a connection is established and the transfer speed (baud rate) of the serial communication partner (host) is automatically synchronized in the following steps:

STEP 1: Initialize interface for reception and timer for baud rate measurement

STEP 2: Wait for an incoming LIN frame from host

STEP 3: Synchronize the baud rate to the host

STEP 4: Enter for Master Request Frame or for Slave Response Frame

*Note: Re-synchronization and setup of baud rate are always done for **every** Master Request Header or Slave Response Header LIN frame.*

3.14 High-Speed Synchronous Serial Interface

The High-Speed Synchronous Serial Interface (SSC) supports full-duplex and half-duplex synchronous communication. The serial clock signal can be generated by the SSC internally (master mode), using its own 16-bit baud-rate generator, 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 or devices using other synchronous serial interfaces.

Features:

- Master and slave mode operation
  - Full-duplex or half-duplex operation
- Transmit and receive buffered
- Flexible data format
  - Programmable number of data bits: 2 to 8 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
- Variable baud rate
- Compatible with Serial Peripheral Interface (SPI)
- Interrupt generation
  - On a transmitter empty condition
  - On a receiver full condition
  - On an error condition (receive, phase, baud rate, transmit error)

Data is transmitted or received on lines TXD and RXD, which are normally connected to the 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 connected to the pin SCLK. Transmission and reception of data are double-buffered.

Figure 29 shows the block diagram of the SSC.

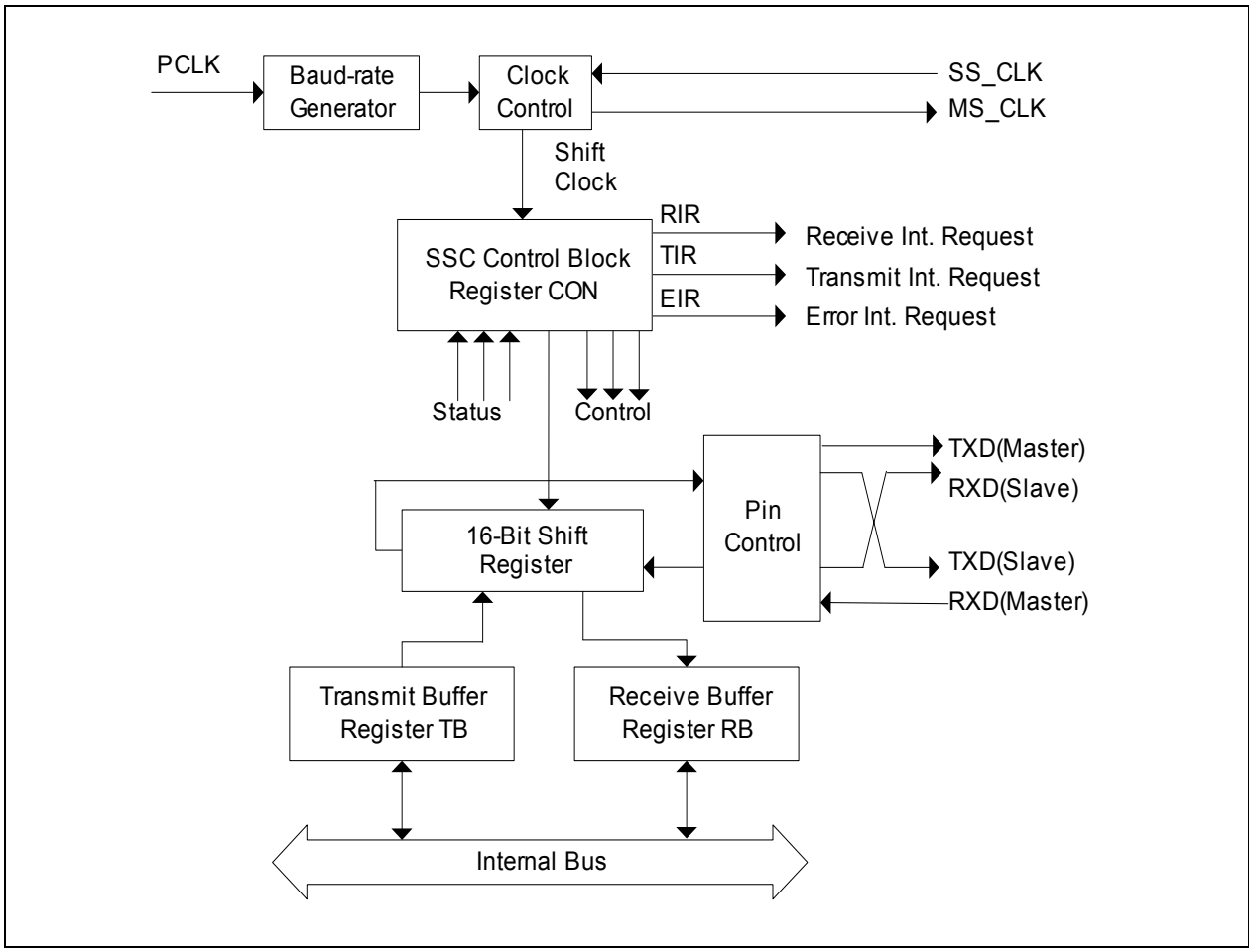


Figure 29 SSC Block Diagram

3.15 Timer 0 and Timer 1

Timers 0 and 1 are count-up timers which are incremented every machine cycle, or in terms of the input clock, every 2 PCLK cycles. Timer 0 can also be incremented in response to a 1-to-0 transition (falling edge) at the external input pin, T0.

Both timers are fully compatible and can be configured in four different operating modes for use in a variety of applications, see **Table 28**. In modes 0, 1 and 2, the two timers operate independently, but in mode 3, their functions are specialized.

Table 28 Timer 0 and Timer 1 Modes

Mode	Operation
0	**13-bit timer** The timer is essentially an 8-bit counter with a divide-by-32 prescaler. This mode is included solely for compatibility with Intel 8048 devices.
1	**16-bit timer** The timer registers, TLx and THx, are concatenated to form a 16-bit counter.
2	**8-bit timer with auto-reload** The timer register TLx is reloaded with a user-defined 8-bit value in THx upon overflow.
3	**Timer 0 operates as two 8-bit timers** The timer registers, TL0 and TH0, operate as two separate 8-bit counters. Timer 1 is halted and retains its count even if enabled.

3.16 Timer 2

Timer 2 is a 16-bit general purpose timer (THL2) that has two modes of operation, a 16-bit auto-reload mode and a 16-bit one channel capture mode. If the prescalar is disabled, Timer 2 counts with an input clock of PCLK/12. Timer 2 continues counting as long as it is enabled.

Table 29 Timer 2 Modes

Mode	Description
Auto-reload	**Up/Down Count Disabled** • Count up only • Start counting from 16-bit reload value, overflow at $FFFF_H$ • Reload event configurable for trigger by overflow condition only, or by negative/positive edge at input pin T2EX as well • Programmble reload value in register RC2 • Interrupt is generated with reload event
	Up/Down Count Enabled • Count up or down, direction determined by level at input pin T2EX • No interrupt is generated • Count up – Start counting from 16-bit reload value, overflow at $FFFF_H$ – Reload event triggered by overflow condition – Programmble reload value in register RC2 • Count down – Start counting from $FFFF_H$, underflow at value defined in register RC2 – Reload event triggered by underflow condition – Reload value fixed at $FFFF_H$
Channel capture	• Count up only • Start counting from 0000_H, overflow at $FFFF_H$ • Reload event triggered by overflow condition • Reload value fixed at 0000_H • Capture event triggered by falling/rising edge at pin T2EX • Captured timer value stored in register RC2 • Interrupt is generated with reload or capture event

3.17 Capture/Compare Unit 6

The Capture/Compare Unit 6 (CCU6) provides two independent timers (T12, T13), which can be used for Pulse Width Modulation (PWM) generation, especially for AC-motor control. The CCU6 also supports special control modes for block commutation and multi-phase machines.

The timer T12 can function in capture and/or compare mode for its three channels. The timer T13 can work in compare mode only.

The multi-channel control unit generates output patterns, which can be modulated by T12 and/or T13. The modulation sources can be selected and combined for the signal modulation.

Timer T12 Features:

- Three capture/compare channels, each channel can be used either as a capture or as a compare channel
- Supports generation of a three-phase PWM (six outputs, individual signals for highside and lowside switches)
- 16-bit resolution, maximum count frequency = peripheral clock frequency
- Dead-time control for each channel to avoid short-circuits in the power stage
- Concurrent update of the required T12/13 registers
- Generation of center-aligned and edge-aligned PWM
- Supports single-shot mode
- Supports many interrupt request sources
- Hysteresis-like control mode

Timer T13 Features:

- One independent compare channel with one output
- 16-bit resolution, maximum count frequency = peripheral clock frequency
- Can be synchronized to T12
- Interrupt generation at period-match and compare-match
- Supports single-shot mode

Additional Features:

- Implements block commutation for Brushless DC-drives
- Position detection via Hall-sensor pattern
- Automatic rotational speed measurement for block commutation
- Integrated error handling
- Fast emergency stop without CPU load via external signal ($\overline{\text{CTRAP}}$)
- Control modes for multi-channel AC-drives
- Output levels can be selected and adapted to the power stage

The block diagram of the CCU6 module is shown in **Figure 30**.

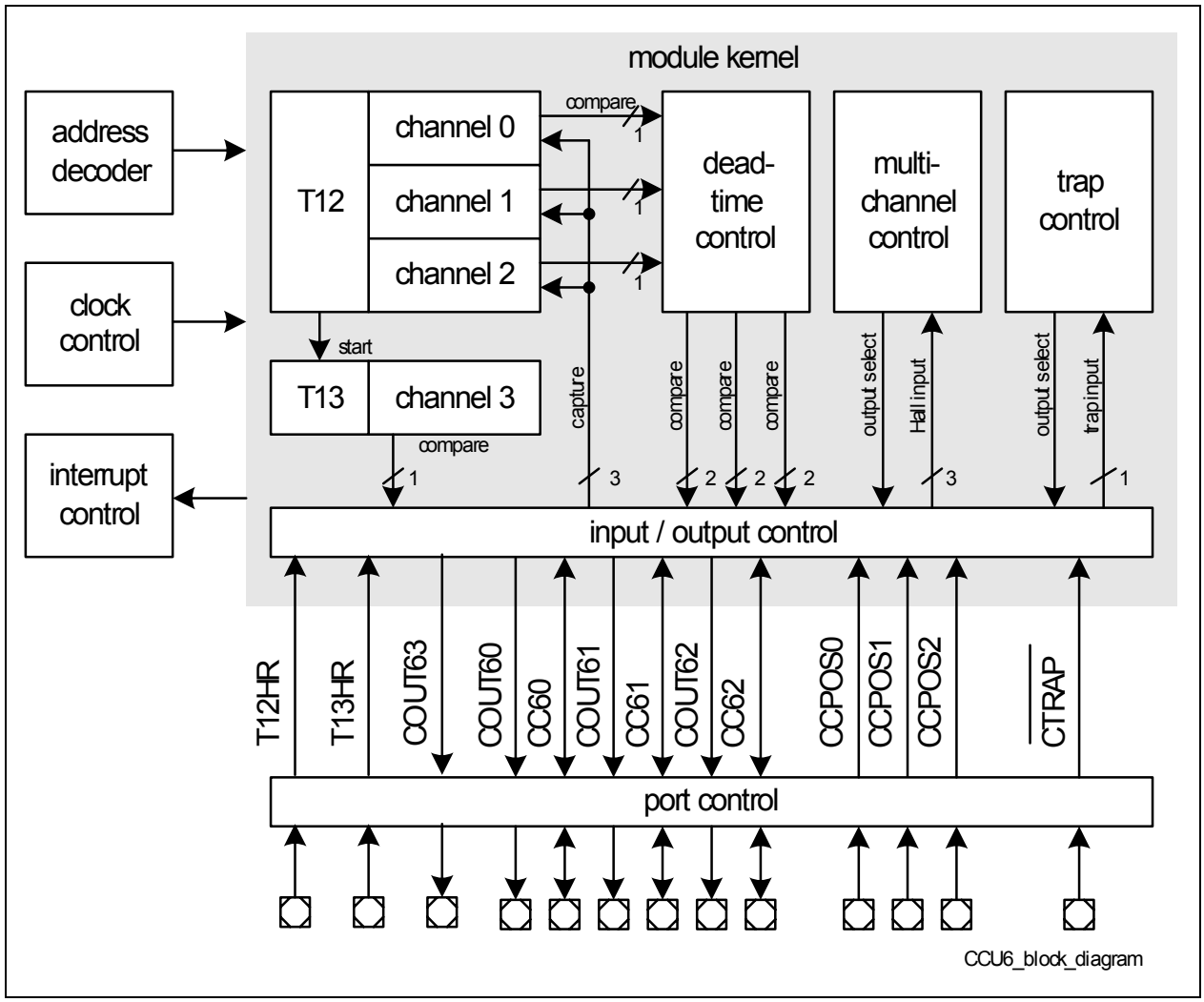


Figure 30 CCU6 Block Diagram

3.18 Analog-to-Digital Converter

The XC864 includes a high-performance 8-bit Analog-to-Digital Converter (ADC) with eight multiplexed analog input channels. The ADC uses a successive approximation technique to convert the analog voltage levels from up to eight different sources. The analog input channels of the ADC are available at Port 2.

Features:

- Successive approximation
- 8-bit resolution or 10-bit resolution
- Eight analog channels
- Four independent result registers
- Result data protection for slow CPU access (wait-for-read mode)
- Single conversion mode
- Autoscan functionality
- Limit checking for conversion results
- Data reduction filter (accumulation of up to 2 conversion results)
- Two independent conversion request sources with programmable priority
- Selectable conversion request trigger
- Flexible interrupt generation with configurable service nodes
- Programmable sample time
- Programmable clock divider
- Cancel/restart feature for running conversions
- Integrated sample and hold circuitry
- Compensation of offset errors
- Low power modes

3.18.1 ADC Clocking Scheme

A common module clock f_{ADC} generates the various clock signals used by the analog and digital parts of the ADC module:

- f_{ADCA} is input clock for the analog part.
- f_{ADCI} is internal clock for the analog part (defines the time base for conversion length and the sample time). This clock is generated internally in the analog part, based on the input clock f_{ADCA} to generate a correct duty cycle for the analog components.
- f_{ADCD} is input clock for the digital part.

The internal clock for the analog part f_{ADCI} is limited to a maximum frequency of 10 MHz. Therefore, the ADC clock prescaler must be programmed to a value that ensures f_{ADCI} does not exceed 10 MHz. The prescaler ratio is selected by bit field CTC in register GLOBCTR. A prescaling ratio of 32 can be selected when the maximum performance of the ADC is not required.

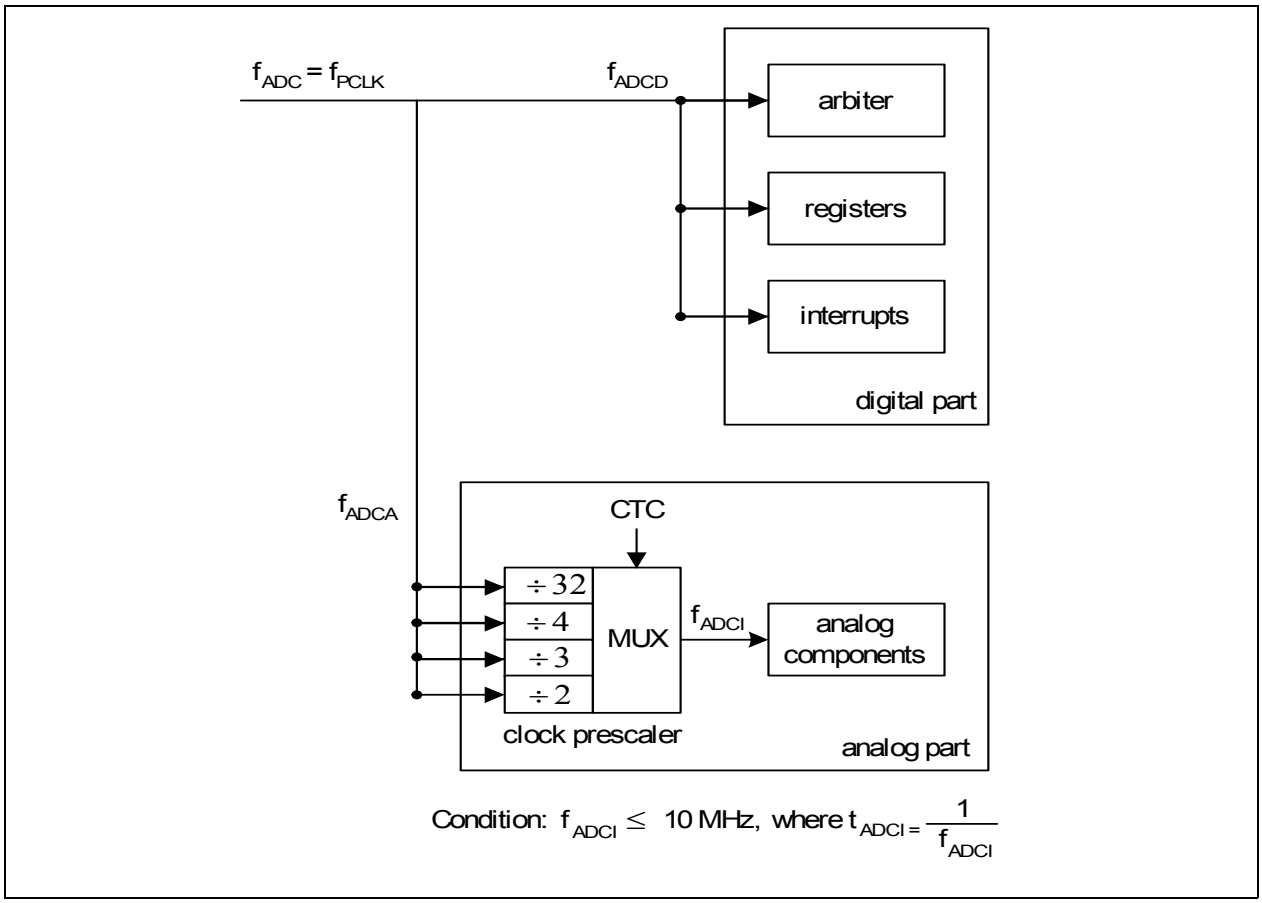


Figure 31 ADC Clocking Scheme

For module clock f_{ADC} = 26.7 MHz, the analog clock f_{ADCI} frequency can be selected as shown in **Table 30**.

Table 30 f_{ADCI} Frequency Selection

Module Clock f_{ADC}	CTC	Prescaling Ratio	Analog Clock f_{ADCI}
26.7 MHz	00_B	÷ 2	13.3 MHz (N.A)
	01_B	÷ 3	8.9 MHz
	10_B	÷ 4	6.7 MHz
	11_B (default)	÷ 32	833.3 kHz

As f_{ADCI} cannot exceed 10 MHz, bit field CTC should not be set to 00_B when f_{ADC} is 26.7 MHz. During slow-down mode where f_{ADC} may be reduced to 13.3 MHz, 6.7 MHz etc., CTC can be set to 00_B as long as the divided analog clock f_{ADCI} does not exceed 10 MHz. However, it is important to note that the conversion error could increase due to loss of charges on the capacitors, if f_{ADC} becomes too low during slow-down mode.

3.18.2 ADC Conversion Sequence

The analog-to-digital conversion procedure consists of the following phases:

- Synchronization phase (t_{SYN})
- Sample phase (t_S)
- Conversion phase
- Write result phase (t_{WR})

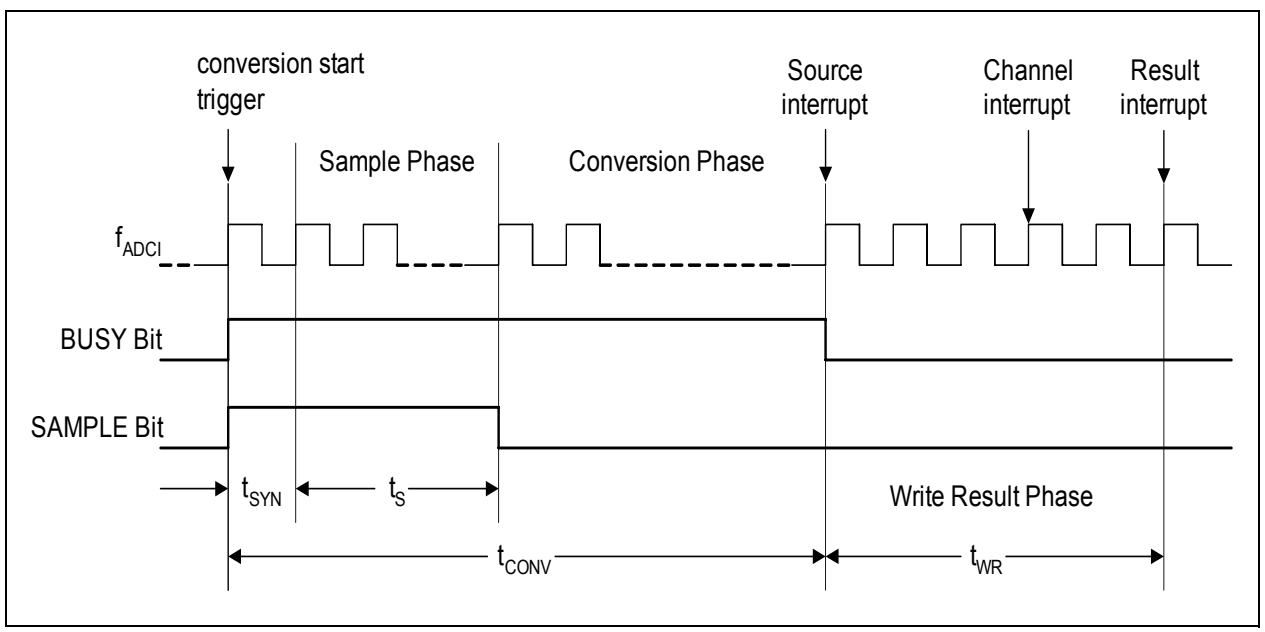


Figure 32 ADC Conversion Timing

3.19 On-Chip Debug Support

The On-Chip Debug Support (OCDS) provides the basic functionality required for the software development and debugging of XC800-based systems.

The OCDS design is based on these principles:

- use the built-in debug functionality of the XC800 Core
- add a minimum of hardware overhead
- provide support for most of the operations by a Monitor Program
- use standard interfaces to communicate with the Host (a Debugger)

Features:

- Set breakpoints on instruction address and within a specified address range
- Set breakpoints on internal RAM address
- Support unlimited software breakpoints in Flash/RAM code region
- Process external breaks
- Step through the program code

The OCDS functional blocks are shown in **Figure 33**. The Monitor Mode Control (MMC) block at the center of OCDS system brings together control signals and supports the overall functionality. The MMC communicates with the XC800 Core, primarily via the Debug Interface, and also receives reset and clock signals. After processing memory address and control signals from the core, the MMC provides proper access to the dedicated extra-memories: a Monitor ROM (holding the code) and a Monitor RAM (for work-data and Monitor-stack). The OCDS system is accessed through the JTAG[1)], which is an interface dedicated exclusively for testing and debugging activities and is not normally used in an application.

Note: All the debug functionality described here can normally be used only after XC864 has been started in OCDS mode.

1) The pins of the JTAG port can be assigned to either Port 0 (primary) or Ports 1 and 2 (secondary). User must set the JTAG pins (TCK and TDI) as input during connection with the OCDS system.

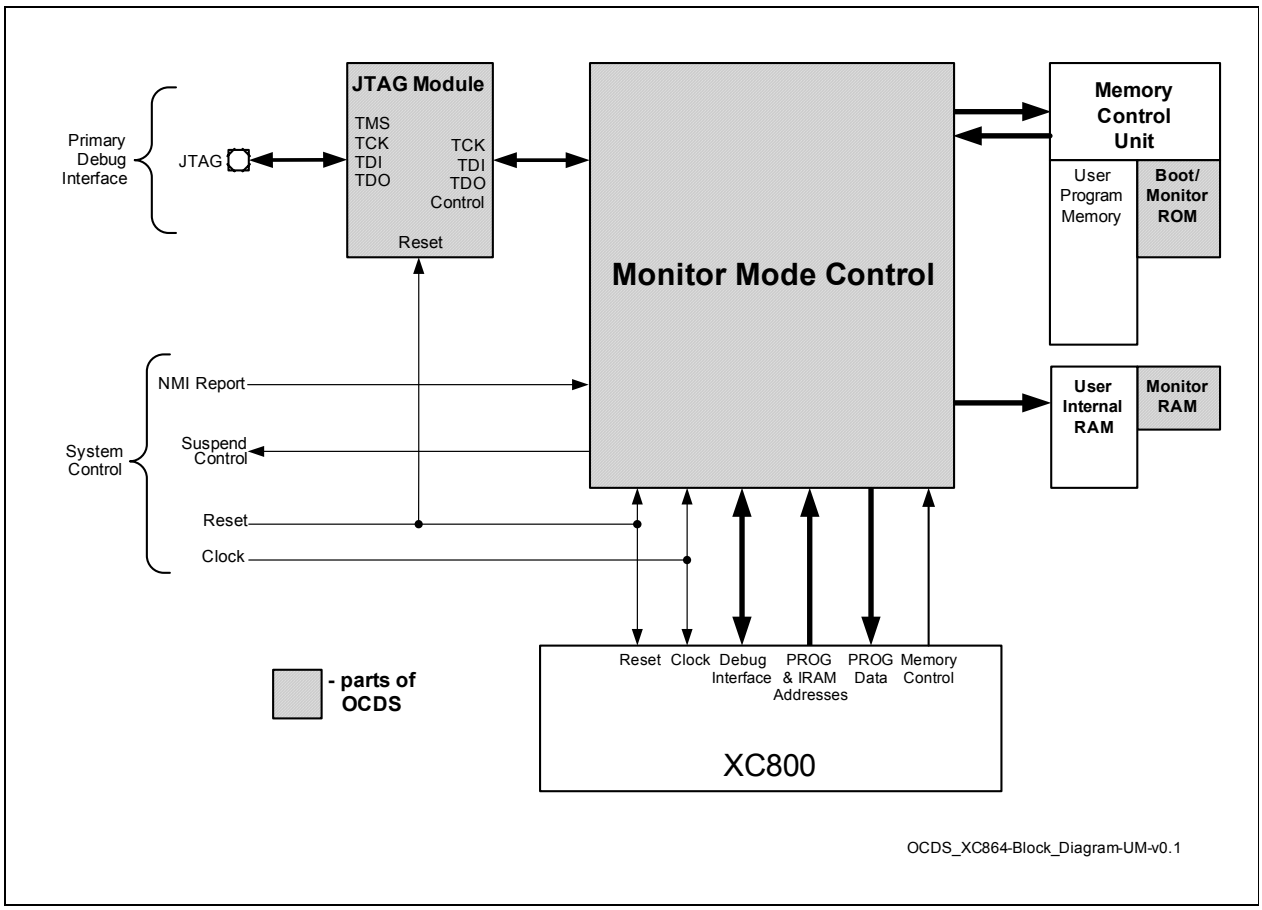


Figure 33 OCDS Block Diagram

3.19.1 NMI-mode priority over Debug-mode

While the core is in NMI-mode (after an NMI-request has been accepted and before the RETI instruction is executed, i.e. the time during a NMI-servicing routine), certain debug functions are blocked/restricted:

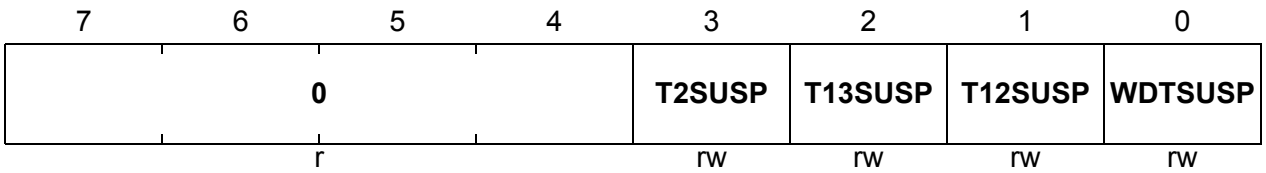
1. No external break is possible while the core is servicing an NMI.
   External break requested inside a NMI-servicing routine will be taken only after RETI is executed.
2. A breakpoint into NMI-servicing routine is taken, but single-step is not possible afterwards.
   If a step is requested, the servicing routine will run as coded and monitor mode will be invoked again only after a RETI is executed.

Hardware breakpoints and software breakpoints proceed as normal while CPU is in NMI-mode.

3.19.2 Debug-Suspend of Timers

During debugging (while in Monitor Mode) and the debug-suspend functionality is enabled (MMCR2.DSUSP = 1, default), timers in certain modules in XC864 can be suspended based on the settings of their corresponding module suspend bits in the register MODSUSP. When suspended, only the timer stops counting as the counter input clock is gated off. The module is still clocked so that module registers are accessible.

MODSUSP
Module Suspend Control Register **Reset Value: 01_H**



Field	Bits	Typ	Description
WDTSUSP	0	rw	**Watchdog Timer Debug Suspend Bit** 0 Watchdog Timer will not be suspended 1 Watchdog Timer will be suspended
T12SUSP	1	rw	**Timer 12 Debug Suspend Bit** 0 Timer 12 in Capture/Compare Unit will not be suspended 1 Timer 12 in Capture/Compare Unit will be suspended
T13SUSP	2	rw	**Timer 13 Debug Suspend Bit** 0 Timer 13 in Capture/Compare Unit will not be suspended 1 Timer 13 in Capture/Compare Unit will be suspended
T2SUSP	3	rw	**Timer 2 Debug Suspend Bit** 0 Timer 2 will not be suspended 1 Timer 2 will be suspended
0	[7:4]	r	**Reserved** Returns 0 if read; should be written with 0.

This feature could be quite useful, especially regarding the Watchdog Timer: it allows to prevent XC864 from unintentional WDT-resets while the user software is not executed and respectively - not able to service the Watchdog.

Also suspending the other timer-modules makes sense for debugging: once the application is not running, stopping counters helps for a more complete “freeze” of the device-status during a break.

It must be noted, in XC864 all of the debug suspend control bits other than that of WDT, have values 0 after reset, i.e. by default the module will not be suspended upon a break. But normally for debugging, the device will be started in OCDS mode and then the monitor will be invoked before to start any user code. Then it is possible using a debugger to configure suspend-controls as desired and only afterwards start the debug-session.

3.19.3 JTAG ID Register

This is a read-only register located inside the JTAG module, and is used to recognize the device(s) connected to the JTAG interface. Its content is shifted out when INSTRUCTION register contains the IDCODE command (opcode 04_H), and the same is also true immediately after reset.

The JTAG ID for XC864 is 1013 8083_H.

3.20 Chip Identification Number

The XC864 identity (ID) register is located at Page 1 of address $B3_H$. The value of ID register is $1B_H$. However, for easy identification of product variants, the Chip Identification Number, which is an unique number assigned to each product variant, is available. The differentiation is based on the product, variant type and device step information.

The Chip Identification Numbers associated with XC864 are $1B810C00_H$ for 5V device and $1B010C00_H$ for 3.3V device.

Two methods are provided to read a device's Chip Identification Number:

- In-application subroutine, GET_CHIP_INFO
- Bootstrap loader (BSL) mode A

4 Electrical Parameters

This chapter provides the characteristics of the electrical parameters which are implementation-specific for the XC864.

4.1 General Parameters

The general parameters are described here to aid the users in interpreting the parameters mainly in **Chapter 4.2** and **Chapter 4.3**.

4.1.1 Parameter Interpretation

The parameters listed in this section represent partly the characteristics of the XC864 and partly its requirements on the system. To aid interpreting the parameters easily when evaluating them for a design, they are indicated by the abbreviations in the “Symbol” column:

- **CC**
  These parameters indicate **C**ontroller **C**haracteristics, which are distinctive features of the XC864 and must be regarded for a system design.
- **SR**
  These parameters indicate **S**ystem **R**equirements, which must be provided by the microcontroller system in which the XC864 is designed in.

4.1.2 Absolute Maximum Rating

Maximum ratings are the extreme limits to which the XC864 can be subjected to without permanent damage.

Table 31 Absolute Maximum Rating Parameters

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Ambient temperature	T_A	-40	125	°C	under bias
Storage temperature	T_{ST}	-65	150	°C	
Junction temperature	T_J	-40	150	°C	under bias
Voltage on power supply pin with respect to V_{SS}	V_{DDP}	-0.5	6	V	
Voltage on core supply pin with respect to V_{SS}	V_{DDC}	-0.5	3.25	V	
Voltage on any pin with respect to V_{SS}	V_{IN}	-0.5	V_{DDP} + 0.5 or max. 6	V	Whatever is lower
Input current on any pin during overload condition	I_{IN}	-10	10	mA	
Absolute sum of all input currents during overload condition	$\Sigma\|I_{IN}\|$	–	50	mA	

Note: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. During absolute maximum rating overload conditions ($V_{IN} > V_{DDP}$ or $V_{IN} < V_{SS}$) the voltage on V_{DDP} pin with respect to ground (V_{SS}) must not exceed the values defined by the absolute maximum ratings.

4.1.3 Operating Conditions

The following operating conditions must not be exceeded in order to ensure correct operation of the XC864. All parameters mentioned in the following table refer to these operating conditions, unless otherwise noted.

Table 32 Operating Condition Parameters

Parameter	Symbol	Limit Values		Unit	Notes/ Conditions
		min.	max.		
Digital power supply voltage	V_{DDP}	4.5	5.5	V	
Digital power supply voltage	V_{DDP}	3.0	3.6	V	
Digital ground voltage	V_{SS}	0		V	
Digital core supply voltage	V_{DDC}	2.3	2.7	V	
System Clock Frequency[1)]	f_{SYS}	74	86	MHz	
Ambient temperature	T_A	-40	85	°C	SAF-XC864...
		-40	125	°C	SAK-XC864...

1) f_{SYS} is the PLL output clock. During normal operating mode, CPU clock is f_{SYS} / 3. Refer to **Figure 23** for details.

4.2 DC Parameters

4.2.1 Input/Output Characteristics

Table 33 Input/Output Characteristics (Operating Conditions apply)

Parameter	Symbol		Limit Values		Unit	Test Conditions
			min.	max.		
V_{DDP} = 5V Range						
Output low voltage	V_{OL}	CC	–	1.0	V	I_{OL} = 15 mA
			–	0.4	V	I_{OL} = 5 mA
Output high voltage	V_{OH}	CC	V_{DDP} - 1.0	–	V	I_{OH} = -15 mA
			V_{DDP} - 0.4	–	V	I_{OH} = -5 mA
Input low voltage on port pins (all except P0.0 & P0.1)	V_{ILP}	SR	–	0.3 × V_{DDP}	V	CMOS Mode
Input low voltage on P0.0 & P0.1	V_{ILP0}	SR	-0.2	0.3 × V_{DDP}	V	CMOS Mode
Input low voltage on RESET pin	V_{ILR}	SR	–	0.3 × V_{DDP}	V	CMOS Mode
Input low voltage on TMS pin	V_{ILT}	SR	–	0.3 × V_{DDP}	V	CMOS Mode
Input high voltage on port pins (all except P0.0 & P0.1)	V_{IHP}	SR	0.7 × V_{DDP}	–	V	CMOS Mode
Input high voltage on P0.0 & P0.1	V_{IHP0}	SR	0.7 × V_{DDP}	V_{DDP}	V	CMOS Mode
Input high voltage on RESET pin	V_{IHR}	SR	0.7 × V_{DDP}	–	V	CMOS Mode
Input high voltage on TMS pin	V_{IHT}	SR	0.7 × V_{DDP}	–	V	CMOS Mode
Input Hysteresis[1)]	HYS	CC	0.08 × V_{DDP}	–	V	CMOS Mode
Pull-up current[2)]	I_{PU}	SR	–	-10	µA	$V_{IH,min}$
			-150	–	µA	$V_{IL,max}$

Table 33 Input/Output Characteristics (Operating Conditions apply)

Parameter	Symbol		Limit Values		Unit	Test Conditions
			min.	max.		
Pull-down current[2]	I_{PD}	SR	–	10	µA	$V_{IL,max}$
			150	–	µA	$V_{IH,min}$
Input leakage current[3]	I_{OZ1}	CC	-2.5	1	µA	$0 < V_{IN} < V_{DDP}$, $T_A \leq 125°C$
Overload current on any pin	I_{OV}	SR	-5	5	mA	
Absolute sum of overload currents	$\Sigma\lvert I_{OV}\rvert$	SR	–	25	mA	[4]
Voltage on any pin during V_{DDP} power off	V_{PO}	SR	–	0.3	V	[5]
Maximum current per pin (excluding V_{DDP} and V_{SS})	I_M	SR	–	15	mA	
Maximum current for all pins (excluding V_{DDP} and V_{SS})	$\Sigma\lvert I_M\rvert$	SR	–	60	mA	
Maximum current into V_{DDP}	I_{MVDDP}	SR	–	80	mA	
Maximum current out of V_{SS}	I_{MVSS}	SR	–	80	mA	
V_{DDP} = 3.3V Range						
Output low voltage	V_{OL}	CC	–	1.0	V	I_{OL} = 8 mA
			–	0.4	V	I_{OL} = 2.5 mA
Output high voltage	V_{OH}	CC	V_{DDP} - 1.0	–	V	I_{OH} = -8 mA
			V_{DDP} - 0.4	–	V	I_{OH} = -2.5 mA
Input low voltage on port pins (all except P0.0 & P0.1)	V_{ILP}	SR	–	0.3 × V_{DDP}	V	CMOS Mode
Input low voltage on P0.0 & P0.1	V_{ILP0}	SR	-0.2	0.3 × V_{DDP}	V	CMOS Mode

Table 33 Input/Output Characteristics (Operating Conditions apply)

Parameter	Symbol		Limit Values		Unit	Test Conditions
			min.	max.		
Input low voltage on RESET pin	V_{ILR}	SR	–	0.3 × V_{DDP}	V	CMOS Mode
Input low voltage on TMS pin	V_{ILT}	SR	–	0.3 × V_{DDP}	V	CMOS Mode
Input high voltage on port pins (all except P0.0 & P0.1)	V_{IHP}	SR	0.7 × $\boldsymbol{V_{DDP}}$	–	V	CMOS Mode
Input high voltage on P0.0 & P0.1	V_{IHP0}	SR	0.7 × V_{DDP}	V_{DDP}	V	CMOS Mode
Input high voltage on RESET pin	V_{IHR}	SR	0.7 × V_{DDP}	–	V	CMOS Mode
Input high voltage on TMS pin	V_{IHT}	SR	0.75 × V_{DDP}	–	V	CMOS Mode
Input Hysteresis[1]	HYS	CC	0.03 × $\boldsymbol{V_{DDP}}$	–	V	CMOS Mode
Pull-up current[2]	I_{PU}	SR	–	-5	µA	$V_{IH,min}$
			-50	–	µA	$V_{IL,max}$
Pull-down current[2]	I_{PD}	SR	–	5	µA	$V_{IL,max}$
			50	–	µA	$V_{IH,min}$
Input leakage current[3]	I_{OZ1}	CC	-2.5	1	µA	$0 < V_{IN} < V_{DDP}$, $T_A \leq 125°C$
Overload current on any pin	I_{OV}	SR	-5	5	mA	
Absolute sum of overload currents	$\Sigma\lvert I_{OV}\rvert$	SR	–	25	mA	[4]
Voltage on any pin during V_{DDP} power off	V_{PO}	SR	–	0.3	V	[5]
Maximum current per pin (excluding V_{DDP} and V_{SS})	I_M	SR	–	15	mA	
Maximum current for all pins (excluding V_{DDP} and V_{SS})	$\Sigma\lvert I_M\rvert$	SR	–	60	mA	

Table 33 Input/Output Characteristics (Operating Conditions apply)

Parameter	Symbol	Limit Values		Unit	Test Conditions
		min.	max.		
Maximum current into V_{DDP}	I_{MVDDP} SR	–	80	mA	
Maximum current out of V_{SS}	I_{MVSS} SR	–	80	mA	

1) Not subjected to production test, verified by design/characterization. Hysteresis is implemented to avoid meta stable states and switching due to internal ground bounce. It cannot be guaranteed that it suppresses switching due to external system noise.

2) Single pull device is enabled for the measurement of P0.0/P1.0.

3) An additional error current (I_{INJ}) will flow if an overload current flows through an adjacent pin. TMS pin and RESET pin have internal pull devices and are not included in the input leakage current characteristic.

4) Not subjected to production test, verified by design/characterization.

5) Not subjected to production test, verified by design/characterization. However, for applications with strict low power-down current requirements, it is mandatory that no active voltage source is supplied at any GPIO pin when V_{DDP} is powered off.

4.2.2 Supply Threshold Characteristics

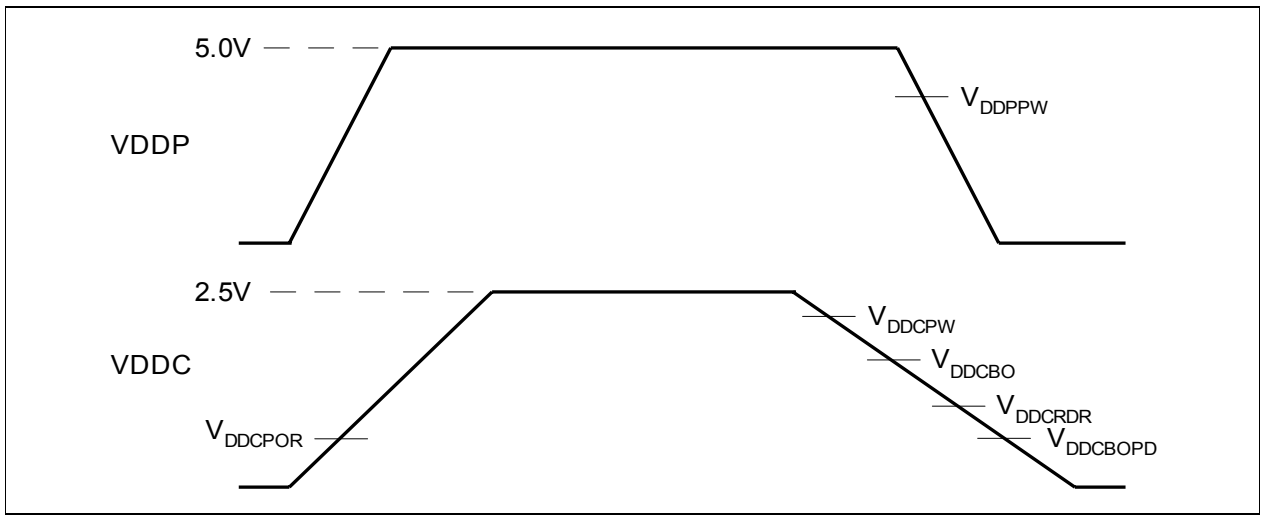


Figure 34 Supply Threshold Parameters

Table 34 Supply Threshold Parameters (Operating Conditions apply)

Parameters	Symbol		Limit Values			Unit
			min.	typ.	max.	
V_{DDC} prewarning voltage[1]	V_{DDCPW}	CC	2.2	2.3	2.4	V
V_{DDC} brownout voltage in active mode[1]	V_{DDCBO}	CC	2.0	2.1	2.3	V
RAM data retention voltage	V_{DDCRDR}	CC	0.9	1.0	1.1	V
V_{DDC} brownout voltage in power-down mode[2]	$V_{DDCBOPD}$	CC	1.3	1.5	1.7	V
V_{DDP} prewarning voltage	V_{DDPPW}	CC	3.4	4.0	4.6	V
Power-on reset voltage[2][3]	V_{DDCPOR}	CC	1.3	1.5	1.7	V

1) Detection is disabled in power-down mode.

2) Detection is enabled in both active and power-down mode.

3) The reset of EVR is extended by 300 µs typically after the VDDC reaches the power-on reset voltage.

4.2.3 ADC Characteristics

The values in the table below are given for an analog power supply between 4.5 V to 5.5 V. The ADC can be used with an analog power supply down to 3 V. Note that in this case, the analog part may show a reduced performance.All ground pins (V_{SS}) must be externally connected to one single star point in the system. The voltage difference between the ground pins must not exceed 200mV.

Table 35 ADC Characteristics (Operating Conditions apply; V_{DDP} = 5V Range)

Parameter	Symbol	Limit Values			Unit	Test Conditions/ Remarks
		min.	typ .	max.		
Analog reference voltage	V_{AREF} SR	V_{AGND} + 1	V_{DDP}	V_{DDP} + 0.05	V	
Analog reference ground	V_{AGND} SR	V_{SS} - 0.05	V_{SS}	V_{AREF} - 1	V	
Analog input voltage range	V_{AIN} SR	V_{AGND}	–	V_{AREF}	V	
ADC clocks	f_{ADC}	–	20	40	MHz	module clock
	f_{ADCI}	–	–	10	MHz	internal analog clock See **Figure 31**
Sample time	t_S CC	(2 + INPCR0.STC) × t_{ADCI}			µs	
Conversion time	t_C CC	See **Section 4.2.3.1**			µs	
Total unadjusted error	TUE[1] CC	–	–	±1	LSB	8-bit conversion.[2]
		–	–	±2	LSB	10-bit conversion.
Differential Nonlinearity	DNL CC	–	±1	–	LSB	10-bit conversion[4]
Integral Nonlinearity	INL CC	–	±1	–	LSB	10-bit conversion[4]
Offset	OFF CC	–	±1	–	LSB	10-bit conversion[4]
Gain	GAIN CC	–	±1	–	LSB	10-bit conversion[4]
Switched capacitance at the reference voltage input	C_{AREFSW} CC	–	10	20	pF	[2][3]

Table 35 ADC Characteristics (Operating Conditions apply; V_{DDP} = 5V Range)

Parameter	Symbol	Limit Values			Unit	Test Conditions/ Remarks
		min.	typ .	max.		
Switched capacitance at the analog voltage inputs	C_{AINSW} CC	–	5	7	pF	2)4)
Input resistance of the reference input	R_{AREF}CC	–	1	2	kΩ	2)
Input resistance of the selected analog channel	R_{AIN} CC	–	1	1.5	kΩ	2)

1) TUE is tested at V_{AREF} = 5.0 V, V_{AGND} = 0 V , V_{DDP} = 5.0 V.

2) Not subject to production test, verified by design/characterization.

3) This represents an equivalent switched capacitance. This capacitance is not switched to the reference voltage at once. Instead of this, smaller capacitances are successively switched to the reference voltage.

4) The sampling capacity of the conversion C-Network is pre-charged to V_{AREF}/2 before connecting the input to the C-Network. Because of the parasitic elements, the voltage measured at ANx is lower than V_{AREF}/2.

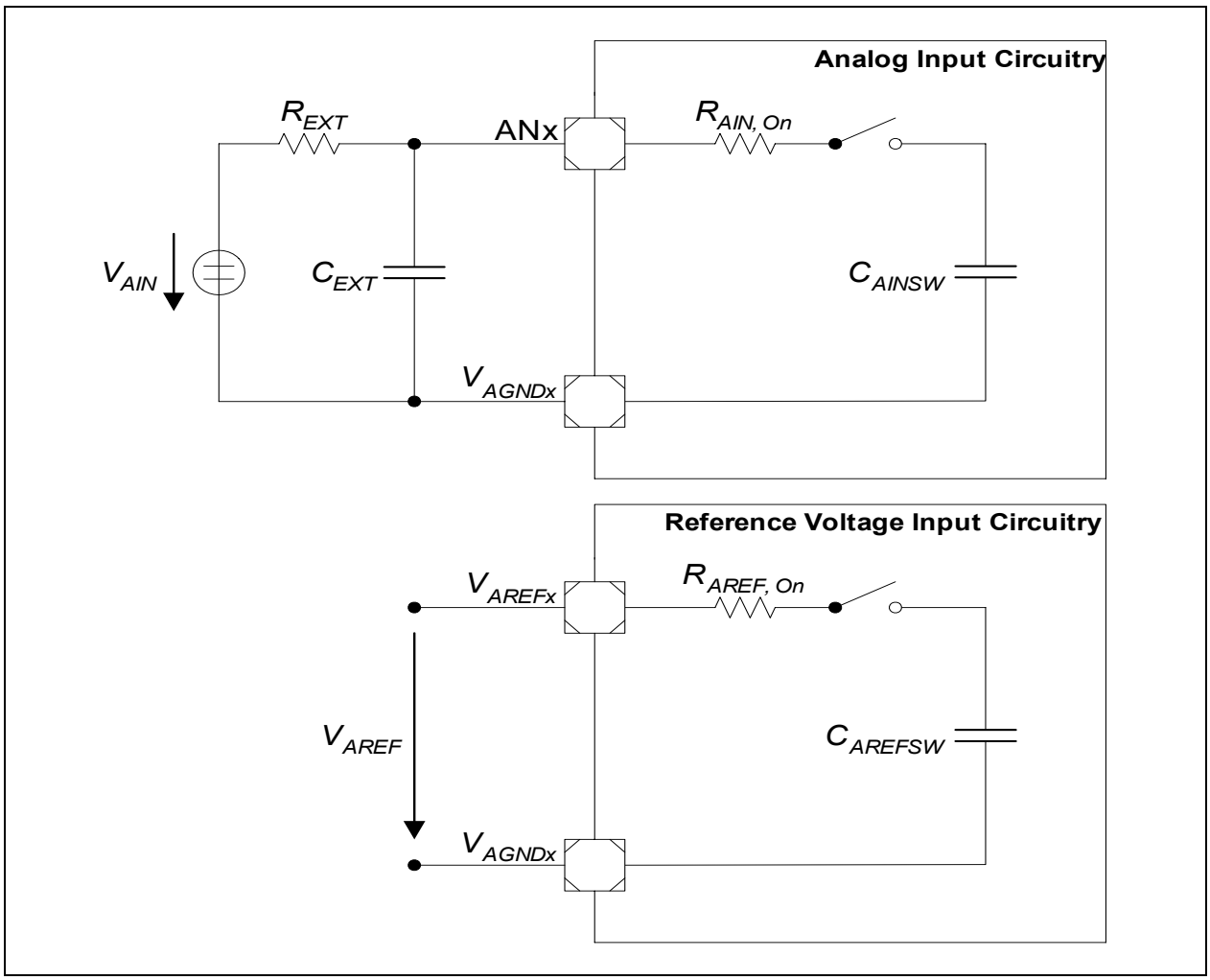


Figure 35 ADC Input Circuits

4.2.3.1 ADC Conversion Timing

Conversion time, $t_C = t_{ADC} \times (1 + r \times (3 + n + STC))$, where

r = CTC + 2 for CTC = 00_B, 01_B or 10_B,

r = 32 for CTC = 11_B,

CTC = Conversion Time Control (GLOBCTR.CTC),

STC = Sample Time Control (INPCR0.STC),

n = 8 or 10 (for 8-bitand 10-bit conversion respectively),

$t_{ADC} = 1 / f_{ADC}$

4.2.4 Power Supply Current

Table 36 Power Supply Current Parameters (Operating Conditions apply; V_{DDP} = 5V range)

Parameter	Symbol	Limit Values		Unit	Test Condition
		typ.[1]	max.[2]		
V_{DDP} = 5V Range					
Active Mode	I_{DDP}	22.6	24.5	mA	[3]
Idle Mode	I_{DDP}	12.5	14	mA	[4]
Active Mode with slow-down enabled	I_{DDP}	5.6	7.5	mA	[5]
Idle Mode with slow-down enabled	I_{DDP}	5.1	7.2	mA	[6]

1) The typical I_{DDP} values are periodically measured at T_A = + 25 °C and V_{DDP} = 5.0 V.

2) The maximum I_{DDP} values are measured under worst case conditions (T_A = + 125 °C and V_{DDP} = 5.5 V).

3) I_{DDP} (active mode) is measured with: CPU clock and input clock to all peripherals running at 26.7 MHz(set by on-chip oscillator of 10 MHz and NDIV in PLL_CON to 0010_B), $\overline{RESET}$ = V_{DDP}, no load on ports.

4) I_{DDP} (idle mode) is measured with: CPU clock disabled, watchdog timer disabled, input clock to all peripherals enabled and running at 26.7 MHz, $\overline{RESET}$ = V_{DDP}, no load on ports.

5) I_{DDP} (active mode with slow-down mode) is measured with: CPU clock and input clock to all peripherals running at 833 KHz by setting CLKREL in CMCON to 0101_B, $\overline{RESET}$ = V_{DDP}, no load on ports.

6) I_{DDP} (idle mode with slow-down mode) is measured with: CPU clock disabled, watchdog timer disabled, input clock to all peripherals enabled and running at 833 KHz by setting CLKREL in CMCON to 0101_B, $\overline{RESET}$ = V_{DDP}, no load on ports.

Table 37 Power Down Current (Operating Conditions apply; V_{DDP} = 5V range)

Parameter	Symbol	Limit Values		Unit	Test Condition
		typ.[1]	max.[2]		
V_{DDP} = 5V Range					
Power-Down Mode[3]	I_{PDP}	1	10	µA	T_A = + 25 °C.[4]
		-	35	µA	T_A = + 85 °C.[4][5]

1) The typical I_{PDP} values are measured at V_{DDP} = 5.0 V.

2) The maximum I_{PDP} values are measured at V_{DDP} = 5.5 V.

3) I_{PDP} (power-down mode) has a maximum value of 200 µA at T_A = + 125 °C.

4) I_{PDP} (power-down mode) is measured with: RESET = V_{DDP}, V_{AGND}= V_{SS}, RXD/INT0 = V_{DDP}; rest of the ports are programmed to be input with either internal pull devices enabled or driven externally to ensure no floating inputs.

5) Not subject to production test, verified by design/characterization.

Table 38 Power Supply Current Parameters (Operating Conditions apply; V_{DDP} = 3.3V range)

Parameter	Symbol	Limit Values		Unit	Test Condition
		typ.[1]	max.[2]		
V_{DDP} = 3.3V Range					
Active Mode	I_{DDP}	21.6	23.3	mA	[3]
Idle Mode	I_{DDP}	12	13.5	mA	[4]
Active Mode with slow-down enabled	I_{DDP}	5.7	7.3	mA	[5]
Idle Mode with slow-down enabled	I_{DDP}	5.4	6.9	mA	[6]

1) The typical I_{DDP} values are periodically measured at T_A = + 25 °C and V_{DDP} = 3.3 V.

2) The maximum I_{DDP} values are measured under worst case conditions (T_A = + 125 °C and V_{DDP} = 3.6 V).

3) I_{DDP} (active mode) is measured with: CPU clock and input clock to all peripherals running at 26.7 MHz (set by on-chip oscillator of 10 MHz and NDIV in PLL_CON to 0010_B), $\overline{RESET}$ = V_{DDP}, no load on ports.

4) I_{DDP} (idle mode) is measured with: CPU clock disabled, watchdog timer disabled, input clock to all peripherals enabled and running at 26.7 MHz, $\overline{RESET}$ = V_{DDP}, no load on ports.

5) I_{DDP} (active mode with slow-down mode) is measured with: CPU clock and input clock to all peripherals running at 833 KHz by setting CLKREL in CMCON to 0101_B, $\overline{RESET}$ = V_{DDP}, no load on ports.

6) I_{DDP} (idle mode with slow-down mode) is measured with: CPU clock disabled, watchdog timer disabled, input clock to all peripherals enable and running at 833 KHz by setting CLKREL in CMCON to 0101_B,, $\overline{RESET}$ = V_{DDP}, no load on ports.

Table 39 Power Down Current (Operating Conditions apply; V_{DDP} = 3.3V range)

Parameter	Symbol	Limit Values		Unit	Test Condition
		typ.[1]	max.[2]		
V_{DDP} = 3.3V Range					
Power-Down Mode[3]	I_{PDP}	1	10	µA	T_A = + 25 °C.[4]
		-	35	µA	T_A = + 85 °C.[4][5]

1) The typical I_{PDP} values are measured at V_{DDP} = 3.3 V.

2) The maximum I_{PDP} values are measured at V_{DDP} = 3.6 V.

3) I_{PDP} (power-down mode) has a maximum value of 200 µA at T_A = + 125 °C.

4) I_{PDP} (power-down mode) is measured with: $\overline{RESET}$ = V_{DDP}, V_{AGND}= V_{SS}, RXD/INT0= V_{DDP}; rest of the ports are programmed to be input with either internal pull devices enabled or driven externally to ensure no floating inputs.

5) Not subject to production test, verified by design/characterization.

4.3 AC Parameters

4.3.1 Testing Waveforms

The testing waveforms for rise/fall time, output delay and output high impedance are shown in **Figure 36**, **Figure 37** and **Figure 38**.

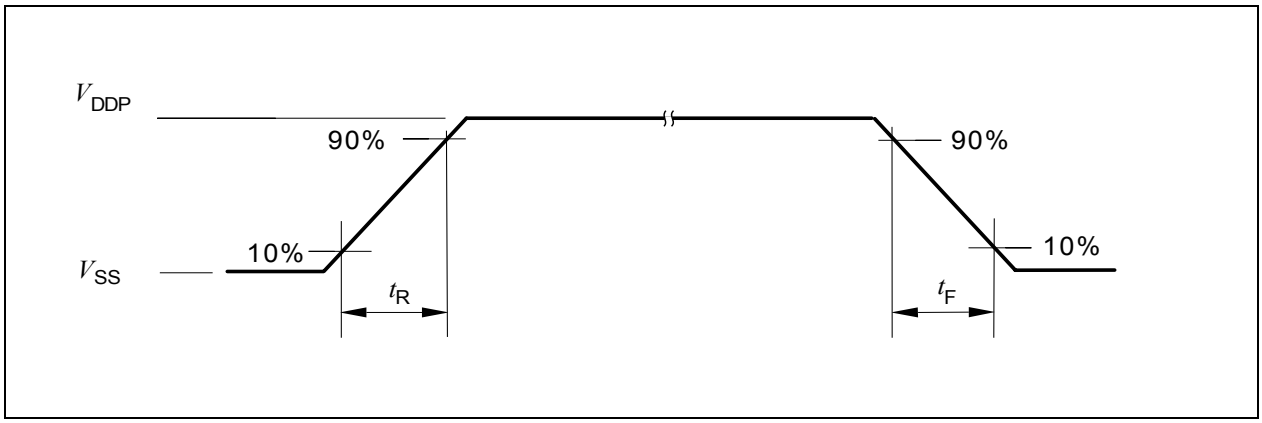


Figure 36 Rise/Fall Time Parameters

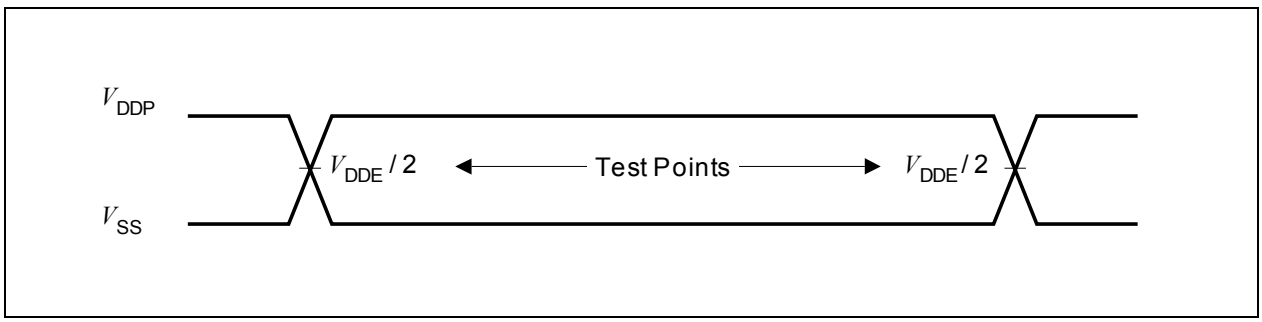


Figure 37 Testing Waveform, Output Delay

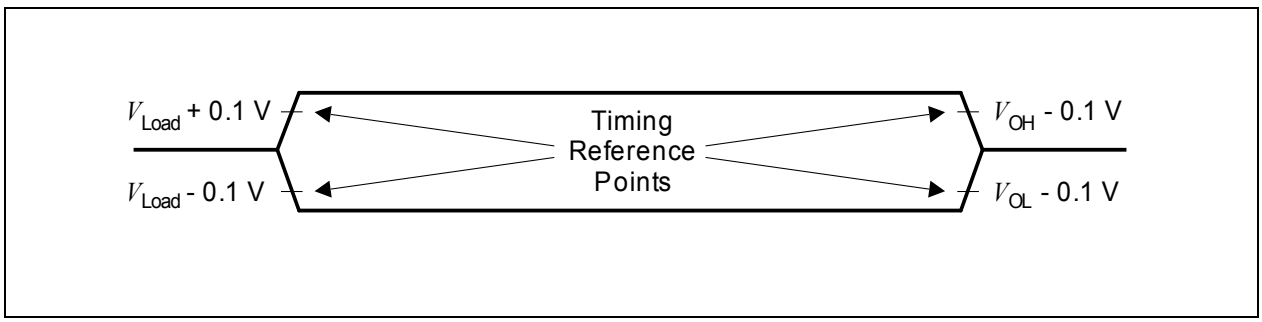


Figure 38 Testing Waveform, Output High Impedance

4.3.2 Output Rise/Fall Times

Table 40 Output Rise/Fall Times Parameters (Operating Conditions apply)

Parameter	Symbol	Limit Values		Unit	Test Conditions
		min.	max.		
V_{DDP} = 5V Range					
Rise/fall times [1] [2]	t_R, t_F	–	10	ns	20 pF. [3]
V_{DDP} = 3.3V Range					
Rise/fall times [1][2]	t_R, t_F	–	10	ns	20 pF. [4]

1) Rise/Fall time measurements are taken with 10% - 90% of the pad supply.

2) Not all parameters are 100% tested, but are verified by design/characterization and test correlation.

3) Additional rise/fall time valid for C_L = 20pF - 100pF @ 0.125 ns/pF.

4) Additional rise/fall time valid for C_L = 20pF - 100pF @ 0.225 ns/pF.

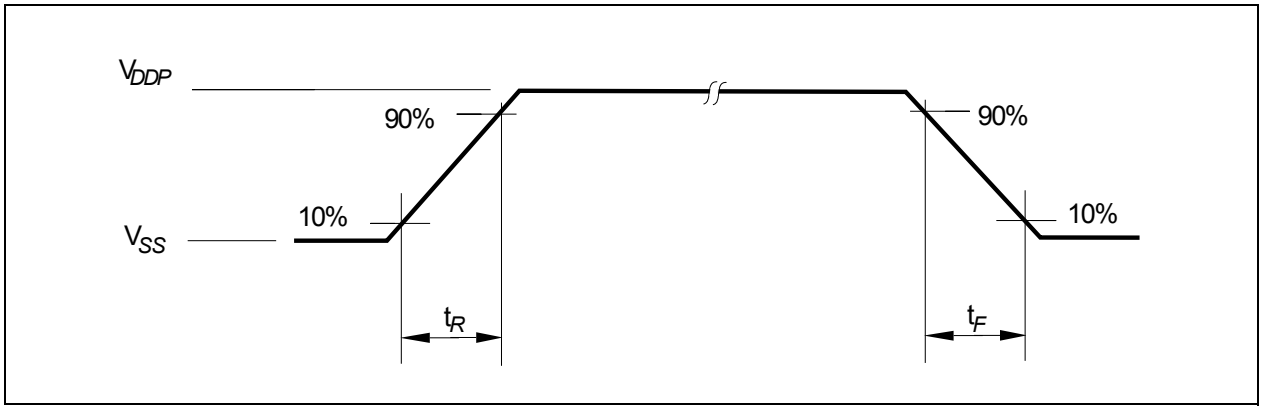


Figure 39 Rise/Fall Times Parameters

4.3.3 Power-on Reset and PLL Timing

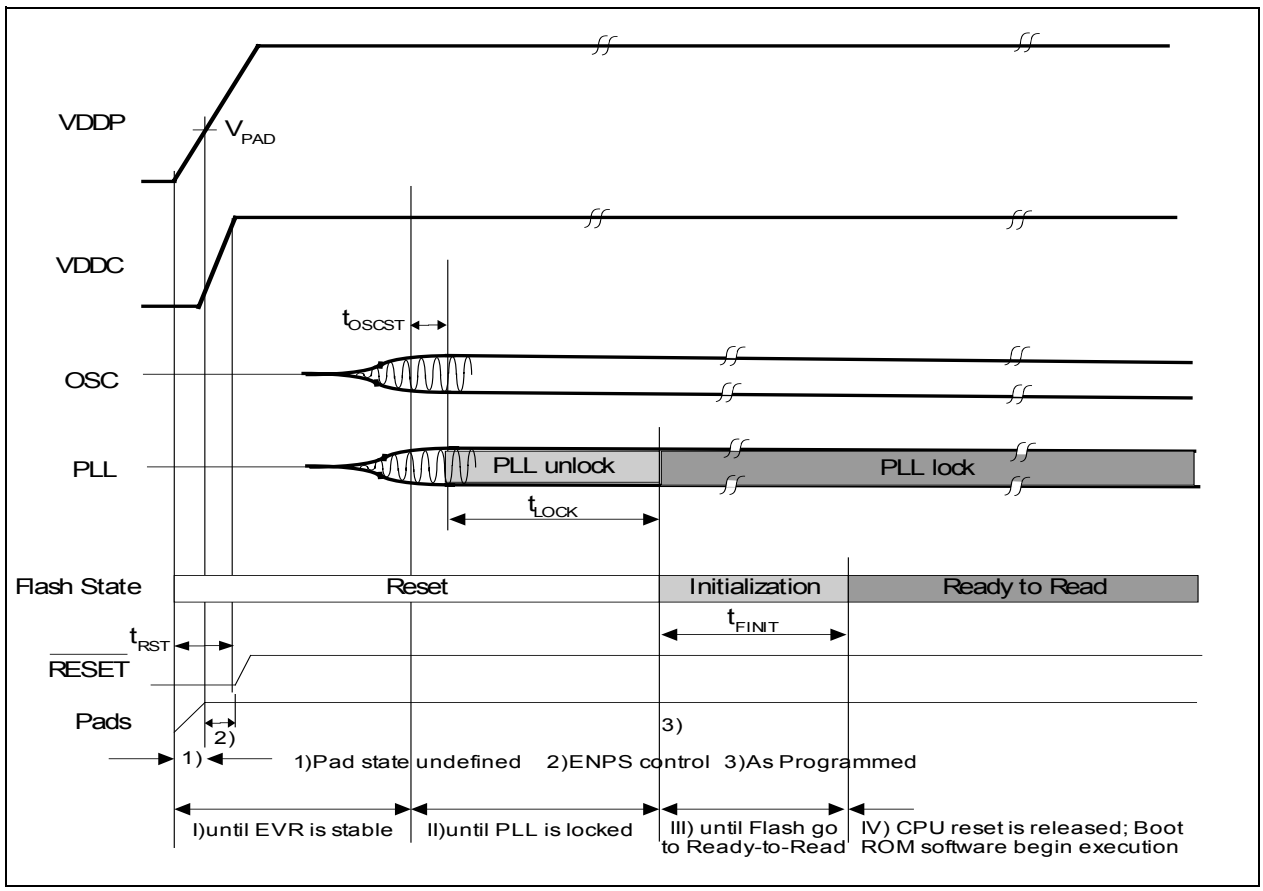


Figure 40 Power-on Reset Timing

Table 41 Power-On Reset and PLL Timing (Operating Conditions apply)

Parameter	Symbol	Limit Values			Unit	Test Conditions
		min.	typ.	max.		
Pad operating voltage	V_{PAD} CC	2.3	–	–	V	
On-Chip Oscillator start-up time	t_{OSCST} CC	–	–	500	ns	
Flash initialization time	t_{FINIT} CC	–	160	–	µs	
$\overline{\text{RESET}}$ hold time[1)]	t_{RST} SR	–	500	–	µs	V_{DDP} rise time (10% – 90%) ≤ 500µs
PLL lock-in time	t_{LOCK} CC	–	–	200	µs	
PLL accumulated jitter	D_P	–	–	0.7	ns	2)

1) $\overline{\text{RESET}}$ signal has to be active (low) until V_{DDC} has reached 90% of its maximum value (typ. 2.5V).

2) PLL lock at 80 MHz using a 4 MHz external oscillator. The PLL Divider settings are K = 2, N = 40 and P = 1.

4.3.4 On-Chip Oscillator Characteristics

Table 42 On-Chip Oscillator Characteristics (Operating Conditions apply)

Parameter	Symbol	Limit Values			Unit	Test Conditions
		min.	typ.	max.		
Nominal frequency	f_{NOM} CC	9.75	10	10.25	MHz	under nominal conditions[1] after IFX-backend trimming
Long term frequency deviation[2]	Δf_{LT} CC	-5.0	–	5.0	%	with respect to f_{NOM}, over lifetime and temperature (-10°C to 85°C), for one device after trimming
		-6	–	0	%	with respect to f_{NOM}, over lifetime and temperature (-40°C to -10°C), for one device after trimming
Short term frequency deviation	Δf_{ST} CC	-1.0	–	1.0	%	with respect to f_{NOM}, from <10 ms to 100 ms

1) Nominal condition: V_{DDC} = 2.5 V, T_A = + 25°C.

2) Not all parameters are 100% tested, but are verified by design/characterization and test correlation.

4.3.5 JTAG Timing

Table 43 TCK Clock Timing (Operating Conditions apply; C_L = 50 pF)

Parameter	Symbol		Limits		Unit
			min	max	
TCK clock period	t_{TCK}	SR	50	–	ns
TCK high time	t_1	SR	20	–	ns
TCK low time	t_2	SR	20	–	ns
TCK clock rise time	t_3	SR	–	4	ns
TCK clock fall time	t_4	SR	–	4	ns

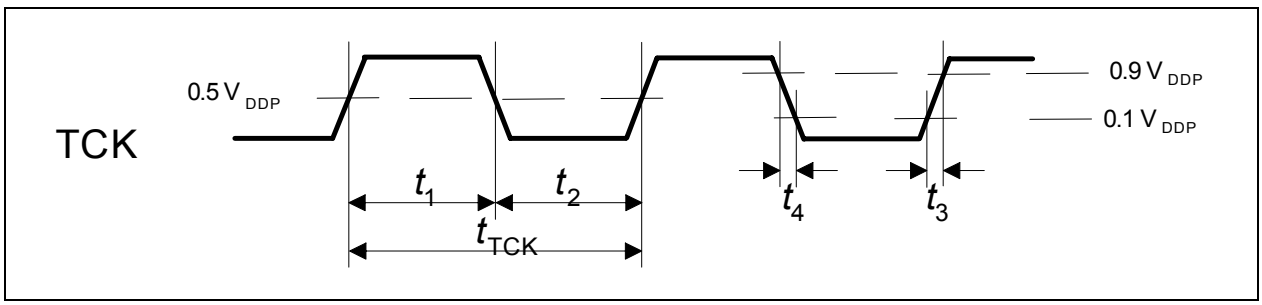


Figure 41 TCK Clock Timing

Table 44 JTAG Timing (Operating Conditions apply; C_L = 50 pF)

Parameter	Symbol		Limits		Unit
			min	max	
TMS setup to TCK ↑	t_1	SR	8.0	–	ns
TMS hold to TCK ↑	t_2	SR	5.0	–	ns
TDI setup to TCK ↑	t_1	SR	11.0	–	ns
TDI hold to TCK ↑	t_2	SR	6.0	–	ns
TDO valid output from TCK ↓	t_3	CC	–	23	ns
TDO high impedance to valid output from TCK ↓	t_4	CC	–	26	ns
TDO valid output to high impedance from TCK ↓	t_5	CC	–	18	ns

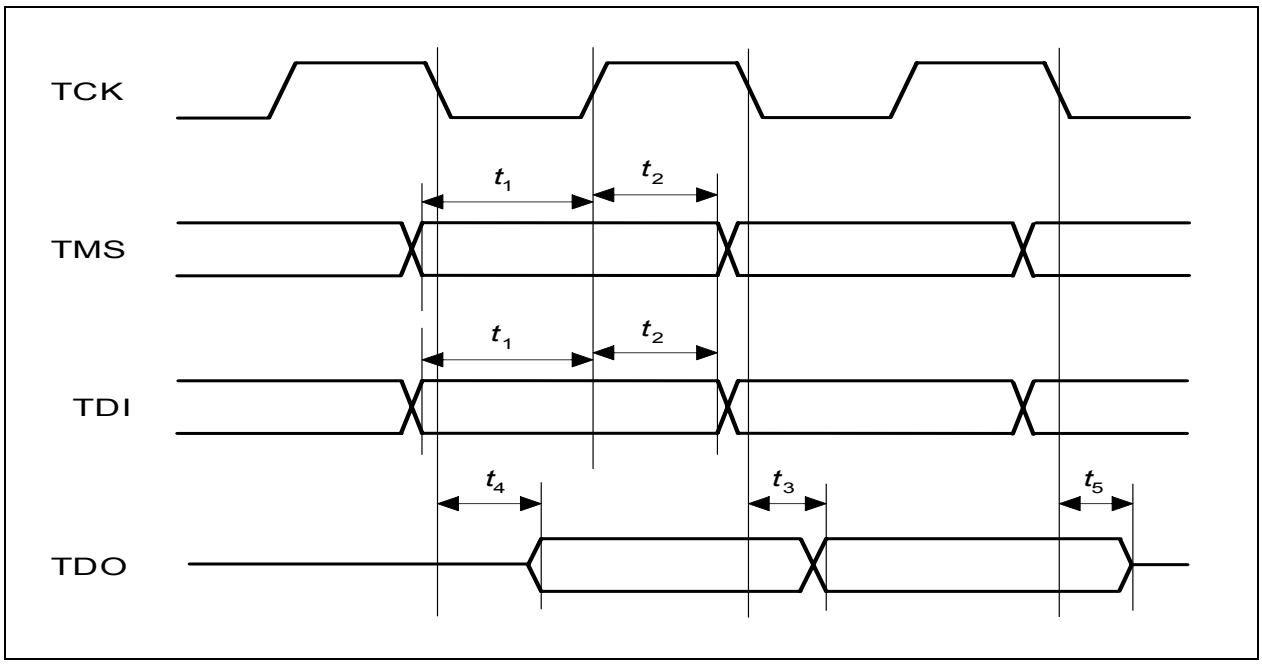


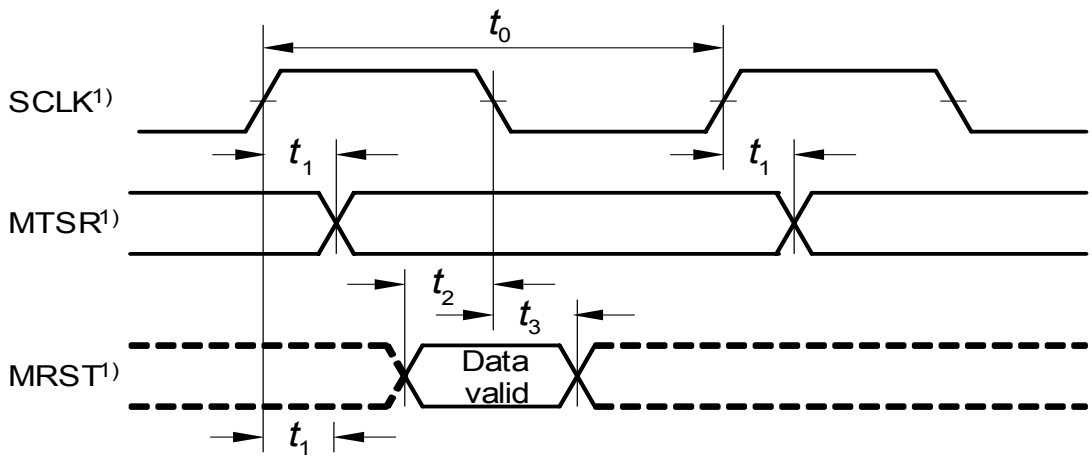
Figure 42 JTAG Timing

4.3.6 SSC Master Mode Timing

Table 45 SSC Master Mode Timing (Operating Conditions apply; C_L = 50 pF)

Parameter	Symbol		Limit Values		Unit
			min.	max.	
SCLK clock period	t_0	CC	2*T_{SSC} 1)	–	ns
MTSR delay from SCLK ↓	t_1	CC	0	8	ns
MRST setup to SCLK ↓	t_2	SR	22	–	ns
MRST hold from SCLK ↓	t_3	SR	0	–	ns

1) T_{SSCmin} = T_{CPU} = 1/f_{CPU}. When f_{CPU} = 26.7MHz, t_0 = 74.9ns. T_{CPU} is the CPU clock period.



SSC Master Mode Timing

5 Package and Reliability

5.1 Package Parameters (PG-TSSOP-20)

Table 46 provides the thermal characteristics of the package.

Table 46 Thermal Characteristics of the Package

Parameter	Symbol	Limit Values		Unit	Notes
		Min.	Max.		
Thermal resistance junction case top[1)]	R_{TJCT} CC	–	28.5	K/W	–
Thermal resistance junction case bottom[1)]	R_{TJCB} CC	–	43.7	K/W	–

1) The top and bottom thermal resistances between the case and the ambient (R_{TCAT}, R_{TCAB}) are to be combined with the thermal resistances between the junction and the case given above (R_{TJCT}, R_{TJCB}), in order to calculate the total thermal resistance between the junction and the ambient (R_{TJA}). The thermal resistances between the case and the ambient (R_{TCAT}, R_{TCAB}) depend on the external system (PCB, case) characteristics, and are under user responsibility.
The junction temperature can be calculated using the following equation: $T_J=T_A+R_{TJA} \times P_D$, where the R_{TJA} is the total thermal resistance between the junction and the ambient. This total junction ambient resistance R_{TJA} can be obtained from the upper four partial thermal resistances, by
a) simply adding only the two bottom thermal resistances (junction case bottom and case ambient bottom), or
b) by taking all four resistances into account,
depending on the precision needed.

5.2 Package Outline

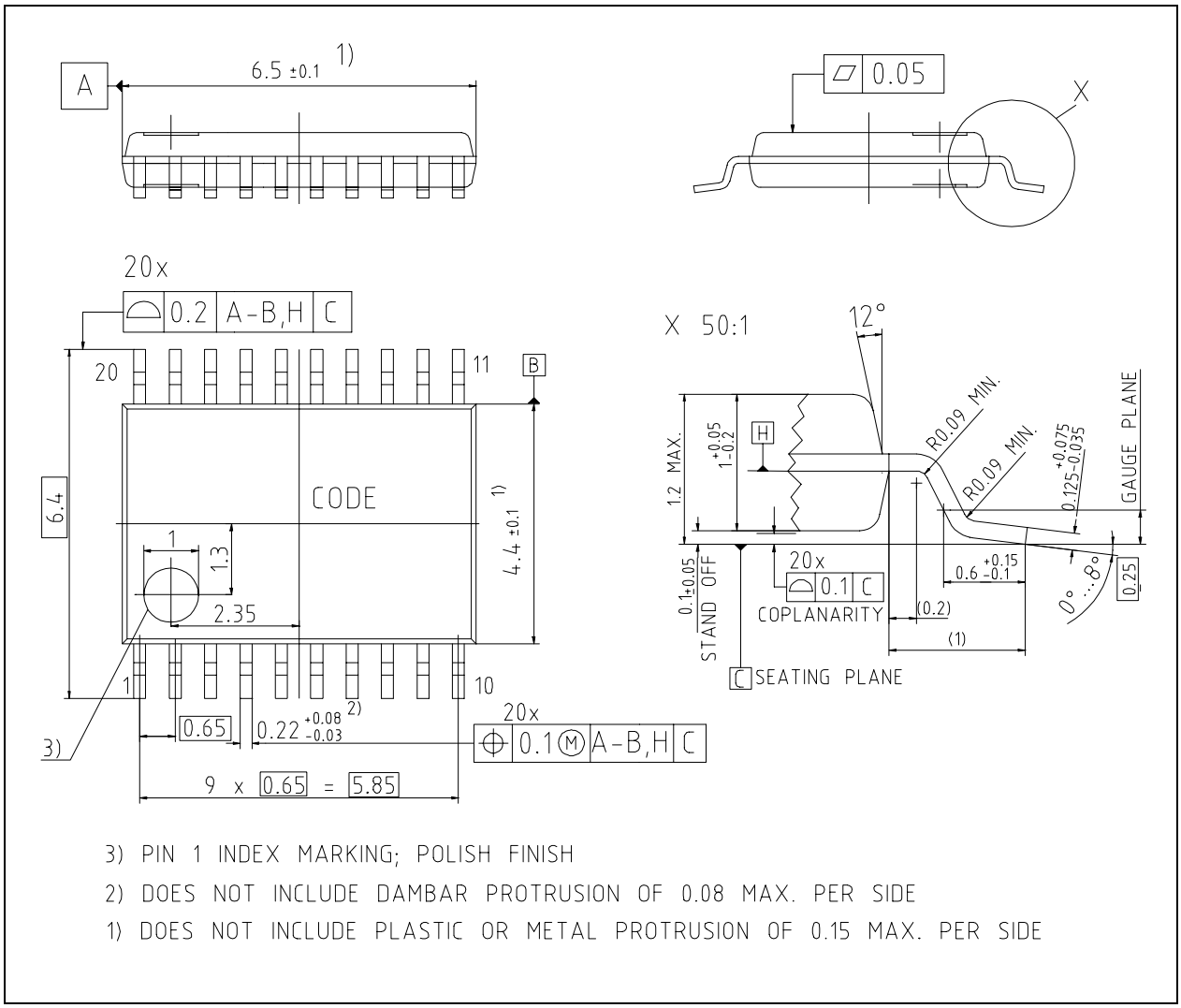


Figure 43 PG-TSSOP-20 Package Outline

5.3 Quality Declaration

Table 47 shows the characteristics of the quality parameters in the XC864.

Table 47 Quality Parameters

Parameter	Symbol	Limit Values		Unit	Notes
		Min.	Max.		
ESD susceptibility according to Human Body Model (HBM)	V_{HBM}	–	2000	V	Conforming to EIA/JESD22-A114-B
ESD susceptibility according to Charged Device Model (CDM) pins	V_{CDM}	–	500	V	Conforming to JESD22-C101-C

w w w . i n f i n e o n . c o m

Published by Infineon Technologies AG