# THE DATASHEET OF

# SAK-XC886CM-6FFA 5V AC

Never stop thinking

# 8-Bit

# XC886/888CLM

## 8-Bit Single Chip Microcontroller

## Data Sheet

V1.2 2009-07

Microcontrollers

**Edition 2009-07**

**Published by**
**Infineon Technologies AG**
**81726 Munich, Germany**

**© 2009 Infineon Technologies AG**
**All Rights Reserved.**

### 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

# XC886/888CLM

## 8-Bit Single Chip Microcontroller

## Data Sheet

V1.2 2009-07

Microcontrollers

**XC886/888 Data Sheet**

**Revision History: V1.2 2009-07**

| Previous Versions: V1.0, V1.1 | |
|---|---|
| Page | Subjects (major changes since last revision) |
| Changes from V1.1 2009-01 to V1.2 2009-07 | |
| 89 | Note on LIN baud rate detection is added. |
| 92 | RXD slave line in SSC block diagram is updated. |
| 108 | Electrical parameters are now valid for all variants, previous note on exclusion of ROM variants is removed. |
| 116 | Symbol for ADC error parameters are updated. |
| 120 | Power supply current parameters for ROM variants are updated. |
| 128 | Test condition for the on-chip oscillator short term deviation is updated. |
| | |
| | |
| | |
| | |
| | |

**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**

# Table of Contents


| | | |
|---|---|---|
| **1** | **Summary of Features** | 1 |
| **2** | **General Device Information** | 5 |
| 2.1 | Block Diagram | 5 |
| 2.2 | Logic Symbol | 6 |
| 2.3 | Pin Configuration | 7 |
| 2.4 | Pin Definitions and Functions | 9 |
| **3** | **Functional Description** | 19 |
| 3.1 | Processor Architecture | 19 |
| 3.2 | Memory Organization | 20 |
| 3.2.1 | Memory Protection Strategy | 21 |
| 3.2.1.1 | Flash Memory Protection | 21 |
| 3.2.2 | Special Function Register | 23 |
| 3.2.2.1 | Address Extension by Mapping | 23 |
| 3.2.2.2 | Address Extension by Paging | 25 |
| 3.2.3 | Bit Protection Scheme | 29 |
| 3.2.3.1 | Password Register | 30 |
| 3.2.4 | XC886/888 Register Overview | 31 |
| 3.2.4.1 | CPU Registers | 31 |
| 3.2.4.2 | MDU Registers | 32 |
| 3.2.4.3 | CORDIC Registers | 33 |
| 3.2.4.4 | System Control Registers | 34 |
| 3.2.4.5 | WDT Registers | 36 |
| 3.2.4.6 | Port Registers | 37 |
| 3.2.4.7 | ADC Registers | 39 |
| 3.2.4.8 | Timer 2 Registers | 43 |
| 3.2.4.9 | Timer 21 Registers | 43 |
| 3.2.4.10 | CCU6 Registers | 44 |
| 3.2.4.11 | UART1 Registers | 48 |
| 3.2.4.12 | SSC Registers | 49 |
| 3.2.4.13 | MultiCAN Registers | 49 |
| 3.2.4.14 | OCDS Registers | 50 |
| 3.3 | Flash Memory | 52 |
| 3.3.1 | Flash Bank Sectorization | 53 |
| 3.3.2 | Parallel Read Access of P-Flash | 54 |
| 3.3.3 | Flash Programming Width | 55 |
| 3.4 | Interrupt System | 56 |
| 3.4.1 | Interrupt Source | 56 |
| 3.4.2 | Interrupt Source and Vector | 62 |
| 3.4.3 | Interrupt Priority | 64 |
| 3.5 | Parallel Ports | 65 |

3.6 Power Supply System with Embedded Voltage Regulator . . . . . 68
3.7 Reset Control . . . . . 69
3.7.1 Module Reset Behavior . . . . . 71
3.7.2 Booting Scheme . . . . . 71
3.8 Clock Generation Unit . . . . . 72
3.8.1 Recommended External Oscillator Circuits . . . . . 75
3.8.2 Clock Management . . . . . 77
3.9 Power Saving Modes . . . . . 79
3.10 Watchdog Timer . . . . . 80
3.11 Multiplication/Division Unit . . . . . 83
3.12 CORDIC Coprocessor . . . . . 84
3.13 UART and UART1 . . . . . 85
3.13.1 Baud-Rate Generator . . . . . 85
3.13.2 Baud Rate Generation using Timer 1 . . . . . 88
3.14 Normal Divider Mode (8-bit Auto-reload Timer) . . . . . 88
3.15 LIN Protocol . . . . . 89
3.15.1 LIN Header Transmission . . . . . 89
3.16 High-Speed Synchronous Serial Interface . . . . . 91
3.17 Timer 0 and Timer 1 . . . . . 93
3.18 Timer 2 and Timer 21 . . . . . 94
3.19 Capture/Compare Unit 6 . . . . . 95
3.20 Controller Area Network (MultiCAN) . . . . . 97
3.21 Analog-to-Digital Converter . . . . . 99
3.21.1 ADC Clocking Scheme . . . . . 99
3.21.2 ADC Conversion Sequence . . . . . 101
3.22 On-Chip Debug Support . . . . . 102
3.22.1 JTAG ID Register . . . . . 103
3.23 Chip Identification Number . . . . . 104

**4 Electrical Parameters** . . . . . 108
4.1 General Parameters . . . . . 108
4.1.1 Parameter Interpretation . . . . . 108
4.1.2 Absolute Maximum Rating . . . . . 109
4.1.3 Operating Conditions . . . . . 110
4.2 DC Parameters . . . . . 111
4.2.1 Input/Output Characteristics . . . . . 111
4.2.2 Supply Threshold Characteristics . . . . . 115
4.2.3 ADC Characteristics . . . . . 116
4.2.3.1 ADC Conversion Timing . . . . . 119
4.2.4 Power Supply Current . . . . . 120
4.3 AC Parameters . . . . . 124
4.3.1 Testing Waveforms . . . . . 124
4.3.2 Output Rise/Fall Times . . . . . 125

4.3.3 Power-on Reset and PLL Timing . . . . . 126
4.3.4 On-Chip Oscillator Characteristics . . . . . 128
4.3.5 External Clock Drive XTAL1 . . . . . 129
4.3.6 JTAG Timing . . . . . 130
4.3.7 SSC Master Mode Timing . . . . . 132

**5 Package and Quality Declaration** . . . . . 133
5.1 Package Parameters . . . . . 133
5.2 Package Outline . . . . . 134
5.3 Quality Declaration . . . . . 136

# 1 Summary of Features

The XC886/888 has the following 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
  - 12 Kbytes of Boot ROM
  - 256 bytes of RAM
  - 1.5 Kbytes of XRAM
  - 24/32 Kbytes of Flash; or
    24/32 Kbytes of ROM, with additional 4 Kbytes of Flash
    (includes memory protection strategy)
- I/O port supply at 3.3 V or 5.0 V and core logic supply at 2.5 V (generated by embedded voltage regulator)

(more features on next page)

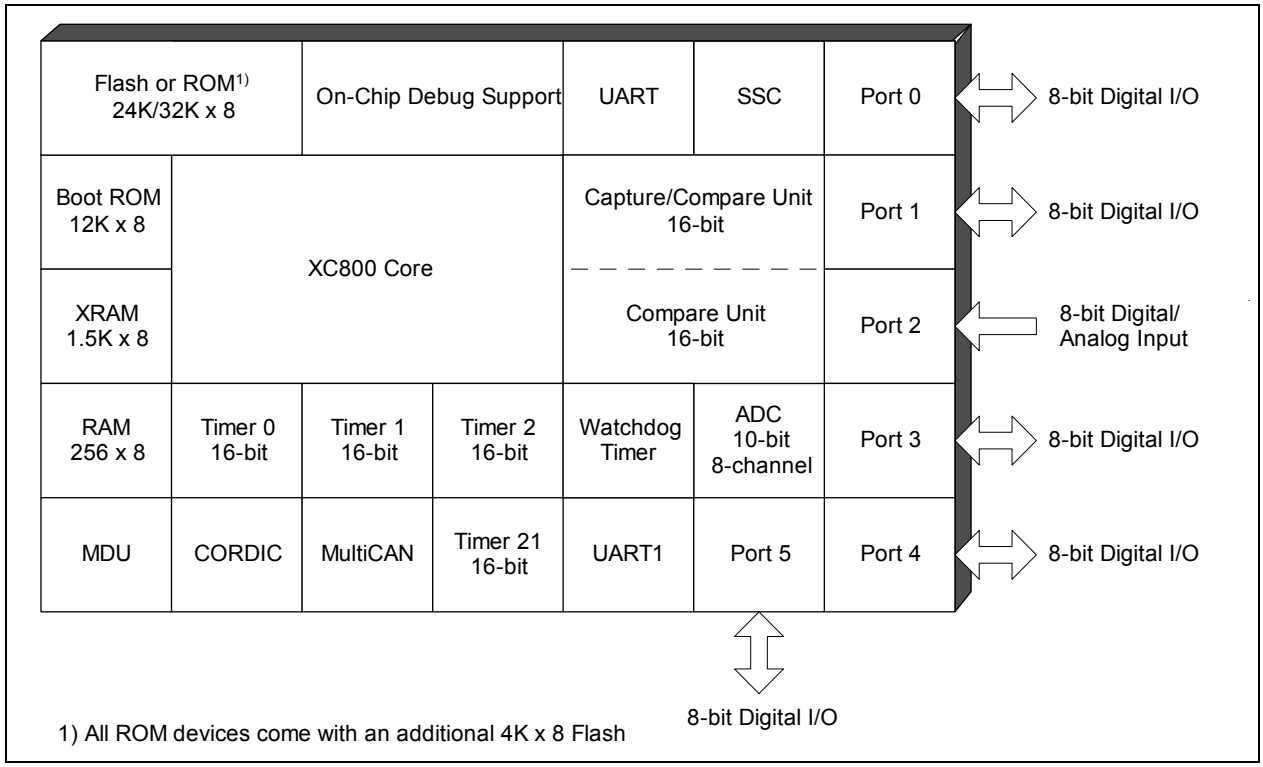



**Figure 1 XC886/888 Functional Units**

Features: (continued)

- Power-on reset generation
- Brownout detection for core logic supply
- 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)
- Six ports
  - Up to 48 pins as digital I/O
  - 8 pins as digital/analog input
- 8-channel, 10-bit ADC
- Four 16-bit timers
  - Timer 0 and Timer 1 (T0 and T1)
  - Timer 2 and Timer 21 (T2 and T21)
- Multiplication/Division Unit for arithmetic operations (MDU)
- Software libraries to support floating point and MDU calculations
- CORDIC Coprocessor for computation of trigonometric, hyperbolic and linear functions
- MultiCAN with 2 nodes, 32 message objects
- Capture/compare unit for PWM signal generation (CCU6)
- Two full-duplex serial interfaces (UART and UART1)
- Synchronous serial channel (SSC)
- On-chip debug support
  - 1 Kbyte of monitor ROM (part of the 12-Kbyte Boot ROM)
  - 64 bytes of monitor RAM
- Packages:
  - PG-TQFP-48
  - PG-TQFP-64
- Temperature range $T_A$:
  - SAF (-40 to 85 °C)
  - SAK (-40 to 125 °C)

### XC886/888 Variant Devices

The XC886/888 product family features devices with different configurations, program memory sizes, package options, power supply voltage, temperature and quality profiles (Automotive or Industrial), to offer cost-effective solutions for different application requirements.

The list of XC886/888 device configurations are summarized in **Table 1**. For each configuration, 2 types of packages are available:

- PG-TQFP-48, which is denoted by XC886 and;
- PG-TQFP-64, which is denoted by XC888.

**Table 1 Device Configuration**

| Device Name | CAN Module | LIN BSL Support | MDU Module |
|---|---|---|---|
| XC886/888 | No | No | No |
| XC886/888C | Yes | No | No |
| XC886/888CM | Yes | No | Yes |
| XC886/888LM | No | Yes | Yes |
| XC886/888CLM | Yes | Yes | Yes |

*Note: For variants with LIN BSL support, only LIN BSL is available regardless of the availability of the CAN module.*

From these 10 different combinations of configuration and package type, each are further made available in many sales types, which are grouped according to device type, program memory sizes, power supply voltage, temperature and quality profile (Automotive or Industrial), as shown in **Table 2**.

**Table 2 Device Profile**

| Sales Type | Device Type | Program Memory (Kbytes) | Power Supply (V) | Temp-erature (°C) | Quality Profile |
|---|---|---|---|---|---|
| SAK-XC886*/888*-8FFA 5V | Flash | 32 | 5.0 | -40 to 125 | Automotive |
| SAK-XC886*/888*-6FFA 5V | Flash | 24 | 5.0 | -40 to 125 | Automotive |
| SAF-XC886*/888*-8FFA 5V | Flash | 32 | 5.0 | -40 to 85 | Automotive |
| SAF-XC886*/888*-6FFA 5V | Flash | 24 | 5.0 | -40 to 85 | Automotive |
| SAF-XC886*/888*-8FFI 5V | Flash | 32 | 5.0 | -40 to 85 | Industrial |
| SAF-XC886*/888*-6FFI 5V | Flash | 24 | 5.0 | -40 to 85 | Industrial |

**Table 2 Device Profile** (cont'd)

| Sales Type | Device Type | Program Memory (Kbytes) | Power Supply (V) | Temp-erature (°C) | Quality Profile |
|---|---|---|---|---|---|
| SAK-XC886*/888*-8FFA 3V3 | Flash | 32 | 3.3 | -40 to 125 | Automotive |
| SAK-XC886*/888*-6FFA 3V3 | Flash | 24 | 3.3 | -40 to 125 | Automotive |
| SAF-XC886*/888*-8FFA 3V3 | Flash | 32 | 3.3 | -40 to 85 | Automotive |
| SAF-XC886*/888*-6FFA 3V3 | Flash | 24 | 3.3 | -40 to 85 | Automotive |
| SAF-XC886*/888*-8FFI 3V3 | Flash | 32 | 3.3 | -40 to 85 | Industrial |
| SAF-XC886*/888*-6FFI 3V3 | Flash | 24 | 3.3 | -40 to 85 | Industrial |

*Note: The asterisk (*) above denotes the device configuration letters from **Table 1**. Corresponding ROM derivatives will be available on request.*

*Note: For variants with LIN BSL support, only LIN BSL is available regardless of the availability of the CAN module.*

As this document refers to all the derivatives, some description may not apply to a specific product. For simplicity, all versions are referred to by the term XC886/888 throughout this document.

### 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 XC886/888, please refer to your responsible sales representative or your local distributor.

*Note: The ordering codes for the Mask-ROM versions are defined for each product after verification of the respective ROM code.*

# 2 General Device Information

**Chapter 2** contains the block diagram, pin configurations, definitions and functions of the XC886/888.

## 2.1 Block Diagram

The block diagram of the XC886/888 is shown in **Figure 2**.

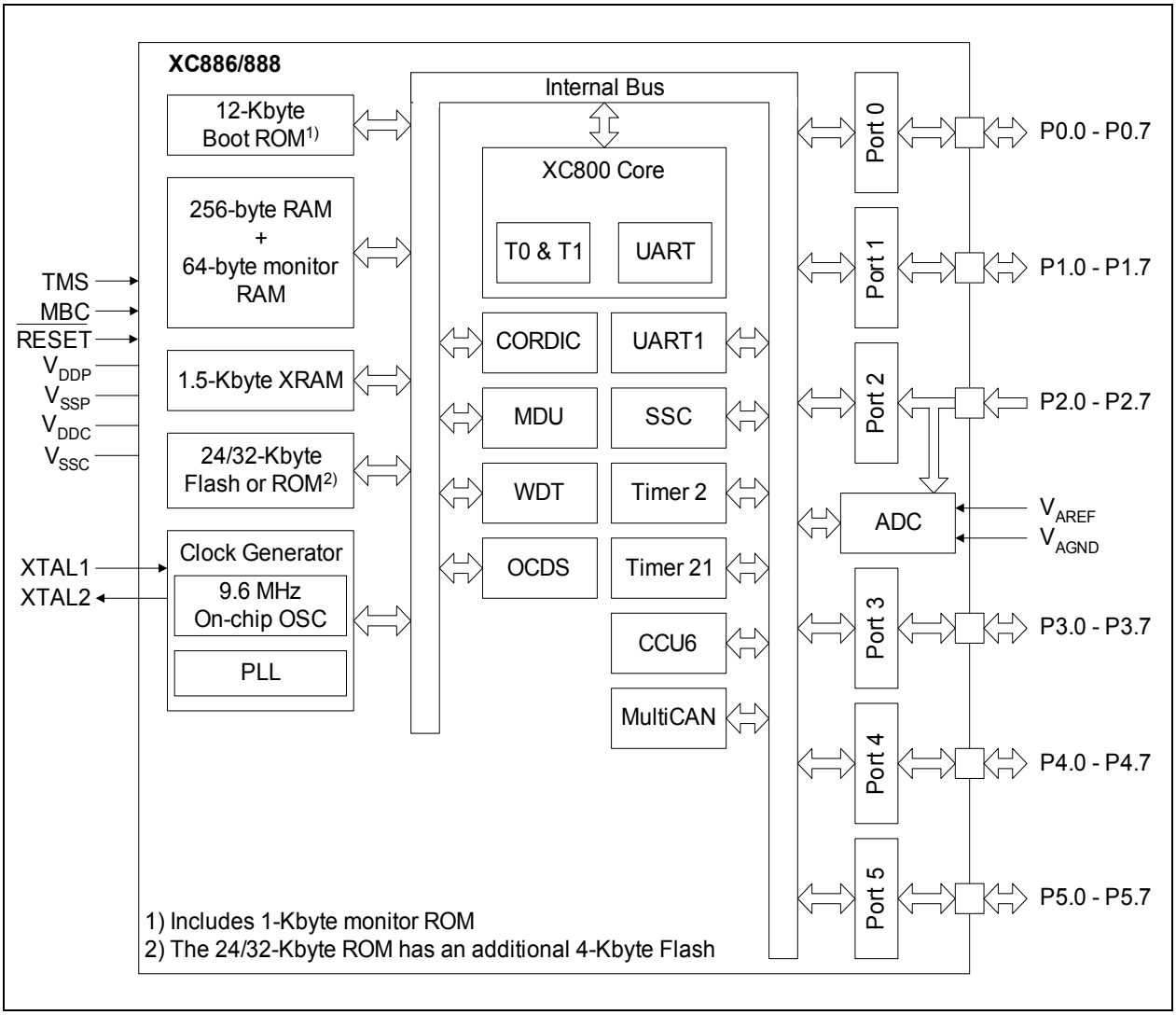



**Figure 2 XC886/888 Block Diagram**

## 2.2 Logic Symbol

The logic symbols of the XC886/888 are shown in **Figure 3**.

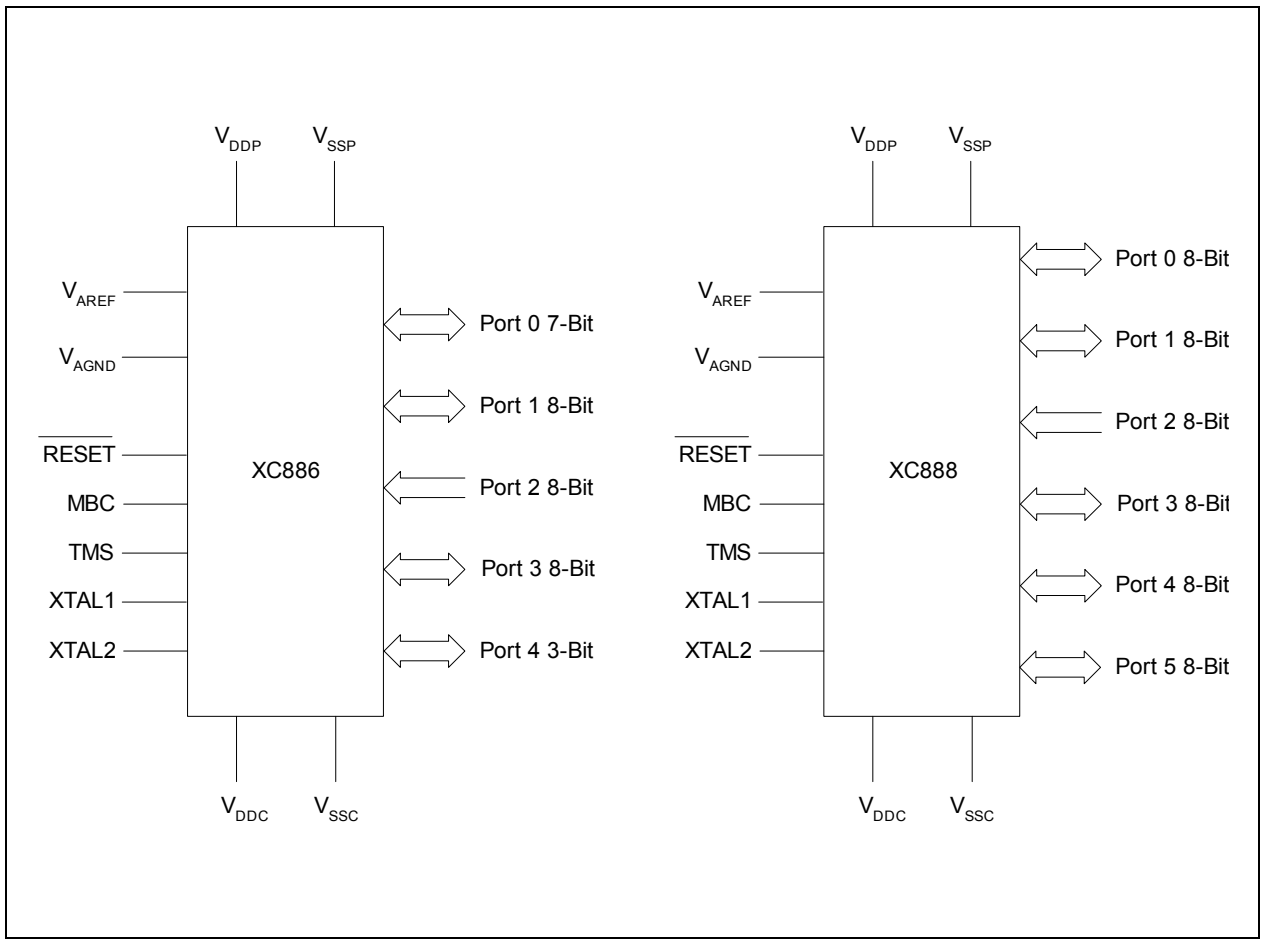



**Figure 3 XC886/888 Logic Symbol**

## 2.3 Pin Configuration

The pin configuration of the XC886, which is based on the PG-TQFP-48 package, is shown in **Figure 4**, while that of the XC888, which is based on the PG-TQFP-64 package, is shown in **Figure 5**.

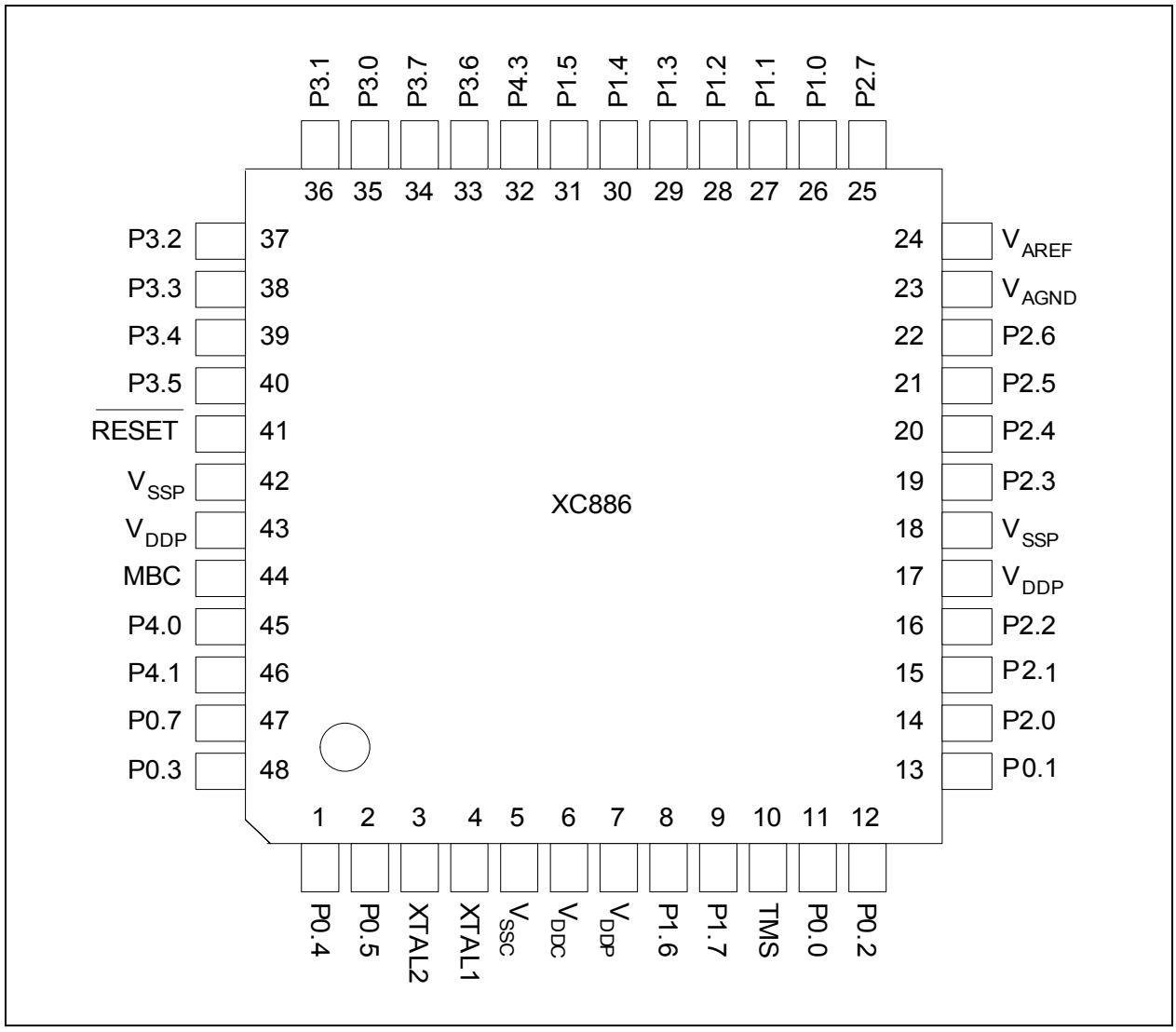
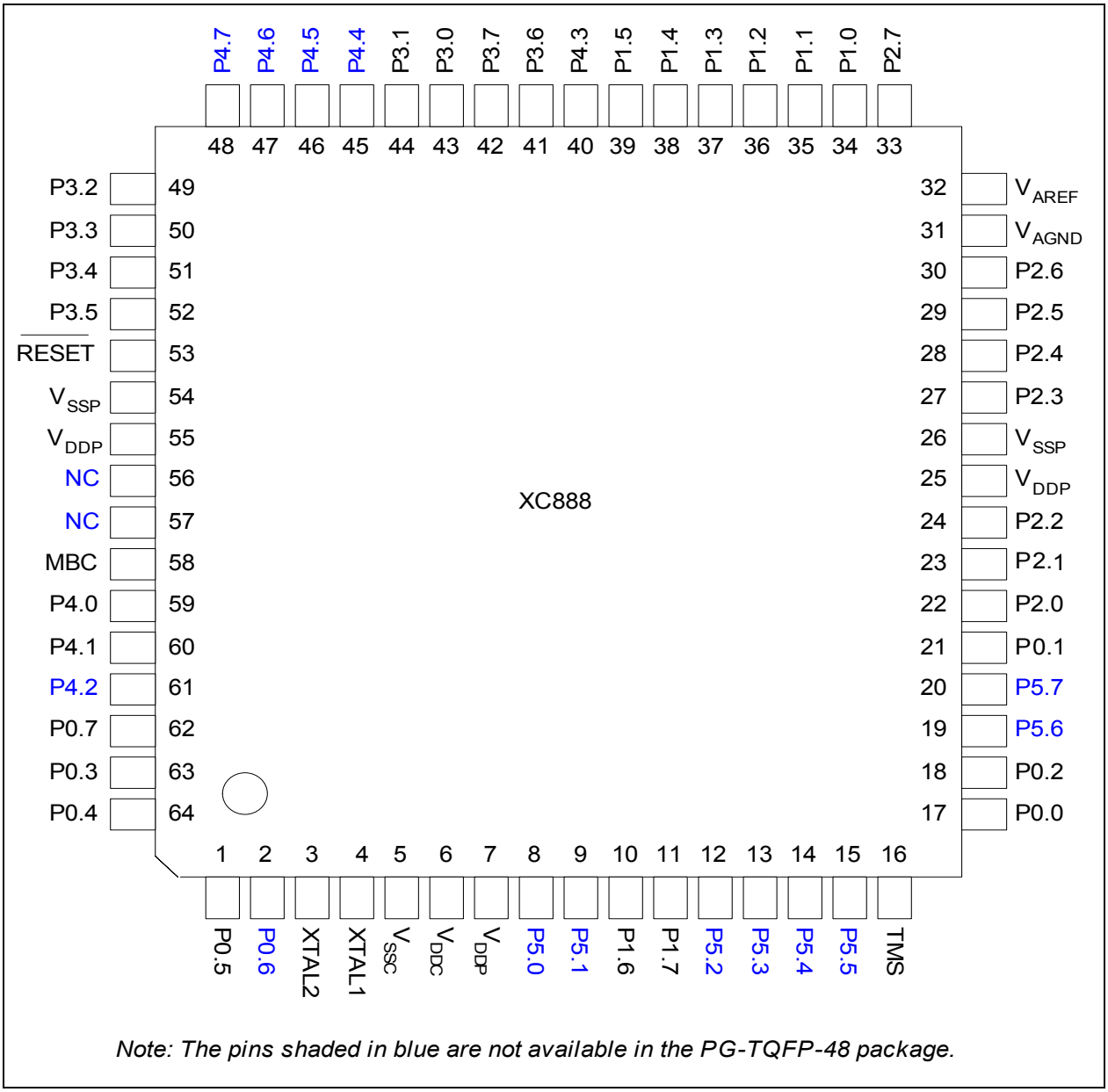



**Figure 4 XC886 Pin Configuration, PG-TQFP-48 Package** (top view)




*Note: The pins shaded in blue are not available in the PG-TQFP-48 package.*

**Figure 5 XC888 Pin Configuration, PG-TQFP-64 Package** (top view)

## 2.4 Pin Definitions and Functions

The functions and default states of the XC886/888 external pins are provided in **Table 3**.

**Table 3 Pin Definitions and Functions**

| Symbol | Pin Number (TQFP-48/64) | Type | Reset State | Function |
|---|---|---|---|---|
| **P0** | | I/O | | **Port 0**<br>Port 0 is an 8-bit bidirectional general purpose I/O port. It can be used as alternate functions for the JTAG, CCU6, UART, UART1, Timer 2, Timer 21, MultiCAN and SSC. |
| P0.0 | 11/17 | | Hi-Z | TCK_0 JTAG Clock Input<br>T12HR_1 CCU6 Timer 12 Hardware Run Input<br>CC61_1 Input/Output of Capture/Compare channel 1<br>CLKOUT_0 Clock Output<br>RXDO_1 UART Transmit Data Output |
| P0.1 | 13/21 | | Hi-Z | TDI_0 JTAG Serial Data Input<br>T13HR_1 CCU6 Timer 13 Hardware Run Input<br>RXD_1 UART Receive Data Input<br>RXDC1_0 MultiCAN Node 1 Receiver Input<br>COUT61_1 Output of Capture/Compare channel 1<br>EXF2_1 Timer 2 External Flag Output |
| P0.2 | 12/18 | | PU | $\overline{\text{CTRAP}}$_2 CCU6 Trap Input<br>TDO_0 JTAG Serial Data Output<br>TXD_1 UART Transmit Data Output/Clock Output<br>TXDC1_0 MultiCAN Node 1 Transmitter Output |
| P0.3 | 48/63 | | Hi-Z | SCK_1 SSC Clock Input/Output<br>COUT63_1 Output of Capture/Compare channel 3<br>RXDO1_0 UART1 Transmit Data Output |

**Table 3 Pin Definitions and Functions** (cont'd)

| Symbol | Pin Number (TQFP-48/64) | Type | Reset State | Function | |
|---|---|---|---|---|---|
| P0.4 | 1/64 | | Hi-Z | MTSR_1 | SSC Master Transmit Output/ Slave Receive Input |
| | | | | CC62_1 | Input/Output of Capture/Compare channel 2 |
| | | | | TXD1_0 | UART1 Transmit Data Output/Clock Output |
| P0.5 | 2/1 | | Hi-Z | MRST_1 | SSC Master Receive Input/Slave Transmit Output |
| | | | | EXINT0_0 | External Interrupt Input 0 |
| | | | | T2EX1_1 | Timer 21 External Trigger Input |
| | | | | RXD1_0 | UART1 Receive Data Input |
| | | | | COUT62_1 | Output of Capture/Compare channel 2 |
| P0.6 | –/2 | | PU | GPIO | |
| P0.7 | 47/62 | | PU | CLKOUT_1 | Clock Output |

**Table 3 Pin Definitions and Functions** (cont'd)

| Symbol | Pin Number (TQFP-48/64) | Type | Reset State | Function | |
|---|---|---|---|---|---|
| **P1** | | I/O | | **Port 1**<br>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 1, Timer 2, Timer 21, MultiCAN and SSC. | |
| P1.0 | 26/34 | | PU | RXD_0<br>T2EX<br>RXDC0_0 | UART Receive Data Input<br>Timer 2 External Trigger Input<br>MultiCAN Node 0 Receiver Input |
| P1.1 | 27/35 | | PU | EXINT3<br>T0_1<br>TDO_1<br>TXD_0<br><br>TXDC0_0 | External Interrupt Input 3<br>Timer 0 Input<br>JTAG Serial Data Output<br>UART Transmit Data Output/Clock Output<br>MultiCAN Node 0 Transmitter Output |
| P1.2 | 28/36 | | PU | SCK_0 | SSC Clock Input/Output |
| P1.3 | 29/37 | | PU | MTSR_0<br><br>TXDC1_3 | SSC Master Transmit Output/Slave Receive Input<br>MultiCAN Node 1 Transmitter Output |
| P1.4 | 30/38 | | PU | MRST_0<br><br>EXINT0_1<br>RXDC1_3 | SSC Master Receive Input/ Slave Transmit Output<br>External Interrupt Input 0<br>MultiCAN Node 1 Receiver Input |
| P1.5 | 31/39 | | PU | CCPOS0_1<br>EXINT5<br>T1_1<br>EXF2_0<br>RXDO_0 | CCU6 Hall Input 0<br>External Interrupt Input 5<br>Timer 1 Input<br>Timer 2 External Flag Output<br>UART Transmit Data Output |

**Table 3 Pin Definitions and Functions** (cont'd)

| Symbol | Pin Number (TQFP-48/64) | Type | Reset State | Function |
|---|---|---|---|---|
| P1.6 | 8/10 | | PU | CCPOS1_1 CCU6 Hall Input 1<br>T12HR_0 CCU6 Timer 12 Hardware Run Input<br>EXINT6_0 External Interrupt Input 6<br>RXDC0_2 MultiCAN Node 0 Receiver Input<br>T21_1 Timer 21 Input |
| P1.7 | 9/11 | | PU | CCPOS2_1 CCU6 Hall Input 2<br>T13HR_0 CCU6 Timer 13 Hardware Run Input<br>T2_1 Timer 2 Input<br>TXDC0_2 MultiCAN Node 0 Transmitter Output<br>P1.5 and P1.6 can be used as a software chip select output for the SSC. |

**Table 3 Pin Definitions and Functions** (cont'd)

| Symbol | Pin Number (TQFP-48/64) | Type | Reset State | Function |
|---|---|---|---|---|
| **P2** | | I | | **Port 2**<br>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 | 14/22 | | Hi-Z | CCPOS0_0 CCU6 Hall Input 0<br>EXINT1_0 External Interrupt Input 1<br>T12HR_2 CCU6 Timer 12 Hardware Run Input<br>TCK_1 JTAG Clock Input<br>CC61_3 Input of Capture/Compare channel 1<br>AN0 Analog Input 0 |
| P2.1 | 15/23 | | Hi-Z | CCPOS1_0 CCU6 Hall Input 1<br>EXINT2_0 External Interrupt Input 2<br>T13HR_2 CCU6 Timer 13 Hardware Run Input<br>TDI_1 JTAG Serial Data Input<br>CC62_3 Input of Capture/Compare channel 2<br>AN1 Analog Input 1 |
| P2.2 | 16/24 | | Hi-Z | CCPOS2_0 CCU6 Hall Input 2<br>CTRAP_1 CCU6 Trap Input<br>CC60_3 Input of Capture/Compare channel 0<br>AN2 Analog Input 2 |
| P2.3 | 19/27 | | Hi-Z | AN3 Analog Input 3 |
| P2.4 | 20/28 | | Hi-Z | AN4 Analog Input 4 |
| P2.5 | 21/29 | | Hi-Z | AN5 Analog Input 5 |
| P2.6 | 22/30 | | Hi-Z | AN6 Analog Input 6 |
| P2.7 | 25/33 | | Hi-Z | AN7 Analog Input 7 |

**Table 3 Pin Definitions and Functions** (cont'd)

| Symbol | Pin Number (TQFP-48/64) | Type | Reset State | Function | |
|---|---|---|---|---|---|
| **P3** | | I/O | | **Port 3**<br>Port 3 is an 8-bit bidirectional general purpose I/O port. It can be used as alternate functions for CCU6, UART1, Timer 21 and MultiCAN. | |
| P3.0 | 35/43 | | Hi-Z | CCPOS1_2<br>CC60_0<br><br>RXDO1_1 | CCU6 Hall Input 1<br>Input/Output of Capture/Compare channel 0<br>UART1 Transmit Data Output |
| P3.1 | 36/44 | | Hi-Z | CCPOS0_2<br>CC61_2<br><br>COUT60_0<br><br>TXD1_1 | CCU6 Hall Input 0<br>Input/Output of Capture/Compare channel 1<br>Output of Capture/Compare channel 0<br>UART1 Transmit Data Output/Clock Output |
| P3.2 | 37/49 | | Hi-Z | CCPOS2_2<br>RXDC1_1<br>RXD1_1<br>CC61_0 | CCU6 Hall Input 2<br>MultiCAN Node 1 Receiver Input<br>UART1 Receive Data Input<br>Input/Output of Capture/Compare channel 1 |
| P3.3 | 38/50 | | Hi-Z | COUT61_0<br><br>TXDC1_1 | Output of Capture/Compare channel 1<br>MultiCAN Node 1 Transmitter Output |
| P3.4 | 39/51 | | Hi-Z | CC62_0<br><br>RXDC0_1<br>T2EX1_0 | Input/Output of Capture/Compare channel 2<br>MultiCAN Node 0 Receiver Input<br>Timer 21 External Trigger Input |
| P3.5 | 40/52 | | Hi-Z | COUT62_0<br><br>EXF21_0<br>TXDC0_1 | Output of Capture/Compare channel 2<br>Timer 21 External Flag Output<br>MultiCAN Node 0 Transmitter Output |
| P3.6 | 33/41 | | PD | $\overline{\text{CTRAP_0}}$ | CCU6 Trap Input |

**Table 3 Pin Definitions and Functions** (cont'd)

| Symbol | Pin Number (TQFP-48/64) | Type | Reset State | Function | |
|---|---|---|---|---|---|
| P3.7 | 34/42 | | Hi-Z | EXINT4<br>COUT63_0 | External Interrupt Input 4<br>Output of Capture/Compare channel 3 |

**Table 3 Pin Definitions and Functions** (cont'd)

| Symbol | Pin Number (TQFP-48/64) | Type | Reset State | Function | |
|---|---|---|---|---|---|
| **P4** | | I/O | | **Port 4**<br>Port 4 is an 8-bit bidirectional general purpose I/O port. It can be used as alternate functions for CCU6, Timer 0, Timer 1, Timer 21 and MultiCAN. | |
| P4.0 | 45/59 | | Hi-Z | RXDC0_3<br>CC60_1 | MultiCAN Node 0 Receiver Input<br>Output of Capture/Compare channel 0 |
| P4.1 | 46/60 | | Hi-Z | TXDC0_3<br><br>COUT60_1 | MultiCAN Node 0 Transmitter Output<br><br>Output of Capture/Compare channel 0 |
| P4.2 | –/61 | | PU | EXINT6_1<br>T21_0 | External Interrupt Input 6<br>Timer 21 Input |
| P4.3 | 32/40 | | Hi-Z | EXF21_1<br>COUT63_2 | Timer 21 External Flag Output<br>Output of Capture/Compare channel 3 |
| P4.4 | –/45 | | Hi-Z | CCPOS0_3<br>T0_0<br>CC61_4 | CCU6 Hall Input 0<br>Timer 0 Input<br>Output of Capture/Compare channel 1 |
| P4.5 | –/46 | | Hi-Z | CCPOS1_3<br>T1_0<br>COUT61_2 | CCU6 Hall Input 1<br>Timer 1 Input<br>Output of Capture/Compare channel 1 |
| P4.6 | –/47 | | Hi-Z | CCPOS2_3<br>T2_0<br>CC62_2 | CCU6 Hall Input 2<br>Timer 2 Input<br>Output of Capture/Compare channel 2 |
| P4.7 | –/48 | | Hi-Z | $\overline{\text{CTRAP}}$_3<br>COUT62_2 | CCU6 Trap Input<br>Output of Capture/Compare channel 2 |

**Table 3 Pin Definitions and Functions** (cont'd)

| Symbol | Pin Number (TQFP-48/64) | Type | Reset State | Function | |
|---|---|---|---|---|---|
| **P5** | | I/O | | **Port 5**<br>Port 5 is an 8-bit bidirectional general purpose I/O port. It can be used as alternate functions for UART, UART1 and JTAG. | |
| P5.0 | –/8 | | PU | EXINT1_1 | External Interrupt Input 1 |
| P5.1 | –/9 | | PU | EXINT2_1 | External Interrupt Input 2 |
| P5.2 | –/12 | | PU | RXD_2 | UART Receive Data Input |
| P5.3 | –/13 | | PU | TXD_2 | UART Transmit Data Output/Clock Output |
| P5.4 | –/14 | | PU | RXDO_2 | UART Transmit Data Output |
| P5.5 | –/15 | | PU | TDO_2<br>TXD1_2 | JTAG Serial Data Output<br>UART1 Transmit Data Output/ Clock Output |
| P5.6 | –/19 | | PU | TCK_2<br>RXDO1_2 | JTAG Clock Input<br>UART1 Transmit Data Output |
| P5.7 | –/20 | | PU | TDI_2<br>RXD1_2 | JTAG Serial Data Input<br>UART1 Receive Data Input |

**Table 3 Pin Definitions and Functions** (cont'd)

| Symbol | Pin Number (TQFP-48/64) | Type | Reset State | Function |
|---|---|---|---|---|
| $V_{DDP}$ | 7, 17, 43/ 7, 25, 55 | – | – | **I/O Port Supply (3.3 or 5.0 V)** Also used by EVR and analog modules. All pins must be connected. |
| $V_{SSP}$ | 18, 42/26, 54 | – | – | **I/O Port Ground** All pins must be connected. |
| $V_{DDC}$ | 6/6 | – | – | **Core Supply Monitor (2.5 V)** |
| $V_{SSC}$ | 5/5 | – | – | **Core Supply Ground** |
| $V_{AREF}$ | 24/32 | – | – | **ADC Reference Voltage** |
| $V_{AGND}$ | 23/31 | – | – | **ADC Reference Ground** |
| **XTAL1** | 4/4 | I | Hi-Z | **External Oscillator Input (backup for on-chip OSC, normally NC)** |
| **XTAL2** | 3/3 | O | Hi-Z | **External Oscillator Output (backup for on-chip OSC, normally NC)** |
| **TMS** | 10/16 | I | PD | **Test Mode Select** |
| **RESET** | 41/53 | I | PU | **Reset Input** |
| **MBC**[1)] | 44/58 | I | PU | **Monitor & BootStrap Loader Control** |
| **NC** | –/56, 57 | – | – | **No Connection** |

1) An external pull-up device in the range of 4.7 kΩ to 100 kΩ. is required to enter user mode. Alternatively MBC can be tied to high if alternate functions (for debugging) of the pin are not utilized.

# 3 Functional Description

**Chapter 3** provides an overview of the XC886/888 functional description.

## 3.1 Processor Architecture

The XC886/888 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 XC886/888 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 XC886/888 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 Special Function Registers (SFRs).

**Figure 6** shows the CPU functional blocks.

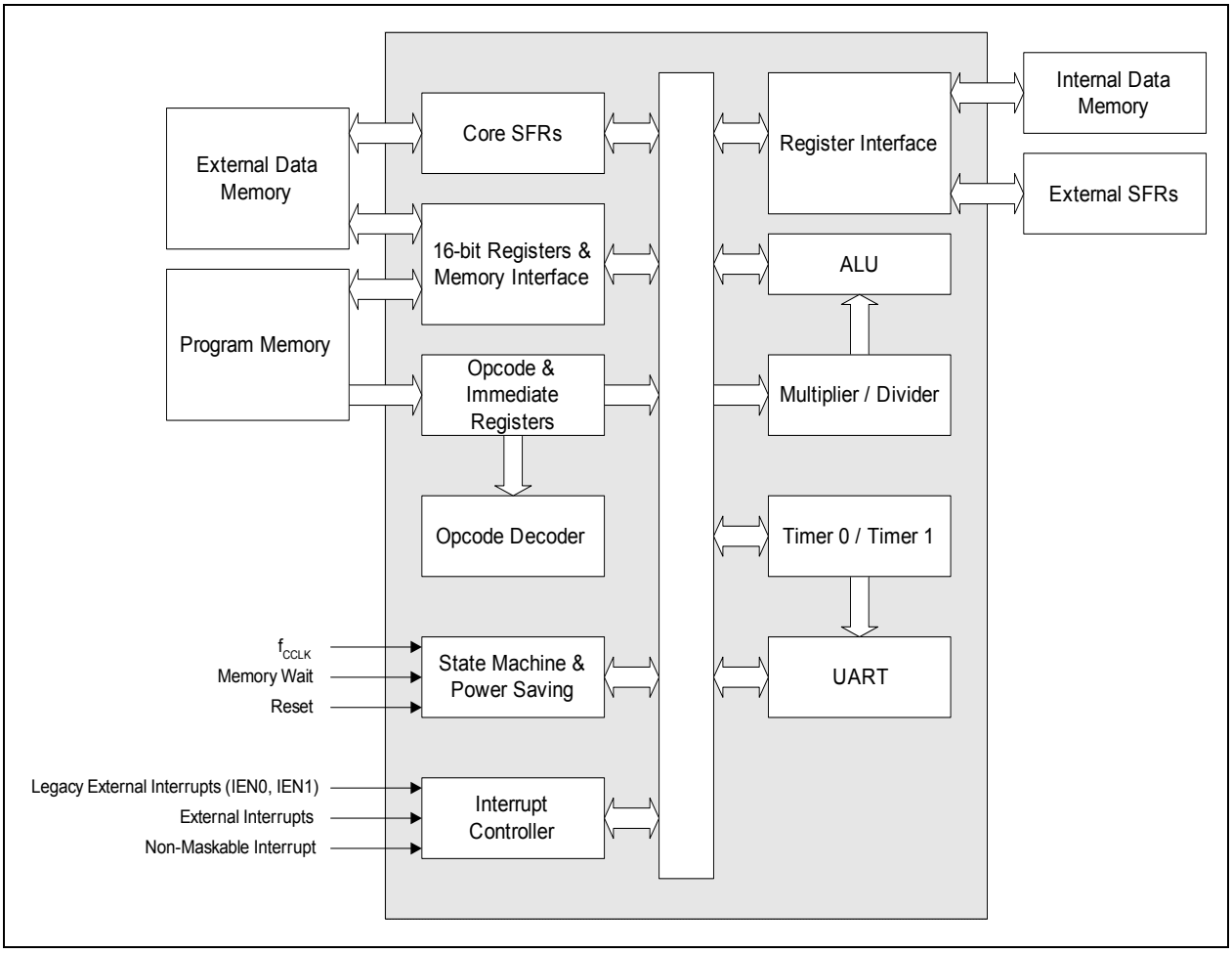



**Figure 6 CPU Block Diagram**

## 3.2 Memory Organization

The XC886/888 CPU operates in the following five address spaces:

- 12 Kbytes of Boot ROM program memory
- 256 bytes of internal RAM data memory
- 1.5 Kbytes of XRAM memory
  (XRAM can be read/written as program memory or external data memory)
- A 128-byte Special Function Register area
- 24/32 Kbytes of Flash program memory (Flash devices); or
  24/32 Kbytes of ROM program memory, with additional 4 Kbytes of Flash
  (ROM devices)

**Figure 7** illustrates the memory address spaces of the 32-Kbyte Flash devices. For the 24-Kbyte Flash devices, the shaded banks are not available.

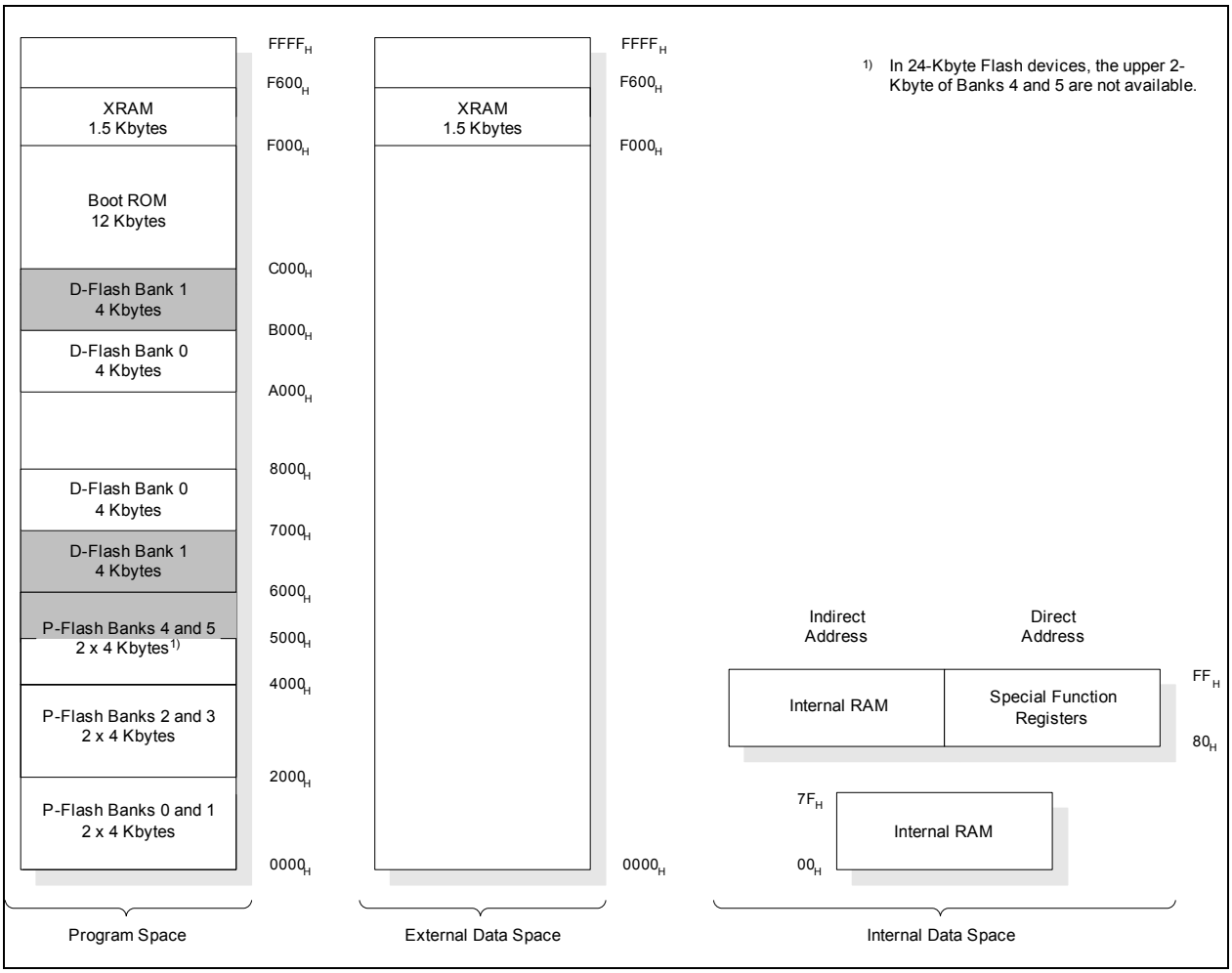



**Figure 7 Memory Map of XC886/888 Flash Device**

For both 24-Kbyte and 32-Kbyte ROM devices, the last four bytes of the ROM from $7FFC_H$ to $7FFF_H$ are reserved for the ROM signature and cannot be used to store user

code or data. Therefore, even though the ROM device contains either a 24-Kbyte or 32-Kbyte ROM, the maximum size of code that can be placed in the ROM is the given size less four bytes.

### 3.2.1 Memory Protection Strategy

The XC886/888 memory protection strategy includes:

- Read-out protection: The user is able to protect the contents in the Flash (for Flash devices) and ROM (for ROM devices) memory from being read
  - Flash protection is enabled by programming a valid password (8-bit non-zero value) via BSL mode 6.
  - ROM protection is fixed with the ROM mask and is always enabled.
- Flash program and erase protection: This feature is available only for Flash devices.

#### 3.2.1.1 Flash Memory Protection

As long as a valid password is available, all external access to the device, including the Flash, will be blocked.

For additional security, the Flash hardware protection can be enabled to implement a second layer of read-out protection, as well as to enable program and erase protection.

Flash hardware protection is available only for Flash devices and comes in two modes:

- Mode 0: Only the P-Flash is protected; the D-Flash is unprotected
- Mode 1: Both the P-Flash and D-Flash are protected

The selection of each protection mode and the restrictions imposed are summarized in Table 4.

**Table 4 Flash Protection Modes**

| Flash Protection | Without hardware protection | With hardware protection | |
|---|---|---|---|
| **Hardware Protection Mode** | - | 0 | 1 |
| **Activation** | Program a valid password via BSL mode 6 | | |
| **Selection** | Bit 4 of password = 0 | Bit 4 of password = 1<br>MSB of password = 0 | Bit 4 of password = 1<br>MSB of password = 1 |
| **P-Flash contents can be read by** | Read instructions in any program memory | Read instructions in the P-Flash | Read instructions in the P-Flash or D-Flash |
| **External access to P-Flash** | Not possible | Not possible | Not possible |

**Table 4 Flash Protection Modes** (cont'd)

| Flash Protection | Without hardware protection | With hardware protection | |
|---|---|---|---|
| **P-Flash program and erase** | Possible | Not possible | Not possible |
| **D-Flash contents can be read by** | Read instructions in any program memory | Read instructions in any program memory | Read instructions in the P-Flash or D-Flash |
| **External access to D-Flash** | Not possible | Not possible | Not possible |
| **D-Flash program** | Possible | Possible | Not possible |
| **D-Flash erase** | Possible | Possible, on condition that bit DFLASHEN in register MISC_CON is set to 1 prior to each erase operation | Not possible |

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 protected P-Flash and D-Flash contents, including the programmed password. The Flash protection is then disabled upon the next reset.

For the ROM device, the ROM is protected at all times and BSL mode 6 is used only to block external access to the device. However, unlike the Flash device, it is not possible to disable the memory protection of the ROM device. Here, entering BSL mode 6 will result in a protection error.

*Note: If ROM read-out protection is enabled, only read instructions in the ROM memory can target the ROM contents.*

Although no protection scheme can be considered infallible, the XC886/888 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 8**.

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.

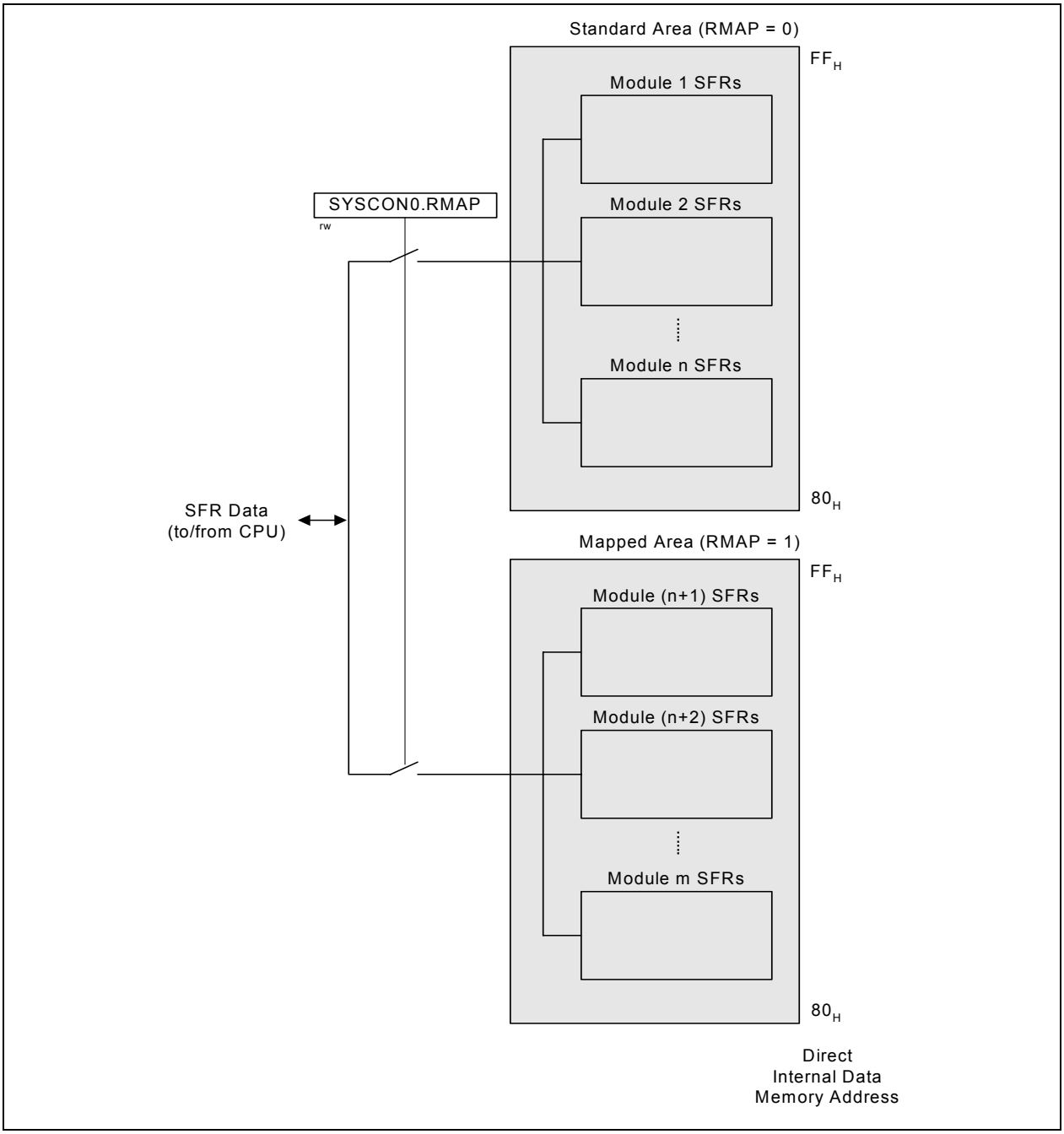



**Figure 8 Address Extension by Mapping**

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

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

| Field | Bits | Type | Description |
|---|---|---|---|
| **RMAP** | 0 | rw | **Interrupt Node XINTR0 Enable**<br>0 The access to the standard SFR area is enabled<br>1 The access to the mapped SFR area is enabled |
| **1** | 2 | r | **Reserved**<br>Returns 1 if read; should be written with 1. |
| **0** | [7:5], 3,1 | r | **Reserved**<br>Returns 0 if read; should be written with 0. |

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

### 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 XC886/888 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 9**.

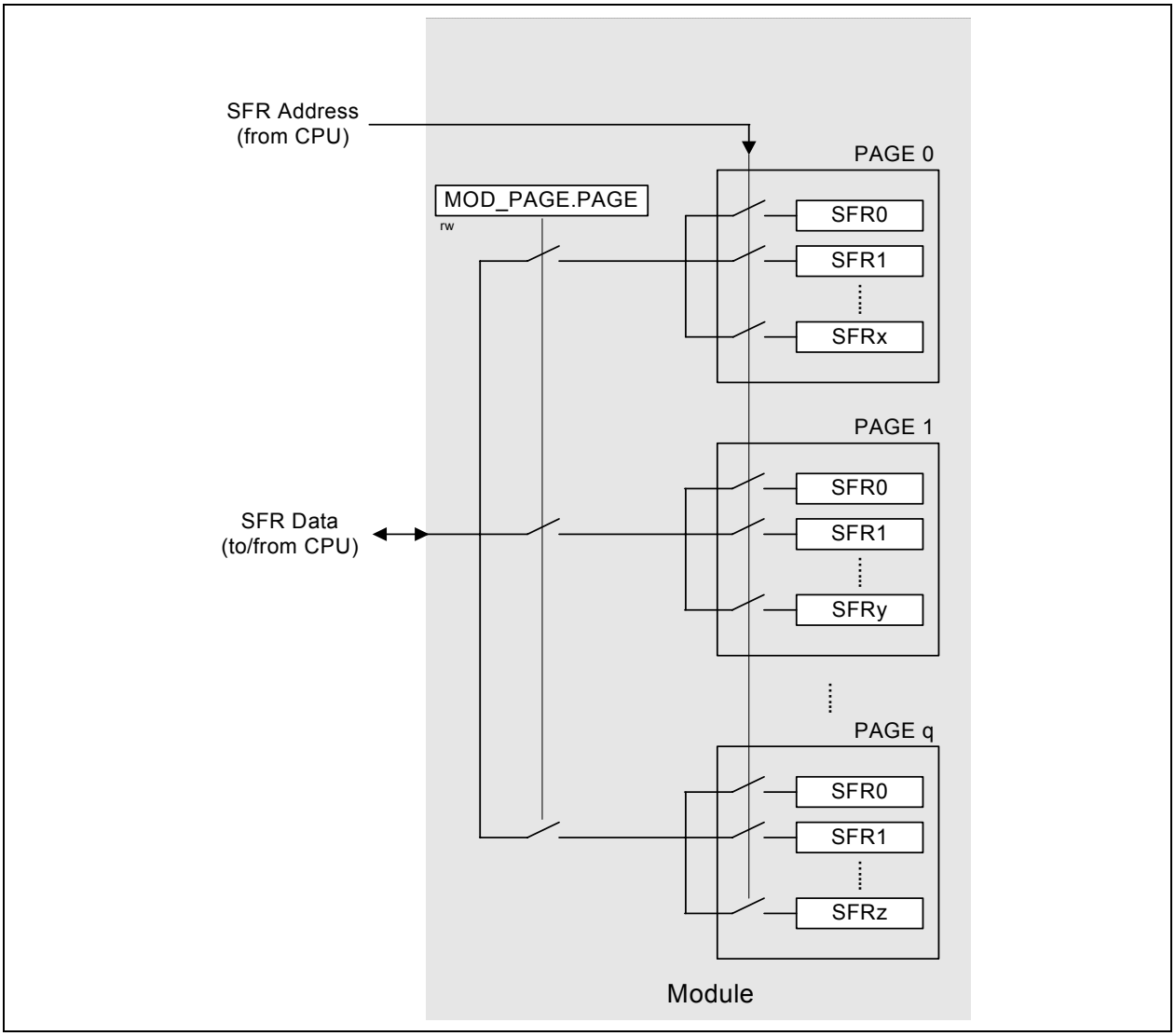



**Figure 9 Address Extension by Paging**

In order to access a register located in a page different from the actual one, the current page must be exited. 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 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)

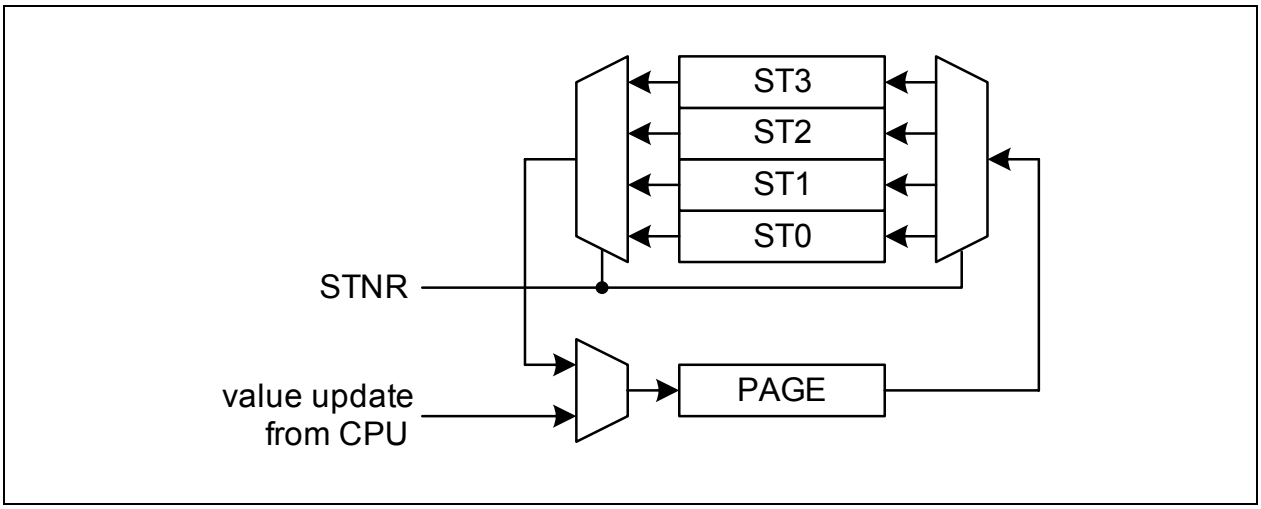



**Figure 10 Storage Elements for Paging**

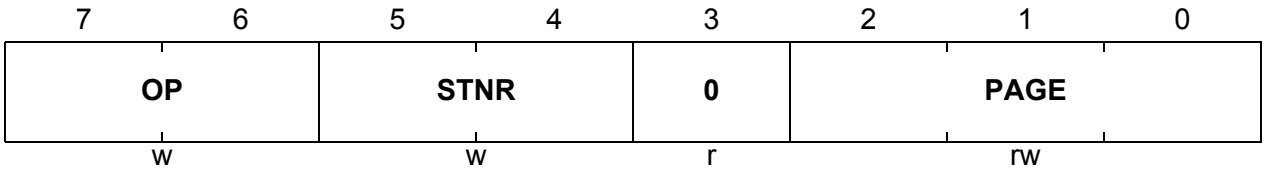
With this mechanism, a certain number of interrupt routines (or other routines) can perform page changes without reading and storing the previously used page information. The use of only write operations makes the system simpler and faster. Consequently, this mechanism significantly improves the performance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

The XC886/888 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] | rw | **Page Bits**<br>When written, the value indicates the new page.<br>When read, the value indicates the currently active page. |
| **STNR** | [5:4] | w | **Storage Number**<br>This number indicates which storage bit field is the target of the operation defined by bit field OP.<br>If OP = $10_B$,<br>the contents of PAGE are saved in STx before being overwritten with the new value.<br>If OP = $11_B$,<br>the contents of PAGE are overwritten by the contents of STx. The value written to the bit positions of PAGE is ignored.<br><br>00 ST0 is selected.<br>01 ST1 is selected.<br>10 ST2 is selected.<br>11 ST3 is selected. |

| Field | Bits | Type | Description |
|---|---|---|---|
| **OP** | [7:6] | w | **Operation**<br>0X Manual page mode. The value of STNR is ignored and PAGE is directly written.<br>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.<br>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. |
| **0** | 3 | r | **Reserved**<br>Returns 0 if read; should be written with 0. |

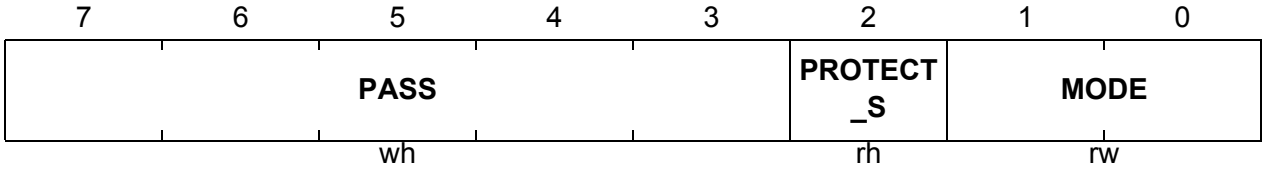
### 3.2.3 Bit Protection Scheme

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. In both cases, the value of the bit field MODE is not changed even if PASSWD register is written with $98_H$ or $A8_H$. It can only be changed when bit field PASS is written with $11000_B$, for example, writing $D0_H$ to PASSWD register disables the bit protection scheme.

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 the N- and K-Divider bits, NDIV and KDIV; the Watchdog Timer enable bit, WDTEN; and the power-down and slow-down enable bits, PD and SD.

### 3.2.3.1 Password Register

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




| Field | Bits | Type | Description |
|---|---|---|---|
| **MODE** | [1:0] | rw | **Bit Protection Scheme Control Bits**<br>00 Scheme disabled - direct access to the protected bits is allowed.<br>11 Scheme enabled - the bit field PASS has to be written with the passwords to open and close the access to protected bits. (default)<br>Others:Scheme Enabled.<br><br>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**<br>This bit shows the status of the protection.<br>0 Software is able to write to all protected bits.<br>1 Software is unable to write to any protected bits. |
| **PASS** | [7:3] | wh | **Password Bits**<br>The Bit Protection Scheme only recognizes three patterns.<br>$11000_B$ Enables writing of the bit field MODE.<br>$10011_B$ Opens access to writing of all protected bits.<br>$10101_B$ Closes access to writing of all protected bits |

### 3.2.4 XC886/888 Register Overview

The SFRs of the XC886/888 are organized into groups according to their functional units. The contents (bits) of the SFRs are summarized in **Chapter 3.2.4.1** to **Chapter 3.2.4.14**.

*Note: The addresses of the bitaddressable SFRs appear in bold typeface.*

#### 3.2.4.1 CPU Registers

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

**Table 5 CPU Register Overview**

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| RMAP = 0 or 1 | | | | | | | | | | |
| $81_H$ | **SP** **Reset: $07_H$**<br>Stack Pointer Register | Bit Field | SP | | | | | | | |
| | | Type | rw | | | | | | | |
| $82_H$ | **DPL** **Reset: $00_H$**<br>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: $00_H$**<br>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: $00_H$**<br>Power Control Register | Bit Field | SMOD | 0 | | | GF1 | GF0 | 0 | IDLE |
| | | Type | rw | r | | | rw | rw | r | rw |
| **$88_H$** | **TCON** **Reset: $00_H$**<br>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: $00_H$**<br>Timer Mode Register | Bit Field | GATE 1 | T1S | T1M | | GATE 0 | T0S | T0M | |
| | | Type | rw | rw | rw | | rw | rw | rw | |
| $8A_H$ | **TL0** **Reset: $00_H$**<br>Timer 0 Register Low | Bit Field | VAL | | | | | | | |
| | | Type | rwh | | | | | | | |
| $8B_H$ | **TL1** **Reset: $00_H$**<br>Timer 1 Register Low | Bit Field | VAL | | | | | | | |
| | | Type | rwh | | | | | | | |
| $8C_H$ | **TH0** **Reset: $00_H$**<br>Timer 0 Register High | Bit Field | VAL | | | | | | | |
| | | Type | rwh | | | | | | | |
| $8D_H$ | **TH1** **Reset: $00_H$**<br>Timer 1 Register High | Bit Field | VAL | | | | | | | |
| | | Type | rwh | | | | | | | |
| **$98_H$** | **SCON** **Reset: $00_H$**<br>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: $00_H$**<br>Serial Data Buffer Register | Bit Field | VAL | | | | | | | |
| | | Type | rwh | | | | | | | |
| $A2_H$ | **EO** **Reset: $00_H$**<br>Extended Operation Register | Bit Field | 0 | | | TRAP_EN | 0 | | | DPSEL0 |
| | | Type | r | | | rw | r | | | rw |

**Table 5 CPU Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $A8_H$ | **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 High Register | 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 |
| $E0_H$ | **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 1 Register | 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 1 High Register | Bit Field | PCCIP 3H | PCCIP 2H | PCCIP 1H | PCCIP 0H | PXMH | PX2H | PSSC H | PADC H |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |

### 3.2.4.2 MDU Registers

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

**Table 6 MDU Register Overview**

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| RMAP = 1 | | | | | | | | | | |
| $B0_H$ | **MDUSTAT** **Reset: $00_H$** MDU Status Register | Bit Field | 0 | | | | | BSY | IERR | IRDY |
| | | Type | r | | | | | rh | rwh | rwh |
| $B1_H$ | **MDUCON** **Reset: $00_H$** MDU Control Register | Bit Field | IE | IR | RSEL | START | OPCODE | | | |
| | | Type | rw | rw | rw | rwh | rw | | | |
| $B2_H$ | **MD0** **Reset: $00_H$** MDU Operand Register 0 | Bit Field | DATA | | | | | | | |
| | | Type | rw | | | | | | | |
| $B2_H$ | **MR0** **Reset: $00_H$** MDU Result Register 0 | Bit Field | DATA | | | | | | | |
| | | Type | rh | | | | | | | |
| $B3_H$ | **MD1** **Reset: $00_H$** MDU Operand Register 1 | Bit Field | DATA | | | | | | | |
| | | Type | rw | | | | | | | |

**Table 6 MDU Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $B3_H$ | **MR1** **Reset: $00_H$** MDU Result Register 1 | Bit Field | DATA | | | | | | | |
| | | Type | rh | | | | | | | |
| $B4_H$ | **MD2** **Reset: $00_H$** MDU Operand Register 2 | Bit Field | DATA | | | | | | | |
| | | Type | rw | | | | | | | |
| $B4_H$ | **MR2** **Reset: $00_H$** MDU Result Register 2 | Bit Field | DATA | | | | | | | |
| | | Type | rh | | | | | | | |
| $B5_H$ | **MD3** **Reset: $00_H$** MDU Operand Register 3 | Bit Field | DATA | | | | | | | |
| | | Type | rw | | | | | | | |
| $B5_H$ | **MR3** **Reset: $00_H$** MDU Result Register 3 | Bit Field | DATA | | | | | | | |
| | | Type | rh | | | | | | | |
| $B6_H$ | **MD4** **Reset: $00_H$** MDU Operand Register 4 | Bit Field | DATA | | | | | | | |
| | | Type | rw | | | | | | | |
| $B6_H$ | **MR4** **Reset: $00_H$** MDU Result Register 4 | Bit Field | DATA | | | | | | | |
| | | Type | rh | | | | | | | |
| $B7_H$ | **MD5** **Reset: $00_H$** MDU Operand Register 5 | Bit Field | DATA | | | | | | | |
| | | Type | rw | | | | | | | |
| $B7_H$ | **MR5** **Reset: $00_H$** MDU Result Register 5 | Bit Field | DATA | | | | | | | |
| | | Type | rh | | | | | | | |

### 3.2.4.3 CORDIC Registers

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

**Table 7 CORDIC Register Overview**

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| RMAP = 1 | | | | | | | | | | |
| $9A_H$ | **CD_CORDXL** **Reset: $00_H$** CORDIC X Data Low Byte | Bit Field | DATAL | | | | | | | |
| | | Type | rw | | | | | | | |
| $9B_H$ | **CD_CORDXH** **Reset: $00_H$** CORDIC X Data High Byte | Bit Field | DATAH | | | | | | | |
| | | Type | rw | | | | | | | |
| $9C_H$ | **CD_CORDYL** **Reset: $00_H$** CORDIC Y Data Low Byte | Bit Field | DATAL | | | | | | | |
| | | Type | rw | | | | | | | |
| $9D_H$ | **CD_CORDYH** **Reset: $00_H$** CORDIC Y Data High Byte | Bit Field | DATAH | | | | | | | |
| | | Type | rw | | | | | | | |
| $9E_H$ | **CD_CORDZL** **Reset: $00_H$** CORDIC Z Data Low Byte | Bit Field | DATAL | | | | | | | |
| | | Type | rw | | | | | | | |
| $9F_H$ | **CD_CORDZH** **Reset: $00_H$** CORDIC Z Data High Byte | Bit Field | DATAH | | | | | | | |
| | | Type | rw | | | | | | | |

**Table 7 CORDIC Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $A0_H$ | **CD_STATC** **Reset: $00_H$** CORDIC Status and Data Control Register | Bit Field | KEEPZ | KEEPY | KEEPX | DMAP | INT_EN | EOC | ERROR | BSY |
| | | Type | rw | rw | rw | rw | rw | rwh | rh | rh |
| $A1_H$ | **CD_CON** **Reset: $00_H$** CORDIC Control Register | Bit Field | MPS | | X_USIGN | ST_MODE | ROTVEC | MODE | | ST |
| | | Type | rw | | rw | rw | rw | rw | | rwh |

### 3.2.4.4 System Control Registers

The system control SFRs can be accessed in the mapped memory area (RMAP = 0).

**Table 8 SCU Register Overview**

| 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 | | | IMODE | 0 | 1 | 0 | RMAP |
| | | Type | r | | | rw | r | r | r | rw |
| RMAP = 0 | | | | | | | | | | |
| $BF_H$ | **SCU_PAGE** **Reset: $00_H$** Page Register | Bit Field | OP | | STNR | | 0 | PAGE | | |
| | | Type | w | | w | | r | rw | | |
| RMAP = 0, PAGE 0 | | | | | | | | | | |
| $B3_H$ | **MODPISEL** **Reset: $00_H$** Peripheral Input Select Register | Bit Field | 0 | URRISH | JTAGTDIS | JTAGTCKS | EXINT2IS | EXINT1IS | EXINT0IS | URRIS |
| | | Type | r | rw | rw | rw | rw | rw | rw | rw |
| $B4_H$ | **IRCON0** **Reset: $00_H$** Interrupt Request Register 0 | Bit Field | 0 | EXINT6 | EXINT5 | EXINT4 | EXINT3 | EXINT2 | EXINT1 | EXINT0 |
| | | Type | r | rwh | rwh | rwh | rwh | rwh | rwh | rwh |
| $B5_H$ | **IRCON1** **Reset: $00_H$** Interrupt Request Register 1 | Bit Field | 0 | CANSRC2 | CANSRC1 | ADCSR1 | ADCSR0 | RIR | TIR | EIR |
| | | Type | r | rwh | rwh | rwh | rwh | rwh | rwh | rwh |
| $B6_H$ | **IRCON2** **Reset: $00_H$** Interrupt Request Register 2 | Bit Field | 0 | | | CANSRC3 | 0 | | | CANSRC0 |
| | | Type | r | | | rwh | r | | | rwh |
| $B7_H$ | **EXICON0** **Reset: $F0_H$** External Interrupt Control Register 0 | Bit Field | EXINT3 | | EXINT2 | | EXINT1 | | EXINT0 | |
| | | Type | rw | | rw | | rw | | rw | |
| $BA_H$ | **EXICON1** **Reset: $3F_H$** External Interrupt Control Register 1 | Bit Field | 0 | | EXINT6 | | EXINT5 | | EXINT4 | |
| | | Type | r | | rw | | rw | | rw | |
| $BB_H$ | **NMICON** **Reset: $00_H$** NMI Control Register | Bit Field | 0 | NMIECC | NMIVDDP | NMIVDD | NMIOCDS | NMIFLASH | NMIPLL | NMIWDT |
| | | Type | r | rw | rw | rw | rw | rw | rw | rw |

**Table 8 SCU Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $BC_H$ | **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 | | | | | | | |
| $E9_H$ | **FDCON** **Reset: $00_H$** Fractional Divider Control Register | Bit Field | BGS | SYNEN | ERRSYN | EOFSYN | BRK | 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: $UU_H$** Identity Register | Bit Field | PRODID | | | | | VERID | | |
| | | Type | r | | | | | r | | |
| $B4_H$ | **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 | CDC_DIS | CAN_DIS | MDU_DIS | T2_DIS | CCU_DIS | SSC_DIS | ADC_DIS |
| | | Type | r | rw | rw | rw | rw | rw | rw | rw |
| $B6_H$ | **OSC_CON** **Reset: $08_H$** OSC Control Register | Bit Field | 0 | | | OSCPD | XPD | OSCSS | ORDRES | OSCR |
| | | Type | r | | | rw | rw | rw | rwh | rh |
| $B7_H$ | **PLL_CON** **Reset: $90_H$** PLL Control Register | Bit Field | NDIV | | | | VCOBYP | OSCDISC | RESLD | LOCK |
| | | Type | rw | | | | rw | rw | rwh | rh |
| $BA_H$ | **CMCON** **Reset: $10_H$** Clock Control Register | Bit Field | VCOSEL | KDIV | 0 | FCCFG | CLKREL | | | |
| | | Type | rw | rw | r | rw | 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 | | | | | | | |
| | | Type | rh | | | | | | | |
| $BD_H$ | **FEAH** **Reset: $00_H$** Flash Error Address Register High | Bit Field | ECCERRADDR | | | | | | | |
| | | Type | rh | | | | | | | |

**Table 8 SCU Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $BE_H$ | **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 | | CANSRC5 | CCU6SR1 | 0 | | CANSRC4 | CCU6SR0 |
| | | Type | r | | rwh | rwh | r | | rwh | rwh |
| $B5_H$ | **IRCON4** **Reset: $00_H$** Interrupt Request Register 4 | Bit Field | 0 | | CANSRC7 | CCU6SR3 | 0 | | CANSRC6 | CCU6SR2 |
| | | Type | r | | rwh | rwh | r | | rwh | rwh |
| $B7_H$ | **MODPISEL1** **Reset: $00_H$** Peripheral Input Select Register 1 | Bit Field | EXINT6IS | 0 | | UR1RIS | | T21EXIS | JTAGTDIS1 | JTAGTCKS1 |
| | | Type | rw | r | | rw | | rw | rw | rw |
| $BA_H$ | **MODPISEL2** **Reset: $00_H$** Peripheral Input Select Register 2 | Bit Field | 0 | | | | T21IS | T2IS | T1IS | T0IS |
| | | Type | r | | | | rw | rw | rw | rw |
| $BB_H$ | **PMCON2** **Reset: $00_H$** Power Mode Control Register 2 | Bit Field | 0 | | | | | | UART1_DIS | T21_DIS |
| | | Type | r | | | | | | rw | rw |
| $BD_H$ | **MODSUSP** **Reset: $01_H$** Module Suspend Control Register | Bit Field | 0 | | | T21SUSP | T2SUSP | T13SUSP | T12SUSP | WDTSUSP |
| | | Type | r | | | rw | rw | rw | rw | rw |

### 3.2.4.5 WDT Registers

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

**Table 9 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 | | WINBEN | WDTPR | 0 | WDTEN | WDTRS | WDTIN |
| | | 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 | | | | | | | |

**Table 9 WDT Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $BE_H$ | **WDTL** **Reset: $00_H$** Watchdog Timer Register Low | Bit Field | WDT | | | | | | | |
| | | Type | rh | | | | | | | |
| $BF_H$ | **WDTH** **Reset: $00_H$** Watchdog Timer Register High | Bit Field | WDT | | | | | | | |
| | | Type | rh | | | | | | | |

### 3.2.4.6 Port Registers

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

**Table 10 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 | Bit Field | OP | | STNR | | 0 | PAGE | | |
| | | Type | w | | w | | r | rw | | |
| RMAP = 0, PAGE 0 | | | | | | | | | | |
| **$80_H$** | **P0_DATA** **Reset: $00_H$** P0 Data Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $86_H$ | **P0_DIR** **Reset: $00_H$** P0 Direction Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| **$90_H$** | **P1_DATA** **Reset: $00_H$** P1 Data Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $91_H$ | **P1_DIR** **Reset: $00_H$** P1 Direction Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $92_H$ | **P5_DATA** **Reset: $00_H$** P5 Data Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $93_H$ | **P5_DIR** **Reset: $00_H$** P5 Direction Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| **$A0_H$** | **P2_DATA** **Reset: $00_H$** P2 Data Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $A1_H$ | **P2_DIR** **Reset: $00_H$** P2 Direction Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| **$B0_H$** | **P3_DATA** **Reset: $00_H$** P3 Data Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $B1_H$ | **P3_DIR** **Reset: $00_H$** P3 Direction Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| **$C8_H$** | **P4_DATA** **Reset: $00_H$** P4 Data Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $C9_H$ | **P4_DIR** **Reset: $00_H$** P4 Direction Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |

**Table 10 Port Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| RMAP = 0, PAGE 1 | | | | | | | | | | |
| **$80_H$** | **P0_PUDSEL Reset: $FF_H$**<br>P0 Pull-Up/Pull-Down Select Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $86_H$ | **P0_PUDEN Reset: $C4_H$**<br>P0 Pull-Up/Pull-Down Enable Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| **$90_H$** | **P1_PUDSEL Reset: $FF_H$**<br>P1 Pull-Up/Pull-Down Select Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $91_H$ | **P1_PUDEN Reset: $FF_H$**<br>P1 Pull-Up/Pull-Down Enable Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $92_H$ | **P5_PUDSEL Reset: $FF_H$**<br>P5 Pull-Up/Pull-Down Select Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $93_H$ | **P5_PUDEN Reset: $FF_H$**<br>P5 Pull-Up/Pull-Down Enable Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| **$A0_H$** | **P2_PUDSEL Reset: $FF_H$**<br>P2 Pull-Up/Pull-Down Select Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $A1_H$ | **P2_PUDEN Reset: $00_H$**<br>P2 Pull-Up/Pull-Down Enable Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| **$B0_H$** | **P3_PUDSEL Reset: $BF_H$**<br>P3 Pull-Up/Pull-Down Select Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $B1_H$ | **P3_PUDEN Reset: $40_H$**<br>P3 Pull-Up/Pull-Down Enable Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| **$C8_H$** | **P4_PUDSEL Reset: $FF_H$**<br>P4 Pull-Up/Pull-Down Select Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $C9_H$ | **P4_PUDEN Reset: $04_H$**<br>P4 Pull-Up/Pull-Down Enable Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| RMAP = 0, PAGE 2 | | | | | | | | | | |
| **$80_H$** | **P0_ALTSEL0 Reset: $00_H$**<br>P0 Alternate Select 0 Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $86_H$ | **P0_ALTSEL1 Reset: $00_H$**<br>P0 Alternate Select 1 Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| **$90_H$** | **P1_ALTSEL0 Reset: $00_H$**<br>P1 Alternate Select 0 Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $91_H$ | **P1_ALTSEL1 Reset: $00_H$**<br>P1 Alternate Select 1 Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $92_H$ | **P5_ALTSEL0 Reset: $00_H$**<br>P5 Alternate Select 0 Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |

**Table 10 Port Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $93_H$ | **P5_ALTSEL1 Reset: $00_H$**<br>P5 Alternate Select 1 Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| **$B0_H$** | **P3_ALTSEL0 Reset: $00_H$**<br>P3 Alternate Select 0 Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $B1_H$ | **P3_ALTSEL1 Reset: $00_H$**<br>P3 Alternate Select 1 Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| **$C8_H$** | **P4_ALTSEL0 Reset: $00_H$**<br>P4 Alternate Select 0 Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $C9_H$ | **P4_ALTSEL1 Reset: $00_H$**<br>P4 Alternate Select 1 Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| RMAP = 0, PAGE 3 | | | | | | | | | | |
| **$80_H$** | **P0_OD Reset: $00_H$**<br>P0 Open Drain Control Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| **$90_H$** | **P1_OD Reset: $00_H$**<br>P1 Open Drain Control Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $92_H$ | **P5_OD Reset: $00_H$**<br>P5 Open Drain Control Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| **$B0_H$** | **P3_OD Reset: $00_H$**<br>P3 Open Drain Control Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| **$C8_H$** | **P4_OD Reset: $00_H$**<br>P4 Open Drain Control Register | Bit Field | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |

### 3.2.4.7 ADC Registers

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

**Table 11 ADC Register Overview**

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| RMAP = 0 | | | | | | | | | | |
| $D1_H$ | **ADC_PAGE Reset: $00_H$**<br>Page Register | Bit Field | OP | | STNR | | 0 | PAGE | | |
| | | Type | w | | w | | r | rw | | |
| RMAP = 0, PAGE 0 | | | | | | | | | | |
| $CA_H$ | **ADC_GLOBCTR Reset: $30_H$**<br>Global Control Register | Bit Field | ANON | DW | CTC | | 0 | | | |
| | | Type | rw | rw | rw | | r | | | |
| $CB_H$ | **ADC_GLOBSTR Reset: $00_H$**<br>Global Status Register | Bit Field | 0 | | CHNR | | | 0 | SAMPLE | BUSY |
| | | Type | r | | rh | | | r | rh | rh |
| $CC_H$ | **ADC_PRAR Reset: $00_H$**<br>Priority and Arbitration Register | Bit Field | ASEN1 | ASEN0 | 0 | ARBM | CSM1 | PRIO1 | CSM0 | PRIO0 |
| | | Type | rw | rw | r | rw | rw | rw | rw | rw |

**Table 11 ADC Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $CD_H$ | **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 0 Register | Bit Field | STC | | | | | | | |
| | | Type | rw | | | | | | | |
| $CF_H$ | **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 | |
| $CD_H$ | **ADC_CHCTR3 Reset: $00_H$** Channel Control Register 3 | Bit Field | 0 | LCC | | | 0 | | RESRSEL | |
| | | Type | r | rw | | | r | | rw | |
| $CE_H$ | **ADC_CHCTR4 Reset: $00_H$** Channel Control Register 4 | Bit Field | 0 | LCC | | | 0 | | RESRSEL | |
| | | Type | r | rw | | | r | | rw | |
| $CF_H$ | **ADC_CHCTR5 Reset: $00_H$** Channel Control Register 5 | Bit Field | 0 | LCC | | | 0 | | RESRSEL | |
| | | Type | r | rw | | | r | | rw | |
| $D2_H$ | **ADC_CHCTR6 Reset: $00_H$** Channel Control Register 6 | 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 | | 0 | VF | DRC | CHNR | | |
| | | Type | rh | | r | rh | rh | rh | | |
| $CB_H$ | **ADC_RESR0H Reset: $00_H$** Result Register 0 High | Bit Field | RESULT | | | | | | | |
| | | Type | rh | | | | | | | |
| $CC_H$ | **ADC_RESR1L Reset: $00_H$** Result Register 1 Low | Bit Field | RESULT | | 0 | VF | DRC | CHNR | | |
| | | Type | rh | | r | rh | rh | rh | | |
| $CD_H$ | **ADC_RESR1H Reset: $00_H$** Result Register 1 High | Bit Field | RESULT | | | | | | | |
| | | Type | rh | | | | | | | |
| $CE_H$ | **ADC_RESR2L Reset: $00_H$** Result Register 2 Low | Bit Field | RESULT | | 0 | VF | DRC | CHNR | | |
| | | Type | rh | | r | rh | rh | rh | | |
| $CF_H$ | **ADC_RESR2H Reset: $00_H$** Result Register 2 High | Bit Field | RESULT | | | | | | | |
| | | Type | rh | | | | | | | |
| $D2_H$ | **ADC_RESR3L Reset: $00_H$** Result Register 3 Low | Bit Field | RESULT | | 0 | VF | DRC | CHNR | | |
| | | Type | rh | | r | rh | rh | rh | | |

**Table 11 ADC Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $D3_H$ | **ADC_RESR3H Reset: $00_H$**<br>Result Register 3 High | Bit Field | RESULT | | | | | | | |
| | | Type | rh | | | | | | | |
| RMAP = 0, PAGE 3 | | | | | | | | | | |
| $CA_H$ | **ADC_RESRA0L Reset: $00_H$**<br>Result Register 0, View A Low | Bit Field | RESULT | | | VF | DRC | CHNR | | |
| | | Type | rh | | | rh | rh | rh | | |
| $CB_H$ | **ADC_RESRA0H Reset: $00_H$**<br>Result Register 0, View A High | Bit Field | RESULT | | | | | | | |
| | | Type | rh | | | | | | | |
| $CC_H$ | **ADC_RESRA1L Reset: $00_H$**<br>Result Register 1, View A Low | Bit Field | RESULT | | | VF | DRC | CHNR | | |
| | | Type | rh | | | rh | rh | rh | | |
| $CD_H$ | **ADC_RESRA1H Reset: $00_H$**<br>Result Register 1, View A High | Bit Field | RESULT | | | | | | | |
| | | Type | rh | | | | | | | |
| $CE_H$ | **ADC_RESRA2L Reset: $00_H$**<br>Result Register 2, View A Low | Bit Field | RESULT | | | VF | DRC | CHNR | | |
| | | Type | rh | | | rh | rh | rh | | |
| $CF_H$ | **ADC_RESRA2H Reset: $00_H$**<br>Result Register 2, View A High | Bit Field | RESULT | | | | | | | |
| | | Type | rh | | | | | | | |
| $D2_H$ | **ADC_RESRA3L Reset: $00_H$**<br>Result Register 3, View A Low | Bit Field | RESULT | | | VF | DRC | CHNR | | |
| | | Type | rh | | | rh | rh | rh | | |
| $D3_H$ | **ADC_RESRA3H Reset: $00_H$**<br>Result Register 3, View A High | Bit Field | RESULT | | | | | | | |
| | | Type | rh | | | | | | | |
| RMAP = 0, PAGE 4 | | | | | | | | | | |
| $CA_H$ | **ADC_RCR0 Reset: $00_H$**<br>Result Control Register 0 | Bit Field | VFCTR | WFR | 0 | IEN | 0 | | | DRCTR |
| | | Type | rw | rw | r | rw | r | | | rw |
| $CB_H$ | **ADC_RCR1 Reset: $00_H$**<br>Result Control Register 1 | Bit Field | VFCTR | WFR | 0 | IEN | 0 | | | DRCTR |
| | | Type | rw | rw | r | rw | r | | | rw |
| $CC_H$ | **ADC_RCR2 Reset: $00_H$**<br>Result Control Register 2 | Bit Field | VFCTR | WFR | 0 | IEN | 0 | | | DRCTR |
| | | Type | rw | rw | r | rw | r | | | rw |
| $CD_H$ | **ADC_RCR3 Reset: $00_H$**<br>Result Control Register 3 | Bit Field | VFCTR | WFR | 0 | IEN | 0 | | | DRCTR |
| | | Type | rw | rw | r | rw | r | | | rw |
| $CE_H$ | **ADC_VFCR Reset: $00_H$**<br>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: $00_H$**<br>Channel Interrupt Flag Register | Bit Field | CHINF7 | CHINF6 | CHINF5 | CHINF4 | CHINF3 | CHINF2 | CHINF1 | CHINF0 |
| | | Type | rh | rh | rh | rh | rh | rh | rh | rh |
| $CB_H$ | **ADC_CHINCR Reset: $00_H$**<br>Channel Interrupt Clear Register | Bit Field | CHINC7 | CHINC6 | CHINC5 | CHINC4 | CHINC3 | CHINC2 | CHINC1 | CHINC0 |
| | | Type | w | w | w | w | w | w | w | w |

**Table 11 ADC Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $CC_H$ | **ADC_CHINSR Reset: $00_H$** Channel Interrupt Set Register | Bit Field | CHINS 7 | CHINS 6 | CHINS 5 | CHINS 4 | CHINS 3 | CHINS 2 | CHINS 1 | CHINS 0 |
| | | Type | w | w | w | w | w | w | w | w |
| $CD_H$ | **ADC_CHINPR Reset: $00_H$** Channel Interrupt Node Pointer Register | Bit Field | CHINP 7 | CHINP 6 | CHINP 5 | CHINP 4 | CHINP 3 | CHINP 2 | CHINP 1 | CHINP 0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $CE_H$ | **ADC_EVINFR Reset: $00_H$** Event Interrupt Flag Register | Bit Field | EVINF 7 | EVINF 6 | EVINF 5 | EVINF 4 | 0 | | EVINF 1 | EVINF 0 |
| | | Type | rh | rh | rh | rh | r | | rh | rh |
| $CF_H$ | **ADC_EVINCR Reset: $00_H$** Event Interrupt Clear Flag Register | Bit Field | EVINC 7 | EVINC 6 | EVINC 5 | EVINC 4 | 0 | | EVINC 1 | EVINC 0 |
| | | Type | w | w | w | w | r | | w | w |
| $D2_H$ | **ADC_EVINSR Reset: $00_H$** Event Interrupt Set Flag Register | Bit Field | EVINS 7 | EVINS 6 | EVINS 5 | EVINS 4 | 0 | | EVINS 1 | EVINS 0 |
| | | Type | w | w | w | w | r | | w | w |
| $D3_H$ | **ADC_EVINPR Reset: $00_H$** Event Interrupt Node Pointer Register | Bit Field | EVINP 7 | EVINP 6 | EVINP 5 | EVINP 4 | 0 | | EVINP 1 | EVINP 0 |
| | | 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 | CH6 | CH5 | CH4 | 0 | | | |
| | | Type | rwh | rwh | rwh | rwh | r | | | |
| $CB_H$ | **ADC_CRPR1 Reset: $00_H$** Conversion Request Pending Register 1 | Bit Field | CHP7 | CHP6 | CHP5 | CHP4 | 0 | | | |
| | | Type | rwh | rwh | rwh | rwh | r | | | |
| $CC_H$ | **ADC_CRMR1 Reset: $00_H$** Conversion Request Mode Register 1 | Bit Field | Rsv | LDEV | CLRP ND | 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 | FLUS H | CLRV | 0 | ENTR | 0 | ENGT |
| | | Type | w | w | w | w | r | rw | r | rw |
| $CE_H$ | **ADC_QSR0 Reset: $20_H$** Queue Status Register 0 | Bit Field | Rsv | 0 | EMPT Y | EV | 0 | | FILL | |
| | | Type | r | r | rh | rh | r | | rh | |
| $CF_H$ | **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 | | |
| $D2_H$ | **ADC_QINR0 Reset: $00_H$** Queue Input Register 0 | Bit Field | EXTR | ENSI | RF | 0 | | REQCHNR | | |
| | | Type | w | w | w | r | | w | | |

### 3.2.4.8 Timer 2 Registers

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

**Table 12 T2 Register Overview**

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| RMAP = 0 | | | | | | | | | | |
| $C0_H$ | **T2_T2CON** **Reset: $00_H$** Timer 2 Control Register | Bit Field | TF2 | EXF2 | 0 | | EXEN2 | TR2 | C/$\overline{T2}$ | $\underline{CP}/\overline{RL2}$ |
| | | Type | rwh | rwh | r | | rw | rwh | rw | rw |
| $C1_H$ | **T2_T2MOD** **Reset: $00_H$** Timer 2 Mode Register | Bit Field | T2REGS | T2RHEN | EDGESEL | PREN | T2PRE | | | DCEN |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $C2_H$ | **T2_RC2L** **Reset: $00_H$** Timer 2 Reload/Capture Register Low | Bit Field | RC2 | | | | | | | |
| | | Type | rwh | | | | | | | |
| $C3_H$ | **T2_RC2H** **Reset: $00_H$** Timer 2 Reload/Capture Register High | Bit Field | RC2 | | | | | | | |
| | | Type | rwh | | | | | | | |
| $C4_H$ | **T2_T2L** **Reset: $00_H$** Timer 2 Register Low | Bit Field | THL2 | | | | | | | |
| | | Type | rwh | | | | | | | |
| $C5_H$ | **T2_T2H** **Reset: $00_H$** Timer 2 Register High | Bit Field | THL2 | | | | | | | |
| | | Type | rwh | | | | | | | |

### 3.2.4.9 Timer 21 Registers

The Timer 21 SFRs can be accessed in the mapped memory area (RMAP = 1).

**Table 13 T21 Register Overview**

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| RMAP = 1 | | | | | | | | | | |
| $C0_H$ | **T21_T2CON** **Reset: $00_H$** Timer 2 Control Register | Bit Field | TF2 | EXF2 | 0 | | EXEN2 | TR2 | C/$\overline{T2}$ | $\underline{CP}/\overline{RL2}$ |
| | | Type | rwh | rwh | r | | rw | rwh | rw | rw |
| $C1_H$ | **T21_T2MOD** **Reset: $00_H$** Timer 2 Mode Register | Bit Field | T2REGS | T2RHEN | EDGESEL | PREN | T2PRE | | | DCEN |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $C2_H$ | **T21_RC2L** **Reset: $00_H$** Timer 2 Reload/Capture Register Low | Bit Field | RC2 | | | | | | | |
| | | Type | rwh | | | | | | | |
| $C3_H$ | **T21_RC2H** **Reset: $00_H$** Timer 2 Reload/Capture Register High | Bit Field | RC2 | | | | | | | |
| | | Type | rwh | | | | | | | |
| $C4_H$ | **T21_T2L** **Reset: $00_H$** Timer 2 Register Low | Bit Field | THL2 | | | | | | | |
| | | Type | rwh | | | | | | | |

**Table 13 T21 Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $C5_H$ | **T21_T2H** **Reset: $00_H$** Timer 2 Register High | Bit Field | THL2 | | | | | | | |
| | | Type | rwh | | | | | | | |

### 3.2.4.10 CCU6 Registers

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

**Table 14 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 | Bit Field | OP | | STNR | | 0 | PAGE | | |
| | | Type | w | | w | | r | rw | | |
| 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 | | DT RES | T12 RES | T12R S | T12R R |
| | | 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 | T13R S | T13R R |
| | | 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 | STRH P | 0 | CURHS | | | EXPHS | | |
| | | Type | w | r | rw | | | rw | | |
| $A4_H$ | **CCU6_ISRL** **Reset: $00_H$** Capture/Compare Interrupt Status Reset Register Low | Bit Field | RT12 PM | RT12 OM | RCC6 2F | RCC6 2R | RCC6 1F | RCC6 1R | RCC6 0F | RCC6 0R |
| | | 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 | RWH E | RCHE | 0 | RTRP F | RT13 PM | RT13 CM |
| | | Type | w | w | w | w | r | w | w | w |
| $A6_H$ | **CCU6_CMPMODIFL** **Reset: $00_H$** Compare State Modification Register Low | Bit Field | 0 | MCC6 3S | 0 | | | MCC6 2S | MCC6 1S | MCC6 0S |
| | | Type | r | w | r | | | w | w | w |
| $A7_H$ | **CCU6_CMPMODIFH** **Reset: $00_H$** Compare State Modification Register High | Bit Field | 0 | MCC6 3R | 0 | | | MCC6 2R | MCC6 1R | MCC6 0R |
| | | Type | r | w | r | | | w | w | w |

**Table 14 CCU6 Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $FA_H$ | **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 |
| $A6_H$ | **CCU6_TCTR0L** **Reset: $00_H$** Timer Control Register 0 Low | Bit Field | CTM | CDIR | STE12 | T12R | T12PRE | 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 | T13PRE | 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 | | | | | | | |

**Table 14 CCU6 Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $FB_H$ | **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 | ENT12PM | ENT12OM | ENCC62F | ENCC62R | ENCC61F | ENCC61R | ENCC60F | ENCC60R |
| | | 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 | ENIDLE | ENWHE | ENCHE | 0 | ENTRPF | ENT13PM | ENT13CM |
| | | 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 | ST13PM | ST13CM |
| | | 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 | | |
| $FA_H$ | **CCU6_TCTR2L** **Reset: $00_H$** Timer Control Register 2 Low | Bit Field | 0 | T13TED | | T13TEC | | | T13SSC | T12SSC |
| | | Type | r | rw | | rw | | | rw | rw |

**Table 14 CCU6 Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $FB_H$ | **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 | MCM EN | 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 | | | | | | | |

**Table 14 CCU6 Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $FE_H$ | **CCU6_CMPSTATL** **Reset: $00_H$** Compare State Register Low | Bit Field | 0 | CC63 ST | CC POS2 | CC POS1 | CC POS0 | CC62 ST | CC61 ST | CC60 ST |
| | | Type | r | rh | rh | rh | rh | rh | rh | rh |
| $FF_H$ | **CCU6_CMPSTATH** **Reset: $00_H$** Compare State Register High | Bit Field | T13IM | COUT 63PS | COUT 62PS | CC62 PS | COUT 61PS | CC61 PS | COUT 60PS | CC60 PS |
| | | Type | rwh | rwh | rwh | rwh | rwh | rwh | rwh | rwh |

### 3.2.4.11 UART1 Registers

The UART1 SFRs can be accessed in the mapped memory area (RMAP = 1).

**Table 15 UART1 Register Overview**

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| RMAP = 1 | | | | | | | | | | |
| **$C8_H$** | **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 |
| $C9_H$ | **SBUF** **Reset: $00_H$** Serial Data Buffer Register | Bit Field | VAL | | | | | | | |
| | | Type | rwh | | | | | | | |
| $CA_H$ | **BCON** **Reset: $00_H$** Baud Rate Control Register | Bit Field | 0 | | | | BRPRE | | | R |
| | | Type | r | | | | rw | | | rw |
| $CB_H$ | **BG** **Reset: $00_H$** Baud Rate Timer/Reload Register | Bit Field | BR_VALUE | | | | | | | |
| | | Type | rwh | | | | | | | |
| $CC_H$ | **FDCON** **Reset: $00_H$** Fractional Divider Control Register | Bit Field | 0 | | | | | NDOV | FDM | FDEN |
| | | Type | r | | | | | rwh | rw | rw |
| $CD_H$ | **FDSTEP** **Reset: $00_H$** Fractional Divider Reload Register | Bit Field | STEP | | | | | | | |
| | | Type | rw | | | | | | | |
| $CE_H$ | **FDRES** **Reset: $00_H$** Fractional Divider Result Register | Bit Field | RESULT | | | | | | | |
| | | Type | rh | | | | | | | |

### 3.2.4.12 SSC Registers

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

**Table 16 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: $00_H$** Baud Rate Timer Reload Register Low | Bit Field | BR_VALUE | | | | | | | |
| | | Type | rw | | | | | | | |
| $AF_H$ | **SSC_BRH** **Reset: $00_H$** Baud Rate Timer Reload Register High | Bit Field | BR_VALUE | | | | | | | |
| | | Type | rw | | | | | | | |

### 3.2.4.13 MultiCAN Registers

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

**Table 17 CAN Register Overview**

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| RMAP = 0 | | | | | | | | | | |
| **$D8_H$** | **ADCON** **Reset: $00_H$** CAN Address/Data Control Register | Bit Field | V3 | V2 | V1 | V0 | AUAD | | BSY | RWEN |
| | | Type | rw | rw | rw | rw | rw | | rh | rw |
| $D9_H$ | **ADL** **Reset: $00_H$** CAN Address Register Low | Bit Field | CA9 | CA8 | CA7 | CA6 | CA5 | CA4 | CA3 | CA2 |
| | | Type | rwh | rwh | rwh | rwh | rwh | rwh | rwh | rwh |
| $DA_H$ | **ADH** **Reset: $00_H$** CAN Address Register High | Bit Field | 0 | | | | CA13 | CA12 | CA11 | CA10 |
| | | Type | r | | | | rwh | rwh | rwh | rwh |

**Table 17 CAN Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $DB_H$ | **DATA0** **Reset: $00_H$** CAN Data Register 0 | Bit Field | CD | | | | | | | |
| | | Type | rwh | | | | | | | |
| $DC_H$ | **DATA1** **Reset: $00_H$** CAN Data Register 1 | Bit Field | CD | | | | | | | |
| | | Type | rwh | | | | | | | |
| $DD_H$ | **DATA2** **Reset: $00_H$** CAN Data Register 2 | Bit Field | CD | | | | | | | |
| | | Type | rwh | | | | | | | |
| $DE_H$ | **DATA3** **Reset: $00_H$** CAN Data Register 3 | Bit Field | CD | | | | | | | |
| | | Type | rwh | | | | | | | |

### 3.2.4.14 OCDS Registers

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

**Table 18 OCDS Register Overview**

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| RMAP = 1 | | | | | | | | | | |
| $E9_H$ | **MMCR2** **Reset: $1U_H$** Monitor Mode Control 2 Register | Bit Field | STMODE | EXBC | DSUSP | MBCON | ALTDI | MMEP | MMODE | JENA |
| | | Type | rw | rw | rw | rwh | rw | rwh | rh | rh |
| $F1_H$ | **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 | |
| $F4_H$ | **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 | | | | | | | |
| $EB_H$ | **MMWR1** **Reset: $00_H$** Monitor Work Register 1 | Bit Field | MMWR1 | | | | | | | |
| | | Type | rw | | | | | | | |

**Table 18 OCDS Register Overview** (cont'd)

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $EC_H$ | **MMWR2** **Reset: $00_H$**<br>Monitor Work Register 2 | Bit Field | MMWR2 | | | | | | | |
| | | 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
- Minimum program width[1] of 32-byte for D-Flash and 64-byte for P-Flash
- 1-sector minimum erase width
- 1-byte read access
- Flash is delivered in erased state (read all zeros)
- Operating supply voltage: 2.5 V ± 7.5 %
- Read access time: 3 × $t_{CCLK}$ = 125 ns[2]
- Program time: 248256 / $f_{SYS}$ = 2.6 ms[3]
- Erase time: 9807360 / $f_{SYS}$ = 102 ms[3]

1) P-Flash: 64-byte wordline can only be programmed once, i.e., one gate disturb allowed.
D-Flash: 32-byte wordline can be programmed twice, i.e., two gate disturbs allowed.

2) Values shown here are typical values. $f_{sys}$ = 96 MHz ± 7.5% ($f_{CCLK}$ = 24 MHz ± 7.5 %) is the maximum frequency range for Flash read access.

3) Values shown here are typical values. $f_{sys}$ = 96 MHz ± 7.5% is the only frequency range for Flash programming and erasing. $f_{sysmin}$ is used for obtaining the worst case timing.

Table 19 shows the Flash data retention and endurance targets.

**Table 19 Flash Data Retention and Endurance (Operating Conditions apply)**

| Retention | Endurance[1)] | Size | Remarks |
|---|---|---|---|
| **Program Flash** | | | |
| 20 years | 1,000 cycles | up to 32 Kbytes[2)] | for 32-Kbyte Variant |
| 20 years | 1,000 cycles | up to 24 Kbytes[2)] | for 24-Kbyte Variant |
| **Data Flash** | | | |
| 20 years | 1,000 cycles | 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 19 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.

2) If no Flash is used for data, the Program Flash size can be up to the maximum Flash size available in the device variant. Having more Data Flash will mean less Flash is available for Program Flash.

### 3.3.1 Flash Bank Sectorization

The XC886/888 product family offers Flash devices with either 24 Kbytes or 32 Kbytes of embedded Flash memory. Each Flash device consists of Program Flash (P-Flash) and Data Flash (D-Flash) bank(s) with different sectorization shown in Figure 11. Both types can be used for code and data storage. The label “Data” neither implies that the D-Flash is mapped to the data memory region, nor that it can only be used for data storage. It is used to distinguish the different Flash bank sectorizations.

The 32-Kbyte Flash device consists of 6 P-Flash and 2 D-Flash banks, while the 24-Kbyte Flash device consists of also of 6 P-Flash banks but with the upper 2 banks only 2 Kbytes each, and only 1 D-Flash bank. The XC886/888 ROM devices offer a single 4-Kbyte D-Flash bank.

The P-Flash banks are always grouped in pairs. As such, the P-Flash banks are also sometimes referred to as P-Flash bank pair. Each sector in a P-Flash bank is grouped with the corresponding sector from the other bank within a bank pair to form a P-Flash bank pair sector.

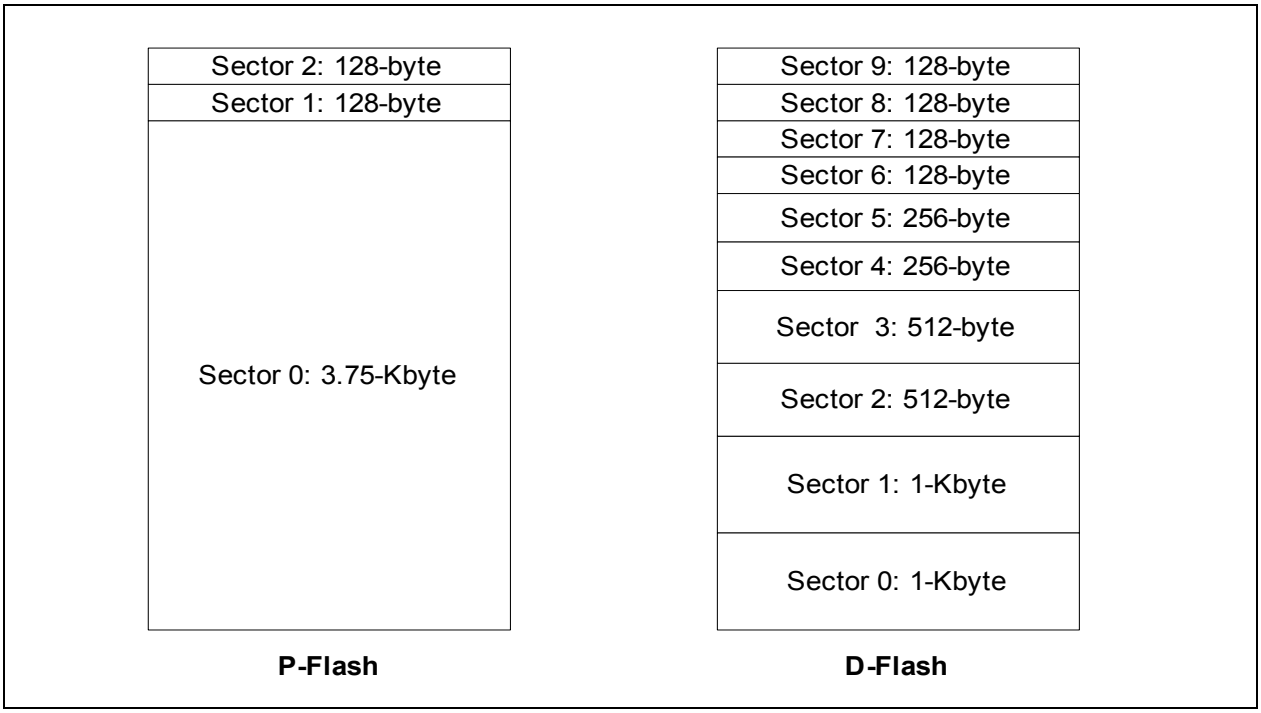



**Figure 11 Flash Bank Sectorization**

The internal structure of each Flash bank represents a sector architecture for flexible erase capability. 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.

The D-Flash bank is divided into more 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.

### 3.3.2 Parallel Read Access of P-Flash

To enhance system performance, the P-Flash banks are configured for parallel read to allow two bytes of linear code to be read in 4 x CCLK cycles, compared to 6 x CCLK cycles if serial read is performed. This is achieved by reading two bytes in parallel from a P-Flash bank pair within the 3 x CCLK cycles access time and storing them in a cache. Subsequent read from the cache by the CPU does not require a wait state and can be completed within 1 x CCLK cycle. The result is the average instruction fetch time from the P-Flash banks is reduced and thus, the MIPS (Mega Instruction Per Second) of the system is increased.

However, if the parallel read feature is not desired due to certain timing constraints, it can be disabled by calling the parallel read disable subroutine.

### 3.3.3 Flash Programming Width

For the P-Flash banks, a programmed wordline (WL) must be erased before it can be reprogrammed as the Flash cells can only withstand one gate disturb. This means that the entire sector containing the WL must be erased since it is impossible to erase a single WL.

For the D-Flash bank, 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 two times (see **Figure 12**)

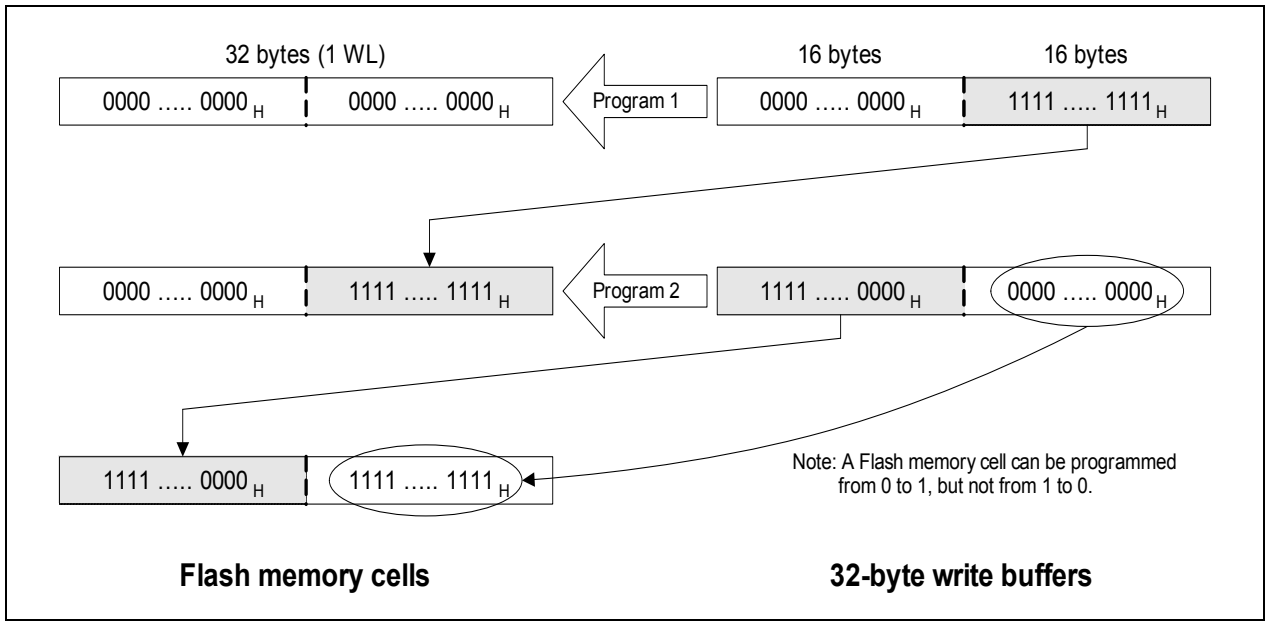



**Figure 12 D-Flash Programming**

*Note: When programming a D-Flash 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”.*

## 3.4 Interrupt System

The XC800 Core supports one non-maskable interrupt (NMI) and 14 maskable interrupt requests. In addition to the standard interrupt functions supported by the core, e.g., configurable interrupt priority and interrupt masking, the XC886/888 interrupt system provides extended interrupt support capabilities such as the mapping of each interrupt vector to several interrupt sources 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 13** to **Figure 17** give a general overview of the interrupt sources and nodes, and their corresponding control and status flags.

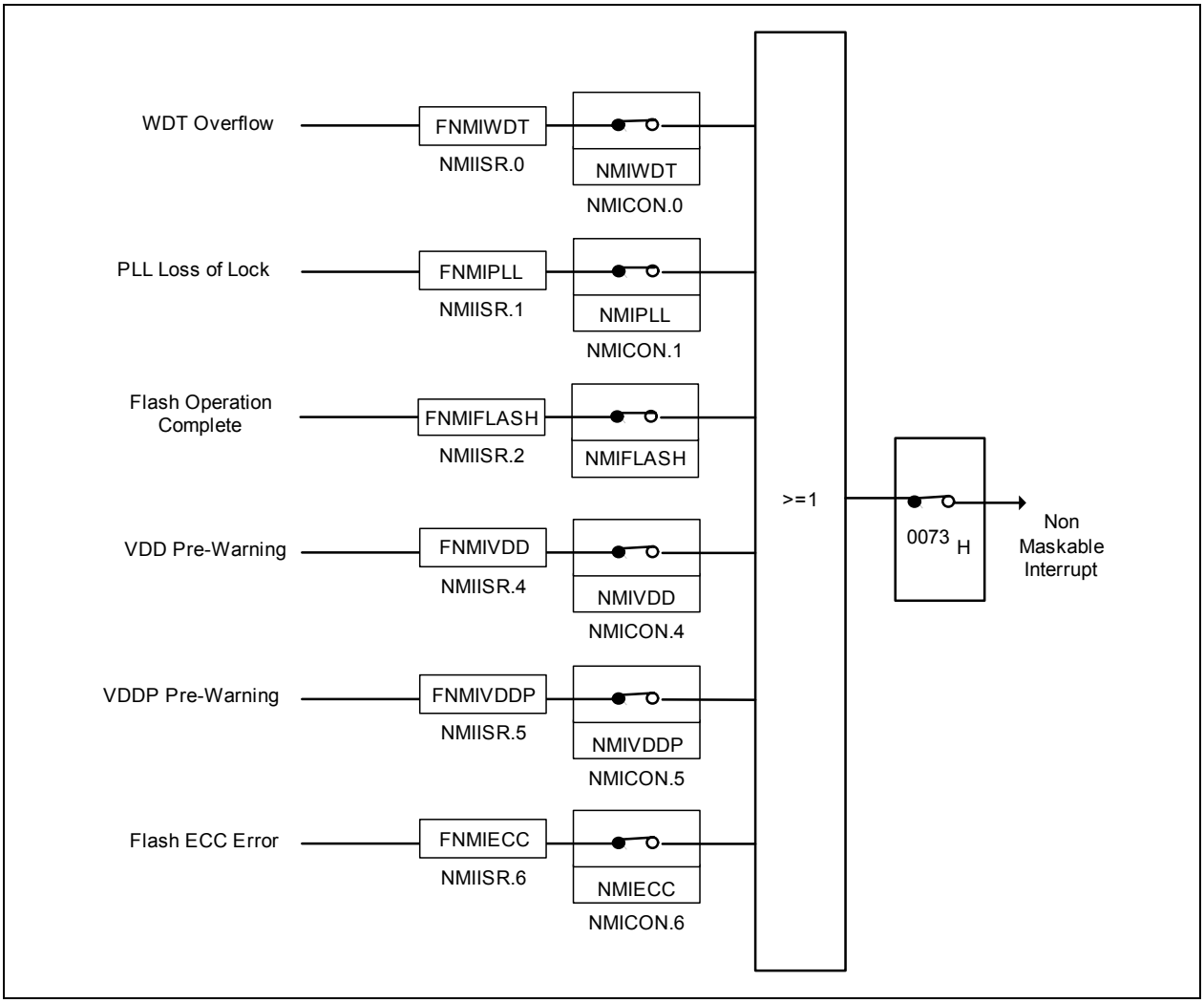



**Figure 13 Non-Maskable Interrupt Request Sources**

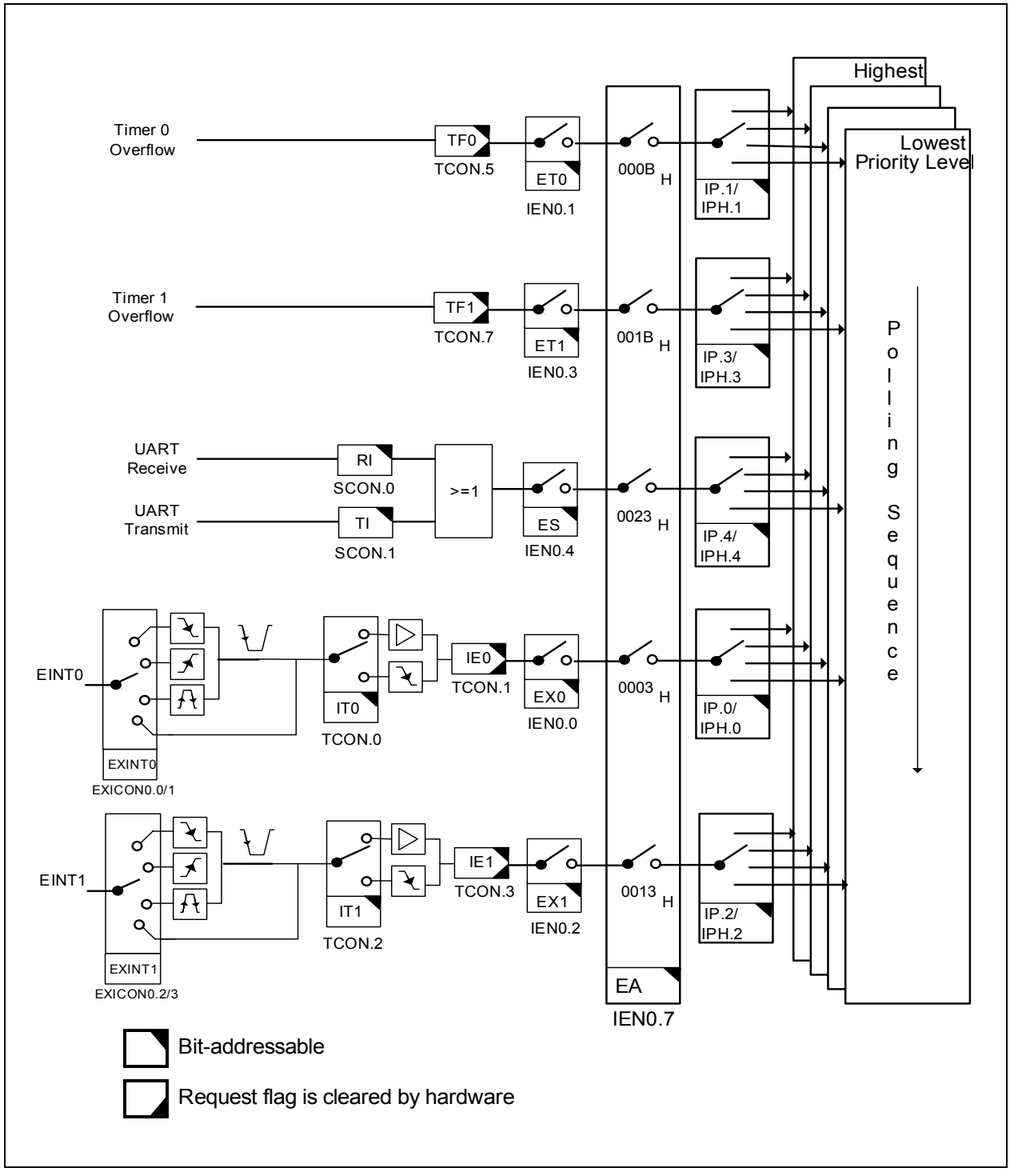



**Figure 14 Interrupt Request Sources (Part 1)**

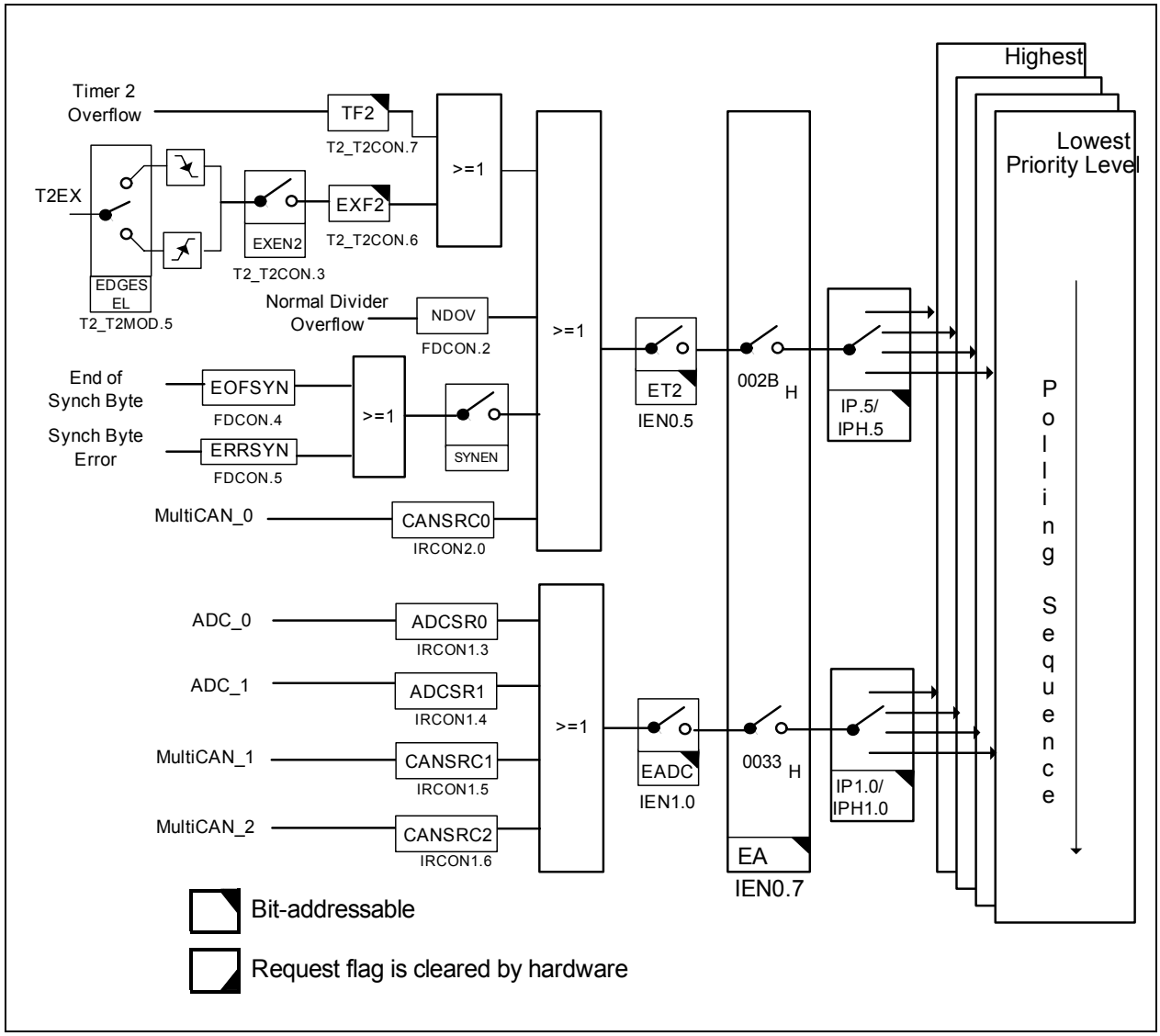



**Figure 15 Interrupt Request Sources (Part 2)**

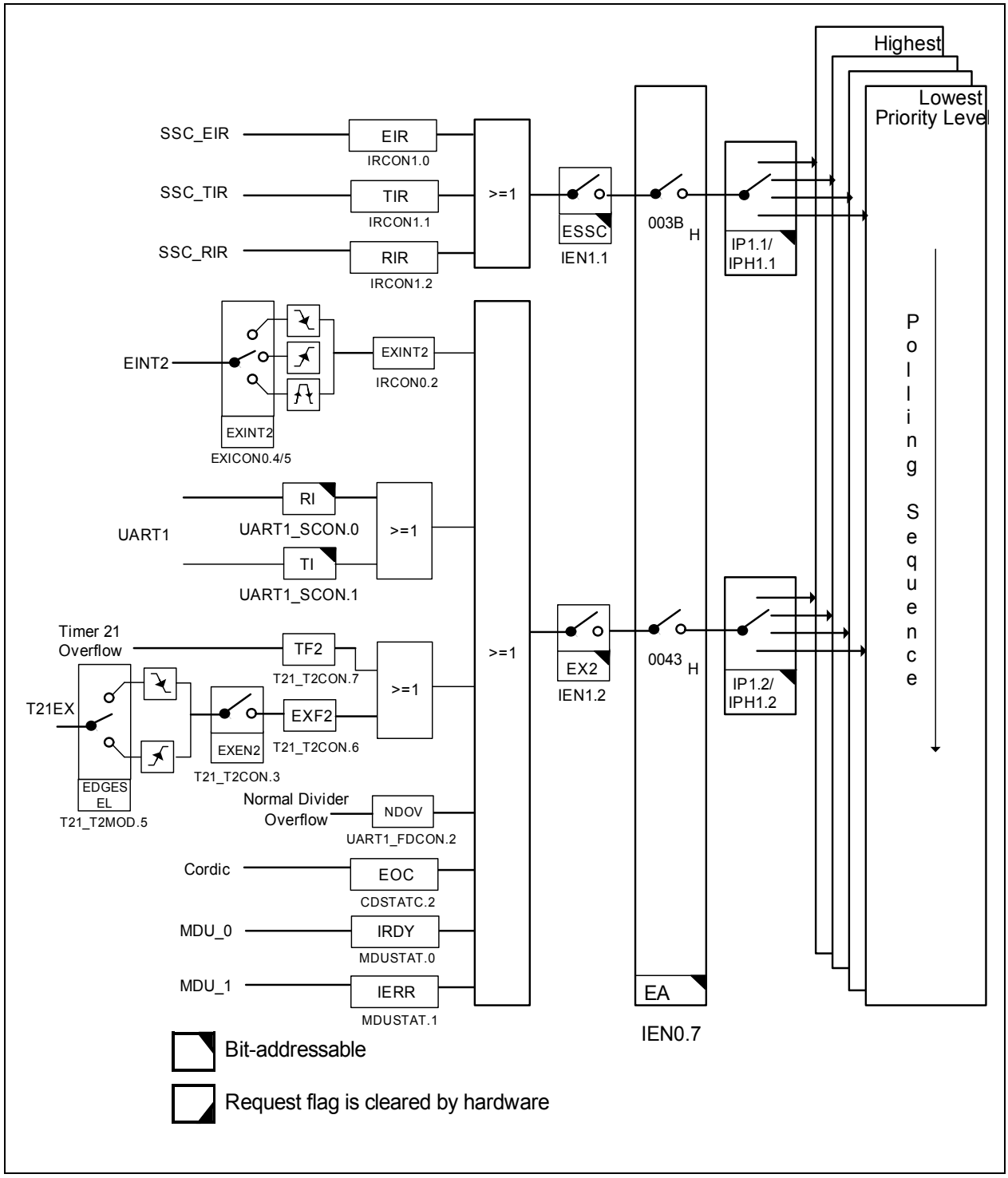



**Figure 16 Interrupt Request Sources (Part 3)**

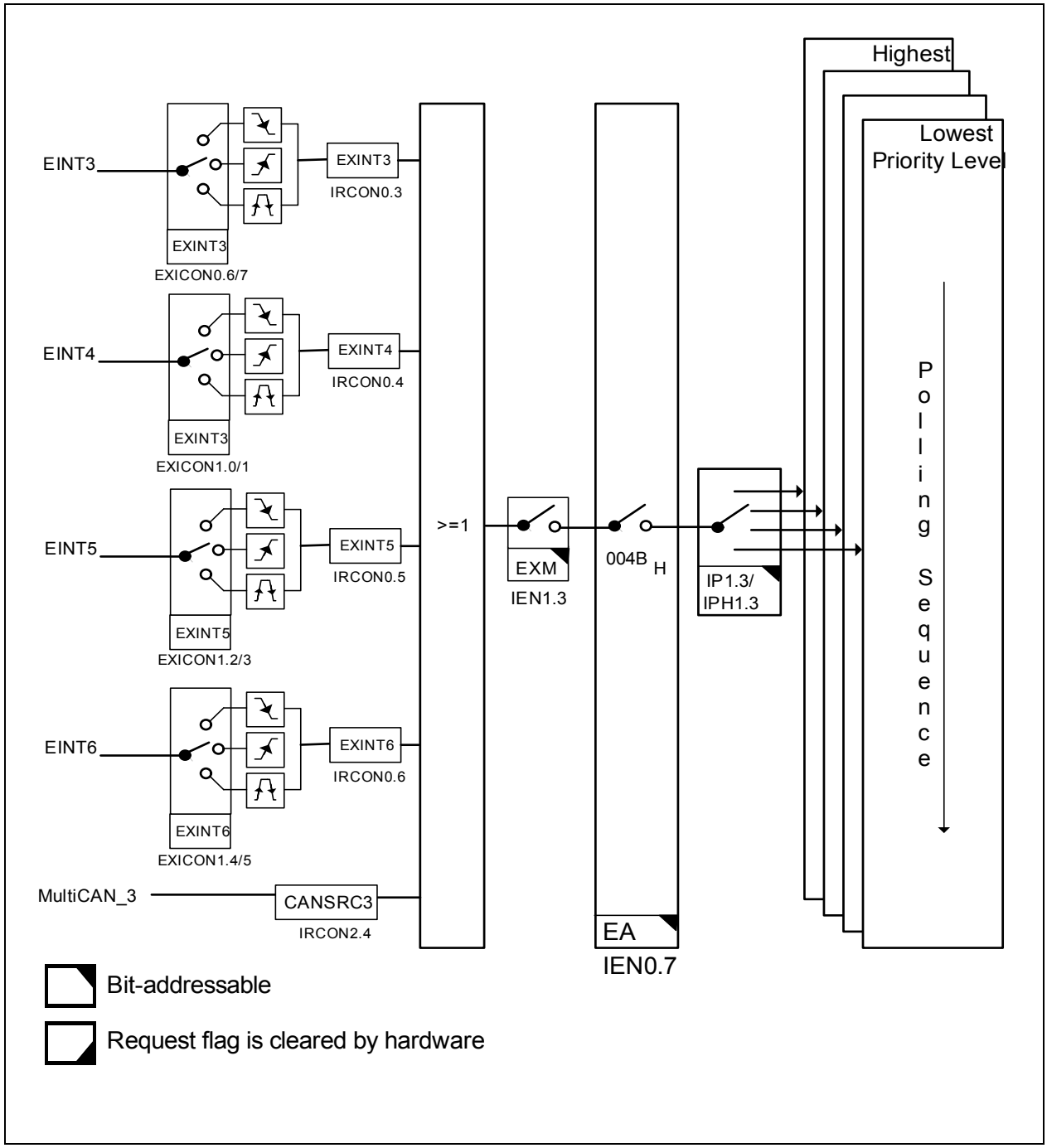



**Figure 17 Interrupt Request Sources (Part 4)**

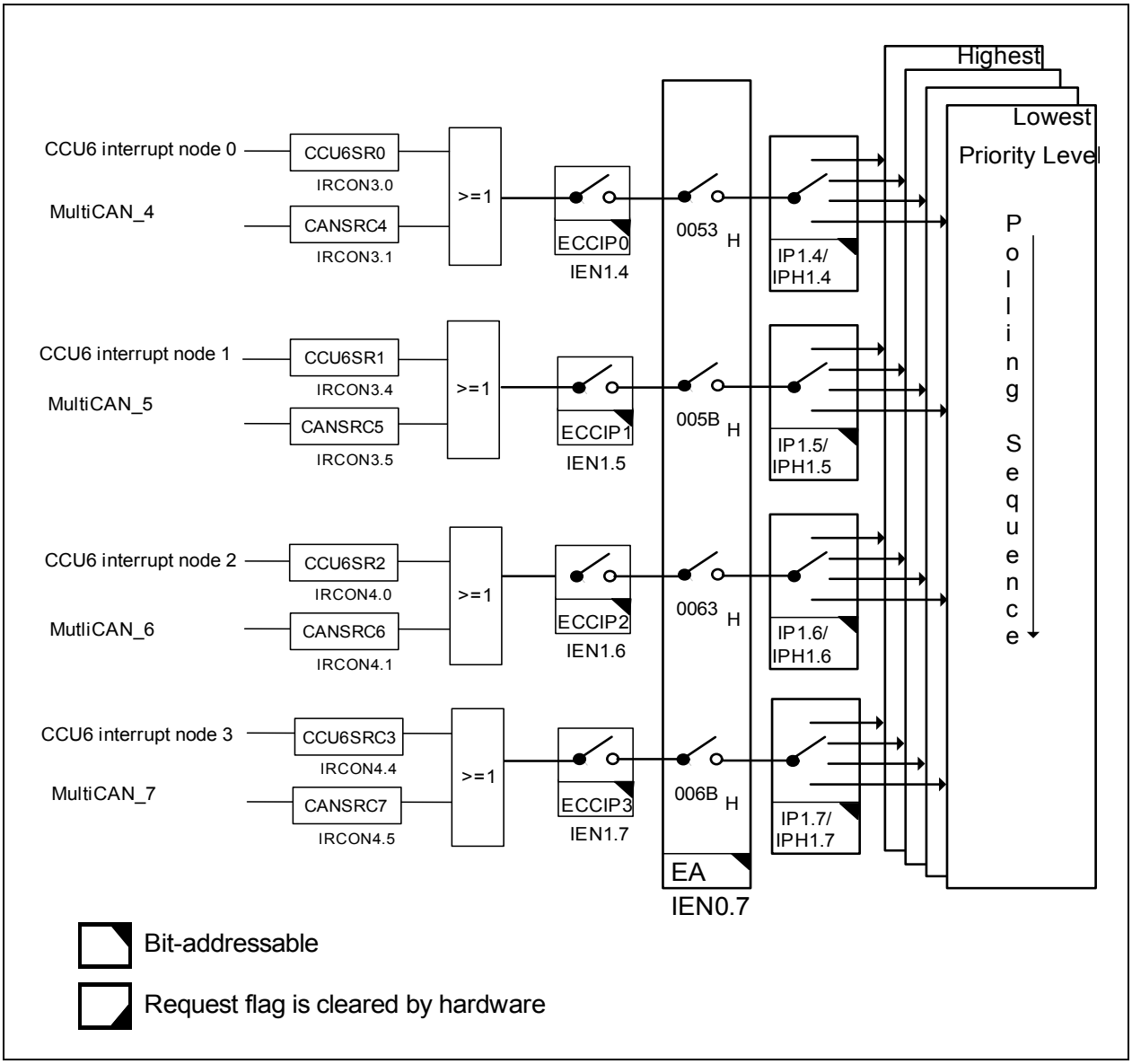



**Figure 18 Interrupt Request Sources (Part 5)**

### 3.4.2 Interrupt Source and Vector

Each interrupt event source has an associated interrupt vector address for the interrupt node it belongs to. This vector is accessed to service the corresponding interrupt node request. The interrupt service of each interrupt source can be individually enabled or disabled via an enable bit. The assignment of the XC886/888 interrupt sources to the interrupt vector address and the corresponding interrupt node enable bits are summarized in **Table 20**.

**Table 20 Interrupt Vector Addresses**

| Interrupt Source | Vector Address | Assignment for XC886/888 | 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 | |
| | | UART Fractional Divider (Normal Divider Overflow) | | |
| | | MultiCAN Node 0 | | |
| | | LIN | | |

**Table 20 Interrupt Vector Addresses** (cont'd)

| Interrupt Source | Vector Address | Assignment for XC886/888 | Enable Bit | SFR |
|---|---|---|---|---|
| XINTR6 | $0033_H$ | MultiCAN Nodes 1 and 2 | EADC | IEN1 |
| | | ADC[1:0] | | |
| XINTR7 | $003B_H$ | SSC | ESSC | |
| XINTR8 | $0043_H$ | External Interrupt 2 | EX2 | |
| | | T21 | | |
| | | CORDIC | | |
| | | UART1 | | |
| | | UART1 Fractional Divider (Normal Divider Overflow) | | |
| | | MDU[1:0] | | |
| XINTR9 | $004B_H$ | External Interrupt 3 | EXM | |
| | | External Interrupt 4 | | |
| | | External Interrupt 5 | | |
| | | External Interrupt 6 | | |
| | | MultiCAN Node 3 | | |
| XINTR10 | $0053_H$ | CCU6 INP0 | ECCIP0 | |
| | | MultiCAN Node 4 | | |
| XINTR11 | $005B_H$ | CCU6 INP1 | ECCIP1 | |
| | | MultiCAN Node 5 | | |
| XINTR12 | $0063_H$ | CCU6 INP2 | ECCIP2 | |
| | | MultiCAN Node 6 | | |
| XINTR13 | $006B_H$ | CCU6 INP3 | ECCIP3 | |
| | | MultiCAN Node 7 | | |

### 3.4.3 Interrupt Priority

An interrupt that is currently being serviced can only be interrupted by a higher-priority interrupt, but not by another interrupt of the same or lower priority. Hence, an interrupt of the highest priority cannot be interrupted by any other interrupt request.

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 21**.

**Table 21 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, MultiCAN, LIN Interrupt | 6 |
| ADC, MultiCAN Interrupt | 7 |
| SSC Interrupt | 8 |
| External Interrupt 2, Timer 21, UART1, UART1 Normal Divider Overflow, MDU, CORDIC Interrupt | 9 |
| External Interrupt [6:3], MultiCAN Interrupt | 10 |
| CCU6 Interrupt Node Pointer 0, MultiCAN interrupt | 11 |
| CCU6 Interrupt Node Pointer 1, MultiCAN Interrupt | 12 |
| CCU6 Interrupt Node Pointer 2, MultiCAN Interrupt | 13 |
| CCU6 Interrupt Node Pointer 3, MultiCAN Interrupt | 14 |

## 3.5 Parallel Ports

The XC886 has 34 port pins organized into five parallel ports, Port 0 (P0) to Port 4 (P4), while the XC888 has 48 port pins organized into six parallel ports, Port 0 (P0) to Port 5 (P5). Each pin has a pair of internal pull-up and pull-down devices that can be individually enabled or disabled. Ports P0, P1, P3, P4 and P5 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).

### 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 19** shows the structure of a bidirectional port pin.

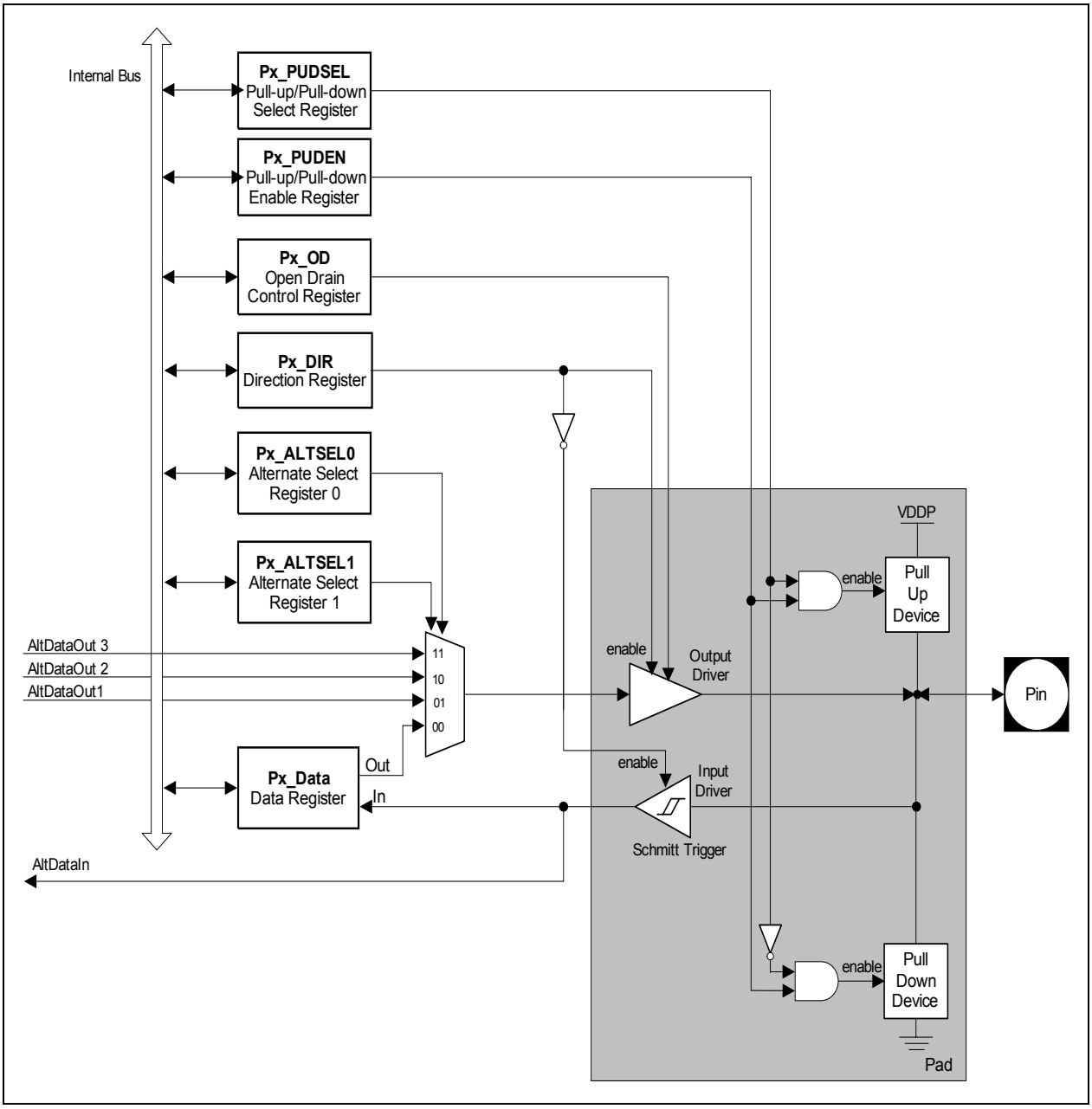



**Figure 19 General Structure of Bidirectional Port**

**Figure 20** shows the structure of an input-only port pin.

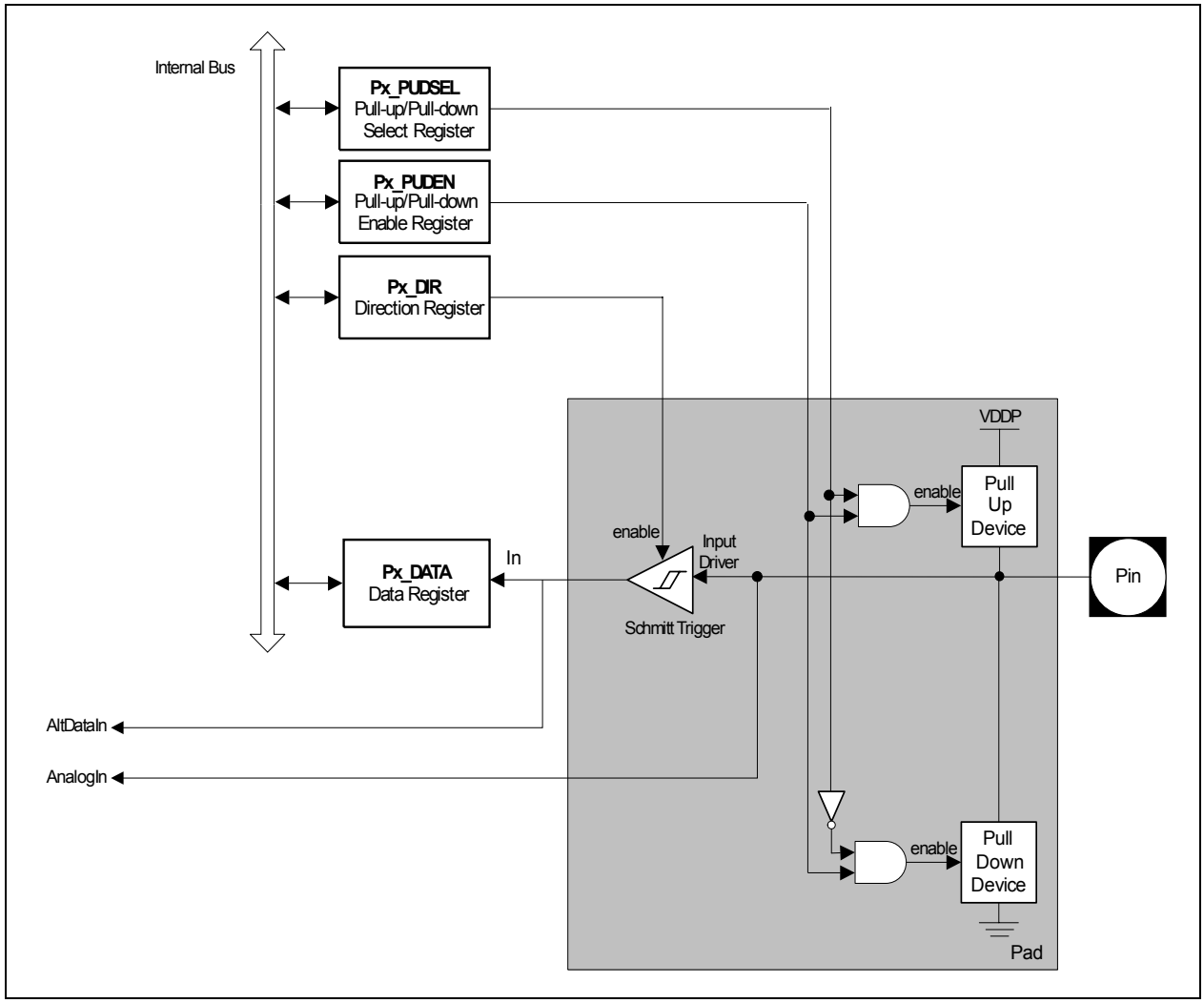



**Figure 20 General Structure of Input Port**

## 3.6 Power Supply System with Embedded Voltage Regulator

The XC886/888 microcontroller requires two different levels of power supply:

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

**Figure 21** shows the XC886/888 power supply system. A power supply of 3.3 V or 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.

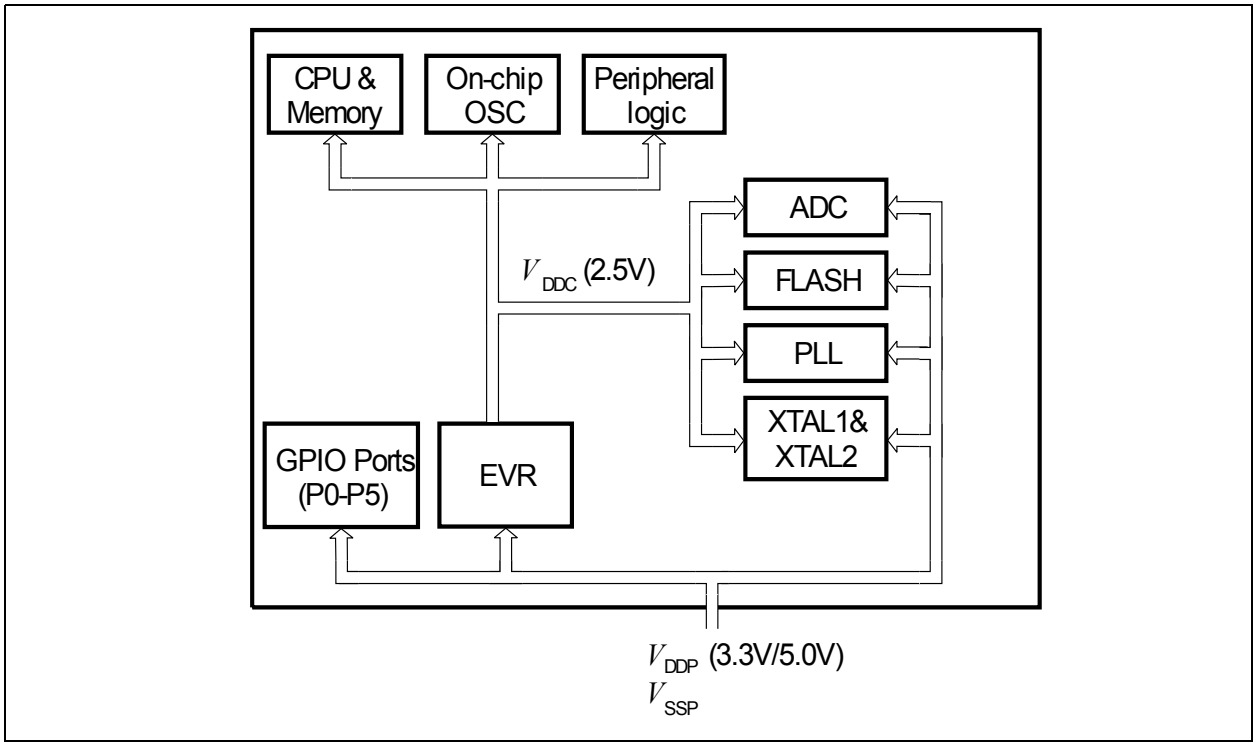



**Figure 21 XC886/888 Power Supply System**

### EVR Features

- Input voltage ($V_{DDP}$): 3.3 V/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 XC886/888 has five types of reset: power-on reset, hardware reset, watchdog timer reset, power-down wake-up reset, and brownout reset.

When the XC886/888 is first powered up, the status of certain pins (see **Table 23**) 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 22**. The $V_{DDP}$ capacitor value is 100 nF while the $V_{DDC}$ 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 23**.

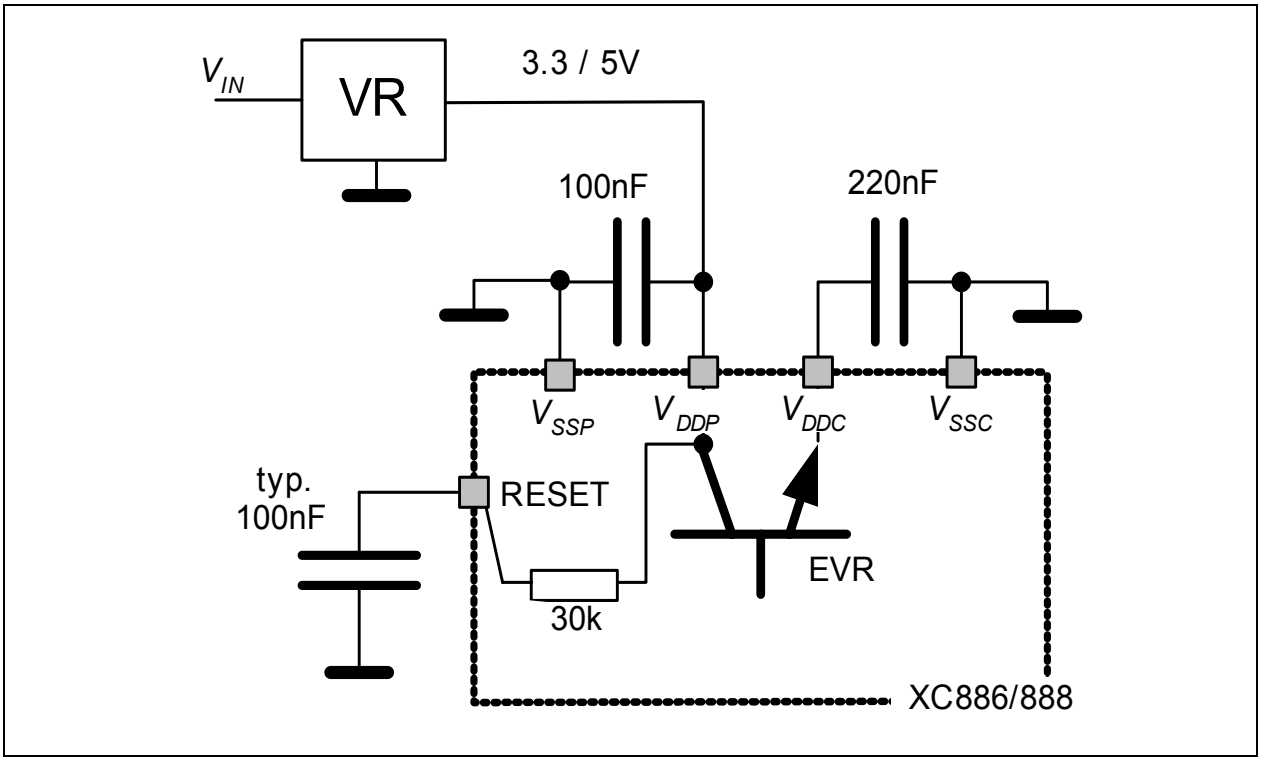



**Figure 22 Reset Circuitry**

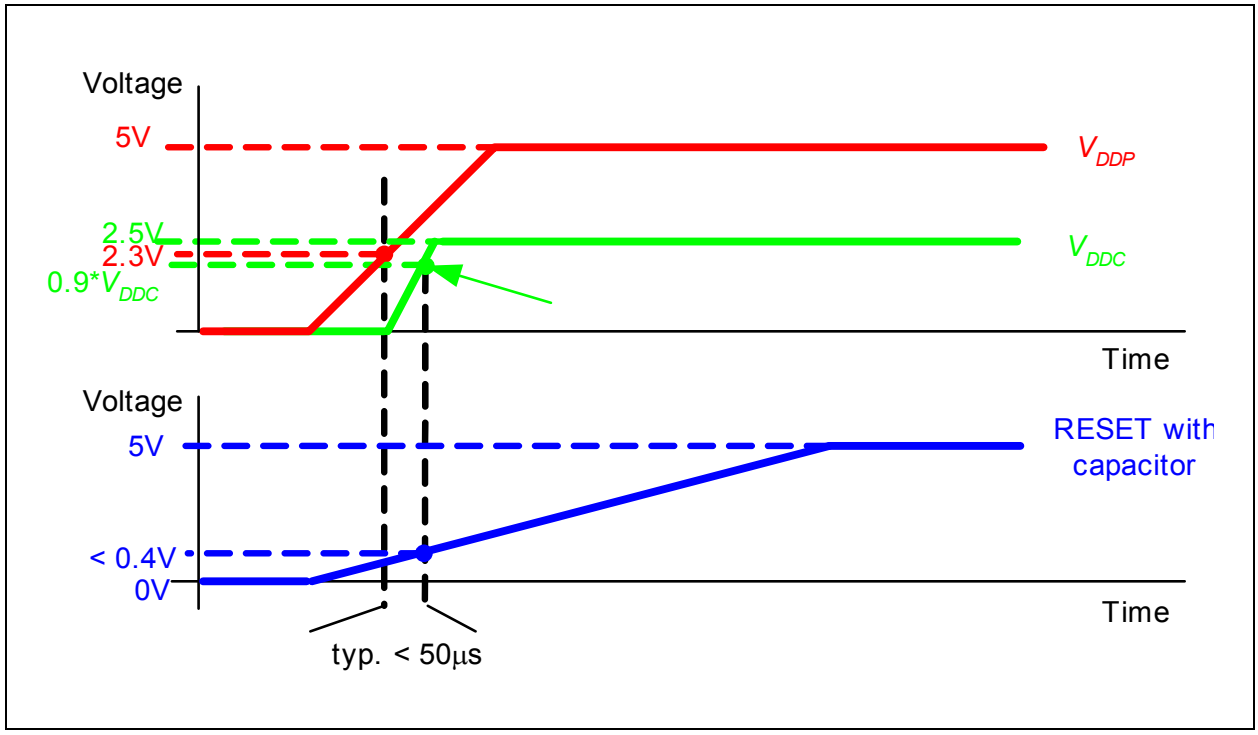



**Figure 23 $V_{DDP}$, $V_{DDC}$ and $V_{RESET}$ during Power-on Reset**

The second type of reset in XC886/888 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.

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 22 lists the functions of the XC886/888 and the various reset types that affect these functions. The symbol "■" signifies that the particular function is reset to its default state.

**Table 22 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 XC886/888 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 MBC, TMS and P0.0 collectively select the different boot options. Table 23 shows the available boot options in the XC886/888.

**Table 23 XC886/888 Boot Selection**

| MBC | TMS | P0.0 | Type of Mode | PC Start Value |
|---|---|---|---|---|
| 1 | 0 | X | User Mode[1]; on-chip OSC/PLL non-bypassed | $0000_H$ |
| 0 | 0 | X | BSL Mode; on-chip OSC/PLL non-bypassed[2] | $0000_H$ |
| 0 | 1 | 0 | OCDS Mode; on-chip OSC/PLL non-bypassed | $0000_H$ |
| 1 | 1 | 0 | User (JTAG) Mode[3]; on-chip OSC/PLL non-bypassed (normal) | $0000_H$ |

1) BSL mode is automatically entered if no valid password is installed and data at memory address 0000H equals zero.
2) OSC is bypassed in MultiCAN BSL mode
3) Normal user mode with standard JTAG (TCK,TDI,TDO) pins for hot-attach purpose.

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

## 3.8 Clock Generation Unit

The Clock Generation Unit (CGU) allows great flexibility in the clock generation for the XC886/888. 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 XC886/888, the oscillator can be from either of these two sources: the on-chip oscillator (9.6 MHz) or the external oscillator (4 MHz to 12 MHz). The term “oscillator” is used to refer to both on-chip oscillator and external oscillator, unless otherwise stated. After the reset, the on-chip oscillator will be used by default.The external oscillator can be selected via software. 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.

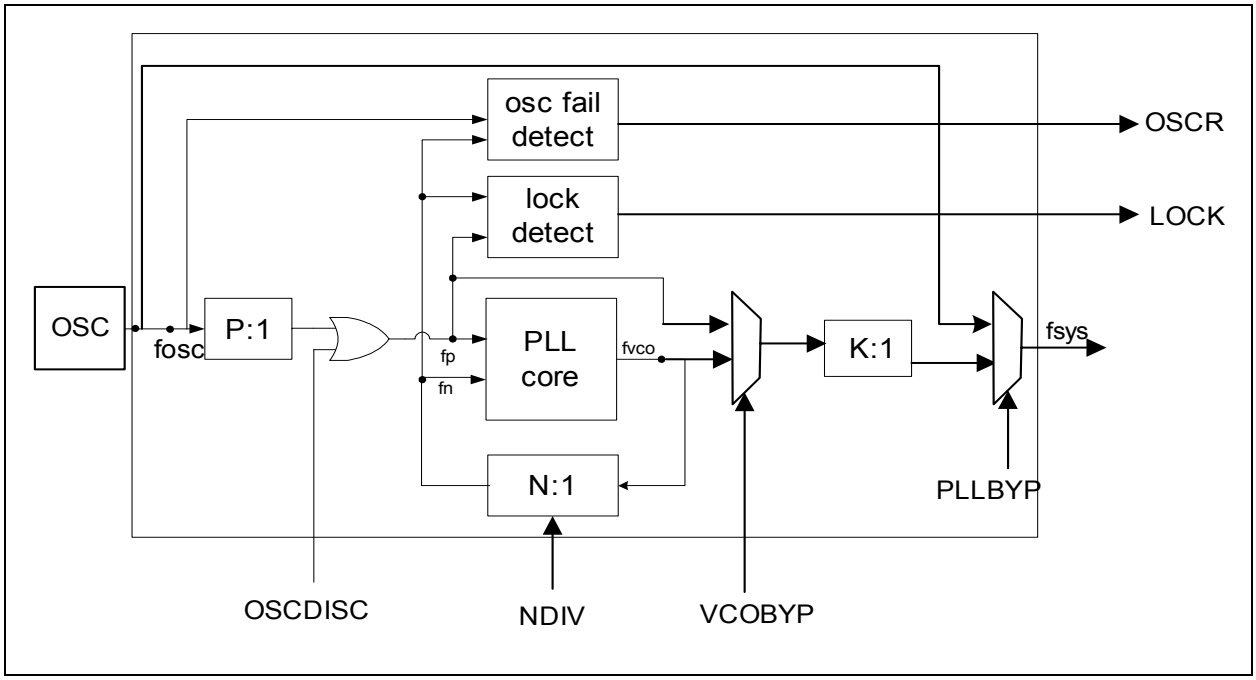



**Figure 24 CGU Block Diagram**

### PLL Base Mode

When the oscillator is disconnected from the PLL, the system clock is derived from the VCO base (free running) frequency clock (**Table 25**) divided by the K factor.

$$f_{SYS} = f_{VCObase} \times \frac{1}{K} \quad (3.1)$$

### 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.

$$f_{SYS} = f_{OSC} \times \frac{1}{P \times K} \quad (3.2)$$

### PLL Mode

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

$$f_{SYS} = f_{OSC} \times \frac{N}{P \times K} \quad (3.3)$$

### System Frequency Selection

For the XC886/888, the value of P is fixed to 1. In order to obtain the required fsys, the value of N and K can be selected by bits NDIV and KDIV respectively for different oscillator inputs. The output frequency must always be configured for 96 MHz. **Table 24** provides examples on how $f_{sys}$ = 96 MHz can be obtained for the different oscillator sources.

**Table 24 System frequency ($f_{sys}$ = 96 MHz)**

| Oscillator | Fosc | N | P | K | Fsys |
|---|---|---|---|---|---|
| On-chip | 9.6 MHz | 20 | 1 | 2 | 96 MHz |
| External | 8 MHz | 24 | 1 | 2 | 96 MHz |
| | 6 MHz | 32 | 1 | 2 | 96 MHz |
| | 4 MHz | 48 | 1 | 2 | 96 MHz |

Table 25 shows the VCO range for the XC886/888.

**Table 25 VCO Range**

| $f_{VCOmin}$ | $f_{VCOmax}$ | $f_{VCOFREEmin}$ | $f_{VCOFREEmax}$ | Unit |
|---|---|---|---|---|
| 150 | 200 | 20 | 80 | MHz |
| 100 | 150 | 10 | 80 | MHz |

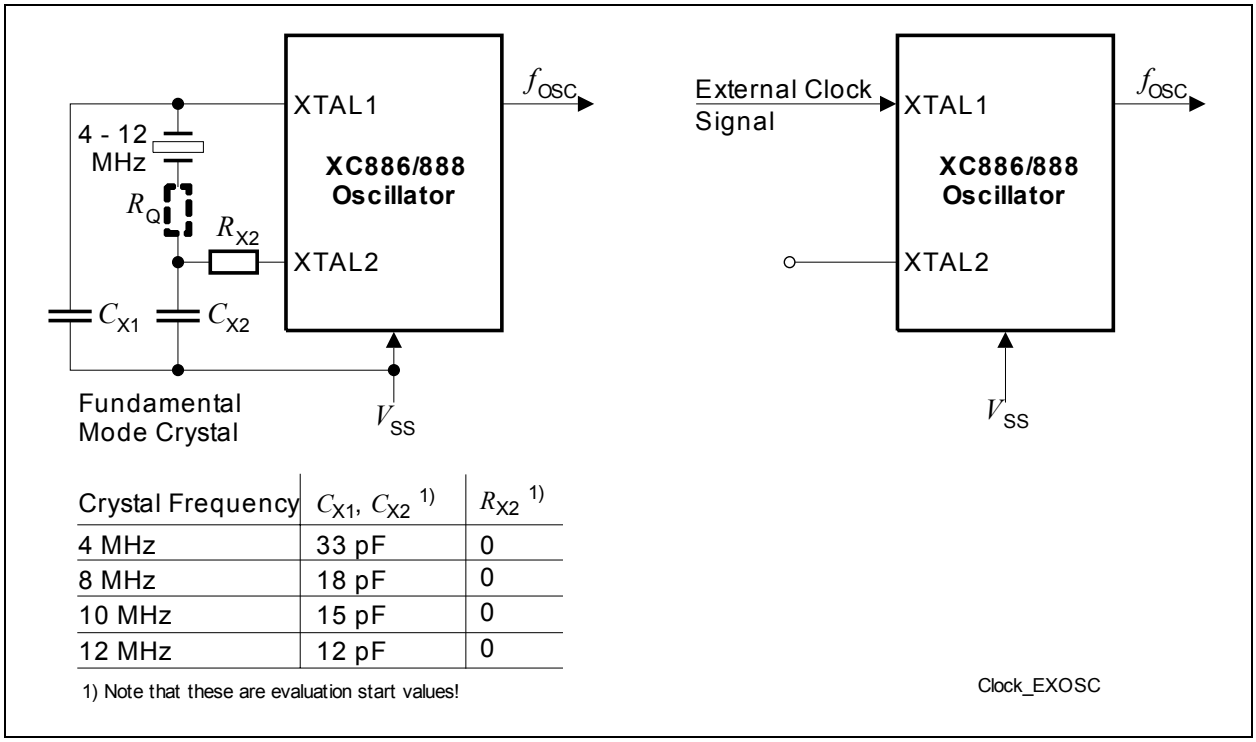
### 3.8.1 Recommended External Oscillator Circuits

The oscillator circuit, a Pierce oscillator, is designed to work with both, an external crystal oscillator or an external stable clock source. It basically consists of an inverting amplifier and a feedback element with XTAL1 as input, and XTAL2 as output.

When using a crystal, a proper external oscillator circuitry must be connected to both pins, XTAL1 and XTAL2. The crystal frequency can be within the range of 4 MHz to 12 MHz. Additionally, it is necessary to have two load capacitances $C_{X1}$ and $C_{X2}$, and depending on the crystal type, a series resistor $R_{X2}$, to limit the current. A test resistor $R_Q$ may be temporarily inserted to measure the oscillation allowance (negative resistance) of the oscillator circuitry. $R_Q$ values are typically specified by the crystal vendor. The $C_{X1}$ and $C_{X2}$ values shown in Figure 25 can be used as starting points for the negative resistance evaluation and for non-productive systems. The exact values and related operating range are dependent on the crystal frequency and have to be determined and optimized together with the crystal vendor using the negative resistance method. Oscillation measurement with the final target system is strongly recommended to verify the input amplitude at XTAL1 and to determine the actual oscillation allowance (margin negative resistance) for the oscillator-crystal system.

When using an external clock signal, the signal must be connected to XTAL1. XTAL2 is left open (unconnected).

The oscillator can also be used in combination with a ceramic resonator. The final circuitry must also be verified by the resonator vendor. Figure 25 shows the recommended external oscillator circuitries for both operating modes, external crystal mode and external input clock mode.




| Crystal Frequency | $C_{X1}$, $C_{X2}$ 1) | $R_{X2}$ 1) |
|---|---|---|
| 4 MHz | 33 pF | 0 |
| 8 MHz | 18 pF | 0 |
| 10 MHz | 15 pF | 0 |
| 12 MHz | 12 pF | 0 |

1) Note that these are evaluation start values!

**Figure 25 External Oscillator Circuitry**

*Note: For crystal operation, it is strongly recommended to measure the negative resistance in the final target system (layout) to determine the optimum parameters for the oscillator operation. Please refer to the minimum and maximum values of the negative resistance specified by the crystal supplier.*

### 3.8.2 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 = 24 MHz
- Fast clock (used by MultiCAN): FCLK = 24 or 48 MHz
- Peripheral clock: PCLK = 24 MHz
- Flash Interface clock: CCLK2 = 48 MHz and CCLK = 24 MHz

In addition, different clock frequencies can be output to pin CLKOUT (P0.0 or P0.7). The clock output frequency, which is derived from the clock output divider (bit COREL), can further be divided by 2 using toggle latch (bit TLEN is set to 1). The resulting output frequency has a 50% duty cycle. **Figure 26** shows the clock distribution of the XC886/888.

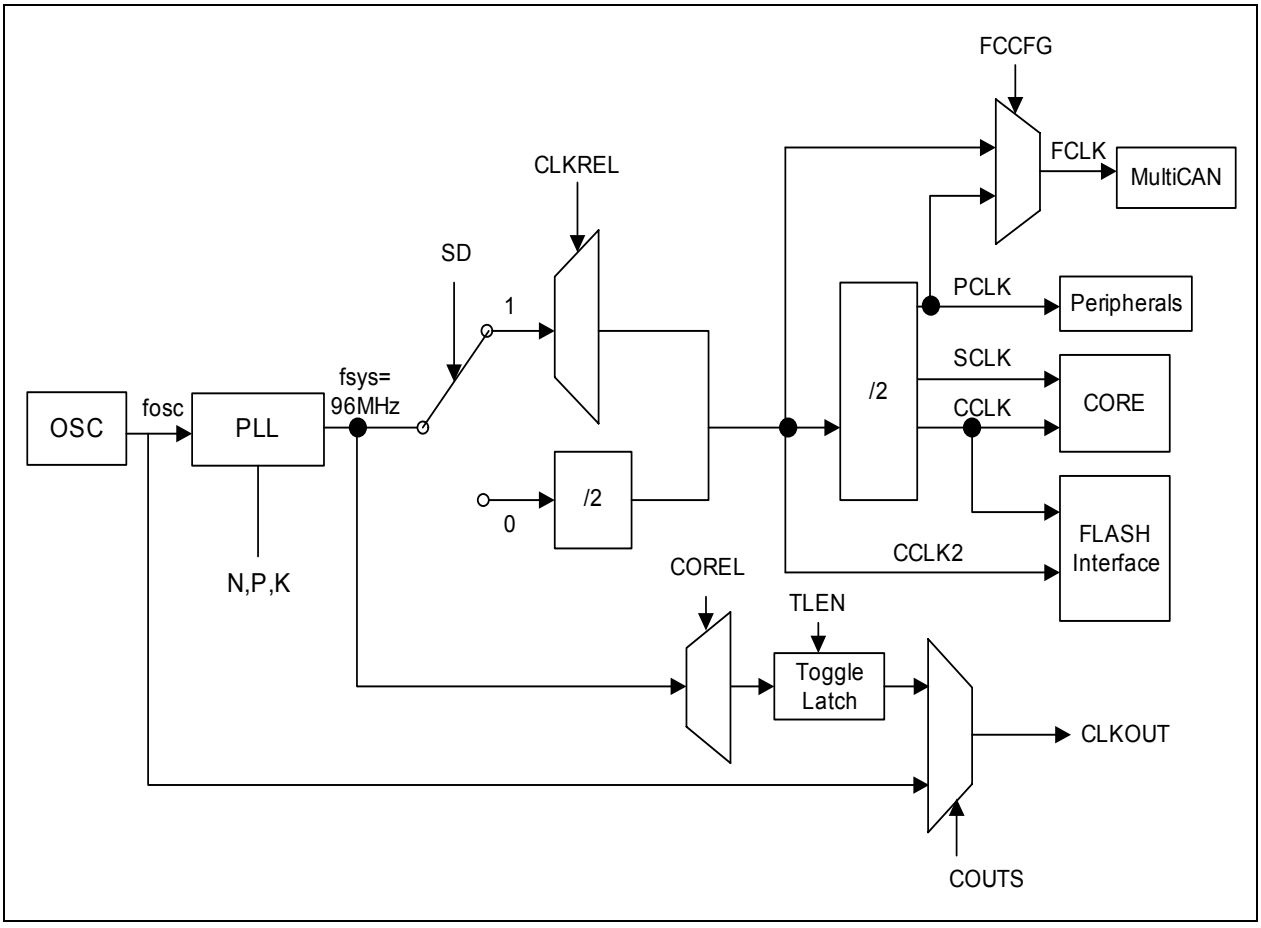



**Figure 26 Clock Generation from $f_{sys}$**

For power saving purposes, the clocks may be disabled or slowed down according to Table 26.

**Table 26 System frequency ($f_{sys}$ = 96 MHz)**

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

## 3.9 Power Saving Modes

The power saving modes of the XC886/888 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 27**) 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

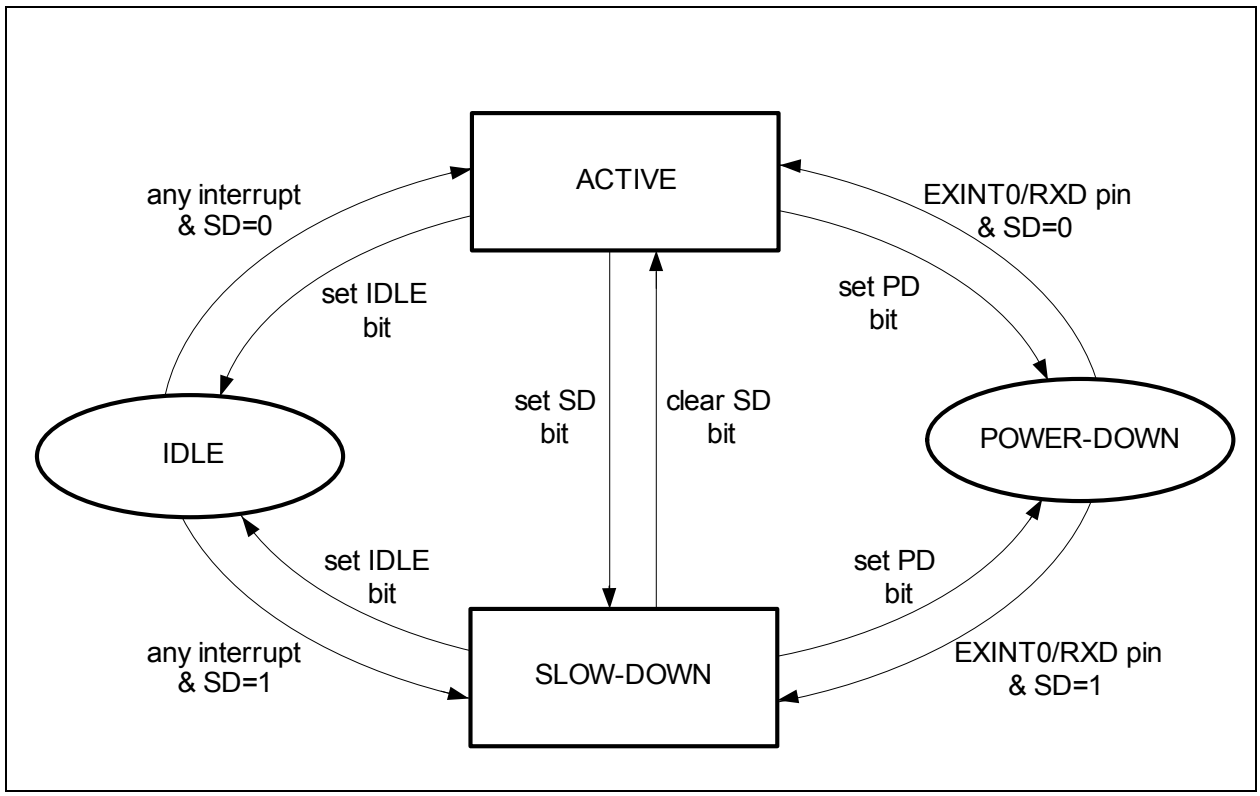



**Figure 27 Transition between Power Saving Modes**

## 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 XC886/888 system reset. Hence, routine service of the WDT confirms that the system is functioning properly. This ensures that an accidental malfunction of the XC886/888 will be aborted in a user-specified time period.

In debug mode, the WDT is default 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 28** shows the block diagram of the WDT unit.

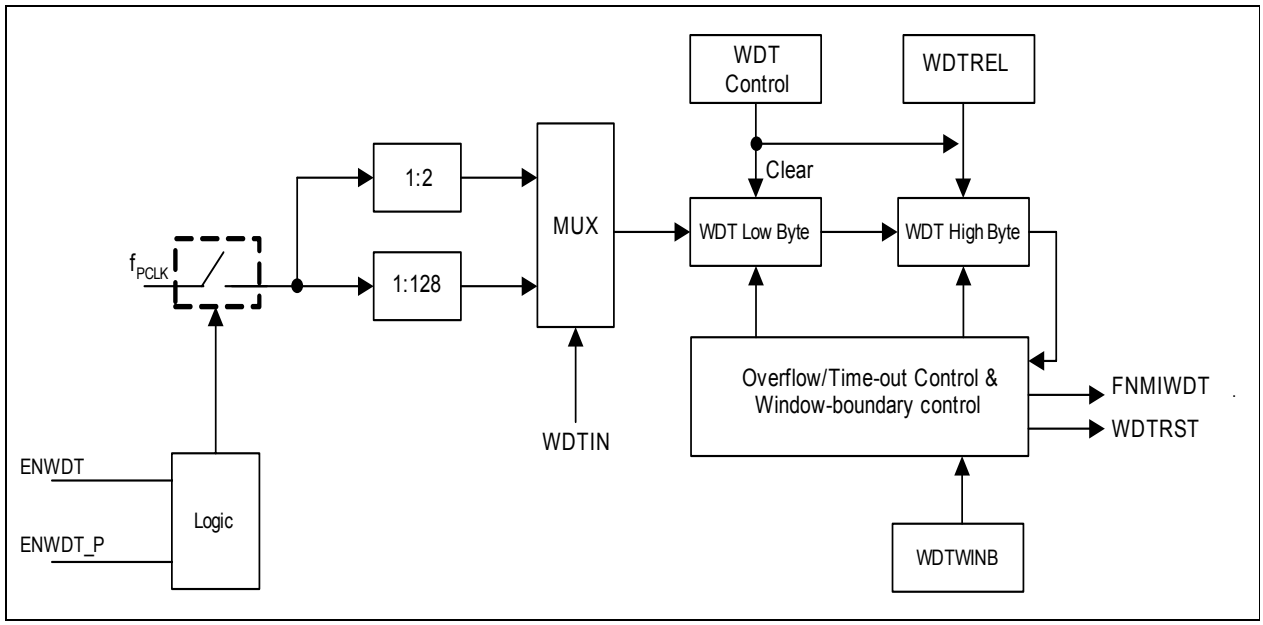



**Figure 28 WDT Block Diagram**

If the WDT is not serviced before the timer overflow, a system malfunction is assumed. As a result, the WDT NMI is triggered (assert FNMIWDT) 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:

$$P_{WDT} = \frac{2^{(1 + WDTIN \times 6)} \times (2^{16} - WDTREL \times 2^8)}{f_{PCLK}} \quad (3.4)$$

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 29**. This period can be calculated using the same formula by replacing WDTREL with WDTWINB. For this feature to be useful, WDTWINB cannot be smaller than WDTREL.

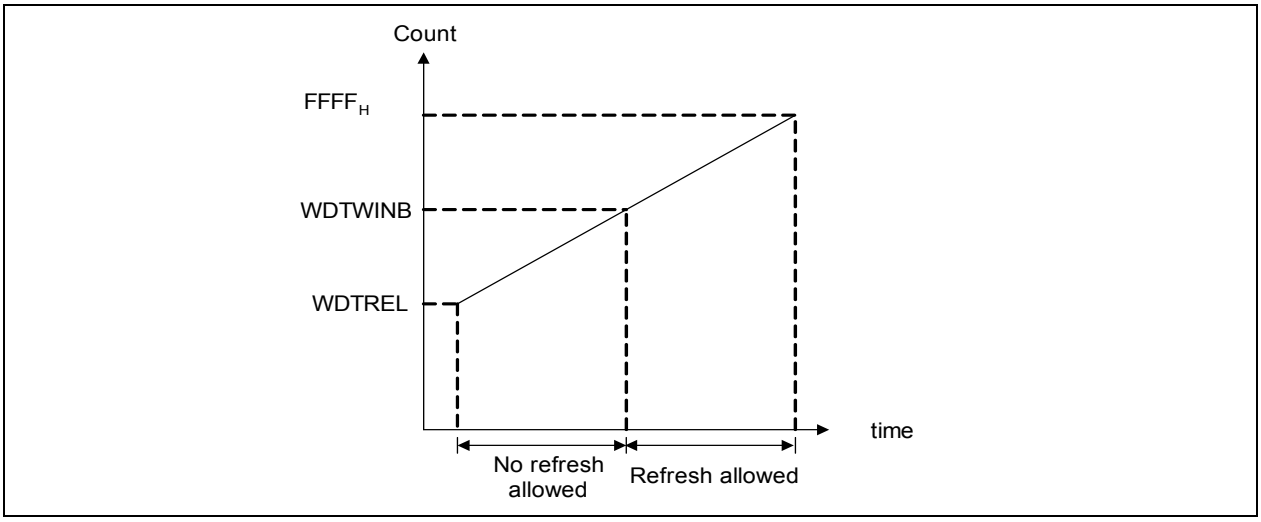



**Figure 29 WDT Timing Diagram**

Table 27 lists the possible watchdog time ranges that can be achieved using a certain module clock. Some numbers are rounded to 3 significant digits.

**Table 27 Watchdog Time Ranges**

| Reload value In WDTREL | Prescaler for $f_{PCLK}$ | |
|---|---|---|
| | **2 (WDTIN = 0)** | **128 (WDTIN = 1)** |
| | **24 MHz** | **24 MHz** |
| $FF_H$ | 21.3 µs | 1.37 ms |
| $7F_H$ | 2.75 ms | 176 ms |
| $00_H$ | 5.46 ms | 350 ms |

## 3.11 Multiplication/Division Unit

The Multiplication/Division Unit (MDU) provides fast 16-bit multiplication, 16-bit and 32-bit division as well as shift and normalize features. It has been integrated to support the XC886/888 Core in real-time control applications, which require fast mathematical computations.

### Features

- Fast signed/unsigned 16-bit multiplication
- Fast signed/unsigned 32-bit divide by 16-bit and 16-bit divide by 16-bit operations
- 32-bit unsigned normalize operation
- 32-bit arithmetic/logical shift operations

Table 28 specifies the number of clock cycles used for calculation in various operations.

**Table 28 MDU Operation Characteristics**

| Operation | Result | Remainder | No. of Clock Cycles used for calculation |
|---|---|---|---|
| Signed 32-bit/16-bit | 32-bit | 16-bit | 33 |
| Signed 16-bit/16bit | 16-bit | 16-bit | 17 |
| Signed 16-bit x 16-bit | 32-bit | - | 16 |
| Unsigned 32-bit/16-bit | 32-bit | 16-bit | 32 |
| Unsigned 16-bit/16-bit | 16-bit | 16-bit | 16 |
| Unsigned 16-bit x 16-bit | 32-bit | - | 16 |
| 32-bit normalize | - | - | No. of shifts + 1 (Max. 32) |
| 32-bit shift L/R | - | - | No. of shifts + 1 (Max. 32) |

## 3.12 CORDIC Coprocessor

The CORDIC Coprocessor provides CPU with hardware support for the solving of circular (trigonometric), linear (multiply-add, divide-add) and hyperbolic functions.

**Features**

- Modes of operation
  - Supports all CORDIC operating modes for solving circular (trigonometric), linear (multiply-add, divide-add) and hyperbolic functions
  - Integrated look-up tables (LUTs) for all operating modes
- Circular vectoring mode: Extended support for values of initial X and Y data up to full range of $[-2^{15},(2^{15}-1)]$ for solving angle and magnitude
- Circular rotation mode: Extended support for values of initial Z data up to full range of $[-2^{15},(2^{15}-1)]$, representing angles in the range $[-\pi,((2^{15}-1)/2^{15})\pi]$ for solving trigonometry
- Implementation-dependent operational frequency of up to 80 MHz
- Gated clock input to support disabling of module
- 16-bit accessible data width
  - 24-bit kernel data width plus 2 overflow bits for X and Y each
  - 20-bit kernel data width plus 1 overflow bit for Z
  - With KEEP bit to retain the last value in the kernel register for a new calculation
- 16 iterations per calculation: Approximately 41 clock-cycles or less, from set of start (ST) bit to set of end-of-calculation flag, excluding time taken for write and read access of data bytes.
- Twos complement data processing
  - Only exception: X result data with user selectable option for unsigned result
- X and Y data generally accepted as integer or rational number; X and Y must be of the same data form
- Entries of LUTs are 20-bit signed integers
  - Entries of atan and atanh LUTs are integer representations (S19) of angles with the scaling such that $[-2^{15},(2^{15}-1)]$ represents the range $[-\pi,((2^{15}-1)/2^{15})\pi]$
  - Accessible Z result data for circular and hyperbolic functions is integer in data form of S15
- Emulated LUT for linear function
  - Data form is 1 integer bit and 15-bit fractional part (1.15)
  - Accessible Z result data for linear function is rational number with fixed data form of S4.11 (signed 4Q16)
- Truncation Error
  - The result of a CORDIC calculation may return an approximation due to truncation of LSBs
  - Good accuracy of the CORDIC calculated result data, especially in circular mode
- Interrupt
  - On completion of a calculation

– Interrupt enabling and corresponding flag

## 3.13 UART and UART1

The XC886/888 provides two Universal Asynchronous Receiver/Transmitter (UART and UART1) modules for full-duplex asynchronous reception/transmission. Both are also receive-buffered, i.e., they 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.

**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 modules can operate in the four modes shown in **Table 29**.

**Table 29 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 $f_{PCLK}/64$ [1)] |
| Mode 3: 9-bit shift UART | Variable |

1) For UART1 module, the baud rate is fixed at $f_{PCLK}/64$.

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 $f_{PCLK}/64$. For UART1 module, only $f_{PCLK}/64$ is available. The variable baud rate is set by the underflow rate on the dedicated baud-rate generator. For UART module, the variable baud rate alternatively can be set by the overflow rate on Timer 1.

### 3.13.1 Baud-Rate Generator

Both UART modules have their own dedicated baud-rate generator, which 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 30**.

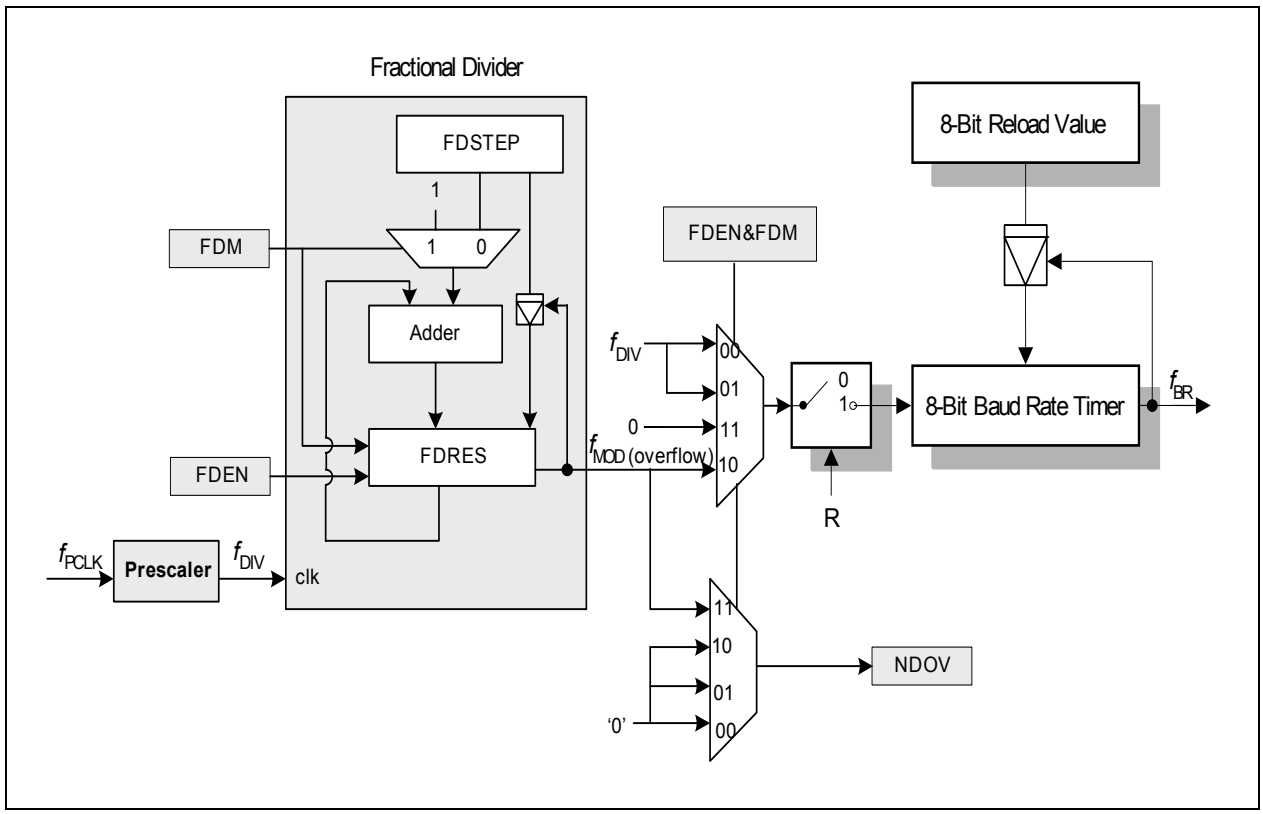



**Figure 30 Baud-rate Generator Circuitry**

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 3.14**.

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:

$$\text{baud rate} = \frac{f_{PCLK}}{16 \times 2^{BRPRE} \times (BR_VALUE + 1)} \quad \text{where} \quad 2^{BRPRE} \times (BR_VALUE + 1) > 1 \tag{3.5}$$

$$\text{baud rate} = \frac{f_{PCLK}}{16 \times 2^{BRPRE} \times (BR_VALUE + 1)} \times \frac{STEP}{256} \tag{3.6}$$

The maximum baud rate that can be generated is limited to $f_{PCLK}/32$. Hence, for a module clock of 24 MHz, the maximum achievable baud rate is 0.75 MBaud.

Standard LIN protocol can support a maximum baud rate of 20 kHz, 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 20 kHz to 115.2 kHz, the higher baud rates require the use of the fractional divider for greater accuracy.

**Table 30** 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 24 MHz is used.

**Table 30 Typical Baud rates for UART with Fractional Divider disabled**

| Baud rate | Prescaling Factor (2BRPRE) | Reload Value (BR_VALUE + 1) | Deviation Error |
|---|---|---|---|
| 19.2 kBaud | 1 (BRPRE=$000_B$) | 78 ($4E_H$) | 0.17 % |
| 9600 Baud | 1 (BRPRE=$000_B$) | 156 ($9C_H$) | 0.17 % |
| 4800 Baud | 2 (BRPRE=$001_B$) | 156 ($9C_H$) | 0.17 % |
| 2400 Baud | 4 (BRPRE=$010_B$) | 156 ($9C_H$) | 0.17 % |

The fractional divider allows baud rates of higher accuracy (lower deviation error) to be generated. **Table 31** 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 31 Deviation Error for UART with Fractional Divider enabled**

| $f_{PCLK}$ | Prescaling Factor (2BRPRE) | Reload Value (BR_VALUE + 1) | STEP | Deviation Error |
|---|---|---|---|---|
| 24 MHz | 1 | 10 ($A_H$) | 197 ($C5_H$) | +0.20 % |
| 12 MHz | 1 | 6 ($6_H$) | 236 ($EC_H$) | +0.03 % |
| 8 MHz | 1 | 4 ($4_H$) | 236 ($EC_H$) | +0.03 % |
| 6 MHz | 1 | 3 ($3_H$) | 236 ($EC_H$) | +0.03 % |

### 3.13.2 Baud Rate Generation using Timer 1

In UART modes 1 and 3 of UART module, 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:

$$\text{Mode 1, 3 baud rate} = \frac{2^{SMOD} \times f_{PCLK}}{32 \times 2 \times (256 - TH1)} \quad (3.7)$$

## 3.14 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 30**). 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:

$$f_{MOD} = f_{DIV} \times \frac{1}{256 - STEP} \quad (3.8)$$

## 3.15 LIN Protocol

The UART module can be used to support the Local Interconnect Network (LIN) protocol for both master and slave operations. The LIN baud rate detection feature, which consists of the hardware logic for Break and Synch Byte detection, 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.

*Note: The LIN baud rate detection feature is available for use only with UART. To use UART1 for LIN communication, software has to be implemented to detect the Break and Synch Byte.*

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 31**. 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

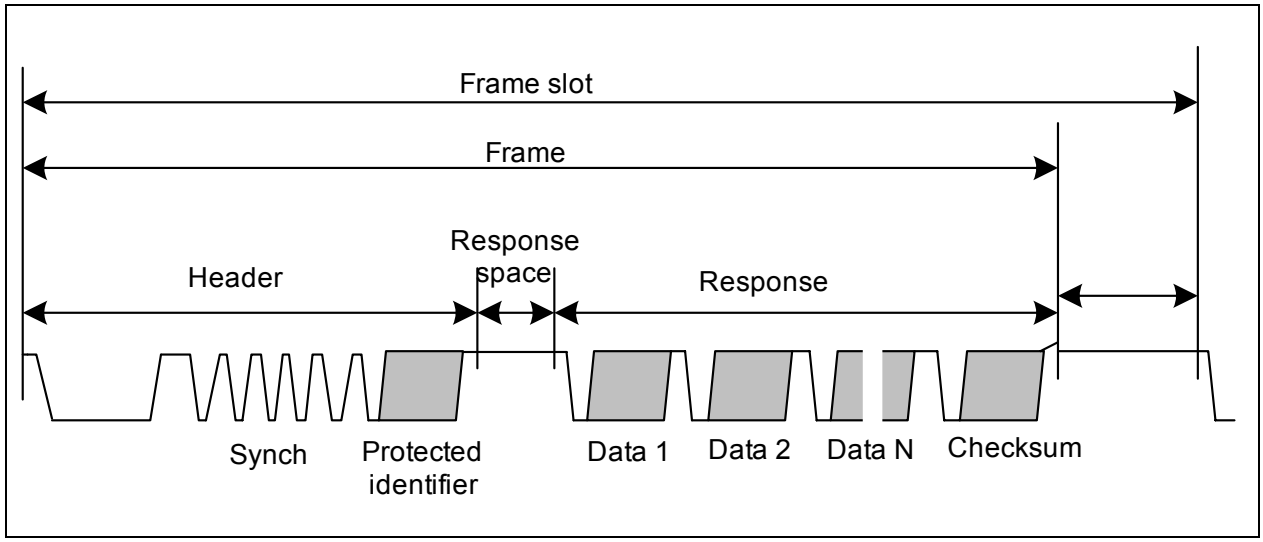



**Figure 31 Structure of LIN Frame**

### 3.15.1 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.*

## 3.16 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 32** shows the block diagram of the SSC.

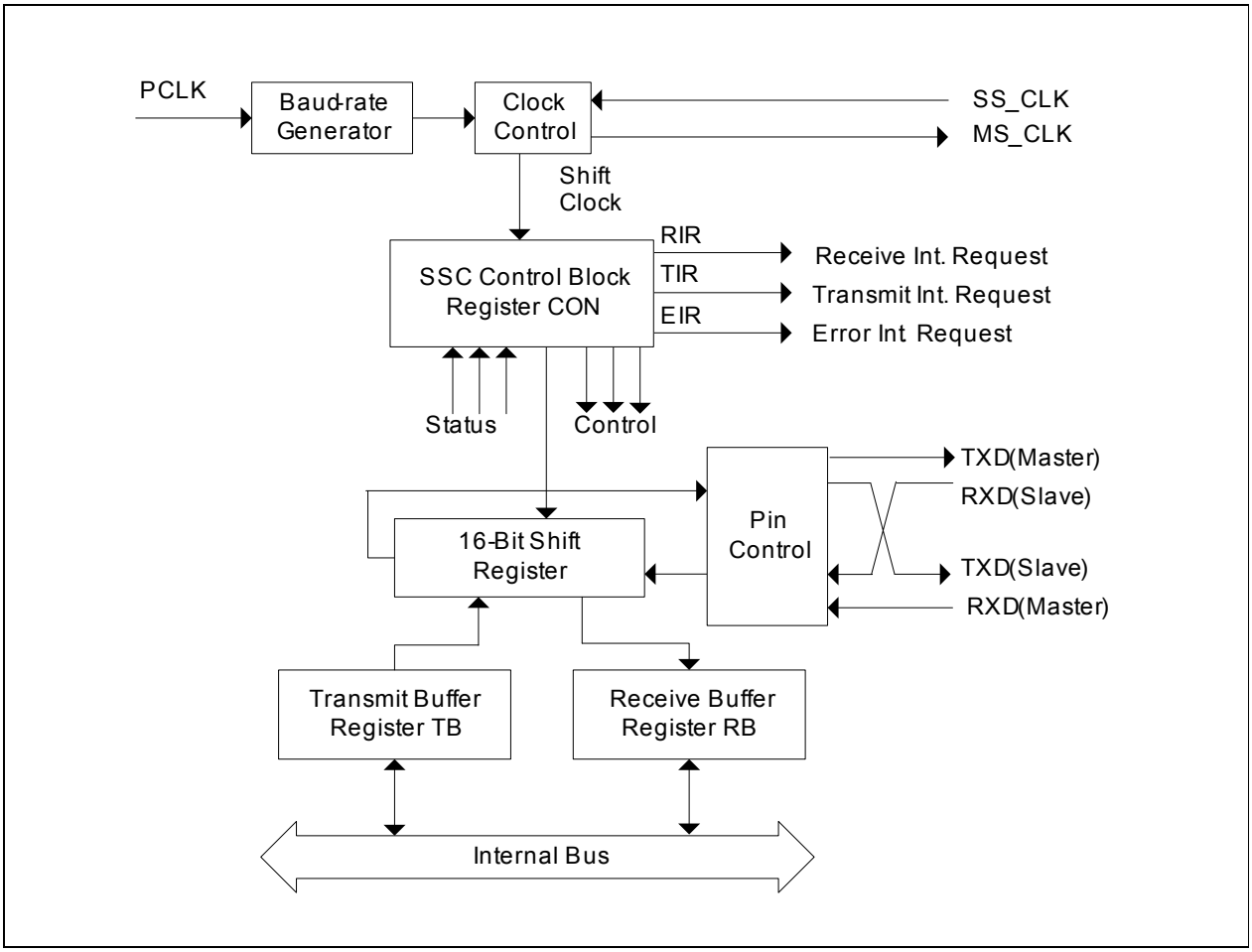



**Figure 32 SSC Block Diagram**

## 3.17 Timer 0 and Timer 1

Timer 0 and Timer 1 can function as both timers or counters. When functioning as a timer, Timer 0 and Timer 1 are incremented every machine cycle, i.e. every 2 input clocks (or 2 PCLKs). When functioning as a counter, Timer 0 and Timer 1 are incremented in response to a 1-to-0 transition (falling edge) at their respective external input pins, T0 or T1.

Timer 0 and 1 are fully compatible and can be configured in four different operating modes for use in a variety of applications, see **Table 32**. In modes 0, 1 and 2, the two timers operate independently, but in mode 3, their functions are specialized.

**Table 32 Timer 0 and Timer 1 Modes**

| Mode | Operation |
|---|---|
| **0** | **13-bit timer**<br>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**<br>The timer registers, TLx and THx, are concatenated to form a 16-bit counter. |
| **2** | **8-bit timer with auto-reload**<br>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**<br>The timer registers, TL0 and TH0, operate as two separate 8-bit counters. Timer 1 is halted and retains its count even if enabled. |

## 3.18 Timer 2 and Timer 21

Timer 2 and Timer 21 are 16-bit general purpose timers (THL2) that are fully compatible and have two modes of operation, a 16-bit auto-reload mode and a 16-bit one channel capture mode, see **Table 33**. As a timer, the timers count with an input clock of PCLK/12 (if prescaler is disabled). As a counter, they count 1-to-0 transitions on pin T2. In the counter mode, the maximum resolution for the count is PCLK/24 (if prescaler is disabled).

**Table 33 Timer 2 Modes**

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

## 3.19 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 33**.

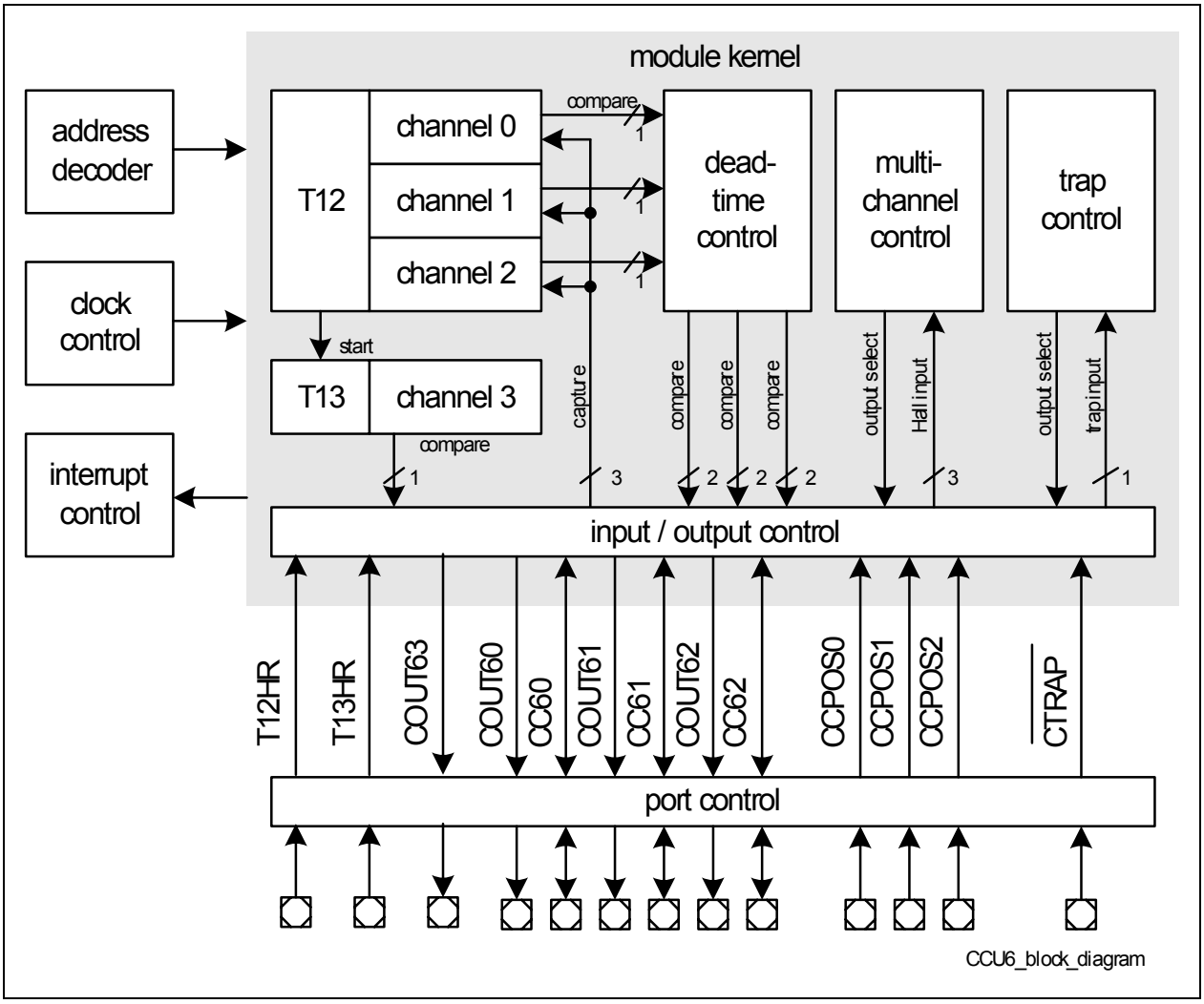



**Figure 33 CCU6 Block Diagram**

## 3.20 Controller Area Network (MultiCAN)

The MultiCAN module contains two Full-CAN nodes operating independently or exchanging data and remote frames via a gateway function. Transmission and reception of CAN frames is handled in accordance to CAN specification V2.0 B active. Each CAN node can receive and transmit standard frames with 11-bit identifiers as well as extended frames with 29-bit identifiers.

Both CAN nodes share a common set of message objects, where each message object may be individually allocated to one of the CAN nodes. Besides serving as a storage container for incoming and outgoing frames, message objects may be combined to build gateways between the CAN nodes or to setup a FIFO buffer.

The message objects are organized in double chained lists, where each CAN node has it's own list of message objects. A CAN node stores frames only into message objects that are allocated to the list of the CAN node. It only transmits messages from objects of this list. A powerful, command driven list controller performs all list operations.

The bit timings for the CAN nodes are derived from the peripheral clock ($f_{CAN}$) and are programmable up to a data rate of 1 MBaud. A pair of receive and transmit pins connects each CAN node to a bus transceiver.

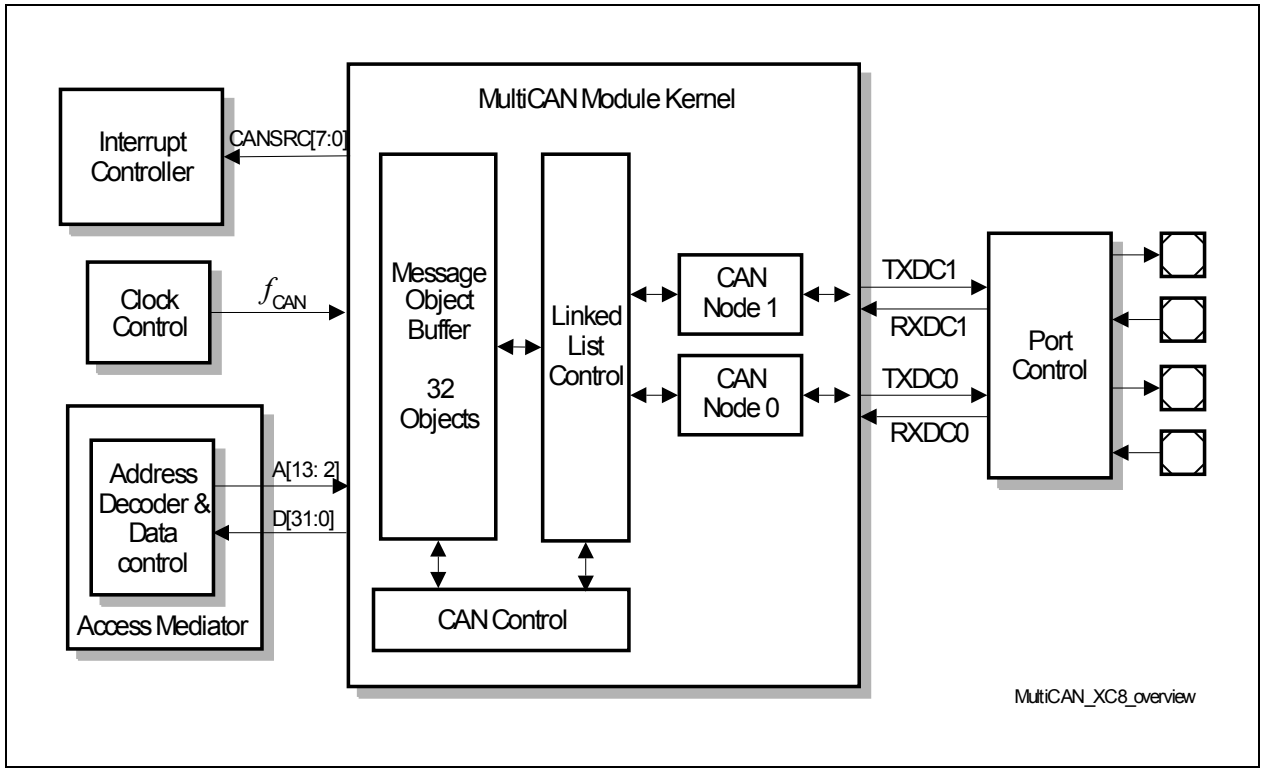



**Figure 34 Overview of the MultiCAN**

### Features

- Compliant to ISO 11898.

- CAN functionality according to CAN specification V2.0 B active.
- Dedicated control registers are provided for each CAN node.
- A data transfer rate up to 1 MBaud is supported.
- Flexible and powerful message transfer control and error handling capabilities are implemented.
- Advanced CAN bus bit timing analysis and baud rate detection can be performed for each CAN node via the frame counter.
- Full-CAN functionality: A set of 32 message objects can be individually
  - allocated (assigned) to any CAN node
  - configured as transmit or receive object
  - setup to handle frames with 11-bit or 29-bit identifier
  - counted or assigned a timestamp via a frame counter
  - configured to remote monitoring mode
- Advanced Acceptance Filtering:
  - Each message object provides an individual acceptance mask to filter incoming frames.
  - A message object can be configured to accept only standard or only extended frames or to accept both standard and extended frames.
  - Message objects can be grouped into 4 priority classes.
  - The selection of the message to be transmitted first can be performed on the basis of frame identifier, IDE bit and RTR bit according to CAN arbitration rules.
- Advanced Message Object Functionality:
  - Message Objects can be combined to build FIFO message buffers of arbitrary size, which is only limited by the total number of message objects.
  - Message objects can be linked to form a gateway to automatically transfer frames between 2 different CAN buses. A single gateway can link any two CAN nodes. An arbitrary number of gateways may be defined.
- Advanced Data Management:
  - The Message objects are organized in double chained lists.
  - List reorganizations may be performed any time, even during full operation of the CAN nodes.
  - A powerful, command driven list controller manages the organization of the list structure and ensures consistency of the list.
  - Message FIFOs are based on the list structure and can easily be scaled in size during CAN operation.
  - Static Allocation Commands offer compatibility with TwinCAN applications, which are not list based.
- Advanced Interrupt Handling:
  - Up to 8 interrupt output lines are available. Most interrupt requests can be individually routed to one of the 8 interrupt output lines.
  - Message postprocessing notifications can be flexibly aggregated into a dedicated register field of 64 notification bits.

## 3.21 Analog-to-Digital Converter

The XC886/888 includes a high-performance 10-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 or 10-bit resolution
  (TUE of ± 1 LSB and ± 2 LSB, respectively)
- 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

### 3.21.1 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.

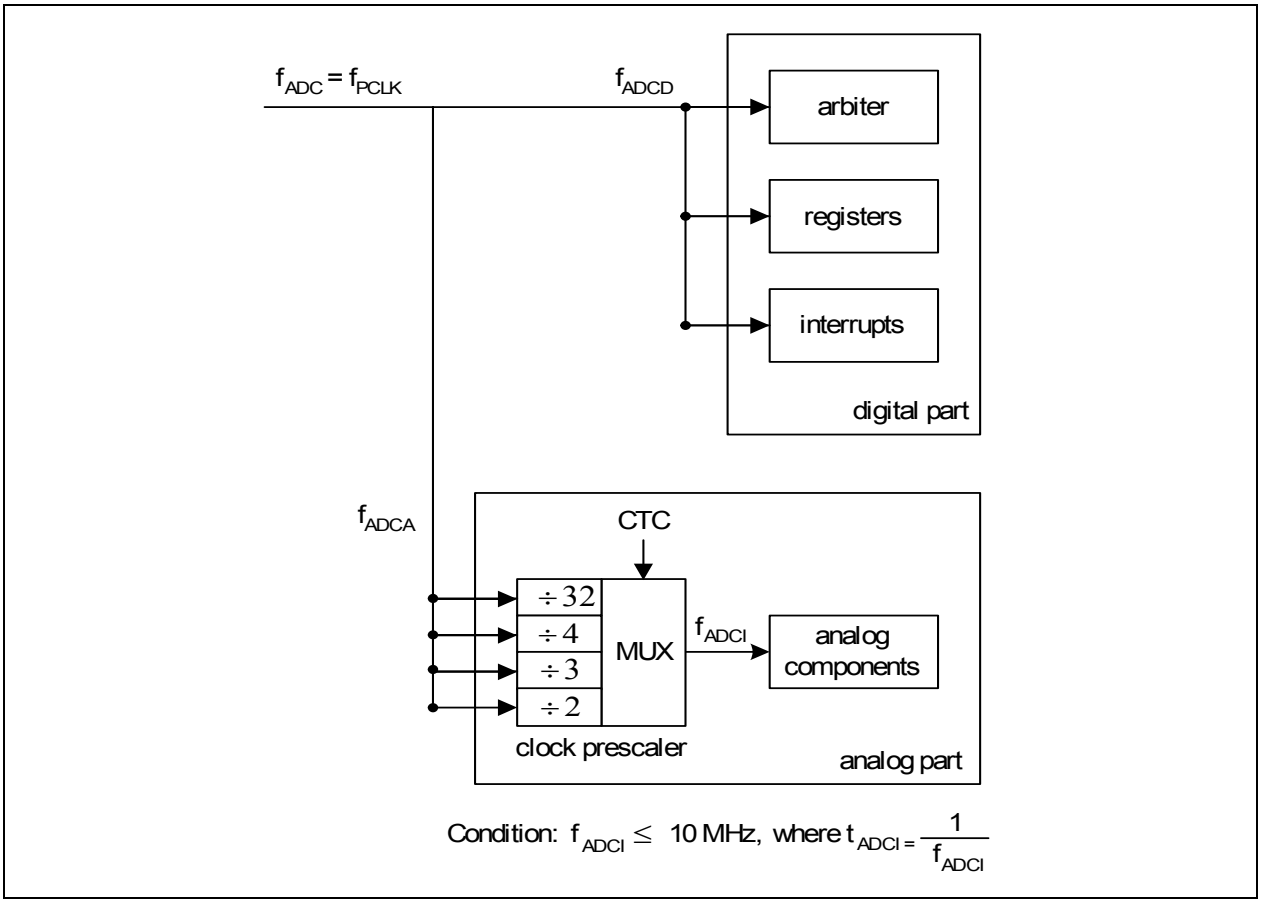



**Figure 35 ADC Clocking Scheme**

For module clock $f_{ADC}$ = 24 MHz, the analog clock $f_{ADCI}$ frequency can be selected as shown in **Table 34**.

**Table 34 $f_{ADCI}$ Frequency Selection**

| Module Clock $f_{ADC}$ | CTC | Prescaling Ratio | Analog Clock $f_{ADCI}$ |
|---|---|---|---|
| 24 MHz | $00_B$ | ÷ 2 | 12 MHz (N.A) |
| | $01_B$ | ÷ 3 | 8 MHz |
| | $10_B$ | ÷ 4 | 6 MHz |
| | $11_B$ (default) | ÷ 32 | 750 kHz |

As $f_{ADCI}$ cannot exceed 10 MHz, bit field CTC should not be set to $00_B$ when $f_{ADC}$ is 24 MHz. During slow-down mode where $f_{ADC}$ may be reduced to 12 MHz, 6 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.

### 3.21.2 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 ($t_{WR}$)

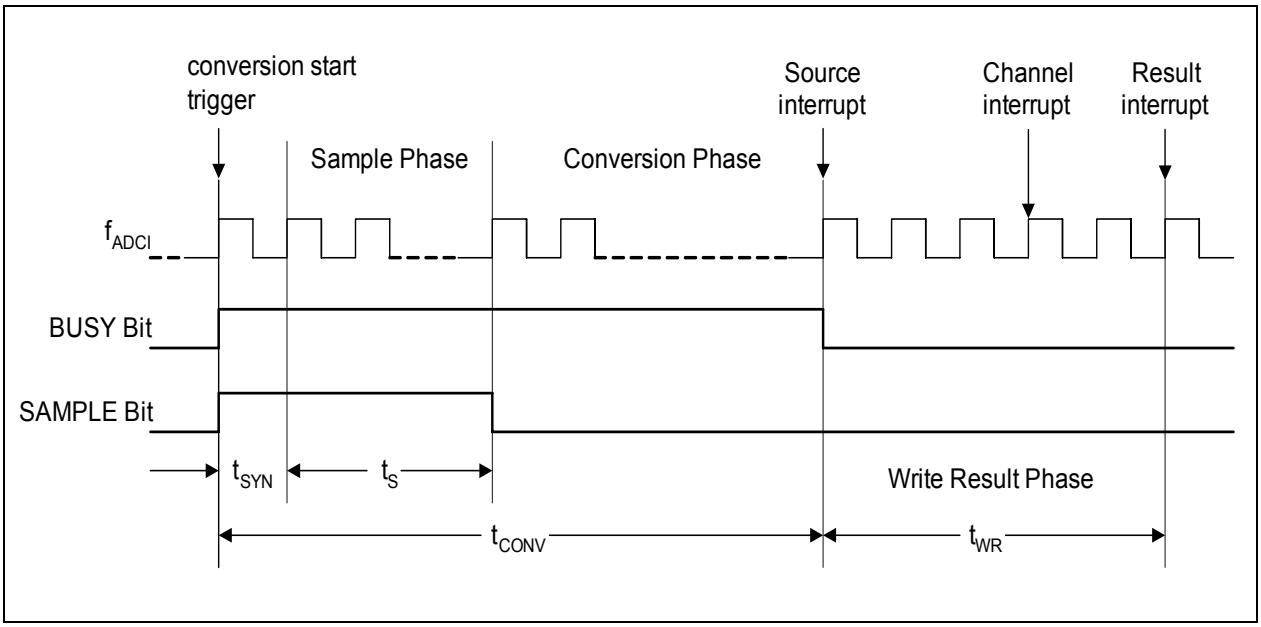



**Figure 36 ADC Conversion Timing**

## 3.22 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 on address range within the Program Memory
- Set breakpoints on internal RAM address range
- Support unlimited software breakpoints in Flash/RAM code region
- Process external breaks via JTAG and upon activating a dedicated pin
- Step through the program code

The OCDS functional blocks are shown in **Figure 37**. 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. The dedicated MBC pin is used for external configuration and debugging control.

*Note: All the debug functionality described here can normally be used only after XC886/888 has been started in OCDS mode.*

1) The pins of the JTAG port can be assigned to either the primary port (Port 0) or either of the secondary ports (Ports 1 and 2/Port 5).
User must set the JTAG pins (TCK and TDI) as input during connection with the OCDS system.

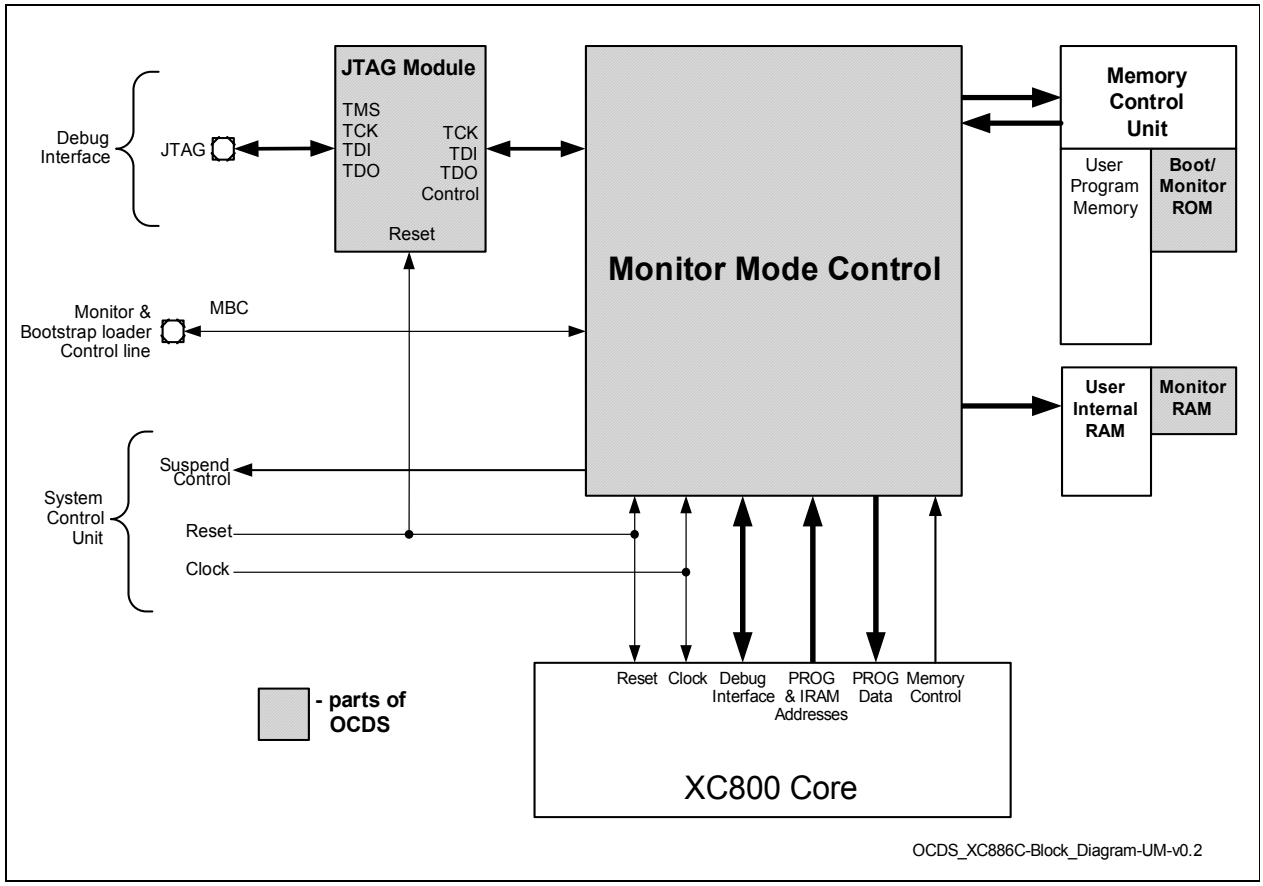



**Figure 37 OCDS Block Diagram**

### 3.22.1 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 register contents for the XC886/888 Flash devices are given in **Table 35**.

**Table 35 JTAG ID Summary**

| Device Type | Device Name | JTAG ID |
|---|---|---|
| Flash | XC886/888*-8FF | 1012 0083$_H$ |
| | XC886/888*-6FF | 1012 5083$_H$ |
| ROM | XC886/888*-8RF | 1013 C083$_H$ |
| | XC886/888*-6RF | 1013 D083$_H$ |

*Note: The asterisk (*) above denotes all possible device configurations.*

## 3.23 Chip Identification Number

The XC886/888 identity (ID) register is located at Page 1 of address $B3_H$. The value of ID register is $09_H$ for Flash devices and $22_H$ for ROM devices. 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.

Two methods are provided to read a device's chip identification number:

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

**Table 36** lists the chip identification numbers of available XC886/888 Flash and ROM device variants.

**Table 36 Chip Identification Number**

| Product Variant | Chip Identification Number | | |
|---|---|---|---|
| | AA-Step | AB-Step | AC-Step |
| **Flash Devices** | | | |
| XC886CLM-8FFA 3V3 | - | $09500102_H$ | $0B500102_H$ |
| XC888CLM-8FFA 3V3 | - | $09500103_H$ | $0B500103_H$ |
| XC886LM-8FFA 3V3 | - | $09500122_H$ | $0B500122_H$ |
| XC888LM-8FFA 3V3 | - | $09500123_H$ | $0B500123_H$ |
| XC886CLM-6FFA 3V3 | - | $09551502_H$ | $0B551502_H$ |
| XC888CLM-6FFA 3V3 | - | $09551503_H$ | $0B551503_H$ |
| XC886LM-6FFA 3V3 | - | $09551522_H$ | $0B551522_H$ |
| XC888LM-6FFA 3V3 | - | $09551523_H$ | $0B551523_H$ |
| XC886CM-8FFA 3V3 | - | $09580102_H$ | $0B580102_H$ |
| XC888CM-8FFA 3V3 | - | $09580103_H$ | $0B580103_H$ |
| XC886C-8FFA 3V3 | - | $09580142_H$ | $0B580142_H$ |
| XC888C-8FFA 3V3 | - | $09580143_H$ | $0B580143_H$ |
| XC886-8FFA 3V3 | - | $09580162_H$ | $0B580162_H$ |
| XC888-8FFA 3V3 | - | $09580163_H$ | $0B580163_H$ |
| XC886CM-6FFA 3V3 | - | $095D1502_H$ | $0B5D1502_H$ |
| XC888CM-6FFA 3V3 | - | $095D1503_H$ | $0B5D1503_H$ |
| XC886C-6FFA 3V3 | - | $095D1542_H$ | $0B5D1542_H$ |
| XC888C-6FFA 3V3 | - | $095D1543_H$ | $0B5D1543_H$ |

**Table 36 Chip Identification Number** (cont'd)

| Product Variant | Chip Identification Number | | |
|---|---|---|---|
| | **AA-Step** | **AB-Step** | **AC-Step** |
| XC886-6FFA 3V3 | - | $095D1562_H$ | $0B5D1562_H$ |
| XC888-6FFA 3V3 | - | $095D1563_H$ | $0B5D1563_H$ |
| XC886CLM-8FFA 5V | - | $09900102_H$ | $0B900102_H$ |
| XC888CLM-8FFA 5V | - | $09900103_H$ | $0B900103_H$ |
| XC886LM-8FFA 5V | - | $09900122_H$ | $0B900122_H$ |
| XC888LM-8FFA 5V | - | $09900123_H$ | $0B900123_H$ |
| XC886CLM-6FFA 5V | - | $09951502_H$ | $0B951502_H$ |
| XC888CLM-6FFA 5V | - | $09951503_H$ | $0B951503_H$ |
| XC886LM-6FFA 5V | - | $09951522_H$ | $0B951522_H$ |
| XC888LM-6FFA 5V | - | $09951523_H$ | $0B951523_H$ |
| XC886CM-8FFA 5V | - | $09980102_H$ | $0B980102_H$ |
| XC888CM-8FFA 5V | - | $09980103_H$ | $0B980103_H$ |
| XC886C-8FFA 5V | - | $09980142_H$ | $0B980142_H$ |
| XC888C-8FFA 5V | - | $09980143_H$ | $0B980143_H$ |
| XC886-8FFA 5V | - | $09980162_H$ | $0B980162_H$ |
| XC888-8FFA 5V | - | $09980163_H$ | $0B980163_H$ |
| XC886CM-6FFA 5V | - | $099D1502_H$ | $0B9D1502_H$ |
| XC888CM-6FFA 5V | - | $099D1503_H$ | $0B9D1503_H$ |
| XC886C-6FFA 5V | - | $099D1542_H$ | $0B9D1542_H$ |
| XC888C-6FFA 5V | - | $099D1543_H$ | $0B9D1543_H$ |
| XC886-6FFA 5V | - | $099D1562_H$ | $0B9D1562_H$ |
| XC888-6FFA 5V | - | $099D1563_H$ | $0B9D1563_H$ |
| **ROM Devices** | | | |
| XC886CLM-8RFA 3V3 | $22400502_H$ | - | - |
| XC888CLM-8RFA 3V3 | $22400503_H$ | - | - |
| XC886LM-8RFA 3V3 | $22400522_H$ | - | - |
| XC888LM-8RFA 3V3 | $22400523_H$ | - | - |
| XC886CLM-6RFA 3V3 | $22411502_H$ | - | - |
| XC888CLM-6RFA 3V3 | $22411503_H$ | - | - |

**Table 36 Chip Identification Number** (cont'd)

| Product Variant | Chip Identification Number | | |
|---|---|---|---|
| | AA-Step | AB-Step | AC-Step |
| XC886LM-6RFA 3V3 | $22411522_H$ | - | - |
| XC888LM-6RFA 3V3 | $22411523_H$ | - | - |
| XC886CM-8RFA 3V3 | $22480502_H$ | - | - |
| XC888CM-8RFA 3V3 | $22480503_H$ | - | - |
| XC886C-8RFA 3V3 | $22480542_H$ | - | - |
| XC888C-8RFA 3V3 | $22480543_H$ | - | - |
| XC886-8RFA 3V3 | $22480562_H$ | - | - |
| XC888-8RFA 3V3 | $22480563_H$ | - | - |
| XC886CM-6RFA 3V3 | $22491502_H$ | - | - |
| XC888CM-6RFA 3V3 | $22491503_H$ | - | - |
| XC886C-6RFA 3V3 | $22491542_H$ | - | - |
| XC888C-6RFA 3V3 | $22491543_H$ | - | - |
| XC886-6RFA 3V3 | $22491562_H$ | - | - |
| XC888-6RFA 3V3 | $22491563_H$ | - | - |
| XC886CLM-8RFA 5V | $22800502_H$ | - | - |
| XC888CLM-8RFA 5V | $22800503_H$ | - | - |
| XC886LM-8RFA 5V | $22800522_H$ | - | - |
| XC888LM-8RFA 5V | $22800523_H$ | - | - |
| XC886CLM-6RFA 5V | $22811502_H$ | - | - |
| XC888CLM-6RFA 5V | $22811503_H$ | - | - |
| XC886LM-6RFA 5V | $22811522_H$ | - | - |
| XC888LM-6RFA 5V | $22811523_H$ | - | - |
| XC886CM-8RFA 5V | $22880502_H$ | - | - |
| XC888CM-8RFA 5V | $22880503_H$ | - | - |
| XC886C-8RFA 5V | $22880542_H$ | - | - |
| XC888C-8RFA 5V | $22880543_H$ | - | - |
| XC886-8RFA 5V | $22880562_H$ | - | - |
| XC888-8RFA 5V | $22880563_H$ | - | - |
| XC886CM-6RFA 5V | $22891502_H$ | - | - |

**Table 36 Chip Identification Number** (cont'd)

| Product Variant | Chip Identification Number | | |
|---|---|---|---|
| | **AA-Step** | **AB-Step** | **AC-Step** |
| XC888CM-6RFA 5V | $22891503_H$ | - | - |
| XC886C-6RFA 5V | $22891542_H$ | - | - |
| XC888C-6RFA 5V | $22891543_H$ | - | - |
| XC886-6RFA 5V | $22891562_H$ | - | - |
| XC888-6RFA 5V | $22891563_H$ | - | - |

# 4 Electrical Parameters

Chapter 4 provides the characteristics of the electrical parameters which are implementation-specific for the XC886/888.

## 4.1 General Parameters

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

### 4.1.1 Parameter Interpretation

The parameters listed in this section represent partly the characteristics of the XC886/888 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 XC886/888 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 XC886/888 is designed in.

### 4.1.2 Absolute Maximum Rating

Maximum ratings are the extreme limits to which the XC886/888 can be subjected to without permanent damage.

**Table 4-1 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 | 1) |
| Junction temperature | $T_J$ | -40 | 150 | °C | under bias1) |
| Voltage on power supply pin with respect to $V_{SS}$ | $V_{DDP}$ | -0.5 | 6 | V | 1) |
| Voltage on any pin with respect to $V_{SS}$ | $V_{IN}$ | -0.5 | $V_{DDP}$ + 0.5 or max. 6 | V | whichever is lower1) |
| Input current on any pin during overload condition | $I_{IN}$ | -10 | 10 | mA | 1) |
| Absolute sum of all input currents during overload condition | $\Sigma\|I_{IN}\|$ | – | 50 | mA | 1) |

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

*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 XC886/888. All parameters mentioned in the following table refer to these operating conditions, unless otherwise noted.

**Table 37 Operating Condition Parameters**

| Parameter | Symbol | Limit Values | | Unit | Notes/ Conditions |
|---|---|---|---|---|---|
| | | min. | max. | | |
| Digital power supply voltage | $V_{DDP}$ | 4.5 | 5.5 | V | 5V Device |
| Digital power supply voltage | $V_{DDP}$ | 3.0 | 3.6 | V | 3.3V Device |
| Digital ground voltage | $V_{SS}$ | 0 | | V | |
| Digital core supply voltage | $V_{DDC}$ | 2.3 | 2.7 | V | |
| System Clock Frequency[1)] | $f_{SYS}$ | 88.8 | 103.2 | MHz | |
| Ambient temperature | $T_A$ | -40 | 85 | °C | SAF-XC886/888... |
| | | -40 | 125 | °C | SAK-XC886/888... |

1) $f_{SYS}$ is the PLL output clock. During normal operating mode, CPU clock is $f_{SYS}$ / 4. Please refer to **Figure 26** for detailed description.

## 4.2 DC Parameters

The electrical characteristics of the DC Parameters are detailed in this section.

### 4.2.1 Input/Output Characteristics

Table 38 provides the characteristics of the input/output pins of the XC886/888.

**Table 38 Input/Output Characteristics (Operating Conditions apply)**

| Parameter | Symbol | | Limit Values | | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| | | | min. | max. | | |
| **$V_{DDP}$ = 5 V Range** | | | | | | |
| Output low voltage | $V_{OL}$ | CC | – | 1.0 | V | $I_{OL}$ = 15 mA |
| | | | – | 1.0 | V | $I_{OL}$ = 5 mA, current into all pins > 60 mA |
| | | | – | 0.4 | V | $I_{OL}$ = 5 mA, current into all pins ≤ 60 mA |
| Output high voltage | $V_{OH}$ | CC | $V_{DDP}$ - 1.0 | – | V | $I_{OH}$ = -15 mA |
| | | | $V_{DDP}$ - 1.0 | – | V | $I_{OH}$ = -5 mA, current from all pins > 60 mA |
| | | | $V_{DDP}$ - 0.4 | – | V | $I_{OH}$ = -5 mA, current from all pins ≤ 60 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 |

**Table 38 Input/Output Characteristics (Operating Conditions apply)** (cont'd)

| Parameter | Symbol | | Limit Values | | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| | | | min. | max. | | |
| Input high voltage on $\overline{\text{RESET}}$ 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 on port pins | $HYSP$ | CC | 0.07 × $V_{DDP}$ | – | V | CMOS Mode[1] |
| Input Hysteresis on XTAL1 | $HYSX$ | CC | 0.07 × $V_{DDC}$ | – | V | [1] |
| Input low voltage at XTAL1 | $V_{ILX}$ | SR | $V_{SS}$ - 0.5 | 0.3 × $V_{DDC}$ | V | |
| Input high voltage at XTAL1 | $V_{IHX}$ | SR | 0.7 × $V_{DDC}$ | $V_{DDC}$ + 0.5 | V | |
| Pull-up current | $I_{PU}$ | SR | – | -10 | µA | $V_{IHP,min}$ |
| | | | -150 | – | µA | $V_{ILP,max}$ |
| Pull-down current | $I_{PD}$ | SR | – | 10 | µA | $V_{ILP,max}$ |
| | | | 150 | – | µA | $V_{IHP,min}$ |
| Input leakage current | $I_{OZ1}$ | CC | -1 | 1 | µA | $0 < V_{IN} < V_{DDP}$, $T_A \leq 125°C$[2] |
| Input current at XTAL1 | $I_{ILX}$ | CC | -10 | 10 | µA | |
| Overload current on any pin | $I_{OV}$ | SR | -5 | 5 | mA | |
| Absolute sum of overload currents | $\Sigma\|I_{OV}\|$ | SR | – | 25 | mA | [3] |
| Voltage on any pin during $V_{DDP}$ power off | $V_{PO}$ | SR | – | 0.3 | V | [4] |
| Maximum current per pin (excluding $V_{DDP}$ and $V_{SS}$) | $I_M$ SR | SR | – | 15 | mA | |
| Maximum current for all pins (excluding $V_{DDP}$ and $V_{SS}$) | $\Sigma\|I_M\|$ | SR | – | 90 | mA | |
| Maximum current into $V_{DDP}$ | $I_{MVDDP}$ | SR | – | 120 | mA | [3] |

**Table 38 Input/Output Characteristics (Operating Conditions apply)** (cont'd)

| Parameter | Symbol | | Limit Values | | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| | | | min. | max. | | |
| Maximum current out of $V_{SS}$ | $I_{MVSS}$ | SR | – | 120 | mA | [3] |
| **$V_{DDP}$ = 3.3 V 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 |
| 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 | 0.75 × $V_{DDP}$ | – | V | CMOS Mode |
| Input Hysteresis | $HYS$ | CC | 0.03 × $V_{DDP}$ | – | V | CMOS Mode[1] |
| Input Hysteresis on XTAL1 | $HYSX$ | CC | 0.07 × $V_{DDC}$ | – | V | [1] |
| Input low voltage at XTAL1 | $V_{ILX}$ | SR | $V_{SS}$ - 0.5 | 0.3 × $V_{DDC}$ | V | |

**Table 38 Input/Output Characteristics (Operating Conditions apply)** (cont'd)

| Parameter | Symbol | | Limit Values | | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| | | | min. | max. | | |
| Input high voltage at XTAL1 | $V_{IHX}$ | SR | 0.7 × $V_{DDC}$ | $V_{DDC}$ + 0.5 | V | |
| Pull-up current | $I_{PU}$ | SR | – | -5 | µA | $V_{IHP,min}$ |
| | | | -50 | – | µA | $V_{ILP,max}$ |
| Pull-down current | $I_{PD}$ | SR | – | 5 | µA | $V_{ILP,max}$ |
| | | | 50 | – | µA | $V_{IHP,min}$ |
| Input leakage current | $I_{OZ1}$ | CC | -1 | 1 | µA | $0 < V_{IN} < V_{DDP}$, $T_A \leq 125°C$ [2] |
| Input current at XTAL1 | $I_{ILX}$ | CC | - 10 | 10 | µA | |
| Overload current on any pin | $I_{OV}$ | SR | -5 | 5 | mA | |
| Absolute sum of overload currents | $\Sigma\|I_{OV}\|$ | SR | – | 25 | mA | [3] |
| Voltage on any pin during $V_{DDP}$ power off | $V_{PO}$ | SR | – | 0.3 | V | [4] |
| Maximum current per pin (excluding $V_{DDP}$ and $V_{SS}$) | $I_M$ SR | SR | – | 15 | mA | |
| Maximum current for all pins (excluding $V_{DDP}$ and $V_{SS}$) | $\Sigma\|I_M\|$ | SR | – | 90 | mA | |
| Maximum current into $V_{DDP}$ | $I_{MVDDP}$ | SR | – | 120 | mA | [3] |
| Maximum current out of $V_{SS}$ | $I_{MVSS}$ | SR | – | 120 | mA | [3] |

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

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

4) 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

Table 39 provides the characteristics of the supply threshold in the XC886/888.

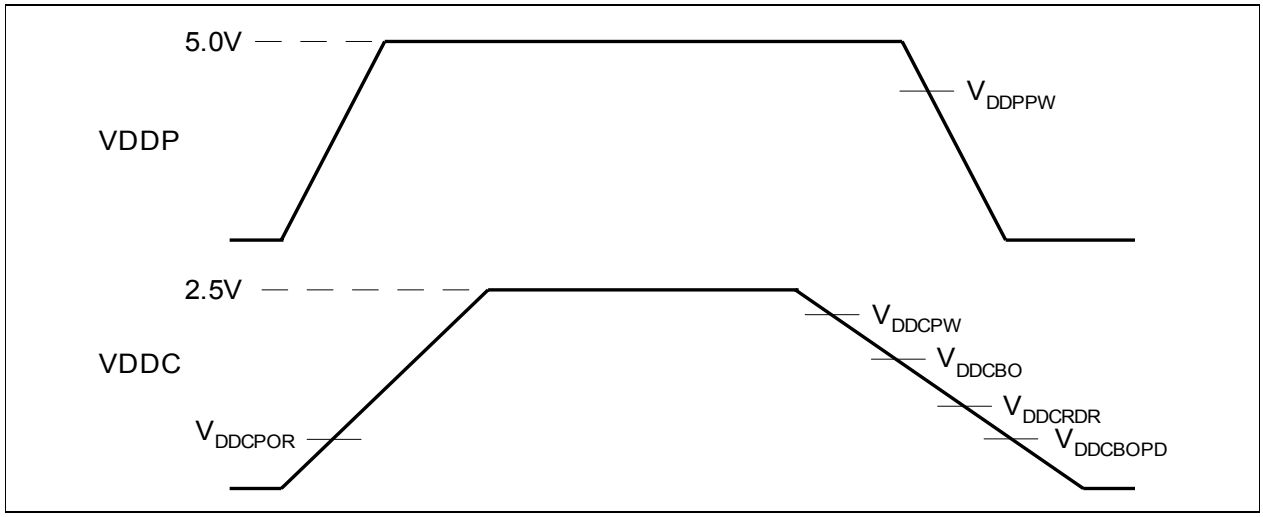



**Figure 38 Supply Threshold Parameters**

**Table 39 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 | 2.2 | 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[3] | $V_{DDPPW}$ | CC | 3.4 | 4.0 | 4.6 | V |
| Power-on reset voltage[2][4] | $V_{DDCPOR}$ | 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) Detection is enabled for external power supply of 5.0V.
Detection must be disabled for external power supply of 3.3V.

4) 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. But in this case, the analog parameters 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 40 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 | [1] |
| Analog reference ground | $V_{AGND}$ | SR | $V_{SS}$ - 0.05 | $V_{SS}$ | $V_{AREF}$ - 1 | V | [1] |
| Analog input voltage range | $V_{AIN}$ | SR | $V_{AGND}$ | – | $V_{AREF}$ | V | |
| ADC clocks | $f_{ADC}$ | | – | 24 | 25.8 | MHz | module clock[1] |
| | $f_{ADCI}$ | | – | – | 10 | MHz | internal analog clock[1] See **Figure 35** |
| Sample time | $t_S$ | CC | (2 + INPCR0.STC) × $t_{ADCI}$ | | | µs | [1] |
| Conversion time | $t_C$ | CC | See **Section 4.2.3.1** | | | µs | [1] |
| Total unadjusted error | \|TUE\| | CC | – | – | 1 | LSB | 8-bit conversion[2] |
| | | | – | – | 2 | LSB | 10-bit conversion[2] |
| Differential Nonlinearity | $\|EA_{DNL}\|$ | CC | – | 1 | – | LSB | 10-bit conversion[1] |
| Integral Nonlinearity | $\|EA_{INL}\|$ | CC | – | 1 | – | LSB | 10-bit conversion[1] |
| Offset | $\|EA_{OFF}\|$ | CC | – | 1 | – | LSB | 10-bit conversion[1] |
| Gain | $\|EA_{GAIN}\|$ | CC | – | 1 | – | LSB | 10-bit conversion[1] |
| Overload current coupling factor for analog inputs | $K_{OVA}$ | CC | – | – | $1.0 \times 10^{-4}$ | – | $I_{OV} > 0$[1][3] |
| | | | – | – | $1.5 \times 10^{-3}$ | – | $I_{OV} < 0$[1][3] |

**Table 40 ADC Characteristics (Operating Conditions apply; $V_{DDP}$ = 5V Range)**

| Parameter | Symbol | | Limit Values | | | Unit | Test Conditions/ Remarks |
|---|---|---|---|---|---|---|---|
| | | | min. | typ . | max. | | |
| Overload current coupling factor for digital I/O pins | $K_{OVD}$ | CC | – | – | 5.0 x $10^{-3}$ | – | $I_{OV} > 0$ [1)3)] |
| | | | – | – | 1.0 x $10^{-2}$ | – | $I_{OV} < 0$ [1)3)] |
| Switched capacitance at the reference voltage input | $C_{AREFSW}$ | CC | – | 10 | 20 | pF | [1)4)] |
| Switched capacitance at the analog voltage inputs | $C_{AINSW}$ | CC | – | 5 | 7 | pF | [1)5)] |
| Input resistance of the reference input | $R_{AREF}$ | CC | – | 1 | 2 | kΩ | [1)] |
| Input resistance of the selected analog channel | $R_{AIN}$ | CC | – | 1 | 1.5 | kΩ | [1)] |

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

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

3) An overload current ($I_{OV}$) through a pin injects a certain error current ($I_{INJ}$) into the adjacent pins. This error current adds to the respective pin's leakage current ($I_{OZ}$). The amount of error current depends on the overload current and is defined by the overload coupling factor $K_{OV}$. The polarity of the injected error current is inverse compared to the polarity of the overload current that produces it. The total current through a pin is $|I_{TOT}| = |I_{OZ1}| + (|I_{OV}| \times K_{OV})$. The additional error current may distort the input voltage on analog inputs.

4) 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.

5) 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 $V_{AREF}/2$.

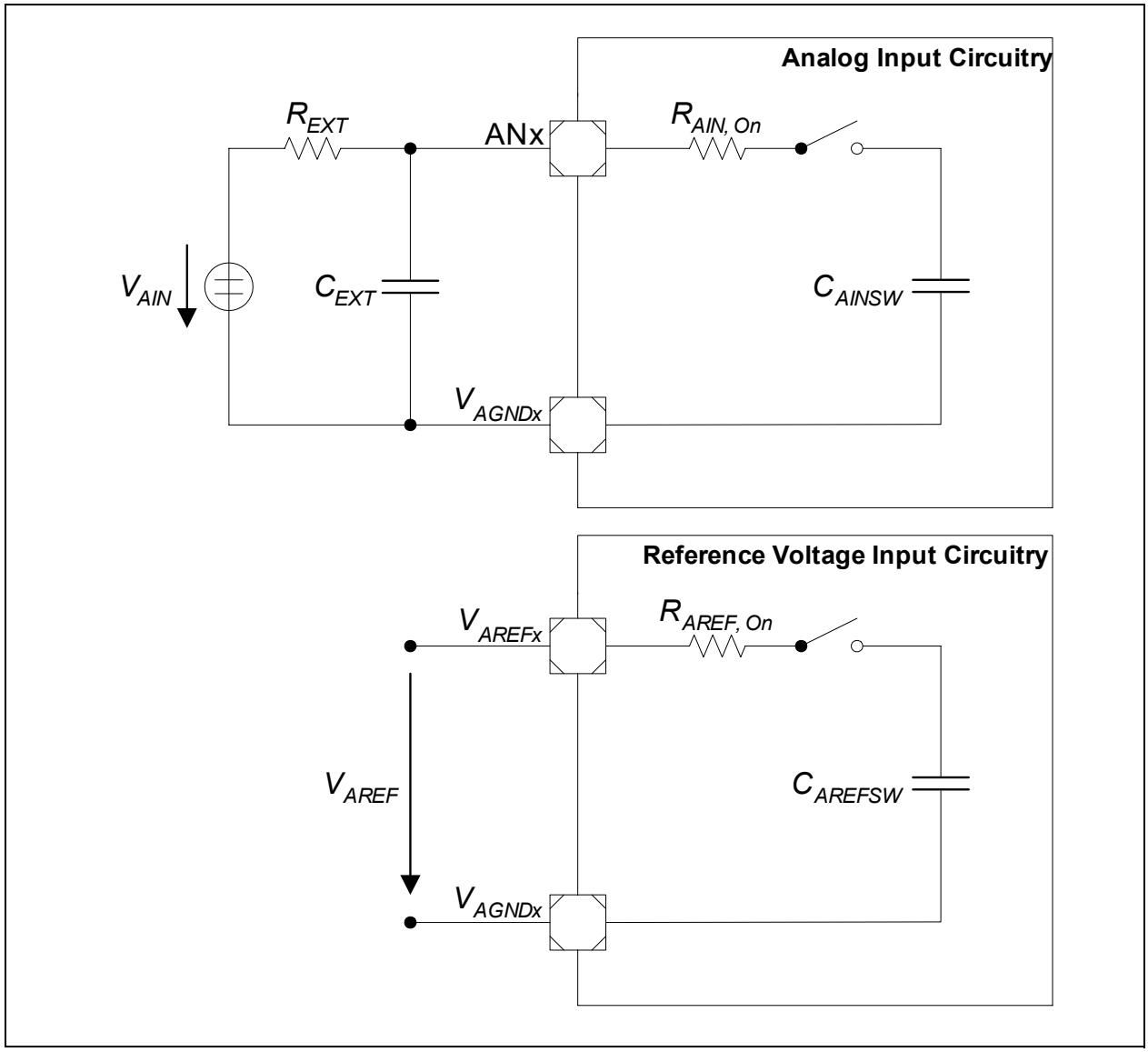



**Figure 39 ADC Input Circuits**

#### 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-bit and 10-bit conversion respectively),

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

### 4.2.4 Power Supply Current

Table 41, Table 42, Table 43 and Table 44 provide the characteristics of the power supply current in the XC886/888.

**Table 41 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 | $I_{DDP}$ | 27.2 | 32.8 | mA | Flash Device[3] |
| | | 24.3 | 29.8 | mA | ROM Device[3] |
| Idle Mode | $I_{DDP}$ | 21.1 | 25.3 | mA | Flash Device[4] |
| | | 18.2 | 21.6 | mA | ROM Device[4] |
| Active Mode with slow-down enabled | $I_{DDP}$ | 14.1 | 17.0 | mA | Flash Device[5] |
| | | 11.9 | 14.3 | mA | ROM Device[5] |
| Idle Mode with slow-down enabled | $I_{DDP}$ | 11.7 | 15.0 | mA | Flash Device[6] |
| | | 9.7 | 11.9 | mA | ROM Device[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 24 MHz(set by on-chip oscillator of 9.6 MHz and NDIV in PLL_CON to $1001_B$), $\overline{\text{RESET}}$ = $V_{DDP}$, 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 24 MHz, $\overline{\text{RESET}}$ = $V_{DDP}$, 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 8 MHz by setting CLKREL in CMCON to $0110_B$, $\overline{\text{RESET}}$ = $V_{DDP}$, 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 8 MHz by setting CLKREL in CMCON to $0110_B$, $\overline{\text{RESET}}$ = $V_{DDP}$, no load on ports.

**Table 42 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 | $I_{PDP}$ | 1 | 10 | µA | $T_A$ = + 25 °C[3)4)] |
| | | - | 30 | 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) $I_{PDP}$ has a maximum value of 200 µA at $T_A$ = + 125 °C.

4) $I_{PDP}$ is measured with: $\overline{\text{RESET}}$ = $V_{DDP}$, $V_{AGND}$= $V_{SS}$, 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 subjected to production test, verified by design/characterization.

**Table 43 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 | $I_{DDP}$ | 25.6 | 31.0 | mA | Flash Device[3] |
| | | 23.4 | 28.6 | mA | ROM Device[3] |
| Idle Mode | $I_{DDP}$ | 19.9 | 24.7 | mA | Flash Device[4] |
| | | 17.5 | 20.7 | mA | ROM Device[4] |
| Active Mode with slow-down enabled | $I_{DDP}$ | 13.3 | 16.2 | mA | Flash Device[5] |
| | | 11.5 | 13.7 | mA | ROM Device[5] |
| Idle Mode with slow-down enabled | $I_{DDP}$ | 11.1 | 14.4 | mA | Flash Device[6] |
| | | 9.3 | 11.4 | mA | ROM Device[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 24 MHz(set by on-chip oscillator of 9.6 MHz and NDIV in PLL_CON to $1001_B$), $\overline{\text{RESET}}$ = $V_{DDP}$, 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 24 MHz, $\overline{\text{RESET}}$ = $V_{DDP}$, 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 8 MHz by setting CLKREL in CMCON to $0110_B$, $\overline{\text{RESET}}$ = $V_{DDP}$, 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 8 MHz by setting CLKREL in CMCON to $0110_B$,, $\overline{\text{RESET}}$ = $V_{DDP}$, no load on ports.

**Table 44 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 | $I_{PDP}$ | 1 | 10 | µA | $T_A$ = + 25 °C[3][4] |
| | | - | 30 | 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) $I_{PDP}$ has a maximum value of 200 µA at $T_A$ = + 125 °C.

4) $I_{PDP}$ is measured with: $\overline{\text{RESET}}$ = $V_{DDP}$, $V_{AGND}$= $V_{SS}$, 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 subjected to production test, verified by design/characterization.

## 4.3 AC Parameters

The electrical characteristics of the AC Parameters are detailed in this section.

### 4.3.1 Testing Waveforms

The testing waveforms for rise/fall time, output delay and output high impedance are shown in **Figure 40**, **Figure 41** and **Figure 42**.

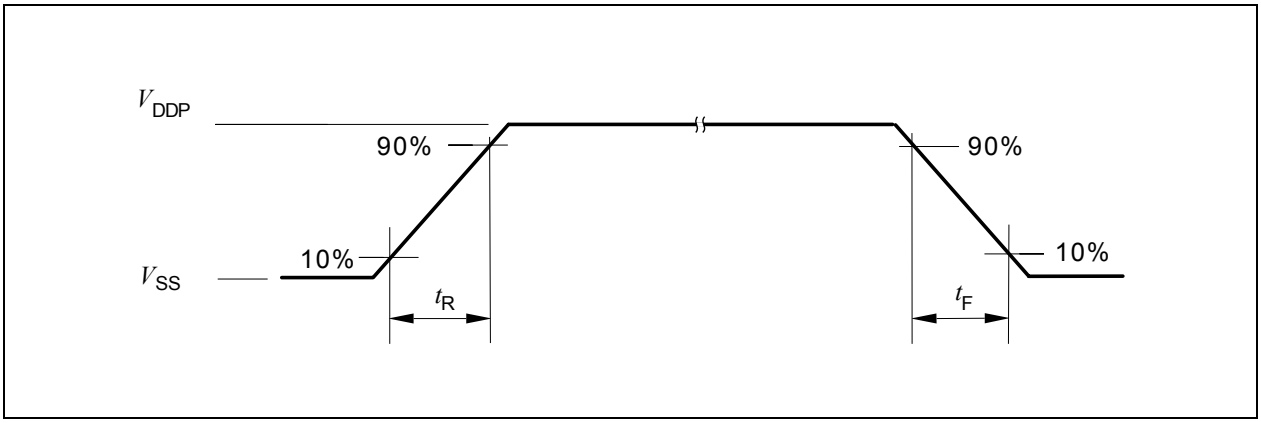



**Figure 40 Rise/Fall Time Parameters**

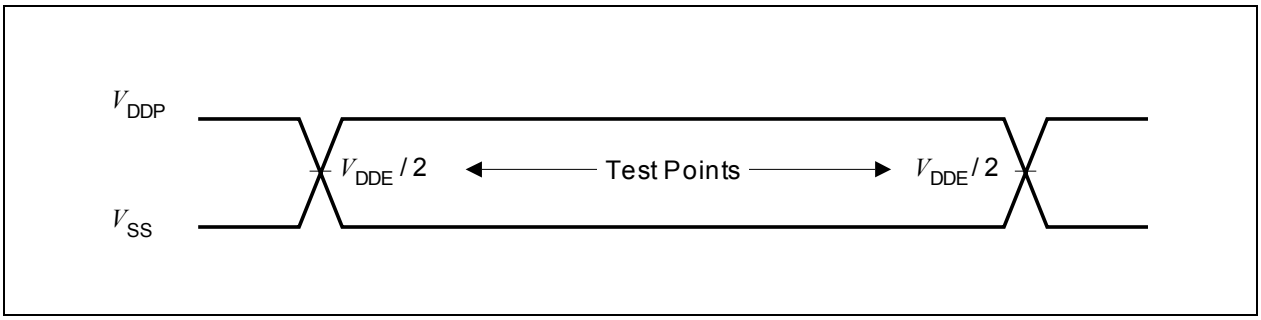



**Figure 41 Testing Waveform, Output Delay**

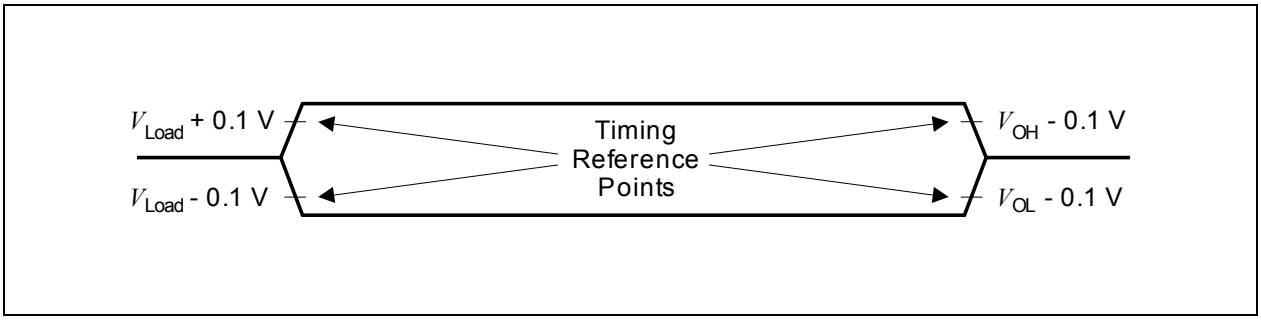



**Figure 42 Testing Waveform, Output High Impedance**

### 4.3.2 Output Rise/Fall Times

Table 45 provides the characteristics of the output rise/fall times in the XC886/888.

**Table 45 Output Rise/Fall Times Parameters (Operating Conditions apply)**

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

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

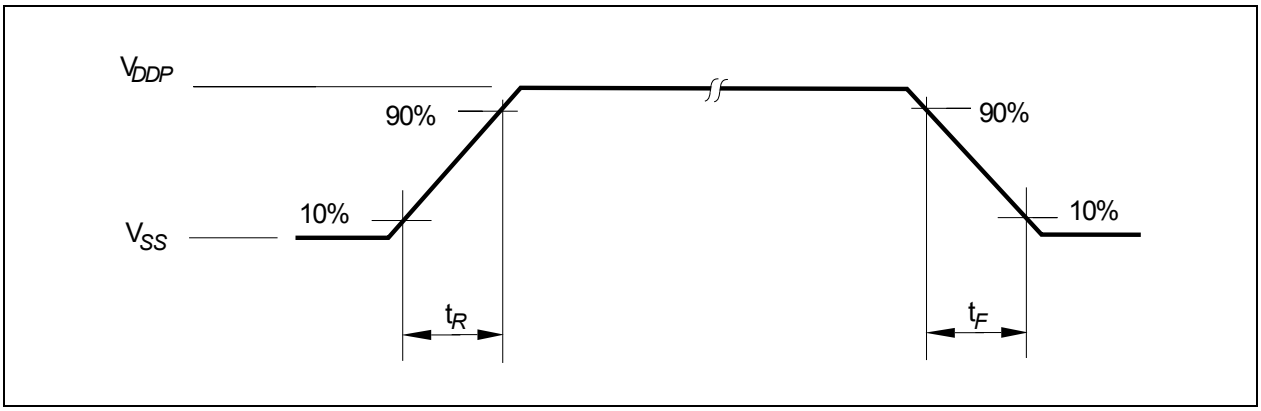



**Figure 43 Rise/Fall Times Parameters**

### 4.3.3 Power-on Reset and PLL Timing

Table 49 provides the characteristics of the power-on reset and PLL timing in the XC886/888.

**Table 46 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 | 1) |
| On-Chip Oscillator start-up time | $t_{OSCST}$ | CC | – | – | 500 | ns | 1) |
| Flash initialization time | $t_{FINIT}$ | CC | – | 160 | – | µs | 1) |
| $\overline{\text{RESET}}$ hold time | $t_{RST}$ | SR | – | 500 | – | µs | $V_{DDP}$ rise time (10% – 90%) ≤ 500µs 1)2) |
| PLL lock-in in time | $t_{LOCK}$ | CC | – | – | 200 | µs | 1) |
| PLL accumulated jitter | $D_P$ | | – | – | 0.7 | ns | 1)3) |

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

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

3) PLL lock at 96 MHz using a 4 MHz external oscillator. The PLL Divider settings are K = 2, N = 48 and P = 1.

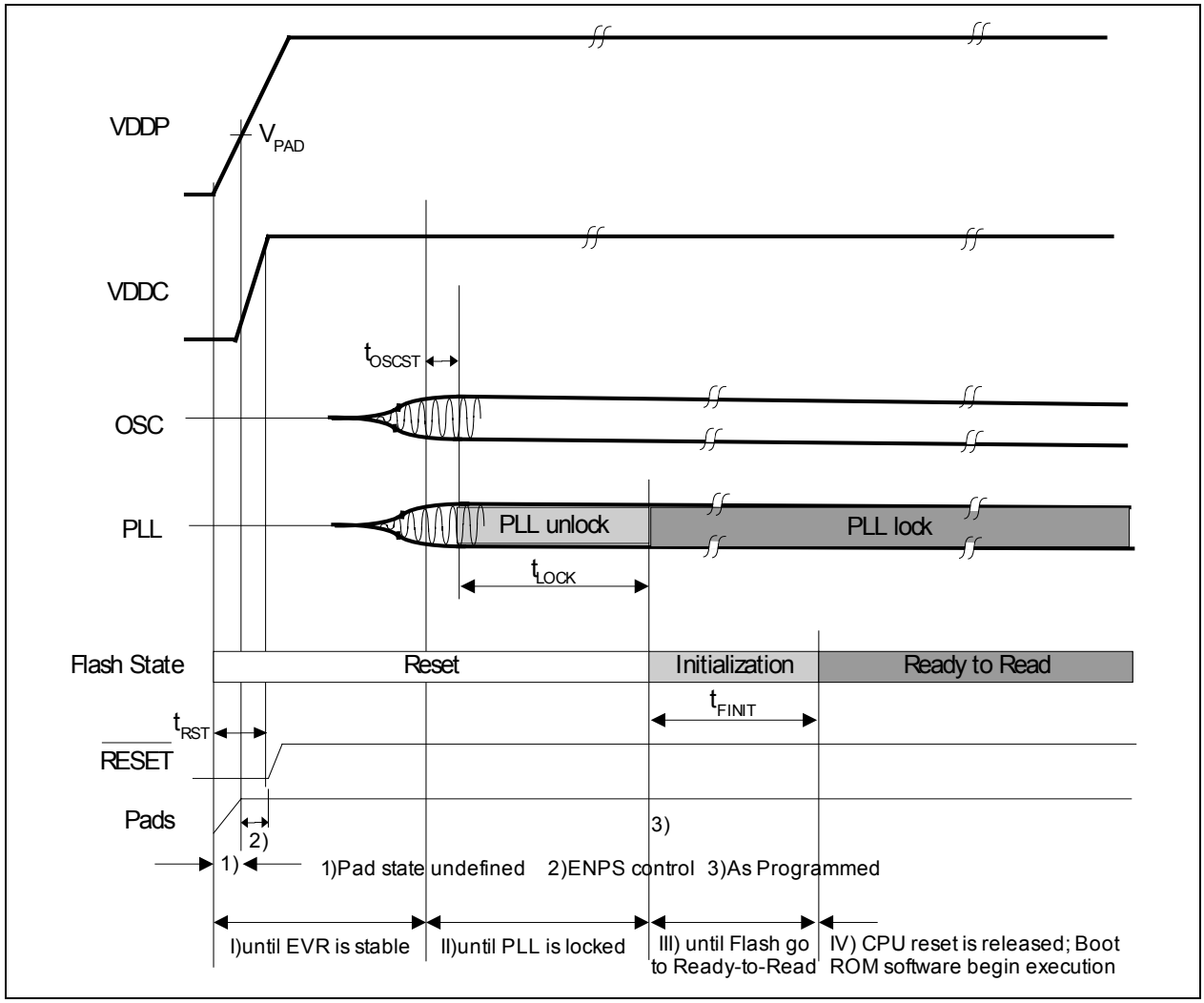



**Figure 44 Power-on Reset Timing**

### 4.3.4 On-Chip Oscillator Characteristics

Table 47 provides the characteristics of the on-chip oscillator in the XC886/888.

**Table 47 On-chip Oscillator Characteristics (Operating Conditions apply)**

| Parameter | Symbol | | Limit Values | | | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| | | | min. | typ. | max. | | |
| Nominal frequency | $f_{NOM}$ | CC | 9.36 | 9.6 | 9.84 | MHz | under nominal conditions[1)] |
| Long term frequency deviation | $\Delta f_{LT}$ | CC | -5.0 | – | 5.0 | % | with respect to $f_{NOM}$, over lifetime and temperature (-10°C to 125°C), for one given device after trimming |
| | | | -6.0 | – | 0 | % | with respect to $f_{NOM}$, over lifetime and temperature (-40°C to -10°C), for one given device after trimming |
| Short term frequency deviation | $\Delta f_{ST}$ | CC | -1.0 | – | 1.0 | % | within one LIN message (<10 ms .... 100 ms) |

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

### 4.3.5 External Clock Drive XTAL1

Table 48 shows the parameters that define the external clock supply for XC886/888. These timing parameters are based on the direct XTAL1 drive of clock input signals. They are not applicable if an external crystal or ceramic resonator is considered.

**Table 48 External Clock Drive Characteristics (Operating Conditions apply)**

| Parameter | Symbol | | Limit Values | | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| | | | Min. | Max. | | |
| Oscillator period | $t_{osc}$ | SR | 83.3 | 250 | ns | 1)2) |
| High time | $t_1$ | SR | 25 | - | ns | 2)3) |
| Low time | $t_2$ | SR | 25 | - | ns | 2)3) |
| Rise time | $t_3$ | SR | - | 20 | ns | 2)3) |
| Fall time | $t_4$ | SR | - | 20 | ns | 2)3) |

1) The clock input signals with 45-55% duty cycle are used.

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

3) The clock input signal must reach the defined levels $V_{ILX}$ and $V_{IHX}$.

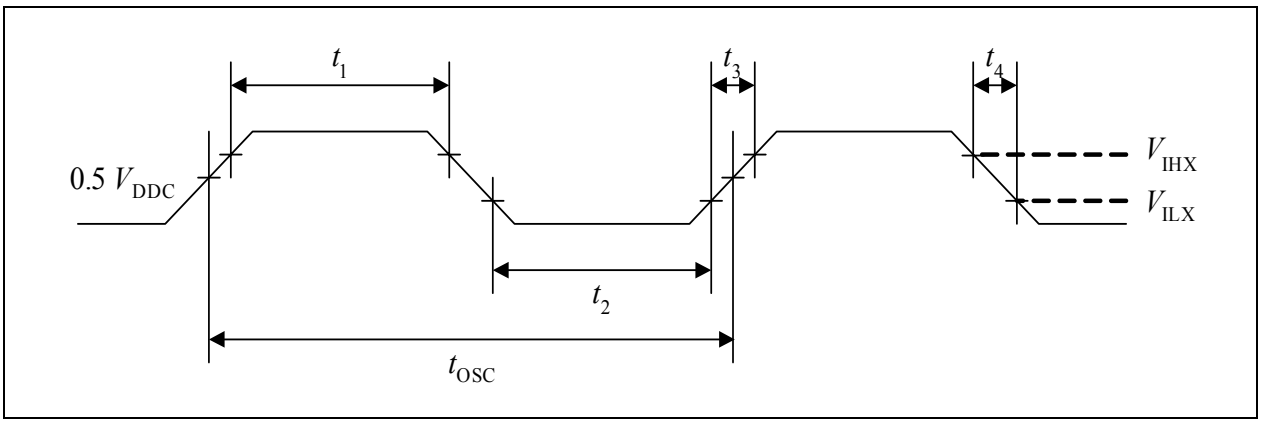



**Figure 45 External Clock Drive XTAL1**

### 4.3.6 JTAG Timing

Table 49 provides the characteristics of the JTAG timing in the XC886/888.

**Table 49 TCK Clock Timing (Operating Conditions apply; CL = 50 pF)**

| Parameter | Symbol | | Limits | | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| | | | min | max | | |
| TCK clock period | $t_{TCK}$ | SR | 50 | - | ns | 1) |
| TCK high time | $t_1$ | SR | 20 | – | ns | 1) |
| TCK low time | $t_2$ | SR | 20 | - | ns | 1) |
| TCK clock rise time | $t_3$ | SR | - | 4 | ns | 1) |
| TCK clock fall time | $t_4$ | SR | - | 4 | ns | 1) |

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

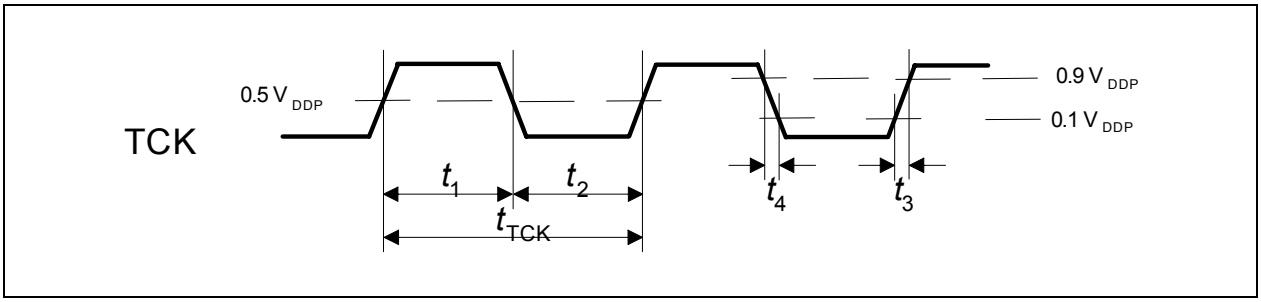



**Figure 46 TCK Clock Timing**

**Table 50 JTAG Timing (Operating Conditions apply; CL = 50 pF)**

| Parameter | Symbol | | Limits | | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| | | | min | max | | |
| TMS setup to TCK | $t_1$ | SR | 8 | - | ns | 1) |
| TMS hold to TCK | $t_2$ | SR | 24 | - | ns | 1) |
| TDI setup to TCK | $t_1$ | SR | 11 | - | ns | 1) |
| TDI hold to TCK | $t_2$ | SR | 24 | - | ns | 1) |
| TDO valid output from TCK | $t_3$ | CC | - | 21 | ns | 5V Device[1)] |
| | | | - | 28 | ns | 3.3V Device[1)] |

**Table 50 JTAG Timing (Operating Conditions apply; CL = 50 pF)** (cont'd)

| Parameter | Symbol | | Limits | | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| | | | min | max | | |
| TDO high impedance to valid output from TCK | $t_4$ | CC | - | 27 | ns | 5V Device[1] |
| | | | - | 36 | ns | 3.3V Device[1] |
| TDO valid output to high impedance from TCK | $t_5$ | CC | - | 22 | ns | 5V Device[1] |
| | | | - | 28 | ns | 3.3V Device[1] |

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

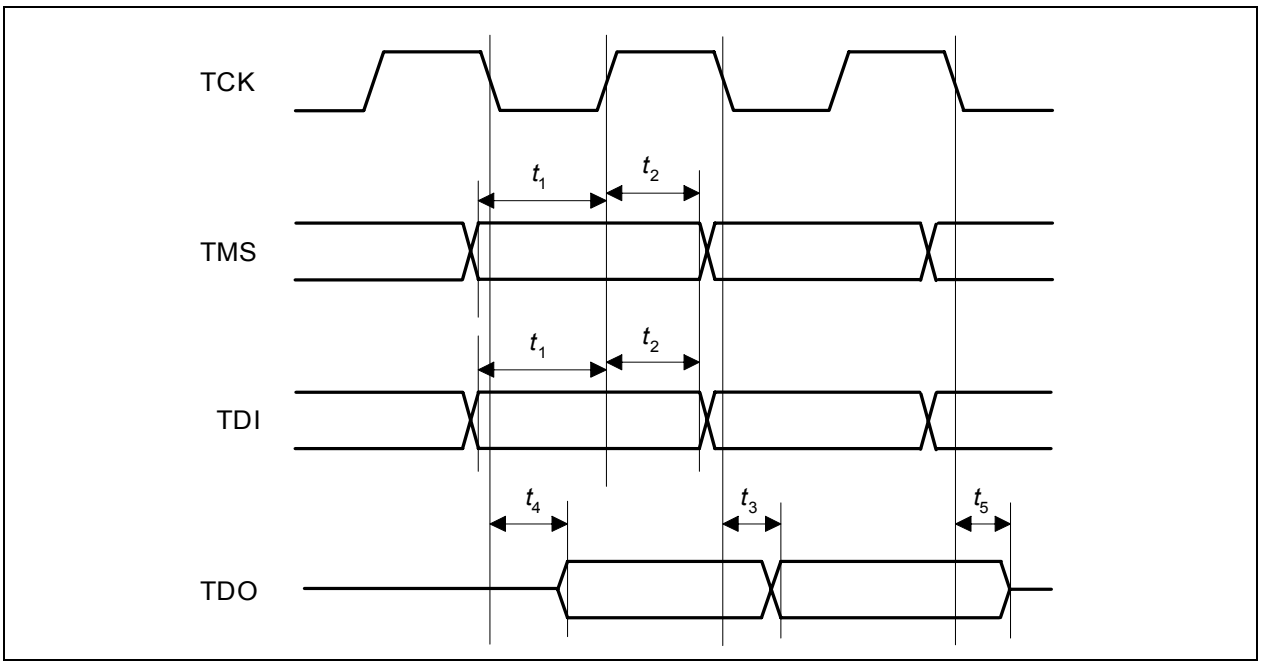



**Figure 47 JTAG Timing**

### 4.3.7 SSC Master Mode Timing

Table 51 provides the characteristics of the SSC timing in the XC886/888.

**Table 51 SSC Master Mode Timing (Operating Conditions apply; CL = 50 pF)**

| Parameter | Symbol | | Limit Values | | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| | | | min. | max. | | |
| SCLK clock period | $t_0$ | CC | $2*T_{SSC}$ | – | ns | 1)2) |
| MTSR delay from SCLK ↓ | $t_1$ | CC | 0 | 8 | ns | 2) |
| MRST setup to SCLK ↓ | $t_2$ | SR | 24 | – | ns | 2) |
| MRST hold from SCLK ↓ | $t_3$ | SR | 0 | – | ns | 2) |

1) $T_{SSCmin} = T_{CPU} = 1/f_{CPU}$. When $f_{CPU}$ = 24 MHz, $t_0$ = 83.3ns. $T_{CPU}$ is the CPU clock period.

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

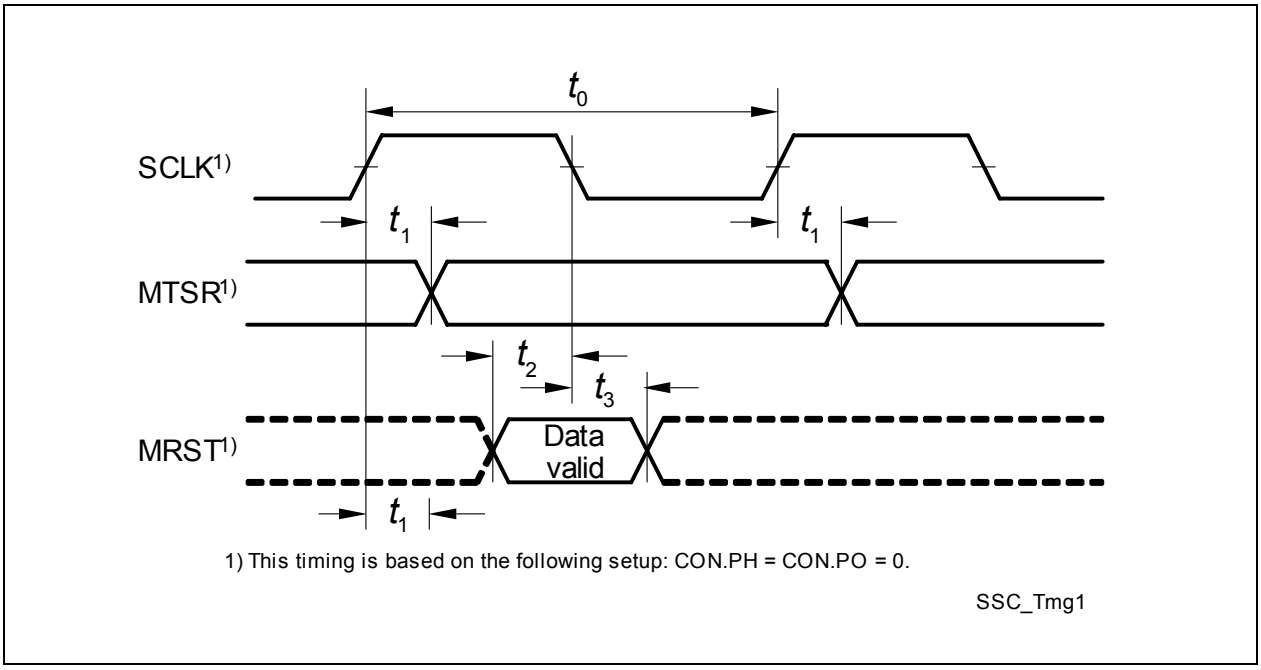



**Figure 52 SSC Master Mode Timing**

# 5 Package and Quality Declaration

**Chapter 5** provides the information of the XC886/888 package and reliability section.

## 5.1 Package Parameters

**Table 1** provides the thermal characteristics of the package used in XC886 and XC888.

**Table 1 Thermal Characteristics of the Packages**

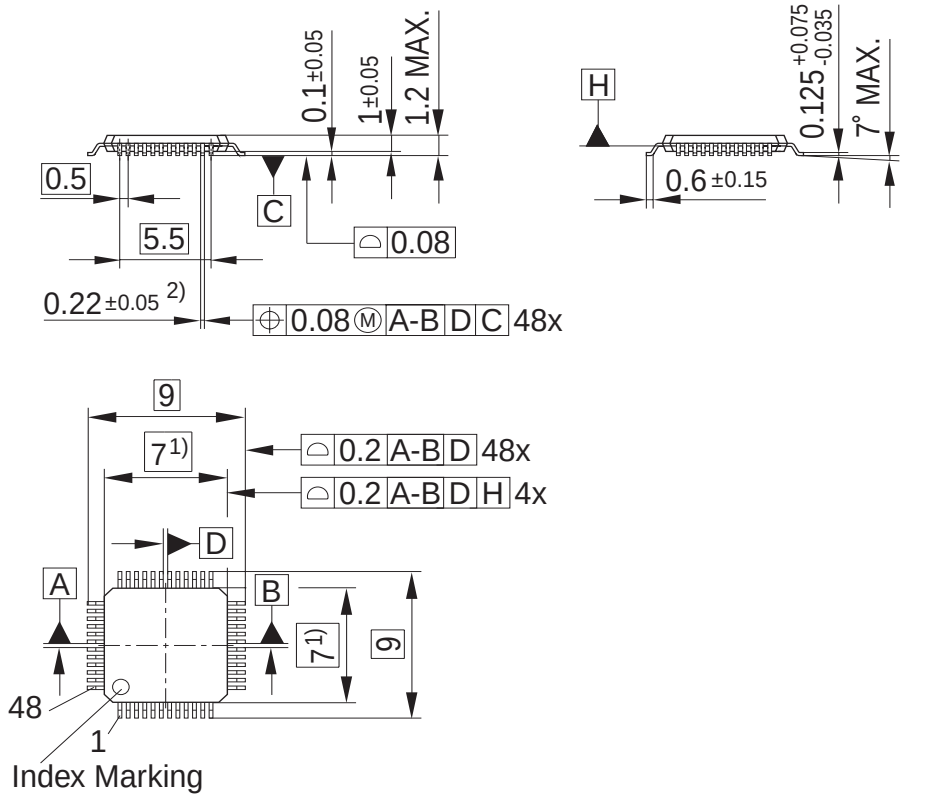
| Parameter | Symbol | | Limit Values | | Unit | Notes |
|---|---|---|---|---|---|---|
| | | | Min. | Max. | | |
| **PG-TQFP-48 (XC886)** | | | | | | |
| Thermal resistance junction case | $R_{TJC}$ | CC | - | 13 | K/W | 1)2) |
| Thermal resistance junction lead | $R_{TJL}$ | CC | - | 32.5 | K/W | 1)2) |
| **PG-TQFP-64 (XC888)** | | | | | | |
| Thermal resistance junction case | $R_{TJC}$ | CC | - | 12.6 | K/W | 1)2) |
| Thermal resistance junction lead | $R_{TJL}$ | CC | - | 33.4 | K/W | 1)2) |

1) The thermal resistances between the case and the ambient ($R_{TCA}$) , the lead and the ambient ($R_{TLA}$) are to be combined with the thermal resistances between the junction and the case ($R_{TJC}$), the junction and the lead ($R_{TJL}$) given above, 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 ($R_{TCA}$), the lead and the ambient ($R_{TLA}$) 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 thermal resistances (junction lead and lead ambient), or
b) by taking all four resistances into account, depending on the precision needed.

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

## 5.2 Package Outline

**Figure 48** shows the package outlines of the XC886.




1) Does not include plastic or metal protrusion of 0.25 max. per side
2) Does not include dambar protrusion of 0.08 max. per side

GPP09237

**Figure 48 PG-TQFP-48 Package Outline**

**Figure 49** shows the package outlines of the XC888.

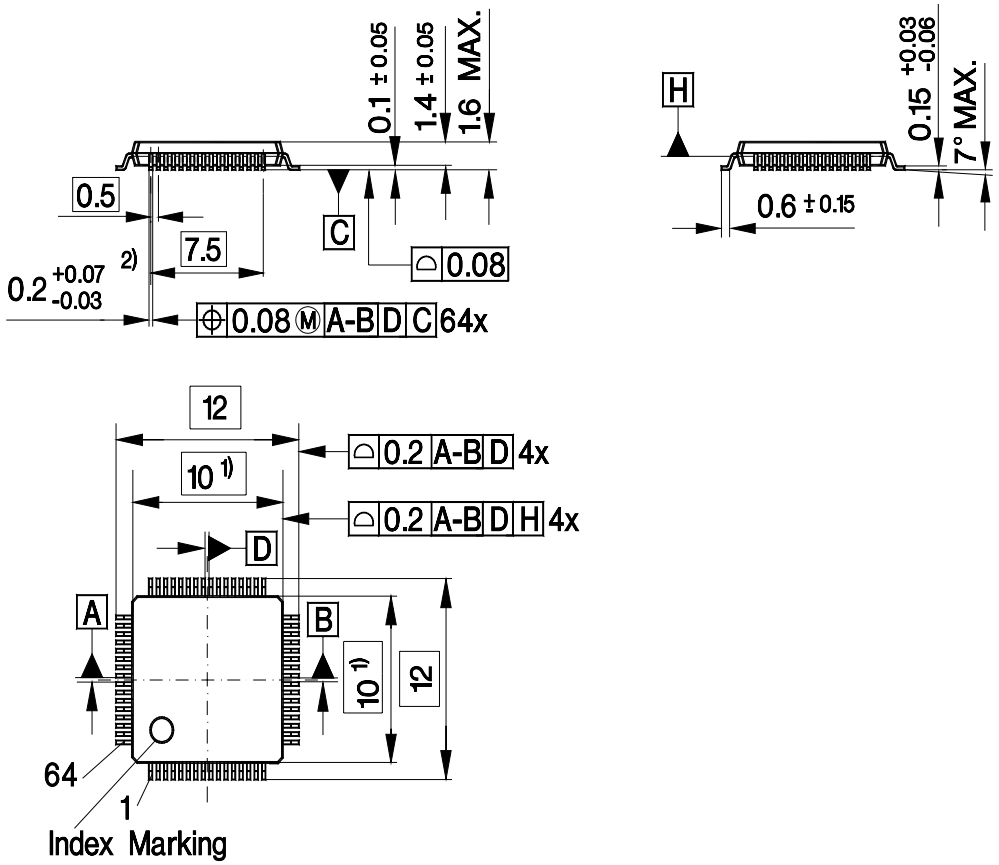



**Figure 49 PG-TQFP-64 Package Outline**

## 5.3 Quality Declaration

Table 2 shows the characteristics of the quality parameters in the XC886/888.

**Table 2 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[1)] |
| ESD susceptibility according to Charged Device Model (CDM) pins | $V_{CDM}$ | - | 500 | V | Conforming to JESD22-C101-C[1)] |

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

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

Published by Infineon Technologies AG

## Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- View SAK-XC886CM-6FFA 5V AC on WIN SOURCE
- Infineon Technologies Information

## Optimize Your Supply Chain with WIN SOURCE Solutions

- Global Sourcing Solution
- Obsolete Management
- Cost Control Management
- Shortage Management
- Alternative Solution
- Excess Inventory Management