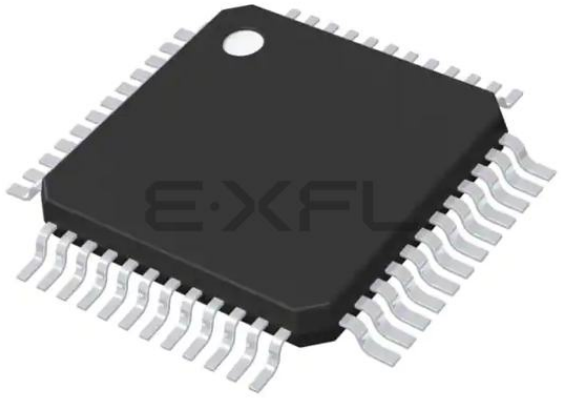
# Infineon Technologies - SAA-XC886CLM-6FFA AC Datasheet



**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

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

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | XC800 |
| Core Size | 8-Bit |
| Speed | 24MHz |
| Connectivity | CANbus, LINbus, SSI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 34 |
| Program Memory Size | 24KB (24K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 1.75K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 8x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 140°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 48-LQFP |
| Supplier Device Package | PG-TQFP-48 |
| Purchase URL | https://www.e-xfl.com/product-detail/infineon-technologies/saa-xc886clm-6ffa-ac |

## 2.2 Logic Symbol

The logic symbols of the SAA-XC886 are shown in **Figure 3**.

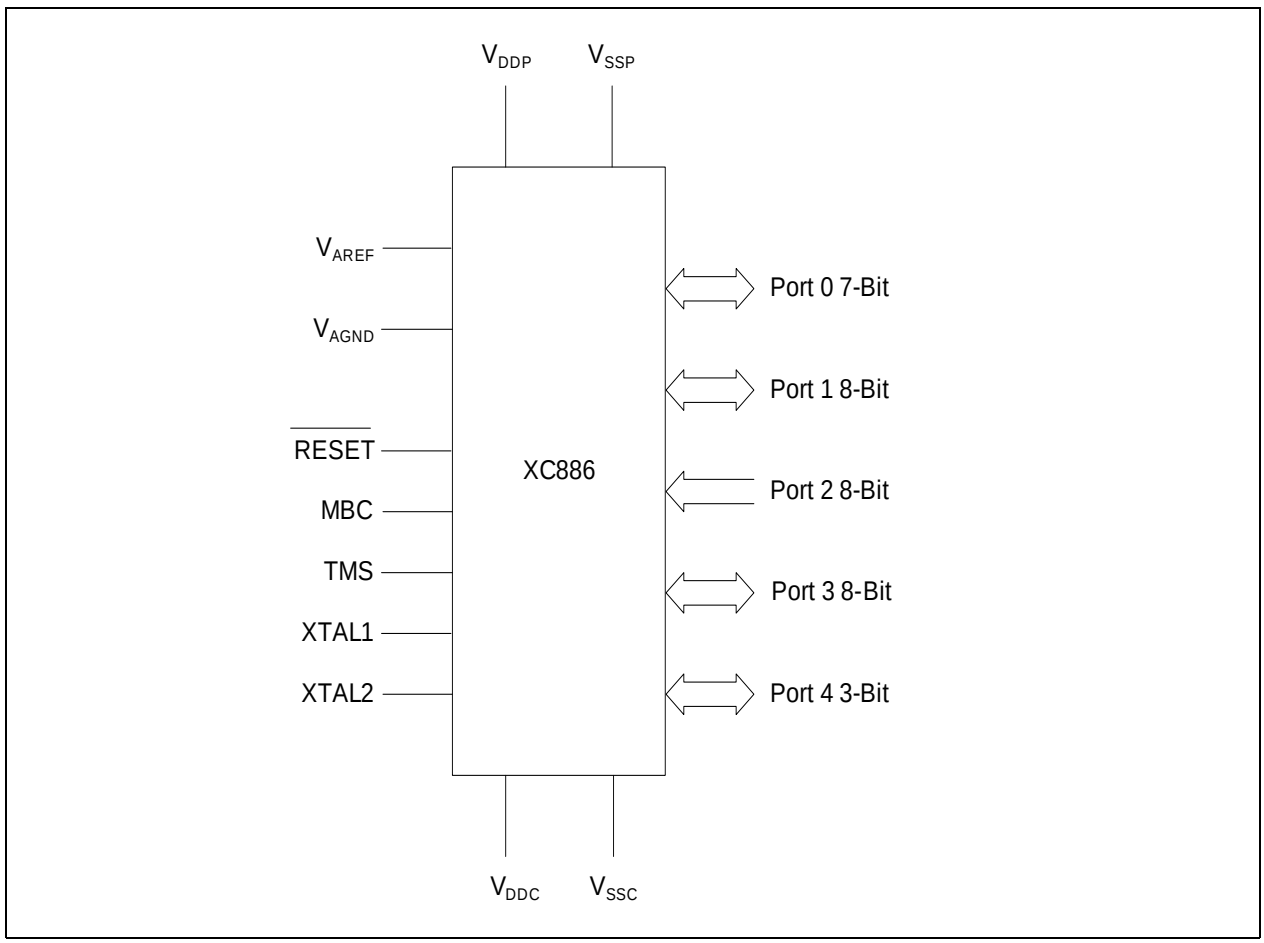



**Figure 3 SAA-XC886 Logic Symbol**

**Table 2 Pin Definitions and Functions** (cont'd)

| Symbol | Pin Number | 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 | | 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 | | 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 | | 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 | | Hi-Z | AN3 Analog Input 3 |
| P2.4 | 20 | | Hi-Z | AN4 Analog Input 4 |
| P2.5 | 21 | | Hi-Z | AN5 Analog Input 5 |
| P2.6 | 22 | | Hi-Z | AN6 Analog Input 6 |
| P2.7 | 25 | | Hi-Z | AN7 Analog Input 7 |

### 3.2.4 SAA-XC886 Register Overview

The SFRs of the SAA-XC886 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 4 CPU Register Overview**

| Addr | Register Name | Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| RMAP = 0 or 1 | | | | | | | | | | |
| $81_H$ | **SP** **Reset: $07_H$** Stack Pointer Register | Bit Field | SP | | | | | | | |
| | | Type | rw | | | | | | | |
| $82_H$ | **DPL** **Reset: $00_H$** Data Pointer Register Low | Bit Field | DPL7 | DPL6 | DPL5 | DPL4 | DPL3 | DPL2 | DPL1 | DPL0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $83_H$ | **DPH** **Reset: $00_H$** Data Pointer Register High | Bit Field | DPH7 | DPH6 | DPH5 | DPH4 | DPH3 | DPH2 | DPH1 | DPH0 |
| | | Type | rw | rw | rw | rw | rw | rw | rw | rw |
| $87_H$ | **PCON** **Reset: $00_H$** Power Control Register | Bit Field | SMOD | 0 | | | GF1 | GF0 | 0 | IDLE |
| | | Type | rw | r | | | rw | rw | r | rw |
| **$88_H$** | **TCON** **Reset: $00_H$** Timer Control Register | Bit Field | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |
| | | Type | rwh | rw | rwh | rw | rwh | rw | rwh | rw |
| $89_H$ | **TMOD** **Reset: $00_H$** Timer Mode Register | Bit Field | GATE 1 | T1S | T1M | | GATE 0 | T0S | T0M | |
| | | Type | rw | rw | rw | | rw | rw | rw | |
| $8A_H$ | **TL0** **Reset: $00_H$** Timer 0 Register Low | Bit Field | VAL | | | | | | | |
| | | Type | rwh | | | | | | | |
| $8B_H$ | **TL1** **Reset: $00_H$** Timer 1 Register Low | Bit Field | VAL | | | | | | | |
| | | Type | rwh | | | | | | | |
| $8C_H$ | **TH0** **Reset: $00_H$** Timer 0 Register High | Bit Field | VAL | | | | | | | |
| | | Type | rwh | | | | | | | |
| $8D_H$ | **TH1** **Reset: $00_H$** Timer 1 Register High | Bit Field | VAL | | | | | | | |
| | | Type | rwh | | | | | | | |
| **$98_H$** | **SCON** **Reset: $00_H$** Serial Channel Control Register | Bit Field | SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
| | | Type | rw | rw | rw | rw | rw | rwh | rwh | rwh |
| $99_H$ | **SBUF** **Reset: $00_H$** Serial Data Buffer Register | Bit Field | VAL | | | | | | | |
| | | Type | rwh | | | | | | | |

**Table 8 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 9 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 9 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 10 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 13 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 13 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 | | | | | | | |

### 3.2.4.12 SSC Registers

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

**Table 15 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 16 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 |

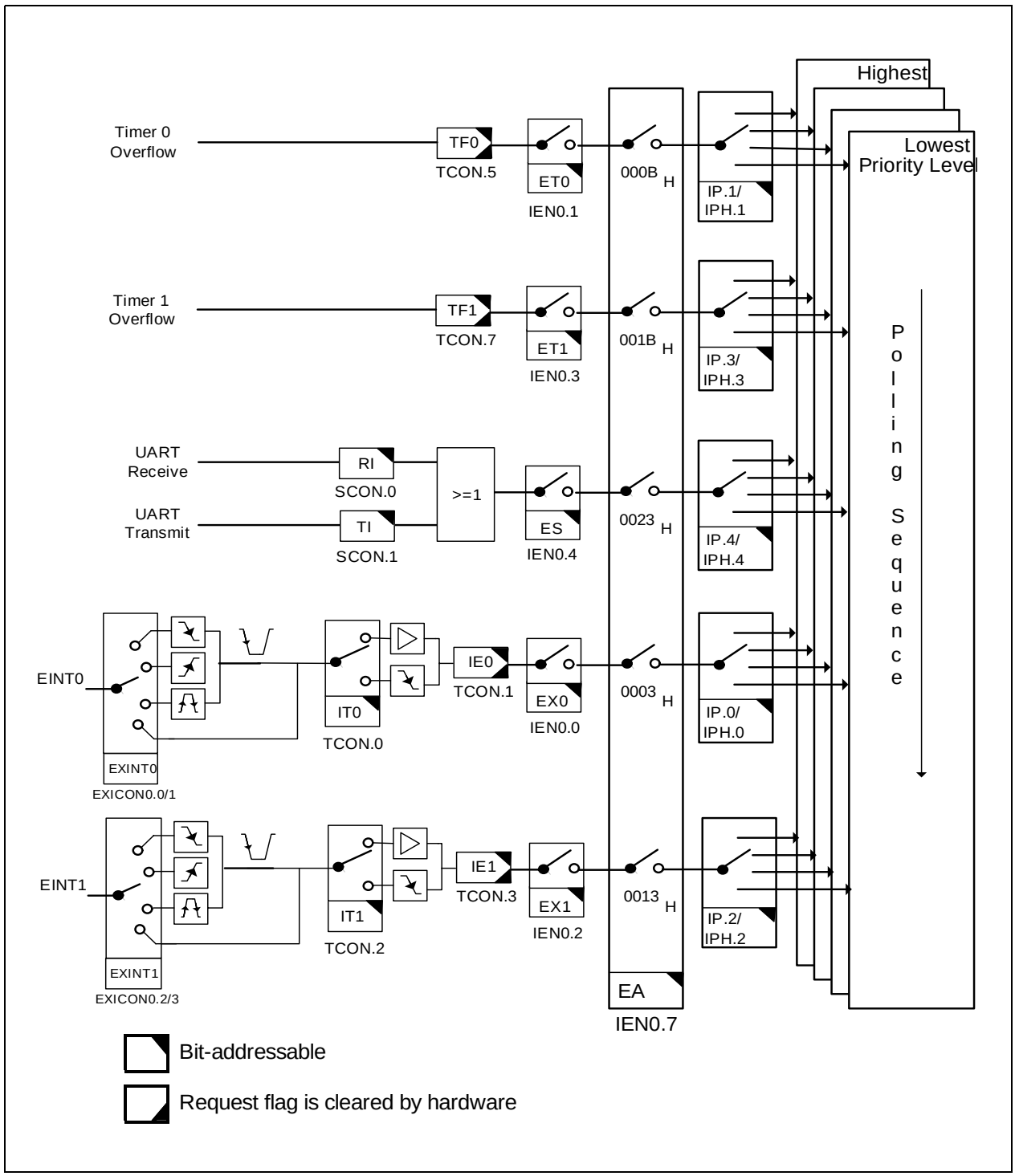



**Figure 13 Interrupt Request Sources (Part 1)**

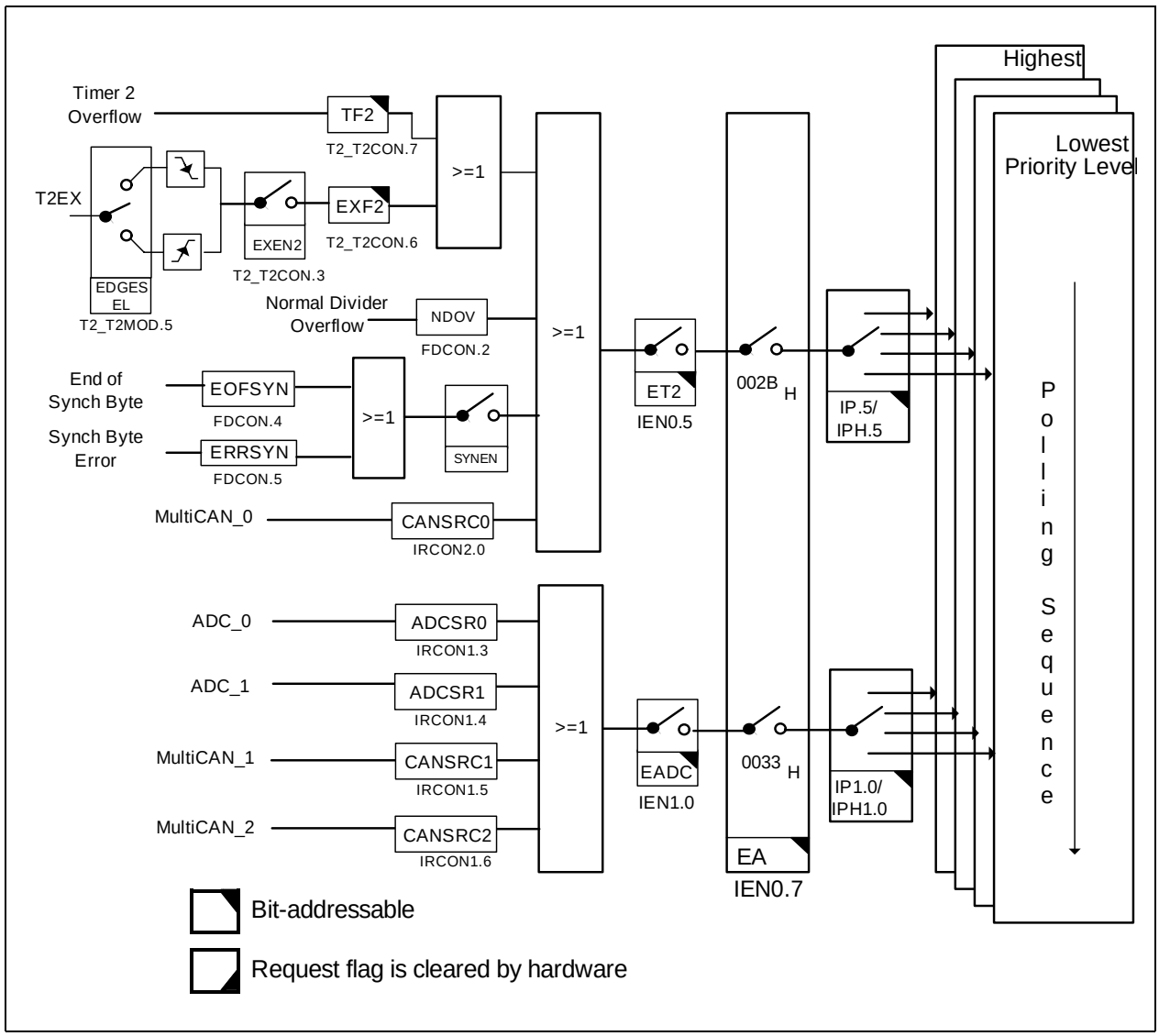



**Figure 14 Interrupt Request Sources (Part 2)**

**Figure 18** shows the structure of a bidirectional port pin.

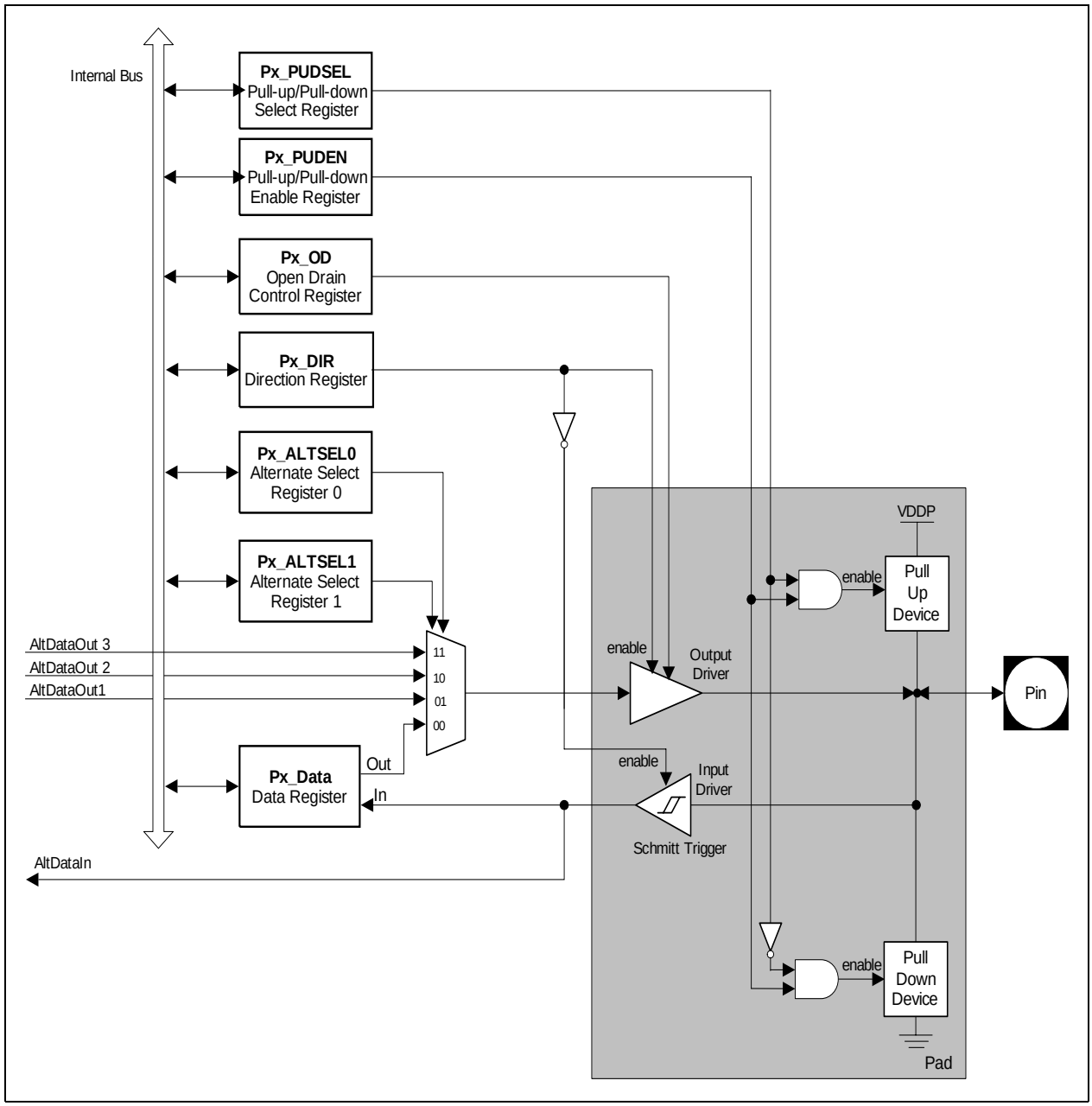



**Figure 18 General Structure of Bidirectional Port**

**Figure 19** shows the structure of an input-only port pin.

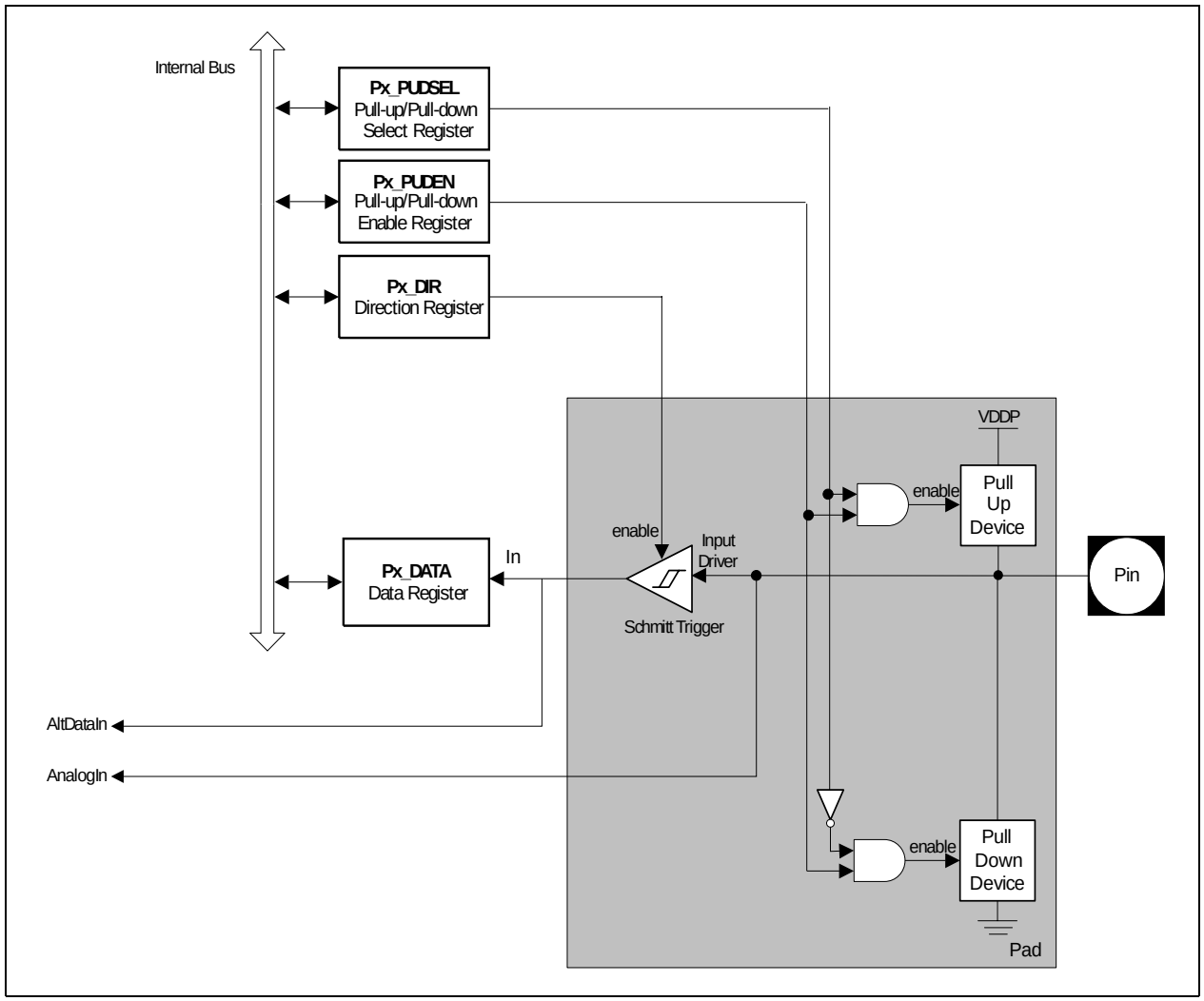



**Figure 19 General Structure of Input Port**

**Table 24** shows the VCO range for the SAA-XC886.

**Table 24 VCO Range**

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

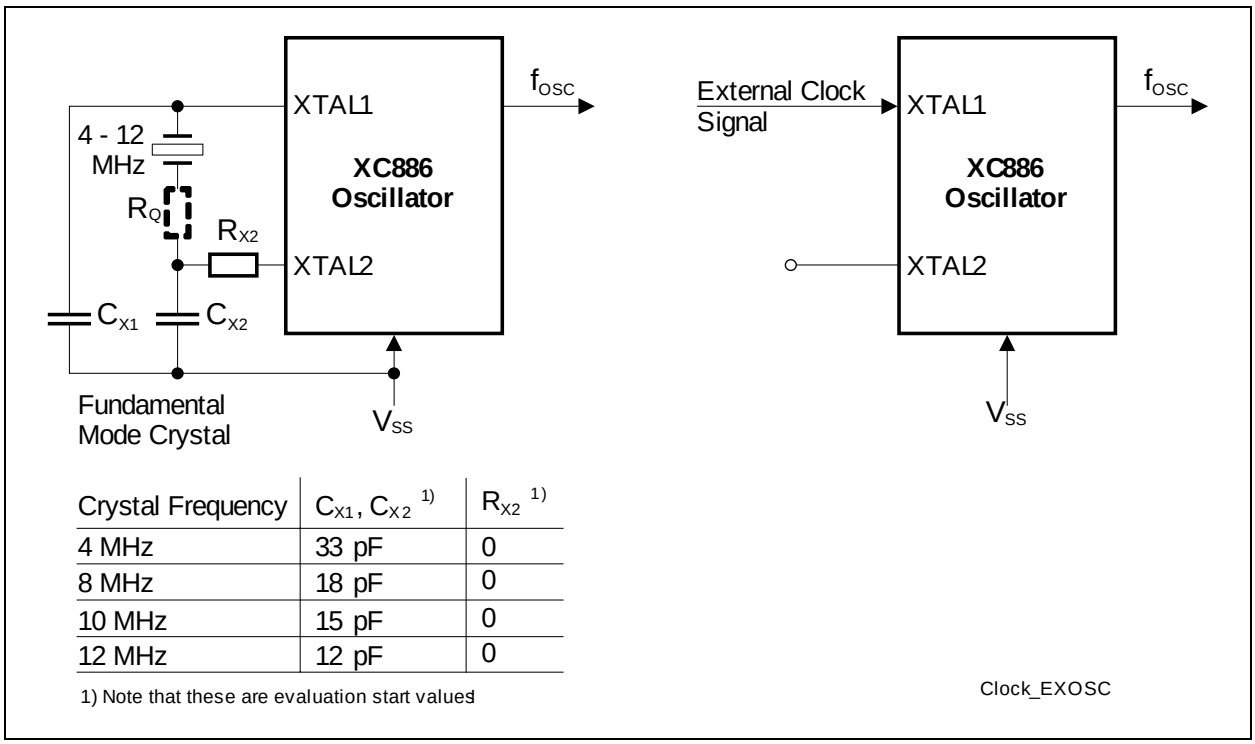
### 3.8.1 Recommended External Oscillator Circuits

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 24** 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 24** 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 24 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.*

– Interrupt enabling and corresponding flag

## 3.13 UART and UART1

The SAA-XC886 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 28**.

**Table 28 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

### 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.*

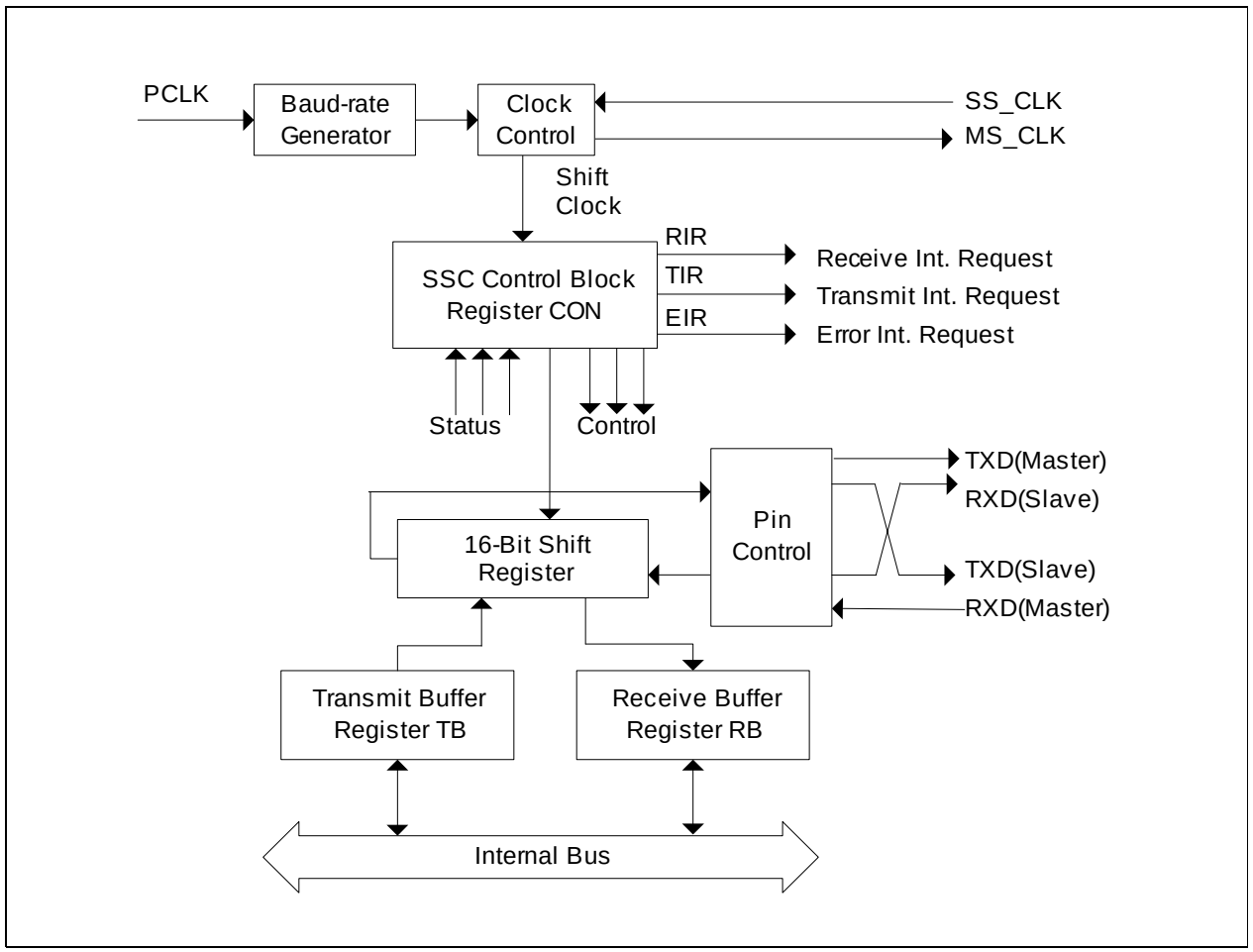



**Figure 31 SSC Block Diagram**

## 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 36**. 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 SAA-XC886 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.

**Table 48 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 | - | 35 | ns | 1) |
| TDO valid output to high impedance from TCK | $t_5$ | CC | - | 27 | ns | 1) |

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

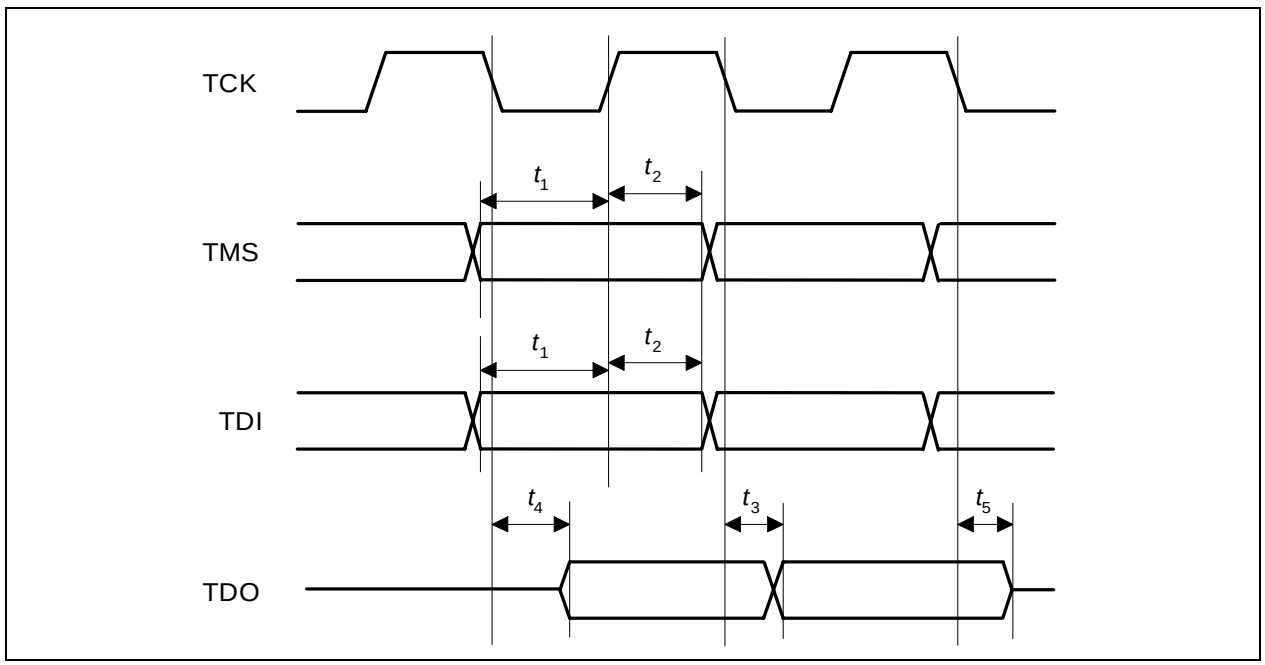



**Figure 46 JTAG Timing**

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

Published by Infineon Technologies AG