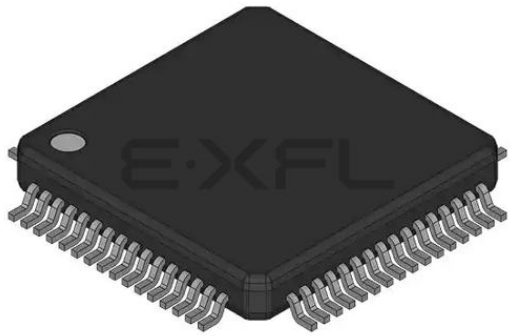

**Welcome to E-XFL.COM**

## 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 | HCS12X |
| Core Size | 16-Bit |
| Speed | 40MHz |
| Connectivity | CANbus, SCI, SPI |
| Peripherals | LVD, POR, PWM, WDT |
| Number of I/O | 44 |
| Program Memory Size | 128KB (128K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 8K x 8 |
| RAM Size | 8K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.72V ~ 5.5V |
| Data Converters | A/D 8x12b |
| Oscillator Type | External |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-LQFP |
| Supplier Device Package | 64-LQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12xs128maer |

### 1.1.4 Device Memory Map

Table 1-1 shows the device register memory map.

**Table 1-1. Device Register Memory Map**

| Address | Module | Size (Bytes) |
|---|---|---|
| 0x0000–0x0009 | PIM (port integration module) | 10 |
| 0x000A–0x000B | MMC (memory map control) | 2 |
| 0x000C–0x000D | PIM (port integration module) | 2 |
| 0x000E–0x000F | Reserved | 2 |
| 0x0010–0x0017 | MMC (memory map control) | 8 |
| 0x0018–0x0019 | Reserved | 2 |
| 0x001A–0x001B | Device ID register | 2 |
| 0x001C–0x001F | PIM (port integration module) | 4 |
| 0x0020–0x002F | DBG (debug module) | 16 |
| 0x0030–0x0031 | Reserved | 2 |
| 0x0032–0x0033 | PIM (port integration module) | 2 |
| 0x0034–0x003F | ECRG (clock and reset generator) | 12 |
| 0x0040–0x006F | TIM (timer module) | 48 |
| 0x0070–0x00C7 | Reserved | 88 |
| 0x00C8–0x00CF | SCI0 (serial communications interface) | 8 |
| 0x00D0–0x00D7 | SCI1 (serial communications interface) | 8 |
| 0x00D8–0x00DF | SPI0 (serial peripheral interface) | 8 |
| 0x00E0–0x00FF | Reserved | 32 |
| 0x0100–0x0113 | FTMR control registers | 20 |
| 0x0114–0x011F | Reserved | 12 |
| 0x0120–0x012F | INT (interrupt module) | 16 |
| 0x0130–0x013F | Reserved | 16 |
| 0x0140–0x017F | CAN0 | 64 |
| 0x0180–0x023F | Reserved | 192 |
| 0x0240–0x027F | PIM (port integration module) | 64 |
| 0x0280–0x02BF | Reserved | 64 |
| 0x02C0–0x02EF | ATD0 (analog-to-digital converter 12 bit 16-channel) | 48 |
| 0x02F0–0x02F7 | Voltage regulator | 8 |
| 0x02F8–0x02FF | Reserved | 8 |
| 0x0300–0x0327 | PWM (pulse-width modulator 8 channels) | 40 |
| 0x0328–0x033F | Reserved | 24 |
| 0x0340–0x0367 | PIT (periodic interrupt timer) | 40 |

**Table 2-3. Pin Configuration Summary**

| DDR | IO | RDR | PE | PS[1] | IE[2] | Function | Pull Device | Interrupt |
|---|---|---|---|---|---|---|---|---|
| 0 | x | x | 0 | x | 0 | Input | Disabled | Disabled |
| 0 | x | x | 1 | 0 | 0 | Input | Pull Up | Disabled |
| 0 | x | x | 1 | 1 | 0 | Input | Pull Down | Disabled |
| 0 | x | x | 0 | 0 | 1 | Input | Disabled | Falling edge |
| 0 | x | x | 0 | 1 | 1 | Input | Disabled | Rising edge |
| 0 | x | x | 1 | 0 | 1 | Input | Pull Up | Falling edge |
| 0 | x | x | 1 | 1 | 1 | Input | Pull Down | Rising edge |
| 1 | 0 | 0 | x | x | 0 | Output, full drive to 0 | Disabled | Disabled |
| 1 | 1 | 0 | x | x | 0 | Output, full drive to 1 | Disabled | Disabled |
| 1 | 0 | 1 | x | x | 0 | Output, reduced drive to 0 | Disabled | Disabled |
| 1 | 1 | 1 | x | x | 0 | Output, reduced drive to 1 | Disabled | Disabled |
| 1 | 0 | 0 | x | 0 | 1 | Output, full drive to 0 | Disabled | Falling edge |
| 1 | 1 | 0 | x | 1 | 1 | Output, full drive to 1 | Disabled | Rising edge |
| 1 | 0 | 1 | x | 0 | 1 | Output, reduced drive to 0 | Disabled | Falling edge |
| 1 | 1 | 1 | x | 1 | 1 | Output, reduced drive to 1 | Disabled | Rising edge |

1 Always “0” on Port A, B, E, K, and AD.

2 Applicable only on Port P, H, and J.

**NOTE**

All register bits in this module are completely synchronous to internal clocks during a register read.

**NOTE**

Figures of port data registers also display the alternative functions if applicable on the related pin as defined in Table 2-1. Names in brackets denote the availability of the function when using a specific routing option.

**NOTE**

Figures of module routing registers also display the module instance or module channel associated with the related routing bit.

**Table 2-21. PPST Register Field Descriptions**

| Field | Description |
|---|---|
| 7-0<br>PPST | **Port T pull device select**—Configure pull device polarity on input pin<br>This bit selects a pull-up or a pull-down device if enabled on the associated port input pin.<br><br>1 A pull-down device selected<br>0 A pull-up device selected |

## 2.3.24 PIM Reserved Register

Address 0x0246 Access: User read[1]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 2-22. PIM Reserved Register**

1 Read: Always reads 0x00
Write: Unimplemented

## 2.3.25 Port T Routing Register (PTTRR)

Address 0x0247 Access: User read[1]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | PTTRR7 | PTTRR6 | PTTRR5 | PTTRR4 | 0 | PTTRR2 | PTTRR1 | PTTRR0 |
| W | | | | | | | | |
| Routing Option | PWM7 | PWM6 | PWM5 | PWM4 | — | IOC2 | IOC1 | IOC0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 2-23. Port T Routing Register (PTTRR)**

1 Read: Anytime
Write: Anytime

This register configures the re-routing of PWM and TIM channels on alternative pins.

## 2.3.69 Port AD0 Data Direction Register 1 (DDR1AD0)

Address 0x0273 Access: User read/write[1]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | DDR1AD07 | DDR1AD06 | DDR1AD05 | DDR1AD04 | DDR1AD03 | DDR1AD02 | DDR1AD01 | DDR1AD00 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 2-67. Port AD0 Data Direction Register 1 (DDR1AD0)**

1 Read: Anytime
Write: Anytime

**Table 2-66. DDR1AD0 Register Field Descriptions**

| Field | Description |
|---|---|
| 7-0<br>DDR1AD0 | **Port AD0 data direction**—<br>This bit determines whether the associated pin is an input or output.<br>To use the digital input function the ATD Digital Input Enable Register (ATD0DIEN) has to be set to logic level “1”.<br><br>1 Associated pin configured as output<br>0 Associated pin configured as input |

## 2.3.70 Port AD0 Reduced Drive Register 0 (RDR0AD0)

Address 0x0274 Access: User read/write[1]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | RDR0AD07 | RDR0AD06 | RDR0AD05 | RDR0AD04 | RDR0AD03 | RDR0AD02 | RDR0AD01 | RDR0AD00 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 2-68. Port AD0 Reduced Drive Register 0 (RDR0AD0)**

1 Read: Anytime
Write: Anytime

**Table 2-67. RDR0AD0 Register Field Descriptions**

| Field | Description |
|---|---|
| 7-0<br>RDR0AD0 | **Port AD0 reduced drive**—Select reduced drive for output pin<br>This bit configures the drive strength of the associated output pin as either full or reduced. If a pin is used as input this bit has no effect. The reduced drive function is independent of which function is being used on a particular pin.<br><br>1 Reduced drive selected (approx. 1/5 of the full drive strength)<br>0 Full drive strength enabled |

## 3.4 Functional Description

The MMC block performs several basic functions of the S12X sub-system operation: MCU operation modes, priority control, address mapping, select signal generation and access limitations for the system. Each aspect is described in the following subsections.

### 3.4.1 MCU Operating Mode

- Normal single-chip mode

  There is no external bus in this mode. The MCU program is executed from the internal memory and no external accesses are allowed.
- Special single-chip mode

  This mode is generally used for debugging single-chip operation, boot-strapping or security related operations. The active background debug mode is in control of the CPU code execution and the BDM firmware is waiting for serial commands sent through the BKGD pin. There is no external bus in this mode.

### 3.4.2 Memory Map Scheme

#### 3.4.2.1 CPU and BDM Memory Map Scheme

The BDM firmware lookup tables and BDM register memory locations share addresses with other modules; however they are not visible in the global memory map during user's code execution. The BDM memory resources are enabled only during the READ_BD and WRITE_BD access cycles to distinguish between accesses to the BDM memory area and accesses to the other modules. (Refer to BDM Block Guide for further details).

When the MCU enters active BDM mode, the BDM firmware lookup tables and the BDM registers become visible in the local memory map in the range 0xFF00-0xFFFF (global address 0x7F_FF00 - 0x7F_FFFF) and the CPU begins execution of firmware commands or the BDM begins execution of hardware commands. The resources which share memory space with the BDM module will not be visible in the global memory map during active BDM mode.

Please note that after the MCU enters active BDM mode the BDM firmware lookup tables and the BDM registers will also be visible between addresses 0xBF00 and 0xBFFF if the PPAGE register contains value of 0xFF.

within a few percent of the actual target speed and the communication protocol can easily tolerate speed errors of several percent.

As soon as the SYNC request is detected by the target, any partially received command or bit retrieved is discarded. This is referred to as a soft-reset, equivalent to a time-out in the serial communication. After the SYNC response, the target will consider the next negative edge (issued by the host) as the start of a new BDM command or the start of new SYNC request.

Another use of the SYNC command pulse is to abort a pending ACK pulse. The behavior is exactly the same as in a regular SYNC command. Note that one of the possible causes for a command to not be acknowledged by the target is a host-target synchronization problem. In this case, the command may not have been understood by the target and so an ACK response pulse will not be issued.

### 5.4.10 Instruction Tracing

When a TRACE1 command is issued to the BDM in active BDM, the CPU exits the standard BDM firmware and executes a single instruction in the user code. Once this has occurred, the CPU is forced to return to the standard BDM firmware and the BDM is active and ready to receive a new command. If the TRACE1 command is issued again, the next user instruction will be executed. This facilitates stepping or tracing through the user code one instruction at a time.

If an interrupt is pending when a TRACE1 command is issued, the interrupt stacking operation occurs but no user instruction is executed. Once back in standard BDM firmware execution, the program counter points to the first instruction in the interrupt service routine.

Be aware when tracing through the user code that the execution of the user code is done step by step but all peripherals are free running. Hence possible timing relations between CPU code execution and occurrence of events of other peripherals no longer exist.

Do not trace the CPU instruction BGND used for soft breakpoints. Tracing the BGND instruction will result in a return address pointing to BDM firmware address space.

When tracing through user code which contains stop or wait instructions the following will happen when the stop or wait instruction is traced:

The CPU enters stop or wait mode and the TRACE1 command can not be finished before leaving the low power mode. This is the case because BDM active mode can not be entered after CPU executed the stop instruction. However all BDM hardware commands except the BACKGROUND command are operational after tracing a stop or wait instruction and still being in stop or wait mode. If system stop mode is entered (all bus masters are in stop mode) no BDM command is operational.

As soon as stop or wait mode is exited the CPU enters BDM active mode and the saved PC value points to the entry of the corresponding interrupt service routine.

In case the handshake feature is enabled the corresponding ACK pulse of the TRACE1 command will be discarded when tracing a stop or wait instruction. Hence there is no ACK pulse when BDM active mode is entered as part of the TRACE1 command after CPU exited from stop or wait mode. All valid commands sent during CPU being in stop or wait mode or after CPU exited from stop or wait mode will have an ACK pulse. The handshake feature becomes disabled only when system stop mode has been reached. Hence

### 6.3.2.6 Debug Count Register (DBGCNT)

Address: 0x0026

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | CNT | | | | | | |
| W | | | | | | | | |
| Reset | 0 | — | — | — | — | — | — | — |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 6-8. Debug Count Register (DBGCNT)**

Read: Anytime

Write: Never

**Table 6-16. DBGCNT Field Descriptions**

| Field | Description |
|---|---|
| 6–0<br>CNT[6:0] | **Count Value** — The CNT bits [6:0] indicate the number of valid data 64-bit data lines stored in the Trace Buffer. Table 6-17 shows the correlation between the CNT bits and the number of valid data lines in the Trace Buffer. When the CNT rolls over to zero, the TBF bit in DBGSR is set and incrementing of CNT will continue in end-trigger or mid-trigger mode. The DBGCNT register is cleared when ARM in DBGC1 is written to a one. The DBGCNT register is cleared by power-on-reset initialization but is not cleared by other system resets. Thus should a reset occur during a debug session, the DBGCNT register still indicates after the reset, the number of valid trace buffer entries stored before the reset occurred. The DBGCNT register is not decremented when reading from the trace buffer. |

**Table 6-17. CNT Decoding Table**

| TBF (DBGSR) | CNT[6:0] | Description |
|---|---|---|
| 0 | 0000000 | No data valid |
| 0 | 0000001 | 32 bits of one line valid |
| 0 | 0000010<br>0000100<br>0000110<br>..<br>1111100 | 1 line valid<br>2 lines valid<br>3 lines valid<br>..<br>62 lines valid |
| 0 | 1111110 | 63 lines valid |
| 1 | 0000000 | 64 lines valid; if using Begin trigger alignment,<br>ARM bit will be cleared and the tracing session ends. |
| 1 | 0000010<br>..<br>..<br>1111110 | 64 lines valid,<br>oldest data has been overwritten by most recent data |

### 6.3.2.7 Debug State Control Registers

There is a dedicated control register for each of the state sequencer states 1 to 3 that determines if transitions from that state are allowed, depending upon comparator matches or tag hits, and defines the

### 6.4.5.3 Trace Buffer Organization

Referring to Table 6-40. ADRH, ADRM, ADRL denote address high, middle and low byte respectively. INF bytes contain control information (R/W, S/D etc.). The numerical suffix indicates which tracing step. The information format for Loop1 Mode and PurePC Mode is the same as that of Normal Mode. Whilst tracing in Normal or Loop1 modes each array line contains 2 data entries, thus in this case the DBGCNT[0] is incremented after each separate entry. In Detail mode DBGCNT[0] remains cleared whilst the other DBGCNT bits are incremented on each trace buffer entry.

When a COF occurs a trace buffer entry is made and the corresponding CDV bit is set.

Single byte data accesses in Detail Mode are always stored to the low byte of the trace buffer (CDATAL ) and the high byte is cleared. When tracing word accesses, the byte at the lower address is always stored to trace buffer byte3 and the byte at the higher address is stored to byte2.

**Table 6-40. Trace Buffer Organization**

| Mode | 8-Byte Wide Word Buffer | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| S12XCPU Detail | CXINF1 | CADRH1 | CADRM1 | CADRL1 | CDATAH1 | CDATAL1 | | |
| | CXINF2 | CADRH2 | CADRM2 | CADRL2 | CDATAH2 | CDATAL2 | | |
| CPU12X Other Modes | CINF1 | CPCH1 | CPCM1 | CPCL1 | CINF0 | CPCH0 | CPCM0 | CPCL0 |
| | CINF3 | CPCH3 | CPCM3 | CPCL3 | CINF2 | CPCH2 | CPCM2 | CPCL2 |

## 8.3 Memory Map and Registers

This section provides a detailed description of all registers accessible in the S12XECRG.

### 8.3.1 Module Memory Map

Figure 8-2 gives an overview on all S12XECRG registers.

| Address | Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0000 | SYNR | R | VCOFRQ[1:0] | | SYNDIV[5:0] | | | | | |
| | | W | | | | | | | | |
| 0x0001 | REFDV | R | REFFRQ[1:0] | | REFDIV[5:0] | | | | | |
| | | W | | | | | | | | |
| 0x0002 | POSTDIV | R | 0 | 0 | 0 | POSTDIV[4:0] | | | | |
| | | W | | | | | | | | |
| 0x0003 | CRGFLG | R | RTIF | PORF | LVRF | LOCKIF | LOCK | ILAF | SCMIF | SCM |
| | | W | | | | | | | | |
| 0x0004 | CRGINT | R | RTIE | 0 | 0 | LOCKIE | 0 | 0 | SCMIE | 0 |
| | | W | | | | | | | | |
| 0x0005 | CLKSEL | R | PLLSEL | PSTP | XCLKS | 0 | PLLWAI | 0 | RTIWAI | COPWAI |
| | | W | | | | | | | | |
| 0x0006 | PLLCTL | R | CME | PLLON | FM1 | FM0 | FSTWKP | PRE | PCE | SCME |
| | | W | | | | | | | | |
| 0x0007 | RTICTL | R | RTDEC | RTR6 | RTR5 | RTR4 | RTR3 | RTR2 | RTR1 | RTR0 |
| | | W | | | | | | | | |
| 0x0008 | COPCTL | R | WCOP | RSBCK | 0 | 0 | 0 | CR2 | CR1 | CR0 |
| | | W | | | WRTMASK | | | | | |
| 0x0009 | FORBYP[2] | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x000A | CTCTL[2] | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x000B | ARMCOP | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |

2. FORBYP and CTCTL are intended for factory test purposes only.

= Unimplemented or Reserved

**Figure 8-2. CRG Register Summary**

**NOTE**

Register Address = Base Address + Address Offset, where the Base Address is defined at the MCU level and the Address Offset is defined at the module level.

## 10.1.3 Block Diagram

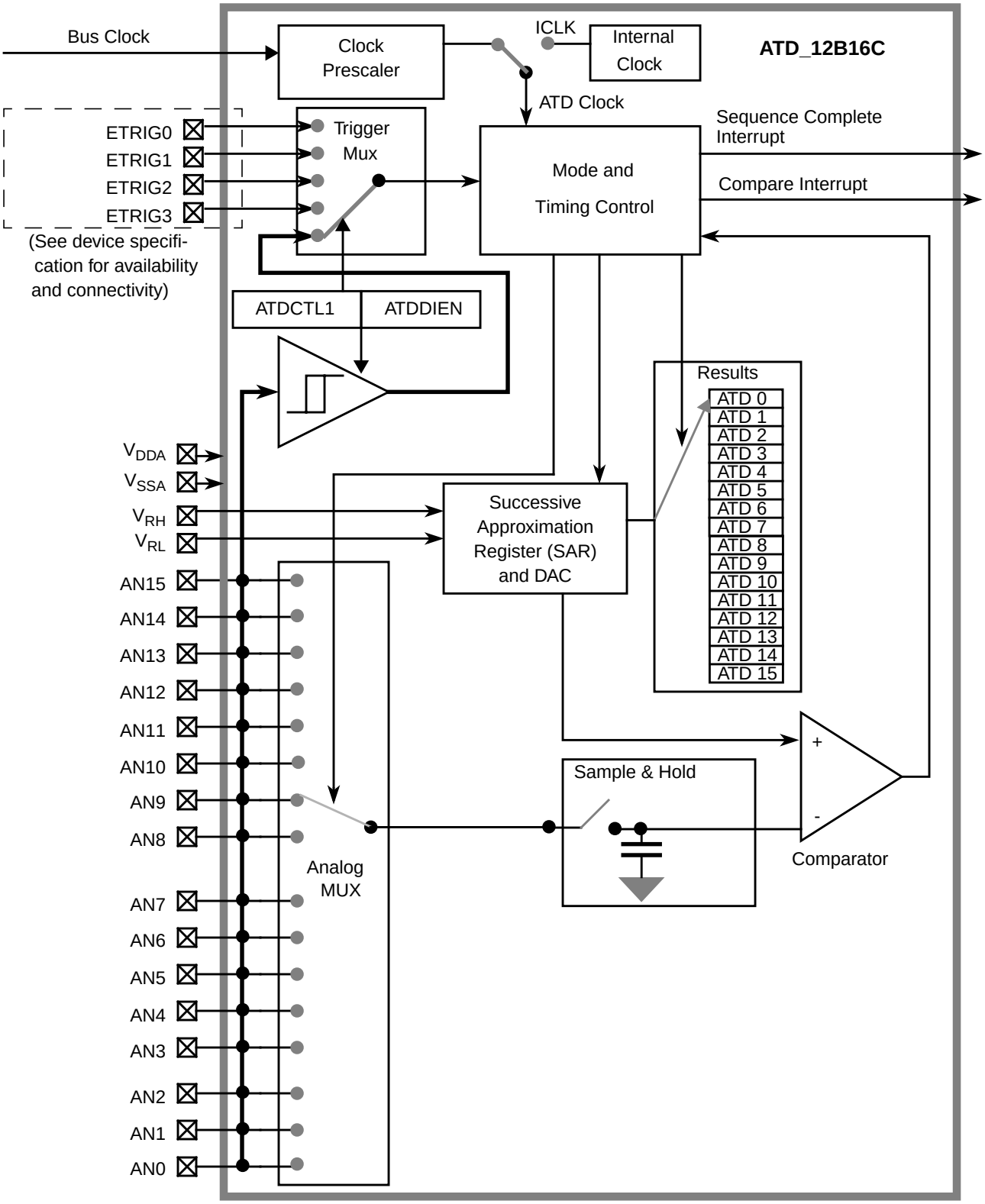


**Figure 10-1. ADC12B16C Block Diagram**

Module Base + 0x00X2

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | ID14 | ID13 | ID12 | ID11 | ID10 | ID9 | ID8 | ID7 |
| Reset: | x | x | x | x | x | x | x | x |

**Figure 11-28. Identifier Register 2 (IDR2) — Extended Identifier Mapping**

**Table 11-29. IDR2 Register Field Descriptions — Extended**

| Field | Description |
|---|---|
| 7-0<br>ID[14:7] | **Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number. |

Module Base + 0x00X3

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | ID6 | ID5 | ID4 | ID3 | ID2 | ID1 | ID0 | RTR |
| Reset: | x | x | x | x | x | x | x | x |

**Figure 11-29. Identifier Register 3 (IDR3) — Extended Identifier Mapping**

**Table 11-30. IDR3 Register Field Descriptions — Extended**

| Field | Description |
|---|---|
| 7-1<br>ID[6:0] | **Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number. |
| 0<br>RTR | **Remote Transmission Request** — This flag reflects the status of the remote transmission request bit in the CAN frame. In the case of a receive buffer, it indicates the status of the received frame and supports the transmission of an answering frame in software. In the case of a transmit buffer, this flag defines the setting of the RTR bit to be sent.<br>0 Data frame<br>1 Remote frame |

**Table 11-36. Time Segment Syntax**

| Syntax | Description |
|---|---|
| SYNC_SEG | System expects transitions to occur on the CAN bus during this period. |
| Transmit Point | A node in transmit mode transfers a new value to the CAN bus at this point. |
| Sample Point | A node in receive mode samples the CAN bus at this point. If the three samples per bit option is selected, then this point marks the position of the third sample. |

The synchronization jump width (see the Bosch CAN 2.0A/B specification for details) can be programmed in a range of 1 to 4 time quanta by setting the SJW parameter.

The SYNC_SEG, TSEG1, TSEG2, and SJW parameters are set by programming the MSCAN bus timing registers (CANBTR0, CANBTR1) (see Section 11.3.2.3, "MSCAN Bus Timing Register 0 (CANBTR0)" and Section 11.3.2.4, "MSCAN Bus Timing Register 1 (CANBTR1)").

Table 11-37 gives an overview of the Bosch CAN 2.0A/B specification compliant segment settings and the related parameter values.

**NOTE**

It is the user's responsibility to ensure the bit time settings are in compliance with the CAN standard.

**Table 11-37. Bosch CAN 2.0A/B Compliant Bit Time Segment Settings**

| Time Segment 1 | TSEG1 | Time Segment 2 | TSEG2 | Synchronization Jump Width | SJW |
|---|---|---|---|---|---|
| 5 .. 10 | 4 .. 9 | 2 | 1 | 1 .. 2 | 0 .. 1 |
| 4 .. 11 | 3 .. 10 | 3 | 2 | 1 .. 3 | 0 .. 2 |
| 5 .. 12 | 4 .. 11 | 4 | 3 | 1 .. 4 | 0 .. 3 |
| 6 .. 13 | 5 .. 12 | 5 | 4 | 1 .. 4 | 0 .. 3 |
| 7 .. 14 | 6 .. 13 | 6 | 5 | 1 .. 4 | 0 .. 3 |
| 8 .. 15 | 7 .. 14 | 7 | 6 | 1 .. 4 | 0 .. 3 |
| 9 .. 16 | 8 .. 15 | 8 | 7 | 1 .. 4 | 0 .. 3 |

## 11.4.4 Modes of Operation

### 11.4.4.1 Normal System Operating Modes

The MSCAN module behaves as described within this specification in all normal system operating modes. Write restrictions exist for some registers.

### 17.3.2.2 Control Register (VREGCTRL)

The VREGCTRL register allows the configuration of the VREG_3V3 low-voltage detect features.

0x02F1

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | 0 | LVDS | LVIE | LVIF |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 17-2. Control Register (VREGCTRL)**

**Table 17-5. VREGCTRL Field Descriptions**

| Field | Description |
|---|---|
| 2<br>LVDS | **Low-Voltage Detect Status Bit** — This read-only status bit reflects the input voltage. Writes have no effect.<br>0 Input voltage $V_{DDA}$ is above level $V_{LVID}$ or RPM or shutdown mode.<br>1 Input voltage $V_{DDA}$ is below level $V_{LVIA}$ and FPM. |
| 1<br>LVIE | **Low-Voltage Interrupt Enable Bit**<br>0 Interrupt request is disabled.<br>1 Interrupt will be requested whenever LVIF is set. |
| 0<br>LVIF | **Low-Voltage Interrupt Flag** — LVIF is set to 1 when LVDS status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (LVIE = 1), LVIF causes an interrupt request.<br>0 No change in LVDS bit.<br>1 LVDS bit has changed.<br>**Note:** On entering the Reduced Power Mode the LVIF is not cleared by the VREG_3V3. |

Upon clearing CCIF to launch the Set Field Margin Level command, the Memory Controller will set the field margin level for the targeted block and then set the CCIF flag. Valid margin level settings for the Set Field Margin Level command are defined in Table 18-57.

**Table 18-57. Valid Set Field Margin Level Settings**

| CCOB (CCOBIX=001) | Level Description |
|---|---|
| 0x0000 | Return to Normal Level |
| 0x0001 | User Margin-1 Level[1] |
| 0x0002 | User Margin-0 Level[2] |
| 0x0003 | Field Margin-1 Level[1] |
| 0x0004 | Field Margin-0 Level[2] |

1 Read margin to the erased state

2 Read margin to the programmed state

**Table 18-58. Set Field Margin Level Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 001 at command launch |
| | | Set if command not available in current mode (see Table 18-28) |
| | | Set if an invalid global address [22:16] is supplied |
| | | Set if an invalid margin level setting is supplied |
| | FPVIOL | None |
| | MGSTAT1 | None |
| | MGSTAT0 | None |

CAUTION

Field margin levels must only be used during verify of the initial factory programming.

**NOTE**

Field margin levels can be used to check that Flash memory contents have adequate margin for data retention at the normal level setting. If unexpected results are encountered when checking Flash memory contents at field margin levels, the Flash memory contents should be erased and reprogrammed.

### 18.4.2.14 Erase Verify D-Flash Section Command

The Erase Verify D-Flash Section command will verify that a section of code in the D-Flash is erased. The Erase Verify D-Flash Section command defines the starting point of the data to be verified and the number of words.

**Table 20-16. FERSTAT Field Descriptions**

| Field | Description |
|---|---|
| 1<br>DFDIF | **Double Bit Fault Detect Interrupt Flag** — The setting of the DFDIF flag indicates that a double bit fault was detected in the stored parity and data bits during a Flash array read operation or that a Flash array read operation was attempted on a Flash block that was under a Flash command operation. The DFDIF flag is cleared by writing a 1 to DFDIF. Writing a 0 to DFDIF has no effect on DFDIF.<br>0 No double bit fault detected<br>1 Double bit fault detected or an invalid Flash array read operation attempted |
| 0<br>SFDIF | **Single Bit Fault Detect Interrupt Flag** — With the IGNSF bit in the FCNFG register clear, the SFDIF flag indicates that a single bit fault was detected in the stored parity and data bits during a Flash array read operation or that a Flash array read operation was attempted on a Flash block that was under a Flash command operation. The SFDIF flag is cleared by writing a 1 to SFDIF. Writing a 0 to SFDIF has no effect on SFDIF.<br>0 No single bit fault detected<br>1 Single bit fault detected and corrected or an invalid Flash array read operation attempted |

### 20.3.2.9 P-Flash Protection Register (FPROT)

The FPROT register defines which P-Flash sectors are protected against program and erase operations.

Offset Module Base + 0x0008

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | FPOPEN | RNV6 | FPHDIS | FPHS[1:0] | | FPLDIS | FPLS[1:0] | |
| W | | | | | | | | |
| Reset | F | F | F | F | F | F | F | F |

= Unimplemented or Reserved

**Figure 20-13. Flash Protection Register (FPROT)**

The (unreserved) bits of the FPROT register are writable with the restriction that the size of the protected region can only be increased (see Section 20.3.2.9.1, “P-Flash Protection Restrictions,” and Table 20-21).

During the reset sequence, the FPROT register is loaded with the contents of the P-Flash protection byte in the Flash configuration field at global address 0x7F_FF0C located in P-Flash memory (see Table 20-3) as indicated by reset condition ‘F’ in Figure 20-13. To change the P-Flash protection that will be loaded during the reset sequence, the upper sector of the P-Flash memory must be unprotected, then the P-Flash protection byte must be reprogrammed. If a double bit fault is detected while reading the P-Flash phrase containing the P-Flash protection byte during the reset sequence, the FPOPEN bit will be cleared and remaining bits in the FPROT register will be set to leave the P-Flash memory fully protected.

Trying to alter data in any protected area in the P-Flash memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. The block erase of a P-Flash block is not possible if any of the P-Flash sectors contained in the same P-Flash block are protected.

**Table 20-17. FPROT Field Descriptions**

| Field | Description |
|---|---|
| 7<br>FPOPEN | **Flash Protection Operation Enable** — The FPOPEN bit determines the protection function for program or erase operations as shown in Table 20-18 for the P-Flash block.<br>0 When FPOPEN is clear, the FPHDIS and FPLDIS bits define unprotected address ranges as specified by the corresponding FPHS and FPLS bits<br>1 When FPOPEN is set, the FPHDIS and FPLDIS bits enable protection for the address range specified by the corresponding FPHS and FPLS bits |
| 6<br>RNV[6] | **Reserved Nonvolatile Bit** — The RNV bit should remain in the erased state for future enhancements. |
| 5<br>FPHDIS | **Flash Protection Higher Address Range Disable** — The FPHDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory ending with global address 0x7F_FFFF.<br>0 Protection/Unprotection enabled<br>1 Protection/Unprotection disabled |
| 4–3<br>FPHS[1:0] | **Flash Protection Higher Address Size** — The FPHS bits determine the size of the protected/unprotected area in P-Flash memory as shown inTable 20-19. The FPHS bits can only be written to while the FPHDIS bit is set. |
| 2<br>FPLDIS | **Flash Protection Lower Address Range Disable** — The FPLDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory beginning with global address 0x7F_8000.<br>0 Protection/Unprotection enabled<br>1 Protection/Unprotection disabled |
| 1–0<br>FPLS[1:0] | **Flash Protection Lower Address Size** — The FPLS bits determine the size of the protected/unprotected area in P-Flash memory as shown in Table 20-20. The FPLS bits can only be written to while the FPLDIS bit is set. |

**Table 20-18. P-Flash Protection Function**

| FPOPEN | FPHDIS | FPLDIS | Function[1] |
|---|---|---|---|
| 1 | 1 | 1 | No P-Flash Protection |
| 1 | 1 | 0 | Protected Low Range |
| 1 | 0 | 1 | Protected High Range |
| 1 | 0 | 0 | Protected High and Low Ranges |
| 0 | 1 | 1 | Full P-Flash Memory Protected |
| 0 | 1 | 0 | Unprotected Low Range |
| 0 | 0 | 1 | Unprotected High Range |
| 0 | 0 | 0 | Unprotected High and Low Ranges |

1 For range sizes, refer to Table 20-19 and Table 20-20.

**Table 20-19. P-Flash Protection Higher Address Range**

| FPHS[1:0] | Global Address Range | Protected Size |
|---|---|---|
| 00 | 0x7F_F800–0x7F_FFFF | 2 Kbytes |
| 01 | 0x7F_F000–0x7F_FFFF | 4 Kbytes |
| 10 | 0x7F_E000–0x7F_FFFF | 8 Kbytes |
| 11 | 0x7F_C000–0x7F_FFFF | 16 Kbytes |

**Table 20-24. FCCOB - NVM Command Mode (Typical Usage)**

| CCOBIX[2:0] | Byte | FCCOB Parameter Fields (NVM Command Mode) |
|---|---|---|
| 011 | HI | Data 1 [15:8] |
| | LO | Data 1 [7:0] |
| 100 | HI | Data 2 [15:8] |
| | LO | Data 2 [7:0] |
| 101 | HI | Data 3 [15:8] |
| | LO | Data 3 [7:0] |

### 20.3.2.12 Flash Reserved0 Register (FRSV0)

This Flash register is reserved for factory testing.

Offset Module Base + 0x000C

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 20-18. Flash Reserved0 Register (FRSV0)**

All bits in the FRSV0 register read 0 and are not writable.

### 20.3.2.13 Flash Reserved1 Register (FRSV1)

This Flash register is reserved for factory testing.

Offset Module Base + 0x000D

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 20-19. Flash Reserved1 Register (FRSV1)**

All bits in the FRSV1 register read 0 and are not writable.

### 20.3.2.14 Flash ECC Error Results Register (FECCR)

The FECCR registers contain the result of a detected ECC fault for both single bit and double bit faults. The FECCR register provides access to several ECC related fields as defined by the ECCRIX index bits in the FECCRIX register (see Section 20.3.2.4). Once ECC fault information has been stored, no other

## 0x00C8–0x00CF Asynchronous Serial Interface (SCI0) Map

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00C8 | SCI0BDH[1] | R | IREN | TNP1 | TNP0 | SBR12 | SBR11 | SBR10 | SBR9 | SBR8 |
| | | W | | | | | | | | |
| 0x00C9 | SCI0BDL[1] | R | SBR7 | SBR6 | SBR5 | SBR4 | SBR3 | SBR2 | SBR1 | SBR0 |
| | | W | | | | | | | | |
| 0x00CA | SCI0CR1[1] | R | LOOPS | SCISWAI | RSRC | M | WAKE | ILT | PE | PT |
| | | W | | | | | | | | |
| 0x00C8 | SCI0ASR1[2] | R | RXEDGIF | 0 | 0 | 0 | 0 | BERRV | BERRIF | BKDIF |
| | | W | | | | | | | | |
| 0x00C9 | SCI0ACR1[2] | R | RXEDGIE | 0 | 0 | 0 | 0 | 0 | BERRIE | BKDIE |
| | | W | | | | | | | | |
| 0x00CA | SCI0ACR2[2] | R | 0 | 0 | 0 | 0 | 0 | BERRM1 | BERRM0 | BKDFE |
| | | W | | | | | | | | |
| 0x00CB | SCI0CR2 | R | TIE | TCIE | RIE | ILIE | TE | RE | RWU | SBK |
| | | W | | | | | | | | |
| 0x00CC | SCI0SR1 | R | TDRE | TC | RDRF | IDLE | OR | NF | FE | PF |
| | | W | | | | | | | | |
| 0x00CD | SCI0SR2 | R | AMAP | 0 | 0 | TXPOL | RXPOL | BRK13 | TXDIR | RAF |
| | | W | | | | | | | | |
| 0x00CE | SCI0DRH | R | R8 | T8 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x00CF | SCI0DRL | R | R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 |
| | | W | T7 | T6 | T5 | T4 | T3 | T2 | T1 | T0 |

1 Those registers are accessible if the AMAP bit in the SCI0SR2 register is set to zero
2 Those registers are accessible if the AMAP bit in the SCI0SR2 register is set to one

## 0x00D0–0x00D7 Asynchronous Serial Interface (SCI1) Map

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00D0 | SCI1BDH[1] | R | IREN | TNP1 | TNP0 | SBR12 | SBR11 | SBR10 | SBR9 | SBR8 |
| | | W | | | | | | | | |
| 0x00D1 | SCI1BDL[1] | R | SBR7 | SBR6 | SBR5 | SBR4 | SBR3 | SBR2 | SBR1 | SBR0 |
| | | W | | | | | | | | |
| 0x00D2 | SCI1CR1[1] | R | LOOPS | SCISWAI | RSRC | M | WAKE | ILT | PE | PT |
| | | W | | | | | | | | |
| 0x00D0 | SCI1ASR1[2] | R | RXEDGIF | 0 | 0 | 0 | 0 | BERRV | BERRIF | BKDIF |
| | | W | | | | | | | | |
| 0x00D1 | SCI1ACR1[2] | R | RXEDGIE | 0 | 0 | 0 | 0 | 0 | BERRIE | BKDIE |
| | | W | | | | | | | | |
| 0x00D2 | SCI1ACR2[2] | R | 0 | 0 | 0 | 0 | 0 | BERRM1 | BERRM0 | BKDFE |
| | | W | | | | | | | | |
| 0x00D3 | SCI1CR2 | R | TIE | TCIE | RIE | ILIE | TE | RE | RWU | SBK |
| | | W | | | | | | | | |
| 0x00D4 | SCI1SR1 | R | TDRE | TC | RDRF | IDLE | OR | NF | FE | PF |
| | | W | | | | | | | | |

## 0x0240–0x027F Port Integration Module (PIM) Map 5 of 5

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0276 | PER0AD0 | R<br>W | PER0AD07 | PER0AD06 | PER0AD05 | PER0AD04 | PER0AD03 | PER0AD02 | PER0AD01 | PER0AD00 |
| 0x0277 | PER1AD0 | R<br>W | PER1AD07 | PER1AD06 | PER1AD05 | PER1AD04 | PER1AD03 | PER1AD02 | PER1AD01 | PER1AD00 |
| 0x0278-<br>0x027F | Reserved | R<br>W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 0x0280–0x02BF Reserved Register Space

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0280-<br>0x02BF | Reserved | R<br>W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 0x02C0–0x02EF Analog-to-Digital Converter 12-Bit 16-Channel (ATD0) Map

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x02C0 | ATD0CTL0 | R<br>W | 0 | 0 | 0 | 0 | WRAP3 | WRAP2 | WRAP1 | WRAP0 |
| 0x02C1 | ATD0CTL1 | R<br>W | ETRIGSEL | SRES1 | SRES0 | SMP_DIS | ETRIGCH3 | ETRIGCH2 | ETRIGCH1 | ETRIGCH0 |
| 0x02C2 | ATD0CTL2 | R<br>W | 0 | AFFC | ICLKSTP | ETRIGLE | ETRIGP | ETRIGE | ASCIE | ACMPIE |
| 0x02C3 | ATD0CTL3 | R<br>W | DJM | S8C | S4C | S2C | S1C | FIFO | FRZ1 | FRZ0 |
| 0x02C4 | ATD0CTL4 | R<br>W | SMP2 | SMP1 | SMP0 | PRS4 | PRS3 | PRS2 | PRS1 | PRS0 |
| 0x02C5 | ATD0CTL5 | R<br>W | 0 | SC | SCAN | MULT | CD | CC | CB | CA |
| 0x02C6 | ATD0STAT0 | R<br>W | SCF | 0 | ETORF | FIFOR | CC3 | CC2 | CC1 | CC0 |
| 0x02C7 | Reserved | R<br>W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x02C8 | ATD0CMPEH | R<br>W | CMPE15 | CMPE14 | CMPE13 | CMPE12 | CMPE11 | CMPE10 | CMPE9 | CMPE8 |
| 0x02C9 | ATD0CMPEL | R<br>W | CMPE7 | CMPE6 | CMPE5 | CMPE4 | CMPE3 | CMPE2 | CMPE1 | CMPE0 |
| 0x02CA | ATD0STAT2H | R<br>W | CCF15 | CCF14 | CCF13 | CCF12 | CCF11 | CCF10 | CCF9 | CCF8 |
| 0x02CB | ATD0STAT2L | R<br>W | CCF7 | CCF6 | CCF5 | CCF4 | CCF3 | CCF2 | CCF1 | CCF0 |
| 0x02CC | ATD0DIENH | R<br>W | IEN15 | IEN14 | IEN13 | IEN12 | IEN11 | IEN10 | IEN9 | IEN8 |
| 0x02CD | ATD0DIENL | R<br>W | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| 0x02CE | ATD0CMPHTH | R<br>W | CMPHT15 | CMPHT14 | CMPHT13 | CMPHT12 | CMPHT11 | CMPHT10 | CMPHT9 | CMPHT8 |

## 0x0300–0x0327 Pulse Width Modulator 8-Bit 8-Channel (PWM) Map

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0300 | PWME | R | PWME7 | PWME6 | PWME5 | PWME4 | PWME3 | PWME2 | PWME1 | PWME0 |
| | | W | | | | | | | | |
| 0x0301 | PWMPOL | R | PPOL7 | PPOL6 | PPOL5 | PPOL4 | PPOL3 | PPOL2 | PPOL1 | PPOL0 |
| | | W | | | | | | | | |
| 0x0302 | PWMCLK | R | PCLK7 | PCLK6 | PCLK5 | PCLK4 | PCLK3 | PCLK2 | PCLK1 | PCLK0 |
| | | W | | | | | | | | |
| 0x0303 | PWMPRCLK | R | 0 | PCKB2 | PCKB1 | PCKB0 | 0 | PCKA2 | PCKA1 | PCKA0 |
| | | W | | | | | | | | |
| 0x0304 | PWMCAE | R | CAE7 | CAE6 | CAE5 | CAE4 | CAE3 | CAE2 | CAE1 | CAE0 |
| | | W | | | | | | | | |
| 0x0305 | PWMCTL | R | CON67 | CON45 | CON23 | CON01 | PSWAI | PFRZ | 0 | 0 |
| | | W | | | | | | | | |
| 0x0306 | PWMTST Test Only | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0307 | PWMPRSC | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0308 | PWMSCLA | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | | | | | | | | |
| 0x0309 | PWMSCLB | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | | | | | | | | |
| 0x030A | PWMSCNTA | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x030B | PWMSCNTB | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x030C | PWMCNT0 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x030D | PWMCNT1 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x030E | PWMCNT2 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x030F | PWMCNT3 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x0310 | PWMCNT4 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x0311 | PWMCNT5 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x0312 | PWMCNT6 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x0313 | PWMCNT7 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x0314 | PWMPER0 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | | | | | | | | |
| 0x0315 | PWMPER1 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | | | | | | | | |
| 0x0316 | PWMPER2 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | | | | | | | | |