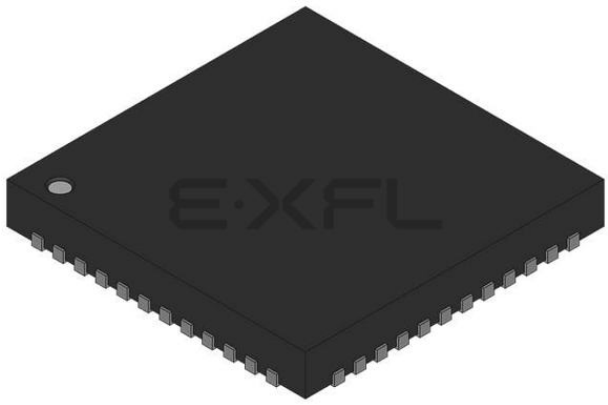

**Welcome to E-XFL.COM**

## 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 | Active |
| Core Processor | HCS12 |
| Core Size | 16-Bit |
| Speed | 32MHz |
| Connectivity | CANbus, SCI, SPI |
| Peripherals | LVD, POR, PWM, WDT |
| Number of I/O | 34 |
| Program Memory Size | 128KB (128K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 4K x 8 |
| RAM Size | 6K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.72V ~ 5.5V |
| Data Converters | A/D 10x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 48-TFQFN Exposed Pad |
| Supplier Device Package | 48-QFN-EP (7x7) |
| Purchase URL | https://www.e-xfl.com/pro/item?MUrl=&PartUrl=s9s12p128j0vft |

- Pulse width modulation (PWM) module with 6 x 8-bit channels
- 10-channel, 12-bit resolution successive approximation analog-to-digital converter (ATD)
- One serial peripheral interface (SPI) module
- One serial communication interface (SCI) module supporting LIN communications
- One multi-scalable controller area network (MSCAN) module (supporting CAN protocol 2.0A/B)
- On-chip voltage regulator (VREG) for regulation of input supply and all internal voltages
- Autonomous periodic interrupt (API)

## 1.3 Module Features

The following sections provide more details of the modules implemented on the MC9S12P family.

### 1.3.1 S12 16-Bit Central Processor Unit (CPU)

S12 CPU is a high-speed 16-bit processing unit:

- Full 16-bit data paths supports efficient arithmetic operation and high-speed math execution
- Includes many single-byte instructions. This allows much more efficient use of ROM space.
- Extensive set of indexed addressing capabilities, including:
  - Using the stack pointer as an indexing register in all indexed operations
  - Using the program counter as an indexing register in all but auto increment/decrement mode
  - Accumulator offsets using A, B, or D accumulators
  - Automatic index predecrement, preincrement, postdecrement, and postincrement (by –8 to +8)

### 1.3.2 On-Chip Flash with ECC

On-chip flash memory on the MC9S12P features the following:

- Up to 128 Kbyte of program flash memory
  - 32 data bits plus 7 syndrome ECC (error correction code) bits allow single bit error correction and double fault detection
  - Erase sector size 512 bytes
  - Automated program and erase algorithm
  - User margin level setting for reads
  - Protection scheme to prevent accidental program or erase
- 4 Kbyte data flash space
  - 16 data bits plus 6 syndrome ECC (error correction code) bits allow single bit error correction and double fault detection
  - Erase sector size 256 bytes
  - Automated program and erase algorithm
  - User margin level setting for reads

## 1.5 Device Memory Map

Table 1-2 shows the device register memory map.

**Table 1-2. 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 |
| 0x0030–0x0033 | Reserved | 4 |
| 0x0034–0x003F | CPMU (clock and power management) | 12 |
| 0x0040–0x006F | TIM (timer module) | 48 |
| 0x0070–0x009F | ATD (analog-to-digital converter 12 bit 10-channel) | 48 |
| 0x00A0–0x00C7 | PWM (pulse-width modulator 6 channels) | 40 |
| 0x00C8–0x00CF | SCI (serial communications interface) | 8 |
| 0x00D0–0x00D7 | Reserved | 8 |
| 0x00D8–0x00DF | SPI (serial peripheral interface) | 8 |
| 0x00E0–0x00FF | Reserved | 32 |
| 0x0100–0x0113 | FTMRC control registers | 20 |
| 0x0114–0x011F | Reserved | 12 |
| 0x0120 | INT (interrupt module) | 1 |
| 0x0121–0x013F | Reserved | 31 |
| 0x0140–0x017F | CAN | 64 |
| 0x0180–0x023F | Reserved | 192 |
| 0x0240–0x027F | PIM (port integration module) | 64 |
| 0x0280–0x02BF | Reserved | 64 |
| 0x02C0–0x02EF | Reserved | 48 |
| 0x02F0–0x02FF | CPMU (clock and power management ) | 16 |
| 0x0300–0x03FF | Reserved | 256 |

**NOTE**

Reserved register space shown in Table 1-2 is not allocated to any module. This register space is reserved for future use. Writing to these locations have no effect. Read access to these locations returns zero.

Figure 1-2 shows S12P CPU and BDM local address translation to the global memory map. It indicates also the location of the internal resources in the memory map. Table 1-3. shows the mapping of D-Flash and unpaged P-Flash memory. The whole 256K global memory space is visible through the P-Flash window located in the 64k local memory map located at 0x8000 - 0xBFFF using the PPAGE register.

**Table 1-3. MC9S12P -Family mapping for D-Flash and unpaged P-Flash**

| | Local 64K memory map | Global 256K memory map |
|---|---|---|
| D-Flash | 0x0400 - 0x13FF | 0x0_4400 - 0x0_53FF |
| P-Flash | 0x1400 - 0x27FF(1) | 0x3_1400 -0x3_27FF(2) |
| | 0x4000 - 0x7FFF | 0x3_4000 - 0x3_7FFF |
| | 0xC000 - 0xFFFF | 0x3_C000 - 0x3_FFFF |

1. 0x2FFF for MC9S12P64 because of 4K RAM size
2. 0x3_2FFF for MC9S12P64 because of 4K RAM size

**Table 1-4. Derivatives**

| Feature | MC9S12P32 | MC9S12P64 | MC9S12P96 | MC9S12P128 |
|---|---|---|---|---|
| P-Flash size | 32KB | 64KB | 96KB | 128KB |
| PF_LOW<br>PPAGES | 0x3_8000<br>0x0E - 0x0F | 0x3_0000<br>0x0C - 0x0F | 0x2_8000<br>0x0A - 0x0F | 0x2_0000<br>0x08 - 0x0F |
| RAMSIZE | 2KB | 4KB | 6KB | |
| RAM_LOW | 0x0_3800 | 0x0_3000 | 0x0_2800 | |

**Table 1-6. Port Availability by Package Option**

| Port | 80 QFP | 64 LQFP | 48 QFN |
|---|---|---|---|
| Port T | 8 | 8 | 8 |
| **Sum of Ports** | **64** | **49** | **34** |
| I/O Power Pairs VDDX/VSSX | 2/2 | 2/2 | 2/2 |

**Table 1-7. Peripheral - Port Routing Options[1]**

| | PWM0 | PWM4 | PWM5 |
|---|---|---|---|
| PT0 | O | | |
| PT4 | | O | |
| PT5 | | | O |

1. "O" denotes a possible rerouting under software control

Table 1-8 provides a pin out summary listing the availability and functionality of individual pins for each package option.

For example selecting a pull-up device: This device does not become active while the port is used as a push-pull output.

**Table 2-59. Register availability per port[1]**

| Port | Data | Input | Data Direction | Reduced Drive | Pull Enable | Polarity Select | Wired-Or Mode | Interrupt Enable | Interrupt Flag | Routing |
|---|---|---|---|---|---|---|---|---|---|---|
| A | yes | - | yes | yes | yes | - | - | - | - | - |
| B | yes | - | yes | | | - | - | - | - | - |
| E | yes | - | yes | | | - | - | - | - | - |
| T | yes | yes | yes | yes | yes | yes | - | - | - | yes |
| S | yes | yes | yes | yes | yes | yes | yes | - | - | - |
| M | yes | yes | yes | yes | yes | yes | yes | - | - | yes |
| P | yes | yes | yes | yes | yes | yes | - | yes | yes | - |
| J | yes | yes | yes | yes | yes | yes | - | yes | yes | - |
| AD | yes | - | yes | yes | yes | - | - | - | - | - |

1. Each cell represents one register with individual configuration bits

### 2.4.2.1 Data register (PORTx, PTx)

This register holds the value driven out to the pin if the pin is used as a general purpose I/O.

Writing to this register has only an effect on the pin if the pin is used as general purpose output. When reading this address, the buffered state of the pin is returned if the associated data direction register bit is set to “0”.

If the data direction register bits are set to logic level “1”, the contents of the data register is returned. This is independent of any other configuration (Figure 2-64).

### 2.4.2.2 Input register (PTIx)

This register is read-only and always returns the buffered state of the pin (Figure 2-64).

### 2.4.2.3 Data direction register (DDRx)

This register defines whether the pin is used as an general purpose input or an output.

If a peripheral module controls the pin the contents of the data direction register is ignored (Figure 2-64).

Independent of the pin usage with a peripheral module this register determines the source of data when reading the associated data register address (2.4.2.1/2-101).

**NOTE**

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on port data or port input registers, when changing the data direction register.

Port E pin PE[0] can be used for either general purpose input or as the level-sensitive $\overline{\text{XIRQ}}$ interrupt. $\overline{\text{XIRQ}}$ can be enabled by clearing the X-bit in the CPU condition code register. It is inhibited at reset so this pin is initially configured as a high-impedance input with a pull-up.

#### 2.4.3.4 Port T

This port is associated with TIM and PWM.

Port T pins PT[5:4,0] can be used for either general purpose I/O, or with the routed PWM or with the channels of the standard Timer subsystem.

Port T pins PT[7:6,3:1] can be used for either general purpose I/O, or with the channels of the standard Timer subsystem.

#### 2.4.3.5 Port S

This port is associated with SCI.

Port S pins PS[1:0] can be used either for general purpose I/O, or with the SCI subsystem.

Port S pins PS[3:2] can be used for general purpose I/O.

#### 2.4.3.6 Port M

This port is associated with CAN and SPI.

Port M pins PM[1:0] can be used for either general purpose I/O, or with the CAN subsystem.

Port M pins PM[5:2] can be used for general purpose I/O, or with the SPI subsystem.

#### 2.4.3.7 Port P

This port is associated with the PWM.

Port P pins PP[7,5:0] can be used for either general purpose I/O with pin interrupt capability, or with the PWM subsystem.

#### 2.4.3.8 Port J

Port J pins PJ[7:6,2:0] can be used for general purpose I/O with pin-interrupt capability.

#### 2.4.3.9 Port AD

This port is associated with the ATD.

Port AD pins PAD[9:0] can be used for either general purpose I/O, or with the ATD subsystem.

For hardware data read commands, the external host must wait at least 150 bus clock cycles after sending the address before attempting to obtain the read data. This is to be certain that valid data is available in the BDM shift register, ready to be shifted out. For hardware write commands, the external host must wait 150 bus clock cycles after sending the data to be written before attempting to send a new command. This is to avoid disturbing the BDM shift register before the write has been completed. The 150 bus clock cycle delay in both cases includes the maximum 128 cycle delay that can be incurred as the BDM waits for a free cycle before stealing a cycle.

For BDM firmware read commands, the external host should wait at least 48 bus clock cycles after sending the command opcode and before attempting to obtain the read data. The 48 cycle wait allows enough time for the requested data to be made available in the BDM shift register, ready to be shifted out.

For BDM firmware write commands, the external host must wait 36 bus clock cycles after sending the data to be written before attempting to send a new command. This is to avoid disturbing the BDM shift register before the write has been completed.

The external host should wait for at least for 76 bus clock cycles after a TRACE1 or GO command before starting any new serial command. This is to allow the CPU to exit gracefully from the standard BDM firmware lookup table and resume execution of the user code. Disturbing the BDM shift register prematurely may adversely affect the exit from the standard BDM firmware lookup table.

**NOTE**

If the bus rate of the target processor is unknown or could be changing, it is recommended that the ACK (acknowledge function) is used to indicate when an operation is complete. When using ACK, the delay times are automated.

Figure 5-6 represents the BDM command structure. The command blocks illustrate a series of eight bit times starting with a falling edge. The bar across the top of the blocks indicates that the BKGD line idles in the high state. The time for an 8-bit command is $8 \times 16$ target clock cycles.[1]

1. Target clock cycles are cycles measured using the target MCU's serial clock rate. See Section 5.4.6, "BDM Serial Interface" and Section 5.3.2.1, "BDM Status Register (BDMSTS)" for information on how serial clock rate is selected.

each trace buffer entry. In Detail mode CINF comprises of R/W and size access information (CRW and CSZ respectively).

Single byte data accesses in Detail Mode are always stored to the low byte of the trace buffer (DATAL) and the high byte is cleared. When tracing word accesses, the byte at the lower address is always stored to trace buffer byte1 and the byte at the higher address is stored to byte0.

**Table 6-37. Trace Buffer Organization (Normal,Loop1,Detail modes)**

| Mode | Entry Number | 4-bits | 8-bits | 8-bits |
|---|---|---|---|---|
| | | Field 2 | Field 1 | Field 0 |
| Detail Mode | Entry 1 | CINF1,ADRH1 | ADRM1 | ADRL1 |
| | | 0 | DATAH1 | DATAL1 |
| | Entry 2 | CINF2,ADRH2 | ADRM2 | ADRL2 |
| | | 0 | DATAH2 | DATAL2 |
| Normal/Loop1 Modes | Entry 1 | PCH1 | PCM1 | PCL1 |
| | Entry 2 | PCH2 | PCM2 | PCL2 |

#### 6.4.5.3.1 Information Bit Organization

The format of the bits is dependent upon the active trace mode as described below.

### Field2 Bits in Detail Mode

| Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|
| CSZ | CRW | ADDR[17] | ADDR[16] |

**Figure 6-25. Field2 Bits in Detail Mode**

In Detail Mode the CSZ and CRW bits indicate the type of access being made by the CPU.

**Table 6-38. Field Descriptions**

| Bit | Description |
|---|---|
| 3<br>CSZ | **Access Type Indicator**— This bit indicates if the access was a byte or word size when tracing in Detail Mode<br>0 Word Access<br>1 Byte Access |
| 2<br>CRW | **Read Write Indicator** — This bit indicates if the corresponding stored address corresponds to a read or write access when tracing in Detail Mode.<br>0 Write Access<br>1 Read Access |
| 1<br>ADDR[17] | **Address Bus bit 17**— Corresponds to system address bus bit 17. |
| 0<br>ADDR[16] | **Address Bus bit 16**— Corresponds to system address bus bit 16. |

**Table 8-3. CANCTL0 Register Field Descriptions (continued)**

| Field | Description |
|---|---|
| 2<br>WUPE[4] | **Wake-Up Enable** — This configuration bit allows the MSCAN to restart from sleep mode or from power down mode (entered from sleep) when traffic on CAN is detected (see Section 8.4.5.5, "MSCAN Sleep Mode"). This bit must be configured before sleep mode entry for the selected function to take effect.<br>0 Wake-up disabled — The MSCAN ignores traffic on CAN<br>1 Wake-up enabled — The MSCAN is able to restart |
| 1<br>SLPRQ[5] | **Sleep Mode Request** — This bit requests the MSCAN to enter sleep mode, which is an internal power saving mode (see Section 8.4.5.5, "MSCAN Sleep Mode"). The sleep mode request is serviced when the CAN bus is idle, i.e., the module is not receiving a message and all transmit buffers are empty. The module indicates entry to sleep mode by setting SLPAK = 1 (see Section 8.3.2.2, "MSCAN Control Register 1 (CANCTL1)"). SLPRQ cannot be set while the WUPIF flag is set (see Section 8.3.2.5, "MSCAN Receiver Flag Register (CANRFLG)"). Sleep mode will be active until SLPRQ is cleared by the CPU or, depending on the setting of WUPE, the MSCAN detects activity on the CAN bus and clears SLPRQ itself.<br>0 Running — The MSCAN functions normally<br>1 Sleep mode request — The MSCAN enters sleep mode when CAN bus idle |
| 0<br>INITRQ[6],[7] | **Initialization Mode Request** — When this bit is set by the CPU, the MSCAN skips to initialization mode (see Section 8.4.4.5, "MSCAN Initialization Mode"). Any ongoing transmission or reception is aborted and synchronization to the CAN bus is lost. The module indicates entry to initialization mode by setting INITAK = 1 (Section 8.3.2.2, "MSCAN Control Register 1 (CANCTL1)").<br>The following registers enter their hard reset state and restore their default values: CANCTL0[8], CANRFLG[9], CANRIER[10], CANTFLG, CANTIER, CANTARQ, CANTAAK, and CANTBSEL.<br>The registers CANCTL1, CANBTR0, CANBTR1, CANIDAC, CANIDAR0-7, and CANIDMR0-7 can only be written by the CPU when the MSCAN is in initialization mode (INITRQ = 1 and INITAK = 1). The values of the error counters are not affected by initialization mode.<br>When this bit is cleared by the CPU, the MSCAN restarts and then tries to synchronize to the CAN bus. If the MSCAN is not in bus-off state, it synchronizes after 11 consecutive recessive bits on the CAN bus; if the MSCAN is in bus-off state, it continues to wait for 128 occurrences of 11 consecutive recessive bits.<br>Writing to other bits in CANCTL0, CANRFLG, CANRIER, CANTFLG, or CANTIER must be done only after initialization mode is exited, which is INITRQ = 0 and INITAK = 0.<br>0 Normal operation<br>1 MSCAN in initialization mode |

1. The MSCAN must be in normal mode for this bit to become set.
2. See the Bosch CAN 2.0A/B specification for a detailed definition of transmitter and receiver states.
3. In order to protect from accidentally violating the CAN protocol, TXCAN is immediately forced to a recessive state when the CPU enters wait (CSWAI = 1) or stop mode (see Section 8.4.5.2, "Operation in Wait Mode" and Section 8.4.5.3, "Operation in Stop Mode").
4. The CPU has to make sure that the WUPE register and the WUPIE wake-up interrupt enable register (see Section 8.3.2.6, "MSCAN Receiver Interrupt Enable Register (CANRIER)) is enabled, if the recovery mechanism from stop or wait is required.
5. The CPU cannot clear SLPRQ before the MSCAN has entered sleep mode (SLPRQ = 1 and SLPAK = 1).
6. The CPU cannot clear INITRQ before the MSCAN has entered initialization mode (INITRQ = 1 and INITAK = 1).
7. In order to protect from accidentally violating the CAN protocol, TXCAN is immediately forced to a recessive state when the initialization mode is requested by the CPU. Thus, the recommended procedure is to bring the MSCAN into sleep mode (SLPRQ = 1 and SLPAK = 1) before requesting initialization mode.
8. Not including WUPE, INITRQ, and SLPRQ.
9. TSTAT1 and TSTAT0 are not affected by initialization mode.
10. RSTAT1 and RSTAT0 are not affected by initialization mode.

### 8.3.3.3 Data Length Register (DLR)

This register keeps the data length field of the CAN frame.

Module Base + 0x00XC

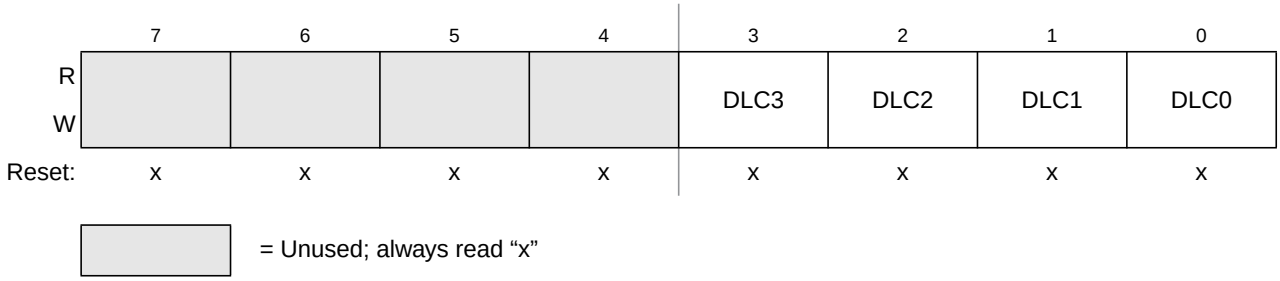


**Figure 8-35. Data Length Register (DLR) — Extended Identifier Mapping**

**Table 8-34. DLR Register Field Descriptions**

| Field | Description |
|---|---|
| 3-0<br>DLC[3:0] | **Data Length Code Bits** — The data length code contains the number of bytes (data byte count) of the respective message. During the transmission of a remote frame, the data length code is transmitted as programmed while the number of transmitted data bytes is always 0. The data byte count ranges from 0 to 8 for a data frame. Table 8-35 shows the effect of setting the DLC bits. |

**Table 8-35. Data Length Codes**

| Data Length Code | | | | Data Byte Count |
|---|---|---|---|---|
| DLC3 | DLC2 | DLC1 | DLC0 | |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 1 | 3 |
| 0 | 1 | 0 | 0 | 4 |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 1 | 0 | 6 |
| 0 | 1 | 1 | 1 | 7 |
| 1 | 0 | 0 | 0 | 8 |

### 8.3.3.4 Transmit Buffer Priority Register (TBPR)

This register defines the local priority of the associated message buffer. The local priority is used for the internal prioritization process of the MSCAN and is defined to be highest for the smallest binary number. The MSCAN implements the following internal prioritization mechanisms:

- All transmission buffers with a cleared TXEx flag participate in the prioritization immediately before the SOF (start of frame) is sent.
- The transmission buffer with the lowest local priority field wins the prioritization.

### 8.4.3.1 Protocol Violation Protection

The MSCAN protects the user from accidentally violating the CAN protocol through programming errors. The protection logic implements the following features:

- The receive and transmit error counters cannot be written or otherwise manipulated.
- All registers which control the configuration of the MSCAN cannot be modified while the MSCAN is on-line. The MSCAN has to be in Initialization Mode. The corresponding INITRQ/INITAK handshake bits in the CANCTL0/CANCTL1 registers (see Section 8.3.2.1, "MSCAN Control Register 0 (CANCTL0)") serve as a lock to protect the following registers:
  - MSCAN control 1 register (CANCTL1)
  - MSCAN bus timing registers 0 and 1 (CANBTR0, CANBTR1)
  - MSCAN identifier acceptance control register (CANIDAC)
  - MSCAN identifier acceptance registers (CANIDAR0–CANIDAR7)
  - MSCAN identifier mask registers (CANIDMR0–CANIDMR7)
- The TXCAN is immediately forced to a recessive state when the MSCAN goes into the power down mode or initialization mode (see Section 8.4.5.6, "MSCAN Power Down Mode," and Section 8.4.4.5, "MSCAN Initialization Mode").
- The MSCAN enable bit (CANE) is writable only once in normal system operation modes, which provides further protection against inadvertently disabling the MSCAN.

### 8.4.3.2 Clock System

Figure 8-43 shows the structure of the MSCAN clock generation circuitry.

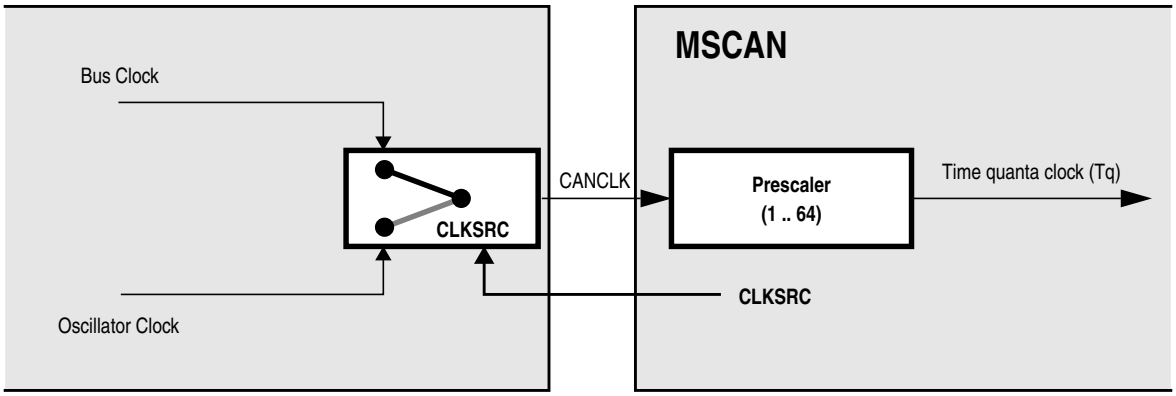


**Figure 8-43. MSCAN Clocking Scheme**

The clock source bit (CLKSRC) in the CANCTL1 register (8.3.2.2/8-259) defines whether the internal CANCLK is connected to the output of a crystal oscillator (oscillator clock) or to the bus clock.

The clock source has to be chosen such that the tight oscillator tolerance requirements (up to 0.4%) of the CAN protocol are met. Additionally, for high CAN bus rates (1 Mbps), a 45% to 55% duty cycle of the clock is required.

If the bus clock is generated from a PLL, it is recommended to select the oscillator clock rather than the bus clock due to jitter considerations, especially at the faster CAN bus rates.

| Address | Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0003 | ATDCTL3 | R | DJM | S8C | S4C | S2C | S1C | FIFO | FRZ1 | FRZ0 |
| | | W | | | | | | | | |
| 0x0004 | ATDCTL4 | R | SMP2 | SMP1 | SMP0 | PRS[4:0] | | | | |
| | | W | | | | | | | | |
| 0x0005 | ATDCTL5 | R | 0 | SC | SCAN | MULT | CD | CC | CB | CA |
| | | W | | | | | | | | |
| 0x0006 | ATDSTAT0 | R | SCF | 0 | ETORF | FIFOR | CC3 | CC2 | CC1 | CC0 |
| | | W | | | | | | | | |
| 0x0007 | Unimple-mented | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0008 | ATDCMPEH | R | 0 | 0 | 0 | 0 | 0 | 0 | CMPE[9:8] | |
| | | W | | | | | | | | |
| 0x0009 | ATDCMPEL | R | CMPE[7:0] | | | | | | | |
| | | W | | | | | | | | |
| 0x000A | ATDSTAT2H | R | 0 | 0 | 0 | 0 | 0 | 0 | CCF[9:8] | |
| | | W | | | | | | | | |
| 0x000B | ATDSTAT2L | R | CCF[7:0] | | | | | | | |
| | | W | | | | | | | | |
| 0x000C | ATDDIENH | R | 0 | 0 | 0 | 0 | 0 | 0 | IEN[9:8] | |
| | | W | | | | | | | | |
| 0x000D | ATDDIENL | R | IEN[7:0] | | | | | | | |
| | | W | | | | | | | | |
| 0x000E | ATDCMPHTH | R | 0 | 0 | 0 | 0 | 0 | 0 | CMPHT[9:8] | |
| | | W | | | | | | | | |
| 0x000F | ATDCMPHTL | R | CMPHT[7:0] | | | | | | | |
| | | W | | | | | | | | |
| 0x0010 | ATDDR0 | R | See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)” | | | | | | | |
| | | W | | | | | | | | |
| 0x0012 | ATDDR1 | R | See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)” | | | | | | | |
| | | W | | | | | | | | |
| 0x0014 | ATDDR2 | R | See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)” | | | | | | | |
| | | W | | | | | | | | |
| 0x0016 | ATDDR3 | R | See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)” | | | | | | | |
| | | W | | | | | | | | |
| 0x0018 | ATDDR4 | R | See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)” | | | | | | | |
| | | W | | | | | | | | |
| 0x001A | ATDDR5 | R | See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)” | | | | | | | |
| | | W | | | | | | | | |
| 0x001C | ATDDR6 | R | See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)” | | | | | | | |
| | | W | | | | | | | | |
| 0x001E | ATDDR7 | R | See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)” | | | | | | | |
| | | W | | | | | | | | |

= Unimplemented or Reserved

**Figure 9-2. ADC12B10C Register Summary (Sheet 2 of 3)**

### 9.4.1 Analog Sub-Block

The analog sub-block contains all analog electronics required to perform a single conversion. Separate power supplies $V_{DDA}$ and $V_{SSA}$ allow to isolate noise of other MCU circuitry from the analog sub-block.

#### 9.4.1.1 Sample and Hold Machine

The Sample and Hold (S/H) Machine accepts analog signals from the external world and stores them as capacitor charge on a storage node.

During the sample process the analog input connects directly to the storage node.

The input analog signals are unipolar and must fall within the potential range of $V_{SSA}$ to $V_{DDA}$.

During the hold process the analog input is disconnected from the storage node.

#### 9.4.1.2 Analog Input Multiplexer

The analog input multiplexer connects one of the 10 external analog input channels to the sample and hold machine.

#### 9.4.1.3 Analog-to-Digital (A/D) Machine

The A/D Machine performs analog to digital conversions. The resolution is program selectable at either 8 or 10 or 12 bits. The A/D machine uses a successive approximation architecture. It functions by comparing the stored analog sample potential with a series of digitally generated analog potentials. By following a binary search algorithm, the A/D machine locates the approximating potential that is nearest to the sampled potential.

When not converting the A/D machine is automatically powered down.

Only analog input signals within the potential range of $V_{RL}$ to $V_{RH}$ (A/D reference potentials) will result in a non-railed digital output code.

### 9.4.2 Digital Sub-Block

This subsection explains some of the digital features in more detail. See Section 9.3.2, “Register Descriptions” for all details.

#### 9.4.2.1 External Trigger Input

The external trigger feature allows the user to synchronize ATD conversions to the external environment events rather than relying on software to signal the ATD module when ATD conversions are to take place. The external trigger signal (out of reset ATD channel 9, configurable in ATDCTL1) is programmable to

Read: always read 0x0000 in normal modes

Write: unimplemented in normal modes

**NOTE**

Writing to this register when in special modes can alter the PWM functionality.

### 10.3.2.8 Reserved Register (PWMPRSC)

This register is reserved for factory testing of the PWM module and is not available in normal modes.

Module Base + 0x0007

| | 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 10-10. Reserved Register (PWMPRSC)**

Read: always read 0x0000 in normal modes

Write: unimplemented in normal modes

**NOTE**

Writing to this register when in special modes can alter the PWM functionality.

### 10.3.2.9 PWM Scale A Register (PWMSCLA)

PWMSCLA is the programmable scale value used in scaling clock A to generate clock SA. Clock SA is generated by taking clock A, dividing it by the value in the PWMSCLA register and dividing that by two.

Clock SA = Clock A / (2 * PWMSCLA)

**NOTE**

When PWMSCLA = 0x0000, PWMSCLA value is considered a full scale value of 256. Clock A is thus divided by 512.

Any value written to this register will cause the scale counter to load the new scale value (PWMSCLA).

Module Base + 0x0008

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

**Figure 10-11. PWM Scale A Register (PWMSCLA)**

due to the synchronization of PWMEx and the clock source. An exception to this is when channels are concatenated. Refer to Section 10.4.2.7, “PWM 16-Bit Functions,” for more detail.

**NOTE**

The first PWM cycle after enabling the channel can be irregular.

On the front end of the PWM timer, the clock is enabled to the PWM circuit by the PWMEx bit being high. There is an edge-synchronizing circuit to guarantee that the clock will only be enabled or disabled at an edge. When the channel is disabled (PWMEx = 0), the counter for the channel does not count.

### 10.4.2.2 PWM Polarity

Each channel has a polarity bit to allow starting a waveform cycle with a high or low signal. This is shown on the block diagram as a mux select of either the Q output or the $\overline{Q}$ output of the PWM output flip-flop. When one of the bits in the PWMPOL register is set, the associated PWM channel output is high at the beginning of the waveform, then goes low when the duty count is reached. Conversely, if the polarity bit is 0, the output starts low and then goes high when the duty count is reached.

### 10.4.2.3 PWM Period and Duty

Dedicated period and duty registers exist for each channel and are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to 0x0000)
- The channel is disabled

In this way, the output of the PWM will always be either the old waveform or the new waveform, not some variation in between. If the channel is not enabled, then writes to the period and duty registers will go directly to the latches as well as the buffer.

A change in duty or period can be forced into effect “immediately” by writing the new value to the duty and/or period registers and then writing to the counter. This forces the counter to reset and the new duty and/or period values to be latched. In addition, because the counter is readable it is possible to know where the count is with respect to the duty value and software can be used to make adjustments.

**NOTE**

When forcing a new period or duty into effect immediately, an irregular PWM cycle can occur.

Depending on the polarity bit, the duty registers will contain the count of either the high time or the low time.

### 10.4.2.4 PWM Timer Counters

Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source (reference Figure 10-34 for the available clock sources and rates). The counter compares to two registers, a duty register and a period register as shown in Figure 10-35. When the PWM counter matches the duty register the output flip-flop changes state causing the PWM waveform to also change state. A match

Table 10-12 is used to summarize which channels are used to set the various control bits when in 16-bit mode.

**Table 10-12. 16-bit Concatenation Mode Summary**

| CONxx | PWMEx | PPOLx | PCLKx | CAEx | PWMx Output |
|---|---|---|---|---|---|
| CON45 | PWME5 | PPOL5 | PCLK5 | CAE5 | PWM5 |
| CON23 | PWME3 | PPOL3 | PCLK3 | CAE3 | PWM3 |
| CON01 | PWME1 | PPOL1 | PCLK1 | CAE1 | PWM1 |

### 10.4.2.8 PWM Boundary Cases

Table 10-13 summarizes the boundary conditions for the PWM regardless of the output mode (left aligned or center aligned) and 8-bit (normal) or 16-bit (concatenation):

**Table 10-13. PWM Boundary Cases**

| PWMDTYx | PWMPERx | PPOLx | PWMx Output |
|---|---|---|---|
| 0x0000 (indicates no duty) | >0x0000 | 1 | Always Low |
| 0x0000 (indicates no duty) | >0x0000 | 0 | Always High |
| XX | 0x0000(1) (indicates no period) | 1 | Always High |
| XX | 0x0000[1] (indicates no period) | 0 | Always Low |
| >= PWMPERx | XX | 1 | Always High |
| >= PWMPERx | XX | 0 | Always Low |

1. Counter = 0x0000 and does not count.

## 10.5 Resets

The reset state of each individual bit is listed within the register description section (see Section 10.3, “Memory Map and Register Definition,” which details the registers and their bit-fields. All special functions or modes which are initialized during or just following reset are described within this section.

- The 8-bit up/down counter is configured as an up counter out of reset.
- All the channels are disabled and all the counters don’t count.

## 10.6 Interrupts

The PMW8B6CV1 module has only one interrupt which is generated at the time of emergency shutdown, if the corresponding enable bit (PWMIE) is set. This bit is the enable for the interrupt. The interrupt flag PWMIF is set whenever the input level of the PWM5 channel changes while PWM5ENA=1 or when PWMENA is being asserted while the level at PWM5 is active.

A description of the registers involved and affected due to this interrupt is explained in Section 10.3.2.15, “PWM Shutdown Register (PWMSDN).”

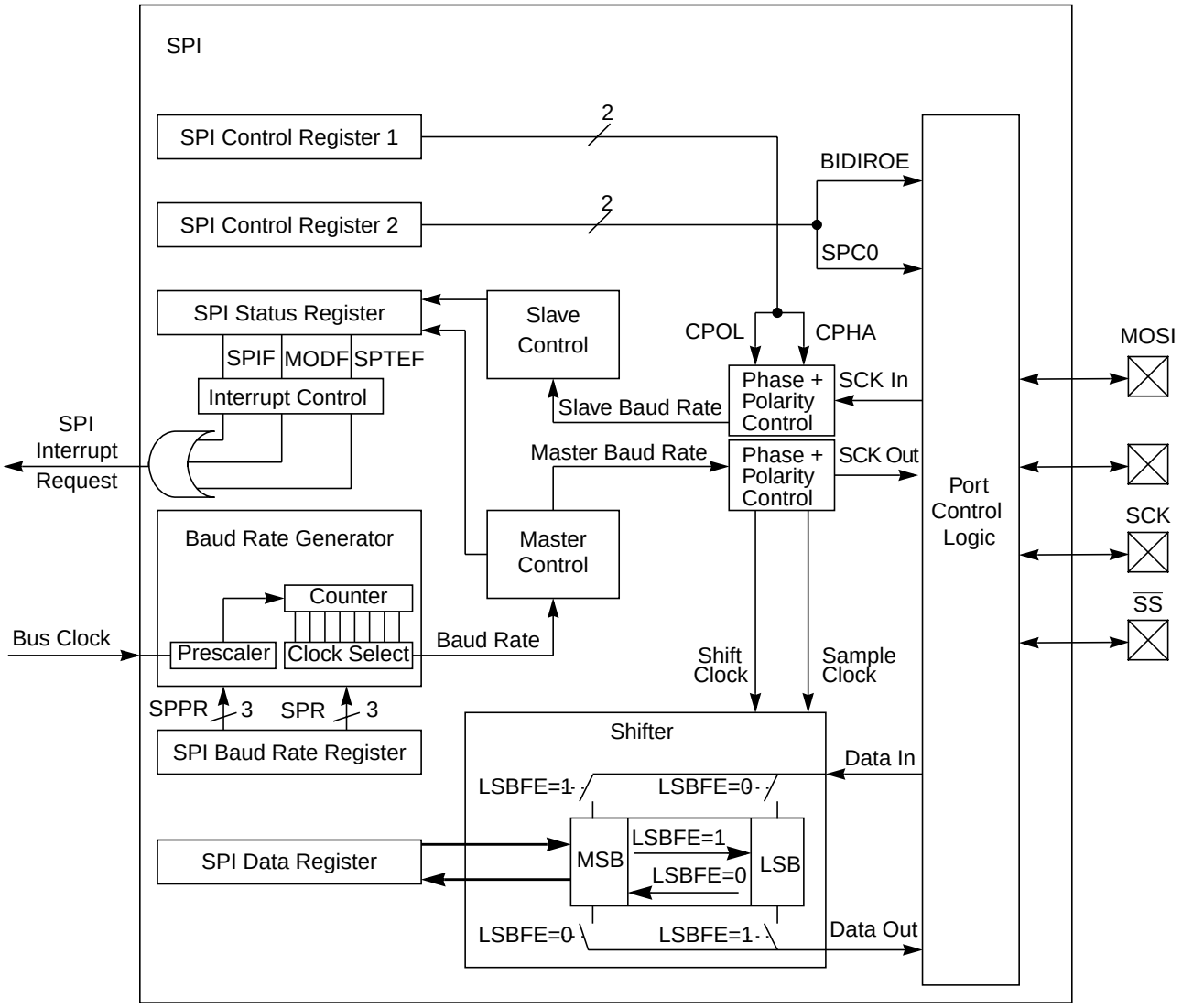


**Figure 12-1. SPI Block Diagram**

## 12.2 External Signal Description

This section lists the name and description of all ports including inputs and outputs that do, or may, connect off chip. The SPI module has a total of four external pins.

### 12.2.1 MOSI — Master Out/Slave In Pin

This pin is used to transmit data out of the SPI module when it is configured as a master and receive data when it is configured as slave.

### 12.2.2 MISO — Master In/Slave Out Pin

This pin is used to transmit data out of the SPI module when it is configured as a slave and receive data when it is configured as master.

## A.8 Electrical Specification for Voltage Regulator

**Table A-24. IVREG Characteristics**

| Num | C | Characteristic | Symbol | Min | Typical | Max | Unit |
|---|---|---|---|---|---|---|---|
| 1 | P | Input Voltages | $V_{VDDR,A}$ | 3.13 | — | 5.5 | V |
| 2 | P | VDDA Low Voltage Interrupt Assert Level (1)<br>VDDA Low Voltage Interrupt Deassert Level | $V_{LVIA}$<br>$V_{LVID}$ | 4.04<br>4.19 | 4.23<br>4.38 | 4.40<br>4.49 | V<br>V |
| 3 | P | VDDX Low Voltage Reset Deassert (2) (3) | $V_{LVRXD}$ | — | — | 3.13 | V |
| 4 | P | VDDX Low Voltage Reset Assert (2) (3) | $V_{LVRXA}$ | 2.95 | — | — | V |
| 5 | T | API ACLK frequency<br>(APITR[5:0] = %000000) | $f_{ACLK}$ | — | 10 | — | KHz |
| 6 | C | Trimmed API internal clock(4) $\Delta f / f_{nominal}$ | $df_{ACLK}$ | - 5% | — | + 5% | — |
| 7 | D | The first period after enabling the counter by APIFE might be reduced by API start up delay | $t_{sdel}$ | — | — | 100 | us |
| 8 | T | Temperature Sensor Slope | $dV_{TS}$ | 4.0 | 5.5 | 6.5 | mV/ $^{o}C$ |
| 9 | T | High Temperature Interrupt Assert (CPMUHTTR=$88)(5)<br>High Temperature Interrupt Deassert (CPMUHTTR=$88) | $T_{HTIA}$<br>$T_{HTID}$ | | 125<br>105 | | $^{o}C$ |
| 10 | T | Bandgap Reference Voltage | $V_{BG}$ | 1.13 | 1.21 | 1.32 | V |

1. Monitors VDDA, active only in Full Performance Mode. Indicates I/O & ADC performance degradation due to low supply voltage.
2. Device functionality is guaranteed on power down to the LVR assert level
3. Monitors VDDX, active only in Full Performance Mode. MCU is monitored by the POR in RPM (see Figure A-4)
4. The API Trimming APITR[5:0] bits must be set so that $f_{ACLK}$=10KHz.
5. A hysteresis is guaranteed by design

**NOTE**

The LVR monitors the voltages $V_{DD}$, $V_{DDF}$ and $V_{DDX}$. As soon as voltage drops on these supplies which would prohibit the correct function of the microcontroller, the LVR is triggering a reset.

## A.9 Chip Power-up and Voltage Drops

LVI (low voltage interrupt), POR (power-on reset) and LVRs (low voltage reset) handle chip power-up or drops of the supply voltage.

## 0x0020-0x002F Debug Module (S12SDBG) Map

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0028 (3) | DBGCCTL | R | 0 | 0 | TAG | BRK | RW | RWE | 0 | COMPE |
| | | W | | | | | | | | |
| 0x0029 | DBGXAH | R | 0 | 0 | 0 | 0 | 0 | 0 | 17 | Bit 16 |
| | | W | | | | | | | | |
| 0x002A | DBGXAM | R | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit 8 |
| | | W | | | | | | | | |
| 0x002B | DBGXAL | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | | | | | | | | |
| 0x002C | DBGADH | R | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit 8 |
| | | W | | | | | | | | |
| 0x002D | DBGADL | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | | | | | | | | |
| 0x002E | DBGADHM | R | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit 8 |
| | | W | | | | | | | | |
| 0x002F | DBGADLM | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | W | | | | | | | | |

1. This represents the contents if the Comparator A or C control register is blended into this address
2. This represents the contents if the Comparator B or D control register is blended into this address
3. This represents the contents if the Comparator B or D control register is blended into this address

## 0x0030-0x0033 Reserved

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0030 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0031 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0032 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0033 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

## 0x0034-0x003F Clock Reset and Power Management (CPMU) Map

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0034 | CPMUSYNR | R | VCOFRQ[1:0] | | SYNDIV[5:0] | | | | | |
| | | W | | | | | | | | |
| 0x0035 | CPMUREFDIV | R | REFFRQ[1:0] | | 0 | 0 | REFDIV[3:0] | | | |
| | | W | | | | | | | | |
| 0x0036 | CPMUPOSTDIV | R | 0 | 0 | 0 | POSTDIV[4:0] | | | | |
| | | W | | | | | | | | |
| 0x0037 | CPMUFLG | R | RTIF | PORF | LVRF | LOCKIF | LOCK | ILAF | OSCIF | UPOSC |
| | | W | | | | | | | | |