**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 | HCS12X |
| Core Size | 16-Bit |
| Speed | 40MHz |
| Connectivity | CANbus, SCI, SPI |
| Peripherals | LVD, POR, PWM, WDT |
| Number of I/O | 59 |
| Program Memory Size | 128KB (128K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| 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 | 80-QFP |
| Supplier Device Package | 80-QFP (14x14) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12xs128j1maa |

### 1.2.4.3 VDD, VSS2, VSS3 — Core Power Pins

The voltage supply of nominally 1.8 V is derived from the internal voltage regulator. The return current path is through the VSS2 and VSS3 pins. No static external loading of these pins is permitted.

### 1.2.4.4 VDDF, VSS1 — NVM Power Pins

The voltage supply of nominally 2.8 V is derived from the internal voltage regulator. The return current path is through the VSS1 pin. No static external loading of these pins is permitted.

### 1.2.4.5 VDDA, VSSA — Power Supply Pins for ATD and Voltage Regulator

These are the power supply and ground input pins for the analog-to-digital converters and the voltage regulator.

### 1.2.4.6 VRH, VRL — ATD Reference Voltage Input Pins

$V_{RH}$ and $V_{RL}$ are the reference voltage input pins for the analog-to-digital converter.

### 1.2.4.7 VDDPLL, VSSPLL — Power Supply Pins for PLL

These pins provide operating voltage and ground for the oscillator and the phased-locked loop. The voltage supply of nominally 1.8 V is derived from the internal voltage regulator. This allows the supply voltage to the oscillator and PLL to be bypassed independently. This voltage is generated by the internal voltage regulator. No static external loading of these pins is permitted.

**Table 1-7. Power and Ground Connection Summary**

| Mnemonic | Nominal Voltage | Description |
|---|---|---|
| VDDR | 5.0 V | External power supply to internal voltage regulator |
| VDDX[2:1] | 5.0 V | External power and ground, supply to pin drivers |
| VSSX[2:1] | 0 V | |
| VDDA | 5.0 V | Operating voltage and ground for the analog-to-digital converters and the reference for the internal voltage regulator, allows the supply voltage to the A/D to be bypassed independently. |
| VSSA | 0 V | |
| VRL | 0 V | Reference voltages for the analog-to-digital converter. |
| VRH | 5.0 V | |
| VDD | 1.8 V | Internal power and ground generated by internal regulator for the internal core. |
| VSS1, VSS2, VSS3 | 0 V | |
| VDDF | 2.8 V | Internal power and ground generated by internal regulator for the internal NVM. |

**Table 2-9. DDRE Register Field Descriptions**

| Field | Description |
|---|---|
| 7-2<br>DDRE | **Port E Data Direction**—<br>This bit determines whether the associated pin is an input or output.<br><br>1 Associated pin configured as output<br>0 Associated pin configured as input |

## 2.3.10 Ports ABEK, BKGD pin Pull-up Control Register (PUCR)

Address 0x000C (PRR) Access: User read/write[1]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | PUPKE | BKPUE | 0 | PUPEE | 0 | 0 | PUPBE | PUPAE |
| W | | | | | | | | |
| Reset | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 2-8. Ports ABEK, BKGD pin Pull-up Control Register (PUCR)**

[1] Read: Anytime in single-chip modes
Write: Anytime, except BKPUE which is writable in Special Single-Chip Mode only

**Table 2-10. PUCR Register Field Descriptions**

| Field | Description |
|---|---|
| 7<br>PUPKE | **Port K Pull-up Enable**—Enable pull-up devices on all port input pins<br>This bit configures whether a pull-up device is activated on all associated port input pins. If a pin is used as output this bit has no effect.<br><br>1 Pull-up device enabled<br>0 Pull-up device disabled |
| 6<br>BKPUE | **BKGD pin pull-up Enable**—Enable pull-up device on pin<br>This bit configures whether a pull-up device is activated, if the pin is used as input. If a pin is used as output this bit has no effect.<br><br>1 Pull-up device enabled<br>0 Pull-up device disabled |
| 4<br>PUPEE | **Port E Pull-up Enable**—Enable pull-up devices on all port input pins except pins 5 and 6<br>This bit configures whether a pull-up device is activated on all associated port input pins. If a pin is used as output this bit has no effect.<br>Pins 5 and 6 have pull-down devices enabled only during reset. This bit has no effect on these pins.<br><br>1 Pull-up device enabled<br>0 Pull-up device disabled |

## 2.3.71 Port AD0 Reduced Drive Register 1 (RDR1AD0)

Address 0x0275 Access: User read/write[1]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | RDR1AD07 | RDR1AD06 | RDR1AD05 | RDR1AD04 | RDR1AD03 | RDR1AD02 | RDR1AD01 | RDR1AD00 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 2-69. Port AD0 Reduced Drive Register 1 (RDR1AD0)**

[1] Read: Anytime
Write: Anytime

**Table 2-68. RDR1AD0 Register Field Descriptions**

| Field | Description |
|---|---|
| 7-0<br>RDR1AD0 | **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 |

## 2.3.72 Port AD0 Pull Up Enable Register 0 (PER0AD0)

Address 0x0276 Access: User read/write[1]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | PER0AD07 | PER0AD06 | PER0AD05 | PER0AD04 | PER0AD03 | PER0AD02 | PER0AD01 | PER0AD00 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 2-70. Port AD0 Pull Device Up Register 0 (PER0AD0)**

[1] Read: Anytime
Write: Anytime

**Table 2-69. PER0AD0 Register Field Descriptions**

| Field | Description |
|---|---|
| 7-0<br>PER0AD0 | **Port AD0 pull device enable**—Enable pull-up device on input pin<br>This bit controls whether a pull device on the associated port input pin is active. If a pin is used as output this bit has no effect. The polarity is selected by the related polarity select register bit.<br><br>1 Pull device enabled<br>0 Pull device disabled |

16-bit misaligned reads and writes are generally not allowed. If attempted by BDM hardware command, the BDM will ignore the least significant bit of the address and will assume an even address from the remaining bits.

For devices with external bus:

The following cycle count information is only valid when the external wait function is not used (see wait bit of EBI sub-block). During an external wait the BDM can not steal a cycle. Hence be careful with the external wait function if the BDM serial interface is much faster than the bus, because of the BDM soft-reset after time-out (see Section 5.4.11, “Serial Communication Time Out”).

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 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. This includes the potential of extra cycles when the access is external and stretched (+1 to maximum +7 cycles) or to registers of the PRU (port replacement unit) in emulation modes (if modes available). 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.

**NOTE**

This timing has increased from previous BDM modules due to the new capability in which the BDM serial interface can potentially run faster than the bus. On previous BDM modules this extra time could be hidden within the serial time.

For 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 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 or the external wait function is used, 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 5-7 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]

#### 6.4.5.3.1 Information Byte Organization

The format of the control information byte is dependent upon the active trace mode as described below. In Normal, Loop1, or Pure PC modes tracing of CPU12X activity, CINF is used to store control information. In Detail Mode, CXINF contains the control information.

### CPU12X Information Byte

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| CSD | CVA | 0 | CDV | 0 | 0 | 0 | 0 |

**Figure 6-23. CPU12X Information Byte CINF**

**Table 6-41. CINF Field Descriptions**

| Field | Description |
|---|---|
| 7<br>CSD | **Source Destination Indicator** — This bit indicates if the corresponding stored address is a source or destination address. This is only used in Normal and Loop1 mode tracing.<br>0 Source address<br>1 Destination address |
| 6<br>CVA | **Vector Indicator** — This bit indicates if the corresponding stored address is a vector address. Vector addresses are destination addresses, thus if CVA is set, then the corresponding CSD is also set. This is only used in Normal and Loop1 mode tracing. This bit has no meaning in Pure PC mode.<br>0 Indexed jump destination address<br>1 Vector destination address |
| 4<br>CDV | **Data Invalid Indicator** — This bit indicates if the trace buffer entry is invalid. It is only used when tracing from both sources in Normal, Loop1 and Pure PC modes, to indicate that the CPU12X trace buffer entry is valid.<br>0 Trace buffer entry is invalid<br>1 Trace buffer entry is valid |

### CXINF Information Byte

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| | CSZ | CRW | | | | | |

**Figure 6-24. Information Byte CXINF**

This describes the format of the information byte used only when tracing in Detail Mode. When tracing from the CPU12X in Detail Mode, information is stored to the trace buffer on all cycles except opcode fetch and free cycles. In this case the CSZ and CRW bits indicate the type of access being made by the CPU12X.

**Table 6-42. CXINF Field Descriptions**

| Field | Description |
|---|---|
| 6<br>CSZ | **Access Type Indicator** — This bit indicates if the access was a byte or word size access.This bit only contains valid information when tracing CPU12X activity in Detail Mode.<br>0 Word Access<br>1 Byte Access |

# Chapter 9
# Pierce Oscillator (S12XOSCLCPV2)

**Table 9-1. Revision History**

| Revision Number | Revision Date | Sections Affected | Description of Changes |
|---|---|---|---|
| V01.05 | 19 Jul 2006 | | - All xclks info was removed |
| V02.00 | 04 Aug 2006 | | - Incremented revision to match the design system spec revision |

## 9.1 Introduction

The Pierce oscillator (XOSC) module provides a robust, low-noise and low-power clock source. The module will be operated from the $V_{DDPLL}$ supply rail (1.8 V nominal) and require the minimum number of external components. It is designed for optimal start-up margin with typical crystal oscillators.

### 9.1.1 Features

The XOSC will contain circuitry to dynamically control current gain in the output amplitude. This ensures a signal with low harmonic distortion, low power and good noise immunity.

- High noise immunity due to input hysteresis
- Low RF emissions with peak-to-peak swing limited dynamically
- Transconductance (gm) sized for optimum start-up margin for typical oscillators
- Dynamic gain control eliminates the need for external current limiting resistor
- Integrated resistor eliminates the need for external bias resistor in loop controlled Pierce mode.
- Low power consumption:
  - Operates from 1.8 V (nominal) supply
  - Amplitude control limits power
- Clock monitor

### 9.1.2 Modes of Operation

Two modes of operation exist:

1. Loop controlled Pierce (LCP) oscillator
2. External square wave mode featuring also full swing Pierce (FSP) without internal bias resistor

The oscillator mode selection is described in the Device Overview section, subsection Oscillator Configuration.

### 11.4.5.5 MSCAN Sleep Mode

The CPU can request the MSCAN to enter this low power mode by asserting the SLPRQ bit in the CANCTL0 register. The time when the MSCAN enters sleep mode depends on a fixed synchronization delay and its current activity:

- If there are one or more message buffers scheduled for transmission (TXEx = 0), the MSCAN will continue to transmit until all transmit message buffers are empty (TXEx = 1, transmitted successfully or aborted) and then goes into sleep mode.
- If the MSCAN is receiving, it continues to receive and goes into sleep mode as soon as the CAN bus next becomes idle.
- If the MSCAN is neither transmitting nor receiving, it immediately goes into sleep mode.

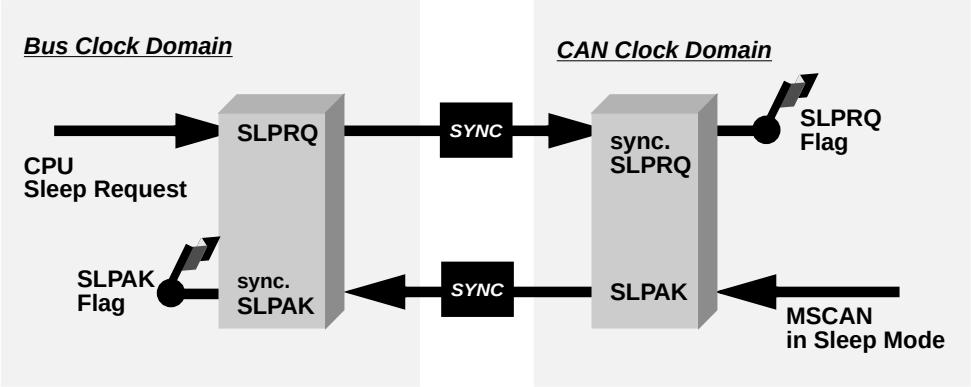


**Figure 11-46. Sleep Request / Acknowledge Cycle**

**NOTE**

The application software must avoid setting up a transmission (by clearing one or more TXEx flag(s)) and immediately request sleep mode (by setting SLPRQ). Whether the MSCAN starts transmitting or goes into sleep mode directly depends on the exact sequence of operations.

If sleep mode is active, the SLPRQ and SLPAK bits are set (Figure 11-46). The application software must use SLPAK as a handshake indication for the request (SLPRQ) to go into sleep mode.

When in sleep mode (SLPRQ = 1 and SLPAK = 1), the MSCAN stops its internal clocks. However, clocks that allow register accesses from the CPU side continue to run.

If the MSCAN is in bus-off state, it stops counting the 128 occurrences of 11 consecutive recessive bits due to the stopped clocks. TXCAN remains in a recessive state. If RXF = 1, the message can be read and RXF can be cleared. Shifting a new message into the foreground buffer of the receiver FIFO (RxFG) does not take place while in sleep mode.

It is possible to access the transmit buffers and to clear the associated TXE flags. No message abort takes place while in sleep mode.

#### 11.4.5.7 Disabled Mode

The MSCAN is in disabled mode out of reset (CANE=0). All module clocks are stopped for power saving, however the register map can still be accessed as specified.

#### 11.4.5.8 Programmable Wake-Up Function

The MSCAN can be programmed to wake up from sleep or power down mode as soon as CAN bus activity is detected (see control bit WUPE in MSCAN Control Register 0 (CANCTL0). The sensitivity to existing CAN bus action can be modified by applying a low-pass filter function to the RXCAN input line (see control bit WUPM in Section 11.3.2.2, "MSCAN Control Register 1 (CANCTL1)").

This feature can be used to protect the MSCAN from wake-up due to short glitches on the CAN bus lines. Such glitches can result from—for example—electromagnetic interference within noisy environments.

### 11.4.6 Reset Initialization

The reset state of each individual bit is listed in Section 11.3.2, "Register Descriptions," which details all the registers and their bit-fields.

### 11.4.7 Interrupts

This section describes all interrupts originated by the MSCAN. It documents the enable bits and generated flags. Each interrupt is listed and described separately.

#### 11.4.7.1 Description of Interrupt Operation

The MSCAN supports four interrupt vectors (see Table 11-39), any of which can be individually masked (for details see Section 11.3.2.6, "MSCAN Receiver Interrupt Enable Register (CANRIER)" to Section 11.3.2.8, "MSCAN Transmitter Interrupt Enable Register (CANTIER)").

Refer to the device overview section to determine the dedicated interrupt vector addresses.

**Table 11-39. Interrupt Vectors**

| Interrupt Source | CCR Mask | Local Enable |
|---|---|---|
| Wake-Up Interrupt (WUPIF) | I bit | CANRIER (WUPIE) |
| Error Interrupts Interrupt (CSCIF, OVRIF) | I bit | CANRIER (CSCIE, OVRIE) |
| Receive Interrupt (RXF) | I bit | CANRIER (RXFIE) |
| Transmit Interrupts (TXE[2:0]) | I bit | CANTIER (TXEIE[2:0]) |

#### 11.4.7.2 Transmit Interrupt

At least one of the three transmit buffers is empty (not scheduled) and can be loaded to schedule a message for transmission. The TXEx flag of the empty message buffer is set.

### 14.3.2.3 SCI Alternative Status Register 1 (SCIASR1)

Module Base + 0x0000

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | RXEDGIF | 0 | 0 | 0 | 0 | BERRV | BERRIF | BKDIF |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 14-6. SCI Alternative Status Register 1 (SCIASR1)**

Read: Anytime, if AMAP = 1

Write: Anytime, if AMAP = 1

**Table 14-6. SCIASR1 Field Descriptions**

| Field | Description |
|---|---|
| 7<br>RXEDGIF | **Receive Input Active Edge Interrupt Flag** — RXEDGIF is asserted, if an active edge (falling if RXPOL = 0, rising if RXPOL = 1) on the RXD input occurs. RXEDGIF bit is cleared by writing a “1” to it.<br>0 No active receive on the receive input has occurred<br>1 An active edge on the receive input has occurred |
| 2<br>BERRV | **Bit Error Value** — BERRV reflects the state of the RXD input when the bit error detect circuitry is enabled and a mismatch to the expected value happened. The value is only meaningful, if BERRIF = 1.<br>0 A low input was sampled, when a high was expected<br>1 A high input reassembled, when a low was expected |
| 1<br>BERRIF | **Bit Error Interrupt Flag** — BERRIF is asserted, when the bit error detect circuitry is enabled and if the value sampled at the RXD input does not match the transmitted value. If the BERRIE interrupt enable bit is set an interrupt will be generated. The BERRIF bit is cleared by writing a “1” to it.<br>0 No mismatch detected<br>1 A mismatch has occurred |
| 0<br>BKDIF | **Break Detect Interrupt Flag** — BKDIF is asserted, if the break detect circuitry is enabled and a break signal is received. If the BKDIE interrupt enable bit is set an interrupt will be generated. The BKDIF bit is cleared by writing a “1” to it.<br>0 No break signal was received<br>1 A break signal was received |

### 14.3.2.5 SCI Alternative Control Register 2 (SCIACR2)

Module Base + 0x0002

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | 0 | BERRM1 | BERRM0 | BKDFE |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 14-8. SCI Alternative Control Register 2 (SCIACR2)**

Read: Anytime, if AMAP = 1

Write: Anytime, if AMAP = 1

**Table 14-8. SCIACR2 Field Descriptions**

| Field | Description |
|---|---|
| 2:1<br>BERRM[1:0] | **Bit Error Mode** — Those two bits determines the functionality of the bit error detect feature. See Table 14-9. |
| 0<br>BKDFE | **Break Detect Feature Enable** — BKDFE enables the break detect circuitry.<br>0 Break detect circuit disabled<br>1 Break detect circuit enabled |

**Table 14-9. Bit Error Mode Coding**

| BERRM1 | BERRM0 | Function |
|---|---|---|
| 0 | 0 | Bit error detect circuit is disabled |
| 0 | 1 | Receive input sampling occurs during the 9th time tick of a transmitted bit (refer to Figure 14-19) |
| 1 | 0 | Receive input sampling occurs during the 13th time tick of a transmitted bit (refer to Figure 14-19) |
| 1 | 1 | Reserved |

# Chapter 15
# Serial Peripheral Interface (S12SPIV5)

**Table 15-1. Revision History**

| Revision Number | Revision Date | Sections Affected | Description of Changes |
|---|---|---|---|
| V05.00 | 24 Mar 2005 | 15.3.2/15-439 | - Added 16-bit transfer width feature. |

## 15.1 Introduction

The SPI module allows a duplex, synchronous, serial communication between the MCU and peripheral devices. Software can poll the SPI status flags or the SPI operation can be interrupt driven.

### 15.1.1 Glossary of Terms

| | |
|---|---|
| SPI | Serial Peripheral Interface |
| $\overline{SS}$ | Slave Select |
| SCK | Serial Clock |
| MOSI | Master Output, Slave Input |
| MISO | Master Input, Slave Output |
| MOMI | Master Output, Master Input |
| SISO | Slave Input, Slave Output |

### 15.1.2 Features

The SPI includes these distinctive features:

- Master mode and slave mode
- Selectable 8 or 16-bit transfer width
- Bidirectional mode
- Slave select output
- Mode fault error flag with CPU interrupt capability
- Double-buffered data register
- Serial clock with programmable polarity and phase
- Control of SPI operation during wait mode

### 15.1.3 Modes of Operation

The SPI functions in three modes: run, wait, and stop.

**Table 15-4. SPICR2 Field Descriptions**

| Field | Description |
|---|---|
| 6<br>XFRW | **Transfer Width —** This bit is used for selecting the data transfer width. If 8-bit transfer width is selected, SPIDRL becomes the dedicated data register and SPIDRH is unused. If 16-bit transfer width is selected, SPIDRH and SPIDRL form a 16-bit data register. Please refer to Section 15.3.2.4, “SPI Status Register (SPISR) for information about transmit/receive data handling and the interrupt flag clearing mechanism. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state.<br>0 8-bit Transfer Width (n = 8)[1]<br>1 16-bit Transfer Width (n = 16)[1] |
| 4<br>MODFEN | **Mode Fault Enable Bit** — This bit allows the MODF failure to be detected. If the SPI is in master mode and MODFEN is cleared, then the $\overline{SS}$ port pin is not used by the SPI. In slave mode, the $\overline{SS}$ is available only as an input regardless of the value of MODFEN. For an overview on the impact of the MODFEN bit on the $\overline{SS}$ port pin configuration, refer to Table 15-3. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state.<br>0 $\overline{SS}$ port pin is not used by the SPI.<br>1 $\overline{SS}$ port pin with MODF feature. |
| 3<br>BIDIROE | **Output Enable in the Bidirectional Mode of Operation** — This bit controls the MOSI and MISO output buffer of the SPI, when in bidirectional mode of operation (SPC0 is set). In master mode, this bit controls the output buffer of the MOSI port, in slave mode it controls the output buffer of the MISO port. In master mode, with SPC0 set, a change of this bit will abort a transmission in progress and force the SPI into idle state.<br>0 Output buffer disabled.<br>1 Output buffer enabled. |
| 1<br>SPISWAI | **SPI Stop in Wait Mode Bit** — This bit is used for power conservation while in wait mode.<br>0 SPI clock operates normally in wait mode.<br>1 Stop SPI clock generation when in wait mode. |
| 0<br>SPC0 | **Serial Pin Control Bit 0** — This bit enables bidirectional pin configurations as shown in Table 15-5. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. |

1 n is used later in this document as a placeholder for the selected transfer width.

**Table 15-5. Bidirectional Pin Configurations**

| Pin Mode | SPC0 | BIDIROE | MISO | MOSI |
|---|---|---|---|---|
| **Master Mode of Operation** | | | | |
| Normal | 0 | X | Master In | Master Out |
| Bidirectional | 1 | 0 | MISO not used by SPI | Master In |
| | | 1 | | Master I/O |
| **Slave Mode of Operation** | | | | |
| Normal | 0 | X | Slave Out | Slave In |
| Bidirectional | 1 | 0 | Slave In | MOSI not used by SPI |
| | | 1 | Slave I/O | |

CAUTION

The FCLKDIV register should never be written while a Flash command is executing (CCIF=0). The FCLKDIV register is writable during the Flash reset sequence even though CCIF is clear.

### 20.5.1 Unsecuring the MCU using Backdoor Key Access

The MCU may be unsecured by using the backdoor key access feature which requires knowledge of the contents of the backdoor keys (four 16-bit words programmed at addresses 0x7F_FF00–0x7F_FF07). If the KEYEN[1:0] bits are in the enabled state (see Section 20.3.2.2), the Verify Backdoor Access Key command (see Section 20.4.2.11) allows the user to present four prospective keys for comparison to the keys stored in the Flash memory via the Memory Controller. If the keys presented in the Verify Backdoor Access Key command match the backdoor keys stored in the Flash memory, the SEC bits in the FSEC register (see Table 20-10) will be changed to unsecure the MCU. Key values of 0x0000 and 0xFFFF are not permitted as backdoor keys. While the Verify Backdoor Access Key command is active, P-Flash block 0 will not be available for read access and will return invalid data.

The user code stored in the P-Flash memory must have a method of receiving the backdoor keys from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN[1:0] bits are in the enabled state (see Section 20.3.2.2), the MCU can be unsecured by the backdoor key access sequence described below:

1. Follow the command sequence for the Verify Backdoor Access Key command as explained in Section 20.4.2.11
2. If the Verify Backdoor Access Key command is successful, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 10

The Verify Backdoor Access Key command is monitored by the Memory Controller and an illegal key will prohibit future use of the Verify Backdoor Access Key command. A reset of the MCU is the only method to re-enable the Verify Backdoor Access Key command.

After the backdoor keys have been correctly matched, the MCU will be unsecured. After the MCU is unsecured, the sector containing the Flash security byte can be erased and the Flash security byte can be reprogrammed to the unsecure state, if desired.

In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0x7F_FF00–0x7F_FF07 in the Flash configuration field.

The security as defined in the Flash security byte (0x7F_FF0F) is not changed by using the Verify Backdoor Access Key command sequence. The backdoor keys stored in addresses 0x7F_FF00–0x7F_FF07 are unaffected by the Verify Backdoor Access Key command sequence. After the next reset of the MCU, the security state of the Flash module is determined by the Flash security byte (0x7F_FF0F). The Verify Backdoor Access Key command sequence has no effect on the program and erase protections defined in the Flash protection register, FPROT.

### 20.5.2 Unsecuring the MCU in Special Single Chip Mode using BDM

The MCU can be unsecured in special single chip mode by erasing the P-Flash and D-Flash memory by one of the following methods:

- Reset the MCU into special single chip mode, delay while the erase test is performed by the BDM, send BDM commands to disable protection in the P-Flash and D-Flash memory, and execute the Erase All Blocks command write sequence to erase the P-Flash and D-Flash memory.

4MHz loop controlled Pierce oscillator. Production test parameters are tested with a 4MHz square wave oscillator.

Table A-10 shows the configuration of the peripherals for maximum run current

**Table A-10. Module Configurations for Maximum Run Supply (VDDR+VDDA) Current $V_{DD35}$=5.5V**

| Peripheral | Configuration |
|---|---|
| S12XCPU | 420 cycle loop: 384 DBNE cycles plus subroutine entry to stimulate stacking (RAM access) |
| MSCAN | Configured to loop-back mode using a bit rate of 1Mbit/s |
| SPI | Configured to master mode, continuously transmit data (0x55 or 0xAA) at 4Mbit/s |
| SCI | Configured into loop mode, continuously transmit data (0x55) at speed of 57600 baud |
| PWM | Configured to toggle its pins at the rate of 40kHz |
| TIM | The peripheral shall be configured in output compare mode. Pulse accumulator and modulus counter enabled. |
| ATD | The peripheral is configured to operate at its maximum specified frequency and to continuously convert voltages on all input channels in sequence. |
| Overhead | VREG supplying 1.8V from a 5V input voltage, PLL on |

### A.1.10.3 Stop Current Conditions

Unbonded ports must be correctly initialized to prevent current consumption due to floating inputs. Typical Stop current is measured with $V_{DD35}$=5V, maximum Stop current is measured with $V_{DD35}$=5.5V. Pseudo Stop currents are measured with the oscillator configured for 4MHz LCP mode. Production test parameters are tested with a 4MHz square wave oscillator.

### A.1.10.4 Measurement Results

**Table A-11. Module Run Supply Currents**

| Conditions are shown in Table A-9 at ambient temperature unless otherwise noted | | | | | | |
|---|---|---|---|---|---|---|
| **Num** | **C** | **Rating** | **Min** | **Typ** | **Max** | **Unit** |
| 1 | T | S12XCPU | — | 1.1 | — | mA |
| 2 | T | MSCAN | — | 0.5 | — | |
| 3 | T | SPI | — | 0.4 | — | |
| 4 | T | SCI | — | 0.6 | — | |
| 5 | T | PWM | — | 0.9 | — | |
| 6 | T | TIM | — | 0.3 | — | |
| 7 | T | ATD | — | 1.7 | — | |
| 8 | T | Overhead | — | 13.6 | — | |

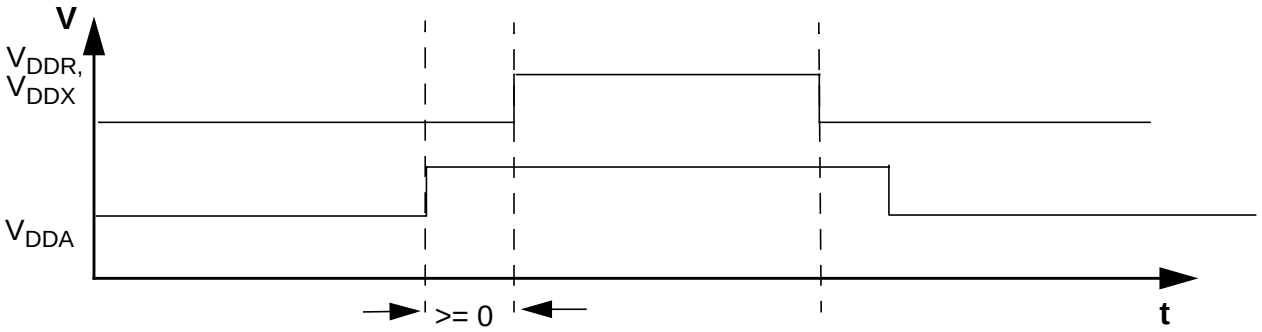


**Figure A-3. S12XS family Power Sequencing**

During power sequencing $V_{DDA}$ can be powered up before $V_{DDR}$, $V_{DDX}$.
$V_{DDR}$ and $V_{DDX}$ must be powered up together adhering to the operating conditions differential.
$V_{RH}$ power up must follow $V_{DDA}$ to avoid current injection.

### C.1.1 112-Pin LQFP Recommended PCB Layout

**Figure C-1. 112-Pin LQFP Recommended PCB Layout (Loop Controlled Pierce Oscillator)**

## 0x0120–0x012F Interrupt Module (S12XINT) Map

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0120 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0121 | IVBR | R | IVB_ADDR[7:0] | | | | | | | |
| | | W | | | | | | | | |
| 0x0122 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0123 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0124 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0125 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0126 | INT_XGPRIO | R | 0 | 0 | 0 | 0 | 0 | XILVL[2:0] | | |
| | | W | | | | | | | | |
| 0x0127 | INT_CFADDR | R | INT_CFADDR[7:4] | | | | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0128 | INT_CFDATA0 | R | RQST | 0 | 0 | 0 | 0 | PRIOLVL[2:0] | | |
| | | W | | | | | | | | |
| 0x0129 | INT_CFDATA1 | R | RQST | 0 | 0 | 0 | 0 | PRIOLVL[2:0] | | |
| | | W | | | | | | | | |
| 0x012A | INT_CFDATA2 | R | RQST | 0 | 0 | 0 | 0 | PRIOLVL[2:0] | | |
| | | W | | | | | | | | |
| 0x012B | INT_CFDATA3 | R | RQST | 0 | 0 | 0 | 0 | PRIOLVL[2:0] | | |
| | | W | | | | | | | | |
| 0x012C | INT_CFDATA4 | R | RQST | 0 | 0 | 0 | 0 | PRIOLVL[2:0] | | |
| | | W | | | | | | | | |
| 0x012D | INT_CFDATA5 | R | RQST | 0 | 0 | 0 | 0 | PRIOLVL[2:0] | | |
| | | W | | | | | | | | |
| 0x012E | INT_CFDATA6 | R | RQST | 0 | 0 | 0 | 0 | PRIOLVL[2:0] | | |
| | | W | | | | | | | | |
| 0x012F | INT_CFDATA7 | R | RQST | 0 | 0 | 0 | 0 | PRIOLVL[2:0] | | |
| | | W | | | | | | | | |

## 0x00130–0x013F Reserved Register Space

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0130-0x013F | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

## 0x0140–0x017F MSCAN (CAN0) Map

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0140 | CAN0CTL0 | R | RXFRM | RXACT | CSWAI | SYNCH | TIME | WUPE | SLPRQ | INITRQ |
| | | W | | | | | | | | |
| 0x0141 | CAN0CTL1 | R | CANE | CLKSRC | LOOPB | LISTEN | BORM | WUPM | SLPAK | INITAK |
| | | 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0260 | PTH | R | PTH7 | PTH6 | PTH5 | PTH4 | PTH3 | PTH2 | PTH1 | PTH0 |
| | | W | | | | | | | | |
| 0x0261 | PTIH | R | PTIH7 | PTIH6 | PTIH5 | PTIH4 | PTIH3 | PTIH2 | PTIH1 | PTIH0 |
| | | W | | | | | | | | |
| 0x0262 | DDRH | R | DDRH7 | DDRH6 | DDRH5 | DDRH4 | DDRH3 | DDRH2 | DDRH1 | DDRH0 |
| | | W | | | | | | | | |
| 0x0263 | RDRH | R | RDRH7 | RDRH6 | RDRH5 | RDRH4 | RDRH3 | RDRH2 | RDRH1 | RDRH0 |
| | | W | | | | | | | | |
| 0x0264 | PERH | R | PERH7 | PERH6 | PERH5 | PERH4 | PERH3 | PERH2 | PERH1 | PERH0 |
| | | W | | | | | | | | |
| 0x0265 | PPSH | R | PPSH7 | PPSH6 | PPSH5 | PPSH4 | PPSH3 | PPSH2 | PPSH1 | PPSH0 |
| | | W | | | | | | | | |
| 0x0266 | PIEH | R | PIEH7 | PIEH6 | PIEH5 | PIEH4 | PIEH3 | PIEH2 | PIEH1 | PIEH0 |
| | | W | | | | | | | | |
| 0x0267 | PIFH | R | PIFH7 | PIFH6 | PIFH5 | PIFH4 | PIFH3 | PIFH2 | PIFH1 | PIFH0 |
| | | W | | | | | | | | |
| 0x0268 | PTJ | R | PTJ7 | PTJ6 | 0 | 0 | 0 | 0 | PTJ1 | PTJ0 |
| | | W | | | | | | | | |
| 0x0269 | PTIJ | R | PTIJ7 | PTIJ6 | 0 | 0 | 0 | 0 | PTIJ1 | PTIJ0 |
| | | W | | | | | | | | |
| 0x026A | DDRJ | R | DDRJ7 | DDRJ6 | 0 | 0 | 0 | 0 | DDRJ1 | DDRJ0 |
| | | W | | | | | | | | |
| 0x026B | RDRJ | R | RDRJ7 | RDRJ6 | 0 | 0 | 0 | 0 | RDRJ1 | RDRJ0 |
| | | W | | | | | | | | |
| 0x026C | PERJ | R | PERJ7 | PERJ6 | 0 | 0 | 0 | 0 | PERJ1 | PERJ0 |
| | | W | | | | | | | | |
| 0x026D | PPSJ | R | PPSJ7 | PPSJ6 | 0 | 0 | 0 | 0 | PPSJ1 | PPSJ0 |
| | | W | | | | | | | | |
| 0x026E | PIEJ | R | PIEJ7 | PIEJ6 | 0 | 0 | 0 | 0 | PIEJ1 | PIEJ0 |
| | | W | | | | | | | | |
| 0x026f | PIFJ | R | PIFJ7 | PIFJ6 | 0 | 0 | 0 | 0 | PIFJ1 | PIFJ0 |
| | | W | | | | | | | | |
| 0x0270 | PT0AD0 | R | PT0AD07 | PT0AD06 | PT0AD05 | PT0AD04 | PT0AD03 | PT0AD02 | PT0AD01 | PT0AD00 |
| | | W | | | | | | | | |
| 0x0271 | PT1AD0 | R | PT1AD07 | PT1AD06 | PT1AD05 | PT1AD04 | PT1AD03 | PT1AD02 | PT1AD01 | PT1AD00 |
| | | W | | | | | | | | |
| 0x0272 | DDR0AD0 | R | DDR0AD07 | DDR0AD06 | DDR0AD05 | DDR0AD04 | DDR0AD03 | DDR0AD02 | DDR0AD01 | DDR0AD00 |
| | | W | | | | | | | | |
| 0x0273 | DDR1AD0 | R | DDR1AD07 | DDR1AD06 | DDR1AD05 | DDR1AD04 | DDR1AD03 | DDR1AD02 | DDR1AD01 | DDR1AD00 |
| | | W | | | | | | | | |
| 0x0274 | RDR0AD0 | R | RDR0AD07 | RDR0AD06 | RDR0AD05 | RDR0AD04 | RDR0AD03 | RDR0AD02 | RDR0AD01 | RDR0AD00 |
| | | W | | | | | | | | |
| 0x0275 | RDR1AD0 | R | RDR1AD07 | RDR1AD06 | RDR1AD05 | RDR1AD04 | RDR1AD03 | RDR1AD02 | RDR1AD01 | RDR1AD00 |
| | | W | | | | | | | | |