Document Number: MM912_634D1
Rev. 13.0, 6/2015

# Integrated S12 Based Relay Driver with LIN

**912_634**



**AE SUFFIX (PB-FREE)**
**98ASA00173D**
**48 PIN LQFP-EP**
**(7.0 X7.0 mm)**

**AP SUFFIX (PB-FREE)**
**98ASH00962A**
**48 PIN LQFP**
**(7.0 X7.0 mm)**

The MM912G634 (48 kB) and MM912H634 (64 kB) are integrated, single package solutions which integrate an HCS12 microcontroller with a SMARTMOS analog control IC. The Die to Die Interface (D2D) controlled analog die combines system base chip and application specific functions, including a LIN transceiver.

**Features**

- 16-Bit S12 CPU, 64/48 kByte P-FLASH
- 6.0 kByte RAM; 4/2 kByte D-FLASH
- Background debug (BDM) & debug module (DBG)
- Die to Die bus interface for transparent memory mapping
- On-chip oscillator & two independent watchdogs
- LIN 2.1 Physical layer interface with integrated SCI
- 10 digital MCU GPIOs shared with SPI (PA7…0, PE1…0)
- 10-Bit, 15 Channel - Analog to Digital converter (ADC)
- 16-Bit, 4 Channel - Timer module (TIM16B4C)
- 8-Bit, 2 Channel - Pulse width modulation module (PWM)
- Six high-voltage / Wake-up inputs (L5…0)
- Three low voltage GPIOs (PB2…0)
- Low power modes with cyclic sense & forced wake-up
- Current sense module with selectable gain
- Reverse battery protected voltage sense module
- Two protected low-side outputs to drive inductive loads
- Two protected high-side outputs
- Chip temperature sensor
- Hall sensor supply & integrated voltage regulator(s)

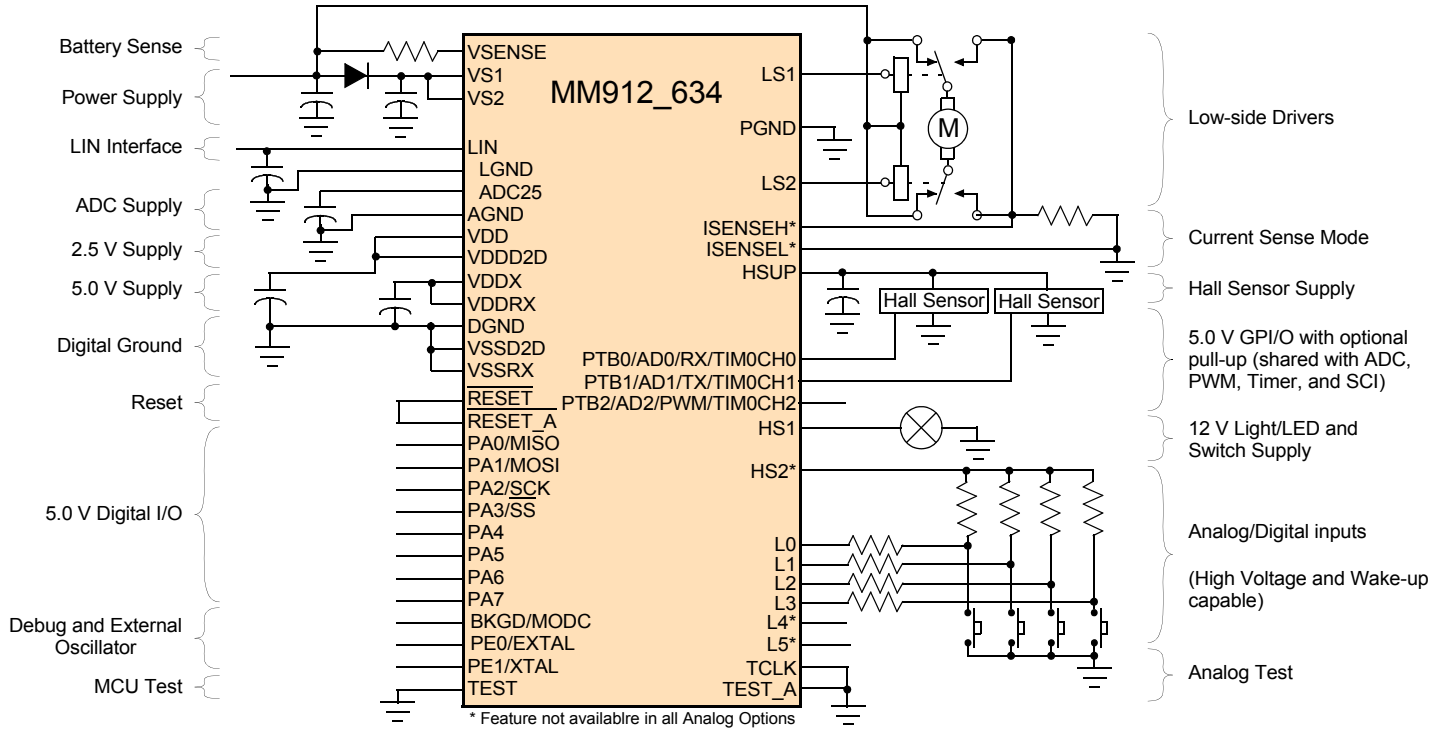



**Figure 1. Simplified Application Diagram**

© Freescale Semiconductor, Inc., 2010 - 2015. All rights reserved.

# 1 Ordering Information

**Table 1. Ordering Information**

| Device (Add an R2 suffix for Tape and Reel orders) | Temperature Range ($T_A$) | Package | Max. Bus Frequency in MHz ($f_{BUSMAX}$) | Flash (kB) | Data Flash (kB) | RAM (kB) | Analog Option[1] |
|---|---|---|---|---|---|---|---|
| MM912G634DM1AE | -40°C to 125°C | LQFP48-EP | 20 | 48[2] | 2[3] | 2[4] | A1 |
| MM912G634DV1AE | -40°C to 105°C | | | | | | |
| MM912G634DV2AP | | LQFP48 | 16 | | | | A2 |
| MM912H634DM1AE | -40°C to 125°C | LQFP48-EP | 20 | 64 | 4 | 6 | A1 |
| MM912H634DV1AE | -40°C to 105°C | | | | | | |

Note:

1. See Table 2.
2. The 48 kB Flash option (MM912**G**634) using the same S12I64 MCU with the tested FLASHSIZE reduced to 48 kB. This limits the usable Flash area to the first 48 kB (0x3_4000-0x3_FFFF).
3. The 48 kB Flash option (MM912**G**634) using the same S12I64 MCU with the tested Data - FLASHSIZE reduced to 2.0 kB. This limits the usable Data Flash area to the first 2.0 kB (0x0_4400-0x0_4BFF).
4. The 48 kB Flash option (MM912**G**634) using the same S12I64 MCU with the tested RAMSIZE reduced to 2.0 kB. This limits the usable RAM area to the first 2.0 kB (0x0_2800-0x0_2FFF).

**Table 2. Analog Options[5]**

| Feature | A1 | A2 |
|---|---|---|
| Battery Sense Module | YES | YES |
| Current Sense Module | YES | NO |
| 2nd High-side Output (HS2) | YES | YES |
| Wake-up Inputs (Lx) | L0…L5 | L0…L3 |
| Hall Supply Output (HSUP) | YES | YES |
| LIN Module | YES | YES |

Note:

5. This table only highlights the analog die differences between the derivatives. Features highlighted as “NO” or the Lx Inputs not mentioned are not available in the specific option and not bonded out and/or not tested. See Analog Die Options for detailed information.

# Table of Contents


1 Ordering Information . . . 2
2 Internal Block Diagram . . . 4
3 Pin Assignment . . . 5
3.1 MM912_634 Pin Description . . . 6
3.2 MCU Die Signal Properties . . . 9
4 Electrical Characteristics . . . 10
4.1 General . . . 10
4.2 Absolute Maximum Ratings . . . 10
4.3 Operating Conditions . . . 12
4.4 Supply Currents . . . 13
4.5 Static Electrical Characteristics . . . 14
4.6 Dynamic Electrical Characteristics . . . 22
4.7 Thermal Protection Characteristics . . . 36
4.8 ESD Protection and Latch-up Immunity . . . 36
4.9 Additional Test Information ISO7637-2 . . . 37
5 Functional Description and Application Information . . . 38
5.1 Introduction . . . 38
5.2 Device Register Maps . . . 38
5.3 MM912_634 - Analog Die Overview . . . 54
5.4 Modes of Operation . . . 57
5.5 Power Supply . . . 61
5.6 Die to Die Interface - Target . . . 64
5.7 Interrupts . . . 65
5.8 Resets . . . 68
5.9 Wake-up / Cyclic Sense . . . 70
5.10 Window Watchdog . . . 73
5.11 Hall Sensor Supply Output - HSUP . . . 76
5.12 High-side Drivers - HS . . . 76
5.13 Low-side Drivers - LSx . . . 78
5.14 PWM Control Module (PWM8B2C) . . . 83
5.15 LIN Physical Layer Interface - LIN . . . 97
5.16 Serial Communication Interface (S08SCIV4) . . . 99
5.17 High Voltage Inputs - Lx . . . 112
5.18 General Purpose I/O - PTB[0…2] . . . 113
5.19 Basic Timer Module - TIM (TIM16B4C) . . . 115
5.20 Analog Digital Converter - ADC . . . 129
5.21 Current Sense Module - ISENSE . . . 137
5.22 Temperature Sensor - TSENSE . . . 138
5.23 Supply Voltage Sense - VSENSE . . . 139
5.24 Internal Supply Voltage Sense - VS1SENSE . . . 139
5.25 Internal Bandgap Reference Voltage Sense - BANDGAP . . . 139
5.26 MM912_634 - Analog Die Trimming . . . 140
5.27 MM912_634 - MCU Die Overview . . . 144
5.28 Port Integration Module (S12IPIMV1) . . . 151
5.29 Memory Map Control (S12PMMCV1) . . . 161
5.30 Interrupt Module (S12SINTV1) . . . 173
5.31 Background Debug Module (S12SBDMV1) . . . 177
5.32 S12S Debug Module (S12SDBGV2) . . . 193
5.33 Security (S12X9SECV2) . . . 225
5.34 Impact on MCU modules . . . 226
5.35 Secure firmware Code Overview . . . 228
5.36 Initialization of a Virgin Device . . . 230
5.37 Impact of Security on Test . . . 231
5.38 S12 Clock, Reset and Power Management Unit (S12CPMU) . . . 231
5.39 Serial Peripheral Interface (S12SPIV5) . . . 268
5.40 64 KByte Flash Module (S12FTMRC64K1V1) . . . 287
5.41 Die-to-Die Initiator (D2DIV1) . . . 322
6 Packaging . . . 333
6.1 Package Dimensions . . . 333
7 Revision History . . . 338

# 2 Internal Block Diagram

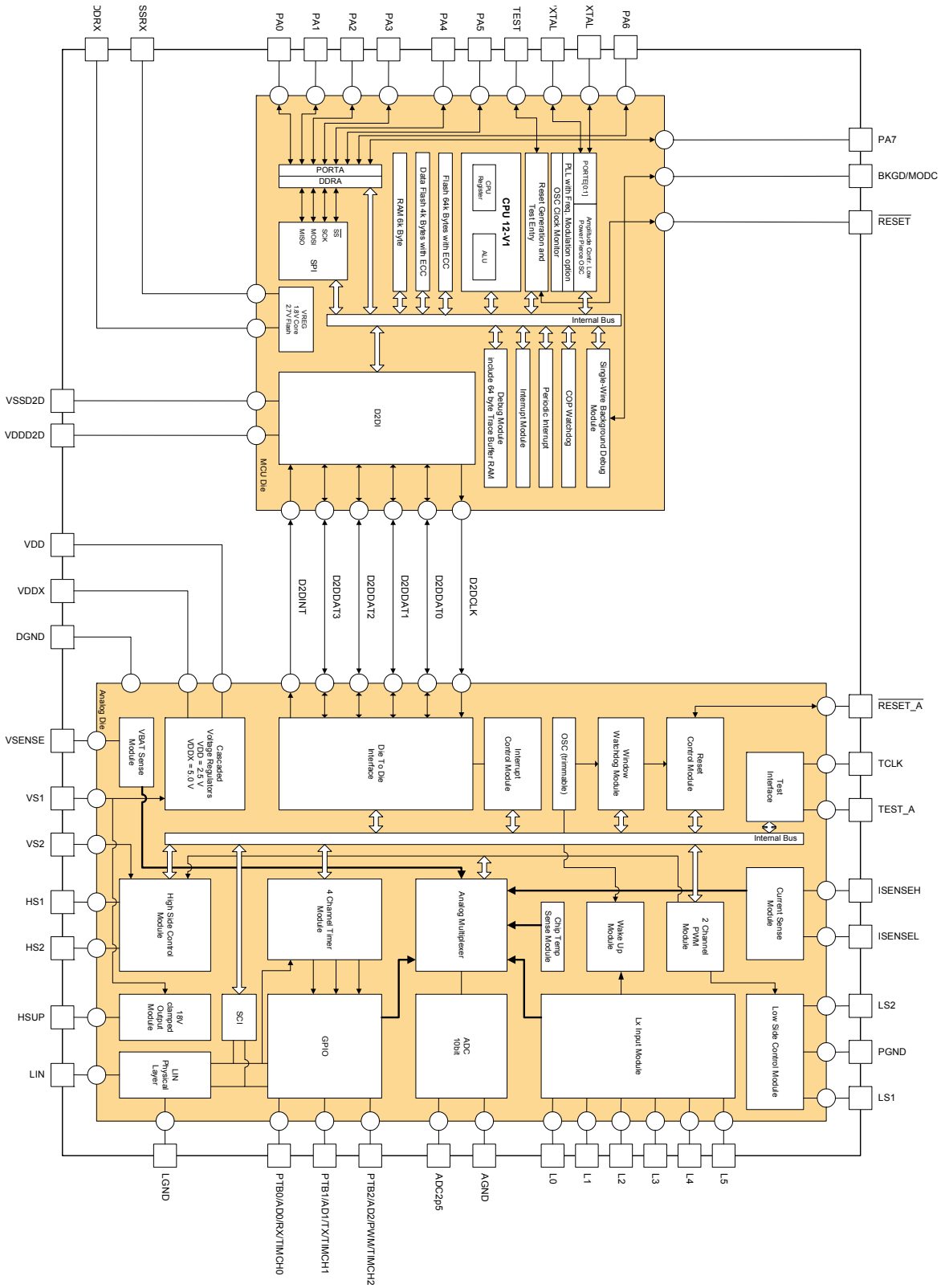



**Figure 2. MM912_634 Block Diagram**

# 3 Pin Assignment

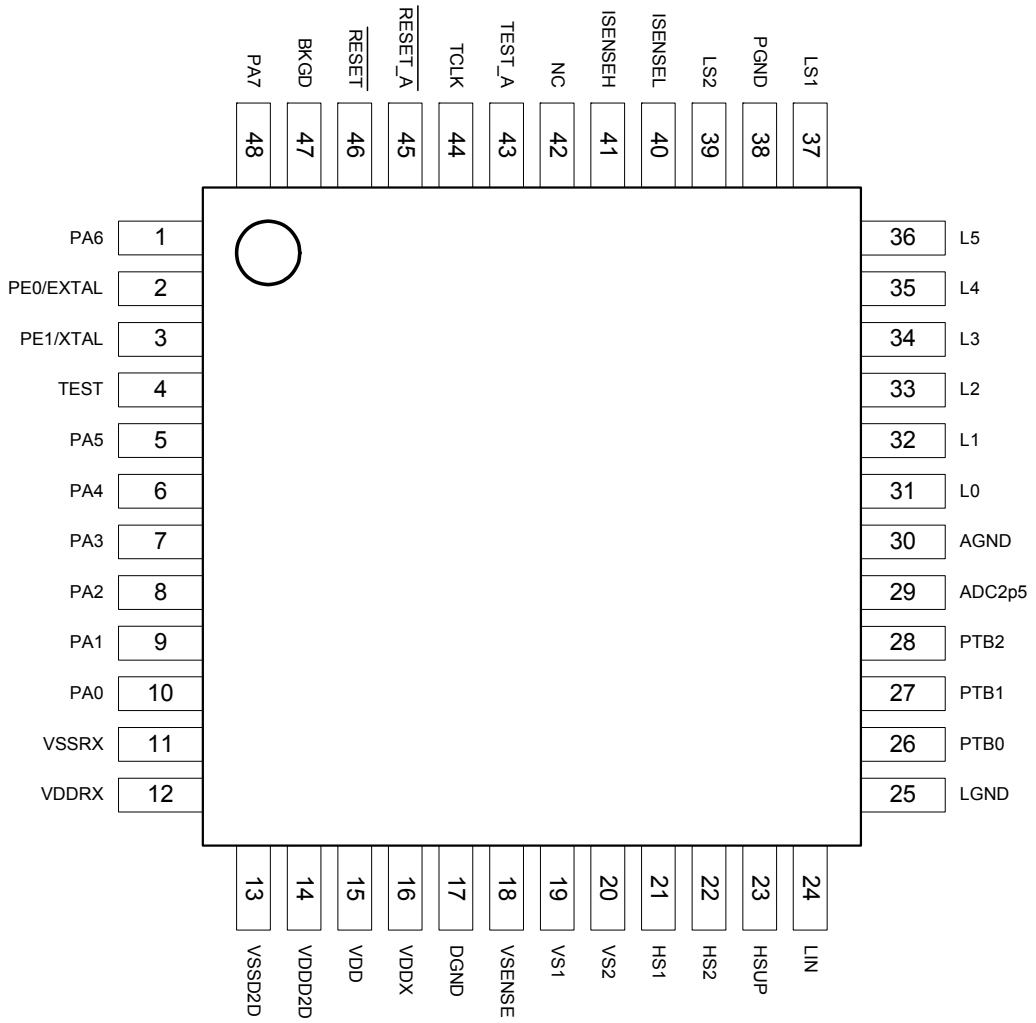



**Figure 3. MM912_634 Pin Out**

**NOTE**

The device exposed pad (package option AE only) is recommended to be connected to GND.

Not all pins are available for analog die option 2. See Analog Die Options for details.

## 3.1 MM912_634 Pin Description

The following table gives a brief description of all available pins on the MM912_634 package. Refer to the highlighted chapter for detailed information.

**Table 3. MM912_634 Pin Description**

| Pin # | Pin Name | Formal Name | Description |
|---|---|---|---|
| 1 | PA6 | MCU PA6 | General purpose port A input or output pin 6. See Section 5.28, “Port Integration Module (S12IPIMV1)" |
| 2 | PE0/EXTAL | MCU Oscillator | EXTAL is one of the optional crystal/resonator driver and external clock pins. On reset, all the device clocks are derived from the Internal Reference Clock and port PE may be used for general purpose I/O. See Section 5.38.2.2, “EXTAL and XTAL" and Section 5.28, “Port Integration Module (S12IPIMV1)". |
| 3 | PE1/XTAL | MCU Oscillator | XTAL is one of the optional crystal/resonator driver and external clock pins. On reset, all the device clocks are derived from the Internal Reference Clock and port PE may be used for general purpose I/O. See Section 5.38.2.2, “EXTAL and XTAL" and Section 5.28, “Port Integration Module (S12IPIMV1)". |
| 4 | TEST | MCU Test | This input only pin is reserved for test. This pin has a pull-down device. The TEST pin must be tied to EVSS in user mode. |
| 5 | PA5 | MCU PA5 | General purpose port A input or output pin 5. See Section 5.28, “Port Integration Module (S12IPIMV1)" |
| 6 | PA4 | MCU PA4 | General purpose port A input or output pin 4. See Section 5.28, “Port Integration Module (S12IPIMV1)". |
| 7 | PA3 | MCU PA3 / $\overline{SS}$ | General purpose port A input or output pin 3, shared with the $\overline{SS}$ signal of the integrated SPI Interface. See Section 5.28, “Port Integration Module (S12IPIMV1)". |
| 8 | PA2 | MCU PA2 / SCK | General purpose port A input or output pin 2, shared with the SCLK signal of the integrated SPI Interface. See Section 5.28, “Port Integration Module (S12IPIMV1)". |
| 9 | PA1 | MCU PA1 / MOSI | General purpose port A input or output pin 1, shared with the MOSI signal of the integrated SPI Interface. See Section 5.28, “Port Integration Module (S12IPIMV1)". |
| 10 | PA0 | MCU PA0 / MISO | General-purpose port A input or output pin 0, shared with the MISO signal of the integrated SPI Interface. See Section 5.28, “Port Integration Module (S12IPIMV1)". |
| 11 | VSSRX | MCU 5.0 V Ground | Ground for the MCU 5.0 V power supply. |
| 12 | VDDRX | MCU 5.0 V Supply | MCU 5.0 V - Core- and Flash Voltage Regulator supply. See Section 5.27, “MM912_634 - MCU Die Overview". |
| 13 | VSSD2D | MCU 2.5 V Ground | Ground for the MCU 2.5 V power supply. |
| 14 | VDDD2D | MCU 2.5 V Supply | MCU 2.5 V - MCU Die-to-Die Interface power supply. See Section 5.27, “MM912_634 - MCU Die Overview". |
| 15 | VDD | Voltage Regulator Output 2.5 V | +2.5 V main voltage regulator output pin. External capacitor (CVDD) needed. See Section 5.5, “Power Supply". |
| 16 | VDDX | Voltage Regulator Output 5.0 V | +5.0 V main voltage regulator output pin. External capacitor (CVDDX) needed. See Section 5.5, “Power Supply". |
| 17 | DGND | Digital Ground | This pin is the device digital ground connection for the 5.0 V and 2.5V logic. DGND, LGND, and AGND are internally connected to PGND via a back to back diode. |
| 18 | VSENSE | Voltage Sense | Battery voltage sense input. This pin can be connected directly to the battery line for voltage measurements. The voltage present at this input is scaled down by an internal voltage divider, and can be routed to the internal ADC via the analog multiplexer.The pin is self-protected against reverse battery connections. An external resistor (RVSENXSE) is needed for protection [6]. See Section 5.23, “Supply Voltage Sense - VSENSE". Note: This pin function is not available on all device configurations. |

Note:

6. An optional filter capacitor CVSENSE is recommended to be placed between the board connector and DVSENSE to GND for increased ESD performance.

**Table 3. MM912_634 Pin Description (continued)**

| Pin # | Pin Name | Formal Name | Description |
|---|---|---|---|
| 19 | VS1 | Power Supply Pin 1 | This pin is the device power supply pin 1. VS1 is primarily supplying the VDDX Voltage regulator and the Hall Sensor Supply Regulator (HSUP). VS1 can be sensed via a voltage divider through the AD converter. Reverse battery protection diode is required. See Section 5.5, “Power Supply". |
| 20 | VS2 | Power Supply Pin 2 | This pin is the device power supply pin 2. VS2 supplies the High-side Drivers (HSx). Reverse battery protection diode required. See Section 5.5, “Power Supply". |
| 21 | HS1 | High-side Output 1 | This pin is the first High-side output. It is supplied through the VS2 pin. It is designed to drive small resistive loads with optional PWM. In cyclic sense mode, this output activates periodically during low power mode. See Section 5.12, “High-side Drivers - HS". |
| 22 | HS2 | High-side Output 2 | This pin is the second High-side output. It is supplied through the VS2 pin. It is designed to drive small resistive loads with optional PWM. In cyclic sense mode, this output activates periodically during low power mode. See Section 5.12, “High-side Drivers - HS". Note: This pin function is not available on all device configurations. |
| 23 | HSUP | Hall Sensor Supply Output | This pin is designed as an 18 V Regulator to drive Hall Sensor Elements. It is supplied through the VS1 pin. An external capacitor (CHSUP) is needed. See Section 5.11, “Hall Sensor Supply Output - HSUP". Note: This pin function is not available on all device configurations. |
| 24 | LIN | LIN Bus I/O | This pin represents the single-wire bus transmitter and receiver. See Section 5.15, “LIN Physical Layer Interface - LIN". Note: This pin function is not available on all device configurations. |
| 25 | LGND | LIN Ground Pin | This pin is the device LIN Ground connection. DGND, LGND, and AGND are internally connected to PGND via a back to back diode. |
| 26 | PTB0 | General Purpose I/O 0 | This is the General Purpose I/O pin 0 based on VDDX with the following shared functions:<br>• PTB0 - Bidirectional 5.0 V (VDDX) digital port I/O with selectable internal pull-up resistor.<br>• AD0 - Analog Input Channel 0, 0…2.5V (ADC2p5) analog input<br>• TIM0CH0 - Timer Channel 0 Input/Output<br>• Rx - Selectable connection to LIN / SCI<br>See Section 5.18, “General Purpose I/O - PTB[0…2]". |
| 27 | PTB1 | General Purpose I/O 1 | This is the General Purpose I/O pin 1 based on VDDX with the following shared functions:<br>• PTB1 - Bidirectional 5.0 V (VDDX) digital port I/O with selectable internal pull-up resistor.<br>• AD1 - Analog Input Channel 1, 0…2.5 V (ADC2p5) analog input<br>• TIM0CH1 - Timer Channel 1 Input/Output<br>• Tx - Selectable connection to LIN / SCI<br>See Section 5.18, “General Purpose I/O - PTB[0…2]". |
| 28 | PTB2 | General Purpose I/O 2 | This is the General Purpose I/O pin 2 based on VDDX with the following shared functions:<br>• PTB2 - Bidirectional 5.0 V (VDDX) digital port I/O with selectable internal pull-up resistor.<br>• AD2 - Analog Input Channel 2, 0…2.5V (ADC2p5) analog input<br>• TIM0CH2 - Timer Channel 2 Input/Output<br>• PWM - Selectable connection to PWM Channel 0 or 1<br>See Section 5.18, “General Purpose I/O - PTB[0…2]". |
| 29 | ADC2p5 | ADC Reference Voltage | This pin represents the ADC reference voltage and has to be connected to a filter capacitor. See Section 5.20, “Analog Digital Converter - ADC". |
| 30 | AGND | Analog Ground Pin | This pin is the device Analog to Digital converter ground connection. DGND, LGND and AGND are internally connected to PGND via a back to back diode. |
| 31 | L0 | High Voltage Input 0 | This pins is the High Voltage Input 0 with the following shared functions:<br>• L0 - Digital High Voltage Input 0. When used as digital input, a series resistor (RLX) must be used to protect against automotive transients.(7)<br>• AD3 - Analog Input 3 with selectable divider for 0…5.0 V and 0…18 V measurement range.<br>• WU0 - Selectable Wake-up input 0 for wake up and cyclic sense during low power mode.<br>See Section 5.17, “High Voltage Inputs - Lx". |

Note:

7. An optional filter capacitor CLX is recommended to be placed between the board connector and RLX to GND for increased ESD performance.

**Table 3. MM912_634 Pin Description (continued)**

| Pin # | Pin Name | Formal Name | Description |
|---|---|---|---|
| 32 | L1 | High Voltage Input 1 | This pins is the High Voltage Input 1 with the following shared functions:<br>• L1 - Digital High Voltage Input 1. When used as digital input, a series resistor (RLX) must be used to protect against automotive transients.(8)<br>• AD4 - Analog Input 4 with selectable divider for 0…5.0 V and 0…18 V measurement range.<br>• WU1 - Selectable Wake-up input 1 for wake-up and cyclic sense during low power mode.<br>See Section 5.17, “High Voltage Inputs - Lx". |
| 33 | L2 | High Voltage Input 2 | This pins is the High Voltage Input 2 with the following shared functions:<br>• L2 - Digital High Voltage Input 2. When used as digital input, a series resistor (RLX) must be used to protect against automotive transients.(8)<br>• AD5 - Analog Input 5 with selectable divider for 0…5.0 V and 0…18 V measurement range.<br>• WU2 - Selectable Wake-up input 2 for wake-up and cyclic sense during low power mode.<br>See Section 5.17, “High Voltage Inputs - Lx". Note: This pin function is not available on all device configurations. |
| 34 | L3 | High Voltage Input 3 | This pins is the High Voltage Input 3 with the following shared functions:<br>• L3 - Digital High Voltage Input 3. When used as digital input, a series resistor (RLX) must be used to protect against automotive transients.(8)<br>• AD6 - Analog Input 6 with selectable divider for 0…5.0 V and 0…18 V measurement range.<br>• WU3 - Selectable Wake-up input 3 for wake-up and cyclic sense during low power mode.<br>See Section 5.17, “High Voltage Inputs - Lx". Note: This pin function is not available on all device configurations. |
| 35 | L4 | High Voltage Input 4 | This pins is the High Voltage Input 4 with the following shared functions:<br>• L4 - Digital High Voltage Input 4. When used as digital input, a series resistor (RLX) must be used to protect against automotive transients.(8)<br>• AD7 - Analog Input 7 with selectable divider for 0…5.0 V and 0…18 V measurement range.<br>• WU4 - Selectable Wake-up input 4 for wake-up and cyclic sense during low power mode.<br>See Section 5.17, “High Voltage Inputs - Lx". Note: This pin function is not available on all device configurations. |
| 36 | L5 | High Voltage Input 5 | This pins is the High Voltage Input 5 with the following shared functions:<br>• L5 - Digital High Voltage Input 5. When used as digital input, a series resistor (RLX) must be used to protect against automotive transients.(8)<br>• AD8 - Analog Input 8 with selectable divider for 0…5.0 V and 0…18 V measurement range.<br>• WU5 - Selectable Wake-up input 5 for wake-up and cyclic sense during low power mode.<br>See Section 5.17, “High Voltage Inputs - Lx". Note: This pin function is not available on all device configurations. |
| 37 | LS1 | Low-side Output 1 | Low-side output 1 used to drive small inductive loads like relays. The output is short-circuit protected, includes active clamp circuitry and can be also controlled by the PWM module.<br>See Section 5.17, “High Voltage Inputs - Lx" |
| 38 | PGND | Power Ground Pin | This pin is the device Low-side Ground connection. DGND, LGND and AGND are internally connected to PGND via a back to back diode. |
| 39 | LS2 | Low-side Output 2 | Low-side output 2 used to drive small inductive loads like relays. The output is short-circuit protected, includes active clamp circuitry and can be also controlled by the PWM module.<br>See Section 5.17, “High Voltage Inputs - Lx". |
| 40 | ISENSEL | Current Sense Pin L | Current Sense differential input “Low”. This pin is used in combination with ISENSEH to measure the voltage drop across a shunt resistor. See Section 5.21, “Current Sense Module - ISENSE". Note: This pin function is not available on all device configurations. |
| 41 | ISENSEH | Current Sense Pin H | Current Sense differential input “High”. This pin is used in combination with ISENSEL to measure the voltage drop across a shunt resistor. See Section 5.21, “Current Sense Module - ISENSE". Note: This pin function is not available on all device configurations. |
| 42 | NC | Not connected | This pin is reserved for alternative function and should be left floating. |
| 43 | TEST_A | Test Mode | Analog die Test Mode pin for Test Mode only. This pin must be grounded in user mode. |

Note:

8. An optional filter capacitor CLX is recommended to be placed between the board connector and RLX to GND for increased ESD performance.

**Table 3. MM912_634 Pin Description (continued)**

| Pin # | Pin Name | Formal Name | Description |
|---|---|---|---|
| 44 | TCLK | Test Clock Input | Test Mode Clock Input pin for Test Mode only. The pin can be used to disable the internal watchdog for development purpose in user mode. See Section 5.10, “Window Watchdog". The pin is recommended to be grounded in user mode. |
| 45 | $\overline{\text{RESET_A}}$ | Reset I/O | Bidirectional Reset I/O pin of the analog die. Active low signal. Internal pull-up. $V_{DDX}$ based. See Section 5.8, “Resets". To be externally connected to the $\overline{\text{RESET}}$ pin. |
| 46 | $\overline{\text{RESET}}$ | MCU Reset | The RESET pin is an active low bidirectional control signal. It acts as an input to initialize the MCU to a known start-up state, and an output when an internal MCU function causes a reset. The $\overline{\text{RESET}}$ pin has an internal pull-up device to VDDRX. |
| 47 | BKGD | MCU Background Debug and Mode | The BKGD/MODC pin is used as a pseudo-open-drain pin for the background debug communication. It is used as MCU operating mode select pin during reset. The state of this pin is latched to the MODC bit at the rising edge of $\overline{\text{RESET}}$. The BKGD pin has a pull-up device. |
| 48 | PA7 | MCU PA7 | General purpose port A input or output pin 7. See Section 5.28, “Port Integration Module (S12IPIMV1)" |

## 3.2 MCU Die Signal Properties

This section describes the external MCU signals. It includes a table of signal properties.

**Table 4. Signal Properties Summary**

| Pin Name Function 1 | Pin Name Function 2 | Power Supply | Internal Pull Resistor | | Description |
|---|---|---|---|---|---|
| | | | CTRL | Reset State | |
| PE0 | EXTAL | $VD_{DRX}$ | PUPEE/ OSCPINS_EN | DOWN | Port E I/O, Oscillator pin |
| PE1 | XTAL | $V_{DDRX}$ | PUPBE/ OSCPINS_EN | DOWN | Port E I/O, Oscillator pin |
| $\overline{\text{RESET}}$ | — | VDDRX | PULLUP | | External reset |
| TEST | — | N.A. | $\overline{\text{RESET}}$ pin | DOWN | Test input |
| BKGD | MODC | VDDRX | BKPUE | UP | Background debug |
| PA7 | — | VDDRX | NA | NA | Port A I/O |
| PA6 | — | VDDRX | NA | NA | Port A I/O |
| PA5 | — | VDDRX | NA | NA | Port A I/O |
| PA4 | — | VDDRX | NA | NA | Port A I/O |
| PA3 | SS | VDDRX | NA | NA | Port A I/O, SPI |
| PA2 | SCK | VDDRX | NA | NA | Port A I/O, SPI |
| PA1 | MOSI | VDDRX | NA | NA | Port A I/O, SPI |
| PA0 | MISO | VDDRX | NA | NA | Port A I/O, SPI |

# 4 Electrical Characteristics

## 4.1 General

This section contains electrical information for the embedded 9S12I64 microcontroller die, as well as the 912_634 analog die.

## 4.2 Absolute Maximum Ratings

Absolute maximum ratings are stress ratings only. A functional operation under or outside those maxima is not guaranteed. Stress beyond those limits may affect the reliability or cause permanent damage of the device.

This device contains circuitry protecting against damage due to high static voltage or electrical fields. However, it is advised normal precautions be taken to avoid application of any voltages higher than maximum rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate voltage level. All voltages are with respect to ground, unless otherwise noted.

**Table 5. Absolute Maximum Electrical Ratings - Analog Die**

| Symbol | Ratings | Value | Unit |
|---|---|---|---|
| $V_{SUP(SS)}$<br>$V_{SUP(PK)}$<br>$V_{SUP(TR)}$ | Supply Voltage at VS1 and VS2<br>Normal operation (DC)<br>Transient conditions (load dump)<br>Transient input voltage with external component (according to LIN Conformance Test Specification / ISO7637-2) | <br>-0.3 to 27<br>-0.3 to 40 (9) | V |
| $V_{LxDC}$<br>$V_{LxTR}$ | L0…L5 - Pin Voltage<br>Normal operation with a series $R_{LX}$ resistor (DC)<br>Transient input voltage with external component (according to LIN Conformance Test Specification / ISO7637-2) | <br>27 to 40 (9) | V |
| $V_{BUSDC}$<br>$V_{BUSTR}$ | LIN Pin Voltage<br>Normal operation (DC)<br>Transient input voltage with external component (according to LIN Conformance Test Specification / ISO7637-2) | <br>-33 to 40 (9) | V |
| $V_{DDX}$ | Supply Voltage at VDDX | -0.3 to 5.5 | V |
| $V_{DD}$ | Supply Voltage at VDD (10) | -0.3 to 2.75 | V |
| $I_{VDD}$ | VDD Output Current | Internally Limited | A |
| $I_{VDDX}$ | VDDX Output Current | Internally Limited | A |
| $V_{TCLK}$ | TCLK Pin Voltage | -0.3 to 10 | V |
| $V_{IN}$ | $\overline{\text{RESET_A}}$ Pin Voltage | -0.3 to $V_{DDx}$+0.3 | V |
| $V_{IN}$ | Input / Output Pins PTB[0:2] Voltage | -0.3 to $V_{DDx}$+0.3 | V |
| $V_{HS}$ | HS1 and HS2 Pin Voltage (DC) | -0.3 to VS2+0.3 | V |
| $V_{LS}$ | LS1 and LS2 Pin Voltage (DC) | -0.3 to 45 | V |
| $V_{ISENSE}$ | ISENSEH and ISENSEL Pin Voltage (DC) | -0.3 to 40 | V |
| $V_{HSUP}$ | HSUP Pin Voltage (DC) | -0.3 to VS1+0.3 | V |
| $V_{VSENSE}$ | VSENSE Pin Voltage (DC) | -27 to 40 | V |

Note:

9. See Section 4.9, "Additional Test Information ISO7637-2"
10. Caution: As this pin is adjacent to the VDDX pin, care should be taken to avoid a short between VDD and VDDX, for example during soldering process. A short-circuit between these pins might lead to permanent damage.

### Table 6. Maximum Electrical Ratings - MCU Die[11]

| Symbol | Ratings | Value | Unit |
|---|---|---|---|
| $V_{DDRX}$ | 5.0 V Supply Voltage (Supplying the MCU internal regulator for core and flash) | -0.3 to 6.0 | V |
| $V_{DDD2D}$ | 2.5 V D2D - Supply Voltage | -0.3 to 3.6 | V |
| $V_{IN}$ | Digital I/O input voltage (PA0...PA7, PE0, PE1) | -0.3 to 6.0 | V |
| $V_{ILV}$ | EXTAL, XTAL (PE0 and PE1 in alternative configuration) | -0.3 to 2.16 | V |
| $V_{TEST}$ | TEST Input | -0.3 to 10.0 | V |
| $I_D$ | Instantaneous Maximum Current<br>Single pin limit for all digital I/O pins | -25 to 25 | mA |
| $I_{DL}$ | Instantaneous Maximum Current<br>Single pin limit for EXTAL, XTAL | -25 to 25 | mA |

Note:
11. All digital I/O pins are internally clamped to VSSRX and VDDRX.

### Table 7. Maximum Thermal Ratings

| Symbol | Ratings | Value | Unit |
|---|---|---|---|
| $T_{STG}$ | Storage Temperature | -55 to 150 | °C |
| | Package, Thermal Resistance - LQFP48-EP | | °C/W |
| | Four layer board (JEDEC 2s2p) | | |
| $R_{\theta JA}$ | Junction to Ambient Natural Convection [12] | 38 | |
| $R_{\theta JB}$ | Junction to Board [14] | 16 | |
| | Two layer board (JEDEC 1s) | | |
| $R_{\theta JA}$ | Junction to Ambient Natural Convection [12], [13] | 91 | |
| | Package, Thermal Resistance - LQFP48 | | °C/W |
| | Four layer board (JEDEC 2s2p) | | |
| $R_{\theta JA}$ | Junction to Ambient Natural Convection [12] | 59 | |
| $R_{\theta JB}$ | Junction to Board [14] | 31 | |
| | Two layer board (JEDEC 1s) | | |
| $R_{\theta JA}$ | Junction to Ambient Natural Convection [12], [13] | 96 | |
| $T_{PPRT}$ | Peak Package Reflow Temperature During Reflow[15], [16] | Note 16 | °C |

Notes
12. Junction temperature is a function of die size, on-chip power dissipation, package thermal resistance, mounting site (board) temperature, ambient temperature, air flow, power dissipation of other components on the board, and board thermal resistance.
13. Per JEDEC JESD51-6 with the board (JESD51-7) horizontal.
14. Thermal resistance between the die and the printed circuit board per JEDEC JESD51-8. Board temperature is measured on the top surface of the board near the package.
15. Pin soldering temperature limit is for 10 seconds maximum duration. Not designed for immersion soldering. Exceeding these limits may cause malfunction or permanent damage to the device.
16. Freescale's Package Reflow capability meets Pb-free requirements for JEDEC standard J-STD-020C. For Peak Package Reflow Temperature and Moisture Sensitivity Levels (MSL), Go to www.freescale.com, search by part number [e.g. remove prefixes/suffixes and enter the core ID to view all orderable parts (i.e. MC33xxxD enter 33xxx), and review parametrics.

## 4.3 Operating Conditions

This section describes the operating conditions of the device. Unless otherwise noted those conditions apply to all the following data.

**Table 8. Operating Conditions**

| Symbol | Ratings | Value | Unit |
|---|---|---|---|
| $V_{SUP}$ | Analog Die Nominal Operating Voltage | 5.5 to 18 | V |
| $V_{SUPOP}$ | Analog Die Functional Operating Voltage - Device is fully functional. All features are operating. | 5.5 to 27 | V |
| $V_{DDRX}$ | MCU I/O and Supply Voltage(17) | 4.75 to 5.25 | V |
| $V_{DDD2D}$ | MCU Digital Logic Supply Voltage(17) | 2.25 to 2.75 | V |
| $f_{OSC}$ | MCU Oscillator<br>MM912x634xxxAE<br>MM912x634xxxAP | <br>4.0 to 16<br>4.0 to 16 | MHz |
| $f_{BUS}$ | MCU Bus frequency<br>MM912x634xxxAE<br>MM912x634xxxAP | $f_{BUSMAX}$(18) | MHz |
| $T_A$ | Operating Ambient Temperature<br>MM912x634xMxxx<br>MM912x634xVxxx | <br>-40 to 125<br>-40 to 105 | °C |
| $T_{J_A}$ | Operating Junction Temperature - Analog Die | -40 to 150 | °C |
| $T_{J_M}$ | Operating Junction Temperature - MCU Die | -40 to 150 | °C |

Note:

17. During power up and power down sequence always $V_{DDD2D} < V_{DDRX}$
18. $f_{BUSMAX}$ frequency ratings differ by device and is specified in Table 1

## 4.4 Supply Currents

This section describes the current consumption characteristics of the device as well as the conditions for the measurements.

### 4.4.1 Measurement Conditions

All measurements are without output loads. Currents are measured in MCU special single chip mode and the CPU code is executed from RAM, unless otherwise noted.

**Table 9. Supply Currents**

| Symbol | Ratings | Min | Typ[19] | Max | Unit |
|---|---|---|---|---|---|
| $I_{RUN_A}$ | Normal Mode analog die only, excluding external loads, LIN Recessive State ($5.5\ V \le V_{SUP} \le 18\ V$, $2.25\ V \le V_{DD} \le 2.75\ V$, $4.5\ V \le V_{DDX} \le 5.5\ V$, $-40\ °C \le T_{J_A} \le 150\ °C$). | - | 5.0 | 8.0 | mA |
| $I_{RUN_M}$ | Normal Mode MCU die only ($T_{J_M}$=150 °C; $V_{DDD2D}$ = 2.75 V, $V_{DDRX}$ = 5.5 V, $f_{OSC}$ = 4.0 MHz, $f_{BUS}$= $f_{BUSMAX}$[20][21] | - | 18 | 20 | mA |
| $I_{STOP_A}$ | Stop Mode internal analog die only, excluding external loads, LIN Recessive State, Lx enabled, measured at VS1+VS2 ($5.5\ V \le V_{SUP} \le 18\ V$, $2.25\ V \le V_{DD} \le 2.75\ V$, $4.5\ V \le V_{DDX} \le 5.5\ V$)<br>$-40\ °C \le T_{J_A} \le 125\ °C$ | <br><br>- | <br><br>20 | <br><br>40 | µA |
| $I_{STOP_M}$ | Stop Mode MCU die only ($V_{DDD2D}$ = 2.75 V, $V_{DDRX}$ = 5.5 V, $f_{OSC}$= 4.0 MHz; MCU in STOP; RTI and COP off)[22]<br>$T_{J_M}$=150°C<br>$T_{J_M}$=-40°C<br>$T_{J_M}$=25°C | <br>-<br>-<br>- | <br>85<br>31<br>31 | <br>150<br>50<br>50 | µA |
| $I_{SLEEP}$ | Sleep Mode (VDD = VDDX = OFF; $5.5\ V \le V_{SUP} \le 18\ V$; $-40\ °C \le T_{J_A} \le 125\ °C$; $3.0\ V > Lx > 1.0\ V$) | - | 15 | 28 | µA |
| $I_{CS}$ | Cyclic Sense Supply Current Adder (5.0 ms Cycle) | - | 15 | 20 | µA |

Note:

19. Typical values noted reflect the approximate parameter mean at $T_A$ = 25 °C
20. $f_{BUSMAX}$ frequency ratings differ by device and is specified in Table 1
21. $I_{RUN_M}$ denotes the sum of the currents flowing into VDD and VDDX.
22. $I_{STOP_M}$ denotes the sum of the currents flowing into VDD and VDDX.

## 4.5 Static Electrical Characteristics

All characteristics noted under the following conditions:

- 5.5 V ≤ $V_{SUP}$ ≤ 18 V
- -40 °C ≤ $T_A$ ≤ 125 °C (MM912x634xMxxx)
- -40 °C ≤ $T_A$ ≤ 105 °C (MM912x634xVxxx)

Typical values noted reflect the approximate parameter mean at $T_A$ = 25 °C under nominal conditions, unless otherwise noted.

### 4.5.1 Static Electrical Characteristics Analog Die

**Table 10. Static Electrical Characteristics - Power Supply**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $V_{POR}$ | Power-On Reset (POR) Threshold (measured on VS1) | 1.5 | - | 3.5 | V |
| $V_{LVI}$<br>$V_{LVI_H}$ | Low Voltage Warning (LVI)<br>Threshold (measured on VS1, falling edge)<br>Hysteresis (measured on VS1) | <br>5.55<br>- | <br>6.0<br>1.0 | <br>6.6<br>- | V |
| $V_{HVI}$<br>$V_{HVI_H}$ | High Voltage Warning (HVI)<br>Threshold (measured on VS2, rising edge)<br>Hysteresis (measured on VS2) | <br>18<br>- | <br>19.25<br>1.0 | <br>20.5<br>- | V |
| $V_{LBI}$<br>$V_{LBI_H}$ | Low Battery Warning (LBI)<br>Threshold (measured on VSENSE, falling edge)<br>Hysteresis (measured on VSENSE) | <br>5.55<br>- | <br>6.0<br>1.0 | <br>6.6<br>- | V |
| $V_{J2602UV}$ | J2602 Undervoltage threshold | 5.5 | 5.7 | 6.2 | V |
| $V_{LVRX}$ | Low VDDX Voltage (LVRX) Threshold | 2.7 | 3.0 | 3.3 | V |
| $V_{LVR}$ | Low VDD Voltage Reset (LVR) Threshold Normal Mode | 2.30 | 2.35 | 2.4 | V |
| $V_{LVRS}$ | Low VDD Voltage Reset (LVR) Threshold Stop Mode (23) | 1.6 | 1.85 | 2.1 | V |
| $V_{VDDOV}$ | VDD Overvoltage Threshold (VROV) | 2.575 | 2.7875 | 3.0 | V |
| $V_{VDDXOV}$ | VDDX Overvoltage Threshold (VROVX) | 5.25 | 5.675 | 6.1 | V |

Note:
23. See MM912_634ER - MM912_634, Silicon Analog Mask (M91W) / Digital Mask (N53A) Errata

**Table 11. Static Electrical Characteristics - Resets**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $V_{OL}$ | Low-state Output Voltage $I_{OUT}$ = 2.0 mA | - | - | 0.8 | V |
| $R_{RPU}$ | Pull-up Resistor | 25 | - | 50 | kΩ |
| $V_{IL}$ | Low-state Input Voltage | - | - | $0.3V_{DDX}$ | V |
| $V_{IH}$ | High-state Input Voltage | $0.7V_{DDX}$ | - | - | V |
| $V_{RSTRV}$ | Reset Release Voltage (VDDX) | - | 1.5 | - | V |
| | RESET_A pin Current Limitation | 5.0 | 7.5 | 10 | mA |

**Table 12. Static Electrical Characteristics - Window Watchdog**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $V_{TST}$ | Watchdog Disable Voltage (fixed voltage) | 7.0 | - | 10 | V |
| $V_{TSTEN}$ | Watchdog Enable Voltage (fixed voltage) | - | - | 5.5 | V |

**Table 13. Static Electrical Characteristics - Voltage Regulator 5.0 V (VDDX)**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $V_{DDXRUN}$ | Normal Mode Output Voltage<br>1.0 mA < $I_{VDDX}$ + $I_{VDDXINTERNAL}$ < 80 mA; 5.5 V < $V_{SUP}$ < 27 V (24) | 4.75 | 5.00 | 5.25 | V |
| $I_{VDDXRUN}$ | Normal Mode Output Current Limitation ($I_{VDDX}$) | 80 | 130 | 200 | mA |
| $V_{DDXSTOP}$ | Stop Mode Output Voltage ($I_{VDDX}$ + $I_{VDDXINTERNAL}$ < 500 µA for $T_J \geq 25$ °C; $I_{VDDX}$ + $I_{VDDXINTERNAL}$ < 400 µA for $T_J$ < 25 °C) (24) | - | 5.0 | 5.5 | V |
| $I_{VDDXSTOP}$ | Stop Mode Output Current Limitation ($I_{VDDX}$) | 1.0 | - | 20 | mA |
| $LR_{XRUN}$<br>$LR_{XSTOP}$ | Line Regulation<br>Normal Mode, $I_{VDDX}$ = 80 mA<br>Stop Mode, $I_{VDDX}$ = 500 µA | <br>-<br>- | <br>20<br>- | <br>25<br>200 | mV |
| $LD_{XRUN}$<br>$LD_{XCRK}$<br>$LD_{XSTOP}$ | Load Regulation<br>Normal Mode, 1.0 mA < $I_{VDDX}$ < 80 mA<br>Normal Mode, $V_{SUP}$ = 3.6 V, 1.0 mA < $I_{VDDX}$ < 40 mA<br>Stop Mode, 0.1 mA < $I_{VDDX}$ < 500 µA | <br>-<br>-<br>- | <br>15<br>-<br>- | <br>80<br>200<br>250 | mV |
| $C_{VDDX}$ | External Capacitor | 1.0 | - | 10 | µF |
| $C_{VDDX_R}$ | External Capacitor ESR | - | - | 10 | Ω |

Note:
24. $I_{VDDXINTERNAL}$ includes internal consumption from both analog and MCU die.

**Table 14. Static Electrical Characteristics - Voltage Regulator 2.5 V (VDD)**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $V_{DDRUN}$ | Normal Mode Output Voltage<br>1.0 mA < $I_{VDD}$ + $I_{VDDINTERNAL}$ ≤ 45 mA; 5.5 V < $V_{SUP}$ < 27 V (25) | 2,425 | 2.5 | 2,575 | V |
| $I_{VDDLIMRUN}$ | Normal Mode Output Current Limitation ($I_{VDD}$)<br>$T_J$ < 25 °C<br>$T_J \geq$ 25 °C | <br>-<br>- | <br>80<br>80 | <br>120<br>143 | mA |
| $V_{DDSTOP}$ | Stop Mode Output Voltage ($I_{VDD}$ + $I_{VDDINTERNAL}$ < 500 µA for $T_J \geq 25$ °C; $I_{VDD}$ + $I_{VDDINTERNAL}$ < 400 µA for $T_J$ < 25 °C) (25) | 2.25 | 2.5 | 2.75 | V |
| $I_{VDDLIMSTOP}$ | Stop Mode Output Current Limitation ($I_{VDD}$) | - | - | 10 | mA |
| $LR_{RUN}$<br>$LR_{STOP}$ | Line Regulation<br>Normal Mode, $I_{VDD}$ = 45 mA<br>Stop Mode, $I_{VDD}$ = 1.0 mA | <br>-<br>- | <br>10<br>- | <br>12.5<br>200 | mV |
| $LD_{RUN}$<br>$LD_{CRK}$<br>$LD_{STOP}$ | Load Regulation<br>Normal Mode, 1.0 mA < $I_{VDD}$ < 45 mA<br>Normal Mode, $V_{SUP}$ = 3.6 V, 1.0 mA < $I_{VDD}$ < 30 mA<br>Stop Mode, 0.1 mA < $I_{VDD}$ < 1.0 mA | <br>-<br>-<br>- | <br>7.5<br>-<br>- | <br>40<br>40<br>200 | mV |
| $C_{VDD}$ | External Capacitor | 1.0 | - | 10 | µF |
| $C_{VDD_R}$ | External Capacitor ESR | - | - | 10 | Ω |

Note:
25. $I_{VDDINTERNAL}$ includes internal consumption from both analog and MCU die.

**Table 15. Static Electrical Characteristics - Hall Sensor Supply Output - HSUP**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $I_{HSUP}$ | Current Limitation | 40 | 70 | 90 | mA |
| $R_{DS(ON)}$ | Output Drain-to-Source On resistance<br>$T_J$ = 150 °C, ILOAD = 30 mA; 5.5 V ≤ VSUP ≤ 16 V<br>$T_J$ = 150 °C, ILOAD = 30 mA; 3.7 V ≤ VSUP < 5.5 V | <br>-<br>- | <br>-<br>- | <br>10<br>13 | Ω |
| $VHSUP_{MAX}$ | Output Voltage: (18 V ≤ $V_{SUP}$ ≤ 27 V) | 16 | 17.5 | 18 | V |
| $LD_{HSUP}$ | Load Regulation (1.0 mA < $I_{HSUP}$ < 30 mA; $V_{SUP}$ > 18 V) | - | - | 500 | mV |
| $C_{HSUP}$ | Hall Supply Capacitor Range | 0.22 | - | 10 | µF |
| $C_{HSUP_R}$ | External Capacitor ESR | - | - | 10 | Ω |

**Table 16. Static Electrical Characteristics - High-side Drivers - HS**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $R_{DS(on)}$ | Output Drain-to-Source On resistance<br>$T_J$ = 25 °C, $I_{LOAD}$ = 50 mA; $V_{SUP}$ > 9.0 V<br>$T_J$ = 150 °C, $I_{LOAD}$ = 50 mA; $V_{SUP}$ > 9.0 V<br>$T_J$ = 150 °C, $I_{LOAD}$ = 30 mA; 5.5 V < $V_{SUP}$ < 9.0 V | <br>-<br>-<br>- | <br>-<br>-<br>- | <br>7.0<br>10<br>14 | Ω |
| $I_{LIMHSX}$ | Output Current Limitation (0 V < $V_{OUT}$ < $V_{SUP}$ - 2.0 V) | 60 | 110 | 250 | mA |
| $I_{OLHSX}$ | Open Load Current Detection | - | 5.0 | 7.5 | mA |
| $I_{LEAK}$ | Leakage Current (-0.2 V < $V_{HSx}$ < $V_{S2}$ + 0.2 V) | - | - | 10 | µA |
| $V_{THSC}$ | Current Limitation Flag Threshold (5.5 V < $V_{SUP}$ < 27 V) | $V_{SUP}$ -2 | - | - | V |

**Table 17. Static Electrical Characteristics - Low-side Drivers - LS**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $R_{DS(on)}$ | Output Drain-to-Source On resistance<br>$T_J$ = 25 °C, $I_{LOAD}$ = 150 mA, $V_{SUP}$ > 9.0 V<br>$T_J$ = 150 °C, $I_{LOAD}$ = 150 mA, $V_{SUP}$ > 9.0 V<br>$T_J$ = 150 °C, $I_{LOAD}$ = 120 mA, 5.5 V < $V_{SUP}$ < 9.0 V | <br>–<br>–<br>– | <br>–<br>–<br>– | <br>2.5<br>4.5<br>10 | Ω |
| $I_{LIMLSX}$ | Output Current Limitation (2.0 V < $V_{OUT}$ < $V_{SUP}$) | 180 | 275 | 380 | mA |
| $I_{OLLSX}$ | Open Load Current Detection | - | 8.0 | 12 | mA |
| $I_{LEAK}$ | Leakage Current (-0.2 V < $V_{OUT}$ < VS1) | - | - | 10 | µA |
| $V_{CLAMP}$ | Active Output Energy Clamp ($I_{OUT}$ = 150 mA) | 40 | - | 45 | V |
| $R_{COIL}$ | Coil Series Resistance ($I_{OUT}$ = 150 mA) | 120 | - | | Ω |
| $R_{COIL}$ | Coil Inductance ($I_{OUT}$ = 150 mA) | - | - | 400 | mH |
| $V_{THSC}$ | Current Limitation Flag Threshold (5.5 V < $V_{SUP}$ < 27 V) | 2.0 | - | - | V |

**Table 18. Static Electrical Characteristics - LIN Physical Layer Interface - LIN**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $I_{BUSLIM}$ | Current Limitation for Driver dominant state. $V_{BUS}$ = 18 V | 40 | 120 | 200 | mA |
| $I_{BUS_PAS_DOM}$ | Input Leakage Current at the Receiver incl. Pull-up Resistor RSLAVE; Driver OFF; $V_{BUS}$ = 0 V; $V_{BAT}$ = 12 V | -1.0 | - | - | mA |
| $I_{BUS_PAS_REC}$ | Input Leakage Current at the Receiver incl. Pull-up Resistor RSLAVE; Driver OFF;<br>8.0 V < $V_{BAT}$ < 18 V; 8.0 V < $V_{BUS}$ < 18 V; $V_{BUS} \geq V_{BAT}$ | - | - | 20 | µA |
| $I_{BUS_NO_GND}$ | Input Leakage Current; GND Disconnected; GNDDEVICE = VSUP; 0 < $V_{BUS}$ < 18 V; $V_{BAT}$ = 12 V | -1.0 | - | 1.0 | mA |
| IBUS_NO_BAT | Input Leakage Current; VBAT disconnected; VSUP_DEVICE = GND; 0 < $V_{BUS}$ < 18 V | - | - | 100 | µA |
| $V_{BUSDOM}$ | Receiver Input Voltage; Receiver Dominant State | - | - | 0.4 | $V_{SUP}$ |
| $V_{BUSREC}$ | Receiver Input Voltage; Receiver Recessive State | 0.6 | - | - | $V_{SUP}$ |
| $V_{BUS_CNT}$ | Receiver Threshold Center ($V_{TH_DOM}$ + $V_{TH_REC}$)/2 | 0.475 | 0.5 | 0.525 | $V_{SUP}$ |
| $V_{BUS_HYS}$ | Receiver Threshold Hysteresis ($V_{TH_REC}$ - $V_{TH_DOM}$) | - | - | 0.175 | $V_{SUP}$ |
| $D_{SER_INT}$ | Voltage Drop at the serial Diode | 0.4 | 0.7 | 1.0 | V |
| $R_{SLAVE}$ | LIN Pull-up Resistor | 20 | 30 | 60 | kΩ |
| $V_{WUP}$ | Bus Wake-up Threshold from Stop or Sleep | 4.0 | 5.0 | 6.0 | V |
| $V_{DOM}$ | Bus Dominant Voltage | - | - | 2.5 | V |

**Table 19. Static Electrical Characteristics - High Voltage Inputs - Lx**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $V_{THL}$ | Low Detection Threshold<br>(7.0 V ≤ $V_{SUP}$ ≤ 27 V)<br>(5.5 V ≤ $V_{SUP}$ ≤ 7.0 V) | <br>2.2<br>1.5 | <br>2.5<br>2.5 | <br>3.4<br>4.0 | V |
| $V_{THH}$ | High Detection Threshold<br>(7.0 V ≤ $V_{SUP}$ ≤ 27 V)<br>(5.5 V ≤ $V_{SUP}$ ≤ 7.0 V) | <br>2.6<br>2.0 | <br>3.0<br>3.0 | <br>3.7<br>4.5 | V |
| $V_{HYS}$ | Hysteresis<br>(5.5 V < $V_{SUP}$ < 27 V) | <br>0.25 | <br>0.45 | <br>1.0 | V |
| $I_{IN}$ | Input Current Lx (-0.2 V < $V_{IN}$ < VS1) | -10 | - | 10 | µA |
| $R_{LxIN}$ | Analog Input Impedance Lx | - | - | 1.2 | MΩ |
| $R_{LX}$ | Lx Series Resistor | 9.5 | 10 | 10.5 | kΩ |
| $C_{LX}$ | Lx Capacitor (optional)(26) | - | 100 | - | nF |
| $RATIO_{Lx}$ | Analog Input Divider Ratio ($RATIO_{Lx}$ = $V_{Lx}$ / $V_{ADOUT0}$)<br>LXDS (Lx Divider Select) = 0<br>LXDS (Lx Divider Select) = 1 | <br>-<br>- | <br>2.0<br>7.2 | <br>-<br>- | |
| $RATIO_{LX}$ | Analog Input Divider Ratio Accuracy | -5.5 | - | 5.5 | % |

**Table 19. Static Electrical Characteristics - High Voltage Inputs - Lx**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $Lx_{MATCH}$ | Analog Inputs Channel Ratio - Mismatch<br>LXDS (Lx Divider Select) = 0<br>LXDS (Lx Divider Select) = 1 | <br>-<br>- | <br>-<br>- | <br>5.0<br>5.0 | <br>%<br> |

Note:
26. The ESD behavior specified in Section 4.8, “ESD Protection and Latch-up Immunity" are guaranteed without the optional capacitor.

**Table 20. Static Electrical Characteristics - General Purpose I/O - PTB[0…2]**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $V_{IH}$ | Input High Voltage | $0.7V_{DDX}$ | - | $V_{DDX}$+0.3 | V |
| $V_{IL}$ | Input Low Voltage | $V_{SS}$-0.3 | - | $0.35V_{DDX}$ | V |
| $V_{HYS}$ | Input Hysteresis | - | 140 | - | mV |
| $V_{IH3.7}$ | Input High Voltage (VS1 = 3.7 V) | 2.1 | - | $V_{DDX}$+0.3 | V |
| $V_{IL3.7}$ | Input Low Voltage (VS1 = 3.7 V) | $V_{SS}$-0.3 | - | 1.4 | V |
| $V_{HYS3.7}$ | Input Hysteresis (VS1 = 3.7 V) | 100 | 200 | 300 | mV |
| $I_{IN}$ | Input Leakage Current (pins in high-impedance input mode) ($V_{IN}$ = $V_{DDX}$ or $V_{SSX}$) | -1.0 | - | 1.0 | µA |
| $V_{OH}$ | Output High Voltage (pins in output mode) Full drive $I_{OH}$ = -10 mA | $V_{DDX}$-0.8 | - | - | V |
| $V_{OL}$ | Output Low Voltage (pins in output mode) Full drive $I_{OL}$ = 10 mA | - | - | 0.8 | V |
| $R_{PUL}$ | Internal Pull-up Resistance ($V_{IH}$ min > Input voltage > $V_{IL}$ max) | 26.25 | 37.5 | 48.75 | kΩ |
| $C_{IN}$ | Input Capacitance | - | 6.0 | - | pF |
| $V_{CL_AIN}$ | Clamp Voltage when selected as analog input | VDD | - | - | V |
| $R_{AIN}$ | Analog Input impedance = 10 kΩ max, Capacitance = 12 pF | - | - | 10 | kΩ |
| $C_{AIN}$ | Analog Input Capacitance = 12 pF | - | 12 | - | pF |
| $I_{BMAX}$ | Maximum current all PTB combined (VDDX capability) | -15 | - | 15 | mA |
| $C_{OUT}$ | Output Drive strength at 10 MHz | - | - | 100 | pF |

**Table 21. Static Electrical Characteristics - Analog Digital Converter - ADC(27)**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $V_{ADC2p5RUN}$ | ADC2p5 Reference Voltage, 5.5 V < $V_{SUP}$ < 27 V | 2,45 | 2.5 | 2.55 | V |
| $V_{ADC2p5STOP}$ | ADC2p5 Reference Stop Mode Output Voltage | - | - | 100 | mV |
| $LR_{RUNA}$ | Line Regulation, Normal Mode | - | 10 | 12.5 | mV |
| $C_{ADC2p5}$ | External Capacitor | 0.1 | - | 1.0 | µF |
| $C_{VDD_R}$ | External Capacitor ESR | - | - | 10 | Ω |
| $E_{SCALE}$ | Scale Factor Error | -1 | - | 1 | LSB |
| $E_{DNL}$ | Differential Linearity Error | -1.5 | - | 1.5 | LSB |
| $E_{INL}$ | Integral Linearity Error | -1.5 | - | 1.5 | LSB |
| $E_{OFF}$ | Zero Offset Error | -2.0 | - | 2.0 | LSB |
| $E_Q$ | Quantization Error | -0.5 | - | 0.5 | LSB |

**Table 21. Static Electrical Characteristics - Analog Digital Converter - ADC[27]**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| TE | Total Error with offset compensation | -5.0 | - | 5.0 | LSB |
| $AD_{CH14}$ | Bandgap measurement Channel (CH14) Valid Result Range (including ±7.0% bg1p25sleep accuracy + high-impedance measurement error of ±5.0% at $f_{ADC}$)[28] | 1.1 | 1.25 | 1.4 | V |

Note:
27. No external load allowed on the ADC2p5 pin.
28. Reduced ADC frequency lowers measurement error.

**Table 22. Static Electrical Characteristics - Current Sense Module - ISENSE**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| G | Gain | | | | |
| | CSGS (Current Sense Gain Select) = 000 | - | 7 | - | |
| | CSGS (Current Sense Gain Select) = 001 | - | 9 | - | |
| | CSGS (Current Sense Gain Select) = 010 | - | 10 | - | |
| | CSGS (Current Sense Gain Select) = 011 | - | 12 | - | |
| | CSGS (Current Sense Gain Select) = 100 | - | 14 | - | |
| | CSGS (Current Sense Gain Select) = 101 | - | 18 | - | |
| | CSGS (Current Sense Gain Select) = 110 | - | 24 | - | |
| | CSGS (Current Sense Gain Select) = 111 | - | 36 | - | |
| | Gain Accuracy | -3.0 | - | 3.0 | % |
| | Offset | -1.5 | - | 1.5 | % |
| RES | Resolution[29] | - | 51 | - | mA/LSB |
| $V_{IN}$ | ISENSEH, ISENSEL Input Common Mode Voltage Range | -0.2 | - | 3.0 | V |
| $I_{ISENSE}$ | Current Sense Module - Normal Mode Current Consumption Adder (CSE = 1) | - | 600 | - | µA |

Note:
29. RES = 2.44 mV/(GAIN*$R_{SHUNT}$)

**Table 23. Static Electrical Characteristics - Temperature Sensor - TSENSE**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $TS_G$ | Internal Chip Temperature Sense Gain[30] | - | 9.17 | - | mV/k |
| $TS_{ERR}$ | Internal Chip Temperature Sense Error at the end of conversion[30] | –5.0 | - | 5.0 | °C |
| $T_{0.15V}$ | Temperature represented by a $ADC_{IN}$ Voltage of 0.150 V[30] | -55 | -50 | -45 | °C |
| $T_{1.984V}$ | Temperature represented by a $ADC_{IN}$ Voltage of 1.984 V[30] | 145 | 150 | 155 | °C |

Note:
30. Guaranteed by design and characterization.

**Table 24. Static Electrical Characteristics - Supply Voltage Sense - VSENSE and VS1SENSE**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $RATIO_{VSENSE}$ | VSENSE Input Divider Ratio ($RATIO_{VSENSE} = V_{VSENSE}$ / ADCIN)<br>5.5 V < $V_{SUP}$ < 27 V | - | 10.8 | 5.0% | |
| $Er_{VSENSE}$ | VSENSE error - whole path (VSENSE pad to Digital value) | - | - | 5.0 | % |
| $RATIO_{VS1SENSE}$ | VS1SENSE Input Divider Ratio ($RATIO_{VS1SENSE} = V_{VS1SENSE}$ / ADCIN)<br>5.5 V < $V_{SUP}$ < 27 V | - | 10.8 | 5.0% | |
| $Er_{VS1SENSE}$ | VS1SENSE error - whole path (VS1 pad to Digital value) | - | - | 5.0 | % |
| $R_{VSENSE}$ | VSENSE Series Resistor | 9.5 | 10 | 10.5 | kΩ |
| $C_{VSENSE}$ | VSENSE Capacitor (optional)(31) | - | 100 | - | nF |

Note:
31. The ESD behavior specified in Section 4.8, “ESD Protection and Latch-up Immunity" is guaranteed without the optional capacitor.

### 4.5.2 Static Electrical Characteristics MCU Die

#### 4.5.2.1 I/O Characteristics

This section describes the characteristics of all I/O pins except EXTAL, XTAL, TEST and supply pins.

**Table 25. 5.0 V I/O Characteristics for PTA, PTE, $\overline{\text{RESET}}$ and BKGD Pins**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $V_{IH}$ | Input high voltage | 0.65 *$V_{DDRX}$ | - | - | V |
| $V_{IH}$ | Input high voltage | - | - | $V_{DDRX}$ + 0.3 | V |
| $V_{IL}$ | Input low voltage | - | - | 0.35*$V_{DDRX}$ | V |
| $V_{IL}$ | Input low voltage | $V_{SSRX}$ - 0.3 | - | - | V |
| $V_{HYS}$ | Input hysteresis | - | 250 | - | mV |
| $I_{IN}$ | Input leakage current (pins in high-impedance input mode)<br>$V_{IN}$ = $V_{DDRX}$ or $V_{SSRX}$ | -1.0 | - | 1.0 | μA |
| $V_{OH}$ | Output high voltage (pins in output mode) $I_{OH}$ = -4.0 mA | $V_{DDRX}$ – 0.8 | - | - | V |
| $V_{OL}$ | Output low voltage (pins in output mode) $I_{OL}$ = +4.0 mA | - | - | 0.8 | V |
| $R_{PUL}$ | Internal pull-up resistance ($V_{IH}$min > input voltage > $V_{IL}$max) | 25 | - | 50 | kΩ |
| $R_{PDH}$ | Internal pull-down resistance ($V_{IH}$min > input voltage > $V_{IL}$max) | 25 | - | 50 | kΩ |
| $C_{in}$ | Input capacitance | - | 6.0 | - | pF |
| $I_{ICS}$<br>$I_{ICP}$ | Injection current(32)<br>Single pin limit<br>Total device Limit, sum of all injected currents | -2.5<br>-25 | -<br>- | 2.5<br>25 | mA |

Note:
32. Refer to Section 4.8, “ESD Protection and Latch-up Immunity" for more details.

### 4.5.2.2 Electrical Specification for MCU internal Voltage Regulator

**Table 26. IVREG Characteristics**

| Symbol | Characteristic | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| $V_{VDDRA}$ | Input Voltages | 3.13 | — | 5.5 | V |
| $V_{LVIA}$<br>$V_{LVID}$ | $V_{DDRX}$ Low Voltage Interrupt Assert Level<br>$V_{DDRX}$ Low Voltage Interrupt Deassert Level | 4.04<br>4.19 | 4.23<br>4.38 | 4.40<br>4.49 | V |
| $V_{LVRXD}$ | $V_{DDRX}$ Low Voltage Reset Deassert (33) (34) (35) | — | 3.05 | 3.13 | V |
| $V_{LVRXA}$ | $V_{DDRX}$ Low Voltage Reset Assert (33) (34) (35) | 2.95 | 3.02 | — | V |

Note:
33. Device functionality is guaranteed on power down to the LVR assert level.
34. Monitors VDDRX, active only in Full Performance mode. MCU is monitored by the POR in RPM (see Figure 4).
35. Monitors VDDRX, active only in Full Performance mode. $V_{LVRA}$ and $V_{PORD}$.

**NOTE**

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

### 4.5.2.3 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.

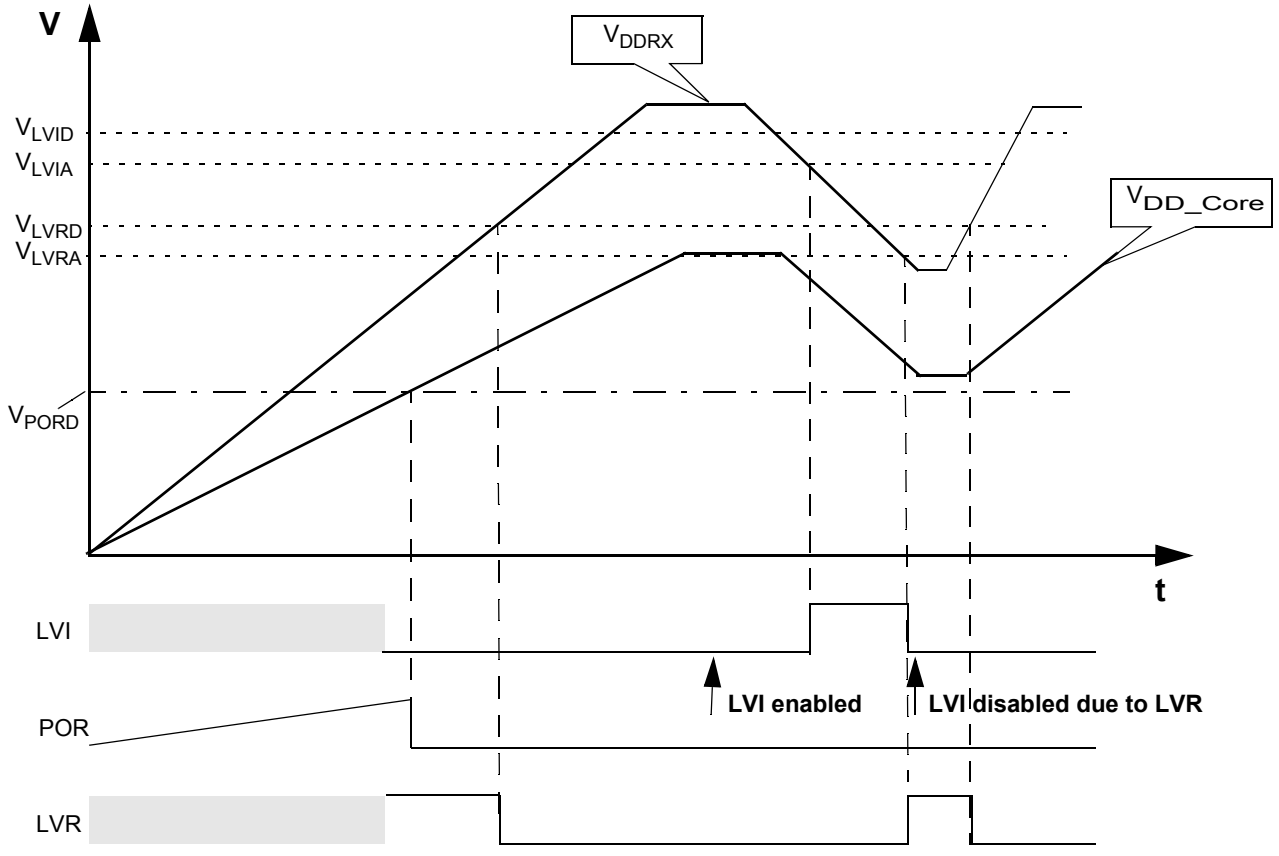



**Figure 4. MC9S12I64 - Chip Power-up and Voltage Drops (not scaled)**

## 4.6 Dynamic Electrical Characteristics

Dynamic characteristics noted under conditions 5.5 V ≤ $V_{SUP}$ ≤ 18 V, -40 °C ≤ $T_A$ ≤ 125 °C, unless otherwise noted. Typical values noted reflect the approximate parameter mean at $T_A$ = 25 °C under nominal conditions, unless otherwise noted.

### 4.6.1 Dynamic Electrical Characteristics Analog Die

**Table 27. Dynamic Electrical Characteristics - Modes of Operation**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $t_{VTO}$ | VDD Short Timeout | 110 | 150 | 205 | ms |
| $f_{BASE}$ | Analog Base Clock | - | 100 | - | kHz |
| $t_{RST}$ | Reset Delay | 140 | 200 | 280 | µs |

**Table 28. Dynamic Electrical Characteristics - Power Supply(36)**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $t_{LB}$ | Glitch Filter Low Battery Warning (LBI) | - | 2.0 | - | µs |
| $t_{LV}$ | Glitch Filter Low Voltage Warning (LVI) | - | 2.0 | - | µs |
| $t_{HV}$ | Glitch Filter High Voltage Warning (HVI) | - | 2.0 | - | µs |

Note:
36. Guaranteed by design.

**Table 29. Dynamic Electrical Characteristics - Die to Die Interface - D2D**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $f_{D2D}$ | Operating Frequency (D2DCLK, D2D[0:3]) | - | - | $f_{BUSMAX}$(37) | MHz |

Note:
37. $f_{BUSMAX}$ frequency ratings differ by device and is specified in Table 1

**Table 30. Dynamic Electrical Characteristics - Resets**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $t_{RSTDF}$ | Reset Deglitch Filter Time | 1.2 | 2.0 | 3.0 | µs |
| $t_{RSTLOW}$ | Reset Low Level Duration | 140 | 200 | 280 | µs |

**Table 31. Dynamic Electrical Characteristics - Wake-up / Cyclic Sense**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $t_{WUF}$ | Lx Wake-up Filter Time | - | 20 | | µs |
| $CS_{AC}$ | Cyclic Sense/Forced Wake-up Timing Accuracy - not trimmed | -35 | - | 35 | % |
| $CS_{ACT}$ | Cyclic Sense/Forced Wake-up Timing Accuracy - trimmed(38) | -5.0 | - | 5.0 | % |
| $t_S$ | Time Between HSx on and Lx sense during cyclic sense | same as $t_{HSON}$ / $t_{HSONT}$ | | | - |
| $t_{HSON}$ | HSx ON Duration During Cyclic Sense | 140 | 200 | 280 | µs |
| $t_{HSONT}$ | HSx ON Duration During Cyclic Sense - trimmed(38) | 180 | 200 | 220 | µs |

Note:
38. No trimming possible in Sleep mode.

**Table 32. Dynamic Electrical Characteristics - Window Watchdog**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $t_{IWDTO}$ | Initial Non-window Watchdog Timeout | 110 | 150 | 190 | ms |
| $WD_{AC}$ | Watchdog Timeout Accuracy - not trimmed | -35 | - | 35 | % |
| $WD_{ACT}$ | Watchdog Timeout Accuracy - trimmed | -5.0 | - | 5.0 | % |

**Table 33. Dynamic Electrical Characteristics - High-side Drivers - HS**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $f_{HS}$ | High-side Operating Frequency[39]<br>Load Condition: $C_{LOAD} \leq 2.2$ nF; $R_{LOAD} \geq 500\ \Omega$ | - | - | 50 | kHz |

Note:
39. Guaranteed by design.

**Table 34. Dynamic Electrical Characteristics - Low-side Drivers - LS**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $f_{LS}$ | Low-side Operating Frequency [40]<br>Load Condition: $C_{LOAD} \leq 2.2$ nF; $R_{LOAD} \geq 500\ \Omega$ | - | - | 50 | kHz |

Note:
40. Guaranteed by design.

**Table 35. Dynamic Electrical Characteristics - LIN Physical Layer Interface - LIN**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $t_{PROPWL}$ | Bus Wake-up Deglitcher (Sleep and Stop mode) | 60 | 80 | 100 | µs |
| $BR_{FAST}$ | Fast Bit Rate (Programming mode) | - | - | 100 | kBit/s |
| $t_{REC_PD}$ | Propagation Delay of Receiver, $t_{REC_PD}$ = MAX ($t_{REC_PDR}$, $t_{REC_PDF}$) | - | - | 6.0 | µs |
| $t_{REC_SYM}$ | Symmetry of Receiver Propagation Delay, $t_{REC_PDF}$ - $t_{REC_PDR}$ | -2.0 | - | 2.0 | µs |

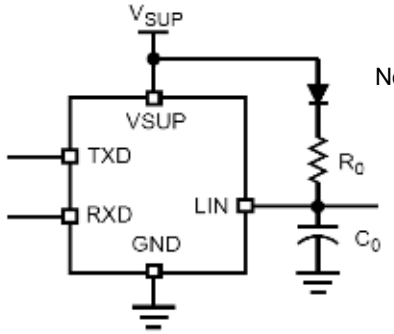
LIN Driver - 20.0 kBit/s; Bus load conditions ($C_{BUS}$; $R_{BUS}$): 1.0 nF; 1.0 kΩ / 6,8 nF;660 Ω / 10 nF;500 Ω. Measurement thresholds: 50% of TXD signal to LIN signal threshold defined at each parameter. See Figure 5 and Figure 6.

| | | | | | |
|---|---|---|---|---|---|
| D1 | Duty Cycle 1:<br>$TH_{REC(MAX)} = 0.744 \times V_{SUP}$<br>$TH_{DOM(MAX)} = 0.581 \times V_{SUP}$<br>$7.0\ V \leq V_{SUP} \leq 18\ V$; $t_{Bit} = 50\ \mu s$;<br>$D1 = t_{BUS_REC(MIN)}/(2 \times t_{Bit})$ | 0.396 | - | - | |
| D2 | Duty Cycle 2:<br>$TH_{REC(MIN)} = 0.422 \times V_{SUP}$<br>$TH_{DOM(MIN)} = 0.284 \times V_{SUP}$<br>$7.6\ V \leq V_{SUP} \leq 18\ V$; $t_{BIT} = 50\ \mu s$<br>$D2 = t_{BUS_REC(MAX)}/(2 \times t_{BIT})$ | - | - | 0.581 | |

LIN Driver - 10.0 kBit/s; Bus load conditions ($C_{BUS}$; $R_{BUS}$): 1.0 nF; 1.0 kΩ / 6,8 nF;660 Ω / 10 nF;500 Ω. Measurement thresholds: 50% of TXD signal to LIN signal threshold defined at each parameter. See Figure 5 and Figure 7.

**Table 35. Dynamic Electrical Characteristics - LIN Physical Layer Interface - LIN**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| D3 | Duty Cycle 3:<br>$TH_{REC(MAX)} = 0.778 \times V_{SUP}$<br>$TH_{DOM(MAX)} = 0.616 \times V_{SUP}$<br>$7.0\ V \le V_{SUP} \le 18\ V$; $t_{BIT} = 96\ \mu s$<br>$D3 = T_{BUS_REC(MIN)}/(2 \times t_{BIT})$ | 0.417 | - | - | |
| D4 | Duty Cycle 4:<br>$TH_{REC(MIN)} = 0.389 \times V_{SUP}$<br>$TH_{DOM(MIN)} = 0.251 \times V_{SUP}$<br>$7.6\ V \le V_{SUP} \le 18\ V$; $t_{BIT} = 96\ \mu s$<br>$D4 = t_{BUS_REC(MAX)}/(2 \times t_{BIT})$ | - | - | 0.590 | |
| LIN Transmitter Timing, ($V_{SUP}$ from 7.0 to 18 V) - See Figure 9 | | | | | |
| $t_{TRAN_SYM}$ | Transmitter Symmetry<br>$t_{TRAN_SYM} < MAX\ (t_{TRAN_SYM}60\%, t_{TRAN_SYM}40\%)$<br>$t_{TRAN_SYM}60\% = t_{TRAN_PDF}60\% - t_{TRAN_PDR}60\%$<br>$t_{TRAN_SYM}40\% = t_{TRAN_PDF}40\% - t_{TRAN_PDR}40\%$ | -7.25 | 0 | 7.25 | µs |




Note: $R_0$ and $C_0$ = 1.0 kΩ/ 1.0 nF,
660 Ω/6.8 nF, a, d 500 Ω/10 nF

**Figure 5. Test Circuit for Timing Measurements**

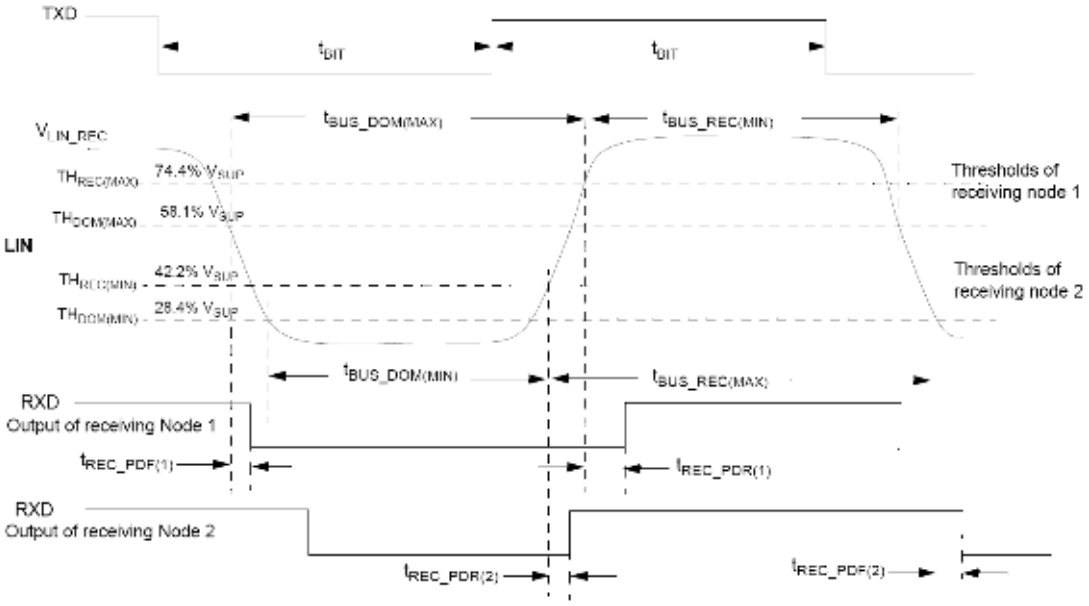



**Figure 6. LIN Timing Measurements for Normal Baud Rate**

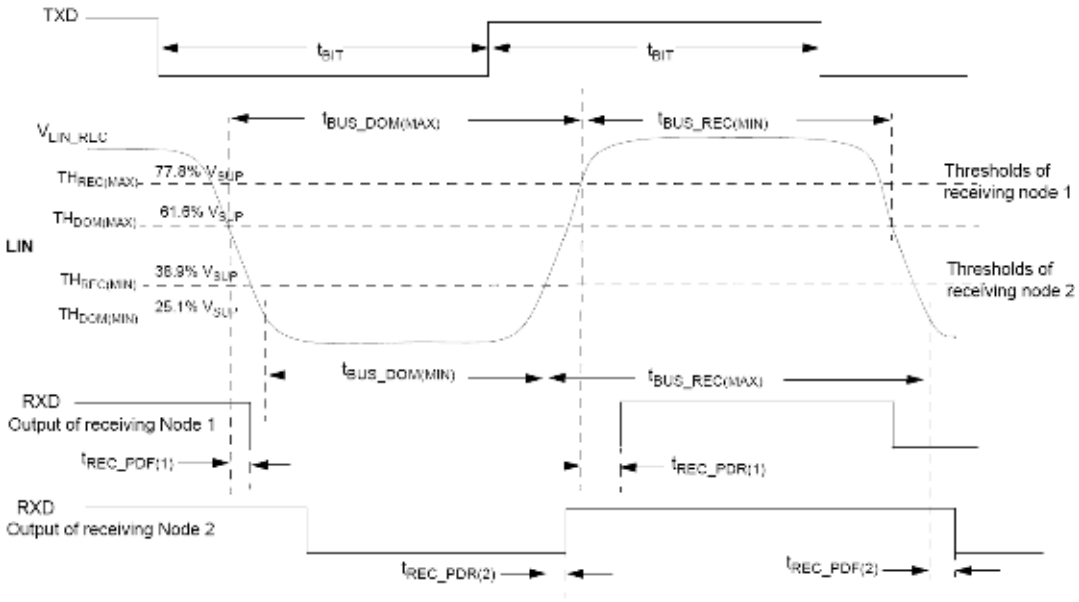



**Figure 7. LIN Timing Measurements for Slow Baud Rate**

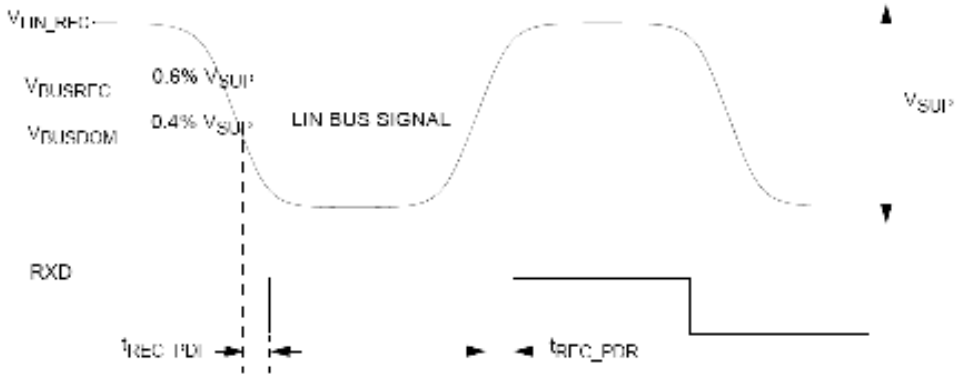



**Figure 8. LIN Receiver Timing**

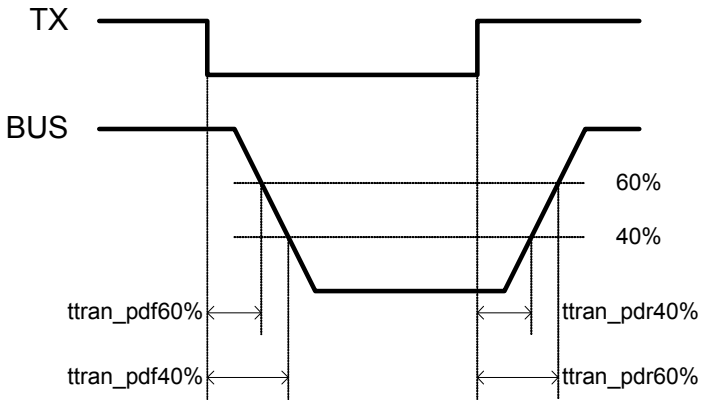



**Figure 9. LIN Transmitter Timing**

**Table 36. Dynamic Electrical Characteristics - General Purpose I/O - PTB[0…2][41]**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $f_{PTB}$ | GPIO Digital Frequency | - | - | 10 | MHz |
| $t_{PDR}$ | Propagation Delay - Rising Edge[42] | - | - | 20 | ns |
| $t_{RISE}$ | Rise Time - Rising Edge[41] | - | - | 17.5 | ns |
| $t_{PDF}$ | Propagation Delay - Falling Edge[41] | - | - | 20 | ns |
| $t_{FALL}$ | Rise Time - Falling Edge[41] | - | - | 17.5 | ns |

Note:
41. Guaranteed by design.
42. Load PTBx = 100 pF.

**Table 37. Dynamic Electrical Characteristics - Analog Digital Converter - ADC[43]**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $f_{ADC}$ | ADC Operating Frequency | 1.6 | 2.0 | 2.4 | MHz |
| $t_{CONV}$ | Conversion Time (from ACCR write to CC Flag) | | 26 | | clk |
| $f_{CH14}$ | Sample Frequency Channel 14 (Bandgap) | - | - | 2.5 | kHz |

Note:
43. Guaranteed by design.

## 4.6.2 Dynamic Electrical Characteristics MCU Die

### 4.6.2.1 NVM

#### 4.6.2.1.1 Timing Parameters

The time base for all NVM program or erase operations is derived from the bus clock using the FCLKDIV register. The frequency of this derived clock must be set within the limits specified as $f_{NVMOP}$. The NVM module does not have any means to monitor the frequency and does not prevent program or erase operations at frequencies above or below the specified minimum. When attempting to program or erase the NVM module at a lower frequency, a full program or erase transition is not assured.

The following sections provide equations which can be used to determine the time required to execute specific flash commands. All timing parameters are a function of the bus clock frequency, $f_{NVMBUS}$. All program and erase times are also a function of the NVM operating frequency, $f_{NVMOP}$. A summary of key timing parameters can be found in Table 38.

##### 4.6.2.1.1.1 Erase Verify All Blocks (Blank Check) (FCMD=0x01)

The time required to perform a blank check on all blocks is dependent on the location of the first non-blank word starting at relative address zero. It takes one bus cycle per phrase to verify plus a setup of the command. Assuming no non-blank location is found, then the time to erase verify all blocks is given by:

$$t_{check} = 19200 \cdot \frac{1}{f_{NVMBUS}}$$

##### 4.6.2.1.1.2 Erase Verify Block (Blank Check) (FCMD=0x02)

The time required to perform a blank check is dependent on the location of the first non-blank word starting at relative address zero. It takes one bus cycle per phrase to verify plus a setup of the command.

Assuming no non-blank location is found, then the time to erase verify a P-Flash block is given by:

$$t_{pcheck} = 17200 \cdot \frac{1}{f_{NVMBUS}}$$

Assuming no non-blank location is found, then the time to erase verify a D-Flash block is given by:

$$t_{dcheck} = 2800 \cdot \frac{1}{f_{NVMBUS}}$$

#### 4.6.2.1.1.3 Erase Verify P-Flash Section (FCMD=0x03)

The maximum time to erase verify a section of P-Flash depends on the number of phrases being verified ($N_{VP}$) and is given by:

$$t \approx (450 + N_{VP}) \cdot \frac{1}{f_{NVMBUS}}$$

#### 4.6.2.1.1.4 Read Once (FCMD=0x04)

The maximum read once time is given by:

$$t = 400 \cdot \frac{1}{f_{NVMBUS}}$$

#### 4.6.2.1.1.5 Program P-Flash (FCMD=0x06)

The programming time for a single phrase of four P-Flash words and the two seven-bit ECC fields is dependent on the bus frequency, $f_{NVMBUS}$, as well as on the NVM operating frequency, $f_{NVMOP}$.

The typical phrase programming time is given by:

$$t_{ppgm} \approx 164 \cdot \frac{1}{f_{NVMOP}} + 2000 \cdot \frac{1}{f_{NVMBUS}}$$

The maximum phrase programming time is given by:

$$t_{ppgm} \approx 164 \cdot \frac{1}{f_{NVMOP}} + 2500 \cdot \frac{1}{f_{NVMBUS}}$$

#### 4.6.2.1.1.6 Program Once (FCMD=0x07)

The maximum time required to program a P-Flash Program Once field is given by:

$$t \approx 164 \cdot \frac{1}{f_{NVMOP}} + 2150 \cdot \frac{1}{f_{NVMBUS}}$$

#### 4.6.2.1.1.7 Erase All Blocks (FCMD=0x08)

The time required to erase all blocks is given by:

$$t_{mass} \approx 100100 \cdot \frac{1}{f_{NVMOP}} + 38000 \cdot \frac{1}{f_{NVMBUS}}$$

#### 4.6.2.1.1.8 Erase P-Flash Block (FCMD=0x09)

The time required to erase the P-Flash block is given by:

$$t_{pmass} \approx 100100 \cdot \frac{1}{f_{NVMOP}} + 35000 \cdot \frac{1}{f_{NVMBUS}}$$

#### 4.6.2.1.1.9 Erase P-Flash Sector (FCMD=0x0A)

The typical time to erase a 512-byte P-Flash sector is given by:

$$t_{pera} \approx 20020 \cdot \frac{1}{f_{NVMOP}} + 700 \cdot \frac{1}{f_{NVMBUS}}$$

The maximum time to erase a 512-byte P-Flash sector is given by:

$$t_{pera} \approx 20020 \cdot \frac{1}{f_{NVMOP}} + 1400 \cdot \frac{1}{f_{NVMBUS}}$$

#### 4.6.2.1.1.10 Unsecure Flash (FCMD=0x0B)

The maximum time required to erase and unsecure the Flash is given by:

$$t_{uns} \approx 100100 \cdot \frac{1}{f_{NVMOP}} + 38000 \cdot \frac{1}{f_{NVMBUS}}$$

#### 4.6.2.1.1.11 Verify Back door Access Key (FCMD=0x0C)

The maximum verify back door access key time is given by:

$$t = 400 \cdot \frac{1}{f_{NVMBUS}}$$

#### 4.6.2.1.1.12 Set User Margin Level (FCMD=0x0D)

The maximum set user margin level time is given by:

$$t = 350 \cdot \frac{1}{f_{NVMBUS}}$$

#### 4.6.2.1.1.13 Set Field Margin Level (FCMD=0x0E)

The maximum set field margin level time is given by:

$$t = 350 \cdot \frac{1}{f_{NVMBUS}}$$

#### 4.6.2.1.1.14 Erase Verify D-Flash Section (FCMD=0x10)

The time required to Erase Verify D-Flash for a given number of words $N_W$ is given by:

$$t_{dcheck} \approx (450 + N_W) \cdot \frac{1}{f_{NVMBUS}}$$

#### 4.6.2.1.1.15 Program D-Flash (FCMD=0x11)

D-Flash programming time is dependent on the number of words being programmed and their location with respect to a row boundary, since programming across a row boundary requires extra steps. The D-Flash programming time is specified for different cases: 1,2,3,4 words and 4 words across a row boundary.

The typical D-Flash programming time is given by the following equation, where $N_W$ denotes the number of words; BC=0 if no row boundary is crossed and BC=1 if a row boundary is crossed:

$$t_{dpgm} \approx \left((14 + (54 \cdot N_W) + (14 \cdot BC)) \cdot \frac{1}{f_{NVMOP}}\right) + \left((500 + (525 \cdot N_W) + (100 \cdot BC)) \cdot \frac{1}{f_{NVMBUS}}\right)$$

The maximum D-Flash programming time is given by:

$$t_{dpgm} \approx \left((14 + (54 \cdot N_W) + (14 \cdot BC)) \cdot \frac{1}{f_{NVMOP}}\right) + \left((500 + (750 \cdot N_W) + (100 \cdot BC)) \cdot \frac{1}{f_{NVMBUS}}\right)$$

#### 4.6.2.1.1.16 Erase D-Flash Sector (FCMD=0x12)

Typical D-Flash sector erase times, expected on a new device where no margin verify fails occur, is given by:

$$t_{dera} \approx 5025 \cdot \frac{1}{f_{NVMOP}} + 700 \cdot \frac{1}{f_{NVMBUS}}$$

Maximum D-Flash sector erase times is given by:

$$t_{dera} \approx 20100 \cdot \frac{1}{f_{NVMOP}} + 3400 \cdot \frac{1}{f_{NVMBUS}}$$

The D-Flash sector erase time is ~5.0 ms on a new device and can extend to ~20 ms as the flash is cycled.

**Table 38. NVM Timing Characteristics (FTMRC)**

| C | Rating | Symbol | Min | Typ[44] | Max[45] | Unit[46] |
|---|---|---|---|---|---|---|
| | Bus frequency[47] | $f_{NVMBUS}$ | 1 | — | 32 | MHz |
| | Operating frequency | $f_{NVMOP}$ | 0.8 | 1.0 | 1.05 | MHz |
| D | Erase all blocks (mass erase) time | $t_{mass}$ | — | 100 | 130 | ms |
| D | Erase verify all blocks (blank check) time | $t_{CHECK}$ | — | — | 19200 | $t_{CYC}$ |
| D | Unsecure Flash time | $t_{UNS}$ | — | 100 | 130 | ms |
| D | P-Flash block erase time | $t_{PMASS}$ | — | 100 | 130 | ms |
| D | P-Flash erase verify (blank check) time | $t_{PCHECK}$ | — | — | 17200 | $t_{CYC}$ |
| D | P-Flash sector erase time | $t_{PERA}$ | — | 20 | 26 | ms |
| D | P-Flash phrase programming time | $t_{PPGM}$ | — | 226 | 285 | µs |
| D | D-Flash sector erase time | $t_{DERA}$ | — | 5.0 [48] | 26 | ms |
| D | D-Flash erase verify (blank check) time | $t_{DCHECK}$ | — | — | 2800 | $t_{CYC}$ |
| D | D-Flash one word programming time | $t_{DPGM1}$ | — | 100 | 107 | µs |
| D | D-Flash two word programming time | $t_{DPGM2}$ | — | 170 | 185 | µs |
| D | D-Flash three word programming time | $t_{DPGM3}$ | — | 241 | 262 | µs |
| D | D-Flash four word programming time | $t_{DPGM4}$ | — | 311 | 339 | µs |
| D | D-Flash four word programming time crossing row boundary | $t_{DPGM4C}$ | — | 328 | 357 | µs |

Note:
44. Typical program and erase times are based on typical $f_{NVMOP}$ and maximum $f_{NVMBUS}$.
45. Maximum program and erase times are based on minimum $f_{NVMOP}$ and maximum $f_{NVMBUS}$.
46. $t_{CYC} = 1 / f_{NVMBUS}$
47. The maximum device bus clock is specified as fBUS.
48. Typical value for a new device.

#### 4.6.2.1.2 NVM Reliability Parameters

The reliability of the NVM blocks is guaranteed by stress test during qualification, constant process monitors, and burn-in to screen early life failures.

The data retention and program/erase cycling failure rates are specified at the operating conditions noted. The program/erase cycle count on the sector is incremented every time a sector or mass erase event is executed.

**NOTE**

All values shown in Table 39 are preliminary and subject to further characterization.

**Table 39. NVM Reliability Characteristics**

| Symbol | Rating | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| | **Program Flash Arrays** | | | | |
| $t_{NVMRET}$ | Data retention at an average junction temperature of $T_{JAVG}$ = 85 °C[49] after up to 10,000 program/erase cycles | 20 | 100[50] | — | Years |
| $n_{FLPE}$ | Program Flash number of program/erase cycles (-40 °C ≤ $T_J$ ≤ 150 °C) | 10K | 100K[51] | — | Cycles |
| | **Data Flash Array** | | | | |
| $t_{NVMRET}$ | Data retention at an average junction temperature of $T_{JAVG}$ = 85 °C[49] after up to 50,000 program/erase cycles | 5 | 100[50] | — | Years |
| $t_{NVMRET}$ | Data retention at an average junction temperature of $T_{JAVG}$ = 85 °C[49] after up to 10,000 program/erase cycles | 10 | 100[50] | — | Years |
| $t_{NVMRET}$ | Data retention at an average junction temperature of $T_{JAVG}$ = 85 °C[49] after less than 100 program/erase cycles | 20 | 100[50] | — | Years |
| $n_{FLPE}$ | Data Flash number of program/erase cycles (-40 °C ≤ $T_J$ ≤ 150 °C) | 50K | 500K[51] | — | Cycles |

Note:

49. $T_{JAVG}$ does not exceed 85 °C in a typical temperature profile over the lifetime of a consumer, industrial or automotive application.
50. Typical data retention values are based on intrinsic capability of the technology measured at high temperature and de-rated to 25 °C using the Arrhenius equation. For additional information on how Freescale defines Typical Data Retention, refer to Engineering Bulletin EB618
51. Spec table quotes typical endurance evaluated at 25 °C for this product family. For additional information on how Freescale defines Typical Endurance, refer to Engineering Bulletin EB619.

### 4.6.2.2 Phase Locked Loop

#### 4.6.2.2.1 Jitter Definitions

With each transition of the feedback clock, the deviation from the reference clock is measured and input voltage to the VCO is adjusted accordingly.The adjustment is done continuously with no abrupt changes in the VCOCLK frequency. Noise, voltage, temperature, and other factors cause slight variations in the control loop resulting in a clock jitter. This jitter affects the real minimum and maximum clock periods, as illustrated in Figure 10.

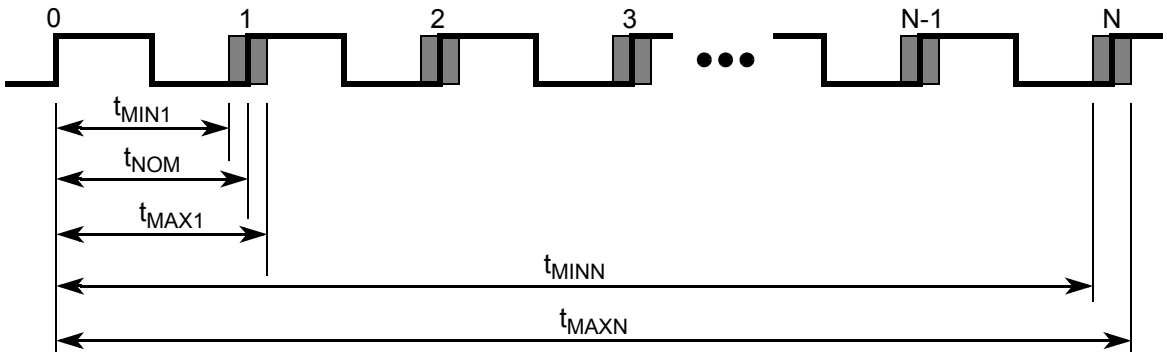



**Figure 10. Jitter Definitions**

The relative deviation of $t_{NOM}$ is at its maximum for one clock period, and decreases towards zero for larger number of clock periods (N). Jitter is defined as:

$$J(N) = max\left(\left|1 - \frac{t_{max}(N)}{N \cdot t_{nom}}\right|, \left|1 - \frac{t_{min}(N)}{N \cdot t_{nom}}\right|\right)$$

For N < 100, the following equation is a good fit for the maximum jitter:

$$J(N) = \frac{j_1}{\sqrt{N}}$$

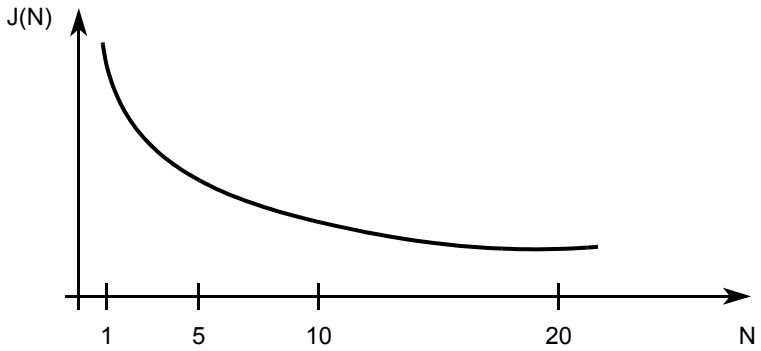



**Figure 11. Maximum Bus Clock Jitter Approximation**

**NOTE**

On timers and serial modules a prescaler eliminates the effect of the jitter to a large extent.

#### 4.6.2.2.2 Electrical Characteristics for the PLL[52]

**Table 40. PLL Characteristics**

| Symbol | Rating | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $f_{VCORST}$ | VCO Frequency During System Reset | 8.0 | — | 32 | MHz |
| $f_{VCO}$ | VCO Locking Range | 32 | — | 64 | MHz |
| $f_{REF}$ | Reference Clock | 1.0 | — | — | MHz |
| $\lvert\Delta_{LOCK}\rvert$ | Lock Detection | 0 | — | 1.5 | %[53] |
| $\lvert\Delta_{UNL}\rvert$ | Un-lock Detection | 0.5 | — | 2.5 | %[53] |
| $t_{LOCK}$ | Time to Lock | — | — | $150 + 256/f_{REF}$ | µs |
| $j_1$ | Jitter Fit Parameter 1[54] | — | — | 1.2 | % |

Note:
52. the maximum device bus clock is specified as $f_{BUS}$.
53. % deviation from target frequency.
54. $f_{REF}$ = 1.0 MHz, $f_{BUS}$ = 32 MHz equivalent $f_{PLL}$ = 64 MHz, REFRQ=00, SYNDIV=$1F, VCOFRQ=01, POSTDIV=$00.

### 4.6.2.3 Electrical Characteristics for the IRC1M

**Table 41. IRC1M Characteristics**

| Symbol | Rating | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $f_{IRC1M_TRIM}$ | Internal Reference Frequency, Factory Trimmed<br>-40 °C ≤ $T_J$ ≤ 150 °C | 0.987 | 1.0 | 1.013 | MHz |

#### 4.6.2.4 Electrical Characteristics for the Oscillator (OSCLCP)

**Table 42. OSCLCP Characteristics**

| Symbol | Rating | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $f_{OSC}$ | Crystal Oscillator Range | 4.0 | — | 16 | MHz |
| $i_{OSC}$ | Startup Current | 100 | — | — | μA |
| $t_{UPOSC}$ | Oscillator Start-up time (LCP, 4.0 MHz)(55) | — | 2.0 | 10 | ms |
| $t_{UPOSC}$ | Oscillator Start-up time (LCP, 8.0 MHz)(55) | — | 1.6 | 8.0 | ms |
| $t_{UPOSC}$ | Oscillator Start-up time (LCP, 16 MHz)(55) | — | 1.0 | 5.0 | ms |
| $f_{CMFA}$ | Clock Monitor Failure Assert Frequency | 200 | 450 | 1200 | KHz |
| $C_{IN}$ | Input Capacitance (EXTAL, XTAL pins) | — | 7.0 | — | pF |
| $V_{HYS,EXTAL}$ | EXTAL Pin Input Hysteresis | — | 120 | — | mV |
| $V_{PP,EXTAL}$ | EXTAL Pin Oscillation Amplitude (loop controlled Pierce) | — | 0.9 | — | V |

Note:
55. These values apply for carefully designed PCB layouts with capacitors which match the crystal/resonator requirements.
56. Only applies if EXTAL is externally driven.

#### 4.6.2.5 Reset Characteristics

**Table 43. Reset and Stop Characteristics**

| Symbol | Rating | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $PW_{RSTL}$ | Reset Input Pulse Width, Minimum Input Time | 2.0 | — | — | $t_{VCORST}$ |
| $n_{RST}$ | Startup from Reset | — | 768 | — | $t_{VCORST}$ |
| $t_{STP_REC}$ | STOP Recovery Time | — | 50 | — | μs |

#### 4.6.2.6 SPI Timing

This section provides electrical parametrics and ratings for the SPI. In Table 44 the measurement conditions are listed.

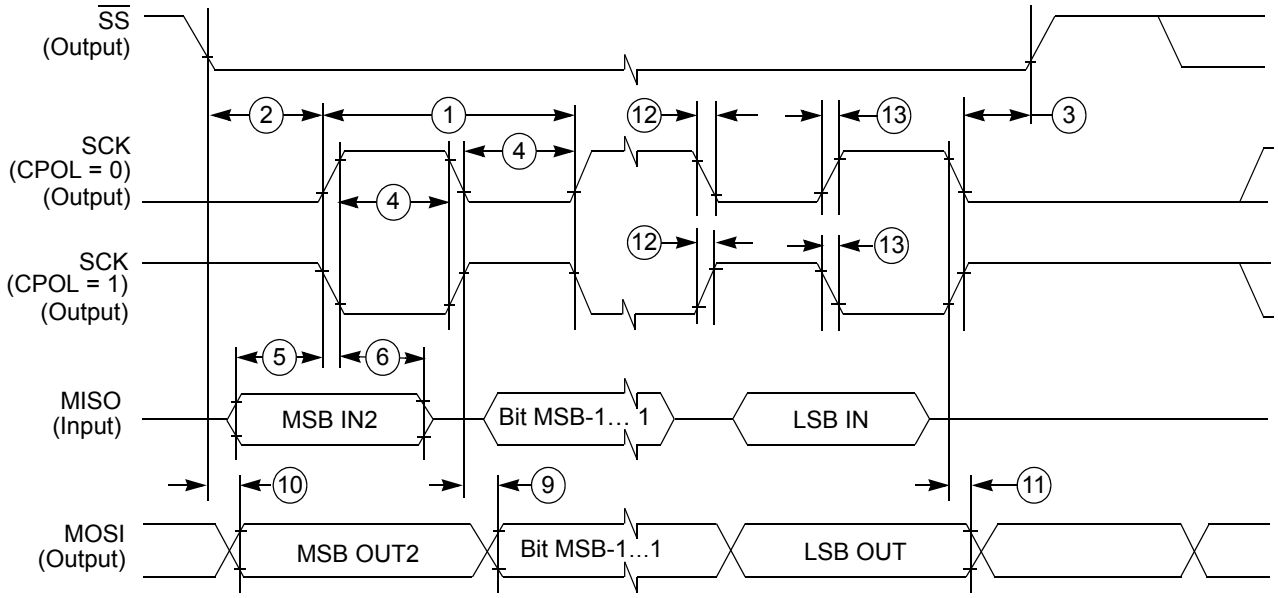
**Table 44. Measurement Conditions**

| Description | Value | Unit |
|---|---|---|
| Drive mode | Full drive mode | — |
| Load capacitance $C_{LOAD}$(57), on all outputs | 50 | pF |
| Thresholds for delay measurement points | (20% / 80%) $V_{DDRX}$ | V |

Note:
57. Timing specified for equal load on all SPI output pins. Avoid asymmetric load.

#### 4.6.2.6.1 Master Mode

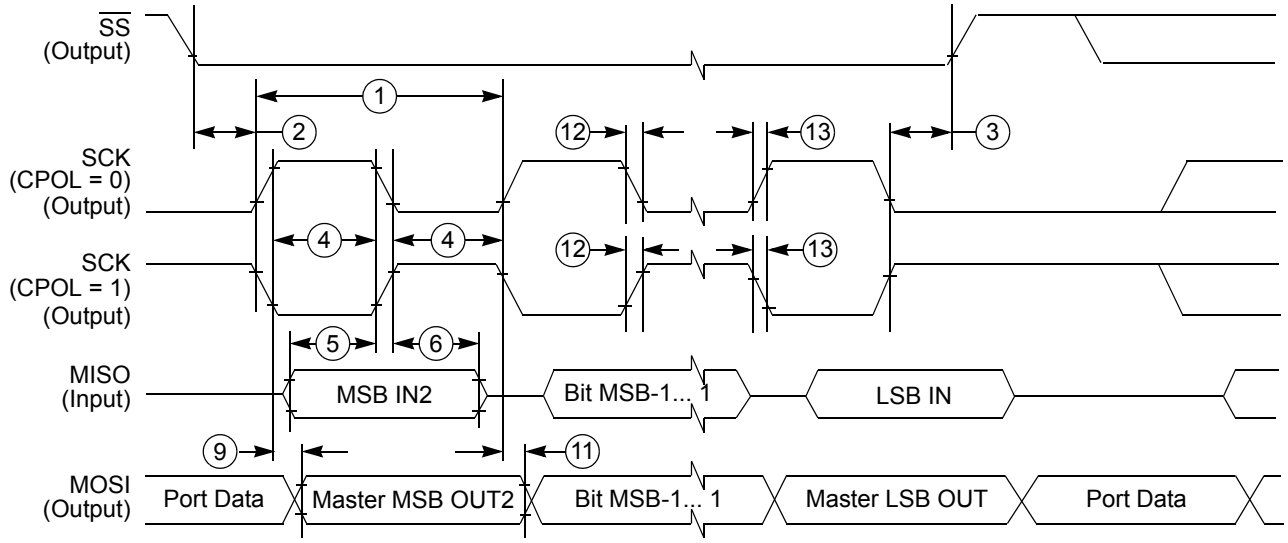
In Figure 12 the timing diagram for master mode with transmission format CPHA = 0 is depicted.




1. If configured as an output.
2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1, bit 2... MSB.

**Figure 12. SPI Master Timing (CPHA = 0)**

In Figure 13 the timing diagram for master mode with transmission format CPHA=1 is depicted.




1.If configured as output
2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1,bit 2... MSB.

**Figure 13. SPI Master Timing (CPHA = 1)**

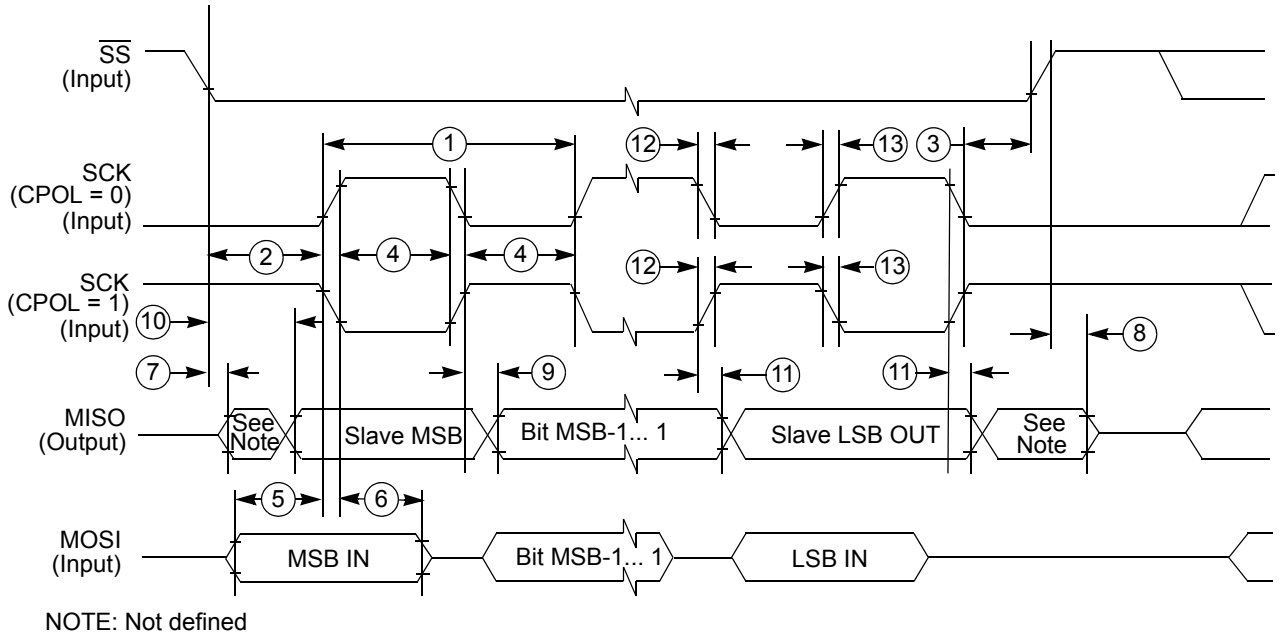
In Table 45 the timing characteristics for master mode are listed.

**Table 45. SPI Master Mode Timing Characteristics**

| Symbol | Characteristic | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $f_{SCK}$ | SCK Frequency | 1/2048 | — | 1/2 | $f_{BUS}$ |
| $t_{SCK}$ | SCK Period | 2.0 | — | 2048 | $t_{BUS}$ |
| $t_{LEAD}$ | Enable Lead Time | — | 1/2 | — | $t_{SCK}$ |
| $t_{LAG}$ | Enable Lag Time | — | 1/2 | — | $t_{SCK}$ |
| $t_{WSCK}$ | Clock (SCK) High or Low Time | — | 1/2 | — | $t_{SCK}$ |
| $t_{SU}$ | Data Setup Time (inputs) | 8.0 | — | — | ns |
| $t_{HI}$ | Data Hold Time (inputs) | 8.0 | — | — | ns |
| $t_{VSCK}$ | Data Valid After SCK Edge | — | — | 29 | ns |
| $t_{VSS}$ | Data Valid After $\overline{SS}$ Fall (CPHA = 0) | — | — | 15 | ns |
| $t_{HO}$ | Data Hold Time (outputs) | 20 | — | — | ns |
| $t_{RFI}$ | Rise and Fall Time Inputs | — | — | 8.0 | ns |
| $t_{RFO}$ | Rise and Fall Time Outputs | — | — | 8.0 | ns |

#### 4.6.2.6.2 Slave Mode

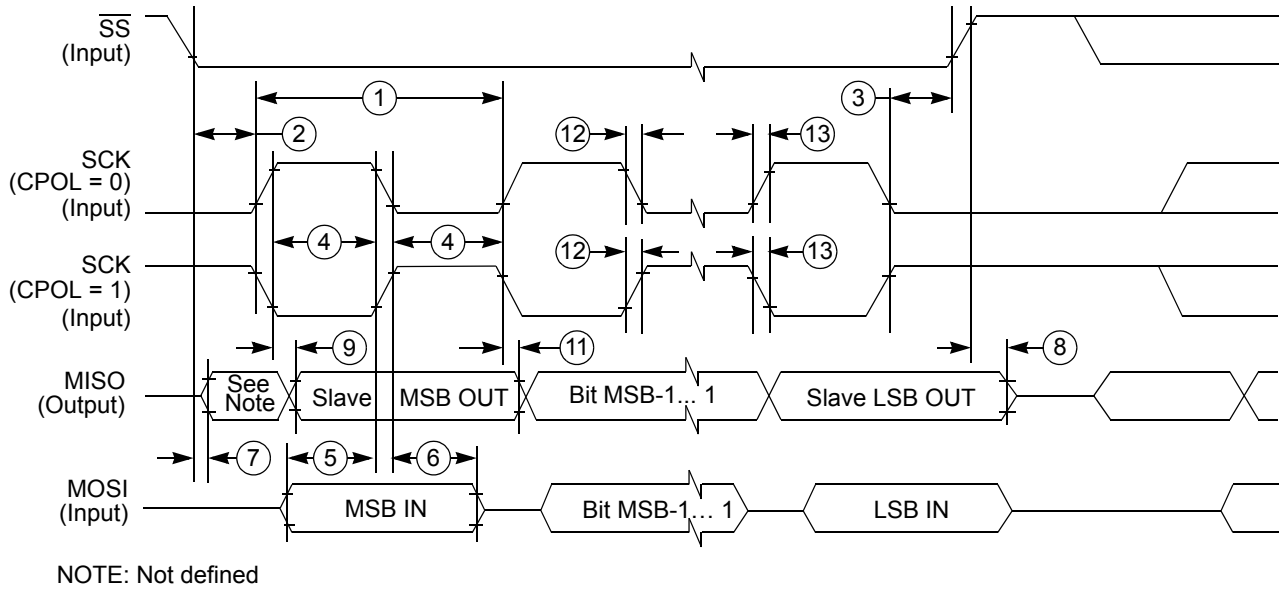
In Figure 14 the timing diagram for slave mode with transmission format CPHA = 0 is depicted.




NOTE: Not defined

**Figure 14. SPI Slave Timing (CPHA = 0)**

In Figure 15 the timing diagram for slave mode with transmission format CPHA = 1 is depicted.




NOTE: Not defined

**Figure 15. SPI Slave Timing (CPHA = 1)**

In Table 46 the timing characteristics for slave mode are listed.

**Table 46. SPI Slave Mode Timing Characteristics**

| Symbol | Characteristic | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $f_{SCK}$ | SCK Frequency | DC | — | 1/4 | $f_{BUS}$ |
| $t_{SCK}$ | SCK Period | 4 | — | ∞ | $t_{BUS}$ |
| $t_{LEAD}$ | Enable Lead Time | 4 | — | — | $t_{BUS}$ |
| $t_{LAG}$ | Enable Lag Time | 4.0 | — | — | $t_{BUS}$ |
| $t_{WSCK}$ | Clock (SCK) High or Low Time | 4.0 | — | — | $t_{BUS}$ |
| $t_{SU}$ | Data Setup Time (inputs) | 8.0 | — | — | ns |
| $t_{HI}$ | Data Hold Time (inputs) | 8.0 | — | — | ns |
| $t_A$ | Slave Access Time (time to data active) | — | — | 20 | ns |
| $t_{DIS}$ | Slave MISO Disable Time | — | — | 22 | ns |
| $t_{VSCK}$ | Data Valid After SCK Edge | — | — | 29 + 0.5 · $t_{BUS}$ (58) | ns |
| $t_{VSS}$ | Data Valid After $\overline{SS}$ Fall | — | — | 29 + 0.5 · $t_{BUS}$ (58) | ns |
| $t_{HO}$ | Data Hold Time (outputs) | 20 | — | — | ns |
| $t_{RFI}$ | Rise and Fall Time Inputs | — | — | 8.0 | ns |
| $t_{RFO}$ | Rise and Fall Time Outputs | — | — | 8.0 | ns |

Note:
58. 0.5 $t_{BUS}$ added due to internal synchronization delay

## 4.7 Thermal Protection Characteristics

Characteristics noted under conditions 5.5 V ≤ $V_{SUP}$ ≤ 18 V, -40 °C ≤ $T_A$ ≤ 125 °C, unless otherwise noted. Typical values noted reflect the approximate parameter mean at $T_A$ = 25 °C under nominal conditions, unless otherwise noted.

**Table 47. Thermal Characteristics - Voltage Regulators VDD (2.5 V) & VDDX (5.0 V)[59]**

| Symbol | Ratings | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $T_{HTI}$<br>$T_{HTI_H}$ | VDD/VDDX High-temperature Warning (HTI)<br>Threshold<br>Hysteresis | <br>110<br>- | <br>125<br>10 | <br>140<br>- | °C |
| $T_{SD}$<br>$T_{SD_H}$ | VDD/VDDX Overtemperature Shutdown<br>Threshold<br>Hysteresis | <br>155<br>- | <br>170<br>10 | <br>185<br>- | °C |
| $T_{HSUPSD}$ | HSUP Overtemperature Shutdown | 150 | 165 | 180 | °C |
| $T_{HSUPSD_HYS}$ | HSUP Overtemperature Shutdown Hysteresis | - | 10 | - | °C |
| $T_{HSSD}$ | HS Overtemperature Shutdown | 150 | 165 | 180 | °C |
| $T_{HSSD_HYS}$ | HS Overtemperature Shutdown Hysteresis | - | 10 | - | °C |
| $T_{LSSD}$ | LS Overtemperature Shutdown | 150 | 165 | 180 | °C |
| $T_{LSSD_HYS}$ | LS Overtemperature Shutdown Hysteresis | - | 10 | - | °C |
| $T_{LINSD}$ | LIN Overtemperature Shutdown | 150 | 165 | 200 | °C |
| $T_{LINSD_HYS}$ | LIN Overtemperature Shutdown Hysteresis | - | 20 | - | °C |

Note:
59. Guaranteed by characterization. Functionality tested.

## 4.8 ESD Protection and Latch-up Immunity

All ESD testing is in conformity with CDF-AEC-Q100 stress test qualification for automotive grade integrated circuits. During the device qualification, ESD stresses were performed for the Human Body Model (HBM), Machine Model (MM), Charge Device Model (CDM), as well as LIN transceiver specific specifications.

A device is defined as a failure if after exposure to ESD pulses, the device no longer meets the device specification. Complete DC parametric and functional testing is performed per the applicable device specification at room temperature, followed by hot temperature, unless specified otherwise in the device specification.

**Table 48. ESD and Latch-up Protection Characteristics**

| Symbol | Ratings | Value | Unit |
|---|---|---|---|
| $V_{HBM}$ | ESD - Human Body Model (HBM) following AEC-Q100 / JESD22-A114<br>($C_{ZAP}$ = 100 pF, $R_{ZAP}$ = 1500 Ω)<br>- LIN (DGND, PGND, AGND, and LGND shorted)<br>- VS1, VS2, VSENSE, Lx<br>- HSx<br>- All other Pins | <br><br>±8000<br>±4000<br>±3000<br>±2000 | V |
| $V_{CDM}$ | ESD - Charged Device Model (CDM) following AEC-Q100,<br>Corner Pins (1, 12, 13, 24, 25, 36, 37, and 48)<br>All other Pins | <br>±750<br>±500 | V |
| $V_{MM}$ | ESD - Machine Model (MM) following AEC-Q100<br>($C_{ZAP}$ = 200 pF, $R_{ZAP}$ = 0 Ω), All Pins | ±200 | V |
| $I_{LAT}$ | Latch-up current at $T_A$ = 125 °C[60] | ±100 | mA |

**Table 48. ESD and Latch-up Protection Characteristics (continued)**

| Symbol | Ratings | Value | Unit |
|---|---|---|---|
| | ESD GUN - LIN Conformance Test Specification[62], unpowered, contact discharge, $C_{ZAP}$= 150 pF, $R_{ZAP}$ = 330 Ω.<br>- LIN (with or without bus filter $C_{BUS}$=220 pF)<br>- VS1, VS2 with $C_{VS}$<br>- Lx with serial $R_{LX}$ | <br>±15000<br>±20000<br>±6000 | V |
| | ESD GUN - following IEC 61000-4-2 Test Specification[63], unpowered, contact discharge, $C_{ZAP}$= 150 pF, $R_{ZAP}$ = 330 Ω<br>- LIN (with or without bus filter $C_{BUS}$=220 pF)<br>- VSENSE with serial $R_{VSENSE}$[61]<br>- VS1, VS2 with $C_{VS}$<br>- Lx with serial $R_{LX}$ | <br>±8000<br>±8000<br>±8000<br>±8000 | V |
| | ESD GUN - following ISO10605 Test Specification[63], unpowered, contact discharge, $C_{ZAP}$= 150 pF, $R_{ZAP}$ = 2.0 kΩ<br>- LIN (with or without bus filter $C_{BUS}$=220pF)<br>- VSENSE with serial $R_{VSENSE}$[61]<br>- VS1, VS2 with $C_{VS}$<br>- Lx with serial $R_{LX}$ | <br>±6000<br>±6000<br>±6000<br>±6000 | V |
| | ESD GUN - following ISO10605 Test Specification[63], powered, contact discharge, $C_{ZAP}$= 330 pF, $R_{ZAP}$ = 2.0 kΩ<br>- LIN (with or without bus filter $C_{BUS}$=220 pF)<br>- VSENSE with serial $R_{VSENSE}$[61]<br>- VS1, VS2 with $C_{VS}$<br>- Lx with serial $R_{LX}$ | <br>±8000<br>±8000<br>±8000<br>±8000 | V |

Note:

60. Input Voltage Limit = -2.5 to 7.5 V.
61. With $C_{VBAT}$ (10…100 nF) as part of the battery path.
62. Certification available on request
63. Tested internally only; certification pending

## 4.9 Additional Test Information ISO7637-2

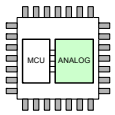
Immunity against transients for the LIN, Lx, and VBAT, is specified according to the LIN Conformance Test Specification - Section LIN EMC Test Specification refer to the LIN Conformance Test Certification Report - available as separate document.

# 5 Functional Description and Application Information

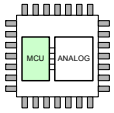
## 5.1 Introduction

This chapter describes the MM912_634 dual die device functions on a block by block base. To distinguish between the module location being the MCU die or the analog die, the following symbols are shown on all module cover pages:

The documented module is physically located on the Analog die. This applies to Section 5.3, “MM912_634 - Analog Die Overview" through Section 5.26, “MM912_634 - Analog Die Trimming".




The documented module is physically located on the Microcontroller die. This applies to Section 5.27, “MM912_634 - MCU Die Overview" through Section 5.39, “Serial Peripheral Interface (S12SPIV5)".




Sections concerning both dies or the complete device do not have a specific indication.

## 5.2 Device Register Maps

Table 49 shows the device register memory map overview for the 64 kByte MCU die (MC9S12I64).

**Table 49. Device Register Memory Map Overview**

| 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 |
| 0x0010–0x0015 | MMC (memory map control) | 8 |
| 0x0016–0x0019 | Reserved | 2 |
| 0x001A–0x001B | Device ID register | 2 |
| 0x001C–0x001E | Reserved | 4 |
| 0x001F | INT (interrupt module) | 1 |
| 0x0020–0x002F | DBG (debug module) | 16 |
| 0x0030–0x0033 | Reserved | 4 |
| 0x0034–0x003F | CPMU (clock and power management) | 12 |
| 0x0040–0x00D7 | Reserved | 152 |
| 0x00D8–0x00DF | D2DI (die 2 die initiator) | 8 |
| 0x00E0–0x00E7 | Reserved | 32 |
| 0x00E8–0x00EF | SPI (serial peripheral interface) | 8 |
| 0x00F0–0x00FF | Reserved | 32 |
| 0x0100–0x0113 | FTMRC control registers | 20 |
| 0x0114–0x011F | Reserved | 12 |
| 0x0120–0x017F | PIM (port integration module) | 96 |
| 0x0180–0x01EF | Reserved | 112 |
| 0x01F0–0x01FC | CPMU (clock and power management) | 13 |

**Table 49. Device Register Memory Map Overview (continued)**

| Address | Module | Size (Bytes) |
|---|---|---|
| 0x01FD–0x01FF | Reserved | 3 |
| 0x0200-0x02FF | D2DI (die 2 die initiator, blocking access window) | 256 |
| 0x0300–0x03FF | D2DI (die 2 die initiator, non-blocking write window) | 256 |

**NOTE**

Reserved register space shown in Table 49 is not allocated to any module. This register space is reserved for future use, and shows as grayed areas in tables throughout this document. Writing to these locations has no effect. Read access to these locations returns zero.

### 5.2.1 Detailed Module Register Maps

Table 50 to Table 72 show the detailed module maps of the 9S12I64 MCU die.

**Table 50. 0x0000–0x0007 Port Integration Module (PIM) Map 1 of 3**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0000 | PORTA | R | PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| | | W | | | | | | | | |
| 0x0001 | PORTE | R | 0 | 0 | 0 | 0 | 0 | 0 | PE1 | PE0 |
| | | W | | | | | | | | |
| 0x0002 | DDRA | R | DDRA7 | DDRA6 | DDRA5 | DDRA4 | DDRA3 | DDRA2 | DDRA1 | DDRA0 |
| | | W | | | | | | | | |
| 0x0003 | DDRE | R | 0 | 0 | 0 | 0 | 0 | 0 | DDRE1 | DDRE0 |
| | | W | | | | | | | | |
| 0x0004 | PTC | R | 0 | 0 | 0 | 0 | 0 | 0 | PTC1 | PTC0 |
| | | W | | | | | | | | |
| 0x0005 | PTD | R | PTD7 | PTD6 | PTD5 | PTD4 | PTD3 | PTD2 | PTD1 | PTD0 |
| | | W | | | | | | | | |
| 0x0006 | DDRC | R | 0 | 0 | 0 | 0 | 0 | 0 | DDRC1 | DDRC0 |
| | | W | | | | | | | | |
| 0x0007 | DDRD | R | DDRD7 | DDRD6 | DDRD5 | DDRD4 | DDRD3 | DDRD2 | DDRD1 | DDRD0 |
| | | W | | | | | | | | |
| 0x0008 - 0x0009 | Reserved | R | | | | | | | | |
| | | W | | | | | | | | |

**Table 51. 0x000A–0x000B Memory Map Control (MMC) Map 1 of 2**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x000A | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x000B | MODE | R | MODC | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

**Table 52. 0x000C–0x000D Port Integration Module (PIM) Map 2 of 3**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x000C | PUCR | R | 0 | BKPUE | 0 | 0 | 0 | 0 | PDPEE | 0 |
| | | W | | | | | | | | |
| 0x000D | RDRIV | R | 0 | 0 | 0 | 0 | RDRD | RDRC | 0 | 0 |
| | | W | | | | | | | | |

**Table 53. 0x000E–0x000F Reserved**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x000E-<br>0x000F | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

**Table 54. 0x0010–0x001B Memory Map Control (MMC) Map 2 of 2**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0010 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0011 | DIRECT | R | DP15 | DP14 | DP13 | DP12 | DP11 | DP10 | DP9 | DP8 |
| | | W | | | | | | | | |
| 0x0012 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0013 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0014 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0015 | PPAGE | R | 0 | 0 | 0 | 0 | PIX3 | PIX2 | PIX1 | PIX0 |
| | | W | | | | | | | | |

**Table 55. 0x0016–0x0019 Reserved**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0016-0x0019 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

**Table 56. 0x001A–0x001B Device ID Register (PARTIDH/PARTIDL)**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x001A | PARTIDH | R | PARTIDH | | | | | | | |
| | | W | | | | | | | | |
| 0x001B | PARTIDL | R | PARTIDL | | | | | | | |
| | | W | | | | | | | | |

**Table 57. 0x001C–0x001E Reserved**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x001C-0x001E | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

**Table 58. 0x001F Interrupt Module (INT)**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x001F | IVBR | R | IVB_ADDR[7:0] | | | | | | | |
| | | W | | | | | | | | |

**Table 59. 0x0020–0x002F Debug Module (DBG)**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0020 | DBGC1 | R | ARM | 0 | 0 | BDM | DBGBRK | 0 | COMRV | |
| | | W | | TRIG | | | | | | |
| 0x0021 | DBGSR | R | TBF | 0 | 0 | 0 | 0 | SSF2 | SSF1 | SSF0 |
| | | W | | | | | | | | |
| 0x0022 | DBGTCR | R | 0 | TSOURCE | 0 | 0 | TRCMOD | | 0 | TALIGN |
| | | W | | | | | | | | |
| 0x0023 | DBGC2 | R | 0 | 0 | 0 | 0 | 0 | 0 | ABCM | |
| | | W | | | | | | | | |
| 0x0024 | DBGTBH | R | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | | W | | | | | | | | |
| 0x0025 | DBGTBL | R | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | W | | | | | | | | |

**Table 59. 0x0020–0x002F Debug Module (DBG) (continued)**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0026 | DBGCNT | R | TBF | 0 | CNT | | | | | |
| | | W | | | | | | | | |
| 0x0027 | DBGSCRX | R | 0 | 0 | 0 | 0 | SC3 | SC2 | SC1 | SC0 |
| | | W | | | | | | | | |
| | DBGMFR | R | 0 | 0 | 0 | 0 | 0 | MC2 | MC1 | MC0 |
| | | W | | | | | | | | |
| | DBGACTL | R | SZE | SZ | TAG | BRK | RW | RWE | NDB | COMPE |
| | | W | | | | | | | | |
| 0x0028 | DBGBCTL | R | SZE | SZ | TAG | BRK | RW | RWE | 0 | COMPE |
| | | W | | | | | | | | |
| | DBGCCTL | R | 0 | 0 | TAG | BRK | RW | RWE | 0 | COMPE |
| | | W | | | | | | | | |
| 0x0029 | DBGXAH | R | 0 | 0 | 0 | 0 | 0 | 0 | Bit 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 | | | | | | | | |

**Table 60. 0x0030–0x033 Reserved**

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

**Table 61. 0x0034–0x003F Clock and Power Management (CPMU) Map 1 of 2**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0034 | CPMU SYNR | R | VCOFRQ[1:0] | | SYNDIV[5:0] | | | | | |
| | | W | | | | | | | | |
| 0x0035 | CPMU REFDIV | R | REFFRQ[1:0] | | 0 | 0 | REFDIV[3:0] | | | |
| | | W | | | | | | | | |
| 0x0036 | CPMU POSTDIV | R | 0 | 0 | 0 | POSTDIV[4:0] | | | | |
| | | W | | | | | | | | |
| 0x0037 | CPMUFLG | R | RTIF | PORF | LVRF | LOCKIF | LOCK | ILAF | OSCIF | UPOSC |
| | | W | | | | | | | | |
| 0x0038 | CPMUINT | R | RTIE | 0 | 0 | LOCKIE | 0 | 0 | OSCIE | 0 |
| | | W | | | | | | | | |
| 0x0039 | CPMUCLKS | R | PLLSEL | PSTP | 0 | 0 | PRE | PCE | RTI OSCSEL | COP OSCSEL |
| | | W | | | | | | | | |
| 0x003A | CPMUPLL | R | 0 | 0 | FM1 | FM0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x003B | CPMURTI | R | RTDEC | RTR6 | RTR5 | RTR4 | RTR3 | RTR2 | RTR1 | RTR0 |
| | | W | | | | | | | | |
| 0x003C | CPMUCOP | R | WCOP | RSBCK | 0 | 0 | 0 | CR2 | CR1 | CR0 |
| | | W | | | WRTMASK | | | | | |
| 0x003D | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x003E | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x003F | CPMU 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 |

**Table 62. 0x0040–0x0D7 Reserved**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0040-0x00D7 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

**Table 63. 0x00D8–0x00DF Die 2 Die Initiator (D2DI) Map 1 of 3**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00D8 | D2DCTL0 | R | D2DEN | D2DCW | D2DSWAI | 0 | 0 | 0 | D2DCLKDIV[1:0] | |
| | | W | | | | | | | | |
| 0x00D9 | D2DCTL1 | R | D2DIE | 0 | 0 | 0 | TIMEOUT[3:0] | | | |
| | | W | | | | | | | | |
| 0x00DA | D2DSTAT0 | R | ERRIF | ACKERF | CNCLF | TIMEF | TERRF | PARF | PAR1 | PAR0 |
| | | W | | | | | | | | |
| 0x00DB | D2DSTAT1 | R | D2DIF | D2DBSY | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x00DC | D2DADRHI | R | RWB | SZ8 | 0 | NBLK | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x00DD | D2DADRLO | R | ADR[7:0] | | | | | | | |
| | | W | | | | | | | | |
| 0x00DE | D2DDATAHI | R | DATA[15:8] | | | | | | | |
| | | W | | | | | | | | |
| 0x00DF | D2DDATALO | R | DATA[7:0] | | | | | | | |
| | | W | | | | | | | | |

**Table 64. 0x00E0–0x0E7 Reserved**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00E0-0x00E7 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

**Table 65. 0x00E8–0x00EF Serial Peripheral Interface (SPI)**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00E8 | SPICR1 | R | SPIE | SPE | SPTIE | MSTR | CPOL | CPHA | SSOE | LSBFE |
| | | W | | | | | | | | |
| 0x00E9 | SPICR2 | R | 0 | XFRW | 0 | MODFEN | BIDIROE | 0 | SPISWAI | SPC0 |
| | | W | | | | | | | | |
| 0x00EA | SPIBR | R | 0 | SPPR2 | SPPR1 | SPPR0 | 0 | SPR2 | SPR1 | SPR0 |
| | | W | | | | | | | | |
| 0x00EB | SPISR | R | SPIF | 0 | SPTEF | MODF | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x00EC | SPIDRH | R | R15 | R14 | R13 | R12 | R11 | R10 | R9 | R8 |
| | | W | T15 | T14 | T13 | T12 | T11 | T10 | T9 | T8 |

**Table 65. 0x00E8–0x00EF Serial Peripheral Interface (SPI) (continued)**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0x00ED | SPIDRL | R | R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 |
| | | W | T7 | T6 | T5 | T4 | T3 | T2 | T1 | T0 |
| 0x00EE | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x00EF | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

**Table 66. 0x00F0–0x0FF Reserved**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00E0-0x00FF | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

**Table 67. 0x0100–0x011F Flash Module (FTMRC)**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0100 | FCLKDIV | R | FDIVLD | FDIVLCK | FDIV5 | FDIV4 | FDIV3 | FDIV2 | FDIV1 | FDIV0 |
| | | W | | | | | | | | |
| 0x0101 | FSEC | R | KEYEN1 | KEYEN0 | RNV5 | RNV4 | RNV3 | RNV2 | SEC1 | SEC0 |
| | | W | | | | | | | | |
| 0x0102 | FCCOBIX | R | 0 | 0 | 0 | 0 | 0 | CCOBIX2 | CCOBIX1 | CCOBIX0 |
| | | W | | | | | | | | |
| 0x0103 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0104 | FCNFG | R | CCIE | 0 | 0 | IGNSF | 0 | 0 | FDFD | FSFD |
| | | W | | | | | | | | |
| 0x0105 | FERCNFG | R | 0 | 0 | 0 | 0 | 0 | 0 | DFDIE | SFDIE |
| | | W | | | | | | | | |
| 0x0106 | FSTAT | R | CCIF | 0 | ACCERR | FPVIOL | MGBUSY | RSVD | MGSTAT1 | MGSTAT0 |
| | | W | | | | | | | | |
| 0x0107 | FERSTAT | R | 0 | 0 | 0 | 0 | 0 | 0 | DFDIF | SFDIF |
| | | W | | | | | | | | |
| 0x0108 | FPROT | R | FPOPEN | RNV6 | FPHDIS | FPHS1 | FPHS0 | FPLDIS | FPLS1 | FPLS0 |
| | | W | | | | | | | | |
| 0x0109 | DFPROT | R | DPOPEN | 0 | 0 | 0 | DPS3 | DPS2 | DPS1 | DPS0 |
| | | W | | | | | | | | |
| 0x010A | FCCOBHI | R | CCOB15 | CCOB14 | CCOB13 | CCOB12 | CCOB11 | CCOB10 | CCOB9 | CCOB8 |
| | | W | | | | | | | | |

**Table 67. 0x0100–0x011F Flash Module (FTMRC) (continued)**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x010B | FCCOBLO | R | CCOB7 | CCOB6 | CCOB5 | CCOB4 | CCOB3 | CCOB2 | CCOB1 | CCOB0 |
| | | W | | | | | | | | |
| 0x010C | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x010D | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x010E | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x010F | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0110 | FOPT | R | NV7 | NV6 | NV5 | NV4 | NV3 | NV2 | NV1 | NV0 |
| | | W | | | | | | | | |
| 0x0111 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0112 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0113 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

**Table 68. 0x0120 Port Integration Module (PIM) Map 3 of 3**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0120 | PTIA | R | PTIA7 | PTIA6 | PTIA5 | PTIA4 | PTIA3 | PTIA2 | PTIA1 | PTIA0 |
| | | W | | | | | | | | |
| 0x0121 | PTIE | R | 0 | 0 | 0 | 0 | 0 | 0 | PTIE1 | PTIE0 |
| | | W | | | | | | | | |
| 0x0122-0x017F | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

**Table 69. 0x0180–0x1EF Reserved**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0180-0x01EF | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

**Table 70. 0x01F0–0x01FF Clock and Power Management (CPMU) Map 2 of 2**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x01F0 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x01F1 | CPMU LVCTL | R | 0 | 0 | 0 | 0 | 0 | LVDS | LVIE | LVIF |
| | | W | | | | | | | | |
| 0x01F6 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x01F7 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x01F8 | CPMU IRCTRIMH | R | TCTRIM[3:0] | | | | 0 | 0 | IRCTRIM[9:8] | |
| | | W | | | | | | | | |
| 0x01F9 | CPMU IRCTRIML | R | IRCTRIM[7:0] | | | | | | | |
| | | W | | | | | | | | |
| 0x01FA | CPMUOSC | R | OSCE | OSCBW | OSCPINS_EN | OSCFILT[4:0] | | | | |
| | | W | | | | | | | | |
| 0x01FB | CPMUPROT | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PROT |
| | | W | | | | | | | | |
| 0x01FC | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

**Table 71. 0x01FD–0x1FF Reserved**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x01FD-0x01FF | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

**Table 72. 0x0200–0x03FF Die-To-Die Initiator Blocking and Non-Blocking Access Window**

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0200-0x02FF | Blocking Access Window | R | | | | | | | | |
| | | W | | | | | | | | |
| 0x0300-0x03FF | Non-Blocking Access Window | R | | | | | | | | |
| | | W | | | | | | | | |

Table 73 shows the detailed module maps of the MM912_634 analog die.

**Table 73. Analog die Registers[64] - 0x0200–0x02FF D2D Blocking Access (D2DI) 2 of 3/ 0x0300–0x03FF D2D Non Blocking Access (D2DI) 3 of 3**

| Offset | Name | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | ISR (hi) | R | 0 | 0 | HOT | LSOT | HSOT | LINOT | SCI | RX |
| | Interrupt Source Register | W | | | | | | | | |
| 0x01 | ISR (lo) | R | TX | ERR | TOV | CH3 | CH2 | CH1 | CH0 | VSI |
| | Interrupt Source Register | W | | | | | | | | |
| 0x02 | IVR | R | 0 | 0 | IRQ | | | | | |
| | Interrupt Vector Register | W | | | | | | | | |
| 0x04 | VCR | R | 0 | 0 | 0 | VROVIE | HTIE | HVIE | LVIE | LBIE |
| | Voltage Control Register | W | | | | | | | | |
| 0x05 | VSR | R | 0 | 0 | 0 | VROVC | HTC | HVC | LVC | LBC |
| | Voltage Status Register | W | | | | | | | | |
| 0x08 | LXR | R | 0 | 0 | L5 | L4 | L3 | L2 | L1 | L0 |
| | Lx Status Register | W | | | | | | | | |
| 0x09 | LXCR | R | 0 | 0 | L5DS | L4DS | L3DS | L2DS | L1DS | L0DS |
| | Lx Control Register | W | | | | | | | | |
| 0x10 | WDR | R | WDOFF | WDWO | 0 | 0 | 0 | WDTO | | |
| | Watchdog Register | W | | | | | | | | |
| 0x11 | WDSR | R | WDSR | | | | | | | |
| | Watchdog Service Register | W | | | | | | | | |
| 0x12 | WCR | R | CSSEL | | L5WE | L4WE | L3WE | L2WE | L1WE | L0WE |
| | Wake Up Control Register | W | | | | | | | | |
| 0x13 | TCR | R | FWM | | | | CST | | | |
| | Timing Control Register | W | | | | | | | | |
| 0x14 | WSR | R | FWU | LINWU | L5WU | L4WU | L3WU | L2WU | L1WU | L0WU |
| | Wake Up Source Register | W | | | | | | | | |
| 0x15 | RSR | R | 0 | 0 | WDR | EXR | WUR | LVRX | LVR | POR |
| | Reset Status Register | W | | | | | | | | |
| 0x16 | MCR | R | 0 | 0 | 0 | 0 | 0 | 0 | MODE | |
| | Mode Control Register | W | | | | | | | | |
| 0x18 | LINR | R | LINOTIE | LINOTC | RX | TX | LVSD | LINEN | LINSR | |
| | LIN Register | W | | | | | | | | |
| 0x20 | PTBC1 | R | 0 | PUEB2 | PUEB1 | PUEB0 | 0 | DDRB2 | DDRB1 | DDRB0 |
| | Port B Configuration Register 1 | W | | | | | | | | |
| 0x21 | PTBC2 | R | 0 | 0 | 0 | 0 | PWMCS | PWMEN | SERMOD | |
| | Port B Config Register 2 | W | | | | | | | | |

**Table 73. Analog die Registers[64] - 0x0200–0x02FF D2D Blocking Access (D2DI) 2 of 3/ 0x0300–0x03FF D2D Non Blocking Access (D2DI) 3 of 3 (continued)**

| Offset | Name | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x22 | PTB | R | 0 | 0 | 0 | 0 | 0 | PTB2 | PTB1 | PTB0 |
| | Port B Data Register | W | | | | | | | | |
| 0x28 | HSCR | R | HSOTIE | HSHVSDE | PWMCS2 | PWMCS1 | PWMHS2 | PWMHS1 | HS2 | HS1 |
| | High-side Control Register | W | | | | | | | | |
| 0x29 | HSSR | R | HSOTC | 0 | 0 | 0 | HS2CL | HS1CL | HS2OL | HS1OL |
| | High-side Status Register | W | | | | | | | | |
| 0x30 | LSCR | R | LSOTIE | 0 | PWMCS2 | PWMCS1 | PWMLS2 | PWMLS1 | LS2 | LS1 |
| | Low-side Control Register | W | | | | | | | | |
| 0x31 | LSSR | R | LSOTC | 0 | 0 | 0 | LS2CL | LS1CL | LS2OL | LS1OL |
| | Low-side Status Register | W | | | | | | | | |
| 0x32 | LSCEN | R | 0 | 0 | 0 | 0 | LSCEN | | | |
| | Low-Side Control Enable Register | W | | | | | | | | |
| 0x38 | HSR | R | HOTIE | HOTC | 0 | 0 | 0 | 0 | 0 | HSUPON |
| | Hall Supply Register | W | | | | | | | | |
| 0x3C | CSR | R | CSE | 0 | 0 | 0 | CCD | CSGS | | |
| | Current Sense Register | W | | | | | | | | |
| 0x40 | SCIBD (hi) | R | LBKDIE | RXEDGIE | 0 | SBR12 | SBR11 | SBR10 | SBR9 | SBR8 |
| | SCI Baud Rate Register | W | | | | | | | | |
| 0x41 | SCIBD (lo) | R | SBR7 | SBR6 | SBR5 | SBR4 | SBR3 | SBR2 | SBR1 | SBR0 |
| | SCI Baud Rate Register | W | | | | | | | | |
| 0x42 | SCIC1 | R | LOOPS | 0 | RSRC | M | 0 | ILT | PE | PT |
| | SCI Control Register 1 | W | | | | | | | | |
| 0x43 | SCIC2 | R | TIE | TCIE | RIE | ILIE | TE | RE | RWU | SBK |
| | SCI Control Register 2 | W | | | | | | | | |
| 0x44 | SCIS1 | R | TDRE | TC | RDRF | IDLE | OR | NF | FE | PF |
| | SCI Status Register 1 | W | | | | | | | | |
| 0x45 | SCIS2 | R | LBKDIF | RXEDGIF | 0 | RXINV | RWUID | BRK13 | LBKDE | RAF |
| | SCI Status Register 2 | W | | | | | | | | |
| 0x46 | SCIC3 | R | R8 | T8 | TXDIR | TXINV | ORIE | NEIE | FEIE | PEIE |
| | SCI Control Register 3 | W | | | | | | | | |
| 0x47 | SCID | R | R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 |
| | SCI Data Register | W | T7 | T6 | T5 | T4 | T3 | T2 | T1 | T0 |
| 0x60 | PWMCTL | R | CAE1 | CAE0 | PCLK1 | PCLK0 | PPOL1 | PPOL0 | PWME1 | PWME0 |
| | PWM Control Register | W | | | | | | | | |

**Table 73. Analog die Registers[64] - 0x0200–0x02FF D2D Blocking Access (D2DI) 2 of 3/ 0x0300–0x03FF D2D Non Blocking Access (D2DI) 3 of 3 (continued)**

| Offset | Name | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x61 | PWMPRCLK | R | 0 | PCKB2 | PCKB1 | PCKB0 | 0 | PCKA2 | PCKA1 | PCKA0 |
| | PWM Presc. Clk Select Reg | W | | | | | | | | |
| 0x62 | PWMSCLA | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | PWM Scale A Register | W | | | | | | | | |
| 0x63 | PWMSCLB | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | PWM Scale B Register | W | | | | | | | | |
| 0x64 | PWMCNT0 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | PWM Ch Counter Reg 0 | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x65 | PWMCNT1 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | PWM Ch Counter Reg 1 | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x66 | PWMPER0 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | PWM Ch Period Register 0 | W | | | | | | | | |
| 0x67 | PWMPER1 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | PWM Ch Period Register 1 | W | | | | | | | | |
| 0x68 | PWMDTY0 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | PWM Ch Duty Register 0 | W | | | | | | | | |
| 0x69 | PWMDTY1 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | PWM Ch Duty Register 1 | W | | | | | | | | |
| 0x80 | ACR | R | SCIE | CCE | OCE | ADCRST | 0 | PS2 | PS1 | PS0 |
| | ADC Config Register | W | | | | | | | | |
| 0x81 | ASR | R | SCF | 2p5CLF | 0 | 0 | CCNT3 | CCNT2 | CCNT1 | CCNT0 |
| | ADC Status Register | W | | | | | | | | |
| 0x82 | ACCR (hi) | R | CH15 | CH14 | 0 | CH12 | CH11 | CH10 | CH9 | CH8 |
| | ADC Conversion Ctrl Reg | W | | | | | | | | |
| 0x83 | ACCR (lo) | R | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | CH0 |
| | ADC Conversion Ctrl Reg | W | | | | | | | | |
| 0x84 | ACCSR (hi) | R | CC15 | CC14 | 0 | CC12 | CC11 | CC10 | CC9 | CC8 |
| | ADC Conv Complete Reg | W | | | | | | | | |
| 0x85 | ACCSR (lo) | R | CC7 | CC6 | CC5 | CC4 | CC3 | CC2 | CC1 | CC0 |
| | ADC Conv Complete Reg | W | | | | | | | | |
| 0x86 | ADR0 (hi) | R | adr0 9 | adr0 8 | adr0 7 | adr0 6 | adr0 5 | adr0 4 | adr0 3 | adr0 2 |
| | ADC Data Result Register 0 | W | | | | | | | | |
| 0x87 | ADR0 (lo) | R | adr0 1 | adr0 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Register 0 | W | | | | | | | | |

**Table 73. Analog die Registers[64] - 0x0200–0x02FF D2D Blocking Access (D2DI) 2 of 3/ 0x0300–0x03FF D2D Non Blocking Access (D2DI) 3 of 3 (continued)**

| Offset | Name | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x88 | ADR1 (hi) | R | adr1 9 | adr1 8 | adr1 7 | adr1 6 | adr1 5 | adr1 4 | adr1 3 | adr1 2 |
| | ADC Data Result Register 1 | W | | | | | | | | |
| 0x89 | ADR1 (lo) | R | adr1 1 | adr1 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Register 1 | W | | | | | | | | |
| 0x8A | ADR2 (hi) | R | adr2 9 | adr2 8 | adr2 7 | adr2 6 | adr2 5 | adr2 4 | adr2 3 | adr2 2 |
| | ADC Data Result Register 2 | W | | | | | | | | |
| 0x8B | ADR2 (lo) | R | adr2 1 | adr2 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Register 2 | W | | | | | | | | |
| 0x8C | ADR3 (hi) | R | adr3 9 | adr3 8 | adr3 7 | adr3 6 | adr3 5 | adr3 4 | adr3 3 | adr3 2 |
| | ADC Data Result Register 3 | W | | | | | | | | |
| 0x8D | ADR3 (lo) | R | adr3 1 | adr3 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Register 3 | W | | | | | | | | |
| 0x8E | ADR4 (hi) | R | adr4 9 | adr4 8 | adr4 7 | adr4 6 | adr4 5 | adr4 4 | adr4 3 | adr4 2 |
| | ADC Data Result Register 4 | W | | | | | | | | |
| 0x8F | ADR4 (lo) | R | adr4 1 | adr4 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Register 4 | W | | | | | | | | |
| 0x90 | ADR5 (hi) | R | adr5 9 | adr5 8 | adr5 7 | adr5 6 | adr5 5 | adr5 4 | adr5 3 | adr5 2 |
| | ADC Data Result Register 5 | W | | | | | | | | |
| 0x91 | ADR5 (lo) | R | adr5 1 | adr5 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Register 5 | W | | | | | | | | |
| 0x92 | ADR6 (hi) | R | adr6 9 | adr6 8 | adr6 7 | adr6 6 | adr6 5 | adr6 4 | adr6 3 | adr6 2 |
| | ADC Data Result Register 6 | W | | | | | | | | |
| 0x93 | ADR6 (lo) | R | adr6 1 | adr6 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Register 6 | W | | | | | | | | |
| 0x94 | ADR7 (hi) | R | adr7 9 | adr7 8 | adr7 7 | adr7 6 | adr7 5 | adr7 4 | adr7 3 | adr7 2 |
| | ADC Data Result Register 7 | W | | | | | | | | |
| 0x95 | ADR7 (lo) | R | adr7 1 | adr7 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Register 7 | W | | | | | | | | |
| 0x96 | ADR8 (hi) | R | adr8 9 | adr8 8 | adr8 7 | adr8 6 | adr8 5 | adr8 4 | adr8 3 | adr8 2 |
| | ADC Data Result Register 8 | W | | | | | | | | |
| 0x97 | ADR8 (lo) | R | adr8 1 | adr8 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Register 8 | W | | | | | | | | |
| 0x98 | ADR9 (hi) | R | adr9 9 | adr9 8 | adr9 7 | adr9 6 | adr9 5 | adr9 4 | adr9 3 | adr9 2 |
| | ADC Data Result Register 9 | W | | | | | | | | |

**Table 73. Analog die Registers[64] - 0x0200–0x02FF D2D Blocking Access (D2DI) 2 of 3/ 0x0300–0x03FF D2D Non Blocking Access (D2DI) 3 of 3 (continued)**

| Offset | Name | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x99 | ADR9 (lo) | R | adr9 1 | adr9 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Register 9 | W | | | | | | | | |
| 0x9A | ADR10 (hi) | R | adr10 9 | adr10 8 | adr10 7 | adr10 6 | adr10 5 | adr10 4 | adr10 3 | adr10 2 |
| | ADC Data Result Reg 10 | W | | | | | | | | |
| 0x9B | ADR10 (lo) | R | adr10 1 | adr10 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Reg 10 | W | | | | | | | | |
| 0x9C | ADR11 (hi) | R | adr11 9 | adr11 8 | adr11 7 | adr11 6 | adr11 5 | adr11 4 | adr11 3 | adr11 2 |
| | ADC Data Result Reg 11 | W | | | | | | | | |
| 0x9D | ADR11 (lo) | R | adr11 1 | adr11 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Reg 11 | W | | | | | | | | |
| 0x9E | ADR12 (hi) | R | adr12 9 | adr12 8 | adr12 7 | adr12 6 | adr12 5 | adr12 4 | adr12 3 | adr12 2 |
| | ADC Data Result Reg 12 | W | | | | | | | | |
| 0x9F | ADR12 (lo) | R | adr12 1 | adr12 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Reg 12 | W | | | | | | | | |
| 0xA2 | ADR14 (hi) | R | adr14 9 | adr14 8 | adr14 7 | adr14 6 | adr14 5 | adr14 4 | adr14 3 | adr14 2 |
| | ADC Data Result Reg 14 | W | | | | | | | | |
| 0xA3 | ADR14 (lo) | R | adr14 1 | adr14 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Reg 14 | W | | | | | | | | |
| 0xA4 | ADR15 (hi) | R | adr15 9 | adr15 8 | adr15 7 | adr15 6 | adr15 5 | adr15 4 | adr15 3 | adr15 2 |
| | ADC Data Result Reg 15 | W | | | | | | | | |
| 0xA5 | ADR15 (lo) | R | adr15 1 | adr15 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Reg 15 | W | | | | | | | | |
| 0xC0 | TIOS | R | 0 | 0 | 0 | 0 | IOS3 | IOS2 | IOS1 | IOS0 |
| | TIM InCap/OutComp Select | W | | | | | | | | |
| 0xC1 | CFORC | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Timer Compare Force Reg | W | | | | | FOC3 | FOC2 | FOC1 | FOC0 |
| 0xC2 | OC3M | R | 0 | 0 | 0 | 0 | OC3M3 | OC3M2 | OC3M1 | OC3M0 |
| | Output Comp 3 Mask Reg | W | | | | | | | | |
| 0xC3 | OC3D | R | 0 | 0 | 0 | 0 | OC3D3 | OC3D2 | OC3D1 | OC3D0 |
| | Output Comp 3 Data Reg | W | | | | | | | | |
| 0xC4 | TCNT (hi) | R | tcnt 15 | tcnt 14 | tcnt 13 | tcnt 12 | tcnt 11 | tcnt 10 | tcnt 9 | tcnt 8 |
| | Timer Count Register | W | | | | | | | | |
| 0xC5 | TCNT (lo) | R | tcnt 7 | tcnt 6 | tcnt 5 | tcnt 4 | tcnt 3 | tcnt 2 | tcnt 1 | tcnt 0 |
| | Timer Count Register | W | | | | | | | | |

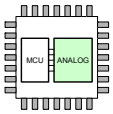
**Table 73. Analog die Registers[64] - 0x0200–0x02FF D2D Blocking Access (D2DI) 2 of 3/ 0x0300–0x03FF D2D Non Blocking Access (D2DI) 3 of 3 (continued)**

| Offset | Name | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0xC6 | TSCR1 | R | TEN | 0 | 0 | TFFCA | 0 | 0 | 0 | 0 |
| | Timer System Control Reg 1 | W | | | | | | | | |
| 0xC7 | TTOV | R | 0 | 0 | 0 | 0 | TOV3 | TOV2 | TOV1 | TOV0 |
| | Timer Toggle Overflow Reg | W | | | | | | | | |
| 0xC8 | TCTL1 | R | OM3 | OL3 | OM2 | OL2 | OM1 | OL1 | OM0 | OL0 |
| | Timer Control Register 1 | W | | | | | | | | |
| 0xC9 | TCTL2 | R | EDG3B | EDG3A | EDG2B | EDG2A | EDG1B | EDG1A | EDG0B | EDG0A |
| | Timer Control Register 2 | W | | | | | | | | |
| 0xCA | TIE | R | 0 | 0 | 0 | 0 | C3I | C2I | C1I | C0I |
| | Timer Interrupt Enable Reg | W | | | | | | | | |
| 0xCB | TSCR2 | R | TOI | 0 | 0 | 0 | TCRE | PR2 | PR1 | PR0 |
| | Timer System Control Reg 2 | W | | | | | | | | |
| 0xCC | TFLG1 | R | 0 | 0 | 0 | 0 | C3F | C2F | C1F | C0F |
| | Main Timer Interrupt Flag 1 | W | | | | | | | | |
| 0xCD | TFLG2 | R | TOF | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Main Timer Interrupt Flag 2 | W | | | | | | | | |
| 0xCE | TC0 (hi) | R | tc0 15 | tc0 14 | tc0 13 | tc0 12 | tc0 11 | tc0 10 | tc0 9 | tc0 8 |
| | TIM InCap/OutComp Reg 0 | W | | | | | | | | |
| 0xCF | TC0 (lo) | R | tc0 7 | tc0 6 | tc0 5 | tc0 4 | tc0 3 | tc0 2 | tc0 1 | tc0 0 |
| | TIM InCap/OutComp Reg 0 | W | | | | | | | | |
| 0xD0 | TC1 (hi) | R | tc1 15 | tc1 14 | tc1 13 | tc1 12 | tc1 11 | tc1 10 | tc1 9 | tc1 8 |
| | TIM InCap/OutComp Reg 1 | W | | | | | | | | |
| 0xD1 | TC1 (lo) | R | tc1 7 | tc1 6 | tc1 5 | tc1 4 | tc1 3 | tc1 2 | tc1 1 | tc1 0 |
| | TIM InCap/OutComp Reg 1 | W | | | | | | | | |
| 0xD2 | TC2 (hi) | R | tc2 15 | tc2 14 | tc2 13 | tc2 12 | tc2 11 | tc2 10 | tc2 9 | tc2 8 |
| | TIM InCap/OutComp Reg 2 | W | | | | | | | | |
| 0xD3 | TC2 (lo) | R | tc2 7 | tc2 6 | tc2 5 | tc2 4 | tc2 3 | tc2 2 | tc2 1 | tc2 0 |
| | TIM InCap/OutComp Reg 2 | W | | | | | | | | |
| 0xD4 | TC3 (hi) | R | tc3 15 | tc3 14 | tc3 13 | tc3 12 | tc3 11 | tc3 10 | tc3 9 | tc3 8 |
| | TIM InCap/OutComp Reg 3 | W | | | | | | | | |
| 0xD5 | TC3 (lo) | R | tc3 7 | tc3 6 | tc3 5 | tc3 4 | tc3 3 | tc3 2 | tc3 1 | tc3 0 |
| | TIM InCap/OutComp Reg 3 | W | | | | | | | | |
| 0xF0 | CTR0 | R | LINTRE | LINTR | WDCTRE | CTR0_4 | CTR0_3 | WDCTR2 | WDCTR1 | WDCTR0 |
| | Trimming Reg 0 | W | | | | | | | | |

**Table 73. Analog die Registers[64] - 0x0200–0x02FF D2D Blocking Access (D2DI) 2 of 3/ 0x0300–0x03FF D2D Non Blocking Access (D2DI) 3 of 3 (continued)**

| Offset | Name | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0xF1 | CTR1 Trimming Reg 1 | R<br>W | BGTRE | CTR1_6 | BGTRIMUP | BGTRIMDN | IREFTRE | IREFTR2 | IREFTR1 | IREFTR0 |
| 0xF2 | CTR2 Trimming Reg 2 | R<br>W | CTR2_E | CTR2_1 | CTR2_0 | SLPBGTRE | SLPBG_LOCK | SLPBGTR2 | SLPBGTR1 | SLPBGTR0 |
| 0xF3 | CTR3 Trimming Reg 3 | R<br>W | OFFCTRE | OFFCTR2 | OFFCTR1 | OFFCTR0 | CTR3_E | CTR3_2 | CTR3_1 | CTR3_0 |
| 0xF4 | SRR Silicon Revision Register | R | 0 | 0 | 0 | 0 | FMREV | | MMREV | |
| | | W | | | | | | | | |

Note:
64. Registers not shown are reserved and must not be accessed.




# 5.3MM912_634 - Analog Die Overview

## 5.3.1Introduction

The MM912_634 analog die implements all system base functionality to operate the integrated microcontroller, and delivers application specific actuator control as well as input capturing.

## 5.3.2 System Registers

### 5.3.2.1 Silicon Revision Register (SRR)

**Table 74. Silicon Revision Register (SRR)**

Offset[64] 0xF4 Access: User read

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

Note:
65. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 75. SRR - Register Field Descriptions**

| Field | Description |
|---|---|
| 3-2 FMREV | MM912_634 analog die Silicon Revision Register - These bits represent the revision of Silicon of the analog die. They are incremented for every full mask or metal mask issued of the device. One number is set for one revision of the silicon of the analog die. [66] |
| 1-0 MMREV | |

Note:
66. Refer to MM912_634ER - MM912_634, Silicon Analog Mask (M91W) / Digital Mask (N53A) Errata.

### 5.3.3 Analog Die Options

The following section describes the differences between analog die options 1 and 2.

**Table 76. Analog Die Options**

| Feature | Option 1 | Option 2 |
|---|---|---|
| Current Sense Module | YES | NO |
| Wake Up Inputs (Lx) | L0…L5 | L0.L3 |

**NOTE**

This document describes the features and functions of option 1 (all modules available and tested). Beyond this chapter, there is no additional note or differentiation between the different implementations.

#### 5.3.3.1 Current Sense Module

For device options with the current sense module not available, the following considerations are to be made.

##### 5.3.3.1.1 Pinout Considerations

**Table 77. ISENSE - Pin Considerations**

| Pin | PIN name for option 1 | New PIN name | Comment |
|---|---|---|---|
| 40 | ISENSEL | NC | ISENSE feature not bonded and/or not tested. Connect PINs 40 and 41 (NC) to GND. |
| 41 | ISENSEH | NC | |

##### 5.3.3.1.2 Register Considerations

The Current Sense Register must remain in default (0x00) state.

| Offset | Name | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x3C | CSR | R | CSE | 0 | 0 | 0 | CCD | CSGS | | |
| | Current Sense Register | W | | | | | | | | |

The Conversion Control Register - Bit 9 must always be written 0.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x82 | ACCR (hi) | R | CH15 | CH14 | 0 | CH12 | CH11 | CH10 | CH9 | CH8 |
| | ADC Conversion Ctrl Reg | W | | | | | | | | |

The Conversion Complete Register - Bit 9 must be ignored.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x84 | ACCSR (hi) | R | CC15 | CC14 | 0 | CC12 | CC11 | CC10 | CC9 | CC8 |
| | ADC Conv Complete Reg | W | | | | | | | | |

The ADC Data Result Reg 9 must be ignored.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x98 | ADR9 (hi) | R | adr9 9 | adr9 8 | adr9 7 | adr9 6 | adr9 5 | adr9 4 | adr9 3 | adr9 2 |
| | ADC Data Result Register 9 | W | | | | | | | | |
| 0x99 | ADR9 (lo) | R | adr9 1 | adr9 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Register 9 | W | | | | | | | | |

#### 5.3.3.1.3 Functional Considerations

- The complete Current Sense Module is not available.
- The ADC Channel 9 is not available.

### 5.3.3.2 Wake-up Inputs (Lx)

For device options with reduced number of wake-up inputs (Lx), the following considerations are to be made.

#### 5.3.3.2.1 Pinout Considerations

**Table 78. Lx - Pin Considerations**

| Pin | PIN Name for Option 1 | New PIN name | Comment |
|---|---|---|---|
| 31…36 | Lx | NC | One or more Lx wake-up inputs are not available based on the analog die option. Not available Lx inputs are not bonded and/or not tested. Connect not available Lx pins (NC) to GND. RLx is not required on those pins. |

#### 5.3.3.2.2 Register Considerations

The Lx - Bit for the not available Lx input in the Lx Status Register must be ignored.

| Offset | Name | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x08 | LXR | R | 0 | 0 | L5 | L4 | L3 | L2 | L1 | L0 |
| | Lx Status Register | W | | | | | | | | |

The Lx Control register for the not available Lx input must be written 0.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x09 | LXCR | R | 0 | 0 | L5DS | L4DS | L3DS | L2DS | L1DS | L0DS |
| | Lx Control Register | W | | | | | | | | |

A not available Lx input can not be selected as Wake-up Source and must have its LxWE bit set to 0.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x12 | WCR | R | CSSEL | | L5WE | L4WE | L3WE | L2WE | L1WE | L0WE |
| | Wake Up Control Register | W | | | | | | | | |

The Wake-up Source Register for not available Lx inputs must be ignored.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x14 | WSR | R | FWU | LINWU | L5WU | L4WU | L3WU | L2WU | L1WU | L0WU |
| | Wake Up Source Register | W | | | | | | | | |

The Conversion Control Register for the not available Lx analog input (3…8) must always be written 0.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x82 | ACCR (hi) | R | CH15 | CH14 | 0 | CH12 | CH11 | CH10 | CH9 | CH8 |
| | ADC Conversion Ctrl Reg | W | | | | | | | | |
| 0x83 | ACCR (lo) | R | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | CH0 |
| | ADC Conversion Ctrl Reg | W | | | | | | | | |

The Conversion Complete Register for the not available Lx analog input (3.8) must be ignored.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x84 | ACCSR (hi) | R | CC15 | CC14 | 0 | CC12 | CC11 | CC10 | CC9 | CC8 |
| | ADC Conv Complete Reg | W | | | | | | | | |
| 0x85 | ACCSR (lo) | R | CC7 | CC6 | CC5 | CC4 | CC3 | CC2 | CC1 | CC0 |
| | ADC Conv Complete Reg | W | | | | | | | | |

The ADC Data Result Register for the not available Lx analog input (3.8) must be ignored.

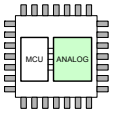
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0x8C-0x97 | ADRx (hi) | R | adrx 9 | adrx 8 | adrx 7 | adrx 6 | adrx 5 | adrx 4 | adrx 3 | adrx 2 |
| | ADC Data Result Register x | W | | | | | | | | |
| | ADRx (lo) | R | adrx 1 | adrx 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | ADC Data Result Register x | W | | | | | | | | |

#### 5.3.3.2.3 Functional Considerations

For the not available Lx inputs, the following functions are limited:

- No Wake-up feature / Cyclic Sense
- No Digital Input
- No Analog Input and conversion via ADC

## 5.4 Modes of Operation

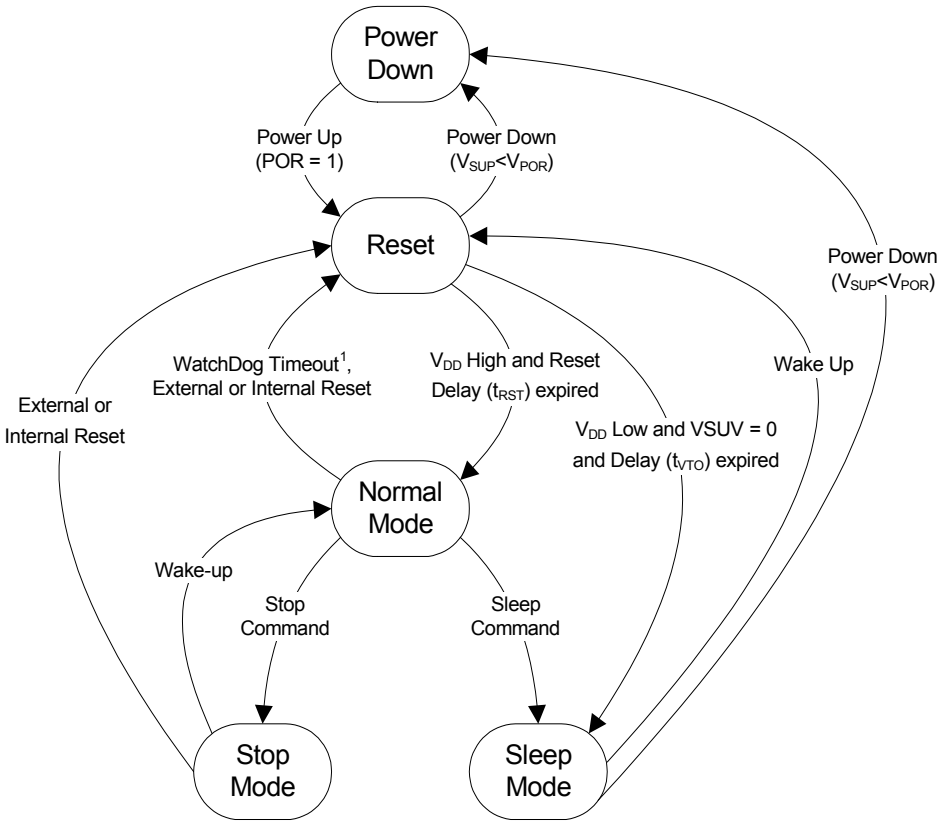



The MM912_634 analog die offers three main operating modes: Normal (Run), Stop, and Sleep. In Normal mode, the device is active and is operating under normal application conditions. In Stop mode, the voltage regulator operates with limited current capability, the external load is expected to be reduced while in Stop mode. In Sleep mode both voltage regulators are turned off ($V_{DD}$ = $V_{DDX}$ = 0 V).

Wake-up from Stop mode is indicated by an interrupt signal. Wake-up from Sleep mode changes the MM912_634 analog die into reset mode while the voltage regulator is turned back on.

The selection of the different modes is controlled by the Mode Control Register (MCR).

Figure 16 describes how transitions are done between the different operating modes.




[1] Initial WD to be served within $t_{WDTO}$ to enable Window WD

**Figure 16. Modes of Operation and Transitions**

### 5.4.1 Power Down Mode

For the device power (VS1) below $V_{POR}$, the MM912_634 analog die is virtually in Power Down mode. Once VS1>VPOR, the MM912_634 analog die enters Reset mode with the condition “Power On Reset - POR”.

### 5.4.2 Reset Mode

The MM912_634 analog die enters Reset mode if a reset condition occurs (POR - Power On Reset, LVR- Low Voltage Reset, Low Voltage VDDX Reset - LVRX, WDR - Watchdog Reset, EXR - External Reset, and WUR - Wake-up Sleep Reset).

For internal reset sources, the $\overline{\text{RESET_A}}$ pin is driven low for $t_{RST}$ after the reset condition is gone. After this delay, the $\overline{\text{RESET_A}}$ pin is released. With a high detected on the $\overline{\text{RESET_A}}$ pin, VDD>VLVR and VDDX>VLVRX the MM912_634 analog die enters in Normal mode.

To avoid short-circuit conditions being present for a long time, a $t_{VTO}$ timeout is implemented. Once VDD < VLVR or VDDX < VLVRX with VS1 > (VLVI + VLVI_H) for more than $t_{VTO}$, the MM912_634 analog die transits directly to Sleep mode.

The Reset Status Register (RSR) indicates the source of the reset by individual flags.

- POR - Power On Reset
- LVR - Low Voltage Reset VDD
- LVRX - Low Voltage Reset VDDX
- WDR - Watchdog Reset
- EXR - External Reset
- WUR - Wake-up Sleep Reset

See also Section 5.8, “Resets".

### 5.4.3 Normal Mode

In Normal mode, all MM912_634 analog die user functions are active and can be controlled by the D2D Interface. Both regulators (VDD and VDDX) are active and operate with full current capability.

Once entered in Normal mode, the Watchdog operates as a simple non-window watchdog with an initial timeout ($tI_{WDTO}$) to be reset via the D2D Interface. After the initial reset, the watchdog operates in standard window mode. See Window Watchdog for details.

### 5.4.4 Stop Mode

The Stop mode allows reduced current consumption with fast startup time. In this mode, both voltage regulators (VDD and VDDX) are active, with limited current drive capability. In this condition, the MCU is supposed to operate in Low Power mode (STOP).

**NOTE**

To avoid any pending analog die interrupts prevent the MCU from entering MCU stop resulting in unexpected system behavior, the analog die IRQ sources should be disabled and the corresponding flags be cleared before entering stop.

The device can enter in Stop mode by configuring the Mode Control Register (MCR) via the D2D Interface. The MCU has to enter a Low Power mode immediately afterwards executing the STOP instruction. The Wake-up Source Register (WSR) has to be read after a wake-up condition in order to execute a new STOP mode command. Two base clock cycles ($f_{BASE}$) delay are required between WSR read and MCR write.

While in Stop mode, the MM912_634 analog die wakes up on the following sources:

- Lx - Wake-up (maskable with selectable cyclic sense)
- Forced Wake-up (configurable timeout)
- LIN Wake-up
- D2D Wake-up (special command)

After Wake-up from the sources listed above, the device transits to Normal mode.

Reset wakes up the device directly to Reset mode.

See Section 5.9, “Wake-up / Cyclic Sense" for details.

### 5.4.5 Sleep Mode

The Sleep mode allows very low current consumption. In this mode, both voltage regulators (VDD and VDDX) are inactive.

The device can enter into Sleep mode by configuring the Mode Control Register (MCR) via the D2D- Interface. During Sleep mode, all unused internal blocks are deactivated to allow the lowest possible consumption. Power consumption decreases further if the Cyclic Sense or Forced Wake-up feature are disabled. While in Sleep mode, the MM912_634 analog die wakes up on the following sources:

- Lx - Wake-up (maskable with selectable cyclic sense)
- Forced Wake-up (configurable timeout)
- LIN Wake-up

After Wake-up from the sources listed above or a reset condition, the device transits to Reset mode.

See Section 5.9, “Wake-up / Cyclic Sense" for details.

### 5.4.6 Analog Die Functionality by Operation Mode

**Table 79. Operation Mode Overview**

| Function | Reset | Normal | Stop | Sleep |
|---|---|---|---|---|
| VDD/VDDX | full | full | stop | OFF |
| HSUP | OFF | full | OFF | OFF |
| LSx | | full | OFF | OFF |
| HSx | | full | Cyclic Sense[64] | Cyclic Sense[64] |
| ADC | | full | OFF | OFF |
| D2D | | full | functional | OFF |
| Lx | | full | Wake-up[64] | Wake-up[64] |
| PTBx | | full | OFF | OFF |
| LIN | | full | Wake-up[64] | Wake-up[64] |
| Watchdog | | full[68] | OFF | OFF |
| VSENSE | | full | OFF | OFF |
| CSENSE | | full | OFF | OFF |
| Cyclic Sense | | not active | Cyclic Sense[64] | Cyclic Sense[64] |

Note:
67. If configured.
68. Special init through non window watchdog.

### 5.4.7 Register Definition

#### 5.4.7.1 Mode Control Register (MCR)

**Table 80. Mode Control Register (MCR)**

Offset[69] 0x16 Access: User read/write

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

Note:
69. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 81. MCR - Register Field Descriptions**

| Field | Description |
|---|---|
| 1-0 MODE | Mode Select - These bits issue a transition from to the selected Operating mode.<br>00 - Normal mode. Only with effect in Stop mode. Issues Wake-up and transitions to Normal mode.<br>01 - Stop mode. Initiates transition to Stop mode.[70]<br>10 - Sleep mode. Initiates transition to Sleep mode.<br>11 - Normal mode. |

Note:
70. The Wake-up Source Register (WSR) has to be read after a wake-up condition in order to execute a new STOP mode command. Two base clock cycles ($f_{BASE}$) delay are required between WSR read and MCR write.

## 5.5 Power Supply

The MM912_634 analog die supplies VDD (2.5 V), VDDX (5.0 V), and HSUP, based on the supply voltage applied to the VS1 pin. VDD is cascaded of the VDDX regulator. To separate the high-side outputs from the main power supply, the VS2 pin does only power the high-side drivers. Both supply pins have to be externally protected against reverse battery conditions. To supply external Hall Effect Sensors, the HSUP pin supplies a switchable regulated supply. See Section 5.11, “Hall Sensor Supply Output - HSUP".

A reverse battery protected input (VSENSE) is implemented to measure the Battery Voltage directly. A serial resistor (RVSENSE) is required on this pin. See Section 5.23, “Supply Voltage Sense - VSENSE". In addition, the VS1 supply can be routed to the ADC (VS1SENSE) to measure the VS1 pin voltage directly. See Section 5.24, “Internal Supply Voltage Sense - VS1SENSE".

To have an independent ADC verification, the internal sleep mode bandgap voltage can be routed to the ADC (BANDGAP). As this node is independent from the ADC reference, any out of range result would indicate malfunctioning ADC or Bandgap reference. See Section 5.25, “Internal Bandgap Reference Voltage Sense - BANDGAP".

To stabilize the internal ADC reference voltage for higher precision measurements, the current limited ADC2p5 pin needs to be connected to an external filter capacitor (CADC2p5). It is not recommended to connect additional loads to this pin. See Section 5.20, “Analog Digital Converter - ADC".

The following safety features are implemented:

- LBI - Low Battery Interrupt, internally measured at VSENSE
- LVI - Low Voltage Interrupt, internally measured at VS1
- HVI - High Voltage Interrupt, internally measured at VS2
- VROVI - Voltage Regulator Overvoltage Interrupt internally measured at VDD and VDDX
- LVR - Low Voltage Reset, internally measured at VDD
- LVRX - Low Voltage Reset, internally measured at VDDX
- HTI - High Temperature Interrupt measured between the VDD and VDDX regulators
- Overtemperature Shutdown measured between the VDD and VDDX regulators

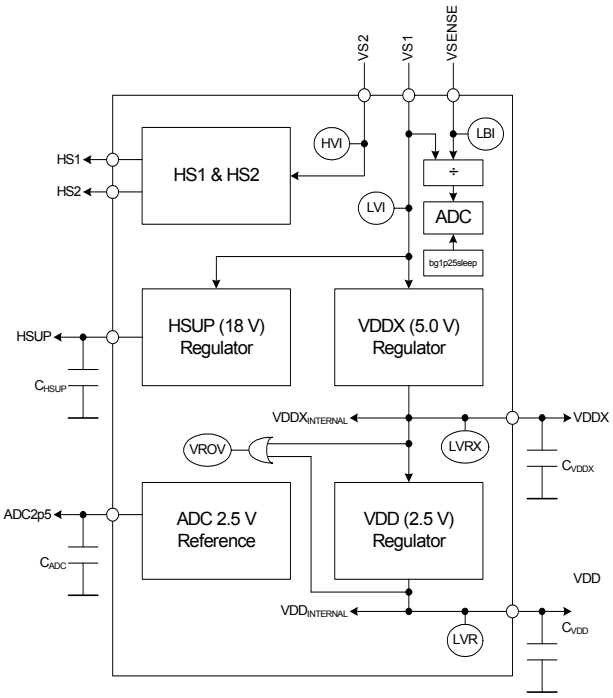



**Figure 17. MM912_634 Power Supply**

### 5.5.1 Voltage Regulators VDD (2.5 V) & VDDX (5.0 V)

To supply the MCU die and minor additional loads two cascaded voltage regulators have been implemented, VDDX (5.0 V) and VDD (2.5 V). External capacitors (CVDD) and (CVDDX) are required for proper regulation.

### 5.5.2 Power Up Behavior / Power Down Behavior - I64

To guarantee safe power up and down behavior, special dependencies are implemented to prevent unwanted MCU execution. Figure 18 shows a standard power up and power down sequence.

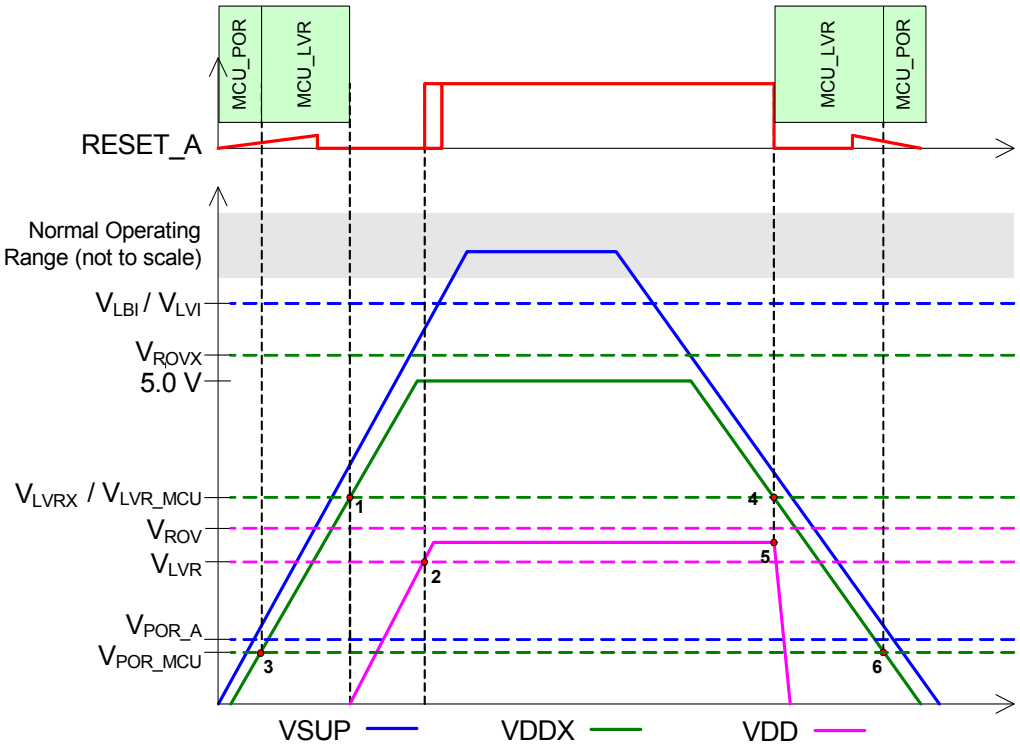



**Figure 18. Power Up / Down Sequence**

To avoid any abnormal device behavior, it is essential to have the MCU Power on Reset (POR) block complete its start-up sequence before the analog die reset signal ($\overline{\text{RESET_A}}$) is asserted. Since the $\overline{\text{RESET_A}}$ circuitry is supplied by VDDX, the voltage on the 2.5 V supply ($V_{DD}$) needs to remain below the POR threshold whenever $V_{DDX}$ is too low to guarantee $\overline{\text{RESET_A}}$ can be properly asserted (3;6). This is achieved with the following implementation.

Power Up:

- The $V_{DD}$ regulator is enabled after $V_{DDX}$ reaches the $V_{LVRX}$ threshold (1).
- Once $V_{DD}$ reaches $V_{LVR}$, the $\overline{\text{RESET_A}}$ is released (2).
- The MCU is also protected by the MCU_LVR.

Power Down:

- Once $V_{DDX}$ has reached the $V_{LVRX}$ threshold (4), the $V_{DD}$ regulator is disabled and the regulator output is actively pulled down to discharge any $V_{DD}$ capacitance (5). $\overline{\text{RESET_A}}$ is activated as well.
- The active discharge guarantees $V_{DD}$ to be below POR level before $V_{DDX}$ discharges below critical level for the reset circuity.

**NOTE**

The behavior explained previously is essential for the 9S12I64 MCU die used, as this MCU does have an internal regulator stage, but the LVR function only active in normal mode 9S12I64.

The shutdown behavior should be considered when sizing the external capacitors $C_{VDD}$ and $C_{VDDX}$ for extended low voltage operation.

### 5.5.3 Register Definition

#### 5.5.3.1 Voltage Control Register (VCR)

**Table 82. Voltage Control Register (VCR)**

Offset[69] 0x04 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | VROVIE | HTIE | HVIE | LVIE | LBIE |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

71. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 83. VCR - Register Field Descriptions**

| Field | Description |
|---|---|
| 4<br>VROVIE | Voltage Regulator Overvoltage Interrupt Enable — Enables the interrupt for the Regulator Overvoltage Condition.<br>0 - Voltage Regulator Overvoltage Interrupt is disabled<br>1 - Voltage Regulator Overvoltage Interrupt is enabled |
| 3<br>HTIE | High Temperature Interrupt Enable — Enables the interrupt for the Voltage Regulator (VDD/VDDX) Temperature Warning.<br>0 - High Temperature Interrupt is disabled<br>1 - High Temperature Interrupt is enabled |
| 2<br>HVIE | High Voltage Interrupt Enable — Enables the interrupt for the VS2 - High Voltage Warning.<br>0 - High Voltage Interrupt is disabled<br>1 - High Voltage Interrupt is enabled |
| 1<br>LVIE | Low Voltage Interrupt Enable — Enables the interrupt for the VS1 - Low Voltage Warning.<br>0 - Low Voltage Interrupt is disabled<br>1 - Low Voltage Interrupt is enabled |
| 0<br>LBIE | Low Battery Interrupt Enable — Enables the interrupt for the VSENSE - Low Battery Voltage Warning.<br>0 - Low Battery Interrupt is disabled<br>1 - Low Battery Interrupt is enabled |

#### 5.5.3.2 Voltage Status Register (VSR)

**Table 84. Voltage Status Register (VSR)**

Offset[72] 0x05 Access: User read

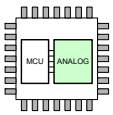
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | VROVC | HTC | HVC | LVC | LBC |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

72. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 85. VSR - Register Field Descriptions**

| Field | Description |
|---|---|
| 4<br>VROVC | Voltage Regulator Overvoltage Condition - This status bit indicates an overvoltage warning is present for at least one of the main voltage regulators (VDD or VDDX). Reading the register clears the VROVI flag if present. See Section 5.7, “Interrupts" for details. Note: This feature requires the trimming of Section 5.26.1.2.3, “Trimming Register 2 (CTR2)" to be done to be effective. Untrimmed devices may issue the VROVC condition including the LS turn off at normal operation.<br>0 - No Voltage Regulator Overvoltage Condition present.<br>1 - Voltage Regulator Overvoltage Condition present. |
| 3<br>HTC | High Temperature Condition - This status bit indicates a high temperature warning is present for the Voltage regulators (VDD/VDDX). Reading the register clears the HTI flag if present. See Section 5.7, “Interrupts" for details.<br>0 - No High Temperature Condition present.<br>1 - High Temperature Condition present. |
| 2<br>HVC | High Voltage Condition - This status bit indicates a high voltage warning for VS2 is present. Reading the register clears the HVI flag if present. See Section 5.7, “Interrupts" for details.<br>0 - No High Voltage Condition present.<br>1 - High Voltage Condition present. |
| 1<br>LVC | Low Voltage Condition - This status bit indicates a low voltage warning for VS1 is present. Reading the register clears the LVI flag if present. See Section 5.7, “Interrupts" for details.<br>0 - No Low Voltage Condition present.<br>1 - Low Voltage Condition present. |
| 0<br>LBC | Low Battery Condition - This status bit indicates a low voltage warning for VSENSE is present. Reading the register clears the LBI flag if present. See Section 5.7, “Interrupts" for details.<br>0 - No Low Battery Condition present.<br>1 - Low Battery Condition present. |




## 5.6Die to Die Interface - Target

The D2D Interface is the bus interface to the Microcontroller. Access to the MM912_634 analog die is controlled by the D2D Interface module. This section describes the functionality of the die-to-die target block (D2D).

### 5.6.1 Overview

The D2D is the target for a data transfer from the target to the initiator (MCU). The initiator provides a set of configuration registers and two memory mapped 256 Byte address windows. When writing to a window, a transaction is initiated sending a write command, followed by an 8-bit address, and the data byte or word is received from the initiator. When reading from a window, a transaction is received with the read command, followed by an 8-bit address. The target then responds with the data. The basic idea is that a peripheral located on the MM912_634 analog die, can be addressed like an on-chip peripheral.

Features:

- software transparent register access to peripherals on the MM912_634 analog die
- 256 Byte address window
- supports blocking read or write, as well as non-blocking write transactions
- 4-bit physical bus width
- automatic synchronization of the target when initiator starts driving the interface clock
- generates transaction and error status as well as EOT acknowledge
- providing single interrupt interface to D2D Initiator

### 5.6.2 Low Power Mode Operation

The D2D module is disabled in SLEEP mode. In Stop mode, the D2DINT signal is used to wake-up a powered down MCU. As the MCU could wake-up without the MM912_634 analog die, a special command is recognized as a wake-up event during Stop mode. See .

#### 5.6.2.1 Normal Mode / Stop Mode

While in Normal or Stop mode, D2DCLK acts as input only with pull present. D2D[3:0] operates as an input/output with pull-down always present. D2DINT acts as output only.

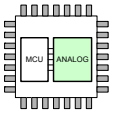
**NOTE**

The maximum allowed clock speed of the interface is limited to $f_{D2D}$.

#### 5.6.2.2 Sleep Mode

While in Sleep mode, all Interface data pins are pulled down to DGND to reduce power consumption.

## 5.7 Interrupts




Interrupts are used to signal a microcontroller which a peripheral needs to be serviced. While in Stop mode, the interrupt signal is used to signal Wake-up events. The interrupts are signaled by an active high level of the D2DINT pin, which remains high until the interrupt is acknowledged via the D2D-Interface. Interrupts are only asserted while in Normal mode.

### 5.7.1 Interrupt Source Identification

Once an Interrupt is signalized, there are two options to identify the corresponding source(s).

#### 5.7.1.1 Interrupt Source Mirror

All Interrupt sources in MM912_634 analog die are mirrored to a special Interrupt Source Register (ISR). This register is read only and indicates all currently pending Interrupts. Reading this register does not acknowledge any interrupt. An additional D2D access is necessary to serve the specific module.

**NOTE**

The VSI - Voltage Status Interrupt combines the five status flags for the Low Battery Interrupt, Low Voltage Interrupt, High Voltage Interrupt, Voltage Regulator Overvoltage Interrupt, and the Voltage Regulator High Temperature Interrupt. The specific source can be identified by reading the Voltage Status Register - VSR.

##### 5.7.1.1.1 Interrupt Source Register (ISR)

**Table 86. Interrupt Source Register (ISR)**

Offset[(69)] 0x00 (0x00 and 0x01 for 8Bit access) Access: User read

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | HOT | LSOT | HSOT | LINOT | SCI | RX | TX | ERR | TOV | CH3 | CH2 | CH1 | CH0 | VSI |
| W | | | | | | | | | | | | | | | | |

Note:
73. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 87. ISR - Register Field Descriptions**

| Field | Description |
|---|---|
| 0 - VSI | VSI - Voltage Status Interrupt combining the following sources:<br>• Low Battery Interrupt<br>• Low Voltage Interrupt<br>• High Voltage Interrupt<br>• Voltage Regulator Overvoltage Interrupt<br>• Voltage Regulator High Temperature Interrupt |
| 1 - CH0 | CH0 - TIM Channel 0 Interrupt |

**Table 87. ISR - Register Field Descriptions (continued)**

| Field | Description |
|---|---|
| 2 - CH1 | CH1 - TIM Channel 1 Interrupt |
| 3 - CH2 | CH2 - TIM Channel 2 Interrupt |
| 4 - CH3 | CH3 - TIM Channel 3 Interrupt |
| 5 - TOV | TOV - Timer Overflow Interrupt |
| 6 - ERR | ERR - SCI Error Interrupt |
| 7 - TX | TX - SCI Transmit Interrupt |
| 8 - RX | RX - SCI Receive Interrupt |
| 9 - SCI | SCI - ADC Sequence Complete Interrupt |
| 10 - LINOT | LINOT - LIN Driver Overtemperature Interrupt |
| 11 - HSOT | HSOT - High-side Overtemperature Interrupt |
| 12 - LSOT | LSOT - Low-side Overtemperature Interrupt |
| 13 - HOT | HOT - HSUP Overtemperature Interrupt |

### 5.7.1.2 Interrupt Vector Emulation by Priority

To allow a vector based interrupt handling by the MCU, the number of the highest prioritized interrupt pending is returned in the Interrupt Vector Register. To allow an offset based vector table, the result is pre-shifted (multiple of 2). Reading this register does not acknowledge an interrupt. An additional D2D access is necessary to serve the specific module.

#### 5.7.1.2.1 Interrupt Vector Register (IVR)

**Table 88. Interrupt Vector Register (IVR)**

Offset[74] 0x02 Access: User read

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

Note:
74. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 89. IVR - Register Field Descriptions**

| Field | Description |
|---|---|
| 5:0 IRQ | Represents the highest prioritized interrupt pending. See Table 90. If no interrupt is pending, the result is 0. |

The following table is listing all MM912_634 analog die interrupt sources with the corresponding priority.

**Table 90. Interrupt Source Priority**

| Interrupt Source | IRQ | Priority |
|---|---|---|
| no interrupt pending or wake-up from Stop mode | 0x00 | 1 (highest) |
| LVI - Low Voltage Interrupt | 0x02 | 2 |
| HTI - Voltage Regulator High Temperature Interrupt | 0x04 | 3 |
| LBI - Low Battery Interrupt | 0x06 | 4 |
| CH0 - TIM Channel 0 Interrupt | 0x08 | 5 |

**Table 90. Interrupt Source Priority (continued)**

| Interrupt Source | IRQ | Priority |
|---|---|---|
| CH1 - TIM Channel 1 Interrupt | 0x0A | 6 |
| CH2 - TIM Channel 2 Interrupt | 0x0C | 7 |
| CH3 - TIM Channel 3 Interrupt | 0x0E | 8 |
| TOV - Timer Overflow Interrupt | 0x10 | 9 |
| ERR - SCI Error Interrupt | 0x12 | 10 |
| TX - SCI Transmit Interrupt | 0x14 | 11 |
| RX - SCI Receive Interrupt | 0x16 | 12 |
| SCI - ADC Sequence Complete Interrupt | 0x18 | 13 |
| LINOT - LIN Driver Overtemperature Interrupt | 0x1A | 14 |
| HSOT - High-side Overtemperature Interrupt | 0x1C | 15 |
| LSOT - Low-side Overtemperature Interrupt | 0x1E | 16 |
| HOT - HSUP Overtemperature Interrupt | 0x20 | 17 |
| HVI - High Voltage Interrupt | 0x22 | 18 |
| VROVI - Voltage Regulator Overvoltage Interrupt | 0x24 | 19 (lowest) |

## 5.7.2 Interrupt Sources

### 5.7.2.1 Voltage Status Interrupt (VSI)

The Voltage Status Interrupt - VSI combines the five interrupt sources of the Voltage Status Register. It is only available in the Interrupt Source Register (ISR). Acknowledge the interrupt by reading the Voltage Status Register - VSR. To issue a new interrupt, the condition has to vanish and occur again. See Section 5.5, “Power Supply" for details on the Voltage Status Register including masking information.

### 5.7.2.2 Low Voltage Interrupt (LVI)

Acknowledge the interrupt by reading the Voltage Status Register - VSR. To issue a new interrupt, the condition has to vanish and occur again. See Section 5.5, “Power Supply" for details on the Voltage Status Register including masking information.

### 5.7.2.3 Voltage Regulator High Temperature Interrupt (HTI)

Acknowledge the interrupt by reading the Voltage Status Register - VSR. To issue a new interrupt, the condition has to vanish and occur again. See Section 5.5, “Power Supply" for details on the Voltage Status Register including masking information.

### 5.7.2.4 Low Battery Interrupt (LBI)

Acknowledge the interrupt by reading the Voltage Status Register - VSR. To issue a new interrupt, the condition has to vanish and occur again. See Section 5.5, “Power Supply" for details on the Voltage Status Register including masking information.

### 5.7.2.5 TIM Channel 0 Interrupt (CH0)

See Section 5.19, “Basic Timer Module - TIM (TIM16B4C)".

### 5.7.2.6 TIM Channel 1 Interrupt (CH1)

See Section 5.19, “Basic Timer Module - TIM (TIM16B4C)".

### 5.7.2.7 TIM Channel 2 Interrupt (CH2)

See Section 5.19, “Basic Timer Module - TIM (TIM16B4C)".

### 5.7.2.8 TIM Channel 3 Interrupt (CH3)

See Section 5.19, “Basic Timer Module - TIM (TIM16B4C)".

#### 5.7.2.9 TIM Timer Overflow Interrupt (TOV)

See Section 5.19, “Basic Timer Module - TIM (TIM16B4C)".

#### 5.7.2.10 SCI Error Interrupt (ERR)

See Section 5.16, “Serial Communication Interface (S08SCIV4)".

#### 5.7.2.11 SCI Transmit Interrupt (TX)

See Section 5.16, “Serial Communication Interface (S08SCIV4)".

#### 5.7.2.12 SCI Receive Interrupt (RX)

See Section 5.16, “Serial Communication Interface (S08SCIV4)".

#### 5.7.2.13 LIN Driver Overtemperature Interrupt (LINOT)

Acknowledge the interrupt by reading the LIN Register - LINR. To issue a new interrupt, the condition has to vanish and occur again. See Section 5.15, “LIN Physical Layer Interface - LIN" for details on the LIN Register including masking information.

#### 5.7.2.14 High-side Overtemperature Interrupt (HSOT)

Acknowledge the interrupt by reading the High-side Status Register - HSSR. To issue a new interrupt, the condition has to vanish and occur again. See Section 5.12, “High-side Drivers - HS" for details on the High-side Status Register including masking information.

#### 5.7.2.15 Low-side Overtemperature Interrupt (LSOT)

Acknowledge the interrupt by reading the Low-side Status Register - LSSR. To issue a new interrupt, the condition has to vanish and occur again. See Section 5.13, “Low-side Drivers - LSx" for details on the Low-side Status Register including masking information.

#### 5.7.2.16 HSUP Overtemperature Interrupt (HOT)

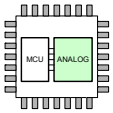
Acknowledge the interrupt by reading the Hall Supply Register - HSR. To issue a new interrupt, the condition has to vanish and occur again. See Section 5.11, “Hall Sensor Supply Output - HSUP" for details on the Hall Supply Register including masking information.

#### 5.7.2.17 High Voltage Interrupt (HVI)

Acknowledge the interrupt by reading the Voltage Status Register - VSR. To issue a new interrupt, the condition has to vanish and occur again. See Section 5.5, “Power Supply" for details on the Voltage Status Register including masking information.

#### 5.7.2.18 Voltage Regulator Overvoltage Interrupt (VROVI)

Acknowledge the interrupt by reading the Voltage Status Register - VSR. To issue a new interrupt, the condition has to vanish and occur again. See Section 5.5, “Power Supply" for details on the Voltage Status Register including masking information.




## 5.8Resets

To protect the system during critical events, the MM912_634 analog die drives the $\overline{\text{RESET_A}}$ pin low during the presence of the reset condition. In addition, the $\overline{\text{RESET_A}}$ pin is monitored for external reset events. To match the MCU, the $\overline{\text{RESET_A}}$ pin is based on the VDDX voltage level.

After an internal reset condition has gone, the $\overline{\text{RESET_A}}$ stays low for an additional time $t_{RST}$ before being released. Entering reset mode causes all MM912_634 analog die registers to be initialized to their RESET default. The only registers with valid information are the Reset Status Register (RSR) and the Wake-up Source Register (WUS).

### 5.8.1 Reset Sources

In the MM912_634 six reset sources exist.

#### 5.8.1.1 POR - Analog Die Power On Reset

To indicate the device power supply (VS1) was below VPOR or the MM912_634 analog die was powered up, the POR condition is set. See Section 5.4, “Modes of Operation".

#### 5.8.1.2 LVR - Low Voltage Reset - VDD

With the VDD voltage regulator output voltage falling below VLVR, the Low Voltage Reset condition becomes present. As the VDD Regulator is shutdown once a LVRX condition is detected, The actual cause could be also a low voltage condition at the VDDX regulator. See Section 5.5, “Power Supply".

#### 5.8.1.3 LVRX - Low Voltage Reset - VDDX

With the VDDX voltage regulator output voltage falling below VLVRX, the Low Voltage Reset condition becomes present. See Section 5.5, “Power Supply".

#### 5.8.1.4 WUR - Wake-up Reset

While in Sleep mode, any active wake-up event causes a MM912_634 analog die transition from Sleep to Reset Mode. To determine the wake-up source, refer to Section 5.9, “Wake-up / Cyclic Sense".

#### 5.8.1.5 EXR - External Reset

Any low level voltage at the $\overline{\text{RESET_A}}$ pin with a duration > $t_{RSTDF}$ issues an External Reset event. This reset source is also active in Stop mode.

#### 5.8.1.6 WDR - Watchdog Reset

Any incorrect serving if the MM912_634 analog die Watchdog results in a Watchdog Reset. Refer to the Section 5.10, “Window Watchdog" for details.

### 5.8.2 Register Definition

#### 5.8.2.1 Reset Status Register (RSR)

**Table 91. Reset Status Register (RSR)**

Offset[74] 0x15 Access: User read

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | WDR | EXR | WUR | LVRX | LVR | POR |
| W | | | | | | | | |

Note:

75. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 92. RSR - Register Field Descriptions**

| Field | Description |
|---|---|
| 5 - WDR | Watchdog Reset - Reset caused by an incorrect serving of the watchdog. |
| 4 - EXR | External Reset - Reset caused by the $\overline{\text{RESET_A}}$ pin driven low externally for > $t_{RSTDF}$. |
| 3 - WUR | Wake-up Reset - Reset caused by a wake-up from Sleep mode. To determine the wake-up source, refer to Wake-up / Cyclic Sense. |
| 2 - LVRX | Low Voltage Reset VDDX - Reset caused by a low voltage condition monitored at the VDDX output. |
| 1 - LVR | Low Voltage Reset VDD - Reset caused by a low voltage condition monitored at the VDD output.[76] |
| 0 - POR | Power On Reset - Supply Voltage was below $V_{POR}$. |

Note:

76. As the VDD Regulator is shutdown once a LVRX condition is detected, The actual cause could be also a low voltage condition at the VDDX regulator.

Reading the Reset Status register clears the information inside. Writing has no effect. LVR and LVRX are masked when POR or WUR are set.

## 5.9Wake-up / Cyclic Sense

To wake-up the MM912_634 analog die from Stop or Sleep mode, several wake-up sources are implemented. As described in , a wake-up from Stop mode results in an interrupt (D2DINT) to the MCU combined with a transition to Normal mode. A wake-up from Sleep mode results in a transition to Reset mode. In any case, the source of the wake-up can be identified by reading the Wake-up Source Register (WSR). The Wake-up Source Register (WSR) has to be read after a wake-up condition to execute a new STOP mode command. Two base clock cycles ($f_{BASE}$) delay are required between the WSR read and MCR write.

In general, there are the following seven main wake-up sources:

- Wake-up by a state change of one of the Lx inputs
- Wake-up by a state change of one of the Lx inputs during a cyclic sense
- Wake-up due to a forced wake-up
- Wake-up by the LIN module
- Wake-up by D2D interface (Stop mode only)
- Wake-up due to internal / external Reset (Stop mode only)
- Wake-up due to loss of supply voltage (Sleep mode only)

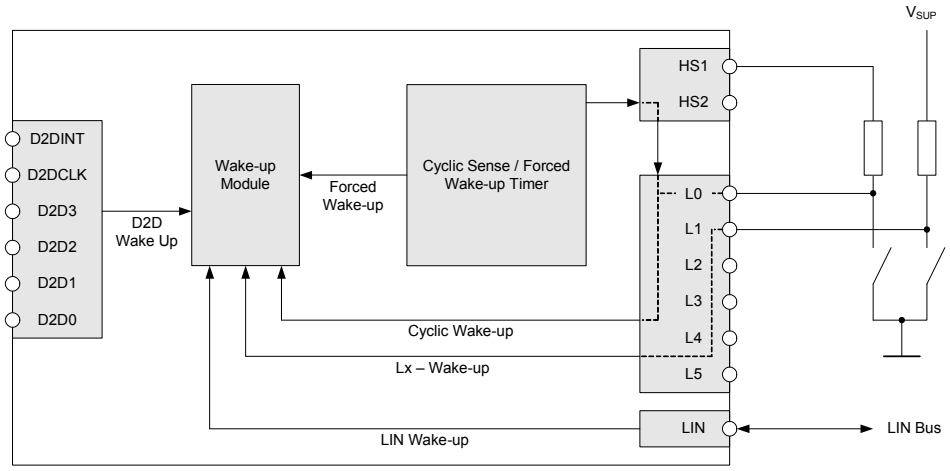



**Figure 19. Wake-up Sources**

### 5.9.1 Wake-up Sources

#### 5.9.1.1 Lx - Wake-up (Cyclic Sense Disabled)

Any state digital change on a Wake-up Enabled Lx input issues a wake-up. In order to select and activate a Wake-up Input (Lx), the Wake-up Control Register (WCR) must be configured with appropriate LxWE inputs enabled or disabled before entering low power mode. The Lx - Wake-up may be combined with the Forced Wake-up.

Note: Selecting a Lx Input for wake-up disables a selected analog input once entering low power mode.

#### 5.9.1.2 Lx - Cyclic Sense Wake-up

To reduce external power consumption during low power mode a cyclic wake-up has been implemented. Configuring the Timing Control Register (TCR) a specific cycle time can be selected to implement a periodic switching of the HS1 or HS2 output with the corresponding detection of an Lx state change. Any configuration of the HSx in the High-side Control Register (HSCR) is ignored when entering low power mode. The Lx - Cyclic Sense Wake-up may be combined with the Forced Wake-up. In case both (forced and Lx change) events are present at the same time, the Forced Wake-up is indicated as Wake-up source.

**NOTE**

Once Cyclic Sense is configured (CSSEL!=0), the state change is only recognized from one cyclic sense event to the next. The additional accuracy of the cyclic sense cycle by the WD clock trimming is only active during STOP mode. There is no trimmed clock available during SLEEP mode.

#### 5.9.1.3 Forced Wake-up

Configuring the Forced Wake-up Multiplier (FWM) in the Timing Control Register (TCR) enables the forced wake-up based on the selected Cyclic Sense Timing (CST). Forced Wake-up can be combined with all other wake-up sources considering the timing dependencies.

#### 5.9.1.4 LIN - Wake-up

While in Low-Power mode the MM912_634 analog die monitors the activity on the LIN bus. A dominant pulse longer than $t_{PROPWL}$ followed by a dominant to recessive transition causes a LIN Wake-up. This behavior protects the system from a short-to-ground bus condition.

#### 5.9.1.5 D2D - Wake-up (Stop Mode only)

Receiving a Normal mode request via the D2D interface (MODE=0, Mode Control Register (MCR)) results in a wake-up from stop mode. As this condition is controlled by the MCU, no wake-up status bit does indicate this wake-up source.

#### 5.9.1.6 Wake-up Due to Internal / External Reset (STOP Mode Only)

While in Stop mode, a Reset due to a VDD low voltage condition or an external Reset applied on the $\overline{\text{RESET_A}}$ pin results in a Wake-up with immediate transition to Reset mode. In this case, the LVR or EXR bits in the Reset Status Register indicates the source of the event.

#### 5.9.1.7 Wake-up Due to Loss of Supply Voltage (SLEEP Mode Only)

While in Sleep mode, a supply voltage VS1 < VPOR results in a transition to Power On mode.

### 5.9.2 Register Definition

#### 5.9.2.1 Wake-up Control Register (WCR)

**Table 93. Wake-up Control Register (WCR)**

Offset[76] 0x12 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | CSSEL | | L5WE | L4WE | L3WE | L2WE | L1WE | L0WE |
| Reset | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

Note:
77. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 94. WCR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7-6<br>CSSEL | Cyclic Sense Select - Configures the HSx output for the cyclic sense event. Note, with no LxWE selected - only the selected HSx output is switched periodically, no Lx state change would be detected. For all configurations, the Forced Wake-up can be activated in parallel in Section 5.9.2.2, "Timing Control Register (TCR)"<br>00 - Cyclic Sense Off<br>01 - Cyclic Sense with periodic HS1on<br>10 - Cyclic Sense with periodic HS2 on<br>11 - Cyclic Sense with periodic HS1 and HS2 on. |
| 5 - L5WE | Wake-up Input 5 Enabled - L5 Wake-up Select Bit.<br>0 - L5 Wake-up Disabled<br>1 - L5 Wake-up Enabled |
| 4 - L4WE | Wake-up Input 4 Enabled - L4 Wake-up Select Bit.<br>0 - L4 Wake-up Disabled<br>1 - L4 Wake-up Enabled |
| 3 - L3WE | Wake-up Input 3 Enabled - L3 Wake-up Select Bit.<br>0 - L3Wake-up Disabled<br>1 - L3 Wake-up Enabled |
| 2- L2WE | Wake-up Input 2 Enabled - L2 Wake-up Select Bit.<br>0 - L2 Wake-up Disabled<br>1 - L2 Wake-up Enabled |

**Table 94. WCR - Register Field Descriptions (continued)**

| Field | Description |
|---|---|
| 1 - L1WE | Wake-up Input 1 Enabled - L1 Wake-up Select Bit.<br>0 - L1 Wake-up Disabled<br>1 - L1 Wake-up Enabled |
| 0 - L0WE | Wake-up Input 0 Enabled - L0 Wake-up Select Bit.<br>0 - L0 Wake-up Disabled<br>1 - L0 Wake-up Enabled |

### 5.9.2.2 Timing Control Register (TCR)

**Table 95. Timing Control Register (TCR)**

Offset[78] 0x13 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | FWM | | | | CST | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
78. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 96. TCR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7-4<br>FWM | Forced Wake-up Multiplicator - Configures the multiplicator for the forced wake-up. The selected multiplicator (FWM!=0) forces a wake-up every FWM x CST ms. With this implementation, Forced and Cyclic wake-up can be performed in parallel with the cyclic sense period <= the forced wake-up period.<br>0000 - Forced Wake-up = Off<br>0001 - 1x<br>0010 - 2x<br>0011 - 4x<br>0100 - 8x<br>0101 - 16x<br>0110 - 32x<br>0111 - 64x<br>1000 - 128x<br>1001 - 256x<br>1010 - 512x<br>1011 - 1024x<br>11xx - not implemented (Forced Wake Multiplicator = 1024x) |
| 3-0<br>CST | Cyclic Sense Timing[79]<br>0000 - 1.0 ms<br>0001 - 2.0 ms<br>0010 - 5.0 ms<br>0011 - 10 ms<br>0100 - 20 ms<br>0101 - 50 ms<br>0110 - 100 ms<br>0111 - 200 ms<br>1000 - 500 ms<br>1001 - 1000 ms<br>1010 - 1111 - not implemented (Cyclic Sense Timing = 1000 ms) |

Note:
79. Cyclic Sense Timing with Accuracy CSAC and CSACT.

### 5.9.2.3 Wake-up Source Register (WSR)

**Table 97. Wake-up Source Register (WSR)**

Offset[80] 0x14 Access: User read

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | FWU | LINWU | L5WU | L4WU | L3WU | L2WU | L1WU | L0WU |
| W | | | | | | | | |

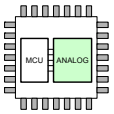
Note:
80. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 98. WSR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7 - FWU | Forced Wake-up - Wake-up caused by a forced wake-up |
| 6 - LINWU | LIN Wake-up - Wake-up caused by a LIN wake-up |
| 5 - L5WU | L5 Wake-up - Wake-up caused by a state change of the L6 Input |
| 4 - L4WU | L4 Wake-up - Wake-up caused by a state change of the L5 Input |
| 3 - L3WU | L3 Wake-up - Wake-up caused by a state change of the L4 Input |
| 2 - L2WU | L2 Wake-up - Wake-up caused by a state change of the L3 Input |
| 1 - L1WU | L1 Wake-up - Wake-up caused by a state change of the L2 Input |
| 0 - L0WU | L0 Wake-up - Wake-up caused by a state change of the L1 Input |

Reading the WSR clears the wake-up status bit(s). Writing has no effect. The Wake-up Source Register (WSR) has to be read after a wake-up condition, to execute a new STOP mode command. Two base clock cycles ($f_{BASE}$) delays are required between the WSR read and the MCR write.

## 5.10 Window Watchdog




The MM912_634 analog die includes a configurable window watchdog, which is active in Normal mode. The watchdog module is based on a separate clock source ($f_{BASE}$) operating independent from the MCU based D2DCLK clock. The watchdog timeout ($t_{WDTO}$) can be configured between 10 ms and 1280 ms (typ.) using the Watchdog Register (WDR).

During Low Power mode, the watchdog feature is not active, a D2D read during Stop mode has the WDOFF bit set.

To clear the watchdog counter, a alternating write must be performed to the Watchdog Service Register (WDSR). The first write after the $\overline{\text{RESET_A}}$ has been released has to be 0xAA. The next one must be 0x55.

After the $\overline{\text{RESET_A}}$ has been released, there be a standard (non-window) watchdog active with a fixed timeout of $tI_{WDTO}$. The Watchdog Window Open (WDWO) bit is set during this time and the window watchdog can be configured (WDR) without changing the initial timeout, and can be trimmed using the trim value given in the MCU trimming Flash section. See Section 5.26, “MM912_634 - Analog Die Trimming".

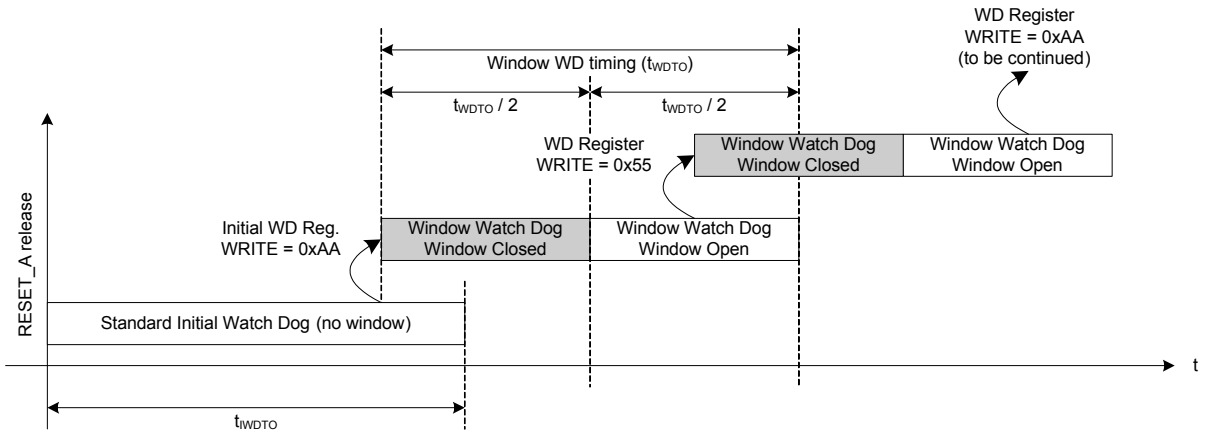



**Figure 20. MM912_634 Analog Die Watchdog Operation**

To enable the window watchdog, the initial counter reset has to be performed by writing 0xAA to the Watchdog Service Register (WDSR) before $tI_{WDTO}$ is reached. If the $tI_{WDTO}$ timeout is reached with no counter reset or a value different from 0xAA was written to the WDSR, a watchdog reset occurs.

Once entering Window Watchdog mode, the first half of the time $t_{WDTO}$ forbids a counter reset. To reset the watchdog counter, an alternating write of 0x55 and 0xAA must be performed within the second half of the $t_{WDTO}$. A Window Open (WDWO) flag indicates the current status of the window. A timeout or wrong value written to the WDSR forces a watchdog reset.

For debug purpose, the watchdog can be completely disabled by applying VTST to the TCLK pin while TEST_A is grounded. The watchdog is disabled as long as VTST is present. The watchdog is guaranteed functional for VTSTEN. The WDOFF bit indicates the watchdog being disabled. The WDSR register resets to default once the watchdog is disabled. Once the watchdog is re-enabled, the initial watchdog sequence has to be performed.

During Low Power mode, the Watchdog clock is halted and the Watchdog Service Register (WDSR) is reset to the default state.

### 5.10.1 Register Definition

#### 5.10.1.1 Watchdog Register (WDR)

**Table 99. Watchdog Register (WDR)**

Offset[(76)] 0x10 Access: User read/write

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

Note:

81. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 100. WDR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7 - WDOFF | Watchdog Off - Indicating the Watchdog module is being disabled externally. |

**Table 100. WDR - Register Field Descriptions (continued)**

| Field | Description |
|---|---|
| 6 - WDWO | Watchdog Window Open - Indicating the Watchdog Window is currently open for counter reset. |
| 2-0<br>WDTO[2:0] | Watchdog Timeout Configuration - configuring the Watchdog timeout duration $t_{WDTO}$.<br>000 - 10 ms<br>001 - 20 ms<br>010 - 40 ms<br>011 - 80 ms<br>100 - 160 ms<br>101 - 320 ms<br>110 - 640 ms<br>111 - 1280 ms |

### 5.10.1.2 Watchdog Service Register (WDSR)

**Table 101. Watchdog Service Register (WDSR)**

Offset[82] 0x11 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | WDSR | | | | | | | |
| Reset | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

Note:
82. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 102. WDSR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7-0<br>WDSR | Watchdog Service Register - Writing this register with the correct value (0xAA alternating 0x55) while the window is open resets the watchdog counter. Writing the register while the watchdog is disabled has no effect. |

## 5.11 Hall Sensor Supply Output - HSUP

To supply Hall Effect Sensors or similar external loads, the HSUP output is implemented. To reduce power dissipation inside the device, the output is implemented as a switchable Voltage Regulator, internally connected to the VS1 supply input. For protection, an Overtemperature Shutdown and a Current Limitation is implemented. A write to the Hall Supply Register (HSR), when the overtemperature condition is gone, re-enables the Hall Supply Output.

The HSUP output is active only during Normal mode. A capacitor CHSUP is recommended for operation.

### 5.11.1 Register Definition

#### 5.11.1.1 Hall Supply Register (HSR)

**Table 103. Hall Supply Register (HSR)**

Offset[82] 0x38 Access: User read/write

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

Note:

83. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 104. HSR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7 - HOTIE | Hall Supply Overtemperature Interrupt Enable |
| 6 - HOTC | Hall Supply Overtemperature Condition present. During the event, the Hall Supply is shut down. Reading the register clears the HOT flag if present. See Section 5.7, “Interrupts" for details. |
| 0 - HSUPON | Hall Supply On:<br>0 - Hall Supply Regulator disabled<br>1 - Hall Supply Regulator enabled |

## 5.12 High-side Drivers - HS

These outputs are two high-side drivers, intended to drive small resistive loads or LEDs incorporating the following features:

- PWM capability via the PWM Module
- Open load detection
- Current limitation
- Overtemperature shutdown (with maskable interrupt)
- High voltage shutdown - HVI (software maskable)
- Cyclic-Sense, See Section 5.9, “Wake-up / Cyclic Sense"

### 5.12.1 Open Load Detection

Each high-side driver signals an open load condition if the current through the high-side is below the OpenLoad current threshold. The open load condition is indicated with the bits HS1OL and HS2OL in the High-side Status Register (HSSR).

When the high-side is in OFF state, the open load detection function is not operating. When reading the HSSR register while the high-side is operating in PWM and is in the OFF state, the HS1OL and HS2OL bits does not indicate open load.

### 5.12.2 Current Limitation

Each high-side driver has an output current limitation. In combination with the overtemperature shutdown the high-side drivers are protected against overcurrent and short-circuit failures. The driver operating in the current limitation area is indicated with the bits HS1CL and HS2CL in the High-side Status Register (HSSR).

### 5.12.3 Overtemperature Protection (HS Interrupt)

Both high-side drivers are protected against overtemperature. In overtemperature conditions, both high-side drivers are shutdown and the event is latched in the Interrupt Control Module. The shutdown is indicated as HS Interrupt in the Interrupt Source Register (ISR).

A thermal shutdown of the high-side drivers is indicated by setting the HSOT bit in the High-side Status Register (HSSR). A write to the High-side Control Register (HSCR), when the overtemperature condition is gone, re- enables the high-side drivers.

### 5.12.4 High Voltage Shutdown

In case of a high voltage condition (HVI), and if the high voltage shutdown is enabled (bit HVSDE in the High-side Control Register (HSCR) is set), both high-side drivers are shut down. A write to the High-side Control Register (HSCR), when the high voltage condition is gone, re-enables the high-side drivers.

### 5.12.5 Sleep And Stop Mode

The high-side drivers can be enabled to operate in Sleep and Stop mode for cyclic sensing. See Section 5.9, “Wake-up / Cyclic Sense".

### 5.12.6 PWM Capability

Section 5.14, “PWM Control Module (PWM8B2C)"

### 5.12.7 Register Definition

#### 5.12.7.1 High-side Control Register (HSCR)

**Table 105. High-side Control Register (HSCR)**

Offset[82] 0x28 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | HSOTIE | HSHVSDE | PWMCS2 | PWMCS1 | PWMHS2 | PWMHS1 | HS2 | HS1 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:

84. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 106. HSCR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7 - HSOTIE | High-side Overtemperature Interrupt Enable |
| 6 - HSHVSDE | High-side High Voltage Shutdown. Once enabled, both high-sides shuts down when a high voltage condition - HVC is present. See Section 5.5, “Power Supply" for the Voltage Status Register. |
| 5 - PWMCS2 | PWM Channel Select HS2<br>0 - PWM Channel 0 selected as PWM Channel<br>1 - PWM Channel 1 selected as PWM Channel |
| 4 - PWMCS1 | PWM Channel Select HS1<br>0 - PWM Channel 0 selected as PWM Channel<br>1 - PWM Channel 1 selected as PWM Channel |
| 3 - PWMHS2 | PWM Enable for HS2<br>0 - PWM disabled on HS2<br>1 - PWM enabled on HS2 (Channel as selected with PWMCS2) |

**Table 106. HSCR - Register Field Descriptions (continued)**

| Field | Description |
|---|---|
| 2 - PWMHS1 | PWM Enable for HS1<br>0 - PWM disabled on HS1<br>1 - PWM enabled on HS1 (Channel as selected with PWMCS1) |
| 1 - HS2 (85) | HS2 Control<br>0 - HS2 disabled<br>1 - HS2 enabled |
| 0 - HS1 | HS2 Control<br>0 - HS1 disabled<br>1 - HS1 enabled |

Note:
85. When the high-side is in OFF state, the OpenLoad Detection function is not operating. When reading the HSSR register while the high-side is operating in PWM and is in the OFF state, the HS1OL and HS2OL bits does not indicate open load.

### 5.12.7.2 High-side Status Register (HSSR)

**Table 107. High-side Status Register (HSSR)**

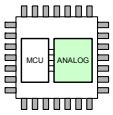
Offset(86) 0x29 Access: User read

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | HSOTC | 0 | 0 | 0 | HS2CL | HS1CL | HS2OL | HS1OL |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
86. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 108. HSSR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7 - HSOTC | High-side Overtemperature Condition present. Both drivers are turned off. Reading the register clears the HSOT interrupt flag if present. See Section 5.7, “Interrupts" for details. |
| 3 - HS2CL | High-side 2 Current Limitation |
| 2 - HS1CL | High-side 1 Current Limitation |
| 1 - HS2OL | High-side 2 Open Load |
| 1 - HS1OL | High-side 1Open Load |




## 5.13Low-side Drivers - LSx

### 5.13.1Introduction / Features

These outputs are two low-side drivers intended to drive relays (inductive loads) incorporating the following features:

- PWM capability
- Open load detection
- Current limitation
- Overtemperature shutdown (with maskable interrupt)
- Active clamp
- Independent VREG - High Voltage Shutdown

### 5.13.1.1 Block Diagram

The following Figure shows the basic structure of the LS drivers.

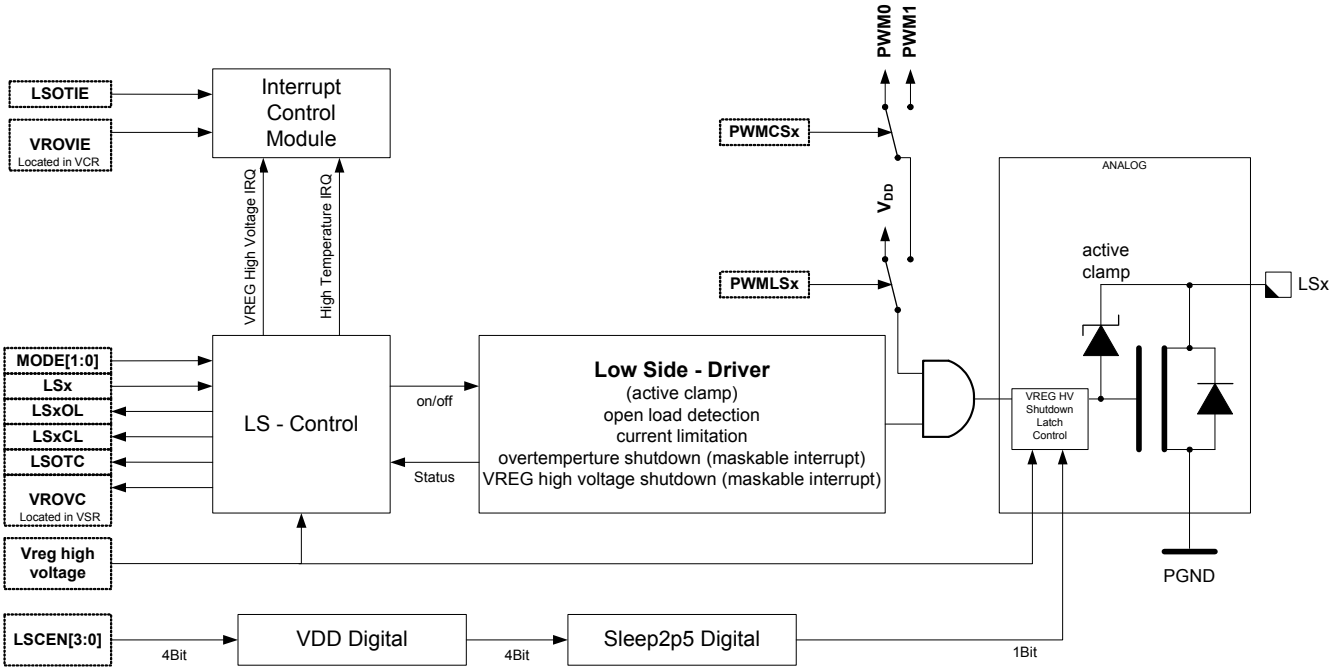



**Figure 21. Low-side Drivers - Block Diagram**

### 5.13.1.2 Modes of Operation

The Low-side module is active only in Normal mode; the Low-side drivers are disabled in Sleep and Stop mode.

## 5.13.2 External Signal Description

This section lists and describes the signals which do connect off-chip. Table 109 shows all the pins and their functions which are controlled by the low-side module.

**Table 109. Pin Functions and Priorities**

| Pin Name | Pin Function & Priority | I/O | Description | Pin Function after Reset |
|---|---|---|---|---|
| LS1 | High Voltage Output | O | Low-side Power Output Driver, Active Clamping | LS1 |
| LS2 | | O | | LS2 |

### 5.13.3 Memory Map and Registers

#### 5.13.3.1 Module Memory Map

Table 110 shows the register map of the Low-side Driver module. All Register addresses given are referenced to the D2D interface offset.

**Table 110. Low-side Module - Memory Map**

| Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0x30 LSCR | R | LSOTIE | 0 | PWMCS2 | PWMCS1 | PWMLS2 | PWMLS1 | LS2 | LS1 |
| | W | | | | | | | | |
| 0x31 LSSR | R | LSOTC | 0 | 0 | 0 | LS2CL | LS1CL | LS2OL | LS1OL |
| | W | | | | | | | | |
| 0x32 LSCEN | R | 0 | 0 | 0 | 0 | LSCEN | | | |
| | W | | | | | | | | |

#### 5.13.3.2 Register Descriptions

##### 5.13.3.2.1 Low-side Control Register (LSCR)

**Table 111. Low-side Control Register (LSCR)**

Offset[86] 0x30 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | LSOTIE | 0 | PWMCS2 | PWMCS1 | PWMLS2 | PWMLS1 | LS2 | LS1 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
87. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 112. LSCR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7 - LSOTIE | Low-side Overtemperature Interrupt Enable |
| 5 - PWMCS2 | PWM Channel Select LS2<br>0 - PWM Channel 0 selected as PWM Channel<br>1 - PWM Channel 1 selected as PWM Channel |
| 4 - PWMCS1 | PWM Channel Select LS1<br>0 - PWM Channel 0 selected as PWM Channel<br>1 - PWM Channel 1 selected as PWM Channel |
| 3 - PWMLS2 | PWM Enable for LS2<br>0 - PWM disabled on LS2<br>1 - PWM enabled on LS2 (Channel as selected with PWMCS2) |
| 2 - PWMLS1 | PWM Enable for LS1<br>0 - PWM disabled on LS1<br>1 - PWM enabled on LS1 (Channel as selected with PWMCS1) |
| 1 - LS2 | LS2 Enable; LSEN has to be written once to control the LS2 Driver |
| 0 - LS1 | LS1 Enable; LSEN has to be written once to control the LS1 Driver |

#### 5.13.3.2.2 Low-side Status Register (LSSR)

**Table 113. Low-side Status Register (LSSR)**

Offset[88] 0x31 Access: User read

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | LSOTC | 0 | 0 | 0 | LS2CL | LS1CL | LS2OL | LS1OL |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
88. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 114. LSSR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7 - LSOTC | Low-side Overtemperature condition present. Both drivers are turned off. Reading the register clears the LSOT interrupt flag if present. See Section 5.7, “Interrupts" for details. |
| 3 - LS2CL | Low-side 2 Current Limitation |
| 2 - LS1CL | Low-side 1 Current Limitation |
| 1 - LS2OL | Low-side 2 Open Load [89] |

Note:
89. When the low-side is in OFF state, the OpenLoad Detection function is not operating. When reading the LSSR register while the low-side is operating in PWM and is in the OFF state, the LS1OL and LS2OL bits do not indicate open load.

#### 5.13.3.2.3 Low-side Control Enable Register (LSCEN)

**Table 115. Low-side Enable Register (LSEN)**

Offset[90] 0x32 Access: User read/write

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

Note:
90. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 116. LSEN - Register Field Descriptions**

| Field | Description |
|---|---|
| 3-0 LSCEN | Low-side Control Enable - To allow the LS Control via LSx, the correct value has to be written into the LSCEN Register.<br>0x5 - Low-side Control Enabled<br>all other values - Low-side Control Disabled |

### 5.13.4 Functional Description

The Low-side switches are controlled by the bits LS1:2 in the Low-side Control Register (LSCR). In order to control the low-sides, the LSCEN register has to be correctly written once after RESET or VROV. To protect the device against overvoltage when an inductive load (relay) is turned off an active clamp circuit is implemented.

#### 5.13.4.1 Voltage Regulator Overvoltage Protection

To protect the application for an unintentional activation of the drivers in case of a voltage regulator overvoltage failure, the Low-side Drivers automatically shuts down if an overvoltage on one of the two regulators occurs.

The shutdown is fully handled in the analog section of the driver. This secures the feature in case the digital logic is damaged due to the overvoltage condition.

Once an overvoltage condition on one of the voltage regulators occurs, the LSx control bits in the Low-side Control Register (LSCR) is reset to 0. The Voltage Regulator Overvoltage Condition Bit (VROVC) in the Voltage Status Register (VSR) stays set as long as the condition is present. If the Voltage Regulator Overvoltage Interrupt was enabled (VROVIE=1), the VROV- Interrupt is issued. Reading the Voltage Regulator Overvoltage Condition Bit (VROVC) in the Voltage Status Register (VSR) clears the interrupt. To issue another VROV - Interrupt, the condition has to vanish and be present again.

To re-enable the Low-side Drivers after a Voltage Regulator Overvoltage condition occurred, first the LSCEN register has to be written with “0x05” - this information is processed through the main digital blocks, and would secure a minimum functionality before enabling the LS drivers again. In a second step, the LSx Control Bits in the Low-side Control Register (LSCR) must be enabled again after the overvoltage condition has vanished (VROVC=0).

**NOTE**

The overvoltage threshold has to be trimmed at system power up. Refer to Section 5.26.1.2.3, “Trimming Register 2 (CTR2)" for details. The default trim is worst case and may have disabled the LS function already. An initial LS enable would be needed.

#### 5.13.4.2 Open Load Detection

Each low-side driver signals an open load condition if the current through the low-side is below the open load current threshold. The open load condition is indicated with the bits LS1OL and LS2OL in the Low-side Status Register (LSSR). When the low-side is in OFF state, the OpenLoad Detection function is not operating. When reading the LSSR register while the low-side is operating in PWM and is in the OFF state, the LS1OL and LS2OL bits does not indicate open load.

#### 5.13.4.3 Current Limitation

Each Low-side driver has a current limitation. In combination with the overtemperature shutdown, the Low-side drivers are protected against overcurrent and short-circuit failures. The driver operates in current limitation, and is indicated with the bits LS1CL and LS2CL in the Low-side Status Register (LSSR). Note: If the drivers is operating in current limitation mode excessive power might be dissipated.

#### 5.13.4.4 Overtemperature Protection (LS Interrupt)

Both Low-side drivers are protected against overtemperature. In case of an overtemperature condition, both Low-side drivers are shut down and the event is latched in the Interrupt Control Module. The shutdown is indicated as LS Interrupt in the Interrupt Source Register (ISR). If the bit LSM is set in the Interrupt Mask Register (IMR) than an Interrupt (IRQ) is generated. A write to the Low-side Control Register (LSCR) re-enables the Low-side drivers when the overtemperature condition is gone.

### 5.13.5 PWM Capability

See Section 5.14, “PWM Control Module (PWM8B2C)".

### 5.13.6 Low-side Driver in Reverse Polarity

Freescale recommends the user implement their systems using an external diode to protect the low-side driver (LSx) from potential damage (see Figure 22) in the event of a reverse polarity condition. However, if the user chooses to implement their system without an external diode, and a reverse polarity condition were to occur, the current would flow through the body diode of the LSx driver and through the closing coil of the relay, as shown in Figure 23. The magnitude of current through this body diode is determined by the characteristics of this relay coil. Since this body diode is not specified to carry a sustained minimum amount of current, Freescale strongly recommends users implement their systems using an external reverse polarity protection diode.

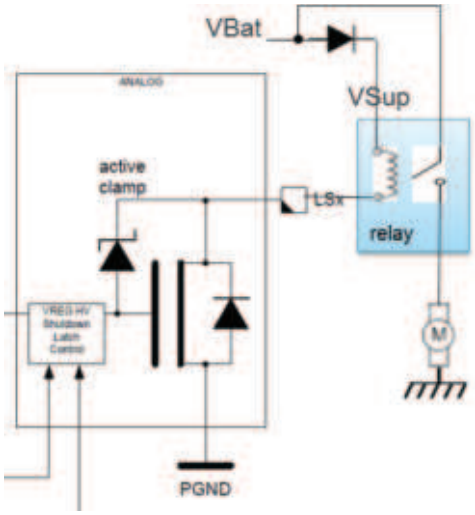



**Figure 22. Low-side Driver Implemented With External Reverse Polarity Diode**

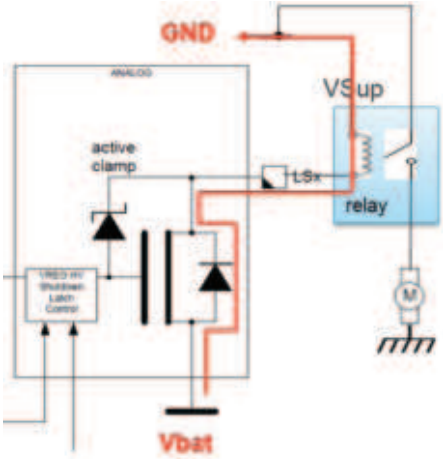



**Figure 23. Current Flow in Low-side Driver Under Reverse Polarity Condition Without External Diode**

## 5.14 PWM Control Module (PWM8B2C)

### 5.14.1 Introduction

To control the High-side (HS1, HS2) and the Low-side (LS1, LS2) duty cycle as well as the PTB2 output, the PWM module is implemented. Refer to the individual driver section for details on the use of the internal PWM1 and PWM0 signal (Section 5.12, “High-side Drivers - HS", Section 5.13, “Low-side Drivers - LSx", and Section 5.18, “General Purpose I/O - PTB[0…2]").

The PWM definition is based on the HC12 PWM definitions with some of the simplifications incorporated. The PWM module has two channels with independent controls of left and center aligned outputs on each channel.

Each of the two channels has a programmable period and duty cycle as well as a dedicated counter. A flexible clock select scheme allows a total of four different clock sources to be used with the counters. Each of the modulators can create independent continuous waveforms with software-selectable duty rates from 0% to 100%.

#### 5.14.1.1 Features

The PWM block includes these distinctive features:

- Two independent PWM channels with programmable periods and duty cycles
- Dedicated counter for each PWM channel
- Programmable PWM enable/disable for each channel
- Software selection of PWM duty pulse polarity for each channel
- Period and duty cycle are double buffered. Change takes effect when the end of the effective period is reached (PWM counter reaches zero), or when the channel is disabled
- Programmable center or left aligned outputs on individual channels
- Four clock sources (A, B, SA, and SB) provide for a wide range of frequencies
- Programmable clock select logic

#### 5.14.1.2 Modes of Operation

The PWM8B2C module does operate in Normal mode only.

#### 5.14.1.3 Block Diagram

Figure 24 shows the block diagram for the 8-bit 2-channel PWM block.

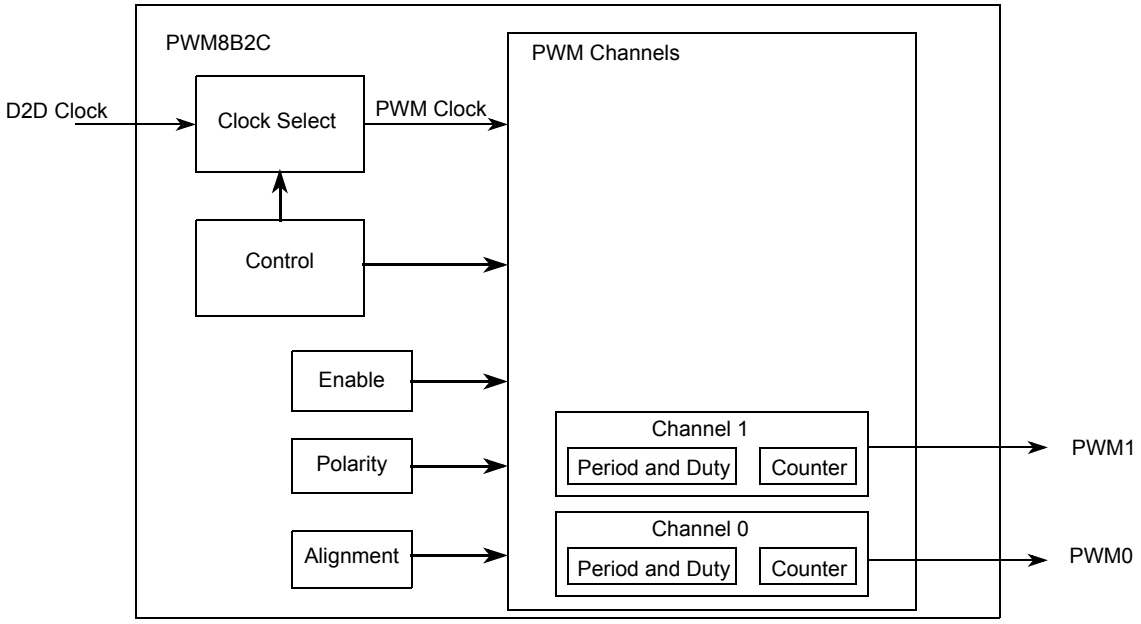



**Figure 24. PWM Block Diagram**

### 5.14.2 Signal Description

The PWM module has a total of two internal outputs to control the Low-side Outputs, the High-side Outputs and / or the PTB2 output with pulse width modulation. See Section 5.12, “High-side Drivers - HS", Section 5.13, “Low-side Drivers - LSx", and Section 5.18, “General Purpose I/O - PTB[0…2]" for configuration details.

**NOTE**

Based on the D2D clock speed, the PWM8B2C module is capable of generating PWM signal frequencies higher than the maximum output frequency of the connected driver (HS, LS). Refer to Section 4.6, “Dynamic Electrical Characteristics" for details. Do not exceed the driver maximum output frequency.

#### 5.14.2.1 D2DCLK

Die 2 Die Interface Clock.

#### 5.14.2.2 PWM1 — Pulse Width Modulator Channel 1

This signal serves as waveform output of PWM channel 1.

### 5.14.2.3 PWM0 — Pulse Width Modulator Channel 0

This signal serves as waveform output of PWM channel 0.

## 5.14.3 Register Descriptions

This section describes in detail all the registers and register bits in the PWM module. Reserved bits within a register are always read as 0 and the write is unimplemented. Unimplemented functions are indicated by shading the bit.

**Table 117. PWM Register Summary**

| Name / Offset[90] | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0x60 PWMCTL | R | CAE1 | CAE0 | PCLK1 | PCLK0 | PPOL1 | PPOL0 | PWME1 | PWME0 |
| | W | | | | | | | | |
| 0x61 PWMPRCLK | R | 0 | PCKB2 | PCKB1 | PCKB0 | 0 | PCKA2 | PCKA1 | PCKA0 |
| | W | | | | | | | | |
| 0x62 PWMSCLA | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | W | | | | | | | | |
| 0x63 PWMSCLB | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | W | | | | | | | | |
| 0x64 PWMCNT0 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x65 PWMCNT1 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x66 PWMPER0 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | W | | | | | | | | |
| 0x67 PWMPER1 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | W | | | | | | | | |
| 0x68 PWMDTY0 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | W | | | | | | | | |
| 0x69 PWMDTY1 | R | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | W | | | | | | | | |

Note:

91. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

### 5.14.3.1 PWM Control Register (PWMCTL)

**Table 118. PWM Control Register (PWMCTL)**

Offset[92] 0x60 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | CAE1 | CAE0 | PCLK1 | PCLK0 | PPOL1 | PPOL0 | PWME1 | PWME0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
92. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 119. PWMCTL - Register Field Descriptions**

| Field | Description |
|---|---|
| 7–6<br>CAE[1:0] | Center Aligned Output Modes on Channels 1–0<br>0 Channels 1–0 operate in left aligned output mode.<br>1 Channels 1–0 operate in center aligned output mode. |
| 5<br>PCLK1 | Pulse Width Channel 1 Clock Select<br>0 Clock B is the clock source for PWM channel 1.<br>1 Clock SB is the clock source for PWM channel 1. |
| 4<br>PCLK0 | Pulse Width Channel 0 Clock Select<br>0 Clock A is the clock source for PWM channel 0.<br>1 Clock SA is the clock source for PWM channel 0. |
| 3–2<br>PPOL[1:0] | Pulse Width Channel 1–0 Polarity Bits<br>0 PWM channel 1–0 outputs are low at the beginning of the period, then go high when the duty count is reached.<br>1 PWM channel 1–0 outputs are high at the beginning of the period, then go low when the duty count is reached. |
| 1-0<br>PWME[1:0] | Pulse Width Channel 1–0 Enable<br>0 Pulse width channel 1–0 is disabled.<br>1 Pulse width channel 1–0 is enabled. The pulse modulated signal becomes available at PWM, output bit 1 when its clock source begins its next cycle. |

#### 5.14.3.1.1 PWM Enable (PWMEx)

Each PWM channel has an enable bit (PWMEx) to start its waveform output. When any of the PWMEx bits are set (PWMEx = 1), the associated PWM output is enabled immediately. However, the actual PWM waveform is not available on the associated PWM output until its clock source begins its next cycle, due to the synchronization of PWMEx and the clock source.

**NOTE**

The first PWM cycle after enabling the channel can be irregular. If both PWM channels are disabled (PWME1–0 = 0), the prescaler counter shuts off for power savings.

#### 5.14.3.1.2 PWM Polarity (PPOLx)

The starting polarity of each PWM channel waveform is determined by the associated PPOLx bit. If the polarity bit is one, the PWM channel output is high at the beginning of the cycle and then goes low when the duty count is reached. Conversely, if the polarity bit is zero, the output starts low and then goes high when the duty count is reached.

**NOTE**

PPOLx register bits can be written anytime. If the polarity changes while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition

#### 5.14.3.1.3 PWM Clock Select (PCLKx)

Each PWM channel has a choice of two clocks to use as the clock source for this channel as described by the following.

**NOTE**

Register bits PCLK0 and PCLK1 can be written anytime. If a clock select changes while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition.

#### 5.14.3.1.4 PWM Center Align Enable (CAEx)

The CAEx bits select either center aligned outputs or left aligned output for both PWM channels. If the CAEx bit is set to a one, the corresponding PWM output is center aligned. If the CAEx bit is cleared, the corresponding PWM output is left aligned. See Section 5.14.4.2.5, “Left Aligned Outputs" and Section 5.14.4.2.6, “Center Aligned Outputs" for a more detailed description of the PWM output modes.

**NOTE**

Write these bits only when the corresponding channel is disabled.

### 5.14.3.2 PWM Prescale Clock Select Register (PWMPRCLK)

This register selects the prescale clock source for clocks A and B independently.

**Table 120. PWM Prescale Clock Select Register (PWMPRCLK)**

Offset[93] 0x61 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | PCKB2 | PCKB1 | PCKB0 | 0 | PCKA2 | PCKA1 | PCKA0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
93. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 121. PWMPRCLK - Register Field Descriptions**

| Field | Description |
|---|---|
| 6–4<br>PCKB[2:0] | Prescaler Select for Clock B — Clock B is one of two clock sources which can be used for channel 1. These three bits determine the rate of clock B, as shown in Table 122. |
| 2–0<br>PCKA[2:0] | Prescaler Select for Clock A — Clock A is one of two clock sources which can be used for channel 0. These three bits determine the rate of clock A, as shown in Table 123. |

**Table 122. Clock B Prescaler Selects**

| PCKB2 | PCKB1 | PCKB0 | Value of Clock B |
|---|---|---|---|
| 0 | 0 | 0 | D2D clock |
| 0 | 0 | 1 | D2D clock / 2 |
| 0 | 1 | 0 | D2D clock / 4 |
| 0 | 1 | 1 | D2D clock / 8 |
| 1 | 0 | 0 | D2D clock / 16 |
| 1 | 0 | 1 | D2D clock / 32 |
| 1 | 1 | 0 | D2D clock / 64 |
| 1 | 1 | 1 | D2D clock / 128 |

**Table 123. Clock A Prescaler Selects**

| PCKA2 | PCKA1 | PCKA0 | Value of Clock A |
|---|---|---|---|
| 0 | 0 | 0 | D2D clock |
| 0 | 0 | 1 | D2D clock / 2 |
| 0 | 1 | 0 | D2D clock / 4 |
| 0 | 1 | 1 | D2D clock / 8 |
| 1 | 0 | 0 | D2D clock / 16 |
| 1 | 0 | 1 | D2D clock / 32 |
| 1 | 1 | 0 | D2D clock / 64 |
| 1 | 1 | 1 | D2D clock / 128 |

**NOTE**

PCKB2–0 and PCKA2–0 register bits can be written anytime. If the clock pre-scale is changed while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition.

### 5.14.3.3 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 by two.

Clock SA = Clock A / (2 * PWMSCLA)

**NOTE**

When PWMSCLA = $00, PWMSCLA value is considered a full scale value of 256. Clock A is thus divided by 512.

Any value written to this register causes the scale counter to load the new scale value (PWMSCLA)

.

**Table 124. PWM Scale A Register (PWMSCLA)**

Offset[94] 0x62 Access: User read/write

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

Note:

94. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

### 5.14.3.4 PWM Scale B Register (PWMSCLB)

PWMSCLB is the programmable scale value used in scaling clock B to generate clock SB. Clock SB is generated by taking clock B, dividing it by the value in the PWMSCLB register and dividing by two.

Clock SB = Clock B / (2 * PWMSCLB)

**NOTE**

When PWMSCLB = $00, PWMSCLB value is considered a full scale value of 256. Clock B is thus divided by 512.

Any value written to this register causes the scale counter to load the new scale value (PWMSCLB).

**Table 125. PWM Scale B Register (PWMSCLB)**

Offset[95] 0x63 Access: User read/write

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

Note:
95. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

### 5.14.3.5 PWM Channel Counter Registers (PWMCNTx)

Each channel has a dedicated 8-bit up/down counter, which runs at the rate of the selected clock source. The counter can be read at any time without affecting the count or the operation of the PWM channel. In left aligned output mode, the counter counts from 0 to the value in the period register - 1. In center aligned output mode, the counter counts from 0 up to the value in the period register and then back down to 0.

Any value written to the counter causes the counter to reset to $00, the counter direction to be set to up, the immediate load of both duty and period registers with values from the buffers, and the output to change according to the polarity bit. The counter is also cleared at the end of the effective period (see Section 5.14.4.2.5, “Left Aligned Outputs" and Section 5.14.4.2.6, “Center Aligned Outputs" for more details). When the channel is disabled (PWMEx = 0), the PWMCNTx register does not count. When a channel becomes enabled (PWMEx = 1), the associated PWM counter starts at the count in the PWMCNTx register. For more detailed information on the operation of the counters, see Section 5.14.4.2.4, “PWM Timer Counters".

**NOTE**

Writing to the counter while the channel is enabled can cause an irregular PWM cycle to occur.

**Table 126. PWM Channel Counter Registers (PWMCNTx)**

Offset[96] 0x64/0x65 Access: User read/write

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

Note:
96. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

### 5.14.3.6 PWM Channel Period Registers (PWMPERx)

There is a dedicated period register for each channel. The value in this register determines the period of the associated PWM channel.

The period registers for each channel are double buffered, so if they change while the channel is enabled, the change does NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to $00)
- The channel is disabled

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

**NOTE**

Reads of this register return the most recent value written. Reads do not necessarily return the value of the currently active period due to the double buffering scheme.

See Section 5.14.4.2.3, “PWM Period and Duty" for more information.

To calculate the output period, take the selected clock source period for the channel of interest (A, B, SA, or SB) and multiply it by the value in the period register for this channel:

- Left aligned output (CAEx = 0)
- PWMx Period = Channel Clock Period * PWMPERx Center Aligned Output (CAEx = 1)
- PWMx Period = Channel Clock Period * (2 * PWMPERx)

For boundary case programming values, refer to Section 5.14.4.2.7, “PWM Boundary Cases".

**Table 127. PWM Channel Period Registers (PWMPERx)**

Offset[97] 0x66/0x67 Access: User read/write

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

Note:
97. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

### 5.14.3.7 PWM Channel Duty Registers (PWMDTYx)

There is a dedicated duty register for each channel. The value in this register determines the duty of the associated PWM channel. The duty value is compared to the counter and if it is equal to the counter value a match occurs and the output changes state.

The duty registers for each channel are double buffered, so if they change while the channel is enabled, the change does NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to $00)
- The channel is disabled

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

**NOTE**

Reads of this register return the most recent value written. Reads do not necessarily return the value of the currently active duty due to the double buffering scheme.

See Section 5.14.4.2.3, “PWM Period and Duty" for more information.

**NOTE**

Depending on the polarity bit, the duty registers contains the count of either the high time or the low time. If the polarity bit is one, the output starts high and then goes low when the duty count is reached, so the duty registers contain a count of the high time. If the polarity bit is zero, the output starts low and then goes high when the duty count is reached, so the duty registers contain a count of the low time.

To calculate the output duty cycle (high time as a% of period) for a particular channel:

- Polarity = 0 (PPOL x =0)
  - Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
- Polarity = 1 (PPOLx = 1)
  - Duty Cycle = [PWMDTYx / PWMPERx] * 100%

For boundary case programming values, refer to Section 5.14.4.2.7, “PWM Boundary Cases".

**Table 128. PWM Channel Duty Registers (PWMDTYx)**

Offset[98] 0x68/0x69 Access: User read/write

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

Note:
98. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

## 5.14.4 Functional Description

### 5.14.4.1 PWM Clock Select

There are four available clocks: clock A, clock B, clock SA (scaled A), and clock SB (scaled B). These four clocks are based on the D2D clock.

Clock A and B can be software selected to be 1, 1/2, 1/4, 1/8,..., 1/64, 1/128 times the D2D clock. Clock SA uses clock A as an input and divides it further with a reloadable counter. Similarly, clock SB uses clock B as an input and divides it further with a reloadable counter. The rates available for clock SA are software selectable to be clock A divided by 2, 4, 6, 8,..., or 512 in increments of divide by 2. Similar rates are available for clock SB. Each PWM channel has the capability of selecting one of two clocks, either the pre-scaled clock (clock A or B) or the scaled clock (clock SA or SB).

The block diagram in Figure 25 shows the four different clocks and how the scaled clocks are created.

#### 5.14.4.1.1 Prescale

The input clock to the PWM prescaler is the D2D clock. The input clock can also be disabled when both PWM channels are disabled (PWME1-0 = 0). This is useful for reducing power by disabling the prescale counter.

Clock A and clock B are scaled values of the input clock. The value is software selectable for both clock A and clock B and has options of 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, or 1/128 times the D2D clock. The value selected for clock A is determined by the PCKA2, PCKA1, PCKA0 bits in the PWMPRCLK register. The value selected for clock B is determined by the PCKB2, PCKB1, PCKB0 bits also in the PWMPRCLK register.

#### 5.14.4.1.2 Clock Scale

The scaled A clock uses clock A as an input and divides it further with a user programmable value and then divides this by 2. The scaled B clock uses clock B as an input and divides it further with a user programmable value and then divides this by 2. The rates available for clock SA are software selectable to be clock A divided by 2, 4, 6, 8,..., or 512 in increments of divide by 2. Similar rates are available for clock SB.

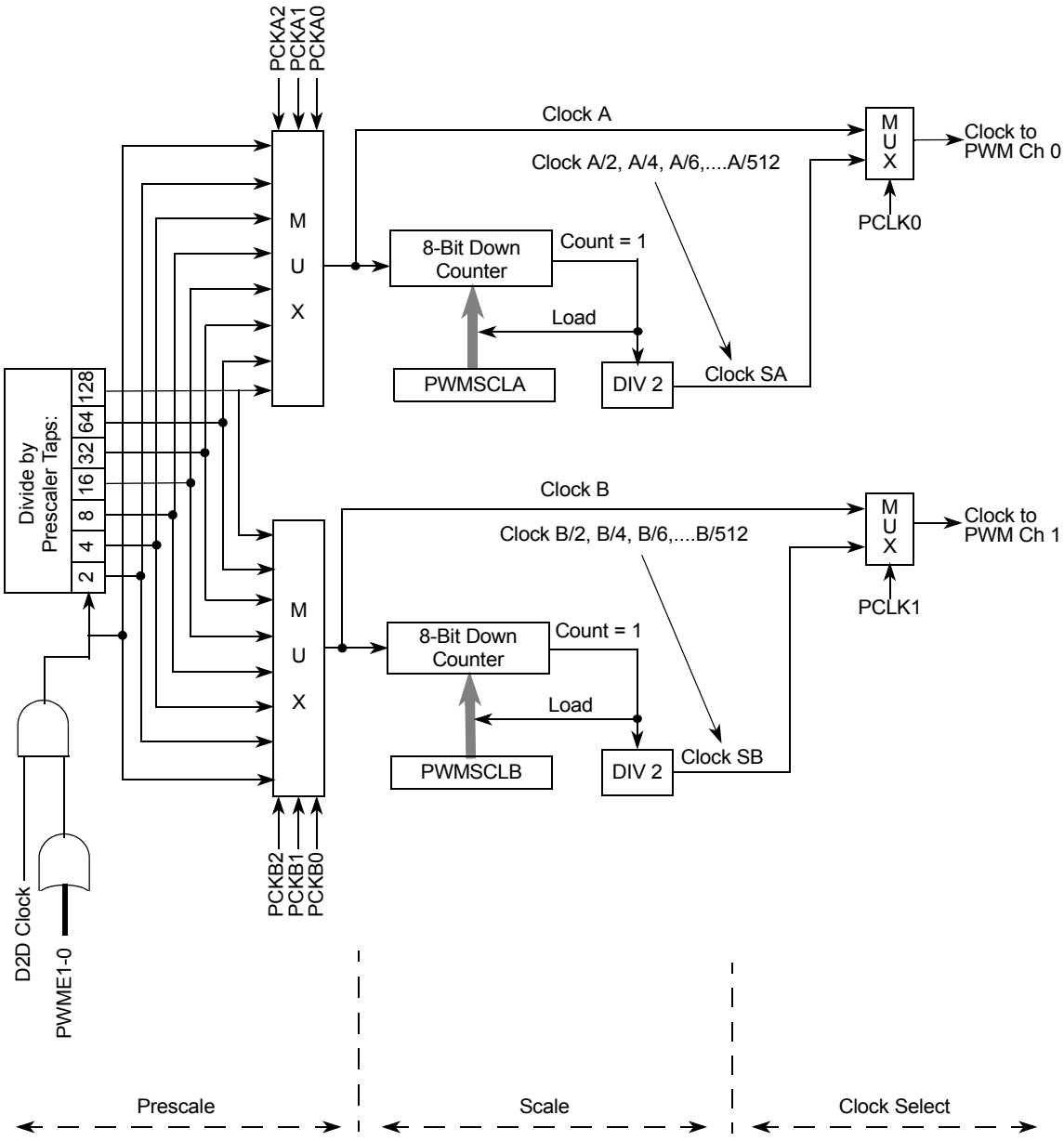



**Figure 25. PWM Clock Select Block Diagram**

Clock A is used as an input to an 8-bit down counter. This down counter loads a user programmable scale value from the scale register (PWMSCLA). When the down counter reaches one, a pulse is output and the 8-bit counter is re-loaded. The output signal from this circuit is further divided by two. This gives a greater range with only a slight reduction in granularity. Clock SA equals clock A divided by two times the value in the PWMSCLA register.

**NOTE**

Clock SA = Clock A / (2 * PWMSCLA)

When PWMSCLA = $00, PWMSCLA value is considered a full scale value of 256. Clock A is thus divided by 512.

Similarly, clock B is used as an input to an 8-bit down counter followed by a divide by two producing clock SB. Thus, clock SB equals clock B divided by two times the value in the PWMSCLB register.

**NOTE**

Clock SB = Clock B / (2 * PWMSCLB)

When PWMSCLB = $00, PWMSCLB value is considered a full scale value of 256. Clock B is thus divided by 512.

As an example, consider the case in which the user writes $FF into the PWMSCLA register. Clock A for this case is E divided by 4. A pulse occurs at a rate of once every 255x4 E cycles. Passing this through the divide by two circuit produces a clock signal at an E divided by 2040 rate. Similarly, a value of $01 in the PWMSCLA register when clock A is E divided by 4 produces a clock at an E divided by 8 rate.

Writing to PWMSCLA or PWMSCLB causes the associated 8-bit down counter to be re-loaded. Otherwise, when changing rates the counter would have to count down to $01 before counting at the proper rate. Forcing the associated counter to re-load the scale register value every time PWMSCLA or PWMSCLB is written prevents this.

**NOTE**

Writing to the scale registers while channels are operating can cause irregularities in the PWM outputs.

#### 5.14.4.1.3 Clock Select

Each PWM channel has the capability of selecting one of two clocks. For channels 0 the clock choice is clock A or clock SA. For channels 1 the choice is clock B or clock SB. The clock selection is done with the PCLKx control bits in the PWMCTL register.

**NOTE**

Changing clock control bits while channels are operating can cause irregularities in the PWM outputs.

### 5.14.4.2 PWM Channel Timers

The main part of the PWM module are the actual timers. Each of the timer channels has a counter, a period register, and a duty register (each are 8-bit). The waveform output period is controlled by a match between the period register and the value in the counter. The duty is controlled by a match between the duty register and the counter value, and causes the state of the output to change during the period. The starting polarity of the output is also selectable on a per channel basis. Shown in Figure 26 is the block diagram for the PWM timer.

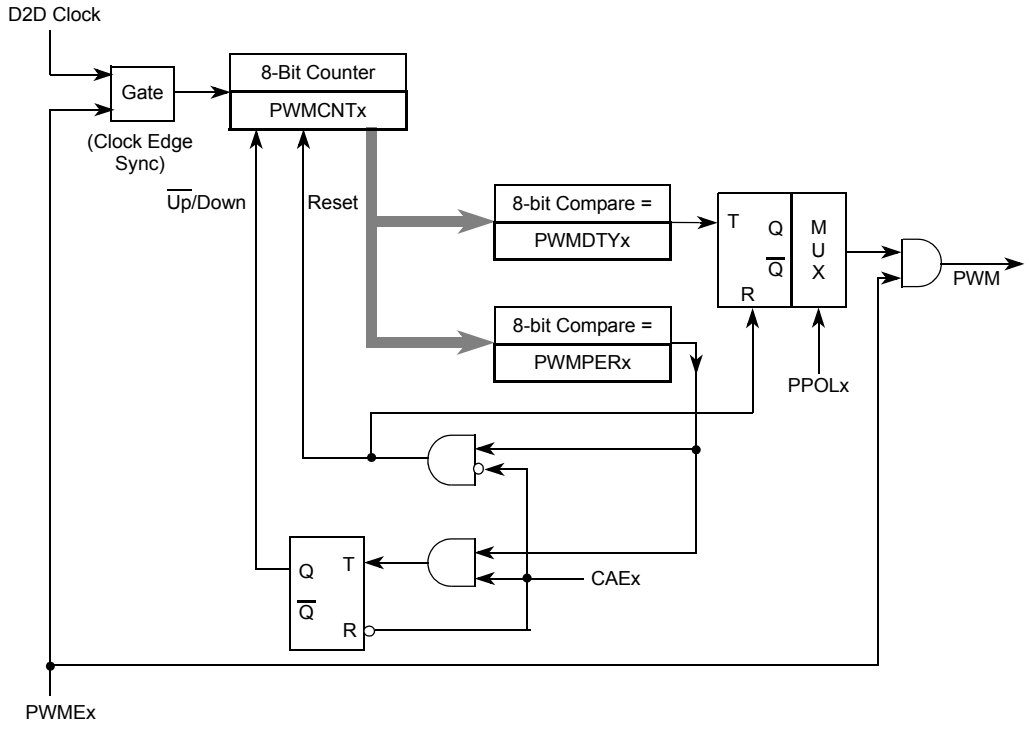



**Figure 26. PWM Timer Channel Block Diagram**

#### 5.14.4.2.1 PWM Enable

Each PWM channel has an enable bit (PWMEx) to start its waveform output. When any of the PWMEx bits are set (PWMEx = 1), the associated PWM output signal is enabled immediately. However, the actual PWM waveform is not available on the associated PWM output until its clock source begins its next cycle due to the synchronization of PWMEx and the clock source.

**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 the clock is only enabled or disabled at an edge. When the channel is disabled (PWMEx = 0), the counter for the channel does not count.

#### 5.14.4.2.2 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 zero, the output starts low and then goes high when the duty count is reached.

#### 5.14.4.2.3 PWM Period and Duty

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

- The effective period ends
- The counter is written (counter resets to $00)
- The channel is disabled

In this way, the output of the PWM is always 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 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, since 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 contains the count of either the high time or the low time.

#### 5.14.4.2.4 PWM Timer Counters

Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source (see Section 5.14.4.1, “PWM Clock Select" for the available clock sources and rates). The counter compares to two registers, a duty register and a period register as shown in Figure 26. When the PWM counter matches the duty register, the output flip-flop changes state, causing the PWM waveform to also change state. A match between the PWM counter and the period register behaves differently depending on what output mode is selected as shown in Figure 26 and described in Section 5.14.4.2.5, “Left Aligned Outputs" and Section 5.14.4.2.6, “Center Aligned Outputs".

Each channel counter can be read at anytime without affecting the count or the operation of the PWM channel.

Any value written to the counter causes the counter to reset to $00, the counter direction to be set to up, the immediate load of both duty and period registers with values from the buffers, and the output to change according to the polarity bit. When the channel is disabled (PWMEx = 0), the counter stops. When a channel becomes enabled (PWMEx = 1), the associated PWM counter continues from the count in the PWMCNTx register. This allows the waveform to continue where it left off when the channel is re-enabled. When the channel is disabled, writing “0” to the period register causes the counter to reset on the next selected clock.

**NOTE**

To start a new “clean” PWM waveform without any “history” from the old waveform, writing the channel counter (PWMCNTx) must happen prior to enabling the PWM channel (PWMEx = 1).

Generally, writes to the counter are done prior to enabling a channel in order to start from a known state. However, writing a counter can also be done while the PWM channel is enabled (counting). The effect is similar to writing the counter when the channel is disabled, except the new period is started immediately with the output set according to the polarity bit.

**NOTE**

Writing to the counter while the channel is enabled can cause an irregular PWM cycle to occur.

The counter is cleared at the end of the effective period (see Section 5.14.4.2.5, “Left Aligned Outputs" and Section 5.14.4.2.6, “Center Aligned Outputs" for more details).

**Table 129. PWM Timer Counter Conditions**

| Counter Clears ($00) | Counter Counts | Counter Stops |
|---|---|---|
| When PWMCNTx register written to any value | When PWM channel is enabled (PWMEx = 1). Counts from last value in PWMCNTx. | When PWM channel is disabled (PWMEx = 0) |
| Effective period ends | | |

#### 5.14.4.2.5 Left Aligned Outputs

The PWM timer provides the choice of two types of outputs, left aligned or center aligned. They are selected with the CAEx bits in the PWMCTL register. If the CAEx bit is cleared (CAEx = 0), the corresponding PWM output is left aligned.

In left aligned output mode, the 8-bit counter is configured as an up counter only. It compares to two registers, a duty register and a period register as shown in the block diagram in Figure 26. When the PWM counter matches the duty register the output flip-flop changes state causing the PWM waveform to also change state. A match between the PWM counter and the period register resets the counter and the output flip-flop, as shown in Figure 26, as well as performing a load from the double buffer period and duty register to the associated registers, as described in Section 5.14.4.2.3, “PWM Period and Duty". The counter counts from 0 to the value in the period register – 1.

**NOTE**

Changing the PWM output mode from left aligned to center aligned output (or vice versa) while channels are operating can cause irregularities in the PWM output. It is recommended to program the output mode before enabling the PWM channel.

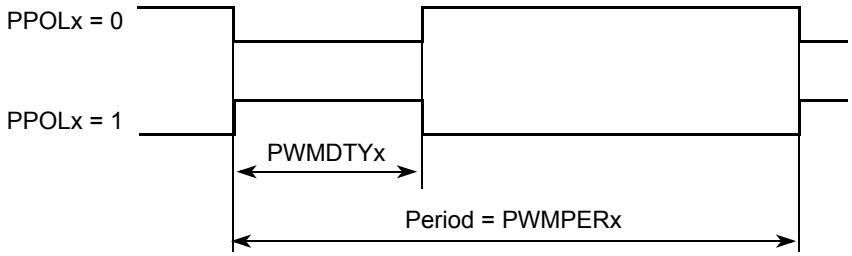



**Figure 27. PWM Left Aligned Output Waveform**

To calculate the output frequency in left aligned output mode for a particular channel, take the selected clock source frequency for the channel (A, B, SA, or SB), and divide it by the value in the period register for this channel.

- PWMx Frequency = Clock (A, B, SA, or SB) / PWMPERx
- PWMx Duty Cycle (high time as a% of period):
  - Polarity = 0 (PPOLx = 0)
- Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
  - Polarity = 1 (PPOLx = 1)

  Duty Cycle = [PWMDTYx / PWMPERx] * 100%

As an example of a left aligned output, consider the following case:

Clock Source = E, where E = 10 kHz (100 µs period)

PPOLx = 0

PWMPERx = 4

PWMDTYx = 1

PWMx Frequency = 10 kHz/4 = 2.5 kHz

PWMx Period = 400 µs

PWMx Duty Cycle = 3/4 *100% = 75%

The output waveform generated is shown in Figure 28.

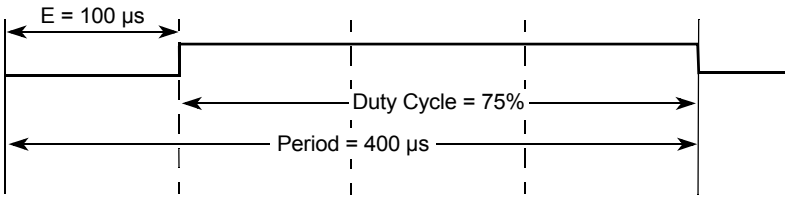



**Figure 28. PWM Left Aligned Output Example Waveform**

#### 5.14.4.2.6 Center Aligned Outputs

For a center aligned output mode selection, set the CAEx bit (CAEx = 1) in the PWMCTL register, and the corresponding PWM output is center aligned.

The 8-bit counter operates as an up/down counter in this mode, and is set to up whenever the counter is equal to $00. The counter compares to two registers, a duty register and a period register, as shown in the block diagram in Figure 26. When the PWM counter matches the duty register, the output flip-flop changes state, causing the PWM waveform to also change state. A match between the PWM counter and the period register changes the counter direction from an up-count to a down-count. When the PWM counter decrements and matches the duty register again, the output flip-flop changes state, causing the PWM output to also change state. When the PWM counter decrements and reaches zero, the counter direction changes from a down-count back to an up-count, and a load from the double buffer period and duty registers to the associated registers is performed, as described in Section 5.14.4.2.3, “PWM Period and Duty". The counter counts from 0 up to the value in the period register and then back down to 0. Thus the effective period is PWMPERx*2.

**NOTE**

Changing the PWM output mode from left aligned to center aligned output (or vice versa) while channels are operating can cause irregularities in the PWM output. It is recommended to program the output mode before enabling the PWM channel.

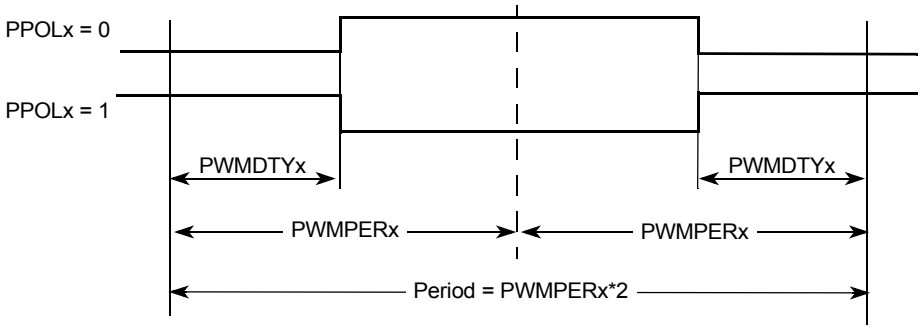



**Figure 29. PWM Center Aligned Output Waveform**

To calculate the output frequency in center aligned output mode for a particular channel, take the selected clock source frequency for the channel (A, B, SA, or SB) and divide it by twice the value in the period register for this channel.

- PWMx Frequency = Clock (A, B, SA, or SB) / (2*PWMPERx)
- PWMx Duty Cycle (high time as a% of period):
  - Polarity = 0 (PPOLx = 0)

  Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
  - Polarity = 1 (PPOLx = 1)

  Duty Cycle = [PWMDTYx / PWMPERx] * 100%

As an example of a center aligned output, consider the following case:

Clock Source = E, where E = 10 kHz (100 µs period)

PPOLx = 0

PWMPERx = 4

PWMDTYx = 1

PWMx Frequency = 10 kHz/8 = 1.25 kHz

PWMx Period = 800 µs

PWMx Duty Cycle = 3/4 *100% = 75%

Figure 30 shows the output waveform generated.

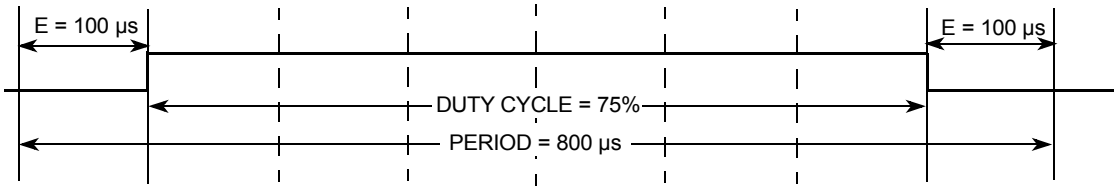



**Figure 30. PWM Center Aligned Output Example Waveform**

#### 5.14.4.2.7 PWM Boundary Cases

Table 130 summarizes the boundary conditions for the PWM, regardless of the output mode (left aligned or center aligned).

**Table 130. PWM Boundary Cases**

| PWMDTYx | PWMPERx | PPOLx | PWMx Output |
|---|---|---|---|
| $00 (indicates no duty) | >$00 | 1 | Always low |
| $00 (indicates no duty) | >$00 | 0 | Always high |
| XX | $00[99] (indicates no period) | 1 | Always high |
| XX | $00[99] (indicates no period) | 0 | Always low |
| >= PWMPERx | XX | 1 | Always high |

Note:
99. Counter = $00 and does not count.

### 5.14.5 Resets

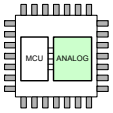
The reset state of each individual bit is listed within the Section 5.14.3, “Register Descriptions", which details the registers and their bit-fields. All special functions or modes which are initialized during or just following reset are described within this section.

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

### 5.14.6 Interrupts

The PWM module has no Interrupts.

## 5.15 LIN Physical Layer Interface - LIN




The LIN bus pin provides a physical layer for single-wire communication in automotive applications. The LIN physical layer is designed to meet the LIN physical layer version 2.1 specification, and has the following features:

- LIN physical layer 2.1 compliant
- Slew rate selection 20 kBit, 10 kBit, and fast Mode (100 kBit)
- Overtemperature Shutdown - HTI
- Permanent Pull-up in Normal mode 30 kΩ, 1.0 MΩ in low power
- Current limitation
- External Rx / Tx access. See Section 5.18, “General Purpose I/O - PTB[0…2]"
- Slew Rate Trim Bit. See Section 5.26, “MM912_634 - Analog Die Trimming"

The LIN driver is a Low-side MOSFET with current limitation and thermal shutdown. An internal pull-up resistor with a serial diode structure is integrated, so no external pull-up components are required for the application in a slave node. The fall time from dominant to recessive and the rise time from recessive to dominant is controlled. The symmetry between both slopes is guaranteed.

### 5.15.1 LIN Pin

The LIN pin offers high susceptibility immunity level from external disturbance, guaranteeing communication during external disturbance. See Section 4.8, “ESD Protection and Latch-up Immunity".

### 5.15.2 Slew Rate Selection

The slew rate can be selected for optimized operation at 10 kBit/s and 20 kBit/s as well as a fast baud rate (100 kBit) for test and programming. The slew rate can be adapted with the bits LINSR[1:0] in the LIN Register (LINR). The initial slew rate is 20 kBit/s.

### 5.15.3 Overtemperature Shutdown (LIN Interrupt)

The output Low-side FET (transmitter) is protected against overtemperature conditions. In case of an overtemperature condition, the transmitter is shut down and the bit LINOTC in the LIN Register (LINR) is set as long as the condition is present.

If the LINOTIE bit is set in the LIN Register (LINR), an Interrupt IRQ is generated. Acknowledge the interrupt by reading the LIN Register (LINR). To issue a new interrupt, the condition has to vanish and occur again.

The transmitter is automatically re-enabled once the overtemperature condition is gone and TxD is High.

### 5.15.4 Low Power Mode and Wake-up Feature

During Low Power mode operation the transmitter of the physical layer is disabled. The receiver is still active and able to detect Wake-up events on the LIN bus line.

A dominant level longer than $t_{PROPWL}$ followed by a rising edge, generates a wake-up event and be reported in the Wake-up Source Register (WSR).

### 5.15.5 J2602 Compliance

A Low Voltage Shutdown feature was implemented to allow controlled J2602 compliant LIN driver behavior under Low Voltage conditions (LVSD=0).

When an undervoltage occurs on VS1 (LVI), the LIN stays in recessive mode if it was in recessive state. If it was in a dominant state, it waits until the next dominant to recessive transition, then it stays in the recessive state.

When the undervoltage condition (LVI) is gone, the LIN starts operating when Tx is in a recessive state or on the next dominant to recessive transition.

### 5.15.6 Register Definition

#### 5.15.6.1 LIN Register (LINR)

**Table 131. LIN Register (LINR)**

Offset[(99)] 0x18 Access: User read

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | LINOTIE | LINOTC | RX | TX | LVSD | LINEN | LINSR | |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
100. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

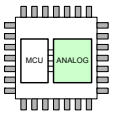
**Table 132. LINR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7 - LINOTIE | LIN - Overtemperature Interrupt Enable |
| 6 - LINOTC | LIN - Overtemperature condition present. LIN driver is shut down. Reading this bit clears the LINOT interrupt flag. |
| 5 - RX | LIN - Receiver (Rx) Status.<br>0 - LIN Bus Dominant<br>1 - LIN Bus Recessive |
| 4 - TX | LIN - Direct Transmitter Control. The inverted signal is OR<br>0 - Transmitter not controlled<br>1 - Transmitter Dominant |

**Table 132. LINR - Register Field Descriptions (continued)**

| Field | Description |
|---|---|
| 3 - LVSD | LIN - Low Voltage Shutdown Disable (J2602 Compliance Control)<br>0 - LIN is set to recessive state in case of VS1 undervoltage condition<br>1 - LIN stays functional even with a VS1 undervoltage condition |
| 2 - LINEN | LIN Module Enable<br>0 - LIN Module Disabled<br>1 - LIN Module Enabled |
| 1-0 - LINSR | LIN - Slew Rate Select<br>00 - Normal Slew Rate (20 kBit)<br>01 - Slow Slew Rate (10.4 kBit)<br>10 - Fast Slew Rate (100 kBit)<br>11 - Normal Slew Rate (20 kBit) |

## 5.16 Serial Communication Interface (S08SCIV4)




### 5.16.1 Introduction

#### 5.16.1.1 Features

Features of the SCI module include:

- Full-duplex, standard non-return-to-zero (NRZ) format
- Double-buffered transmitter and receiver with separate enables
- Programmable baud rates (13-bit modulo divider)
- Interrupt-driven or polled operation:
  - Transmit data register empty and transmission complete
  - Receive data register full
  - Receive overrun, parity error, framing error, and noise error
  - Idle receiver detect
  - Active edge on receive pin
  - Break detect supporting LIN
- Hardware parity generation and checking
- Programmable 8-bit or 9-bit character length
- Receiver wake-up by idle-line or address-mark
- Optional 13-bit break character generation / 11-bit break character detection
- Selectable transmitter output polarity

#### 5.16.1.2 Modes of Operation

See Section 5.16.3, “Functional Description", for details concerning SCI operation in these modes:

- 8 and 9-bit data modes
- Loop mode
- Single-wire mode

#### 5.16.1.3 Block Diagram

Figure 31 shows the transmitter portion of the SCI.

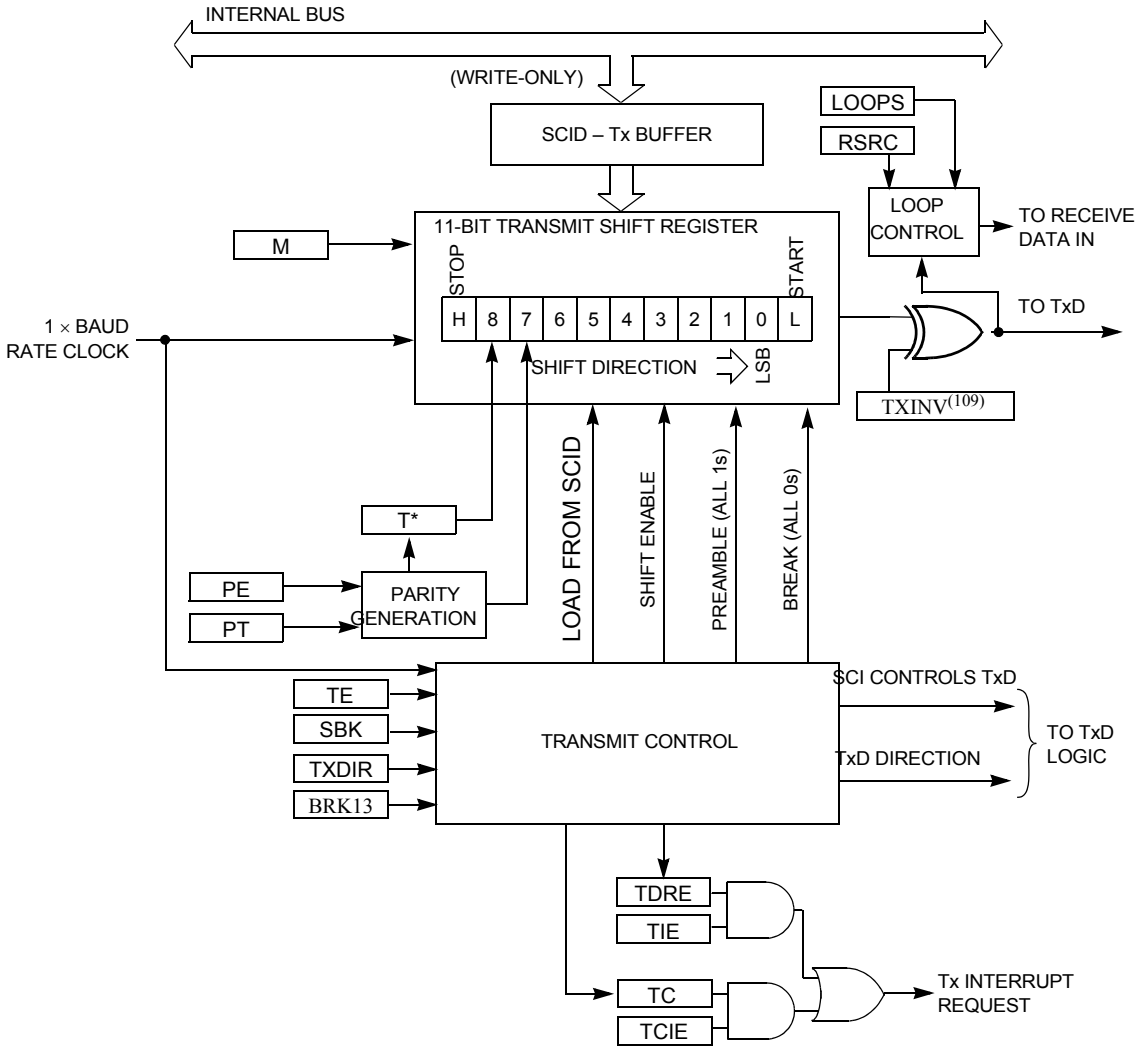



**Figure 31. SCI Transmitter Block Diagram**

Figure 32 shows the receiver portion of the SCI.

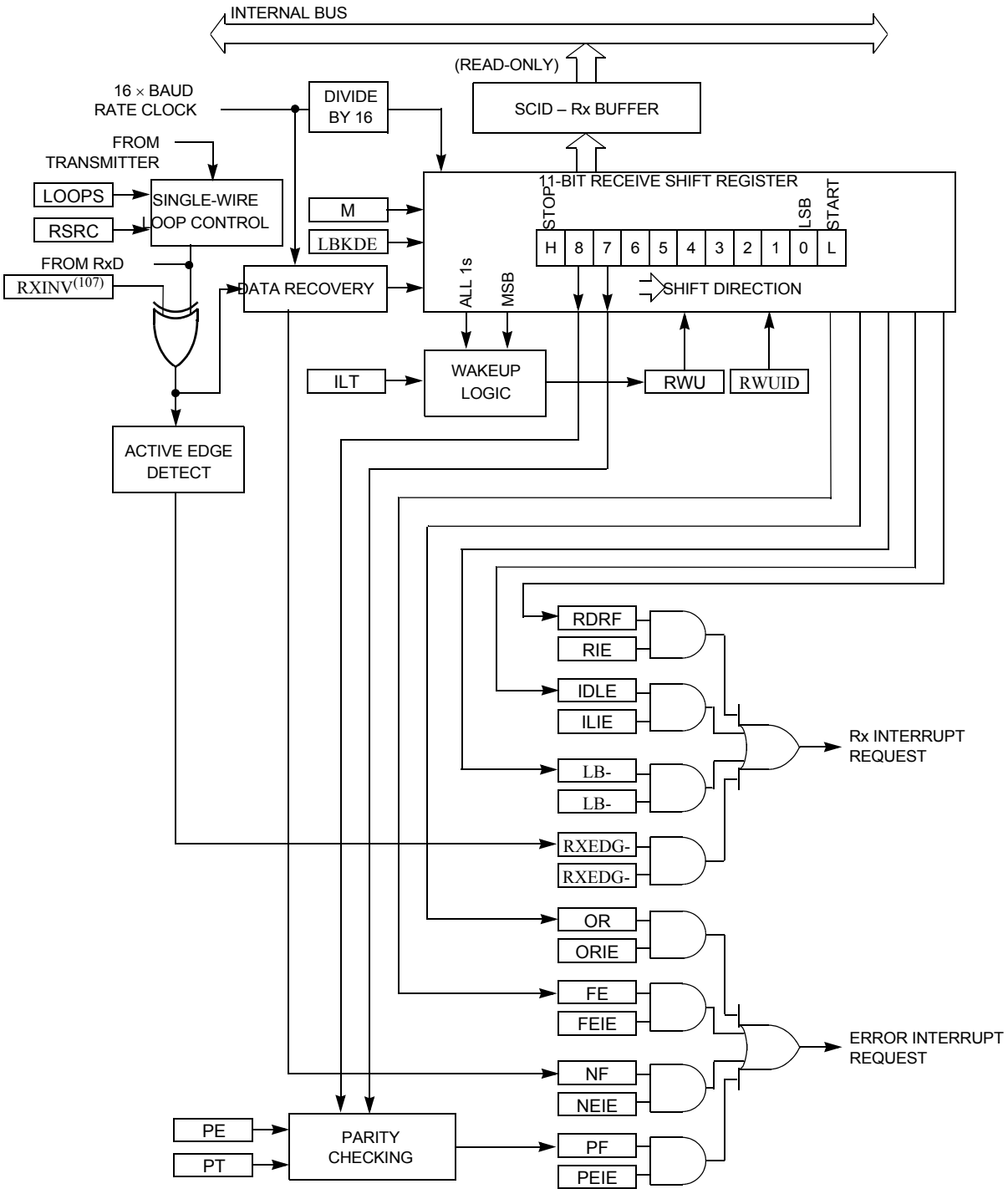



**Figure 32. SCI Receiver Block Diagram**

### 5.16.2 Register Definition

The SCI has eight 8-bit registers to control baud rate, select SCI options, report SCI status, and for transmit/receive data.

Refer to Section 5.6, “Die to Die Interface - Target" of this data sheet for the absolute address assignments for all SCI registers. This section refers to registers and control bits only by their names.

### 5.16.2.1 SCI Baud Rate Registers (SCIBD (hi), SCIBD (lo))

This pair of registers controls the prescale divisor for SCI baud rate generation. To update the 13-bit baud rate setting [SBR12:SBR0], first write to SCIBD (hi) to buffer the high half of the new value, and then write to SCIBD (lo). The working value in SCIBD (hi) does not change until SCIBD (lo) is written.

SCIBDL is reset to a non-zero value, so after reset the baud rate generator remains disabled until the first time the receiver or transmitter is enabled (RE or TE bits in SCIC2 are written to 1).

**Table 133. SCI Baud Rate Register (SCIBD (hi))**

Offset[99] 0x40 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | LBKDIE | RXEDGIE | 0 | SBR12 | SBR11 | SBR10 | SBR9 | SBR8 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
101. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 134. SCIBD (hi) Field Descriptions**

| Field | Description |
|---|---|
| 7<br>LBKDIE | LIN Break Detect Interrupt Enable (for LBKDIF)<br>0 Hardware interrupts from LBKDIF disabled (use polling).<br>1 Hardware interrupt requested when LBKDIF flag is 1. |
| 6<br>RXEDGIE | RxD Input Active Edge Interrupt Enable (for RXEDGIF)<br>0 Hardware interrupts from RXEDGIF disabled (use polling).<br>1 Hardware interrupt requested when RXEDGIF flag is 1. |
| 4:0<br>SBR[128] | Baud Rate Modulo Divisor — The 13 bits in SBR[12:0] are referred to collectively as BR, and they set the modulo divide rate for the SCI baud rate generator. When BR = 0, the SCI baud rate generator is disabled to reduce supply current. When BR = 1 to 8191, the SCI baud rate = BUSCLK/(16×BR). See also BR bits in Table 136. |

**Table 135. SCI Baud Rate Register (SCIBDL)**

Offset[102] 0x41 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | SBR7 | SBR6 | SBR5 | SBR4 | SBR3 | SBR2 | SBR1 | SBR0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

Note:
102. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 136. SCIBDL Field Descriptions**

| Field | Description |
|---|---|
| 7:0<br>SBR[7:0] | Baud Rate Modulo Divisor — These 13 bits in SBR[12:0] are referred to collectively as BR, and they set the modulo divide rate for the SCI baud rate generator. When BR = 0, the SCI baud rate generator is disabled to reduce supply current. When BR = 1 to 8191, the SCI baud rate = BUSCLK/(16×BR). See also BR bits in Table 134. |

### 5.16.2.2 SCI Control Register 1 (SCIC1)

This read/write register is used to control various optional features of the SCI system.

**Table 137. SCI Control Register 1 (SCIC1)**

Offset[103] 0x42 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | LOOPS | 0 | RSRC | M | 0 | ILT | PE | PT |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
103. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 138. SCIC1 Field Descriptions**

| Field | Description |
|---|---|
| 7<br>LOOPS | Loop Mode Select — Selects between loop back modes and normal 2-pin full-duplex modes. When LOOPS = 1, the transmitter output is internally connected to the receiver input.<br>0 Normal operation — RxD and TxD use separate pins.<br>1 Loop mode or single-wire mode where transmitter outputs are internally connected to receiver input. (See RSRC bit.) RxD pin is not used by SCI. |
| 5<br>RSRC | Receiver Source Select — This bit has no meaning or effect unless the LOOPS bit is set to 1. When LOOPS = 1, the receiver input is internally connected to the TxD pin and RSRC determines whether this connection is also connected to the transmitter output.<br>0 Provided LOOPS = 1, RSRC = 0 selects internal loop back mode and the SCI does not use the RxD pins.<br>1 Single-wire SCI mode where the TxD pin is connected to the transmitter output and receiver input. |
| 4<br>M | 9-Bit or 8-Bit Mode Select<br>0 Normal — start + 8 data bits (LSB first) + stop.<br>1 Receiver and transmitter use 9-bit data characters start + 8 data bits (LSB first) + 9th data bit + stop. |
| 2<br>ILT | Idle Line Type Select — Setting this bit to 1 ensures the stop bit and logic 1 bits at the end of a character do not count toward the 10 or 11 bit times of logic high level needed by the idle line detection logic. Refer to Section 5.16.3.3.2.1, “Idle-line Wake-up" for more information.<br>0 Idle character bit count starts after start bit.<br>1 Idle character bit count starts after stop bit. |
| 1<br>PE | Parity Enable — Enables hardware parity generation and checking. When parity is enabled, the most significant bit (MSB) of the data character (eighth or ninth data bit) is treated as the parity bit.<br>0 No hardware parity generation or checking.<br>1 Parity enabled. |
| 0<br>PT | Parity Type — Provided parity is enabled (PE = 1), this bit selects even or odd parity. Odd parity means the total number of 1s in the data character, including the parity bit, is odd. Even parity means the total number of 1s in the data character, including the parity bit, is even.<br>0 Even parity.<br>1 Odd parity. |

### 5.16.2.3 SCI Control Register 2 (SCIC2)

This register can be read or written at any time.

**Table 139. SCI Control Register 2 (SCIC2)**

Offset[104] 0x43 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | TIE | TCIE | RIE | ILIE | TE | RE | RWU | SBK |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
104. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 140. SCIC2 Field Descriptions**

| Field | Description |
|---|---|
| 7<br>TIE | Transmit Interrupt Enable (for TDRE)<br>0 Hardware interrupts from TDRE disabled (use polling).<br>1 Hardware interrupt requested when TDRE flag is 1. |
| 6<br>TCIE | Transmission Complete Interrupt Enable (for TC)<br>0 Hardware interrupts from TC disabled (use polling).<br>1 Hardware interrupt requested when TC flag is 1. |
| 5<br>RIE | Receiver Interrupt Enable (for RDRF)<br>0 Hardware interrupts from RDRF disabled (use polling).<br>1 Hardware interrupt requested when RDRF flag is 1. |
| 4<br>LIE | Idle Line Interrupt Enable (for IDLE)<br>0 Hardware interrupts from IDLE disabled (use polling).<br>1 Hardware interrupt requested when IDLE flag is 1. |
| 3<br>TE | Transmitter Enable<br>0 Transmitter off.<br>1 Transmitter on.<br>TE must be 1 in order to use the SCI transmitter. When TE = 1, the SCI forces the TxD pin to act as an output for the SCI system.<br>When the SCI is configured for single-wire operation (LOOPS = RSRC = 1), TXDIR controls the direction of traffic on the single SCI communication line (TxD pin).<br>TE also can be used to queue an idle character by writing TE = 0 then TE = 1 while a transmission is in progress. Refer to Section 5.16.3.2.1, “Send Break and Queued Idle" for more details.<br>When TE is written to 0, the transmitter keeps control of the port TxD pin until any data, queued idle, or queued break character finishes transmitting before allowing the pin to revert to a general-purpose I/O pin. |
| 2<br>RE | Receiver Enable — When the SCI receiver is off, the RxD pin reverts to being a general-purpose port I/O pin. If LOOPS = 1 the RxD pin reverts to being a general-purpose I/O pin even if RE = 1.<br>0 Receiver off.<br>1 Receiver on. |
| 1<br>RWU | Receiver Wake-up Control — This bit can be written to 1 to place the SCI receiver in a standby state where it waits for automatic hardware detection of a selected wake-up condition. The wake-up condition is either an idle line between messages (WAKE = 0, idle-line wake-up), or a logic 1 in the most significant data bit in a character (WAKE = 1, address-mark wake-up). Application software sets RWU and (normally) a selected hardware condition automatically clears RWU. Refer to Section 5.16.3.3.2, “Receiver Wake-up Operation" for more details.<br>0 Normal SCI receiver operation.<br>1 SCI receiver in standby waiting for wake-up condition. |
| 0<br>SBK | Send Break — Writing a 1 and then a 0 to SBK queues a break character in the transmit data stream. Additional break characters of 10 or 11 (13 or 14 if BRK13 = 1) bit times of logic 0 are queued as long as SBK = 1. Depending on the timing of the set and clear of SBK relative to the information currently being transmitted, a second break character may be queued before software clears SBK. Refer to Section 5.16.3.2.1, “Send Break and Queued Idle" for more details.<br>0 Normal transmitter operation.<br>1 Queue break character(s) to be sent. |

### 5.16.2.4 SCI Status Register 1 (SCIS1)

This register has eight read-only status flags. Writes have no effect. Special software sequences (which do not involve writing to this register) are used to clear these status flags.

**Table 141. SCI Status Register 1 (SCIS1)**

Offset[105] 0x44 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | TDRE | TC | RDRF | IDLE | OR | NF | FE | pF |
| W | | | | | | | | |
| Reset | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
105. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 142. SCIS1 Field Descriptions**

| Field | Description |
|---|---|
| 7<br>TDRE | Transmit Data Register Empty Flag — TDRE is set out of reset and when a transmit data value transfers from the transmit data buffer to the transmit shifter, leaving room for a new character in the buffer. To clear TDRE, read SCIS1 with TDRE = 1 and then write to the SCI data register (SCID).<br>0 Transmit data register (buffer) full.<br>1 Transmit data register (buffer) empty. |
| 6<br>TC | Transmission Complete Flag — TC is set out of reset and when TDRE = 1 and no data, preamble, or break character is being transmitted.<br>0 Transmitter active (sending data, a preamble, or a break).<br>1 Transmitter idle (transmission activity complete).<br>TC is cleared automatically by reading SCIS1 with TC = 1 and then doing one of the following three things:<br>• Write to the SCI data register (SCID) to transmit new data<br>• Queue a preamble by changing TE from 0 to 1<br>• Queue a break character by writing 1 to SBK in SCIC2 |
| 5<br>RDRF | Receive Data Register Full Flag — RDRF becomes set when a character transfers from the receive shifter into the receive data register (SCID). To clear RDRF, read SCIS1 with RDRF = 1 and then read the SCI data register (SCID).<br>0 Receive data register empty.<br>1 Receive data register full. |
| 4<br>IDLE | Idle Line Flag — IDLE is set when the SCI receive line becomes idle for a full character time after a period of activity. When ILT = 0, the receiver starts counting idle bit times after the start bit. So if the receive character is all 1s, these bit times and the stop bit time count toward the full character time of logic high (10 or 11 bit times depending on the M control bit) needed for the receiver to detect an idle line. When ILT = 1, the receiver doesn't start counting idle bit times until after the stop bit. So the stop bit and any logic high bit times at the end of the previous character do not count toward the full character time of logic high needed for the receiver to detect an idle line.<br>To clear IDLE, read SCIS1 with IDLE = 1 and then read the SCI data register (SCID). After IDLE has been cleared, it cannot become set again until after a new character has been received and RDRF has been set. IDLE gets set only once even if the receive line remains idle for an extended period.<br>0 No idle line detected.<br>1 Idle line was detected. |
| 3<br>OR | Receiver Overrun Flag — OR is set when a new serial character is ready to be transferred to the receive data register (buffer), but the previously received character has not been read from SCID yet. In this case, the new character (and all associated error information) is lost because there is no room to move it into SCID. To clear OR, read SCIS1 with OR = 1 and then read the SCI data register (SCID).<br>0 No overrun.<br>1 Receive overrun (new SCI data lost). |
| 2<br>NF | Noise Flag — The advanced sampling technique used in the receiver takes seven samples during the start bit and three samples in each data bit and the stop bit. If any of these samples disagrees with the rest of the samples within any bit time in the frame, the flag NF is set at the same time as the flag RDRF gets set for the character. To clear NF, read SCIS1 and then read the SCI data register (SCID).<br>0 No noise detected.<br>1 Noise detected in the received character in SCID. |

**Table 142. SCIS1 Field Descriptions (continued)**

| Field | Description |
|---|---|
| 1<br>FE | Framing Error Flag — FE is set at the same time as RDRF when the receiver detects a logic 0 where the stop bit was expected. This suggests the receiver was not properly aligned to a character frame. To clear FE, read SCIS1 with FE = 1 and then read the SCI data register (SCID).<br>0 No framing error detected. This does not guarantee the framing is correct.<br>1 Framing error. |
| 0<br>PF | Parity Error Flag — PF is set at the same time as RDRF when parity is enabled (PE = 1) and the parity bit in the received character does not agree with the expected parity value. To clear PF, read SCIS1 and then read the SCI data register (SCID).<br>0 No parity error.<br>1 Parity error. |

### 5.16.2.5 SCI Status Register 2 (SCIS2)

This register has one read-only status flag.

**Table 143. SCI Status Register 2 (SCIS2)**

Offset(106) 0x45 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | LBKDIF | RXEDGIF | 0 | RXINV(107) | RWUID | BRK13 | LBKDE | RAF |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
106. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 144. SCIS2 Field Descriptions**

| Field | Description |
|---|---|
| 7<br>LBKDIF | LIN Break Detect Interrupt Flag — LBKDIF is set when the LIN break detect circuitry is enabled and a LIN break character is detected. LBKDIF is cleared by writing a “1” to it.<br>0 No LIN break character has been detected.<br>1 LIN break character has been detected. |
| 6<br>RXEDGIF | RxD Pin Active Edge Interrupt Flag — RXEDGIF is set when an active edge (falling if RXINV = 0, rising if RXINV=1) on the RxD pin occurs. RXEDGIF is cleared by writing a “1” to it.<br>0 No active edge on the receive pin has occurred.<br>1 An active edge on the receive pin has occurred. |
| 4<br>RXINV(107) | Receive Data Inversion — Setting this bit reverses the polarity of the received data input.<br>0 Receive data not inverted<br>1 Receive data inverted |
| 3<br>RWUID | Receive Wake Up Idle Detect— RWUID controls whether the idle character waking up the receiver sets the IDLE bit.<br>0 During receive standby state (RWU = 1), the IDLE bit does not get set upon detection of an idle character.<br>1 During receive standby state (RWU = 1), the IDLE bit gets set upon detection of an idle character. |
| 2<br>BRK13 | Break Character Generation Length — BRK13 is used to select a longer transmitted break character length. Detection of a framing error is not affected by the state of this bit.<br>0 Break character is transmitted with length of 10 bit times (11 if M = 1)<br>1 Break character is transmitted with length of 13 bit times (14 if M = 1) |
| 1<br>LBKDE | LIN Break Detection Enable— LBKDE is used to select a longer break character detection length. While LBKDE is set, framing error (FE) and receive data register full (RDRF) flags are prevented from setting.<br>0 Break character detection disabled.<br>1 Break character detection enabled. |

**Table 144. SCIS2 Field Descriptions (continued)**

| Field | Description |
|---|---|
| 0<br>RAF | Receiver Active Flag—RAF is set when the receiver detects a logic 0 during the RT1 Time period of the start bit search. RAF is cleared when the receiver detects an idle character.<br>0 No reception in progress.<br>1 Reception in progress. |

Note:
107. Setting RXINV inverts the RxD input for all cases: data bits, start and stop bits, break, and idle.

When using an internal oscillator in a LIN system, it is necessary to raise the break detection threshold by one bit time. Under the worst case timing conditions allowed in LIN, it is possible a 0x00 data character can appear to be 10.26 bit times long at a slave which is running 14% faster than the master. This would trigger normal break detection circuitry which is designed to detect a 10 bit break symbol. When the LBKDE bit is set, framing errors are inhibited and the break detection threshold changes from 10 bits to 11 bits, preventing false detection of a 0x00 data character as a LIN break symbol.

### 5.16.2.6 SCI Control Register 3 (SCIC3)

**Table 145. SCI Control Register 3 (SCIC3)**

Offset[108] 0x46 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | R8 | T8 | TXDIR | TXINV[109] | ORIE | NEIE | FEIE | PEIE |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
108. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 146. SCIC3 Field Descriptions**

| Field | Description |
|---|---|
| 7<br>R8 | Ninth Data Bit for Receiver — When the SCI is configured for 9-bit data (M = 1), R8 can be thought of as a ninth receive data bit to the left of the MSB of the buffered data in the SCID register. When reading 9-bit data, read R8 before reading SCID because reading SCID completes automatic flag clearing sequences which could allow R8 and SCID to be overwritten with new data. |
| 6<br>T8 | Ninth Data Bit for Transmitter — When the SCI is configured for 9-bit data (M = 1), T8 may be thought of as a ninth transmit data bit to the left of the MSB of the data in the SCID register. When writing 9-bit data, the entire 9-bit value is transferred to the SCI shift register after SCID is written so T8 should be written (if it needs to change from its previous value) before SCID is written. If T8 does not need to change in the new value (such as when it is used to generate mark or space parity), it need not be written each time SCID is written. |
| 5<br>TXDIR | TxD Pin Direction in Single-wire Mode — When the SCI is configured for single-wire half-duplex operation (LOOPS = RSRC = 1), this bit determines the direction of data at the TxD pin.<br>0 TxD pin is an input in single-wire mode.<br>1 TxD pin is an output in single-wire mode. |
| 4<br>TXINV[109] | Transmit Data Inversion — Setting this bit reverses the polarity of the transmitted data output.<br>0 Transmit data not inverted<br>1 Transmit data inverted |
| 3<br>ORIE | Overrun Interrupt Enable — This bit enables the overrun flag (OR) to generate hardware interrupt requests.<br>0 OR interrupts disabled (use polling).<br>1 Hardware interrupt requested when OR = 1. |
| 2<br>NEIE | Noise Error Interrupt Enable — This bit enables the noise flag (NF) to generate hardware interrupt requests.<br>0 NF interrupts disabled (use polling).<br>1 Hardware interrupt requested when NF = 1. |
| 1<br>FEIE | Framing Error Interrupt Enable — This bit enables the framing error flag (FE) to generate hardware interrupt requests.<br>0 FE interrupts disabled (use polling).<br>1 Hardware interrupt requested when FE = 1. |

**Table 146. SCIC3 Field Descriptions (continued)**

| Field | Description |
|---|---|
| 0<br>PEIE | Parity Error Interrupt Enable — This bit enables the parity error flag (PF) to generate hardware interrupt requests.<br>0 PF interrupts disabled (use polling).<br>1 Hardware interrupt requested when PF = 1. |

Note:
109. Setting TXINV inverts the TxD output for all cases: data bits, start and stop bits, break, and idle.

### 5.16.2.7 SCI Data Register (SCID)

This register is actually two separate registers. Reads return the contents of the read-only receive data buffer and writes go to the write-only transmit data buffer. Reads and writes of this register are also involved in the automatic flag clearing mechanisms for the SCI status flags.

**Table 147. SCI Data Register (SCID)**

Offset[110] 0x47 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 |
| W | T7 | T6 | T5 | T4 | T3 | T2 | T1 | T0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
110. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

## 5.16.3 Functional Description

The SCI allows full-duplex, asynchronous, NRZ serial communication among the MCU and remote devices, including other MCUs. The SCI comprises a baud rate generator, transmitter, and receiver block. The transmitter and receiver operate independently, although they use the same baud rate generator. During normal operation, the MCU monitors the status of the SCI, writes the data to be transmitted, and processes received data. The following describes each of the blocks of the SCI.

### 5.16.3.1 Baud Rate Generation

As shown in Figure 33, the clock source for the SCI baud rate generator is the D2D clock.

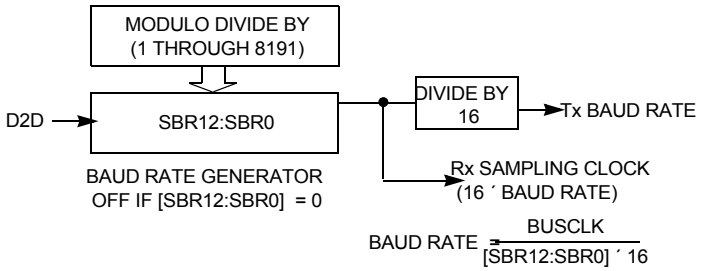



**Figure 33. SCI Baud Rate Generation**

SCI communications require the transmitter and receiver (which typically derive baud rates from independent clock sources) to use the same baud rate. Allowed tolerance on this baud frequency depends on the details of how the receiver synchronizes to the leading edge of the start bit and how bit sampling is performed.

The MCU resynchronizes to bit boundaries on every high-to-low transition, but in the worst case, there are no such transitions in the full 10- or 11-bit time character frame so any mismatch in baud rate is accumulated for the whole character time. For a Freescale Semiconductor SCI system whose bus frequency is driven by a crystal, the allowed baud rate mismatch is about ±4.5 percent for 8-bit data format and about ±4.0 percent for 9-bit data format. Although baud rate modulo divider settings do not always produce baud rates which exactly match standard rates, it is normally possible to get within a few percent, which is acceptable for reliable communications.

### 5.16.3.2 Transmitter Functional Description

This section describes the overall block diagram for the SCI transmitter, as well as specialized functions for sending break and idle characters. The transmitter block diagram is shown in Figure 31.

The transmitter output (TxD) idle state defaults to logic high (TXINV = 0 following reset). The transmitter output is inverted by setting TXINV = 1. The transmitter is enabled by setting the TE bit in SCIC2. This queues a preamble character which is one full character frame of the idle state. The transmitter then remains idle until data is available in the transmit data buffer. Programs store data into the transmit data buffer by writing to the SCI data register (SCID).

The central element of the SCI transmitter is the transmit shift register which is either 10 or 11 bits long depending on the setting in the M control bit. For the remainder of this section, it is assumed M = 0, selecting the normal 8-bit data mode. In 8-bit data mode, the shift register holds a start bit, eight data bits, and a stop bit. When the transmit shift register is available for a new SCI character, the value waiting in the transmit data register is transferred to the shift register (synchronized with the baud rate clock) and the transmit data register empty (TDRE) status flag is set to indicate another character may be written to the transmit data buffer at SCID.

If no new character is waiting in the transmit data buffer after a stop bit is shifted out the TxD pin, the transmitter sets the transmit complete flag and enters an idle mode, with TxD high, waiting for more characters to transmit.

Writing 0 to TE does not immediately release the pin to be a general-purpose I/O pin. Any transmit activity in progress must first be completed. This includes data characters in progress, queued idle characters, and queued break characters.

#### 5.16.3.2.1 Send Break and Queued Idle

The SBK control bit in SCIC2 is used to send break characters which were originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0 (10 bit times including the start and stop bits). A longer break of 13 bit times can be enabled by setting BRK13 = 1. Normally, a program would wait for TDRE to become set to indicate the last character of a message has moved to the transmit shifter, then write 1 and then write 0 to the SBK bit. This action queues a break character to be sent as soon as the shifter is available. If SBK is still 1 when the queued break moves into the shifter (synchronized to the baud rate clock), an additional break character is queued. If the receiving device is another Freescale Semiconductor SCI, the break characters are received as 0s in all eight data bits and a framing error (FE = 1) occurs.

When idle-line wake-up is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program would wait for TDRE to become set to indicate the last character of a message has moved to the transmit shifter, then write 0 and then write 1 to the TE bit. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish while TE = 0, the SCI transmitter never actually releases control of the TxD pin. If there is a possibility of the shifter finishing while TE = 0, set the general-purpose I/O controls so the pin which is shared with TxD is an output driving a logic 1. This ensures the TxD line looks like a normal idle line even if the SCI loses control of the port pin between writing 0 and then 1 to TE.

The length of the break character is affected by the BRK13 and M bits as shown below.

**Table 148. Break Character Length**

| BRK13 | M | Break Character Length |
|---|---|---|
| 0 | 0 | 10 bit times |
| 0 | 1 | 11 bit times |
| 1 | 0 | 13 bit times |
| 1 | 1 | 14 bit times |

### 5.16.3.3 Receiver Functional Description

In this section, the receiver block diagram (Figure 32) is used as a guide for the overall receiver functional description. Next, the data sampling technique used to reconstruct receiver data is described in more detail. Finally, two variations of the receiver wake-up function are explained.

The receiver input is inverted by setting RXINV = 1. The receiver is enabled by setting the RE bit in SCIC2. Character frames consist of a start bit of logic 0, eight (or nine) data bits (LSB first), and a stop bit of logic 1. For information about 9-bit data mode, refer to Section 5.16.1.2, “Modes of Operation". For the remainder of this discussion, we assume the SCI is configured for normal 8-bit data mode.

After receiving the stop bit into the receive shifter, and provided the receive data register is not already full, the data character is transferred to the receive data register and the receive data register full (RDRF) status flag is set. If RDRF was already set indicating the receive data register (buffer) was already full, the overrun (OR) status flag is set and the new data is lost. Because the SCI receiver is double-buffered, the program has one full character time after RDRF is set before the data in the receive data buffer must be read to avoid a receiver overrun.

When a program detects the receive data register is full (RDRF = 1), it gets the data from the receive data register by reading SCID. The RDRF flag is cleared automatically by a 2-step sequence which is normally satisfied in the course of the user's program which handles receive data. Refer to Section 5.16.3.4, "Interrupts and Status Flags" for more details about flag clearing.

#### 5.16.3.3.1 Data Sampling Technique

The SCI receiver uses a 16× baud rate clock for sampling. The receiver starts by taking logic level samples at 16 times the baud rate to search for a falling edge on the RxD serial data input pin. A falling edge is defined as a logic 0 sample after three consecutive logic 1 samples. The 16× baud rate clock is used to divide the bit time into 16 segments labeled RT1 through RT16. When a falling edge is located, three more samples are taken at RT3, RT5, and RT7 to make sure this was a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes it is synchronized to a receive character.

The receiver then samples each bit time, including the start and stop bits, at RT8, RT9, and RT10 to determine the logic level for this bit. The logic level is interpreted to be of the majority of the samples taken during the bit time. In the case of the start bit, the bit is assumed to be 0 if at least two of the samples at RT3, RT5, and RT7 are 0 even if one or all of the samples taken at RT8, RT9, and RT10 are 1s. If any sample in any bit time (including the start and stop bits) in a character frame fails to agree with the logic level for this bit, the noise flag (NF) is set when the received character is transferred to the receive data buffer.

The falling edge detection logic continuously looks for falling edges, and if an edge is detected, the sample clock is resynchronized to bit times. This improves the reliability of the receiver in the presence of noise or mismatched baud rates. It does not improve worst case analysis because some characters do not have any extra falling edges anywhere in the character frame.

In the case of a framing error, provided the received character was not a break character, the sampling logic which searches for a falling edge is filled with three logic 1 samples so a new start bit can be detected almost immediately.

In the case of a framing error, the receiver is inhibited from receiving any new characters until the framing error flag is cleared. The receive shift register continues to function, but a complete character cannot transfer to the receive data buffer if FE is still set.

#### 5.16.3.3.2 Receiver Wake-up Operation

Receiver wake-up is a hardware mechanism allowing an SCI receiver to ignore the characters in a message is intended for a different SCI receiver. In such a system, all receivers evaluate the first character(s) of each message, and as soon as they determine the message is intended for a different receiver, they write logic 1 to the receiver wake up (RWU) control bit in SCIC2. When RWU bit is set, the status flags associated with the receiver (with the exception of the idle bit, IDLE, when RWUID bit is set) are inhibited from setting, thus eliminating the software overhead for handling the unimportant message characters. At the end of a message, or at the beginning of the next message, all receivers automatically force RWU to 0 so all receivers wake up in time to look at the first character(s) of the next message.

##### 5.16.3.3.2.1 Idle-line Wake-up

When WAKE = 0, the receiver is configured for idle-line wake-up. In this mode, RWU is cleared automatically when the receiver detects a full character time of the idle-line level. The M control bit selects 8-bit or 9-bit data mode which determines how many bit times of idle are needed to constitute a full character time (10 or 11 bit times because of the start and stop bits).

When RWU is one and RWUID is zero, the idle condition waking up the receiver does not set the IDLE flag. The receiver wakes up and waits for the first data character of the next message which sets the RDRF flag and generate an interrupt if enabled. When RWUID is one, any idle condition sets the IDLE flag and generates an interrupt if enabled, regardless of whether RWU is zero or one.

The idle-line type (ILT) control bit selects one of two ways to detect an idle line. When ILT = 0, the idle bit counter starts after the start bit so the stop bit and any logic 1s at the end of a character count toward the full character time of idle. When ILT = 1, the idle bit counter does not start until after a stop bit time, so the idle detection is not affected by the data in the last character of the previous message.

##### 5.16.3.3.2.2 Address-Mark Wake-up

When WAKE = 1, the receiver is configured for address-mark wake-up. In this mode, RWU is cleared automatically when the receiver detects a logic 1 in the most significant bit of a received character (eighth bit in M = 0 mode and ninth bit in M = 1 mode).

Address-mark wake-up allows messages to contain idle characters but requires the MSB be reserved for use in address frames. The logic 1 MSB of an address frame clears the RWU bit before the stop bit is received and sets the RDRF flag. In this case the character with the MSB set is received even though the receiver was sleeping during most of this character time.

### 5.16.3.4 Interrupts and Status Flags

The SCI system has three separate interrupt vectors to reduce the amount of software needed to isolate the cause of the interrupt. One interrupt vector is associated with the transmitter for TDRE and TC events. Another interrupt vector is associated with the receiver for RDRF, IDLE, RXEDGIF and LBKDIF events, and a third vector is used for OR, NF, FE, and PF error conditions. Each of these ten interrupt sources can be separately masked by local interrupt enable masks. The flags can still be polled by software when the local masks are cleared to disable generation of hardware interrupt requests.

The SCI transmitter has two status flags which optionally can generate hardware interrupt requests. Transmit data register empty (TDRE) indicates when there is room in the transmit data buffer to write another transmit character to SCID. If the transmit interrupt enable (TIE) bit is set, a hardware interrupt is requested whenever TDRE = 1. Transmit complete (TC) indicates the transmitter is finished transmitting all data, preamble, and break characters and is idle with TxD at the inactive level. This flag is often used in systems with modems to determine when it is safe to turn off the modem. If the transmit complete interrupt enable (TCIE) bit is set, a hardware interrupt is requested whenever TC = 1. Instead of hardware interrupts, software polling may be used to monitor the TDRE and TC status flags if the corresponding TIE or TCIE local interrupt masks are 0s.

When a program detects the receive data register is full (RDRF = 1), it gets the data from the receive data register by reading SCID. The RDRF flag is cleared by reading SCIS1 while RDRF = 1 and then reading SCID.

When polling is used, this sequence is naturally satisfied in the normal course of the user program. If hardware interrupts are used, SCIS1 must be read in the interrupt service routine (ISR). Normally, this is done in the ISR anyway to check for receive errors, so the sequence is automatically satisfied.

The IDLE status flag includes logic which prevents it from getting set repeatedly when the RxD line remains idle for an extended period of time. IDLE is cleared by reading SCIS1 while IDLE = 1 and then reading SCID. After IDLE has been cleared, it cannot become set again until the receiver has received at least one new character and has set RDRF.

If the associated error was detected in the received character which caused RDRF to be set, the error flags — noise flag (NF), framing error (FE), and parity error flag (PF) — get set at the same time as RDRF. These flags are not set in overrun cases.

If RDRF was already set when a new character is ready to be transferred from the receive shifter to the receive data buffer, the overrun (OR) flag gets set instead the data along with any associated NF, FE, or PF condition is lost.

At any time, an active edge on the RxD serial data input pin causes the RXEDGIF flag to set. The RXEDGIF flag is cleared by writing a “1” to it. This function does depend on the receiver being enabled (RE = 1).

### 5.16.3.5 Additional SCI Functions

The following sections describe additional SCI functions.

#### 5.16.3.5.1 8- and 9-Bit Data Modes

The SCI system (transmitter and receiver) can be configured to operate in 9-bit data mode by setting the M control bit in SCIC1. In 9-bit mode, there is a ninth data bit to the left of the MSB of the SCI data register. For the transmit data buffer, this bit is stored in T8 in SCIC3. For the receiver, the ninth bit is held in R8 in SCIC3.

For coherent writes to the transmit data buffer, write to the T8 bit before writing to SCID.

If the bit value to be transmitted as the ninth bit of a new character is the same as for the previous character, it is not necessary to write to T8 again. When data is transferred from the transmit data buffer to the transmit shifter, the value in T8 is copied at the same time data is transferred from SCID to the shifter.

9-bit data mode typically is used in conjunction with parity to allow eight bits of data plus the parity in the ninth bit. Or it is used with address-mark wake-up so the ninth data bit can serve as the wake-up bit. In custom protocols, the ninth bit can also serve as a software-controlled marker.

#### 5.16.3.5.2 Stop Mode Operation

During all stop modes, clocks to the SCI module are halted. In stop1 and stop2 modes, all SCI register data is lost and must be re-initialized upon recovery from these two stop modes. No SCI module registers are affected in stop3 mode.

The receive input active edge detect circuit is still active in stop3 mode, but not in stop2. An active edge on the receive input brings the CPU out of stop3 mode if the interrupt is not masked (RXEDGIE = 1).

Note that because the clocks are halted, the SCI module resumes operation upon exit from stop (only in stop3 mode). Software should ensure stop mode is not entered while there is a character being transmitted out of or received into the SCI module.

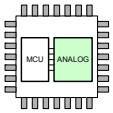
#### 5.16.3.5.3 Loop Mode

When LOOPS = 1, the RSRC bit in the same register chooses between loop mode (RSRC = 0) or single-wire mode (RSRC = 1). Loop mode is sometimes used to check software, independent of connections in the external system, to help isolate system problems. In this mode, the transmitter output is internally connected to the receiver input and the RxD pin is not used by the SCI, so it reverts to a general purpose port I/O pin.

#### 5.16.3.5.4 Single-wire Operation

When LOOPS = 1, the RSRC bit in the same register chooses between loop mode (RSRC = 0) or single-wire mode (RSRC = 1). Single-wire mode is used to implement a half-duplex serial connection. The receiver is internally connected to the transmitter output and to the TxD pin. The RxD pin is not used and reverts to a general purpose port I/O pin.

In single-wire mode, the TXDIR bit in SCIC3 controls the direction of serial data on the TxD pin. When TXDIR = 0, the TxD pin is an input to the SCI receiver and the transmitter is temporarily disconnected from the TxD pin so an external device can send serial data to the receiver. When TXDIR = 1, the TxD pin is an output driven by the transmitter. In single-wire mode, the internal loop back connection from the transmitter to the receiver causes the receiver to receive characters sent out by the transmitter.




## 5.17 High Voltage Inputs - Lx

Six high voltage capable inputs are implemented with the following features:

•Digital Input Capable

- Analog Input Capable with selectable voltage divider.
- Wake-up Capable during Low Power mode. See Section 5.9, “Wake-up / Cyclic Sense".

When used as analog inputs to sense voltages outside the module a series resistor must be used on the used input. When a Lx input is not selected in the analog multiplexer, the voltage divider is disconnected from this input. When a Lx input is selected in the analog multiplexer, it is disconnected in low power mode if configured as Wake-up input. Unused Lx pins are recommended to be connected to GND to improve EMC behavior.

### 5.17.1 Register Definition

#### 5.17.1.1 Lx Status Register (LXR)

**Table 149. Lx Status Register (LXR)**

Offset[111] 0x08 — Access: User read

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | L5 | L4 | L3 | L2 | L1 | L0 |
| W | | | | | | | | |

Note:
111. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 150. LXR - Register Field Descriptions**

| Field | Description |
|---|---|
| L[5-0] | Lx Status Register - Current Digital State of the Lx Input |

#### 5.17.1.2 Lx Control Register (LXCR)

**Table 151. Lx Control Register (LXCR)**

Offset[112] 0x09 — Access: User read/write

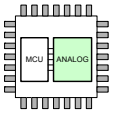
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | L5DS | L4DS | L3DS | L2DS | L1DS | L0DS |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
112. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 152. LXCR - Register Field Descriptions**

| Field | Description |
|---|---|
| 5-0<br>L[5-0]DS | Analog Input Divider Ratio Selection - Lx<br>0 - 2 (typ.)<br>1 - 7.2 (typ) |

## 5.18 General Purpose I/O - PTB[0…2]




The three multipurpose I/O pins can be configured to operate as documented in Table 153.

**Table 153. General purpose I/O - Operating modes**

| Priority | Function | PTB2 | PTB1 | PTB0 | Chp/Pg |
|---|---|---|---|---|---|
| 1 (H) | 2.5 V Analog Input | AD2 | AD1 | AD0 | 5.20/135 |
| 2 | Timer Input Capture / Output Compare | TIMCH2 | TIMCH1 | TIMCH0 | 5.19/122 |
| 3 | LIN / SCI - Rx / Tx (PTB0…1) or PWM (PTB2) | PWM | Tx | Rx | 5.15/102 |
| 4 (L) | 5.0 V Input Output | PTB2 | PTB1 | PTB0 | current |

The alternate function of PTB2, PTB1 and PTB0 can be configured by selecting the function in the corresponding module (e.g. TIMER). The selection with the highest priority takes effect when more than one function is selected.

### 5.18.1 Digital I/O Functionality

All three pins act as standard digital Inputs / Outputs with selectable pull-up resistor.

### 5.18.2 Alternative SCI / LIN Functionality

For alternative serial configuration and for debug and certification purpose, PTB0 and PTB1 can be configured to connect to the internal LIN and / or SCI signals (RxD and TxD). Figure 34 shows the 4 available configurations.

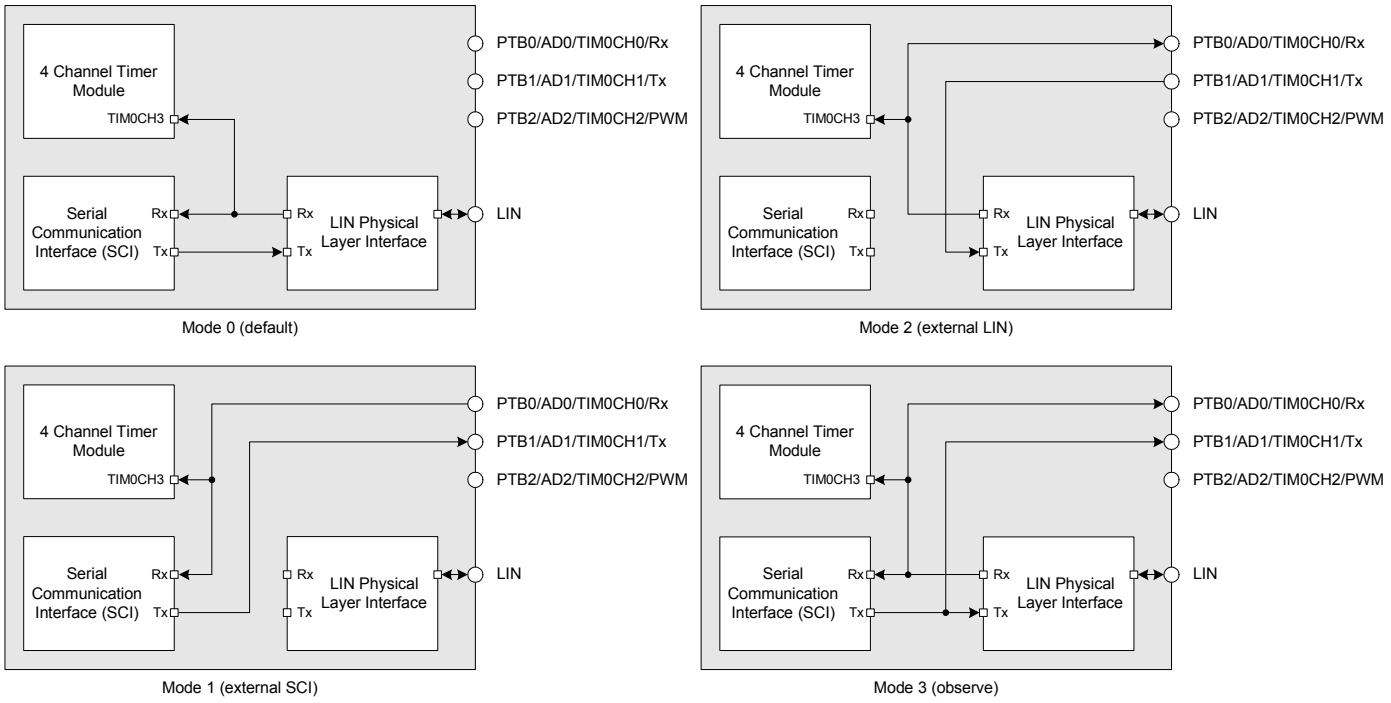



**Figure 34. Alternative SCI / LIN Functionality**

MM912_634

### 5.18.3 Alternative PWM Functionality

As an alternative routing for the PWM channel (0 or 1) output, the PortB 2 (PTB2) can be configured to output one of the two PWM channels defined in Section 5.14, “PWM Control Module (PWM8B2C)". The selection and output enable can be configured in the Port B Configuration Register 2 (PTBC2).

### 5.18.4 Register definition

#### 5.18.4.1 Port B Configuration Register 1 (PTBC1)

**Table 154. Port B Configuration Register 1 (PTBC1)**

Offset[111] 0x20 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | PUEB2 | PUEB1 | PUEB0 | 0 | DDRB2 | DDRB1 | DDRB0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
113. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 155. PTBC1 - Register Field Descriptions**

| Field | Description |
|---|---|
| 6-4<br>PUEB[2-0] | Pull-up Enable Port B[2…0]<br>0 - Pull-up disabled on PTBx pin.<br>1- Pull-up enabled on PTBx pin. |
| 2-0<br>DDRB[2-0] | Data Direction Port B[2…0]<br>0 - PTBx configured as input.<br>1 - PTBx configured as output. |

**NOTE**

The pull-up resistor is not active once the port is configured as an output.

#### 5.18.4.2 Port B Configuration Register 2 (PTBC2)

**Table 156. Port B Configuration Register 2 (PTBC2)**

Offset[114] 0x21 Access: User read/write

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

Note:
114. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 157. PTBC2 - Register Field Descriptions**

| Field | Description |
|---|---|
| 3<br>PWMCS | PWM Channel Select PTB2. See Section 5.14, “PWM Control Module (PWM8B2C)".<br>0 - PWM Channel 0 selected as PWM Channel for PTB2<br>1 - PWM Channel 1 selected as PWM Channel for PTB2 |
| 2<br>PWMEN | PWM Enable for PTB2. See Section 5.14, “PWM Control Module (PWM8B2C)".<br>0 - PWM disabled on PTB2<br>1 - PWM enabled on PTB2 (Channel as selected with PWMCS) |
| 1-0<br>SERMOD | Serial Mode Select for PTB0 and PTB1. See Figure 34 for details.<br>00 - Mode 0, SCI internally connected the LIN Physical Layer Interface. PTB0 and PTB1 are Digital I/Os<br>01 - Mode 1, SCI connected to PTB0 and PTB1 (external SCI mode)<br>10 - Mode 2, LIN Physical Layer Interface connected to PTB0 and PTB1 (external LIN mode)<br>11 - Mode 3, SCI internally connected the LIN Physical Layer Interface and PTB0 and PTB1 are connected both as outputs (Observe mode) |

### 5.18.4.3 Port B Data Register (PTB)

**Table 158. Port B Data Register (PTB)**

Offset[115] 0x22 Access: User read/write

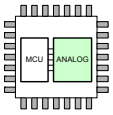
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | 0 | PTB2 | PTB1 | PTB0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
115. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 159. PTB - Register Field Descriptions**

| Field | Description |
|---|---|
| 2-0<br>PTB[2-0] | Port B general purpose input/output data — Data Register<br>If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered and synchronized pin input state is read. |

## 5.19 Basic Timer Module - TIM (TIM16B4C)




### 5.19.1 Introduction

#### 5.19.1.1 Overview

The basic timer consists of a 16-bit, software-programmable counter driven by a seven-stage programmable prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from microseconds to many seconds.

This timer contains 4 complete input capture/output compare channels [IOC 3:2]. The input capture function is used to detect a selected transition edge and record the time. The output compare function is used for generating output signals or for timer software delays.

A full access for the counter registers or the input capture/output compare registers should take place in 16bit word access. Accessing high byte and low byte separately for all of these registers may not yield the same result as accessing them in one word.

#### 5.19.1.2 Features

The TIM16B4C includes these distinctive features:

- Four input capture/output compare channels.
- Clock prescaler
- 16-bit counter

#### 5.19.1.3 Modes of Operation

The TIM16B4C is only active during Normal mode.

#### 5.19.1.4 Block Diagram

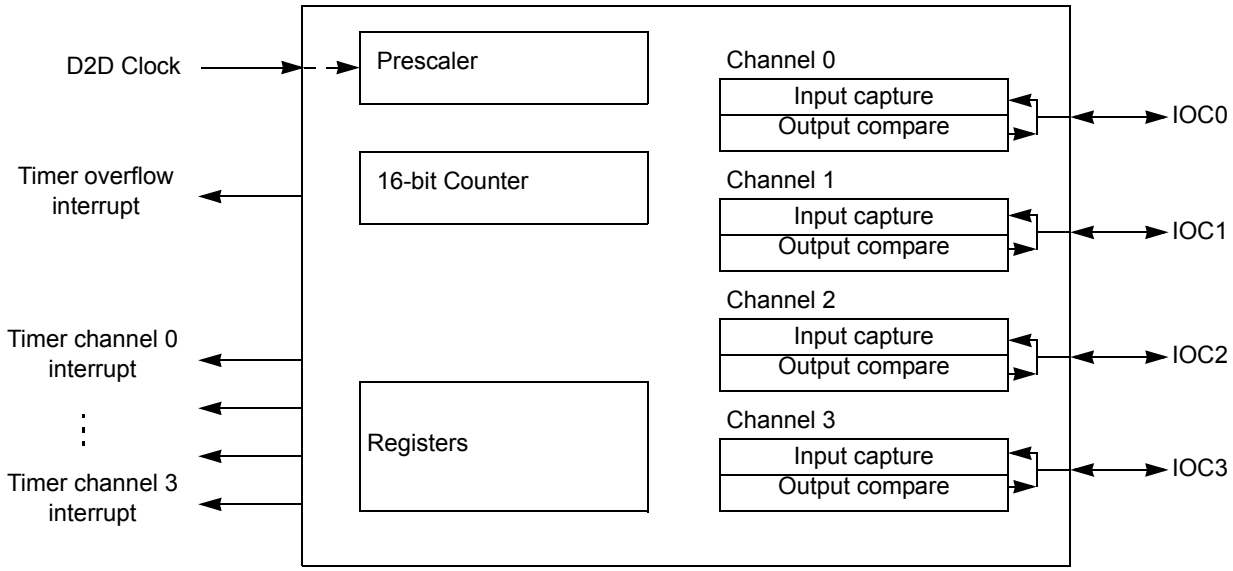



**Figure 35. Timer Block Diagram**

For more information see the respective functional descriptions see Section 5.19.4, “Functional Description" of this chapter.

### 5.19.2 Signal Description

#### 5.19.2.1 Overview

The TIM16B4C module is internally connected to the PTB (IOC0, IOC1, IOC2) and to the Rx signal as specified in Section 5.18, “General Purpose I/O - PTB[0…2]" (IOC3).

#### 5.19.2.2 Detailed Signal Descriptions

##### 5.19.2.2.1 IOC3 – Input Capture and Output Compare Channel 3

This pin serves as input capture or output compare for channel 3 and is internally connected to the Rx signal as specified in Section 5.18.2, “Alternative SCI / LIN Functionality".

**NOTE**

Since the Rx signal is only available as an input, using the output compare feature for this channel would have no effect.

##### 5.19.2.2.2 IOC2 – Input Capture and Output Compare Channel 2

This pin serves as an input capture or output compare for channel 2 and can be routed to the PTB2 general purpose I/O.

##### 5.19.2.2.3 IOC1 – Input Capture and Output Compare Channel 1

This pin serves as an input capture or output compare for channel 1 and can be routed to the PTB1 general purpose I/O.

#### 5.19.2.2.4 IOC0 – Input Capture and Output Compare Channel 0

This pin serves as an input capture or output compare for channel 0 and can be routed to the PTB0 general purpose I/O.

**NOTE**

For the description of interrupts see Section 5.19.6, “Interrupts".

### 5.19.3 Memory Map and Registers

#### 5.19.3.1 Overview

This section provides a detailed description of all memory and registers.

#### 5.19.3.2 Module Memory Map

The memory map for the TIM16B4C module is given below in Table 160.

**Table 160. Module Memory Map**

| Offset[111] | Use | Access |
|---|---|---|
| 0xC0 | Timer Input Capture/Output Compare Select (TIOS) | Read/Write |
| 0xC1 | Timer Compare Force Register (CFORC) | Read/Write[117] |
| 0xC2 | Output Compare 3 Mask Register (OC3M) | Read/Write |
| 0xC3 | Output Compare 3 Data Register (OC3D) | Read/Write |
| 0xC4 | Timer Count Register (TCNT(hi)) | Read/Write[118] |
| 0xC5 | Timer Count Register (TCNT(lo)) | Read/Write[117] |
| 0xC6 | Timer System Control Register 1 (TSCR1) | Read/Write |
| 0xC7 | Timer Toggle Overflow Register (TTOV) | Read/Write |
| 0xC8 | Timer Control Register 1 (TCTL1) | Read/Write |
| 0xC9 | Timer Control Register 2 (TCTL2) | Read/Write |
| 0xCA | Timer Interrupt Enable Register (TIE) | Read/Write |
| 0xCB | Timer System Control Register 2 (TSCR2) | Read/Write |
| 0xCC | Main Timer Interrupt Flag 1 (TFLG1) | Read/Write |
| 0xCD | Main Timer Interrupt Flag 2 (TFLG2) | Read/Write |
| 0xCE | Timer Input Capture/Output Compare Register 0 (TC0(hi)) | Read/Write[118] |
| 0xCF | Timer Input Capture/Output Compare Register 0 (TC0(lo)) | Read/Write[118] |
| 0xD0 | Timer Input Capture/Output Compare Register 1 (TC1(hi)) | Read/Write[118] |
| 0xD1 | Timer Input Capture/Output Compare Register 1 (TC1(lo)) | Read/Write[118] |
| 0xD2 | Timer Input Capture/Output Compare Register 2 (TC2(hi)) | Read/Write[118] |
| 0xD3 | Timer Input Capture/Output Compare Register 2 (TC2(lo)) | Read/Write[118] |
| 0xD4 | Timer Input Capture/Output Compare Register 3 (TC3(hi)) | Read/Write[118] |

Note:

116. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.
117. Always read $00.
118. Write to these registers have no meaning or effect during input capture.

### 5.19.3.3 Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

#### 5.19.3.3.1 Timer Input Capture/Output Compare Select (TIOS)

**Table 161. Timer Input Capture/Output Compare Select (TIOS)**

Offset[119] 0xC0 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | IOS3 | IOS2 | IOS1 | IOS0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
119. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 162. TIOS - Register Field Descriptions**

| Field | Description |
|---|---|
| 3-0<br>IOS[3-0] | Input Capture or Output Compare Channel Configuration<br>0 - The corresponding channel acts as an input capture.<br>1 - The corresponding channel acts as an output compare. |

#### 5.19.3.3.2 Timer Compare Force Register (CFORC)

**Table 163. Timer Compare Force Register (CFORC)**

Offset[120] 0xC1 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| W | | | | | FOC3 | FOC2 | FOC1 | FOC0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
120. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 164. CFORC - Register Field Descriptions**

| Field | Description |
|---|---|
| 3-0<br>FOC[3-0] | Force Output Compare Action for Channel 3-0<br>0 - Force Output Compare Action disabled. Input Capture or Output Compare Channel Configuration<br>1 - Force Output Compare Action enabled |

A write to this register with the corresponding (FOC 3:0) data bit(s) set causes the action programmed for output compare on channel “n” to occur immediately.The action taken is the same as if a successful comparison had just taken place with the TCn register except the interrupt flag does not get set.

**NOTE**

A successful channel 3 output compare overrides any channel 2:0 compares. If forced output compare on any channel occurs at the same time as the successful output compare then forced output compare action takes precedence and interrupt flag does not get set.

#### 5.19.3.3.3 Output Compare 3 Mask Register (OC3M)

**Table 165. Output Compare 3 Mask Register (OC3M)**

Offset[121] 0xC2 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | OC3M3 | OC3M2 | OC3M1 | OC3M0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
121. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 166. OC3M - Register Field Descriptions**

| Field | Description |
|---|---|
| 3-0<br>OC3M[3-0] | Output Compare 3 Mask “n” Channel bit<br>0 - Does not set the corresponding port to be an output port<br>1 - Sets the corresponding port to be an output port when this corresponding TIOS bit is set to be an output compare |

Setting the OC3Mn (n ranges from 0 to 2) sets the corresponding port to be an output port when the corresponding TIOSn (n ranges from 0 to 2) bit is set to be an output compare.

**NOTE**

A successful channel 3 output compare overrides any channel 2:0 compares. For each OC3M bit set, the output compare action reflects the corresponding OC3D bit.

#### 5.19.3.3.4 Output Compare 3 Data Register (OC3D)

**Table 167. Output Compare 3 Data Register (OC3D)**

Offset[122] 0xC3 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | OC3D3 | OC3D2 | OC3D1 | OC3D0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
122. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 168. OC3D - Register Field Descriptions**

| Field | Description |
|---|---|
| 3-0<br>OC3D[3-0] | Output Compare 3 Data for Channel “n” |

**NOTE**

A channel 3 output compare causes bits in the output compare 3 data register to transfer to the timer port data register if the corresponding output compare 3 mask register bits are set.

#### 5.19.3.3.5 Timer Count Register (TCNT)

**Table 169. Timer Count Register (TCNT)**

Offset[123] 0xC4, 0xC5 — Access: User read (anytime)/write (special mode)

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | tcnt15 | tcnt14 | tcnt13 | tcnt12 | tcnt11 | tcnt10 | tcnt9 | tcnt8 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| R<br>W | tcnt7 | tcnt6 | tcnt5 | tcnt4 | tcnt3 | tcnt2 | tcnt1 | tcnt0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
123. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 170. TCNT - Register Field Descriptions**

| Field | Description |
|---|---|
| 15-0<br>tcnt[15-0] | 16 Bit Timer Count Register |

**NOTE**

The 16-bit main timer is an up counter. A full access for the counter register should take place in one clock cycle. A separate read/write for high byte and low byte gives a different result than accessing them as a word. The period of the first count after a write to the TCNT registers may be a different length because the write is not synchronized with the prescaler clock.

#### 5.19.3.3.6 Timer System Control Register 1 (TSCR1)

**Table 171. Timer System Control Register 1 (TSCR1)**

Offset[124] 0xC6 — Access: User read/write

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

Note:
124. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 172. TSCR1 - Register Field Descriptions**

| Field | Description |
|---|---|
| 7<br>TEN | Timer Enable<br>1 = Enables the timer.<br>0 = Disables the timer. (Used for reducing power consumption). |
| 4<br>TFFCA | Timer Fast Flag Clear All<br>1 = For TFLG1 register, a read from an input capture or a write to the output compare channel [TC 3:0] causes the corresponding channel flag, CnF, to be cleared.For TFLG2 register, any access to the TCNT register clears the TOF flag. Any access to the PACNT registers clears the PAOVF and PAIF bits in the PAFLG register. This has the advantage of eliminating software overhead in a separate clear sequence. Extra care is required to avoid accidental flag clearing due to unintended accesses.<br>0 = Allows the timer flag clearing. |

### 5.19.3.3.7 Timer Toggle On Overflow Register 1 (TTOV)

**Table 173. Timer Toggle On Overflow Register 1 (TTOV)**

Offset[125] 0xC7 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | TOV3 | TOV2 | TOV1 | TOV0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
125. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 174. TTOV - Register Field Descriptions**

| Field | Description |
|---|---|
| 3-0<br>TOV[3-0] | Toggle On Overflow Bits<br>1 = Toggle output compare pin on overflow feature enabled.<br>0 = Toggle output compare pin on overflow feature disabled. |

**NOTE**

TOVn toggles output compare pin on overflow. This feature only takes effect when the corresponding channel is configured for an output compare mode. When set, an overflow toggle on the output compare pin takes precedence over forced output compare events.

### 5.19.3.3.8 Timer Control Register 1 (TCTL1)

**Table 175. Timer Control Register 1 (TCTL1)**

Offset[126] 0xC8 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | OM3 | OL3 | OM2 | OL2 | OM1 | OL1 | OM0 | OL0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
126. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 176. TCTL1 - Register Field Descriptions**

| Field | Description |
|---|---|
| 7,5,3,1<br>OMn | Output Mode bit |
| 6,4,2,0<br>OLn | Output Level bit |

**NOTE**

These four pairs of control bits are encoded to specify the output action to be taken as a result of a successful Output Compare on “n” channel. When either OMn or OLn, the pin associated with the corresponding channel becomes an output tied to its IOC. To enable output action by the OMn and OLn bits on a timer port, the corresponding bit in OC3M should be cleared.

**Table 177. Compare Result Output Action**

| OMn | OLn | Action |
|---|---|---|
| 0 | 0 | Timer disconnected from output pin logic |
| 0 | 1 | Toggle OCn output line |
| 1 | 0 | Clear OCn output line to zero |
| 1 | 1 | Set OCn output line to one |

#### 5.19.3.3.9 Timer Control Register 2 (TCTL2)

**Table 178. Timer Control Register 2 (TCTL2)**

Offset[127] 0xC9 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | EDG3B | EDG3A | EDG2B | EDG2A | EDG1B | EDG1A | EDG0B | EDG0A |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
127. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 179. TCTL2 - Register Field Descriptions**

| Field | Description |
|---|---|
| EDGnB,EDGnA | Input Capture Edge Control |

These four pairs of control bits configure the input capture edge detector circuits.

**Table 180. Edge Detector Circuit Configuration**

| EDGnB | EDGnA | Configuration |
|---|---|---|
| 0 | 0 | Capture disabled |
| 0 | 1 | Capture on rising edges only |
| 1 | 0 | Capture on falling edges only |
| 1 | 1 | Capture on any edge (rising or falling) |

#### 5.19.3.3.10 Timer Interrupt Enable Register (TIE)

**Table 181. Timer Interrupt Enable Register (TIE)**

Offset[128] 0xCA — Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | C3I | C2I | C1I | C0I |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
128. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 182. TIE - Register Field Descriptions**

| Field | Description |
|---|---|
| 3-0<br>C[3-0]I | Input Capture/Output Compare Interrupt Enable.<br>1 = Enables corresponding Interrupt flag (CnF of TFLG1 register) to cause a hardware interrupt<br>0 = Disables corresponding Interrupt flag (CnF of TFLG1 register) from causing a hardware interrupt |

#### 5.19.3.3.11 Timer System Control Register 2 (TSCR2)

**Table 183. Timer System Control Register 2 (TSCR2)**

Offset[129] 0xCB — Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | TOI | 0 | 0 | 0 | TCRE | PR2 | PR1 | PR0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
129. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 184. TIE - Register Field Descriptions**

| Field | Description |
|---|---|
| 7<br>TOI | Timer Overflow Interrupt Enable<br>1 = Hardware interrupt requested when TOF flag set in TFLG2 register.<br>0 = Hardware Interrupt request inhibited. |
| 3<br>TCRE | TCRE — Timer Counter Reset Enable<br>1 = Enables Timer Counter reset by a successful output compare on channel 3<br>0 = Inhibits Timer Counter reset and counter continues to run. |
| 3-0<br>PR[2:0] | Timer Prescaler Select<br>These three bits select the frequency of the timer prescaler clock derived from the Bus Clock as shown in Table 185. |

**NOTE**

This mode of operation is similar to an up-counting modulus counter.

If register TC3 = $0000 and TCRE = 1, the timer counter register (TCNT) stays at $0000 continuously. If register TC3 = $FFFF and TCRE = 1, TOF does not set when the timer counter register (TCNT) is reset from $FFFF to $0000.

The newly selected prescale factor does not take effect until the next synchronized edge, where all prescale counter stages equal zero.

**Table 185. Timer Clock Selection**

| PR2 | PR1 | PR0 | Timer Clock |
|---|---|---|---|
| 0 | 0 | 0 | D2D Clock / 1 |
| 0 | 0 | 1 | D2D Clock / 2 |
| 0 | 1 | 0 | D2D Clock / 4 |
| 0 | 1 | 1 | D2D Clock / 8 |
| 1 | 0 | 0 | D2D Clock / 16 |
| 1 | 0 | 1 | D2D Clock / 32 |
| 1 | 1 | 0 | D2D Clock / 64 |
| 1 | 1 | 1 | D2D Clock / 128 |

#### 5.19.3.3.12 Main Timer Interrupt Flag 1 (TFLG1)

**Table 186. Main Timer Interrupt Flag 1 (TFLG1)**

Offset[130] 0xCC — Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | C3F | C2F | C1F | C0F |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
130. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 187. TFLG1 - Register Field Descriptions**

| Field | Description |
|---|---|
| 3-0<br>C[3:0]F | Input Capture/Output Compare Channel Flag.<br>1 = Input Capture or Output Compare event occurred<br>0 = No event (Input Capture or Output Compare event) occurred. |

**NOTE**

These flags are set when an input capture or output compare event occurs. Flag set on a particular channel is cleared by writing a one to the corresponding CnF bit. Writing a zero to CnF bit has no effect on its status. When TFFCA bit in TSCR register is set, a read from an input capture or a write into an output compare channel causes the corresponding channel flag CnF to be cleared.

#### 5.19.3.3.13 Main Timer Interrupt Flag 2 (TFLG2)

**Table 188. Main Timer Interrupt Flag 2 (TFLG2)**

Offset[121] 0xCD — Access: User read/write

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

Note:
131. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 189. TFLG2 - Register Field Descriptions**

| Field | Description |
|---|---|
| 7<br>TOF | Timer Overflow Flag<br>1 = Indicates an Interrupt has occurred (Set when 16-bit free-running timer counter overflows from $FFFF to $0000)<br>0 = Flag indicates an Interrupt has not occurred. |

**NOTE**

The TFLG2 register indicates when an interrupt has occurred. Writing a one to the TOF bit clears it. Any access to TCNT clears TOF bit of TFLG2 register if the TFFCA bit in TSCR register is set.

#### 5.19.3.3.14 Timer Input Capture/Output Compare Registers (TC3 - TC0)

**Table 190. Timer Input Capture/Output Compare Register 0 (TC0)**

Offset[132] 0xCE, 0xCF — Access: User read (anytime)/write [133]

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | tc0_15 | tc0_14 | tc0_13 | tc0_12 | tc0_11 | tc0_10 | tc0_9 | tc0_8 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| R<br>W | tc0_7 | tc0_6 | tc0_5 | tc0_4 | tc0_3 | tc0_2 | tc0_1 | tc0_0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
132. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.
133. TRead anytime. Write anytime for output compare functions Writes to these registers have no effect during input capture.

**Table 191. Timer Input Capture/Output Compare Register 1(TC1)**

Offset[134] 0xD0, 0xD1 — Access: User read (anytime)/write [135]

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | tc1_15 | tc1_14 | tc1_13 | tc1_12 | tc1_11 | tc1_10 | tc1_9 | tc1_8 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| R<br>W | tc1_7 | tc1_6 | tc1_5 | tc1_4 | tc1_3 | tc1_2 | tc1_1 | tc1_0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
134. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.
135. TRead anytime. Write anytime for output compare functions Writes to these registers have no effect during input capture.

**Table 192. Timer Input Capture/Output Compare Register 2(TC2)**

Offset[136] 0xD2, 0xD3 Access: User read (anytime)/write [137]

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | tc2_15 | tc2_14 | tc2_13 | tc2_12 | tc2_11 | tc2_10 | tc2_9 | tc2_8 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| R<br>W | tc2_7 | tc2_6 | tc2_5 | tc2_4 | tc2_3 | tc2_2 | tc2_1 | tc2_0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
136. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.
137. TRead anytime. Write anytime for output compare functions Writes to these registers have no effect during input capture.

**Table 193. Timer Input Capture/Output Compare Register 3(TC3)**

Offset[138] 0xD4, 0xD5 Access: User read (anytime)/write [139]

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | tc3_15 | tc3_14 | tc3_13 | tc3_12 | tc3_11 | tc3_10 | tc3_9 | tc3_8 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| R<br>W | tc3_7 | tc3_6 | tc3_5 | tc3_4 | tc3_3 | tc3_2 | tc3_1 | tc3_0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
138. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.
139. TRead anytime. Write anytime for output compare functions Writes to these registers have no effect during input capture.

**Table 194. TCn - Register Field Descriptions**

| Field | Description |
|---|---|
| 15-0<br>tcn[15-0] | 16 Timer Input Capture/Output Compare Registers |

**NOTE**

TRead anytime. Write anytime for output compare function. Writes to these registers have no effect during input capture.

Depending on the TIOS bit for the corresponding channel, these registers are used to latch the value of the free-running counter when a defined transition is sensed by the corresponding input capture edge detector or to trigger an output action for output compare.

Read/Write access in byte mode for high byte should takes place before low byte otherwise it gives a different result.

### 5.19.4 Functional Description

#### 5.19.4.1 General

This section provides a complete functional description of the timer TIM16B4C block. Refer to the detailed timer block diagram in Figure 36 as necessary.

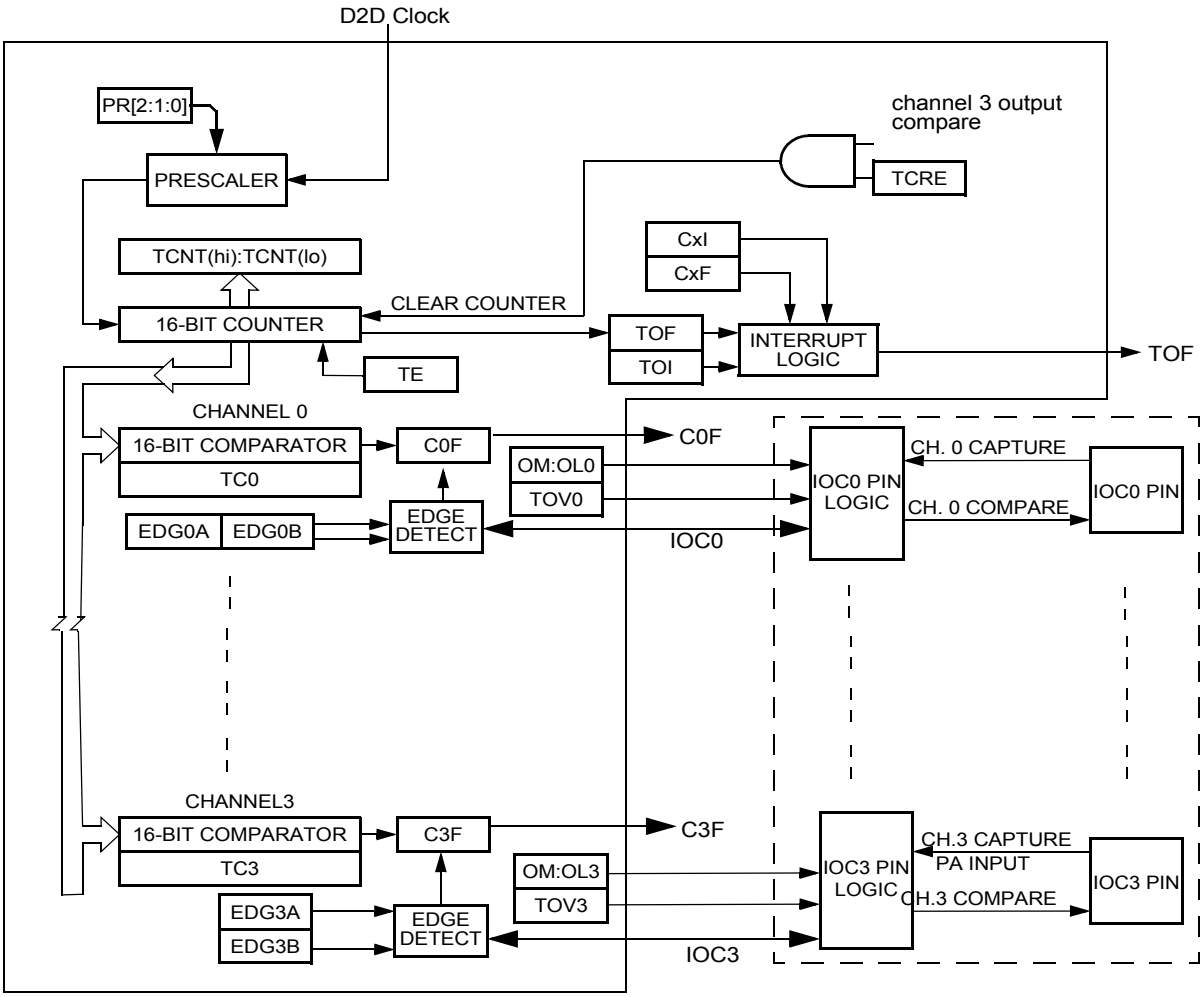



**Figure 36. Detailed Timer Block Diagram**

#### 5.19.4.2 Prescaler

The prescaler divides the bus clock by 1, 2, 4, 8, 16, 32, 64, or 128. The prescaler select bits, PR[2:0], select the prescaler divisor. PR[2:0] are in the timer system control register 2 (TSCR2).

#### 5.19.4.3 Input Capture

Clearing the I/O (input/output) select bit, IOSn, configures channel n as an input capture channel. The input capture function captures the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the timer transfers the value in the timer counter into the timer channel registers, TCn.

The minimum pulse width for the input capture input is greater than two bus clocks. An input capture on channel n sets the CnF flag. The CnI bit enables the CnF flag to generate interrupt requests.

#### 5.19.4.4 Output Compare

Setting the I/O select bit, IOSn, configures channel n as an output compare channel. The output compare function can generate a periodic pulse with a programmable polarity, duration, and frequency. When the timer counter reaches the value in the channel registers of an output compare channel, the timer can set, clear, or toggle the channel pin. An output compare on channel n sets the CnF flag. The CnI bit enables the CnF flag to generate interrupt requests.

The output mode and level bits, OMn and OLn, select set, clear, toggle on output compare. Clearing both OMn and OLn disconnects the pin from the output logic. Setting a force output compare bit, FOCn, causes an output compare on channel n. A forced output compare does not set the channel flag.

A successful output compare on channel 3 overrides output compares on all other output compare channels. The output compare 3 mask register masks the bits in the output compare 3 data register. The timer counter reset enable bit, TCRE, enables channel 3 output compares to reset the timer counter. A channel 3 output compare can reset the timer counter even if the IOC3 pin is being used as the pulse accumulator input.

Writing to the timer port bit of an output compare pin does not affect the pin state. The value written is stored in an internal latch. When the pin becomes available for general-purpose output, the last value written to the bit appears at the pin.

### 5.19.5 Resets

#### 5.19.5.1 General

The reset state of each individual bit is listed within the Register Description section 5.19.3, “Memory Map and Registers“, which details the registers and their bit-fields.

### 5.19.6 Interrupts

#### 5.19.6.1 General

This section describes interrupts originated by the TIM16B4C block. Table 195 lists the interrupts generated by the TIM16B4C to communicate with the MCU.

**Table 195. TIM16B4C Interrupts**

| Interrupt | Offset | Vector | Priority | Source | Description |
|---|---|---|---|---|---|
| C[3:0]F | - | - | - | Timer Channel 3-0 | Active high timer channel interrupts 3-0 |
| TOF | - | - | - | Timer Overflow | Timer Overflow interrupt |

#### 5.19.6.2 Description of Interrupt Operation

The TIM16B4C uses a total of 5 interrupt vectors. The interrupt vector offsets and interrupt numbers are chip dependent. More information on interrupt vector offsets and interrupt numbers can be found in Section 5.7, “Interrupts".

##### 5.19.6.2.1 Channel [3:0] Interrupt

These active high outputs are asserted by the module to request a timer channel 3–0 interrupt, following an input capture or output compare event on these channels [3-0]. For the interrupt to be asserted on a specific channel, the enable, CnI bit of TIE register should be set. These interrupts are serviced by the system controller.

##### 5.19.6.2.2 Timer Overflow Interrupt (TOF)

This active high output is asserted by the module to request a timer overflow interrupt, following the timer counter overflow when the overflow enable bit (TOI) bit of TFLG2 register is set. This interrupt is serviced by the system controller.

## 5.20 Analog Digital Converter - ADC

### 5.20.1 Introduction

#### 5.20.1.1 Overview

In order to sample the MM912_634 analog die analog sources, a 10-bit resolution successive approximation Analog to Digital Converter has been implemented. Controlled by the A/D Control Logic (ADC Wrapper), the Analog Digital Converter allows fast and high precision conversions.

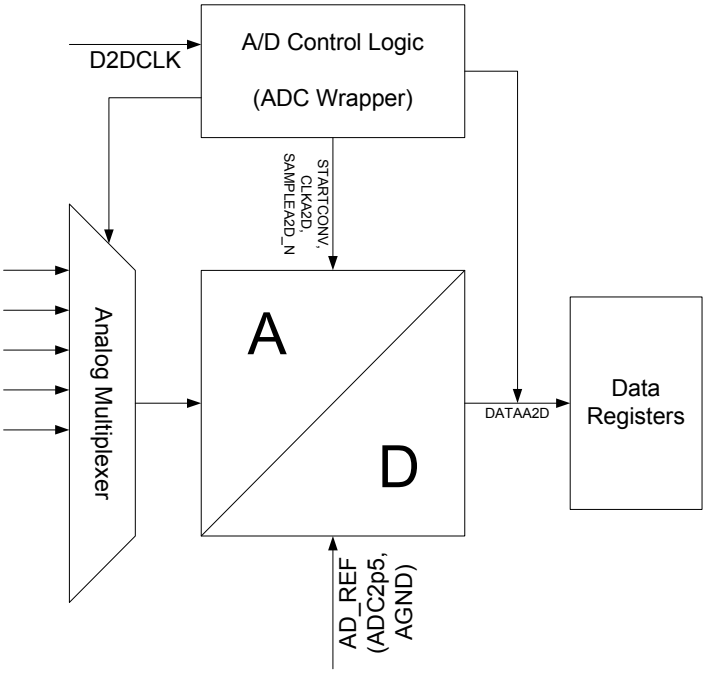



**Figure 37. Analog Digital Converter Block Diagram**

#### 5.20.1.2 Features

- 10-bit resolution
- 13 µs (typ.), 10-bit Single Sample + Conversion Time
- External ADC2p5 pin with overcurrent protection to filter the analog reference voltage
- Total Error (TE) of ± 5 LSB without offset calibration active
- Integrated selectable offset compensation
- 14 + 1 analog channels (AD0…8; ISENSE, TSENSE and VSENSE, VS1SENSE, BANDGAP, plus calibration channel)
- Sequence- and Continuous Conversion Mode with IRQ for Sequence Complete indication
- Dedicated Result register for each channel

### 5.20.2 Modes of Operation

The Analog Digital Converter Module is active only in normal mode; it is disabled in Sleep and Stop mode.

### 5.20.3 External Signal Description

This section lists and describes the signals connected off-chip. Table 196 shows all the pins and their functions which are controlled by the Analog Digital Converter Module.

**Table 196. ADC - Pin Functions and Priorities**

| Pin Name | Pin Function & Priority | I/O | Description | Pin Function after Reset |
|---|---|---|---|---|
| AGND | Analog Ground | - | Analog Ground Connection | - |
| ADC2p5 | Analog Regulator | - | Analog Digital Converter Regulator Filter Terminal. A capacitor $C_{ADC2p5}$ is required for operation. | - |

### 5.20.4 Memory Map and Register Definition

#### 5.20.4.1 Module Memory Map

Table 197 shows the register map of the Analog Digital Converter Module. All Register addresses given are referenced to the D2D interface offset.

**Table 197. Analog Digital Converter Module - Memory Map**

| Register / Offset[140] | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0x80 ACR | R | SCIE | CCE | OCE | ADCRST | 0 | PS2 | PS1 | PS0 |
| | W | | | | | | | | |
| 0x81 ASR | R | SCF | 2p5CLF | 0 | 0 | CCNT3 | CCNT2 | CCNT1 | CCNT0 |
| | W | | | | | | | | |
| 0x82 ACCR (hi) | R | CH15 | CH14 | 0 | CH12 | CH11 | CH10 | CH9 | CH8 |
| | W | | | | | | | | |
| 0x83 ACCR (lo) | R | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | CH0 |
| | W | | | | | | | | |
| 0x84 ACCSR (hi) | R | CC15 | CC14 | 0 | CC12 | CC11 | CC10 | CC9 | CC8 |
| | W | | | | | | | | |
| 0x85 ACCSR (lo) | R | CC7 | CC6 | CC5 | CC4 | CC3 | CC2 | CC1 | CC0 |
| | W | | | | | | | | |
| 0x86 ADR0 (hi) | R | ADR0[9:2] | | | | | | | |
| | W | | | | | | | | |
| 0x87 ADR0 (lo) | R | ADR0[1:0] | | | | | | | |
| | W | | | | | | | | |
| 0x88 ADR1 (hi) | R | ADR1[9:2] | | | | | | | |
| | W | | | | | | | | |
| 0x89 ADR1 (lo) | R | ADR1[1:0] | | | | | | | |
| | W | | | | | | | | |
| 0x8A ADR2 (hi) | R | ADR2[9:2] | | | | | | | |
| | W | | | | | | | | |
| 0x8B ADR2 (lo) | R | ADR2[1:0] | | | | | | | |
| | W | | | | | | | | |

**Table 197. Analog Digital Converter Module - Memory Map (continued)**

| Register / Offset[140] | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0x8C ADR3 (hi) | R | ADR3[9:2] | | | | | | | |
| | W | | | | | | | | |
| 0x8D ADR3 (lo) | R | ADR3[1:0] | | | | | | | |
| | W | | | | | | | | |
| 0x8E ADR4 (hi) | R | ADR4[9:2] | | | | | | | |
| | W | | | | | | | | |
| 0x8F ADR4 (lo) | R | ADR4[1:0] | | | | | | | |
| | W | | | | | | | | |
| 0x90 ADR5 (hi) | R | ADR5[9:2] | | | | | | | |
| | W | | | | | | | | |
| 0x91 ADR5 (lo) | R | ADR5[1:0] | | | | | | | |
| | W | | | | | | | | |
| 0x92 ADR6 (hi) | R | ADR6[9:2] | | | | | | | |
| | W | | | | | | | | |
| 0x93 ADR6 (lo) | R | ADR6[1:0] | | | | | | | |
| | W | | | | | | | | |
| 0x94 ADR7 (hi) | R | ADR7[9:2] | | | | | | | |
| | W | | | | | | | | |
| 0x95 ADR7 (lo) | R | ADR7[1:0] | | | | | | | |
| | W | | | | | | | | |
| 0x96 ADR8 (hi) | R | ADR8[9:2] | | | | | | | |
| | W | | | | | | | | |
| 0x97 ADR8 (lo) | R | ADR8[1:0] | | | | | | | |
| | W | | | | | | | | |
| 0x98 ADR9 (hi) | R | ADR9[9:2] | | | | | | | |
| | W | | | | | | | | |
| 0x99 ADR9 (lo) | R | ADR9[1:0] | | | | | | | |
| | W | | | | | | | | |
| 0x9A ADR10 (hi) | R | ADR10[9:2] | | | | | | | |
| | W | | | | | | | | |
| 0x9B ADR10 (lo) | R | ADR10[1:0] | | | | | | | |
| | W | | | | | | | | |
| 0x9C ADR11 (hi) | R | ADR11[9:2] | | | | | | | |
| | W | | | | | | | | |

**Table 197. Analog Digital Converter Module - Memory Map (continued)**

| Register / Offset[140] | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0x9D ADR11 (lo) | R | ADR11[1:0] | | | | | | | |
| | W | | | | | | | | |
| 0x9E ADR12 (hi) | R | ADR12[9:2] | | | | | | | |
| | W | | | | | | | | |
| 0x9F ADR12 (lo) | R | ADR12[1:0] | | | | | | | |
| | W | | | | | | | | |
| 0xA0 Reserved | R | | | | | | | | |
| | W | | | | | | | | |
| 0xA1 Reserved | R | | | | | | | | |
| | W | | | | | | | | |
| 0xA2 ADR14 (hi) | R | ADR14[9:2] | | | | | | | |
| | W | | | | | | | | |
| 0xA3 ADR14 (lo) | R | ADR14[1:0] | | | | | | | |
| | W | | | | | | | | |
| 0xA4 ADR15 (hi) | R | ADR15[9:2] | | | | | | | |
| | W | | | | | | | | |
| 0xA5 ADR15 (lo) | R | ADR15[1:0] | | | | | | | |
| | W | | | | | | | | |

Note:
140. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

## 5.20.4.2 Register Definition

### 5.20.4.2.1 ADC Config Register (ACR)

**Table 198. ADC Config Register (ACR)**

Offset[141] 0x80 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | SCIE | CCE | OCE | ADCRST | 0 | PS2 | PS1 | PS0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
141. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 199. ACR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7 - SCIE | Sequence Complete Interrupt Enable<br>0 - Sequence Complete Interrupt Disabled<br>1 - Sequence Complete Interrupt Enabled |
| 6 - CCE | Continuous Conversion Enable<br>0 - Continuous Conversion Disabled<br>1 - Continuous Conversion Enabled |
| 5 - OCE | Offset Compensation Enable<br>0 - Offset Compensation Disabled<br>1 - Offset Compensation Enabled. This feature requires the CH15 bit in the ADC Conversion Control Register (ACCR) to be set for all conversions |
| 4 - ADCRST | Analog Digital Converter RESET<br>0 - Analog Digital Converter in Normal Operation<br>1 - Analog Digital Converter in Reset Mode. All ADC registers are reset to initial values. The bit has to be cleared to allow ADC operation |
| 2-0<br>PS2…0 | ADC Clock Prescaler Select (D2DCLK to ADCCLK divider)<br>000 - 10<br>001 - 8<br>010 - 6<br>011 - 4<br>100 - 2<br>101 - 1<br>110 - 1<br>111 - 1 |

**NOTE**

ADCRST is strongly recommended to be set during D2D clock frequency changes.

#### 5.20.4.2.2 ADC Status Register (ASR)

**Table 200. ADC Status Register (ASR)**

Offset[142] 0x81 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | SCF | 2p5CLF | 0 | 0 | CCNT3 | CCNT2 | CCNT1 | CCNT0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

Note:
142. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 201. ACR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7 - SCF | Sequence Complete Flag. Reading the ADC Status Register (ASR) clears the Flag. |
| 6 - 2p5CLF | ADC Reference Voltage Current Limitation Flag |
| 3-0<br>CCNT3…0 | Conversion Counter Status. The content of CCNT reflects the current channel in conversion and the conversion of CCNT-1 being complete. The conversion order is CH15, CH0, CH1,..., CH14. |

#### 5.20.4.2.3 ADC Conversion Control Register (ACCR)

**Table 202. ADC Conversion Control Register (ACCR)**

Offset[143] 0x82 (0x82 and 0x83 for 8-Bit access) Access: User read/write

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | CH15 | CH14 | 0 | CH12 | CH11 | CH10 | CH9 | CH8 | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | CH0 |
| W | | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
143. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 203. ACCR - Register Field Descriptions**

| Field | Description |
|---|---|
| 15-0 CHx | Channel Select - If 1, the selected channel is included into the sequence. Writing ACCR stops the current sequence and restart. Writing ACCR=0 stops the conversion, All CCx flags are cleared when ACCR is written.Conversion starts after write. 16-Bit write operation recommended, writing 8-bit: Only writing the High Byte starts the conversion with Channel 15, if selected. Write to the Low Byte does not start a conversion.<br>Measure individual Channels by writing a sequence of one channel. Channel 15 needs to be selected in order to have the offset compensation functional. |

#### 5.20.4.2.4 ADC Conversion Complete Status Register (ACCSR)

**Table 204. ADC Conversion Complete Status Register (ACCSR)**

Offset[144] 0x84 (0x84 and 0x85 for 8-Bit access) Access: User read

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | CC15 | CC14 | 0 | CC12 | CC11 | CC10 | CC9 | CC8 | CC7 | CC6 | CC5 | CC4 | CC3 | CC2 | CC1 | CC0 |
| W | | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
144. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 205. ACCSR - Register Field Descriptions**

| Field | Description |
|---|---|
| 15-0 CCx | Conversion Complete Flag - Indicates the conversion being complete for channel x. Read operation only.16-bit read recommended. 8-Bit read returns the current status, no latching is performed. |

#### 5.20.4.2.5 ADC Data Result Register x (ADRx)

**Table 206. ADC Data Result Register x (ADRx)**

Offset[145] 0x86+x (0x86 and 0x87 for 8-Bit access) Access: User read

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | ADRx | | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| W | | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
145. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 207. ADRx - Register Field Descriptions**

| Field | Description |
|---|---|
| 15-6 ADRx | ADC - Channel X left adjusted Result Register. Reading the register clears the corresponding CCx register in the ACCSR register. 16-bit read recommended. 8-Bit read: Reading the low byte latches the high byte for the next read, reading the high byte clears the cc flag. |

### 5.20.5 Functional Description

#### 5.20.5.1 Analog Channel Definitions

The following analog Channels are routed to the analog multiplexer:

**Table 208. Analog Channels**

| Channel | Description | |
|---|---|---|
| 0 | AD0 - PTB0 Analog Input | AD0 |
| 1 | AD1 - PTB1 Analog Input | AD1 |
| 2 | AD2 - PTB2 Analog Input | AD2 |
| 3 | AD3 - L0 Analog Input | AD3 |
| 4 | AD4 - L1 Analog Input | AD4 |
| 5 | AD5 - L2 Analog Input | AD5 |
| 6 | AD6 - L3 Analog Input | AD6 |
| 7 | AD7 - L4 Analog Input | AD7 |
| 8 | AD8 - L5 Analog Input | AD8 |
| 9 | Current Sense | ISENSE |
| 10 | Voltage Sense | VSENSE |
| 11 | Temperature Sense | TSENSE |
| 12 | VS1 Sense | VS1SENSE |
| 13 | not implemented | n.i. |
| 14 | Bandgap[146] | BANDGAP |
| 15 | Calibration Reference | CAL |

Note:
146. Internal “bg1p25sleep” reference.

### 5.20.5.2 Automatic Offset Compensation

To eliminate the Analog Digital Converter Offset, an automatic compensation is implemented. The compensation is based on a calibrated voltage reference connected to ADC Channel 15. The reference trim is accomplished by the correct CTRx Register content. See Section 5.26, “MM912_634 - Analog Die Trimming". The reference is factory trimmed to 8 LSB.

To activate the Offset compensation feature, the OCE bit in the ADC Config Register (ACR) has to be set, and the CH15 has to be enabled when starting a new conversion, by writing to the ADC Conversion Control Register (ACCR). The compensation works with single and sequence conversion.

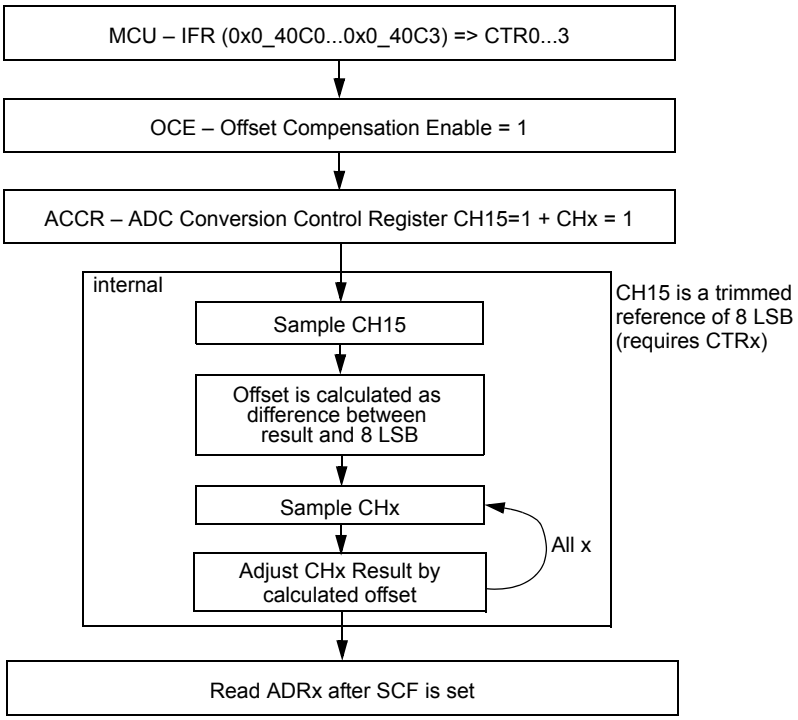



**Figure 38. Automatic Offset Compensation**

### 5.20.5.3 Conversion Timing

The conversion timing is based on the ADCCLK generated by the ADC prescaler (PS) out of the D2DCLK signal. The prescaler needs to be configured to have the ADCCLK match the specified $f_{ADC}$ clock limits.

A conversion is divided into the following 27+ clock cycles:

- 9 cycle sampling time
- 18 cycle remaining conversion time
- A worst case (only channel 14) of 15 clock cycles to count up to the selected channel (15, 0, 1,....14)
- 4 cycles between two channels

**Example 1. Single Conversion Channel 10 (VSENSE)**

12c (count up to Ch10) + 9c (sample) + 18c (conversion) = 39 cycles from start to end of conversion.

**Example 2. Sequence of Channel 10 (VSENSE) + Channel 15 (Offset Compensation)**

1c (count) + 9c (sample Ch15) + 18c (conversion Ch15) + 4c (in between) + 0c (count further to Ch10 is performed while converting ch15) + 9c (sample) + 18c (conversion) = 59 cycles from start to end of both conversions.

## 5.21 Current Sense Module - ISENSE

The Current Sense Module is implemented to amplify the voltage drop across an external shunt resistor to measure the actual application current using the internal Analog Digital Converter Channel 9. Typical application is the motor current in a window lift control module

.

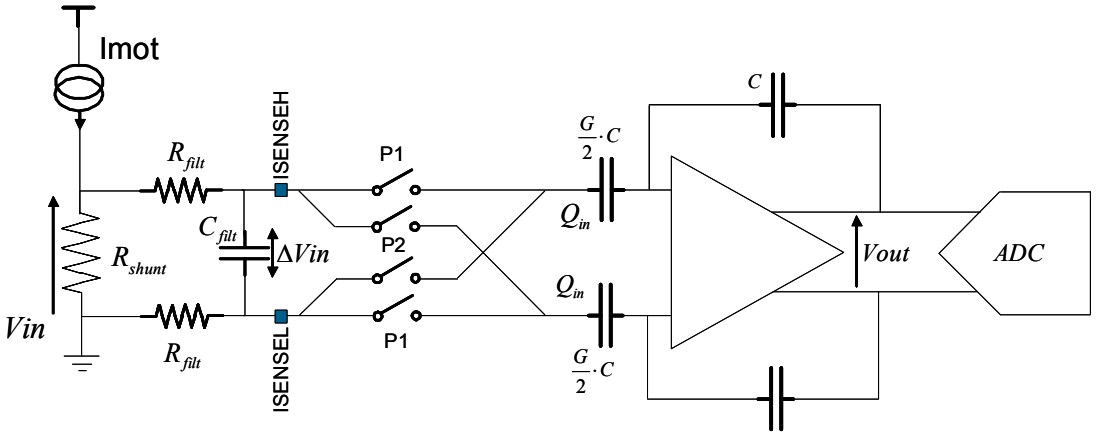



**Figure 39. Current Sense Module with External Filter Option**

The implementation is based on a switched capacitor solution to eliminate unwanted offset.To fit several application scenarios, eight different GAIN setting are implemented.

### 5.21.1 Register Definition

#### 5.21.1.1 Current Sense Register (CSR)

**Table 209. Current Sense Register (CSR)**

Offset[(144)] 0x3C — Access: User read/write

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

Note:
147. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 210. CSR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7<br>CSE | Current Sense Enable Bit<br>0 - Current Sense Module Disabled<br>1 - Current Sense Module Enabled |
| 3<br>CCD | Input Filter Charge Compensation Disable Bit(148)<br>0 - Enabled<br>1 - Disabled |
| 2-0<br>CSGS | Current Sense Gain Select - Selects the amplification GAIN for the current sense module<br>000 - 7 (typ.)<br>001 - 9 (typ.)<br>010 - 10 (typ.)<br>011 - 12 (typ.)<br>100 - 14 (typ.)<br>101 - 18 (typ.)<br>110 - 24 (typ.)<br>111 - 36 (typ.) |

Note:
148. This feature should be used when implementing an external filter to the current sense ISENSEx inputs. In principal an internal charge compensation is activated in synch with the conversion to avoid the sample capacitors to be discharged by the external filter.

## 5.22Temperature Sensor - TSENSE

To be able to measure the current MM912_634 analog die chip temperature, the TSENSE feature is implemented. A constant temperature related gain of TSG can be routed to the internal Analog Digital Converter (Channel 11).

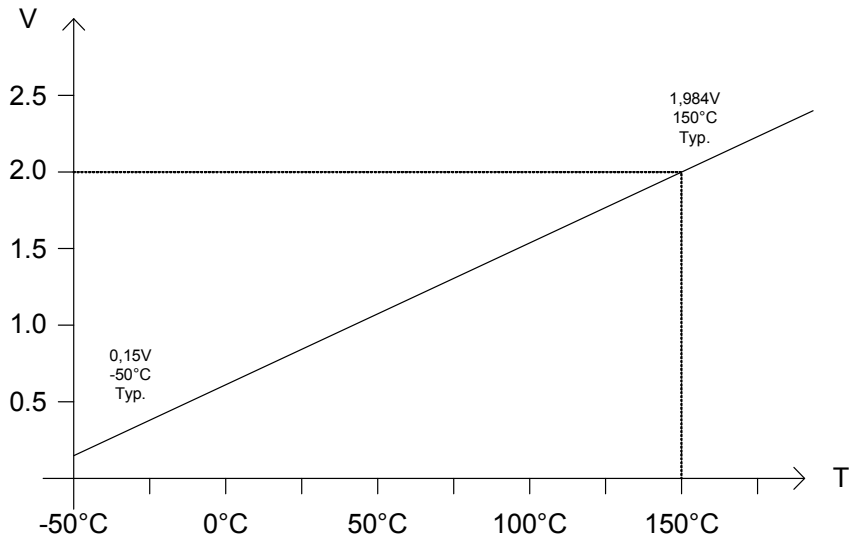



**Figure 40. TSENSE - Graph**

Refer to Section 5.20, “Analog Digital Converter - ADC" for details on the channel selection and analog measurement.

**NOTE**

Due to internal capacitor charging, temperature measurements are valid 200 ms (max) after system power up and wake-up.

## 5.23 Supply Voltage Sense - VSENSE

The reverse battery protected VSENSE pin has been implemented to allow a direct measurement of the Battery level voltage. Bypassing the device VSUP capacitor and external reverse battery diode detects undervoltage conditions without delay. A series resistor is required to protect the MM912_634 analog die from fast transients.

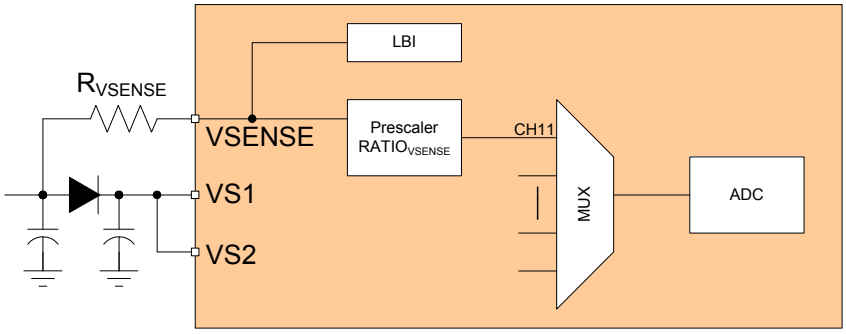



**Figure 41. VSENSE Module**

The voltage present on the VSENSE pin can be routed via an internal divider to the internal Analog Digital Converter or issue an interrupt (LBI) to alert the MCU.

For the interrupt based alert, see Section 5.5, “Power Supply". For VSENSE measurement using the internal ADC see Section 5.20, “Analog Digital Converter - ADC".

## 5.24 Internal Supply Voltage Sense - VS1SENSE

In addition to the VSENSE module, the internal VS1 supply can be routed to the analog digital converter as well. See Section 5.20, “Analog Digital Converter - ADC" for details on the acquisition.

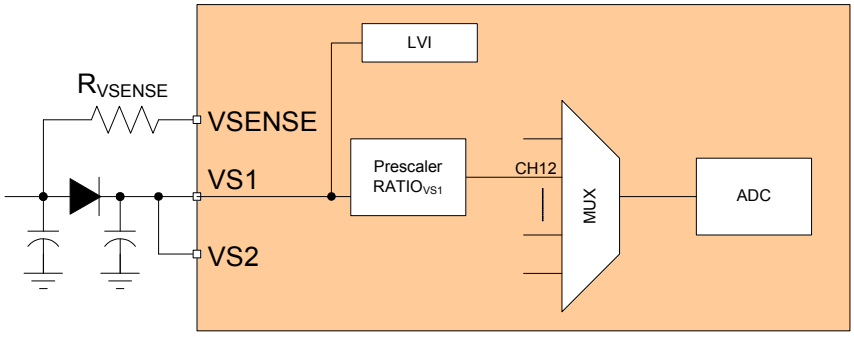



**Figure 42. VS1Sense Module**

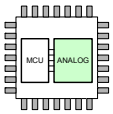
## 5.25 Internal Bandgap Reference Voltage Sense - BANDGAP

The internal reference bandgap voltage “bg1p25sleep” is generated fully independent from the Analog Digital Converter reference voltages.

Measuring[149] the “bg1p25sleep” reference through the ADC-CH14 allows should return a conversion result within ADCH14 under normal conditions. Any result outside the range would indicate faulty behavior of either the ADC chain or the 2p5sleep Bandgap circuity.

Note:

149. The maximum allowed sample frequency for Channel 14 is limited to fCH14. Increasing the sample frequency above can result in unwanted turn off of the LS drivers due to a false VREG overvoltage.




## 5.26MM912_634 - Analog Die Trimming

A trimming option is implemented to increase the accuracy of some device parameters. As the MM912_634 analog die is exclusively combined with a FLASH- MCU, the required trimming values are calculated during the final test of the device and stored to a fixed position in the flash memory.The four trimming register are completely filled (data bits, trim enable bit, trim lock bit for CTR2) and stored in flash memory. During start-up of the system, complete CTRx registers have to be copied into the MM912_634 analog die trimming registers.

The trimming registers maintain their contents when the device enters Low-power mode. If a RESET is asserted, the default value is set for the register contents for CTRx. However, for CTR2, even if the CTR2 register value is reset, the effect on the Band Gap is not reset if the Sleep Band Gap (SLPBG) Lock Bit was set to 1.

There is a possibility for the user to experience the unintended loading and locking of the default SLPBG trim value into CTR2 during abnormal power-up conditions. It is therefore recommended the user mask this Lock Bit through software during loading of the CTR2 trim values. See MM912_634ER - MM912_634, Silicon Analog Mask (M91W) / Digital Mask (N53A) Errata.

### 5.26.1 Memory Map and Register Definition

#### 5.26.1.1 Module Memory Map

There are four trimming registers implemented (CTR0… CTR3), with CTR3 being reserved for future use. The following table shows the registers used.

**Table 211. MM912_634 Analog Die Trimming Registers**

| Offset (150) | Name | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0xF0 | CTR0 | R | LINTRE | LINTR | WDCTRE | CTR0_4 | CTR0_3 | WDCTR2 | WDCTR1 | WDCTR0 |
| | Trimming Reg 0 | W | | | | | | | | |
| 0xF1 | CTR1 | R | BGTRE | CTR1_6 | BGTRIMUP | BGTRIMDN | IREFTRE | IREFTR2 | IREFTR1 | IREFTR0 |
| | Trimming Reg 1 | W | | | | | | | | |
| 0xF2 | CTR2 | R | 0 | 0 | 0 | SLPBGTRE | SLPBG_LOCK | SLPBGTR2 | SLPBGTR1 | SLPBGTR0 |
| | Trimming Reg 2 | W | | | | | | | | |
| 0xF3 | CTR3 | R | OFFCTRE | OFFCTR2 | OFFCTR1 | OFFCTR0 | CTR3_E | CTR3_2 | CTR3_1 | CTR3_0 |
| | Trimming Reg 3 | W | | | | | | | | |

Note:
150. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

At system startup, the trimming information have to be copied from the MCU IFR Flash location to the corresponding MM912_634 analog die trimming registers. The following table shows the register correlation.

**Table 212. MM912_634 - MCU vs. Analog Die Trimming Register Correlation**

| Name | MCU IFR Address | Analog Offset(151) |
|---|---|---|
| CTR0 | 0x0_40C0 | 0xF0 |
| CTR1 | 0x0_40C1 | 0xF1 |
| CTR2 | 0x0_40C2 | 0xF2 |
| CTR3 | 0x0_40C3 | 0xF3 |

Note:
151. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**NOTE**

Two word (16-Bit) transfers including CTR2 are recommended at system startup. The IFR register has to be enabled for reading (Section 5.29.3.2.3, “Program Page Index Register (PPAGE)")

### 5.26.1.2 Register Descriptions

#### 5.26.1.2.1 Trimming Register 0 (CTR0)

**Table 213. Trimming Register 0 (CTR0)**

Offset[152] 0xF0 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | LINTRE | LINTR | WDCTRE | CTR0_4 | CTR0_3 | WDCTR2 | WDCTR1 | WDCTR0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
152. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 214. CTR0 - Register Field Descriptions**

| Field | Description |
|---|---|
| 7<br>LINTRE | LIN trim enable<br>0 - no trim can be done<br>1- trim can be done by setting LINTR bit |
| 6<br>LINTR | LIN trim bit<br>0 - default slope<br>1 - adjust the slope |
| 5<br>WDCTRE | Watchdog trim enable<br>0 - no trim can be done<br>1 - trim can be done by setting WDCTR[2:0] bits |
| 4<br>CTR0_4 | Spare Trim bit 4 |
| 3<br>CTR0_3 | Spare Trim bit 3 |
| 2-0<br>WDCTR2…0 | Watchdog clock trim (Trim effect to the 100 kHz Watch dog base clock)<br>000: 0%<br>001: +5%<br>010: +10%<br>011: +15%<br>100: -20%<br>101: -15%<br>110: -10%<br>111: -5% |

#### 5.26.1.2.2 Trimming Register 1 (CTR1)

**Table 215. Trimming Register 1 (CTR1)**

Offset[140] 0xF1 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | BGTRE | CTR1_6 | BGTRIMUP | BGTRIMDN | IREFTRE | IREFTR2 | IREFTR1 | IREFTR0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
153. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 216. CTR1 - Register Field Descriptions**

| Field | Description |
|---|---|
| 7<br>BGTRE | Bandgap trim enable<br>0 - no trim can be done<br>1 - trim can be done by setting BGTRIMUP and BGTRIMDN bits |
| 6<br>CTR1_6 | Spare Trim Bit |
| 5<br>BGTRIMUP | Bandgap trim up bit<br>0 - default slope<br>1 - increase bandgap slope |
| 4<br>BGTRIMDN | Bandgap trim down bit<br>0 - default slope<br>1 - decrease bandgap slope |
| 3<br>IREFTRE | Iref trim enable bit<br>0 - no trim can be done<br>1 - trim can be done by setting IREFTR[2:0] bits |
| 2-0<br>IREFTR2…0 | Iref trim - This trim is used to adjust the internal zero TC current reference<br>000: 0%<br>001: +7.6%<br>010: +16.43%<br>011: +26.83%<br>100: -8.54%<br>101: -15.75%<br>110: -21.79%<br>111: 0% |

#### 5.26.1.2.3 Trimming Register 2 (CTR2)

**Table 217. Trimming Register 2 (CTR2)**

Offset[154] 0xF2 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | SLPBGTRE | SLPBG_LOCK | SLPBGTR2 | SLPBGTR1 | SLPBGTR0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
154. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 218. CTR2 - Register Field Descriptions**

| Field | Description |
|---|---|
| 4<br>SLPBGTRE | Sleep Bandgap trim enable<br>0 no trim can be done<br>1 trim lock can be done by setting SLPBGTR[2:0] bits and SLPBG_LOCK bit |
| 3<br>SLPBG_LOCK | bg1p25sleep trim lock bit. The bit enables the contents in SLPBGTR2… 0 to be locked. Once this occurs, no other trim values can be loaded into the analog trim registers of CTR2. Asserting a device RESET only clears the contents of the digital trim registers. To reset or change the contents of the analog trim registers, the device must undergo a complete power down and restart. (155) |
| 2-0<br>SLPBGTR2…0 | bg1p25sleep trim - This trim is used to adjust the internal sleep mode 1.25 V bandgap used as a reference for the VDD and VDDx overvoltage detection.<br>000: -12.2% (default)<br>001: -8.2%<br>010: -4.2%<br>011: 0%<br>100: +4.2%<br>101: +8.3%<br>110: +12.5%<br>111: -12.2% (default) |

Note:
155. See MM912_634ER - MM912_634, Silicon Analog Mask (M91W) / Digital Mask (N53A) Errata.

#### 5.26.1.2.4 Trimming Register 3 (CTR3)

**Table 219. Trimming Register 3 (CTR3)**

Offset(156) 0xF3 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | OFFCTRE | OFFCTR2 | OFFCTR1 | OFFCTR0 | CTR3_E | CTR3_2 | CTR3_1 | CTR3_0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

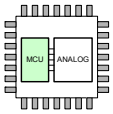
Note:
156. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 220. CTR3 - Register Field Descriptions**

| Field | Description |
|---|---|
| 7<br>OFFCTRE | ADC offset compensation voltage trim enable bit<br>0 - no trim can be done<br>1 - trim can be done by setting OFFCTR[2:0] bits |
| 6-4<br>OFFCTR2…0 | ADCOFFC trim - This trim is used to adjust the internal ADC offset compensation voltage<br>000: 0%<br>001: +7.98%<br>010: +15.97%<br>011: +23.95%<br>100: -23.95%<br>101: -15.97%<br>110: -7.98%<br>111: 0% |
| 3<br>CTR3_E | Spare Trim enable bit |
| 2<br>CTR3_2 | Spare Trim bit 2 |

**Table 220. CTR3 - Register Field Descriptions (continued)**

| Field | Description |
|---|---|
| 1<br>CTR3_1 | Spare Trim bit 1 |
| 0<br>CTR3_0 | Spare Trim bit 0 |




# 5.27MM912_634 - MCU Die Overview

## 5.27.1Introduction

The MC9S12I64 micro controller implemented in the MM912_634 is designed as counter part to an analog die, and is not being offered as a standalone MCU.

The MC9S12I64 device contains a S12 Central Processing Unit (CPU), offers 64kB of Flash memory and 6.0 kB of system SRAM, up to eight general purpose I/Os, an on-chip oscillator and clock multiplier, one Serial Peripheral Interface (SPI), an interrupt module and debug capabilities via the on-chip debug module (DBG) in combination with the Background Debug Module (BDM) interface. Additionally there is a die-to-die initiator (D2DI) which represents the communication interface to the companion (analog) die.

## 5.27.2 Features

This section describes the key features of the MC9S12I64 micro controller unit.

### 5.27.2.1 Chip-Level Features

On-chip modules available within the family include the following features:

- S12 CPU core (CPU12_V1)
- kbyte on-chip flash with ECC
- 4.0 kbyte on-chip data flash with ECC
- 6.0 kbyte on-chip SRAM
- Phase locked loop (IPLL) frequency multiplier with internal filter
- 4.0–16 MHz amplitude controlled Pierce oscillator
- 1.0 MHz internal RC oscillator
- One serial peripheral interface (SPI) module
- On-chip voltage regulator (VREG) for regulation of input supply and all internal voltages
- Die to Die Initiator (D2DI)

## 5.27.3 Module Features

The following sections provide more details of the modules implemented on the 9S12I64.

### 5.27.3.1 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 pre-decrement, pre-increment, post-decrement, and post-increment (by –8 to +8)

#### 5.27.3.2 On-Chip Flash with ECC

On-chip flash memory on the 9S12I64 features the following:

- 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
- 4.0 kbyte data flash memory
  - 16 data bits plus 6 syndrome ECC (Error Correction Code) bits allow single bit error correction and double-bit error detection
  - Erase sector size 256 bytes
  - Automated program and erase algorithm
  - User margin level setting for reads

#### 5.27.3.3 On-Chip SRAM

6.0 kBytes of general-purpose RAM

#### 5.27.3.4 Main External Oscillator (XOSC)

Loop controlled Pierce oscillator using a 4.0 MHz to 16 MHz crystal or resonator

- Current gain control on amplitude output
- Signal with low harmonic distortion
- Low power
- Good noise immunity
- Eliminates need for external current limiting resistor
- Transconductance sized for optimum start-up margin for typical crystals

#### 5.27.3.5 Internal RC Oscillator (IRC)

Trimmable internal reference clock.

#### 5.27.3.6 Internal Phase-locked Loop (IPLL)

Phase-locked loop clock frequency multiplier

- No external components required
- Reference divider and multiplier allow large variety of clock rates
- Automatic bandwidth control mode for low-jitter operation
- Automatic frequency lock detector
- Configurable option to spread spectrum for reduced EMC radiation (frequency modulation)
- Reference clock sources:
  - External 4.0 to 16 MHz resonator/crystal (XOSC)
  - Internal 1.0 MHz RC oscillator (IRC)

#### 5.27.3.7 System Integrity Support

- Power-on reset (POR)
- System reset generation
- Illegal address detection with reset
- Low-voltage detection with interrupt or reset
- Real time interrupt (RTI)
- Computer operating properly (COP) watchdog
  - Configurable as window COP for enhanced failure detection
  - Initialized out of reset using option bits located in flash memory
- Clock monitor supervising the correct function of the oscillator

#### 5.27.3.8 Serial Peripheral Interface Module (SPI)

- Configurable 8 or 16-bit data size
- Full-duplex or single-wire bidirectional
- Double-buffered transmit and receive
- Master or slave mode
- MSB-first or LSB-first shifting
- Serial clock phase and polarity options

#### 5.27.3.9 On-Chip Voltage Regulator (VREG)

- Linear voltage regulator with bandgap reference
- Low-voltage detect (LVD) with low-voltage interrupt (LVI)
- Power-on reset (POR) circuit
- Low-voltage reset (LVR)

#### 5.27.3.10 Background Debug (BDM)

- Non-intrusive memory access commands
- Supports in-circuit programming of on-chip nonvolatile memory

#### 5.27.3.11 Debugger (DBG)

- Trace buffer with depth of 64 entries
- Three comparators (A, B and C)
  - Comparator A compares the full address bus and full 16-bit data bus
  - Exact address or address range comparisons
- Two types of comparator matches
  - Tagged: This matches just before a specific instruction begins execution
  - Force: This is valid on the first instruction boundary after a match occurs
- Four trace modes
- Four stage state sequencer

#### 5.27.3.12 Die to Die Initiator (D2DI)

- Up to 2.0 Mbyte/s data rate
- Configurable 4-bit or 8-bit wide data path

Figure 43 shows 9S12I64 CPU and BDM local address translation to the global memory map. It indicates also the location of the internal resources in the memory map. The whole 256 k global memory space is visible through the P-Flash window located in the 64 k local memory map located at 0x8000 - 0xBFFF using the PPAGE register.

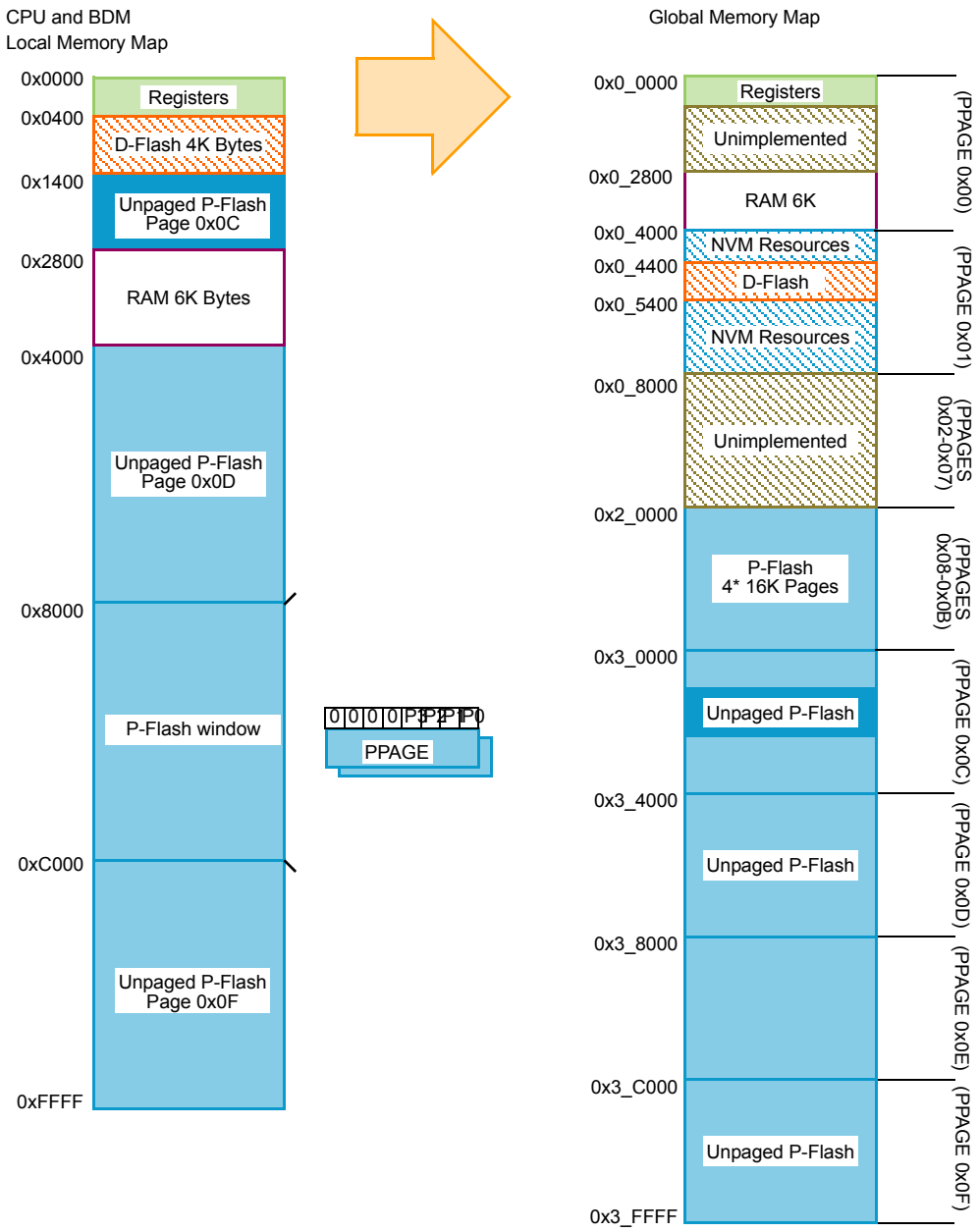



**Figure 43. 9S12I128 Global Memory Map**

### 5.27.4 Part ID Assignments

The part ID is located in two 8-bit registers PARTIDH and PARTIDL (addresses 0x001A and 0x001B). The read-only value is a unique part ID for each revision of the chip. Table 221 shows the assigned part ID number and Mask Set number.

The Version ID in Table 221 is a word located in a flash information row. The version ID number indicates a specific version of internal NVM controller.

**Table 221. Assigned Part ID Numbers**

| Device | Mask Set Number | Part ID[157] | Version ID |
|---|---|---|---|
| 9S12I64 | 0N53A | 0x38C0 | 0x0000 |

Note:
157. The coding is as follows:
Bit 15-12: Major family identifier
Bit 11-6: Minor family identifier
Bit 5-4: Major mask set revision number including FAB transfers
Bit 3-0: Minor — non full — mask set revision

### 5.27.5 System Clock Description

For the system clock description refer to Section 5.28, “Port Integration Module (S12IPIMV1)".

### 5.27.6 Modes of Operation

The MCU can operate in different modes. These are described in Section 5.27.6.1, “Chip Configuration Summary". The MCU can operate in different power modes to facilitate power saving when full system performance is not required. These are described in Section 5.27.6.2, “Low Power Operation". Some modules feature a software programmable option to freeze the module status whilst the background debug module is active to facilitate debugging.

#### 5.27.6.1 Chip Configuration Summary

The different modes and the security state of the MCU affect the debug features (enabled or disabled).

The operating mode out of reset is determined by the state of the MODC signal during reset (see Table 222). The MODC bit in the MODE register shows the current operating mode and provides limited mode switching during operation. The state of the MODC signal is latched into this bit on the rising edge of $\overline{\text{RESET}}$.

**Table 222. Chip Modes**

| Chip Modes | MODC |
|---|---|
| Normal single chip | 1 |
| Special single chip | 0 |

##### 5.27.6.1.1 Normal Single-Chip Mode

This mode is intended for normal device operation. The opcode from the on-chip memory is being executed after reset (requires the reset vector to be programmed correctly). The processor program is executed from internal memory.

##### 5.27.6.1.2 Special Single-Chip Mode

This mode is used for debugging single-chip operation, boot-strapping, or security related operations. The background debug module BDM is active in this mode. The CPU executes a monitor program located in an on-chip ROM. BDM firmware waits for additional serial commands through the BKGD pin.

#### 5.27.6.2 Low Power Operation

The MM912_634 has two static low-power modes Pseudo Stop and Stop Mode. For a detailed description refer to S12CPMU section.

### 5.27.7 Security

The MCU security mechanism prevents unauthorized access to the Flash memory. Refer to Section 5.28, “Port Integration Module (S12IPIMV1)", Section 5.31.4.1, “Security", and Section 5.40.5, “Security".

### 5.27.8 Resets and Interrupts

Consult the S12 CPU manual and the S12SINT section for information on exception processing.

#### 5.27.8.1 Resets

Table 223 lists all Reset sources and the vector locations. Resets are explained in detail in the Section 5.28, “Port Integration Module (S12IPIMV1)".

**Table 223. Reset Sources and Vector Locations**

| Vector Address | Reset Source | CCR Mask | Local Enable |
|---|---|---|---|
| $FFFE | Power-On Reset (POR) | None | None |
| $FFFE | Low Voltage Reset (LVR) | None | None |
| $FFFE | External pin $\overline{\text{RESET}}$ | None | None |
| $FFFE | Illegal Address Reset | None | None |
| $FFFC | Clock monitor reset | None | OSCE Bit in CPMUOSC register |
| $FFFA | COP watchdog reset | None | CR[2:0] in CPMUCOP register |

### 5.27.8.2 Interrupt Vectors

Table 224 lists all interrupt sources and vectors in the default order of priority. The interrupt module (see Section 5.28, “Port Integration Module (S12IPIMV1)") provides an interrupt vector base register (IVBR) to relocate the vectors.

**Table 224. Interrupt Vector Locations**

| Vector Address[158] | Interrupt Source | CCR Mask | Local Enable | Wake-up from STOP |
|---|---|---|---|---|
| Vector base + $F8 | Unimplemented instruction trap | None | None | - |
| Vector base+ $F6 | SWI | None | None | - |
| Vector base+ $F4 | D2DI Error Interrupt | X Bit | None | Yes |
| Vector base+ $F2 | D2DI External Error Interrupt | I bit | D2DCTL (D2DIE) | Yes |
| Vector base+ $F0 | RTI timeout interrupt | I bit | CPMUINT (RTIE) | |
| Vector base + $EE to Vector base + $DA | Reserved | | | |
| Vector base + $D8 | SPI | I bit | SPICR1 (SPIE, SPTIE) | No |
| Vector base + $D6 to Vector base + $CA | Reserved | | | |
| Vector base + $C8 | Oscillator status interrupt | I bit | CPMUINT (OSCIE) | No |
| Vector base + $C6 | PLL lock interrupt | I bit | CPMUINT (LOCKIE) | No |
| Vector base + $C4 to Vector base + $BC | Reserved | | | |
| Vector base + $BA | FLASH error | I bit | FERCNFG (SFDIE, DFDIE) | No |
| Vector base + $B8 | FLASH command | I bit | FCNFG (CCIE) | No |
| Vector base + $B6 to Vector base + $8C | Reserved | | | |
| Vector base + $8A | Low-voltage interrupt (LVI) | I bit | CPMUCTRL (LVIE) | No |
| Vector base + $88 to Vector base + $82 | Reserved | | | |
| Vector base + $80 | Spurious interrupt | — | None | - |

Note:
158. 16 bits vector address based

#### 5.27.8.3 Effects of Reset

When a reset occurs, MCU registers and control bits are initialized. Refer to the respective block sections for register reset states.

On each reset, the Flash module executes a reset sequence to load Flash configuration registers.

##### 5.27.8.3.1 Flash Configuration Reset Sequence Phase

On each reset, the Flash module holds CPU activity while loading Flash module registers from the Flash memory. If double faults are detected in the reset phase, Flash module protection and security may be active on leaving reset. This is explained in more detail in the Flash module Section 5.40.6, “Initialization".

##### 5.27.8.3.2 Reset While Flash Command Active

If a reset occurs while any Flash command is in progress, this command is immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed.

##### 5.27.8.3.3 I/O Pins

Refer to the PIM section for reset configurations of all peripheral module ports.

##### 5.27.8.3.4 Memory

The RAM arrays are not initialized out of reset.

### 5.27.9 COP Configuration

The COP time-out rate bits CR[2:0] and the WCOP bit in the CPMUCOP register at address 0x003C are loaded from the Flash register FOPT. See Table 225 and Table 226 for coding. The FOPT register is loaded from the Flash configuration field byte at global address 0x3_FF0E during the reset sequence.

**Table 225. Initial COP Rate Configuration**

| NV[2:0] in FOPT Register | CR[2:0] in COPCTL Register |
|---|---|
| 000 | 111 |
| 001 | 110 |
| 010 | 101 |
| 011 | 100 |
| 100 | 011 |
| 101 | 010 |
| 110 | 001 |
| 111 | 000 |

**Table 226. Initial WCOP Configuration**

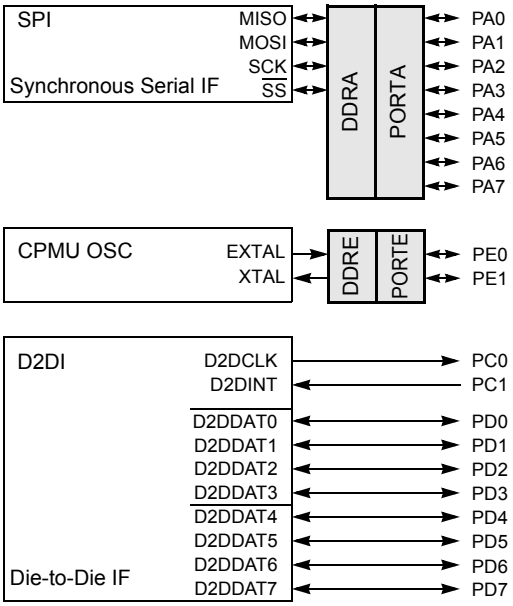
| NV[3] in FOPT Register | WCOP in COPCTL Register |
|---|---|
| 1 | 0 |
| 0 | 1 |

## 5.28 Port Integration Module (S12IPIMV1)

### 5.28.1 Introduction

The Port Integration Module (PIM) establishes the interface between the 9S12I128 peripheral modules SPI and Die-To-Die Interface module (D2DI) to the I/O pins of the MCU.

All port A and port E pins support general purpose I/O functionality if not in use with other functions. The PIM controls the signal prioritization and multiplexing on shared pins.




Port Integration Module Elements

**Figure 44. Block Diagram**

#### 5.28.1.1 Features

- 8-pin port A associated with the SPI module
- 2-pin port C used as D2DI clock output and D2DI interrupt input
- 8-pin port D used as 8 or 4 bit data I/O for the D2DI module
- 2-pin port E associated with the CPMU OSC module
- GPIO function shared on port A, E pins
- Pull-down devices on PC1 and PD7-0 if used as D2DI inputs
- Reduced drive capability on PC0 and PD7-0 on per pin basis

The Port Integration Module includes these distinctive registers:

- Data registers for ports A, E when used as general-purpose I/O
- Data direction registers for ports A, E when used as general-purpose I/O
- Port input register on ports A and E
- Reduced drive register on port C and D

### 5.28.2 Memory Map and Register Definition

This section provides a detailed description of all Port Integration Module registers.

#### 5.28.2.1 Memory Map

| Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0x0000 PORTA | R | PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| | W | | | | | | | | |
| 0x0001 PORTE | R | 0 | 0 | 0 | 0 | 0 | 0 | PE1 | PE0 |
| | W | | | | | | | | |
| 0x0002 DDRA | R | DDRA7 | DDRA6 | DDRA5 | DDRA4 | DDRA3 | DDRA2 | DDRA1 | DDRA0 |
| | W | | | | | | | | |
| 0x0003 DDRE | R | 0 | 0 | 0 | 0 | 0 | 0 | DDRE1 | DDRE0 |
| | W | | | | | | | | |
| 0x0004 PTC | R | 0 | 0 | 0 | 0 | 0 | 0 | PTC1 | PTC0 |
| | W | | | | | | | | |
| 0x0005 PTD | R | PTD7 | PTD6 | PTD5 | PTD4 | PTD3 | PTD2 | PTD1 | PTD0 |
| | W | | | | | | | | |
| 0x0006 DDRC | R | 0 | 0 | 0 | 0 | 0 | 0 | DDRC1 | DDRC0 |
| | W | | | | | | | | |
| 0x0007 DDRD | R | DDRD7 | DDRD6 | DDRD5 | DDRD4 | DDRD3 | DDRD2 | DDRD1 | DDRD0 |
| | W | | | | | | | | |
| 0x0009 - 0x0009 Reserved | | | | | | | | | |
| 0x000C PUCR | R | 0 | BKPUE | 0 | 0 | 0 | 0 | PDPEE | 0 |
| | W | | | | | | | | |
| 0x000D RDRIV | R | 0 | 0 | 0 | 0 | RDPD | RDPC | 0 | 0 |
| | W | | | | | | | | |
| 0x0120 PTIA | R | PTIA7 | PTIA6 | PTIA5 | PTIA4 | PTIA3 | PTIA2 | PTIA1 | PTIA0 |
| | W | | | | | | | | |
| 0x0121 PTIB | R | 0 | 0 | 0 | 0 | 0 | 0 | PTIE1 | PTIE0 |
| | W | | | | | | | | |
| 0x0122- 0x17F Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | W | | | | | | | | |

= Unimplemented or Reserved

### 5.28.2.2 Port A Data Register (PORTA)

**Table 227. Port A Data Register (PORTA)**

Address 0x0000 Access: User read/write[159]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| SPI Function | — | — | — | — | SS | SCK | MOSI | MISO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
159. Read: Anytime.
Write: Anytime.

**Table 228. PORTA Register Field Descriptions**

| Field | Description |
|---|---|
| 7–4<br>PA | **Port A general purpose input/output data**—Data Register In output mode the register bit is driven to the pin.<br>If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered and synchronized pin input state is read. |
| 3<br>PA | **Port A general purpose input/output data**—Data Register, SPI $\overline{SS}$ input/output<br>When not used with the alternative function, this pin can be used as general purpose I/O.<br>In general purpose output mode the register bit is driven to the pin.<br>If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.<br>• The SPI function takes precedence over the general purpose I/O function if enabled. |
| 2<br>PA | **Port A general purpose input/output data**—Data Register, SPI SCK input/output<br>When not used with the alternative function, this pin can be used as general purpose I/O.<br>In general purpose output mode the register bit is driven to the pin.<br>If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.<br>• The SPI function takes precedence over the general purpose I/O function if enabled. |
| 1<br>PA | **Port A general purpose input/output data**—Data Register, SPI MOSI input/output<br>When not used with the alternative function, this pin can be used as general purpose I/O.<br>In general purpose output mode the register bit is driven to the pin.<br>If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.<br>• The SPI function takes precedence over the general purpose I/O function if enabled. |
| 0<br>PA | **Port A general purpose input/output data**—Data Register, SPI MISO input/output<br>When not used with the alternative function, this pin can be used as general purpose I/O.<br>In general purpose output mode the register bit is driven to the pin.<br>If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.<br>• The SPI function takes precedence over the general purpose I/O function if enabled. |

### 5.28.2.3 Port E Data Register (PORTE)

**Table 229. Port E Data Register (PORTE)**

Address 0x0001 — Access: User read/write[160]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | 0 | 0 | PE1 | PE0 |
| W | | | | | | | | |
| CPMU OSC Function | — | — | — | — | — | — | XTAL | EXTAL |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
160. Read: Anytime.
Write: Anytime.

**Table 230. PORTE Register Field Descriptions**

| Field | Description |
|---|---|
| 1<br>PE | **Port E general purpose input/output data**—Data Register, CPMU OSC XTAL signal<br>When not used with the alternative function, this pin can be used as general purpose I/O.<br>In general purpose output mode the register bit is driven to the pin.<br>If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.<br>• The CPMU OSC function takes precedence over the general purpose I/O function if enabled. |
| 0<br>PE | **Port E general purpose input/output data**—Data Register, CPMU OSC EXTAL signal<br>When not used with the alternative function, this pin can be used as general purpose I/O.<br>In general purpose output mode the register bit is driven to the pin.<br>If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.<br>• The CPMU OSC function takes precedence over the general purpose I/O function if enabled. |

### 5.28.2.4 Port A Data Direction Register (DDRA)

**Table 231. Port A Data Direction Register (DDRA)**

Address 0x0002 — Access: User read/write[161]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | DDRA7 | DDRA6 | DDRA5 | DDRA4 | DDRA3 | DDRA2 | DDRA1 | DDRA0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
161. Read: Anytime.
Write: Anytime.

**Table 232. DDRA Register Field Descriptions**

| Field | Description |
|---|---|
| 7–4<br>DDRA | **Port A Data Direction**— This bit determines whether the associated pin is an input or output.<br>1 Associated pin is configured as output.<br>0 Associated pin is configured as input. |
| 3–0<br>DDRA | **Port A Data Direction**— This bit determines whether the associated pin is an input or output.<br>Depending on the configuration of the enabled SPI the I/O state is forced to input or output. In this case the data direction bits do not change.<br>1 Associated pin is configured as output.<br>0 Associated pin is configured as input. |

### 5.28.2.5 Port E Data Direction Register (DDRE)

**Table 233. Port E Data Direction Register (DDRE)**

Address 0x0003 Access: User read/write(162)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | 0 | 0 | DDRE1 | DDRE0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
162. Read: Anytime.
Write: Anytime.

**Table 234. DDRE Register Field Descriptions**

| Field | Description |
|---|---|
| 1–0<br>DDRE | **Port E Data Direction**— This bit determines whether the associated pin is an input or output. The enabled CPMU OSC function connects the associated pins directly to the oscillator module. In this case the data direction bits do not change.<br>1 Associated pin is configured as output.<br>0 Associated pin is configured as input. |

### 5.28.2.6 Port C Data Register (PTC)

**Table 235. Port C Data Register (PTC)**

Address 0x0004 Access: User read/write(163)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | 0 | 0 | PTC1 | PTC0 |
| W | | | | | | | | |
| D2DI Function | — | — | — | — | — | — | D2DINT | D2DCLK |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
163. Read: Anytime.
Write: Anytime.

**Table 236. PTC Register Field Descriptions**

| Field | Description |
|---|---|
| 1<br>PTC | **Port C general purpose input/output data**—Data Register<br>Port C pin 1 is associated with the D2DINT signal of the D2DI module.<br>When not used with the alternative function, this pin can be used as general purpose I/O.<br>It the associated data direction of this pin is set to 1, a read returns the value of the port register. Otherwise the buffered and synchronized pin in/out status is read. |
| 0<br>PTC | **Port C general purpose input/output data**—Data Register<br>Port C pin 0 is associated with the D2DCLK signal of the D2DI module.<br>When not used with the alternative function, this pin can be used as general purpose I/O.<br>It the associated data direction of this pin is set to 1, a read returns the value of the port register. Otherwise the buffered and synchronized pin in/out status is read. |

### 5.28.2.7 Port D Data Register (PTD)

**Table 237. Port D Data Register (PTD)**

Address 0x0005 Access: User read/write(164)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | PTD7 | PTD6 | PTD5 | PTD4 | PTD3 | PTD2 | PTD1 | PTD0 |
| D2DI Function | D2DDAT7 | D2DDAT6 | D2DDAT5 | D2DDAT4 | D2DDAT3 | D2DDAT2 | D2DDAT1 | D2DDAT0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
164. Read: Anytime.
Write: Anytime.

**Table 238. PTD Register Field Descriptions**

| Field | Description |
|---|---|
| 7-0<br>PTD | **Port D general purpose input/output data**—Data Register<br>Port D pins 7 through 0 are associated with the D2dI data signals of the D2DI module, if enabled in 8-bit mode.<br>Port D pins 3 through 0 are associated with the D2dI data signals of the D2DI module, if enabled in 4-bit mode.<br>When not used with the alternative function, this pin can be used as general purpose I/O.<br>It the associated data direction of this pin is set to 1, a read returns the value of the port register. Otherwise the buffered and synchronized pin in/out status is read. |

### 5.28.2.8 Port C Data Direction Register (DDRC)

**Table 239. Port C Data Register (PTC)**

Address 0x0006 Access: User read/write(165)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | 0 | 0 | DDRC1 | DDRC0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
165. Read: Anytime.
Write: Anytime.

**Table 240. DDRC Register Field Descriptions**

| Field | Description |
|---|---|
| 1-0<br>DDRC | **Port C Data Direction**<br>This register controls the data direction of pins 1 and 0.<br>The D2DI function controls the data direction for the associated pins. In this case, the data direction bits do not change.<br>When operating a pin as a general purpose I/O, the associated data direction bit determines whether it is an input or an output.<br>1. Associated pin is configured as an output.<br>0. Associated pin in configured as a high-impedance input. |

### 5.28.2.9 Port D Data Direction Register (DDRD)

**Table 241. Port D Data Direction Register (DDRD)**

Address 0x0007 Access: User read/write(166)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | DDRD7 | DDRD6 | DDRD5 | DDRD4 | DDRD3 | DDRD2 | DDRD1 | DDRD0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note:
166. Read: Anytime.
Write: Anytime.

**Table 242. PTD Register Field Descriptions**

| Field | Description |
|---|---|
| 7-0<br>DDRD | **Port D Data Direction**<br>This register controls the data direction of pins 7 through 0.<br>The D2DI function controls the data direction for the associated pins. In this case, the data direction bits do not change.<br>When operating a pin as a general purpose I/O, the associated data direction bit determines whether it is an input or an output.<br>1. Associated pin is configured as an output.<br>0. Associated pin in configured as a high-impedance input. |

### 5.28.2.10 PIM Reserved Registers

These registers are reserved for factory testing of the PIM module. Writing to these addresses can alter the module functionality.

**Table 243. PIM Reserved Registers**

Address 0x0007-0x0009 Access: User read(167)

| | 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

Note:
167. Read: Always reads 0x00
Write: Not allowed

### 5.28.2.11 Pull Control Register (PUCR)

**Table 244. Pull Control Register (PUCR)**

Address 0x0124 Access: User read/write[168]

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

Note:
168. Read: Anytime.
Write: Anytime.

**Table 245. PUCR Register Field Descriptions**

| Field | Description |
|---|---|
| 6<br>BKPUE | BKGD pin pull-up Enable—Enable pull-up devices on BKGD pin. This bit configures whether a pull-up device is activated, if the pin is used as input. This bit has no effect if the pin is used as output. Out of reset the pull-up device is enabled.<br>1 Pull-up device enabled.<br>0 Pull-up device disabled. |
| 1<br>PDPEE | Pull-down Port E Enable—Enable pull-down devices on all Port E input pins. This bit configures whether pull-down devices are activated, if the pins are used as inputs. This bit has no effect if the pins are used as outputs. Out of reset the pull-down devices are enabled. If the CPMU OSC function is active the pull-down devices are disabled. In this case, the register bit does not change.<br>1 Pull-down devices enabled.<br>0 Pull-down devices disabled. |

### 5.28.2.12 Reduced Drive Register (RDRIV)

**Table 246. Reduced Drive Register (RDRIV)**

Address 0x000D Access: User read/write[169]

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

Note:
169. Read: Anytime.
Write: Anytime.

**Table 247. RDRIV Register Field Descriptions**

| Field | Description |
|---|---|
| 3<br>RDPD | **Port D reduced drive—Select reduced drive for output pins.** This bit configures the drive strength of output pins as either full or reduced. If a pin is used as input this bit has no effect.<br>1 Reduced drive selected (1/5 of the full drive strength)<br>0 Full drive strength enabled |
| 2<br>RDPC | **Port C reduced drive—Select reduced drive for D2DCLK output pin.** This bit configures the drive strength of D2DCLK output pin as either full or reduced.<br>1 Reduced drive selected (1/5 of the full drive strength)<br>0 Full drive strength enabled |

### 5.28.2.13 Port A Input Register (PTIA)

**Table 248. Port A Input Register (PTIA)**

Address 0x0120 Access: User read[170]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | PTIA7 | PTIA6 | PTIA5 | PTIA4 | PTIA3 | PTIA2 | PTIA1 | PTIA0 |
| W | | | | | | | | |
| Reset[171] | u | u | u | u | u | u | u | u |

Note:
170. Read: Anytime.
Write: Unimplemented. Writing to this register has no effect.
171. u = Unaffected by reset

**Table 249. PTIA Register Field Descriptions**

| Field | Description |
|---|---|
| 7–0 PTIA | **Port A input data**— A read always returns the buffered input state of the associated pin.It can be used to detect overload or short-circuit conditions on output pins. |

### 5.28.2.14 Port E Input Register (PTIE)

**Table 250. Port E Input Register (PTIE)**

Address 0x0121 Access: User read[172]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | 0 | 0 | PTIE1 | PTIE0 |
| W | | | | | | | | |
| Reset[173] | u | u | u | u | u | u | u | u |

Note:
172. Read: Anytime.
Write: Unimplemented. Writing to this register has no effect.
173. u = Unaffected by reset

**Table 251. PTIE Register Field Descriptions**

| Field | Description |
|---|---|
| 1–0 PTIE | **Port E input data**— A read always returns the buffered input state of the associated pin.It can be used to detect overload or short-circuit conditions on output pins. |

### 5.28.2.15 PIM Reserved Registers

i

**Table 252. PIM Reserved Register**

Address 0x0122-0x017F — Access: User read[174]

| | 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 |

Note:
174. Read: Anytime.
Write: Unimplemented. Writing to this register has no effect.

## 5.28.3 Functional Description

### 5.28.3.1 Registers

#### 5.28.3.1.1 Data register (PORTx)

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 and synchronized 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 45).

#### 5.28.3.1.2 Data direction register (DDRx)

This register defines whether the pin is used as an input or an output. If a peripheral module controls the pin the contents of the data direction register is ignored (Figure 45).

#### 5.28.3.1.3 Input register (PTIx)

This is a read-only register and always returns the buffered and synchronized state of the pin (Figure 45).

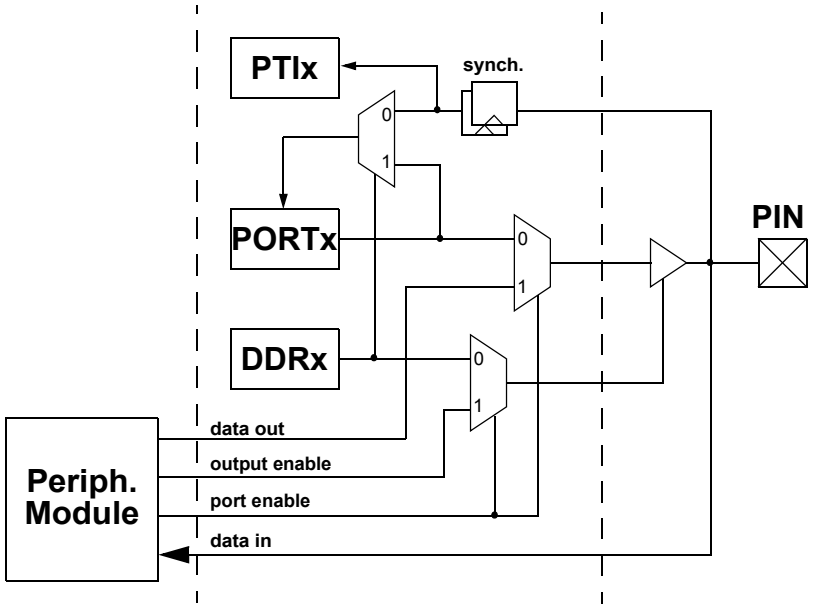



**Figure 45. Illustration of I/O Pin Functionality**

#### 5.28.3.1.4 Reduced Drive Register (RDRIV)

If the pin is used as an output this register allows the configuration of the drive strength.

#### 5.28.3.1.5 Pull Device Enable Register (PUCR)

This register turns on a pull-up or pull-down device. It becomes active only if the pin is used as an input.

### 5.28.3.2 Ports

#### 5.28.3.2.1 Port A

This port is associated with the SPI. Port A pins PA7-0 can be used for general-purpose I/O and PA3-0 also with the SPI subsystem.

#### 5.28.3.2.2 Port C

This port is associated with the D2DI interface. Port C pins PC1-0 can be used as the D2DI interrupt input and D2DI clock output, respectively. A pull-down device is enabled on pin PC1 if used as D2DI input.A reduced drive strength can be selected on PC0 if used as D2DI output. The D2DI interrupt input is synchronized and has an asynchronous bypass in STOP mode to allow the generation of a wake-up interrupt.

#### 5.28.3.2.3 Port D

This port is associated with the D2DI interface. Port D pins PD7-0 can be used with the D2DI data I/O. Pull-down devices are enabled on all pins if used as D2DI inputs.A reduced drive strength can be selected on all pins if used as D2DI outputs.

#### 5.28.3.2.4 Port E

This port is associated with the CPMU OSC. Port E pins PE1-0 can be used for general-purpose or with the CPMU OSC module.

## 5.28.4 Initialization Information

### 5.28.4.1 Port Data and Data Direction Register writes

It is not recommended to write PTx and DDRx in a word access. When changing the register pins from inputs to outputs, the data may have extra transitions during the write access. Initialize the port data register before enabling the outputs.

# 5.29 Memory Map Control (S12PMMCV1)

## 5.29.1 Introduction

The S12PMMC module controls the access to all internal memories and peripherals for the CPU12 and S12SBDM module. It regulates access priorities and determines the address mapping of the on-chip resources. Figure 46 shows a block diagram of the S12PMMC module.

#### 5.29.1.1 Glossary

**Table 253. Glossary Of Terms**

| Term | Definition |
|---|---|
| Local Addresses | Address within the CPU12's Local Address Map (Figure 51) |
| Global Address | Address within the Global Address Map (Figure 51) |
| Aligned Bus Access | Bus access to an even address. |
| Misaligned Bus Access | Bus access to an odd address. |
| NS | Normal Single-chip Mode |
| SS | Special Single-chip Mode |
| Unimplemented Address Ranges | Address ranges which are not mapped to any on-chip resource. |
| P-Flash | Program Flash |
| D-Plash | Data Flash |
| NVM | Non-volatile Memory; P-Flash or D-Flash |
| IFR | NVM Information Row. Refer to FTMRC Block Guide |

### 5.29.1.2 Overview

The S12PMMC connects the CPU12's and the S12SBDM's bus interfaces to the MCU's on-chip resources (memories and peripherals). It arbitrates the bus accesses and determines all of the MCU's memory maps. Furthermore, the S12PMMC is responsible for constraining memory accesses on secured devices and for selecting the MCU's functional mode.

### 5.29.1.3 Features

The main features of this block are:

- Paging capability to support a global 256 kByte memory address space
- Bus arbitration between the masters CPU12, S12SBDM to different resources.
- MCU operation mode control
- MCU security control
- Separate memory map schemes for each master CPU12, S12SBDM
- Generation of system reset when CPU12 accesses an unimplemented address (i.e., an address which does not belong to any of the on-chip modules) in single-chip modes

### 5.29.1.4 Modes of Operation

The S12PMMC selects the MCU's functional mode. It also determines the devices behavior in secured and unsecured state.

#### 5.29.1.4.1 Functional Modes

Two functional modes are implements on devices of the S12I product family:

- Normal Single Chip (NS)
  The mode used for running applications.
- Special Single Chip Mode (SS)
  A debug mode which causes the device to enter BDM Active Mode after each reset. Peripherals may also provide special debug features in this mode.

#### 5.29.1.4.2 Security

S12I derives can be secured to prohibit external access to the on-chip P-Flash. The S12PMMC module determines the access permissions to the on-chip memories in secured and unsecured state.

### 5.29.1.5 Block Diagram

Figure 46 shows a block diagram of the S12PMMC.

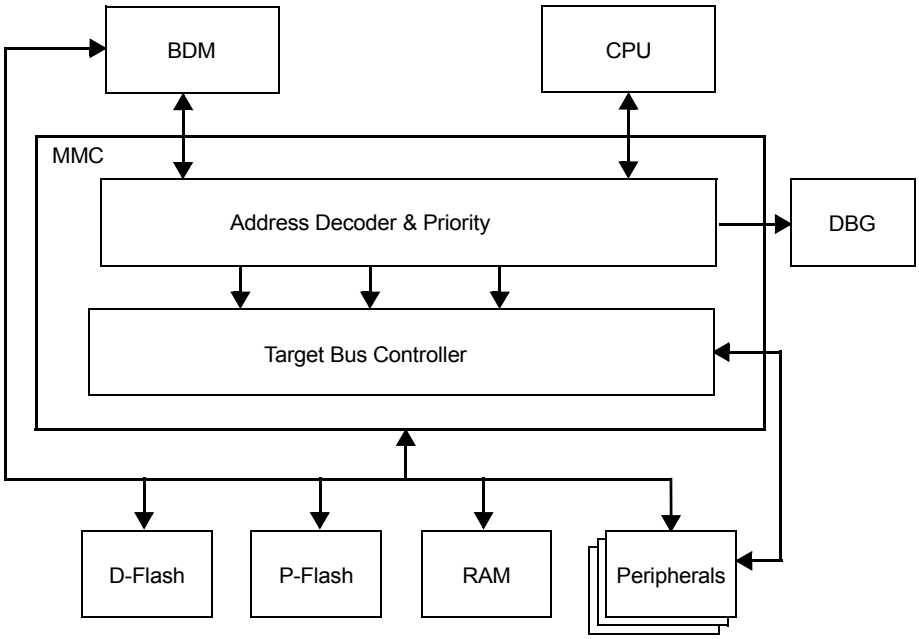



**Figure 46. S12PMMC Block Diagram**

### 5.29.2 External Signal Description

The S12PMMC uses two external pins to determine the devices operating mode: RESET and MODC (Table 254) See the User Guide for the mapping of these signals to device pins.

**Table 254. External System Pins Associated With S12PMMC**

| Pin Name | Pin Functions | Description |
|---|---|---|
| $\overline{\text{RESET}}$ (See User Guide) | $\overline{\text{RESET}}$ | The $\overline{\text{RESET}}$ pin is used the select the MCU's operating mode. |
| $\overline{\text{MODC}}$ (See User Guide) | MODC | The MODC pin is captured at the rising edge of the $\overline{\text{RESET}}$ pin. The captured value determines the MCU's operating mode. |

### 5.29.3 Memory Map and Registers

#### 5.29.3.1 Module Memory Map

A summary of the registers associated with the S12PMMC block is shown in Table 255. Detailed descriptions of the registers and bits are given in the subsections which follow.

**Table 255. MMC Register Summary**

| Address | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x000A | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x000B | MODE | R | MODC | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0010 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0011 | DIRECT | R | DP15 | DP14 | DP13 | DP12 | DP11 | DP10 | DP9 | DP8 |
| | | W | | | | | | | | |
| 0x0012 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0013 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0014 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0015 | PPAGE | R | 0 | 0 | 0 | 0 | PIX3 | PIX2 | PIX1 | PIX0 |
| | | W | | | | | | | | |
| | | | | = Unimplemented or Reserved | | | | | | |

### 5.29.3.2 Register Descriptions

This section consists of the S12PMMC control register descriptions in address order.

#### 5.29.3.2.1 Mode Register (MODE)

**Table 256. Mode Register (MODE)**

Address: 0x000B

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | MODC | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| W | | | | | | | | |
| Reset | MODC[175] | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Note:
175. External signal (see Table 254).

Read: Anytime.

Write: Only if a transition is allowed (see Figure 47).

The MODC bit of the MODE register is used to select the MCU's operating mode.

**Table 257. MODE Field Descriptions**

| Field | Description |
|---|---|
| 7<br>MODC | Mode Select Bit — This bit controls the current operating mode during RESET high (inactive). The external mode pin MODC determines the operating mode during RESET low (active). The state of the pin is registered into the respective register bit after the RESET signal goes inactive (see Figure 47).<br>Write restrictions exist to disallow transitions between certain modes. Figure 47 illustrates all allowed mode changes. Attempting non authorized transitions do not change the MODE bit, but it blocks further writes to the register bit except in special modes.<br>Write accesses to the MODE register are blocked when the device is secured. |

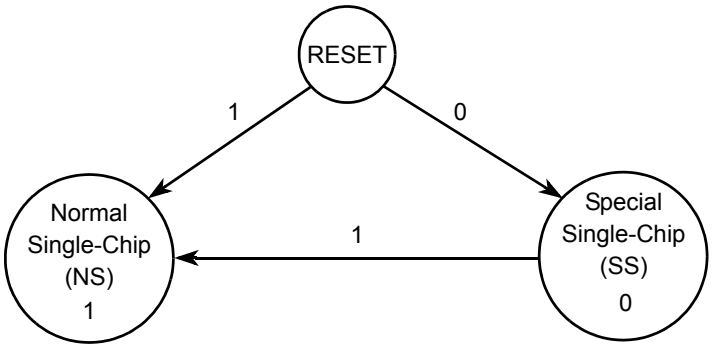



**Figure 47. Mode Transition Diagram When MCU is Unsecured**

#### 5.29.3.2.2 Direct Page Register (DIRECT)

**Table 258. Direct Register (DIRECT)**

Address: 0x0011

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | DP15 | DP14 | DP13 | DP12 | DP11 | DP10 | DP9 | DP8 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Read: Anytime

Write: anytime in special SS, write-one in NS.

This register determines the position of the 256 Byte direct page within the memory map.It is valid for both global and local mapping scheme.

**Table 259. DIRECT Field Descriptions**

| Field | Description |
|---|---|
| 7–0<br>DP[15:8] | Direct Page Index Bits 15–8 — These bits are used by the CPU when performing accesses using the direct addressing mode. These register bits form bits [15:8] of the local address (see Figure 48). |

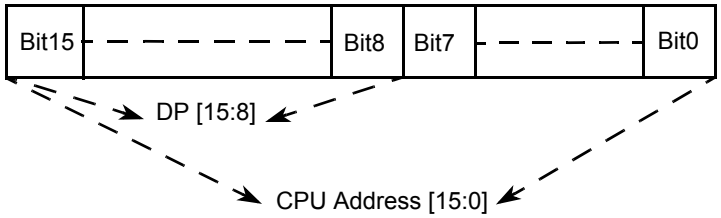



**Figure 48. DIRECT Address Mapping**

**Example 1. This example demonstrates usage of the Direct Addressing Mode**

```
MOVB    #$80,DIRECT     ;Set DIRECT register to 0x80. Write once only.
                        ;Global data accesses to the range 0xXX_80XX can be direct.
                        ;Logical data accesses to the range 0x80XX are direct.

LDY     <$00            ;Load the Y index register from 0x8000 (direct access).
                        ;< operator forces direct access on some assemblers but in
                        ;many cases assemblers are “direct page aware” and can
                        ;automatically select direct mode.
```

#### 5.29.3.2.3 Program Page Index Register (PPAGE)

**Table 260. Program Page Index Register (PPAGE)**

Address: 0x0015

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | PIX3 | PIX2 | PIX1 | PIX0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |

Read: Anytime

Write: Anytime

These four index bits are used to map 16 kB blocks into the Flash page window located in the local (CPU or BDM) memory map from address 0x8000 to address 0xBFFF (see Figure 49). This supports accessing up to 256 kB of Flash (in the Global map) within the 6 kB Local map. The PPAGE index register is effectively used to construct paged Flash addresses in the Local map format. The CPU has special access to read and write this register directly during execution of CALL and RTC instructions.

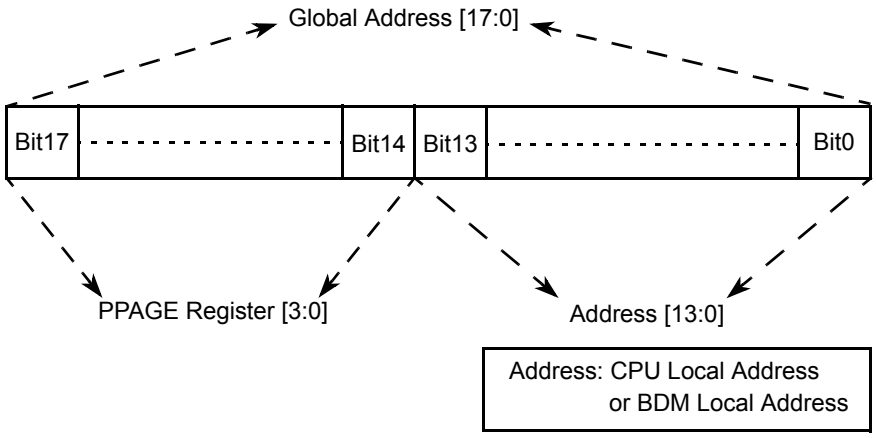



**Figure 49. PAGE Address Mapping**

**NOTE**

Writes to this register using the special access of the CALL and RTC instructions are complete before the end of the instruction execution.

**Table 261. PPAGE Field Descriptions**

| Field | Description |
|---|---|
| 3–0 PIX[3:0] | Program Page Index Bits 3–0 — These page index bits are used to select which of the 256 P-Flash or ROM array pages is to be accessed in the Program Page Window. |

The fixed 16 kB page from 0x0000 to 0x3FFF is the page number 0x0C. Parts of this page are covered by Registers, D-Flash and RAM space. See SoC Guide for details.

The fixed 16 kB page from 0x4000–0x7FFF is the page number 0x0D.

The reset value of 0x0E ensures there is linear Flash space available between addresses 0x0000 and 0xFFFF out of reset.

The fixed 16 kB page from 0xC000-0xFFFF is the page number 0x0F.

### 5.29.4 Functional Description

The S12PMMC block performs several basic functions of the S12I 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.

#### 5.29.4.1 MCU Operating Modes

- Normal single chip mode
  This is the operation mode for running application code. There is no external bus in this mode.
- Special single chip mode
  This mode is generally used for debugging 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.

#### 5.29.4.2 Memory Map Scheme

##### 5.29.4.2.1 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 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 0x3_FF00 - 0x3_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 is not visible in the memory map during active BDM mode.

Note: after the MCU enters active BDM mode the BDM firmware lookup tables and the BDM registers is also visible between addresses 0xBF00 and 0xBFFF if the PPAGE register contains value of 0x0F.

#### 5.29.4.2.1.1 Expansion of the Local Address Map

**Expansion of the CPU Local Address Map**

The program page index register in S12PMMC allows accessing up to 256 kB of P-Flash in the global memory map by using the four index bits (PPAGE[3:0]) to page 16x16 kB blocks into the program page window located from address 0x8000 to address 0xBFFF in the local CPU memory map.

The page value for the program page window is stored in the PPAGE register. The value of the PPAGE register can be read or written by normal memory accesses as well as by the CALL and RTC instructions (see Section 5.29.6.1, “CALL and RTC Instructions").

Control registers, vector space and parts of the on-chip memories are located in unpaged portions of the 64 kB local CPU address space.

The starting address of an interrupt service routine must be located in unpaged memory unless the user is certain the PPAGE register is set to the appropriate value when the service routine is called. However an interrupt service routine can call other routines in paged memory. The upper 16 kB block of the local CPU memory space (0xC000–0xFFFF) is unpaged. It is recommended all reset and interrupt vectors point to locations in this area or to the other unmapped pages sections of the local CPU memory map.

**Expansion of the BDM Local Address Map**

PPAGE and BDMPPR register is also used for the expansion of the BDM local address to the global address. These registers can be read and written by the BDM.

The BDM expansion scheme is the same as the CPU expansion scheme.

The four BDMPPR Program Page index bits allow access to the full 256 kB address map which can be accessed with 18 address bits.

The BDM program page index register (BDMPPR) is used only when the feature is enabled in BDM and, in the case the CPU is executing a firmware command which uses CPU instructions, or by a BDM hardware commands. Refer to Section 5.31, “Background Debug Module (S12SBDMV1)" for further details. (see Figure 50).

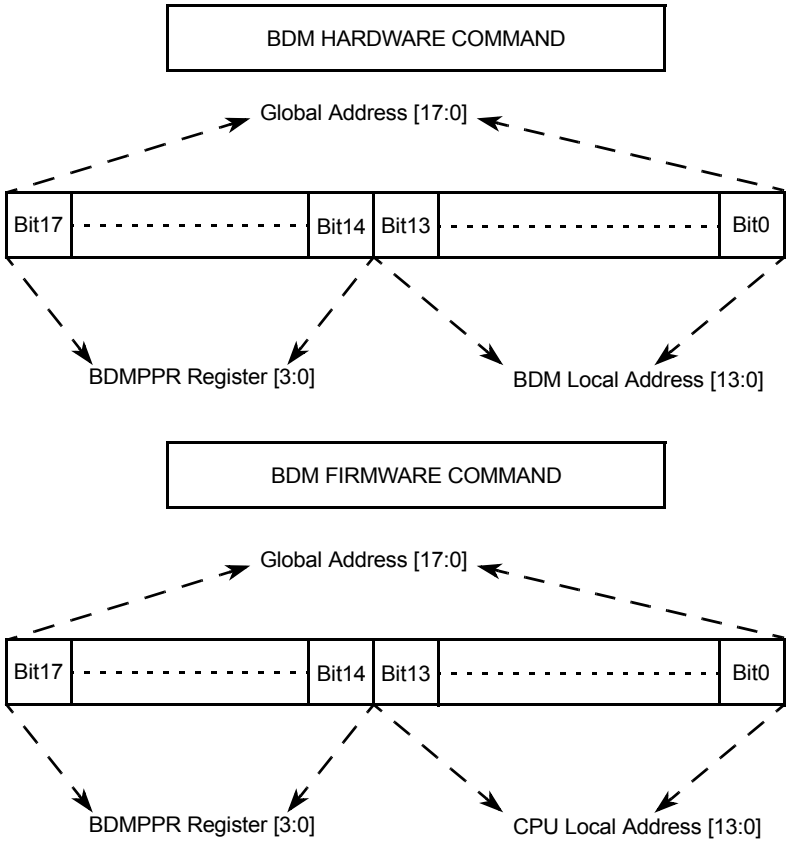



**Figure 50. BDMPPR Address Mapping**

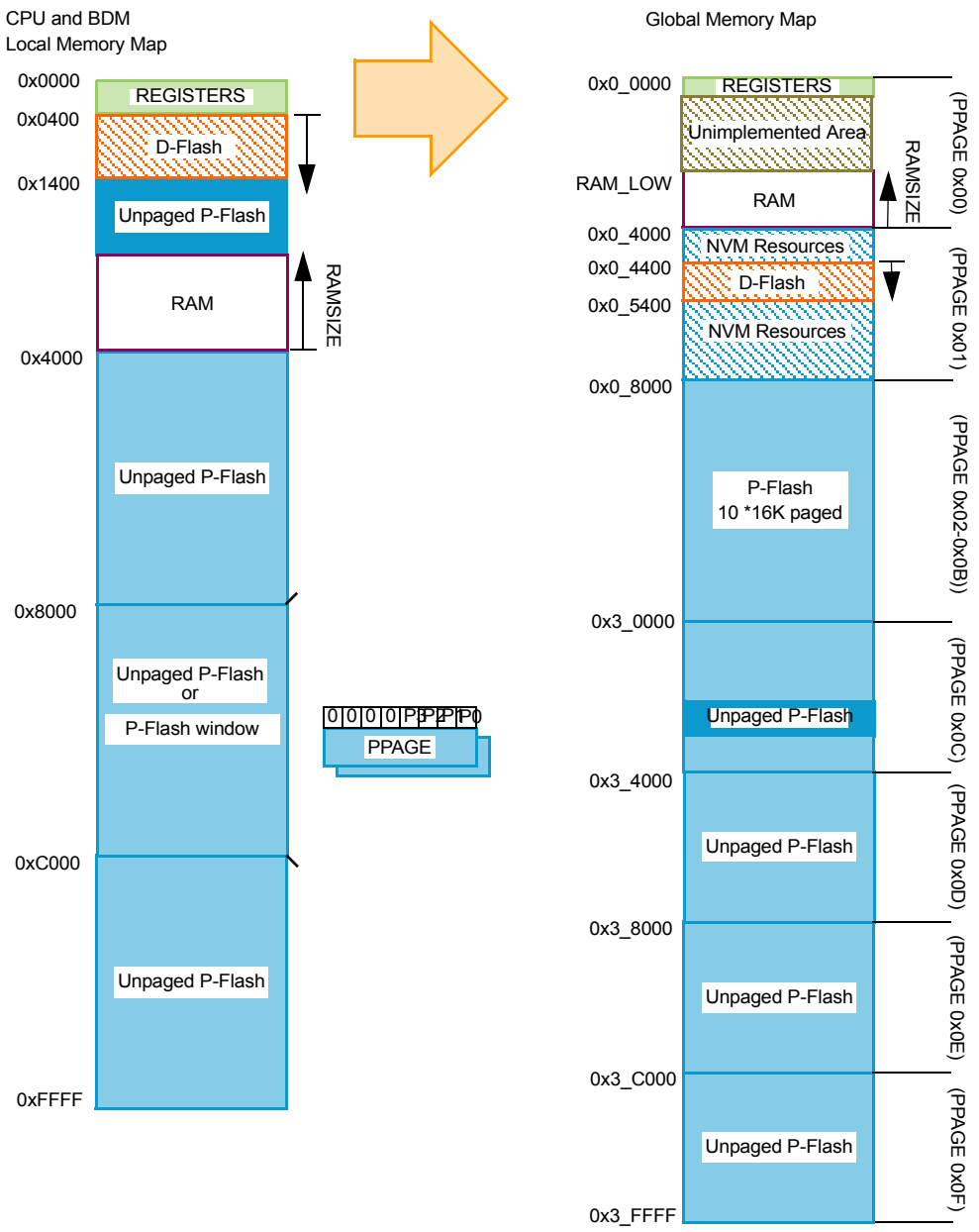



**Figure 51. Local to Global Address Mapping**

### 5.29.5 Implemented Memory in the System Memory Architecture

Each memory can be implemented in its maximum allowed size. But some devices have been defined for smaller sizes, which means less implemented pages. All non implemented pages are called unimplemented areas.

- Registers has a fixed size of 1.0 kB, accessible via xbus0.
- SRAM has a maximum size of 11 kB, accessible via xbus0.
- D-Flash has a fixed size of 4.0 kB accessible via xbus0.
- P-Flash has a maximum size of 224 kB, accessible via xbus0.
- NVM resources (IFR, Scratch-RAM, ROM) including D-Flash have maximum size of 16 kB (PPAGE 0x01).

#### 5.29.5.0.1 Implemented Memory Map

The global memory spaces reserved for the internal resources (RAM, D-Flash, and P-Flash) are not determined by the MMC module. Size of the individual internal resources are however fixed in the design of the device cannot be changed by the user. Refer to the SoC Guide for further details. Figure 52 and Table 262 show the memory spaces occupied by the on-chip resources. Note: the memory spaces have fixed top addresses.

**Table 262. Global Implemented Memory Space**

| Internal Resource | Bottom Address | Top Address |
|---|---|---|
| Registers | 0x0_0000 | 0x0_03FF |
| System RAM | RAM_LOW = 0x0_4000 minus RAMSIZE(176) | 0x0_3FFF |
| D-Flash | 0x0_4400 | 0x0_53FF |
| P-Flash | PF_LOW = 0x4_0000 minus FLASHSIZE(177) | 0x3_FFFF |

Note:
176. RAMSIZE is the hexadecimal value of RAM SIZE in bytes
177. FLASHSIZE is the hexadecimal value of FLASH SIZE in bytes

In single-chip modes accesses by the CPU12 (except for firmware commands) to any of the unimplemented areas (see Figure 52) result in an illegal access reset (system reset). BDM accesses to the unimplemented areas are allowed but the data are undefined.

No misaligned word access from the BDM module occurs; these accesses are blocked in the BDM module (Refer to Section 5.31, “Background Debug Module (S12SBDMV1)").

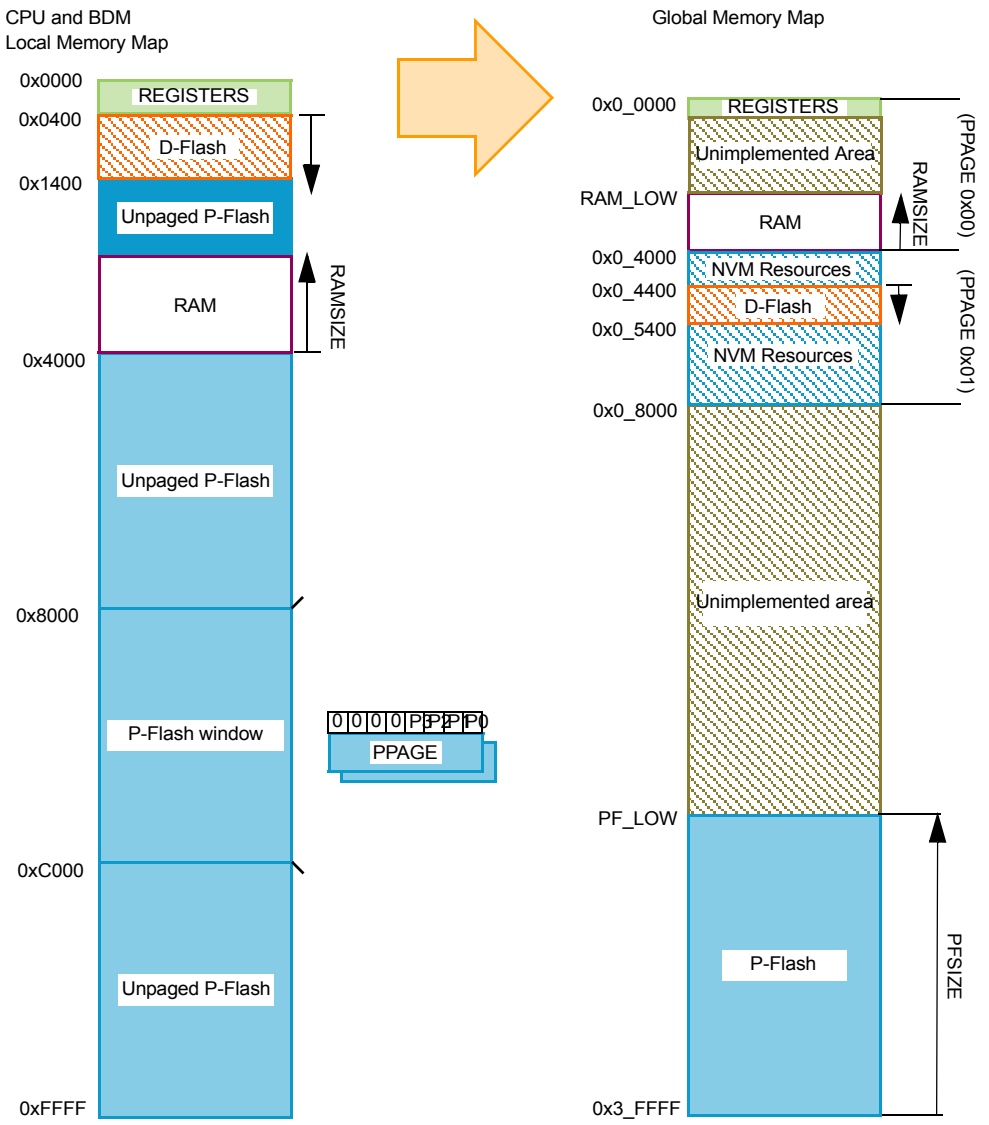



**Figure 52. Implemented Global Address Mapping**

### 5.29.5.1 Chip Bus Control

The S12PMMC controls the address buses and the data buses which interface the bus masters (CPU12, S12SBDM) with the rest of the system (master buses). In addition the MMC handles all CPU read data bus swapping operations. All internal resources are connected to specific target buses (see Figure 53).

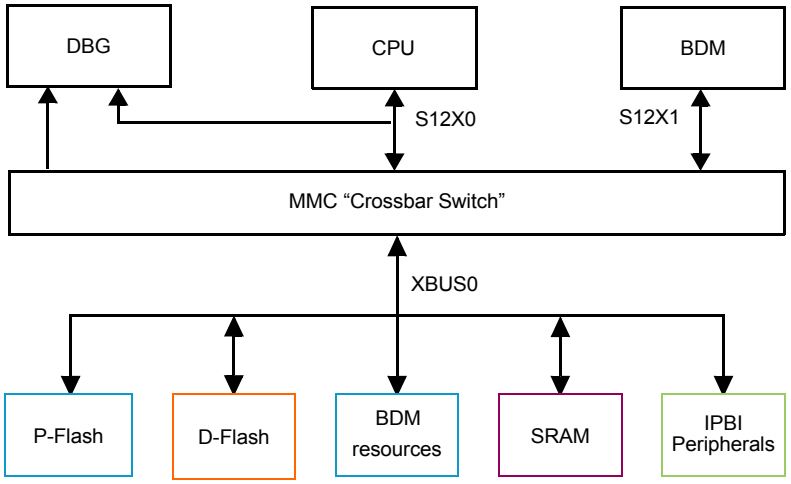



**Figure 53. S12I Platform**

#### 5.29.5.1.1 Master Bus Prioritization regarding Access Conflicts on Target Buses

The arbitration scheme allows only one master to be connected to a target at any given time. The following rules apply when prioritizing accesses from different masters to the same target bus:

- CPU12 always has priority over BDM.
- BDM has priority over CPU12 when its access is stalled for more than 128 cycles. In the later case, the CPU is stalled after finishing the current operation and the BDM gains access to the bus.

### 5.29.5.2 Interrupts

The MMC does not generate any interrupts.

## 5.29.6 Initialization/Application Information

### 5.29.6.1 CALL and RTC Instructions

CALL and RTC instructions are uninterruptable CPU instructions which automate page switching in the program page window. The CALL instruction is similar to the JSR instruction, but the subroutine which is called can be located anywhere in the local address space or in any Flash or ROM page visible through the program page window. The CALL instruction calculates and stacks a return address, stacks the current PPAGE value and writes a new instruction-supplied value to the PPAGE register. The PPAGE value controls which of the 256 possible pages is visible through the 16 kbyte program page window in the 64 kbyte local CPU memory map. Execution then begins at the address of the called subroutine.

During the execution of the CALL instruction, the CPU performs the following steps:

1. Writes the current PPAGE value into an internal temporary register and writes the new instruction-supplied PPAGE value into the PPAGE register
2. Calculates the address of the next instruction after the CALL instruction (the return address) and pushes this 16-bit value onto the stack
3. Pushes the temporarily stored PPAGE value onto the stack
4. Calculates the effective address of the subroutine, refills the queue and begins execution at the new address

This sequence is uninterruptable. There is no need to inhibit interrupts during the CALL instruction execution. A CALL instruction can be performed from any address to any other address in the local CPU memory space.

The PPAGE value supplied by the instruction is part of the effective address of the CPU. For all addressing mode variations (except indexed-indirect modes) the new page value is provided by an immediate operand in the instruction. In indexed-indirect variations of the CALL instruction a pointer specifies memory locations where the new page value and the address of the called subroutine are stored. Using indirect addressing for both the new page value and the address within the page allows usage of values calculated at run time rather than immediate values which must be known at the time of assembly.

## 5.30 Interrupt Module (S12SINTV1)

### 5.30.1 Introduction

The INT module decodes the priority of all system exception requests and provides the applicable vector for processing the exception to the CPU. The INT module supports:

- I bit and X bit maskable interrupt requests
- A non-maskable unimplemented op-code trap
- A non-maskable software interrupt (SWI) or background debug mode request
- Three system reset vector requests
- A spurious interrupt vector

Each of the I bit maskable interrupt requests is assigned to a fixed priority level.

#### 5.30.1.1 Glossary

Table 263 contains terms and abbreviations used in the document.

**Table 263. Terminology**

| Term | Meaning |
|---|---|
| CCR | Condition Code Register (in the CPU) |
| ISR | Interrupt Service Routine |
| MCU | Micro-Controller Unit |

#### 5.30.1.2 Features

- Interrupt vector base register (IVBR)
- One spurious interrupt vector (at address vector base[178] + 0x0080).
- 2–58 I bit maskable interrupt vector requests (at addresses vector base + 0x0082–0x00F2).
- I bit maskable interrupts can be nested.
- One X bit maskable interrupt vector request (at address vector base + 0x00F4).
- One non-maskable software interrupt request (SWI) or background debug mode vector request (at address vector base + 0x00F6).
- One non-maskable unimplemented op-code trap (TRAP) vector (at address vector base + 0x00F8).
- Three system reset vectors (at addresses 0xFFFA–0xFFFE).
- Determines the highest priority interrupt vector requests, drives the vector to the bus on CPU request
- Wakes up the system from stop mode when an appropriate interrupt request occurs.

Note:

178. The vector base is a 16-bit address which is accumulated from the contents of the interrupt vector base register (IVBR, used as upper byte) and 0x00 (used as lower byte).

#### 5.30.1.3 Modes of Operation

- Run mode
  This is the basic mode of operation.
- Stop Mode
  In stop mode, the clock to the INT module is disabled. The INT module is however capable of waking-up the CPU from stop mode if an interrupt occurs. Refer to Section 5.30.5.3, “Wake-up from Stop Mode" for details.
- Freeze mode (BDM active)
  In freeze mode (BDM active), the interrupt vector base register is overridden internally. Refer to Section 5.30.3.1.1, “Interrupt Vector Base Register (IVBR)" for details.

### 5.30.1.4 Block Diagram

Figure 54 shows a block diagram of the S12SBDMV1 module.

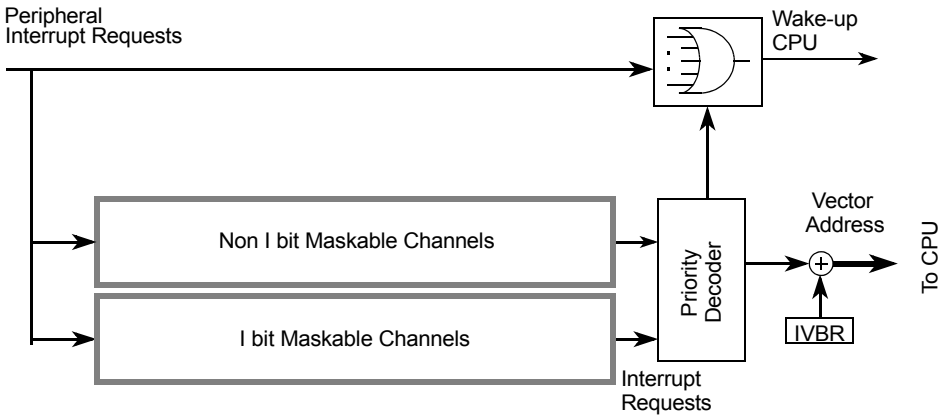



**Figure 54. S12SBDMV1 Block Diagram**

### 5.30.2 External Signal Description

The S12SBDMV1 module has no external signals.

### 5.30.3 Memory Map and Register Definition

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

#### 5.30.3.1 Register Descriptions

This section describes in address order all the INT registers and their individual bits.

##### 5.30.3.1.1 Interrupt Vector Base Register (IVBR)

**Table 264. Interrupt Vector Base Register (IVBR)**

Address: 0x001F

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | IVB_ADDR[7:0] | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Read: Anytime

Write: Anytime

**Table 265. IVBR Field Descriptions**

| Field | Description |
|---|---|
| 7–0<br>IVB_ADDR[7:0] | Interrupt Vector Base Address Bits — These bits represent the upper byte of all vector addresses. Out of reset these bits are set to 0xFF (i.e., vectors are located at 0xFF80–0xFFFE) to ensure compatibility to HCS12.<br>Note: A system reset initializes the interrupt vector base register with “0xFF” before it is used to determine the reset vector address. Therefore, changing the IVBR has no effect on the location of the three reset vectors (0xFFFA–0xFFFE).<br>Note: If the BDM is active (i.e., the CPU is in the process of executing BDM firmware code), the contents of IVBR are ignored and the upper byte of the vector address is fixed as “0xFF”. This is done to enable handling of all non-maskable interrupts in the BDM firmware. |

### 5.30.4 Functional Description

The INT module processes all exception requests to be serviced by the CPU module. These exceptions include interrupt vector requests and reset vector requests. Each of these exception types and their overall priority level is discussed in the subsections below.

#### 5.30.4.1 S12S Exception Requests

The CPU handles both reset requests and interrupt requests. A priority decoder is used to evaluate the priority of pending interrupt requests.

#### 5.30.4.2 Interrupt Prioritization

The INT module contains a priority decoder to determine the priority for all interrupt requests pending for the CPU. If more than one interrupt request is pending, the interrupt request with the higher vector address wins the prioritization.

The following conditions must be met for an I bit maskable interrupt request to be processed.

1. The local interrupt enabled bit in the peripheral module must be set.
2. The I bit in the condition code register (CCR) of the CPU must be cleared.
3. There is no SWI, TRAP, or X bit maskable request pending.

**NOTE**

All non I bit maskable interrupt requests always have higher priority than the I bit maskable interrupt requests. If the X bit in the CCR is cleared, it is possible to interrupt an I bit maskable interrupt by an X bit maskable interrupt. It is possible to nest non maskable interrupt requests, e.g., by nesting SWI or TRAP calls.

Since an interrupt vector is only supplied at the time when the CPU requests it, it is possible a higher priority interrupt request could override the original interrupt request which caused the CPU to request the vector. In this case, the CPU receives the highest priority vector and the system processes this interrupt request first, before the original interrupt request is processed.

If the interrupt source is unknown (for example, in the case where an interrupt request becomes inactive after the interrupt has been recognized, but prior to the CPU vector request), the vector address supplied to the CPU defaults to the spurious interrupt vector.

**NOTE**

Care must be taken to ensure all interrupt requests remain active until the system begins execution of the applicable service routine; otherwise, the exception request may not get processed at all or the result may be a spurious interrupt request (vector at address (vector base + 0x0080)).

#### 5.30.4.3 Reset Exception Requests

The INT module supports three system reset exception request types (refer to the Clock and Reset generator module for details):

1. Pin reset, power-on reset or illegal address reset, low voltage reset (if applicable)
2. Clock monitor reset request
3. COP watchdog reset request

#### 5.30.4.4 Exception Priority

The priority (from highest to lowest) and address of all exception vectors issued by the INT module upon request by the CPU is shown in Table 266.

**Table 266. Exception Vector Map and Priority**

| Vector Address[179] | Source |
|---|---|
| 0xFFFE | Pin reset, power-on reset, illegal address reset, low voltage reset (if applicable) |
| 0xFFFC | Clock monitor reset |
| 0xFFFA | COP watchdog reset |
| (Vector base + 0x00F8) | Unimplemented opcode trap |
| (Vector base + 0x00F6) | Software interrupt instruction (SWI) or BDM vector request |
| (Vector base + 0x00F4) | X bit maskable interrupt request (XIRQ or D2D error interrupt)[180] |
| (Vector base + 0x00F2) | IRQ or D2D interrupt request[181] |

**Table 266. Exception Vector Map and Priority (continued)**

| Vector Address[179] | Source |
|---|---|
| (Vector base + 0x00F0–0x0082) | Device specific I bit maskable interrupt sources (priority determined by the low byte of the vector address, in descending order) |
| (Vector base + 0x0080) | Spurious interrupt |

Note:
179. 16 bits vector address based
180. D2D error interrupt on MCUs featuring a D2D initiator module, otherwise XIRQ pin interrupt
181. D2D interrupt on MCUs featuring a D2D initiator module, otherwise IRQ pin interrupt

## 5.30.5 Initialization/Application Information

### 5.30.5.1 Initialization

After system reset, software should:

1. Initialize the interrupt vector base register if the interrupt vector table is not located at the default location (0xFF80–0xFFF9).
2. Enable I bit maskable interrupts by clearing the I bit in the CCR.
3. Enable the X bit maskable interrupt by clearing the X bit in the CCR.

### 5.30.5.2 Interrupt Nesting

The interrupt request scheme makes it possible to nest I bit maskable interrupt requests handled by the CPU. I bit maskable interrupt requests can be interrupted by an interrupt request with a higher priority.

I bit maskable interrupt requests cannot be interrupted by other I bit maskable interrupt requests per default. In order to make an interrupt service routine (ISR) interruptible, the ISR must explicitly clear the I bit in the CCR (CLI). After clearing the I bit, other I bit maskable interrupt requests can interrupt the current ISR.

An ISR of an interruptible I bit maskable interrupt request could basically look like this:

1. Service interrupt, e.g., clear interrupt flags, copy data, etc.
2. Clear I bit in the CCR by executing the instruction CLI (thus allowing other I bit maskable interrupt requests)
3. Process data
4. Return from interrupt by executing the instruction RTI

### 5.30.5.3 Wake-up from Stop Mode

#### 5.30.5.3.1 CPU Wake-up from Stop Mode

Every I bit maskable interrupt request is capable of waking the MCU from stop mode. To determine whether an I bit maskable interrupts is qualified to wake-up the CPU or not, the same conditions as in normal run mode are applied during stop mode: If the I bit in the CCR is set, all I bit maskable interrupts are masked from waking-up the MCU.

Since there are no clocks running in stop mode, only interrupts which can be asserted asynchronously can wake-up the MCU from stop mode.

The X bit maskable interrupt request can wake up the MCU from stop mode at anytime, even if the X bit in CCR is set.

If the X bit maskable interrupt request is used to wake-up the MCU with the X bit in the CCR set, the associated ISR is not called. The CPU then resumes program execution with the instruction following the WAI or STOP instruction. This features works the same rules like any interrupt request, i.e. care must be taken so the X interrupt request used for wake-up remains active at least until the system begins execution of the instruction following the WAI or STOP instruction; otherwise, wake-up may not occur.

## 5.31 Background Debug Module (S12SBDMV1)

### 5.31.1 Introduction

This section describes the functionality of the background debug module (BDM) sub-block of the HCS12S core platform.

The background debug module (BDM) sub-block is a single-wire, background debug system implemented in on-chip hardware for minimal CPU intervention. All interfacing with the BDM is done via the BKGD pin.

The BDM has enhanced capability for maintaining synchronization between the target and host while allowing more flexibility in clock rates. This includes a sync signal to determine the communication rate and a handshake signal to indicate when an operation is complete. The system is backwards compatible to the BDM of the S12 family with the following exceptions:

- TAGGO command not supported by S12SBDM
- External instruction tagging feature is part of the DBG module
- S12SBDM register map and register content modified
- Family ID readable from BDM ROM at global address 0x3_FF0F in active BDM (value for devices with HCS12S core is 0xC2)
- Clock switch removed from BDM (CLKSW bit removed from BDMSTS register)

#### 5.31.1.1 Features

The BDM includes these distinctive features:

- Single-wire communication with host development system
- Enhanced capability for allowing more flexibility in clock rates
- SYNC command to determine communication rate
- GO_UNTIL command
- Hardware handshake protocol to increase the performance of the serial communication
- Active out of reset in special single chip mode
- Nine hardware commands using free cycles, if available, for minimal CPU intervention
- Hardware commands not requiring active BDM
- 14 firmware commands execute from the standard BDM firmware lookup table
- When secured, hardware commands are allowed to access the register space in special single chip mode, if the Flash erase tests fail.
- Family ID readable from BDM ROM at global address 0x3_FF0F in active BDM (value for devices with HCS12S core is 0xC2)
- BDM hardware commands are operational until system stop mode is entered

#### 5.31.1.2 Modes of Operation

BDM is available in all operating modes but must be enabled before firmware commands are executed. Some systems may have a control bit which allow suspending the function during background debug mode.

##### 5.31.1.2.1 Regular Run Modes

All of these operations refer to the part in run mode and not being secured. The BDM does not provide controls to conserve power during run mode.

- Normal modes
  General operation of the BDM is available and operates the same in all normal modes.
- Special single chip mode
  In special single chip mode, background operation is enabled and active out of reset.This allows programming a system with blank memory.

##### 5.31.1.2.2 Secure Mode Operation

If the device is in secure mode, the operation of the BDM is reduced to a small subset of its regular run mode operation. Secure operation prevents access to Flash other than allowing erasure. For more information see Section 5.31.4.1, “Security".

##### 5.31.1.2.3 Low-power Modes

The BDM can be used until stop mode is entered. The CPU cannot enter stop mode during BDM active mode.

In stop mode the BDM clocks are stopped. When BDM clocks are disabled and stop mode is exited, the BDM clocks restarts and BDM has a soft reset (clearing the instruction register, any command in progress and disable the ACK function). The BDM is now ready to receive a new command.

### 5.31.1.3 Block Diagram

A block diagram of the BDM is shown in Figure 55.

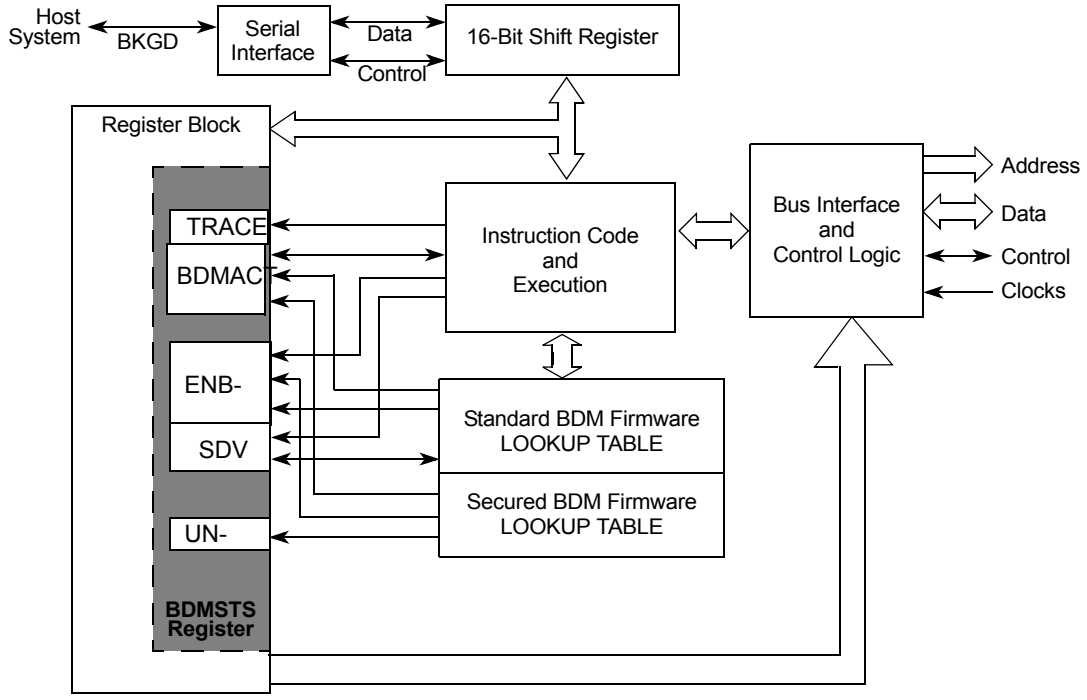



**Figure 55. BDM Block Diagram**

## 5.31.2 External Signal Description

A single-wire interface pin called the background debug interface (BKGD) pin is used to communicate with the BDM system. During reset, this pin is a mode select input which selects between normal and special modes of operation. After reset, this pin becomes the dedicated serial interface pin for the background debug mode.

## 5.31.3 Memory Map and Register Definition

### 5.31.3.1 Module Memory Map

Table 267 shows the BDM memory map when BDM is active.

**Table 267. BDM Memory Map**

| Global Address | Module | Size (Bytes) |
|---|---|---|
| 0x3_FF00–0x3_FF0B | BDM registers | 12 |
| 0x3_FF0C–0x3_FF0E | BDM firmware ROM | 3 |
| 0x3_FF0F | Family ID (part of BDM firmware ROM) | 1 |
| 0x3_FF10–0x3_FFFF | BDM firmware ROM | 240 |

### 5.31.3.2 Register Descriptions

A summary of the registers associated with the BDM is shown in Table 268. Registers are accessed by host-driven communications to the BDM hardware using READ_BD and WRITE_BD commands.

**Table 268. BDM Register Summary**

| Global Address | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x3_FF00 | Reserved | R | X | X | X | X | X | X | 0 | 0 |
| | | W | | | | | | | | |
| 0x3_FF01 | BDMSTS | R | ENBDM | BDMACT | 0 | SDV | TRACE | 0 | UNSEC | 0 |
| | | W | | | | | | | | |
| 0x3_FF02 | Reserved | R | X | X | X | X | X | X | X | X |
| | | W | | | | | | | | |
| 0x3_FF03 | Reserved | R | X | X | X | X | X | X | X | X |
| | | W | | | | | | | | |
| 0x3_FF04 | Reserved | R | X | X | X | X | X | X | X | X |
| | | W | | | | | | | | |
| 0x3_FF05 | Reserved | R | X | X | X | X | X | X | X | X |
| | | W | | | | | | | | |
| 0x3_FF06 | BDMCCR | R | CCR7 | CCR6 | CCR5 | CCR4 | CCR3 | CCR2 | CCR1 | CCR0 |
| | | W | | | | | | | | |
| 0x3_FF07 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x3_FF08 | BDMPPR | R | BPAE | 0 | 0 | 0 | BPP3 | BPP2 | BPP1 | BPP0 |
| | | W | | | | | | | | |
| 0x3_FF09 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x3_FF0A | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x3_FF0B | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

| | | | |
|---|---|---|---|
| | = Unimplemented, Reserved | | = Implemented (do not alter) |
| X | = Indeterminate | 0 | = Always read zero |

#### 5.31.3.2.1 BDM Status Register (BDMSTS)

Register Global Address 0x3_FF01

**Table 269. BDM Status Register (BDMSTS)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | ENBDM | BDMACT | 0 | SDV | TRACE | 0 | UNSEC | 0 |
| W | | | | | | | | |
| Reset | | | | | | | | |
| Special Single-Chip Mode | 0[182] | 1 | 0 | 0 | 0 | 0 | 0[183] | 0 |
| All Other Modes | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented, Reserved

= Implemented (do not alter)

0 = Always read zero

Note:

182. ENBDM is read as 1 by a debugging environment in special single chip mode when the device is not secured or secured but fully erased (Flash). This is because the ENBDM bit is set by the standard BDM firmware before a BDM command can be fully transmitted and executed.

183. UNSEC is read as 1 by a debugging environment in special single chip mode when the device is secured and fully erased, else it is 0 and can only be read if not secure (see also bit description).

Read: All modes through BDM operation when not secured

Write: All modes through BDM operation when not secured, but subject to the following:

- ENBDM should only be set via a BDM hardware command if the BDM firmware commands are needed. (This does not apply in special single chip mode).
- BDMACT can only be set by BDM hardware upon entry into BDM. It can only be cleared by the standard BDM firmware lookup table upon exit from BDM active mode.
- All other bits, while writable via BDM hardware or standard BDM firmware write commands, should only be altered by the BDM hardware or standard firmware lookup table as part of BDM command execution.

**Table 270. BDMSTS Field Descriptions**

| Field | Description |
|---|---|
| 7<br>ENBDM | **Enable BDM** — This bit controls whether the BDM is enabled or disabled. When enabled, BDM can be made active to allow firmware commands to be executed. When disabled, BDM cannot be made active but BDM hardware commands are still allowed.<br>0 BDM disabled<br>1 BDM enabled<br>Note: ENBDM is set out of reset in special single chip mode. In special single chip mode with the device secured, this bit is not set until after the Flash erase verify tests are complete. |
| 6<br>BDMACT | **BDM Active Status** — This bit becomes set upon entering BDM. The standard BDM firmware lookup table is then enabled and put into the memory map. BDMACT is cleared by a carefully timed store instruction in the standard BDM firmware as part of the exit sequence to return to user code and remove the BDM memory from the map.<br>0 BDM not active<br>1 BDM active |
| 4<br>SDV | **Shift Data Valid** — This bit is set and cleared by the BDM hardware. It is set after data has been transmitted as part of a BDM firmware or hardware read command or after data has been received as part of a BDM firmware or hardware write command. It is cleared when the next BDM command has been received or BDM is exited. SDV is used by the standard BDM firmware to control program flow execution.<br>0 Data phase of command not complete<br>1 Data phase of command is complete |

**Table 270. BDMSTS Field Descriptions (continued)**

| Field | Description |
|---|---|
| 3<br>TRACE | **TRACE1 BDM Firmware Command is Being Executed** — This bit gets set when a BDM TRACE1 firmware command is first recognized. It stays set until BDM firmware is exited by one of the following BDM commands: GO or GO_UNTIL[189].<br>0 TRACE1 command is not being executed<br>1 TRACE1 command is being executed |
| 1<br>UNSEC | **Unsecure** — If the device is secured this bit is only writable in special single chip mode from the BDM secure firmware. It is in a zero state as secure mode is entered so the secure BDM firmware lookup table is enabled and put into the memory map overlapping the standard BDM firmware lookup table.<br>The secure BDM firmware lookup table verifies the on-chip Flash is erased. This being the case, the UNSEC bit is set and the BDM program jumps to the start of the standard BDM firmware lookup table and the secure BDM firmware lookup table is turned off. If the erase test fails, the UNSEC bit is not asserted.<br>0 System is in a secured mode.<br>1 System is in a unsecured mode.<br>Note: When UNSEC is set, security is off and the user can change the state of the secure bits in the on-chip Flash EEPROM. Note that if the user does not change the state of the bits to “unsecured” mode, the system is secured again when it is next taken out of reset.After reset this bit has no meaning or effect when the security byte in the Flash EEPROM is configured for unsecure mode. |

Register Global Address 0x3_FF06

**Table 271. BDM CCR Holding Register (BDMCCR)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | CCR7 | CCR6 | CCR5 | CCR4 | CCR3 | CCR2 | CCR1 | CCR0 |
| Reset | | | | | | | | |
| Special Single-Chip Mode | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| All Other Modes | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Read: All modes through BDM operation when not secured

Write: All modes through BDM operation when not secured

**NOTE**

When BDM is made active, the CPU stores the content of its CCR register in the BDMCCR register. However, out of special single-chip reset, the BDMCCR is set to 0xD8 and not 0xD0 which is the reset value of the CCR register in this CPU mode. Out of reset in all other modes the BDMCCR register is read zero.

When entering background debug mode, the BDM CCR holding register is used to save the condition code register of the user’s program. It is also used for temporary storage in the standard BDM firmware mode. The BDM CCR holding register can be written to modify the CCR value.

#### 5.31.3.2.2 BDM Program Page Index Register (BDMPPR)

Register Global Address 0x3_FF08

**Table 272. BDM Program Page Register (BDMPPR)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | BPAE | 0 | 0 | 0 | BPP3 | BPP2 | BPP1 | BPP0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented, Reserved

Read: All modes through BDM operation when not secured

Write: All modes through BDM operation when not secured

**Table 273. BDMPPR Field Descriptions**

| Field | Description |
|---|---|
| 7<br>BPAE | **BDM Program Page Access Enable Bit** — BPAE enables program page access for BDM hardware and firmware read/write instructions. The BDM hardware commands used to access the BDM registers (READ_BD and WRITE_BD) can not be used for global accesses even if the BGAE bit is set.<br>0 BDM Program Paging disabled<br>1 BDM Program Paging enabled |
| 3–0<br>BPP[3:0] | **BDM Program Page Index Bits 3–0** — These bits define the selected program page. For more detailed information regarding the program page window scheme, refer to the S12S_MMC Block Guide. |

### 5.31.3.3 Family ID Assignment

The family ID is an 8-bit value located in the BDM ROM in active BDM (at global address: 0x3_FF0F). The read-only value is a unique family ID which is 0xC2 for devices with an HCS12S core.

## 5.31.4 Functional Description

The BDM receives and executes commands from a host via a single wire serial interface. There are two types of BDM commands: hardware and firmware commands.

Hardware commands are used to read and write target system memory locations and to enter active background debug mode, see Section 5.31.4.3, “BDM Hardware Commands". Target system memory includes all memory accessible by the CPU.

Firmware commands are used to read and write CPU resources and to exit from active background debug mode, see Section 5.31.4.4, “Standard BDM Firmware Commands". The CPU resources referred to are the accumulator (D), X index register (X), Y index register (Y), stack pointer (SP), and program counter (PC).

Hardware commands can be executed at any time and in any mode excluding a few exceptions as highlighted (see Section 5.31.4.3, “BDM Hardware Commands") and in secure mode (see Section 5.31.4.1, “Security"). BDM firmware commands can only be executed when the system is not secure and is in active background debug mode (BDM).

### 5.31.4.1 Security

If the user resets into special single chip mode with the system secured, a secured mode BDM firmware lookup table is brought into the map overlapping a portion of the standard BDM firmware lookup table. The secure BDM firmware verifies the on-chip Flash EEPROM is erased. This being the case, the UNSEC and ENBDM bit gets set. The BDM program jumps to the start of the standard BDM firmware and the secured mode BDM firmware is turned off and all BDM commands are allowed. If the Flash do not verify as erased, the BDM firmware sets the ENBDM bit, without asserting UNSEC, and the firmware enters a loop. This causes the BDM hardware commands to become enabled, but does not enable the firmware commands. This allows the BDM hardware to be used to erase the Flash.

BDM operation is not possible in any other mode than special single chip mode when the device is secured. The device can only be unsecured via BDM serial interface in special single chip mode. For more information regarding security, see the S12S_9SEC Block Guide.

### 5.31.4.2 Enabling and Activating BDM

The system must be in active BDM to execute standard BDM firmware commands. BDM can be activated only after being enabled. BDM is enabled by setting the ENBDM bit in the BDM status (BDMSTS) register. The ENBDM bit is set by writing to the BDM status (BDMSTS) register, via the single-wire interface, using a hardware command such as WRITE_BD_BYTE.

After being enabled, BDM is activated by one of the following[184]:

- Hardware BACKGROUND command
- CPU BGND instruction
- Breakpoint force or tag mechanism[185]

When BDM is activated, the CPU finishes executing the current instruction and then begins executing the firmware in the standard BDM firmware lookup table. When BDM is activated by a breakpoint, the type of breakpoint used determines if BDM becomes active before or after execution of the next instruction.

Note:

184. BDM is enabled and active immediately out of special single-chip reset.

185. This method is provided by the S12S_DBG module.

**NOTE**

If an attempt is made to activate BDM before being enabled, the CPU resumes normal instruction execution after a brief delay. If BDM is not enabled, any hardware BACKGROUND commands issued are ignored by the BDM and the CPU is not delayed.

In active BDM, the BDM registers and standard BDM firmware lookup table are mapped to addresses 0x3_FF00 to 0x3_FFFF. BDM registers are mapped to addresses 0x3_FF00 to 0x3_FF0B. The BDM uses these registers which are readable anytime by the BDM. However, these registers are not readable by user programs.

When BDM is activated while CPU executes code overlapping with BDM firmware space the saved program counter (PC) is auto incremented by one from the BDM firmware, no matter what caused the entry into BDM active mode (BGND instruction, BACKGROUND command or breakpoints). In such a case the PC must be set to the next valid address via a WRITE_PC command before executing the GO command.

#### 5.31.4.3 BDM Hardware Commands

Hardware commands are used to read and write target system memory locations and to enter active background debug mode. Target system memory includes all memory accessible by the CPU such as on-chip RAM, Flash, I/O and control registers.

Hardware commands are executed with minimal or no CPU intervention and do not require the system to be in active BDM for execution, although, they can still be executed in this mode. When executing a hardware command, the BDM sub-block waits for a free bus cycle so the background access does not disturb the running application program. If a free cycle is not found within 128 clock cycles, the CPU is momentarily frozen so the BDM can steal a cycle. When the BDM finds a free cycle, the operation does not intrude on normal CPU operation provided it can be completed in a single cycle. However, if an operation requires multiple cycles the CPU is frozen until the operation is complete, even though the BDM found a free cycle. The BDM hardware commands are listed in Table 274.

The READ_BD and WRITE_BD commands allow access to the BDM register locations. These locations are not normally in the system memory map but share addresses with the application in memory. To distinguish between physical memory locations which share the same address, BDM memory resources are enabled just for the READ_BD and WRITE_BD access cycle. This allows the BDM to access BDM locations unobtrusively, even if the addresses conflict with the application memory map.

**Table 274. Hardware Commands**

| Command | Opcode (hex) | Data | Description |
|---|---|---|---|
| BACKGROUND | 90 | None | Enter background mode if BDM is enabled. If enabled, an ACK is issued when the part enters active background mode. |
| ACK_ENABLE | D5 | None | Enable Handshake. Issues an ACK pulse after the command is executed. |
| ACK_DISABLE | D6 | None | Disable Handshake. This command does not issue an ACK pulse. |
| READ_BD_BYTE | E4 | 16-bit address<br>16-bit data out | Read from memory with standard BDM firmware lookup table in map.<br>Odd address data on low byte; even address data on high byte. |
| READ_BD_WORD | EC | 16-bit address<br>16-bit data out | Read from memory with standard BDM firmware lookup table in map.<br>Must be aligned access. |
| READ_BYTE | E0 | 16-bit address<br>16-bit data out | Read from memory with standard BDM firmware lookup table out of map.<br>Odd address data on low byte; even address data on high byte. |
| READ_WORD | E8 | 16-bit address<br>16-bit data out | Read from memory with standard BDM firmware lookup table out of map. Must be aligned access. |
| WRITE_BD_BYTE | C4 | 16-bit address<br>16-bit data in | Write to memory with standard BDM firmware lookup table in map.<br>Odd address data on low byte; even address data on high byte. |
| WRITE_BD_WORD | CC | 16-bit address<br>16-bit data in | Write to memory with standard BDM firmware lookup table in map. Must be aligned access. |
| WRITE_BYTE | C0 | 16-bit address<br>16-bit data in | Write to memory with standard BDM firmware lookup table out of map.<br>Odd address data on low byte; even address data on high byte. |

**Table 274. Hardware Commands (continued)**

| Command | Opcode (hex) | Data | Description |
|---|---|---|---|
| WRITE_WORD | C8 | 16-bit address<br>16-bit data in | Write to memory with standard BDM firmware lookup table out of map. Must be aligned access. |

Note:
186. If enabled, ACK occurs when data is ready for transmission for all BDM READ commands and occurs after the write is complete for all BDM WRITE commands.

### 5.31.4.4 Standard BDM Firmware Commands

BDM firmware commands are used to access and manipulate CPU resources. The system must be in active BDM to execute standard BDM firmware commands, see Section 5.31.4.2, “Enabling and Activating BDM". Normal instruction execution is suspended while the CPU executes the firmware located in the standard BDM firmware lookup table. The hardware command BACKGROUND is the usual way to activate BDM.

As the system enters active BDM, the standard BDM firmware lookup table and BDM registers become visible in the on-chip memory map at 0x3_FF00–0x3_FFFF, and the CPU begins executing the standard BDM firmware. The standard BDM firmware watches for serial commands and executes them as they are received.

The firmware commands are shown in Table 275.

**Table 275. Firmware Commands**

| Command[187] | Opcode (hex) | Data | Description |
|---|---|---|---|
| READ_NEXT[188] | 62 | 16-bit data out | Increment X index register by 2 (X = X + 2), then read word X points to. |
| READ_PC | 63 | 16-bit data out | Read program counter. |
| READ_D | 64 | 16-bit data out | Read D accumulator. |
| READ_X | 65 | 16-bit data out | Read X index register. |
| READ_Y | 66 | 16-bit data out | Read Y index register. |
| READ_SP | 67 | 16-bit data out | Read stack pointer. |
| WRITE_NEXT | 42 | 16-bit data in | Increment X index register by 2 (X = X + 2), then write word to location pointed to by X. |
| WRITE_PC | 43 | 16-bit data in | Write program counter. |
| WRITE_D | 44 | 16-bit data in | Write D accumulator. |
| WRITE_X | 45 | 16-bit data in | Write X index register. |
| WRITE_Y | 46 | 16-bit data in | Write Y index register. |
| WRITE_SP | 47 | 16-bit data in | Write stack pointer. |
| GO | 08 | none | Go to user program. If enabled, ACK occurs when leaving active background mode. |
| GO_UNTIL[189] | 0C | none | Go to user program. If enabled, ACK occurs upon returning to active background mode. |
| TRACE1 | 10 | none | Execute one user instruction then return to active BDM. If enabled, ACK occurs upon returning to active background mode. |

**Table 275. Firmware Commands (continued)**

| Command[187] | Opcode (hex) | Data | Description |
|---|---|---|---|
| TAGGO -> GO | 18 | none | (Previous enable tagging and go to user program.)<br>This command is deprecated and should not be used.<br>Opcode is executed as a GO command. |

Note:

187. If enabled, ACK occurs when data is ready for transmission for all BDM READ commands and occurs after the write is complete for all BDM WRITE commands.

188. When the firmware command READ_NEXT or WRITE_NEXT is used to access the BDM address space the BDM resources are accessed rather than user code. Writing BDM firmware is not possible.

189. System stop disables the ACK function and ignored commands do not have an ACK-pulse (e.g., CPU in stop mode). The GO_UNTIL command does not get an Acknowledge if CPU executes the stop instruction before the “UNTIL” condition (BDM active again) is reached (see Section 5.31.4.7, “Serial Interface Hardware Handshake Protocol" last note).

### 5.31.4.5 BDM Command Structure

Hardware and firmware BDM commands start with an 8-bit opcode followed by a 16-bit address and/or a 16-bit data word, depending on the command. All the read commands return 16 bits of data despite the byte or word implication in the command name.{Satatement}

8-bit reads return 16-bits of data, only one byte of which contains valid data. If reading an even address, the valid data appears in the MSB. If reading an odd address, the valid data appears in the LSB.

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

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 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 which 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 the ACK (acknowledge function) is used to indicate when an operation is complete. When using ACK, the delay times are automated.

Figure 56 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 the BKGD line idles in the high state. The time for an 8-bit command is 8 × 16 target clock cycles.[190]

Note:

190. Target clock cycles are cycles measured using the target MCU’s serial clock rate. See Section 5.31.4.6, “BDM Serial Interface" and Section 5.31.3.2.1, “BDM Status Register (BDMSTS)" for information on how serial clock rate is selected.

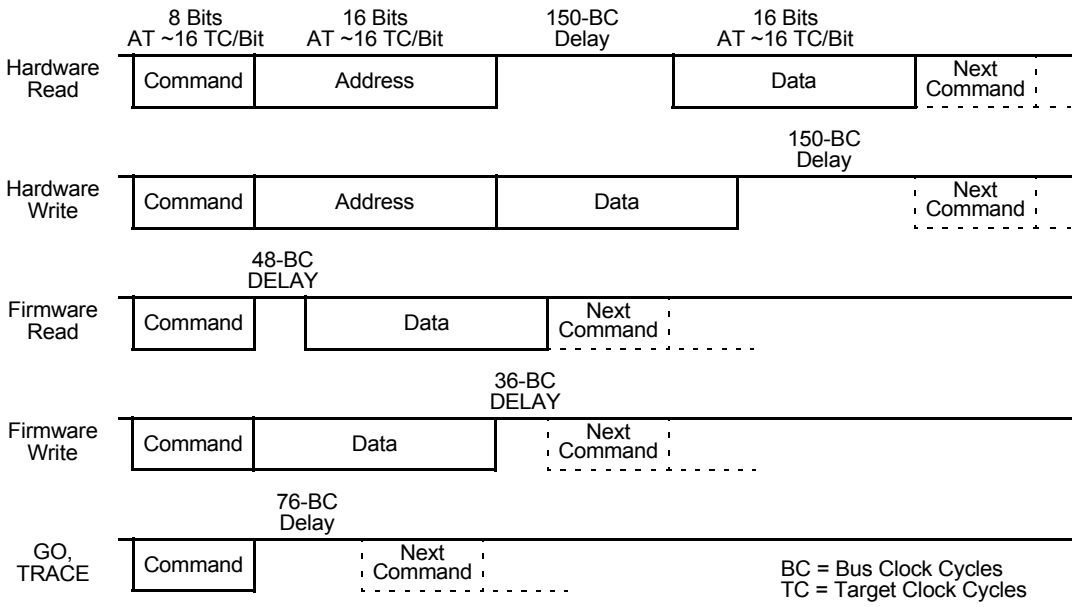



**Figure 56. BDM Command Structure**

### 5.31.4.6 BDM Serial Interface

The BDM communicates with external devices serially via the BKGD pin. During reset, this pin is a mode select input which selects between normal and special modes of operation. After reset, this pin becomes the dedicated serial interface pin for the BDM.

This clock is referred to as the target clock in the following explanation.

The BDM serial interface uses a clocking scheme in which the external host generates a falling edge on the BKGD pin to indicate the start of each bit time. This falling edge is sent for every bit whether data is transmitted or received. Data is transferred most significant bit (MSB) first at 16 target clock cycles per bit. The interface times out if 512 clock cycles occur between falling edges from the host.

The BKGD pin is a pseudo open-drain pin and has an weak on-chip active pull-up which is enabled at all times. It is assumed there is an external pull-up and so drivers connected to BKGD do not typically drive the high level. Since R-C rise time could be unacceptably long, the target system and host provide brief driven-high (speedup) pulses to drive BKGD to a logic 1. The source of this speedup pulse is the host for transmit cases and the target for receive cases.

The timing for host-to-target is shown in Figure 57 and of target-to-host in Figure 58 and Figure 59. All four cases begin when the host drives the BKGD pin low to generate a falling edge. Since the host and target are operating from separate clocks, it can take the target system up to one full clock cycle to recognize this edge. The target measures delays from this perceived start of the bit time while the host measures delays from the point it actually drove BKGD low to start the bit up to one target clock cycle earlier. Synchronization between the host and target is established in this manner at the start of every bit time.

Figure 57 shows an external host transmitting a logic 1 and transmitting a logic 0 to the BKGD pin of a target system. The host is asynchronous to the target, so there is up to a one clock-cycle delay from the host-generated falling edge to where the target recognizes this edge as the beginning of the bit time. Ten target clock cycles later, the target senses the bit level on the BKGD pin. Internal glitch detect logic requires the pin be driven high no later than eight target clock cycles after the falling edge for a logic 1 transmission.

Since the host drives the high speedup pulses in these two cases, the rising edges look like digitally driven signals.

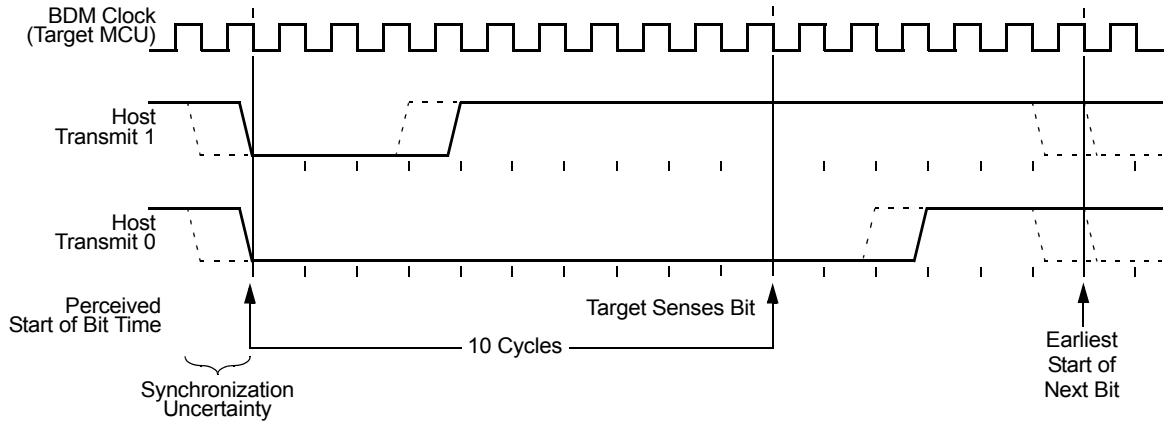



**Figure 57. BDM Host-to-Target Serial Bit Timing**

The receive cases are more complicated. Figure 58 shows the host receiving a logic 1 from the target system. Since the host is asynchronous to the target, there is up to one clock-cycle delay from the host-generated falling edge on BKGD to the perceived start of the bit time in the target. The host holds the BKGD pin low long enough for the target to recognize it (at least two target clock cycles). The host must release the low drive before the target drives a brief high speedup pulse seven target clock cycles after the perceived start of the bit time. The host should sample the bit level about 10 target clock cycles after it started the bit time.

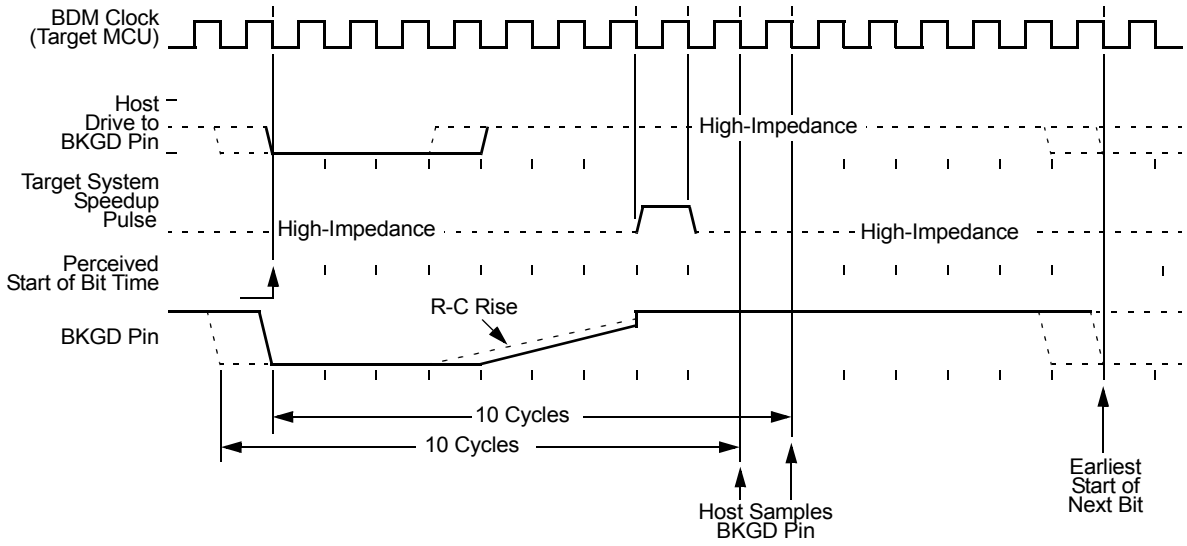



**Figure 58. BDM Target-to-Host Serial Bit Timing (Logic 1)**

Figure 59 shows the host receiving a logic 0 from the target. Since the host is asynchronous to the target, there is up to a one clock-cycle delay from the host-generated falling edge on BKGD to the start of the bit time as perceived by the target. The host initiates the bit time but the target finishes it. Since the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 target clock cycles then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 target clock cycles after starting the bit time.

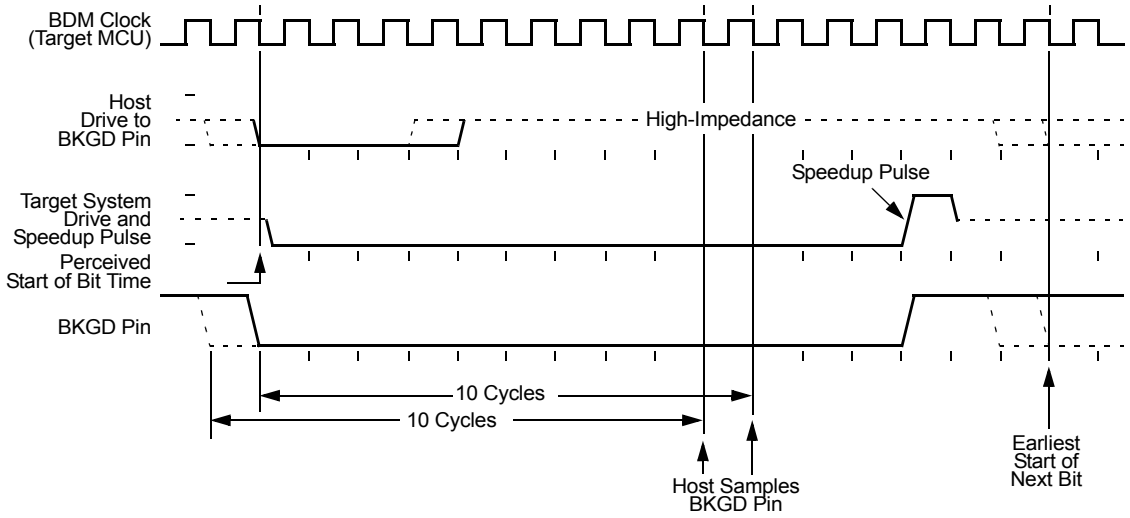



**Figure 59. BDM Target-to-Host Serial Bit Timing (Logic 0)**

### 5.31.4.7 Serial Interface Hardware Handshake Protocol

BDM commands require CPU execution is ultimately treated at the MCU bus rate. Since the BDM clock source can be modified, it is very helpful to provide a handshake protocol in which the host could determine when an issued command is executed by the CPU. The alternative is to always wait the amount of time equal to the appropriate number of cycles at the slowest possible rate the clock could be running. This sub-section describes the hardware handshake protocol.

The hardware handshake protocol signals to the host controller when an issued command was successfully executed by the target. This protocol is implemented by a 16 serial clock cycle low pulse followed by a brief speedup pulse in the BKGD pin. This pulse is generated by the target MCU when a command, issued by the host, has been successfully executed (see Figure 60). This pulse is referred to as the ACK pulse. After the ACK pulse has finished: the host can start the bit retrieval if the last issued command was a read command, or start a new command if the last command was a write command or a control command (BACKGROUND, GO, GO_UNTIL or TRACE1). The ACK pulse is not issued earlier than 32 serial clock cycles after the BDM command was issued. The end of the BDM command is assumed to be the 16th tick of the last bit. This minimum delay assures enough time for the host to perceive the ACK pulse. Note also, there is no upper limit for the delay between the command and the related ACK pulse, since the command execution depends upon the CPU bus, which in some cases could be very slow due to long accesses taking place.This protocol allows a great flexibility for the POD designers, since it does not rely on any accurate time measurement or short response time to any event in the serial communication.

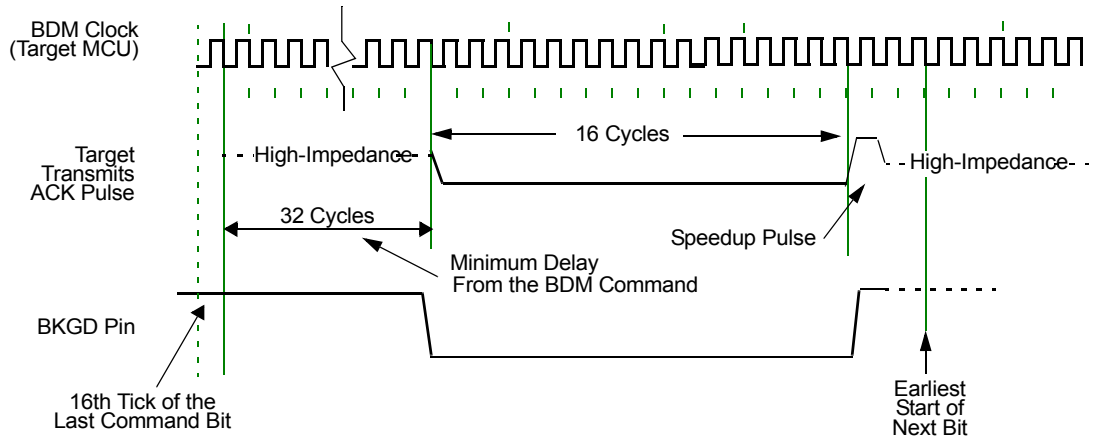



**Figure 60. Target Acknowledge Pulse (ACK)**

**NOTE**

If the ACK pulse was issued by the target, the host assumes the previous command was executed. If the CPU enters stop prior to executing a hardware command, the ACK pulse is not issued meaning the BDM command was not executed. After entering stop mode, the BDM command is no longer pending.

Figure 61 shows the ACK handshake protocol in a command level timing diagram. The READ_BYTE instruction is used as an example. First, the 8-bit instruction opcode is sent by the host, followed by the address of the memory location to be read. The target BDM decodes the instruction. A bus cycle is grabbed (free or stolen) by the BDM and it executes the READ_BYTE operation. Having retrieved the data, the BDM issues an ACK pulse to the host controller, indicating the addressed byte is ready to be retrieved. After detecting the ACK pulse, the host initiates the byte retrieval process. Note that data is sent in the form of a word and the host needs to determine which is the appropriate byte based on whether the address was odd or even.

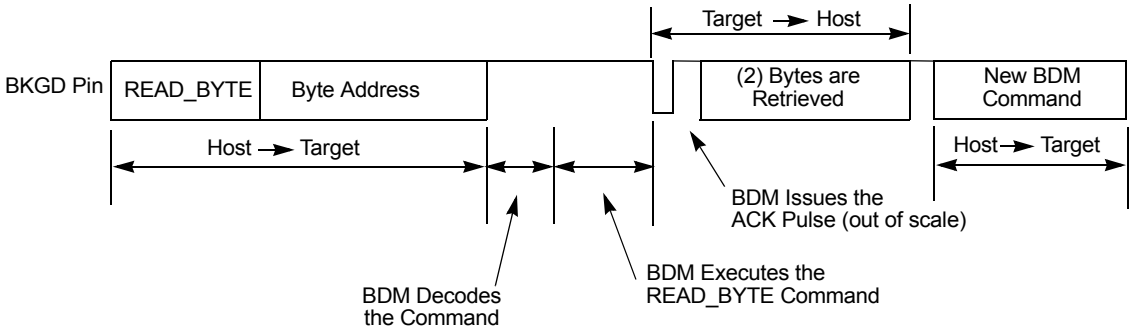



**Figure 61. Handshake Protocol at the Command Level**

Differently from the normal bit transfer (where the host initiates the transmission), the serial interface ACK handshake pulse is initiated by the target MCU by issuing a negative edge on the BKGD pin. The hardware handshake protocol in Figure 60 specifies the timing when the BKGD pin is being driven, so the host should follow this timing constraint in order to avoid the risk of an electrical conflict on the BKGD pin.

**NOTE**

The only place the BKGD pin can have an electrical conflict is when one side is driving low and the other side is issuing a speedup pulse (high). Other “highs” are pulled rather than driven. However, at low rates the time of the speedup pulse can become lengthy and so the potential conflict time becomes longer as well.

The ACK handshake protocol does not support nested ACK pulses. If a BDM command is not acknowledge by an ACK pulse, the host needs to abort the pending command first in order to be able to issue a new BDM command. When the CPU enters stop while the host issues a hardware command (e.g., WRITE_BYTE), the target discards the incoming command due to the stop being detected. Therefore, the command is not acknowledged by the target, which means the ACK pulse is not issued in this case. After a certain time the host (not aware of stop) should decide to abort any possible pending ACK pulse in order to be sure a new command can be issued. Therefore, the protocol provides a mechanism in which a command, and its corresponding ACK, can be aborted.

**NOTE**

The ACK pulse does not provide a timeout. This means for the GO_UNTIL command, it cannot be distinguished if a stop has been executed (command discarded and ACK not issued) or if the “UNTIL” condition (BDM active) is just not reached yet. Hence in any case where the ACK pulse of a command is not issued the possible pending command should be aborted before issuing a new command. See the handshake abort procedure described in Section 5.31.4.8, “Hardware Handshake Abort Procedure".

### 5.31.4.8 Hardware Handshake Abort Procedure

The abort procedure is based on the SYNC command. In order to abort a command, which had not issued the corresponding ACK pulse, the host controller should generate a low pulse in the BKGD pin by driving it low for at least 128 serial clock cycles and then driving it high for one serial clock cycle, providing a speedup pulse.By detecting this long low pulse in the BKGD pin, the target executes the SYNC protocol, see Section 5.31.4.9, “SYNC — Request Timed Reference Pulse", and assumes the pending command and therefore the related ACK pulse, are being aborted. Therefore, after the SYNC protocol has been completed the host is free to issue new BDM commands. For BDM firmware READ or WRITE commands it can not be guaranteed the pending command is aborted when issuing a SYNC before the corresponding ACK pulse. There is a short latency time from the time the READ or WRITE access begins until it is finished and the corresponding ACK pulse is issued. The latency time depends on the firmware READ or WRITE command is issued and on the selected bus clock rate. When the SYNC command starts during this latency time the READ or WRITE command is not aborted, but the corresponding ACK pulse is aborted. A pending GO, TRACE1 or GO_UNTIL command can not be aborted. Only the corresponding ACK pulse can be aborted by the SYNC command.

Although it is not recommended, the host could abort a pending BDM command by issuing a low pulse in the BKGD pin shorter than 128 serial clock cycles, which is not interpreted as the SYNC command. The ACK is actually aborted when a negative edge is perceived by the target in the BKGD pin. The short abort pulse should have at least 4 clock cycles keeping the BKGD pin low, in order to allow the negative edge to be detected by the target.In this case, the target does not execute the SYNC protocol but the pending command is aborted along with the ACK pulse. The potential problem with this abort procedure is when there is a conflict between the ACK pulse and the short abort pulse. In this case, the target may not perceive the abort pulse. The worst case is when the pending command is a read command (i.e., READ_BYTE). If the abort pulse is not perceived by the target the host attempts to send a new command after the abort pulse was issued, while the target expects the host to retrieve the accessed memory byte. In this case, host and target runs out of synchronism. However, if the command to be aborted is not a read command the short abort pulse could be used. After a command is aborted the target assumes the next negative edge, after the abort pulse, is the first bit of a new BDM command.

**NOTE**

The details about the short abort pulse are being provided only as a reference for the reader to better understand the BDM internal behavior. It is not recommended this procedure be used in a real application.

Since the host knows the target serial clock frequency, the SYNC command (used to abort a command) does not need to consider the lower possible target frequency. In this case, the host could issue a SYNC very close to the 128 serial clock cycles length. Providing a small overhead on the pulse length in order to assure the SYNC pulse is not misinterpreted by the target. See Section 5.31.4.9, “SYNC — Request Timed Reference Pulse".

Figure 62 shows a SYNC command being issued after a READ_BYTE, which aborts the READ_BYTE command. Note that, after the command is aborted a new command could be issued by the host computer.

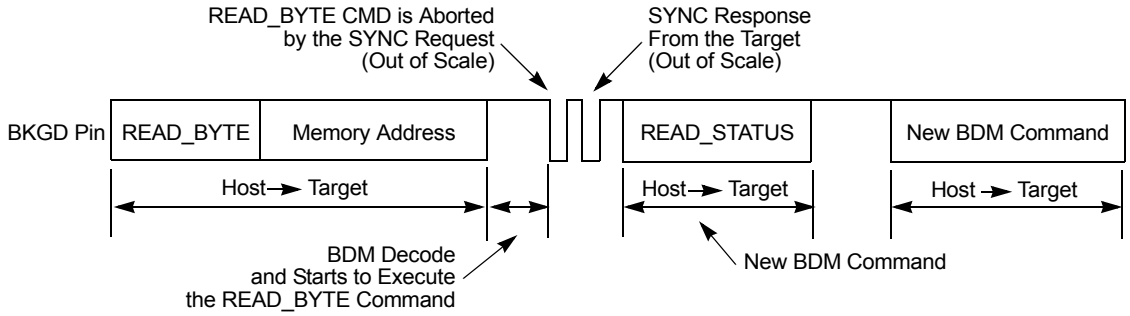



**Figure 62. ACK Abort Procedure at the Command Level**

**NOTE**

Figure 62 does not represent the signals in a true timing scale

Figure 63 shows a conflict between the ACK pulse and the SYNC request pulse. This conflict could occur if a POD device is connected to the target BKGD pin and the target is already in debug active mode. Consider that the target CPU is executing a pending BDM command at the exact moment the POD is being connected to the BKGD pin. In this case, an ACK pulse is issued along with the SYNC command. In this case, there is an electrical conflict between the ACK speedup pulse and the SYNC pulse. Since this is not a probable situation, the protocol does not prevent this conflict from happening.

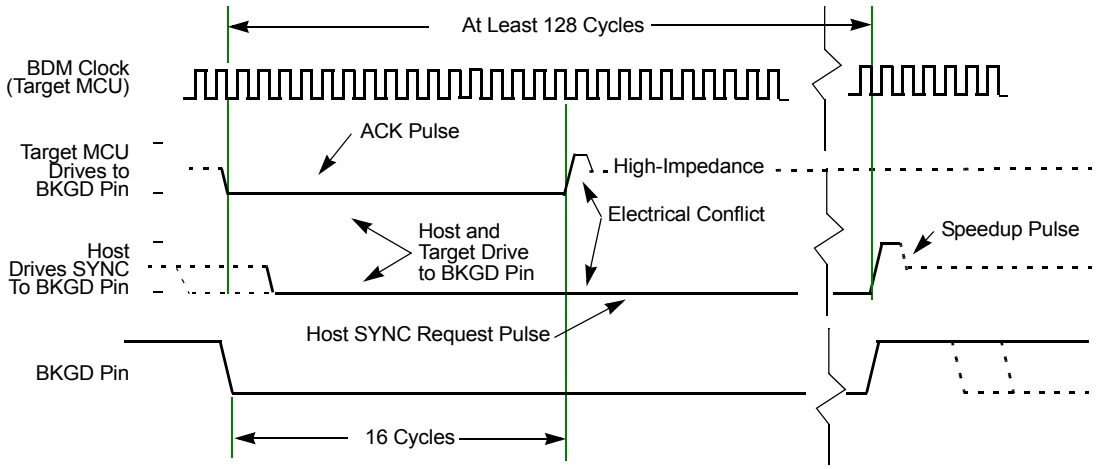



**Figure 63. ACK Pulse and SYNC Request Conflict**

**NOTE**

This information is being provided so the MCU integrator is aware such a conflict could occur.

The hardware handshake protocol is enabled by the ACK_ENABLE and disabled by the ACK_DISABLE BDM commands. This provides backwards compatibility with the existing POD devices which are not able to execute the hardware handshake protocol. It also allows for new POD devices, supporting the hardware handshake protocol, to freely communicate with the target device. If desired, without the need for waiting for the ACK pulse.

The commands are described as follows:

- ACK_ENABLE — enables the hardware handshake protocol. The target issues the ACK pulse when a CPU command is executed by the CPU. The ACK_ENABLE command itself also has the ACK pulse as a response.
- ACK_DISABLE — disables the ACK pulse protocol. In this case, the host needs to use the worst case delay time at the appropriate places in the protocol.

The default state of the BDM after reset is hardware handshake protocol disabled.

All the read commands ACK (if enabled) when the data bus cycle has completed and the data is then ready for reading out by the BKGD serial pin. All the write commands ACK (if enabled) after the data has been received by the BDM through the BKGD serial pin and when the data bus cycle is complete. See Section 5.31.4.3, “BDM Hardware Commands" and Section 5.31.4.4, “Standard BDM Firmware Commands" for more information on the BDM commands.

The ACK_ENABLE sends an ACK pulse when the command has been completed. This feature could be used by the host to evaluate if the target supports the hardware handshake protocol. If an ACK pulse is issued in response to this command, the host knows the target supports the hardware handshake protocol. If the target does not support the hardware handshake protocol the ACK pulse is not issued. In this case, the ACK_ENABLE command is ignored by the target since it is not recognized as a valid command.

The BACKGROUND command issues an ACK pulse when the CPU changes from normal to background mode. The ACK pulse related to this command could be aborted using the SYNC command.

The GO command issues an ACK pulse when the CPU exits from background mode. The ACK pulse related to this command could be aborted using the SYNC command.

The GO_UNTIL command is equivalent to a GO command with exception that the ACK pulse, in this case, is issued when the CPU enters into background mode. This command is an alternative to the GO command and should be used when the host wants to trace if a breakpoint match occurs and causes the CPU to enter active background mode. Note that the ACK is issued whenever the CPU enters BDM, which could be caused by a breakpoint match or by a BGND instruction being executed. The ACK pulse related to this command could be aborted using the SYNC command.

The TRACE1 command has the related ACK pulse issued when the CPU enters background active mode after one instruction of the application program is executed. The ACK pulse related to this command could be aborted using the SYNC command.

#### 5.31.4.9 SYNC — Request Timed Reference Pulse

The SYNC command is unlike other BDM commands because the host does not necessarily know the correct communication speed to use for BDM communications until after it has analyzed the response to the SYNC command. To issue a SYNC command, the host should perform the following steps:

1. Drive the BKGD pin low for at least 128 cycles at the lowest possible BDM serial communication frequency
2. Drive BKGD high for a brief speedup pulse to get a fast rise time (this speedup pulse is typically one cycle of the host clock.)
3. Remove all drive to the BKGD pin so it reverts to high impedance.
4. Listen to the BKGD pin for the sync response pulse.

Upon detecting the SYNC request from the host, the target performs the following steps:

1. Discards any incomplete command received or bit retrieved.
2. Waits for BKGD to return to a logic one.
3. Delays 16 cycles to allow the host to stop driving the high speedup pulse.
4. Drives BKGD low for 128 cycles at the current BDM serial communication frequency.
5. Drives a one-cycle high speedup pulse to force a fast rise time on BKGD.
6. Removes all drive to the BKGD pin so it reverts to high impedance.

The host measures the low time of this 128 cycle SYNC response pulse and determines the correct speed for subsequent BDM communications. Typically, the host can determine the correct communication speed 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 considers 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 is not issued.

#### 5.31.4.10 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 is 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, the execution of the user code is done step by step but 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 over the BGND instruction results in a return address pointing to BDM firmware address space.

When tracing through user code which contains stop instructions the following happens when the stop instruction is traced:

The CPU enters stop 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 instruction and still being in stop mode. If system stop mode is entered (all bus masters are in stop mode) no BDM command is operational.

As soon as stop 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 is discarded when tracing a stop instruction. Hence there is no ACK pulse when BDM active mode is entered as part of the TRACE1 command after CPU exited from stop mode. All valid commands sent during CPU being in stop mode or after CPU exited from stop mode has an ACK pulse. The handshake feature becomes disabled only when system stop mode has been reached. Hence after a system stop mode the handshake feature must be enabled again by sending the ACK_ENABLE command.

#### 5.31.4.11 Serial Communication Timeout

The host initiates a host-to-target serial transmission by generating a falling edge on the BKGD pin. If BKGD is kept low for more than 128 target clock cycles, the target understands a SYNC command was issued. In this case, the target keeps waiting for a rising edge on BKGD in order to answer the SYNC request pulse. If the rising edge is not detected, the target keeps waiting forever without any timeout limit.

Consider now the case where the host returns BKGD to logic one before 128 cycles. This is interpreted as a valid bit transmission, and not as a SYNC request. The target keeps waiting for another falling edge marking the start of a new bit. If, however, a new falling edge is not detected by the target within 512 clock cycles since the last falling edge, a timeout occurs and the current command is discarded without affecting memory or the operating mode of the MCU. This is referred to as a soft-reset.

If a read command is issued but the data is not retrieved within 512 serial clock cycles, a soft-reset occurs causing the command to be disregarded. The data is not available for retrieval after the timeout has occurred. This is the expected behavior if the handshake protocol is not enabled. In order to allow the data to be retrieved even with a large clock frequency mismatch (between BDM and CPU) when the hardware handshake protocol is enabled, the time out between a read command and the data retrieval is disabled. Therefore, the host could wait for more then 512 serial clock cycles and still be able to retrieve the data from an issued read command. However, once the handshake pulse (ACK pulse) is issued, the timeout feature is re-activated, meaning the target times out after 512 clock cycles. Therefore, the host needs to retrieve the data within a 512 serial clock cycles time frame after the ACK pulse had been issued. After this period, the read command is discarded and the data is no longer available for retrieval. Any negative edge in the BKGD pin after the timeout period is considered to be a new command or a SYNC request.

Note that whenever a partially issued command, or partially retrieved data, has occurred the timeout in the serial communication is active. This means if a time frame higher than 512 serial clock cycles is observed between two consecutive negative edges and the command being issued or data being retrieved is not complete, a soft-reset occurs causing the partially received command or data retrieved to be disregarded. The next negative edge in the BKGD pin, after a soft-reset has occurred, is considered by the target as the start of a new BDM command, or the start of a SYNC request pulse.

## 5.32 S12S Debug Module (S12SDBGV2)

### 5.32.1 Introduction

The S12SDBG module provides an on-chip trace buffer with flexible triggering capability to allow non-intrusive debug of application software. The S12SDBG module is optimized for S12SCPU debugging.

Typically the S12SDBG module is used in conjunction with the S12SBDM module, whereby the user configures the S12SDBG module for a debugging session over the BDM interface. Once configured the S12SDBG module is armed and the device leaves BDM returning control to the user program, which is then monitored by the S12SDBG module. Alternatively the S12SDBG module can be configured over a serial interface using SWI routines.

#### 5.32.1.1 Glossary Of Terms

COF: Change Of Flow. Change in the program flow due to a conditional branch, indexed jump or interrupt.

BDM: Background Debug Mode

S12SBDM: Background Debug Module

DUG: Device User Guide, describing the features of the device into which the DBG is integrated.

WORD: 16 bit data entity

Data Line: 20 bit data entity

CPU: S12SCPU module

DBG: S12SDBG module

POR: Power On Reset

Tag: Tags can be attached to CPU opcodes as they enter the instruction pipe. If the tagged opcode reaches the execution stage a tag hit occurs.

#### 5.32.1.2 Overview

The comparators monitor the bus activity of the CPU module. A match can initiate a state sequencer transition. On a transition to the Final State, bus tracing is triggered and/or a breakpoint can be generated.

Independent of comparator matches a transition to Final State with associated tracing and breakpoint can be triggered immediately by writing to the TRIG control bit.

The trace buffer is visible through a 2-byte window in the register address map and can be read out using standard 16-bit word reads. Tracing is disabled when the MCU system is secured.

#### 5.32.1.3 Features

- Three comparators (A, B and C)
  - Comparators A compares the full address bus and full 16-bit data bus
  - Comparator A features a data bus mask register
  - Comparators B and C compare the full address bus only
  - Each comparator features selection of read or write access cycles
  - Comparator B allows selection of byte or word access cycles
  - Comparator matches can initiate state sequencer transitions
- Three comparator modes
  - Simple address/data comparator match mode
  - Inside address range mode, Addmin $\leq$ Address $\leq$ Addmax
  - Outside address range match mode, Address $<$ Addmin or Address $>$ Addmax
- Two types of matches
  - Tagged — This matches just before a specific instruction begins execution
  - Force — This is valid on the first instruction boundary after a match occurs
- Two types of breakpoints
  - CPU breakpoint entering BDM on breakpoint (BDM)
  - CPU breakpoint executing SWI on breakpoint (SWI)
- Trigger mode independent of comparators
  - TRIG Immediate software trigger

- Four trace modes
  - Normal: change of flow (COF) PC information is stored (see Section 5.32.4.5.2.1, “Normal Mode") for change of flow definition.
  - Loop1: same as Normal but inhibits consecutive duplicate source address entries
  - Detail: address and data for all cycles except free cycles and opcode fetches are stored
  - Compressed Pure PC: all program counter addresses are stored
- 4-stage state sequencer for trace buffer control
  - Tracing session trigger linked to Final State of state sequencer
  - Begin and End alignment of tracing to trigger

### 5.32.1.4 Modes of Operation

The DBG module can be used in all MCU functional modes.

During BDM hardware accesses and whilst the BDM module is active, CPU monitoring is disabled. When the CPU enters active BDM Mode through a BACKGROUND command, the DBG module, if already armed, remains armed.

The DBG module tracing is disabled if the MCU is secure, however, breakpoints can still be generated

**Table 276. Mode Dependent Restriction Summary**

| BDM Enable | BDM Active | MCU Secure | Comparator Matches Enabled | Breakpoints Possible | Tagging Possible | Tracing Possible |
|---|---|---|---|---|---|---|
| x | x | 1 | Yes | Yes | Yes | No |
| 0 | 0 | 0 | Yes | Only SWI | Yes | Yes |
| 0 | 1 | 0 | Active BDM not possible when not enabled | | | |
| 1 | 0 | 0 | Yes | Yes | Yes | Yes |
| 1 | 1 | 0 | No | No | No | No |

### 5.32.1.5 Block Diagram

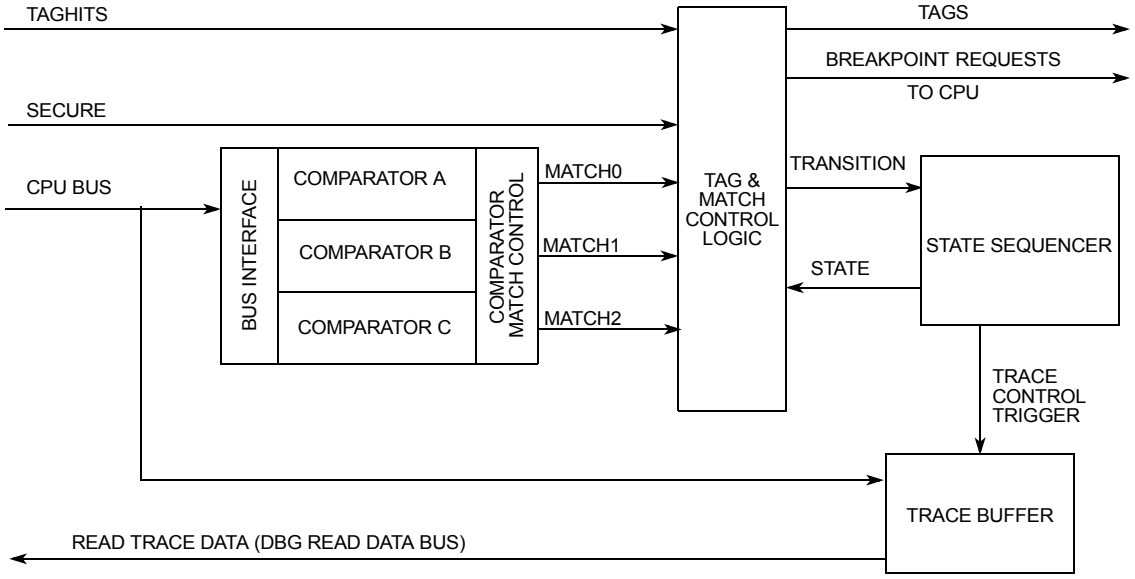



**Figure 64. Debug Module Block Diagram**

## 5.32.2 External Signal Description

There are no external signals associated with this module.

### 5.32.3 Memory Map and Registers

#### 5.32.3.1 Module Memory Map

A summary of the registers associated with the DBG sub-block is shown in Figure 277. Detailed descriptions of the registers and bits are given in the subsections which follow.

**Table 277. Quick Reference to DBG Registers**

| Address | Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0020 | DBGC1 | R | ARM | 0 | 0 | BDM | DBGBRK | 0 | COMRV | |
| | | W | | TRIG | | | | | | |
| 0x0021 | DBGSR | R | TBF[191] | 0 | 0 | 0 | 0 | SSF2 | SSF1 | SSF0 |
| | | W | | | | | | | | |
| 0x0022 | DBGTCR | R | 0 | TSOURCE | 0 | 0 | TRCMOD | | 0 | TALIGN |
| | | W | | | | | | | | |
| 0x0023 | DBGC2 | R | 0 | 0 | 0 | 0 | 0 | 0 | ABCM | |
| | | W | | | | | | | | |
| 0x0024 | DBGTBH | R | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | | W | | | | | | | | |
| 0x0025 | DBGTBL | R | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | W | | | | | | | | |
| 0x0026 | DBGCNT | R | TBF | 0 | CNT | | | | | |
| | | W | | | | | | | | |
| 0x0027 | DBGSCRX | R | 0 | 0 | 0 | 0 | SC3 | SC2 | SC1 | SC0 |
| | | W | | | | | | | | |
| 0x0027 | DBGMFR | R | 0 | 0 | 0 | 0 | 0 | MC2 | MC1 | MC0 |
| | | W | | | | | | | | |
| 0x0028 | DBGACTL | R | SZE | SZ | TAG | BRK | RW | RWE | NDB | COMPE |
| | | W | | | | | | | | |
| 0x0028 | DBGBCTL | R | SZE | SZ | TAG | BRK | RW | RWE | 0 | COMPE |
| | | W | | | | | | | | |
| 0x0028 | DBGCCTL | R | 0 | 0 | TAG | BRK | RW | RWE | 0 | COMPE |
| | | W | | | | | | | | |
| 0x0029 | DBGXAH | R | 0 | 0 | 0 | 0 | 0 | 0 | Bit 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 | | | | | | | | |

**Table 277. Quick Reference to DBG Registers (continued)**

| Address | Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 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 | | | | | | | | |

Note:
191. This bit is visible at DBGCNT[7] and DBGSR[7]
192. This represents the contents if the Comparator A control register is blended into this address.
193. This represents the contents if the Comparator B control register is blended into this address.
194. This represents the contents if the Comparator C control register is blended into this address.

### 5.32.3.2 Register Descriptions

This section consists of the DBG control and trace buffer register descriptions in address order. Each comparator has a bank of registers visible through an 8-byte window between 0x0028 and 0x002F in the DBG module register address map. When ARM is set in DBGC1, the only bits in the DBG module registers which can be written are ARM, TRIG, and COMRV[1:0]

#### 5.32.3.2.1 Debug Control Register 1 (DBGC1)

**Table 278. Debug Control Register (DBGC1)**

Address: 0x0020

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | ARM | 0 | 0 | BDM | DBGBRK | 0 | COMRV | |
| W | | TRIG | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Read: Anytime

Write: Bits 7, 1, 0 anytime
Bit 6 can be written anytime but always reads back as 0.
Bits 4:3 anytime DBG is not armed.

**NOTE**

When disarming the DBG by clearing ARM with software, the contents of bits[4:3] are not affected by the write, since up until the write operation, ARM = 1 preventing these bits from being written. These bits must be cleared using a second write if required.

**Table 279. DBGC1 Field Descriptions**

| Field | Description |
|---|---|
| 7<br>ARM | **Arm Bit** — The ARM bit controls whether the DBG module is armed. This bit can be set and cleared by user software and is automatically cleared on completion of a debug session, or if a breakpoint is generated with tracing not enabled. On setting this bit the state sequencer enters State1.<br>0 Debugger disarmed<br>1 Debugger armed |
| 6<br>TRIG | **Immediate Trigger Request Bit** — This bit when written to 1 requests an immediate trigger independent of state sequencer status. When tracing is complete a forced breakpoint may be generated depending upon DBGBRK and BDM bit settings. This bit always reads back a 0. Writing a 0 to this bit has no effect. If the DBGTCR_TSOURCE bit is clear no tracing is carried out. If tracing has already commenced using BEGIN trigger alignment, it continues until the end of the tracing session as defined by the TALIGN bit, thus TRIG has no affect. In secure mode tracing is disabled and writing to this bit cannot initiate a tracing session.<br>The session is ended by setting TRIG and ARM simultaneously.<br>0 Do not trigger until the state sequencer enters the Final State.<br>1 Trigger immediately |
| 4<br>BDM | **Background Debug Mode Enable** — This bit determines if a breakpoint causes the system to enter Background Debug Mode (BDM) or initiate a Software Interrupt (SWI). If this bit is set but the BDM is not enabled by the ENBDM bit in the BDM module, then breakpoints default to SWI.<br>0 Breakpoint to Software Interrupt if BDM inactive. Otherwise no breakpoint.<br>1 Breakpoint to BDM, if BDM enabled. Otherwise breakpoint to SWI |
| 3<br>DBGBRK | **S12SDBG Breakpoint Enable Bit** — The DBGBRK bit controls whether the debugger is request a breakpoint on reaching the state sequencer Final State. If tracing is enabled, the breakpoint is generated on completion of the tracing session. If tracing is not enabled, the breakpoint is generated immediately.<br>0 No Breakpoint generated<br>1 Breakpoint generated |
| 1–0<br>COMRV | **Comparator Register Visibility Bits** — These bits determine which bank of comparator register is visible in the 8-byte window of the S12SDBG module address map, located between 0x0028 to 0x002F. Furthermore these bits determine which register is visible at the address 0x0027. See Table 280. |

**Table 280. COMRV Encoding**

| COMRV | Visible Comparator | Visible Register at 0x0027 |
|---|---|---|
| 00 | Comparator A | DBGSCR1 |
| 01 | Comparator B | DBGSCR2 |
| 10 | Comparator C | DBGSCR3 |
| 11 | None | DBGMFR |

#### 5.32.3.2.2 Debug Status Register (DBGSR)

**Table 281. Debug Status Register (DBGSR)**

Address: 0x0021

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | TBF | 0 | 0 | 0 | 0 | SSF2 | SSF1 | SSF0 |
| W | | | | | | | | |
| Reset | — | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Read: Anytime

Write: Never

**Table 282. DBGSR Field Descriptions**

| Field | Description |
|---|---|
| 7<br>TBF | **Trace Buffer Full** — The TBF bit indicates the trace buffer has stored 64 or more lines of data since it was last armed. If this bit is set, then all 64 lines is valid data, regardless of the value of DBGCNT bits. The TBF bit is cleared when ARM in DBGC1 is written to a one. The TBF is cleared by the power on reset initialization. Other system generated resets have no affect on this bit<br>This bit is also visible at DBGCNT[7] |
| 2–0<br>SSF[2:0] | **State Sequencer Flag Bits** — The SSF bits indicate in which state the State Sequencer is currently in. During a debug session on each transition to a new state these bits are updated. If the debug session is ended by software clearing the ARM bit, then these bits retain their value to reflect the last state of the state sequencer before disarming. If a debug session is ended by an internal event, then the state sequencer returns to state0 and these bits are cleared to indicate state0 was entered during the session. On arming the module the state sequencer enters state1 and these bits are forced to SSF[2:0] = 001. See Table 283. |

**Table 283. SSF[2:0] — State Sequence Flag Bit Encoding**

| SSF[2:0] | Current State |
|---|---|
| 000 | State0 (disarmed) |
| 001 | State1 |
| 010 | State2 |
| 011 | State3 |
| 100 | Final State |
| 101,110,111 | Reserved |

#### 5.32.3.2.3 Debug Trace Control Register (DBGTCR)

**Table 284. Debug Trace Control Register (DBGTCR)**

Address: 0x0022

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

Read: Anytime

Write: Bit 6 only when DBG is neither secure nor armed.Bits 3,2,0 anytime the module is disarmed.

**Table 285. DBGTCR Field Descriptions**

| Field | Description |
|---|---|
| 6<br>TSOURCE | **Trace Source Control Bit** — The TSOURCE bit enables a tracing session given a trigger condition. If the MCU system is secured, this bit cannot be set and tracing is inhibited.<br>This bit must be set to read the trace buffer.<br>0 Debug session without tracing requested<br>1 Debug session with tracing requested |
| 3–2<br>TRCMOD | **Trace Mode Bits** — See Section 5.32.4.5.2, “Trace Modes" for detailed Trace Mode descriptions. In Normal Mode, change of flow information is stored. In Loop1 Mode, change of flow information is stored but redundant entries into trace memory are inhibited. In Detail Mode, address and data for all memory and register accesses is stored. In Compressed Pure PC mode the program counter value for each instruction executed is stored. See Table 286. |
| 0<br>TALIGN | **Trigger Align Bit** — This bit controls whether the trigger is aligned to the beginning or end of a tracing session.<br>0 Trigger at end of stored data<br>1 Trigger before storing data |

**Table 286. TRCMOD Trace Mode Bit Encoding**

| TRCMOD | Description |
|---|---|
| 00 | Normal |
| 01 | Loop1 |
| 10 | Detail |
| 11 | Compressed Pure PC |

#### 5.32.3.2.4 Debug Control Register2 (DBGC2)

**Table 287. Debug Control Register2 (DBGC2)**

Address: 0x0023

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

= Unimplemented or Reserved

Read: Anytime

Write: Anytime the module is disarmed.

This register configures the comparators for range matching.

**Table 288. DBGC2 Field Descriptions**

| Field | Description |
|---|---|
| 1–0 ABCM[1:0] | A and B Comparator Match Control — These bits determine the A and B comparator match mapping as described in Table 289. |

**Table 289. ABCM Encoding**

| ABCM | Description |
|---|---|
| 00 | Match0 mapped to comparator A match: Match1 mapped to comparator B match. |
| 01 | Match 0 mapped to comparator A/B inside range: Match1 disabled. |
| 10 | Match 0 mapped to comparator A/B outside range: Match1 disabled. |
| 11 | Reserved(195) |

Note:

195. Currently defaults to Comparator A, Comparator B disabled

#### 5.32.3.2.5 Debug Trace Buffer Register (DBGTBH:DBGTBL)

**Table 290. Debug Trace Buffer Register (DBGTB)**

Address: 0x0024, 0x0025

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<br>W | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| POR | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| Other Resets | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

Read: Only when unlocked AND unsecured AND not armed AND TSOURCE set.

Write: Aligned word writes when disarmed unlock the trace buffer for reading but do not affect trace buffer contents.

**Table 291. DBGTB Field Descriptions**

| Field | Description |
|---|---|
| 15–0<br>Bit[15:0] | **Trace Buffer Data Bits** — The Trace Buffer Register is a window through which the 20-bit wide data lines of the Trace Buffer may be read 16 bits at a time. Each valid read of DBGTB increments an internal trace buffer pointer which points to the next address to be read. When the ARM bit is set the trace buffer is locked to prevent reading. The trace buffer can only be unlocked for reading by writing to DBGTB with an aligned word write when the module is disarmed. The DBGTB register can be read only as an aligned word, any byte reads or misaligned access of these registers return 0 and do not cause the trace buffer pointer to increment to the next trace buffer address. Similarly reads while the debugger is armed or with the TSOURCE bit clear, return 0 and do not affect the trace buffer pointer. The POR state is undefined. Other resets do not affect the trace buffer contents. |

#### 5.32.3.2.6 Debug Count Register (DBGCNT)

**Table 292. Debug Count Register (DBGCNT)**

Address: 0x0026

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

= Unimplemented or Reserved

Read: Anytime

Write: Never

**Table 293. DBGCNT Field Descriptions**

| Field | Description |
|---|---|
| 7<br>TBF | **Trace Buffer Full** — The TBF bit indicates the trace buffer has stored 64 or more lines of data since it was last armed. If this bit is set, then all 64 lines is valid data, regardless of the value of DBGCNT bits. The TBF bit is cleared when ARM in DBGC1 is written to a one. The TBF is cleared by the power on reset initialization. Other system generated resets have no affect on this bit<br>This bit is also visible at DBGSR[7] |
| 5–0<br>CNT[5:0] | **Count Value** — The CNT bits indicate the number of valid data 20-bit data lines stored in the Trace Buffer. Table 294 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 continues in end-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 294. CNT Decoding Table**

| TBF | CNT[5:0] | Description |
|---|---|---|
| 0 | 000000 | No data valid |
| 0 | 000001<br>000010<br>000100<br>000110<br>…<br>111111 | 1 line valid<br>2 lines valid<br>4 lines valid<br>6 lines valid<br>…<br>63 lines valid |
| 1 | 000000 | 64 lines valid; if using Begin trigger alignment,<br>ARM bit is cleared and the tracing session ends. |
| 1 | 000001<br>…<br>…<br>111110 | 64 lines valid,<br>oldest data has been overwritten by most recent data |

#### 5.32.3.2.7 Debug State Control Registers

There is a dedicated control register for each of the state sequencer states 1 to 3 which determines if transitions from this state are allowed, depending upon comparator matches or tag hits, and defines the next state for the state sequencer following a match. The three debug state control registers are located at the same address in the register address map (0x0027). Each register can be accessed using the COMRV bits in DBGC1 to blend in the required register. The COMRV = 11 value blends in the match flag register (DBGMFR).

**Table 295. State Control Register Access Encoding**

| COMRV | Visible State Control Register |
|---|---|
| 00 | DBGSCR1 |
| 01 | DBGSCR2 |
| 10 | DBGSCR3 |
| 11 | DBGMFR |

##### 5.32.3.2.7.1 Debug State Control Register 1 (DBGSCR1)

**Table 296. Debug State Control Register 1 (DBGSCR1)**

Address: 0x0027

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | SC3 | SC2 | SC1 | SC0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Read: If COMRV[1:0] = 00

Write: If COMRV[1:0] = 00 and DBG is not armed.

This register is visible at 0x0027 only with COMRV[1:0] = 00. The state control register 1 selects the targeted next state whilst in State1. The matches refer to the match channels of the comparator match control logic as depicted in Figure 64 and described in Section 5.32.3.2.8.1, “Debug Comparator Control Register (DBGXCTL)". Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

**Table 297. DBGSCR1 Field Descriptions**

| Field | Description |
|---|---|
| 3–0 SC[3:0] | These bits select the targeted next state whilst in State1, based upon the match event. |

**Table 298. State1 Sequencer Next State Selection**

| SC[3:0] | Description (Unspecified matches have no effect) |
|---|---|
| 0000 | Any match to Final State |
| 0001 | Match1 to State3 |
| 0010 | Match2 to State2 |
| 0011 | Match1 to State2 |
| 0100 | Match0 to State2..... Match1 to State3 |
| 0101 | Match1 to State3.....Match0 to Final State |
| 0110 | Match0 to State2..... Match2 to State3 |
| 0111 | Either Match0 or Match1 to State2 |
| 1000 | Reserved |
| 1001 | Match0 to State3 |
| 1010 | Reserved |
| 1011 | Reserved |
| 1100 | Reserved |
| 1101 | Either Match0 or Match2 to Final State.....Match1 to State2 |
| 1110 | Reserved |
| 1111 | Reserved |

The priorities described in Table 331 dictate in the case of simultaneous matches, a match leading to final state has priority followed by the match on the lower channel number (0,1,2). Thus with SC[3:0]=1101 a simultaneous match0/match1 transitions to final state.

#### 5.32.3.2.7.2 Debug State Control Register 2 (DBGSCR2)

**Table 299. Debug State Control Register 2 (DBGSCR2)**

Address: 0x0027

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | SC3 | SC2 | SC1 | SC0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Read: If COMRV[1:0] = 01

Write: If COMRV[1:0] = 01 and DBG is not armed.

This register is visible at 0x0027 only with COMRV[1:0] = 01. The state control register 2 selects the targeted next state whilst in State2. The matches refer to the match channels of the comparator match control logic as depicted in Figure 64 and described in Section 5.32.3.2.8.1, “Debug Comparator Control Register (DBGXCTL)". Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

**Table 300. DBGSCR2 Field Descriptions**

| Field | Description |
|---|---|
| 3–0 SC[3:0] | These bits select the targeted next state whilst in State2, based upon the match event. |

**Table 301. State2 —Sequencer Next State Selection**

| SC[3:0] | Description (Unspecified matches have no effect) |
|---|---|
| 0000 | Match0 to State1..... Match2 to State3. |
| 0001 | Match1 to State3 |
| 0010 | Match2 to State3 |
| 0011 | Match1 to State3..... Match0 Final State |
| 0100 | Match1 to State1..... Match2 to State3. |
| 0101 | Match2 to Final State |
| 0110 | Match2 to State1..... Match0 to Final State |
| 0111 | Either Match0 or Match1 to Final State |
| 1000 | Reserved |
| 1001 | Reserved |
| 1010 | Reserved |
| 1011 | Reserved |
| 1100 | Either Match0 or Match1 to Final State.....Match2 to State3 |
| 1101 | Reserved |
| 1110 | Reserved |
| 1111 | Either Match0 or Match1 to Final State.....Match2 to State1 |

The priorities described in Table 331 dictate in the case of simultaneous matches, a match leading to final state has priority followed by the match on the lower channel number (0,1,2)

#### 5.32.3.2.7.3 Debug State Control Register 3 (DBGSCR3)

**Table 302. Debug State Control Register 3 (DBGSCR3)**

Address: 0x0027

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | SC3 | SC2 | SC1 | SC0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Read: If COMRV[1:0] = 10

Write: If COMRV[1:0] = 10 and DBG is not armed.

This register is visible at 0x0027 only with COMRV[1:0] = 10. The state control register three selects the targeted next state whilst in State3. The matches refer to the match channels of the comparator match control logic as depicted in Figure 64 and described in Section 5.32.3.2.8.1, “Debug Comparator Control Register (DBGXCTL)". Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

**Table 303. DBGSCR3 Field Descriptions**

| Field | Description |
|---|---|
| 3–0 SC[3:0] | These bits select the targeted next state whilst in State3, based upon the match event. |

**Table 304. State3 — Sequencer Next State Selection**

| SC[3:0] | Description (Unspecified matches have no effect) |
|---|---|
| 0000 | Match0 to State1 |
| 0001 | Match2 to State2..... Match1 to Final State |
| 0010 | Match0 to Final State..... Match1 to State1 |
| 0011 | Match1 to Final State..... Match2 to State1 |
| 0100 | Match1 to State2 |
| 0101 | Match1 to Final State |
| 0110 | Match2 to State2..... Match0 to Final State |
| 0111 | Match0 to Final State |
| 1000 | Reserved |
| 1001 | Reserved |
| 1010 | Either Match1 or Match2 to State1..... Match0 to Final State |
| 1011 | Reserved |
| 1100 | Reserved |
| 1101 | Either Match1 or Match2 to Final State..... Match0 to State1 |
| 1110 | Match0 to State2..... Match2 to Final State |
| 1111 | Reserved |

The priorities described in Table 331 dictate in the case of simultaneous matches, a match leading to final state has priority followed by the match on the lower channel number (0,1,2).

#### 5.32.3.2.7.4 Debug Match Flag Register (DBGMFR)

**Table 305. Debug Match Flag Register (DBGMFR)**

Address: 0x0027

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | 0 | MC2 | MC1 | MC0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Read: If COMRV[1:0] = 11

Write: Never

DBGMFR is visible at 0x0027 only with COMRV[1:0] = 11. It features 3 flag bits each mapped directly to a channel. Should a match occur on the channel during the debug session, then the corresponding flag is set and remains set until the next time the module is armed by writing to the ARM bit. Thus the contents are retained after a debug session for evaluation purposes. These flags cannot be cleared by software, they are cleared only when arming the module. A set flag does not inhibit the setting of other flags. Once a flag is set, further comparator matches on the same channel in the same session have no affect on this flag.

#### 5.32.3.2.8 Comparator Register Descriptions

Each comparator has a bank of registers visible through an 8-byte window in the DBG module register address map. Comparator A consists of 8 register bytes (3 address bus compare registers, two data bus compare registers, two data bus mask registers and a control register). Comparator B consists of four register bytes (three address bus compare registers and a control register). Comparator C consists of four register bytes (three address bus compare registers and a control register).

Each set of comparator registers can be accessed using the COMRV bits in the DBGC1 register. Unimplemented registers (e.g. Comparator B data bus and data bus masking) read as zero and cannot be written. The control register for comparator B differs from those of comparators A and C.

**Table 306. Comparator Register Layout**

| | | | |
|---|---|---|---|
| 0x0028 | CONTROL | Read/Write | Comparators A,B and C |
| 0x0029 | ADDRESS HIGH | Read/Write | Comparators A,B and C |
| 0x002A | ADDRESS MEDIUM | Read/Write | Comparators A,B and C |
| 0x002B | ADDRESS LOW | Read/Write | Comparators A,B and C |
| 0x002C | DATA HIGH COMPARATOR | Read/Write | Comparator A only |
| 0x002D | DATA LOW COMPARATOR | Read/Write | Comparator A only |
| 0x002E | DATA HIGH MASK | Read/Write | Comparator A only |
| 0x002F | DATA LOW MASK | Read/Write | Comparator A only |

#### 5.32.3.2.8.1 Debug Comparator Control Register (DBGXCTL)

The contents of this register bits 7 and 6 differ depending upon which comparator registers are visible in the 8-byte window of the DBG module register address map.

**Table 307. Debug Comparator Control Register DBGACTL (Comparator A)**

Address: 0x0028

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | SZE | SZ | TAG | BRK | RW | RWE | NDB | COMPE |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Table 308. Debug Comparator Control Register DBGBCTL (Comparator B)**

Address: 0x0028

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | SZE | SZ | TAG | BRK | RW | RWE | 0 | COMPE |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Table 309. Debug Comparator Control Register DBGCCTL (Comparator C)**

Address: 0x0028

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | TAG | BRK | RW | RWE | 0 | COMPE |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Read: DBGACTL if COMRV[1:0] = 00
DBGBCTL if COMRV[1:0] = 01
DBGCCTL if COMRV[1:0] = 10

Write: DBGACTL if COMRV[1:0] = 00 and DBG not armed
DBGBCTL if COMRV[1:0] = 01 and DBG not armed
DBGCCTL if COMRV[1:0] = 10 and DBG not armed

**Table 310. DBGXCTL Field Descriptions**

| Field | Description |
|---|---|
| 7<br>SZE<br>(Comparators A and B) | **Size Comparator Enable Bit** — The SZE bit controls whether access size comparison is enabled for the associated comparator. This bit is ignored if the TAG bit in the same register is set.<br>0 Word/Byte access size is not used in comparison<br>1 Word/Byte access size is used in comparison |
| 6<br>SZ<br>(Comparators A and B) | **Size Comparator Value Bit** — The SZ bit selects either word or byte access size in comparison for the associated comparator. This bit is ignored if the SZE bit is cleared or if the TAG bit in the same register is set.<br>0 Word access size is compared<br>1 Byte access size is compared |

**Table 310. DBGXCTL Field Descriptions (continued)**

| Field | Description |
|---|---|
| 5<br>TAG | **Tag Select** — This bit controls whether the comparator match has immediate effect, causing an immediate state sequencer transition or tag the opcode at the matched address. Tagged opcodes trigger only if they reach the execution stage of the instruction queue.<br>0 Allow state sequencer transition immediately on match<br>1 On match, tag the opcode. If the opcode is about to be executed allow a state sequencer transition |
| 4<br>BRK | **Break** — This bit controls whether a comparator match terminates a debug session immediately, independent of state sequencer state. To generate an immediate breakpoint the module breakpoints must be enabled using the DBGC1 bit DBGBRK.<br>0 The debug session termination is dependent upon the state sequencer and trigger conditions.<br>1 A match on this channel terminates the debug session immediately; breakpoints if active are generated, tracing, if active, is terminated and the module disarmed. |
| 3<br>RW | **Read/Write Comparator Value Bit** — The RW bit controls whether read or write is used in compare for the associated comparator. The RW bit is not used if RWE = 0. This bit is ignored if the TAG bit in the same register is set.<br>0 Write cycle is matched1 Read cycle is matched |
| 2<br>RWE | **Read/Write Enable Bit** — The RWE bit controls whether read or write comparison is enabled for the associated comparator.This bit is ignored if the TAG bit in the same register is set<br>0 Read/Write is not used in comparison<br>1 Read/Write is used in comparison |
| 1<br>NDB<br>(Comparator A) | **Not Data Bus** — The NDB bit controls whether the match occurs when the data bus matches the comparator register value or when the data bus differs from the register value. This bit is ignored if the TAG bit in the same register is set. This bit is only available for comparator A.<br>0 Match on data bus equivalence to comparator register contents<br>1 Match on data bus difference to comparator register contents |
| 0<br>COMPE | Determines if comparator is enabled<br>0 The comparator is not enabled<br>1 The comparator is enabled |

Table 311 shows the effect for RWE and RW on the comparison conditions. These bits are ignored if the corresponding TAG bit is set since the match occurs based on the tagged opcode reaching the execution stage of the instruction queue.

**Table 311. Read or Write Comparison Logic Table**

| RWE Bit | RW Bit | RW Signal | Comment |
|---|---|---|---|
| 0 | x | 0 | RW not used in comparison |
| 0 | x | 1 | RW not used in comparison |
| 1 | 0 | 0 | Write data bus |
| 1 | 0 | 1 | No match |
| 1 | 1 | 0 | No match |
| 1 | 1 | 1 | Read data bus |

#### 5.32.3.2.8.2 Debug Comparator Address High Register (DBGXAH)

**Table 312. Debug Comparator Address High Register (DBGXAH)**

Address: 0x0029

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | 0 | 0 | Bit 17 | Bit 16 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

MM912_634

The DBGC1_COMRV bits determine which comparator address registers are visible in the 8-byte window from 0x0028 to 0x002F as shown in Table 313.

**Table 313. Comparator Address Register Visibility**

| COMRV | Visible Comparator |
|---|---|
| 00 | DBGAAH, DBGAAM, DBGAAL |
| 01 | DBGBAH, DBGBAM, DBGBAL |
| 10 | DBGCAH, DBGCAM, DBGCAL |
| 11 | None |

Read: Anytime. See Table 313 for visible register encoding.

Write: If DBG not armed. See Table 313 for visible register encoding.

**Table 314. DBGXAH Field Descriptions**

| Field | Description |
|---|---|
| 1–0<br>Bit[17:16] | **Comparator Address High Compare Bits** — The Comparator address high compare bits control whether the selected comparator compares the address bus bits [17:16] to a logic one or logic zero.<br>0 Compare corresponding address bit to a logic zero<br>1 Compare corresponding address bit to a logic one |

#### 5.32.3.2.8.3 Debug Comparator Address Mid Register (DBGXAM)

**Table 315. Debug Comparator Address Mid Register (DBGXAM)**

Address: 0x002A

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Read: Anytime. See Table 313 for visible register encoding.

Write: If DBG not armed. See Table 313 for visible register encoding.

**Table 316. DBGXAM Field Descriptions**

| Field | Description |
|---|---|
| 7–0<br>Bit[15:8] | **Comparator Address Mid Compare Bits** — The Comparator address mid compare bits control whether the selected comparator compares the address bus bits [15:8] to a logic one or logic zero.<br>0 Compare corresponding address bit to a logic zero<br>1 Compare corresponding address bit to a logic one |

#### 5.32.3.2.8.4 Debug Comparator Address Low Register (DBGXAL)

**Table 317. Debug Comparator Address Low Register (DBGXAL)**

Address: 0x002B

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

Read: Anytime. See Table 313 for visible register encoding.

Write: If DBG not armed. See Table 313 for visible register encoding.

**Table 318. DBGXAL Field Descriptions**

| Field | Description |
|---|---|
| 7–0<br>Bits[7:0] | **Comparator Address Low Compare Bits** — The Comparator address low compare bits control whether the selected comparator compares the address bus bits [7:0] to a logic one or logic zero.<br>0 Compare corresponding address bit to a logic zero<br>1 Compare corresponding address bit to a logic one |

#### 5.32.3.2.8.5 Debug Comparator Data High Register (DBGADH)

**Table 319. Debug Comparator Data High Register (DBGADH)**

Address: 0x002C

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Read: If COMRV[1:0] = 00

Write: If COMRV[1:0] = 00 and DBG not armed.

**Table 320. DBGADH Field Descriptions**

| Field | Description |
|---|---|
| 7–0<br>Bits[15:8] | **Comparator Data High Compare Bits**— The Comparator data high compare bits control whether the selected comparator compares the data bus bits [15:8] to a logic one or logic zero. The comparator data compare bits are only used in comparison if the corresponding data mask bit is logic 1. This register is available only for comparator A. Data bus comparisons are only performed if the TAG bit in DBGACTL is clear.<br>0 Compare corresponding data bit to a logic zero<br>1 Compare corresponding data bit to a logic one |

#### 5.32.3.2.8.6 Debug Comparator Data Low Register (DBGADL)

**Table 321. Debug Comparator Data Low Register (DBGADL)**

Address: 0x002D

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

Read: If COMRV[1:0] = 00

Write: If COMRV[1:0] = 00 and DBG not armed.

**Table 322. DBGADL Field Descriptions**

| Field | Description |
|---|---|
| 7–0<br>Bits[7:0] | **Comparator Data Low Compare Bits** — The Comparator data low compare bits control whether the selected comparator compares the data bus bits [7:0] to a logic one or logic zero. The comparator data compare bits are only used in comparison if the corresponding data mask bit is logic 1. This register is available only for comparator A. Data bus comparisons are only performed if the TAG bit in DBGACTL is clear<br>0 Compare corresponding data bit to a logic zero<br>1 Compare corresponding data bit to a logic one |

Debug Comparator Data High Mask Register (DBGADHM)

**Table 323. Debug Comparator Data High Mask Register (DBGADHM)**

Address: 0x002E

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Read: If COMRV[1:0] = 00

Write: If COMRV[1:0] = 00 and DBG not armed.

**Table 324. DBGADHM Field Descriptions**

| Field | Description |
|---|---|
| 7–0<br>Bits[15:8] | **Comparator Data High Mask Bits** — The Comparator data high mask bits control whether the selected comparator compares the data bus bits [15:8] to the corresponding comparator data compare bits. Data bus comparisons are only performed if the TAG bit in DBGACTL is clear<br>0 Do not compare corresponding data bit Any value of corresponding data bit allows match.<br>1 Compare corresponding data bit |

#### 5.32.3.2.8.7 Debug Comparator Data Low Mask Register (DBGADLM)

**Table 325. Debug Comparator Data Low Mask Register (DBGADLM)**

Address: 0x002F

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

Read: If COMRV[1:0] = 00

Write: If COMRV[1:0] = 00 and DBG not armed.

**Table 326. DBGADLM Field Descriptions**

| Field | Description |
|---|---|
| 7–0<br>Bits[7:0] | **Comparator Data Low Mask Bits** — The Comparator data low mask bits control whether the selected comparator compares the data bus bits [7:0] to the corresponding comparator data compare bits. Data bus comparisons are only performed if the TAG bit in DBGACTL is clear<br>0 Do not compare corresponding data bit. Any value of corresponding data bit allows match<br>1 Compare corresponding data bit |

### 5.32.4 Functional Description

This section provides a complete functional description of the DBG module. If the part is in secure mode, the DBG module can generate breakpoints but tracing is not possible.

#### 5.32.4.1 S12SDBG Operation

Arming the DBG module by setting ARM in DBGC1 allows triggering the state sequencer, storing of data in the trace buffer and generation of breakpoints to the CPU. The DBG module is made up of four main blocks, the comparators, control logic, the state sequencer, and the trace buffer.

The comparators monitor the bus activity of the CPU. All comparators can be configured to monitor address bus activity. Comparator A can also be configured to monitor data bus activity and mask out individual data bus bits during a compare. Comparators can be configured to use R/W and word/byte access qualification in the comparison. A match with a comparator register value can initiate a state sequencer transition to another state (see Figure 66). Either forced or tagged matches are possible. Using a forced match, a state sequencer transition can occur immediately on a successful match of system busses and comparator registers. Whilst tagging, at a comparator match, the instruction opcode is tagged and only if the instruction reaches the execution stage of the instruction queue can a state sequencer transition occur. In the case of a transition to Final State, bus tracing is triggered and/or a breakpoint can be generated.

A state sequencer transition to final state (with associated breakpoint, if enabled) can be initiated by writing to the TRIG bit in the DBGC1 control register.

The trace buffer is visible through a 2-byte window in the register address map and must be read out using standard 16-bit word reads.

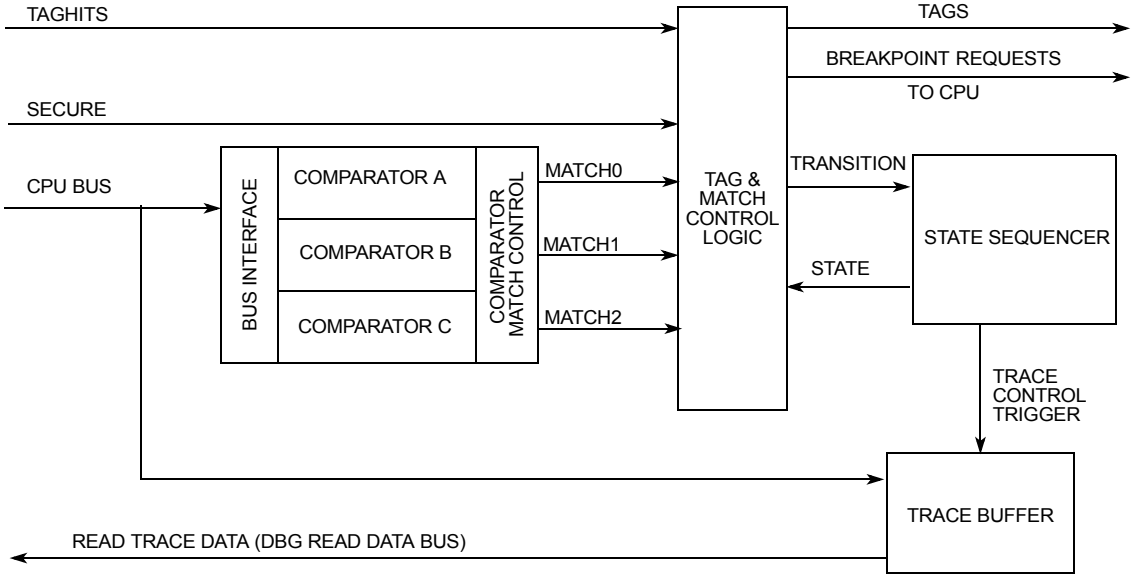



**Figure 65. DBG Overview**

#### 5.32.4.2 Comparator Modes

The DBG contains three comparators, A, B and C. Each comparator compares the system address bus with the address stored in DBGXAH, DBGXAM, and DBGXAL. Furthermore, comparator A also compares the data buses to the data stored in DBGADH, DBGADL and allows masking of individual data bus bits.

All comparators are disabled in BDM and during BDM accesses.

The comparator match control logic (see Figure 65) configures comparators to monitor the buses for an exact address or an address range, whereby either an access inside or outside the specified range generates a match condition. The comparator configuration is controlled by the control register contents and the range control by the DBGC2 contents.

A match can initiate a transition to another state sequencer state (see Section 5.32.4.4, “State Sequence Control"). The comparator control register also allows the type of access to be included in the comparison through the use of the RWE, RW, SZE, and SZ bits. The RWE bit controls whether read or write comparison is enabled for the associated comparator and the RW bit selects either a read or write access for a valid match. Similarly the SZE and SZ bits allow the size of access (word or byte) to be considered in the compare. Only comparators A and B feature SZE and SZ.

The TAG bit in each comparator control register is used to determine the match condition. By setting TAG, the comparator qualifies a match with the output of opcode tracking logic and a state sequencer transition occurs when the tagged instruction reaches the CPU execution stage. Whilst tagging the RW, RWE, SZE, and SZ bits and the comparator data registers are ignored; the comparator address register must be loaded with the exact opcode address.

If the TAG bit is clear (forced type match) a comparator match is generated when the selected address appears on the system address bus. If the selected address is an opcode address, the match is generated when the opcode is fetched from the memory, which precedes the instruction execution by an indefinite number of cycles due to instruction pipelining. For a comparator match of an opcode at an odd address when TAG = 0, the corresponding even address must be contained in the comparator register. Thus for an opcode at odd address (n), the comparator register must contain address (n–1).

Once a successful comparator match has occurred, the condition which caused the original match is not verified again on subsequent matches. Thus if a particular data value is verified at a given address, this address may not still contain this data value when a subsequent match occurs.

Match[0, 1, 2] map directly to Comparators [A, B, C] respectively, except in range modes (see Section 5.32.3.2.4, “Debug Control Register2 (DBGC2)"). Comparator channel priority rules are described in the priority section (Section 5.32.4.3.4, “Channel Priorities").

#### 5.32.4.2.1 Single Address Comparator Match

With range comparisons disabled, the match condition is an exact equivalence of address bus with the value stored in the comparator address registers. Further qualification of the type of access (R/W, word/byte) and data bus contents is possible, depending on comparator channel.

##### 5.32.4.2.1.1 Comparator C

Comparator C offers only address and direction (R/W) comparison. The exact address is compared, thus with the comparator address register loaded with address (n) a word access of address (n–1) also accesses (n) but does not cause a match.

**Table 327. Comparator C Access Considerations**

| Condition For Valid Match | Comp C Address | RWE | RW | Examples |
|---|---|---|---|---|
| Read and write accesses of ADDR[n] | ADDR[n](196) | 0 | X | LDAA ADDR[n]<br>STAA #$BYTE ADDR[n] |
| Write accesses of ADDR[n] | ADDR[n] | 1 | 0 | STAA #$BYTE ADDR[n] |
| Read accesses of ADDR[n] | ADDR[n] | 1 | 1 | LDAA #$BYTE ADDR[n] |

Note:
196. A word access of ADDR[n-1] also accesses ADDR[n] but does not generate a match. The comparator address register must contain the exact address from the code.

##### 5.32.4.2.1.2 Comparator B

Comparator B offers address, direction (R/W) and access size (word/byte) comparison. If the SZE bit is set the access size (word or byte) is compared with the SZ bit value such that only the specified size of access causes a match. Thus if configured for a byte access of a particular address, a word access covering the same address does not lead to match.

Assuming the access direction is not qualified (RWE=0), for simplicity, the size access considerations are shown in Table 328.

**Table 328. Comparator B Access Size Considerations**

| Condition For Valid Match | Comp B Address | RWE | SZE | SZ8 | Examples |
|---|---|---|---|---|---|
| Word and byte accesses of ADDR[n] | ADDR[n](197) | 0 | 0 | X | MOVB #$BYTE ADDR[n]<br>MOVW #$WORD ADDR[n] |
| Word accesses of ADDR[n] only | ADDR[n] | 0 | 1 | 0 | MOVW #$WORD ADDR[n]<br>LDD ADDR[n] |

Note:
197. A word access of ADDR[n-1] also accesses ADDR[n] but does not generate a match. The comparator address register must contain the exact address from the code.

Access direction can also be used to qualify a match for Comparator B in the same way as described for Comparator C in Table 327.

#### 5.32.4.2.1.3 Comparator A

Comparator A offers address, direction (R/W), access size (word/byte) and data bus comparison.

Table 329 lists access considerations with data bus comparison. On word accesses the data byte of the lower address is mapped to DBGADH. Access direction can also be used to qualify a match for Comparator A in the same way as described for Comparator C in Table 327.

**Table 329. Comparator A Matches When Accessing ADDR[n]**

| SZE | SZ | DBGADHM, DBGADLM | Access DH=DBGADH, DL=DBGADL | Comment |
|---|---|---|---|---|
| 0 | X | $0000 | Byte<br>Word | No data bus comparison |
| 0 | X | $FF00 | Byte, data (ADDR[n])=DH<br>Word, data (ADDR[n])=DH, data(ADDR[n+1])=X | Match data (ADDR[n]) |
| 0 | X | $00FF | Word, data (ADDR[n])=X, data(ADDR[n+1])=DL | Match data(ADDR[n+1]) |
| 0 | X | $00FF | Byte, data (ADDR[n])=X, data(ADDR[n+1])=DL | Possible unintended match |
| 0 | X | $FFFF | Word, data (ADDR[n])=DH, data(ADDR[n+1])=DL | Match data (ADDR[n], ADDR[n+1]) |
| 0 | X | $FFFF | Byte, data (ADDR[n])=DH, data(ADDR[n+1])=DL | Possible unintended match |
| 1 | 0 | $0000 | Word | No data bus comparison |
| 1 | 0 | $00FF | Word, data (ADDR[n])=X, data(ADDR[n+1])=DL | Match only data at ADDR[n+1] |
| 1 | 0 | $FF00 | Word, data (ADDR[n])=DH, data(ADDR[n+1])=X | Match only data at ADDR[n] |
| 1 | 0 | $FFFF | Word, data (ADDR[n])=DH, data(ADDR[n+1])=DL | Match data at ADDR[n] & ADDR[n+1] |
| 1 | 1 | $0000 | Byte | No data bus comparison |
| 1 | 1 | $FF00 | Byte, data (ADDR[n])=DH | Match data at ADDR[n] |

#### 5.32.4.2.1.4 Comparator A Data Bus Comparison NDB Dependency

Comparator A features an NDB control bit, which allows data bus comparators to be configured to either trigger on equivalence or trigger on difference. This allows monitoring of a difference in the contents of an address location from an expected value.

When matching on an equivalence (NDB=0), each individual data bus bit position can be masked out by clearing the corresponding mask bit (DBGADHM/DBGADLM) so it is ignored in the comparison. A match occurs when all data bus bits with corresponding mask bits set are equivalent. If all mask register bits are clear, then a match is based on the address bus only, the data bus is ignored.

When matching on a difference, mask bits can be cleared to ignore bit positions. A match occurs when any data bus bit with corresponding mask bit set is different. Clearing all mask bits, causes all bits to be ignored and prevents a match because no difference can be detected. In this case address bus equivalence does not cause a match.

**Table 330. NDB and MASK Bit Dependency**

| NDB | DBGADHM[n] /DBGADLM[n] | Comment |
|---|---|---|
| 0 | 0 | Do not compare data bus bit. |
| 0 | 1 | Compare data bus bit. Match on equivalence. |
| 1 | 0 | Do not compare data bus bit. |
| 1 | 1 | Compare data bus bit. Match on difference. |

#### 5.32.4.2.2 Range Comparisons

Using the AB comparator pair for a range comparison, the data bus can also be used for qualification by using the comparator A data registers. Furthermore the DBGACTL RW and RWE bits can be used to qualify the range comparison on either a read or a write access. The corresponding DBGBCTL bits are ignored. The SZE and SZ control bits are ignored in range mode. The comparator A TAG bit is used to tag range comparisons. The comparator B TAG bit is ignored in range modes. In order for a range comparison using comparators A and B, both COMPEA and COMPEB must be set; to disable range comparisons both must be cleared. The comparator A BRK bit is used to for the AB range, the comparator B BRK bit is ignored in range mode.

When configured for range comparisons and tagging, the ranges are accurate only to word boundaries.

##### 5.32.4.2.2.1 Inside Range (CompA_Addr ≤ Address ≤ CompB_Addr)

In the Inside Range comparator mode, comparator pair A and B can be configured for range comparisons. This configuration depends upon the control register (DBGC2). The match condition requires a valid match for both comparators happens on the same bus cycle. A match condition on only one comparator is not valid. An aligned word access which straddles the range boundary is valid only if the aligned address is inside the range.

##### 5.32.4.2.2.2 Outside Range (address < CompA_Addr or Address > CompB_Addr)

In the Outside Range comparator mode, comparator pair A and B can be configured for range comparisons. A single match condition on either of the comparators is recognized as valid. An aligned word access which straddles the range boundary is valid only if the aligned address is outside the range.

Outside range mode in combination with tagging can be used to detect if the opcode fetches are from an unexpected range. In forced match mode the outside range match would typically be activated at any interrupt vector fetch or register access. This can be avoided by setting the upper range limit to $3FFFF or lower range limit to $00000 respectively.

### 5.32.4.3 Match Modes (Forced or Tagged)

Match modes are used as qualifiers for a state sequencer change of state. The Comparator control register TAG bits select the match mode. The modes are described in the following sections.

#### 5.32.4.3.1 Forced Match

When configured for forced matching, a comparator channel match can immediately initiate a transition to the next state sequencer state whereby the corresponding flags in DBGSR are set. The state control register for the current state determines the next state. Forced matches are typically generated 2-3 bus cycles after the final matching address bus cycle, independent of comparator RWE/RW settings. Furthermore since opcode fetches occur several cycles before the opcode execution a forced match of an opcode address typically precedes a tagged match at the same address.

#### 5.32.4.3.2 Tagged Match

If a CPU taghit occurs a transition to another state sequencer state is initiated and the corresponding DBGSR flags are set. For a comparator related taghit to occur, the DBG must first attach tags to instructions as they are fetched from memory. When the tagged instruction reaches the execution stage of the instruction queue a taghit is generated by the CPU. This can initiate a state sequencer transition.

#### 5.32.4.3.3 Immediate Trigger

Independent of comparator matches it is possible to initiate a tracing session and/or breakpoint by writing to the TRIG bit in DBGC1. If configured for begin aligned tracing, this triggers the state sequencer into the Final State, if configured for end alignment, setting the TRIG bit disarms the module, ending the session and issues a forced breakpoint request to the CPU.

It is possible to set both TRIG and ARM simultaneously to generate an immediate trigger, independent of the current state of ARM.

#### 5.32.4.3.4 Channel Priorities

In case of simultaneous matches the priority is resolved according to Table 331. The lower priority is suppressed. It is thus possible to miss a lower priority match if it occurs simultaneously with a higher priority. The priorities described in Table 331 dictate in the case of simultaneous matches, the match pointing to final state has highest priority followed by the lower channel number (0,1,2).

**Table 331. Channel Priorities**

| Priority | Source | Action |
|---|---|---|
| Highest | TRIG | Enter Final State |
| | Channel pointing to Final State | Transition to next state as defined by state control registers |
| | Match0 (force or tag hit) | Transition to next state as defined by state control registers |
| | Match1 (force or tag hit) | Transition to next state as defined by state control registers |
| Lowest | Match2 (force or tag hit) | Transition to next state as defined by state control registers |

### 5.32.4.4 State Sequence Control

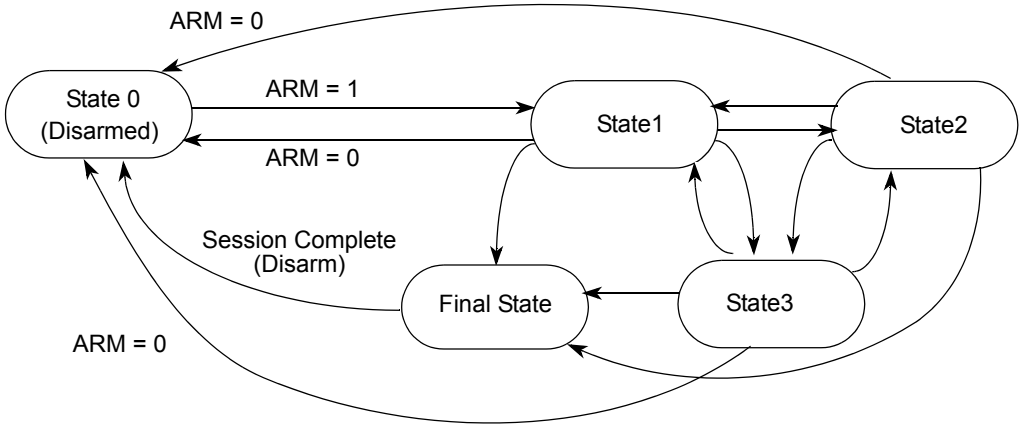



**Figure 66. State Sequencer Diagram**

The state sequencer allows a defined sequence of events to provide a trigger point for tracing of data in the trace buffer. Once the DBG module has been armed by setting the ARM bit in the DBGC1 register, then state1 of the state sequencer is entered. Further transitions between the states are then controlled by the state control registers and channel matches. From Final State the only permitted transition is back to the disarmed state0. Transition between any of the states 1 to 3 is not restricted. Each transition updates the SSF[2:0] flags in DBGSR accordingly to indicate the current state.

Alternatively writing to the TRIG bit in DBGSC1, provides an immediate trigger independent of comparator matches.

Independent of the state sequencer, each comparator channel can be individually configured to generate an immediate breakpoint when a match occurs through the use of the BRK bits in the DBGxCTL registers. Thus it is possible to generate an immediate breakpoint on selected channels, whilst a state sequencer transition can be initiated by a match on other channels. If a debug session is ended by a match on a channel the state sequencer transitions through Final State for a clock cycle to state0. This is independent of tracing and breakpoint activity, thus with tracing and breakpoints disabled, the state sequencer enters state0 and the debug module is disarmed.

#### 5.32.4.4.1 Final State

On entering Final State a trigger may be issued to the trace buffer according to the trace alignment control as defined by the TALIGN bit (see Section 5.32.3.2.3, “Debug Trace Control Register (DBGTCR)"). If the TSOURCE bit in DBGTCR is clear then the trace buffer is disabled and the transition to Final State can only generate a breakpoint request. In this case or upon completion of a tracing session when tracing is enabled, the ARM bit in the DBGC1 register is cleared, returning the module to the disarmed state0. If tracing is enabled a breakpoint request can occur at the end of the tracing session. If neither tracing nor breakpoints are enabled then when the final state is reached it returns automatically to state0 and the debug module is disarmed.

### 5.32.4.5 Trace Buffer Operation

The trace buffer is a 64 lines deep by 20-bits wide RAM array. The DBG module stores trace information in the RAM array in a circular buffer format. The system accesses the RAM array through a register window (DBGTBH:DBGTBL) using 16-bit wide word accesses. After each complete 20-bit trace buffer line is read, an internal pointer into the RAM increments so the next read receives fresh information. Data is stored in the format shown in Table 332 and Table 336. After each store the counter register DBGCNT is incremented. Tracing of CPU activity is disabled when the BDM is active. Reading the trace buffer whilst the DBG is armed returns invalid data and the trace buffer pointer is not incremented.

#### 5.32.4.5.1 Trace Trigger Alignment

Using the TALIGN bit (see Section 5.32.3.2.3, “Debug Trace Control Register (DBGTCR)") it is possible to align the trigger with the end or the beginning of a tracing session.

If end alignment is selected, tracing begins when the ARM bit in DBGC1 is set and State1 is entered; the transition to Final State signals the end of the tracing session. Tracing with Begin-Trigger starts at the opcode of the trigger. Using end alignment or when the tracing is initiated by writing to the TRIG bit whilst configured for begin alignment, tracing starts in the second cycle after the DBGC1 write cycle.

##### 5.32.4.5.1.1 Storing with Begin Trigger Alignment

Storing with begin alignment, data is not stored in the Trace Buffer until the Final State is entered. Once the trigger condition is met the DBG module remains armed until 64 lines are stored in the Trace Buffer. If the trigger is at the address of the change-of-flow instruction the change of flow associated with the trigger is stored in the Trace Buffer. Using begin alignment together with tagging, if the tagged instruction is about to be executed then the trace is started. Upon completion of the tracing session the breakpoint is generated, thus the breakpoint does not occur at the tagged instruction boundary.

##### 5.32.4.5.1.2 Storing with End Trigger Alignment

Storing with end alignment, data is stored in the Trace Buffer until the Final State is entered, at which point the DBG module becomes disarmed and no more data is stored. If the trigger is at the address of a change of flow instruction, the trigger event is not stored in the Trace Buffer. If all trace buffer lines have been used before a trigger event occurs then the trace continues at the first line, overwriting the oldest entries.

#### 5.32.4.5.2 Trace Modes

Four trace modes are available. The mode is selected using the TRCMOD bits in the DBGTCR register. Tracing is enabled using the TSOURCE bit in the DBGTCR register. The modes are described in the following subsections.

##### 5.32.4.5.2.1 Normal Mode

In Normal Mode, change of flow (COF) program counter (PC) addresses are stored.

COF addresses are defined as follows:

- Source address of taken conditional branches (long, short, bit-conditional, and loop primitives)
- Destination address of indexed JMP, JSR, and CALL instruction
- Destination address of RTI, RTS, and RTC instructions
- Vector address of interrupts, except for BDM vectors

LBRA, BRA, BSR, BGND as well as non-indexed JMP, JSR, and CALL instructions are not classified as change of flow and are not stored in the trace buffer.

Stored information includes the full 18-bit address bus and information bits, which contains a source/destination bit to indicate whether the stored address was a source address or destination address.

**NOTE**

When a COF instruction with destination address is executed, the destination address is stored to the trace buffer on instruction completion, indicating the COF has taken place. If an interrupt occurs simultaneously then the next instruction carried out is actually from the interrupt service routine. The instruction at the destination address of the original program flow gets executed after the interrupt service routine.

In the following example an IRQ interrupt occurs during execution of the indexed JMP at address MARK1. The BRN at the destination (SUB_1) is not executed until after the IRQ service routine but the destination address is entered into the trace buffer to indicate the indexed JMP COF has taken place.

```
        LDX     #SUB_1
MARK1   JMP     0,X                     ; IRQ interrupt occurs during execution of this
MARK2   NOP                             ;

SUB_1   BRN     *                       ; JMP Destination address TRACE BUFFER ENTRY 1
                                        ; RTI Destination address TRACE BUFFER ENTRY 3
        NOP                             ;
ADDR1   DBNE    A,PART5                 ; Source address TRACE BUFFER ENTRY 4

IRQ_ISR LDAB    #$F0                    ; IRQ Vector $FFF2 = TRACE BUFFER ENTRY 2
        STAB    VAR_C1
        RTI                             ;
```

The execution flow taking into account the IRQ is as follows

```
        LDX     #SUB_1
MARK1   JMP     0,X                     ;
IRQ_ISR LDAB    #$F0                    ;
        STAB    VAR_C1
        RTI                             ;
SUB_1   BRN     *
        NOP                             ;
ADDR1   DBNE    A,PART5                 ;
```

#### 5.32.4.5.2.2 Loop1 Mode

Loop1 Mode, similarly to Normal Mode also stores only COF address information to the trace buffer, it however allows the filtering out of redundant information.

The intent of Loop1 Mode is to prevent the Trace Buffer from being filled entirely with duplicate information from a looping construct such as delays using the DBNE instruction or polling loops using BRSET/BRCLR instructions. Immediately after address information is placed in the Trace Buffer, the DBG module writes this value into a background register. This prevents consecutive duplicate address entries in the Trace Buffer resulting from repeated branches.

Loop1 Mode only inhibits consecutive duplicate source address entries which would typically be stored in most tight looping constructs. It does not inhibit repeated entries of destination addresses or vector addresses, since repeated entries of these would most likely indicate a bug in the user's code which the DBG module is designed to help find.

#### 5.32.4.5.2.3 Detail Mode

In Detail Mode, address and data for all memory and register accesses is stored in the trace buffer. This mode is intended to supply additional information on indexed, indirect addressing modes where storing only the destination address would not provide all information required for a user to determine where the code is in error. This mode also features information bit storage to the trace buffer, for each address byte storage. The information bits indicate the size of access (word or byte) and the type of access (read or write).

When tracing in Detail Mode, all cycles are traced except those when the CPU is either in a free or opcode fetch cycle.

#### 5.32.4.5.2.4 Compressed Pure PC Mode

In Compressed Pure PC Mode, the PC addresses of all executed opcodes, including illegal opcodes are stored. A compressed storage format is used to increase the effective depth of the trace buffer. This is achieved by storing the lower order bits each time and using 2 information bits to indicate if a 64 byte boundary has been crossed, in which case the full PC is stored.

Each Trace Buffer row consists of 2 information bits and 18 PC address bits

**NOTE:**

When tracing is terminated using forced breakpoints, latency in breakpoint generation means opcodes following the opcode causing the breakpoint can be stored to the trace buffer. The number of opcodes is dependent on program flow. This can be avoided by using tagged breakpoints.

#### 5.32.4.5.3 Trace Buffer Organization (Normal, Loop1, Detail modes)

ADRH, ADRM, ADRL denote address high, middle and low byte respectively. The numerical suffix refers to the tracing count. The information format for Loop1 and Normal modes is identical. In Detail mode, the address and data for each entry are stored on consecutive lines, thus the maximum number of entries is 32. In this case DBGCNT bits are incremented twice, once for the address line and once for the data line, on 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 332. 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 |

#### 5.32.4.5.3.1 Information Bit Organization

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

**Field2 Bits in Detail Mode**

**Table 333. Field2 Bits in Detail Mode**

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

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

**Table 334. 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. |

**Field2 Bits in Normal and Loop1 Modes**

**Figure 67. Information Bits PCH**

| Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|
| CSD | CVA | PC17 | PC16 |

**Table 335. PCH Field Descriptions**

| Bit | Description |
|---|---|
| 3<br>CSD | **Source Destination Indicator** — In Normal and Loop1 mode this bit indicates if the corresponding stored address is a source or destination address. This bit has no meaning in Compressed Pure PC mode.<br>0 Source Address<br>1 Destination Address |
| 2<br>CVA | **Vector Indicator** — In Normal and Loop1 mode 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 bit has no meaning in Compressed Pure PC mode.<br>0 Non-Vector Destination Address<br>1 Vector Destination Address |
| 1<br>PC17 | **Program Counter bit 17** — In Normal and Loop1 mode this bit corresponds to program counter bit 17. |
| 0<br>PC16 | **Program Counter bit 16** — In Normal and Loop1 mode this bit corresponds to program counter bit 16. |

#### 5.32.4.5.4 Trace Buffer Organization (Compressed Pure PC mode)

**Table 336. Trace Buffer Organization Example (Compressed PurePC mode)**

| Mode | Line Number | 2-bits | 6-bits | 6-bits | 6-bits |
|---|---|---|---|---|---|
| | | Field 3 | Field 2 | Field 1 | Field 0 |
| Compressed Pure PC Mode | Line 1 | 00 | PC1 (Initial 18-bit PC Base Address) | | |
| | Line 2 | 11 | PC4 | PC3 | PC2 |
| | Line 3 | 01 | 0 | 0 | PC5 |
| | Line 4 | 00 | PC6 (New 18-bit PC Base Address) | | |
| | Line 5 | 10 | 0 | PC8 | PC7 |
| | Line 6 | 00 | PC9 (New 18-bit PC Base Address) | | |

**NOTE**

Configured for end aligned triggering in compressed PurePC mode, then after rollover it is possible the oldest base address is overwritten. In this case all entries between the pointer and the next base address have lost their base address following rollover. For example in Table 337 if one line of rollover has occurred, Line 1, PC1, is overwritten with a new entry. Thus the entries on Lines 2 and 3 have lost their base address. For reconstruction of program flow the first base address following the pointer must be used, in the example, Line 4. The pointer points to the oldest entry, Line 2.

#### 5.32.4.5.5 Field3 Bits in Compressed Pure PC Modes

**Table 337. Compressed Pure PC Mode Field 3 Information Bit Encoding**

| INF1 | INF0 | TRACE BUFFER ROW CONTENT |
|---|---|---|
| 0 | 0 | Base PC address TB[17:0] contains a full PC[17:0] value |
| 0 | 1 | Trace Buffer[5:0] contain incremental PC relative to base address zero value |
| 1 | 0 | Trace Buffer[11:0] contain next 2 incremental PCs relative to base address zero value |
| 1 | 1 | Trace Buffer[17:0] contain next 3 incremental PCs relative to base address zero value |

Each time that PC[17:6] differs from the previous base PC[17:6], then a new base address is stored. The base address zero value is the lowest address in the 64 address range. The first line of the trace buffer always gets a base PC address, this applies also on rollover.

#### 5.32.4.5.6 Reading Data from Trace Buffer

The data stored in the Trace Buffer can be read provided the DBG module is not armed, is configured for tracing (TSOURCE bit is set) and the system not secured. When the ARM bit is written to 1 the trace buffer is locked to prevent reading. The trace buffer can only be unlocked for reading by a single aligned word write to DBGTB when the module is disarmed.

The Trace Buffer can only be read through the DBGTB register using aligned word reads, any byte or misaligned reads return 0 and do not cause the trace buffer pointer to increment to the next trace buffer address. The Trace Buffer data is read out first-in first-out. By reading CNT in DBGCNT the number of valid lines can be determined. DBGCNT does not decrement as data is read.

Whilst reading an internal pointer is used to determine the next line to be read. After a tracing session, the pointer points to the oldest data entry, thus if no rollover has occurred, the pointer points to line0, otherwise it points to the line with the oldest entry. In compressed Pure PC mode on rollover the line with the oldest data entry may also contain newer data entries in fields 0 and 1. Thus if rollover is indicated by the TBF bit, the line status must be decoded using the INF bits in field3 of this line. If both INF bits are clear then the line contains only entries from before the last rollover.
If INF0=1 then field 0 contains post rollover data but fields 1 and 2 contain pre rollover data.
If INF1=1 then fields 0 and 1 contain post rollover data but field 2 contains pre rollover data.

The pointer is initialized by each aligned write to DBGTBH to point to the oldest data again. This enables an interrupted trace buffer read sequence to be easily restarted from the oldest data entry.

The least significant word of line is read out first. This corresponds to the fields 1 and 0 of Table 332. The next word read returns field 2 in the least significant bits [3:0] and “0” for bits [15:4].

Reading the Trace Buffer while the DBG module is armed returns invalid data and no shifting of the RAM pointer occurs.

#### 5.32.4.5.7 Trace Buffer Reset State

The Trace Buffer contents and DBGCNT bits are not initialized by a system reset. Thus should a system reset occur, the trace session information from immediately before the reset occurred can be read out and the number of valid lines in the trace buffer is indicated by DBGCNT. The internal pointer to the current trace buffer address is initialized by unlocking the trace buffer and points to the oldest valid data even if a reset occurred during the tracing session. To read the trace buffer after a reset, TSOURCE must be set, otherwise the trace buffer reads as all zeroes. Generally debugging occurrences of system resets is best handled using end trigger alignment since the reset may occur before the trace trigger, which in the begin trigger alignment case means no information would be stored in the trace buffer.

The Trace Buffer contents and DBGCNT bits are undefined following a POR.

**NOTE**

An external pin RESET occurs simultaneous to a trace buffer entry can, in very seldom cases, lead to either this entry being corrupted or the first entry of the session being corrupted. In such cases the other contents of the trace buffer still contain valid tracing information. The case occurs when the reset assertion coincides with the trace buffer entry clock edge.

### 5.32.4.6 Tagging

A tag follows program information as it advances through the instruction queue. When a tagged instruction reaches the head of the queue a tag hit occurs and can initiate a state sequencer transition.

Each comparator control register features a TAG bit, which controls whether the comparator match causes a state sequencer transition immediately or tags the opcode at the matched address. If a comparator is enabled for tagged comparisons, the address stored in the comparator match address registers must be an opcode address.

Using Begin trigger together with tagging, if the tagged instruction is about to be executed then the transition to the next state sequencer state occurs. If the transition is to the Final State, tracing is started. Only upon completion of the tracing session can a breakpoint be generated. Using End alignment, when the tagged instruction is about to be executed and the next transition is to Final State then a breakpoint is generated immediately, before the tagged instruction is carried out.

R/W monitoring, access size (SZ) monitoring and data bus monitoring are not useful if tagging is selected, since the tag is attached to the opcode at the matched address and is not dependent on the data bus nor on the type of access. Thus these bits are ignored if tagging is selected.

When configured for range comparisons and tagging, the ranges are accurate only to word boundaries.

Tagging is disabled when the BDM becomes active.

### 5.32.4.7 Breakpoints

It is possible to generate breakpoints from channel transitions to final state or using software to write to the TRIG bit in the DBGC1 register.

#### 5.32.4.7.1 Breakpoints From Comparator Channels

Breakpoints can be generated when the state sequencer transitions to the Final State. If configured for tagging, then the breakpoint is generated when the tagged opcode reaches the execution stage of the instruction queue.

If a tracing session is selected by the TSOURCE bit, breakpoints are requested when the tracing session has completed, thus if Begin aligned triggering is selected, the breakpoint is requested only on completion of the subsequent trace (see Table 338). If no tracing session is selected, breakpoints are requested immediately.

If the BRK bit is set, then the associated breakpoint is generated immediately independent of tracing trigger alignment.

**Table 338. Breakpoint Setup For CPU Breakpoints**

| BRK | TALIGN | DBGBRK | Breakpoint Alignment |
|---|---|---|---|
| 0 | 0 | 0 | Fill Trace Buffer until trigger then disarm (no breakpoints) |
| 0 | 0 | 1 | Fill Trace Buffer until trigger, then breakpoint request occurs |
| 0 | 1 | 0 | Start Trace Buffer at trigger (no breakpoints) |
| 0 | 1 | 1 | Start Trace Buffer at trigger. A breakpoint request occurs when Trace Buffer is full |
| 1 | x | 1 | Terminate tracing and generate breakpoint immediately on trigger |
| 1 | x | 0 | Terminate tracing immediately on trigger |

#### 5.32.4.7.2 Breakpoints Generated Via The TRIG Bit

If a TRIG triggers occur, the Final State is entered whereby tracing trigger alignment is defined by the TALIGN bit. If a tracing session is selected by the TSOURCE bit, breakpoints are requested when the tracing session has completed, thus if Begin aligned triggering is selected, the breakpoint is requested only on completion of the subsequent trace (see Table 338). If no tracing session is selected, breakpoints are requested immediately. TRIG breakpoints are possible with a single write to DBGC1, setting ARM and TRIG simultaneously.

#### 5.32.4.7.3 Breakpoint Priorities

If a TRIG trigger occurs after Begin aligned tracing has already started, then the TRIG no longer has an effect. When the associated tracing session is complete, the breakpoint occurs. Similarly if a TRIG is followed by a subsequent comparator channel match, it has no effect, since tracing has already started.

If a forced SWI breakpoint coincides with a BGND in user code with BDM enabled, then the BDM is activated by the BGND and the breakpoint to SWI is suppressed.

##### 5.32.4.7.3.1 DBG Breakpoint Priorities And BDM Interfacing

Breakpoint operation is dependent on the state of the BDM module. If the BDM module is active, the CPU is executing out of BDM firmware, thus comparator matches and associated breakpoints are disabled. In addition, while executing a BDM TRACE command, tagging into BDM is disabled. If BDM is not active, the breakpoint gives priority to BDM requests over SWI requests if the breakpoint happens to coincide with a SWI instruction in user code. On returning from BDM, the SWI from user code gets executed.

**Table 339. Breakpoint Mapping Summary**

| DBGBRK | BDM Bit (DBGC1[4]) | BDM Enabled | BDM Active | Breakpoint Mapping |
|---|---|---|---|---|
| 0 | X | X | X | No Breakpoint |
| 1 | 0 | X | 0 | Breakpoint to SWI |
| X | X | 1 | 1 | No Breakpoint |
| 1 | 1 | 0 | X | Breakpoint to SWI |
| 1 | 1 | 1 | 0 | Breakpoint to BDM |

BDM cannot be entered from a breakpoint unless the ENABLE bit is set in the BDM. If entry to BDM via a BGND instruction is attempted and the ENABLE bit in the BDM is cleared, the CPU actually executes the BDM firmware code, checks the ENABLE and returns if ENABLE is not set. If not serviced by the monitor then the breakpoint is re-asserted when the BDM returns to normal CPU flow.

If the comparator register contents coincide with the SWI/BDM vector address then an SWI in user code could coincide with a DBG breakpoint. The CPU ensures BDM requests have a higher priority than SWI requests. Returning from the BDM/SWI service routine care must be taken to avoid a repeated breakpoint at the same address.

Should a tagged or forced breakpoint coincide with a BGND in user code, then the instruction which follows the BGND instruction is the first instruction executed when normal program execution resumes.

**NOTE**

When program control returns from a tagged breakpoint using an RTI or BDM GO command without program counter modification it returns to the instruction whose tag generated the breakpoint. To avoid a repeated breakpoint at the same location reconfigure the DBG module in the SWI routine, if configured for an SWI breakpoint, or over the BDM interface by executing a TRACE command before the GO to increment the program flow past the tagged instruction.

### 5.32.5 Application Information

#### 5.32.5.1 State Machine scenarios

Defining the state control registers as SCR1,SCR2, SCR3 and M0,M1,M2 as matches on channels 0,1,2 respectively. SCR encoding supported by S12SDBGV1 are shown in black. SCR encoding supported only in S12SDBGV2 are shown in red. For backwards compatibility the new scenarios use a 4th bit in each SCR register. Thus the existing encoding for SCRx[2:0] is not changed.

#### 5.32.5.2 Scenario 1

A trigger is generated if a given sequence of 3 code events is executed.

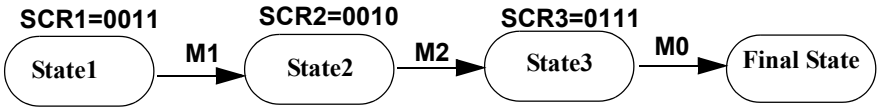



**Figure 68. Scenario 1**

Scenario 1 is possible with S12SDBGV1 SCR encoding

#### 5.32.5.3 Scenario 2

A trigger is generated if a given sequence of 2 code events is executed.




**Figure 69. Scenario 2a**

A trigger is generated if a given sequence of 2 code events is executed, whereby the first event is entry into a range (COMPA,COMPB configured for range mode). M1 is disabled in range modes.




**Figure 70. Scenario 2b**

A trigger is generated if a given sequence of 2 code events is executed, whereby the second event is entry into a range (COMPA,COMPB configured for range mode)




**Figure 71. Scenario 2c**

All 3 scenarios 2a,2b,2c are possible with the S12SDBGV1 SCR encoding

#### 5.32.5.4 Scenario 3

A trigger is generated immediately when one of up to 3 given events occurs

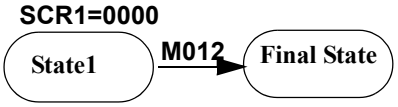



**Figure 72. Scenario 3**

Scenario 3 is possible with S12SDBGV1 SCR encoding

#### 5.32.5.5 Scenario 4

Trigger if a sequence of 2 events is carried out in an incorrect order. Event A must be followed by event B and event B must be followed by event A. 2 consecutive occurrences of event A without an intermediate event B cause a trigger. Similarly 2 consecutive occurrences of event B without an intermediate event A cause a trigger. This is possible by using CompA and CompC to match on the same address as shown.

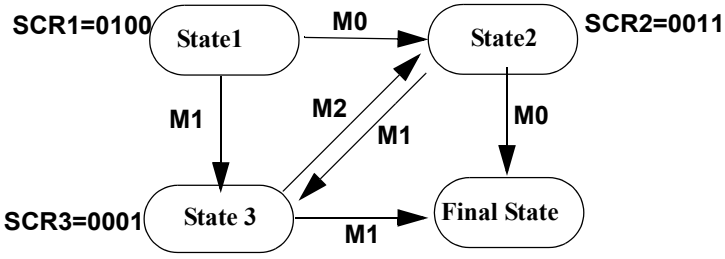



**Figure 73. Scenario 4a**

This scenario is currently not possible using 2 comparators only. S12SDBGV2 makes it possible with 2 comparators, State 3 allowing a M0 to return to state 2, whilst a M2 leads to final state as shown.

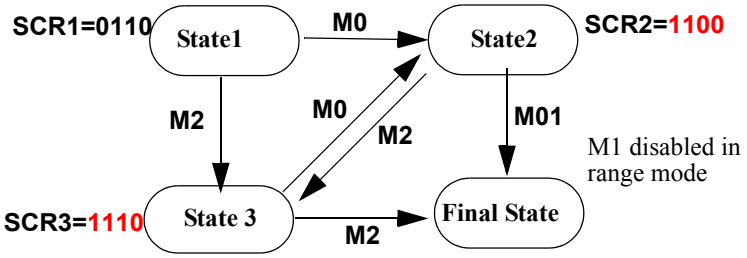



**Figure 74. Scenario 4b (with 2 comparators)**

The advantage of using only 2 channels is now range comparisons can be included (channel0)

This however violates the S12SDBGV1 specification, which states a match leading to final state always has priority in case of a simultaneous match, whilst priority is also given to the lowest channel number. For S12SDBG the corresponding CPU priority decoder is removed to support this, such that on simultaneous taghits, taghits pointing to final state have highest priority. If no taghit points to final state then the lowest channel number has priority. Thus with the above encoding from State3, the CPU and DBG would break on a simultaneous M0/M2.

### 5.32.5.6 Scenario 5

Trigger if following event A, event C precedes event B. i.e. the expected execution flow is A->B->C.




**Figure 75. Scenario 5**

Scenario 5 is possible with the S12SDBGV1 SCR encoding

### 5.32.5.7 Scenario 6

Trigger if event A occurs twice in succession before any of 2 other events (BC) occurs. This scenario is not possible using the S12SDBGV1 SCR encoding. S12SDBGV2 includes additions shown in red. The change in SCR1 encoding also has the advantage that a State1->State3 transition using M0 is now possible. This is advantageous because range and data bus comparisons use channel0 only.




**Figure 76. Scenario 6**

### 5.32.5.8 Scenario 7

Trigger when a series of 3 events is executed out of order. Specifying the event order as M1,M2,M0 to run in loops (120120120). Any deviation from this order should trigger. This scenario is not possible using the S12SDBGV1 SCR encoding because OR possibilities are very limited in the channel encoding. By adding OR forks as shown in red this scenario is possible.

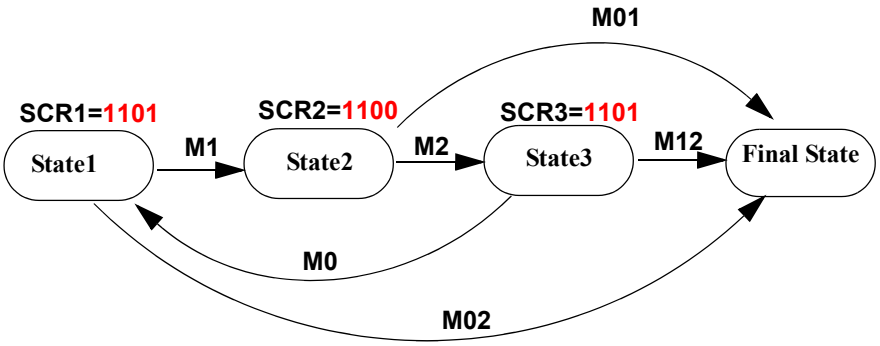



**Figure 77. Scenario 7**

On simultaneous matches the lowest channel number has priority so with this configuration the forking from State1 has the peculiar effect that a simultaneous match0/match1 transitions to final state, but a simultaneous match2/match1transitions to state2.

### 5.32.5.9 Scenario 8

Trigger when a routine/event at M2 follows either M1 or M0.




**Figure 78. Scenario 8a**

Trigger when an event M2 is followed by either event M0 or event M1




**Figure 79. Scenario 8b**

Scenario 8a and 8b are possible with the S12SDBGV1 and S12SDBGV2 SCR encoding.

#### 5.32.5.10 Scenario 9

Trigger when a routine/event at A (M2) does not follow either B or C (M1 or M0) before they are executed again. This cannot be realized with theS12SDBGV1 SCR encoding due to OR limitations. By changing the SCR2 encoding as shown in red this scenario becomes possible.




**Figure 80. Scenario 9**

#### 5.32.5.11 Scenario 10

Trigger if an event M0 occurs following up to two successive M2 events without the resetting event M1. As shown up to 2 consecutive M2 events are allowed, whereby a reset to State1 is possible after either one or two M2 events. If an event M0 occurs following the second M2, before M1 resets to State1 then a trigger is generated. Configuring CompA and CompC the same, it is possible to generate a breakpoint on the third consecutive occurrence of event M0 without a reset M1.

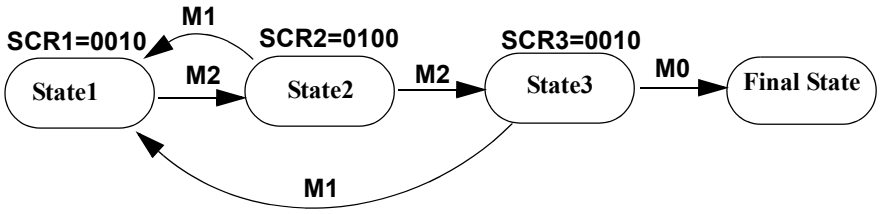



**Figure 81. Scenario 10a**

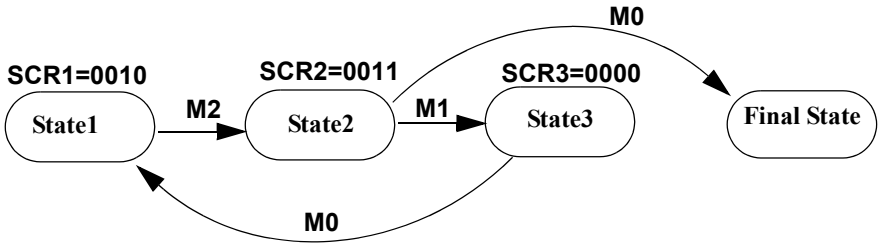



**Figure 82. Scenario 10b**

Scenario 10b shows the case that after M2 then M1 must occur before M0. Starting from a particular point in code, event M2 must always be followed by M1 before M0. If after any M2, event M0 occurs before M1 then a trigger is generated.

## 5.33 Security (S12X9SECV2)

### 5.33.1 Introduction

This specification describes the function of the security mechanism in the S12I chip family (9SEC).

**NOTE**

No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH and/or EEPROM difficult for unauthorized users.

### 5.33.1.1 Features

The user must be reminded part of the security must lie with the application code. An extreme example would be application code which dumps the contents of the internal memory. This would defeat the purpose of security. At the same time, the user may also wish to put a back door in the application program. An example of this is the user downloads a security key through the SCI, which allows access to a programming routine which updates parameters stored in another section of the Flash memory.

The security features of the S12I chip family (in secure mode) are:

- Protect the content of non-volatile memories (Flash, EEPROM)
- Execution of NVM commands is restricted
- Disable access to internal memory via background debug module (BDM)

### 5.33.1.2 Modes of Operation

Table 340 gives an overview over availability of security relevant features in unsecure and secure modes.

Figure 83 shows all modules affected by security in an MCU.

**Table 340. Feature Availability in Unsecure and Secure Modes on S12I**

| | Unsecure Mode | | | | | | Secure Mode | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NS | SS | NX | ES | EX | ST | NS | SS | NX | ES | EX | ST |

Note:

198. Restricted NVM command set only. Refer to the NVM wrapper block guides for detailed information.

199. BDM hardware commands restricted to peripheral registers only.

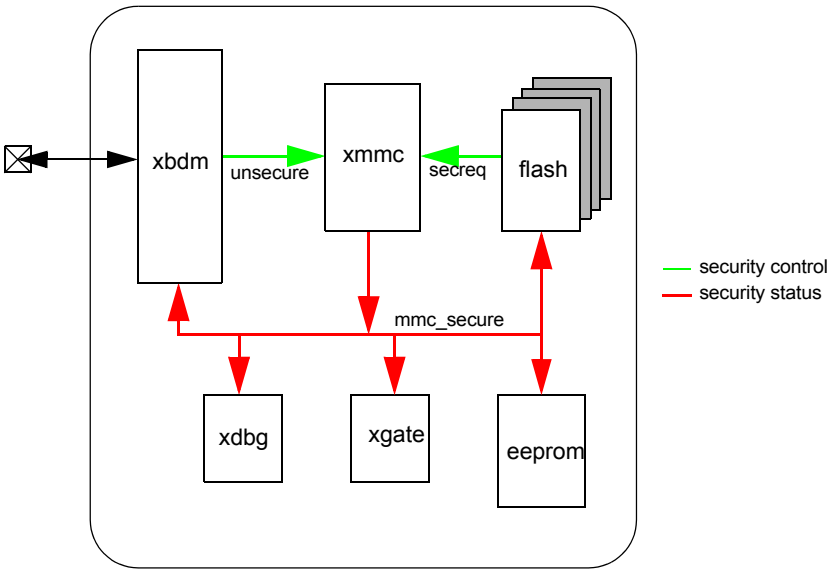



**Figure 83. Chip Security Block Diagram**

The security mechanism relies on non-volatile bits contained in the FLASH module. The state of these bits is passed to the S12XMMC. Several of the MCU modules are involved in blocking certain operations which would reveal the contents of the protected FLASH and EEPROM.

## 5.34 Impact on MCU modules

When the device is in secure mode, the following blocks are affected by security.

### 5.34.1 MMC

There is a signal called “secure” from the FLASH or EEPROM which indicates if the security is enabled. There is also a signal from the xbdm, which is used in the process of unsecuring the chip. This signal is called “unsecure”. These two signals and the resulting state of “device security” are shown in Table 341.

**Table 341. : Security Bits - System Control**

| secreq | bdm_unsecure | mmc_secure |
|---|---|---|
| 0 | 0 | 0 (unsecured) |
| 0 | 1 | 0 (unsecured) |
| 1 | 0 | 1 (secured) |
| 1 | 1 | 0 (unsecured) |

In expanded modes, if the “mmc_secure_t2” signal is asserted, the ROMON and EEON bits are forced to zero. This operation is independent of how the part got to expanded mode (straight out of reset or by writing the mode register).

When security is enabled and the part is brought up in special single chip mode, the secure BDM firmware is brought into the map along with the standard BDM firmware. The secure firmware has higher priority, but does not fill the whole space. It occupies $7F_FF80 to $7F_FFFF.

One cycle after bdm_unsecure is asserted the secure firmware is disabled from the map.

In secure mode aBDM access to a non register address is translated to a peripheral register address, and BDM registers are not accessible.

No BDM global access is possible if the chip is secured.

In secured expanded mode or emulation mode, FLASH and EEPROM are disabled by the MMC.

### 5.34.2 BDM

When security is active and the blank check is performed and failed, only BDM hardware commands are available. If the blank check is succeeds, all BDM commands are available.

The BDM status register contains a bit called UNSEC. This bit is only writable by the secure firmware in special single chip mode. Based on the state of this bit, the BDM generates a signal called “unsecure”. The bit and signal are always reset to 0 (= de-asserted = secure).

If the user resets into special single chip mode with the part secured, an alternate BDM firmware (“SECURE firmware”), is placed in the map along with the standard BDM firmware. The secure firmware has higher priority than the standard firmware, but it is smaller (less bytes). The secure firmware covers the vector space, but does not reach the beginning of the BDM firmware space.

When blank check is successfully performed, UNSEC is asserted. The BDM program jumps to the start of the standard BDM firmware program and the secure firmware is turned off. If the blank check fails, then the ENBDM bit in the BDMSTS register is set without asserting UNSEC, and the BDM firmware code enters a loop. This enables the BDM hardware commands. In secure mode the MMC restricts BDM accesses to the register space.

With UNSEC asserted, security is off and the user can change the state of the secure bits in the FLASH. Note that if the user does not change the state of these bits to “unsecured”, the part is secured again when it is next taken out of reset.

### 5.34.3 DBG

S12X_DBG disables the trace buffer, but breakpoints are still valid.

### 5.34.4 XGATE

XGATE internal registers XGCCR, XGPC, and XGR1 - XGR7 can not be written and reads zero from IPBI.

Single stepping in XGATE is not possible.

XGATE code residing in the internal RAM cannot be protected:

1. start MCU in NSC, let it run for a while
2. reset into SSC, MASERS the NVM
3. reset into SSC, blank check of BDM secure firmware succeeds
4. MCU is temporarily unsecured
5. BDM can be used to read internal RAM (contents not affected by reset)

## 5.35 Secure firmware Code Overview

The BDM contains a secure firmware code. This firmware code is invoked when the user comes out of reset in special single chip mode with security enabled. The function of the firmware code is straight forward:

- Verify the FLASH is erased
- Verify the EEPROM is erased
- If both are erased, release security

If either the FLASH or the EEPROM is not erased, then security is not released. The ENBDM bit is set and the code enters a loop. This allows BDM hardware commands, which may be used to erase the EEPROM and FLASH.

Note that erasing the memories and erasing / reprogramming the security bits is NOT part of the firmware code. The user must perform these operations.

The blank check of FLASH and EEPROM is done in the BDM firmware. As such it could be changed on future parts. The current scheme uses the NVM command state-machines (FTX, EETX) to perform the blank check.

### 5.35.0.1 Securing the Microcontroller

Once the user has programmed the Flash and EEPROM, the chip can be secured by programming the security bits located in the options/security byte in the Flash memory array. These non-volatile bits keeps the device secured through reset and power-down.

The options/security byte is located at address 0xFF0F (= global address 0x7F_FF0F) in the Flash memory array. This byte can be erased and programmed like any other Flash location. Two bits of this byte are used for security (SEC[1:0]). On devices which have a memory page window, the Flash options/security byte is also available at address 0xBF0F by selecting page 0x3F with the PPAGE register. The contents of this byte are copied into the Flash security register (FSEC) during a reset sequence.

**Table 342. Flash Options/Security Byte**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0xFF0F | KEYEN1 | KEYEN0 | NV5 | NV4 | NV3 | NV2 | SEC1 | SEC0 |

The meaning of the bits KEYEN[1:0] is shown in Table 343. Refer to Section 5.35.0.2.5, “Unsecuring the MCU Using the Back door Key Access" for more information.

**Table 343. Back door Key Access Enable Bits**

| KEYEN[1:0] | Back door Key Access Enabled |
|---|---|
| 00 | 0 (disabled) |
| 01 | 0 (disabled) |
| 10 | 1 (enabled) |
| 11 | 0 (disabled) |

The meaning of the security bits SEC[1:0] is shown in Table 344. For security reasons, the state of device security is controlled by two bits. To put the device in unsecured mode, these bits must be programmed to SEC[1:0] = ‘10’. All other combinations put the device in a secured mode. The recommended value to put the device in secured state is the inverse of the unsecured state, i.e. SEC[1:0] = ‘01’.

**Table 344. Security Bits**

| SEC[1:0] | Security State |
|---|---|
| 00 | 1 (secured) |
| 01 | 1 (secured) |
| 10 | 0 (unsecured) |
| 11 | 1 (secured) |

**NOTE**

Refer to the Flash block guide for actual security configuration (in section “Flash Module Security”).

#### 5.35.0.2 Operation of the Secured Microcontroller

By securing the device, unauthorized access to the EEPROM and Flash memory contents can be prevented. However, it must be understood the security of the EEPROM and Flash memory contents also depends on the design of the application program. For example, if the application has the capability of downloading code through a serial port and then executing this code (e.g. an application containing bootloader code), then this capability could potentially be used to read the EEPROM and Flash memory contents even when the microcontroller is in the secure state. In this example, the security of the application could be enhanced by requiring a challenge/response authentication before any code can be downloaded.

Secured operation has the following effects on the microcontroller:

##### 5.35.0.2.1 Normal Single Chip Mode (NS)

- Background debug module (BDM) operation is completely disabled.
- Execution of Flash and EEPROM commands is restricted. Refer to the NVM block guide for details.
- Tracing code execution using the DBG module is disabled.

##### 5.35.0.2.2 Special Single Chip Mode (SS)

- BDM firmware commands are disabled.
- BDM hardware commands are restricted to the register space.
- Execution of Flash and EEPROM commands is restricted. Refer to the NVM block guide for details.
- Tracing code execution using the DBG module is disabled.

Special single chip mode means BDM is active after reset. The availability of BDM firmware commands depends on the security state of the device. The BDM secure firmware first performs a blank check of both the Flash memory and the EEPROM. If the blank check succeeds, security is temporarily turned off and the state of the security bits in the appropriate Flash memory location can be changed If the blank check fails, security remains active, only the BDM hardware commands is enabled, and the accessible memory space is restricted to the peripheral register area. This allows the BDM to be used to erase the EEPROM and Flash memory without giving access to their contents. After erasing both Flash memory and EEPROM, another reset into special single chip mode causes the blank check to succeed and the options/security byte can be programmed to “unsecured” state via BDM.

While the BDM is executing the blank check, the BDM interface is completely blocked, which means all BDM commands are temporarily blocked.

##### 5.35.0.2.3 Executing from Internal Memory in Expanded Mode

The user may choose to operate from internal memory while in expanded mode. To do this the user must start in single chip mode and write to the mode bits selecting expanded operation. In this mode internal visibility and IPIPE are blocked. If the users program tries to execute from outside the program memory space (internal space occupied by the FLASH), the FLASH and EEPROM is disabled. BDM operations is blocked.

If the user begins operation in single chip mode with security on, the user is constrained to operate out of internal memory - even if the user changes to expanded mode. To accomplish this, the MMC needs to register the part started in single chip mode and was secured. The CPU provides the state of the two high-order bits of the Program Counter. All this information, plus the firmware size information is used to determine the part is executing in the proper space. If the program strays, the selects for FLASH and EEPROM are disabled by the MMC until the part goes through reset.

##### 5.35.0.2.4 Unsecuring the Microcontroller

Unsecuring the microcontroller can be done by three different methods:

1. Back door key access
2. Reprogramming the security bits
3. Complete memory erase (special modes)

##### 5.35.0.2.5 Unsecuring the MCU Using the Back door Key Access

In normal modes (single chip and expanded), security can be temporarily disabled using the back door key access method. This method requires that:

- The back door key at 0xFF00–0xFF07 (= global addresses 0x7F_FF00–0x7F_FF07) has been programmed to a valid value.
- The KEYEN[1:0] bits within the Flash options/security byte select ‘enabled’.
- In single chip mode, the application program programmed into the microcontroller must be designed to have the capability to write to the back door key locations.

The back door key values themselves would not normally be stored within the application data, which means the application program would have to be designed to receive the back door key values from an external source (e.g. through a serial port).

The back door key access method allows debugging of a secured microcontroller without having to erase the Flash. This is particularly useful for failure analysis.

**NOTE**

No word of the back door key is allowed to have the value 0x0000 or 0xFFFF.

#### 5.35.0.3 Reprogramming the Security Bits

In normal single chip mode (NS), security can also be disabled by erasing and reprogramming the security bits within Flash options/security byte to the unsecured value. Because the erase operation erases the entire sector from 0xFE00–0xFFFF (0x7F_FE00–0x7F_FFFF), the back door key and the interrupt vectors are also erased; this method is not recommended for normal single chip mode. The application software can only erase and program the Flash options/security byte if the Flash sector containing the Flash options/security byte is not protected (see Flash protection). Thus Flash protection is a useful means of preventing this method. The microcontroller enters the unsecured state after the next reset following the programming of the security bits to the unsecured value.

This method requires:

- The application software previously programmed into the microcontroller has been designed to have the capability to erase and program the Flash options/security byte, or security is first disabled using the back door key method, allowing BDM to be used to issue commands to erase and program the Flash options/security byte.
- The Flash sector containing the Flash options/security byte is not protected.

#### 5.35.0.4 Complete Memory Erase (Special Modes)

The microcontroller can be unsecured in special modes by erasing the entire EEPROM and Flash memory contents.

When a secure microcontroller is reset into special single chip mode (SS), the BDM firmware verifies whether the EEPROM and Flash memory are erased. If any EEPROM or Flash memory address is not erased, only BDM hardware commands are enabled. BDM hardware commands can then be used to write to the EEPROM and Flash registers to mass erase the EEPROM and all Flash memory blocks.

When next reset into special single chip mode, the BDM firmware again verifies whether all EEPROM and Flash memory are erased, and this being the case, enables all BDM commands, allowing the Flash options/security byte to be programmed to the unsecured value. The security bits SEC[1:0] in the Flash security register indicates the unsecure state following the next reset.

## 5.36 Initialization of a Virgin Device

“Virgin” cells in the Flash array reads all programmed and the MCU is secured as the SEC[1:0] bits would be loaded with ‘00’ from the Flash security byte.

At wafer probe NVM BIST mode is used to test and initialize the Flash IFR block. Wafer probe leaves the Flash block erased so the MCU is secured.

For blind-assembled products, the following sequence must be used to initialize the Flash array:

- Reset the MCU into special mode.
- Set FCLKDIV to provide a proper FCLK period.
- Set FPROT register to the unprotected state.
- Set the WRALL bit in the FTSTMOD register, if available.
- Load the Flash Pulse Timer with the mass erase time by executing a LDPTMR command write sequence.
- Execute MASERSI commands to mass erase the Flash main block and Flash IFR block.
- Execute the LDPTMR and PGMI command write sequence to program all timing parameters into the Flash IFR block.
- Reset the MCU into special single chip mode. After the reset the BDM secure firmware executes a blank check command. If the blank check succeeds the MCU is temporarily unsecured.
- Execute the PGM command write sequence to program the security byte to the unsecured state.

Blocking access to memories which can be secured during SCAN testing is necessary. While it would take a fair amount of sophistication on the part of a “thief”, our DFT people still consider this a major risk to security. It is therefore highly recommended accesses to the FLASH and EEPROM arrays be blocked at chip level during scan test. Blocking or not blocking security at the core level does not help.

## 5.37 Impact of Security on Test

When silicon comes out of processing, it is extremely unlikely the security bits are configured for unsecure. There needs to be “hooks” for running BIST (if present) or Burn-in by bypassing the security.

If wafer level burn-in is to be used, security must have a bypass which can be connected to by the burn-in layer. In burn-in, security is bypassed, but when the burn-in layer is removed, the state of secreq determines whether the part is secured or not. This may require some sort of weak pull-up device. At some point during testing the internal FLASH and EEPROM needs to be unsecured. This test program should follow the same sequence as a user to unsecure the part: erase the memories, bring the part up in special mode, erase and program the security bits to the unsecured state.

## 5.38 S12 Clock, Reset and Power Management Unit (S12CPMU)

### 5.38.1 Introduction

This specification describes the function of the Clock, Reset and Power Management Unit.

- The Pierce oscillator (OSCLCP) provides a robust, low-noise and low-power external clock source. It is designed for optimal start-up margin with typical crystal oscillators.
- The Voltage regulator (IVREG) operates from the range 3.13 V to 5.5 V. It provides all the required chip internal voltages and voltage monitors.
- The Phase Locked Loop (PLL) provides a highly accurate frequency multiplier with internal filter.
- The Internal Reference Clock (IRC1M) provides a 1.0 MHz clock.

#### 5.38.1.1 Features

The Pierce Oscillator (OSCLCP) contains circuitry to dynamically control current gain in the output amplitude. This ensures a signal with low harmonic distortion, low power and good noise immunity.

- Supports crystals or resonators from 4.0 to 16 MHz.
- High noise immunity due to input hysteresis and spike filtering.
- Low RF emissions with peak-to-peak swing limited dynamically
- Transconductance (gm) sized for optimum start-up margin for typical crystals
- Dynamic gain control eliminates the need for external current limiting resistor
- Integrated resistor eliminates the need for external bias resistor.
- Low power consumption: Operates from internal 1.8 V (nominal) supply, Amplitude control limits power

The Voltage Regulator (IVREG) has the following features:

- Input voltage range from 3.13 V to 5.5 V
- Low-voltage detect (LVD) with low-voltage interrupt (LVI)
- Power-on reset (POR)
- Low-voltage reset (LVR)
- during scan pattern execution option to go to RPM to support IDDq test.
- external voltage reference used for HV-stress test and MIM screen, the external voltage on VDDA, divided by series resistors, are used as input to the regulating loop of the IVREG

The Phase Locked Loop (PLL) has the following features:

- highly accurate and phase locked frequency multiplier
- Configurable internal filter for best stability and lock time.
- Frequency modulation for defined jitter and reduced emission
- Automatic frequency lock detector
- Interrupt request on entry or exit from locked condition
- Reference clock either external (crystal) or internal square wave (1.0 MHz IRC1M) based.
- PLL stability is sufficient for LIN communication, even if using IRC1M as reference clock

The Internal Reference Clock (IRC1M) has the following features:

- Trimmable in frequency
- Factory trimmed value for 1.0 MHz in Flash Memory, can be overwritten by application if required

Other features of the S12CPMU include

- Clock monitor to detect loss of crystal
- Autonomous periodical interrupt (API)

- Bus Clock Generator
  - Clock switch to select either PLLCLK or external crystal/resonator based Bus Clock
  - PLLCLK divider to adjust system speed
- System Reset generation from the following possible sources:
  - Power-on reset (POR)
  - Low-voltage reset (LVR)
  - Illegal address access
  - COP timeout
  - Loss of oscillation (clock monitor fail)
  - External pin $\overline{\text{RESET}}$

### 5.38.1.2 Modes of Operation

This subsection lists and briefly describes all operating modes supported by the S12CPMU.

#### 5.38.1.2.1 Run Mode

The voltage regulator is in Full Performance Mode (FPM).

The Phase-locked Loop (PLL) is on.

The Internal Reference Clock (IRC1M) is on.

The API is available.

- PLL Engaged Internal (PEI)
  - This is the default mode after System Reset and Power-on Reset.
  - The Bus Clock is based on the PLLCLK.
  - After reset the PLL is configured for 64 MHz VCOCLK operation
    Post divider is 0x03, so PLLCLK is VCOCLK divided by 4, that is 16 MHz and Bus Clock is 8.0 MHz.
    The PLL can be reconfigured for other bus frequencies.
  - The reference clock for the PLL (REFCLK) is based on internal reference clock IRC1M
- PLL Engaged External (PEE)
  - The Bus Clock is based on the PLLCLK.
  - This mode can be entered from default mode PEI by performing the following steps:
    - Configure the PLL for desired bus frequency.
    - Program the reference divider (REFDIV[3:0] bits) to divide down oscillator frequency if necessary.
    - Enable the external oscillator (OSCE bit)
- PLL Bypassed External (PBE)
  - The Bus Clock is based on the Oscillator Clock (OSCCLK).
  - This mode can be entered from default mode PEI by performing the following steps:
    - Enable the external oscillator (OSCE bit)
    - Wait for oscillator to start up (UPOSC=1)
    - Select the Oscillator Clock (OSCCLK) as Bus Clock (PLLSEL=0)
  - The PLLCLK is still on to filter possible spikes of the external oscillator clock

#### 5.38.1.2.2 Stop Mode

This mode is entered by executing the CPU STOP instruction.

The voltage regulator is in Reduced Power mode (RPM)

The API is available

The Phase Locked Loop (PLL) is off

The Internal Reference Clock (IRC1M) is off

Core Clock, Bus Clock and BDM Clock are stopped

Depending on the setting of the PSTP and the OSCE bit, Stop mode can be differentiated between Full Stop mode (PSTP = 0 or OSCE=0) and Pseudo Stop mode (PSTP = 1 and OSCE=1).

- Full Stop mode (pstp = 0 or osce=0)
  The external oscillator (OSCLCP) is disabled
  After wake-up from Full Stop mode the Core Clock and Bus Clock are running on PLLCLK (PLLSEL=1). After wake-up from Full Stop mode the COP and RTI are running on IRCCLK (COPOSCSEL=0, RTIOSCSEL=0)
- Pseudo Stop Mode (PSTP = 1 and OSCE=1)
  The external oscillator (OSCLCP) continues to run. If the respective enable bits are set the COP and RTI continues to run.
  The clock configuration bits PLLSEL, COPOSCSEL, RTIOSCSEL are unchanged

**NOTE**

When starting up the external oscillator (either by programming OSCE bit to 1 or on exit from Full Stop Mode with OSCE bit already 1) the software must wait for a minimum time equivalent to the startup-time of the external oscillator $t_{UPOSC}$ before entering Pseudo Stop mode.

### 5.38.1.3 S12IPIMV1 Block Diagram

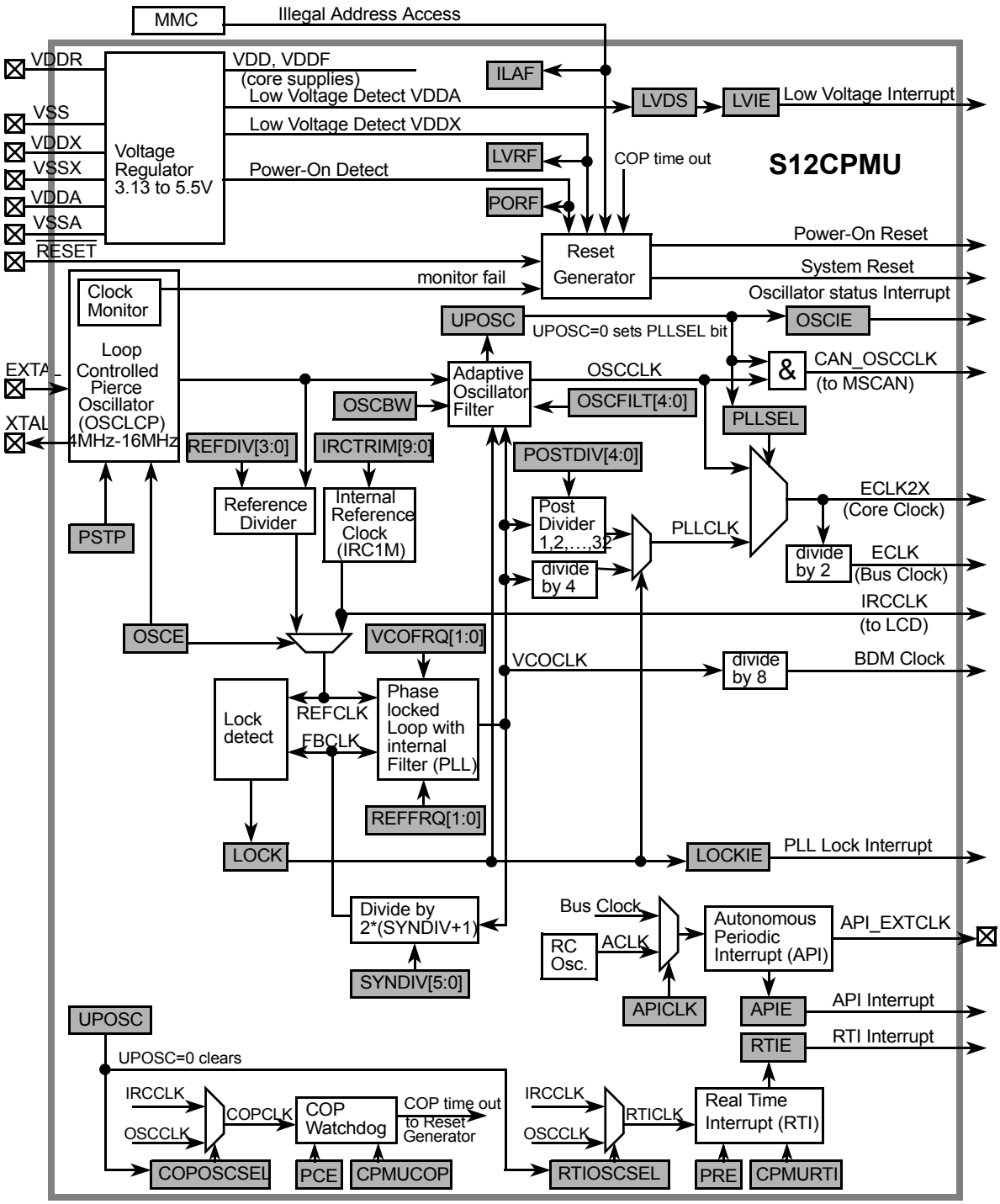



**Figure 84. Block diagram of S12CPMU**

Figure 85 shows a block diagram of the OSCLCP.

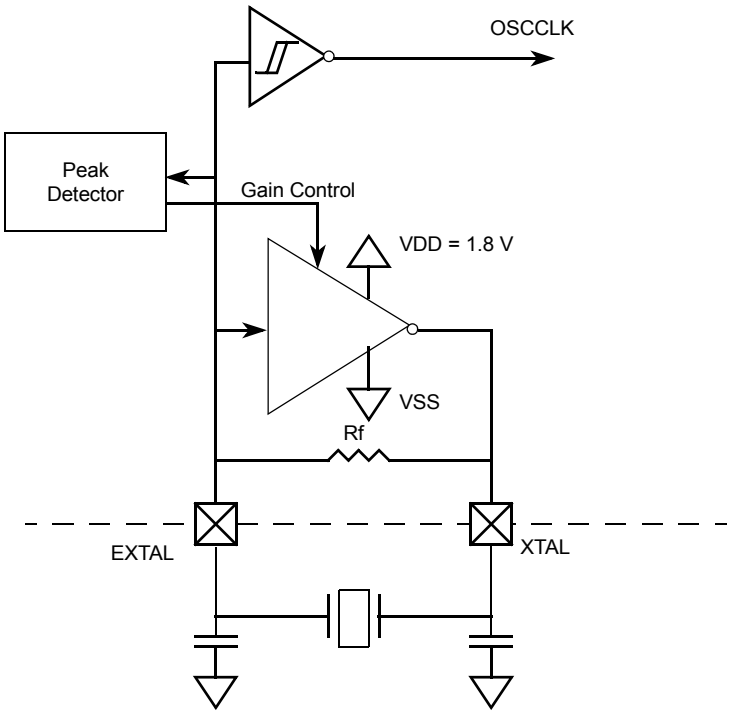



**Figure 85. OSCLCP Block Diagram**

### 5.38.2 Signal Description

This section lists and describes the signals which connect off chip.

#### 5.38.2.1 $\overline{\text{RESET}}$

$\overline{\text{RESET}}$ is an active-low bidirectional pin. As an input it initializes the MCU asynchronously to a known start-up state. As an open-drain output it indicates an MCU-internal reset has been triggered.

#### 5.38.2.2 EXTAL and XTAL

These pins provide the interface for a crystal to control the internal clock generator circuitry. EXTAL is the external clock input or the input to the crystal oscillator amplifier. XTAL is the output of the crystal oscillator amplifier. The MCU internal OSCCLK is derived from the EXTAL input frequency. If OSCE=0, the EXTAL pin is pulled down by an internal resistor of approximately 200 kΩ and the XTAL pin is pulled down by an internal resistor of approximately 700 kΩ.

**NOTE**

Freescale recommends an evaluation of the application board and chosen resonator or crystal by the resonator or crystal supplier. Loop controlled circuit is not suited for overtone resonators and crystals.

#### 5.38.2.3 VDDR — Regulator Power Input Pin

Pin $V_{DDR}$ is the power input of IVREG. All currents sourced into the regulator loads flow through this pin.

An off-chip decoupling capacitor (100 nF...220 nF, X7R ceramic) between $V_{DDR}$ and $V_{SS}$ can smooth ripple on $V_{DDR}$.

#### 5.38.2.4 VSS — Ground Pin

#### 5.38.2.5 VSS must be grounded.VDDA, VSSA — Regulator Reference Supply Pins

Pins $V_{DDA}$ and $V_{SSA}$ are used to supply the analog parts of the regulator.
Internal precision reference circuits are supplied from these signals.

An off-chip decoupling capacitor (100 nF...220 nF, X7R ceramic) between $V_{DDA}$ and $V_{SSA}$ can improve the quality of this supply.

### 5.38.2.6 VDDX, VSSX— Pad Supply Pins

This supply domain is monitored by the Low Voltage Reset circuit.

An off-chip decoupling capacitor (100 nF...220 nF, X7R ceramic) between VDDX and VSSX can improve the quality of this supply.

**NOTE**

Depending on the device package following device supply pins are maybe combined into one supply pin: VDDR, VDDX and VDDA.
Depending on the device package following device supply pins are maybe combined into one supply pin: VSS, VSSX and VSSA.
Refer to the device Reference Manual for information if device supply pins are combined into one supply pin for certain packages and which supply pins are combined together.
An off-chip decoupling capacitor (100 nF...220 nF, X7R ceramic) between the combined supply pin pair can improve the quality of this supply.

### 5.38.2.7 VDD — Internal Regulator Output Supply (Core Logic)

Node VDD is a device internal supply output of the voltage regulator which provides the power supply for the core logic. This supply domain is monitored by the Low Voltage Reset circuit.

### 5.38.2.8 VDDF — Internal Regulator Output Supply (NVM Logic)

Node VDDF is a device internal supply output of the voltage regulator which provides the power supply for the NVM logic. This supply domain is monitored by the Low Voltage Reset circuit

### 5.38.2.9 API_EXTCLK — API external clock output pin

This pin provides the signal selected via APIES and is enabled with APIEA bit. See device specification to which pin it connects.

### 5.38.2.10 vddf_test, vdd_test, vddpll_test — supply testmode pins

These pins allow to measure internal VDDF, VDD, VDDPLL.

### 5.38.2.11 cpmu_test_clk

This signal is connected to a device pin and allows measuring internal clocks if cpmu_test_clk_en bit is set.

### 5.38.2.12 cpmu_test_xfc

This signal is connected to a device pin and allows measuring the internal PLL filter node if cpmu_test_xfc_en bit is set.

### 5.38.2.13 REGFT[2:0] and REGT[2:0]

With the ipt_trim_ld_en signal of the PTI, the trim values for VDD and VDDF of the VREG are loaded into CPMUTEST3 register which directly trims the VREG.

## 5.38.3 Memory Map and Registers

This section provides a detailed description of all registers accessible in the S12CPMU.

### 5.38.3.1 Module Memory Map

The S12CPMU registers are shown in Figure 345.

**Table 345. CPMU Register Summary**

| Address | Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0034 | CPMU SYNR | R | VCOFRQ[1:0] | | SYNDIV[5:0] | | | | | |
| | | W | | | | | | | | |
| 0x0035 | CPMU REFDIV | R | REFFRQ[1:0] | | 0 | 0 | REFDIV[3:0] | | | |
| | | W | | | | | | | | |
| | | | = Unimplemented or Reserved | | | | | | | |

**Table 345. CPMU Register Summary (continued)**

| Address | Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0036 | CPMU POSTDIV | R | 0 | 0 | 0 | POSTDIV[4:0] | | | | |
| | | W | | | | | | | | |
| 0x0037 | CPMUFLG | R | RTIF | PORF | LVRF | LOCKIF | LOCK | ILAF | OSCIF | UPOSC |
| | | W | | | | | | | | |
| 0x0038 | CPMUINT | R | RTIE | 0 | 0 | LOCKIE | 0 | 0 | OSCIE | 0 |
| | | W | | | | | | | | |
| 0x0039 | CPMUCLKS | R | PLLSEL | PSTP | 0 | 0 | PRE | PCE | RTI OSCSEL | COP OSCSEL |
| | | W | | | | | | | | |
| 0x003A | CPMUPLL | R | 0 | 0 | FM1 | FM0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x003B | CPMURTI | R | RTDEC | RTR6 | RTR5 | RTR4 | RTR3 | RTR2 | RTR1 | RTR0 |
| | | W | | | | | | | | |
| 0x003C | CPMUCOP | R | WCOP | RSBCK | 0 | 0 | 0 | CR2 | CR1 | CR0 |
| | | W | | | WRTMASK | | | | | |
| 0x003D | RESERVEDCPMUTEST0 | R | fmcs_reg_sel0 | cpmu_test_gfe0 | cpmu_test_xfc_en0 | fc_force_en0 | vcofrq20 | 0 | fm_test0 | test_sqw_osc0 |
| | | W | | | | | | | | |
| 0x003E | RESERVEDCPMUTEST1 | R | pfd_force_en0 | cpmu_test_clk_en0 | cpmu_test_clk_sel[1]0 | cpmu_test_clk_sel[0]0 | osc_lcp_monitor_disable0 | osc_lcp_extsqw_enable0 | pfd_force_up0 | pfd_force_down0 |
| | | W | | | | | | | | |
| 0x003E | RESERVEDCPMUFMCS | R | fmcs_cs[7:0] | | | | | | | |
| | | W | | | | | | | | |
| 0x003F | CPMU 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 |
| 0x02F0 | RESERVED | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x02F1 | CPMU LVCTL | R | 0 | 0 | 0 | 0 | 0 | LVDS | LVIE | LVIF |
| | | W | | | | | | | | |
| 0x02F2 | CPMU APICTL | R | APICLK | 0 | 0 | APIES | APIEA | APIFE | APIE | APIF |
| | | W | | | | | | | | |
| 0x02F3 | CPMUAPITR | R | APITR5 | APITR4 | APITR3 | APITR2 | APITR1 | APITR0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x02F4 | CPMUAPIRH | R | APIR15 | APIR14 | APIR13 | APIR12 | APIR11 | APIR10 | APIR9 | APIR8 |
| | | W | | | | | | | | |

= Unimplemented or Reserved

**Table 345. CPMU Register Summary (continued)**

| Address | Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x02F5 | CPMUAPIRL | R | APIR7 | APIR6 | APIR5 | APIR4 | APIR3 | APIR2 | APIR1 | APIR0 |
| | | W | | | | | | | | |
| 0x02F6 | RESERVEDCPMUTEST3 | R | 0 LVRT | 0vdd_external_en | 0 REGFT2 | 0 REGFT1 | 0 REGFT0 | 0 REGT2 | 0 REGT1 | 0 REGT0 |
| | | W | | | | | | | | |
| 0x02F7 | RESERVED | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x02F8 | CPMU IRCTRIMH | R | TCTRIM[4:0] | | | | | 0 | IRCTRIM[9:8] | |
| | | W | | | | | | | | |
| 0x02F9 | CPMU IRCTRIML | R | IRCTRIM[7:0] | | | | | | | |
| | | W | | | | | | | | |
| 0x02FA | CPMUOSC | R | OSCE | OSCBW | OSCPINS_EN | OSCFILT[4:0] | | | | |
| | | W | | | | | | | | |
| 0x02FB | CPMUPROT | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PROT |
| | | W | | | | | | | | |
| 0x02FC | RESERVEDCPMUTEST2 | R | 0 | 0 | 0 LVRS | 0 LVRFS | 0 LVRXS | 0 | 0 | 0RCEXA |
| | | W | | | | | | | | |
| | | | = Unimplemented or Reserved | | | | | | | |

### 5.38.3.2 Register Descriptions

This section describes all the S12CPMU registers and their individual bits.

Address order is as listed in Table 345.

#### 5.38.3.2.1 S12CPMU Synthesizer Register (CPMUSYNR)

The CPMUSYNR register controls the multiplication factor of the PLL and selects the VCO frequency range.

**Table 346. S12CPMU Synthesizer Register (CPMUSYNR)**

0x0034

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | VCOFRQ[1:0] | | SYNDIV[5:0] | | | | | |
| Reset | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register). Else write has no effect.

**NOTE**

Writing to this register clears the LOCK and UPOSC status bits.

If PLL has locked (LOCK=1) $f_{VCO} = 2 \times f_{REF} \times (SYNDIV + 1)$

**NOTE**

$f_{VCO}$ must be within the specified VCO frequency lock range. Bus frequency $f_{BUS}$ must not exceed the specified maximum.

The VCOFRQ[1:0] bits are used to configure the VCO gain for optimal stability and lock time. For correct PLL operation the VCOFRQ[1:0] bits have to be selected according to the actual target VCOCLK frequency as shown in Table 347. Setting the VCOFRQ[1:0] bits incorrectly can result in a non functional PLL (no locking and/or insufficient stability).

**Table 347. VCO Clock Frequency Selection**

| VCOCLK Frequency Ranges | VCOFRQ[1:0] |
|---|---|
| 32 MHz <= $f_{VCO}$<= 48 MHz | 00 |
| 48 MHz < $f_{VCO}$<= 6 4MHz | 01 |
| Reserved | 10 |
| Reserved | 11 |

#### 5.38.3.2.2 S12CPMU Reference Divider Register (CPMUREFDIV)

The CPMUREFDIV register provides a finer granularity for the PLL multiplier steps when using the external oscillator as reference.

**Table 348. S12CPMU Reference Divider Register (CPMUREFDIV)**

0x0035

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | REFFRQ[1:0] | | 0 | 0 | REFDIV[3:0] | | | |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register). Else write has no effect.

**NOTE**

Write to this register clears the LOCK and UPOSC status bits.

If OSCLCP is enabled (OSCE=1) $$f_{REF} = \frac{f_{OSC}}{(REFDIV + 1)}$$

If OSCLCP is disabled (OSCE=0) $$f_{REF} = f_{IRC1M}$$

The REFFRQ[1:0] bits are used to configure the internal PLL filter for optimal stability and lock time. For correct PLL operation the REFFRQ[1:0] bits have to be selected according to the actual REFCLK frequency as shown in Table 349.

If IRC1M is selected as REFCLK (OSCE=0) the PLL filter is fixed configured for the 1.0 MHz <= $f_{REF}$ <= 2.0 MHz range. The bits can still be written but has no effect on the PLL filter configuration.

For OSCE=1, setting the REFFRQ[1:0] bits incorrectly can result in a non functional PLL (no locking and/or insufficient stability).

**Table 349. Reference Clock Frequency Selection if OSC_LCP is enabled**

| REFCLK Frequency Ranges (OSCE=1) | REFFRQ[1:0] |
|---|---|
| 1.0 MHz <= $f_{REF}$ <= 2.0 MHz | 00 |
| 2.0 MHz < $f_{REF}$ <= 6.0 MHz | 01 |
| 6.0 MHz < $f_{REF}$ <= 12.0 MHz | 10 |
| $f_{REF}$ >12 MHz | 11 |

#### 5.38.3.2.3 S12CPMU Post Divider Register (CPMUPOSTDIV)

The POSTDIV register controls the frequency ratio between the VCOCLK and the PLLCLK.

**Table 350. S12CPMU Post Divider Register (CPMUPOSTDIV)**

0x0036

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | POSTDIV[4:0] | | | | |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

= Unimplemented or Reserved

Read: Anytime

Write: Anytime if PLLSEL=1. Else write has no effect.

If PLL is locked (LOCK=1) $f_{PLL} = \frac{f_{VCO}}{(POSTDIV + 1)}$

If PLL is not locked (LOCK=0) $f_{PLL} = \frac{f_{VCO}}{4}$

If PLL is selected (PLLSEL=1) $f_{bus} = \frac{f_{PLL}}{2}$

#### 5.38.3.2.4 S12CPMU Flags Register (CPMUFLG)

This register provides S12IPIMV1 status bits and flags.

**Table 351. S12CPMU Flags Register (CPMUFLG)**

0x0037

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | RTIF | PORF | LVRF | LOCKIF | LOCK | ILAF | OSCIF | UPOSC |
| W | | | | | | | | |
| Reset | 0 | (200) | (201) | 0 | 0 | (202) | 0 | 0 |

= Unimplemented or Reserved

Note:

200. PORF is set to 1 when a power on reset occurs. Unaffected by System Reset.

201. LVRF is set to 1 when a low voltage reset occurs. Unaffected by System Reset. Set by power on reset.

202. ILAF is set to 1 when an illegal address reset occurs. Unaffected by System Reset. Cleared by power on reset.

Read: Anytime

Write: Refer to each bit for individual write conditions

**Table 352. CPMUFLG Field Descriptions**

| Field | Description |
|---|---|
| 7<br>RTIF | **Real Time Interrupt Flag** — RTIF is set to 1 at the end of the RTI period. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (RTIE=1), RTIF causes an interrupt request.<br>0 RTI timeout has not yet occurred.<br>1 RTI timeout has occurred. |
| 6<br>PORF | **Power on Reset Flag** — PORF is set to 1 when a power on reset occurs. This flag can only be cleared by writing a 1. Writing a 0 has no effect.<br>0 Power on reset has not occurred.<br>1 Power on reset has occurred. |
| 5<br>LVRF | **Low Voltage Reset Flag** — LVRF is set to 1 when a low voltage reset occurs. This flag can only be cleared by writing a 1. Writing a 0 has no effect.<br>0 Low voltage reset has not occurred.<br>1 Low voltage reset has occurred. |
| 4<br>LOCKIF | **PLL Lock Interrupt Flag** — LOCKIF is set to 1 when LOCK status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect.If enabled (LOCKIE=1), LOCKIF causes an interrupt request.<br>0 No change in LOCK bit.<br>1 LOCK bit has changed. |
| 3<br>LOCK | **Lock Status Bit** — LOCK reflects the current state of PLL lock condition. Writes have no effect. While PLL is unlocked (LOCK=0) fPLL is fVCO / 4 to protect the system from high core clock frequencies during the PLL stabilization time tlock.<br>0 VCOCLK is not within the desired tolerance of the target frequency. $f_{PLL} = f_{VCO}/4$.<br>1 VCOCLK is within the desired tolerance of the target frequency. $f_{PLL} = f_{VCO}/(POSTDIV+1)$. |
| 2<br>ILAF | **Illegal Address Reset Flag** — ILAF is set to 1 when an illegal address reset occurs. Refer to MMC chapter for details. This flag can only be cleared by writing a 1. Writing a 0 has no effect.<br>0 Illegal address reset has not occurred.<br>1 Illegal address reset has occurred. |
| 1<br>OSCIF | **Oscillator Interrupt Flag** — OSCIF is set to 1 when UPOSC status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect.If enabled (OSCIE=1), OSCIF causes an interrupt request.<br>0 No change in UPOSC bit.<br>1 UPOSC bit has changed. |
| 0<br>UPOSC | **Oscillator Status Bit** — UPOSC reflects the status of the oscillator. Writes have no effect. While UPOSC=0 the OSCCLK going to the MSCAN module is off. Entering Full Stop Mode UPOSC is cleared.<br>0 The oscillator is off or oscillation is not qualified by the PLL.<br>1 The oscillator is qualified by the PLL. |

**NOTE**

The Adaptive Oscillator Filter uses the VCO clock as a reference to continuously qualify the external oscillator clock. Because of this, the PLL is always active and a valid PLL configuration is required for the system to work properly. Furthermore, the Adaptive Oscillator Filter is used to determine the status of the external oscillator (reflected in the UPOSC bit). Since this function also relies on the VCO clock, loosing PLL lock status (LOCK=0, except for entering Pseudo Stop mode) means loosing the oscillator status information as well (UPOSC=0).

#### 5.38.3.2.5 S12CPMU Interrupt Enable Register (CPMUINT)

This register enables S12CPMU interrupt requests.

**Table 353. S12CPMU Interrupt Enable Register (CPMUINT)**

0x0038

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

= Unimplemented or Reserved

Read: Anytime

Write: Anytime

**Table 354. CRGINT Field Descriptions**

| Field | Description |
|---|---|
| 7<br>RTIE | **Real Time Interrupt Enable Bit**<br>0 Interrupt requests from RTI are disabled.<br>1 Interrupt is requested whenever RTIF is set. |
| 4<br>LOCKIE | **PLL Lock Interrupt Enable Bit**<br>0 PLL LOCK interrupt requests are disabled.<br>1 Interrupt is requested whenever LOCKIF is set. |
| 1<br>OSCIE | **Oscillator Corrupt Interrupt Enable Bit**<br>0 Oscillator Corrupt interrupt requests are disabled.<br>1 Interrupt is requested whenever OSCIF is set. |

#### 5.38.3.2.6 S12CPMU Clock Select Register (CPMUCLKS)

This register controls S12CPMU clock selection.

**Table 355. S12CPMU Clock Select Register (CPMUCLKS)**

0x0039

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | PLLSEL | PSTP | 0 | 0 | PRE | PCE | RTI OSCSEL | COP OSCSEL |
| W | | | | | | | | |
| Reset | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Read: Anytime

Write:

1. Only possible if PROT=0 (CPMUPROT register) in all MCU Modes (Normal and Special mode).
2. All bits in Special mode (if PROT=0).
3. PLLSEL, PSTP, PRE, PCE, RTIOSCSEL: In Normal mode (if PROT=0).
4. COPOSCSEL: In Normal mode (if PROT=0) until CPMUCOP write once is taken. If COPOSCSEL was cleared by UPOSC=0 (entering full stop mode with COPOSCSEL=1 or insufficient OSCCLK quality), then COPOSCSEL can be set again once.

**NOTE**

After writing CPMUCLKS register, it is strongly recommended to read back the CPMUCLKS register to make sure the write of PLLSEL, RTIOSCSEL and COPOSCSEL was successful.

**Table 356. CPMUCLKS Descriptions**

| Field | Description |
|---|---|
| 7<br>PLLSEL | **PLL Select Bit**<br>This bit selects the PLLCLK as source of the System Clocks (Core Clock and Bus Clock).<br>PLLSEL can only be set to 0, if UPOSC=1.<br>UPOSC= 0 sets the PLLSEL bit.<br>Entering Full Stop Mode sets the PLLSEL bit.<br>0 System clocks are derived from OSCCLK if oscillator is up (UPOSC=1, $f_{BUS} = f_{OSC} / 2$.<br>1 System clocks are derived from PLLCLK, $f_{BUS} = f_{PLL} / 2$. |
| 6<br>PSTP | **Pseudo Stop Bit**<br>This bit controls the functionality of the oscillator during Stop Mode.<br>0 Oscillator is disabled in Stop Mode (Full Stop Mode).<br>1 Oscillator continues to run in Stop Mode (Pseudo Stop Mode), option to run RTI and COP.<br>Note: Pseudo Stop Mode allows for faster STOP recovery and reduces the mechanical stress and aging of the resonator in case of frequent STOP conditions at the expense of a slightly increased power consumption.<br>Note: When starting up the external oscillator (either by programming OSCE bit to 1 or on exit from Full Stop Mode with OSCE bit is already 1) the software must wait for a minimum time equivalent to the startup-time of the external oscillator $t_{UPOSC}$ before entering Pseudo Stop Mode. |
| 3<br>PRE | **RTI Enable During Pseudo Stop Bit** — PRE enables the RTI during Pseudo Stop Mode.<br>0 RTI stops running during Pseudo Stop Mode.<br>1 RTI continues running during Pseudo Stop Mode if RTIOSCSEL=1.<br>Note: If PRE=0 or RTIOSCSEL=0 then the RTI goes static while Stop Mode is active. The RTI counter is not reset. |
| 2<br>PCE | **COP Enable During Pseudo Stop Bit** — PCE enables the COP during Pseudo Stop Mode.<br>0 COP stops running during Pseudo Stop Mode<br>1 COP continues running during Pseudo Stop Mode if COPOSCSEL=1<br>Note: If PCE=0 or COPOSCSEL=0 then the COP goes static while Stop Mode is active. The COP counter is not reset. |
| 1<br>RTIOSCSEL | **RTI Clock Select** — RTIOSCSEL selects the clock source to the RTI. Either IRCCLK or OSCCLK. Changing the RTIOSCSEL bit re-starts the RTI timeout period.<br>RTIOSCSEL can only be set to 1, if UPOSC=1.<br>UPOSC= 0 clears the RTIOSCSEL bit.<br>0 RTI clock source is IRCCLK.<br>1 RTI clock source is OSCCLK. |
| 0<br>COPOSCSEL | **COP Clock Select** — COPOSCSEL selects the clock source to the COP. Either IRCCLK or OSCCLK. Changing the COPOSCSEL bit re-starts the COP timeout period.<br>COPOSCSEL can only be set to 1, if UPOSC=1.<br>UPOSC= 0 clears the COPOSCSEL bit.<br>0 COP clock source is IRCCLK.<br>1 COP clock source is OSCCLK |

#### 5.38.3.2.7 S12CPMU PLL Control Register (CPMUPLL)

This register controls the PLL functionality.

**Table 357. S12CPMU PLL Control Register (CPMUPLL)**

0x003A

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | FM1 | FM0 | 0 | 0 | 0 | 0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register). Else write has no effect.

**NOTE**

Write to this register clears the LOCK and UPOSC status bits.

Care should be taken to ensure the bus frequency does not exceed the specified maximum when frequency modulation is enabled.

The frequency modulation (FM1 and FM0) can not be used if the Adaptive Oscillator Filter is enabled.

**Table 358. CPMUPLL Field Descriptions**

| Field | Description |
|---|---|
| 5, 4<br>FM1, FM0 | PLL **Frequency Modulation Enable Bits** — FM1 and FM0 enable frequency modulation on the VCOCLK. This is to reduce noise emission. The modulation frequency is fref divided by 16. See Table 359 for coding. |

**Table 359. FM Amplitude selection**

| FM1 | FM0 | FM Amplitude / $f_{VCO}$ Variation |
|---|---|---|
| 0 | 0 | FM off |
| 0 | 1 | ±1% |
| 1 | 0 | ±2% |
| 1 | 1 | ±4% |

#### 5.38.3.2.8 S12CPMU RTI Control Register (CPMURTI)

This register selects the timeout period for the Real Time Interrupt.

The clock source for the RTI is either IRCCLK or OSCCLK depending on the setting of the RTIOSCSEL bit. In Stop Mode with PSTP=1 (Pseudo Stop Mode) and RTIOSCSEL=1 the RTI continues to run, else the RTI counter halts in Stop Mode.

**Table 360. S12CPMU RTI Control Register (CPMURTI)**

0x003B

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | RTDEC | RTR6 | RTR5 | RTR4 | RTR3 | RTR2 | RTR1 | RTR0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Read: Anytime

Write: Anytime

**NOTE**

A write to this register starts the RTI timeout period. A change of the RTIOSCSEL bit (writing a different value or loosing UPOSC status) restarts the RTI timeout period.

**Table 361. CPMURTI Field Descriptions**

| Field | Description |
|---|---|
| 7<br>RTDEC | **Decimal or Binary Divider Select Bit** — RTDEC selects decimal or binary based prescaler values.<br>0 Binary based divider value. See Table 362<br>1 Decimal based divider value. See Table 363 |
| 6–4<br>RTR[6:4] | **Real Time Interrupt Prescale Rate Select Bits** — These bits select the prescale rate for the RTI. See Table 362 and Table 363. |
| 3–0<br>RTR[3:0] | **Real Time Interrupt Modulus Counter Select Bits** — These bits select the modulus counter target value to provide additional granularity.Table 362 and Table 363 show all possible divide values selectable by the CPMURTI register. |

**Table 362. RTI Frequency Divide Rates for RTDEC = 0**

| RTR[3:0] | RTR[6:4] = | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **000 (OFF)** | **001 ($2^{10}$)** | **010 ($2^{11}$)** | **011 ($2^{12}$)** | **100 ($2^{13}$)** | **101 ($2^{14}$)** | **110 ($2^{15}$)** | **111 ($2^{16}$)** |
| **0000 (÷1)** | OFF[(203)] | $2^{10}$ | $2^{11}$ | $2^{12}$ | $2^{13}$ | $2^{14}$ | $2^{15}$ | $2^{16}$ |
| **0001 (÷2)** | OFF | $2x2^{10}$ | $2x2^{11}$ | $2x2^{12}$ | $2x2^{13}$ | $2x2^{14}$ | $2x2^{15}$ | $2x2^{16}$ |
| **0010 (÷3)** | OFF | $3x2^{10}$ | $3x2^{11}$ | $3x2^{12}$ | $3x2^{13}$ | $3x2^{14}$ | $3x2^{15}$ | $3x2^{16}$ |
| **0011 (÷4)** | OFF | $4x2^{10}$ | $4x2^{11}$ | $4x2^{12}$ | $4x2^{13}$ | $4x2^{14}$ | $4x2^{15}$ | $4x2^{16}$ |
| **0100 (÷5)** | OFF | $5x2^{10}$ | $5x2^{11}$ | $5x2^{12}$ | $5x2^{13}$ | $5x2^{14}$ | $5x2^{15}$ | $5x2^{16}$ |
| **0101 (÷6)** | OFF | $6x2^{10}$ | $6x2^{11}$ | $6x2^{12}$ | $6x2^{13}$ | $6x2^{14}$ | $6x2^{15}$ | $6x2^{16}$ |
| **0110 (÷7)** | OFF | $7x2^{10}$ | $7x2^{11}$ | $7x2^{12}$ | $7x2^{13}$ | $7x2^{14}$ | $7x2^{15}$ | $7x2^{16}$ |
| **0111 (÷8)** | OFF | $8x2^{10}$ | $8x2^{11}$ | $8x2^{12}$ | $8x2^{13}$ | $8x2^{14}$ | $8x2^{15}$ | $8x2^{16}$ |
| **1000 (÷9)** | OFF | $9x2^{10}$ | $9x2^{11}$ | $9x2^{12}$ | $9x2^{13}$ | $9x2^{14}$ | $9x2^{15}$ | $9x2^{16}$ |
| **1001 (÷10)** | OFF | $10x2^{10}$ | $10x2^{11}$ | $10x2^{12}$ | $10x2^{13}$ | $10x2^{14}$ | $10x2^{15}$ | $10x2^{16}$ |
| **1010 (÷11)** | OFF | $11x2^{10}$ | $11x2^{11}$ | $11x2^{12}$ | $11x2^{13}$ | $11x2^{14}$ | $11x2^{15}$ | $11x2^{16}$ |
| **1011 (÷12)** | OFF | $12x2^{10}$ | $12x2^{11}$ | $12x2^{12}$ | $12x2^{13}$ | $12x2^{14}$ | $12x2^{15}$ | $12x2^{16}$ |
| **1100 (÷13)** | OFF | $13x2^{10}$ | $13x2^{11}$ | $13x2^{12}$ | $13x2^{13}$ | $13x2^{14}$ | $13x2^{15}$ | $13x2^{16}$ |
| **1101 (÷14)** | OFF | $14x2^{10}$ | $14x2^{11}$ | $14x2^{12}$ | $14x2^{13}$ | $14x2^{14}$ | $14x2^{15}$ | $14x2^{16}$ |
| **1110 (÷15)** | OFF | $15x2^{10}$ | $15x2^{11}$ | $15x2^{12}$ | $15x2^{13}$ | $15x2^{14}$ | $15x2^{15}$ | $15x2^{16}$ |
| **1111 (÷16)** | OFF | $16x2^{10}$ | $16x2^{11}$ | $16x2^{12}$ | $16x2^{13}$ | $16x2^{14}$ | $16x2^{15}$ | $16x2^{16}$ |

Note:
203. Denotes the default value out of reset.This value should be used to disable the RTI to ensure future backwards compatibility.

**Table 363. RTI Frequency Divide Rates for RTDEC=1**

| RTR[3:0] | RTR[6:4] = | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **000 ($1x10^3$)** | **001 ($2x10^3$)** | **010 ($5x10^3$)** | **011 ($10x10^3$)** | **100 ($20x10^3$)** | **101 ($50x10^3$)** | **110 ($100x10^3$)** | **111 ($200x10^3$)** |
| **0000 (÷1)** | $1x10^3$ | $2x10^3$ | $5x10^3$ | $10x10^3$ | $20x10^3$ | $50x10^3$ | $100x10^3$ | $200x10^3$ |
| **0001 (÷2)** | $2x10^3$ | $4x10^3$ | $10x10^3$ | $20x10^3$ | $40x10^3$ | $100x10^3$ | $200x10^3$ | $400x10^3$ |
| **0010 (÷3)** | $3x10^3$ | $6x10^3$ | $15x10^3$ | $30x10^3$ | $60x10^3$ | $150x10^3$ | $300x10^3$ | $600x10^3$ |
| **0011 (÷4)** | $4x10^3$ | $8x10^3$ | $20x10^3$ | $40x10^3$ | $80x10^3$ | $200x10^3$ | $400x10^3$ | $800x10^3$ |
| **0100 (÷5)** | $5x10^3$ | $10x10^3$ | $25x10^3$ | $50x10^3$ | $100x10^3$ | $250x10^3$ | $500x10^3$ | $1x10^6$ |
| **0101 (÷6)** | $6x10^3$ | $12x10^3$ | $30x10^3$ | $60x10^3$ | $120x10^3$ | $300x10^3$ | $600x10^3$ | $1.2x10^6$ |
| **0110 (÷7)** | $7x10^3$ | $14x10^3$ | $35x10^3$ | $70x10^3$ | $140x10^3$ | $350x10^3$ | $700x10^3$ | $1.4x10^6$ |
| **0111 (÷8)** | $8x10^3$ | $16x10^3$ | $40x10^3$ | $80x10^3$ | $160x10^3$ | $400x10^3$ | $800x10^3$ | $1.6x10^6$ |
| **1000 (÷9)** | $9x10^3$ | $18x10^3$ | $45x10^3$ | $90x10^3$ | $180x10^3$ | $450x10^3$ | $900x10^3$ | $1.8x10^6$ |
| **1001 (÷10)** | $10 x10^3$ | $20x10^3$ | $50x10^3$ | $100x10^3$ | $200x10^3$ | $500x10^3$ | $1x10^6$ | $2x10^6$ |
| **1010 (÷11)** | $11 x10^3$ | $22x10^3$ | $55x10^3$ | $110x10^3$ | $220x10^3$ | $550x10^3$ | $1.1x10^6$ | $2.2x10^6$ |

**Table 363. RTI Frequency Divide Rates for RTDEC=1 (continued)**

| RTR[3:0] | RTR[6:4] = | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **000 ($1x10^3$)** | **001 ($2x10^3$)** | **010 ($5x10^3$)** | **011 ($10x10^3$)** | **100 ($20x10^3$)** | **101 ($50x10^3$)** | **110 ($100x10^3$)** | **111 ($200x10^3$)** |
| **1011 (÷12)** | $12x10^3$ | $24x10^3$ | $60x10^3$ | $120x10^3$ | $240x10^3$ | $600x10^3$ | $1.2x10^6$ | $2.4x10^6$ |
| **1100 (÷13)** | $13x10^3$ | $26x10^3$ | $65x10^3$ | $130x10^3$ | $260x10^3$ | $650x10^3$ | $1.3x10^6$ | $2.6x10^6$ |
| **1101 (÷14)** | $14x10^3$ | $28x10^3$ | $70x10^3$ | $140x10^3$ | $280x10^3$ | $700x10^3$ | $1.4x10^6$ | $2.8x10^6$ |
| **1110 (÷15)** | $15x10^3$ | $30x10^3$ | $75x10^3$ | $150x10^3$ | $300x10^3$ | $750x10^3$ | $1.5x10^6$ | $3x10^6$ |
| **1111 (÷16)** | $16x10^3$ | $32x10^3$ | $80x10^3$ | $160x10^3$ | $320x10^3$ | $800x10^3$ | $1.6x10^6$ | $3.2x10^6$ |

### 5.38.3.2.9 S12CPMU COP Control Register (CPMUCOP)

This register controls the COP (Computer Operating Properly) watchdog.

The clock source for the COP is either IRCCLK or OSCCLK depending on the setting of the COPOSCSEL bit. In Stop Mode with PSTP=1(Pseudo Stop Mode), COPOSCSEL=1 and PCE=1 the COP continues to run, else the COP counter halts in Stop Mode.

**Table 364. S12CPMU COP Control Register (CPMUCOP)**

0x003C

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | WCOP | RSBCK | 0 | 0 | 0 | CR2 | CR1 | CR0 |
| W | | | WRTMASK | | | | | |
| Reset | F | 0 | 0 | 0 | 0 | F | F | F |
| Reset | 0 | 0 | | | | 0 | 0 | 0 |

= Unimplemented or Reserved

Note:

204. After de-assert of System Reset the values are automatically loaded from the Flash memory. See Device specification for details.

Read: Anytime

Write:

1. RSBCK: anytime in Special Mode; write to “1” but not to “0” in Normal Mode
2. WCOP, CR2, CR1, CR0:
   - — Anytime in Special Mode, when WRTMASK is 0, otherwise it has no effect
   - — Write once in Normal Mode, when WRTMASK is 0, otherwise it has no effect.
     - – Writing CR[2:0] to “000” has no effect, but counts for the “write once” condition.
     - – Writing WCOP to “0” has no effect, but counts for the “write once” condition.

When a non-zero value is loaded from Flash to CR[2:0] the COP timeout period is started.

A change of the COPOSCSEL bit (writing a different value or loosing UPOSC status) re-starts the COP timeout period.

In Normal Mode the COP timeout period is restarted if either of these conditions is true:

1. Writing a non-zero value to CR[2:0] (anytime in Special Mode, once in Normal Mode) with WRTMASK = 0.
2. Writing WCOP bit (anytime in Special Mode, once in Normal Mode) with WRTMASK = 0.
3. Changing RSBCK bit from “0” to “1”.

In Special Mode, any write access to CPMUCOP register restarts the COP timeout period.

**Table 365. CPMUCOP Field Descriptions**

| Field | Description |
|---|---|
| 7<br>WCOP | **Window COP Mode Bit** — When set, a write to the CPMUARMCOP register must occur in the last 25% of the selected period. A write during the first 75% of the selected period generates a COP reset. As long as all writes occur during this window, $55 can be written as often as desired. Once $AA is written after the $55, the time-out logic restarts and the user must wait until the next window before writing to CPMUARMCOP. Table 366 shows the duration of this window for the seven available COP rates.<br>0 Normal COP operation<br>1 Window COP operation |
| 6<br>RSBCK | **COP and RTI Stop in Active BDM Mode Bit**<br>0 Allows the COP and RTI to keep running in Active BDM mode.<br>1 Stops the COP and RTI counters whenever the part is in Active BDM mode. |
| 5<br>WRTMASK | **Write Mask for WCOP and CR[2:0] Bit** — This write-only bit serves as a mask for the WCOP and CR[2:0] bits while writing the CPMUCOP register. It is intended for BDM writing the RSBCK without changing the content of WCOP and CR[2:0].<br>0 Write of WCOP and CR[2:0] has an effect with this write of CPMUCOP<br>1 Write of WCOP and CR[2:0] has no effect with this write of CPMUCOP. (Does not count for “write once”.) |
| 2–0<br>CR[2:0] | **COP Watchdog Timer Rate Select** — These bits select the COP timeout rate (see Table 366). Writing a nonzero value to CR[2:0] enables the COP counter and starts the timeout period. A COP counter timeout causes a System Reset. This can be avoided by periodically (before timeout) initializing the COP counter via the CPMUARMCOP register.<br>While all of the following four conditions are true the CR[2:0], WCOP bits are ignored and the COP operates at highest timeout period ($2^{24}$ cycles) in normal COP mode (Window COP mode disabled):<br>1) COP is enabled (CR[2:0] is not 000)<br>2) BDM mode active<br>3) RSBCK = 0<br>4) Operation in Special Mode |

**Table 366. COP Watchdog Rates**

| CR2 | CR1 | CR0 | COPCLK - Cycles to Timeout (COPCLK is either IRCCLK or OSCCLK depending on the COPOSCSEL bit) |
|---|---|---|---|
| 0 | 0 | 0 | COP disabled |
| 0 | 0 | 1 | $2^{14}$ |
| 0 | 1 | 0 | $2^{16}$ |
| 0 | 1 | 1 | $2^{18}$ |
| 1 | 0 | 0 | $2^{20}$ |
| 1 | 0 | 1 | $2^{22}$ |
| 1 | 1 | 0 | $2^{23}$ |
| 1 | 1 | 1 | $2^{24}$ |

#### 5.38.3.2.10 Reserved Register CPMUTEST0

**NOTE**

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special Mode can alter the S12CPMU’s functionality.

**Table 367. Reserved Register (CPMUTEST0)**

0x003D

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | fmcs_reg_sel 0 | cpmu_test_ gfe0 | cpmu_test_xf c_en0 | fc_force_en0 | vcofrq20 | 0 | fm_test0 | test_sqw_os c0 |
| W | | | | | | | | |
| Reset | 0 | Note 10 | 0 | 0 | 0 | 0 | 0 | 0 |

1) Power on reset clears the crg_test_gfe bit

= Unimplemented or Reserved

Read: Anytime

Write: Only in Special Mode

**Table 368. CPMUTEST0 Field Descriptions**

| Field | Description |
|---|---|
| 7<br>fmcs_reg_sel | **FMCS Register Select Bit**— This bit switches either CPMUTEST1 or CPMUFMCS test register to address 0x003E.<br>0 CPMUTEST1 register is visible on address 0x003E, fm_cs[7:0] of hard macro driven dynamically (triangular) if fm_enable=1.<br>1 CPMUFMCS register is visible on address 0x003E, fm_cs[7:0] of hard macro driven to value of FMCS register. |
| 6<br>cpmu_test_gfe | **Glitch Filter Enable Test Bit** — This bit goes to the PTI controller, it is intended to enable the RESET pad glitch filter in functional test mode (where it is by default disabled).<br>0 Glitch Filter disable request<br>1 Glitch Filter enable request |
| 5<br>cpmu_test_xfc_en | **XFC Test Pin Enable** — This bit routes the external XFC test pin to the internal filter node. Using this test feature make sure only one source is driving the internal filter node (FC). So for this case write fc_force_en=0, pfd_force_en=1, pfd_force_up=pfd_force_down=0.<br>0 external XFC test pin not connected to internal filter node<br>1 external XFC test pin connected to internal filter node |
| 4<br>fc_force_en | **FC Force Enable Bit** — This bit allows to force the internal filter node FC to defined values. If fc_force_en=1, REFFRQ[1] bit (in CPMUREFDIV register) selects either 1/2 or 1/3 $V_{DDPLL}$ voltage to be driven on FC node. Using this test feature make sure only one source is driving the internal filter node (FC). So for this case write cpmu_test_xfc_en=0, pfd_force_en=1, pfd_force_up=pfd_force_down=0.<br>0 Internal filter node (FC) not driven from defined values (1/2 or 1/3 $V_{DDPLL}$)<br>1 If REFFRQ[1]=1 then internal filter node (FC) is driven to $V_{DDPLL}/3$.<br>If REFFRQ[1]=0 then internal filter node (FC) is driven to $V_{DDPLL}/2$. |
| 3<br>vcofrq2 | **VCO gain Bit 2** — This bit selects together with the VCOFRQ[1:0] bits of the CPMUSYNR register the gain of the VtoI converter in the PLL. Setting vcofrq2-0 all to 1 is intended for 160MHz VCOCLK generation. |
| 1<br>fm_test | **FM test amplitude Bit** — This bit multiplies FM amplitude determined by FM1,FM0 and CPMUFMCS[7:0] by 4. This is to amplify frequency variation on VCOCLK when using PLL test modes (pfd_force_en, fc_force_en). A higher frequency variation is easier to measure on tester.<br>0 FM amplitude multiplied by 1<br>1 FM amplitude multiplied by 4 |
| 0<br>test_sqw_osc | **Test square wave enable Bit** — Enables XTAL pin digital input data used for Oscillator test.<br>0 XTAL pin as digital input disabled<br>1 XTAL pin as digital input enabled |

### 5.38.3.2.11 Reserved Register CPMUTEST1

**NOTE**

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special Mode can alter the S12CPMU's functionality.

**Table 369. Reserved Register (CPMUTEST1)**

**0x003E**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | pfd_force_en 0 | cpmu_test_clk_en0 | cpmu_test_clk_sel[1]0 | cpmu_test_clk_sel[0]0 | osc_lcp_monitor_disable0 | osc_lcp_extsqw_enable0 | pfd_force_up 0 | pfd_force_down0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Read: Anytime

Write: Only in Special Mode

**Table 370. CPMUTEST1 Field Descriptions**

| Field | Description |
|---|---|
| 7<br>pfd_force_en | **Phase Detector Force Enable Bit**— This bit breaks the PLL feedback loop and allows force of phase detector via pfd_force_up or pfd_force_down bit or cpmu_test_xfc pin or fc_force_en.<br>0 Normal functionality of Phase Detector using REFCLK and FBCLK.<br>1 Phase detector de-connected from REFCLK and FBCLK (PLL loop open). |
| 6<br>cpmu_test_clk_en | **CPMU test clock enable Bit**— This bits routes the clock selected by cpmu_test_clk_sel[1:0] to external pin cpmu_test_clk.<br>0 CPMU test clock not observable.<br>1 CPMU test clock observable at external pin. |
| 5, 4<br>cpmu_test_clk_sel[1:0] | **CPMU test clock select Bits**— These bits select the CPMU test clock to be observed on external pin cpmu_test_clk for test or characterization purposes.<br>00 = IRCCLK, 01=OSCCLK, 10=VCOCLK, 11=VCOCLK_DIV4. |
| 3<br>osc_lcp_monitor_disable | **Oscillator clock monitor disable Bit** — to disable the clock monitor in special single chip mode.<br>0 Clock monitor always enabled with OSCE=1.<br>1 Clock monitor disabled regardless of OSCE Bit. |
| 2<br>osc_lcp_extsqw_enable | **Oscillator external square wave enable Bit** — Drives directly osc_lcp_extsqw_enable input of OSCLCP hardmacro. |
| 1<br>pfd_force_up | **Phase Detector Force Up Bit** — If pfd_force_en=1, this bits forces the PLL charge pump to drive the internal filter voltage down, which is VCOCLK frequency goes up. Using this test feature make sure only one source is driving the internal filter node (FC). So for this case write xfc_en=fc_force_en=pfd_force_down=0.<br>0 No effect.<br>1 If pfd_force_en=1 then the charge pump continuously drives internal filter node down. |
| 0<br>pfd_force_up | **Phase Detector Force Down Bit** — If pfd_force_en=1, this bits force the PLL charge pump to drive the internal filter voltage up, which is VCOCLK frequency goes down. Using this test feature make sure only one source is driving the internal filter node (FC). So for this case write xfc_en=fc_force_en=pfd_force_up=0.<br>0 No effect.<br>1 If pfd_force_en=1 then the charge pump continuously drives internal filter node up. |

#### 5.38.3.2.12 Reserved Register CPMUFMCS

**Table 371. Reserved Register (CPMUFMCS)**

0x003E

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | fm_cs[7:0] | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Read: Anytime

Write: Only in special mode

**Table 372. CPMUFMCS Field Descriptions**

| Field | Description |
|---|---|
| 7<br>fm_cs[7:0] | **Frequency Modulation Amplitude Bits**— If fmcs_reg_sel = 1 (CPMUTEST0 register) then fm_cs[7:0] adds current of 0(fm_cs=$00) to 1 (fm_cs=$ff) times the value determined by FME1, FM0 bits to the VCO current. As a result VCOCLK frequency increases. |

#### 5.38.3.2.13 S12CPMU COP Timer Arm/Reset Register (CPMUARMCOP)

This register is used to restart the COP timeout period.

**Table 373. S12CPMU CPMUARMCOP Register**

0x003F

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

Read: Always reads $00

Write: Anytime

When the COP is disabled (CR[2:0] = “000”) writing to this register has no effect.

When the COP is enabled by setting CR[2:0] nonzero, the following applies:

Writing any value other than $55 or $AA causes a COP reset. To restart the COP timeout period write $55 followed by a write of $AA. These writes do not need to occur back to back, but the sequence ($55, $AA) must be completed prior to COP end of timeout period to avoid a COP reset. Sequences of $55 writes are allowed. When the WCOP bit is set, $55 and $AA writes must be done in the last 25% of the selected timeout period; writing any value in the first 75% of the selected period causes a COP reset.

#### 5.38.3.2.14 Low Voltage Control Register (CPMULVCTL)

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

**Table 374. Low Voltage Control Register (CPMULVCTL)**

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 | U | 0 | U |

= Unimplemented or Reserved

Note:
205. The Reset state of LVDS and LVIF depends on the external supplied VDDA level

Read: Anytime

Write: LVIE and LVIF are write anytime, LVDS is read only

**Table 375. CPMULVCTL Field Descriptions**

| Field | Description |
|---|---|
| 2<br>LVDS | **Low-voltage Detect Status Bit** — This read-only status bit reflects the voltage level on VDDRX. Writes have no effect.<br>0 Input voltage VDDRX is above level $V_{LVID}$ or RPM.<br>1 Input voltage VDDRX is below level $V_{LVIA}$ and FPM. |
| 1<br>LVIE | **Low-voltage Interrupt Enable Bit**<br>0 Interrupt request is disabled.<br>1 Interrupt is 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. |

#### 5.38.3.2.15 Autonomous Periodical Interrupt Control Register (CPMUAPICTL)

The CPMUAPICTL register allows the configuration of the autonomous periodical interrupt features.

**Table 376. Autonomous Periodical Interrupt Control Register (CPMUAPICTL)**

0x02F2

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | APICLK | 0 | 0 | APIES | APIEA | APIFE | APIE | APIF |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Read: Anytime

Write: Anytime

**Table 377. CPMUAPICTL Field Descriptions**

| Field | Description |
|---|---|
| 7<br>APICLK | **Autonomous Periodical Interrupt Clock Select Bit** — Selects the clock source for the API. Writable only if APIFE = 0. APICLK cannot be changed if APIFE is set by the same write operation.<br>0 Autonomous periodical interrupt clock used as source.<br>1 Bus Clock used as source. |
| 4<br>APIES | **Autonomous Periodical Interrupt External Select Bit** — Selects the waveform at the external pin API_EXTCLK as shown in Figure . See device level specification for connectivity of API_EXTCLK pin.<br>0 If APIEA and APIFE are set, at the external pin API_EXTCLK periodic high pulses are visible at the end of every selected period with the size of half of the minimum period (APIR=0x0000 in Table 384).<br>1 If APIEA and APIFE are set, at the external pin API_EXTCLK a clock is visible with 2 times the selected API Period. |
| 3<br>APIEA | **Autonomous Periodical Interrupt External Access Enable Bit** — If set, the waveform selected by bit APIES can be accessed externally. See device level specification for connectivity.<br>0 Waveform selected by APIES can not be accessed externally.<br>1 Waveform selected by APIES can be accessed externally, if APIFE is set. |
| 2<br>APIFE | **Autonomous Periodical Interrupt Feature Enable Bit** — Enables the API feature and starts the API timer when set.<br>0 Autonomous periodical interrupt is disabled.<br>1 Autonomous periodical interrupt is enabled and timer starts running. |
| 1<br>APIE | Autonomous Periodical Interrupt Enable Bit<br>0 API interrupt request is disabled.<br>1 API interrupt is requested whenever APIF is set. |
| 0<br>APIF | **Autonomous Periodical Interrupt Flag** — APIF is set to 1 when the in the API configured time has elapsed. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (APIE = 1), APIF causes an interrupt request.<br>0 API time-out has not yet occurred.<br>1 API time-out has occurred. |

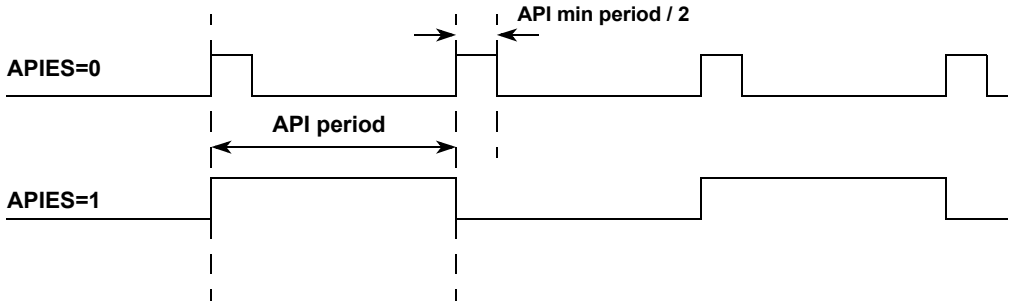



**Figure 86. Waveform selected on API_EXTCLK pin (APIEA=1, APIFE=1)**

### 5.38.3.2.16 Autonomous Periodical Interrupt Trimming Register (CPMUAPITR)

The CPMUAPITR register configures the trimming of the API time-out period.

**Table 378. Autonomous Periodical Interrupt Trimming Register (CPMUAPITR)**

0x02F3

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | APITR5 | APITR4 | APITR3 | APITR2 | APITR1 | APITR0 | 0 | 0 |
| W | | | | | | | | |
| Reset | F | F | F | F | F | F | 0 | 0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

After de-assert of System Reset a value is automatically loaded from the Flash memory.

Read: Anytime

Write: Anytime

**Table 379. CPMUAPITR Field Descriptions**

| Field | Description |
|---|---|
| 7–2<br>APITR[5:0] | **Autonomous Periodical Interrupt Period Trimming Bits** — See Table 380 for trimming effects. The APITR[5:0] value represents a signed number influencing the ACLK period time. |

**Table 380. Trimming Effect of APITR**

| Bit | Trimming Effect |
|---|---|
| APITR[5] | Increases period |
| APITR[4] | Decreases period less than APITR[5] increased it |
| APITR[3] | Decreases period less than APITR[4] |
| APITR[2] | Decreases period less than APITR[3] |
| APITR[1] | Decreases period less than APITR[2] |
| APITR[0] | Decreases period less than APITR[1] |

#### 5.38.3.2.17 Autonomous Periodical Interrupt Rate High and Low Register (CPMUAPIRH / CPMUAPIRL)

The CPMUAPIRH and CPMUAPIRL registers allow the configuration of the autonomous periodical interrupt rate.

**Table 381. Autonomous Periodical Interrupt Rate High Register (CPMUAPIRH)**

0x02F4

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | APIR15 | APIR14 | APIR13 | APIR12 | APIR11 | APIR10 | APIR9 | APIR8 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Table 382. Autonomous Periodical Interrupt Rate Low Register (CPMUAPIRL)**

0x02F5

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | APIR7 | APIR6 | APIR5 | APIR4 | APIR3 | APIR2 | APIR1 | APIR0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Read: Anytime

Write: Anytime if APIFE=0. Else writes have no effect.

**Table 383. CPMUAPIRH / CPMUAPIRL Field Descriptions**

| Field | Description |
|---|---|
| 15-0<br>APIR[15:0] | **Autonomous Periodical Interrupt Rate Bits** — These bits define the time-out period of the API. See Table 384 for details of the effect of the autonomous periodical interrupt rate bits. |

The period can be calculated as follows depending on logical value of the APICLK bit:

APICLK=0: Period = 2*(APIR[15:0] + 1) * $f_{ACLK}$

APICLK=1: Period = 2*(APIR[15:0] + 1) * Bus Clock period

**Table 384. Selectable Autonomous Periodical Interrupt Periods**

| APICLK | APIR[15:0] | Selected Period |
|---|---|---|
| 0 | 0000 | 0.2 ms[1] |
| 0 | 0001 | 0.4 ms[1] |
| 0 | 0002 | 0.6 ms[1] |
| 0 | 0003 | 0.8 ms[1] |
| 0 | 0004 | 1.0 ms[1] |
| 0 | 0005 | 1.2 ms[1] |
| 0 | ..... | ..... |
| 0 | FFFD | 13106.8 ms[1] |
| 0 | FFFE | 13107.0 ms[1] |
| 0 | FFFF | 13107.2 ms[1] |
| 1 | 0000 | 2 * Bus Clock period |
| 1 | 0001 | 4 * Bus Clock period |
| 1 | 0002 | 6 * Bus Clock period |
| 1 | 0003 | 8 * Bus Clock period |
| 1 | 0004 | 10 * Bus Clock period |
| 1 | 0005 | 12 * Bus Clock period |
| 1 | ..... | ..... |
| 1 | FFFD | 131068 * Bus Clock period |
| 1 | FFFE | 131070 * Bus Clock period |
| 1 | FFFF | 131072 * Bus Clock period |

[1] When $f_{ACLK}$ is trimmed to 10KHz.

#### 5.38.3.2.18 Reserved Register CPMUTEST3

**NOTE**

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special mode can alter the S12CPMU's functionality.

**Table 385. Reserved Register (CPMUTEST3)**

0x02F6

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0LVRT | 0vdd_external_en | 0REGFT2 | 0REGFT1 | 0REGFT0 | 0REGT2 | 0REGT1 | 0REGT0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Power on Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Read: Anytime

Write: Only in Special Mode

### 5.38.3.2.19 S12CPMU IRC1M Trim Registers (CPMUIRCTRIMH / CPMUIRCTRIML)

**Table 386. S12CPMU IRC1M Trim High Register (CPMUIRCTRIMH)**

0x02F8

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| R | TCTRIM[4:0] | | | | | 0 | IRCTRIM[9:8] | |
| W | | | | | | | | |
| Reset | F | F | F | F | 0 | 0 | F | F |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

Note:

206. After de-assert of System Reset a factory programmed trim value is automatically loaded from the Flash memory to provide trimmed Internal Reference Frequency $f_{IRC1M_TRIM}$.

**Table 387. S12CPMU IRC1M Trim Low Register (CPMUIRCTRIML)**

0x02F9

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | IRCTRIM[7:0] | | | | | | | |
| W | | | | | | | | |
| Reset | F | F | F | F | F | F | F | F |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Note:

207. After de-assert of System Reset a factory programmed trim value is automatically loaded from the Flash memory to provide trimmed Internal Reference Frequency $f_{IRC1M_TRIM}$.

Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register). Else write has no effect

**NOTE**

Writes to these registers while PLLSEL=1 clears the LOCK and UPOSC status bits.

**Table 388. CPMUIRCTRIMH/L Field Descriptions**

| Field | Description |
|---|---|
| 15-11<br>TCTRIM | **IRC1M temperature coefficient Trim Bits**<br>Trim bits for the Temperature Coefficient (TC) of the IRC1M frequency.<br>Figure 88 shows the influence of the bits TCTRIM4:0] on the relationship between frequency and temperature.<br>Figure 88 shows an approximate TC variation, relative to the nominal TC of the IRC1M (i.e. for TCTRIM[4:0]=0x00000 or 0x10000). |
| 9-0<br>IRCTRIM | **IRC1M Frequency Trim Bits** — Trim bits for Internal Reference Clock<br>After System Reset the factory programmed trim value is automatically loaded into these registers, resulting in an Internal Reference Frequency $f_{IRC1M_TRIM}$. See device electrical characteristics for value of $f_{IRC1M_TRIM}$.<br>The frequency trimming consists of two different trimming methods:<br>A rough trimming controlled by bits IRCTRIM[9:6] can be done with frequency leaps of about 6% in average.<br>A fine trimming controlled by the bits IRCTRIM[5:0] can be done with frequency leaps of about 0.3% (this trimming determines the precision of the frequency setting of 0.15%, i.e. 0.3% is the distance between two trimming values).<br>Figure 87 shows the relationship between the trim bits and the resulting IRC1M frequency. |

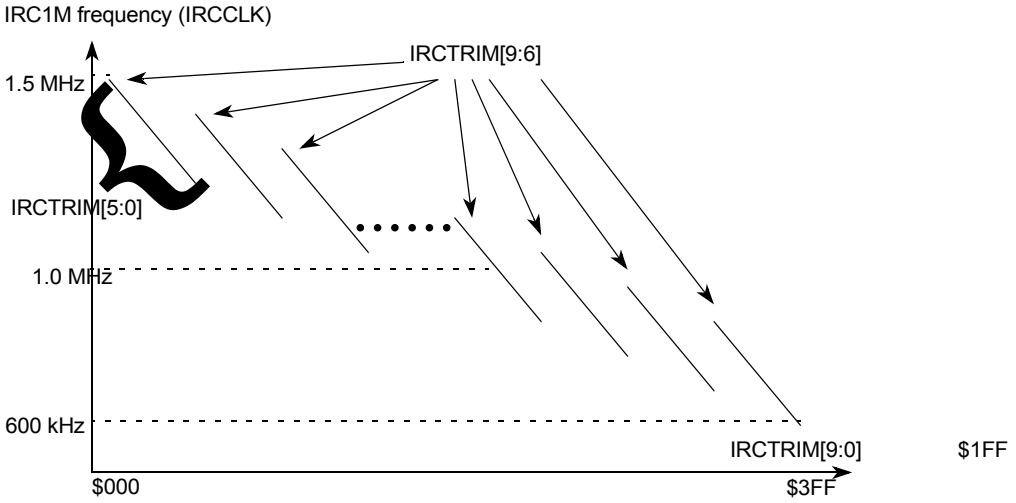



**Figure 87. IRC1M Frequency Trimming Diagram**

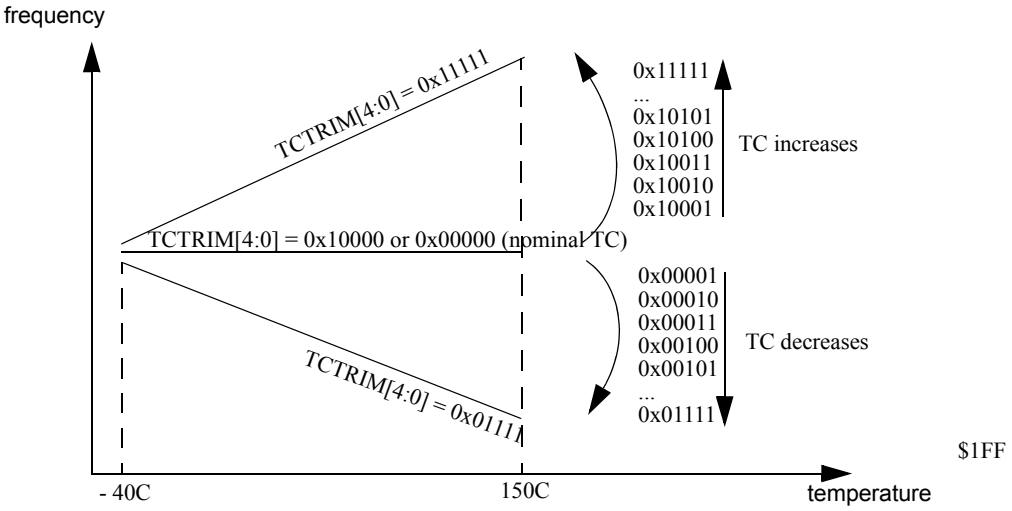



**Figure 88. Influence of TCTRIM[4:0] on the Temperature Coefficient**

**NOTE**

The frequency is not necessarily linear with the temperature (in most cases it does not). The above diagram is meant only to give the direction (positive or negative) of the variation of the TC, relative to the nominal TC.

Setting TCTRIM[4:0] at 0x00000 or 0x10000 does not mean the temperature coefficient is zero. These two combinations basically switch off the TC compensation module, which result in the nominal TC of the IRC1M.

**Table 389. TC Trimming of the Frequency of the IRC1M**

| TCTRIM[4:0] | IRC1M indicative relative TC variation | IRC1M indicative frequency drift for relative TC variation |
|---|---|---|
| 00000 | 0 (nominal TC of the IRC) | 0% |
| 00001 | -0.27% | -0.5% |
| 00010 | -0.54% | -0.9% |
| 00011 | -0.81% | -1.3% |
| 00100 | -1.08% | -1.7% |
| 00101 | -1.35% | -2.0% |
| 00110 | -1.63% | -2.2% |
| 00111 | -1.9% | -2.5% |
| 01000 | -2.20% | -3.0% |
| 01001 | -2.47% | -3.4% |
| 01010 | -2.77% | -3.9% |
| 01011 | -3.04 | -4.3% |
| 01100 | -3.33% | -4.7% |
| 01101 | -3.6% | -5.1% |
| 01110 | -3.91% | -5.6% |
| 01111 | -4.18% | -5.9% |
| 10000 | 0 (nominal TC of the IRC) | 0% |
| 10001 | +0.27% | +0.5% |
| 10010 | +0.54% | +0.9% |
| 10011 | +0.81% | +1.3% |
| 10100 | +1.07% | +1.7% |
| 10101 | +1.34% | +2.0% |
| 10110 | +1.59% | +2.2% |
| 10111 | +1.86% | +2.5% |
| 11000 | +2.11% | +3.0% |
| 11001 | +2.38% | +3.4% |
| 11010 | +2.62% | +3.9% |
| 11011 | +2.89% | +4.3% |
| 11100 | +3.12% | +4.7% |
| 11101 | +3.39% | +5.1% |

**Table 389. TC Trimming of the Frequency of the IRC1M (continued)**

| TCTRIM[4:0] | IRC1M indicative relative TC variation | IRC1M indicative frequency drift for relative TC variation |
|---|---|---|
| 11110 | +3.62% | +5.6% |
| 11111 | +3.89% | +5.9% |

**NOTE**

Since the IRC1M frequency is not a linear function of the temperature, but more like a parabola, the above relative variation is only an indication and should be considered with care.

Be aware the output frequency varies with the TC trimming. A frequency trimming correction is therefore necessary. The values provided in Table 389 are typical values at ambient temperature which can vary from device to device.

#### 5.38.3.2.20 S12CPMU Oscillator Register (CPMUOSC)

This register configures the external oscillator (OSCLCP).

**Table 390. S12CPMU Oscillator Register (CPMUOSC)**

0x02FA

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | OSCE | OSCBW | OSCPINS_EN | OSCFILT[4:0] | | | | |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register). Else write has no effect.

**NOTE**

Write to this register clears the LOCK and UPOSC status bits.

If the chosen VCOCLK-to-OSCCLK ratio divided by two (($f_{VCO}$ / $f_{OSC}$)/2) is not an integer number, the filter can not be used and the OSCFILT[4:0] bits must be set to 0.

The frequency modulation (FM1 and FM0) can not be used if the Adaptive Oscillator Filter is enabled.

**Table 391. CPMUOSC Field Descriptions**

| Field | Description |
|---|---|
| 7<br>OSCE | **Oscillator Enable Bit** — This bit enables the external oscillator (OSCLCP). The UPOSC status bit in the CPMUFLG register indicates when the oscillation is stable and OSCCLK can be selected as Bus Clock or source of the COP or RTI. A loss of oscillation leads to a clock monitor reset.<br>0 External oscillator is disabled. REFCLK for PLL is IRCCLK.<br>1 External oscillator is enabled.Clock monitor is enabled. REFCLK for PLL is external oscillator clock divided by REFDIV.<br>Note: When starting up the external oscillator (either by programming OSCE bit to 1 or on exit from Full Stop Mode with OSCE bit already 1) the software must wait for a minimum time equivalent to the startup-time of the external oscillator $t_{UPOSC}$ before entering Pseudo Stop Mode. |
| 6<br>OSCBW | **Oscillator Filter Bandwidth Bit** — If the VCOCLK frequency exceeds 25 MHz wide bandwidth must be selected. The Oscillator Filter is described in more detail at Section 5.38.4.5.2, “The Adaptive Oscillator Filter"<br>0 Oscillator filter bandwidth is narrow (window for expected OSCCLK edge is one VCOCLK cycle).<br>1 Oscillator filter bandwidth is wide (window for expected OSCCLK edge is three VCOCLK cycles). |

**Table 391. CPMUOSC Field Descriptions (continued)**

| Field | Description |
|---|---|
| 5<br>OSCPINS_EN | **Oscillator Pins EXTAL and XTAL Enable Bit**<br>If OSCE=1 this read-only bit is set. It can only be cleared with the next reset.<br>Enabling the external oscillator reserves the EXTAL and XTAL pins exclusively for oscillator application.<br>0 EXTAL and XTAL pins are not reserved for oscillator.<br>1 EXTAL and XTAL pins exclusively reserved for oscillator. |
| 4-0<br>OSCFILT | **Oscillator Filter Bits** — When using the oscillator a noise filter can be enabled, which filters noise from the incoming external oscillator clock and detects if the external oscillator clock is qualified or not (quality status shown by bit UPOSC).<br>The VCOCLK-to-OSCCLK ratio divided by two (($f_{VCO}$ / $f_{OSC}$)/2) must be an integer value. This value must be written to the OSCFILT[4:0] bits to enable the Adaptive Oscillator Filter.<br>0x0000 Adaptive Oscillator Filter disabled.<br>else Adaptive Oscillator Filter enabled] |

#### 5.38.3.2.21 S12CPMU Protection Register (CPMUPROT)

This register protects the following clock configuration registers from accidental overwrite:
CPMUSYNR, CPMUREFDIV, CPMUCLKS, CPMUPLL, CPMUIRCTRIMH/L and CPMUOSC

**Table 392. S12CPMU Protection Register (CPMUPROT)**

0x02FB

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

Read: Anytime
Write: Anytime

**Table 393. Clock Configuration Registers Protection Bit**

| Field | Description |
|---|---|
| 0 | **Clock Configuration Registers Protection Bit** — This bit protects the clock configuration registers from accidental overwrite (see list of protected registers above).<br>Writing 0x26 to the CPMUPROT register clears the PROT bit, other write accesses set the PROT bit.<br>0 Protection of clock configuration registers is disabled.<br>1 Protection of clock configuration registers is enabled. (see list of protected registers above) |

#### 5.38.3.2.22 Reserved Register CPMUTEST2

**NOTE**

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special Mode can alter the S12CPMU's functionality.

**Table 394. Reserved Register CPMUTEST2**

0x02FC

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 LVRS | 0 LVRFS | 0 LVRXS | 0 | 0 | 0RCEXA |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Read: Anytime

Write: Only in Special Mode

The reserved register allows several setting to aid to perform device parametric tests This register can only be written after writing a $E3 before into this register.

**Table 395. CPMUTEST2 Field Descriptions**

| Field | Description |
|---|---|
| 5<br>LVRS | **Low Voltage Reset Detect VDD Status Bit** — This read-only status bit reflects the Status of the Low Voltage Reset on VDD when core reset disabled for parametric tests. Writes have no effect.<br>0 Input voltage $V_{DD}$ is above level $V_{LVRA}$ or device is in Reduced Performance Mode (RPM).<br>1 Input voltage $V_{DD}$ is below level $V_{LVRA}$ and device is in Full Performance Mode (FPM). |
| 4<br>LVRFS | **Low Voltage Reset Detect VDDF Status Bit** — This read-only status bit reflects the Status of the Low Voltage Reset on VDDF when core reset disabled for parametric tests. Writes have no effect.<br>0 Input voltage $V_{DDF}$ is above level $V_{LVRFA}$ or device is in RPM.<br>1 Input voltage $V_{DDF}$ is below level $V_{LVRFA}$ and device is in FPM. |
| 3<br>LVRXS | **Low Voltage Reset Detect VDDX Status Bit** — This read-only status bit reflects the Status of the Low Voltage Reset on VDDX when core reset disabled for parametric tests. Writes have no effect.<br>0 Input voltage $V_{DDX}$ is above level $V_{LVRXA}$ or device is in RPM.<br>1 Input voltage $V_{DDX}$ is below level $V_{LVRXA}$ and device is in FPM. |
| 0<br>RCEXA | **Autonomous Periodical Interrupt clock (ACLK) External Access Enable Bit** — The Autonomous Periodical Interrupt clock (ACLK) can be mapped also to an output pin. See Section 1 (Device Overview) and Section Port Integration Module for details.<br>0 The Autonomous Periodical Interrupt clock (ACLK) is not mapped to an output pin.<br>1 The Autonomous Periodical Interrupt clock (ACLK) is mapped to an output pin if APIFE is set and APIEA=0. |

## 5.38.4 Functional Description

### 5.38.4.1 Phase-locked Loop with Internal Filter (PLL)

The PLL is used to generate a high speed PLLCLK based on a low frequency REFCLK.

The REFCLK is by default the IRCCLK which is trimmed to $f_{IRC1M_TRIM}$=1.0 MHz.

If using the oscillator (OSCE=1) REFCLK is based on OSCCLK. For increased flexibility, OSCCLK can be divided in a range of 1 to 16 to generate the reference frequency REFCLK using the REFDIV[3:0] bits. Based on the SYNDIV[5:0] bits the PLL generates the VCOCLK by multiplying the reference clock by a 2, 4, 6,... 126, 128. Based on the POSTDIV[4:0] bits the VCOCLK can be divided in a range of 1,2, 3, 4, 5, 6,... to 32 to generate the PLLCLK.

If Oscillator is enabled (OSCE=1) $f_{REF} = \frac{f_{OSC}}{(REFDIV + 1)}$

If Oscillator is disabled (OSCE=0) $f_{REF} = f_{IRC1M}$

$f_{VCO} = 2 \times f_{REF} \times (SYNDIV + 1)$

If PLL is locked (LOCK=1) $f_{PLL} = \frac{f_{VCO}}{(POSTDIV + 1)}$

If PLL is not locked (LOCK=0) $f_{PLL} = \frac{f_{VCO}}{4}$

If PLL is selected (PLLSEL=1) $f_{bus} = \frac{f_{PLL}}{2}$

**NOTE**

Although it is possible to set the dividers to command a very high clock frequency, do not exceed the specified bus frequency limit for the MCU.

Several examples of PLL divider settings are shown in Table 396. The following rules help to achieve optimum stability and shortest lock time:

- Use lowest possible $f_{VCO}$ / $f_{REF}$ ratio (SYNDIV value).
- Use highest possible REFCLK frequency $f_{REF}$.

**Table 396. Examples of PLL Divider Settings**

| $f_{osc}$ | REFDIV[3:0] | $f_{REF}$ | REFFRQ[1:0] | SYNDIV[5:0] | $f_{VCO}$ | VCOFRQ[1:0] | POSTDIV[4:0] | $f_{PLL}$ | $f_{bus}$ |
|---|---|---|---|---|---|---|---|---|---|
| off | $00 | 1.0 MHz | 00 | $1F | 64 MHz | 01 | $03 | 16 MHz | 8.0 MHz |
| off | $00 | 1.0 MHz | 00 | $1F | 64 MHz | 01 | $00 | 64 MHz | 32 MHz |
| off | $00 | 1.0 MHz | 00 | $0F | 32 MHz | 00 | $00 | 32 MHz | 16 MHz |
| 4.0 MHz | $00 | 4.0 MHz | 01 | $03 | 32 MHz | 01 | $00 | 32 MHz | 16 MHz |

The phase detector inside the PLL compares the feedback clock (FBCLK = VCOCLK/(SYNDIV+1) with the reference clock (REFCLK = (IRC1M or OSCCLK)/(REFDIV+1)). Correction pulses are generated based on the phase difference between the two signals. The loop filter alters the DC voltage on the internal filter capacitor, based on the width and direction of the correction pulse, which leads to a higher or lower VCO frequency.

The user must select the range of the REFCLK frequency (REFFRQ[1:0] bits) and the range of the VCOCLK frequency (VCOFRQ[1:0] bits) to ensure the correct PLL loop bandwidth is set.

The lock detector compares the frequencies of the FBCLK and the REFCLK. Therefore the speed of the lock detector is directly proportional to the reference clock frequency. The circuit determines the lock condition based on this comparison.

If PLL LOCK interrupt requests are enabled, the software can wait for an interrupt request and for instance check the LOCK bit. If interrupt requests are disabled, software can poll the LOCK bit continuously (during PLL start-up) or at periodic intervals. In either case, only when the LOCK bit is set, the VCOCLK has stabilized to the programmed frequency.

- The LOCK bit is a read-only indicator of the locked state of the PLL.
- The LOCK bit is set when the VCO frequency is within the tolerance, $\Delta_{Lock}$, and is cleared when the VCO frequency is out of the tolerance, $\Delta_{unl}$.
- Interrupt requests can occur if enabled (LOCKIE = 1) when the lock condition changes, toggling the LOCK bit.

### 5.38.4.2 Startup from Reset

An example of startup of clock system from Reset is given in Figure 89.

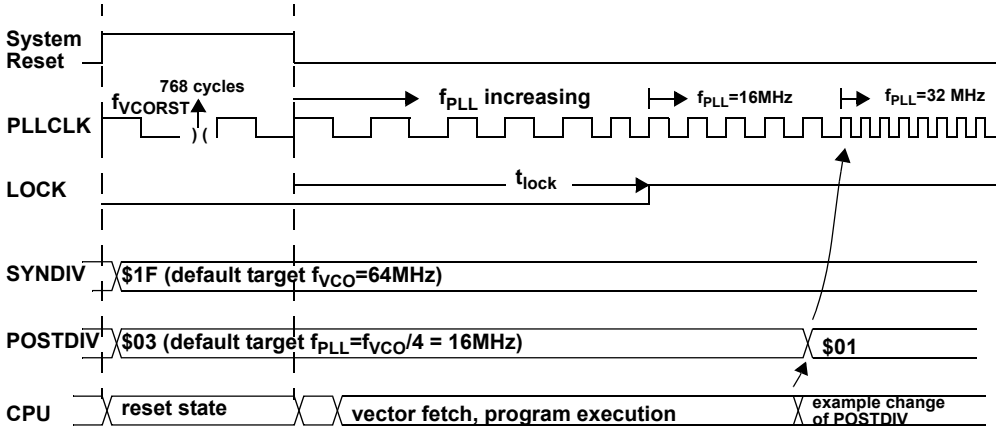



**Figure 89. Startup of Clock System After Reset**

### 5.38.4.3 Stop Mode using PLLCLK as Bus Clock

An example of what happens going into Stop mode and exiting Stop mode after an interrupt is shown in Figure 90. Disable PLL Lock interrupt (LOCKIE=0) before going into Stop mode.

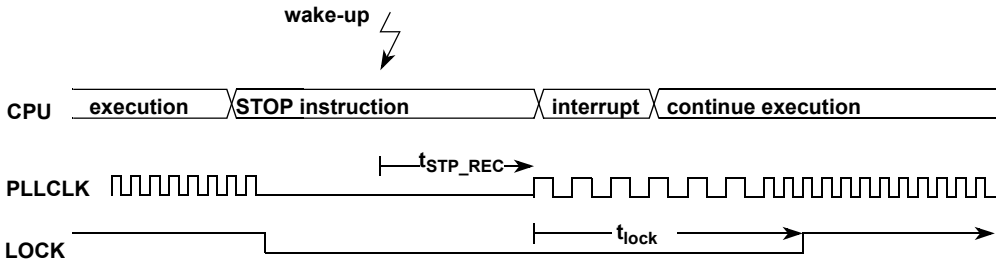



**Figure 90. Stop Mode Using PLLCLK as Bus Clock**

### 5.38.4.4 Full Stop Mode using Oscillator Clock as Bus Clock

An example of what happens going into Full Stop mode and exiting Full Stop mode after an interrupt is shown in Figure 91.

Disable PLL Lock interrupt (LOCKIE=0) and oscillator status change interrupt (OSCIE=0) before going into Full Stop mode.

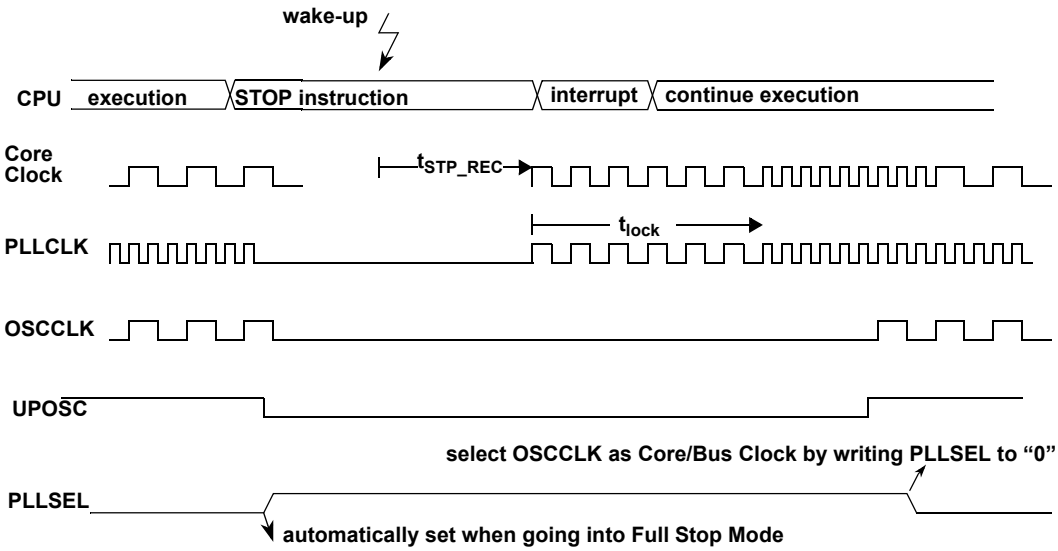



**Figure 91. Full Stop Mode Using Oscillator Clock as Bus Clock**

### 5.38.4.5 External Oscillator

#### 5.38.4.5.1 Enabling the External Oscillator

An example of how to use the oscillator as Bus Clock is shown in Figure 92.

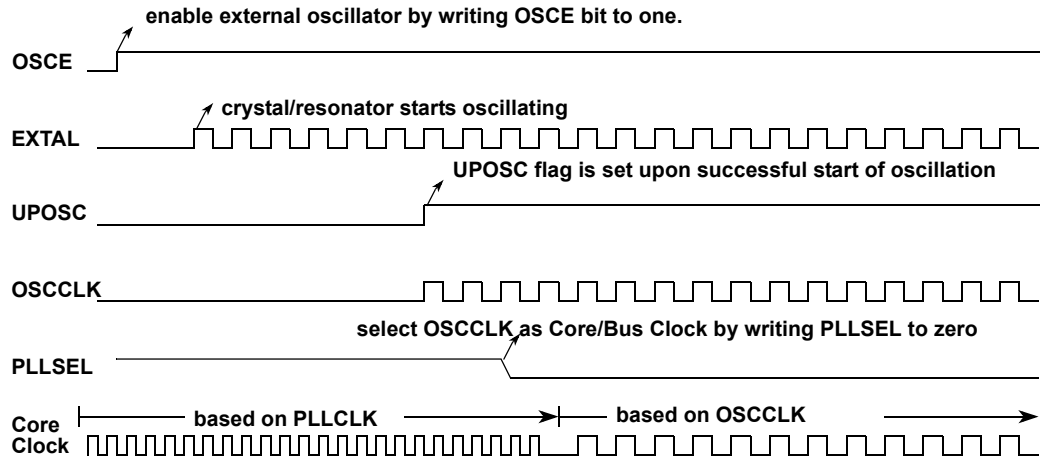



**Figure 92. Enabling the External Oscillator**

#### 5.38.4.5.2 The Adaptive Oscillator Filter

A spike in the oscillator clock can disturb the function of the modules driven by this clock.

The Adaptive Oscillator Filter includes two features:

1. Filter noise (spikes) from the incoming external oscillator clock. The filter feature is illustrated in Figure 93.

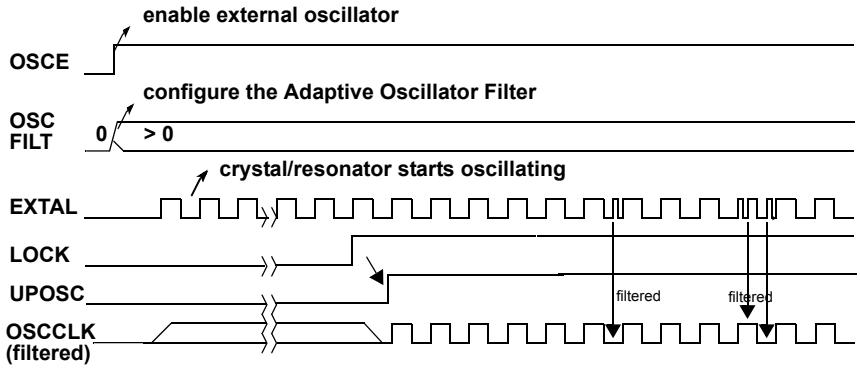



**Figure 93. Noise Filtered by the Adaptive Oscillator Filter**

2. Detect severe noise disturbance on external oscillator clock which can not be filtered and indicate the critical situation to the software by clearing the UPOSC and LOCK status bit and setting the OSCIF and LOCKIF flag. An example for the detection of critical noise is illustrated in Figure 94

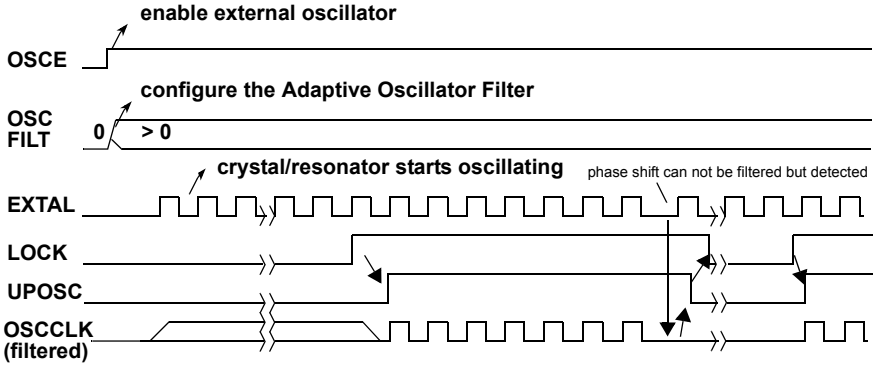



**Figure 94. Critical Noise Detected by the Adaptive Oscillator Filter**

**NOTE**

If the LOCK bit is clear due to severe noise disturbance on the external oscillator clock the PLLCLK is derived from the VCO clock (with its actual frequency) divided by four (see also Section 5.38.3.2.3, “S12CPMU Post Divider Register (CPMUPOSTDIV)").

The use of the filter function is only possible if the VCOCLK-to-OSCCLK ratio divided by two (($f_{VCO}$ / $f_{OSC}$)/2) is an integer number. This integer value must be written to the OSCFILT[4:0] bits.

If enabled, the Adaptive Oscillator Filter is sampling the incoming external oscillator clock signal (EXTAL) with the VCOCLK frequency.

Using VCOCLK, a time window is defined during which an edge of the OSCCLK is expected. In case of OSCBW = 1 the width of this window is three VCOCLK cycles, if the OSCBW = 0 it is one VCOCLK cycle.

The noise detection is active for certain combinations of OSCFILT[4:0] and OSCBW bit settings as shown in Table 397.

**Table 397. Noise Detection Settings**

| OSCFILT[4:0] | OSCBW | Detection | Filter |
|---|---|---|---|
| 0 | x | disabled | disabled |
| 1 | x | disabled | active |
| 2 or 3 | 0 | active | active |
| | 1 | disabled | active |
| >=4 | x | active | active |

**NOTE**

If the VCOCLK frequency is higher than 25 MHz the wide bandwidth must be selected (OSCBW = 1).

### 5.38.4.6 System Clock Configurations

#### 5.38.4.6.1 PLL Engaged Internal Mode (PEI)

This mode is the default mode after System Reset or Power-on Reset.

The Bus Clock is based on the PLLCLK, the reference clock for the PLL is internally generated (IRC1M). The PLL is configured to 64 MHz VCOCLK with POSTDIV set to 0x03. If locked (LOCK=1) this results in a PLLCLK of 16 MHz and a Bus Clock of 8.0 MHz. The PLL can be re-configured to other bus frequencies.

The clock sources for COP and RTI are based on the internal reference clock generator (IRC1M).

#### 5.38.4.6.2 PLL Engaged External Mode (PEE)

In this mode, the Bus Clock is based on the PLLCLK as well (like PEI). The reference clock for the PLL is based on the external oscillator. The adaptive spike filter and detection logic which uses the VCOCLK to filter and qualify the external oscillator clock can be enabled.

The clock sources for COP and RTI can be based on the internal reference clock generator or on the external oscillator clock.

This mode can be entered from default mode PEI by performing the following steps:

1. Configure the PLL for desired bus frequency.
2. Optionally the adaptive spike filter and detection logic can be enabled by calculating the integer value for the OSCFIL[4:0] bits and setting the bandwidth (OSCBW) accordingly.
3. Enable the external Oscillator (OSCE bit).
4. Wait for the PLL being locked (LOCK = 1) and the oscillator to start-up and additionally being qualified if the Adaptive Oscillator Filter is enabled (UPOSC =1).
5. Clear all flags in the CPMUFLG register to be able to detect any future status bit change.
6. Optionally status interrupts can be enabled (CPMUINT register).

Since the Adaptive Oscillator Filter (adaptive spike filter and detection logic) uses the VCOCLK to continuously filter and qualify the external oscillator clock, loosing PLL lock status (LOCK=0) means loosing the oscillator status information as well (UPOSC=0).

The impact of loosing the oscillator status in PEE mode is as follows:

The PLLCLK is derived from the VCO clock (with its actual frequency) divided by four until the PLL locks again.

Application software needs to be prepared to deal with the impact of loosing the oscillator status at any time.

#### 5.38.4.6.3 PLL Bypassed External Mode (PBE)

In this mode, the Bus Clock is based on the external oscillator clock. The reference clock for the PLL is based on the external oscillator. The adaptive spike filter and detection logic can be enabled which uses the VCOCLK to filter and qualify the external oscillator clock.

The clock sources for COP and RTI can be based on the internal reference clock generator or on the external oscillator clock.

This mode can be entered from default mode PEI by performing the following steps:

1. Make sure the PLL configuration is valid.
2. Optionally the adaptive spike filter and detection logic can be enabled by calculating the integer value for the OSCFIL[4:0] bits and setting the bandwidth (OSCBW) accordingly.
3. Enable the external Oscillator (OSCE bit)
4. Wait for the PLL being locked (LOCK = 1) and the oscillator to start-up and additionally being qualified if the Adaptive Oscillator Filter is enabled (UPOSC=1).
5. Clear all flags in the CPMUFLG register to be able to detect any status bit change.
6. Optionally status interrupts can be enabled (CPMUINT register).
7. Select the Oscillator Clock (OSCCLK) as Bus Clock (PLLSEL=0)

Since the Adaptive Oscillator Filter (adaptive spike filter and detection logic) uses the VCOCLK to continuously filter and qualify the external oscillator clock, loosing PLL lock status (LOCK=0) means loosing the oscillator status information as well (UPOSC=0).

The impact of loosing the oscillator status in PBE mode is as follows:

- PLLSEL is set automatically and the Bus Clock is switched back to the PLLCLK.
- The PLLCLK is derived from the VCO clock (with its actual frequency) divided by four until the PLL locks again.

Application software needs to be prepared to deal with the impact of loosing the oscillator status at any time.

In the PBE mode, not every noise disturbance can be indicated by bits LOCK and UPOSC (both bits are based on the Bus Clock domain). There are clock disturbances possible, after which UPOSC and LOCK both stay asserted while occasional pauses on the filtered OSCCLK and resulting Bus Clock occur. The adaptive spike filter is still functional and protects the Bus Clock from frequency overshoot due to spikes on the external oscillator clock. The filtered OSCCLK and resulting Bus Clock pauses until the PLL has stabilized again.

### 5.38.5 Resets

#### 5.38.5.1 General

All reset sources are listed in Table 398. Refer to MCU specification for related vector addresses and priorities

.

**Table 398. Reset Summary**

| Reset Source | Local Enable |
|---|---|
| Power-On Reset (POR) | None |
| Low Voltage Reset (LVR) | None |
| External pin $\overline{\text{RESET}}$ | None |
| Illegal Address Reset | None |
| Clock Monitor Reset | OSCE Bit in CPMUOSC register |
| COP Reset | CR[2:0] in CPMUCOP register |

#### 5.38.5.2 Description of Reset Operation

Upon detection of any reset of Table 398, an internal circuit drives the $\overline{\text{RESET}}$ pin low for 512 PLLCLK cycles. After 512 PLLCLK cycles the $\overline{\text{RESET}}$ pin is released. The reset generator of the S12CPMU waits for additional 256 PLLCLK cycles and then samples the $\overline{\text{RESET}}$ pin to determine the originating source. Table 399 shows which vector is fetched.

**Table 399. Reset Vector Selection**

| Sampled $\overline{\text{RESET}}$ Pin (256 cycles after release) | Oscillator monitor fail pending | COP - timeout pending | Vector Fetch |
|---|---|---|---|
| 1 | 0 | 0 | POR<br>LVR<br>Illegal Address Reset<br>External pin $\overline{\text{RESET}}$ |
| 1 | 1 | X | Clock Monitor Reset |
| 1 | 0 | 1 | COP Reset |
| 0 | X | X | POR<br>LVR<br>Illegal Address Reset<br>External pin $\overline{\text{RESET}}$ |

**NOTE**

While System Reset is asserted the PLLCLK runs with the frequency $f_{VCORST}$.

The internal reset of the MCU remains asserted while the reset generator completes the 768 PLLCLK cycles long reset sequence. In case the $\overline{\text{RESET}}$ pin is externally driven low for more than these 768 PLLCLK cycles (External Reset), the internal reset remains asserted longer.

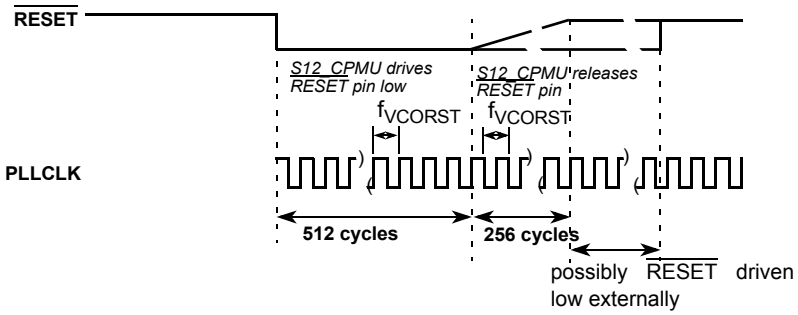



**Figure 95. RESET Timing**

#### 5.38.5.2.1 Clock Monitor Reset

If the external oscillator is enabled (OSCE=1) in case of loss of oscillation or the oscillator frequency is below the failure assert frequency $f_{CMFA}$ (see device electrical characteristics for values), the S12CPMU generates a Clock Monitor Reset. In Full Stop mode the external oscillator and the clock monitor are disabled.

#### 5.38.5.2.2 Computer Operating Properly Watchdog (COP) Reset

The COP (free running watchdog timer) enables the user to check that a program is running and sequencing properly. When the COP is being used, software is responsible for keeping the COP from timing out. If the COP times out it is an indication the software is no longer being executed in the intended sequence; thus COP reset is generated.

The clock source for the COP is either IRCCLK or OSCCLK depending on the setting of the COPOSCSEL bit. In Stop Mode with PSTP=1 (Pseudo Stop Mode), COPOSCSEL=1 and PCE=1 the COP continues to run, else the COP counter halts in Stop Mode.

Three control bits in the CPMUCOP register allow selection of seven COP time-out periods.

When COP is enabled, the program must write $55 and $AA (in this order) to the CPMUARMCOP register during the selected time-out period. Once this is done, the COP time-out period is restarted. If the program fails to do this and the COP times out, a COP reset is generated. Also, if any value other than $55 or $AA is written, a COP reset is generated.

Windowed COP operation is enabled by setting WCOP in the CPMUCOP register. In this mode, writes to the CPMUARMCOP register to clear the COP timer must occur in the last 25% of the selected time-out period. A premature write immediately resets the part.

### 5.38.5.3 Power-On Reset (POR)

The on-chip POR circuitry detects when the internal supply VDD drops below an appropriate voltage level. The POR is deasserted, if the internal supply VDD exceeds an appropriate voltage level (voltage levels are not specified in this document because this internal supply is not visible on device pins).

### 5.38.5.4 Low-voltage Reset (LVR)

The on-chip LVR circuitry detects when one of the supply voltages VDD, VDDF or VDDX drops below an appropriate voltage level. If LVR is deasserted the MCU is fully operational at the specified maximum speed. The LVR assert and deassert levels for the supply voltage VDDX are $V_{LVRXA}$ and $V_{LVRXD}$ and are specified in the device Reference Manual.

## 5.38.6 Interrupts

The interrupt/reset vectors requested by the S12CPMU are listed in Table 400. Refer to MCU specification for related vector addresses and priorities.

**Table 400. S12CPMU Interrupt Vectors**

| Interrupt Source | CCR Mask | Local Enable |
|---|---|---|
| RTI timeout interrupt | I bit | CPMUINT (RTIE) |
| PLL lock interrupt | I bit | CPMUINT (LOCKIE) |
| Oscillator status interrupt | I bit | CPMUINT (OSCIE) |
| Low voltage interrupt | I bit | CPMULVCTL (LVIE) |
| Autonomous Periodical Interrupt | I bit | CPMUAPICTL (APIE) |

### 5.38.6.1 Description of Interrupt Operation

#### 5.38.6.1.1 Real Time Interrupt (RTI)

The clock source for the RTI is either IRCCLK or OSCCLK depending on the setting of the RTIOSCSEL bit. In Stop mode with PSTP=1 (Pseudo Stop mode), RTIOSCSEL=1 and PRE=1 the RTI continues to run, else the RTI counter halts in Stop mode.

The RTI can be used to generate hardware interrupts at a fixed periodic rate. If enabled (by setting RTIE=1), this interrupt occurs at the rate selected by the CPMURTI register. At the end of the RTI timeout period the RTIF flag is set to one and a new RTI timeout period starts immediately.

A write to the CPMURTI register restarts the RTI timeout period.

#### 5.38.6.1.2 PLL Lock Interrupt

The S12CPMU generates a PLL Lock interrupt when the lock condition (LOCK status bit) of the PLL changes, either from a locked state to an unlocked state or vice versa. Lock interrupts are locally disabled by setting the LOCKIE bit to zero. The PLL Lock interrupt flag (LOCKIF) is set to 1 when the lock condition has changed, and is cleared to 0 by writing a 1 to the LOCKIF bit.

#### 5.38.6.1.3 Oscillator Status Interrupt

The Adaptive Oscillator Filter contains two different features:

1. Filters spikes of the external oscillator clock.

2. Qualify the external oscillator clock.

When the OSCE bit is 0, then UPOSC stays 0. When OSCE=1 and OSCFILT = 0, then the filter is transparent and no spikes are filtered. The UPOSC bit is then set after the LOCK bit is set.

Upon detection of a status change (UPOSC), which is an unqualified oscillation becomes qualified or vice versa, the OSCIF flag is set. Going into Full Stop Mode or disabling the oscillator can also cause a status change of UPOSC.

Also, since the Adaptive Oscillator Filter is based on the PLLCLK, any change in PLL configuration or any other event which causes the PLL lock status to be cleared leads to a loss of the oscillator status information as well (UPOSC=0).

Oscillator status change interrupts are locally enabled with the OSCIE bit.

**NOTE**

Loosing the oscillator status (UPOSC=0) affects the clock configuration of the system[(208)]. This needs to be dealt with in application software.

Note:
208. For details refer to “5.38.4.6, “System Clock Configurations“

## 5.39 Serial Peripheral Interface (S12SPIV5)

### 5.39.1 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.

#### 5.39.1.1 Glossary of Terms

| | |
|---|---|
| SPI | Serial Peripheral Interface |
| 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 |

### 5.39.1.2 Features

The S12IPIMV1 includes these distinctive features:

- Master mode and slave mode
- Selectable 8 or 16-bit transfer width
- Bi-directional mode
- Slave select output
- Mode fault error flag with CPU interrupt capability
- Double-buffered data register
- Serial clock with programmable polarity and phase

### 5.39.1.3 Modes of Operation

The SPI functions in two modes: run and stop.

- Run mode
  This is the basic mode of operation.
- Stop mode
  The SPI is inactive in stop mode for reduced power consumption. If the SPI is configured as a master, any transmission in progress stops, but is resumed after CPU goes into run mode. If the SPI is configured as a slave, reception and transmission of data continues, so the slave stays synchronized to the master.

For a detailed description of operating modes, refer to Section 5.39.4.7, “Low Power Mode Options".

### 5.39.1.4 Block Diagram

Figure 96 gives an overview on the SPI architecture. The main parts of the SPI are status, control and data registers, shifter logic, baud rate generator, master/slave control logic, and port control logic.

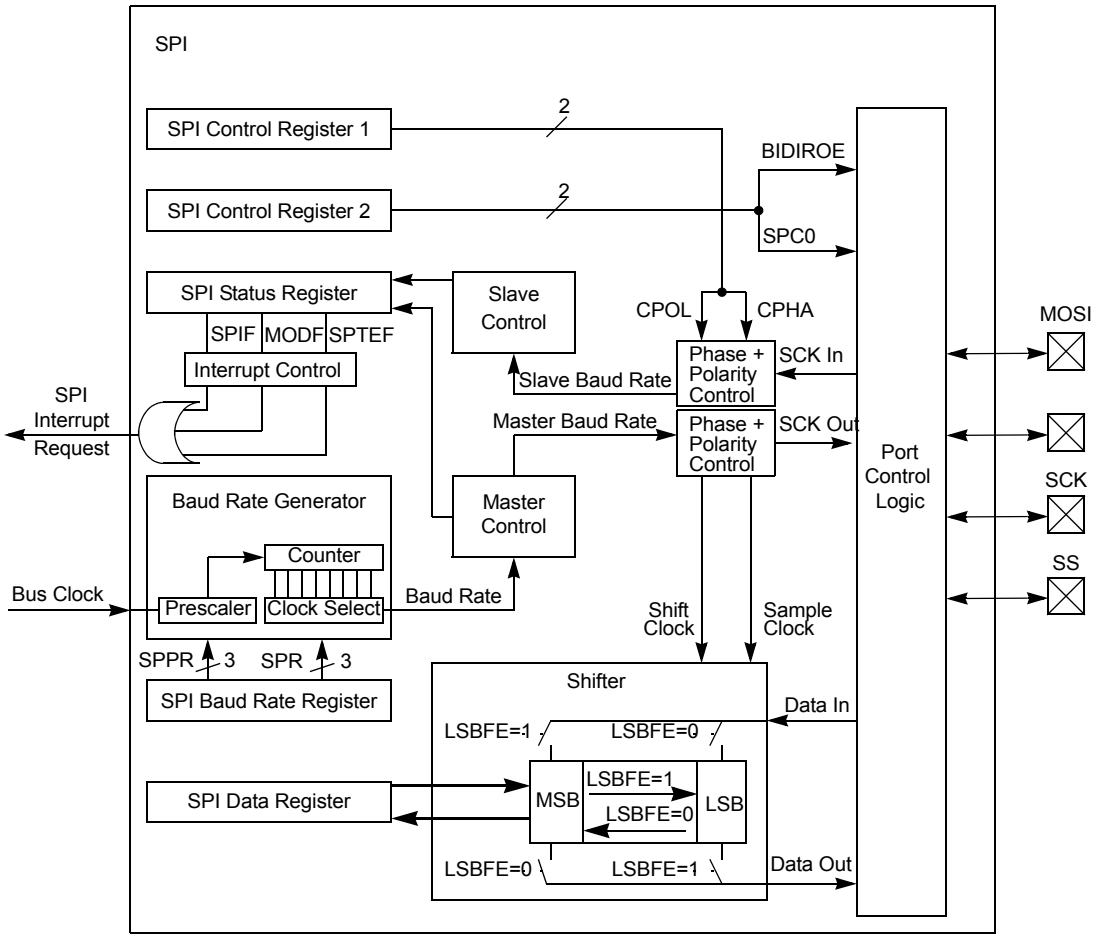



**Figure 96. SPI Block Diagram**

### 5.39.2 External Signal Description

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

#### 5.39.2.1 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.

#### 5.39.2.2 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.

#### 5.39.2.3 SS — Slave Select Pin

This pin is used to output the select signal from the SPI module to another peripheral with which a data transfer is to take place when it is configured as a master and it is used as an input to receive the slave select signal when the SPI is configured as slave.

#### 5.39.2.4 SCK — Serial Clock Pin

In master mode, this is the synchronous output clock. In slave mode, this is the synchronous input clock.

### 5.39.3 Memory Map and Register Definition

This section provides a detailed description of address space and registers used by the SPI.

#### 5.39.3.1 Module Memory Map

The memory map for the SPI is given in Figure 401. The address listed for each register is the sum of a base address and an address offset. The base address is defined at the SoC level and the address offset is defined at the module level. Reads from the reserved bits return zeros and writes to the reserved bits have no effect.

**Table 401. SPI Register Summary**

| Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| SPICR1 | R | SPIE | SPE | SPTIE | MSTR | CPOL | CPHA | SSOE | LSBFE |
| | W | | | | | | | | |
| SPICR2 | R | 0 | XFRW | 0 | MODFEN | BIDIROE | 0 | SPISWAI | SPC0 |
| | W | | | | | | | | |
| SPIBR | R | 0 | SPPR2 | SPPR1 | SPPR0 | 0 | SPR2 | SPR1 | SPR0 |
| | W | | | | | | | | |
| SPISR | R | SPIF | 0 | SPTEF | MODF | 0 | 0 | 0 | 0 |
| | W | | | | | | | | |
| SPIDRH | R | R15 | R14 | R13 | R12 | R11 | R10 | R9 | R8 |
| | W | T15 | T14 | T13 | T12 | T11 | T10 | T9 | T8 |
| SPIDRL | R | R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 |
| | W | T7 | T6 | T5 | T4 | T3 | T2 | T1 | T0 |
| Reserved | R | | | | | | | | |
| | W | | | | | | | | |
| Reserved | R | | | | | | | | |
| | W | | | | | | | | |
| | | = Unimplemented or Reserved | | | | | | | |

### 5.39.3.2 Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

#### 5.39.3.2.1 SPI Control Register 1 (SPICR1)

**Table 402. SPI Control Register 1 (SPICR1)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | SPIE | SPE | SPTIE | MSTR | CPOL | CPHA | SSOE | LSBFE |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

Read: Anytime

Write: Anytime

**Table 403. SPICR1 Field Descriptions**

| Field | Description |
|---|---|
| 7<br>SPIE | **SPI Interrupt Enable Bit** — This bit enables SPI interrupt requests, if SPIF or MODF status flag is set.<br>0 SPI interrupts disabled.<br>1 SPI interrupts enabled. |
| 6<br>SPE | **SPI System Enable Bit** — This bit enables the SPI system and dedicates the SPI port pins to SPI system functions. If SPE is cleared, SPI is disabled and forced into idle state, status bits in SPISR register are reset.<br>0 SPI disabled (lower power consumption).<br>1 SPI enabled, port pins are dedicated to SPI functions. |
| 5<br>SPTIE | **SPI Transmit Interrupt Enable** — This bit enables SPI interrupt requests, if SPTEF flag is set.<br>0 SPTEF interrupt disabled.<br>1 SPTEF interrupt enabled. |
| 4<br>MSTR | **SPI Master/Slave Mode Select Bit** — This bit selects whether the SPI operates in master or slave mode. Switching the SPI from master to slave or vice versa forces the SPI system into idle state.<br>0 SPI is in slave mode.<br>1 SPI is in master mode. |
| 3<br>CPOL | **SPI Clock Polarity Bit** — This bit selects an inverted or non-inverted SPI clock. To transmit data between SPI modules, the SPI modules must have identical CPOL values. In master mode, a change of this bit aborts a transmission in progress and force the SPI system into idle state.<br>0 Active-high clocks selected. In idle state SCK is low.<br>1 Active-low clocks selected. In idle state SCK is high. |
| 2<br>CPHA | **SPI Clock Phase Bit** — This bit is used to select the SPI clock format. In master mode, a change of this bit aborts a transmission in progress and force the SPI system into idle state.<br>0 Sampling of data occurs at odd edges (1,3,5,...) of the SCK clock.<br>1 Sampling of data occurs at even edges (2,4,6,...) of the SCK clock. |
| 1<br>SSOE | **Slave Select Output Enable** — The SS output feature is enabled only in master mode, if MODFEN is set, by asserting the SSOE as shown in Table 404. In master mode, a change of this bit aborts a transmission in progress and force the SPI system into idle state. |
| 0<br>LSBFE | **LSB-First Enable** — This bit does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register always have the MSB in the highest bit position. In master mode, a change of this bit aborts a transmission in progress and force the SPI system into idle state.<br>0 Data is transferred most significant bit first.<br>1 Data is transferred least significant bit first. |

**Table 404. $\overline{SS}$ Input / Output Selection**

| MODFEN | SSOE | Master Mode | Slave Mode |
|---|---|---|---|
| 0 | 0 | $\overline{SS}$ not used by SPI | $\overline{SS}$ input |
| 0 | 1 | $\overline{SS}$ not used by SPI | $\overline{SS}$ input |
| 1 | 0 | $\overline{SS}$ input with MODF feature | $\overline{SS}$ input |
| 1 | 1 | $\overline{SS}$ is slave select output | $\overline{SS}$ input |

#### 5.39.3.2.2 SPI Control Register 2 (SPICR2)

**Table 405. SPI Control Register 2 (SPICR2)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | XFRW | 0 | MODFEN | BIDIROE | 0 | SPISWAI | SPC0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Read: Anytime

Write: Anytime; writes to the reserved bits have no effect

**Table 406. 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. Refer to Section 5.39.3.2.4, "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 aborts a transmission in progress and force the SPI system into idle state.<br>0 8-bit Transfer Width (n = 8)(209)<br>1 16-bit Transfer Width (n = 16)(209) |
| 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 SS port pin is not used by the SPI. In slave mode, the 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 SS port pin configuration, refer to Table 404. In master mode, a change of this bit aborts 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 aborts a transmission in progress and force the SPI into idle state.<br>0 Output buffer disabled.<br>1 Output buffer enabled. |
| 1<br>- | **Reserved** — For internal use |
| 0<br>SPC0 | **Serial Pin Control Bit 0** — This bit enables bidirectional pin configurations as shown in Table 407. In master mode, a change of this bit aborts a transmission in progress and force the SPI system into idle state. |

Note:

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

**Table 407. 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 | |

#### 5.39.3.2.3 SPI Baud Rate Register (SPIBR)

**Table 408. SPI Baud Rate Register (SPIBR)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | SPPR2 | SPPR1 | SPPR0 | 0 | SPR2 | SPR1 | SPR0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

Read: Anytime

Write: Anytime; writes to the reserved bits have no effect

**Table 409. SPIBR Field Descriptions**

| Field | Description |
|---|---|
| 6–4<br>SPPR[2:0] | **SPI Baud Rate Preselection Bits** — These bits specify the SPI baud rates as shown in Table 410. In master mode, a change of these bits aborts a transmission in progress and force the SPI system into idle state. |
| 2–0<br>SPR[2:0] | **SPI Baud Rate Selection Bits** — These bits specify the SPI baud rates as shown in Table 410. In master mode, a change of these bits aborts a transmission in progress and force the SPI system into idle state. |

The baud rate divisor equation is as follows:

$$\text{BaudRateDivisor} = (\text{SPPR} + 1) \bullet 2^{(\text{SPR} + 1)} \qquad \textit{Eqn. 97}$$

The baud rate can be calculated with the following equation:

$$\text{Baud Rate} = \text{BusClock} / \text{BaudRateDivisor} \qquad \textit{Eqn. 98}$$

**NOTE**

For maximum allowed baud rates, refer to the SPI Electrical Specification in the Electricals chapter of this data sheet.

**Table 410. Example SPI Baud Rate Selection (25 MHz Bus Clock)**

| SPPR2 | SPPR1 | SPPR0 | SPR2 | SPR1 | SPR0 | Baud Rate Divisor | Baud Rate |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 2 | 12.5 Mbit/s |
| 0 | 0 | 0 | 0 | 0 | 1 | 4 | 6.25 Mbit/s |
| 0 | 0 | 0 | 0 | 1 | 0 | 8 | 3.125 Mbit/s |
| 0 | 0 | 0 | 0 | 1 | 1 | 16 | 1.5625 Mbit/s |
| 0 | 0 | 0 | 1 | 0 | 0 | 32 | 781.25 kbit/s |
| 0 | 0 | 0 | 1 | 0 | 1 | 64 | 390.63 kbit/s |
| 0 | 0 | 0 | 1 | 1 | 0 | 128 | 195.31 kbit/s |
| 0 | 0 | 0 | 1 | 1 | 1 | 256 | 97.66 kbit/s |
| 0 | 0 | 1 | 0 | 0 | 0 | 4 | 6.25 Mbit/s |
| 0 | 0 | 1 | 0 | 0 | 1 | 8 | 3.125 Mbit/s |
| 0 | 0 | 1 | 0 | 1 | 0 | 16 | 1.5625 Mbit/s |
| 0 | 0 | 1 | 0 | 1 | 1 | 32 | 781.25 kbit/s |
| 0 | 0 | 1 | 1 | 0 | 0 | 64 | 390.63 kbit/s |
| 0 | 0 | 1 | 1 | 0 | 1 | 128 | 195.31 kbit/s |
| 0 | 0 | 1 | 1 | 1 | 0 | 256 | 97.66 kbit/s |
| 0 | 0 | 1 | 1 | 1 | 1 | 512 | 48.83 kbit/s |
| 0 | 1 | 0 | 0 | 0 | 0 | 6 | 4.16667 Mbit/s |
| 0 | 1 | 0 | 0 | 0 | 1 | 12 | 2.08333 Mbit/s |
| 0 | 1 | 0 | 0 | 1 | 0 | 24 | 1.04167 Mbit/s |
| 0 | 1 | 0 | 0 | 1 | 1 | 48 | 520.83 kbit/s |
| 0 | 1 | 0 | 1 | 0 | 0 | 96 | 260.42 kbit/s |
| 0 | 1 | 0 | 1 | 0 | 1 | 192 | 130.21 kbit/s |
| 0 | 1 | 0 | 1 | 1 | 0 | 384 | 65.10 kbit/s |
| 0 | 1 | 0 | 1 | 1 | 1 | 768 | 32.55 kbit/s |
| 0 | 1 | 1 | 0 | 0 | 0 | 8 | 3.125 Mbit/s |
| 0 | 1 | 1 | 0 | 0 | 1 | 16 | 1.5625 Mbit/s |
| 0 | 1 | 1 | 0 | 1 | 0 | 32 | 781.25 kbit/s |
| 0 | 1 | 1 | 0 | 1 | 1 | 64 | 390.63 kbit/s |
| 0 | 1 | 1 | 1 | 0 | 0 | 128 | 195.31 kbit/s |
| 0 | 1 | 1 | 1 | 0 | 1 | 256 | 97.66 kbit/s |
| 0 | 1 | 1 | 1 | 1 | 0 | 512 | 48.83 kbit/s |
| 0 | 1 | 1 | 1 | 1 | 1 | 1024 | 24.41 kbit/s |
| 1 | 0 | 0 | 0 | 0 | 0 | 10 | 2.5 Mbit/s |
| 1 | 0 | 0 | 0 | 0 | 1 | 20 | 1.25 Mbit/s |

**Table 410. Example SPI Baud Rate Selection (25 MHz Bus Clock) (continued)**

| SPPR2 | SPPR1 | SPPR0 | SPR2 | SPR1 | SPR0 | Baud Rate Divisor | Baud Rate |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 0 | 40 | 625 kbit/s |
| 1 | 0 | 0 | 0 | 1 | 1 | 80 | 312.5 kbit/s |
| 1 | 0 | 0 | 1 | 0 | 0 | 160 | 156.25 kbit/s |
| 1 | 0 | 0 | 1 | 0 | 1 | 320 | 78.13 kbit/s |
| 1 | 0 | 0 | 1 | 1 | 0 | 640 | 39.06 kbit/s |
| 1 | 0 | 0 | 1 | 1 | 1 | 1280 | 19.53 kbit/s |
| 1 | 0 | 1 | 0 | 0 | 0 | 12 | 2.08333 Mbit/s |
| 1 | 0 | 1 | 0 | 0 | 1 | 24 | 1.04167 Mbit/s |
| 1 | 0 | 1 | 0 | 1 | 0 | 48 | 520.83 kbit/s |
| 1 | 0 | 1 | 0 | 1 | 1 | 96 | 260.42 kbit/s |
| 1 | 0 | 1 | 1 | 0 | 0 | 192 | 130.21 kbit/s |
| 1 | 0 | 1 | 1 | 0 | 1 | 384 | 65.10 kbit/s |
| 1 | 0 | 1 | 1 | 1 | 0 | 768 | 32.55 kbit/s |
| 1 | 0 | 1 | 1 | 1 | 1 | 1536 | 16.28 kbit/s |
| 1 | 1 | 0 | 0 | 0 | 0 | 14 | 1.78571 Mbit/s |
| 1 | 1 | 0 | 0 | 0 | 1 | 28 | 892.86 kbit/s |
| 1 | 1 | 0 | 0 | 1 | 0 | 56 | 446.43 kbit/s |
| 1 | 1 | 0 | 0 | 1 | 1 | 112 | 223.21 kbit/s |
| 1 | 1 | 0 | 1 | 0 | 0 | 224 | 111.61 kbit/s |
| 1 | 1 | 0 | 1 | 0 | 1 | 448 | 55.80 kbit/s |
| 1 | 1 | 0 | 1 | 1 | 0 | 896 | 27.90 kbit/s |
| 1 | 1 | 0 | 1 | 1 | 1 | 1792 | 13.95 kbit/s |
| 1 | 1 | 1 | 0 | 0 | 0 | 16 | 1.5625 Mbit/s |
| 1 | 1 | 1 | 0 | 0 | 1 | 32 | 781.25 kbit/s |
| 1 | 1 | 1 | 0 | 1 | 0 | 64 | 390.63 kbit/s |
| 1 | 1 | 1 | 0 | 1 | 1 | 128 | 195.31 kbit/s |
| 1 | 1 | 1 | 1 | 0 | 0 | 256 | 97.66 kbit/s |
| 1 | 1 | 1 | 1 | 0 | 1 | 512 | 48.83 kbit/s |
| 1 | 1 | 1 | 1 | 1 | 0 | 1024 | 24.41 kbit/s |
| 1 | 1 | 1 | 1 | 1 | 1 | 2048 | 12.21 kbit/s |

#### 5.39.3.2.4 SPI Status Register (SPISR)

**Table 411. SPI Status Register (SPISR)**

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

= Unimplemented or Reserved

Read: Anytime

Write: Has no effect

**Table 412. SPISR Field Descriptions**

| Field | Description |
|---|---|
| 7<br>SPIF | **SPIF Interrupt Flag** — This bit is set after received data has been transferred into the SPI data register. For information about clearing SPIF Flag, refer to Table 413.<br>0 Transfer not yet complete.<br>1 New data copied to SPIDR. |
| 5<br>SPTEF | **SPI Transmit Empty Interrupt Flag** — If set, this bit indicates the transmit data register is empty. For information about clearing this bit and placing data into the transmit data register, refer to Table 414.<br>0 SPI data register not empty.<br>1 SPI data register empty. |
| 4<br>MODF | **Mode Fault Flag** — This bit is set if the SS input becomes low while the SPI is configured as a master and mode fault detection is enabled, MODFEN bit of SPICR2 register is set. Refer to MODFEN bit description in Section 5.39.3.2.2, "SPI Control Register 2 (SPICR2)". The flag is cleared automatically by a read of the SPI status register (with MODF set) followed by a write to the SPI control register 1.<br>0 Mode fault has not occurred.<br>1 Mode fault has occurred. |

**Table 413. SPIF Interrupt Flag Clearing Sequence**

| XFRW Bit | SPIF Interrupt Flag Clearing Sequence | | | |
|---|---|---|---|---|
| 0 | Read SPISR with SPIF = 1 | then | Read SPIDRL | |
| 1 | Read SPISR with SPIF = 1 | then | Byte Read SPIDRL (210) | |
| | | | or | |
| | | | Byte Read SPIDRH (211) | Byte Read SPIDRL |
| | | | or | |
| | | | Word Read (SPIDRH:SPIDRL) | |

Note:

210. Data in SPIDRH is lost in this case.

211. SPIDRH can be read repeatedly without any effect on SPIF. SPIF Flag is cleared only by the read of SPIDRL after reading SPISR with SPIF = 1.

**Table 414. SPTEF Interrupt Flag Clearing Sequence**

| XFRW Bit | SPTEF Interrupt Flag Clearing Sequence | | |
|---|---|---|---|
| 0 | Read SPISR with SPTEF = 1 | then | Write to SPIDRL (212) |
| 1 | Read SPISR with SPTEF = 1 | then | Byte Write to SPIDRL (212)(213) |
| | | | or |
| | | | Byte Write to SPIDRH (212)(214) then Byte Write to SPIDRL (212) |
| | | | or |
| | | | Word Write to (SPIDRH:SPIDRL) (212) |

Note:
212. Any write to SPIDRH or SPIDRL with SPTEF = 0 is effectively ignored.
213. Data in SPIDRH is undefined in this case.
214. SPIDRH can be written repeatedly without any effect on SPTEF. SPTEF Flag is cleared only by writing to SPIDRL after reading SPISR with SPTEF = 1.

#### 5.39.3.2.5 SPI Data Register (SPIDR = SPIDRH:SPIDRL)

**Table 415. SPI Data Register High (SPIDRH)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | R15 | R14 | R13 | R12 | R11 | R10 | R9 | R8 |
| W | T15 | T14 | T13 | T12 | T11 | T10 | T9 | T8 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 416. SPI Data Register Low (SPIDRL)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 |
| W | T7 | T6 | T5 | T4 | T3 | T2 | T1 | T0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Read: Anytime; read data only valid when SPIF is set

Write: Anytime

The SPI data register is both the input and output register for SPI data. A write to this register allows data to be queued and transmitted. For an SPI configured as a master, queued data is transmitted immediately after the previous transmission has completed. The SPI transmitter empty flag SPTEF in the SPISR register indicates when the SPI data register is ready to accept new data.

Received data in the SPIDR is valid when SPIF is set.

If SPIF is cleared and data has been received, the received data is transferred from the receive shift register to the SPIDR and SPIF is set.

If SPIF is set and not serviced, and a second data value has been received, the second received data is kept as valid data in the receive shift register until the start of another transmission. The data in the SPIDR does not change.

If SPIF is set and valid data is in the receive shift register, and SPIF is serviced before the start of a third transmission, the data in the receive shift register is transferred into the SPIDR and SPIF remains set (see Figure 99).

If SPIF is set and valid data is in the receive shift register, and SPIF is serviced after the start of a third transmission, the data in the receive shift register has become invalid and is not transferred into the SPIDR (see Figure 100).

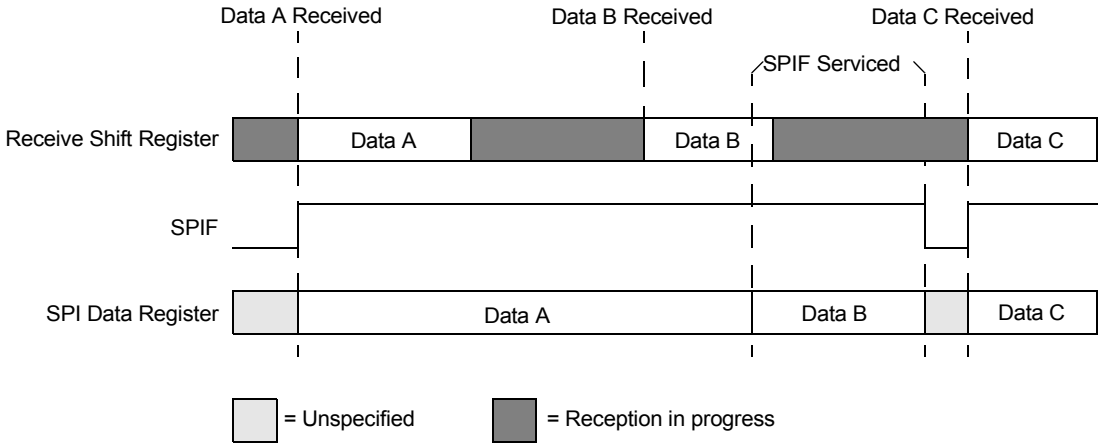



**Figure 99. Reception with SPIF Serviced in Time**

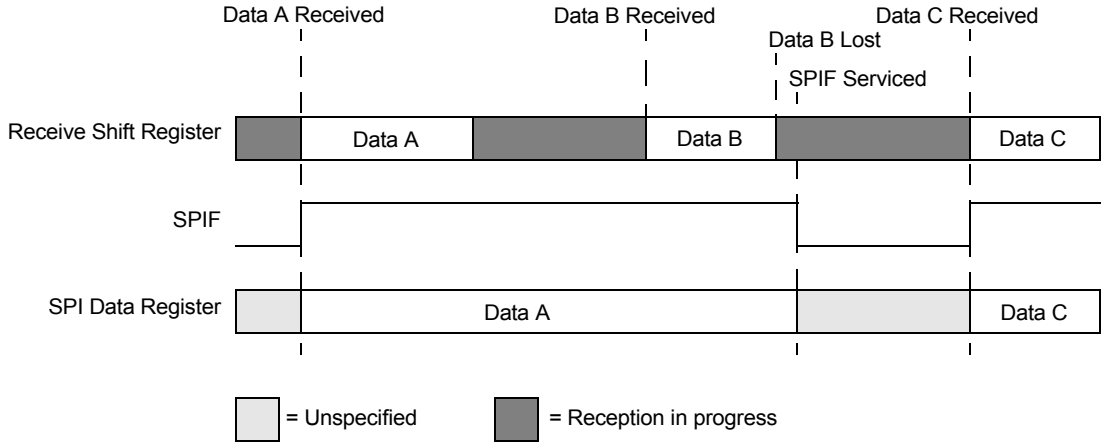



**Figure 100. Reception with SPIF Serviced Too Late**

### 5.39.4 Functional Description

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

The SPI system is enabled by setting the SPI enable (SPE) bit in SPI control register 1. While SPE is set, the four associated SPI port pins are dedicated to the SPI function as:

- Slave select ($\overline{SS}$)
- Serial clock (SCK)
- Master out/slave in (MOSI)
- Master in/slave out (MISO)

The main element of the SPI system is the SPI data register. The n-bit[(215)] data register in the master and the n-bit[(215)] data register in the slave are linked by the MOSI and MISO pins to form a distributed 2n-bit[(215)] register. When a data transfer operation is performed, this 2n-bit[(215)] register is serially shifted n[(215)] bit positions by the S-clock from the master, so data is exchanged between the master and the slave. Data written to the master SPI data register becomes the output data for the slave, and data read from the master SPI data register after a transfer operation is the input data from the slave.

A read of SPISR with SPTEF = 1 followed by a write to SPIDR puts data into the transmit data register. When a transfer is complete and SPIF is cleared, received data is moved into the receive data register. This data register acts as the SPI receive data register for reads and as the SPI transmit data register for writes. A common SPI data register address is shared for reading data from the read data buffer and for writing data to the transmit data register.

The clock phase control bit (CPHA) and a clock polarity control bit (CPOL) in the SPI control register 1 (SPICR1) select one of four possible clock formats to be used by the SPI system. The CPOL bit simply selects a non-inverted or inverted clock. The CPHA bit is used to accommodate two fundamentally different protocols by sampling data on odd numbered SCK edges or on even numbered SCK edges (see Section 5.39.4.3, “Transmission Formats").

The SPI can be configured to operate as a master or as a slave. When the MSTR bit in SPI control register1 is set, master mode is selected, when the MSTR bit is clear, slave mode is selected.

Note:

215. n depends on the selected transfer width, refer to Section 5.39.3.2.2, “SPI Control Register 2 (SPICR2)"

**NOTE**

A change of CPOL or MSTR bit while there is a received byte pending in the receive shift register destroys the received byte and must be avoided.

### 5.39.4.1 Master Mode

The SPI operates in master mode when the MSTR bit is set. Only a master SPI module can initiate transmissions. A transmission begins by writing to the master SPI data register. If the shift register is empty, data immediately transfers to the shift register. Data begins shifting out on the MOSI pin under the control of the serial clock.

- Serial clock
  The SPR2, SPR1, and SPR0 baud rate selection bits, in conjunction with the SPPR2, SPPR1, and SPPR0 baud rate preselection bits in the SPI baud rate register, control the baud rate generator and determine the speed of the transmission. The SCK pin is the SPI clock output. Through the SCK pin, the baud rate generator of the master controls the shift register of the slave peripheral.
- MOSI, MISO pin
  In master mode, the function of the serial data output pin (MOSI) and the serial data input pin (MISO) is determined by the SPC0 and BIDIROE control bits.
- $\overline{SS}$ pin
  If MODFEN and SSOE are set, the $\overline{SS}$ pin is configured as slave select output. The $\overline{SS}$ output becomes low during each transmission and is high when the SPI is in idle state.
  If MODFEN is set and SSOE is cleared, the $\overline{SS}$ pin is configured as input for detecting mode fault error. If the $\overline{SS}$ input becomes low this indicates a mode fault error where another master tries to drive the MOSI and SCK lines. In this case, the SPI immediately switches to slave mode, by clearing the MSTR bit and also disables the slave output buffer MISO (or SISO in bidirectional mode). So the result is all outputs are disabled and SCK, MOSI, and MISO are inputs. If a transmission is in progress when the mode fault occurs, the transmission is aborted and the SPI is forced into idle state.
  This mode fault error also sets the mode fault (MODF) flag in the SPI status register (SPISR). If the SPI interrupt enable bit (SPIE) is set when the MODF flag becomes set, then an SPI interrupt sequence is also requested.
  When a write to the SPI data register in the master occurs, there is a half SCK-cycle delay. After the delay, SCK is started within the master. The rest of the transfer operation differs slightly, depending on the clock format specified by the SPI clock phase bit, CPHA, in SPI control register 1 (see Section 5.39.4.3, “Transmission Formats").

**NOTE**

A change of the bits CPOL, CPHA, SSOE, LSBFE, XFRW, MODFEN, SPC0, or BIDIROE with SPC0 set, SPPR2-SPPR0 and SPR2-SPR0 in master mode aborts a transmission in progress and force the SPI into idle state. The remote slave cannot detect this, therefore the master must ensure the remote slave is returned to idle state.

### 5.39.4.2 Slave Mode

The SPI operates in slave mode when the MSTR bit in SPI control register 1 is clear.

- Serial clock
  In slave mode, SCK is the SPI clock input from the master.
- MISO, MOSI pin
  In slave mode, the function of the serial data output pin (MISO) and serial data input pin (MOSI) is determined by the SPC0 bit and BIDIROE bit in SPI control register 2.
- $\overline{SS}$ pin
  The $\overline{SS}$ pin is the slave select input. Before a data transmission occurs, the $\overline{SS}$ pin of the slave SPI must be low. $\overline{SS}$ must remain low until the transmission is complete. If $\overline{SS}$ goes high, the SPI is forced into idle state.
  The $\overline{SS}$ input also controls the serial data output pin, if $\overline{SS}$ is high (not selected), the serial data output pin is high impedance, and, if $\overline{SS}$ is low, the first bit in the SPI data register is driven out of the serial data output pin. Also, if the slave is not selected ($\overline{SS}$ is high), then the SCK input is ignored and no internal shifting of the SPI shift register occurs.
  Although the SPI is capable of duplex operation, some SPI peripherals are capable of only receiving SPI data in a slave mode. For these simpler devices, there is no serial data out pin.

**NOTE**

When peripherals with duplex capability are used, take care not to simultaneously enable two receivers whose serial outputs drive the same system slave’s serial data output line.

As long as no more than one slave device drives the system slave's serial data output line, it is possible for several slaves to receive the same transmission from a master, although the master would not receive return information from all of the receiving slaves.

If the CPHA bit in SPI control register 1 is clear, odd numbered edges on the SCK input cause the data at the serial data input pin to be latched. Even numbered edges cause the value previously latched from the serial data input pin to shift into the LSB or MSB of the SPI shift register, depending on the LSBFE bit.

If the CPHA bit is set, even numbered edges on the SCK input cause the data at the serial data input pin to be latched. Odd numbered edges cause the value previously latched from the serial data input pin to shift into the LSB or MSB of the SPI shift register, depending on the LSBFE bit.

When CPHA is set, the first edge is used to get the first data bit onto the serial data output pin. When CPHA is clear and the $\overline{SS}$ input is low (slave selected), the first bit of the SPI data is driven out of the serial data output pin. After the nth[(216)] shift, the transfer is considered complete and the received data is transferred into the SPI data register. To indicate transfer is complete, the SPIF flag in the SPI status register is set.

Note:

216. n depends on the selected transfer width, refer to Section 5.39.3.2.2, "SPI Control Register 2 (SPICR2)"

**NOTE**

A change of the bits CPOL, CPHA, SSOE, LSBFE, MODFEN, SPC0, or BIDIROE with SPC0 set in slave mode corrupts a transmission in progress and must be avoided.

### 5.39.4.3 Transmission Formats

During an SPI transmission, data is transmitted (shifted out serially) and received (shifted in serially) simultaneously. The serial clock (SCK) synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows selection of an individual slave SPI device; slave devices which are not selected do not interfere with SPI bus activities. Optionally, on a master SPI device, the slave select line can be used to indicate multiple-master bus contention.

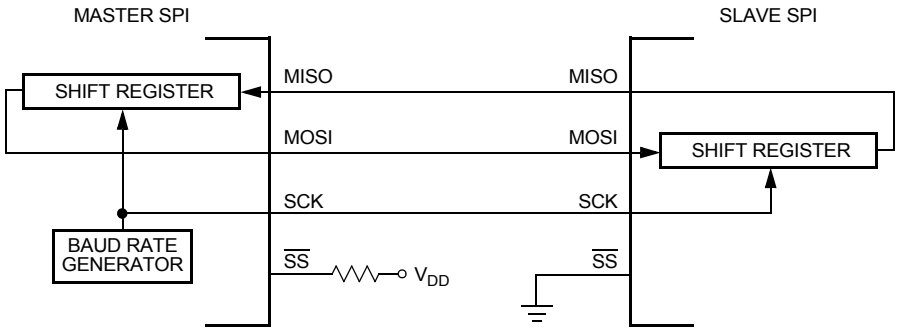



**Figure 101. Master/Slave Transfer Block Diagram**

#### 5.39.4.3.1 Clock Phase and Polarity Controls

Using two bits in the SPI control register 1, software selects one of four combinations of serial clock phase and polarity.

The CPOL clock polarity control bit specifies an active high or low clock and has no significant effect on the transmission format.

The CPHA clock phase control bit selects one of two fundamentally different transmission formats.

Clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

#### 5.39.4.3.2 CPHA = 0 Transfer Format

The first edge on the SCK line is used to clock the first data bit of the slave into the master and the first data bit of the master into the slave. In some peripherals, the first bit of the slave's data is available at the slave's data out pin as soon as the slave is selected. In this format, the first SCK edge is issued a half cycle after $\overline{SS}$ has become low.

A half SCK cycle later, the second edge appears on the SCK line. When this second edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the shift register, depending on LSBFE bit.

After this second edge, the next bit of the SPI master data is transmitted out of the serial data output pin of the master to the serial input pin on the slave. This process continues for a total of 16 edges on the SCK line, with data being latched on odd numbered edges and shifted on even numbered edges.

Data reception is double buffered. Data is shifted serially into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in.

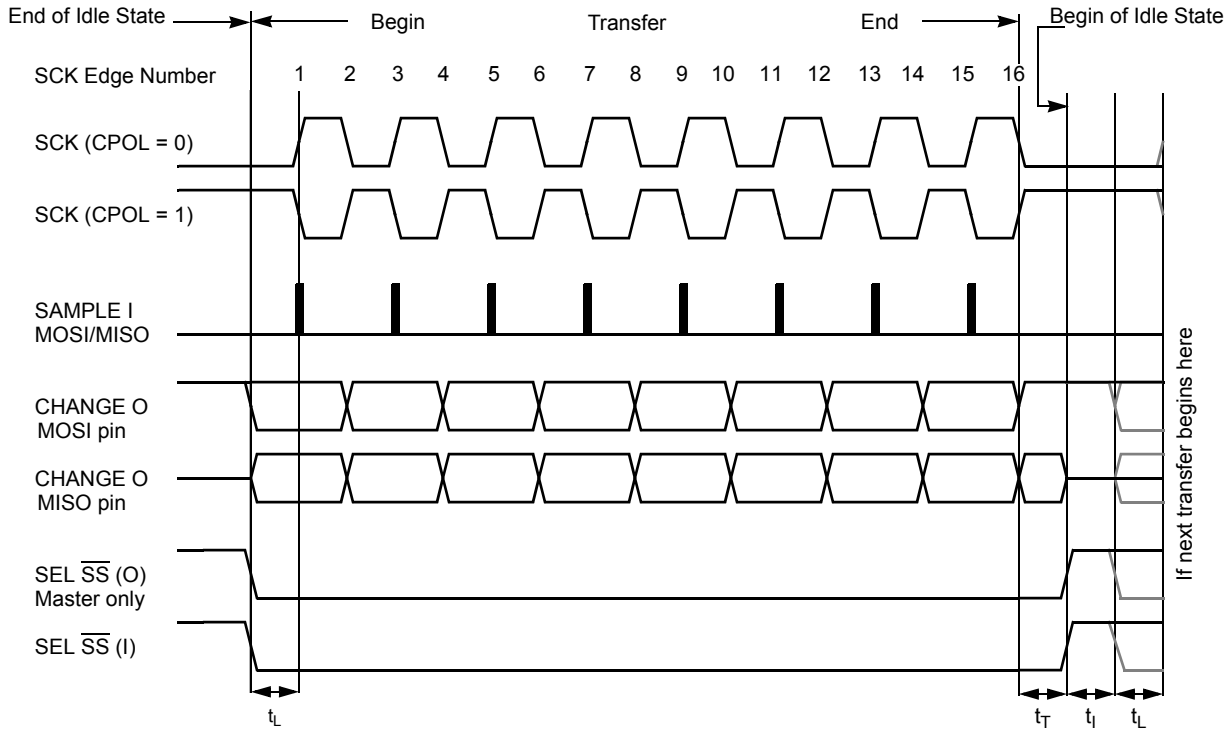
After 2n[(217)] (last) SCK edges:

- Data which was previously in the master SPI data register should now be in the slave data register and the data in the slave data register should be in the master.
- The SPIF flag in the SPI status register is set, indicating the transfer is complete.

Figure 102 is a timing diagram of an SPI transfer where CPHA = 0. SCK waveforms are shown for CPOL = 0 and CPOL = 1. The diagram may be interpreted as a master or slave timing diagram because the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave and the MOSI signal is the output from the master. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general-purpose output not affecting the SPI.

Note:

217. n depends on the selected transfer width, refer to Section 5.39.3.2.2, “SPI Control Register 2 (SPICR2)"



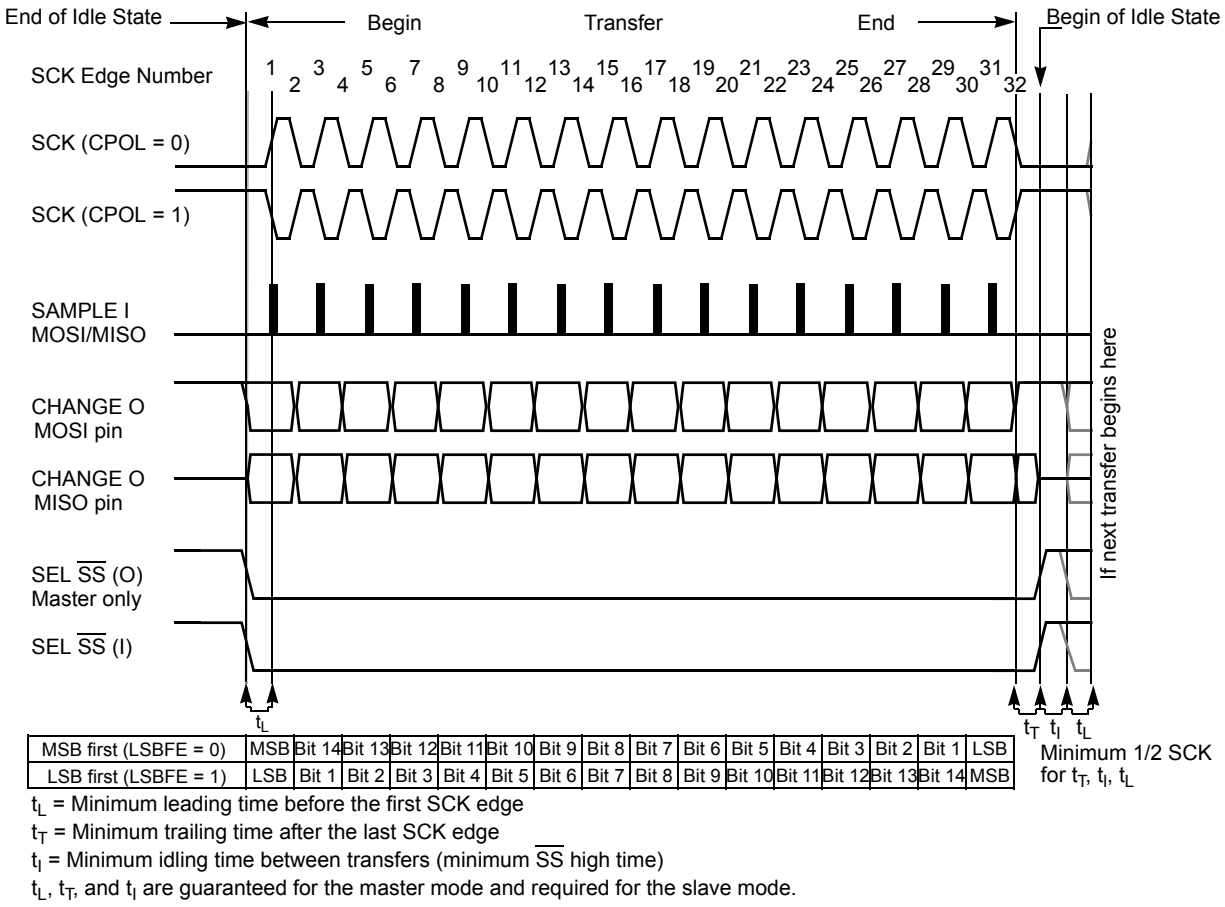

$t_L$ = Minimum leading time before the first SCK edge

$t_T$ = Minimum trailing time after the last SCK edge

$t_I$ = Minimum idling time between transfers (minimum $\overline{SS}$ high time)

$t_L$, $t_T$, and $t_I$ are guaranteed for the master mode and required for the slave mode.

**Figure 102. SPI Clock Format 0 (CPHA = 0), with 8-bit Transfer Width Selected (XFRW = 0)**




$t_L$ = Minimum leading time before the first SCK edge

$t_T$ = Minimum trailing time after the last SCK edge

$t_I$ = Minimum idling time between transfers (minimum $\overline{SS}$ high time)

$t_L$, $t_T$, and $t_I$ are guaranteed for the master mode and required for the slave mode.

**Figure 103. SPI Clock Format 0 (CPHA = 0), with 16-Bit Transfer Width Selected (XFRW = 1)**

In slave mode, if the $\overline{SS}$ line is not deasserted between the successive transmissions then the content of the SPI data register is not transmitted; instead the last received data is transmitted. If the $\overline{SS}$ line is deasserted for at least minimum idle time (half SCK cycle) between successive transmissions, then the content of the SPI data register is transmitted.

In master mode, with slave select output enabled the $\overline{SS}$ line is always deasserted and reasserted between successive transfers for at least minimum idle time.

#### 5.39.4.3.3 CPHA = 1 Transfer Format

Some peripherals require the first SCK edge before the first data bit becomes available at the data out pin, the second edge clocks data into the system. In this format, the first SCK edge is issued by setting the CPHA bit at the beginning of the n[(218)]-cycle transfer operation.

The first edge of SCK occurs immediately after the half SCK clock cycle synchronization delay. This first edge commands the slave to transfer its first data bit to the serial data input pin of the master.

A half SCK cycle later, the second edge appears on the SCK pin. This is the latching edge for both the master and slave.

When the third edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the SPI shift register, depending on LSBFE bit. After this edge, the next bit of the master data is coupled out of the serial data output pin of the master to the serial input pin on the slave.

This process continues for a total of n edges on the SCK line with data being latched on even numbered edges and shifting taking place on odd numbered edges.

Note:

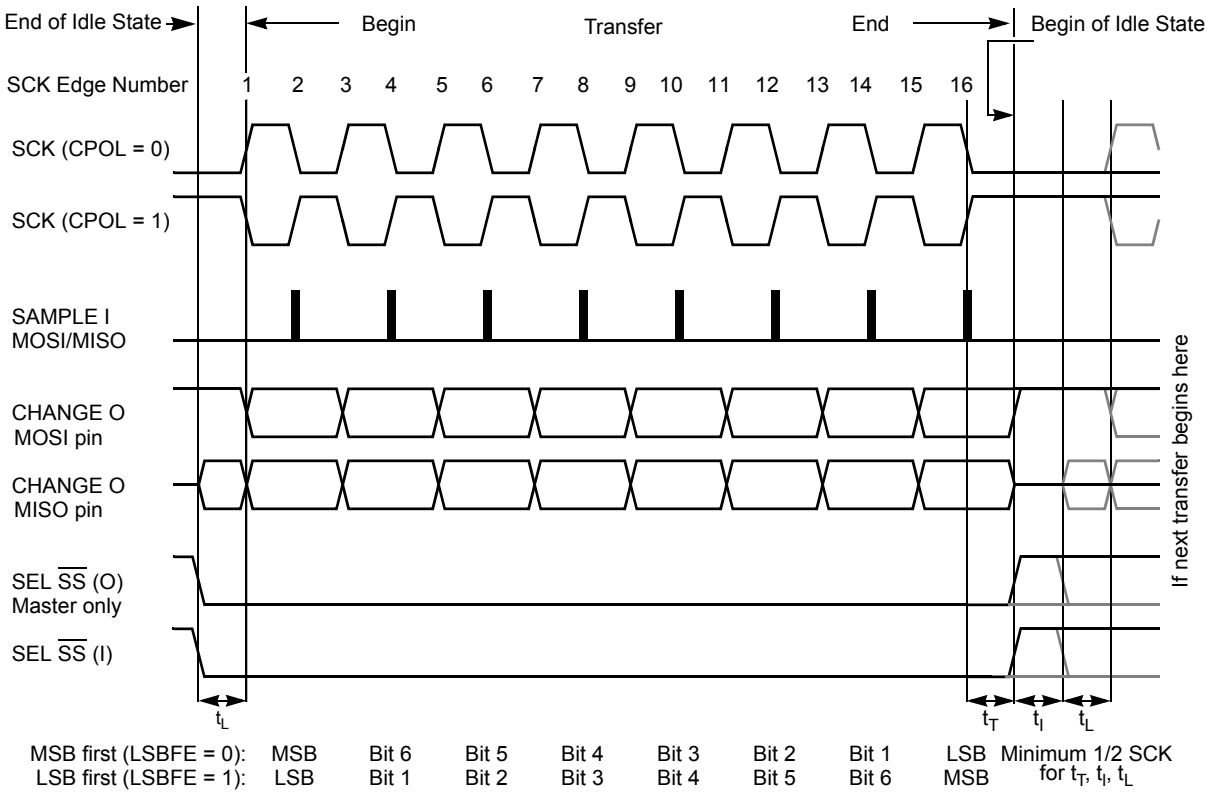
218. n depends on the selected transfer width, refer to Section 5.39.3.2.2, “SPI Control Register 2 (SPICR2)"

Data reception is double buffered, data is serially shifted into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in.

After 2n SCK edges:

- Data which was previously in the SPI data register of the master is now in the data register of the slave, and data in the data register of the slave is in the master.
- The SPIF flag bit in SPISR is set indicating the transfer is complete.

Figure 104 shows two clocking variations for CPHA = 1. The diagram may be interpreted as a master or slave timing diagram because the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The $\overline{SS}$ line is the slave select input to the slave. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general-purpose output not affecting the SPI.




$t_L$ = Minimum leading time before the first SCK edge, not required for back-to-back transfers

$t_T$ = Minimum trailing time after the last SCK edge

$t_I$ = Minimum idling time between transfers (minimum $\overline{SS}$ high time), not required for back-to-back transfers

**Figure 104. SPI Clock Format 1 (CPHA = 1), with 8-Bit Transfer Width Selected (XFRW = 0)**

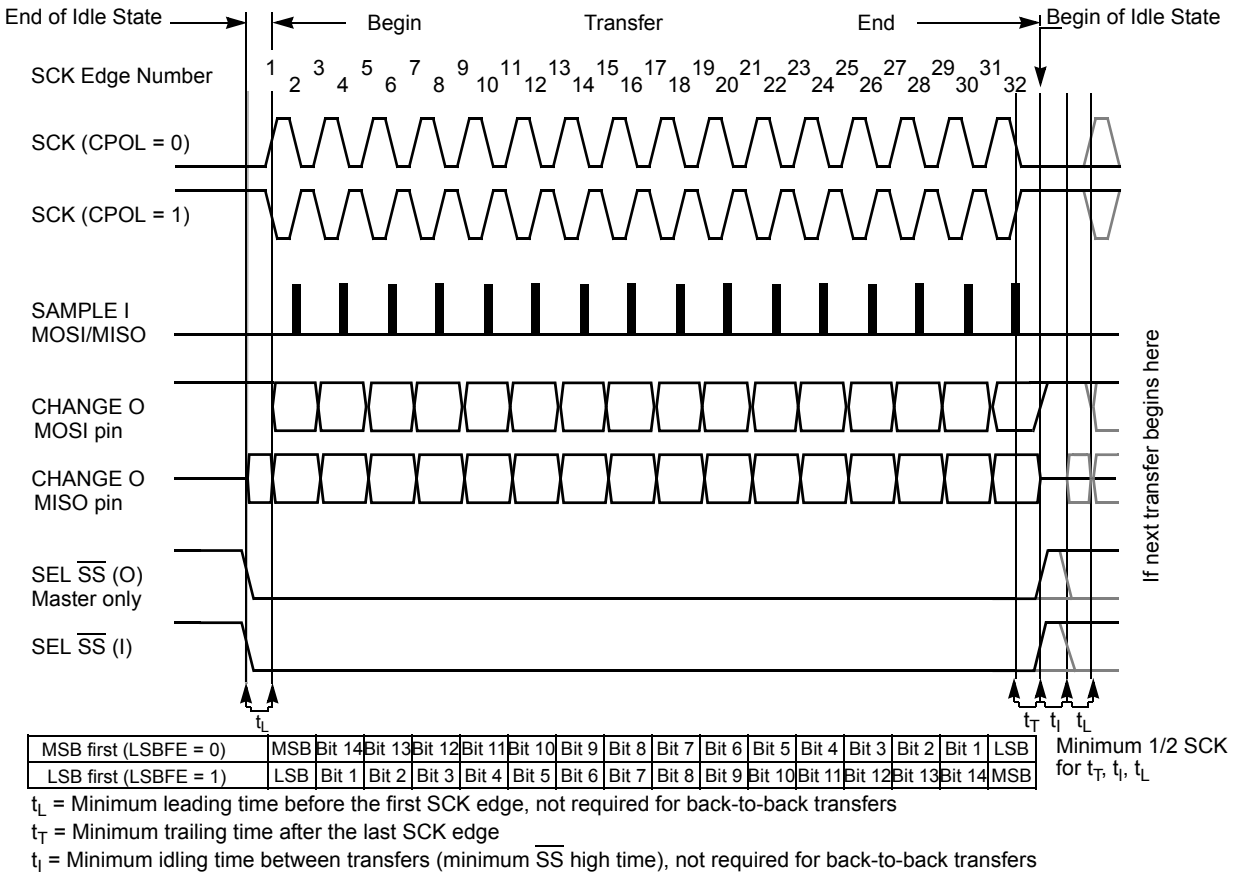



**Figure 105. SPI Clock Format 1 (CPHA = 1), with 16-Bit Transfer Width Selected (XFRW = 1)**

The $\overline{SS}$ line can remain active low between successive transfers (can be tied low at all times). This format is sometimes preferred in systems having a single fixed master and a single slave driving the MISO data line.

Back-to-back transfers in master mode

In master mode, if a transmission has completed and new data is available in the SPI data register, this data is sent out immediately without a trailing and minimum idle time.

The SPI interrupt request flag (SPIF) is common to both the master and slave modes. SPIF gets set one half SCK cycle after the last SCK edge.

### 5.39.4.4 SPI Baud Rate Generation

Baud rate generation consists of a series of divider stages. Six bits in the SPI baud rate register (SPPR2, SPPR1, SPPR0, SPR2, SPR1, and SPR0) determine the divisor to the SPI module clock which results in the SPI baud rate.

The SPI clock rate is determined by the product of the value in the baud rate preselection bits (SPPR2–SPPR0) and the value in the baud rate selection bits (SPR2–SPR0). The module clock divisor equation is shown in Equation 106.

$$\text{BaudRateDivisor} = (\text{SPPR} + 1) \bullet 2^{(\text{SPR} + 1)} \qquad \textit{Eqn. 106}$$

When all bits are clear (the default condition), the SPI module clock is divided by 2. When the selection bits (SPR2–SPR0) are 001 and the preselection bits (SPPR2–SPPR0) are 000, the module clock divisor becomes 4. When the selection bits are 010, the module clock divisor becomes 8, etc.

When the preselection bits are 001, the divisor determined by the selection bits is multiplied by 2. When the preselection bits are 010, the divisor is multiplied by 3, etc. See Table 410 for baud rate calculations for all bit conditions, based on a 25 MHz bus clock. The two sets of selects allows the clock to be divided by a non-power of two to achieve other baud rates such as divide by 6, divide by 10, etc.

The baud rate generator is activated only when the SPI is in master mode and a serial transfer is taking place. In the other cases, the divider is disabled to decrease $I_{DD}$ current.

**NOTE**

For maximum allowed baud rates, refer to the SPI Electrical Specification in the Electricals chapter of this data sheet.

### 5.39.4.5 Special Features

#### 5.39.4.5.1 $\overline{SS}$ Output

The $\overline{SS}$ output feature automatically drives the $\overline{SS}$ pin low during transmission to select external devices and drives it high during idle to deselect external devices. When $\overline{SS}$ output is selected, the $\overline{SS}$ output pin is connected to the $\overline{SS}$ input pin of the external device. The $\overline{SS}$ output is available only in master mode during normal SPI operation by asserting SSOE and MODFEN bit as shown in Table 404. The mode fault feature is disabled while $\overline{SS}$ output is enabled.

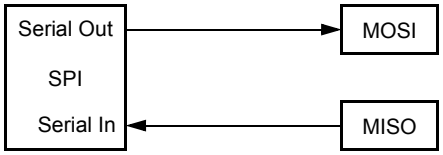
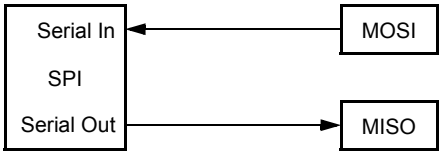
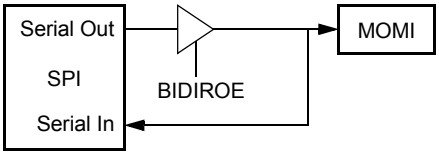
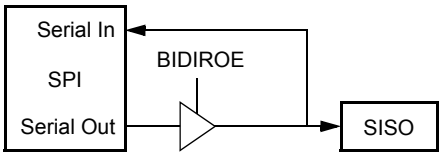
**NOTE**

Care must be taken when using the $\overline{SS}$ output feature in a multimaster system because the mode fault feature is not available for detecting system errors between masters.

#### 5.39.4.5.2 Bidirectional Mode (MOMI or SISO)

The bidirectional mode is selected when the SPC0 bit is set in SPI control register 2 (see Table 417). In this mode, the SPI uses only one serial data pin for the interface with external device(s). The MSTR bit decides which pin to use. The MOSI pin becomes the serial data I/O (MOMI) pin for the master mode, and the MISO pin becomes serial data I/O (SISO) pin for the slave mode. The MISO pin in master mode and MOSI pin in slave mode are not used by the SPI.

**Table 417. Normal Mode and Bidirectional Mode**

| When SPE = 1 | Master Mode MSTR = 1 | Slave Mode MSTR = 0 |
|---|---|---|
| **Normal Mode SPC0 = 0** |  |  |
| **Bidirectional Mode SPC0 = 1** |  |  |

The direction of each serial I/O pin depends on the BIDIROE bit. If the pin is configured as an output, serial data from the shift register is driven out on the pin. The same pin is also the serial input to the shift register.

- The SCK is output for the master mode and input for the slave mode.
- The $\overline{SS}$ is the input or output for the master mode, and it is always the input for the slave mode.
- The bidirectional mode does not affect SCK and $\overline{SS}$ functions.

**NOTE**

In bidirectional master mode, with mode fault enabled, both data pins MISO and MOSI can be occupied by the SPI, though MOSI is normally used for transmissions in bidirectional mode and MISO is not used by the SPI. If a mode fault occurs, the SPI is automatically switched to slave mode. In this case MISO becomes occupied by the SPI and MOSI is not used. This must be considered, if the MISO pin is used for another purpose.

### 5.39.4.6 Error Conditions

The SPI has one error condition:

Mode fault error

#### 5.39.4.6.1 Mode Fault Error

If the $\overline{SS}$ input becomes low while the SPI is configured as a master, it indicates a system error where more than one master may be trying to drive the MOSI and SCK lines simultaneously. This condition is not permitted in normal operation, the MODF bit in the SPI status register is set automatically, provided the MODFEN bit is set.

In the special case where the SPI is in master mode and MODFEN bit is cleared, the $\overline{SS}$ pin is not used by the SPI. In this special case, the mode fault error function is inhibited and MODF remains cleared. In case the SPI system is configured as a slave, the $\overline{SS}$ pin is a dedicated input pin. Mode fault error doesn't occur in slave mode.

If a mode fault error occurs, the SPI is switched to slave mode, with the exception that the slave output buffer is disabled. So SCK, MISO, and MOSI pins are forced to be high impedance inputs to avoid any possibility of conflict with another output driver. A transmission in progress is aborted and the SPI is forced into idle state.

If the mode fault error occurs in the bidirectional mode for a SPI system configured in master mode, output enable of the MOMI (MOSI in bidirectional mode) is cleared if it was set. No mode fault error occurs in the bidirectional mode for SPI system configured in slave mode.

The mode fault flag is cleared automatically by a read of the SPI status register (with MODF set) followed by a write to SPI control register 1. If the mode fault flag is cleared, the SPI becomes a normal master or slave again.

**NOTE**

If a mode fault error occurs and a received data byte is pending in the receive shift register, this data byte is lost.

### 5.39.4.7 Low Power Mode Options

#### 5.39.4.7.1 SPI in Run Mode

In run mode with the SPI system enable (SPE) bit in the SPI control register clear, the SPI system is in a low-power, disabled state. SPI registers remain accessible, but clocks to the core of this module are disabled.

#### 5.39.4.7.2 SPI in Stop Mode

Stop mode is dependent on the system. The SPI enters stop mode when the module clock is disabled (held high or low). If the SPI is in master mode and exchanging data when the CPU enters stop mode, the transmission is frozen until the CPU exits stop mode. After stop, data to and from the external SPI is exchanged correctly. In slave mode, the SPI stays synchronized with the master.

The stop mode is not dependent on the SPISWAI bit.

#### 5.39.4.7.3 Reset

The reset values of registers and signals are described in Section 5.29.3, "Memory Map and Registers", which details the registers and their bit fields.

- If a data transmission occurs in slave mode after reset without a write to SPIDR, it transmits garbage, or the data last received from the master before the reset.
- Reading from the SPIDR after reset always reads zeros.

#### 5.39.4.7.4 Interrupts

The SPI only originates interrupt requests when the SPI is enabled (SPE bit in SPICR1 set). The following is a description of how the SPI makes a request and how the MCU should acknowledge this request. The interrupt vector offset and interrupt priority are chip dependent.

The interrupt flags MODF, SPIF, and SPTEF are logically ORed to generate an interrupt request.

##### 5.39.4.7.4.1 MODF

MODF occurs when the master detects an error on the $\overline{SS}$ pin. The master SPI must be configured for the MODF feature (see Table 404). After MODF is set, the current transfer is aborted and the following bit is changed:

MSTR = 0, The master bit in SPICR1 resets.

The MODF interrupt is reflected in the status register MODF flag. Clearing the flag also clears the interrupt. This interrupt stays active while the MODF flag is set. MODF has an automatic clearing process which is described in Section 5.39.3.2.4, "SPI Status Register (SPISR)".

#### 5.39.4.7.4.2 SPIF

SPIF occurs when new data has been received and copied to the SPI data register. After SPIF is set, it does not clear until it is serviced. SPIF has an automatic clearing process, which is described in Section 5.39.3.2.4, “SPI Status Register (SPISR)".

#### 5.39.4.7.4.3 SPTEF

SPTEF occurs when the SPI data register is ready to accept new data. After SPTEF is set, it does not clear until it is serviced. SPTEF has an automatic clearing process, which is described in Section 5.39.3.2.4, “SPI Status Register (SPISR)".

## 5.40 64 KByte Flash Module (S12FTMRC64K1V1)

### 5.40.1 Introduction

The module implements the following:

- kbytes of P-Flash (Program Flash) memory
- kbytes of D-Flash (Data Flash) memory

The Flash memory is ideal for single-supply applications allowing for field reprogramming without requiring external high voltage sources for program or erase operations. The Flash module includes a memory controller which executes commands to modify Flash memory contents. The user interface to the memory controller consists of the indexed Flash Common Command Object (FCCOB) register which is written to with the command, global address, data, and any required command parameters. The memory controller must complete the execution of a command before the FCCOB register can be written to with a new command.

**CAUTION**

A Flash word or phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash word or phrase is not allowed.

The Flash memory may be read as bytes, aligned words, or misaligned words. Read access time is one bus cycle for bytes and aligned words, and two bus cycles for misaligned words. For Flash memory, an erased bit reads 1 and a programmed bit reads 0.

It is possible to read from P-Flash memory while some commands are executing on D-Flash memory. It is not possible to read from D-Flash memory while a command is executing on P-Flash memory. Simultaneous P-Flash and D-Flash operations are discussed in Allowed Simultaneous P-Flash and D-Flash Operations.

Both P-Flash and D-Flash memories are implemented with Error Correction Codes (ECC) which can resolve single bit faults and detect double bit faults. For P-Flash memory, the ECC implementation requires programming be done on an aligned 8 byte basis (a Flash phrase). Since P-Flash memory is always read by half-phrase, only one single bit fault in an aligned 4 byte half-phrase containing the byte or word accessed is corrected.

#### 5.40.1.1 Glossary

**Command Write Sequence** — An MCU instruction sequence to execute built-in algorithms (including program and erase) on the Flash memory.

**D-Flash Memory** — The D-Flash memory constitutes the nonvolatile memory store for data.

**D-Flash Sector** — The D-Flash sector is the smallest portion of the D-Flash memory which can be erased. The D-Flash sector consists of four 64 byte rows for a total of 256 bytes.

**NVM Command Mode** — An NVM mode using the CPU to setup the FCCOB register to pass parameters required for Flash command execution.

**Phrase** — An aligned group of four 16-bit words within the P-Flash memory. Each phrase includes two sets of aligned double words with each set including 7 ECC bits for single bit fault correction and double bit fault detection within each double word.

**P-Flash Memory** — The P-Flash memory constitutes the main nonvolatile memory store for applications.

**P-Flash Sector** — The P-Flash sector is the smallest portion of the P-Flash memory which can be erased. Each P-Flash sector contains 512 bytes.

**Program IFR** — Nonvolatile information register located in the P-Flash block containing the Device ID, Version ID, and the Program Once field.

#### 5.40.1.2 Features

##### 5.40.1.2.1 P-Flash Features

- Single bit fault correction and double bit fault detection within a 32-bit double word during read operations
- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and phrase program operation

- Ability to read the P-Flash memory while programming a word in the D-Flash memory
- Flexible protection scheme to prevent accidental program or erase of P-Flash memory

#### 5.40.1.2.2 D-Flash Features

- Single bit fault correction and double bit fault detection within a word during read operations
- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and word program operation
- Protection scheme to prevent accidental program or erase of D-Flash memory
- Ability to program up to four words in a burst sequence

#### 5.40.1.2.3 Other Flash Module Features

- No external high-voltage power supply required for Flash memory program and erase operations
- Interrupt generation on Flash command completion and Flash error detection
- Security mechanism to prevent unauthorized access to the Flash memory

### 5.40.1.3 Block Diagram

The block diagram of the Flash module is shown in Figure 107.

## 5.40.2 External Signal Description

The Flash module contains no signals connected off-chip.

## 5.40.3 Memory Map and Registers

This section describes the memory map and registers for the Flash module. Read data from unimplemented memory space in the Flash module is undefined. Write access to unimplemented or reserved memory space in the Flash module is ignored by the Flash module.

### 5.40.3.1 Module Memory Map

The S12 architecture places the P-Flash memory between global addresses.The P-Flash memory map is shown in.

The FPROT register, described in P-Flash Protection Register (FPROT), can be set to protect regions in the Flash memory from accidental program or erase. The Flash memory addresses covered by these protectable regions are shown in the P-Flash memory map. The higher address region is mainly targeted to hold the boot loader code since it covers the vector space. Default protection settings as well as security information allowing the MCU to restrict access to the Flash module are stored in the Flash configuration field as described in Table 418.

**Table 418. Flash Configuration Field**

| Global Address | Size (Bytes) | Description |
|---|---|---|
| 0x3_FF00-0x3_FF07 | 8 | Back door Comparison Key<br>Refer to Section 5.40.4.5.11, “Verify Back Door Access Key Command", and Section 5.40.5.1, “Unsecuring the MCU using Back door Key Access" |
| 0x3_FF08-0x3_FF0B | 4 | Reserved |
| 0x3_FF0C | 1 | P-Flash Protection byte. Refer to Section 5.40.3.2.9, “P-Flash Protection Register (FPROT)" |
| 0x3_FF0D | 1 | D-Flash Protection byte. Refer to Section 5.40.3.2.11, “D-Flash Protection Register (DFPROT)" |
| 0x3_FF0E | 1 | Flash Nonvolatile byte. Refer to Section 5.40.3.2.17, “Flash Option Register (FOPT)" |
| 0x3_FF0F | 1 | Flash Security byte. Refer to Section 5.40.3.2.2, “Flash Security Register (FSEC)" |

Note:
219. 0x3FF08-0x3_FF0F form a Flash phrase and must be programmed in a single command write sequence. Each byte in the 0x3_FF08 - 0x3_FF0B reserved field should be programmed to 0xFF.

**Table 419. Program IFR Fields**

| Global Address | Size (Bytes) | Field Description |
|---|---|---|
| 0x0_4000 – 0x0_4007 | 8.0 | Reserved |
| 0x0_4008 – 0x0_40B5 | 174 | Reserved |
| 0x0_40B6 – 0x0_40B7 | 2.0 | Version ID |
| 0x0_40B8 – 0x0_40BF | 8.0 | Reserved |
| 0x0_40C0 – 0x0_40FF | 64 | Program Once Field<br>Refer to Section 5.40.4.5.6, “Program Once Command" |

Note:
220. For patch code storage, see Section 5.40.4.2, “IFR Version ID Word"
221. Used to track firmware patch versions, see Section 5.40.4.2, “IFR Version ID Word"

**Table 420. D-Flash and Memory Controller Resource Fields**

| Global Address | Size (Bytes) | Description |
|---|---|---|
| 0x0_4000 – 0x0_43FF | 1,024 | Reserved |
| 0x0_4400 – 0x0_53FF | 4,096 | D-Flash Memory |
| 0x0_5400 – 0x0_57FF | 1,024 | Reserved |
| 0x0_5800 – 0x0_5AFF | 768 | Memory Controller Scratch RAM |
| 0x0_5B00 – 0x0_5FFF | 1,280 | Reserved |
| 0x0_6000 – 0x0_67FF | 2,048 | Reserved |
| 0x0_6800 – 0x0_7FFF | 6,144 | Reserved |

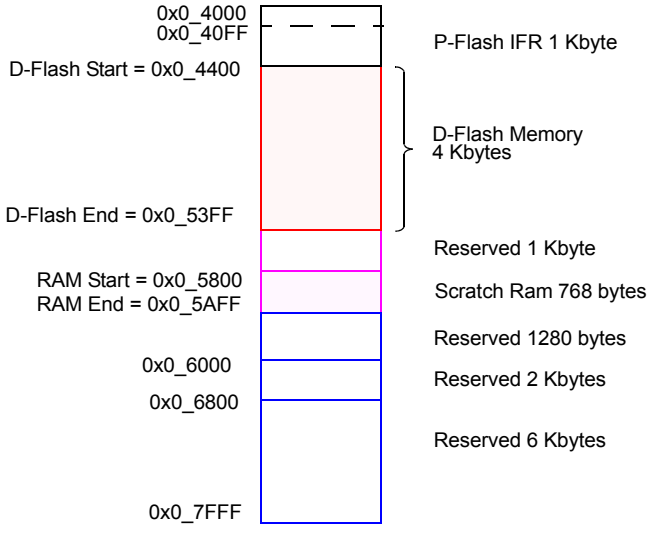



**Figure 107. D-Flash and Memory Controller Resource Memory Map**

### 5.40.3.2 Register Descriptions

The Flash module contains a set of 20 control and status registers located between Flash module base + 0x0000 and 0x0013. A summary of the Flash module registers is given in Figure 421 with detailed descriptions in the following subsections.

**CAUTION**

Writes to any Flash register must be avoided while a Flash command is active (CCIF=0) to prevent corruption of Flash register contents and adversely affect Memory Controller behavior.

**Table 421. FTMRC64K1 Register Summary**

| Address & Name | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| FCLKDIV | R | FDIVLD | FDIVLCK | FDIV5 | FDIV4 | FDIV3 | FDIV2 | FDIV1 | FDIV0 |
| | W | | | | | | | | |
| FSEC | R | KEYEN1 | KEYEN0 | RNV5 | RNV4 | RNV3 | RNV2 | SEC1 | SEC0 |
| | W | | | | | | | | |
| FCCOBIX | R | 0 | 0 | 0 | 0 | 0 | CCOBIX2 | CCOBIX1 | CCOBIX0 |
| | W | | | | | | | | |
| FRSV0 | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | W | | | | | | | | |
| FCNFG | R | CCIE | 0 | 0 | IGNSF | 0 | 0 | FDFD | FSFD |
| | W | | | | | | | | |
| FERCNFG | R | 0 | 0 | 0 | 0 | 0 | 0 | DFDIE | SFDIE |
| | W | | | | | | | | |
| FSTAT | R | CCIF | 0 | ACCERR | FPVIOL | MGBUSY | RSVD | MGSTAT1 | MGSTAT0 |
| | W | | | | | | | | |
| FERSTAT | R | 0 | 0 | 0 | 0 | 0 | 0 | DFDIF | SFDIF |
| | W | | | | | | | | |
| DFPROT | R | DPOPEN | 0 | 0 | 0 | DPS3 | DPS2 | DPS1 | DPS0 |
| | W | | | | | | | | |
| FCCOBHI | R | CCOB15 | CCOB14 | CCOB13 | CCOB12 | CCOB11 | CCOB10 | CCOB9 | CCOB8 |
| | W | | | | | | | | |
| FCCOBLO | R | CCOB7 | CCOB6 | CCOB5 | CCOB4 | CCOB3 | CCOB2 | CCOB1 | CCOB0 |
| | W | | | | | | | | |
| FRSV1 | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | W | | | | | | | | |
| FRSV2 | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | W | | | | | | | | |
| FRSV3 | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | W | | | | | | | | |

**Table 421. FTMRC64K1 Register Summary (continued)**

| Address & Name | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| FRSV4 | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | W | | | | | | | | |
| FOPT | R | NV7 | NV6 | NV5 | NV4 | NV3 | NV2 | NV1 | NV0 |
| | W | | | | | | | | |
| FRSV5 | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | W | | | | | | | | |
| FRSV6 | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | W | | | | | | | | |
| FRSV7 | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | W | | | | | | | | |

= Unimplemented or Reserved

#### 5.40.3.2.1 Flash Clock Divider Register (FCLKDIV)

The FCLKDIV register is used to control timed events in program and erase algorithms.

**Table 422. Flash Clock Divider Register (FCLKDIV)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | FDIVLD | FDIVLCK | FDIV[5:0] | | | | | |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

All bits in the FCLKDIV register are readable, bit 7 is not writable, bit 6 is write-once-hi and controls the writability of the FDIV field.

**CAUTION**

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

**Table 423. FCLKDIV Field Descriptions**

| Field | Description |
|---|---|
| 7<br>FDIVLD | **Clock Divider Loaded**<br>0 FCLKDIV register has not been written since the last reset<br>1 FCLKDIV register has been written since the last reset |
| 6<br>FDIVLCK | **Clock Divider Locked**<br>0 FDIV field is open for writing<br>1 FDIV value is locked and cannot be changed. Once the lock bit is set high, only reset can clear this bit and restore writability to the FDIV field. |
| 5–0<br>FDIV[5:0] | **Clock Divider Bits** — FDIV[5:0] must be set to effectively divide BUSCLK down to 1 MHz to control timed events during Flash program and erase algorithms. Table 424 shows recommended values for FDIV[5:0] based on the BUSCLK frequency. Refer to Section 5.40.4.3, "Flash Command Operations", for more information. |

**Table 424. FDIV values for various BUSCLK Frequencies**

| BUSCLK Frequency (MHz) | | FDIV[5:0] | BUSCLK Frequency (MHz) | | FDIV[5:0] |
|---|---|---|---|---|---|
| MIN[222] | MAX[223] | | MIN[222] | MAX[223] | |
| 1.0 | 1.6 | 0x00 | 16.6 | 17.6 | 0x10 |
| 1.6 | 2.6 | 0x01 | 17.6 | 18.6 | 0x11 |
| 2.6 | 3.6 | 0x02 | 18.6 | 19.6 | 0x12 |
| 3.6 | 4.6 | 0x03 | 19.6 | 20.6 | 0x13 |
| 4.6 | 5.6 | 0x04 | 20.6 | 21.6 | 0x14 |
| 5.6 | 6.6 | 0x05 | 21.6 | 22.6 | 0x15 |
| 6.6 | 7.6 | 0x06 | 22.6 | 23.6 | 0x16 |
| 7.6 | 8.6 | 0x07 | 23.6 | 24.6 | 0x17 |
| 8.6 | 9.6 | 0x08 | 24.6 | 25.6 | 0x18 |
| 9.6 | 10.6 | 0x09 | 25.6 | 26.6 | 0x19 |
| 10.6 | 11.6 | 0x0A | 26.6 | 27.6 | 0x1A |
| 11.6 | 12.6 | 0x0B | 27.6 | 28.6 | 0x1B |
| 12.6 | 13.6 | 0x0C | 28.6 | 29.6 | 0x1C |
| 13.6 | 14.6 | 0x0D | 29.6 | 30.6 | 0x1D |
| 14.6 | 15.6 | 0x0E | 30.6 | 31.6 | 0x1E |
| 15.6 | 16.6 | 0x0F | 31.6 | 32.6 | 0x1F |

Note:
222. BUSCLK is Greater Than this value.
223. BUSCLK is Less Than or Equal to this value.

#### 5.40.3.2.2 Flash Security Register (FSEC)

The FSEC register holds all bits associated with the security of the MCU and Flash module.

**Table 425. Flash Security Register (FSEC)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | KEYEN[1:0] | | RNV[5:2] | | | | SEC[1:0] | |
| W | | | | | | | | |
| Reset | F | F | F | F | F | F | F | F |

= Unimplemented or Reserved

All bits in the FSEC register are readable but not writable.

During the reset sequence, the FSEC register is loaded with the contents of the Flash security byte in the Flash configuration field at global address 0x3_FF0F located in P-Flash memory (see Table 418) as indicated by reset condition F in Figure 425. If a double bit fault is detected while reading the P-Flash phrase containing the Flash security byte during the reset sequence, all bits in the FSEC register is set to leave the Flash module in a secured state with back door key access disabled.

**Table 426. FSEC Field Descriptions**

| Field | Description |
|---|---|
| 7–6<br>KEYEN[1:0] | **Backdoor Key Security Enable Bits** — The KEYEN[1:0] bits define the enabling of back door key access to the Flash module as shown in Table 427. |
| 5–2<br>RNV[5:2} | **Reserved Nonvolatile Bits** — The RNV bits should remain in the erased state for future enhancements. |
| 1–0<br>SEC[1:0] | **Flash Security Bits** — The SEC[1:0] bits define the security state of the MCU as shown in Table 428. If the Flash module is unsecured using back door key access, the SEC bits are forced to 10. |

**Table 427. Flash KEYEN States**

| KEYEN[1:0] | Status of Back Door Key Access |
|---|---|
| 00 | DISABLED |
| 01 | DISABLED(224) |
| 10 | ENABLED |
| 11 | DISABLED |

Note:
224. Preferred KEYEN state to disable back door key access.

**Table 428. Flash Security States**

| SEC[1:0] | Status of Security |
|---|---|
| 00 | SECURED |
| 01 | SECURED(225) |
| 10 | UNSECURED |
| 11 | SECURED |

Note:
225. Preferred SEC state to set MCU to secured state

The security function in the Flash module is described in Section 5.40.5, “Security".

#### 5.40.3.2.3 Flash CCOB Index Register (FCCOBIX)

The FCCOBIX register is used to index the FCCOB register for Flash memory operations.

**Table 429. FCCOB Index Register (FCCOBIX)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | 0 | 0 | 0 | 0 | CCOBIX[2:0] | | |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

CCOBIX bits are readable and writable while remaining bits read 0 and are not writable.

**Table 430. FCCOBIX Field Descriptions**

| Field | Description |
|---|---|
| 2–0<br>CCOBIX[1:0] | **Common Command Register Index** — The CCOBIX bits are used to select which word of the FCCOB register array is being read or written to. See Section 5.40.3.2.12, “Flash Common Command Object Register (FCCOB)”, for more details. |

#### 5.40.3.2.4 Flash Reserved0 Register (FRSV0)

This Flash register is reserved for factory testing.

**Table 431. Flash Reserved0 Register (FRSV0)**

| | 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

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

#### 5.40.3.2.5 Flash Configuration Register (FCNFG)

The FCNFG register enables the Flash command complete interrupt and forces ECC faults on Flash array read access from the CPU.

**Table 432. Flash Configuration Register (FCNFG)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | CCIE | 0 | 0 | IGNSF | 0 | 0 | FDFD | FSFD |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

CCIE, IGNSF, FDFD, and FSFD bits are readable and writable while remaining bits read 0 and are not writable.

**Table 433. FCNFG Field Descriptions**

| Field | Description |
|---|---|
| 7<br>CCIE | **Command Complete Interrupt Enable** — The CCIE bit controls interrupt generation when a Flash command has completed.<br>0 Command complete interrupt disabled<br>1 An interrupt is requested whenever the CCIF flag in the FSTAT register is set (see Section 5.40.3.2.7, “Flash Status Register (FSTAT)”) |
| 4<br>IGNSF | **Ignore Single Bit Fault** — The IGNSF controls single bit fault reporting in the FERSTAT register (see Section 5.40.3.2.8, “Flash Error Status Register (FERSTAT)”).<br>0 All single bit faults detected during array reads are reported<br>1 Single bit faults detected during array reads are not reported and the single bit fault interrupt is not generated |

**Table 433. FCNFG Field Descriptions (continued)**

| Field | Description |
|---|---|
| 1<br>FDFD | **Force Double Bit Fault Detect** — The FDFD bit allows the user to simulate a double bit fault during Flash array read operations and check the associated interrupt routine. The FDFD bit is cleared by writing a 0 to FDFD. The FECCR registers are not updated during the Flash array read operation with FDFD set unless an actual double bit fault is detected.<br>0 Flash array read operations set the DFDIF flag in the FERSTAT register only if a double bit fault is detected<br>1 Any Flash array read operation forces the DFDIF flag in the FERSTAT register to be set (see Section 5.40.3.2.7, “Flash Status Register (FSTAT)") and an interrupt is generated as long as the DFDIE interrupt enable in the FERCNFG register is set (see Section 5.40.3.2.6, “Flash Error Configuration Register (FERCNFG)") |
| 0<br>FSFD | **Force Single Bit Fault Detect** — The FSFD bit allows the user to simulate a single bit fault during Flash array read operations and check the associated interrupt routine. The FSFD bit is cleared by writing a 0 to FSFD. The FECCR registers are not updated during the Flash array read operation with FSFD set unless an actual single bit fault is detected.<br>0 Flash array read operations set the SFDIF flag in the FERSTAT register only if a single bit fault is detected<br>1 Flash array read operation forces the SFDIF flag in the FERSTAT register to be set (see Section 5.40.3.2.7, “Flash Status Register (FSTAT)") and an interrupt is generated as long as the SFDIE interrupt enable in the FERCNFG register is set (see Section 5.40.3.2.6, “Flash Error Configuration Register (FERCNFG)") |

### 5.40.3.2.6 Flash Error Configuration Register (FERCNFG)

The FERCNFG register enables the Flash error interrupts for the FERSTAT flags.

**Table 434. Flash Error Configuration Register (FERCNFG)**

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

= Unimplemented or Reserved

All assigned bits in the FERCNFG register are readable and writable.

**Table 435. FERCNFG Field Descriptions**

| Field | Description |
|---|---|
| 1<br>DFDIE | **Double Bit Fault Detect Interrupt Enable** — The DFDIE bit controls interrupt generation when a double bit fault is detected during a Flash block read operation.<br>0 DFDIF interrupt disabled<br>1 An interrupt is requested whenever the DFDIF flag is set (see Section 5.40.3.2.8, “Flash Error Status Register (FERSTAT)") |
| 0<br>SFDIE | **Single Bit Fault Detect Interrupt Enable** — The SFDIE bit controls interrupt generation when a single bit fault is detected during a Flash block read operation.<br>0 SFDIF interrupt disabled whenever the SFDIF flag is set (see Section 5.40.3.2.8, “Flash Error Status Register (FERSTAT)")<br>1 An interrupt is requested whenever the SFDIF flag is set (see Section 5.40.3.2.8, “Flash Error Status Register (FERSTAT)") |

### 5.40.3.2.7 Flash Status Register (FSTAT)

The FSTAT register reports the operational status of the Flash module.

**Table 436. Flash Status Register (FSTAT)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | CCIF | 0 | ACCERR | FPVIOL | MGBUSY | RSVD | MGSTAT[1:0] | |
| W | | | | | | | | |
| Reset | 1 | 0 | 0 | 0 | 0 | 0 | 0(226) | 0(226) |

= Unimplemented or Reserved

Note:
226. Reset value can deviate from the value shown if a double bit fault is detected during the reset sequence (see Section 5.40.6, “Initialization").

CCIF, ACCERR, and FPVIOL bits are readable and writable, MGBUSY and MGSTAT bits are readable but not writable, while remaining bits read 0 and are not writable.

**Table 437. FSTAT Field Descriptions**

| Field | Description |
|---|---|
| 7<br>CCIF | **Command Complete Interrupt Flag** — The CCIF flag indicates a Flash command has completed. The CCIF flag is cleared by writing a 1 to CCIF to launch a command and CCIF stays low until command completion or command violation.<br>0 Flash command in progress<br>1 Flash command has completed |
| 5<br>ACCERR | **Flash Access Error Flag** — The ACCERR bit indicates an illegal access has occurred to the Flash memory caused by either a violation of the command write sequence (see Section 5.40.4.3.2, “Command Write Sequence") or issuing an illegal Flash command. While ACCERR is set, the CCIF flag cannot be cleared to launch a command. The ACCERR bit is cleared by writing a 1 to ACCERR. Writing a 0 to the ACCERR bit has no effect on ACCERR.<br>0 No access error detected<br>1 Access error detected |
| 4<br>FPVIOL | **Flash Protection Violation Flag** —The FPVIOL bit indicates an attempt was made to program or erase an address in a protected area of P-Flash or D-Flash memory during a command write sequence. The FPVIOL bit is cleared by writing a 1 to FPVIOL. Writing a 0 to the FPVIOL bit has no effect on FPVIOL. While FPVIOL is set, it is not possible to launch a command or start a command write sequence.<br>0 No protection violation detected<br>1 Protection violation detected |
| 3<br>MGBUSY | **Memory Controller Busy Flag** — The MGBUSY flag reflects the active state of the Memory Controller.<br>0 Memory Controller is idle<br>1 Memory Controller is busy executing a Flash command (CCIF = 0) |
| 2<br>RSVD | **Reserved Bit** — This bit is reserved and always reads 0. |
| 1–0<br>MGSTAT[1:0] | **Memory Controller Command Completion Status Flag** — One or more MGSTAT flag bits are set if an error is detected during execution of a Flash command or during the Flash reset sequence. See Section 5.40.4.5, “Flash Command Description" and Section 5.40.6, “Initialization" for details. |

#### 5.40.3.2.8 Flash Error Status Register (FERSTAT)

The FERSTAT register reflects the error status of internal Flash operations.

**Table 438. Flash Error Status Register (FERSTAT)**

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

= Unimplemented or Reserved

All flags in the FERSTAT register are readable and only writable to clear the flag.

**Table 439. FERSTAT Field Descriptions**

| Field | Description |
|---|---|
| 1<br>DFDIF | **Double Bit Fault Detect Interrupt Flag** — The setting of the DFDIF flag indicates a double bit fault was detected in the stored parity and data bits during a Flash array read operation, or a Flash array read operation was attempted on a Flash block under a Flash command operation.(227) 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 a single bit fault was detected in the stored parity and data bits during a Flash array read operation, or a Flash array read operation was attempted on a Flash block under a Flash command operation. (227) 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 |

Note:
227. The single bit fault and double bit fault flags are mutually exclusive for parity errors (an ECC fault occurrence can be either single fault or double fault, but never both). A simultaneous access collision (read attempted while command running) is indicated when both SFDIF and DFDIF flags are high.

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

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

**Table 440. P-Flash Protection Register (FPROT)**

| | 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

The (unreserved) bits of the FPROT register are writable with the restriction that the size of the protected region can only be increased.

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 0x3_FF0C located in P-Flash memory (see Table 418) as indicated by reset condition ‘F’ in Table 440. To change the P-Flash protection is 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 is cleared and remaining bits in the FPROT register is set to leave the P-Flash memory fully protected.

Trying to alter data in any protected area in the P-Flash memory results in a protection violation error and the FPVIOL bit is 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 441. 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 10-17 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 bid<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. |

**Table 441. FPROT Field Descriptions (continued)**

| Field | Description |
|---|---|
| 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 0x3_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 443. 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 0x3_8000.<br>0 Protection/Unprotection enabled<br>1 Protection/Unprotection disabled |
| 1-0<br>FPLS | **Flash Protection Lower Address Size** — The FPLS bits determine the size of the protected/unprotected area in P-Flash memory as shown in Table 444. The FPLS bits can only be written to while the FPLDIS bit is set. |

**Table 442. P-Flash Protection Function**

| FPOPEN | FPHDIS | FPLDIS | Function |
|---|---|---|---|
| 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 |

Note:

228. For range sizes, refer to Table 443 and Table 444.

**Table 443. P-Flash Protection Higher Address Range**

| FPHS[1:0] | Global Address Range | Protected Size |
|---|---|---|
| 00 | 0x3_F800–0x3_FFFF | 2.0 kByte |
| 01 | 0x3_F000–0x3_FFFF | 4.0 kByte |
| 10 | 0x3_E000–0x3_FFFF | 8.0 kByte |
| 11 | 0x3_C000–0x3_FFFF | 16 kByte |

**Table 444. P-Flash Protection Lower Address Range**

| FPLS[1:0] | Global Address Range | Protected Size |
|---|---|---|
| 00 | 0x3_8000–0x3_83FF | 1.0 kByte |
| 01 | 0x3_8000–0x3_87FF | 2.0 kByte |
| 10 | 0x3_8000–0x3_8FFF | 4.0 kByte |
| 11 | 0x3_8000–0x3_9FFF | 8.0 kByte |

All possible P-Flash protection scenarios are shown in Table 108. Although the protection scheme is loaded from the Flash memory at global address 0x3_FF0C during the reset sequence, it can be changed by the user. The P-Flash protection scheme can be used by applications requiring reprogramming in single chip mode while providing as much protection as possible if reprogramming is not required.

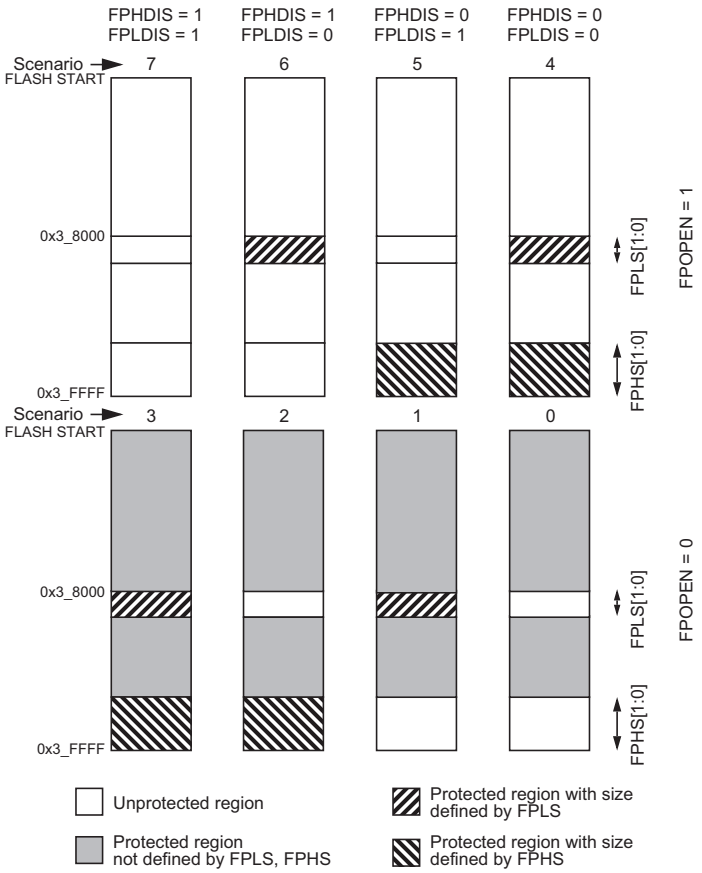



**Figure 108. P-Flash Protection Scenarios**

#### 5.40.3.2.10 P-Flash Protection Restrictions

The general guideline is P-Flash protection can only be added and not removed. Table 445 specifies all valid transitions between P-Flash protection scenarios. Any attempt to write an invalid scenario to the FPROT register is ignored. The contents of the FPROT register reflect the active protection scenario. See the FPHS and FPLS bit descriptions for additional restrictions.

**Table 445. P-Flash Protection Scenario Transitions**

| From Protection Scenario | To Protection Scenario [229] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | X | X | X | X | | | | |
| 1 | | X | | X | | | | |
| 2 | | | X | X | | | | |
| 3 | | | | X | | | | |
| 4 | | | | X | X | | | |
| 5 | | | X | X | X | X | | |
| 6 | | X | | X | X | | X | |

**Table 445. P-Flash Protection Scenario Transitions (continued)**

| From Protection Scenario | To Protection Scenario [229] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | X | X | X | X | | | | |
| 1 | | X | | X | | | | |
| 2 | | | X | X | | | | |
| 7 | X | X | X | X | X | X | X | X |

Note:
229. Allowed transitions marked with X, see Figure 10-14 for a definition of the scenarios.

#### 5.40.3.2.11 D-Flash Protection Register (DFPROT)

The DFPROT register defines which D-Flash sectors are protected against program and erase operations.

**Table 446. D-Flash Protection Register (DFPROT)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | DPOPEN | 0 | 0 | 0 | DPS[3:0] | | | |
| W | | | | | | | | |
| Reset | F | 0 | 0 | 0 | F | F | F | F |

= Unimplemented or Reserved

The (unreserved) bits of the DFPROT register are writable with the restriction that protection can be added but not removed. Writes must increase the DPS value and the DPOPEN bit can only be written from 1 (protection disabled) to 0 (protection enabled). If the DPOPEN bit is set, the state of the DPS bits is irrelevant.

During the reset sequence, the DFPROT register is loaded with the contents of the D-Flash protection byte in the Flash configuration field at global address 0x3_FF0D located in P-Flash memory (see Table 418) as indicated by reset condition F in Figure 446. To change the D-Flash protection is loaded during the reset sequence, the P-Flash sector containing the D-Flash protection byte must be unprotected, then the D-Flash protection byte must be programmed. If a double bit fault is detected while reading the P-Flash phrase containing the D-Flash protection byte during the reset sequence, the DPOPEN bit is cleared and DPS bits is set to leave the D-Flash memory fully protected.

Trying to alter data in any protected area in the D-Flash memory results in a protection violation error and the FPVIOL bit is set in the FSTAT register. Block erase of the D-Flash memory is not possible if any of the D-Flash sectors are protected.

**Table 447. DFPROT Field Descriptions**

| Field | Description |
|---|---|
| 7<br>DPOPEN | **D-Flash Protection Control**<br>0 Enables D-Flash memory protection from program and erase with protected address range defined by DPS bits<br>1 Disables D-Flash memory protection from program and erase |
| 3–0<br>DPS[3:0] | **D-Flash Protection Size** — The DPS[3:0] bits determine the size of the protected area in the D-Flash memory as shown in Table 448. |

#### 5.40.3.2.12 Flash Common Command Object Register (FCCOB)

**Table 448. D-Flash Protection Address Range**

| DPS[3:0] | Global Address Range | Protected Size |
|---|---|---|
| 0000 | 0x0_4400 – 0x0_44FF | 256 bytes |
| 0001 | 0x0_4400 – 0x0_45FF | 512 bytes |
| 0010 | 0x0_4400 – 0x0_46FF | 768 bytes |

**Table 448. D-Flash Protection Address Range (continued)**

| DPS[3:0] | Global Address Range | Protected Size |
|---|---|---|
| 0011 | 0x0_4400 – 0x0_47FF | 1024 bytes |
| 0100 | 0x0_4400 – 0x0_48FF | 1280 bytes |
| 0101 | 0x0_4400 – 0x0_49FF | 1536 bytes |
| 0110 | 0x0_4400 – 0x0_4AFF | 1792 bytes |
| 0111 | 0x0_4400 – 0x0_4BFF | 2048 bytes |
| 1000 | 0x0_4400 – 0x0_4CFF | 2304 bytes |
| 1001 | 0x0_4400 – 0x0_4DFF | 2560 bytes |
| 1010 | 0x0_4400 – 0x0_4EFF | 2816 bytes |
| 1011 | 0x0_4400 – 0x0_4FFF | 3072 bytes |
| 1100 | 0x0_4400 – 0x0_50FF | 3328 bytes |
| 1101 | 0x0_4400 – 0x0_51FF | 3584 bytes |
| 1110 | 0x0_4400 – 0x0_52FF | 3840 bytes |
| 1111 | 0x0_4400 – 0x0_53FF | 4096 bytes |

The FCCOB is an array of six words addressed via the CCOBIX index found in the FCCOBIX register. Byte wide reads and writes are allowed to the FCCOB register.

**Table 449. Flash Common Command Object High Register (FCCOBHI)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | CCOB[15:8] | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 450. Flash Common Command Object Low Register (FCCOBLO)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | CCOB[7:0] | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

#### 5.40.3.2.12.1 FCCOB - NVM Command Mode

NVM command mode uses the indexed FCCOB register to provide a command code and its relevant parameters to the Memory Controller. The user first sets up all required FCCOB fields and then initiates the command's execution by writing a 1 to the CCIF bit in the FSTAT register (a 1 written by the user clears the CCIF command completion flag to 0). When the user clears the CCIF bit in the FSTAT register all FCCOB parameter fields are locked and cannot be changed by the user until the command completes (as evidenced by the Memory Controller returning CCIF to 1). Some commands return information to the FCCOB register array.

The generic format for the FCCOB parameter fields in NVM command mode is shown in Table 451. The return values are available for reading after the CCIF flag in the FSTAT register has been returned to 1 by the Memory Controller. Writes to the unimplemented parameter fields (CCOBIX = 110 and CCOBIX = 111) are ignored with reads from these fields returning 0x0000.

Table 451 shows the generic Flash command format. The high byte of the first word in the CCOB array contains the command code, followed by the parameters for this specific Flash command. For details on the FCCOB settings required by each command, see the Flash command descriptions in Section 5.40.4.5, "Flash Command Description".

**Table 451. FCCOB - NVM Command Mode (Typical Usage)**

| CCOBIX[2:0] | Byte | FCCOB Parameter Fields (NVM Command Mode) |
|---|---|---|
| 000 | HI | FCMD[7:0] defining Flash command |
| | LO | 6'h0, Global address [17:16] |
| 001 | HI | Global address [15:8] |
| | LO | Global address [7:0] |
| 010 | HI | Data 0 [15:8] |
| | LO | Data 0 [7:0] |
| 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] |

#### 5.40.3.2.13 Flash Reserved1 Register (FRSV1)

This Flash register is reserved for factory testing.

**Table 452. Flash Reserved1 Register (FRSV1)**

| | 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

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

#### 5.40.3.2.14 Flash Reserved2 Register (FRSV2)

This Flash register is reserved for factory testing.

**Table 453. Flash Reserved2 Register (FRSV2)**

| | 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

All bits in the FRSV2 register read 0 and are not writable.

#### 5.40.3.2.15 Flash Reserved3 Register (FRSV3)

This Flash register is reserved for factory testing.

**Table 454. Flash Reserved3 Register (FRSV3)**

| | 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

All bits in the FRSV3 register read 0 and are not writable.

#### 5.40.3.2.16 Flash Reserved4 Register (FRSV4)

This Flash register is reserved for factory testing.

**Table 455. Flash Reserved4 Register (FRSV4)**

| | 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

All bits in the FRSV4 register read 0 and are not writable.

#### 5.40.3.2.17 Flash Option Register (FOPT)

The FOPT register is the Flash option register.

**Table 456. Flash Option Register (FOPT)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | NV[7:0] | | | | | | | |
| W | | | | | | | | |
| Reset | F | F | F | F | F | F | F | F |

= Unimplemented or Reserved

All bits in the FOPT register are readable but are not writable.

During the reset sequence, the FOPT register is loaded from the Flash nonvolatile byte in the Flash configuration field at global address 0x3_FF0E located in P-Flash memory (see Table 418) as indicated by reset condition F in Table 456. If a double bit fault is detected while reading the P-Flash phrase containing the Flash nonvolatile byte during the reset sequence, all bits in the FOPT register is set.

**Table 457. FOPT Field Descriptions**

| Field | Description |
|---|---|
| 7–0<br>NV[7:0] | **Nonvolatile Bits** — The NV[7:0] bits are available as nonvolatile bits. Refer to the device user guide for proper use of the NV bits. |

#### 5.40.3.2.18 Flash Reserved5 Register (FRSV5)

This Flash register is reserved for factory testing.

**Table 458. Flash Reserved5 Register (FRSV5)**

| | 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

All bits in the FRSV5 register read 0 and are not writable.

#### 5.40.3.2.19 Flash Reserved6 Register (FRSV6)

This Flash register is reserved for factory testing.

**Table 459. Flash Reserved6 Register (FRSV6)**

| | 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

All bits in the FRSV6 register read 0 and are not writable.

#### 5.40.3.2.20 Flash Reserved7 Register (FRSV7)

This Flash register is reserved for factory testing.

**Table 460. Flash Reserved7 Register (FRSV7)**

| | 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

All bits in the FRSV7 register read 0 and are not writable.

### 5.40.4 Functional Description

#### 5.40.4.1 Modes of Operation

The FTMRC64K1 module provides the modes of operation shown in Table 461. The operating mode is determined by module-level inputs and affects the FCLKDIV, FCNFG, and DFPROT registers, Scratch RAM writes, and the command set availability (see Table 463).

**Table 461. Modes and Mode Control Inputs**

| Operating Mode | FTMRC Input |
|---|---|
| | mmc_mode_ss_t2 |
| Normal: | 0 |
| Special: | 1 |

### 5.40.4.2 IFR Version ID Word

The version ID word is stored in the IFR at address 0x0_40B6. The contents of the word are defined in Table 462.

**Table 462. IFR Version ID Fields**

| [15:4] | [3:0] |
|---|---|
| Reserved | VERNUM |

VERNUM: Version number. The first version is number 0b_0001 with both 0b_0000 and 0b_1111 meaning ‘none’.

### 5.40.4.3 Flash Command Operations

Flash command operations are used to modify Flash memory contents.

The next sections describe:

- How to write the FCLKDIV register used to generate a time base (FCLK) derived from BUSCLK for Flash program and erase command operations
- The command write sequence used to set Flash command parameters and launch execution
- Valid Flash commands available for execution

#### 5.40.4.3.1 Writing the FCLKDIV Register

Prior to issuing any Flash program or erase command after a reset, the user is required to write the FCLKDIV register to divide BUSCLK down to a target FCLK of 1 MHz. Table 424 shows recommended values for the FDIV field based on BUSCLK frequency.

**NOTE**

Programming or erasing the Flash memory cannot be performed if the bus clock runs at less than 0.8 MHz. Setting FDIV too high can destroy the Flash memory due to overstress. Setting FDIV too low can result in incomplete programming or erasure of the Flash memory cells.

When the FCLKDIV register is written, the FDIVLD bit is set automatically. If the FDIVLD bit is 0, the FCLKDIV register has not been written since the last reset. If the FCLKDIV register has not been written, any Flash program or erase command loaded during a command write sequence does not execute and the ACCERR bit in the FSTAT register sets.

#### 5.40.4.3.2 Command Write Sequence

The Memory Controller launches all valid Flash commands entered using a command write sequence.

Before launching a command, the ACCERR and FPVIOL bits in the FSTAT register must be clear (see Section 5.40.3.2.7, “Flash Status Register (FSTAT)") and the CCIF flag should be tested to determine the status of the current command write sequence. If CCIF is 0, the previous command write sequence is still active, a new command write sequence cannot be started, and all writes to the FCCOB register are ignored.

**CAUTION**

Writes to any Flash register must be avoided while a Flash command is active (CCIF=0) to prevent corruption of Flash register contents and Memory Controller behavior.

##### 5.40.4.3.2.1 Define FCCOB Contents

The FCCOB parameter fields must be loaded with all required parameters for the Flash command being executed. Access to the FCCOB parameter fields is controlled via the CCOBIX bits in the FCCOBIX register (see Section 5.40.3.2.3, “Flash CCOB Index Register (FCCOBIX)").

The contents of the FCCOB parameter fields are transferred to the Memory Controller when the user clears the CCIF command completion flag in the FSTAT register (writing 1 clears the CCIF to 0). The CCIF flag remains clear until the Flash command has completed. Upon completion, the Memory Controller returns CCIF to 1 and the FCCOB register is used to communicate any results. The flow for a generic command write sequence is shown in Figure 109.

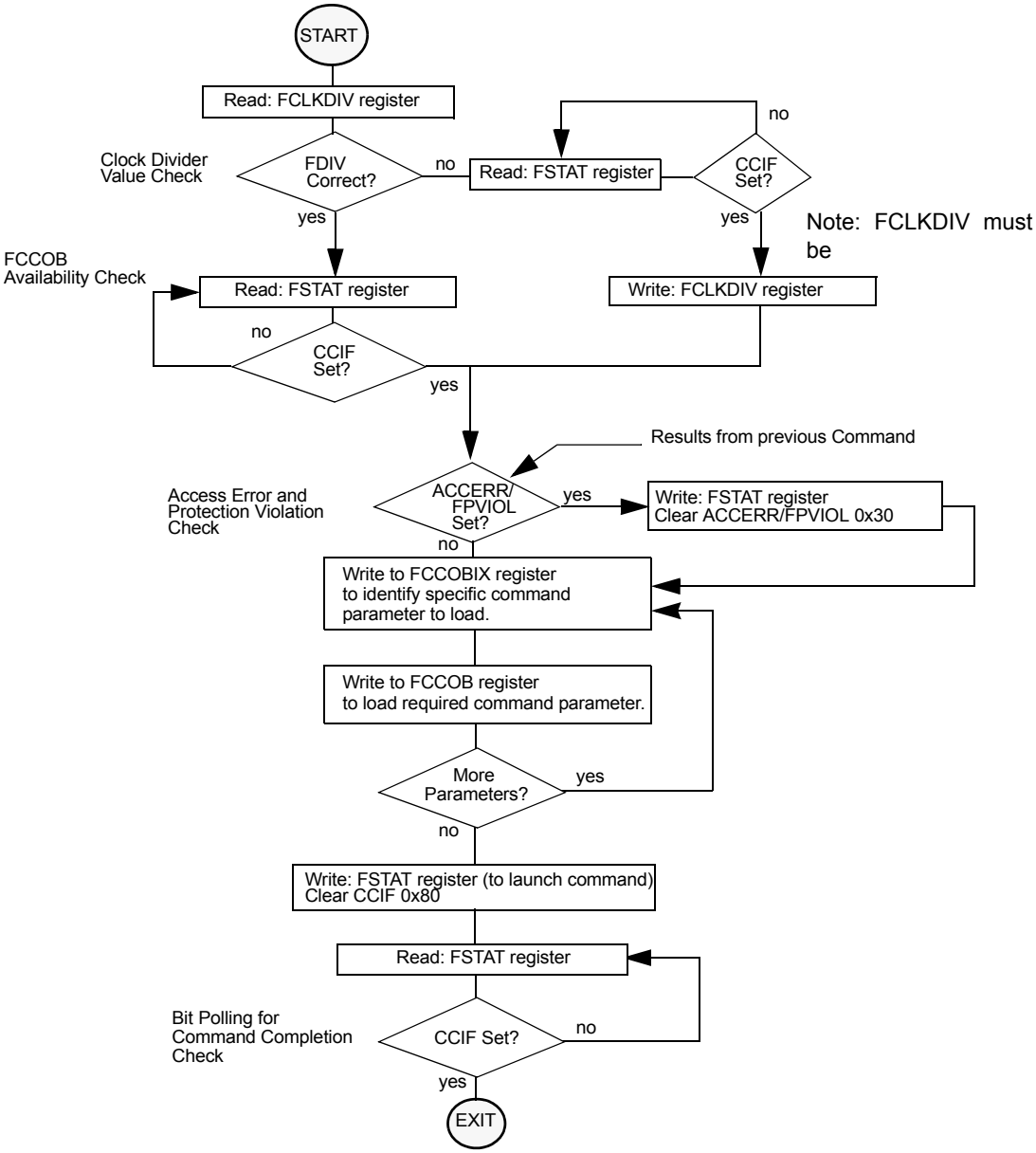



**Figure 109. Generic Flash Command Write Sequence Flowchart**

#### 5.40.4.3.3 Valid Flash Module Commands

**Table 463. Flash Commands by Mode**

| FCMD | Command | Unsecured | | Secured | |
|---|---|---|---|---|---|
| | | NS(230) | SS(231) | NS(232) | SS(233) |
| 0x01 | Erase Verify All Blocks | * | * | * | * |
| 0x02 | Erase Verify Block | * | * | * | * |
| 0x03 | Erase Verify P-Flash Section | * | * | * | |

**Table 463. Flash Commands by Mode (continued)**

| FCMD | Command | Unsecured | | Secured | |
|---|---|---|---|---|---|
| | | NS[230] | SS[231] | NS[232] | SS[233] |
| 0x04 | Read Once | * | * | * | |
| 0x06 | Program P-Flash | * | * | * | |
| 0x07 | Program Once | * | * | * | |
| 0x08 | Erase All Blocks | | * | | * |
| 0x09 | Erase Flash Block | * | * | * | |
| 0x0A | Erase P-Flash Sector | * | * | * | |
| 0x0B | Unsecure Flash | | * | | * |
| 0x0C | Verify Back door Access Key | * | | * | |
| 0x0D | Set User Margin Level | * | * | * | |
| 0x0E | Set Field Margin Level | | * | | |
| 0x10 | Erase Verify D-Flash Section | * | * | * | |
| 0x11 | Program D-Flash | * | * | * | |
| 0x12 | Erase D-Flash Sector | * | * | * | |

Note:
230. Unsecured Normal Single Chip mode
231. Unsecured Special Single Chip mode
232. Secured Normal Single Chip mode
233. Secured Special Single Chip mode

#### 5.40.4.3.4 P-Flash Commands

Table 464 summarizes the valid P-Flash commands along with the effects of the commands on the P-Flash block and other resources within the Flash module.

**Table 464. P-Flash Commands**

| FCMD | Command | Function on P-Flash Memory |
|---|---|---|
| 0x01 | Erase Verify All Blocks | Verify all P-Flash (and D-Flash) blocks are erased. |
| 0x02 | Erase Verify Block | Verify a P-Flash block is erased. |
| 0x03 | Erase Verify P-Flash Section | Verify a given number of words starting at the address provided are erased. |
| 0x04 | Read Once | Read a dedicated 64 byte field in the nonvolatile information register in P-Flash block previously programmed using the Program Once command. |
| 0x06 | Program P-Flash | Program a phrase in a P-Flash block. |
| 0x07 | Program Once | Program a dedicated 64 byte field in the nonvolatile information register in P-Flash block allowed to be programmed only once. |
| 0x08 | Erase All Blocks | Erase all P-Flash (and D-Flash) blocks.<br>An erase of all Flash blocks is only possible when the FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register and the DPOPEN bit in the DFPROT register are set prior to launching the command. |
| 0x09 | Erase Flash Block | Erase a P-Flash (or D-Flash) block.<br>An erase of the full P-Flash block is only possible when FPLDIS, FPHDIS and FPOPEN bits in the FPROT register are set prior to launching the command. |
| 0x0A | Erase P-Flash Sector | Erase all bytes in a P-Flash sector. |

**Table 464. P-Flash Commands (continued)**

| FCMD | Command | Function on P-Flash Memory |
|---|---|---|
| 0x0B | Unsecure Flash | Supports a method of releasing MCU security by erasing all P-Flash (and D-Flash) blocks and verifying all P-Flash (and D-Flash) blocks are erased. |
| 0x0C | Verify Back door Access Key | Supports a method of releasing MCU security by verifying a set of security keys. |
| 0x0D | Set User Margin Level | Specifies a user margin read level for all P-Flash blocks. |
| 0x0E | Set Field Margin Level | Specifies a field margin read level for all P-Flash blocks (special modes only). |

#### 5.40.4.3.5 D-Flash Commands

Table 465 summarizes the valid D-Flash commands along with the effects of the commands on the D-Flash block.

**Table 465. D-Flash Commands**

| FCMD | Command | Function on D-Flash Memory |
|---|---|---|
| 0x01 | Erase Verify All Blocks | Verify all D-Flash (and P-Flash) blocks are erased. |
| 0x02 | Erase Verify Block | Verify the D-Flash block is erased. |
| 0x08 | Erase All Blocks | Erase all D-Flash (and P-Flash) blocks.<br>An erase of all Flash blocks is only possible when the FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register and the DPOPEN bit in the DFPROT register are set prior to launching the command. |
| 0x09 | Erase Flash Block | Erase a D-Flash (or P-Flash) block.<br>An erase of the full D-Flash block is only possible when DPOPEN bit in the DFPROT register is set prior to launching the command. |
| 0x0B | Unsecure Flash | Supports a method of releasing MCU security by erasing all D-Flash (and P-Flash) blocks and verifying all D-Flash (and P-Flash) blocks are erased. |
| 0x0D | Set User Margin Level | Specifies a user margin read level for the D-Flash block. |
| 0x0E | Set Field Margin Level | Specifies a field margin read level for the D-Flash block (special modes only). |
| 0x10 | Erase Verify D-Flash Section | Verify a given number of words starting at the address provided are erased. |
| 0x11 | Program D-Flash | Program up to four words in the D-Flash block. |
| 0x12 | Erase D-Flash Sector | Erase all bytes in a sector of the D-Flash block. |

### 5.40.4.4 Allowed Simultaneous P-Flash and D-Flash Operations

Only the operations marked ‘OK’ in Table 466 are permitted to be run simultaneously on the Program Flash and Data Flash blocks. Some operations cannot be executed simultaneously because certain hardware resources are shared by the two memories. The priority has been placed on permitting Program Flash reads while program and erase operations execute on the Data Flash, providing read (P-Flash) while write (D-Flash) functionality.

**Table 466. Allowed P-Flash and D-Flash Simultaneous Operations**

| | Data Flash | | | | |
|---|---|---|---|---|---|
| **Program Flash** | **Read** | **Margin Read(234)** | **Program** | **Sector Erase** | **Mass Erase(236)** |
| **Read** | | OK | OK | OK | |
| **Margin Read(234)** | | OK(235) | | | |
| **Program** | | | | | |
| **Sector Erase** | | | | OK | |
| **Mass Erase(234)** | | | | | OK |

Note:

234. A 'Margin Read' is any read after executing the margin setting commands 'Set User Margin Level' or 'Set Field Margin Level' with anything but the 'normal' level specified.
235. See the Note on margin settings in Section 5.40.4.5.12, "Set User Margin Level Command" and Section 5.40.4.5.13, "Set Field Margin Level Command".
236. The 'Mass Erase' operations are commands 'Erase All Blocks' and 'Erase Flash Block'.

### 5.40.4.5 Flash Command Description

This section provides details of all available Flash commands launched by a command write sequence. The ACCERR bit in the FSTAT register is set during the command write sequence if any of the following illegal steps are performed, causing the command not to be processed by the Memory Controller:

- Starting any command write sequence which programs or erases Flash memory before initializing the FCLKDIV register
- Writing an invalid command as part of the command write sequence
- For additional possible errors, refer to the error handling table provided for each command

If a Flash block is read during execution of an algorithm (CCIF = 0) on the same block, the read operation returns invalid data. If the SFDIF or DFDIF flags were not previously set when the invalid read operation occurred, both the SFDIF and DFDIF flags is set.

If the ACCERR or FPVIOL bits are set in the FSTAT register, the user must clear these bits before starting any command write sequence (see Section 5.40.3.2.7, "Flash Status Register (FSTAT)").

**CAUTION**

A Flash word or phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash word or phrase is not allowed.

#### 5.40.4.5.1 Erase Verify All Blocks Command

The Erase Verify All Blocks command verifies all P-Flash and D-Flash blocks have been erased.

**Table 467. Erase Verify All Blocks Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x01 | Not required |

Upon clearing CCIF to launch the Erase Verify All Blocks command, the Memory Controller verifies the entire Flash memory space is erased. The CCIF flag sets after the Erase Verify All Blocks operation has completed.

**Table 468. Erase Verify All Blocks Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 000 at command launch |
| | FPVIOL | None |
| | MGSTAT1 | Set if any errors have been encountered during the read |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the read |

#### 5.40.4.5.2 Erase Verify Block Command

The Erase Verify Block command allows the user to verify an entire P-Flash or D-Flash block has been erased. The FCCOB upper global address bits determine which block must be verified.

**Table 469. Erase Verify Block Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x02 | Global address [17:16] of the Flash block to be verified. |

Upon clearing CCIF to launch the Erase Verify Block command, the Memory Controller verifies the selected P-Flash or D-Flash block is erased. The CCIF flag sets after the Erase Verify Block operation has completed.

**Table 470. Erase Verify Block Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 000 at command launch |
| | | Set if an invalid global address [17:16] is supplied |
| | FPVIOL | None |
| | MGSTAT1 | Set if any errors have been encountered during the read |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the read |

#### 5.40.4.5.3 Erase Verify P-Flash Section Command

The Erase Verify P-Flash Section command verifies a section of code in the P-Flash memory is erased. The Erase Verify P-Flash Section command defines the starting point of the code to be verified and the number of phrases.

**Table 471. Erase Verify P-Flash Section Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x03 | Global address [17:16] of a P-Flash block |
| 001 | Global address [15:0] of the first phrase to be verified | |
| 010 | Number of phrases to be verified | |

Upon clearing CCIF to launch the Erase Verify P-Flash Section command, the Memory Controller verifies the selected section of Flash memory is erased. The CCIF flag sets after the Erase Verify P-Flash Section operation has completed.

**Table 472. Erase Verify P-Flash Section Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 010 at command launch |
| | | Set if command not available in current mode (see Table 463) |
| | | Set if an invalid global address [17:0] is supplied |
| | | Set if a misaligned phrase address is supplied (global address [2:0] != 000) |
| | | Set if the requested section crosses a 128 Kbyte boundary |
| | FPVIOL | None |
| | MGSTAT1 | Set if any errors have been encountered during the read |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the read |

#### 5.40.4.5.4 Read Once Command

The Read Once command provides read access to a reserved 64 byte field (8 phrases) located in the nonvolatile information register of P-Flash. The Read Once field is programmed using the Program Once command described in Section 5.40.4.5.6, “Program Once Command". The Read Once command must not be executed from the Flash block containing the Program Once reserved field to avoid code runaway.

**Table 473. Read Once Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x04 | Not Required |
| 001 | Read Once phrase index (0x0000 - 0x0007) | |
| 010 | Read Once word 0 value | |
| 011 | Read Once word 1 value | |
| 100 | Read Once word 2 value | |
| 101 | Read Once word 3 value | |

Upon clearing CCIF to launch the Read Once command, a Read Once phrase is fetched and stored in the FCCOB indexed register. The CCIF flag sets after the Read Once operation has completed. Valid phrase index values for the Read Once command range from 0x0000 to 0x0007. During execution of the Read Once command, any attempt to read addresses within P-Flash block returns invalid data.

**Table 474. Read Once 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 463) |
| | | Set if an invalid phrase index is supplied |
| | FPVIOL | None |
| | MGSTAT1 | Set if any errors have been encountered during the read |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the read |

#### 5.40.4.5.5 Program P-Flash Command

The Program P-Flash operation programs a previously erased phrase in the P-Flash memory using an embedded algorithm.

**CAUTION**

A P-Flash phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash phrase is not allowed.

**Table 475. Program P-Flash Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x06 | Global address [17:16] to identify P-Flash block |
| 001 | Global address [15:0] of phrase location to be programmed[237] | |
| 010 | Word 0 program value | |
| 011 | Word 1 program value | |
| 100 | Word 2 program value | |
| 101 | Word 3 program value | |

Note:
237. Global address [2:0] must be 000

Upon clearing CCIF to launch the Program P-Flash command, the Memory Controller programs the data words to the supplied global address and then proceeds to verify the data words read back as expected. The CCIF flag sets after the Program P-Flash operation has completed.

**Table 476. Program P-Flash Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 101 at command launch |
| | | Set if command not available in current mode (see Table 463) |
| | | Set if an invalid global address [17:0] is supplied |
| | | Set if a misaligned phrase address is supplied (global address [2:0] != 000) |
| | FPVIOL | Set if the global address [17:0] points to a protected area |
| | MGSTAT1 | Set if any errors have been encountered during the verify operation |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the verify operation |

#### 5.40.4.5.6 Program Once Command

The Program Once command restricts programming to a reserved 64 byte field (8 phrases) in the nonvolatile information register located in P-Flash. The Program Once reserved field can be read using the Read Once command as described in Section 5.40.4.5.4, “Read Once Command". The Program Once command must only be issued once since the nonvolatile information register in P-Flash cannot be erased. The Program Once command must not be executed from the Flash block containing the Program Once reserved field to avoid code runaway.

**Table 477. Program Once Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x07 | Not Required |
| 001 | Program Once phrase index (0x0000 - 0x0007) | |
| 010 | Program Once word 0 value | |
| 011 | Program Once word 1 value | |
| 100 | Program Once word 2 value | |
| 101 | Program Once word 3 value | |

Upon clearing CCIF to launch the Program Once command, the Memory Controller first verifies the selected phrase is erased. If erased, then the selected phrase is programmed and then verified with read back. The CCIF flag remains clear, setting only after the Program Once operation has completed.

The reserved nonvolatile information register accessed by the Program Once command cannot be erased and any attempt to program one of these phrases a second time does not allowed. Valid phrase index values for the Program Once command range from 0x0000 to 0x0007. During execution of the Program Once command, any attempt to read addresses within P-Flash returns invalid data.

**Table 478. Program Once Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 101 at command launch |
| | | Set if command not available in current mode (see Table 463) |
| | | Set if an invalid phrase index is supplied |
| | | Set if the requested phrase has already been programmed(238) |
| | FPVIOL | None |
| | MGSTAT1 | Set if any errors have been encountered during the verify operation |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the verify operation |

Note:
238. If a Program Once phrase is initially programmed to 0xFFFF_FFFF_FFFF_FFFF, the Program Once command is allowed to execute again on the same phrase.

#### 5.40.4.5.7 Erase All Blocks Command

The Erase All Blocks operation erases the entire P-Flash and D-Flash memory space.

**Table 479. Erase All Blocks Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x08 | Not required |

Upon clearing CCIF to launch the Erase All Blocks command, the Memory Controller erases the entire Flash memory space and verify it is erased. If the Memory Controller verifies the entire Flash memory space was properly erased, security is released. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag is set after the Erase All Blocks operation has completed.

**Table 480. Erase All Blocks Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 000 at command launch |
| | | Set if command not available in current mode (see Table 463) |
| | FPVIOL | Set if any area of the P-Flash or D-Flash memory is protected |
| | MGSTAT1 | Set if any errors have been encountered during the verify operation |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the verify operation |

#### 5.40.4.5.8 Erase Flash Block Command

The Erase Flash Block operation erases all addresses in a P-Flash or D-Flash block.

**Table 481. Erase Flash Block Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x09 | Global address [17:16] to identify Flash block |
| 001 | Global address [15:0] in Flash block to be erased | |

Upon clearing CCIF to launch the Erase Flash Block command, the Memory Controller erases the selected Flash block and verify it is erased. The CCIF flag sets after the Erase Flash Block operation has completed.

**Table 482. Erase Flash Block 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 463) |
| | | Set if an invalid global address [17:16] is supplied |
| | | Set if the supplied P-Flash address is not phrase-aligned or if the D-Flash address is not word-aligned |
| | FPVIOL | Set if an area of the selected Flash block is protected |
| | MGSTAT1 | Set if any errors have been encountered during the verify operation |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the verify operation |

#### 5.40.4.5.9 Erase P-Flash Sector Command

The Erase P-Flash Sector operation erases all addresses in a P-Flash sector.

**Table 483. Erase P-Flash Sector Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x0A | Global address [17:16] to identify P-Flash block to be erased |
| 001 | Global address [15:0] anywhere within the sector to be erased. Refer to Section 5.40.1.2.1, “P-Flash Features" for the P-Flash sector size. | |

Upon clearing CCIF to launch the Erase P-Flash Sector command, the Memory Controller erases the selected Flash sector and then verify it is erased. The CCIF flag sets after the Erase P-Flash Sector operation has completed.

**Table 484. Erase P-Flash Sector 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 463) |
| | | Set if an invalid global address [17:16] is supplied |
| | | Set if a misaligned phrase address is supplied (global address [2:0] != 000) |
| | FPVIOL | Set if the selected P-Flash sector is protected |
| | MGSTAT1 | Set if any errors have been encountered during the verify operation |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the verify operation |

#### 5.40.4.5.10 Unsecure Flash Command

The Unsecure Flash command erases the entire P-Flash and D-Flash memory space and, if the erase is successful, releases security.

**Table 485. Unsecure Flash Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x0B | Not required |

Upon clearing CCIF to launch the Unsecure Flash command, the Memory Controller erases the entire P-Flash and D-Flash memory space and verify it is erased. If the Memory Controller verifies the entire Flash memory space was properly erased, security is released. If the erase verify is not successful, the Unsecure Flash operation sets MGSTAT1 and terminates without changing the security state. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag is set after the Unsecure Flash operation has completed.

**Table 486. Unsecure Flash Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 000 at command launch |
| | | Set if command not available in current mode (see Table 463) |
| | FPVIOL | Set if any area of the P-Flash or D-Flash memory is protected |
| | MGSTAT1 | Set if any errors have been encountered during the verify operation |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the verify operation |

#### 5.40.4.5.11 Verify Back Door Access Key Command

The Verify Back Door Access Key command only executes if it is enabled by the KEYEN bits in the FSEC register (see Table 427). The Verify Back Door Access Key command releases security if user-supplied keys match those stored in the Flash security bytes of the Flash configuration field (see Table 418). The Verify Back Door Access Key command must not be executed from the Flash block containing the back door comparison key to avoid code runaway.

**Table 487. Verify Back Door Access Key Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x0C | Not required |
| 001 | Key 0 | |
| 010 | Key 1 | |
| 011 | Key 2 | |
| 100 | Key 3 | |

Upon clearing CCIF to launch the Verify Back Door Access Key command, the Memory Controller checks the FSEC KEYEN bits to verify this command is enabled. If not enabled, the Memory Controller sets the ACCERR bit in the FSTAT register and terminates. If the command is enabled, the Memory Controller compares the key provided in FCCOB to the buckaroo comparison key in the Flash configuration field with Key 0 compared to 0x3_FF00, etc. If the back door keys match, security is released. If the back door keys do not match, security is not released and all future attempts to execute the Verify Back Door Access Key command are aborted (set ACCERR) until a reset occurs. The CCIF flag is set after the Verify Back Door Access Key operation has completed.

**Table 488. Verify Back Door Access Key Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 100 at command launch |
| | | Set if an incorrect back door key is supplied |
| | | Set if back door key access has not been enabled (KEYEN[1:0] != 10, see Section 5.40.3.2.2, “Flash Security Register (FSEC)”) |
| | | Set if the back door key has mismatched since the last reset |
| | FPVIOL | None |
| | MGSTAT1 | None |
| | MGSTAT0 | None |

#### 5.40.4.5.12 Set User Margin Level Command

The Set User Margin Level command causes the Memory Controller to set the margin level for future read operations of the P-Flash or D-Flash block.

**Table 489. Set User Margin Level Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x0D | Global address [17:16] to identify the Flash block |
| 001 | Margin level setting | |

Upon clearing CCIF to launch the Set User Margin Level command, the Memory Controller sets the user margin level for the targeted block and then set the CCIF flag.

**NOTE**

When the D-Flash block is targeted, the D-Flash user margin levels are applied only to the D-Flash reads. However, when the P-Flash block is targeted, the P-Flash user margin levels are applied to both P-Flash and D-Flash reads. It is not possible to apply user margin levels to the P-Flash block only.

Valid margin level settings for the Set User Margin Level command are defined in Table 490.

**Table 490. Valid Set User Margin Level Settings**

| CCOB (CCOBIX=001) | Level Description |
|---|---|
| 0x0000 | Return to Normal Level |
| 0x0001 | User Margin-1 Level(239) |
| 0x0002 | User Margin-0 Level(240) |

Note:
239. Read margin to the erased state
240. Read margin to the programmed state

**Table 491. Set User 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 463) |
| | | Set if an invalid global address [17:16] is supplied |
| | | Set if an invalid margin level setting is supplied |
| | FPVIOL | None |
| | MGSTAT1 | None |
| | MGSTAT0 | None |

**NOTE**

User margin levels can be used to check the Flash memory contents have adequate margin for normal level read operations. If unexpected results are encountered when checking Flash memory contents at user margin levels, a potential loss of information has been detected.

#### 5.40.4.5.13 Set Field Margin Level Command

The Set Field Margin Level command, valid in special modes only, causes the Memory Controller to set the margin level specified for future read operations of the P-Flash or D-Flash block.

**Table 492. Set Field Margin Level Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x0E | Global address [17:16] to identify the Flash block |
| 001 | Margin level setting | |

Upon clearing CCIF to launch the Set Field Margin Level command, the Memory Controller sets the field margin level for the targeted block and then set the CCIF flag.

**NOTE**

When the D-Flash block is targeted, the D-Flash field margin levels are applied only to the D-Flash reads. However, when the P-Flash block is targeted, the P-Flash field margin levels are applied to both P-Flash and D-Flash reads. It is not possible to apply field margin levels to the P-Flash block only.

Valid margin level settings for the Set Field Margin Level command are defined in Table 493.

**Table 493. Valid Set Field Margin Level Settings**

| CCOB (CCOBIX=001) | Level Description |
|---|---|
| 0x0000 | Return to Normal Level |
| 0x0001 | User Margin-1 Level(241) |
| 0x0002 | User Margin-0 Level(242) |
| 0x0003 | Field Margin-1 Level(241) |
| 0x0004 | Field Margin-0 Level(242) |

Note:
241. Read margin to the erased state
242. Read margin to the programmed state

**Table 494. 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 463) |
| | | Set if an invalid global address [17:16] 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 the 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.

#### 5.40.4.5.14 Erase Verify D-Flash Section Command

The Erase Verify D-Flash Section command verifies 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 495. Erase Verify D-Flash Section Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x10 | Global address [17:16] to identify the D-Flash block |
| 001 | Global address [15:0] of the first word to be verified | |
| 010 | Number of words to be verified | |

Upon clearing CCIF to launch the Erase Verify D-Flash Section command, the Memory Controller verifies the selected section of D-Flash memory is erased. The CCIF flag sets after the Erase Verify D-Flash Section operation has completed.

**Table 496. Erase Verify D-Flash Section Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 010 at command launch |
| | | Set if command not available in current mode (see Table 463) |
| | | Set if an invalid global address [17:0] is supplied |
| | | Set if a misaligned word address is supplied (global address [0] != 0) |
| | | Set if the requested section breaches the end of the D-Flash block |
| | FPVIOL | None |
| | MGSTAT1 | Set if any errors have been encountered during the read |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the read |

#### 5.40.4.5.15 Program D-Flash Command

The Program D-Flash operation programs one to four previously erased words in the D-Flash block. The Program D-Flash operation confirms the targeted location(s) were successfully programmed upon completion.

**CAUTION**

A Flash word must be in the erased state before being programmed. Cumulative programming of bits within a Flash word is not allowed.

**Table 497. Program D-Flash Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x11 | Global address [17:16] to identify the D-Flash block |
| 001 | Global address [15:0] of word to be programmed | |
| 010 | Word 0 program value | |
| 011 | Word 1 program value, if desired | |
| 100 | Word 2 program value, if desired | |
| 101 | Word 3 program value, if desired | |

Upon clearing CCIF to launch the Program D-Flash command, the user-supplied words are transferred to the Memory Controller and be programmed if the area is unprotected. The CCOBIX index value at Program D-Flash command launch determines how many words are programmed in the D-Flash block. The CCIF flag is set when the operation has completed.

**Table 498. Program D-Flash Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] < 010 at command launch |
| | | Set if CCOBIX[2:0] > 101 at command launch |
| | | Set if command not available in current mode (see Table 463) |
| | | Set if an invalid global address [17:0] is supplied |
| | | Set if a misaligned word address is supplied (global address [0] != 0) |
| | | Set if the requested group of words breaches the end of the D-Flash block |
| | FPVIOL | Set if the selected area of the D-Flash memory is protected |
| | MGSTAT1 | Set if any errors have been encountered during the verify operation |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the verify operation |

#### 5.40.4.5.16 Erase D-Flash Sector Command

The Erase D-Flash Sector operation erases all addresses in a sector of the D-Flash block.

**Table 499. Erase D-Flash Sector Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x12 | Global address [17:16] to identify D-Flash block |
| 001 | Global address [15:0] anywhere within the sector to be erased. See Section 5.40.1.2.2, “D-Flash Features" for D-Flash sector size. | |

Upon clearing CCIF to launch the Erase D-Flash Sector command, the Memory Controller erases the selected Flash sector and verify it is erased. The CCIF flag sets after the Erase D-Flash Sector operation has completed.

**Table 500. Erase D-Flash Sector 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 463) |
| | | Set if an invalid global address [17:0] is supplied |
| | | Set if a misaligned word address is supplied (global address [0] != 0) |
| | FPVIOL | Set if the selected area of the D-Flash memory is protected |
| | MGSTAT1 | Set if any errors have been encountered during the verify operation |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the verify operation |

### 5.40.4.6 Interrupts

The Flash module can generate an interrupt when a Flash command operation has completed or when a Flash command operation has detected an ECC fault.

**Table 501. Flash Interrupt Sources**

| Interrupt Source | Interrupt Flag | Local Enable | Global (CCR) Mask |
|---|---|---|---|
| Flash Command Complete | CCIF (FSTAT register) | CCIE (FCNFG register) | I Bit |
| ECC Double Bit Fault on Flash Read | DFDIF (FERSTAT register) | DFDIE (FERCNFG register) | I Bit |
| ECC Single Bit Fault on Flash Read | SFDIF (FERSTAT register) | SFDIE FERCNFG register) | I Bit |

**NOTE**

Vector addresses and their relative interrupt priority are determined at the MCU level.

#### 5.40.4.6.1 Description of Flash Interrupt Operation

The Flash module uses the CCIF flag in combination with the CCIE interrupt enable bit to generate the Flash command interrupt request. The Flash module uses the DFDIF and SFDIF flags in combination with the DFDIE and SFDIE interrupt enable bits to generate the Flash error interrupt request. For a detailed description of the register bits involved, refer to Section 5.40.3.2.5, “Flash Configuration Register (FCNFG)", Section 5.40.3.2.6, “Flash Error Configuration Register (FERCNFG)", Section 5.40.3.2.7, “Flash Status Register (FSTAT)", and Section 5.40.3.2.8, “Flash Error Status Register (FERSTAT)".

The logic used for generating the Flash module interrupts is shown in Figure 110.

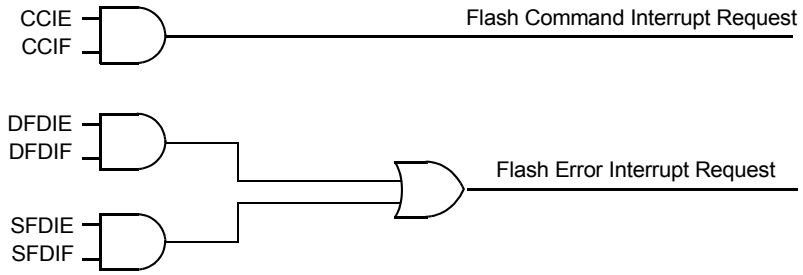



**Figure 110. Flash Module Interrupts Implementation**

### 5.40.4.7 Stop Mode

If a Flash command is active (CCIF = 0) when the MCU requests stop mode, the current Flash operation is completed before the CPU is allowed to enter stop mode.

### 5.40.5 Security

The Flash module provides security information to the MCU. The Flash security state is defined by the SEC bits of the FSEC register (see Table 428). During reset, the Flash module initializes the FSEC register using data read from the security byte of the Flash configuration field at global address 0x3_FF0F. The security state out of reset can be permanently changed by programming the security byte assuming the MCU is starting from a mode where the necessary P-Flash erase and program commands are available and the upper region of the P-Flash is unprotected. If the Flash security byte is successfully programmed, its new value takes affect after the next MCU reset.

The following subsections describe these security-related subjects:

- Unsecuring the MCU using Back door Key Access
- Unsecuring the MCU in Special Single Chip Mode using BDM
- Mode and Security Effects on Flash Command Availability

### 5.40.5.1 Unsecuring the MCU using Back door Key Access

The MCU may be unsecured by using the back door key access feature which requires knowledge of the contents of the back door keys (four 16-bit words programmed at addresses 0x3_FF00-0x3_FF07). If the KEYEN[1:0] bits are in the enabled state (see Section 5.40.3.2.2, “Flash Security Register (FSEC)"), the Verify Back Door Access Key command (see Section 5.40.4.5.11, “Verify Back Door Access Key Command") 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 Back Door Access Key command match the back door keys stored in the Flash memory, the SEC bits in the FSEC register (see Table 428) is changed to unsecure the MCU. Key values of 0x0000 and 0xFFFF are not permitted as back door keys. While the Verify Back Door Access Key command is active, P-Flash memory and D-Flash memory is not available for read access and returns invalid data.

The user code stored in the P-Flash memory must have a method of receiving the back door 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 5.40.3.2.2, “Flash Security Register (FSEC)"), the MCU can be unsecured by the back door key access sequence described below:

1. Follow the command sequence for the Verify Back Door Access Key command as explained in Section 5.40.4.5.11, “Verify Back Door Access Key Command"
2. If the Verify Back Door 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 Back Door Access Key command is monitored by the Memory Controller and an illegal key prohibits future use of the Verify Back Door Access Key command. A reset of the MCU is the only method to re-enable the Verify Back Door Access Key command. The security as defined in the Flash security byte (0x3_FF0F) is not changed by using the Verify Back Door Access Key command sequence. The back door keys stored in addresses 0x3_FF00-0x3_FF07 are unaffected by the Verify Back Door Access Key command sequence. The Verify Back Door Access Key command sequence has no effect on the program and erase protections defined in the Flash protection register, FPROT.

After the back door keys have been correctly matched, the MCU is 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 back door keys by programming addresses 0x3_FF00-0x3_FF07 in the Flash configuration field.

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

A secured MCU can be unsecured in special single chip mode by using the following method to erase the P-Flash and D-Flash memory:

1. Reset the MCU into special single chip mode
2. Delay while the BDM executes the Erase Verify All Blocks command write sequence to check if the P-Flash and D-Flash memories are erased
3. Send BDM commands to disable protection in the P-Flash and D-Flash memory
4. Execute the Erase All Blocks command write sequence to erase the P-Flash and D-Flash memory
5. After the CCIF flag sets to indicate the Erase All Blocks operation has completed, reset the MCU into special single chip mode
6. Delay while the BDM executes the Erase Verify All Blocks command write sequence to verify the P-Flash and D-Flash memory are erased

If the P-Flash and D-Flash memory are verified as erased, the MCU is unsecured. All BDM commands is now enabled and the Flash security byte may be programmed to the unsecure state by continuing with the following steps:

7. Send BDM commands to execute the Program P-Flash command write sequence to program the Flash security byte to the unsecured state
8. Reset the MCU

### 5.40.5.3 Mode and Security Effects on Flash Command Availability

The availability of Flash module commands depends on the MCU operating mode and security state as shown in Table 463.

## 5.40.6 Initialization

On each system reset the Flash module executes a reset sequence which establishes initial values for the Flash Block Configuration Parameters, the FPROT and DFPROT protection registers, and the FOPT and FSEC registers. The Flash module reverts to using built-in default values which leave the module in a fully protected and secured state if errors are encountered during execution of the reset sequence. If a double bit fault is detected during the reset sequence, both MGSTAT bits in the FSTAT register is set.

CCIF remains clear throughout the reset sequence. The Flash module holds off all CPU access for the initial portion of the reset sequence. While Flash memory reads and access to most Flash registers are possible when the hold is removed, writes to the FCCOBIX, FCCOBHI, and FCCOBLO registers are ignored. Completion of the reset sequence is marked by setting CCIF high which enables writes to the FCCOBIX, FCCOBHI, and FCCOBLO registers to launch any available Flash command.

If a reset occurs while any Flash command is in progress, the command is immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed.

## 5.41 Die-to-Die Initiator (D2DIV1)

### 5.41.1 Preface

This document contains the user specification of the D2D Initiator.

#### 5.41.1.1 Acronyms and Abbreviations

Table 502 contains sample acronyms and abbreviations used in this document.

**Table 502. Acronyms and Abbreviated Terms**

| Term | Meaning |
|---|---|
| D2D | Die-to-Die |

##### 5.41.1.1.1 Glossary

Table 253 shows a glossary of the major terms used in this document.

**Table 503. Glossary**

| Term | Definition |
|---|---|
| Active low | The signal is asserted when it changes to logic-level zero. |
| Active high | The signal is asserted when it changes to logic-level one. |
| Asserted | Discrete signal is in active logic state. |
| Customer | The end user of an SoC design or device. |
| EOT | End of Transaction |
| Negated | A discrete signal is in inactive logic state. |
| Pin | External physical connection. |
| Revision | Revised or new version of a document. Revisions produce versions; there can be no ‘Rev 0.0.’ |
| Signal | Electronic construct whose state or change in state conveys information. |
| Transfer | A read or write on the CPU bus following the IP-Bus protocol. |
| Transaction | Command, address and if required data sent on the D2D interface. A transaction is finished by the EOT acknowledge cycle. |
| Version | Particular form or variation of an earlier or original document. |

### 5.41.2 Introduction

This section describes the functionality of the die-to-die (D2DIV1) initiator block especially designed for low cost connections between a microcontroller die (Interface Initiator) and an analog die (Interface Target) located in the same package.

The D2DI block

- realizes the initiator part of the D2D interface, including supervision and error interrupt generation
- generates the clock for this interface
- disables/enables the interrupt from the D2D interface

### 5.41.2.1 Overview

The D2DI is the initiator for a data transfer to and from a target typically located on another die in the same package. It provides a set of configuration registers and two memory mapped 256 Byte address windows. When writing to a window a transaction is initiated sending a write command, followed by an 8-bit address and the data byte or word to the target. When reading from a window a transaction is initiated sending a read command, followed by an 8-bit address to the target. The target then responds with the data. The basic idea is a peripheral located on another die, can be addressed like an on-chip peripheral, except for a small transaction delay.

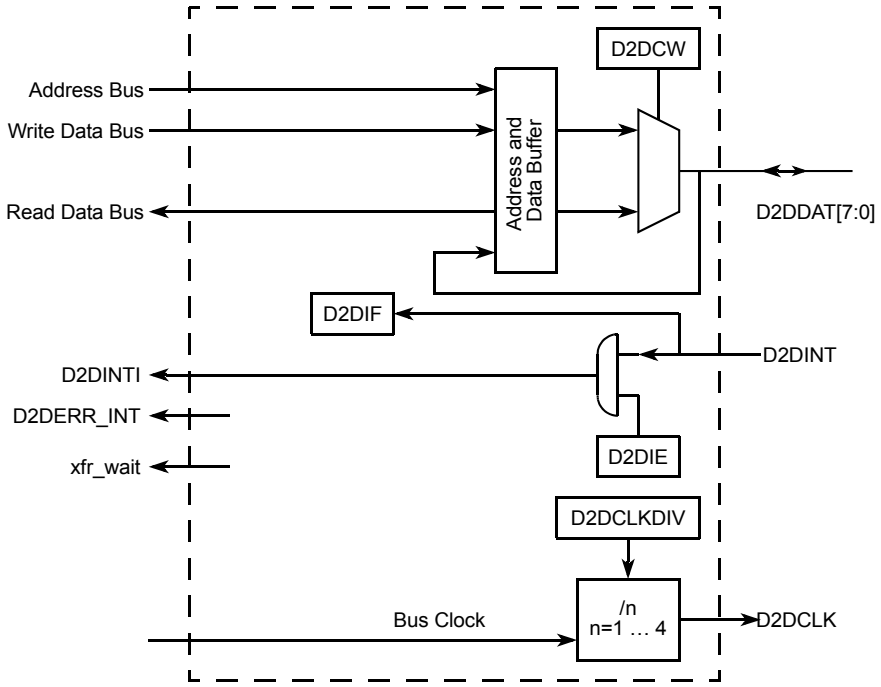



**Figure 111. Die-to-Die Initiator (D2DI) Block Diagram**

### 5.41.2.2 Features

The main features of this block are

- Software transparent, memory mapped access to peripherals on target die
  - 256 Byte address window
  - Supports blocking read or write as well as non-blocking write transactions
- Scalable interface clock divide by 1, 2, 3, or 4 of bus clock
- Clock halt on system STOP
- Configurable for 4 or 8-bit wide transfers
- Configurable timeout period
- Non-maskable interrupt on transaction errors
- Transaction Status and Error Flags
- Interrupt enable for receiving interrupt (from D2D target)

### 5.41.2.3 Modes of Operation

#### 5.41.2.3.1 D2DI in STOP mode

The D2DI stops working in STOP mode. The D2DCLK signal as well as the data signals used are driven low (only after the end of the current high phase, as defined by D2DCLKDIV).

Waking from STOP mode, the D2DCLK line starts clocking again and the data lines is driven low until the first transaction starts.

STOP mode is entered by different CPU instructions. Every (enabled) interrupt can be used to leave the STOP mode.

#### 5.41.2.3.2 D2DI in special modes

The MCU can enter a special mode (used for test and debugging purposes as well as programming the FLASH). In the D2DI the “write-once” feature is disabled. See the MCU description for details.

### 5.41.3 External Signal Description

The D2DI optionally uses 6 or 10 port pins. The functions of those pins depends on the settings in the D2DCTL0 register, when the D2DI module is enabled.

#### 5.41.3.1 D2DCLK

When the D2DI is enabled this pin is the clock output. This signal is low if the initiator is disabled, in STOP mode (with D2DSWAI asserted), otherwise it is a continuos clock. This pin may be shared with general purpose functionality if the D2DI is disabled.

#### 5.41.3.2 D2DDAT[7:4]

When the D2DI is enabled and the interface connection width D2DCW is set to be 8-bit wide, those lines carry the data bits 7:4 acting as outputs or inputs. When they act as inputs pull-down elements are enabled. If the D2DI is disabled or if the interface connection width is set as 4-bit wide, the pins may be shared with general purpose pin functionality. In MM912_634 configuration, D2DDAT [7:4] are not bonded, and act as GPIO inputs. These pads are in high-impedance mode. To avoid any leakage in Stop mode, D2DDAT [7:4] should be configured as an output by writing 0xF0 in register DDRD (0x0007).

#### 5.41.3.3 D2DDAT[3:0]

When the D2DI is enabled those lines carry the data bits 3:0 acting as outputs or inputs. When they act as inputs pull-down elements are enabled. If the D2DI is disabled the pins and may be shared with general purpose pin functionality.

#### 5.41.3.4 D2DINT

The D2DINT is an active input interrupt input driven by the target device. The pin has an active pull-down device. If the D2DI is disabled the pin may be shared with general purpose pin functionality.

**Table 504. Signal Properties**

| Name | Primary (D2DEN=1) | I/O | Secondary (D2DEN=0) | Reset | Comment | Pull down |
|---|---|---|---|---|---|---|
| D2DDAT[7:0] | Bi-directional Data Lines | I/O | GPIO | 0 | driven low if in STOP mode | Active(243) |
| D2DCLK | Interface Clock Signal | O | GPIO | 0 | low if in STOP mode | — |
| $\overline{\text{D2DINT}}$ | Active High Interrupt | I | GPIO | — | — | Active(244) |

Note:
243. Active if in input state, only if D2DEN=1
244. only if D2DEN=1

See the port interface module (PIM) guide for details of the GPIO function.

### 5.41.4 Memory Map and Register Definition

#### 5.41.4.1 Memory Map

The D2DI memory map is split into three sections.

1. An eight-byte set of control registers
2. A 256 byte window for blocking transactions
3. A 256 byte window for non-blocking transactions

See the chapter “Device Memory Map” for the register layout (distribution of these sections).

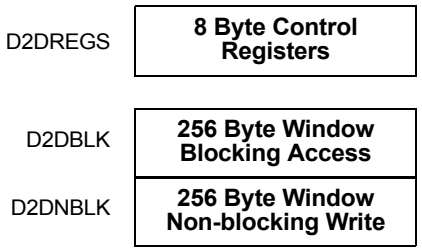



**Figure 112. D2DI Top Level Memory Map**

A summary of the registers associated with the D2DI block is shown in Table 255. Detailed descriptions of the registers and bits are given in the subsections which follow.

**Table 505. D2DI Register Summary**

| Offset | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0 | D2DCTL0 | R | D2DEN | D2DCW | D2DSWAI | 0 | 0 | 0 | D2DCLKDIV[1:0] | |
| | | W | | | | | | | | |
| 0x1 | D2DCTL1 | | D2DIE | 0 | 0 | 0 | TIMEOUT[3:0] | | | |
| | | | | | | | | | | |
| 0x2 | D2DSTAT0 | R | ERRIF | ACKERF | CNCLF | TIMEF | TERRF | PARF | PAR1 | PAR0 |
| | | W | | | | | | | | |
| 0x3 | D2DSTAT1 | | D2DIF | D2DBSY | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | | |
| 0x4 | D2DADRHI | R | RWB | SZ8 | 0 | NBLK | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x5 | D2DADRLO | R | ADR[7:0] | | | | | | | |
| | | W | | | | | | | | |
| 0x6 | D2DDATAHI | R | DATA[15:8] | | | | | | | |
| | | W | | | | | | | | |
| 0x7 | D2DDATALO | R | DATA[7:0] | | | | | | | |
| | | W | | | | | | | | |
| | | | = Unimplemented or Reserved | | | | | | | |

### 5.41.4.2 Register Definition

### 5.41.4.3 D2DI Control Register 0 (D2DCTL0)

This register is used to enable and configure the interface width, the wait behavior and the frequency of the interface clock.

**Table 506. D2DI Control Register 0 (D2DCTL0)**

Offset 0x0 Access: User read/write

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|

**Table 506. D2DI Control Register 0 (D2DCTL0)**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| R | D2DEN | D2DCW | D2DSWAI | 0 | 0 | 0 | D2DCLKDIV[1:0] | |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 507. D2DCTL0 Register Field Descriptions**

| Field | Description |
|---|---|
| 7<br>D2DEN | **D2DI Enable** — Enables the D2DI module. This bit is write-once in normal mode and can always be written in special modes.<br>0 D2DI initiator is disabled. No lines are not used, the pins have their GPIO (secondary) function.<br>1 D2DI initiator is enabled. After setting D2DEN=1 the D2DDAT[7:0] (or [3:0], see D2DCW) lines are driven low with the IDLE command; the D2DCLK is driven by the divided bus clock. |
| 6<br>D2DCW | **D2D Connection Width** — Sets the number of data lines used by the interface. This bit is write-once in normal modes and can always be written in special modes.<br>0 Lines D2DDAT[3:0] are used for four line data transfer. D2DDAT[7:4] are unused.<br>1 All eight interface lines D2DDAT[7:0] are used for data transfer. (245) |
| 5<br>- | **Reserved** — For internal use |
| 4:2 | Reserved, should be written to 0 to ensure compatibility with future versions of this interface. |
| 1:0<br>D2DCLKDIV | **Interface Clock Divider** — Determines the frequency of the interface clock. These bits are write-once in normal modes and can be always written in special modes. See Figure 113 for details on the clock waveforms<br>00 Encoding 0. Bus clock divide by 1.<br>01 Encoding 1. Bus clock divide by 2.<br>10 Encoding 2. Bus clock divide by 3.<br>11 Encoding 3. Bus clock divide by 4. |

Note:
245. 8-lines configuration cannot be used in the MM912_634 configuration.

The Clock Divider provides the waveforms as shown in Figure 113. The duty cycle of the clock is not always 50%, the high cycle is shorter than 50% or equal but never longer, since this is beneficial for the transaction speed.

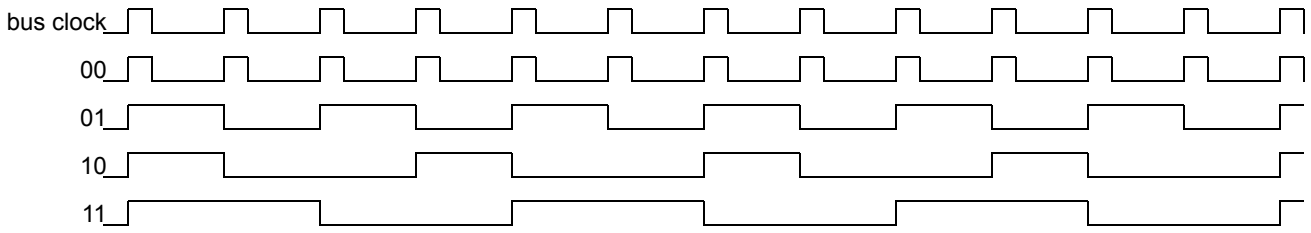



**Figure 113. Interface Clock Waveforms for various D2DCLKDIV Encoding**

### 5.41.4.4 D2DI Control Register 1 (D2DCTL1)

This register is used to enable the D2DI interrupt and set number of D2DCLK cycles before a timeout error is asserted.

**Table 508. D2DI Control Register 1 (D2DCTL1)**

Offset 0x1 Access: User read/write

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|

**Table 508. D2DI Control Register 1 (D2DCTL1)**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | D2DIE | 0 | 0 | 0 | TIMOUT[3:0] | | | |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 509. D2DCTL1 Register Field Descriptions**

| Field | Description |
|---|---|
| 7<br>D2DIE | **D2D Interrupt Enable** — Enables the external interrupt<br>0 External Interrupt is disabled<br>1 External Interrupt is enabled |
| 6:4 | Reserved, should be written to 0 to ensure compatibility with future versions of this interface. |
| 3:0<br>TIMOUT | **Timeout Setting** — Defines the number of D2DCLK cycles to wait after the last transaction cycle until a timeout is asserted. In case of a timeout the TIMEF flag in the D2DSTAT0 register is set.<br>These bits are write-once in normal modes and can always be written in special modes.<br>0000: The acknowledge is expected directly after the last transfer, i.e. the target must not insert a wait cycle.<br>0001 - 1111: The target may insert up to TIMOUT wait states before acknowledging a transaction until a timeout is asserted<br>The selected value of TIMOUT depends on the application. Determining factors include the frequency of blocking accesses or consecutive accesses on the D2D interface. |

**NOTE**

"Write-once" means after writing D2DCNTL0.D2DEN=1 the write accesses to these bits have no effect.

### 5.41.4.5 D2DI Status Register 0 (D2DSTAT0)

This register reflects the status of the D2DI transactions.

**Table 510. D2DI Status Register 0 (D2DSTAT0)**

Offset 0x2 Access: User read/write

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | ERRIF | ACKERF | CNCLF | TIMEF | TERRF | PARF | PAR1 | PAR0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 511. D2DI Status Register 0 Field Descriptions**

| Field | Description |
|---|---|
| 7<br>ERRIF | **D2DI error interrupt flag** — This status bit indicates the D2D initiator has detected an error condition (summary of the following five flags).This interrupt is not locally maskable. Write a 1 to clear the flag. Writing a 0 has no effect.<br>0 D2DI has not detected an error during a transaction.<br>1 D2DI has detected an error during a transaction. |
| 6<br>ACKERF | **Acknowledge Error Flag**— This read-only flag indicates in the acknowledge cycle not all data inputs are sampled high, indicating a potential broken wire. This flag is cleared when the ERRIF bit is cleared by writing a 1 to the ERRIF bit. |
| 5<br>CNCLF | **CNCLF** — This read-only flag indicates the initiator has canceled a transaction and replaced it by an IDLE command due to a pending error flag (ERRIF). This flag is cleared when the ERRIF bit is cleared by writing a 1 to the ERRIF bit. |
| 4<br>TIMEF | **Time Out Error Flag** — This read-only flag indicates the initiator has detected a timeout error. This flag is cleared when the ERRIF bit is cleared by writing a 1 to the ERRIF bit. |
| 3<br>TERRF | **Transaction Error Flag** — This read-only flag indicates the initiator has detected the error signal during the acknowledge cycle of the transaction. This flag is cleared when the ERRIF bit is cleared by writing a 1 to the ERRIF bit. |

**Table 511. D2DI Status Register 0 Field Descriptions (continued)**

| Field | Description |
|---|---|
| 2<br>PARF | **Parity Error Flag** — This read-only flag indicates the initiator has detected a parity error. Parity bits[1:0] contain further information. This flag is cleared when the ERRIF bit is cleared by writing a 1 to the ERRIF bit. |
| 1<br>PAR1 | **Parity Bit** — P[1] as received by the D2DI |
| 0<br>PAR0 | **Parity Bit** — P[0] as received by the D2DI |

### 5.41.4.6 D2DI Status Register 1 (D2DSTAT1)

This register holds the status of the external interrupt pin and an indicator about the D2DI transaction status.

**Table 512. D2DI Status Register 1 (D2DSTAT1)**

Offset 0x3 — Access: User read

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | D2DIF | D2DBSY | 0 | 0 | 0 | 0 | 0 | 0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 513. D2DSTAT1 Register Field Descriptions**

| Field | Description |
|---|---|
| 7<br>D2DIF | **D2D Interrupt Flag** — This read-only flag reflects the status of the D2DINT Pin. The D2D interrupt flag can only be cleared by a target specific interrupt acknowledge sequence.<br>0 External Interrupt is negated<br>1 External Interrupt is asserted |
| 6<br>D2DBSY | **D2D Initiator Busy** — This read-only status bit indicates a D2D transaction is ongoing.<br>0 D2D initiator idle.<br>1 D2D initiator transaction ongoing. |
| 5:0 | Reserved, should be masked to ensure compatibility with future versions of this interface. |

### 5.41.4.7 D2DI Address Buffer Register (D2DADR)

This read-only register contains information about the ongoing D2D interface transaction. The register content is updated when a new transaction starts. In error cases the user can track back, which transaction failed.

**Table 514. D2DI Address Buffer Register (D2DADR)**

Offset 0x4/0x5 — Access: User read

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | RWB | SZ8 | 0 | NBLK | 0 | 0 | 0 | 0 | ADR[7:0] | | | | | | | |
| W | | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 515. D2DI Address Buffer Register Bit Descriptions**

| Field | Description |
|---|---|
| 15<br>RWB | **Transaction Read-Write Direction** — This read-only bit reflects the direction of the transaction<br>0 Write Transaction<br>1 Read Transaction |
| 14<br>SZ8 | **Transaction Size** — This read-only bit reflects the data size of the transaction<br>0 16-bit transaction.<br>1 8-bit transaction. |
| 13 | Reserved, should be masked to ensure compatibility with future versions of this interface. |
| 12<br>NBLK | **Transaction Mode** — This read-only bit reflects the mode of the transaction<br>0 Blocking transaction.<br>1 Non-blocking transaction. |
| 11:8 | Reserved, should be masked to ensure compatibility with future versions of this interface. |
| 7:0<br>ADR[7:0] | **Transaction Address** — Those read-only bits contain the address of the transaction |

### 5.41.4.8 D2DI Data Buffer Register (D2DDATA)

This read-only register contains information about the ongoing D2D interface transaction. For a write transaction the data becomes valid at the begin of the transaction. For a read transaction the data is updated during the transaction and is finalized when the transaction is acknowledged by the target. In error cases the user can track back what has happened.

**Table 516. D2DI Data Buffer Register (D2DDATA)**

Offset 0x6/0x7 Access: User read

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R | DATA15:0 | | | | | | | | | | | | | | | |
| W | | | | | | | | | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 517. D2DI Data Buffer Register Bit Descriptions**

| Field | Description |
|---|---|
| 15:0<br>DATA | **Transaction Data** — Those read-only bits contain the data of the transaction |

Both D2DDATA and D2DADR can be read with byte accesses.

## 5.41.5 Functional Description

### 5.41.5.1 Initialization

Out of reset the interface is disabled. The interface must be initialized by setting the interface clock speed, the time-out value, the transfer width and finally enabling the interface. This should be done using a 16-bit write or if using 8-bit write D2DCTL1 must be written before D2D2CTL0.D2DEN=1 is written. Once it is enabled in normal modes, only a reset can disable it again (write-once feature).

### 5.41.5.2 Transactions

A transaction on the D2D Interface is triggered by writing to either the 256 byte address window or reading from the address window (see STAA/LDAA 0/1 in the next figure). Depending on which address window is used a blocking or a non-blocking transaction is performed. The address for the transaction is the 8-bit wide window relative address. The data width of the CPU read or write instructions determines if 8-bit or 16-bit wide data are transferred. There is always only one transaction active. Figure 114 shows the various types of transactions explained in more detail below.

For all 16-bit read/write accesses of the CPU the addresses are assigned according the big-endian model:

word [15:8]: addr word[7:0]: addr+1

addr: byte-address (8 bit wide) inside the blocking or non-blocking window, as provided by the CPU and transferred to the D2D target word: CPU data, to be transferred from/to the D2D target The application must care for the stretched CPU cycles (limited by the TIMOUT value, caused by blocking or consecutive accesses), which could affect time limits, including COP (computer operates properly) supervision. The stretched CPU cycles cause the “CPU halted” phases (see Figure 114).

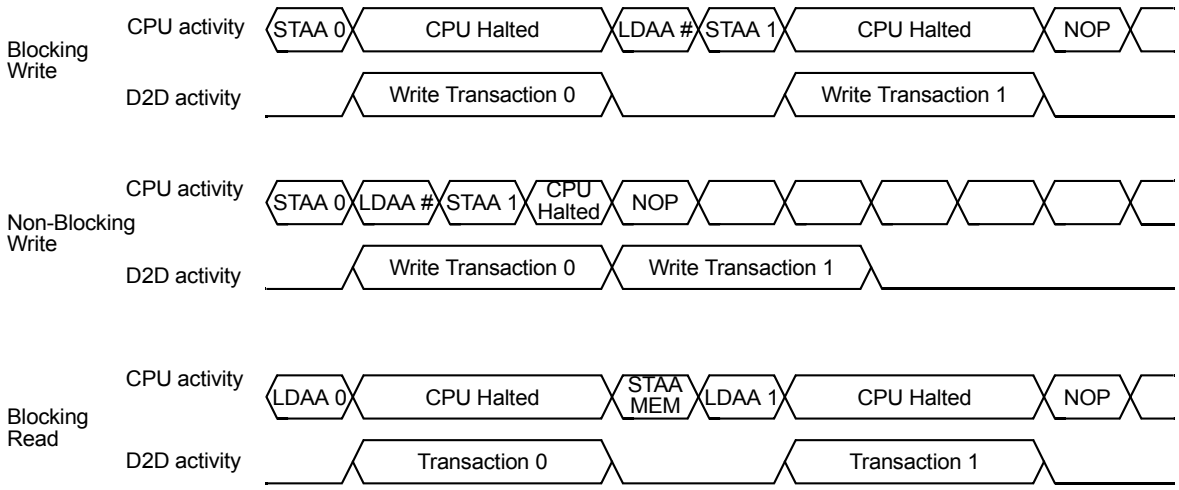



**Figure 114. Blocking and Non-Blocking Transfers.**

#### 5.41.5.2.1 Blocking Writes

When writing to the address window associated with blocking transactions, the CPU is held until the transaction is completed, before completing the instruction. Figure 114 shows the behavior of the CPU for a blocking write transaction shown in the following example.

```
STAA    BLK_WINDOW+OFFS0 ; WRITE0 8-bit as a blocking transaction
LDAA    #BYTE1
STAA    BLK_WINDOW+OFFS1 ; WRITE1 is executed after WRITE0 transaction is completed
NOP
```

Blocking writes should be used when clearing interrupt flags located in the target or other writes which require the operation at the target is completed before proceeding with the CPU instruction stream.

### 5.41.5.3 Non-Blocking Writes

When writing to the address window associated with non-blocking transactions, the CPU can continue before the transaction is completed. However if there was a transaction ongoing when doing the 2nd write the CPU is held until the first one is completed, before executing the 2nd one. Figure 114 shows the behavior of the CPU for a blocking write transaction shown in the following example.

```
STAA    NONBLK_WINDOW+OFFS0; write 8-bit as a non blocking transaction
LDAA    #BYTE1              ; load next byte
STAA    NONBLK_WINDOW+OFFS1; executed right after the first
NOP
```

As the figure illustrates non-blocking writes have a performance advantage, but care must be taken the following instructions are not affected by the change in the target caused by the previous transaction.

### 5.41.5.4 Blocking Read

When reading from the address window associated with blocking transactions, the CPU is held until the data is returned from the target, before completing the instruction.Figure 114 shows the behavior of the CPU for a blocking read transaction shown in the following example.

```
LDAA    BLK_WINDOW+OFFS0 ; Read 8-bit as a blocking transaction
STAA    MEM              ; Store result to local Memory
LDAA    BLK_WINDOW+OFFS1 ; Read 8-bit as a blocking transaction
```

### 5.41.5.5 Non-Blocking Read

Read access to the non-blocking window is reserved for future use. When reading from the address window associated with non-blocking writes, the read returns an all 0s data byte or word. This behavior can change in future revisions.

#### 5.41.5.6 Transfer Width

8-bit wide writes or reads are translated into 8-bit wide interface transactions. 16-bit wide, aligned writes or reads are translated into a16-bit wide interface transactions. 16-bit wide, misaligned writes or reads are split up into two consecutive 8-bit transactions with the transaction on the odd address first followed by the transaction on the next higher even address. Due to the much more complex error handling (by the MCU), misaligned 16-bit transfers should be avoided.

#### 5.41.5.7 Error Conditions and Handling faults

Since the S12 CPU (as well as the S08) do not provide a method to abort a transfer once started, the D2DI asserts an D2DERRINT. The ERRIF Flag is set in the D2DSTAT0 register. Depending on the error condition further error flags are set as described below. The content of the address and data buffers are frozen and all transactions are replaced by an IDLE command, until the error flag is cleared. If an error is detected during the read transaction of a read-modify-write instruction or a non-blocking write transaction was followed by another write or read transaction, the second transaction is cancelled. The CNCLF is set in the D2DSTAT0 register to indicate a transaction has been cancelled. The D2DERRINT handler can read the address and data buffer register to assess the error situation. Any further transaction is replaced by IDLE until the ERRIF is cleared.

##### 5.41.5.7.1 Missing Acknowledge

If the target detects a wrong command it does not send back an acknowledge. The same situation occurs if the acknowledge is corrupted. The D2DI detects this missing acknowledge after the timeout period configured in the TIMOUT parameter of the D2DCTL1 register. In case of a timeout the ERRIF and the TIMEF flags in the D2DSTAT0 register is set.

##### 5.41.5.7.2 Parity error

In the final acknowledge cycle of a transaction the target sends two parity bits. If this parity does not match the parity calculated by the initiator, the ERRIF and the PARF flags in the D2DSTAT0 register is set. The PAR[1:0] bits contain the parity value received by the D2DI.

##### 5.41.5.7.3 Error Signal

During the acknowledge cycle the target can signal a target specific error condition. If the D2DI finds the error signal asserted during a transaction, the ERRIF and the TERRF flags in the D2DSTAT0 register is set.

#### 5.41.5.8 Low Power Mode Options

##### 5.41.5.8.1 D2DI in Run Mode

In run mode with the D2D Interface enable (D2DEN) bit in the D2D control register 0 clear, the D2DI system is in a low-power, disabled state. D2D registers remain accessible, but clocks to the core of this module are disabled. On D2D lines the GPIO function is activated.

##### 5.41.5.8.2 D2DI in Stop Mode

If the CPU enters the STOP mode, any pending transmission is completed. When the D2DCLK output is driven low, clock generation is stopped. All internal clocks to the D2DCLK are stopped as well, and the module enters a power saving state.

##### 5.41.5.8.3 Reset

In case of reset any transaction is immediately stopped and the D2DI module is disabled.

##### 5.41.5.8.4 Interrupts

The D2DI only originates interrupt requests, when D2DI is enabled (D2DIE bit in D2DCTL0 set). There are two different interrupt requests from the D2D module. The interrupt vector offset and interrupt priority are chip dependent.

###### 5.41.5.8.4.1 D2D External Interrupt

This is a level sensitive active high external interrupt driven by the D2DINT input. This interrupt is enabled if the D2DIE bit in the D2DCTL1 register is set. The interrupt must be cleared using an target specific clearing sequence. The status of the D2D input pin can be observed by reading the D2DIF bit in the D2DSTAT1 register.

The D2DINIT signal is asserted also in the stop mode; it can be used to leave these modes.

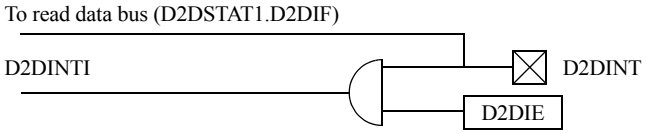



**Figure 115. D2D External Interrupt Scheme**

#### 5.41.5.8.4.2 D2D Error Interrupt

Those D2D interface specific interrupts are level sensitive and are all cleared by writing a 1 to the ERRIF flag in the D2DSTAT0 register. This interrupt is not locally maskable and should be tied to the highest possible interrupt level in the system, on an S12 architecture to the XIRQ. See the chapter “Vectors” of the MCU description for details.

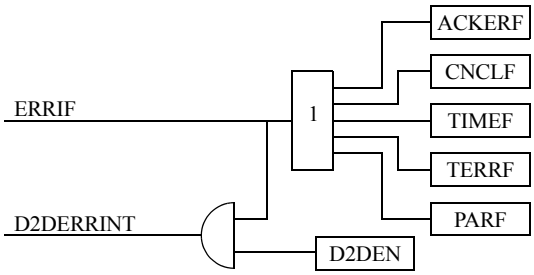



**Figure 116. D2D Internal Interrupts**

### 5.41.6 Initialization Information

During initialization the transfer width, clock divider and timeout value must be set according to the capabilities of the target device before starting any transaction. See the D2D Target specification for details.

### 5.41.7 Application Information

#### 5.41.7.1 Entering low power mode

The D2DI module is typically used on a microcontroller along with an analog companion device containing the D2D target interface and supplying the power. Interface specification does not provide special wires for signalling low power modes to the target device. The CPU should determine when it is time to enter one of the above power modes.The basic flow is as follows:

1. CPU determines there is no more work pending.
2. CPU writes a byte to a register on the analog die using blocking write configuring which mode to enter.
3. Analog die acknowledges the write sending back an acknowledge symbol on the interface.
4. CPU executes STOP command.
5. Analog die can enter low power mode - (S12 needs some more cycles to stack data)

```
; Example shows S12 code
SEI         ; disable interrupts during test
; check is there is work pending?
; if yes, branch off and re-enable interrupt
; else
LDAA        #STOP_ENTRY
STAA        MODE_REG            ; store to the analog die mode reg (use blocking write here)
CLI         ; re-enable right before the STOP instruction
STOP        ; stack and turn off all clocks inc. interface clock
```

For wake-up from STOP the basic flow is as follows:

1. Analog die detects a wake-up condition, e.g. on a switch input or start bit of a LIN message.
2. Analog die exits Voltage Regulator low power mode.
3. Analog die asserts the interrupt signal D2DINT.
4. CPU starts clock generation.
5. CPU enters interrupt handler routine.
6. CPU services interrupt and acknowledges the source on the analog die.

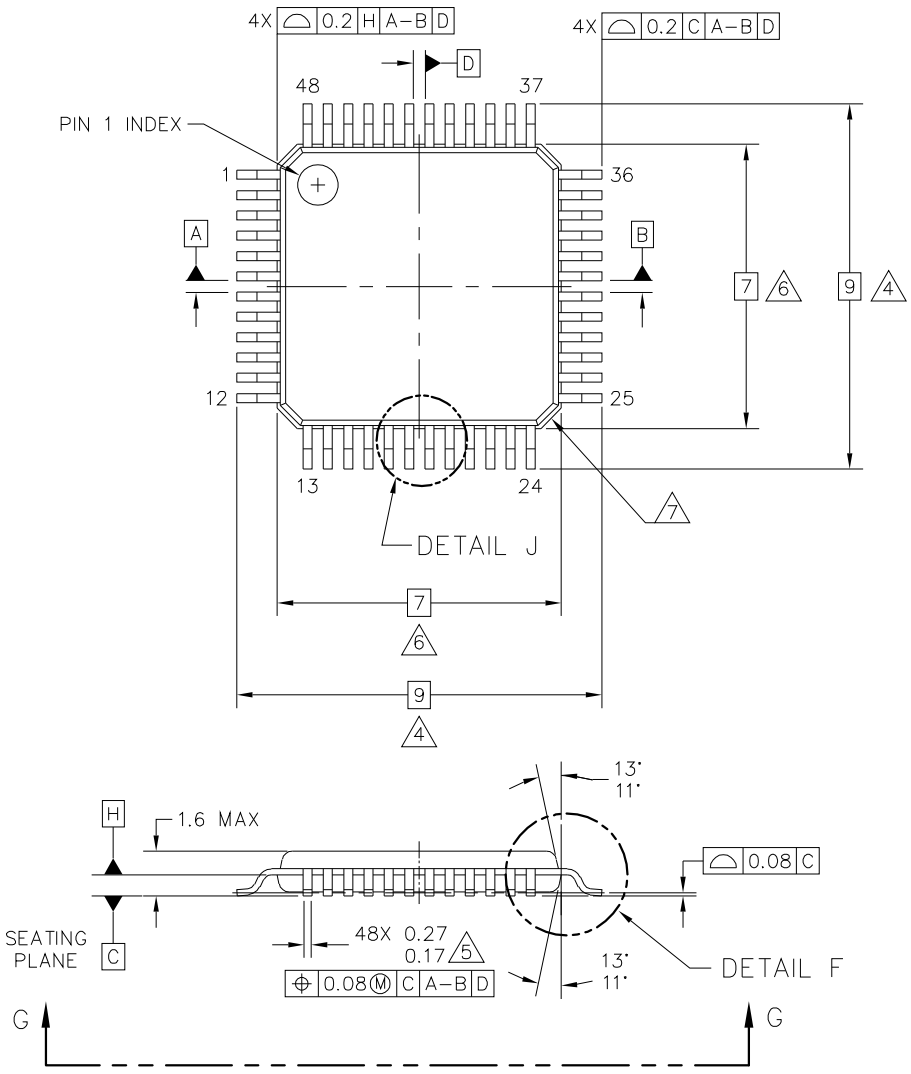
**NOTE**

Entering STOP mode with D2DSWAI asserted the clock completes the high duty cycle portion and settle at low level.

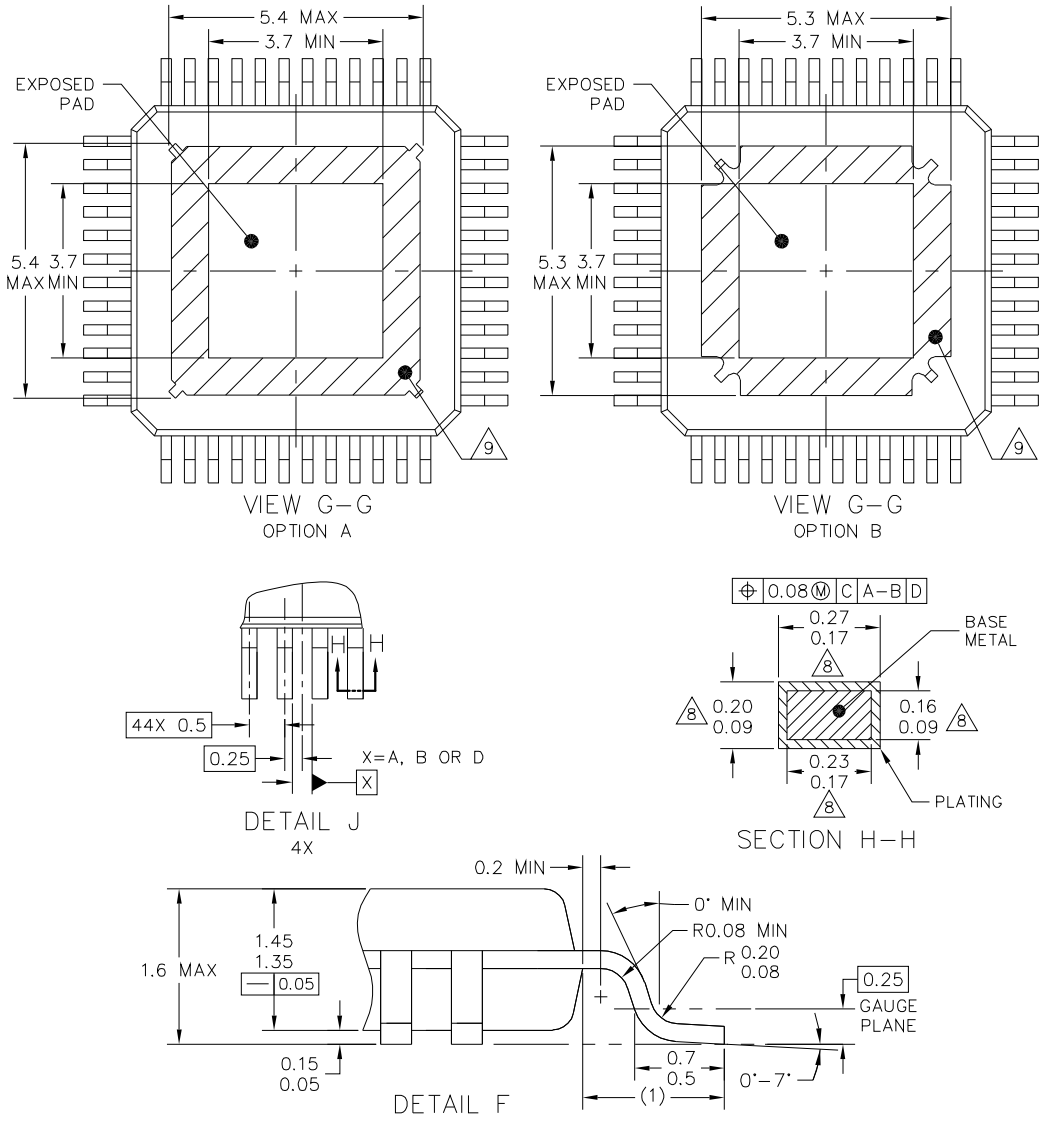
# 6 Packaging

## 6.1 Package Dimensions

For the most current package revision, visit **www.freescale.com** and perform a keyword search using the “98A” listed below.




| © FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED. | MECHANICAL OUTLINE | PRINT VERSION NOT TO SCALE |
|---|---|---|
| TITLE: 48 LEAD LQFP, 7X7X1.4 PKG, 0.5 PITCH, 4.5X4.5 EXPOSED PAD | DOCUMENT NO: 98ASA00173D | REV: C |
| | STANDARD: JEDEC MS-026 BBC | |
| | 17 FEB 2015 | |




| © FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED. | MECHANICAL OUTLINE | PRINT VERSION NOT TO SCALE |
|---|---|---|
| TITLE: 48 LEAD LQFP, 7X7X1.4 PKG, 0.5 PITCH, 4.5X4.5 EXPOSED PAD | DOCUMENT NO: 98ASA00173D | REV: C |
| | STANDARD: JEDEC MS-026 BBC | |
| | 17 FEB 2015 | |

NOTES:

1. DIMENSIONS ARE IN MILLIMETERS.

2. DIMENSIONING AND TOLERANCING PER ASME Y14.5M–1994.

3. DATUMS A, B AND D TO BE DETERMINED AT DATUM PLANE H.

4. DIMENSION TO BE DETERMINED AT SEATING PLANE C.

5. THIS DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED THE UPPER LIMIT BY MORE THAN 0.08MM AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT. MINIMUM SPACE BETWEEN PROTRUSION AND ADJACENT LEAD SHALL NOT BE LESS THAN 0.07MM.

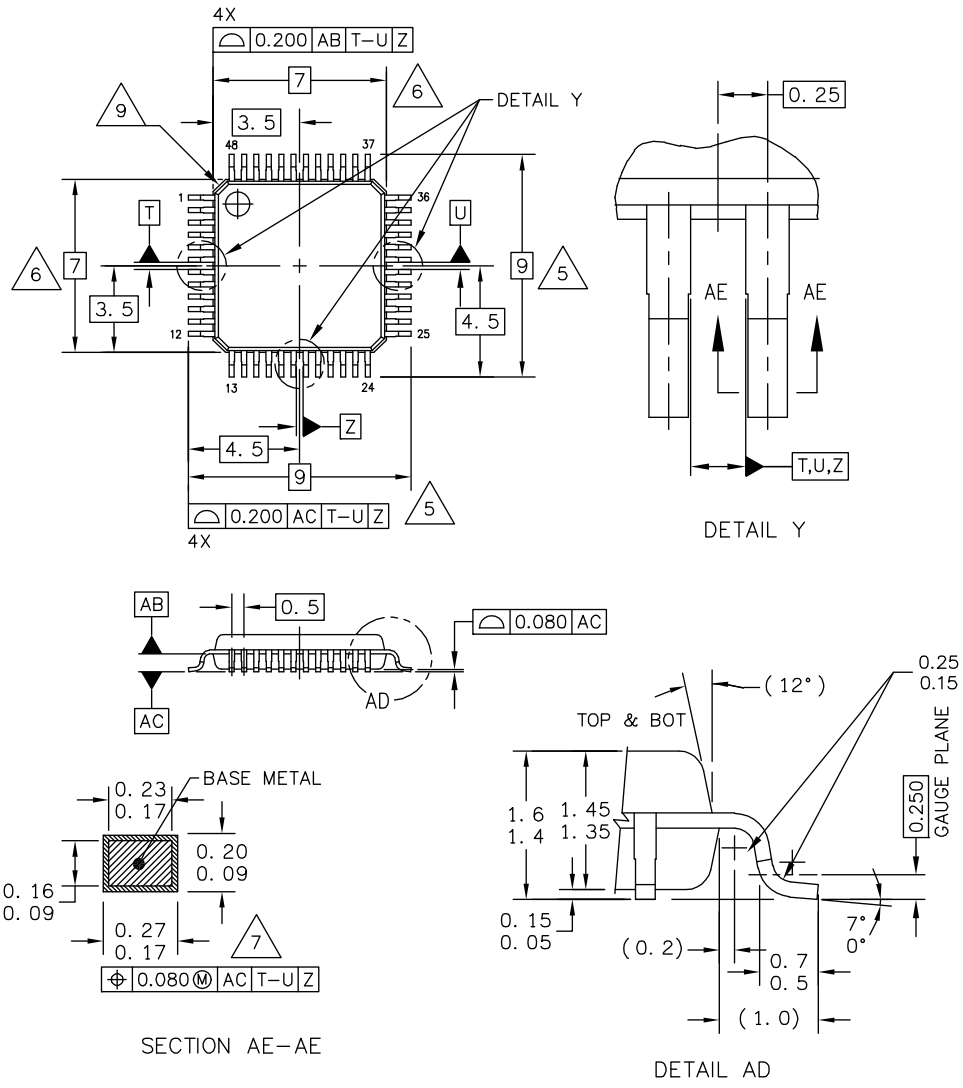
6. THIS DIMENSION DOES NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25MM PER SIDE. THIS DIMENSION IS MAXIMUM PLASTIC BODY SIZE DIMENSION INCLUDING MOLD MISMATCH.

7. EXACT SHAPE OF EACH CORNER IS OPTIONAL.

8. THESE DIMENSIONS APPLY TO THE FLAT SECTION OF THE LEAD BETWEEN 0.1MM AND 0.25MM FROM THE LEAD TIP.

9. HATCHED AREA TO BE KEEP OUT ZONE FOR PCB ROUTING.

| © FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED. | MECHANICAL OUTLINE | PRINT VERSION NOT TO SCALE |
|---|---|---|
| TITLE: 48 LEAD LQFP, 7X7X1.4 PKG, 0.5 PITCH, 4.5X4.5 EXPOSED PAD | DOCUMENT NO: 98ASA00173D | REV: C |
| | STANDARD: JEDEC MS-026 BBC | |
| | 17 FEB 2015 | |




| © FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED. | MECHANICAL OUTLINE | PRINT VERSION NOT TO SCALE |
|---|---|---|
| TITLE: LQFP, 48 LEAD, 0.50 PITCH (7.0 X 7.0 X 1.4) | DOCUMENT NO: 98ASH00962A | REV: G |
| | CASE NUMBER: 932-03 | 14 APR 2005 |
| | STANDARD: JEDEC MS-026-BBC | |

NOTES:

1. DIMENSIONS AND TOLERANCING PER ASME Y14.5M–1994.

2. CONTROLLING DIMENSION: MILLIMETER.

3. DATUM PLANE AB IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.

4. DATUMS T, U, AND Z TO BE DETERMINED AT DATUM PLANE AB.

5. DIMENSIONS TO BE DETERMINED AT SEATING PLANE AC.

6. DIMENSIONS DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.250 PER SIDE. DIMENSIONS DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE AB.

7. THIS DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED 0.350.

8. MINIMUM SOLDER PLATE THICKNESS SHALL BE 0.0076.

9. EXACT SHAPE OF EACH CORNER IS OPTIONAL.

© FREESCALE SEMICONDUCTOR, INC.
ALL RIGHTS RESERVED.

| MECHANICAL OUTLINE | PRINT VERSION NOT TO SCALE |
|---|---|

| TITLE: | DOCUMENT NO: 98ASH00962A | REV: G |
|---|---|---|
| LQFP, 48 LEAD, 0.50 PITCH (7.0 X 7.0 X 1.4) | CASE NUMBER: 932-03 | 14 APR 2005 |
| | STANDARD: JEDEC MS-026-BBC | |

# 7 Revision History

.

| Revision | Date | Description |
|---|---|---|
| 1.0 | 11/2010 | • Initial release. Preliminary. |
| 2.0 | 4/2011 | • Advance Information release. |
| 3.0 | 5/2011 | • Added a note to the Ordering Information Table defining the addition of R2 to the part number<br>• Updated the Table 3 part numbering scheme |
| 4.0 | 5/2011 | • Corrected errors in Table 49. ESD and Latch-up Protection Characteristics |
| 5.0 | 9/2011 | • Corrected errors in table 213, MCU IFR Address<br>• Corrected an address error in table 252. |
| 6.0 | 1/2012 | • Corrected register note in table Table 160.<br>• Corrected header “Access” in Table 190, Table 191, Table 192, and Table 193.<br>• Removed section on the MMC Control Register (MMCCTL1), and references to RAMON, IFRON, and ROMON bits. |
| 7.0 | 3/2012 | • Added to description of pin 46<br>• Corrected Figure 36.<br>• Corrected Table 213<br>• Added P-Flash Protection Register table.<br>• Added information to D@DCTL1 Register Field Description table. |
| 8.0 | 11/2012 | • Various corrections and adjustments throughout the spec. No change of content. |
| 9.0 | 12/2012 | • Split out and better identified Package Thermal Resistance<br>• Added voltage definitions in Table 10<br>• Added note to table 11<br>• Added comments regarding OpenLoad detection<br>• Validated current part numbers |
| 10.0 | 1/2013 | • Update of the FPROT register description |
| 11.0 | 4/2014 | • Updated back page and format links. No change to content. |
| 12.0 | 5/2015 | • Removed all the C devices from the Ordering Information table (obsolete)<br>• Changed Dynamic Electrical Characteristics - Low-side Drivers - LS. Added condition and note.<br>• Added register PTC/PTD/DDRC/DDRD<br>• Corrected typo VLVR and VROV<br>• Added section 5.13.6, “Low-side Driver in Reverse Polarity“ regarding low-side driver in reverse polarity<br>• In 5.26, “MM912_634 - Analog Die Trimming“ changed the wording to better explain the trim<br>• Removed the note: To trim the bg1p25sleep there are two steps:<br>• Added explanation on SLPBG_Lock bit + note to refer to errata<br>• Chapter 5.41.3.2, “D2DDAT[7:4]“: added note regarding D2D non bonded pad<br>• Added note (245)<br>• Updated case outline |
| 13.0 | 6/2015 | • Register PORTB name corrected to PORTE.<br>• $R_{DS(on)}$ of the Hall sensor supply output for 3.7 V< $V_{SUP}$ < 5.5 V range modified<br>• Updated Freescale form and style |

***How to Reach Us:***

**Home Page:**
freescale.com

**Web Support:**
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. SMARTMOS is a trademark of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2015 Freescale Semiconductor, Inc.

Document Number: MM912_634D1
Rev. 13.0
6/2015