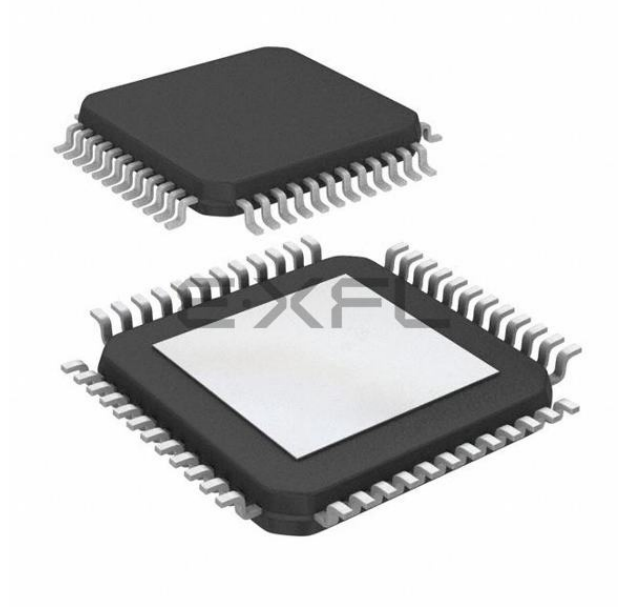


**Welcome to E-XFL.COM**

## Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

**Embedded - Microcontrollers - Application Specific** represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

## What Are Embedded - Microcontrollers - Application Specific?

Application-specific microcontrollers are engineered to

| Details | |
|---|---|
| Product Status | Active |
| Applications | Automotive |
| Core Processor | S12 |
| Program Memory Type | FLASH (48kB) |
| Controller Series | HCS12 |
| RAM Size | 2K x 8 |
| Interface | LIN, SCI |
| Number of I/O | 9 |
| Voltage - Supply | 2.25V ~ 5.5V |
| Operating Temperature | -40°C ~ 105°C |
| Mounting Type | Surface Mount |
| Package / Case | 48-LQFP Exposed Pad |
| Supplier Device Package | 48-HLQFP (7x7) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/mm912g634dm1aer2 |

**Table 5. Signal Properties Summary**

| Pin Name Function 1 | Pin Name Function 2 | Power Supply | Internal Pull Resistor | | Description |
|---|---|---|---|---|---|
| | | | CTRL | Reset State | |
| 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 |
| PC1 | D2DINT | VDDD2D | PUPCE/ D2DEN | Disabled | Port C I/O, D2DI |
| PC0 | D2DCLK | VDDD2D | NA | NA | Port C I/O, D2DI |
| PD7-0 | D2DDAT7-0 | VDDD2D | PUPDE/ D2DEN | Disabled | Port D I/O, D2DI |

**Table 23. Static Electrical Characteristics - Current Sense Module - ISENSE**

| Ratings | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Gain | | | | | |
| CSGS (Current Sense Gain Select) = 000 | G | - | 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 | % |
| Resolution(29) | RES | - | 51 | - | mA/LSB |
| ISENSEH, ISENSEL Input Common Mode Voltage Range | $V_{IN}$ | -0.2 | - | 3.0 | V |
| Current Sense Module - Normal Mode Current Consumption Adder (CSE = 1) | $I_{ISENSE}$ | - | 600 | - | µA |

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

**Table 24. Static Electrical Characteristics - Temperature Sensor - TSENSE**

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

Note:
30. Guaranteed by design and characterization.

**Table 25. Static Electrical Characteristics - Supply Voltage Sense - VSENSE and VS1SENSE**

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

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

### 5.4.6 Analog Die Functionality by Operation Mode

**Table 80. 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(65) | Cyclic Sense(65) |
| ADC | | full | OFF | OFF |
| D2D | | full | functional | OFF |
| Lx | | full | Wake-up(65) | Wake-up(65) |
| PTBx | | full | OFF | OFF |
| LIN | | full | Wake-up(65) | Wake-up(65) |
| Watchdog | | full(66) | OFF | OFF |
| VSENSE | | full | OFF | OFF |
| CSENSE | | full | OFF | OFF |
| Cyclic Sense | | not active | Cyclic Sense(65) | Cyclic Sense(65) |

Note:
65. If configured.
66. Special init through non window watchdog.

### 5.4.7 Register Definition

#### 5.4.7.1 Mode Control Register (MCR)

**Table 81. Mode Control Register (MCR)**

Offset(67) 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:
67. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

**Table 82. MCR - Register Field Descriptions**

| Field | Description |
|---|---|
| 1-0 MODE | Mode Select - These bits will issue a transition from to the selected Operating Mode.<br>00 - Normal Mode. Only with effect in Stop Mode. Will issue Wake Up and transition to Normal Mode.<br>01 - Stop Mode. Will initiate transition to Stop Mode.(68)<br>10 - Sleep Mode. Will initiate transition to Sleep Mode.<br>11 - Normal Mode. |

Note:
68. 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 (fBASE) delay are required between WSR read and MCR write.

### 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.

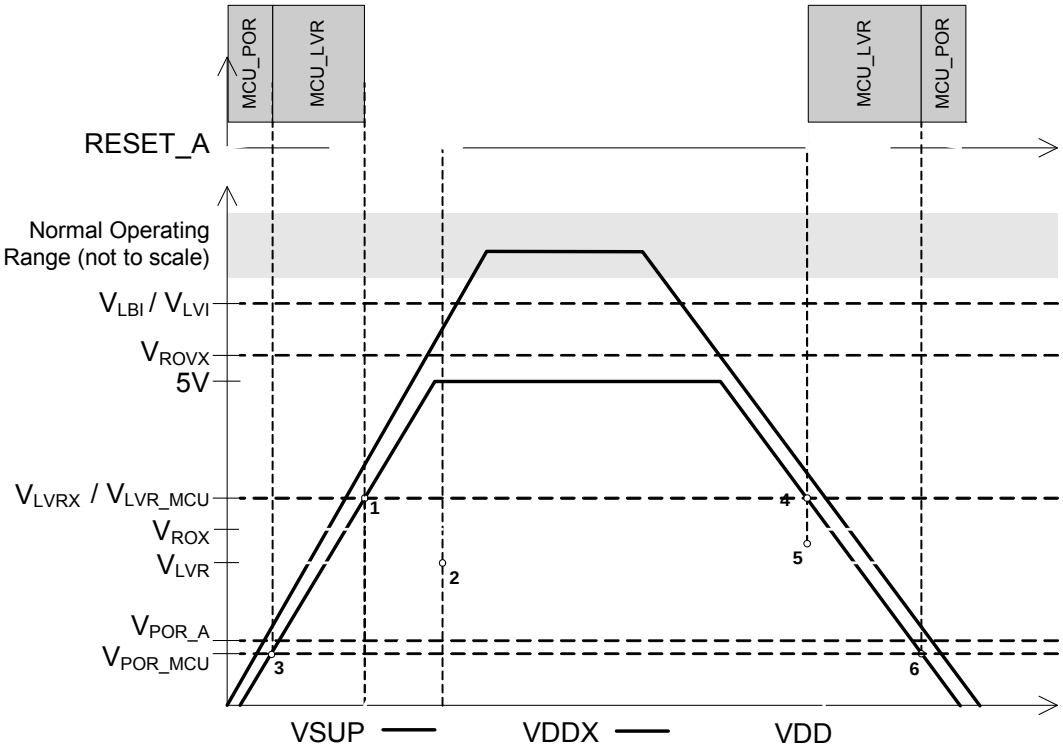



**Figure 18. Power Up / Down Sequence**

To avoid any critical behavior, it is essential to have the MCU Power On Reset (POR) active when the analog die reset ($\overline{\text{RESET\_A}}$) is not fully active. As the $\overline{\text{RESET\_A}}$ circuity is supplied by VDDX, VDD needs to be below the POR threshold when VDDX is to low to guarantee $\overline{\text{RESET\_A}}$ active (3;6). This is achieved with the following implementation.

Power Up:

- The VDD regulator is enabled after VDDX has reached the $V_{LVRX}$ threshold (1).
- Once VDD reaches $V_{LRV}$, the $\overline{\text{RESET\_A}}$ is released (2).
- The MCU is also protected by the MCU_LVR.

Power Down:

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

**NOTE**

The behavior explained previously is essential for the MC9S12I64 MCU die used, as this MCU does have an internal regulator stage, but the LVR function only active in normal modeMC9S12I64.

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

**Table 101. WDR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7 - WDOFF | Watchdog Off - Indicating the Watchdog module is being disabled externally. |
| 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 102. Watchdog Service Register (WDSR)**

Offset(80) 0x11 Access: User read/write

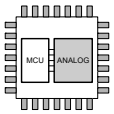
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | WDSR | | | | | | | |
| Reset | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

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

**Table 103. 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 will reset the watchdog counter. Writing the register while the watchdog is disabled will have 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 Over-temperature Shutdown and a Current Limitation is implemented. A write to the Hall Supply Register (HSR), when the over-temperature condition is gone, will re-enable 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 104. Hall Supply Register (HSR)**

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

**Table 105. HSR - Register Field Descriptions**

| Field | Description |
|---|---|
| 7 - HOTIE | Hall Supply Over-temperature Interrupt Enable |
| 6 - HOTC | Hall Supply Over-temperature Condition present. During the event, the Hall Supply is shut down. Reading the register will clear 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 |

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.

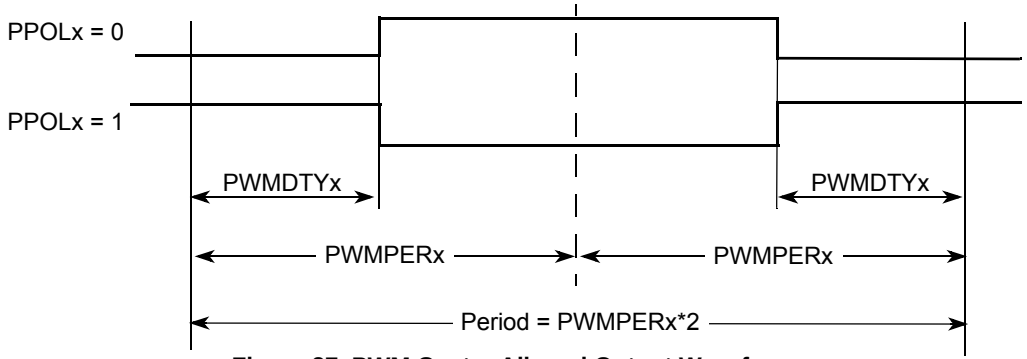



**Figure 27. PWM Center Aligned Output Waveform**

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 that 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 28 shows the output waveform generated.

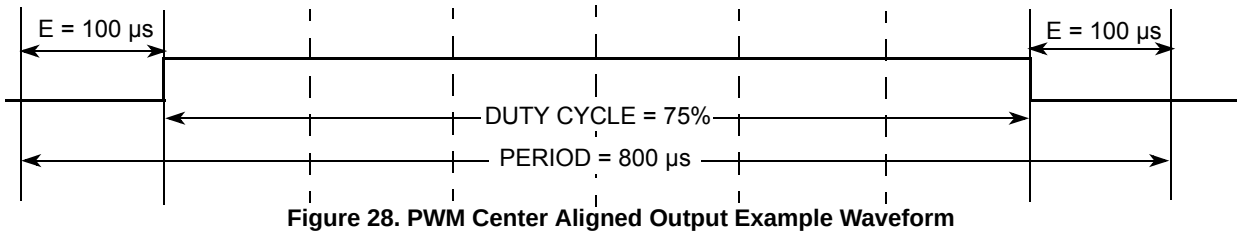
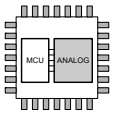



**Figure 28. PWM Center Aligned Output Example Waveform**

## 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)
- Over-temperature 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 Over-temperature Shutdown (LIN Interrupt)

The output Low Side FET (transmitter) is protected against over-temperature conditions. In case of an over-temperature condition, the transmitter will be 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 will be 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 over-temperature 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, will generate a wake-up event and be reported in the Wake-up Source Register (WSR).

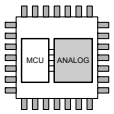
### 5.15.5 J2602 Compliance

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

When an under-voltage 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 under-voltage condition (LVI) is gone, the LIN will start operating when Tx is in a recessive state or on the next dominant to recessive transition.

## 5.26 MM912_634 - Analog Die Trimming




A trimming option is implemented to increase some device parameter accuracy. As the MM912_634 analog die is exclusively combined with a FLASH- MCU, the required trimming values can be calculated during the final test of the device, and stored to a fixed position in the FLASH memory. During start-up of the system, the trimming values have to be copied into the MM912_634 analog die trimming registers.

The trimming registers will maintain their content during Low Power mode, Reset will set the default value.

### 5.26.1 Memory Map and Register Definition

#### 5.26.1.1 Module Memory Map

There are four trimming registers implemented (CTR0…CTR3), with CTR2 being reserved for future use. The following table shows the registers used.

**Table 212. MM912_634 Analog Die Trimming Registers**

| Offset | 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:
148. 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 213. MM912_634 - MCU vs. Analog Die Trimming Register Correlation**

| Name | MCU IFR Address | Analog Offset[149] |
|---|---|---|
| CTR0 | 0x0_40C0 | 0xF0 |
| CTR1 | 0x0_40C1 | 0xF1 |
| CTR2 | 0x0_40C2 | 0xF2 |
| CTR3 | 0x0_40C3 | 0xF3 |

Note:
149. 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 (Program Page Index Register (PPAGE))

**NOTE**

To trim the bg1p25sleep there is two steps:

Step 1: First choose the right trim step by adjusting SLPBGTR[2:0] with SLPBGTRE=1, SLPBG_LOCK bit has to stay at 0.

Step 2: Once the trim value is known, correct SLPBGTR[2:0], SLPBGTRE and SLPBG_LOCK bits have to be set at the same time to apply and lock the trim. Once the trim is locked, no other trim on the parameter is possible.

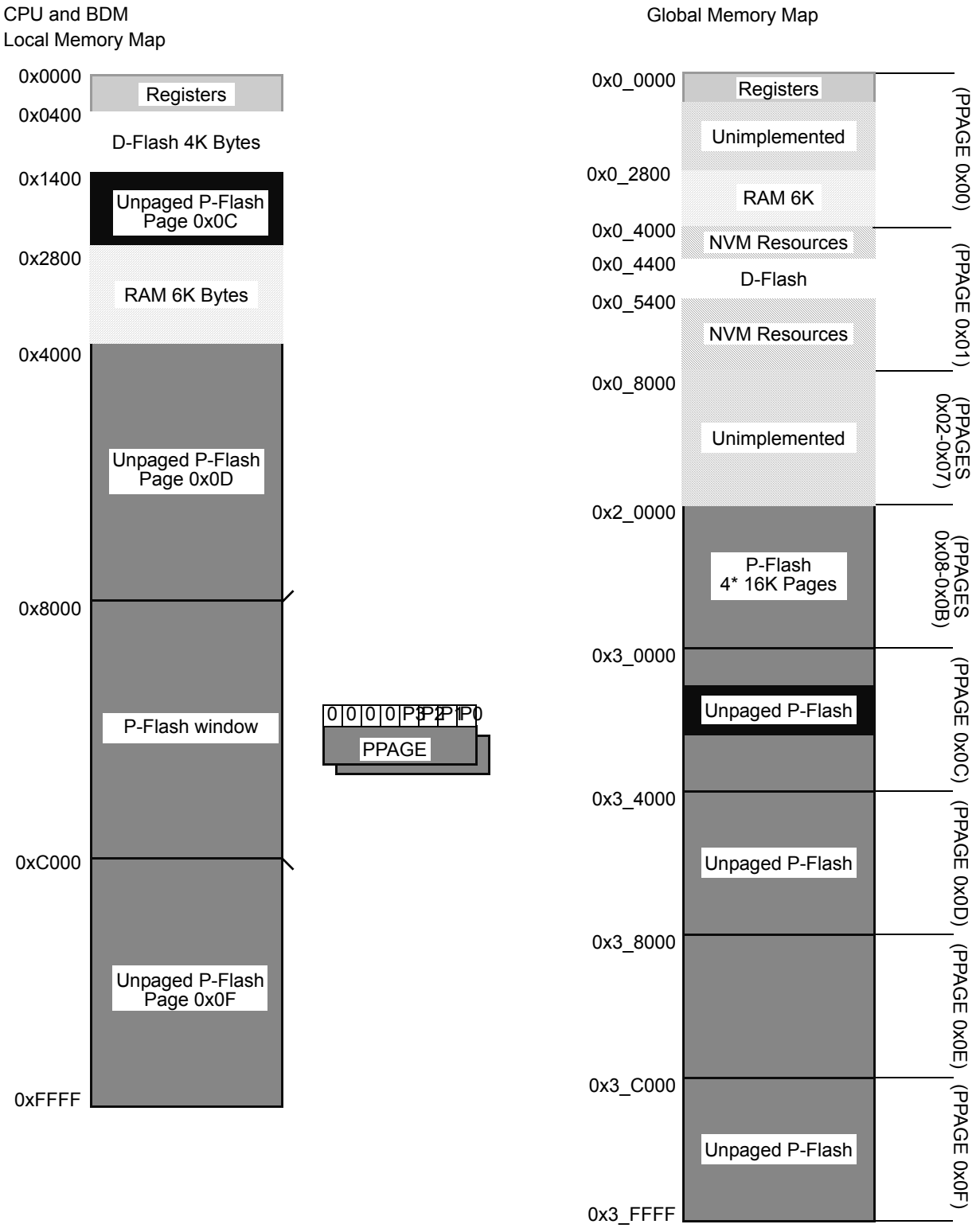



**Figure 41. MC9S12I128 Global Memory Map**

**Table 259. Exception Vector Map and Priority**

| Vector Address[172] | Source |
|---|---|
| (Vector base + 0x00F4) | X bit maskable interrupt request (XIRQ or D2D error interrupt)[173] |
| (Vector base + 0x00F2) | IRQ or D2D interrupt request[174] |
| (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:
172. 16 bits vector address based
173. D2D error interrupt on MCUs featuring a D2D initiator module, otherwise XIRQ pin interrupt
174. 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 that 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.

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.

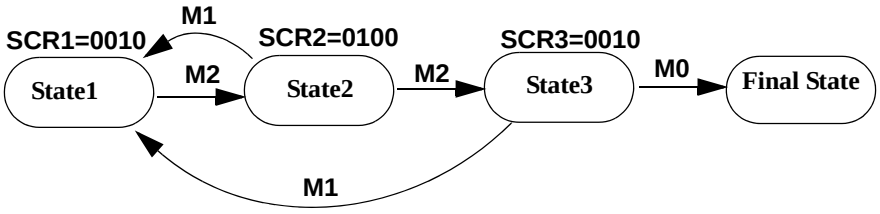



**Figure 79. Scenario 10a**

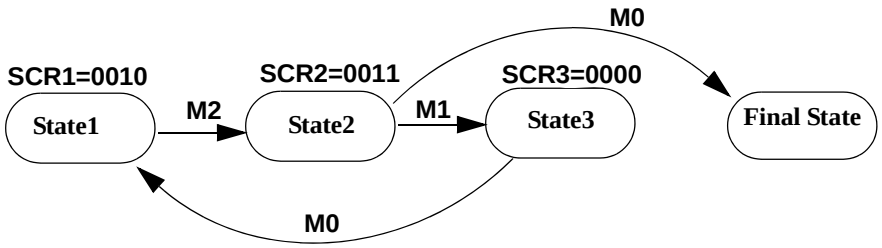



**Figure 80. Scenario 10b**

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.

### 5.38.1.3 S12CPMU Block Diagram

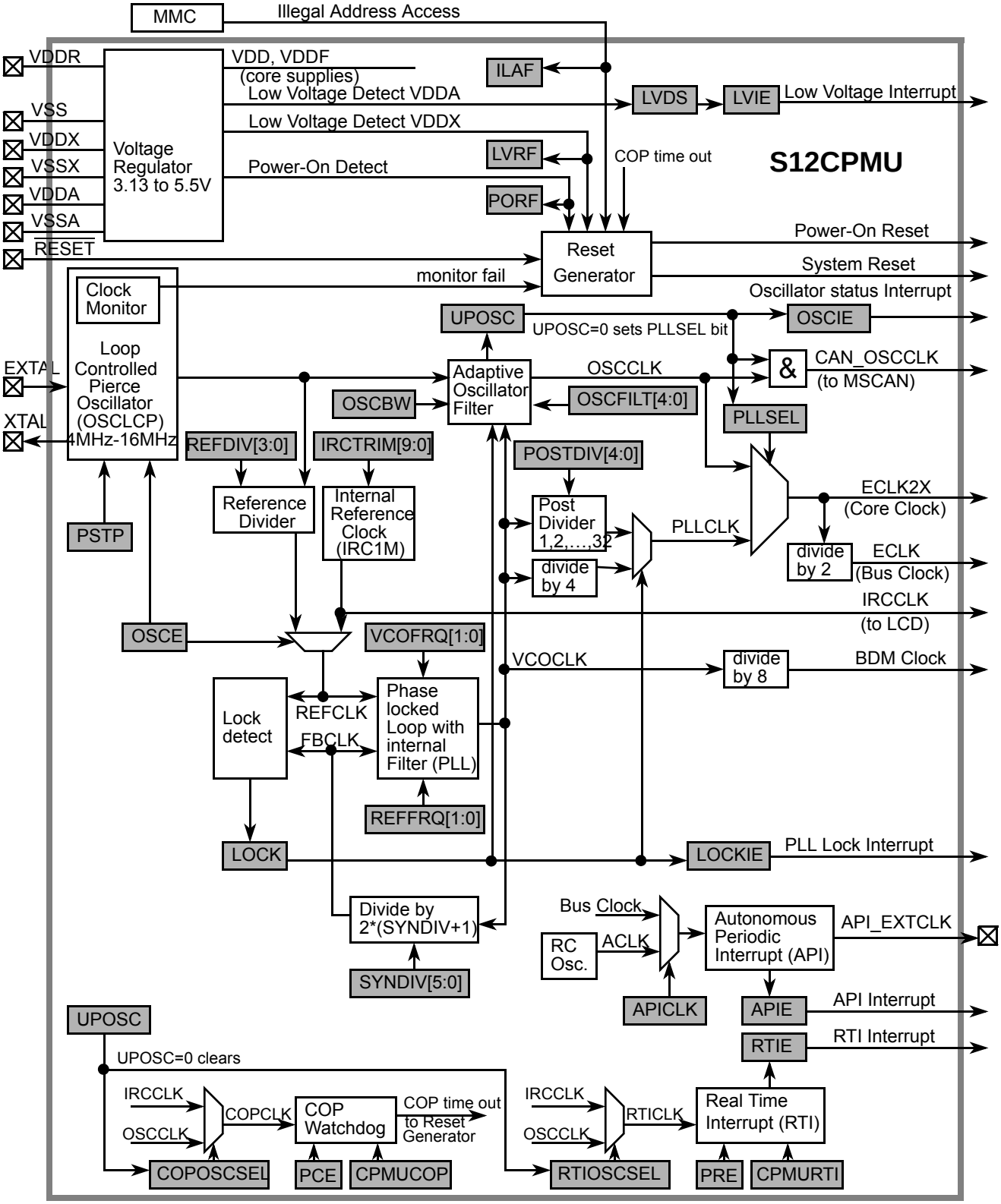



**Figure 82. Block diagram of S12CPMU**

**NOTE**

Write to this register clears the LOCK and UPOSC status bits.

**NOTE**

Care should be taken to ensure that the bus frequency does not exceed the specified maximum when frequency modulation is enabled.

**NOTE**

The frequency modulation (FM1 and FM0) can not be used if the Adaptive Oscillator Filter is enabled.

**Table 351. 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 352 for coding. |

**Table 352. 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 353. 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 374. 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 375. 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 376. 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 377 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 377. 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 |

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(201). This needs to be dealt with in application software.

Note:

201. For details please refer to “5.38.4.6, “System Clock Configurations”

**Table 489. 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 456) |
| | | 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 will confirm that the targeted location(s) were successfully programmed upon completion.

A Flash word must be in the erased state before being programmed. Cumulative programming of bits within a Flash word is not allowed.

**Table 490. 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 will be 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 will be programmed in the D-Flash block. The CCIF flag is set when the operation has completed.

**Table 491. 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 456) |
| | | 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 |

### 5.41.3.8 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 will be 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 509. 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 510. 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.

## 5.41.4 Functional Description

### 5.41.4.1 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).

### 5.41.4.2 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 112 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 112).

#### 5.41.4.8.4.1 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.

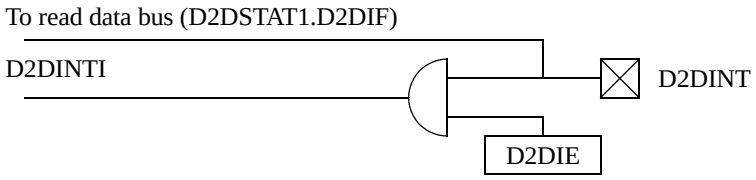



**Figure 113. D2D External Interrupt Scheme**

#### 5.41.4.8.4.2 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.

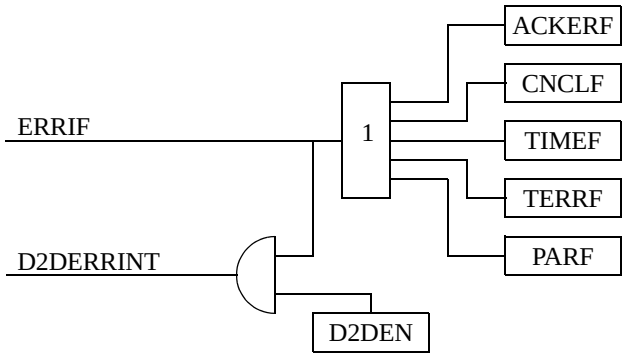



**Figure 114. D2D Internal Interrupts**

### 5.41.5 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.

### 5.41.6 Application Information

#### 5.41.6.1 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 that 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)
```

***How to Reach Us:***

**Home Page:**
freescale.com

**Web Support:**
freescale.com/support



Document Number: MM912_634D1
Rev. 10.0
1/2013