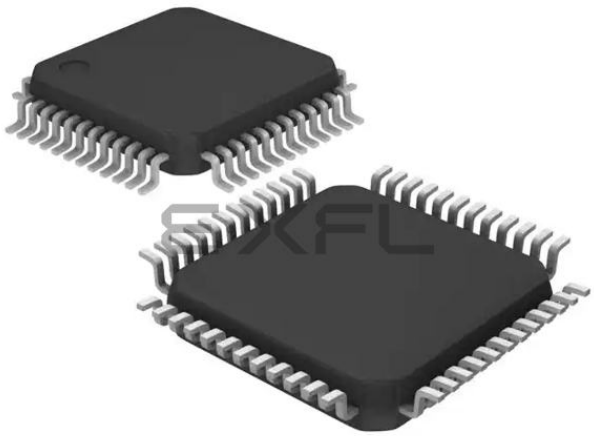


Welcome to E-XFL.COM

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
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mm912g634dv2ap

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 16. Static Electrical Characteristics - Hall Sensor Supply Output - HSUP

Ratings	Symbol	Min	Typ	Max	Unit
Current Limitation	I_{HSUP}	40	70	90	mA
Output Drain-to-Source On resistance T_J = 150 °C, ILOAD = 30 mA; 5.5 V ≤ VSUP ≤ 16 V T_J = 150 °C, ILOAD = 30 mA; 3.7 V ≤ VSUP < 5.5 V	$R_{DS(ON)}$	- -	- -	10 12	Ohm
Output Voltage: (18 V ≤ V_{SUP} ≤ 27 V)	$VHSUP_{MAX}$	16	17.5	18	V
Load Regulation (1.0 mA < I_{HSUP} < 30 mA; V_{SUP} > 18 V)	LD_{HSUP}	-	-	500	mV
Hall Supply Capacitor Range	C_{HSUP}	0.22	-	10	µF
External Capacitor ESR	C_{HSUP_R}	-	-	10	Ohm

Table 17. Static Electrical Characteristics - High Side Drivers - HS

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

Table 18. Static Electrical Characteristics - Low Side Drivers - LS

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

Table 19. Static Electrical Characteristics - LIN Physical Layer Interface - LIN

Ratings	Symbol	Min	Typ	Max	Unit
Current Limitation for Driver dominant state. V_{BUS} = 18 V	I_{BUSLIM}	40	120	200	mA
Input Leakage Current at the Receiver incl. Pull-up Resistor RSLAVE; Driver OFF; V_{BUS} = 0 V; V_{BAT} = 12 V	$I_{BUS\_PAS\_DOM}$	-1.0	-	-	mA
Input Leakage Current at the Receiver incl. Pull-up Resistor RSLAVE; Driver OFF; 8.0 V < V_{BAT} < 18 V; 8.0 V < V_{BUS} < 18 V; V_{BUS} ≥ V_{BAT}	$I_{BUS\_PAS\_REC}$	-	-	20	µA

Assuming that 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:

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

Table 74. Analog die Registers[63] - 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
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								
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								

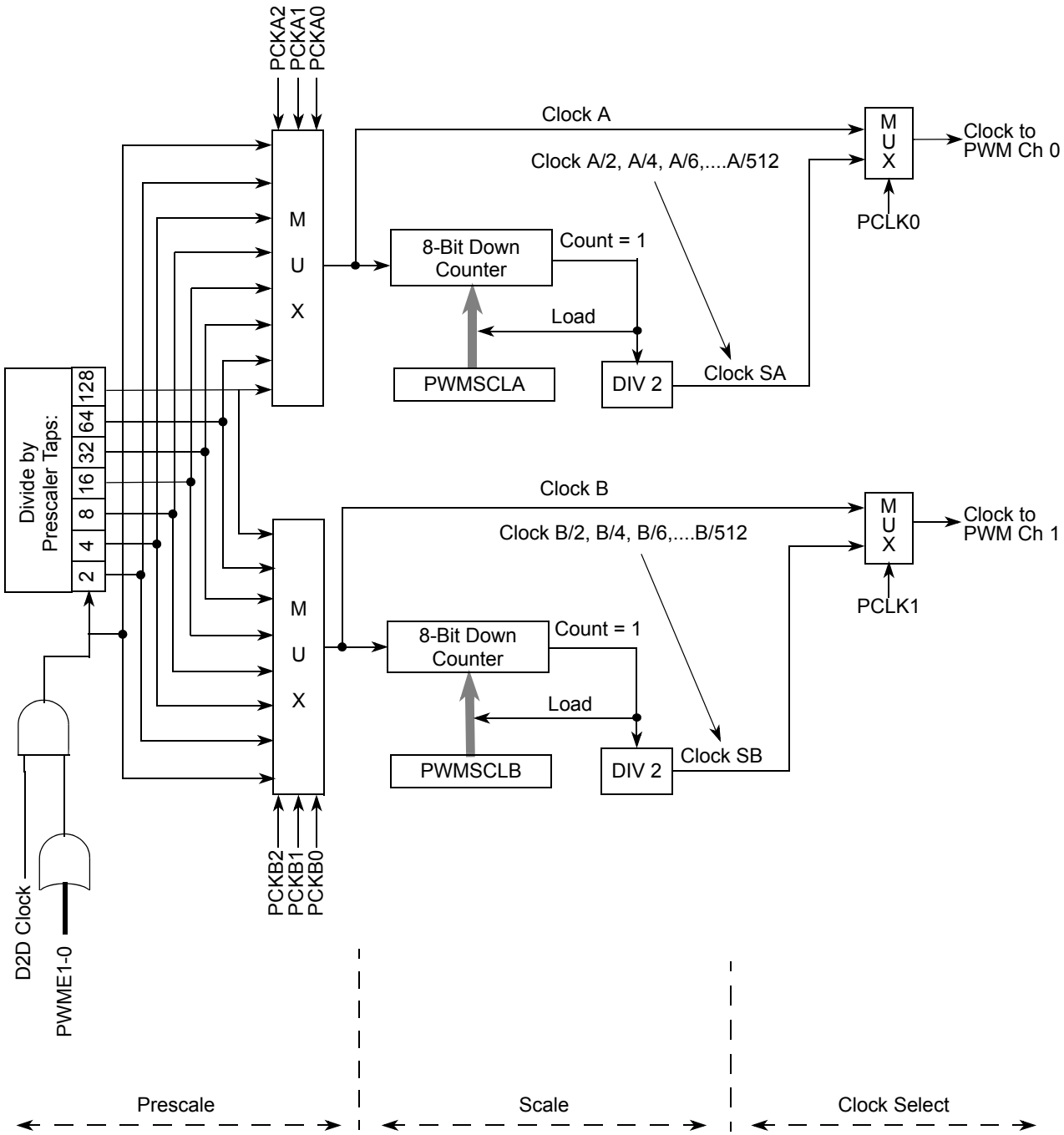


Figure 23. PWM Clock Select Block Diagram

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.

Table 158. PTBC2 - Register Field Descriptions

Field	Description
3 PWMCS	PWM Channel Select PTB2. See Section 5.14, “PWM Control Module (PWM8B2C)". 0 - PWM Channel 0 selected as PWM Channel for PTB2 1 - PWM Channel 1 selected as PWM Channel for PTB2
2 PWMEN	PWM Enable for PTB2. See Section 5.14, “PWM Control Module (PWM8B2C)". 0 - PWM disabled on PTB2 1 - PWM enabled on PTB2 (Channel as selected with PWMCS)
1-0 SERMOD	Serial Mode Select for PTB0 and PTB1. See Figure 32 for details. 00 - Mode 0, SCI internally connected the LIN Physical Layer Interface. PTB0 and PTB1 are Digital I/Os 01 - Mode 1, SCI connected to PTB0 and PTB1 (external SCI mode) 10 - Mode 2, LIN Physical Layer Interface connected to PTB0 and PTB1 (external LIN mode) 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 159. Port B Data Register (PTB)

Offset[113] 0x22 — Access: User read/write

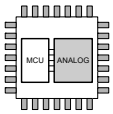
	7	6	5	4	3	2	1	0
R	0	0	0	0	0	PTB2	PTB1	PTB0
W								
Reset	0	0	0	0	0	0	0	0

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

Table 160. PTB - Register Field Descriptions

Field	Description
2-0 PTB[2-0]	Port B general purpose input/output data — Data Register 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.

5.20 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.

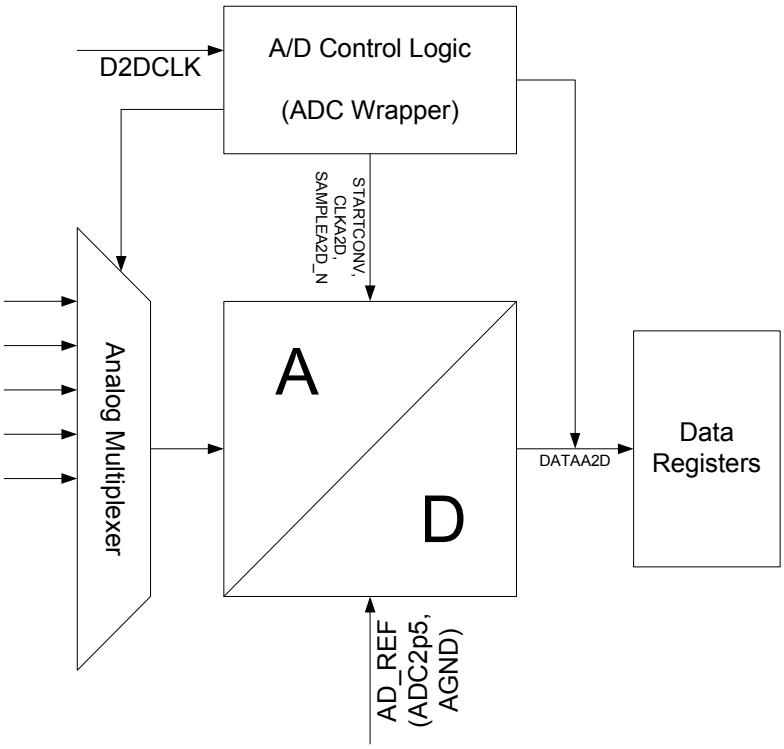


Figure 35. Analog Digital Converter Block Diagram

5.20.1.2 Features

- 10-bit resolution
- 13 µs (typ.), 10-bit Single Sample + Conversion Time
- External ADC2p5 pin with over-current 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 that do connect off-chip. Table 197 shows all the pins and their functions that are controlled by the Analog Digital Converter Module.

5.20.4.2.4 ADC Conversion Complete Status Register (ACCSR)

Table 205. ADC Conversion Complete Status Register (ACCSR)

Offset[142] 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:
142. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 206. 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 will return the current status, no latching will be performed.

5.20.4.2.5 ADC Data Result Register x (ADRx)

Table 207. ADC Data Result Register x (ADRx)

Offset[143] 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:
143. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 208. ADRx - Register Field Descriptions

Field	Description
15-6 ADRx	ADC - Channel X left adjusted Result Register. Reading the register will clear the corresponding CCx register in the ACCSR register. 16-bit read recommended. 8-Bit read: Reading the low byte will latch the high byte for the next read, reading the high byte will clear 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 209. 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

Table 211. CSR - Register Field Descriptions

Field	Description
7 CSE	Current Sense Enable Bit 0 - Current Sense Module Disabled 1 - Current Sense Module Enabled
3 CCD	Input Filter Charge Compensation Disable Bit(146) 0 - Enabled 1 - Disabled
2-0 CSGS	Current Sense Gain Select - Selects the amplification GAIN for the current sense module 000 - 7 (typ.) 001 - 9 (typ.) 010 - 10 (typ.) 011 - 12 (typ.) 100 - 14 (typ.) 101 - 18 (typ.) 110 - 24 (typ.) 111 - 36 (typ.)

Note:

146. 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.

Table 229. PORTA Register Field Descriptions

Field	Description
7–4 PA	**Port A general purpose input/output data**—Data RegisterIn output mode the register bit is driven to the pin. 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 PA	**Port A general purpose input/output data**—Data Register, SPI $\overline{SS}$ input/output When not used with the alternative function, this pin can be used as general purpose I/O. In general purpose output mode the register bit is driven to the pin. 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. • The SPI function takes precedence over the general purpose I/O function if enabled.
2 PA	**Port A general purpose input/output data**—Data Register, SPI SCK input/output When not used with the alternative function, this pin can be used as general purpose I/O. In general purpose output mode the register bit is driven to the pin. 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. • The SPI function takes precedence over the general purpose I/O function if enabled.
1 PA	**Port A general purpose input/output data**—Data Register, SPI MOSI input/output When not used with the alternative function, this pin can be used as general purpose I/O. In general purpose output mode the register bit is driven to the pin. 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. • The SPI function takes precedence over the general purpose I/O function if enabled.
0 PA	**Port A general purpose input/output data**—Data Register, SPI MISO input/output When not used with the alternative function, this pin can be used as general purpose I/O. In general purpose output mode the register bit is driven to the pin. 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. • The SPI function takes precedence over the general purpose I/O function if enabled.

5.28.2.3 Port E Data Register (PORTE)

Table 230. Port E Data Register (PORTE)

Address 0x0001 Access: User read/write(157)

	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:
157. Read: Anytime.
Write: Anytime.

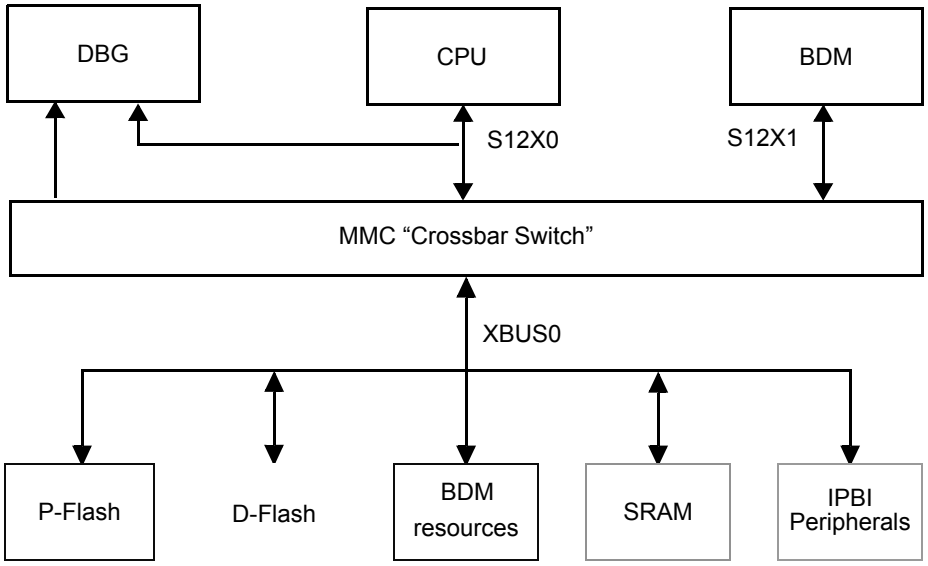


Figure 51. S12I Platform

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 will be stalled after finishing the current operation and the BDM will gain 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 that automate page switching in the program page window. The CALL instruction is similar to the JSR instruction, but the subroutine that 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 that must be known at the time of assembly.

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.

Table 297. State3 — Sequencer Next State Selection

SC[3:0]	Description (Unspecified matches have no effect)
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 324 dictate that 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 298. 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 that flag.

5.32.3.2.8 Comparator Register Descriptions

Each comparator has a bank of registers that are 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 385. 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 386. 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). Writing 0x26 to the CPMUPROT register clears the PROT bit, other write accesses set the PROT bit. 0 Protection of clock configuration registers is disabled. 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 387. 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.

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 105.

5.40.2 External Signal Description

The Flash module contains no signals that connect 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 will be 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 that allows the MCU to restrict access to the Flash module are stored in the Flash configuration field as described in Table 411.

Table 440. DFPROT Field Descriptions

Field	Description
7 DPOPEN	**D-Flash Protection Control** 0 Enables D-Flash memory protection from program and erase with protected address range defined by DPS bits 1 Disables D-Flash memory protection from program and erase
3–0 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 441.

5.40.3.2.12 Flash Common Command Object Register (FCCOB)

Table 441. 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
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 442. Flash Common Command Object High Register (FCCOBHI)

	7	6	5	4	3	2	1	0
R W	CCOB[15:8]							
Reset	0	0	0	0	0	0	0	0

Table 443. Flash Common Command Object Low Register (FCCOBLO)

	7	6	5	4	3	2	1	0
R 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

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 455.

Table 455. 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 that is 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 417 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 will not execute and the ACCERR bit in the FSTAT register will set.

5.40.4.3.2 Command Write Sequence

The Memory Controller will launch 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.

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 will remain clear until the Flash command has completed. Upon completion, the Memory Controller will return CCIF to 1 and the FCCOB register will be used to communicate any results. The flow for a generic command write sequence is shown in Figure 107.

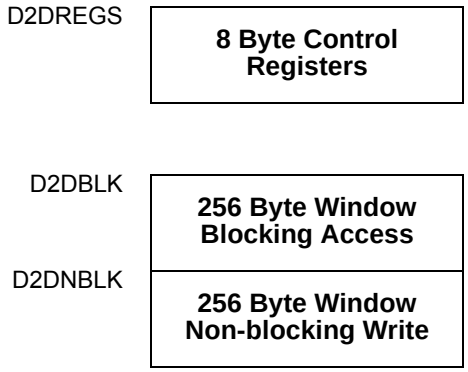


Figure 110. D2DI Top Level Memory Map

A summary of the registers associated with the D2DI block is shown in Table 498. Detailed descriptions of the registers and bits are given in the subsections that follow.

Table 498. 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							

5.41.3.2 Register Definition

5.41.3.3 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 499. D2DI Control Register 0 (D2DCTL0)

Offset 0x0 — Access: User read/write

	7	6	5	4	3	2	1	0
R	D2DEN	D2DCW	D2DSWAI	0	0	0	D2DCLKDIV[1:0]	
W								
Reset	0	0	0	0	0	0	0	0

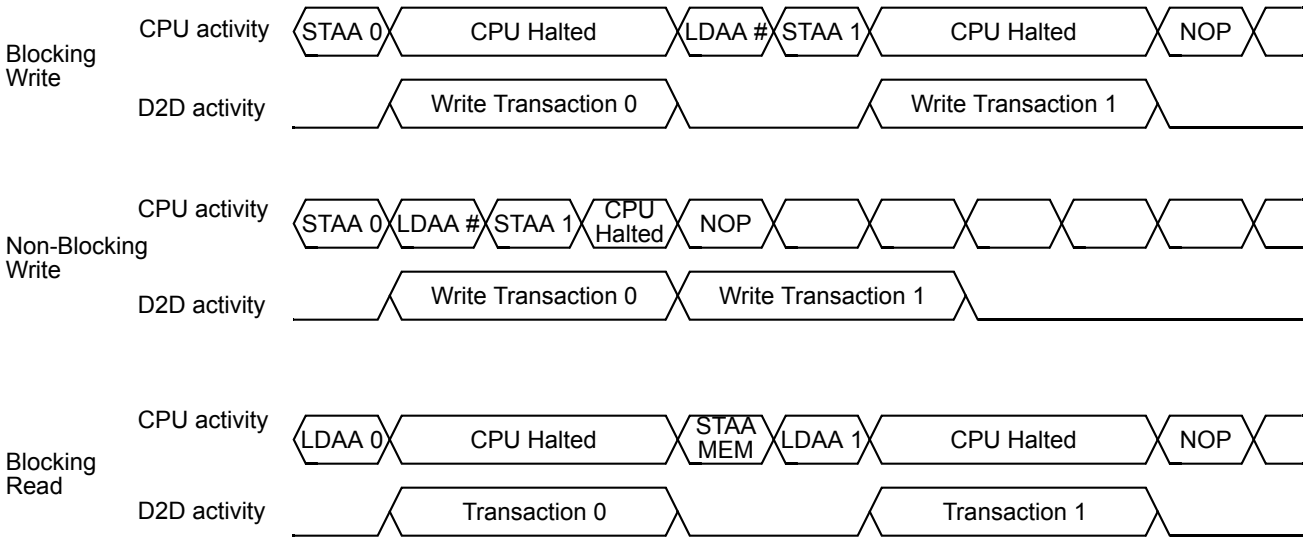


Figure 112. Blocking and Non-Blocking Transfers.

5.41.4.2.1 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 112 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 that the operation at the target is completed before proceeding with the CPU instruction stream.

5.41.4.3 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 112 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 that the following instructions are not affected by the change in the target caused by the previous transaction.

5.41.4.4 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 112 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
```

5.41.4.8.4.1 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.

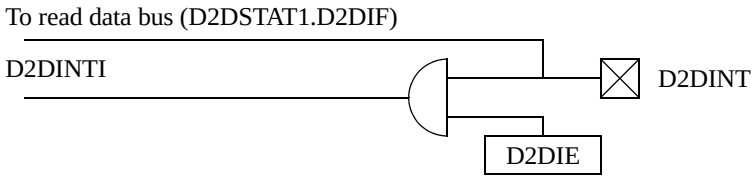


Figure 113. D2D External Interrupt Scheme

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.

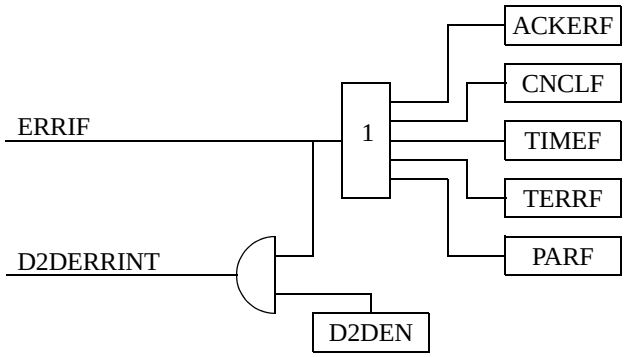


Figure 114. D2D Internal Interrupts

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)
```