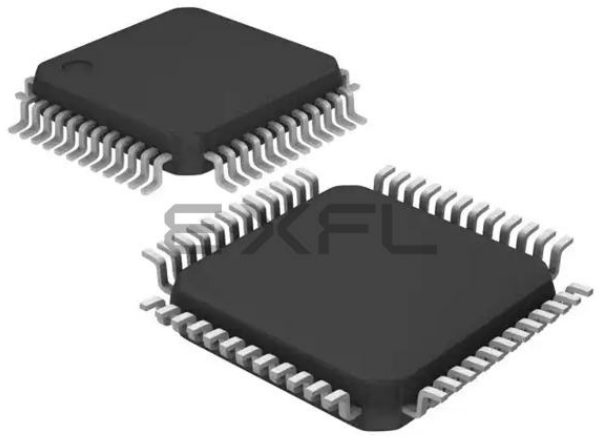


Welcome to E-XFL.COM

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	Obsolete
Applications	Automotive
Core Processor	S12
Program Memory Type	FLASH (32KB)
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/mm912f634cv2apr2

Table of Contents

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

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

Table 21. Static Electrical Characteristics - Analog Digital Converter - ADC(24)

Ratings	Symbol	Min	Typ	Max	Unit
ADC2p5 Reference Voltage 5.5 V < V_{SUP} < 27 V	$V_{ADC2p5RUN}$	2,45	2.5	2,55	V
ADC2p5 Reference Stop Mode Output Voltage	$V_{ADC2p5STOP}$	-	-	100	mV
Line Regulation, Normal Mode	LR_{RUNA}	-	10	12.5	mV
External Capacitor	C_{ADC2p5}	0.1	-	1.0	µF
External Capacitor ESR	C_{VDD_R}	-	-	10	W
Scale Factor Error	E_{SCALE}	-1	-	1	LSB
Differential Linearity Error	E_{DNL}	-1.5	-	1.5	LSB
Integral Linearity Error	E_{INL}	-1.5	-	1.5	LSB
Zero Offset Error	E_{OFF}	-2.0	-	2.0	LSB
Quantization Error	E_Q	-0.5	-	0.5	LSB
Total Error with offset compensation	TE	-5.0	-	5.0	LSB
Bandgap measurement Channel (CH14) Valid Result Range (including ±7.0% bg1p25 sleep accuracy + high-impedance measurement error of ±5.0% at f_{ADC})(25)	AD_{CH14}	1.1	1.25	1.4	V

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

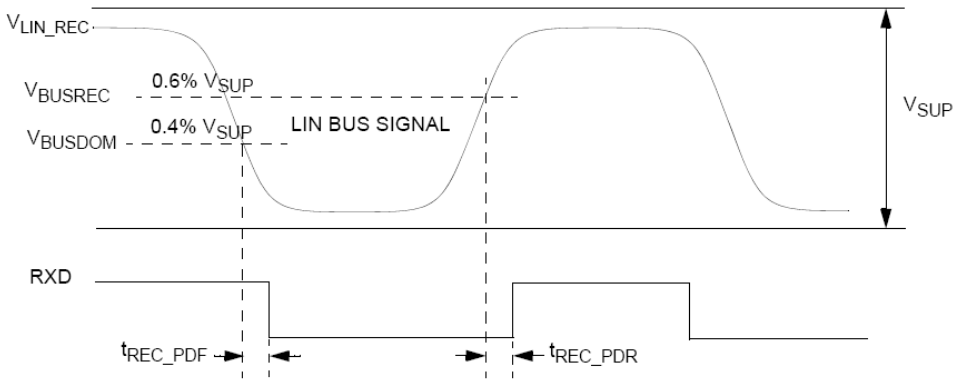


Figure 8. LIN Receiver Timing

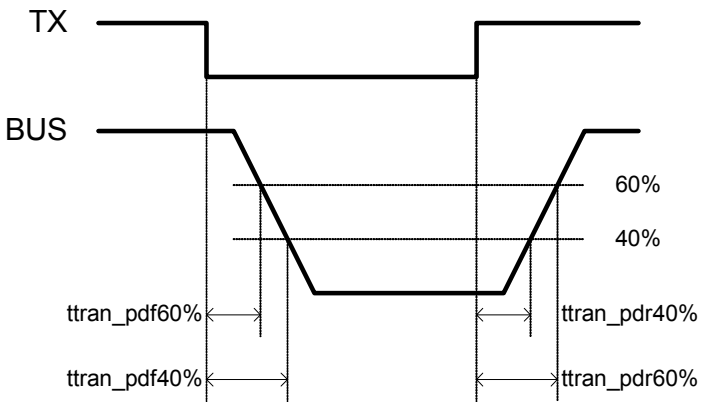


Figure 9. LIN Transmitter Timing

Table 35. Dynamic Electrical Characteristics - General Purpose I/O - PTB[0…2]

Ratings	Symbol	Min	Typ	Max	Unit
GPIO Digital Frequency[37]	f_{PTB}	-	-	10	MHz
Propagation Delay - Rising Edge[37], [38]	t_{PDR}	-	-	20	ns
Rise Time - Rising Edge[37]	t_{RISE}	-	-	17.5	ns
Propagation Delay - Falling Edge[37]	t_{PDF}	-	-	20	ns
Rise Time - Falling Edge[37]	t_{FALL}	-	-	17.5	ns

Note:
37. Guaranteed by design.
38. Load PTBx = 100 pF.

Table 63. 0x0120–0x0123 Port Integration Module (PIM) 2 of 2 (continued)

0x0122	RDRA	R	0	0	RDRA5	RDRA4	RDRA3	RDRA2	RDRA1	RDRA0
		W								
0x0123	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 64. 0x0124–0x01FF Reserved Register Space

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0124-0x01FF	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 65 shows the detailed module maps of the MM912F634 analog die.

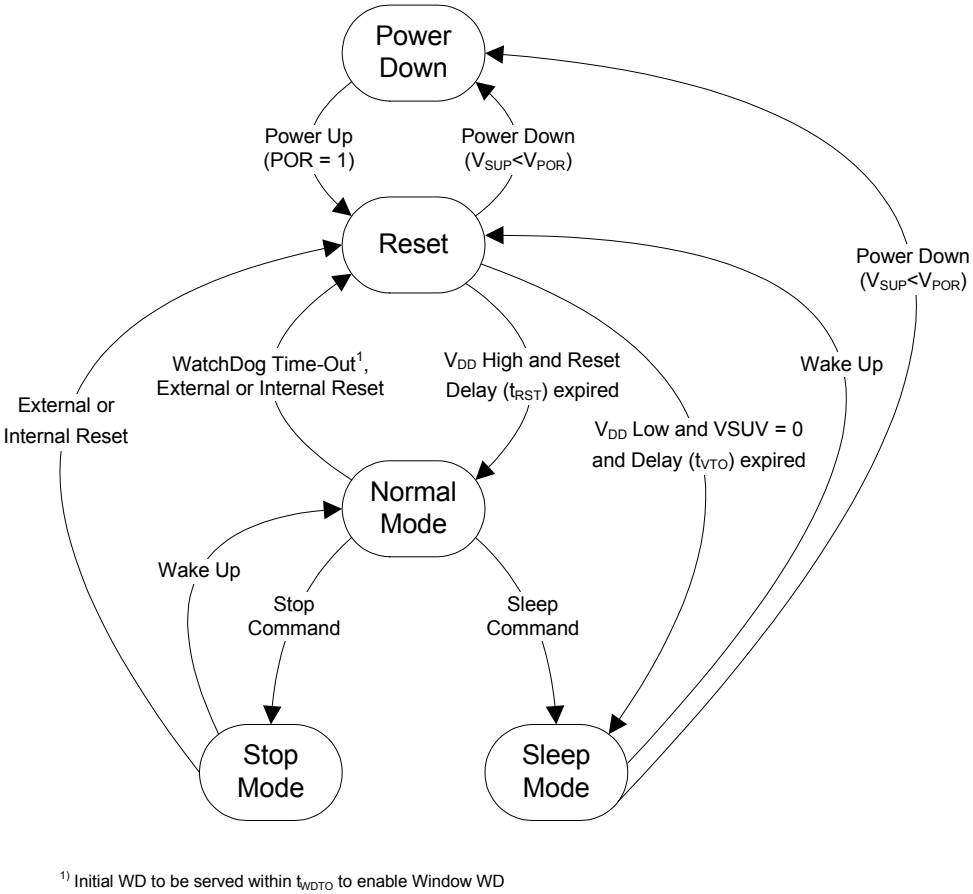
Table 65. Analog die Registers[64] - 0x0200–0x02FF D2D Blocking Access (D2DI) 2 of 3/ 0x0300–0x03FF D2D Non Blocking Access (D2DI) 3 of 3

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								

Table 65. 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
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								
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								
0xF4	SRR	R	0	0	0	0	FMREV		MMREV	
	Silicon Revision Register	W								

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



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

Figure 16. Modes of Operation and Transitions

4.3.1 Power Down Mode

For the device power (VS1) below V_{POR}, the MM912F634 analog die is virtually in Power Down mode. Once VS1>V_{POR}, the MM912F634 analog die enters Reset mode with the condition “Power On Reset - POR”.

4.3.2 Reset Mode

The MM912F634 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>V_{LVR} and VDDX>V_{LVRX} the MM912F634 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 < V_{LVR} or VDDX < V_{LVRX} with VS1 > (V_{LVRI}+ V_{LVR_H}) for more than t_{VTO}, the MM912F634 analog die transitions 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

4.4.1 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 (C_{VDD}) and (C_{VDDX}) are required for proper regulation.

4.4.2 Power Up Behavior / Power Down Behavior

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.

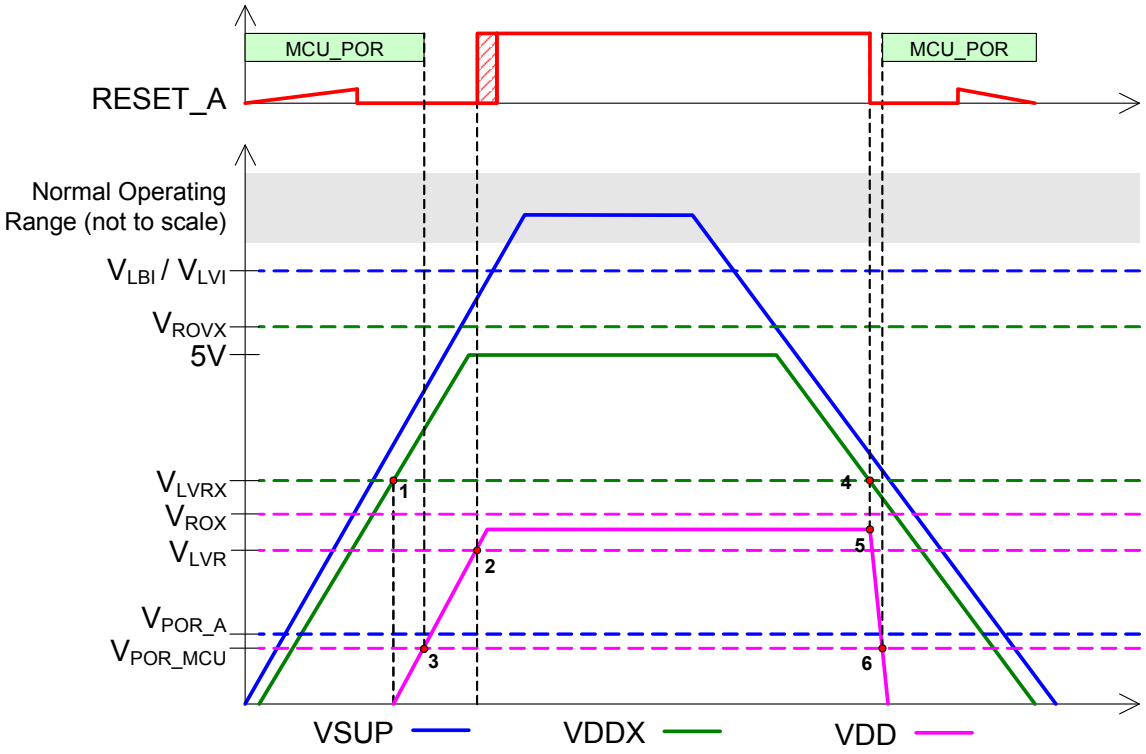


Figure 18. Power-up / Down Sequence

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 V_{DDX}, 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 VDD regulator is enabled after VDDX has reached the V_{LVRX} threshold (1).
- Once VDD reaches V_{LVR}, the $\overline{\text{RESET\_A}}$ is released (2).

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.

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 31 shows the output waveform generated.

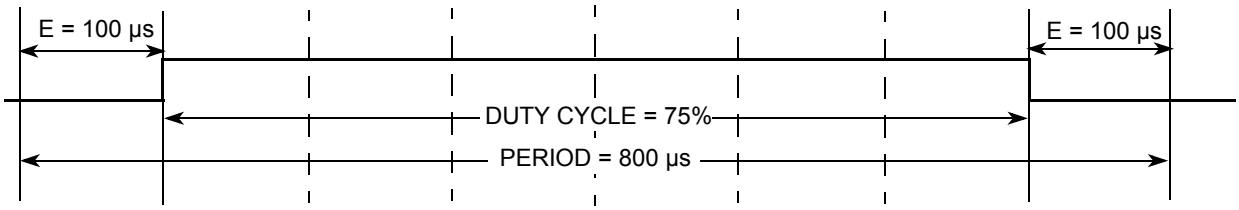


Figure 31. PWM Center Aligned Output Example Waveform

4.13.4.2.7 PWM Boundary Cases

Table 121 summarizes the boundary conditions for the PWM, regardless of the output mode (left aligned or center aligned).

Table 121. 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[97] (indicates no period)	1	Always high
XX	$00[97] (indicates no period)	0	Always low
>= PWMPERx	XX	1	Always high

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

4.13.5 Resets

The reset state of each individual bit is listed within the Section 4.13.3, “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.

Table 188. Analog Digital Converter Module - Memory Map (continued)

Register / Offset[135]		Bit 7	6	5	4	3	2	1	Bit 0
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								
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								

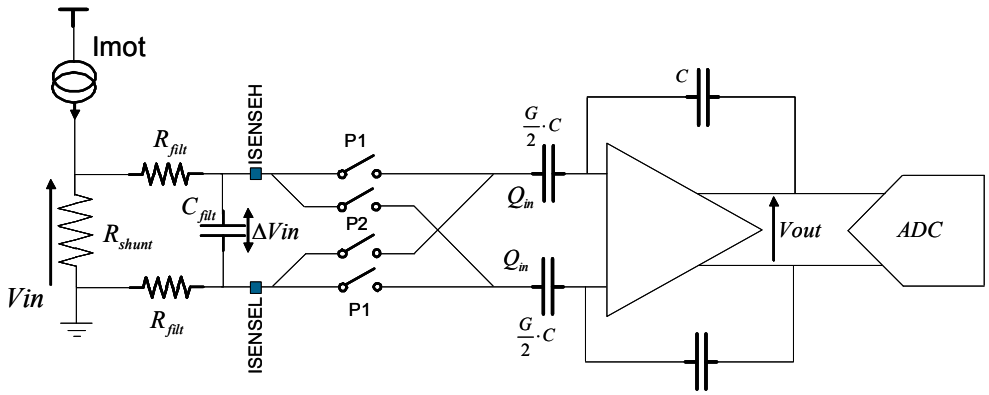


Figure 40. Current Sense Module with External Filter Option

The implementation is based on a switched capacitor solution to eliminate unwanted offset.To fit several application scenarios, eight different GAIN setting are implemented.

4.20.1 Register Definition

4.20.1.1 Current Sense Register (CSR)

Table 200. Current Sense Register (CSR)

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

Table 201. 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[143] 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:
143. 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.

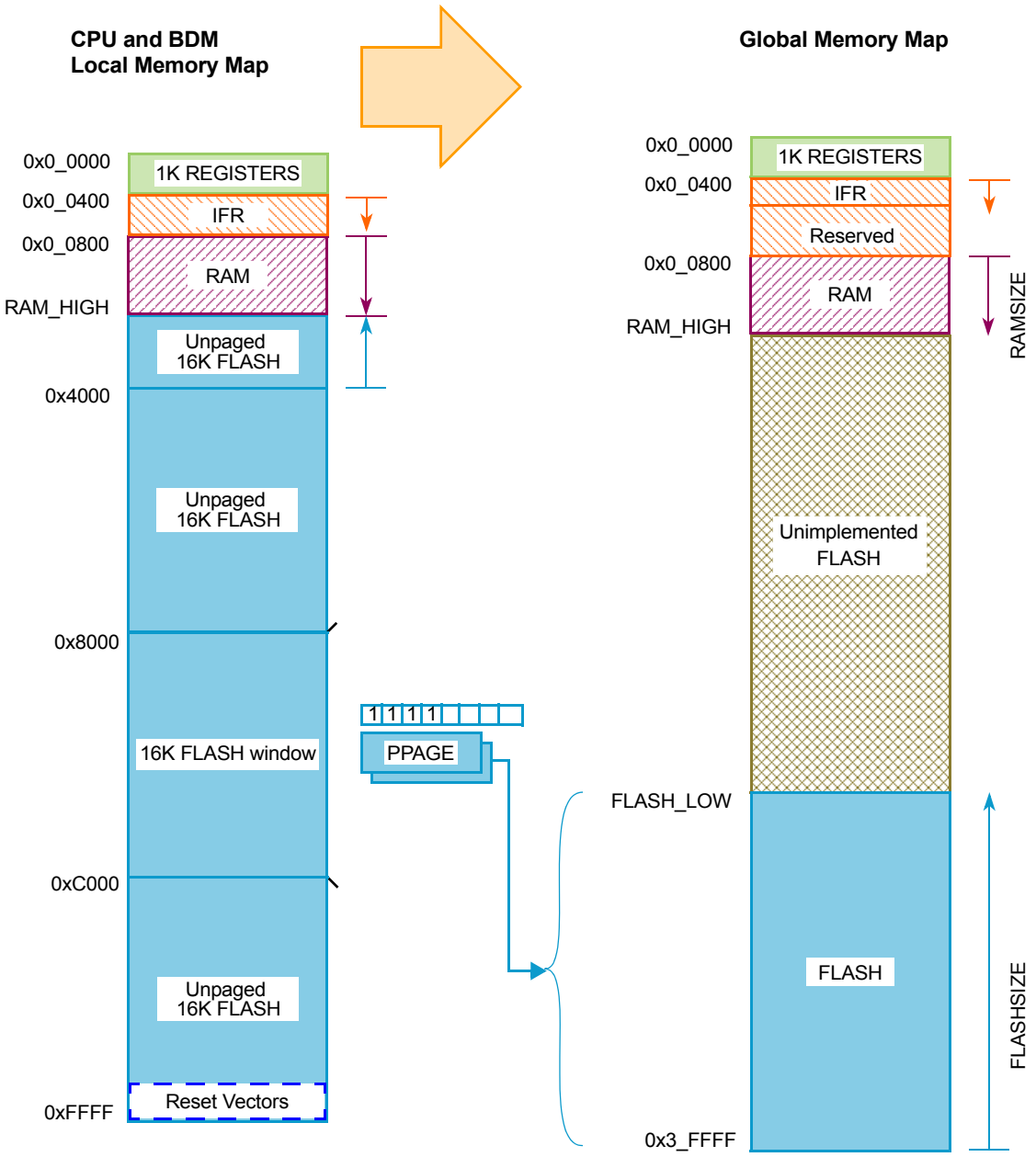


Figure 45. MC9S12I32 Global Address Mapping

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 wait or stop while the host issues a hardware command (e.g., WRITE_BYTE), the target discards the incoming command due to the wait or 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 or wait) 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.

4.30.4.8 Hardware Handshake Abort Procedure

The abort procedure is based on the SYNC command. In order to abort a command, which has 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 4.30.4.9, “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 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 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.

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.

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.

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 4.30.4.9, “SYNC — Request Timed Reference Pulse".

Figure 65 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 65 does not represent the signals in a true timing scale

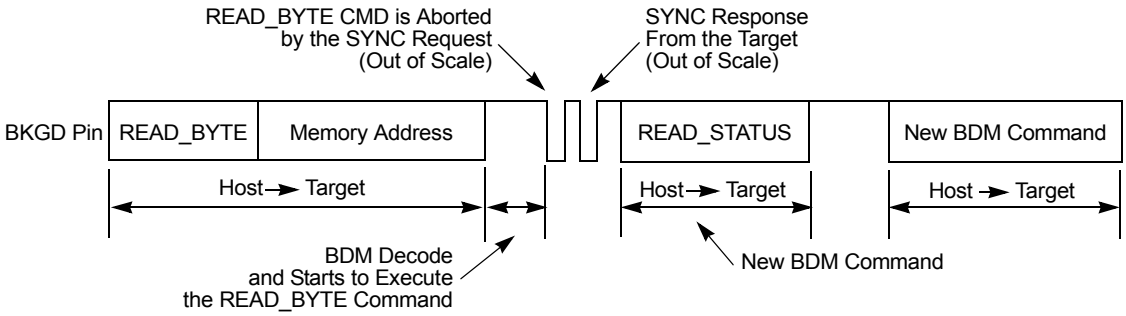


Figure 65. ACK Abort Procedure at the Command Level

Table 261. 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

4.31.1.5 Block Diagram

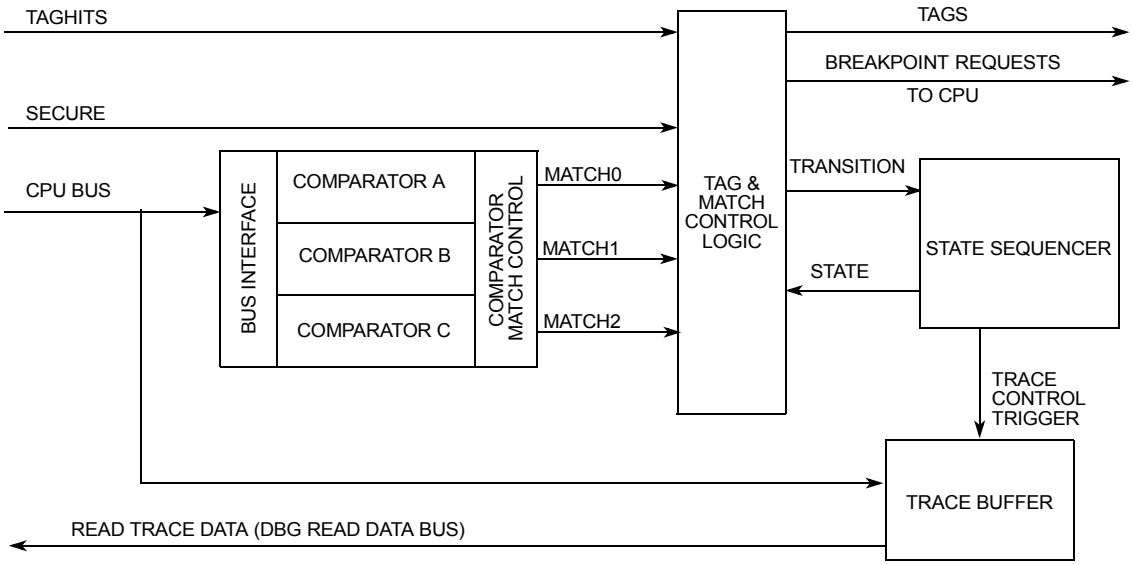


Figure 67. Debug Module Block Diagram

4.31.2 External Signal Description

There are no external signals associated with this module.

4.31.3 Memory Map and Registers

4.31.3.1 Module Memory Map

A summary of the registers associated with the DBG sub-block is shown in Table 262. Detailed descriptions of the registers and bits are given in the following subsections.

Table 262. 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[181]	0	0	0	0	SSF2	SSF1	SSF0
		W								

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 67 and described in Section 4.31.3.2.8.1, "Debug Comparator Control Register (DBGXCTL)". Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 288. DBGSCR3 Field Descriptions

Field	Description
2–0 SC[2:0]	These bits select the targeted next state whilst in State3, based upon the match event.

Table 289. State3 — Sequencer Next State Selection

SC[2:0]	Description
000	Match0 to State1
001	Match2 to State2..... Match1 to Final State
010	Match0 to Final State.....Match1 to State1
011	Match1 to Final State..... Match2 to State1
100	Match1 to State2
101	Match1 to Final State
110	Match2 to State2..... Match0 to Final State
111	Match0 to Final State

The priorities described in Table 314 dictate in the case of simultaneous matches, the match on the lower channel number (0,1,2) has priority. The SC[2:0] encoding ensures a match leading to final state has priority over all other matches.

4.31.3.2.7.4 Debug Match Flag Register (DBGMFR)

Table 290. 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

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.

4.31.3.2.8 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).

4.32.3.2.1 9S12I32PIMV1 Control Register 0 (CRGCTL0)

Table 323. 9S12I32PIMV1 Control Register 0 (CRGCTL0)

0x0034

	7	6	5	4	3	2	1	0
R	OSCEN	RDIV[2:0]			BCLKS	REFS	OSC4MHZ	0
W								
Reset	0	0	0	0	0	0	0	0

Read: Anytime
Write: See individual bit descriptions.

Writing the CRGCTL0 register clears the LOCKST bit, but does not set the LOCKIF bit in the CRGFLG register.

Table 324. CRGCTL0 Field Descriptions

Field	Description
7 OSCEN	Oscillator Enable Bit 0 Oscillator Clock and Oscillator Monitor are disabled. 1 Oscillator Clock and Oscillator Monitor are enabled.
6, 5, 4 RDIV[2:0]	Reference Divider Bits These bits divide the Oscillator Clock down in frequency. Divided down frequency must be in the allowed range for f_{FLLREF}. See device electrical characteristics for details. 000 divide by 128 001 divide by 160 010 divide by 192 011 divide by 256 100 divide by 320 101 divide by 384 110 divide by 512 111 Reserved
3 BCLKS	Bus Clock Source Select Bit Writing BCLKS = 1 is only possible if oscillator startup flag is set (UPOSC = 1). BCLKS is cleared with disabling the Oscillator, which is either OSCEN = 0 or entering Stop Mode. 0 DCO Clock is selected as basis for the Bus Clock. 1 Oscillator Clock is selected as basis for the Bus Clock. DCO is disabled.
2 REFS	Reference Select Bit Writing REFS = 1 is only possible if oscillator startup flag is set (UPOSC = 1). REFS is cleared with disabling the Oscillator, which is either OSCEN = 0 or entering Stop Mode. 0 Internal Reference Clock is selected as FLL Reference Clock. 1 Divided down Oscillator Clock is selected as FLL Reference Clock.
1 OSC4MHZ	4.0 MHz Oscillator low pass filter select Bit The Oscillator contains a noise filter in its signal path from EXTAL/XTAL to chip internal Oscillator Clock. This is to improve high frequency noise immunity. Writing OSC4MHZ is only possible if OSCEN was zero before. 0 Oscillator uses noise filter with high bandwidth. To be used with crystals/resonators > 4.0 Mhz. 1 Oscillator uses noise filter with low bandwidth. To be used with crystals/resonators = 4.0 Mhz. Choosing a low bandwidth in case of a 4.0 MHz crystal/resonator further improves noise immunity at lower frequencies.

4.35.3.2.1 COP Control Register (COPCTL)

This register controls the COP (Computer Operating Properly) watchdog.

Table 345. COP Control Register (COPCTL)

0x003E

	7	6	5	4	3	2	1	0
R	WCOP	RSBCK	0	COPSWAI	COPRSTP	CR2	CR1	CR0
W			WRTMASK					
Reset[190]	see note	0	0	0	0	see note	see note	see note

Note:
190. Refer to Device User Guide (Section 4.35.4.1, “COP Configuration") for reset values of WCOP, CR2, CR1 and CR0.

Read: Anytime
Write:

1. RSBCK: anytime in special modes; write to “1” but not to “0” in all other modes
2. WCOP, CR2, CR1, CR0:
   - — Anytime in special modes
   - — Write once in all other modes
     - – 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.

The COP timeout period is restarted if one these two conditions are true:

1. Writing a non-zero value to CR[2:0] (anytime in special modes, once in all other modes) with WRTMASK = 0.
   or
2. Changing RSBCK bit from “0” to “1”.

Table 346. COPCTL Field Descriptions

Field	Description
7 WCOP	**Window COP Mode Bit** — When set, a write to the ARMCOP register must occur in the last 25% of the selected period. A write during the first 75% of the selected period resets the part. 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 timeout logic restarts and the user must wait until the next window before writing to ARMCOP. Table 347 shows the duration of this window for the seven available COP rates. 0 Normal COP operation 1 Window COP operation
6 RSBCK	COP and RTI Stop in Active BDM Mode Bit 0 Allows the COP and RTI to keep running in Active BDM mode. 1 Stops the COP and RTI counters whenever the part is in Active BDM mode.
5 WRTMASK	**Write Mask for WCOP and CR[2:0] Bit** — This write-only bit serves as a mask for the WCOP, CR[2:0], COPSWAI and COPRSTP bits while writing the COPCTL register. It is intended for BDM writing the RSBCK without touching the contents of WCOP, CR[2:0], COPSWAI, and COPRSTP. 0 Write of WCOP, CR[2:0], COPSWAI and COPRSTP has an effect with this write of COPCTL 1 Write of WCOP, CR[2:0], COPSWAI and COPRSTP has no effect with this write of COPCTL. (Does not count for “write once”)
4 COPSWAI	COP Stops in Wait mode bit Normal modes: Write once Special modes: Write anytime 0 COP continues in Wait mode. 1 COP stops and initializes the COP counter whenever the part enters Wait mode.

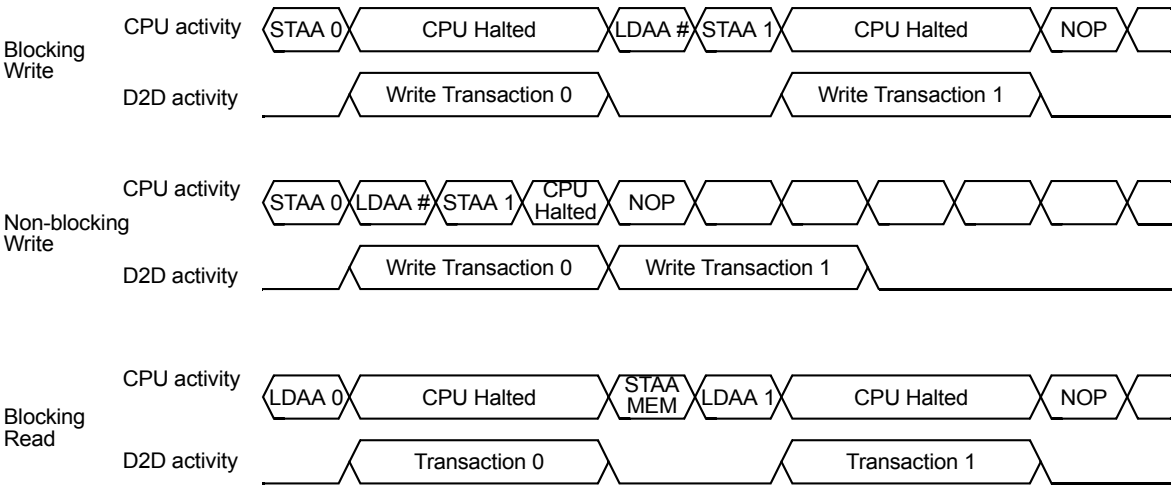


Figure 94. Blocking and Non-blocking Transfers.

4.37.4.2.1 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 94 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.

4.37.4.2.2 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 94 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 blocking transaction
LDAA    #BYTE1              ; load next byte
STAA    NONBLK_WINDOW+OFFS1; executed right after the first
NOP
```

As Figure 94 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.

4.37.4.2.3 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 94 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
```

4.37.4.2.4 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.

4.37.4.3 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 16-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.

4.37.4.4 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.

4.37.4.4.1 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 are set.

4.37.4.4.2 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 are set. The PAR[1:0] bits contain the parity value received by the D2DI.

4.37.4.4.3 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 are set.

4.37.4.5 Low Power Mode Options

4.37.4.5.1 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.

4.37.4.5.2 D2DI in Wait Mode

D2DI operation in wait mode depends upon the state of the D2DSWAI bit in D2D control register 0.

- If D2DSWAI is clear, the D2DI operates normally when the CPU is in the wait mode
- If D2DSWAI is set and the CPU enters the wait mode, any pending transmission is completed. When the D2DCLK output is driven low, the clock generation is stopped, all internal clocks to the D2DI module are stopped, and the module enters a power saving state.

4.37.4.5.3 D2DI in Stop Mode

If the CPU enters the STOP mode, the D2DI shows the same behavior as with the wait mode with an activated D2DSWAI bit.

4.37.4.6 Reset

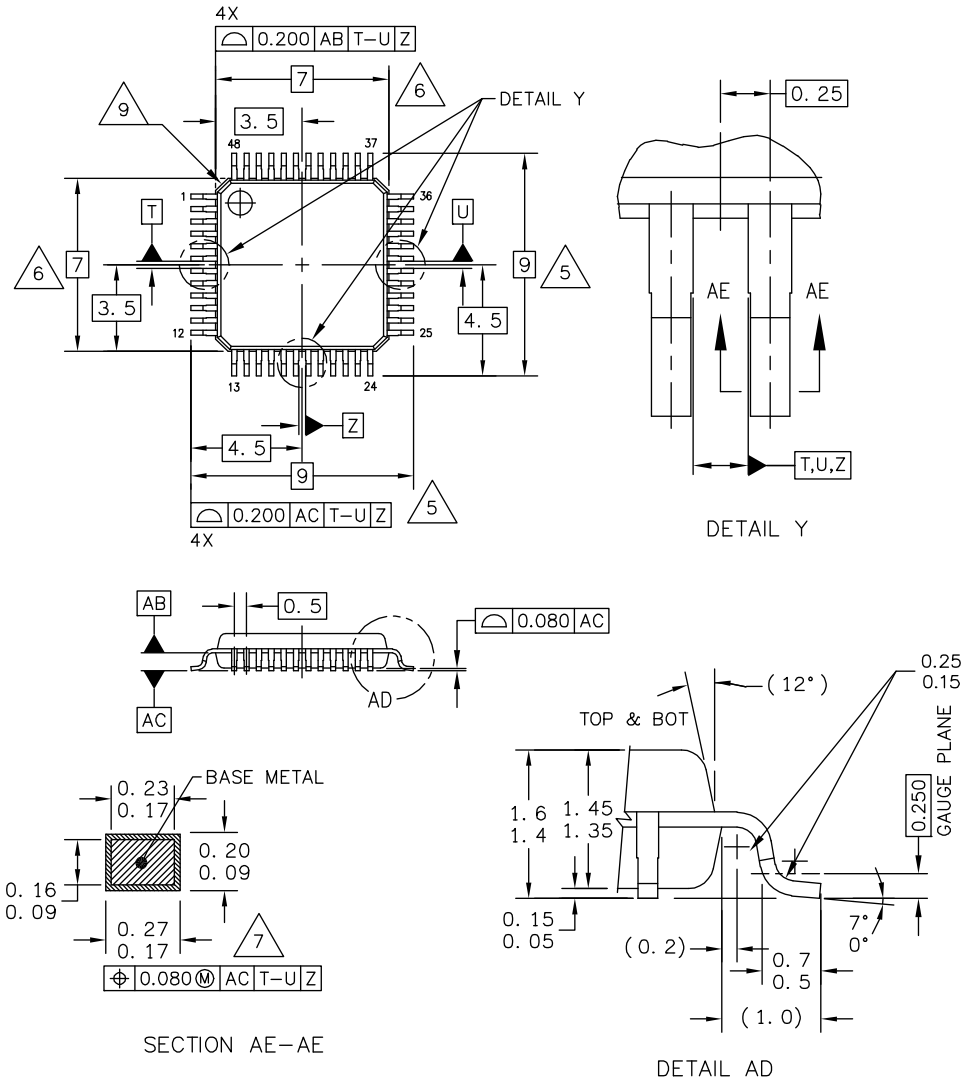
In case of reset, any transaction is immediately stopped and the D2DI module is disabled.

4.38.1.3 Modes of Operation

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

- Run mode
  This is the basic mode of operation.
- Wait mode
  SPI operation in wait mode is a configurable low-power mode, controlled by the SPISWAI bit located in the SPICR2 register. In wait mode, if the SPISWAI bit is clear, the SPI operates like in run mode. If the SPISWAI bit is set, the SPI goes into a power conservative state, with the SPI clock generation turned off. 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 a byte continues, so the slave stays synchronized to the master.
- 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 a byte continues, so the slave stays synchronized to the master.

This is a high level description only, detailed descriptions of operating modes are contained in Section 4.38.4.7, “Low Power Mode Options".



© 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	