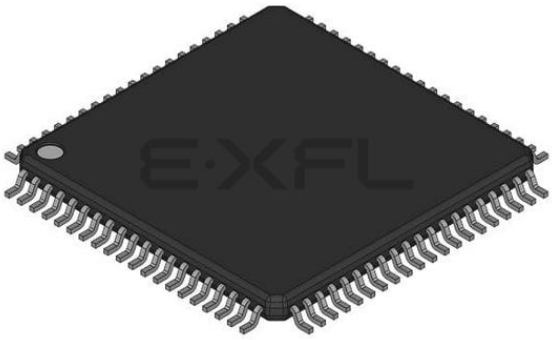


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	HCS12
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	64
Program Memory Size	96KB (96K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	6K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 10x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=s9s12p96j0mqk

1.7.4.6 VSSPLL — Ground Pin for PLL

This pin provides ground for the oscillator and the phased-locked loop. The voltage supply of nominally 1.8V is derived from the internal voltage regulator.

1.7.4.7 Power and Ground Connection Summary

Table 1-9. Power and Ground Connection Summary

Mnemonic	Nominal Voltage	Description
VDDR	5.0 V	External power supply to internal voltage regulator
VDDX[2:1]	5.0 V	External power and ground, supply to pin drivers
VSSX[2:1]	0 V	
VDDA	5.0 V	Operating voltage and ground for the analog-to-digital converters and the reference for the internal voltage regulator, allows the supply voltage to the A/D to be bypassed independently.
VSSA	0 V	
VRL	0 V	Reference voltages for the analog-to-digital converter.
VRH	5.0 V	
VSS3	0V	Internal power and ground generated by internal regulator for the internal core.
VSSPLL	0V	Provides operating voltage and ground for the phased-locked loop. This allows the supply voltage to the PLL to be bypassed independently. Internal power and ground generated by internal regulator.

1.8 System Clock Description

For the system clock description please refer to chapter Chapter 7, “S12 Clock, Reset and Power Management Unit (S12CPMU).

1.9 Modes of Operation

The MCU can operate in different modes. These are described in 1.9.1 Chip Configuration Summary.

The MCU can operate in different power modes to facilitate power saving when full system performance is not required. These are described in 1.9.2 Low Power Operation.

Some modules feature a software programmable option to freeze the module status whilst the background debug module is active to facilitate debugging.

Table 2-2. Block Memory Map (continued)

Port	Offset or Address	Register	Access	Reset Value	Section/Page
J	0x0268	PTJ—Port J Data Register	R/W	0x00	2.3.49/2-92
	0x0269	PTIJ—Port J Input Register	R	[4]	2.3.50/2-93
	0x026A	DDRJ—Port J Data Direction Register	R/W	0x00	2.3.51/2-93
	0x026B	RDRJ—Port J Reduced Drive Register	R/W	0x00	2.3.52/2-94
	0x026C	PERJ—Port J Pull Device Enable Register	R/W	0xFF	2.3.53/2-94
	0x026D	PPSJ—Port J Polarity Select Register	R/W	0x00	2.3.54/2-95
	0x026E	PIEJ—Port J Interrupt Enable Register	R/W	0x00	2.3.55/2-95
	0x026F	PIFJ—Port J Interrupt Flag Register	R/W	0x00	2.3.56/2-96
AD	0x0270	PT0AD—Port AD Data Register	R	0x00	2.3.57/2-96
	0x0271	PT1AD—Port AD Data Register	R/W	0x00	2.3.58/2-97
	0x0272	DDR0AD—Port AD Data Direction Register	R	0x00	2.3.59/2-97
	0x0273	DDR1AD—Port AD Data Direction Register	R/W	0x00	2.3.60/2-98
	0x0274	RDR0AD—Port AD Reduced Drive Register	R	0x00	2.3.61/2-98
	0x0275	RDR1AD—Port AD Reduced Drive Register	R/W	0x00	2.3.62/2-99
	0x0276	PER0AD—Port AD Pull Up Enable Register	R	0x00	2.3.62/2-99
	0x0277	PER1AD—Port AD Pull Up Enable Register	R/W	0x00	2.3.64/2-100
	0x0278 : 0x027F	PIM Reserved	R	0x00	2.3.65/2-100

1. Write access not applicable for one or more register bits. Refer to register description.
2. Refer to device memory map to determine related module.
3. Mode dependent.
4. Read always returns logic level on pins.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 PORTA	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
	W								
0x0001 PORTB	R	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
	W								
0x0002 DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
	W								
0x0003 DDRB	R	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
	W								

= Unimplemented or Reserved

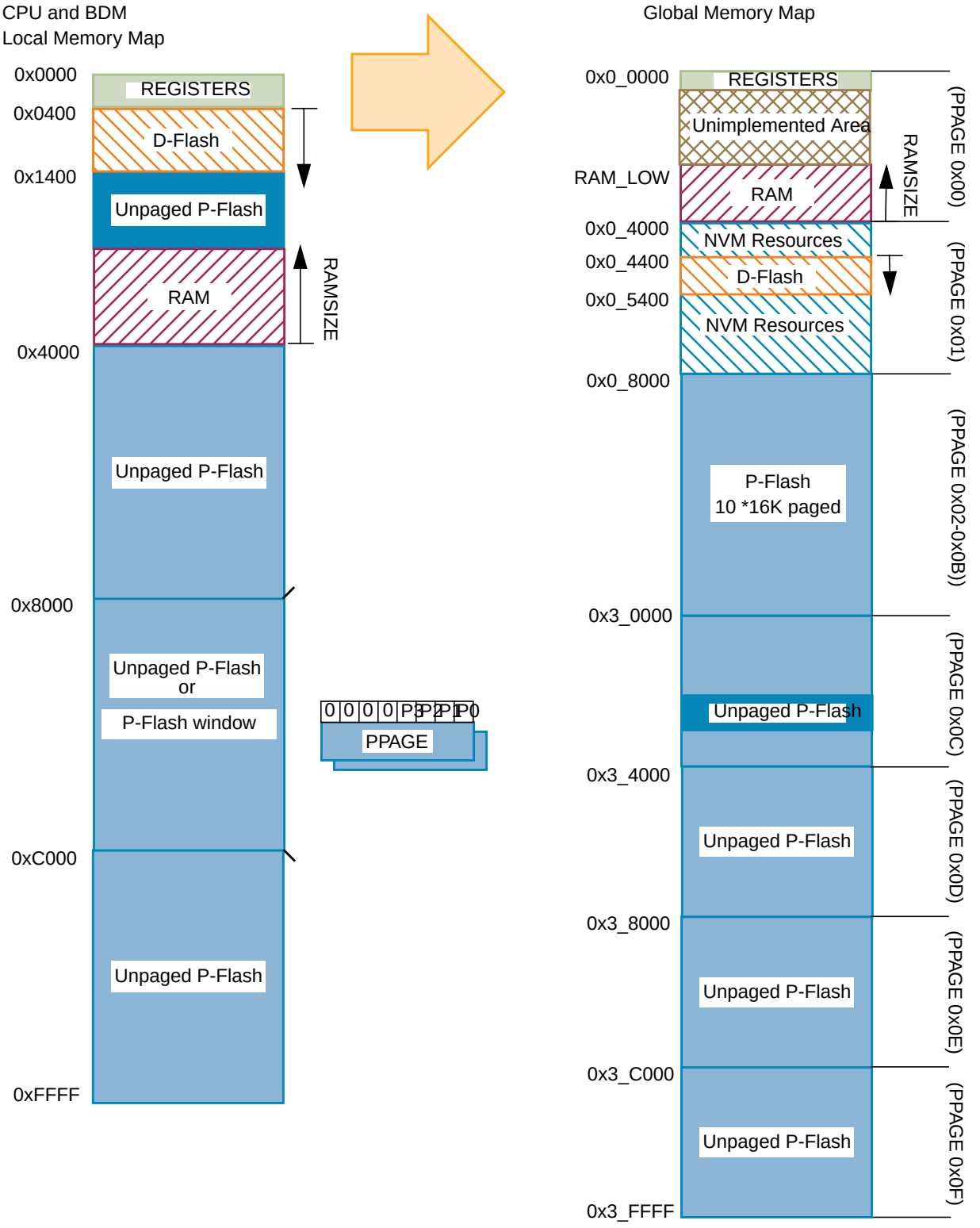


Figure 3-10. Local to Global Address Mapping

5.3.3 Family ID Assignment

The family ID is an 8-bit value located in the BDM ROM in active BDM (at global address: 0x3_FF0F). The read-only value is a unique family ID which is 0xC2 for devices with an HCS12S core.

5.4 Functional Description

The BDM receives and executes commands from a host via a single wire serial interface. There are two types of BDM commands: hardware and firmware commands.

Hardware commands are used to read and write target system memory locations and to enter active background debug mode, see Section 5.4.3, “BDM Hardware Commands”. Target system memory includes all memory that is accessible by the CPU.

Firmware commands are used to read and write CPU resources and to exit from active background debug mode, see Section 5.4.4, “Standard BDM Firmware Commands”. The CPU resources referred to are the accumulator (D), X index register (X), Y index register (Y), stack pointer (SP), and program counter (PC).

Hardware commands can be executed at any time and in any mode excluding a few exceptions as highlighted (see Section 5.4.3, “BDM Hardware Commands”) and in secure mode (see Section 5.4.1, “Security”). BDM firmware commands can only be executed when the system is not secure and is in active background debug mode (BDM).

5.4.1 Security

If the user resets into special single chip mode with the system secured, a secured mode BDM firmware lookup table is brought into the map overlapping a portion of the standard BDM firmware lookup table. The secure BDM firmware verifies that the on-chip Flash EEPROM are erased. This being the case, the UNSEC and ENBDM bit will get set. The BDM program jumps to the start of the standard BDM firmware and the secured mode BDM firmware is turned off and all BDM commands are allowed. If the Flash does not verify as erased, the BDM firmware sets the ENBDM bit, without asserting UNSEC, and the firmware enters a loop. This causes the BDM hardware commands to become enabled, but does not enable the firmware commands. This allows the BDM hardware to be used to erase the Flash.

BDM operation is not possible in any other mode than special single chip mode when the device is secured. The device can only be unsecured via BDM serial interface in special single chip mode. For more information regarding security, please see the S12S_9SEC Block Guide.

5.4.2 Enabling and Activating BDM

The system must be in active BDM to execute standard BDM firmware commands. BDM can be activated only after being enabled. BDM is enabled by setting the ENBDM bit in the BDM status (BDMSTS) register. The ENBDM bit is set by writing to the BDM status (BDMSTS) register, via the single-wire interface, using a hardware command such as WRITE_BD_BYTE.

After being enabled, BDM is activated by one of the following[1]:

- Hardware BACKGROUND command

1. BDM is enabled and active immediately out of special single-chip reset.

- CPU BGND instruction
- Breakpoint force or tag mechanism[1]

When BDM is activated, the CPU finishes executing the current instruction and then begins executing the firmware in the standard BDM firmware lookup table. When BDM is activated by a breakpoint, the type of breakpoint used determines if BDM becomes active before or after execution of the next instruction.

NOTE

If an attempt is made to activate BDM before being enabled, the CPU resumes normal instruction execution after a brief delay. If BDM is not enabled, any hardware BACKGROUND commands issued are ignored by the BDM and the CPU is not delayed.

In active BDM, the BDM registers and standard BDM firmware lookup table are mapped to addresses 0x3_FF00 to 0x3_FFFF. BDM registers are mapped to addresses 0x3_FF00 to 0x3_FF0B. The BDM uses these registers which are readable anytime by the BDM. However, these registers are not readable by user programs.

When BDM is activated while CPU executes code overlapping with BDM firmware space the saved program counter (PC) will be auto incremented by one from the BDM firmware, no matter what caused the entry into BDM active mode (BGND instruction, BACKGROUND command or breakpoints). In such a case the PC must be set to the next valid address via a WRITE_PC command before executing the GO command.

5.4.3 BDM Hardware Commands

Hardware commands are used to read and write target system memory locations and to enter active background debug mode. Target system memory includes all memory that is accessible by the CPU such as on-chip RAM, Flash, I/O and control registers.

Hardware commands are executed with minimal or no CPU intervention and do not require the system to be in active BDM for execution, although, they can still be executed in this mode. When executing a hardware command, the BDM sub-block waits for a free bus cycle so that the background access does not disturb the running application program. If a free cycle is not found within 128 clock cycles, the CPU is momentarily frozen so that the BDM can steal a cycle. When the BDM finds a free cycle, the operation does not intrude on normal CPU operation provided that it can be completed in a single cycle. However, if an operation requires multiple cycles the CPU is frozen until the operation is complete, even though the BDM found a free cycle.

The BDM hardware commands are listed in Table 5-4.

The READ_BD and WRITE_BD commands allow access to the BDM register locations. These locations are not normally in the system memory map but share addresses with the application in memory. To distinguish between physical memory locations that share the same address, BDM memory resources are

1. This method is provided by the S12S_DBG module.

Table 6-30. DBGADHM Field Descriptions

Field	Description
7–0 Bits[15:8]	**Comparator Data High Mask Bits** — The Comparator data high mask bits control whether the selected comparator compares the data bus bits [15:8] to the corresponding comparator data compare bits. Data bus comparisons are only performed if the TAG bit in DBGACTL is clear 0 Do not compare corresponding data bit Any value of corresponding data bit allows match. 1 Compare corresponding data bit

6.3.2.8.8 Debug Comparator Data Low Mask Register (DBGADLM)

Address: 0x002F

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

Figure 6-22. Debug Comparator Data Low Mask Register (DBGADLM)

Read: If COMRV[1:0] = 00

Write: If COMRV[1:0] = 00 and DBG not armed.

Table 6-31. DBGADLM Field Descriptions

Field	Description
7–0 Bits[7:0]	**Comparator Data Low Mask Bits** — The Comparator data low mask bits control whether the selected comparator compares the data bus bits [7:0] to the corresponding comparator data compare bits. Data bus comparisons are only performed if the TAG bit in DBGACTL is clear 0 Do not compare corresponding data bit. Any value of corresponding data bit allows match 1 Compare corresponding data bit

6.4 Functional Description

This section provides a complete functional description of the DBG module. If the part is in secure mode, the DBG module can generate breakpoints but tracing is not possible.

6.4.1 S12SDBG Operation

Arming the DBG module by setting ARM in DBGC1 allows triggering the state sequencer, storing of data in the trace buffer and generation of breakpoints to the CPU. The DBG module is made up of four main blocks, the comparators, control logic, the state sequencer, and the trace buffer.

The comparators monitor the bus activity of the CPU. All comparators can be configured to monitor address bus activity. Comparator A can also be configured to monitor databus activity and mask out individual data bus bits during a compare. Comparators can be configured to use R/W and word/byte access qualification in the comparison. A match with a comparator register value can initiate a state sequencer transition to another state (see Figure 6-24). Either forced or tagged matches are possible. Using

6.4.2.2.2 Outside Range (address < CompA_Addr or address > CompB_Addr)

In the Outside Range comparator mode, comparator pair A and B can be configured for range comparisons. A single match condition on either of the comparators is recognized as valid. An aligned word access which straddles the range boundary is valid only if the aligned address is outside the range.

Outside range mode in combination with tagging can be used to detect if the opcode fetches are from an unexpected range. In forced match mode the outside range match would typically be activated at any interrupt vector fetch or register access. This can be avoided by setting the upper range limit to $3FFFF or lower range limit to $00000 respectively.

6.4.3 Match Modes (Forced or Tagged)

Match modes are used as qualifiers for a state sequencer change of state. The Comparator control register TAG bits select the match mode. The modes are described in the following sections.

6.4.3.1 Forced Match

When configured for forced matching, a comparator channel match can immediately initiate a transition to the next state sequencer state whereby the corresponding flags in DBGSR are set. The state control register for the current state determines the next state. Forced matches are typically generated 2-3 bus cycles after the final matching address bus cycle, independent of comparator RWE/RW settings. Furthermore since opcode fetches occur several cycles before the opcode execution a forced match of an opcode address typically precedes a tagged match at the same address.

6.4.3.2 Tagged Match

If a CPU taghit occurs a transition to another state sequencer state is initiated and the corresponding DBGSR flags are set. For a comparator related taghit to occur, the DBG must first attach tags to instructions as they are fetched from memory. When the tagged instruction reaches the execution stage of the instruction queue a taghit is generated by the CPU. This can initiate a state sequencer transition.

6.4.3.3 Immediate Trigger

Independent of comparator matches it is possible to initiate a tracing session and/or breakpoint by writing to the TRIG bit in DBGC1. If configured for begin aligned tracing, this triggers the state sequencer into the Final State, if configured for end alignment, setting the TRIG bit disarms the module, ending the session and issues a forced breakpoint request to the CPU.

It is possible to set both TRIG and ARM simultaneously to generate an immediate trigger, independent of the current state of ARM.

6.4.3.4 Channel Priorities

In case of simultaneous matches the priority is resolved according to Table 6-36. The lower priority is suppressed. It is thus possible to miss a lower priority match if it occurs simultaneously with a higher priority. The priorities described in Table 6-36 dictate that in the case of simultaneous matches, the match pointing to final state has highest priority followed by the lower channel number (0,1,2).

6.5.10 Scenario 9

Trigger when a routine/event at A (M2) does not follow either B or C (M1 or M0) before they are executed again. This cannot be realized with theS12SDBGV1 SCR encoding due to OR limitations. By changing the SCR2 encoding as shown in red this scenario becomes possible.

Figure 6-39. Scenario 9

6.5.11 Scenario 10

Trigger if an event M0 occurs following up to two successive M2 events without the resetting event M1. As shown up to 2 consecutive M2 events are allowed, whereby a reset to State1 is possible after either one or two M2 events. If an event M0 occurs following the second M2, before M1 resets to State1 then a trigger is generated. Configuring CompA and CompC the same, it is possible to generate a breakpoint on the third consecutive occurrence of event M0 without a reset M1.

Figure 6-40. Scenario 10a

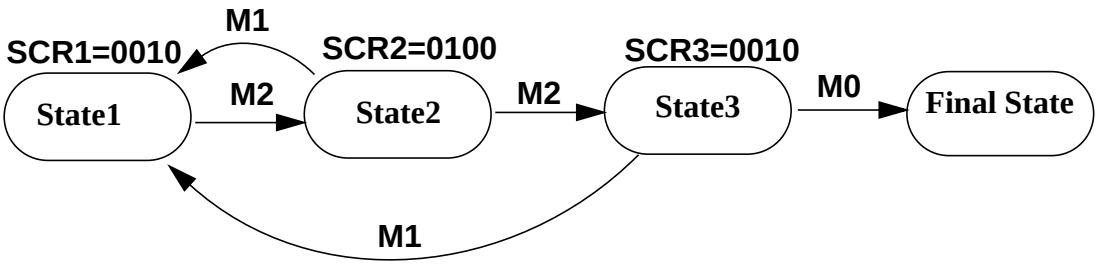
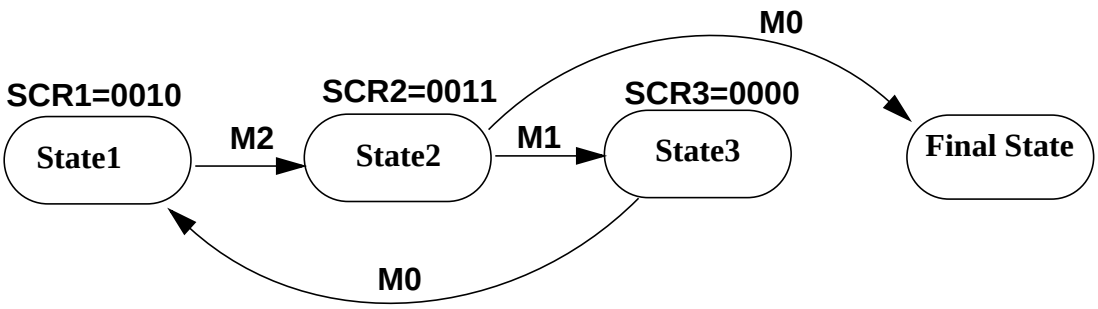


Figure 6-41. Scenario 10b



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.

Write: Refer to each bit for individual write conditions

Table 7-3. CPMUFLG Field Descriptions

Field	Description
7 RTIF	**Real Time Interrupt Flag** — RTIF is set to 1 at the end of the RTI period. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (RTIE=1), RTIF causes an interrupt request. 0 RTI time-out has not yet occurred. 1 RTI time-out has occurred.
6 PORF	**Power on Reset Flag** — PORF is set to 1 when a power on reset occurs. This flag can only be cleared by writing a 1. Writing a 0 has no effect. 0 Power on reset has not occurred. 1 Power on reset has occurred.
5 LVRF	**Low Voltage Reset Flag** — LVRF is set to 1 when a low voltage reset occurs. This flag can only be cleared by writing a 1. Writing a 0 has no effect. 0 Low voltage reset has not occurred. 1 Low voltage reset has occurred.
4 LOCKIF	**PLL Lock Interrupt Flag** — LOCKIF is set to 1 when LOCK status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect.If enabled (LOCKIE=1), LOCKIF causes an interrupt request. 0 No change in LOCK bit. 1 LOCK bit has changed.
3 LOCK	**Lock Status Bit** — LOCK reflects the current state of PLL lock condition. Writes have no effect. While PLL is unlocked (LOCK=0) f_{PLL} is f_{VCO} / 4 to protect the system from high core clock frequencies during the PLL stabilization time tlock. 0 VCOCLK is not within the desired tolerance of the target frequency. $f_{PLL} = f_{VCO}/4$. 1 VCOCLK is within the desired tolerance of the target frequency. $f_{PLL} = f_{VCO}/(POSTDIV+1)$.
2 ILAF	**Illegal Address Reset Flag** — ILAF is set to 1 when an illegal address reset occurs. Refer to MMC chapter for details. This flag can only be cleared by writing a 1. Writing a 0 has no effect. 0 Illegal address reset has not occurred. 1 Illegal address reset has occurred.
1 OSCIF	**Oscillator Interrupt Flag** — OSCIF is set to 1 when UPOSC status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect.If enabled (OSCIE=1), OSCIF causes an interrupt request. 0 No change in UPOSC bit. 1 UPOSC bit has changed.
0 UPOSC	**Oscillator Status Bit** — UPOSC reflects the status of the oscillator. Writes have no effect. While UPOSC=0 the OSCCLK going to the MSCAN module is off. Entering Full Stop Mode UPOSC is cleared. 0 The oscillator is off or oscillation is not qualified by the PLL. 1 The oscillator is qualified by the PLL.

NOTE

The adaptive spike filter uses the VCO clock as a reference to continuously qualify the external oscillator clock. Because of this, the PLL is always active and a valid PLL configuration is required for the system to work properly. Furthermore, the adaptive spike filter is used to determine the status of the external oscillator (reflected in the UPOSC bit). Since this function also relies on the VCO clock, loosing PLL lock status (LOCK=0, except for entering Pseudo Stop Mode) means loosing the oscillator status information as well (UPOSC=0).

7.3.2.10 Reserved Register CPMUTEST0

NOTE

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in special mode can alter the S12CPMU's functionality.

0x003D

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-13. Reserved Register (CPMUTEST0)

Read: Anytime

Write: Only in special mode

7.3.2.11 Reserved Register CPMUTEST1

NOTE

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in special mode can alter the S12CPMU's functionality.

0x003E

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-14. Reserved Register (CPMUTEST1)

Read: Anytime

Write: Only in special mode

7.3.2.12 S12CPMU COP Timer Arm/Reset Register (CPMUARMCOP)

This register is used to restart the COP time-out period.

0x003F

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

Figure 7-15. S12CPMU CPMUARMCOP Register

Read: Always reads $00

Write: Anytime

When the COP is disabled (CR[2:0] = “000”) writing to this register has no effect.

When the COP is enabled by setting CR[2:0] nonzero, the following applies:

Writing any value other than $55 or $AA causes a COP reset. To restart the COP time-out period write $55 followed by a write of $AA. These writes do not need to occur back-to-back, but the sequence ($55, $AA) must be completed prior to COP end of time-out period to avoid a COP reset. Sequences of $55 writes are allowed. When the WCOP bit is set, $55 and $AA writes must be done in the last 25% of the selected time-out period; writing any value in the first 75% of the selected period will cause a COP reset.

7.3.2.13 High Temperature Control Register (CPMUHTCTL)

The CPMUHTCTL register configures the temperature sense features.

0x02F0

	7	6	5	4	3	2	1	0
R	0	0	VSEL	0	HTE	HTDS	HTIE	HTIF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Read: Anytime

Write: VSEL, HTE, HTIE and HTIF are write anytime, HTDS is read only

1. In this case, PHASE_SEG1 must be at least 2 time quanta (Tq).

Table 8-9. Time Segment 2 Values

TSEG22	TSEG21	TSEG20	Time Segment 2
0	0	0	1 Tq clock cycle(1)
0	0	1	2 Tq clock cycles
:	:	:	:
1	1	0	7 Tq clock cycles
1	1	1	8 Tq clock cycles

1. This setting is not valid. Please refer to Table 8-37 for valid settings.

Table 8-10. Time Segment 1 Values

TSEG13	TSEG12	TSEG11	TSEG10	Time segment 1
0	0	0	0	1 Tq clock cycle(1)
0	0	0	1	2 Tq clock cycles1
0	0	1	0	3 Tq clock cycles1
0	0	1	1	4 Tq clock cycles
:	:	:	:	:
1	1	1	0	15 Tq clock cycles
1	1	1	1	16 Tq clock cycles

1. This setting is not valid. Please refer to Table 8-37 for valid settings.

The bit time is determined by the oscillator frequency, the baud rate prescaler, and the number of time quanta (Tq) clock cycles per bit (as shown in Table 8-9 and Table 8-10).

Eqn. 8-1

$$\text{Bit Time} = \frac{(\text{Prescaler value})}{f_{CANCLK}} \bullet (1 + \text{TimeSegment1} + \text{TimeSegment2})$$

8.3.2.5 MSCAN Receiver Flag Register (CANRFLG)

A flag can be cleared only by software (writing a 1 to the corresponding bit position) when the condition which caused the setting is no longer valid. Every flag has an associated interrupt enable bit in the CANRIER register.

NOTE

The CANTBSEL register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK=1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 8-17. CANTBSEL Register Field Descriptions

Field	Description
2-0 TX[2:0]	**Transmit Buffer Select** — The lowest numbered bit places the respective transmit buffer in the CANTXFG register space (e.g., TX1 = 1 and TX0 = 1 selects transmit buffer TX0; TX1 = 1 and TX0 = 0 selects transmit buffer TX1). Read and write accesses to the selected transmit buffer will be blocked, if the corresponding TXEx bit is cleared and the buffer is scheduled for transmission (see Section 8.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)"). 0 The associated message buffer is deselected 1 The associated message buffer is selected, if lowest numbered bit

The following gives a short programming example of the usage of the CANTBSEL register:

To get the next available transmit buffer, application software must read the CANTFLG register and write this value back into the CANTBSEL register. In this example Tx buffers TX1 and TX2 are available. The value read from CANTFLG is therefore 0b0000_0110. When writing this value back to CANTBSEL, the Tx buffer TX1 is selected in the CANTXFG because the lowest numbered bit set to 1 is at bit position 1. Reading back this value out of CANTBSEL results in 0b0000_0010, because only the lowest numbered bit position set to 1 is presented. This mechanism eases the application software the selection of the next available Tx buffer.

- LDAA CANTFLG; value read is 0b0000_0110
- STAA CANTBSEL; value written is 0b0000_0110
- LDAA CANTBSEL; value read is 0b0000_0010

If all transmit message buffers are deselected, no accesses are allowed to the CANTXFG registers.

8.3.2.12 MSCAN Identifier Acceptance Control Register (CANIDAC)

The CANIDAC register is used for identifier acceptance control as described below.

Module Base + 0x000B — Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	0	0	IDAM1	IDAM0	0	IDHIT2	IDHIT1	IDHIT0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 8-15. MSCAN Identifier Acceptance Control Register (CANIDAC)

1. Read: Anytime
   Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1), except bits IDHITx, which are read-only

Module Base + 0x00011

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

Figure 10-20. PWM Channel Counter Registers (PWMCNT5)

Read: anytime

Write: anytime (any value written causes PWM counter to be reset to 0x0000).

10.3.2.13 PWM Channel Period Registers (PWMPERx)

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

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

- The effective period ends
- The counter is written (counter resets to 0x0000)
- The channel is disabled

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

NOTE

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

Reference Section 10.4.2.3, “PWM Period and Duty,” for more information.

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

- Left aligned output (CAEx = 0)
- PWMx period = channel clock period * PWMPERx center aligned output (CAEx = 1)
- PWMx period = channel clock period * (2 * PWMPERx)

For boundary case programming values, please refer to Section 10.4.2.8, “PWM Boundary Cases.”

due to the synchronization of PWMEx and the clock source. An exception to this is when channels are concatenated. Refer to Section 10.4.2.7, “PWM 16-Bit Functions,” for more detail.

NOTE

The first PWM cycle after enabling the channel can be irregular.

On the front end of the PWM timer, the clock is enabled to the PWM circuit by the PWMEx bit being high. There is an edge-synchronizing circuit to guarantee that the clock will only be enabled or disabled at an edge. When the channel is disabled (PWMEx = 0), the counter for the channel does not count.

10.4.2.2 PWM Polarity

Each channel has a polarity bit to allow starting a waveform cycle with a high or low signal. This is shown on the block diagram as a mux select of either the Q output or the $\overline{Q}$ output of the PWM output flip-flop. When one of the bits in the PWMPOL register is set, the associated PWM channel output is high at the beginning of the waveform, then goes low when the duty count is reached. Conversely, if the polarity bit is 0, the output starts low and then goes high when the duty count is reached.

10.4.2.3 PWM Period and Duty

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

- The effective period ends
- The counter is written (counter resets to 0x0000)
- The channel is disabled

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

A change in duty or period can be forced into effect “immediately” by writing the new value to the duty and/or period registers and then writing to the counter. This forces the counter to reset and the new duty and/or period values to be latched. In addition, because the counter is readable it is possible to know where the count is with respect to the duty value and software can be used to make adjustments.

NOTE

When forcing a new period or duty into effect immediately, an irregular PWM cycle can occur.

Depending on the polarity bit, the duty registers will contain the count of either the high time or the low time.

10.4.2.4 PWM Timer Counters

Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source (reference Figure 10-34 for the available clock sources and rates). The counter compares to two registers, a duty register and a period register as shown in Figure 10-35. When the PWM counter matches the duty register the output flip-flop changes state causing the PWM waveform to also change state. A match

To determine the value of a data bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 11-18 summarizes the results of the data bit samples.

Table 11-18. Data Bit Recovery

RT8, RT9, and RT10 Samples	Data Bit Determination	Noise Flag
000	0	0
001	0	1
010	0	1
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

NOTE

The RT8, RT9, and RT10 samples do not affect start bit verification. If any or all of the RT8, RT9, and RT10 start bit samples are logic 1s following a successful start bit verification, the noise flag (NF) is set and the receiver assumes that the bit is a start bit (logic 0).

To verify a stop bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 11-19 summarizes the results of the stop bit samples.

Table 11-19. Stop Bit Recovery

RT8, RT9, and RT10 Samples	Framing Error Flag	Noise Flag
000	1	0
001	1	1
010	1	1
011	0	1
100	1	1
101	0	1
110	0	1
111	0	0

In Figure 11-22 the verification samples RT3 and RT5 determine that the first low detected was noise and not the beginning of a start bit. The RT clock is reset and the start bit search begins again. The noise flag is not set because the noise occurred before the start bit was found.

Table 13-4. Program IFR Fields

Global Address	Size (Bytes)	Field Description
0x0_40B8 – 0x0_40BF	8	Reserved
0x0_40C0 – 0x0_40FF	64	Program Once Field Refer to Section 13.4.5.6, “Program Once Command”

1. Used to track firmware patch versions, see Section 13.4.2

Table 13-5. D-Flash and Memory Controller Resource Fields

Global Address	Size (Bytes)	Description
0x0_4000 – 0x0_43FF	1,024	Reserved
0x0_4400 – 0x0_53FF	4,096	D-Flash Memory
0x0_5400 – 0x0_57FF	1,024	Reserved
0x0_5800 – 0x0_5AFF	768	Memory Controller Scratch RAM (RAMON(1) = 1)
0x0_5B00 – 0x0_5FFF	1,280	Reserved
0x0_6000 – 0x0_67FF	2,048	Reserved
0x0_6800 – 0x0_7FFF	6,144	Reserved

1. MMCCTL1 register bit

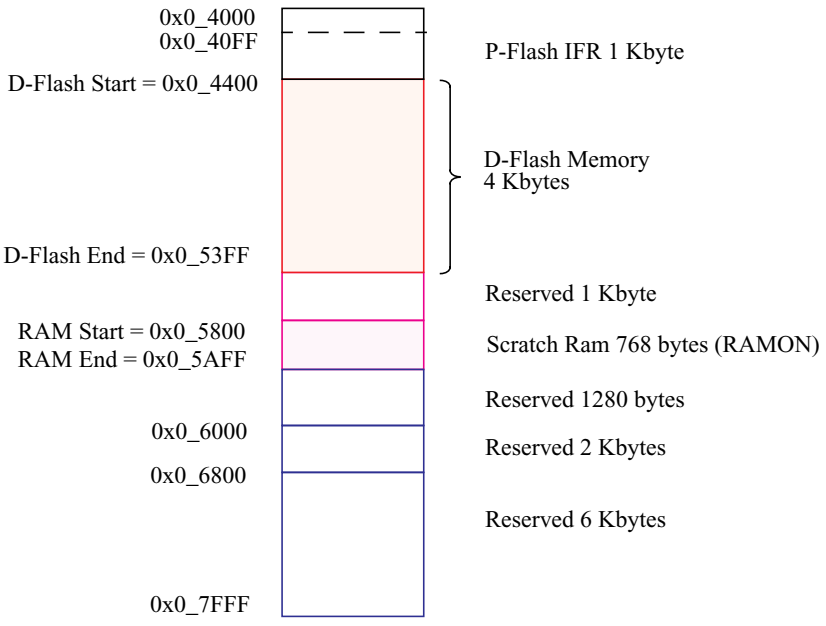


Figure 13-3. D-Flash and Memory Controller Resource Memory Map

Table 13-39. Program P-Flash Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters
011	Word 1 program value
100	Word 2 program value
101	Word 3 program value

1. Global address [2:0] must be 000

Upon clearing CCIF to launch the Program P-Flash command, the Memory Controller will program the data words to the supplied global address and will then proceed to verify the data words read back as expected. The CCIF flag will set after the Program P-Flash operation has completed.

Table 13-40. Program P-Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 101 at command launch
		Set if command not available in current mode (see Table 13-27)
		Set if an invalid global address [17:0] is supplied
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
	FPVIOL	Set if the global address [17:0] points to a protected area
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

13.4.5.6 Program Once Command

The Program Once command restricts programming to a reserved 64 byte field (8 phrases) in the nonvolatile information register located in P-Flash. The Program Once reserved field can be read using the Read Once command as described in Section 13.4.5.4. The Program Once command must only be issued once since the nonvolatile information register in P-Flash cannot be erased. The Program Once command must not be executed from the Flash block containing the Program Once reserved field to avoid code runaway.

Table 13-41. Program Once Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x07	Not Required
001	Program Once phrase index (0x0000 - 0x0007)	
010	Program Once word 0 value	
011	Program Once word 1 value	
100	Program Once word 2 value	
101	Program Once word 3 value	

A.2.2 Factors Influencing Accuracy

Source resistance, source capacitance and current injection have an influence on the accuracy of the ATD. A further factor is that PortAD pins that are configured as output drivers switching.

A.2.2.1 Port AD Output Drivers Switching

PortAD output drivers switching can adversely affect the ATD accuracy whilst converting the analog voltage on other PortAD pins because the output drivers are supplied from the VDDA/VSSA ATD supply pins. Although internal design measures are implemented to minimize the affect of output driver noise, it is recommended to configure PortAD pins as outputs only for low frequency, low load outputs. The impact on ATD accuracy is load dependent and not specified. The values specified are valid under condition that no PortAD output drivers switch during conversion.

A.2.2.2 Source Resistance

Due to the input pin leakage current as specified in Table A-6 and Table A-7 in conjunction with the source resistance there will be a voltage drop from the signal source to the ATD input. The maximum source resistance R_S specifies results in an error (10-bit resolution) of less than 1/2 LSB (2.5 mV) at the maximum leakage current. If device or operating conditions are less than worst case or leakage-induced error is acceptable, larger values of source resistance of up to 10Kohm are allowed.

A.2.2.3 Source Capacitance

When sampling an additional internal capacitor is switched to the input. This can cause a voltage drop due to charge sharing with the external and the pin capacitance. For a maximum sampling error of the input voltage ≤ 1LSB (10-bit resilution), then the external filter capacitor, $C_f \geq 1024 * (C_{INS}–C_{INN})$.

A.2.2.4 Current Injection

There are two cases to consider.

1. A current is injected into the channel being converted. The channel being stressed has conversion values of $3FF (in 10-bit mode) for analog inputs greater than V_{RH} and $000 for values less than V_{RL} unless the current is higher than specified as disruptive condition.
2. Current is injected into pins in the neighborhood of the channel being converted. A portion of this current is picked up by the channel (coupling ratio K), This additional current impacts the accuracy of the conversion depending on the source resistance.

   The additional input voltage error on the converted channel can be calculated as:

   $$V_{ERR} = K * R_S * I_{INJ}$$

   with I_{INJ} being the sum of the currents injected into the two pins adjacent to the converted channel.