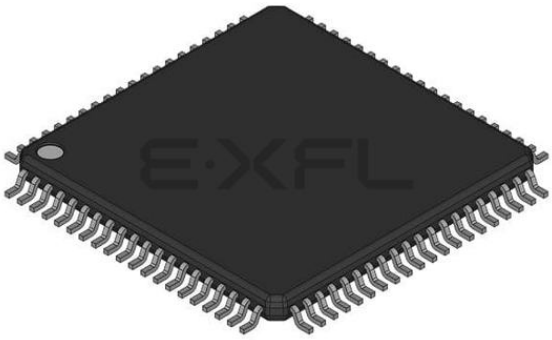

Welcome to E-XFL.COM

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	HCS12X
Core Size	16-Bit
Speed	80MHz
Connectivity	CANbus, I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	59
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 8x10b, 16x10b
Oscillator Type	External
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=s912xdg128f2maa

2.4.3.3.1 Wake-up from Pseudo Stop Mode (PSTP=1)

Wake-up from pseudo stop mode is the same as wake-up from wait mode. There are also four different scenarios for the CRG to restart the MCU from pseudo stop mode:

- External reset
- Clock monitor fail
- COP reset
- Wake-up interrupt

If the MCU gets an external reset or COP reset during pseudo stop mode active, the CRG asynchronously restores all configuration bits in the register space to its default settings and starts the reset generator. After completing the reset sequence processing begins by fetching the normal or COP reset vector. pseudo stop mode is left and the MCU is in run mode again.

If the clock monitor is enabled (CME = 1), the MCU is able to leave pseudo stop mode when loss of oscillator/external clock is detected by a clock monitor fail. If the SCME bit is not asserted the CRG generates a clock monitor fail reset (CMRESET). The CRG's behavior for CMRESET is the same compared to external reset, but another reset vector is fetched after completion of the reset sequence. If the SCME bit is asserted the CRG generates a SCM interrupt if enabled (SCMIE = 1). After generating the interrupt the CRG enters self-clock mode and starts the clock quality checker (Section 2.4.1.4, "Clock Quality Checker"). Then the MCU continues with normal operation. If the SCM interrupt is blocked by SCMIE=0, the SCMIF flag will be asserted but the CRG will not wake-up from pseudo stop mode.

If any other interrupt source (e.g., RTI) triggers exit from pseudo stop mode, the MCU immediately continues with normal operation. Because the PLL has been powered-down during stop mode, the PLLSEL bit is cleared and the MCU runs on OSCCLK after leaving stop mode. The software must set the PLLSEL bit again, in order to switch system and core clocks to the PLLCLK.

Table 2-13 summarizes the outcome of a clock loss while in pseudo stop mode.

4.3.2.4 ATD Control Register 3 (ATDCTL3)

This register controls the conversion sequence length, FIFO for results registers and behavior in Freeze Mode. Writes to this register will abort current conversion sequence but will not start a new sequence.

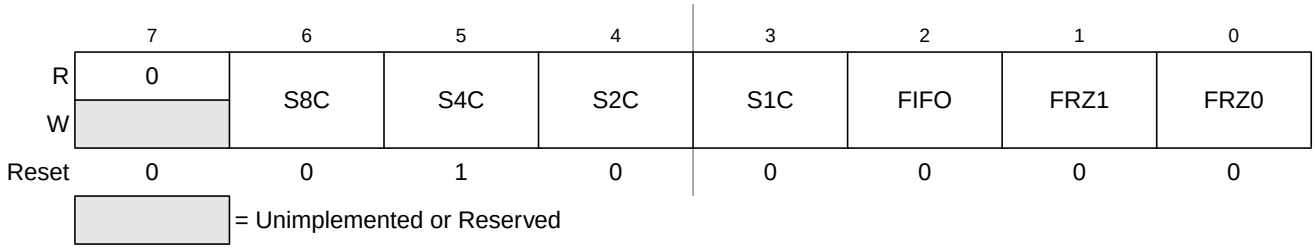

Figure 4-6. ATD Control Register 3 (ATDCTL3)

Read: Anytime

Write: Anytime

Table 4-8. ATDCTL3 Field Descriptions

Field	Description
6 S8C	**Conversion Sequence Length** — This bit controls the number of conversions per sequence. Table 4-9 shows all combinations. At reset, S4C is set to 1 (sequence length is 4). This is to maintain software continuity to HC12 Family.
5 S4C	**Conversion Sequence Length** — This bit controls the number of conversions per sequence. Table 4-9 shows all combinations. At reset, S4C is set to 1 (sequence length is 4). This is to maintain software continuity to HC12 Family.
4 S2C	**Conversion Sequence Length** — This bit controls the number of conversions per sequence. Table 4-9 shows all combinations. At reset, S4C is set to 1 (sequence length is 4). This is to maintain software continuity to HC12 Family.
3 S1C	**Conversion Sequence Length** — This bit controls the number of conversions per sequence. Table 4-9 shows all combinations. At reset, S4C is set to 1 (sequence length is 4). This is to maintain software continuity to HC12 Family.

6.3.1.12 XGATE Register 4 (XGR4)

The XGR4 register (Figure 6-15) provides access to the RISC core's register 4.

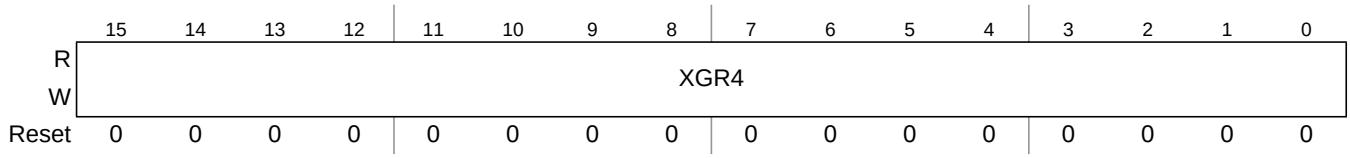

Figure 6-15. XGATE Register 4 (XGR4)

Read: In debug mode if unsecured

Write: In debug mode if unsecured

Table 6-12. XGR4 Field Descriptions

Field	Description
15–0 XGR4[15:0]	**XGATE Register 4** — The RISC core's register 4

6.3.1.13 XGATE Register 5 (XGR5)

The XGR5 register (Figure 6-16) provides access to the RISC core's register 5.

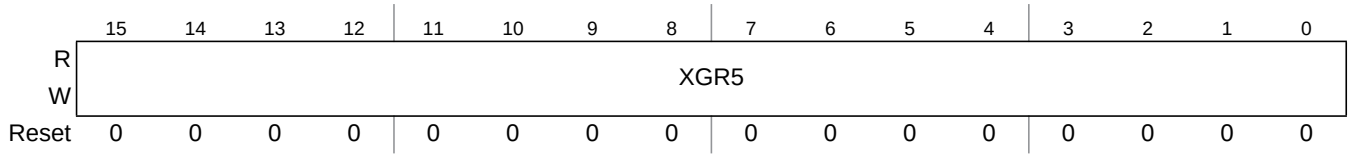

Figure 6-16. XGATE Register 5 (XGR5)

Read: In debug mode if unsecured

Write: In debug mode if unsecured

Table 6-13. XGR5 Field Descriptions

Field	Description
15–0 XGR5[15:0]	**XGATE Register 5** — The RISC core's register 5

BEQ Branch if Equal BEQ

Operation

If Z = 1, then PC + $0002 + (REL9 << 1) ⇒ PC

Tests the Zero flag and branches if Z = 1.

CCR Effects

N	Z	V	C
—	—	—	—

N: Not affected.
Z: Not affected.
V: Not affected.
C: Not affected.

Code and CPU Cycles

Source Form	Address Mode	Machine Code								Cycles
BEQ REL9	REL9	0	0	1	0	0	1	1	REL9	PP/P

BPL Branch if Plus BPL

Operation

If N = 0, then PC + $0002 + (REL9 << 1) ⇒ PC

Tests the Sign flag and branches if N = 0.

CCR Effects

N	Z	V	C
—	—	—	—

N: Not affected.
Z: Not affected.
V: Not affected.
C: Not affected.

Code and CPU Cycles

Source Form	Address Mode	Machine Code								Cycles
BPL REL9	REL9	0	0	1	0	1	0	0	REL9	PP/P

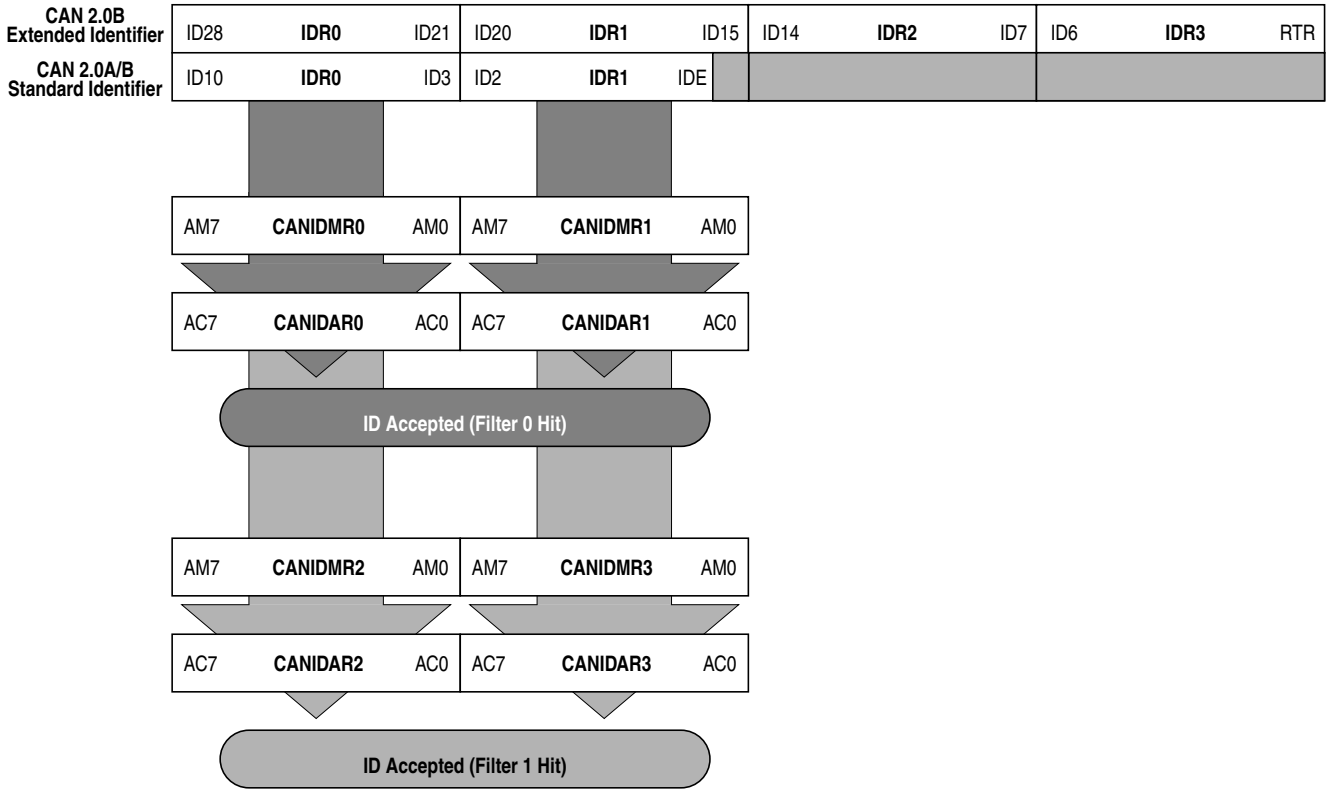

Figure 10-41. 16-bit Maskable Identifier Acceptance Filters

NOTE

The user is responsible for ensuring that the MSCAN is not active when initialization mode is entered. The recommended procedure is to bring the MSCAN into sleep mode (SLPRQ = 1 and SLPAK = 1) before setting the INITRQ bit in the CANCTL0 register. Otherwise, the abort of an on-going message can cause an error condition and can impact other CAN bus devices.

In initialization mode, the MSCAN is stopped. However, interface registers remain accessible. This mode is used to reset the CANCTL0, CANRFLG, CANRIER, CANTFLG, CANTIER, CANTARQ, CANTAAK, and CANTBSEL registers to their default values. In addition, the MSCAN enables the configuration of the CANBTR0, CANBTR1 bit timing registers; CANIDAC; and the CANIDAR, CANIDMR message filters. See Section 10.3.2.1, "MSCAN Control Register 0 (CANCTL0)," for a detailed description of the initialization mode.

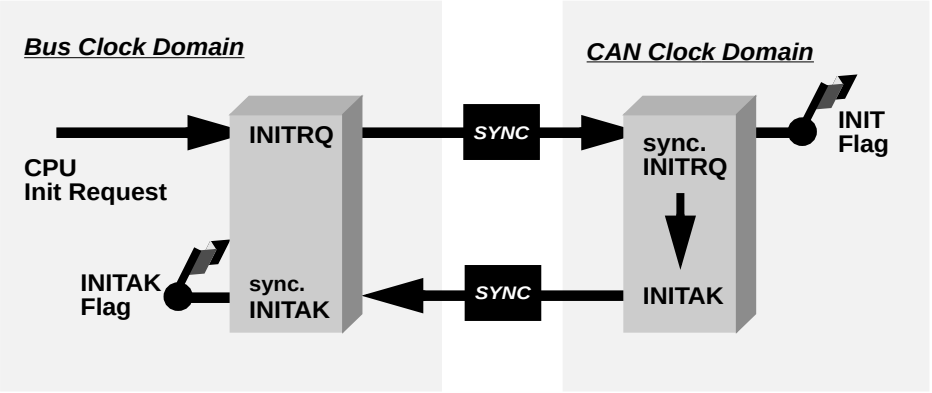

Figure 10-47. Initialization Request/Acknowledge Cycle

Due to independent clock domains within the MSCAN, INITRQ must be synchronized to all domains by using a special handshake mechanism. This handshake causes additional synchronization delay (see Section Figure 10-47., "Initialization Request/Acknowledge Cycle").

If there is no message transfer ongoing on the CAN bus, the minimum delay will be two additional bus clocks and three additional CAN clocks. When all parts of the MSCAN are in initialization mode, the INITAK flag is set. The application software must use INITAK as a handshake indication for the request (INITRQ) to go into initialization mode.

NOTE

The CPU cannot clear INITRQ before initialization mode (INITRQ = 1 and INITAK = 1) is active.

10.4.5.6 MSCAN Power Down Mode

The MSCAN is in power down mode (Table 10-36) when

- CPU is in stop mode

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
PITLD1 (Low)	R W	PLD7	PLD6	PLD5	PLD4	PLD3	PLD2	PLD1	PLD0
PITCNT1 (High)	R W	PCNT15	PCNT14	PCNT13	PCNT12	PCNT11	PCNT10	PCNT9	PCNT8
PITCNT1 (Low)	R W	PCNT7	PCNT6	PCNT5	PCNT4	PCNT3	PCNT2	PCNT1	PCNT0
PITLD2 (High)	R W	PLD15	PLD14	PLD13	PLD12	PLD11	PLD10	PLD9	PLD8
PITLD2 (Low)	R W	PLD7	PLD6	PLD5	PLD4	PLD3	PLD2	PLD1	PLD0
PITCNT2 (High)	R W	PCNT15	PCNT14	PCNT13	PCNT12	PCNT11	PCNT10	PCNT9	PCNT8
PITCNT2 (Low)	R W	PCNT7	PCNT6	PCNT5	PCNT4	PCNT3	PCNT2	PCNT1	PCNT0
PITLD3 (High)	R W	PLD15	PLD14	PLD13	PLD12	PLD11	PLD10	PLD9	PLD8
PITLD3 (Low)	R W	PLD7	PLD6	PLD5	PLD4	PLD3	PLD2	PLD1	PLD0
PITCNT3 (High)	R W	PCNT15	PCNT14	PCNT13	PCNT12	PCNT11	PCNT10	PCNT9	PCNT8
PITCNT3 (Low)	R W	PCNT7	PCNT6	PCNT5	PCNT4	PCNT3	PCNT2	PCNT1	PCNT0

= Unimplemented or Reserved

Figure 13-2. PIT Register Summary (Sheet 2 of 2)

17.4.2.1.1 Expansion of the Local Address Map

Expansion of the CPU Local Address Map

The program page index register in MMC allows accessing up to 4 Mbyte of FLASH or ROM in the global memory map by using the eight page index bits to page 256 16 Kbyte blocks into the program page window located from address $8000 to address $BFFF in the local CPU memory map.

The page value for the program page window is stored in the PPAGE register. The value of the PPAGE register can be read or written by normal memory accesses as well as by the CALL and RTC instructions (see Section 1.5.1, “CALL and RTC Instructions”).

Control registers, vector space and parts of the on-chip memories are located in unpaged portions of the 64-kilobyte local CPU address space.

The starting address of an interrupt service routine must be located in unpaged memory unless the user is certain that the PPAGE register will be set to the appropriate value when the service routine is called. However an interrupt service routine can call other routines that are in paged memory. The upper 16-kilobyte block of the local CPU memory space ($C000–$FFFF) is unpaged. It is recommended that all reset and interrupt vectors point to locations in this area or to the other upages sections of the local CPU memory map.

Table 1-19 summarizes mapping of the address bus in Flash/External space based on the address, the PPAGE register value and value of the ROMHM bit in the MMCCTL1 register.

Table 17-18. Global FLASH/ROM Allocated

Local CPU Address	ROMHM	External Access	Global Address
$4000–$7FFF	0	No	$7F_4000 –$7F_7FFF
	1	Yes	$14_4000–$14_7FFF
$8000–$BFFF	N/A	No[1]	$40_0000–$7F_FFFF
	N/A	Yes[1]	
$C000–$FFFF	N/A	No	$7F_C000–$7F_FFFF

1 The internal or the external bus is accessed based on the size of the memory resources implemented on-chip. Please refer to Figure 1-23 for further details.

The RAM page index register allows accessing up to 1 Mbyte –2 Kbytes of RAM in the global memory map by using the eight RPAGE index bits to page 4 Kbyte blocks into the RAM page window located in the local CPU memory space from address $1000 to address $1FFF. The EEPROM page index register EPAGE allows accessing up to 256 Kbytes of EEPROM in the system by using the eight EPAGE index bits to page 1 Kbyte blocks into the EEPROM page window located in the local CPU memory space from address $0800 to address $0BFF.

Expansion of the BDM Local Address Map

PPAGE, RPAGE, and EPAGE registers are also used for the expansion of the BDM local address to the global address. These registers can be read and written by the BDM.

The BDM expansion scheme is the same as the CPU expansion scheme.

17.4.2.2 Global Addresses Based on the Global Page

CPU Global Addresses Based on the Global Page

The seven global page index bits allow access to the full 8 Mbyte address map that can be accessed with 23 address bits. This provides an alternative way to access all of the various pages of FLASH, RAM and EEPROM as well as additional external memory.

The GPAGE Register is used only when the CPU is executing a global instruction (see Section 1.3.2.3, “Global Page Index Register (GPAGE)”). The generated global address is the result of concatenation of the CPU local address [15:0] with the GPAGE register [22:16] (see Figure 1-7).

BDM Global Addresses Based on the Global Page

The seven BDMGPR Global Page index bits allow access to the full 8 Mbyte address map that can be accessed with 23 address bits. This provides an alternative way to access all of the various pages of FLASH, RAM and EEPROM as well as additional external memory.

The BDM global page index register (BDMGPR) is used only in the case the CPU is executing a firmware command which uses a global instruction (like GLDD, GSTD) or by a BDM hardware command (like WRITE_W, WRITE_BYTE, READ_W, READ_BYTE). See the BDM Block Guide for further details.

The generated global address is a result of concatenation of the BDM local address with the BDMGPR register [22:16] in the case of a hardware command or concatenation of the CPU local address and the BDMGPR register [22:16] in the case of a firmware command (see Figure 1-22).

- Simultaneous accesses to different resources[1] (internal, external, and peripherals) (see)
- Resolution of target bus access collision
- Access restriction control from masters to some targets (e.g., RAM write access protection for user specified areas)
- MCU operation mode control
- MCU security control
- Separate memory map schemes for each master CPU, BDM and XGATE
- ROM control bits to enable the on-chip FLASH or ROM selection
- Port replacement registers access control
- Generation of system reset when CPU accesses an unimplemented address (i.e., an address which does not belong to any of the on-chip modules) in single-chip modes

18.1.3 S12X Memory Mapping

The S12X architecture implements a number of memory mapping schemes including

- a CPU 8 MByte global map, defined using a global page (GPAGE) register and dedicated 23-bit address load/store instructions.
- a BDM 8 MByte global map, defined using a global page (BDMGPR) register and dedicated 23-bit address load/store instructions.
- a (CPU or BDM) 64 KByte local map, defined using specific resource page (RPAGE, EPAGE and PPAGE) registers and the default instruction set. The 64 KBytes visible at any instant can be considered as the local map accessed by the 16-bit (CPU or BDM) address.
- The XGATE 64 Kbyte local map.

The MMC module performs translation of the different memory mapping schemes to the specific global (physical) memory implementation.

18.1.4 Modes of Operation

This subsection lists and briefly describes all operating modes supported by the MMC.

18.1.4.1 Power Saving Modes

- Run mode

  MMC is functional during normal run mode.
- Wait mode

  MMC is functional during wait mode.
- Stop mode

  MMC is inactive during stop mode.

1. Resources are also called targets.

Table 19-3. DBGC1 Field Descriptions

Field	Description
7 ARM	**Arm Bit** — The ARM bit controls whether the DBG module is armed. This bit can be set and cleared by user software and is automatically cleared on completion of a tracing session, or if a breakpoint is generated with tracing not enabled. On setting this bit the state sequencer enters State1. When ARM is set, the only bits in the DBG module registers that can be written are ARM and TRIG. 0 Debugger disarmed 1 Debugger armed
6 TRIG	**Immediate Trigger Request Bit** — This bit when written to 1 requests an immediate trigger independent of comparator or external tag signal status. When tracing is complete a forced breakpoint may be generated depending upon DBGBRK and BDM bit settings. This bit always reads back a “0”. Writing a “0” to this bit has no effect. If both TSOURCE bits are clear no tracing is carried out. If tracing has already commenced using BEGIN- or mid-trigger alignment, it continues until the end of the tracing session as defined by the TALIGN bit settings, thus TRIG has no affect. In secure mode tracing is disabled and writing to this bit has no effect. 0 Do not trigger until the state sequencer enters the final state. 1 Enter final state immediately and issue forced breakpoint request when trace buffer is full.
5 XGSBPE	**XGATE S/W Breakpoint Enable** — The XGSBPE bit controls whether an XGATE S/W breakpoint request is passed to the CPU. The XGATE S/W breakpoint request is handled by the DBG module, which can request an CPU breakpoint depending on the state of this bit. 0 XGATE S/W breakpoint request is disabled 1 XGATE S/W breakpoint request is enabled
4 BDM	**Background Debug Mode Enable** — This bit determines if a CPU breakpoint causes the system to enter background debug mode (BDM) or initiate a software interrupt (SWI). It has no affect on DBG functionality. This bit must be set if the BDM is enabled by the ENBDM bit in the BDM module to map breakpoints to BDM and must be cleared if the BDM module is disabled to map breakpoints to SWI. 0 Go to software interrupt on a breakpoint 1 Go to BDM on a breakpoint.
3–2 DBGBRK	**DBG Breakpoint Enable Bits** — The DBGBRK bits control whether the debugger will request a breakpoint to either CPU, XGATE or both upon reaching the state sequencer final state. If tracing is enabled, the breakpoint is generated on completion of the tracing session. If tracing is not enabled, the breakpoint is generated immediately. Please refer to Section 19.4.7, “Breakpoints” for further details. XGATE generated breakpoints are independent of the DBGBRK bits. XGATE generates a forced breakpoint to the CPU only. See Table 19-4.
1–0 COMRV	**Comparator Register Visibility Bits** — These bits determine which bank of comparator register is visible in the 8-byte window of the DBG module address map, located between 0x0028 to 0x002F. Furthermore these bits determine which state control register is visible at the address 0x0027. See Table 19-5.

Table 19-4. DBGBRK Encoding

DBGBRK	Resource Halted by Breakpoint
00	No breakpoint generated
01	XGATE breakpoint generated
10	CPU breakpoint generated
11	Breakpoints generated for CPU and XGATE

Table 19-5. COMRV Encoding

COMRV	Visible Comparator	Visible State Control Register
00	Comparator A	DBGSCR1

19.4.4 State Sequence Control

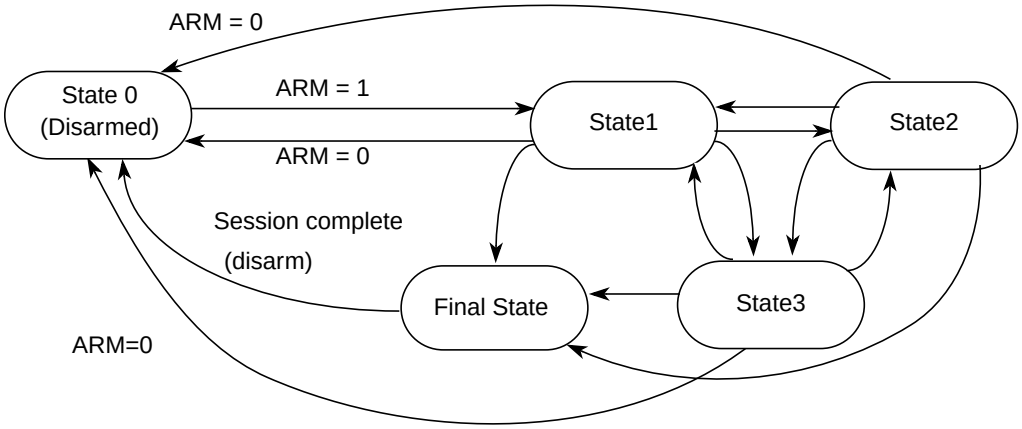

Figure 19-23. State Sequencer Diagram

The state sequence control allows a defined sequence of events to provide a trigger point for tracing of data in the trace buffer. Once the DBG module has been armed by setting the ARM bit in the DBGC1 register, then State1 of the state sequencer is entered. Further transitions between the states are then controlled by the state control registers and depend upon a selected trigger mode condition being met. From final state the only permitted transition is back to the disarmed state0. Transition between any of the states 1 to 3 is not restricted. Each transition updates the SSF[2:0] flags in DBGSR accordingly to indicate the current state.

Alternatively writing to the TRIG bit in DBGSC1, the final state is entered and tracing starts immediately if the TSOURCE bits are configured for tracing.

A tag hit through $\overline{\text{TAGHI}}/\overline{\text{TAGLO}}$ causes a breakpoint, if breakpoints are enabled, and ends tracing immediately independent of the trigger alignment bits TALIGN[1:0].

Furthermore, each comparator channel can be individually configured to generate an immediate breakpoint when a match occurs through the use of the BRK bits in the DBGxCTL registers independent of the state sequencer state. Thus it is possible to generate an immediate breakpoint on selected channels, while a state sequencer transition can be initiated by a match on other channels.

An XGATE S/W breakpoint request, if enabled causes a transition to the final state and generates a breakpoint request to the CPU immediately.

If neither tracing nor breakpoints are enabled then, when a forced match triggers to final state, it can only be returned to the disarmed state0 by clearing the ARM bit by software. This also applies to the case that BDM breakpoints are enabled, but the BDM is disabled. Furthermore if neither tracing nor breakpoints are enabled, forced triggers on channels with BRK set cause a transition to the state determined by the state sequencer as if the BRK bit were not being used.

If neither tracing nor breakpoints are enabled then when a tagged match triggers to final state, the state sequencer returns to the disarmed state0.

Table 22-1. Pin Functions and Priorities (Sheet 3 of 7)

Port	Pin Name	Pin Function and Priority	I/O	Description	Pin Function after Reset
K	PK[7]	ROMCTL[1]	I	ROMON bit control input during $\overline{\text{RESET}}$	Mode dependent[3]
		$\overline{\text{EWAIT}}$	I	External Wait signal Configurable for reduced input threshold	
		GPIO	I/O	General-purpose I/O	
	PK[6:4]	ADDR[22:20] mux ACC[2:0][2]	O	Extended external bus address output (multiplexed with access master output)	
		GPIO	I/O	General-purpose I/O	
	PK[3:0]	ADDR[19:16] mux IQSTAT[3:0][2]	O	Extended external bus address output (multiplexed with instruction pipe status bits)	
		GPIO	I/O	General-purpose I/O	
T	PT[7:0]	IOC[7:0]	I/O	Enhanced Capture Timer Channels 7–0 input/output	GPIO
		GPIO	I/O	General-purpose I/O	
S	PS7	$\overline{\text{SS0}}$	I/O	Serial Peripheral Interface 0 slave select output in master mode, input in slave mode or master mode.	GPIO
		GPIO	I/O	General-purpose I/O	
	PS6	SCK0	I/O	Serial Peripheral Interface 0 serial clock pin	
		GPIO	I/O	General-purpose I/O	
	PS5	MOSI0	I/O	Serial Peripheral Interface 0 master out/slave in pin	
		GPIO	I/O	General-purpose I/O	
	PS4	MISO0	I/O	Serial Peripheral Interface 0 master in/slave out pin	
		GPIO	I/O	General-purpose I/O	
	PS3	TXD1	O	Serial Communication Interface 1 transmit pin	
		GPIO	I/O	General-purpose I/O	
	PS2	RXD1	I	Serial Communication Interface 1 receive pin	
		GPIO	I/O	General-purpose I/O	
	PS1	TXD0	O	Serial Communication Interface 0 transmit pin	
		GPIO	I/O	General-purpose I/O	
	PS0	RXD0	I	Serial Communication Interface 0 receive pin	
		GPIO	I/O	General-purpose I/O	

22.4.1.8 Interrupt Enable Register

If the pin is used as an interrupt input this register serves as a mask to the interrupt flag to enable/disable the interrupt.

22.4.1.9 Interrupt Flag Register

If the pin is used as an interrupt input this register holds the interrupt flag after a valid pin event.

22.4.1.10 Module Routing Register

This register supports the re-routing of the CAN0, CAN4, SPI0, SPI1, and SPI2 pins to alternative ports. This allows a software re-configuration of the pinouts of the different package options with respect to above peripherals.

NOTE

The purpose of the module routing register is to provide maximum flexibility for derivatives with a lower number of MSCAN and SPI modules.

Table 22-68. Module Implementations on Derivatives

Number of Modules	MSCAN Modules					SPI Modules		
	CAN0	CAN1	CAN2	CAN3	CAN4	SPI0	SPI1	SPI2
5	yes	yes	yes	yes	yes	—	—	—
4	yes	yes	yes	—	yes	—	—	—
3	yes	yes	—	—	yes	yes	yes	yes
2	yes	—	—	—	yes	yes	yes	—
1	yes	—	—	—	—	yes	—	—

22.4.2 Ports

22.4.2.1 BKGD Pin

The BKGD pin is associated with the S12X_BDM and S12X_EBI modules. During reset, the BKGD pin is used as MODC input.

22.4.2.2 Port A and B

Port A pins PA[7:0] and Port B pins PB[7:0] can be used for either general-purpose I/O, or, in 144-pin packages, also with the external bus interface. In this case port A and port B are associated with the external address bus outputs ADDR15–ADDR8 and ADDR7–ADDR0, respectively. PB0 is the ADDR0 or $\overline{\text{UDS}}$ output.

Table 22-70. Expanded Bus Pin Functions versus Operating Modes (continued)

Pin	Single-Chip Modes		Expanded Modes			
	Normal Single-Chip	Special Single-Chip	Normal Expanded	Emulation Single-Chip	Emulation Expanded	Special Test
PE6	GPIO	GPIO	GPIO	TAGHI	TAGHI	GPIO
PE5	GPIO	GPIO	RE	TAGLO	TAGLO	GPIO
PE4	GPIO *or* *ECLK*	ECLK *or* *GPIO*	ECLK *or* *GPIO*	ECLK	ECLK	ECLK *or* *GPIO*
PE3	GPIO	GPIO	$\overline{\text{LDS}}$ *or* *GPIO*	$\overline{\text{LSTRB}}$	$\overline{\text{LSTRB}}$	$\overline{\text{LSTRB}}$
PE2	GPIO	GPIO	$\overline{\text{WE}}$	R/$\overline{\text{W}}$	R/$\overline{\text{W}}$	R/$\overline{\text{W}}$
PJ5	GPIO	GPIO	GPIO *or* $\overline{\text{CS2}}$	GPIO	GPIO *or* $\overline{\text{CS2}}$	GPIO *or* $\overline{\text{CS2}}$
PJ4	GPIO	GPIO	GPIO *or* $\overline{\text{CS0}}$ (1)	GPIO	GPIO *or* $\overline{\text{CS0}}$ (1)	GPIO *or* $\overline{\text{CS0}}$
PJ2	GPIO	GPIO	GPIO *or* $\overline{\text{CS1}}$	GPIO	GPIO *or* $\overline{\text{CS1}}$	GPIO *or* $\overline{\text{CS1}}$
PJ0	GPIO	GPIO	GPIO *or* $\overline{\text{CS3}}$	GPIO	GPIO *or* $\overline{\text{CS3}}$	GPIO *or* $\overline{\text{CS3}}$

1 Depending on ROMON bit. *Refer to Device Guide, S12X_EBI section and S12X_MMC section for details.*

22.4.5 Low-Power Options

22.4.5.1 Run Mode

No low-power options exist for this module in run mode.

22.4.5.2 Wait Mode

No low-power options exist for this module in wait mode.

22.4.5.3 Stop Mode

All clocks are stopped. There are asynchronous paths to generate interrupts from stop on port P, H, and J.

22.5 Initialization and Application Information

- It is not recommended to write PORTx and DDRx in a word access. When changing the register pins from inputs to outputs, the data may have extra transitions during the write access. Initialize the port data register before enabling the outputs.

A.2 ATD Characteristics

This section describes the characteristics of the analog-to-digital converter.

A.2.1 ATD Operating Characteristics

The Table A-12 and Table A-13 show conditions under which the ATD operates.

The following constraints exist to obtain full-scale, full range results:

$$V_{SSA} \le V_{RL} \le V_{IN} \le V_{RH} \le V_{DDA}.$$

This constraint exists since the sample buffer amplifier can not drive beyond the power supply levels that it ties to. If the input level goes outside of this range it will effectively be clipped.

Table A-12. ATD 5-V Operating Characteristics

Conditions are shown in Table A-4 unless otherwise noted, supply voltage 4.5 V < V_{DDA} < 5.5 V							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	D	Reference potential Low High	 V_{RL} V_{RH}	 V_{SSA} $V_{DDA}/2$	 — —	 $V_{DDA}/2$ V_{DDA}	 V V
2	C	Differential reference voltage[1]	V_{RH}-V_{RL}	4.50	5.00	5.5	V
3	D	ATD clock frequency	f_{ATDCLK}	0.5		2.0	MHz
4	D	ATD 10-bit conversion period Clock cycles[2] Conv, time at 2.0 MHz ATD clock f_{ATDCLK}	 N_{CONV10} T_{CONV10}	 14 7	 — —	 28 14	 Cycles µs
5	D	ATD 8-Bit conversion period Clock cycles[2] Conv, time at 2.0 MHz ATD clock f_{ATDCLK}	 N_{CONV8} T_{CONV8}	 12 6	 — —	 26 13	 Cycles µs
6	D	Recovery time (V_{DDA} = 5.0 Volts)	t_{REC}	—	—	20	µs
7	P	Reference supply current 2 ATD blocks on	I_{REF}	—	—	0.750	mA
8	P	Reference supply current 1 ATD block on	I_{REF}	—	—	0.375	mA

1 Full accuracy is not guaranteed when differential voltage is less than 4.50 V

2 The minimum time assumes a final sample period of 2 ATD clocks cycles while the maximum time assumes a final sample period of 16 ATD clocks.

0x00130–0x0137 Asynchronous Serial Interface (SCI4) Map

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0130	SCI4BDH[1]	R	IREN	TNP1	TNP0	SBR12	SBR11	SBR10	SBR9	SBR8
		W								
0x0131	SCI4BDL[1]	R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
		W								
0x0132	SCI4CR1[1]	R	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
		W								
0x0130	SCI4ASR1[2]	R	RXEDGIF	0	0	0	0	BERRV	BERRIF	BKDIF
		W								
0x0131	SCI4ACR1[2]	R	RXEDGIE	0	0	0	0	0	BERRIE	BKDIE
		W								
0x0132	SCI4ACR2[2]	R	0	0	0	0	0	BERRM1	BERRM0	BKDFE
		W								
0x0133	SCI4CR2	R	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
		W								
0x0134	SCI4SR1	R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
		W								
0x0135	SCI4SR2	R	AMAP	0	0	TXPOL	RXPOL	BRK13	TXDIR	RAF
		W								
0x0136	SCI4DRH	R	R8	T8	0	0	0	0	0	0
		W								
0x0137	SCI4DRL	R	R7	R6	R5	R4	R3	R2	R1	R0
		W	T7	T6	T5	T4	T3	T2	T1	T0

1 Those registers are accessible if the AMAP bit in the SCI4SR2 register is set to zero

2 Those registers are accessible if the AMAP bit in the SCI4SR2 register is set to one

0x0138–0x013F Asynchronous Serial Interface (SCI5) Map

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0138	SCI5BDH[1]	R	IREN	TNP1	TNP0	SBR12	SBR11	SBR10	SBR9	SBR8
		W								
0x0139	SCI5BDL[1]	R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
		W								
0x013A	SCI5CR1[1]	R	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
		W								
0x0138	SCI5ASR1[2]	R	RXEDGIF	0	0	0	0	BERRV	BERRIF	BKDIF
		W								
0x0139	SCI5ACR1[2]	R	RXEDGIE	0	0	0	0	0	BERRIE	BKDIE
		W								
0x013A	SCI5ACR2[2]	R	0	0	0	0	0	BERRM1	BERRM0	BKDFE
		W								
0x013B	SCI5CR2	R	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
		W								
0x013C	SCI5SR1	R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
		W								

0x0240–0x027F Port Integration Module PIM_9DX (PIM) Map (Sheet 2 of 4)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0248	PTS	R W	PTS7	PTS6	PTS5	PTS4	PTS3	PTS2	PTS1	PTS0
0x0249	PTIS	R W	PTIS7	PTIS6	PTIS5	PTIS4	PTIS3	PTIS2	PTIS1	PTIS0
0x024A	DDRS	R W	DDRS7	DDRS7	DDRS5	DDRS4	DDRS3	DDRS2	DDRS1	DDRS0
0x024B	RDRS	R W	RDRS7	RDRS6	RDRS5	RDRS4	RDRS3	RDRS2	RDRS1	RDRS0
0x024C	PERS	R W	PERS7	PERS6	PERS5	PERS4	PERS3	PERS2	PERS1	PERS0
0x024D	PPSS	R W	PPSS7	PPSS6	PPSS5	PPSS4	PPSS3	PPSS2	PPSS1	PPSS0
0x024E	WOMS	R W	WOMS7	WOMS6	WOMS5	WOMS4	WOMS3	WOMS2	WOMS1	WOMS0
0x024F	Reserved	R W	0	0	0	0	0	0	0	0
0x0250	PTM	R W	PTM7	PTM6	PTM5	PTM4	PTM3	PTM2	PTM1	PTM0
0x0251	PTIM	R W	PTIM7	PTIM6	PTIM5	PTIM4	PTIM3	PTIM2	PTIM1	PTIM0
0x0252	DDRM	R W	DDRM7	DDRM7	DDRM5	DDRM4	DDRM3	DDRM2	DDRM1	DDRM0
0x0253	RDRM	R W	RDRM7	RDRM6	RDRM5	RDRM4	RDRM3	RDRM2	RDRM1	RDRM0
0x0254	PERM	R W	PERM7	PERM6	PERM5	PERM4	PERM3	PERM2	PERM1	PERM0
0x0255	PPSM	R W	PPSM7	PPSM6	PPSM5	PPSM4	PPSM3	PPSM2	PPSM1	PPSM0
0x0256	WOMM	R W	WOMM7	WOMM6	WOMM5	WOMM4	WOMM3	WOMM2	WOMM1	WOMM0
0x0257	MODRR	R W	0	MODRR6	MODRR5	MODRR4	MODRR3	MODRR2	MODRR1	MODRR0
0x0258	PTP	R W	PTP7	PTP6	PTP5	PTP4	PTP3	PTP2	PTP1	PTP0
0x0259	PTIP	R W	PTIP7	PTIP6	PTIP5	PTIP4	PTIP3	PTIP2	PTIP1	PTIP0
0x025A	DDRP	R W	DDRP7	DDRP7	DDRP5	DDRP4	DDRP3	DDRP2	DDRP1	DDRP0
0x025B	RDRP	R W	RDRP7	RDRP6	RDRP5	RDRP4	RDRP3	RDRP2	RDRP1	RDRP0
0x025C	PERP	R W	PERP7	PERP6	PERP5	PERP4	PERP3	PERP2	PERP1	PERP0
0x025D	PPSP	R W	PPSP7	PPSP6	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSS0
0x025E	PIEP	R W	PIEP7	PIEP6	PIEP5	PIEP4	PIEP3	PIEP2	PIEP1	PIEP0
0x025F	PIFP	R W	PIFP7	PIFP6	PIFP5	PIFP4	PIFP3	PIFP2	PIFP1	PIFP0

0x0328–0x033F Reserved

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0328–0x033F	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x0340–0x0367 Periodic Interrupt Timer (PIT) Map

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0340	PITCFLMT	R	PITE	PITSWAI	PITFRZ	0	0	0	0	0
		W							PFLMT1	PFLMT0
0x0341	PITFLT	R	0	0	0	0	0	0	0	0
		W					PFLT3	PFLT2	PFLT1	PFLT0
0x0342	PITCE	R	0	0	0	0	PCE3	PCE2	PCE1	PCE0
		W								
0x0343	PITMUX	R	0	0	0	0	PMUX3	PMUX2	PMUX1	PMUX0
		W								
0x0344	PITINTE	R	0	0	0		PINTE3	PINTE2	PINTE1	PINTE0
		W								
0x0345	PITTF	R	0	0	0	0	PTF3	PTF2	PTF1	PTF0
		W								
0x0346	PITMTLD0	R	PMTLD7	PMTLD6	PMTLD5	PMTLD4	PMTLD3	PMTLD2	PMTLD1	PMTLD0
		W								
0x0347	PITMTLD1	R	PMTLD7	PMTLD6	PMTLD5	PMTLD4	PMTLD3	PMTLD2	PMTLD1	PMTLD0
		W								
0x0348	PITLD0 (hi)	R	PLD15	PLD14	PLD13	PLD12	PLD11	PLD10	PLD9	PLD8
		W								
0x0349	PITLD0 (lo)	R	PLD7	PLD6	PLD5	PLD4	PLD3	PLD2	PLD1	PLD0
		W								
0x034A	PITCNT0 (hi)	R	PCNT15	PCNT14	PCNT13	PCNT12	PCNT11	PCNT10	PCNT9	PCNT8
		W								
0x034B	PITCNT0 (lo)	R	PCNT7	PCNT6	PCNT5	PCNT4	PCNT3	PCNT2	PCNT1	PCNT0
		W								
0x034C	PITLD1 (hi)	R	PLD15	PLD14	PLD13	PLD12	PLD11	PLD10	PLD9	PLD8
		W								
0x034D	PITLD1 (lo)	R	PLD7	PLD6	PLD5	PLD4	PLD3	PLD2	PLD1	PLD0
		W								
0x034E	PITCNT1 (hi)	R	PCNT15	PCNT14	PCNT13	PCNT12	PCNT11	PCNT10	PCNT9	PCNT8
		W								
0x034F	PITCNT1 (lo)	R	PCNT7	PCNT6	PCNT5	PCNT4	PCNT3	PCNT2	PCNT1	PCNT0
		W								
0x0350	PITLD2 (hi)	R	PLD15	PLD14	PLD13	PLD12	PLD11	PLD10	PLD9	PLD8
		W								
0x0351	PITLD2 (lo)	R	PLD7	PLD6	PLD5	PLD4	PLD3	PLD2	PLD1	PLD0
		W								
0x0352	PITCNT2 (hi)	R	PCNT15	PCNT14	PCNT13	PCNT12	PCNT11	PCNT10	PCNT9	PCNT8
		W								
0x0353	PITCNT2 (lo)	R	PCNT7	PCNT6	PCNT5	PCNT4	PCNT3	PCNT2	PCNT1	PCNT0
		W								