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	40MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	59
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12xs256j0vaa

Chapter 13
Pulse-Width Modulator (S12PWM8B8CV1)

Chapter 14
Serial Communication Interface (S12SCIV5)

1.2.3 Detailed Signal Descriptions

NOTE

> The pin list of the largest package version of each S12XS Family derivative gives the complete of interface signals that also exist on smaller package options, although some of them are not bonded out. For devices assembled in smaller packages all non-bonded out pins should be configured as outputs after reset in order to avoid current drawn from floating inputs. Refer to Table 1-6 for affected pins.

1.2.3.1 EXTAL, XTAL — Oscillator Pins

EXTAL and XTAL are the crystal driver and external clock pins. On reset all the device clocks are derived from the EXTAL input frequency. XTAL is the oscillator output.

1.2.3.2 $\overline{\text{RESET}}$ — External Reset Pin

The $\overline{\text{RESET}}$ pin is an active low bidirectional control signal. It acts as an input to initialize the MCU to a known start-up state. As an output it is driven low to indicate when any internal MCU reset source triggers. The $\overline{\text{RESET}}$ pin has an internal pull-up device.

1.2.3.3 TEST — Test Pin

This input only pin is reserved for factory test. This pin has a pull-down device.

NOTE

> The TEST pin must be tied to V_{SS} in all applications.

1.2.3.4 BKGD / MODC — Background Debug and Mode Pin

The BKGD/MODC pin is used as a pseudo-open-drain pin for the background debug communication. It is used as a MCU operating mode select pin during reset. The state of this pin is latched to the MODC bit at the rising edge of $\overline{\text{RESET}}$. The BKGD pin has an internal pull-up device.

1.2.3.5 PAD[15:0] / AN[15:0] — Port AD Input Pins of ATD0

PAD[15:0] are general-purpose input or output pins and analog inputs AN[15:0] of the analog-to-digital converter ATD0.

1.2.3.6 PA[7:0] — Port A I/O Pins

PA[7:0] are general-purpose input or output pins.

1.2.3.7 PB[7:0] — Port B I/O Pins

PB[7:0] are general-purpose input or output pins.

2.3.55 Port H Polarity Select Register (PPSH)

Address 0x0265 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PPSH7	PPSH6	PPSH5	PPSH4	PPSH3	PPSH2	PPSH1	PPSH0
Reset	0	0	0	0	0	0	0	0

Figure 2-53. Port H Polarity Select Register (PPSH)

[1] Read: Anytime
Write: Anytime

Table 2-52. PPSH Register Field Descriptions

Field	Description
7-0 PPSH	**Port H pull device select**—Configure pull device and pin interrupt edge polarity on input pin This bit selects a pull-up or a pull-down device if enabled on the associated port input pin. This bit also selects the polarity of the active pin interrupt edge. 1 A pull-down device selected; rising edge selected 0 A pull-up device selected; falling edge selected

2.3.56 Port H Interrupt Enable Register (PIEH)

Address 0x0266 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PIEH7	PIEH6	PIEH5	PIEH4	PIEH3	PIEH2	PIEH1	PIEH0
Reset	0	0	0	0	0	0	0	0

Figure 2-54. Port H Interrupt Enable Register (PIEH)

[1] Read: Anytime
Write: Anytime

Table 2-53. PIEH Register Field Descriptions

Field	Description
7-0 PIEH	**Port H interrupt enable**— This bit enables or disables on the edge sensitive pin interrupt on the associated pin. 1 Interrupt enabled 0 Interrupt disabled (interrupt flag masked)

3.3.2.1 Mode Register (MODE)

Address: 0x000B PRR

	7	6	5	4	3	2	1	0
R	MODC	0	0	0	0	0	0	0
W		▨	▨	▨	▨	▨	▨	▨
Reset	MODC[1]	0	0	0	0	0	0	0

[1] External signal (see Table 3-2).

▨ = Unimplemented or Reserved

Figure 3-3. Mode Register (MODE)

Read: Anytime. Write: Only if a transition is allowed (see Figure 3-5).

The MODE bits of the MODE register are used to establish the MCU operating mode.

Table 3-3. MODE Field Descriptions

Field	Description
7 MODC	**Mode Select Bit** — This bit controls the current operating mode during $\overline{\text{RESET}}$ high (inactive). The external mode pin MODC determines the operating mode during $\overline{\text{RESET}}$ low (active). The state of the pin is latched into the respective register bit after the $\overline{\text{RESET}}$ signal goes inactive (see Figure 3-3). Write restrictions exist to disallow transitions between certain modes. Figure 3-5 illustrates all allowed mode changes. Attempting non authorized transitions will not change the MODE bits, but it will block further writes to these register bits except in special modes. Write accesses to the MODE register are blocked when the device is secured.

Figure 3-4.

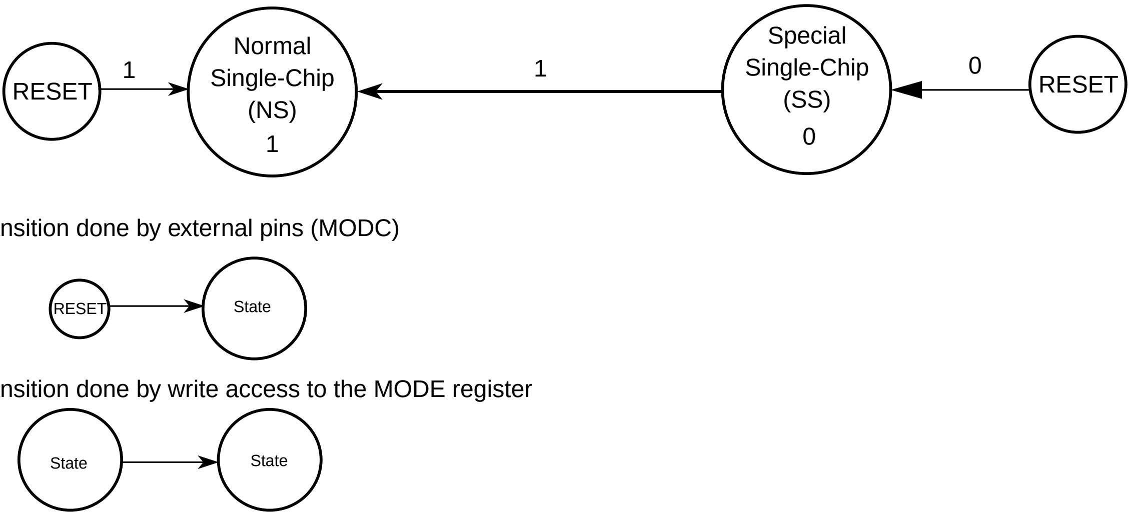

RESET
1
Normal
Single-Chip
(NS)
1
1
Special
Single-Chip
(SS)
0
0
RESET
Transition done by external pins (MODC)
RESET
State
Transition done by write access to the MODE register
State
State

Figure 3-5. Mode Transition Diagram when MCU is Unsecured

3.3.2.7 Data FLASH Page Index Register (EPAGE)

Address: 0x0017

	7	6	5	4	3	2	1	0
R W	EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0
Reset	1	1	1	1	1	1	1	0

Figure 3-15. Data FLASH Page Index Register (EPAGE)

Read: Anytime

Write: Anytime

These eight index bits are used to page 1KB blocks into the Data FLASH page window located in the local (CPU or BDM) memory map from address 0x0800 to address 0x0BFF (see Figure 3-16). This supports accessing up to 256KB of Data FLASH (in the Global map) within the 64KB Local map. The Data FLASH page index register is effectively used to construct paged Data FLASH addresses in the Local map format.

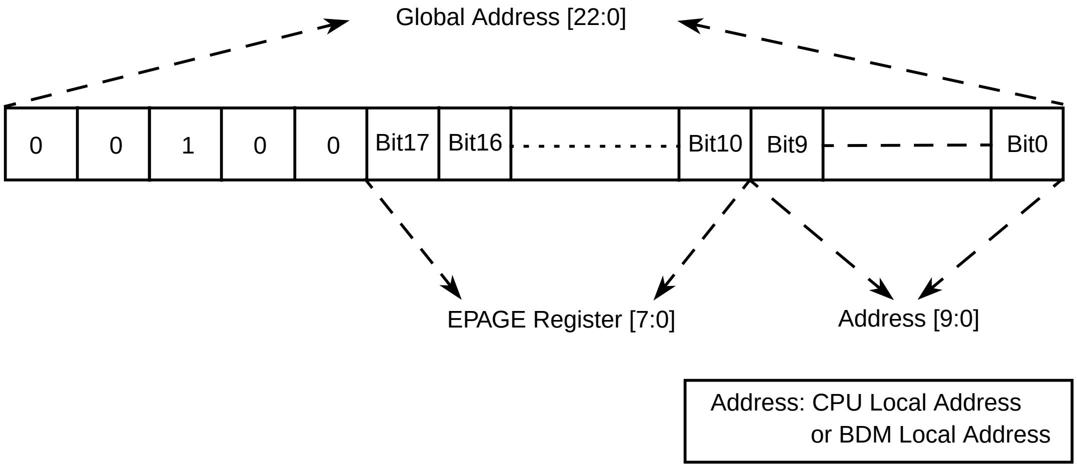


Figure 3-16. EPAGE Address Mapping

Table 3-9. EPAGE Field Descriptions

Field	Description
7–0 EP[7:0]	**Data FLASH Page Index Bits 7–0** — These page index bits are used to select which of the 256 Data FLASH array pages is to be accessed in the Data FLASH Page Window.

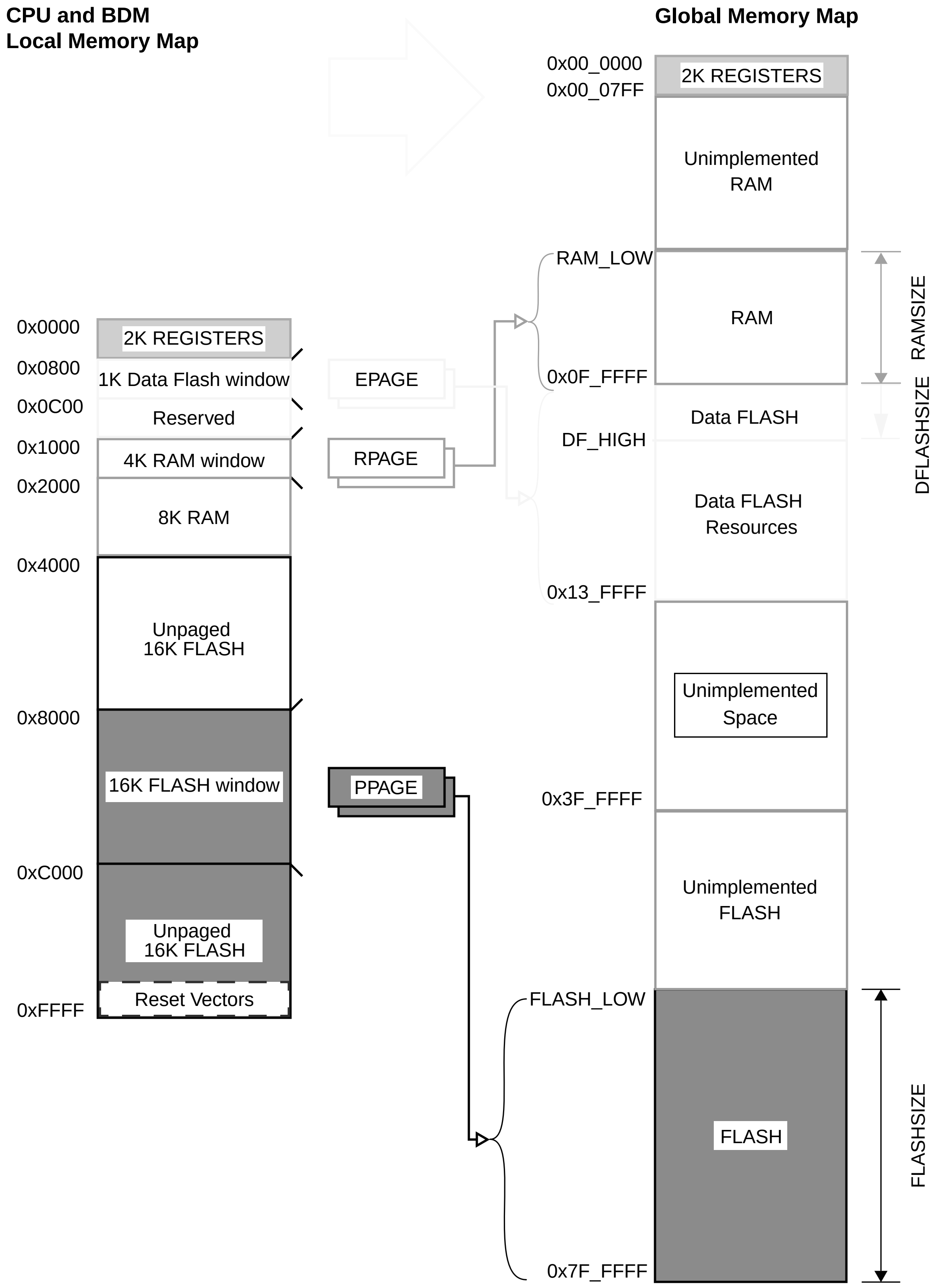


Figure 3-19. S12X CPU & BDM Global Address Mapping

Chapter 4
Interrupt (S12XINTV2)

Table 4-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V02.00	01 Jul 2005	4.1.2/4-152	Initial V2 release, added new features: - XGATE threads can be interrupted. - SYS instruction vector. - Access violation interrupt vectors.
V02.04	11 Jan 2007	4.3.2.2/4-157 4.3.2.4/4-158	- Added Notes for devices without XGATE module.
V02.05	20 Mar 2007	4.4.6/4-164	- Fixed priority definition for software exceptions.
V02.07	13 Dec 2011	4.5.3.1/4-166	- Re-worded for difference of Wake-up feature between STOP and WAIT modes.

4.1 Introduction

The XINT module decodes the priority of all system exception requests and provides the applicable vector for processing the exception to either the CPU or the XGATE module. The XINT module supports:

- I bit and X bit maskable interrupt requests
- One non-maskable unimplemented op-code trap
- One non-maskable software interrupt (SWI) or background debug mode request
- One non-maskable system call interrupt (SYS)
- Three non-maskable access violation interrupts
- One spurious interrupt vector request
- Three system reset vector requests

Each of the I bit maskable interrupt requests can be assigned to one of seven priority levels supporting a flexible priority scheme. For interrupt requests that are configured to be handled by the CPU, the priority scheme can be used to implement nested interrupt capability where interrupts from a lower level are automatically blocked if a higher level interrupt is being processed. Interrupt requests configured to be handled by the XGATE module can be nested one level deep.

NOTE

The HPRIO register and functionality of the original S12 interrupt module is no longer supported. It is superseded by the 7-level interrupt request priority scheme.

Address: 0x012C

	7	6	5	4	3	2	1	0
R	RQST	0	0	0	0	PRIOLVL[2:0]		
W		▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	1[1]

▨ = Unimplemented or Reserved

Figure 4-10. Interrupt Request Configuration Data Register 4 (INT_CFDATA4)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x012D

	7	6	5	4	3	2	1	0
R	RQST	0	0	0	0	PRIOLVL[2:0]		
W		▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	1[1]

▨ = Unimplemented or Reserved

Figure 4-11. Interrupt Request Configuration Data Register 5 (INT_CFDATA5)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x012E

	7	6	5	4	3	2	1	0
R	RQST	0	0	0	0	PRIOLVL[2:0]		
W		▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	1[1]

▨ = Unimplemented or Reserved

Figure 4-12. Interrupt Request Configuration Data Register 6 (INT_CFDATA6)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x012F

	7	6	5	4	3	2	1	0
R	RQST	0	0	0	0	PRIOLVL[2:0]		
W		▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	1[1]

▨ = Unimplemented or Reserved

Figure 4-13. Interrupt Request Configuration Data Register 7 (INT_CFDATA7)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Read: Anytime

Write: Anytime

Table 5-7. Firmware Commands

Command[1]	Opcode (hex)	Data	Description
READ_NEXT[2]	62	16-bit data out	Increment X index register by 2 (X = X + 2), then read word X points to.
READ_PC	63	16-bit data out	Read program counter.
READ_D	64	16-bit data out	Read D accumulator.
READ_X	65	16-bit data out	Read X index register.
READ_Y	66	16-bit data out	Read Y index register.
READ_SP	67	16-bit data out	Read stack pointer.
WRITE_NEXT<f-helvetica><st-superscript>	42	16-bit data in	Increment X index register by 2 (X = X + 2), then write word to location pointed to by X.
WRITE_PC	43	16-bit data in	Write program counter.
WRITE_D	44	16-bit data in	Write D accumulator.
WRITE_X	45	16-bit data in	Write X index register.
WRITE_Y	46	16-bit data in	Write Y index register.
WRITE_SP	47	16-bit data in	Write stack pointer.
GO	08	none	Go to user program. If enabled, ACK will occur when leaving active background mode.
GO_UNTIL[3]	0C	none	Go to user program. If enabled, ACK will occur upon returning to active background mode.
TRACE1	10	none	Execute one user instruction then return to active BDM. If enabled, ACK will occur upon returning to active background mode.
TAGGO -> GO	18	none	(Previous enable tagging and go to user program.) This command will be deprecated and should not be used anymore. Opcode will be executed as a GO command.

[1] If enabled, ACK will occur when data is ready for transmission for all BDM READ commands and will occur after the write is complete for all BDM WRITE commands.

[2] When the firmware command READ_NEXT or WRITE_NEXT is used to access the BDM address space the BDM resources are accessed rather than user code. Writing BDM firmware is not possible.

[3] System stop disables the ACK function and ignored commands will not have an ACK-pulse (e.g., CPU in stop or wait mode). The GO_UNTIL command will not get an Acknowledge if CPU executes the wait or stop instruction before the “UNTIL” condition (BDM active again) is reached (see Section 5.4.7, “Serial Interface Hardware Handshake Protocol” last Note).

5.4.5 BDM Command Structure

Hardware and firmware BDM commands start with an 8-bit opcode followed by a 16-bit address and/or a 16-bit data word depending on the command. All the read commands return 16 bits of data despite the byte or word implication in the command name.

8-bit reads return 16-bits of data, of which, only one byte will contain valid data. If reading an even address, the valid data will appear in the MSB. If reading an odd address, the valid data will appear in the LSB.

Table 8-11. RTI Frequency Divide Rates for RTDEC=1

RTR[3:0]	RTR[6:4] =							
	000 (1x10^{3})	**001 (2x10^{3})**	**010 (5x10^{3})**	**011 (10x10^{3})**	**100 (20x10^{3})**	**101 (50x10^{3})**	**110 (100x10^{3})**	**111 (200x10^{3})**
0110 (÷7)	7x10^{3}	14x10^{3}	35x10^{3}	70x10^{3}	140x10^{3}	350x10^{3}	700x10^{3}	1.4x10^{6}
0111 (÷8)	8x10^{3}	16x10^{3}	40x10^{3}	80x10^{3}	160x10^{3}	400x10^{3}	800x10^{3}	1.6x10^{6}
1000 (÷9)	9x10^{3}	18x10^{3}	45x10^{3}	90x10^{3}	180x10^{3}	450x10^{3}	900x10^{3}	1.8x10^{6}
1001 (÷10)	10 x10^{3}	20x10^{3}	50x10^{3}	100x10^{3}	200x10^{3}	500x10^{3}	1x10^{6}	2x10^{6}
1010 (÷11)	11 x10^{3}	22x10^{3}	55x10^{3}	110x10^{3}	220x10^{3}	550x10^{3}	1.1x10^{6}	2.2x10^{6}
1011 (÷12)	12x10^{3}	24x10^{3}	60x10^{3}	120x10^{3}	240x10^{3}	600x10^{3}	1.2x10^{6}	2.4x10^{6}
1100 (÷13)	13x10^{3}	26x10^{3}	65x10^{3}	130x10^{3}	260x10^{3}	650x10^{3}	1.3x10^{6}	2.6x10^{6}
1101 (÷14)	14x10^{3}	28x10^{3}	70x10^{3}	140x10^{3}	280x10^{3}	700x10^{3}	1.4x10^{6}	2.8x10^{6}
1110 (÷15)	15x10^{3}	30x10^{3}	75x10^{3}	150x10^{3}	300x10^{3}	750x10^{3}	1.5x10^{6}	3x10^{6}
1111 (÷16)	16x10^{3}	32x10^{3}	80x10^{3}	160x10^{3}	320x10^{3}	800x10^{3}	1.6x10^{6}	3.2x10^{6}

8.3.2.9 S12XECRG COP Control Register (COPCTL)

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

Module Base + 0x0008

	7	6	5	4	3	2	1	0
R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
W			WRTMASK	▨	▨			
Reset[1]	0	0	0	0	0	0	0	0

[1] Refer to Device User Guide (Section: S12XECRG) for reset values of WCOP, CR2, CR1 and CR0.

▨ = Unimplemented or Reserved

Figure 8-11. S12XECRG COP Control Register (COPCTL)

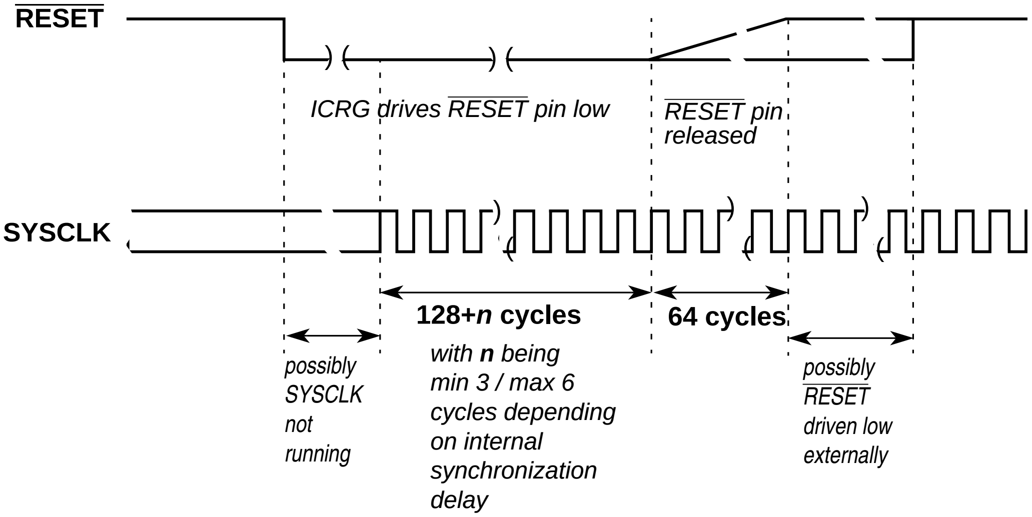
Read: Anytime

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 internal reset of the MCU remains asserted while the reset generator completes the 192 SYSCLK long reset sequence. In case the $\overline{\text{RESET}}$ pin is externally driven low for more than these 192 SYSCLK cycles (External Reset), the internal reset remains asserted longer.

Figure 8-21. RESET Timing


RESET
ICRG drives RESET pin low
RESET pin released
SYSCLK
128+n cycles
with n being min 3 / max 6 cycles depending on internal synchronization delay
64 cycles
possibly SYSCLK not running
possibly RESET driven low externally

8.5.1.1 Clock Monitor Reset

The S12XECRG generates a Clock Monitor Reset in case all of the following conditions are true:

- Clock monitor is enabled (CME = 1)
- Loss of clock is detected
- Self-Clock Mode is disabled (SCME = 0).

The reset event asynchronously forces the configuration registers to their default settings. In detail the CME and the SCME are reset to logical ‘1’ (which changes the state of the SCME bit. As a consequence the S12XECRG immediately enters Self Clock Mode and starts its internal reset sequence. In parallel the clock quality check starts. As soon as clock quality check indicates a valid Oscillator Clock the S12XECRG switches to OSCCLK and leaves Self Clock Mode. Since the clock quality checker is running in parallel to the reset generator, the S12XECRG may leave Self Clock Mode while still completing the internal reset sequence.

8.5.1.2 Computer Operating Properly Watchdog (COP) Reset

When COP is enabled, the S12XECRG expects sequential write of $55 and $AA (in this order) to the ARMCOP register during the selected time-out period. Once this is done, the COP time-out period restarts. If the program fails to do this the S12XECRG will generate a reset.

8.5.1.3 Power On Reset, Low Voltage Reset

The on-chip voltage regulator detects when V_{DD} to the MCU has reached a certain level and asserts power on reset or low voltage reset or both. As soon as a power on reset or low voltage reset is triggered the

Table 11-4. CANCTL1 Register Field Descriptions (continued)

Field	Description
1 SLPAK	**Sleep Mode Acknowledge** — This flag indicates whether the MSCAN module has entered sleep mode (see Section 11.4.5.5, "MSCAN Sleep Mode"). It is used as a handshake flag for the SLPRQ sleep mode request. Sleep mode is active when SLPRQ = 1 and SLPAK = 1. Depending on the setting of WUPE, the MSCAN will clear the flag if it detects activity on the CAN bus while in sleep mode. 0 Running — The MSCAN operates normally 1 Sleep mode active — The MSCAN has entered sleep mode
0 INITAK	**Initialization Mode Acknowledge** — This flag indicates whether the MSCAN module is in initialization mode (see Section 11.4.4.5, "MSCAN Initialization Mode"). It is used as a handshake flag for the INITRQ initialization mode request. Initialization mode is active when INITRQ = 1 and INITAK = 1. The registers CANCTL1, CANBTR0, CANBTR1, CANIDAC, CANIDAR0–CANIDAR7, and CANIDMR0–CANIDMR7 can be written only by the CPU when the MSCAN is in initialization mode. 0 Running — The MSCAN operates normally 1 Initialization mode active — The MSCAN has entered initialization mode

11.3.2.3 MSCAN Bus Timing Register 0 (CANBTR0)

The CANBTR0 register configures various CAN bus timing parameters of the MSCAN module.

Module Base + 0x0002 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
Reset:	0	0	0	0	0	0	0	0

Figure 11-6. MSCAN Bus Timing Register 0 (CANBTR0)

1. Read: Anytime
   Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1)

Table 11-5. CANBTR0 Register Field Descriptions

Field	Description
7-6 SJW[1:0]	**Synchronization Jump Width** — The synchronization jump width defines the maximum number of time quanta (Tq) clock cycles a bit can be shortened or lengthened to achieve resynchronization to data transitions on the CAN bus (see Table 11-6).
5-0 BRP[5:0]	**Baud Rate Prescaler** — These bits determine the time quanta (Tq) clock which is used to build up the bit timing (see Table 11-7).

Table 11-6. Synchronization Jump Width

SJW1	SJW0	Synchronization Jump Width
0	0	1 Tq clock cycle
0	1	2 Tq clock cycles
1	0	3 Tq clock cycles
1	1	4 Tq clock cycles

The MSCAN facilitates a sophisticated message storage system which addresses the requirements of a broad range of network applications.

11.4.2.1 Message Transmit Background

Modern application layer software is built upon two fundamental assumptions:

- Any CAN node is able to send out a stream of scheduled messages without releasing the CAN bus between the two messages. Such nodes arbitrate for the CAN bus immediately after sending the previous message and only release the CAN bus in case of lost arbitration.
- The internal message queue within any CAN node is organized such that the highest priority message is sent out first, if more than one message is ready to be sent.

The behavior described in the bullets above cannot be achieved with a single transmit buffer. That buffer must be reloaded immediately after the previous message is sent. This loading process lasts a finite amount of time and must be completed within the inter-frame sequence (IFS) to be able to send an uninterrupted stream of messages. Even if this is feasible for limited CAN bus speeds, it requires that the CPU reacts with short latencies to the transmit interrupt.

A double buffer scheme de-couples the reloading of the transmit buffer from the actual message sending and, therefore, reduces the reactiveness requirements of the CPU. Problems can arise if the sending of a message is finished while the CPU re-loads the second buffer. No buffer would then be ready for transmission, and the CAN bus would be released.

At least three transmit buffers are required to meet the first of the above requirements under all circumstances. The MSCAN has three transmit buffers.

The second requirement calls for some sort of internal prioritization which the MSCAN implements with the "local priority" concept described in Section 11.4.2.2, "Transmit Structures."

11.4.2.2 Transmit Structures

The MSCAN triple transmit buffer scheme optimizes real-time performance by allowing multiple messages to be set up in advance. The three buffers are arranged as shown in Figure 11-39.

All three buffers have a 13-byte data structure similar to the outline of the receive buffers (see Section 11.3.3, "Programmer's Model of Message Storage"). An additional Transmit Buffer Priority Register (TBPR) contains an 8-bit local priority field (PRIO) (see Section 11.3.3.4, "Transmit Buffer Priority Register (TBPR)"). The remaining two bytes are used for time stamping of a message, if required (see Section 11.3.3.5, "Time Stamp Register (TSRH–TSRL)").

To transmit a message, the CPU must identify an available transmit buffer, which is indicated by a set transmitter buffer empty (TXEx) flag (see Section 11.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)"). If a transmit buffer is available, the CPU must set a pointer to this buffer by writing to the CANTBSEL register (see Section 11.3.2.11, "MSCAN Transmit Buffer Selection Register (CANTBSEL)"). This makes the respective buffer accessible within the CANTXFG address space (see Section 11.3.3, "Programmer's Model of Message Storage"). The algorithmic feature associated with the CANTBSEL register simplifies the transmit buffer selection. In addition, this scheme makes the handler

Table 11-36. Time Segment Syntax

Syntax	Description
SYNC_SEG	System expects transitions to occur on the CAN bus during this period.
Transmit Point	A node in transmit mode transfers a new value to the CAN bus at this point.
Sample Point	A node in receive mode samples the CAN bus at this point. If the three samples per bit option is selected, then this point marks the position of the third sample.

The synchronization jump width (see the Bosch CAN 2.0A/B specification for details) can be programmed in a range of 1 to 4 time quanta by setting the SJW parameter.

The SYNC_SEG, TSEG1, TSEG2, and SJW parameters are set by programming the MSCAN bus timing registers (CANBTR0, CANBTR1) (see Section 11.3.2.3, "MSCAN Bus Timing Register 0 (CANBTR0)" and Section 11.3.2.4, "MSCAN Bus Timing Register 1 (CANBTR1)").

Table 11-37 gives an overview of the Bosch CAN 2.0A/B specification compliant segment settings and the related parameter values.

NOTE

It is the user's responsibility to ensure the bit time settings are in compliance with the CAN standard.

Table 11-37. Bosch CAN 2.0A/B Compliant Bit Time Segment Settings

Time Segment 1	TSEG1	Time Segment 2	TSEG2	Synchronization Jump Width	SJW
5 .. 10	4 .. 9	2	1	1 .. 2	0 .. 1
4 .. 11	3 .. 10	3	2	1 .. 3	0 .. 2
5 .. 12	4 .. 11	4	3	1 .. 4	0 .. 3
6 .. 13	5 .. 12	5	4	1 .. 4	0 .. 3
7 .. 14	6 .. 13	6	5	1 .. 4	0 .. 3
8 .. 15	7 .. 14	7	6	1 .. 4	0 .. 3
9 .. 16	8 .. 15	8	7	1 .. 4	0 .. 3

11.4.4 Modes of Operation

11.4.4.1 Normal System Operating Modes

The MSCAN module behaves as described within this specification in all normal system operating modes. Write restrictions exist for some registers.

Chapter 15
Serial Peripheral Interface (S12SPIV5)

Table 15-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V05.00	24 Mar 2005	15.3.2/15-439	- Added 16-bit transfer width feature.

15.1 Introduction

The SPI module allows a duplex, synchronous, serial communication between the MCU and peripheral devices. Software can poll the SPI status flags or the SPI operation can be interrupt driven.

15.1.1 Glossary of Terms

SPI	Serial Peripheral Interface
$\overline{\mathrm{SS}}$	Slave Select
SCK	Serial Clock
MOSI	Master Output, Slave Input
MISO	Master Input, Slave Output
MOMI	Master Output, Master Input
SISO	Slave Input, Slave Output

15.1.2 Features

The SPI includes these distinctive features:

- Master mode and slave mode
- Selectable 8 or 16-bit transfer width
- Bidirectional mode
- Slave select output
- Mode fault error flag with CPU interrupt capability
- Double-buffered data register
- Serial clock with programmable polarity and phase
- Control of SPI operation during wait mode

15.1.3 Modes of Operation

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

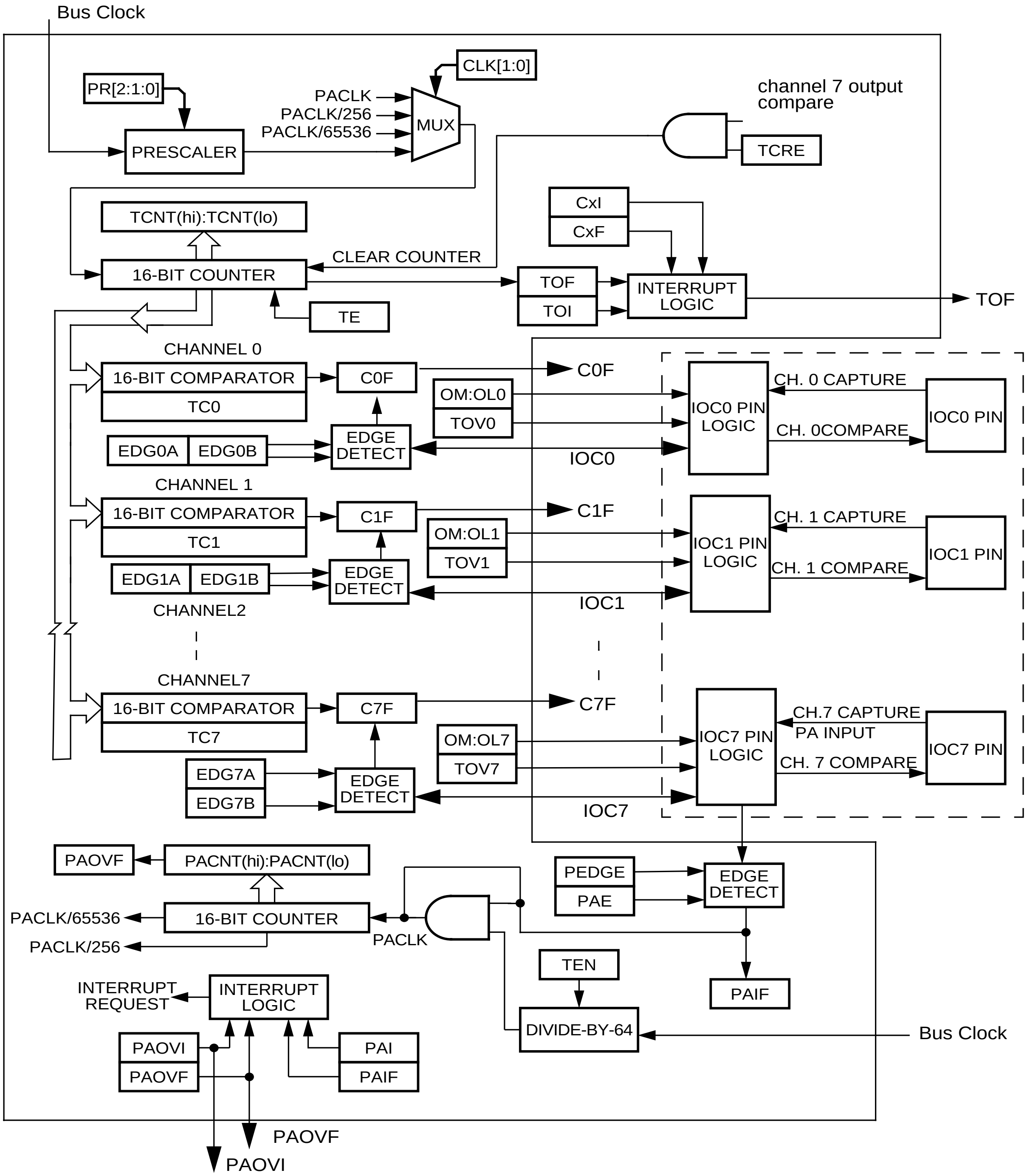


Figure 16-30. Detailed Timer Block Diagram

16.4.1 Prescaler

The prescaler divides the bus clock by 1,2,4,8,16,32,64 or 128. The prescaler select bits, PR[2:0], select the prescaler divisor. PR[2:0] are in timer system control register 2 (TSCR2).

17.2 External Signal Description

Due to the nature of VREG_3V3 being a voltage regulator providing the chip internal power supply voltages, most signals are power supply signals connected to pads.

Table 17-2 shows all signals of VREG_3V3 associated with pins.

Table 17-2. Signal Properties

Name	Function	Reset State	Pull Up
VDDR	Power input (positive supply)	—	—
VDDA	Quiet input (positive supply)	—	—
VSSA	Quiet input (ground)	—	—
VDDX	Power input (positive supply)	—	—
VDD	Primary output (positive supply)	—	—
VSS	Primary output (ground)	—	—
VDDF	Secondary output (positive supply)	—	—
VDDPLL	Tertiary output (positive supply)	—	—
VSSPLL	Tertiary output (ground)	—	—
VREGEN (optional)	Optional Regulator Enable	—	—
VREG_API (optional)	VREG Autonomous Periodical Interrupt output	—	—

NOTE

Check device level specification for connectivity of the signals.

17.2.1 VDDR — Regulator Power Input Pins

Signal VDDR is the power input of VREG_3V3. All currents sourced into the regulator loads flow through this pin. A chip external decoupling capacitor (100 nF...220 nF, X7R ceramic) between VDDR and VSSR (if VSSR is not available VSS) can smooth ripple on VDDR.

For entering Shutdown Mode, pin VDDR should also be tied to ground on devices without VREGEN pin.

17.2.2 VDDA, VSSA — Regulator Reference Supply Pins

Signals VDDA/VSSA, which are supposed to be relatively quiet, are used to supply the analog parts of the regulator. Internal precision reference circuits are supplied from these signals. A chip external decoupling capacitor (100 nF...220 nF, X7R ceramic) between VDDA and VSSA can further improve the quality of this supply.

17.2.3 VDD, VSS — Regulator Output1 (Core Logic) Pins

Signals VDD/VSS are the primary outputs of VREG_3V3 that provide the power supply for the core logic. These signals are connected to device pins to allow external decoupling capacitors (220 nF, X7R ceramic).

[1] Preferred SEC state to set MCU to secured state.

The security function in the Flash module is described in Section 19.5.

19.3.2.3 Flash CCOB Index Register (FCCOBIX)

The FCCOBIX register is used to index the FCCOB register for Flash memory operations.

Offset Module Base + 0x0002

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	CCOBIX[2:0]		
W	▨	▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 19-7. FCCOB Index Register (FCCOBIX)

CCOBIX bits are readable and writable while remaining bits read 0 and are not writable.

Table 19-11. FCCOBIX Field Descriptions

Field	Description
2–0 CCOBIX[1:0]	**Common Command Register Index**— The CCOBIX bits are used to select which word of the FCCOB register array is being read or written to. See Section 19.3.2.11, “Flash Common Command Object Register (FCCOB),” for more details.

19.3.2.4 Flash ECCR Index Register (FECCRIX)

The FECCRIX register is used to index the FECCR register for ECC fault reporting.

Offset Module Base + 0x0003

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	ECCRIX[2:0]		
W	▨	▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 19-8. FECCR Index Register (FECCRIX)

ECCRIX bits are readable and writable while remaining bits read 0 and are not writable.

Table 19-12. FECCRIX Field Descriptions

Field	Description
2-0 ECCRIX[2:0]	**ECC Error Register Index**— The ECCRIX bits are used to select which word of the FECCR register array is being read. See Section 19.3.2.14, “Flash ECC Error Results Register (FECCR),” for more details.

20.3.2.17 Flash Reserved3 Register (FRSV3)

This Flash register is reserved for factory testing.

Offset Module Base + 0x0012

	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 20-24. Flash Reserved3 Register (FRSV3)

All bits in the FRSV3 register read 0 and are not writable.

20.3.2.18 Flash Reserved4 Register (FRSV4)

This Flash register is reserved for factory testing.

Offset Module Base + 0x0013

	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 20-25. Flash Reserved4 Register (FRSV4)

All bits in the FRSV4 register read 0 and are not writable.

20.4 Functional Description

20.4.1 Flash Command Operations

Flash command operations are used to modify Flash memory contents.

The next sections describe:

- How to write the FCLKDIV register that is used to generate a time base (FCLK) derived from OSCCLK for Flash program and erase command operations
- The command write sequence used to set Flash command parameters and launch execution
- Valid Flash commands available for execution

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0356	PITCNT3 (hi)	R W	PCNT15	PCNT14	PCNT13	PCNT12	PCNT11	PCNT10	PCNT9	PCNT8
0x0357	PITCNT3 (lo)	R W	PCNT7	PCNT6	PCNT5	PCNT4	PCNT3	PCNT2	PCNT1	PCNT0
0x0358– 0x0367	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x0368–0x077F Reserved

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0368	Reserved	R	0	0	0	0	0	0	0	0
		W								