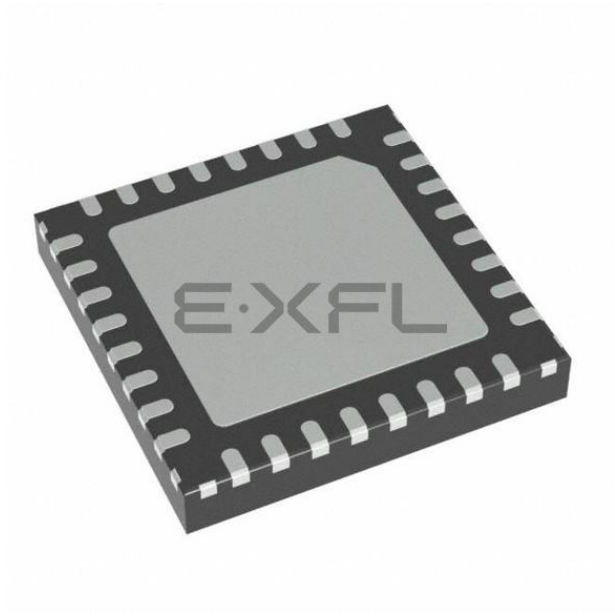

Welcome to E-XFL.COM

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	S12Z
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, I²C, IrDA, LINbus, SCI, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	19
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	5.5V ~ 18V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount, Wettable Flank
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-QFN-EP (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvl64f0mfm

Table 2-15. Port Interrupt Enable Register Field Descriptions

Field	Description
7-0 PIEx7-0	**Port Interrupt Enable** — Activate pin interrupt (KWU) This bit enables or disables the edge sensitive pin interrupt on the associated pin. An interrupt can be generated if the pin is operating in input or output mode when in use with the general-purpose or related peripheral function. 1 Interrupt is enabled 0 Interrupt is disabled (interrupt flag masked)

2.3.3.7 Port Interrupt Flag Register

Address 0x028E PIFADH
0x028F PIFADL
0x02D7 PIFS
0x02F7 PIFP
0x0337 PIFL

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PIFx7	PIFx6	PIFx5	PIFx4	PIFx3	PIFx2	PIFx1	PIFx0
Reset	0	0	0	0	0	0	0	0

Figure 2-18. Port Interrupt Flag Register

[1] Read: Anytime
Write: Anytime, write 1 to clear

Table 2-16. Port Interrupt Flag Register Field Descriptions

Field	Description
7-0 PIFx7-0	**Port Interrupt Flag** — Signal pin event (KWU) This flag asserts after a valid active edge was detected on the related pin (see Section 2.4.4.2, “Pin interrupts and Key-Wakeup (KWU)”). This can be a rising or a falling edge based on the state of the polarity select register. An interrupt will occur if the associated interrupt enable bit is set. Writing a logic “1” to the corresponding bit field clears the flag. 1 Active edge on the associated bit has occurred 0 No active edge occurred

2.3.3.8 Digital Input Enable Register

Address 0x0298 DIENADH
0x0299 DIENADL

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DIENx7	DIENx6	DIENx5	DIENx4	DIENx3	DIENx2	DIENx1	DIENx0
Reset	0	0	0	0	0	0	0	0

Figure 2-19. Digital Input Enable Register

Address: 0x00001D

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

= Unimplemented or Reserved

Figure 6-10. Interrupt Request Configuration Data Register 5 (INT_CFDATA5)

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

Address: 0x00001E

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

= Unimplemented or Reserved

Figure 6-11. Interrupt Request Configuration Data Register 6 (INT_CFDATA6)

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

Address: 0x00001F

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

= Unimplemented or Reserved

Figure 6-12. 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

8.2.2.4 ECC Debug Pointer Register (ECCDPTRH, ECCDPTRM, ECCDPTRL)

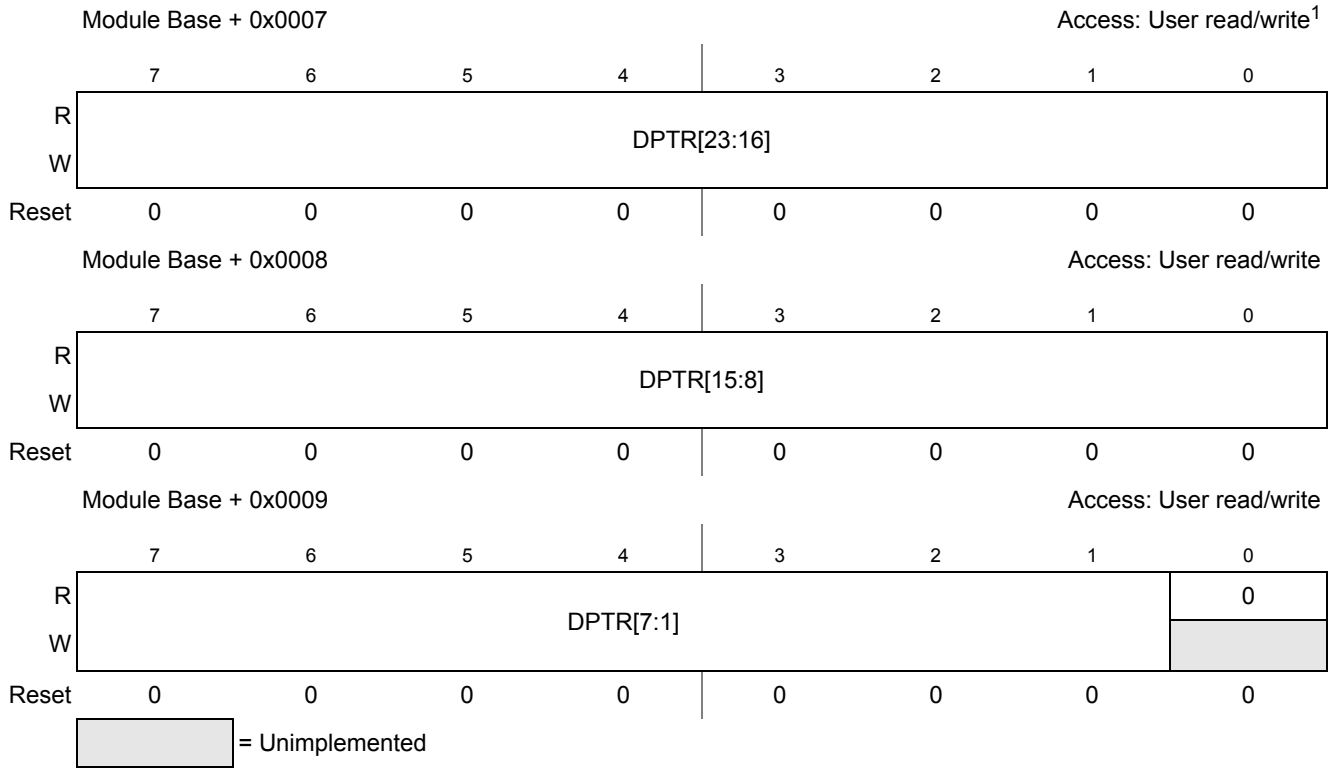

Figure 8-5. ECC Debug Pointer Register (ECCDPTRH, ECCDPTRM, ECCDPTRL)

1 Read: Anytime
Write: Anytime

Table 8-5. ECCDPTR Register Field Descriptions

Field	Description
DPTR [23:0]	**ECC Debug Pointer** — This register contains the system memory address which will be used for a debug access. Address bits not relevant for SRAM address space are not writeable, so the software should read back the pointer value to make sure the register contains the intended memory address. It is possible to write an address value to this register which points outside the system memory. There is no additional monitoring of the register content; therefore, the software must make sure that the address value points to the system memory space.

9.3.2.21 Reserved Register CPMUTEST3

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_UHV's functionality.

Module Base + 0x0016

	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 9-27. Reserved Register (CPMUTEST3)

Read: Anytime

Write: Only in Special Mode

```
/* Procedure proposed by to setup PLL and Oscillator */
/* example for OSC = 4 MHz and Bus Clock = 25MHz, That is VCOCLK = 50MHz */

/* Initialize */
/* PLL Clock = 50 MHz, divide by one */
CPMUPOSTDIV = 0x00;

/* Generally: Whenever changing PLL reference clock (REFCLK) frequency to a higher value */
/* it is recommended to write CPMUSYNR = 0x00 in order to stay within specified */
/* maximum frequency of the MCU */
CPMUSYNR = 0x00;

/* configure PLL reference clock (REFCLK) for usage with Oscillator */
/* OSC=4MHz divide by 4 (3+1) = 1MHz, REFCLK range 1MHz to 2 MHz (REFFRQ[1:0] = 00) */
CPMUREFDV = 0x03;

/* enable external Oscillator, switch PLL reference clock (REFCLK) to OSC */
CPMUOSC = 0x80;

/* multiply REFCLK = 1MHz by 2*(24+1)*1MHz = 50MHz */
/* VCO range 48 to 80 MHz (VCOFRQ[1:0] = 01) */
CPMUSYNR = 0x58;

/* clear all flags, especially LOCKIF and OSCIF */
CPMUIFLG = 0xFF;

/* put your code to loop and wait for the LOCKIF and OSCIF or */
/* poll CPMUIFLG register until both UPOSC and LOCK status are “1” */
/* that is CPMIFLG == 0x1B */

/*..............continue to your main code execution here..............*/

/* in case later in your code you want to disable the Oscillator and use the */
/* 1MHz IRCCLK as PLL reference clock */

/* Generally: Whenever changing PLL reference clock (REFCLK) frequency to a higher value */
/* it is recommended to write CPMUSYNR = 0x00 in order to stay within specified */
/* maximum frequency of the MCU */
CPMUSYNR = 0x00;

/* disable OSC and switch PLL reference clock to IRC */
CPMUOSC = 0x00;

/* multiply REFCLK = 1MHz by 2*(24+1)*1MHz = 50MHz */
/* VCO range 48 to 80 MHz (VCOFRQ[1:0] = 01) */
CPMUSYNR = 0x58;

/* clear all flags, especially LOCKIF and OSCIF */
CPMUIFLG = 0xFF;

/* put your code to loop and wait for the LOCKIF or */
/* poll CPMUIFLG register until both LOCK status is “1” */
/* that is CPMIFLG == 0x18 */

/*..............continue to your main code execution here..............*/
```

NOTE

Each conversion flow control bit (SEQA, RSTA, TRIG, LDOK) must be controlled by software or internal interface according to the requirements described in Section 10.6.3.2.4, “The two conversion flow control Mode Configurations and overview summary in Table 10-11.

10.5.2.3 ADC Status Register (ADCSTS)

It is important to note that if flag DBECC_ERR is set the ADC ceases operation. In order to make the ADC operational again an ADC Soft-Reset must be issued. An ADC Soft-Reset clears bits CSL_SEL and RVL_SEL.

Module Base + 0x0002

	7	6	5	4	3	2	1	0
R	CSL_SEL	RVL_SEL	DBECC_ERR	Reserved	READY	0	0	0
W								
Reset	0	0	0	0	1	0	0	0

= Unimplemented or Reserved

Figure 10-6. ADC Status Register (ADCSTS)

Read: Anytime

Write:

- Bits CSL_SEL and RVL_SEL anytime if bit ADC_EN is clear or bit SMOD_ACC is set
- Bits DBECC_ERR and READY not writable

Table 10-6. ADCSTS Field Descriptions

Field	Description
7 CSL_SEL	**Command Sequence List Select bit** — This bit controls and indicates which ADC Command List is active. This bit can only be written if ADC_EN bit is clear. This bit toggles in CSL double buffer mode when no conversion or conversion sequence is ongoing and bit LDOK is set and bit RSTA is set. In CSL single buffer mode this bit is forced to 1'b0 by bit CSL_BMOD. 0 ADC Command List 0 is active. 1 ADC Command List 1 is active.
6 RVL_SEL	**Result Value List Select Bit** — This bit controls and indicates which ADC Result List is active. This bit can only be written if bit ADC_EN is clear. After storage of the initial Result Value List this bit toggles in RVL double buffer mode whenever the conversion result of the first conversion of the current CSL is stored or a CSL got aborted. In RVL single buffer mode this bit is forced to 1'b0 by bit RVL_BMOD. Please see also Section 10.3.1.2, "MCU Operating Modes for information regarding Result List usage in case of Stop or Wait Mode. 0 ADC Result List 0 is active. 1 ADC Result List 1 is active.
5 DBECC_ERR	**Double Bit ECC Error Flag** — This flag indicates that a double bit ECC error occurred during conversion command load or result storage and ADC ceases operation. In order to make the ADC operational again an ADC Soft-Reset must be issued. This bit is cleared if bit ADC_EN is clear. 0 No double bit ECC error occurred. 1 A double bit ECC error occurred.
3 READY	**Ready For Restart Event Flag** — This flag indicates that ADC is in its idle state and ready for a Restart Event. It can be used to verify after exit from Wait Mode if a Restart Event can be issued and processed immediately without any latency time due to an ongoing Sequence Abort Event after exit from MCU Wait Mode (see also the Note in Section 10.3.1.2, "MCU Operating Modes). 0 ADC not in idle state. 1 ADC is in idle state.

10.5.2.18 ADC Command Register 3 (ADCCMD_3)

Module Base + 0x0017

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

= Unimplemented or Reserved

Figure 10-21. ADC Command Register 3 (ADCCMD_3)

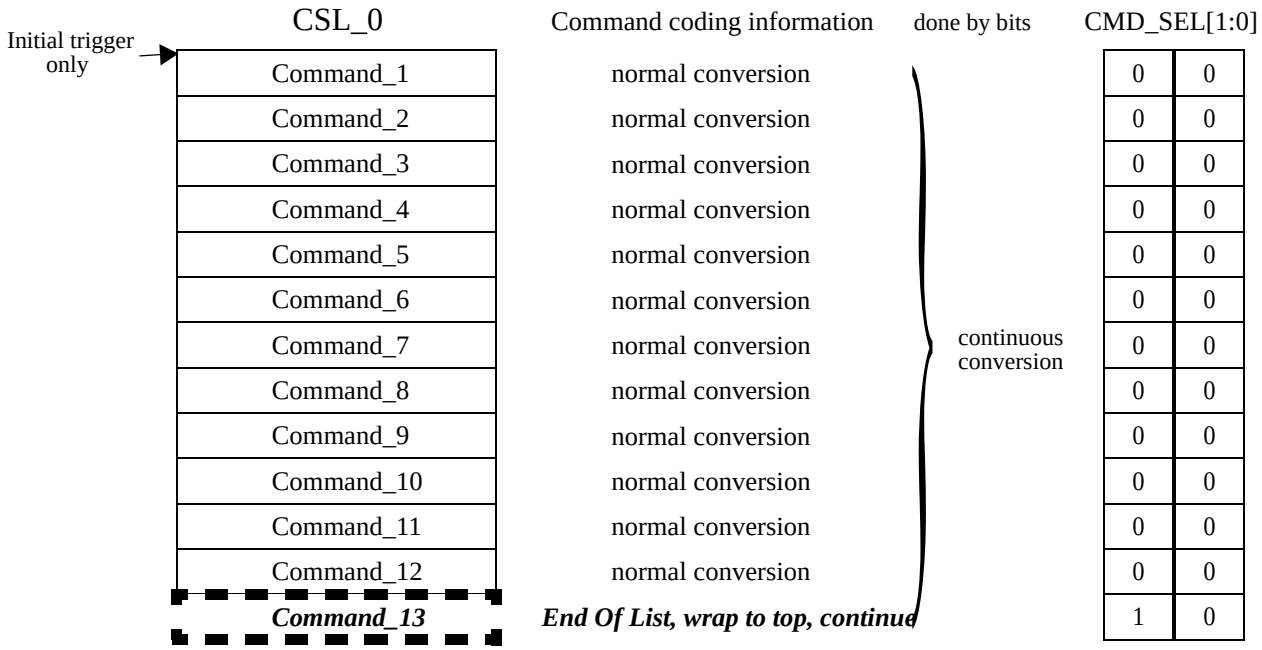

Figure 10-30. Example CSL for continues conversion

Table 13-18. CANIDAC Register Field Descriptions

Field	Description
5-4 IDAM[1:0]	**Identifier Acceptance Mode** — The CPU sets these flags to define the identifier acceptance filter organization (see Section 13.4.3, “Identifier Acceptance Filter”). Table 13-19 summarizes the different settings. In filter closed mode, no message is accepted such that the foreground buffer is never reloaded.
2-0 IDHIT[2:0]	Identifier Acceptance Hit Indicator — The MSCAN sets these flags to indicate an identifier acceptance hit (see Section 13.4.3, “Identifier Acceptance Filter”). Table 13-20 summarizes the different settings.

Table 13-19. Identifier Acceptance Mode Settings

IDAM1	IDAM0	Identifier Acceptance Mode
0	0	Two 32-bit acceptance filters
0	1	Four 16-bit acceptance filters
1	0	Eight 8-bit acceptance filters
1	1	Filter closed

Table 13-20. Identifier Acceptance Hit Indication

IDHIT2	IDHIT1	IDHIT0	Identifier Acceptance Hit
0	0	0	Filter 0 hit
0	0	1	Filter 1 hit
0	1	0	Filter 2 hit
0	1	1	Filter 3 hit
1	0	0	Filter 4 hit
1	0	1	Filter 5 hit
1	1	0	Filter 6 hit
1	1	1	Filter 7 hit

The IDHITx indicators are always related to the message in the foreground buffer (RxFG). When a message gets shifted into the foreground buffer of the receiver FIFO the indicators are updated as well.

13.3.2.13 MSCAN Reserved Registers

These registers are reserved for factory testing of the MSCAN module and is not available in normal system operating modes.

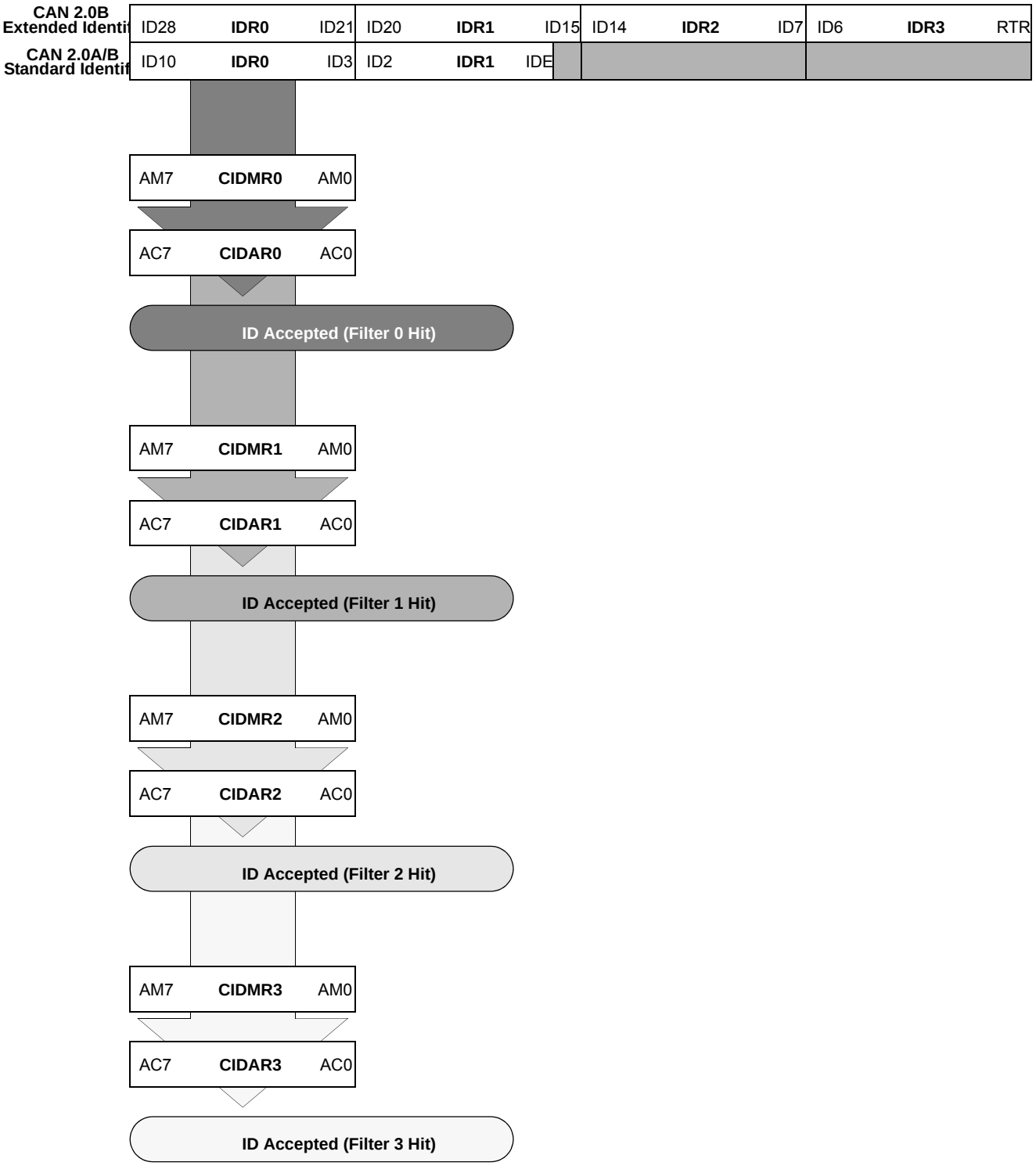

Figure 13-41. 8-bit Maskable Identifier Acceptance Filters

16.3.2.11 Main Timer Interrupt Flag 2 (TFLG2)

Module Base + 0x000F

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

Unimplemented or Reserved

Figure 16-17. Main Timer Interrupt Flag 2 (TFLG2)

TFLG2 indicates when interrupt conditions have occurred. To clear a bit in the flag register, write the bit to one while TEN bit of TSCR1 .

Read: Anytime

Write: Used in clearing mechanism (set bits cause corresponding bits to be cleared).

Any access to TCNT will clear TFLG2 register if the TFFCA bit in TSCR register is set.

Table 16-14. TRLG2 Field Descriptions

Field	Description
7 TOF	**Timer Overflow Flag** — Set when 16-bit free-running timer overflows from 0xFFFF to 0x0000. Clearing this bit requires writing a one to bit 7 of TFLG2 register while the TEN bit of TSCR1 is set to one .

18.3.1 Module Memory Map and Register Definition

The memory map for the SCI module is given below in Figure 18-2. The address listed for each register is the address offset. The total address for each register is the sum of the base address for the SCI module and the address offset for each register.

18.3.2 Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Writes to a reserved register locations do not have any effect and reads of these locations return a zero. Details of register bit and field function follow the register diagrams, in bit order.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 SCIBDH[1]	R	SBR15	SBR14	SBR13	SBR12	SBR11	SBR10	SBR9	SBR8
	W								
0x0001 SCIBDL[1]	R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
	W								
0x0002 SCICR1[1]	R	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
	W								
0x0000 SCIASR1[2]	R	RXEDGIF	0	0	0	0	BERRV	BERRIF	BKDIF
	W								
0x0001 SCIACR1[2]	R	RXEDGIE	0	0	0	0	0	BERRIE	BKDIE
	W								
0x0002 SCIACR2[2]	R	IREN	TNP1	TNP0	0	0	BERRM1	BERRM0	BKDFE
	W								
0x0003 SCICR2	R	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
	W								
0x0004 SCISR1	R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
	W								
0x0005 SCISR2	R	AMAP	0	0	TXPOL	RXPOL	BRK13	TXDIR	RAF
	W								

= Unimplemented or Reserved

Figure 18-2. SCI Register Summary (Sheet 1 of 2)

Table 19-2. SPICR1 Field Descriptions (continued)

Field	Description
3 CPOL	**SPI Clock Polarity Bit** — This bit selects an inverted or non-inverted SPI clock. To transmit data between SPI modules, the SPI modules must have identical CPOL values. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Active-high clocks selected. In idle state SCK is low. 1 Active-low clocks selected. In idle state SCK is high.
2 CPHA	**SPI Clock Phase Bit** — This bit is used to select the SPI clock format. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Sampling of data occurs at odd edges (1,3,5,...) of the SCK clock. 1 Sampling of data occurs at even edges (2,4,6,...) of the SCK clock.
1 SSOE	**Slave Select Output Enable** — The $\overline{SS}$ output feature is enabled only in master mode, if MODFEN is set, by asserting the SSOE as shown in Table 19-3. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state.
0 LSBFE	**LSB-First Enable** — This bit does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register always have the MSB in the highest bit position. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Data is transferred most significant bit first. 1 Data is transferred least significant bit first.

Table 19-3. $\overline{SS}$ Input / Output Selection

MODFEN	SSOE	Master Mode	Slave Mode
0	0	$\overline{SS}$ not used by SPI	$\overline{SS}$ input
0	1	$\overline{SS}$ not used by SPI	$\overline{SS}$ input
1	0	$\overline{SS}$ input with MODF feature	$\overline{SS}$ input
1	1	$\overline{SS}$ is slave select output	$\overline{SS}$ input

19.3.2.2 SPI Control Register 2 (SPICR2)

Module Base +0x0001

	7	6	5	4	3	2	1	0
R	0	XFRW	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 19-4. SPI Control Register 2 (SPICR2)

Read: Anytime

Write: Anytime; writes to the reserved bits have no effect

22.4.7.10 Unsecure Flash Command

The Unsecure Flash command will erase the entire P-Flash and EEPROM memory space and, if the erase is successful, will release security.

Table 22-51. Unsecure Flash Command FCCOB Requirements

Register	FCCOB Parameters	
FCCOB0	0x0B	Not required

Upon clearing CCIF to launch the Unsecure Flash command, the Memory Controller will erase the entire P-Flash and EEPROM memory space and verify that it is erased. If the Memory Controller verifies that the entire Flash memory space was properly erased, security will be released. If the erase verify is not successful, the Unsecure Flash operation sets MGSTAT1 and terminates without changing the security state. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag is set after the Unsecure Flash operation has completed.

Table 22-52. Unsecure Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if command not available in current mode (see Table 22-28)
	FPVIOL	Set if any area of the P-Flash or EEPROM memory is protected
	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

22.4.7.11 Verify Backdoor Access Key Command

The Verify Backdoor Access Key command will only execute if it is enabled by the KEYEN bits in the FSEC register (see Table 22-9). The Verify Backdoor Access Key command releases security if user-supplied keys match those stored in the Flash security bytes of the Flash configuration field (see Table 22-3). The Verify Backdoor Access Key command must not be executed from the Flash block containing the backdoor comparison key to avoid code runaway.

Table 22-53. Verify Backdoor Access Key Command FCCOB Requirements

Register	FCCOB Parameters	
FCCOB0	0x0C	Not required
FCCOB1	Key 0	
FCCOB2	Key 1	
FCCOB3	Key 2	
FCCOB4	Key 3	

22.4.10 Stop Mode

If a Flash command is active (CCIF = 0) when the MCU requests stop mode, the current Flash operation will be completed before the MCU is allowed to enter stop mode.

22.5 Security

The Flash module provides security information to the MCU. The Flash security state is defined by the SEC bits of the FSEC register (see Table 22-10). During reset, the Flash module initializes the FSEC register using data read from the security byte of the Flash configuration field at global address 0xFF_FE0F. The security state out of reset can be permanently changed by programming the security byte assuming that the MCU is starting from a mode where the necessary P-Flash erase and program commands are available and that the upper region of the P-Flash is unprotected. If the Flash security byte is successfully programmed, its new value will take affect after the next MCU reset.

The following subsections describe these security-related subjects:

- Unsecuring the MCU using Backdoor Key Access
- Unsecuring the MCU in Special Single Chip Mode using BDM
- Mode and Security Effects on Flash Command Availability

22.5.1 Unsecuring the MCU using Backdoor Key Access

The MCU may be unsecured by using the backdoor key access feature which requires knowledge of the contents of the backdoor keys (four 16-bit words programmed at addresses 0xFF_FE00-0xFF_FE07). If the KEYEN[1:0] bits are in the enabled state (see Section 22.3.2.2), the Verify Backdoor Access Key command (see Section 22.4.7.11) allows the user to present four prospective keys for comparison to the keys stored in the Flash memory via the Memory Controller. If the keys presented in the Verify Backdoor Access Key command match the backdoor keys stored in the Flash memory, the SEC bits in the FSEC register (see Table 22-10) will be changed to unsecure the MCU. Key values of 0x0000 and 0xFFFF are not permitted as backdoor keys. While the Verify Backdoor Access Key command is active, P-Flash memory and EEPROM memory will not be available for read access and will return invalid data.

The user code stored in the P-Flash memory must have a method of receiving the backdoor keys from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN[1:0] bits are in the enabled state (see Section 22.3.2.2), the MCU can be unsecured by the backdoor key access sequence described below:

1. Follow the command sequence for the Verify Backdoor Access Key command as explained in Section 22.4.7.11
2. If the Verify Backdoor Access Key command is successful, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 10

The Verify Backdoor Access Key command is monitored by the Memory Controller and an illegal key will prohibit future use of the Verify Backdoor Access Key command. A reset of the MCU is the only method to re-enable the Verify Backdoor Access Key command. The security as defined in the Flash security byte (0xFF_FE0F) is not changed by using the Verify Backdoor Access Key command sequence. The backdoor

O.8 0x0400-0x042F TIM1 (continued)

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0413	TIM1TC1L	R W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0414– 0x042B	Reserved	R W								
0x042C	TIM1OCPD	R W	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	OCPD1	OCPD0
0x042D	Reserved	R W								
0x042E	TIM1PTPSR	R W	PTPS7	PTPS6	PTPS5	PTPS4	PTPS3	PTPS2	PTPS1	PTPS0
0x042F	Reserved	R W W								

O.9 0x0480-x04AF PWM0

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0480	PWME	R W	PWME7	PWME6	PWME5	PWME4	PWME3	PWME2	PWME1	PWME0
0x0481	PWMPOL	R W	PPOL7	PPOL6	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0
0x0482	PWMCLK	R W	PCLK7	PCLKL6	PCLK5	PCLK4	PCLK3	PCLK2	PCLK1	PCLK0
0x0483	PWMPRCLK	R W	0	PCKB2	PCKB1	PCKB0	0	PCKA2	PCKA1	PCKA0
0x0484	PWMCAE	R W	CAE7	CAE6	CAE5	CAE4	CAE3	CAE2	CAE1	CAE0
0x0485	PWMCTL	R W	CON67	CON45	CON23	CON01	PSWAI	PFRZ	0	0
0x0486	PWMCLKAB	R W	PCLKAB7	PCLKAB6	PCLKAB5	PCLKAB4	PCLKAB3	PCLKAB2	PCLKAB1	PCLKAB0
0x0487	RESERVED	R W	0	0	0	0	0	0	0	0
0x0488	PWMSCLA	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0489	PWMSCLB	R W	Bit 7	6	5	4	3	2	1	Bit 0

O.14 0x0690-0x0697 ACMP

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0690	ACMPC0	R	ACE	ACOPE	ACOPS	ACDLY	ACHYS1-0		ACMOD1-0	
		W								
0x0b91	ACMPC1	R	0	0	ACPSEL1-0		0	0	ACNSEL1-0	
		W								
0x0692	ACMPC2	R	0	0	0	0	0	0	0	ACIE
		W								
0x0693	ACMPS	R	ACO	0	0	0	0	0	0	ACIF
		W								
0x0694–0x0697	Reserved	R	0	0	0	0	0	0	0	0
		W								

O.15 0x06C0-0x06DF CPMU

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x06C0	CPMU RESERVED00	R	0	0	0	0	0	0	0	0
		W								
0x06C1	CPMU RESERVED01	R	0	0	0	0	0	0	0	0
		W								
0x06C2	CPMU RESERVED02	R	0	0	0	0	0	0	0	0
		W								
0x06C3	CPMURFLG	R	0	PORF	LVRF	0	COPRF	0	OMRF	PMRF
		W								
0x06C4	CPMU SYNR	R	VCOFRQ[1:0]		SYNDIV[5:0]					
		W								
0x06C5	CPMU REFDIV	R	REFFRQ[1:0]		0	0	REFDIV[3:0]			
		W								
0x06C6	CPMU POSTDIV	R	0	0	0	POSTDIV[4:0]				
		W								
0x06C7	CPMUIFLG	R	RTIF	0	0	LOCKIF	LOCK	0	OSCIF	UPOSC
		W								
0x06C8	CPMUINT	R	RTIE	0	0	LOCKIE	0	0	OSCIE	0
		W								
0x06C9	CPMUCLKS	R	PLLSEL	PSTP	CSAD	COP OSCSEL1	PRE	PCE	RTI OSCSEL	COP OSCSEL0
		W								
0x06CA	CPMUPLL	R	0	0	FM1	FM0	0	0	0	0
		W								
0x06CB	CPMURTI	R	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
		W								