NXP USA Inc. - S9S12P64J0MLH Datasheet

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	49
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	4K 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	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12p64j0mlh

Chapter 2
Port Integration Module (S12PPIMV1)

Chapter 3
Memory Map Control (S12PMMCV1)

Chapter 4
Interrupt Module (S12SINTV1)

Chapter 5
Background Debug Module (S12SBDMV1)

Table 1-12. Interrupt Vector Locations (Sheet 2 of 3)

Vector Address[1]	Interrupt Source	CCR Mask	Local Enable	Wake up from STOP	Wakeup from WAIT
Vector base+ $F0	RTI timeout interrupt	I bit	CPMUINT (RTIE)	7.6 Interrupts	
Vector base+ $EE	TIM timer channel 0	I bit	TIE (C0I)	No	Yes
Vector base + $EC	TIM timer channel 1	I bit	TIE (C1I)	No	Yes
Vector base+ $EA	TIM timer channel 2	I bit	TIE (C2I)	No	Yes
Vector base+ $E8	TIM timer channel 3	I bit	TIE (C3I)	No	Yes
Vector base+ $E6	TIM timer channel 4	I bit	TIE (C4I)	No	Yes
Vector base+ $E4	TIM timer channel 5	I bit	TIE (C5I)	No	Yes
Vector base + $E2	TIM timer channel 6	I bit	TIE (C6I)	No	Yes
Vector base+ $E0	TIM timer channel 7	I bit	TIE (C7I)	No	Yes
Vector base+ $DE	TIM timer overflow	I bit	TSRC2 (TOF)	No	Yes
Vector base+ $DC	TIM Pulse accumulator A overflow	I bit	PACTL (PAOVI)	No	Yes
Vector base + $DA	TIM Pulse accumulator input edge	I bit	PACTL (PAI)	No	Yes
Vector base + $D8	SPI	I bit	SPICR1 (SPIE, SPTIE)	No	Yes
Vector base+ $D6	SCI	I bit	SCICR2 (TIE, TCIE, RIE, ILIE)	Yes	Yes
Vector base + $D4	Reserved				
Vector base + $D2	ATD	I bit	ATDCTL2 (ASCIE)	Yes	Yes
Vector base + $D0	Reserved				
Vector base + $CE	Port J	I bit	PIEJ (PIEJ7-PIEJ6, PIEJ2-PIEJ0)	Yes	Yes
Vector base + $CC to Vector base + $CA	Reserved				
Vector base + $C8	Oscillator status interrupt	I bit	CPMUINT (OSCIE)	No	No
Vector base + $C6	PLL lock interrupt	I bit	CPMUINT (LOCKIE)	No	No
Vector base + $C4 to Vector base + $BC	Reserved				
Vector base + $BA	FLASH error	I bit	FERCNFG (SFDIE, DFDIE)	No	No
Vector base + $B8	FLASH command	I bit	FCNFG (CCIE)	No	Yes
Vector base + $B6	CAN wake-up	I bit	CANRIER (WUPIE)	8.4.7 Interrupts	
Vector base + $B4	CAN errors	I bit	CANRIER (CSCIE, OVRIE)		
Vector base + $B2	CAN receive	I bit	CANRIER (RXFIE)		
Vector base + $B0	CAN transmit	I bit	CANTIER (TXEIE[2:0])		

2.3.24 Port S Data Register (PTS)

Address 0x0248 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTS3	PTS2	PTS1	PTS0
W	▨	▨	▨	▨				
Altern. Function	—	—	—	—	—	—	TXD	RXD
Reset	0	0	0	0	0	0	0	0

Figure 2-22. Port S Data Register (PTS)

[1] Read: Anytime The data source is depending on the data direction value.
Write: Anytime

Table 2-21. PTS Register Field Descriptions

Field	Description
3-2 PTS	**Port S general purpose input/output data**—Data Register When not used with the alternative function, the associated pin can be used as general purpose I/O. In general purpose output mode the register bit value is driven to the pin. If the associated data direction bit is set to 1, a read returns the value of the port register bit, otherwise the buffered pin input state is read.
1 PTS	**Port S general purpose input/output data**—Data Register, SCI TXD output When not used with the alternative function, the associated pin can be used as general purpose I/O. In general purpose output mode the register bit value is driven to the pin. If the associated data direction bit is set to 1, a read returns the value of the port register bit, otherwise the buffered pin input state is read. • The SCI function takes precedence over the general purpose I/O function if enabled.
0 PTS	**Port S general purpose input/output data**—Data Register, SCI RXD input When not used with the alternative function, the associated pin can be used as general purpose I/O. In general purpose output mode the register bit value is driven to the pin. If the associated data direction bit is set to 1, a read returns the value of the port register bit, otherwise the buffered pin input state is read. • The SCI function takes precedence over the general purpose I/O function if enabled.

2.3.25 Port S Input Register (PTIS)

Address 0x0249 Access: User read[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTIS3	PTIS2	PTIS1	PTIS0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	u	u	u	u	u	u	u	u

▨ = Unimplemented or Reserved u = Unaffected by reset

Figure 2-23. Port S Input Register (PTIS)

2.3.44 Port P Pull Device Enable Register (PERP)

Address 0x025C Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	PPSP7	0	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSP0
W		▨						
Reset	0	0	0	0	0	0	0	0

Figure 2-42. Port P Pull Device Enable Register (PERP)

[1] Read: Anytime
Write: Anytime

Table 2-39. PERP Register Field Descriptions

Field	Description
7,5-0 PERP	**Port P pull device enable**—Enable pull device on input pin This bit controls whether a pull device on the associated port input pin is active. If a pin is used as output this bit has no effect. The polarity is selected by the related polarity select register bit. 1 Pull device enabled 0 Pull device disabled

2.3.45 Port P Polarity Select Register (PPSP)

Address 0x025D Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	PPSP7	0	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSP0
W		▨						
Reset	0	0	0	0	0	0	0	0

Figure 2-43. Port P Polarity Select Register (PPSP)

[1] Read: Anytime
Write: Anytime

Table 2-40. PPSP Register Field Descriptions

Field	Description
7,5-0 PPSP	**Port P 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 is selected; rising edge selected 0 A pull-up device is selected; falling edge selected

2.3.50 Port J Input Register (PTIJ)

Address 0x0269 Access: User read(1)

	7	6	5	4	3	2	1	0
R	PTIJ7	PTIJ6	0	0	0	PTIJ2	PTIJ1	PTIJ0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	u	u	u	u	u	u	u	u

▨ = Unimplemented or Reserved u = Unaffected by reset

Figure 2-48. Port J Input Register (PTIJ)

1. Read: Anytime
   Write:Never, writes to this register have no effect.

Table 2-44. PTIJ Register Field Descriptions

Field	Description
7-6, 2-0 PTIJ	**Port J input data**— A read always returns the buffered input state of the associated pin. It can be used to detect overload or short circuit conditions on output pins.

2.3.51 Port J Data Direction Register (DDRJ)

Address 0x026A Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	DDRJ7	DDRJ6	0	0	0	DDRJ2	DDRJ1	DDRJ0
W			▨	▨	▨			
Reset	0	0	0	0	0	0	0	0

Figure 2-49. Port J Data Direction Register (DDRJ)

1. Read: Anytime
   Write: Anytime

Table 2-45. DDRJ Register Field Descriptions

Field	Description
7-6, 2-0 DDRJ	**Port J data direction**— This bit determines whether the associated pin is an input or output. 1 Associated pin is configured as output 0 Associated pin is configured as input

2.3.60 Port AD Data Direction Register (DDR1AD)

Address 0x0273 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDR1AD7	DDR1AD6	DDR1AD5	DDR1AD4	DDR1AD3	DDR1AD2	DDR1AD1	DDR1AD0
Reset	0	0	0	0	0	0	0	0

Figure 2-58. Port AD Data Direction Register (DDR1AD)

1. Read: Anytime
   Write: Anytime

Table 2-54. DDR1AD Register Field Descriptions

Field	Description
7-0 DDR1AD	**Port AD data direction**— This bit determines whether the associated pin is an input or output. To use the digital input function the ATD Digital Input Enable Register (ATDDIEN) has to be set to logic level “1”. 1 Associated pin is configured as output 0 Associated pin is configured as input

2.3.61 Port AD Reduced Drive Register (RDR0AD)

Address 0x0274 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	RDR0AD1	RDR0AD0
W	▨	▨	▨	▨	▨	▨		
Reset	0	0	0	0	0	0	0	0

Figure 2-59. Port AD Reduced Drive Register (RDR0AD)

1. Read: Anytime
   Write: Anytime

Table 2-55. RDR0AD Register Field Descriptions

Field	Description
1-0 RDR0AD	**Port AD reduced drive**—Select reduced drive for output pin This bit configures the drive strength of the associated output pin as either full or reduced. If a pin is used as input this bit has no effect. The reduced drive function is independent of which function is being used on a particular pin. 1 Reduced drive selected (approx. 1/5 of the full drive strength) 0 Full drive strength enabled

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).

Table 7-9. RTI Frequency Divide Rates for RTDEC = 0

RTR[3:0]	RTR[6:4] =							
	000 (OFF)	001 (2^{10})	010 (2^{11})	011 (2^{12})	100 (2^{13})	101 (2^{14})	110 (2^{15})	111 (2^{16})
0000 (÷1)	OFF[1]	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}
0001 (÷2)	OFF	$2x2^{10}$	$2x2^{11}$	$2x2^{12}$	$2x2^{13}$	$2x2^{14}$	$2x2^{15}$	$2x2^{16}$
0010 (÷3)	OFF	$3x2^{10}$	$3x2^{11}$	$3x2^{12}$	$3x2^{13}$	$3x2^{14}$	$3x2^{15}$	$3x2^{16}$
0011 (÷4)	OFF	$4x2^{10}$	$4x2^{11}$	$4x2^{12}$	$4x2^{13}$	$4x2^{14}$	$4x2^{15}$	$4x2^{16}$
0100 (÷5)	OFF	$5x2^{10}$	$5x2^{11}$	$5x2^{12}$	$5x2^{13}$	$5x2^{14}$	$5x2^{15}$	$5x2^{16}$
0101 (÷6)	OFF	$6x2^{10}$	$6x2^{11}$	$6x2^{12}$	$6x2^{13}$	$6x2^{14}$	$6x2^{15}$	$6x2^{16}$
0110 (÷7)	OFF	$7x2^{10}$	$7x2^{11}$	$7x2^{12}$	$7x2^{13}$	$7x2^{14}$	$7x2^{15}$	$7x2^{16}$
0111 (÷8)	OFF	$8x2^{10}$	$8x2^{11}$	$8x2^{12}$	$8x2^{13}$	$8x2^{14}$	$8x2^{15}$	$8x2^{16}$
1000 (÷9)	OFF	$9x2^{10}$	$9x2^{11}$	$9x2^{12}$	$9x2^{13}$	$9x2^{14}$	$9x2^{15}$	$9x2^{16}$
1001 (÷10)	OFF	$10x2^{10}$	$10x2^{11}$	$10x2^{12}$	$10x2^{13}$	$10x2^{14}$	$10x2^{15}$	$10x2^{16}$
1010 (÷11)	OFF	$11x2^{10}$	$11x2^{11}$	$11x2^{12}$	$11x2^{13}$	$11x2^{14}$	$11x2^{15}$	$11x2^{16}$
1011 (÷12)	OFF	$12x2^{10}$	$12x2^{11}$	$12x2^{12}$	$12x2^{13}$	$12x2^{14}$	$12x2^{15}$	$12x2^{16}$
1100 (÷13)	OFF	$13x2^{10}$	$13x2^{11}$	$13x2^{12}$	$13x2^{13}$	$13x2^{14}$	$13x2^{15}$	$13x2^{16}$
1101 (÷14)	OFF	$14x2^{10}$	$14x2^{11}$	$14x2^{12}$	$14x2^{13}$	$14x2^{14}$	$14x2^{15}$	$14x2^{16}$
1110 (÷15)	OFF	$15x2^{10}$	$15x2^{11}$	$15x2^{12}$	$15x2^{13}$	$15x2^{14}$	$15x2^{15}$	$15x2^{16}$
1111 (÷16)	OFF	$16x2^{10}$	$16x2^{11}$	$16x2^{12}$	$16x2^{13}$	$16x2^{14}$	$16x2^{15}$	$16x2^{16}$

[1] Denotes the default value out of reset.This value should be used to disable the RTI to ensure future backwards compatibility.

7.3.2.18 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's functionality.

0x02F6

	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 7-23. Reserved Register (CPMUTEST3)

Read: Anytime

Write: Only in special mode

Several examples of PLL divider settings are shown in Table 7-23. The following rules help to achieve optimum stability and shortest lock time:

- Use lowest possible f_{VCO} / f_{REF} ratio (SYNDIV value).
- Use highest possible REFCLK frequency f_{REF}.

Table 7-23. Examples of PLL Divider Settings

f_{osc}	REFDIV[3:0]	f_{REF}	REFFRQ[1:0]	SYNDIV[5:0]	f_{VCO}	VCOFRQ[1:0]	POSTDIV[4:0]	f_{PLL}	f_{bus}
off	$00	1MHz	00	$1F	64MHz	01	$03	16MHz	8MHz
off	$00	1MHz	00	$1F	64MHz	01	$00	64MHz	32MHz
off	$00	1MHz	00	$0F	32MHz	00	$00	32MHz	16MHz
4MHz	$00	4MHz	01	$03	32MHz	01	$00	32MHz	16MHz

The phase detector inside the PLL compares the feedback clock (FBCLK = VCOCLK/(SYNDIV+1)) with the reference clock (REFCLK = IRC1M or OSCCLK/REFDIV+1)). Correction pulses are generated based on the phase difference between the two signals. The loop filter alters the DC voltage on the internal filter capacitor, based on the width and direction of the correction pulse, which leads to a higher or lower VCO frequency.

The user must select the range of the REFCLK frequency (REFFRQ[1:0] bits) and the range of the VCOCLK frequency (VCOFRQ[1:0] bits) to ensure that the correct PLL loop bandwidth is set.

The lock detector compares the frequencies of the FBCLK and the REFCLK. Therefore the speed of the lock detector is directly proportional to the reference clock frequency. The circuit determines the lock condition based on this comparison.

If PLL LOCK interrupt requests are enabled, the software can wait for an interrupt request and for instance check the LOCK bit. If interrupt requests are disabled, software can poll the LOCK bit continuously (during PLL start-up) or at periodic intervals. In either case, only when the LOCK bit is set, the VCOCLK will have stabilized to the programmed frequency.

- The LOCK bit is a read-only indicator of the locked state of the PLL.
- The LOCK bit is set when the VCO frequency is within the tolerance Δ_{Lock} and is cleared when the VCO frequency is out of the tolerance Δ_{unl}.
- Interrupt requests can occur if enabled (LOCKIE = 1) when the lock condition changes, toggling the LOCK bit.

For microcontrollers without a clock and reset generator (CRG), CANCLK is driven from the crystal oscillator (oscillator clock).

A programmable prescaler generates the time quanta (Tq) clock from CANCLK. A time quantum is the atomic unit of time handled by the MSCAN.

Eqn. 8-2

$$Tq = \frac{f_{CANCLK}}{(Prescaler\ value)}$$

A bit time is subdivided into three segments as described in the Bosch CAN specification. (see Figure 8-44):

- SYNC_SEG: This segment has a fixed length of one time quantum. Signal edges are expected to happen within this section.
- Time Segment 1: This segment includes the PROP_SEG and the PHASE_SEG1 of the CAN standard. It can be programmed by setting the parameter TSEG1 to consist of 4 to 16 time quanta.
- Time Segment 2: This segment represents the PHASE_SEG2 of the CAN standard. It can be programmed by setting the TSEG2 parameter to be 2 to 8 time quanta long.

Eqn. 8-3

$$Bit\ Rate = \frac{f_{Tq}}{(number\ of\ Time\ Quanta)}$$

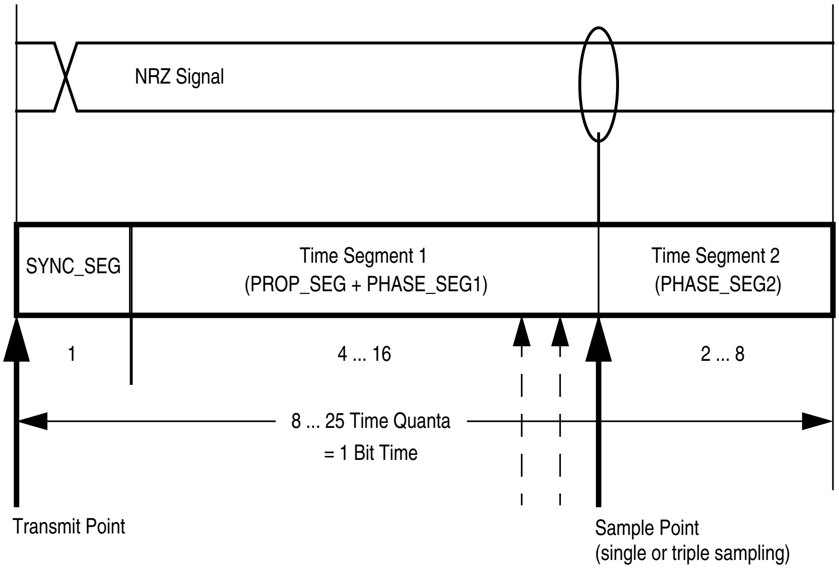

Figure 8-44. Segments within the Bit Time

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0003	ATDCTL3	R	DJM	S8C	S4C	S2C	S1C	FIFO	FRZ1	FRZ0
		W								
0x0004	ATDCTL4	R	SMP2	SMP1	SMP0	PRS[4:0]				
		W								
0x0005	ATDCTL5	R	0	SC	SCAN	MULT	CD	CC	CB	CA
		W	▨							
0x0006	ATDSTAT0	R	SCF	0	ETORF	FIFOR	CC3	CC2	CC1	CC0
		W		▨			▨	▨	▨	▨
0x0007	Unimplemented	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x0008	ATDCMPEH	R	0	0	0	0	0	0	CMPE[9:8]	
		W	▨	▨	▨	▨	▨	▨		
0x0009	ATDCMPEL	R	CMPE[7:0]							
		W								
0x000A	ATDSTAT2H	R	0	0	0	0	0	0	CCF[9:8]	
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x000B	ATDSTAT2L	R	CCF[7:0]							
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x000C	ATDDIENH	R	0	0	0	0	0	0	IEN[9:8]	
		W	▨	▨	▨	▨	▨	▨		
0x000D	ATDDIENL	R	IEN[7:0]							
		W								
0x000E	ATDCMPHTH	R	0	0	0	0	0	0	CMPHT[9:8]	
		W	▨	▨	▨	▨	▨	▨		
0x000F	ATDCMPHTL	R	CMPHT[7:0]							
		W								
0x0010	ATDDR0	R	See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0012	ATDDR1	R	See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0014	ATDDR2	R	See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0016	ATDDR3	R	See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x0018	ATDDR4	R	See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x001A	ATDDR5	R	See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x001C	ATDDR6	R	See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								
0x001E	ATDDR7	R	See Section 9.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 9.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								

▨ = Unimplemented or Reserved

Figure 9-2. ADC12B10C Register Summary (Sheet 2 of 3)

Read: anytime

Write: anytime (causes the scale counter to load the PWMSCLA value)

10.3.2.10 PWM Scale B Register (PWMSCLB)

PWMSCLB is the programmable scale value used in scaling clock B to generate clock SB. Clock SB is generated by taking clock B, dividing it by the value in the PWMSCLB register and dividing that by two.

Clock SB = Clock B / (2 * PWMSCLB)

NOTE

When PWMSCLB = 0x0000, PWMSCLB value is considered a full scale value of 256. Clock B is thus divided by 512.

Any value written to this register will cause the scale counter to load the new scale value (PWMSCLB).

Module Base + 0x0009

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

Figure 10-12. PWM Scale B Register (PWMSCLB)

Read: anytime

Write: anytime (causes the scale counter to load the PWMSCLB value).

10.3.2.11 Reserved Registers (PWMSCNTx)

The registers PWMSCNTA and PWMSCNTB are reserved for factory testing of the PWM module and are not available in normal modes.

Module Base + 0x000A

	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 10-13. Reserved Register (PWMSCNTA)

12.3.2.5 SPI Data Register (SPIDR = SPIDRH:SPIDRL)

Module Base +0x0004

	7	6	5	4	3	2	1	0
R	R15	R14	R13	R12	R11	R10	R9	R8
W	T15	T14	T13	T12	T11	T10	T9	T8
Reset	0	0	0	0	0	0	0	0

Figure 12-7. SPI Data Register High (SPIDRH)

Module Base +0x0005

	7	6	5	4	3	2	1	0
R	R7	R6	R5	R4	R3	R2	R1	R0
W	T7	T6	T5	T4	T3	T2	T1	T0
Reset	0	0	0	0	0	0	0	0

Figure 12-8. SPI Data Register Low (SPIDRL)

Read: Anytime; read data only valid when SPIF is set

Write: Anytime

The SPI data register is both the input and output register for SPI data. A write to this register allows data to be queued and transmitted. For an SPI configured as a master, queued data is transmitted immediately after the previous transmission has completed. The SPI transmitter empty flag SPTEF in the SPISR register indicates when the SPI data register is ready to accept new data.

Received data in the SPIDR is valid when SPIF is set.

If SPIF is cleared and data has been received, the received data is transferred from the receive shift register to the SPIDR and SPIF is set.

If SPIF is set and not serviced, and a second data value has been received, the second received data is kept as valid data in the receive shift register until the start of another transmission. The data in the SPIDR does not change.

If SPIF is set and valid data is in the receive shift register, and SPIF is serviced before the start of a third transmission, the data in the receive shift register is transferred into the SPIDR and SPIF remains set (see Figure 12-9).

If SPIF is set and valid data is in the receive shift register, and SPIF is serviced after the start of a third transmission, the data in the receive shift register has become invalid and is not transferred into the SPIDR (see Figure 12-10).

13.1.2 Features

13.1.2.1 P-Flash Features

- 128 Kbytes of P-Flash memory composed of one 128 Kbyte Flash block divided into 256 sectors of 512 bytes
- Single bit fault correction and double bit fault detection within a 32-bit double word during read operations
- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and phrase program operation
- Ability to read the P-Flash memory while programming a word in the D-Flash memory
- Flexible protection scheme to prevent accidental program or erase of P-Flash memory

13.1.2.2 D-Flash Features

- 4 Kbytes of D-Flash memory composed of one 4 Kbyte Flash block divided into 16 sectors of 256 bytes
- Single bit fault correction and double bit fault detection within a word during read operations
- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and word program operation
- Protection scheme to prevent accidental program or erase of D-Flash memory
- Ability to program up to four words in a burst sequence

13.1.2.3 Other Flash Module Features

- No external high-voltage power supply required for Flash memory program and erase operations
- Interrupt generation on Flash command completion and Flash error detection
- Security mechanism to prevent unauthorized access to the Flash memory

13.1.3 Block Diagram

The block diagram of the Flash module is shown in Figure 13-1.

Offset Module Base + 0x000A

	7	6	5	4	3	2	1	0
R W	CCOB[15:8]							
Reset	0	0	0	0	0	0	0	0

Figure 13-16. Flash Common Command Object High Register (FCCOBHI)

Offset Module Base + 0x000B

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

Figure 13-17. Flash Common Command Object Low Register (FCCOBLO)

13.3.2.11.1 FCCOB - NVM Command Mode

NVM command mode uses the indexed FCCOB register to provide a command code and its relevant parameters to the Memory Controller. The user first sets up all required FCCOB fields and then initiates the command's execution by writing a 1 to the CCIF bit in the FSTAT register (a 1 written by the user clears the CCIF command completion flag to 0). When the user clears the CCIF bit in the FSTAT register all FCCOB parameter fields are locked and cannot be changed by the user until the command completes (as evidenced by the Memory Controller returning CCIF to 1). Some commands return information to the FCCOB register array.

The generic format for the FCCOB parameter fields in NVM command mode is shown in Table 13-23. The return values are available for reading after the CCIF flag in the FSTAT register has been returned to 1 by the Memory Controller. Writes to the unimplemented parameter fields (CCOBIX = 110 and CCOBIX = 111) are ignored with reads from these fields returning 0x0000.

Table 13-23 shows the generic Flash command format. The high byte of the first word in the CCOB array contains the command code, followed by the parameters for this specific Flash command. For details on the FCCOB settings required by each command, see the Flash command descriptions in Section 13.4.5.

Table 13-23. FCCOB - NVM Command Mode (Typical Usage)

CCOBIX[2:0]	Byte	FCCOB Parameter Fields (NVM Command Mode)
000	HI	FCMD[7:0] defining Flash command
	LO	6'h0, Global address [17:16]
001	HI	Global address [15:8]
	LO	Global address [7:0]
010	HI	Data 0 [15:8]
	LO	Data 0 [7:0]

A.2.3.1 ATD Accuracy Definitions

For the following definitions see also Figure A-1.
Differential non-linearity (DNL) is defined as the difference between two adjacent switching steps.

$$\mathrm{DNL}(i) = \frac{V_i - V_{i-1}}{1\mathrm{LSB}} - 1$$

The integral non-linearity (INL) is defined as the sum of all DNLs:

$$\mathrm{INL}(n) = \sum_{i=1}^{n} \mathrm{DNL}(i) = \frac{V_n - V_0}{1\mathrm{LSB}} - n$$