



Welcome to [E-XFL.COM](#)

What is "[Embedded - Microcontrollers](#)"?

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

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	HCS12
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	64
Program Memory Size	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	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12p64j0mqk

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
<u>0x0020–0x0023F</u> Non-PIM Address Range	R W	Non-PIM Address Range							
<u>0x0240</u> PTT	R W	PTT7	PTT6	PTT5	PTT4	PTT3	PTT2	PTT1	PTT0
<u>0x0241</u> PTIT	R W	PTIT7	PTIT6	PTIT5	PTIT4	PTIT3	PTIT2	PTIT1	PTIT0
<u>0x0242</u> DDRT	R W	DDRT7	DDRT6	DDRT5	DDRT4	DDRT3	DDRT2	DDRT1	DDRT0
<u>0x0243</u> RDRT	R W	RDRT7	RDRT6	RDRT5	RDRT4	RDRT3	RDRT2	RDRT1	RDRT0
<u>0x0244</u> PERT	R W	PERT7	PERT6	PERT5	PERT4	PERT3	PERT2	PERT1	PERT0
<u>0x0245</u> PPST	R W	PPST7	PPST6	PPST5	PPST4	PPST3	PPST2	PPST1	PPST0
<u>0x0246</u> Reserved	R W	0	0	0	0	0	0	0	0
<u>0x0247</u> PTTTR	R W	0	0	PTTTR5	PTTTR4	0	0	0	PTTTR0
<u>0x0248</u> PTS	R W	0	0	0	0	PTS3	PTS2	PTS1	PTS0
<u>0x0249</u> PTIS	R W	0	0	0	0	PTIS3	PTIS2	PTIS1	PTIS0
<u>0x024A</u> DDRS	R W	0	0	0	0	DDRS3	DDRS2	DDRS1	DDRS0
<u>0x024B</u> RDRS	R W	0	0	0	0	RDRS3	RDRS2	RDRS1	RDRS0
<u>0x024C</u> PERS	R W	0	0	0	0	PERS3	PERS2	PERS1	PERS0
<u>0x024D</u> PPSS	R W	0	0	0	0	PPSS3	PPSS2	PPSS1	PPSS0

= Unimplemented or Reserved

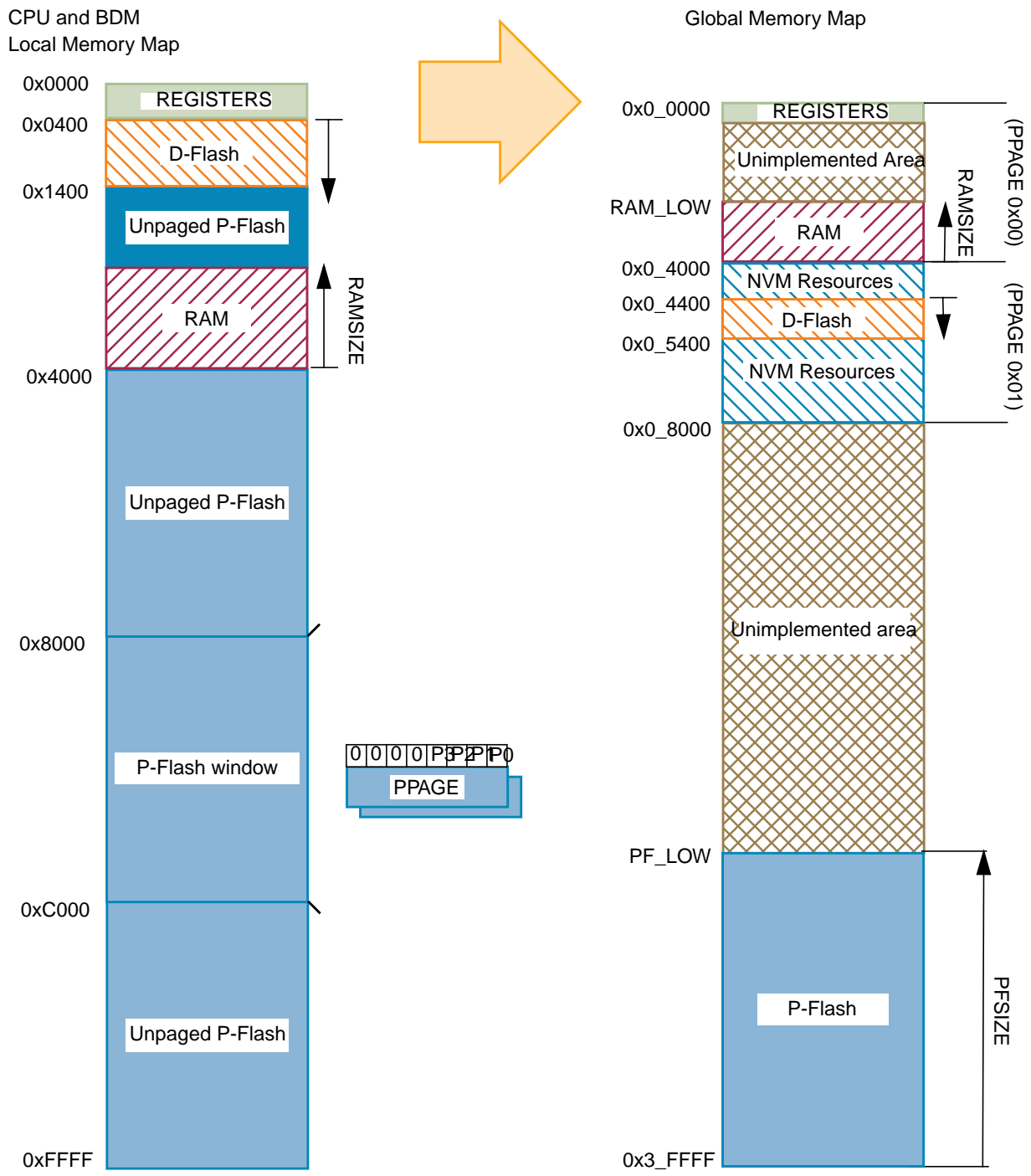


Figure 3-11. Implemented Global Address Mapping

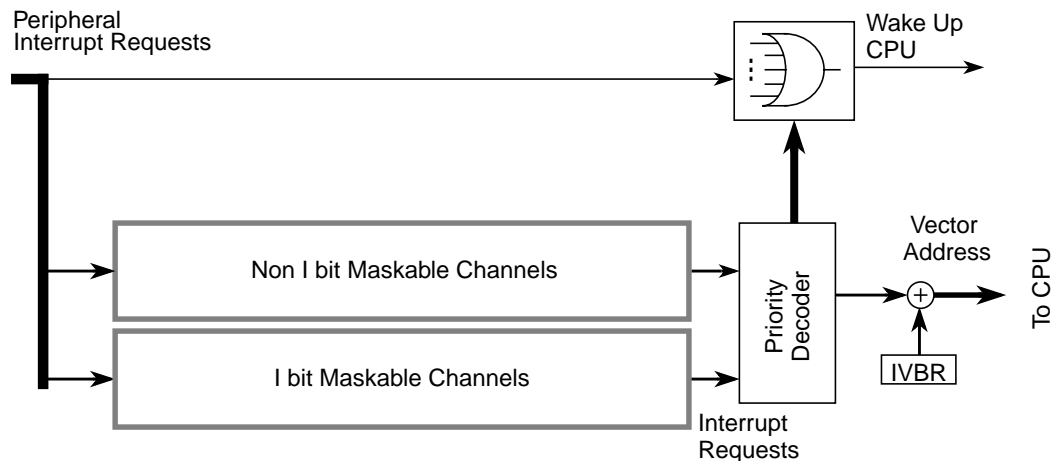


Figure 4-1. INT Block Diagram

4.2 External Signal Description

The INT module has no external signals.

4.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in the INT module.

4.3.1 Register Descriptions

This section describes in address order all the INT registers and their individual bits.

4.3.1.1 Interrupt Vector Base Register (IVBR)

Address: 0x0120

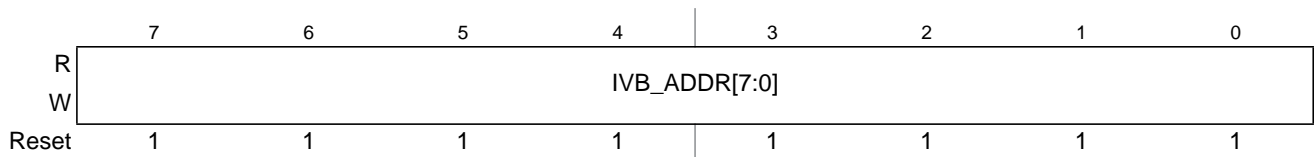


Figure 4-2. Interrupt Vector Base Register (IVBR)

Read: Anytime

Write: Anytime

5.1.1 Features

The BDM includes these distinctive features:

- Single-wire communication with host development system
- Enhanced capability for allowing more flexibility in clock rates
- SYNC command to determine communication rate
- GO_UNTIL command
- Hardware handshake protocol to increase the performance of the serial communication
- Active out of reset in special single chip mode
- Nine hardware commands using free cycles, if available, for minimal CPU intervention
- Hardware commands not requiring active BDM
- 14 firmware commands execute from the standard BDM firmware lookup table
- Software control of BDM operation during wait mode
- When secured, hardware commands are allowed to access the register space in special single chip mode, if the Flash erase tests fail.
- Family ID readable from BDM ROM at global address 0x3_FF0F in active BDM (value for devices with HCS12S core is 0xC2)
- BDM hardware commands are operational until system stop mode is entered

5.1.2 Modes of Operation

BDM is available in all operating modes but must be enabled before firmware commands are executed. Some systems may have a control bit that allows suspending the function during background debug mode.

5.1.2.1 Regular Run Modes

All of these operations refer to the part in run mode and not being secured. The BDM does not provide controls to conserve power during run mode.

- Normal modes
General operation of the BDM is available and operates the same in all normal modes.
- Special single chip mode
In special single chip mode, background operation is enabled and active out of reset. This allows programming a system with blank memory.

5.1.2.2 Secure Mode Operation

If the device is in secure mode, the operation of the BDM is reduced to a small subset of its regular run mode operation. Secure operation prevents access to Flash other than allowing erasure. For more information please see [Section 5.4.1, “Security”](#).

5.1.2.3 Low-Power Modes

The BDM can be used until stop mode is entered. When CPU is in wait mode all BDM firmware commands as well as the hardware BACKGROUND command cannot be used and are ignored. In this case the CPU can not enter BDM active mode, and only hardware read and write commands are available. Also the CPU can not enter a low power mode (stop or wait) during BDM active mode.

In stop mode the BDM clocks are stopped. When BDM clocks are disabled and stop mode is exited, the BDM clocks will restart and BDM will have a soft reset (clearing the instruction register, any command in progress and disable the ACK function). The BDM is now ready to receive a new command.

5.1.3 Block Diagram

A block diagram of the BDM is shown in Figure 5-1.

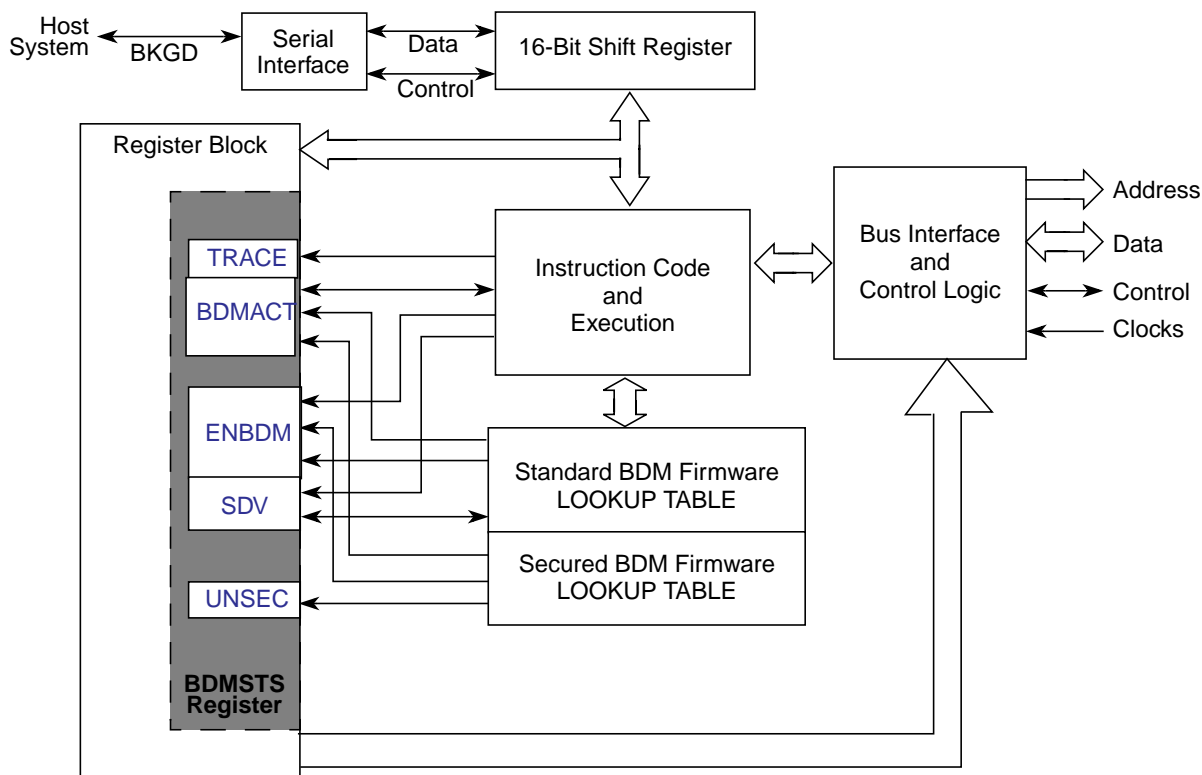


Figure 5-1. BDM Block Diagram

5.2 External Signal Description

A single-wire interface pin called the background debug interface (BKGD) pin is used to communicate with the BDM system. During reset, this pin is a mode select input which selects between normal and special modes of operation. After reset, this pin becomes the dedicated serial interface pin for the background debug mode. The communication rate of this pin is based on the settings for the VCO clock (CPMUSYNR). The BDM clock frequency is always VCO clock frequency divided by 8. After reset the BDM clock is based on the reset values of the CPMUSYNR register (4 MHz). When modifying the VCO

access causes a match. Thus if configured for a byte access of a particular address, a word access covering the same address does not lead to match.

Assuming the access direction is not qualified (RWE=0), for simplicity, the size access considerations are shown in [Table 6-33](#).

Table 6-33. Comparator B Access Size Considerations

Condition For Valid Match	Comp B Address	RWE	SZE	SZ8	Examples
Word and byte accesses of ADDR[n]	ADDR[n] ⁽¹⁾	0	0	X	MOVB #\$BYTE ADDR[n] MOVW #\$WORD ADDR[n]
Word accesses of ADDR[n] only	ADDR[n]	0	1	0	MOVW #\$WORD ADDR[n] LDD ADDR[n]
Byte accesses of ADDR[n] only	ADDR[n]	0	1	1	MOVB #\$BYTE ADDR[n] LDAB ADDR[n]

1. A word access of ADDR[n-1] also accesses ADDR[n] but does not generate a match.

The comparator address register must contain the exact address from the code.

Access direction can also be used to qualify a match for Comparator B in the same way as described for Comparator C in [Table 6-32](#).

6.4.2.1.3 Comparator A

Comparator A offers address, direction (R/W), access size (word/byte) and data bus comparison.

[Table 6-34](#) lists access considerations with data bus comparison. On word accesses the data byte of the lower address is mapped to DBGADH. Access direction can also be used to qualify a match for Comparator A in the same way as described for Comparator C in [Table 6-32](#).

Table 6-34. Comparator A Matches When Accessing ADDR[n]

SZE	SZ	DBGADHM, DBGADLM	Access DH=DBGADH, DL=DBGADL	Comment
0	X	\$0000	Byte Word	No databus comparison
0	X	\$FF00	Byte, data(ADDR[n])=DH Word, data(ADDR[n])=DH, data(ADDR[n+1])=X	Match data(ADDR[n])
0	X	\$00FF	Word, data(ADDR[n])=X, data(ADDR[n+1])=DL	Match data(ADDR[n+1])
0	X	\$00FF	Byte, data(ADDR[n])=X, data(ADDR[n+1])=DL	Possible unintended match
0	X	\$FFFF	Word, data(ADDR[n])=DH, data(ADDR[n+1])=DL	Match data(ADDR[n], ADDR[n+1])
0	X	\$FFFF	Byte, data(ADDR[n])=DH, data(ADDR[n+1])=DL	Possible unintended match
1	0	\$0000	Word	No databus comparison
1	0	\$00FF	Word, data(ADDR[n])=X, data(ADDR[n+1])=DL	Match only data at ADDR[n+1]
1	0	\$FF00	Word, data(ADDR[n])=DH, data(ADDR[n+1])=X	Match only data at ADDR[n]
1	0	\$FFFF	Word, data(ADDR[n])=DH, data(ADDR[n+1])=DL	Match data at ADDR[n] & ADDR[n+1]
1	1	\$0000	Byte	No databus comparison
1	1	\$FF00	Byte, data(ADDR[n])=DH	Match data at ADDR[n]

7.3.2.22 S12CPMU Protection Register (CPMUPROT)

This register protects the clock configuration registers from accidental overwrite:
CPMUSYNR, CPMUREFDIV, CPMUCLKS, CPMUPLL, CPMUIRCTRIMH/L and CPMUOSC

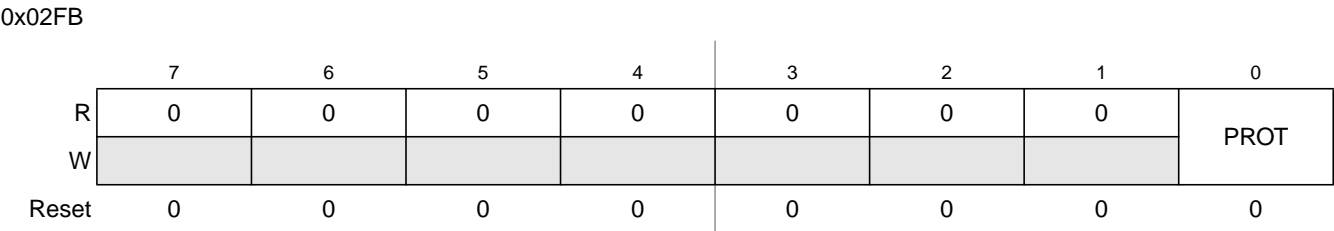


Figure 7-29. S12CPMU Protection Register (CPMUPROT)

Read: Anytime
Write: Anytime

Field	Description
0 PROT	Clock Configuration Registers Protection Bit — This bit protects the clock configuration registers from accidental overwrite (see list of affected registers above). Writing 0x26 to the CPMUPROT register clears the PROT bit, other write accesses set the PROT bit. 0 Protection of clock configuration registers is disabled. 1 Protection of clock configuration registers is enabled. CPMUSYNR, CPMUREFDIV, CPMUCLKS, CPMUPLL, CPMUIRCTRIMH/L and CPMUOSC registers are not writable.

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.

Read: always read 0x0000 in normal modes

Write: unimplemented in normal modes

NOTE

Writing to this register when in special modes can alter the PWM functionality.

10.3.2.8 Reserved Register (PWMPRSC)

This register is reserved for factory testing of the PWM module and is not available in normal modes.

Module Base + 0x0007

	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

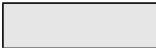
 = Unimplemented or Reserved

Figure 10-10. Reserved Register (PWMPRSC)

Read: always read 0x0000 in normal modes

Write: unimplemented in normal modes

NOTE

Writing to this register when in special modes can alter the PWM functionality.

10.3.2.9 PWM Scale A Register (PWMSCLA)

PWMSCLA is the programmable scale value used in scaling clock A to generate clock SA. Clock SA is generated by taking clock A, dividing it by the value in the PWMSCLA register and dividing that by two.

$$\text{Clock SA} = \text{Clock A} / (2 * \text{PWMSCLA})$$

NOTE

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

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

Module Base + 0x0008

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

Figure 10-11. PWM Scale A Register (PWMSCLA)

Module Base + 0x001D

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

Figure 10-32. PWM Channel Duty Registers (PWMDTY5)

Read: anytime

Write: anytime

10.3.2.15 PWM Shutdown Register (PWMSDN)

The PWMSDN register provides for the shutdown functionality of the PWM module in the emergency cases.

Module Base + 0x00E

	7	6	5	4	3	2	1	0
R	PWMIF	PWMIE	0	PWMLVL	0	PWM5IN	PWM5INL	PWM5ENA
W			PWMRSTRT					
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-33. PWM Shutdown Register (PWMSDN)

Read: anytime

Write: anytime

Table 10-10. PWMSDN Field Descriptions

Field	Description
7 PWMIF	PWM Interrupt Flag — Any change from passive to asserted (active) state or from active to passive state will be flagged by setting the PWMIF flag = 1. The flag is cleared by writing a logic 1 to it. Writing a 0 has no effect. 0 No change on PWM5IN input. 1 Change on PWM5IN input
6 PWMIE	PWM Interrupt Enable — If interrupt is enabled an interrupt to the CPU is asserted. 0 PWM interrupt is disabled. 1 PWM interrupt is enabled.
5 PWMRSTRT	PWM Restart — The PWM can only be restarted if the PWM channel input 5 is deasserted. After writing a logic 1 to the PWMRSTRT bit (trigger event) the PWM channels start running after the corresponding counter passes next “counter = 0” phase. Also, if the PWM5ENA bit is reset to 0, the PWM do not start before the counter passes 0x0000. The bit is always read as 0.
4 PWMLVL	PWM Shutdown Output Level — If active level as defined by the PWM5IN input, gets asserted all enabled PWM channels are immediately driven to the level defined by PWMLVL. 0 PWM outputs are forced to 0 1 PWM outputs are forced to 1.

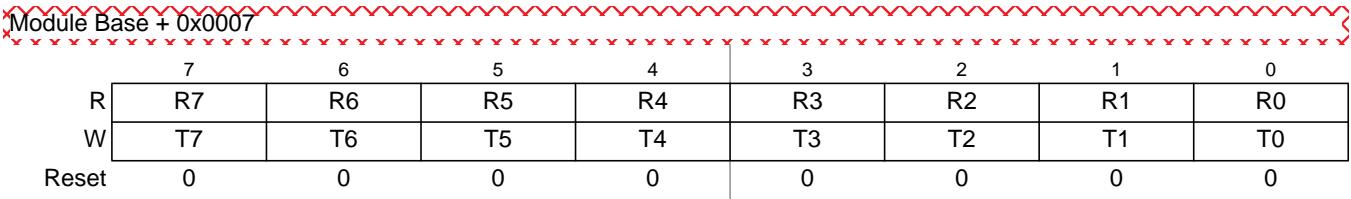


Figure 11-13. SCI Data Registers (SCIDRL)

Read: Anytime; reading accesses SCI receive data register
 Write: Anytime; writing accesses SCI transmit data register; writing to R8 has no effect

Table 11-13. SCIDRH and SCIDRL Field Descriptions

Field	Description
SCIDRH 7 R8	Received Bit 8 — R8 is the ninth data bit received when the SCI is configured for 9-bit data format (M = 1).
SCIDRH 6 T8	Transmit Bit 8 — T8 is the ninth data bit transmitted when the SCI is configured for 9-bit data format (M = 1).
SCIDRL 7:0 R[7:0] T[7:0]	R7:R0 — Received bits seven through zero for 9-bit or 8-bit data formats T7:T0 — Transmit bits seven through zero for 9-bit or 8-bit formats

NOTE

If the value of T8 is the same as in the previous transmission, T8 does not have to be rewritten.The same value is transmitted until T8 is rewritten

In 8-bit data format, only SCI data register low (SCIDRL) needs to be accessed.

When transmitting in 9-bit data format and using 8-bit write instructions, write first to SCI data register high (SCIDRH), then SCIDRL.

11.4 Functional Description

This section provides a complete functional description of the SCI block, detailing the operation of the design from the end user perspective in a number of subsections.

Figure 11-14 shows the structure of the SCI module. The SCI allows full duplex, asynchronous, serial communication between the CPU and remote devices, including other CPUs. The SCI transmitter and receiver operate independently, although they use the same baud rate generator. The CPU monitors the status of the SCI, writes the data to be transmitted, and processes received data.

Table 12-7. Example SPI Baud Rate Selection (25 MHz Bus Clock) (Sheet 2 of 3)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Baud Rate Divisor	Baud Rate
0	0	1	1	0	0	64	390.63 kbit/s
0	0	1	1	0	1	128	195.31 kbit/s
0	0	1	1	1	0	256	97.66 kbit/s
0	0	1	1	1	1	512	48.83 kbit/s
0	1	0	0	0	0	6	4.16667 Mbit/s
0	1	0	0	0	1	12	2.08333 Mbit/s
0	1	0	0	1	0	24	1.04167 Mbit/s
0	1	0	0	1	1	48	520.83 kbit/s
0	1	0	1	0	0	96	260.42 kbit/s
0	1	0	1	0	1	192	130.21 kbit/s
0	1	0	1	1	0	384	65.10 kbit/s
0	1	0	1	1	1	768	32.55 kbit/s
0	1	1	0	0	0	8	3.125 Mbit/s
0	1	1	0	0	1	16	1.5625 Mbit/s
0	1	1	0	1	0	32	781.25 kbit/s
0	1	1	0	1	1	64	390.63 kbit/s
0	1	1	1	0	0	128	195.31 kbit/s
0	1	1	1	0	1	256	97.66 kbit/s
0	1	1	1	1	0	512	48.83 kbit/s
0	1	1	1	1	1	1024	24.41 kbit/s
1	0	0	0	0	0	10	2.5 Mbit/s
1	0	0	0	0	1	20	1.25 Mbit/s
1	0	0	0	1	0	40	625 kbit/s
1	0	0	0	1	1	80	312.5 kbit/s
1	0	0	1	0	0	160	156.25 kbit/s
1	0	0	1	0	1	320	78.13 kbit/s
1	0	0	1	1	0	640	39.06 kbit/s
1	0	0	1	1	1	1280	19.53 kbit/s
1	0	1	0	0	0	12	2.08333 Mbit/s
1	0	1	0	0	1	24	1.04167 Mbit/s
1	0	1	0	1	0	48	520.83 kbit/s
1	0	1	0	1	1	96	260.42 kbit/s
1	0	1	1	0	0	192	130.21 kbit/s
1	0	1	1	0	1	384	65.10 kbit/s
1	0	1	1	1	0	768	32.55 kbit/s
1	0	1	1	1	1	1536	16.28 kbit/s
1	1	0	0	0	0	14	1.78571 Mbit/s
1	1	0	0	0	1	28	892.86 kbit/s
1	1	0	0	1	0	56	446.43 kbit/s
1	1	0	0	1	1	112	223.21 kbit/s
1	1	0	1	0	0	224	111.61 kbit/s
1	1	0	1	0	1	448	55.80 kbit/s

NOTE

Care must be taken when expecting data from a master while the slave is in wait or stop mode. Even though the shift register will continue to operate, the rest of the SPI is shut down (i.e., a SPIF interrupt will **not** be generated until exiting stop or wait mode). Also, the byte from the shift register will not be copied into the SPIDR register until after the slave SPI has exited wait or stop mode. In slave mode, a received byte pending in the receive shift register will be lost when entering wait or stop mode. An SPIF flag and SPIDR copy is generated only if wait mode is entered or exited during a transmission. If the slave enters wait mode in idle mode and exits wait mode in idle mode, neither a SPIF nor a SPIDR copy will occur.

12.4.7.3 SPI in Stop Mode

Stop mode is dependent on the system. The SPI enters stop mode when the module clock is disabled (held high or low). If the SPI is in master mode and exchanging data when the CPU enters stop mode, the transmission is frozen until the CPU exits stop mode. After stop, data to and from the external SPI is exchanged correctly. In slave mode, the SPI will stay synchronized with the master.

The stop mode is not dependent on the SPISWAI bit.

12.4.7.4 Reset

The reset values of registers and signals are described in [Section 12.3, “Memory Map and Register Definition”](#), which details the registers and their bit fields.

- If a data transmission occurs in slave mode after reset without a write to SPIDR, it will transmit garbage, or the data last received from the master before the reset.
- Reading from the SPIDR after reset will always read zeros.

12.4.7.5 Interrupts

The SPI only originates interrupt requests when SPI is enabled (SPE bit in SPICR1 set). The following is a description of how the SPI makes a request and how the MCU should acknowledge that request. The interrupt vector offset and interrupt priority are chip dependent.

The interrupt flags MODF, SPIF, and SPTEF are logically ORed to generate an interrupt request.

12.4.7.5.1 MODF

MODF occurs when the master detects an error on the \overline{SS} pin. The master SPI must be configured for the MODF feature (see [Table 12-3](#)). After MODF is set, the current transfer is aborted and the following bit is changed:

- MSTR = 0, The master bit in SPICR1 resets.

The MODF interrupt is reflected in the status register MODF flag. Clearing the flag will also clear the interrupt. This interrupt will stay active while the MODF flag is set. MODF has an automatic clearing process which is described in [Section 12.3.2.4, “SPI Status Register \(SPISR\)”](#).

phrase containing the Flash security byte during the reset sequence, all bits in the FSEC register will be set to leave the Flash module in a secured state with backdoor key access disabled.

Table 13-8. FSEC Field Descriptions

Field	Description
7–6 KEYEN[1:0]	Backdoor Key Security Enable Bits — The KEYEN[1:0] bits define the enabling of backdoor key access to the Flash module as shown in Table 13-9.
5–2 RNV[5:2]	Reserved Nonvolatile Bits — The RNV bits should remain in the erased state for future enhancements.
1–0 SEC[1:0]	Flash Security Bits — The SEC[1:0] bits define the security state of the MCU as shown in Table 13-10. If the Flash module is unsecured using backdoor key access, the SEC bits are forced to 10.

Table 13-9. Flash KEYEN States

KEYEN[1:0]	Status of Backdoor Key Access
00	DISABLED
01	DISABLED ⁽¹⁾
10	ENABLED
11	DISABLED

1. Preferred KEYEN state to disable backdoor key access.

Table 13-10. Flash Security States

SEC[1:0]	Status of Security
00	SECURED
01	SECURED ⁽¹⁾
10	UNSECURED
11	SECURED

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

The security function in the Flash module is described in Section 13.5.

13.3.2.3 Flash CCOB Index Register (FCCOBIX)

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

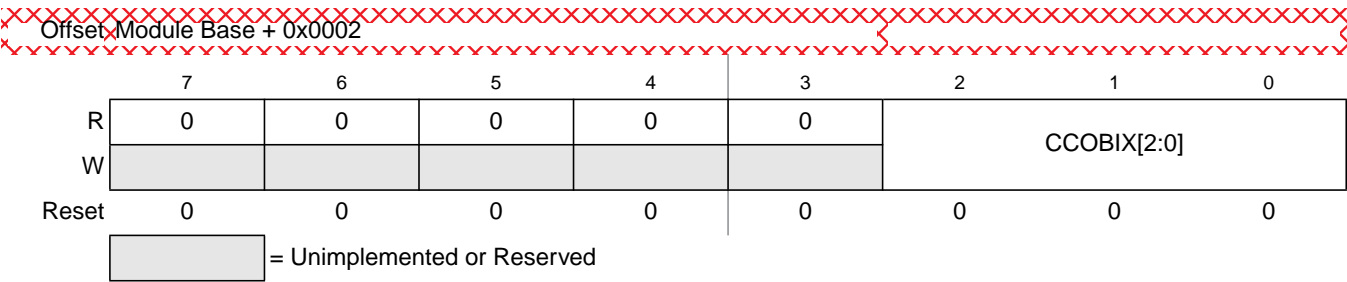


Figure 13-7. FCCOB Index Register (FCCOBIX)

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

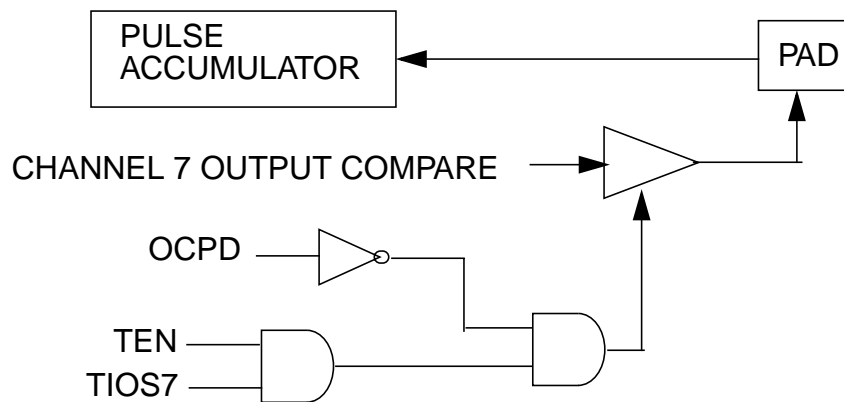


Figure 14-4. Channel 7 Output Compare/Pulse Accumulator Logic

14.2 External Signal Description

The TIM16B8CV2 module has a total of eight external pins.

14.2.1 IOC7 — Input Capture and Output Compare Channel 7 Pin

This pin serves as input capture or output compare for channel 7. This can also be configured as pulse accumulator input.

14.2.2 IOC6 — Input Capture and Output Compare Channel 6 Pin

This pin serves as input capture or output compare for channel 6.

14.2.3 IOC5 — Input Capture and Output Compare Channel 5 Pin

This pin serves as input capture or output compare for channel 5.

14.2.4 IOC4 — Input Capture and Output Compare Channel 4 Pin

This pin serves as input capture or output compare for channel 4. Pin

14.2.5 IOC3 — Input Capture and Output Compare Channel 3 Pin

This pin serves as input capture or output compare for channel 3.

14.2.6 IOC2 — Input Capture and Output Compare Channel 2 Pin

This pin serves as input capture or output compare for channel 2.

Write: Has no meaning or effect in the normal mode; only writable in special modes (test_mode = 1).
 The period of the first count after a write to the TCNT registers may be a different size because the write is not synchronized with the prescaler clock.

14.3.2.6 Timer System Control Register 1 (TSCR1)

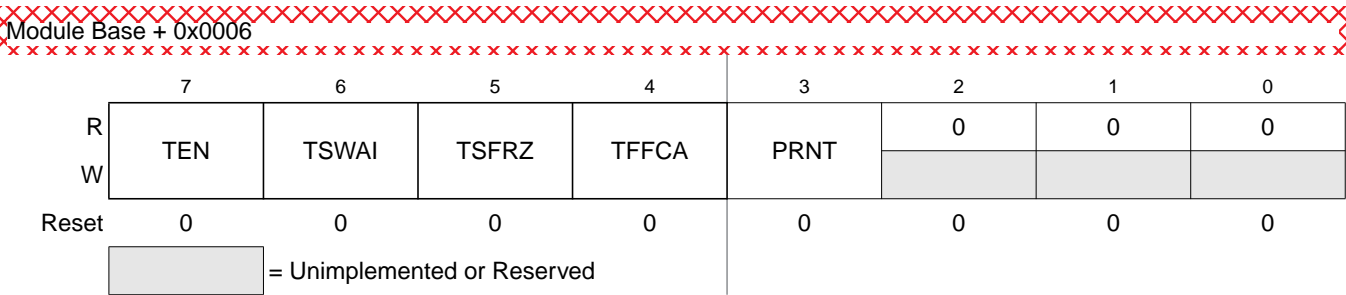


Figure 14-12. Timer System Control Register 1 (TSCR1)

Read: Anytime
 Write: Anytime

Table 14-6. TSCR1 Field Descriptions

Field	Description
7 TEN	Timer Enable 0 Disables the main timer, including the counter. Can be used for reducing power consumption. 1 Allows the timer to function normally. If for any reason the timer is not active, there is no +64 clock for the pulse accumulator because the +64 is generated by the timer prescaler.
6 TSWAI	Timer Module Stops While in Wait 0 Allows the timer module to continue running during wait. 1 Disables the timer module when the MCU is in the wait mode. Timer interrupts cannot be used to get the MCU out of wait. TSWAI also affects pulse accumulator.
5 TSFRZ	Timer Stops While in Freeze Mode 0 Allows the timer counter to continue running while in freeze mode. 1 Disables the timer counter whenever the MCU is in freeze mode. This is useful for emulation. TSFRZ does not stop the pulse accumulator.

Appendix A

Electrical Characteristics

A.1 General

This supplement contains the most accurate electrical information for the MC9S12P-Family microcontroller available at the time of publication.

This introduction is intended to give an overview on several common topics like power supply, current injection etc.

A.1.1 Parameter Classification

The electrical parameters shown in this supplement are guaranteed by various methods. To give the customer a better understanding the following classification is used and the parameters are tagged accordingly in the tables where appropriate.

NOTE

This classification is shown in the column labeled “C” in the parameter tables where appropriate.

- P: Those parameters are guaranteed during production testing on each individual device.
- C: Those parameters are achieved by the design characterization by measuring a statistically relevant sample size across process variations.
- T: Those parameters are achieved by design characterization on a small sample size from typical devices under typical conditions unless otherwise noted. All values shown in the typical column are within this category.
- D: Those parameters are derived mainly from simulations.

A.1.2 Power Supply

The VDDA, VSSA pin pairs supply the A/D converter and parts of the internal voltage regulator.

The VDDX, VSSX pin pairs [2:1] supply the I/O pins.

VDDR supplies the internal voltage regulator.

All VDDX pins are internally connected by metal.

All VSSX pins are internally connected by metal.

VDDA, VDDX and VSSA, VSSX are connected by diodes for ESD protection.

Table A-15. ATD Electrical Characteristics

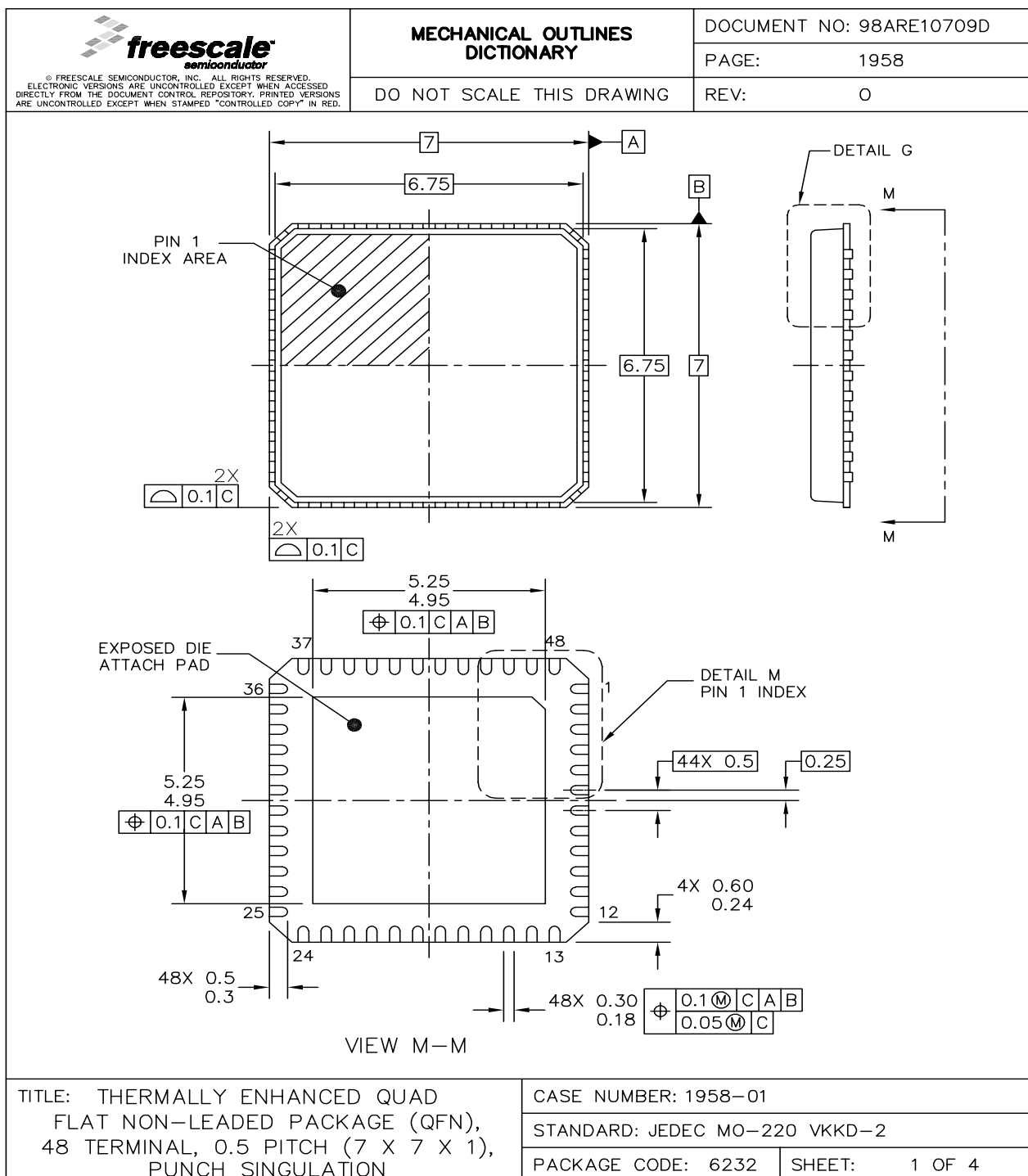
Conditions are shown in Table A-4 unless otherwise noted							
Num	C	Rating	Symbol	Min	Typ	Max	Unit
1	C	Max input source resistance ⁽¹⁾	R_S	—	—	1	$K\Omega$
2	D	Total input capacitance Non sampling	C_{INN}	—	—	10	pF
		Total input capacitance Sampling	C_{INS}	—	—	16	
3	D	Input internal Resistance	R_{INA}	-	5	15	$k\Omega$
4	C	Disruptive analog input current	I_{NA}	-2.5	—	2.5	mA
5	C	Coupling ratio positive current injection	K_p	—	—	1E-4	A/A
6	C	Coupling ratio negative current injection	K_n	—	—	5E-3	A/A

1. 1 Refer to [A.2.2.2](#) for further information concerning source resistance

A.2.3 ATD Accuracy

Table A-16. and Table A-17. specifies the ATD conversion performance excluding any errors due to current injection, input capacitance and source resistance.

C.2 48 QFN Package Mechanical Outline



0x0240 -0x027F Port Integration Module (PIM) Map 4 of 4

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0276	PER0AD0	R	PER0AD0	PER0AD0	PER0AD0	PER0AD0	PER0AD0	PER0AD0	PER0AD0	PER0AD0
		W	7	6	5	4	3	2	1	0
0x0277	PER1AD0	R	PER1AD0	PER1AD0	PER1AD0	PER1AD0	PER1AD0	PER1AD0	PER1AD0	PER1AD0
		W	7	6	5	4	3	2	1	0
0x0278-0x027F	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x0280-0x02EF Reserved

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0280-0x02EF	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x02F0-0x02FF Clock and Power Management Unit (CPMU) Map 2 of 2

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
0x02F0	CPMUHTCL	R	0	0	VSEL	0	HTEN	HTDS	HTIE	HTIF		
		W										
0x02F1	CPMULVCTL	R	0	0	0	0	0	LVDS	LVIE	LVIF		
		W										
0x02F2	CPMUAPICTL	R	APICLK	0	0	APIFES	APIEA	APIFE	APIE	APIF		
		W										
0x02F3	VREGAPITR	R	APITR5	APITR4	APITR3	APITR2	APITR1	APITR0	0	0		
		W										
0x02F4	CPMUAPIRH	R	APIR15	APIR14	APIR13	APIR12	APIR11	APIR10	APIR9	APIR8		
		W										
0x02F5	CPMUAPIRL	R	APIR7	APIR6	APIR5	APIR4	APIR3	APIR2	APIR1	APIR0		
		W										
0x02F6	Reserved	R	0	0	0	0	0	0	0	0		
		W										
0x02F7	CPMUHTTR	R	HTOEN	0	0	0	HTTR3	HTTR2	HTTR1	HTTR0		
		W										
0x02F8	CPMU IRCTRIMH	R	TCTRIM[3:0]				0	0	IRCTRIM[9:8]			
		W										
0x02F9	CPMU IRCTRIML	R	IRCTRIM[7:0]									
		W										
0x02FA	CPMUOSC	R	OSCE	OSCBW	0	OSCFILT[4:0]						
		W										
0x02FB	CPMUPROT	R	0	0	0	0	0	0	0	PROT		
		W										
0x02FC	Reserved	R	0	0	0	0	0	0	0	0		
		W										