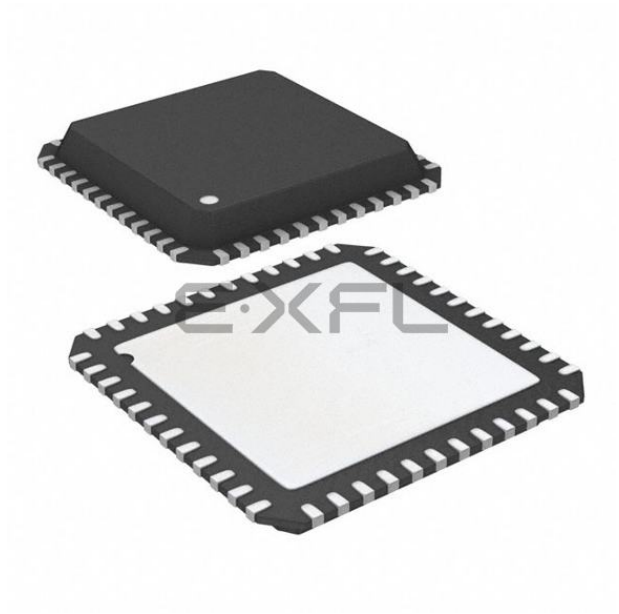


Welcome to E-XFL.COM

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	34
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	48-TFQFN Exposed Pad
Supplier Device Package	48-QFN-EP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12p64j0mft

Table 1-6. Port Availability by Package Option

Port	80 QFP	64 LQFP	48 QFN
Port T	8	8	8
Sum of Ports	**64**	**49**	**34**
I/O Power Pairs VDDX/VSSX	2/2	2/2	2/2

Table 1-7. Peripheral - Port Routing Options[1]

	PWM0	PWM4	PWM5
PT0	O		
PT4		O	
PT5			O

1. "O" denotes a possible rerouting under software control

Table 1-8 provides a pin out summary listing the availability and functionality of individual pins for each package option.

Table 1-8. Pin-Out Summary[1]

Package Pin			Function			Power Supply	Internal Pull Resistor		Description
QFP 80	LQFP 64	QFN 48	Pin	2nd Func.	3rd Func.		CTRL	Reset State	
53	41	30	PAD02	AN02	—	VDDA	PER1AD	Disabled	Port AD I/O, analog input of ATD
54	42	31	PAD03	AN03	—	VDDA	PER1AD	Disabled	Port AD I/O, analog input of ATD
55	43	32	PAD04	AN04	—	VDDA	PER1AD	Disabled	Port AD I/O, analog input of ATD
56	44	33	PAD05	AN05	—	VDDA	PER1AD	Disabled	Port AD I/O, analog input of ATD
57	45	34	PAD06	AN06	—	VDDA	PER1AD	Disabled	Port AD I/O, analog input of ATD
58	46	35	PAD07	AN07	—	VDDA	PER1AD	Disabled	Port AD I/O, analog input of ATD
59	47	36	VDDA	—	—	—	—	—	—
60	48	36	VRH[2]	—	—	—	—	—	—
61	49	37	VRL[3]	—	—	—	—	—	—
62	49	37	VSSA	—	—	—	—	—	—
63	50	38	PS0	RXD	—	VDDX	PERS/PPSS	Up	Port S I/O, RXD of SCI
64	51	39	PS1	TXD	—	VDDX	PERS/PPSS	Up	Port S I/O, TXD of SCI
65	52	-	PS2		—	VDDX	PERS/PPSS	Up	Port S I/O
66	53	-	PS3		—	VDDX	PERS/PPSS	Up	Port S I/O
67	54	40	TEST	—	—	N.A.	$\overline{\text{RESET}}$ pin	DOWN	Test input
68	-	-	PJ7	KWJ7	—	VDDX	PERJ/PPSJ	Up	Port J I/O, interrupt
69	-	-	PJ6	KWJ6	—	VDDX	PERJ/PPSJ	Up	Port J I/O, interrupt
70	55	41	PM5	SCK	—	VDDX	PERM/PPSM	Disabled	Port M I/O, MISO of SPI

Port E pin PE[0] can be used for either general purpose input or as the level-sensitive $\overline{\text{XIRQ}}$ interrupt. $\overline{\text{XIRQ}}$ can be enabled by clearing the X-bit in the CPU condition code register. It is inhibited at reset so this pin is initially configured as a high-impedance input with a pull-up.

2.4.3.4 Port T

This port is associated with TIM and PWM.

Port T pins PT[5:4,0] can be used for either general purpose I/O, or with the routed PWM or with the channels of the standard Timer subsystem.

Port T pins PT[7:6,3:1] can be used for either general purpose I/O, or with the channels of the standard Timer subsystem.

2.4.3.5 Port S

This port is associated with SCI.

Port S pins PS[1:0] can be used either for general purpose I/O, or with the SCI subsystem.

Port S pins PS[3:2] can be used for general purpose I/O.

2.4.3.6 Port M

This port is associated with CAN and SPI.

Port M pins PM[1:0] can be used for either general purpose I/O, or with the CAN subsystem.

Port M pins PM[5:2] can be used for general purpose I/O, or with the SPI subsystem.

2.4.3.7 Port P

This port is associated with the PWM.

Port P pins PP[7,5:0] can be used for either general purpose I/O with pin interrupt capability, or with the PWM subsystem.

2.4.3.8 Port J

Port J pins PJ[7:6,2:0] can be used for general purpose I/O with pin-interrupt capability.

2.4.3.9 Port AD

This port is associated with the ATD.

Port AD pins PAD[9:0] can be used for either general purpose I/O, or with the ATD subsystem.

2.4.4 Pin interrupts

Ports P and J offer pin interrupt capability. The interrupt enable as well as the sensitivity to rising or falling edges can be individually configured on per-pin basis. All bits/pins in a port share the same interrupt vector. Interrupts can be used with the pins configured as inputs or outputs.

An interrupt is generated when a bit in the port interrupt flag register and its corresponding port interrupt enable bit are both set. The pin interrupt feature is also capable to wake up the CPU when it is in STOP or WAIT mode.

A digital filter on each pin prevents pulses (Figure 2-66) shorter than a specified time from generating an interrupt. The minimum time varies over process conditions, temperature and voltage (Figure 2-65 and Table 2-60).

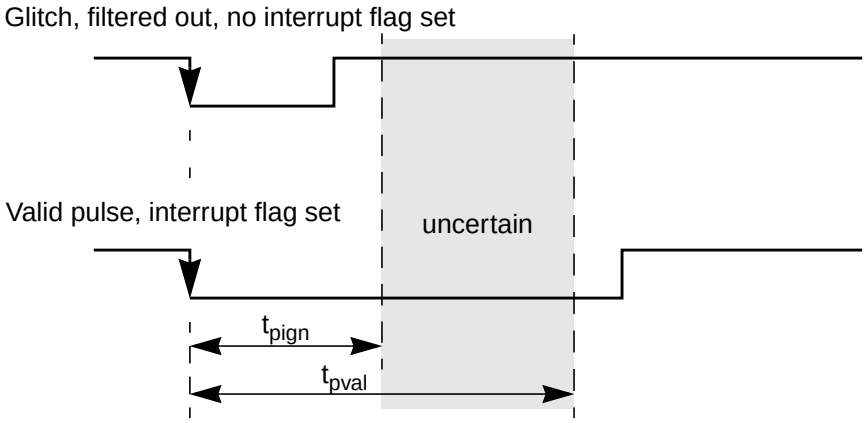


Figure 2-65. Interrupt Glitch Filter on Port P and J (PPS=0)

Table 2-60. Pulse Detection Criteria

Pulse	Mode		
	$\overline{STOP}$		STOP(1)
		Unit	
Ignored	$t_{pulse} \leq 3$	bus clocks	$t_{pulse} \leq t_{pign}$
Uncertain	$3 < t_{pulse} < 4$	bus clocks	$t_{pign} < t_{pulse} < t_{pval}$
Valid	$t_{pulse} \geq 4$	bus clocks	$t_{pulse} \geq t_{pval}$

1. These values include the spread of the oscillator frequency over temperature, voltage and process.

Global Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x3_FF05	Reserved	R	X	X	X	X	X	X	X	X
		W								
0x3_FF06	BDMCCR	R	CCR7	CCR6	CCR5	CCR4	CCR3	CCR2	CCR1	CCR0
		W								
0x3_FF07	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x3_FF08	BDMPPR	R	BPAE	0	0	0	BPP3	BPP2	BPP1	BPP0
		W								
0x3_FF09	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x3_FF0A	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x3_FF0B	Reserved	R	0	0	0	0	0	0	0	0
		W								

= Unimplemented, Reserved

= Implemented (do not alter)

X = Indeterminate

0 = Always read zero

Figure 5-2. BDM Register Summary (continued)

5.3.2.1 BDM Status Register (BDMSTS)

Register Global Address 0x3_FF01

	7	6	5	4	3	2	1	0
R	ENBDM	BDMACT	0	SDV	TRACE	0	UNSEC	0
W								
Reset								
Special Single-Chip Mode	0(1)	1	0	0	0	0	0(2)	0
All Other Modes	0	0	0	0	0	0	0	0

= Unimplemented, Reserved

= Implemented (do not alter)

0 = Always read zero

1. ENBDM is read as 1 by a debugging environment in special single chip mode when the device is not secured or secured but fully erased (Flash). This is because the ENBDM bit is set by the standard BDM firmware before a BDM command can be fully transmitted and executed.
2. UNSEC is read as 1 by a debugging environment in special single chip mode when the device is secured and fully erased, else it is 0 and can only be read if not secure (see also bit description).

Figure 5-3. BDM Status Register (BDMSTS)

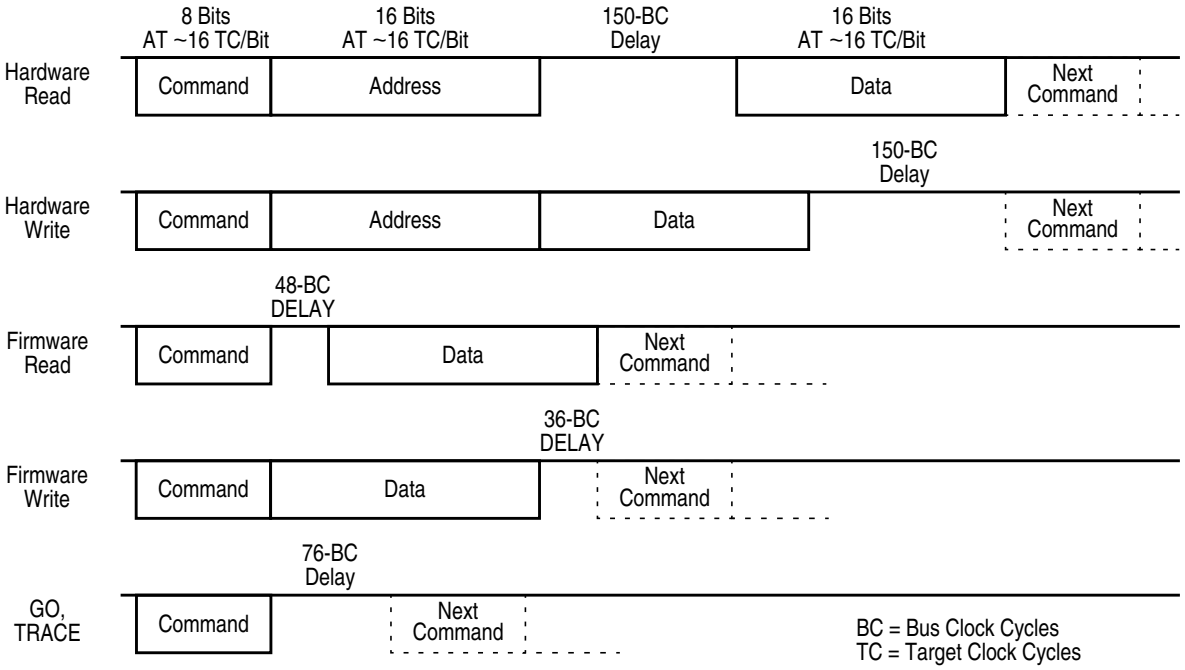


Figure 5-6. BDM Command Structure

5.4.6 BDM Serial Interface

The BDM communicates with external devices serially via the BKGD pin. 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 BDM.

The BDM serial interface is timed based on the VCO clock (please refer to the CPMU Block Guide for more details), which gets divided by 8. This clock will be referred to as the target clock in the following explanation.

The BDM serial interface uses a clocking scheme in which the external host generates a falling edge on the BKGD pin to indicate the start of each bit time. This falling edge is sent for every bit whether data is transmitted or received. Data is transferred most significant bit (MSB) first at 16 target clock cycles per bit. The interface times out if 512 clock cycles occur between falling edges from the host.

The BKGD pin is a pseudo open-drain pin and has an weak on-chip active pull-up that is enabled at all times. It is assumed that there is an external pull-up and that drivers connected to BKGD do not typically drive the high level. Since R-C rise time could be unacceptably long, the target system and host provide brief driven-high (speedup) pulses to drive BKGD to a logic 1. The source of this speedup pulse is the host for transmit cases and the target for receive cases.

The timing for host-to-target is shown in Figure 5-7 and that of target-to-host in Figure 5-8 and Figure 5-9. All four cases begin when the host drives the BKGD pin low to generate a falling edge. Since the host and target are operating from separate clocks, it can take the target system up to one full clock cycle to recognize this edge. The target measures delays from this perceived start of the bit time while the host measures delays from the point it actually drove BKGD low to start the bit up to one target clock cycle

Table 6-3. DBGC1 Field Descriptions

Field	Description
7 ARM	**Arm Bit** — The ARM bit controls whether the DBG module is armed. This bit can be set and cleared by user software and is automatically cleared on completion of a debug session, or if a breakpoint is generated with tracing not enabled. On setting this bit the state sequencer enters State1. 0 Debugger disarmed 1 Debugger armed
6 TRIG	**Immediate Trigger Request Bit** — This bit when written to 1 requests an immediate trigger independent of state sequencer status. When tracing is complete a forced breakpoint may be generated depending upon DBGBRK and BDM bit settings. This bit always reads back a 0. Writing a 0 to this bit has no effect. If the DBGTCR_TSOURCE bit is clear no tracing is carried out. If tracing has already commenced using BEGIN trigger alignment, it continues until the end of the tracing session as defined by the TALIGN bit, thus TRIG has no affect. In secure mode tracing is disabled and writing to this bit cannot initiate a tracing session. The session is ended by setting TRIG and ARM simultaneously. 0 Do not trigger until the state sequencer enters the Final State. 1 Trigger immediately
4 BDM	**Background Debug Mode Enable** — This bit determines if a breakpoint causes the system to enter Background Debug Mode (BDM) or initiate a Software Interrupt (SWI). If this bit is set but the BDM is not enabled by the ENBDM bit in the BDM module, then breakpoints default to SWI. 0 Breakpoint to Software Interrupt if BDM inactive. Otherwise no breakpoint. 1 Breakpoint to BDM, if BDM enabled. Otherwise breakpoint to SWI
3 DBGBRK	**S12SDBG Breakpoint Enable Bit** — The DBGBRK bit controls whether the debugger will request a breakpoint on reaching the state sequencer Final State. If tracing is enabled, the breakpoint is generated on completion of the tracing session. If tracing is not enabled, the breakpoint is generated immediately. 0 No Breakpoint generated 1 Breakpoint generated
1–0 COMRV	**Comparator Register Visibility Bits** — These bits determine which bank of comparator register is visible in the 8-byte window of the S12SDBG module address map, located between 0x0028 to 0x002F. Furthermore these bits determine which register is visible at the address 0x0027. See Table 6-4.

Table 6-4. COMRV Encoding

COMRV	Visible Comparator	Visible Register at 0x0027
00	Comparator A	DBGSCR1
01	Comparator B	DBGSCR2
10	Comparator C	DBGSCR3
11	None	DBGMFR

6.3.2.2 Debug Status Register (DBGSR)

Address: 0x0021

	7	6	5	4	3	2	1	0
R	TBF	0	0	0	0	SSF2	SSF1	SSF0
W								
Reset	—	0	0	0	0	0	0	0
POR	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 6-4. Debug Status Register (DBGSR)

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.

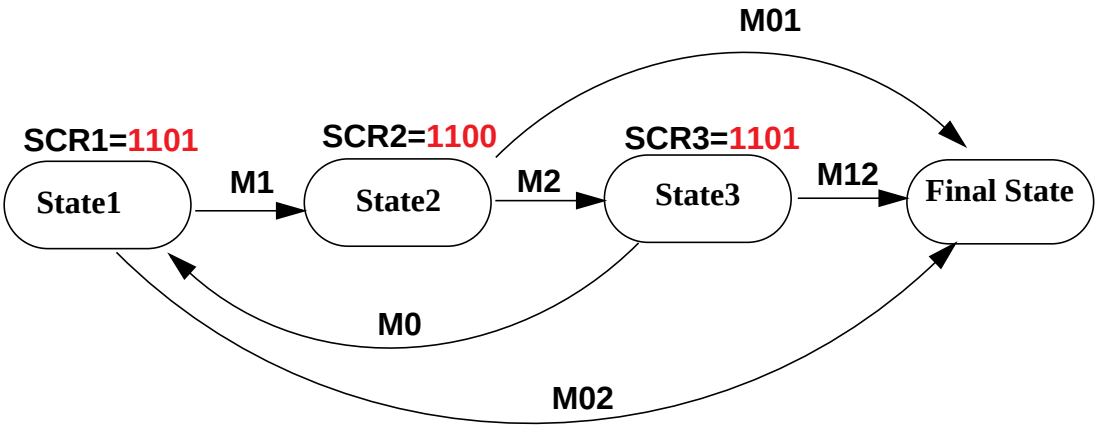
Table 6-34 lists access considerations with data bus comparison. On word accesses the data byte of the lower address is mapped to DBGADH. Access direction can also be used to qualify a match for Comparator A in the same way as described for Comparator C in Table 6-32.

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]

S12SDBGV1 SCR encoding because OR possibilities are very limited in the channel encoding. By adding OR forks as shown in red this scenario is possible.

Figure 6-36. Scenario 7



On simultaneous matches the lowest channel number has priority so with this configuration the forking from State1 has the peculiar effect that a simultaneous match0/match1 transitions to final state but a simultaneous match2/match1transitions to state2.

6.5.9 Scenario 8

Trigger when a routine/event at M2 follows either M1 or M0.

Figure 6-37. Scenario 8a

Trigger when an event M2 is followed by either event M0 or event M1

Figure 6-38. Scenario 8b

Scenario 8a and 8b are possible with the S12SDBGV1 and S12SDBGV2 SCR encoding

7.3.2.8 S12CPMU RTI Control Register (CPMURTI)

This register selects the time-out period for the Real Time Interrupt.

The clock source for the RTI is either IRCCLK or OSCCLK depending on the setting of the RTIOSCSEL bit. In Stop Mode with PSTP=1 (Pseudo Stop Mode) and RTIOSCSEL=1 the RTI continues to run, else the RTI counter halts in Stop Mode.

0x003B

	7	6	5	4	3	2	1	0
R W	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
Reset	0	0	0	0	0	0	0	0

Figure 7-11. S12CPMU RTI Control Register (CPMURTI)

Read: Anytime

Write: Anytime

NOTE

A write to this register starts the RTI time-out period. A change of the RTIOSCSEL bit (writing a different value or loosing UPOSC status) re-starts the RTI time-out period.

Table 7-8. CPMURTI Field Descriptions

Field	Description
7 RTDEC	**Decimal or Binary Divider Select Bit** — RTDEC selects decimal or binary based prescaler values. 0 Binary based divider value. See Table 7-9 1 Decimal based divider value. See Table 7-10
6–4 RTR[6:4]	**Real Time Interrupt Prescale Rate Select Bits** — These bits select the prescale rate for the RTI. See Table 7-9 and Table 7-10.
3–0 RTR[3:0]	**Real Time Interrupt Modulus Counter Select Bits** — These bits select the modulus counter target value to provide additional granularity.Table 7-9 and Table 7-10 show all possible divide values selectable by the CPMURTI register**.**

The use of the filter function is only possible if the VCOCLK-to-OSCCLK ratio divided by two ((f_{VCO} / f_{OSC})/2) is an integer number. This integer value must be written to the OSCFILT[4:0] bits.

If enabled, the oscillator filter is sampling the incoming oscillator clock signal (EXTAL) with the VCOCLK frequency.

Using VCOCLK, a time window is defined during which an edge of the OSCCLK is expected. In case of OSCBW = 1 the width of this window is three VCOCLK cycles, if the OSCBW = 0 it is one VCOCLK cycle.

The noise detection is active for certain combinations of OSCFILT[4:0] and OSCBW bit settings as shown in Table 7-24

Table 7-24. Noise Detection Settings

OSCFILT[4:0]	OSCBW	Detection	Filter
0	x	disabled	disabled
1	x	disabled	active
2 or 3	0	active	active
	1	disabled	active
>=4	x	active	active

NOTE

If the VCOCLK frequency is higher than 25 MHz the wide bandwidth must be selected (OSCBW = 1).

NOTE

The CANTBSEL register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK=1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 8-17. CANTBSEL Register Field Descriptions

Field	Description
2-0 TX[2:0]	**Transmit Buffer Select** — The lowest numbered bit places the respective transmit buffer in the CANTXFG register space (e.g., TX1 = 1 and TX0 = 1 selects transmit buffer TX0; TX1 = 1 and TX0 = 0 selects transmit buffer TX1). Read and write accesses to the selected transmit buffer will be blocked, if the corresponding TXEx bit is cleared and the buffer is scheduled for transmission (see Section 8.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)"). 0 The associated message buffer is deselected 1 The associated message buffer is selected, if lowest numbered bit

The following gives a short programming example of the usage of the CANTBSEL register:

To get the next available transmit buffer, application software must read the CANTFLG register and write this value back into the CANTBSEL register. In this example Tx buffers TX1 and TX2 are available. The value read from CANTFLG is therefore 0b0000_0110. When writing this value back to CANTBSEL, the Tx buffer TX1 is selected in the CANTXFG because the lowest numbered bit set to 1 is at bit position 1. Reading back this value out of CANTBSEL results in 0b0000_0010, because only the lowest numbered bit position set to 1 is presented. This mechanism eases the application software the selection of the next available Tx buffer.

- LDAA CANTFLG; value read is 0b0000_0110
- STAA CANTBSEL; value written is 0b0000_0110
- LDAA CANTBSEL; value read is 0b0000_0010

If all transmit message buffers are deselected, no accesses are allowed to the CANTXFG registers.

8.3.2.12 MSCAN Identifier Acceptance Control Register (CANIDAC)

The CANIDAC register is used for identifier acceptance control as described below.

Module Base + 0x000B — Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	0	0	IDAM1	IDAM0	0	IDHIT2	IDHIT1	IDHIT0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 8-15. MSCAN Identifier Acceptance Control Register (CANIDAC)

1. Read: Anytime
   Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1), except bits IDHITx, which are read-only

Module Base + 0x000C to Module Base + 0x000D — Access: User read/write(1)

	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

Figure 8-16. MSCAN Reserved Register

1. Read: Always reads zero in normal system operation modes
   Write: Unimplemented in normal system operation modes

NOTE

Writing to this register when in special systm operating modes can alter the MSCAN functionality.

8.3.2.14 MSCAN Miscellaneous Register (CANMISC)

This register provides additional features.

Module Base + 0x000D — Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	BOHOLD
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 8-17. MSCAN Miscellaneous Register (CANMISC)

1. Read: Anytime
   Write: Anytime; write of '1' clears flag; write of '0' ignored

Table 8-21. CANMISC Register Field Descriptions

Field	Description
0 BOHOLD	**Bus-off State Hold Until User Request** — If BORM is set in MSCAN Control Register 1 (CANCTL1), this bit indicates whether the module has entered the bus-off state. Clearing this bit requests the recovery from bus-off. Refer to Section 8.5.2, "Bus-Off Recovery," for details. 0 Module is not bus-off or recovery has been requested by user in bus-off state 1 Module is bus-off and holds this state until user request

8.3.2.15 MSCAN Receive Error Counter (CANRXERR)

This register reflects the status of the MSCAN receive error counter.

Table 8-36. 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 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 8.3.2.3, "MSCAN Bus Timing Register 0 (CANBTR0)" and Section 8.3.2.4, "MSCAN Bus Timing Register 1 (CANBTR1)").

Table 8-37 gives an overview of the CAN 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 8-37. CAN Standard 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

8.4.4 Modes of Operation

8.4.4.1 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.

Table 8-38. CPU vs. MSCAN Operating Modes

CPU Mode	MSCAN Mode			
	Normal	Reduced Power Consumption		
		Sleep	Power Down	Disabled (CANE=0)
RUN	CSWAI = X[1] SLPRQ = 0 SLPAK = 0	CSWAI = X SLPRQ = 1 SLPAK = 1		CSWAI = X SLPRQ = X SLPAK = X
WAIT	CSWAI = 0 SLPRQ = 0 SLPAK = 0	CSWAI = 0 SLPRQ = 1 SLPAK = 1	CSWAI = 1 SLPRQ = X SLPAK = X	CSWAI = X SLPRQ = X SLPAK = X
STOP			CSWAI = X SLPRQ = X SLPAK = X	CSWAI = X SLPRQ = X SLPAK = X

1. 'X' means don't care.

8.4.5.1 Operation in Run Mode

As shown in Table 8-38, only MSCAN sleep mode is available as low power option when the CPU is in run mode.

8.4.5.2 Operation in Wait Mode

The WAI instruction puts the MCU in a low power consumption stand-by mode. If the CSWAI bit is set, additional power can be saved in power down mode because the CPU clocks are stopped. After leaving this power down mode, the MSCAN restarts and enters normal mode again.

While the CPU is in wait mode, the MSCAN can be operated in normal mode and generate interrupts (registers can be accessed via background debug mode).

8.4.5.3 Operation in Stop Mode

The STOP instruction puts the MCU in a low power consumption stand-by mode. In stop mode, the MSCAN is set in power down mode regardless of the value of the SLPRQ/SLPAK and CSWAI bits (Table 8-38).

8.4.5.4 MSCAN Normal Mode

This is a non-power-saving mode. Enabling the MSCAN puts the module from disabled mode into normal mode. In this mode the module can either be in initialization mode or out of initialization mode. See Section 8.4.4.5, "MSCAN Initialization Mode".

9.2 Signal Description

This section lists all inputs to the ADC12B10C block.

9.2.1 Detailed Signal Descriptions

9.2.1.1 AN*x* (*x* = 9, 8, 7, 6, 5, 4, 3, 2, 1, 0)

This pin serves as the analog input Channel *x*. It can also be configured as digital port or external trigger for the ATD conversion.

9.2.1.2 ETRIG3, ETRIG2, ETRIG1, ETRIG0

These inputs can be configured to serve as an external trigger for the ATD conversion.

Refer to device specification for availability and connection of these inputs!

9.2.1.3 V_{RH}, V_{RL}

V_{RH} is the high reference voltage, V_{RL} is the low reference voltage for ATD conversion.

9.2.1.4 V_{DDA}, V_{SSA}

These pins are the power supplies for the analog circuitry of the ADC12B10C block.

9.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in the ADC12B10C.

9.3.1 Module Memory Map

Figure 9-2 gives an overview on all ADC12B10C registers.

NOTE

Register Address = Base Address + Address Offset, where the Base Address is defined at the MCU level and the Address Offset is defined at the module level.

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000	ATDCTL0	R W	Reserved	0	0	0	WRAP3	WRAP2	WRAP1	WRAP0
0x0001	ATDCTL1	R W	ETRIGSEL	SRES1	SRES0	SMP_DIS	ETRIGCH3	ETRIGCH2	ETRIGCH1	ETRIGCH0
0x0002	ATDCTL2	R W	0	AFFC	ICLKSTP	ETRIGLE	ETRIGP	ETRIGE	ASCIE	ACMPIE

= Unimplemented or Reserved

Figure 9-2. ADC12B10C Register Summary (Sheet 1 of 3)

Module Base + 0x0017

	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 10-26. PWM Channel Period Registers (PWMPER5)

Read: anytime

Write: anytime

10.3.2.14 PWM Channel Duty Registers (PWMDTYx)

There is a dedicated duty register for each channel. The value in this register determines the duty of the associated PWM channel. The duty value is compared to the counter and if it is equal to the counter value a match occurs and the output changes state.

The duty registers for each channel are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to 0x0000)
- The channel is disabled

In this way, the output of the PWM will always be either the old duty waveform or the new duty waveform, not some variation in between. If the channel is not enabled, then writes to the duty register will go directly to the latches as well as the buffer.

NOTE

Reads of this register return the most recent value written. Reads do not necessarily return the value of the currently active duty due to the double buffering scheme.

Reference Section 10.4.2.3, “PWM Period and Duty,” for more information.

NOTE

Depending on the polarity bit, the duty registers will contain the count of either the high time or the low time. If the polarity bit is 1, the output starts high and then goes low when the duty count is reached, so the duty registers contain a count of the high time. If the polarity bit is 0, the output starts low and then goes high when the duty count is reached, so the duty registers contain a count of the low time.

To calculate the output duty cycle (high time as a % of period) for a particular channel:

- Polarity = 0 (PPOLx = 0)
  Duty cycle = [(PWMPERx PWMDTYx)/PWMPERx] * 100%
- Polarity = 1 (PPOLx = 1)
  Duty cycle = [PWMDTYx / PWMPERx] * 100%

due to the synchronization of PWMEx and the clock source. An exception to this is when channels are concatenated. Refer to Section 10.4.2.7, “PWM 16-Bit Functions,” for more detail.

NOTE

The first PWM cycle after enabling the channel can be irregular.

On the front end of the PWM timer, the clock is enabled to the PWM circuit by the PWMEx bit being high. There is an edge-synchronizing circuit to guarantee that the clock will only be enabled or disabled at an edge. When the channel is disabled (PWMEx = 0), the counter for the channel does not count.

10.4.2.2 PWM Polarity

Each channel has a polarity bit to allow starting a waveform cycle with a high or low signal. This is shown on the block diagram as a mux select of either the Q output or the $\overline{Q}$ output of the PWM output flip-flop. When one of the bits in the PWMPOL register is set, the associated PWM channel output is high at the beginning of the waveform, then goes low when the duty count is reached. Conversely, if the polarity bit is 0, the output starts low and then goes high when the duty count is reached.

10.4.2.3 PWM Period and Duty

Dedicated period and duty registers exist for each channel and are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to 0x0000)
- The channel is disabled

In this way, the output of the PWM will always be either the old waveform or the new waveform, not some variation in between. If the channel is not enabled, then writes to the period and duty registers will go directly to the latches as well as the buffer.

A change in duty or period can be forced into effect “immediately” by writing the new value to the duty and/or period registers and then writing to the counter. This forces the counter to reset and the new duty and/or period values to be latched. In addition, because the counter is readable it is possible to know where the count is with respect to the duty value and software can be used to make adjustments.

NOTE

When forcing a new period or duty into effect immediately, an irregular PWM cycle can occur.

Depending on the polarity bit, the duty registers will contain the count of either the high time or the low time.

10.4.2.4 PWM Timer Counters

Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source (reference Figure 10-34 for the available clock sources and rates). The counter compares to two registers, a duty register and a period register as shown in Figure 10-35. When the PWM counter matches the duty register the output flip-flop changes state causing the PWM waveform to also change state. A match

drive the MOSI and SCK lines. In this case, the SPI immediately switches to slave mode, by clearing the MSTR bit and also disables the slave output buffer MISO (or SISO in bidirectional mode). So the result is that all outputs are disabled and SCK, MOSI, and MISO are inputs. If a transmission is in progress when the mode fault occurs, the transmission is aborted and the SPI is forced into idle state.

This mode fault error also sets the mode fault (MODF) flag in the SPI status register (SPISR). If the SPI interrupt enable bit (SPIE) is set when the MODF flag becomes set, then an SPI interrupt sequence is also requested.

When a write to the SPI data register in the master occurs, there is a half SCK-cycle delay. After the delay, SCK is started within the master. The rest of the transfer operation differs slightly, depending on the clock format specified by the SPI clock phase bit, CPHA, in SPI control register 1 (see Section 12.4.3, “Transmission Formats”).

NOTE

A change of the bits CPOL, CPHA, SSOE, LSBFE, XFRW, MODFEN, SPC0, or BIDIROE with SPC0 set, SPPR2-SPPR0 and SPR2-SPR0 in master mode will abort a transmission in progress and force the SPI into idle state. The remote slave cannot detect this, therefore the master must ensure that the remote slave is returned to idle state.

12.4.2 Slave Mode

The SPI operates in slave mode when the MSTR bit in SPI control register 1 is clear.

- Serial clock

  In slave mode, SCK is the SPI clock input from the master.

- MISO, MOSI pin

  In slave mode, the function of the serial data output pin (MISO) and serial data input pin (MOSI) is determined by the SPC0 bit and BIDIROE bit in SPI control register 2.

- $\overline{SS}$ pin

  The $\overline{SS}$ pin is the slave select input. Before a data transmission occurs, the $\overline{SS}$ pin of the slave SPI must be low. $\overline{SS}$ must remain low until the transmission is complete. If $\overline{SS}$ goes high, the SPI is forced into idle state.

  The $\overline{SS}$ input also controls the serial data output pin, if $\overline{SS}$ is high (not selected), the serial data output pin is high impedance, and, if $\overline{SS}$ is low, the first bit in the SPI data register is driven out of the serial data output pin. Also, if the slave is not selected ($\overline{SS}$ is high), then the SCK input is ignored and no internal shifting of the SPI shift register occurs.

  Although the SPI is capable of duplex operation, some SPI peripherals are capable of only receiving SPI data in a slave mode. For these simpler devices, there is no serial data out pin.

NOTE

When peripherals with duplex capability are used, take care not to simultaneously enable two receivers whose serial outputs drive the same system slave’s serial data output line.

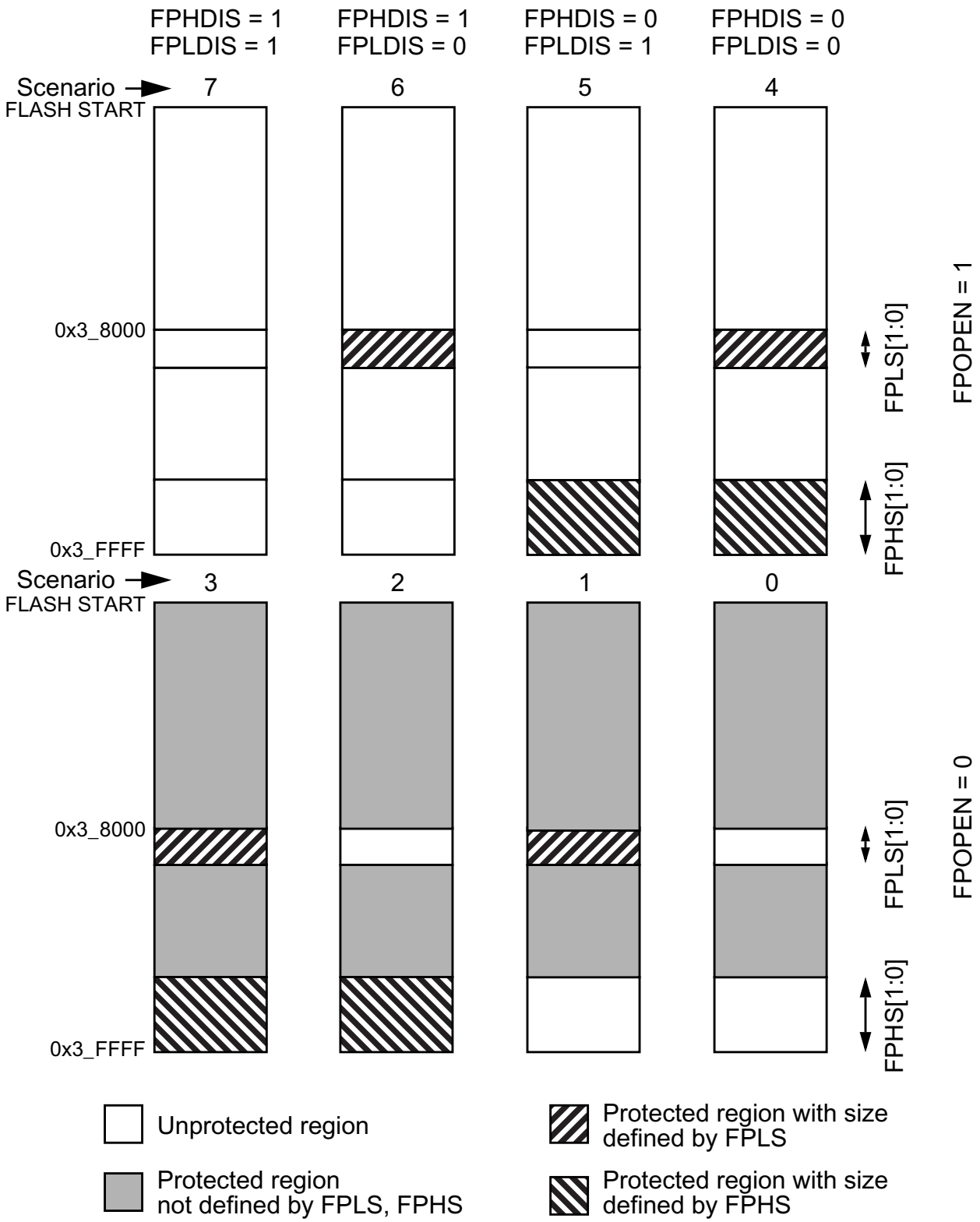


Figure 13-14. P-Flash Protection Scenarios