NXP USA Inc. - <u>S9S08DV32F2MLH Datasheet</u>





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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I ² C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	External
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/s9s08dv32f2mlh

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong



NOTE

To avoid extra current drain from floating input pins, the reset initialization routine in the application program should either enable on-chip pull-up devices or change the direction of unused or non-bonded pins to outputs so they do not float.

Table 4-14.	FSTAT	Register	Field	Descriptions	(continued)
-------------	-------	----------	-------	--------------	-------------

Field	Description
4 FACCERR	 Access Error Flag — FACCERR is set automatically when the proper command sequence is not obeyed exactly (the erroneous command is ignored), if a program or erase operation is attempted before the FCDIV register has been initialized, or if the MCU enters stop while a command was in progress. For a more detailed discussion of the exact actions that are considered access errors, see Section 4.5.6, "Access Errors." FACCERR is cleared by writing a 1 to FACCERR. Writing a 0 to FACCERR has no meaning or effect. 0 No access error. 1 An access error has occurred.
2 FBLANK	 Verified as All Blank (erased) Flag — FBLANK is set automatically at the conclusion of a blank check command if the entire Flash array was verified to be erased. FBLANK is cleared by clearing FCBEF to write a new valid command. Writing to FBLANK has no meaning or effect. O After a blank check command is completed and FCCF = 1, FBLANK = 0 indicates the Flash array is not completely erased. 1 After a blank check command is completed and FCCF = 1, FBLANK = 1 indicates the Flash array is completely erased (all 0xFFFF).

4.5.10.6 Flash Command Register (FCMD)

Only six command codes are recognized in normal user modes, as shown in Table 4-15. All other command codes are illegal and generate an access error. Refer to Section 4.5.3, "Program and Erase Command Execution," for a detailed discussion of Flash programming and erase operations.



Figure 4-10. Flash Command Register (FCMD)

Command	FCMD	Equate File Label
Blank check	0x05	mBlank
Byte program	0x20	mByteProg
Burst program	0x25	mBurstProg
Sector erase	0x40	mSectorErase
Mass erase	0x41	mMassErase
Sector erase abort	0x47	mEraseAbort

Table 4-15. Flash Commands

It is not necessary to perform a blank check command after a mass erase operation. Only blank check is required as part of the security unlocking mechanism.



Chapter 6 Parallel Input/Output Control

6.3.2 Edge and Level Sensitivity

A valid edge or level on an enabled port pin will set PTxIF in PTxSC. If PTxIE in PTxSC is set, an interrupt request will be presented to the CPU. Clearing of PTxIF is accomplished by writing a 1 to PTxACK in PTxSC provided all enabled port inputs are at their deasserted levels. PTxIF will remain set if any enabled port pin is asserted while attempting to clear by writing a 1 to PTxACK.

6.3.3 Pull-up/Pull-down Resistors

The port interrupt pins can be configured to use an internal pull-up/pull-down resistor using the associated I/O port pull-up enable register. If an internal resistor is enabled, the PTxES register is used to select whether the resistor is a pull-up (PTxESn = 0) or a pull-down (PTxESn = 1).

6.3.4 Pin Interrupt Initialization

When an interrupt pin is first enabled, it is possible to get a false interrupt flag. To prevent a false interrupt request during pin interrupt initialization, the user should do the following:

- 1. Mask interrupts by clearing PTxIE in PTxSC.
- 2. Select the pin polarity by setting the appropriate PTxESn bits in PTxES.
- 3. If using internal pull-up/pull-down device, configure the associated pull enable bits in PTxPE.
- 4. Enable the interrupt pins by setting the appropriate PTxPSn bits in PTxPS.
- 5. Write to PTxACK in PTxSC to clear any false interrupts.
- 6. Set PTxIE in PTxSC to enable interrupts.

6.4 Pin Behavior in Stop Modes

Pin behavior following execution of a STOP instruction depends on the stop mode that is entered. An explanation of pin behavior for the various stop modes follows:

- Stop2 mode is a partial power-down mode, whereby I/O latches are maintained in their state as before the STOP instruction was executed. CPU register status and the state of I/O registers should be saved in RAM before the STOP instruction is executed to place the MCU in stop2 mode. Upon recovery from stop2 mode, before accessing any I/O, the user should examine the state of the PPDF bit in the SPMSC2 register. If the PPDF bit is 0, I/O must be initialized as if a power on reset had occurred. If the PPDF bit is 1, peripherals may require initialization to be restored to their pre-stop condition. This can be done using data previously stored in RAM if it was saved before the STOP instruction was executed. The user must then write a 1 to the PPDACK bit in the SPMSC2 register. Access to I/O is now permitted again in the user application program.
- In stop3 mode, all I/O is maintained because internal logic circuity stays powered up. Upon recovery, normal I/O function is available to the user.



Source	Operation	dress ode Opi	Object Code	rcles	Cyc-by-Cyc	Affect on CCR	
i onn		βq V		රි	Details	V 1 1 H	INZC
CMP #opr8i CMP opr8a CMP opr16a CMP oprx16,X CMP oprx8,X CMP ,X CMP oprx16,SP CMP oprx8,SP	Compare Accumulator with Memory A – M (CCR Updated But Operands Not Changed)	IMM DIR EXT IX2 IX1 IX SP2 SP1	Al ii Bl dd Cl hh ll Dl ee ff El ff Fl 9E Dl ee ff 9E El ff	2 3 4 3 3 5 4	pp rpp prpp rpp rfp pprpp prpp	↓11 -	- ↓ ↓ ↓
COM opr8a COMA COMX COM oprx8,X COM ,X COM oprx8,SP	$\begin{array}{ll} \mbox{Complement} & \mbox{M} \leftarrow (\overline{M}) = \$ FF - (M) \\ (\mbox{One's Complement}) & \mbox{A} \leftarrow (\overline{A}) = \$ FF - (A) \\ & \mbox{X} \leftarrow (\overline{X}) = \$ FF - (X) \\ & \mbox{M} \leftarrow (\overline{M}) = \$ FF - (M) \\ & \mbox{M} \leftarrow (\overline{M}) = \$ FF - (M) \\ & \mbox{M} \leftarrow (\overline{M}) = \$ FF - (M) \end{array}$	DIR INH INH IX1 IX SP1	33 dd 43 53 63 ff 73 9E 63 ff	5 1 5 4 6	rfwpp p rfwpp rfwp prfwpp	011-	- ↓ ↓ 1
CPHX opr16a CPHX #opr16i CPHX opr8a CPHX oprx8,SP	Compare Index Register (H:X) with Memory (H:X) – (M:M + \$0001) (CCR Updated But Operands Not Changed)	EXT IMM DIR SP1	3E hh ll 65 jj kk 75 dd 9E F3 ff	6 3 5 6	prrfpp ppp rrfpp prrfpp	↓11-	- ↓ ↓ ↓
CPX #opr8i CPX opr8a CPX opr16a CPX oprx16,X CPX oprx8,X CPX ,X CPX oprx16,SP CPX oprx8,SP	Compare X (Index Register Low) with Memory X – M (CCR Updated But Operands Not Changed)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A3 ii B3 dd C3 hh ll D3 ee ff E3 ff F3 9E D3 ee ff 9E E3 ff	2 3 4 3 3 5 4	pp rpp prpp rpp rfp pprpp prpp	↓11-	- \$ \$ \$
DAA	Decimal Adjust Accumulator After ADD or ADC of BCD Values	INH	72	1	q	U 1 1 –	- ↓ ↓ ↓
DBNZ opr8a,rel DBNZA rel DBNZX rel DBNZ oprx8,X,rel DBNZ ,X,rel DBNZ oprx8,SP,rel	Decrement A, X, or M and Branch if Not Zero (if (result) \neq 0) DBNZX Affects X Not H	DIR INH INH IX1 IX SP1	3B dd rr 4B rr 5B rr 6B ff rr 7B rr 9E 6B ff rr	7 4 7 6 8	rfwpppp fppp fppp rfwpppp rfwppp prfwppp	- 1 1 -	
DEC opr8a DECA DECX DEC oprx8,X DEC ,X DEC oprx8,SP	$\begin{array}{llllllllllllllllllllllllllllllllllll$	DIR INH INH IX1 IX SP1	3A dd 4A 5A 6A ff 7A 9E 6A ff	5 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↓11-	- \$ \$ -
DIV	Divide $A \leftarrow (H:A) \div (X); H \leftarrow Remainder$	INH	52	6	fffffp	- 1 1 -	‡‡
EOR #opr8i EOR opr8a EOR opr16a EOR oprx16,X EOR oprx8,X EOR ,X EOR oprx16,SP EOR oprx8,SP	Exclusive OR Memory with Accumulator $A \leftarrow (A \oplus M)$	IMM DIR EXT IX2 IX1 IX SP2 SP1	A8 ii B8 dd C8 hh ll D8 ee ff E8 ff F8 9E D8 ee ff 9E E8 ff	2 3 4 3 3 5 4	pp rpp prpp rpp rfp pprpp prpp	011-	- ‡ ‡ -

Table 7-2. Instruction	Set	Summary	((Sheet	4	of	9))
------------------------	-----	---------	---	--------	---	----	----	---



Chapter 8 Multi-Purpose Clock Generator (S08MCGV1)

- LP bit is written to 1
- BDM mode is not active

In bypassed low power external mode, the MCGOUT clock is derived from the external reference clock. The external reference clock which is enabled can be an external crystal/resonator or it can be another external clock source.

The PLL and the FLL are disabled at all times in BLPE mode and the MCGLCLK will not be available for BDC communications. If the BDM becomes active the mode will switch to one of the bypassed external modes as determined by the state of the PLLS bit.

8.4.1.9 Stop

Stop mode is entered whenever the MCU enters a STOP state. In this mode, the FLL and PLL are disabled and all MCG clock signals are static except in the following cases:

MCGIRCLK will be active in stop mode when all the following conditions occur:

- IRCLKEN = 1
- IREFSTEN = 1

MCGERCLK will be active in stop mode when all the following conditions occur:

- ERCLKEN = 1
- EREFSTEN = 1

8.4.2 Mode Switching

When switching between engaged internal and engaged external modes the IREFS bit can be changed at anytime, but the RDIV bits must be changed simultaneously so that the reference frequency stays in the range required by the state of the PLLS bit (31.25 kHz to 39.0625 kHz if the FLL is selected, or 1 MHz to 2 MHz if the PLL is selected). After a change in the IREFS value the FLL or PLL will begin locking again after the switch is completed. The completion of the switch is shown by the IREFST bit .

For the special case of entering stop mode immediately after switching to FBE mode, if the external clock and the internal clock are disabled in stop mode, (EREFSTEN = 0 and IREFSTEN = 0), it is necessary to allow 100us after the IREFST bit is cleared to allow the internal reference to shutdown. For most cases the delay due to instruction execution times will be sufficient.

The CLKS bits can also be changed at anytime, but in order for the MCGLCLK to be configured correctly the RDIV bits must be changed simultaneously so that the reference frequency stays in the range required by the state of the PLLS bit (31.25 kHz to 39.0625 kHz if the FLL is selected, or 1 MHz to 2MHz if the PLL is selected). The actual switch to the newly selected clock will be shown by the CLKST bits. If the newly selected clock is not available, the previous clock will remain selected.

For details see Figure 8-8.



Chapter 8 Multi-Purpose Clock Generator (S08MCGV1)



Figure 8-9. Flowchart of FEI to PEE Mode Transition using a 4 MHz crystal



```
Chapter 10 Analog-to-Digital Converter (S08ADC12V1)
```

ALTCLK for this MCU. Consult the module introduction for information on ALTCLK specific to this MCU.

A conversion complete event sets the COCO and generates an ADC interrupt to wake the MCU from wait mode if the ADC interrupt is enabled (AIEN = 1).

10.4.7 MCU Stop3 Mode Operation

Stop mode is a low power-consumption standby mode during which most or all clock sources on the MCU are disabled.

10.4.7.1 Stop3 Mode With ADACK Disabled

If the asynchronous clock, ADACK, is not selected as the conversion clock, executing a stop instruction aborts the current conversion and places the ADC in its idle state. The contents of ADCRH and ADCRL are unaffected by stop3 mode. After exiting from stop3 mode, a software or hardware trigger is required to resume conversions.

10.4.7.2 Stop3 Mode With ADACK Enabled

If ADACK is selected as the conversion clock, the ADC continues operation during stop3 mode. For guaranteed ADC operation, the MCU's voltage regulator must remain active during stop3 mode. Consult the module introduction for configuration information for this MCU.

If a conversion is in progress when the MCU enters stop3 mode, it continues until completion. Conversions can be initiated while the MCU is in stop3 mode by means of the hardware trigger or if continuous conversions are enabled.

A conversion complete event sets the COCO and generates an ADC interrupt to wake the MCU from stop3 mode if the ADC interrupt is enabled (AIEN = 1).

NOTE

The ADC module can wake the system from low-power stop and cause the MCU to begin consuming run-level currents without generating a system level interrupt. To prevent this scenario, software should ensure the data transfer blocking mechanism (discussed in Section 10.4.4.2, "Completing Conversions) is cleared when entering stop3 and continuing ADC conversions.

10.4.8 MCU Stop2 Mode Operation

The ADC module is automatically disabled when the MCU enters stop2 mode. All module registers contain their reset values following exit from stop2. Therefore, the module must be re-enabled and re-configured following exit from stop2.



Chapter 11 Inter-Integrated Circuit (S08IICV2)

Refer to the direct-page register summary in the memory chapter of this document for the absolute address assignments for all IIC registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

11.3.1 IIC Address Register (IICA)



Figure 11-3. IIC Address Register (IICA)

Table	11-1.	IICA	Field	Descriptions
-------	-------	------	-------	--------------

Field	Description
7–1 AD[7:1]	Slave Address. The AD[7:1] field contains the slave address to be used by the IIC module. This field is used on the 7-bit address scheme and the lower seven bits of the 10-bit address scheme.

11.3.2 IIC Frequency Divider Register (IICF)



Figure 11-4. IIC Frequency Divider Register (IICF)



Chapter 11 Inter-Integrated Circuit (S08IICV2)

11.4.3 General Call Address

General calls can be requested in 7-bit address or 10-bit address. If the GCAEN bit is set, the IIC matches the general call address as well as its own slave address. When the IIC responds to a general call, it acts as a slave-receiver and the IAAS bit is set after the address cycle. Software must read the IICD register after the first byte transfer to determine whether the address matches is its own slave address or a general call. If the value is 00, the match is a general call. If the GCAEN bit is clear, the IIC ignores any data supplied from a general call address by not issuing an acknowledgement.

11.5 Resets

The IIC is disabled after reset. The IIC cannot cause an MCU reset.

11.6 Interrupts

The IIC generates a single interrupt.

An interrupt from the IIC is generated when any of the events in Table 11-11 occur, provided the IICIE bit is set. The interrupt is driven by bit IICIF (of the IIC status register) and masked with bit IICIE (of the IIC control register). The IICIF bit must be cleared by software by writing a 1 to it in the interrupt routine. You can determine the interrupt type by reading the status register.

Interrupt Source	Status	Flag	Local Enable
Complete 1-byte transfer	TCF	IICIF	IICIE
Match of received calling address	IAAS	licif	IICIE
Arbitration Lost	ARBL	IICIF	IICIE

Table 11-11. Interrupt Summary

11.6.1 Byte Transfer Interrupt

The TCF (transfer complete flag) bit is set at the falling edge of the ninth clock to indicate the completion of byte transfer.

11.6.2 Address Detect Interrupt

When the calling address matches the programmed slave address (IIC address register) or when the GCAEN bit is set and a general call is received, the IAAS bit in the status register is set. The CPU is interrupted, provided the IICIE is set. The CPU must check the SRW bit and set its Tx mode accordingly.

11.6.3 Arbitration Lost Interrupt

The IIC is a true multi-master bus that allows more than one master to be connected on it. If two or more masters try to control the bus at the same time, the relative priority of the contending masters is determined by a data arbitration procedure. The IIC module asserts this interrupt when it loses the data arbitration process and the ARBL bit in the status register is set.

Chapter 12 Freescale Controller Area Network (S08MSCANV1)









Figure 12-10. MSCAN Transmitter Flag Register (CANTFLG)

NOTE

The CANTFLG 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).

Read: Anytime

Write: Anytime for TXEx flags when not in initialization mode; write of 1 clears flag, write of 0 is ignored

Field	Description
2:0 TXE[2:0]	Transmitter Buffer Empty — This flag indicates that the associated transmit message buffer is empty, and thus not scheduled for transmission. The CPU must clear the flag after a message is set up in the transmit buffer and is due for transmission. The MSCAN sets the flag after the message is sent successfully. The flag is also set by the MSCAN when the transmission request is successfully aborted due to a pending abort request (see Section 12.3.8, "MSCAN Transmitter Message Abort Request Register (CANTARQ)"). If not masked, a transmit interrupt is pending while this flag is set. Clearing a TXEx flag also clears the corresponding ABTAKx (see Section 12.3.9, "MSCAN Transmitter Message Abort Acknowledge Register (CANTAAK)"). When a TXEx flag is set, the corresponding ABTRQx bit is cleared (see Section 12.3.8, "MSCAN Transmitter Message Abort Request Register (CANTARQ)"). When listen-mode is active (see Section 12.3.2, "MSCAN Control Register 1 (CANCTL1)") the TXEx flags cannot be cleared and no transmission is started. Read and write accesses to the transmit buffer are blocked, if the corresponding TXEx bit is cleared (TXEx = 0) and the buffer is scheduled for transmission. 0 The associated message buffer is full (loaded with a message due for transmission) 1 The associated message buffer is empty (not scheduled)

Table 12-11. CANTFLG Register Field Descriptions

12.3.7 MSCAN Transmitter Interrupt Enable Register (CANTIER)

This register contains the interrupt enable bits for the transmit buffer empty interrupt flags.







Table 12-30. IDR1	Register Field	Descriptions
-------------------	-----------------------	--------------

Field	Description								
7:5 ID[2:0]	Standard Format Identifier — The identifiers consist of 11 bits (ID[10:0]) for the standard format. ID10 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number. See also ID bits in Table 12-29.								
4 RTR	 Remote Transmission Request — This flag reflects the status of the Remote Transmission Request bit in the CAN frame. In the case of a receive buffer, it indicates the status of the received frame and supports the transmission of an answering frame in software. In the case of a transmit buffer, this flag defines the setting of the RTR bit to be sent. 0 Data frame 1 Remote frame 								
3 IDE	ID Extended — This flag indicates whether the extended or standard identifier format is applied in this buffer. In the case of a receive buffer, the flag is set as received and indicates to the CPU how to process the buffer identifier registers. In the case of a transmit buffer, the flag indicates to the MSCAN what type of identifier to send. 0 Standard format (11 bit) 1 Extended format (29 bit)								
R	7	6	5	4	3	2	1	0	
vv									



12.4.3 Data Segment Registers (DSR0-7)

The eight data segment registers, each with bits DB[7:0], contain the data to be transmitted or received. The number of bytes to be transmitted or received is determined by the data length code in the corresponding DLR register.



Table 13-7. SPIS Register Field Descriptions

Field	Description					
7 SPRF	 SPI Read Buffer Full Flag — SPRF is set at the completion of an SPI transfer to indicate that received data may be read from the SPI data register (SPID). SPRF is cleared by reading SPRF while it is set, then reading the SPI data register. No data available in the receive data buffer Data available in the receive data buffer 					
5 SPTEF	SPI Transmit Buffer Empty Flag — This bit is set when there is room in the transmit data buffer. It is cleared by reading SPIS with SPTEF set, followed by writing a data value to the transmit buffer at SPID. SPIS must be read with SPTEF = 1 before writing data to SPID or the SPID write will be ignored. SPTEF generates an SPTEF CPU interrupt request if the SPTIE bit in the SPIC1 is also set. SPTEF is automatically set when a data byte transfers from the transmit buffer into the transmit shift register. For an idle SPI (no data in the transmit buffer or the shift register and no transfer in progress), data written to SPID is transferred to the shifter almost immediately so SPTEF is set within two bus cycles allowing a second 8-bit data value to be queued into the transmit buffer. After completion of the transfer of the value in the shift register, the queued value from the transmit buffer will automatically move to the shifter and SPTEF will be set to indicate there is room for new data in the transmit buffer not empty at a second 8-DTEF simply remains set and no data moves from the buffer to the shifter. If no new data is waiting in the transmit buffer, SPTEF simply remains set and no data moves from the buffer to the shifter. SPTEF simply remains buffer not empty 1 SPI transmit buffer empty					
4 MODF	Master Mode Fault Flag — MODF is set if the SPI is configured as a master and the slave select input goes low, indicating some other SPI device is also configured as a master. The SS pin acts as a mode fault error input only when MSTR = 1, MODFEN = 1, and SSOE = 0; otherwise, MODF will never be set. MODF is cleared by reading MODF while it is 1, then writing to SPI control register 1 (SPIC1). 0 No mode fault error 1 Mode fault error detected					

13.4.5 SPI Data Register (SPID)

	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 13-9. SPI Data Register (SPID)

Reads of this register return the data read from the receive data buffer. Writes to this register write data to the transmit data buffer. When the SPI is configured as a master, writing data to the transmit data buffer initiates an SPI transfer.

Data should not be written to the transmit data buffer unless the SPI transmit buffer empty flag (SPTEF) is set, indicating there is room in the transmit buffer to queue a new transmit byte.

Data may be read from SPID any time after SPRF is set and before another transfer is finished. Failure to read the data out of the receive data buffer before a new transfer ends causes a receive overrun condition and the data from the new transfer is lost.



Chapter 14 Serial Communications Interface (S08SCIV4)

flag is set. If RDRF was already set indicating the receive data register (buffer) was already full, the overrun (OR) status flag is set and the new data is lost. Because the SCI receiver is double-buffered, the program has one full character time after RDRF is set before the data in the receive data buffer must be read to avoid a receiver overrun.

When a program detects that the receive data register is full (RDRF = 1), it gets the data from the receive data register by reading SCIxD. The RDRF flag is cleared automatically by a 2-step sequence which is normally satisfied in the course of the user's program that handles receive data. Refer to Section 14.3.4, "Interrupts and Status Flags" for more details about flag clearing.

14.3.3.1 Data Sampling Technique

The SCI receiver uses a 16× baud rate clock for sampling. The receiver starts by taking logic level samples at 16 times the baud rate to search for a falling edge on the RxD serial data input pin. A falling edge is defined as a logic 0 sample after three consecutive logic 1 samples. The 16× baud rate clock is used to divide the bit time into 16 segments labeled RT1 through RT16. When a falling edge is located, three more samples are taken at RT3, RT5, and RT7 to make sure this was a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes it is synchronized to a receive character.

The receiver then samples each bit time, including the start and stop bits, at RT8, RT9, and RT10 to determine the logic level for that bit. The logic level is interpreted to be that of the majority of the samples taken during the bit time. In the case of the start bit, the bit is assumed to be 0 if at least two of the samples at RT3, RT5, and RT7 are 0 even if one or all of the samples taken at RT8, RT9, and RT10 are 1s. If any sample in any bit time (including the start and stop bits) in a character frame fails to agree with the logic level for that bit, the noise flag (NF) will be set when the received character is transferred to the receive data buffer.

The falling edge detection logic continuously looks for falling edges, and if an edge is detected, the sample clock is resynchronized to bit times. This improves the reliability of the receiver in the presence of noise or mismatched baud rates. It does not improve worst case analysis because some characters do not have any extra falling edges anywhere in the character frame.

In the case of a framing error, provided the received character was not a break character, the sampling logic that searches for a falling edge is filled with three logic 1 samples so that a new start bit can be detected almost immediately.

In the case of a framing error, the receiver is inhibited from receiving any new characters until the framing error flag is cleared. The receive shift register continues to function, but a complete character cannot transfer to the receive data buffer if FE is still set.

14.3.3.2 Receiver Wakeup Operation

Receiver wakeup is a hardware mechanism that allows an SCI receiver to ignore the characters in a message that is intended for a different SCI receiver. In such a system, all receivers evaluate the first character(s) of each message, and as soon as they determine the message is intended for a different receiver, they write logic 1 to the receiver wake up (RWU) control bit in SCIxC2. When RWU bit is set, the status flags associated with the receiver (with the exception of the idle bit, IDLE, when RWUID bit is set) are inhibited from setting, thus eliminating the software overhead for handling the unimportant







In input capture mode, reading either byte (TPMxCnVH or TPMxCnVL) latches the contents of both bytes into a buffer where they remain latched until the other half is read. This latching mechanism also resets (becomes unlatched) when the TPMxCnSC register is written (whether BDM mode is active or not). Any write to the channel registers will be ignored during the input capture mode.

When BDM is active, the coherency mechanism is frozen (unless reset by writing to TPMxCnSC register) such that the buffer latches remain in the state they were in when the BDM became active, even if one or both halves of the channel register are read while BDM is active. This assures that if the user was in the middle of reading a 16-bit register when BDM became active, it will read the appropriate value from the other half of the 16-bit value after returning to normal execution. The value read from the TPMxCnVH and TPMxCnVL registers in BDM mode is the value of these registers and not the value of their read buffer.

In output compare or PWM modes, writing to either byte (TPMxCnVH or TPMxCnVL) latches the value into a buffer. After both bytes are written, they are transferred as a coherent 16-bit value into the timer-channel registers according to the value of CLKSB:CLKSA bits and the selected mode, so:

- If (CLKSB:CLKSA = 0:0), then the registers are updated when the second byte is written.
- If (CLKSB:CLKSA not = 0:0 and in output compare mode) then the registers are updated after the second byte is written and on the next change of the TPM counter (end of the prescaler counting).
- If (CLKSB:CLKSA not = 0:0 and in EPWM or CPWM modes), then the registers are updated after the both bytes were written, and the TPM counter changes from (TPMxMODH:TPMxMODL 1) to (TPMxMODH:TPMxMODL). If the TPM counter is a free-running counter then the update is made when the TPM counter changes from 0xFFFE to 0xFFFF.

The latching mechanism may be manually reset by writing to the TPMxCnSC register (whether BDM mode is active or not). This latching mechanism allows coherent 16-bit writes in either big-endian or little-endian order which is friendly to various compiler implementations.

When BDM is active, the coherency mechanism is frozen such that the buffer latches remain in the state they were in when the BDM became active even if one or both halves of the channel register are written while BDM is active. Any write to the channel registers bypasses the buffer latches and directly write to the channel register while BDM is active. The values written to the channel register while BDM is active



Chapter 17 Development Support

A-Only — Trigger when the address matches the value in comparator A

A OR B — Trigger when the address matches either the value in comparator A or the value in comparator B

A Then B — Trigger when the address matches the value in comparator B but only after the address for another cycle matched the value in comparator A. There can be any number of cycles after the A match and before the B match.

A AND B Data (Full Mode) — This is called a full mode because address, data, and R/W (optionally) must match within the same bus cycle to cause a trigger event. Comparator A checks address, the low byte of comparator B checks data, and R/W is checked against RWA if RWAEN = 1. The high-order half of comparator B is not used.

In full trigger modes it is not useful to specify a tag-type CPU breakpoint (BRKEN = TAG = 1), but if you do, the comparator B data match is ignored for the purpose of issuing the tag request to the CPU and the CPU breakpoint is issued when the comparator A address matches.

A AND NOT B Data (Full Mode) — Address must match comparator A, data must not match the low half of comparator B, and R/W must match RWA if RWAEN = 1. All three conditions must be met within the same bus cycle to cause a trigger.

In full trigger modes it is not useful to specify a tag-type CPU breakpoint (BRKEN = TAG = 1), but if you do, the comparator B data match is ignored for the purpose of issuing the tag request to the CPU and the CPU breakpoint is issued when the comparator A address matches.

Event-Only B (Store Data) — Trigger events occur each time the address matches the value in comparator B. Trigger events cause the data to be captured into the FIFO. The debug run ends when the FIFO becomes full.

A Then Event-Only B (Store Data) — After the address has matched the value in comparator A, a trigger event occurs each time the address matches the value in comparator B. Trigger events cause the data to be captured into the FIFO. The debug run ends when the FIFO becomes full.

Inside Range ($A \le Address \le B$) — A trigger occurs when the address is greater than or equal to the value in comparator A and less than or equal to the value in comparator B at the same time.

Outside Range (Address < A or Address > B) — A trigger occurs when the address is either less than the value in comparator A or greater than the value in comparator B.



Chapter 17 Development Support

17.4.1.1 BDC Status and Control Register (BDCSCR)

This register can be read or written by serial BDC commands (READ_STATUS and WRITE_CONTROL) but is not accessible to user programs because it is not located in the normal memory map of the MCU.



= Unimplemented or Reserved

Figure 17-5. BDC Status and Control Register (BDCSCR)

Table 17-2. BDCSCR Register Field Descriptions

Field	Description					
7 ENBDM	 Enable BDM (Permit Active Background Mode) — Typically, this bit is written to 1 by the debug host shortly after the beginning of a debug session or whenever the debug host resets the target and remains 1 until a normal reset clears it. 0 BDM cannot be made active (non-intrusive commands still allowed) 1 BDM can be made active to allow active background mode commands 					
6 BDMACT	 Background Mode Active Status — This is a read-only status bit. 0 BDM not active (user application program running) 1 BDM active and waiting for serial commands 					
5 BKPTEN	 BDC Breakpoint Enable — If this bit is clear, the BDC breakpoint is disabled and the FTS (force tag select) control bit and BDCBKPT match register are ignored. 0 BDC breakpoint disabled 1 BDC breakpoint enabled 					
4 FTS	 Force/Tag Select — When FTS = 1, a breakpoint is requested whenever the CPU address bus matches the BDCBKPT match register. When FTS = 0, a match between the CPU address bus and the BDCBKPT register causes the fetched opcode to be tagged. If this tagged opcode ever reaches the end of the instruction queue, the CPU enters active background mode rather than executing the tagged opcode. 0 Tag opcode at breakpoint address and enter active background mode if CPU attempts to execute that instruction 1 Breakpoint match forces active background mode at next instruction boundary (address need not be an opcode) 					
3 CLKSW	Select Source for BDC Communications Clock — CLKSW defaults to 0, which selects the alternate BDC clock source. 0 Alternate BDC clock source 1 MCU bus clock					



Appendix A Electrical Characteristics

- ³ Monotonicity and No-Missing-Codes guaranteed in 10 bit and 8 bit modes
- ⁴ Based on input pad leakage current. Refer to pad electricals.

A.10 External Oscillator (XOSC) Characteristics

Table A-11. Oscillator Electrical Specifications (Temperature Range = -40 to 125°C Ambient)

Num	С	Rating	Symbol	Min	Typ ¹	Max	Unit	
1		Oscillator crystal or resonator (EREFS = 1, ERCLKEN = 1						
		Low range (RANGE = 0)	f _{lo}	32	—	38.4	kHz	
	С	High range (RANGE = 1) FEE or FBE mode 2	f _{hi-fll}	1	—	5	MHz	
		High range (RANGE = 1) PEE or PBE mode ³	f _{hi-pll}	1	—	16	MHz	
		High range (RANGE = 1, HGO = 1) BLPE mode	f _{hi-hgo}	1	—	16	MHz	
		High range (RANGE = 1, HGO = 0) BLPE mode	f _{hi-lp}	1	—	8	MHz	
2	_		C ₁	See crystal or resonator			or	
2		Load capacitors		manufacturer's recommendation				
3	_	Feedback resistor						
		Low range (32 kHz to 100 kHz)	R _F	_	10	—	MΩ	
		High range (1 MHz to 16 MHz)		—	1	_	MΩ	
		Series resistor						
	_	Low range, low gain (RANGE = 0, HGO = 0)		—	0	_		
		Low range, high gain (RANGE = 0, HGO = 1)		—	100	—		
4		High range, low gain (RANGE = 1, HGO = 0)	R _S	—	0	—	kΩ	
		High range, high gain (RANGE = 1, HGO = 1) $\geq 8 \text{ MHz}$			0	0		
		4 MHz			0	10		
		1 MHz			0	20		
	т	Crystal start-up time ⁴						
5		Low range, low gain (RANGE = 0, HGO = 0)	t CSTL-LP		200	—		
		Low range, high gain (RANGE = 0, HGO = 1)	t CSTL-HGO		400	—		
		High range, low gain (RANGE = 1, HGO = $0)^5$	t CSTH-LP		5	—	ms	
		High range, high gain (RANGE = 1, HGO = 1) ⁴	^t CSTH-HGO		15	—		
6		Square wave input clock frequency (EREFS = 0, ERCLKEN = 1)						
	Т	FEE or FBE mode ²		0.03125	—	5		
		PEE or PBE mode ³	f _{extal}	1		16	MHz	
		BLPE mode		0	—	40		

¹ Typical data was characterized at 3.0 V, 25°C or is recommended value.

² When MCG is configured for FEE or FBE mode, the input clock source must be divisible using RDIV to within the range of 31.25 kHz to 39.0625 kHz.

- ³ When MCG is configured for PEE or PBE mode, input clock source must be divisible using RDIV to within the range of 1 MHz to 2 MHz.
- ⁴ This parameter is characterized and not tested on each device. Proper PC board layout procedures must be followed to achieve specifications. This data will vary based upon the crystal manufacturer and board design. The crystal should be characterized by the crystal manufacturer.

⁵ 4 MHz crystal.



Appendix B Timer Pulse-Width Modulator (TPMV2)

NOTE

This chapter refers to S08TPM version 2, which applies to the 3M05C and older mask sets of this device.)M74K and newer mask set devices use S08TPM version 3. If your device uses mask 0M74K or newer, please refer to Chapter 16, "Timer Pulse-Width Modulator (S08TPMV3) for information pertaining to that module.

The TPM uses one input/output (I/O) pin per channel, TPMxCHn where x is the TPM number (for example, 1 or 2) and n is the channel number (for example, 0–4). The TPM shares its I/O pins with general-purpose I/O port pins (refer to the Pins and Connections chapter for more information).

B.0.1 Features

The TPM has the following features:

- Each TPM may be configured for buffered, center-aligned pulse-width modulation (CPWM) on all channels
- Clock sources independently selectable per TPM (multiple TPMs device)
- Selectable clock sources (device dependent): bus clock, fixed system clock, external pin
- Clock prescaler taps for divide by 1, 2, 4, 8, 16, 32, 64, or 128
- 16-bit free-running or up/down (CPWM) count operation
- 16-bit modulus register to control counter range
- Timer system enable
- One interrupt per channel plus a terminal count interrupt for each TPM module (multiple TPMs device)
- Channel features:
 - Each channel may be input capture, output compare, or buffered edge-aligned PWM
 - Rising-edge, falling-edge, or any-edge input capture trigger
 - Set, clear, or toggle output compare action
 - Selectable polarity on PWM outputs

B.0.2 Block Diagram

Figure B-1 shows the structure of a TPM. Some MCUs include more than one TPM, with various numbers of channels.



In input capture mode, reading either byte (TPMxCnVH or TPMxCnVL) latches the contents of both bytes into a buffer where they remain latched until the other byte is read. This latching mechanism also resets (becomes unlatched) when the TPMxCnSC register is written.

In output compare or PWM modes, writing to either byte (TPMxCnVH or TPMxCnVL) latches the value into a buffer. When both bytes have been written, they are transferred as a coherent 16-bit value into the timer channel value registers. This latching mechanism may be manually reset by writing to the TPMxCnSC register.

This latching mechanism allows coherent 16-bit writes in either order, which is friendly to various compiler implementations.

B.3 Functional Description

All TPM functions are associated with a main 16-bit counter that allows flexible selection of the clock source and prescale divisor. A 16-bit modulo register also is associated with the main 16-bit counter in the TPM. Each TPM channel is optionally associated with an MCU pin and a maskable interrupt function.

The TPM has center-aligned PWM capabilities controlled by the CPWMS control bit in TPMxSC. When CPWMS is set to 1, timer counter TPMxCNT changes to an up-/down-counter and all channels in the associated TPM act as center-aligned PWM channels. When CPWMS = 0, each channel can independently be configured to operate in input capture, output compare, or buffered edge-aligned PWM mode.

The following sections describe the main 16-bit counter and each of the timer operating modes (input capture, output compare, edge-aligned PWM, and center-aligned PWM). Because details of pin operation and interrupt activity depend on the operating mode, these topics are covered in the associated mode sections.

B.3.1 Counter

All timer functions are based on the main 16-bit counter (TPMxCNTH:TPMxCNTL). This section discusses selection of the clock source, up-counting vs. up-/down-counting, end-of-count overflow, and manual counter reset.

After any MCU reset, CLKSB:CLKSA = 0:0 so no clock source is selected and the TPM is inactive. Normally, CLKSB:CLKSA would be set to 0:1 so the bus clock drives the timer counter. The clock source for the TPM can be selected to be off, the bus clock (BUSCLK), the fixed system clock (XCLK), or an external input. The maximum frequency allowed for the external clock option is one-fourth the bus rate. Refer to Section B.2.1, "Timer Status and Control Register (TPMxSC)" and Table B-2 for more information about clock source selection.

When the microcontroller is in active background mode, the TPM temporarily suspends all counting until the microcontroller returns to normal user operating mode. During stop mode, all TPM clocks are stopped; therefore, the TPM is effectively disabled until clocks resume. During wait mode, the TPM continues to operate normally.

The main 16-bit counter has two counting modes. When center-aligned PWM is selected (CPWMS = 1), the counter operates in up-/down-counting mode. Otherwise, the counter operates as a simple up-counter.