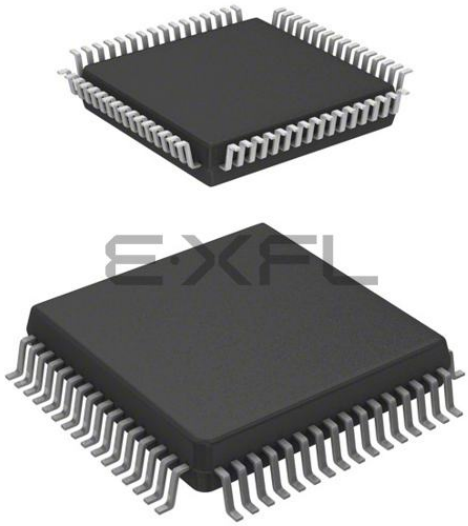


Welcome to E-XFL.COM

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	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	53
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-QFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908gz48mfue

Chapter 4
Clock Generator Module (CGM)

Chapter 5
Configuration Register (CONFIG)

Chapter 15
System Integration Module (SIM)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0459	TIM2 Channel 3 Status and Control Register (T2SC3) See page 293.	Read:	CH3F	CH3IE	0	MS3A	ELS3B	ELS3A	TOV3	CH3MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$045A	TIM2 Channel 3 Register High (T2CH3H) See page 297.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$045B	TIM2 Channel 3 Register Low (T2CH3L) See page 297.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$045C	TIM2 Channel 4 Status and Control Register (T2SC4) See page 293.	Read:	CH4F	CH4IE	MS4B	MS4A	ELS4B	ELS4A	TOV4	CH4MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$045D	TIM2 Channel 4 Register High (T2CH4H) See page 297.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$045E	TIM2 Channel 4 Register Low (T2CH4L) See page 297.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$045F	TIM2 Channel 5 Status and Control Register (T2SC5) See page 293.	Read:	CH5F	CH5IE	0	MS5A	ELS5B	ELS5A	TOV 5	CH5MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0460	TIM2 Channel 5 Register High (T2CH5H) See page 297.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0461	TIM2 Channel 5 Register Low (T2CH5L) See page 297.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$FE00	Break Status Register (BSR) See page 237.	Read:	R	R	R	R	R	R	SBSW	R
		Write:							NOTE 1	
		Reset:	0	0	0	0	0	0	0	0

1. Writing a 0 clears SBSW.

$FE01	SIM Reset Status Register (SRSR) See page 237.	Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
		Write:								
		POR:	1	0	0	0	0	0	0	0
$FE02	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 7 of 9)

3.8.2.3 Left Justified Signed Data Mode

In left justified signed data mode, the ADRH register holds the eight MSBs of the 10-bit result. The only difference from left justified mode is that the AD9 is complemented. The ADRL register holds the two LSBs of the 10-bit result. All other bits read as 0. ADRH and ADRL are updated each time an ADC single channel conversion completes. Reading ADRH latches the contents of ADRL until ADRL is read. All subsequent results will be lost until the ADRH and ADRL reads are completed.

Address:	$003D							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	$\overline{AD9}$	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Write:								
Reset:	Unaffected by reset							
Address:	$003E							
Read:	AD1	AD0	0	0	0	0	0	0
Write:								
Reset:	Unaffected by reset							

= Unimplemented

Figure 3-7. ADC Data Register High (ADRH) and Low (ADRL)

3.8.2.4 Eight Bit Truncation Mode

In 8-bit truncation mode, the ADRL register holds the eight MSBs of the 10-bit result. The ADRH register is unused and reads as 0. The ADRL register is updated each time an ADC single channel conversion completes. In 8-bit mode, the ADRL register contains no interlocking with ADRH.

Address:	$003D							ADRH
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	0	0
Write:								
Reset:	Unaffected by reset							
Address:	$003E							ADRL
Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Write:								
Reset:	Unaffected by reset							

= Unimplemented

Figure 3-8. ADC Data Register High (ADRH) and Low (ADRL)

OSCENINSTOP — Oscillator Enable In Stop Mode Bit

OSCENINSTOP, when set, will enable the oscillator to continue to generate clocks in stop mode. See Chapter 4 Clock Generator Module (CGM). This function is used to keep the timebase running while the rest of the MCU stops. See Chapter 17 Timebase Module (TBM). When clear, the oscillator will cease to generate clocks while in stop mode. The default state for this option is clear, disabling the oscillator in stop mode.

1 = Oscillator enabled during stop mode
0 = Oscillator disabled during stop mode (default)

SCIBDSRC — SCI Baud Rate Clock Source Bit

SCIBDSRC controls the clock source used for the serial communications interface (SCI). The setting of this bit affects the frequency at which the SCI operates.See Chapter 14 Enhanced Serial Communications Interface (ESCI) Module.

1 = Internal data bus clock used as clock source for SCI (default)
0 = External oscillator used as clock source for SCI

Address: $001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVI5OR3	SSREC	STOP	COPD
Reset:	0	0	0	0	See note	0	0	0

Note: LVI5OR3 is only reset via POR (power-on reset).

Figure 5-2. Configuration Register 1 (CONFIG1)

COPRS — COP Rate Select Bit

COPRS selects the COP timeout period. Reset clears COPRS. See Chapter 6 Computer Operating Properly (COP) Module

1 = COP timeout period = 8176 CGMXCLK cycles
0 = COP timeout period = 262,128 CGMXCLK cycles

LVISTOP — LVI Enable in Stop Mode Bit

When the LVIPWRD bit is clear, setting the LVISTOP bit enables the LVI to operate during stop mode. Reset clears LVISTOP.

1 = LVI enabled during stop mode
0 = LVI disabled during stop mode

LVIRSTD — LVI Reset Disable Bit

LVIRSTD disables the reset signal from the LVI module.
See Chapter 11 Low-Voltage Inhibit (LVI).

1 = LVI module resets disabled
0 = LVI module resets enabled

LVIPWRD — LVI Power Disable Bit

LVIPWRD disables the LVI module. See Chapter 11 Low-Voltage Inhibit (LVI).

1 = LVI module power disabled
0 = LVI module power enabled

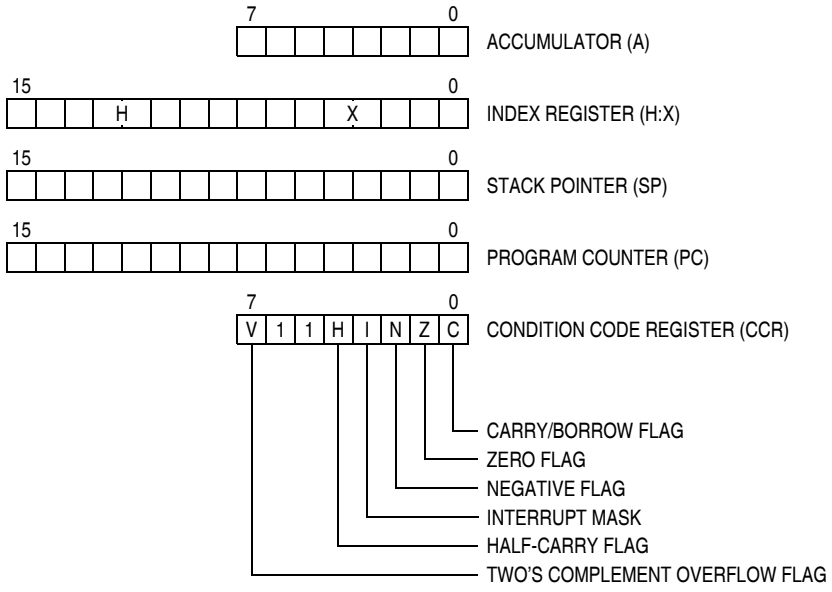


Figure 7-1. CPU Registers

7.3.1 Accumulator

The accumulator is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and the results of arithmetic/logic operations.

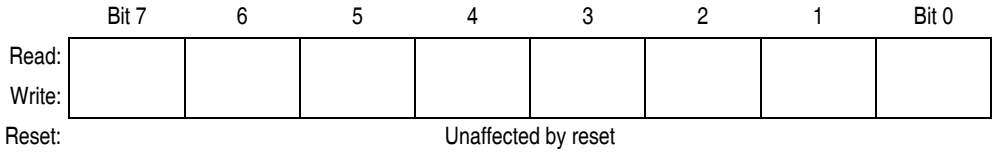


Figure 7-2. Accumulator (A)

7.3.2 Index Register

The 16-bit index register allows indexed addressing of a 64-Kbyte memory space. H is the upper byte of the index register, and X is the lower byte. H:X is the concatenated 16-bit index register.

In the indexed addressing modes, the CPU uses the contents of the index register to determine the conditional address of the operand.

The index register can serve also as a temporary data storage location.

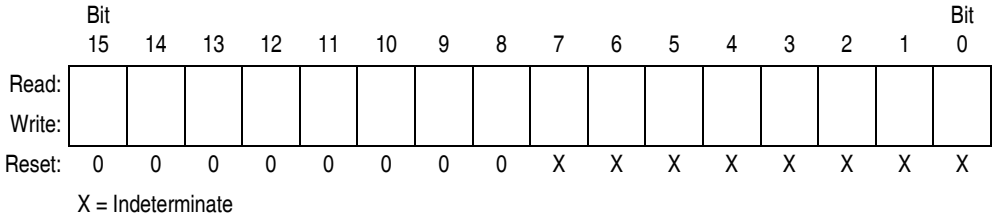


Figure 7-3. Index Register (H:X)

10.9 Low-Voltage Inhibit Module (LVI)

10.9.1 Wait Mode

If enabled, the low-voltage inhibit (LVI) module remains active in wait mode. If enabled to generate resets, the LVI module can generate a reset and bring the MCU out of wait mode.

10.9.2 Stop Mode

If enabled, the LVI module remains active in stop mode. If enabled to generate resets, the LVI module can generate a reset and bring the MCU out of stop mode.

10.10 Enhanced Serial Communications Interface Module (ESCI)

10.10.1 Wait Mode

The enhanced serial communications interface (ESCI), or SCI module for short, module remains active in wait mode. Any enabled CPU interrupt request from the SCI module can bring the MCU out of wait mode.

If SCI module functions are not required during wait mode, reduce power consumption by disabling the module before executing the WAIT instruction.

10.10.2 Stop Mode

The SCI module is inactive in stop mode. The STOP instruction does not affect SCI register states. SCI module operation resumes after the MCU exits stop mode.

Because the internal clock is inactive during stop mode, entering stop mode during an SCI transmission or reception results in invalid data.

10.11 Serial Peripheral Interface Module (SPI)

10.11.1 Wait Mode

The serial peripheral interface (SPI) module remains active in wait mode. Any enabled CPU interrupt request from the SPI module can bring the MCU out of wait mode.

If SPI module functions are not required during wait mode, reduce power consumption by disabling the SPI module before executing the WAIT instruction.

10.11.2 Stop Mode

The SPI module is inactive in stop mode. The STOP instruction does not affect SPI register states. SPI operation resumes after an external interrupt. If stop mode is exited by reset, any transfer in progress is aborted, and the SPI is reset.

To transmit a message, the CPU08 has to identify an available transmit buffer which is indicated by a set transmit buffer empty (TXE) flag in the MSCAN08 transmitter flag register (CTFLG) (see 12.13.7 MSCAN08 Transmitter Flag Register).

The CPU08 then stores the identifier, the control bits and the data content into one of the transmit buffers. Finally, the buffer has to be flagged ready for transmission by clearing the TXE flag.

The MSCAN08 then will schedule the message for transmission and will signal the successful transmission of the buffer by setting the TXE flag. A transmit interrupt is generated[1] when TXE is set and can be used to drive the application software to re-load the buffer.

In case more than one buffer is scheduled for transmission when the CAN bus becomes available for arbitration, the MSCAN08 uses the local priority setting of the three buffers for prioritization. For this purpose, every transmit buffer has an 8-bit local priority field (PRIO). The application software sets this field when the message is set up. The local priority reflects the priority of this particular message relative to the set of messages being emitted from this node. The lowest binary value of the PRIO field is defined as the highest priority.

The internal scheduling process takes place whenever the MSCAN08 arbitrates for the bus. This is also the case after the occurrence of a transmission error.

When a high priority message is scheduled by the application software, it may become necessary to abort a lower priority message being set up in one of the three transmit buffers. As messages that are already under transmission cannot be aborted, the user has to request the abort by setting the corresponding abort request flag (ABTRQ) in the transmission control register (CTCR). The MSCAN08 will then grant the request, if possible, by setting the corresponding abort request acknowledge (ABTAK) and the TXE flag in order to release the buffer and by generating a transmit interrupt. The transmit interrupt handler software can tell from the setting of the ABTAK flag whether the message was actually aborted (ABTAK = 1) or sent (ABTAK = 0).

12.5 Identifier Acceptance Filter

The identifier acceptance registers (CIDAR0–CIDAR3) define the acceptance patterns of the standard or extended identifier (ID10–ID0 or ID28–ID0). Any of these bits can be marked ‘don’t care’ in the identifier mask registers (CIDMR0–CIDMR3).

A filter hit is indicated to the application on software by a set RXF (receive buffer full flag, see 12.13.5 MSCAN08 Receiver Flag Register (CRFLG)) and two bits in the identifier acceptance control register (see 12.13.9 MSCAN08 Identifier Acceptance Control Register). These identifier hit flags (IDHIT1 and IDHIT0) clearly identify the filter section that caused the acceptance. They simplify the application software’s task to identify the cause of the receiver interrupt. In case that more than one hit occurs (two or more filters match) the lower hit has priority.

A very flexible programmable generic identifier acceptance filter has been introduced to reduce the CPU interrupt loading. The filter is programmable to operate in four different modes:

1. Single identifier acceptance filter, each to be applied to a) the full 29 bits of the extended identifier and to the following bits of the CAN frame: RTR, IDE, SRR or b) the 11 bits of the standard identifier plus the RTR and IDE bits of CAN 2.0A/B messages. This mode implements a single filter for a full length CAN 2.0B compliant extended identifier. Figure 12-4 shows how the 32-bit filter bank (CIDAR0-3, CIDMR0-3) produces a filter 0 hit.

1. The transmit interrupt will occur only if not masked. A polling scheme can be applied on TXE also.

2. Two identifier acceptance filters, each to be applied to:
   a. The 14 most significant bits of the extended identifier plus the SRR and the IDE bits of CAN2.0B messages, or
   b. The 11 bits of the identifier plus the RTR and IDE bits of CAN 2.0A/B messages.

   Figure 12-5 shows how the 32-bit filter bank (CIDAR0–CIDAR3 and CIDMR0–CIDMR3) produces filter 0 and 1 hits.
3. Four identifier acceptance filters, each to be applied to the first eight bits of the identifier. This mode implements four independent filters for the first eight bits of a CAN 2.0A/B compliant standard identifier. Figure 12-6 shows how the 32-bit filter bank (CIDAR0–CIDAR3 and CIDMR0–CIDMR3) produces filter 0 to 3 hits.
4. Closed filter. No CAN message will be copied into the foreground buffer RxFG, and the RXF flag will never be set.

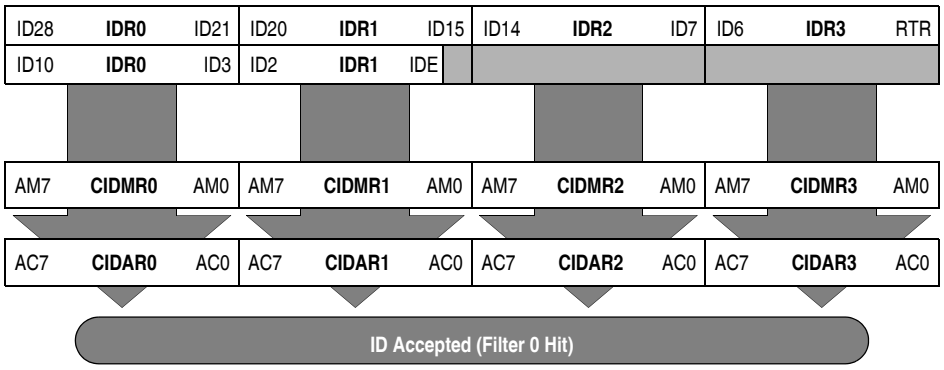


Figure 12-4. Single 32-Bit Maskable Identifier Acceptance Filter

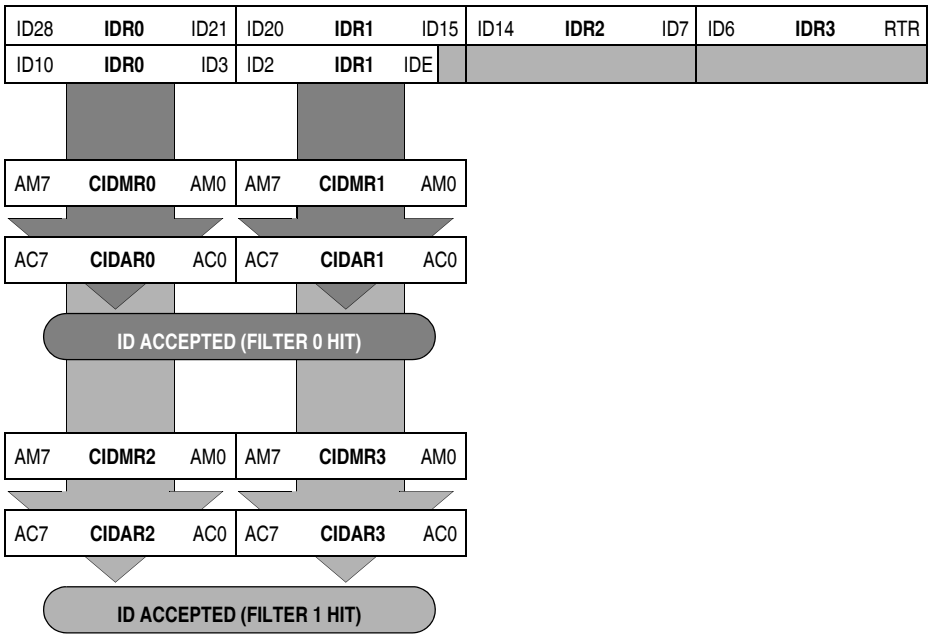


Figure 12-5. Dual 16-Bit Maskable Acceptance Filters

During sleep mode, the SLPAK flag is set. The application software should use SLPAK as a handshake indication for the request (SLPRQ) to go into sleep mode. When in sleep mode, the MSCAN08 stops its internal clocks. However, clocks to allow register accesses still run. If the MSCAN08 is in bus-off state, it stops counting the 128*11 consecutive recessive bits due to the stopped clocks. The CAN_{TX} pin stays in recessive state. If RXF = 1, the message can be read and RXF can be cleared. Copying of RxGB into RxFG doesn't take place while in sleep mode. It is possible to access the transmit buffers and to clear the TXE flags. No message abort takes place while in sleep mode.

The MSCAN08 leaves sleep mode (wakes-up) when:

- Bus activity occurs, or
- The MCU clears the SLPRQ bit, or
- The MCU sets the SFTRES bit

NOTE

The MCU cannot clear the SLPRQ bit before the MSCAN08 is in sleep mode (SLPAK=1).

After wakeup, the MSCAN08 waits for 11 consecutive recessive bits to synchronize to the bus. As a consequence, if the MSCAN08 is woken-up by a CAN frame, this frame is not received. The receive message buffers (RxFG and RxBG) contain messages if they were received before sleep mode was entered. All pending actions are executed upon wakeup: copying of RxBG into RxFG, message aborts and message transmissions. If the MSCAN08 is still in bus-off state after sleep mode was left, it continues counting the 128*11 consecutive recessive bits.

12.8.2 MSCAN08 Soft Reset Mode

In soft reset mode, the MSCAN08 is stopped. Registers can still be accessed. This mode is used to initialize the module configuration, bit timing and the CAN message filter. See 12.13.1 MSCAN08 Module Control Register 0 for a complete description of the soft reset mode.

When setting the SFTRES bit, the MSCAN08 immediately stops all ongoing transmissions and receptions, potentially causing CAN protocol violations.

NOTE

The user is responsible to take care that the MSCAN08 is not active when soft reset mode is entered. The recommended procedure is to bring the MSCAN08 into sleep mode before the SFTRES bit is set.

12.8.3 MSCAN08 Power-Down Mode

The MSCAN08 is in power-down mode when the CPU is in stop mode.

When entering the power-down mode, the MSCAN08 immediately stops all ongoing transmissions and receptions, potentially causing CAN protocol violations.

NOTE

The user is responsible to take care that the MSCAN08 is not active when power-down mode is entered. The recommended procedure is to bring the MSCAN08 into sleep mode before the STOP instruction is executed.

To protect the CAN bus system from fatal consequences resulting from violations of the above rule, the MSCAN08 drives the CAN_{TX} pin into recessive state.

In power-down mode, no registers can be accessed.

does not set the receiver idle bit, IDLE, or the ESCI receiver full bit, SCRF. The idle line type bit, ILTY, determines whether the receiver begins counting 1s as idle character bits after the start bit or after the stop bit.

NOTE

With the WAKE bit clear, setting the RWU bit after the RxD pin has been idle will cause the receiver to wake up.

14.4.3.7 Receiver Interrupts

These sources can generate CPU interrupt requests from the ESCI receiver:

- ESCI receiver full (SCRF) — The SCRF bit in SCS1 indicates that the receive shift register has transferred a character to the SCDR. SCRF can generate a receiver CPU interrupt request. Setting the ESCI receive interrupt enable bit, SCRIE, in SCC2 enables the SCRF bit to generate receiver CPU interrupts.
- Idle input (IDLE) — The IDLE bit in SCS1 indicates that 10 or 11 consecutive 1s shifted in from the RxD pin. The idle line interrupt enable bit, ILIE, in SCC2 enables the IDLE bit to generate CPU interrupt requests.

14.4.3.8 Error Interrupts

These receiver error flags in SCS1 can generate CPU interrupt requests:

- Receiver overrun (OR) — The OR bit indicates that the receive shift register shifted in a new character before the previous character was read from the SCDR. The previous character remains in the SCDR, and the new character is lost. The overrun interrupt enable bit, ORIE, in SCC3 enables OR to generate ESCI error CPU interrupt requests.
- Noise flag (NF) — The NF bit is set when the ESCI detects noise on incoming data or break characters, including start, data, and stop bits. The noise error interrupt enable bit, NEIE, in SCC3 enables NF to generate ESCI error CPU interrupt requests.
- Framing error (FE) — The FE bit in SCS1 is set when a 0 occurs where the receiver expects a stop bit. The framing error interrupt enable bit, FEIE, in SCC3 enables FE to generate ESCI error CPU interrupt requests.
- Parity error (PE) — The PE bit in SCS1 is set when the ESCI detects a parity error in incoming data. The parity error interrupt enable bit, PEIE, in SCC3 enables PE to generate ESCI error CPU interrupt requests.

14.5 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

14.5.1 Wait Mode

The ESCI module remains active in wait mode. Any enabled CPU interrupt request from the ESCI module can bring the MCU out of wait mode.

If ESCI module functions are not required during wait mode, reduce power consumption by disabling the module before executing the WAIT instruction.

IDLE — Receiver Idle Bit

This clearable, read-only bit is set when 10 or 11 consecutive 1s appear on the receiver input. IDLE generates an ESCI receiver CPU interrupt request if the ILIE bit in SCC2 is also set. Clear the IDLE bit by reading SCS1 with IDLE set and then reading the SCDR. After the receiver is enabled, it must receive a valid character that sets the SCRF bit before an idle condition can set the IDLE bit. Also, after the IDLE bit has been cleared, a valid character must again set the SCRF bit before an idle condition can set the IDLE bit. Reset clears the IDLE bit.

1 = Receiver input idle
0 = Receiver input active (or idle since the IDLE bit was cleared)

OR — Receiver Overrun Bit

This clearable, read-only bit is set when software fails to read the SCDR before the receive shift register receives the next character. The OR bit generates an ESCI error CPU interrupt request if the ORIE bit in SCC3 is also set. The data in the shift register is lost, but the data already in the SCDR is not affected. Clear the OR bit by reading SCS1 with OR set and then reading the SCDR. Reset clears the OR bit.

1 = Receive shift register full and SCRF = 1
0 = No receiver overrun

Software latency may allow an overrun to occur between reads of SCS1 and SCDR in the flag-clearing sequence. Figure 14-14 shows the normal flag-clearing sequence and an example of an overrun caused by a delayed flag-clearing sequence. The delayed read of SCDR does not clear the OR bit because OR was not set when SCS1 was read. Byte 2 caused the overrun and is lost. The next flag-clearing sequence reads byte 3 in the SCDR instead of byte 2.

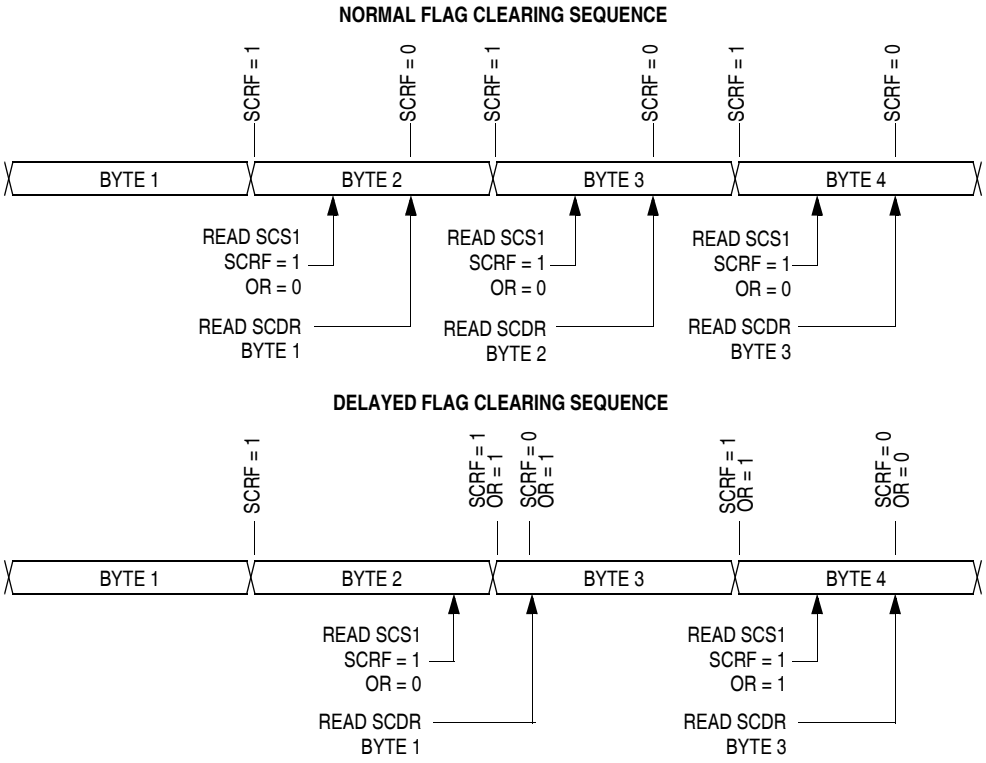
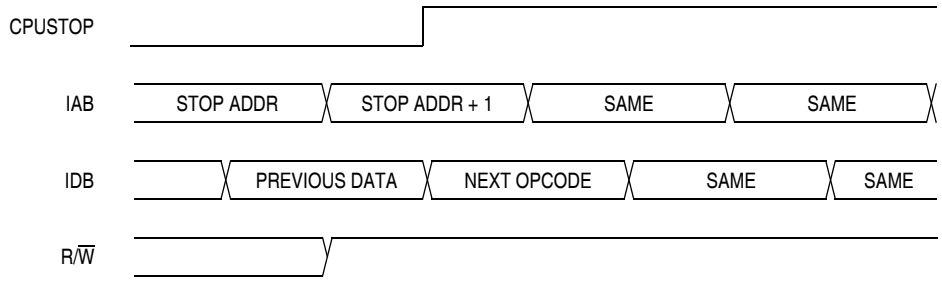


Figure 14-14. Flag Clearing Sequence

The SIM counter is held in reset from the execution of the STOP instruction until the beginning of stop recovery. It is then used to time the recovery period. Figure 15-19 shows stop mode entry timing. Figure 15-20 shows stop mode recovery time from interrupt.

NOTE

To minimize stop current, all pins configured as inputs should be driven to a 1 or 0.



Note: Previous data can be operand data or the STOP opcode, depending on the last instruction.

Figure 15-19. Stop Mode Entry Timing

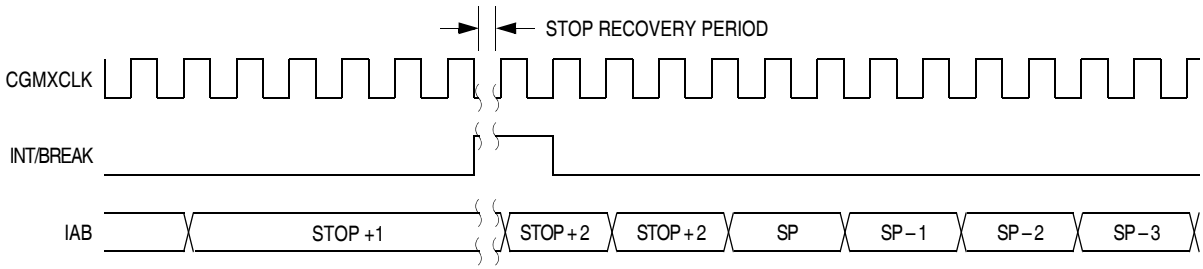


Figure 15-20. Stop Mode Recovery from Interrupt

15.7 SIM Registers

The SIM has three memory-mapped registers. Table 15-4 shows the mapping of these registers.

Table 15-4. SIM Registers

Address	Register	Access Mode
$FE00	BSR	User
$FE01	SRSR	User
$FE03	BFCR	User

16.7 Interrupts

Four SPI status flags can be enabled to generate CPU interrupt requests. See Table 16-1.

Table 16-1. SPI Interrupts

Flag	Request
SPTE Transmitter empty	SPI transmitter CPU interrupt request (SPTIE = 1, SPE = 1)
SPRF Receiver full	SPI receiver CPU interrupt request (SPRIE = 1)
OVRF Overflow	SPI receiver/error interrupt request (ERRIE = 1)
MODF Mode fault	SPI receiver/error interrupt request (ERRIE = 1)

Reading the SPI status and control register with SPRF set and then reading the receive data register clears SPRF. The clearing mechanism for the SPTE flag is always just a write to the transmit data register.

The SPI transmitter interrupt enable bit (SPTIE) enables the SPTE flag to generate transmitter CPU interrupt requests, provided that the SPI is enabled (SPE = 1).

The SPI receiver interrupt enable bit (SPRIE) enables SPRF to generate receiver CPU interrupt requests, regardless of the state of SPE. See Figure 16-12.

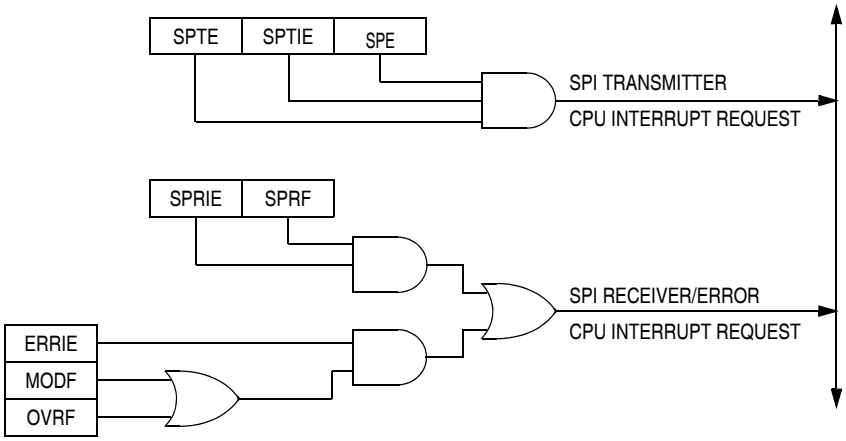


Figure 16-12. SPI Interrupt Request Generation

The error interrupt enable bit (ERRIE) enables both the MODF and OVRF bits to generate a receiver/error CPU interrupt request.

The mode fault enable bit (MODFEN) can prevent the MODF flag from being set so that only the OVRF bit is enabled by the ERRIE bit to generate receiver/error CPU interrupt requests.

18.8.2 TIM1 Counter Registers

The two read-only TIM1 counter registers contain the high and low bytes of the value in the TIM1 counter. Reading the high byte (T1CNTH) latches the contents of the low byte (T1CNTL) into a buffer. Subsequent reads of T1CNTH do not affect the latched T1CNTL value until T1CNTL is read. Reset clears the TIM1 counter registers. Setting the TIM1 reset bit (TRST) also clears the TIM1 counter registers.

NOTE

If you read T1CNTH during a break interrupt, be sure to unlatch T1CNTL by reading T1CNTL before exiting the break interrupt. Otherwise, T1CNTL retains the value latched during the break.

Address: $0021 T1CNTH

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Write:								
Reset:	0	0	0	0	0	0	0	0

Address: $0022 T1CNTL

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 18-6. TIM1 Counter Registers (T1CNTH:T1CNTL)

18.8.3 TIM1 Counter Modulo Registers

The read/write TIM1 modulo registers contain the modulo value for the TIM1 counter. When the TIM1 counter reaches the modulo value, the overflow flag (TOF) becomes set, and the TIM1 counter resumes counting from $0000 at the next timer clock. Writing to the high byte (T1MODH) inhibits the TOF bit and overflow interrupts until the low byte (T1MODL) is written. Reset sets the TIM1 counter modulo registers.

Address: $0023 T1MODH

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
Reset:	1	1	1	1	1	1	1	1

Address: $0024 T1MODL

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reset:	1	1	1	1	1	1	1	1

Figure 18-7. TIM1 Counter Modulo Registers (T1MODH:T1MODL)

NOTE

Reset the TIM1 counter before writing to the TIM1 counter modulo registers.

19.3.1 TIM2 Counter Prescaler

The TIM2 clock source can be one of the seven prescaler outputs or the TIM2 clock pin, T2CH0. The prescaler generates seven clock rates from the internal bus clock. The prescaler select bits, PS[2:0], in the TIM2 status and control register select the TIM2 clock source.

19.3.2 Input Capture

An input capture function has three basic parts: edge select logic, an input capture latch, and a 16-bit counter. Two 8-bit registers, which make up the 16-bit input capture register, are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The polarity of the active edge is programmable. The level transition which triggers the counter transfer is defined by the corresponding input edge bits (ELSxB and ELSxA in T2SC0 through T2SC5 control registers with x referring to the active channel number). When an active edge occurs on the pin of an input capture channel, the TIM2 latches the contents of the TIM2 counter into the TIM2 channel registers, T2CHxH:T2CHxL. Input captures can generate TIM2 CPU interrupt requests. Software can determine that an input capture event has occurred by enabling input capture interrupts or by polling the status flag bit.

The free-running counter contents are transferred to the TIM2 channel registers (T2CHxH:T2CHxL) (see 19.8.5 TIM2 Channel Registers) on each proper signal transition regardless of whether the TIM2 channel flag (CH0F–CH5F in T2SC0–T2SC5 registers) is set or clear. When the status flag is set, a CPU interrupt is generated if enabled. The value of the count latched or “captured” is the time of the event. Because this value is stored in the input capture register when the actual event occurs, user software can respond to this event at a later time and determine the actual time of the event. However, this must be done prior to another input capture on the same pin; otherwise, the previous time value will be lost.

By recording the times for successive edges on an incoming signal, software can determine the period and/or pulse width of the signal. To measure a period, two successive edges of the same polarity are captured. To measure a pulse width, two alternate polarity edges are captured. Software should track the overflows at the 16-bit module counter to extend its range.

Another use for the input capture function is to establish a time reference. In this case, an input capture function is used in conjunction with an output compare function. For example, to activate an output signal a specified number of clock cycles after detecting an input event (edge), use the input capture function to record the time at which the edge occurred. A number corresponding to the desired delay is added to this captured value and stored to an output compare register (see 19.8.5 TIM2 Channel Registers). Because both input captures and output compares are referenced to the same 16-bit modulo counter, the delay can be controlled to the resolution of the counter independent of software latencies.

Reset does not affect the contents of the input capture channel (T2CHxH:T2CHxL) registers.

19.3.3 Output Compare

With the output compare function, the TIM2 can generate a periodic pulse with a programmable polarity, duration, and frequency. When the counter reaches the value in the registers of an output compare channel, the TIM2 can set, clear, or toggle the channel pin. Output compares can generate TIM2 CPU interrupt requests.

19.6 TIM2 During Break Interrupts

A break interrupt stops the TIM2 counter and inhibits input captures.

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state. (See 15.7.3 Break Flag Control Register.)

To allow software to clear status bits during a break interrupt, write a 1 to the BCFE bit. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect status bits during the break state, write a 0 to the BCFE bit. With BCFE at 0 (its default state), software can read and write I/O registers during the break state without affecting status bits. Some status bits have a 2-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is at 0. After the break, doing the second step clears the status bit.

19.7 I/O Signals

Port D shares two of its pins with the TIM2. Port F shares four of its pins with the TIM2. PTD6/T2CH0 is an external clock input to the TIM2 prescaler. The six TIM2 channel I/O pins are PTD6/T2CH0, PTD7/T2CH1, PTF4/T2CH2, PTF5/T2CH3, PTF6/T2CH4, and PTF7/T2CH5.

19.7.1 TIM2 Clock Pin (T2CH0)

T2CH0 is an external clock input that can be the clock source for the TIM2 counter instead of the prescaled internal bus clock. Select the T2CH0 input by writing 1s to the three prescaler select bits, PS[2:0]. (See 19.8.1 TIM2 Status and Control Register.) The minimum TCLK pulse width is specified in 21.14 Timer Interface Module Characteristics. The maximum TCLK frequency is the least: 4 MHz or bus frequency ÷ 2.

When the PTD6/T2CH0 pin is the TIM2 clock input, it is an input regardless of the state of the DDRD6 bit in data direction register D.

19.7.2 TIM2 Channel I/O Pins (T2CH5:T2CH2 and T2CH1:T2CH0)

Each channel I/O pin is programmable independently as an input capture pin or an output compare pin. T2CH0, T2CH2, and T2CH4 can be configured as buffered output compare or buffered PWM pins.

19.8 I/O Registers

These I/O registers control and monitor TIM2 operation:

- TIM2 status and control register (T2SC)
- TIM2 counter registers (T2CNTH:T2CNTL)
- TIM2 counter modulo registers (T2MODH:T2MODL)
- TIM2 channel status and control registers (T2SC0, T2SC1, T2SC2, T2SC3, T2SC4, and T2SC5)
- TIM2 channel registers (T2CH0H:T2CH0L, T2CH1H:T2CH1L, T2CH2H:T2CH2L, T2CH3H:T2CH3L, T2CH4H:T2CH4L, and T2CH5H:T2CH5L)

Address:	$0033	T2SC1						
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0
Address:	$0456	T2SC2						
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH2F	CH2IE	MS2B	MS2A	ELS2B	ELS2A	TOV2	CH2MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0
Address:	$0459	T2SC3						
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH3F	CH3IE	0	MS3A	ELS3B	ELS3A	TOV3	CH3MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0
Address:	$045C	T2SC4						
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH4F	CH4IE	MS4B	MS4A	ELS4B	ELS4A	TOV4	CH4MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0
Address:	$045F	T2SC5						
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH5F	CH5IE	0	MS5A	ELS5B	ELS5A	TOV5	CH5MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 19-8. TIM2 Channel Status and Control Registers (T2SC0:T2SC5) (Continued)

CHxF — Channel x Flag Bit

When channel x is an input capture channel, this read/write bit is set when an active edge occurs on the channel x pin. When channel x is an output compare channel, CHxF is set when the value in the TIM2 counter registers matches the value in the TIM2 channel x registers.

When CHxIE = 1, clear CHxF by reading TIM2 channel x status and control register with CHxF set, and then writing a 0 to CHxF. If another interrupt request occurs before the clearing sequence is complete, then writing 0 to CHxF has no effect. Therefore, an interrupt request cannot be lost due to inadvertent clearing of CHxF.

Reset clears the CHxF bit. Writing a 1 to CHxF has no effect.

1 = Input capture or output compare on channel x
0 = No input capture or output compare on channel x

CHxIE — Channel x Interrupt Enable Bit

This read/write bit enables TIM2 CPU interrupts on channel x.
Reset clears the CHxIE bit.

1 = Channel x CPU interrupt requests enabled
0 = Channel x CPU interrupt requests disabled

21.9.3 CGM Acquisition/Lock Time Information

Characteristic	Symbol	Min	Typ	Max	Unit
Acquisition mode entry frequency tolerance(1)	Δ_{ACQ}	± 3.6	—	± 7.2	%
Tracking mode entry frequency tolerance(2)	Δ_{TRK}	0	—	± 3.6	%
LOCK entry frequency tolerance(3)	Δ_{LOCK}	0	—	± 0.9	%
LOCK exit frequency tolerance(4)	Δ_{UNL}	± 0.9	—	± 1.8	%
Reference cycles per acquisition mode period	n_{ACQ}	—	32	—	
Reference cycles per tracking mode period	n_{TRK}	—	128	—	
Automatic mode time to stable	t_{ACQ}	n_{ACQ}/f_{RCLK}	See note(5)	—	s
Automatic stable to lock time	t_{AL}	n_{TRK}/f_{RCLK}	See note(6)	—	s
Automatic lock time ($t_{ACQ} + t_{AL}$)(7)	t_{LOCK}	—	5	25	ms
PLL jitter, deviation of average bus frequency over 2 ms period	f_J	0	—	f_{RCLK} x 0.025% x N/4	Hz

1. Deviation between VCO frequency and desired frequency to enter PLL acquisition mode.
2. Deviation between VCO frequency and desired frequency to enter PLL tracking mode (stable).
3. Deviation between VCO frequency and desired frequency to enter locked mode.
4. Deviation between VCO frequency and desired frequency to exit locked mode.
5. Acquisition time is an integer multiple of reference cycles divided by reference clock.
6. Stable to lock time is an integer multiple of reference cycles divided by reference clock.
7. Maximum lock time depends on CGMXFC filter components, power supply filtering, and reference clock stability. PLL may not lock if improper components or poor filtering and layout are used.