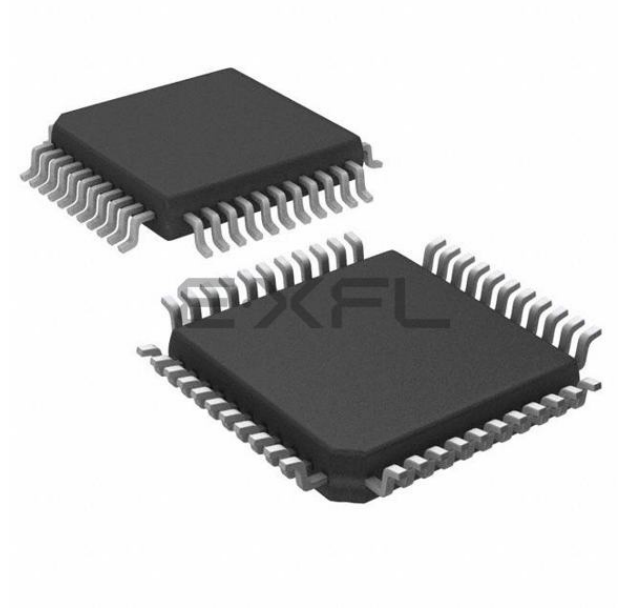

Welcome to E-XFL.COM

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	I²C, IRSCI, SCI, SPI
Peripherals	LED, LVD, POR, PWM
Number of I/O	32
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-QFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc908ap16cfb

Chapter 18
Keyboard Interrupt Module (KBI)

Chapter 19
Computer Operating Properly (COP)

Chapter 20
Low-Voltage Inhibit (LVI)

- Timebase module
- Serial communications interface module 1 (SCI)
- Serial communications interface module 2 (SCI) with infrared (IR) encoder/decoder
- Serial peripheral interface module (SPI)
- System management bus (SMBus), version 1.0/1.1 (multi-master IIC bus)
- 8-channel, 10-bit analog-to-digital converter (ADC)
- $\overline{\text{IRQ1}}$ external interrupt pin with integrated pullup
- $\overline{\text{IRQ2}}$ external interrupt pin with programmable pullup
- 8-bit keyboard wakeup port with integrated pullup
- 32 general-purpose input/output (I/O) pins:
  - 31 shared-function I/O pins
  - 8 LED drivers (sink)
  - 6 × 25mA open-drain I/O with pullup
- Low-power design (fully static with stop and wait modes)
- Master reset pin (with integrated pullup) and power-on reset
- System protection features
  - Optional computer operating properly (COP) reset, driven by internal RC oscillator
  - Low-voltage detection with optional reset or interrupt
  - Illegal opcode detection with reset
  - Illegal address detection with reset
- 48-pin low quad flat pack (LQFP), 44-pin quad flat pack (QFP), and 42-pin shrink dual-in-line package (SDIP)
- Specific features of the MC68HC908AP64 in 42-pin SDIP are:
  - 30 general-purpose I/Os only
  - External interrupt on $\overline{\text{IRQ1}}$ only

Features of the CPU08 include the following:

- Enhanced HC05 programming model
- Extensive loop control functions
- 16 addressing modes (eight more than the HC05)
- 16-bit Index register and stack pointer
- Memory-to-memory data transfers
- Fast 8 × 8 multiply instruction
- Fast 16/8 divide instruction
- Binary-coded decimal (BCD) instructions
- Optimization for controller applications
- Efficient C language support

1.3 MCU Block Diagram

Figure 1-1 shows the structure of the MC68HC908AP64.

Table 1-2. Pin Functions

PIN NAME	PIN DESCRIPTION	IN/OUT	VOLTAGE LEVEL
PTB0/SDA PTB1/SCL PTB2/TxD PTB3/RxD PTB4/T1CH0 PTB5/T1CH1 PTB6/T2CH0 PTB7/T2CH1	8-bit general purpose I/O port; PTB0–PTB3 are open drain when configured as output. PTB4–PTB7 have schmitt trigger inputs.	In/Out	V_{DD}
	PTB0 as SDA of MMIIC.	In/Out	V_{DD}
	PTB1 as SCL of MMIIC.	In/Out	V_{DD}
	PTB2 as TxD of SCI; open drain output.	Out	V_{DD}
	PTB3 as RxD of SCI.	In	V_{DD}
	PTB4 as T1CH0 of TIM1.	In/Out	V_{DD}
	PTB5 as T1CH1 of TIM1.	In/Out	V_{DD}
	PTB6 as T2CH0 of TIM2.	In/Out	V_{DD}
	PTB7 as T2CH1 of TIM2.	In/Out	V_{DD}
PTC0/$\overline{IRQ2}$ PTC1 PTC2/MISO PTC3/MOSI PTC4/$\overline{SS}$ PTC5/SPSCK PTC6/SCTxD PTC7/SCRxD	8-bit general purpose I/O port; PTC6 and PTC7 are open drain when configured as output.	In/Out	V_{DD}
	PTC0 is shared with $\overline{IRQ2}$ and has schmitt trigger input.	In	V_{DD}
	PTC2 as MISO of SPI.	In	V_{DD}
	PTC3 as MOSI of SPI.	Out	V_{DD}
	PTC4 as $\overline{SS}$ of SPI.	In	V_{DD}
	PTC5 as SPSCK of SPI.	In/Out	V_{DD}
	PTC6 as SCTxD of IRSCI; open drain output.	Out	V_{DD}
	PTC7 as SCRxD of IRSCI.	In	V_{DD}
PTD0/KBI0 : PTD7/KBI7	8-bit general purpose I/O port with schmitt trigger inputs.	In/Out	V_{DD}
	Pins as keyboard interrupts (with pullup), KBI0–KBI7.	In	V_{DD}

1. See Chapter 22 Electrical Specifications for V_{REG} tolerance.

1.6 Power Supply Bypassing (VDD, VDDA, VSS, VSSA)

V_{DD} and V_{SS} are the power supply and ground pins, the MCU operates from a single power supply together with an on chip voltage regulator.

Fast signal transitions on MCU pins place high. short-duration current demands on the power supply. To prevent noise problems, take special care to provide power supply bypassing at the MCU as Figure 1-5 shows. Place the bypass capacitors as close to the MCU power pins as possible. Use high-frequency-response ceramic capacitor for C_{BYPASS}, C_{BULK} are optional bulk current bypass capacitors for use in applications that require the port pins to source high current level.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0036	PLL Control Register (PTCL)	Read:	PLLIE	PLLF	PLLON	BCS	PRE1	PRE0	VPR1	VPR0
		Write:								
		Reset:	0	0	1	0	0	0	0	0
$0037	PLL Bandwidth Control Register (PBWC)	Read:	AUTO	LOCK	$\overline{ACQ}$	0	0	0	0	R
		Write:								
		Reset:	0	0	0	0	0	0	0	
$0038	PLL Multiplier Select Register High (PMSH)	Read:	0	0	0	0	MUL11	MUL10	MUL9	MUL8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0039	PLL Multiplier Select Register Low (PMSL)	Read:	MUL7	MUL6	MUL5	MUL4	MUL3	MUL2	MUL1	MUL0
		Write:								
		Reset:	0	1	0	0	0	0	0	0
$003A	PLL VCO Range Select Register (PMRS)	Read:	VRS7	VRS6	VRS5	VRS4	VRS3	VRS2	VRS1	VRS0
		Write:								
		Reset:	0	1	0	0	0	0	0	0
$003B	PLL Reference Divider Select Register (PMDS)	Read:	0	0	0	0	RDS3	RDS2	RDS1	RDS0
		Write:								
		Reset:	0	0	0	0	0	0	0	1

= Unimplemented R = Reserved

NOTES:
1. When AUTO = 0, PLLIE is forced clear and is read-only.
2. When AUTO = 0, PLLF and LOCK read as clear.
3. When AUTO = 1, $\overline{ACQ}$ is read-only.
4. When PLLON = 0 or VRS7:VRS0 = $0, BCS is forced clear and is read-only.
5. When PLLON = 1, the PLL programming register is read-only.
6. When BCS = 1, PLLON is forced set and is read-only.

Figure 6-2. CGM I/O Register Summary

6.3.1 Oscillator Module

The oscillator module provides two clock outputs CGMXCLK and CGMRCLK to the CGM module. CGMXCLK when selected, is driven to SIM module to generate the system bus clock. CGMRCLK is used by the phase-lock-loop to provide a higher frequency system bus clock. The oscillator module also provides the reference clock for the timebase module (TBM). See Chapter 5 Oscillator (OSC) for detailed oscillator circuit description. See Chapter 10 Timebase Module (TBM) for detailed description on TBM.

6.3.2 Phase-Locked Loop Circuit (PLL)

The PLL is a frequency generator that can operate in either acquisition mode or tracking mode, depending on the accuracy of the output frequency. The PLL can change between acquisition and tracking modes either automatically or manually.

6.3.3 PLL Circuits

The PLL consists of these circuits:

- Voltage-controlled oscillator (VCO)
- Reference divider
- Frequency pre-scaler
- Modulo VCO frequency divider

During the software execution, it does not consume any dedicated RAM location, the run-time heap will extend the system stack, all other RAM location will not be affected.

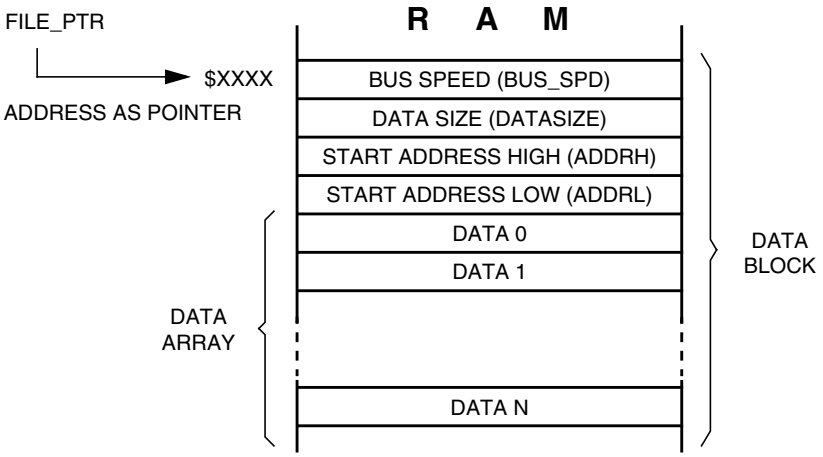

Figure 8-9. Data Block Format for ROM-Resident Routines

The control and data bytes are described below.

- **Bus speed** — This one byte indicates the operating bus speed of the MCU. The value of this byte should be equal to 4 times the bus speed. E.g., for a 4MHz bus, the value is 16 ($10). This control byte is useful where the MCU clock source is switched between the PLL clock and the crystal clock.
- **Data size** — This one byte indicates the number of bytes in the data array that are to be manipulated. The maximum data array size is 255. Routines EE_WRITE and EE_READ are restricted to manipulate a data array between 7 to 15 bytes. Whereas routines ERARNGE and MON_ERARNGE do not manipulate a data array, thus, this data size byte has no meaning.
- **Start address** — These two bytes, high byte followed by low byte, indicate the start address of the FLASH memory to be manipulated.
- **Data array** — This data array contains data that are to be manipulated. Data in this array are programmed to FLASH memory by the programming routines: PRGRNGE, MON_PRGRNGE, EE_WRITE. For the read routines: LDRNGE and EE_READ, data is read from FLASH and stored in this array.

8.5.1 PRGRNGE

PRGRNGE is used to program a range of FLASH locations with data loaded into the data array.

Table 8-11. PRGRNGE Routine

Routine Name	PRGRNGE
Routine Description	Program a range of locations
Calling Address	$FC34
Stack Used	15 bytes
Data Block Format	Bus speed (BUS_SPD) Data size (DATASIZE) Start address high (ADDRH) Start address (ADDRL) Data 1 (DATA1) : Data N (DATAN)

9.9.2 TIM Counter Registers

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

NOTE

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

Address: T1CNTH, $0021 and T2CNTH, $002C

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

Figure 9-5. TIM Counter Registers High (TCNTH)

Address: T1CNTL, $0022 and T2CNTL, $002D

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

Figure 9-6. TIM Counter Registers Low (TCNTL)

9.9.3 TIM Counter Modulo Registers

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

Address: T1MODH, $0023 and T2MODH, $002E

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

Figure 9-7. TIM Counter Modulo Register High (TMODH)

Address: T1MODL, $0024 and T2MODL, $002F

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

Figure 9-8. TIM Counter Modulo Register Low (TMODL)

NOTE

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

NOTE

Before enabling a TIM channel register for input capture operation, make sure that the TCHx pin is stable for at least two bus clocks.

TOVx — Toggle On Overflow Bit

When channel x is an output compare channel, this read/write bit controls the behavior of the channel x output when the TIM counter overflows. When channel x is an input capture channel, TOVx has no effect.

Reset clears the TOVx bit.

1 = Channel x pin toggles on TIM counter overflow

0 = Channel x pin does not toggle on TIM counter overflow

NOTE

When TOVx is set, a TIM counter overflow takes precedence over a channel x output compare if both occur at the same time.

CHxMAX — Channel x Maximum Duty Cycle Bit

When the TOVx bit is at logic 1, setting the CHxMAX bit forces the duty cycle of buffered and unbuffered PWM signals to 100%. As Figure 9-11 shows, the CHxMAX bit takes effect in the cycle after it is set or cleared. The output stays at the 100% duty cycle level until the cycle after CHxMAX is cleared.

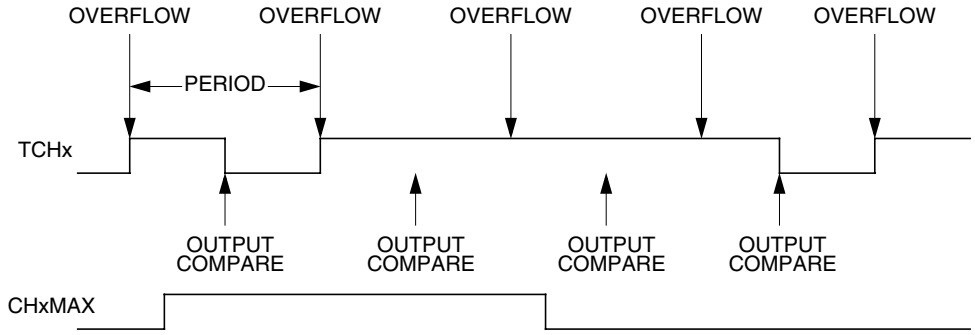

Figure 9-11. CHxMAX Latency

9.9.5 TIM Channel Registers

These read/write registers contain the captured TIM counter value of the input capture function or the output compare value of the output compare function. The state of the TIM channel registers after reset is unknown.

In input capture mode (MSxB:MSxA = 0:0), reading the high byte of the TIM channel x registers (TCHxH) inhibits input captures until the low byte (TCHxL) is read.

In output compare mode (MSxB:MSxA ≠ 0:0), writing to the high byte of the TIM channel x registers (TCHxH) inhibits output compares until the low byte (TCHxL) is written.

TBR[2:0] — Timebase Rate Selection

These read/write bits are used to select the rate of timebase interrupts as shown in Table 10-1.

NOTE

Do not change TBR[2:0] bits while the timebase is enabled (TBON = 1).

Table 10-1. Timebase Rate Selection for OSCCLK = 32.768-kHz

TBR2	TBR1	TBR0	Divider	Timebase Interrupt Rate	
				Hz	ms
0	0	0	262144	0.125	8000
0	0	1	131072	0.25	4000
0	1	0	65536	0.5	2000
0	1	1	32768	1	1000
1	0	0	64	512	~2
1	0	1	32	1024	~1
1	1	0	16	2048	~0.5
1	1	1	8	4096	~0.24

TACK — Timebase ACKnowledge

The TACK bit is a write-only bit and always reads as 0. Writing a logic 1 to this bit clears TBIF, the timebase interrupt flag bit. Writing a logic 0 to this bit has no effect.

1 = Clear timebase interrupt flag
0 = No effect

TBIE — Timebase Interrupt Enabled

This read/write bit enables the timebase interrupt when the TBIF bit becomes set. Reset clears the TBIE bit.

1 = Timebase interrupt enabled
0 = Timebase interrupt disabled

TBON — Timebase Enabled

This read/write bit enables the timebase. Timebase may be turned off to reduce power consumption when its function is not necessary. The counter can be initialized by clearing and then setting this bit. Reset clears the TBON bit.

1 = Timebase enabled
0 = Timebase disabled and the counter initialized to 0’s

11.4.2.4 Idle Characters

An idle character contains all logic 1s and has no start, stop, or parity bit. Idle character length depends on the M bit in SCC1. The preamble is a synchronizing idle character that begins every transmission.

If the TE bit is cleared during a transmission, the TxD pin becomes idle after completion of the transmission in progress. Clearing and then setting the TE bit during a transmission queues an idle character to be sent after the character currently being transmitted.

NOTE

When queueing an idle character, return the TE bit to logic 1 before the stop bit of the current character shifts out to the TxD pin. Setting TE after the stop bit appears on TxD causes data previously written to the SCDR to be lost.

Toggle the TE bit for a queued idle character when the SCTE bit becomes set and just before writing the next byte to the SCDR.

11.4.2.5 Inversion of Transmitted Output

The transmit inversion bit (TXINV) in SCI control register 1 (SCC1) reverses the polarity of transmitted data. All transmitted values, including idle, break, start, and stop bits, are inverted when TXINV is at logic 1. (See 11.8.1 SCI Control Register 1.)

11.4.2.6 Transmitter Interrupts

These conditions can generate CPU interrupt requests from the SCI transmitter:

- SCI transmitter empty (SCTE) — The SCTE bit in SCS1 indicates that the SCDR has transferred a character to the transmit shift register. SCTE can generate a transmitter CPU interrupt request. Setting the SCI transmit interrupt enable bit, SCTIE, in SCC2 enables the SCTE bit to generate transmitter CPU interrupt requests.
- Transmission complete (TC) — The TC bit in SCS1 indicates that the transmit shift register and the SCDR are empty and that no break or idle character has been generated. The transmission complete interrupt enable bit, TCIE, in SCC2 enables the TC bit to generate transmitter CPU interrupt requests.

11.4.3 Receiver

Figure 11-5 shows the structure of the SCI receiver.

11.4.3.1 Character Length

The receiver can accommodate either 8-bit or 9-bit data. The state of the M bit in SCI control register 1 (SCC1) determines character length. When receiving 9-bit data, bit R8 in SCI control register 2 (SCC2) is the ninth bit (bit 8). When receiving 8-bit data, bit R8 is a copy of the eighth bit (bit 7).

11.4.3.2 Character Reception

During an SCI reception, the receive shift register shifts characters in from the RxD pin. The SCI data register (SCDR) is the read-only buffer between the internal data bus and the receive shift register.

After a complete character shifts into the receive shift register, the data portion of the character transfers to the SCDR. The SCI receiver full bit, SCRF, in SCI status register 1 (SCS1) becomes set, indicating that

Start bit verification is not successful if any two of the three verification samples are logic 1s. If start bit verification is not successful, the RT clock is reset and a new search for a start bit begins.

To determine the value of a data bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 11-3 summarizes the results of the data bit samples.

Table 11-3. Data Bit Recovery

RT8, RT9, and RT10 Samples	Data Bit Determination	Noise Flag
000	0	0
001	0	1
010	0	1
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

NOTE

The RT8, RT9, and RT10 samples do not affect start bit verification. If any or all of the RT8, RT9, and RT10 start bit samples are logic 1s following a successful start bit verification, the noise flag (NF) is set and the receiver assumes that the bit is a start bit.

To verify a stop bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 11-4 summarizes the results of the stop bit samples.

Table 11-4. Stop Bit Recovery

RT8, RT9, and RT10 Samples	Framing Error Flag	Noise Flag
000	1	0
001	1	1
010	1	1
011	0	1
100	1	1
101	0	1
110	0	1
111	0	0

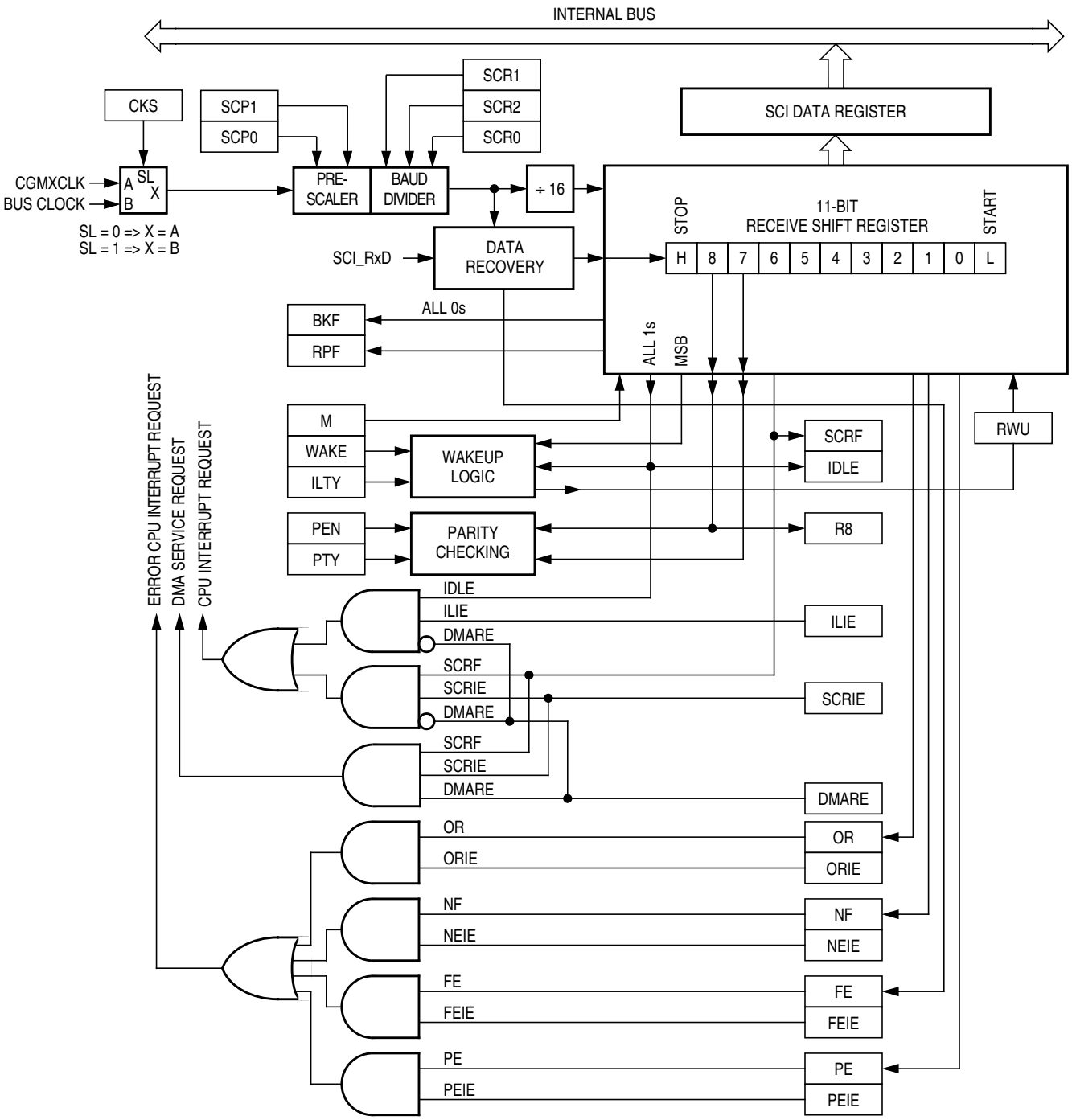

Figure 12-8. SCI Receiver Block Diagram

12.5.3.1 Character Length

The receiver can accommodate either 8-bit or 9-bit data. The state of the M bit in IRSCI control register 1 (IRSCC1) determines character length. When receiving 9-bit data, bit R8 in IRSCI control register 2 (IRSCC2) is the ninth bit (bit 8). When receiving 8-bit data, bit R8 is a copy of the eighth bit (bit 7).

Table 12-2. Start Bit Verification

RT3, RT5, and RT7 Samples	Start Bit Verification	Noise Flag
011	No	0
100	Yes	1
101	No	0
110	No	0
111	No	0

If start bit verification is not successful, the RT clock is reset and a new search for a start bit begins.

To determine the value of a data bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 12-3 summarizes the results of the data bit samples.

Table 12-3. Data Bit Recovery

RT8, RT9, and RT10 Samples	Data Bit Determination	Noise Flag
000	0	0
001	0	1
010	0	1
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

NOTE

The RT8, RT9, and RT10 samples do not affect start bit verification. If any or all of the RT8, RT9, and RT10 start bit samples are logic 1s following a successful start bit verification, the noise flag (NF) is set and the receiver assumes that the bit is a start bit.

To verify a stop bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 12-4 summarizes the results of the stop bit samples.

Table 12-4. Stop Bit Recovery

RT8, RT9, and RT10 Samples	Framing Error Flag	Noise Flag
000	1	0
001	1	1

- The shift register is cleared.
- The SPI state counter is cleared, making it ready for a new complete transmission.
- All the SPI port logic is defaulted back to being general-purpose I/O.

These items are reset only by a system reset:

- All control bits in the SPCR register
- All control bits in the SPSCR register (MODFEN, ERRIE, SPR1, and SPR0)
- The status flags SPRF, OVRF, and MODF

By not resetting the control bits when SPE is low, the user can clear SPE between transmissions without having to set all control bits again when SPE is set back high for the next transmission.

By not resetting the SPRF, OVRF, and MODF flags, the user can still service these interrupts after the SPI has been disabled. The user can disable the SPI by writing 0 to the SPE bit. The SPI can also be disabled by a mode fault occurring in an SPI that was configured as a master with the MODFEN bit set.

13.10 Low-Power Modes

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

13.10.1 Wait Mode

The SPI module remains active after the execution of a WAIT instruction. In wait mode the SPI module registers are not accessible by the CPU. 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.

To exit wait mode when an overflow condition occurs, enable the OVRF bit to generate CPU interrupt requests by setting the error interrupt enable bit (ERRIE). (See 13.8 Interrupts.)

13.10.2 Stop Mode

The SPI module is inactive after the execution of a STOP instruction. The STOP instruction does not affect register conditions. 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.

13.11 SPI During Break Interrupts

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 Chapter 7 System Integration Module (SIM).)

To allow software to clear status bits during a break interrupt, write a logic 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 logic 0 to the BCFE bit. With BCFE at logic 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 logic 0. After the break, doing the second step clears the status bit.

14.9 SMBus Protocol Implementation

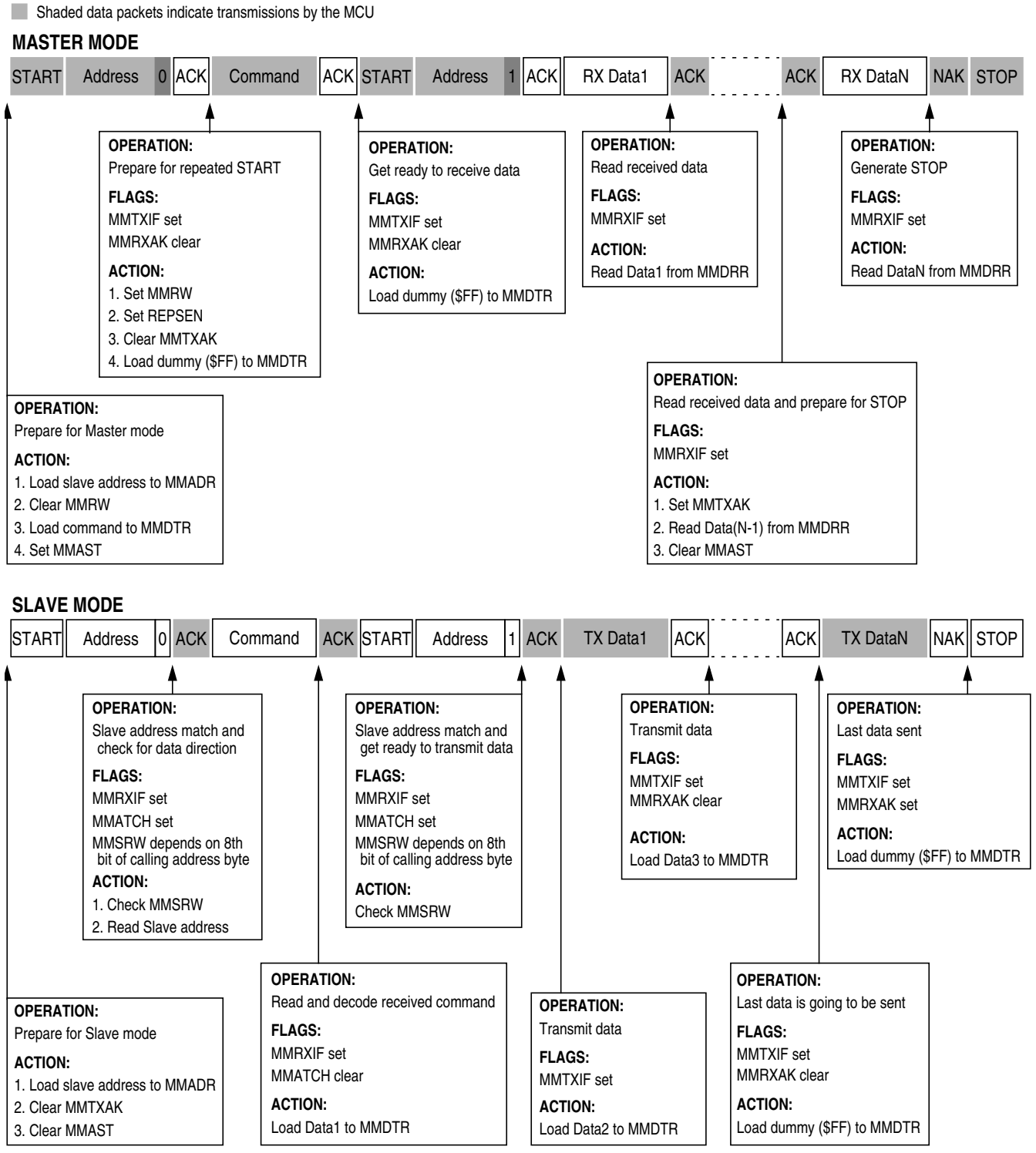

Figure 14-20. SMBus Protocol Implementation

Chapter 17
External Interrupt (IRQ)

17.1 Introduction

The external interrupt (IRQ) module provides two maskable interrupt inputs: $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$.

17.2 Features

Features of the IRQ module include:

- A dedicated external interrupt pin, $\overline{\text{IRQ1}}$
- An external interrupt pin shared with a port pin, PTC0/$\overline{\text{IRQ2}}$
- Separate IRQ interrupt control bits for $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$
- Hysteresis buffers
- Programmable edge-only or edge and level interrupt sensitivity
- Automatic interrupt acknowledge
- Internal pullup resistor, with disable option on $\overline{\text{IRQ2}}$

NOTE

References to either IRQ1 or IRQ2 may be made in the following text by omitting the IRQ number. For example, IRQF may refer generically to IRQ1F and IRQ2F, and IMASK may refer to IMASK1 and IMASK2.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001C	IRQ2 Status and Control Register (INTSCR2)	Read:	0	PUC0ENB	0	0	IRQ2F	0	IMASK2	MODE2
		Write:						ACK2		
		Reset:	0	0	0	0	0	0	0	0
$001E	IRQ1 Status and Control Register (INTSCR1)	Read:	0	0	0	0	IRQ1F	0	IMASK1	MODE1
		Write:						ACK1		
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 17-1. External Interrupt I/O Register Summary

17.3 Functional Description

A logic 0 applied to the external interrupt pin can latch a CPU interrupt request. Figure 17-2 and Figure 17-3 shows the structure of the IRQ module.

Interrupt signals on the $\overline{\text{IRQ}}$ pin are latched into the IRQ latch. An interrupt latch remains set until one of the following actions occurs:

- Vector fetch — A vector fetch automatically generates an interrupt acknowledge signal that clears the latch that caused the vector fetch.

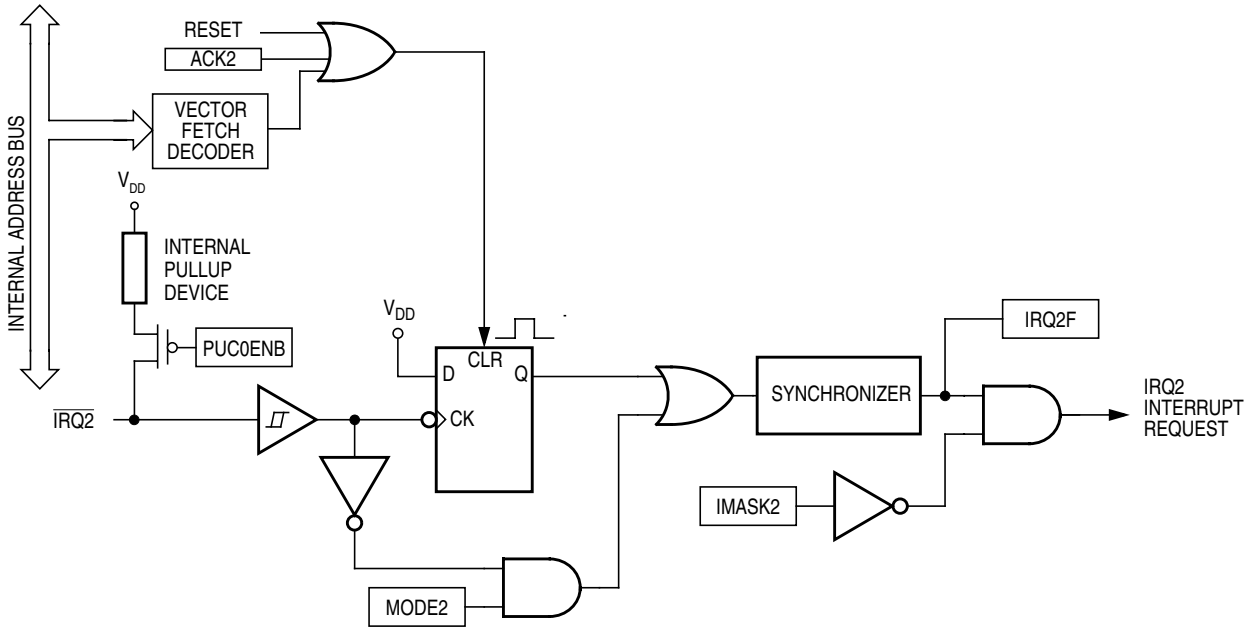

Figure 17-3. IRQ2 Block Diagram

17.4 $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$ Pins

A logic 0 on the $\overline{\text{IRQ}}$ pin can latch an interrupt request into the IRQ latch. A vector fetch, software clear, or reset clears the IRQ latch.

If the MODE bit is set, the $\overline{\text{IRQ}}$ pin is both falling-edge-sensitive and low-level-sensitive. With MODE set, both of the following actions must occur to clear IRQ:

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the latch. Software may generate the interrupt acknowledge signal by writing a logic 1 to the ACK bit in the interrupt status and control register (INTSCR). The ACK bit is useful in applications that poll the $\overline{\text{IRQ}}$ pin and require software to clear the IRQ latch. Writing to the ACK bit prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACK does not affect subsequent transitions on the $\overline{\text{IRQ}}$ pin. A falling edge that occurs after writing to the ACK bit another interrupt request. If the IRQ mask bit, IMASK, is clear, the CPU loads the program counter with the vector address at location defined in Table 2-1 . Vector Addresses.
- Return of the $\overline{\text{IRQ}}$ pin to logic 1 — As long as the $\overline{\text{IRQ}}$ pin is at logic 0, IRQ remains active.

The vector fetch or software clear and the return of the $\overline{\text{IRQ}}$ pin to logic 1 may occur in any order. The interrupt request remains pending as long as the $\overline{\text{IRQ}}$ pin is at logic 0. A reset will clear the latch and the MODE control bit, thereby clearing the interrupt even if the pin stays low.

If the MODE bit is clear, the $\overline{\text{IRQ}}$ pin is falling-edge-sensitive only. With MODE clear, a vector fetch or software clear immediately clears the IRQ latch.

The IRQF bit in the INTSCR register can be used to check for pending interrupts. The IRQF bit is not affected by the IMASK bit, which makes it useful in applications where polling is preferred.

Use the BIH or BIL instruction to read the logic level on the $\overline{\text{IRQ1}}$ pin.

NOTE

The BIH and BIL instructions do not read the logic level on the $\overline{\text{IRQ2}}$ pin.

NOTE

When using the level-sensitive interrupt trigger, avoid false interrupts by masking interrupt requests in the interrupt routine.

The $\overline{\text{IRQ1}}$ pin has a permanent internal pullup device connected, while the $\overline{\text{IRQ2}}$ pin has an optional pullup device that can be enabled or disabled by the PUC0ENB bit in the INTSCR2 register.

17.5 IRQ Module During Break Interrupts

The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear the latch during the break state. (See Chapter 21 Break Module (BRK).)

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

To protect CPU interrupt flags during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), writing to the ACK bit in the IRQ status and control register during the break state has no effect on the IRQ interrupt flags.

17.6 IRQ Registers

Each IRQ is controlled and monitored by an status and control register.

- IRQ1 Status and Control Register — $001E
- IRQ2 Status and Control Register — $001C

17.6.1 IRQ1 Status and Control Register

The IRQ1 status and control register (INTSCR1) controls and monitors operation of IRQ1. The INTSCR1 has the following functions:

- Shows the state of the IRQ1 flag
- Clears the IRQ1 latch
- Masks IRQ1 interrupt request
- Controls triggering sensitivity of the $\overline{\text{IRQ1}}$ interrupt pin

Address: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	IRQ1F	0	IMASK1	MODE1
Write:						ACK1		
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 17-4. IRQ1 Status and Control Register (INTSCR1)

IRQ1F — IRQ1 Flag Bit

This read-only status bit is high when the IRQ1 interrupt is pending.

1 = $\overline{\text{IRQ1}}$ interrupt pending
0 = $\overline{\text{IRQ1}}$ interrupt not pending

ACK1 — IRQ1 Interrupt Request Acknowledge Bit

Writing a logic 1 to this write-only bit clears the IRQ1 latch. ACK1 always reads as logic 0. Reset clears ACK1.

18.6 Low-Power Modes

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

18.6.1 Wait Mode

The keyboard interrupt module remains active in wait mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of wait mode.

18.6.2 Stop Mode

The keyboard interrupt module remains active in stop mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of stop mode.

18.7 Keyboard Module During Break Interrupts

The system integration module (SIM) controls whether the keyboard interrupt latch can be cleared during the break state. The BCFE bit in the SIM break flag control register (BFCR) enables software to clear status bits during the break state.

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

To protect the latch during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), writing to the keyboard acknowledge bit (ACKK) in the keyboard status and control register during the break state has no effect.

22.8 5V ADC Electrical Characteristics

Table 22-7. ADC Electrical Characteristics (5V)

Characteristic[1]	Symbol	Min	Max	Unit	Notes
Supply voltage	V_{DDA}	4.5	5.5	V	V_{DDA} is an dedicated pin and should be tied to V_{DD} on the PCB with proper decoupling.
Input range	V_{ADIN}	0	V_{DDA}	V	$V_{ADIN} \leq V_{DDA}$
Resolution	B_{AD}	10	10	bits	
Absolute accuracy	A_{AD}	—	± 1.5	LSB	Includes quantization. ±0.5 LSB = ±1 ADC step.
ADC internal clock	f_{ADIC}	500k	1.048M	Hz	$t_{ADIC} = 1/f_{ADIC}$
Conversion range	R_{AD}	V_{REFL}	V_{REFH}	V	
ADC voltage reference high	V_{REFH}	—	V_{DDA} + 0.1	V	
ADC voltage reference low	V_{REFL}	V_{SSA} – 0.1	—	V	
Conversion time	t_{ADC}	16	17	t_{ADIC} cycles	
Sample time	t_{ADS}	5	—	t_{ADIC} cycles	
Monotonicity	M_{AD}	Guaranteed			
Zero input reading	Z_{ADI}	000	001	HEX	$V_{ADIN} = V_{REFL}$
Full-scale reading	F_{ADI}	3FD	3FF	HEX	$V_{ADIN} = V_{REFH}$
Input capacitance	C_{ADI}	—	20	pF	Not tested.
Input impedance	R_{ADI}	20M	—	Ω	
V_{REFH}/V_{REFL}	I_{VREF}	—	1.6	mA	Not tested.

1. V_{DD} = 4.5 to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.

22.15 5V SPI Characteristics

Table 22-15. SPI Characteristics (5V)

Diagram Number[1]	Characteristic[2]	Symbol	Min	Max	Unit
	Operating frequency Master Slave	 $f_{OP(M)}$ $f_{OP(S)}$	 $f_{OP}/128$ dc	 $f_{OP}/2$ f_{OP}	 MHz MHz
1	Cycle time Master Slave	 $t_{CYC(M)}$ $t_{CYC(S)}$	 2 1	 128 —	 t_{CYC} t_{CYC}
2	Enable lead time	$t_{Lead(S)}$	1	—	t_{CYC}
3	Enable lag time	$t_{Lag(S)}$	1	—	t_{CYC}
4	Clock (SPSCK) high time Master Slave	 $t_{SCKH(M)}$ $t_{SCKH(S)}$	 t_{CYC} –25 1/2 t_{CYC} –25	 64 t_{CYC} —	 ns ns
5	Clock (SPSCK) low time Master Slave	 $t_{SCKL(M)}$ $t_{SCKL(S)}$	 t_{CYC} –25 1/2 t_{CYC} –25	 64 t_{CYC} —	 ns ns
6	Data setup time (inputs) Master Slave	 $t_{SU(M)}$ $t_{SU(S)}$	 30 30	 — —	 ns ns
7	Data hold time (inputs) Master Slave	 $t_{H(M)}$ $t_{H(S)}$	 30 30	 — —	 ns ns
8	Access time, slave[3] CPHA = 0 CPHA = 1	 $t_{A(CP0)}$ $t_{A(CP1)}$	 0 0	 40 40	 ns ns
9	Disable time, slave[4]	$t_{DIS(S)}$	—	40	ns
10	Data valid time, after enable edge Master Slave[5]	 $t_{V(M)}$ $t_{V(S)}$	 — —	 50 50	 ns ns
11	Data hold time, outputs, after enable edge Master Slave	 $t_{HO(M)}$ $t_{HO(S)}$	 0 0	 — —	 ns ns

1. Numbers refer to dimensions in Figure 22-3 and Figure 22-4.
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless noted; 100 pF load on all SPI pins.
3. Time to data active from high-impedance state
4. Hold time to high-impedance state
5. With 100 pF on all SPI pins