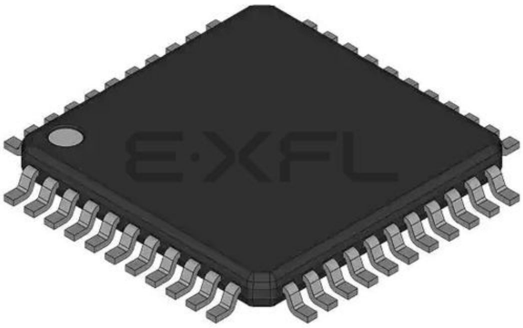
Freescale Semiconductor - MC908AP64ACBE Datasheet



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	30
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	42-SDIP (0.600", 15.24mm)
Supplier Device Package	42-PDIP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908ap64acbe

Chapter 23
Mechanical Specifications

Chapter 24
Ordering Information

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0026	Timer 1 Channel 0 Register High (T1CH0H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0027	Timer 1 Channel 0 Register Low (T1CH0L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$0028	Timer 1 Channel 1 Status and Control Register (T1SC1)	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0		▨					
		Reset:	0	0	0	0	0	0	0	0
$0029	Timer 1 Channel 1 Register High (T1CH1H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$002A	Timer 1 Channel 1 Register Low (T1CH1L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$002B	Timer 2 Status and Control Register (T2SC)	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST	▨			
		Reset:	0	0	1	0	0	0	0	0
$002C	Timer 2 Counter Register High (T2CNTH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	0	0	0	0	0	0	0	0
$002D	Timer 2 Counter Register Low (T2CNTL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	0	0	0	0	0	0	0	0
$002E	Timer 2 Counter Modulo Register High (T2MODH)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$002F	Timer 2 Counter Modulo Register Low (T2MODL)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0030	Timer 2 Channel 0 Status and Control Register (T2SC0)	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0031	Timer 2 Channel 0 Register High (T2CH0H)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0032	Timer 2 Channel 0 Register Low (T2CH0L)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							

U = Unaffected X = Indeterminate ▨ = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 4 of 9)

Table 4-2. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write						Control		Register/Memory							
	DIR	DIR	REL	DIR	INH	INH	IX1	SP1	IX	INH	INH	IMM	DIR	EXT	IX2	SP2	IX1	SP1	IX
MSB LSB	0	1	2	3	4	5	6	9E6	7	8	9	A	B	C	D	9ED	E	9EE	F
0	5 BRSET0 3 DIR	4 BSET0 2 DIR	3 BRA 2 REL	4 NEG 2 DIR	1 NEGA 1 INH	1 NEGX 1 INH	4 NEG 2 IX1	5 NEG 3 SP1	3 NEG 1 IX	7 RTI 1 INH	3 BGE 2 REL	2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	4 SUB 3 IX2	5 SUB 4 SP2	3 SUB 2 IX1	4 SUB 3 SP1	2 SUB 1 IX
1	5 BRCLR0 3 DIR	4 BCLR0 2 DIR	3 BRN 2 REL	5 CBEQ 3 DIR	4 CBEQA 3 IMM	4 CBEQX 3 IMM	5 CBEQ 3 IX1+	6 CBEQ 4 SP1	4 CBEQ 2 IX+	4 RTS 1 INH	3 BLT 2 REL	2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	4 CMP 3 IX2	5 CMP 4 SP2	3 CMP 2 IX1	4 CMP 3 SP1	2 CMP 1 IX
2	5 BRSET1 3 DIR	4 BSET1 2 DIR	3 BHI 2 REL		5 MUL 1 INH	7 DIV 1 INH	3 NSA 1 INH		2 DAA 1 INH		3 BGT 2 REL	2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	4 SBC 3 IX2	5 SBC 4 SP2	3 SBC 2 IX1	4 SBC 3 SP1	2 SBC 1 IX
3	5 BRCLR1 3 DIR	4 BCLR1 2 DIR	3 BLS 2 REL	4 COM 2 DIR	1 COMA 1 INH	1 COMX 1 INH	4 COM 2 IX1	5 COM 3 SP1	3 COM 1 IX	9 SWI 1 INH	3 BLE 2 REL	2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	4 CPX 3 IX2	5 CPX 4 SP2	3 CPX 2 IX1	4 CPX 3 SP1	2 CPX 1 IX
4	5 BRSET2 3 DIR	4 BSET2 2 DIR	3 BCC 2 REL	4 LSR 2 DIR	1 LSRA 1 INH	1 LSRX 1 INH	4 LSR 2 IX1	5 LSR 3 SP1	3 LSR 1 IX	2 TAP 1 INH	2 TXS 1 INH	2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	4 AND 3 IX2	5 AND 4 SP2	3 AND 2 IX1	4 AND 3 SP1	2 AND 1 IX
5	5 BRCLR2 3 DIR	4 BCLR2 2 DIR	3 BCS 2 REL	4 STHX 2 DIR	3 LDHX 3 IMM	4 LDHX 2 DIR	3 CPHX 3 IMM		4 CPHX 2 DIR	1 TPA 1 INH	2 TSX 1 INH	2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	4 BIT 3 IX2	5 BIT 4 SP2	3 BIT 2 IX1	4 BIT 3 SP1	2 BIT 1 IX
6	5 BRSET3 3 DIR	4 BSET3 2 DIR	3 BNE 2 REL	4 ROR 2 DIR	1 RORA 1 INH	1 RORX 1 INH	4 ROR 2 IX1	5 ROR 3 SP1	3 ROR 1 IX	2 PULA 1 INH		2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	4 LDA 3 IX2	5 LDA 4 SP2	3 LDA 2 IX1	4 LDA 3 SP1	2 LDA 1 IX
7	5 BRCLR3 3 DIR	4 BCLR3 2 DIR	3 BEQ 2 REL	4 ASR 2 DIR	1 ASRA 1 INH	1 ASRX 1 INH	4 ASR 2 IX1	5 ASR 3 SP1	3 ASR 1 IX	2 PSHA 1 INH	1 TAX 1 INH	2 AIS 2 IMM	3 STA 2 DIR	4 STA 3 EXT	4 STA 3 IX2	5 STA 4 SP2	3 STA 2 IX1	4 STA 3 SP1	2 STA 1 IX
8	5 BRSET4 3 DIR	4 BSET4 2 DIR	3 BHCC 2 REL	4 LSL 2 DIR	1 LSLA 1 INH	1 LSLX 1 INH	4 LSL 2 IX1	5 LSL 3 SP1	3 LSL 1 IX	2 PULX 1 INH	1 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	4 EOR 3 IX2	5 EOR 4 SP2	3 EOR 2 IX1	4 EOR 3 SP1	2 EOR 1 IX
9	5 BRCLR4 3 DIR	4 BCLR4 2 DIR	3 BHCS 2 REL	4 ROL 2 DIR	1 ROLA 1 INH	1 ROLX 1 INH	4 ROL 2 IX1	5 ROL 3 SP1	3 ROL 1 IX	2 PSHX 1 INH	1 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	4 ADC 3 IX2	5 ADC 4 SP2	3 ADC 2 IX1	4 ADC 3 SP1	2 ADC 1 IX
A	5 BRSET5 3 DIR	4 BSET5 2 DIR	3 BPL 2 REL	4 DEC 2 DIR	1 DECA 1 INH	1 DECX 1 INH	4 DEC 2 IX1	5 DEC 3 SP1	3 DEC 1 IX	2 PULH 1 INH	2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	4 ORA 3 IX2	5 ORA 4 SP2	3 ORA 2 IX1	4 ORA 3 SP1	2 ORA 1 IX
B	5 BRCLR5 3 DIR	4 BCLR5 2 DIR	3 BMI 2 REL	5 DBNZ 3 DIR	3 DBNZA 2 INH	3 DBNZX 2 INH	5 DBNZ 3 IX1	6 DBNZ 4 SP1	4 DBNZ 2 IX	2 PSHH 1 INH	2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	4 ADD 3 IX2	5 ADD 4 SP2	3 ADD 2 IX1	4 ADD 3 SP1	2 ADD 1 IX
C	5 BRSET6 3 DIR	4 BSET6 2 DIR	3 BMC 2 REL	4 INC 2 DIR	1 INCA 1 INH	1 INCX 1 INH	4 INC 2 IX1	5 INC 3 SP1	3 INC 1 IX	1 CLRH 1 INH	1 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2		3 JMP 2 IX1		2 JMP 1 IX
D	5 BRCLR6 3 DIR	4 BCLR6 2 DIR	3 BMS 2 REL	3 TST 2 DIR	1 TSTA 1 INH	1 TSTX 1 INH	3 TST 2 IX1	4 TST 3 SP1	2 TST 1 IX		1 NOP 1 INH	4 BSR 2 REL	4 JSR 2 DIR	5 JSR 3 EXT	6 JSR 3 IX2		5 JSR 2 IX1		4 JSR 1 IX
E	5 BRSET7 3 DIR	4 BSET7 2 DIR	3 BIL 2 REL		5 MOV 3 DD	4 MOV 2 DIX+	4 MOV 3 IMD		4 MOV 2 IX+D	1 STOP 1 INH	*	2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	4 LDX 3 IX2	5 LDX 4 SP2	3 LDX 2 IX1	4 LDX 3 SP1	2 LDX 1 IX
F	5 BRCLR7 3 DIR	4 BCLR7 2 DIR	3 BIH 2 REL	3 CLR 2 DIR	1 CLRA 1 INH	1 CLRX 1 INH	3 CLR 2 IX1	4 CLR 3 SP1	2 CLR 1 IX	1 WAIT 1 INH	1 TXA 1 INH	2 AIX 2 IMM	3 STX 2 DIR	4 STX 3 EXT	4 STX 3 IX2	5 STX 4 SP2	3 STX 2 IX1	4 STX 3 SP1	2 STX 1 IX

INH	Inherent	REL	Relative	SP1	Stack Pointer, 8-Bit Offset
IMM	Immediate	IX	Indexed, No Offset	SP2	Stack Pointer, 16-Bit Offset
DIR	Direct	IX1	Indexed, 8-Bit Offset	IX+	Indexed, No Offset with Post Increment
EXT	Extended	IX2	Indexed, 16-Bit Offset		
DD	Direct-Direct	IMD	Immediate-Direct	IX1+	Indexed, 1-Byte Offset with Post Increment
IX+D	Indexed-Direct	DIX+	Direct-Indexed		

*Pre-byte for stack pointer indexed instructions

	MSB LSB	0	High Byte of Opcode in Hexadecimal
Low Byte of Opcode in Hexadecimal	0	5 BRSET0 3 DIR	Cycles Opcode Mnemonic Number of Bytes / Addressing Mode

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	$\overline{\text{ACQ}}$	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, $\overline{\text{PLLF}}$ and LOCK read as clear.
3. When AUTO = 1, $\overline{\text{ACQ}}$ 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

Chapter 7
System Integration Module (SIM)

7.1 Introduction

This section describes the system integration module (SIM). Together with the CPU, the SIM controls all MCU activities. A block diagram of the SIM is shown in Figure 7-1. Figure 7-2 is a summary of the SIM input/output (I/O) registers. The SIM is a system state controller that coordinates CPU and exception timing. The SIM is responsible for:

- Bus clock generation and control for CPU and peripherals:
  - Stop/wait/reset/break entry and recovery
  - Internal clock control
- Master reset control, including power-on reset (POR) and COP timeout
- Interrupt control:
  - Acknowledge timing
  - Arbitration control timing
  - Vector address generation
- CPU enable/disable timing

Table 7-1 shows the internal signal names used in this section.

Table 7-1. Signal Name Conventions

Signal Name	Description
ICLK	Internal oscillator clock
CGMXCLK	Selected oscillator clock from oscillator module
CGMVCLK, CGMPCLK	PLL output and the divided PLL output
CGMOUT	CGMPCLK-based or oscillator-based clock output from CGM module (Bus clock = CGMOUT ÷ 2)
IAB	Internal address bus
IDB	Internal data bus
PORRST	Signal from the power-on reset module to the SIM
IRST	Internal reset signal
$R/\overline{W}$	Read/write signal

7.2.2 Clock Start-up from POR or LVI Reset

When the power-on reset module or the low-voltage inhibit module generates a reset, the clocks to the CPU and peripherals are inactive and held in an inactive phase until after the 4096 ICLK cycle POR timeout has completed. The $\overline{\text{RST}}$ pin is driven low by the SIM during this entire period. The IBUS clocks start upon completion of the timeout.

7.2.3 Clocks in Stop Mode and Wait Mode

Upon exit from stop mode by an interrupt, break, or reset, the SIM allows ICLK to clock the SIM counter. The CPU and peripheral clocks do not become active until after the stop delay timeout. This timeout is selectable as 4096 or 32 ICLK cycles. (See 7.6.2 Stop Mode.)

In wait mode, the CPU clocks are inactive. The SIM also produces two sets of clocks for other modules. Refer to the wait mode subsection of each module to see if the module is active or inactive in wait mode. Some modules can be programmed to be active in wait mode.

7.3 Reset and System Initialization

The MCU has these reset sources:

- Power-on reset module (POR)
- External reset pin ($\overline{\text{RST}}$)
- Computer operating properly module (COP)
- Low-voltage inhibit module (LVI)
- Illegal opcode
- Illegal address

All of these resets produce the vector $FFFE:$FFFF ($FEFE:$FEFF in monitor mode) and assert the internal reset signal (IRST). IRST causes all registers to be returned to their default values and all modules to be returned to their reset states.

An internal reset clears the SIM counter (see 7.4 SIM Counter), but an external reset does not. Each of the resets sets a corresponding bit in the SIM reset status register (SRSR). (See 7.7 SIM Registers.)

7.3.1 External Pin Reset

The $\overline{\text{RST}}$ pin circuit includes an internal pull-up device. Pulling the asynchronous $\overline{\text{RST}}$ pin low halts all processing. The PIN bit of the SIM reset status register (SRSR) is set as long as $\overline{\text{RST}}$ is held low for at least the minimum t_{RL}time and no other reset sources are present. See Table 7-2 for details. Figure 7-4 shows the relative timing.

Table 7-2. Reset Recovery

Reset Recovery Type	Actual Number of Cycles
POR/LVI	4163 (4096 + 64 + 3)
All others	67 (64 + 3)

the received byte can be read. If the SCI receive interrupt enable bit, SCRIE, in SCC2 is also set, the SCRF bit generates a receiver CPU interrupt request.

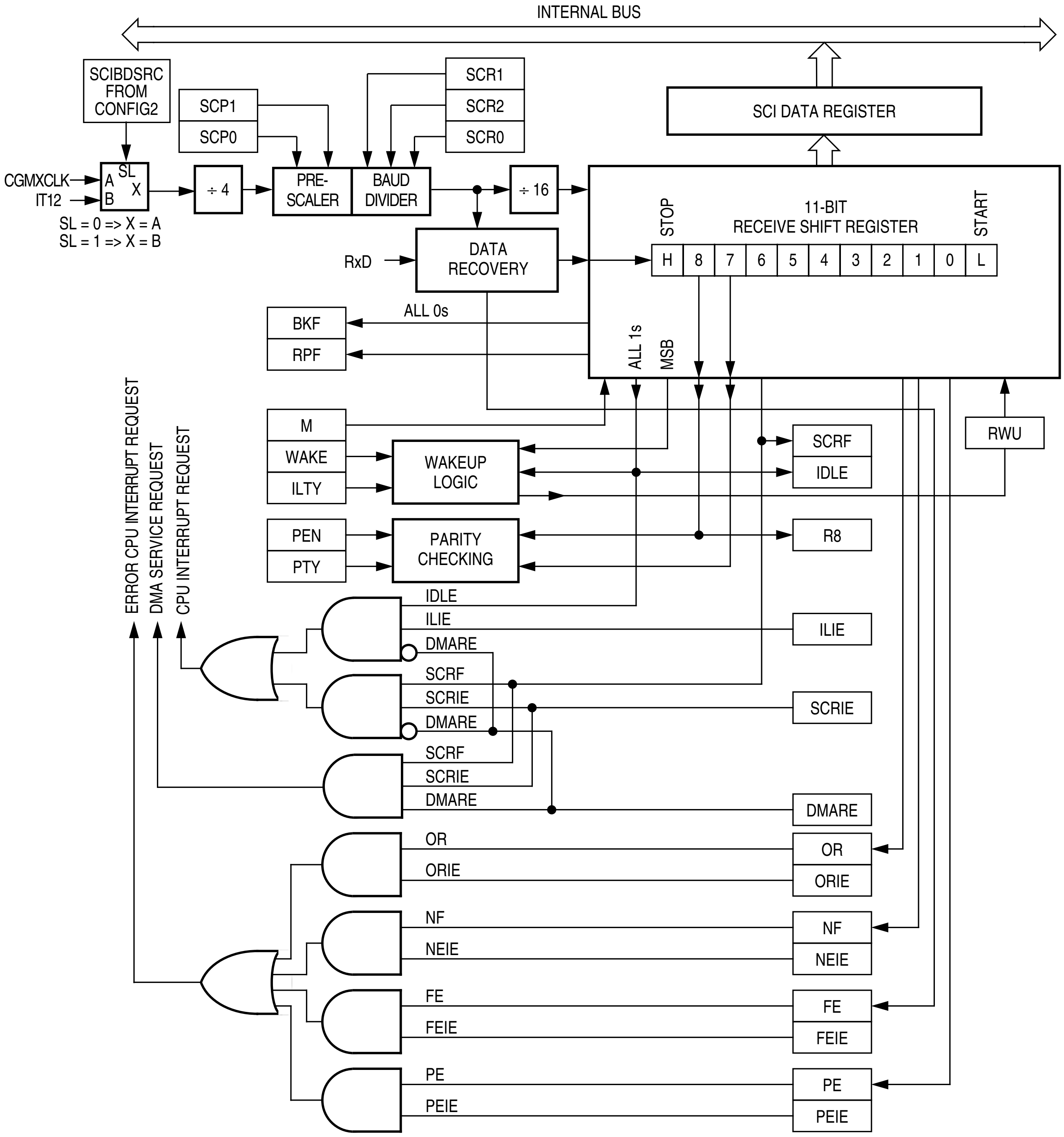


Figure 11-5. SCI Receiver Block Diagram

- 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.

Only the slave with a matched address will respond by sending back an acknowledge bit by pulling SDA low on the 9th clock cycle.
(See Figure 14-2.)

14.5.3 Data Transfer

Once a successful slave addressing is achieved, the data transfer can proceed byte by byte in the direction specified by the R/W-bit sent by the calling master.

Each data byte is 8 bits. Data can be changed only when SCL is low and must be held stable when SCL is high as shown in Figure 14-2. The MSB is transmitted first and each byte has to be followed by an acknowledge bit. This is signalled by the receiving device by pulling the SDA low on the 9th clock cycle. Therefore, one complete data byte transfer requires 9 clock cycles.

If the slave receiver does not acknowledge the master, the SDA line should be left high by the slave. The master can then generate a STOP signal to abort the data transfer or a START signal (repeated START) to commence a new transfer.

If the master receiver does not acknowledge the slave transmitter after a byte has been transmitted, it means an “end of data” to the slave. The slave should release the SDA line for the master to generate a STOP or START signal.

14.5.4 Repeated START Signal

As shown in Figure 14-2, a repeated START signal is used to generate START signal without first generating a STOP to terminate the communication. This is used by the master to communicate with another slave or with the same slave in a different mode (transmit/receive mode) without releasing the bus.

14.5.5 STOP Signal

The master can terminate the communication by generating a STOP signal to free the bus. However, the master may generate a START signal followed by a calling command without first generating a STOP signal. This is called repeat START. A STOP signal is defined as a low to high transition of SDA while SCL is at logic high (see Figure 14-2).

14.5.6 Arbitration Procedure

The interface circuit is a multi-master system which allows more than one master to be connected. If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus clock. The clock low period is equal to the longest clock low period and the clock high period is equal to the shortest one among the masters. A data arbitration procedure determines the priority. A master will lose arbitration if it transmits a logic 1 while another transmits a logic 0. The losing master will immediately switch over to slave receive mode and stops its data and clock outputs. The transition from master to slave will not generate a STOP condition. Meanwhile a software bit will be set by hardware to indicates loss of arbitration.

14.5.7 Clock Synchronization

Since wired-AND logic is performed on SCL line, a high to low transition on the SCL line will affect the devices connected to the bus. The devices start counting their low period once a device’s clock has gone low, it will hold the SCL line low until the clock high state is reached. However, the change of low to high

14.6.1 MMIIC Address Register (MMADR)

Address: $0048

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	MMAD7	MMAD6	MMAD5	MMAD4	MMAD3	MMAD2	MMAD1	MMEXTAD
Reset:	1	0	1	0	0	0	0	0

Figure 14-4. MMIIC Address Register (MMADR)

MMAD[7:1] — Multi-Master Address

These seven bits represent the MMIIC interface's own specific slave address when in slave mode, and the calling address when in master mode. Software must update MMAD[7:1] as the calling address while entering master mode and restore its own slave address after master mode is relinquished. This register is cleared as $A0 upon reset.

MMEXTAD — Multi-Master Expanded Address

This bit is set to expand the address of the MMIIC in slave mode. When set, the MMIIC will acknowledge the following addresses from a calling master: $MMAD[7:1], 0000000, and 0001100. Reset clears this bit.

1 = MMIIC responds to the following calling addresses:
$MMAD[7:1], 0000000, and 0001100.
0 = MMIIC responds to address $MMAD[7:1]

For example, when MMADR is configured as:

MMAD7	MMAD6	MMAD5	MMAD4	MMAD3	MMAD2	MMAD1	MMEXTAD
1	1	0	1	0	1	0	1

The MMIIC module will respond to the calling address:

Bit 7	6	5	4	3	2	Bit 1
1	1	0	1	0	1	0

or the general calling address:

0	0	0	0	0	0	0

or the calling address:

Bit 7	6	5	4	3	2	Bit 1
0	0	0	1	1	0	0

Note that bit-0 of the 8-bit calling address is the MMRW bit from the calling master.

Table 14-2. MMIIC Baud Rate Selection

MMBR2	MMBR1	MMBR0	Divider	MMIIC Baud Rates for Bus Clocks:			
				8 MHz	4 MHz	2 MHz	1 MHz
0	0	0	20	400kHz	200kHz	100kHz	50kHz
0	0	1	40	200kHz	100kHz	50kHz	25kHz
0	1	0	80	100kHz	50kHz	25kHz	12.5kHz
0	1	1	160	50kHz	25kHz	12.5kHz	6.25kHz
1	0	0	320	25kHz	12.5kHz	6.25kHz	3.125kHz
1	0	1	640	12.5kHz	6.25kHz	3.125kHz	1.5625kHz
1	1	0	1280	6.25kHz	3.125kHz	1.5625kHz	0.78125kHz
1	1	1	2560	3.125kHz	1.5625kHz	0.78125kHz	0.3906kHz

NOTE

The frequency of the MMIIC baud rate is only guaranteed for 100 kHz to 10 kHz. The divider is available for the flexibility on bus frequency selection.

14.7 Program Algorithm

When the MMIIC module detects an arbitration loss in master mode, it releases both SDA and SCL lines immediately. But if there are no further STOP conditions detected, the module will hang up. Therefore, it is recommended to have time-out software to recover from this condition. The software can start the time-out counter by looking at the MMBB (bus busy) flag and reset the counter on the completion of one byte transmission. If a time-out has occurred, software can clear the MMEN bit (disable MMIIC module) to release the bus, and hence clear the MMBB flag. This is the only way to clear the MMBB flag by software if the module hangs up due to a no STOP condition received. The MMIIC can resume operation again by setting the MMEN bit.

14.8 SMBus Protocols with PEC and without PEC

Following is a description of the various MMIIC bus protocols with and without a packet error code (PEC).

14.8.1 Quick Command

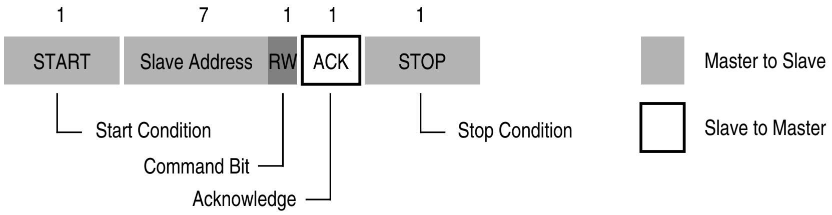
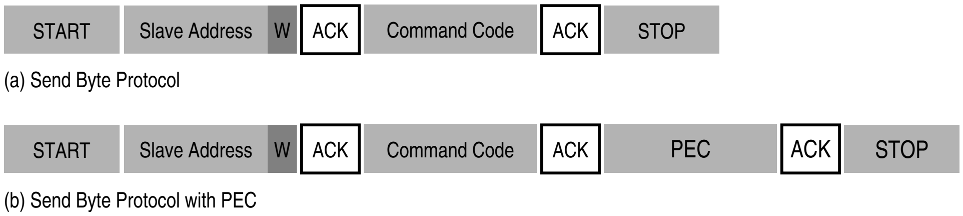


Figure 14-13. Quick Command

14.8.2 Send Byte

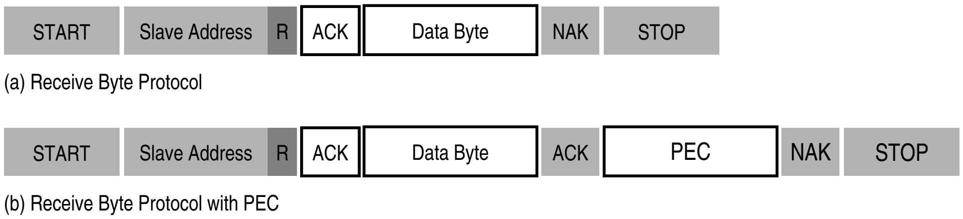


(a) Send Byte Protocol

(b) Send Byte Protocol with PEC

Figure 14-14. Send Byte

14.8.3 Receive Byte



(a) Receive Byte Protocol

(b) Receive Byte Protocol with PEC

Figure 14-15. Receive Byte

15.5.1 Wait Mode

The ADC continues normal operation in wait mode. Any enabled CPU interrupt request from the ADC can bring the MCU out of wait mode. If the ADC is not required to bring the MCU out of wait mode, power down the ADC by setting the ADCH[4:0] bits to logic 1's before executing the WAIT instruction.

15.5.2 Stop Mode

The ADC module is inactive after the execution of a STOP instruction. Any pending conversion is aborted. ADC conversions resume when the MCU exits stop mode. Allow one conversion cycle to stabilize the analog circuitry before attempting a new ADC conversion after exiting stop mode.

15.6 I/O Signals

The ADC module has eight channels shared with port A I/O pins.

15.6.1 ADC Voltage In (V_{ADIN})

V_{ADIN} is the input voltage signal from one of the eight ADC channels to the ADC module.

15.6.2 ADC Analog Power Pin (V_{DDA})

The ADC analog portion uses V_{DDA} as its power pin. Connect the V_{DDA} pin to the same voltage potential as V_{DD}. External filtering may be necessary to ensure clean V_{DDA} for good results.

NOTE

Route V_{DDA} carefully for maximum noise immunity and place bypass capacitors as close as possible to the package.

15.6.3 ADC Analog Ground Pin (V_{SSA})

The ADC analog portion uses V_{SSA} as its ground pin. Connect the V_{SSA} pin to the same voltage potential as V_{SS}.

15.6.4 ADC Voltage Reference High Pin (V_{REFH})

V_{REFH} is the power supply for setting the reference voltage V_{REFH}. Connect the V_{REFH} pin to the same voltage potential as V_{DDA}. There will be a finite current associated with V_{REFH} (see Chapter 22 Electrical Specifications).

NOTE

Route V_{REFH} carefully for maximum noise immunity and place bypass capacitors as close as possible to the package.

15.6.5 ADC Voltage Reference Low Pin (V_{REFL})

V_{REFL} is the lower reference supply for the ADC. Connect the V_{REFL} pin to the same voltage potential as V_{SSA}. There will be a finite current associated with V_{REFL} (see Chapter 22 Electrical Specifications).

IMASK1 — IRQ1 Interrupt Mask Bit

Writing a logic 1 to this read/write bit disables IRQ1 interrupt requests. Reset clears IMASK1.

1 = $\overline{\text{IRQ1}}$ interrupt requests disabled
0 = $\overline{\text{IRQ1}}$ interrupt requests enabled

MODE1 — IRQ1 Edge/Level Select Bit

This read/write bit controls the triggering sensitivity of the $\overline{\text{IRQ1}}$ pin. Reset clears MODE1.

1 = $\overline{\text{IRQ1}}$ interrupt requests on falling edges and low levels
0 = $\overline{\text{IRQ1}}$ interrupt requests on falling edges only

17.6.2 IRQ2 Status and Control Register

The IRQ2 status and control register (INTSCR2) controls and monitors operation of IRQ2. The INTSCR2 has the following functions:

- Enables/disables the internal pullup device on $\overline{\text{IRQ2}}$ pin
- Shows the state of the IRQ2 flag
- Clears the IRQ2 latch
- Masks IRQ2 interrupt request
- Controls triggering sensitivity of the $\overline{\text{IRQ2}}$ interrupt pin

Address: $001C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	PUC0ENB	0	0	IRQ2F	0	IMASK2	MODE2
Write:	▨		▨	▨	▨	ACK2		
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 17-5. IRQ2 Status and Control Register (INTSCR2)

PUC0ENB — $\overline{\text{IRQ2}}$ Pin Pullup Enable Bit.

Setting this bit to logic 1 disables the pullup on PTC0/$\overline{\text{IRQ2}}$ pin.
Reset clears this bit.

1 = $\overline{\text{IRQ2}}$ pin internal pullup is disabled
0 = $\overline{\text{IRQ2}}$ pin internal pullup is enabled

IRQ2F — IRQ2 Flag Bit

This read-only status bit is high when the IRQ2 interrupt is pending.

1 = $\overline{\text{IRQ2}}$ interrupt pending
0 = $\overline{\text{IRQ2}}$ interrupt not pending

ACK2 — IRQ2 Interrupt Request Acknowledge Bit

Writing a logic 1 to this write-only bit clears the IRQ2 latch. ACK2 always reads as logic 0. Reset clears ACK2.

IMASK2 — IRQ2 Interrupt Mask Bit

Writing a logic 1 to this read/write bit disables IRQ2 interrupt requests. Reset clears IMASK2.

1 = IRQ2 interrupt requests disabled
0 = IRQ2 interrupt requests enabled

MODE2 — IRQ2 Edge/Level Select Bit

This read/write bit controls the triggering sensitivity of the $\overline{\text{IRQ2}}$ pin. Reset clears MODE2.

1 = $\overline{\text{IRQ2}}$ interrupt requests on falling edges and low levels
0 = $\overline{\text{IRQ2}}$ interrupt requests on falling edges only

22.7 5 V Oscillator Characteristics

Table 22-6. Oscillator Specifications (5 V)

Characteristic[1]	Symbol	Min	Typ	Max	Unit
Internal oscillator clock frequency	f_{ICLK}	64 k	88 k[2]	104 k	Hz
External reference clock to OSC1[3]	f_{OSC}	dc		32 M	Hz
Crystal reference frequency[4]	$f_{XTALCLK}$	1 M	—	8 M	Hz
Crystal load capacitance[5]	C_L	—	—	—	
Crystal fixed capacitance[5]	C_1	—	$2 \times C_L$	—	
Crystal tuning capacitance[5]	C_2	—	$2 \times C_L$	—	
Feedback bias resistor	R_B	—	1 M	—	Ω
Series resistor[5]	R_S	—	0	—	Ω
External RC clock frequency	f_{RCCLK}			7.6 M	Hz
RC oscillator external R	R_{EXT}	See Figure 22-1			Ω
RC oscillator external C	C_{EXT}	—	10	—	pF

1. The oscillator circuit operates at V_{REG}.
2. Typical value reflect average measurements at midpoint of voltage range, 25 °C only.
3. No more than 10% duty cycle deviation from 50%. The max. frequency is limited by an EMC filter.
4. Fundamental mode crystals only.
5. Consult crystal vendor data sheet.

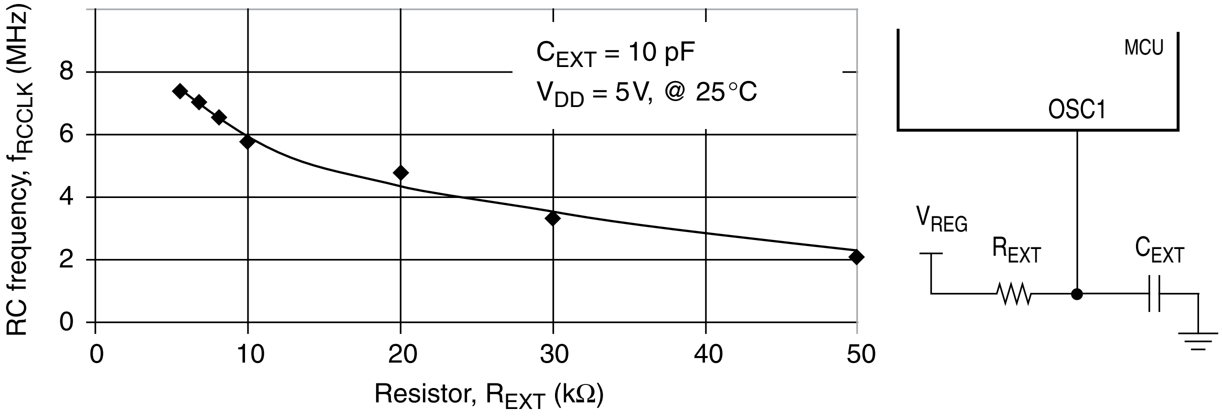


Figure 22-1. RC vs. Frequency

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.

22.10 CGM Electrical Specification

Table 22-10. CGM Electrical Specifications

Characteristic	Symbol	Min	Typ	Max	Unit
Reference frequency	f_{RDV}	1	—	8	MHz
Range nominal multiplies	f_{NOM}	—	125	—	kHz
VCO center-of-range frequency	f_{VRS}	125k	—	40M	Hz
VCO range linear range multiplier	L	1	—	255	
VCO power-of-two-range multiplier	2^E	1	—	4	
VCO multiply factor	N	1	—	4095	
VCO prescale multiplier	2^P	1	—	8	
Reference divider factor	R	1	1	15	
VCO operating frequency	f_{VCLK}	125k	—	40M	Hz
Manual acquisition time	t_{LOCK}	—	—	50	ms
Automatic lock time	t_{LOCK}	—	—	50	ms
PLL jitter[1]	f_J	0	—	$f_{RCLK} \times$ $0.025\% \times 2^P$ N/4	Hz

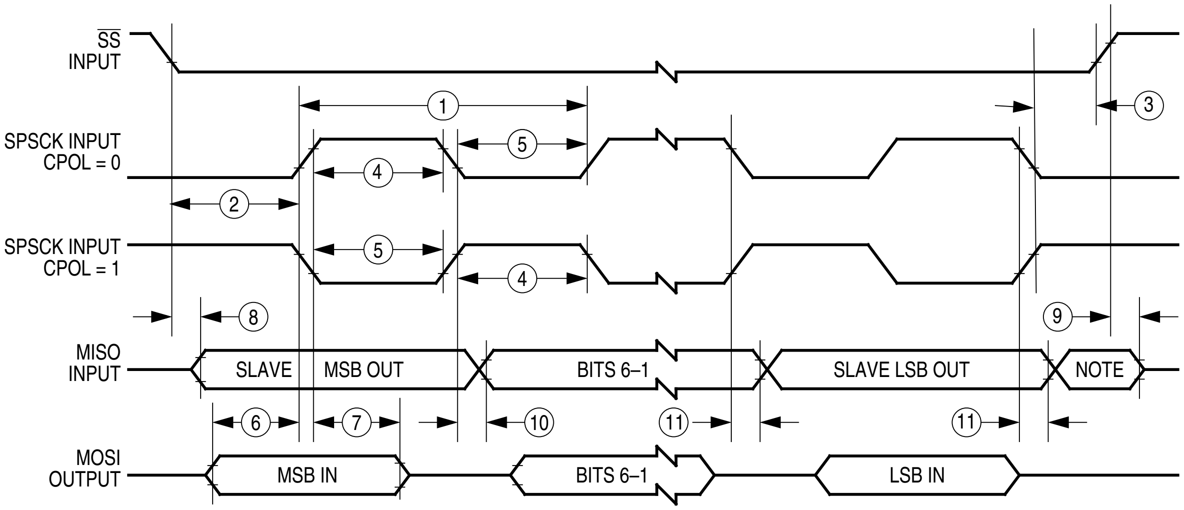
[1] Deviation of average bus frequency over 2ms. N = VCO multiplier.

22.11 5 V SPI Characteristics

Table 22-11. SPI Characteristics (5 V)

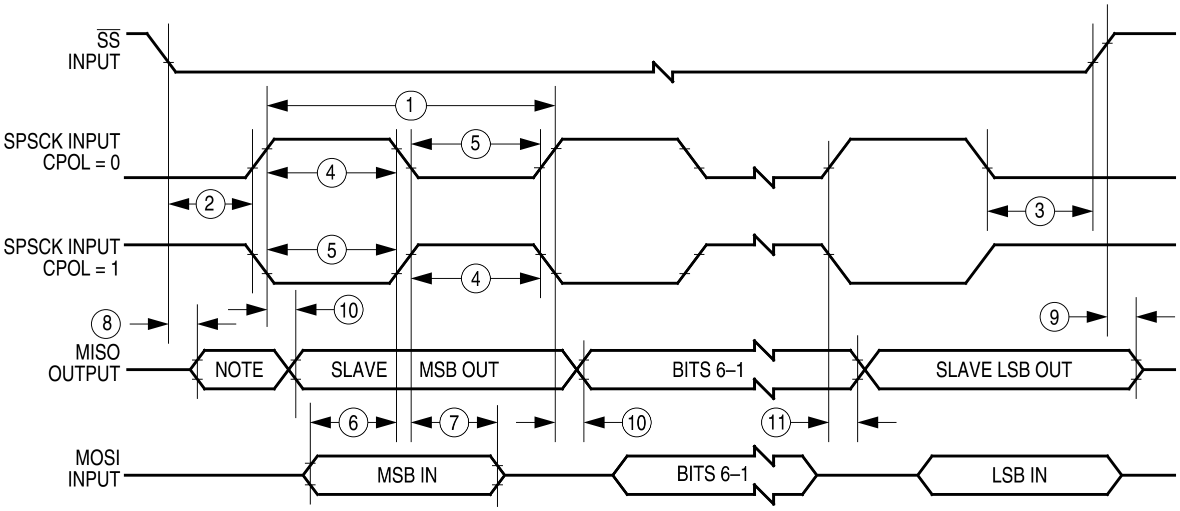
Diagram Number[1]	Characteristic[2]	Symbol	Min	Max	Unit
	Operating frequency Master Slave	$f_{OP(M)}$ $f_{OP(S)}$	$f_{OP}/128$ dc	$f_{OP}/2$ f_{OP}	MHz MHz
1	Cycle time Master Slave	$t_{CYC(M)}$ $t_{CYC(S)}$	2 1	128 —	t_{CYC} t_{CYC}
2	Enable lead time	$t_{Lead(S)}$	1	—	t_{CYC}
3	Enable lag time	$t_{Lag(S)}$	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



Note: Not defined but normally MSB of character just received

a) SPI Slave Timing (CPHA = 0)



Note: Not defined but normally LSB of character previously transmitted

b) SPI Slave Timing (CPHA = 1)

Figure 22-4. SPI Slave Timing