MC68HC05C4

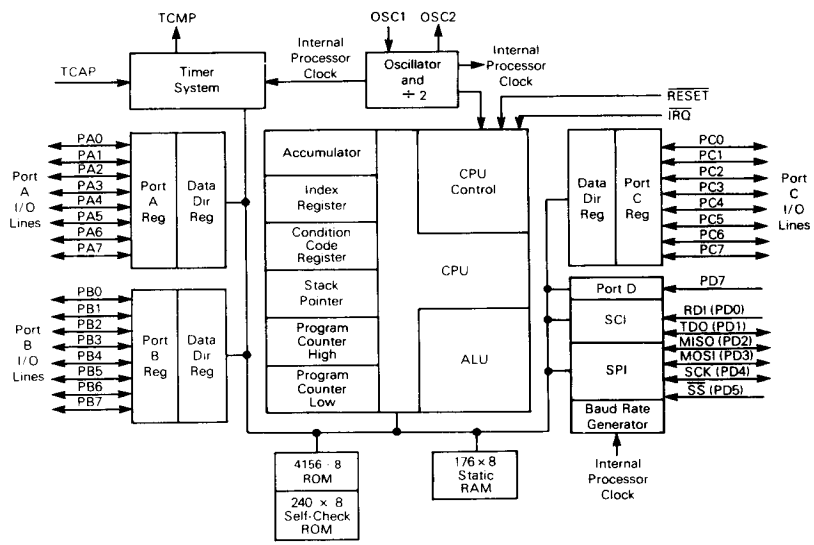
Technical Summary
8-Bit Microcontroller Unit

The MC68HC05C4 (HCMOS) microcontroller unit (MCU) is a member of the M68HC05 Family of microcontrollers. This high-performance, low-power MCU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for more detailed information, contact your local Motorola sales office.

The following block diagram depicts the hardware features; additional features available on the MCU are as follows:

- On-Chip Oscillator with RC or Crystal/Ceramic Resonator Mask Options
- Memory-Mapped I/O
- 176 Bytes of On-Chip RAM
- 4156 Bytes of User ROM
- 24 Bidirectional I/O Lines and 7 Input-Only Lines
- Serial Communications Interface (SCI) System
- Serial Peripheral Interface (SPI) System
- Self-Check Mode
- Power-Saving STOP, WAIT, and Data Retention Modes
- Single 3.0- to 5.5-Volt Supply (2-Volt Data Retention Mode)
- Fully Static Operation
- 8 × 8 Unsigned Multiply Instruction

BLOCK DIAGRAM



This document contains information on a new product. Specifications and information herein are subject to change without notice.

This Material Copyrighted By Its Respective Manufacturer

SIGNAL DESCRIPTION

The signal descriptions of the MCU are discussed in the following paragraphs.

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is the positive supply, and V_{SS} is ground.

$\overline{IRQ}$

This pin is a programmable option that provides two different choices of interrupt triggering sensitivity. Refer to **INTERRUPTS** for more detail.

OSC1, OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, a resistor capacitor combination, or an external signal connects to these pins providing a system clock. A mask option selects either a crystal ceramic resonator or a resistor capacitor as the frequency determining element. The oscillator frequency is two times the internal bus rate.

RC Oscillator

With this option, a resistor is connected to the oscillator pins as shown in Figure 1(d). The relationship between R and f_{osc} is shown in Figure 2.

Crystal

The circuit shown in Figure 1(b) is recommended when using a crystal. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Refer to **ELECTRICAL SPECIFICATIONS** for V_{DD} specifications.

3

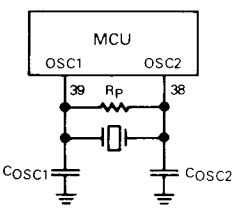
Crystal

	2 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30	40	K

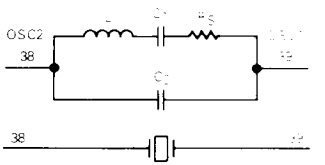
Ceramic Resonator

	2-4 MHz	Units
R_S (typical)	10	Ω
C_O	40	pF
C_1	4.3	pF
C_{OSC1}	30	pF
C_{OSC2}	30	pF
R_P	1.0	MΩ
Q	1250	

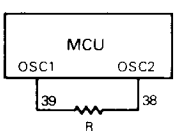
(a) Crystal Ceramic Resonator Parameters



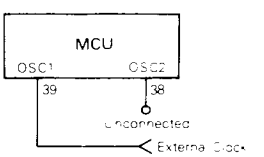
(b) Crystal/Ceramic Resonator Oscillator Connections



(c) Equivalent Crystal Circuit



(d) RC Oscillator Connections



(e) External Clock Source Connections (For Crystal Mask Option Only)

Figure 1. Oscillator Connections

This Material Copyrighted By Its Respective Manufacturer

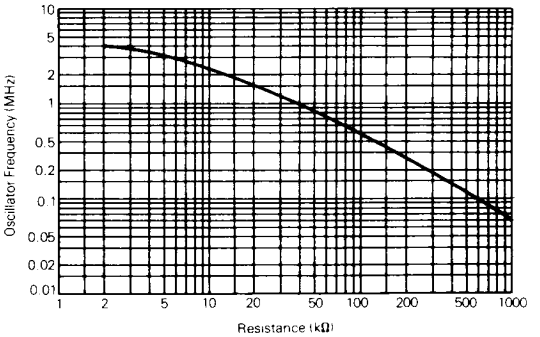


Figure 2. Typical Frequency vs Resistance for RC Oscillator Option Only

Ceramic Resonator

A ceramic resonator may be used in place of the crystal in cost-sensitive applications. The circuit in Figure 1(b) is recommended when using a ceramic resonator. Figure 1(a) lists the recommended capacitance and resistance values. The manufacturer of the resonator considered should be consulted for specific information on resonator operation.

External Clock

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 1(e). This option may only be used with the crystal oscillator mask option.

INPUT CAPTURE (TCAP)

This pin controls the input capture feature for the on-chip programmable timer.

OUTPUT COMPARE (TCMP)

This pin provides an output for the output compare feature of the on-chip timer.

$\overline{\text{RESET}}$

This pin is used to reset the MCU and provide an orderly start-up procedure by pulling $\overline{\text{RESET}}$ low.

INPUT/OUTPUT PORTS (PA0-PA7, PB0-PB7, PC0-PC7)

These 24 lines are arranged into three 8-bit ports (A, B, and C). These ports are programmable as either inputs or outputs under software control of the data direction registers. Refer to **PROGRAMMING** for additional information.

FIXED INPUT PORT (PD0-PD5, PD7)

These seven lines comprise port D, a fixed input port. All special functions that are enabled (SPI, SCI) affect this port. Refer to **PROGRAMMING** for additional information.

PROGRAMMING

Input output port programming, fixed input port programming, and serial port programming are discussed in the following paragraphs.

INPUT/OUTPUT PORT PROGRAMMING

Any port pin is programmable as either an input or an output under software control of the corresponding data direction register (DDR). Each port bit can be selected as output or input by writing the corresponding bit in the port DDR to a logic one for output and logic zero for input. On reset, all DDRs are initialized to logic zero to put the ports in the input mode. The port output registers are not initialized on reset but may be written to before setting the DDR bits to avoid undefined levels.

When programmed as outputs, the latched output data is readable as input data regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. Refer to Figure 3 for typical port circuitry and to Table 1 for a list of the I O pin functions.

Table 1. I/O Pin Functions

R/$\overline{\text{W}}$*	DDR	I/O Pin Functions
0	0	The I O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I O pin.
1	0	The state of the I O pin is read.
1	1	The I O pin is in an output mode. The output data latch is read.

*R $\overline{\text{W}}$ is an internal signal.

This Material Copyrighted By Its Respective Manufacturer

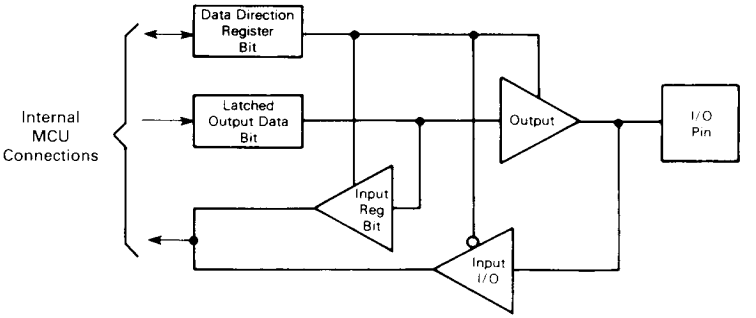


Figure 3. Typical Port I/O Circuit

3

FIXED INPUT PORT PROGRAMMING

Port D is a fixed input port (PD0-PD5, PD7) that monitors the external pins whenever the SCI or SPI is disabled. After reset, all seven bits become valid inputs because all special function drivers are disabled. For example, with the SCI enabled, PD0 and PD1 inputs will read zero. With the SPI disabled, PD2 through PD5 will read the state of the pin at the time of the read operation.

NOTE

Any unused inputs and I/O ports should be tied to an appropriate logic level (e.g., either V_{DD} or V_{SS}).

SERIAL PORT (SCI AND SPI) PROGRAMMING

The SCI and SPI use the port D pins for their functions. The SCI requires two pins (PD0-PD1) for its receive data input (RDI) and transmit data output (TD0), respectively. The SPI function requires four of the pins (PD2-PD5) for its serial data input/output (MISO), serial data output input (MOSI), serial clock (SCK), and slave select ($\overline{SS}$), respectively.

MEMORY

The MCU is capable of addressing 8192 bytes of memory and I/O registers, as shown in Figure 4. The locations consist of user ROM, user RAM, self-check ROM, control registers, and I/O. The user-defined reset and interrupt vectors are located from $1FF4 to $1FFF.

The shared stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **INTERRUPTS** for additional information.

NOTE

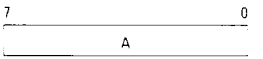
Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

REGISTERS

The MCU contains the registers described in the following paragraphs.

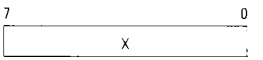
ACCUMULATOR (A)

The accumulator is a general-purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



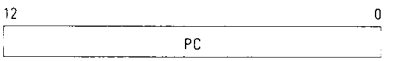
INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an 8- or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.



PROGRAM COUNTER (PC)

The program counter is a 13-bit register that contains the address of the next byte to be fetched.



STACK POINTER (SP)

The stack pointer is a 13-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the seven most significant bits are permanently set to 0000011. These seven bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0.

This Material Copyrighted By Its Respective Manufacturer

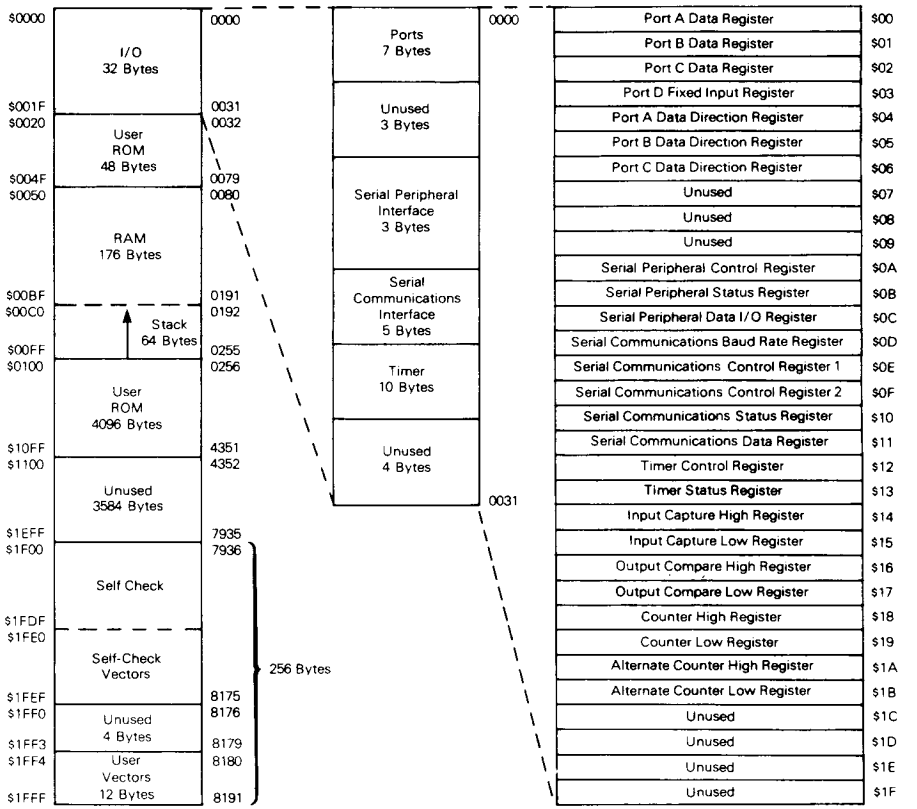
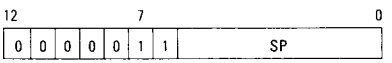


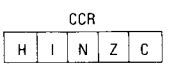
Figure 4. Memory Map

Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.



CONDITION CODE REGISTER (CCR)

The CCR is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.



Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

When this bit is set, the timer and external interrupt is masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and processed as soon as the interrupt bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic one).

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

This Material Copyrighted By Its Respective Manufacturer

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions and during shifts and rotates.

SELF-CHECK

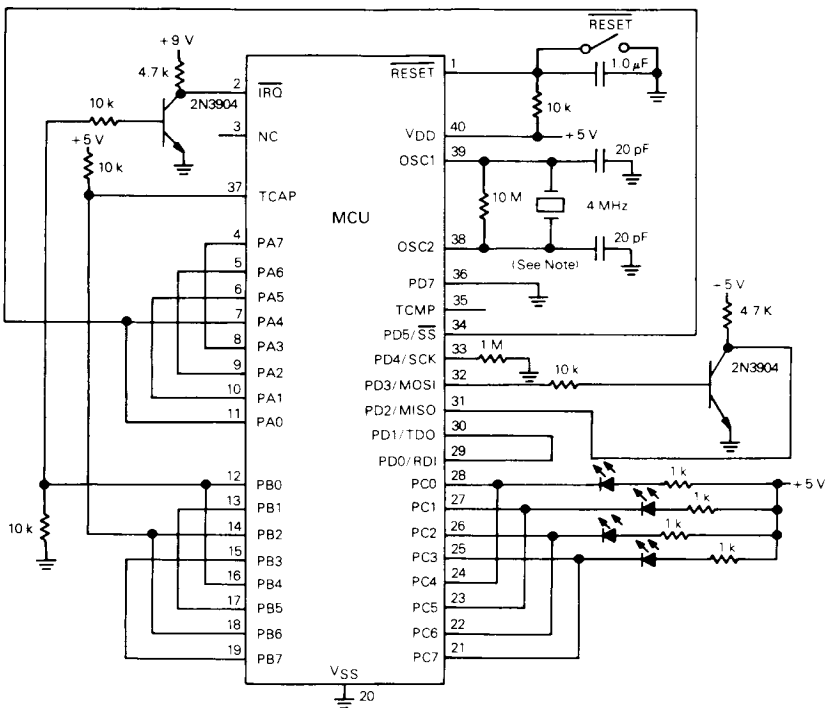
The self-check capability provides the ability to determine if the device is functional. Self-check is performed using the circuit shown in Figure 5. Port C pins PC0-PC3 are monitored for the self-check results. After reset, the following seven tests are performed automatically:

I/O — Exercise of ports A, B, and C
RAM — Counter test for each RAM byte
ROM — Exclusive OR with odd ones parity result
Timer — Tracks counter register and checks OCF flag
Interrupts — Tests external, timer, SCI and SPI interrupts
SCI — Transmission test; checks RDRF, TDRE, TC, and FE flags
SPI — Transmission test; checks SPIF, WCOL, and MODF flags

Self-check results (using the LEDs as monitors) are shown in Table 2. The following subroutines are available to the user and do not require any external hardware.

TIMER TEST SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. The timer test subroutine is called at location $1FOE. The output compare register is first set to the current timer state. Because the timer is free running and has only a divide-by-four prescaler, each timer count cannot be tested. The test reads the timer once every 10 counts (40 cycles) and checks for correct counting. The test tracks the counter until the timer wraps around, triggering the output compare flag in the timer status register. RAM locations $0050 and $0051 are overwritten. Upon return to the user's program, X = 40. If the test passed, A = 0.



NOTE: The RC Oscillator Option may also be used in this circuit

Figure 5. Self-Check Circuit Schematic Diagram

This Material Copyrighted By Its Respective Manufacturer

Table 2. Self-Check Results

PC3	PC2	PC1	PC0	Remarks
1	0	0	1	Bad I/O
1	0	1	0	Bad RAM
1	0	1	1	Bad Timer
1	1	0	0	Bad SCI
1	1	0	1	Bad ROM
1	1	1	0	Bad SPI
1	1	1	1	Bad Interrupts or $\overline{IRQ}$ Request
Flashing				Good Device
All Others				Bad Device, Bad Port C, etc.

0 indicates LED is on; 1 indicates LED is off.

ROM CHECKSUM SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set. The ROM checksum subroutine is called at location $1F93 with RAM location $0053 equal to $01 and A = 0. A short routine is set up and executed in RAM to compute a checksum of the entire ROM pattern. RAM locations $0050 through $0053 are overwritten. Upon return to the user's program, X = 0. If the test passed, A = 0.

RESETS

The MCU can be reset two ways: by initial power-up and by the external reset input ($\overline{RESET}$). The $\overline{RESET}$ input consists mainly of a Schmitt trigger that senses the $\overline{RESET}$ line logic level.

POWER-ON RESET (POR)

An internal reset is generated on power-up to allow the internal clock generator to stabilize. The power-on reset is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 4064 internal processor clock cycle (t_{cyc}) delay after the oscillator becomes active. If the $\overline{RESET}$ pin is low at the end of 4064 t_{cyc}, the MCU will remain in the reset condition until $\overline{RESET}$ goes high.

EXTERNAL RESET INPUT

The MCU is reset when a logic zero is applied to the $\overline{RESET}$ input for a period of one and one-half machine cycles (t_{cyc}).

INTERRUPTS

The MCU can be interrupted five different ways: the four maskable hardware interrupts ($\overline{IRQ}$, SPI, SCI, and timer) and the nonmaskable software interrupt instruction (SWI).

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume. The stacking order is shown in Figure 6.

Unlike $\overline{RESET}$, hardware interrupts do not cause the current instruction execution to be halted but are considered pending until the current instruction is complete.

NOTE

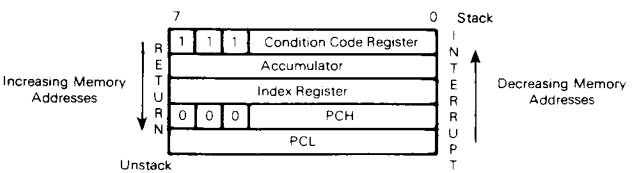
The current instruction is the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts. If unmasked (I bit clear) and if the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction, regardless of the I-bit state. Refer to Figure 7 for the reset and interrupt instruction processing sequence.

TIMER INTERRUPT

There are three different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags are in the timer status register (TSR), and the enable bits are in the timer control register (TCR). Refer to **TIMER** for more information.



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Figure 6. Interrupt Stacking Order

This Material Copyrighted By Its Respective Manufacturer

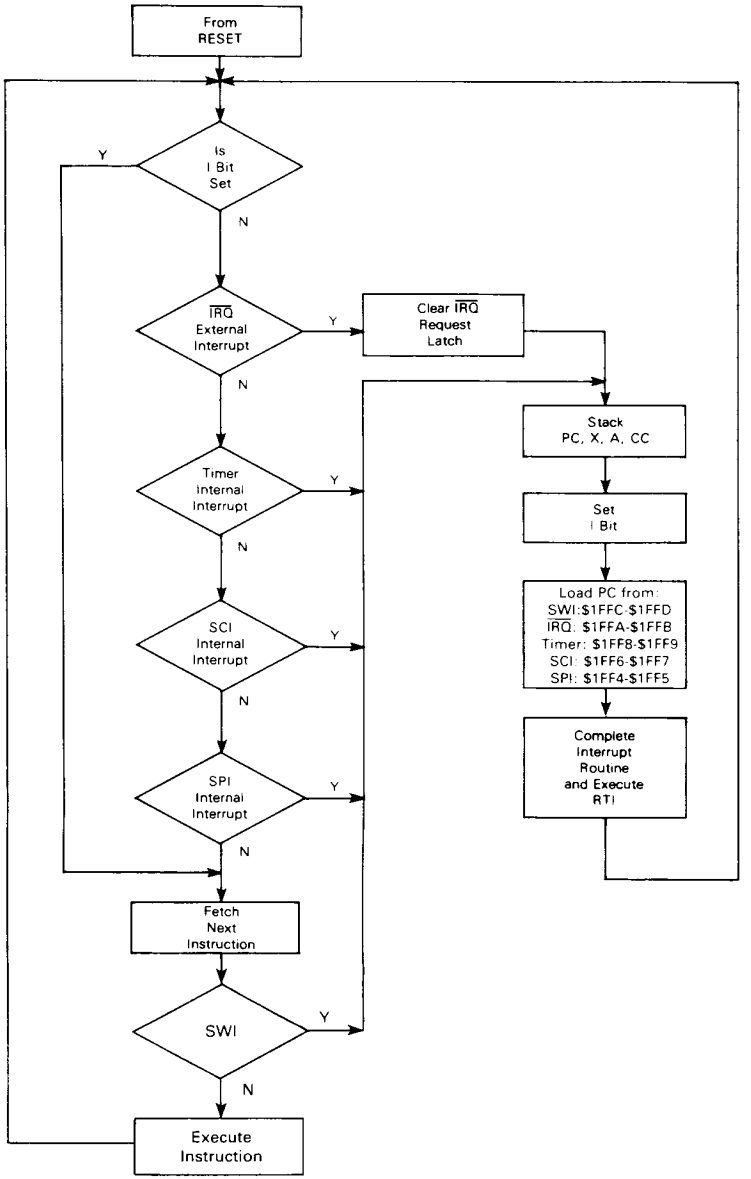


Figure 7. Reset and Interrupt Processing Flowchart

This Material Copyrighted By Its Respective Manufacturer

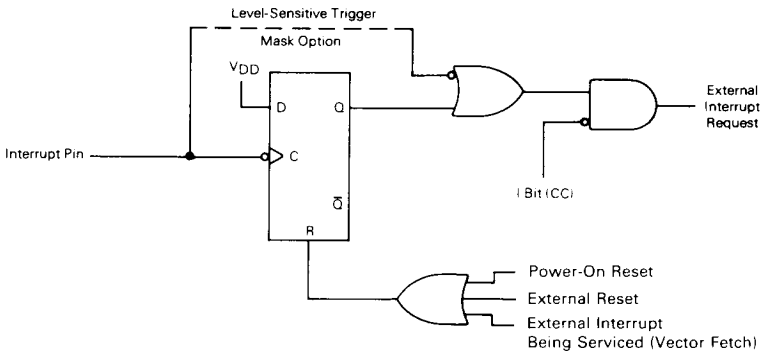
EXTERNAL INTERRUPT

If the interrupt mask bit (I bit) of the CCR is set, all interrupts are disabled. Clearing the I bit enables the external interrupt. The external interrupt is internally synchronized and then latched on the falling edge of $\overline{IRQ}$. The action of the external interrupt is identical to the timer interrupt with the exception that the interrupt request input at $\overline{IRQ}$ is latched internally and the service routine address is specified by the contents of $1FFA and $1FFB.

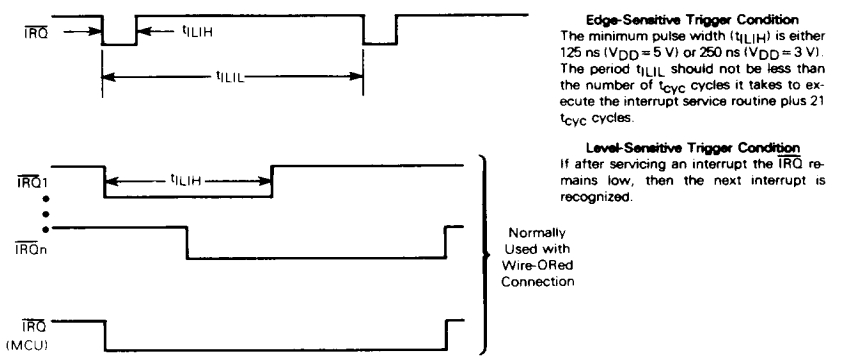
Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive-only trigger are available as a mask option. Figure 8 shows both a functional internal diagram and a mode timing diagram for the interrupt line. The timing diagram shows two treatments of the interrupt line to the processor. The first method shows a single pulse on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service. Once a pulse occurs, the next pulse should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 21 instruction cycles to the total number of cycles it takes to complete the service routine (not including the RTI instruction). The second method shows many interrupt lines "wire-ORed" to form the interrupts at the processor. If the interrupt line remains low after servicing an interrupt, then the next interrupt is recognized.

NOTE

The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse could be latched and serviced as soon as the I bit is cleared.



(a) Interrupt Internal Function Diagram



(b) Interrupt Mode Diagram

Figure 8. External Interrupt

This Material Copyrighted By Its Respective Manufacturer

SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. If the I bit is zero, SWI executes after the other interrupts. The SWI operation is similar to the hardware interrupts. The interrupt service routine address is specified by the contents of memory locations $1FFC and $1FFD.

SCI INTERRUPTS

An interrupt in the SCI occurs when one of the interrupt flag bits in the serial communications status register is set, provided the I bit in the CCR is clear and the enable bit in the serial communications control register 2 is set. Software in the serial interrupt service routine must determine the cause and priority of the SCI interrupt by examining the interrupt flags and status bits in the SCI status register.

SPI INTERRUPTS

An interrupt in the SPI occurs when one of the interrupt flag bits in the serial peripheral status register is set, provided the I bit in the CCR is clear and the enable bit in the serial peripheral control register is set. Software in the serial peripheral interrupt service routine must determine the cause and priority of the SPI interrupt by examining the interrupt flag bits in the SPI status register.

3

LOW-POWER MODES

STOP

The STOP instruction places the MCU in its lowest power consumption mode. In the STOP mode, the internal oscillator is turned off, halting all internal processing including timer, SCI, and SPI operation (refer to Figure 9).

During the STOP mode, the TCR bits are altered to remove any pending timer interrupt request and to disable any further timer interrupts. The timer prescaler is cleared. The I bit in the CCR is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt or reset.

SCI during STOP Mode

When the MCU enters the STOP mode, the baud rate generator stops, halting all SCI activity. If the STOP instruction is executed during a transmitter transfer, that transfer is halted. If a low input to the $\overline{IRQ}$ pin is used to exit STOP mode, the transfer resumes. If the SCI receiver is receiving data and the STOP mode is entered, received data sampling stops because the baud rate generator stops, and all subsequent data is lost. For these reasons, all SCI transfers should be in the idle state when the STOP instruction is executed.

SPI during Stop Mode

When the MCU enters the STOP mode, the baud rate generator stops, terminating all master mode SPI operations. If the STOP instruction is executed during an SPI transfer, that transfer halts until the MCU exits the STOP mode by a low signal on the $\overline{IRQ}$ pin. If reset is used to exit the STOP mode, then the SPI control and status bits are cleared, and the SPI is disabled. If the MCU is in the slave mode when the STOP instruction is executed, the slave SPI continues to operate and can still accept data and clock information in addition to transmitting its own data back to a master device.

At the end of a possible transmission with a slave SPI in the STOP mode, no flags are set until a low on the $\overline{IRQ}$ pin wakes up the MCU. Caution should be observed when operating the SPI as a slave during the STOP mode because the protective circuitry (WCOL, MODF, etc.) is inactive.

WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes more

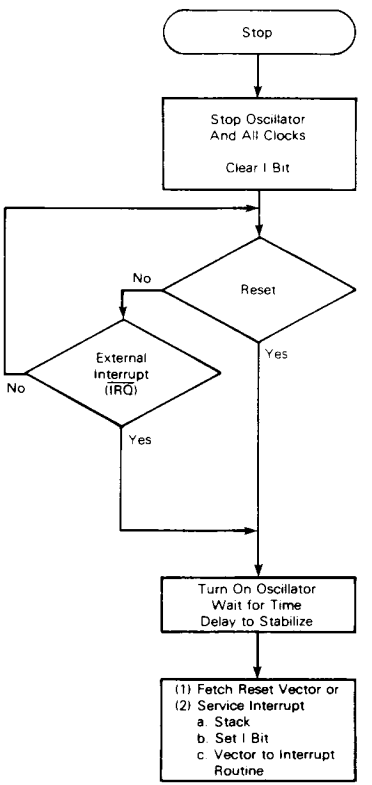


Figure 9. STOP Function Flowchart

This Material Copyrighted By Its Respective Manufacturer

power than the STOP mode. All CPU action is suspended, but the timer, SCI, and SPI remain active (refer to Figure 10). An interrupt from the timer, SCI, or SPI can cause the MCU to exit the WAIT mode.

During the WAIT mode, the I bit in the CCR is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode.

DATA RETENTION MODE

The contents of RAM and CPU registers are retained at supply voltages as low as 2.0 Vdc. This is called the data retention mode where the data is held, but the device is not guaranteed to operate. The MCU should be in $\overline{\text{RESET}}$ during data retention mode.

TIMER

The timer consists of a 16-bit, software-programmable counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. Refer to Figure 11 for a timer block diagram.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

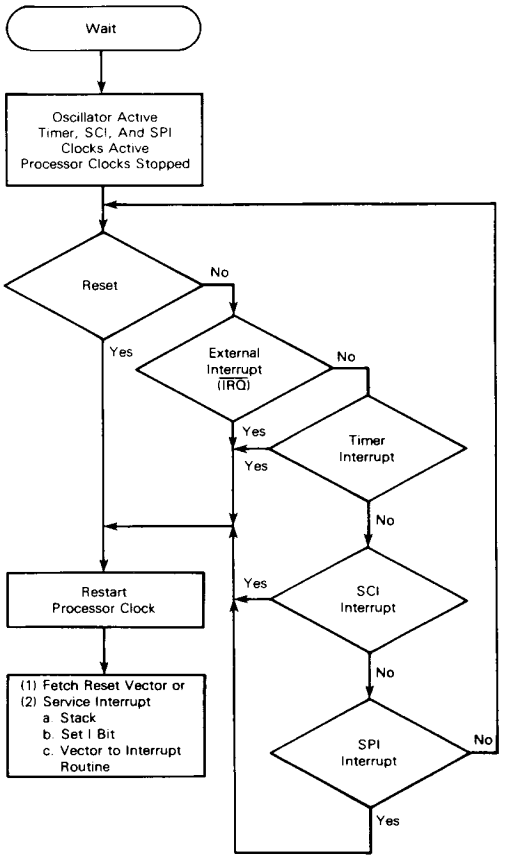


Figure 10. WAIT Function Flowchart

This Material Copyrighted By Its Respective Manufacturer

NOTE

The I bit in the CCR should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur.

COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations, $18–$19 (counter register) or $1A–$1B (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter ($19, $1B) receives the count value at the time of the read. If a read of the free-running counter or counter alternate register first addresses the most significant byte (MSB) ($18, $1A), the LSB ($19, $1B) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($19 or $1B) and, thus, completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB must also be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: a read of the counter register MSB can clear the timer overflow flag (TOF). Therefore, the counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.

The free-running counter is configured to $FFFC during reset and is always a read-only register. During a power-on reset, the counter is also preset to $FFFC and begins

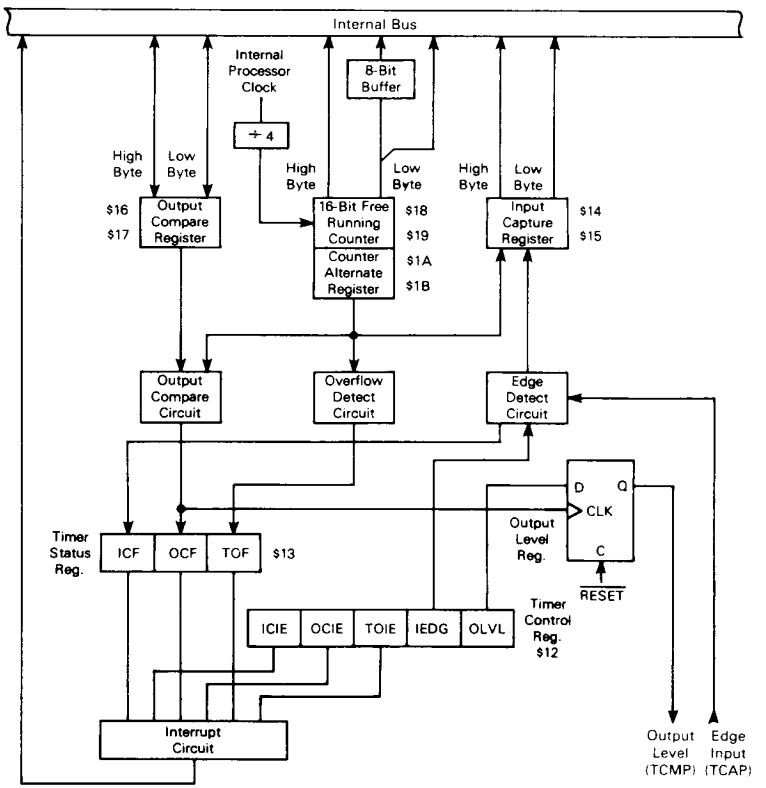


Figure 11. Timer Block Diagram

This Material Copyrighted By Its Respective Manufacturer

running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from $FFFF to $0000, the TOF bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

OUTPUT COMPARE REGISTER

The 16-bit output compare register is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

The output compare register contents are compared with the contents of the free-running counter continually, and if a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLCL) bit is clocked to an output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE) is set.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear.

INPUT CAPTURE REGISTER

Two 8-bit registers, which make up the 16-bit input capture register, are read-only and are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the input capture register.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture.

After a read of the input capture register ($14) MSB, the counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($15) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

TIMER CONTROL REGISTER (TCR) $12

The TCR is a read/write register containing five control bits. Three bits control interrupts associated with the timer status register flags ICF, OCF, and TOF.

7	6	5	4	3	2	1	0
ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
RESET:							
0	0	0	0	0	0	U	0

ICIE — Input Capture Interrupt Enable
- 1 = Interrupt enabled
- 0 = Interrupt disabled

OCIE — Output Compare Interrupt Enable
- 1 = Interrupt enabled
- 0 = Interrupt disabled

TOIE — Timer Overflow Interrupt Enable
- 1 = Interrupt enabled
- 0 = Interrupt disabled

IEDG — Input Edge

Value of input edge determines which level transition on TCAP pin will trigger free-running counter transfer to the input capture register
- 1 = Positive edge
- 0 = Negative edge

Reset does not affect te IEDG bit (U = unaffected).

OLVL — Output Level

Value of output level is clocked into output level register by the next successful output compare and will appear on the TCMP pin
- 1 = High output
- 0 = Low output

Bits 2, 3, and 4 — Not used

Always read zero

TIMER STATUS REGISTER (TSR) $13

The TSR is a read-only register containing three status flag bits.

7	6	5	4	3	2	1	0
ICF	OCF	TOF	0	0	0	0	0
RESET:							
U	U	U	0	0	0	0	0

ICF — Input Capture Flag
- 1 = Flag set when selected polarity edge is sensed by input capture edge detector
- 0 = Flag cleared when TSR and input capture low register ($15) are accessed

This Material Copyrighted By Its Respective Manufacturer

OCF — Output Compare Flag
- 1 = Flag set when output compare register contents match the free-running counter contents
- 0 = Flag cleared when TSR and output compare low register ($17) are accessed

TOF — Timer Overflow Flag
- 1 = Flag set when free-running counter transition from $FFFF to $0000 occurs
- 0 = Flag cleared when TSR and counter low register ($19) are accessed

Bits 0–4 — Not used
Always read zero

Accessing the timer status register satisfies the first condition required to clear status bits. The remaining step is to access the register corresponding to the status bit.

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if:

1) The timer status register is read or written when TOF is set, and
2) The LSB of the free-running counter is read but not for the purpose of servicing the flag.

The counter alternate register at address $1A and $1B contains the same value as the free-running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

TIMER DURING WAIT MODE

The CPU clock halts during the WAIT mode, but the timer remains active. An interrupt from the timer causes the processor to exit the WAIT mode.

TIMER DURING STOP MODE

In the STOP mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If RESET is used, the counter is forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags nor wake up the MCU, but when the MCU does wake up, there is an active input capture flag and data from the first valid edge that occurred during the STOP mode. If RESET is used to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

SERIAL COMMUNICATIONS INTERFACE

A full-duplex asynchronous SCI is provided with a standard NRZ format and a variety of baud rates. The SCI transmitter and receiver are functionally independent but use the same data format and baud rate. The terms baud and bit rate are used synonymously in the following description.

SCI TWO-WIRE SYSTEM FEATURES

- Standard NRZ (mark/space) format
- Advanced error detection method includes noise detection for noise duration of up to one-sixteenth bit time
- Full-duplex operation (simultaneous transmit and receive)
- Software programmable for one of 32 different baud rates
- Software-selectable word length (eight- or nine-bit words)
- Separate transmitter and receiver enable bits
- SCI may be interrupt driven
- Four separate interrupt conditions

SCI RECEIVER FEATURES

- Receiver wake-up function (idle or address bit)
- Idle line detect
- Framing error detect
- Noise detect
- Overrun detect
- Receiver data register full flag

SCI TRANSMITTER FEATURES

- Transmit data register empty flag
- Transmit complete flag
- Break send

Any SCI two-wire system requires receive data in (RDI) and transmit data out (TDO).

DATA FORMAT

Receive data in (RDI) or transmit data out (TDO) is the serial data presented between the internal data bus and the output pin (TDO) and between the input pin (RDI) and the internal data bus. Data format is as shown for the NRZ in Figure 12.

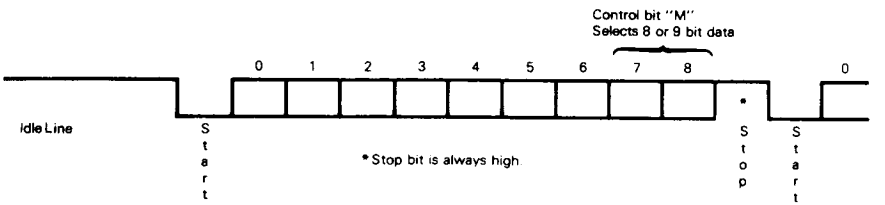


Figure 12. Data Format

This Material Copyrighted By Its Respective Manufacturer

WAKE-UP FEATURE

In a typical multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. To permit uninterested MPUs to ignore the remainder of the message, a wake-up feature is included, whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line returns to the idle state. An SCI receiver is re-enabled by an idle string of at least ten (or eleven) consecutive ones. Software for the transmitter must provide for the required idle string between consecutive messages and prevent it from occurring within messages.

A second wake-up method is available in which sleeping SCI receivers can be awakened by a logic one in the high-order bit of a received character.

RECEIVE DATA IN

Receive data in (RDI) is the serial data which is presented from the input pin via the SCI to the receive data register (RDR). While waiting for a start bit, the receiver samples the input at a rate 16 times higher than the set baud rate. This increased rate is referred to as the RT rate. When the input (idle) line is detected low, it is tested for three more sample times. If at least two of these three samples detect a logic low, a valid start bit is assumed to be detected. If in two or more samples, a logic high is detected, the line is assumed to be idle. The receive clock generator is controlled by the baud rate register (see Figure 13); however, the SCI is synchronized by the start bit independent of the transmitter. Once a valid start bit is detected, the start bit, each data bit, and the stop bit are each sampled three times. The value of the bit is determined by voting logic, which takes the value of a majority of samples. A noise flag is set when all three samples on a valid start bit, data bit, or stop bit do not agree. A noise flag is also set when the start verification samples do not agree.

START BIT DETECTION FOLLOWING A FRAMING ERROR

If there has been a framing error (FE) without detection of a break (10 zeros for 8-bit format or 11 zeros for a 9-bit format), the circuit continues to operate as if there actually were a stop bit, and the start edge will be placed artificially. The last bit received in the data shift register is inverted to a logic one, and the three logic-one start qualifiers are forced into the sample shift register during the interval when detection of a start bit is anticipated; therefore, the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break (RDRF = 1, FE = 1, receiver data register = $00) produced the framing error, the start bit will not be artificially induced, and the receiver must actually receive a logic one before start.

TRANSMIT DATA OUT

Transmit data out (TDO) is the serial data presented from the transmit data register (TDR) via the SCI to the output pin. The transmitter generates a bit time by using a derivative of the RT clock, producing a transmission rate equal to one-sixteenth that of the receiver sample clock.

FUNCTIONAL DESCRIPTION

A block diagram of the SCI is shown in Figure 13. The user has option bits in the serial communications control register 1 (SCCR1) to determine the SCI wake-up method and data word length. Serial communications control register 2 (SCCR2) provides control bits that individually enable/disable the transmitter or receiver, enable system interrupts, and provide wake-up enable, and send break code bits. The baud rate register bits allow the user to select different baud rates, which are used as the rate control for the transmitter and receiver.

Data transmission is initiated by a write to the serial communications data register (SCDAT). Provided the transmitter is enabled, data stored in the SCDAT is transferred to the transmit data shift register. This data transfer sets the SCI status register (SCSR) transmit data register empty (TDRE) bit and generates an interrupt if the transmit interrupt is enabled. Data transfer to the transmit data shift register is synchronized with the bit rate clock. All data is transmitted LSB first. Upon completion of data transmission, the transmission complete (TC) bit is set (provided no pending data, preamble, or break code is sent), and an interrupt is generated if the transmit complete interrupt is enabled. If the transmitter is disabled, and the data, preamble, or break code has been sent, the TC bit will also be set, which will also generate an interrupt if the TCIE bit is set. If the transmitter is disabled in the middle of a transmission, that character will be completed before the transmitter gives up control of the TDO pin.

When the SCDAT is read, it contains the last data byte received, provided that the receiver is enabled. The SCSR receive data register full (RDRF) bit is set to indicate that a data byte is transferred from the input serial shift register to the SCDAT, which can cause an interrupt if the receiver interrupt is enabled. Data transfer from the input serial shift register to the SCDAT is synchronized by the receiver bit rate clock. The SCSR overrun (OR), noise flag (NF), or FE bits are set if data reception errors occur.

An idle line interrupt is generated if the idle line interrupt is enabled and the SCSR IDLE bit (which detects idle line transmission) is set. This allows a receiver that is not in the wake-up mode to detect the end of a message, the preamble of a new message, or to resynchronize with the transmitter. A valid character must be received before the idle line condition for the IDLE bit to be set and for an idle line interrupt to be generated.

REGISTERS

There are five registers used in the SCI; the internal configuration of these registers is discussed in the following paragraphs.

Serial Communications Data Register (SCDAT) $11

The SCDAT is a read write register used to receive and transmit SCI data.

7	6	5	4	3	2	1	0
SCD7	SCD6	SCD5	SCD4	SCD3	SCD2	SCD1	SCD0
RESET: U	U	U	U	U	U	U	U

This Material Copyrighted By Its Respective Manufacturer

As shown in Figure 13, SCDAT functions as two separate registers. The transmit data register (TDR) provides the parallel interface from the internal data bus to the transmit shift register. The receive data register (RDR) provides the interface from the receive shift register to the internal data bus.

Serial Communications Control Register 1 (SCCR1) $0E

The SCCR1 provides control bits that determine word length and select the wake-up method.

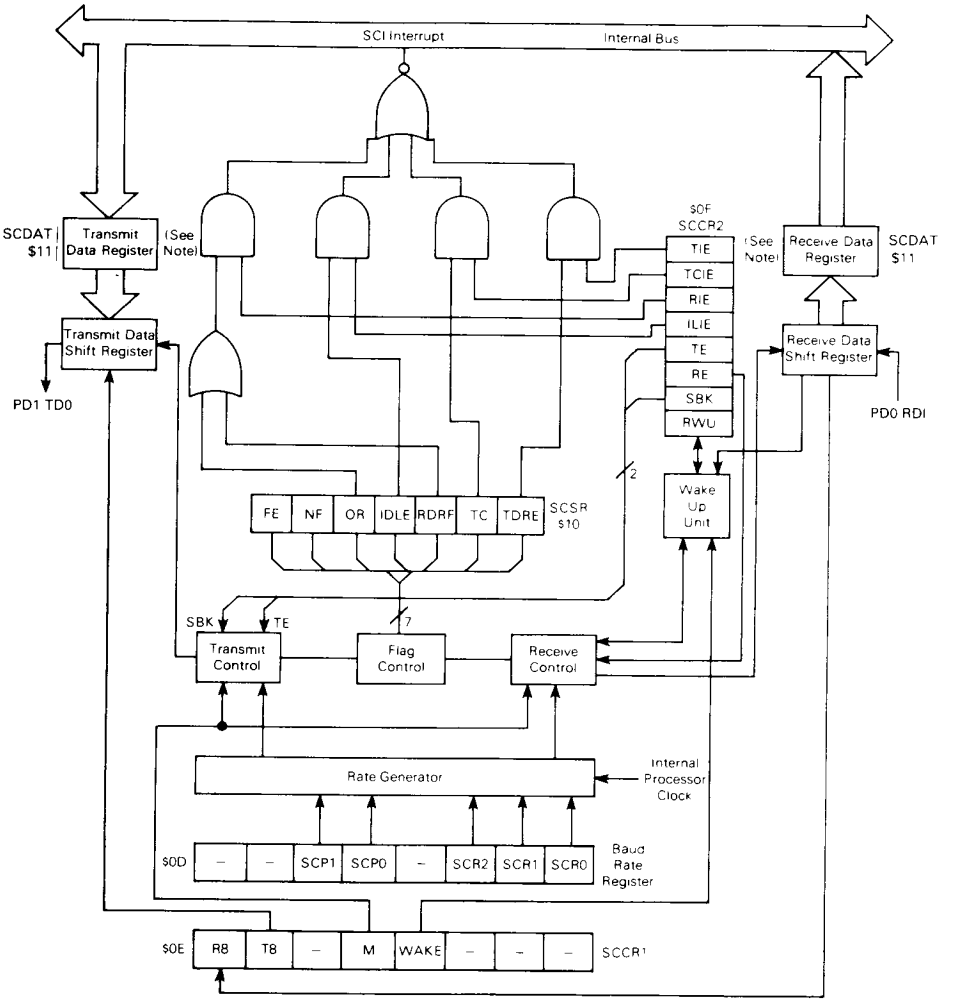
7	6	5	4	3	2	1	0
R8	T8	—	M	WAKE	—	—	—
RESET:							
U	U	—	U	U	—	—	—

R8 — Receive Data Bit 8

R8 bit provides storage location for the ninth bit in the receive data byte (if M = 1).

T8 — Transmit Data Bit 8

T8 bit provides storage location for the ninth bit in the transmit data byte (if M = 1).



NOTE: The Serial Communications Data Register (SCDAT) is controlled by the internal R/W signal. It is the transmit data register when written and receive data register when read.

Figure 13. SCI Block Diagram

This Material Copyrighted By Its Respective Manufacturer

M — SCI Character Word Length
1 = one start bit, nine data bits, one stop bit
0 = one start bit, eight data bits, one stop bit
WAKE — Wake-Up Select
Wake bit selects the receiver wake-up method.
1 = Address bit (most significant bit)
0 = Idle line condition
Bits 0–2, and 5 — Not used
Can read either one or zero

The address bit is dependent on both the wake-bit and the M-bit level. Additionally, the receiver does not use the wake-up feature unless the RWU control bit in SCCR2 is set.

Wake	M	Receiver Wake-Up
0	X	Detection of an idle line allows the next data byte received to cause the receive data register to fill and produce an RDRF flag.
1	0	Detection of a received one in the eighth data bit allows an RDRF flag and associated error flags.
1	1	Detection of a received one in the ninth data bit allows an RDRF flag and associated error flags.

Serial Communications Control Register 2 (SCCR2) $OF

The SCCR2 provides control of individual SCI functions such as interrupts, transmit/receive enabling, receiver wake-up, and break code.

7	6	5	4	3	2	1	0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET: 0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable
1 = SCI interrupt enabled
0 = TDRE interrupt disabled
TCIE — Transmit Complete Interrupt Enable
1 = SCI interrupt enabled
0 = TC interrupt disabled
RIE — Receive Interrupt Enable
1 = SCI interrupt enabled
0 = RDRF and OR interrupts disabled
ILIE — Idle Line Interrut Enable
1 = SCI interrupt enabled
0 = Idle interrupt disabled
TE — Transmit Enable
1 = Transmit shift register output is applied to the TD0 line. Depending upon the SCCR1 M bit, a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones is transmitted.
0 = Transmitter disabled after last byte is loaded in the SCDAT and TDRE is set. After last byte is transmitted, TD0 line becomes a high-impedance line.
RE — Receive Enable
1 = Receiver shift register input is applied to the RDI line.
0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE status bits are inhibited.
RWU — Receiver Wake-Up
1 = Places receiver in sleep mode and enables wake-up function
0 = Wake-up function disabled after receiving data word with MSB set (if WAKE = 1)
Wake-up function also disabled after receiving 10 (M = 0) or 11 (M = 1) consecutive ones (if WAKE = 0)
SBK — Send Break
1 = Transmitter continually sends blocks of zeros (sets of 10 or 11) until cleared. Upon completion of break code, transmitter sends one high bit for recognition of valid start bit.
0 = Transmitter sends 10 (M = 0) or 11 (M = 1) zeros then reverts to an idle state or continues sending data. If transmitter is empty and idle, setting and clearing the SBK bit may queue up to two character times of break because the first break transfers immediately to the shift register, and the second is queued into the parallel transmit buffer.

Serial Communications Status Register (SCSR) $10

The SCSR provides inputs to the SCI interrupt logic circuits. Noise flag and framing error bits are also contained in the SCSR.

7	6	5	4	3	2	1	0
TDRE	TC	RDRF	IDLE	OR	NF	FE	—
RESET: 1	1	0	0	0	0	0	—

TDRE — Transmit Data Register (TDR) Empty
1 = TDR contents transferred to the transmit data shift register
0 = TDR still contains data. TDRE is cleared by reading the SCSR (with TDRE = 1), followed by a write to the TDR.
TC — Transmit Complete
1 = Indicates end of data frame, preamble, or break condition has occurred
0 = TC bit cleared by reading the SCSR (with TC = 1), followed by a write to the TDR
RDRF — Receive Data Register (RDR) Full
1 = Receive data shift register contents transferred to the RDR
0 = Receive data shift register transfer did not occur. RDRF is cleared by reading the SCSR (with RDRF = 1) followed by a read of the RDR
IDLE — Idle Line Detect
1 = Indicates receiver has detected an idle line
0 = IDLE is cleared by reading the SCSR (with IDLE = 1), followed by a read of the RDR. Once IDLE is cleared, IDLE cannot be set until RDI line becomes active and idle again.
OR — Overrun Error
1 = Indicates receive data shift register data is sent to a full RDR (RDRF = 1). Data causing the overrun is lost, and RDR data is not disturbed.
0 = OR is cleared by reading the SCSR (with OR = 1), followed by a read of the RDR.

3

This Material Copyrighted By Its Respective Manufacturer

NF — Noise Flag
1 = Indicates noise is present on the receive bits, including the start and stop bits. NF is not set until RDRF = 1.
0 = NF is cleared by reading the SCSR (with NF = 1), followed by a read of the RDR.
FE — Framing Error
1 = Indicates stop bit not detected in received data character. FE is set the same time RDRF is set. If received byte causes both framing and overrun errors, processor will only recognize the overrun error. Further data transfer into the RDR is inhibited until FE is cleared.
0 = NF is cleared by reading the SCSR (with FE = 1), followed by a read of the RDR.
Bit 0 — Not used
Can read either one or zero

Baud Rate Register $0D

The baud rate register is used to select the SCI transmitter and receiver baud rate. SCP0 and SCP1 prescaler bits are used in conjunction with the SCR0 through SCR2 baud rate bits to provide multiple baud rate combinations for a given crystal frequency. Bits 3, 6, and 7 always read zero.

7	6	5	4	3	2	1	0
—	—	SCP1	SCP0	—	SCR2	SCR1	SCR0
RESET:							
—	—	0	0	—	U	U	U

SCP0 — SCI Prescaler Bit 0
SCP1 — SCI Prescaler Bit 1
Two prescaler bits are used to increase the range of standard baud rates controlled by the SCR0–SCR2 bits. Prescaler internal processor clock division versus bit levels are listed in Table 2.

SCR0 — SCI Baud Rate Bit 0
SCR1 — SCI Baud Rate Bit 1
SCR2 — SCI Baud Rate Bit 2
Three baud rate bits are used to select the baud rates of the SCI transmitter and SCI receiver. Baud rates versus bit levels are listed in Table 3.

Tables 3 and 4 tabulate the divide chain used to obtain the baud rate clock (transmit clock). The actual divider chain is controlled by the combined SCP0-SCP1 and SCR0-SCR2 bits in the baud rate register. All divided frequencies shown in Table 3 represent the final baud rate resulting from the internal processor clock division shown in the divided-by column only (prescaler division only). Table 4 lists the prescaler output divided by the action of the SCI select bits (SCR0-SCR2). For example, assume that a 9600-Hz baud rate is required with a 2.4576-MHz external crystal. In this case, the prescaler bits (SCP0-SCP1) could be configured as a divide-by-one or a divide-by-four. If a divide-by-four prescaler is used, then the SCR0-SCR2 bits must be configured as a divide-by-two. Using the same crystal, the 9600 baud rate can be obtained with a prescaler divide-by-one and the SCR0-SCR2 bits configured for a divide-by-eight.

Table 3. Prescaler Highest Baud Rate Frequency Output

SCP Bit		Clock* Divided By	Crystal Frequency MHz				
1	0		4.194304	4.0	2.4576	2.0	1.8432
0	0	1	131.072 kHz	125.000 kHz	76.80 kHz	62.50 kHz	57.60 kHz
0	1	3	43.691 kHz	41.666 kHz	25.60 kHz	20.833 kHz	19.20 kHz
1	0	4	32.768 kHz	31.250 kHz	19.20 kHz	15.625 kHz	14.40 kHz
1	1	13	10.082 kHz	9600 Hz	5.907 kHz	4800 Hz	4430 Hz

*Refers to the internal processor clock.

NOTE: The divided frequencies shown in Table 3 represent baud rates which are the highest transmit baud rate (Tx) that can be obtained by a specific crystal frequency and only using the prescaler division. Lower baud rates may be obtained by providing a further division using the SCI rate select bits as shown below for some representative prescaler outputs.

Table 4. Transmit Baud Rate Output for a Given Prescaler Output

SCR Bits			Divided By	Representative Highest Prescaler Baud Rate Output				
2	1	0		131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	0	1	131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	1	2	65.536 kHz	16.384 kHz	38.40 kHz	9600 Hz	4800 Hz
0	1	0	4	32.768 kHz	8.192 kHz	19.20 kHz	4800 Hz	2400 Hz
0	1	1	8	16.384 kHz	4.096 kHz	9600 Hz	2400 Hz	1200 Hz
1	0	0	16	8.192 kHz	2.048 kHz	4800 Hz	1200 Hz	600 Hz
1	0	1	32	4.096 kHz	1.024 kHz	2400 Hz	600 Hz	300 Hz
1	1	0	64	2.048 kHz	512 Hz	1200 Hz	300 Hz	150 Hz
1	1	1	128	1.024 kHz	256 Hz	600 Hz	150 Hz	75 Hz

NOTE: Table 4 illustrates how the SCI select bits can be used to provide lower transmitter baud rates by further dividing the prescaler output frequency. The five examples are only representative samples. In all cases, the baud rates shown are transmit baud rates (transmit clock), and the receive clock is 16 times higher in frequency than the actual baud rate.

This Material Copyrighted By Its Respective Manufacturer

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is an interface built into the MCU which allows several MCUs or MCUs plus peripherals to be interconnected within the same black box. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. An SPI system may consist of one master MCU and several slaves (Figure 14) or MCUs that can be either masters or slaves.

Features:

- Full-duplex, three-wire synchronous transfers
- Master or slave operation
- 1.05 MHz (maximum) master bit frequency
- 2.1 MHz (maximum) slave bit frequency
- Four programmable master bit rates
- Programmable clock polarity and phase
- End-of-transmission interrupt flag
- Write collision flag protection
- Master-master mode fault protection capability

SIGNAL DESCRIPTION

The four basic signals (MOSI, MISO, SCK, and $\overline{SS}$) are described in the following paragraphs. Each signal function is described for both master and slave mode.

Master Out, Slave In

The master out, slave in (MOSI) line is configured as an output in a master device and as an input in a slave device. The MOSI line is one of two lines that transfer serial data in one direction with the most significant bit sent first.

Master In, Slave Out

The master in, slave out (MISO) line is configured as an input in a master device and as an output in a slave device. The MISO is one of two lines that transfer serial data in one direction with the most significant bit sent first. The MISO line of a slave device is placed in a high-impedance state if slave is not selected ($\overline{SS}=1$).

Serial Clock

The serial clock (SCK) is used to synchronize both data in and out of a device via the MOSI and MISO lines. The master and slave devices can exchange a byte of information during a sequence of eight clock cycles. Since SCK is generated by the master device, this line becomes an input on a slave device.

As shown in Figure 15, four possible timing relationships may be chosen by using control bits CPOL and CPHA in the serial peripheral control register (SPCR). Both master and slave devices must operate with the same timing.

Two bits (SPR0 and SPR1) in the SPCR of the master device select the clock rate. In a slave device, SPR0 and SPR1 have no effect on SPI operation.

Slave Select

The slave select ($\overline{SS}$) input line selects a slave device. The $\overline{SS}$ line must be low prior to data transactions and must stay low for the duration of the transaction. The $\overline{SS}$ line on the master must be tied high; if the $\overline{SS}$ line goes low, a mode fault error flag (MODF) is set in the serial peripheral status register (SPSR).

When CPHA=0, the shift clock is the OR of $\overline{SS}$ with SCK. In this clock phase mode, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA=1, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA=1, $\overline{SS}$ may be left low for several SPI characters. In cases where there is only one SPI slave MCU, the slave MCU $\overline{SS}$ line could be tied to V_{SS} as long as CPHA=1 clock modes are used.

FUNCTIONAL DESCRIPTION

A block diagram of the SPI is shown in Figure 16. In a master configuration, the CPU sends a signal to the master start logic, which originates an SPI clock (SCK) based on the internal processor clock. As a master device, data is parallel loaded into the 8-bit shift register from the internal bus during a write cycle and then serially shifted

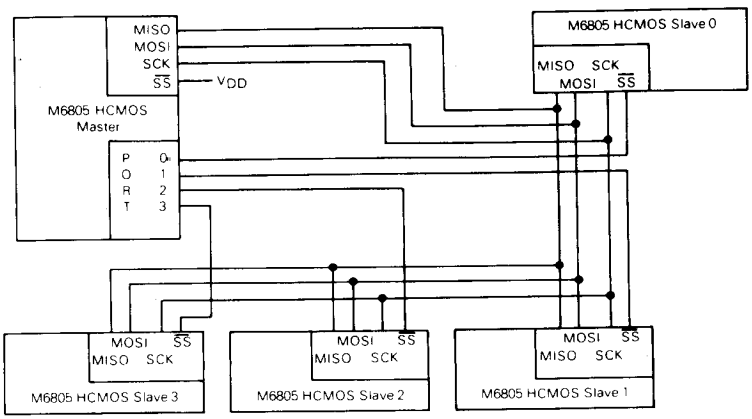


Figure 14. Master-Slave System Configuration

This Material Copyrighted By Its Respective Manufacturer

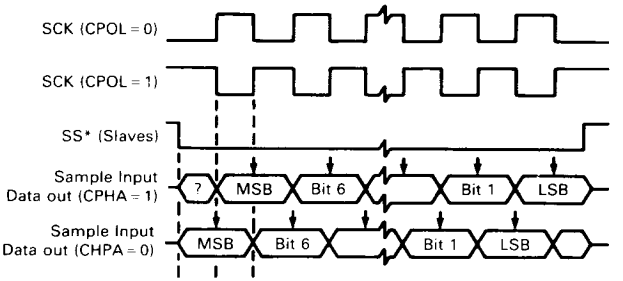


Figure 15. Data Clock Timing Diagram

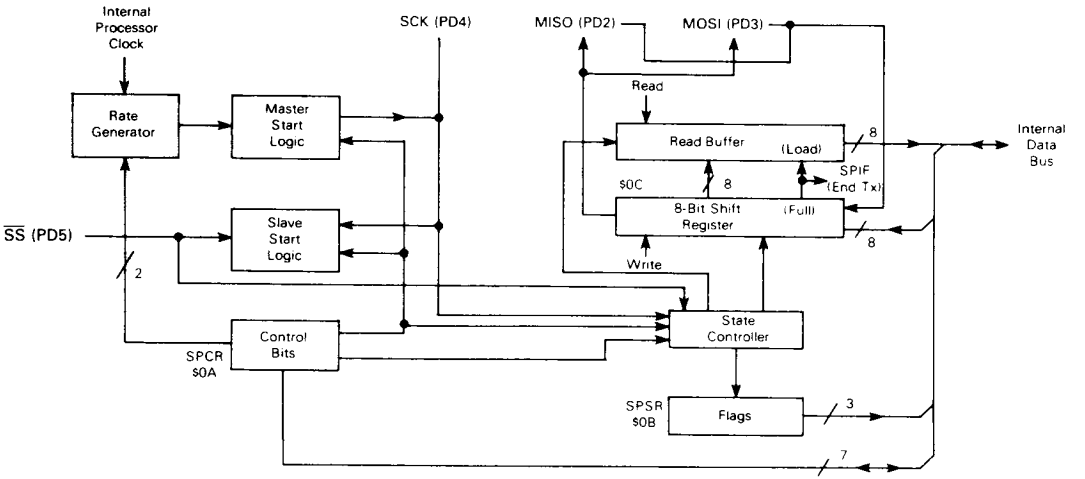


Figure 16. SPI Block Diagram

via the MOSI pin to the slave devices. During a read cycle, data is applied serially from a slave device via the MISO pin to the 8-bit shift register. Data is then parallel transferred to the read buffer and made available to the internal data bus during a CPU read cycle.

In a slave configuration, the slave start logic receives a logic low at the $\overline{SS}$ pin and a clock input at the SCK pin. This synchronizes the slave with the master. Data from the master is received serially at the slave MOSI pin and shifted into the 8-bit shift register for a parallel transfer to the read buffer. During a write cycle, data is parallel loaded into the 8-bit shift register from the internal data bus, awaiting the clocks from the master to shift out serially to the MISO pin and then to the master device.

Figure 17 illustrates the MOSI, MISO, SCK, and $\overline{SS}$ master-slave interconnections.

REGISTERS

There are three registers in the SPI that provide control, status, and data storage functions. These registers, the serial peripheral control register (SPCR), serial peripheral status register (SPSR), and serial peripheral data I/O register (SPDR), are described in the following paragraphs.

Serial Peripheral Control Register $0A

The SPCR provides control of individual SPI functions such as interrupt and system enabling/disabling, master/slave mode select, and clock polarity/phase/rate select.

7	6	5	4	3	2	1	0
SPIE	SPE	—	MSTR	CPOL	CPHA	SPR1	SPR0
RESET:							
0	0	—	0	U	U	U	U

SPIE — Serial Peripheral Interrupt Enable
1 = SPI interrupt enabled
0 = SPI interrupt disabled
SPE — Serial Peripheral System Enable
1 = SPI system on
0 = SPI system off

This Material Copyrighted By Its Respective Manufacturer

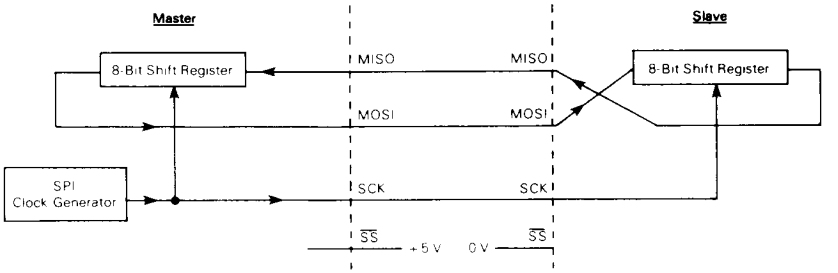


Figure 17. SPI Master-Slave Interconnections

MSTR — Master Mode Select
1 = Master mode
0 = Slave mode

CPOL — Clock Polarity
Clock polarity bit controls the clock value and is used in conjunction with the clock phase (CPHA) bit.
1 = SCK line idles high
0 = SCK line idles in low state

CPHA — Clock Phase
Clock phase bit along with CPOL controls the clock-data relationship between the master and slave devices. CPOL selects one of two clocking protocols.
1 = $\overline{SS}$ is an output enable control.
0 = Shift clock is the OR of SCK with $\overline{SS}$.
When $\overline{SS}$ is low, first edge of SCK invokes first data sample.

SPR0, SPR1 — SPI Clock Rate Bits
Two clock rate bits are used to select one of four clock rates to be used as SCK in the master mode. In the slave mode, the two clock rate bits have no effect. Clock rate selection is shown in the following table.

Bit 5 – Not used
Can read either one or zero

SPI Clock Rate Selection

SPR1	SPR0	Internal Processor Clock Divided By
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register $0B

The SPSR contains three status bits.

7	6	5	4	3	2	1	0
SPIF	WCOL		MODF	—	—	—	—
RESET:							
0	0	—	0	—	—	—	—

SPIF — Serial Peripheral Data Transfer Flag
1 = Indicates data transfer completed between processor and external device.
(If SPIF = 1 and SPIE = 1, SPI interrupt is enabled.)
0 = Clearing is accomplished by reading SPSR (with SPIF = 1) followed by SPDR access.

WCOL — Write Collision
1 = Indicates an attempt is made to write to SPDR while data transfer is in process.
0 = Clearing is accomplished by reading SPSR (with WCOL = 1), followed by SPDR access.

MODF — Mode Fault Flag
1 = Indicates multi-master system control conflict.
0 = Clearing is accomplished by reading SPSR (with MODF = 1), followed by a write to the SPCR.

Bits 0–3, and 5 — Not used
Can read either zero or one

Serial Peripheral Data I/O Register $0C

The SPDR is a read write register used to receive and transmit SPI data.

7	6	5	4	3	2	1	0
SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
RESET:							
U	U	U	U	U	U	U	U

A write to the SPDR places data directly into the shift register for transmission. Only a write to this register will initiate transmission reception of another byte and will only occur in the master device. On completion of byte transmission, the SPIF status bit is set in both master and slave devices.

A read to the SPDR causes the buffer to be read. The first SPIF status bit must be cleared by the time a second data transfer from the shift register to the read buffer begins, or an overrun condition will exist. In overrun cases, the byte causing the overrun is lost.

INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

This MCU uses all the instructions available in the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. This instruction allows unsigned multiplication of the contents of the accumulator

This Material Copyrighted By Its Respective Manufacturer

(A) and the index register (X). The high-order product is then stored in the index register, and the low-order product is stored in the accumulator. A detailed definition of the MUL instruction is shown below.

Operation	X:A X*A			
Description	Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator and index register			
Condition Codes	H: Cleared I: Not affected N: Not affected Z: Not affected C: Cleared			
Source	MUL			
Form(s)	Addressing Mode	Cycles	Bytes	Opcode
	Inherent	11	1	$42

3

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic Compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR

READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (Twos Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST
Multiply	MUL

BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, ROM, and on-chip RAM reside. An additional feature allows the software to test and branch on the state

This Material Copyrighted By Its Respective Manufacturer

of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n = 0 . . . 7)
Branch if Bit n is Clear	BRCLR n (n = 0 . . . 7)
Set Bit n	BSET n (n = 0 . . . 7)
Clear Bit n	BCLR n (n = 0 . . . 7)

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

OPCODE MAP SUMMARY

Table 5 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions; the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this two-byte instruction, K would typically be in X with the address of

This Material Copyrighted By Its Respective Manufacturer

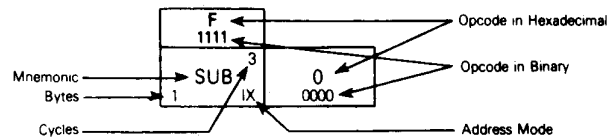
Table 5. Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB 0 0000	BSC 1 0001	REL 2 0010	DIR 3 0011	INH 4 0100	INH 5 0101	IX1 6 0110	IX 7 0111	INH 8 1000	INH 9 1001	IMM A 1010	DIR B 1011	EXT C 1100	IX2 D 1101	IX1 E 1110	IX F 1111	
0 0000	BRSET0 3 BTB	BSET0 2 BSC	BRA 2 REL	NEG 5 2 DIR	NEGA 3 1 INH	NEGX 3 1 INH	NEG 6 2 IX1	NEG 5 1 IX	RTI 9 1 INH		SUB 2 2 IMM	SUB 3 2 DIR	SUB 4 3 EXT	SUB 5 3 IX2	SUB 4 2 IX1	SUB 3 1 IX	0 0000
1 0001	BRCLR0 3 BTB	BCLR0 2 BSC	BRN 3 2 REL						RTS 6 1 INH		CMP 2 2 IMM	CMP 3 2 DIR	CMP 4 3 EXT	CMP 5 3 IX2	CMP 4 2 IX1	CMP 3 1 IX	1 0001
2 0010	BRSET1 3 BTB	BSET1 2 BSC	BHI 3 2 REL		MUL 11 1 INH						SBC 2 2 IMM	SBC 3 2 DIR	SBC 4 3 EXT	SBC 5 3 IX2	SBC 4 2 IX1	SBC 3 1 IX	2 0010
3 0011	BRCLR1 3 BTB	BCLR1 2 BSC	BLS 3 2 REL	COM 5 2 DIR	COMA 3 1 INH	COMX 3 1 INH	COM 6 2 IX1	COM 5 1 IX	SWI 10 1 INH		CPX 2 2 IMM	CPX 3 2 DIR	CPX 4 3 EXT	CPX 5 3 IX2	CPX 4 2 IX1	CPX 3 1 IX	3 0011
4 0100	BRSET2 3 BTB	BSET2 2 BSC	BCC 3 2 REL	LSR 5 2 DIR	LSRA 3 1 INH	LSRX 3 1 INH	LSR 6 2 IX1	LSR 5 1 IX			AND 2 2 IMM	AND 3 2 DIR	AND 4 3 EXT	AND 5 3 IX2	AND 4 2 IX1	AND 3 1 IX	4 0100
5 0101	BRCLR2 3 BTB	BCLR2 2 BSC	BCS 3 2 REL								BIT 2 2 IMM	BIT 3 2 DIR	BIT 4 3 EXT	BIT 5 3 IX2	BIT 4 2 IX1	BIT 3 1 IX	5 0101
6 0110	BRSET3 3 BTB	BSET3 2 BSC	BNE 3 2 REL	ROR 5 2 DIR	RORA 3 1 INH	RORX 3 1 INH	ROR 6 2 IX1	ROR 5 1 IX			LDA 2 2 IMM	LDA 3 2 DIR	LDA 4 3 EXT	LDA 5 3 IX2	LDA 4 2 IX1	LDA 3 1 IX	6 0110
7 0111	BRCLR3 3 BTB	BCLR3 2 BSC	BEQ 3 2 REL	ASR 5 2 DIR	ASRA 3 1 INH	ASRX 3 1 INH	ASR 6 2 IX1	ASR 5 1 IX		TAX 2 1 INH		STA 4 2 DIR	STA 5 3 EXT	STA 6 3 IX2	STA 5 2 IX1	STA 4 1 IX	7 0111
8 1000	BRSET4 3 BTB	BSET4 2 BSC	BHCC 3 2 REL	LSL 5 2 DIR	LSLA 3 1 INH	LSLX 3 1 INH	LSL 6 2 IX1	LSL 5 1 IX		CLC 2 1 INH	EOR 2 2 IMM	EOR 3 2 DIR	EOR 4 3 EXT	EOR 5 3 IX2	EOR 4 2 IX1	EOR 3 1 IX	8 1000
9 1001	BRCLR4 3 BTB	BCLR4 2 BSC	BHCS 3 2 REL	ROL 5 2 DIR	ROLA 3 1 INH	ROLX 3 1 INH	ROL 6 2 IX1	ROL 5 1 IX		SEC 2 1 INH	ADC 2 2 IMM	ADC 3 2 DIR	ADC 4 3 EXT	ADC 5 3 IX2	ADC 4 2 IX1	ADC 3 1 IX	9 1001
A 1010	BRSET5 3 BTB	BSET5 2 BSC	BPL 3 2 REL	DEC 5 2 DIR	DECA 3 1 INH	DECX 3 1 INH	DEC 6 2 IX1	DEC 5 1 IX		CLI 2 1 INH	ORA 2 2 IMM	ORA 3 2 DIR	ORA 4 3 EXT	ORA 5 3 IX2	ORA 4 2 IX1	ORA 3 1 IX	A 1010
B 1011	BRCLR5 3 BTB	BCLR5 2 BSC	BMI 3 2 REL							SEI 2 1 INH	ADD 2 2 IMM	ADD 3 2 DIR	ADD 4 3 EXT	ADD 5 3 IX2	ADD 4 2 IX1	ADD 3 1 IX	B 1011
C 1100	BRSET6 3 BTB	BSET6 2 BSC	BMC 3 2 REL	INC 5 2 DIR	INCA 3 1 INH	INCX 3 1 INH	INC 6 2 IX1	INC 5 1 IX		RSP 2 1 INH		JMP 2 2 DIR	JMP 3 3 EXT	JMP 4 3 IX2	JMP 3 2 IX1	JMP 2 1 IX	C 1100
D 1101	BRCLR6 3 BTB	BCLR6 2 BSC	BMS 3 2 REL	TST 4 2 DIR	TSTA 3 1 INH	TSTX 3 1 INH	TST 5 2 IX1	TST 4 1 IX		NOP 2 1 INH	BSR 6 2 REL	JSR 5 2 DIR	JSR 6 3 EXT	JSR 7 3 IX2	JSR 6 2 IX1	JSR 5 1 IX	D 1101
E 1110	BRSET7 3 BTB	BSET7 2 BSC	BIL 3 2 REL						STOP 2 1 INH		LDX 2 2 IMM	LDX 3 2 DIR	LDX 4 3 EXT	LDX 5 3 IX2	LDX 4 2 IX1	LDX 3 1 IX	E 1110
F 1111	BRCLR7 3 BTB	BCLR7 2 BSC	BIH 3 2 REL	CLR 5 2 DIR	CLRA 3 1 INH	CLRX 3 1 INH	CLR 6 2 IX1	CLR 5 1 IX	WAIT 2 1 INH	TXA 2 1 INH		STX 4 2 DIR	STX 5 3 EXT	STX 6 3 IX2	STX 5 2 IX1	STX 4 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



This Material Copyrighted By Its Respective Manufacturer

the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct addressing of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction.

BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from −125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

3

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS (Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	V_{SS} −0.3 to V_{DD} +0.3	V
Self-Check Mode ($\overline{IRQ}$ Pin Only)	V_{in}	V_{SS} −0.3 to 2 × V_{DD} + 0.3	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range MC68HC05C4P, FN (Standard) MC68HC05C4CP, CFN (Extended) MC68HC05C4MP, MFN (Automotive)	T_A	T_L to T_H 0 to +70 −40 to +85 −40 to +125	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Plastic Leaded Chip Carrier (PLCC)	θ_{JA}	 60 70	°C/W

This Material Copyrighted By Its Respective Manufacturer

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{I/O}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

V_{DD} = 4.5 V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	3.26 kΩ	2.38 kΩ	50 pF
PD0,PD5, PD7	1.9 kΩ	2.26 kΩ	200 pF

V_{DD} = 3.0 V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	10.91 kΩ	6.32 kΩ	50 pF
PD0, PD5, PD7	6 kΩ	6 kΩ	200 pF

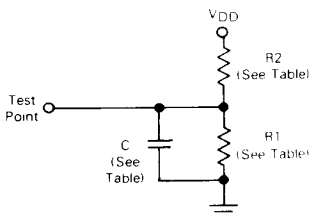


Figure 18. Equivalent Test Load

This Material Copyrighted By Its Respective Manufacturer

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage (I_{Load} = 0.8 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 19) (I_{Load} = 1.6 mA) PD1-PD4 (see Figure 20)	V_{OH}	 $V_{DD} - 0.8$ $V_{DD} - 0.8$	 — —	 — —	V
Output Low Voltage (see Figure 21) (I_{Load} = 1.6 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	—	—	0.4	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes) Run (see Figures 22 and 23) Wait (see Figures 22 and 23) Stop (see Figure 23) 25°C 0° to 70°C (Standard) -40° to +85°C -40° to +125°C	I_{DD}	 — — — — — —	 3.5 1.6 2.0 — — —	 7.0 4.0 50 140 180 250	 mA mA µA µA µA µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	—	—	±10	µA
Input Current $\overline{RESET}$, IRQ, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, IRQ, TCAP, PD0-PD5, PD7	 C_{out} C_{in}	 — —	 — —	 12 8	pF

NOTES:
1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Standard temperature range is 0° to 70°C. Extended temperature (−40° to +85°C, −40° to +125°C) versions and a 25°C only version are available.
8. Wait I_{DD} is affected linearly by the OSC2 capacitance.

This Material Copyrighted By Its Respective Manufacturer

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage (I_{Load} = 0.2 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (see Figure 19) (I_{Load} = 1.6 mA) PD1-PD4 (see Figure 20)	V_{OH}	 $V_{DD} - 0.3$ $V_{DD} - 0.3$	 — —	 — —	V
Output Low Voltage (see Figure 21) (I_{Load} = 0.4 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	—	—	0.3	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	$0.7 \cdot V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \cdot V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes) Run (see Figures 22 and 24) Wait (see Figures 22 and 24) Stop (see Figure 24) 25°C 0° to 70°C (Standard) −40° to +85°C −40° to +125°C	I_{DD}	 — — — — — —	 1.0 0.5 1.0 — — —	 2.5 1.4 30 80 120 175	 mA mA µA µA µA µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	—	—	±10	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	—	—	−1	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$, TCAP, PD0-PD5, PD7	 C_{out} C_{in}	 — —	 — —	 12 8	pF

NOTES:
1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active (SPE = TE = RE = 0). If SPI, SCI active (SPE = TE = RE = 1) add 10% current draw.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Standard temperature range is 0° to 70°C. Extended temperature (−40° to −85°C, −40° to −125°C) versions and a 25°C only version are available.
8. Wait I_{DD} is affected linearly by the OSC2 capacitance.

This Material Copyrighted By Its Respective Manufacturer

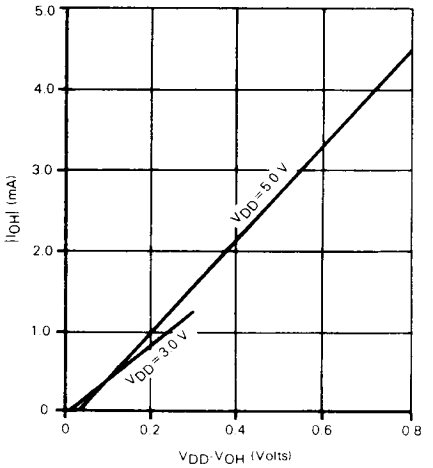


Figure 19. Typical V_{OH} vs I_{OH} for Ports A, B, C, and TCMP

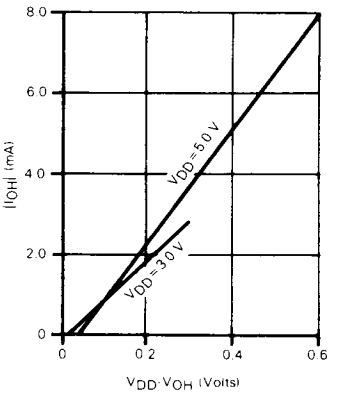


Figure 20. Typical V_{OH} vs I_{OH} for PD1-PD4

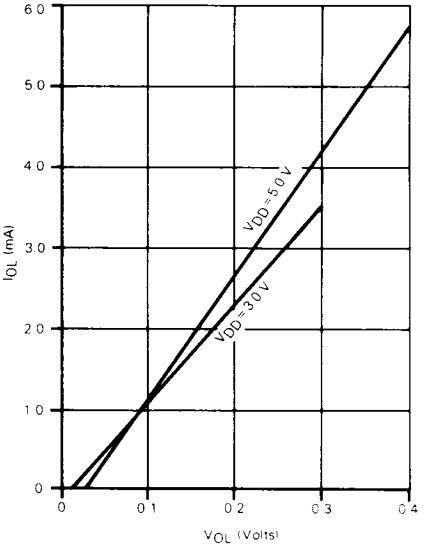


Figure 21. Typical V_{OL} vs I_{OL} for All Ports

This Material Copyrighted By Its Respective Manufacturer

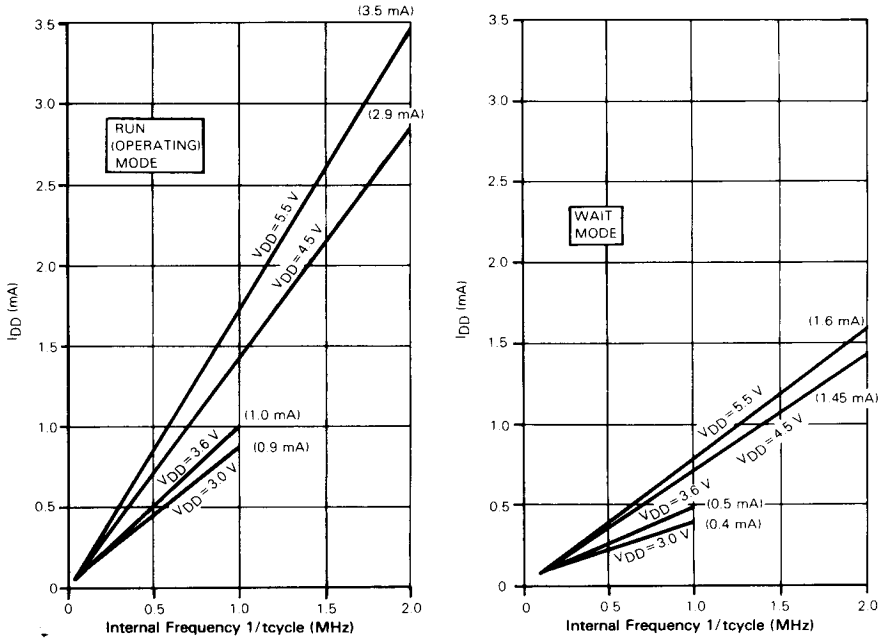


Figure 22. Typical Current vs Internal Frequency for Run and Wait Modes

This Material Copyrighted By Its Respective Manufacturer

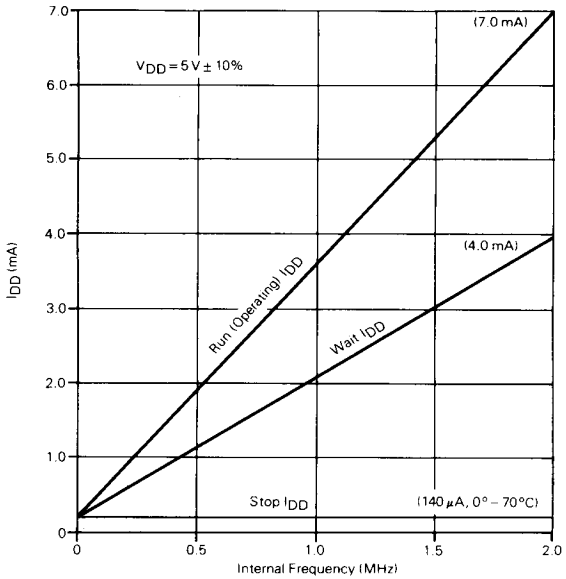


Figure 23. Maximum I_{DD} vs Frequency for V_{DD} = 5.0 Vdc

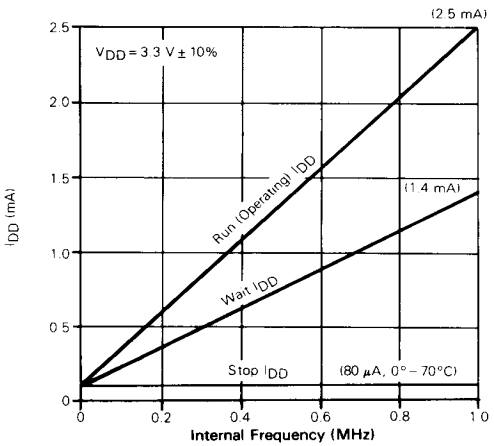


Figure 24. Maximum I_{DD} vs Frequency for V_{DD} = 3.3 Vdc

This Material Copyrighted By Its Respective Manufacturer

CONTROL TIMING

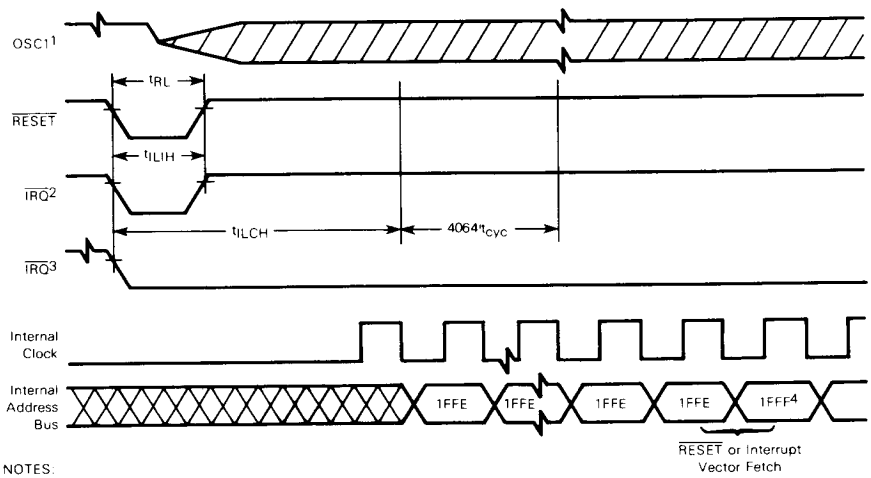
(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	f_{osc}	 — dc	 4.2 4.2	MHz
Internal Operating Frequency Crystal ($f_{osc} \div 2$) External Clock ($f_{osc} \div 2$)	f_{op}	 — dc	 2.1 2.1	MHz
Cycle Time (see Figure 28)	t_{cyc}	480	—	ns
Crystal Oscillator Startup Time (see Figure 28)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 25)	t_{ILCH}	—	100	ms
$\overline{RESET}$ Pulse Width (see Figure 28)	t_{RL}	1.5	—	t_{cyc}
Timer Resolution** Input Capture Pulse Width (see Figure 26) Input Capture Pulse Period (see Figure 26)	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4.0 125 ***	 — — —	 t_{cyc} ns t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	125	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.



NOTES:
1. Represents the internal gating of the OSC1 pin.
2. $\overline{IRQ}$ pin edge-sensitive mask option.
3. $\overline{IRQ}$ pin level and edge-sensitive mask option.
4. $\overline{RESET}$ vector address shown for timing example.

Figure 25. Stop Recovery Timing Diagram

This Material Copyrighted By Its Respective Manufacturer

CONTROL TIMING

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	f_{osc}	 — dc	 2.0 2.0	MHz
Internal Operating Frequency Crystal ($f_{osc} \div 2$) External Clock ($f_{osc} \div 2$)	f_{op}	 — dc	 1.0 1.0	MHz
Cycle Time (see Figure 28)	t_{cyc}	1000	—	ns
Crystal Oscillator Startup Time (see Figure 28)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (see Figure 25)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width — Excluding Power-Up (see Figure 28)	t_{RL}	1.5	—	t_{cyc}
Timer Resolution** Input Capture Pulse Width (see Figure 26) Input Capture Pulse Period (see Figure 26)	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4.0 250 ***	 — — —	 t_{cyc} ns t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 8)	t_{ILIH}	250	—	ns
Interrupt Pulse Period (see Figure 8)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

***The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

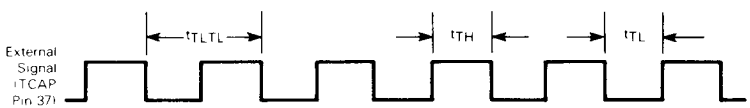


Figure 26. Timer Relationships

This Material Copyrighted By Its Respective Manufacturer

SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H) (see Figure 27)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency				
	Master	$f_{op(m)}$	dc	0.5	f_{op}
	Slave	$f_{op(s)}$	dc	2.1	MHz
1	Cycle Time				
	Master	$t_{cyc(m)}$	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	480	—	ns
2	Enable Lead Time				
	Master	$t_{lead(m)}$	*	—	ns
	Slave	$t_{lead(s)}$	240	—	ns
3	Enable Lag Time				
	Master	$t_{lag(m)}$	*	—	ns
	Slave	$t_{lag(s)}$	240	—	ns
4	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	340	—	ns
	Slave	$t_{w(SCKH)s}$	190	—	ns
5	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	340	—	ns
	Slave	$t_{w(SCKL)s}$	190	—	ns
6	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	100	—	ns
	Slave	$t_{su(s)}$	100	—	ns
7	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	100	—	ns
	Slave	$t_{h(s)}$	100	—	ns
8	Access Time (Time to Data Active from High-Impedance State)				
	Slave	t_a	0	120	ns
9	Disable Time (Hold Time to High-Impedance State)				
	Slave	t_{dis}	—	240	ns
10	Data Valid				
	Master (Before Capture Edge)	$t_{v(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)**	$t_{v(s)}$	—	240	ns
11	Data Hold Time (Outputs)				
	Master (After Capture Edge)	$t_{ho(m)}$	0.25	—	$t_{cyc(m)}$
	Slave (After Enable Edge)	$t_{ho(s)}$	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{rm}	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{rs}	—	2.0	µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, and MISO)	t_{fm}	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, and SS)	t_{fs}	—	2.0	µs

*Signal production depends on software.
**Assumes 200 pF load on all SPI pins.

This Material Copyrighted By Its Respective Manufacturer

SERIAL PERIPHERAL INTERFACE (SPI) TIMING

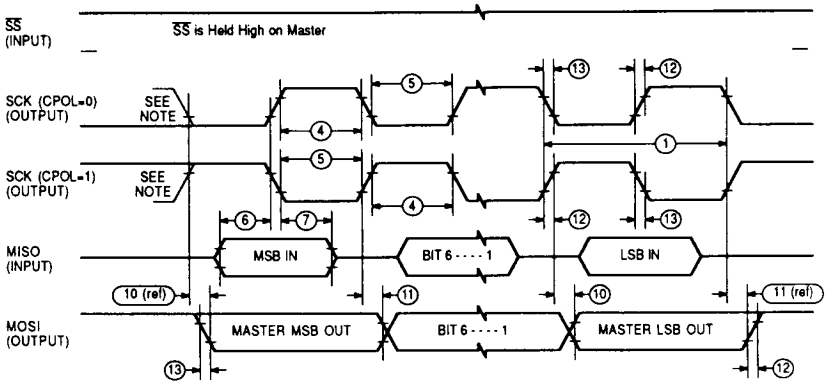
(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H) (see Figure 27)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	 $f_{op(m)}$ $f_{op(s)}$	 dc dc	 0.5 1.0	 f_{op} MHz
1	Cycle Time Master Slave	 $t_{cyc(m)}$ $t_{cyc(s)}$	 2.0 1.0	 — —	 t_{cyc} µs
2	Enable Lead Time Master Slave	 $t_{lead(m)}$ $t_{lead(s)}$	 * 500	 — —	 ns ns
3	Enable Lag Time Master Slave	 $t_{lag(m)}$ $t_{lag(s)}$	 * 500	 — —	 ns ns
4	Clock (SCK) High Time Master Slave	 $t_{w(SCKH)m}$ $t_{w(SCKH)s}$	 720 400	 — —	 µs ns
5	Clock (SCK) Low Time Master Slave	 $t_{w(SCKL)m}$ $t_{w(SCKL)s}$	 720 400	 — —	 µs ns
6	Data Setup Time (Inputs) Master Slave	 $t_{su(m)}$ $t_{su(s)}$	 200 200	 — —	 ns ns
7	Data Hold Time (Inputs) Master Slave	 $t_{h(m)}$ $t_{h(s)}$	 200 200	 — —	 ns ns
8	Access Time (Time to Data Active from High-Impedance State) Slave	 t_a	 0	 250	 ns
9	Disable Time (Hold Time to High-Impedance State) Slave	 t_{dis}	 —	 500	 ns
10	Data Valid Master (Before Capture Edge) Slave (After Enable Edge)**	 $t_{v(m)}$ $t_{v(s)}$	 0.25 —	 — 500	 $t_{cyc(m)}$ ns
11	Data Hold Time (Outputs) Master (After Capture Edge) Slave (After Enable Edge)	 $t_{ho(m)}$ $t_{ho(s)}$	 0.25 0	 — —	 $t_{cyc(m)}$ ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{rm} t_{rs}	 — —	 200 2.0	 ns µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{fm} t_{fs}	 — —	 200 2.0	 ns µs

*Signal production depends on software.
**Assumes 200 pF load on all SPI pins.

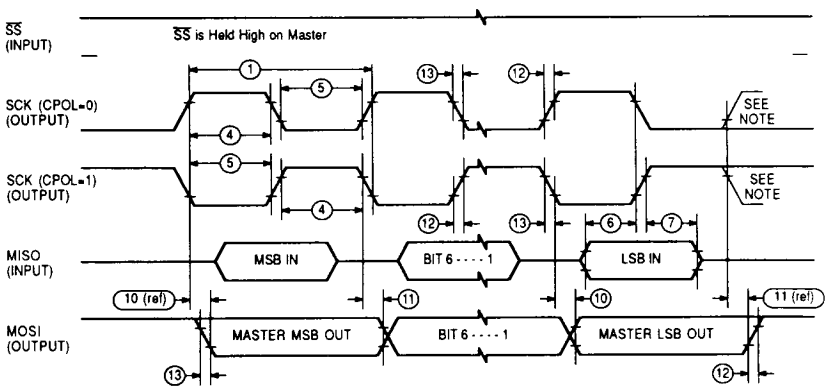
3

This Material Copyrighted By Its Respective Manufacturer



NOTE: This first clock edge is generated internally but is not seen at the SCK pin.

a) SPI MASTER TIMING (CPHA = 0)

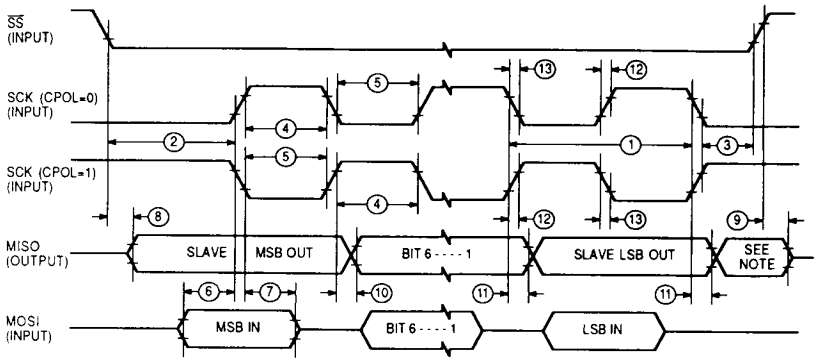


NOTE: This last clock edge is generated internally but is not seen at the SCK pin.

b) SPI MASTER TIMING (CPHA = 1)

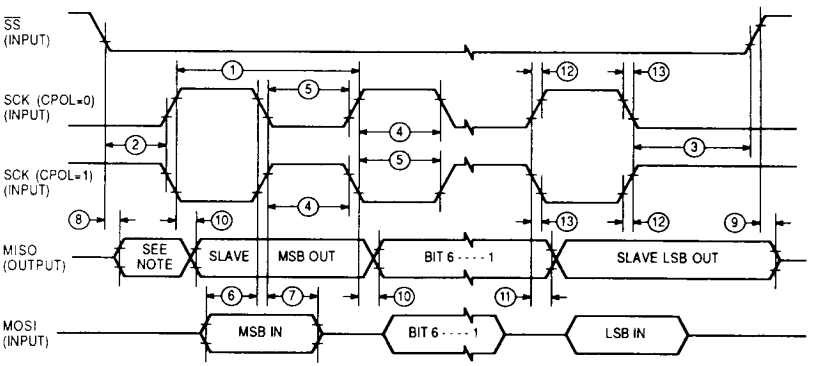
Figure 27. SPI Timing Diagrams (Sheet 1 of 2)

This Material Copyrighted By Its Respective Manufacturer



NOTE: Not defined but normally MSB of character just received.

c) SPI SLAVE TIMING (CPHA = 0)



NOTE: Not defined but normally LSB of character previously transmitted.

d) SPI SLAVE TIMING (CPHA = 1)

Figure 27. SPI Timing Diagrams (Sheet 2 of 2)

This Material Copyrighted By Its Respective Manufacturer

3

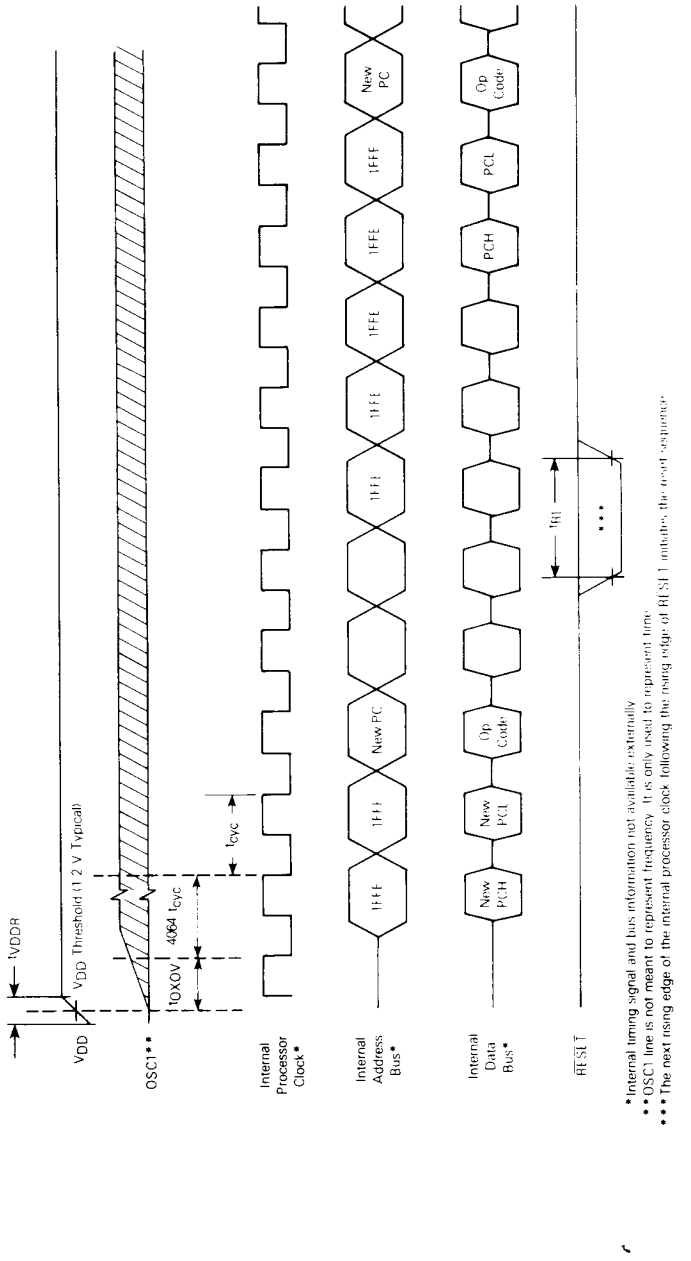


Figure 28. Power-On Reset and RESET

This Material Copyrighted By Its Respective Manufacturer

ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola in the following media:

MDOS™, disk file
MS™-DOS PC-DOS disk file (360K)
EPROM(s) 2764, MCM68764, MCM68766, or EEPROM MC68HC805C4

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field service office, a sales person, or a Motorola representative.

FLEXIBLE DISKS

A flexible disk (MS-DOS PC-DOS disk file), programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. The diskette should be clearly labeled with the customer's name, data, project or product name, and the name of the file containing the pattern.

In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

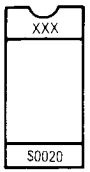
MS-DOS/PC-DOS Disk File

MS-DOS is Microsoft's Disk Operating System. PC-DOS is the IBM® Personal Computer (PC) Disk Operating System. Disk media submitted must be a standard density (360K) double-sided 5 1 4 inch compatible floppy diskette. The diskette must contain object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by M6805 cross assemblers and linkers on IBM PC style machines.

EPROMs

A 2764, 68764, or 68766 type EPROM, programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one 2764, 68764, or 68766 EPROM device, the EPROM must be programmed as described in the following paragraphs.

For an MC68HC805C4 MCU start the page zero, user ROM at EEPROM address $0020 through $004F. Start the user ROM at EEPROM address $0100 through $10FF with vectors from $1FF4 to $1FFF. All unused bytes, including the user's space, must be set to zero. For shipment to Motorola, EPROMs should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.



xxx Customer ID

Verification Media

All original pattern media (EPROMs or floppy disks) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked, and the verification form should be completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for the creation of the customer mask. To aid in the verification process, Motorola will program *customer supplied* blank EPROM(s) or DOS disks from the data file used to create the custom mask.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask, but are for the purpose of ROM verification only. For expediency, the MCUs are unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with the minimum order quantity, but are not production parts. RVUs are not backed or guaranteed by Motorola Quality Assurance.

ORDERING INFORMATION

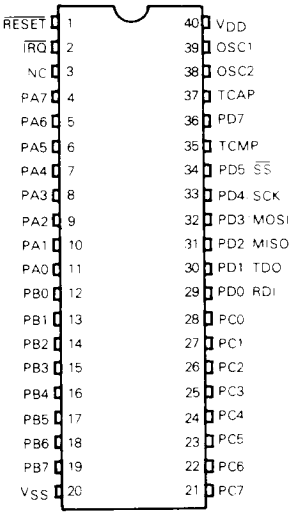
The following table provides ordering information pertaining to the package type, temperature, and MC order numbers for the MC68HC05C4 device.

Package Type	Temperature	MC Order Number
Plastic (P Suffix)	0 C to + 70 C	MC68HC05C4P
	40 C to + 85 C	MC68HC05C4CP
	- 40 to + 105 C	MC68HC05C4VP
	40 C to + 125 C	MC68HC05C4MP
PLCC (FN Suffix)	0 C to + 70 C	MC68HC05C4FN
	40 C to + 85 C	MC68HC05C4CFN
	40 C to + 105 C	MC68HC05C4VFN
	40 C to + 125 C	MC68HC05C4MFN

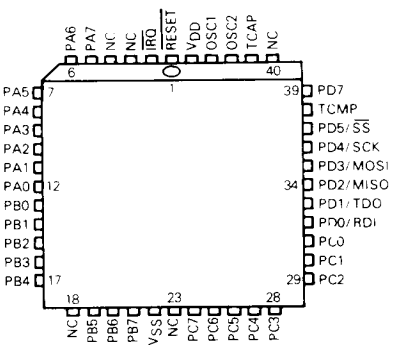
MDOS is a trademark of Motorola Inc.
MS is a trademark of Microsoft, Inc.
IBM is a registered trademark of International Business Machines Corporation.

This Material Copyrighted By Its Respective Manufacturer

40-PIN DUAL-IN-LINE PACKAGE



44-LEAD PLCC PACKAGE



NOTE: Bulk substrate tied to V_{SS}.

This Material Copyrighted By Its Respective Manufacturer