MC68705S3

Technical Summary
8-Bit EPROM Microcontroller Unit

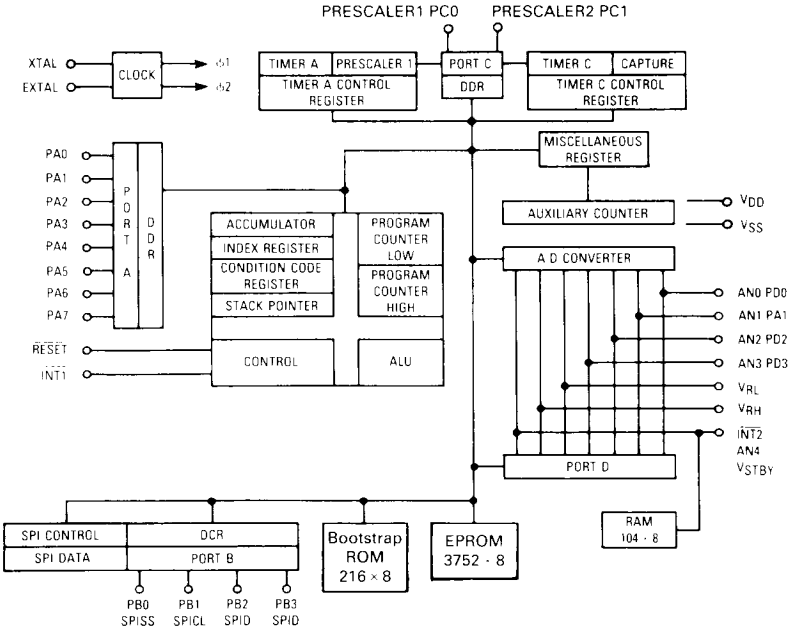
The MC68705 (HMOS) Microcontroller Unit (MCU)is an EPROM member of the MC6805 Family of microcontrollers. The user programmable EPROM allows program changes and lower volume applications. This high performance MCU has parallel I O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for detailed information, refer to *Advance Information 8-Bit Microcontroller* (ADI997R1) or contact your local Motorola sales office.

Refer to the block diagram for the hardware features and to the list below for additional features available on the MCU.

- One 7-Bit and One 15-Bit Software Programmable Prescaler
- On-chip Oscillator
- Memory Mapped I O
- Versatile Interrupt Handling
- Bit Manipulation
- Bit Test and Branch Instruction
- Vectored Interrupts
- Bootstrap Program in ROM
- 3752 Bytes of EPROM
- 104 Bytes of RAM
- Serial Peripheral Interface
- Two 8-Bit and One 16-Bit Timers
- A D Converter
- EPROM Read Inhibit Security Bit

3

BLOCK DIAGRAM



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

SIGNAL DESCRIPTION

V_{CC} and V_{SS}

Power is supplied to the microcontroller using these two pins. V_{CC} is +5.25 volts (±0.5Δ) power, and V_{SS} is ground.

NUM

This pin is for factory use only. It should be connected to V_{SS}.

$\overline{INT1}$, $\overline{INT2}$

These pins provide the capability for asynchronously applying an external interrupt to the MCU. Refer to **INTERRUPTS** for more detailed information.

XTAL, EXTAL

These pins provide control input for the on-chip clock oscillator circuit. A crystal, a ceramic resonator, a resistor capacitor combination, or an external signal (depending on setting of the Mask Option Register) is connected to these pins to provide a system clock.

RC Oscillator

With this option, a resistor/capacitor combination is connected to the oscillator pins as shown in Figure 1(c). Refer to Figure 2 for the relationship between R and f_{osc}.

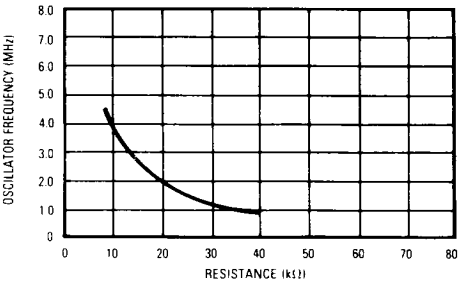


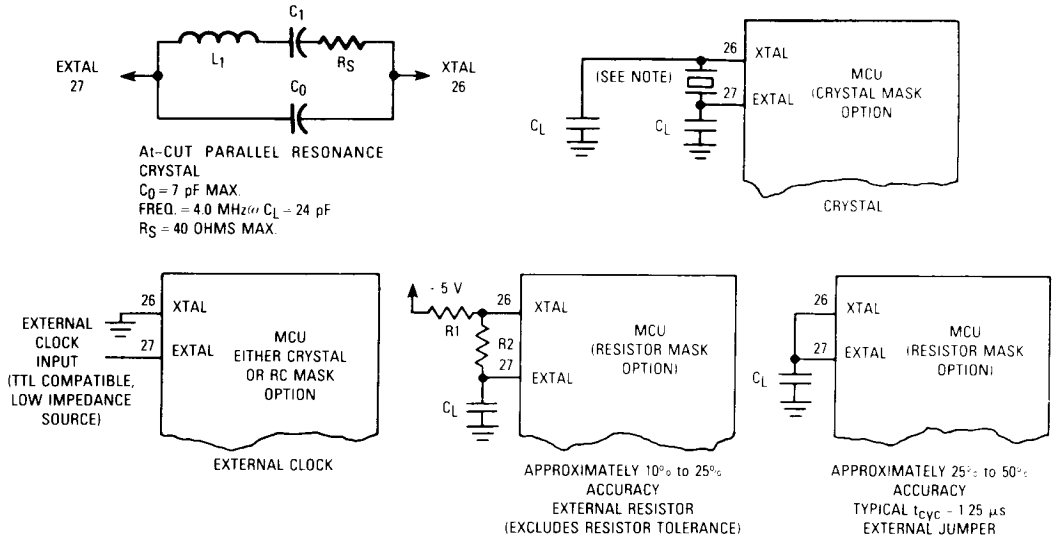
Figure 2. Typical Frequency vs Resistance for RC Oscillator Option Only

Crystal

The circuit shown in Figure 1(b) is recommended when using a crystal. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time.

External Clock

An external clock should be applied to the EXTAL input with the XTAL input grounded, as shown in Figure 1(d).



NOTE: The recommended C_L value with a 4.0 MHz crystal is 27 pF, maximum, including system distributed capacitance. For crystal frequencies other than 4 MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with 2 MHz crystal, use approximately 50 pF on EXTAL and approximately 50 pF on XTAL. The exact value depends on the motional-arm parametes of the crystal used.

Figure 1. Oscillator Connections

This option may only be used with the crystal oscillator option selected in the mask option register.

PC0, PC1

This pins allow an external input to decrement the internal timer/counter circuitry. Refer to **TIMERS** for additional information.

$\overline{\text{RESET}}$/VPP

This pin has a Schmitt trigger input. The MCU can be reset by pulling $\overline{\text{RESET}}$ low. The VPP input is used to input the programming voltage to the MCU EPROM. A 1K ohm pullup resistor should be used to allow proper operation of the reset and watchdog timer operations.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB3, PC0-PC1, PD0-PD6)

Ports A, B, and C are programmable as either inputs or outputs under software control of the data direction registers. Port D is a fixed input port and not controlled by any data direction register. Port D has up to five analog inputs, plus two voltage reference inputs when the analog-to-digital (A/D) converter is used (PD5/V_{RH}, PD4/V_{RL}) and an $\overline{\text{INT2}}$ input. If the analog input is used, the voltage reference pins (PD5/V_{RH} and PD4/V_{RL}) must be used in the analog mode. Refer to **INPUT/OUTPUT PORTS** for additional information.

INPUT/OUTPUT PORTS

INPUT/OUTPUT PROGRAMMING

Ports A, B, and C are programmable as either input or output under software control of the corresponding data direction register (DDR). The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic one for output and a logic zero for input. On reset, all DDRs are initialized to a logic zero state to put the ports in the input mode. The port output registers are not initialized on reset and should be written to before setting the DDR bits.

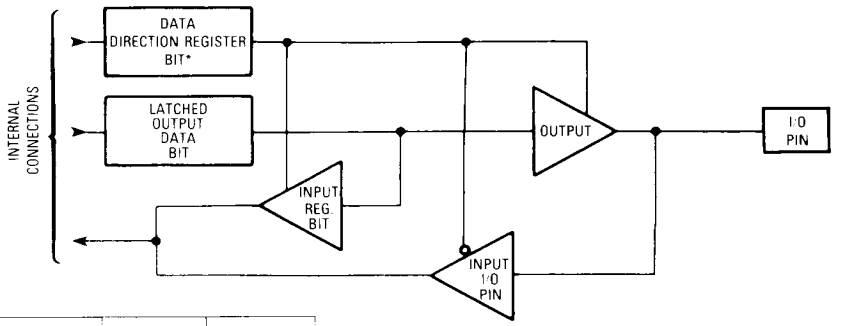
Port D provides the multiplexed analog inputs, reference voltages, and $\overline{\text{INT2}}$. These lines are shared with the port D digital inputs. PD0–PD3 may always be used as digital or analog inputs. The V_{RL} and V_{RH} lines are internally connected to the A/D resistor. Analog inputs may be prescaled to obtain the V_{RL} and V_{RH} recommended input voltage range.

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. Care must be exercised when using read-modify-write instructions since the data read corresponds to the pin level if the DDR is an input (zero) and, also, to the latched output when the DDR is an output (one). Refer to Table 1 for I/O functions and to Figure 3 for typical port circuitry.

PORT B TOGGLE CAPABILITY

Port B0 and B1 registers have toggle capability at the timer underflow times. Under the control of the timer output cross-couple bit in the miscellaneous register (MR0), the overflow pulses from timer A, B, and C are directed to port B0 and B1 data registers. See Figure 4 for port B configuration flow chart.

An incoming toggle pulse on port B0 is allowed to toggle the data register if port B DCR bit 4 (DCR4) is



Data Direction Register Bit	Output Data Bit	Output State	Input To MCU
1	0	0	0
1	1	1	1
0	X	High-Z**	Pin

*DDR is a write-only register and reads as all "1s".
**Ports A, B, and C are three-state ports. Port A has optional internal pullup devices to provide CMOS drive capability. See Electrical Characteristics tables for complete information.

Figure 3. Typical Port I/O Circuitry and Register Configuration

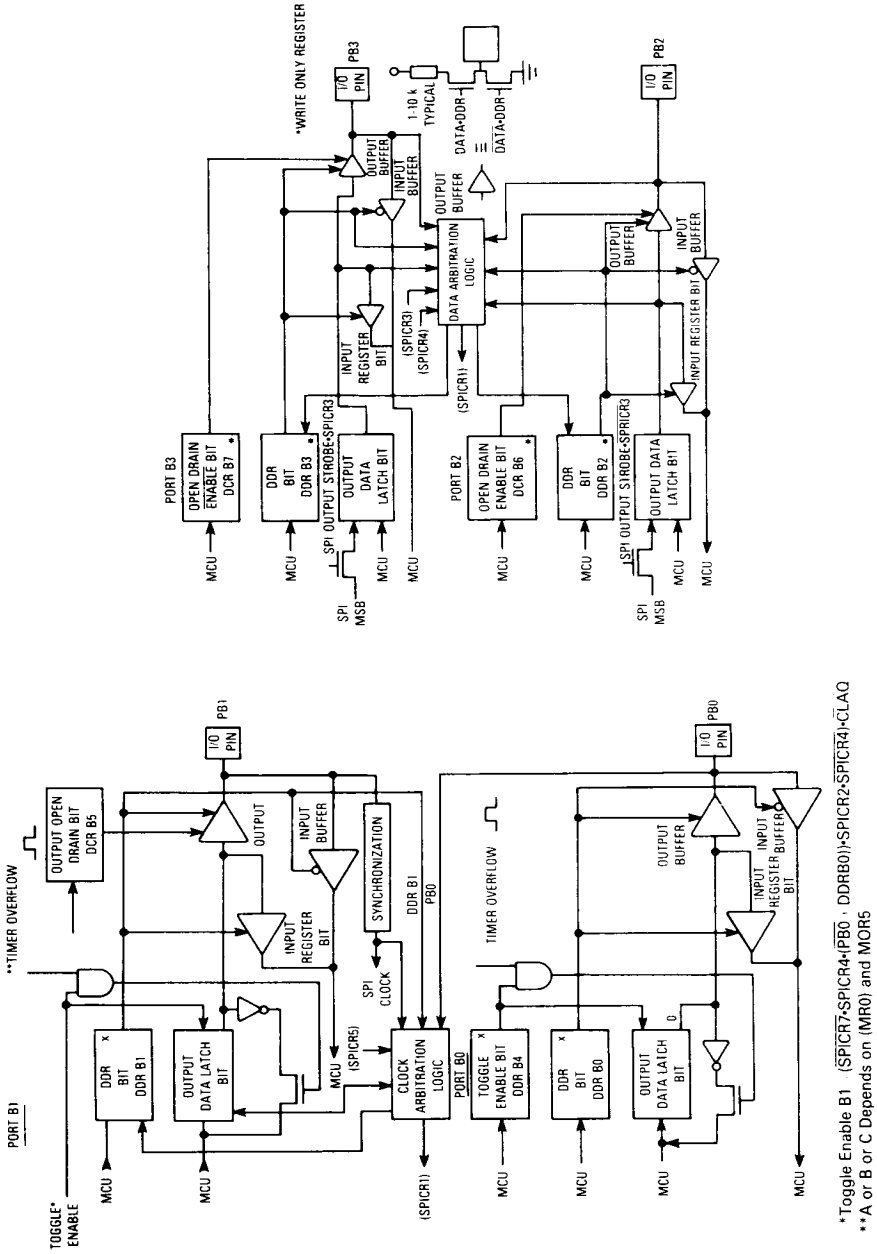


Figure 4. Port B Configuration

cleared. This bit is set on reset. An incoming toggle pulse on port B1 is allowed to toggle the port B1 data register under the following conditions governed by control bits in SPI control register and SPI clock arbitration flip-flop status.

PB1 toggle enable = $(\overline{SPICR7})$•SPICR4• $(\overline{PB0} + DDRB0)$ + SPICR2• $\overline{SPICR4}$•$\overline{CLAQ}$

where: SPICR7 = SPI interrupt request bit
SPICR4 = SPI operation enable bit
SPICR2 = port B1 toggle enable/start bit
CLAQ = clock arbitration flip-flop output

When PB1 toggle enable is asserted, the MCU write to PB1 data register is inhibited. When SPI is not used, SPICR4 and CLAQ are reset. Therefore, SPICR2 can directly control the port B1 toggle capability. Port toggle capability allows action on port B0 or B1 or both as a result of timer overflows. This method speeds up timer overflow to port service. A write to port B0 or B1 data registers is inhibited while the individual port toggle enable is asserted.

The port B DCR consists of four status bits (DCR4-DCR7) and four data direction bits (DCR0-DCR3). DCR4 is a toggle enable control bit for port B0. When cleared, the timer overflow pulse causes the data register on port B0 to toggle. Port A has an 8-bit and port C has a 2-bit wide data direction register.

MEMORY

The MCU is capable of addressing 4096 bytes of memory and I/O registers. The memory map is shown in Figure 5. The locations consist of user EPROM, bootstrap ROM, user RAM, eight timer registers, a mask option register (MOR), a miscellaneous register, a program control register, two A/D registers, two SPI registers, and four I/O port registers. The interrupt vectors are located from $FF8 to $FFF. The bootstrap is a mask-programmed ROM that allows the MCU to program its own EPROM.

The 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

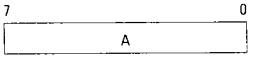
Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

REGISTERS

The MCU contains the registers described in the following paragraphs.

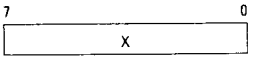
ACCUMULATOR (A)

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



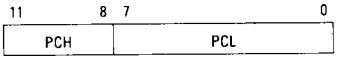
INDEX REGISTER (X)

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)

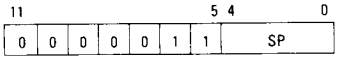
The program counter is a 12-bit register that contains the address of the next byte to be fetched.



STACK POINTER (SP)

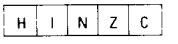
The stack pointer is a 12-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set at location $07F. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

The seven most-significant bits of the stack pointer are permanently set at 0000011. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum), which allows the programmer to use up to 15 levels of subroutine calls (less if interrupts are allowed).



CONDITION CODE REGISTER (CC)

The condition code register 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 (A, B, and C), the external ($\overline{INT1}$ and $\overline{INT2}$) interrupts, and the SPI interrupt are masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and is 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).

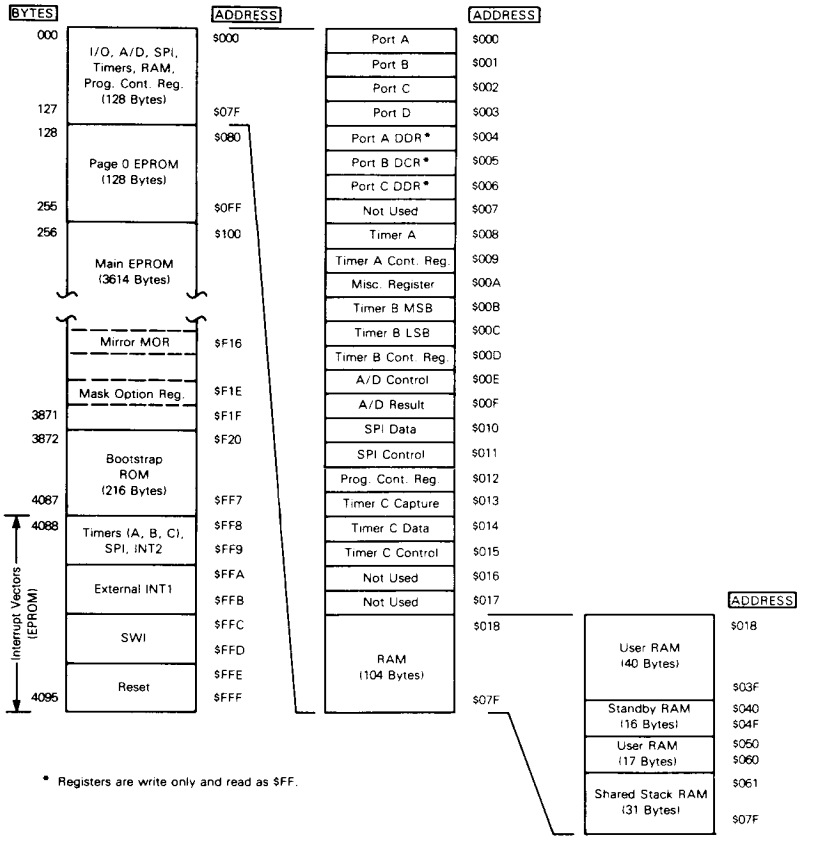


Figure 5. Memory Map

Zero (Z)

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

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.

MISCELLANEOUS REGISTERS (MR) $0A

This register contains control and status information related to $\overline{INT2}$, auxiliary counter, prescalers 1 and 2, and timer overflow.

MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
RESET:							
0	1	0	1			0	0

MR7 — $\overline{INT2}$ Interrupt Request Bit
If not masked by MR6, it causes an interrupt to the MCU; if the I bit in the CCR is clear, the MCU will acknowledge the interrupt.
1 = Interrupt requested
0 = Interrupt not requested

MR6 — $\overline{INT2}$ Interrupt Request Mask
1 = Inhibits $\overline{INT2}$ interrupt request
0 = Does not inhibit $\overline{INT2}$ interrupt request

MR5 — Auxiliary Counter Status/Preset Bit
If not masked by MR4, it will drive a switch to V_{SS} on the $\overline{RESET}$ pin causing the MCU to reset. This bit may

be used as an auxiliary counter preset bit. If MR5 is clear, a write of logic one will preset the auxiliary counter (MR5 will remain zero), and if set, a write of logic zero will preset the auxiliary counter.
1 = Auxiliary counter overflow
0 = Auxiliary counter clear

MR4 — Watchdog Control Bit
This bit cannot be set via software. The watchdog timer can only be disabled by reset.
1 = Watchdog timer disabled
0 = Watchdog timer enabled

MR3 — Prescaler 1 Clear Bit
Presets the contents of prescaler 1 to $7F.
1 = Prescaler 1 preset
0 = Prescaler 1 not preset

MR2 — Prescaler 2 Clear Bit
Presets the contents of prescaler 2 to $7FFF.
1 = Prescaler 2 preset
0 = Prescaler 2 not preset

MR1 — Prescaler Cross-Couple Bit
This bit controls the output of prescalers 1 and 2 and directs them to either timer A or B clock inputs.
1 = Prescaler 1 feeds timer B clock input, and prescaler 2 feeds timer A input
0 = Prescaler 1 output is used as clock input for timer A, and prescaler 2 output is used as clock input for timer B

MR0 — Port B Toggle Cross-Couple Bit
This bit controls the overflow pulses of timers A and B and directs them to either port B0 or B1.
1 = Timer A overflow output is directed to port B0, and timer B or timer C (depending on the status of MOR5) output is directed to port B1
0 = Overflow output pulse of timer A is used as a port B1 data register toggle clock source, and timer B or timer C overflow output pulse is directed to port B0 toggle clock input

RESETS

The MCU can be reset four ways: (1) by initial power-up; (2) by the external reset input ($\overline{\text{RESET}}$); (3) by a forced reset generated by the "watchdog" counter; and (4) by an optional internal low voltage detect circuit. The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger that senses the line logic level. Figure 6 shows the MCU reset circuit.

POWER-ON-RESET (POR)

An internal reset is generated on power-up that allows the internal clock generator to stabilize. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drop in the power supply voltage. A delay of t_{RHL} milliseconds is required before allowing $\overline{\text{RESET}}$ input to go high. Connecting a capacitor to the $\overline{\text{RESET}}$ input (Figure 7) typically provides sufficient delay.

EXTERNAL RESET INPUT

The MCU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a period longer than one machine cycle (t_{cyc}). Under this type of reset, the Schmitt trigger switches off at V_{IRES-} to provide an internal reset voltage.

FORCED RESET

If the auxiliary counter reset mask bit in the miscellaneous counter (MR4) is cleared and the auxiliary counter

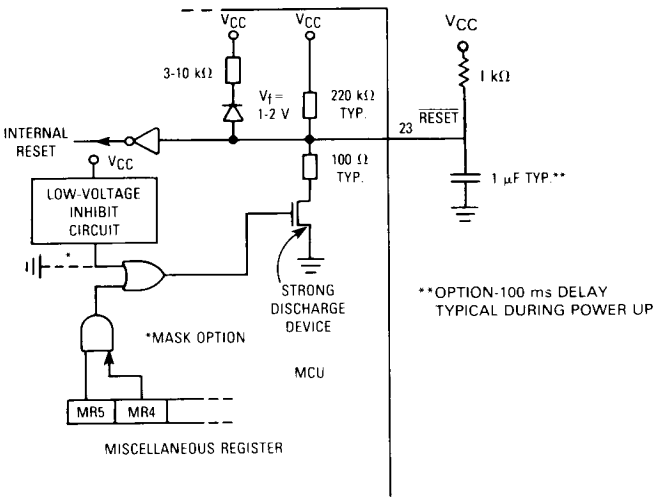


Figure 6. MCU Reset Circuit

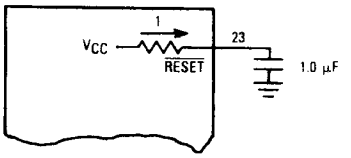


Figure 7. Power-Up Reset Delay Circuit

status bit (MR5) is set, as a result of counter overflow, a switch to V_{SS} is turned on pulling the $\overline{RESET}$ pin low. A consequent voltage drop below V_{IRES-} on $\overline{RESET}$ causes a reset, which in turn sets MR4. Switching to V_{SS} when the $\overline{RESET}$ pin is turned off allows voltage to rise above V_{IRES+}, after which the reset is released. $\overline{RESET}$ pin voltage variation occurring as a result of forced reset may be amplified externally in order to provide a reset to other peripheral circuits in the system. The reset output from the MCU is not TTL compatible.

3

LOW-VOLTAGE INHIBIT (LVI)

The optional low-voltage detection circuit causes a reset of the MCU if the power supply voltage falls below a certain level (V_{LVI}). The only requirement is that the V_{CC} must remain at or below the V_{LVI} threshold for one t_{cyc} minimum.

In typical applications, the V_{CC} bus filter capacitor will eliminate negative-going voltage glitches of less than one t_{cyc}. The output from the low-voltage detector is connected directly to the internal reset circuitry. It also forces the $\overline{RESET}$ pin low via a strong discharge device through a resistor. The internal reset is removed once the power supply voltage rises above a recovery level (V_{LVR}) at which time a normal power-on reset occurs.

INTERRUPTS

The MCU can be interrupted eight different ways: through the external interrupt $\overline{INT1}$ input pin, with the internal timer (either A, B, or C) interrupt request, using the software interrupt instruction (SWI), SPI interrupt request, external port D bit 6 ($\overline{INT2}$) input pin, or at reset.

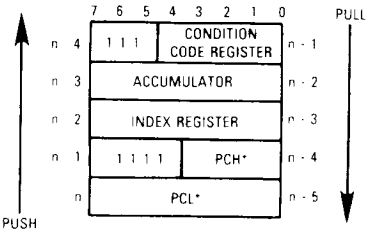
Interrupts cause the processor registers to be saved on the stack and the interrupt mask (I bit) set to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack after which normal processing resumes. The stacking order is shown in Figure 8.

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 considered to be the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts and, if unmasked



*For subroutine calls, only PCH and PCL are stacked.

Figure 8. Interrupt Stacking Order

(I bit clear), proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Masked interrupts are latched for later interrupt service. If the timer interrupt status bit is cleared before unmasking the interrupt, then the interrupt is not latched.

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 setting of the I bit. Refer to Figure 9 for the reset and interrupt instruction processing sequence.

TIMER INTERRUPT

Each interrupt, except $\overline{INT1}$, has a separate mask bit which must also be cleared, in addition to the I bit, for the MCU to acknowledge the interrupt. The $\overline{INT2}$, timer A, timer B, timer C, and SPI interrupts each have their own independent mask bits contained in MR6, TACR6, TBCR6, TCOM, TCCM, and SPICR6. The interrupt routine must determine the source of the interrupt by examining the interrupt request bits, TACR7, TBCR7, MR7, TCOF, TCCF, and SPICR7. These bits must be cleared by software. The $\overline{INT1}$ interrupt has its own vector address. Therefore, the $\overline{INT1}$ interrupt request is cleared automatically, and then the $\overline{INT1}$ vector is serviced.

EXTERNAL INTERRUPT

The external interrupt is internally synchronized and then latched on the falling edge of $\overline{INT1}$ and $\overline{INT2}$. Clearing the I bit enables the external interrupt. The $\overline{INT2}$ interrupt has an interrupt request bit (bit 7) and a mask bit (bit 6) in the miscellaneous register (MR). The $\overline{INT2}$ interrupt is inhibited when the mask bit is set. The $\overline{INT2}$ is always read as a digital input on port D. The $\overline{INT2}$ and timer interrupt request bits, if set, cause the MCU to process an interrupt when the condition code I bit is clear. The following paragraphs describe two typical external interrupt circuits.

Zero-Crossing Interrupt

A sinusoidal input signal (f_{INT1} maximum) can be used to generate an external interrupt (see Figure 10a) for use as a zero-crossing detector (for negative transitions of the ac sinusoid). This type of circuit allows applications

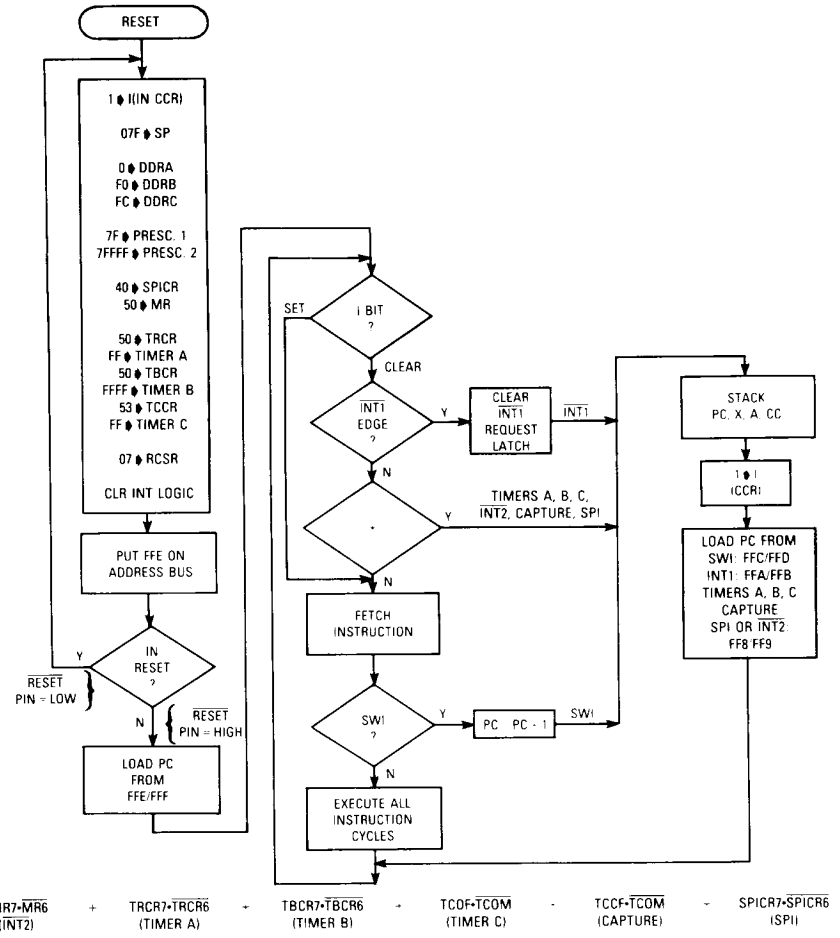


Figure 9. Reset and Interrupt Processing Flowchart

such as servicing time-of-day routines and engaging/disengaging ac power control devices. Off-chip, full-wave rectification provides an interrupt at every zero crossing of the ac signal and, thereby, provides a 2f clock.

Digital-Signal Interrupt

With this type of circuit (Figure 10b), the $\overline{INT1}$ pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{INT1}$ pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. Refer to **TIMER** for additional information.

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 execution is similar to the hardware interrupts.

TIMERS

The MCU has four timers and two programmable prescalers. The timers are identified as timer A, B, C, and the auxiliary counter. Refer to Figure 11 for timers A, B, and C block diagram. The following paragraphs described the different timers.

TIMER A

Timer A is an 8-bit programmable down counter, which can be loaded under program control. Timer A also

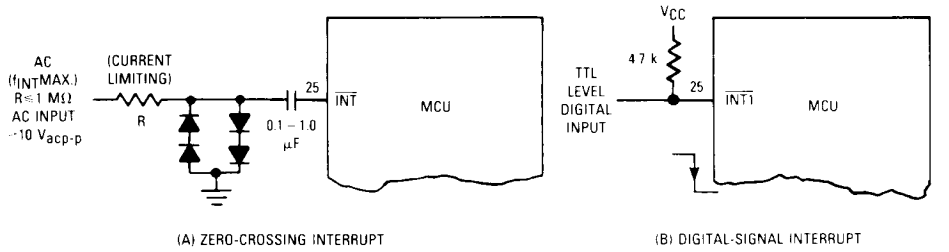


Figure 10. External Interrupt

includes a modulus latch which allows the timer to be "auto-reloaded." As clock inputs are received, timer A decrements toward $00. When $00 is reached, bit 7 in the timer A control register is set and the timer is reloaded with the contents of the modulus latch. An underflow condition is also generated when value $00 is reached. This state can be used to toggle bit 0 or bit 1 of port B directly under the control of the miscellaneous register (MR0), the SPI control register, and the port B data direction register. Setting TACR6 or the I bit in the condition control register will prevent timer interrupts from being processed. The timer interrupt request bit *MUST* be cleared by software. There are three ways of loading data from the modulus latch into timer A as described in the following paragrahs.

Direct Loading

When the MCU writes to timer A data register, the data is latched by the modulus latch, and forced into the timer. This operation requires that TACR3 be cleared.

Asynchronous External Event Loading

When TACR3 is a logic one, the contents of the modulus latch are transferred to the timer at the rising edge of $\overline{INT2}$ interrupt request bit (MR7) gated with interrupt request mask bit (MR6). If this loading is used, care must be taken in programming as it will start an interrupt service routine if the I bit in the CCR is clear. Loading $00 to timer A allows a countdown of 256 clocks before the next $00 state is reached.

Auto-Loading

The modulus latch is automatically loaded when the timer reaches $00. This loading is dependent on the setting of TACR3. Auto-loading also occurs in both the previous loading modes. Timer A can be read at any time without affecting the countdown of the timer. The timer and modulus latch are set to $FF on reset.

NOTE

Loading $01 to timer A should be avoided when operating with a divide-by-one prescaler. Doing so will inhibit timer A auto-loading, interrupt generation, and port B toggle mechanisms.

TIMER A CONTROL REGISTER $09

7	6	5	4	3	2	1	0
TACR7	TACR6	TACR5	TACR4	TACR3	TACR2	TACR1	TACR0
RESET:							
0	1	0	0	0	0	0	0

TACR7 — Timer A Interrupt Request Flag
1 = Timer A has transition to $00
0 = Software or reset cleared
TACR6 — Timer A Interrupt Request Mask
1 = Interrupt request inhibited
0 = Interrupt request not inhibited
TACR5 — External or Internal Bit
1 = External clock source for prescaler 1
0 = Internal clock source for prescaler 1
TACR4 — External Enable Bit
Control bit used to enable the external timer pin (PRESCALER1 PC0).

TACR5	TACR4	Prescaler 1 Clock Source
0	0	Internal Clock
0	1	AND of Internal Clock and PRESCALER1 PC0*
1	0	Inputs Disabled
1	1	PRESCALER1 PC0* Low-to-High Transition

*The status of PRESCALER1 PC0 depends upon the data direction status of PRESCALER1 PC0. If PRESCALER1 PC0 is an output, then the clock source is equal to the port data register content, independent of the port electrical loading. If an input, then the clock source is the logic level of PRESCALER1 PC0.

TACR3 — Timer A Load Mode Control
1 = Asynchronous external event loading ($\overline{INT2}$ driven loading is enabled)
0 = Allows direct loading of timer A
TACR2, TACR1, TACR0 — Prescaler 1 Division Ratio Control Bits
When set, these bits select one of eight possible outputs on prescaler 1.

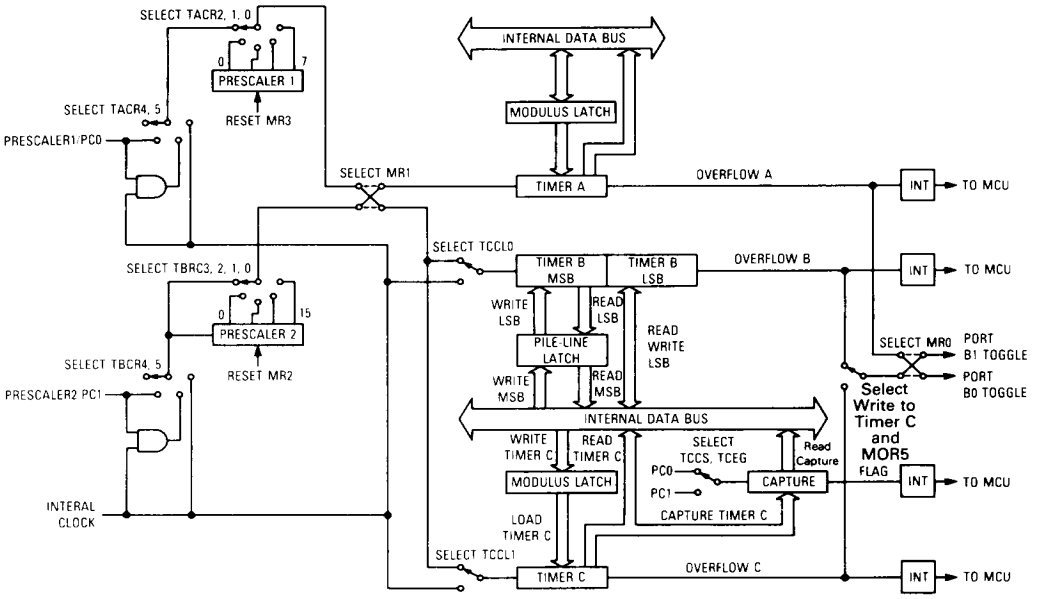


Figure 11. Timers A, B, and C Block Diagram

TACR2	TACR1	TACR0	Divide By
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

TIMER B

This is a 16-bit timer which is accessed via two registers ($0B for the most-significant byte (MSB) and $0C for the least-significant byte (LSB)). The MSB has a "pipeline" latch that allows a "snap shot" value of the entire 16 bits to be read. Read/write operations to the LSB are direct. The LSB can be read at anytime without disturbing the count. When the LSB is read, the contents of the MSB are loaded into the pipeline latch so a read of the MSB is actually the contents of the latch.

When writing to the LSB, the contents are immediately entered into the timer. At the same time the pipeline contents are forced into the MSB of the timer. This allows a 16-bit word to be placed into the timer data register during a LSB write operation. An underflow condition is also generated when value $00 is reached. This state can be used to toggle bit 0 or bit 1 of port B directly under the control of the miscellaneous register (MR0), the SPI control register, and the port B data direction register. Setting TBCR6 or the I bit in the condition control register will prevent timer interrupts from being processed. The timer interrupt request bit *MUST* be cleared by software.

TIMER B CONTROL AND STATUS REGISTER $0D

7	6	5	4	3	2	1	0
TBCR7	TBCR6	TBCR5	TBCR4	TBCR3	TBCR2	TBCR1	TBCR0
RESET:							
0	1	0	0	0	0	0	0

TBCR7 — Timer B Interrupt Request Flag
1 = Timer B has transition to $00
0 = Software or reset cleared

TBCR6 — Timer B Interrupt Request Mask
1 = Interrupt request inhibited
0 = Interrupt request not inhibited

TBCR5 — External or Internal Bit
1 = External clock source for prescaler 2
0 = Internal clock source for prescaler 2

TBCR4 — External Enable Bit
Control bit used to enable the external timer pin (PRESCALER2/PC1).

TBCR5	TBCR4	Prescaler 2 Clock Source
0	0	Internal Clock
0	1	AND of Internal Clock and PRESCALER2/PC1*
1	0	Inputs Disabled
1	1	PRESCALER2/PC1* Low-to-High Transition

*The status of PRESCALER2/PC1 depends upon the data direction status of PRESCALER2/PC1. If PRESCALER2/PC1 is an output, then the clock source is equal to the port data register content, independent of the port electrical loading. If an input, then the clock source is the logic level of PRESCALER2 PC1.

TBCR3, TBCR2, TBCR1, TBCR0 — Prescaler 2 Division Ratio Control Bits
When set, these bits select one of eight possible output on prescaler 2.

TBCR3	TBCR2	TBCR1	TBCR0	Divide By
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	16
0	1	0	1	32
0	1	1	0	64
0	1	1	1	128
1	0	0	0	256
1	0	0	1	512
1	0	1	0	1024
1	0	1	1	2048
1	1	0	0	4096
1	1	0	1	8192
1	1	1	0	16384
1	1	1	1	32768

TIMER C

Timer C is an 8-bit programmable down counter. The timer contains a modulus latch which allows the timer to be auto reloaded. The timer auto reloads with the contents of the modulus latch upon every $01 to $00 transition. Timer C contains a capture register. This read-only register and the contents are refreshed by the contents of the data register during the capture instance. The timer can be written to at any time, and the contents of both the data register and modulus latch are updated immediately. The timer is set to $FF on reset, but the contents of the capture register are not valid until the first capture after reset.

TIMER C CONTROL REGISTER $015

7	6	5	4	3	2	1	0
TCOF	TCOM	TCCF	TCCM	TCEG	TCCS	TCCL1	TCCL0
RESET:							
0	1	0	0	0	0	0	0

TCOF — Timer C Overflow Flag
1 = Timer C has transition to $00
0 = Software or reset cleared

TCOM — Timer C Interrupt Mask
1 = Interrupt request inhibited
0 = Interrupt request not inhibited

TCCF — Timer C Capture Flag
1 = Proper capture occurred on PRESCALER1 or PRESCALER2. No new capture occurs when set
0 = Software or reset cleared

TCCM — Timer C Capture Interrupt Request Mask
- 1 = Inhibits interrupt request generated from TCCF
- 0 = Does not inhibit interrupt request generated from TCCF

TCEG — Timer C Capture Edge Select
- 1 = Selects rising edge of PC0 or PC1 to be capture instance
- 0 = Selects falling edge of PC0 or PC1 to be capture instance

TCCS — Timer C Capture Source Select
- 1 = Select PRESCALER2/PC1 as capture source
- 0 = Select PRESCALER1/PC0 as capture source

TCCL1 and TCCL0 — Timer C Clock Source Select
Clock source selection is defined below.

TCCL1	Timer C Source
0	Internal Clock
0	Internal Clock
1	MR1 Status*
1	MR1 Status*

TCCL0	Timer B Source
0	Internal Clock
1	MR1 Status*
0	Internal Clock
1	MR1 Status*

NOTES:
1. *Denotes prescaler 1 or 2 clock source depending on miscellaneous register bit 1 (MR1) status.
2. MR1 bit cleared (logic zero) at reset:
   Prescaler 1 clock selected to timer A
   Prescaler 2 clock selected to timer B and C
3. MR1 bit set (logic one):
   Prescaler 1 clock selected to timer B and C
   Prescaler 2 clock selected to timer A
4. Prescaler 1 output determined by the status of Timer A control register bits 2, 1, and 0 (TACR2, TACR1, and TACR0).
5. Prescaler 2 output determined by the status of Timer B control register bits 3, 2, 1, and 0 (TBCR3, TBCR2, TBCR1, and TBCR0).

PRESCALER 1

Prescaler 1 is a 7-bit binary down counter whose value is selected by TACR2, TACR1, and TACR0. The selected output is used as the clock input to either timer A or B, depending upon the status of the prescaler cross-couple bit (MR1). The type of clock source to prescaler 1 may be selected by TACR5 and TACR4. Prescaler 1 is set to $7F at reset or under program control when a one is written to prescaler 1 clear bit (MR3).

PRESCALER 2

Prescaler 2 is a 15-bit down counter; its value is selected by TBCR3, TBCR2, TBCR1, and TBCR0. The selected output is used as the clock input to either timer A or B, depending upon the status of the prescaler cross-couple bit (MR1). The type of clock source to prescaler 2 may be selected by TBCR5 and TBCR4. Prescaler 2 is set to $7FFF at reset or under program control when a one is written to prescaler 2 clear bit (MR2).

AUXILIARY COUNTER

This register is a fixed counter which is clocked by the internal clock (f_{OSC} divided by four). Total count period is 4095 cycles. The MCU communicates with this counter via the miscellaneous register (MR5 and MR4). Countdown may be aborted at any time under program control, which also resets the counter to 4095 and clears MR5. When MR4 is clear and MR5 is set as a result of counter time out, the reset pin is internally pulled to ground. If the MCU loses control of the program, the "watchdog" timer will bring the MCU back to reset. Refer to Figure 12 for counter operation diagram.

EPROM PROGRAMMING

ERASING THE EPROM

The EPROM can be erased by exposure to high-intensity ultraviolet (UV) light with a wavelength of 2537A. The recommended integrated dose (UV intensity × exposure time) is 25Ws/cm^2. The lamps should be used without software filters, and the MCU should be positioned about one inch from the UV tubes. Ultraviolet erasure clears all bits of the EPROM to the "zero" state. Data can then be entered by progamming "ones" into the desired bit locations.

CAUTION

Be sure that the EPROM window is shielded from light except when erasing. This protects both the EPROM and light-sensitive nodes.

MASK OPTION REGISTER (MOR) $F1E

The MOR is implemented in EPROM and contains all zeros prior to programming. The MOR bits are described in the following paragraphs. This register is not affected by reset.

7	6	5	4	3	2	1	0
CLK	TOPT	PBTS	LVI	*	*	*	SEC

CLK — Clock (oscillator type)
- 1 = Resistor Capacitor (RC)
- 0 = Crystal

TOPT — Timer Option
- 1 = Enables timer C
- 0 = Disables timer C

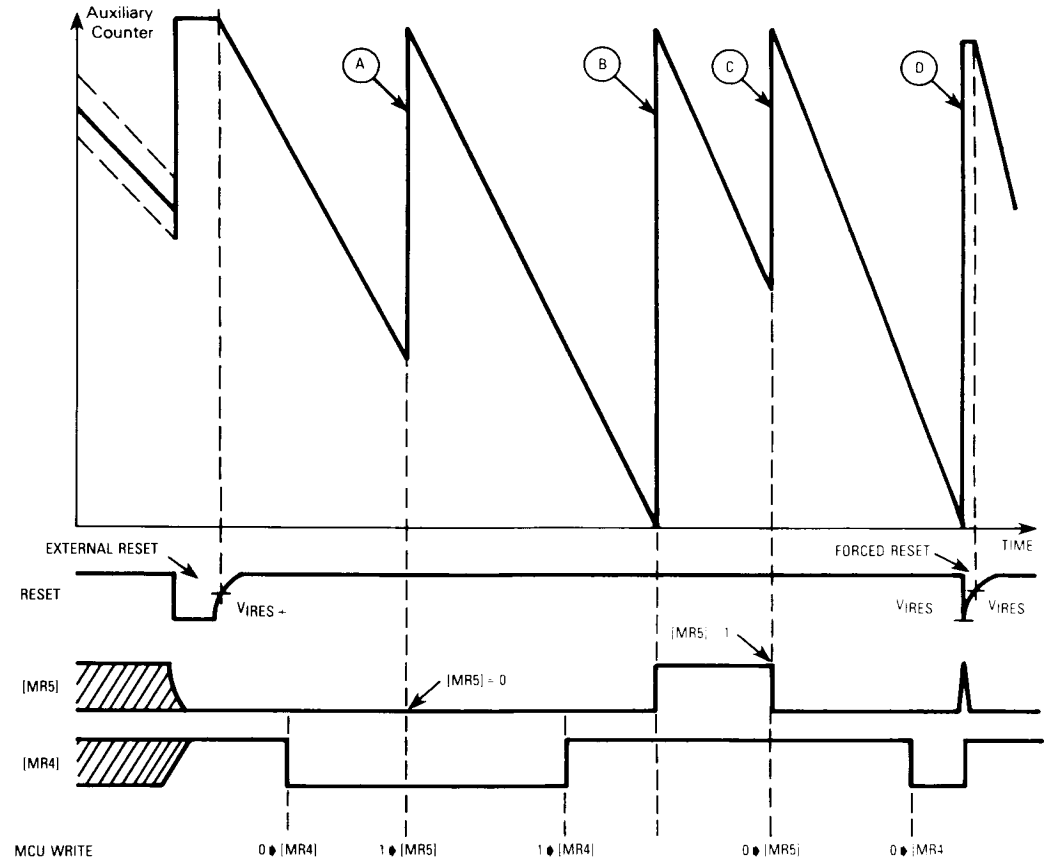
PBTS — Port B Toggle Source
This bit is not used on the TJ6 mask set. When cleared the operation is the same as the TJ6 mask set operation.
- 1 = Port B toggle source will come from the timer B overflow even if a write operation is performed on timer C
- 0 = After the first write operation to timer C, the toggle source coming from the timer B overflow is replaced by the timer C overflow. If no write operation is performed on timer C, then timer B is the port B toggle source.

LVI — Low Voltage Inhibit
- 1 = Enables low-voltage detection circuitry
- 0 = Disables low-voltage detection circuitry

Bits 1-3
User available register bits during normal mode of operation

SEC — Security
For full security, this bit must be set in the MOR and mirror MOR ($F16).
- 1 = Enables EPROM read protection
- 0 = Disables EPROM read protection

3



(A) Counter Preset by Writing "1"

(B) Underflow: MR5 ◗ 1; No Forced Reset

(C) Counter Reset by Writing "0"

(D) Underflow MR5 ◗ 1 Forced Reset

Figure 12. Auxiliary Counter Operation

PROGRAMMING CONTROL REGISTER (PCR) $012

The PCR is an 8-bit register which provides the necessary control bits to program the EPROM. The bootstrap program manipulates the PCR when programming so the user need not be concerned with PCR in most applications.

7	6	5	4	3	2	1	0
1	1	0	1	1	$\overline{VPON}$	$\overline{PGE}$	$\overline{PLE}$
RESET:							
U	U	U	U	U	1	1	1

$\overline{PLE}$ — Programming Latch Enable
Controls address and data being latched into the EPROM. Set during reset, but may be cleared anytime.
1 = Read EPROM
0 = Latch address and data on EPROM

$\overline{PGE}$ — Program Enable
Enables programming of EPROM. Must be set when changing the address and data. Set during reset.
1 = Inhibit EPROM programming
0 = Enable EPROM programming (if $\overline{PLE}$ is low)

$\overline{VPON}$ — V_{PP} On
A read-only bit that indicates high voltage at the V_{PP} pin. When set to "one", disconnects $\overline{PGE}$ and $\overline{PLE}$ from the chip.
1 = No high voltage of V_{PP} pin
0 = High voltage on V_{PP} pin

NOTE

$\overline{VPON}$ being "zero" does not indicate that the V_{PP} level is correct for programming. It is used as a safety interlock for the user in the normal operating mode.

$\overline{VPON}$	$\overline{PGE}$	$\overline{PLE}$	Programming Conditions
0	0	0	Programming enabled (program EPROM byte)
1	0	0	$\overline{PGE}$ and $\overline{PLE}$ bits disabled
0	1	1	Programming disabled (latch address and data in EPROM)
1	1	0	$\overline{PGE}$ and $\overline{PLE}$ disabled
0	0	0	Invalid state
1	0	1	Invalid state
0	1	1	Voltage applied to $\overline{RESET}$ V_{PP} pin
1	1	1	$\overline{PGE}$ and $\overline{PLE}$ disabled (operating mode)

PROGRAMMING

The MCU bootstrap program can be used to program the EPROM. The vectors at address $FF6 and $FF7 are used to start the program. This vector is fetched when V_{IHTP} is applied to the PRESCALER PC0 pin and the $\overline{RESET}$ pin is allowed to rise above V_{IRES+}. The level on the PRESCALER PC1 pin, when the $\overline{RESET}$ V_{PP} pin rises above V_{IRES+}, determines which programming mode is selected. A high level on PRESCALER PC1 selects the auto-programming operation.

A M2532/2732 UV EPROM must first be programmed with the same information that is to be transferred to the MCU EPROM. Unprogrammed EPROM address locations should contain $00 to speed up the programming operation. Figure 13 is a schematic diagram for a board and circuitry that can be used to program the MCU EPROM.

Perform the following steps to program the MCU EPROM:

1. Insert the programmed EPROM and erased MCU EPROM into U2 and U3.
2. Programming operation starts when S1 is placed to the ON position.
   a) DS1 and DS2 illuminate.
   b) MCU control is transferred to the bootstrap ROM, and the programming routine executed by the bootstrap loader program.
   c) DS3 blinks during programming. When programming is complete, DS3 remains illuminated.
   d) After two seconds DS4 will illuminate indicating the MCU has been programmed and verified.
3. Remove power by placing S1 to the OFF position and remove programmed MCU.

NOTE

No programming can be done once the MOR and mirror MOR security bit has been programmed to logic one. The only way to proceed from the secure mode to the non-secure mode is by erasing the MCU. The MCU must be reset following programming of the SEC bits to enable the security feature.

EMULATION

The MCU is designed to emulate the functions of either the MC6805S2 or MC6805S3. However, due to pin assignments, processing, and mask options, the MCU has some differences. The differences are listed as follows:

1. Port A output on the MC6805S2/S3 is a mask option. The CMOS pullup option on port A is not implemented on MC68705S3. If this option is required, pullup resistors must be installed.
2. The RC clock on the MC6805S2/S3 is a mask option. To enable the MC68705S3 RC clock, MOR bit 7 must be programmed to a logical one.
3. The LVI on the MC6805S2/S3 is a mask option. To enable the LVI on MC68705S3, MOR bit 4 must be programmed to a logical one.
4. The MC68705S3 $\overline{RESET}$/V_{PP} and V_{STBY}/AN4/$\overline{INT2}$/PD6 electrical characteristics are different for the MC6805S2 S3.
5. Pin 4 (AN4 and V_{STBY}) on MC6805S2 S3 is a mask option. On the MC68705S3, pin 4 is enabled for V_{STBY} AN4 $\overline{INT2}$ PD6.
6. On MC6805S2/S3 pin 4, standby RAM contents will be lost if the voltage drops below 3.0 V. Standby RAM on the MC68705S3 will not be lost unless voltage drops below 4.0 V.
7. Above certain voltages (3.7 V typical), pin 4 will exhibit lower input impedance than the MC6805S2 S3. This may cause A/D conversion inaccuracies if the

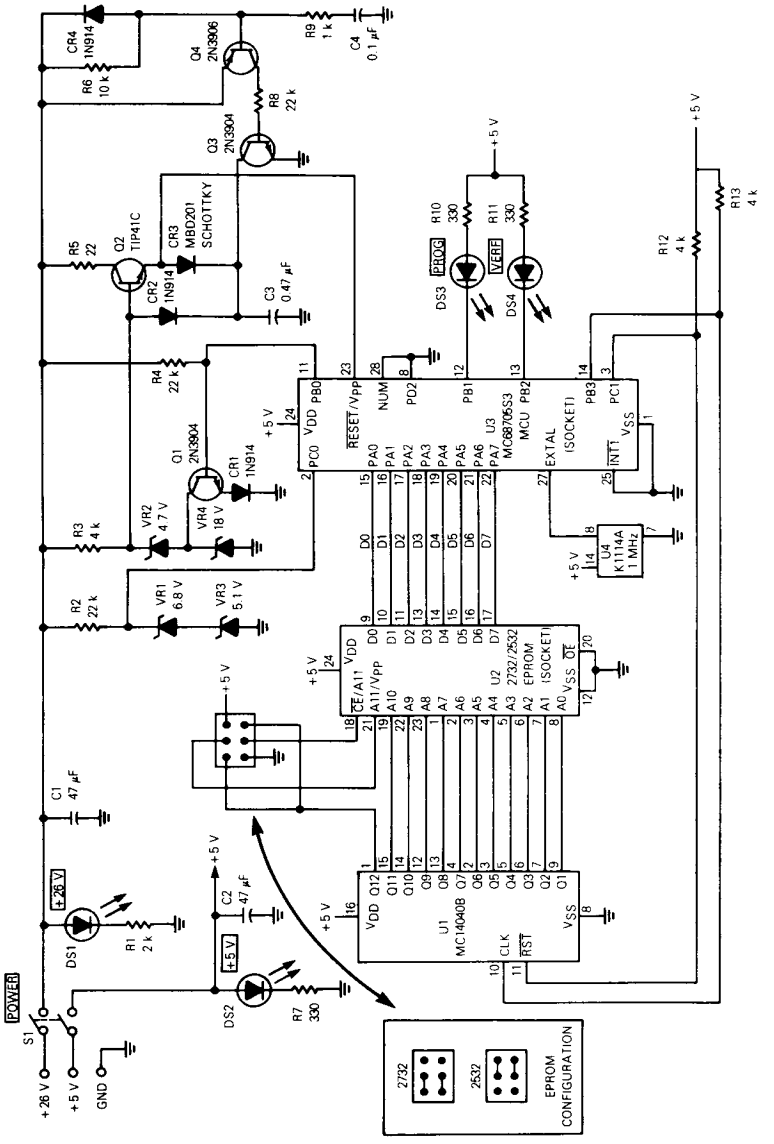


Figure 13. Programming Connections Schematics Diagram

pin is used as fifth A/D input channel. Pin 4 is always a high impedance input on the MC6805S2/S3.

8. Reset and Vpp functions share a common pin (23) on the MC68705S3. Therefore, electrical characteristics on this pin may vary from the same pin on MC6805S2/S3. The input impedance on the MC68705S3 pin is approximately equivalent to the 1.0 ohm pulldown resistor; whereas, on the MC6805S2/S3, this pin is a high impedance (220K ohms) input. Therefore, the MC68705S3 requires a pullup resistor on the $\overline{\text{RESET}}$ pin to recover from a reset condition.

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) has arbitration on the data and clock lines. The SPI communicates with the MCU via data and control registers. The SPI data and clock inputs are always taken from their respective I/O ports, regardless of the status of the data direction registers relative to that port. The SPI can operate in modes from auto clocked (NRZ), half duplex, and full duplex with from a one to a four wire combination. Refer to Figure 14 for the SPI block diagram.

SPI CONTROL AND STATUS REGISTER

This 8-bit register contains the status and control bits relative to SPI operations. The SPI control register operation is shown in Figure 15. The SPI control and status register bits can be set or cleared under program control.

7	6	5	4	3	2	1	0
SPICR7	SPICR6	SPICR5	SPICR4	SPICR3	SPICR2	SPICR1	SPICR0
RESET:							
0	1	0	0	0	0	0	0

SPICR7 — SPI Interrupt Request Bit
Set on eighth data input strobe. MCU services this interrupt if I bit is clear in CCR.
1 = Interrupt request (if SPICR6 not masked)
0 = No interrupt pending

SPICR6 — SPI Interrupt Request Mask Bit
1 = Disables interrupt request from SPICR7
0 = Enables interrupt request from SPICR7

SPICR5 — SPI Clock Sense Bit/Bus-Busy Flag
Dual-function bit controlled by the status of SPICR4.
1 = Start SPI operation when SPICR4 = 1. Input data latched on positive edge and output data changed on negative edge of SPI clock when SPICR4 = 0.
0 = Stop SPI operation when SPICR4 = 1. Input data latched on negative edge and output data changed on positive edge of SPI clock when SPICR4 = 0.

SPICR4 — SPI Operation Enable Bit
This bit determines the functions of SPICR5 and SPICR2.
1 = Enables SPI data register shifting, data and clock arbitration logic, and slave select input logic
0 = Disables SPI data register shifting, data and clock arbitration logic, and slave select input logic

SPICR3 — SPI Data Output Select Bit
1 = Output of the SPI data register is loaded to port B3 data register at the appropriate SPI clock edge selected by SPICR5, during the active transaction mode
0 = Output of the SPI data register is loaded to port B2 data register at the appropriate SPI clock edge selected by SPICR5, during the active transaction mode

SPICR2 — Port B1 Toggle Enable/Start Bit
Dual-function bit controlled by the status of SPICR4.
1 = Start bit is set by negative transition of the data input of the SPI data shift register while the clock is at the idle level when SPICR4 = 1. Start bit set under program control to enable port B1 data register toggle facility when SPICR4 = 0.
0 = Stop SPI operation when SPICR4 = 1. Cleared under program control when SPICR4 = 0.

SPICR1 — Mode Fault Flag
1 = (a) Mode flag is set when SPI data output arbitration occurs on the SPI data output port (PB3 or PB2) selected by SPICR3. The MCU loses data mastership, and the SPI data output port DDR is cleared.
(b) Mode flag is set if a low level is detected on slave input PB0. Then, the MCU loses clock mastership switching to the clock slave mode, and port B1 DDR is cleared.
(c) Mode flag is set during the idle mode when a negative clock edge is detected on the SPI clock input, and the port B1 data register is cleared.
0 = Cleared under program control

SPICR0 — SPI Input Data Select Bit
1 = SPI data from port B3 is latched into the SPI data register
0 = SPI data from port B2 is routed to the input of the SPI data register

SPI DATA REGISTER

This register can be written to any time and can also be read, regardless of serial operations, without disturbing the data. A one bit shift to the left occurs each time there is a data input strobe while the LSB is loaded with data from port B2 or B3. The MSB is loaded every time there is data output strobe. Data input and output strobes are generated internally only during the active transaction time.

SPI DIVIDE-BY-EIGHT COUNTER

The counter is cleared during SPI deselect or idle modes. A count occurs at every data input strobe during the active transaction mode. At overflow, SPICR7 is set which puts the SPI in idle mode and blocks all data input and output strobes. The counter is cleared when PB0 is high if the SPI is in the slave mode or when a "start" condition is detected.

SPI OPERATION

The SPI can operate in a variety of modes. Software assisted protocols may be defined to upgrade the hardware versatility and/or system performance of the MCU. Some features common to all operating modes are summarized in Table 1 and in the following paragraphs.

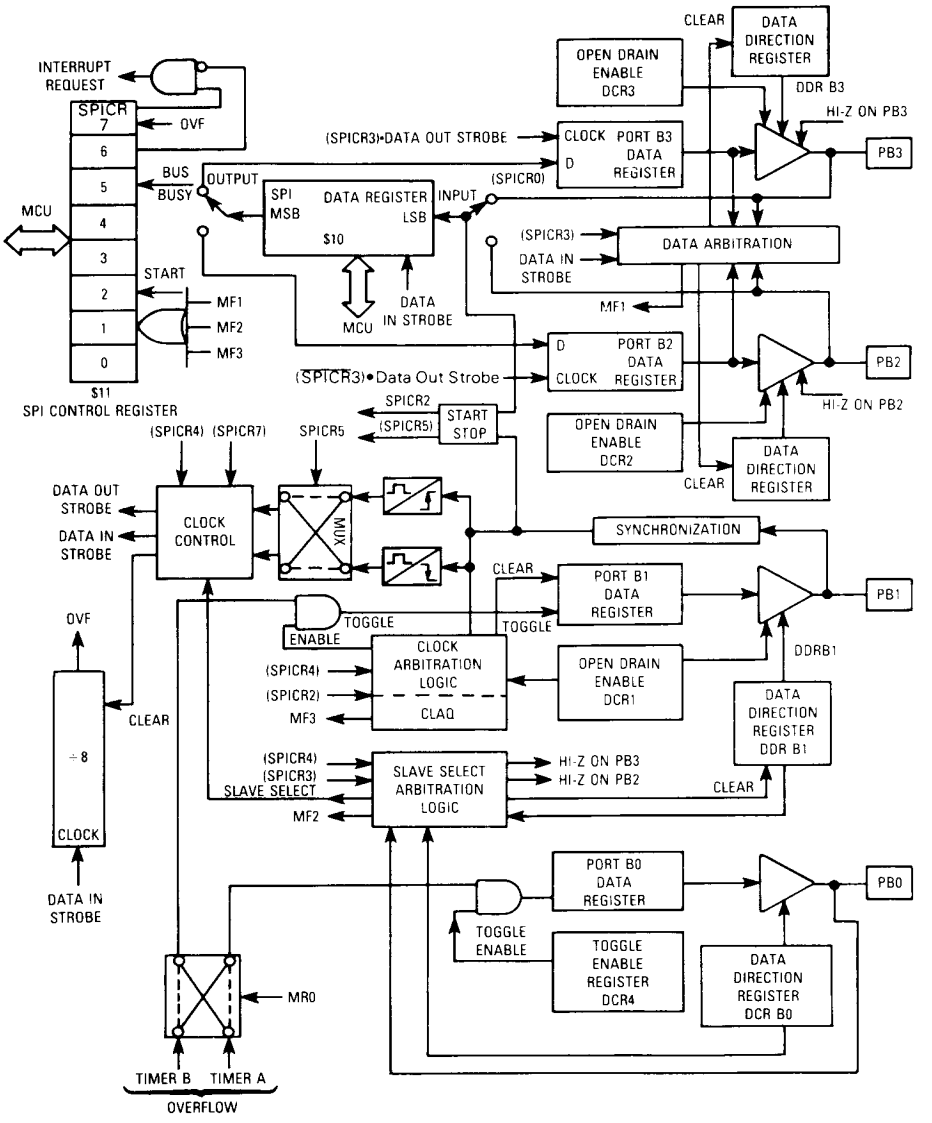


Figure 14. Serial Peripheral Interface Block Diagram

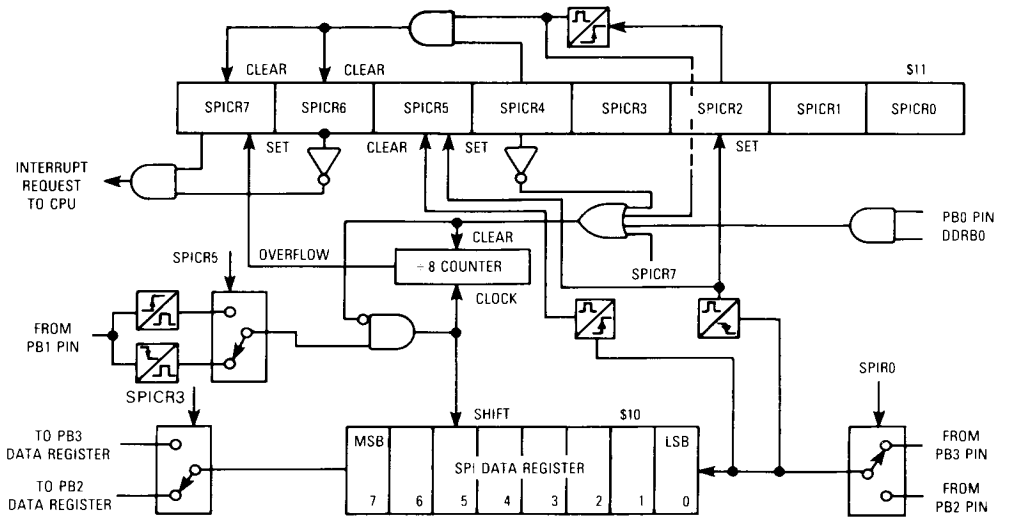


Figure 15. SPI Control Register Operation

1) SPI data input and output may be individually routed to or from PB2 or PB3 (Table 2). These four routings provide half and full duplex operations, as well as allowing bidirectional information to flow in daisy-chained systems.
2) When data input and output is done on PB2, PB3 is available for any other use and vice versa.
3) Data input is always relative to the port pin logic level regardless of the data direction register status on that pin.
4) In full duplex operation, 16 bits of information may be transferred with eight clock pulses between at least two devices with transmit capability. Both PB2 and PB3 are used for data transfer. The same shift register is used for data in and data out. The byte transmitted replaces the byte received. SPICR7 is used to signify that the I O operation is complete.
5) SPI clock is always provided on port B1. In the clock slave mode, port B1 DDR is in the input mode (cleared). In the clock master mode, port B1 DDR is set; therefore, the MCU imposes the clock level on PB1 until there is clock arbitration on the clock line or until the MCU loses clock mastership when PB0 goes low.
6) No fixed baud rate generation exists. The clock frequency is dependent on the prescaleer clock source option, prescaler divide ratio, and timer divide ratio as well as the port C status in case of external clocking for the timer. Toggling of the port B1 data register is automatically allowed during the active transmission mode.
7) All devices connected to the SPI must have their output and input data strobe on the same clock edge for correct transfer of data.
8) During the active transmission mode, the first clock edge must be the output data strobe. When this occurs, the MSBs of the data registers of all transmitters are copied onto the data output pins, and the MCU copies the MSB of its SPI onto the port B2 or B3 data register.
9) Port B data direction registers and port B data control registers are accessible during SPI operation. During active transaction mode, the PB1 data register, PB2 data register (if SPICR3 = 0), and PB3 (if SPICR3 = 1) are not write accessible under program control.
10) Port B lines not used for SPI can be used for other digital functions.

SELECT INPUT OPERATION

An external device supplies slave select information via port B0. If slave select is not used, set port B0 to output mode to inhibit slave select function.

The following paragraphs describe clock master and clock slave operating modes of the SPI.

Master Mode Slave Select Actions

The MCU monitors slave select input in master mode to assure that it stays false. If slave select goes true, the MCU exits master mode and becomes a slave. This implies that a write collision has occurred which means two

Table 1. Summary of SPI Operations

DEFINITIONS
- Transmitter — Data Master: DDRB2 or 3 = 1
- Receiver — Data Slave: DDRB2 or 3 = 0
- Clock Master: DDRB1 = 1
- Clock Slave: DDRB1 = 0
- Transaction Mode: SPICR4 = 1
  1) Active: $\overline{SPICR7}\bullet(\overline{DDRB0}\bullet\overline{PB0} + DDRB0)$ if DDRB1 = 0 (clock slave mode) or
     $\overline{SPICR7}\bullet(\overline{DDRB0}\bullet\overline{PB0} + DDRB0)$ if DDRB1 = 1 (clock master mode)
     Clock Pulses allowed, data shifted
  2) Idle: $SPICR7 + \overline{DDRB0}\bullet PB0$ if DDRB1 = 0 (clock slave mode)
     Clock pulses blocked, data output line in high-impedance state
- Deselect Mode: SPICR4 = 0 – No SPI Operations

SLAVE SELECT INPUT
- Slave Select Input: SPISS – PB0
- If DDRB0 = 0 then so SPISS action on MCU
  1) Master Mode: SPISS = 1 DDRB1 = 1
     SPISS 1 – 0: Switch to Slave Mode (DDRB1 1 – 0)
     Set SPICR1 (Mode Fault Flag)
  2) Slave Mode: SPISS = 0 DDRB1 = 0
     External clock is allowed to shift data in out. If SPISS is pulled high, the external clock input pulses are inhibited; no data shift; divide-by-eight counter cleared; SPID (PB2 or PB3) switched to high-impedance state.
- Used as Chip-Select Input

DATA ARBITRATION

Data master loses data mastership when data collision occurs during internal data strobe time.

If SPID output port (PB2 or PB3) = 1 while actual pin level is pulled low externally — conflict detected at internal data strobe time.

Then SPICR1 (mode fault flag) is set; SPID output port DDR (B2 or B3) 1 ♦ 0 (high-impedance state).

CLOCK ARBITRATION

MCU has clock mastership (DDRB1 = 1)
1) Via SPISS line (DDRB0 = 0). If SPISS is pulled low, then clock mastership lost; DDRB1 1 ♦ 0 (high-impedance state); SPICR1 is set (mode fault flag).
2) Via clock line SPICL (DDRB1 = 1 and DCRB5 = 0)
   Condition: SPICL must have open-drain output (DCRB5 = 0)

   If clock line is held low externally then clock mastership is not lost; minimum t_{CLH} and t_{CLK} times are guaranteed.

   If SPICL goes low during idle mode then SPICR1 = 1 and clock line is switched low to inhibit the system clock.

MODE FAULT FLAGE OPERATION (SPICR1)

Flag set when any of the following conditions occur:
- Data arbitration occurs on SPID output.
- Clock arbitration with SPISS during master to slave switching.
- Clock arbitration via clock line if SPICL 1 ♦ 0 during idle.

START, STOP, AND CLOCK IDLE CONDITIONS

Clock Idle: The clock level just prior to the transition that causes data on the serial output data line to be changed is defined as the SPI clock idle state.
SPICR5 = 0: SPICL Idle = Low State
SPICR5 = 1: SPICL Idle = High State

These definitions are necessary for determining start and stop conditions.

NOTE

Clock idle state can only be defined if SPICR4 = 0 (Deselect Mode)

Start Condition: Any negative transition of the data input line (PB2 or PB3) during an SPICL idle state.
Stop Condition: Any positive transition of the data input line during an SPICL idle state.

Table 2. Port B Status During SPI Operation

Port Name	Use	Input	Output	Comments
PB0	SPISS	Yes	No	Used as slave select input
PB0	Data	No	Yes	Used as "busy" signal or any digital output
PB1	SPICL	Yes	No	Clock slave
PB1	SPICL	No	Yes	Clock master
PB2	SPID	Yes	No	SPI data input SPICR0 = 0
PB2	SPID	No	Yes	SPI data output SPICR3 = 0
PB2	Data	Yes	Yes	Any digital signal SPICR3 = 1
PB3	SPID	Yes	No	SPI data input SPICR0 = 1
PB3	SPID	No	Yes	SPI data output SPICR3 = 1
PB3	Data	Yes	Yes	Any digital signal SPICR3 = 0

devices attempted to become masters. Write collisions normally result from a software error, and the default master must clean up the system. The mode fault flag is set to signal that clock mastership is lost. Slave select actions can take place during either active or idle transaction modes.

Slave Select Input Actions During Slave Mode

The current clock master generates slave select to enable one of several slaves to accept or return data. The $\overline{SS}$ signal must go low before serial clock pulses occur and must remain low until after the eighth serial clock cycle. Individual lines or a daisy chain can be used for multiple slaves. When $\overline{SS}$ is high, the following occur:

- Serial data output is forced to a high-impedance state without affecting the DDR status.
- Serial clock input pulses are inhibited from generating internal data output and input strobe pulses.
- The eight-bit counter is cleared.

SPI OPERATING MODES

Six methods of operating the SPI are discussed in the following paragraphs.

One-Wire Autoclocked Mode

Various SPI devices can be connected on a single wire, with data transmission using an implicit clock, and each device being its own clock master.

Two-Wire Half-Duplex Mode

In this mode, separate data and clock lines connect the elements in the system. Data and clock mastership should be monitored via protocol included in the data patterns. A transmitter can send all zeros to take all other transmitters off the bus.

Three-Wire Half-Duplex Mode with Slave Select Input

This mode is the same as the half-duplex mode except that the slave select input allows using the MCU as a peripheral in a system where clock mastership is passed through the slave select line. Typically, the slave select lines can be wired together. The current master sets its slave select line in the output mode prior to a serial transmission and pulls it low to indicate that the system is busy. This allows the clock master to retain mastership until the end of transmission. Software protocol can be arranged so that slaves do not request mastership until their slave select lines go high. At the end of a transmission, the current master pulls $\overline{SPISS}$ high and puts the $\overline{SPISS}$ port (PB0) in the input mode. A slave requesting clock mastership pulls the $\overline{SPISS}$ line low, removing the current master from the line. Time multiplexed protocols may be required to avoid simultaneous mastership requests.

Three-Wire Full-Duplex Mode

This mode allows the MCU to operate simultaneously as transmitter and receiver. Bus or daisy-chain networks are feasible. Protocols in the data stream are required to change:

- Clock masters
- The number of transmitters in the system
- The direction of data flow in daisy-chained systems with collision

It is possible for the MCU to shift out one byte of data while receiving another, as illustrated in Figure 16. This eliminates the need for XMIT EMPTY or REC FULL status bits.

Three-Wire Full-Duplex Mode with Clock Arbitration

This mode is a mix of the three-wire full-duplex mode and two-wire half-duplex mode with clock arbitration, where the SPI clock line operates as a wire-or. Simultaneous masters are allowed, and clock arbitration is via the clock line.

Four-Wire Full-Duplex Mode with Slave-Select Input

This mode is similar to the three-wire full-duplex mode in network construction and to the three-wire half-duplex mode with slave-select input in clock arbitration and slave selection. Refer to Figure 17.

ANALOG-TO-DIGITAL CONVERTER

The chip resident 8-bit analog-to-digital (A/D) converter uses a successive approximation technique as show in

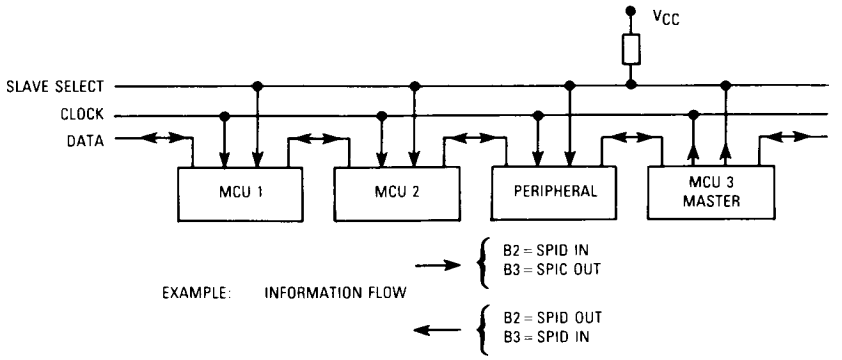


Figure 16. Daisy Chain/Cascade Organization

Figure 18. Four external analog inputs can be connected to the A/D through a multiplexer via port D. Four internal analog channels ($V_{RH} - V_{RL}$, $V_{RH} - V_{RL}/2$, $V_{RH} - V_{RL}/4$, and V_{RL}) may be selected for calibration. The accuracy of these internal channels may not meet the accuracy specifications of the external channels.

A fifth external analog input (AN4) is available via the mask option. When selected, it replaces the V_{RH} internal channel. Due to signal routing, the accuracy of this fifth channel may be slightly less than AN0-AN3.

Multiplexer selection is controlled by the A/D control register (ACR) bits 0, 1, and 2. Refer to Table 3 for multiplexer selection. The ACR is shown in Figure 18. The converter uses 30 machine cycles to complete a conversion of a sampled analog input. When the conversion is complete, the digital value is placed in the A D result register (ARR); the conversion flag is set; selected input is sampled again; and a new conversion begins. When ACR7 is cleared, the conversion in progress is aborted and the selected input, which is held internally, is sampled for five machine cycles.

The converter uses V_{RH} and V_{RL} as reference voltages. An input voltage equal to or greater than V_{RH} converts to $FF. An input voltage equal to or less than V_{RL}, but greater than V_{SS}, converts to $00. Maximum and minimum ratings must not be exceeded. Each analog input source should use V_{RH} as the supply voltage and be referenced to V_{RL} for the ratiometric conversion. To maintain full accuracy of the A D, three requirements should be followed: (1) V_{RH} should be equal to or less

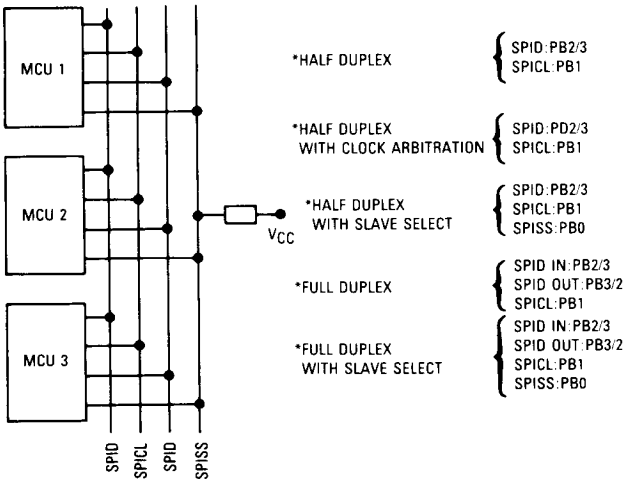


Figure 17. SPI Operation Bus Organization

Table 3. A/D Input MUX Selection

A/D Control Register			Input Selected	A/D Output (Hex)		
ACR2	ACR1	ACR0		Min	Typ	Max
0	0	0	AN0			
0	0	1	AN1			
0	1	0	AN2			
0	1	1	AN3			
1	0	0	V_{RH}**	FE**	FF**	FF**
1	0	1	V_{RL}*	00	00	01
1	1	0	V_{RH} 4*	3F	40	41
1	1	1	V_{RH} 2*	7F	80	81

*Internal (calibration) levels
**AN4 may replace the V_{RH} calibration channel if selected via mask option.

than V_{CC}, (2) V_{RL} should be equal to or greater than V_{SS} but less than maximum specifications, and (3) $V_{RH} - V_{RL}$ should be equal to or greater than 4 volts.

The A/D has a built-in 1/2 LSB offset intended to reduce the magnitude of the quantizing error to ±1/2 LSB, rather than +0, −1 LSB with no offset. This implies that, ignoring errors, the transition point from $00 to $01 occurs at 1/2 LSB above V_{RL}. Similarly, the transition from $FE to $FF occurs 1-1/2 LSB below V_{RH}, ideally.

INSTRUCTION SET

The MCU has a set of 59 basic instructions which can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

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

BIT MANIPULATION INSTRUCTIONS

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

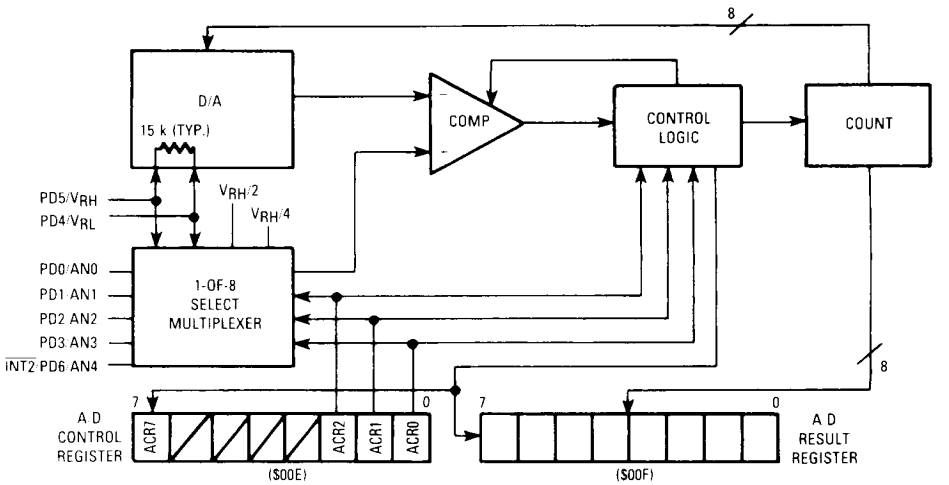


Figure 18. A/D Block Diagram

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

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 (2's 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

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 IFF Higher	BHI
Branch IFF Lower or Same	BLS
Branch IFF Carry Clear	BCC
(Branch IFF Higher or Same)	(BHS)
Branch IFF Carry Set	BCS
(Branch IFF Lower)	(BLO)
Branch IFF Not Equal	BNE
Branch IFF Equal	BEQ
Branch IFF Half Carry Clear	BHCC
Branch IFF Half Carry Set	BHCS
Branch IFF Plus	BPL
Branch IFF Minus	BMI
Branch IFF Interrupt Mask Bit is Clear	BMC
Branch IFF Interrupt Mask Bit is Set	BMS
Branch IFF Interrupt Line is Low	BIL
Branch IFF Interrupt Line is High	BIH
Branch to Subroutine	BSR

OPCODE MAP SUMMARY

Table 4 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, while the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. 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

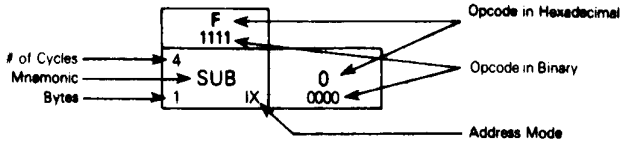
Table 4. Opcode Map

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

Abbreviations for Address Modes

INH	Inherent
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



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.

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.

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, 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 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 addressing 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.

BIT SET/CLEAR

In the bit set clear addressing mode, the bit to be set or cleared is part of the opcode. The byte following the opcode specifies the direct addressing of the byte to which the specified bit is to be set or cleared. Thus, 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 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.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage PC0 in Self-Check Mode All Other	V_{in}	−0.3 to +15.0 −0.3 to +7.0	V
Port A and C Source Current per Pin (One at a Time)	I_{out}	10	mA
Operating Temperature Range MC68705S3S MC68705S3CS	T_A	0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Junction Temperature Cerdip	T_J	175	°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 the V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$. Reliability of operation is enhanced if unused inputs except EXTAL are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Cerdip	θ_{JA}	60	°C W

3

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 = $P_{INT} + P_{PORT}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications, $P_{PORT} < P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

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

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.

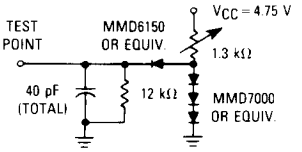


Figure 19. TTL Equivalent Test Load (Port B)

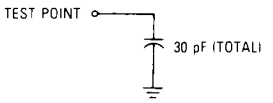


Figure 20. CMOS Equivalent Test Low (Port A)

ELECTRICAL CHARACTERISTICS

(V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
RESET Hysteresis Voltages "Out of Reset" "Into Reset"	 V_{IRES+} V_{IRES-}	1.3 1.5 0.8	— — —	2.0 2.5 1.8	V
Standby Supply Voltage (@ V_{CC} = 0 V	V_{STBY}	4.0	—	V_{CC} + 0.7	V
Standby Current (V_{STBY} = 4.0 V)	I_{STBY}	—	1.0	5.0	mA
Power Dissipation — No Port Loading (V_{CC} = 5.75 V, T_A = 0°C) (V_{CC} = 5.75 V, T_A = −40°C)	P_D	 — —	 800 925	 1006 1092	mW
Low Voltage Recover	V_{LVR}	—	—	4.75	V
Low Voltage Inhibit	V_{LVI}	—	3.75	—	V
Input Current $\overline{INT}$ (V_{in} = 2.4 V to V_{CC}) EXTAL (V_{in} = 2.4 V to V_{CC} Crystal Option) (V_{in} = 0.4 V Crystal Option) $\overline{RESET}$ (V_{in} = 5.75 V)	I_{in}	 — — — —	 20 — — 2500	 50 10 −1600 3800	µA

PROGRAMMING OPERATION ELECTRICAL CHARACTERISTICS

(V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 20° to 30°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Programming Voltage	V_{PP}	20.0	21.0	22.0	V
V_{PP} Supply Current V_{PP} = 21.0 V	I_{PP}	 —	 —	 30	mA
Programming Oscillator Frequency	f_{oscp}	0.9	1.0	1.1	MHz
Bootstrap Programming Mode Voltage (PC0 Pin) (@ I_{IHTP} = 100 µA Max)	V_{IHTP}	9.0	12.0	15.0	V

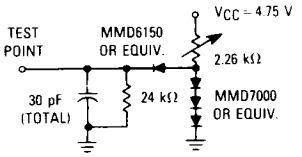


Figure 21. TTL Equivalent Test Load (Ports A and C)

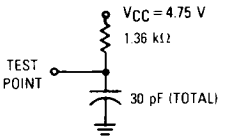


Figure 22. Open-Drain Equivalent Test Load (PB1, PB2, and PB3)

SWITCHING CHARACTERISTICS

(V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H), unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency	f_{osc}	0.4	—	4.2	MHz
Cycle time (4 f_{osc})	t_{cyc}	0.95	—	10	µs
$\overline{INT}$, $\overline{INT2}$, and TIMER Pulse Width $\overline{RESET}$ Pulse Width	t_{WL}, t_{WH} t_{RWL}	t_{cyc} + 250	—	—	ns
$\overline{RESET}$ Delay Time (External Capacitance = 1 µF)	t_{RHL}	—	100	—	ns
$\overline{INT}$ Zero-Crossing Detection Input Frequency (for ±5 Accuracy)	f_{INT}	0.03	—	1	kHz
External Clock Input Duty Cycle (EXTAL)	—	40	50	60	%
Oscillator Startup Time Crystal	t_{su}	—	—	100	ms
SPICL High Time	t_{SPICLH}	4	—	—	t_{cyc}
SPICL Low Time	t_{SPICHL}	4	—	—	t_{cyc}
SPICL Rise and Fall Time	t_{Sr}, t_{Sf}	—	—	1	µs
SPID Input Data Setup Time	t_{SDS}	2	—	—	tcyc
SPID Input Data Hold Time	t_{SDH}	2	—	—	t_{cyc}
SPICL to SPISS Lag Time	t_{SStG}	4	—	—	t_{cyc}
SPISS to SPICL Lead Time	t_{SSLD}	4	—	—	t_{cyc}
Start Bit to First Clock Lead Time	t_{STL}	1	—	—	t_{cyc}
External Timer Input to Timer Change Time	t_{PCT}	3	—	—	t_{cyc}
Timer Change to Port B Toggle Time	t_{TPB}	2	—	—	t_{cyc}
$\overline{INT2}$ to Timer A Load Time	t_{INTL}	3	—	—	t_{cyc}

A/D CONVERTER CHARACTERISTICS

(V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Min	Typ	Max	Unit	Comments
Resolution	8	8	8	Bits	
Non-Linearity*	—	—	±1/2	LSB	After removing zero-offset and full-scale errors
Quantizing Error	—	—	±1/2	LSB	
Conversion Range V_{RH} V_{RL}	— V_{SS}	— —	V_{CC} 0.2	V	A D accuracy may decrease proportionately as V_{RH}–V_{RL} is reduced below 4.0 V. The sum of V_{RH} and V_{RL} must not exceed V_{CC}
Conversion Time	30	30	30	t_{cyc}	Includes sampling time
Monotonicity	(Inherent within total error)				
Sample Time	5	5	5	t_{cyc}	
Sample Hold Capacitance, Input	—	—	25	pF	
Analog Input Voltage	V_{RL}	—	V_{RH}	V	Transients on any analog lines are not allowed at any time during sampling or accuracy may be degraded

*For V_{RH} = 4.0 V to 5.0 V and V_{RL} = 0 V.

PORT ELECTRICAL CHARACTERISTICS

(V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Port B					
Output Low Voltage, I_{Load} = 3.2 mA	V_{OL}	—	—	0.4	V
Output Low Voltage, I_{Load} = 10 mA (Sink)	V_{OL}	—	—	1.0	V
Output High Voltage, I_{Load} = −200 µA	V_{OH}	2.4	8	—	V
Darlington Current Drive (Source)*, V_O = 1.5 V	I_{OH}	1.0	—	10	mA
Input High Voltage	V_{IH}	2.0	—	V_{CC} + 0.7	V
Input Low Voltage	V_{IL}	V_{SS}	—	0.8	V
Hi-Z State Input Current	I_{TSI}	—	2	10	µA
Port C and Port A					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	—	—	0.4	V
Output High Voltage, I_{Load} = −100 µA	V_{OH}	2.4	—	—	V
Input High Voltage	V_{IH}	2.0	—	V_{CC} + 0.7	V
Input Low Voltage	V_{IL}	V_{SS}	—	0.8	V
Hi-Z State Input Current	I_{TSI}	—	2	10	µA
Port D (Digital Inputs Only)					
Input High Voltage	V_{IH}	2.0	—	V_{CC} + 0.7	V
Input Low Voltage	V_{IL}	V_{SS}	—	0.8	V
Input Current**	I_{in}	—	1	10	µA

*Not applicable if programmed to open-drain state.

**PD4 V_{RL} — PD5 V_{RH}.
The A/D conversion resistor (15 kΩ typical) is connected internally between these two lines, impacting their use as digital inputs in some applications.

ORDERING INFORMATION

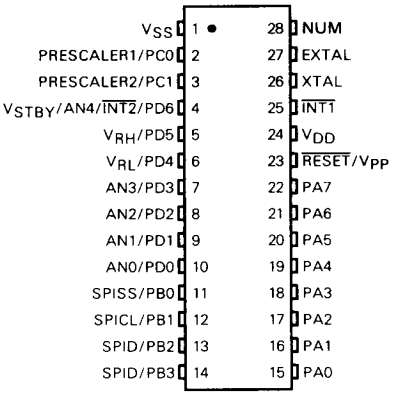
The following table provides generic information pertaining to the package type, temperature, and MC order numbers for the MC68705S3.

Table 5. Generic Information

Package Type	Temperature	Order Number
Cerdip (S Suffix)	0 C to 70 C	MC68705S3S
	−40 C to +85 C	MC68705S3CS

MECHANICAL DATA

PIN ASSIGNMENTS



3