Technical Summary
8-Bit Microcontroller Unit

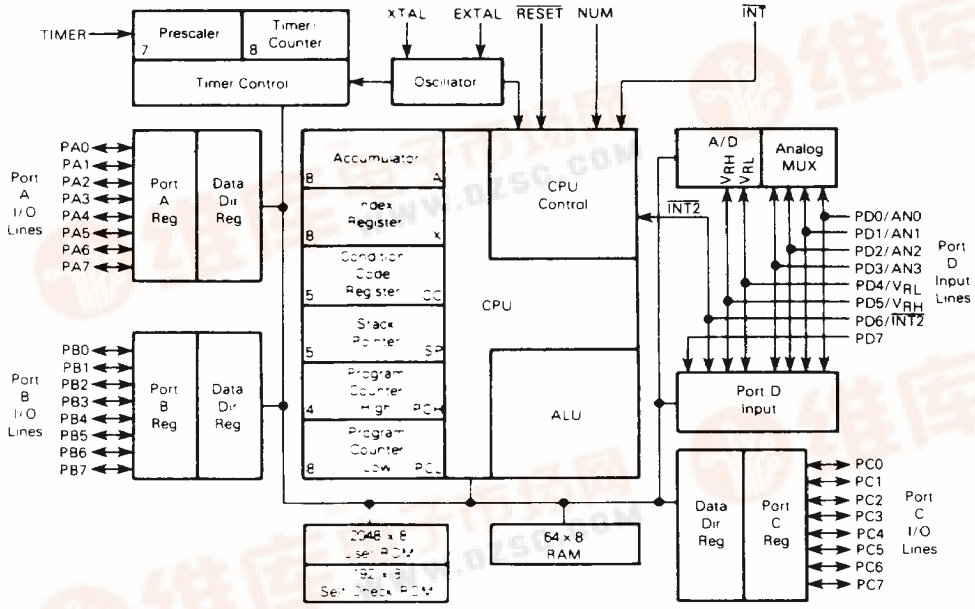
The MC6805R2 (HMOS) Microcontroller Unit (MCU) is a member of the MC6805 Family of microcontrollers. This low cost and high-speed MCU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for detailed information, refer to *M6805 HMOS, M146805 CMOS Family User's Manual* (M6805UM(AD2)) 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.

- Internal 8-Bit Timer with 7-Bit Prescaler
- On-chip Oscillator
- Memory Mapped I/O
- Versatile Interrupt Handling
- True Bit Manipulation
- Bit Test and Branch Instruction
- Vectored Interrupts
- Self-Check Mode
- 2048 Bytes of ROM
- 64 Bytes of RAM
- 24 Bidirectional I/O Ports
- A/D Converter

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 microcomputer using these two pins. V_{CC} is +5.25 volts (±0.5Δ) power, and V_{SS} is ground.

NUM

This pin is not for user applications and must be connected to V_{SS}.

$\overline{INT}$

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to **INTERRUPTS** for more detailed information.

EXTAL, XTAL

These pins provide control input for the on-chip clock oscillator circuit. A crystal, a resistor/capacitor combination, or an external signal (depending upon selected manufacturing mask option) is connected to these pins to provide a system clock.

RC Oscillator

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

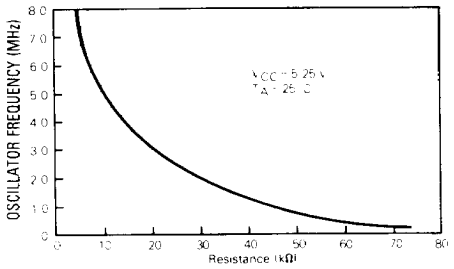
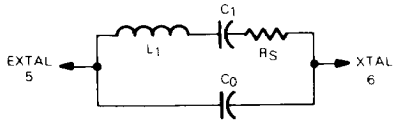


Figure 2. Typical Frequency vs Resistance for RC Oscillator Option Only

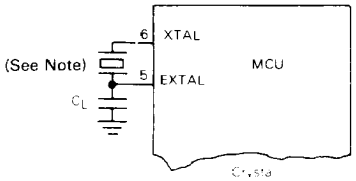
Crystal

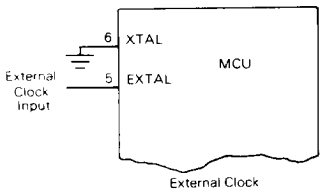
The circuit shown in Figure 1 is recommended when using a crystal. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and startup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



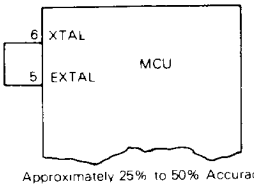
AT — Cut Parallel Resonance Crystal
C_0 = 7 pF Max
Freq. = 4.0 MHz @ C_L = 24 pF
R_S = 50 ohms Max

Piezoelectric ceramic resonators which have the equivalent specifications may be used instead of crystal oscillators. Follow ceramic resonator manufacturer's suggestions for C_0, C_1, and R_S values.

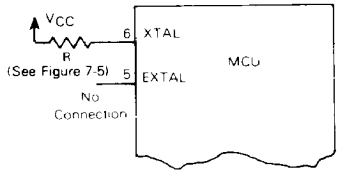




External Clock



Approximately 25% to 50% Accuracy
Typical t_{CYC} = 1.25 µs
External Jumper



Approximately 10% to 25% Accuracy
(Excludes Resistor Tolerance)
External Resistor

NOTE: The recommended C_L value with a 4.0 MHz crystal is 27 pF maximum, including system distributed capacitance. There is an internal capacitance of approximately 25 pF on the XTAL pin. 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 a 2 MHz crystal, use approximately 50 pF on EXTAL and approximately 25 pF on XTAL. The exact value depends on the Motional-Arm parameters of the crystal used.

Figure 1. Oscillator Connections

External Clock

An external clock should be applied to the EXTAL input with the XTAL input connected to ground, as shown in Figure 1. This option may only be used with the crystal oscillator option selected in the mask option register. The t_{OXOV} or t_{ILCH} specifications do not apply when using an external clock input.

TIMER

This pin is used as an external input to control the internal timer/counter circuitry. This pin also detects a higher voltage level used to initiate the self-test program.

$\overline{\text{RESET}}$

This pin has a Schmitt trigger input and an on-chip pullup. The MCU can be reset by pulling $\overline{\text{RESET}}$ low.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7)

These 32 lines are arranged into four 8-bit ports (A, B, C, and D). 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 four analog inputs, plus two voltage reference inputs when the A/D converter is used (PD5/V_{RH}, PD4/V_{RL}) and an $\overline{\text{INT2}}$ input. All Port D lines can be read directly and used as binary inputs. If any analog input is used, then PD5/V_{RH} and PD4/V_{RL} must be used in the analog mode. Refer to **PROGRAMMING** and **ANALOG-TO-DIGITAL CONVERTER** for additional information.

PROGRAMMING

INPUT/OUTPUT PROGRAMMING

Port A, B, and C pins are programmable as either input or output under software control of the corresponding data direction register (DDR). Port D lines are input only. 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 the 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.

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 D provides reference voltage ($\overline{\text{INT2}}$) and multiplexed analog inputs. Port D can always be used as digital input and may be used for analog if V_{RH} and V_{RL} are connected to the appropriate reference voltage. The V_{RH} (PD5) and V_{RL} (PD4) are internally connected to the A/D resistor.

Table 1. I/O Pin Functions

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

**Ports B and C are three-state ports. Port A has optional internal pullup devices to provide CMOS data drive capability.

MEMORY

The MCU is capable of addressing 4096 bytes of memory and I/O registers. The memory map is shown in Figure 4. The locations consist of user ROM, self-check ROM, user RAM, A/D registers, a miscellaneous control register,

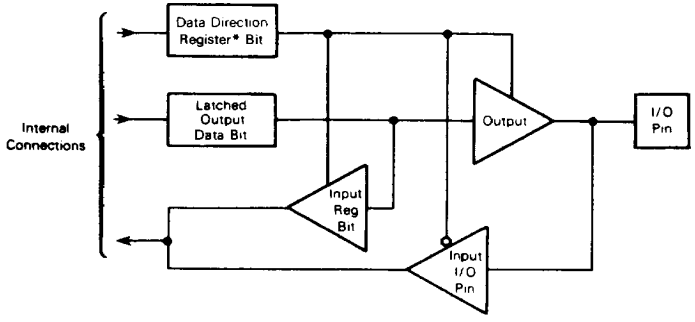
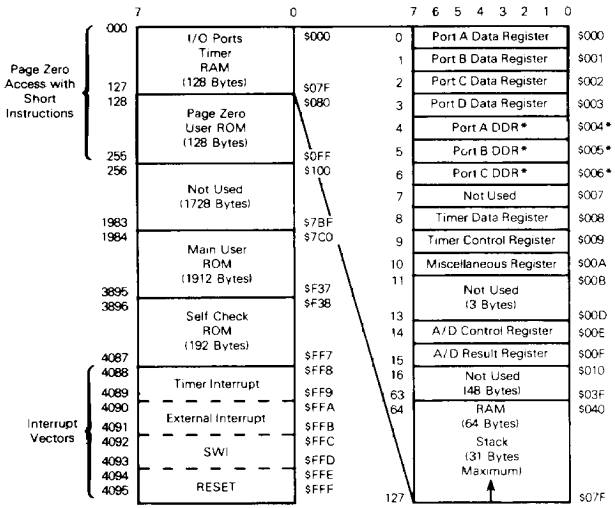


Figure 3. Typical Port I/O Circuitry and Register Configuration



* Caution: Data direction registers (DDRs) are write-only; they read as $FF.

Figure 4. Memory Map

and I/O. The interrupt and reset vectors are located from $FF8 to $FFF.

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

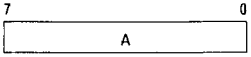
Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

REGISTERS

The MCU contains the registers described in the following paragraphs.

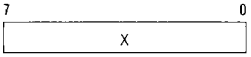
ACCUMULATOR (A)

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



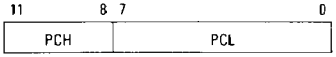
INDEX REGISTER (X)

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)

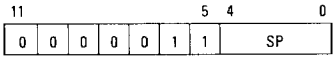
The program counter is an 12-bit register that contains the address of the next byte to be fetched.



STACK POINTER (SP)

The stack pointer is an 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.

4				0
H	I	N	Z	C

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 external 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).

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.

SELF-CHECK

The self-check is initiated by connecting the MCU as shown in Figure 5 and then monitoring the output of port C (bit 3) for an oscillation of approximately 7 Hz. The following test are executed automatically:

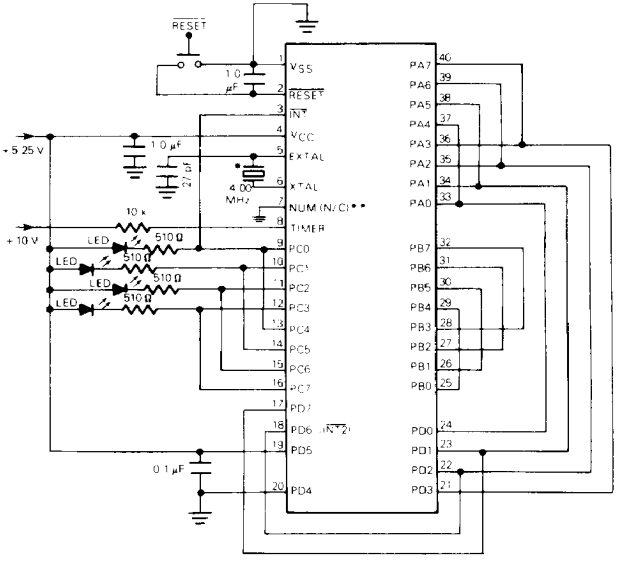
I/O — Functionally exercise I/O ports.
RAM — Walking bit test.
ROM — Exclusive OR with ODD "1st" parity result.
Timer — functionally exercise timer.
Interrupts — Functionally exercise external and timer interrupts.
A/D Converter — Functionally test the Analog-to-Digital Converter.

The RAM, ROM, and the A/D test can be called by a user program. The timer test may be called if the timer input is the internal clock. Table 2 shows the status of the LEDs as a result of a failure. Port C is tested only once (just after reset). If port C fails, no lights will appear.

Table 2. Self-Check Error Patterns

PC0	PC1	PC2	PC3	Remarks (1: LED ON; 0: LED OFF)
1	0	1	0	Bad I/O
0	0	1	0	Bad Timer
1	1	0	0	Bad RAM
0	1	0	0	Bad ROM
1	0	0	0	Bad A/D
0	0	0	0	Bad Interrupts or Request Flag
All Flashing				Good Device

Anything else Bad Part, Bad Port C, etc.



*This connection depends on clock oscillator user selectable mask option. Use jumper if the RC mask option is selected.
**Pin 7 is not for user application and must be connected to V_{SS}.

Figure 5. Self-Check Connections

RESETS

The MCU can be reset three ways: (1) by initial power-up, (2) by the external reset input ($\overline{\text{RESET}}$), and (3) by an optional, internal, low-voltage detect circuit. The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger that senses the $\overline{\text{RESET}}$ line logic level.

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 the $\overline{\text{RESET}}$ input to go high. Connecting a capacitor to the $\overline{\text{RESET}}$ input (Figure 6) 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.

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 $\overline{\text{RESET}}$ 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 four different ways: (1) through the external interrupt $\overline{\text{INT}}$ input pin, (2) with the internal timer interrupt request, (3) using the software interrupt instruction (SWI) or (4) the external port D bit 6 ($\overline{\text{INT2}}$) input pin.

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, and then normal processing resumes. The stacking order is shown in Figure 7.

Unlike $\overline{\text{RESET}}$, 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.

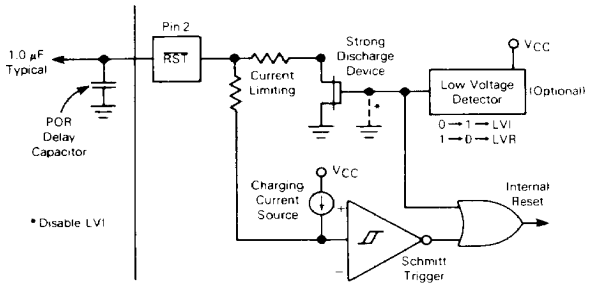


Figure 6. $\overline{\text{RESET}}$ Configuration

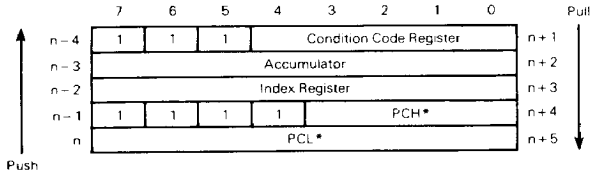


Figure 7. Interrupt Stacking Order

When the current instruction is complete, the processor checks all pending hardware interrupts and, if unmasked (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 8 for the reset and interrupt instruction processing sequence.

TIMER INTERRUPT

If the timer mask bit (TCR6) is cleared, then, each time the timer decrements to zero (transitions from $01 to $00), an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register (CCR) is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I bit in the CCR is set, masking further interrupts until the present one is serviced. The contents of the timer interrupt vector, containing the location of the timer interrupt service routine, is then loaded into the program counter. At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

EXTERNAL INTERRUPT

The external interrupt is internally synchronized and then latched on the falling edge of $\overline{INT}$ 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 and 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_{INT} maximum) can be used to generate an external interrupt (see Figure 9a) for use as a zero-crossing detector (for negative transitions of the ac sinusoid). This type of circuit allows applications

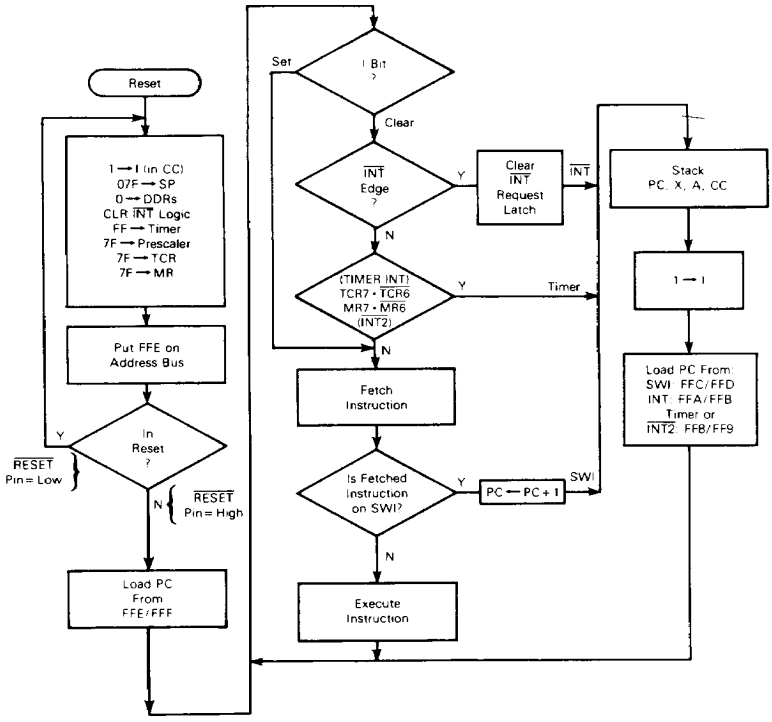


Figure 8. Reset and Interrupt Processing Flowchart

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 9b), the $\overline{INT}$ pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{INT}$ 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.

TIMER

The MCU consists of an 8-bit software programmable counter driven by a 7-bit prescaler. The timer source is made during manufacturing as a mask option. The 8-bit counter may be loaded under program control and is decremented toward zero. When the timer reaches zero, the timer interrupt request bit (bit 7) in the timer control register (TCR) is set. Refer to Figure 10 for timer block diagram.

The timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. When the I bit in the condition code register is cleared, the processor receives the interrupt. The MCU responds to this interrupt by 1) saving the present CPU state on the stack, 2) fetching the timer interrupt vector, and 3) executing the interrupt routine. The timer interrupt request bit must be cleared by software. The TIMER and $\overline{INT2}$ share the same interrupt vector, therefore the interrupt routine must check the request bits to determine the source of the interrupt. Refer to **RESETS** and **INTERRUPTS** for additional information.

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. To avoid truncation errors, the prescaler is cleared when TCR bit 3 is set to a logic one; however, the TCR bit 3 always reads as a logic zero to ensure proper operation with read-modify-write instructions.

The timer continues to count past zero, falling from $00 through $FF, and continues the countdown. The counter can be read at any time by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred without disturbing the counting process. Three machine cycles are required for a change in state of the TIMER pin to decrement the timer prescaler.

Clock input to the timer can be from an external source or from the internal phase two signal. Clock source is one of the mask options. A prescaler mask option is available to select a divide option of a power of two up to 128.

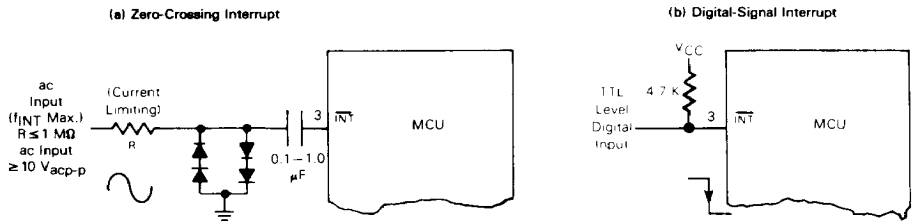


Figure 9. Typical Interrupt Circuits

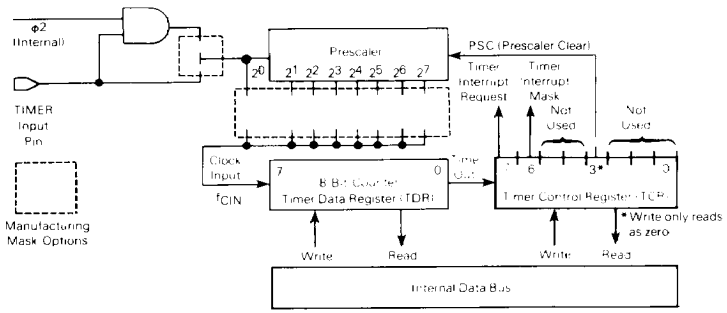


Figure 10. Timer Block Diagram

TIMER CONTROL REGISTER (TCR) ($009)

This 8-bit register controls various functions such as write timer interrupt request, timer interrupt inhibit, and prescaler clear. Bit 3 is write only.

7	6	5	4	3	2	1	0
TIR	TIM	1	1	PSC	1	1	1
RESET:							
0	1	U	U	U	U	U	U

TIR — Timer Interrupt Request
Used to indicate the timer interrupt when it is logic one.
1 = Set when the timer data register changes to all zeros.
0 = Cleared by external reset, power-on reset, or under program control.

TIM — Timer Interrupt Mask
Used to inhibit the timer interrupt.
1 = Interrupt inhibited.
0 = Interrupt enabled.

PSC — Prescaler Clear
Write only bit. Writing a one to this bit resets the prescaler to zero. A read of this location always indicates a zero.

Bits 5, 4, 2, 1, 0 — Not Used.

ANALOG-TO-DIGITAL CONVERTER

The chip resident 8-bit analog-to-digital (A/D) converter uses a successive approximation technique as shown in Figure 11. 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.

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 11. 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 set, selected input is sampled again, and a new conversion starts. When ACR7 is cleared, the conversion in progress is aborted and the selected input is sampled for five machine cycles and held internally.

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

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 than V_{DD}, (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.

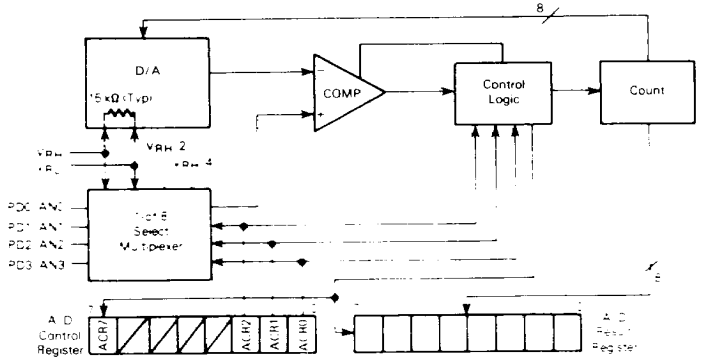


Figure 11. A/D Block Diagram

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 list of instructions.

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 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)

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 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

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

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 coversion 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 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.

INDEX, 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 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.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write

3

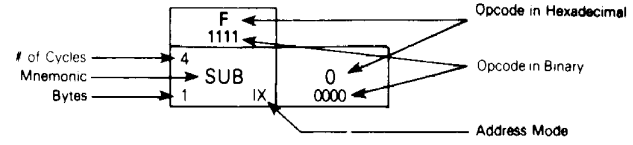
Table 4. Opcode Map

Low \ Hi	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 DIR	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	8 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



functions, these instructions cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store 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.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to −7.0	V
Input Voltage Self-Check Mode (TIMER Pin Only)	V_{in}	0.3 to −15.0	V
Operating Temperature Range MC6805R2 MC6805R2C MC6805R2V	T_A	T_L to T_H 0 to +70 −40 to 85 40 to 105	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Junction Temperature Plastic PLCC Cerdip	T_J	 50 150 175	°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 $V_{SS} \leq (V_{in}$ and $V_{out}) \leq V_{CC}$. 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 Plastic (P Suffix) PLCC (FN Suffix) Cerdip (S Suffix)	θ_{JA}	 60 100 60	°C/W

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.

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
Input High Voltage	V_{IH}				V
$\overline{RESET}$ (4.75 ≤ V_{CC} ≤ 5.75)		4.0	—	V_{CC}	
V_{CC} (4.75)		V_{CC} − 0.5	—	V_{CC}	
$\overline{INT}$ (4.75 ≤ V_{CC} ≤ 5.75)		4.0	*	V_{CC}	
(V_{CC} (4.75)		V_{CC} − 0.5	*	V_{CC}	
All Other		2.0	—	V_{CC}	
Input High Voltage Timer	V_{IH}				V
Timer Mode		2.0	—	V_{CC} + 1.0	
Self-Check Mode		9.0	10.0	15.0	
Input Low Voltage	V_{IL}				V
$\overline{RESET}$		V_{SS}	—	0.8	
$\overline{INT}$		V_{SS}	*	1.5	
All Other (Except A/D Inputs)		V_{SS}	—	0.8	
$\overline{RESET}$ Hysteresis Voltages					V
"Out of Reset"	V_{IRES+}	2.1	—	4.0	
"Into Reset"	V_{IRES-}	0.8	—	2.0	
$\overline{INT}$ Zero-Crossing Input Voltage, Through a Capacitor	V_{INT}	2	—	4	$V_{ac\ p\text{-}p}$
Power Dissipation — (No Port Loading, V_{CC} = 5.75 V, T_A = 0°C	P_D	—	520	740	mW
for Steady-State Operation) T_A = −40°C		—	580	800	
Input Capacitance	C_{in}				pF
XTAL		—	25	—	
All Other Except Analog Inputs (See Note)		—	10	—	
Low Voltage Recover	V_{LVR}	—	—	4.75	V
Low Voltage Inhibit	V_{LVI}	2.75	3.75	4.70	V
Input Current					μA
TIMER (V_{in} = 0.4)		—	—	20	
$\overline{INT}$ (V_{in} = 2.4 V to V_{CC})		—	20	50	
EXTAL (V_{in} = 2.4 V to V_{CC} Crystal Option)	I_{in}	—	—	10	
(V_{in} = 0.4 V Crystal Option)		—	—	−1600	
$\overline{RESET}$ (V_{in} = 0.8 V)	I_{RES}	−4.0	—	40	
(External Capacitor Charging Current)					

NOTE: Port D Analog Inputs, when selected C_{in} = 25 pF for the first 5 out of 30 cycles.
*Due to internal biasing this input (when unused) floats to approximately 2.0 V.

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

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	t_{WL}, t_{WH}	t_{cyc} + 250	—	—	ns
$\overline{RESET}$ Pulse Width	t_{RWL}	t_{cyc} + 250	—	—	ns
$\overline{INT}$ Zero-Crossing Detection Input Frequency	f_{INT}	0.03	—	1	kHz
External Clock Input Duty Cycle (EXTAL)	—	40	50	60	%
Crystal Oscillator Start-Up Time	—	—	—	100	ms

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)

	Min	Typ	Max	Unit	Comments
Resolution	8	8	8	Bits	
Total Error	—	—	± 2.25*	LSB	Difference between ideal and actual transfer characteristics (includes non-linearity, zero offset and full scale errors)
Absolute Accuracy	—	—	± 2.75*	LSB	Difference between the actual input voltage and the full-scale weighted equivalent of the binary output code. All error sources included
Quantizing Error	—	—	± .5	LSB	Uncertainty due to converter resolution (inherent)
Conversion Range	V_{RL}	—	VRH	V	
V_{RH}	—	—	V_{CC}	V	A/D accuracy may decrease proportionately as VRH is reduced below 4.75 V. The sum of V_{RH} and V_{RL} must not exceed V_{CC}
V_{RL}	V_{SS}	—	0.2	V	
Conversion Time	30	30	30	t_{cyc}	Includes sample time
Monotonicity					Inherent with total error
Sample Time	5	5	5	t_{cyc}	
Sample/Hold Capacitance, Input	—	—	25	pF	
Analog Input Voltage	V_{RL}	—	V_{RH}	V	Negative transients on any analog lines (Pins 19-24) are not allowed at any time during conversion.

*Note: Accuracy may decrease at temperatures above T_A = 85°C or f_{osc} < 3.57 MHz.

3

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 A with CMOS Drive Enabled					
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
Output High Voltage, I_{Load} = −10 μA	V_{OH}	V_{CC} − 1.0	—	—	V
Input High Voltage, I_{Load} = −300 μA (max.)	V_{IH}	2.0	—	V_{CC}	V
Input Low Voltage, I_{Load} = −500 μA (max.)	V_{IL}	V_{SS}	—	0.8	V
Hi-Z State Input Current (V_{in} = 2.0 V to V_{CC})	I_{IH}	—	—	−300	μA
Hi-Z State Input Current (V_{in} = 0.4 V)	I_{IL}	—	—	500	μA
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	—	—	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}	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 with TTL Drive					
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}	V
Input Low Voltage	V_{IL}	V_{SS}	—	0.8	V
Hi-Z State Input Current	I_{TSI}	—	<2	10	μA
Port C (Open-Drain Option)					
Input High Voltage	V_{IH}	2.0	—	13.0	V
Input Low Voltage	V_{IL}	V_{SS}	—	0.8	V
Input Leakage Current (V_{in} = 13.0 V)	I_{LOD}	—	<3	15	μA
Output Low Voltage I_{Load} = 1.6 mA	V_{OL}	—	—	0.4	V
Port D (Digital Inputs Only)					
Input High Voltage	V_{IH}	2.0	—	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	—	0.8	V
Input Current	I_{in}	—	<1	5	μA

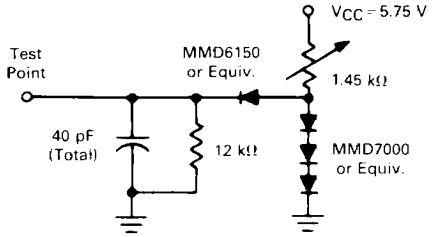


Figure 12. TTL Equivalent Test Load (Port B)

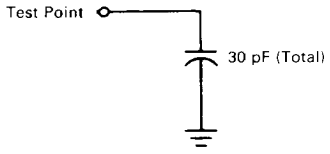


Figure 13. CMOS Equivalent Test Load (Port A)

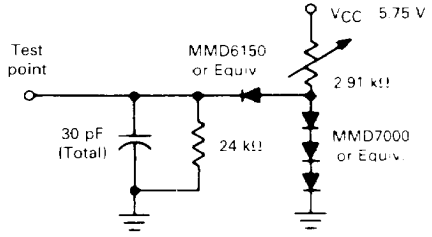


Figure 14. TTL Equivalent Test Load (Ports A and C)

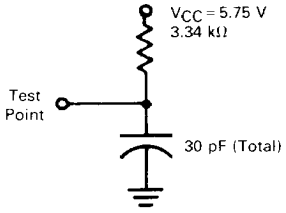


Figure 15. Open-Drain Equivalent Test Load (Port C)

3

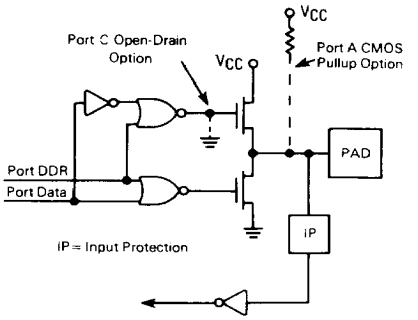


Figure 16. Ports A and C Logic Diagram

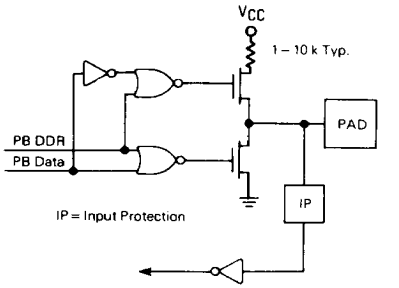


Figure 17. Port B Logic Diagram

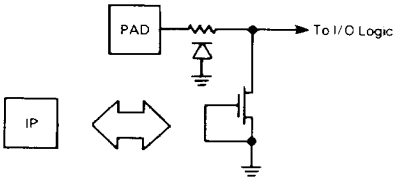


Figure 18. Typical Input Protection

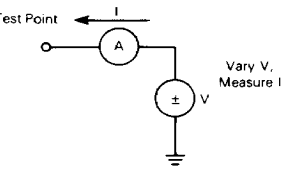


Figure 19. I/O Characteristic Measurement Circuit

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
EPROM(s) MC68705R3, 2532, 2732, or two 2516/2716

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

FLEXIBLE DISKS

Several types of flexible disks (MDOS or MS-DOS/PC-DOS disk file), programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. In either case, the diskette should be clearly labeled with the customers name, date, 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 speed up the process in case of any difficulty with the pattern file.

MDOS Disk File

MDOS is Motorola's Disk Operating System available on the EXORciser® development system. The disk media submitted must be a single-side, single-density, 8-inch MDOS compatible floppy diskette. The diskette must contain the minimum set of MDOS system files in addition to the pattern file.

The .LO output of the M6805 cross assembler should be furnished. In addition, the file must be produced (using the ROLLOUT command) containing the absolute image of the M6805 memory. Include the entire memory image of both data and program space. All unused bytes, including those in the user space, must be set to zero.

MS-DOS/PC-DOS Disk File

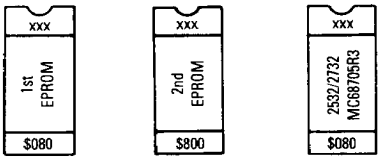
MS-DOS is Microsoft's Disk Operating System. PC-DOS is IBM® Personal Computer (PC) Disk Operating System. Disk media submitted must be a standard density (360K) double sided 5 1/4 inch compatible floppy diskette. The diskette must contain object file code in Motorola's S-record format. The S-record format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

EPROMs

An MC68705R3, 2532, 2732, 2516 (2), or 2716 (2) type EPROM(s), programmed with the customer 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 MC68705R3/2532/2732 or two 2516 2716 type EPROM(s), the EPROM(s) must be programmed as described in the following paragraph.

For the 2532, 2732, or the MC68705R3, the ROM code should be located from $080 to $FF; and $700 to $F37 and the interrupt vectors from $FF8 to $FFF. For the 2516's or 2716's, the ROM code should be located from $080 to $FF and $7C0 to $7FF in the first EPROM and from $0 to $737 in the second EPROM. The interrupt vectors should be in the second EPROM from $7F8 to $7FF.

EPROM MARKING



xxx = Customer ID

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) 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 completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for creation of the customer mask. To aid in the verification process, Motorola will program (customer supplied) blank EPROM(s) or DOS disk 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 at room temperature and five volts. These RVUs are free with the minimum order quantity but are not production parts. These RVUs are not guaranteed by Motorola Quality Assurance.

ORDERING INFORMATION

The following table provides generic information pertaining to the package type, temperature, and MC part numbers for the MC6805R2.

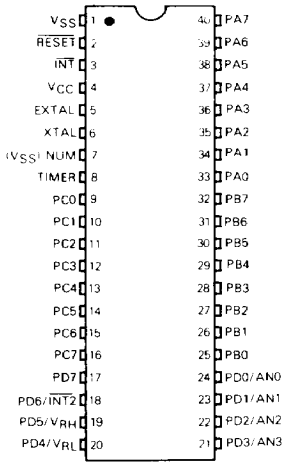
Package Type	Temperature	Part Number
Plastic (P Suffix)	0°C to 70°C −40°C to +85°C	MC6805R2P MC6805R2CP
Cerdip S Suffix	0°C to 70°C −40°C to +85°C	MC6805R2S MC6805R2CS
PLCC FN Suffix	0°C to 70°C −40°C to +85°C	MC6805R2FN MC6805R2CFN

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

MECHANICAL DATA

PIN ASSIGNMENTS

Dual-in-Line Package



PLCC Package