MC146805F2

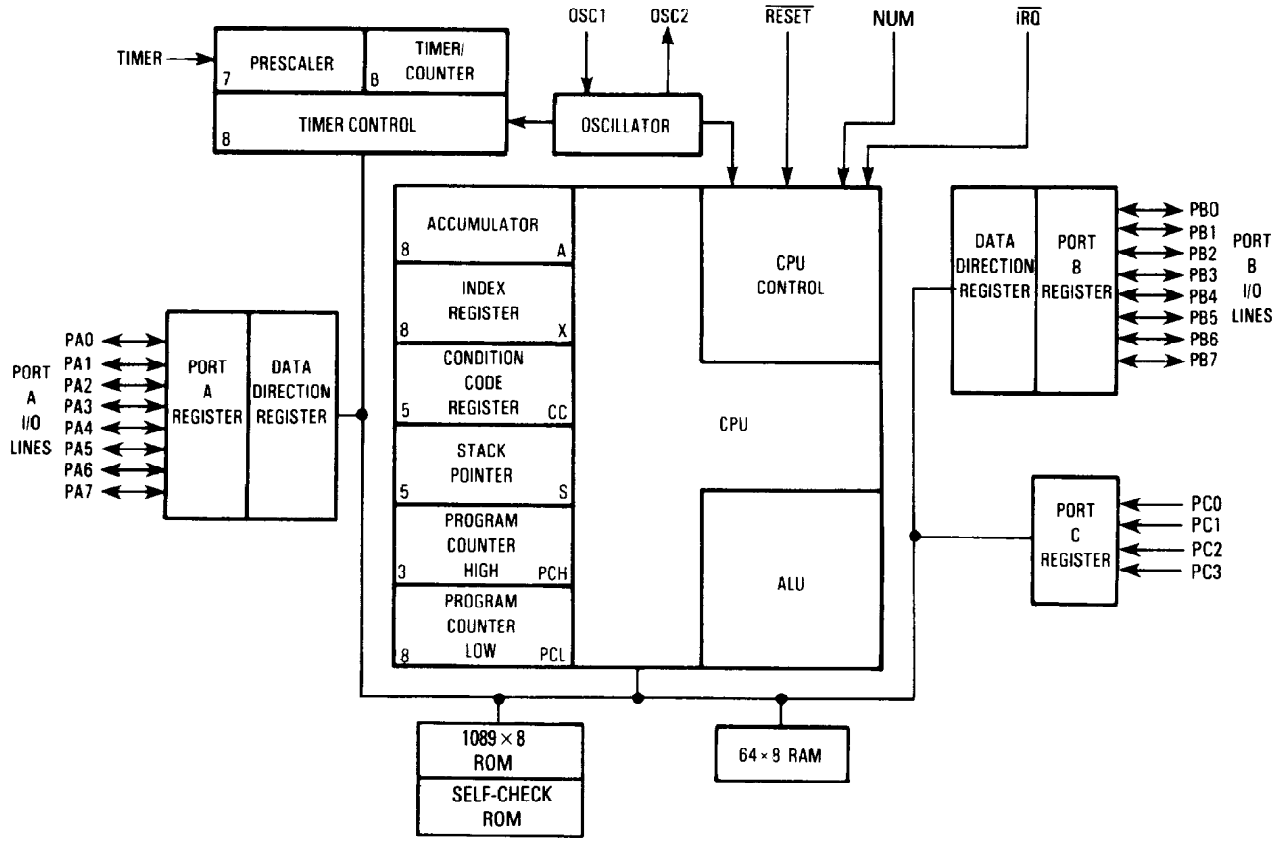
Technical Summary
8-Bit Microcomputer Unit

The MC146805F2 (CMOS) Microcomputer Unit (MCU) is a member of the MC146805 Family of microcomputers. This low cost and low power 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 Programmable Prescaler
- On-chip Clock
- Memory Mapped I/O
- Versatile Interrupt Handling
- True Bit Manipulation
- Bit Test and Branch Instruction
- Vectored Interrupts
- Self-check
- Power-saving STOP and WAIT Modes
- Single 3.0- to 6.0-Volt Supply
- Fully Static Operation
- 1089 Bytes of ROM
- 64 Bytes of RAM
- 20 I/O Ports

BLOCK DIAGRAM



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

SIGNAL DESCRIPTION

V_{DD} AND V_{SS}

Power is supplied to the microcomputer using these two pins. V_{DD} is +5 volts (±0.5Δ) power, and V_{SS} is ground.

NUM

This pin is intended for use in self-check only. In normal operation, this pin is connected to V_{SS}. A resistor of up to 10 kilohms to ground may be used if the oscillator is 32.768 KHz or less.

$\overline{IRQ}$

This pin is a photomask option with two different choices of interrupt triggering sensitivity; level and negative edge or negative edge only. Refer to **INTERRUPTS** for more detailed information.

OSC1, OSC2

These pins provide control input for the on-chip clock oscillator circuit. A crystal, a resistor, or an external signal 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(b). The relationship between R and f_{osc} is shown in Figure 2.

Crystal

The circuit shown in Figure 1(a) 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_{DD} specifications.

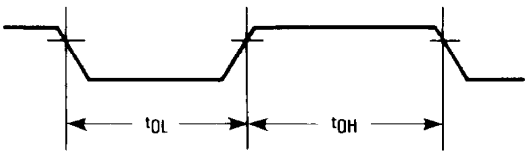
External Clock

An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 1(c). This option may only be used with the crystal oscillator option. The t_{OXOV} or t_{ILCH} specifications do not apply when using an external clock input.

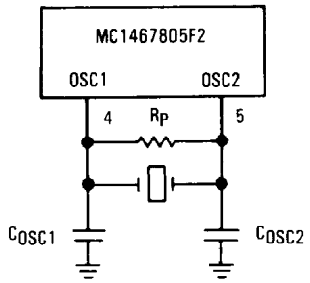
CRYSTAL PARAMETERS

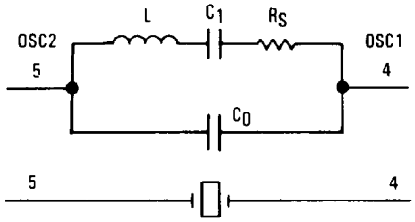
	1 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30 k	40 k	–

(a) OSCILLATOR WAVEFORM

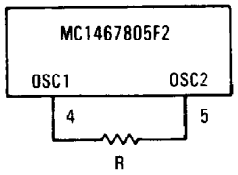


(b) CRYSTAL OSCILLATOR CONNECTIONS AND EQUIVALENT CRYSTAL CIRCUIT





(c) RC OSCILLATOR CONNECTION*



*Approximately 10% to 25% accuracy (excludes resistor tolerance) external register

(d) EXTERNAL CLOCK SOURCE CONNECTIONS

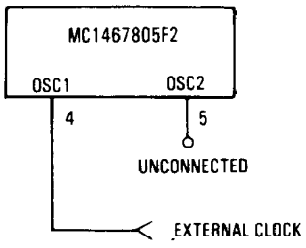


Figure 1. Oscillator Connections

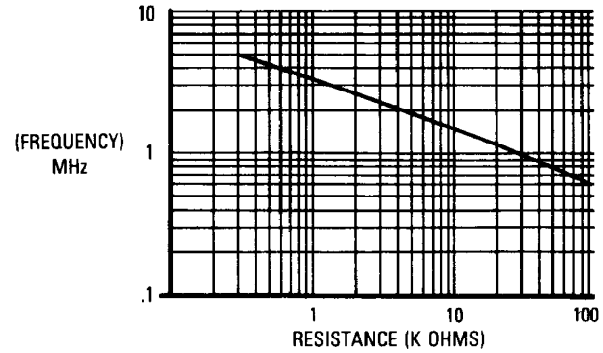


Figure 2. Typical Frequency vs Resistance for RC Oscillator Option Only

TIMER

This pin is used as an external input to control the internal timer/counter circuitry.

$\overline{\text{RESET}}$

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-PC3)

These 20 lines are arranged into two 8-bit ports (A and B) and one 4-bit port (C). Port A and Port B are programmable as either inputs or outputs under software control of the data direction registers. Port C is fixed input ports and not controlled by any data register. Refer to **PROGRAMMING** for additional information.

PROGRAMMING

INPUT/OUTPUT PROGRAMMING

Ports A and B are programmable as either input or output under software control of the corresponding write-only data direction register (DDR). The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic 1 for output and a logic 0 for input. On reset, all the DDRs are initialized to a logic 0 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 (0) and also to the latched output when the DDR is an output (1). Refer to Table 1 for I/O functions and to Figure 3 for typical port circuitry.

MEMORY

The MCU is capable of addressing 2048 bytes of memory and I/O registers. The memory map is shown in Figure 4. The locations consist of self-check ROM, user ROM, user RAM, a timer control register, and I/O. The interrupt vectors are located from $7F8 to $7FF.

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

NOTE

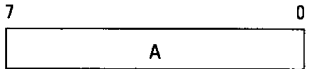
Using the shared stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

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

Table 1. I/O Pin Functions

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

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

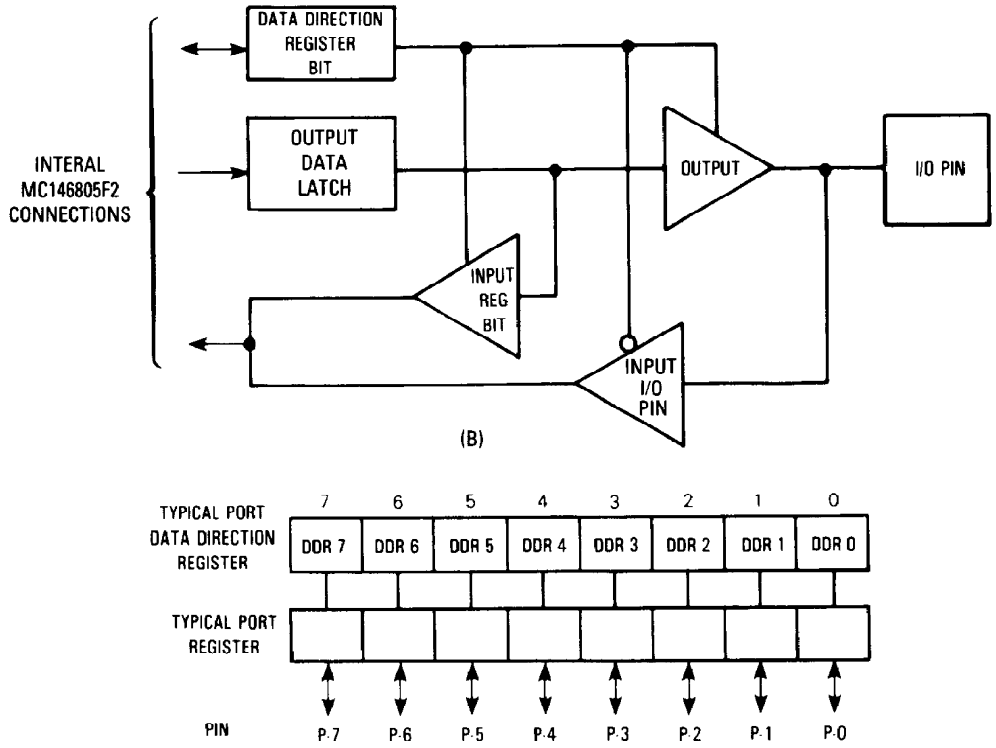
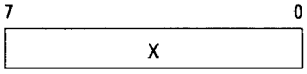


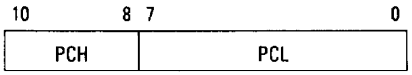
Figure 3. Typical Port I/O Circuitry and Register Configuration

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)

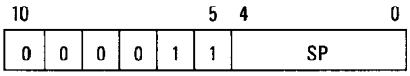
The program counter is an 11-bit register that contains the address of the next byte to be fetched.



STACK POINTER (SP)

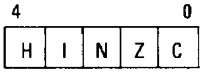
The stack pointer is an 11-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set at location $07F. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

The six most-significant bits of the stack pointer are permanently set at 000011. 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 specifications can be taken as a result of their state. Each bit is explained in the following paragraphs.



Half Carry (H)

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

Interrupt (I)

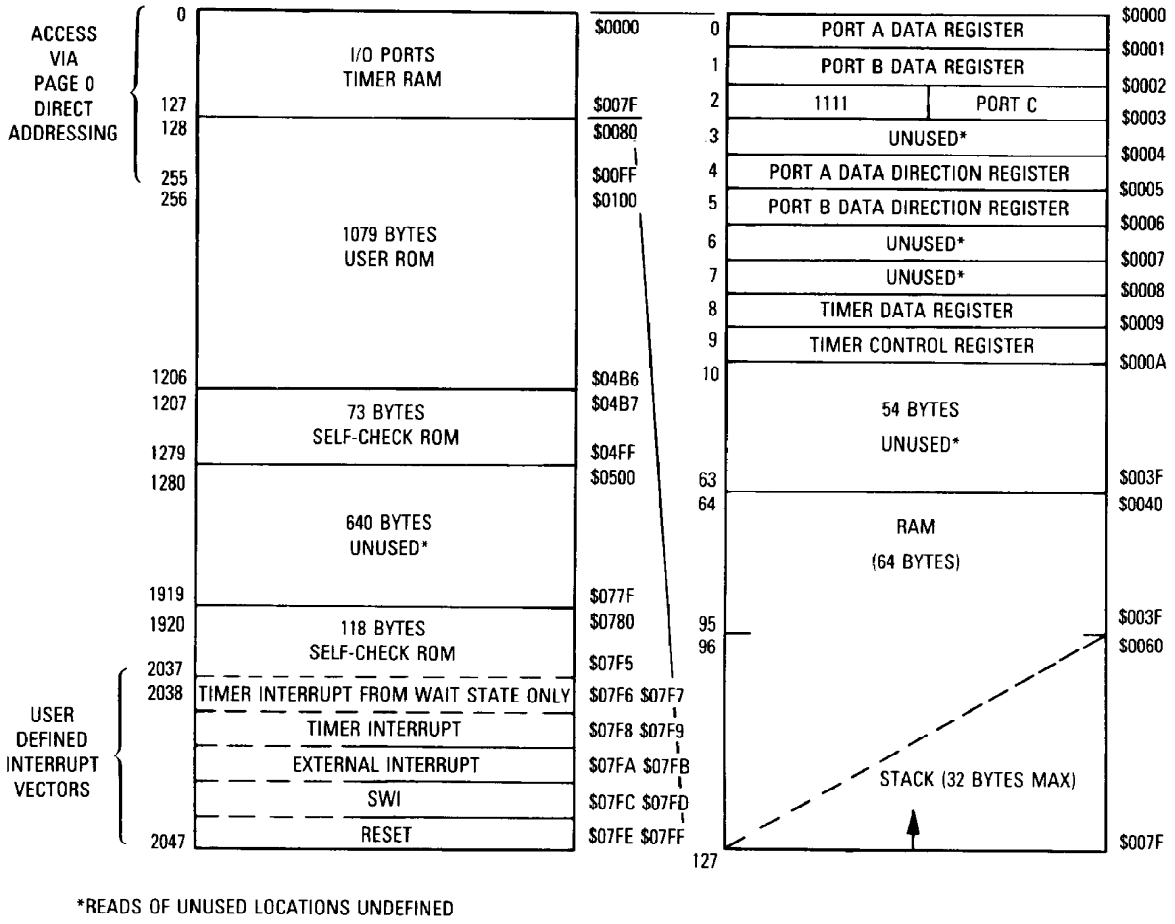
When this bit is set, the timer and external interrupt is masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

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 1).

Zero (Z)

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



*READS OF UNUSED LOCATIONS UNDEFINED

Figure 4. Memory Map

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 tying NUM and TIMER pins to a logic "1", and then executing a reset. The following tests are executed automatically:

I/O — Functionally exercise ports A, B, and C
RAM — Walking bit test
ROM — Exclusive OR with ODD "1s" parity result
Timer — Functionally exercise timer
Interrupts — Functionally exercise external and timer interrupts

The RAM self check, ROM checksum, and timer test are available to the user and do not require any external hardware.

RESETS

The MCU can be reset two ways: by initial power-up and by the external reset input ($\overline{\text{RESET}}$). 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. There is a 1920 t_{cyc} delay after the oscillator becomes active. If the $\overline{\text{RESET}}$ pin is low at the end of 1920 t_{cyc}, the MCU will remain in the reset condition until $\overline{\text{RESET}}$ goes high.

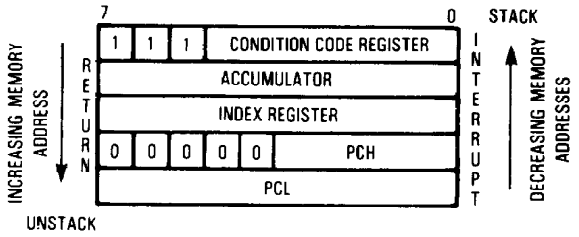
EXTERNAL RESET INPUT

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.

INTERRUPTS

The MCU can be interrupted three different ways: (1) through the external interrupt $\overline{IRQ}$ input pin, (2) with the internal timer interrupt request, or (3) using the software interrupt instruction (SWI).

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 5.



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

Figure 5. Interrupt Stacking Order

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 (I bit clear) proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Masked interrupts are latched for later interrupt service.

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 if the I bit is set (hardware interrupts masked). Refer to Figure 6 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

If the interrupt mask bit (I bit) of the condition code register is set, all interrupts are disabled. Clearing the I bit enables the external interrupts. The external interrupt is internally synchronized and then latched on the falling edge of $\overline{IRQ}$. The action of the external interrupt is identical to the timer interrupt with the exception that the interrupt request input at $\overline{IRQ}$ is latched internally, and the service routine address is specified by the contents of $7FA and $7FB.

Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive only trigger are available as mask options. Figure 7 shows both a functional and mode timing diagram for the interrupt line. The timing diagram shows two treatments of the interrupt line to the processor. The first method shows a single pulse on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service. Once a pulse occurs, the next pulse should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 20 instructions cycles to the total number of cycles it takes to complete the service routine. The second method shows many interrupt lines "wire-ORed" to form the interrupts at the processor. If the interrupt line remains low after servicing an interrupt, then the next interrupt is recognized.

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 is executed similar to the hardware interrupts.

LOW-POWER MODES

STOP

The STOP instruction places the MCU in its lowest power consumption mode. In the STOP mode, the internal oscillator is turned off, causing all internal processing and the timer to be halted; refer to Figure 8.

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt request and to disable any further time interrupts. The timer prescaler is cleared. The I bit in the condition code register is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can only be brought out of the STOP mode by an external interrupt or reset.

WAIT

The WAIT instruction places the MCU in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode. In the WAIT mode, the internal clock is disabled from all internal circuitry except for the timer; refer to Figure 9. Thus, all internal processing is halted; however, the timer continues to count normally.

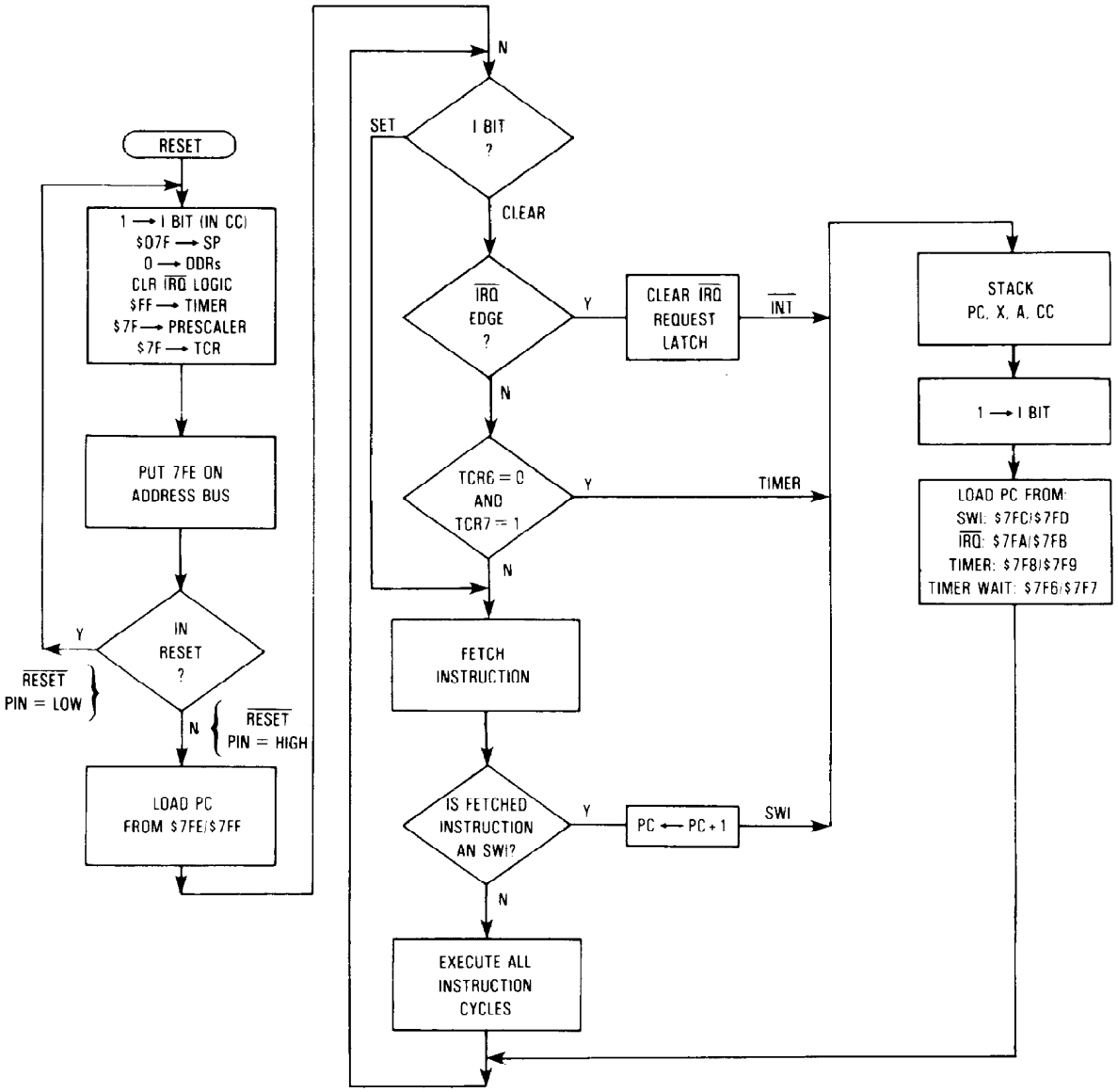


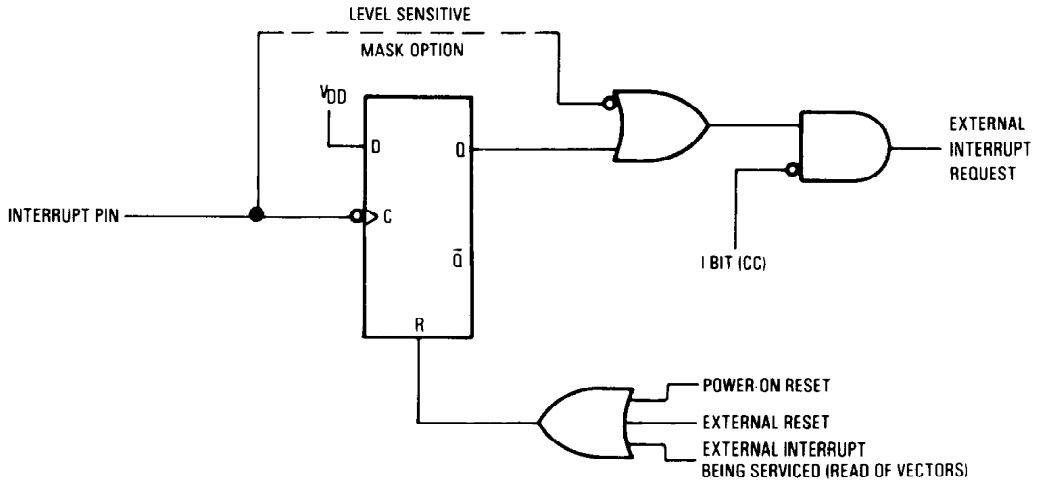
Figure 6. Reset and Interrupt Processing Flowchart

During the WAIT mode, the I bit in the condition code register is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode. If an external and a timer interrupt occur at the same time, the external interrupt is serviced first; then, if the timer interrupt request is not cleared in the external interrupt routine, the normal timer interrupt (not the timer wait interrupt) is serviced since the MCU is no longer in the WAIT mode.

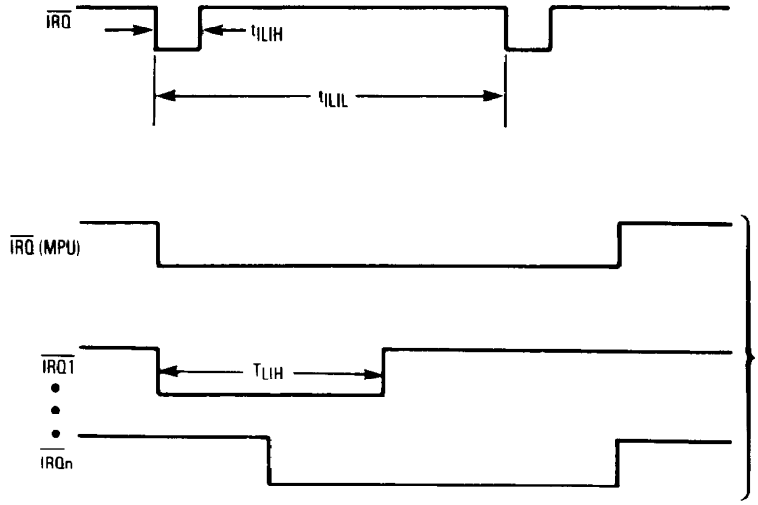
TIMER

The MCU consists of an 8-bit software programmable counter driven by a 7-bit software programmable prescaler. The various timer sources are made via the timer control register (TCR). 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.

(a) INTERRUPT FUNCTIONAL DIAGRAM



(b) INTERRUPT MODE DIAGRAM



Edge Condition

The minimum pulse width (t_{ILIH}) is one t_{cyc}. The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles.

Mask Optional Level Sensitive

If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

Figure 7. External Interrupt

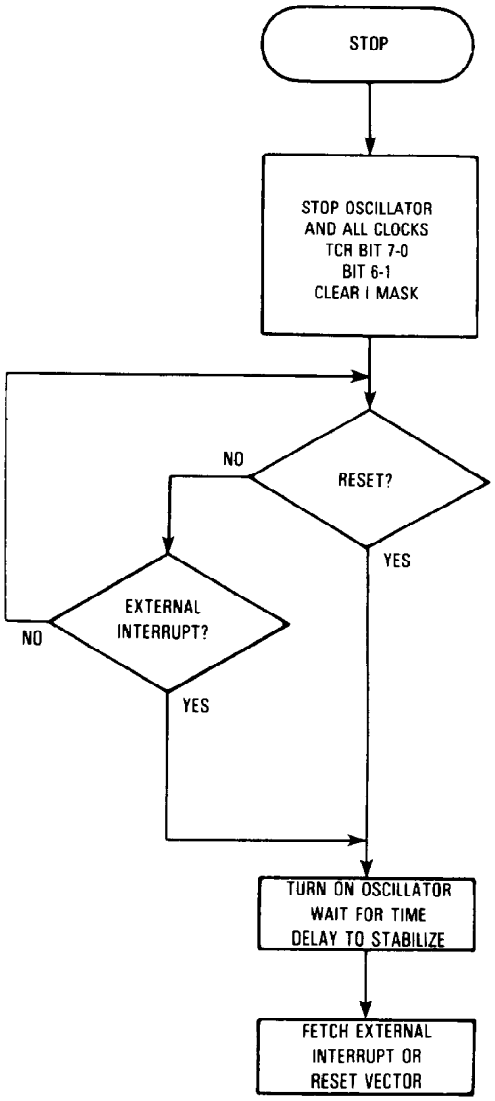


Figure 8. STOP Function Flowchart

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. 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 1; however, the TCR bit 3 always reads as a logic 0 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.

SOFTWARE CONTROLLED MODE

The timer prescaler input can be configured for three different operating modes plus a disable mode, depending on the value written to TCR control bits 4 and 5 (TCR4 and TCR5). The following paragraphs describe the different modes.

Timer Input Mode 1

When TCR4 and TCR5 are both programmed to zero, the timer input is from the internal clock (phase 2) and the timer input pin is disabled. The internal clock mode can be used for periodic interrupt generation as well as a reference for frequency and event measurement. During the WAIT instruction, the internal clock to the timer continues to run at its normal rate.

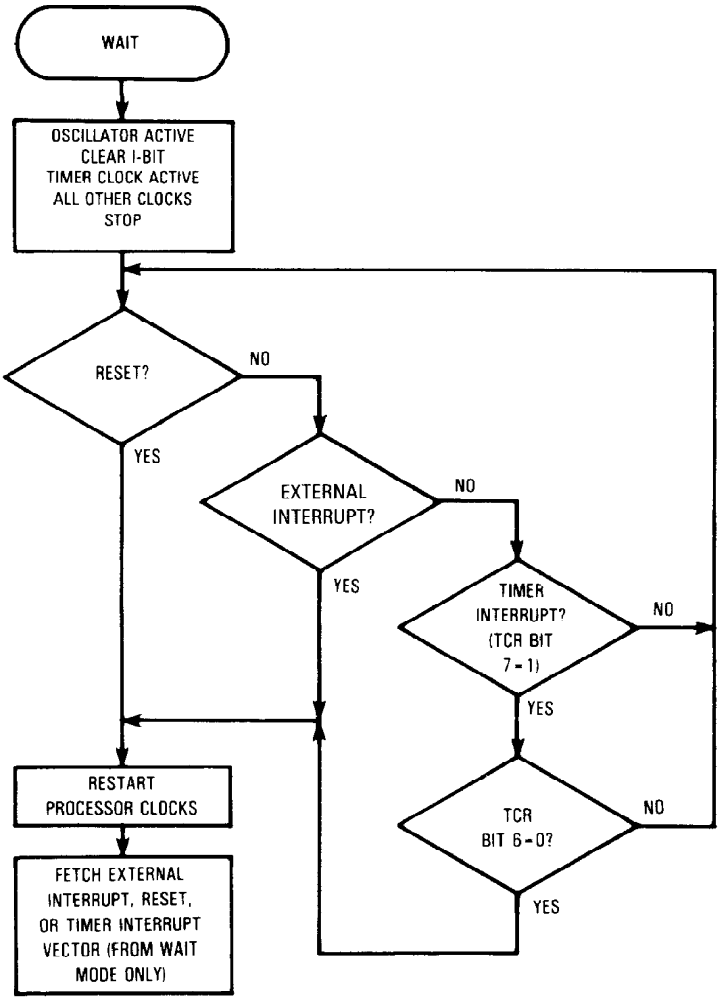
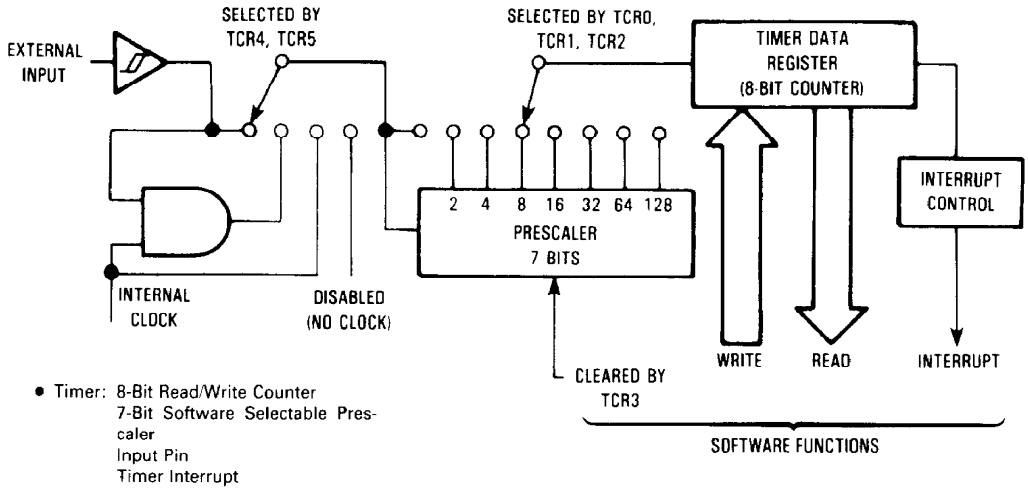


Figure 9. WAIT Function Flowchart



- Timer: 8-Bit Read/Write Counter
  7-Bit Software Selectable Prescaler
  Input Pin
  Timer Interrupt

NOTES:
1. Prescaler and timer data register are clocked on the falling edge of the internal clock or external input.
2. The timer data register counts down continuously.

Figure 10. Timer Block Diagram

Timer Input Mode 2

When TCR4 = 1 and TCR5 = 0, the internal clock and the timer input signals are ANDed to form the timer input. This mode can be used to measure external pulse widths. The external pulse gates in the internal clock for the duration of the external pulse. The accuracy of the count is ±1.

Timer Input Mode 3

When TCR4 = 0 and TCR5 = 1, no prescaler input frequency is applied to the prescaler and the timer is disabled.

Timer Input Mode 4

When TCR4 and TCR5 are both one, the timer input is from the external clock. The external clock can be used to count external events as well as to provide an external frequency for generating periodic interrupts.

TIMER CONTROL REGISTER (TCR) $009

This is an 8-bit register that controls various functions such as configuring operation mode, setting ratio of the prescaler, and generating timer interrupt request signal. All bits are read/write except bit 3.

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0
RESET:							
0	1	U	U	U	U	U	U

TCR7 — 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.

TCR6 — Timer Interrupt Mask
Used to inhibit the timer interrupt.
1 = Interrupt inhibited.
0 = Interrupt enabled.

TCR5 — External or Internal
Selects input clock source.
1 = External clock selected.
0 = Internal clock selected ($f_{OSC}/4$).

TCR4 — TIMER External Enable
Used to enable external TIMER pin or to enable the internal clock.
1 = Enables external timer pin.
0 = Disables external timer pin.

TCR3 — Prescaler Clear
Write only bit. Writing a 1 to this bit resets the prescaler to zero. A read of this location always indicates a zero.

TCR2, TCR1, TCR0 — Prescaler select bits
Decoded to select one of eight outputs of the prescaler.

Prescaler

TCR2	TCR1	TCR0	Result
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

INSTRUCTION SET

The MCU has a set of 61 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 listing.

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

READ-MODIFY-WRITE INSTRUCTIONS

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

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

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any 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 listing of 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 listing of control instructions.

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

OPCODE MAP SUMMARY

Table 2 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 immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT

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

EXTENDED

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

RELATIVE

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

INDEX, 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 2-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 3-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode. 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 2-byte instruction.

CAUTION

The corresponding DDRs for ports, A and B are write only registers (registers at $005 and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write 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 3-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 (Voltages Referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +6.0	V
All Input Voltages except OSC1	V_{in}	V_{SS} −0.5 to V_{DD} +0.5	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range MC146805F2 MC146805F2C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation it is recommended the V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs except OSC2 and Vpp are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Cerdip Quad Package	θ_{JA}	 115 65 TBD	°C/W

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

Characteristic	Symbol	Min	Max	Unit
Output Voltage, $I_{Load} \leq 10.0$ µA	V_{OL} V_{OH}	— V_{DD} −0.1	0.1 —	V
Output High Voltage (I_{Load} = −200 µA) PA0-PA7, PB0-PB7	V_{OH}	4.1	—	V
Output Low Voltage (I_{Load} = 800 µA) PA0-PA7, PB0-PB7	V_{OL}	—	0.4	V
Input High Voltage Ports PA0-PA7, PB0-PB7, PC0-PC3 TIMER, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	 V_{DD} −2.0 V_{DD} −0.8	 V_{DD} V_{DD}	V
Input Low Voltage (All Inputs)	V_{IL}	V_{SS}	0.8	V
Total Supply Current (C_L = 50 pF on Ports, no dc Loads, t_{cyc} = 1 µs) RUN (V_{IL} = 0.2 V, $V_{IH} = V_{DD}$ − 0.2 V) WAIT (See Note) STOP (See Note)	I_{DD}	 — — —	 4 1.5 150	 mA mA µA
I/O Ports Input Leakage — PA0-PA7, PB0-PB7	I_{IL}	—	±10	µA
Input Current — $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1, PC0-PC3	I_{in}	—	±1	µA
Output Capacitance — Ports A and B	C_{out}	—	12	pF
Input Capacitance — $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1, PC0-PC3	C_{in}	—	8	pF

NOTE:
Test conditions for I_{DD} are as follows:
All ports programmed as inputs
V_{IL} = 0.2 V (PA0-PA7, PB0-PB7, PC0-PC3)
$V_{IH} = V_{DD}$ − 0.2 V for $\overline{RESET}$, $\overline{IRQ}$, and TIMER
OSC1 input is a square wave from 0.2 V to V_{DD} − 0.2 V (for WAIT I_{DD} measurement only)
OSC2 output load = 20 pF (WAIT I_{DD} is affected linearly by the OSC2 capacitance)

Table 8. Control Timing (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0, $T_A = T_L$ to T_H, f_{osc} = 4 MHz, t_{cyc} = 1 µs)

Characteristic	Symbol	Min	Max	Unit
Crystal Oscillator Startup Time (see Figure 15)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time — Crystal Oscillator (See Figure 16)	t_{ILCH}	—	100	ms
Timer Pulse Width (see Figure 14)	t_{TH}, t_{TL}	0.5	—	t_{cyc}
$\overline{\text{RESET}}$ Pulse Width (see Figure 15)	t_{RL}	1.5	—	t_{cyc}
Timer Period (see Figure 14)	t_{TLTL}	1.0	—	t_{cyc}
Interrupt Pulse Width Low (see Figure 7)	t_{ILIH}	1.0	—	t_{cyc}
Interrupt Pulse Period (see Figure 7)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	100	—	ns
Cycle Time	t_{cyc}	1000	—	ns
Frequency of Operation Crystal External Clock	f_{osc}	— dc	4.0 4.0	MHz

*The minimum period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routines plus 20 t_{cyc} cycles.

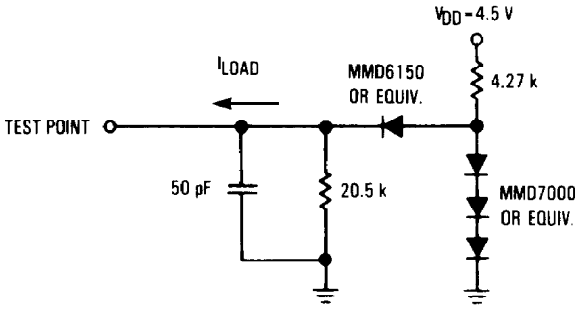


Figure 11. Equivalent Test Load

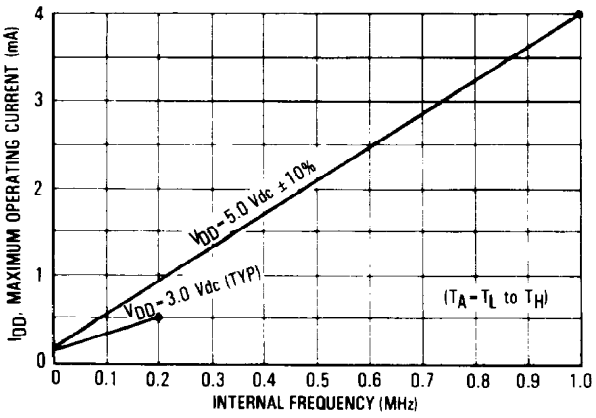


Figure 12. Maximum Operating Current vs Internal Frequency ($T_A = T_L$ to T_H)

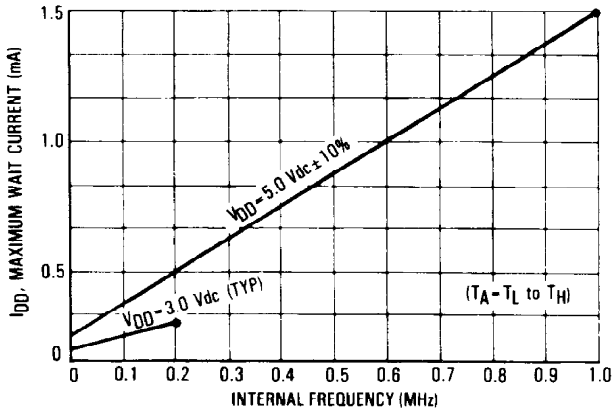


Figure 13. Maximum Wait Current vs Internal Frequency ($T_A = T_L$ to T_H)

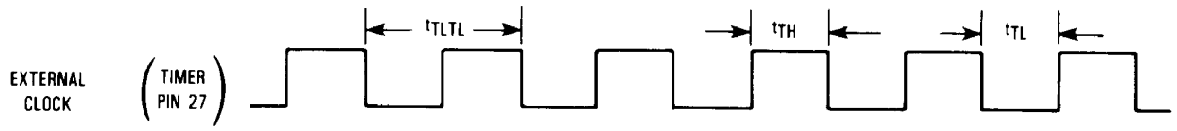


Figure 14. Timer Relationships

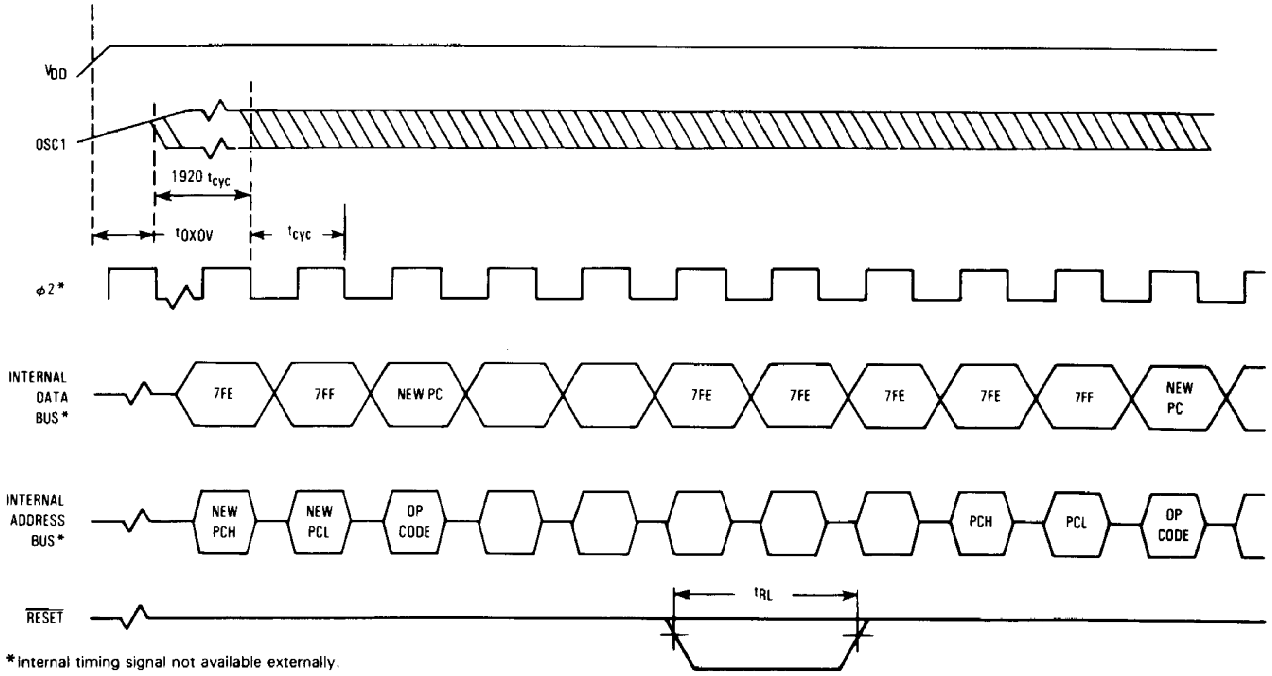
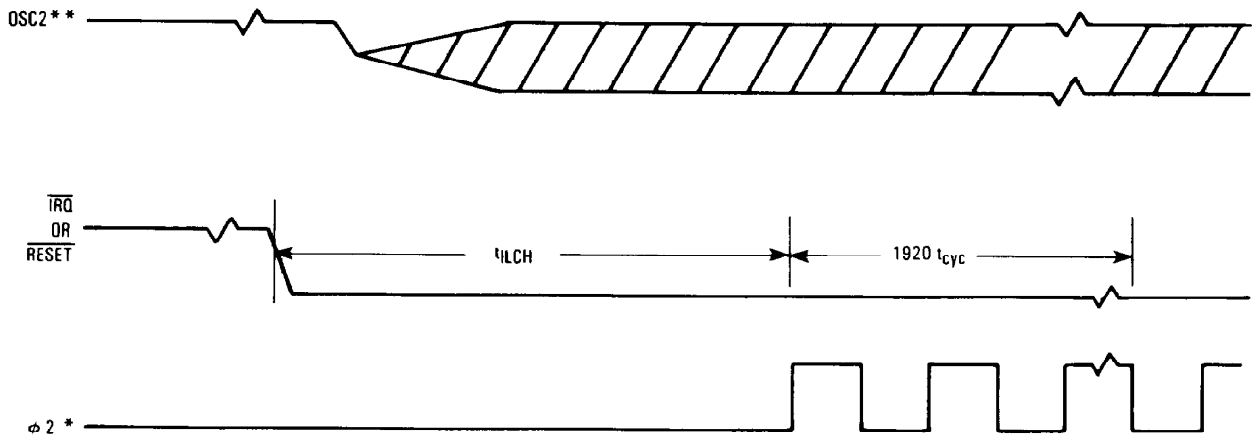


Figure 15. Power-On Reset and $\overline{\text{RESET}}$



*Internal timing signals not available externally.
**Represents the internal gating of the OSC1 input pin.

Figure 16. Stop Recovery

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)MC1468705F2, 2716, or 2516

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

NOTE

The low cost resistor oscillator option is not available on B54F mask.

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 customer's 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 expedite 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-sided, 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. It is necessary to 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

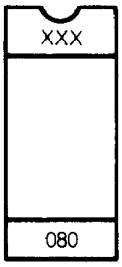
A MC1468705F2, 2716, or 2516 type EPROM, programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one of these EPROM devices, the EPROM must be programmed as described in the following paragraphs.

The program space ROM must start at EPROM address $080. If the customer program starts at any other address,

MDOS is a trademark of Motorola Inc.
MS-DOS is a trademark of Microsoft, Inc.
EXORciser® is a registered trademark of Motorola Inc.

the EPROM must be marked accordingly. All unused bytes, including the user's space, must be set at zero. For shipment to Motorola, EPROMs should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.

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 should be 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 disks from the data file used to create the custom mask.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency, the MCUs are usually unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with a 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 order numbers for the MC146805F2.

Table 3. Generic Information

Package Type	Frequency (MHz)	Temperature	Order Number
Plastic	4.0	0° to 70°C	MC146805F2P
P Suffix	4.0	−40° to +85°C	MC146805F2CP
PLCC	4.0	0° to 70°C	MC146805F2FN
FN Suffix	4.0	−40° to +85°C	MC146805F2CFN
Cerdip	4.0	0° to 70°C	MC146805F2S
S Suffix	4.0	−40° to +85°C	MC146805F2CS

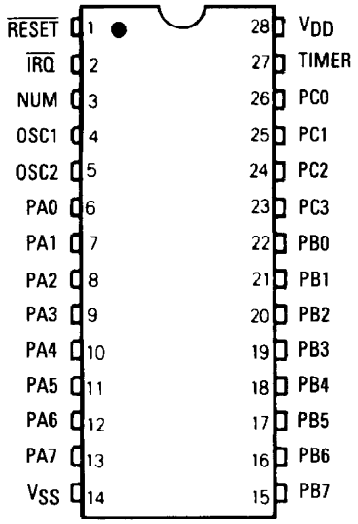
IBM is a registered trademark of International Business Machines Corporation.

MECHANICAL DATA

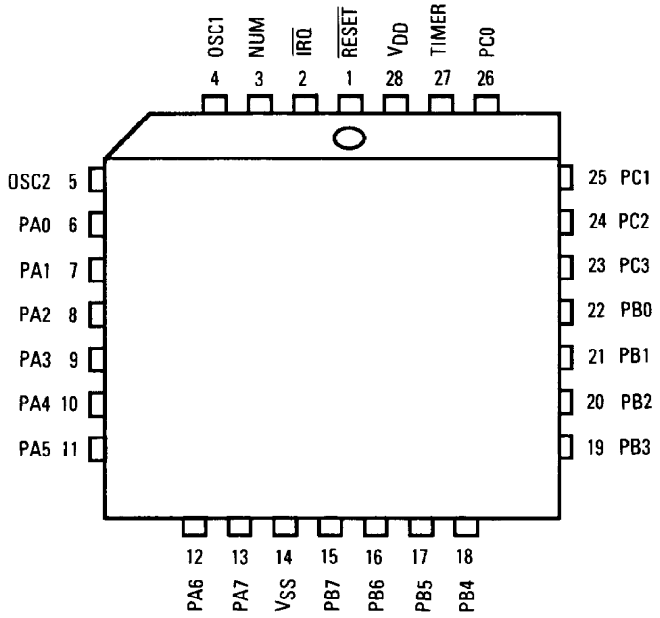
This section contains the pin assignments and package dimensions for the MC146805F2.

PIN ASSIGNMENTS

28-Pin Dual-in-Line Package

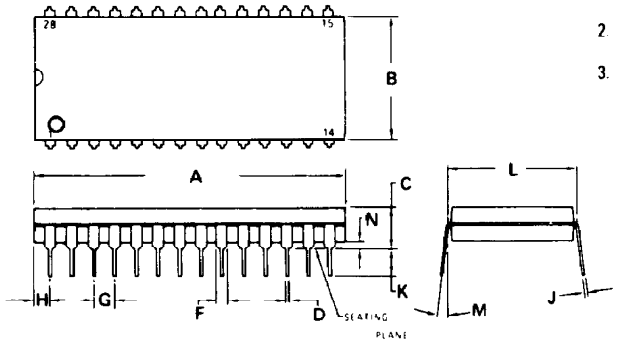


Quad Pin Out



PACKAGE DIMENSIONS

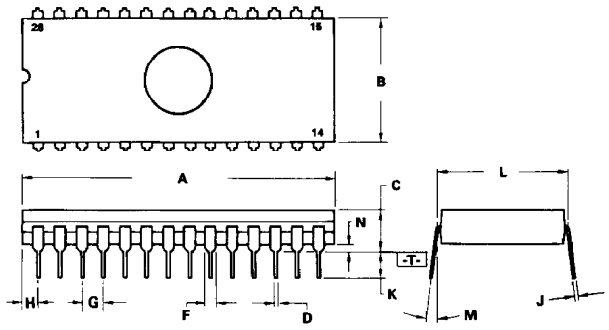
P SUFFIX
PLASTIC PACKAGE
CASE 710-02



NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25mm(0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	36.45	37.21	1.435	1.465
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0^0	15^0	0^0	15^0
N	0.51	1.02	0.020	0.040

CERDIP PACKAGE
CASE 733A-01

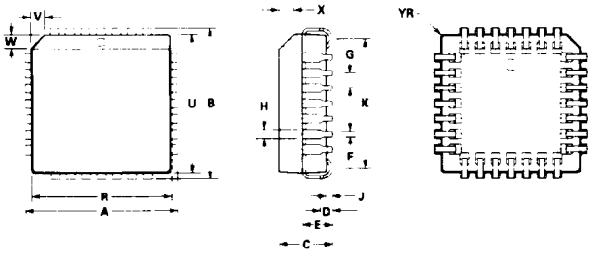


NOTES:
1. DIMENSION "A" IS A DATUM. T IS BOTH A DATUM AND A SEATING PLANE.
2. POSITIONAL TOLERANCE FOR LEADS: (28 PLACES) ⌖ | ϕ 0.25 (0.010) Ⓜ | T | A Ⓜ
3. DIMENSIONS "A" & B INCLUDE MENISCUS.
4. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
6. CONTROLLING DIMENSION: INCH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	36.45	37.84	1.435	1.490
B	12.70	15.36	0.500	0.605
C	4.06	6.09	0.160	0.240
D	0.38	0.55	0.015	0.022
F	1.27	1.65	0.050	0.065
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	3.17	4.06	0.125	0.160
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.27	0.020	0.050

CASE 733A-01

FN SUFFIX
PLASTIC LEADED CHIP CARRIER PACKAGE
CASE 776-01



NOTES:
1. DIMENSIONS R AND U DO NOT INCLUDE MOLD FLASH.
2. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
3. CONTROLLING DIMENSION: INCH

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	12.32	12.57	0.485	0.495
B	12.32	12.57	0.485	0.495
C	4.19	4.57	0.165	0.180
D	0.64	1.01	0.025	0.040
E	2.16	2.79	0.085	0.110
F	0.33	0.53	0.013	0.021
G	1.27 BSC		0.050 BSC	
H	0.66	0.81	0.026	0.032
J	0.38	0.63	0.015	0.025
K	9.91	10.92	0.390	0.430
R	11.43	11.58	0.450	0.456
U	11.43	11.58	0.450	0.456
V	1.07	1.21	0.042	0.048
W	1.07	1.21	0.042	0.048
X	1.07	1.42	0.042	0.056
Y	0.00	0.50	0.000	0.020

Literature Distribution Centers:
USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Center; 88 Tanners Drive, Blakelands Milton Keynes, MK145BP, England.
ASIA PACIFIC: Motorola Semiconductors H.K. Ltd.; P.O. Box 80300; Cheung Sha Wan Post Office; Kowloon Hong Kong.

A21988 PRINTED IN USA 8-88 IMPERIAL LITHO C58833 18,000 YGAGAA