MOTOROLA

MC146805G2

Advance Information

CMOS
(HIGH-PERFORMANCE SILICON-GATE)

8-BIT MICROCOMPUTER

8-BIT MICROCOMPUTER UNIT

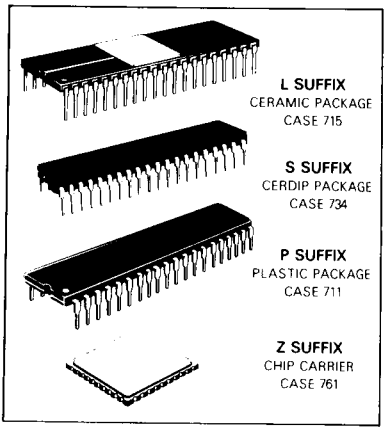
The MC146805G2 Microcomputer Unit (MCU) belongs to the M146805 CMOS Family of Microcomputers. This 8-bit MCU contains on-chip oscillator, CPU, RAM, ROM, I/O, and TIMER. The fully static design allows operation at frequencies down to dc, further reducing its already low-power consumption. It is a low-power processor designed for low-end to mid-range applications in the consumer, automotive, industrial, and communications markets where very low power consumption constitutes an important factor. The following are the major features of the MC146805G2 MCU.

HARDWARE FEATURES

- Typical Full Speed Operating Power of 12 mW at 5 V
- Typical WAIT Mode Power of 4 mW
- Typical STOP Mode Power of 5 μW
- Fully Static Operation
- 112 Bytes of On-Chip RAM
- 2106 Bytes of On-Chip ROM
- 32 Bidirectional I/O Lines
- High Current Drive
- Internal 8-Bit Timer with Software Programmable 7-Bit Prescaler
- External Timer Input
- External and Timer Interrupts
- Self-Check Mode
- Master Reset and Power-On Reset
- Single 3 to 6 Volt Supply
- On-Chip Oscillator with RC or Crystal Mask Options
- 40-Pin Dual-In-Line Package
- Chip Carrier Also Available

SOFTWARE FEATURES

- Similar to the MC6800
- Efficient Use of Program Space
- Versatile Interrupt Handling
- True Bit Manipulation
- Addressing Modes with Indexed Addressing for Tables
- Efficient Instruction Set
- Memory Mapped I/O
- Most Self-Check Routines User Callable
- Two Power Saving Standby Modes



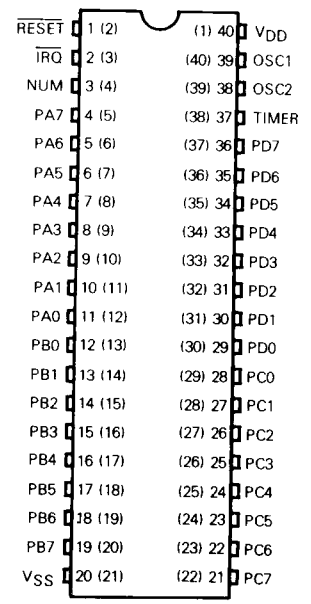
L SUFFIX
CERAMIC PACKAGE
CASE 715

S SUFFIX
CERDIP PACKAGE
CASE 734

P SUFFIX
PLASTIC PACKAGE
CASE 711

Z SUFFIX
CHIP CARRIER
CASE 761

PIN ASSIGNMENT



Pin numbers in parentheses represent equivalent Z suffix chip carrier pins.

GENERIC INFORMATION

Package Type	Frequency (MHz)	Temperature	Generic Number
Ceramic	1.0	0°C to 70°C	MC146905G2L
L Suffix	1.0	−40°C to 85°C	MC146805G2CL
Cerdip	1.0	0°C to 70°C	MC146805G2S
S Suffix	1.0	−40°C to 85°C	MC146805G2CS
Plastic	1.0	0°C to 70°C	MC146805G2P
P Suffix	1.0	−40°C to 85°C	MC146805G2CP
Leadless Chip Carrier	1.0	0°C to 70°C	MC 146805G2Z
Z Suffix	1.0	−40°C to 85°C	MC146805G2CZ

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

MAXIMUM RATINGS (Voltages Referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_D	−0.3 to +8.0	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 MC146805G2 MC146805G2C	T_A	T_L to T_H 0 to 70 −40 to 85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Current Drain Total (PD4-PD7 only)	I_{OH}	40	mA

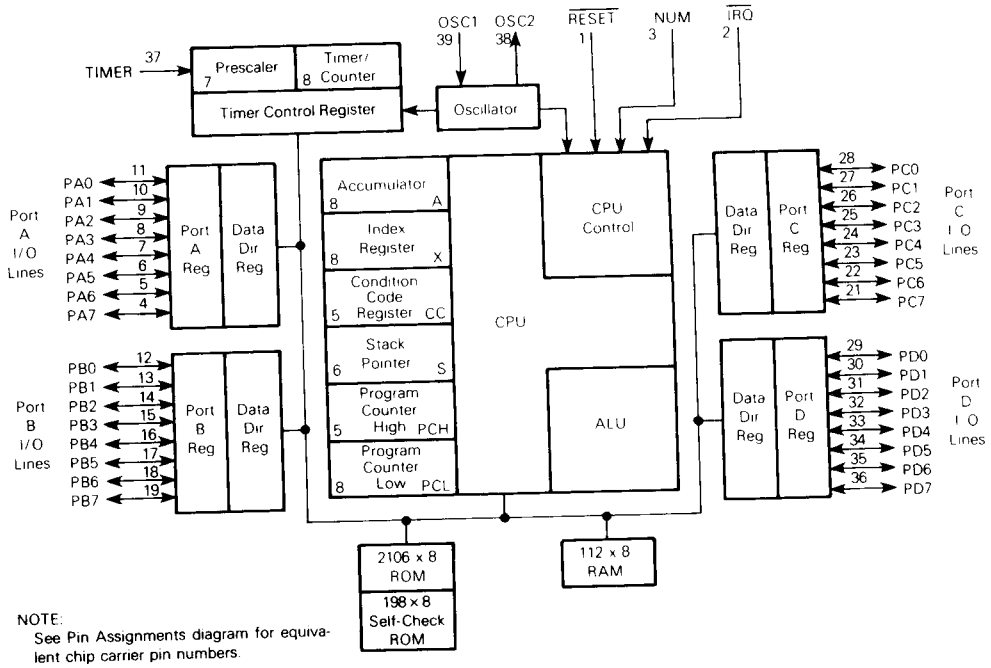
THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Unit
Thermal Resistance Plastic Cerdip Ceramic Chip Carrier	θ_{JA}	 100 60 50 100	°C/W

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

3

FIGURE 1 – MC146805G2 CMOS MICROCOMPUTER



NOTE:
See Pin Assignments diagram for equivalent chip carrier pin numbers.

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage $I_{Load} \leq 10.0\ \mu A$	V_{OL}	–	0.1	V
	V_{OH}	$V_{DD} - 0.1$	–	V
Output High Voltage				
(I_{Load} = – 100 μA) PB0-PB7, PC0-PC7	V_{OH}	2.4	–	V
(I_{Load} = – 2 mA) PA0-PA7, PD0-PD3	V_{OH}	2.4	–	V
(I_{Load} = – 8 mA) PD4-PD7	V_{OH}	2.4	–	V
Output Low Voltage				
(I_{Load} = 800 μA) All Ports	V_{OL}	–	0.4	V
PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7				
Input High Voltage				
Ports PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	V_{IH}	$V_{DD} - 2.0$	V_{DD}	V
TIMER, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	$V_{DD} - 0.8$	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)	I_{DD}	–	4	mA
$\overline{WAIT}$ (See Note)	I_{DD}	–	1.5	mA
$\overline{STOP}$ (See Note)	I_{DD}	–	150	μA
I/O Ports Input Leakage				
PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	I_{IL}	–	± 10	μA
Input Current				
$\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	I_{in}	–	± 1	μA
Capacitance				
Ports	C_{out}	–	12	pF
$\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	C_{in}	–	8	pF

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 3.0 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage $I_{Load} \leq 1.0\ \mu A$	V_{OL}	–	0.1	V
	V_{OH}	$V_{DD} - 0.1$	–	V
Output High Voltage				
(I_{Load} = – 50 μA) PB0-PB7, PC0-PC7	V_{OH}	1.4	–	V
(I_{Load} = – 0.5 mA) PA0-PA7, PD0-PD3	V_{OH}	1.4	–	V
(I_{Load} = – 2 mA) PD4-PD7	V_{OH}	1.4	–	V
Output Low Voltage				
(I_{Load} = 300 μA) All Ports PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	V_{OL}	–	0.3	V
Input High Voltage				
Ports PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	V_{IH}	2.7	V_{DD}	V
TIMER, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	2.7	V_{DD}	V
Input Low Voltage All Inputs	V_{IL}	V_{SS}	0.3	V
Total Supply Current (no dc Loads, t_{cyc} = 5 μs)				
RUN (V_{IL} = 0.1 V, $V_{IH} = V_{DD} - 0.1$ V)	I_{DD}	–	0.5	mA
$\overline{WAIT}$ (See Note)	I_{DD}	–	200	μA
$\overline{STOP}$ (See Note)	I_{DD}	–	100	μA
I/O Ports Input Leakage				
PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	I_{IL}	–	± 5	μA
Input Current				
$\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	I_{in}	–	± 1	μA
Capacitance				
Ports	C_{out}	–	12	pF
$\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	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-PC7, PD0-PD7)
$V_{IH} = V_{DD} - 0.2$ V for $\overline{RESET}$, $\overline{IRQ}$, TIMER
OSC1 input is a squarewave from 0.2 V to $V_{DD} - 0.2$ V
OSC2 output load = 20 pF (wait I_{DD} is affected linearly by the OSC2 capacitance)

TABLE 1 — CONTROL TIMING
(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0, T_A = 0° to 70°C, f_{osc} = 4 MHz)

Characteristics	Symbol	Min	Max	Unit
Crystal Oscillator Startup Time (Figure 11)	t_{OXOV}	–	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (Figure 12)	t_{ILCH}	–	100	ms
Timer Pulse Width (Figure 10)	t_{TH}, t_{TL}	0.5	–	t_{cyc}
Reset Pulse Width (Figure 11)	t_{RL}	1.5	–	t_{cyc}
Timer Period (Figure 10)	t_{TLTL}	1.0	–	t_{cyc}
Interrupt Pulse Width Low (Figure 21)	t_{ILIH}	1.0	–	t_{cyc}
Interrupt Pulse Period (Figure 21)	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} f_{osc}	– DC	4.0 4.0	MHz 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.

3

FIGURE 2 — EQUIVALENT TEST LOAD

Port	R_1	R_2
B and C	24.3 kΩ	4.32 kΩ
A, PD0-PD3	1.21 kΩ	3.1 kΩ
PD4-PD7	300 Ω	1.64 kΩ

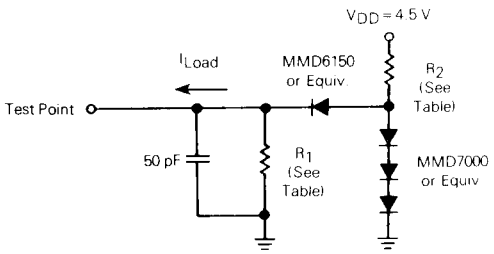


FIGURE 3 — TYPICAL OPERATING CURRENT vs INTERNAL FREQUENCY

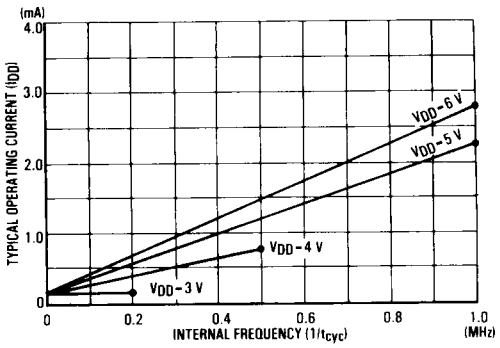


FIGURE 4 — MAXIMUM I_{DD} vs FREQUENCY

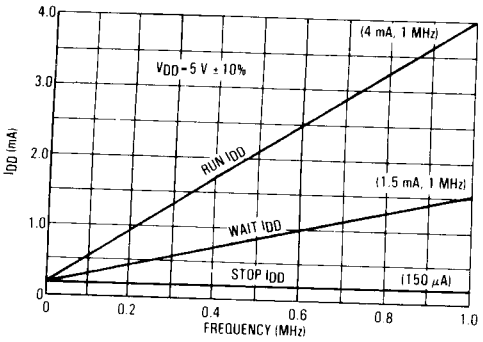


FIGURE 5 — MAXIMUM I_{DD} vs FREQUENCY

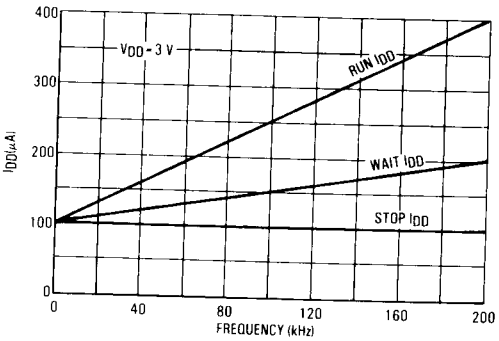


FIGURE 6 — MINIMUM I_{OH}, PORT D PINS 33-36

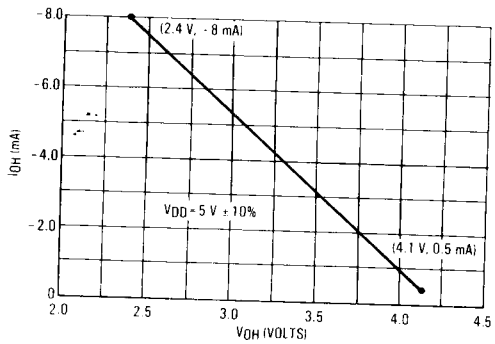


FIGURE 7 — MINIMUM I_{OH}, PORT C

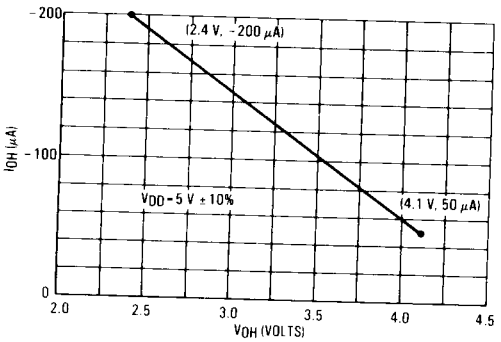


FIGURE 8 — MINIMUM I_{OH}, PORT A AND 8

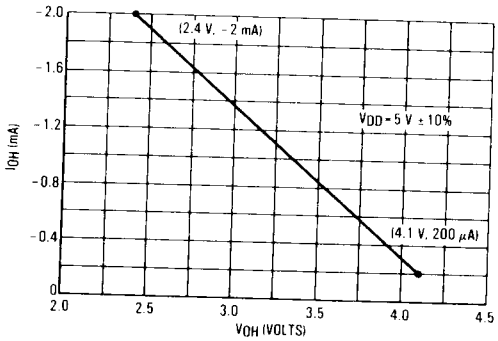


FIGURE 9 — MINIMUM I_{OL}, ALL PORTS

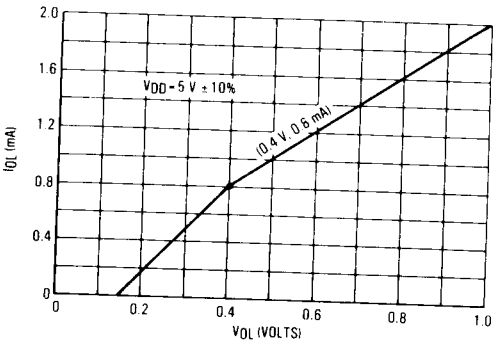


FIGURE 10 — TIMER RELATIONSHIPS

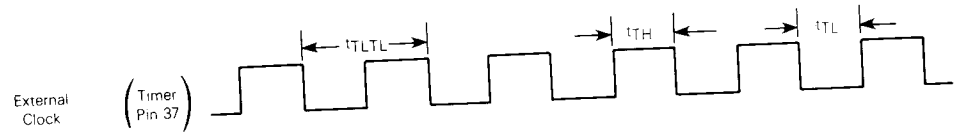
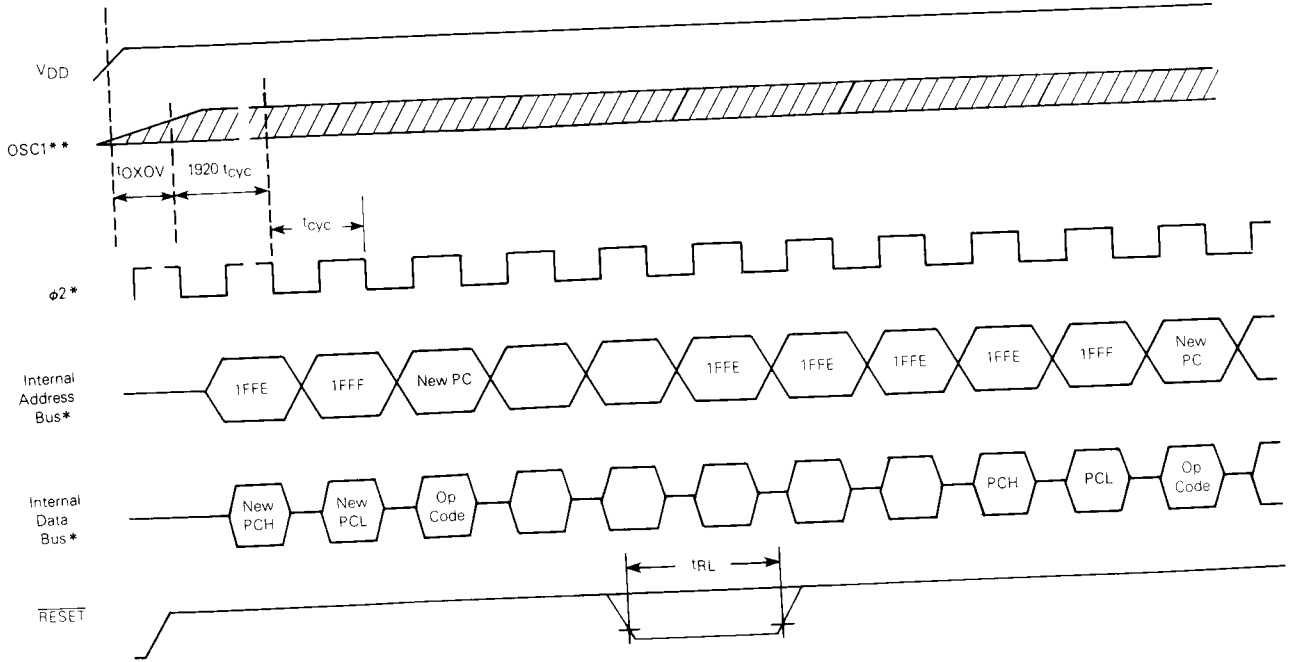
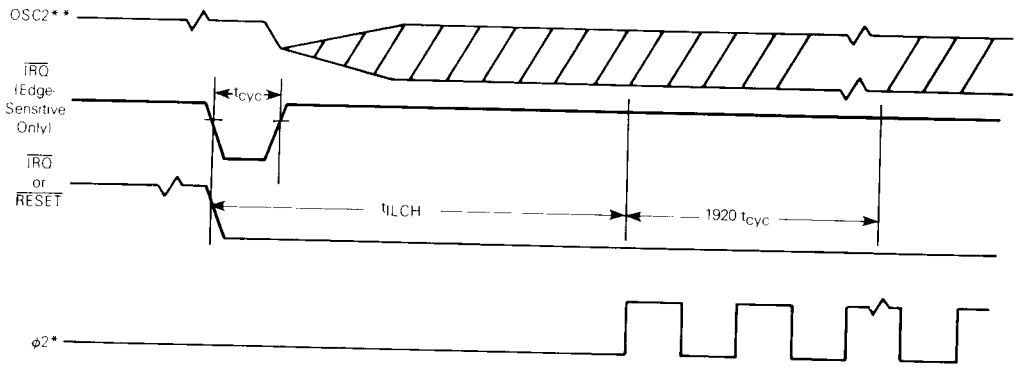


FIGURE 11 — POWER-ON RESET AND $\overline{\text{RESET}}$



*Internal timing signal and bus information not available externally.
**OSC1 line is not meant to represent frequency. It is only used to represent time.

FIGURE 12 — STOP RECOVERY AND POWER-ON RESET



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

FUNCTIONAL PIN DESCRIPTION

V_{DD} and V_{SS}

Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is ground.

IRQ (MASKABLE INTERRUPT REQUEST)

IRQ is mask option selectable with the choice of interrupt sensitivity being both level-sensitive, and negative edge-sensitive or negative edge-sensitive only. The MCU completes the current instruction before it responds to the request. If IRQ is low and the interrupt mask bit (I bit) in the condition code register is clear, the MCU begins an interrupt sequence at the end of the current instruction.

If the mask option is selected to include level sensitivity, then the IRQ input requires an external resistor to V_{DD} for "wire-OR" operation. See INTERRUPTS for more detail.

RESET

The RESET input is not required for start-up but can be used to reset the MCU's internal state and provide an orderly software start-up procedure. Refer to RESETS for a detailed description.

TIMER

The TIMER input may be used as an external clock for the on-chip timer. Refer to TIMER for additional information about the timer circuitry.

NUM — NON-USER MODE

This pin is intended for use in self-check only. In user applications, connect this pin to ground through a 10 kΩ resistor.

OSC1, OSC2

The MC146805G2 can be configured to accept either a crystal input or an RC network to control the internal oscillator. Additionally, the internal clocks can be derived by either a divide-by-two or divide-by-four of the internal oscillator output frequency (f_{osc}). Both of these options are mask selectable.

RC — If the RC oscillator option is selected, then a resistor is connected to the oscillator pins as shown in Figure 13(d). The relationship between R and f_{osc} is shown in Figure 14.

CRYSTAL — The circuit shown in Figure 13(b) is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{osc} in the electrical characteristics table. Using an external CMOS oscillator is suggested when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Crystal frequency limits are also affected by V_{DD}. Refer to Control Timing Characteristics for limits. See Table 1.

EXTERNAL CLOCK — An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 13(c). An external clock should be used with the crystal oscillator mask option only. The t_{OXOV} or t_{ILCH} specifications do not apply when using an external clock input.

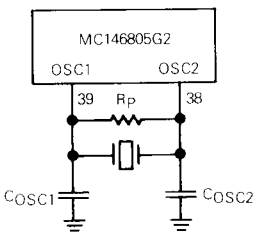
PA0-PA7

These eight I/O lines comprise Port A. The state of any pin is software programmable. Refer to INPUT/OUTPUT PROGRAMMING for a description of I/O programming.

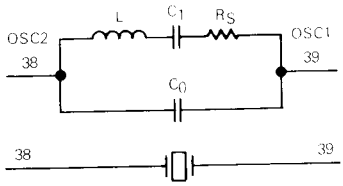
FIGURE 13 — OSCILLATOR CONNECTIONS

	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	40	K

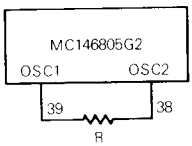
(a) Crystal Parameters



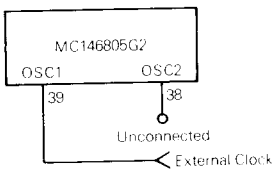
(b) Crystal Oscillator Connections



(c) Equivalent Crystal Circuit

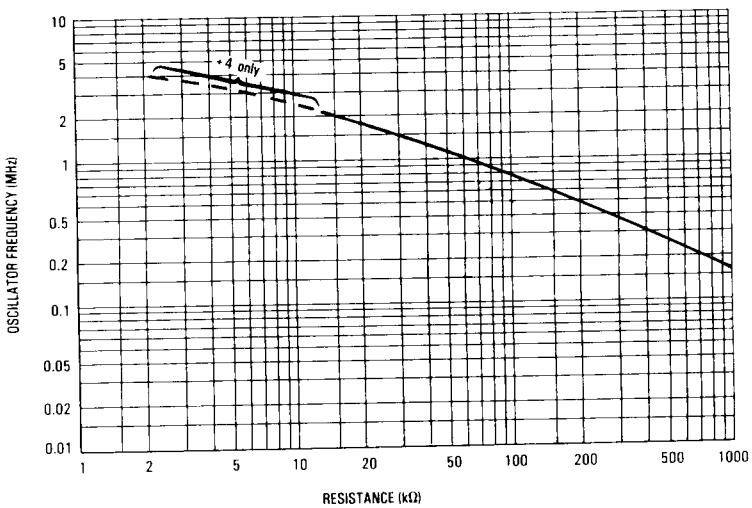


(d) RC Oscillator Connection



(e) External Clock Source Connections

FIGURE 14 — TYPICAL FREQUENCY vs RESISTANCE FOR RC OSCILLATOR OPTION ONLY



PB0-PB7

These eight lines comprise Port B. The state of any pin is software programmable. Refer to INPUT/OUTPUT PROGRAMMING for a description of I/O programming.

PC0-PC7

These eight lines comprise Port C. The state of any pin is software programmable. Refer to INPUT/OUTPUT PROGRAMMING for a description of I/O programming.

PD0-PD7

These eight lines comprise Port D. PD4-PD7 also are capable of driving LEDs directly. The state of any pin is software programmable. Refer to INPUT/OUTPUT PROGRAMMING for a description of I/O programming.

INPUT/OUTPUT PROGRAMMING

Any port pin may be software programmed as an input or output by the state of the corresponding bit in the port data direction register (DDR). A pin is configured as an output if its corresponding DDR bit is set to a logic 1. A pin is configured as an input if its corresponding DDR bit is cleared to a logic 0. At reset, all DDRs are cleared, which configures all port pins as inputs. A port pin configured as an output will output the data in the corresponding bit of its port data latch. Refer to Figure 15 and Table 2.

FIGURE 15 — TYPICAL PORT I/O CIRCUITRY

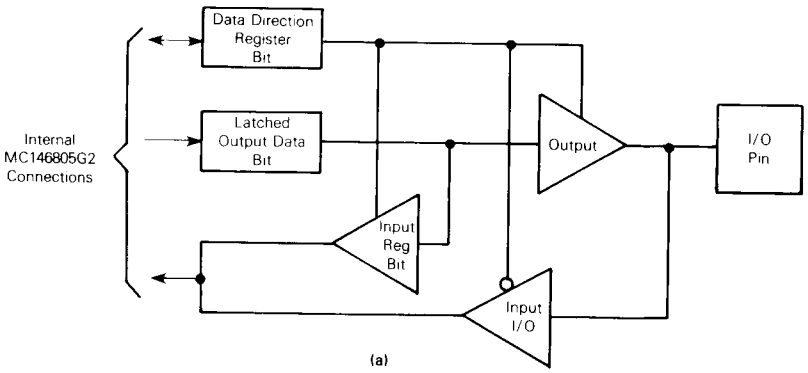


TABLE 2 — I/O PIN FUNCTIONS

R/$\overline{W}$*	DDR	I/O Pin Function
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/$\overline{W}$ is an internal signal.

SELF-CHECK

The MC146805G2 self-check is performed using the circuit in Figure 16. Self-check is initiated by connecting NUM and TIMER pins to a logic 1 then executing a reset. After reset, five subroutines are called that execute the following tests:

I/O – Functionally exercise ports A, B, C, D

RAM – Walking bit test

ROM – Exclusive OR with odd 1s parity result

Timer – Functionally exercise timer

Interrupts – Functionally exercise external and timer interrupts

Self-check results are shown in Table 3. The following subroutines are available to user programs and do not require any external hardware.

RAM SELF-CHECK SUBROUTINES

Returns with the Z bit clear if any error is detected; otherwise the Z bit is set.

The RAM test must be called with the stack pointer at $007F. When run, the test checks every RAM cell except for $007F and $007E which are assumed to contain the return address.

A and X are modified. All RAM locations except the top 2 are modified. (Enter at location $1F80.)

ROM CHECKSUM SUBROUTINE

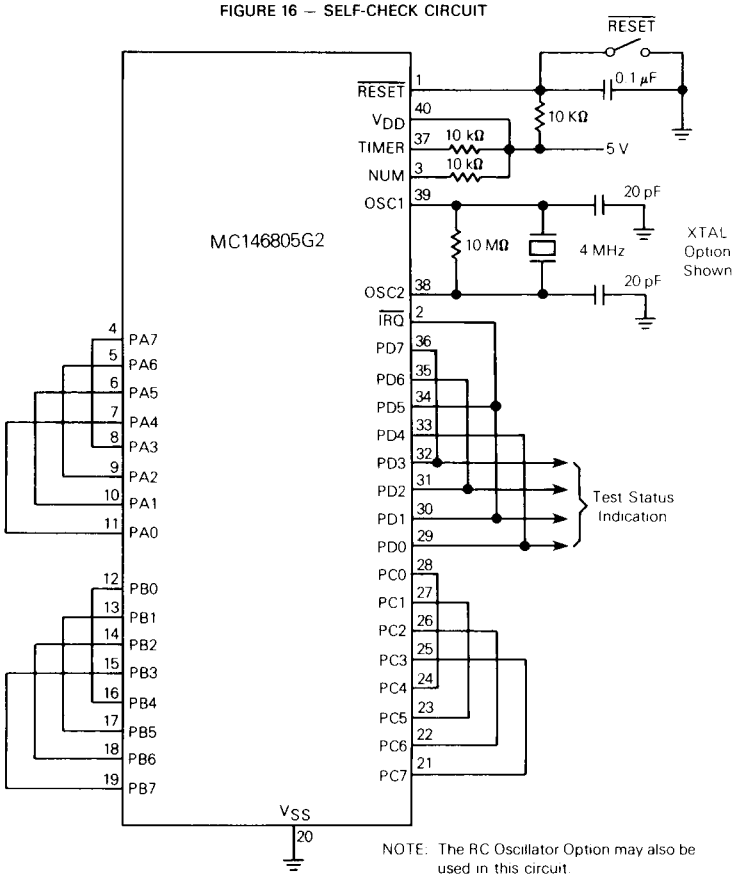
Returns with Z bit cleared if any error was found, otherwise Z = 1. X = 0 on return, and A is zero if the test passed. RAM locations $0040-$0043 are overwritten. (Enter at location $1F9B.)

TIMER TEST SUBROUTINE

Return with Z bit cleared if any error was found; otherwise Z = 1.

This routine runs a simple test on the timer. In order to work correctly as a user subroutine, the internal clock must be the clocking source and interrupts must be disabled. Also, on exit, the clock will be running and the interrupt mask not set so the caller must protect himself from interrupts if necessary.

FIGURE 16 – SELF-CHECK CIRCUIT

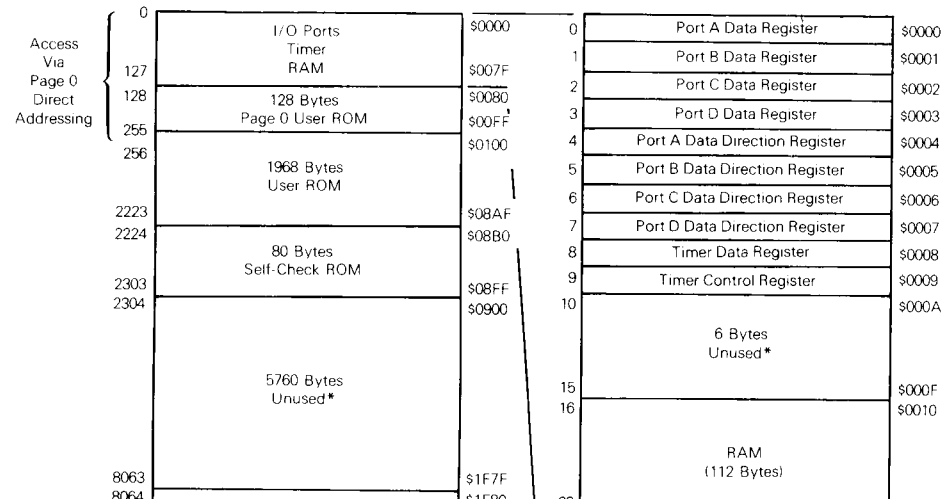


NOTE: The RC Oscillator Option may also be used in this circuit.

TABLE 3 — SELF-CHECK RESULTS

PD3	PD2	PD1	PD0	Remarks
1	0	1	0	Bad I/O
1	0	1	1	Bad Timer
1	1	0	0	Bad RAM
1	1	0	1	Bad ROM
1	1	1	0	Bad Interrupt or Request Flag
Cycling				Good Part
All Others				Bad Part

FIGURE 17 — ADDRESS MAP



*Reads of unused locations undefined.

A and X register contents are lost; this routine counts how many times the clock counts in 128 cycles. The number of counts should be a power of two since the prescaler is a power of two. If not, the timer probably is not counting correctly. The routine also detects if the timer is running at all. (Enter at location $1FB5.)

MEMORY

The MC146805G2 has a total address space of 8192 bytes of memory and I/O registers. The address space is shown in Figure 17.

The first 128 bytes of memory (first half of page zero) are comprised of the I/O port locations, timer locations, and 112 bytes of RAM. The next 2096 bytes (including the 128 bytes of the second half of page zero) comprise the user ROM. The 10 highest address bytes contain the reset and the interrupt vectors.

The stack pointer is used to address data stored on the stack. Data is stored on the stack during interrupts and subroutine calls. At power-up, the stack pointer is set to $007F and it is decremented as data is pushed on the stack. When data is removed from the stack, the stack pointer is incremented. A maximum of 64 bytes of RAM is available for stack usage. Since most programs use only a small part of the allocated stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are usable for program data storage.

REGISTERS

The MC146805G2 contains five registers, as shown in the programming model in Figure 18. The interrupt stacking order is shown in Figure 19.

FIGURE 18 — PROGRAMMING MODEL

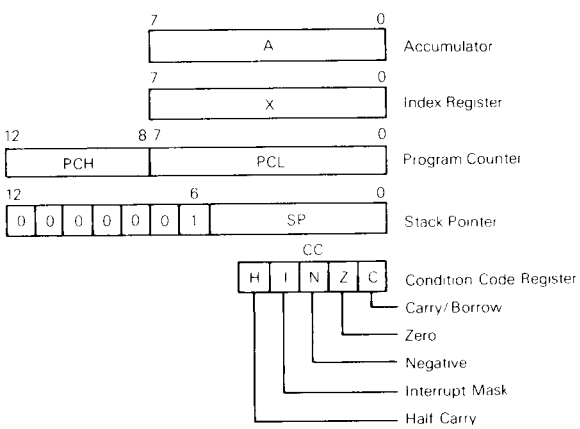
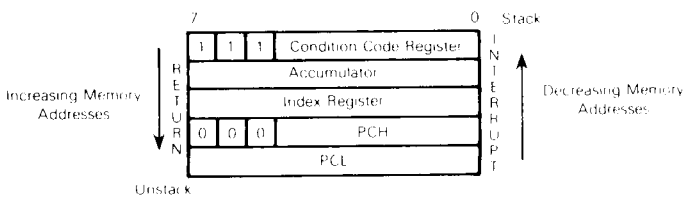


FIGURE 19 — STACKING ORDER



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.

ACCUMULATOR (A)

This accumulator is an 8-bit general purpose register used to hold operands, results of the arithmetic calculations, and data manipulations.

INDEX REGISTER (X)

The X register is an 8-bit register which is used during the indexed modes of addressing. It provides an 8-bit value which is used to create an effective address. The index register is also used for data manipulations with the read-modify-write type of instructions and as a temporary storage register when not performing addressing operations.

PROGRAM COUNTER (PC)

The program counter is a 13-bit register that contains the address of the next instruction to be executed by the processor.

STACK POINTER (SP)

The stack pointer is a 13-bit register containing the address of the next free location on the stack. When accessing memory, the seven most significant bits are permanently configured to 0000001. These seven bits are appended to the six least significant register bits to produce an address within the range of $007F to $0040. The stack area of RAM is used to store the return address on subroutine calls and the machine state during interrupts. During external or power-on reset, and during a reset stack pointer (RSP) instruction, the stack pointer is set to its upper limit ($007F). Nested interrupts and/or subroutines may use up to 64 (decimal) locations, beyond which the stack pointer wraps around and points to its upper limit thereby losing the previously stored information. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five RAM bytes.

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register which indicates the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each bit is explained in the following paragraphs.

HALF CARRY BITS (H) – The H bit is set to a one when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. The H bit is useful in binary coded decimal subroutines.

INTERRUPT MASK BIT (I) – When the I bit is set, both the external interrupt and the timer interrupt are disabled. Clearing this bit enables the above interrupts. If an interrupt occurs while the I bit is set, the interrupt is latched and is processed when the I bit is next cleared.

NEGATIVE (N) – When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is negative (bit 7 in the result is a logical 1).

ZERO (Z) – When set, this bit indicates that the result of the last arithmetic, logical, or data manipulations is 0.

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

RESETS

The MC146805G2 has two reset modes: an active low external reset pin ($\overline{\text{RESET}}$) and a power-on reset function; refer to Figure 11.

$\overline{\text{RESET}}$

The $\overline{\text{RESET}}$ input pin is used to reset the MCU to provide an orderly software start-up procedure. When using the external reset mode, the $\overline{\text{RESET}}$ pin must stay low for a minimum of one t_{CYC}. The $\overline{\text{RESET}}$ pin is provided with a Schmitt Trigger input (internally) to improve its noise immunity.

POWER-ON RESET

The power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuitry provides for a 1920 t_{CYC} delay from the time that the oscillator becomes active. If the external $\overline{\text{RESET}}$ pin is low at the end of the 1920 t_{CYC} time out, the processor remains in the reset condition.

Either of the two types of reset conditions causes the following to occur:

- Timer control register interrupt request bit TCR7 is cleared to a "0".
- Timer control register interrupt mask bit TCR6 is set to a "1".
- All data direction register bits are cleared to logical zeros. All ports are defined as inputs.
- Stack pointer is preset to $007F.
- The internal address bus is forced to the reset vector ($1FFE, $1FFF).
- Condition code register interrupt mask bit (I) is set to a "1".
- STOP and WAIT latches are cleared.
- External interrupt latch is cleared.

All other functions, such as other registers (including output ports), the timer, etc. are not cleared by the reset conditions.

INTERRUPTS

The MC146805G2 may be interrupted by one of three different methods: either one of two maskable hardware interrupts (external input or timer) or a nonmaskable software interrupt (SWI). Systems often require that normal processing be interrupted so that some external event may be serviced.

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 followed by a return to normal processing. The stack order is shown in Figure 19.

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution 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. Note that 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 and as such takes precedence over hardware interrupts only if the I bit is set (hardware interrupts masked). Refer to Figure 20 for the interrupt and instruction processing sequence.

Table 4 shows the execution priority of the $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$ and timer interrupts, and the software interrupt, SWI. Two conditions are shown, one with the I bit set and the other with I bit clear; however, in either case $\overline{\text{RESET}}$ has the highest priority of execution. If the I bit is set as per Table 4(a), the second highest priority is assigned to SWI. This is illustrated in Figure 20 which shows that the $\overline{\text{IRQ}}$ or Timer interrupts are not executed when the I bit is set and the next instruction (including SWI) is fetched. If the I bit is cleared as per Table 4(b), the priorities change in that the next instruction (including SWI) is not fetched until after the $\overline{\text{IRQ}}$ and Timer interrupts have been recognized (and serviced). Also, when the I bit is clear, if both $\overline{\text{IRQ}}$ and Timer interrupts are pending, the $\overline{\text{IRQ}}$ interrupt is always serviced before the Timer interrupt.

FIGURE 20 — $\overline{\text{RESET}}$ AND INTERRUPT PROCESSING FLOWCHART

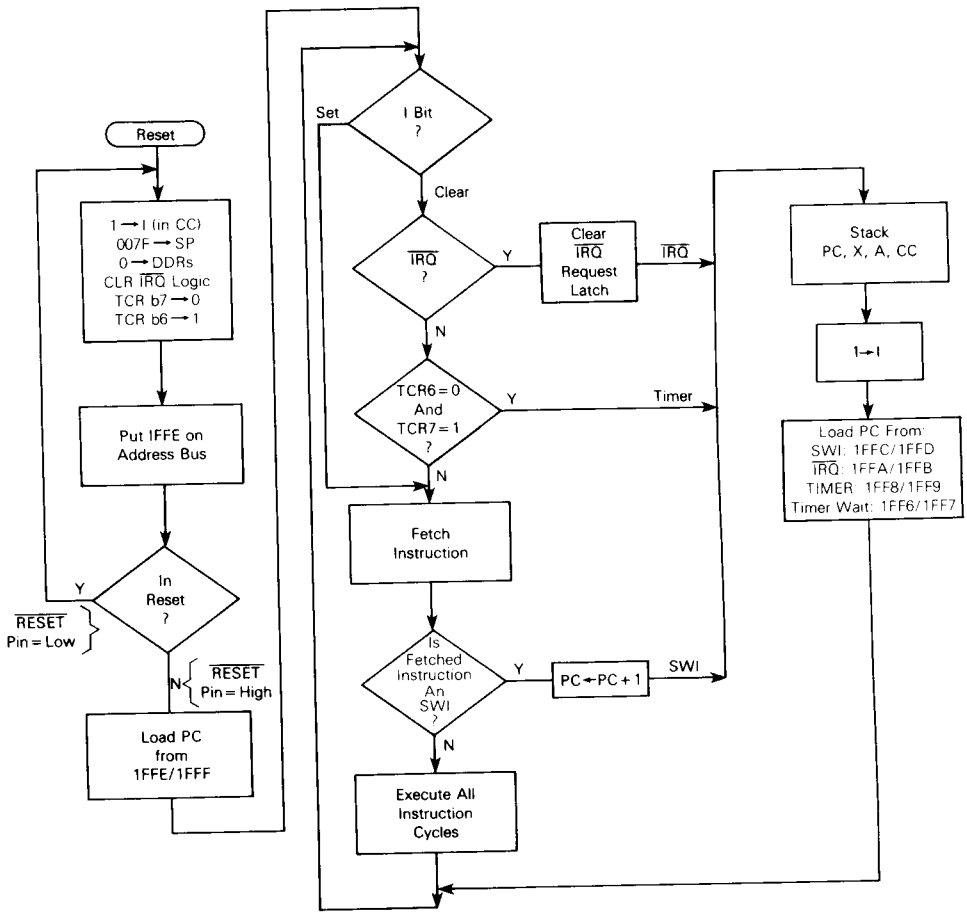


TABLE 4 — INTERRUPT/INSTRUCTION EXECUTION PRIORITY AND VECTOR ADDRESS

(a) I Bit Set

Interrupt/Instruction	Priority	Vector Address
$\overline{\text{RESET}}$	1	$1FFE-$1FFF
SWI	2	$1FFC-$1FFD

NOTE: $\overline{\text{IRQ}}$ and Timer Interrupts are not executed when the I bit is set, therefore, they are not shown.

(b) I Bit Clear

Interrupt/Instruction	Priority	Vector Address
$\overline{\text{RESET}}$	1	$1FFE-$1FFF
$\overline{\text{IRQ}}$	2	$1FFA-$1FFB
Timer	3	$1FF8-$1FF9 $1FF6-$1FF7*
SWI	4	$1FFC-$1FFD

* The Timer vector address from the WAIT mode is $1FF6-$1FF7

NOTE

Processing is such that at the end of the current instruction execution, the I bit is tested and if set the next instruction (including SWI) is fetched. If the I bit is cleared, the hardware interrupt latches are tested, and if no hardware interrupt is pending, the program falls through and the next instruction is fetched.

TIMER INTERRUPT

If the timer interrupt 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 is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the interrupt mask bit in the condition code register is set. This masks further interrupts until the present one is serviced. The processor now vectors to the timer interrupt service routine. The address for this service routine is specified by the contents of $1FF8 and $1FF9 unless the processor is in a WAIT mode in which case the contents of $1FF6 and $1FF7 specify the timer service routine address. Software must be used to clear the timer interrupt request bit (TCR7). 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

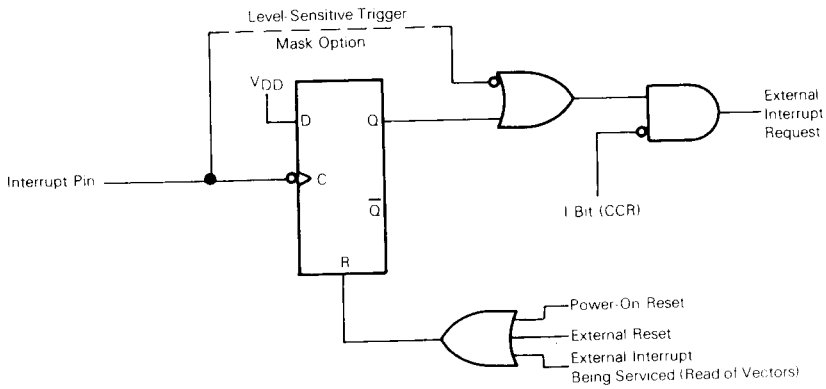
If the interrupt mask bit of the condition code register is cleared and the external interrupt pin ($\overline{\text{IRQ}}$) is low, then the external interrupt occurs. The action of the external interrupt is identical to the timer interrupt with the exception that the service routine address is specified by the contents of $1FFA and $1FFB. Either a level- and edge-sensitive trigger (or edge-sensitive only) are available as mask options. Figure 21 shows both a functional diagram and timing for the interrupt line. The timing diagram shows two different treatments of the interrupt line ($\overline{\text{IRQ}}$) to the processor. The first method is single pulses 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 routine. Once a pulse occurs, the next pulse should not occur until the MPU software has exited the routine (an RTI occurs). This time (t_{ILIL}) is obtained by adding 20 instruction cycles (t_{cyc}) to the total number of cycles it takes to complete the service routine including the RTI instruction; refer to Figure 21. The second configuration shows many interrupt lines "wire-ORed" to form the interrupts at the processor. Thus, if after servicing an interrupt the $\overline{\text{IRQ}}$ remains low, then the next interrupt is recognized.

SOFTWARE INTERRUPT (SWI)

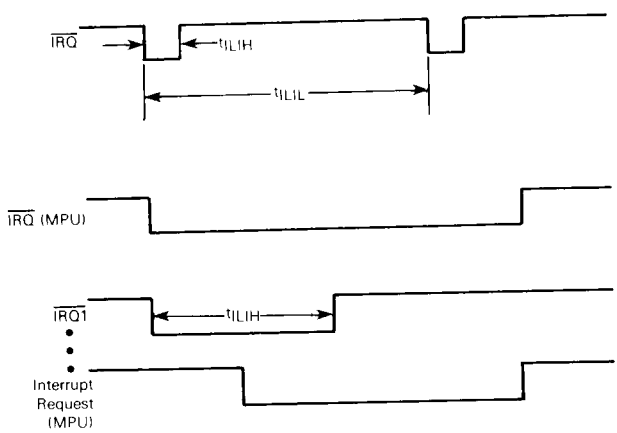
The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask in the condition code register. The service routine address is specified by the contents of memory locations $1FFC and $1FFD. See Figure 20 for interrupt and instruction processing flowchart.

FIGURE 21 – EXTERNAL INTERRUPT

(a) Interrupt Functional Diagram



(b) Interrupt Mode Diagram



Edge-Sensitive Trigger 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

Level-Sensitive Trigger Condition

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

Normally used with Wire-ORed Connection

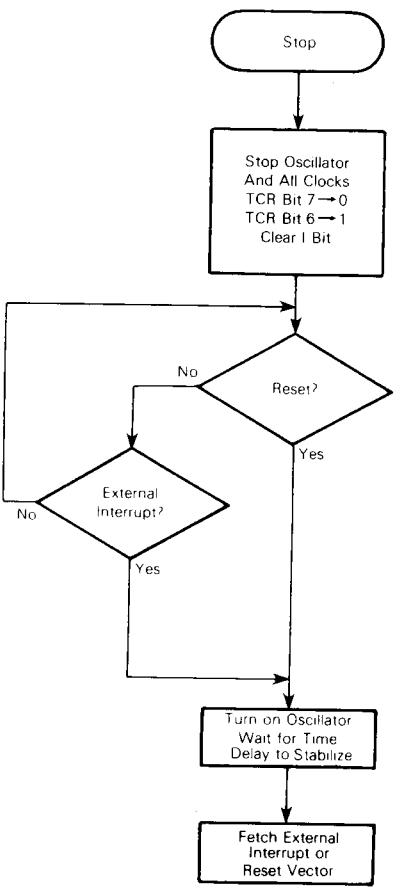
LOW-POWER MODES

STOP

The STOP instruction places the MC146805G2 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 22.

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt requests and to disable any further timer interrupts. The timer prescaler is cleared. External interrupts are enabled in the condition code register. All other registers and memory remain unaltered. All I/O lines remain unchanged.

FIGURE 22 — STOP FUNCTION FLOWCHART



WAIT

The WAIT instruction places the MC146805G2 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 the timer circuit; refer to Figure 23. Thus, all internal processing is halted; however, the timer continues to count normally.

During the WAIT mode, the I bit in the condition code register is cleared to enable interrupts. All other registers, memory, and I/O 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

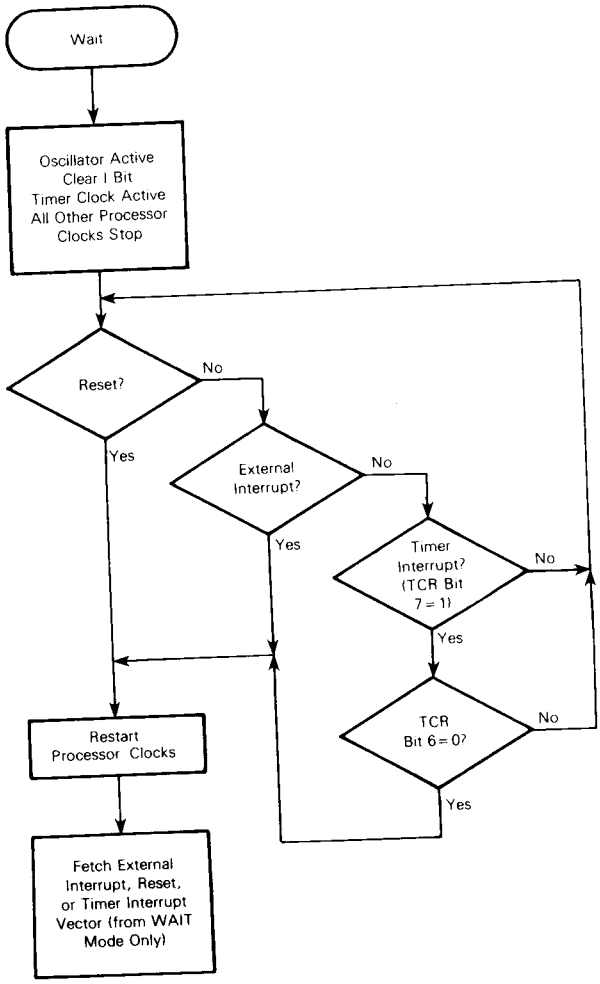
The MCU timer contains an 8-bit software programmable counter (timer data register) with a 7-bit software selectable prescaler. Figure 24 contains a block diagram of the timer. The counter may be loaded under program control and is decremented towards zero by the clock input (prescaler output). When the counter decrements to zero, the timer interrupt request bit (i.e., bit 7 of the timer control register TCR) is set. Then, if the timer interrupt is not masked (i.e., bit 6 of the TCR and the I bit in the condition code register are both cleared) the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack, and then fetches the timer vector address from locations $1FF8 and $1FF9 (or $1FF6 and $1FF7 if in the WAIT mode) in order to begin servicing.

The counter continues to count after it reaches zero, allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without disturbing the count. The contents of the counter become stable prior to the read portion of a cycle, and do not change during the read. The timer interrupt request bit (TCR7) remains set until cleared by the software. If the timer interrupt request bit (TCR7) is cleared before the timer interrupt is serviced, the interrupt is lost. TCR7 may also be used as a scanned status bit in a non-interrupt mode of operation (TCR6 = 1).

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output which is used as the counter input. The processor cannot write into or read from the prescaler; however, its contents are cleared to all zeros by the write operation into TCR when bit 3 of the written data equals a logic one. This allows for truncation-free counting.

The timer input can be configured for three different operating modes plus a disable mode, depending on the value written to the TCR4 and TCR5 control register bits. Refer to TIMER CONTROL REGISTER.

FIGURE 23 – WAIT FUNCTION FLOWCHART



TIMER INPUT MODE 1

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

TIMER INPUT MODE 2

With TCR4 = 1 and TCR5 = 0, the internal clock and the TIMER input pin are ANDed to form the timer input signal. This mode can be used to measure external pulse widths. The external timer input pulse simply turns on the internal clock for the duration of the pulse. The resolution of the count in this mode is ±1 clock; therefore, accuracy improves with longer input pulse widths.

TIMER INPUT MODE 3

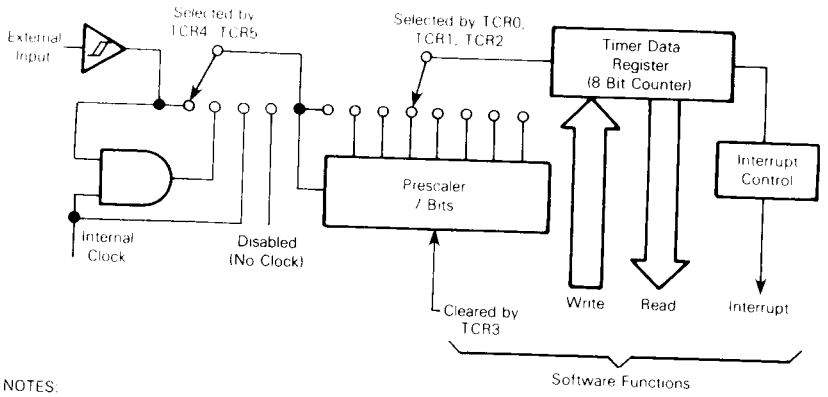
If TCR4 = 0 and TCR5 = 1, then all inputs to the timer are disabled.

TIMER INPUT MODE 4

If TCR4 = 1 and TCR5 = 1, the internal clock input to the timer is disabled and the TIMER input pin becomes the input to the timer. The timer can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts. The counter is clocked on the falling edge of the external signal.

Figure 24 shows a block diagram of the timer subsystem. Power-on Reset and the STOP instruction cause the counter to be set to $F0.

FIGURE 24 — TIMER BLOCK DIAGRAM



NOTES:
1. Prescaler and timer data register (8-bit counter) are clocked on the falling edge of the internal clock or external input
2. The timer data register counts down continuously.

TIMER CONTROL REGISTER (TCR)

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

All bits in this register except bit 3 are Read/Write bits.

TCR7 — Timer interrupt request bit: bit used to indicate the timer interrupt when it is logic one.
1 — Set whenever the counter decrements to zero, or under program control.
0 — Cleared on external reset, power-on reset, STOP instruction, or program control.

TCR6 — Timer interrupt mask bit: when this bit is a logic one it inhibits the timer interrupt to the processor.
1 — Set on external reset, power-on reset, STOP instruction, or program control.
0 — Cleared under program control.

TCR5 — External or internal bit: selects the input clock source to be either the external TIMER pin or the internal clock. (Unaffected by reset.)
1 — Select external clock source.
0 — Select internal clock source (period = t_{cyc}).

TCR4 — External enable bit: control bit used to enable the external TIMER pin. (Unaffected by reset.)
1 — Enable external TIMER pin.
0 — Disable external TIMER pin.

TCR5	TCR4	
0	0	Internal clock to timer
0	1	AND of internal clock and TIMER pin to timer
1	0	Inputs to timer disabled
1	1	TIMER pin to timer

TCR3 — Timer prescaler reset bit: writing a "1" to this bit resets the prescaler to zero. A read of this location always indicates a "0". (Unaffected by reset.)

TCR2, TCR1, TCR0 — Prescaler select bits: decoded to select one of eight outputs of the prescaler. (Unaffected by reset.)

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. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

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 operand for the jump unconditional (JMP) and jump to subroutine (JSR) instructions is the program counter. Refer to Table 5.

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 Table 6.

BRANCH INSTRUCTIONS

Most branch instructions test the state of the condition code register and if certain criteria are met, a branch is executed. This adds an offset between −127 and +128 to the current program counter. Refer to Table 7.

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit which resides in the first 128 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 the first 256 locations. The bit set, bit clear, and bit test and branch functions are all implemented with a single instruction. For the test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to Table 8.

NOTE

The MCU is actually capable of operating on the bit set and bit clear instructions anywhere in the first 256 bytes; however, since only ROM resides in the upper 128 bytes the bit set/clear instructions have no effect on the upper 128 bytes.

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 9.

OPCODE MAP

Table 10 is an opcode map for the instructions used on the MCU.

ALPHABETICAL LISTING

The complete instruction set is given in alphabetical order in Table 11.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code to 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 tables throughout memory. Short absolute (direct) and long absolute addressing are also included. Table 11 shows the addressing modes for each instruction, with the effects each instruction has on the condition code register. An opcode map is shown in Table 10.

The term "Effective Address" (EA) is used in describing the various addressing modes, and is defined as the byte address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of" the location or register referred to; e.g., (PC) indicates the contents of the location pointed to by the PC. An arrow indicates "is replaced by", and a colon indicates concatenation of two bytes. For additional details and graphical illustrations, refer to the M6805 Family User's Manual.

INHERENT

In inherent instructions, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, and no other arguments, are included in this mode.

IMMEDIATE

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

$$EA = PC + 1;\ PC \leftarrow PC + 2$$

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. This includes all on-chip RAM and I/O registers and 128 bytes of on-chip ROM. Direct addressing is efficient in both memory and time.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$

Address Bus High ← 0; Address Bus Low ← (PC + 1)

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes 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 most efficient address mode.

$$EA = (PC + 1):(PC + 2);\ PC \leftarrow PC + 3$$

Address Bus High ← (PC + 1); Address Bus Low ← (PC + 2)

INDEXED, NO OFFSET

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

EA = X; PC ← PC + 1
Address Bus High ← 0; Address Bus Low ← X

INDEXED, 8-BIT OFFSET

Here, the EA is obtained by adding the contents of the byte following the opcode to that of the index register; therefore, the operand is located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the m-th element in an n element table. All instructions are two bytes. The contents of the index register (X) is not changed. The contents of (PC + 1) is an unsigned 8-bit integer. One byte offset indexing permits look-up tables to be easily accessed in either RAM or ROM.

EA = X + (PC + 1); PC ← PC + 2)
Address Bus High ← K; Address Bus Low ← X + (PC + 1)
Where: K = The carry from the addition of X + (PC + 1)

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 (e.g., jump tables in ROM). As with direct and extended, the M6805 assembler determines the most efficient form of indexed offset; 8 or 16-bit. The content of the index register is not changed.

EA = X + [(PC + 1):(PC + 2)]; PC ← PC + 3
Address Bus High ← (PC + 1) + K;
Address Bus Low ← X + (PC + 2)
Where: K = The carry from the addition of X + (PC + 2)

RELATIVE

Relative addressing is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is limited to the range of −126 to +129 bytes from the branch instruction opcode location. The Motorola assembler calculates the proper offset and checks to see if it is within the span of the branch.

EA = PC + 2 + (PC + 1); PC ← EA if branch taken;
otherwise, EA = PC ← PC + 2

BIT SET/CLEAR

Direct addressing and bit addressing are combined in instructions which set and clear individual memory and I/O bits. In the bit set and clear instructions, the byte is specified as a direct address in the location following the opcode. The first 256 addressable locations are thus accessed. The bit to be modified within that byte is specified within the first three bits of the opcode. The bit set and clear instructions occupy two bytes, one for the opcode (including the bit number) and the other to address the byte which contains the bit of interest.

EA = (PC + 1); PC ← PC + 2
Address Bus High ← 0; Address Bus Low ← (PC + 1)

BIT TEST AND BRANCH

Bit test and branch is a combination of direct addressing, bit set or bit clear addressing, and relative addressing. The actual bit to be tested, within the byte, is specified within the low order nibble of the opcode. The address of the data byte to be tested is located via a direct address in the location immediately following the opcode byte (EA1). The signed relative 8-bit offset is in the third byte (EA2) and is added to the PC if the specified bit is set or clear in the specified memory location. This single three byte instruction allows the program to branch based on the condition of any bit in the first 256 locations of memory.

EA1 = (PC + 1)
Address Bus High ← 0; Address Bus Low ← (PC + 1)
EA2 = PC + 3 + (PC + 2); PC ← EA2 if branch taken;
otherwise, PC ← PC + 3

TABLE 5 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in Memory	STA	–	–	–	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in Memory	STX	–	–	–	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump Unconditional	JMP	–	–	–	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to Subroutine	JSR	–	–	–	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

TABLE 6 — READ-MODIFY-WRITE INSTRUCTIONS

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (2's Complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate Left Thru Carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate Right Thru Carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical Shift Left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical Shift Right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic Shift Right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for Negative or Zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5

TABLE 7 — BRANCH INSTRUCTIONS

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	3
Branch IFF Higher	BHI	22	2	3
Branch IFF Lower or Same	BLS	23	2	3
Branch IFF Carry Clear	BCC	24	2	3
(Branch IFF Higher or Same)	(BHS)	24	2	3
Branch IFF Carry Set	BCS	25	2	3
(Branch IFF Lower)	(BLO)	25	2	3
Branch IFF Not Equal	BNE	26	2	3
Branch IFF Equal	BEQ	27	2	3
Branch IFF Half Carry Clear	BHCC	28	2	3
Branch IFF Half Carry Set	BHCS	29	2	3
Branch IFF Plus	BPL	2A	2	3
Branch IFF Minus	BMI	2B	2	3
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	3
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	3
Branch IFF Interrupt Line is Low	BIL	2E	2	3
Branch IFF Interrupt Line is High	BIH	2F	2	3
Branch to Subroutine	BSR	AD	2	6

TABLE 8 — BIT MANIPULATION INSTRUCTIONS

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n (n = 0 7)	—	—	—	2•n	3	5
Branch IFF Bit n is Clear	BRCLR n (n = 0 7)	—	—	—	01 + 2•n	3	5
Set Bit n	BSET n (n = 0 7)	10 + 2•n	2	5	—	—	—
Clear Bit n	BCLR n (n = 0 7)	11 + 2•n	2	5	—	—	—

TABLE 9 — CONTROL INSTRUCTIONS

Function	Mnemonic	Inherent		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

TABLE 10 — M146805 CMOS FAMILY INSTRUCTION SET OPCODE MAP

Hi / Low	Bit Manipulation BTB 0 0000	Bit Manipulation BSC 1 0001	Branch REL 2 0010	Read/Modify/Write DIR 3 0011	Read/Modify/Write INH 4 0100	Read/Modify/Write INH 5 0101	Read/Modify/Write IX1 6 0110	Read/Modify/Write IX 7 0111	Control INH 8 1000	Control INH 9 1001	Register/Memory IMM A 1010	Register/Memory DIR B 1011	Register/Memory EXT C 1100	Register/Memory IX2 D 1101	Register/Memory IX1 E 1110	Register/Memory IX F 1111	Hi / Low
0 0000	BRSET0 3 BTB	BSET0 2 BSC	BRA 2 REL	NEG 5 DIR	NEG 1 INH	NEG 3 NH	NEG 6 X1	NEG 5 X	RTI 9 INH		SUB 2 IMM	SUB 3 DIR	SUB 4 EXT	SUB 5 IX2	SUB 4 IX1	SUB 3 IX	0 0000
1 0001	BRCLR0 3 BTB	BCLR0 2 BSC	BRN 2 REL						RTS 6 INH		CMP 2 IMM	CMP 3 DIR	CMP 4 EXT	CMP 5 IX2	CMP 4 IX1	CMP 3 IX	1 0001
2 0010	BRSET1 3 BTB	BSET1 2 BSC	BHI 2 REL								SBC 2 IMM	SBC 3 DIR	SBC 4 EXT	SBC 5 IX2	SBC 4 IX1	SBC 3 IX	2 0010
3 0011	BRCLR1 3 BTB	BCLR1 2 BSC	BLS 2 REL	COM 5 DIR	COMA 1 INH	COMX 3 NH	COM 6 X1	COM 5 X	SWI 10 INH		CPX 2 IMM	CPX 3 DIR	CPX 4 EXT	CPX 5 IX2	CPX 4 IX1	CPX 3 IX	3 0011
4 0100	BRSET2 3 BTB	BSET2 2 BSC	BCC 2 REL	LSR 5 DIR	LSRA 1 INH	LSRX 3 NH	LSR 6 X1	LSR 5 X			AND 2 IMM	AND 3 DIR	AND 4 EXT	AND 5 IX2	AND 4 IX1	AND 3 IX	4 0100
5 0101	BRCLR2 3 BTB	BCLR2 2 BSC	BCS 2 REL								BIT 2 IMM	BIT 3 DIR	BIT 4 EXT	BIT 5 IX2	BIT 4 IX1	BIT 3 IX	5 0101
6 0110	BRSET3 3 BTB	BSET3 2 BSC	BNE 2 REL	ROR 5 DIR	RORA 1 INH	RORX 3 NH	ROR 6 X1	ROR 5 X			LDA 2 IMM	LDA 3 DIR	LDA 4 EXT	LDA 5 IX2	LDA 4 IX1	LDA 3 IX	6 0110
7 0111	BRCLR3 3 BTB	BCLR3 2 BSC	BEQ 2 REL	ASR 5 DIR	ASRA 1 INH	ASRX 3 NH	ASR 6 X1	ASR 5 X	TAX 2 INH			STA 4 DIR	STA 5 EXT	STA 6 IX2	STA 5 IX1	STA 4 IX	7 0111
8 1000	BRSET4 3 BTB	BSET4 2 BSC	BHCC 2 REL	LSL 5 DIR	LSLA 1 INH	LSLX 3 NH	LSL 6 X1	LSL 5 X	CLC 2 INH		EOR 2 IMM	EOR 3 DIR	EOR 4 EXT	EOR 5 IX2	EOR 4 IX1	EOR 3 IX	8 1000
9 1001	BRCLR4 3 BTB	BCLR4 2 BSC	BHCS 2 REL	ROL 5 DIR	ROLA 1 INH	ROLX 3 NH	ROL 6 X1	ROL 5 X	SEC 2 INH		ADC 2 IMM	ADC 3 DIR	ADC 4 EXT	ADC 5 IX2	ADC 4 IX1	ADC 3 IX	9 1001
A 1010	BRSET5 3 BTB	BSET5 2 BSC	BPL 2 REL	DEC 5 DIR	DECA 1 INH	DECX 3 NH	DEC 6 X1	DEC 5 X	CLI 2 INH		ORA 2 IMM	ORA 3 DIR	ORA 4 EXT	ORA 5 IX2	ORA 4 IX1	ORA 3 IX	A 1010
B 1011	BRCLR5 3 BTB	BCLR5 2 BSC	BMI 2 REL						SEI 2 INH		ADD 2 IMM	ADD 3 DIR	ADD 4 EXT	ADD 5 IX2	ADD 4 IX1	ADD 3 IX	B 1011
C 1100	BRSET6 3 BTB	BSET6 2 BSC	BMC 2 REL	INC 5 DIR	INCA 1 INH	INCX 3 NH	INC 6 X1	INC 5 X	RSP 2 INH			JMP 2 DIR	JMP 3 EXT	JMP 4 IX2	JMP 3 IX1	JMP 2 IX	C 1100
D 1101	BRCLR6 3 BTB	BCLR6 2 BSC	BMS 2 REL	TST 4 DIR	TSTA 1 INH	TSTX 2 NH	TST 5 X1	TST 4 X	NOP 2 INH	BSR 6 REL		JSR 5 DIR	JSR 6 EXT	JSR 7 IX2	JSR 6 IX1	JSR 5 IX	D 1101
E 1110	BRSET7 3 BTB	BSET7 2 BSC	BIL 2 REL						STOP 2 INH		LDX 2 IMM	LDX 3 DIR	LDX 4 EXT	LDX 5 IX2	LDX 4 IX1	LDX 3 IX	E 1110
F 1111	BRCLR7 3 BTB	BCLR7 2 BSC	BIH 2 REL	CLR 5 DIR	CLRA 1 INH	CLRX 3 NH	CLR 6 X1	CLR 5 X	WAIT 2 INH	TXA 2 INH		STX 4 DIR	STX 5 EXT	STX 6 IX2	STX 5 IX1	STX 4 IX	F 1111

Abbreviations for Address Modes

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

LEGEND

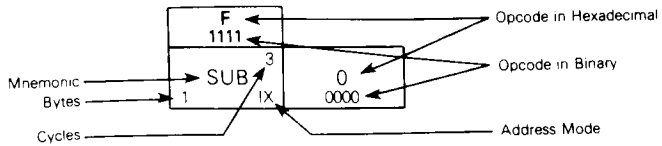


TABLE 11 – INSTRUCTION SET

Mnemonic	Addressing Modes										Condition Codes				
	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
AND		X	X	X		X	X	X			•	•	Λ	Λ	•
ASL	X		X			X	X				•	•	Λ	Λ	Λ
ASR	X		X			X	X				•	•	Λ	Λ	Λ
BCC					X						•	•	Λ	Λ	Λ
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	Λ	Λ	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	Λ
BRSET										X	•	•	•	•	Λ
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLC	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	Λ	Λ	Λ
COM	X		X			X	X	X			•	•	Λ	Λ	1
CPX		X	X	X		X	X	X			•	•	Λ	Λ	Λ
DEC	X		X			X	X				•	•	Λ	Λ	•
EOR		X	X	X		X	X	X			•	•	Λ	Λ	•
INC	X		X			X	X				•	•	Λ	Λ	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	Λ	Λ	•
LDX		X	X	X		X	X	X			•	•	Λ	Λ	•
LSL	X		X			X	X				•	•	Λ	Λ	Λ
LSR	X		X			X	X				•	•	Λ	Λ	Λ
NEG	X		X			X	X				•	•	0	Λ	Λ
NOP	X										•	•	Λ	Λ	Λ
ORA		X	X	X		X	X	X			•	•	Λ	Λ	•
ROL	X		X			X	X				•	•	Λ	Λ	Λ
ROR	X		X			X	X				•	•	Λ	Λ	Λ
RSP	X										•	•	•	•	•
RTI	X										?	?	?	?	?
RTS	X										•	•	•	•	•
SBC		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SEC	X										•	•	•	•	1
SEI	X										•	1	•	•	•
STA			X	X		X	X	X			•	•	Λ	Λ	•
STOP	X										•	0	•	•	•
STX			X	X		X	X	X			•	•	Λ	Λ	•
SUB		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SWI	X										•	1	•	•	•
TAX	X										•	•	•	•	•
TST	X		X			X	X				•	•	Λ	Λ	•
TXA	X										•	•	•	•	•
WAIT	X										•	0	•	•	•

Condition Code Symbols

- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- Λ Test and Set if True, Cleared Otherwise
- • Not Affected
- ? Load CC Register From Stack
- 0 Cleared
- 1 Set

ORDERING INFORMATION

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

EPROM(s) MCM2716s or MCM2532s

MDOS disk file

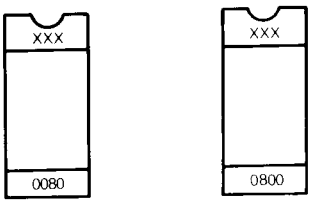
To initiate a ROM pattern for the MCU, it is necessary to first contact your local field service office, local sales person, or your local Motorola representative.

EPROMs

The MCM2716 or MCM2532 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROMs must be clearly marked to indicate which EPROM corresponds to which address space. Figure A-1 illustrates the recommended marking procedure for two MCM2716 EPROMs.

After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

FIGURE A-1 – 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. If desired, Motorola will program a blank 2716 EPROM (supplied by the customer) from the data file used to create the custom mask to aid in the verification process.

ROM VERIFICATION UNITS

The 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 they are usually unmarked, packaged in ceramic, and tested only at room temperature and 5 volts. These RVUs are included in the mask charge and are not production parts. These RVUs are not backed nor guaranteed by Motorola Quality Assurance.

FLEXIBLE DISKS

The disk media submitted must be single-sided, single-density, 8-inch, MDOS compatible floppies. The customer must write the binary file name and company name on the disk with a felt-tip pen. The floppies are not to be returned by Motorola as they are used for archival storage. The minimum MDOS system files as well as the absolute binary object file (filename.LO type of file) from the M6805 cross assembler must be on the disk. An object file made from a memory dump using the ROLLOUT command is also admissable. Consider submitting a source listing as well as the following files: filename.LX (EXORciser loadable format) and filename.SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process in house if any problems arise, and 2) to speed up our customer to factory interface if a user finds any software errors and needs assistance quickly from the factory representatives.

MDOS is Motorola's Disk Operating System available on development systems such as EXORciser, EXORset, etc.

Option List

Select the options for your MCU from the following list. A manufacturing mask will be generated from this information. Select one in each section.

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Internal Divide
- ☐ ÷4
- ☐ ÷2

Interrupt Trigger
- ☐ Edge-Sensitive Only
- ☐ Level-Sensitive and Edge-Sensitive

Customer Name _____

Address _____

City _____ State _____ Zip _____

Phone (_____) _____ Extension _____

Contact Ms/Mr _____

Customer Part Number _____

Pattern Media

- 2708 EPROM
- 2716 EPROM
- MDOS Disk File
- Silent 700 Cassette
- Card Deck
- Tape of Card Deck
- (Note 2) _____

Notes: (2) Other media require prior factory approval.

Signature _____

Title _____

Silent 700 Cassette is a trademark of Texas Instruments Incorporated