SGS-THOMSON MICROELECTRONICS

EF6805P2

8-BIT MICROCOMPUTER UNIT

ADVANCE DATA

HARDWARE FEATURES

- 8-BIT ARCHITECTURE
- 64 BYTES OF RAM
- MEMORY MAPPED I/O
- 1100 BYTES OF USER ROM
- 20 TTL/CMOS COMPATIBLE BIDIRECTIONAL I/O LINES (8 Lines are LED Compatible)
- ON-CHIP CLOCK GENERATOR
- SELF-CHECK MODE
- ZERO CROSSING DETECTION
- MASTER RESET
- COMPLETE DEVELOPMENT SYSTEM SUPPORT ON INICE®
- 5V SINGLE SUPPLY

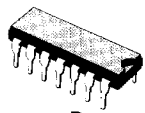
SOFTWARE FEATURES

- SIMILAR TO 6800 FAMILY
- BYTE EFFICIENT INSTRUCTION SET
- EASY TO PROGRAM
- TRUE BIT MANIPULATION
- BIT TEST AND BRANCH INSTRUCTION
- VERSATILE INTERRUPT HANDLING
- VERSATILE INDEX REGISTER
- POWERFUL INDEXED ADDRESSING FOR TABLES
- FULL SET OF CONDITIONAL BRANCHES
- MEMORY USABLE AS REGISTER/FLAGS
- SINGLE INSTRUCTION MEMORY EXAMINE/CHANGE
- 10 POWERFUL ADDRESSING MODES
- ALL ADDRESSING MODES APPLY TO ROM, RAM, AND I/O

USER SELECTABLE OPTIONS

- INTERNAL 8-BIT TIMER WITH SELECTABLE CLOCK SOURCE (external timer input or internal machine clock)
- TIMER PRESCALER OPTION (7 Bits, 2^n)
- 8 BIDIRECTIONAL I/O LINES WITH ttl OR TTL/CMOS INTERFACE OPTION
- CRYSTAL OR LOW-COST RESISTOR OSCILLATOR OPTION
- LOW VOLTAGE INHIBIT OPTION
- VECTORED INTERRUPTS TIMER, SOFTWARE, AND EXTERNAL

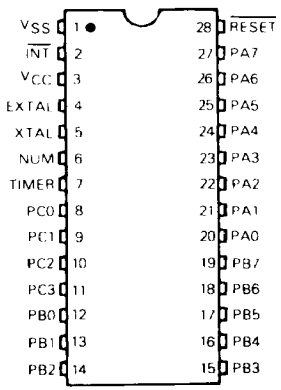
Inice : SGS THOMSON development/emulation tool.

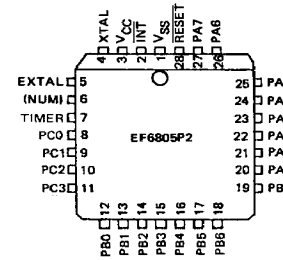


P
(Plastic Package)

FN
(PLCC 28)

PIN CONNECTIONS

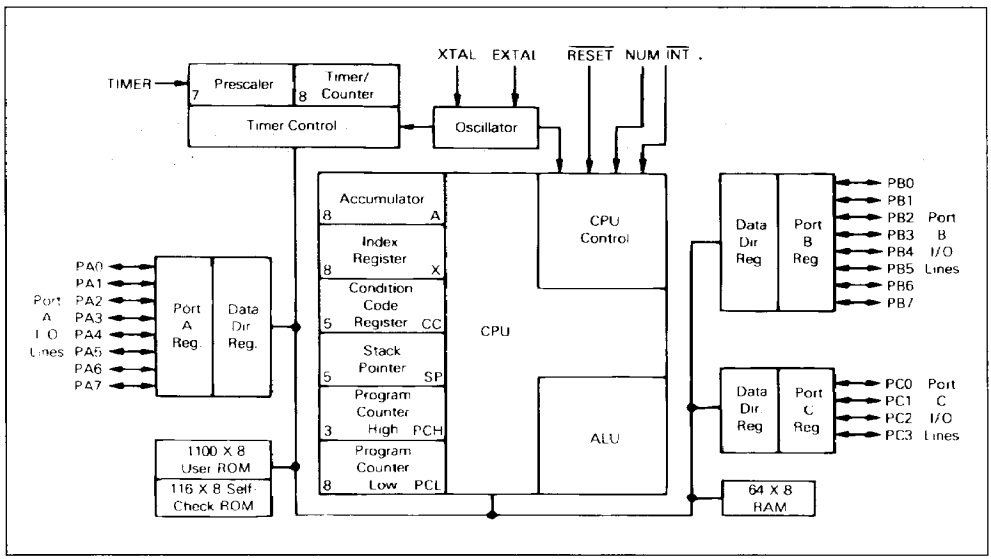




DESCRIPTION

The EF6805P2 Microcomputer Unit (MCU) is a member of the EF6805 Family of low-cost single-chip microcomputers. This 8-bit microcomputer contains a CPU, on-chip CLOCK, ROM, RAM, I/O, and TIMER. It is designed for the user who needs an economical microcomputer with the proven capabilities of the 6800-based instruction set. A comparison of the key features of several members of the 6805 Family is shown at the end of this data sheet. The following are some of the hardware and software highlights of the EF6805P2 MCU.

Figure 1 : EF6805P2 HMOS Microcomputer Block Diagram.



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	− 0.3 to + 7.0	V
V_{in}	Input Voltage (except pin 6)	− 0.3 to + 7.0	V
T_A	Operating Temperature Range (T_L to T_H) V Suffix	0 to 70 − 40 to + 85	°C
T_{stg}	Storage Temperature Range	− 55 to + 150	°C
T_J	Junction Temperature Plastic PLCC	 150 150	°C

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

THERMAL DATA

θ_{JA}	Thermal Resistance Plastic PLCC	70 110	°C/W

POWER CONSIDERATIONS

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

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

Where :

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts - Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts - User Determined

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

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

$P_D = K \div (T_J + 273°C)$ (2)

Solving equations 1 and 2 for K gives :

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

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

ELECTRICAL CHARACTERISTICS

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

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage				V
	$\overline{\text{RESET}}$ (4.75 ≤ V_{CC} ≤ 5.75)	4.0		V_{CC}	
	(V_{CC} < 4.75)	V_{CC} – 0.5		V_{CC}	
	$\overline{\text{INT}}$ (4.75 ≤ V_{CC} ≤ 5.75)	4.0	*	V_{CC}	
	(V_{CC} < 4.75)	V_{CC} – 0.5	*	V_{CC}	
	All Other (except TIMER)	2.0		V_{CC}	
V_{IH}	Input High Voltage Timer				V
	Timer Mode	2.0		V_{CC} + 1	
	Self-check Mode	9.0	10.0	15.0	
V_{IL}	Input Low Voltage				V
	$\overline{\text{INT}}$	V_{SS}	*	1.5	
	All Other	V_{SS}		0.8	
V_{IRES+} V_{IRES-}	$\overline{\text{RESET}}$ Hystereris Voltage (see figures 10, 11 and 12)				V
	"Out of Reset"	2.1		4.0	
	"Into Reset"	0.8		2.0	
V_{INT}	$\overline{\text{INT}}$ Zero Crossing Input Voltage, through a Capacitor	2.0		4.0	$V_{ac\ p\text{-}p}$
P_{INT}	Internal Power Dissipation - No Port Loading V_{CC} = 5.75V, T_A = 0°C		400	690	mW
C_{in}	Input Capacitance				pF
	EXTAL		25		
	All Other		10		
V_{LVR}	Low Voltage Recover			4.75	V
V_{LVI}	Low Voltage Inhibit 0 to + 70°C	2.75	3.5		V
	– 40 to + 85°C	3.1	3.5		
I_{in}	Input Current				µA
	$\overline{\text{TIMER}}$ (V_{in} = 0.4V)			20	
	$\overline{\text{INT}}$ (V_{in} = 2.4V to V_{CC})		20	50	
	EXTAL (V_{in} = 2.4V to V_{CC}, crystal option)			10	
	(V_{in} = 0.4V, crystal option)			– 1600	
	$\overline{\text{RESET}}$ (V_{in} = 0.8V)	– 4.0		– 40	
	(external capacitor charging current)				

* Due to internal biasing, this input (when unused) floats to approximately 2.0 Vdc.

PORT DC ELECTRICAL CHARACTERISTICS

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

PORT A WITH CMOS DRIVE ENABLED

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.6mA			0.4	V
V_{OH}	Output High Voltage, I_{Load} = – 100µA	2.4			V
V_{OH}	Output High Voltage, I_{Load} = – 10µA	V_{CC} – 1.0			V
V_{IH}	Input High Voltage, I_{Load} = – 300µA (max.)	2.0		V_{CC}	V
V_{IL}	Input Low Voltage, I_{Load} = – 500µA (max.)	V_{SS}		0.8	V
I_{IH}	Hi-Z State Input Current (V_{IN} = 2.0V to V_{CC})			– 300	µA
I_{IL}	Hi-Z State Input Current (V_{IN} = 0.4V)			– 500	µA

PORT B

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 3.2mA			0.4	V
V_{OL}	Output Low Voltage, I_{Load} = 10mA (sink)			1.0	V
V_{OH}	Output High Voltage, I_{Load} = – 200µA	2.4			V
I_{OH}	Darlington Current Drive (source), V_O = 1.5V	– 1.0		– 10	mA
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current		2	10	µA

PORT C AND PORT A WITH CMOS DRIVE DISABLED

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.6mA			0.4	V
V_{OH}	Output High Voltage, I_{Load} = – 100µA	2.4			V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current		2	10	µA

SWITCHING CHARACTERISTICS

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

Symbol	Parameter	Min.	Typ.	Max.	Unit
f_{osc}	Oscillator Frequency EF6805P2 EF68A05P2 EF68B05P2	0.4 0.4 0.4		4.2 6.0 8.0	MHz
t_{cyc}	Cycle Time ($4/f_{osc}$)	0.95		10	µs
t_{WL}, t_{WH}	$\overline{INT}$ and TIMER Pulse Width (see interrupt section)	t_{cyc} + 250			ns
t_{RWL}	$\overline{RESET}$ Pulse Width	t_{cyc} + 250			ns
t_{RHL}	$\overline{RESET}$ Delay Time (external capacitance = 1.0µF)		100		ms
f_{INT}	$\overline{INT}$ Zero Crossing Detection Input Frequency (± 5° Accuracy)	0.03		1.0	kHz
	External Clock Input Duty Cycle (EXTAL)	40	50	60	%

Figure 2 : TTL Equivalent Test Load (port B).

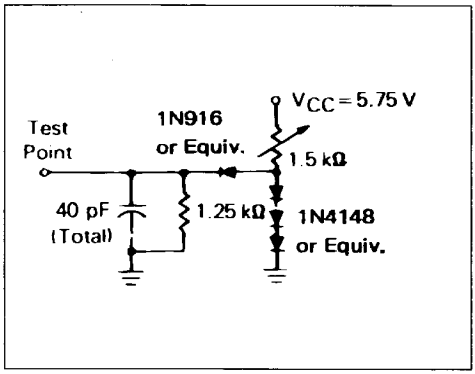


Figure 3 : CMOS Equivalent Test Load (port A).

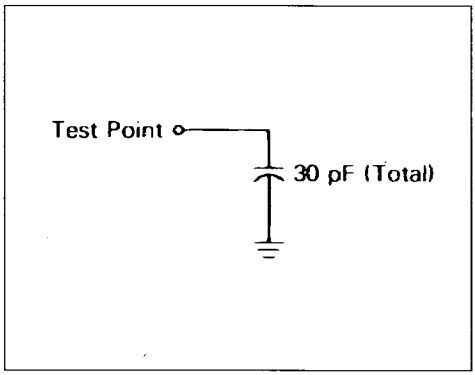
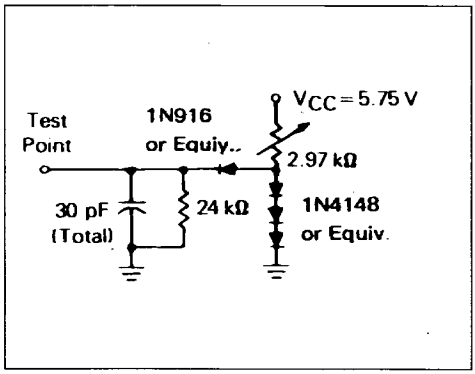


Figure 4 : TTL Equivalent Test Load (port A and C).



SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in figure 1, are described in the following paragraphs.

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

$\overline{INT}$. This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to Interrupts section for additional information.

XTAL AND EXTAL. These pins provide connections to the on-chip clock oscillator circuit. A crystal, a resistor, or an external signal, depending on the user selectable manufacturing mask option, can be connected to these pins to provide a system clock source with various stability/cost tradeoffs. Lead lengths and stray capacitance on these two pins should be minimized. Refer to Internal Clock Generator Options section for recommendations about these inputs.

TIMER. This pin allow an external input to be used to decrement the internal timer circuitry. Refer to Timer section for additional information about the timer circuitry.

RESET. This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. Refer to Resets section for additional information.

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

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). All lines are programmable as either inputs or outputs under software control of the data direction registers. Refer to Inputs/Outputs section for additional information.

MEMORY

As shown in figure 5, the MCU is capable of addressing 2048 bytes of memory and I/O registers with is program counter. The EF6805P2 MCU has implemented 1288 of these locations. This consists of : 1100 bytes of user ROM, (from $080 to $OFF and from $3CO to $783) 116 bytes of self-check ROM, 64 bytes of user RAM, 6 bytes of port I/O, and 2 timers registers. The ROM division allows 128 bytes of ROM to be addressed with direct instructions.

The stack area is used during the processing of interrupt and subroutine calls to save the processor state. The register contents are pushed onto the stack in the order shown in figure 6. Because the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first ; then the high order three bits (PCH) are stacked. This ensures that the program counter is loaded correctly, during pulls from the stack, since the stack pointer increments during pulls. A subroutine call results in only the program counter (PCL, PCH) contents being pushed onto the stack. The remaining CPU registers are not pushed.

CENTRAL PROCESSING UNIT

The CPU of the EF6805 Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal address, data, and control buses.

REGISTERS

The 6805 Family CPU has five registers available to the programmer. They are shown in figure 7 and are explained on the following paragraphs.

Figure 5 : MCU Address Map.

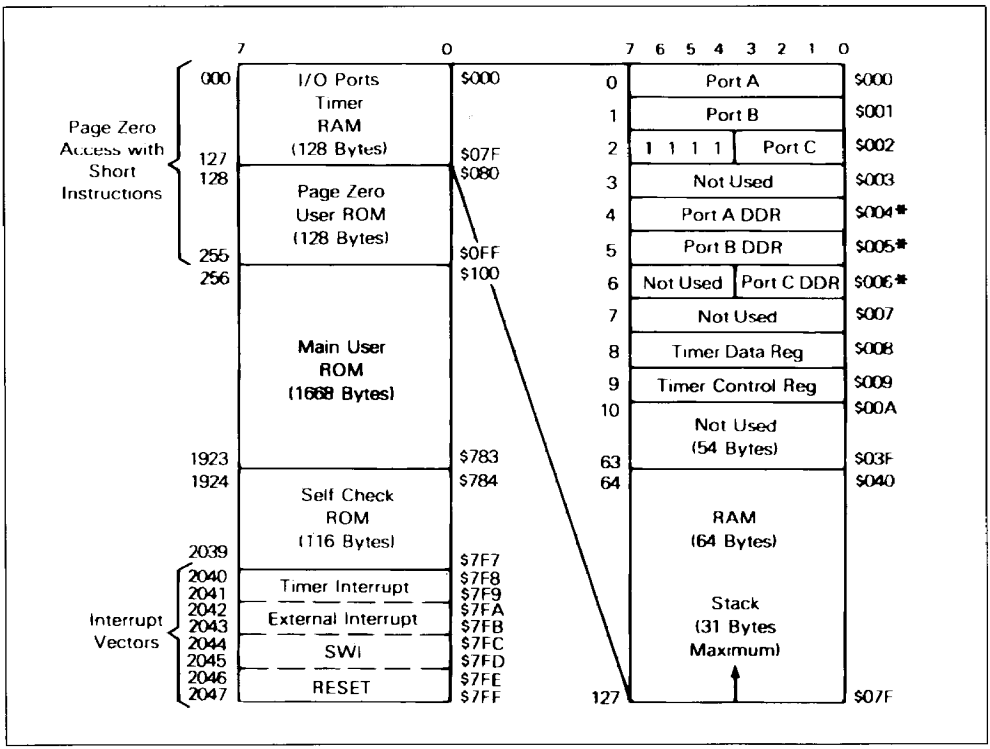


Figure 6 : Interrupt Stacking Order.

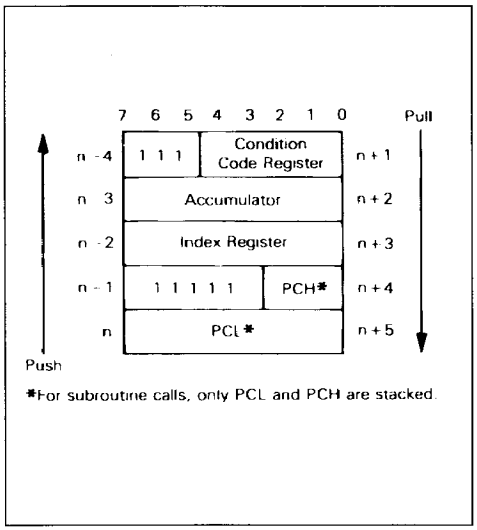
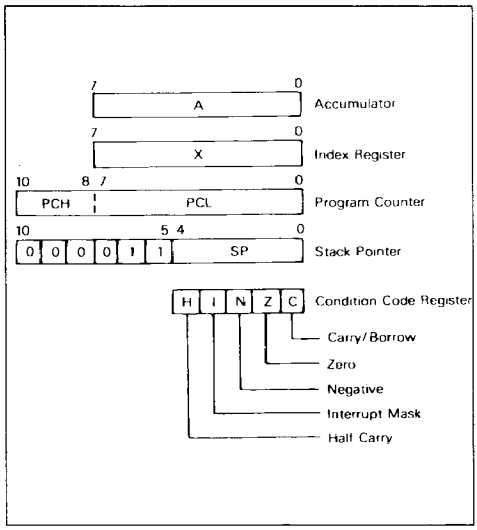


Figure 7 : Programming Model.



ACCUMULATOR (A)

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

INDEX REGISTER (X). The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an instruction value to create an effective address. The index register can also be used for data manipulations using the read modify-write instructions. 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 instruction to be executed.

STACK POINTER (SP). The stack pointer is an 11-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $07F and is 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 configured to 000011. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. 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.

CONDITION CODE REGISTER (CC). The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained in the following paragraphs.

HALF CARRY (H) - Set during ADD and ADC instructions to indicate that a carry occurred between bits 3 and 4.

INTERRUPT (I) - This bit is set to mask (disable) the timer and external interrupt (INT). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt is cleared.

NEGATIVE (N) - Used to indicate that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in result equal to a logical one).

ZERO (Z) - Used to indicate that the result of the last arithmetic, logical, or data manipulation was zero.

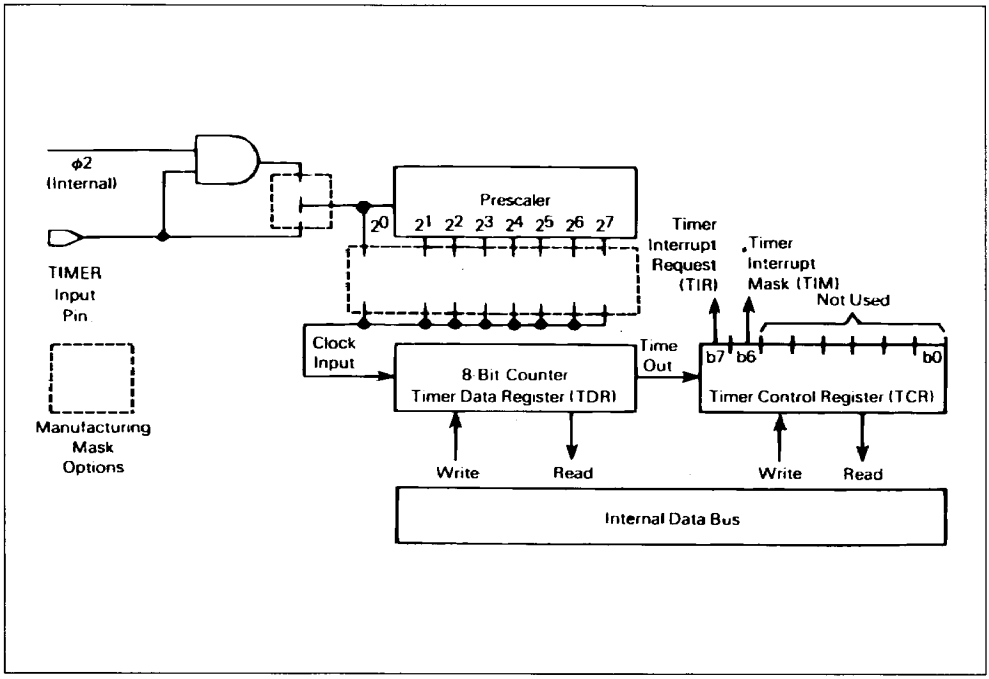
CARRY/BORROW (C) - Used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic logic unit (ALU) occurred during the last airhtmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

TIMER

The EF6805P2 MCU timer circuitry is shown in figure 8. The 8-bit counter may be loaded under program control and is decremented toward zero by the clock input (prescaler output). When the timer reaches zero, the timer interrupt request bit (bit 7) in the timer control register (TCR) is set. The timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. The interrupt bit (I bit) in the condition code register also prevents a timer interrupt from being processed. The MCU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $7F8 and $7F9, and exexuting the interrupt routine ; see the Interrupts section. THE TIMER INTERRUPT REQUEST BIT MUST BE CLEARED BY SOFTWARE.

The clock input to the timer can be from an external source (decrementing of timer counter occurs on a positive transition of the external source) applied to the TIMER input pin or it can be the internal 2 signal. Three machine cycles are required for a change in state of the TIMER pin to decrement the timer prescaler. The maximum frequency of a signal that can be recognized by the TIMER or INT pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high (or low) state on

Figure 8 : Timer Block Diagram.



the pin must also recognize the low state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows : (assumes 50/50 duty cycle for a given period).

$$t_{cyc} \times 2 + 250ns = period = \frac{1}{freq}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250ns twice).

When the $\phi 2$ signal is used as the source, it can be gated by an input applied to the TIMER input pin allowing the user to easily perform pulse-width measurements. (NOTE : For ungated $\phi 2$ clock inputs to the timer prescaler, the TIMER pin should be tied to V_{CC}). The source of the clock input is one of the mask options that is specified before manufacture of the MCU.

A prescaler option can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter. This prescaling mask option is also specified before manufacture. To avoid truncation errors, the prescaler is cleared when bit 3 of the timer counter register is written to a logic one. (This bit always needs a logic 0).

The timer continues to count past zero, falling from $00 to $FF and then continuing the countdown. Thus, 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.

At power-up or reset, the prescaler and counter are initialized with all logical ones, the timer interupt request bit (bit 7) is cleared and the timer interrupt mask bit (bit 6) is set.

SELF-CHECK

The self-check capability of the EF6805P2 MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in figure 9 and monitor the output of port C bit 3 for an oscillation of approximately 7Hz. A 10 volt level on the TIMER input, pin 7, energizes the ROM-based self-check feature. The self-check program exercises the RAM, ROM, TIMER, interrupts, and I/O ports.

RESETS

The MCU can be reset three ways : by initial power-up, by the external reset input (RESET) and by optional, internal, low-voltage detect circuits. The RESET input consists mainly of a Schmitt trigger that senses the RESET line logic level. A typical reset Schmitt trigger hysteresis curve is shown in figure 11. The Schmitt trigger provides an internal reset voltage if it senses a logical zero on the RESET pin.

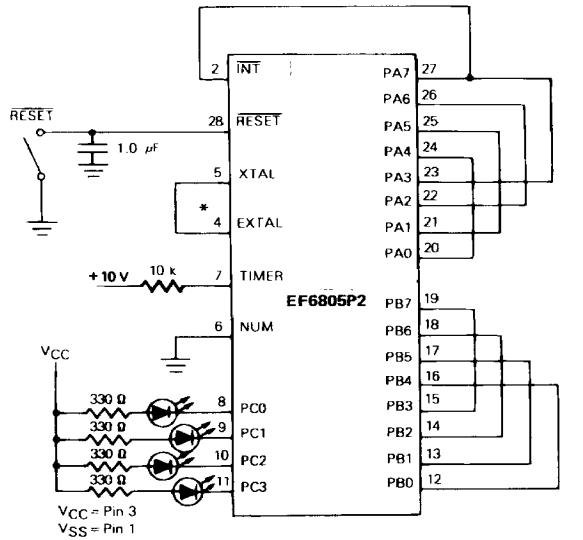
POWER-ON RESET (POR)

An internal reset is generated upon power-up that allows the internal clock generator to stabilize. A delay of t_{RHL} milliseconds is required before allowings the RESET input to go high. See the power and reset timing diagram (see figure 10). Connecting a capacitor to the RESET input (see figure 12) typically provides sufficient delay. During power-up, the Schmitt trigger switches on (removes reset) when RESET rise to V_{IRES^+}.

EXTERNAL RESET INPUT

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

Figure 9 : Self-check Connections.



*This connection depends on the clock oscillator user selectable mask option. Use crystal if that option is selected.

SELF-CHECK ERROR PATTERNS

PC1	PC0	Problem
0	0	Interrupt Failure
0	1	Bad Port A or Port B
1	0	Bad RAM
1	1	Bad RAM
All 4 LEDs Flashing		Good Device

Note : When PC1 or PC0 is 0, the LED is on.

LOW VOLTAGE INHIBIT (LVI)

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

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

INTERNAL CLOCK GENERATORS OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to generate a system clock with various stability/cost tradeoffs.

A manufacturing mask option is used to select crystal or resistor operation.

The different connection methods are shown in figure 13. Crystal specifications and suggested PC board layouts are given in figure 14. A resistor selection graph is given in figure 15.

The crystal oscillator start-up time is a function of many variables : crystal parameters (especially R_S, oscillator load capacitances, IC parameters, ambient temperature, supply voltage and supply voltage turn-on time). To ensure rapid oscillator start-up, neitherthe crystal characteristics nor the load capacitances should exceed recommendations.

When utilizing the on-board oscillator, the MCU should remain in the reset condition ($\overline{\text{RESET}}$ pin voltage below V_{IRES}^+) until the oscillator has stabilized at its operating frequency. Several factors are involved in calculating current specifications.

One V_{CC} minimum is reached, the external $\overline{\text{RESET}}$ capacitor will begin to charge at a rate dependent on the capacitor value. The charging current is supplied from V_{CC} through a large resistor, so its functions almost like a constant current source until the reset voltages rises above V_{IRES}^+. Therefore, the RESET pin will charge a approximately

$(V_{IRES}^+) \cdot C_{ext} = I_{RES} \cdot t_{RHL}$

Figure 10 : Power and Reset Timing.

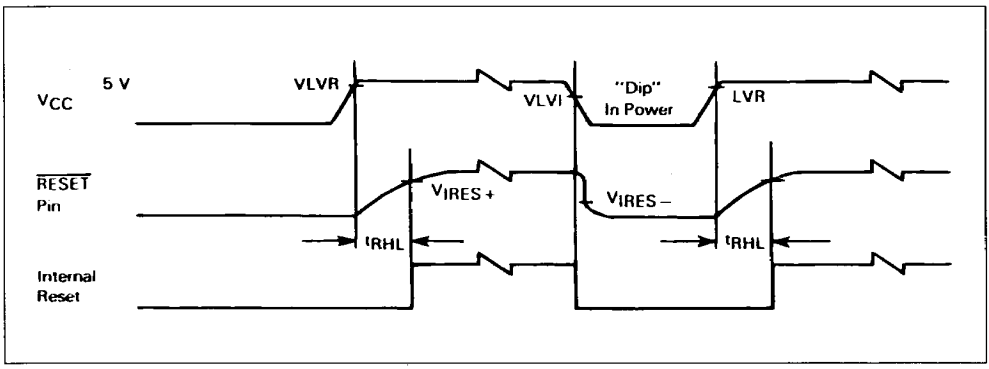


Figure 11 : Typical Reset Schmitt Trigger Hysteresis.

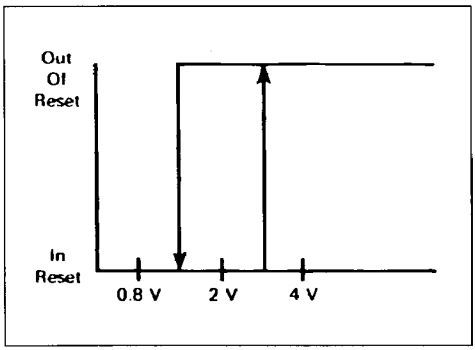


Figure 12 : Power-up Reset Delay Circuit.

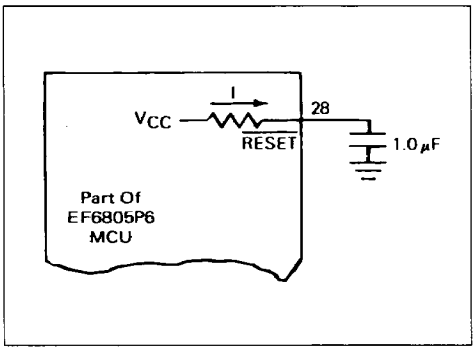
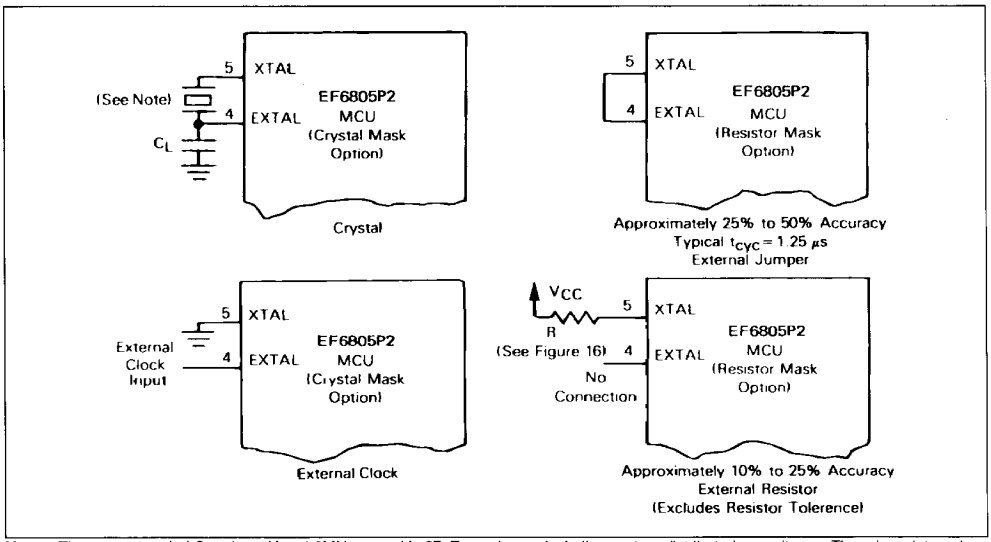
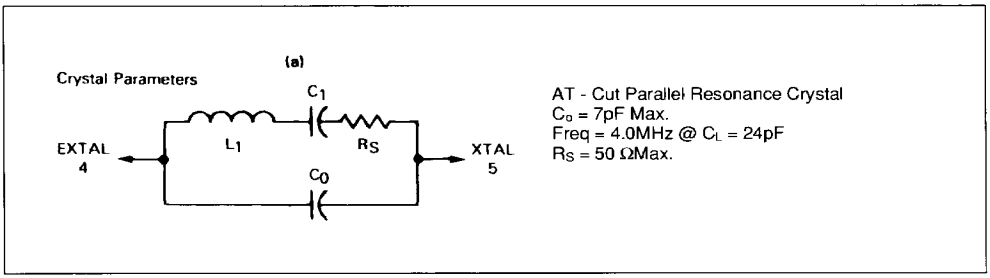


Figure 13 : Clock Generator Options.



Note : The recommended C_L value with a 4.0MHz crystal is 27pF, maximum, including system distributed capacitance. There is an internal capacitance of approximately 25pF on the XTAL pin. For crystal frequencies other than 4MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with a 2MHz crystal, use approximately 50pF on EXTAL and approximately 25pF on XTAL. The exact value depends on the Motional-Arm parameters of the crystal used.

Figure 14 : Crystal Motional ARM Parameters and Suggested PC Board Layout.



Note : Piezoelectric ceramic resonators which have the equivalent specifications may be use instead of crystal oscillator. Follow ceramic resonator manufacturer's suggestions for C_o, C_1 and R_S values..

Figure 14 : Crystal Motional Arm Parameters and Suggested PC Board Layout (continued).

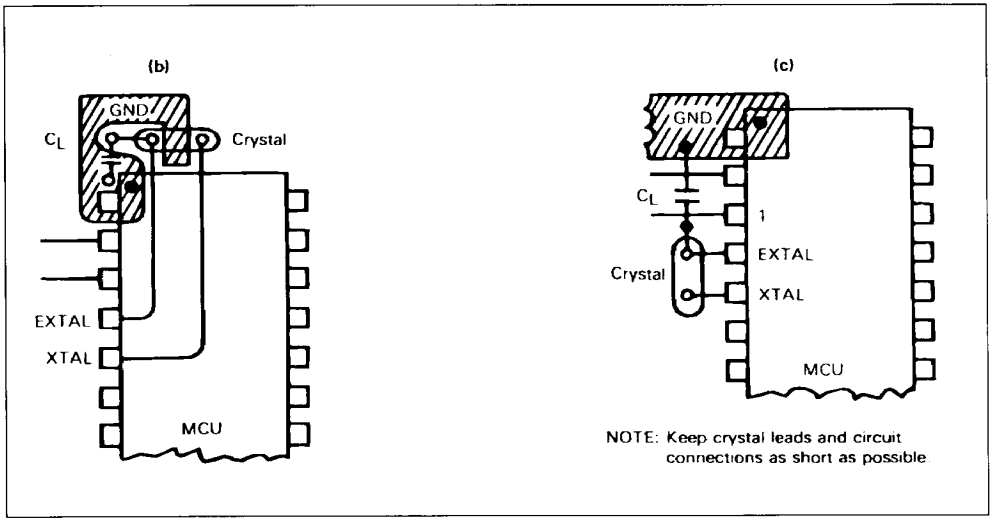
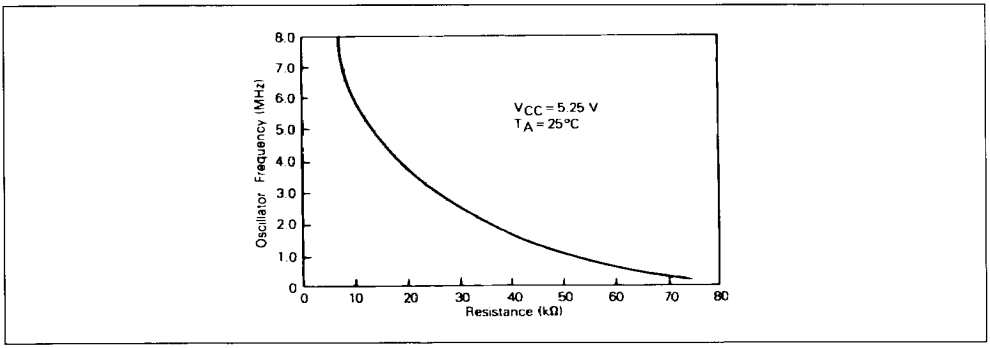


Figure 15 : Typical Frequency Selection for Resistor Oscillator Option.



INTERRUPTS

The EF6805P2 MCU can be interrupted three different ways through the external interrupt (INT) input pin, the internal timer interrupt request, or the software interrupt instruction (SWI). When any interrupt occurs, the current instruction (including SWI) is completed, processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address and the interrupt routine is executed. Stacking the CPU registers, setting the I bit, and vector fetching requires a total of 11 t_{cyc} periods for completion.

A flowchart of the interrupt sequence is shown in figure 16. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the MCU to resume processing of the program prior to the interrupt (by unstacking the previous CPU state). Unlike RESET, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing ; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt device.

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 as any other instruction.

The external interrupt is internally synchronized and then latched on the falling edge of $\overline{INT}$. A sinusoidal input signal (f_{INT} maximum) can be used to generate an external interrupt, as shown in figure 17(a), for use as a zero-crossing detector (for negative transitions of the ac sinusoid). This allows applications such as servicing time-of-day routines and engaging/disengaging ac power control devices. Off-chip full-wave rectification provides an interrupt at every zero crossing of the ac signal and thereby provides a 2f clock.

For digital applications, the $\overline{INT}$ pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{INT}$ pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high (or low) state on the pin must also recognize the low (or high) state on the pin in order to "rearm" the internal logic. Therefore, the period can be calculated as follows : (assumes 50/50 duty cycle for a given period)

$$t_{cyc} \times 2 + 250ns = period = \frac{1}{freq}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 ns twice). See figure 17(b).

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I bit in the condition code register. Note that if the I bit is zero, SWI executes after the other interrupts. SWIs are usually used as break-points for debugging or as system calls.

Figure 16 : $\overline{RESET}$ and Interrupt Processing Flowchart.

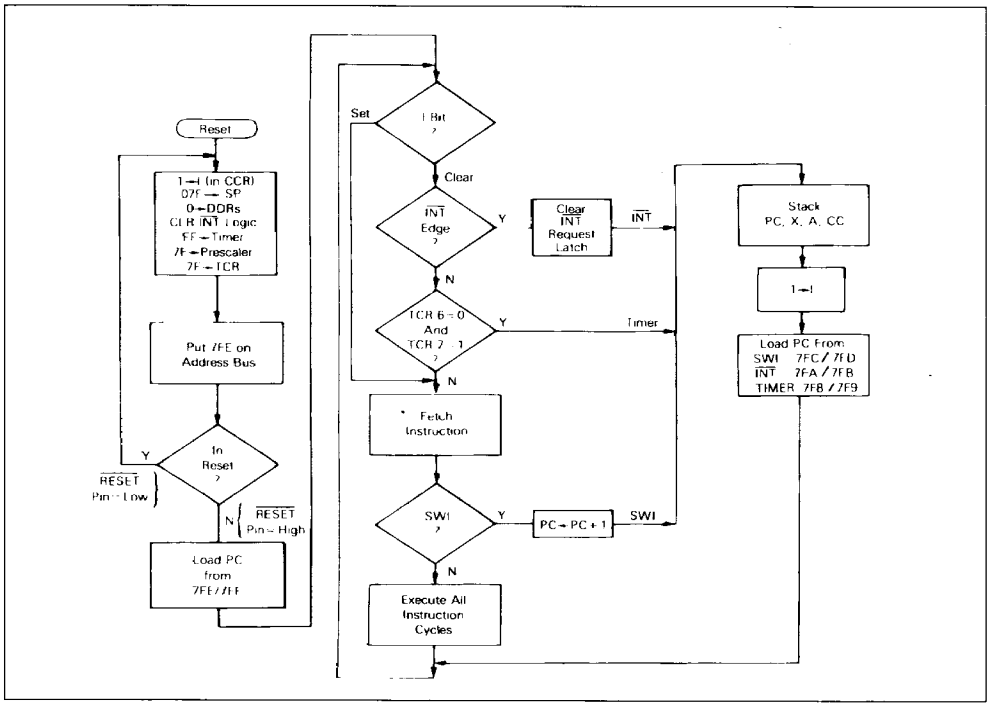
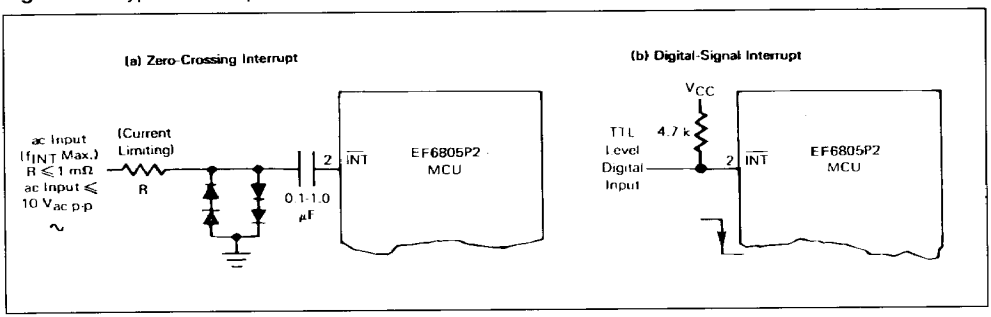


Figure 17 : Typical Interrupt Circuits.



INPUT/OUTPUT

There are 20 input/output pins. The $\overline{INT}$ pin may also be polled with branch instructions to provide an additional input pin. All pins (port A, B, and C) are programmable as either inputs or outputs 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 or a logic "0" state to put the ports in the input mode. To avoid undefined levels, the port output registers are not initialized on reset, but may be written 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 ; see figure 18. When port B is programmed for outputs, it is capable of sinking 10mA and sourcing 1mA on each pin.

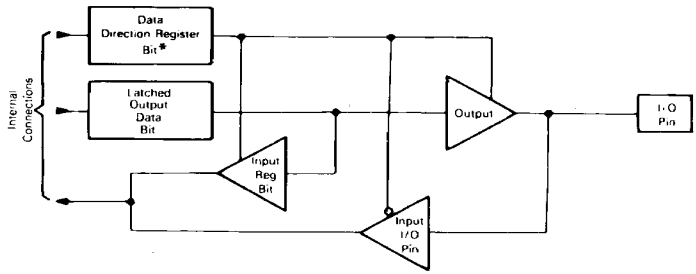
All input/output lines are TTL compatible as both inputs and outputs. Ports B and C are CMOS compatible as inputs. Port A may be made CMOS compatible as outputs with a mask option. The address map in figure 5 gives the address of data registers and DDRs. The register configuration is provided in figure 19 and figure 20 provides some examples of port connections.

Caution

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

The latched output data bit (see figure 18) 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 may be used to initialize the data registers and avoid undefined outputs ; however, 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 corresponds to the latched output data when the DDR is an output ("1").

Figure 18 : Typical Port I/O Circuitry.



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

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

Figure 19 : MCU Register Configuration.

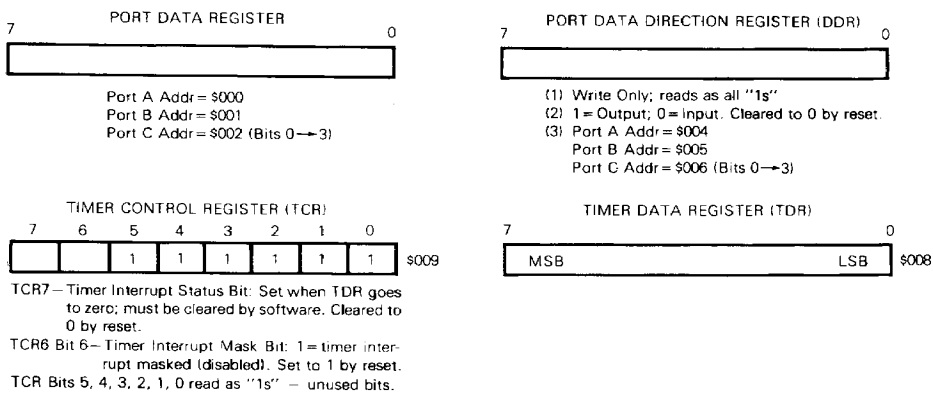


Figure 20 (a) : Typical Output Mode Port Connections.

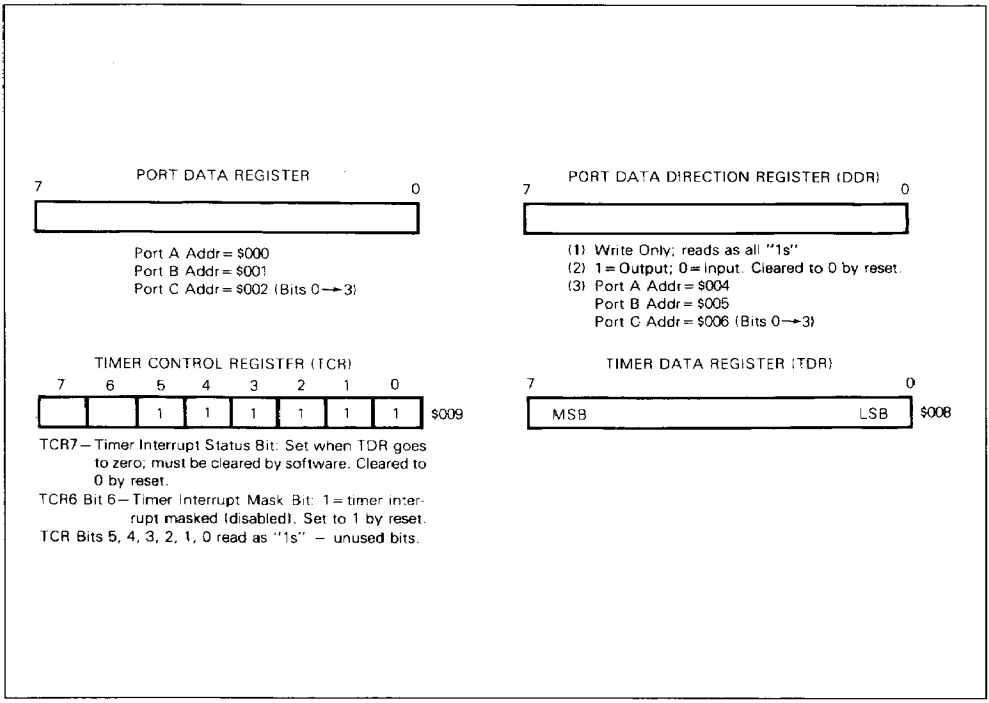
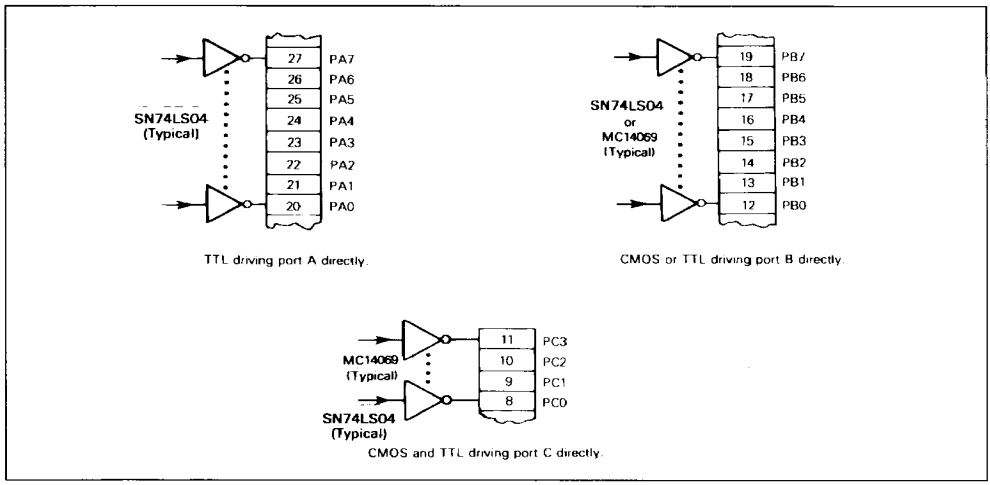


Figure 20 (b) : Typical Output Mode Port Connections.



SOFTWARE

BIT MANIPULATION

The EF6805P2 MCU has the ability to set or clear any single random access memory or input/output bit (except the data direction register, see Caution below), with a single instruction (BSET, BCLR). Any bit in page zero including ROM, except the DDRs, can be tested, using the BRSET and BRCLR instructions, and the program branches as a result of its state. The carry bit equals the value of the bit referenced by BRSET or BRCLR. A rotate instruction may then be used to accumulate serial input data in a RAM location or register. The capability to work with any bit in RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle I/O bits as control lines.

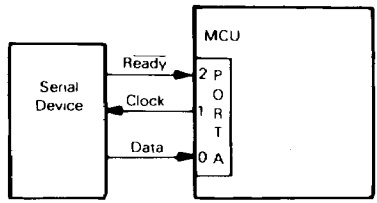
The coding example in figure 21 illustrates the usefulness of the bit manipulation and test instructions. Assume that the MCU is to communicate with an external serial device.

The external device has a data ready signal, a data output line, and a clock line to clock data one bit at a time. LSB first, out of the device. The MCU waits until the data is ready, clocks the external device, picks up the data in the carry flag (C bit), clears the clock line, and finally accumulates the data bit in a RAM location.

Caution

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

Figure 21 : Bit Manipulation Example.




```
                .
                .
                .
SELF    BRSET   2, PORTA, SELF
                .
                .
                .
        BSET    1, PORTA
        BRCLR   0, PORTA, CONT
CONT    BCLR    1, PORTA
        ASR     RAMLOC
                .
                .
                .
```

ADDRESSING MODES

The EF6805P2 MCU has 10 addressing modes which are explained briefly in the following paragraphs. For additional details and graphical illustrations, refer to the 6805 Family User's Manual.

The term "effective address" (EA) is used in describing the address modes. EA 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 which 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 addresing allows the user to directly address the lowest 256 bytes in memory with a single 2-byte instruction. This includes the on-chip RAM and I/O registers and 128 bytes of ROM. Direct addressing is an effective use of both memory and time.

EXTENDED - In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions using extended addressing are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the Motorola assembler, the programmer need not specify whether an instruction uses direct or extended addressing. The as-

sembler automatically selects the shortest for 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 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 from - 126 to + 129 from the opcode address. The programmer need not worry about calculating the correct offset when using the Motorola assembler since it calculates the proper offset and checks to see if it is within the span of the branch.

INDEXED, NO OFFSET - In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. 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. This addressing mode is useful in 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, and the byte following the opcode specifies the direct address of the byte in 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, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they 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 and condition (set or clear) which is to be tested is included in the opcode, and the address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset is in the third byte and is added to the value of the PC if the branch condition is true. 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.

Caution

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

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 control instruction with no other arguments, are included in this mode. These instructions are one byte long.

INSTRUCTION SET

The EF6805P2 MCU has a set of 59 basic instructions, which when combined with the 10 addressing modes produce 207 usable opcodes. 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 jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operands. Refer to table 1.

READ-MODIFY-WRITE MODIFICATIONS - 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 included in read-modify-write instructions through it does not perform the write. Rfer to table 2.

Caution

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

BRANCH INSTRUCTIONS - The branch instructions cause a branch from the program when a certain condition is met. Refer to table 3.

BIT MANIPULATION INSTRUCTIONS - These instructions are used on any bit in the first 256 bytes of the memory. One group either sets or clears. The other group performs the bit test branch operations. Refer to table 4.

Caution

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

CONTROL INSTRUCTIONS - The control instructions control the MCU operations during program execution. Refer to table 5.

ALPHABETICAL LISTING - The complete instruction set is given in alphabetical order in table 6.

OPCODE MAP SUMMARY - Table 7 is an opcode map for the instructions used on the MCU.

Table 1 : Register/Memory Instructions.

Function	Mnemonic	Addressing Modes: Immediate Op Code	Immediate # Bytes	Immediate # Cycles	Direct Op Code	Direct # Bytes	Direct # Cycles	Extended Op Code	Extended # Bytes	Extended # Cycles	Indexed (No Offset) Op Code	Indexed (No Offset) # Bytes	Indexed (No Offset) # Cycles	Indexed (8 Bit Offset) Op Code	Indexed (8 Bit Offset) # Bytes	Indexed (8 Bit Offset) # Cycles	Indexed (16 Bit Offset) Op Code	Indexed (16 Bit Offset) # Bytes	Indexed (16 Bit Offset) # Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DF	3	6
Store A in Memory	STA	—	—	—	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	—	—	—	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	—	—	—	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	—	—	—	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Table 2 : Read-Modify-Write Instructions.

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

Table 3 : Branch Instructions.

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
(branch IFF higher or same)	(BHS)	24	2	4
Branch IFF Carry Set	BCS	25	2	4
(branch IFF lower)	(BLO)	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF interrupt mask bit is clear.	BMC	2C	2	4
Branch IFF interrupt mask bit is set.	BMS	2D	2	4
Branch IFF interrupt line is low.	BIL	2E	2	4
Branch IFF interrupt line is high.	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Table 4 : 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	10
Branch IFF Bit n is clear	BRCLR n (n = 0... 7)				01 + 2 • n	3	10
Set Bit n	BSET n (n = 0... 7)	10 + 2 • n	2	7			
Clear Bit n	BCLR n (n = 0... 7)	11 + 2 • n	2	7			

Table 5 : 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	11
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

Table 6 : Instruction Set.

	Addressing Modes										Condition Code				
Mnem	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			^	●	^	^	^
ADD		X	X	X		X	X	X			^	●	^	^	^
AND		X	X	X		X	X	X			●	●	^	^	●
ASL	X		X			X	X				●	●	^	^	^
ASR	X		X			X	X				●	●	^	^	^
BCC					X						●	●	●	●	●
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			●	●	^	^	●
BLO					X						●	●	●	●	●
BLS					X						●	●	●	●	●
BMC					X						●	●	●	●	●
BMI					X						●	●	●	●	●
BMS					X						●	●	●	●	●
BNE					X						●	●	●	●	●
BPL					X						●	●	●	●	●
BRA					X						●	●	●	●	●
BRN					X						●	●	●	●	●
BRCLR										X	●	●	●	●	^
BRSET										X	●	●	●	●	^
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLL	X										●	●	●	●	0

Table 6 : Instruction Set (continued).

	Addressing Modes										Condition Code				
Mnem	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
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	^	^	^
COM	X		X			X	X				●	●	^	^	1
CPX		X	X	X		X	X	X			●	●	^	^	^
DEC	X		X			X	X				●	●	^	^	●
EOR		X	X	X		X	X	X			●	●	^	^	●
INC	X		X			X	X				●	●	^	^	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	^	^	●
LDX		X	X	X		X	X	X			●	●	^	^	●
LSL	X		X			X	X				●	●	^	^	^
LSR	X		X			X	X				●	●	0	^	^
NEQ	X		X			X	X				●	●	^	^	^
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	^	^	●
ROL	X		X			X	X				●	●	^	^	^
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	^	^	^
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	^	^	●
STX			X	X		X	X	X			●	●	^	^	●
SUB		X	X	X		X	X	X			●	●	^	^	^
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	^	^	●
TXA	X										●	●	●	●	●

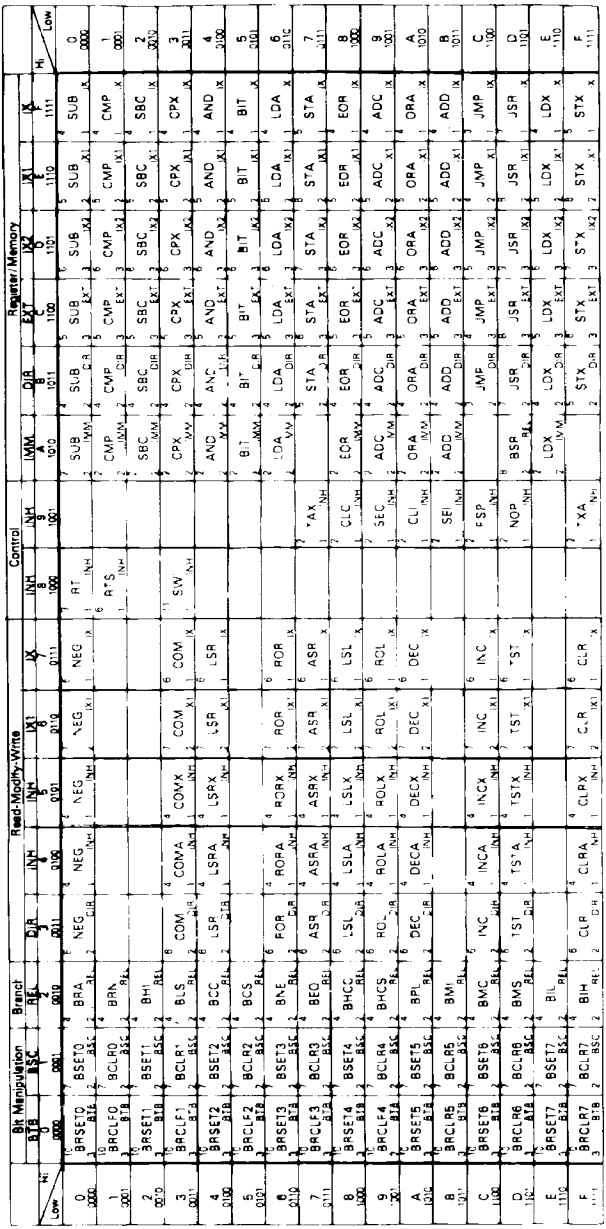
Condition Code Symbols :

H	Half Carry (from bit 3)	Z	Zero
I	Interrupt Mask	C	Carry/borrow
N	Negative (sign bit)	^	Test and Set if True, Cleared Otherwise
		•	Not Affected

HMOS 6805 FAMILY

Features	EF6805P2	EF6805P6	EF6805R2	EF6805R3	EF6805U2	EF6805U3
Technology	HMOS	HMOS	HMOS	HMOS	HMOS	HMOS
Number of Pins	28	28	40	40	40	40
On-chip RAM (bytes)	64	64	64	112	64	112
On-chip User ROM (bytes)	1100	1796	2048	3776	2048	3776
External Bus	None	None	None	None	None	None
Bidirectional I/O Lines	20	20	24	24	24	24
Unidirectional I/O Lines	None	None	6 Inputs	6 Inputs	8 Inputs	8 Inputs
Other I/O Features	Timer	Timer	Timer, A/D	Timer, A/D	Timer	Timer
External Interrupt Inputs	1	1	2	2	2	2
STOP and WAIT	No	No	No	No	No	No

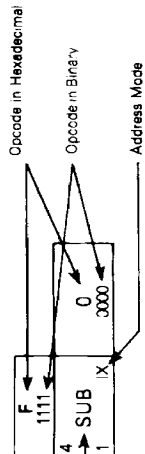
Table 7 : 6805 HMOS Family Opcode MAP.



Abbreviations for Address Modes

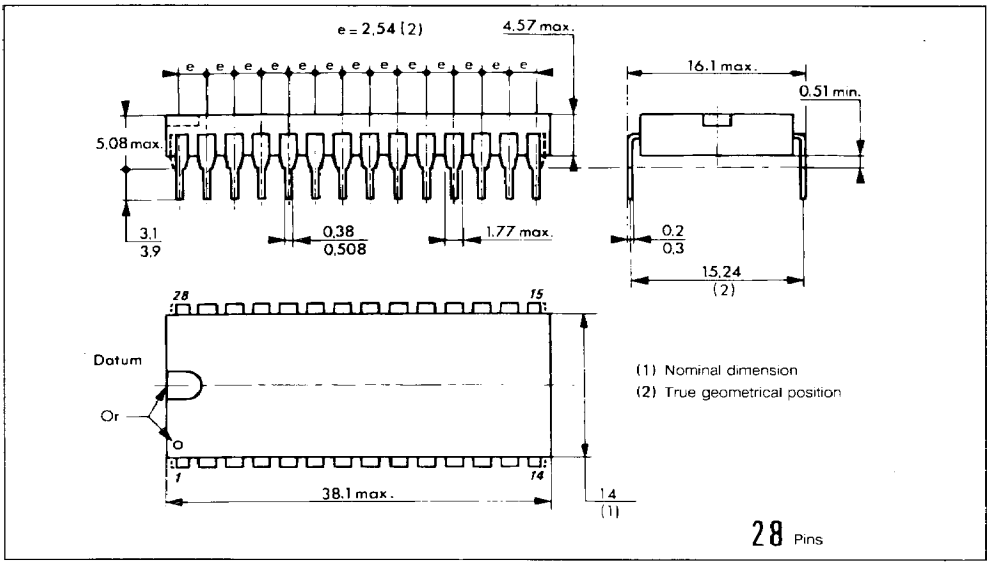
INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND

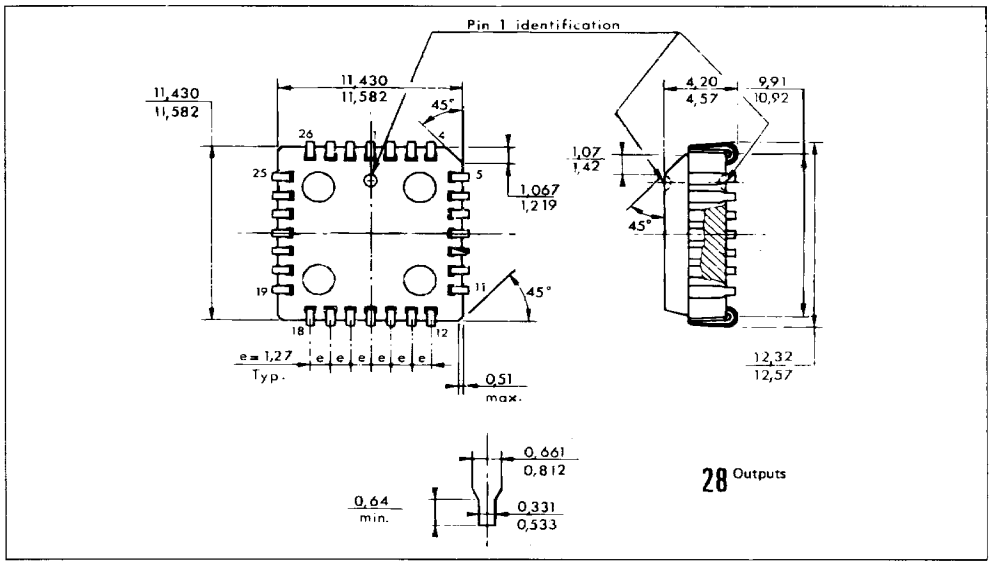


PACKAGE MECHANICAL DATA

CB–132 PLASTIC PACKAGE



CB–520 PLCC28



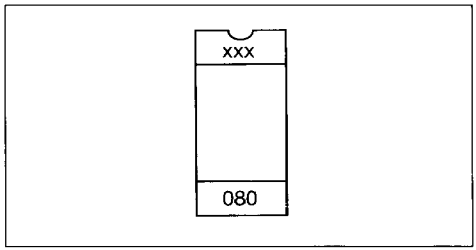
ORDERING INFORMATION

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to SGS-THOMSON on EPROM(s) or an EFDOS/MDOS* disk file.

To initiate a ROM pattern for the MCU, it is necessary to first contact your local SGS-THOMSON representative or distributor.

EPROMs

One 2716 or 2732 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation.



XXX = Customer ID)

After the EPROM is marked, it should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

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 SGS-THOMSON. The signed verification form constitutes the contractual agreement for creation of the customer mask. If desired, SGS-THOMSON will program on blank EPROM from the data file used to create the custom mask and aid in the verification process.

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 they are usually unmarked, packaged in ceramic, and tested only at room temperature and 5 volts. These RVUs are included in the mask change and are not production parts. The RVUs are thus not guaranteed by SGS THOMSON. Quality Assurance, and should be discarded after verification is completed.

FLEXIBLE DISKS

The disk media submitted must be single-sided, EFDOS/MDOS* compatible floppies.

The customer must write the binary file name and company name on the disk with a felt-tip-pen. The minimum EFDOS/MDOS* system files, as well as the absolute binary object file (Filename .LO type of file) from the 6805 cross assembler, must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files : filename .LX (DEVICE/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 the user-to-factory interface if the user finds any software errors and needs assistance quickly from SGS-THOMSON factory representatives.

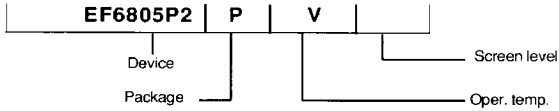
EFDOS is SGS-THOMSON Disk Operating System available on development systems such as DEVICE...

MDOS* is MOTOROLA's Disk Operating System available on development systems such as EXORciser...

* Requires prior factory approval.

Whenever ordering a custom MCU is required, please contact your local SGS-THOMSON representative or SGS-THOMSON distributor and/or complete and send the attached "MCU customer ordering sheet" to your local SGS-THOMSON Microelectronics representative.

ORDER CODES



The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

Device	Package					Oper. Temp			Screening Level			
	C		P		FN	L*	V	T	Std	D		
EF6805P2 (1.0MHz)			●		●	●	●		●	●		
Examples : EF6805P2P, EF6805P2FN, EF6805P2PLD, EF6805P2FNLD.												

Package : C : Ceramic DIL, **J** : Cerdip DIL, **P** : Plastic DIL, **E** : LCCC, **FN** : PLCC
Oper. temp. : **L*** : 0°C to + 70°C, **V** : – 40 °C to + 85°C, **T** : – 40°C to + 105°C, * : may be omitted.
Screening level : Std : (no-end suffix), **D** : NFC 96883 level D,
EXORciser is a registered trademark of MOTOROLA Inc.

EF6805 FAMILY - MCU CUSTOMER ORDERING SHEET

Commercial reference : | | | | | | | | | |

Customer's marking : | | | | | | | | |

Application : .....
.....

Customer name : .....
Company : .....
Address : .....
Phone : .....

Specification reference ;
❑ SGS-THOMSON Microelectronic reference
.....
❑ Special customer data reference*
.....

ROM capacity required : | | | | bytes

Temperature range :
❑ 0°C / + 70°C
❑ – 40°C / + 85°C
❑ – 40°C / + 105°C

Quality level :
❑ STD
❑ D
❑ Other* (customer's quality specification ref.) :
.....

Package
❑ Plastic
❑ PLCC

Software developped by :
❑ SGS-THOMSON Microelectronic application lab.
❑ External lab.
❑ Customer

PATTERN MEDIA (a listing may be supplied in addition for checking purpose) :
❑ EPROM Reference :
❑ EFDOS/MDOS* disk file
  ❑ 8" floppy
  ❑ 5" 1/4 floppy
❑ Other *

OPTION LIST

-Oscillator input
❑ Xtal
❑ RC

- Port A output drive :
❑ CMOS and TTL
❑ TTL only

- Timer Prescaler
❑ 2^0 (divide by 1)
❑ 2^1 (divide by 2)
❑ 2^2 (divide by 4)
❑ 2^3 (divide by 8)
❑ 2^4 (divide by 16)
❑ 2^5 (divide by 32)
❑ 2^6 (divide by 64)
❑ 2^7 (divide by 128)

- Internal max. clock frequency :
❑ 1.0 MHz

- Low voltage inhibit :
❑ Enabled
❑ Disabled

- Timer clock source :
❑ Internal φ2 clock
❑ TIMER input pin

Yearly quantity forecast

- start of production date :
- for a shipment period of :

CUSTOMER CONTACT NAME : DATE : SIGNATURE :