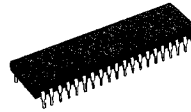



**SGS-THOMSON**  
 MICROELECTRONICS

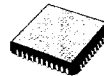
**EF6801**  
**EF6803**

## MICROCOMPUTER/MICROPROCESSOR (MCU/MPU)

- ENHANCED EF6800 INSTRUCTION SET
- 8 X 8 MULTIPLY INSTRUCTION
- SERIAL COMMUNICATIONS INTERFACE (SCI)
- UPWARD SOURCE AND OBJECT CODE COMPATIBILITY WITH THE 6800
- 16-BIT THREE-FUNCTION PROGRAMMABLE TIMER
- SINGLE-CHIP OR EXPANDED OPERATION TO 64K BYTE ADDRESS SPACE
- BUS COMPATIBILITY WITH THE 6800 FAMILY
- 2048 BYTES OF ROM (EF6801)
- 128 BYTES OF RAM
- 64 BYTES OF RAM RETAINABLE DURING POWERDOWN
- 29 PARALLEL I/O AND TWO HANDSHAKE CONTROL LINES
- INTERNAL CLOCK GENERATOR WITH DIVIDE-BY-FOUR OUTPUT
- - 40°C TO + 85°C TEMPERATURE RANGE
- - 40°C TO + 105°C TEMPERATURE RANGE

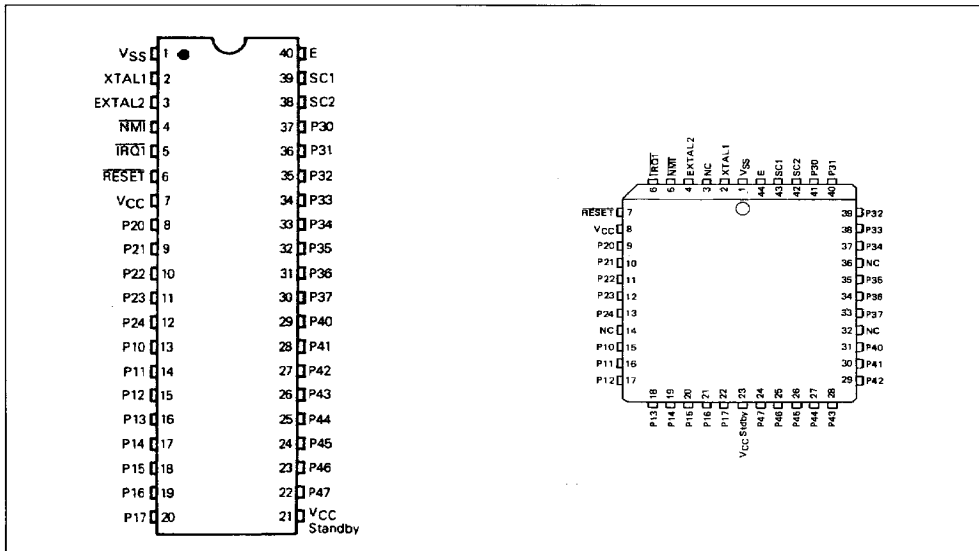


**P SUFFIX**  
(Plastic Package)



**FN SUFFIX**  
(PLCC44)

### PIN CONNECTIONS

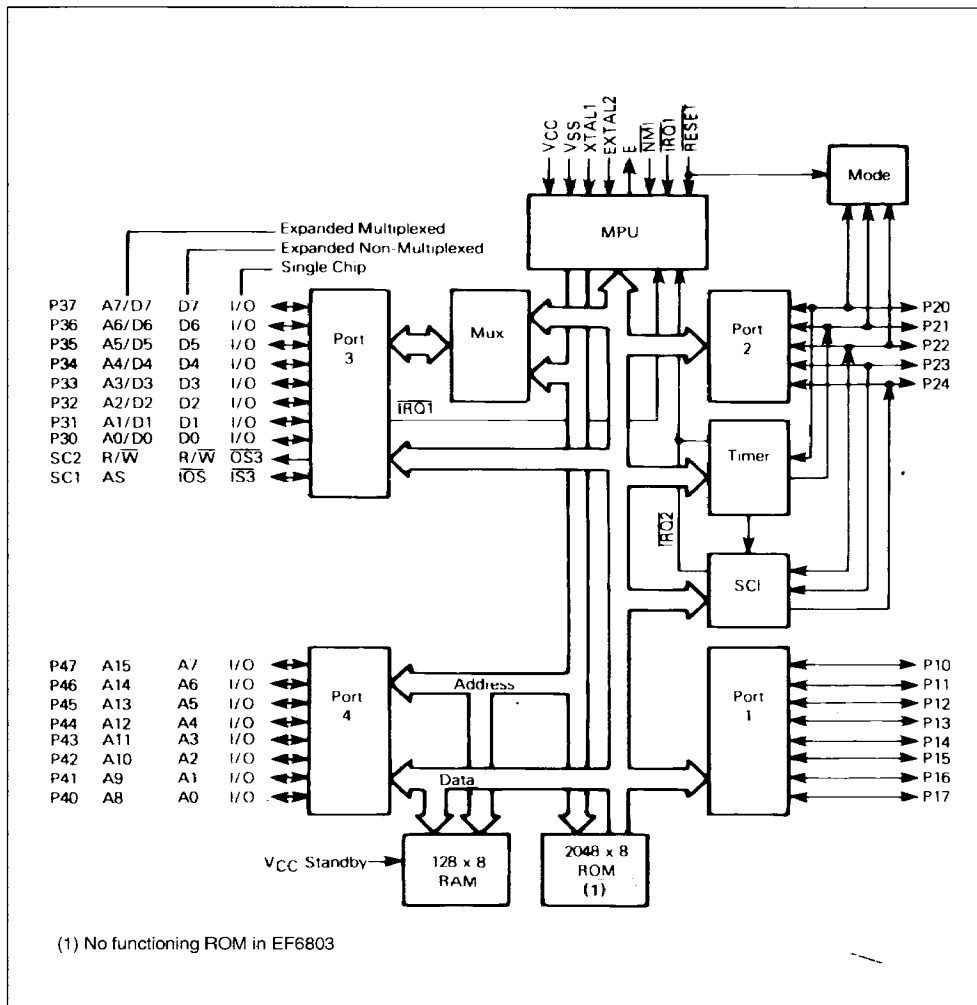


# DESCRIPTION

The EF6801 is an 8-bit single-chip microcomputer unit (MCU) which significantly enhances the capabilities of the 6800 family of parts. It includes an upgraded 6800 microprocessor unit (MPU) with upward-source and object-code compatibility. Execution times of key instructions have been improved and several new instructions have been added including an unsigned multiply. The MCU can

function as a monolithic microcomputer or can be expanded to a 64K byte address space. It is TTL compatible and requires one + 5V power supply. On-chip resources include 2048 bytes of ROM, 128 bytes of RAM, a Serial Communications Interface (SCI), parallel I/O, and a three function Programmable Timer. The EF6803 can be considered as an EF6801 operating in Modes 2 or 3. EF6801 MCU Family features include :

**Figure 1 : 6801/6803 Block Diagram.**



## POWER CONSIDERATIONS

The average chip-junction temperature,  $T_J$ , in  $^{\circ}\text{C}$  can be obtained from :

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

Where :

$T_A \equiv$  Ambient Temperature,  $^{\circ}\text{C}$

$\theta_{JA} \equiv$  Package Thermal Resistance, Junction to-Ambient,  $^{\circ}\text{C/W}$

$P_D \equiv P_{INT} + P_{PORT}$

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

$P_{PORT} \equiv$  Port Power Dissipation, Watts - User Determined

For most applications  $P_{PORT} \ll P_{INT}$  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 :

$$P_D = K + (T_J + 273^{\circ}\text{C}) \quad (2)$$

Solving equations 1 and 2 for  $K$  gives :

$$K = P_D \cdot (T_A + 273^{\circ}\text{C}) + \theta_{JA} \cdot P_D^2 \quad (3)$$

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

## ABSOLUTE MAXIMUM RATINGS

| Symbol    | Parameter                                                                                                                                               | Value                                                  | Unit               |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|--------------------|
| $V_{CC}$  | Supply Voltage                                                                                                                                          | - 0.3 to + 7.0                                         | V                  |
| $V_{in}$  | Input Voltage                                                                                                                                           | - 0.3 to + 7.0                                         | V                  |
| $T_A$     | Operating Temperature Range<br>EF6801/03, EF6801/03-1, EF68A01/03, EF68B01/03<br>EF6801/03, EF6801/03-1 : V Suffix<br>EF6801/03, EF6801/03-1 : A Suffix | $T_L$ to $T_H$<br>0 to 70<br>- 40 to 85<br>- 40 to 105 | $^{\circ}\text{C}$ |
| $T_{stg}$ | Storage Temperature Range                                                                                                                               | - 55 to + 150                                          | $^{\circ}\text{C}$ |

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  $V_{SS} \leq (V_{IN} \text{ or } V_{OUT}) \leq V_{CC}$ . Input protection is enhanced by connecting unused inputs to either  $V_{DD}$  or  $V_{SS}$ .

## THERMAL DATA

|               |                                 |           |                      |
|---------------|---------------------------------|-----------|----------------------|
| $\theta_{JA}$ | Thermal Resistance Plastic PLCC | 50<br>100 | $^{\circ}\text{C/W}$ |
|---------------|---------------------------------|-----------|----------------------|

## CONTROL TIMING ( $V_{CC} = 5.0\text{V} \pm 5\%$ , $V_{SS} = 0$ , $T_A = 0$ to $70^{\circ}\text{C}$ )

| Symbol     | Parameter                        | EF6801 |      | EF6801-1 |      | EF68A01 |      | EF68B01 |      | Unit |
|------------|----------------------------------|--------|------|----------|------|---------|------|---------|------|------|
|            |                                  | Min.   | Max. | Min.     | Max. | Min.    | Max. | Min.    | Max. |      |
| $f_o$      | Frequency of Operation           | 0.5    | 1.0  | 0.5      | 1.25 | 0.5     | 1.5  | 0.5     | 2.0  | MHz  |
| $f_{XTAL}$ | Crystal Frequency                | 2.0    | 4.0  | 2.0      | 5.0  | 2.0     | 6.0  | 2.0     | 8.0  | MHz  |
| $4f_o$     | External Oscillator Frequency    | 2.0    | 4.0  | 2.0      | 5.0  | 2.0     | 6.0  | 2.0     | 8.0  | MHz  |
| $t_{rc}$   | Crystal Oscillator Start Up Time |        | 100  |          | 100  |         | 100  |         | 100  | ms   |
| $t_{PCS}$  | Processor Control Setup Time     | 200    |      | 170      |      | 140     |      | 110     |      | ns   |

## DC ELECTRICAL CHARACTERISTICS

(V<sub>CC</sub> = 5.0Vdc ± 5%, V<sub>SS</sub> = 0, T<sub>A</sub> = T<sub>L</sub> to T<sub>H</sub> unless otherwise noted)

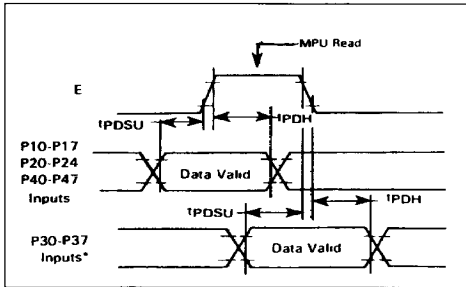
| Symbol                              | Parameter                                                                                                                                                           | EF6801/03<br>0°C to +70°C                      |                                    | EF6801/03<br>-40°C to<br>+85°C / +105°C        |                                    | Unit |
|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|------------------------------------|------------------------------------------------|------------------------------------|------|
|                                     |                                                                                                                                                                     | Min.                                           | Max.                               | Min.                                           | Max.                               |      |
| V <sub>IH</sub>                     | Input High Voltage<br>RESET<br>Other Inputs                                                                                                                         | V <sub>SS</sub> + 4.0<br>V <sub>SS</sub> + 2.0 | V <sub>CC</sub><br>V <sub>CC</sub> | V <sub>SS</sub> + 4.0<br>V <sub>SS</sub> + 2.2 | V <sub>CC</sub><br>V <sub>CC</sub> | V    |
| V <sub>IL</sub>                     | Input Low Voltage<br>All Inputs                                                                                                                                     | V <sub>SS</sub> - 0.3                          | V <sub>SS</sub> + 0.8              | V <sub>SS</sub> - 0.3                          | V <sub>SS</sub> + 0.8              | V    |
| I <sub>in</sub>                     | Input Load Current (V <sub>in</sub> = 0 to 2.4V)<br>Port 4<br>SCI                                                                                                   |                                                | 0.5<br>0.8                         |                                                | 0.8<br>1.0                         | mA   |
| I <sub>in</sub>                     | Input Leakage Current<br>(V <sub>in</sub> = 0 to 5.25V)<br>NMI, IRQ1, RESET                                                                                         |                                                | 2.5                                |                                                | 5.0                                | μA   |
| I <sub>TSI</sub>                    | Hi-Z (off-state) Input Current<br>(V <sub>in</sub> = 0.5 to 2.4V)<br>Ports 1, 2, and 3                                                                              |                                                | 10                                 |                                                | 20                                 | μA   |
| V <sub>OH</sub>                     | Output High Voltage<br>(I <sub>Load</sub> = -65μA, V <sub>CC</sub> = Min)* E, Port 4, SC1, SC2<br>(I <sub>Load</sub> = -100μA, V <sub>CC</sub> = Min) Other Outputs | V <sub>SS</sub> + 2.4<br>V <sub>SS</sub> + 2.4 |                                    | V <sub>SS</sub> + 2.4<br>V <sub>SS</sub> + 2.4 |                                    | V    |
| V <sub>OL</sub>                     | Output Low Voltage<br>(I <sub>Load</sub> = 2.0mA, V <sub>CC</sub> = Min) All Outputs                                                                                |                                                | V <sub>SS</sub> + 0.5              |                                                | V <sub>SS</sub> + 0.6              | V    |
| I <sub>OH</sub>                     | Darlington Drive Current (V <sub>O</sub> = 1.5V) Port 1                                                                                                             | 1.0                                            | 4.0                                | 1.0                                            | 5.0                                | mA   |
| P <sub>INT</sub>                    | Internal Power Dissipation<br>(measured at T <sub>A</sub> = T <sub>L</sub> in steady-state operation)                                                               |                                                | 1200                               |                                                | 1500                               | mW   |
| C <sub>in</sub>                     | Input Capacitance<br>(V <sub>in</sub> = 0, T <sub>A</sub> = 25°C,<br>f <sub>o</sub> = 1.0MHz)<br>Port 3, Port 4, SC1<br>Other Inputs                                |                                                | 12.5<br>10                         |                                                | 12.5<br>10                         | pF   |
| V <sub>SBB</sub><br>V <sub>SB</sub> | V <sub>CC</sub> Standby<br>Powerdown<br>Powerup                                                                                                                     | 4.0<br>4.75                                    | 5.25<br>5.25                       | 4.0<br>4.75                                    | 5.25<br>5.25                       | V    |
| I <sub>SBB</sub>                    | Standby Current<br>Powerdown                                                                                                                                        |                                                | 6.0                                |                                                | 8.0                                | mA   |

\* Negotiable to -100μA (for further information contact the factory)

## PERIPHERAL PORT TIMING (refer to figures 2-5)

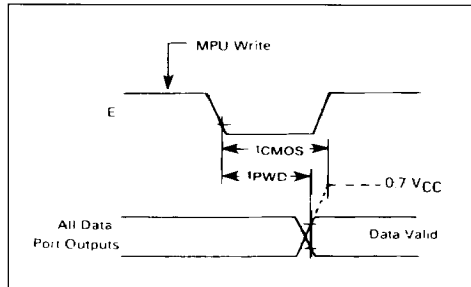
| Symbol     | Parameter                                                               | EF6801<br>EF6803 |      | EF6801-1<br>EF6803-1 |      | EF68A01<br>EF68A03 |      | EF68B01<br>EF68B03 |      | Unit    |
|------------|-------------------------------------------------------------------------|------------------|------|----------------------|------|--------------------|------|--------------------|------|---------|
|            |                                                                         | Min.             | Max. | Min.                 | Max. | Min.               | Max. | Min.               | Max. |         |
| $t_{PDSU}$ | Peripheral Data Setup Time                                              | 200              |      | 200                  |      | 150                |      | 100                |      | ns      |
| $t_{PDH}$  | Peripheral Data Hold Time                                               | 200              |      | 200                  |      | 150                |      | 100                |      | ns      |
| $t_{OSD1}$ | Delay Time, Enable Positive Transition to OS3<br>Negative Transition    |                  | 350  |                      | 350  |                    | 300  |                    | 250  | ns      |
| $t_{OSD2}$ | Delay Time, Enable Positive Transition to OS3<br>Positive Transition    |                  | 350  |                      | 350  |                    | 300  |                    | 250  | ns      |
| $t_{PWD}$  | Delay Time, Enable Negative Transition to<br>Peripheral Data Valid      |                  | 350  |                      | 350  |                    | 300  |                    | 250  | ns      |
| $t_{CMOS}$ | Delay Time, Enable Negative Transition to<br>Peripheral CMOS Data Valid |                  | 2.0  |                      | 2.0  |                    | 2.0  |                    | 2.0  | $\mu s$ |
| $t_{PWIS}$ | Input Strobe Pulse Width                                                | 200              |      | 200                  |      | 150                |      | 100                |      | ns      |
| $t_{IH}$   | Input Data Hold Time                                                    | 50               |      | 50                   |      | 40                 |      | 30                 |      | ns      |
| $t_{IS}$   | Input Data Setup Time                                                   | 20               |      | 20                   |      | 20                 |      | 20                 |      | ns      |

Figure 2 : Data Setup and Hold Times (MPU read).



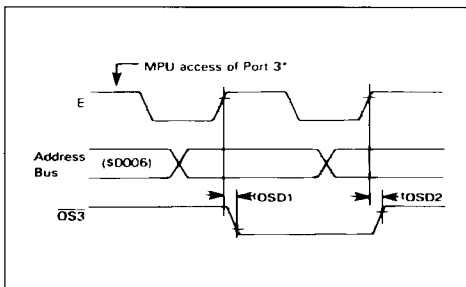
\* Port 3 non-latched operation (latch enable - 0)

Figure 3 : Data Setup and Hold Times (MPU write).



Notes : 1. 10 k Pullup resistor required for Port 2 to reach 0.7 V<sub>CC</sub>.  
 2. Not applicable to P21.  
 3. Port 4 cannot be pulled above V<sub>CC</sub>.

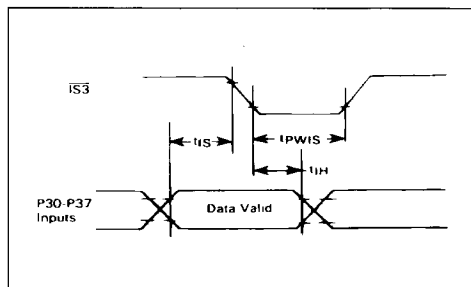
Figure 4 : Port 3 Output Strobe Timing (EF6801 single-chip Mode).



\* Access matches Output Strobe Select (OSS = 0, a read. OSS = 1, a write).

Note : Timing measurements are referenced to and from a low voltage of 0.8V and a high voltage of 2.0V, unless otherwise noted.

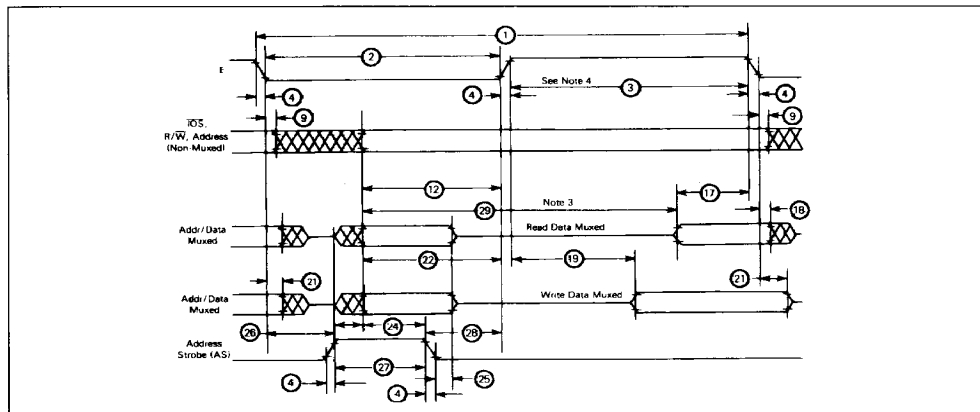
Figure 5 : Port 3 Latch Timing (EF6801 single-chip mode).



**BUS TIMING** (see notes 1 and 2)

| Ident. Number | Symbol     | Parameter                            | EF6801<br>EF6803 |      | EF6801-1<br>EF6803-1 |      | EF68A01<br>EF68A03 |      | EF68B01<br>EF68B03 |      | Unit    |
|---------------|------------|--------------------------------------|------------------|------|----------------------|------|--------------------|------|--------------------|------|---------|
|               |            |                                      | Min.             | Max. | Min.                 | Max. | Min.               | Max. | Min.               | Max. |         |
| 1             | $t_{cyc}$  | Cycle Time                           | 1.0              | 2.0  | 0.8                  | 2.0  | 0.667              | 2.0  | 0.5                | 2.0  | $\mu s$ |
| 2             | $PW_{EL}$  | Pulse Width, E Low                   | 430              | 1000 | 360                  | 1000 | 300                | 1000 | 210                | 1000 | ns      |
| 3             | $PW_{EH}$  | Pulse Width, E High                  | 450              | 1000 | 360                  | 1000 | 300                | 1000 | 220                | 1000 | ns      |
| 4             | $t_r, t_f$ | Clock Rise and Fall Time             |                  | 25   |                      | 25   |                    | 25   |                    | 20   | ns      |
| 9             | $t_{AH}$   | Address Hold Time                    | 20               |      | 20                   |      | 20                 |      | 10                 |      | ns      |
| 12            | $t_{AV}$   | Non-muxed Address Valid Time to E*   | 200              |      | 150                  |      | 115                |      | 70                 |      | ns      |
| 17            | $t_{DSR}$  | Read Data Setup Time                 | 80               |      | 70                   |      | 60                 |      | 40                 |      | ns      |
| 18            | $t_{DHR}$  | Read Data Hold Time                  | 10               |      | 10                   |      | 10                 |      | 10                 |      | ns      |
| 19            | $t_{DDW}$  | Write Data Delay Time                |                  | 225  |                      | 200  |                    | 170  |                    | 120  | ns      |
| 21            | $t_{DHW}$  | Write Data Hold Time                 | 20               |      | 20                   |      | 20                 |      | 10                 |      | ns      |
| 22            | $t_{AVM}$  | Muxed Address Valid Time to E Rise*  | 200              |      | 150                  |      | 115                |      | 80                 |      | ns      |
| 24            | $t_{ASL}$  | Muxed Address Valid Time to AS Fall* | 60               |      | 50                   |      | 40                 |      | 20                 |      | ns      |
| 25            | $t_{AHL}$  | Muxed Address Hold Time              | 20               |      | 20                   |      | 20                 |      | 10                 |      | ns      |
| 26            | $t_{ASD}$  | Delay Time, E to AS Rise*            | 90**             |      | 70**                 |      | 60**               |      | 45**               |      | ns      |
| 27            | $PW_{ASH}$ | Pulse Width, AS High*                | 220              |      | 170                  |      | 140                |      | 110                |      | ns      |
| 28            | $t_{ASED}$ | Delay Time, AS to E Rise*            | 90               |      | 70                   |      | 60                 |      | 45                 |      | ns      |
| 29            | $t_{ACC}$  | Usable Access Time*                  | 595              |      | 465                  |      | 380                |      | 270                |      | ns      |

\* At specified cycle time.

\*\*  $t_{ASD}$  parameters listed assume external TTL clock drive with  $50\% \pm 5\%$  duty cycle. Devices driven by an external TTL clock with  $50\% \pm 1\%$  duty cycle or which use a crystal have the following  $t_{ASD}$  specifications : 100ns min. (1.0MHz devices), 80ns min. (1.25MHz devices), 65ns min. (1.5MHz devices), 50ns min. (2.0MHz devices).**Figure 6 : Bus Timing.**

- Notes :**
1. Voltage levels shown are  $V_L \leq 0.5V$ ,  $V_H \geq 2.4V$ , unless otherwise specified.
  2. Measurement points shown are 0.8V and 2.0V, unless otherwise specified.
  3. Usable access time is computed by  $12 + 3 - 17 + 4$ .
  4. Memory devices should be enabled only during E high to avoid Port 3 bus contention.

Figure 7 : CMOS Load.

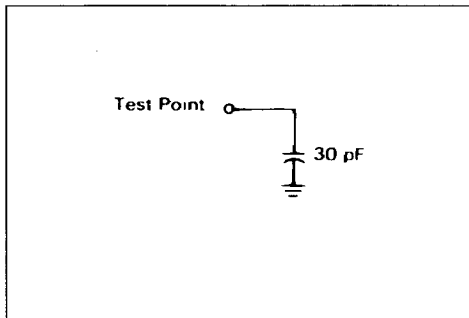
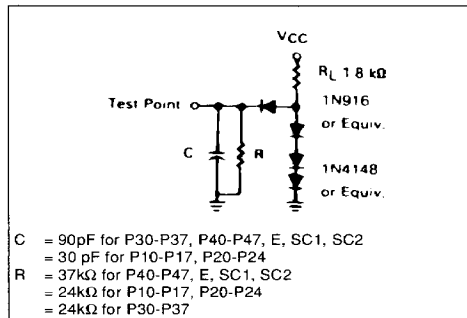


Figure 8 : Timing Test Load Ports 1, 2, 3, 4.



## INTRODUCTION

The EF6801 is an 8-bit monolithic microcomputer which can be configured to function in a wide variety of applications. The facility which provides this extraordinary flexibility is its ability to be hardware programmed into eight different operating modes. The operating mode controls the configuration of 18 of the 40 MCU pins, available on-chip resources, memory map, location (internal or external) of interrupt vectors, and type of external bus. The configuration of the remaining 22 pins is not dependent on the operating mode.

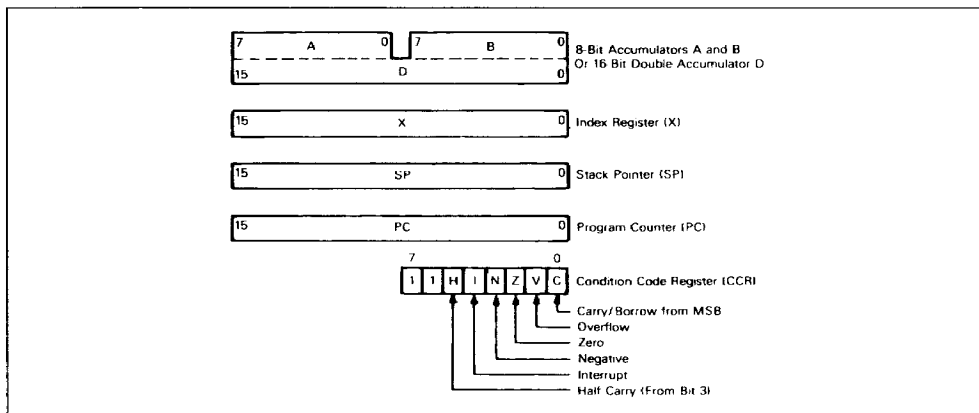
Twenty-nine pins are organized as three 8-bit ports and one 5-bit port. Each port consists of at least a Data Register and a write-only Data Direction Register. The Data Direction Register is used to define whether corresponding bits in the Data Register are configured as an input (clear) or output (set).

The term "port", by itself, refers to all of the hardware associated with the port. When the port is used as a "data port" or "I/O port", it is controlled by the port Data Direction Register and the programmer has direct access to the port pins using the port Data Register. Port pins are labeled as  $P_{ij}$  where  $i$  identifies one of four ports and  $j$  indicates the particular bit.

The Microprocessor Unit (MPU) is an enhanced EF6800 MPU with additional capabilities and greater throughput. It is upward source and object code compatible with the EF6800. The programming model is depicted in figure 9, where Accumulator D is a concatenation of Accumulators A and B. A list of new operations added to the 6800 instruction set are shown in table 1.

The EF6803 can be considered an EF6801 that operates in Modes 2 and 3 only.

Figure 9 : Programming Model.



## OPERATING MODES

The EF6801 provides eight different operating modes (modes 0 through 7), the EF6803 provides two operating modes (modes 2 and 3). The operating modes are hardware selectable and determine the device memory map, the configuration of Port 3, Port 4, SC1, SC2, and the physical location of the interrupt vectors.

## FUNDAMENTAL MODES

The eight operating modes can be grouped into three fundamental modes which refer to the type of bus it supports : Single Chip, Expanded Non-Multiplexed, and Expanded Multiplexed. Single-Chip modes include 4 and 7, Expanded Non-Multiplexed

is Mode 5 and the remaining five are Expanded Multiplexed modes. Table 2 summarizes the characteristics of the operating modes.

EF6801 SINGLE-CHIP MODES (4, 7). In the Single-Chip Mode, the four MCU ports are configured as parallel input/output data ports, as shown in figure 10. The MCU functions as a monolithic micro-computer in these two modes without external address or data buses. A maximum of 29 I/O lines and two Port 3 control lines are provided. Peripherals or another MCU can be interfaced to Port 3 in a loosely coupled dual processor configuration, as shown in figure 11.

**Table 1 : New Instructions.**

| Instruction  | Description                                                                                                                       |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------|
| ABX          | Unsigned addition of accumulator B to index register.                                                                             |
| ADDD         | Adds (without carry) the double accumulator to memory and leaves the sum in the double accumulator.                               |
| ASLD or LSLD | Shifts the double accumulator left (towards MSB) one bit ; the LSB is cleared and the MSB is shifted into the C-bit.              |
| BHS          | Branch if higher or same ; unsigned conditional branch (same as BCC).                                                             |
| BLO          | Branch if lower ; unsigned conditional branch (same as BCS).                                                                      |
| BRN          | Branch never.                                                                                                                     |
| JSR          | Additional addressing mode direct.                                                                                                |
| LDD          | Loads double accumulator from memory.                                                                                             |
| LSL          | Shifts memory or accumulator left (towards MSB) one bit ; the LSB is cleared and the MSB is shifted into the C-bit (same as ASL). |
| LSRD         | Shifts the double accumulator right (towards LSB) one bit ; the MSB is cleared and the LSB is shifted into the C-bit.             |
| MUL          | Unsigned multiply ; multiplies the two accumulators and leaves the product in the double accumulator.                             |
| PSHX         | Pushes the index register to stack.                                                                                               |
| PULX         | Pulls the index register from stack.                                                                                              |
| STD          | Stores the double accumulator to memory.                                                                                          |
| SUBD         | Subtracts memory from the double accumulator and leaves the difference in the double accumulator.                                 |
| CPX          | Internal processing modified to permit its use with any conditional branch instruction.                                           |

In Single-Chip Test Mode (4), the RAM responds to \$XX80 through \$XXFF and the ROM is removed from the internal address map. A test program must first be loaded into the RAM using modes 0, 1, 2, or 6. If the MCU is Reset and then programmed into Mode 4, execution will begin at \$XXFE : XXFF. Mode 5 can be irreversibly entered from Mode 4 without asserting RESET by setting bit 5 of the Port 2 Data Register. This mode is used primarily to test Ports 3 and 4 in the Single-Chip and Non-Multiplexed Modes.

EF6801 EXPANDED NON-MULTIPLEXED MODE (5). A modest amount of external memory space is

provided in the Expanded Non-Multiplexed Mode while significant on-chip resources are retained. Port 3 functions as an 8-bit bidirectional data bus and Port 4 is configured initially as an input data port. Any combination of the eight least-significant address lines may be obtained by writing to the Port 4 Data Direction Register. Stated alternatively, any combination of A0 to A7 may be provided while retaining the remainder as input data lines. Internal pullup resistors pull the Port 4 lines high until the port is configured.

Figure 12 illustrates a typical system configuration in the Expanded Non-Multiplexed Mode. The MCU

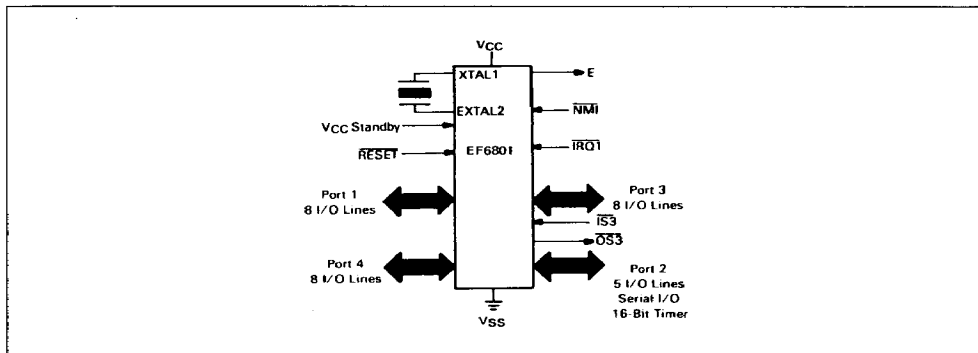
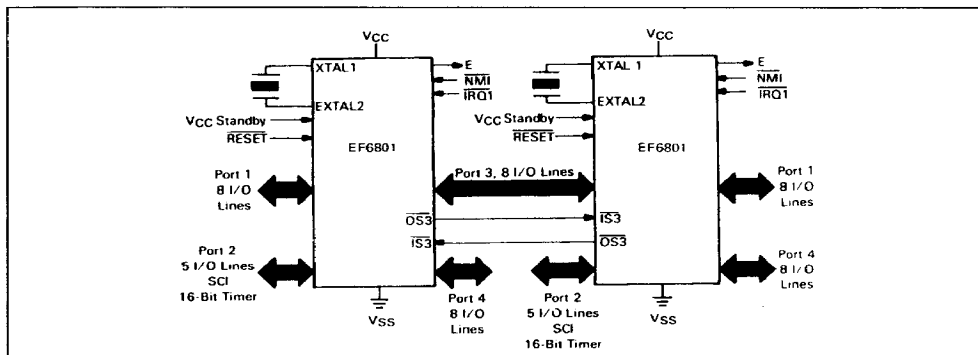
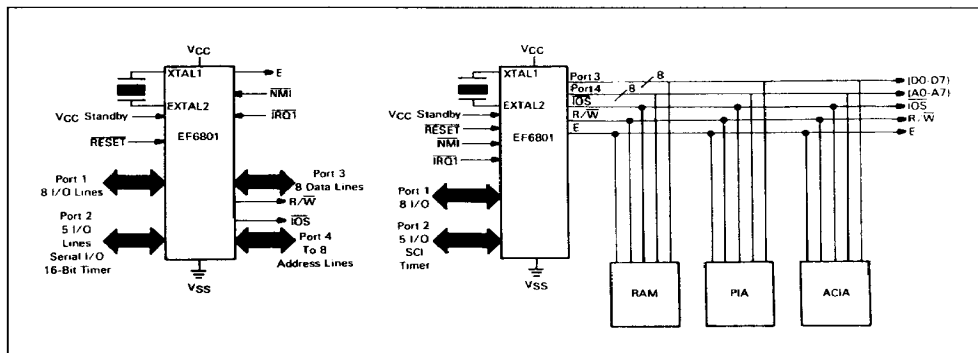
interfaces directly with 6800 family parts and can access 256 bytes of external address space at \$100 through \$1FF. IOS provides an address decode of

external memory (\$100-\$1FF) and can be used as a memory page select or chip select line.

**Table 2 : Summary of EF6801/03 Operating Modes.**

|                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Common to all Modes :</b><br>Reserved Register Area<br>Port 1<br>Port 2<br>Programmable Timer<br>Serial Communications Interface                                                                                                                                                                                                                                                                                                         |
| <b>Single-chip Mode 7 :</b><br>128 Bytes of RAM, 2048 Bytes of ROM<br>Port 3 is a parallel I/O port with two control lines.<br>Port 4 is a parallel I/O port.<br>SC1 is input strobe 3 (IS3).<br>SC2 is output strobe (OS3).                                                                                                                                                                                                                |
| <b>Expanded Non-multiplexed Mode 5 :</b><br>128 Bytes of RAM, 2048 Bytes of ROM<br>256 Bytes of External Memory Space<br>Port 3 is an 8-bit data bus.<br>Port 4 is an input port/address bus.<br>SC1 is input/output select (IOS).<br>SC2 is read/write (R/W).                                                                                                                                                                              |
| <b>Expanded Multiplexed Modes 1, 2, 3, 6* :</b><br>Four memory space options (64K address space).<br>(1) No Internal RAM or ROM (mode 3)<br>(2) Internal RAM, no ROM (mode 2)<br>(3) Internal RAM and ROM (mode 1)<br>(4) Internal RAM, ROM with Partial Address Bus (mode 6)<br>Port 3 is multiplexed address/data bus.<br>Port 4 is an address bus (inputs/address in mode 6).<br>SC1 is address strobe (AS).<br>SC2 is read/write (R/W). |
| <b>Test Modes 0 and 4 :</b><br>Expanded multiplexed test mode 0.<br>May be used to test RAM and ROM.<br>Single chip and non-multiplexed test mode 4.<br>(1) May be changed to mode 5 without going through reset.<br>(2) May be used to test ports 3 and 4 as I/O ports.                                                                                                                                                                    |

\* The EF6803 operates only in modes 2 and 3

**Figure 10** : Single-chip Mode.**Figure 11** : Single-chip Dual Processor Configuration.**Figure 12** : Expanded Non-multiplexed Configuration.

**EXPANDED-MULTIPLEXED MODES (0, 1, 2, 3, 6).** A 64K byte memory space is provided in the expanded multiplexed modes. In each of the expanded multiplexed modes Port 3 functions as a time multiplexed address/data bus with address valid on the negative edge of Address Strobe (AS), and data valid while E is high. In Modes 0 to 3, Port 4 provides address lines A8 to A15. In Mode 6, however, Port 4 initially is configured at RESET as an input data port. The port 4 Data Direction Register can then be changed to provide any combination of address lines, A8 to A15. Stated alternatively, any subset of A8 to A15 can be provided while retaining the remaining port 4 lines as input data lines. Internal pullup resistors pull the Port 4 lines high until software configures the port.

In Mode 0, the Reset vector is external for the first two E-cycles after the positive edge of RESET, and internal thereafter. In addition, the internal and external data buses are connected so there must be no memory map overlap in order to avoid potential bus conflicts. Mode 0 is used primarily to verify the ROM pattern and monitor the internal data bus with the automated test equipment.

Only the EF6801 can operate in each of the expanded multiplexed modes. The EF6803 operates only in Modes 2 and 3.

Figure 13 depicts a typical configuration for the Expanded-Multiplexed Modes. Address Strobe can be used to control a transparent D-type latch to capture addresses A0-A7, as shown in figure 14. This allows Port 3 to function as a Data Bus when E is high.

### PROGRAMMING THE MODE

The operating mode is determined at RESET by the levels asserted on P22, P21, and P20. These levels are latched into PC2, PC1, and PC0 of the program control register on the positive edge of RESET. The operating mode may be read from the Port 2 Data Register as shown below, and programming levels and timing must be met as shown in figure 15. A brief outline of the operating modes is shown in table 3.

| PORT 2 DATA REGISTER |     |     |     |     |     |     |        |
|----------------------|-----|-----|-----|-----|-----|-----|--------|
| 7                    | 6   | 5   | 4   | 3   | 2   | 1   | 0      |
| PC2                  | PC1 | PC0 | P24 | P23 | P22 | P21 | P20    |
|                      |     |     |     |     |     |     | \$0003 |

Circuitry to provide the programming levels is dependent primarily on the normal system usage of the three pins. If configured as outputs, the circuit shown in figure 16 may be used; otherwise, three-state buffers can be used to provide isolation while programming the mode.

**Table 3 : Mode Selection Summary.**

| Mode* | P22<br>PC2 | P21<br>PC1 | P20<br>PC0 | ROM              | RAM              | Interrupt<br>Vectors | Bus Mode               | Operating Mode                 |
|-------|------------|------------|------------|------------------|------------------|----------------------|------------------------|--------------------------------|
| 7     | H          | H          | H          | I                | I                | I                    | I                      | Single Chip                    |
| 6     | H          | H          | L          | I                | I                | I                    | MUX <sup>(5, 6)</sup>  | Multiplexed/partial Decode     |
| 5     | H          | L          | H          | I                | I                | I                    | NMUX <sup>(5, 6)</sup> | Non-multiplexed/partial Decode |
| 4     | H          | L          | L          | I <sup>(2)</sup> | I <sup>(1)</sup> | I                    | I                      | Single Chip Test               |
| 3     | L          | H          | H          | E                | E                | E                    | MUX <sup>(4)</sup>     | Multiplexed/no RAM or ROM      |
| 2     | L          | H          | L          | E                | I                | E                    | MUX <sup>(4)</sup>     | Multiplexed/RAM                |
| 1     | L          | L          | H          | I                | I                | E                    | MUX <sup>(4)</sup>     | Multiplexed/RAM & ROM          |
| 0     | L          | L          | L          | I                | I                | I <sup>(3)</sup>     | MUX <sup>(4)</sup>     | Multiplexed Test               |

#### Legend :

I - Internal

E - External

MUX - Multiplexed

NMUX - Non-Multiplexed

L - Logic "0"

H - Logic "1"

**Notes :** (1) Internal RAM is addressed at \$XX80

(2) Internal ROM is disabled

(3) RESET vector is external for 2 cycles after RESET goes high

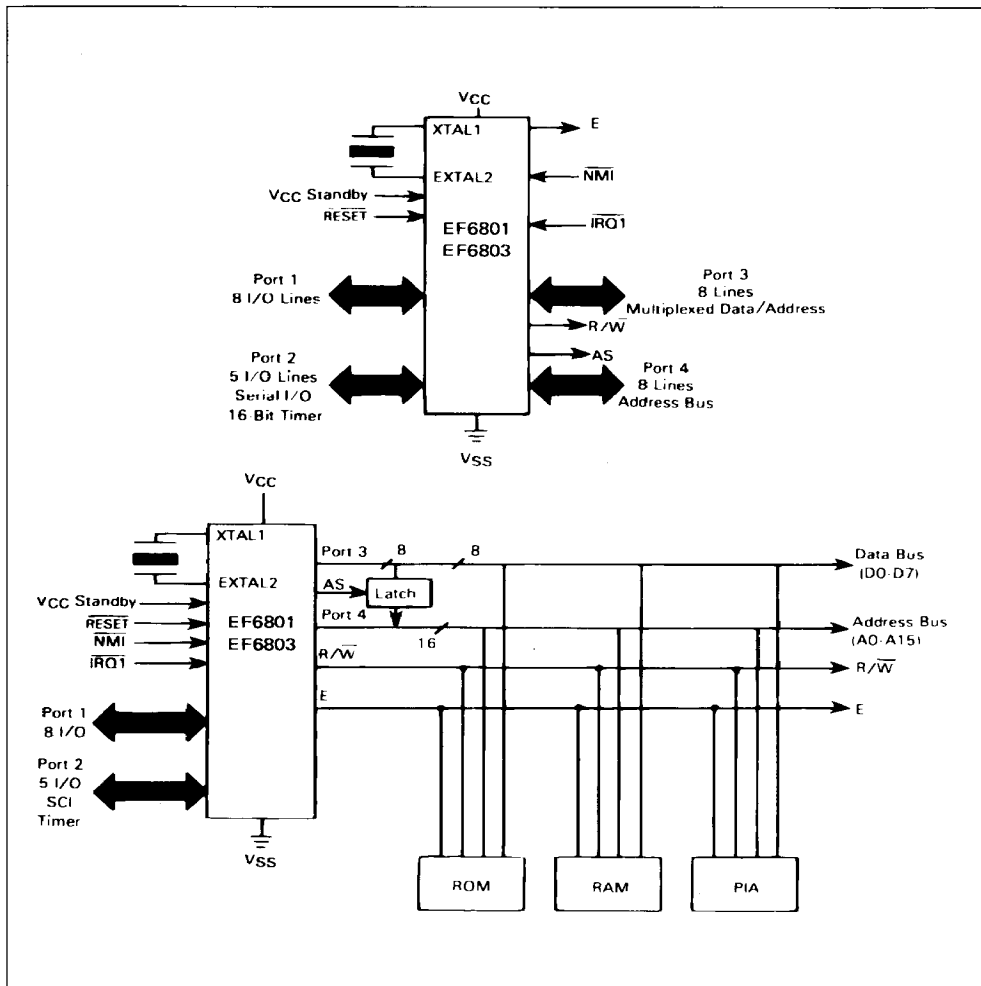
(4) Addresses associated with Ports 3 and 4 are considered external in Modes 0, 1, 2, and 3

(5) Addresses associated with Port 3 are considered external in Modes 5 and 6

(6) Port 4 default is user data input, address output is optional by writing to Port 4 Data Direction Register

\* The EF6803 operates only in Modes 2 and 3

Figure 13 : Expanded Multiplexed Configuration.



**Note :** To avoid data bus (Port 3) contention in the expanded multiplexed modes, memory devices should be enabled only during E high time.

Figure 14 : Typical Latch Arrangement.

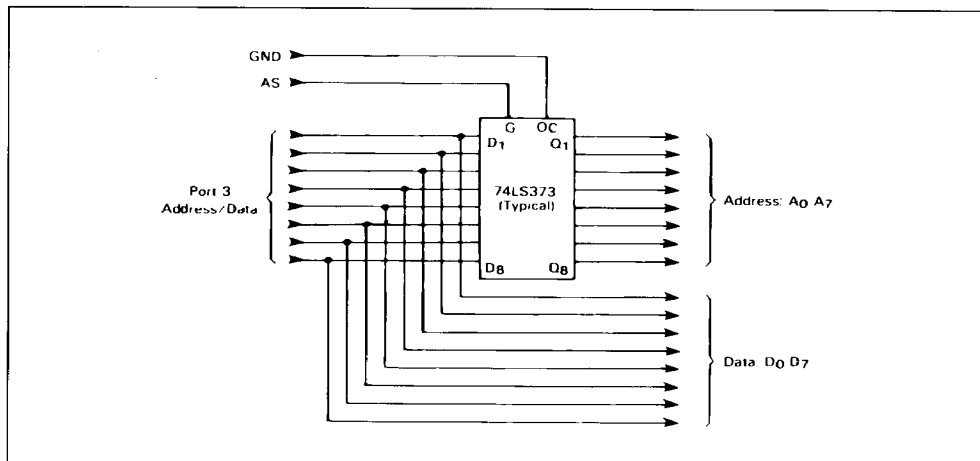
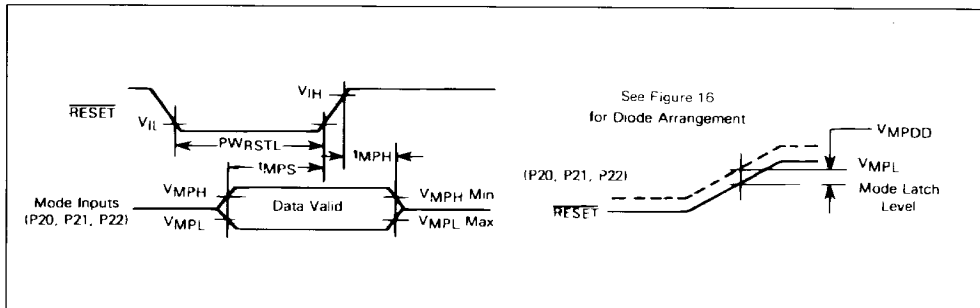


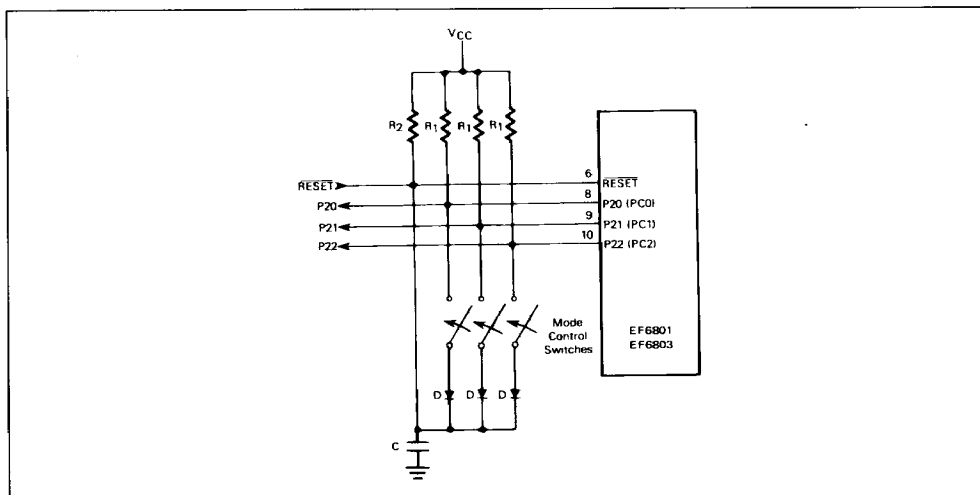
Figure 15 : Mode Programming Timing.

**MODE PROGRAMMING** (refer to figure 15)

| Symbol      | Parameter                                                                                 | Min.     | Max. | Unit     |
|-------------|-------------------------------------------------------------------------------------------|----------|------|----------|
| $V_{MPL}$   | Mode Programming Input Voltage Low*                                                       |          | 1.8  | V        |
| $V_{MPH}$   | Mode Programming Input Voltage High                                                       | 4.0      |      | V        |
| $V_{MPDD}$  | Mode Programming Diode Differential (if diodes are used)                                  | 0.6      |      | V        |
| $PW_{RSTL}$ | RESET Low Pulse Width                                                                     | 3.0      |      | E-Cycles |
| $t_{MPS}$   | Mode Programming Setup Time                                                               | 2.0      |      | E-Cycles |
| $t_{MPH}$   | Mode Programming Hold Time<br>RESET Rise Time $\geq 1\mu s$<br>RESET Rise Time $< 1\mu s$ | 0<br>100 |      | ns       |

\* For  $T_A = -40^\circ C$  to  $+105^\circ C$ ,  $V_{MPL} = 1.7V$ .

Figure 16 : Typical Mode Programming Circuit.



- Notes :
1. Mode 7 as shown
  2.  $R_2 \cdot C$  = Reset time constant
  3.  $R_1 = 10k$  (typical)
  4.  $D = 1N914, 1N4001$  (typical)
  5. Diode  $V_f$  should not exceed  $V_{MPDD \min}$

#### MEMORY MAPS

The 6801 Family can provide up to 64K byte address space depending on the operating mode. A memory map for each operating mode is shown in figure 17.

The first 32 locations of each map are reserved for the internal register area, as shown in table 4, with exceptions as indicated.

Figure 17 : EF6801/03 Memory Maps (sheet 1 of 3).

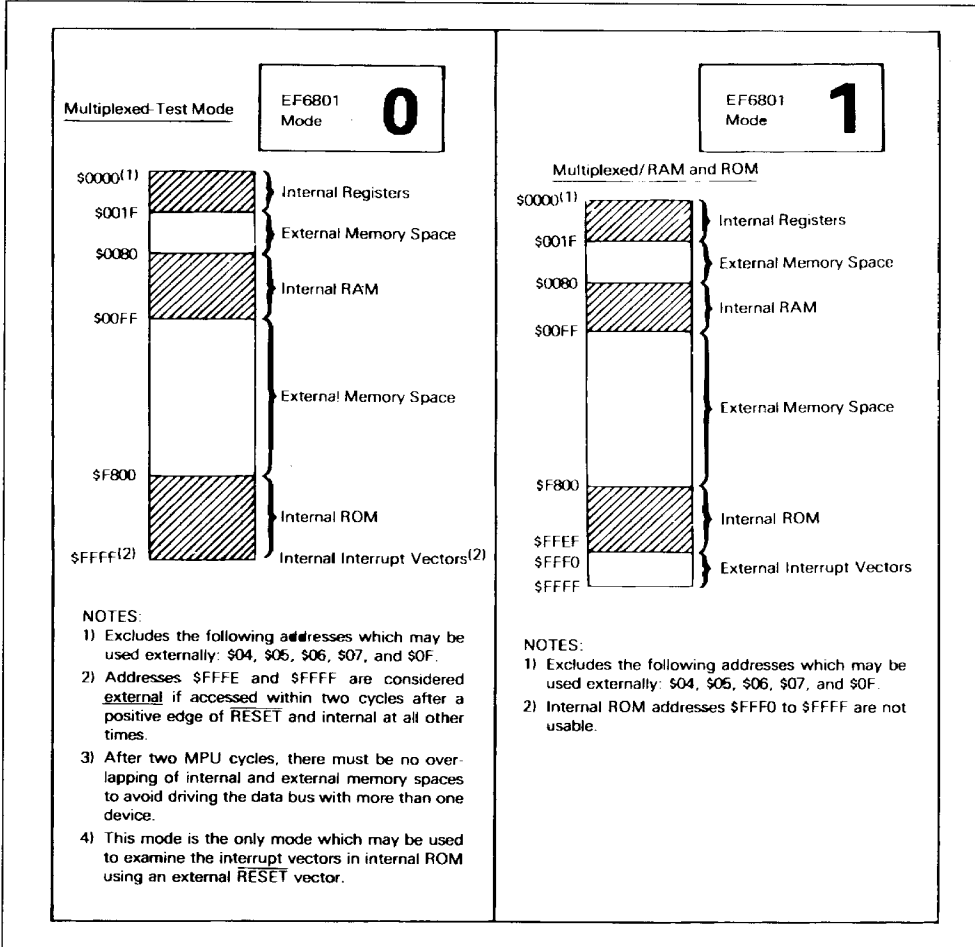


Figure 17 : EF6801/03 Memory Maps (sheet 2 of 3).

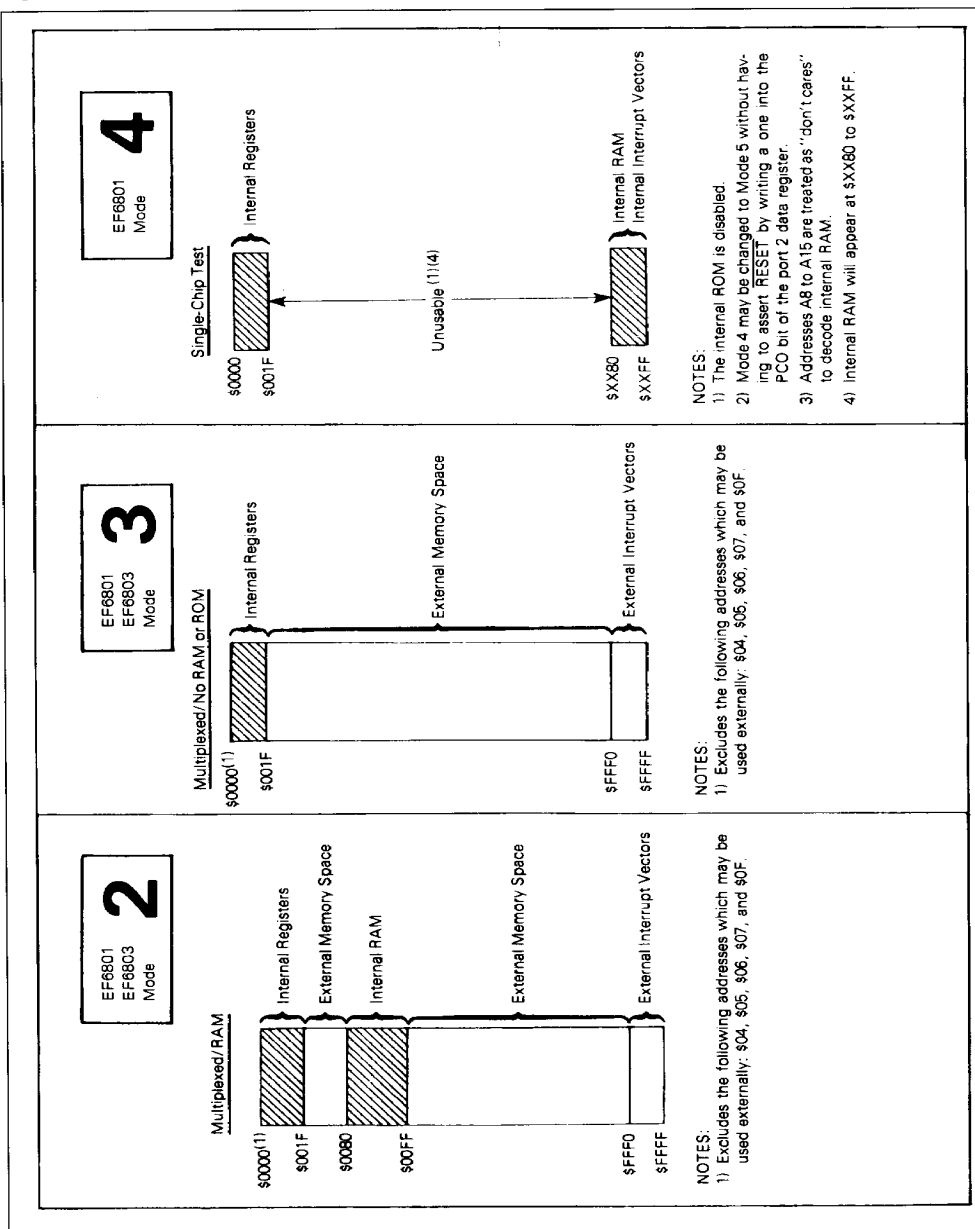
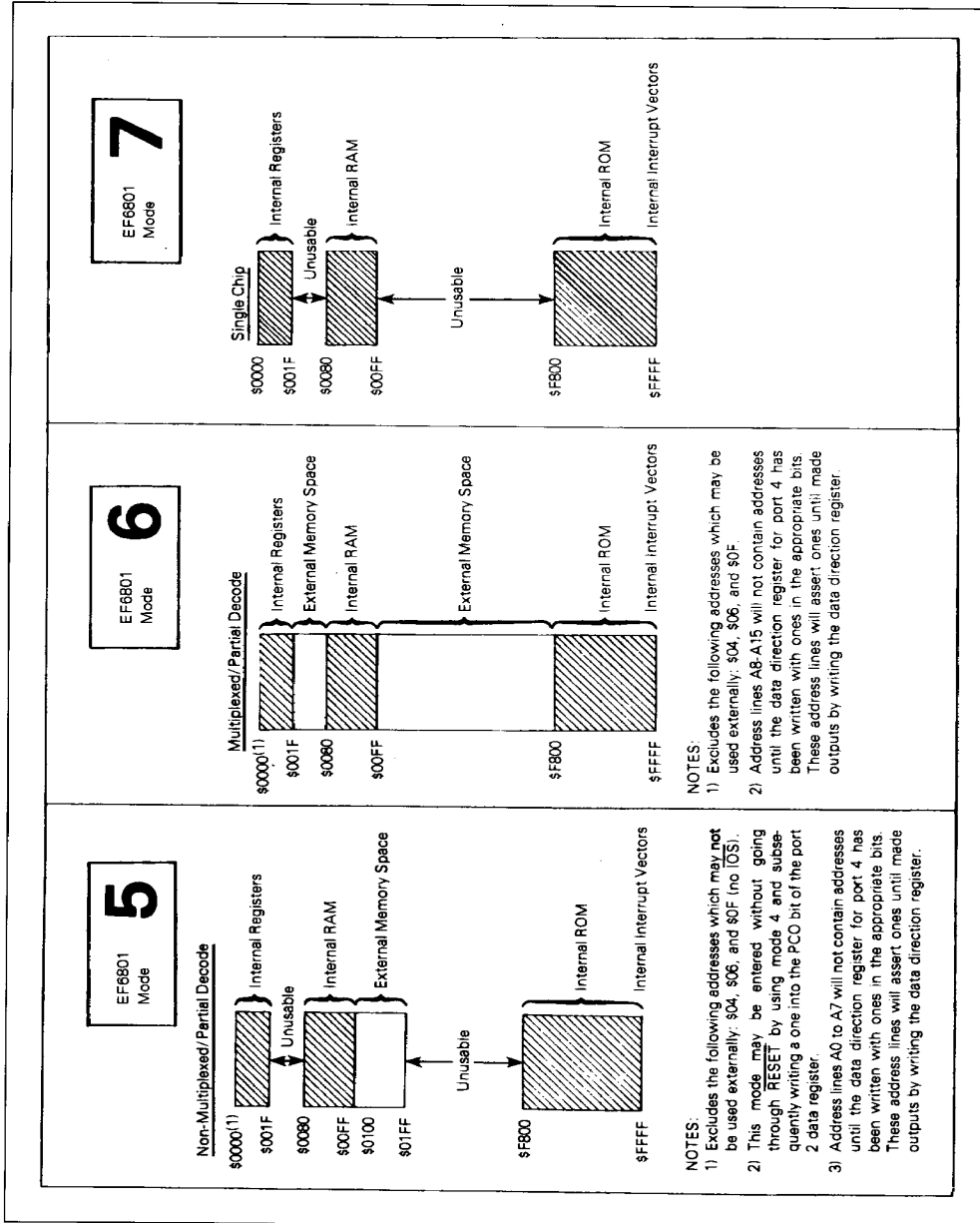


Figure 17 : EF6801/03 Memory Maps (sheet 3 of 3).



## EF6801/03 INTERRUPTS

The 6801 Family supports two types of interrupt requests : maskable and non-maskable. A Non-Maskable Interrupt (NMI) is always recognized and acted upon at the completion of the current instruction. Maskable interrupts are controlled by the Condition Code Register I-bit and by individual enable bits. The I-bit controls all maskable interrupts. Of the maskable interrupts, there are two types : IRQ1 and IRQ2. The Programmable Timer and Serial Communications Interface use an internal IRQ2 interrupt line, as shown in figure 1. External devices (and IS3) use IRQ1. An IRQ1 interrupt is serviced before IRQ2 if both are pending.

All IRQ2 interrupts use hardware prioritized vectors. The single SCI interrupt and three timer interrupts

are serviced in a prioritized order and each is vectored to a separate location. All interrupt vector locations are shown in table 5.

The Interrupt flowchart is depicted in figure 18 and is common to every interrupt excluding reset. During interrupt servicing the Program Counter, Index Register, A Accumulator, B Accumulator, and Condition Code Register are pushed to the stack. The I-bit is set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the Program Counter and instruction execution is resumed. Interrupt and RESET timing are illustrated in figures 19 and 20.

## FUNCTIONAL PIN DESCRIPTIONS

### V<sub>CC</sub> AND V<sub>SS</sub>

V<sub>CC</sub> and V<sub>SS</sub> provide power to a large portion of the MCU. The power supply should provide + 5 volts ( $\pm 5\%$ ) to V<sub>CC</sub>, and V<sub>SS</sub> should be tied to ground. Total power dissipation (including V<sub>CC</sub> Standby), will not exceed P<sub>D</sub> milliwatts.

### V<sub>CC</sub> STANDBY

V<sub>CC</sub> Standby provides power to the standby portion (\$80 through \$BF) of the RAM and the STBY PWR and RAME bits of the RAM Control Register. Voltage requirements depend on whether the device is in a powerup or powerdown state. In the powerup state, the power supply should provide + 5 volts ( $\pm 5\%$ ) and must reach V<sub>SB</sub> volts before RESET reaches 4.0 volts. During powerdown, V<sub>CC</sub> Standby must remain above V<sub>SB</sub> (min) to sustain the standby RAM and STBY PWR bit. While in powerdown operation, the standby current will not exceed I<sub>SB</sub>.

It is typical to power both V<sub>CC</sub> and V<sub>CC</sub> Standby from the same source during normal operation. A diode must be used between them to prevent supplying power to V<sub>CC</sub> during powerdown operation. V<sub>CC</sub> Standby should be tied to ground in Mode 3.

**Table 4 : Internal Register Area.**

| Register                                     | Address |
|----------------------------------------------|---------|
| Port 1 Data Direction Register***            | 00      |
| Port 2 Data Direction Register***            | 01      |
| Port 1 Data Register                         | 02      |
| Port 2 Data Register                         | 03      |
| Port 3 Data Direction Register***            | 04*     |
| Port 4 Data Direction Register***            | 05**    |
| Port 3 Data Register                         | 06*     |
| Port 4 Data Register                         | 07**    |
| Timer Control and Status Register            | 08      |
| Counter (high byte)                          | 09      |
| Counter (low byte)                           | 0A      |
| Output Compare Register (high byte)          | 0B      |
| Output Compare Register (low byte)           | 0C      |
| Input Capture Register (high byte)           | 0D      |
| Input Capture Register (low byte)            | 0E      |
| Port 3 Control and Status Register           | 0F*     |
| Rate and Mode Control Register               | 10      |
| Transmit/receive Control and Status Register | 11      |
| Receive Data Register                        | 12      |
| Transmit Data Register                       | 13      |
| RAM Control Register                         | 14      |
| Reserved                                     | 15-1F   |

\* External addresses in Modes 0, 1, 2, 3, 5, 6 ; cannot be accessed in Mode 5 (No IOS)

\*\* External addresses in Modes 0, 1, 2, 3

\*\*\* 1 = Output, 0 = Input

**Table 5 : MCU Interrupt Vector Locations.**

| MSB  | LSB  | Interrupt                 |
|------|------|---------------------------|
| FFFE | FFFF | RESET                     |
| FFFC | FFFD | NMI                       |
| FFFA | FFFB | Software Interrupt (SWI)  |
| FFF8 | FFF9 | IRQ1 (or IS3)             |
| FFF6 | FFF7 | ICF (input capture)*      |
| FFF4 | FFF5 | OCF (output compare)*     |
| FFF2 | FFF3 | TOF (timer overflow)*     |
| FFF0 | FFF1 | SCI (RDRF + ORFE + TDRE)* |

\* IRQ2 Interrupt

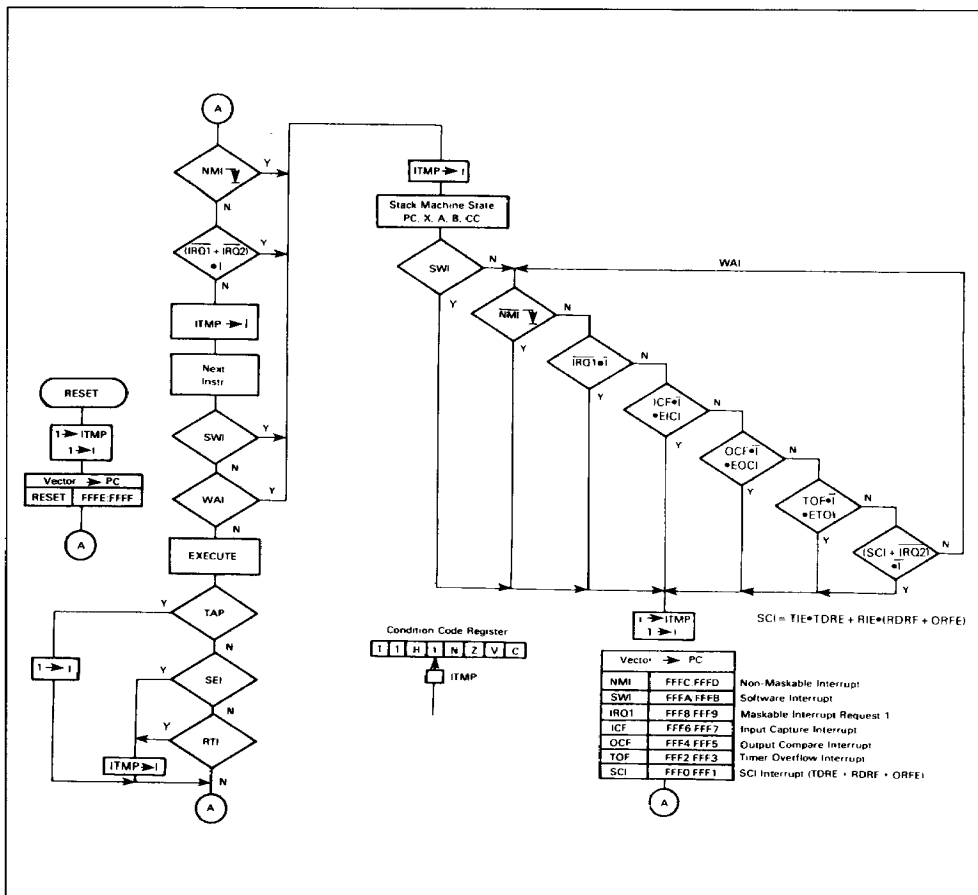
**Figure 18 : Interrupt Flowchart.**

Figure 19 : Interrupt Sequence.

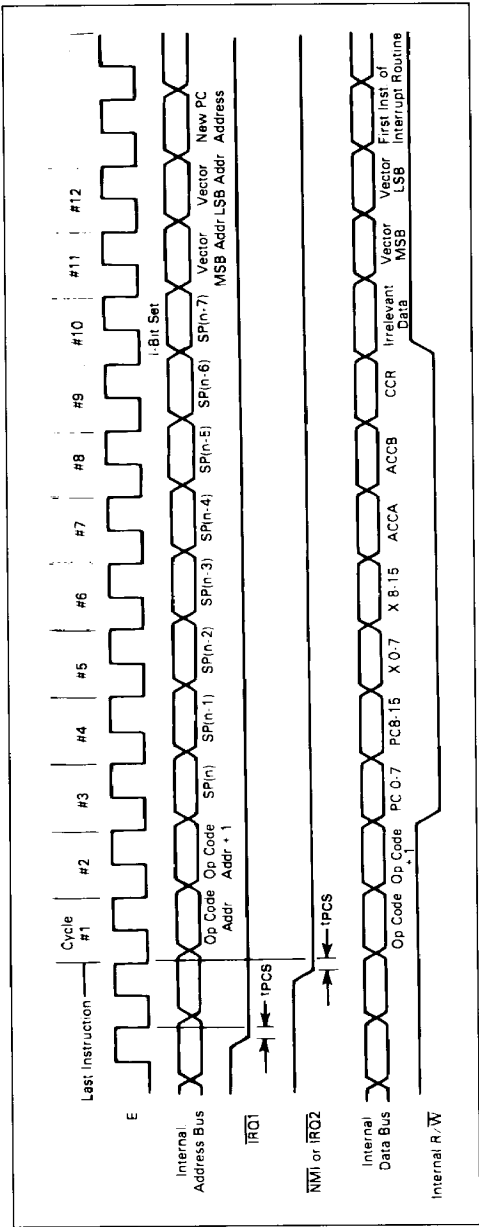
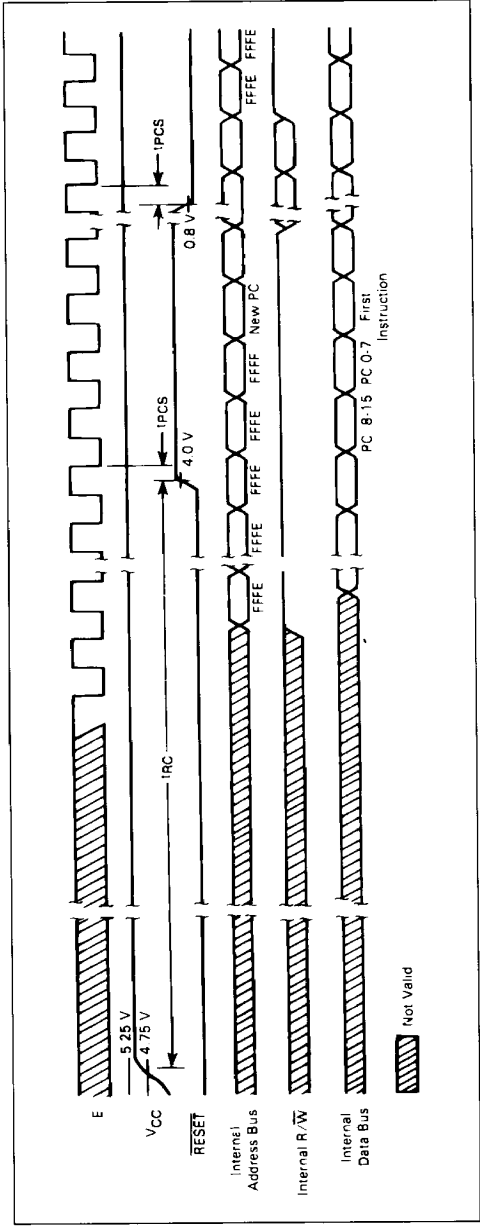


Figure 20 : Reset Timing.



## XTAL1 AND EXTAL2

These two input pins interface either a crystal or TTL compatible clock to the MCU internal clock generator. Divide-by-four circuitry is included which allows use of the inexpensive 3.58MHz or 4.4336MHz Color Burst TV crystals. A 20pF capacitor should be tied from each crystal pin to ground to ensure reliable startup and operation. Alternatively, EXTAL2 may be driven by an external TTL compatible clock at  $4f_0$  with a duty cycle of 50% ( $\pm 5\%$ ) with XTAL1 connected to ground.

The internal oscillator is designed to interface with an AT-cut quartz crystal resonator operated in parallel resonance mode in the frequency range specified for  $f_{XTAL}$ . The crystal should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time.\* The MCU is compatible with most commercially available crystals. Nominal crystal parameters are shown in figure 21.

## RESET

This input is used to reset the internal state of the device and provide an orderly startup procedure. During powerup, RESET must be held below 0.8 volts: (1) at least  $t_{RC}$  after  $V_{CC}$  reaches 4.75 volts in order to provide sufficient time for the clock generator to stabilize, and (2) until  $V_{CC}$  Standby reaches 4.75 volts. RESET must be held low at least three E-cycles if asserted during powerup operation.

## E (ENABLE)

This is an output clock used primarily for bus synchronization. It is TTL compatible and is the slightly skewed divide-by-four result of the device input clock frequency. It will drive one Schottky TTL load and 90pF, and all data given in cycles is referenced to this clock unless otherwise noted.

## NMI (NON-MASKABLE INTERRUPT)

An NMI negative edge requests an MCU interrupt sequence, but the current instruction will be completed before it responds to the request. The MCU will then begin an interrupt sequence. Finally, a vector is fetched from \$FFFC and \$FFFD, transferred to the Program Counter and instruction execution is resumed. NMI typically requires a  $3.3k\Omega$  (nominal) resistor to  $V_{CC}$ . There is no internal NMI pullup resistor. NMI must be held low for at least one E-cycle to be recognized under all conditions.

## IRQ1 (MASKABLE INTERRUPT REQUEST 1)

IRQ1 is a level-sensitive input which can be used to request an interrupt sequence. The MPU will com-

plete the current instruction before it responds to the request. If the interrupt mask bit (I-bit) in the Condition Code Register is clear, the MCU will begin an interrupt sequence. A vector is fetched from \$FFF8 and \$FFF9, transferred to the Program Counter, and instruction execution is resumed.

IRQ1 typically requires an external  $3.3k\Omega$  (nominal) resistor to  $V_{CC}$  for wire-OR applications. IRQ1 has no internal pullup resistor.

## SC1 AND SC2 (STROBE CONTROL 1 AND 2)

The function of SC1 and SC2 depends on the operating mode. SC1 is configured as an output in all modes except single chip mode, whereas SC2 is always an output. SC1 and SC2 can drive one Schottky load and 90pF.

**SC1 AND SC2 IN SINGLE-CHIP MODE.** In Single-Chip Mode, SC1 and SC2 are configured as an input and output, respectively, and both function as Port 3 control lines. SC1 functions as IS3 and can be used to indicate that Port 3 input data is ready or output data has been accepted. Three options associated with IS3 are controlled by Port 3 Control and Status Register and are discussed in the Port 3 description. If unused, IS3 can remain unconnected.

SC2 is configured as OS3 and can be used to strobe output data or acknowledge input data. It is controlled by Output Strobe Select (OSS) in the Port 3 Control and Status Register. The strobe is generated by a read (OSS = 0) or write (OSS = 1) to the Port 3 Data Register. OS3 timing is shown in figure 4.

**SC1 AND SC2 IN EXPANDED NON-MULTIPLEXED MODE.** In the Expanded Non-Multiplexed Mode, both SC1 and SC2 are configured as outputs. SC1 functions as Input/Output Select (IOS) and is asserted only when \$0100 through \$01FF is sensed on the internal address bus.

SC2 is configured as Read/Write and is used to control the direction of data bus transfers. An MPU read is enabled when Read/Write and E are high.

**SC1 AND SC2 IN EXPANDED MULTIPLEXED MODE.** In the Expanded Multiplexed Modes, both SC1 and SC2 are configured as outputs. SC1 functions as Address Strobe and can be used to demultiplex the eight least significant addresses and the data bus. A latch controlled by Address Strobe captures address on the negative edge, as shown in figure 14.

SC2 is configured as Read/Write and is used to control the direction of data bus transfers. An MPU read is enabled when Read/Write and E are high.

P10-P17 (PORT 1)

Port 1 is a mode independent 8-bit I/O port with each line an input or output as defined by the Port 1 Data Direction Register. The TTL compatible three-state output buffers can drive one Schottky TTL load and 30pF, Darlington transistors, or CMOS devices using external pullup resistors. It is configured as a data input port by RESET. Unused lines can remain unconnected.

P20-P24 (PORT 2)

Port 2 is a mode-independent, 5-bit, multipurpose I/O port. The voltage levels present on P20, P21, and P22 on the rising edge of RESET determine the operating mode of the MCU. The entire port is then configured as a data input port. The Port 2 lines can be selectively configured as data output lines by setting the appropriate bits in the Port 2 Data Direction

Register. The Port 2 Data Register is used to move data through the port. However, if P21 is configured as an output, it will be tied to the timer Output Compare function and cannot be used to provide output from the Port 2 Data Register.

Port 2 can also be used to provide an interface for the Serial Communications Interface and the timer Input Edge function. These configurations are described in the Programmable Timer and Serial Communications Interface (SCI) section.

The Port 2 three-state, TTL-compatible output buffers are capable of driving one Schottky TTL load and 30pF, or CMOS devices using external pullup resistors.

PORT2 DATA REGISTER

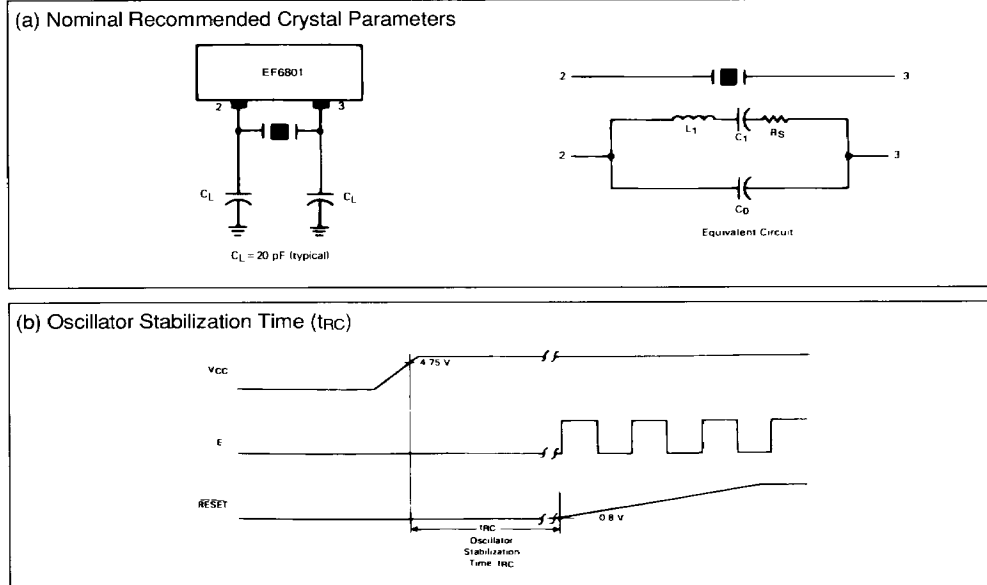
| 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |        |
|-----|-----|-----|-----|-----|-----|-----|-----|--------|
| PC2 | PC1 | PC0 | P24 | P23 | P22 | P21 | P20 | \$0003 |

Figure 21 : 6801 Family Oscillator Characteristics.

Nominal Crystal Parameters\*

|    | 3.58MHz | 4.00MHz | 5.0MHz      | 6.0MHz      | 8.0MHz      |
|----|---------|---------|-------------|-------------|-------------|
| RS | 60Ω     | 50Ω     | 30-50Ω      | 30-50Ω      | 20-40Ω      |
| C0 | 3.5pF   | 6.5pF   | 4-6pF       | 4-6pF       | 4-6pF       |
| C1 | 0.015pF | 0.025pF | 0.01-0.02pF | 0.01-0.02pF | 0.01-0.02pF |
| Q  | > 40K   | > 30K   | > 20K       | > 20K       | > 20K       |

Note : These are representative AT-cut crystal parameters only. Crystals of other types of cut may also be used.



### P30-P37 (PORT 3)

Port 3 can be configured as an I/O port, a bidirectional 8-bit data bus, or a multiplexed address/data bus depending on the operating mode. The TTL compatible three-state output buffers can drive one Schottky TTL load and 90pF. Unused lines can remain unconnected.

**PORT 3 IN SINGLE-CHIP MODE.** Port 3 is an 8-bit I/O port in the Single-Chip Mode, with each line configured by the Port 3 Data Direction Register. There are also two lines, IS3 and OS3, which can be used to control Port 3 data transfers.

Three Port 3 options are controlled by the Port 3 Control and Status Register and are available only in Single-Chip Mode : (1) Port 3 input data can be latched using IS3 as a control signal, (2) OS3 can be generated by either an MPU read or write to the Port 3 Data Register, and (3) an IRQ1 interrupt can be enabled by an IS3 negative edge. Port 3 latch timing is shown in figure 5.

PORT 3 CONTROL AND STATUS REGISTER

| 7   | 6                     | 5 | 4   | 3      | 2 | 1 | 0 |        |
|-----|-----------------------|---|-----|--------|---|---|---|--------|
| IS3 | IS3<br>IRQ1<br>Enable | X | OSS | Enable | X | X | X | \$000F |

Bit 0-2 Not used.

**Bit 3** LATCH ENABLE. This bit controls the input latch for Port 3. If set, input data is latched by an IS3 negative edge. The latch is transparent after a read of the Port 3 Data Register. LATCH ENABLE is cleared during reset.

**Bit 4** OSS (Output Strobe Select). This bit determines whether OS3 will be generated by a read or write of the Port 3 Data Register. When clear, the strobe is generated by a read ; when set, it is generated by a write. OSS is cleared during reset.

**Bit 5** Not used.

**Bit 6** IS3 IRQ1 ENABLE. When set, an IRQ1 interrupt will be enabled whenever IS3 FLAG is set ; when clear, the interrupt is inhibited. This bit is cleared during reset.

Bit 7

IS3 FLAG. This read-only status bit is set by an IS3 negative edge. It is cleared by a read of the Port 3 Control and Status Register (with IS3 FLAG set) followed by a read or write to the Port 3 Data Register or during reset.

**PORT 3 IN EXPANDED NON-MULTIPLEXED MODE.** Port 3 is configured as a bidirectional data bus (D7-D0) in the Expanded Non-Multiplexed Mode. The direction of data transfers is controlled by Read/Write (SC2). Data is clocked by E (Enable).

**PORT 3 IN EXPANDED MULTIPLEXED MODE.** Port 3 is configured as a time multiplexed address (A0-A7) and data bus (D7-D0) in the Expanded Multiplexed Modes, where Address Strobe (AS) can be used to demultiplex the two buses. Port 3 is held in a high impedance state between valid address and data to prevent bus conflicts.

### P40-P47 (PORT 4)

Port 4 is configured as an 8-bit I/O port, as address outputs, or as data inputs depending on the operating mode. Port 4 can drive one Schottky TTL load and 90pF and is the only port with internal pullup resistors. Unused lines can remain unconnected.

**PORT 4 IN SINGLE-CHIP MODE.** In Single-Chip Mode, Port 4 functions as an 8-bit I/O port with each line configured by the Port 4 Data Direction Register. Internal pullup resistors allow the port to directly interface with CMOS at 5 volt levels. External pullup resistors to more than 5 volts, however, cannot be used.

**PORT 4 IN EXPANDED NON-MULTIPLEXED MODE.** Port 4 is configured from reset as an 8-bit input port, where the Port 4 Data Direction Register can be written to provide any or all of eight address lines, A0 to A7. Internal pullup resistors pull the lines high until the Port 4 Data Direction Register is configured.

**PORT 4 IN EXPANDED MULTIPLEXED MODE.** In all Expanded Multiplexed modes except Mode 6, Port 4 functions as half of the address bus and provides A8 to A15. In Mode 6, the port is configured from reset as an 8-bit parallel input port, where the Port 4 Data Direction Register can be written to provide any or all of upper address lines A8 to A15. Internal pullup resistors pull the lines high until the Port 4 Data Direction Register is configured, where bit 0 controls A8.

## RESIDENT MEMORY

The EF6801 provides 2048 bytes of on-board ROM and 128 bytes of on-board RAM.

One half of the RAM is powered through the V<sub>CC</sub> standby pin and is maintainable during V<sub>CC</sub> power-down. This standby portion of the RAM consists of 64 bytes located from \$80 through \$BF.

Power must be supplied to V<sub>CC</sub> standby if the internal RAM is to be used regardless of whether standby power operation is anticipated.

The RAM is controlled by the RAM Control Register.

## RAM CONTROL REGISTER (\$14)

The RAM Control Register includes two bits which can be used to control RAM accesses and determine the adequacy of the standby power source during powerdown operation. It is intended that RAME be cleared and STBY PWR be set as part of a powerdown procedure.

| RAM CONTROL REGISTER |      |   |   |   |   |   |   |
|----------------------|------|---|---|---|---|---|---|
| 7                    | 6    | 5 | 4 | 3 | 2 | 1 | 0 |
| STY<br>PWR           | RAME | X | X | X | X | X | X |

|         |          |
|---------|----------|
| Bit 0-5 | Not used |
|---------|----------|

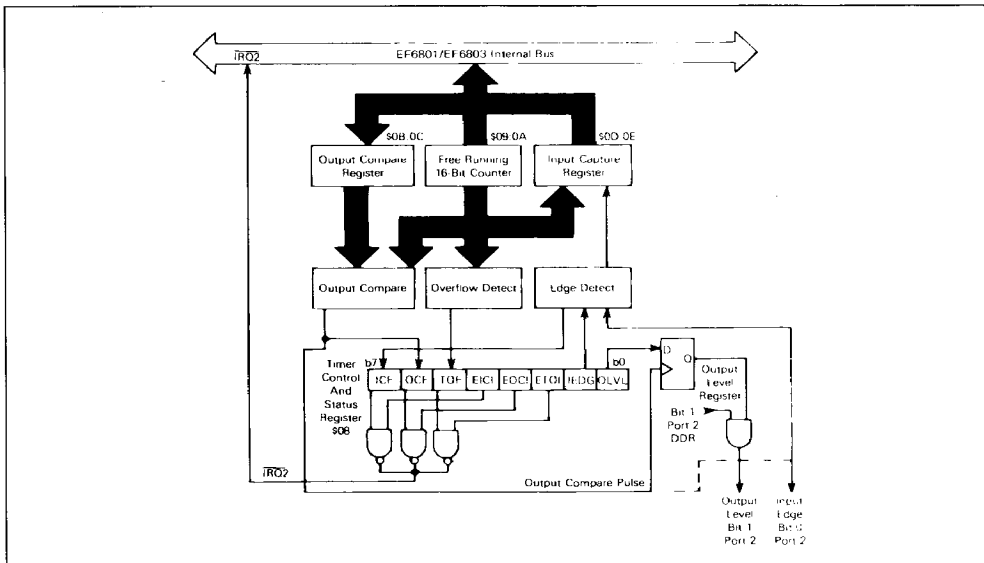
|            |                                                               |
|------------|---------------------------------------------------------------|
| Bit 6 RAME | RAM Enable. This read/write bit can be used to remove the en- |
|------------|---------------------------------------------------------------|

**Figure 22 : Block Diagram of Programmable Timer.**

ture RAM from the internal memory map. RAME is set (enabled) during reset provided standby power is available on the positive edge of RESET. If RAME is clear, any access to a RAM address is external. If RAME is set and not in mode 3, the RAM is included in the internal map.

**Bit 7 STBY PWR** Standby Power. This bit is a read/write status bit which, when once set, remains set as long as  $V_{CC}$  standby remains above  $V_{SBB}$  (minimum). As long as this bit is set following a period of standby operation, the standby power supply has adequately preserved the data in the standby RAM. If this bit is cleared during a period of standby operation, it indicates that  $V_{CC}$  standby had fallen to a level sufficiently below  $V_{SBB}$  (minimum) to suspect that data in the standby RAM is not valid. This bit can be set only by software and is not affected during reset.

**Figure 22 : Block Diagram of Programmable Timer.**



## PROGRAMMABLE TIMER

The programmable timer can be used to perform input waveform measurements while independently generating an output waveform. Pulse widths can vary from several microseconds to many seconds. A block diagram of the timer is shown in figure 22.

### COUNTER (\$09:0A)

The key timer element is a 16-bit free-running counter which is incremented by E (enable). It is cleared during reset and is read-only with one exception : a write to the counter (\$09) will preset it to \$FFF8. This feature, intended for testing, can disturb serial operations because the counter provides the SCI internal bit rate clock. TOF is set whenever the counter contains all ones.

### OUTPUT COMPARE REGISTER (\$0B:0C)

The output compare register is a 16-bit read/write register used to control an output waveform or provide an arbitrary timeout flag. It is compared with the free-running counter on each E cycle. When a match occurs, OCF is set and OLVL is clocked to an output level register. If port 2, bit 1, is configured as an output, OLVL will appear at P21 and the output compare register and OLVL can then be changed for the next and OLVL is clocked to an output level register. If Port 2, bit 1, is configured as an output, OLVL will appear at P21 and the Output Compare Register and OLVL can then be changed for the next compare. The function is inhibited for one cycle after a write to its high byte (\$0B) to ensure a valid compare. The Output Compare Register is set to \$FFFF at RESET.

### INPUT CAPTURE REGISTER (\$0D:0E)

The Input Capture Register is a 16-bit read-only register used to store the free-running counter when a "proper" input transition occurs as defined by IEDG. Port 2, bit 0 should be configured as an input, but the edge detect circuit always senses P20 even when configured as an output. An input capture can occur independently of ICF : the register always contains the most current value. Counter transfer is inhibited, however, between accesses of a double byte MPU read. The input pulse width must be at least two E-cycles to ensure an input capture under all conditions.

### TIMER CONTROL AND STATUS REGISTER (\$08)

The Timer Control and Status Register (TCSR) is an 8-bit register of which all bits are readable, while only bits 0-4 can be written. The three most significant bits provide the timer status and indicate if :

- a proper level transition has been detected,
- a match has occurred between the free-running counter and the output compare register, and
- the free-running counter has overflowed.

Each of the three events can generate an IRQ2 interrupt and is controlled by an individual enable bit in the TCSR.

#### TIMER CONTROL AND STATUS REGISTER (TCSR)

| 7   | 6   | 5   | 4    | 3    | 2    | 1    | 0    |       |
|-----|-----|-----|------|------|------|------|------|-------|
| ICF | OCF | TOF | EICI | EOCI | ETOI | IEDG | OLVL | \$008 |

- Bit 0 OLVL Output level. OLVL is clocked to the output level register by a successful output compare and will appear at P21 if Bit 1 of the Port 2 Data Direction Register is set. It is cleared during reset.
- Bit 1 EIDG Input Edge. IEDG is cleared during reset and controls which level transition will trigger a counter transfer to the Input Capture Register :  
 IEDG = 0 Transfer on a negative-edge  
 IEDG = 1 Transfer on a positive-edge
- Bit 2 ETOI Enable Timer Overflow Interrupt. When set, an IRQ2 interrupt is enabled for a timer overflow ; when clear, the interrupt is inhibited. It is cleared during reset.
- Bit 3 EOCl Enable Output Compare Interrupt. When set, an IRQ2 interrupt is enabled for an output compare ; when clear, the interrupt is inhibited. It is cleared during reset.
- Bit 4 EICI Enable Input Capture Interrupt. When set, an IRQ2 interrupt is enabled for an input capture ; when clear, the interrupt is inhibited. It is cleared during reset.
- Bit 5 TOF Timer Overflow Flag. TOF is set when the counter contains all 1's. It is cleared by reading the TCSR (with TOF set) then reading the counter high byte (\$09), or during reset.

## Bit 6 OCF

Output Compare Flag. OCF is set when the Output Compare Register matches the free-running counter. It is cleared by reading the TCSR (with OCF set) and then writing to the Output Compare Register (\$0B or \$0C), or during reset.

## Bit 7 ICF

Input Capture Flag. ICF is set to indicate a proper level transition ; it is cleared by reading the TCSR (with ICF set) and then the Input Capture Register High Byte (\$0D), or during reset.

## SERIAL COMMUNICATIONS INTERFACE (SCI)

A full-duplex asynchronous Serial Communications Interface (SCI) is provided with two data formats and a variety of rates. The SCI transmitter and receiver are functionally independent, but use the same data format and bit rate. Serial data formats include standard mark/space (NRZ) and Biphasic and both provide one start bit, eight data bits, and one stop bit. "Baud" and "bit rate" are used synonymously in the following description.

## WAKE-UP FEATURE

In a typical serial loop multi-processor configuration, the software protocol will usually identify the address(es) at the beginning of the message. In order to permit uninterested MPU's to ignore the remainder of the message, a wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line goes idle. An SCI receiver is re-enabled by an idle string of ten consecutive 1's or during reset. Software must provide for the required idle string between consecutive messages and prevent it within messages.

## PROGRAMMABLE OPTIONS

The following features of the SCI are programmable :

- format : standard mark/space (NRZ) or Bi-phase
- clock : external or internal bit rate clock
- Baud : one of 4 per E-clock frequency, or external clock (x 8 desired baud)
- wake-up feature : enabled or disabled
- interrupt requests : enabled individually for transmitter and receiver
- clock output : internal bit rate clock enabled or disabled to P22

## SERIAL COMMUNICATIONS REGISTERS

The Serial Communications Interface includes four addressable registers as depicted in figure 23. It is controlled by the Rate and Mode Control Register and the Transmit/Receive Control and Status Register. Data is transmitted and received utilizing a write-only Transmit Register and a read-only Receive Register. The shift registers are not accessible to software.

RATE AND MODE CONTROL REGISTER (RMCR) (\$10). The Rate and Mode Control Register controls the SCI bit rate, format, clock source, and under certain conditions, the configuration of P22. The register consists of four write-only bits which are cleared during reset. The two least significant bits control the bit rate of the internal clock and the remaining two bits control the format and clock source.

RATE AND MODE CONTROL REGISTER (RMCR)

| 7 | 6 | 5 | 4 | 3   | 2   | 1   | 0   |        |
|---|---|---|---|-----|-----|-----|-----|--------|
| X | X | X | X | CC1 | CC0 | SS1 | SS0 | \$0010 |

Bit 1 : Bit 0

SS1 : SS0 Speed Select.

These two-bits select the Baud rate when using the internal clock. Four rates may be selected which are a function of the MCU input frequency. Table 6 lists bit time and rates for three selected MCU frequencies.

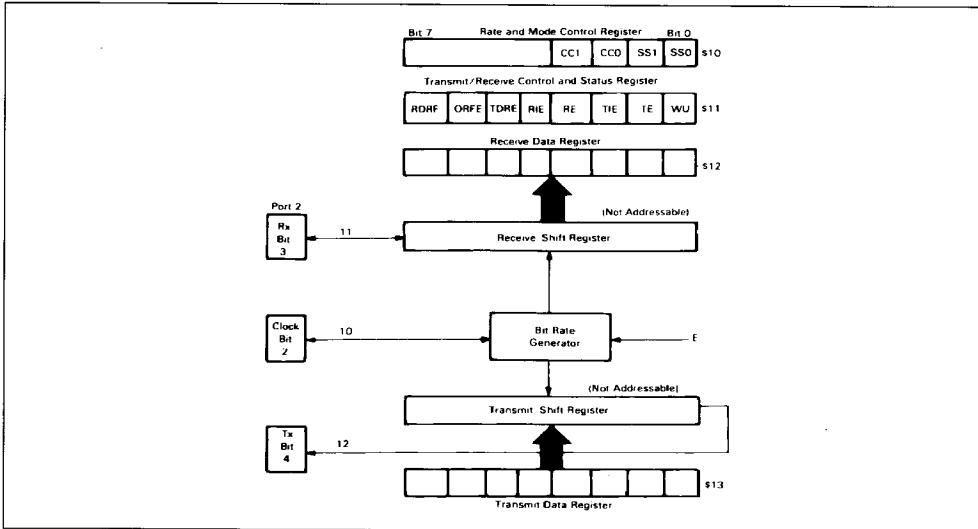
Bit 3 : Bit 2

CC1 : CC0 Clock Control and Format Select. These two bits control the format and select the serial clock source. If CC1 is set, the DDR value for P22 is forced to the complement of CC0 and cannot be altered until CC1 is cleared. If CC1 is cleared after having been set, its DDR value is unchanged. Table 7 defines the formats, clock source, and use of P22.

If both CC1 and CC0 are set, an external TTL compatible clock must be connected to P22 at eight times (8X) the desired bit rate, but not greater than E, with a duty cycle of 50% ( $\pm 10\%$ ). If CC1 : CC0 = 10, the internal bit rate clock is provided at P22 regardless of the values for TE or RE.

NOTE : The source of SCI internal bit rate clock is the timer free running counter. An MPU write to the counter can disturb serial operations.

Figure 23 : SCI Registers.



**TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR) (\$11).** The Transmit/Receive Control and Status Register controls the transmitter, receiver, wake-up feature, and two individual interrupts and monitors the status of serial operations. All eight bits are readable while bits 0 to 4 are also writable. The register is initialized to \$20 by RESET.

#### TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR)

| 7    | 6    | 5    | 4   | 3  | 2   | 1  | 0  |       |
|------|------|------|-----|----|-----|----|----|-------|
| RDRF | ORFE | TDRE | RIE | RE | TIE | TE | WU | \$001 |

- Bit 0 WU** "Wake-up" on Idle Line. When set, WU enables the wake-up function ; it is cleared by ten consecutive 1's or during reset. WU will not set if the line is idle.
- Bit 1 TE** Transmit Enable. When set, P24 DDR bit is set, cannot be changed, and will remain set if TE is subsequently cleared. When TE is changed from clear to set, the transmitter is connected to P24 and a preamble of nine consecutive 1's is transmitted. TE is cleared during reset.
- Bit 2 TIE** Transmit Interrupt Enable. When set, an IRQ2 interrupt is

**Bit 3 RE**

**Bit 4 RIE**

**Bit 5 TDRE**

enabled when TDRE is set ; when clear, the interrupt is inhibited. TE is cleared during reset.

**Receive Enable.** When set, the P23 DDR bit is cleared, cannot be changed, and will remain clear if RE is subsequently cleared. While RE is set, the SCI receiver is enabled. RE is cleared during reset.

**Receiver Interrupt Enable.** When set, an IRQ2 interrupt is enabled when RDRF and/or ORFE is set ; when clear, the interrupt is inhibited. RIE is cleared during reset.

**Transmit Data Register Empty.** TDRE is set when the Transmit Data Register is transferred to the output serial shift register or during reset. It is cleared by reading the TRCSR (with TDRE set) and then writing to the Transmit Data Register. Additional data will be transmitted only if TDRE has been cleared.

## Bit 6 ORFE

Overrun Framing Error. If set, ORFE indicates either an overrun or framing error. An overrun is a new byte ready to transfer to the Receive Data Register with RDRF still set. A receiver framing error has occurred when the byte boundaries of the bit stream are not synchronized to the bit counter. An overrun can be distinguished from a framing error by the state of RDRF : if RDRF is set, then an overrun has occurred ; otherwise a framing error has been detected. Data is not transferred to the Receive Data Register in an overrun condition. Unframed data causing a

framing error is transferred to the Receive Data Register. However, subsequent data transfer is blocked until the framing error flag is cleared.\* ORFE is cleared by reading the TRCSR (with ORFE set) then the Receive Data Register, or during reset.

## Bit 7 RDRF

Receive Data Register Full. RDRF is set when the input serial shift register is transferred to the Receive Data Register. It is cleared by reading the TRCSR (with RDRF set), and then the Receive Data Register, or during reset.

Table 6 : SCI Bit Times and Rates.

| SS1 : SS0       |   | 4 f <sub>o</sub> → | 2.4576MHz          | 4.0MHz             | 4.9152MHz          |
|-----------------|---|--------------------|--------------------|--------------------|--------------------|
|                 |   | E                  | 614.4kHz           | 1.0MHz             | 1.2288MHz          |
| 0               | 0 | + 16               | 26μs/38.400 Baud   | 16μs/62.500 Baud   | 13.0μs/76.800 Baud |
| 0               | 1 | + 128              | 208μs/4.800 Baud   | 128μs/7812.5 Baud  | 104.2μs/9.600 Baud |
| 1               | 0 | + 1024             | 1.67ms/600 Baud\$  | 1.024ms/976.6 Baud | 833.3μs/1.200 Baud |
| 1               | 1 | + 4096             | 6.67ms/150 Baud    | 4.096ms/244.1 Baud | 3.33ms/300 Baud    |
| *External (P22) |   |                    | 13.0μs/76.800 Baud | 8.0μs/125.000 Baud | 6.5μs/153.600 Baud |

\* Using maximum clock rate

Table 7 : SCI Format and Clock Source Control.

| CC1:CC0 | Format   | Clock Source | Port 2 Bit 2 |
|---------|----------|--------------|--------------|
| 00      | Bi-phase | Internal     | Not used     |
| 01      | NRZ      | Internal     | Not used     |
| 10      | NRZ      | Internal     | Output       |
| 11      | NRZ      | External     | Input        |

## SERIAL OPERATIONS

The SCI is initialized by writing control bytes first to the Rate and Mode Control Register and then to the Transmit/Receive Control and Status Register. When TE is set, the output of the transmit serial shift register is connected to P24 and serial output is initiated by transmitting a 9-bit preamble of 1's.

At this point one of two situations exist : 1) if the Transmit Data Register is empty (TDRE = 1), a continuous string of 1's will be sent indicating an idle line, or 2) if a byte has been written to the Transmit-Data Register (TDRE = 0), it will be transferred to

the output serial shift register (synchronized with the bit rate clock), TDRE will be set, and transmission will begin.

The start bit (0), eight data bits (beginning with bit 0) and a stop bit (1), will be transmitted. If TDRE is still set when the next byte transfer should occur, 1's will be sent until more data is provided. In Bi-phase format, the output toggles at the start of each bit and at half-bit time when a "1" is sent. Receive operation is controlled by RE which configures P23 as an input and enables the receiver. SCI data formats are illustrated in figure 24.

## INSTRUCTION SET

The EF6801/03 is upward source and object code compatible with the EF6800. Execution times of key instructions have been reduced and several new instructions have been added, including a hardware multiply. A list of new operations added to the EF6800 instruction set is shown in table 1.

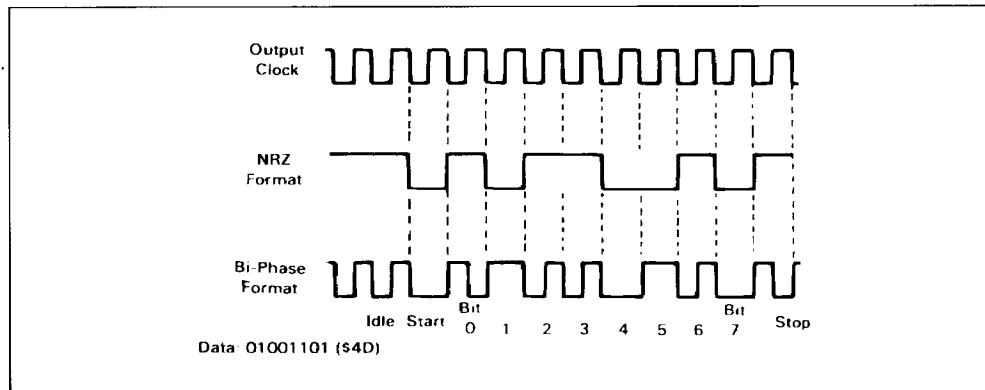
In addition, two new special opcodes, 4E and 5E, are provided for test purposes. These opcodes force the Program Counter to increment like a 16-bit counter, causing address lines used in the expanded modes to increment until the device is reset. These opcodes have no mnemonics.

The coding of the first (or only) byte corresponding to an executable instruction is sufficient to identify the instruction and the addressing mode. The hexadecimal equivalents of the binary codes, which result from the translation of the 82 instructions in all valid modes of addressing, are shown in table 8. There are 220 valid machine codes, 34 unassigned codes, and 2 codes reserved for test purposes.

## PROGRAMMING MODEL

A programming model for the EF6801/03 is shown in figure 10. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most significant byte. Any operation which modifies the double accumulator will also modify accumulator A and/or B. Other registers are defined as follows :

**Figure 24 : SCI Data Formats.**



**PROGRAM COUNTER.** The program counter is a 16-bit register which always points to the next instruction.

**STACK POINTER.** The stack pointer is a 16-bit register which contains the address of the next available location in a pushdown/pullup (LIFO) queue. The stack resides in random access memory at a location defined by the programmer.

**INDEX REGISTER.** The Index Register is a 16-bit register which can be used to store data or provide an address for the indexed mode of addressing.

**ACCUMULATORS.** The MPU contains two 8-bit accumulators, A and B, which are used to store operands and results from the arithmetic logic unit (ALU). They can also be concatenated and referred to as the D (double) accumulator.

**CONDITION CODE REGISTERS.** The condition code register indicates the results of an instruction and includes the following five condition bits : Negative (N), Zero (Z), Overflow (V), Carry/Borrow from MSB (C), and Half Carry from bit 3 (H). These bits are testable by the conditional branch instructions. Bit 4 is the interrupt mask (I-bit) and inhibits all maskable interrupts when set. The two unused bits, B6 and B7, are read as ones.

## ADDRESSING MODES

Six addressing modes can be used to reference memory. A summary of addressing modes for all instructions is presented in table 9, 10, 11, and 12, where execution times are provided in E-cycles. Instruction execution times are summarized in table 13. With an input frequency of 4MHz, E-cycles are equivalent to microseconds. A cycle-by-cycle description of bus activity for each instruction is provided in table 14 and a description of selected instructions is shown in figure 25.

**IMMEDIATE ADDRESSING.** The operand or "immediate byte(s)" is contained in the following byte(s) of the instruction where the number of bytes matches the size of the register. These are two or three byte instructions.

**DIRECT ADDRESSING.** The least significant byte of the operand address is contained in the second byte of the instruction and the most significant byte is assumed to be \$00. Direct addressing allows the user to access \$00 through \$FF using two byte instructions and execution time is reduced by elimi-

nating the additional memory access. In most applications, the 256-byte area is reserved for frequently referenced data.

**EXTENDED ADDRESSING.** The second and third bytes of the instruction contain the absolute address of the operand. These are three byte instructions.

**INDEXED ADDRESSING.** The unsigned offset contained in the second byte of the instruction is added with carry to the Index Register and used to reference memory without changing the Index Register. These are two byte instructions.

**INHERENT ADDRESSING.** The operand(s) are registers and no memory reference is required. These are single byte instructions.

**RELATIVE ADDRESSING.** Relative addressing is used only for branch instructions. If the branch condition is true, the Program Counter is overwritten with the sum of a signed single byte displacement in the second byte of the instruction and the current Program Counter. This provides a branch range of - 126 to 129 bytes from the first byte of the instruction. These are two byte instructions.

Table 8 : CPU Instruction Map.

| OP | MNEM | MODE  | ~ | # | OP | MNEM | MODE  | ~ | #  | OP   | MNEM  | MODE  | ~ | #  | OP   | MNEM  | MODE  | ~ | #  | OP   | MNEM  | MODE  | ~ | # |
|----|------|-------|---|---|----|------|-------|---|----|------|-------|-------|---|----|------|-------|-------|---|----|------|-------|-------|---|---|
| 00 | *    |       |   |   | 34 | DES  | INHER | 3 | 1  | 68   | ASL   | INDXD | 6 | 2  | 9C   | CPX   | DIR   | 5 | 2  | D0   | SUBB  | DIR   | 3 | 2 |
| 01 | NOP  | INHER | 2 | 1 | 35 | TXS  |       | 3 | 1  | 69   | ROL   |       | 6 | 2  | 9D   | JSR   | ▲     | 5 | 2  | D1   | CMPB  | ▲     | 3 | 2 |
| 02 | *    |       |   |   | 36 | PSHA |       | 3 | 1  | 6A   | DEC   |       | 6 | 2  | 9E   | LDS   | ▼     | 4 | 2  | D2   | SBCB  |       | 3 | 2 |
| 03 | *    |       |   |   | 37 | PSHB |       | 3 | 1  | 6B   | *     |       |   |    | 9F   | STS   | DIR   | 4 | 2  | D3   | ADD   |       | 5 | 2 |
| 04 | LSRD |       | 3 | 1 | 38 | PULX |       | 5 | 1  | 6C   | INC   |       | 6 | 2  | A0   | SUBA  | INDXD | 4 | 2  | D4   | ANDB  |       | 3 | 2 |
| 05 | ASLD |       | 3 | 1 | 39 | RTS  |       | 5 | 1  | 6D   | TST   |       | 6 | 2  | A1   | CMPA  | ▲     | 4 | 2  | D5   | BITB  |       | 3 | 2 |
| 06 | TAP  |       | 2 | 1 | 3A | ABX  |       | 3 | 1  | 6E   | JMP   |       | 3 | 2  | A2   | SBCA  |       | 4 | 2  | D6   | LDAB  |       | 3 | 2 |
| 07 | TPA  |       | 2 | 1 | 3B | RTI  | 10    | 1 | 6F | CLR  | INDXD | 6     | 2 | A3 | SUBD |       | 6     | 2 | D7 | STAB |       | 3     | 2 |   |
| 08 | INX  |       | 3 | 1 | 3C | PSHX | 4     | 1 | 70 | NEG  | EXTND | 6     | 3 | A4 | ANDA |       | 4     | 2 | D8 | EORB |       | 3     | 2 |   |
| 09 | DEX  |       | 3 | 1 | 3D | MUL  | 10    | 1 | 71 | *    |       |       |   | A5 | BITA |       | 4     | 2 | D9 | ADCB |       | 3     | 2 |   |
| 0A | CLV  |       | 2 | 1 | 3E | WAI  | 9     | 1 | 72 | *    |       |       |   | A6 | LDAA |       | 4     | 2 | DA | ORAB |       | 3     | 2 |   |
| 0B | SEV  |       | 2 | 1 | 3F | SWI  | 12    | 1 | 73 | COM  |       | 6     | 3 | A7 | STAA |       | 4     | 2 | DB | ADDB |       | 3     | 2 |   |
| 0C | CLC  |       | 2 | 1 | 40 | NEGA | 2     | 1 | 74 | LSR  |       | 6     | 3 | A8 | EORA |       | 4     | 2 | DC | LDD  |       | 4     | 2 |   |
| 0D | SEC  |       | 2 | 1 | 41 | *    |       |   | 75 | *    |       |       |   | A9 | ADCA |       | 4     | 2 | DD | STD  |       | 4     | 2 |   |
| 0E | CLI  |       | 2 | 1 | 42 | *    |       |   | 76 | ROR  |       | 6     | 3 | AA | ORAA |       | 4     | 2 | DE | LDX  |       | 4     | 2 |   |
| 0F | SEI  |       | 2 | 1 | 43 | COMA | 2     | 1 | 77 | ASR  |       | 6     | 3 | AB | ADDA |       | 4     | 2 | DF | STX  | DIR   | 4     | 2 |   |
| 10 | SBA  |       | 2 | 1 | 44 | LSRA | 2     | 1 | 78 | ASL  |       | 6     | 3 | AC | CPX  |       | 6     | 2 | E0 | SUBB | INDXD | 4     | 2 |   |
| 11 | CBA  |       | 2 | 1 | 45 | *    |       |   | 79 | ROL  |       | 6     | 3 | AD | JSR  |       | 6     | 2 | E1 | CMPB |       | 4     | 2 |   |
| 12 | *    |       |   |   | 46 | RORA |       | 2 | 1  | 7A   | DEC   |       | 6 | 3  | AE   | LDS   | ▼     | 5 | 2  | E2   | SBCB  | ▲     | 4 | 2 |
| 13 | *    |       |   |   | 47 | ASRA | 2     | 1 | 7B | *    |       |       |   | AF | STS  | INDXD | 5     | 2 | E3 | ADD  |       | 6     | 2 |   |
| 14 | *    |       |   |   | 48 | ASLA | 2     | 1 | 7C | INC  |       | 6     | 3 | B0 | SUBA | EXTND | 4     | 3 | E4 | ANDB |       | 4     | 2 |   |
| 15 | *    |       |   |   | 49 | ROLA | 2     | 1 | 7D | TST  |       | 6     | 3 | B1 | CMPA | ▲     | 4     | 3 | E5 | BITB |       | 4     | 2 |   |
| 16 | TAB  |       | 2 | 1 | 4A | DECA | 2     | 1 | 7E | JMP  |       | 3     | 3 | B2 | SBCA |       | 4     | 3 | E6 | LDAB |       | 4     | 2 |   |
| 17 | TBA  |       | 2 | 1 | 4B | *    |       |   | 7F | CLR  | EXTND | 6     | 3 | B3 | SUBD |       | 6     | 3 | E7 | STAB |       | 4     | 2 |   |
| 18 | *    |       |   |   | 4C | INCA | 2     | 1 | 80 | SUBA | IMMED | 2     | 2 | B4 | ANDA |       | 4     | 3 | E8 | EORB |       | 4     | 2 |   |
| 19 | DAA  | INHER | 2 | 1 | 4D | TSTA | 2     | 1 | 81 | CMPA |       | 2     | 2 | B5 | BITA |       | 4     | 3 | E9 | ADCB |       | 4     | 2 |   |
| 1A | *    |       |   |   | 4E | T    |       |   | 82 | SBCA |       | 2     | 2 | B6 | LDAA |       | 4     | 3 | EA | ORAB |       | 4     | 2 |   |
| 1B | ABA  | INHER | 2 | 1 | 4F | CLRA | 2     | 1 | 83 | SUBD |       | 4     | 3 | B7 | STAA |       | 4     | 3 | EB | ADDB |       | 4     | 2 |   |
| 1C | *    |       |   |   | 50 | NEGB | 2     | 1 | 84 | ANDA |       | 2     | 2 | B8 | EORA |       | 4     | 3 | EC | LDD  |       | 5     | 2 |   |
| 1D | *    |       |   |   | 51 | *    |       |   | 85 | BITA |       | 2     | 2 | B9 | ADCA |       | 4     | 3 | ED | STD  |       | 5     | 2 |   |
| 1E | *    |       |   |   | 52 | *    |       |   | 86 | LDAA |       | 2     | 2 | BA | ORAA |       | 4     | 3 | EE | LDX  | ▼     | 5     | 2 |   |
| 1F | *    |       |   |   | 53 | COMB | 2     | 1 | 87 | *    |       |       |   | BB | ADDA |       | 4     | 3 | EF | STX  | INDXD | 5     | 2 |   |
| 20 | BRA  | REL   | 3 | 2 | 54 | LSRB | 2     | 1 | 88 | EORA |       | 2     | 2 | BC | CPX  |       | 6     | 3 | F0 | SUBB | EXTND | 4     | 3 |   |
| 21 | BRN  |       | 3 | 2 | 55 | *    |       |   | 89 | ADCA |       | 2     | 2 | BD | JSR  |       | 6     | 3 | F1 | CMPB |       | 4     | 3 |   |
| 22 | BHI  |       | 3 | 2 | 56 | RORB | 2     | 1 | 8A | ORAA |       | 2     | 2 | BE | LDS  |       | 5     | 3 | F2 | SBCB |       | 4     | 3 |   |
| 23 | BLS  |       | 3 | 2 | 57 | ASRB | 2     | 1 | 8B | ADDA |       | 2     | 2 | BF | STS  | EXTND | 5     | 3 | F3 | ADD  |       | 6     | 3 |   |
| 24 | BCC  |       | 3 | 2 | 58 | ASLB | 2     | 1 | 8C | CPX  | IMMED | 4     | 3 | C0 | SUBB | IMMED | 2     | 2 | F4 | ANDB |       | 4     | 3 |   |
| 25 | BCS  |       | 3 | 2 | 59 | ROLB | 2     | 1 | 8D | BSR  | REL   | 6     | 2 | C1 | CMPB |       | 2     | 2 | F5 | BITB |       | 4     | 3 |   |
| 26 | BNE  |       | 3 | 2 | 5A | DECB | 2     | 1 | 8E | LDS  | IMMED | 3     | 3 | C2 | SBCB |       | 2     | 2 | F6 | LDAB |       | 4     | 3 |   |
| 27 | BEQ  |       | 3 | 2 | 5B | *    |       |   | 8F | *    |       |       |   | C3 | ADD  |       | 4     | 3 | F7 | STAB |       | 4     | 3 |   |
| 28 | BVC  |       | 3 | 2 | 5C | INCB | 2     | 1 | 90 | SUBA | DIR   | 3     | 2 | C4 | ANDB |       | 2     | 2 | F8 | EORB |       | 4     | 3 |   |
| 29 | BVS  |       | 3 | 2 | 5D | TSTB | 2     | 1 | 91 | CMPA |       | 3     | 2 | C5 | BITB |       | 2     | 2 | F9 | ADCB |       | 4     | 3 |   |
| 2A | BPL  |       | 3 | 2 | 5E | T    |       |   | 92 | SBCA |       | 3     | 2 | C6 | LDAB |       | 2     | 2 | FA | ORAB |       | 4     | 3 |   |
| 2B | BMI  |       | 3 | 2 | 5F | CLRB | INHER | 2 | 1  | 93   | SUBD  |       | 5 | 2  | C7   | *     |       |   | FB | ADDB |       | 4     | 3 |   |
| 2C | BGE  |       | 3 | 2 | 60 | NEG  | INDXD | 6 | 2  | 94   | ANDA  |       | 3 | 2  | C8   | EORB  |       | 2 | 2  | FC   | LDD   |       | 5 | 3 |
| 2D | BLT  |       | 3 | 2 | 61 | *    |       |   | 95 | BITA |       | 3     | 2 | C9 | ADCB |       | 2     | 2 | FD | STD  |       | 5     | 3 |   |
| 2E | BGT  |       | 3 | 2 | 62 | *    |       |   | 96 | LDAA |       | 3     | 2 | CA | ORAB |       | 2     | 2 | FE | LDX  | ▼     | 5     | 3 |   |
| 2F | BLE  | REL   | 3 | 2 | 63 | COM  |       | 6 | 2  | 97   | STAA  |       | 3 | 2  | CB   | ADDB  |       | 2 | 2  | FF   | STX   | EXTND | 5 | 3 |
| 30 | TSX  | INHER | 3 | 1 | 64 | LSR  |       | 6 | 2  | 98   | EORA  |       | 3 | 2  | CC   | LDD   |       | 3 | 3  |      |       |       |   |   |
| 31 | INS  |       | 3 | 1 | 65 | *    |       |   | 99 | ADCA |       | 3     | 2 | CD | *    |       |       |   |    |      |       |       |   |   |
| 32 | PULA |       | 4 | 1 | 66 | ROR  |       | 6 | 2  | 9A   | ORAA  |       | 3 | 2  | CE   | LDX   | IMMED | 3 | 3  |      |       |       |   |   |
| 33 | PULB |       | 4 | 1 | 67 | ASR  | INDXD | 6 | 2  | 9B   | ADDA  |       | 3 | 2  | CF   | *     |       |   |    |      |       |       |   |   |

UNDEFINED OP CODE

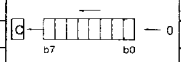
**Notes :**

- Addressing Modes  
 INHER = Inherent  
 REL = Relative  
 INDXD = Indexed  
 EXTND = Extended  
 IMMED = Immediate  
 DIR = Direct
- Unassigned opcodes are indicated by "." and should not be executed.
- Codes marked by "T" force the PC to function as a 16-bit counter.

**Table 9 : Index Register and Stack Manipulation Instructions.**

| Pointer Operations                      | Mnem | Immed |     | Direct |     | Index |     | Extnd |     | Inherent |     | Boolean/<br>Arithmetic Operation | Condition Codes |                                                                                                    |                                                |   |   |   |   |   |   |
|-----------------------------------------|------|-------|-----|--------|-----|-------|-----|-------|-----|----------|-----|----------------------------------|-----------------|----------------------------------------------------------------------------------------------------|------------------------------------------------|---|---|---|---|---|---|
|                                         |      | OP    | ~ # | OP     | ~ # | OP    | ~ # | OP    | ~ # | OP       | ~ # |                                  | 5               | 4                                                                                                  | 3                                              | 2 | 1 | 0 |   |   |   |
|                                         |      |       |     |        |     |       |     |       |     |          |     |                                  | H               | I                                                                                                  | N                                              | Z | V | C |   |   |   |
| Compare Index Register                  | CPX  | 8C    | 4   | 3      | 9C  | 5     | 2   | AC    | 6   | 2        | BC  | 6                                | 3               |                                                                                                    |                                                |   |   |   |   |   |   |
| Decrement Index Register                | DEX  |       |     |        |     |       |     |       |     |          | 09  | 3                                | 1               | $X - 1 \rightarrow X$                                                                              | *                                              | * | * | * | * | * |   |
| Decrement Stack Pointer                 | DES  |       |     |        |     |       |     |       |     |          | 34  | 3                                | 1               | $SP - 1 \rightarrow SP$                                                                            | *                                              | * | * | * | * | * |   |
| Increment Index Register                | INX  |       |     |        |     |       |     |       |     |          | 08  | 3                                | 1               | $X + 1 \rightarrow X$                                                                              | *                                              | * | * | * | * | * |   |
| Increment Stack Pointer                 | INS  |       |     |        |     |       |     |       |     |          | 31  | 3                                | 1               | $1 SP + 1 \rightarrow SP$                                                                          | *                                              | * | * | * | * | * |   |
| Load Index Register                     | LDX  | CE    | 3   | 3      | DE  | 4     | 2   | EE    | 5   | 2        | FE  | 5                                | 3               |                                                                                                    | $M \rightarrow X_H, (M + 1) \rightarrow X_L$   | * | * | * | * | R | * |
| Load Stack Pointer                      | LDS  | 8E    | 3   | 3      | 9E  | 4     | 2   | AE    | 5   | 2        | BE  | 5                                | 3               |                                                                                                    | $M \rightarrow SP_H, (M + 1) \rightarrow SP_L$ | * | * | * | * | R | * |
| Store Index Register                    | STX  |       |     |        | DF  | 4     | 2   | EF    | 5   | 2        | FF  | 5                                | 3               |                                                                                                    | $X_H \rightarrow M, X_L \rightarrow (M + 1)$   | * | * | * | * | R | * |
| Store Stack Pointer                     | STS  |       |     |        | 9F  | 4     | 2   | AF    | 5   | 2        | BF  | 5                                | 3               |                                                                                                    | $SP_H \rightarrow M, SP_L \rightarrow (M + 1)$ | * | * | * | * | R | * |
| Index Reg $\rightarrow$ Stack Pointer   | TXS  |       |     |        |     |       |     |       |     |          | 35  | 3                                | 1               | $X - 1 \rightarrow SP$                                                                             | *                                              | * | * | * | * | * |   |
| Stack Pntr $\rightarrow$ Index Register | TSX  |       |     |        |     |       |     |       |     |          | 30  | 3                                | 1               | $SP + 1 \rightarrow X$                                                                             | *                                              | * | * | * | * | * |   |
| Add                                     | ABX  |       |     |        |     |       |     |       |     |          | 3A  | 3                                | 1               | $B + X \rightarrow X$                                                                              | *                                              | * | * | * | * | * |   |
| Push Data                               | PSHX |       |     |        |     |       |     |       |     |          | 3C  | 4                                | 1               | $X_L \rightarrow M_{SP}, SP - 1 \rightarrow SP$<br>$X_H \rightarrow M_{SP}, SP - 1 \rightarrow SP$ | *                                              | * | * | * | * | * |   |
| Pull Data                               | PULX |       |     |        |     |       |     |       |     |          | 38  | 5                                | 1               | $SP + 1 \rightarrow SP, M_{SP} \rightarrow X_H$<br>$SP + 1 \rightarrow SP, M_{SP} \rightarrow X_L$ | *                                              | * | * | * | * | * |   |

**Table 10 : Accumulator and Memory Instructions.**

| Accumulator and<br>Memory Operations | MNE  | Immed |     | Direct |     | Index |     | Extend |     | Inher |     | Boolean<br>Expression                                                              | Condition Codes |   |   |   |   |   |
|--------------------------------------|------|-------|-----|--------|-----|-------|-----|--------|-----|-------|-----|------------------------------------------------------------------------------------|-----------------|---|---|---|---|---|
|                                      |      | Op    | ~ # | Op     | ~ # | Op    | ~ # | Op     | ~ # | Op    | ~ # |                                                                                    | H               | I | N | Z | V | C |
|                                      |      |       |     |        |     |       |     |        |     |       |     |                                                                                    |                 |   |   |   |   |   |
| Add Acmltrs                          | ABA  |       |     |        |     |       |     |        |     | 1B    | 2 1 | $A + B \rightarrow A$                                                              | *               | * | * | * | * | * |
| Add B to X                           | ABX  |       |     |        |     |       |     |        |     | 3A    | 3 1 | $00 : B + X \rightarrow X$                                                         | *               | * | * | * | * | * |
| Add with Carry                       | ADCA | 89    | 2 2 | 99     | 3 2 | A9    | 4 2 | B9     | 4 3 |       |     | $A + M + C \rightarrow A$                                                          | *               | * | * | * | * | * |
|                                      | ADCB | C9    | 2 2 | D9     | 3 2 | E9    | 4 2 | F9     | 4 3 |       |     | $B + M + C \rightarrow B$                                                          | *               | * | * | * | * | * |
| Add                                  | ADDA | 8B    | 2 2 | 9B     | 3 2 | AB    | 4 2 | BB     | 4 3 |       |     | $A + M \rightarrow A$                                                              | *               | * | * | * | * | * |
|                                      | ADDB | CB    | 2 2 | DB     | 3 2 | EB    | 4 2 | FB     | 4 3 |       |     | $B + M \rightarrow A$                                                              | *               | * | * | * | * | * |
| Add Double                           | ADDD | C3    | 4 3 | D3     | 5 2 | E3    | 6 2 | F3     | 6 3 |       |     | $D + M : M + 1 \rightarrow D$                                                      | *               | * | * | * | * | * |
| And                                  | ANDA | 84    | 2 2 | 94     | 3 2 | A4    | 4 2 | B4     | 4 3 |       |     | $A - M \rightarrow A$                                                              | *               | * | * | * | R | * |
|                                      | ANDB | C4    | 2 2 | D4     | 3 2 | E4    | 4 2 | F4     | 4 3 |       |     | $B - M \rightarrow B$                                                              | *               | * | * | * | R | * |
| Shift Left, Arithmetic               | ASL  |       |     |        |     | 68    | 6 2 | 78     | 6 3 |       |     |  | *               | * | * | * | * | * |
|                                      | ASLA |       |     |        |     |       |     |        | 48  | 2 1   |     |                                                                                    | *               | * | * | * | * | * |
|                                      | ASLB |       |     |        |     |       |     |        |     | 58    | 2 1 |                                                                                    |                 | * | * | * | * | * |

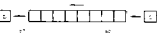
The Condition Code Register notes are listed after Table 12.

Table 10 : Accumulator and Memory Instructions (continued).

| Accumulator and Memory Operations | MNE  | Immed |     | Direct |     | Index |     | Extend |     | Inher |     | Boolean Expression | Condition Codes                                                                   |                                                                                     |                              |   |   |   |   |   |   |
|-----------------------------------|------|-------|-----|--------|-----|-------|-----|--------|-----|-------|-----|--------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|------------------------------|---|---|---|---|---|---|
|                                   |      | Op    | ~ # | Op     | ~ # | Op    | ~ # | Op     | ~ # | Op    | ~ # |                    | H                                                                                 | I                                                                                   | N                            | Z | V | C |   |   |   |
| Shift Left Dbl                    | ASLD |       |     |        |     |       |     |        |     | 05    | 3   | 1                  |                                                                                   | •                                                                                   | •                            | • | • | • | • |   |   |
| Shift Right, Arithmetic           | ASR  |       |     |        |     |       | 67  | 6      | 2   | 77    | 6   | 3                  |                                                                                   | •                                                                                   | •                            | • | • | • | • |   |   |
|                                   | ASRA |       |     |        |     |       |     |        |     | 47    | 2   | 1                  |  | •                                                                                   | •                            | • | • | • | • |   |   |
|                                   | ASRB |       |     |        |     |       |     |        |     |       | 57  | 2                  | 1                                                                                 |                                                                                     | •                            | • | • | • | • | • |   |
| Bit Test                          | BITA | 85    | 2   | 2      | 95  | 3     | 2   | A5     | 4   | 2     | B5  | 4                  | 3                                                                                 |                                                                                     | A ← M                        | • | • | • | • | R |   |
|                                   | BITB | C5    | 2   | 2      | D5  | 3     | 2   | E5     | 4   | 2     | F5  | 4                  | 3                                                                                 |                                                                                     | B ← M                        | • | • | • | • | R |   |
| Compare Acmltrs                   | CBA  |       |     |        |     |       |     |        |     |       |     | 11                 | 2                                                                                 | 1                                                                                   | A ← B                        | • | • | • | • | • |   |
| Clear                             | CLR  |       |     |        |     |       | 6F  | 6      | 2   | 7F    | 6   | 3                  |                                                                                   | 00 → M                                                                              | •                            | • | R | S | R | R |   |
|                                   | CLRA |       |     |        |     |       |     |        |     |       | 4F  | 2                  | 1                                                                                 | 00 → A                                                                              | •                            | • | R | S | R | R |   |
|                                   | CLRB |       |     |        |     |       |     |        |     |       |     | 5F                 | 2                                                                                 | 1                                                                                   | 00 → B                       | • | • | R | S | R | R |
| Compare                           | CMPA | 81    | 2   | 2      | 91  | 3     | 2   | A1     | 4   | 2     | B1  | 4                  | 3                                                                                 |                                                                                     | A ← M                        | • | • | • | • | • |   |
|                                   | CMPB | C1    | 2   | 2      | D1  | 3     | 2   | E1     | 4   | 2     | F1  | 4                  | 3                                                                                 |                                                                                     | B ← M                        | • | • | • | • | • |   |
| 1's Complement                    | COM  |       |     |        |     |       | 63  | 6      | 2   | 73    | 6   | 3                  |                                                                                   | M → M                                                                               | •                            | • | • | • | R | S |   |
|                                   | COMA |       |     |        |     |       |     |        |     |       | 43  | 2                  | 1                                                                                 | $\overline{A} \rightarrow A$                                                        | •                            | • | • | • | R | S |   |
|                                   | COMB |       |     |        |     |       |     |        |     |       |     | 53                 | 2                                                                                 | 1                                                                                   | $\overline{B} \rightarrow B$ | • | • | • | • | R | S |
| Decimal Adj., A                   | DAA  |       |     |        |     |       |     |        |     |       | 19  | 2                  | 1                                                                                 | Adj binary sum to BCD                                                               | •                            | • | • | • | • | • |   |
| Decrement                         | DEC  |       |     |        |     |       | 6A  | 6      | 2   | 7A    | 6   | 3                  |                                                                                   | M ← 1 → M                                                                           | •                            | • | • | • | • | • |   |
|                                   | DECA |       |     |        |     |       |     |        |     |       | 4A  | 2                  | 1                                                                                 | A ← 1 → A                                                                           | •                            | • | • | • | • | • |   |
|                                   | DECB |       |     |        |     |       |     |        |     |       | 5A  | 2                  | 1                                                                                 | B ← 1 → B                                                                           | •                            | • | • | • | • | • |   |
| Exclusive OR                      | EORA | 88    | 2   | 2      | 98  | 3     | 2   | A8     | 4   | 2     | B8  | 4                  | 3                                                                                 |                                                                                     | A ⊕ M → A                    | • | • | • | • | R | • |
|                                   | EORB | C8    | 2   | 2      | D8  | 3     | 2   | E8     | 4   | 2     | F8  | 4                  | 3                                                                                 |                                                                                     | B ⊕ M → B                    | • | • | • | • | R | • |
| Increment                         | INC  |       |     |        |     |       | 6C  | 6      | 2   | 7C    | 6   | 3                  |                                                                                   | M + 1 → M                                                                           | •                            | • | • | • | • | • |   |
|                                   | INCA |       |     |        |     |       |     |        |     |       | 4C  | 2                  | 1                                                                                 | A + 1 → A                                                                           | •                            | • | • | • | • | • |   |
|                                   | INCB |       |     |        |     |       |     |        |     |       |     | 5C                 | 2                                                                                 | 1                                                                                   | B + 1 → B                    | • | • | • | • | • | • |
| Load Acmltrs                      | LDAA | 86    | 2   | 2      | 96  | 3     | 2   | A6     | 4   | 2     | B6  | 4                  | 3                                                                                 |                                                                                     | M → A                        | • | • | • | • | R | • |
|                                   | LDAB | C6    | 2   | 2      | D6  | 3     | 2   | E6     | 4   | 2     | F6  | 4                  | 3                                                                                 |                                                                                     | M → B                        | • | • | • | • | R | • |
| Load Double                       | LDD  | CC    | 3   | 3      | DC  | 4     | 2   | EC     | 5   | 2     | FC  | 5                  | 3                                                                                 |                                                                                     | M : M + 1 → D                | • | • | • | • | R | • |
| Logical Shift, Left               | LSL  |       |     |        |     |       | 68  | 6      | 2   | 78    | 6   | 3                  |                                                                                   |                                                                                     | •                            | • | • | • | • | • |   |
|                                   | LSLA |       |     |        |     |       |     |        |     |       | 48  | 2                  | 1                                                                                 |    | •                            | • | • | • | • | • |   |
|                                   | LSLB |       |     |        |     |       |     |        |     |       | 58  | 2                  | 1                                                                                 |                                                                                     | •                            | • | • | • | • | • |   |
|                                   | LSLD |       |     |        |     |       |     |        |     |       |     | 05                 | 3                                                                                 | 2                                                                                   |                              | • | • | • | • | • | • |
| Shift Right, Logical              | LSR  |       |     |        |     |       | 64  | 6      | 2   | 74    | 6   | 3                  |                                                                                   |                                                                                     | •                            | • | R | • | • | • |   |
|                                   | LSRA |       |     |        |     |       |     |        |     |       | 44  | 2                  | 1                                                                                 |  | •                            | • | R | • | • | • |   |
|                                   | LSRB |       |     |        |     |       |     |        |     |       | 54  | 2                  | 1                                                                                 |                                                                                     | •                            | • | R | • | • | • |   |
|                                   | LSRD |       |     |        |     |       |     |        |     |       |     | 04                 | 3                                                                                 | 1                                                                                   |                              | • | • | R | • | • | • |

The Condition Code Register notes are listed after Table 12.

Table 10 : Accumulator and Memory Instructions (continued).

| Accumulator and Memory Operations | MNE  | Immed |     | Direct |     | Index |     | Extend |     | Inher |      | Boolean Expression                                                                | Condition Codes        |   |   |   |   |   |
|-----------------------------------|------|-------|-----|--------|-----|-------|-----|--------|-----|-------|------|-----------------------------------------------------------------------------------|------------------------|---|---|---|---|---|
|                                   |      | Op    | ~ # | Op     | ~ # | Op    | ~ # | Op     | ~ # | Op    | ~ #  |                                                                                   | H                      | I | N | Z | V | C |
| Multiply                          | MUL  |       |     |        |     |       |     |        |     | 3D    | 10 1 | $A \times B \rightarrow D$                                                        | *                      | * | * | * | * | * |
| 2's Complement (negate)           | NEG  |       |     |        |     | 60    | 6 2 | 70     | 6 3 |       |      | $00 - M \rightarrow M$                                                            | *                      | * | * | * | * | * |
|                                   | NEGA |       |     |        |     |       |     |        |     | 40    | 2 1  | $00 - A \rightarrow A$                                                            | *                      | * | * | * | * | * |
|                                   | NEGB |       |     |        |     |       |     |        |     |       | 50   | 2 1                                                                               | $00 - B \rightarrow B$ | * | * | * | * | * |
| No Operation                      | NOP  |       |     |        |     |       |     |        |     | 01    | 2 1  | $PC + 1 \rightarrow PC$                                                           | *                      | * | * | * | * | * |
| Inclusive OR                      | ORAA | 8A    | 2 2 | 9A     | 3 2 | AA    | 4 2 | BA     | 4 3 |       |      | $A + M \rightarrow A$                                                             | *                      | * | * | * | R | * |
|                                   | ORAB | CA    | 2 2 | DA     | 3 2 | EA    | 4 2 | FA     | 4 3 |       |      | $B + M \rightarrow B$                                                             | *                      | * | * | * | R | * |
| Push Data                         | PSHA |       |     |        |     |       |     |        |     | 36    | 3 1  | $A \rightarrow \text{Stack}$                                                      | *                      | * | * | * | * | * |
|                                   | PSHB |       |     |        |     |       |     |        |     | 37    | 3 1  | $B \rightarrow \text{Stack}$                                                      | *                      | * | * | * | * | * |
| Pull Data                         | PULA |       |     |        |     |       |     |        |     | 32    | 4 1  | $\text{Stack} \rightarrow A$                                                      | *                      | * | * | * | * | * |
|                                   | PULB |       |     |        |     |       |     |        |     | 33    | 4 1  | $\text{Stack} \rightarrow B$                                                      | *                      | * | * | * | * | * |
| Rotate Left                       | ROL  |       |     |        |     | 69    | 6 2 | 79     | 6 3 |       |      |                                                                                   | *                      | * | * | * | * | * |
|                                   | ROLA |       |     |        |     |       |     |        |     | 49    | 2 1  |  | *                      | * | * | * | * | * |
|                                   | ROLB |       |     |        |     |       |     |        |     |       | 59   | 2 1                                                                               |                        | * | * | * | * | * |
| Rotate Right                      | ROR  |       |     |        |     | 66    | 6 2 | 76     | 6 3 |       |      |                                                                                   | *                      | * | * | * | * | * |
|                                   | RORA |       |     |        |     |       |     |        |     | 46    | 2 1  |  | *                      | * | * | * | * | * |
|                                   | RORB |       |     |        |     |       |     |        |     |       | 56   | 2 1                                                                               |                        | * | * | * | * | * |
| Subtract Acmltr                   | SBA  |       |     |        |     |       |     |        |     | 10    | 2 1  | $A - B \rightarrow A$                                                             | *                      | * | * | * | * | * |
| Subtract with Carry               | SBCA | 82    | 2 2 | 92     | 3 2 | A2    | 4 2 | B2     | 4 3 |       |      | $A - M - C \rightarrow A$                                                         | *                      | * | * | * | * | * |
|                                   | SBCB | C2    | 2 2 | D2     | 3 2 | E2    | 4 2 | F2     | 4 3 |       |      | $B - M - C \rightarrow B$                                                         | *                      | * | * | * | * | * |
| Store Acmltrs                     | STAA |       |     |        | 97  | 3 2   | A7  | 4 2    | B7  | 4 3   |      | $A \rightarrow M$                                                                 | *                      | * | * | * | R | * |
|                                   | STAB |       |     |        | D7  | 3 2   | E7  | 4 2    | F7  | 4 3   |      | $B \rightarrow M$                                                                 | *                      | * | * | * | R | * |
|                                   | STD  |       |     |        | DD  | 4 2   | ED  | 5 2    | FD  | 5 3   |      | $D \rightarrow M : M + 1$                                                         | *                      | * | * | * | R | * |
| Subtract                          | SUBA | 80    | 2 2 | 90     | 3 2 | A0    | 4 2 | B0     | 4 3 |       |      | $A - M \rightarrow A$                                                             | *                      | * | * | * | * | * |
|                                   | SUBB | C0    | 2 2 | D0     | 3 2 | E0    | 4 2 | F0     | 4 3 |       |      | $B - M \rightarrow B$                                                             | *                      | * | * | * | * | * |
| Subtract Double                   | SUBD | 83    | 4 3 | 93     | 5 2 | A3    | 6 2 | B3     | 6 3 |       |      | $D - M : M + 1 \rightarrow D$                                                     | *                      | * | * | * | * | * |
| Transfer Acmltr                   | TAB  |       |     |        |     |       |     |        |     | 16    | 2 1  | $A \rightarrow B$                                                                 | *                      | * | * | * | R | * |
|                                   | TBA  |       |     |        |     |       |     |        |     | 17    | 2 1  | $B \rightarrow A$                                                                 | *                      | * | * | * | R | * |
| Test, Zero or Minus               | TST  |       |     |        |     | 6D    | 6 2 | 7D     | 6 3 |       |      | $M - 00$                                                                          | *                      | * | * | * | R | R |
|                                   | TSTA |       |     |        |     |       |     |        |     | 4D    | 2 1  | $A - 00$                                                                          | *                      | * | * | * | R | R |
|                                   | TSTB |       |     |        |     |       |     |        |     | 5D    | 2 1  | $B - 00$                                                                          | *                      | * | * | * | R | R |

The Condition Code Register notes are listed after table 12.

Table 11 : Jump and Branch Instructions.

| Operations               | MNEM | Direct |     | Relative |     | Index |     | Extend |     | Inherent |     | Branch Test                      | Cond. Code Reg. |    |    |   |   |                                  |   |   |   |   |   |   |
|--------------------------|------|--------|-----|----------|-----|-------|-----|--------|-----|----------|-----|----------------------------------|-----------------|----|----|---|---|----------------------------------|---|---|---|---|---|---|
|                          |      | Op     | ~ # | Op       | ~ # | Op    | ~ # | Op     | ~ # | Op       | ~ # |                                  | 5               | 4  | 3  | 2 | 1 | 0                                |   |   |   |   |   |   |
|                          |      |        |     |          |     |       |     |        |     |          |     |                                  | H               | I  | N  | Z | V | C                                |   |   |   |   |   |   |
| Branch Always            | BRA  |        |     | 20       | 3   | 2     |     |        |     |          |     | None                             | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch Never             | BRN  |        |     | 21       | 3   | 2     |     |        |     |          |     | None                             | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if Carry Clear    | BCC  |        |     | 24       | 3   | 2     |     |        |     |          |     | C = 0                            | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if Carry Set      | BCS  |        |     | 25       | 3   | 2     |     |        |     |          |     | C = 1                            | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if = Zero         | BEQ  |        |     | 27       | 3   | 2     |     |        |     |          |     | Z = 1                            | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if ≥ Zero         | BGE  |        |     | 2C       | 3   | 2     |     |        |     |          |     | N ⊕ V = 0                        | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if > Zero         | BGT  |        |     | 2E       | 3   | 2     |     |        |     |          |     | Z + (N ⊕ V) = 0                  | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if Higher         | BHI  |        |     | 22       | 3   | 2     |     |        |     |          |     | C + Z = 0                        | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if Higher or Same | BHS  |        |     | 24       | 3   | 2     |     |        |     |          |     | C = 0                            | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if ≤ Zero         | BLE  |        |     | 2F       | 3   | 2     |     |        |     |          |     | Z + (N ⊕ V) = 1                  | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if Carry Set      | BLO  |        |     | 25       | 3   | 2     |     |        |     |          |     | C = 1                            | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if Lower or Same  | BLS  |        |     | 23       | 3   | 2     |     |        |     |          |     | C + Z = 1                        | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if < Zero         | BLT  |        |     | 2D       | 3   | 2     |     |        |     |          |     | N ⊕ V = 1                        | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if Minus          | BMI  |        |     | 2B       | 3   | 2     |     |        |     |          |     | N = 1                            | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if not Equal Zero | BNE  |        |     | 26       | 3   | 2     |     |        |     |          |     | Z = 0                            | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if Overflow Clear | BVC  |        |     | 28       | 3   | 2     |     |        |     |          |     | V = 0                            | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if Overflow Set   | BVS  |        |     | 29       | 3   | 2     |     |        |     |          |     | V = 1                            | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch if Plus           | BPL  |        |     | 2A       | 3   | 2     |     |        |     |          |     | N = 0                            | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Branch to Subroutine     | BSR  |        |     | 8D       | 6   | 2     |     |        |     |          |     | See Special Operations-figure 26 | *               | *  | *  | * | * | *                                |   |   |   |   |   |   |
| Jump                     | JMP  |        |     |          |     |       | 6E  | 3      | 2   | 7E       | 3   |                                  | 3               |    | *  | * | * | *                                | * |   |   |   |   |   |
| Jump to Subroutine       | JSR  | 9D     | 5   | 2        |     |       |     | AD     | 6   | 2        | BD  | 6                                | 3               |    |    |   |   | *                                | * | * | * | * |   |   |
| No Operation             | NOP  |        |     |          |     |       |     |        |     |          |     |                                  | 01              | 2  | 1  |   |   | *                                | * | * | * | * |   |   |
| Return from Interrupt    | RTI  |        |     |          |     |       |     |        |     |          |     |                                  |                 | 3B | 10 | 1 |   | See Special Operations-figure 26 | ⌢ | ⌢ | ⌢ | ⌢ | ⌢ | ⌢ |
| Return from Subroutine   | RTS  |        |     |          |     |       |     |        |     |          |     |                                  |                 | 39 | 5  | 1 |   |                                  | * | * | * | * | * | * |
| Software Interrupt       | SWI  |        |     |          |     |       |     |        |     |          |     |                                  |                 | 3F | 12 | 1 |   |                                  | * | S | * | * | * | * |
| Wait for Interrupt       | WAI  |        |     |          |     |       |     |        |     |          |     |                                  |                 | 3E | 9  | 1 |   |                                  | * | * | * | * | * | * |

The Condition Code Register notes are listed after Table 12.

**Table 12** : Condition Code Register Manipulation Instructions.

| Operations           | Inherent |    |   |   | Boolean Operation | Condition Code Register |   |   |   |   |   |
|----------------------|----------|----|---|---|-------------------|-------------------------|---|---|---|---|---|
|                      |          |    |   |   |                   | 5                       | 4 | 3 | 2 | 1 | 0 |
|                      | Mnem     | OP | ~ | # |                   | H                       | I | N | Z | V | C |
| Clear Carry          | CLC      | 0C | 2 | 1 | 0 → C             | •                       | • | • | • | • | R |
| Clear Interrupt Mask | CLI      | 0E | 2 | 1 | 0 → I             | •                       | R | • | • | • | • |
| Clear Overflow       | CLV      | 0A | 2 | 1 | 0 → V             | •                       | • | • | • | R | • |
| Set Carry            | SEC      | 0D | 2 | 1 | 1 → C             | •                       | • | • | • | • | S |
| Set Interrupt Mask   | SEI      | 0F | 2 | 1 | 1 → I             | •                       | S | • | • | • | • |
| Set Overflow         | SEV      | 0B | 2 | 1 | 1 → V             | •                       | • | • | • | S | • |
| Accumulator A → CCR  | TAP      | 06 | 2 | 1 | A → CCR           | ↕                       | ↕ | ↕ | ↕ | ↕ | ↕ |
| CCR → Accumulator A  | TPA      | 07 | 2 | 1 | CCR → A           | •                       | • | • | • | • | • |

**LEGEND**

OP Operation Code (Hexadecimal)  
 - Number of MPU Cycles  
 M<sub>SP</sub> Contents of memory location pointed to by Stack Pointer  
 # Number of Program Bytes  
 + Arithmetic Plus  
 - Arithmetic Minus  
 • Boolean AND  
 X Arithmetic Multiply  
 + Boolean Inclusive OR  
 ⊕ Boolean Exclusive OR  
 M Complement of M  
 → Transfer Into  
 0 Bit = Zero  
 00 Byte = 0

**CONDITION CODE SYMBOLS**

H Half-carry from bit 3  
 I Interrupt mask  
 N Negative (sign bit)  
 Z Zero (byte)  
 V Overflow, 2's complement  
 C Carry/Borrow from MSB  
 R Reset Always  
 S Set Always  
 ↕ Affected  
 • Not Affected

**Table 13 : Instruction Execution Times In E-cycles.**

|      | Addressing Mode |        |          |         |          |          |
|------|-----------------|--------|----------|---------|----------|----------|
|      | Immediate       | Direct | Extended | Indexed | Inherent | Relative |
| ABA  | ●               | ●      | ●        | ●       | 2        | ●        |
| ABX  | ●               | ●      | ●        | ●       | 3        | ●        |
| ADC  | 2               | 3      | 4        | 4       | ●        | ●        |
| ADD  | 2               | 3      | 4        | 4       | ●        | ●        |
| ADDD | 4               | 5      | 6        | 6       | ●        | ●        |
| AND  | 2               | 3      | 4        | 4       | ●        | ●        |
| ASL  | ●               | ●      | 6        | 6       | 2        | ●        |
| ASLD | ●               | ●      | ●        | ●       | 3        | ●        |
| ASR  | ●               | ●      | 6        | 6       | 2        | ●        |
| BCC  | ●               | ●      | ●        | ●       | ●        | 3        |
| BCS  | ●               | ●      | ●        | ●       | ●        | 3        |
| BEQ  | ●               | ●      | ●        | ●       | ●        | 3        |
| BGE  | ●               | ●      | ●        | ●       | ●        | 3        |
| BGT  | ●               | ●      | ●        | ●       | ●        | 3        |
| BHI  | ●               | ●      | ●        | ●       | ●        | 3        |
| BHS  | ●               | ●      | ●        | ●       | ●        | 3        |
| BIT  | 2               | 3      | 4        | 4       | ●        | ●        |
| BLE  | ●               | ●      | ●        | ●       | ●        | 3        |
| BLO  | ●               | ●      | ●        | ●       | ●        | 3        |
| BLS  | ●               | ●      | ●        | ●       | ●        | 3        |
| BLT  | ●               | ●      | ●        | ●       | ●        | 3        |
| BMI  | ●               | ●      | ●        | ●       | ●        | 3        |
| BNE  | ●               | ●      | ●        | ●       | ●        | 3        |
| BPL  | ●               | ●      | ●        | ●       | ●        | 3        |
| BRA  | ●               | ●      | ●        | ●       | ●        | 3        |
| BRN  | ●               | ●      | ●        | ●       | ●        | 3        |
| BSR  | ●               | ●      | ●        | ●       | ●        | 6        |
| BVC  | ●               | ●      | ●        | ●       | ●        | 3        |
| BVS  | ●               | ●      | ●        | ●       | ●        | 3        |
| CBA  | ●               | ●      | ●        | ●       | 2        | ●        |
| CLC  | ●               | ●      | ●        | ●       | 2        | ●        |
| CLI  | ●               | ●      | ●        | ●       | 2        | ●        |
| CLR  | ●               | ●      | 6        | 6       | 2        | ●        |
| CLV  | ●               | ●      | ●        | ●       | 2        | ●        |
| CMP  | 2               | 3      | 4        | 4       | ●        | ●        |
| COM  | ●               | ●      | 6        | 6       | 2        | ●        |
| CPX  | 4               | 5      | 6        | 6       | ●        | ●        |
| DAA  | ●               | ●      | ●        | ●       | 2        | ●        |
| DEC  | ●               | ●      | 6        | 6       | 2        | ●        |
| DES  | ●               | ●      | ●        | ●       | 3        | ●        |
| DEX  | ●               | ●      | ●        | ●       | 3        | ●        |
| EOR  | 2               | 3      | 4        | 4       | ●        | ●        |
| INC  | ●               | ●      | 6        | 6       | ●        | ●        |
| INS  | ●               | ●      | ●        | ●       | 3        | ●        |

|      | Addressing Mode |        |          |         |          |          |
|------|-----------------|--------|----------|---------|----------|----------|
|      | Immediate       | Direct | Extended | Indexed | Inherent | Relative |
| INX  | ●               | ●      | ●        | ●       | 3        | ●        |
| JMP  | ●               | ●      | 3        | 3       | ●        | ●        |
| JSR  | ●               | 5      | 6        | 6       | ●        | ●        |
| LDA  | 2               | 3      | 4        | 4       | ●        | ●        |
| LDD  | 3               | 4      | 5        | 5       | ●        | ●        |
| LDS  | 3               | 4      | 5        | 5       | ●        | ●        |
| LDX  | 3               | 4      | 5        | 5       | ●        | ●        |
| LSL  | ●               | ●      | 6        | 6       | 2        | ●        |
| LSLD | ●               | ●      | ●        | ●       | 3        | ●        |
| LSR  | ●               | ●      | 6        | 6       | 2        | ●        |
| LSRD | ●               | ●      | ●        | ●       | 3        | ●        |
| MUL  | ●               | ●      | ●        | ●       | 10       | ●        |
| NEG  | ●               | ●      | 6        | 6       | 2        | ●        |
| NOP  | ●               | ●      | ●        | ●       | 2        | ●        |
| ORA  | 2               | 3      | 4        | 4       | ●        | ●        |
| PSH  | ●               | ●      | ●        | ●       | 3        | ●        |
| PSHX | ●               | ●      | ●        | ●       | 4        | ●        |
| PUL  | ●               | ●      | ●        | ●       | 4        | ●        |
| PULX | ●               | ●      | ●        | ●       | 5        | ●        |
| ROL  | ●               | ●      | 6        | 6       | 2        | ●        |
| ROR  | ●               | ●      | 6        | 6       | 2        | ●        |
| RTI  | ●               | ●      | ●        | ●       | 10       | ●        |
| RTS  | ●               | ●      | ●        | ●       | 5        | ●        |
| SBA  | ●               | ●      | ●        | ●       | 2        | ●        |
| SBC  | 2               | 3      | 4        | 4       | ●        | ●        |
| SEC  | ●               | ●      | ●        | ●       | 2        | ●        |
| SEI  | ●               | ●      | ●        | ●       | 2        | ●        |
| SEV  | ●               | ●      | ●        | ●       | 2        | ●        |
| STA  | ●               | 3      | 4        | 4       | ●        | ●        |
| STD  | ●               | 4      | 5        | 5       | ●        | ●        |
| STS  | ●               | 4      | 5        | 5       | ●        | ●        |
| STX  | ●               | 4      | 5        | 5       | ●        | ●        |
| SUB  | 2               | 3      | 4        | 4       | ●        | ●        |
| SUBD | 4               | 5      | 6        | 6       | ●        | ●        |
| SWI  | ●               | ●      | ●        | ●       | 12       | ●        |
| TAB  | ●               | ●      | ●        | ●       | 2        | ●        |
| TAP  | ●               | ●      | ●        | ●       | 2        | ●        |
| TBA  | ●               | ●      | ●        | ●       | 2        | ●        |
| TPA  | ●               | ●      | ●        | ●       | 2        | ●        |
| TST  | ●               | ●      | 6        | 6       | 2        | ●        |
| TSX  | ●               | ●      | ●        | ●       | 3        | ●        |
| TXS  | ●               | ●      | ●        | ●       | 3        | ●        |
| WAI  | ●               | ●      | ●        | ●       | 9        | ●        |

## SUMMARY OF CYCLE-BY-CYCLE OPERATION

Table 14 provides a detailed description of the information present on the address bus, data bus, and the read/write (R/W) line during each cycle of each instruction.

The information is useful in comparing actual with expected results during debug of both software and hardware as the program is executed. The information is categorized in groups according to addressing

mode and number of cycles per instruction. In general, instructions with the same addressing mode and number of cycles execute in the same manner. Exceptions are indicated in the table.

Note that during MPU reads of internal locations, the resultant value will not appear on the external data bus except in mode 0. "High order" byte refers to the most-significant byte of a 16-bit value.

**Table 14** : Cycle-by-cycle Operation (sheet 1 of 5)

### IMMEDIATE

| Address Mode and Instructions                       | Cycles | Cycle #          | Address Bus                                                                    | R/W Line         | Data Bus                                                                                                |
|-----------------------------------------------------|--------|------------------|--------------------------------------------------------------------------------|------------------|---------------------------------------------------------------------------------------------------------|
| ADC EOR<br>ADD LDA<br>AND ORA<br>BIT SBC<br>CMP SUB | 2      | 1<br>2           | Opcode Address<br>Opcode Address + 1                                           | 1<br>1           | Opcode<br>Operand Data                                                                                  |
| LDS<br>LDX<br>LDD                                   | 3      | 1<br>2<br>3      | Opcode Address<br>Opcode Address + 1<br>Opcode Address + 2                     | 1<br>1<br>1      | Opcode<br>Operand Data (high order byte)<br>Operand Data (low order byte)                               |
| CPX<br>SUBD<br>ADDD                                 | 4      | 1<br>2<br>3<br>4 | Opcode Address<br>Opcode Address + 1<br>Opcode Address + 2<br>Address Bus FFFF | 1<br>1<br>1<br>1 | Opcode<br>Operand Data (high order byte)<br>Operand Data (low order byte)<br>Low Byte of Restart Vector |

### DIRECT

| Address Mode and Instructions                       | Cycles | Cycle #               | Address Bus                                                                                        | R/W Line              | Data Bus                                                                                                                      |
|-----------------------------------------------------|--------|-----------------------|----------------------------------------------------------------------------------------------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------|
| ADC EOR<br>ADD LDA<br>AND ORA<br>BIT SBC<br>CMP SUB | 3      | 1<br>2<br>3           | Opcode Address<br>Opcode Address + 1<br>Address of Operand                                         | 1<br>1<br>1           | Opcode<br>Address of Operand<br>Operand Data                                                                                  |
| STA                                                 | 3      | 1<br>2<br>3           | Opcode Address<br>Opcode Address + 1<br>Destination Address                                        | 1<br>1<br>0           | Opcode<br>Destination Address<br>Data from Accumulator                                                                        |
| LDS<br>LDX<br>LDD                                   | 4      | 1<br>2<br>3<br>4      | Opcode Address<br>Opcode Address + 1<br>Address of Operand<br>Operand Address + 1                  | 1<br>1<br>1<br>1      | Opcode Address<br>Address of Operand<br>Operand Data (high order byte)<br>Operand Data (low order byte)                       |
| STS<br>STX<br>STD                                   | 4      | 1<br>2<br>3<br>4      | Opcode Address<br>Opcode Address + 1<br>Address of Operand<br>Address of Operand + 1               | 1<br>1<br>0<br>0      | Opcode Address<br>Address of Operand<br>Operand Data (high order byte)<br>Operand Data (low order byte)                       |
| CPX<br>SUBD<br>ADDD                                 | 5      | 1<br>2<br>3<br>4<br>5 | Opcode Address<br>Opcode Address + 1<br>Operand Address<br>Operand Address + 1<br>Address Bus FFFF | 1<br>1<br>1<br>1<br>1 | Opcode<br>Address of Operand<br>Operand Data (high order byte)<br>Operand Data (low order byte)<br>Low Byte of Restart Vector |
| JSR                                                 | 5      | 1<br>2<br>3<br>4<br>5 | Opcode Address<br>Opcode Address + 1<br>Subroutine Address<br>Stack Pointer<br>Stack Pointer - 1   | 1<br>1<br>1<br>0<br>0 | Opcode<br>Irrelevant Data<br>First Subroutine Opcode<br>Return Address (low order byte)<br>Return Address (high order byte)   |

**Table 14** : Cycle-by-cycle Operation (sheet 2 of 5)

## EXTENDED

| Address Mode and Instructions                                                                | Cycles | Cycle # | Address Bus                 | R/W Line | Data Bus                              |
|----------------------------------------------------------------------------------------------|--------|---------|-----------------------------|----------|---------------------------------------|
| JMP                                                                                          | 3      | 1       | Opcode Address              | 1        | Opcode                                |
|                                                                                              |        | 2       | Opcode Address + 1          | 1        | Jump Address (high order byte)        |
|                                                                                              |        | 3       | Opcode Address + 2          | 1        | Jump Address (low order byte)         |
| ADC      EOR<br>ADD      LDA<br>AND      ORA<br><br>BIT      SBC<br>CMP      SUB             | 4      | 1       | Opcode Address              | 1        | Opcode                                |
|                                                                                              |        | 2       | Opcode Address + 1          | 1        | Address of Operand                    |
|                                                                                              |        | 3       | Opcode Address + 2          | 1        | Address of Operand (low order byte)   |
|                                                                                              |        | 4       | Address of Operand          | 1        | Operand Data                          |
| STA                                                                                          | 4      | 1       | Opcode Address              | 1        | Opcode                                |
|                                                                                              |        | 2       | Opcode Address + 1          | 1        | Destination Address (high order byte) |
|                                                                                              |        | 3       | Opcode Address + 2          | 1        | Destination Address (low order byte)  |
|                                                                                              |        | 4       | Operand Destination Address | 0        | Data from Accumulator                 |
| LDS<br>LDX<br><br>LDD                                                                        | 5      | 1       | Opcode Address              | 1        | Opcode                                |
|                                                                                              |        | 2       | Opcode Address + 1          | 1        | Address of Operand (high order byte)  |
|                                                                                              |        | 3       | Opcode Address + 2          | 1        | Address of Operand (low order byte)   |
|                                                                                              |        | 4       | Address of Operand          | 1        | Operand Data (high order byte)        |
|                                                                                              |        | 5       | Address of Operand + 1      | 1        | Operand Data (low order byte)         |
| STS<br>STX<br><br>STD                                                                        | 5      | 1       | Opcode Address              | 1        | Opcode                                |
|                                                                                              |        | 2       | Opcode Address + 1          | 1        | Address of Operand (high order byte)  |
|                                                                                              |        | 3       | Opcode Address + 2          | 1        | Address of Operand (low order byte)   |
|                                                                                              |        | 4       | Address of Operand          | 0        | Operand Data (high order byte)        |
|                                                                                              |        | 5       | Address of Operand + 1      | 0        | Operand Data (low order byte)         |
| ASL      LSR<br>ASR      NEG<br><br>CLR      ROL<br><br>COM      ROR<br>DEC      TST*<br>INC | 6      | 1       | Opcode Address              | 1        | Opcode                                |
|                                                                                              |        | 2       | Opcode Address + 1          | 1        | Address of Operand (high order byte)  |
|                                                                                              |        | 3       | Opcode Address + 2          | 1        | Address of Operand (low order byte)   |
|                                                                                              |        | 4       | Address of Operand          | 1        | Current Operand Data                  |
|                                                                                              |        | 5       | Address Bus FFFF            | 1        | Low Byte of Restart Vector            |
|                                                                                              |        | 6       | Address of Operand          | 0        | New Operand Data                      |
| CPX<br>SUBD<br><br>ADDD                                                                      | 6      | 1       | Opcode Address              | 1        | Opcode                                |
|                                                                                              |        | 2       | Opcode Address + 1          | 1        | Operand Address (high order byte)     |
|                                                                                              |        | 3       | Opcode Address + 2          | 1        | Operand Address (low order byte)      |
|                                                                                              |        | 4       | Operand Address             | 1        | Operand Data (high order byte)        |
|                                                                                              |        | 5       | Operand Address + 1         | 1        | Operand Data (low order byte)         |
|                                                                                              |        | 6       | Address Bus FFFF            | 1        | Low Byte of Restart Vector            |

\* TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = SFFF.

Table 14 : Cycle-by-cycle Operation (sheet 2 of 5)

EXTENDED (continued)

| Address Mode and Instructions | Cycles | Cycle # | Address Bus                 | R/W Line | Data Bus                                |
|-------------------------------|--------|---------|-----------------------------|----------|-----------------------------------------|
| JSR                           | 6      | 1       | Opcode Address              | 1        | Opcode                                  |
|                               |        | 2       | Opcode Address + 1          | 1        | Address of Subroutine (high order byte) |
|                               |        | 3       | Opcode Address + 2          | 1        | Address of Subroutine (low order byte)  |
|                               |        | 4       | Subroutine Starting Address | 1        | Opcode of Next Instruction              |
|                               |        | 5       | Stack Pointer               | 0        | Return Address (low order byte)         |
|                               |        | 6       | Stack Pointer – 1           | 0        | Return Address (high order byte)        |

\* TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = \$FFFF.

**Table 14** : Cycle-by-cycle Operation (sheet 3 of 5)

## INDEXED

| Address Mode and Instructions |      | Cycles | Cycle # | Address Bus                    | R/W Line | Data Bus                         |
|-------------------------------|------|--------|---------|--------------------------------|----------|----------------------------------|
| JMP                           |      | 3      | 1       | Opcode Address                 | 1        | Opcode                           |
|                               |      |        | 2       | Opcode Address + 1             | 1        | Offset                           |
|                               |      |        | 3       | Address Bus FFFF               | 1        | Low Byte of Restart Vector       |
| ADC                           | EOR  | 4      | 1       | Opcode Address                 | 1        | Opcode                           |
| ADD                           | LDA  |        | 2       | Opcode Address + 1             | 1        | Offset                           |
| AND                           | ORA  |        | 3       | Address Bus FFFF               | 1        | Low Byte of Restart Vector       |
| BIT                           | SBC  |        | 4       | Index Register Plus Offset     | 1        | Operand Data                     |
| CMP                           | SUB  |        |         |                                |          |                                  |
| STA                           |      | 4      | 1       | Opcode Address                 | 1        | Opcode                           |
|                               |      |        | 2       | Opcode Address + 1             | 1        | Offset                           |
|                               |      |        | 3       | Address Bus FFFF               | 1        | Low Byte of Restart Vector       |
|                               |      |        | 4       | Index Register Plus Offset     | 0        | Operand Data                     |
| LDS                           |      | 5      | 1       | Opcode Address                 | 1        | Opcode                           |
| LDX                           |      |        | 2       | Opcode Address + 1             | 1        | Offset                           |
| LDD                           |      |        | 3       | Address Bus FFFF               | 1        | Low Byte of Restart Vector       |
|                               |      |        | 4       | Index Register Plus Offset     | 1        | Operand Data (high order byte)   |
|                               |      |        | 5       | Index Register Plus Offset + 1 | 1        | Operand Data (low order byte)    |
| STS                           |      | 5      | 1       | Opcode Address                 | 1        | Opcode                           |
| STX                           |      |        | 2       | Opcode Address + 1             | 1        | Offset                           |
| STD                           |      |        | 3       | Address Bus FFFF               | 1        | Low Byte of Restart Vector       |
|                               |      |        | 4       | Index Register Plus Offset     | 0        | Operand Data (high order byte)   |
|                               |      |        | 5       | Index Register Plus Offset + 1 | 0        | Operand Data (low order byte)    |
| ASL                           | LSR  | 6      | 1       | Opcode Address                 | 1        | Opcode                           |
| ASR                           | NEG  |        | 2       | Opcode Address + 1             | 1        | Offset                           |
| CLR                           | ROL  |        | 3       | Address Bus FFFF               | 1        | Low Byte of Restart Vector       |
| COM                           | ROR  |        | 4       | Index Register Plus Offset     | 1        | Current Operand Data             |
| DEC                           | TST* |        | 5       | Address Bus FFFF               | 1        | Low Byte of Restart Vector       |
| INC                           |      |        | 6       | Index Register Plus Offset     | 0        | New Operand Data                 |
| CPX                           |      | 6      | 1       | Opcode Address                 | 1        | Opcode                           |
| SUBD                          |      |        | 2       | Opcode Address + 1             | 1        | Offset                           |
| ADDD                          |      |        | 3       | Address Bus FFFF               | 1        | Low Byte of Restart Vector       |
|                               |      |        | 4       | Index Register + Offset        | 1        | Operand Data (high order byte)   |
|                               |      |        | 5       | Index Register + Offset + 1    | 1        | Operand Data (low order byte)    |
|                               |      |        | 6       | Address Bus FFFF               | 1        | Low Byte of Restart Vector       |
| JSR                           |      | 6      | 1       | Opcode Address                 | 1        | Opcode                           |
|                               |      |        | 2       | Opcode Address + 1             | 1        | Offset                           |
|                               |      |        | 3       | Address Bus FFFF               | 1        | Low Byte of Restart Vector       |
|                               |      |        | 4       | Index Register + Offset        | 1        | First Subroutine Opcode          |
|                               |      |        | 5       | Stack Pointer                  | 0        | Return Address (low order byte)  |
|                               |      |        | 6       | Stack Pointer – 1              | 0        | Return Address (high order byte) |

\* TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = \$FFFF.

**Table 14** : Cycle-by-cycle Operation (sheet 4 of 5)

## INHERENT

| Address Mode and Instructions                                                                                                   | Cycles | Cycle #               | Address Bus                                                                                     | R/W Line              | Data Bus                                                                                                            |
|---------------------------------------------------------------------------------------------------------------------------------|--------|-----------------------|-------------------------------------------------------------------------------------------------|-----------------------|---------------------------------------------------------------------------------------------------------------------|
| ABA DAA SEC<br>ASL DEC SEI<br>ASR INC SEV<br>CBA LSR TAB<br>CLC NEG TAP<br>CLI NOP TBA<br>CLR ROL TPA<br>CLV ROR TST<br>COM SBA | 2      | 1<br>2                | Opcode Address<br>Opcode Address + 1                                                            | 1<br>1                | Opcode<br>Opcode of Next Instruction                                                                                |
| ABX                                                                                                                             | 3      | 1<br>2<br>3           | Opcode Address<br>Opcode Address + 1<br>Address Bus FFFF                                        | 1<br>1<br>1           | Opcode<br>Irrelevant Data<br>Low Byte of Restart Vector                                                             |
| ASLD<br>LSRD                                                                                                                    | 3      | 1<br>2<br>3           | Opcode Address<br>Opcode Address + 1<br>Address Bus FFFF                                        | 1<br>1<br>1           | Opcode<br>Irrelevant Data<br>Low Byte of Restart Vector                                                             |
| DES<br>INS                                                                                                                      | 3      | 1<br>2<br>3           | Opcode Address<br>Opcode Address + 1<br>Previous Stack Pointer Contents                         | 1<br>1<br>1           | Opcode<br>Opcode of Next Instruction<br>Irrelevant Data                                                             |
| INX<br>DEX                                                                                                                      | 3      | 1<br>2<br>3           | Opcode Address<br>Opcode Address + 1<br>Address Bus FFFF                                        | 1<br>1<br>1           | Opcode<br>Opcode of Next Instruction<br>Low Byte of Restart Vector                                                  |
| PSHA<br>PSHB                                                                                                                    | 3      | 1<br>2<br>3           | Opcode Address<br>Opcode Address + 1<br>Stack Pointer                                           | 1<br>1<br>0           | Opcode<br>Opcode of Next Instruction<br>Accumulator Data                                                            |
| TSX                                                                                                                             | 3      | 1<br>2<br>3           | Opcode Address<br>Opcode Address + 1<br>Stack Pointer                                           | 1<br>1<br>1           | Opcode<br>Opcode of Next Instruction<br>Irrelevant Data                                                             |
| TXS                                                                                                                             | 3      | 1<br>2<br>3           | Opcode Address<br>Opcode Address + 1<br>Address Bus FFFF                                        | 1<br>1<br>1           | Opcode<br>Opcode of Next Instruction<br>Low Byte of Restart Vector                                                  |
| PULA<br>PULB                                                                                                                    | 4      | 1<br>2<br>3<br>4      | Opcode Address<br>Opcode Address + 1<br>Stack Pointer<br>Stack Pointer + 1                      | 1<br>1<br>1<br>1      | Opcode<br>Opcode of Next Instruction<br>Irrelevant Data<br>Operand Data from Stack                                  |
| PSHX                                                                                                                            | 4      | 1<br>2<br>3<br>4      | Opcode Address<br>Opcode Address + 1<br>Stack Pointer<br>Stack Pointer - 1                      | 1<br>1<br>0<br>0      | Opcode<br>Irrelevant Data<br>Index Register (low order byte)<br>Index Register (high order byte)                    |
| PULX                                                                                                                            | 5      | 1<br>2<br>3<br>4<br>5 | Opcode Address<br>Opcode Address + 1<br>Stack Pointer<br>Stack Pointer + 1<br>Stack Pointer + 2 | 1<br>1<br>1<br>1<br>1 | Opcode<br>Irrelevant Data<br>Irrelevant Data<br>Index Register (low order byte)<br>Index Register (high order byte) |

**Table 14** : Cycle-by-cycle Operation (sheet 4 of 5)

INHERENT (continued)

| Address Mode and Instructions | Cycles | Cycle # | Address Bus        | R/W Line | Data Bus                                      |
|-------------------------------|--------|---------|--------------------|----------|-----------------------------------------------|
| RTS                           | 5      | 1       | Opcode Address     | 1        | Opcode                                        |
|                               |        | 2       | Opcode Address + 1 | 1        | Irrelevant Data                               |
|                               |        | 3       | Stack Pointer      | 1        | Irrelevant Data                               |
|                               |        | 4       | Stack Pointer + 1  | 1        | Address of Next Instruction (high order byte) |
|                               |        | 5       | Stack Pointer + 2  | 1        | Address of Next Instruction (low order byte)  |
| WAI                           | 9      | 1       | Opcode Address     | 1        | Opcode                                        |
|                               |        | 2       | Opcode Address + 1 | 1        | Opcode of Next Instruction                    |
|                               |        | 3       | Stack Pointer      | 0        | Return Address (low order byte)               |
|                               |        | 4       | Stack Pointer - 1  | 0        | Return Address (high order byte)              |
|                               |        | 5       | Stack Pointer - 2  | 0        | Index Register (low order byte)               |
|                               |        | 6       | Stack Pointer - 3  | 0        | Index Register (high order byte)              |
|                               |        | 7       | Stack Pointer - 4  | 0        | Contents of Accumulator A                     |
|                               |        | 8       | Stack Pointer - 5  | 0        | Contents of Accumulator B                     |
|                               |        | 9       | Stack Pointer - 6  | 0        | Contents of Condition Code Register           |

**Table 14** : Cycle-by-cycle Operation (sheet 5 of 5)

INHERENT

| Address Mode and Instructions | Cycles | Cycle # | Address Bus        | R/W Line | Data Bus                                              |
|-------------------------------|--------|---------|--------------------|----------|-------------------------------------------------------|
| MUL                           | 10     | 1       | Opcode Address     | 1        | Opcode                                                |
|                               |        | 2       | Opcode Address + 1 | 1        | Irrelevant Data                                       |
|                               |        | 3       | Address Bus FFFF   | 1        | Low Byte of Restart Vector                            |
|                               |        | 4       | Address Bus FFFF   | 1        | Low Byte of Restart Vector                            |
|                               |        | 5       | Address Bus FFFF   | 1        | Low Byte of Restart Vector                            |
|                               |        | 6       | Address Bus FFFF   | 1        | Low Byte of Restart Vector                            |
|                               |        | 7       | Address Bus FFFF   | 1        | Low Byte of Restart Vector                            |
|                               |        | 8       | Address Bus FFFF   | 1        | Low Byte of Restart Vector                            |
|                               |        | 9       | Address Bus FFFF   | 1        | Low Byte of Restart Vector                            |
|                               |        | 10      | Address Bus FFFF   | 1        | Low Byte of Restart Vector                            |
| RTI                           | 10     | 1       | Opcode Address     | 1        | Opcode                                                |
|                               |        | 2       | Opcode Address + 1 | 1        | Irrelevant Data                                       |
|                               |        | 3       | Stack Pointer      | 1        | Irrelevant Data                                       |
|                               |        | 4       | Stack Pointer + 1  | 1        | Contents of Condition Code Register from Stack        |
|                               |        | 5       | Stack Pointer + 2  | 1        | Contents of Accumulator B from Stack                  |
|                               |        | 6       | Stack Pointer + 3  | 1        | Contents of Accumulator A from Stack                  |
|                               |        | 7       | Stack Pointer + 4  | 1        | Index Register from Stack (high order byte)           |
|                               |        | 8       | Stack Pointer + 5  | 1        | Index Register from Stack (low order byte)            |
|                               |        | 9       | Stack Pointer + 6  | 1        | Next Instruction Address from Stack (high order byte) |
|                               |        | 10      | Stack Pointer + 7  | 1        | Next Instruction Address from Stack (low order byte)  |

**Table 14** : Cycle-by-cycle Operation (sheet 5 of 5)

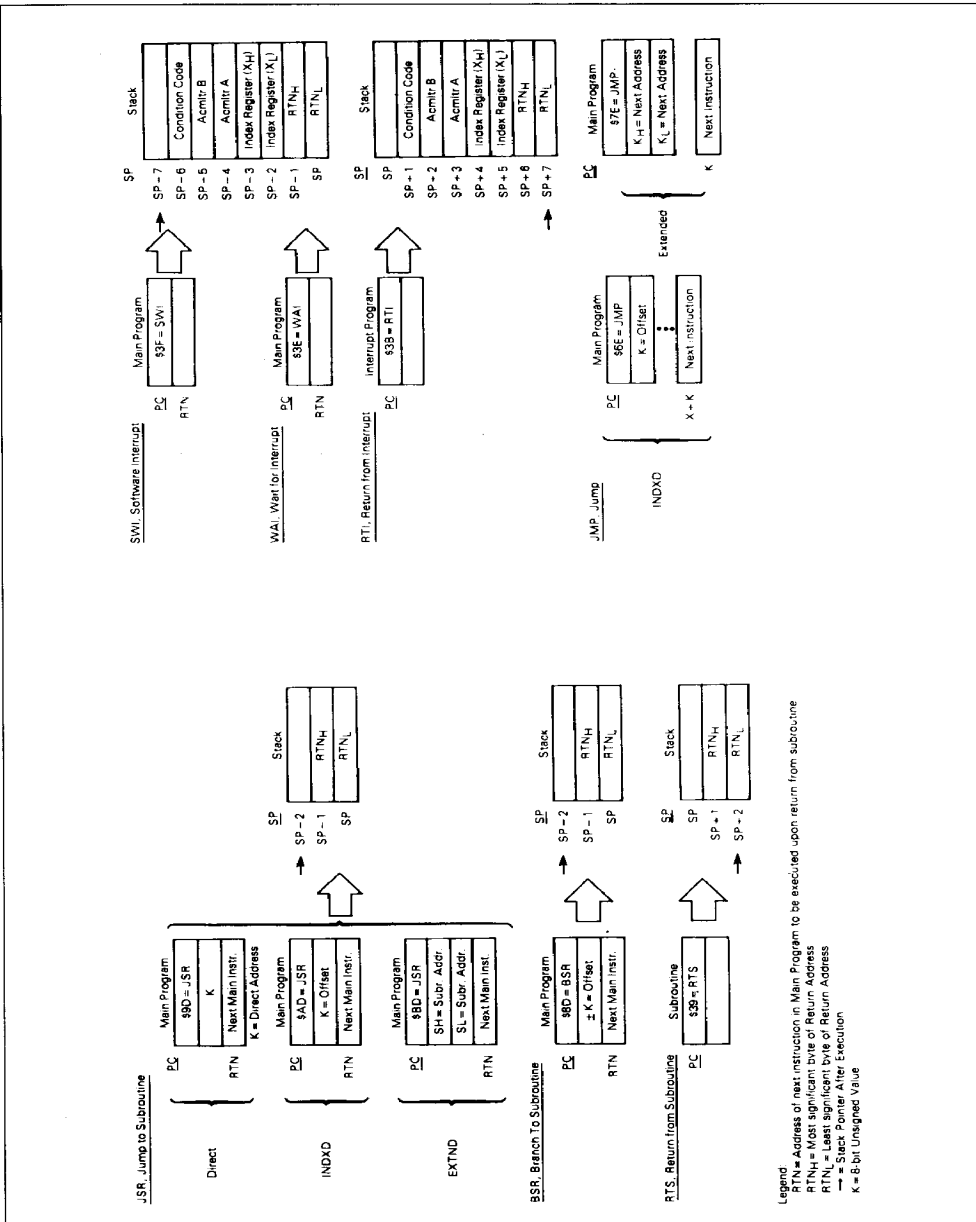
INHERENT (continued)

| Address Mode and Instructions | Cycles | Cycle # | Address Bus               | R/W Line | Data Bus                                |
|-------------------------------|--------|---------|---------------------------|----------|-----------------------------------------|
| SWI                           | 12     | 1       | Opcode Address            | 1        | Opcode                                  |
|                               |        | 2       | Opcode Address + 1        | 1        | Irrelevant Data                         |
|                               |        | 3       | Stack Pointer             | 0        | Return Address (low order byte)         |
|                               |        | 4       | Stack Pointer - 1         | 0        | Return Address (high order byte)        |
|                               |        | 5       | Stack Pointer - 2         | 0        | Index Register (low order byte)         |
|                               |        | 6       | Stack Pointer - 3         | 0        | Index Register (high order byte)        |
|                               |        | 7       | Stack Pointer - 4         | 0        | Contents of Accumulator A               |
|                               |        | 8       | Stack Pointer - 5         | 0        | Contents of Accumulator B               |
|                               |        | 9       | Stack Pointer - 6         | 0        | Contents of Condition Code Register     |
|                               |        | 10      | Stack Pointer - 7         | 1        | Irrelevant Data                         |
|                               |        | 11      | Vector Address FFFA (hex) | 1        | Address of Subroutine (high order byte) |
|                               |        | 12      | Vector Address FFFB (hex) | 1        | Address of Subroutine (low order byte)  |

RELATIVE

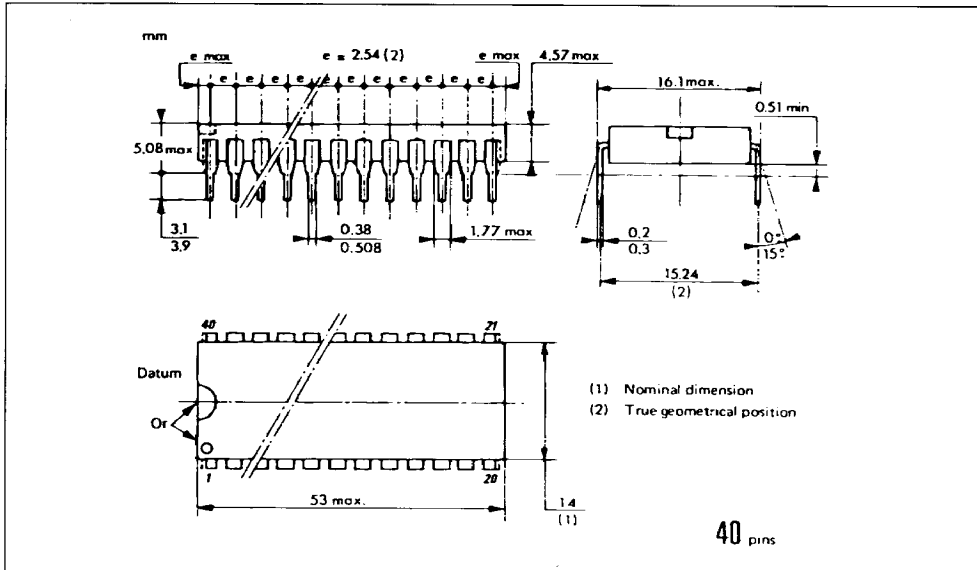
| Address Mode and Instructions                                                       | Cycles | Cycle # | Address Bus                 | R/W Line | Data Bus                         |
|-------------------------------------------------------------------------------------|--------|---------|-----------------------------|----------|----------------------------------|
| BCC BHT BNE BLO<br>BCS BLE BPL BHS<br>BEQ BLS BRA BRN<br>BGE BLT BVC<br>BGT BMI BVS | 3      | 1       | Opcode Address              | 1        | Opcode                           |
|                                                                                     |        | 2       | Opcode Address + 1          | 1        | Branch Offset                    |
|                                                                                     |        | 3       | Address Bus FFFF            | 1        | Low Byte of Restart Vector       |
| BSR                                                                                 | 6      | 1       | Opcode Address              | 1        | Opcode                           |
|                                                                                     |        | 2       | Opcode Address + 1          | 1        | Branch Offset                    |
|                                                                                     |        | 3       | Address Bus FFFF            | 1        | Low Byte of Restart Vector       |
|                                                                                     |        | 4       | Subroutine Starting Address | 1        | Opcode of Next Instruction       |
|                                                                                     |        | 5       | Stack Pointer               | 0        | Return Address (low order byte)  |
|                                                                                     |        | 6       | Stack Pointer - 1           | 0        | Return Address (high order byte) |

Figure 25 : Special Operations.

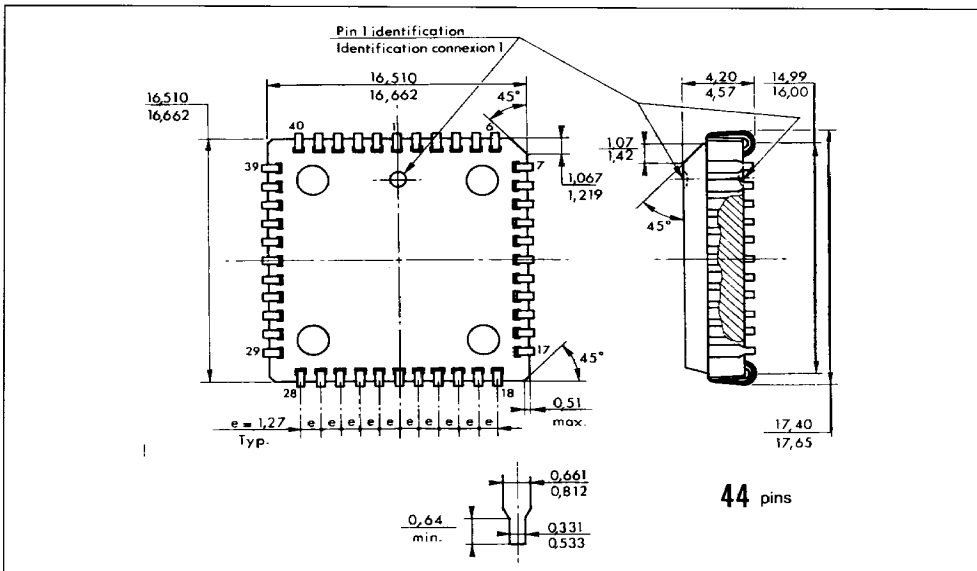


## PACKAGE MECHANICAL DATA

## P SUFFIX - PLASTIC PACKAGE



## FN SUFFIX - PLASTIC PACKAGE



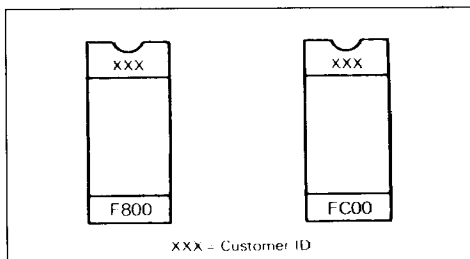
## ORDERING INFORMATION

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

Two 2708 or one 2716 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM must be clearly marked to indicate which EPROM corresponds to which address space. The recommended marking procedure is illustrated below :



After the EPROM (s) are marked, they 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 charge 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 6801 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.

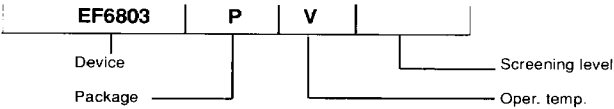
EFDOS is SGS THOMSON Disk Operating System available on development systems such as DEVICE, ...

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       | J | P | E | FN | L*         | V | M | Std             | D | G/B | B/B |
| 1.0MHz                                  | EF6801/03   |         |   | ● |   | ●  | ●          |   |   | ●               |   |     |     |
|                                         |             |         |   | ● |   |    |            | ● |   | ●               |   |     |     |
|                                         | EF6803      |         |   |   |   |    |            | ● |   | ●               |   |     |     |
| 1.25MHz                                 | EF6801/03-1 |         |   | ● |   | ●  | ●          |   |   | ●               |   |     |     |
|                                         |             |         |   | ● |   |    |            | ● |   | ●               |   |     |     |
|                                         | EF6803-1    |         |   |   |   |    |            | ● |   | ●               |   |     |     |
| 1.5MHz                                  | EF68A01/03  |         |   | ● |   |    | ●          |   |   | ●               |   |     |     |
|                                         | EF68A03     |         |   |   |   |    |            | ● |   | ●               |   |     |     |
| 2.0MHz                                  | EF68B01/03  |         |   | ● |   |    | ●          |   |   | ●               |   |     |     |
| Examples : EF6801P, EF6801FN, EF6801PV. |             |         |   |   |   |    |            |   |   |                 |   |     |     |

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, M : -55°C to +125°C, \* : may be omitted.  
Screening level : Std : (no-end suffix), D : NFC 96883 level D,  
G/B : NFC 96883 level G, B/B : NFC 96883 level B and MIL-STD-883C level B.

EXORciser is a registered trade mark of MOTOROLA Inc.

## EF6801 FAMILY - MCU CUSTOMER ORDERING SHEET

Commercial reference : 

Customer name : .....

Company : .....

Address : .....

Phone : .....

Customer's marking : 

Application : .....

Specification reference ;

☐ SGS-THOMSON Microelectronics reference☐ Special customer data reference\*ROM capacity required :  bytesNumber of interrupt vector : 

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 developed by :

☐ SGS-THOMSON Microelectronics 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

-Internal max. clock frequency :

☐ 1.0MHz☐ 1.25MHz☐ 1.5MHz☐ 2.0MHz

\* Requires prior factory approval

Yearly quantity forecast :

• start of production date :

• for a shipment period of :

CUSTOMER CONTACT NAME :

DATE :

SIGNATURE :