SGS-THOMSON
MICROELECTRONICS

EF6804J2

8 BIT MICROCOMPUTER

HARDWARE FEATURES

- 5-VOLT SINGLE SUPPLY
- 32 BYTES OF RAM
- MEMORY MAPPED I/O
- 1012 BYTES OF PROGRAM ROM
- 64 BYTES OF DATA ROM
- 12 BIDIRECTIONAL I/O LINES (eight lines with high current sink capability)
- ON-CHIP CLOCK GENERATOR
- SELF-TEST MODE
- MASTER RESET
- COMPLETE DEVELOPMENT SYSTEM SUPPORT ON INICE ®
- SOFTWARE PROGRAMMABLE 8-BIT TIMER CONTROL REGISTER AND TIMER PRESCALER (7 bits, 2^n)
- TIMER PIN IS PROGRAMMABLE AS INPUT OR OUTPUT
- ON-CHIP CIRCUIT FOR ROM VERIFY

SOFTWARE FEATURES

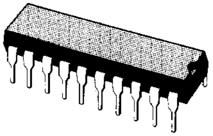
- SOFTWARE FEATURES
- SIMILAR TO EF6805 HMOS FAMILY
- BYTE EFFICIENT INSTRUCTION SET
- EASY TO PROGRAM
- TRUE BIT MANIPULATION
- BIT TEST AND BRANCH INSTRUCTION
- SEPARATE FLAGS FOR INTERRUPT AND NORMAL PROCESSING
- VERSATILE INDIRECT REGISTERS
- CONDITIONAL BRANCHES
- SINGLE INSTRUCTION MEMORY EXAMINE/CHANGE
- TRUE LIFO STACK ELIMINATES STACK POINTER
- NINE POWERFUL ADDRESSING MODES
- ANY BIT IN DATA SPACE MEMORY MAY BE TESTED
- ANY BIT IN DATA SPACE MEMORY CAPABLE OF BEING WRITTEN TO MAY BE SET OR CLEARED

USER SELECTABLE OPTIONS

- 12 BIDIRECTIONAL I/O LINES WITH LSTTL, LSTTL/CMOS, OR OPEN-DRAIN INTERFACE
- CRYSTAL OR LOW-COST RESISTOR-CAPACITOR OSCILLATOR
- MASK SELECTABLE EDGE- OR LEVEL- SENSITIVE INTERRUPT PIN

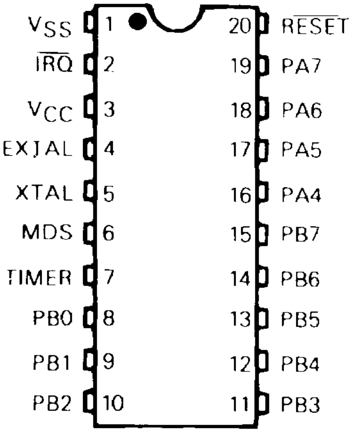
DESCRIPTION

The EF6804J2 microcomputer unit (MCU) is a member of the EF6804 Family of very low-cost single-chip microcomputers. This 8-bit microcomputer contains a CPU, on-chip CLOCK, ROM, RAM, I/O, and TIMER. It is designed for the user who needs an economical microcomputer with the proven capabilities of the EF6800-based instruction set.



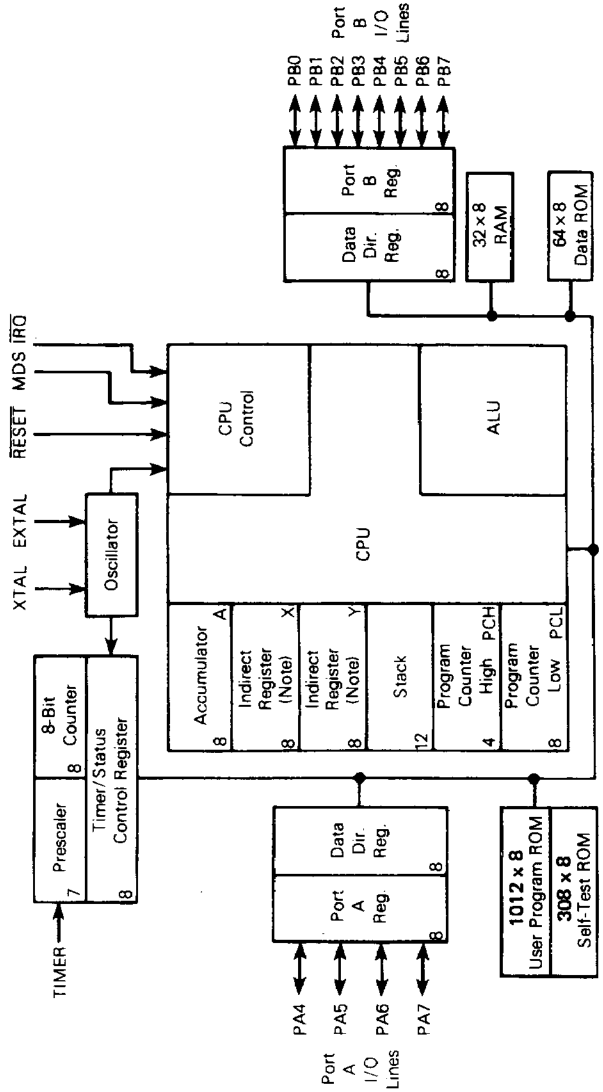
P
(Plastic Package)

PIN CONNECTIONS



INICE ® is SGS-THOMSON development/emulation tool.

Figure 1.1 : EF6804J2 MCU Block Diagram.



Note : 8-bit indirect registers X and Y , although shown as part of the CPU, are actually located in the 32 x 8 RAM at locations $80 and $81.

SECTION 2

FUNCTIONAL PIN DESCRIPTION, MEMORY, CPU, AND REGISTERS

This section provides a description of the functional pins, memory spaces, the central processing unit (CPU), and the various registers and flags.

2.1. FUNCTIONAL PIN DESCRIPTION

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

2.1.2. $\overline{IRQ}$. This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to **4.1. INTERRUPT** for additional information.

2.1.3. XTAL AND EXTAL. These pins provide connections to the on-chip clock oscillator circuit. A crystal, a resistor and capacitor, or an external signal, depending on the user selectable manufacturing mask option, can be connected to these pins to provide a system clock source with various stability/cost tradeoffs. Lead lengths and stray capacitance on these two pins should be minimized. Refer to 4.4. Internal Clock Generator Options for recommendations concerning these inputs.

2.1.4. TIMER. In the input mode, the timer pin is connected to the prescaler input and serves as the timer clock. In the output mode, the timer pin signals that a time out of the timer has occurred. Refer to Section 3 Timer for additional information.

2.1.5. $\overline{RESET}$. The $\overline{RESET}$ pin is used to restart the processor of the EF6804J2 to the beginning of a program. This pin, together with the MDS pin, is also used to select the operating mode of the EF6804J2. If the MDS pin is at zero volts, the normal mode is selected and the program counter is loaded with the user restart vector. However, if the MDS pin is at + 5 volts, then pins PA6 ad PA7 are decoded to allow selection of the operating mode. Refer to 4.3. Reset for additional information.

2.1.6. MDS. The MDS (mode select) pin is used to place the MCU into special operating modes. If MDS is held at + 5 volts at the exit of the reset state, the decoded state of PA6 and PA7 is latched to determine the operating mode (single-chip, self-test, or ROM verify). However, if MDS is held at zero volts at the exit of the reset state, the single-chip operating mode is automatically selected (regardless of PA6 and PA7 state).

For those users familiar with the EF6801 microcomputer, mode selection is similar but much less complex in the EF6804J2. No special external diodes, switches, transistors, etc. are required in the EF6804J2.

2.1.7. INPUT/OUTPUT LINES (PA4-PA7, PB0-PB7). These 12 lines are arranged into one 4-bit port (A) and one 8-bit port (B). All lines are programmable as either inputs or outputs under software control of the data direction registers. Refer to Section 5 Input/output Ports for additional information.

2.2. MEMORY

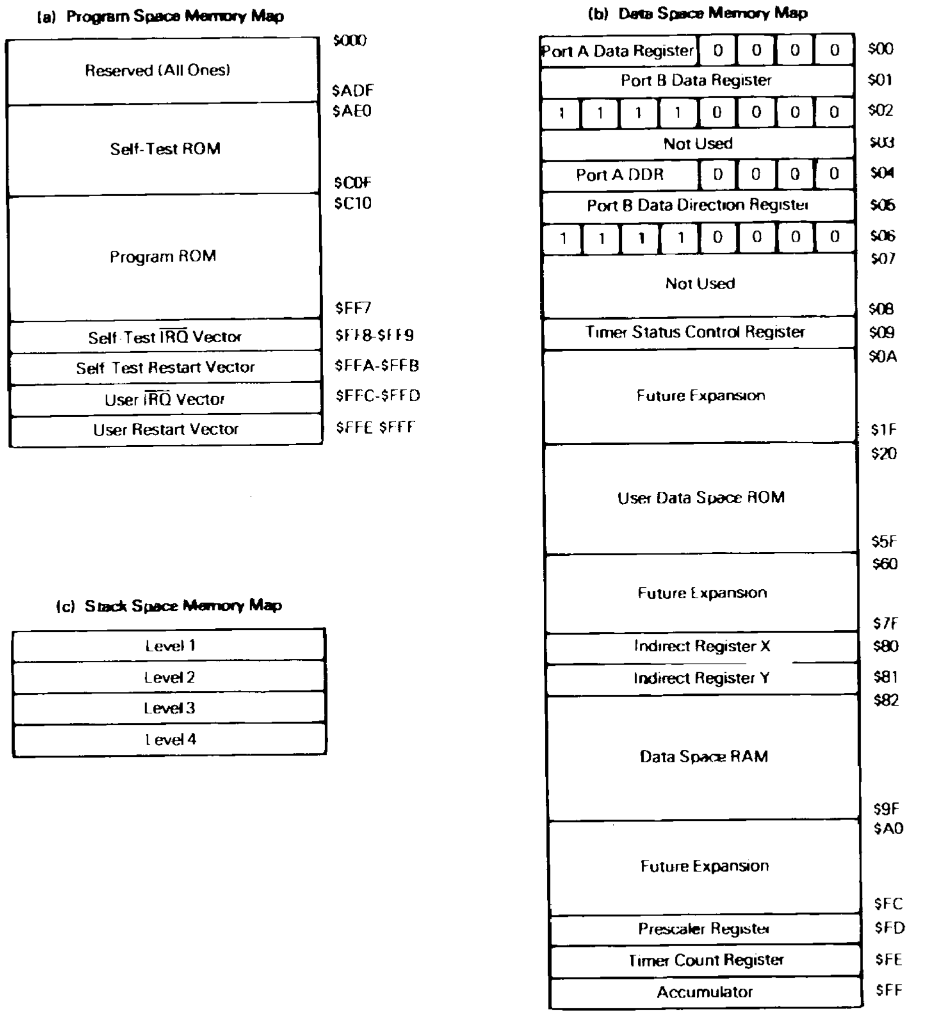
The MCU operates in three different memory spaces : program space, data space, and stack space. A representation of these memory spaces is shown in figure 2.1. The program space (figure 2.1a) contains all of the instructions that are to be executed, as well as the data required for the immediate addressing mode instructions, and the self-test and user vectors. The data space (figure 2.1b) contains all of the RAM locations, plus I/O locations and some ROM used for storage of tables ad constants. The stack space (figure 2.1c) contains RAM which is used for stacking subroutine and interrupt return addresses.

The MCU is capable of addressing 4096 bytes of program space memory with its program counter and 256 bytes of data space memory with its instructions. The data space memory contains three bytes for port data registers, three bytes for port data direction registers, one byte for timer status/control, 64 bytes ROM, 32 bytes RAM (which includes two bytes for X and Y indirect registers), two bytes for timer prescaler and count registers, and one byte for the accumulator. The program space section includes 304 bytes of self-test ROM, 1008 bytes program ROM, and eight bytes of vectors for self-test and user programs.

2.3. CENTRAL PROCESSING UNIT

The PCU of the EF6804 Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal addresses, data, and control buses.

Figure 2.1 : EF6804J2 MCU Address Map.



(a) Program Space Memory Map

Reserved (All Ones)	$000 $ADF
Self-Test ROM	$AE0 $C0F
Program ROM	$C10 $FF7
Self-Test $\overline{IRQ}$ Vector	$FF8-$FF9
Self Test Restart Vector	$FFA-$FFB
User $\overline{IRQ}$ Vector	$FFC-$FFD
User Restart Vector	$FFE $FFF

(b) Data Space Memory Map

Port A Data Register				0	0	0	0	$00
Port B Data Register								$01
1	1	1	1	0	0	0	0	$02
Not Used								$03
Port A DDR				0	0	0	0	$04
Port B Data Direction Register								$05
1	1	1	1	0	0	0	0	$06
Not Used								$07 $08
Timer Status Control Register								$09
Future Expansion								$0A $1F
User Data Space ROM								$20 $5F
Future Expansion								$60 $7F
Indirect Register X								$80
Indirect Register Y								$81
Data Space RAM								$82 $9F
Future Expansion								$A0 $FC
Prescaler Register								$FD
Timer Count Register								$FE
Accumulator								$FF

(c) Stack Space Memory Map

Level 1
Level 2
Level 3
Level 4

2.4. REGISTERS

The EF6804 Family CPU has four registers and two flags available to the programmer. They are shown in figure 2.2. and are explained in the following paragraphs.

2.4.1. ACCUMULATOR (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator is implemented as the highest RAM location ($FF) in data space and thus implies that several instructions exist which are not explicitly implemented. Refer to 6.3. Implied Instructions for additional information.

2.4.2. INDIRECT REGISTERS (X, Y). These two indirect registers are used to maintain pointers to other memory locations in data space. They are used in the register-indirect addressing mode, and can be accessed with the direct, indirect, short direct, or bit set/clear addressing modes. These registers are implemented as two of the 32 RAM locations ($80, $81) and as such generate implied instructions and may be manipulated in a manner similar to any RAM memory location in data space. Refer to 6.3. Implied Instructions for additional information.

2.4.3. PROGRAM COUNTER (PC). The program counter is a 12-bit register that contains the address of the next ROM word to be used (may be opcode, operand, or address of operand). The 12-bit program counter is contained in PCL (low byte) and PCH (high nibble).

2.4.4. FLAGS (C, Z). The carry (C) bit is set on a carry or a borrow out of the ALU. It is cleared if the result of an arithmetic operation does not result in a carry or a borrow. The (C) bit is also set to the value of the bit tested in a bit test instruction, and participates in the rotate left instruction.

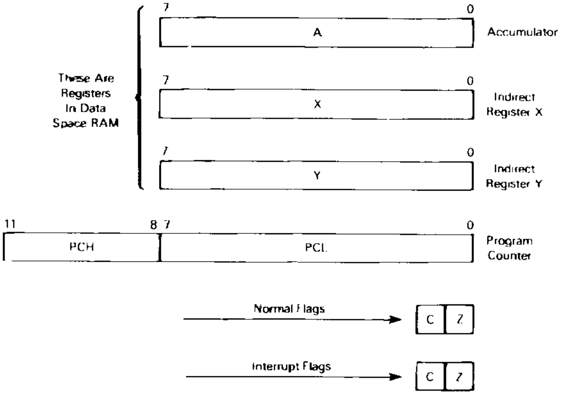
The zero (Z) bit is set if the result of the last arithmetic or logical operation was equal to zero, otherwise it is cleared.

There are two sets of these flags, one set is for interrupt processing, the other for all other routines. When an interrupt occurs, a context switch is made from the program flags to the interrupt flags (interrupt mode). An RTI forces the context switch back to the program flags (program mode). While in either mode only the flags for that mode are available. Further, the interrupt flags will not be cleared upon entering the interrupt mode. Instead, the flags will be as they were at the exit of the last interrupt mode. Both sets of flags are cleared by reset.

2.4.5. STACK. There is a true LIFO stack incorporated in the EF6804J2 which eliminates the need for a stack pointer. Stack space is implemented in separate RAM (12-bits wide) shown in figure 2.1c. Whenever a subroutine call (or interrupt) occurs, the contents of the PC are shifted into the top register of the stack. At the same time (same cycle), the top register is shifted to the next level deeper. This happens to all registers with the bottom register falling out the bottom of the stack.

Whenever a subroutine or interrupt return occurs, the top register is shifted into the PC and all lower registers are shifted up one level higher. The stack RAM is four levels deep. If the stack is pulled more than four times without any pushes, the address that was stored in the bottom level will be shifted into the PC.

Figure 2.2 : Programming Model.



SECTION 3

TIMER

3.1. INTRODUCTION

A block diagram of the EF6804J2 timer circuitry is shown in figure 3.1. The timer logic in the MCU is comprised of a simple 8-bit counter (timer count register, TCR) with a 7-bit prescaler, and a timer status/control register (TSCR). The timer count register, which may be loaded under program control, is decremented towards zero by a clock input (prescaler output). The prescaler is used to extend the maximum interval of the overall timer. The prescaler tap is selected by bits 0-2 (PS0-PS2) of the timer status/control register. Bits PS0-PS2 control the actual division of the prescaler within the range of divide-by-1 (2^0) to divide-by-128 (2^7). The timer count register (TCR) and prescaler are decremented on rising clock edges. The coding of the TCSR PS0-PS2 bits produce a division in the prescaler as shown in table 3.1.

Table 3.1 : Prescaler Coding Table.

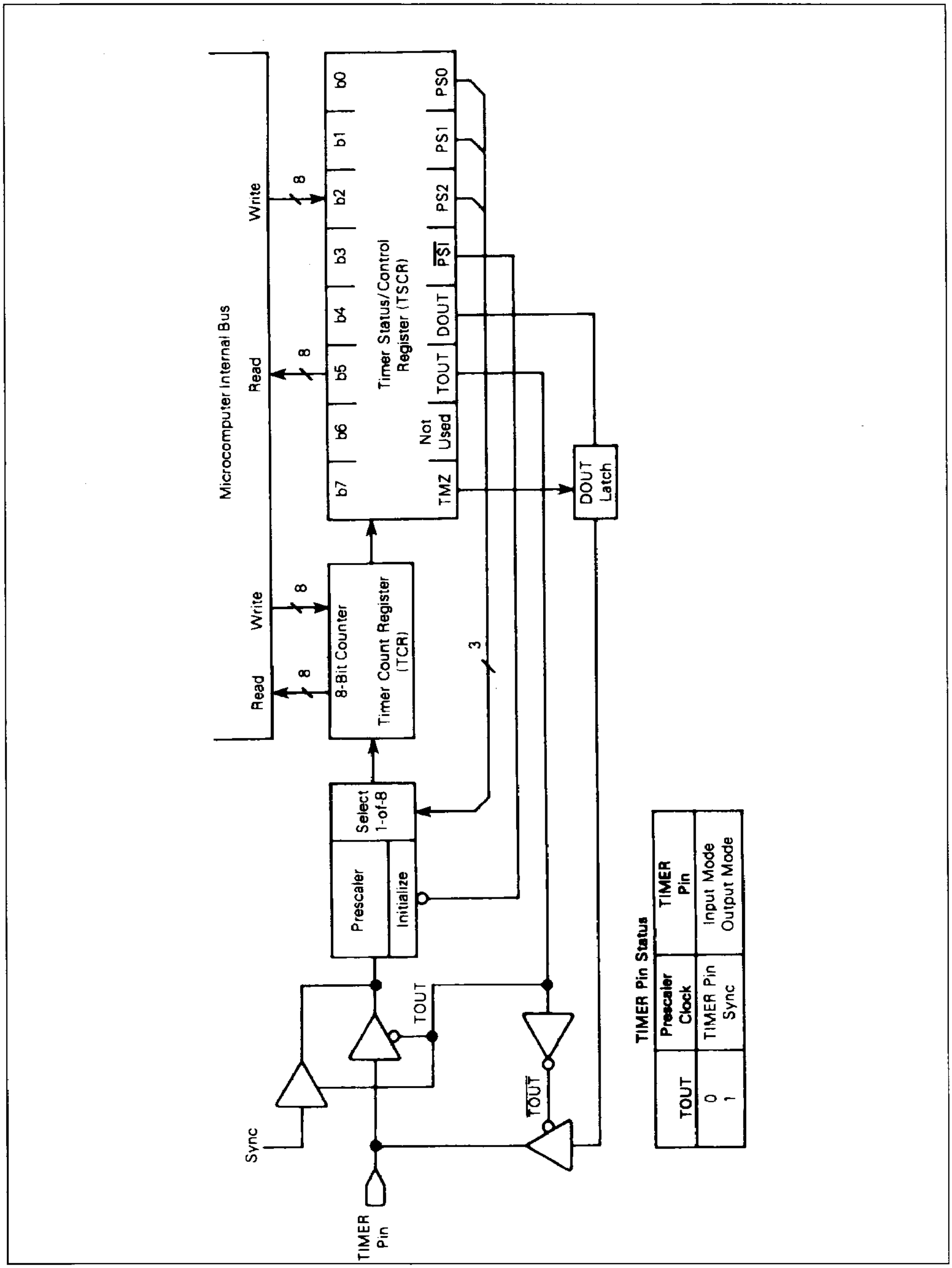
PS2	PS1	PS0	Divide By
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

The TIMER pin may be programmed as either an input or an output depending on the status of TOUT (TSCR bit 5). Refer to figure 3.1. In the input mode, TOUT is a logic zero and the TIMER pin is connected directly to the prescaler input. Therefore, the timer prescaler is clocked by the signal applied from the TIMER pin. The prescaler then divides its clock input by a value determined by the coding of the TSCR bits PS0-PS2 as shown in table 3.1. The divided prescaler output then clocks the 8-bit timer count register (TCR). When the TCR is decremented to zero, it sets the TMZ bit in the timer status/control register (TSCR). The TMZ bit can be tested under program control to perform a timer function whenever it goes high. The frequency of the external clock applied to the TIMER pin must be less than t_{byte} (f_{osc}/48).

In the output mode, TOUT is a logic one and the TIMER pin is connected to the DOUT latch. Therefore, the timer prescaler is clocked by the internal sync pulse (divide-by-48 of the internal oscillator). Operation is similar to that described above for the input mode. However, in the output mode, the low-to-high TMZ bit transition is used to latch the DOUT bit ofhte TSCR and provide it as output of the TIMER pin.

NOTE :
TMZ is normally set to logic one when the timer times out (TCR decrements to $00) ; however, it may be set by a write of $00 to the TCR or by a write to bit 7 of the TSCR.

Figure 3.1 : Timer Block Diagram.



TIMER Pin Status

TOUT	Prescaler Clock	TIMER Pin
0	TIMER Pin	Input Mode
1	Sync	Output Mode

During reset, the timer count register and prescaler are set to $FF, while the timer status/control register is cleared to $00 and the DOUT LATCH (TIMER pin is in the high-impedance input mode) is forced to a logic high. The prescaler and timer count register are implemented in data space RAM locations ($FD, $FE) ; therefore, they are both readable and writeable. A write to either will predominate over the TCR decrement-to-$00 function ; i.e., if a write and a TCR decrement-to-$00 occur simultaneously, the write will take precedence, and the TMZ bit is not set until the next timer time out.

3.2. TIMER REGISTERS

3.2.1. TIMER COUNT REGISTER (TCR). The timer count register indicates the state of the internal 8-bit counter.

7	0
MSB	LSB

TCR addr = $FE

3.2.2. TIMER STATUS/CONTROL REGISTER (TSCR).

7	6	5	4	3	2	1	0
TMZ	Not Used	TOUT	DOUT	PSI	PS2	PS1	PS0

TSCR address = $09

b7, TMZ. Low-to-high transition indicates the timer count register has decremented to zero since the timer status/control register was last read. Cleared by a read of TSCR register if TMZ was read as a logic one.

b6. Not used.

b5, TOUT. When low, this bit selects the input mode for the timer. When high, the output mode is selected.

b4, DOUT. Data sent to the timer output pin when TMZ is set high (output mode only).

b3, $\overline{\text{PSI}}$. Used to initialize the prescaler and inhibit its counting while $\overline{\text{PSI}}$ = 0. The initialized value is set to $FF. The timer count register will also be inhibited (contents unchanged). When PSI = 1 the prescaler begins to count downward.

b0, b1, b2, PS0-PS1-PS2. These bits are used to select the prescaler divide-by ratio ; therefore, effecting the clock input frequency to the timer count register.

3.2.3. TIMER PRESCALER REGISTER. The timer prescaler register indicates the state of the internal 7-bit prescaler. This 7-bit prescaler divide ratio is normally determined by bits PS0-PS2 of the timer status/control register (see table 3.1).

6	0
MSB	LSB

TPR Address = $FD

SECTION 4

INTERRUPT, SELF–TEST, RESET AND INTERNAL CLOCK GENERATOR

4.1. INTERRUPT

The EF6804J2 can be interrupted by applying a logic low signal to the $\overline{IRQ}$ pin ; however, a mask option selected at the time of manufacture determines whether the negative-going edge or the actual low level is sensed to indicate an interrupt.

4.1.1. EDGE-SENSITIVE OPTION. When the $\overline{IRQ}$ pin is pulled low, the internal interrupt request latch is set. Prior to each instruction fetch, the interrupt request latch is tested and, if its output is high, an interrupt sequence is initiated at the end of the current instruction (provided the interrupt mask is cleared). Figure 4.1 contains a flowchart which illustrates both the reset and interrupt sequence. The interrupt sequence consists of one cycle during which : the interrupt request latch is cleared, the interrupt mode flags are selected, the PC is saved on the stack, the interrupt mask is set, and the $\overline{IRQ}$ vector (single chip mode = $FFC/$FFD, self-test mode = $FF8/$FF9) is loaded into the PC. Internal processing of the interrupt continues until an RTI (return from interrupt) instruction is processed. During the RTI instruction, the interrupt mask is cleared and the program mode flags are selected. The next instruction of the program is then fetched and executed. Once the interrupt was initially detected and the interrupt sequence started, the interrupt request latch is cleared so that the next (second) interrupt may be detected even while the previous (first) one is being serviced. However, even though the second interrupt sets the interrupt request latch during processing of the first interrupt, the second interrupt sequence will not be initiated until completion of the interrupt service routine for the first interrupt. Completion of an interrupt service routine is always accomplished using an RTI instruction to return to the main program. The interrupt mask (which is not directly available to the programmer) is cleared during the last cycle of the RTI instruction.

4.1.2. LEVEL-SENSITIVE OPTION. The actual operation of the level-sensitive and edge-sensitive options are similar except that the level-sensitive option does not have an interrupt request latch. With no interrupt request latch, the logic level of the $\overline{IRQ}$ pin is checked for detection of the interrupt. Also, in the interrupt sequence, there is no need to clear the interrupt request latch. These differences are illustrated in the flowchart of figure 4.1.

4.1.3. POWER UP AND TIMING. During the power-up sequence the interrupt mask is set to preclude any false or "ghost" interrupts from occurring. To clear the interrupt mask, the programmer should write a JSR (instead of a JMP) instruction to an initialization routine as the first instruction in a program. The initialization routine should end with an RTI (instead of RTS). Maximum interrupt response time is eight machine (t_{byte}) cycles (see 4.4. Internal Clock Generator Options). This includes five machine cycles for the longest instruction, plus one machine cycle for stacking the PC and switching flags, plus two machine cycles for synchronization of the $\overline{IRQ}$ input with the internal clock. Minimum response time is one machine cycle for stacking PC and switching flags (see 2.4.4. flags (C, Z)).

4.2. SELF-TEST

The EF6804J2 MCU has a unique internal ROM-based off-line self-test capability using signature analysis techniques. A test program stored in the on-chip ROM is initiated by configuring pins PA6 and PA7 during reset. The test results are sampled on a cycle-by-cycle basis by a 16-bit on-chip signature analysis register configured as a linear feedback shift register (LFSR) using the standard CCITT CRC16 polynomial. A schematic diagram of the self-test connections is shown in figure 4.2. To perform a test of the MCU, connect it as shown in figure 4.2a and monitor the LEDs for a 1101 ($D) pattern.

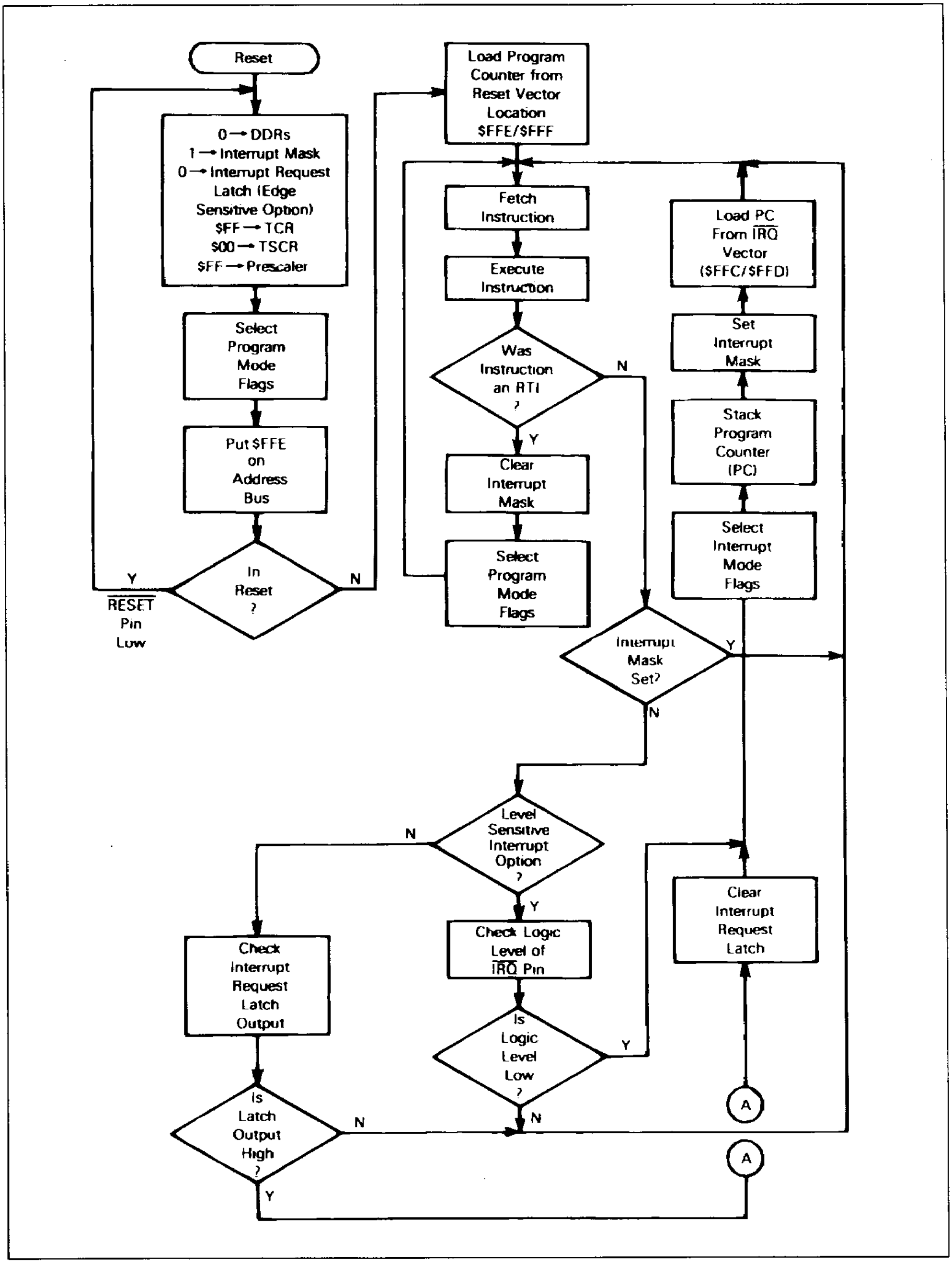
A special ROM self-test utilizing the signature analysis circuitry is also included. To initiate a test of the ROM, connect the circuit as shown in figure 4.2b. This mode also uses the on-chip signature analysis register to verify the contents of the custom ROM by monitoring an internal bus. The "Good" LED indicates that all ROM words have been read and that the result was the correct signature.

The on-chip self-test and the ROM test are the basis of SGS-THOMSON Microelectronics production testing for the EF6804J2. These tests have been fault graded using statistical methods and have been found to provide high fault coverage using automatic test equipment (ATE) or the circuit of figure 4.2.

4.3. RESET

The MCU can be reset in two ways : by initial power up (see figure 4.1) and by the external reset input ($\overline{RESET}$). During power up, a delay of t_{RHL} is needed before allowing the $\overline{RESET}$ input to go high.

Figure 4.1 : Reset and Interrupt Processing Flowchart.


Reset
0 → DDRs
1 → Interrupt Mask
0 → Interrupt Request Latch (Edge Sensitive Option)
$FF → TCR
$00 → TSCR
$FF → Prescaler
Select Program Mode Flags
Put $FFE on Address Bus
In Reset ?
Y
N
RESET Pin Low
Load Program Counter from Reset Vector Location $FFE/$FFF
Fetch Instruction
Execute Instruction
Was Instruction an RTI ?
N
Y
Clear Interrupt Mask
Select Program Mode Flags
Load PC From IRQ Vector ($FFC/$FFD)
Set Interrupt Mask
Stack Program Counter (PC)
Select Interrupt Mode Flags
Interrupt Mask Set?
Y
N
Level Sensitive Interrupt Option ?
N
Y
Check Logic Level of IRQ Pin
Is Logic Level Low ?
Y
N
Clear Interrupt Request Latch
A
Check Interrupt Request Latch Output
Is Latch Output High ?
N
Y
A

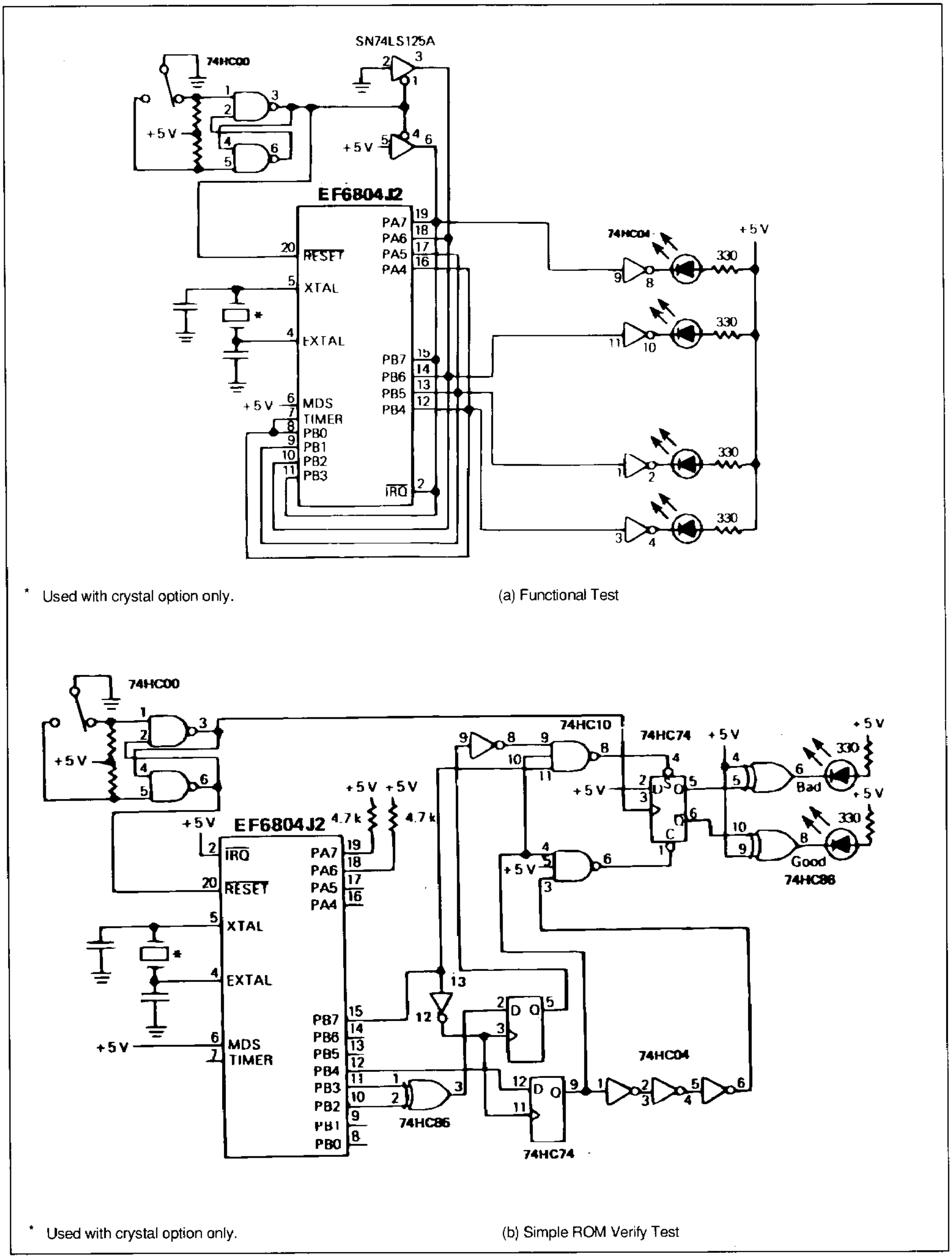
This time delay allows the internal clock generator to stabilize. Connecting a capacitor and resistor to the RESET input, as shown in figure 4.3, typically provides sufficient delay.

4.4. INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor-capacitor, or an external signal may be used to generate a system clock with various stability/cost tradeoffs. A manufacturing mask option is required to select either the crystal oscillator or the RC oscillator circuit. The different clock generator option connection methods are shown in figure 4.4, crystal specifications and suggested PC board layouts are given in figure 4.5, resistor-capacitor selection graph is given in figure 4.6, and a timing diagram is illustrated in figure 4.7. The crystal oscillator startup time is a function of many variables : crystal parameters (especially R_S), oscillator load capacitance (C_L), IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator startup, neither the crystal characteristics nor the load capacitance should exceed recommendations.

The oscillator output frequency is internally divided by four to produce the internal $\phi 1$ and $\phi 2$ clocks. The $\phi 1$ clock is divided by twelve to produce a machine byte (cycle) clock. A byte cycle is the smallest unit needed to execute any operation (i.e., increment the program counter). An instruction may need two, four, or five byte cycles to execute.

Figure 4.2 : Self–test Circuit.



* Used with crystal option only.

(a) Functional Test

* Used with crystal option only.

(b) Simple ROM Verify Test

Figure 4.3 : Power–up Reset Delay Circuit.

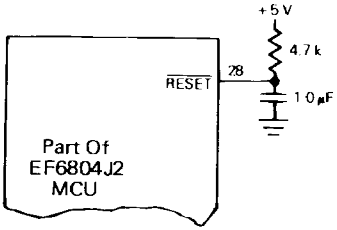


Figure 4.4 : Clock Generator Options.

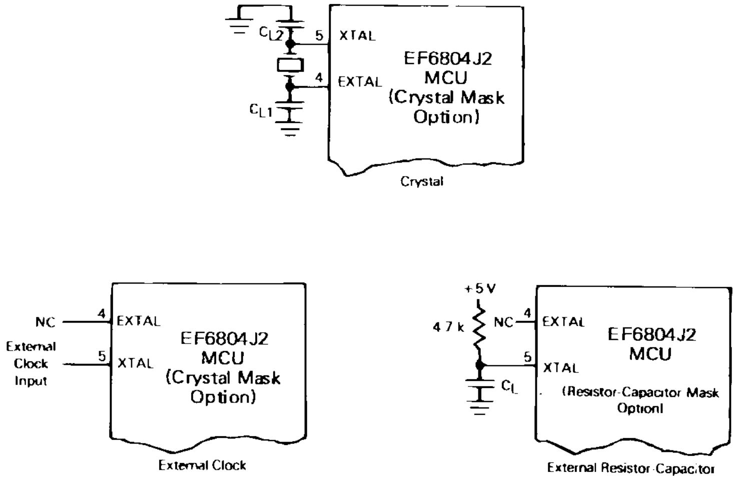
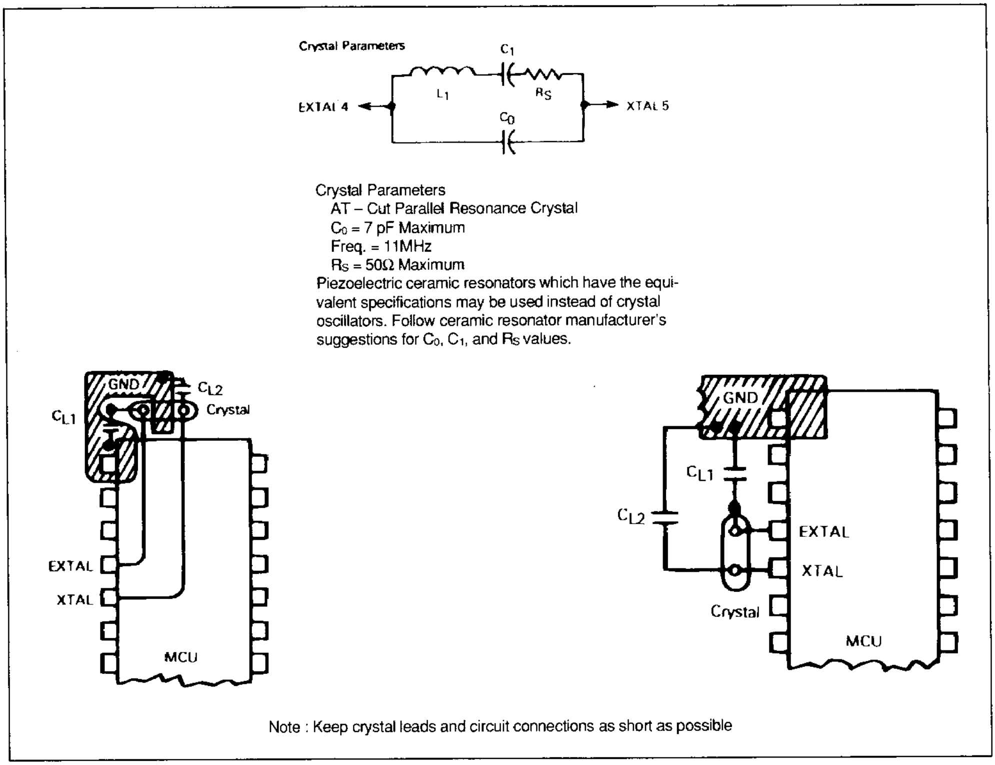


Figure 4.5 : Crystal Motional Arm Parameters and Suggested PC Board Layout.



Crystal Parameters
AT – Cut Parallel Resonance Crystal
C_0 = 7 pF Maximum
Freq. = 11MHz
R_S = 50Ω Maximum
Piezoelectric ceramic resonators which have the equivalent specifications may be used instead of crystal oscillators. Follow ceramic resonator manufacturer's suggestions for C_0, C_1, and R_S values.

Note : Keep crystal leads and circuit connections as short as possible

Figure 4.6 : Typical Frequency Selection for Resistor–capacitor Oscillator Option (C_L = 17pF).

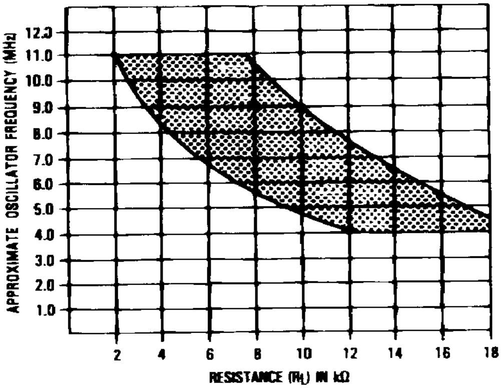
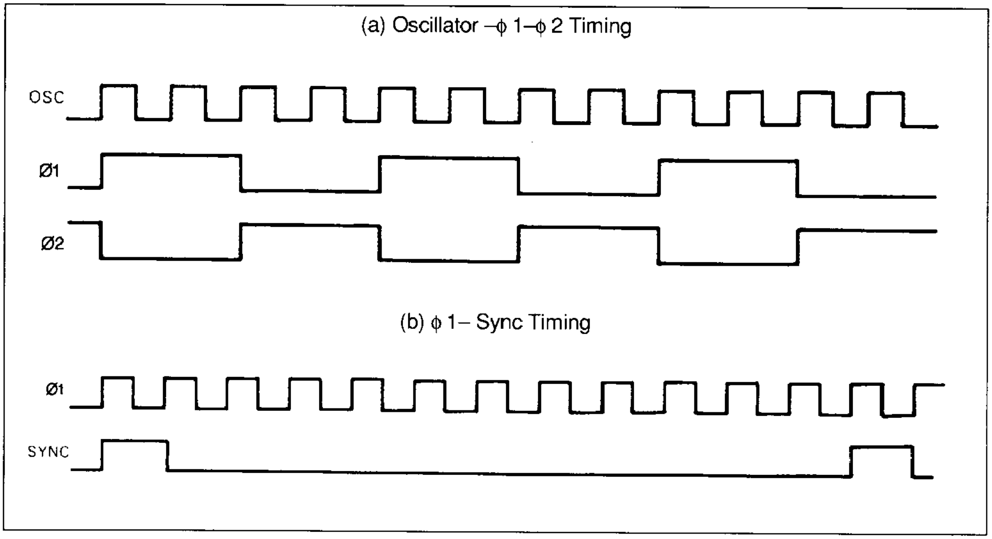


Figure 4.7 : Clock Generator Timing Diagram.



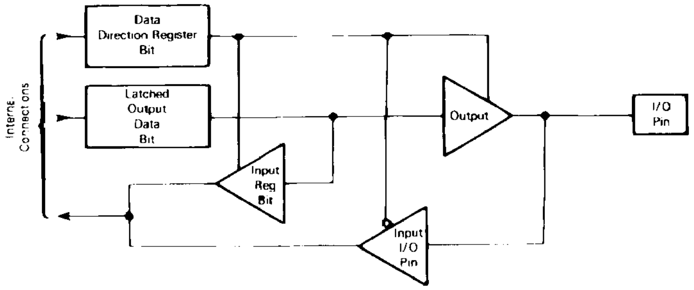
SECTION 5

INPUT/OUTPUT PORTS

5.1. INPUT/OUTPUT

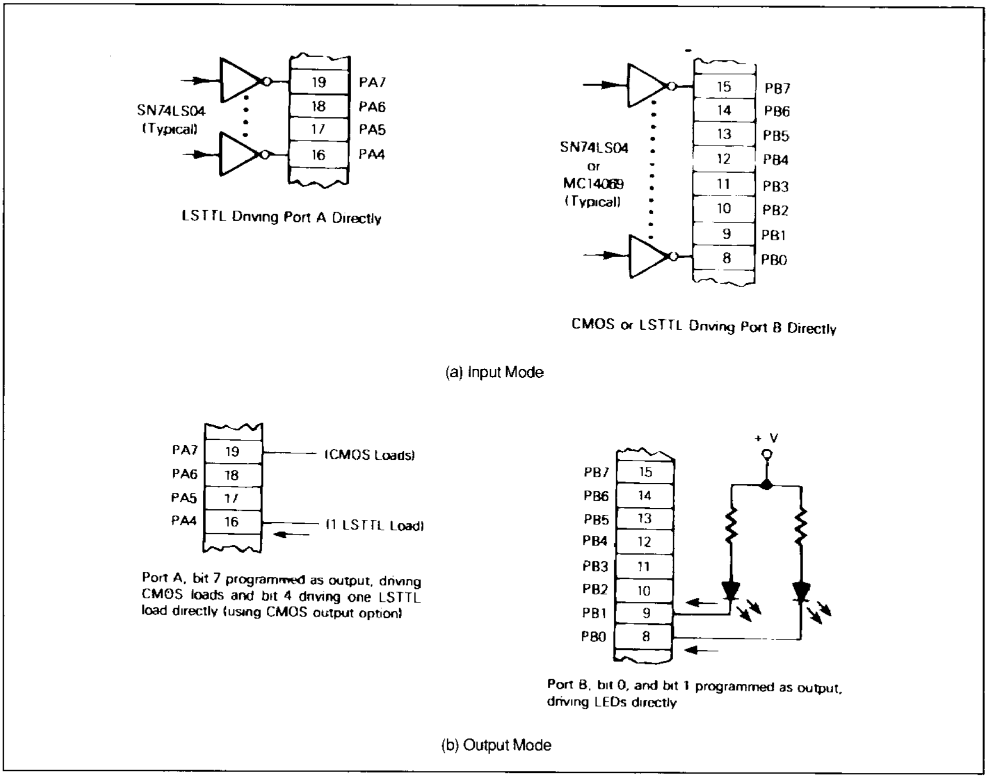
There are 12 input/output pins. All pins (port A and B) are programmable as either inputs or outputs under software control of the corresponding data direction register (DDR). The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic one for output or a logic zero for input. On reset, all the DDRs are initialized to a logic zero state to put the ports in the input mode. The port output registers are not initialized on reset but should be initialized before changing the DDR bits to avoid undefined levels. When programmed as outputs, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading ; see figure 5.1. All input/output pins are LSTTL compatible as both inputs and outputs. In addition, both ports may have one of two mask options : 1) internal pullup resistor for CMOS output compatibility, or 2) open drain output. The address map in figure 2.1 gives the address of data registers and DDRs. The register configuration is discussed under the registers paragraph below and figure 5.2 provides some examples of port connections.

Figure 5.1 : Typical I/O Port Circuitry.



Data Direction Register Bit	Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	X	Hi-Z	Pin

Figure 5.2 : Typical Port Connections.


SN74LS04
(Typical)
19 PA7
18 PA6
17 PA5
16 PA4
LSTTL Driving Port A Directly
SN74LS04
or
MC14069
(Typical)
15 PB7
14 PB6
13 PB5
12 PB4
11 PB3
10 PB2
9 PB1
8 PB0
CMOS or LSTTL Driving Port B Directly
PA7 19
PA6 18
PA5 17
PA4 16
(CMOS Loads)
(1 LSTTL Load)
Port A, bit 7 programmed as output, driving CMOS loads and bit 4 driving one LSTTL load directly (using CMOS output option)
+ V
PB7 15
PB6 14
PB5 13
PB4 12
PB3 11
PB2 10
PB1 9
PB0 8
Port B, bit 0, and bit 1 programmed as output, driving LEDs directly

(a) Input Mode

(b) Output Mode

The latched output data bit (see figure 5.1) may always be written. Therefore, any write to a port writes to all of its data bits even though the port DDR is set to input. This may be used to initialize the data registers and avoid undefined outputs ; however, care must be exercised when using read-modify-write instructions since the data read corresponds to the pin level if the DDR is an input (0) and corresponds to the latched output data when the DDR is an output (1). The 12 bidirectional lines may be configured by port to be LSTTL (standard configuration), LSTTL/CMOS (mask option), or open drain (mask option). Port B outputs are LED compatible.

NOTE
The mask option only allows changes by port. For example, if the customer wishes PA7 to be open drain, then PA4-PA7 must all be open drain.

5.2. REGISTERS

The registers described below are implemented as RAM locations and thus may be read or written.

5.2.1. PORT DATA REGISTER. The source of data read from the port data register will be the port I/O pin or previously latched output data depending upon the contents of the corresponding data direction register (DDR). The destination of data written to the port data register will be an output data latch. If the corresponding data direction register (DDR) for the port I/O pin is programmed as an output, the data will then appear on the port pin.

7	0
MSB	LSB

Port A Address = $00
Port B Address = $01

5.2.2. PORT DATA DIRECTION REGISTER. The port DDRs configure the port pins as either inputs or outputs. Each port pin can be programmed individually to act as an input or an output. A zero in the pins corresponding bit position will program that pin as an input while a one in the pins corresponding bit position will program that pin as an output.

7	0
MSB	LSB

Port A Address = $04
Port B Address = $05

SECTION 6

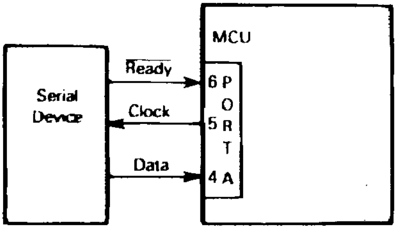
SOFTWARE AND INSTRUCTION SET

6.1. SOFTWARE

6.1.1. BIT MANIPULATION. The EF6804J2 MCU has the ability to set or clear any register or single random access memory (RAM) writable bit with a single instruction (BSET, BCLR). Any bit in data space, including ROM, can be tested, using the BRSET and BRCLR instructions, and the program may branch as a result of its state. The carry bit is set to the value of the bit referenced by BRSET or BRCLR. A rotate instruction may then be used to accumulate serial input data in a RAM location or register. The capability to work with any bit in RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle I/O bits as control lines.

The coding example in figure 6.1 illustrates the usefulness of the bit manipulation and test instructions. Assume that the MCU is to communicate with an external serial device. The external device has a data ready signal, a data output line, and a clock line (to clock data one bit at a time, MSB first, out of the device). The MCU waits until the data is ready, clocks the external device, picks up the data in the carry flag (C bit), clears the clock line, and finally accumulates the data bit in the accumulator.

Figure 6.1 : Bit Manipulation Example.



```
                                  .
SELF    BRSET        6, PORTA, SELF
                                  .
                                  .
                                  .
        BSET         5, PORTA
        BRCLR        4, PORTA, CONT
CONT    BCLR         5, PORTA
        ROLA                      .
                                  .
                                  .
                                  .
```

6.1.2. ADDRESSING MODES. The EF6804J2 MCU has nine addressing modes which are explained briefly in the following paragraphs. The EF6804J2 deals with objects in three different address spaces : program space, data space, and stack space. Program sapce contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains all of the RAM lcoatin, X and Y registers, accumulator, timer, I/O locations,and some ROM (for storage of tables and constants). Stack space contains RAM for use in stacking the return addresses for subroutines and interrupts.

The term "Effective Address" (EA) is used in describing the address modes. EA is defined as the address from which the argument for an instruction is fetched or stored.

6.1.2.1. Immediate

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

6.1.2.2. Direct

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

6.1.2.3. Short Direct

The MCU also has four locations in data space RAM ($80, $81, $82, $83) which may be used in a short-direct addressing mode. In this mode the lower two bits of the opcode determine the data space. RAM location, and the instruction is only one byte. Short direct addressing is a subset of the direct addressing mode. (The X and Y registers are at locations $80 or $81 respectively).

6.1.2.4. Extended

In the extended addressing mode, the effective address is obtained by concatenating the four least significant bits of the opcode with the byte following the opcode (12-bit address). Instructions using the extended addressing mode (JMP, JSR) are capable of branching anywhere in program space. An extended addressing mode instruction is two bytes long.

6.1.2.5. Relative

The relative addressing mode is only used in conditional branch instructions. In relative addressing, the address is formed by adding the sign extended lower five bits of the opcode (the offset) to the program counter if and only if the condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from - 15 to + 16 from the opcode address. The programmer need not worry about calculating the correct offset when using the Motorola assembler since it calculates the proper offset and checks to see if it is within the span of the branch.

6.1.2.6. Bit Set/Clear

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct address of the byte in which the specified bit is to be set or cleared. Thus, any bit in the 256 locations of data space memory, which can be written to, can be set or cleared.

6.1.2.7. Bit Test and Branch

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit which is to be tested is included in the opcode, and the data space address of the byte to be tested is in the single byte immediately following the opcode byte. The third byte is sign extended to twelve bits and becomes the offset added to the program counter if the condition is true. The single three-byte instruction allows the program to branch based on the condition of any bit in data space memory. The span of branching is from - 125 to + 130 from the opcode address. The state of the tested bit is also transferred to the carry flag.

6.1.2.8. Register-indirect

In the register-indirect addressing mode, the operand is at the address (in data space) pointed to by the contents of one of the indirect registers (X or Y). The particular X or Y register is selected by bit 4 of the opcode. Bit 4 of the opcode is then decoded into an address which selects the desired X or Y register ($80 or $81). A register-indirect instruction is one byte long.

6.1.2.9. Inherent

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

6.2. INSTRUCTION SET

The EF6804J2 MCU has a set of 42 basic instructions, which when combined with nine addressing modes produce 242 usable opcodes. They can be divided into five different types : register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

6.2.1. REGISTER/MEMORY INSTRUCTIONS. Most of these instructions use two operands. One operand is the accumulator and the other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operands. Refer to table 6.1.

6.2.2. READ-MODIFY-WRITE INSTRUCTIONS. These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. There are ten instructions which utilize read-modify-write cycles. All INC and DEC forms along with all bit manipulation instructions use this method. Refer to table 6.2.

6.2.3. BRANCH INSTRUCTIONS. The branch instructions cause a branch from the program when a certain condition is met. Refer to table 6.3.

6.2.4. BIT MANIPULATION INSTRUCTIONS. These instructions are used on any bit in data space memory. One group either sets or clears. The other group performs the bit test branch operations. Refer to table 6.4.

Table 6.1 : Register/memory Instructions.

		Addressing Modes																			
		Indirect				Immediate			Direct			Inherent			Extended			Short-Direct			
Function	Mnem	Opcode X	Opcode Y	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Special Notes
Load A from Memory	LDA	E0	F0	1	4	E8	2	4	F8	2	4	–	–	–	–	–	–	AC-AF	1	4	1
Load XP from Memory	LDXI	–	–	–	–	B0	3	4	–	–	–	–	–	–	–	–	–	–	–	–	4
Load YP from Memory	LDYI	–	–	–	–	B0	3	4	–	–	–	–	–	–	–	–	–	–	–	–	4
Store A in Memory	STA	E1	F1	1	4	–	–	–	F9	2	4	–	–	–	–	–	–	BC-BF	1	4	2
Add to A	ADD	E2	F2	1	4	EA	2	4	FA	2	4	–	–	–	–	–	–	–	–	–	–
Subtract from A	SUB	E3	F3	1	4	EB	2	4	FB	2	4	–	–	–	–	–	–	–	–	–	–
Arithmetic Compare with Memory	CMP	E4	F4	1	4	EC	2	4	FC	2	4	–	–	–	–	–	–	–	–	–	–
AND Memory to A	AND	E5	F6	1	4	ED	2	4	FD	2	4	–	–	–	–	–	–	–	–	–	–
Jump to Subroutine	JSR	–	–	–	–	–	–	–	–	–	–	–	–	–	8 (TAR)	2	4	–	–	–	3
Jump Unconditional	JMP	–	–	–	–	–	–	–	–	–	–	–	–	–	9 (TAR)	2	4	–	–	–	3
Clear A	CLRA	–	–	–	–	–	–	–	FB	2	4	–	–	–	–	–	–	–	–	–	–
Clear XP	CLRX	–	–	–	–	–	–	–	FB	2	4	–	–	–	–	–	–	–	–	–	–
Clear YP	CLRY	–	–	–	–	–	–	–	FB	2	4	–			–	–	–	–	–	–	–
Complement A	COMA	–	–	–	–	–	–	–	–	–	–	B4	1	4	–	–	–	–	–	–	–
Move Immediate Value to Memory	MVI	–	–	–	–	B0	3	4	B0	3	4	–	–	–	–	–	–	–	–	–	5
Rotate A Left and Carry	ROLA	–	–	–	–	–	–	–	–	–	–	B6	1	4	–	–	–	–	–	–	–
Arithmetic Left Shift of A	ASLA	–	–	–	–	–	–	–	FA	2	4	–	–	–	–	–	–	–	–	–	–

SPECIAL NOTES

1. In Short-Direct addressing, the LDA mnemonic represents opcode AC, AD, AE, and AF. This is equivalent to RAM locations $80 (AC), $81 (AD), $82 (AE), and $83 (AF)
2. In Short-Direct addressing, the STA mnemonic represents opcode BC BD, BE, and BF. This is equivalent to RAM locations $80 (BC), $81 (BD), $82 (BE), and $83 (BF).
3. In Extended addressing, the four LSBs of the opcode (Mnemonic JSR and JMP) are formed by the four MSBs of the target address. (TAR)
4. In Immediate addressing, the LDXI and LDYI are mnemonics which are recognized as follows:

   LDXI = MVI $80,data

   LDYI = MVI $81,data   Where data is a one-byte hexadecimal number.
5. The MVI instruction refers to both Immediate and Direct addressing.

Table 6.2 : Read-Modify-Write Instructions.

		Addressing Modes										
		Indirect				Direct			Short-Direct			Special Notes
Function	Mnem	Opcode		# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	
		X	Y									
Increment Memory Location	INC	E6	F6	1	4	FE	2	4	A8-AB	1	4	1, 3
Increment A	INCA					FE	2	4				
Increment XP	INCX								A8	1	4	
Increment YP	INCY								A9	1	4	
Decrement Memory Location	DEC	E7	F'	1	4	FF	2	4	B8-BB	1	4	2, 4
Decrement A	DECA					FF	2	4				
Decrement XP	DECX								B8	1	4	
Decrement YP	DECY								B9	1	4	

SPECIAL NOTES

1. In short-direct addressing, the INC mnemonic represents opcode A8, A9, AA and AB. These are equivalent to RAM locations $80 (A8), $81 (A9), $82 (AA) and $83 (AB).
2. In short-direct addressing, the DEC mnemonic represents opcode B8, B9, BA and BB. These are equivalent to RAM locations $80 (B8), $81 (B9), $82 (BA) and $83 (BB).
3. In indirect addressing, the INC mnemonic represents opcode E6 or F6, and causes the location pointec to by X (E6 opcode) or Y (F6 opcode) to be incremented.
4. In indirect addressing, the INC mnemonic represents opcode E7 or F7, and causes the location pointec to by X (E7 opcode) or Y (F7 opcode) to be incremented.

Table 6.3 : Branch Instructions.

		Relative Addressing Mode			
Function	**Mnem**	**Opcode**	**# Bytes**	**# Cycles**	**Special Notes**
Branch if Carry Clear	BCC	40-5F	1	2	1
Branch if Higher or Same	BHS	40-5F	1	2	1, 2
Branch if Carry Set	BCS	60-7F	1	2	1
Branch if Lower	BLO	60-7F	1	2	1, 3
Branch if Not Equal	BNE	00-1F	1	2	1
Branch if Equal	BEQ	20-3F	1	2	1

SPECIAL NOTES
1. Each mnemonic of the Branch Instructions covers a range of 32 opcodes, e.g., BCC ranges from 40 through 5F. The actual memory location (target address) to which the branch is made is formed by adding the sign extended lower five bits of the opcode to the contents of the program counter.
2. The BHS instruction (shown in parentheses) is identical to the BCC instruction. The C bit is clear if the register was higher or the same as the location in the memory to which it was compared.
3. The BLO instruction (shown in parentheses) is identical to the BCS instruction. The C bit is set if the register was lower than the location in memory to which it was compared.

Table 6.4 : Bit Manipulation Instructions.

		Addressing Modes						
		Bit Set/Clear			Bit Test and Branch			
Function	**Mnem**	**Opcode**	**# Bytes**	**# Cycles**	**Opcode**	**# Bytes**	**# Cycles**	**Special Notes**
Branch IFF Bit n is set	BRSET n (n = 0 7)				C8 + n	3	5	1
Branch IFF Bit n is clear	BRCLR n (n = 0 7)				C0 + n	3	5	1
Set Bit n	BSET n (n = 0 7)	D8 + n	2	4				1
Clear Bit n	BCLR n (n = 0 7)	D0 + n	2	4				1

SPECIAL NOTE
1. The opcode is formed by adding the bit number (0-7) to the basic opcode. For example to clear bit six using the BSET6 instruction the opcode becomes DE (D8 + 6), BCLR5 becomes (C0 + 5), etc.

6.2.5. CONTROL INSTRUCTIONS. The control instructions control the MCU operations during program execution. Refer to table 6.5.

6.2.6. ALPHABETICAL LISTING. The complete instruction set is given in alphabetical order in table 6.6. There are certain mnemonics recognized by the assembler and converted to other instructions. The fact that all registers and accumulator are in RAM allows many implied instructions to exist. The implied instructions recognized by the assembler are identified in table 6.6.

6.2.7. OPCODE MAP SUMMARY. Table 6.7 contains an opcode map for the instructions used on the MCU.

6.3. IMPLIED INSTRUCTIONS

Since the accumulator and all other registers are located in RAM many implied instructions exist. The assembler-recognized implied instructions are given in table 6.6. Some examples not recognized by the assembler are shown below.

BCLR, 7 $FF	Ensures accumulator is plus
BSET, 7 $FF	Ensures accumulator is minus
BRCLR, 7 $FF	Branch iff accumulator is plus
BRSET, 7 $FF	Branch iff accumulator is minus
BRCLR, 7 $80	Branch iff X is plus (BXPL)
BRSET, 7 $80	Bracnh iff X is minus (BXMI)
BRCLR, 7 $81	Branch iff Y is plus (BYPL)
BRSET, 7 $81	Branch iff Y is minus (BYMI)

Table 6.5 : Control Instructions.

		Addressing Modes									
		Short-Direct			Inherent			Relative			Special Notes
Function	Mnem	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	
Transfer A to X	TAX	BC	1	4							
Transfer A to Y	TAY	BD	1	4							
Transfer X to A	TXA	AC	1	4							
Transfer Y to A	TYA	AD	1	4							
Return from Subroutine	RTS				B3	1	2				
Return from Interrupt	RTI				B2	1	2				
No-operation	NOP										1

SPECIAL NOTE

1. The NOP instruction is equivalent to a branch if equal (BEQ) to the location designated by PC + 1.

Table 6.6 : Instruction Set.

Mnemonic	Addressing Modes									Flags	
	Inherent	Immediate	Direct	Short Direct	Bit/Set Clear	Bit-Test-Branch	Register Indirect	Extended	Relative	Z	C
ADD		X	X				X			∧	∧
AND		X	X				X			∧	•
ASLA			Assembler converts this to "ADD $FF"							•	•
BCC									X	∧	∧
BCLR					X					•	•
BCS									X	•	•
BEQ									X	•	•
BHS			Assembler converts this to "BCC"							•	•
BLO			Assembler converts this to "BCS"							•	•
BNE									X	•	•
BRCLR						X				•	∧
BRSET						X				•	∧
BSET					X					•	•
CLRA			Assembler converts this to "SUB $FF"							∧	∧
CLRX			Assembler converts this to "MVI = 0, $80"							•	•
CLRY			Assembler converts this to "MVI = 0, $81"							•	•
CMP		X	X				X			∧	∧
COMA	X									∧	∧
DEC			X	X			X			∧	•
DECA			Assembler converts this to "DEC $FF"							∧	•
DECX			Assembler converts this to "DEC $80"							∧	•
DECY			Assembler converts this to "DEC $81"							∧	•
INC			X	X			X			∧	•
INCA			Assembler converts this to "INC $FF"							∧	•
INCX			Assembler converts this to "INC $80"							∧	•
INCY			Assembler converts this to "INC $81"							∧	•
JMP								X		•	•
JSR								X		•	•
LDA		X	X	X			X			∧	•
LDXI			Assembler converts this to "MVI DATA, $80"							•	•
LDYI			Assembler converts this to "MVI DATA, $81"							•	•
MVI		X	X							•	•
NOP			Assembler converts this to "BEQ (PC) + 1"							•	•
ROLA	X									∧	∧
RTI	X									∧	∧
RTS	X									•	•
STA			X	X			X			∧	•
SUB		X	X				X			∧	∧
TAX	Assembler converts this to "STA $80"									∧	•
TAY	Assembler converts this to "STA $81"									∧	•
TXA	Assembler converts this to "LDA $80"									∧	•
TYA	Assembler converts this to "LDA $81"									∧	•

Flag Symbols Z = Zero, C = Carry/borrow, ∧ = Test and Set if True, Cleared Otherwise, • = Not affected

Table 6.7 : EF6804P2 Microcomputer Instruction Set Opcode Map.

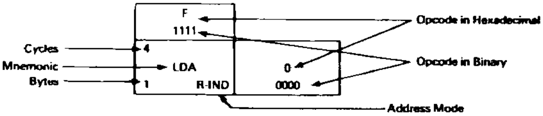
	Branch Instructions							
Hi Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
0 0000	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
1 0001	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
2 0010	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
3 0011	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
4 0100	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
5 0101	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
6 0110	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
7 0111	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
8 1000	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
9 1001	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
A 1010	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
B 1011	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
C 1100	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
D 1101	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
E 1110	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
F 1111	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL

Abbreviations for Address Modes

INH	Inherent
S-D	Short Direct
B-T-B	Bit Test and Branch
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
R-IND	Register Indirect
*	Indicates Instruction Reserved for Future Use
#	Indicates Illegal Instruction

Table 6.7 : (continued).

Register/Memory, Control, and Read/Modify/Write Instructions				Bit Manipulation Instructions		Register/Memory and Read/Modify/Write		
8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
4 JSRn 2 EXT	4 JMPn 2 EXT	*	4 MVI 3 IMM	5 BRCLR0 3 B-T-B	5 BCLR0 3 BSC	4 LDA 1 R-IND	4 LDA 1 R-IND	0 0000
4 JSRn 2 EXT	4 JMPn 2 EXT	*	*	5 BRCLR1 3 B-T-B	5 BCLR1 3 BSC	4 STA 1 R-IND	4 STA 1 R-IND	0 0001
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 RTI 1 INH	5 BRCLR2 3 B-T-B	5 BCLR2 3 BSC	4 ADD 1 R-IND	4 ADD 1 R-IND	2 0010
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 RTS 1 INH	5 BRCLR3 3 B-T-B	5 BCLR2 3 BSC	4 SUB 1 R-IND	4 SUB 1 R-IND	3 0011
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 COMA 1 INH	5 BRCLR4 3 B-T-B	5 BCLR4 3 BSC	4 CMP 1 R-IND	4 CMP 1 R-IND	4 0100
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 ROLA 1 INH	5 BRCLR5 3 B-T-B	5 BCLR5 3 BSC	4 AND 1 R-IND	4 AND 1 R-IND	5 0101
4 JSRn 2 EXT	4 JMPn 2 EXT	*	*	5 BRCLR6 3 B-T-B	5 BCLR6 3 BSC	4 INC 1 R-IND	4 INC 1 R-IND	6 0110
4 JSRn 2 EXT	4 JMPn 2 EXT	*	*	5 BRCLR7 3 B-T-B	5 BCLR7 3 BSC	4 DEC 1 R-IND	4 DEC 1 R-IND	7 0111
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET0 3 B-T-B	5 BSET0 3 BSC	4 LDA 2 IMM	4 LDA 2 DIR	8 1000
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET1 3 B-T-B	5 BSET1 3 BSC	#	4 STA 2 DIR	9 1001
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET2 3 B-T-B	5 BSET2 3 BSC	4 ADD 2 IMM	4 ADD 1 DIR	A 1010
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET3 3 B-T-B	5 BSET3 3 BSC	4 SUB 2 IMM	4 SUB 2 DIR	B 1011
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET4 3 B-T-B	5 BSET4 3 BSC	4 CMP 2 IMM	4 CMP 2 DIR	C 1100
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET5 3 B-T-B	5 BSET5 3 BSC	4 AND 2 IMM	4 AND 2 DIR	D 1101
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET6 3 B-T-B	5 BSET6 3 BSC	#	4 INC 2 DIR	E 1110
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET7 3 B-T-B	5 BSET7 3 BSC	#	4 DEC 2 DIR	F 1111



SECTION 7

ELECTRICAL SPECIFICATIONS

7.1. INTRODUCTION

This section contains the electrical specifications and associated timing for the EF6804J2.

7.2. 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 Standard or L Suffix V Suffix T Suffix	TL to TH 0 to + 70 – 40 to + 85 – 40 to + 105	°C
T_{stg}	Storage Temperature Range	– 55 to + 150	°C
T_J	Junction Temperature Range Plastic	150	°C/W

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

7.3. THERMAL DATA

θ_{JA}	Thermal Resistance Plastic	90	°C/W

Figure 7.3: LSTTL Equivalent Test Load (port A and TIMER).

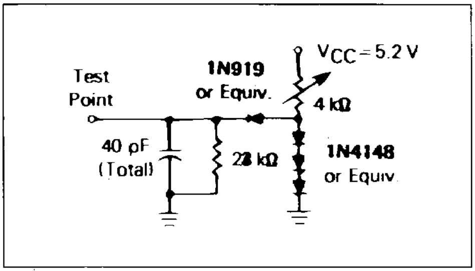


Figure 7.2 : CMOS Equivalent Test Load (port A and B).

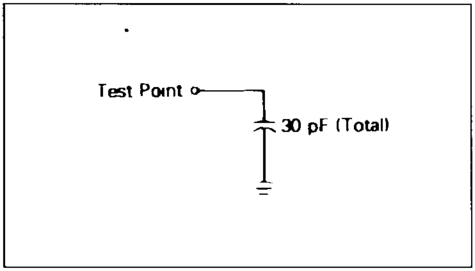
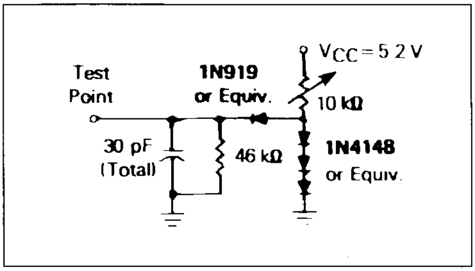


Figure 7.1 : LSTTL Equivalent Test Load (port B).



7.4. POWER CONSIDERATIONS

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

$$T_J = T_A + (P_D \cdot \theta J_A) \qquad (1)$$

Where :

T_A = Ambient Temperature, °C

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

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

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

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

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

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

$$P_D = K \div (T_J + 273^\circ C) \qquad (2)$$

Solving equations 1 and 2 for K gives :

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

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) interatively for any value of T_A.

7.5. ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.0 Vdc ± 0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
P_{INT}	Internal Power Dissipation–No Port Loading T_A = 0°C T_A = – 40°C		135	170 210	mV
V_{IH}	Input High Voltage	4.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
C_{in}	Input Capacitance		10		pF
I_{in}	Input Current($\overline{\text{IRQ}}$, $\overline{\text{RESET}}$)		2	20	µA

7.6. ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.0 Vdc ± 0.5 Vdc, V_{SS} = GND, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
f_{OSC}	Oscillator Frequency	4.0		11.0	MHz
t_{bit}	Bit Time	0.364		1.0	µs
t_{byte}	Byte Cycle Time	4.36		12.0	µs
t_{WL}, t_{WH}	$\overline{\text{IRQ}}$ and $\overline{\text{TIMER}}$ Pulse Width	2 x t_{byte}			
t_{RWL}	$\overline{\text{RESET}}$ Pulse Width	2 x t_{byte}			
t_{RHL}	$\overline{\text{RESET}}$ Delay Time (external capacitance = 1.0µF)	100			ms

7.7. PORT DC ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.0 Vdc ± 0.5 Vdc, V_{SS} = GND, T_A = T_L to T_H unless otherwise noted)

TIMER AND PORTS A (standard)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 0.4mA			0.5	V
V_{OH}	Output High Voltage, I_{Load} = – 50μA	2.3			V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current		4	40	μA

TIMER AND PORTS A (open drain)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 0.4mA			0.5	V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current		4	40	μA
I_{LOD}	Open Drain Leakage (V_{out} = V_{CC})		4	40	μA

TIMER AND PORTS A (CMOS drive)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 0.4mA (sink)			0.5	V
V_{OH}	Output High Voltage, I_{Load} = – 10μA	V_{CC} – 1.0			V
V_{OH}	Output High Voltage, I_{Load} = – 50μA	2.3			V
V_{IH}	Input High Voltage, I_{Load} = – 300μA Max	2.0		V_{CC}	V
V_{IL}	Input Low Voltage, I_{Load} = – 300μA Max	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current (V_{in} = 0.4V to V_{CC})			– 300	μA

PORT B (standard)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.0mA			0.5	V
V_{OL}	Output Low Voltage, I_{Load} = 10mA (sink)			1.5	V
V_{OH}	Output High Voltage, I_{Load} = – 100μA	2.3			V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{ISS}		0.8	V
I_{TSI}	Hi-Z State Input Current		8	80	μA

PORT B (open drain)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.0mA			0.5	V
V_{OL}	Output Low Voltage, I_{Load} = 10mA (sink)			1.5	V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current		8	80	µA
I_{LOD}	Open Drain Leakage (V_{out} = V_{CC})		8	80	µA

PORT B (CMOS drive)

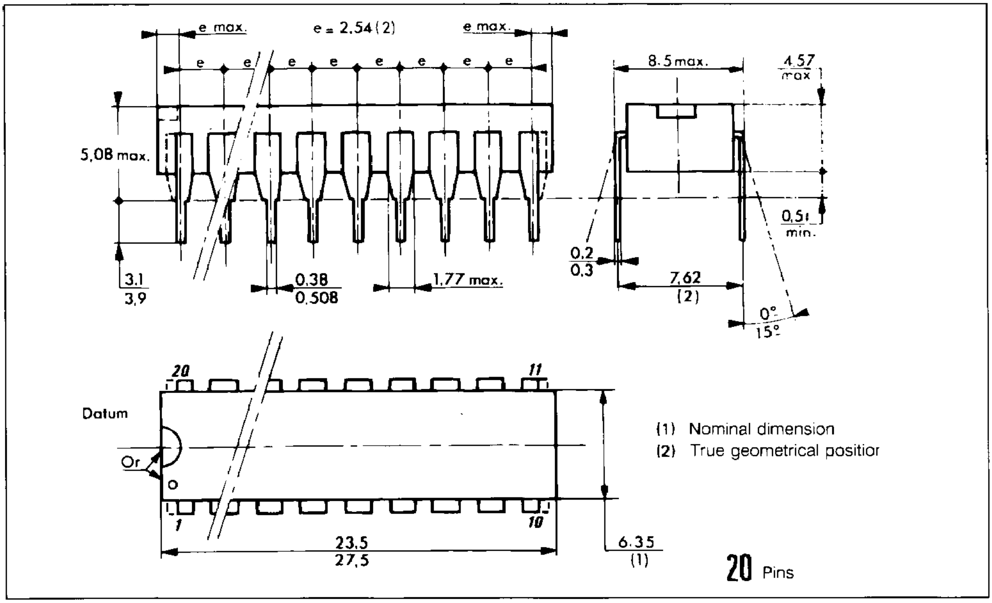
Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.0mA			0.5	V
V_{OL}	Output High Voltage, I_{Load} = 10mA (sink)			1.5	V
V_{OH}	Output High Voltage, I_{Load} = – 10µA	V_{CC} – 1.0			V
V_{OH}	Output High Voltage, I_{Load} = – 100µA	2.3			V
V_{IH}	Input High Voltage, I_{Load} = – 300µA Max	2.0		V_{CC}	V
V_{IL}	Input Low Voltage, I_{Load} = – 300µA Max	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current (V_{in} = 0.4V to V_{CC})			– 300	µA

SECTION 8

MECHANICAL DATA

This section contains the pin assignment and package dimension diagrams for the EF6804J2 microcomputer.

MECHANICAL DATA



(1) Nominal dimension
(2) True geometrical positior

20 Pins

SECTION 9

ORDERING INFORMATION

9.1. INTRODUCTION

The following information is required when ordering a custom MCU. The information may be transmitted to SGS-THOMSON in the following media :

EPROM(s), 2716 or 2732

EFDOS/MDOS *, disk file

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

9.1.1. EPROMs. One 2716 or one 2732 type EPROM, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one 2716 or 2732 EPROM, the EPROM must be programmed as follows in order to emulate the EF6804J2 MCU. For a 2716, start the data space ROM at EPROM address $020 and start program space ROM at EPROM address $410 and continue to memory space $7FF. Memory spaces $7F8 through $7FB are reserved for SGS-THOMSON self-test vectors. For a 2732, the memory map shown in figure 2.1 can be used. All unused bytes, including the user's space, must be set to zero. For shipment to SGS-THOMSON the EPROMs should be placed in a conductive IC carrier and packed securely. Do not use styrofoam.

9.1.2. EFDOS/MDOS * DISK FILE. An EFDOS/MDOS* disk, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. When using the EFDOS/MDOS* disk, include the entire memory image of both data and program space. All unused bytes, including the user's space, must be set to zero.

9.2. 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 contractural agreement for creation of the customer mask. If desired, SGS-THOMSON will program a blank 2716, 2732, or EFDOS/MDOS* disk (supplied by the customer) from the data file used to create the custom mask to aid in the verification process.

* Requires prior factory approval

9.3. 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 tested only at room temperature, five volts and may be unmarked and packaged in ceramic. These RVUs are included in the mask charge and are not production parts.

These RVUs are not backed nor guaranteed by SGS-THOMSON Quality Assurance.

9.4. FLEXIBLE DISKS

The disk media submitted must be single-sided single density, 8-inch, EFDOS/MDOS* compatible floppies. The customer must clearly label the disk with the ROM pattern file name. The minimum EFDOS/MDOS* system files as well as the absolute binary object file (file name. LO type of file) from the 6804 cross assembler must be on the disk. An object file made from a memory dump, using the ROLLOUT command is also admissable. Consider submitting a source listing as well as : file name, LX (DEVICE/EXORciser loadable format). This file will of course be kept confidential and is used 1) to speed up the process in house if any problems arise, and 2) to speed up our customer to factory interface if a user finds any software errors and needs assistance quickly from SGS-THOMSON factory representatives.

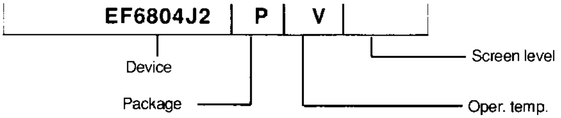
EFDOS is SGS-THOMSON Disk Operating System available on development systems such as DEVICE.

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 Microelectronics distributor and/or complete and send the attached "MCU customer ordering sheet SEMICONDUCTEURS 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			ScreeningLevel			
	C	J	P	E	FN	L*	V	T	Std	D		
EF6804J2			●			●	●		●	●		
Examples :EF6804J2P, EF6804J2PV, EF6804J2PLD, EF6804J2PVD												

Package : C : Ceramic DIL, **P :** Plastic DIL, **FN :** PLCC
Oper. temp. : L* : 0°C to + 70°C, **V :** – 40°C to + 85°C, **T :** – 40°C to + 105°C, ***** : may be omitted.
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D.

EXORciser is a registered trademark of MOTOROLA Inc.

These specifications are subject to change without notice
Please inquire with our sales offices about the availability of the different packages

EF6804 FAMILY - MCU CUSTOMER ORDERING SHEET

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

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

Application : ....
....

ROM capacity required : | | | | | bytes

Tempèrature range :
□ 0°C / + 70°C
□ – 40°C / + 85°C
□ – 40°C / + 105°C

Package
□ Plastic

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

Specification reference ;
□ SGS-THOMSON Microelectronics reference
....
□ Special customer data reference*
....

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

Software developped 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

-Oscillator input :
□ Xtal
□ RC

- Interrupt Trigger :
□ Edge -sensitive
□ Level-and edge-sensitive

- Port A output drive (4I/Os)
□ Enabled
□ Disabled

- Port B output drive :
□ CMOS and TTL
□ TTL only
□ Open drain

* Requires prior factory approval

Yearly quantity forecast :

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

CUSTOMER CONTACT NAME : DATE : SIGNATURE :