FAIRCHILD
A Schlumberger Company

F6802/F6882/F6808 Microprocessor with Clock and RAM

Microprocessor Product

Description

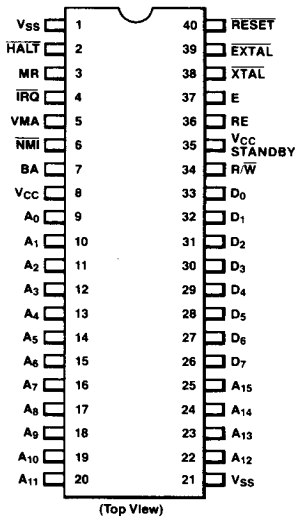
The F6802/F6882 is a monolithic 8-bit microprocessor that contains all the registers and accumulators of the F6800, plus an internal clock oscillator and driver on the same chip. The F6802/F6882 also has 128 bytes of RAM on board, located at hex addresses $0000 to $007F, V_{CC} standby can be utilized on the F6802/F6882 to facilitate memory retention during a power-down situation; the first 8 bytes of RAM at hex addresses $0000 to $0007 can be retained on the F6882, and the first 32 bytes of RAM at hex addresses $0000 to $001F can be retained on the F6802. The F6808 is identical to the F6802 without on-board RAM.

The F6802/F6882 is completely software-compatible with the F6800 microprocessor and the entire F6800 family of parts. (*Figure 1* illustrates a typical application using an F6800 family device.)

- **On-Chip Clock Circuit**
- **128 x 8-bit On-Chip RAM (Not Included on F6808)**
- **8 Bytes of RAM are Retainable on the F6882**
- **32 Bytes of RAM are Retainable on the F6802**
- **Software-Compatible with the F6800**
- **Standard TTL-Compatible Inputs and Outputs**
- **8-bit Bidirectional Data Bus**
- **16-bit Memory Addressing**
- **Interrupt Capability**
- **Speed Grades:**
  - **1.0 MHz F6802/F6882/F6808**
  - **1.5 MHz F68A02/F68A82/F68A08**

Connection Diagram

40-Pin DIP



(Top View)

F6802/F6882/F6808 Signal Functions

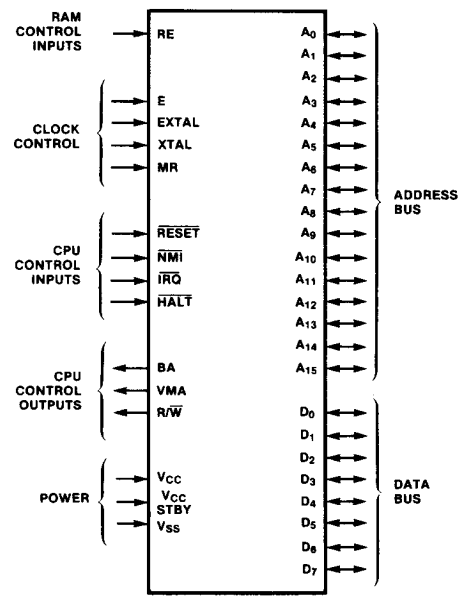
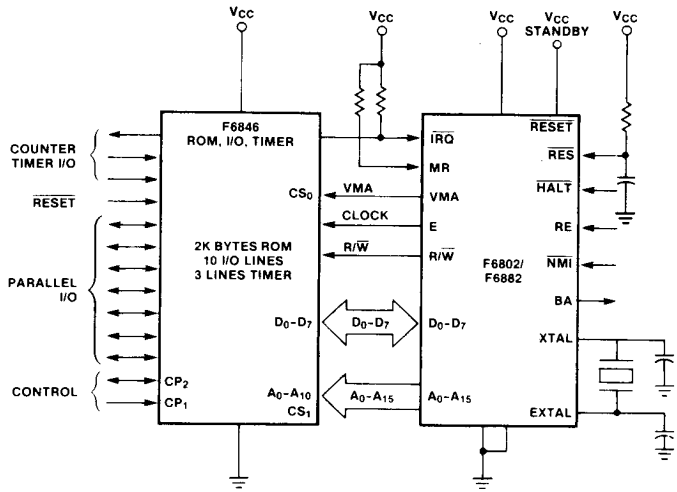


Fig. 1 Typical Microcomputer Block Diagram



Registers

A general block diagram of the F6802/F6882 is shown in *Figure 2*. The number and configuration of the registers are identical to the F6800, as shown, with a 128 x 8-bit RAM* added to the basic microprocessor. The first 8 bytes in the F6882 and the first 32 bytes in the F6802 may be operated in a low-power mode via a V_{CC} standby and can be retained during power-up and power-down conditions via the RE signal. The F6808 is identical to the F6802 except for on-board RAM. Since the F6808 does not have on-board RAM, pin 36 must be tied to ground, allowing the processor to utilize up to 64K bytes of external memory.

The microprocessing unit (MPU) has three 16-bit registers and three 8-bit registers available for use by the programmer, as shown in *Figure 3*.

Program Counter

The program counter is a 2-byte (16-bit) register that points to the current program address.

Stack Pointer

The stack pointer is a 2-byte register that contains the address of the next available location in an external push-down/pop-up stack. This stack is normally a random access read/write memory that may have any location (address) that is convenient. In those applications that require storage of information in the stack when power is lost, the stack must be non-volatile.

Index Register

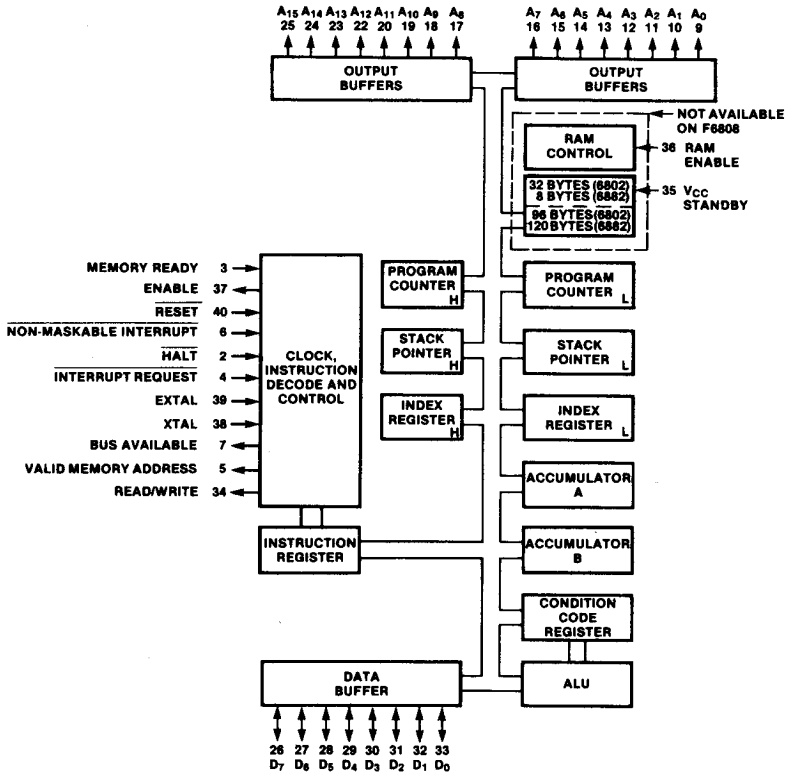
The index register is a 2-byte register that is used to store data or a 16-bit memory address for the indexed mode of memory addressing.

Accumulators

The two 8-bit accumulators are used to hold operands and results from an arithmetic logic unit (ALU).

*If programs are not executed from on-board RAM, TAV1 applies. If programs are to be stored and executed from on-board RAM, TAV2 applies. For normal data storage in the on-board RAM, this extended delay does not apply. Programs can be executed from on-board RAM when using 1.5 parts. On-board RAM can be used for data storage with all parts.

Fig. 2 F6802/F6882 Block Diagram



Condition Code Register (Status Word Register)
The condition code register indicates the results of an arithmetic logic unit operation: negative (N), zero (Z), overflow (V), carry from bit 7 (C), and half-carry from bit 3 (H). These bits of the condition code register are used as testable conditions for the conditional branch instructions. Bit 4 is the interrupt mask bit (I). The unused bits of the condition code register (bit 6 and bit 7) are binary ones (1).

Figure 4 shows the order of saving the microprocessor status within the stack.

Fig. 3 Programming Model of the Microprocessing Unit

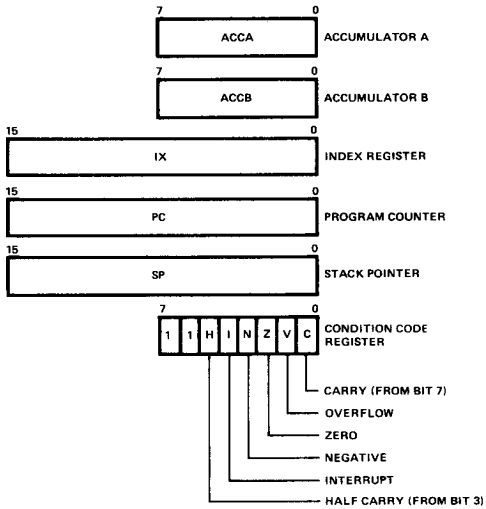
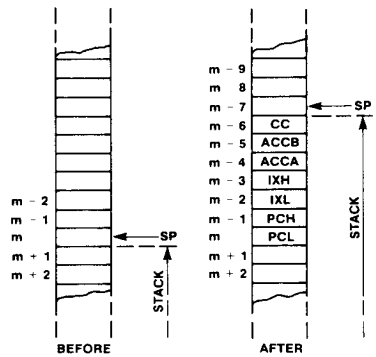


Fig. 4 Saving the Status of the Microprocessor in the Stack



Notes
SP = Stack Pointer
CC = Condition Code (also called the Processor Status Byte)
ACCB = Accumulator B
ACCA = Accumulator A
IXH = Index Register, higher order 8 bits
IXL = Index Register, lower order 8 bits
PCH = Program Counter, higher order 8 bits
PCL = Program Counter, lower order 8 bits

F6802/F6882 Signal Descriptions

The control and timing signals for the F6802/F6882 are identical to those of the F6800, with the following exceptions:

1. TSC, DBE ϕ_1, ϕ_2 input, and two unused pins have been eliminated.

2. The following signal and timing lines have been added:
   RAM Enable (RE)
   Crystal Connections EXtal and Xtal
   Memory Ready (MR)
   V_{CC} Standby
   Enable ϕ_2 Output (E)

The following summarizes the F6802/F6882 MPU signals.

Data Bus

D_0-D_7 (Data Bus Lines), Pins 26-33
Bidirectional bus used to transfer data to and from the memory and peripheral devices. Also has 3-state output buffers capable of driving one standard TTL load and 130 pF.

Data bus lines are in the output mode when the internal RAM is accessed. This prohibits external data from entering the MPU. The internal RAM is fully decoded from addresses $0000 to $007F. External RAM at $0000 to $007F must be disabled when internal RAM is accessed.

Address Bus

A_0-A_{15} (Address Bus Lines), Pins 9-20, 22-25
Sixteen output lines form the address bus. The outputs are capable of driving one standard TTL load and 90 pF. These lines do not have 3-state capability.

CPU Control Inputs

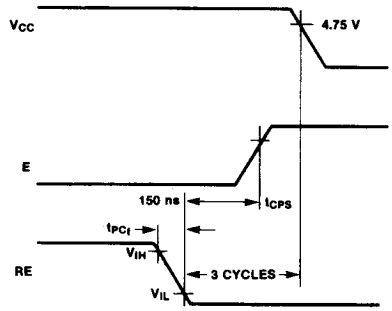
$\overline{\text{RESET}}$ (Reset), Pin 40
Input used to reset and start the MPU from a power-down condition resulting from a power failure or an initial start-up of the processor. When this line is low,the MPU is inactive and the information in the registers is lost. If a high level is detected on the input, this signals the MPU to begin the restart sequence. This starts execution of a routine to initialize the processor from its reset condition. All the higher order address lines are forced high. For the restart, the last two locations in memory

($FFFE, $FFFF) are used to load the program that is addressed by the program counter. During the restart routine, the interrupt mask bit is set and must be reset before the MPU can be interrupted by $\overline{IRQ}$. Power-up and reset timing sequences are shown in *Figures 5* and *6*.

When brought low, $\overline{RESET}$ *must be held low at least three clock cycles.* This is independent of the power-up delay required for oscillator start-up (T_{RC}).

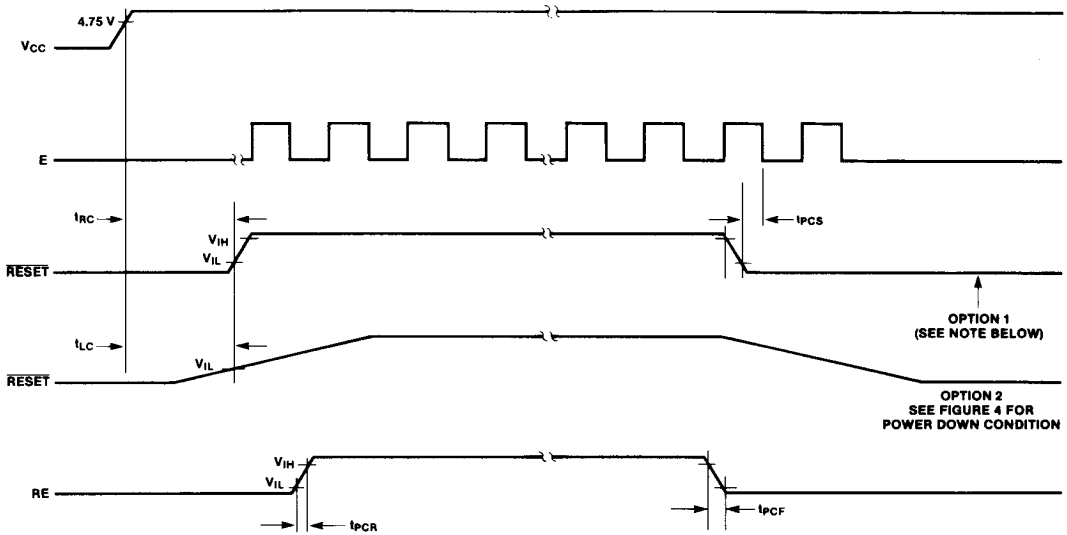
When $\overline{RESET}$ is released, it must go through the low-to-high threshold without bouncing, oscillating, or otherwise causing an erroneous reset (less than three clock cycles) that may cause improper MPU operation until the next valid reset.

Fig. 6 Power-Down Sequence



5

Fig. 5 Power-Up and Reset Timing



Note
If option 1 is chosen, $\overline{RESET}$ and RE pins can be tied together.

$\overline{NMI}$ (Non-Maskable Interrupt), Pin 6
A low-going edge on this input requests that a non-maskable interrupt sequence be generated within the processor. As with the interrupt request ($\overline{IRQ}$) signal, the processor completes the current instruction being executed before it recognizes the $\overline{NMI}$ signal. The interrupt mask bit in the condition code register has no effect on $\overline{NMI}$.

The index register, program counter, accumulators, and condition code register are stored on the stack, as shown in *Figure 4*. At the end of the cycle, a 16-bit address will be loaded from memory locations $FFFC and $FFFD that points to a vectoring address. An address loaded from these locations causes the MPU to branch to a non-maskable interrupt routine in memory.

A nominal 3 kΩ external resistor to V_{CC} should be used for wire-OR and optimum control of interrupts. The $\overline{NMI}$ signal may be tied directly to V_{CC} if not used.

The $\overline{IRQ}$ and $\overline{NMI}$ inputs are hardware interrupt lines that are sampled when E is high and start the interrupt routine on a low E following the completion of an instruction.

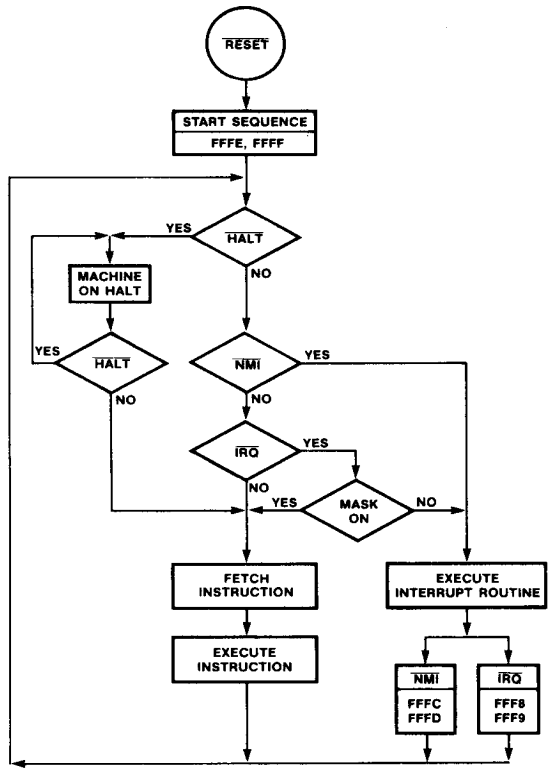
Figure 7 is a flow chart describing the major decision paths and interrupt vectors of the microprocessor. *Table 1* gives the memory map for interrupt vectors.

Table 1 Memory Map for Interrupt Vectors

Vector		
MS	LS	Description
$FFFE	$FFFF	Restart
$FFFC	$FFFD	Non-Maskable Interrupt
$FFFA	$FFFB	Software Interrupt
$FFF8	$FFF9	Interrupt Request

$\overline{IRQ}$ (Interrupt Request), Pin 4
This level-sensitive input requests that an interrupt sequence be generated within the machine. The processor waits until it completes the current instruction that is being executed before it recognizes the request. At that time, if the interrupt mask bit in the condition code register is not set, the machine begins an interrupt sequence. The index register, program counter, accumulators, and condition code register are stored on the stack as shown in *Figure 4*. The MPU responds to the interrupt request by setting the interrupt mask bit high

Fig. 7 MPU Flow Chart



so that no further interrupts may occur. At the end of the cycle, a 16-bit address is loaded from memory locations $FFF8 and $FFF9 that point to a vectoring address. An address loaded from these locations causes the MPU to branch to an interrupt routine in memory.

The $\overline{HALT}$ line must be in the high state for interrupts to be serviced. Interrupts are latched internally while $\overline{HALT}$ is low.

The $\overline{IRQ}$ has a high-impedance pull-up device internal to the chip; however, a 3kΩ external resistor to V_{CC} should be used for the wire-OR and optimum control of interrupts.

$\overline{\text{HALT}}$ (Halt), Pin 2
When this input is in the low state, all activity in the machine is halted. This input is level-sensitive. In the halt mode, the machine stops at the end of an instruction. Bus Available is in a high state, and Valid Memory Address is in a low state. The address bus displays the address of the next instruction.

To ensure single-instruction operation, transition of the $\overline{\text{HALT}}$ line must occur t_{PCS} before the falling edge of E and the $\overline{\text{HALT}}$ line must go high for one clock cycle.

$\overline{\text{HALT}}$ should be tied high if not used.

RAM Control Port

RE (RAM Enable), Pin 36
A TTL-compatible RAM enable input that controls the on-chip RAM. When placed in the high state, the on-chip memory is enabled to respond to the MPU controls. In the low state, the RAM is disabled. This pin may also be utilized to disable reading from and writing to the on-chip RAM during a power-down situation. The RE signal must be low three cycles before V_{CC} goes below 4.75 V during power-down as shown in *Figure 6.*

The RE signal should be tied low on the F6808; it should be tied to the correct high or low state if not used.

CPU Control Outputs

BA (Bus Available), Pin 7
Is normally in the low state; when activated, it goes to the high state, indicating that the microprocessor has stopped and that the address bus is available (but not in a 3-state condition). This occurs if the $\overline{\text{HALT}}$ line is in the low state or the processor is in the wait state as a result of execution of a WAIT instruction. At such time, all 3-state output drivers go to their off state and other outputs to their normally inactive levels. The processor is removed from the wait state by the occurrence of a maskable (mask bit 1 = 0) or nonmaskable interrupt. This output is capable of driving one standard TTL load and 30 pF.

VMA (Valid Memory Address), Pin 5
This output indicates to peripheral devices that there is a valid address on the address bus. In normal operation, this signal should be utilized for enabling peripheral interfaces, such as the PIA and ACIA. This signal is not 3-state. One standard TTL load and 90 pF may be directly driven by this active-high signal.

R/$\overline{\text{W}}$ (Read, Write), Pin 34
This TTL-compatible output signals the periphals and memory devices whether the MPU is in a read (high) or write (low) state. The normal standby state of this signal is read (high). When the processor is halted, it is in the read state. This output is capable of driving one standard TTL load and 90 pF.

Power

V_{CC} (Power Supply), Pin 8
V_{CC} tolerance is ±5%.

V_{CC} STBY (Power Supply Standby), Pin 35
This pin supplies the dc voltage to the first 8 or 32 bytes of RAM as well as the RAM enable (RE) control logic. Thus, retention of data in this portion of the RAM on a power-up, power-down, or standby condition is guaranteed. Maximum current drain at maximum V_{SB} is I_{SBB}.

V_{SS} (Ground), Pins 1, 21
System ground; 0 V reference.

Clock Control

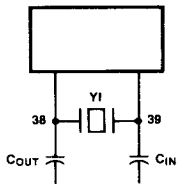
E (Enable), Pin 37
This pin supplies the clock for the MPU and the rest of the system. This is a single-phase, TTL-compatible clock and may be conditioned by a memory ready (MR) signal. The E signal is equivalent to ϕ_2 on the F6800, and is capable of driving one TTL load and 130 pF.

EXTAL (External Crystal Connector), Pin 39
XTAL (Crystal Connector), Pin 38
The F6802/F6882 has an internal oscillator that may be crystal controlled. These connections are for a parallel-resonant, AT cut, fundamental crystal. (*Figure 8* illustrates the crystal specifications.) A divide-by-four circuit has been added so that a 4 MHz crystal may be used in place of a 1 MHz crystal for a more cost-effective system. An example of the crystal circuit layout on a printed circuit board is shown in *Figure 9.*

Pin 39 may be driven externally by a TTL-input signal four times the required clock frequency. Pin 38 is to be grounded in this mode.

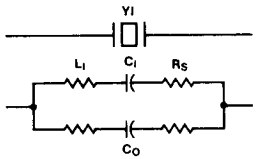
An RC network is not directly usable as a frequency source on pins 38 and 39. An RC network-type TTL or CMOS oscillator works well as long as the TTL or CMOS output drives the on-chip oscillator.

Fig. 8 Crystal Specification



YI	C_{IN}	C_{OUT}
3.58 MHz	27 pF	27 pF
4 MHz	27 pF	27 pF
6 MHz	20 pF	20 pF
8 MHz	18 pF	18 pF

Crystal Loading

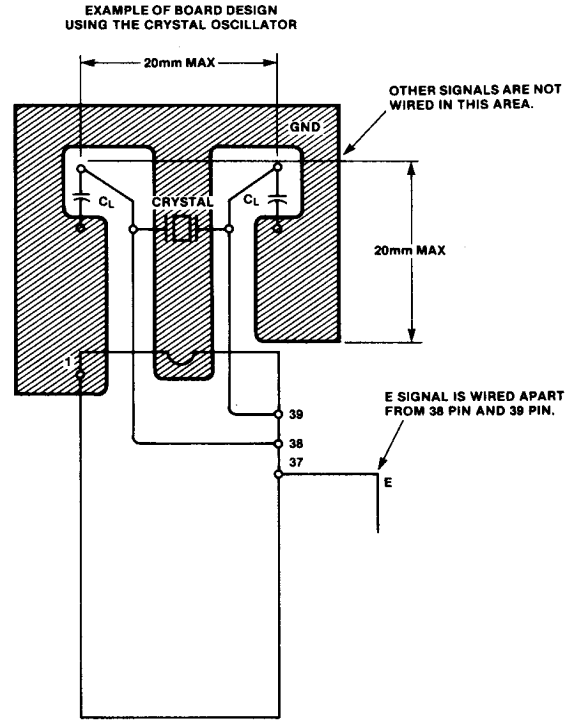


	3.58 MHz	4.0 MHz	6.0 MHz	8.0 MHz
R_S	60Ω	50Ω	30-50Ω	20-40Ω
C_O	3.5 pF	6.5 pF	4-6 pF	4-6 pF
C_I	0.015 pF	0.025 pF	0.01-0.02 pF	0.01-0.02 pF
Q	>40K	>30K	>20K	>20K

Nominal Crystal Parameters*

*These are representative AT-cut parallel-response crystal parameters only. Crystals of other types of cuts may also be used.

Fig. 9 Suggested PC Board Layout



LC networks in place of the crystal are not recommended.

If an external clock is used, it may be halted for more than $PWOL. The F6802/F6882/F6808 is a dynamic part except for the internal RAM, and requires the external clock to retain information.

MR (Memory Ready), Pin 3
A TTL-compatible input control signal that allows stretching of the enable (E) signal. Use of MR requires synchronization with the $4xf_o$ signal, as shown in *Figure 10.* When MR is high, E will be in normal operation. When MR is low, E may be stretched integral multiples of half periods, allowing interface to slow memories or peripherals. A maximum stretch is t_{CYC}; The MR signal should be tied high if not used. Refer to *Figure 11* for MR timing information.

Fig. 10 Memory Ready Synchronization

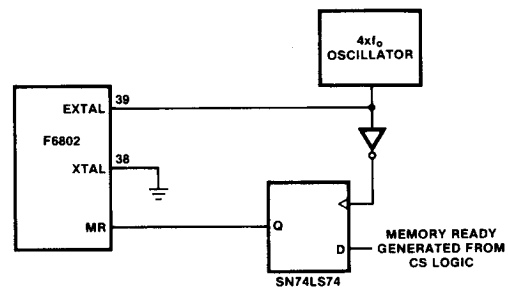
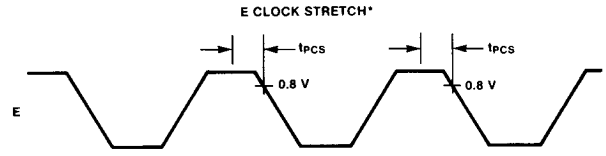
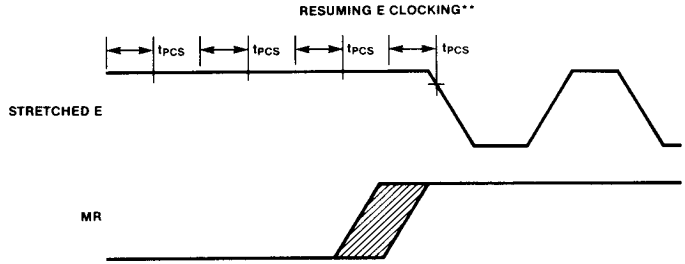


Fig. 11 MR Negative Setup Time Requirement



*The E clock will be stretched at the end of E high of the cycle during which MR negative meets the t_{PCS} setup time. The t_{PCS} setup time is referenced to the fall of E. If the t_{PCS} setup time is not met, E is stretched at the end of the next E high one-half cycle. The E signal isstretched in integral multiples of one-half cycles.



**The E clock resumes normal operation at the end of the one-half cycle during which MR assertion meets the t_{PCS} setup time. The t_{PCS} setup time is referenced to transitions of E were it not stretched. If t_{PCS} setup time is not met, E falls at the second possible transition time after MR is asserted. There is no direct means of determining when the t_{PCS} references occur, unless the synchronizing circuit of *Figure 10* is used.

5

MPU Instruction Set

The F6802/F6882 has a basic set of 70 instructions. Included are binary and decimal arithmetic, logical, shift, rotate, load, store, conditional or unconditional branch, interrupt, and stack manipulation instructions (refer to *Tables 2* through *6*); special operations are illustrated in *Figure 12*. This instruction set is identical to that of the F6800.

MPU Addressing Modes

There are seven address modes that can be used by a programmer. The addressing modes can be used as a function of both the type of instruction and the coding within the instruction. A summary of the addressing modes for a particular instruction can be found in *Table 7*, along with the associated instruction execution time that is given in machine cycles. With a bus frequency of 1 MHz, these times would be microseconds.

Accumulator (ACCX) Addressing

In accumulator-only addressing, either accumulator A or accumulator B is specified. These are 1-byte instructions.

Immediate Addressing

In immediate addressing, the operand is contained in the second byte of the instruction. The exceptions are LDS and LDX, which have the operand in the second and third bytes of the instruction. The MPU addresses this location when it fetches the immediate instruction for execution. These are 2- or 3-byte instructions.

Direct Addressing

In direct addressing, the address of the operand is contained in the second byte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in the machine (i.e., locations zero through 255). Enhanced execution times are achieved by storing data in these locations. In most configurations, it should be a random access memory. These are 2-byte instructions.

Table 2 F6802/6882 Microprocessor Instruction Set—Alphabetic Sequence

ABA	Add Accumulators
ADC	Add with Carry
ADD	Add
AND	Logical And
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
BCC	Branch if Carry Clear
BCS	Branch if Carry Set
BEQ	Branch if Equal to Zero
BGE	Branch if Greater or Equal Zero
BGT	Branch if Greater than Zero
BHI	Branch if Higher
BIT	Bit Test
BLE	Branch if Less or Equal
BLS	Branch if Lower or Same
BLT	Branch if Less than Zero
BMI	Branch if Minus
BNE	Branch if Not Equal to Zero
BPL	Branch if Plus
BRA	Branch Always
BSR	Branch to Subroutine
BVC	Branch if Overflow Clear
BVS	Branch if Overflow Set
CBA	Compare Accumulators
CLC	Clear Carry
CLI	Clear Interrupt Mask
CLR	Clear
CLV	Clear Overflow
CMP	Compare
COM	Complement
CPX	Compare Index Register
DAA	Decimal Adjust
DEC	Decrement
DES	Decrement Stack Pointer
DEX	Decrement Index Register
EOR	Exclusive OR
INC	Increment
INS	Increment Stack Pointer
INX	Increment Index Register
JMP	Jump
JSR	Jump to Subroutine
LDA	Load Accumulator
LDS	Load Stack Pointer
LDX	Load Index Register
LSR	Logical Shift Right
NEG	Negate
NOP	No Operation
ORA	Inclusive OR Accumulator
PSH	Push Data
PUL	Pull Data
ROL	Rotate Left
ROR	Rotate Right
RTI	Return form Interrupt
RTS	Return from Subroutine
SBA	Subtract Accumulators
SBC	Subtract with Carry
SEC	Set Carry
SEI	Set Interrupt Mask
SEV	Set Overflow
STA	Store Accumulator
STS	Store Stack Register
STX	Store Index Register
SUB	Subtract
SWI	Software Interrupt
TAB	Transfer Accumulators
TAP	Transfer Accumulators to Condition Code Reg.
TBA	Transfer Accumulators
TPA	Transfer Condition Code Reg. to Accumulator
TST	Test
TSX	Transfer Stack Pointer to Index Register
TXS	Transfer Index Register to Stack Pointer
WAI	Wait for Interrupt

Table 3 Accumulator and Memory Instructions

OPERATIONS	MNEMONIC	ADDRESSING MODES: IMMED OP ~ =	DIRECT OP ~ =	INDEX OP ~ =	EXTND OP ~ =	IMPLIED OP ~ =	BOOLEAN/ARITHMETIC OPERATION (All register labels refer to contents)	COND. CODE REG: 5 H	4 I	3 N	2 Z	1 V	0 C
Add	ADDA	8B 2 2	9B 3 2	AB 5 2	BB 4 3		A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB 2 2	DB 3 2	EB 5 2	FB 4 3		B + M → B	↕	•	↕	↕	↕	↕
Add Acmltrs	ABA					1B 2 1	A + B → A	↕	•	↕	↕	↕	↕
Add with Carry	ADCA	89 2 2	99 3 2	A9 5 2	B9 4 3		A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9 2 2	D9 3 2	E9 5 2	F9 4 3		B + M + C → B	↕	•	↕	↕	↕	↕
And	ANDA	84 2 2	94 3 2	A4 5 2	B4 4 3		A · M → A	•	•	↕	↕	R	•
	ANDB	C4 2 2	D4 3 2	E4 5 2	F4 4 3		B · M → B	•	•	↕	↕	R	•
Bit Test	BITA	85 2 2	95 3 2	A5 5 2	B5 4 3		A · M	•	•	↕	↕	R	•
	BITB	C5 2 2	D5 3 2	E5 5 2	F5 4 3		B · M	•	•	↕	↕	R	•
Clear	CLR			6F 7 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 5 2	B1 4 3		A − M	•	•	↕	↕	↕	↕
	CMPB	C1 2 2	D1 3 2	E1 5 2	F1 4 3		B − M	•	•	↕	↕	↕	↕
Compare Acmltrs	CBA					11 2 1	A − B	•	•	↕	↕	↕	↕
Complement, 1s	COM			63 7 2	73 6 3		$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA					43 2 1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB					53 2 1	$\overline{B}$ → B	•	•	↕	↕	R	S
Complement, 2s (Negate)	NEG			60 7 2	70 6 3		00 − M → M	•	•	↕	↕	①	②
	NEGA					40 2 1	00 − A → A	•	•	↕	↕	①	②
	NEGB					50 2 1	00 − B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA					19 2 1	Converts Binary Add. of BCD Characters into BCD Format	•	•	↕	↕	↕	③
Decrement	DEC			6A 7 2	7A 6 3		M − 1 → M	•	•	↕	↕	4	•
	DECA					4A 2 1	A − 1 → A	•	•	↕	↕	4	•
	DECB					5A 2 1	B − 1 → B	•	•	↕	↕	4	•
Exclusive OR	EORA	88 2 2	98 3 2	A8 5 2	B8 4 3		A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8 2 2	D8 3 2	E8 5 2	F8 4 3		B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC			6C 7 2	7C 6 3		M + 1 → M	•	•	↕	↕	⑤	•
	INCA					4C 2 1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB					5C 2 1	B + 1 → B	•	•	↕	↕	⑤	•
Load Acmltr	LDAA	86 2 2	96 3 2	A6 5 2	B6 4 3		M → A	•	•	↕	↕	R	•
	LDAB	C6 2 2	D6 3 2	E6 5 2	F6 4 3		M → B	•	•	↕	↕	R	•
Or, Inclusive	ORAA	8A 2 2	9A 3 2	AA 5 2	BA 4 3		A + M → A	•	•	↕	↕	R	•
	ORAB	CA 2 2	DA 3 2	EA 5 2	FA 4 3		B + M → B	•	•	↕	↕	R	•
Push Data	PSHA					36 4 1	$A \rightarrow M_{SP}$, SP − 1 → SP	•	•	•	•	•	•
	PSHB					37 4 1	$B \rightarrow M_{SP}$, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULA					32 4 1	SP + 1 → SP, M_{SP} → A	•	•	•	•	•	•
	PULB					33 4 1	SP + 1 → SP, M_{SP} → B	•	•	•	•	•	•
Rotate Left	ROL			69 7 2	79 6 3		M, A, B: C b7 b0	•	•	↕	↕	⑥	↕
	ROLA					49 2 1		•	•	↕	↕	⑥	↕
	ROLB					59 2 1		•	•	↕	↕	⑥	↕
Rotate Right	ROR			66 7 2	76 6 3		M, A, B: C b7 b0	•	•	↕	↕	⑥	↕
	RORA					46 2 1		•	•	↕	↕	⑥	↕
	RORB					56 2 1		•	•	↕	↕	⑥	↕
Shift Left, Arithmetic	ASL			68 7 2	78 6 3		M, A, B: C b7 b0 → 0	•	•	↕	↕	⑥	↕
	ASLA					48 2 1		•	•	↕	↕	⑥	↕
	ASLB					58 2 1		•	•	↕	↕	⑥	↕
Shift Right, Arithmetic	ASR			67 7 2	77 6 3		M, A, B: b7 b0 → C	•	•	↕	↕	⑥	↕
	ASRA					47 2 1		•	•	↕	↕	⑥	↕
	ASRB					57 2 1		•	•	↕	↕	⑥	↕
Shift Right, Logic	LSR			64 7 2	74 6 3		M, A, B: 0 → b7 b0 → C	•	•	R	↕	⑥	↕
	LSRA					44 2 1		•	•	R	↕	⑥	↕
	LSRB					54 2 1		•	•	R	↕	⑥	↕
Store Acmltr	STAA		97 4 2	A7 6 2	B7 5 3		A → M	•	•	↕	↕	R	•
	STAB		D7 4 2	E7 6 2	F7 5 3		B → M	•	•	↕	↕	R	•
Subtract	SUBA	80 2 2	90 3 2	A0 5 2	B0 4 3		A − M → A	•	•	↕	↕	↕	↕
	SUBB	C0 2 2	D0 3 2	E0 5 2	F0 4 3		B − M → B	•	•	↕	↕	↕	↕
Subtract Acmltrs	SBA					10 2 1	A − B → A	•	•	↕	↕	↕	↕
Subtr. with Carry	SBCA	82 2 2	92 3 2	A2 5 2	B2 4 3		A − M − C → A	•	•	↕	↕	↕	↕
	SBCB	C2 2 2	D2 3 2	E2 5 2	F2 4 3		B − M − C → B	•	•	↕	↕	↕	↕
Transfer Acmltrs	TAB					16 2 1	A → B	•	•	↕	↕	R	•
	TBA					17 2 1	B → A	•	•	↕	↕	R	•
Test, Zero or Minus	TST			6D 7 2	7D 6 3		M − 00	•	•	↕	↕	R	R
	TSTA					4D 2 1	A − 00	•	•	↕	↕	R	R
	TSTB					5D 2 1	B − 00	•	•	↕	↕	R	R
								H	I	N	Z	V	C

Note
Accumulator addressing mode instructions are included in the column for implied addressing.

Legend

OP	Operation code (hexadecimal)
~	Number of MPU cycles
=	Number of program bytes
+	Arithmetic plus
−	Arithmetic minus
·	Boolean AND
M_{SP}	Contents of memory location point to be stack pointer
+	Boolean inclusive-OR
−	Boolean exclusive-OR
M	Complement of M
→	Transfer into
0	Bit = zero
00	Byte = zero

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 from bit 7
R	Reset always
S	Set always
:	Test and set if true, cleared otherwise
•	Not affected

Table 4 Index Register and Stack Manipulation Instructions

		IMMED			DIRECT			INDEX			EXTND			IMPLIED				COND. CODE REG.					
																		5	4	3	2	1	0
POINTER OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#	BOOLEAN/ARITHMETIC OPERATION	H	I	N	Z	V	C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	6	2	BC	5	3				$X_H - M, X_L - M+1$	•	•	⑦	↕	⑧	•
Decrement Index Reg	DEX													09	4	1	$X - 1 \rightarrow X$	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	4	1	$SP - 1 \rightarrow SP$	•	•	•	•	•	•
Increment Index Reg	INX													08	4	1	$X + 1 \rightarrow X$	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	4	1	$SP + 1 \rightarrow SP$	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	6	2	FE	5	3				$M - X_H, M+1 \rightarrow X_L$	•	•	⑨	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	6	2	BE	5	3				$M - SP_H, M+1 \rightarrow SP_L$	•	•	⑨	↕	R	•
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3				$X_H \rightarrow M, X_L \rightarrow M+1$	•	•	⑨	↕	R	•
Store Stack Pntr	STS				9F	5	2	AF	7	2	BF	6	3				$SP_H \rightarrow M, SP_L \rightarrow M+1$	•	•	⑨	↕	R	•
Indx Reg → Stack Pntr	TXS													35	4	1	$X - 1 \rightarrow SP$	•	•	•	•	•	•
Stack Pntr → Indx Reg	TSX													30	4	1	$SP + 1 \rightarrow X$	•	•	•	•	•	•

Table 5 Jump and Branch Instructions

		RELATIVE			INDEX			EXTND			IMPLIED				COND. CODE REG.						
															5	4	3	2	1	0	
OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	BRANCH TEST	H	I	N	Z	V	C	
Branch Always	BRA	20	4	2										None	•	•	•	•	•	•	
Branch If Carry Clear	BCC	24	4	2										C = 0	•	•	•	•	•	•	
Branch If Carry Set	BCS	25	4	2										C = 1	•	•	•	•	•	•	
Branch If = Zero	BEQ	27	4	2										Z = 1	•	•	•	•	•	•	
Branch If ≥ Zero	BGE	2C	4	2										$N \oplus V = 0$	•	•	•	•	•	•	
Branch If > Zero	BGT	2E	4	2										$Z + (N \oplus V) = 0$	•	•	•	•	•	•	
Branch If Higher	BHI	22	4	2										C + Z = 0	•	•	•	•	•	•	
Branch If ≤ Zero	BLE	2F	4	2										$Z + (N \oplus V) = 1$	•	•	•	•	•	•	
Branch If Lower Or Same	BLS	23	4	2										C + Z = 1	•	•	•	•	•	•	
Branch If < Zero	BLT	2D	4	2										$N \oplus V = 1$	•	•	•	•	•	•	
Branch If Minus	BMI	2B	4	2										N = 1	•	•	•	•	•	•	
Branch If Not Equal Zero	BNE	26	4	2										Z = 0	•	•	•	•	•	•	
Branch If Overflow Clear	BVC	28	4	2										V = 0	•	•	•	•	•	•	
Branch If Overflow Set	BVS	29	4	2										V = 1	•	•	•	•	•	•	
Branch If Plus	BPL	2A	4	2										N = 0	•	•	•	•	•	•	
Branch To Subroutine	BSR	8D	8	2											•	•	•	•	•	•	
Jump	JMP				6E	4	2	7E	3	3				See Special Operations	•	•	•	•	•	•	
Jump To Subroutine	JSR				AD	8	2	BD	9	3					•	•	•	•	•	•	
No Operation	NOP										01	2	1	Advances Prog. Cntr. Only	•	•	•	•	•	•	
Return From Interrupt	RTI										3B	10	1		——	⑩	——				
Return From Subroutine	RTS										39	5	1		•	•	•	•	•	•	
Software Interrupt	SWI										3F	12	1	See Special Operations	•	•	•	•	•	•	
Wait for Interrupt*	WAI										3E	9	1		•	⑪	•	•	•	•	

*WAI puts address bus, R/$\overline{W}$, and data bus in the 3-state mode while VMA is held low.

Table 6 Condition Code Register Manipulation Instructions

OPERATIONS	MNEMONIC	IMPLIED OP	~	#	BOOLEAN OPERATION	COND. CODE REG. 5 H	4 I	3 N	2 Z	1 V	0 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	•
Acmltr A → CCR	TAP	06	2	1	A → CCR	———	———	———	⑫	———	———
CCR → Acmltr A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

Condition Code Register Notes (Bit set if test is true and cleared otherwise)

1 (Bit V) Test: Result = 10000000?
2 (Bit C) Test: Result = 00000000?
3 (Bit C) Test: Decimal value of most significant BCD character greater than nine? (Not cleared if previously set.)
4 (Bit V) Test: Operand = 10000000 prior to execution?
5 (Bit V) Test: Operand = 01111111 prior to execution?
6 (Bit V) Test: Set equal to result of N ⊕ C after shift has occurred.
7 (Bit N) Test: Sign bit of most significant (MS) byte = 1?
8 (Bit V) Test: 2's complement overflow from subtraction of MS bytes?
9 (Bit N) Test: Result less than zero? (Bit 15 = 1)
10 (All) Load condition code register from stack (see Figure 10)
11 (Bit I) Set when interrupt occurs. If previously set, a non-maskable interrupt is required to exit the wait state.
12 (All) Set according to the contents of accumulator A.

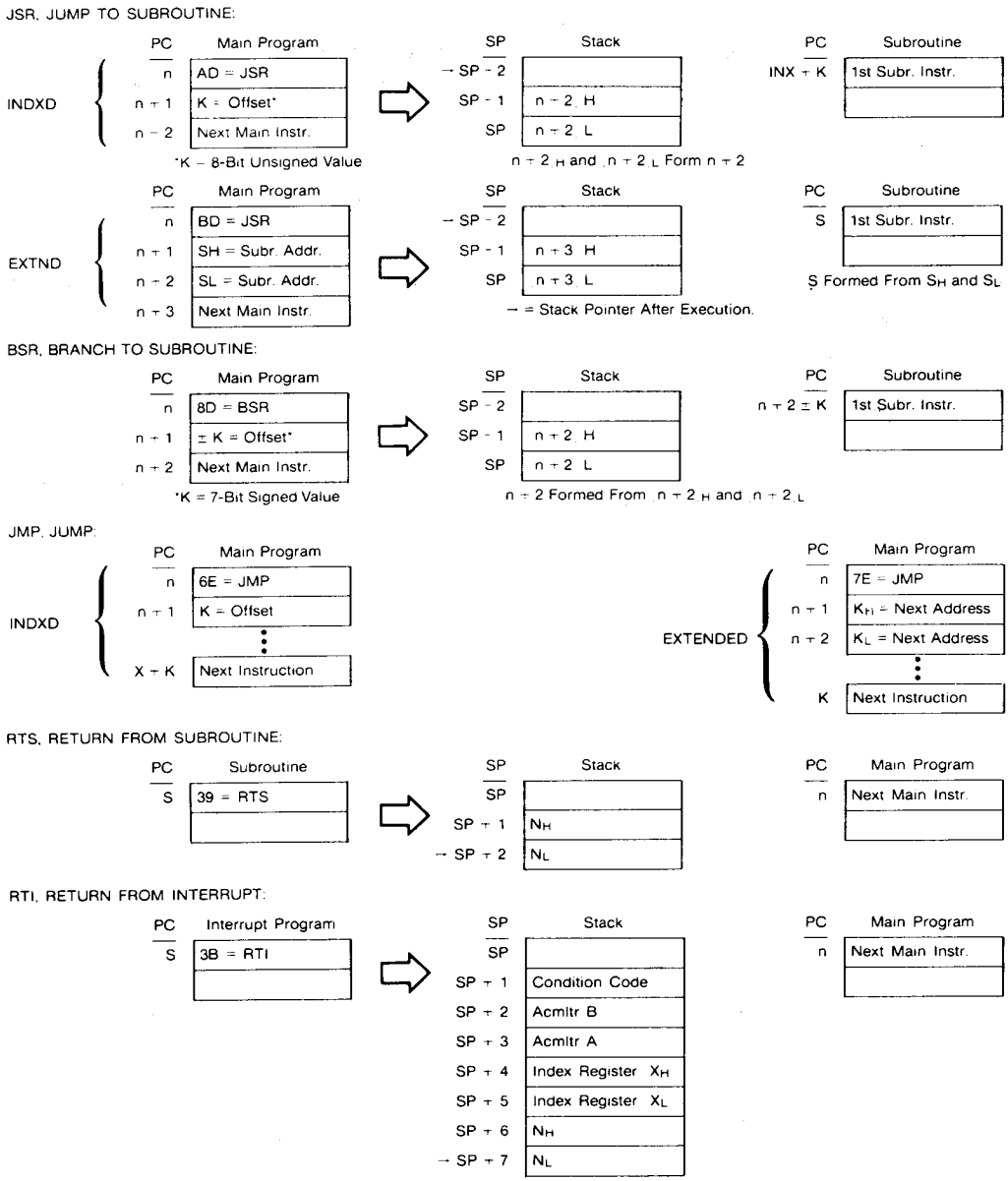
5

Table 7 Instruction Addressing Modes and Associated Execution Times (Times in Machine Cycles)

	(Dual Operand)	ACCX	Immediate	Direct	Extended	Indexed	Implied	Relative		(Dual Operand)	ACCX	Immediate	Direct	Extended	Indexed	Implied
ABA		•	•	•	•	•	2	•	INC		2	•	•	6	7	•
ADC	x	•	2	3	4	5	•	•	INS		•	•	•	•	•	4
ADD	x	•	2	3	4	5	•	•	INX		•	•	•	•	•	4
AND	x	•	2	3	4	5	•	•	JMP		•	•	•	3	4	•
ASL		2	•	•	6	7	•	•	JSR		•	•	•	9	8	•
ASR		2	•	•	6	7	•	•	LDA	x	•	2	3	4	5	•
BCC		•	•	•	•	•	•	4	LDS		•	3	4	5	6	•
BCS		•	•	•	•	•	•	4	LDX		•	3	4	5	6	•
BEA		•	•	•	•	•	•	4	LSR		2	•	•	6	7	•
BGE		•	•	•	•	•	•	4	NEG		2	•	•	6	7	•
BGT		•	•	•	•	•	•	4	NOP		•	•	•	•	•	2
BHI		•	•	•	•	•	•	4	ORA	x	•	2	3	4	5	•
BIT	x	•	2	3	4	5	•	•	PSH		•	•	•	•	•	4
BLE		•	•	•	•	•	•	4	PUL		•	•	•	•	•	4
BLS		•	•	•	•	•	•	4	ROL		2	•	•	6	7	•
BLT		•	•	•	•	•	•	4	ROR		2	•	•	6	7	•
BMI		•	•	•	•	•	•	4	RTI		•	•	•	•	•	10
BNE		•	•	•	•	•	•	4	RTS		•	•	•	•	•	5
BPL		•	•	•	•	•	•	4	SBA		•	•	•	•	•	2
BRA		•	•	•	•	•	•	4	SBC	x	•	2	3	4	5	•
BSR		•	•	•	•	•	•	8	SEC		•	•	•	•	•	2
BVC		•	•	•	•	•	•	4	SEI		•	•	•	•	•	2
BVS		•	•	•	•	•	•	4	SEV		•	•	•	•	•	2
CBA		•	•	•	•	•	2	•	STA	x	•	•	4	5	6	•
CLC		•	•	•	•	•	2	•	STS		•	•	5	6	7	•
CLI		•	•	•	•	•	2	•	STX		•	•	5	6	7	•
CLR		2	•	•	6	7	•	•	SUB	x	•	2	3	4	5	•
CLV		•	•	•	•	•	2	•	SWI		•	•	•	•	•	12
CMP	x	•	2	3	4	5	•	•	TAB		•	•	•	•	•	2
COM		2	•	•	6	7	•	•	TAP		•	•	•	•	•	2
CPX		•	3	4	5	6	•	•	TBA		•	•	•	•	•	2
DAA		•	•	•	•	•	2	•	TPA		•	•	•	•	•	2
DEC		2	•	•	6	7	•	•	TST		2	•	•	6	7	•
DES		•	•	•	•	•	4	•	TSX		•	•	•	•	•	4
DEX		•	•	•	•	•	4	•	TSX		•	•	•	•	•	4
EOR	x	•	2	3	4	5	•	•	WAI		•	•	•	•	•	9

Note
Interrupt time is 12 cycles from the end of the instruction being executed, except following a WAI instruction, when it is four cycles.

Fig. 12 Special Operations



Extended Addressing
In extended addressing, the address contained in the second byte of the instruction is used as the higher 8 bits of the address of the operand. The third byte of the instruction is used as the lower 8 bits of the address for the operand. This is an absolute address in memory. These are 3-byte instructions.

Indexed Addressing
In indexed addressing, the address contained in the second byte of the instruction is added to the index register's lowest 8 bits in the MPU. The carry is then added to the higher order 8 bits of the index register. This result is then used to address memory. The modified address is held in a temporary address register so there is no change to the index register. These are 2-byte instructions.

Implied Addressing
In the implied addressing mode, the instructions give the address (i.e., stack pointer, index register, etc.). These are 1-byte instructions.

Relative Addressing
In relative addressing, the address contained in the second byte of the instruction is added to the program counter's lowest 8 bits plus two. The carry or borrow is then added to the higher 8 bits. This allows the user to address data within a range of – 125 to + 129 bytes of the present instruction. These are 2-byte instructions.

Summary of Cycle-by-Cycle Operation
Table 8 provides a detailed description of the information present on the address bus, data bus, valid memory address (VMA) line, and the read/write (R/$\overline{W}$) line during each cycle for each instruction.

This information is useful in comparing actual with expected results during debug of both software and hardware as the control program is executed. The information is categorized 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.)

Output Delay

Figures 13 and *14* illustrate typical output delays versus capacitance loading.

Fig. 13 Typical Data Bus Output Delay vs. Capacitive Loading

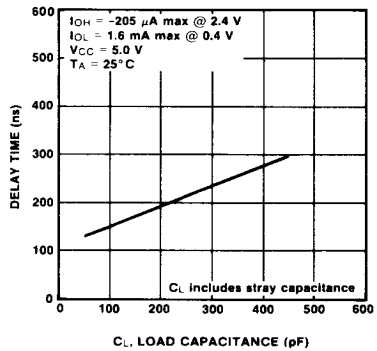


Fig. 14 Typical Read/$\overline{\text{Write}}$, VMA and Address Output Delay vs. Capacitive Loading

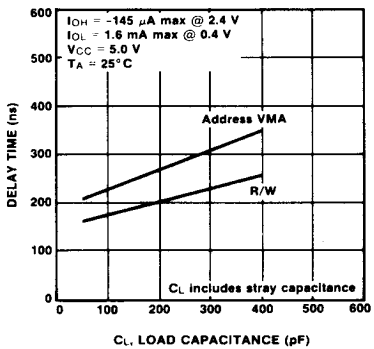


Table 8 Operation Summary

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/W Line	Data Bus
Immediate						
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1 2	1 1	Op Code Address Op Code Address + 1	1 1	Op Code Operand Data
CPX LDS LDX	3	1 2 3	1 1 1	Op Code Address Op Code Address + 1 Op Code Address + 2	1 1 1	Op Code Operand Data (High Order Byte) Operand Data (Low Order Byte)
Direct						
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1 2 3	1 1 1	Op Code Address Op Code Address + 1 Address of Operand	1 1 1	Op Code Address of Operand Operand Data
CPX LDS LDX	4	1 2 3 4	1 1 1 1	Op Code Address Op Code Address + 1 Address of Operand Operand Address + 1	1 1 1 1	Op Code Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte)
STA	4	1 2 3 4	1 1 0 1	Op Code Address Op Code Address + 1 Destination Address Destination Address	1 1 1 0	Op Code Destination Address Irrelevant Data (Note 1) Data from Accumulator
STS STX	5	1 2 3 4 5	1 1 0 1 1	Op Code Address Op Code Address + 1 Address of Operand Address of Operand Address of Operand + 1	1 1 1 0 0	Op Code Address of Operand Irrelevant Data (Note 1) Register Data (High Order Byte) Register Data (Low Order Byte)
Indexed						
JMP	4	1 2 3 4	1 1 0 0	Op Code Address Op Code Address + 1 Index Register Index Register Plus Offset (w/o Carry)	1 1 1 1	Op Code Offset Irrelevant Data (Note 1) Irrelevant Data (Note 1)
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	5	1 2 3 4 5	1 1 0 0 1	Op Code Address Op Code Address + 1 Index Register Index Register Plus Offset (w/o Carry) Index Register Plus Offset	1 1 1 1 1	Op Code Offset Irrelevant Data (Note 1) Irrelevant Data (Note 1) Operand Data
CPX LDS LDX	6	1 2 3 4 5 6	1 1 0 0 1 1	Op Code Address Op Code Address + 1 Index Register Index Register Plus Offset (w/o Carry) Index Register Plus Offset Index Register Plus Offset + 1	1 1 1 1 1 1	Op Code Offset Irrelevant Data (Note 1) Irrelevant Data (Note 1) Operand Data (High Order Byte) Operand Data (Low Order Byte)

Table 8 Operation Summary (Cont.)

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/W Line	Data Bus
Indexed (Continued)						
STA	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	0	Index Register Plus Offset	1	Irrelevant Data (Note 1)
		6	1	Index Register Plus Offset	0	Operand Data
ASL LSR ASR NEG CLR ROL COM ROR DEC TST INC	7	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	1	Index Register Plus Offset	1	Current Operand Data
		6	0	Index Register Plus Offset	1	Irrelevant Data (Note 1)
		7	1/0 (Note 3)	Index Register Plus Offset	0	New Operand Data (Note 3)
STS STX	7	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	0	Index Register Plus Offset	1	Irrelevant Data (Note 1)
		6	1	Index Register Plus Offset	0	Operand Data (High Order Byte)
		7	1	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
JSR	8	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer	0	Return Address (High Order Byte)
		5	1	Stack Pointer — 1	0	Return Address (Low Order Byte)
		6	0	Stack Pointer — 2	1	Irrelevant Data (Note 1)
		7	0	Index Register	1	Irrelevant Data (Note 1)
		8	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
Extended						
JMP	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Jump Address (High Order Byte)
		3	1	Op Code Address + 2	1	Jump Address (LLow Order Byte)
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	1	Address of Operand	1	Operand Data
CPX LDS LDX	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	1	Address of Operand	1	Operand Data (High Order Byte)
		5	1	Address of Operand + 1	1	Operand Data (High Order Byte)
STA A STA B	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Destination Address (High Order Byte)
		3	1	Op Code Address + 2	1	Destination Address (Low Order Byte)
		4	0	Operand Destination Address	1	Irrelevant Data (Note 1)
		5	1	Operand Destination Address	0	Data from Accumulator
ASL LSR ASR NEG CLR ROL COM ROR DEC TST INC	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	1	Address of Operand	1	Current Operand Data
		5	0	Address of Operand	1	Irrelevant Data (Note 1)
		6	1/0 (Note 3)	Address of Operand	0	New Operand Data (Note 3)

Table 8 Operation Summary (Cont.)

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/W Line	Data Bus
Extended (Continued)						
STS STX	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	0	Address of Operand	1	Irrelevant Data (Note 1)
		5	1	Address of Operand	0	Operand Data (High Order Byte)
		6	1	Address of Operand + 1	0	Operand Data (Low Order Byte)
JSR	9	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Subroutine (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
		4	1	Subroutine Starting Address	1	Op Code of Next Instruction
		5	1	Stack Pointer	0	Return Address (Low Order Byte)
		6	1	Stack Pointer — 1	0	Return Address (High Order Byte)
		7	0	Stack Pointer — 2	1	Irrelevant Data (Note 1
		8	0	Op Code Address + 2	1	Irrelevant Data (Note 1)
		9	1	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
Inherent						
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
DES DEX INS INX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Previous Register Contents	1	Irrelevant Data (Note 1)
		4	0	New Register Contents	1	Irrelevant Data (Note 1)
PSH	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	1	Stack Pointer	0	Accumulator Data
		4	0	Stack Pointer — 1	1	Accumulator Data
PUL	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer + 1	1	Operand Data from Stack
TSX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	0	New Index Register	1	Irrelevant Data (Note 1)
TXS	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Index Register	1	Irrelevant Data
		4	0	New Stack Pointer	1	Irrelevant Data
RTS	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Irrelevant Data (Note 2)
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer + 1	1	Address of Next Instruction (High Order Byte)
		5	1	Stack Pointer + 2	1	Address of Next Instruction (Low Order Byte)

Table 8 Operation Summary (Cont.)

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/W Line	Data Bus
Inherent (Continued)						
WAI	9	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	1	Stack Pointer	0	Return Address (Low Order Byte)
		4	1	Stack Pointer — 1	0	Return Address (High Order Byte)
		5	1	Stack Pointer — 2	0	Index Register (Low Order Byte)
		6	1	Stack Pointer — 3	0	Index Register (High Order Byte)
		7	1	Stack Pointer — 4	0	Contents of Accumulator A
		8	1	Stack Pointer — 5	0	Contents of Accumulator B
		9	1	Stack Pointer — 6 (Note 4)	1	Contents of Cond. Code Register
RTI	10	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Irrelevant Data (Note 2)
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer + 1	1	Contents of Cond. Code Register from Stack
		5	1	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	1	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	1	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	1	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	1	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	1	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Irrelevant Data (Note 1)
		3	1	Stack Pointer	0	Return Address (Low Order Byte)
		4	1	Stack Pointer — 1	0	Return Address (High Order Byte)
		5	1	Stack Pointer — 2	0	Index Register (Low Order Byte)
		6	1	Stack Pointer — 3	0	Index Register (High Order Byte)
		7	1	Stack Pointer — 4	0	Contents of Accumulator A
		8	1	Stack Pointer — 5	0	Contents of Accumulator B
		9	1	Stack Pointer — 6	0	Contents of Cond. Code Register
		10	0	Stack Pointer — 7	1	Irrelevant Data (Note 1)
		11	1	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	1	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)
Relative						
BCC BHI BNE BCS BLE BPL BEQ BLS BRA BGE BLT BVC BGT BMI BVS	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Branch Offset
		3	0	Op Code Address + 2	1	Irrelevant Data (Note 1)
		4	0	Branch Address	1	Irrelevant Data (Note 1)
BSR	8	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Branch Offset
		3	0	Return Address of Main Program	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer	0	Return Address (Low Order Byte)
		5	1	Stack Pointer — 1	0	Return Address (High Order Byte)
		6	0	Stack Pointer — 2	1	Irrelevant Data (Note 1)
		7	0	Return Address of Main Program	1	Irrelevant Data (Note 1)
		8	0	Subroutine Address	1	Irrelevant Data (Note 1)

Notes:

1. If device that is addressed during this cycle uses VMA, the data bus goes to the high-impedance 3-state condition. Depending on bus capacitance, data from the previous cycle may be retained on the data bus.
2. Data is ignored by the MPU.
3. For TST, VMA = 0 and operand data does not change.
4. Most significant byte of address bus = most significant byte of address of BSR instruction, and least significant byte of address bus = least significant byte of subroutine address.

DC Characteristics

Table 9 contains the dc characteristics of the F6802/F6882.

Absolute Maximum Ratings

These are stress ratings only, and functional operation at these ratings, or under any conditions above those indicated in this data sheet, is not implied. Exposure to the absolute maximum rating conditions for extended periods of time may affect device reliability, and exposure to stresses greater than those listed may cause permanent damage to the device.

Voltage of any Pin Relative to GND	−0.3 V,	+7.0 V
Storage Temperature	−55°C,	+150°C
Power Dissipation	1.5 W	
Thermal Resistance, θ_{JA}		
(Plastic Package)	1W	
(CER-DIP Package)	55°C/W	

Table 9 DC Characteristics (V_{CC} = 5.0 V ± 5%, V_{SS} = 0, T_A = 0 to 70°C unless otherwise noted)

Symbol	Characteristic	Min	Typ	Max	Unit	Condition
V_{IH}	Input High Voltage				Vdc	
	Logic EXtal	V_{SS} + 2.0	—	V_{CC}		
	Logic $\overline{\text{Reset}}$	V_{SS} + 4.0	—	V_{CC}		
V_{IL}	Input Low Voltage				Vdc	
	Logic Extal, $\overline{\text{Reset}}$	V_{SS} − 0.3	—	V_{SS} + 0.8		
I_{IN}	Input Leakage Current				μAdc	
	Logic	—	1.0	2.5		V_{IN} = 0 to 5.25 V, V_{CC} = Max
V_{OH}	Output High Voltage				Vdc	
	D_0-D_7	V_{SS} + 2.4	—	—		I_{LOAD} = −206 μAdc, V_{CC} = Min
	A_0-A_{15}, R/$\overline{W}$, VMA, E	V_{SS} + 2.4	—	—		I_{LOAD} = −145 μAdc, V_{CC} = Min
	BA	V_{SS} + 2.4	—	—		I_{LOAD} = −100 μAdc, V_{CC} = Min
V_{OL}	Output Low Voltage	—	—	V_{SS} + 0.4	Vdc	I_{LOAD} = 1.0 mAdc, V_{CC} = Min
P_D*	Power Dissipation	—	0.600	1.2	W	
	V_{CC} Standby				Vdc	
V_{SBB}	Power Down	4.0	—	5.25		
V_{SB}	Power Up	4.75	—	5.25		
I_{SBB}	Standby Current				mA	
	F6802	—	—	8.0		
	F6882	—	—	3.0		
	Capacitance				pF	
C_{IN}	D_0-D_7	—	—	12.5		
	Logic Inputs EXtal	—	6	10		V_{IN} = 0, T_A = 25°C, f = 1.0 MHz
C_{OUT}	A_0-A_{15}, R/$\overline{W}$, VMA	—	—	12		

*In power-down mode, maximum power dissipation is less than 42 mW.
Capacitances are periodically sampled rather than 100% tested.

Timing Characteristics

Tables 10 and 11 contain timing characteristics information.

Table 10 Frequency Characteristics

Symbol	Characteristic	F6802 Min	F6802 Max	F680A02 Min	F680A02 Max	Unit
f_o	Frequency of Operation	0.1	1.0	0.1	1.5	MHz
f_{XTAL}	Crystal Frequency	1.0	4.0	1.0	1.5	MHz
$4xf_o$	External Oscillator Frequency	0.4	4.0	0.4	6.0	MHz
t_{CYC}	Cycle Time	1.0	10	0.666	10	μs
t_{PWEH}	Clock Pulse Width E High	450	9500	280	9700	ns
t_{PWEL}	Clock Pulse Width E Low	450	5000	280	5000	ns
t_R, t_F	Fall Time	—	25	—	25	ns
t_{rc}	Crystal Oscillator Startup Time	100	—	100	—	ms

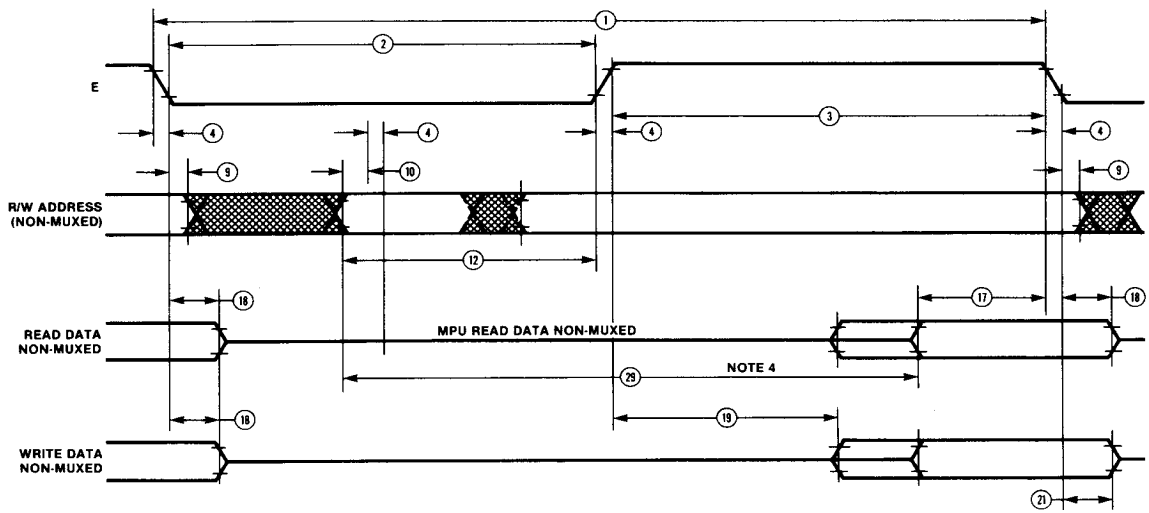
Table 11 Read/Write Timing

Symbol	Characteristic	F6802 Min	F6802 Max	F680A02 Min	F680A02 Max	Unit
t_{AD}	Address Delay	—	270	—	220	ns
t_{AV1}	Address Delay (Internal RAM	—	270	—	240	ns
t_{AV2}	Read Access Time Useable by Peripheral @ 1 MHz $t_{ACC} = t_{CYC} - t_{AD} + t_{DSR} + t_F$	—	605	—	310	
t_{DSR}	Data Setup Time (Read)	100	—	70	—	ns
t_{DHR}	Input Data Hold Time	10	—	10	—	ns
t_{DHW}	Output Data Hold Time	30	—	20	—	ns
t_{AH}	Address Hold Time (Address, R/W, VMA)	20	—	20	—	ns
t_{DDW}	Data Delay Time (Write)	—	225	—	170	ns
	Processor Controls					
t_{PCS}	Processor Control Setup Time	200	—	140	—	
t_{PCR}, t_{PCF}	Processor Control Rise and Fall Time (Does Not Apply to RESET) (Measured between 0.8 V and 2.0 V)	—	100	—	100	ns

Note
If programs are not executed from on-board RAM, TAV1 applies. If programs are to be stored and executed from on-board RAM, TAV2 applies. For normal data storage in the on-board RAM, this extended delay does not apply. Programs cannot be executed from on-board RAM when using A parts (F68A02, F68A08). One-board RAM can be used for data storage with all parts.

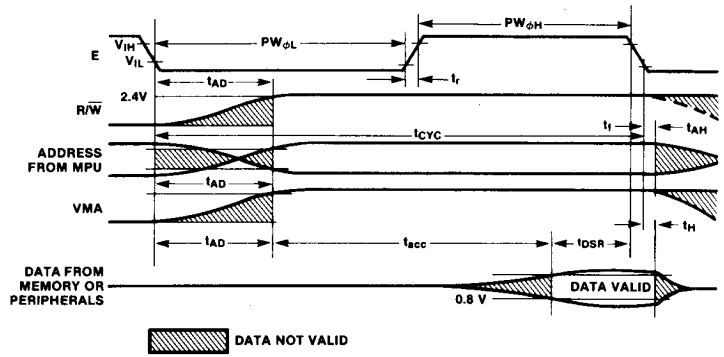
Bus Timing Characteristics

Symbol	Parameter	F6802NS F6802 F6808 Min	F6802NS F6802 F6808 Max	F68A02 F68A08 Min	F68A02 F68A08 Max	Unit
t_{CYC}	(1) Cycle Time	1.0	10	0.667	10	μs
PW_{EL}	(2) Pulse Width, E Low	450	5000	280	5000	ns
PW_{EH}	(3) Pulse Width, E High	450	9500	280	9700	ns
t_r, t_f	(4) Clock Rise and Fall Time	—	25	—	25	ns
t_{AH}	(9) Address Hold Time	20	—	20	—	ns
t_{AV1} t_{AV2}	(12) Non-Muxed Address Valid Time to E	160 —	— 270	100 —	— —	ns
t_{DSR}	(17) Read Data Setup Time	100	—	70	—	ns
t_{DHR}	(18) Read Data Hold Time	10	—	10	—	ns
t_{DDW}	(19) Write Data Delay Time	—	225	—	170	ns
t_{DHW}	(21) Wrote Data Hold Time	30	—	20	—	ns
t_{ACC}	(29) Usable Access Time (See Note 4)	605	—	310	—	ns



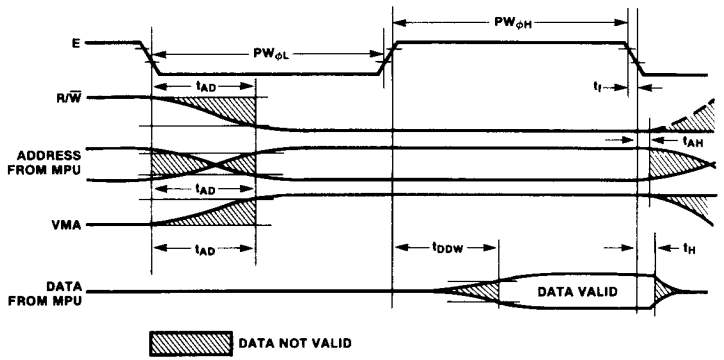
Notes
1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise noted.
3. All electricals shown for the F6802 apply to the F6802NS and F6808, unless otherwise noted.
4. Usable access time is computed by 12 + 3 + 4 – 17.

Fig. 15 Read Data from Memory or Peripherals



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

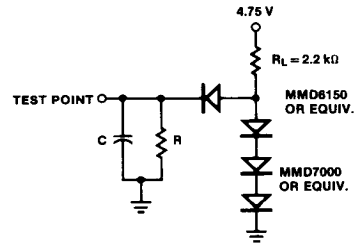
Fig. 16 Write Data in Memory or Peripherals



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

Fig. 17 Bus Timing Test Load

C = 130 pF FOR D_0-D_7, E
= 90 pF FOR A_0-A_{15}, R/$\overline{W}$, AND VMA
= 30 pF FOR BA
R = 11.7 kΩ FOR D_0-D_7, E
= 16.5 kΩ FOR A_0-A_{15}, R/$\overline{W}$, AND VMA
= 24 kΩ FOR BA



Ordering Information

Order Code	Speed	Temperature Range
F6802P, S	1.0 MHz	0°C to +70°C
F6882P,S	1.0 MHz	0°C to +70°C
F6802CP, CS	1.0 MHz	−40°C to +85°C
F68A02P, S	1.5 MHz	0°C to +70°C
F68A02CP, CS	1.5 MHz	−40°C to +85°C

P = Plastic package
S = Ceramic package