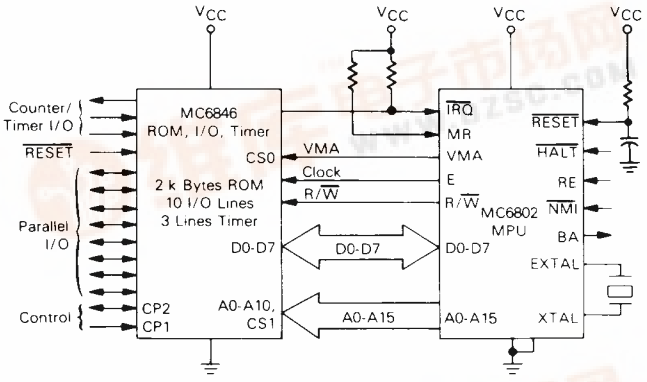
Microprocessor With Clock and Optional RAM

The MC6802 is a monolithic 8-bit microprocessor that contains all the registers and accumulators of the present MC6800 plus an internal clock oscillator and driver on the same chip. In addition, the MC6802 has 128 bytes of on-board RAM located at hex addresses $0000 to $007F. The first 32 bytes of RAM, at hex addresses $0000 to $001F, may be retained in a low power mode by utilizing V_{CC} standby; thus, facilitating memory retention during a power-down situation.

The MC6802 is completely software compatible with the MC6800 as well as the entire M6800 family of parts. Hence, the MC6802 is expandable to 64K words.

- On-Chip Clock Circuit
- 128×8 Bit On-Chip RAM
- 32 Bytes of RAM are Retainable
- Software-Compatible with the MC6800
- Expandable to 64K Words
- Standard TTL-Compatible Inputs and Outputs
- 8-Bit Word Size
- 16-Bit Memory Addressing
- Interrupt Capability

TYPICAL MICROCOMPUTER



This block diagram shows a typical cost effective microcomputer. The MPU is the center of the microcomputer system and is shown in a minimum system interfacing with a ROM combination chip. It is not intended that this system be limited to this function but that it be expandable with other parts in the M6800 Microcomputer family.

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

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC6802, MC680A02, MC680B02 MC6802C, MC680A02C	T_A	 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

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

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Average Thermal Resistance (Junction to Ambient) Plastic	θ_{JA}	100	°C/W

POWER CONSIDERATIONS

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

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

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{PORT}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
P_{PORT} = Port Power Dissipation, Watts — User Determined

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

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

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations (1) and (2) for K gives:

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

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

DC ELECTRICAL CHARACTERISTICS (V_{DD} = +5.0 Vdc ± 0.5%, V_{SS} = 0, T_A = 0 to 70°C, unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	Logic, EXTAL RESET	V_{IH}	V_{SS} + 2.0 V_{SS} + 4.0	— —	V_{CC} V_{CC}	V
Input Low Voltage	Logic, EXTAL, RESET	V_{IL}	V_{SS} − 0.3	—	V_{SS} + 0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V, V_{DD} = max)	Logic	I_{in}	—	1.0	2.5	μA
Output High Voltage (I_{Load} = −205 μA, V_{CC} = min) (I_{Load} = −145 μA, V_{CC} = min) (I_{Load} = −100 μA, V_{CC} = min)	 D0-D7 A0-A15, R/W, VMA, E BA	V_{OH}	 V_{SS} + 2.4 V_{SS} + 2.4 V_{SS} + 2.4	 — — —	 — — —	V
Output Low Voltage (I_{Load} = 1.6 mA, V_{CC} = min)		V_{OL}	—	—	V_{SS} + 0.4	V
Internal Power Dissipation (Measured at T_A = 0°C)		P_{INT}	—	0.750	1.0	W
V_{DD} Standby	Power Down Power Up	V_{SBB} V_{SB}	4.0 4.75	— —	5.25 5.25	V
Standby Current		I_{SBB}	—	—	8.0	mA
Capacitance # (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7 Logic Inputs, EXTAL A0-A15, R/W, VMA	C_{in} C_{out}	— — —	10 6.5 —	12.5 10 12	pF

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

3

CONTROL TIMING (V_{CC} = 5.0 V ± 5%, V_{SS} = 0, T_A = T_L to T_H), unless otherwise noted)

Characteristic	Symbol	MC6802		MC68A02		MC68B02		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	0.1	1.0	0.1	1.5	0.1	2.0	MHz
Crystal Frequency	f_{XTAL}	1.0	4.0	1.0	6.0	1.0	8.0	MHz
External Oscillator Frequency	$4xf_o$	0.4	4.0	0.4	6.0	0.4	8.0	MHz
Crystal Oscillator Start Up Time	t_{rc}	100	—	100	—	100	—	ms
Processor Controls (HALT, MR, RE, RESET, IRQ NMI) Processor Control Setup Time Processor Control Rise and Fall Time (Does Not Apply to RESET)	 t_{PCS} t_{PCr}, t_{PCf}	 200 —	 — 100	 140 —	 — 100	 110 —	 — 100	ns

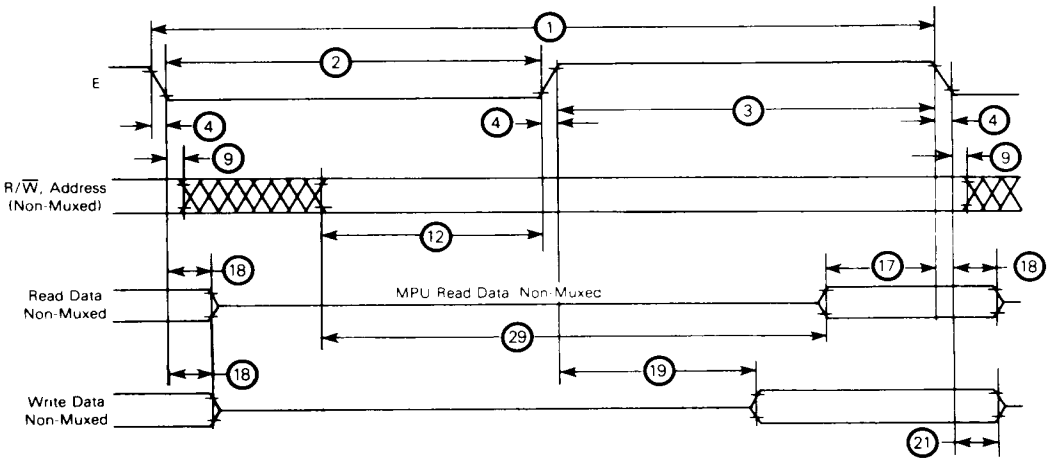
BUS TIMING CHARACTERISTICS

Ident. Number	Characteristic	Symbol	MC6802		MC68A02		MC68B02		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.667	10	0.5	10	µs
2	Pulse Width, E Low	PW_{EL}	450	5000	280	5000	210	5000	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9700	220	9700	ns
4	Clock Rise and Fall Time	t_r, t_f	—	25	—	25	—	25	ns
9	Address Hold Time*	t_{AH}	20	—	20	—	20	—	ns
12	Non-Muxed Address Valid Time to E (see Note 4)	t_{AV1} t_{AV2}	160 —	— 270	100 —	— —	50 —	— —	ns
17	Read Data Setup Time	t_{DSR}	100	—	70	—	60	—	ns
18	Read Data Hold Time	t_{DHR}	10	—	10	—	10	—	ns
19	Write Data Delay Time	t_{DDW}	—	225	—	170	—	160	ns
21	Write Data Hold Time*	t_{DHW}	30	—	20	—	20	—	ns
29	Usable Access Time (see Note 4)	t_{ACC}	535	—	335	—	235	—	ns

*Address and data hold times are periodically tested rather than 100% tested.

3

FIGURE 2 — BUS TIMING



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. Usable access time is computed by: 12 + 3 + 4 − 17.
4. 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 and B parts (MC68A02, MC68B02). On-board RAM can be used for data storage with all parts.
5. All electrical and control characteristics are referenced from: $T_L = 0°C$ minimum and $T_H = 70°C$ maximum.

FIGURE 3 — BUS TIMING TEST LOAD

C = 130 pF for D0-D7, E
= 90 pF for A0-A15, R/$\overline{W}$, and VMA
= 30 pF for BA
R = 11.7 kΩ for D0-D7, E
= 16.5 kΩ for A0-A15, R/$\overline{W}$, and VMA
= 24 kΩ for BA

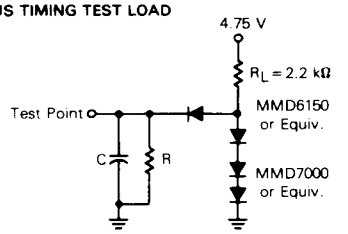


FIGURE 4 — TYPICAL DATA BUS OUTPUT DELAY versus CAPACITIVE LOADING

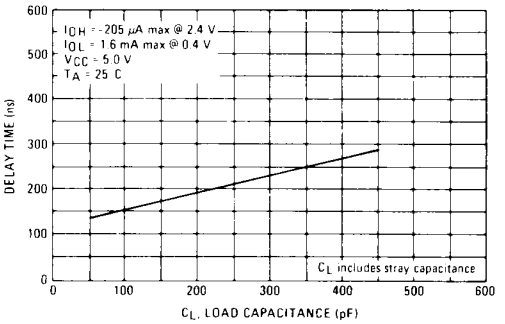


FIGURE 5 — TYPICAL READ/WRITE, VMA AND ADDRESS OUTPUT DELAY versus CAPACITIVE LOADING

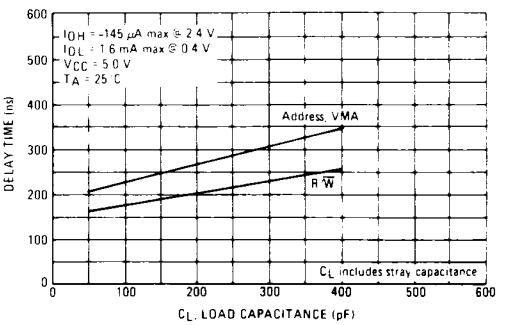
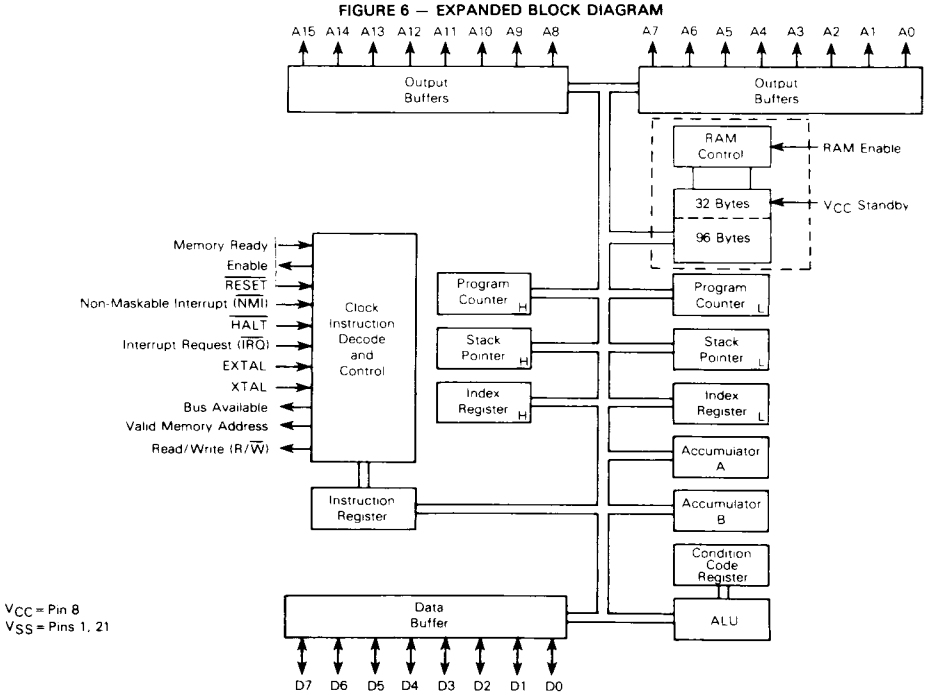


FIGURE 6 — EXPANDED BLOCK DIAGRAM



MPU REGISTERS

A general block diagram of the MC6802 is shown in Figure 1. As shown, the number and configuration of the registers are the same as for the MC6800. The 128 × 8-bit RAM* has been added to the basic MPU. The first 32 bytes can be retained during powerup and power-down conditions via the RE signal.

The MPU has three 16-bit registers and three 8-bit registers available for use by the programmer (Figure 7).

PROGRAM COUNTER

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

STACK POINTER

The stack pointer is a two byte register that contains the address of the next available location in an external pushdown/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 two byte register that is used to store data or a 16-bit memory address for the indexed mode of memory addressing.

ACCUMULATORS

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

CONDITION CODE 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 (b6 and b7) are ones.

Figure 8 shows the order of saving the microprocessor status within the stack.

3

*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 and B parts (MC68A02 and MC68B02). On-board RAM can be used for data storage with all parts.

FIGURE 7 — PROGRAMMING MODEL OF THE MICROPROCESSING UNIT

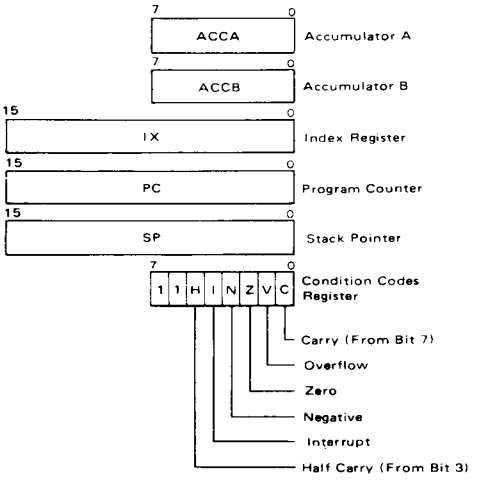
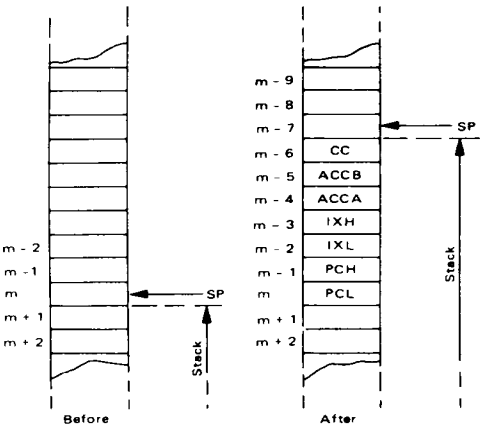


FIGURE 8 — SAVING THE STATUS OF THE MICROPROCESSOR IN THE STACK

SP = Stack Pointer
CC = Condition Codes (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



3

MPU SIGNAL DESCRIPTION

Proper operation of the MPU requires that certain control and timing signals be provided to accomplish specific functions and that other signal lines be monitored to determine the state of the processor. These control and timing signals are similar to those of the MC6800 except that TSC, DBE, ϕ1, ϕ2 input, and two unused pins have been eliminated, and the following signal and timing lines have been added:

RAM Enable (RE)
Crystal Connections EXTAL and XTAL
Memory Ready (MR)
V_{CC} Standby
Enable ϕ2 Output (E)

The following is a summary of the MPU signals:

ADDRESS BUS (A0-A15)

Sixteen pins are used for the address bus. The outputs are capable of driving one standard TTL load and 90 pF. These lines do not have three-state capability.

DATA BUS (D0-D7)

Eight pins are used for the data bus. It is bidirectional, transferring data to and from the memory and peripheral devices. It also has three-state output buffers capable of driving one standard TTL load and 130 pF.

Data bus will be in the output mode when the internal RAM is accessed and RE will be high. This prohibits external data entering the MPU. It should be noted that the internal RAM is fully decoded from $0000 to $007F. External RAM at $0000 to $007F must be disabled when internal RAM is accessed.

$\overline{\text{HALT}}$

When this input is in the low state, all activity in the machine will be halted. This input is level sensitive. In the $\overline{\text{HALT}}$ mode, the machine will stop at the end of an instruction, bus available will be at a high state, valid memory address will be at a low state. The address bus will display the address of the next instruction.

To ensure single instruction operation, transition of the $\overline{\text{HALT}}$ line must occur t_{PCS} before the rising 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. This is good engineering design practice in general and necessary to ensure proper operation of the part.

READ/WRITE (R/$\overline{\text{W}}$)

This TTL-compatible output signals the peripherals 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 will be in the read state. This output is capable of driving one standard TTL load and 90 pF.

VALID MEMORY ADDRESS (VMA)

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 three-state. One standard TTL load and 90 pF may be directly driven by this active high signal.

BUS AVAILABLE (BA) — The bus available signal will normally be in the low state; when activated, it will go to the high state indicating that the microprocessor has stopped and that the address bus is available (but not in a three-state condition). This will occur if the $\overline{\text{HALT}}$ line is in the low state or the processor is in the WAIT state as a result of the execution of a WAIT instruction. At such time, all three-state output drivers will go to their off-state and other outputs to their normally inactive level. The processor is removed from the

WAIT state by the occurrence of a maskable (mask bit I = 0) or nonmaskable interrupt. This output is capable of driving one standard TTL load and 30 pF.

INTERRUPT REQUEST ($\overline{IRQ}$)

A low level on this input requests that an interrupt sequence be generated within the machine. The processor will wait until it completes the current instruction that is being excuted before it recognizes the request. At that time, if the interrupt mask bit in the condition code register is not set, the machine will begin an interrupt sequence. The index register, program counter, accumulators, and condition code register are stored away on the stack. Next the MPU will respond to the interrupt request by setting the interrupt mask bit high so that no further interrupts may occur. At the end of the cycle, a 16-bit vectoring address which is located in memory locations $FFF8 and $FFF9 is loaded which 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 will be latched internally while $\overline{HALT}$ is low.

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

$\overline{RESET}$

This input is 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 will be lost. If a high level is detected on the input, this will signal the MPU to begin the restart sequence. This will start execution of a routine to initialize the processor from its reset condition. All the higher order address lines will be forced high. For the restart, the last two ($FFFE, $FFFF) locations in memory will be 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 and power-down sequences are shown in Figures 9 and 10, respectively.

$\overline{RESET}$, when brought low, must be held low at least three clock cycles. This allows adequate time to respond internally to the reset. This is independent of the t_{rc} power-up reset that is required.

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). This may cause improper MPU operation until the next valid reset.

NON-MASKABLE INTERRUPT ($\overline{NMI}$)

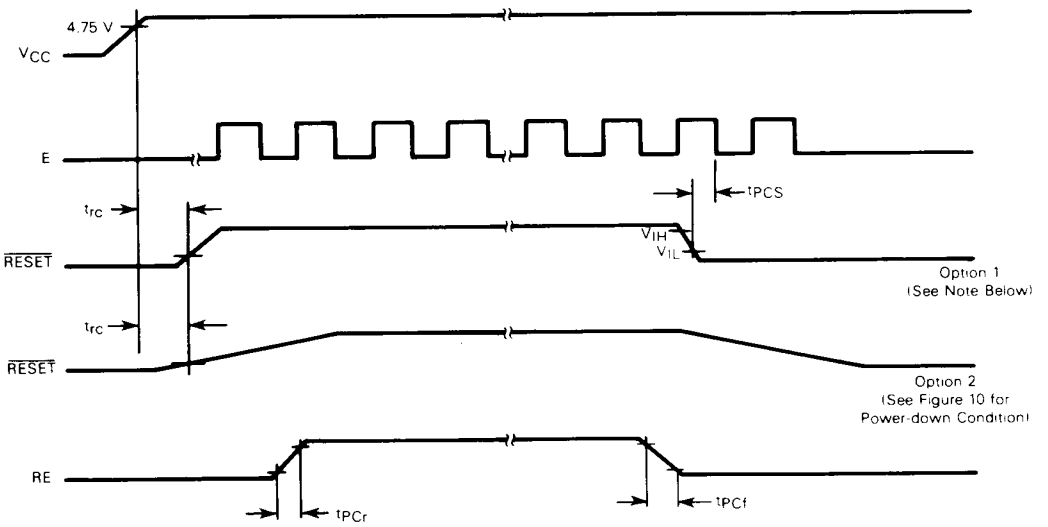
A low-going edge on this input requests that a nonmaskable interrupt sequence be generated within the processor. As with the interrupt request signal, the processor will complete the current instruction that is being executed before it recognizes the $\overline{NMI}$ signal. The interrupt mask bit in the condition code register has no effect on $\overline{NMI}$.

The index register, program counter, accumulators, and condition code registers are stored away on the stack. At the end of the cycle, a 16-bit vectoring address which is located in memory locations $FFFC and $FFFD is loaded causing the MPU to branch to an interrupt service routine in memory.

A nominal 3 kΩ pullup resistor to V_{CC} should be used for wire-OR and optimum control of interrupts. $\overline{NMI}$ may be tied

3

FIGURE 9 — POWER-UP AND RESET TIMING



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

directly to V_{CC} if not used.

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

Figure 11 is a flowchart 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		Description
MS	LS	
$FFFE	$FFFF	Restart
$FFFC	$FFFD	Non-Maskable Interrupt
$FFFA	$FFFB	Software Interrupt
$FFF8	$FFF9	Interrupt Request

FIGURE 10 — POWER-DOWN SEQUENCE

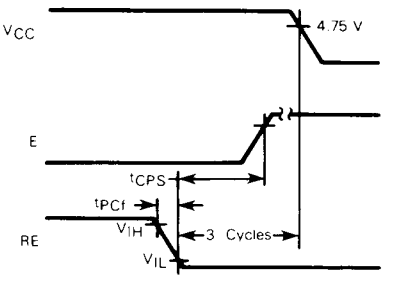


FIGURE 11 — MPU FLOWCHART

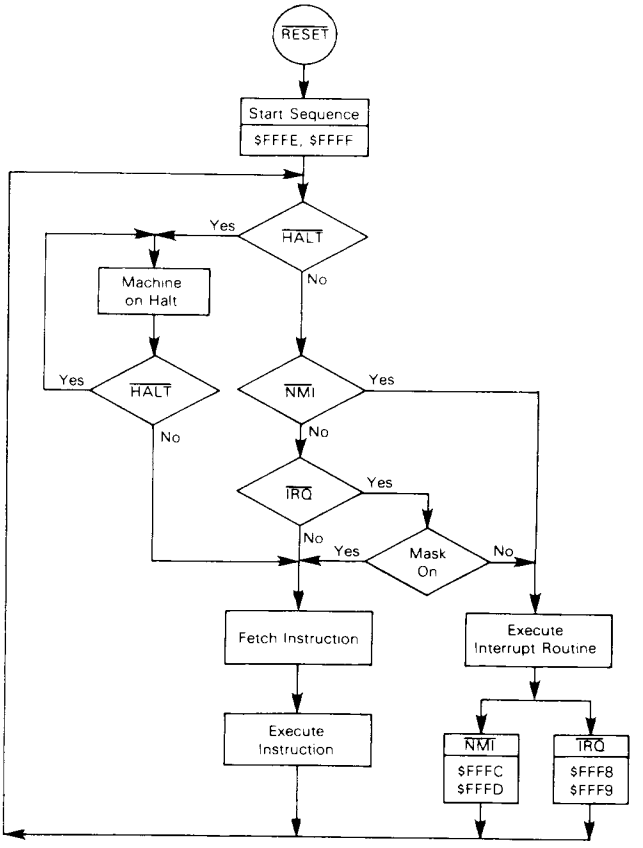
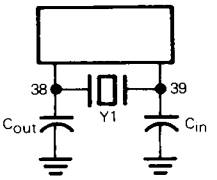
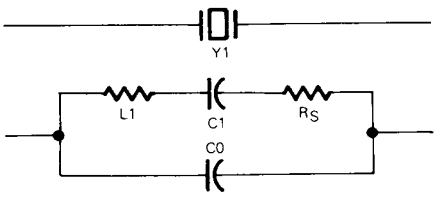


FIGURE 12 — CRYSTAL SPECIFICATIONS



Y1	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



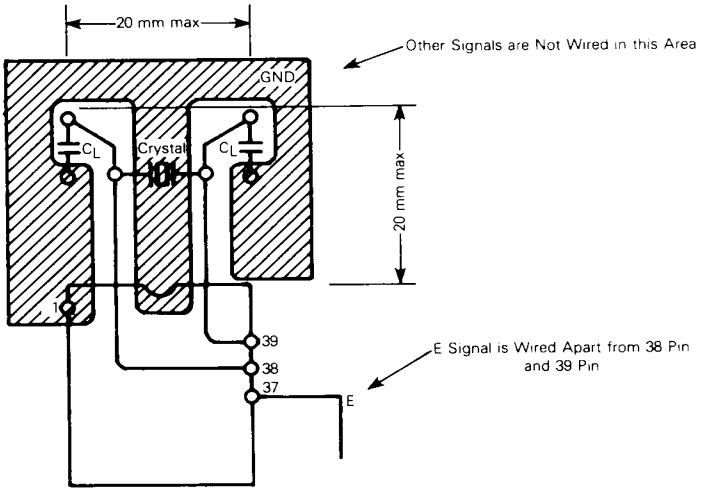
Nominal Crystal Parameters*

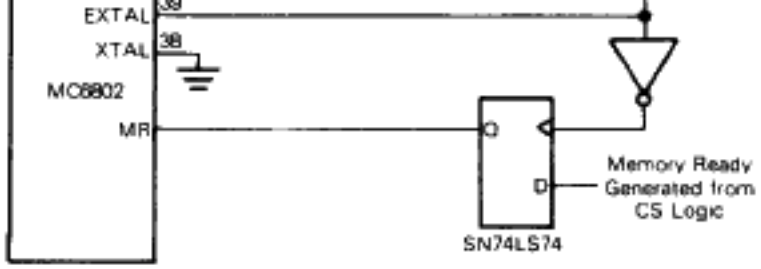
	3.58 MHz	4.0 MHz	6.0 MHz	8.0 MHz
R_S	60 Ω	50 Ω	30-50 Ω	20-40 Ω
C0	3.5 pF	6.5 pF	4-6 pF	4-6 pF
C1	0.015 pF	0.025 pF	0.01-0.02 pF	0.01-0.02 pF
Q	>40K	>30K	>20K	>20K

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

Figure 13 — SUGGESTED PC BOARD LAYOUT

Example of Board Design Using the Crystal Oscillator

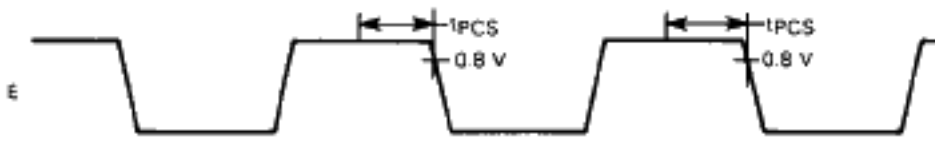




3

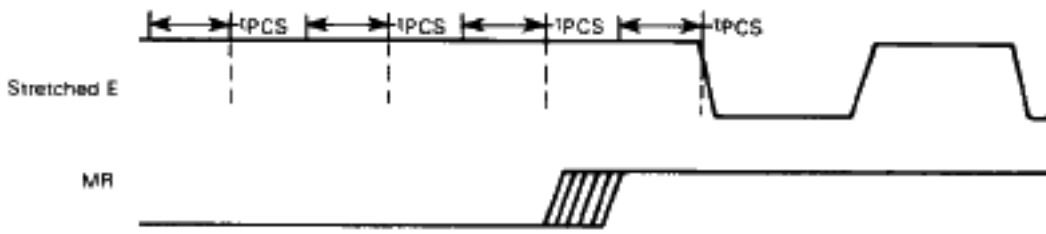
FIGURE 15 — MR NEGATIVE SETUP TIME REQUIREMENT

E Clock Stretch



The E clock will be stretched at end of E high of the cycle during which MR negative meets the t_{PCS} setup time. referenced to the fall of E. If the t_{PCS} setup time is not met, E will be stretched at the end of the next E high ½ cycle. tegral multiples of ½ cycles.

Resuming E Clocking



The E clock will resume normal operation at the end of the ½ cycle during which MR assertion meets the t_{PCS} setup t is referenced to transitions of E were it not stretched. If t_{PCS} setup time is not met, E will fall at the second possible asserted. There is no direct means of determining when the t_{PCS} references occur, unless the synchronizing circu

EXTAL AND XTAL

These inputs are used for the internal oscillator that may be crystal controlled. These connections are for a parallel resonant fundamental crystal (see Figure 12). (AT-cut.) A divide-by-four circuit has been added so a 4 MHz crystal may be used in lieu of a 1 MHz crystal for a more cost-effective system. An example of the crystal circuit layout is shown in Figure 13. Pin 39 may be driven externally by a TTL input signal four times the required E clock frequency. Pin 38 is to be grounded.

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

LC networks are not recommended to be used in place of the crystal.

If an external clock is used, it may not be halted for more than $t_{PW\phi L}$. The MC6802 is a dynamic part except for the internal RAM, and requires the external clock to retain information.

MEMORY READY (MR)

MR is a TTL-compatible input signal controlling the stretching of E. Use of MR requires synchronization with the $4xf_o$ signal, as shown in Figure 14. When MR is high, E will be in normal operation. When MR is low, E will be stretched integral numbers of half periods, thus allowing interface to slow memories. Memory Ready timing is shown in Figure 15.

MR should be tied high (connected directly to V_{CC}) if not used. This is necessary to ensure proper operation of the part. A maximum stretch is t_{cyc}.

ENABLE (E)

This pin supplies the clock for the MPU and the rest of the system. This is a single-phase, TTL-compatible clock. This clock may be conditioned by a memory read signal. This is equivalent to $\phi 2$ on the MC6800. This output is capable of driving one standard TTL load and 130 pF.

V_{CC} STANDBY

This pin supplies the dc voltage to the first 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 V_{SB} maximum is I_{SBB}.

There are seven address modes that
grammer, with the addressing mode a
type of instruction and the coding with
summary of the addressing modes for a
can be found in Table 7 along with the
execution time that is given in machin
frequency of 1 MHz, these times woul

ACCUMULATOR (ACCX) ADDRESSI

In accumulator only addressing, eith
accumulator B is specified. These are

IMMEDIATE ADDRESSING

In immediate addressing, the operan
second byte of the instruction except
have the operand in the second and
struction. The MPU addresses this loc
the immediate instruction for executio
three-byte instructions.

DIRECT ADDRESSING

In direct addressing, the address of t
ed in the second byte of the instructi
allows the user to directly address the l
machine, i.e., locations zero through
tion times are achieved by storing data
most configurations, it should be a ran
These are two-byte instructions.

EXTENDED ADDRESSING

In extended addressing, the addres
cond byte of the instruction is used as
the address of the operand. The third
is used as the lower eight bits of the ad
This is an absolute address in memory
instructions.

INDEXED ADDRESSING

In indexed addressing, the address
cond byte of the instruction is added
lowest eight bits in the MPU. The carr
higher order eight bits of the index
then used to address memory. The m
in a temporary address register so the
index register. These are two-byte ins

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 from Interr
RTS	Return from Subro
SBA	Subtract Accumula
SBC	Subtract with Carry
SEC	Set Carry
SEI	Set Interrupt Mask
SEV	Set Overflow
STA	Store Accumulator
STS	Store Stack Regis
STX	Store Index Regis
SUB	Subtract
SWI	Software Interrupt
TAB	Transfer Accumula
TAP	Transfer Accumula
TBA	Transfer Accumula
TPA	Transfer Condition
TST	Test
TSX	Transfer Stack Po
TXS	Transfer Index Re
WAI	Wait for Interrupt

	BITB	C5	2	2	D5	3	2	E5	5	2	F5	4	3				B · M	*	*
Clear	CLR							6F	7	2	7F	6	3				00 → M	*	*
	CLRA													4F	2	1	00 → A	*	*
	CLRB													5F	2	1	00 → B	*	*
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, 1's	COM							63	7	2	73	6	3				$\overline{M}$ → M	*	*
	COMA													43	2	1	$\overline{A}$ → A	*	*
	COMB													53	2	1	$\overline{B}$ → B	*	*
Complement, 2's	NEG							60	7	2	70	6	3				00 − M → M	*	*
(Negate)	NEGA													40	2	1	00 − A → A	*	*
	NEGB													50	2	1	00 − B → B	*	*
Decimal Adjust, A	DAA													19	2	1	Converts Binary Add. of BCD Characters into BCD Format	*	*
Decrement	DEC							6A	7	2	7A	6	3				M − 1 → M	*	*
	DECA													4A	2	1	A − 1 → A	*	*
	DECB													5A	2	1	B − 1 → B	*	*
Exclusive OR	EORA	88	2	2	98	3	2	A8	5	2	B8	4	3				A⊕M → A	*	*
	EORB	C8	2	2	D8	3	2	E8	5	2	F8	4	3				B⊕M → B	*	*
Increment	INC							6C	7	2	7C	6	3				M + 1 → M	*	*
	INCA													4C	2	1	A + 1 → A	*	*
	INCB													5C	2	1	B + 1 → B	*	*
Load Acmltr	LDAA	86	2	2	96	3	2	A6	5	2	B6	4	3				M → A	*	*
	LDAB	C6	2	2	D6	3	2	E6	5	2	F6	4	3				M → B	*	*
Or, Inclusive	ORAA	8A	2	2	9A	3	2	AA	5	2	BA	4	3				A + M → A	*	*
	ORAB	CA	2	2	DA	3	2	EA	5	2	FA	4	3				B + M → B	*	*
Push Data	PSHA													36	4	1	A → Msp, SP − 1 → SP	*	*
	PSHB													37	4	1	B → Msp, SP − 1 → SP	*	*
Pull Data	PULA													32	4	1	SP + 1 → SP, Msp → A	*	*
	PULB													33	4	1	SP + 1 → SP, Msp → B	*	*
Rotate Left	ROL							69	7	2	79	6	3				M	*	*
	ROLA													49	2	1	A (C ← b7 ← b0)	*	*
	ROLB													59	2	1	B	*	*
Rotate Right	ROR							66	7	2	76	6	3				M	*	*
	RORA													46	2	1	A (C → b7 → b0)	*	*
	RORB													56	2	1	B	*	*
Shift Left, Arithmetic	ASL							68	7	2	78	6	3				M	*	*
	ASLA													48	2	1	A (C ← b7 ← b0 ← 0)	*	*
	ASLB													58	2	1	B	*	*
Shift Right, Arithmetic	ASR							67	7	2	77	6	3				M	*	*
	ASRA													47	2	1	A (b7 → b0 → C)	*	*
	ASRB													57	2	1	B	*	*
Shift Right, Logic	LSR							64	7	2	74	6	3				M	*	*
	LSRA													44	2	1	A (0 → b7 → b0 → C)	*	*
	LSRB													54	2	1	B	*	*
Store Acmltr	STAA				97	4	2	A7	6	2	B7	5	3				A → M	*	*
	STAB				D7	4	2	E7	6	2	F7	5	3				B → M	*	*
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	*	*
	TBA													17	2	1	B → A	*	*
Test, Zero or Minus	TST							6D	7	2	7D	6	3				M − 00	*	*
	TSTA													4D	2	1	A − 00	*	*
	TSTB													5D	2	1	B − 00	*	*
																		H	I

LEGEND:

OP Operation Code (Hexadecimal);
~ Number of MPU Cycles;
\# Number of Program Bytes;
\+ Arithmetic Plus;
− Arithmetic Minus;
· Boolean AND;
Msp Contents of memory location pointed to be Stack Pointer

\+ Boolean Inclusive OR;
⊕ Boolean Exclusive OR;
$\overline{M}$ Complement of M;
→ Transfer Into;
0 Bit = Zero;
00 Byte = Zero;

Note – Accumulator addressing mode instructions are included in the column for IMPLIED addressing

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

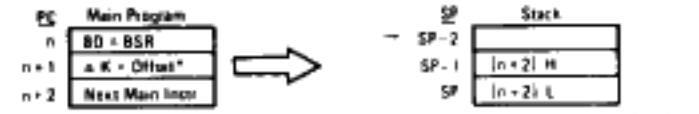
Load Index Reg	LDX	CE	3	2	DE	4	2	EE	6	2	FE	5	3				M → X_H, (M + 1) → X_L
Load Stack Ptr	LDS	8F	3	3	9E	4	2	AE	6	2	BE	5	3				M → SP_H, (M + 1) → SP_L
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3				X_H → M, X_L → (M + 1)
Store Stack Ptr	STS				9F	5	2	AF	7	2	BF	6	3				SP_H → M, SP_L → (M + 1)
Index Reg → Stack Ptr	TXS													35	4	1	X − 1 → SP
Stack Ptr → Index Reg	TSX													30	4	1	SP + 1 → X

3

TABLE 5 — JUMP AND BRANCH INSTRUCTIONS

		RELATIVE			INDEX			EXTND			IMPLIED			
OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	BRANCH TEST
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 ⊕ V = 0
Branch If > Zero	BGT	2E	4	2										Z + (N ⊕ V) = 0
Branch If Higher	BHI	22	4	2										C + Z = 0
Branch If ≤ Zero	BLE	2F	4	2										Z + (N ⊕ V) = 1
Branch If Lower Or Same	BLS	23	4	2										C + Z = 1
Branch If < Zero	BLT	2D	4	2										N ⊕ 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 (Figure 16)
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	
Return From Subroutine	RTS										39	5	1	
Software Interrupt	SWI										3F	12	1	See Special Operations (Figure 16)
Wait for Interrupt	WAI										3E	9	1	

BSR, BRANCH TO SUBROUTINE:



JMP, JUMP:

RTS, RETURN FROM SUBROUTINE:

RTI, RETURN FROM INTERRUPT:

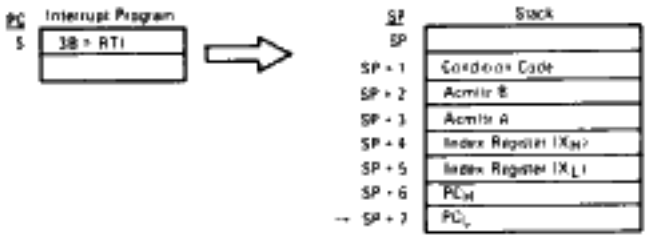


TABLE 6 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

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	———	———	———	———	———	———
CCR → Acmltr A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

(12)

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)
8 (Bit V) Test: 2's complement overflow from su
9 (Bit N) Test: Result less than zero? (Bit 15 = 1
10 (All) Load Condition Code Register from St
11 (Bit I) Set when interrupt occurs. If previousl Interrupt is required to exit the wait st
12 (All) Set according to the contents of Accur

ADC	x	•	2	3	4	5	•	•	INS		•	•	•	•	•
ADD	x	•	2	3	4	5	•	•	INX		•	•	•	•	•
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		•	•	•	•	•
BHI		•	•	•	•	•	•	4	ORA	x	•	2	3	4	5
BIT	x	•	2	3	4	5	•	•	PSH		•	•	•	•	•
BLE		•	•	•	•	•	•	4	PUL		•	•	•	•	•
BLS		•	•	•	•	•	•	4	ROL		2	•	•	6	7
BLT		•	•	•	•	•	•	4	ROR		2	•	•	6	7
BMI		•	•	•	•	•	•	4	RTI		•	•	•	•	•
BNE		•	•	•	•	•	•	4	RTS		•	•	•	•	•
BPL		•	•	•	•	•	•	4	SBA		•	•	•	•	•
BRA		•	•	•	•	•	•	4	SBC	x	•	2	3	4	5
BSR		•	•	•	•	•	•	8	SEC		•	•	•	•	•
BVC		•	•	•	•	•	•	4	SEI		•	•	•	•	•
BVS		•	•	•	•	•	•	4	SEV		•	•	•	•	•
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		•	•	•	•	•
CMP	x	•	2	3	4	5	•	•	TAB		•	•	•	•	•
COM		2	•	•	6	7	•	•	TAP		•	•	•	•	•
CPX		•	3	4	5	6	•	•	TBA		•	•	•	•	•
DAA		•	•	•	•	•	2	•	TPA		•	•	•	•	•
DEC		2	•	•	6	7	•	•	TST		2	•	•	6	7
DES		•	•	•	•	•	4	•	TSX		•	•	•	•	•
DEX		•	•	•	•	•	4	•	TSX		•	•	•	•	•
EOR	x	•	2	3	4	5	•	•	WAI		•	•	•	•	•

NOTE Interrupt time is 12 cycles from the end of the instruction being executed, except following a WAI instruction. Then it is 4 cycles.

TABLE 8 — OPERATIONS SUMMARY

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/$\overline{W}$ Line	Da
IMMEDIATE						
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Operand Data
CPX LDS LDX	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Operand Data (Hig
		3	1	Op Code Address + 2	1	Operand Data (Lov
DIRECT						
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operan
		3	1	Address of Operand	1	Operand Data
CPX LDS LDX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operan
		3	1	Address of Operand	1	Operand Data (Hig
		4	1	Operand Address + 1	1	Operand Data (Lo
STA	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Destination Addre
		3	0	Destination Address	1	Irrelevant Data (No
		4	1	Destination Address	0	Data from Accumu
STS STX	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operan
		3	0	Address of Operand	1	Irrelevant Data (No
		4	1	Address of Operand	0	Register Data (Hig
		5	1	Address of Operand + 1	0	Register Data (Low
INDEXED						
JMP	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (No
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (No
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (No
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (No
		5	1	Index Register Plus Offset	1	Operand Data
CPX LDS LDX	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (No
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (No
		5	1	Index Register Plus Offset	1	Operand Data (Hig
		6	1	Index Register Plus Offset + 1	1	Operand Data (Lo

		6	1	Index Register Plus Offset	0	Operand Da
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 D
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant D
		5	1	Index Register Plus Offset	1	Current Ope
		6	0	Index Register Plus Offset	1	Irrelevant D
		7	1/0 (Note 3)	Index Register Plus Offset	0	New Operan
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 D
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant D
		5	0	Index Register Plus Offset	1	Irrelevant D
		6	1	Index Register Plus Offset	0	Operand Da
		7	1	Index Register Plus Offset + 1	0	Operand Da
JSR	8	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant D
		4	1	Stack Pointer	0	Return Add
		5	1	Stack Pointer – 1	0	Return Add
		6	0	Stack Pointer – 2	1	Irrelevant D
		7	0	Index Register	1	Irrelevant D
		8	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant D

EXTENDED

JMP	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Jump Addr
		3	1	Op Code Address + 2	1	Jump Addr
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
		3	1	Op Code Address + 2	1	Address of
		4	1	Address of Operand	1	Operand Da
CPX LDS LDX	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of
		3	1	Op Code Address + 2	1	Address of
		4	1	Address of Operand	1	Operand Da
		5	1	Address of Operand + 1	1	Operand Da
STA A STA B	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Destination
		3	1	Op Code Address + 2	1	Destination
		4	0	Operand Destination Address	1	Irrelevant D
		5	1	Operand Destination Address	0	Data from
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
		3	1	Op Code Address + 2	1	Address of
		4	1	Address of Operand	1	Current Op
		5	0	Address of Operand	1	Irrelevant D
		6	1/0 (Note 3)	Address of Operand	0	New Opera

		6	1	Address of Operand + 1	0	Operand Data (Low
JSR	9	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Subroutin
		3	1	Op Code Address + 2	1	Address of Subroutin
		4	1	Subroutine Starting Address	1	Op Code of Next Ins
		5	1	Stack Pointer	0	Return Address (Low
		6	1	Stack Pointer − 1	0	Return Address (Hig
		7	0	Stack Pointer − 2	1	Irrelevant Data (Note
		8	0	Op Code Address + 2	1	Irrelevant Data (Note
		9	1	Op Code Address + 2	1	Address of Subroutin
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 Ins
DES DEX INS INX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Ins
		3	0	Previous Register Contents	1	Irrelevant Data (Note
		4	0	New Register Contents	1	Irrelevant Data (Note
PSH	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Ins
		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 Ins
		3	0	Stack Pointer	1	Irrelevant Data (Note
		4	1	Stack Pointer + 1	1	Operand Data from S
TSX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Ins
		3	0	Stack Pointer	1	Irrelevant Data (Note
		4	0	New Index Register	1	Irrelevant Data (Note
TXS	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Ins
		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
		3	0	Stack Pointer	1	Irrelevant Data (Note
		4	1	Stack Pointer + 1	1	Address of Next Inst Order Byte)
		5	1	Stack Pointer + 2	1	Address of Next Inst Order Byte)

		7	1	Stack Pointer – 4	0	Contents of Ac
		8	1	Stack Pointer – 5	0	Contents of Ac
		9	1	Stack Pointer – 6	1	Contents of Co
RTI	10	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Irrelevant Data
		3	0	Stack Pointer	1	Irrelevant Data
		4	1	Stack Pointer + 1	1	Contents of Co Stack
		5	1	Stack Pointer + 2	1	Contents of Ac
		6	1	Stack Pointer + 3	1	Contents of Ac
		7	1	Stack Pointer + 4	1	Index Register Byte)
		8	1	Stack Pointer + 5	1	Index Register Byte)
		9	1	Stack Pointer + 6	1	Next Instructio (High Order By
		10	1	Stack Pointer + 7	1	Next Instructio (Low Order By
SWI	12	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Irrelevant Data
		3	1	Stack Pointer	0	Return Addres
		4	1	Stack Pointer – 1	0	Return Addres
		5	1	Stack Pointer – 2	0	Index Register
		6	1	Stack Pointer – 3	0	Index Register
		7	1	Stack Pointer – 4	0	Contents of Ac
		8	1	Stack Pointer – 5	0	Contents of Ac
		9	1	Stack Pointer – 6	0	Contents of Co
		10	0	Stack Pointer – 7	1	Irrelevant Data
		11	1	Vector Address FFFA (Hex)	1	Address of Sub Byte)
		12	1	Vector Address FFFB (Hex)	1	Address of Sub 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
		4	0	Branch Address	1	Irrelevant Data
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
		4	1	Stack Pointer	0	Return Addres
		5	1	Stack Pointer – 1	0	Return Addres
		6	0	Stack Pointer – 2	1	Irrelevant Data
		7	0	Return Address of Main Program	1	Irrelevant Data
		8	0	Subroutine Address (Note 4)	1	Irrelevant Data

NOTES:
1. If device which is addressed during this cycle uses VMA, then the Data Bus will go to the high-impedanc 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. MS Byte of Address Bus = MS Byte of Address of BSR instruction and LS Byte of Address Bus = LS Byte of

Cerdip	1.0	0°C to 70°C	MC6802S
S Suffix	1.0	−40°C to +85°C	MC6802CS
	1.5	0°C to 70°C	MC68A02S
	1.5	−40°C to +85°C	MC68A02CS
	2.0	0°C to 70°C	MC68B02S

PIN ASSIGNMENT