MOTOROLA

MC68120
MC68121

Advance Information

HMOS

(HIGH-DENSITY N-CHANNEL SILICON-GATE)

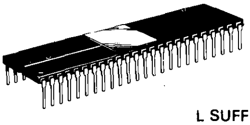
INTELLIGENT PERIPHERAL CONTROLLER

INTELLIGENT PERIPHERAL CONTROLLER

The MC68120/MC68121 Intelligent Peripheral Controller (IPC) is a general purpose, mask programmable peripheral controller. The IPC provides the interface between an M68000 or M6800 Family microprocessor and the final peripheral devices through a system bus and control lines. System bus data is transferred to and from the IPC via dual-port RAM while the software utilizes the semaphore registers to control RAM tasking or any other shared resource. Multiple operating modes range from a single chip mode with 21 I/O lines and 2 control lines to an expanded mode supporting an address space of 64K bytes. The MC68120 has 2K bytes of on-chip ROM to make full use of all operating modes. The MC68121 utilizes only the expanded address modes, due to the absence of on-chip ROM.

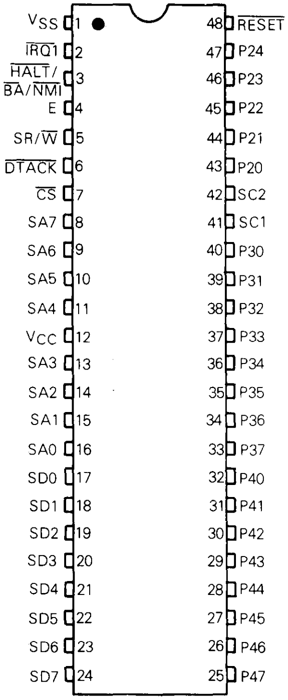
A serial communications interface, 16-bit timer, dual-ported RAM and semaphore registers are available for use by the IPC in all operating modes.

- System Bus Compatible with the Asynchronous M68000 Family
- System Bus Compatible with the MC6809 and Other M6800 Family Processors/Peripherals
- Local Bus Allows Interface with all M6800 Peripherals
- MC6801 Source and Object Code Compatible
- Upward Compatible with MC6800 Source and Object Code
- 2048 Bytes of ROM (MC68120 Only)
- 128 Bytes of Dual-Ported RAM
- Multiple Operation Modes Ranging from Single Chip to Expanded, with 64K Byte Address Space
- Six Shared Semaphore Registers
- 21 Parallel I/O Lines and 2 Handshake Lines (5 I/O Lines on MC68121)
- Serial Communications Interface (SCI)
- 16-Bit Three-Function Timer
- 8-Bit CPU and Internal Bus
- Halt/Bus Available Capability Control
- 8×8 Multiply Instruction
- TTL Compatible Inputs and Outputs
- External and Internal Interrupts



L SUFFIX
CERAMIC PACKAGE
CASE 740

PIN ASSIGNMENT

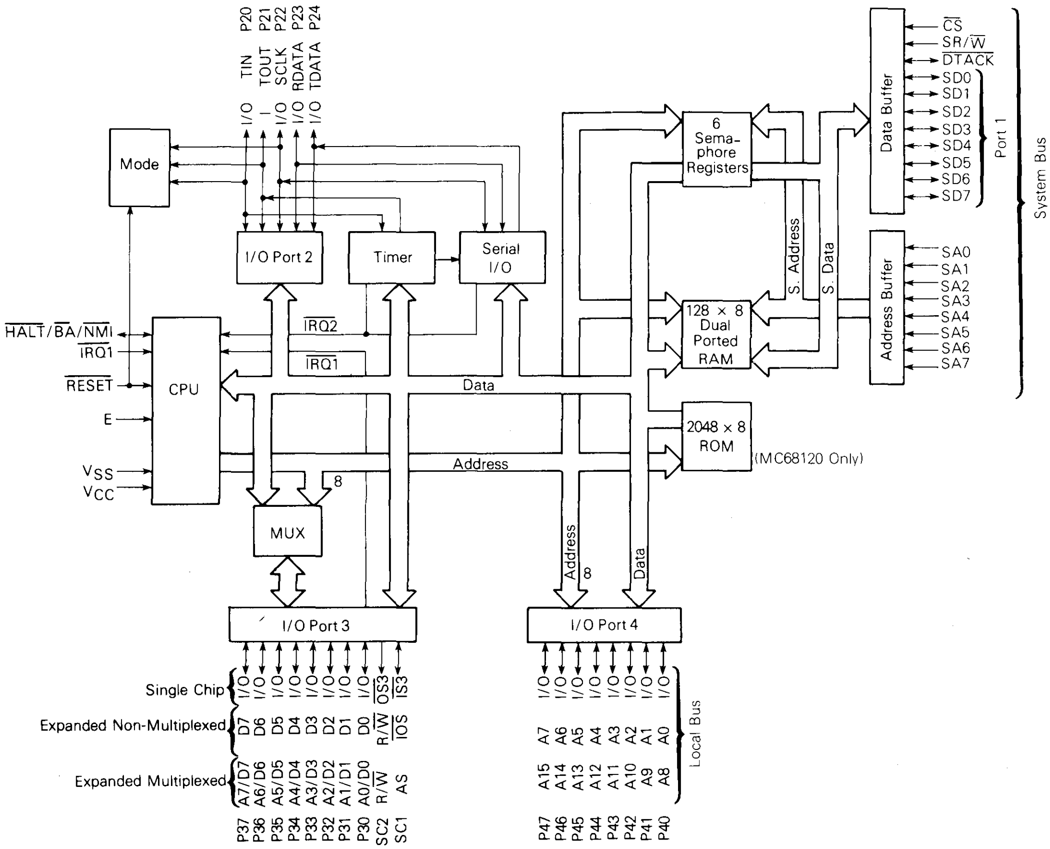


GENERIC INFORMATION
($T_A = 0°C$ to $70°C$)

Package Type	Frequency (MHz)	Generic Number
Ceramic	1.0	MC68120L1 (Unicorn ROM)
L Suffix	1.0	MC68121L
	1.25	MC68120L1-1 (Unicorn ROM)
	1.25	MC68121L-1

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

MC68120/MC68121 INTELLIGENT PERIPHERAL CONTROLLER — BLOCK DIAGRAM



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	T_A	0 to 70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Rating
Thermal Resistance Ceramic Package	θ_{JA}	50	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out} \leq V_{CC}$.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

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

DC LOCAL BUS ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, T_A = 0° to 70°C unless otherwise noted)
(Refer to Figures 1 and 2)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	E	V_{EIH}	$V_{CC}-0.75$	–	V_{CC}	V
Input Low Voltage	E	V_{EIL}	$V_{SS}-0.3$	–	$V_{SS}+0.6$	V
Input High Voltage	$\overline{RESET}$ Other Inputs*	V_{IH}	$V_{SS}+4.0$ $V_{SS}+2.0$	– –	V_{CC} V_{CC}	V
Input Low Voltage	All Inputs*	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Load Current (V_{in} = 0 to 2.4 V)	Port 4	I_{in}	–	–	0.5	mA
Input Leakage Current (V_{in} = 0 to 5.25 V)	SCI, $\overline{HALT}/\overline{NMI}$, $\overline{IRQ1}$, $\overline{RESET}$	I_{in}	–	1.5	2.5	μA
Three-State (Off State) Input Current (V_{in} = 0.5 to 2.4 V)	SD0-SD7, P20-P24, P30-P37	I_{TSI}	–	2.0	10	μA
Output High Voltage (I_{load} = −65 μA, V_{CC} = min) (I_{load} = −100 μA, V_{CC} = min)	P40-P47, SC1, SC2 Other Outputs	V_{OH}	$V_{SS}+2.4$ $V_{SS}+2.4$	– –	– –	V
Output Low Voltage (I_{load} = 2.0 mA, V_{CC} = min)	All Outputs	V_{OL}	–	–	$V_{SS}+0.5$	V
Internal Power Dissipation (measured at T_A = 0°C)		P_{INT}	–	–	1200	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f_o = 1.0 MHz)	E P30-P37, P40-P47, SC1 Other Inputs	C_{in}	– – –	– – –	60.0 12.5 10.0	pF

*Except Mode Programming Levels; See Figure 29.

FIGURE 1 — CMOS LOAD

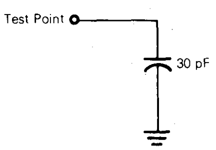
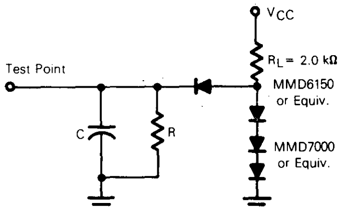


FIGURE 2 — TIMING TEST LOAD PORTS 2, 3, 4



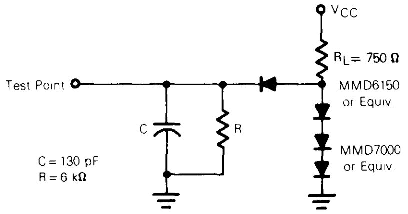
C = 90 pF for P30-P37, P40-P47, $\overline{SC1}$, SC2
= 30 pF for P20-P24, $\overline{HALT}/\overline{BA}/\overline{NMI}$
R = 37 kΩ for P40-P47, $\overline{SC1}$, SC2
= 24 kΩ for P20-P24, $\overline{HALT}/\overline{BA}/\overline{NMI}$, P30-P37

DC SYSTEM BUS ELECTRICAL CHARACTERISTICS

(V_{CC}=5.0 Vdc ±5%, V_{SS}=0, T_A=70°C unless otherwise noted) (Refer to Figure 3)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	$\overline{CS}$, $\overline{DTACK}$, SA0-SA7, SD0-SD7, SR/$\overline{W}$	V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage	$\overline{CS}$, $\overline{DTACK}$, SA0-SA7, SD0-SD7, SR/$\overline{W}$	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Output High Voltage ($I_{Load}=-400\ \mu A$, V_{CC}=min)	$\overline{DTACK}$, SD0-SD7	V_{OH}	$V_{SS}+2.4$	–	–	V
Output Low Voltage (I_{Load}=5.3 mA, V_{CC}=min)	$\overline{DTACK}$, SD0-SD7	V_{OL}	–	–	$V_{SS}+0.5$	V

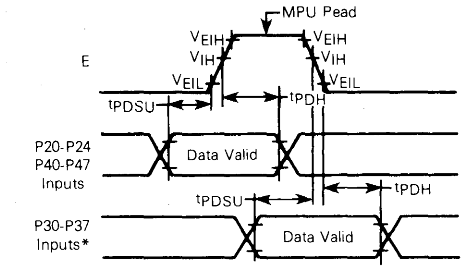
FIGURE 3 – TIMING TEST LOAD SD0-SD7, $\overline{DTACK}$



PERIPHERAL PORT TIMING (Refer to Figures 4 through 7)

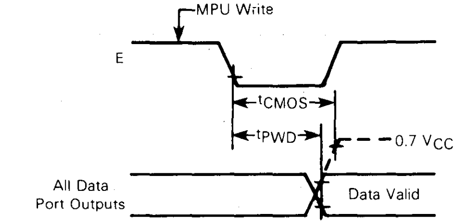
Characteristics	Symbol	Min	Max	Unit
Peripheral Data Setup Time	t_{PDSU}	200	–	ns
Peripheral Data Hold Time	t_{PDH}	200	–	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition	t_{OSD1}	–	350	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition	t_{OSD2}	–	350	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid (Ports 2, 3, 4)	t_{PWD}	–	350	ns
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	t_{CMOS}	–	2.0	μs
Input Strobe Pulse Width	t_{PWIS}	200	–	ns
Input Data Hold Time	t_{IH}	60	–	ns
Input Data Setup Time	t_{IS}	20	–	ns
Input Capture Pulse Width (Timer Function)	t_{PWIC}	2	–	E_{cyc}

FIGURE 4 – DATA SETUP AND HOLD TIMES (MPU READ LOCAL BUS)



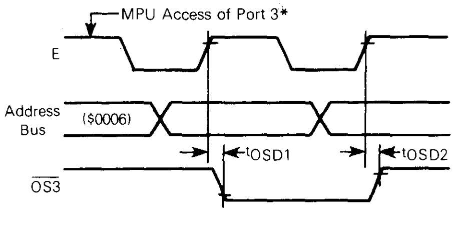
*Port 3 Non-Latched Operation (LATCH ENABLE=0)

FIGURE 5 – DATA SETUP AND HOLD TIMES (MPU WRITE LOCAL BUS)



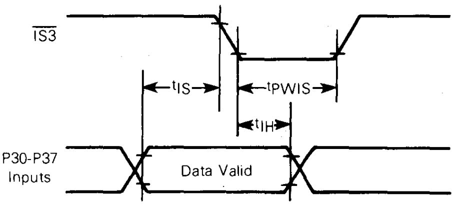
Notes:
1. 10 k Pullup resistor required for Port 2 to reach 0.7 V_{CC}
2. Not applicable to P21
3. Port 4 cannot be pulled above V_{CC}

FIGURE 6 – PORT 3 OUTPUT STROBE TIMING (SINGLE CHIP MODE)



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

FIGURE 7 – PORT 3 LATCH TIMING (SINGLE CHIP MODE)

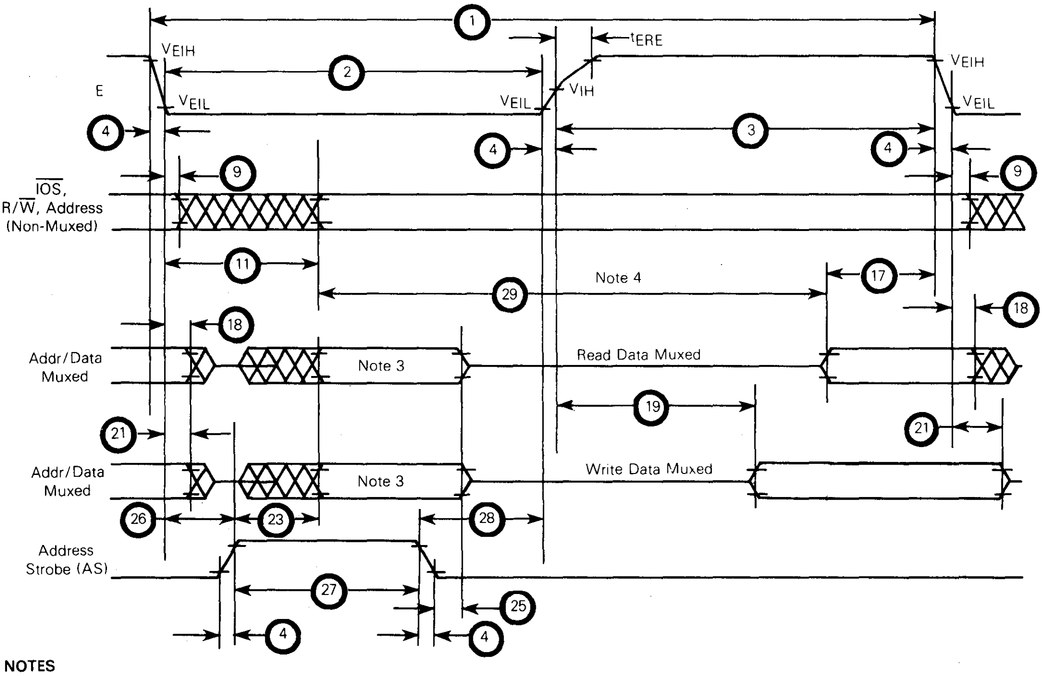


Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

LOCAL BUS TIMING (See Notes 1 and 2)

Ident. Number	Characteristics	Symbol	MC68120/ MC68121		MC68120-1/ MC68121-1		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	2.0	0.8	2.0	μs
2	Pulse Width, E Low	PW_{EL}	430	1000	360	1000	ns
3	Pulse Width, E High	PW_{EH}	450	1000	360	1000	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	ns
9	Non-Muxed Address Hold Time	t_{AH}	20	–	20	–	ns
11	Address Delay From E Low	t_{AD}	–	260	–	220	ns
17	Read Data Setup Time	t_{DSR}	80	–	70	–	ns
18	Read Data Hold Time	t_{DHR}	10	–	10	–	ns
19	Write Data Delay Time	t_{DDW}	–	225	–	200	ns
21	Write Data Hold Time	t_{DHW}	20	–	20	–	ns
23	Muxed Address Delay from AS	t_{ADM}	–	90	–	80	ns
25	Muxed Address Hold Time	t_{AHL}	20	110	20	110	ns
26	Delay Time E to AS Rise	t_{ASD}	100	–	80	–	ns
27	Pulse Width, AS High	PW_{ASH}	220	–	170	–	ns
28	Delay Time AS to E Rise	t_{ASED}	100	–	80	–	ns
29	Usable Access Time (Note 4)	t_{ACC}	570	–	435	–	ns
	Enable Rise Time Extended	t_{ERE}	–	80	–	80	ns
	Processor Control Setup Time	t_{PCS}	200	–	200	–	ns
	Processor Control Hold Time	t_{PCH}	20	40	20	40	ns

FIGURE 8 — LOCAL BUS TIMING



NOTES

1. Voltage levels shown are $V_L \leq 0.5$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
3. Address valid on the occurrence of the latest of 11 or 23.
4. Usable access time is computed by: 1 – (4 + 11 + 17).

ASYNCHRONOUS SYSTEM BUS TIMING (Refer to Figures 9, 10, 11 and 12)

Characterisic	Symbol	Min	Typ	Max	Unit
Cycle Time	t_{cyc}	0.8	–	2.0	µs
System Address Setup	t_{SAS}	30	–	–	ns
System Address Hold	t_{SAH}	0	–	–	ns
System Data Delay Read Semaphore	t_{SDDR}	0.3	–	0.3+1.5 t_{cyc}*	µs
RAM	t_{SDDR}	–	315	–	ns
System Data Valid	t_{SDV}	0	–	–	ns
System Data Hold Read	t_{SDHR}	0	–	100	ns
System Data Delay Write Semaphore	t_{SDDW}	**	–	**	ns
RAM	t_{SDDW}	–	–	60	ns
System Data Hold Write	t_{SDHW}	0	–	–	ns
Data Acknowledge Semaphore	t_{DAL}	0.5	–	0.5+1.5 t_{cyc}*	µs
RAM	t_{DAL}	–	315	–	ns
Data Acknowledge High	t_{DAH}	–	–	60	ns
Data Acknowledge Three-State	t_{DAT}	–	–	90	ns
Data Acknowledge Low to $\overline{CS}$ High	t_{DCS}	60	–	–	ns

*Actual value dependent upon clock period.
**Data need not be valid on write to Semaphore Registers.

FIGURE 9 — ASYNCHRONOUS READ OF SEMAPHORE REGISTER

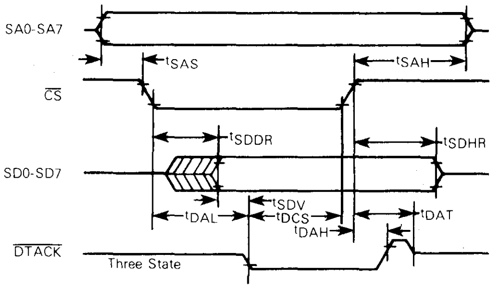


FIGURE 10 — ASYNCHRONOUS WRITE OF SEMAPHORE REGISTER

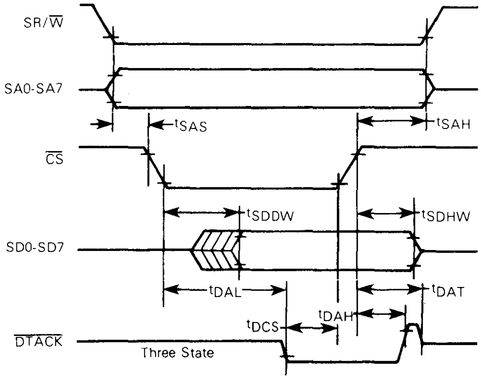


FIGURE 11 — ASYNCHRONOUS READ OF RAM

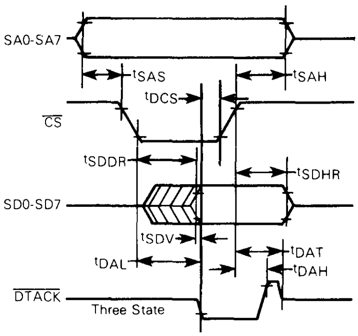
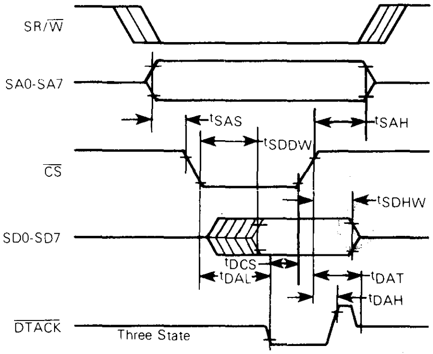


FIGURE 12 — ASYNCHRONOUS WRITE OF RAM

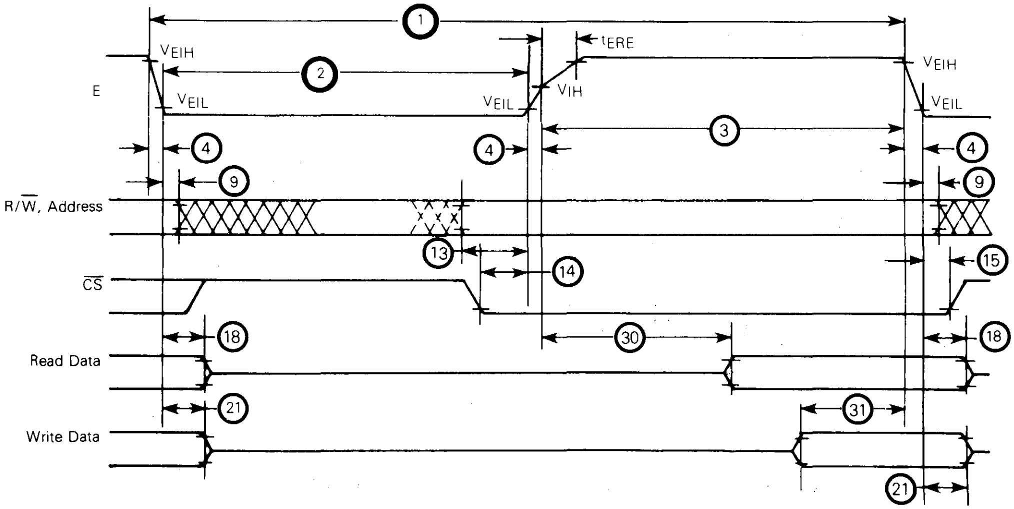


Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

SYNCHRONOUS SYSTEM BUS TIMING (See Notes 1 and 2)

Ident Number	Characteristic	Symbol	MC68120/ MC68121		MC68120-1 MC68121-1		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.80	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	360	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	360	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	ns
9	Address Hold Time	t_{AH}	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	70	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	70	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	30	100	30	85	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	250	ns
31	Input Data Setup Time	t_{DSW}	165	–	120	–	ns
	Clock Enable Rise Time Extended	t_{ERE}	–	80	–	80	ns

FIGURE 13 — SYNCHRONOUS SYSTEM BUS TIMING



Notes:
1. Voltage levels shown are $V_L \leq 0.5$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

INTRODUCTION

The MC68120/MC68121 is an 8-bit Intelligent Peripheral Controller (IPC) which can be configured to function in a wide variety of applications. This extraordinary flexibility is provided by its ability to be hardware programmed into eight different operating modes. These operating modes allow the IPC to operate on its local bus and communicate with an external system bus through the internal dual-ported RAM. The operating mode controls the configuration of 18 of the 48 pins on the IPC, the available on-chip resources, the memory map, the location (internal or external) of interrupt vectors, and the type of local bus. The configuration of the remaining 30 pins is not controlled by the operating mode.

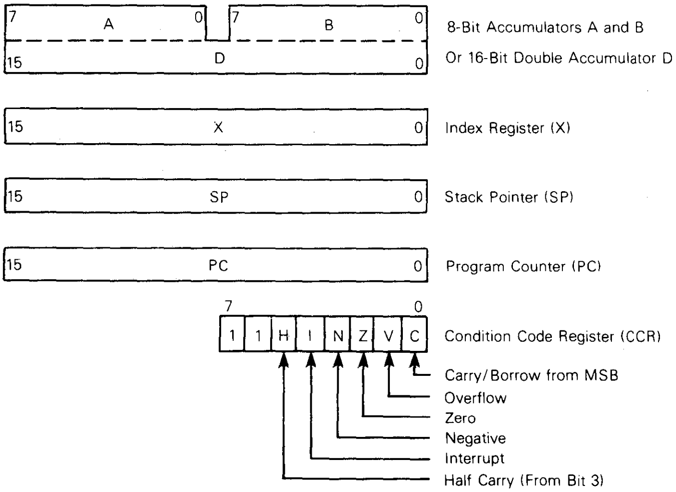
The dual-ported RAM provides a vehicle for devices on two separate buses to exchange data without directly affecting the devices on the other bus. The dual-ported RAM is accessible from the MC68120/MC68121 CPU and accessible synchronously or asynchronously to the system bus through Port 1. Semaphore registers are provided as a software tool to arbitrate shared resources such as the dual-ported RAM. The semaphore registers are accessible from both buses in the same way each bus accesses the dual-ported RAM.

The remaining ports (2, 3, and 4) are I/O ports. Each port is controlled by its Data Direction Register. The CPU has direct access to the port pins of each port through its Data Register. Port pins are labeled as P_{ij} where i identifies one of three ports and j indicates the particular bit. Port 2 is a 5-bit port which may be configured for I/O or for use of the on-chip timer and Serial Communications Interface (SCI). Ports 3 and 4 may be used as 16 bits of I/O or may form a local address and data bus with control lines allowing communications with external memory and peripherals.

The IPC contains an enhanced M6800 MPU with additional capabilities and greater throughput. It is upward source and object code compatible with the MC6800 and directly compatible with the MC6801. The programming model is depicted in Figure 14, where accumulator D is a concatenation of accumulators A and B.

The MC68121 has all of the features of the MC68120 with the exception of on-chip ROM. Thus the MC68121 normally operates in the modes utilizing external ROM (modes 2 and 3). Therefore, modes 0, 1, 4, 5, 6 and 7 should not be used.

FIGURE 14 — PROGRAMMING MODEL



DUAL-PORTED RAM AND SEMAPHORE REGISTERS

The dual-ported RAM may be accessed from both the MC68120/MC68121 CPU and the external system bus. The six semaphore registers are tools provided for the programmer's use in arbitrating simultaneous accesses of the same resource.

For the internal CPU, the dual-ported RAM is located from $0080 through $00FF in all modes except 3 and 4. In mode 3, the dual-ported RAM has been relocated in high memory from $C080 through $C0FF thus allowing use of direct addressing mode on external memory/peripherals. Note that no direct addressing of internal control registers is possible in mode 3. In mode 4, the internal RAM is not fully decoded and appears in locations $XX80 through $XXFF. From the external system bus, the dual-ported RAM is found in locations %10000000-11111111, as shown below in Table 1.

TABLE 1 — LOCATION OF SEMAPHORE REGISTERS AND DUAL-PORTED RAM

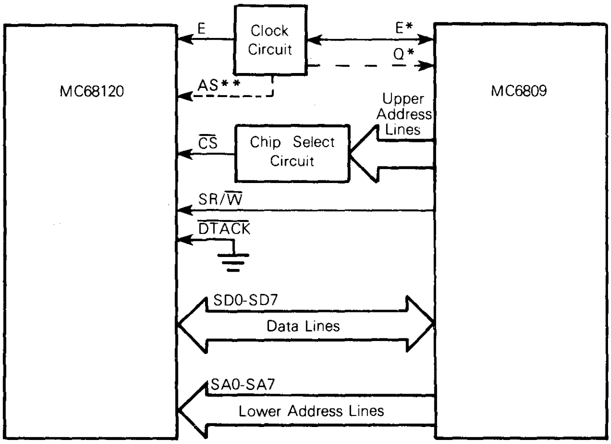
System Bus Address (SA7-SA0)	Feature	IPC Address*
%0000 0000 – 0001 0110	Reserved	------
---- ---- – ---- ----	Internal Registers	$00-16
0001 0111 – 0001 1100	Semaphore Registers	17-1C
0001 1101 – 0111 1111	Reserved	1D-1F
---- ---- – ---- ----	External Mem./Unusable*	20-7F
1000 0000 – 1111 1111	Dual-Ported RAM	80-FF

% = Binary; $ = Hexadecimal
* Mode Dependent

The reserved memory areas %0-0001 0110 and %0001 1101-%0111 1111 cannot be written to from the System bus. If read from the System bus these memory locations return a value of $FF.

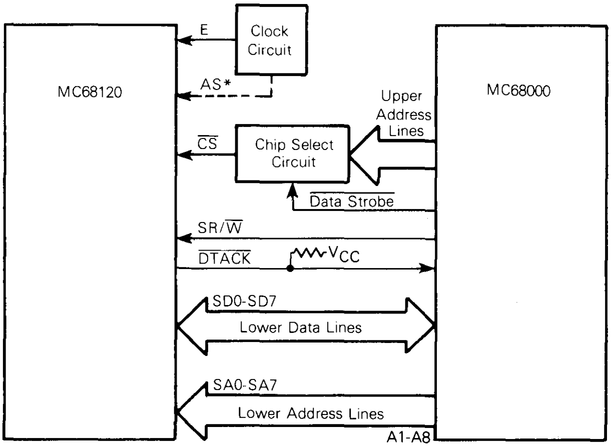
The dual-ported RAM is accessed from the external System bus by way of eight address lines (SA0-SA7) and eight data lines (SD0-SD7). Three control lines provide for synchronous or asynchronous access to the dual-ported RAM through Port 1. Figure 15 shows an example of a synchronous interface (using MC6809) and Figure 16 shows an example of an asynchronous interface (using MC68000). The dual-ported RAM is selected in each case by address lines SA0-SA7 and Chip Select ($\overline{\text{CS}}$) from the system bus. The direction of data transfer is selected by the System Read/Write (SR/$\overline{\text{W}}$) line. The Data Transfer Acknowledge ($\overline{\text{DTACK}}$) signal is the asynchronous handshake required by an MC68000. Refer to $\overline{\text{DTACK}}$ under Functional Pin Description for more information. $\overline{\text{DTACK}}$ can be used to control a Memory Ready signal on the M6800 Family processor where Memory Ready capability is provided (see Figure 17). The latter would allow the M6800 Family processor to run asynchronously with the MC68120/MC68121. It should be noted that if the Memory Ready signal (on M6800 processors) is to be used with the $\overline{\text{DTACK}}$ signal, the system clock must be faster than or equal to the clock driving the IPC. Example clock circuits are shown in Figures 18 and 19.

FIGURE 15 — SYNCHRONOUS SYSTEM BUS ACCESS INTERFACE



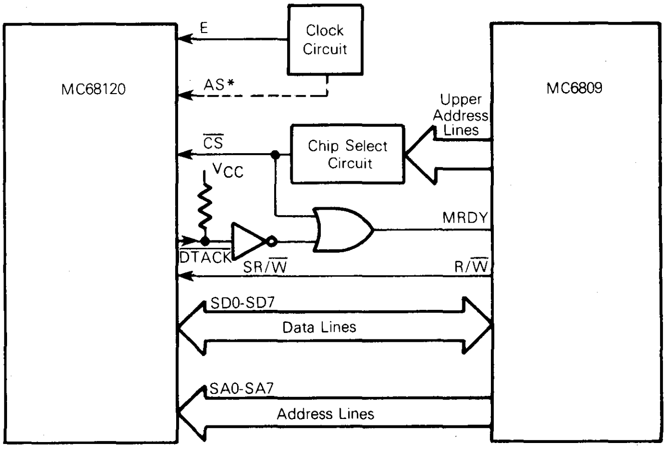
*E and Q are inputs for MC6809E
**Only needed in expanded multiplexed modes.

FIGURE 16 — ASYNCHRONOUS SYSTEM BUS INTERFACE



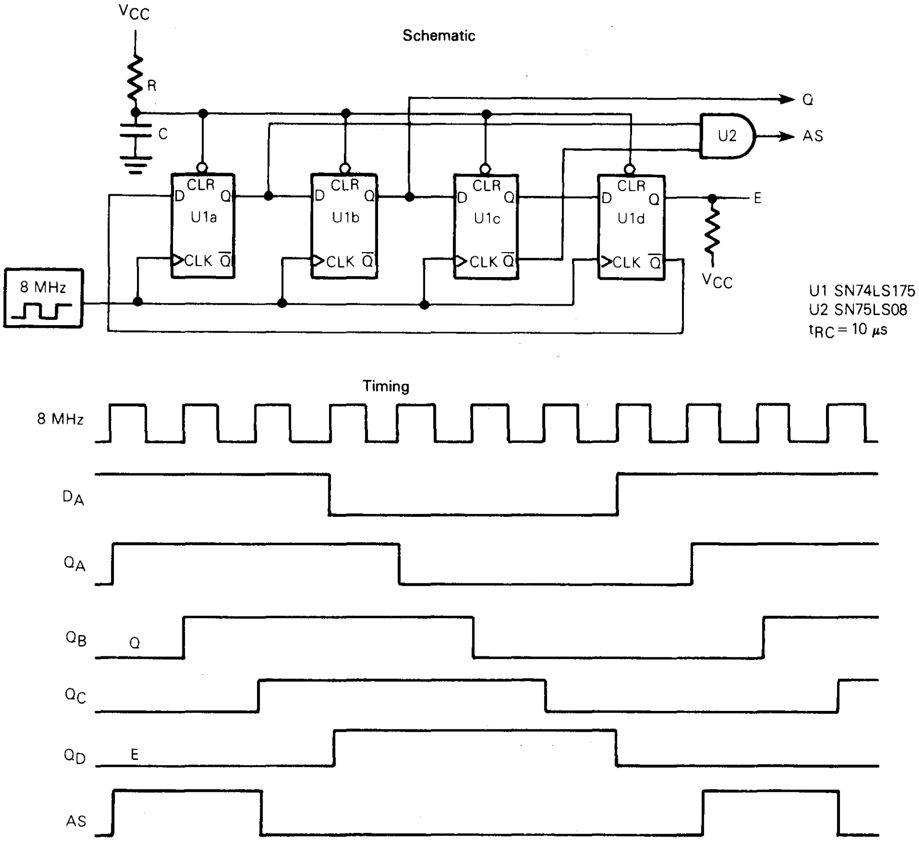
* Only needed in expanded multiplexed modes.

FIGURE 17 — MEMORY READY — DTACK CONFIGURATION



* Only needed in expanded multiplexed modes.

FIGURE 18 — CLOCK CIRCUIT EXAMPLE 1 — SCHEMATIC AND TIMING



Schematic

Timing

The semaphore registers allow arbitration between shared resources, which may be part or all of the dual-port RAM, or a peripheral. The semaphore registers may also be used to indicate that non-reentrant code is in use or that a task is in process or is complete. To prevent the writing or reading of erroneous data from the dual-ported RAM, all simultaneous accesses involving a write to the same byte in the dual-ported RAM should be avoided. The responsibility for mutual exclusion resides in software. The semaphore registers are a convenient means for the software to control the simultaneous accesses involving a write to the dual-ported RAM. Each of the six semaphore registers consist of a semaphore bit (SEM, bit 7) and an ownership bit (OWN, bit 6). The remaining six bits (b0-b5) will read all zeros.

SEMAPHORE REGISTER

7	6	5	4	3	2	1	0
SEM	OWN	0	0	0	0	0	0

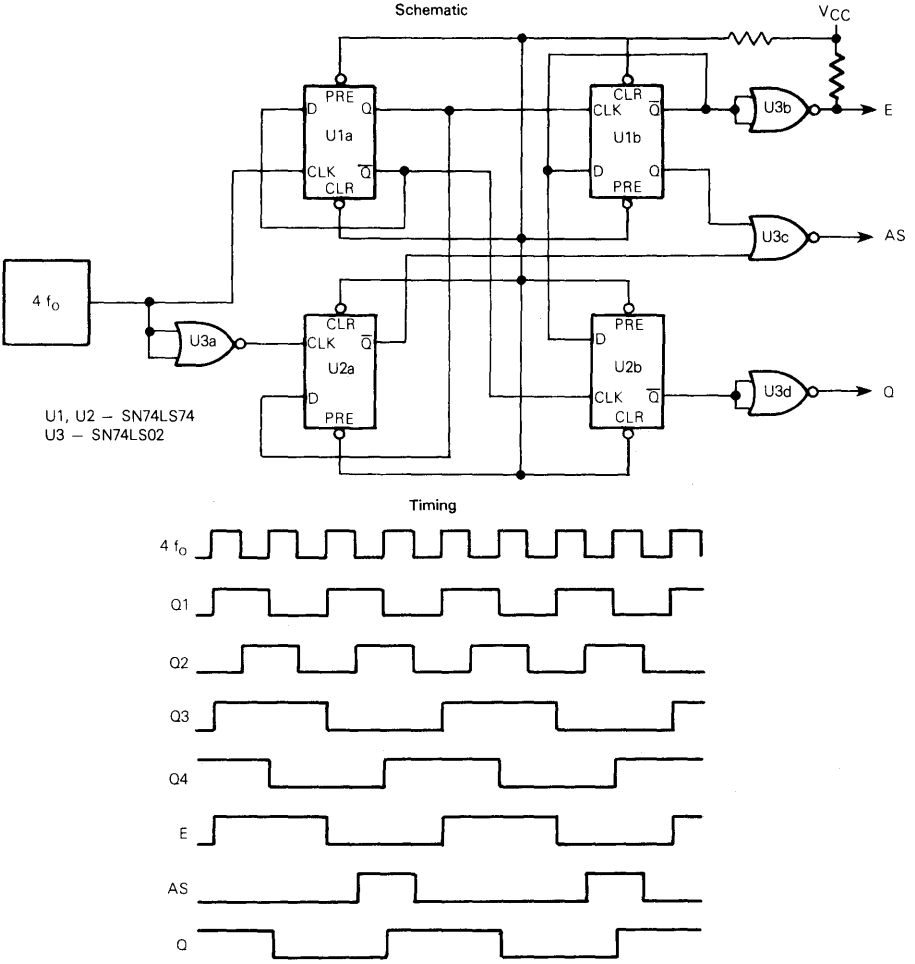
The semaphore bits are test and set bits with hardware arbitration during simultaneous accesses. Basically, the semaphore bit is cleared when written and set when read, during a single processor access. This is shown in Table 2.

TABLE 2 — SINGLE PROCESSOR SEMAPHORE BIT TRUTH TABLE

Original SEM Bit	R/$\overline{W}$	Data Read	Resulting SEM Bit
0	R	0*	1
1	R	1*	1
0	W	–	0
1	W	–	0

*0 — Resource Available
1 — Resource Not Available

FIGURE 19 — CLOCK CIRCUIT EXAMPLE 2 — SCHEMATIC AND TIMING



3

The data written is disregarded and the information obtained from the Read may be interpreted as: 0 — resource available; 1 — resource not available. Thus, any write to a semaphore clears the semaphore bit and makes the associated resource "available."

An access where both the IPC and system processors attempt to read or write the same semaphore register simultaneously is a contested access. During a contested access, the hardware decides which processor reads a clear semaphore bit and which reads a set semaphore bit. Table 3 describes contested operation of a semaphore bit.

The IPC always reads the actual semaphore bit; the system processor reads the semaphore bit in all cases except the simultaneous read of a clear semaphore bit. This arbitration during a simultaneous read ensures that only one processor reads a clear bit and therefore controls the resource; that processor is arbitrarily the IPC.

In Table 3, the first four states are considered proper and they occur in correctly written software. The last four states are improper and only exist in improperly written software.

The ownership bit is a read-only bit that indicates which processor sets the semaphore bit. If the semaphore bit is set, the ownership bit indicates which processor set it. If the semaphore bit is not set, the ownership bit indicates which processor last set the semaphore bit; OWN = 0, the other processor set SEM; OWN = 1, this processor set SEM.

The reset state of the semaphore and ownership bits is defined in Table 4. All of the semaphore bits are set after an M.C68120/MC68121 reset. The IPC owns all of them except the second semaphore which is owned by the system processor. This configuration should prevent the system processor from reading a clear semaphore and implying the system processor set it when the IPC $\overline{\text{RESET}}$ is held low.

TABLE 3 — DUAL PROCESSOR SEMAPHORE BIT TRUTH TABLE

Original SEM Bit	IPC R/$\overline{W}$	IPC Data Read	System R/$\overline{W}$	System Data Read	Resulting SEM Bit	
0	R	0*	R	1*	1	PROPER
1	R	1*	W	–	0	
1	W	–	R	1*	0	
1	R	1	R	1*	1	
0	W	–	W	–	0	IMPROPER
0	R	0*	W	–	1	
1	W	–	W	–	0	
0	W	–	R	0*	1	

*0 — Resource Available
1 — Resource Not Available

TABLE 4 — RESET STATE OF SEMAPHORE REGISTER

SEM Reg No.	IPC		System	
	Sem	Own	Sem	Own
1	1	1	1	0
2	1	0	1	1
3	1	1	1	0
4	1	1	1	0
5	1	1	1	0
6	1	1	1	0

PROGRAM STORAGE MEMORY — ROM

The standard MC68120 comes preprogrammed with a monitor in the ROM. Custom programs are placed in ROM by special order (see Appendix A).

The MC68120 contains 2048 bytes of on-chip, mask programmable read-only memory (ROM) in memory locations $F800 through $FFFF. The contents of this ROM allows the IPC to perform a custom function for the user. The interrupt vectors $FFF0-$FFFF are decoded to provide vectors at the top of resident ROM. Address $FFEF is reserved for the checksum value for the ROM. This value is the complement of the "Exclusive OR" of the 2047 bytes of mask programmed ROM. An IPC without ROM is also available as the MC68121. The MC68121 should only be used in modes 2 and 3 to access external ROM after reset.

FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

V_{CC} and V_{SS} provide power and ground to the IPC. The power supply should provide +5 volts (±5%) to V_{CC} and V_{SS} should be tied to ground. Total power dissipation should not exceed P_D milliwatts.

RESET

The reset function is used for three purposes. The first is to provide the IPC with an orderly and defined start-up procedure from a powerdown condition. The second is to return to start-up conditions without an intervening powerdown condition. The third is to provide a control signal to latch the operating mode.

During reset (low logic level on $\overline{\text{RESET}}$ pin), execution of the current instruction is suspended and the CPU enters a "reset state." The register contents are not pushed onto the stack and their contents become undefined during reset. The "reset state" initializes the IPC, as shown in Table 5.

On the positive edge of $\overline{\text{RESET}}$, the IPC latches the operating mode from P22, P21 and P20, and then configures Port 3, Port 4, SC1 and SC2. The restart vector is then fetched and transferred to the program counter, then instruction execution begins.

Reset timing is illustrated in Figure 20. The $\overline{\text{RESET}}$ line must be held low for a minimum of three E-cycles for the IPC to complete its entire reset sequence. An external RC-network may be used to obtain the required timing.

ENABLE — E

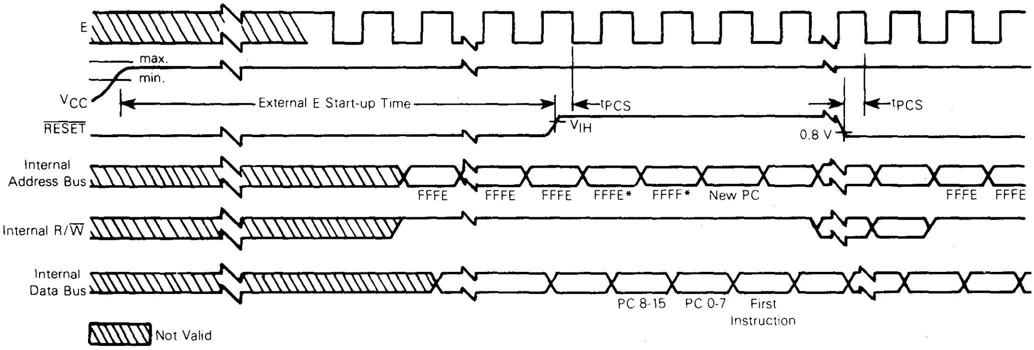
The E clock input is required for timing to synchronize Data Bus transfers. A "CPU E-cycle" (or bus cycle) consists of a negative half-cycle of E followed by a positive half-cycle. For any given bus cycle, the address is valid during the negative half-cycle of E and the selected device must be enabled to the Data Bus during the next positive half-cycle. The data bus is active only while E is high. It should be noted

TABLE 5 — STATE OF IPC DURING RESET

Bits or Registers	Effective State
CPU I-Bit	set (IRQ1 and IRQ2 disabled)
NMI Interrupt Latch	cleared (NMI disabled)
Halt Control Bit	cleared (HALT/BA selected)
All Data Direction Registers	cleared
SCI Rate and Mode Control Register	cleared
Receive Data Register	cleared
Timer Control and Status Register	cleared
Free Running Counter	cleared
Buffer for LSB of Counter	cleared
Port 3 Control and Status Register	cleared
Port 2, 3, 4 Data Registers	undefined after Power-up Reset; and not changed after Reset
SCI Transmit/Receive Control and Status Register	Preset to $20
Output Compare Register	Preset to $FFFF
Semaphore Bits	Preset to 1's
Ownership Bit of Semaphore Register 2	Preset to System Ownership
All other Ownership Bits	Preset to IPC Ownership
All Ports 2 and 3 Lines	High Impedance (inputs)
All Port 4 Lines	High Impedance (inputs) with pullup resistors
SC1*	High Impedance with pullup resistors
SC2	Active High

*If in mode 5, SC1 will go active high; otherwise it will remain in the high impedance state.

FIGURE 20 — RESET TIMING



*Mode 0 — $BFFE, BFFF

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.

that this input should have some provision to obtain the specified logical high level which is greater than standard TTL levels.

Enable is the primary IPC system timing signal and all timing data specified as cycles is assumed to be referenced to this clock unless otherwise noted.

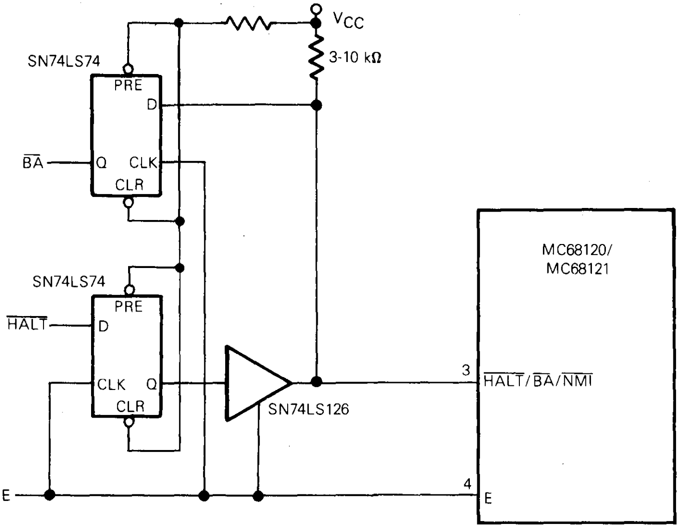
HALT/BUS AVAILABLE/NON-MASKABLE INTERRUPT — $\overline{HALT}/\overline{BA}/\overline{NMI}$

The HALT/BA/NMI (pin 3) serves one of two functions. These functions are $\overline{NMI}$ or Halt/BA and the function selected is determined by the Halt Control (HC, bit 2) bit of the Functional Control Register (location $14). If the HC bit is set (to a "1"), then the $\overline{NMI}$ function is activated. Alternately, if HC is cleared (to a "0" as it is during reset), the Halt/BA function is activated. An external pullup resistor to V_{CC} is required on pin 3 for either function. Typical pullup resistor values range from 3K to 10K depending on the drive capability of the external device.

When the $\overline{NMI}$ function is implemented, pin 3 is configured as an input. A negative edge on pin 3 then requests an IPC non-maskable interrupt sequence, but the current instruction will be completed before responding to this request. To assure an interrupt under all conditions, $\overline{NMI}$ must be held low for at least one E-cycle. $\overline{NMI}$ may be used to cause the IPC to exit the Wait instruction. For interrupt timing specifications, see the interrupt portion of the Operating Mode Section.

When configured to utilize the Halt/BA function of this pin, such as after reset, the circuit of Figure 21 is recommended to detect and supply continuous $\overline{HALT}$ and $\overline{BA}$

FIGURE 21 — $\overline{\mathrm{HALT}}/\overline{\mathrm{BA}}$ DEMULTIPLEXING CIRCUIT

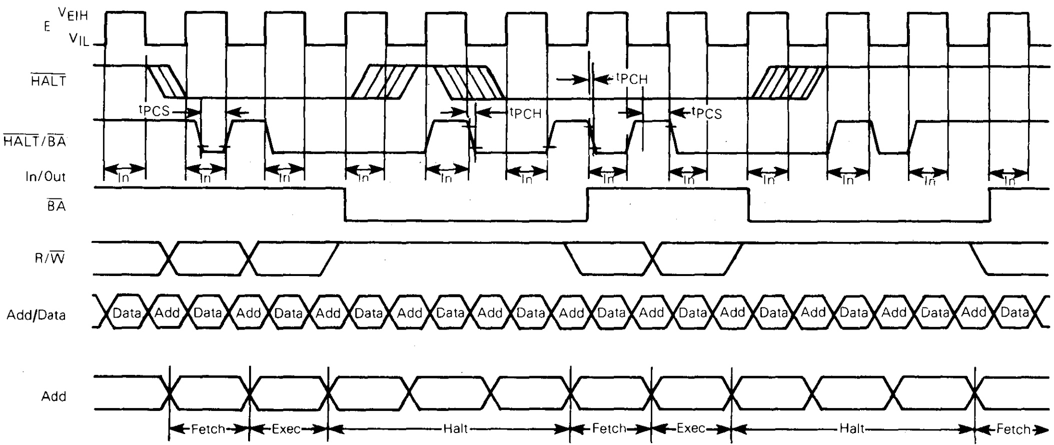


signals. Figure 22 shows the appropriate timing diagram for Halt/BA with the recommended circuit. The pullup resistor shown in the circuit maintains a high logic level when $\overline{\mathrm{HALT}}$ is not active. During a positive half-cycle of E, pin 3 is an input sampled to determine if the Halt State is requested (active low). During the negative half cycle of E, the $\overline{\mathrm{BA}}$ signal is output through pin 3. After the request for Halt State signal is detected and the processor completes its current instruction, the CPU is halted and the active low $\overline{\mathrm{BA}}$ signal is output through pin 3 during the negative half cycle of E. The local bus is then available for other devices to utilize until the Halt State signal has returned to a high level, thus allowing the IPC back on the local bus. During the Halt State, the R/$\overline{\mathrm{W}}$ is high, and the address bus displays the address of the next instruction.

When single instruction operation is desired, in program debug for instance, it is advantageous to single step through instructions. After $\overline{\mathrm{BA}}$ goes low, $\overline{\mathrm{HALT}}$ must be brought high for one E-cycle and returned low again to single step through instructions. Figure 22 illustrates the timing involved while single stepping through a single byte, two bus cycle instruction, such as CLRA.

$\overline{\mathrm{BA}}$ is not output in response to the Wait instruction. If interrupts are to be utilized in removing the processor from a

FIGURE 22 — $\overline{\mathrm{HALT}}/\overline{\mathrm{BA}}$ TIMING DIAGRAM



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.

Wait State while in the Halt/BA mode then, $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$ are the only interrupts which may do so; therefore, their masks must be cleared before entering the Wait State.

MASKABLE INTERRUPT REQUEST 1 – $\overline{\text{IRQ1}}$

This level-sensitive input can be used to request an interrupt sequence. The IPC will complete the current instruction before it responds to the request. If the interrupt mask bit (I-bit) in the Condition Code Register is clear, the IPC will begin an interrupt sequence: a vector is fetched from $FFF8 and $FFF9, transferred to the Program Counter, and instruction execution is continued at the new location. This is explained in greater detail in the Interrupt Section.

$\overline{\text{IRQ1}}$ typically requires an external resistor (3K to 10K depending on external devices drive capability) to V_{CC} for wire-OR applications. $\overline{\text{IRQ1}}$ has no internal pullup resistor.

STROBE CONTROL 1 AND 2 – SC1 and SC2

The functions of SC1 and SC2 depend on the operating mode. SC1 is configured as an input in all modes except the Expanded Non-Multiplexed Mode, whereas SC2 is always an output. SC1 and SC2 can drive one Schottky load and 90 pF.

Single Chip Modes – In these modes, SC1 and SC2 are configured as an input and output, respectively, and both function as Port 3 control lines. SC1 functions as an input strobe (IS3) and can be used to indicate that Port 3 input data is ready or output data has been accepted. Three options associated with $\overline{\text{IS3}}$ are controlled by the Control and Status Register for Port 3 and are discussed in the Port 3 description.

SC2 is configured as an output strobe ($\overline{\text{OS3}}$) and can be used to strobe output data or acknowledge input data for Port 3. It is controlled by Output Strobe Select (OSS) in the Port 3 Control and Status Register. The strobe is generated by a read (OSS = 0) or write (OSS = 1) to the Port 3 Data Register. $\overline{\text{OS3}}$ timing is shown in Figure 6.

Expanded Non-Multiplexed Mode – In this mode, both SC1 and SC2 are configured as outputs. SC1 functions as Input/Output Select (IOS) and is asserted (active-low) only when addresses $0100 through $01FF are accessed. SC2 is configured as R/$\overline{\text{W}}$ and is used to control the direction of local data bus transfers. An MPU read is enabled when R/$\overline{\text{W}}$ and E are high.

Expanded Multiplexed Modes – In these modes, SC1 is configured as an input and SC2 is configured as an output. In the expanded multiplexed modes, the IPC has the ability to access a 64K byte address space. SC1 functions as an input, Address Strobe, which controls demultiplexing and enabling of the eight least significant addresses and the data buses.

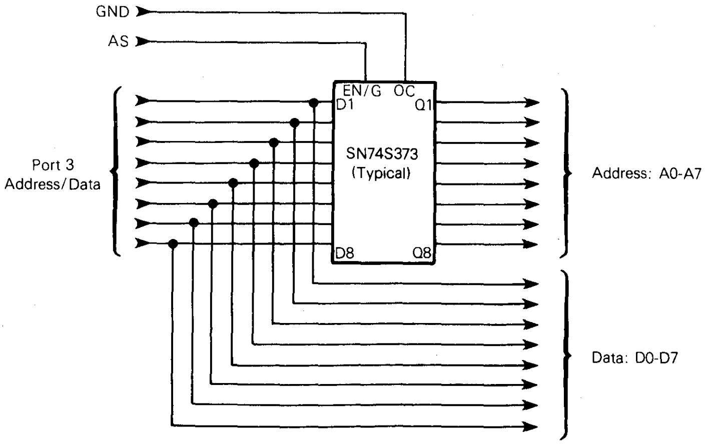
By using a transparent latch such as an SN74LS373 or MC6882, Address Strobe (AS) can also be used to demultiplex the two buses external to the IPC. (See Figure 23.) SC2 provides the local Data Bus control signal called Read/Write (R/$\overline{\text{W}}$). SC2 is configured as R/$\overline{\text{W}}$ and is used to control the direction of local data bus transfers. An MPU read is enabled when R/$\overline{\text{W}}$ and E are high.

SYSTEM BUS INTERFACE

Port 1 is a mode-independent 8-bit data port which permits the external system bus to access the dual-ported RAM and semaphore registers either asynchronously or synchronously with respect to the E clock. In addition to the eight data lines (SD0-SD7), eight address (SA0-SA7) and three control lines (SR/$\overline{\text{W}}$, $\overline{\text{CS}}$, $\overline{\text{DTACK}}$) are used to access the dual-ported RAM and semaphore registers.

Port 1 Data Lines (SD0-SD7) – These data lines are bidirectional data lines which allow data transfer between the dual-ported RAM or the semaphore registers, and the system bus. The data bus output drivers are three-state devices which remain in the high-impedance state except

FIGURE 23 – TYPICAL LATCH ARRANGEMENT



during a read of the IPC dual-ported RAM or semaphore registers by the system processor.

System Address Lines (SA0-SA7) – The address lines together with the Chip Select signal allow any of the 128 bytes of RAM or six semaphore registers to be uniquely selected from the system bus. The address lines must be valid before the $\overline{CS}$ signal goes low for the asynchronous interface and valid before the E signal goes high for the synchronous interface. The system interface must be deselected between reads or between writes for the asynchronous operation.

System Read/Write (SR/$\overline{W}$) – This signal is generated by the system bus to control the direction of data transfer on the data bus. With the IPC selected, a low on the SR/$\overline{W}$ line enables the input buffers, and data is transferred from the system processor to the IPC. When SR/$\overline{W}$ is high and the chip is selected, the data output buffers are turned on and data is transferred from the IPC to the system bus.

Chip Select ($\overline{CS}$) – This signal is a TTL compatible input signal, used to activate the system bus interface and allows transfer of data between the IPC and the system processor during synchronous or asynchronous accesses. $\overline{CS}$ provides the synchronizing signal for the Semaphore registers during access by the system bus.

Data Transfer Acknowledge ($\overline{DTACK}$) – This bidirectional control line is used to determine synchronous or asynchronous system bus accesses and to provide the data acknowledge signal for asynchronous data transfers.

As an input, it is sampled on the falling edge of $\overline{CS}$ by the IPC to determine if the system bus is being accessed synchronously or asynchronously with respect to the E clock.

If $\overline{DTACK}$ is low when sampled, the system bus is synchronous and data will be transferred during E high as shown in Figure 13.

If $\overline{DTACK}$ is high when sampled, the system bus is asynchronous. In this mode $\overline{DTACK}$ becomes an output that is asserted low when data is on the bus during a system read or when a data transfer is completed during a system write. Refer to Figures 9 through 12.

$\overline{DTACK}$ requires an external pullup resistor when the system bus is run asynchronously since it is then a bidirectional handshake line for information transfer on the system data bus.

PORT 2 – P20-P24

Port 2 is a mode independent 5-bit I/O port where each line is configured by its Data Direction Register. During reset, all lines are configured as inputs. The TTL compatible three-state output buffers can drive one Schottky TTL load and 30 pF, or CMOS devices using external pullup resistors. P20, P21 and P22 must always be connected to provide the operating mode.

PORT 2 DATA REGISTER

7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$03

Inputs on P20, P21 and P22 determine the operating mode which is latched into the Program Control Register on the positive edge of $\overline{RESET}$. The mode may be read from the Port 2 Data Register (PC2 is latched from pin 45).

Port 2 also provides an interface for the Serial Communications Interface and Timer. Bit 1, if configured as an output, is dedicated to the Timer Output Compare function and cannot be used to provide output from the Port 2 Data Register.

PORT 3 – P30-P37

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

Single Chip Modes – In these modes, Port 3 is an 8-bit I/O port where each line is configured by the Port 3 Data Direction Register. Associated with Port 3 are two lines, $\overline{IS3}$ and $\overline{OS3}$, which can be used to control Port 3 data transfers.

Three Port 3 options, controlled by the Port 3 Control and Status Register and available only in the Single Chip Modes are: 1) Port 3 input data can be latched using $\overline{IS3}$ as a control signal, 2) $\overline{OS3}$ can be generated by either an IPC read or write to the Port 3 Data Register, and 3) an $\overline{IRQ1}$ interrupt can be enabled by an $\overline{IS3}$ negative edge. Port 3 latch timing is shown in Figure 7.

PORT 3 CONTROL AND STATUS REGISTER

7	6	5	4	3	2	1	0	
IS3 FLAG	IS3 IRQ1 ENABLE	X	OSS	LATCH ENABLE	X	X	X	$0F

Bits 0-2 Not used.

Bit 3 LATCH ENABLE. This bit controls the input latch for Port 3. If set, input data is latched by an $\overline{IS3}$ negative edge. The latch is transparent after a read of the Port 3 Data Register. LATCH ENABLE is cleared by Reset.

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

Bit 5 Not used.

Bit 6 $\overline{IS3}$-$\overline{IRQ1}$ ENABLE. When set, an $\overline{IRQ1}$ interrupt will be enabled whenever $\overline{IS3}$ FLAG is set; when clear, the interrupt is inhibited. This bit is cleared by Reset.

Bit 7 $\overline{IS3}$ FLAG. This read-only status bit is set by an $\overline{IS3}$ negative edge. It is cleared by a read of the Port 3 Control and Status Register (with $\overline{IS3}$ FLAG set) followed by a read or write to the Port 3 Data Register or by Reset.

Expanded Non-Multiplexed Mode – In this mode, Port 3 is configured as a bi-directional data bus (D0-D7). The direction of data transfers is controlled by R/$\overline{W}$ (SC2). Data transfers are clocked by E (Enable).

Expanded Multiplexed Modes — In these modes, Port 3 is configured as a time-multiplexed address (A0-A7) and data bus (D0-D7). Address Strobe (AS) must be input on SC1, and can be used externally to de-multiplex the two buses. Port 3 is held in a high-impedance state between valid address and data to prevent potential bus conflicts.

PORT 4 — P40-P47

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

Single Chip Modes — In these modes, Port 4 functions as an 8-bit I/O port where each line is configured by the Port 4 Data Direction Register. Internal pullup resistors allow the port to directly interface with CMOS at 5 volt levels. External pullup resistors to more than 5 volts, however, cannot be used.

Expanded Non-Multiplexed Mode — In this mode, Port 4 is configured from reset as an 8-bit input port, where the Data Direction Register can be written, to provide any or all of address lines A0-A7. Internal pullup resistors are intended to pull the lines high until the Data Direction Register is configured.,

Expanded Multiplexed Mode — In all these modes except Mode 6, Port 4 functions as half of the address bus and provides A8 to A15. In Mode 6, the port is configured from reset as an 8-bit parallel input port; the Port 4 Data-Direction Register must be written to provide any or all of address lines, A8 to A15. Internal pullup resistors are intended to pull the lines high until the Data Direction Register is configured (bit 0 controls A8, etc.).

OPERATING MODES

The IPC provides eight different operating modes which are selectable by hardware programming and referred to as Modes 0 through 7. The operating mode controls the memory map, configuration of Port 3, Port 4, SC1 and SC2 and the address location of the interrupt vectors.

FUNDAMENTAL MODES

The eight modes of the IPC can be grouped into three fundamental modes which refer to the type of bus it supports: Single Chip, Expanded Non-Multiplexed, and Expanded Multiplexed. Single Chip includes Modes 4 and 7, Expanded Non-Multiplexed is Mode 5 and the remaining five are Expanded Multiplexed modes. A system utilizing three MC68120's, one in each of the fundamental operating modes, is shown in Figure 24. Table 6 summarizes the characteristics of the operating modes.

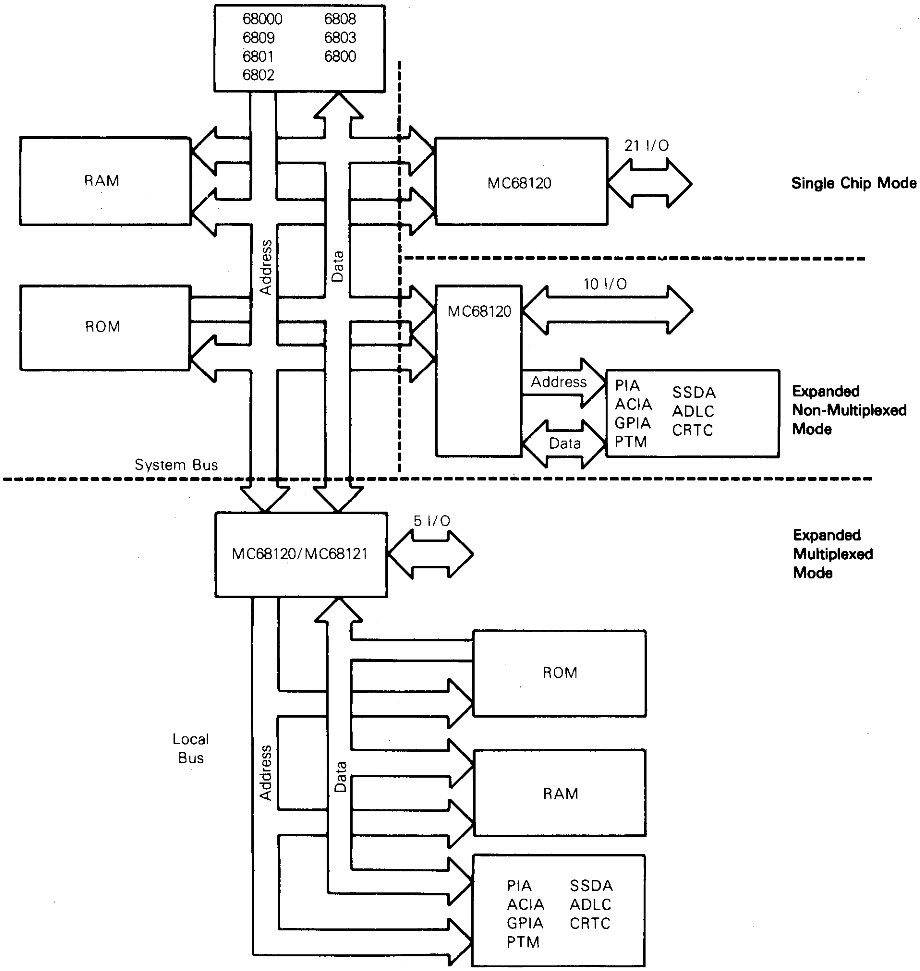
Single Chip Modes (4, 7) — In Single Chip Mode, three of the four IPC ports are configured as parallel input/output data ports, as shown in Figure 25. The IPC functions as a complete microcomputer in these two modes without external address or data buses. A maximum of 21 I/O lines and two Port 3 control lines are provided.

In Single Chip Test Mode (4), the RAM responds to addresses $XX80 (X = don't care) through $XXFF and the ROM is removed from the internal address map. A test program must first be loaded into the RAM using Modes 0, 1, 2, or 6. If the IPC is reset and then programmed into Mode 4, execution will begin at $XXFE:XXFF. Mode 5 can be irreversibly entered from Mode 4 without going through reset by setting bit 5 of the Port 2 Data Register. This mode is used primarily to test Port 3 and 4 in the Single Chip and Non-Multiplexed Modes.

TABLE 6 — SUMMARY OF IPC OPERATING MODES

Common to all Modes: System Bus Interface Reserved Register Area 6 Semaphore Registers I/O Port 2 Programmable Timer Serial Communications Interface 128 bytes of Dual Ported RAM	Expanded Multiplexed Modes Four Memory Space Options (64K Address Space): (1) MDOS Compatible (2) No ROM (3) External Vector Space (4) ROM with Partial Address Bus* External Memory Space Accessed Through: Port 3 as a Multiplexed Address/Data Bus Port 4 as an Address Bus (High) SC1 is Address Strobe Bus (AS) Input SC2 is Read/Write (R/$\overline{W}$)
Single Chip Mode* 2048 Bytes of ROM (Internal) Port 3 is a Parallel I/O Port with Two Control Lines Port 4 is a Parallel I/O Port SC1 is Input Strobe 3 ($\overline{IS3}$) SC2 is Output Strobe 3 ($\overline{OS3}$)	Test Modes Expanded Multiplexed Test Mode May be Used to Test RAM and ROM* Single Chip and Non-Multiplexed Test Mode* May be Used to Test Ports 3 and 4 as I/O Ports
Expanded Non-Multiplexed Mode* 2048 Bytes of ROM (Internal) 256 Bytes of External Memory Space Port 3 is an 8-Bit Data Bus Port 4 is an Address Bus SC1 is Input/Output Select ($\overline{IOS}$) SC2 is Read/Write (R/$\overline{W}$)	*MC68120 only

FIGURE 24 — IPC FUNDAMENTAL OPERATING MODES



Expanded Non-Multiplexed Mode (5) — A modest amount of external memory space is provided in the Expanded Non-Multiplexed Mode while retaining significant on-chip resources. Port 3 functions as an 8-bit bi-directional data bus and Port 4 is configured as an input data port. Any combination of A0 to A7 may be provided while retaining the remainder as input data lines. Any combination of the eight least-significant address lines may be obtained by writing to the Port 4 Data Direction Register. Internal pullup resistors are provided to pull Port 4 lines high until it is configured.

Figure 26 illustrates the external resources available in the Expanded Non-Multiplexed Mode. The IPC interfaces directly with M6800 Family parts and can access 256 bytes of external address space at $100 through $1FF. IOS provides an address decode of external memory ($100-$1FF) and may be used as an address or chip select line.

Expanded-Multiplexed Modes (0, 1, 2, 3, 6) — In the Expanded Multiplexed Modes, the IPC has the ability to access a 64K-byte memory space. Port 3 functions as a time-multiplexed address/data bus with address valid on the negative edge of Address Strobe (AS) and the data bus valid while E is high. In Modes 0 to 3, Port 4 provides address lines A8-A15. However, in Mode 6, Port 4 can provide any subset of A8 to A15 while retaining the remainder as input lines. Writing 1's to the desired bits in the Data Direction Register (DDR) will output the corresponding address lines while the remaining bits will remain inputs (as configured from reset or from 0's written to the DDR). Internal pullup resistors are provided to pull Port 4 lines high until software configures the port. Initialization of Port 4 in Mode six must be done to obtain any upper address lines externally.

FIGURE 25 — SINGLE CHIP MODE

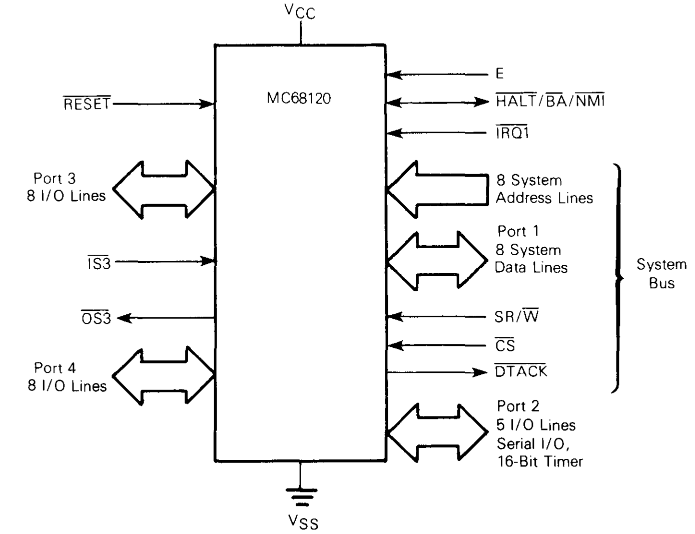


FIGURE 26 — EXPANDED NON-MULTIPLEXED MODE

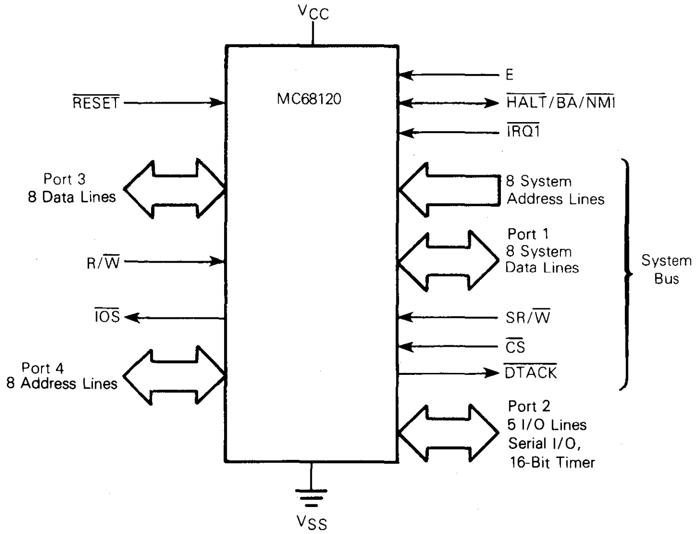
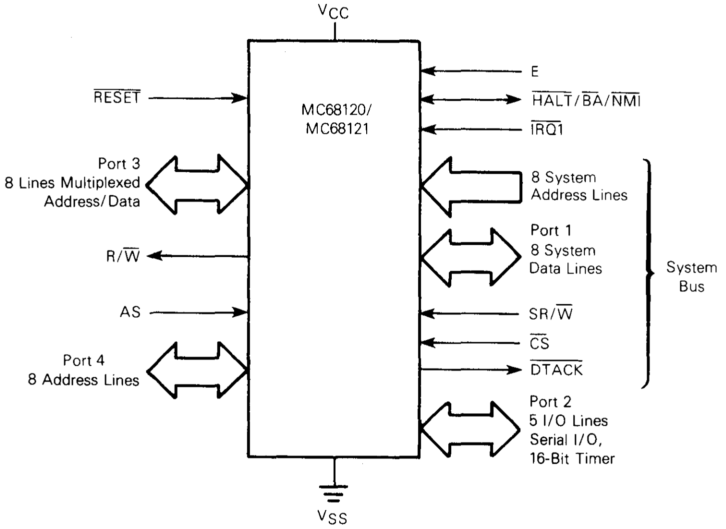


Figure 27 depicts the external resources available in the Expanded-Multiplexed Modes. Address Strobe can be used to control a transparent D-type latch to capture addresses A0-A7, as shown in Figure 23. This allows Port 3 to function as a Data Bus when E is high.

In Mode 0, the reset vector is external at \$BFFE and \$BFFF after the positive edge of RESET. In addition, the internal and external data buses are connected together so there must be no memory map overlap (to avoid potential bus conflicts). Mode 0 is used primarily to verify the ROM pattern and monitor the internal data bus with automated test equipment.

FIGURE 27 — EXPANDED MULTIPLEXED MODE



MODE PROGRAMMING

The operating mode is programmed by the levels asserted on P22, P21, and P20 during the positive edge of RESET. These are latched into PC2, PC1, and PC0 of the program control register. The operating mode may be read from the Port 2 Data Register and programming levels and timing must be met as shown in Figure 28 and Table 7. Any mode may be entered from either Mode 0 or Mode 4 without going through reset by writing the appropriate bits to the port 2 data register. A brief outline of the operating modes is shown in Table 8.

Circuitry to provide the programming levels is primarily dependent on the normal system use of the three pins. If configured as outputs, the circuit shown in Figure 29 may be used; otherwise, the three-state buffers can be used to provide isolation while programming the mode.

MEMORY MAPS

The IPC provides up to 64K bytes of address space depending upon the operating mode. A memory map for each operating mode is shown in Figure 30. In Modes 1R and 6R, the "R" means the ROM has been relocated by a mask option. The first 32 locations of each map are reserved for the IPC internal register area, as shown in Table 9, with exceptions as indicated.

FIGURE 28 — MODE PROGRAMMING TIMING

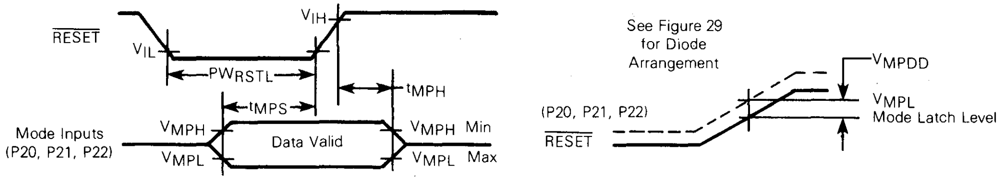


TABLE 7 — MODE PROGRAMMING SPECIFICATIONS (See Figure 30)

Characteristic	Symbol	Min	Typ	Max	Unit
Mode Programming Input Voltage Low	V_{MPL}	–	–	1.8	V
Mode Programming Input Voltage High	V_{MPH}	4.0	–	–	V
Mode Programming Diode Differential (if Diodes are Used)	V_{MPDD}	0.6	–	–	V
$\overline{\text{RESET}}$ Low Pulse Width	PW_{RSTL}	3.0	–	–	E-Cycles
Mode Programming Setup Time	t_{MPS}	2.0	–	–	E-Cycles
Mode Programming Hold Time					
$\overline{\text{RESET}}$ Rise Time ≥ 1 μs	t_{MPH}	0	–	–	ns
$\overline{\text{RESET}}$ Rise Time < 1 μs		100	–	–	

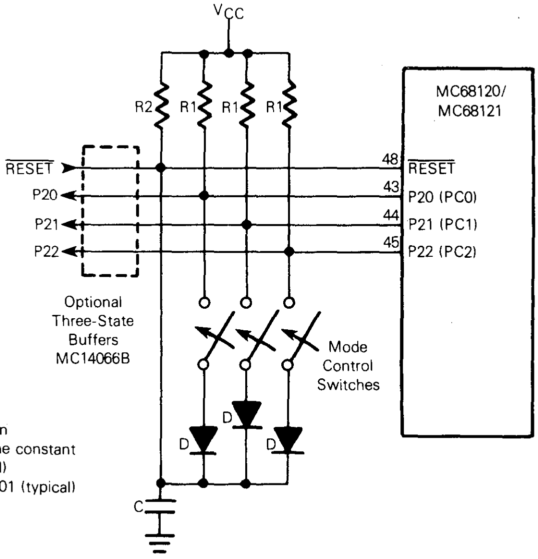
TABLE 8 — MODE SELECTION SUMMARY

Mode	Pin 45 P22 PC2	Pin 44 P21 PC1	Pin 43 P20 PC0	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(5, 6)	Multiplexed/Partial Decode(5)
5	H	L	H	I	I	I	NMUX(5, 6)	Non-Multiplexed/Partial Decode(5)
4	H	L	L	I(2)	I(1)	I	I	Single Chip Test
3	L	H	H	E	I(7)	E	MUX(4)	Multiplexed/RAM(4)
2	L	H	L	E	I	E	MUX(4)	Multiplexed/RAM(4)
1	L	L	H	I	I	E	MUX(4)	Multiplexed/RAM and ROM(4)
0	L	L	L	I	I	E(3)	MUX(4)	Multiplexed Test(4)

Legend:
I — Internal
E — External
MUX — Multiplexed
NMUX — Non-Multiplexed
L — Logic "0"
H — Logic "1"

Notes:
(1) Internal RAM is addressed at $XX80
(2) Internal ROM is disabled
(3) Interrupt vectors externally located at $BFF0-$BFFF
(4) Addresses associated with Ports 3 and 4 are considered external in Modes 0, 1, 2, and 3
(5) Addresses associated with Port 3 are considered external in Modes 5 and 6
(6) Port 4 default is user data input; address output is optional by writing to Port 4 Data Direction Register
(7) Internal RAM and registers located at $C0XX (for use with MDOS)

FIGURE 29 — TYPICAL MODE PROGRAMMING CIRCUIT

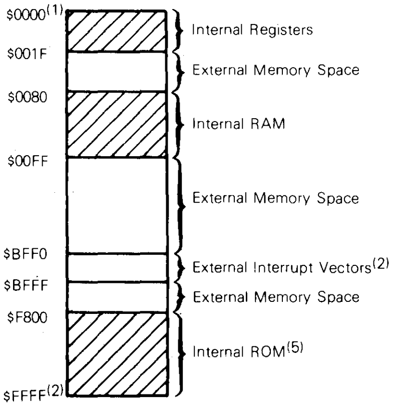


Notes:
1. Mode 7 as shown
2. R2•C = Reset time constant
3. R1 = 10 k (typical)
4. D = 1N914, 1N4001 (typical)

FIGURE 30 — IPC MEMORY MAPS

MC68120 Mode 0

Multiolexed Test Mode

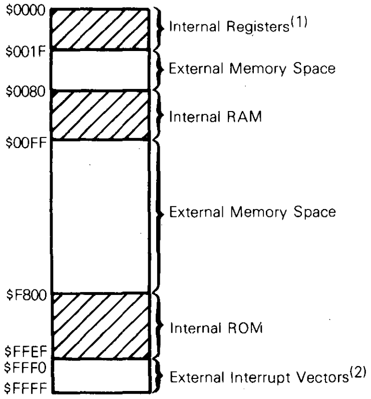


Notes:

1) Excludes the following addresses which may be used externally: $04, $05, $06, $07 and $0F.
2) The interrupt vectors are externally located at $BFF0-$BFFF.
3) There must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.
4) This mode is the only mode which may be used to examine the interrupt vectors in internal ROM using an external $\overline{\text{RESET}}$ vector.
5) MC68120 only.

MC68120 Mode 1

Multiplexed/RAM and ROM



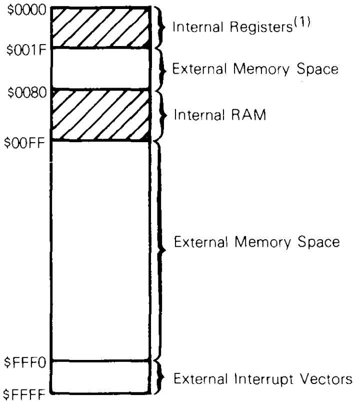
Notes:

1) Excludes the following addresses which may be used externally: $04, $05, $06, $07 and $0F.
2) Internal ROM addresses $FFF0 to $FFFF are not usable.

FIGURE 30 — IPC MEMORY MAPS (CONTINUED)

MC68120/
MC68121
Mode **2**

Multiplexed/RAM

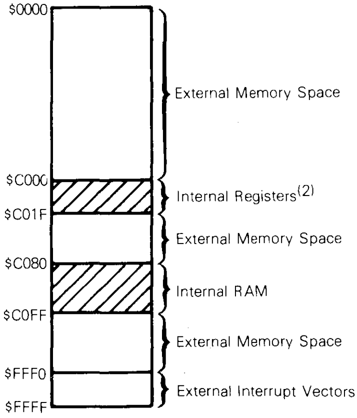


Notes:

1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.

MC68120/
MC68121
Mode **3**

Multiplexed/RAM, MDOS Compatible (1)

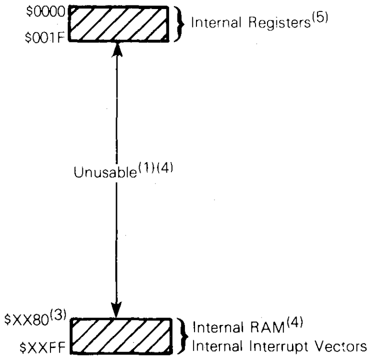


Notes:

1) Relocating the internal registers and the internal RAM to high memory allows processor to run MDOS.
2) Excludes the following addresses which may be used externally: $C004, $C005, $C006, $C007, and $C00F.

MC68120
Mode **4**

Single Chip Test(2)

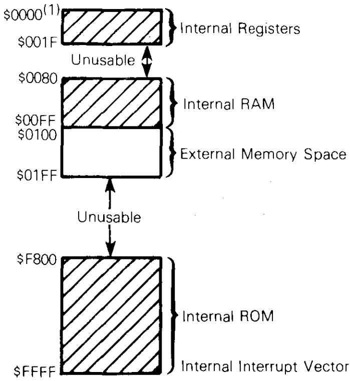


Notes:

1) The internal ROM is disabled.
2) Mode 4 may be changed to Mode 5 without having to assert $\overline{\text{RESET}}$ by writing a "1" into bit 5 (PCO) of Port 2 Data Register.
3) Addresses A8 to A15 are treated as "don't cares" to decode internal RAM.
4) Internal RAM will appear at $XX80 to $XXFF.
5) MPU Read of Port 3 Data Direction Register will access Port 3 Data Register instead.

MC68120
Mode **5**

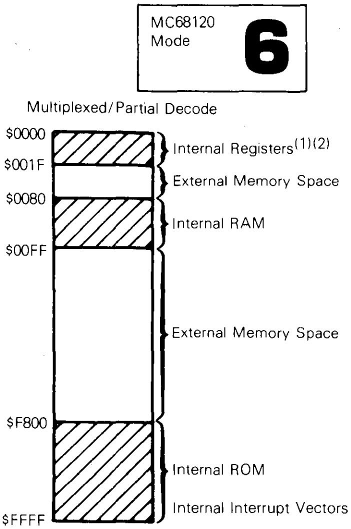
Non-Multiplexed/Partial Decode (2) (3)



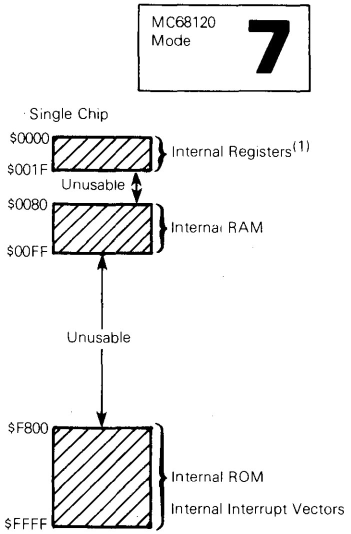
Notes:

1) Excludes the following addresses which may not be used externally: $04, $06, and $0F (no $\overline{\text{IOS}}$).
2) This mode may be entered without going through Reset by using Mode 4 and subsequently writing a "1" into bit 5 (PCO) of Port 2 Data Register.
3) Address lines A0 to A7 will not contain addresses until the Data Direction Register for Port 4 has been written with "1's" in the appropriate bits. These address lines will assert "1's" until made outputs by writing the Data Direction Register.

FIGURE 30 — IPC MEMORY MAPS (CONCLUDED)



Notes:
1) Excludes the following addresses which may be used externally: $04, $06, $0F.
2) Address lines A8-A15 will not contain addresses until the Data Direction Register for Port 4 has been written with "1's" in the appropriate bits. These address lines will assert "1's" until made outputs by writing the Data Direction Register.



Notes:
1) MPU reads of Port 3's Data Direction Register will access Port 3's Data Register instead.

TABLE 9 — INTERNAL REGISTER AREA

Register	Address * * * * (Hexadecimal)	Register	Address * * * * (Hexadecimal)
Reserved	00	SCI Rate and Mode Control Register	10
Port 2 Data Direction Register* * *	01	Transmit/Receive Control and Status Register	11
Reserved	02	SCI Receive Data Register	12
Port 2 Data Register	03	SCI Transmit Data Register	13
Port 3 Data Direction Register* * *	04*		
Port 4 Data Direction Register* * *	05* *	Function Control Register	14
Port 3 Data Register	06*	Counter Alternate Address (High Byte)	15
Port 4 Data Register	07* *	Counter Alternate Address (Low Byte)	16
Timer Control and Status Register	08	Semaphore 1	17
Counter (High Byte)	09	Semaphore 2	18
Counter (Low Byte)	0A	Semaphore 3	19
Output Compare Register (High Byte)	0B	Semaphore 4	1A
Output Compare Register (Low Byte)	0C	Semaphore 5	1B
Input Capture Register (High Byte)	0D	Semaphore 6	1C
Input Capture Register (Low Byte)	0E	Reserved	1D-1F
Port 3 Control and Status Register	0F*		

*These external addresses in Modes 0, 1, 2, 3, 5, 6 cannot be accessed in Mode 5 (no IOS).

* *These are external addresses in Modes 0, 1, 2, 3.

* * *1 = Output, 0 = Input

* * * *These addresses relocated at $C000-$C01F in Mode 3.

INTERRUPTS

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

All $\overline{\mathrm{IRQ2}}$ interrupts use hardware prioritized vectors. The single SCI interrupt and three timer interrupts are serviced in a prioritized order where each is vectored to a separate location. All IPC vector locations are shown in Table 10, from highest (top) to lowest (bottom) priority.

The interrupt flowchart is depicted in Figure 31. The Program Counter, Index Register, Accumulator A, Accumulator B, and Condition Code Register are pushed to the stack. The I-bit is set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the Program Counter and instruction execution is resumed. The general interrupt timing sequence is shown in Figure 32. The Interrupt HALT/$\overline{\mathrm{BA}}$ timing is illustrated in Figure 21 and 22.

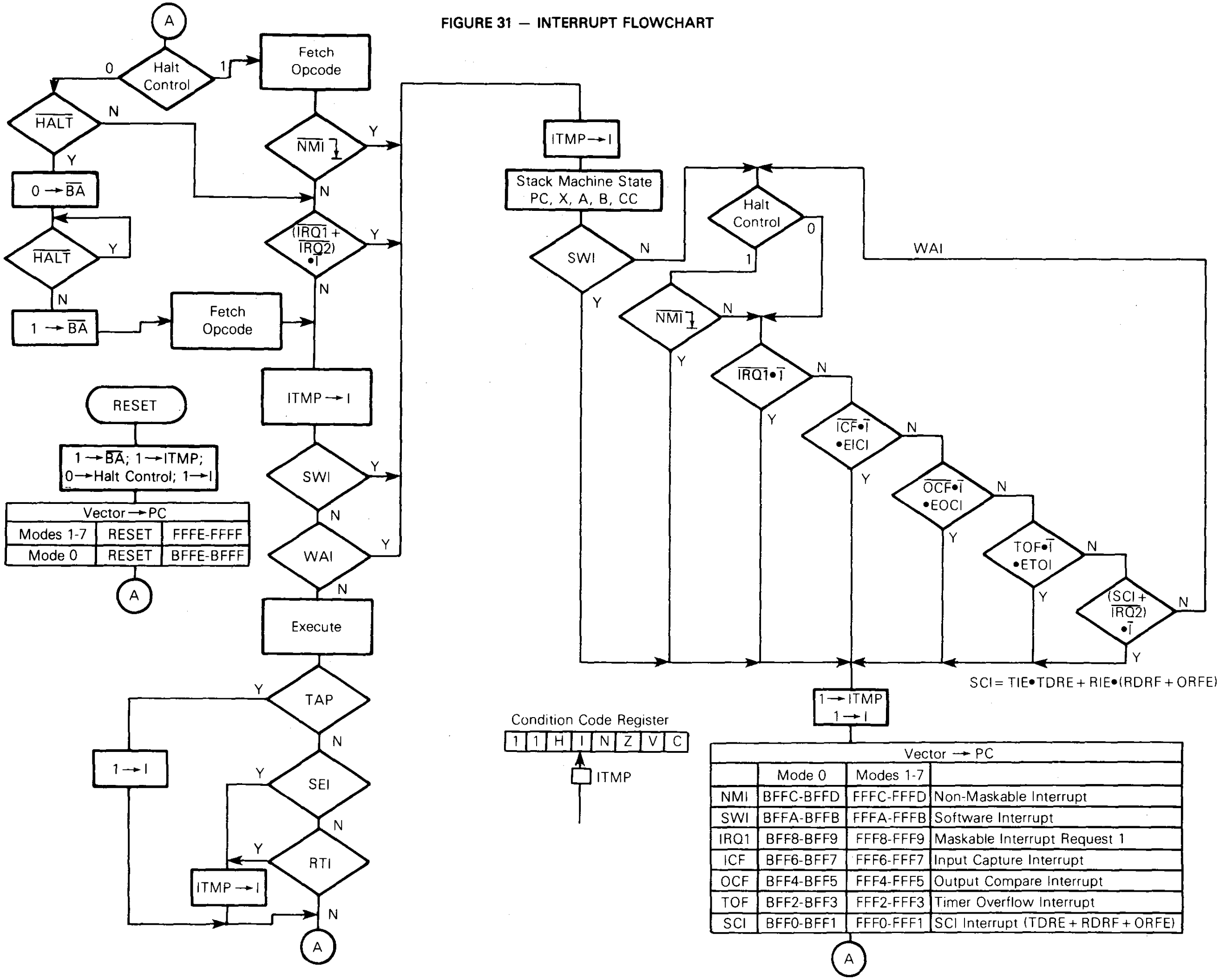
TABLE 10 — MCU VECTOR LOCATIONS *

MSB	LSB	Interrupt
$FFFE	FFFF	$\overline{\mathrm{RESET}}$* *
FFFC	FFFD	$\overline{\mathrm{NMI}}$
FFFA	FFFB	Software Interrupt (SWI)
FFF8	FFF9	$\overline{\mathrm{IRQ1}}$ (or IS3)
FFF6	FFF7	ICF (Input Capture)
FFF4	FFF5	OCF (Output Compare)
FFF2	FFF3	TOF (Timer Overflow)
FFF0	FFF1	SCI (RDRF + ORFE + TDRE)

*These locations are relocated at $BFF0-$BFFF in Mode 0.

* *Highest priority.

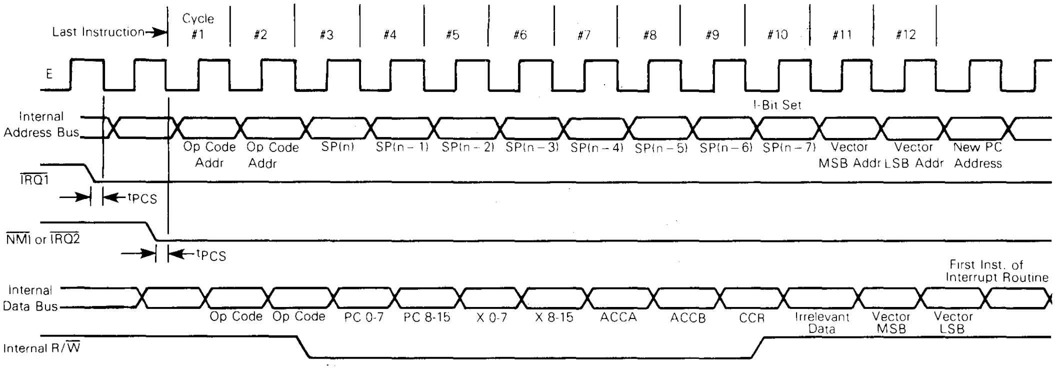
FIGURE 31 — INTERRUPT FLOWCHART



Vector → PC		
Modes 1-7	RESET	FFFE-FFFF
Mode 0	RESET	BFFE-BFFF

Vector → PC			
	Mode 0	Modes 1-7	
NMI	BFFC-BFFD	FFFC-FFFD	Non-Maskable Interrupt
SWI	BFFA-BFFB	FFFA-FFFB	Software Interrupt
IRQ1	BFF8-BFF9	FFF8-FFF9	Maskable Interrupt Request 1
ICF	BFF6-BFF7	FFF6-FFF7	Input Capture Interrupt
OCF	BFF4-BFF5	FFF4-FFF5	Output Compare Interrupt
TOF	BFF2-BFF3	FFF2-FFF3	Timer Overflow Interrupt
SCI	BFF0-BFF1	FFF0-FFF1	SCI Interrupt (TDRE + RDRF + ORFE)

FIGURE 32 — INTERRUPT SEQUENCE



PROGRAMMABLE TIMER

The Programmable Timer can be used to perform input waveform measurements while independently generating an output waveform. Pulse widths can vary from several microseconds to many seconds. A block diagram of the Timer is shown in Figure 33.

TIMER CONTROL AND STATUS REGISTER ($08)

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

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

Each of the three events can generate an $\overline{\mathrm{IRQ2}}$ interrupt and is controlled by an individual enable bit in the TCSR.

TIMER CONTROL AND STATUS REGISTER
(TSCR)

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$08

Bit 0 OLVL Output level. OLVL is clocked to the output level register by a successful output compare and will appear at P21 if Bit 1 of the Port 2 Data Direction Register is set. It is cleared by reset.

Bit 1 IEDG Input Edge. IEDG is cleared by reset and controls which level transition will trigger a counter transfer to the Input Capture Register:

IEDG = 0 Transfer on a negative edge
IEDG = 1 Transfer on a positive edge

Bit 2 ETOI Enable Timer Overflow Interrupt. When set, an $\overline{\mathrm{IRQ2}}$ interrupt is enabled for a timer overflow; when clear, the interrupt is inhibited. It is cleared by reset.

Bit 3 EOCI Enable Output Compare Interrupt. When set, an $\overline{\mathrm{IRQ2}}$ interrupt is enabled for an output compare; when clear, the interrupt is inhibited. It is cleared by reset.

Bit 4 EICI Enable Input Capture Interrupt. When set, an $\overline{\mathrm{IRQ2}}$ interrupt is enabled for an input capture; when clear, the interrupt is inhibited. It is cleared by reset.

Bit 5 TOF Timer Overflow Flag. TOF is set when the counter contains all 1's. It is cleared by reading the TCSR (with TOF set) followed by reading the highest byte of the counter ($09), or by reset. Reading the counter at $15 will not clear TOF.

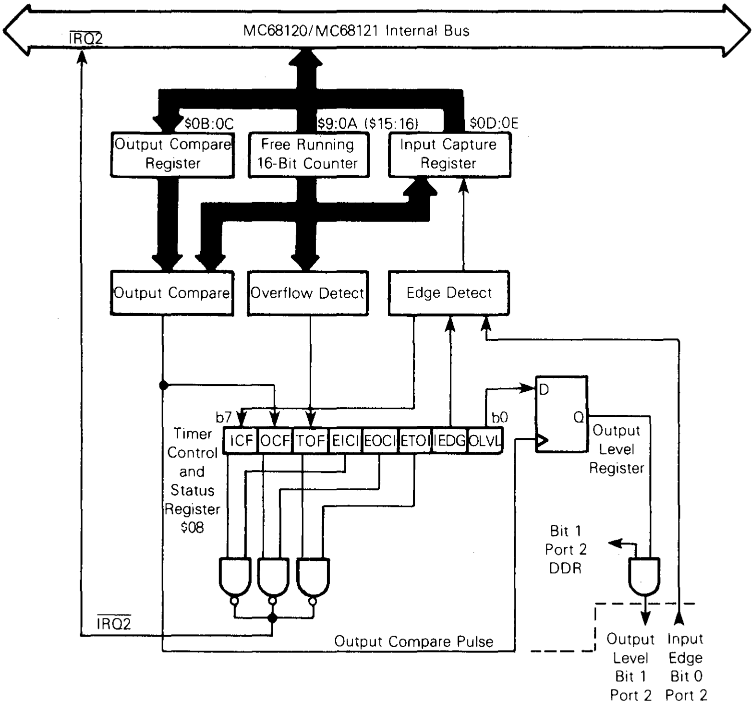
Bit 6 OCF Output Compare Flag. OCF is set when the Output Compare Register matches the free-running counter. It is cleared by reading the TCSR (with OCF set) and then writing to the Output Compare Register ($0B or $0C), or by reset.

Bit 7 ICF Input Capture Flag. ICF is set to indicate a proper level transition. It is cleared by reading the TCSR (with ICF set) and then reading the Input Capture Register High Byte ($0D), or by reset.

COUNTER ($09:0A)

The key timer element is a 16-bit free-running counter which is incremented by E (Enable). It is cleared during reset and is a read-only with one exception: a write to the counter ($09) will preset it to $FFF8. This feature, intended for testing, can disturb serial operations because the counter provides the SCI internal bit rate clock. TOF is set whenever the counter contains all 1's. The counter may also be read at location $15 and $16 to avoid the clearing of the TOF.

FIGURE 33 — PROGRAMMABLE TIMER — BLOCK DIAGRAM



OUTPUT COMPARE REGISTER ($0B:0C)

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

INPUT CAPTURE REGISTER ($0D:0E)

The Input Capture Register is a 16-bit read-only register used to store the free-running counter when a "proper" input transition occurs as defined by IEDG. Port 2, bit 0 should be configured as an input, but the edge detect circuit always senses P20, even when configured as an output. An input capture can occur independently of ICF: the input capture register always contains the most current value regardless of whether ICF was previously set or not. Counter transfer is inhibited, however, between accesses of a double byte IPC read. The input pulse width must be at least two E-cycles to ensure an input capture under all conditions.

SERIAL COMMUNICATIONS INTERFACE (SCI)

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

WAKE-UP FEATURE

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

PROGRAMMABLE OPTIONS

The following features of the SCI are programmable:

- format: standard mark/space (NRZ) or Bi-phase

- clock: external or internal clock source
- Baud rate: one of four per E-clock frequency, or one-eighth of the external clock input to P22
- wake-up features: enabled or disabled
- interrupt requests: enabled individually for transmitter and receiver
- clock output: internal bit rate clock enabled or disabled to P22

SERIAL COMMUNICATIONS REGISTERS

The Serial Communications Interface includes four addressable registers as depicted in Figure 34. It is controlled by the Rate and Mode Control Register and the Transmit/Receive Control and Status Register. Data is transmitted and received utilizing a write-only Transmit Register and read-only Receive Register. The shift registers are not accessible by software.

Rate and Mode Control Register ($10) — The Rate and Mode Control Register (RMCR) controls the SCI Baud rate, format, clock source, and under certain conditions, the configuration of P22. The register consists of four write-only bits which are cleared by reset. The two least significant bits control the Baud rate of the internal clock and the remaining two bits control the format and clock source.

RATE AND MODE CONTROL REGISTER (RMCR)

7	6	5	4	3	2	1	0	
X	X	X	X	CC1	CC0	SS1	SS0	$10

Bit 1: Bit 0 SS1:SS0 Speed Select. These two bits select the Baud rate when using the internal clock. Four rates may be selected which are a function of the IPC input frequency (E). Table 11 lists bit times and rates for three selected IPC frequencies.

Bit 3: Bit 2 CC1:CC0 Clock Control and Format Select. These two bits control the format and select the serial clock source. If CC1 is set, the Data Direction Register (DDR) value for P22 is forced to the complement of CC0 and cannot be altered until CC1 is cleared. If CC1 is cleared after having been set, its DDR value is unchanged. Table 12 defines the format, clock source, and use of P22.

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

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

FIGURE 34 — SCI REGISTERS

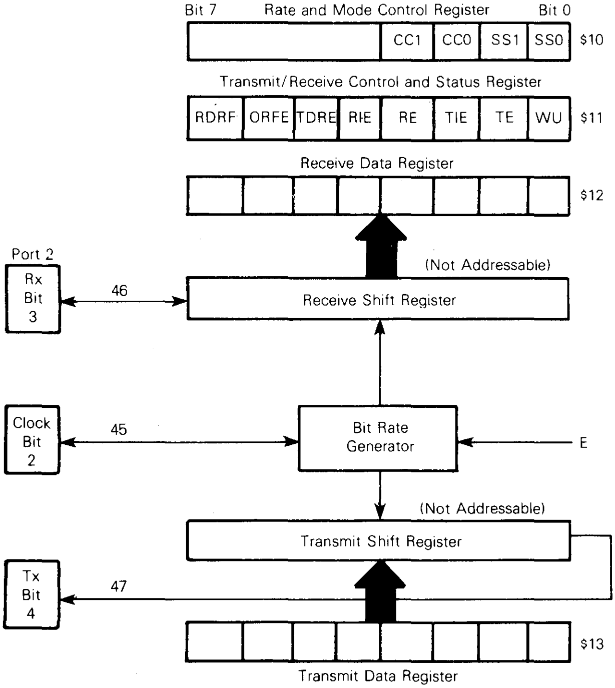


TABLE 11 — SCI BIT TIMES AND RATES

SS1:SS0	E	614.4 kHz	1.0 MHz	1.2288 MHz
0 0	÷16	26 μs/38,400 Baud	16 μs/62,500 Baud	13.0 μs/76,800 Baud
0 1	÷128	208 μs/4,800 Baud	128 μs/7812.5 Baud	104.2 μs/9,600 Baud
1 0	÷1024	1.67 ms/600 Baud	1.024 ms/976.6 Baud	833.3 μs/1,200 Baud
1 1	÷4096	6.67ms/150 Baud	4.096 ms/244.1 Baud	3.33 ms/300 Baud

TABLE 12 — SCI FORMAT AND CLOCK SOURCE CONTROL

CC1:CC0	Format	Clock Source	Port 2 Bit 2
0 0	Bi-Phase	Internal	Not Used
0 1	NRZ	Internal	Not Used
1 0	NRZ	Internal	Output
1 1	NRZ	External	Input

Transmit/Receive Control and Status Register ($11) — The Transmit/Receive Control and Status Register (TRCSR) controls the transmitter, receiver, wake-up features, and two individual interrupts and monitors the status of serial operations. All eight bits are readable while only bits 0 to 4 are writable. The register is initialized to $20 by reset.

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR)

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	$11

Bit 0 WU "Wake-up" on Idle Line. When set, WU enables the wake-up function; it is cleared by ten consecutive 1's or by reset. WU will not set if the line is idle.

Bit 1 TE Transmit Enable. When set, the P24 DDR bit is set, cannot be changed, and will remain set if TE is subsequently cleared. When TE is changed from clear to set, the transmitter is connected to P24 and a preamble of nine consecutive 1's is transmitted. TE is cleared by reset.

Bit 2 TIE Transmit Interrupt Enable. When set, an $\overline{\text{IRQ2}}$ interrupt is enabled when TDRE is set; when clear, the interrupt is inhibited. TIE is cleared by reset.

Bit 3 RE Receive Enable. When set, the P23 DDR bit is cleared, cannot be changed, and will remain clear if RE is subsequently cleared. While RE is set, the SCI receiver is enabled. RE is cleared by reset.

Bit 4 RIE Receiver Interrupt Enable. When set, an $\overline{\text{IRQ2}}$ interrupt is enabled when RDRF and/or ORFE is set; when clear, the interrupt is inhibited. RIE is cleared by reset.

Bit 5 TDRE Transmit Data Register Empty. TDRE is set when the contents of the Transmit Data Register is transferred to the output serial shift register or by reset. It is cleared by reading the TRCSR (with TDRE set) and then writing to the Transmit Data Register. Additional data will be transmitted only if TDRE has been cleared.

Bit 6 ORFE Overrun Framing Error. If set, ORFE indicates either an overrun or framing error. An overrun occurs when a new byte is ready to transfer to the Receiver Data Register with RDRF still set. A receiver framing error has occurred when the byte boundaries of the bit stream are not synchronized to the bit counter. An overrun can be distinguished from a framing error by the value of RDRF: if RDRF is set, then an overrun has occurred; otherwise, a framing error has been detected. Data is not transferred to the Receive Data Register in an overrun condition. ORFE is cleared by reading the TRCSR (with ORFE set) then reading the Receive Data Register, or by reset.

Bit 7 RDRF Receive Data Register Full. RDRF is set when the contents of the input serial shift register is transferred to the Receive Data Register. It is cleared by reading the TRCSR (with RDRF set), and then reading the Receive Data Register, or by reset.

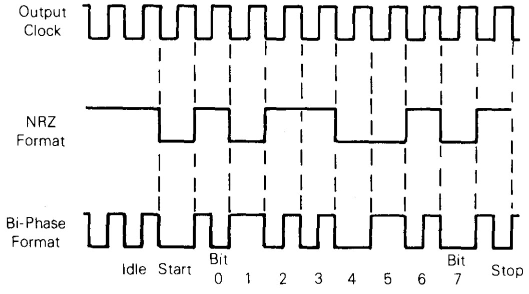
SERIAL OPERATIONS

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

At this point one of two situations exist: 1) if the Transmit Data Register is empty (TDRE = 1), a continuous string of 1's will be sent indicating an idle line, or 2) if a byte has been written to the Transmit Data Register (TDRE = 0), the byte will be transferred to the Transmit Shift Register (synchronized with the bit rate clock), TDRE will be set, and transmission will begin.

The start bit (0), eight data bits (beginning with bit 0) and a stop bit (1), will be transmitted. If TDRE is still set when the next byte transfer should occur, 1's will be sent until more data is provided. Receive operation is controlled by RE which configures P23 as an input and enables the receiver. In Bi-phase format, the output toggles at the start of each bit and at half time when a "1" is sent. SCI data formats are illustrated in Figure 35. In receiving Bi-phase, a "1" is input when two transitions occur in less than 3/4 bit-time, and a "0" is input when more than 3/4 bit-time passes after a transition on P23.

FIGURE 35 — SCI DATA FORMATS



Data: 01001101 ($4D)

INSTRUCTION SET

The MC68120/MC68121 is upward source and object code compatible with the MC6800 processor and directly compatible with the M6801 Family processors.

PROGRAMMING MODEL

A programming model for the MC68120/MC68121 is shown in Figure 14. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most significant byte. Any operation which modifies the double accumulator will also modify accumulator A and/or B. Other registers are defined as follows:

Program Counter — The program counter is a 16-bit register which always points to the next instruction.

Stack Pointer — The Stack Pointer is a 16-bit register which contains the address of the next available location in a pushdown/pullup (LIFO) queue. The stack resides in random access memory at a location specified by the software.

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

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

Condition Code Register — The Condition Code Register indicates the results of an instruction and includes the following five condition bits: Negative (N), Zero (Z), Overflow (V), Carry/Borrow from MSB (C), and half carry from bit 3 (H). These bits are testable by the conditional branch instructions. Bit 4 is the interrupt mask (I-bit) and inhibits all maskable interrupts when set. The two unused bits b6 and b7, are read as ones.

ADDRESSING MODES

The MC68120/MC68121 provides six addressing modes which can be used to reference memory. A summary of addressing modes for all instructions is presented in Tables 13, 14, 15 and 16 where execution times are provided in E-cycles. Instruction execution times are summarized in Table 17. With an input frequency (E) of 1 MHz, E-cycles are equivalent to microseconds. A cycle-by-cycle description of bus activity for each instruction is provided in Table 18 and a description of selected instructions is shown in Figure 38.

Immediate Addressing — The operand is contained in the following byte(s) of the instruction where the number of bytes matches the size of the register. These are two or three byte instructions.

Direct Addressing — The least significant byte of the operand address is contained in the second byte of the instruction and the most significant byte is assumed to be $00. Direct addressing allows the user to access $00 through $FF using two byte instructions and execution time is reduced by eliminating the additional memory access (refer to Table 1). In most applications, this 256-byte area is reserved for frequently referenced data. Note that no direct addressing of internal control registers is possible in Mode 3.

Extended Addressing — The second and third bytes of the instruction contain the absolute address of the operand. These are three byte instructions.

Indexed Addressing — The unsigned offset contained in the second byte of the instructions is added with carry to the Index Register and used to reference memory without changing the Index Register. These are two byte instructions.

Inherent Addressing — The operand(s) are registers and no memory reference is required. These are single byte instructions.

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

TABLE 13 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

																		Condition Codes					
		Immed			Direct			Index			Extend			Inherent				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	4	3	9C	5	2	AC	6	2	BC	6	3				X - M : M + 1	●	●	↕	↕	↕	↕
Decrement Index Reg	DEX													09	3	1	X - 1 → X	●	●	●	↕	●	●
Decrement Stack Pntr	DES													34	3	1	SP - 1 → SP	●	●	●	●	●	●
Increment Index Reg	INX													08	3	1	X + 1 → X	●	●	●	↕	●	●
Increment Stack Pntr	INS													31	3	1	1 SP + 1 → SP	●	●	●	●	●	●
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	●	●	↕	↕	R	●
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M + 1) → SP_L	●	●	↕	↕	R	●
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	●	●	↕	↕	R	●
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	●	●	↕	↕	R	●
Index Reg → Stack Pntr	TXS													35	3	1	X - 1 → SP	●	●	●	●	●	●
Stack Pntr → Index Reg	TSX													30	3	1	SP + 1 → X	●	●	●	●	●	●
Add	ABX													3A	3	1	B + X → X	●	●	●	●	●	●
Push Data	PSHX													3C	4	1	X_L → M_{SP}, SP - 1 → SP X_H → M_{SP}, SP - 1 → SP	●	●	●	●	●	●
Pull Data	PULX													38	5	1	SP + 1 → SP, M_{SP} → X_H SP + 1 → SP, M_{SP} → X_L	●	●	●	●	●	●

TABLE 14 — ACCUMULATOR AND MEMORY INSTRUCTIONS (Sheet 1 of 2)

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	●	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B + X → X	●	●	●	●	●	●
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	●	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	●	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	●	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	●	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M:M + 1 → D	●	●	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A · M → A	●	●	↕	↕	R	●
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B · M → B	●	●	↕	↕	R	●
Shift Left, Arithmetic	ASL							68	6	2	78	6	3					●	●	↕	↕	↕	↕
	ASLA													48	2	1		●	●	↕	↕	↕	↕
	ASLB													58	2	1		●	●	↕	↕	↕	↕
Shift Left Dbl	ASLD													05	3	1		●	●	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					●	●	↕	↕	↕	↕
	ASRA													47	2	1		●	●	↕	↕	↕	↕
	ASRB													57	2	1		●	●	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A · M	●	●	↕	↕	R	●
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B · M	●	●	↕	↕	R	●
Compare Acmltrs	CBA													11	2	1	A - B	●	●	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	●	●	R	S	R	R
	CLRA													4F	2	1	00 → A	●	●	R	S	R	R
	CLRB													5F	2	1	00 → B	●	●	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	●	●	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	●	●	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				$\overline{M}$ → M	●	●	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	●	●	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	●	●	↕	↕	R	S
Decimal Adj, A	DAA													19	2	1	Adj binary sum to BCD	●	●	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	●	●	↕	↕	↕	●
	DECA													4A	2	1	A - 1 → A	●	●	↕	↕	↕	●
	DECB													5A	2	1	B - 1 → B	●	●	↕	↕	↕	●
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	●	●	↕	↕	R	●
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	●	●	↕	↕	R	●

TABLE 14 — ACCUMULATOR AND MEMORY INSTRUCTIONS (Sheet 2 of 2)

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	●	●	↕	↕	↕	●
	INCA													4C	2	1	A + 1 → A	●	●	↕	↕	↕	●
	INCB													5C	2	1	B + 1 → B	●	●	↕	↕	↕	●
Load Acmltrs	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	●	●	↕	↕	R	●
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	●	●	↕	↕	R	●
Load Double	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M:M + 1 → D	●	●	↕	↕	R	●
Logical Shift, Left	LSL							68	6	2	78	6	3					●	●	↕	↕	↕	↕
	LSLA													48	2	1		●	●	↕	↕	↕	↕
	LSLB													58	2	1		●	●	↕	↕	↕	↕
	LSLD													05	3	1		●	●	↕	↕	↕	↕
Shift Right, Logical	LSR							64	6	2	74	6	3					●	●	R	↕	↕	↕
	LSRA													44	2	1		●	●	R	↕	↕	↕
	LSRB													54	2	1		●	●	R	↕	↕	↕
	LSRD													04	3	1		●	●	R	↕	↕	↕
Multiply	MUL													3D	10	1	A X B → D	●	●	●	●	●	↕
2's Complement (Negate)	NEG							60	6	2	70	6	3				00 - M → M	●	●	↕	↕	↕	↕
	NEGA													40	2	1	00 - A → A	●	●	↕	↕	↕	↕
	NEGB													50	2	1	00 - B → B	●	●	↕	↕	↕	↕
No Operation	NOP													01	2	1	PC + 1 → PC	●	●	●	●	●	●
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	●	●	↕	↕	R	●
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	●	●	↕	↕	R	●
Push Data	PSHA													36	3	1	A → Stack	●	●	●	●	●	●
	PSHB													37	3	1	B → Stack	●	●	●	●	●	●
Pull Data	PULA													32	4	1	Stack → A	●	●	●	●	●	●
	PULB													33	4	1	Stack → B	●	●	●	●	●	●
Rotate Left	ROL							69	6	2	79	6	3					●	●	↕	↕	↕	↕
	ROLA													49	2	1		●	●	↕	↕	↕	↕
	ROLB													59	2	1		●	●	↕	↕	↕	↕
Rotate Right	ROR							66	6	2	76	6	3					●	●	↕	↕	↕	↕
	RORA													46	2	1		●	●	↕	↕	↕	↕
	RORB													56	2	1		●	●	↕	↕	↕	↕
Subtract Acmltr	SBA													10	2	1	A - B → A	●	●	↕	↕	↕	↕
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	●	●	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	●	●	↕	↕	↕	↕
Store Acmltrs	STAA				97	3	2	A7	4	2	B7	4	3				A → M	●	●	↕	↕	R	●
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	●	●	↕	↕	R	●
	STD				DD	4	2	ED	5	2	FD	5	3				D → M:M + 1	●	●	↕	↕	R	●
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	●	●	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	●	●	↕	↕	↕	↕
Subtract Double	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				D - M:M + 1 → D	●	●	↕	↕	↕	↕
Transfer Acmltr	TAB													16	2	1	A → B	●	●	↕	↕	R	●
	TBA													17	2	1	B → A	●	●	↕	↕	R	●
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M - 00	●	●	↕	↕	R	R
	TSTA													4D	2	1	A - 00	●	●	↕	↕	R	R
	TSTB													5D	2	1	B - 00	●	●	↕	↕	R	R

The Condition Code Register notes are listed after table 16.

TABLE 15 — JUMP AND BRANCH INSTRUCTIONS

		Direct			Relative			Index			Extnd			Inherent				Cond. Code Reg.					
																		5	4	3	2	1	0
Operations	**Mnemonic**	**OP**	**~**	**#**	**OP**	**~**	**#**	**OP**	**~**	**#**	**OP**	**~**	**#**	**OP**	**~**	**#**	**Branch Test**	**H**	**I**	**N**	**Z**	**V**	**C**
Branch Always	BRA				20	3	2										None	●	●	●	●	●	●
Branch Never	BRN				21	3	2										None	●	●	●	●	●	●
Branch If Carry Clear	BCC				24	3	2										C = 0	●	●	●	●	●	●
Branch If Carry Set	BCS				25	3	2										C = 1	●	●	●	●	●	●
Branch If = Zero	BEQ				27	3	2										Z = 1	●	●	●	●	●	●
Branch If ≥ Zero	BGE				2C	3	2										N⊕V = 0	●	●	●	●	●	●
Branch If > Zero	BGT				2E	3	2										Z + (N⊕V) = 0	●	●	●	●	●	●
Branch If Higher	BHI				22	3	2										C + Z = 0	●	●	●	●	●	●
Branch If Higher or Same	BHS				24	3	2										C = 0	●	●	●	●	●	●
Branch If ≤ Zero	BLE				2F	3	2										Z + (N⊕V) = 1	●	●	●	●	●	●
Branch If Carry Set	BLO				25	3	2										C = 1	●	●	●	●	●	●
Branch If Lower Or Same	BLS				23	3	2										C + Z = 1	●	●	●	●	●	●
Branch If < Zero	BLT				2D	3	2										N⊕V = 1	●	●	●	●	●	●
Branch If Minus	BMI				2B	3	2										N = 1	●	●	●	●	●	●
Branch If Not Equal Zero	BNE				26	3	2										Z = 0	●	●	●	●	●	●
Branch If Overflow Clear	BVC				28	3	2										V = 0	●	●	●	●	●	●
Branch If Overflow Set	BVS				29	3	2										V = 1	●	●	●	●	●	●
Branch If Plus	BPL				2A	3	2										N = 0	●	●	●	●	●	●
Branch To Subroutine	BSR				8D	6	2										See Special Operations — Figure 36	●	●	●	●	●	●
Jump	JMP							6E	3	2	7E	3	3					●	●	●	●	●	●
Jump To Subroutine	JSR	9D	5	2				AD	6	2	BD	6	3					●	●	●	●	●	●
No Operation	NOP													01	2	1		●	●	●	●	●	●
Return From Interrupt	RTI													3B	10	1	See Special Operations — Figure 36	↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1		●	●	●	●	●	●
Software Interrupt	SWI													3F	12	1		●	S	●	●	●	●
Wait For Interrupt	WAI													3E	9	1		●	●	●	●	●	●

TABLE 16 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

	Inherent					Cond. Code Reg.					
						5	4	3	2	1	0
Operations	**Mnemonic**	**OP**	**~**	**#**	**Boolean Operation**	**H**	**I**	**N**	**Z**	**V**	**C**
Clear Carry	CLC	0C	2	1	0 → C	●	●	●	●	●	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	●	R	●	●	●	●
Clear Overflow	CLV	0A	2	1	0 → V	●	●	●	●	R	●
Set Carry	SEC	0D	2	1	1 → C	●	●	●	●	●	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	●	S	●	●	●	●
Set Overflow	SEV	0B	2	1	1 → V	●	●	●	●	S	●
Accumulator A → CCR	TAP	06	2	1	A → CCR	↕	↕	↕	↕	↕	↕
CCR → Accumulator A	TPA	07	2	1	CCR → A	●	●	●	●	●	●

LEGEND

OP Operation Code (Hexadecimal)
~ Number of MPU Cycles
MSP Contents of memory location pointed to by Stack Pointer
# Number of Program Bytes
+ Arithmetic Plus
- Arithmetic Minus
● Boolean AND
X Arithmetic Multiply
+ Boolean Inclusive OR
⊕ Boolean Exclusive OR
$\overline{M}$ Complement of M
→ Transfer Into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

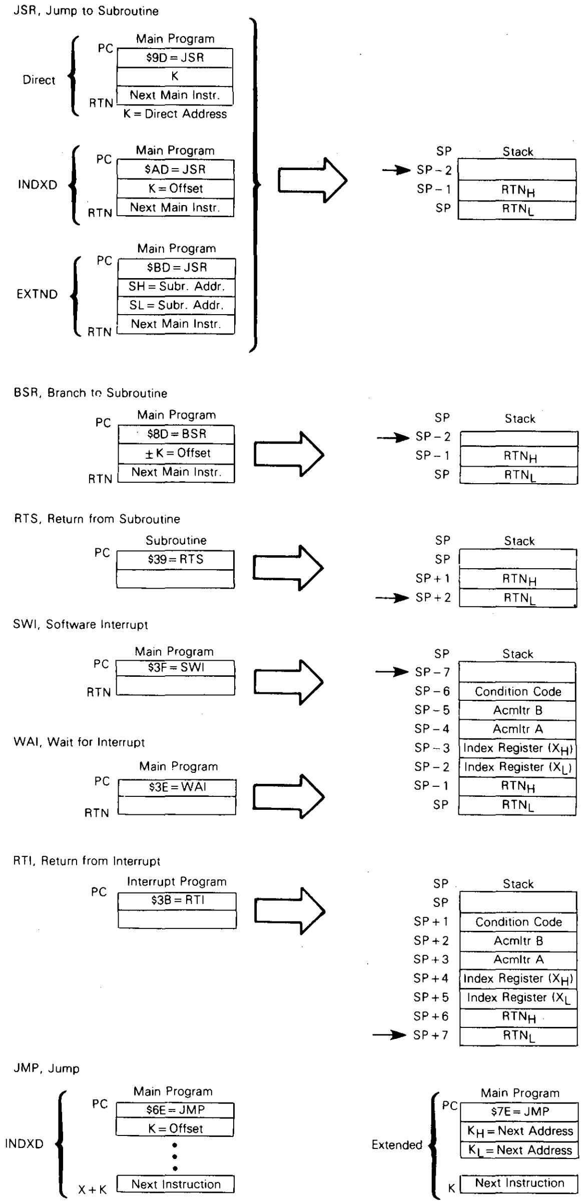
H Half-carry from bit 3
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry/Borrow from MSB
R Reset Always
S Set Always
↕ Affected
● Not Affected

TABLE 17 — INSTRUCTION EXECUTION TIMES IN E CYCLES

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

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

FIGURE 36 — SPECIAL OPERATIONS



Legend:

RTN = Address of next instruction in Main Program to be executed upon return from subroutine

RTN_H = Most significant byte of Return Address

RTN_L = Least significant byte of Return Address

➤ = Stack pointer after execution

K = 8-bit unsigned value

CYCLE-BY-CYCLE OPERATION SUMMARY

Table 18 provides a detailed description of the information present on the Address Bus, Data Bus, and the R/$\overline{W}$ line during cycle of each instructions.

The information is useful in comparing actual with expected results during debug of both software and hardware as the program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction. In general, instructions with the same addressing mode and number of cycles execute in the same manner. Exceptions are indicated in the table.

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

The coding of the first (or only) byte corresponding to an executable instruction is sufficient to identify the instruction and the addressing mode. The hexadecimal equivalents of the binary codes, which result from the translation of the 82 instructions in all valid modes of addressing, are shown in Table 19. There are 220 valid machine codes, 34 unassigned codes and 2 reserved for test purposes.

TABLE 18 — CYCLE BY CYCLE OPERATION (Sheet 1 of 5)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
IMMEDIATE					
ADC EOR	2	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Operand Data
AND ORA					
BIT SBC					
CMP SUB					
LDS	3	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Operand Data (High Order Byte)
LDD		3	Op Code Address + 2	1	Operand Data (Low Order Byte)
CPX	4	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Operand Data (High Order Byte)
ADDD		3	Op Code Address + 2	1	Operand Data (Low Order Byte)
		4	Address Bus FFFF	1	Low Byte of Restart Vector
DIRECT					
ADC EOR	3	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Address of Operand
AND ORA		3	Address of Operand	1	Operand Data
BIT SBC					
CMP SUB					
STA	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Destination Address
		3	Destination Address	0	Data from Accumulator
LDS	4	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Address of Operand
LDD		3	Address of Operand	1	Operand Data (High Order Byte)
		4	Operand Address + 1	1	Operand Data (Low Order Byte)
STS	4	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Address of Operand
STD		3	Address of Operand	0	Register Data (High Order Byte)
		4	Address of Operand + 1	0	Register Data (Low Order Byte)
CPX	5	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Address of Operand
ADDD		3	Operand Address	1	Operand Data (High Order Byte)
		4	Operand Address + 1	1	Operand Data (Low Order Byte)
		5	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	5	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Subroutine Address	1	First Subroutine Op Code
		4	Stack Pointer	0	Return Address (Low Order Byte)
		5	Stack Pointer + 1	0	Return Address (High Order Byte)

— Continued —

TABLE 18 — CYCLE BY CYCLE OPERATION (Sheet 2 of 5)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
EXTENDED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Jump Address (High Order Byte)
		3	Op Code Address + 2	1	Jump Address (Low Order Byte)
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Address of Operand
AND ORA		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
BIT SBC CMP SUB		4	Address of Operand	1	Operand Data
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Destination Address (High Order Byte)
		3	Op Code Address + 2	1	Destination Address (Low Order Byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS	5	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
LDD		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data (High Order Byte)
		5	Address of Operand + 1	1	Operand Data (Low Order Byte)
STS	5	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
STD		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	0	Operand Data (High Order Byte)
		5	Address of Operand + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Op Code Address	1	Op Code
ASR NEG		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
CLR ROL		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
COM ROR		4	Address of Operand	1	Current Operand Data
DEC TST		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Address of Operand	0	New Operand Data
CPX	6	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Operand Address (High Order Byte)
ADDD		3	Op code Address + 2	1	Operand Address (Low Order Byte)
		4	Operand Address	1	Operand Data (High Order Byte)
		5	Operand Address + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Address of Subroutine (High Order Byte)
		3	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
		4	Subroutine Starting Address	1	Op Code of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer - 1	0	Return Address (High Order Byte)

— Continued —

TABLE 18 — CYCLE BY CYCLE OPERATION (Sheet 3 of 5)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Offset
AND ORA		3	Address Bus FFFF	1	Low Byte of Restart Vector
BIT SBC		4	Index Register Plus Offset	1	Operand Data
CMP SUB					
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS	5	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Offset
LDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STS	5	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Offset
STD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Op Code Address	1	Op Code
ASR NEG		2	Op Code Address + 1	1	Offset
CLR ROL		3	Address Bus FFFF	1	Low Byte of Restart Vector
COM ROR		4	Index Register Plus Offset	1	Current Operand Data
DEC TST (1)		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Index Register Plus Offset	0	New Operand Data
CPX	6	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Offset
ADDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (High Order Byte)
		5	Index Register + Offset + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF		Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Op Code
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer - 1	0	Return Address (High Order Byte)

— Continued —

TABLE 18 — CYCLE BY CYCLE OPERATION (Sheet 4 of 5)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
ABA DAA SEC	2	1	Op Code Address	1	Op Code
ASL DEC SEI		2	Op Code Address +1	1	Op Code of Next Instruction
ASR INC SEV					
CBA LSR TAB					
CLC NEG TAP					
CLI NOP TBA					
CLR ROL TPA					
CLV ROR TST					
COM SBA					
ABX	3	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Irrelevent Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ASLD	3	1	Op Code Address	1	Op Code
LSRD		2	Op Code Address +1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
DES	3	1	Op Code Address	1	Op Code
INS		2	Op Code Address +1	1	Op Code of Next Instruction
		3	Previous Register Contents	1	Irrelevant Data
INX	3	1	Op Code Address	1	Op Code
DEX		2	Op Code Address +1	1	Op Code of Next Instruction
		3	Address Bus FFFF	1	Low Byte of Restart Vector
PSHA	3	1	Op Code Address	1	Op Code
PSHB		2	Op Code Address +1	1	Op Code of Next Instruction
		3	Stack Pointer	0	Accumulator Data
TSX	3	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Op Code of Next Instruction
		3	Stack Pointer	1	Irrelevant Data
TXS	3	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Op Code of Next Instruction
		3	Address Bus FFFF	1	Low Byte of Restart Vector
PULA	4	1	Op Code Address	1	Op Code
PULB		2	Op Code Address +1	1	Op Code of Next Instruction
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer +1	1	Operand Data from Stack
PSHX	4	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Irrelevant Data
		3	Stack Pointer	0	Index Register (Low Order Byte)
		4	Stack Pointer −1	0	Index Register (High Order Byte)
PULX	5	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer +1	1	Index Register (High Order Byte)
		5	Stack Pointer +2	1	Index Register (Low Order Byte)
RTS	5	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer +1	1	Address of Next Instruction (High Order Byte)
		5	Stack Pointer +2	1	Address of Next Instruction (Low Order Byte)
WAI	9	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Op Code of Next Instruction
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer −1	0	Return Address (High Order Byte)
		5	Stack Pointer −2	0	Index Register (Low Order Byte)
		6	Stack Pointer −3	0	Index Register (High Order Byte)
		7	Stack Pointer −4	0	Contents of Accumulator A
		8	Stack Pointer −5	0	Contents of Accumulator B
		9	Stack Pointer −6	0	Contents of Cond. Code Register

— Continued —

TABLE 18 — CYCLE BY CYCLE OPERATION (Sheet 5 of 5)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
MUL	10	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer +1	1	Contents of Cond. Code Reg. from Stack
		5	Stack Pointer +2	1	Contents of Accumulator B from Stack
		6	Stack Pointer +3	1	Contents of Accumulator A from Stack
		7	Stack Pointer +4	1	Index Register from Stack (High Order Byte)
		8	Stack Pointer +5	1	Index Register from Stack (Low Order Byte)
		9	Stack Pointer +6	1	Next Instruction Address from Stack (High Order Byte)
		10	Stack Pointer +7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer −1	0	Return Address (High Order Byte)
		5	Stack Pointer −2	0	Index Register (Low Order Byte)
		6	Stack Pointer −3	0	Index Register (High Order Byte)
		7	Stack Pointer −4	0	Contents of Accumulator A
		8	Stack Pointer −5	0	Contents of Accumulator B
		9	Stack Pointer −6	0	Contents of Cond. Code Register
		10	Stack Pointer −7	1	Irrelevant Data
		11	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)
RELATIVE					
BCC BHT BNE BLO BCS BLE BPL BHS BEQ BLS BRA BRN BGE BLT BVC BGT BMT BVS	3	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
BSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Subroutine Starting Address	1	Op Code of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer −1	0	Return Address (High Order Byte)

TABLE 19 – CPU INSTRUCTION MAP

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

*UNDEFINED OP CODE

NOTES:

1. Addressing Modes

   INHER ≡ Inherent INDXD ≡ Indexed IMMED ≡ Immediate
   REL ≡ Relative EXTND ≡ Extended DIR ≡ Direct

2. Unassigned opcodes are indicated by "*" and should not be executed.
3. Codes marked by "T" force the PC to function as a 16-bit counter.

APPENDIX A
MC68120 CUSTOM ORDERING INFORMATION

A.0

Address $FFEF is Reserved for the Checksum value for the ROM, to be generated at the factory.

A.1 CUSTOM MC68120 ORDERING INFORMATION

The custom MC68120 specifications may be transmitted to Motorola in any of the following media:

A) EPROM(s)

B) MDOS diskette

The specification should be formatted and packaged, as indicated in the appropriate paragraph below, and mailed prepaid and insured with a cover letter (see Figure A-1) to:

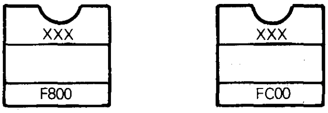
Motorola Inc.
MPU Marketing
3501 Ed Bluestein Blvd.
Austin, Texas 78721

A copy of the cover letter should also be mailed separately.

A.2 EPROMs

MCM2708 and MCM2716 type EPROMs, programmed with the custom program (positive logic notation for address and data), may be submitted for pattern generation. The MC2708s must be clearly marked to indicate which PROM corresponds to which address space ($F800-$FBFF; $FC00-$FFFF). See Figure A-2 for recommended marking procedure.

FIGURE A-2



XXX = Customer ID

After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

A.3 MDOS DISKETTE

The file name and start/end location should be written on the label.

FIGURE A-1

CUSTOMER NAME ____________________

ADDRESS ____________________

STATE ____________________ CITY ____________________ ZIP __________

PHONE ____________________ EXTENSION ____________________

CONTACT MS/MR ____________________

CUSTOMER PART # ____________________

PATTERN MEDIA
□ 2708 EPROM
□ 2716 EPROM
□ Diskette (MDOS)

TEMPERATURE RANGE
□ 0° to 70°C

PACKAGE TYPE
□ Ceramic

MARKING
□ Standard
□ Special

(Note 1) ____________________

NOTE: (1) Other Media Require Prior Factory Approval

SIGNATURE ____________________

TITLE ____________________