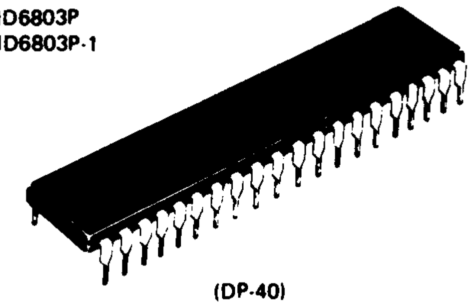
HD6803, HD6803-1

MPU (Micro Processing Unit)

The HD6803 MPU is an 8-bit micro processing unit which is compatible with the HMCS6800 family of parts. The HD6803 MPU is object code compatible with the HD6800 with improved execution times of key instructions plus several new 16-bit and 8-bit instruction including an 8 × 8 unsigned multiply with 16-bit result. The HD6803 MPU can be expanded to 65k bytes. The HD6803 MPU is TTL compatible and requires one +0.5 volt power supply. The HD6803 MPU has 128 bytes of RAM, Serial Communications Interface (S.C.I.), and parallel I/O as well as a three function 16-bit timer. Features and Block Diagram of the HD6803 include the following:

HD6803P
HD6803P-1

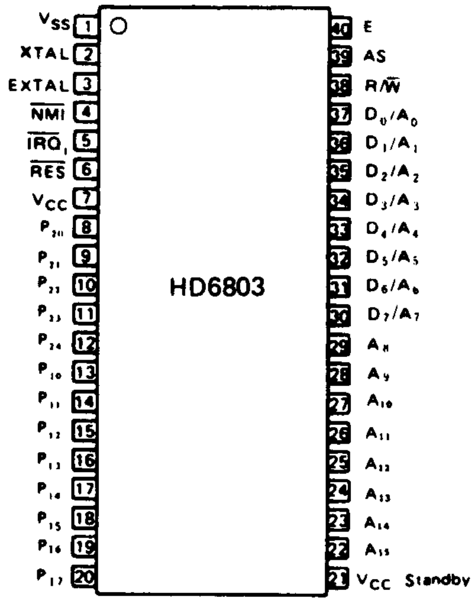


(DP-40)

■ FEATURES

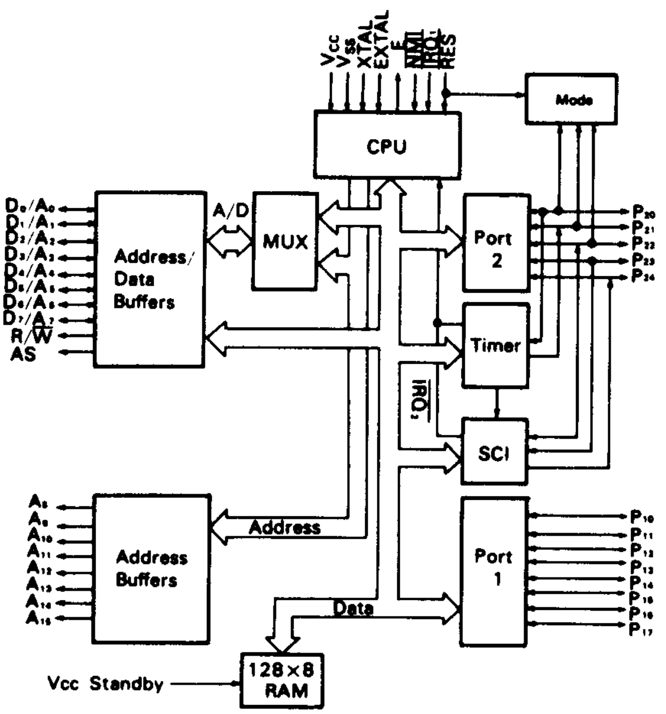
- Expanded HMCS6800 Instruction Set
- 8 x 8 Multiply
- On-Chip Serial Communications Interface (S.C.I.)
- Object Code Compatible with The HD6800 MPU
- 16-Bit Timer
- Expandable to 65k Bytes
- Multiplexed Address and Data
- 128 Bytes of RAM (64 Bytes Retainable On Power Down)
- 13 Parallel I/O Lines
- Internal Clock/Divided-By-Four
- TTL Compatible Inputs and Outputs
- Interrupt Capability
- Compatible with MC6803 and MC6803-1

■ PIN ARRANGEMENT



(Top View)

■ BLOCK DIAGRAM



■ TYPE OF PRODUCTS

Type No.	Bus Timing
HD6803	1.0MHz
HD6803-1	1.25MHz

■ **ABSOLUTE MAXIMUM RATINGS**

Item	Symbol	Value	Unit
Supply Voltage	V_{CC} *	-0.3 ~ +7.0	V
Input Voltage	V_{in} *	-0.3 ~ +7.0	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	- 55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

[NOTE] Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ **ELECTRICAL CHARACTERISTICS**

● **DC CHARACTERISTICS (V_{CC} =5.0V±5%, V_{SS} = 0V, Ta = 0~+70°C, unless otherwise noted.)**

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$	V_{IH}		4.0	–	V_{CC}	V
	Other Inputs*			2.0	–	V_{CC}	
Input "Low" Voltage	All Inputs*	V_{IL}		-0.3	–	0.8	V
Input Load Current	EXTAL	$\lvert I_{in} \rvert$	V_{in} = 0 ~ V_{CC}	–	–	0.8	mA
Input Leakage Current	$\overline{NMI}$, $\overline{IRQ_1}$, $\overline{RES}$	$\lvert I_{in} \rvert$	V_{in} = 0 ~ 5.25V	–	–	2.5	µA
Three State (Offset) Leakage Current	P_{10} ~ P_{17}, D_0/A_0 ~ D_7/A_7	$\lvert I_{TSI} \rvert$	V_{in} = 0.5 ~ 2.4V	–	–	10	µA
	P_{20} ~ P_{24}			–	–	100	
Output "High" Voltage	D_0/A_0 ~ D_7/A_7	V_{OH}	I_{LOAD} = -205 µA	2.4	–	–	V
	A_8 ~ A_{15}, E, R/$\overline{W}$, AS		I_{LOAD} = -145 µA	2.4	–	–	
	Other Outputs		I_{LOAD} = -100 µA	2.4	–	–	
Output "Low" Voltage	All Outputs	V_{OL}	I_{LOAD} = 1.6 mA	–	–	0.5	V
Darlington Drive Current	P_{10} ~ P_{17}	$-I_{OH}$	V_{out} = 1.5V	1.0	–	10.0	mA
Power Dissipation		P_D		–	–	1200	mW
Input Capacitance	D_0/A_0 ~ D_7/A_7	C_{in}	V_{in} = 0V, Ta = 25°C, f = 1.0 MHz	–	–	12.5	pF
	Other Inputs			–	–	10.0	
V_{CC} Standby	Powerdown	V_{SBB}		4.0	–	5.25	V
	Operating	V_{SB}		4.75	–	5.25	
Standby Current	Powerdown	I_{SBB}	V_{SBB} = 4.0V	–	–	8.0	mA

*Except Mode Programming Levels.

2

- **AC CHARACTERISTICS**

BUS TIMING (V_{CC} = 5.0V ± 5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	HD6803			HD6803-1			Unit
				min	typ	max	min	typ	max	
Cycle Time		t_{cyc}	Fig. 1	1	–	10	0.8	–	10	μs
Address Strobe Pulse Width "High" *		PW_{ASH}		200	–	–	150	–	–	ns
Address Strobe Rise Time		t_{ASr}		5	–	50	5	–	50	ns
Address Strobe Fall Time		t_{ASf}		5	–	50	5	–	50	ns
Address Strobe Delay Time *		t_{ASD}		60	–	–	30	–	–	ns
Enable Rise Time		t_{Er}		5	–	50	5	–	50	ns
Enable Fall Time		t_{Ef}		5	–	50	5	–	50	ns
Enable Pulse Width "High" Time *		PW_{EH}		450	–	–	340	–	–	ns
Enable Pulse Width "Low" Time *		PW_{EL}		450	–	–	350	–	–	ns
Address Strobe to Enable Delay Time *		t_{ASED}		60	–	–	30	–	–	ns
Address Delay Time		t_{AD}		–	–	260	–	–	260	ns
Address Delay Time for Latch *		t_{ADL}		–	–	270	–	–	260	ns
Data Set-up Write Time		t_{DSW}		225	–	–	115	–	–	ns
Data Set-up Read Time		t_{DSR}		80	–	–	70	–	–	ns
Data Hold Time	Read	t_{HR}		10	–	–	10	–	–	ns
	Write	t_{HW}		20	–	–	20	–	–	
Address Set-up Time for Latch *		t_{ASL}		60	–	–	50	–	–	ns
Address Hold Time for Latch		t_{AHL}		20	–	–	20	–	–	ns
Address Hold Time		t_{AH}		20	–	–	20	–	–	ns
Peripheral Read Access Time (Multiplexed Bus)*		(t_{ACCM})		–	–	(600)	–	–	(420)	ns
Oscillator stabilization Time		t_{RC}	Fig. 8	100	–	–	100	–	–	ms
Processor Control Set-up Time		t_{PCS}	Fig. 7, 8	200	–	–	200	–	–	ns

*These timings change in approximate proportion to t_{cyc}. The figures in this characteristics represent those when t_{cyc} is minimum (= in the highest speed operation).

PERIPHERAL PORT TIMING (V_{CC} = 5.0V ± 5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Peripheral Data Setup Time	Port 1, 2	t_{PDSU}	Fig. 2	200	–	–	ns
Peripheral Data Hold Time	Port 1, 2	t_{PDH}	Fig. 2	200	–	–	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	Port 1, 2*	t_{PWD}	Fig. 3	–	–	400	ns

* Except P_{21}

TIMER, SCI TIMING (V_{CC} = 5.0V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item	Symbol	Test Condition	min	typ	max	Unit
Timer Input Pulse Width	t_{PWT}		$2t_{cyc}+200$	–	–	ns
Delay Time, Enable Positive Transition to Timer Out	t_{TOD}	Fig. 4	–	–	600	ns
SCI Input Clock Cycle	t_{Scyc}		1	–	–	t_{cyc}
SCI Input Clock Pulse Width	t_{PWSCK}		0.4	–	0.6	t_{Scyc}

MODE PROGRAMMING (V_{CC} = 5.0V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Mode Programming Input "Low" Voltage		V_{MPL}	Fig. 5	–	–	1.7	V
Mode Programming Input "High" Voltage		V_{MPH}		4.0	–	–	V
$\overline{RES}$ "Low" Pulse Width		PW_{RSTL}		3.0	–	-	t_{cyc}
Mode Programming Set-up Time		t_{MPS}		2.0	–	–	t_{cyc}
Mode Programming Hold Time	$\overline{RES}$ Rise Time ≧ 1μs	t_{MPH}		0	–	–	ns
	$\overline{RES}$ Rise Time < 1μs			100	–	–	

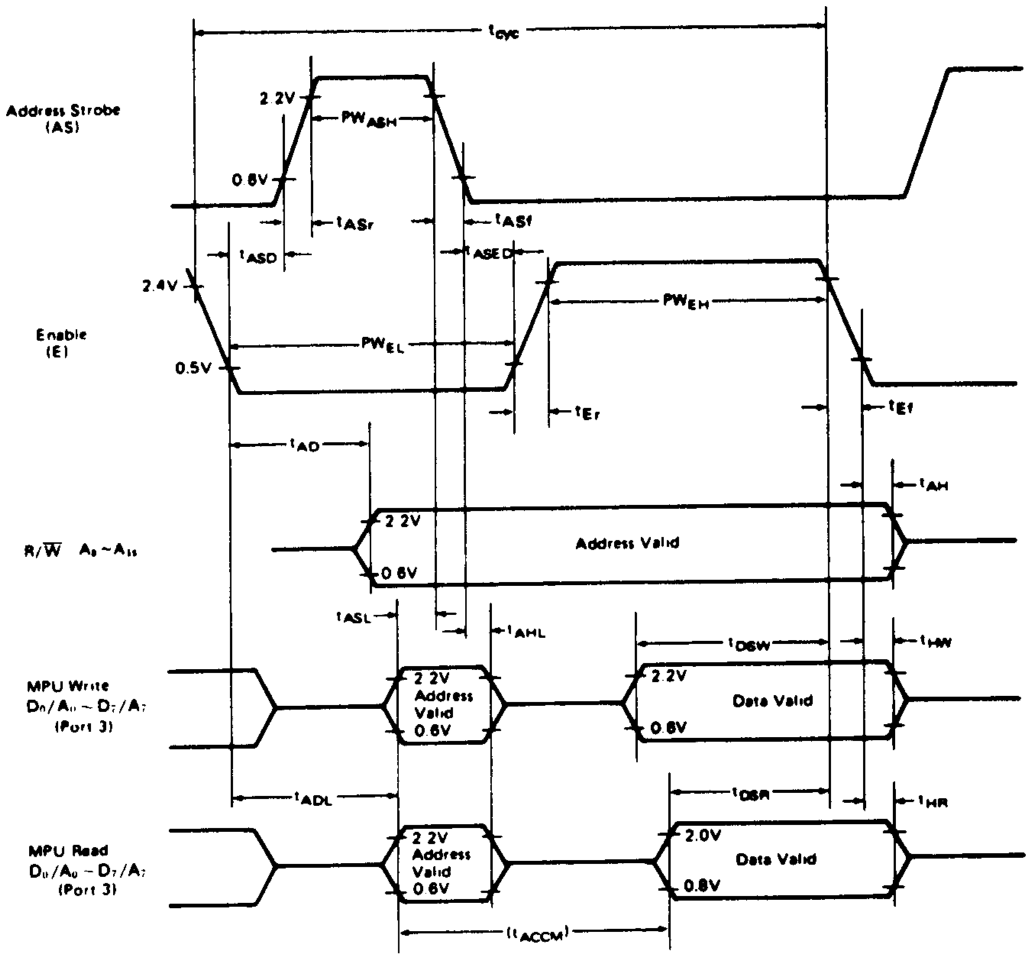


Figure 1 Expanded Multiplexed Bus Timing

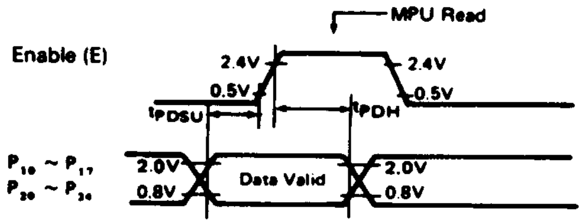


Figure 2 Data Set-up and Hold Times (MPU Read)

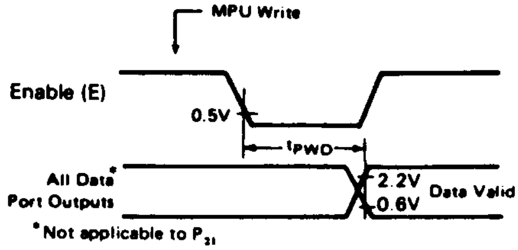


Figure 3 Port Data Delay Timing (MPU Write)

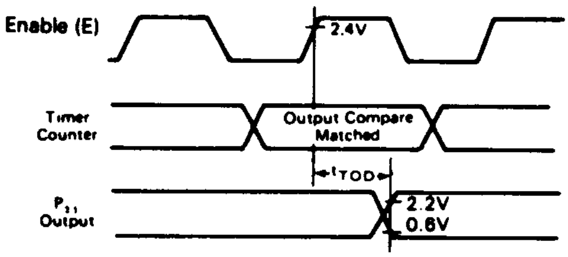


Figure 4 Timer Output Timing

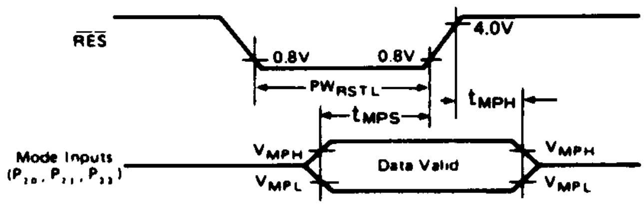
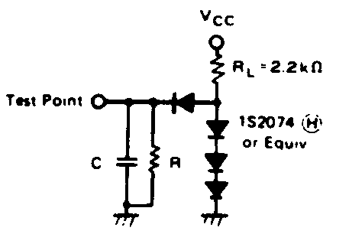


Figure 5 Mode Programming Timing



C = 90 pF for $D_0/A_0 \sim D_7/A_7$, $A_8 \sim A_{15}$, E, AS, R/$\overline{W}$

= 30 pF for $P_{10} \sim P_{17}$, $P_{20} \sim P_{24}$

R = 12 kΩ for $D_0/A_0 \sim D_7/A_7$, $A_8 \sim A_{15}$, E, AS, R/$\overline{W}$

= 24 kΩ for $P_{10} \sim P_{17}$, $P_{20} \sim P_{24}$

TTL Load

Figure 6 Bus Timing Test Load

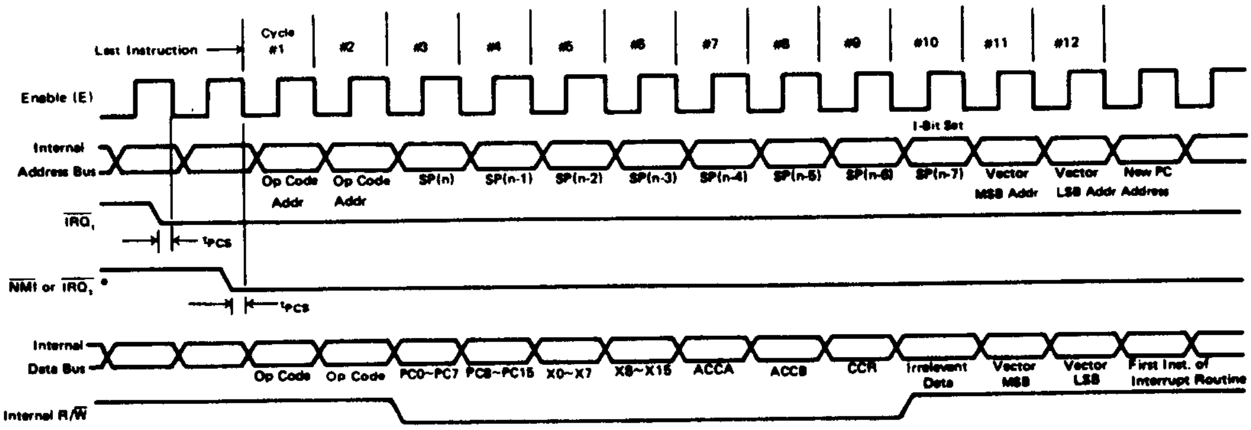


Figure 7 Interrupt Sequence

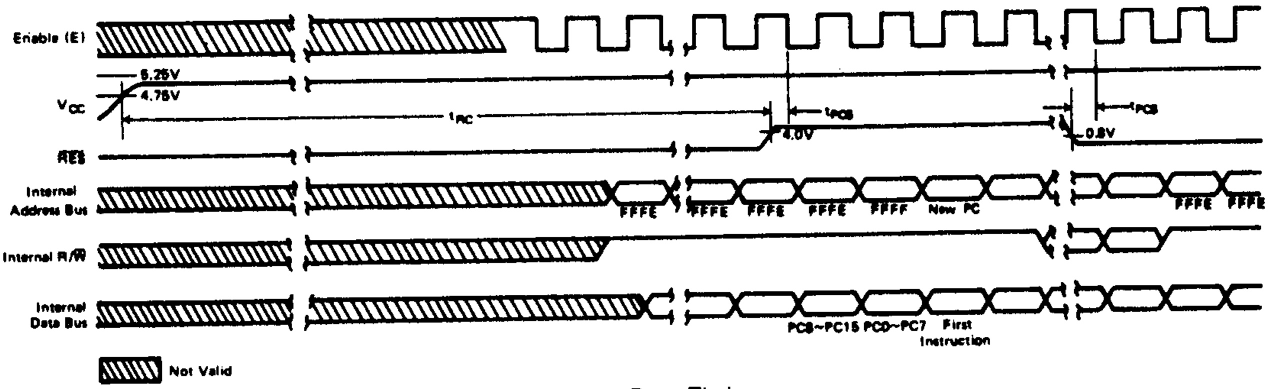


Figure 8 Reset Timing

■ SIGNAL DESCRIPTIONS

● Vcc and Vss

These two pins are used to supply power and ground to the chip. The voltage supplied will be +5 volts ±5%.

● XTAL and EXTAL

These connections are for a parallel resonant fundamental crystal, AT cut. Devide-by-4 circuitry is included with the internal clock, so a 4 MHz crystal may be used to run the system at 1 MHz. The devide-by-4 circuitry allows for use of the inexpensive 3.58 MHz Color TV crystal for non-time critical applications. Two 22pF capacitors are needed from the two crystal pins to ground to insure reliable operation. An example of the crystal interface is shown in Fig. 9. EXTAL may be driven by an external TTL compatible source with a 45% to 55% duty cycle. It will devided by 4 any frequency less than or equal to 5 MHz. XTAL must be grounded if an external clock is used.

Nominal Crystal Parameter

Crystal / Item	4 MHz	5 MHz
C_O	7pF max.	4.7pF max.
R_S	60Ω max.	30Ω typ.

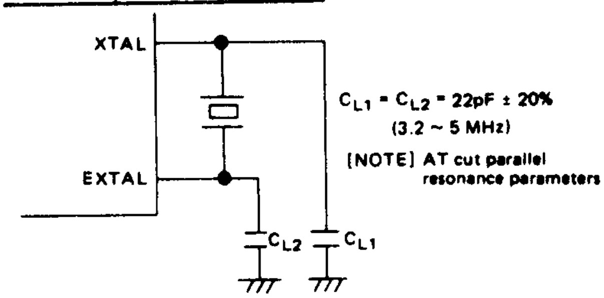


Figure 9 Crystal Interface

2

- **V_{CC} Standby**

This pin will supply +5 volts ±5% to the standby RAM on the chip. The first 64 bytes of RAM will be maintained in the power down mode with 8 mA current max. The circuit of figure 13 can be utilized to assure that V_{CC} Standby does not go below V_{SBB} during power down.

To retain information in the RAM during power down the following procedure is necessary:

1) Write "0" into the RAM enable bit, RAME. RAME is bit 6 of the RAM Control Register at location $0014. This disables the standby RAM, thereby protecting it at power down.
2) Keep V_{CC} Standby greater than V_{SBB}.

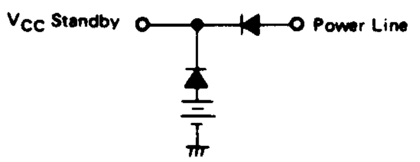


Figure 10 Battery Backup for V_{CC} Standby

- **Reset ($\overline{RES}$)**

This input is used to reset and start the MPU from a power down condition, resulting from a power failure or an initial startup of the processor. On power up, the reset must be held "Low" for at least 100 ms. When reset during operation, $\overline{RES}$ must be held "Low" at least 3 clock cycles.

When a "High" level is detected, the CPU does the following;

1) All the higher order address lines will be forced "High".
2) I/O Port 2 bits, 2, 1, and 0 are latched into programmed control bits PC2, PC1 and PC0.
3) The last two ($FFFE, $FFFF) locations in memory will be used to load the program addressed by the program counter.
4) The interrupt mask bit is set. Clear before the CPU can recognize maskable interrupts.

- **Enable (E)**

This supplies the external clock for the rest of the system when the internal oscillator is used. It is a single phase, TTL compatible clock, and will be the divide-by-4 result of the crystal oscillator frequency. It will drive one TTL load and 90 pF capacitance.

- **Non-Maskable Interrupt ($\overline{NMI}$)**

When the falling edge of the input signal is detected at this pin, the CPU begins non-maskable interrupt sequence internally. As with interrupt Request signal, the processor will complete the current instruction that is being executed before it recognizes the $\overline{NMI}$ signal. The interrupt mask bit in the Condition Code Register has no effect on $\overline{NMI}$.

In response to an $\overline{NMI}$ interrupt, the Index Register, Program Counter, Accumulators, and Condition Code Register are stored on the stack. At the end of the sequence, a 16-bit address will be loaded that points to a vectoring address located in memory locations $FFFC and $FFFD. An address loaded at these locations causes the CPU to branch to a non-maskable interrupt service routine in memory.

A 3.3 kΩ external resistor to V_{CC} should be used for wire-OR and optimum control of interrupts.

Inputs $\overline{IRQ_1}$ and $\overline{NMI}$ are hardware interrupt lines that are sampled during E and will start the interrupt routine on the $\bar{E}$ following the completion of an instruction.

- **Interrupt Request ($\overline{IRQ_1}$)**

This level sensitive input requests that an interrupt sequence be generated within the machine. The processor will complete the current instruction before it recognizes the request. At that time, if the interrupt mask bit in the Condition Code Register is not set, the machine will begin an interrupt sequence. The Index Register, Program Counter, Accumulators, and Condition Code Register are stored on the stack. Next the CPU will respond to the interrupt request by setting the interrupt mask bit "High" so that no further maskable interrupts may occur. At the end of the cycle, a 16-bit address will be loaded that points to a vectoring address which is located in memory locations $FFF8 and $FFF9. An address loaded at these locations causes the CPU to branch to an interrupt routine in memory.

The $\overline{IRQ_1}$ requires a 3.3 kΩ external resistor to V_{CC} which should be used for wire-OR and optimum control of interrupts. Internal Interrupts will use an internal interrupt line ($\overline{IRQ_2}$). This interrupt will operate the same as $\overline{IRQ_1}$ except that it will use the vector address of $FFF0 through $FFF7. $\overline{IRQ_1}$ will have priority to $\overline{IRQ_2}$ if both occur at the same time. The Interrupt Mask Bit in the condition code register masks both interrupts (See Table 1).

Table 1 Interrupt Vector Location

	Vector		Interrupt
	MSB	LSB	
Highest Priority	FFFE	FFFF	$\overline{RES}$
	FFFC	FFFD	$\overline{NMI}$
	FFFA	FFFB	Software Interrupt (SWI)
	FFF8	FFF9	$\overline{IRQ_1}$
	FFF6	FFF7	ICF (Input Capture)
	FFF4	FFF5	OCF (Output Compare)
	FFF2	FFF3	TOF (Timer Overflow)
Lowest Priority	FFF0	FFF1	SCI (RDRF + ORFE + TDRE)

- **Read/Write (R/$\overline{W}$)**

This TTL compatible output signals the peripherals and memory devices whether the CPU is in a Read ("High") or a Write ("Low") state. The normal standby state of this signal is Read ("High"). This output can drive one TTL load and 90pF capacitance.

- **Address Strobe (AS)**

In the expanded multiplexed mode of operation, address strobe is output on this pin. This signal is used to latch the 8 LSB's of address which are multiplexed with data on D_0/A_0 to D_7/A_7. An 8-bit latch is utilized in conjunction with Address Strobe, as shown in figure 11. So D_0/A_0 to D_7/A_7 can become data bus during the E pulse. The timing for this signal is shown in Figure 1 of Bus Timing. This signal is also used to disable the address from the multiplexed bus allowing a deselect time, t_{ASD} before the data is enabled to the bus.

■ PORTS

There are two I/O ports on the HD6803 MPU; one 8-bit port and one 5-bit port. Each port has an associated write only Data Direction Register which allows each I/O line to be programmed to act as an input or an output*. A "1" in the corresponding Data Direction Register bit will cause that I/O line to be an output. A "0" in the corresponding Data Direction Register bit will cause that I/O line to be an input. There are two ports: Port 1, Port 2. Their addresses and the addresses of their Data Direction registers are given in Table 2.

* The only exception is bit 1 of Port 2, which can either be data input or Timer output.

Table 2 Port and Data Direction Register Addresses

Ports	Port Address	Data Direction Register Address
I/O Port 1	$0002	$0000
I/O Port 2	$0003	$0001

● **I/O Port 1**

This is an 8-bit port whose individual bits may be defined as inputs or outputs by the corresponding bit in its data direction register. The 8 output buffers have three-state capability, allowing them to enter a high impedance state when the peripheral data lines are used as inputs. In order to be read properly, the voltage on the input lines must be greater than 2.0 V for a logic "1" and less than 0.8 V for a logic "0". As outputs, these lines are TTL compatible and may also be used as a source of up to 1 mA at 1.5 V to directly drive a Darlington base. After reset, the I/O lines are configured as inputs.

● **I/O Port 2**

This port has five lines that may be defined as inputs or outputs by its data direction register. The 5 output buffers have three-state capability, allowing them to enter a high impedance state when used as an input. In order to be read properly, the voltage on the input lines must be greater than 2.0 V for a logic "1" and less than 0.8 V for a logic "0". As outputs, this port has no internal pullup resistors but will drive TTL inputs directly. For driving CMOS inputs, external pullup resistors are required. After reset, the I/O lines are configured as inputs. Three pins on Port 2 (pin 8, 9 and 10 of the chip) are requested to set following values (Table 3) during reset. The values of above three pins during reset are latched into the three MSBs (Bit 5, 6 and 7) of Port 2 which are read only.

Port 2 can be configured as I/O and provides access to the Serial Communications Interface and the Timer. Bit 1 is the only pin restricted to data input or Timer output.

Table 3 The Values of three pins

Pin Number	Value
8	L
9	H
10	L

(NOTES) L; Logical "0"
H; Logical "1"

■ **BUS**

● **Data/Address Lines ($D_0/A_0 \sim D_7/A_7$)**

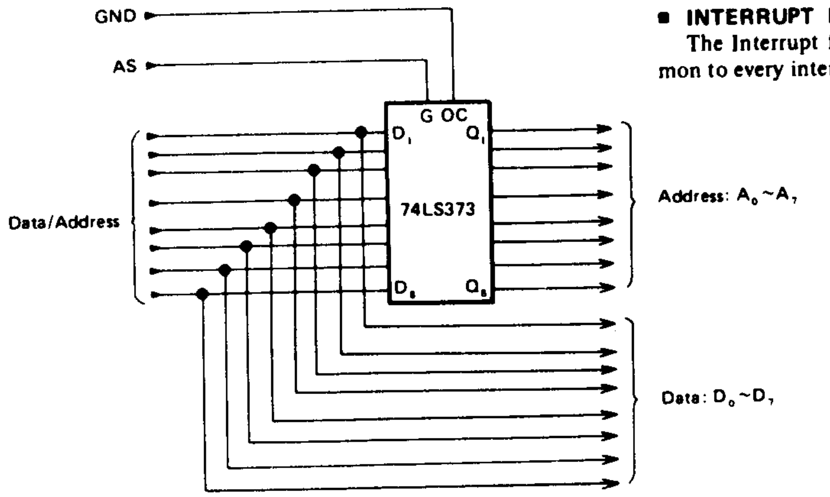
Since the data bus is multiplexed with the lower order address bus in Data/Address, latches are required to latch those address bits. The 74LS373 Transparent Octal D-type latch can be used with the HD6803 to latch the least significant address byte. Figure 11 shows how to connect the latch to the HD6803. The output control to the 74LS373 may be connected to ground.

● **Address Lines ($A_8 \sim A_{15}$)**

Each line is TTL compatible and can drive one TTL load and 90 pF. After reset, these pins become output for upper order address lines (A_8 to A_{15}).

■ **INTERRUPT FLOWCHART**

The Interrupt flowchart is depicted in Figure 16 and is common to every interrupt excluding reset.



Function Table

Output Control	Enable G	Enable D	Output Q
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

Figure 11 Latch Connection

■ MEMORY MAP

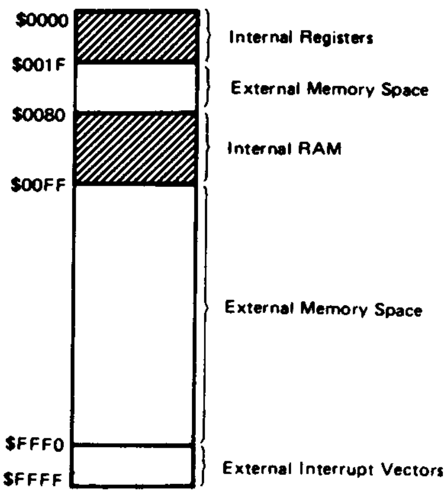
The MPU can provide up to 65k byte address space. A memory map is shown in Figure 12. The first 32 locations are reserved for the MPU's internal register area, as shown in Table 4 with exceptions as indicated.

Table 4 Internal Register Area

Register	Address
Port 1 Data Direction Register**	00
Port 2 Data Direction Register**	01
Port 1 Data Register	02
Port 2 Data Register	03
Not Used	04*
Not Used	05*
Not Used	06*
Not Used	07*
Timer Control and Status Register	08
Counter (High Byte)	09
Counter (Low Byte)	0A
Output Compare Register (High Byte)	0B
Output Compare Register (Low Byte)	0C
Input Capture Register (High Byte)	0D
Input Capture Register (Low Byte)	0E
Not Used	0F*
Rate and Mode Control Register	10
Transmit/Receive Control and Status Register	11
Receive Data Register	12
Transmit Data Register	13
RAM Control Register	14
Reserved	15-1F

* External Address
** 1; Output, 0; Input

Multiplexed/RAM



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

Figure 12 HD6803 Memory Map

■ PROGRAMMABLE TIMER

The HD6803 contains an on-chip 16-bit programmable timer which may be used to measure an input waveform while independently generating an output waveform. Pulse widths for both input and output signals may vary from a few microseconds to many seconds. The timer hardware consists of

- an 8-bit control and status register,
- a 16-bit free running counter,
- a 16-bit output compare register,
- a 16-bit input capture register

A block diagram of the timer registers is shown in Figure 13.

● Free Running Counter ($0009:$000A)

The key element in the programmable timer is a 16-bit free running counter which is driven to increasing values by E (Enable). The counter value may be read by the CPU software at any time. The counter is cleared to zero by reset and may be considered a read-only register with one exception. Any CPU write to the counter's address ($09) will always result in preset value of $FFF8 being loaded into the counter regardless of the value involved in the write. This preset figure is intended for testing operation of the part, but may be of value in some applications.

● Output Compare Register ($000B:$000C)

The Output Compare Register is a 16-bit read/write register which is used to control an output waveform. The contents of this register are constantly compared with the current value of the free running counter. When a match is found, a flag is set (OCF) in the Timer Control and Status Register (TCSR) and the current value of the Output Level bit (OLVL) in the TCSR is clocked to the Output Level Register. Providing the Data Direction Register for Port 2, Bit 1 contains a "1" (Output), the output level register value will appear on the pin for Port 2 Bit 1. The values in the Output Compare Register and Output Level bit may then be changed to control the output level on the next compare value. The Output Compare Register is set to $FFFF during reset. The Compare function is inhibited for one cycle following a write to the high byte of the Output Compare Register to insure a valid 16-bit value is in the register before a compare is made.

● Input Capture Register ($000D:$000E)

The Input Capture Register is a 16-bit read-only register used to store the current value of the free running counter when the proper transition of an external input signal occurs. The input transition change required to trigger the counter transfer is controlled by the input Edge bit (IEDG) in the TCSR. The Data Direction Register bit for Port 2 Bit 0, should* be clear (zero) in order to gate in the external input signal to the edge detect unit in the timer.

The input pulse width must be at least two E-cycles to ensure an input capture under all conditions.

* With Port 2 Bit 0 configured as an output and set to "1", the external input will still be seen by the edge detect unit.

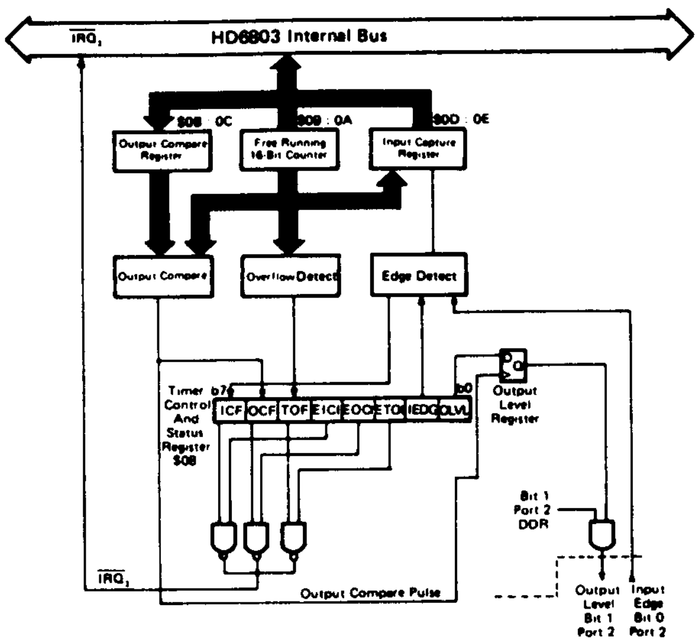


Figure 13 Block Diagram of Programmable Timer

Timer Control and Status Register

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$0008

- **Timer Control and Status Register (TCSR) ($0008)**

The Timer Control and Status Register consists of an 8-bit register of which all 8 bits are readable but only the low order 5 bits may be written. The upper three bits contain read-only timer status information and indicate the followings:

- a proper transition has taken place on the input pin with a subsequent transfer of the current counter value to the input capture register.
- a match has been found between the value in the free running counter and the output compare register, and when $0000 is in the free running counter.

Each of the flags may be enabled onto the HD6803 internal bus ($\overline{IRQ}_2$) with an individual Enable bit in the TCSR. If the I-bit in the HD6803 Condition Code register has been cleared, a prior vectored interrupt will occur corresponding to the flag bit(s) set. A description for each bit follows:

Bit 0 OLVL Output Level – This value is clocked to the output level register on a successful output compare. If the DDR for Port 2 bit 1 is set, the value will appear on the output pin.

Bit 1 IEDG Input Edge – This bit controls which transition of an input will trigger a transfer of the counter to the input capture register. The DDR for Port 2 Bit 0 must be clear for this function to operate. IEDG = 0 Transfer takes place on a negative edge ("High"-to-"Low" transition).
IEDG = 1 Transfer takes place on a positive edge ("Low"-to-"High" transition).

Bit 2 ETOI Enable Timer Overflow Interrupt – When set, this bit enables $\overline{IRQ}_2$ to occur on the internal bus for a TOF interrupt; when clear the interrupt is inhibited.

Bit 3 EOCI Enable Output Compare Interrupt – When set, this bit enables $\overline{IRQ}_2$ to appear on the internal bus for an output compare interrupt; when clear the interrupt is inhibited.

Bit 4 EICI Enable input Capture Interrupt – When set, this bit enables $\overline{IRQ}_2$ to occur on the internal bus for an input capture interrupt; when clear the interrupt is inhibited.

Bit 5 TOF Timer Overflow Flag – This read-only bit is set when the counter contains $FFFF. It is cleared by a read of the TCSR (with TOF set) followed by an CPU read of the Counter ($09).

Bit 6 OCF Output Compare Flag – This read-only bit is set when a match is found between the output compare register and the free running counter. It is cleared by a read of the TCSR (with OCF set) followed by an CPU write to the output compare register ($0B or $0C).

Bit 7 ICF Input Capture Flag – This read-only status bit is set by a proper transition on the input; it is cleared by a read of the TCSR (with ICF set) followed by an CPU read of the Input Capture Register ($0D).

- **SERIAL COMMUNICATIONS INTERFACE**

The HD6803 contains a full-duplex asynchronous serial communications interface (SCI) on chip. The controller comprises a transmitter and a receiver which operate independently or each other but in the same data format and at the same data rate. Both transmitter and receiver communicate with the CPU via the data bus and with the outside world via pins 2, 3, and 4 of Port 2. The hardware, software, and registers are explained in the following paragraphs.

- **Wake-Up Feature**

In a typical multi-processor application, the software protocol will usually contain a destination address in the initial byte(s) of the message. In order to permit non-selected MPU's to ignore the remainder of the message, a wake-up feature is included whereby all further interrupt processing may be optionally inhibited until the beginning of the next message. When the next message appears, the hardware re-enables (or "wakes-up") for the next message. The "wake-up" is automatically triggered by a string of ten consecutive 1's which indicates an idle transmit line. The software protocol must provide for the short idle period between any two consecutive messages.

- **Programmable Options**

The following features of the HD6803 serial I/O section are programmable:

- format – standard mark/space (NRZ)
- Clock – external or internal
- baud rate – one of 4 per given CPU ϕ_2 clock frequency or external clock ×8 input
- wake-up feature – enabled or disabled
- Interrupt requests – enabled or masked individually for transmitter and receiver data registers
- clock output – internal clock enabled or disabled to Port 2 (Bit 2)
- Port 2 (bits 3 and 4) – dedicated or not dedicated to serial I/O individually for transmitter and receiver.

- **Serial Communications Hardware**

The serial communications hardware is controlled by 4 registers as shown in Figure 14. The registers include:

- an 8-bit control and status register
- a 4-bit rate and mode control register (write only)
- an 8-bit read only receive data register and
- an 8-bit write only transmit data register.

In addition to the four registers, the serial I/O section utilizes bit 3 (serial input) and bit 4 (serial output) of Port 2. Bit 2 of Port 2 is utilized if the internal-clock-out or external-clock-in options are selected.

Transmit/Receive Control and Status (TRCS) Register

The TRCS register consists of an 8-bit register of which all 8 bits may be read while only bits 0~4 may be written. The register is initialized to $20 by reset. The bits in the TRCS register are defined as follows:

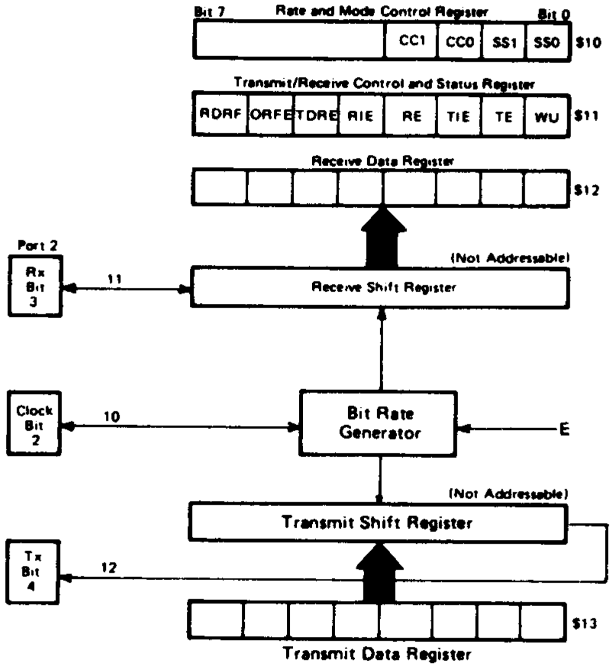


Figure 14 Serial I/O Registers

Bit 0 **WU** "Wake-up" on Next Message – set by HD6803 software and cleared by hardware on receipt of ten consecutive 1's or reset of RE flag. It should be noted that RE flag should be set in advance of CPU set of WU flag.

Bit 1 **TE** Transmit Enable – set by HD6803 to produce preamble of nine consecutive 1's and to enable gating of transmitter output to Port 2, bit 4 regardless of the DDR value corresponding to this bit; when clear, serial I/O has no effect on Port 2 bit 4.

TE set should be after at least one bit time of data transmit rate from the set-up of transmit data rate and mode.

Bit 2 **TIE** Transmit Interrupt Enable – when set, will permit an $\overline{IRQ_2}$ interrupt to occur when bit 5 (TDRE) is set; when clear, the TDRE value is masked from the bus.

Bit 3 **RE** Receiver Enable – when set, gates Port 2 bit 3 to input of receiver regardless of DDR value for this bit; when clear, serial I/O has no effect on Port 2 bit 3.

Bit 4 **RIE** Receiver Interrupt Enable – when set, will permit an $\overline{IRQ_2}$ interrupt to occur when bit 7 (RDRF) or bit 6 (ORFE) is set; when clear, the interrupt is masked.

Transmit/Receive Control and Status Register

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	ADDR : $0011

Bit 5 **TDRE** Transmit Data Register Empty – set by hardware when a transfer is made from the transmit data register to the output shift register. The TDRE bit is cleared by reading the status register, then writing a new byte into the transmit data register. TDRE is initialized to 1 by reset.

Bit 6 **ORFE** Over-Run-Framing Error – set by hardware when an overrun or framing error occurs (receive only).

Rate and Mode Control Register

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

ADDR : $0010

An overrun is defined as a new byte received with last byte still in Data Register/Buffer. A framing error has occured when the byte boundaries in bit stream are not synchronized to bit counter. If WU-flag is set, the ORFE bit will not be set. The ORFE bit is cleard by reading the status register, then reading the Receive Data Register, or by reset.

Bit 7 **RDRF** Receiver Data Register Full-set by hardware when a transfer from the input shift register to the receiver data register is made. If WU-flag is set, the RDRF bit will not be set. The RDRF bit is cleared by reading the status register, then reading the Receive Data Register, or by reset.

Rate and Mode Control Register (RMCR)

The Rate and Mode Control register controls the following serial I/O variables:

- Baud rate
- format
- clocking source,
- Port 2 bit 2 configuration

The register consists of 4 bits all of which are write-only and cleared by reset. The 4 bits in the register may be considered as a pair of 2-bit fields. The two low order bits control the bit rate for internal clocking and the remaining twq bits control the format and clock select logic. The register definition is as follows:

Bit 0 **SS0** } Bit 1 **SS1** } Speed Select – These bits select the Baud rate for the internal clock. The four rates which may be selected are a function of the CPU ϕ_2 clock frequency. Table 5 lists the available Baud rates.

Bit 2 **CC0** } Bit 3 **CC1** } Clock Control and Format Select – this 2-bit field controls the format and clock select logic. Table 6 defines the bit field.

Table 5 SCI Bit Times and Rates

SS1 : SS0	XTAL	2.4576 MHz	4.0 MHz	4.9152 MHz*
	E	614.4 kHz	1.0 MHz	1.2288 MHz
0 0	E ÷ 16	26 μs/38,400 Baud	16 μs/62,500 Baud	13.0 μs/76,800 Baud
0 1	E ÷ 128	208 μs/4,800 Baud	128 μs/7812.5 Baud	104.2 μs/9,600 Baud
1 0	E ÷ 1024	1.67 ms/600 Baud	1.024 ms/976.6 Baud	833.3 μs/1,200 Baud
1 1	E ÷ 4096	6.67 ms/150 Baud	4.096 ms/244.1 Baud	3.33 ms/300 Baud

* HD6803-1 Only

Table 6 SCI Format and Clock Source Control

CC1 : CC0	Format	Clock Source	Port 2 Bit 2	Port 2 Bit 3	Port 2 Bit 4
0 0	–	–	–	–	–
0 1	NRZ	Internal	Not Used	**	**
1 0	NRZ	Internal	Output*	**	**
1 1	NRZ	External	Input	**	**

* Clock output is available regardless of values for bits RE and TE.
** Bit 3 is used for serial input if RE = "1" in TRCS; bit 4 is used for serial output if TE = "1" in TRCS.

Internally Generated Clock

If the user wishes for the serial I/O to furnish a clock, the following requirements are applicable:

- the values of RE and TE are immaterial.
- CC1, CC0 must be set to 10
- the maximum clock rate will be E ÷ 16.
- the clock will be at 1x the bit rate and will have a rising edge at mid-bit.

Externally Generated Clock

If the user wishes to provide an external clock for the serial I/O, the following requirements are applicable:

- the CC1, CC0, field in the Rate and Mode Control Register must be set to 11,
- the external clock must be set to 8 times (x8) the desired baud rate and
- the maximum external clock frequency is 1.0 MHz.

- **Serial Operations**

The serial I/O hardware should be initialized by the HD6803 software prior to operation. This sequence will normally consist of;

- writing the desired operation control bits to the Rate and Mode Control Register and
- writing the desired operational control bits in the Transmit/Receive Control and Status Register.

The Transmitter Enable (TE) and Receiver Enable (RE) bits may be left set for dedicated operations.

Transmit Operations

The transmit operation is enabled by the TE bit in the Transmit/Receive Control and Status Register. This bit when set, gates the output of the serial transmit shift register to Port 2 Bit 4 and takes unconditional control over the Data Direction Register value for Port 2, Bit 4.

Following a $\overline{RES}$ the user should configure both the Rate and Mode Control Register and the Transmit/Receive Control and Status Register for desired operation. Setting the TE bit during this procedure initiates the serial output by first transmitting a nine-bit preamble of 1's. Following the preamble, internal synchronization is established and the transmitter section is ready for operation.

At this point one of two situation exist:

1) if the Transmit Data Register is empty (TDRE = 1), a continuous string of ones will be sent indicating an idle line, or,
2) if data has been loaded into the Transmit Data Register (TDRE = 0), the word is transferred to the output shift register and transmission of the data word will begin.

During the data transmit, the 0 start bit is first transmitted. Then the 8 data bits (beginning with bit 0) followed by the stop bit, are transmitted. When the Transmitter Data Register has been emptied, the hardware sets the TDRE flag bit.

If the HD6803 fails to respond to the flag within the proper time, (TDRE is still set when the next normal transfer from the parallel data register to the serial output register should occur) then a 1 will be sent (instead of a 0) at "Start" bit time, followed by more 1's until more data is supplied to the data register. No 0's will be sent while TDRE remains a 1.

Receive Operation

The receive operation is enabled by the RE bit which gates in the serial input through Port 2, Bit 3. The receiver section operation is conditioned by the contents of the Transmit/Receive Control and Status Register and the Rate and Mode Control Register.

The receiver bit interval is divided into 8 sub-intervals for internal synchronization. In the NRZ Mode, the received bit stream is synchronized by the first 0 (space) encountered.

The approximate center of each bit time is strobed during the next 10 bits. If the tenth bit is not a 1 (stop bit) a framing error is assumed, and bit ORFE is set. If the tenth bit as a 1, the data is transferred to the Receive Data Register, and interrupt flag RDRF is set. If RDRF is still set at the next tenth bit time, ORFE will be set, indicating an overrun has occurred. When the HD6803 responds to either flag (RDRF or ORFE) by reading the status register followed by reading the Data Register, RDRF (or ORFE) will be cleared.

■ RAM CONTROL REGISTER

This register, which is addressed at $0014, gives status information about the standby RAM. A 0 in the RAM enable bit (RAME) will disable the standby RAM, thereby protecting it at power down if V_{CC} Standby is held greater than V_{SBB} volts, as explained previously in the signal description for V_{CC} Standby.

RAM Control Register

	7	6	5	4	3	2	1	0
$0014	STBY PWR	RAME	X	X	X	X	X	X

Bit 0 Not used.
Bit 1 Not used.
Bit 2 Not used.
Bit 3 Not used.
Bit 4 Not used.
Bit 5 Not used.
Bit 6 **RAME** The RAM Enable control bit allows the user the ability to disable the standby RAM. This bit is set to a logic "1" by $\overline{RES}$ which enables the standby RAM and can be written to one or zero under program control. When the RAM is disabled, data is read from external memory.
Bit 7 **STBY PWR** The Standby Power bit is cleared when the standby voltage is removed. This bit is a read/write status flag that the user can read which indicates that the standby RAM voltage has been applied, and the data in the standby RAM is valid.

■ GENERAL DESCRIPTION OF INSTRUCTION SET

The HD6803 is upward object code compatible with the HD6800 as it implements the full HMCS6800 instruction set. The execution times of key instructions have been reduced to increase throughout. In addition, new instructions have been added; these include 16-bit operations and a hardware multiply.

Included in the instruction set section are the following:

- CPU Programming Model–Figure 15.
- Addressing modes
- Accumulator and memory instructions – Table 7
- New instructions
- Index register and stack manipulations instructions – Table 8
- Jump and branch instructions – Table 9
- Condition code register manipulation instructions – Table 10
- Instructions Execution times in machine cycles – Table 11
- Summary of cycle by cycle operation – Table 12
- Summary of undefined instructions – Table 13

- **CPU Programming Model**

The programming model for the HD6803 is shown in Figure 15. The double (D) accumulator is physically the same as the Accumulator A concatenated with the Accumulator B so that any operation using accumulator D will destroy information in A and B.

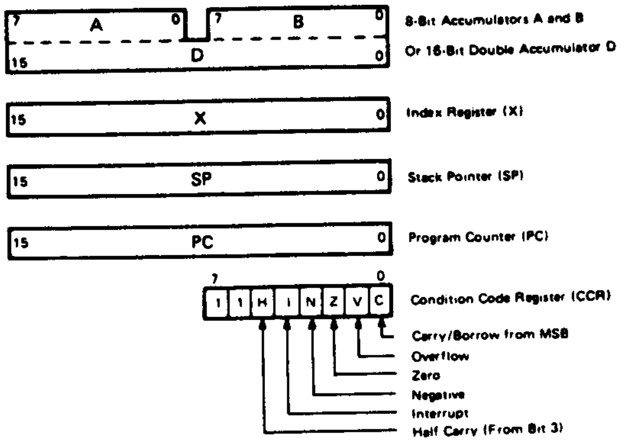


Figure 15 CPU Programming Model

- **CPU Addressing Modes**

The HD6803 8-bit micro processing unit has seven address modes that can be used by a programmer, with the addressing mode a function of both the type of instruction and the coding within the instruction. A summary of the addressing modes for a particular instruction can be found in Table 11 along with the associated instruction execution time that is given in machine cycles. With a clock frequency of 4 MHz, these times would be microseconds.

Accumulator (ACCX) Addressing

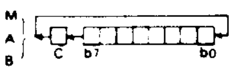
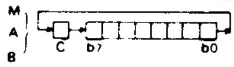
In accumulator only addressing, either accumulator A or accumulator B is specified. These are one-byte instructions.

Immediate Addressing

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

2

Table 7 Accumulator & Memory Instructions

Operations	Mnemonic	Addressing Modes: IMMED. OP	~	#	DIRECT OP	~	#	INDEX OP	~	#	EXTEND OP	~	#	IMPLIED OP	~	#	Boolean/ Arithmetic Operation	Condition Code Register 5 H	4 I	3 N	2 Z	1 V	0 C
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 → B	↕	●	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				A : B + M : M + 1 → A : B	●	●	↕	↕	↕	↕
Add Accumulators	ABA													1B	2	1	A + B → A	↕	●	↕	↕	↕	↕
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	↕	●	↕	↕	↕	↕
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	●
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3				A·M	●	●	↕	↕	R	●
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B·M	●	●	↕	↕	R	●
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	●	●	↕	↕	↕	↕
Compare Accumulators	CBA													11	2	1	A − B	●	●	↕	↕	↕	↕
Complement, 1's	COM							63	6	2	73	6	3				$\bar{M}$ → M	●	●	↕	↕	R	S
	COMA													43	2	1	$\bar{A}$ → A	●	●	↕	↕	R	S
	COMB													53	2	1	$\bar{B}$ → B	●	●	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3				00 − M → M	●	●	↕	↕	①	②
	NEGA													40	2	1	00 − A → A	●	●	↕	↕	①	②
	NEGB													50	2	1	00 − B → B	●	●	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	●	●	↕	↕	↕	③
Decrement	DEC							6A	6	2	7A	6	3				M − 1 → M	●	●	↕	↕	④	●
	DECA													4A	2	1	A − 1 → A	●	●	↕	↕	④	●
	DECB													5A	2	1	B − 1 → B	●	●	↕	↕	④	●
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	●	●	↕	↕	R	●
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	●	●	↕	↕	R	●
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	●	●	↕	↕	⑤	●
	INCA													4C	2	1	A + 1 → A	●	●	↕	↕	⑤	●
	INCB													5C	2	1	B + 1 → B	●	●	↕	↕	⑤	●
Load Accumulator	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 Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M + 1 → B, M → A	●	●	↕	↕	R	●
Multiply Unsigned	MUL													3D	10	1	A × B → A : B	●	●	●	●	●	⑪
OR, Inclusive	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 → Msp, SP − 1 → SP	●	●	●	●	●	●
	PSHB													37	3	1	B → Msp, SP − 1 → SP	●	●	●	●	●	●
Pull Data	PULA													32	4	1	SP + 1 → SP, Msp → A	●	●	●	●	●	●
	PULB													33	4	1	SP + 1 → SP, Msp → B	●	●	●	●	●	●
Rotate Left	ROL							69	6	2	79	6	3				 M A B C b7 b0	●	●	↕	↕	⑥	↕
	ROLA													49	2	1		●	●	↕	↕	⑥	↕
	ROLB													59	2	1		●	●	↕	↕	⑥	↕
Rotate Right	ROR							66	6	2	76	6	3				 M A B C b7 b0	●	●	↕	↕	⑥	↕
	RORA													46	2	1		●	●	↕	↕	⑥	↕
	RORB													56	2	1		●	●	↕	↕	⑥	↕

The Condition Code Register notes are listed after Table 10

(Continued)

HITACHI

Table 7 Accumulator & Memory Instructions (Continued)

Operations	Mnemonic	IMMED. OP	~	#	DIRECT OP	~	#	INDEX OP	~	#	EXTEND OP	~	#	IMPLIED OP	~	#	Boolean/ Arithmetic Operation	5 H	4 I	3 N	2 Z	1 V	0 C
Shift Left Arithmetic	ASL							68	6	2	78	6	3				M: C ← b7 … b0 ← 0	•	•	↕	↕	⑥	↕
	ASLA													48	2	1	A: C ← b7 … b0 ← 0	•	•	↕	↕	⑥	↕
	ASLB													58	2	1	B: C ← b7 … b0 ← 0	•	•	↕	↕	⑥	↕
Double Shift Left, Arithmetic	ASLD													05	3	1	C ← ACC A/ ACC B (A7 … A0 B7 … B0) ← 0	•	•	↕	↕	⑥	↕
Shift Right Arithmetic	ASR							67	6	2	77	6	3				M: b7 (retained) → b7 … b0 → C	•	•	↕	↕	⑥	↕
	ASRA													47	2	1	A: b7 (retained) → b7 … b0 → C	•	•	↕	↕	⑥	↕
	ASRB													57	2	1	B: b7 (retained) → b7 … b0 → C	•	•	↕	↕	⑥	↕
Shift Right Logical	LSR							64	6	2	74	6	3				M: 0 → b7 … b0 → C	•	•	R	↕	⑥	↕
	LSRA													44	2	1	A: 0 → b7 … b0 → C	•	•	R	↕	⑥	↕
	LSRB													54	2	1	B: 0 → b7 … b0 → C	•	•	R	↕	⑥	↕
Double Shift Right Logical	LSRD													04	3	1	0 → ACC A/ ACC B (A7 … A0 B7 … B0) → C	•	•	R	↕	⑥	↕
Store Accumulator	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	•
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3				A → M B → 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	•	•	↕	↕	↕	↕
Double Subtract	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				A : B − M : M + 1 → A : B	•	•	↕	↕	↕	↕
Subtract Accumulators	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	•	•	↕	↕	↕	↕
Transfer Accumulators	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 10.

2

Direct Addressing

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

Extended Addressing

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

Indexed Addressing

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

Implied Addressing

In the implied addressing mode the instruction gives the address (i.e., stack pointer, index register, etc.). These are one-byte instructions.

Relative Addressing

In relative addressing, the address contained in the second byte of the instruction is added to the program counter's lowest 8-bits plus two. The carry or borrow is then added to the high 8-bits. This allows the user to address data within a range of -126 to +129 bytes of the present instruction. These are two-byte instructions.

- **New Instructions**

In addition to the existing 6800 Instruction Set, the following new instructions are incorporated in the HD6803 Microcomputer.

ABX Adds the 8-bit unsigned accumulator B to the 16-bit X-Register taking into account the possible carry out of the low order byte of the X-Register.

ADDD Adds the double precision ACCD* to the double precision value M:M+1 and places the results in ACCD.

ASLD Shifts all bits of ACCD one place to the left. Bit 0 is loaded with zero. The C bit is loaded from the most significant bit of ACCD.

LDD Loads the contents of double precision memory location into the double accumulator A:B. The condition codes are set according to the data.

LSRD Shifts all bits of ACCD one place to the right. Bit 15 is loaded with zero. The C bit is loaded from the least significant bit to ACCD.

MUL Multiplies the 8 bits in accumulator A with the 8 bits in accumulator B to obtain a 16-bit unsigned number in A:B, ACCA contains MSB of result.

PSHX The contents of the index register is pushed onto the stack at the address contained in the stack pointer. The stack pointer is decremented by 2.

PULX The index register is pulled from the stack beginning at the current address contained in the stack pointer +1. The stack pointer is incremented by 2 in total.

STD Stores the contents of double accumulator A:B in memory. The contents of ACCD remain unchanged.

SUBD Subtracts the contents of M:M + 1 from the contents of double accumulator AB and places the result in ACCD.

BRN Never branches. If effect, this instruction can be considered a two byte NOP (No operation) requiring three cycles for execution.

CPX Internal processing modified to permit its use with any conditional branch instruction.

*ACCD is the 16 bit register (A:B) formed by concatenating the A and B accumulators. The A-accumulator is the most significant byte.

Table 8 Index Register and Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		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	SP + 1 → SP	●	●	●	●	●	●
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				$M \rightarrow X_H$, $(M+1) \rightarrow X_L$	●	●	⑦	↕	R	●
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				$M \rightarrow SP_H$, $(M+1) \rightarrow SP_L$	●	●	⑦	↕	R	●
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				$X_H \rightarrow M$, $X_L \rightarrow (M+1)$	●	●	⑦	↕	R	●
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				$SP_H \rightarrow M$, $SP_L \rightarrow (M+1)$	●	●	⑦	↕	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	$X_L \rightarrow M_{sp}$, $SP - 1 \rightarrow SP$ $X_H \rightarrow M_{sp}$, $SP - 1 \rightarrow SP$	●	●	●	●	●	●
Pull Data	PULX													38	5	1	$SP + 1 \rightarrow SP$, $M_{sp} \rightarrow X_H$ $SP + 1 \rightarrow SP$, $M_{sp} \rightarrow X_L$	●	●	●	●	●	●

The Condition Code Register notes are listed after Table 10.

Table 9 Jump and Branch Instructions

Operations	Mnemonic	Addressing Modes															Branch Test	Condition Code Register					
		RELATIVE			DIRECT			INDEX			EXTND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		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													$N \oplus V = 0$	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3	2													$Z + (N \oplus V) = 0$	•	•	•	•	•	•
Branch If Higher	BHI	22	3	2													$C + Z = 0$	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	3	2													$Z + (N \oplus V) = 1$	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3	2													$C + Z = 1$	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3	2													$N \oplus V = 1$	•	•	•	•	•	•
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														•	•	•	•	•	•
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	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		—— ⑧ ——					
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt	WAI													3E	9	1		•	⑨	•	•	•	•

Table 10 Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes			Boolean Operation	Condition Code Register					
		IMPLIED				5	4	3	2	1	0
		OP	~	#		H	I	N	Z	V	C
Clear Carry	CLC	0C	2	1	$0 \rightarrow C$	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	$0 \rightarrow I$	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	$0 \rightarrow V$	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	$1 \rightarrow C$	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	$1 \rightarrow I$	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	$1 \rightarrow V$	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	2	1	$A \rightarrow CCR$	—— ⑩ ——					
CCR → Accumulator A	TPA	07	2	1	$CCR \rightarrow A$	•	•	•	•	•	•

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

① (Bit V) Test: Result = 10000000?
② (Bit C) Test: Result = 00000000?
③ (Bit C) Test: Decimal value of most significant BCD Character greater than nine? (Not cleared if previously set)
④ (Bit V) Test: Operand = 10000000 prior to execution?
⑤ (Bit V) Test: Operand = 01111111 prior to execution?
⑥ (Bit V) Test: Set equal to result of $N \oplus C$ after shift has occurred.
⑦ (Bit N) Test: Result less than zero? (Bit 15 = 1)
⑧ (All) Load Condition Code Register from Stack. (See Special Operations)
⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
⑩ (All) Set according to the contents of Accumulator A.
⑪ (Bit C) Set equal to result of Bit 7 (ACCB)

2

HITACHI

Table 11 Instruction Execution Times in Machine Cycle

	ACCX	Immediate	Direct	Extended	Indexed	Implied	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	2	•	•	6	6	•	•
ASLD	•	•	•	•	•	3	•
ASR	2	•	•	6	6	•	•
BCC	•	•	•	•	•	•	3
BCS	•	•	•	•	•	•	3
BEQ	•	•	•	•	•	•	3
BGE	•	•	•	•	•	•	3
BGT	•	•	•	•	•	•	3
BHI	•	•	•	•	•	•	3
BIT	•	2	3	4	4	•	•
BLE	•	•	•	•	•	•	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	2	•	•	6	6	•	•
CLV	•	•	•	•	•	2	•
CMP	•	2	3	4	4	•	•
COM	2	•	•	6	6	•	•
CPX	•	4	5	6	6	•	•
DAA	•	•	•	•	•	2	•
DEC	2	•	•	6	6	•	•
DES	•	•	•	•	•	3	•
DEX	•	•	•	•	•	3	•
EOR	•	2	3	4	4	•	•
INC	2	•	•	6	6	•	•
INS	•	•	•	•	•	3	•
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	•	•
LSR	2	•	•	6	6	•	•
LSRD	•	•	•	•	•	3	•
MUL	•	•	•	•	•	10	•
NEG	2	•	•	6	6	•	•
NOP	•	•	•	•	•	2	•
ORA	•	2	3	4	4	•	•
PSH	3	•	•	•	•	•	•
PSHX	•	•	•	•	•	4	•
PUL	4	•	•	•	•	•	•
PULX	•	•	•	•	•	5	•
ROL	2	•	•	6	6	•	•
ROR	2	•	•	6	6	•	•
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	2	•	•	6	6	•	•
TSX	•	•	•	•	•	3	•
TXS	•	•	•	•	•	3	•
WAI	•	•	•	•	•	9	•

- **Summary of Cycle by Cycle Operation**

Table 12 provides a detailed description of the information present on the Address Bus, Data Bus, and the Read/Write line (R/$\overline{W}$) during each cycle for each instruction.

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

Table 12 Cycle by Cycle Operation

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

(Continued)

2

Table 12 Cycle by Cycle Operation (Continued)

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
LDD		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*		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	1	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)

* In the TST instruction, R/$\overline{W}$ line of the sixth cycle is "1" level, and AB = FFFF, DB = Low Byte of Reset Vector.

(Continued)

Table 12 Cycle by Cycle Operation (Continued)

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 (High Order Byte)
AND ORA		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
BIT SBC		4	Address of Operand	1	Operand Data
CMP SUB					
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)

* In the TST instruction, R/$\overline{W}$ line of the sixth cycle is "1" level, and AB = FFFF, DB = Low Byte of Reset Vector.

(Continued)

2

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/W̄ Line	Data Bus
IMPLIED					
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Op Code of Next Instruction
ABX	3	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
ASLD LSRD	3	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
DES INS	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Op Code of Next Instruction
		3	Previous Register Contents	1	Irrelevant Data
INX DEX	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
PSHA PSHB	3	1	Op Code Address	1	Op Code
		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 PULB	4	1	Op Code Address	1	Op Code
		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)

(Continued)

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
WAI**		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
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)

(Continued)

** While the MPU is in the "Wait" state, its bus state will appear as a series of MPU reads of an address which is seven locations less than the original contents of the Stack Pointer. Contrary to the HD6800, none of the ports are driven to the high impedance state by a WAI instruction.

Table 12 Cycle by Cycle Operation (Continued)

RELATIVE

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
BCC BHT BNE BCS BLE BPL BEQ BLS BRA BGE BLT BVC BGT BMT BVS BRN	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Branch Offset Low Byte of Restart Vector
BSR	6	1 2 3 4 5 6	Op Code Address Op Code Address + 1 Address Bus FFFF Subroutine Starting Address Stack Pointer Stack Pointer − 1	1 1 1 1 0 0	Op Code Branch Offset Low Byte of Restart Vector Op Code of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte)

• **Summary of Undefined Instruction Operations**

The HD6803 has 36 underfined instructions. When these are carried out, the contents of Register and Memory in MPU change at random.

When the op codes (4E, 5E) are used to execute, the MPU continues to increase the program counter and it will not stop until the Reset signal enters. These op codes are used to test the LSI.

Table 13 Op codes Map

HD6803 MICROPROCESSOR INSTRUCTIONS

OP CODE						ACC A	ACC B	IND	EXT	ACCA or SP IMM	ACCA or SP DIR	ACCA or SP IND	ACCA or SP EXT	ACCB or X IMM	ACCB or X DIR	ACCB or X IND	ACCB or X EXT	
LO \ HI		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0		SBA	BRA	TSX	NEG				SUB								0
0001	1	NOP	CBA	BRN	INS					CMP								1
0010	2			BHI	PULA (+1)					SBC								2
0011	3			BLS	PULB (+1)	COM				•	SUBD (+2)			•	ADDD (+2)			3
0100	4	LSRD (+1)		BCC	DES	LSR				AND								4
0101	5	ASLD (+1)		BCS	TXS					BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7
1000	8	INX (+1)		BVC	PULX (+2)	ASL				EOR								8
1001	9	DEX (+1)	DAA	BVS	RTS (+2)	ROL				ADC								9
1010	A	CLV		BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI (+7)					ADD								B
1100	C	CLC		BGE	PSHX (+1)	INC				•	CPX (+2)			•	LDD (+1)			C
1101	D	SEC		BLT	MUL (+7)	TST				BSR (+4)	JSR (+2)			• (+1)	STD (+1)			D
1110	E	CLI		BGT	WAI (+6)	••		JMP (-3)		•	LDS (+1)			•	LDX (+1)			E
1111	F	SEI		BLE	SWI (+9)	CLR				• (+1)	STS (+1)			• (+1)	STX (+1)			F
BYTE/CYCLE		1/2	1/2	2/3	1/3	1/2	1/2	2/6	3/6	2/2	2/3	2/4	3/4	2/2	2/3	2/4	3/4	

[NOTES]
1) Undefined Op codes are marked with ▭.
2) ( ) indicate that the number in parenthesis must be added to the cycle count for that instruction.
3) The instructions shown below are all 3 bytes and are marked with "•".
Immediate addressing mode of SUBD, CPX, LDS, ADDD, LDD and LDX instructions, and undefined op codes (8F, CD, CF).
4) The Op codes (4E, 5E) are 1 byte/∞ cycles instructions, and are marked with "••"

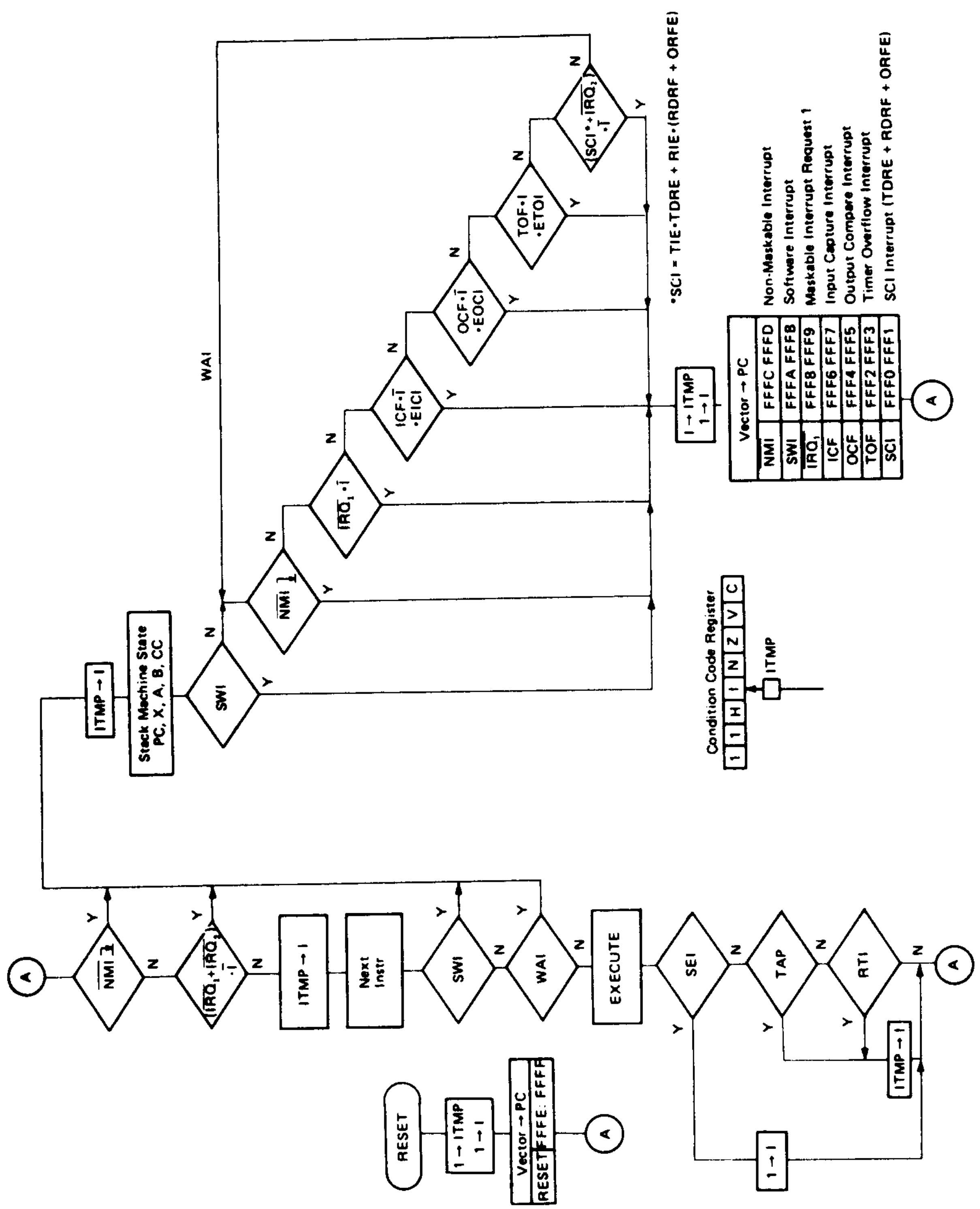


Figure 16 Interrupt Flowchart

2

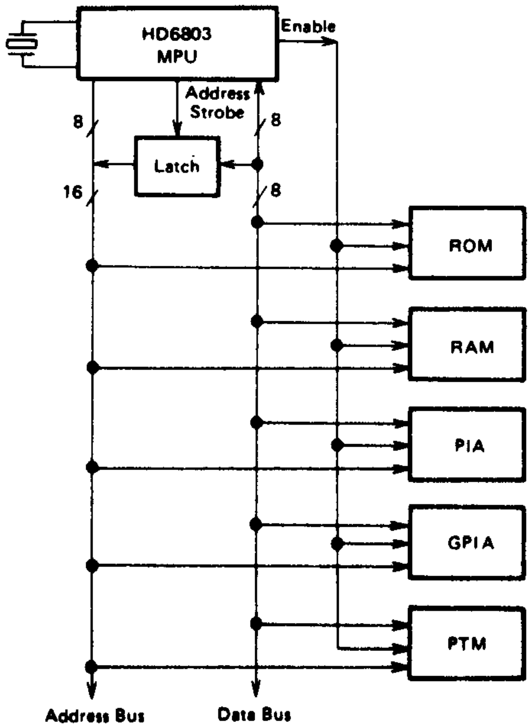


Figure 17 HD6803 MPU Expanded Multiplexed Bus

- **Caution for the HD6803 Family SCI, TIMER Status Flag**

The flags shown in Table 14 are cleared by reading/writing (flag reset condition 2) the data register corresponding to each flag after reading the status register (flag reset condition 1).

To clear the flag correctly, take the following procedure:

1. Read the status register.
2. Test the flag.
3. Read the data register.

Table 14 Status Flag Reset Conditions

	Status Flag	Flag Reset Condition 1 (Status Register)	Flag Reset Condition 2 (Data Register)
TIMER	ICF	When each flag is "1", TRCSR/Read	ICR/Read
	OCF		OCR/Write
	TOF		TC/Read
SCI	RDRF	When each flag is "1", TRCSR/Read	RDR/Read
	ORFE		
	TDRE		TDR/Write