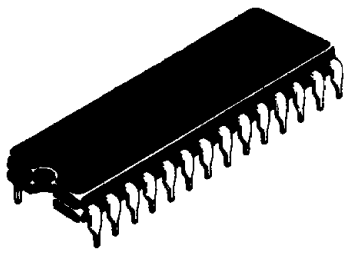
HD6805T2

MCU (Microcomputer Unit with PLL Logic)

The HD6805T2 is an 8-bit Microcomputer Unit (MCU) which contains a CPU, on-chip clock, ROM, RAM, I/O, Timer, and the PLL Logic for an RF synthesizer. It meets the needs of users who needs economical single chip microcomputer with the proven abilities of MPU instruction set of HD6800.

The Principal characteristics of the hardware and software of the MCU are listed below.

HD6805T2P



(DP-28)

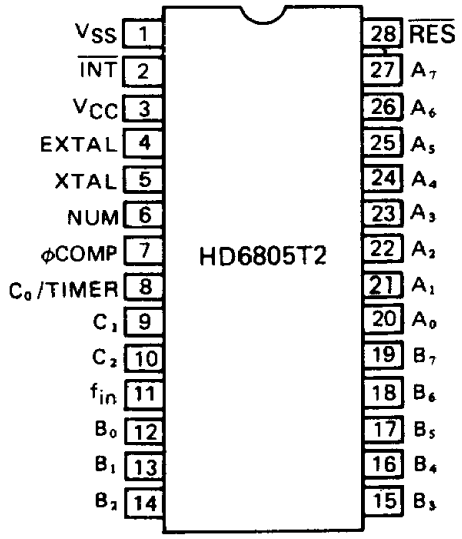
■ HARDWARE FEATURES

- 8-bit Architecture
- 64 Bytes of RAM
- Memory Mapped I/O
- 2508 Bytes of User ROM
- Internal 8-bit Timer with 7-bit Prescaler
- Timer Start/Stop and Source Select
- Vectored Interrupts — External and Timer
- 19 TTL/CMOS Compatible I/O Lines; 8 Lines are LED Compatible
- On-Chip Clock Circuit
- Self-Check Mode
- Master Reset
- Low Voltage Inhibit
- 14-Bit Binary Variable Divider
- 10-Stage Mask-Programmable Reference Divider
- Three-State Phase and Frequency Comparator
- Suitable for TV Frequency Synthesizers
- $5V_{dc}$ Single Supply

■ SOFTWARE FEATURES

- Resembles HD6800
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Handing
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to ROM, RAM, and I/O
- Compatible with MC6805T2

■ PIN ARRANGEMENT

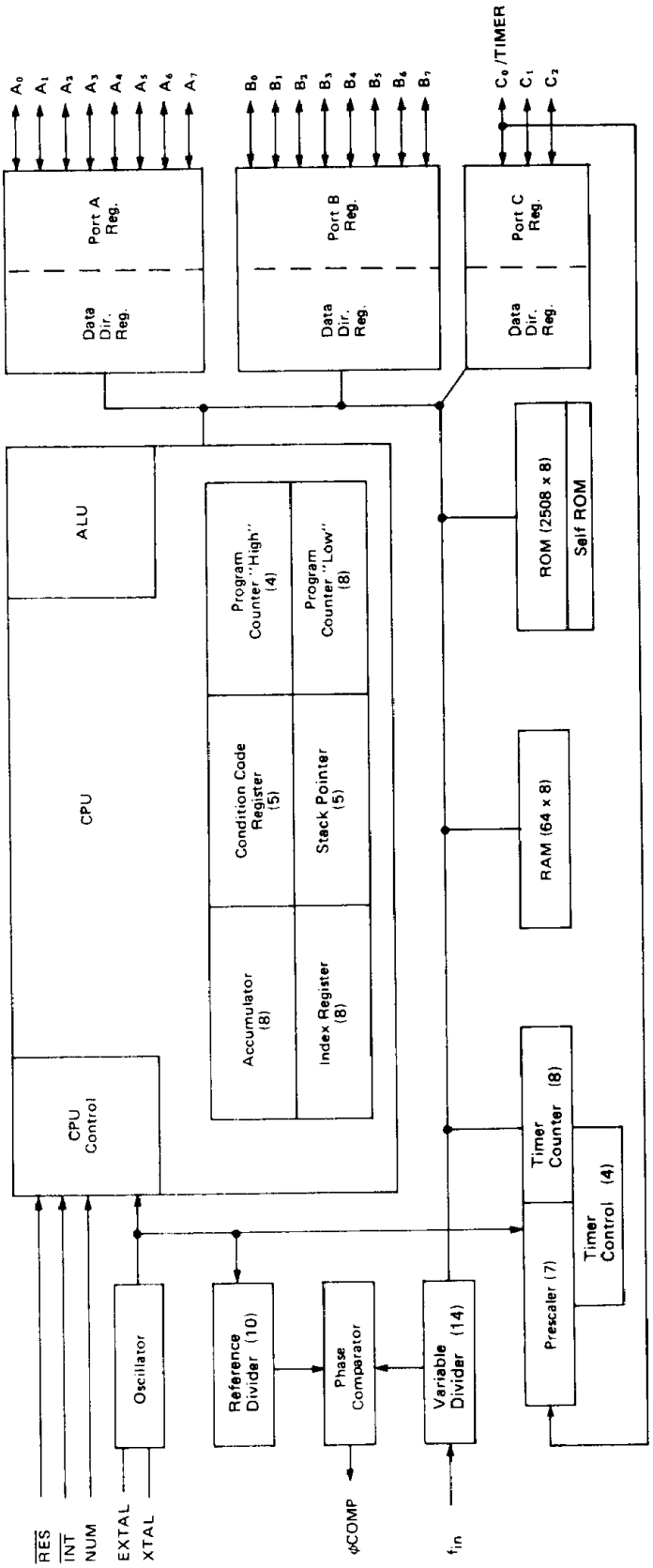


(Top View)

■ PROGRAM DEVELOPMENT SUPPORT TOOLS

- Cross assembler software for use with IBM PCs and compatibles

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}*	−0.3 to +7.0	V
Input Voltage (Except ϕCOMP)	V_{in}*	−0.3 to +7.0	V
Input Voltage (ϕCOMP)		−0.3 to +12.0	V
Operating Temperature	T_{opr}	0 to +70	°C
Storage Temperature	T_{stg}	−55 to +150	°C

* With respect to V_{SS} (SYSTEM GND).
(Note) When the maximum ratings are exceeded, the LSI may be irreparably damaged. Recommended operating conditions should be adhered to.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5.25V ± 0.5V, V_{SS} = GND, Ta = 0 to +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
	$\overline{INT}$*			3.0	–	V_{CC}	V
	All Other Except f_{in}			2.0	–	V_{CC}	V
Input "High" Voltage ϕCOMP	Normal Mode			2.0	–	V_{CC}	V
	Self-Check Mode**			9.0	–	11.0	V
Input "Low" Voltage	$\overline{RES}$	V_{IL}		−0.3	–	0.8	V
	$\overline{INT}$*			−0.3	–	0.8	V
	EXTAL			−0.3	–	0.6	V
	All Other Except f_{in}			−0.3	–	0.8	V
Power Dissipation		P_D	Not Port Loading	–	–	850	mW
AC Coupled Input Voltage Swing	f_{in}	V_{FIP}		0.5	–	2.4	V_{ACP-P}
Input Leak Current	f_{in}	$\lvert I_{IL} \rvert$		–	–	40	μA
Output "Low" Current	ϕCOMP	I_{CML}	V_{OL} = 1.0V	–	–	300	μA
Output "High" Current	ϕCOMP	I_{CMH}	$V_{OH} = V_{CC} - 1V$	–	–	200	μA
Input Leak Current	ϕCOMP	$\lvert I_{IL} \rvert$		–	–	1.0	μA
Low Voltage Recover		LVR		–	–	4.75	V
Low Voltage Inhibit		LVI		–	4.0	–	V
Input Leak Current	$\overline{INT}$	$\lvert I_{IL} \rvert$	V_{in} = 0.4V	–	–	50	μA
	EXTAL			–	–	1600	μA
	$\overline{RES}$		V_{in} = 0.8V	4.0	–	50	μA

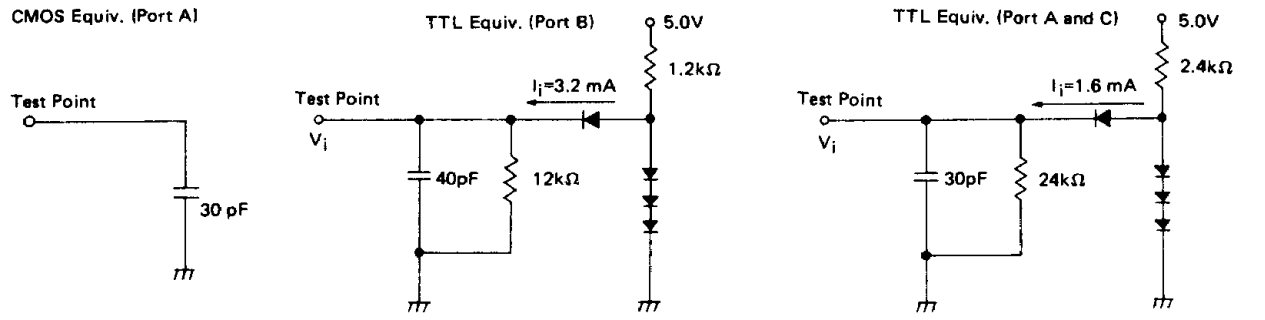
* Internal biasing makes the input float to approximately 2.0V when unused.
** In self-check mode, ϕCOMP may be connected to V_{IH} through 10 kΩ register.

- **AC CHARACTERISTICS (V_{CC} = 5.25V ± 0.5V, V_{SS} = GND, Ta = 0 to +70°C, unless otherwise noted.)**

Item	Symbol	Test Condition	min.	typ.	max.	Unit
Clock Frequency	f_{cl}		0.4	–	4.2	MHz
Cycle Time	t_{cyc}		0.95	–	10	μs
$\overline{INT}$ Pulse Width	t_{IWL}		t_{cyc}+ 250	–	–	ns
$\overline{RES}$ Pulse Width	t_{RWL}		t_{cyc}+ 250	–	–	ns
TIMER Pulse Width	t_{TWL}		t_{cyc}+ 250	–	–	ns
Delay Time Reset	t_{RHL}	External Cap. = 1.0μF	–	100	–	ms
Input Frequency	f_{in}		1	–	16	MHz
Input Frequency Rise Time at f_{in}	t_{inr}		–	–	20	ns
Input Frequency Fall Time at f_{in}	t_{inf}		–	–	20	ns
Duty Cycle of f_{in} and External Input on EXTAL	–		40	–	60	%
Injection Pulse Active Time	t_{err}		–	70	–	ns
Input Capacitance — XTAL	C_{in}	V_{in} = 0V	–	–	35	pF
Input Capacitance — All Other	C_{in}	V_{in} = 0V	–	–	10	pF

- **PORT ELECTRICAL CHARACTERISTICS (V_{CC} = 5.25V ± 0.5V, V_{SS} = GND, Ta = 0 to +70°C, unless otherwise noted.)**

Item		Symbol	Test Condition	min.	typ.	max.	Unit
Output "High" Voltage	Port A with CMOS Drive Enable	V_{OH}	I_{OH} = −100μA	2.4	–	–	V
			I_{OH} = −10μA	3.5	–	–	V
	Port A with CMOS Drive Disable		I_{OH} = −100μA	2.4	–	–	V
	Port B		I_{OH} = −200μA	2.4	–	–	V
	Port C		I_{OH} = −100μA	2.4	–	–	V
Output "Low" Voltage	Port A	V_{OL}	I_{OL} = 1.6mA	–	–	0.4	V
	Port B		I_{OL} = 3.2mA	–	–	0.4	V
			I_{OL} = 10mA	–	–	1.0	V
	Port C		I_{OL} = 1.6mA	–	–	0.4	V
Input "High" Voltage	Port A, B, C	V_{IH}		2.0	–	V_{CC}	V
Input "Low" Voltage		V_{IL}		−0.3	–	0.8	V
Input Leak Current	Port A with CMOS Drive Enable	$\lvert I_{IL} \rvert$	V_{in} = 2.0V	–	–	300	μA
			V_{in} = 0.8V	–	–	500	μA
	Port A with CMOS Drive Disable			–	–	20	μA
	Port B			–	–	20	μA
	Port C			–	–	20	μA



(Note) 1. Load capacitance contains the floating capacitance of the probe, the jig, etc.
2. All diodes are 1S2074 Ⓗ or the similar.

Figure 1 Bus Timing Test Loads

■ SIGNAL DESCRIPTION

The following describes the input and output signals of HD6805T2.

• V_{CC}, V_{SS}

The MCU is supplied power through these pins. V_{CC} is 5.25V ± 0.5V. V_{SS} is grounded.

• $\overline{INT}$

This pin provides an external interrupt to the MCU. For more details, see INTERRUPTS.

• XTAL, EXTAL

These are control input pins for the internal clock circuit. Connection of these pins to a crystal (AT cut, 4.2 MHz maximum) or to an external input provides the internal oscillator with varying degrees of stability. For suggestions pertaining to these pins, see INTERNAL OSCILLATOR OPTIONS.

• f_{in}

This f_{in} pin is a high frequency digital input to the variable divider portion of the on-chip frequency synthesizer. The reference frequency for the phase lock loop is divided down from internal clock ϕ_2. The frequency synthesizer features are explained in PHASE LOCK LOOP.

• ϕCOMP

This is a three-state output signal. The state varies according to the result of the comparison between the internal reference frequency and the variable divided signal. See PLL for details. ϕCOMP is raised to 9V through 10 kΩ in self-check mode.

• $\overline{RES}$

Besides the resetting capability which the MCU already has, this pin also makes resetting of the MCU possible. See RESETS for more details.

• NUM

This pin should be grounded as it is not applicable to users.

• INPUT/OUTPUT Lines (A_0 to A_7, B_0 to B_7, C_0 to C_2)

These 19 lines form two 8-bit ports (Port A, Port B) and one 3-bit port (Port C). By software control of the data direction registers, these lines can be programmed to be either inputs or outputs. See INPUTS/OUTPUTS for more information. The C_0/TIMER pin also can be programmed as an external input to the internal timer. For information on the timer modes, see TIMER.

■ MEMORY

The MCU memory diagram is indicated in Figure 2. The MCU, with its program counter, can address 4096 bytes of memory and I/O registers. The MCU has implemented 2698 of the 4096 memory locations, 2508 bytes user ROM, 116 bytes self-check ROM, 64 bytes of user RAM, 6 bytes of port I/O, 2 timer registers, and 2 PLL registers. The user ROM is divided into four areas. The first area (begins at $080) provides users with access to ROM locations by using the direct and table look-up indexed addressing mode. The second and third user ROM areas begin at memory location $100 and $D40 respectively. The last eight-byte user ROM which begins at $FF8 is for the interrupt vectors.

The first 16 memory locations of the MCU are reserved for I/O features and 10 of them have been implemented. These locations are used for the ports, the port DDRs, the timer, and the PLL registers.

The MCU has 64 bytes of user RAM. 31 bytes out of the 64 bytes are shared with the stack area. Careful utilization of the stack is necessary when data shares the stack area.

While interrupt and subroutine calls are processed to save the processor state, the shared stack area is occupied.

The register contents are saved in the stack as indicated in Figure 3. The low order byte (PCL) of the program counter is stacked first as the stack pointer decrements during saving. Next, the high order four bits (PCH) are stacked. This ensures that the program counter is loaded correctly after pulls from the stack, as the stack pointer increments when it fetches data from the stack. Only when a subroutine call is made, the program counter (PCH, PCL) contents are saved onto the stack, and the remaining CPU registers are not saved.

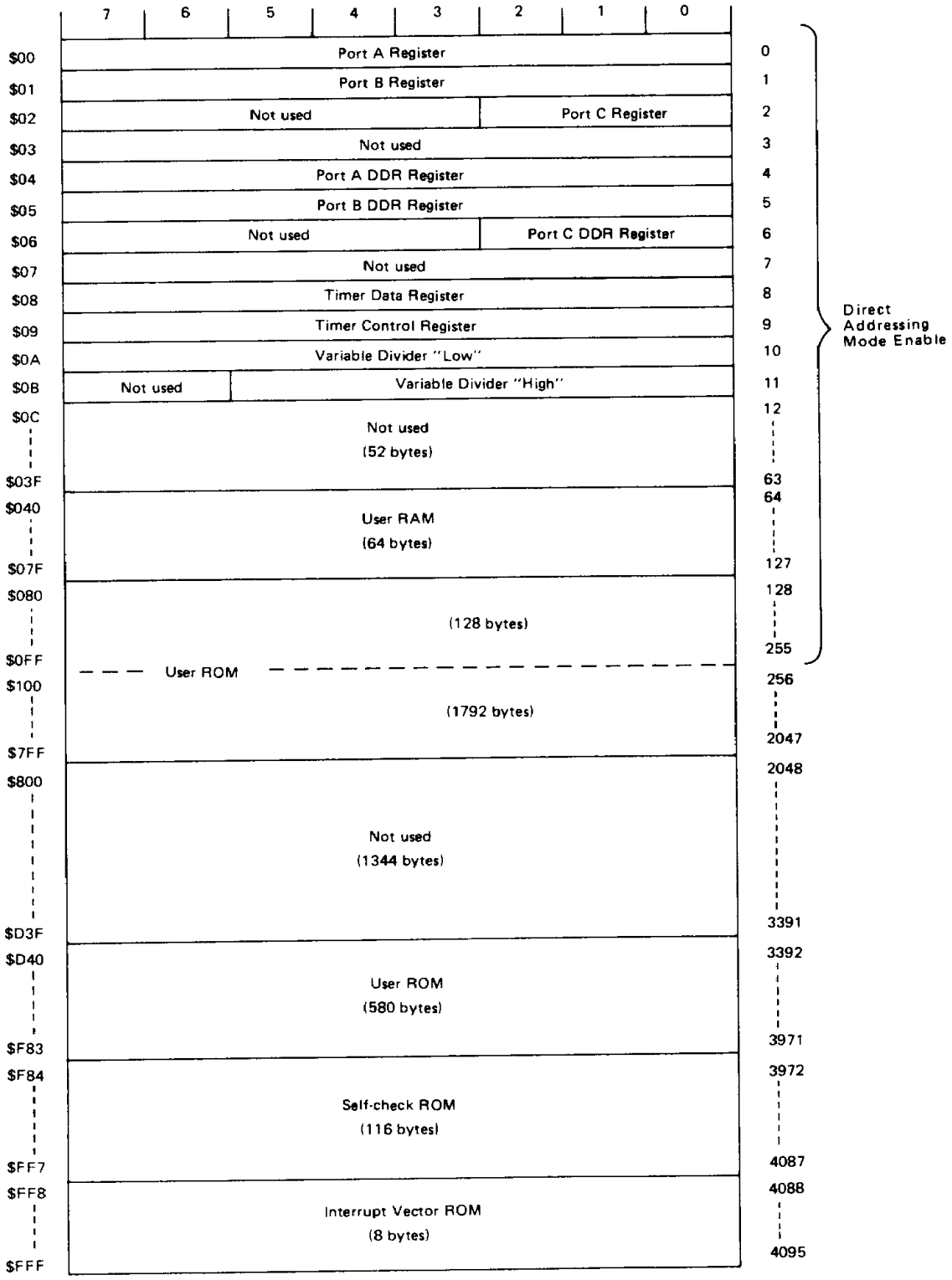
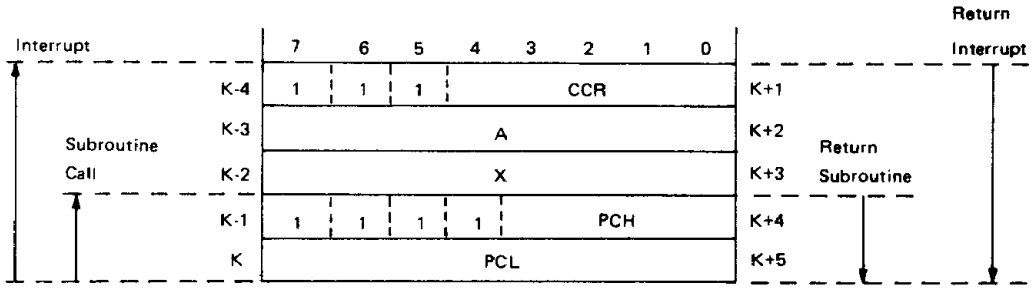


Figure 2 MCU Memory Configuration



Only PCH and PCL are saved for subroutine calls.

Figure 3 Interrupt Stacking Order

■ REGISTERS

The MCU provides five registers for the programmer which are indicated in Figure 4. These registers are explained in the following.

• Accumulator (A)

The accumulator is an 8-bit general purpose register. It holds operands and results of arithmetic calculations or data manipulations.

• Index Register (X)

The index register is an 8-bit register for the indexed addressing mode. It has an 8-bit address which can be added to an offset value to make an effective address. When using read/modify/write instructions, it may also be used for data manipulation and limited calculations. When not required by the code sequence being executed, it can be a temporary storage area.

• Program Counter (PC)

The program counter is an 12-bit register. It contains the address that decides which instruction is to be executed next.

• Stack Pointer (SP)

The stack pointer is a 12-bit register. It contains the address of the next free location in the stack. Firstly, the stack pointer is set to location $07F. Then it is decremented as data is being pushed onto the stack and incremented as data is being pulled from the stack. The seven most crucial bits of the stack pointer are set to 0000011 permanently. The stack pointer is set to location $07F during and MCU reset or the reset stack pointer (RSP) instruction. Subroutines and interrupts can be nested down to location $061 which enables the programmer to use a maximum of 15 levels of subroutine calls.

• Condition Code Register (CC)

The condition code register is a 5-bit register. In the register, the results of the instruction just executed is indicated or flagged by each bit. These bits can be individually program tested specific action taken as a result of their state. The following paragraphs explain each individual code register bit.

Half Carry (H)

Indicates that a carry occurred between bits 3 and 4 during an arithmetic operation (ADD and ADC).

Interrupt (I)

This bit is set to mask external interrupt ($\overline{INT}$) and the timer. If an interrupt takes place while this bit is set, it is latched and will not be processed until the interrupt bit is reset.

Negative (N)

Indicates that the result of the last data, arithmetic, or logical manipulation was negative (bit 7 in result equal to a logic "one").

Zero (Z)

It indicates that the result of the last data, arithmetic, or logical manipulation was zero.

Carry/Borrow (C)

During the last arithmetic operation, it indicates that a carry or borrow out of the arithmetic unit (ALU) occurred. During branch instructions, rotates, bit test, and shifts, this bit is also affected.

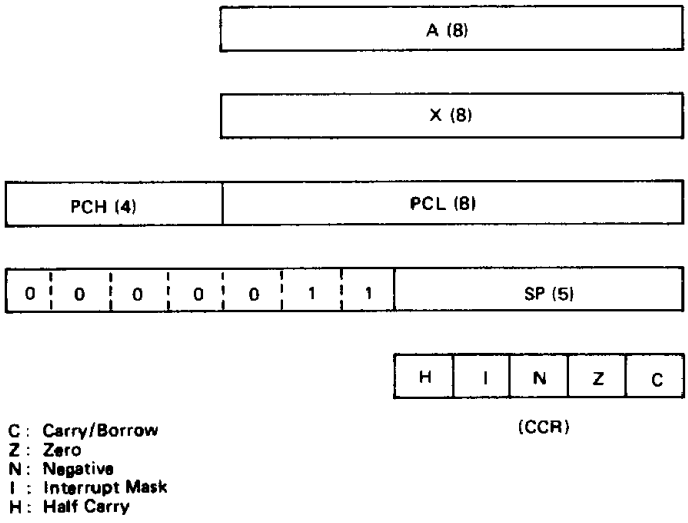


Figure 4 Programming Model

■ TIMER

The MCU timer circuitry is indicated in Figure 5. Program control enables the 8-bit counter to be loaded and the clock input (prescaler output) decrements the counter toward zero. When the timer reaches zero, the timer interrupt request bit (bit 7) in the Timer Control Register (TCR) is set. The timer interrupt mask bit (bit 6) in the TCR enables the timer interrupt to be masked (disabled). The timer also becomes inhibited by the interrupt bit (I-bit) in the Condition Code Register. When the MCU responds to the interrupt requirement, it maintains the present CPU state in the stack, fetches the timer interrupt vectors from locations $FF8 and $FF9, and executes the interrupt routine. See INTERRUPT for more details.

The timer clock input is established by way of bit 5 (TCR 5) in the Timer Control Register. When this bit is set to a logic "one" (external mode), the C_0/TIMER pin is the time clock source. In this mode, a mask option selects either the gated ϕ_2 with C_0 or the positive transition on C_0/TIMER as timer clock source. This makes pulse widths or pulse counts easily measured. The timer clock source is the internal ϕ_2 when TCR 5 is set to a logic "zero". When bit 4 in the Timer Control Register is set to a logic "one", the time clock source is disabled.

The timer continues to count past zero, falling through to $FF from zero, and then continuing to count. The counter can be monitored by reading the Timer Data Register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred without disturbing the counting process.

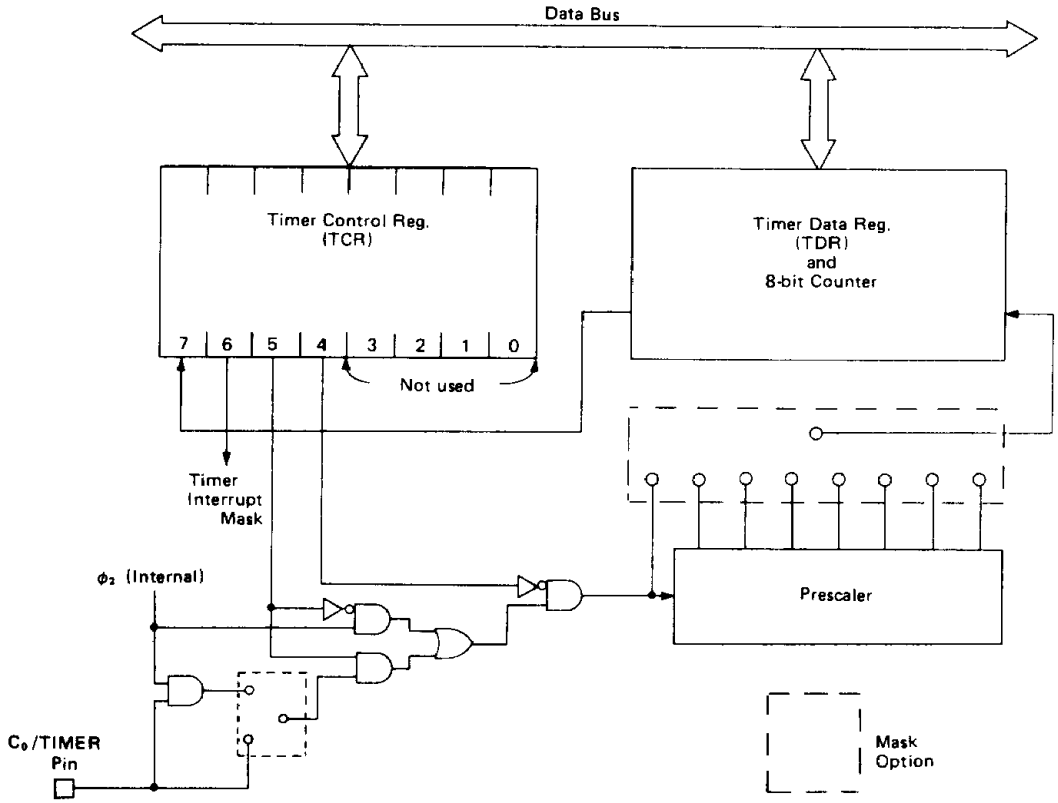


Figure 5 Timer Block Diagram

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

TCR7 – – Timer Interrupt Request Status Bit;
Set when TDR "0"; cleared to "0" during reset.

TCR6 – – Timer Interrupt Mask Bit;
When "1", timer interrupt is masked (disabled). Set to "1" during reset.

TCR5 – – External Timer Source;
External when set to "1" and internal when set to "0". Cleared to "0" during reset.

TCR4 – – Disable Timer;
Timer source is disconnected when set to "1" and timer input enabled when set to "0". Cleared to "0" during reset.

TCR bits 3, 2, 1 and 0; set to all "1" 's (not used).

■ SELF-CHECK

The MCU has a self-check capability which allows an internal check to see whether a port is functional or not. Connect the MCU as indicated in Figure 6 and monitor the output of Port C bit 2 for an oscillation of approximately 7 Hz. Pin 7, a 9 volt level on the ϕCOMP input, energizes the ROM-based self-check feature. The self-check program exercises the timer, interrupts, I/O ports, RAM, and ROM.

■ RESETS

There are three ways to reset the MCU; initial power up, the external reset input $\overline{RES}$), and by an internal low voltage detect circuit. See Figure 7 for details. All the I/O ports are initialized to Input Mode (DDR's are cleared) during RESET.

A delay of t_{RHL} is essential upon power up before allowing the reset input to go "High".

This time allows the internal crystal oscillator to stabilize. Sufficient delay can be provided by connecting a capacitor to the $\overline{RES}$ input as shown in Figure 8.

■ INTERNAL OSCILLATOR

The design of the internal oscillator circuit aims at the requirement of minimum of external components. A crystal or an external signal can be used to drive the internal oscillator. Figures 9 and 10 show different connection methods. Figure 11 indicates the crystal specifications.

The crystal oscillator startup time changes according to many variables: crystal parameters (especially R_S), oscillator load capacitances, IC parameters, ambient temperature, and supply capacitances. To ensure rapid oscillator startup, neither the load capacitance nor the crystal characteristics should exceed the recommended value.

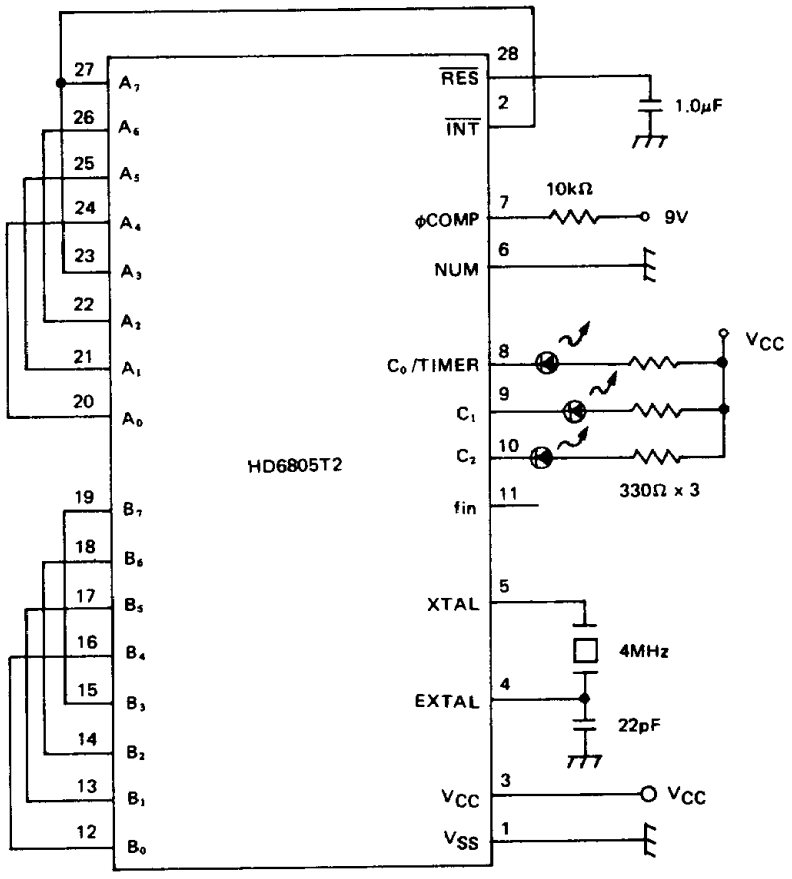


Figure 6 Self Check Connections

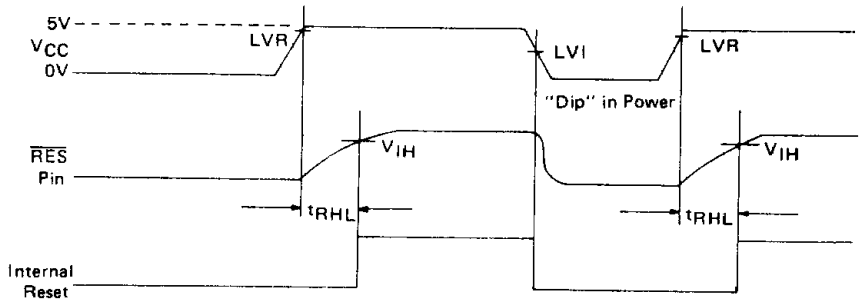


Figure 7 Power Up and $\overline{RES}$ Timing

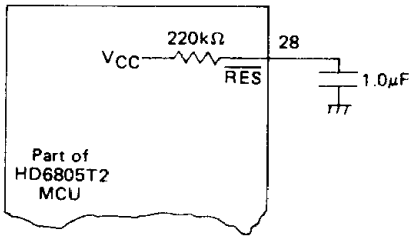


Figure 8 Power Up Reset Delay Circuit

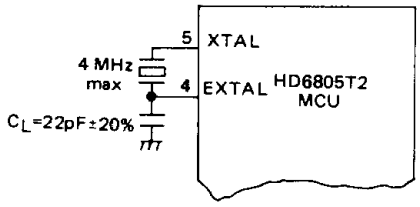


Figure 9 Crystal

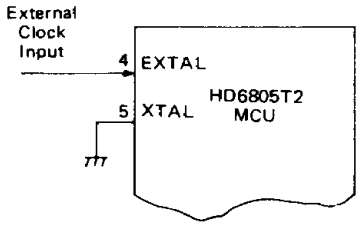


Figure 10 External Clock

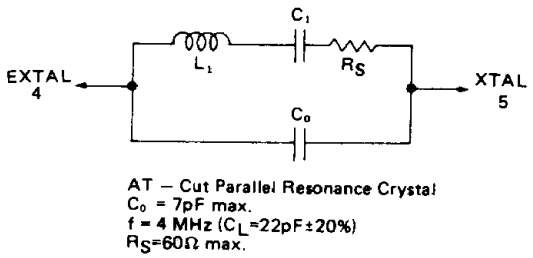


Figure 11 Crystal Parameters

■ INTERRUPTS

The MCU can be interrupted three ways as follows: ① through the external interrupt ($\overline{INT}$) input pin, ② the internal timer interrupt request, ③ the software interrupt instruction (SWI). If any interrupt occurs, processing is in pending, the current CPU state is saved in the stack, the interrupt bit (I) in the condition Code Register is set, the address of the interrupt routine is received from the proper interrupt vector address, and finally the interrupt routine is carried out. The complete stacking the CPU registers, setting the I-bit, and vector fetching, a total of 11 t_{cyc} periods are required.

Figure 12 illustrates a flowchart of the interrupt sequence. When a return from interrupt (RTI) instruction is issued, the interrupt service routine must be terminated to cause the MCU to resume processing of the program held by the interrupt (by popping off the previous CPU state). Table 1 lists the interrupts, their priority, and the address of the vector containing the start address of the proper interrupt service routine. The interrupt priority applies to those suspended when the CPU is ready for a new interrupt. Though $\overline{RES}$ is listed in Table 1 in that it is occasionally regarded as an interrupt, it is not normally used as such. When the interrupt mask bit in the Condition Code Register is set, the interrupt is latched for executing a later interrupt.

The external interrupt is internally synchronized and then latched onto the negative edge of $\overline{INT}$, which can be driven by a digital signal at a maximum period of t_{IWL}.

Table 1

Interrupt	Priority	Vector Address
$\overline{RES}$	1	$FFE and $FFF
SWI	2*	$FFC and $FFD
$\overline{INT}$	3	$FFA and $FFB
TIMER	4	$FF8 and $FF9

* Priority 2 applies when the I-bit in the condition Code Register is set. When I = 0, SWI has priority 4, like any other instruction. Therefore, $\overline{INT}$ has priority 2 and the timer has priority 3.

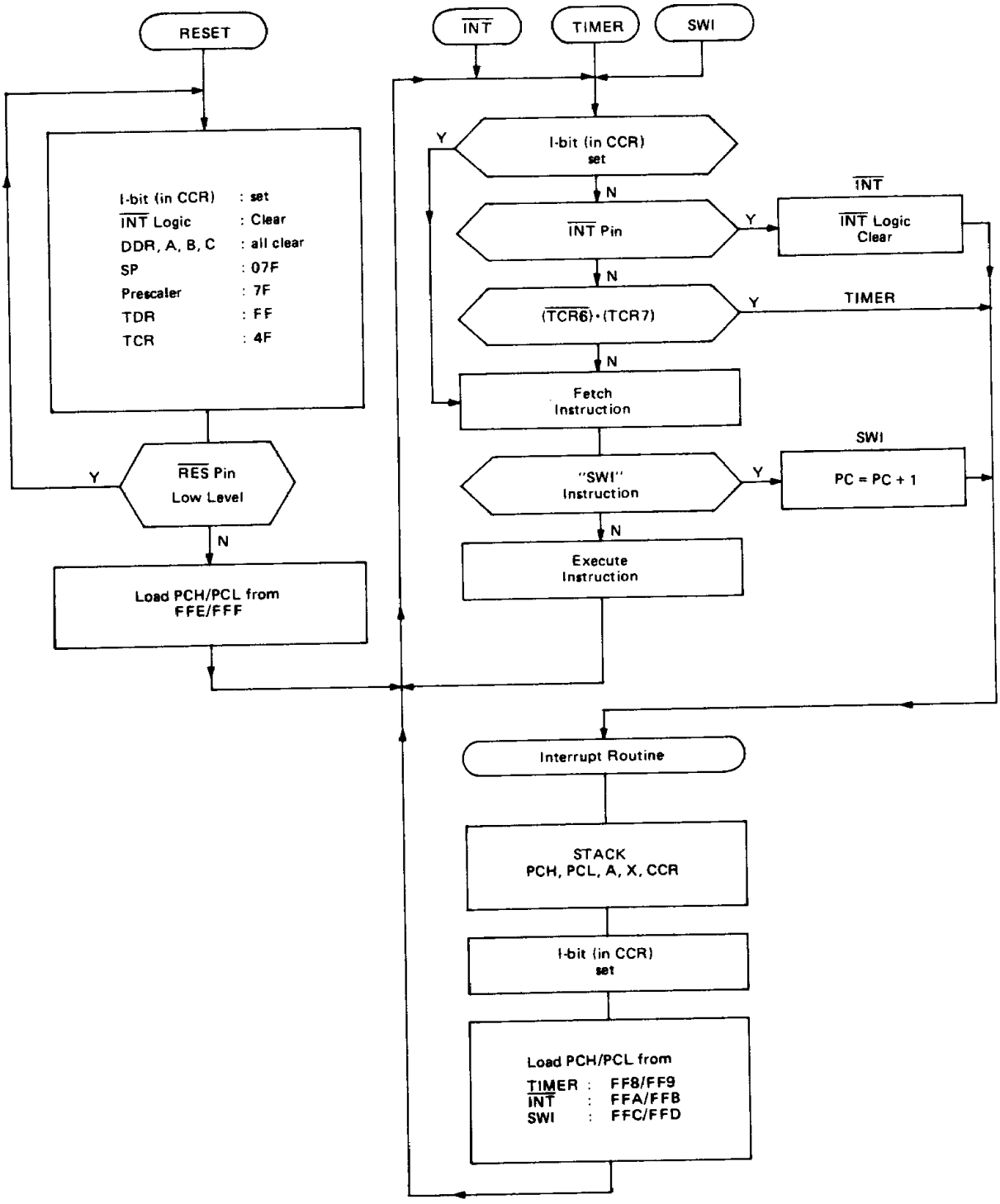
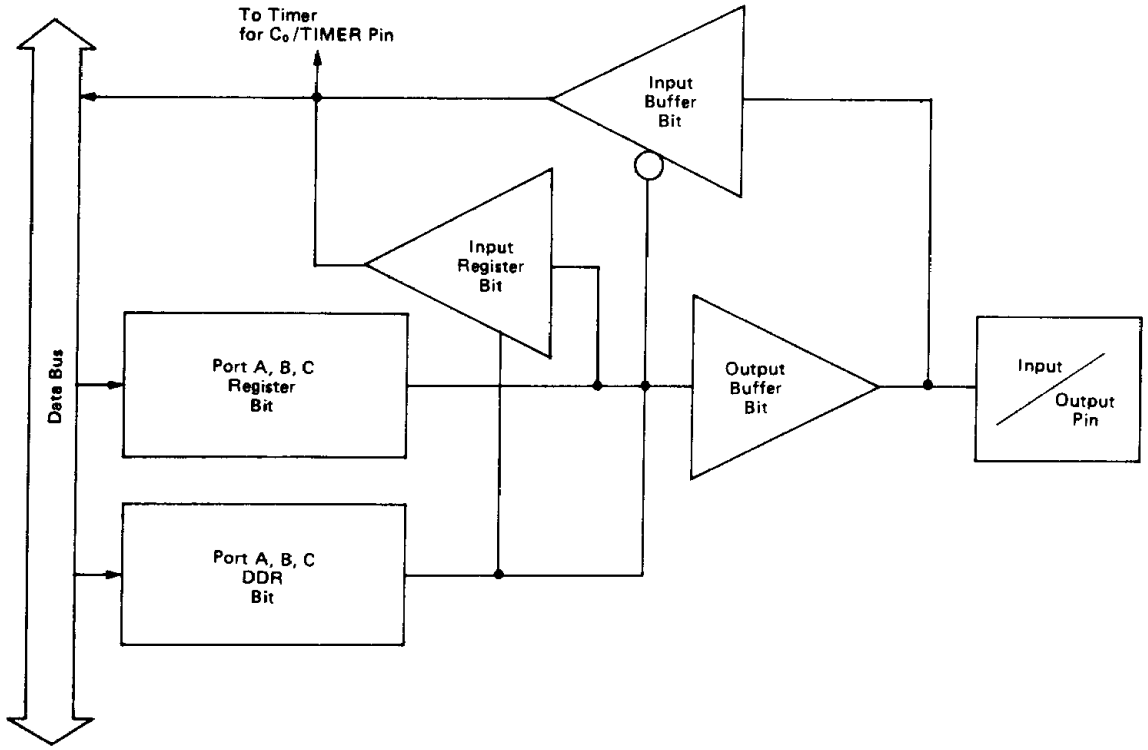


Figure 12 Interrupt Processing Flowchart

■ **INPUT/OUTPUT**

The HD6805T2 is provided with 19 I/O pins ($\overline{INT}$ is also an input pin but is used only as an interrupt). The Data Direction Register (DDR) of each pin defines whether it is an input or an output ("1" is an output; "0" is an input).

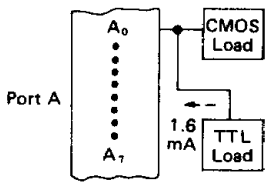


Port A, B, C DDR Bit	1	1	0
Port A, B, C Register Bit	0	1	0/1
Output State	0	1	3-State*
Input to MCU	0	1	Pin

* Except Port A (CMOS drive enable option)

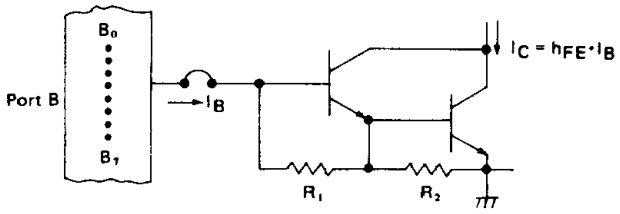
Figure 13 Port A, B, C Block Diagram

The DDR is initialized to all "0"'s by resetting and all pins become input. The output registers, not initialized by resetting, can be set before programming the DDR to prevent undefined level. Fig. 13 shows the latched output data may be read as input data in the case programmed as an output regardless of the logic levels at the output pin due to output loading. If programmed as an output, port B can sink 10 mA (V_{OL} max = 1V) and source 1 mA on each pin.



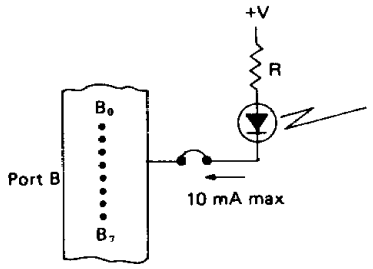
(a)

Port A: Drives CMOS and TTL loads directly when programmed as an output. Internal pull-up devices are included for CMOS drivability. For more details, refer to ELECTRICAL CHARACTERISTICS.



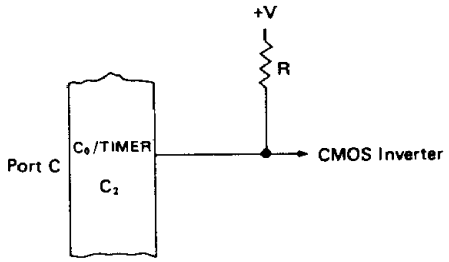
(b)

Port B: Drives Darlington base directly when programmed as an output.



(c)

Port B: Drives LED(s) directly when programmed as an output.



(d)

Port C: Drives CMOS by external pull-up resistors when programmed as an output.

Figure 14 Typical Port Connections

All I/O lines, either for inputs or outputs, are TTL compatible. Both ports B and C as inputs are CMOS compatible; port A as outputs is CMOS compatible by mask option. Some of the port connections are shown in Fig. 14. Fig. 2 shows the data register and the DDR's address.

Caution

The DDR's corresponding to ports A, B and C (at $004, $005 and $006) are provided for write operation only; the read operation is not defined. BSET and BCLR for read/modify/write operations don't set or clear DDRs, but unaffected bits can be set, so it is recommended to use a single-store instruction to write into DDR's.

The latched output data can be written as shown in Fig. 13. When writing to a port, any data can be written even in its DDR "input" mode. This initializes the data registers to prevent any uncertain output. But read/modify/write instructions should be handled carefully because read data depends on the I/O level of the pin with the DDR in the input (0) mode and on the latched output data with the DDR in the output (1) mode.

■ PHASE LOCK LOOP (PLL)

The HD6805T2 has a Phase Lock Loop (PLL) which consists of a 14-bit binary variable divider, a 10-phase reference divider, a frequency and phase comparator which has a three-state output, and the circuits which prevent "backlash" in phase lock states.

Fig. 15 gives an easily established frequency synthesizer system driven by a voltage-controlled oscillator when connecting an adequate high-frequency prescaler and an active integrator. The equations controlling the PLL is shown in Fig. 16.

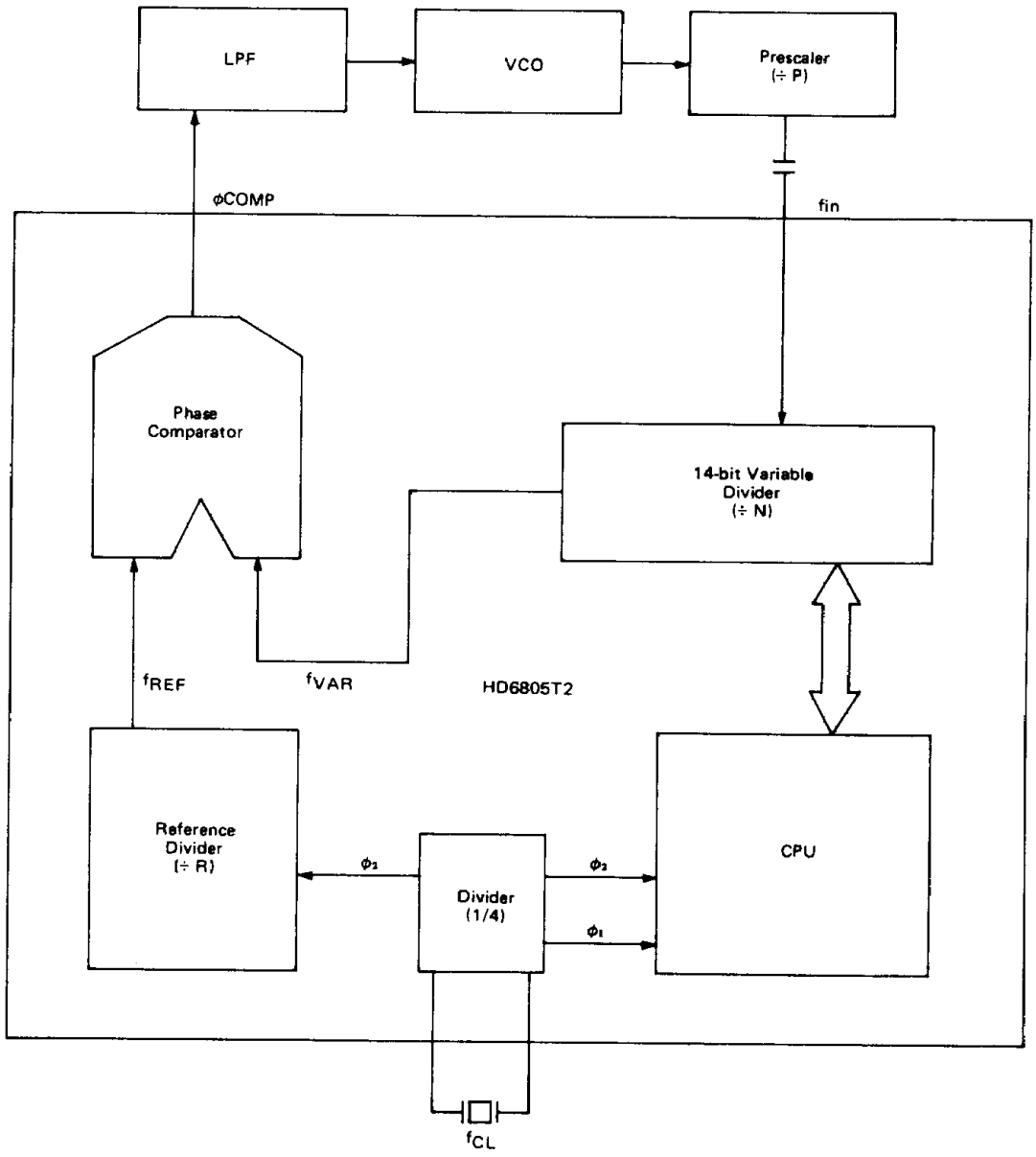


Figure 15 Phase Lock Loop as RF Frequency Synthesizer

For a system in lock: $f_{VAR} = f_{REF}$
since $f_{in} = f_{VAR} \cdot N$, $f_{VCO} = f_{in} \cdot P$ (P = Prescaler diving ratio),
$f_{VCO} = f_{REF} \cdot P \cdot N$

Minimum frequency step = $\frac{\Delta f_{VCO}}{\Delta N} = f_{REF} \cdot P$

e.g.: f_{CL} = 4.00 MHz, R = 2^{10} (R=the reference dividing ratio), P=64

$\frac{\Delta f_{VCO}}{\Delta N}$ = 62.5 kHz, f_{REF} = 976.5 Hz

Figure 16 Principal PLL Equations

■ VARIABLE DIVIDER

The variable devider, a 14-bit binary down counter, sends/receives data to/from the CPU through read/write registers which are located at $00A in the case of the Least Significant (LS) byte and $00B in that of the Most Significant (MS) byte. "1" is always read from the high-order two bits of the $00B register. The variable divider counting zero will generate a preset pulse f_{VAR}. Fig. 17 provides a PLL block diagram where the 14-bit latch is reloaded to the variable divider.

When the $00A register is being written into, data is transferred from $00A and $00B registers to the latch, outside the preset time. Assume that 6-bit data is transferred to $00B register. The data is transferred to the variable divider only when $0A register is next written into. An errorless data transfer in the fine tuning operation is shown in Fig. 18.

The 14-bit latch is activated synchronously with the communication between the CPU and the variable divider, which are asynchronous devices.

When switching on, the PLL registers and the variable divider are fixed to "1".

The variable frequency input pin, f_{in}, is self biased requiring an ac coupled signal with a normal swing of 1.2V. As the input frequency of f_{in} varies with the appropriate prescaler, the whole TV frequency spectrum may be included.

■ REFERENCE DIVIDER

A reference frequency, f_{REF}, is made by the 10-phase binary counter and is compared with the variable divider output. This reference divider is mask optional and the user can select any frequency as shown in Fig. 17.

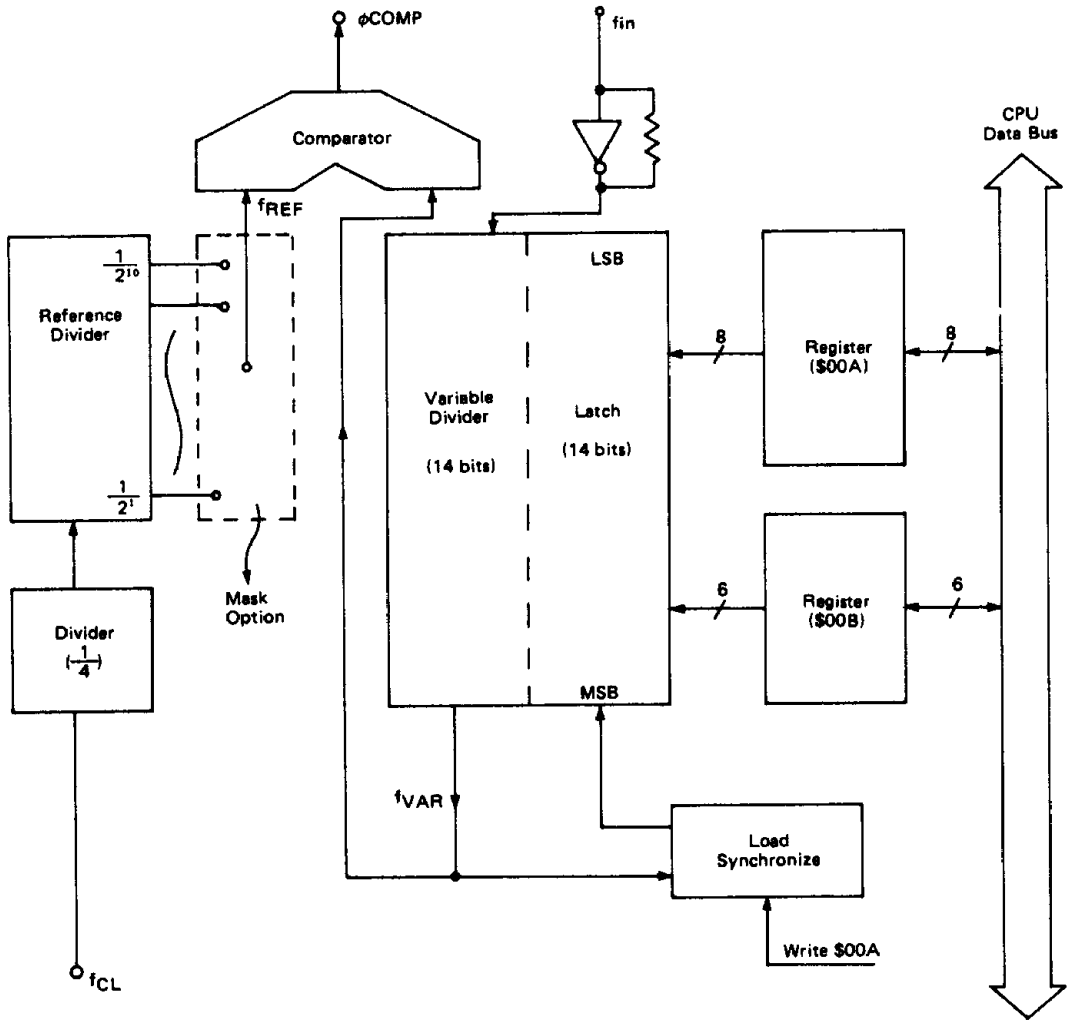


Figure 17 PLL Block Diagram

```
CTD     TST     PLLA    Check if Low byte = "00"
        BNE     CTD1    If not decrement only Low byte
        DEC     PLLB    Decrement High byte
CTD1    DEC     PLLA    Decrement Low byte
                .
                .
                .
CTU     LDA     PLLA
        INCA            Check if Low byte = "FF"
        BNE     CTU1    If not increment only Low byte
        INC     PLLB    Increment High byte
CTU1    INC     PLLA    Increment Low byte
```

Figure 18 Typical Fine Tune Example

■ PHASE COMPARATOR

Both the frequency and phase of f_{VAR} and f_{REF} are compared by the phase comparator, whose relation will generate ϕCOMP, a three-level output, shown in Figs. 19 and 20. ϕCOMP is combined, amplified and the dc voltage is supplied to the voltage control oscillator.

Internal transfer delays will prevent linear features in the stable area. Non-linear characteristics are shown by the phase comparators and this leads to a "backlash" effect which will cause sideband and FM distortion. Insertion of a very short pulse into the device will prevent this. On insertion, the loop tries to cancel the pulse to carry the phase comparator to the linear area shown in Fig. 21.

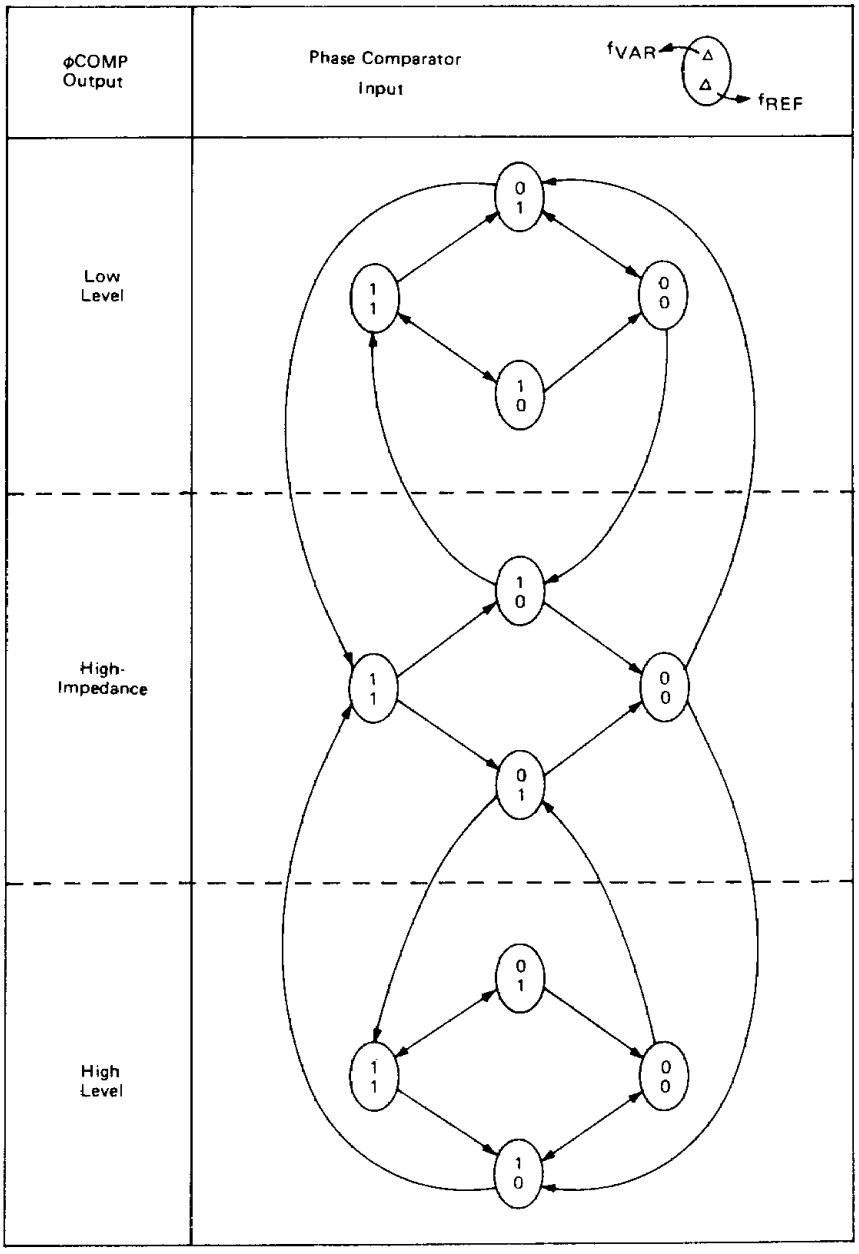


Figure 19 State Diagram (Phase Comparator)

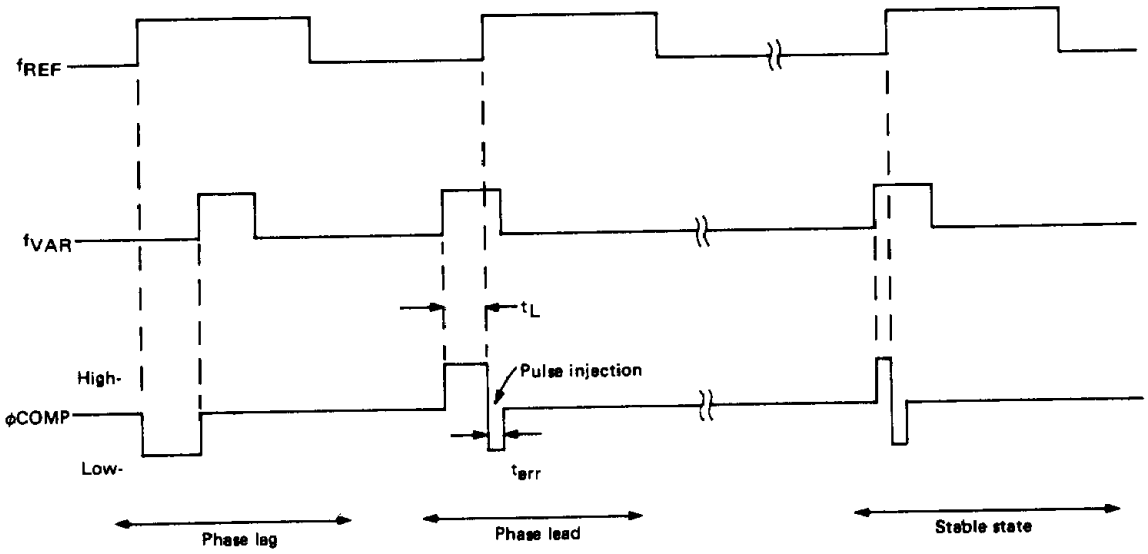


Figure 20 Input/Output Waveform (Phase Comparator)

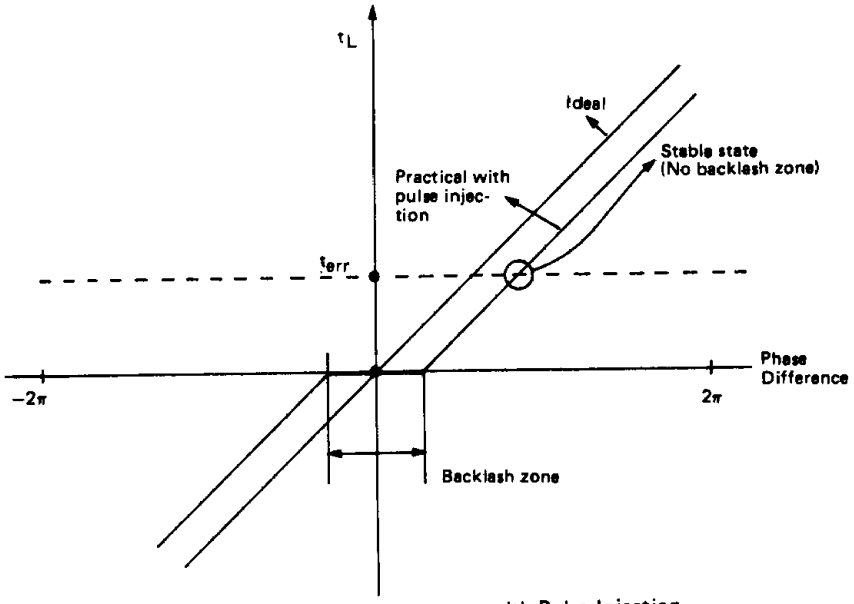


Figure 21 Phase Comparator with Pulse Injection

■ BIT MANIPULATION

With one instruction (BSET, BCLR) the HD6805T2 MCU can set or clear any single bit of the RAM and of the I/O ports (except the Data Direction Registers: – See Caution in the "INPUT/OUTPUT" paragraph). Except for the DDRs, any RAM or I/O bit in page zero including ROM, can be tested with the BRSET and BRCLR instructions, and the program can branch as a result of the state. The carry bit is equal to the value of the bit referenced by BRSET and BRCLR. A rotate instruction is used to pile up serial input data in a RAM location or register. Since the MCU deal with any bit in RAM, ROM and I/O, the user can use the bits in RAM as flags and handle I/O bits as control lines.

See Figure 22. It shows the usefulness of the bit manipulation and test instructions. The program is made up, on the assumption that the MCU is to communicate with an external serial device. The external serial device is provided with a data busy signal, a data input line, and a clock line to clock data one bit at a time, LSB first, out of the MCU.

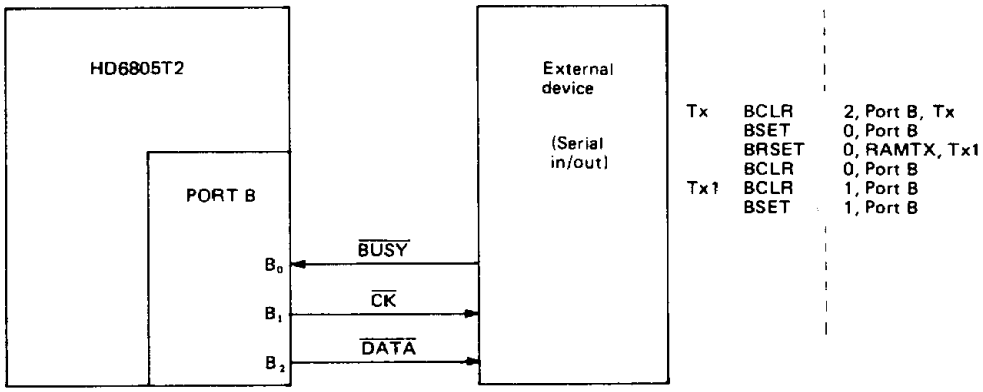


Figure 22 Bit Manipulation Example

■ ADDRESSING MODES

The MCU is provided with ten addressing modes for programming. The following describes them briefly.

• Immediate

Refer to Figure 23. The immediate addressing mode accesses constants which don't change during program execution. Such instructions are two bytes long. The effective address (EA) is the PC and the operand is fetched from the byte following the opcode.

• Direct

Refer to Figure 24. In direct addressing, the address of the operand is contained in the second byte, and the user can directly address the lowest 256 bytes in the memory. All RAM space, I/O registers and 128 bytes of ROM, are located in page zero, to take advantages of this efficient memory addressing mode.

• Extended

Refer to Figure 25. Extended addressing is used to reference any location in the memory space. The EA is the contents of the two bytes following the opcode. Extended addressing instruction are three bytes long.

• Relative

Refer to Figure 26. The relative addressing mode applies only to branch instructions. In this mode the contents of the byte following the opcode is added to the program counter when the branch occurs. EA = (PC) + 2 + Rel. Rel shows the contents of the 7-bit signed bit location following the instruction opcode. If branching doesn't occur, Rel = 0. When a branch takes place, the program goes to somewhere within the range of +129 bytes to 126 of the present instruction. These instructions are two bytes long.

• Indexed (No offset)

Refer to Figure 27. This addressing mode accesses the lowest 256 bytes of the memory. These instruction are one byte long and their EA is the contents of the index register.

• Indexed (8-bit offset)

Refer to Figure 28. The EA is the sum of the contents of the byte following the opcode and the contents of the indexed register. In this mode, 511 low memory locations are accessable. These instructions occupy two bytes.

• Indexed (16-bit offset)

Refer to Figure 29. The EA is the sum of the contents of the two bytes following the opcode and the contents of the index register. Thus, the entire memory space may be accessed. Instructions which use this addressing mode are three bytes long.

• Bit Set/Clear

Refer to Figure 30. This addressing mode applies to instructions which can set or clear any bit on page zero. The lower three bits in the opcode select the bit to be set or cleared. The byte following the opcode specifies the address in page zero.

• Bit Test and Branch

Refer to Figure 31. This addressing mode applies to instructions which can test any bit in the first 256 locations ($00 – $FF) and branch to any location relative to the PC. The byte to be tested is addressed by the byte following the opcode. The individual bit within the byte to be tested is addressed by the low-order three bits of the opcode. The third byte is the relative address to be added to the program counter if the branch condition is met. These instructions are three bytes long. The value of the tested bit is written into the carry bit in the code register.

• Implied

Refer to Figure 32. The implied mode of addressing has not EA. All the information necessary to execute an instruction is included in the opcode. Direct operations on the accumulator and the index register are contained in this addressing mode. In addition, control instructions such as SWI, RTI belong to this group. All implied addressing instructions are only one byte long.

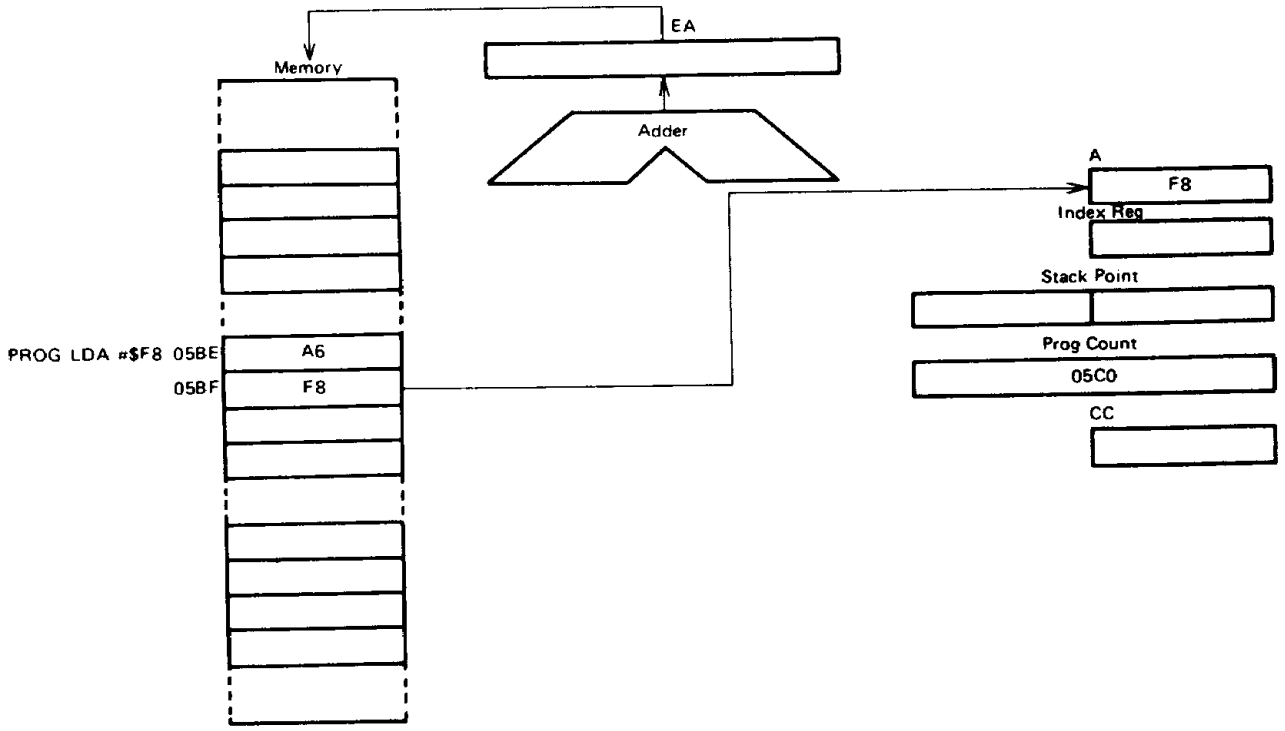


Figure 23 Immediate Addressing Example

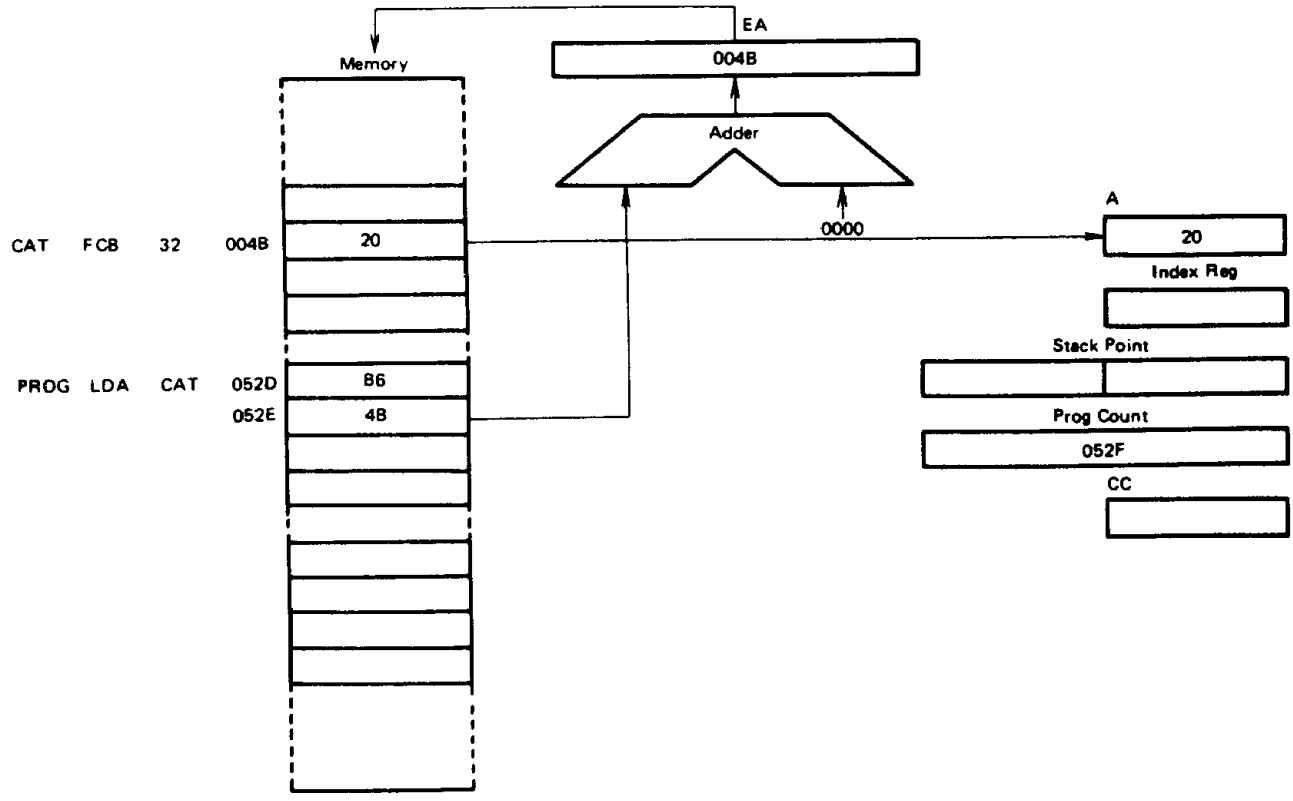


Figure 24 Direct Addressing Example

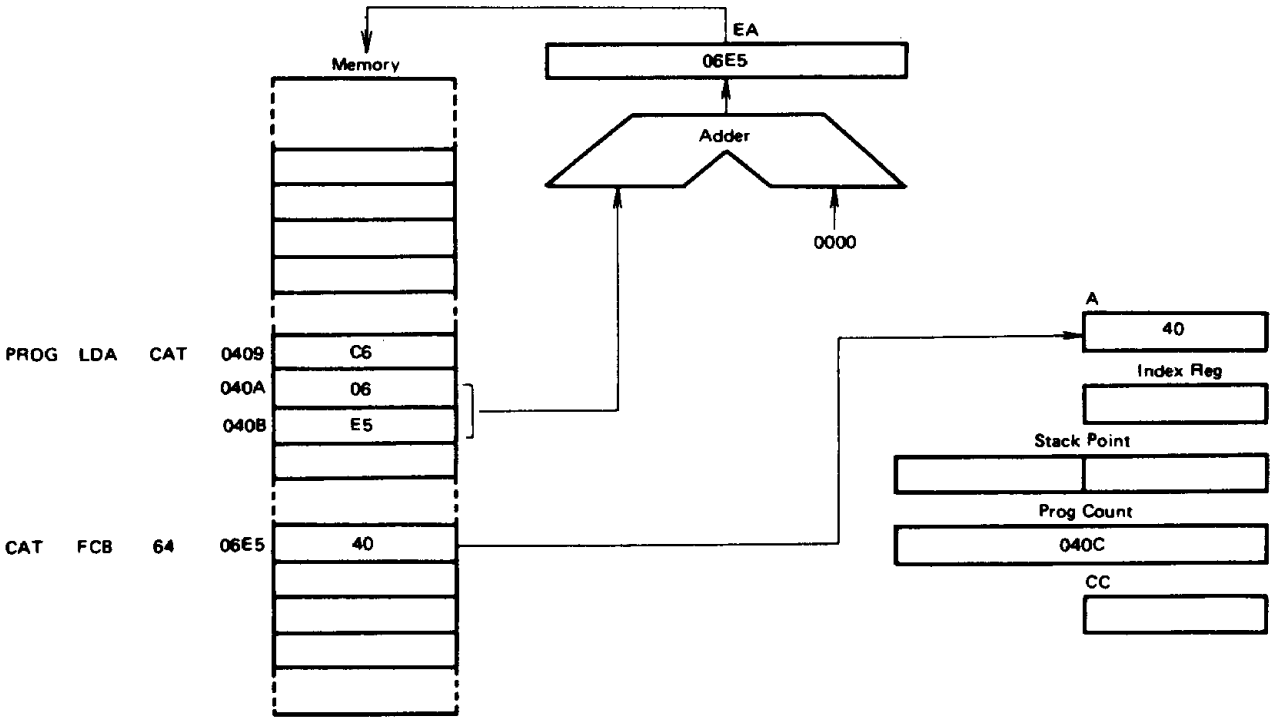


Figure 25 Extended Addressing Example

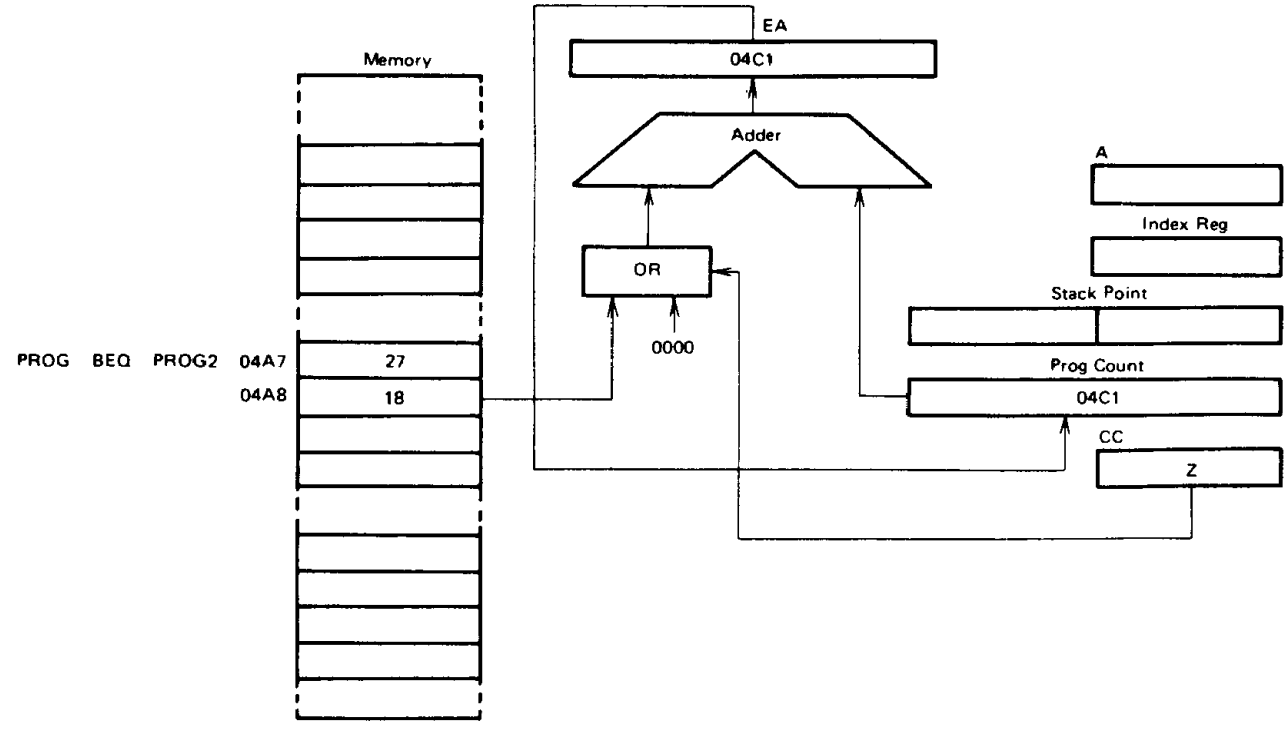


Figure 26 Relative Addressing Example

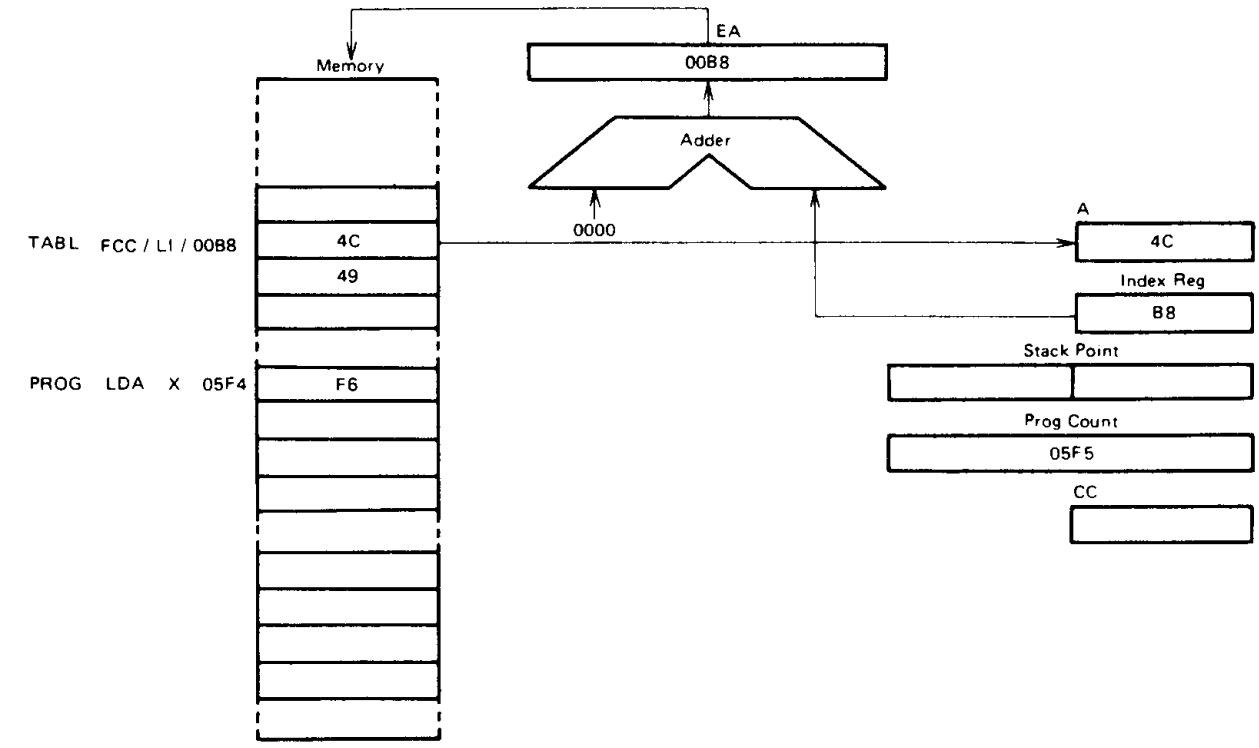


Figure 27 Indexed (No Offset) Addressing Example

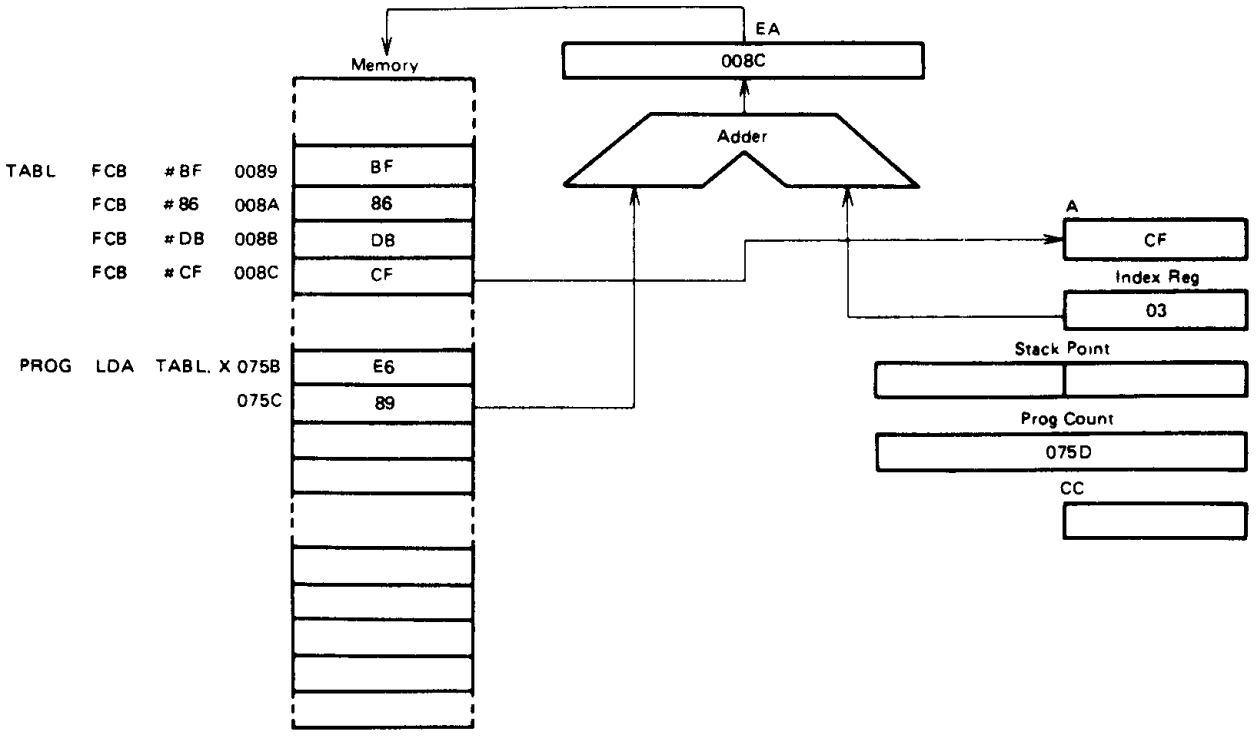


Figure 28 Indexed (8-Bit Offset) Addressing Example

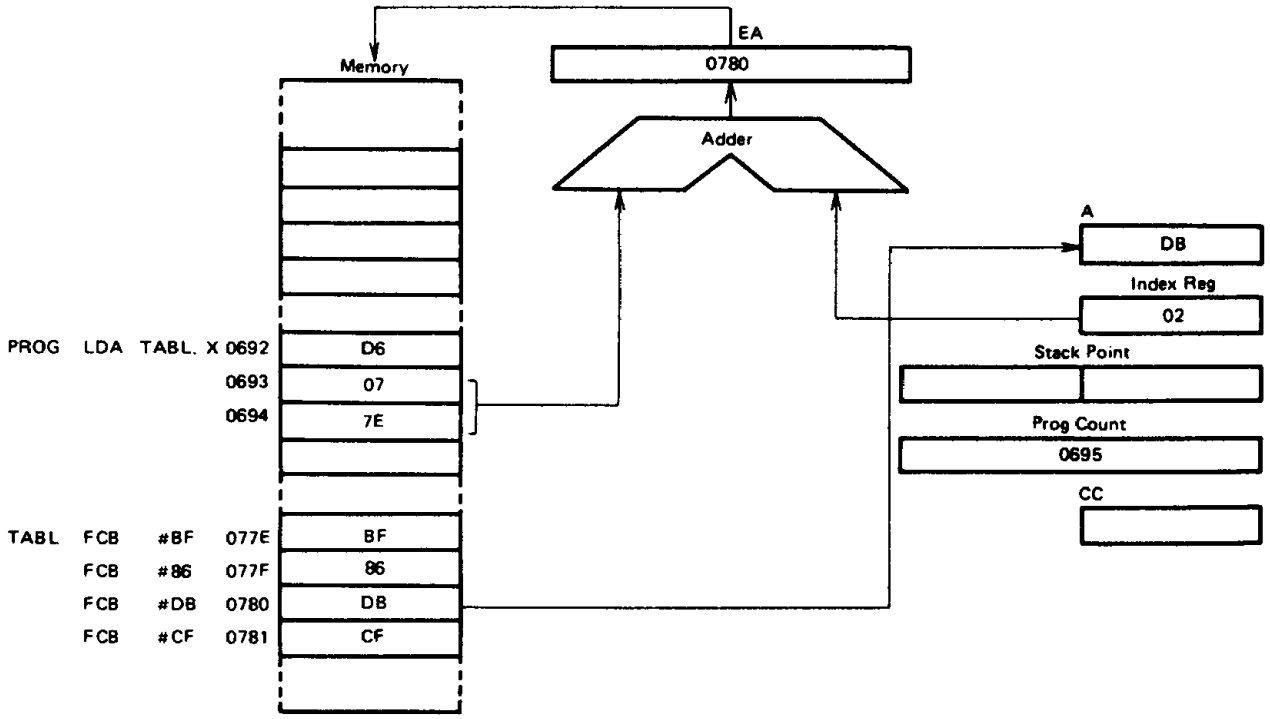


Figure 29 Indexed (16-Bit Offset) Addressing Example

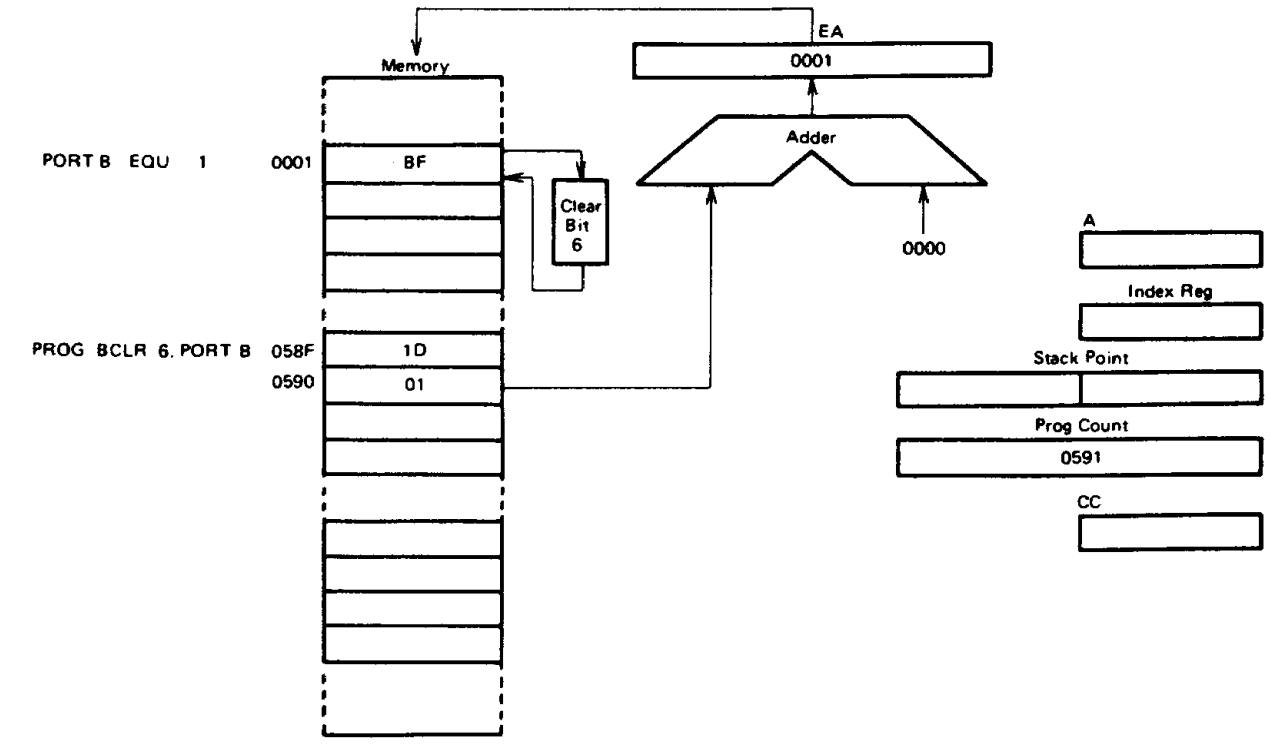


Figure 30 Bit Set/Clear Addressing Example

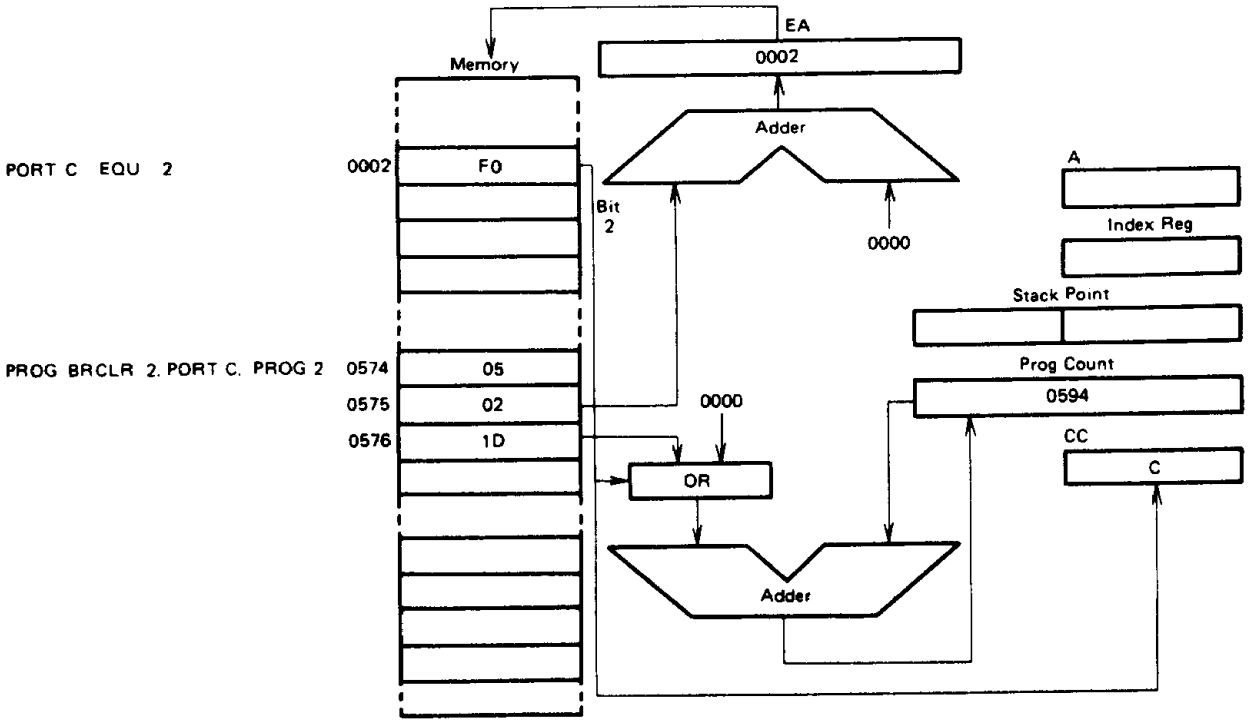


Figure 31 Bit Test and Branch Addressing Example

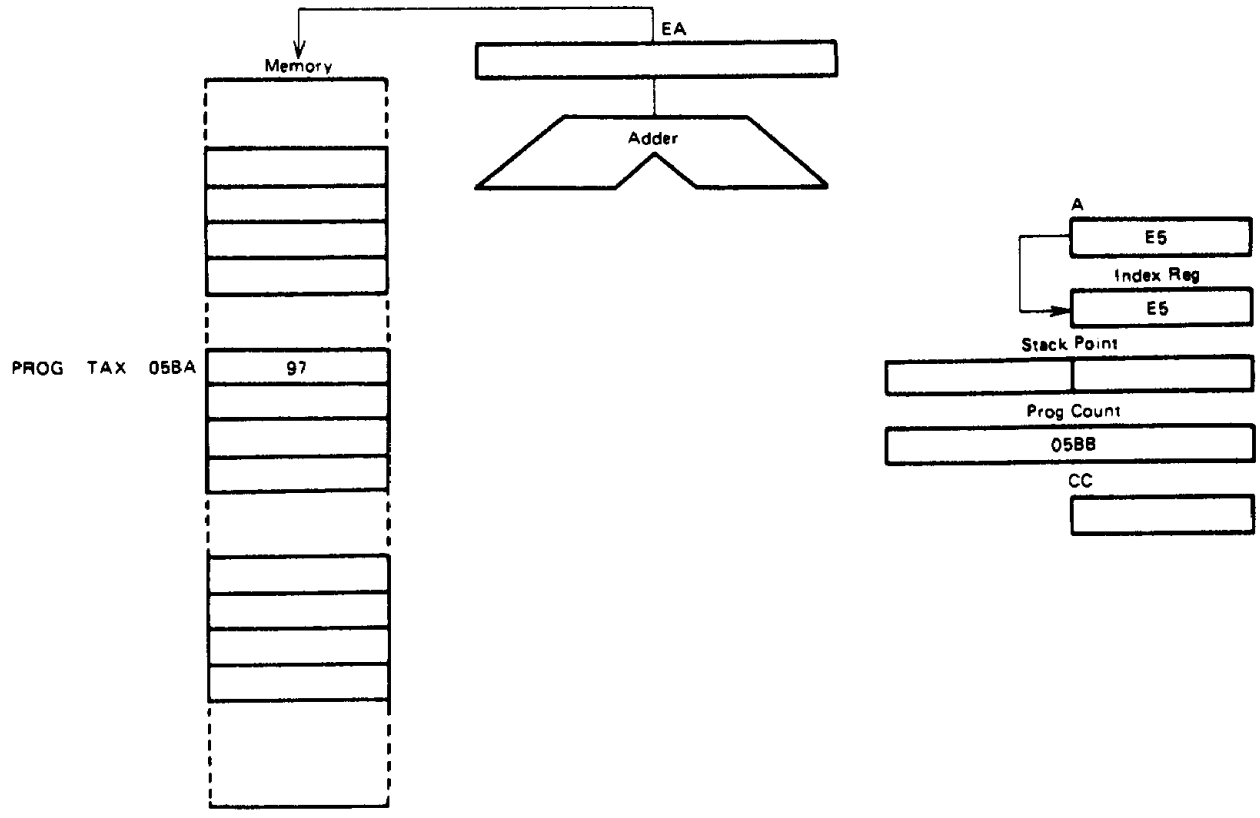


Figure 32 Implied Addressing Example

■ INSTRUCTION SET

There are 59 basic instructions classified into five different types: register/memory, read/modify/write, branch, bit manipulation, and control. The following briefly describes each type. Individual tables present all the instruction in a given type.

- **Register/Memory Instructions**

Refer to Table 2. Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is got from the memory by using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand.

- **Read/Modify/Write Instructions**

Refer to Table 3. These instructions read a memory address or a register, modify or test its contents, and write the modified result back into the memory or to the zero (TST) instruction does not execute "write," it is an exception to these instructions.

- **Branch Instructions**

Refer to Table 4. These instructions cause a branch from the program when a certain condition is met.

- **Bit Manipulation Instructions**

Refer to Table 5. The bit manipulation instructions are applied to any bit in the first 256 bytes of the memory. Some of these instructions set or clear the bits. The others perform the test and branch operations.

- **Control Instructions**

Refer to Table 6. These instructions control the MCU operations during program execution.

- **Alphabetical Listing**

All the instructions above are listed in alphabetical order in Table 7.

- **Opcode Map**

Table 8 is an opcode map for the instructions used on the MCU.

Table 2 Register/Memory Instructions

Function	Mnemonic	Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	–	–	–	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	–	–	–	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	–	–	–	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	–	–	–	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Table 3 Read/Modify/Write Instructions

Function	Mnemonic	Addressing Modes														
		Implied (A)			Implied (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Arithmetic Shift Left	ASL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Table 4 Branch Instructions

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IF Higher	BHI	22	2	4
Branch IF Lower or Same	BLS	23	2	4
Branch IF Carry Clear	BCC	24	2	4
(Branch IF Higher or Same)	(BHS)	24	2	4
Branch IF Carry Set	BCS	25	2	4
(Branch IF Lower)	(BLO)	25	2	4
Branch IF Not Equal	BNE	26	2	4
Branch IF Equal	BEQ	27	2	4
Branch IF Half Carry Clear	BHCC	28	2	4
Branch IF Half Carry Set	BHCS	29	2	4
Branch IF Plus	BPL	2A	2	4
Branch IF Minus	BMI	2B	2	4
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IF Interrupt Line is Low	BIL	2E	2	4
Branch IF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Table 5 Bit Manipulation Instructions

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IF Bit n is set	BRSET n (n=0 ..... 7)	–	–	–	2•n	3	10
Branch IF Bit n is clear	BRCLR n (n=0 .....7)	–	–	–	01+2•n	3	10
Set Bit n	BSET n (n=0 ..... 7)	10+2•n	2	7	–	–	–
Clear bit n	BCLR n (n=0 ..... 7)	11+2•n	2	7	–	–	–

Table 6 Control Instructions

Function	Mnemonic	Implied		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

Table 7 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		○	○	○		○	○	○			^	•	^	^	^
ADD		○	○	○		○	○	○			^	•	^	^	^
AND		○	○	○		○	○	○			•	•	^	^	•
ASL	○		○			○	○				•	•	^	^	^
ASR	○		○			○	○				•	•	^	^	^
BCC					○						•	•	•	•	•
BCLR									○		•	•	•	•	•
BCS					○						•	•	•	•	•
BEQ					○						•	•	•	•	•
BHCC					○						•	•	•	•	•
BHCS					○						•	•	•	•	•
BHI					○						•	•	•	•	•
BHS					○						•	•	•	•	•
BIH					○						•	•	•	•	•
BIL					○						•	•	•	•	•
BIT		○	○	○		○	○	○			•	•	^	^	•
BLO					○						•	•	•	•	•
BLS					○						•	•	•	•	•
BMC					○						•	•	•	•	•
BMI					○						•	•	•	•	•
BMS					○						•	•	•	•	•
BNE					○						•	•	•	•	•
BPL					○						•	•	•	•	•
BRA					○						•	•	•	•	•
BRN					○						•	•	•	•	•
BRCLR										○	•	•	•	•	^
BRSET										○	•	•	•	•	^
BSET									○		•	•	•	•	•
BSR					○						•	•	•	•	•
CLC	○										•	•	•	•	0
CLI	○										•	0	•	•	•
CLR	○		○			○	○				•	•	0	1	•
CMP		○	○	○		○	○	○			•	•	^	^	^
COM	○		○			○	○				•	•	^	^	1
CPX		○	○	○		○	○	○			•	•	^	^	^
DEC	○		○			○	○				•	•	^	^	•
EOR		○	○	○		○	○	○			•	•	^	^	•
INC	○		○			○	○				•	•	^	^	•
JMP			○	○		○	○	○			•	•	•	•	•
JSR			○	○		○	○	○			•	•	•	•	•
LDA		○	○	○		○	○	○			•	•	^	^	•
LDX		○	○	○		○	○	○			•	•	^	^	•

(to be continued)

Condition Code Symbols
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry Borrow
- ^ Test and Set if True, Cleared Otherwise
- • Not Affected

Table 7 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
LSL	○		○			○	○				•	•	^	^	^
LSR	○		○			○	○				•	•	0	^	^
NEG	○		○			○	○				•	•	^	^	^
NOP	○										•	•	•	•	•
ORA		○	○	○		○	○	○			•	•	^	^	•
ROL	○		○			○	○				•	•	^	^	^
ROR	○		○			○	○				•	•	^	^	^
RSP	○										•	•	•	•	•
RTI	○										?	?	?	?	?
RTS	○										•	•	•	•	•
SBC		○	○	○		○	○	○			•	•	^	^	^
SEC	○										•	•	•	•	1
SEI	○										•	1	•	•	•
STA			○	○		○	○	○			•	•	^	^	•
STX			○	○		○	○	○			•	•	^	^	•
SUB		○	○	○		○	○	○			•	•	^	^	^
SWI	○										•	1	•	•	•
TAX	○										•	•	•	•	•
TST	○		○			○	○				•	•	^	^	•
TXA	○										•	•	•	•	•

Condition Code Symbols:
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- ^ Test and Set if True, Cleared Otherwise
- • Not Affected
- ? Load CC Register From Stack

Table 8 Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	–	SUB						0
1	BRCLR0	BCLR0	BRN	–					RTS*	–	CMP						1
2	BRSET1	BSET1	BHI	–					–	–	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	–	CPX						3 L
4	BRSET2	BSET2	BCC	LSR					–	–	AND						4 O
5	BRCLR2	BCLR2	BCS	–					–	–	BIT						5 W
6	BRSET3	BSET3	BNE	ROR					–	–	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					–	TAX	–	STA(+1)					7
8	BRSET4	BSET4	BHCC	LSL/ASL					–	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					–	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					–	CLI	ORA						A
B	BRCLR5	BCLR5	BMI	–					–	SEI	ADD						B
C	BRSET6	BSET6	BMC	INC					–	RSP	–	JMP(-1)					C
D	BRCLR6	BCLR6	BMS	TST					–	NOP	BSR*	JSR(-3)					D
E	BRSET7	BSET7	BIL	–					–	–	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					–	TXA	–	STX(+1)					F
	3/10	2/7	2/4	2/6	1/4	1/4	2/7	1/6	1/*	1/2	2/2	2/4	3/5	3/6	2/5	1/4	

(NOTE) 1. "–" is an undefined operation code.

2. The numbers in the lowermost row represent the number of bytes and cycles required (number of bytes/number of cycles). The number of cycles for the mnemonics asterisked (*) are as follows.

RT1	8	TAX	2
RTS	5	RSP	2
SWI	10	TXA	2
DAA	2	BSR	5
STOP	4	CLI	2
WAIT	4	SEI	2

3. Add the parenthesized numbers to the number or cycle of an instruction.

MASK OPTION LIST

Item	Option	Symbol	Check	Remarks
Timer Clock Source	Internal clock	TIMER I	☐	
	External clock	TIMER E	☐	
Timer Prescaler	Divide by 1	TPR 1	☐	
	Divide by 2	TPR 2	☐	
	Divide by 4	TPR 4	☐	
	Divide by 8	TPR 8	☐	
	Divide by 16	TPR 16	☐	
	Divide by 32	TPR 32	☐	
	Divide by 64	TPR 64	☐	
	Divide by 128	TPR 128	☐	
Reference Divider Ratio	Divide by 2	RDR 2	☐	
	Divide by 4	RDR 4	☐	
	Divide by 8	RDR 8	☐	
	Divide by 16	RDR 16	☐	
	Divide by 32	RDR 32	☐	
	Divide by 64	RDR 64	☐	
	Divide by 128	RDR 128	☐	
	Divide by 256	RDR 256	☐	
	Divide by 512	RDR 512	☐	
	Divide by 1024	RDR 1024	☐	
Low Voltage Inhibit	Disable	LVID	☐	
	Enable	LVIE	☐	
Output Port A	CMOS	CMOS	☐	
	TTL	TTL	☐	