CHAPTER 4
HARDWARE FEATURES

4.1 INTRODUCTION

Each member of the M6805 HMOS/M146805 CMOS Family (except for the MC146805E2) contains, on-chip, nearly all of the support hardware necessary for a complete processor system. The block diagram of Figure 4-1 shows a Central Processing Unit (CPU) which is identical for all members of the family, including the MC146805E2. There is one main difference in various family members and that is the size of the stack pointer and program counter registers. Since the size of these two registers is determined by the amount of device memory, they vary from 11 bits to 13 bits. Each family member contains an on-chip oscillator which provides the processor timing, plus reset, and interrupt logic. Peripheral I/O such as a timer, some bidirectional I/O lines, RAM, and ROM (except for the MC146805E2) are included on-chip in all family members. The peripherals and memory are located in similar locations throughout the family; therefore, once the user is familiar with any family device, he is familiar with all. In addition, new devices can be incorporated in the family by adding to and/or subtracting from the peripheral blocks associated with the CPU. These peripheral blocks could include additional I/O lines, more RAM, EPROM, A/D converter, phase-lock-loop, or an external bus. The choice of using inexpensive HMOS or low-power, static CMOS is also available.

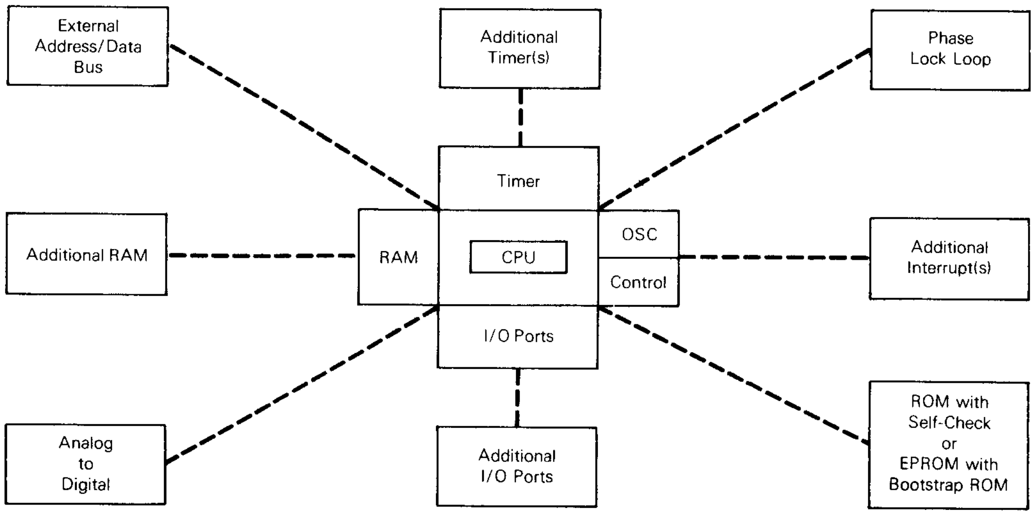


Figure 4-1. M6805 HMOS/M146805 CMOS Family Block Diagram

The M6805 HMOS/M146805 CMOS Family of MCU/MPU devices are implemented using a single address map, memory mapped I/O, and Von Neumann architecture. Peripheral I/O devices (A/D, timer, PLL, etc.) are accessed by the CPU via the peripheral control and/or data registers which are located in the address map. Data is transferred to the peripheral I/O devices with the same instructions that are used to access memory. The key to using the M6805 HMOS/M146805 CMOS Family I/O features is in learning how the peripheral registers effect the device operation. Since a second address map is not used, there is no need for the system designer to learn a second set of specialized I/O instructions.

4.2 PROCESSING TECHNOLOGY

As stated above, system designers have the option of using either HMOS (M6805) or CMOS (M146805) technology. Since each technology has its advantages, there are applications which will favor one over the other. Table 4-1 provides a comparison of representative features between HMOS and CMOS.

Table 4-1. Comparison of Features Between HMOS and CMOS

HMOS	CMOS
Inexpensive Due to Smaller Die Size	Low Power Consumption
Fast	Silicon-Gate Devices are as Fast as HMOS Devices Completely Static Operation Wider Voltage Range (3-6 V) Increased Noise Immunity
Consumes Ten Times More Power than CMOS	More Expensive Since CMOS Cell is Larger
Dynamic Operation (Requires Continuous Clock) Limited Voltage Range	Sensitive to SCR Latchup

4.3 TEMPORARY STORAGE (RAM)

Random Access Memory (RAM) is used as temporary storage by the CPU. The RAM is temporary in that it is volatile and its contents are lost if power is removed. However, since RAM can be read from or written to, it is best used for storing variables. All on-chip RAM is contained in the first 128 memory locations and the top of RAM is presently used by the processor as a program control stack. The stack is used to store return addresses for subroutine calls and the machine state for interrupts. The stack pointer register is used to keep track of (point to) the next free stack address location. The stack operates in a LIFO (last-in-first-out) mode so that operations may be nested. The actual stack size varies between the different family members; however, in all cases, exceeding the stack limit should be avoided. If the stack limit is exceeded, the stack pointer wraps around to the top of the stack ($7F) and more than likely stack data is lost. Each interrupt requires five bytes of stack space and each subroutine requires two bytes. If, at worst case, a program requires five levels of subroutine nesting and one level of interrupt, then 15 bytes of stack space should be reserved. Any unreserved stack RAM may be used for other purposes.

Low-power standby RAM for HMOS is available on the MC6805P4. Although the processor is dynamic, the RAM is static and may be powered from a separate standby supply voltage which does not power any other part of the device, thus, lowering standby supply current requirements. The amount of standby RAM implemented is a mask option and is determined by the minimum necessary for the particular application.

4.4 PERMANENT STORAGE (ROM OR EPROM)

All M6805 HMOS/M146805 CMOS Family devices, except the MC146805E2, contain some form of permanent, non-volatile memory. It may be either mask programmed ROM or UV-light erasable EPROM; however, the M68705 HMOS EPROM versions contain EPROM as the main storage and a small mask ROM which is used to store the bootstrap programming routines. Non-volatile memory is generally used to store the user programs as well as tables and constants. The mask ROM versions are the most economical for large quantities while the EPROM versions are best suited for limited quantities used for production or prototyping. Currently three EPROM versions exist. Each has slightly more storage and versatility than the current mask ROM versions; however, the EPROM versions can emulate the functions of more than one of the current mask ROM versions and could be used for future mask ROM versions.

4.5 OSCILLATOR

This on-chip oscillator contained on every M6805 HMOS/M146805 CMOS Family device essentially generates the timing used by the device. The oscillator can be used in a number of different modes as shown in Figure 4-2. Each mode has its advantages and the basic trade-off is between economy and accuracy.

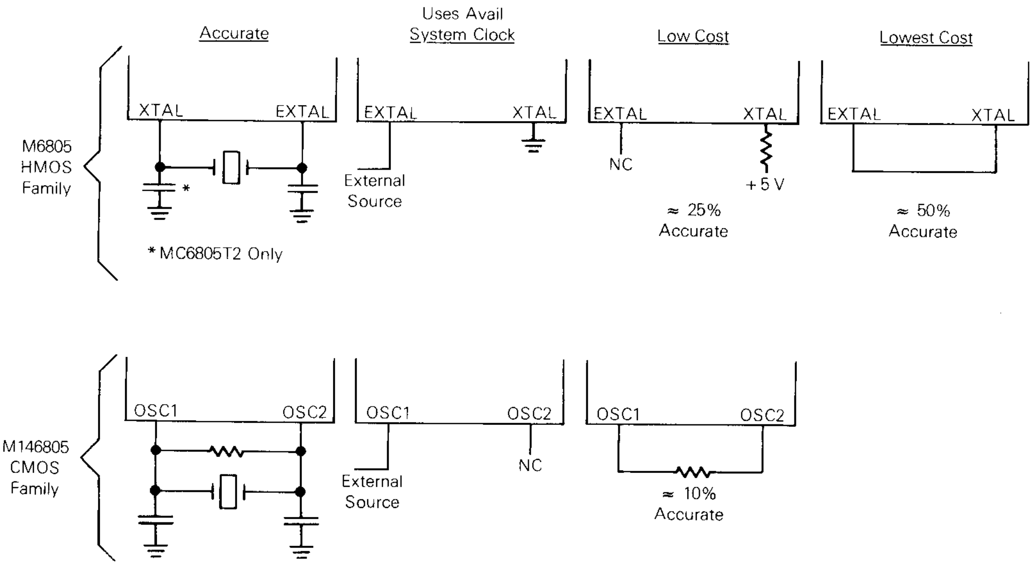


Figure 4-2. M6805 HMOS/M146805 CMOS Family Oscillator Modes

Except for the EPROM members of M6805 HMOS Family, a manufacturing mask option is required to select either the crystal oscillator or the resistor oscillator circuit. The oscillator frequency is internally divided by four to produce the internal system clocks. The EPROM devices of the M6805 HMOS Family utilize the mask option register (MOR) to select the crystal or resistor oscillator circuit.

The M146805 CMOS Family devices also use a manufacturing mask option to select either the crystal or resistor circuit. However, a second manufacturing mask option provides either a divide-by-two or divide-by-four circuit to produce the internal system clock. The EPROM devices of the M146805 CMOS Family also utilize the mask option register (MOR) to select the crystal or resistor oscillator circuit.

4.6 RESETS

The M6805 HMOS/M146805 CMOS Family processor can be reset in two ways: either by the initial power-up or by the external reset input pin ($\overline{\text{RESET}}$). Additionally, a low voltage inhibit (LVI) circuit is included on some HMOS masked ROM versions to force a reset if V_{CC} falls to V_{LVI}. Any of the reset methods allow an orderly start-up; additionally, the $\overline{\text{RESET}}$ input can be used to exit the CMOS STOP and WAIT modes of program execution. Both the LVI and external $\overline{\text{RESET}}$ inputs allow the processor to recover from otherwise catastrophic errors. External reset ($\overline{\text{RESET}}$) is implemented with a Schmitt trigger input for improved noise immunity. Figure 4-3 illustrates the required timing and logic levels for devices implemented with LVI. All M6805 HMOS Family members have the equivalent of an internal pullup resistor as shown in Figure 4-4 so that the $\overline{\text{RESET}}$ pin will reflect the drop in V_{CC}.

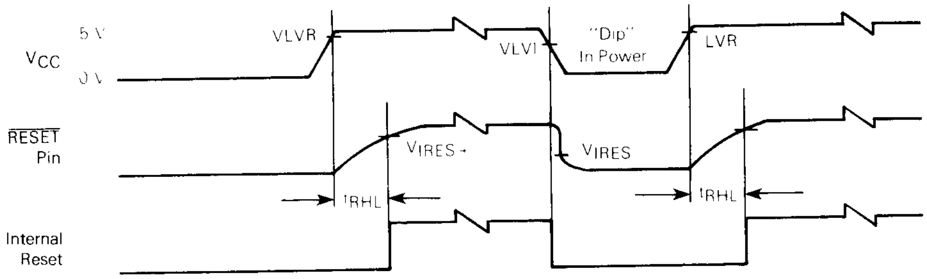


Figure 4-3. Power and Reset Timing

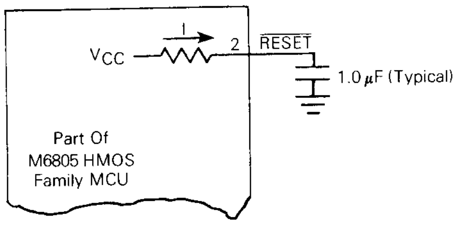


Figure 4-4. Power up Reset Delay Circuit

HMOS power-on reset circuitry includes the equivalent of an internal pullup resistor, so that only a capacitor is required externally (see Figure 4-4). The power-on reset occurs when a positive transition is detected on V_{CC}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. For CMOS devices, the power-on circuitry provides for a 1920 t_{cyc} delay from the time of the first oscillator operation. If the external $\overline{\text{RESET}}$ pin is low at the end of the 1920 t_{cyc} time out, the processor remains in the reset condition until the $\overline{\text{RESET}}$ pin goes high.

Any reset causes the following to occur:

1. All interrupt requests are cleared to "0".
2. All interrupt masks are set to "1".
3. All data direction registers are cleared to "0" (input).
4. The stack pointer is reset to $7F (top of stack).
5. The STOP and WAIT latches (M146805 CMOS only) are reset.
6. The reset vector is fetched and placed in the program counter. (The reset vector contains the address of the reset routine.)

4.7 INTERRUPTS

4.7.1 General

The M6805 HMOS/M146805 CMOS Family program execution may be interrupted in the following ways:

1. Externally via the $\overline{\text{IRQ}}$ (CMOS) or $\overline{\text{INT}}$ (HMOS) pins. Additionally, some M6805 HMOS members include a second external interrupt ($\overline{\text{INT2}}$). External interrupts are maskable.
2. Internally with the on-chip timer. The timer interrupt is maskable.
3. Internally by executing the software interrupt instruction (SWI). The SWI is non-maskable.

When an external or timer interrupt occurs, the interrupt is not immediately serviced until the current instruction being executed is completed. Until the completion of the current instruction, the interrupt is considered pending. After the current instruction execution is completed, unmasked interrupts may be serviced. If both an external and a timer interrupt are pending, the external interrupt is serviced first; however, the timer interrupt request remains pending unless it is cleared during the external interrupt service routine. The software interrupt is executed in much the same manner as any other instruction. The external interrupt pin ($\overline{\text{IRQ}}$ or $\overline{\text{INT}}$) may be tested with the BIL or BIH conditional branch instructions. These instructions may be used to allow the external interrupt pins (except $\overline{\text{INT2}}$) to be used as an additional input pin regardless of the state of the interrupt mask in the condition code register.

4.7.2 Timer Interrupt

If the timer mask bit (TCR6) is cleared, then each time the timer decrements to zero (transitions from \$01 to \$00) an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I bit in the condition code register is set, masking further interrupts until the present one is serviced. The contents of the timer interrupt vector, containing the location of the timer interrupt service routine, is then loaded into the program counter.

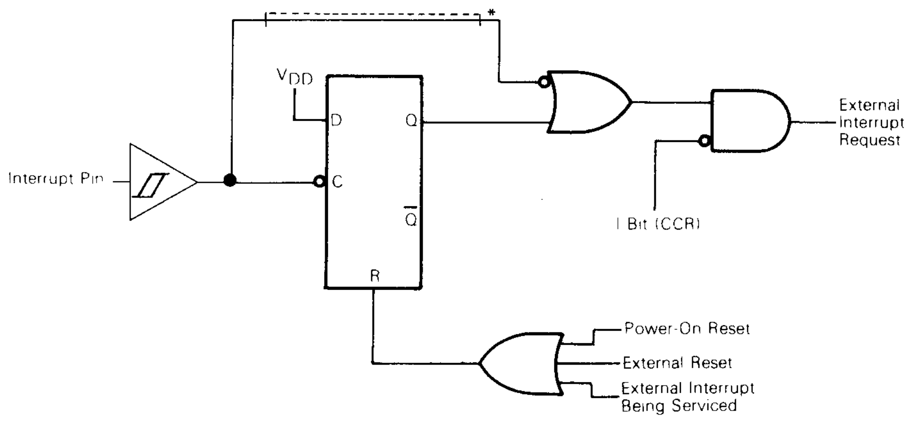
If the CMOS WAIT mode is enabled (M146805 CMOS Family only), the timer may be used to exit the low-power mode and the timer WAIT vector is used instead of the normal timer interrupt vector. Software must be used to clear the timer interrupt request bit (TCR7). At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program. Note that if an external hardware interrupt is used to exit the WAIT mode, the timer interrupt will vector to the normal timer vector instead of the timer WAIT vector.

4.7.3 External Interrupts

All external interrupts are maskable. If the interrupt mask bit (I bit) of the condition code register is set, all interrupts are disabled. Clearing the I bit enables the external interrupts. Additionally, $\overline{\text{INT2}}$ requires that bit 6 of the miscellaneous register also be cleared. The external interrupts recognize both level- and edge-sensitive trigger interrupts for the M146805 CMOS Family as shown in Figure 4-5. The M6805 HMOS Family requires negative edge-sensitive trigger interrupts only. The level-sensitive line is mask optional on the MC146805G2 and MC146805F2 (see Figure 4-5). The level-sensitive triggered interrupts are generally used for multiple "wire-ORed" interrupt sources as shown in Figure 4-5b. Edge-sensitive interrupts may be used for periodic interrupts; however, since the interrupt request is latched by the processor, interrupt sources may return to other tasks. Periodic interrupt requests require that the interrupt request line be held low for at least one t_{cyc} and not be repeated until the end of the service routine and the stacking operations are complete. This ensures that all requests are recognized. The interrupt line must also be released high to allow the interrupt latch to be reset.

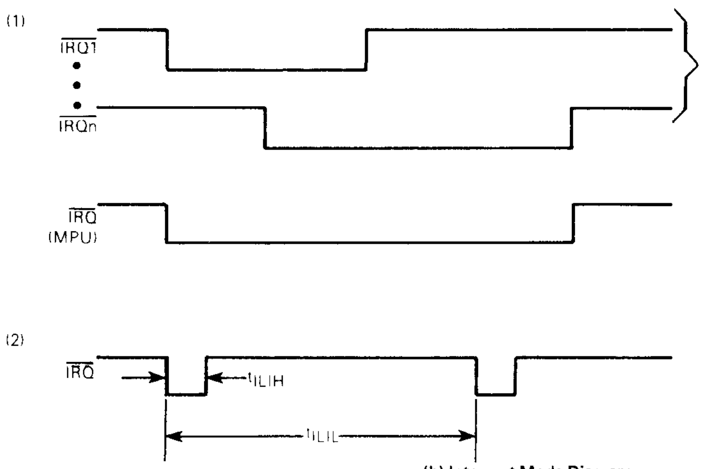
Upon servicing a pending interrupt request, the processor executes the following sequences:

1. Mask all interrupts (set I bit).
2. Stack all CPU registers.
3. Load the program counter with the appropriate vector location contents ($\overline{\text{INT2}}$ uses the same vector location as does the timer).
4. Execute service routine.



* Mask optional for M146805 CMOS Family

(a) Interrupt Functional Diagram



Edge- or Level-Sensitive Wire ORed Condition

If after servicing an interrupt the $\overline{\text{IRQ}}$ remains low, then the next interrupt is recognized.

Edge-Sensitive Pulse Condition

The minimum pulse width (t_{ILIH}) is one t_{cyc}. The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles.

(b) Interrupt Mode Diagram

Figure 4-5. External Interrupt

4.7.4 Software Interrupt (SWI)

The software interrupt is executed the same as any other instructon and as such will take precedence over hardware interrupts only if the I bit is set (interrupts masked). The SWI instruction is executed similar to the hardware interrupts in that the I bit is set, CPU registers are stacked, etc. The SWI is executed regardless of the state of the interrupt mask in the condition code register; however, when the I bit is clear and an external or internal hardware interrupt is pending, the SWI instruction (or any other instruction) is not fetched until after the hardware interrupts have been serviced. The SWI uses its own unique vector location.

4.8 I/O PORTS

At least 16 individually programmable, bidirectional I/O lines are included on each member of the M6805 HMOS/M146805 CMOS Family; however, more than this exists on most family members. Each line is individually programmable as either an input or an output via its corresponding data direction register (DDR) bit as shown in Figure 4-6. Table 4-2 provides a description of the effects of port data register operation. Data is written into the port output data latch regardless of the state of the DDR; therefore, initial output data should be written to the output data latch before programming the DDR. After a port line has been configured as an output, the data on that line reflects the corresponding bit of the output data latch. A read of the port data register reflects the last value written to the port output data latch for output lines and the current status of the input pins. Note that the DDRs in the M6805 HMOS Family are write-only registers and should not be used with any of the read-modify-write (RMW) instructions such as the bit manipulation instructions. The M146805 CMOS Family DDRs are read/write registers and may be used with RMW instructions.

Some devices include a number of input-only lines. These lines have no DDR and have read-only data registers.

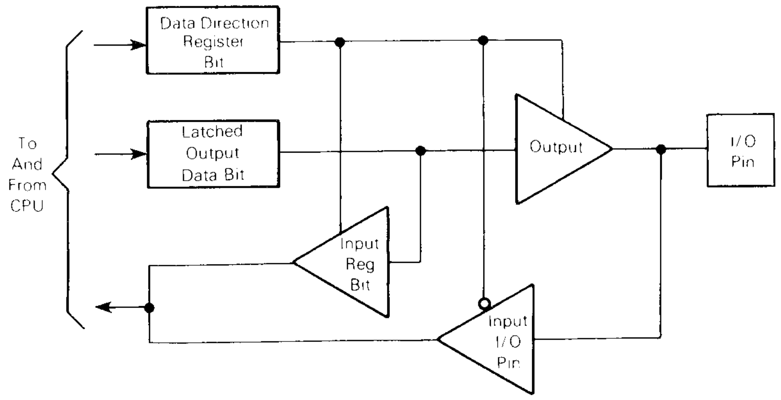


Figure 4-6. Typical Port I/O Circuitry

Table 4-2. Port Data Register Accesses

R/$\overline{W}$	DDR Bit	Results
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

R/$\overline{W}$ is an internal line.

4.9 TIMER DESCRIPTION

4.9.1 General

All M6805 HMOS/M146805 CMOS Family devices contain at least one timer on chip. The timer is basically composed of a 7-bit prescaler, an 8-bit counter, and interrupt logic. The M6805 HMOS and M146805 CMOS devices differ slightly in two areas. First, the input to the timer, as shown in Figures 4-7 and 4-8, is programmed differently. In the M146805 CMOS Family, the input is selected by programming bits 4 and 5 of the timer control register (TCR). In the M6805 HMOS Family they are mask programmable (except for the MC6805R3 and MC6805U3). The second difference is the prescaler which is software programmable in the M146805 CMOS Family and mask programmable in the M6805 HMOS Family.

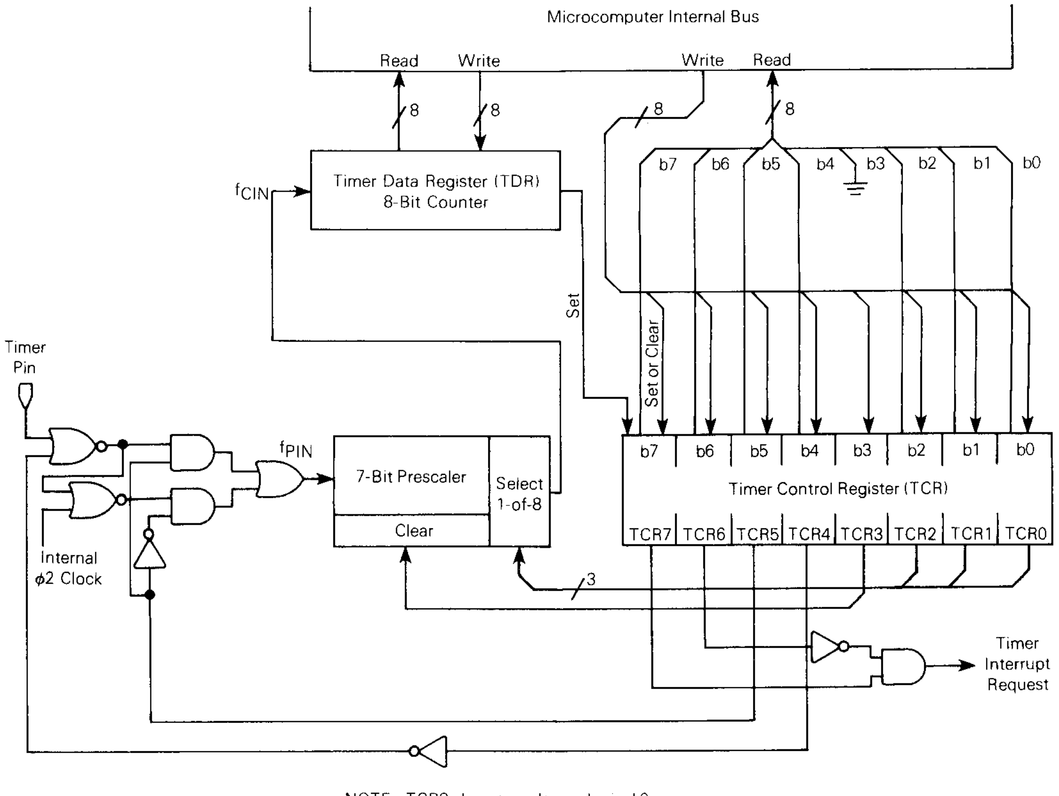
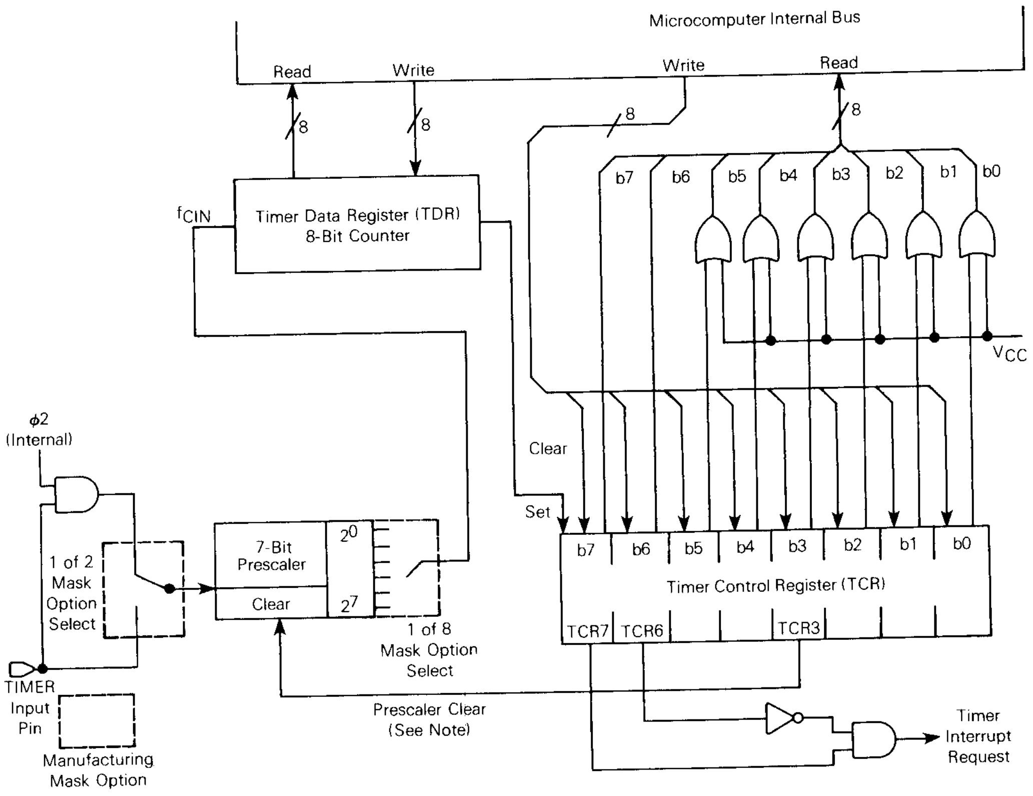


Figure 4-7. M146805 CMOS Family Timer Block Diagram



Microcomputer Internal Bus
Read
Write
Write
Read
8
8
8
8
Timer Data Register (TDR) 8-Bit Counter
f_{CIN}
b7
b6
b5
b4
b3
b2
b1
b0
V_{CC}
$\phi 2$ (Internal)
Clear
Set
1 of 2 Mask Option Select
7-Bit Prescaler
Clear
2^0
2^7
1 of 8 Mask Option Select
Timer Control Register (TCR)
TCR7
TCR6
TCR3
TIMER Input Pin
Manufacturing Mask Option
Prescaler Clear (See Note)
Timer Interrupt Request

NOTE: The TCR3 prescaler clear bit is not available in the MC6805P2, MC6805P4, and MC6805T2; however, it is used as shown in all other M6805 HMOS Family MCUs. The TCR3 bit always reads as a logical 0.

Figure 4-8. M6805 HMOS Family Timer Block Diagram

The timer interrupt operates similarly to the external interrupts; however, users must clear the interrupt request bit (TCR7) to prevent a second timer interrrupt service from occurring.

Descriptions of the HMOS and CMOS timers follow in more detail. The EPROM versions allow either CMOS or HMOS timer operations via the programmable mask option register (MOR).

4.9.2 M146805 CMOS Family Timer

4.9.2.1 GENERAL. The MCU timer contains an 8-bit software programmable counter with 7-bit software selectable prescaler as shown in Figure 4-7. The counter may be preloaded under program control and decrements toward zero. When the counter decrements to zero, the timer interrupt request bit, i.e., bit 7 of the timer control register (TCR), is set. Then, if the timer interrupt is not masked, i.e., bit 6 of the TCR and the I bit in the condition code register are both cleared, the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack, and then fetches the timer vector address in order to begin servicing.

The counter continues to count after it reaches zero, allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without disturbing the count. The contents of the counter become stable prior to the read portion of a cycle and do not change during the read. The timer interrupt request bit remains set until cleared by the software. If a clear (write TCR7 = 0) occurs before the timer interrupt is serviced, the interrupt is lost. The TCR7 bit may also be used as a scanned status bit in a non-interrupt mode of operation (TCR6 = 1).

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output which is used as the counter input. The processor cannot write into or read from the prescaler; however, its contents are cleared to all "0s" by the write operation into TCR when bit 3 of the written data equals 1. This allows for truncation-free counting.

The timer input can be configured in one of three different operating modes, plus a disable mode, depending on the value written to the TCR4 and TCR5 control bits. Refer to the Timer Control Register paragraph.

4.9.2.2 TIMER INPUT MODE 1. If TCR4 and TCR5 are both programmed to a "0", the input to the timer is from an internal clock and the TIMER input pin is disabled. The internal clock mode can be used for periodic interrupt generation, as well as a reference in frequency and event measurement. The internal clock is the instruction cycle clock. During a WAIT instruction, the internal clock to the timer continues to run at its normal rate.

4.9.2.3 TIMER INPUT MODE 2. With TCR4 = 1 and TCR5 = 0, the internal clock and the TIMER input pin are ANDed to form the timer input signal. This mode can be used to measure external pulse widths. The external pulse simply turns on the internal clock for the duration of the pulse. The resolution of the measurement in this mode is ±1 clock.

4.9.2.4 TIMER INPUT MODE 3. If TCR4 = 0 and TCR5 = 1, then all inputs to the timer are disabled.

4.9.2.5 TIMER INPUT MODE 4. If TCR4 = 1 and TCR5 = 1, the internal clock input to the timer is disabled and the TIMER input pin becomes the input to the timer. In this mode, the timer can be used to count external events as well as external frequencies for generating periodic interrupts. The counter is clocked by the falling edge of the external signal.

Figure 4-7 shows a block diagram of the timer subsystem. Power-on reset and the STOP instruction cause the counter to be set to $F0.

4.9.2.6 TIMER CONTROL REGISTER (TCR). The eight bits in the TCR are used to control various functions such as configuring the operation mode, setting the division ratio of the prescaler, and generating the timer interrupt request signal. A description of each TCR bit function is provided below. All bits in this register except bit 3 are read/write bits.

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

TCR7 — Timer interrupt request bit: bit used to indicate the timer interrupt when it is logic "1".
1 — Set whenever the counter decrements to zero, or under program control.
0 — Cleared on external reset, power-on reset, STOP instruction, or program control.

TCR6 — Timer interrupt mask bit: when this bit is a logic "1" it inhibits the timer interrupt to the processor.
1 — Set on external reset, power-on reset, STOP instruction, or program control.
0 — Cleared under program control.

TCR5 — External or internal bit: selects the input clock source to be either the external TIMER pin or the internal clock. (Unaffected by reset.)
1 — Select external clock source.
0 — Select internal clock source.

TCR4 — External enable bit: control bit used to enable the external timer pin (Unaffected by reset.)
1 — Enable external timer pin.
0 — Disable external timer pin.

Summary of Timer Clock Source Options

TCR5	TCR4	Option
0	0	Internal Clock to Timer
0	1	AND of Internal Clock and TIMER Pin to Timer
1	0	Inputs to Timer Disabled
1	1	TIMER Pin to Timer

Refer to Figure 4-7 for logic representation.

TCR3 — Timer prescaler Reset bit: writing a "1" to this bit resets the prescaler to zero. A read of this location always indicates "0". (Unaffected by reset.)

TCR2, TCR1, TCR0 — Prescaler select bits: decoded to select one of eight taps on the prescaler. (Unaffected by reset.)

Prescaler

TCR2	TCR1	TCR0	Result
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8

TCR2	TCR1	TCR0	Result
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

4.9.3 M6805 HMOS Family Timer

The timer block diagram for these family members is shown in Figure 4-8. This timer consists of an 8-bit software programmable counter (timer data register, TDR) which is decremented towards zero by a clock input from a prescaler. The prescaler clock input is received either from the TIMER pin via an external source or from the internal $\phi 2$ of the MCU. The actual clock input to the prescaler is determined by a mask option when the MCU is manufactured.

The mask option allows the prescaler to be triggered either directly from the external TIMER pin or from a gated $\phi 2$ internal clock. When $\phi 2$ signal is used as a clock source, it is only applied whenever the TIMER pin is a logical high. This allows the user to perform a pulse width measurement of the TIMER pin input pulse. In order to provide a continuous $\phi 2$ input to the prescaler in this configuration, it is only necessary to connect the TIMER pin to V_{CC}.

The prescaler divide ratio is selected by a mask option which is determined when the MCU is manufactured. This option allows the TDR to be triggered by every clock input to the prescaler (2^0), by the 128th clock input to the prescaler (2^7), or by any other power of two in between.

The TDR (8-bit counter) may be loaded under program control and is decremented towards zero by each output from the prescaler. Once the TDR has decremented to zero, it sets bit 7 of the timer control register (TCR) to generate a timer interrupt request. Bit 6 of the TCR can be software set to inhibit the timer interrupt request, or software cleared to pass the interrupt request to the processor, provided the I bit is cleared. Since the 8-bit counter (TDR) continues to count (decrement) after falling through $FF to zero, it can be read any time by the processor without disturbing the count. This allows a program to determine the length of time since a timer interrupt has occurred without disturbing the counting process. Once the processor receives the timer interrupt, the MCU responds by saving the present CPU state on the stack, fetching the timer vector, and executing the interrupt routine. The processor is sensitive to the level of the timer interrupt request line; therefore, if the interrupt is masked (I bit set), bit 7 of the TCR may be cleared by the timer interrupt service routine without generating an interrupt. When servicing a timer interrupt, bit 7 of the TCR must be cleared by the timer interrupt service routine in order to clear the timer interrupt request.

At power up or reset, the prescaler and TDR (8-bit counter) are initialized with all logical ones, TCR bit 7 is cleared, and TCR bit 6 is set.

NOTE

The above description does not fully apply to EPROM members of the M6805 HMOS/MC146805 CMOS Family (or the MC6805R3 and MC6805U3). This is because EPROM MCUs use TCR bits 0-5 to select prescaler output divide ratio, determine clocking source, and clear the prescaler. EPROM versions may also be programmed, via the MOR, to allow the prescaler to be software programmed.

4.10 ANALOG-TO-DIGITAL (A/D) CONVERTER

The MC6805R2 MCU and MC68705R3 EPROM MCU both have an 8-bit A/D converter implemented on-chip. This A/D converter uses a successive approximation technique, as shown in Figure 4-9. Up to four external analog inputs, via port D, may be connected to the A/D converter through a multiplexer. Four internal analog channels may be selected for calibration purposes (V_{RH}, V_{RL}, $V_{RH/2}$, and $V_{RH/4}$). The accuracy of these internal channels will not necessarily meet the accuracy specifications of the external channels.

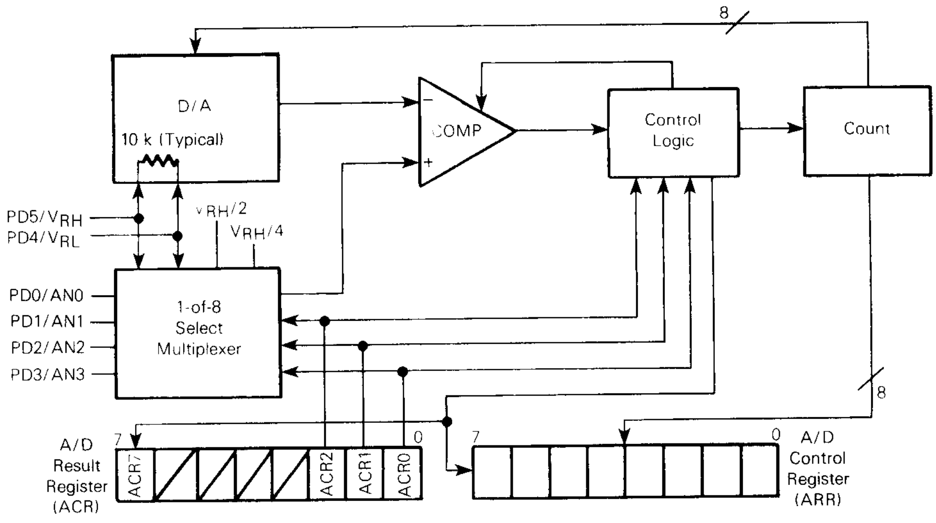


Figure 4-9. A/D Block Diagram

The multiplexer selection is controlled by the A/D control register (ACR) bits 0, 1, and 2; see Table 4-3. This register is cleared during any reset condition.

Table 4-3. A/D Input Multiplexer Selection

A/D Control Register			Input Selected	A/D Output (Hex)		
ACR2	ACR1	ACR0		Min	Typ	Max
0	0	0	AN0			
0	0	1	AN1			
0	1	0	AN2			
0	1	1	AN3			
1	0	0	V_{RH}*	FE	FF	FF
1	0	1	V_{RL}*	00	00	01
1	1	0	$V_{RH/4}$*	3F	40	41
1	1	1	$V_{RH/2}$*	7F	80	81

*Internal (Calibration) levels

Whenever the ACR is written, the conversion in progress is aborted, the conversion complete flag (ACR bit 7) is cleared, and the selected input is sampled and held internally.

The converter operates continuously using 30 machine cycles (including a 5-cycle sample time) to complete a conversion of the sampled analog input. When conversion is complete, the digitized sample or digital value is placed in the A/D result register (ARR), the conversion complete flag is set, the selected input is sampled again, and a new conversion is started. Conversion data is updated during the part of the internal cycle that is not used for a read. This ensures that valid, stable data is continuously available after initial conversion.

NOTE

Negative transients on any analog lines during conversion will result in an erroneous reading.

The A/D is ratiometric. Two reference voltages (V_{RH} and V_{RL}) are supplied to the converter via port D pins. An input voltage greater than V_{RH} converts to $FF and no overflow indication is provided. For ratiometric conversions, the source of each analog input should use V_{RH} as the supply voltage and be referenced by V_{RL}.

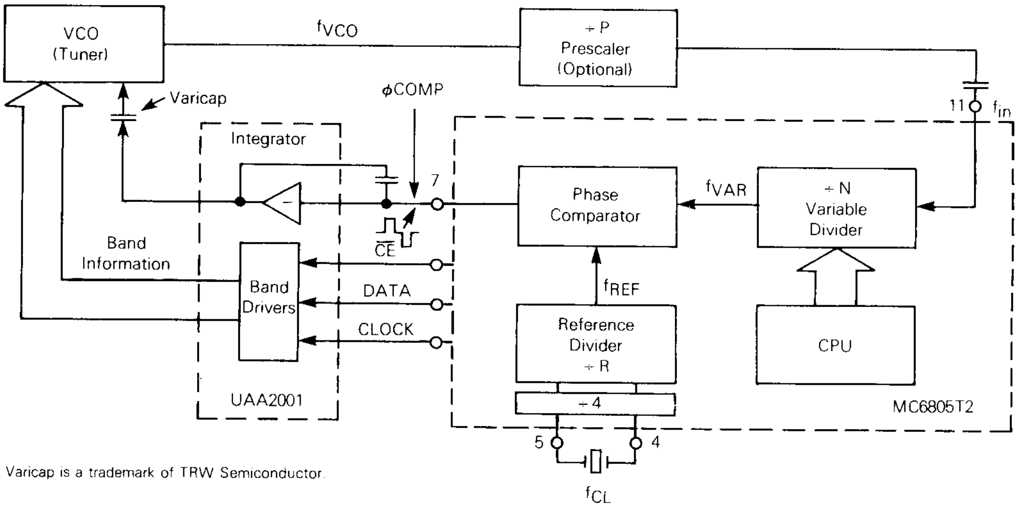
4.11 PHASE-LOCK-LOOP (PLL)

4.11.1 General

The MC6805T2 MCU contains (in addition to the normal ROM, RAM, timer, and I/O functions) a phase-lock-loop (PLL). This feature, not normally found in an MCU, may be used in applications ranging from television tuner control to public service scanner radios.

By providing a PLL which is part of the on-chip MCU circuitry, the functions of frequency control and front panel indication are easily attained. The MC6805T2 contains sufficient ROM and RAM for a program allowing for controlling all the necessary television channels currently used, plus a display showing the channel number.

Figure 4-10 contains a block diagram of a PLL system in an rf synthesizer and Figure 4-11 shows the on-chip MC6805T2 components. As shown, the system components internal to



Varicap is a trademark of TRW Semiconductor.

Figure 4-10. Phase-Lock-Loop in an rf Synthesizer

the MC6805T2 MCU contain: a 14-bit binary variable divider (÷N), a fixed 10-stage reference divider (÷R), a digital phase and frequency comparator with a three-state output, and circuitry to avoid "back-lash" effects in phase lock condition. External to the MCU, a suitable high-frequency prescaler (÷P) and an active integrator loop filter plus a VCO rounds out the system.

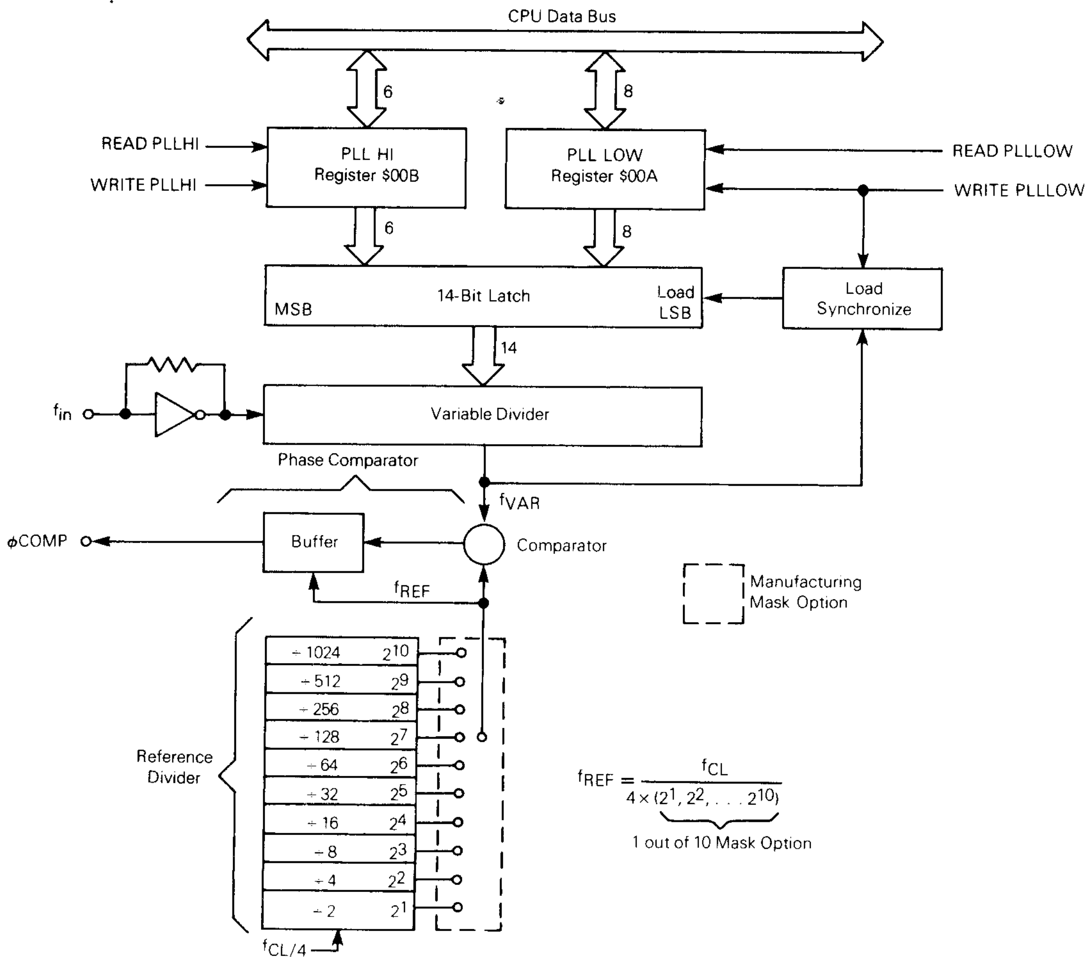


Figure 4-11. MC6805T2 PLL Block Diagram

4.11.2 Reference Divider

Refer to Figure 4-11. This 10-stage binary counter generates a reference frequency which is applied as a constant reference frequency to a phase comparator circuit. The reference divider is mask programmable, thus, allowing the user a choice of reference frequency at the time of manufacture.

4.11.3 Variable Divider

The variable divider (shown in Figure 4-11) is a 14-bit binary down counter which communicates with the CPU via two read/write registers located at address $00A, for the LS byte, and $00B, for the MS byte. The upper two bits in register $00B, always read as logical "1s". When the variable divider count has reached zero, a preset pulse, f_{VAR}, is generated. The f_{VAR} is applied to the phase comparator circuit together with the constant frequency f_{REF} signal. The phase/frequency difference between the two signals results in an error signal output (ϕ COMP, pin 7) which is used to control the VCO frequency. In addition, the f_{VAR} signal is also used to reload the 14-bit divider latch as shown in Figure 4-11.

Data transfers from registers $00A and $00B to the latch occur outside the preset time and only during a write operation performed on register $00A. For example, a 6-bit data transfer to register $00B is only transferred to the variable divider if followed by a write operation to register $00A. Figure 4-12 shows a typical error free manipulation of the 14-bit data in the fine tuning operations.

```
FTUP    LDA     PLLLOW
        INCA                    Check if LS Byte=$FF (Reg $00A)
        BNE     TT1             If Not Increment Only LS Byte
        INC     PLLLOW          Increment MSB (Reg $00B) Before LSB
TT1     INC     PLLLOW
        •
        •
        •
FTDWN   TST     PLLLOW          Check if LS Byte=$00
        BNE     TT2             If Not Decrement Only LS Byte
        DEC     PLLHI           Decrement MSB Before LSB
TT2     DEC     PLLLOW
        •
        •
        •
```

Figure 4-12. Typical Fine Tune Software Example

The use of the 14-bit latch synchronizes the data transfer between two asynchronous systems, namely, the CPU and the variable divider.

At power-up reset both the variable divider and the contents of the PLL registers are set to logical "1s".

The variable frequency input pin, f_{in}, is self biased requiring an ac coupled signal of about 0.5 V. The input frequency range of f_{in} allows the device, together with a suitable prescaler, to cover the entire TV frequency spectrum.

4.11.4 Phase Comparator

The phase comparator compares the frequency and phase of f_{VAR} and f_{REF}, and according to the phase relationship generates a three-level output (1, 0, or Hi-Z), ϕCOMP, as shown in Figure 4-13. The output waveform is then integrated, amplified, and the resultant dc voltage is applied to the voltage controlled oscillator.

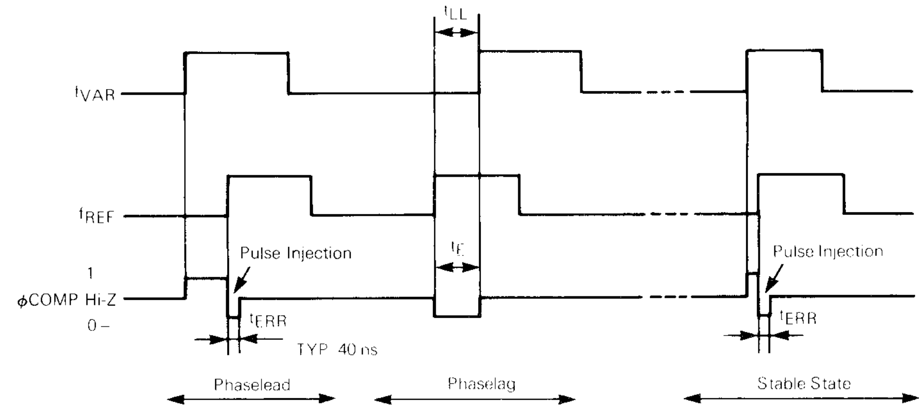


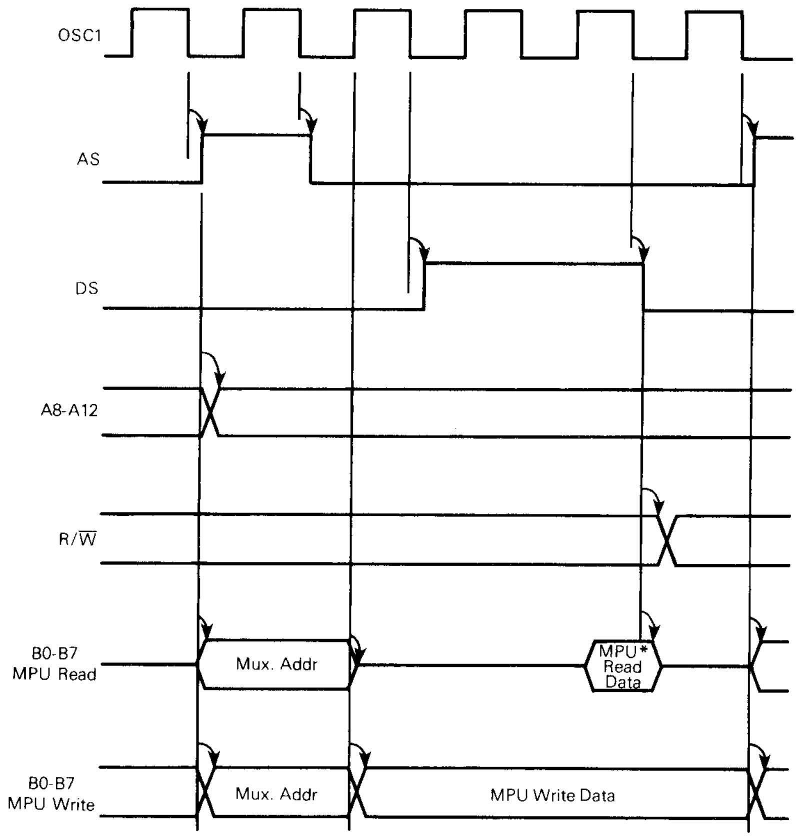
Figure 4-13. Phase Comparator Output Waveform

In practice, a linear characteristic around the steady-state region can not be achieved due to internal propagation delays. Thus, phase comparators exhibit non-linear characteristics and for systems which lock in phase, this results in a "backlash" effect—creating sidebands and FM distortion. To avoid this effect, a very short pulse is injected periodically into the system. The loop, in turn, attempts to cancel this interference and in so doing brings the phase comparator to its linear zone.

4.12 MC146805E2 MICROPROCESSOR (MPU) EXTERNAL BUS DESCRIPTION

The MC146805E2 CMOS MPU does not contain on-chip non-volatile memory; however, by using the external multiplexed address-then-data bus, additional memory and peripherals may be added. In order to conserve pins, the MC146805E2 multiplexes the data bus with the eight lower address bits. The lower address bits appear on the bus first and are valid prior to the falling edge of address strobe (AS). Data is then transferred during data strobe (DS) high. The MC146805E2 latches read data ($R/\overline{W}$ is high) on the falling edge of DS.

The MC146805E2 bus timing is generated from the waveform at the OSC1 input. Figure 4-14 shows the relationship of the MC146805E2 bus timing to the OSC1 input. Because the MC146805E2 is a completely static device, it may be operated at any frequency below its maximum (1 MHz bus) rate. Since generating the timing specifications for all of the possible frequencies is impossible, Figure 4-14 can be used to estimate the effects on bus timing for the oscillator frequency (f_{OSC}). For instance, decreasing f_{OSC} increases the multiplexed address hold time since the multiplexed bus does not switch until a half OSC1 cycle after AS goes low. On the other hand, the required read data hold time is not a function of f_{OSC}.



*Read data "latched" on DS fall.

Figure 4-14. OSC1 to Bus Transitions