

Summer 2003**Lecture 23****07/21/03****8254 Programmable Timer Chip on PC Motherboard**

8254 Programmable Interval Timer on the motherboard. Located at ports 40h-43h.

There is also a parallel interface adapter located at ports 60h-63h.

The timer channel is programmed by accessing ports 40h-43h, but there are a couple of bits on the parallel interface that must also be set to connect the output of the timer to the speaker. (bits 0 and 1 at port 61h)

The 8254 is a general purpose counter/timer chip. **It contains 3 programmable 16 bit counters that can be configured in several different modes.**

It is accessed via 4 parallel port addresses. Ports 0, 1, and 2 (at addresses 40h, 41h, and 42h on PC compatible computers) are the counter ports, and port 3 (at address 43h) is the control port.

The port at address 43h is the control port and is used to set the mode of operation for each channel.

The ports at 40-42 are the counter data ports for channels 0-2 respectively. Each channel contains a 16 bit counter.

The single 8 bit data port for each channel is used to read/write both bits of the counter for that channel. Bits in the control register (43h) indicate whether the high byte, low byte or both bytes are to be read or written. If both bytes are to be written, then two writes are required, the high byte followed by the low byte.

Counter Modes:

- Mode 0: Interrupt on terminal count**
- Mode 1: Hardware retriggerable one shot**
- Mode 2: Rate Generator**
- Mode 3: Square Wave Generator**
- Mode 4: Software Triggered Strobe**
- Mode 5: Hardware Triggered Strobe**

Timer Interrupt

The timer interrupt uses counter channel 0 of the 8254. It is programmed in mode 2, and generates an interrupt periodically. The frequency of this interrupt is approximately 18.2 hz. This is achieved by dividing a 1.19318 Mhz clock that appears on the motherboard by 65535 (the maximum possible count on the counter) The timer signal generated by the 8254 channel is applied to the IRQ 0 pin on the 8259 interrupt controller, which generates an interrupt 8. The timer interrupt is then hooked by hooking the INT 8 vector.

Interrupt Handler Chains

Often when using an interrupt like the hardware timer interrupt, it is necessary to preserve the old operation of the interrupt. In the case of the timer interrupt, the system uses this interrupt to maintain an internal clock. If a program were to simply take over this interrupt, the system clock would stop functioning. In this case the proper action is for the program to insert its timer interrupt use in front of the system behavior. This is accomplished by what is called interrupt hooking and an interrupt handler chain.

Before setting the interrupt vector to point to the its own interrupt handler, the program should get the current setting and save it in a local variable. It then sets the interrupt vector to point to its own interrupt handler.

In the program's interrupt handler, either before or after its own interrupt processing, it should pass control, via the saved interrupt vector, to the previous owner of the interrupt. If this is done before the current program's interrupt servicing is performed, this should be accomplished by pushing the flags and then performing a far call (through the saved vector) to the old interrupt routine. Otherwise, at the end of the current program's handling of the interrupt, it should pass control on to the previous owner of the interrupt vector by performing a far jump (through the saved vector) to the previous owner. This allows the interrupt behavior to be "subclassified" and the new

To generate a tone using the speaker on the PC motherboard, it is necessary to program timer channel 2 of the 8254 with the appropriate count for the desired frequency.

The basic frequency is 1.19318 Mhz, so a divisor needs to be determined to divide that frequency down to the desired output frequency.

$$A=440 \quad \text{div}=2712$$

The timer channel is then **programmed for Mode 3** operation with the appropriate count (e.g., 2712). This will cause the **output of the timer channel to be approximately a square wave at the desired frequency.**

To enable the output, it is necessary to write 1's to the two control bits in the port at 61h. Bit 0 enables the gate that drives the speaker, and bit 1 enables the gate input signal to the counter.