

17 DE MARZO DE 2017



## PRACTICA 5

INTERRUPCIONES TEMPORIZADAS Y E/S MAPEADA A MEMORIA

CARLOS OMAR CALDERON MEZA

MICROPROCESADORES Y MICROCONTROLADORES

FACULTAD DE CIENCIAS QUIMICAS E INGENIERIA – INGENIRO EN COMPUTACION

**Objetivo:** Uso de temporizadores para la implementación de un reloj simple con acceso a un puerto como indicador.

**Material:**

- Programas TCC, TASM, TLINK y MKBINTJ
- Tarjeta T-Juino.
- Programa Terminal.
- Protoboard y compuertas lógicas (según diseño).

**Equipo:**

- Computadora Personal

**Teoría:**

**Vectores de interrupción 08h y 1Ch**

**Interrupción:**

Una interrupción es el rompimiento en la secuencia de un programa para ejecutar un programa especial llamando una rutina de servicio cuya característica principal es que al finalizar regresa al punto donde se interrumpió el programa.

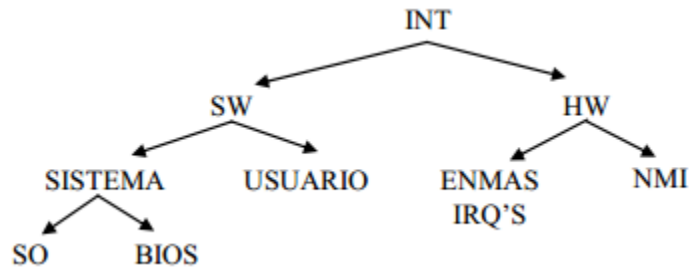
Dentro de una computadora existen dos clases de interrupciones:

- **Interrupciones por software:** Son aquellas programadas por el usuario, es decir, el usuario decide cuando y donde ejecutarlas, generalmente son usadas para realizar entrada y salida.

- **Interrupciones por hardware:** Son aquellas que son provocadas por dispositivos externos al procesador su característica principal es que no son programadas, esto es, pueden ocurrir en cualquier momento en el programa. Existen dos clases de interrupciones de este tipo:

- **Interrupciones por hardware enmascarables:** Aquellas en las que el usuario decide si quiere o no ser interrumpido.

➤ Interrupciones por hardware no enmascarables (NMI): Aquellas que siempre interrumpen al programa.



Las interrupciones por software se ejecutan con ayuda de las instrucciones: INT e IRET, además se tiene 256 interrupciones: de la 00 a la FF.

Asociado al concepto de interrupción se tiene un área de memoria llamada **vector de interrupciones**; la cual contiene las direcciones de las rutinas de servicio de cada interrupción. Esta área se encuentra en el segmento 0000:0000. Para cada una de las direcciones se utilizan 4 bytes, dos bytes para el segmento y 2 para el desplazamiento.

**Timer (INT 8):** Las PC tiene internamente un circuito que es el Timer, y el cual tiene la función de interrumpir el CPU, un número de veces determinado por segundo. Su rutina de servicio tiene la función de actualizar contadores en variables internas de la computadora para que puedan usarse en la actualización de la fecha y la hora; además de apagar el motor que mueve el disco si este no es usado, finalmente provoca una interrupción por software, la **INT 1Ch**, quien inicialmente puede usarse en rutinas de usuario que requieren sincronización con el tiempo.

El contador empieza con un número y cada pulso que le da el timer, éste se decrementa y cuando llega a cero entonces el timer, interrumpe y ejecuta la INT 8, sólo en caso de que sea aceptado por el CPU y después vuelve el contador con el valor inicial. Además, apaga el motor y el foco de las unidades del disco.

Las rutinas de servicio de las interrupciones por hardware deben ser transparentes al usuario, esto es, deben preservar todos sus registros.

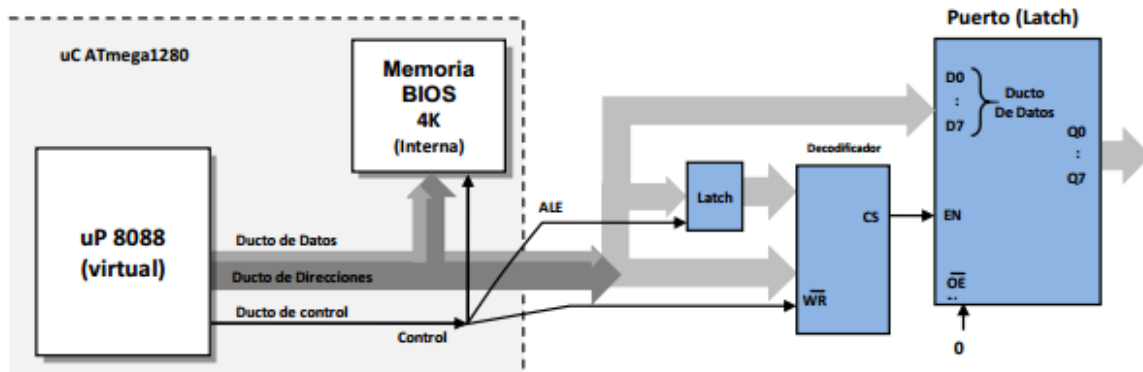
## Desarrollo

### Actividades (diseño e implementación)

#### Hardware:

a) Circuito para incluir a T-Juino un puerto de salida mapeado en memoria en la dirección XXXXH.

#### Dirección 0xFFFF



**Figura 1.** Esquema de puerto de salida mapeado en memoria.

#### Software:

b) Realice los cambios necesarios a las funciones **UpdateClock()** y **DisplayClock()**.

La función **UpdateClock()** tiene como tarea actualizar el reloj; es decir una vez que es ejecutada esta incrementa los segundos para entonces actualizar los minutos, y horas del reloj. Para el caso de **DisplayClock** tiene la tarea de desplegar en pantalla el reloj en el formato militar (24 Hrs) "**hh:mm:ss**".

c) Diseñar e implementar la función **SetClock()** la cual recibe como parámetro la hora, minuto y segundo (formato 24 Hrs) con el fin de inicializar el reloj.

```
void SetClock( BYTE Hora, BYTE minutos, BYTE segundos)
```

d) Diseñar e implementar la secuencia apropiada para escribir un byte a puerto de salida mapeado en la dirección XXXXH de memoria.

#### Dirección 0xFFFF

## **Conclusiones y comentarios**

Las practicas hasta este momento durante el curso se han ido complementando para formar un sistema computacional, primeramente tenemos la sección de procesamiento o cpu con el Tj-uino después agregamos la sección de memoria con la RAM, después decodificadores y ahora la sección de E/S con un latch simulando un puerto, me parece muy interesante como se planeó o se complementaron las practicas.

El principal reto durante esta práctica fue que no se nos impartió sesión de laboratorio y no tenía ni idea de que tenía que hacer pero con compañeros investigando nos dimos cuenta que la práctica era muy sencilla.

## **Bibliografía**

MC Beatriz Beltrán Martínez. (2010). Interrupciones. 17-03-2017, Sitio web:  
<http://bbeltran.cs.buap.mx/Interrupciones.pdf>