Práctica No. 5b

• Sección de Entrada/Salida

Objetivo: Acceso a los puertos genéricos del dispositivo programable PPI-8255 que se encuentra **emulado** en la tarjeta T-Juino.

Material: - Programa ensamblador y encadenador (TASM/TLINK)

- Tarjeta T-Juino.

Equipo: - Computadora Personal

Teoría: *** INVESTIGACION: INTERFAZ DE PUERTOS PARALELO 8255 ***

Desarrollo:

Parte A: Programación y uso de puertos del PPI8255.

1) Crear archivo **Prac5b.c** que contenga a partir del listado 1

Listado 1

```
#define BYTE unsigned char
#define WORD unsigned int
#define PA
                         0x40
#define PB
#define PC
                         0x41
                         0x42
#define RCtr
                        0x43
#define PTOs_all_out 0x80
char dato;
void main( void ){
   puts("Practica 5b\n\r");
   outportb(RCtr, PTOs_all_out);
   outportb(PA,0x55);
   while(1){
      dato = getch();
      outportb(PB,dato);
      printBin(dato);
      puts("\n\r");
}
                                                             /* imprimir mensaje
/* inicializar 8255
                                                             /* presentar 55h en el Pto A
                                                             /* leer tecla
                                                             /* presentar tecla en PB
}
 /* función para lectura de puertos usando ensamblador in-line
void outportb( woRD port, BYTE dato){
  asm mov dx, port
  asm mov al, dato
  asm out dx, al
/* función simple para desplegar un byte en formato binario void printBin( BYTE dato ){ BYTE msk=0x80;
                                                                                                              */
       do{
             putchar( (dato & msk) ? '1' : '0' );
       }while( msk');
}
```

2) Realizar proceso para ejecutar el programa en T-Juino.

3) La siguiente función llamada **SetBitPort** manipula la información de un puerto dado para activar un determinado bit. Es decir mediante ella se puede **activar** (hacer uno) un bit del puerto. El número del bit está en el rango de 0 a 7 siendo el bit 7 es más significativo.

Listado 2

<u>Listado 2 – Simplificado</u>

```
void SetBitPort(WORD Puerto, BYTE num_bit)
{
   outportb( Puerto , inportb( Puerto )|( 0x01 << num_bit ) );
}</pre>
```

4) Diseñe las siguientes funciones para manipulación de bit de puertos .

Nota: Es necesario implementar las funciones básica de E/S **inportb()** y **outportb()** en lenguaje ensamblador en archivo ASM independiente -- no usar funciones in-line.

 a) Función ClrBitPort la cual borrar un bit; es decir hace cero el bit de la posición num_bit del puerto dado por el parámetro Puerto.

```
void ClrBitPort( WORD Puerto, BYTE num_bit )
```

b) Función **NotBitPort** la cual **invierte** el bit de la posición num_bit del puerto dado por el parámetro Puerto.

```
void NotBitPort( WORD Puerto, BYTE num_bit )
```

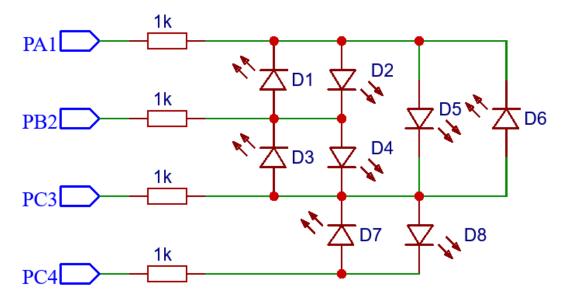
c) Función **TstBitPort** la cual retorna el estado del bit de la posición num_bit del puerto dado por el parámetro Puerto. Si el bit del puerto está en 0 lógico entonces la función retorna valor cero (0) de otra forma retorna valor uno (1).

```
BYTE TstBitPort ( WORD Puerto, BYTE num_bit )
```

d) Función **ReverseBitsPort** la cual invierte el orden de los bits del Puerto, es decir que el menos significativo pasa a ser el más significativo. Ejemplo, si en el puerto se encontraba el valor **0b11001010**, después de llamar esta función tendríamos **0b01010011** en el Puerto.

```
void ReverseBitsPort ( WORD Puerto )
```

- 5) Verifique el funcionamientos de las funciones del punto anterior realizando los siguientes pasos:
 - a. Capturar 8 bits mediante *TstBitPort* en **PC4**, dando tiempo para que el usuario pueda alterar el valor de **PC4** (por ejemplo, detener el programa mediante un *getchar()*), y mostrar los bits capturados en la terminal,
 - b. Invertir el orden de los bits del dato capturado y mostrar los 8 bits resultantes sobre el siguiente arreglo de LEDs:



Respetando D1 como el LSB, y D8 como el MSB.

Conclusiones y Comentarios.

Bibliografía