



**Universidad Autónoma de Baja California
FAC. DE CS. QUIM. E INGENIERIA
INGENIERIA EN COMPUTACION**

PRACTICA 5b

Laboratorio de: Microprocesadores y microcontradores

Equipo:

López Madrigal Leonardo

Maestro:

García López Jesús Adán y Leocundo Aguilar Noriega

Tijuana, B. C.

4 Abril, 2017

Interrupciones temporizadas y E/S mapeada a memoria

Objetivo:

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

Material:

-Resistencias y LEDs para T-Juino.

Equipo:

- Computadora Personal
- Tarjeta T-Juino.
- Protoboard
- Compuertas lógica.
- Diodos

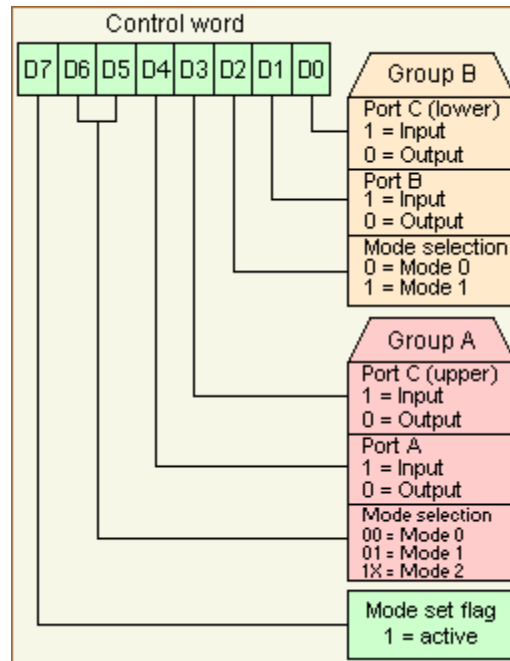
Teoría:

-INVESTIGACION: INTERFAZ DE PUERTOS PARALELO 8255

Teoría

La interfaz de periféricos programable 8255A-5 (PPI) es un componente de bajo costo muy popular encontrado en muchas aplicaciones. El PPI tiene 24 terminales para E/S, programables en grupos de 12, que se emplea en tres modos separados de operación. El 8255A-5 puede interfazar cualquier dispositivo de E/S compatible TTL, al microprocesador 8088.

Sus tres puertos de E/S (etiquetados A, B y C) se programan en grupos de 12 terminales. Las conexiones del grupo A consisten del Puerto A (PA7-PA0) y la mitad superior del puerto C (PC7-PC4), y el grupo B consiste del Puerto B (PB7-PB0) y la mitad inferior del Puerto C (PC3-PC0).



El 8255A-5 se selecciona por su terminal para programación y lectura o escritura a un puerto. La selección del registro es realizada a través de la terminales A1 y A0 que seleccionan un registro interno para operaciones de programación. CS

Los terminales del grupo B (puerto B y la parte baja del puerto C) se programan como terminales de entrada o salida. El grupo B puede operar en el modo 0 o en el modo 1. El modo 0 es el modo básico de E/S que permite que los terminales del grupo B sean programados como simples entradas y conexiones de salida amarradas. La operación del modo 1 es la operación de habilitación periódica para las conexiones del grupo B donde los datos se transfieren a través del puerto B, y el puerto C proporciona las señales de reconocimiento.

Los terminales del grupo A (puerto A y la parte superior del puerto C) son también programadas como terminales de entrada o de salida. La diferencia es que este grupo A puede operar en los modos 0, 1 y 2. La función del modo 2 es operar en modo bidireccional para el puerto A.

Circuito Integrado



Patillas del 8255 PPI

PA3	1	40	PA4
PA2	2	39	PA5
PA1	3	38	PA6
PA0	4	37	PA7
\overline{RD}	5	36	\overline{WR}
\overline{CS}	6	35	RESET
GND	7	34	D0
A1	8	33	D1
A0	9	32	D2
PC7	10	31	D3
PC6	11	30	D4
PC5	12	29	D5
PC4	13	28	D6
PC0	14	27	D7
PC1	15	26	Vcc
PC2	16	25	PB7
PC3	17	24	PB6
PB0	18	23	PB5
PB1	19	22	PB4
PB2	20	21	PB3

Desarrollo:

Programar el 8255^a

Yo utilice el puerto A como salida y el puerto C completo como entrada

```
#define PA      0x40
#define PB      0x41
#define PC      0x42
#define RCtrl   0x43
#define PTOs_Cin_ABout 0x81

extern void outportb( WORD port, BYTE dato );
extern char inportb( WORD port );

void SetBitPort( WORD puerto, BYTE num_bit );
void NotBitPort( WORD puerto, BYTE num_bit );
BYTE IstBitPort ( WORD puerto, BYTE num_bit );
void ClrBitPort( WORD puerto, BYTE num_bit );
BYTE ReverseBitsPort ( WORD puerto );

void putchar( char dato );
char getch( void );
void puts( char *str );
void printBin( BYTE dato );

BYTE dato;

void main( void ){

    puts("Practica 5a\n\r");          /* imprimir mensaje*/
    outportb(RCtrl,PTOs_Cin_ABout);  /* inicializar 8255 */
```

Funciones a realizar:

```

void SetBitPort(WORD puerto, BYTE num_bit){
    outportb(puerto, inportb(puerto) | (1 << num_bit));
}

void NotBitPort( WORD puerto, BYTE num_bit){
    outportb(puerto, inportb(puerto) ^ (1 << num_bit));
}

void ClrBitPort( WORD puerto, BYTE num_bit ){
    outportb(puerto, inportb(puerto) & ~(1 << num_bit));
}

BYTE TstBitPort ( WORD puerto, BYTE num_bit ){
    BYTE mask = 0x01 << num_bit;
    BYTE dato = inportb(puerto);
    if(dato & mask){
        return 1;
    } else {
        return 0;
    }
}

```

BYTE reverseBitPort(WORD puerto);

```

BYTE ReverseBitsPort ( WORD puerto){
    BYTE mask = 0x01;
    BYTE x=7;
    BYTE i=0, temp;
    BYTE datot = inportb(puerto);
    for(i=0; i<8; i++){
        if( (mask << i) & datot ){
            temp |= (1 << x-i);
        } else {
            temp &= ~(1 << x-i);
        }
    }
    outportb(puerto,temp);
}

```

5) Verifique los funcionamientos de las funciones del punto anterior realizando un programa de prueba.

Conclusión:***López Madrigal Leonardo***

En esta práctica aplicamos la explicación en la clase de cómo utilizar los puertos y configurarlos con la interfaz de puertos y el enmascaramiento de bits.

Lo que se tuvo que hacer fue hacer funciones de clearbit, setbit, notbit, test bit y reverseBitPort las cuales funcionan con una máscara que cambia el estado de bit deseado, pero estas funciones se aplican en datos que se obtienen por puertos previamente programados de entrada y salida, entonces lo que se hizo fue el la mitad del puerto C se programó de salida y la otra mitad de entrada, y un bit de la mitad de entrada entrada se mostraba por otro bit del de salida, esto era por medio de un testbit al bit de entrada si daba 1 se hacía un set en el bit de salida.

Se hizo el programa de prueba y debido a que el intérprete podía hacer la función de puertos PPI 8255 pudimos realizarlas y ver cómo funcionaban los puertos en su tiempo.

Bibliografía o referencias:

https://es.wikipedia.org/wiki/Intel_8255