## 

***Instituto politécnico nacional***

Escuela Superior de Ingeniería Mecánica y eléctrica, plantel Culhuacán

*Practica 4*

***Práctica 4***

***“CONVERTIDOR binario a bcd”***

Equipo 5

**Suarez Vega Edgar Alan**

**González Martínez Fernando**

**Marin Izquierdo Aurelio**

*Organización de computadoras*

5CV25

*Profesor: Galván Chávez José*

* ***Objetivo***

Convertir una entrada binaria en una salida BCD que se muestre en un display de 7 segmentos.

* ***Descripción del experimento:***
* Usar el programa “MPLAB” para realizar la codificación del funcionamiento del PIC16F84A que consiste en leer una entrada binaria, esta se tomara para que la salida seleccione los pines correctos del display de 7 segmentos y muestre su equivalente BCD.
* Usar Proteus para armar el circuito que cumplirá con el funcionamiento de la práctica que consiste en la lectura de código binaria.
* Una vez terminado el circuito, añadir el código del PIC al circuito para que funcione correctamente.
* Realizar pruebas y comprobar que el circuito efectivamente realizan la detección de entrada en código binaria.
* ***Materiales***
* Programa MPLAB; instrucciones usadas

***Las instrucciones que se usarán en el programa son:***

* **movf:** Esta instrucción mueve el valor indicado al registro “w”
* **bsf:** Habilita la salida indicada.
* **bcf:** Deshabilita la salida indicada.
* **clrf:** Limpia las entradas o salidas de información.
* **movwf:** Envía el valor que se encuentra en el registro “w” al lugar indicado.
* **subwf:** Resta, por el método de complemento a dos, el contenido del registro “f” con el contenido del registro “w”, y almacena el resultado.
* **btfss:** Comprueba si el bit y salta a la siguiente instrucción si es 1.
* **btfsc:** Comprueba si el bit y salta a la siguiente instrucción si es 0.
* **goto:** Se dirige a la sección de código que se indica.
* **movlw:** Asigna el valor indicado a el registro “w”.
* Programa Proteus; componentes usados
* PIC16F84A
* Display 7 segmentos ánodo comun
* 4 Resistencias de 10 ohms
* 4 interuptores 2 polos
* 7 resistencias 10 ohms
* Transistor BJT 547
* Fuente de alimentacion DC min 5 volts

***Desarrollo***

* El primer paso es programar las instrucciones del PIC16F84A en MPLAB y guardar el archivo ASM, el cual introduciremos en el circuito una vez terminado.
* El segundo paso es abrir proteus, buscar todos los componentes como resistencias, leds, transistores, entre otros y añadir los componentes ya mencionados a la lista de materiales a usar.
* Colocar los componentes en la interfar y acomodarlos, después conectarlos entre sí, las entradas con el PIC y el PIC con las salidas a los les que a su vez estos tienen que estar conctados a las resistencias.
* En las opciones del PIC, cargar el código ya hecho en MPLAB para que este pueda realizar la lectura de la entrada binario.
* Encender el circuito y comprobar que el funcionamiento del este es el correcto.

PROGRAMA DE MPLAB

#include "P16F84.inc" ;Libreria para el pic

TEMP EQU 0XC

BSF STATUS,5 ; IR A LA PÁGINA1 o Bloque de memoria 1

MOVLW b'00001111'

MOVWF TRISA ;Definimos los primeros 4 pines del puerto A como entrada

MOVLW b'00000000'

MOVWF TRISB ;Definimos el puerto B como salida

BCF STATUS,5 ;IR A PÁGINA 0 o banco de memoria 0

BUCLE

;CLRF PORTA

;CLRF PORTB ;INICIO DEL PORTB A CERO

CLRW

MOVLW b'00000000'

SUBWF PORTA,W ; Restamos para comparar

BTFSC STATUS,2 ; si da cero es igual al bcd y esta bien

GOTO B0

MOVLW b'00000001'

SUBWF PORTA,W ; Restamos para comparar

BTFSC STATUS,2 ; si da cero es igual al bcd y esta bien

GOTO B1

MOVLW b'00000010'

SUBWF PORTA,W ; Restamos para comparar

BTFSC STATUS,2 ; si da cero es igual al bcd y esta bien

GOTO B2

MOVLW b'00000011'

SUBWF PORTA,W ; Restamos para comparar

BTFSC STATUS,2 ; si da cero es igual al bcd y esta bien

GOTO B3

MOVLW b'00000100'

SUBWF PORTA,W ; Restamos para comparar

BTFSC STATUS,2 ; si da cero es igual al bcd y esta bien

GOTO B4

MOVLW b'00000101'

SUBWF PORTA,W ; Restamos para comparar

BTFSC STATUS,2 ; si da cero es igual al bcd y esta bien

GOTO B5

MOVLW b'00000110'

SUBWF PORTA,W ; Restamos para comparar

BTFSC STATUS,2 ; si da cero es igual al bcd y esta bien

GOTO B6

MOVLW b'00000111'

SUBWF PORTA,W ; Restamos para comparar

BTFSC STATUS,2 ; si da cero es igual al bcd y esta bien

GOTO B7

MOVLW b'00001000'

SUBWF PORTA,W ; Restamos para comparar

BTFSC STATUS,2 ; si da cero es igual al bcd y esta bien

GOTO B8

MOVLW b'00001001'

SUBWF PORTA,W ; Restamos para comparar

BTFSC STATUS,2 ; si da cero es igual al bcd y esta bien

GOTO B9

GOTO NOBCD

B0

MOVLW b'11000000'

MOVWF PORTB

GOTO BUCLE

B1

MOVLW b'11111001'

MOVWF PORTB

GOTO BUCLE

B2

MOVLW b'10100100'

MOVWF PORTB

GOTO BUCLE

B3

MOVLW b'10110000'

MOVWF PORTB

GOTO BUCLE

B4

MOVLW b'10011001'

;gfedcba

MOVWF PORTB

GOTO BUCLE

B5

MOVLW b'10010010'

;gfedcba

MOVWF PORTB

GOTO BUCLE

B6

MOVLW b'10000010'

;gfedcba

MOVWF PORTB

GOTO BUCLE

B7

MOVLW b'11111000'

;gfedcba

MOVWF PORTB

GOTO BUCLE

B8

MOVLW b'10000000'

;gfedcba

MOVWF PORTB

GOTO BUCLE

B9

MOVLW b'10010000'

;gfedcba

MOVWF PORTB

GOTO BUCLE

NOBCD

MOVLW b'00000000'

MOVWF PORTB

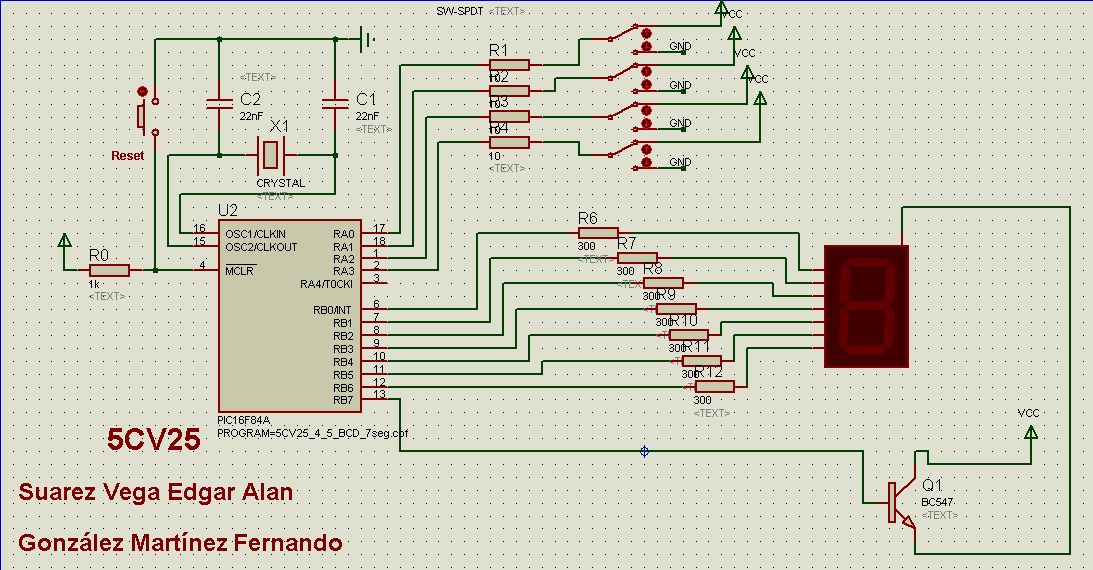
CLRW

GOTO BUCLE

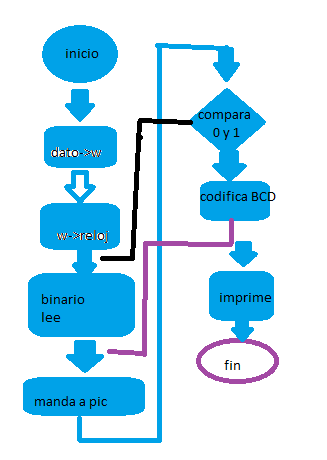
EN

***Diagrama del circuito electronico***

***Imagen que contiene texto, pizarrón

Descripción generada automáticamente***

***Diagrama de bloques y de flujo***

******

***Conclusión:***

El pic16f84a es una herramienta poderosa con la cual podemos controlar una serie de entradas y salidas, siendo que podemos programar el funcionamiento del pic programando podemos conseguir que este mismo interactúe con las salidas y entradas de acuerdo a como lo necesitemos.

Podemos emplearlo para poder entender un conjunto de bits y traducirlo en BCD, lo cual nos facilita la lectura de los valores que lee el PIC y que gráficamente nos es mas fácil entender.