Informe 11
Manejo de PCL

Laboratorio de Arquitectura del Computador

Elaborado por:
Tomás Guzmán, 21615008

1 MARCO TEÓRICO

Por los momentos se ha usado Bit Test y GOTOs para hacer saltos en el programa. Pero ¿qué sucede si se desea hacer un salto de más de una línea?, ¿digamos dos líneas?, ¿tres?

Para esto existen registros en el PIC donde se modifica el PC (Program Counter), de tal forma que se puede brincar un número dado de instrucciones, según desee el programador.

El PC en el PIC mide 13 bits y solo sus últimos bits (bajos) pueden ser accedidos por el programador. Para esto está el registro PCL (PC Low, bajo).

Normalmente la forma de utilizar es brincar a una subrutina (CALL) y sumarle un valor contenido en W, que puede ser cualquiera. Esto hará que se salte n número de líneas en el código al ser llamado.

Este programa tiene algunas instrucciones y operaciones nuevas:

- CALL, llama a una subrutina. Muy parecido al GOTO que hemos usado, pero ahora guarda el PC en el Stack del PIC y accede a la rutina. Luego con RETURN se saca el PC del stack y se continúa desde donde se llamó la subrutina. De un GOTO no hay luego un RETURN. De un CALL sí.
- El RETURN no tiene efectos secundarios, pero RETLW sí. Esta operación que significa "Return with Literal in W", lo que significa que se sale de la rutina con un valor devuelto en W. Muy parecido a lo que sería asignar el resultado de una función en un lenguaje de alto nivel.
- ADDWF, simplemente suma a W con F. Además acepta un parámetro opcional. Si es 0, entonces se guarda el resultado en W, sino en F.

2 IMPLEMENTACIÓN

```
    LIST P=16F84

STATUS  EQU 03H
PORTA   EQU 05H
PORTB   EQU 06H
TRISA   EQU 05H
TRISB   EQU 06H
PCL     EQU 02H

COUNT   EQU 10H

#DEFINE RA0     PORTA, 0
#DEFINE BANK0   BCF STATUS, 5
#DEFINE BANK1   BSF STATUS, 5

    ORG 0

    BANK1
    CLRF    TRISB
    MOVLW   B'11111111'
    MOVWF   TRISA           ; Ahora PORTA es todo de lectura
    BANK0

INICIO
    MOVF    RA0
    ANDLW   B'00001111'     ; Se enmascara para obtener solo los 4 primero
s bits
    CALL    CONVERT         ; Se llama a CONVERT
    MOVWF   PORTB           ; Se mueve W a PORTB
    GOTO    INICIO

CONVERT
    ADDWF   PCL, 1          ; Se suma W a PCL y se guarda en PCL
    RETLW   B'11000000'     ; Se guarda en W el código de segmentos de 0
    RETLW   B'11111001'     ; Se guarda en W el código de segmentos de 1
    RETLW   B'10100100'     ; Se guarda en W el código de segmentos de 2
    RETLW   B'10110000'     ; Se guarda en W el código de segmentos de 3
    RETLW   B'10011001'     ; Se guarda en W el código de segmentos de 4
    RETLW   B'10010010'     ; Se guarda en W el código de segmentos de 5
    RETLW   B'10000010'     ; Se guarda en W el código de segmentos de 6
```

```
    RETLW   B'11111000'     ; Se guarda en W el código de segmentos de 7
    RETLW   B'10000000'     ; Se guarda en W el código de segmentos de 8
    RETLW   B'10011000'     ; Se guarda en W el código de segmentos de 9
    RETLW   B'10001000'     ; Se guarda en W el código de segmentos de A
    RETLW   B'10000011'     ; Se guarda en W el código de segmentos de B
    RETLW   B'11000110'     ; Se guarda en W el código de segmentos de C
    RETLW   B'10100001'     ; Se guarda en W el código de segmentos de D
    RETLW   B'10000110'     ; Se guarda en W el código de segmentos de E
    RETLW   B'10001110'     ; Se guarda en W el código de segmentos de F

END
```

2.1 COMPILACIÓN

Este programa en extensión .asm fue compilado haciendo uso de MPASMWIN.

Este programa se incluyó en el PATH de Windows, por lo cual se puede ejecutar desde cualquier terminal.

Su sintaxis es sencilla, en este caso:

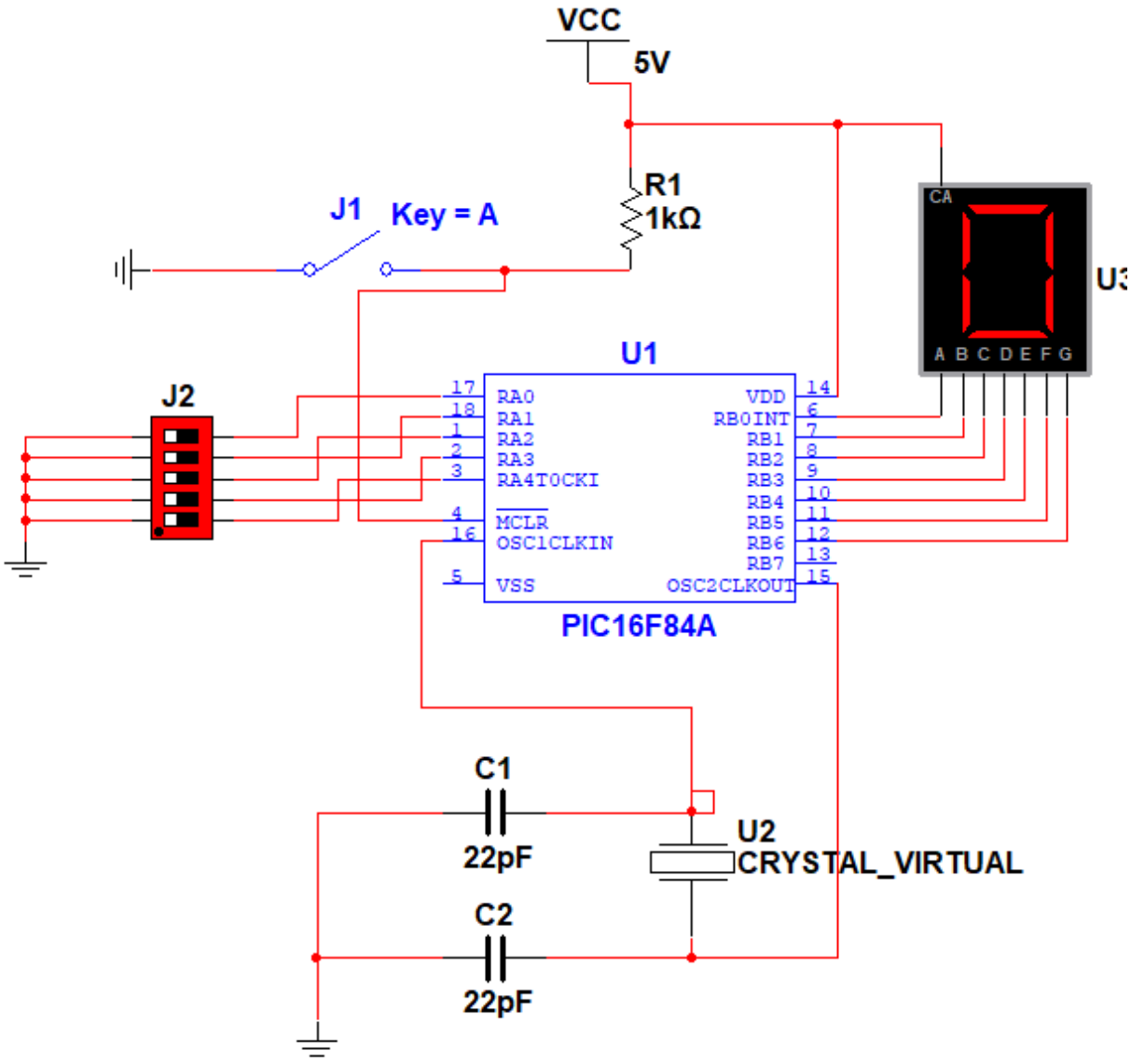
$ mpasmwin practica11.asm

Produce varios archivos, entre ellos el practica11.hex. Este último se usará en el PIC implementado en Multisim para visualizar la funcionalidad.

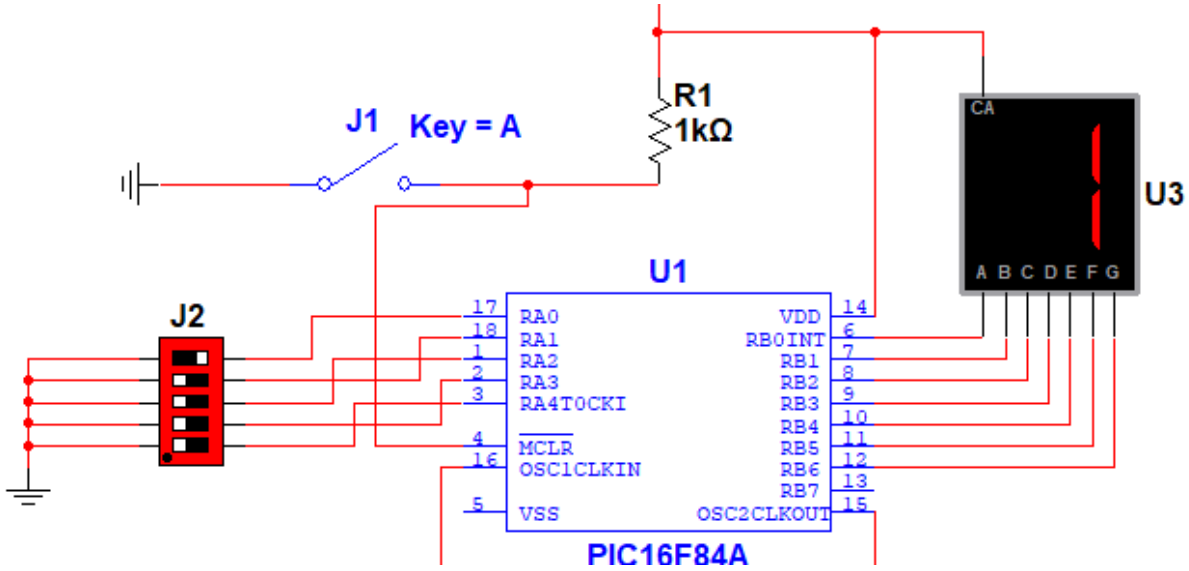
3 SIMULACIÓN

Esta simulación se realizó en Multisim y se puede observar como a medida que el switch físico es modificado, entonces también lo será el display.

Recuerde que siempre hay CALL a CONVERT, no importa que haya en el switch físico.



Veamos el primer caso, que es cuando el switch está completamente apagado por lo cual PORTA = 0000. Esto significa que al ser llamado CONVERT se le va a sumar 0 a PC, por lo cual la siguiente instrucción será ejecutada con normalidad, mostrando un cero en pantalla. En las siguientes imágenes se tomará fotos más pequeñas, para apreciar la funcionalidad del programa:

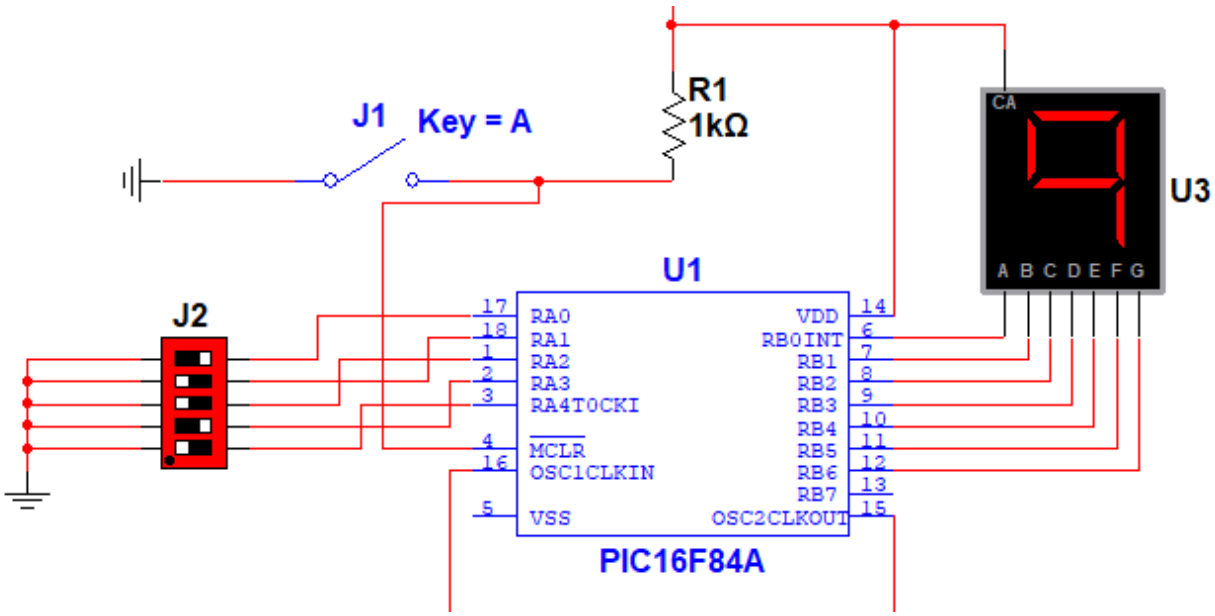


Note como ahora RA0 está activado, por lo cual el valor de PORTA es 1. Lo que significa que una línea de código será saltada en CONVERT

```
CONVERT
    ADDWF   PCL, 1          ; Se suma W a PCL y se guarda en PCL
    RETLW   B'11000000'     ; Se guarda en W el código de segmentos de 0
    RETLW   B'11111001'     ; Se guarda en W el código de segmentos de 1
```

La línea roja será saltada, y se hará el RETLW con el código de 1 a la rutina inicial.

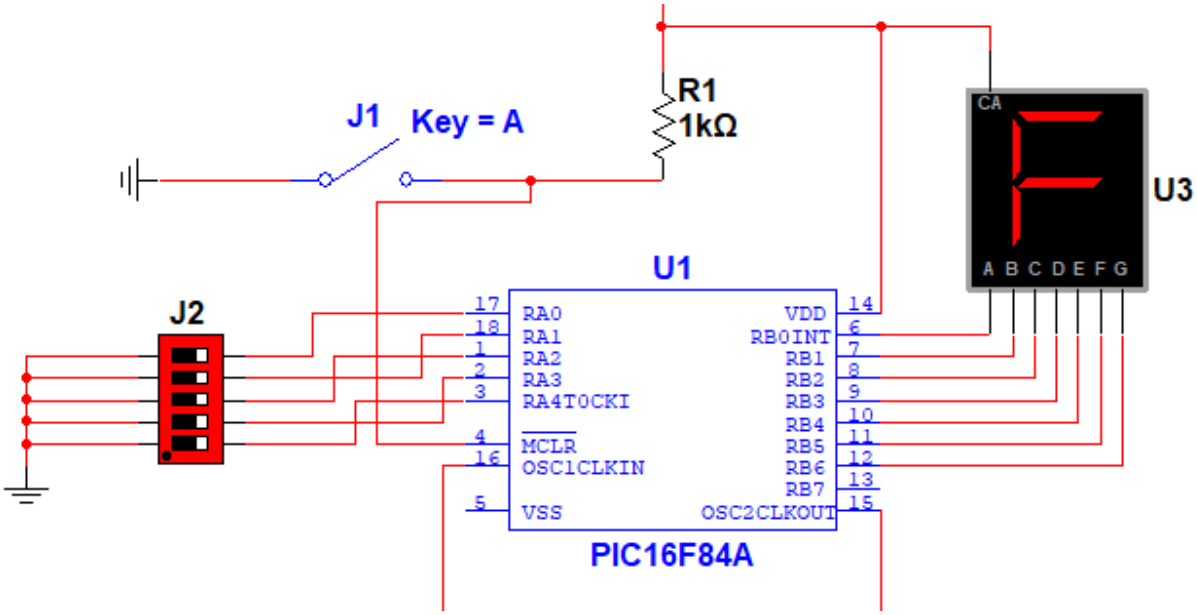
Esto va a suceder para cada número. Veamos más ejemplos



Ahora veamos que RA0 = RA3 = 1, haciendo que PORTA sea igual a 01001, que es 9 en decimal. Esto significa que 9 líneas de código serán saltadas, por lo cual el código del 9 será cargado, veamos en código.

```
CONVERT
    ADDWF   PCL, 1          ; Se suma W a PCL y se guarda en PCL
1   RETLW   B'11000000'     ; Se guarda en W el código de segmentos de 0
2   RETLW   B'11111001'     ; Se guarda en W el código de segmentos de 1
3   RETLW   B'10100100'     ; Se guarda en W el código de segmentos de 2
4   RETLW   B'10110000'     ; Se guarda en W el código de segmentos de 3
5   RETLW   B'10011001'     ; Se guarda en W el código de segmentos de 4
6   RETLW   B'10010010'     ; Se guarda en W el código de segmentos de 5
7   RETLW   B'10000010'     ; Se guarda en W el código de segmentos de 6
8   RETLW   B'11111000'     ; Se guarda en W el código de segmentos de 7
9   RETLW   B'10000000'     ; Se guarda en W el código de segmentos de 8
    RETLW   B'10011000'     ; Se guarda en W el código de segmentos de 9
```

Se puede apreciar cómo las 9 líneas rojas son saltadas y se ejecuta la línea azul. Esto funcionará con todos los casos. Veamos el último caso, que es que PORTA = 15.



Ahora se van a saltar 15 líneas de código

```
CONVERT
    ADDWF   PCL, 1          ; Se suma W a PCL y se guarda en PCL
1   RETLW   B'11000000'     ; Se guarda en W el código de segmentos de 0
2   RETLW   B'11111001'     ; Se guarda en W el código de segmentos de 1
3   RETLW   B'10100100'     ; Se guarda en W el código de segmentos de 2
4   RETLW   B'10110000'     ; Se guarda en W el código de segmentos de 3
5   RETLW   B'10011001'     ; Se guarda en W el código de segmentos de 4
6   RETLW   B'10010010'     ; Se guarda en W el código de segmentos de 5
7   RETLW   B'10000010'     ; Se guarda en W el código de segmentos de 6
8   RETLW   B'11111000'     ; Se guarda en W el código de segmentos de 7
9   RETLW   B'10000000'     ; Se guarda en W el código de segmentos de 8
10  RETLW   B'10011000'     ; Se guarda en W el código de segmentos de 9
11  RETLW   B'10001000'     ; Se guarda en W el código de segmentos de A
12  RETLW   B'10000011'     ; Se guarda en W el código de segmentos de B
13  RETLW   B'11000110'     ; Se guarda en W el código de segmentos de C
14  RETLW   B'10100001'     ; Se guarda en W el código de segmentos de D
15  RETLW   B'10000110'     ; Se guarda en W el código de segmentos de E
    RETLW   B'10001110'     ; Se guarda en W el código de segmentos de F
```

Como en los casos anteriores, las líneas rojas fueron saltadas y se procede a la azul, mostrando una F en la pantalla.

Esto claramente ocurre para las 16 combinaciones del switch físico.

4 CONCLUSIONES

- El PIC permite a su programador implementar saltos del contador de programa (PC), haciendo uso de un registro de un byte, que es PCL o PC(LOW). Los bits altos son manejados indirectamente por otras instrucciones como lo pueden ser GOTO, CALL, RETURN, RETFIE, etc.
- PCL es fácilmente manejable haciendo sumas sobre el mismo, guardándolas en la las mismas.
- La funcionalidad que esto ofrece es amplia, por ejemplo, se podría implementar un for-loop por un arreglo. Con lo que se conoce de PIC, se podría hacer un contador desde 0 a F donde cada letra se muestre 2 segundos y luego se regrese a 0.