Informe 8
PIC usando PORTA y PORTB

Laboratorio de Arquitectura del Computador

Elaborado por:
Tomás Guzmán, 21615008

1 MARCO TEÓRICO

En el trabajo anterior se cargó una palabra PORTB y se visualizó mediante unas luces (“probes”, inglés) azules.

En esta experiencia el programa no será tan lineal, por lo cual se estará trabajando con **control de flujo**.

Dado que el PIC es un dispositivo sencillo, entonces usa evaluaciones de bits de algún registro (hay más formas) para hacer saltos de instrucciones si se cumple la misma.

También será capaz de saltar a otra instrucción, dada su etiqueta (las palabras en mayúsculas que no están identadas en la sección de código).

1.1 INSTRUCCIONES

En esta experiencia usaremos nuevas instrucciones:

1. BTFSC, o Bit Test F Skip if Clear. Es una instrucción que recibe un registro, un bit y determina si el mismo es 0 (clear). De ser así, salta la próxima línea (no la ejecuta). Por eso dice “skip if clear”.
2. GOTO. Recibe una etiqueta. Se brincará a la misma

Estas instrucciones se pueden utilizar para simular un “if” de programación de alto nivel. Además, como en este caso son varios los “if”, se podría ver como un “switch”.

Por otro lado, el uso de GOTO puede crear un “while” o un ciclo de programación.

Veamos el código.

2 IMPLEMENTACIÓN

```
    LIST P=16F84

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

#DEFINE BANK0   BCF STATUS, 5
#DEFINE BANK1   BSF STATUS, 5

    ORG 0
SETUP
    BANK1
    CLRF    TRISB
    MOVLW   B'00011111' ; Todos los pines de PORTA son de lectura
    MOVWF   TRISA
    BANK0
    CLRF    PORTB

INICIO
    BTFSC   PORTA, 4
    GOTO    CUATRO      ; Si PORTA<4> es 1 entonces, se va a CUATRO

    BTFSC   PORTA, 3
    GOTO    TRES        ; Si PORTA<3> es 1 entonces, se va a TRES

    BTFSC   PORTA, 2
    GOTO    DOS         ; Si PORTA<2> es 1 entonces, se va a DOS

    BTFSC   PORTA, 1
    GOTO    UNO         ; Si PORTA<1> es 1 entonces, se va a UNO

    BTFSC   PORTA, 0
    GOTO    CERO        ; Si PORTA<0> es 1 entonces, se va a CERO

    GOTO INICIO         ; Ir a INICIO, creando un CICLO

CUATRO
    MOVLW   B'10011001' ; Codigo en 7 segmentos del número 4
```

```
    MOVWF   PORTB
    GOTO    INICIO

TRES
    MOVLW   B'10110000' ; Código en 7 segmentos del número 3
    MOVWF   PORTB
    GOTO    INICIO

DOS
    MOVLW   B'10100100' ; Código en 7 segmentos del número 2
    MOVWF   PORTB
    GOTO    INICIO

UNO
    MOVLW   B'11111001' ; Código en 7 segmentos del número 1
    MOVWF   PORTB
    GOTO    INICIO

CERO
    MOVLW   B'11000000' ; Código en 7 segmentos del número 0
    MOVWF   PORTB
    GOTO    INICIO
END
```

En la etiqueta INICIO estamos simulando nuestros “if”. El PORTA fue establecido como un puerto de lectura (este recibirá estímulo de un switch físico en el circuito).

Un aspecto particular de esta experiencia es que dado el orden en como esta escrita, si el bit 4 de PORTA está activado, entonces se verá un 4 en el display de 7 segmentos (también el circuito). No importa si los bits anteriores están activados, dado que el flujo del programa siempre pregunta primero por PORTA<4>, va a la etiqueta CUATRO y regresa a INICIO, en un ciclo infinito.

Si se desactiva este bit, entonces se preguntará por PORTA<3> y se brincará a TRES en caso de estar activado.

Esta “cascada” sucede hasta el 0.

Las etiquetas CUATRO, TRES, DOS, UNO, CERO son para señalar que estas mostrarán estos números en los displays de 7 segmentos.

Note como cada una de las mismas es muy parecida en términos de funcionalidad. Solo cargan palabras distintas a W que luego serán cargadas a PORTB, el cual fue configurado como salida al principio del programa.

Estas palabras que son cargadas a W son los códigos utilizados en los displays de 7 segmentos de cátodo común.

2.1 Compilación

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

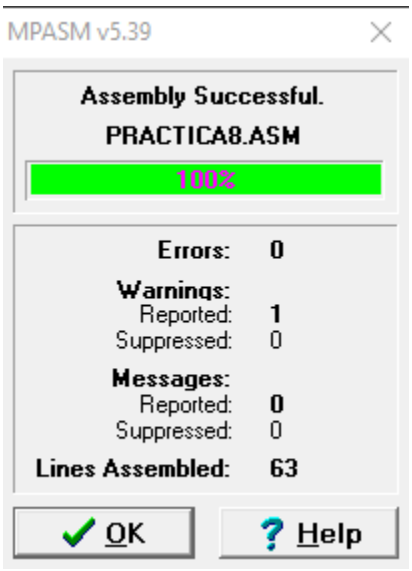
Este programa se incluyó en el PATH de Windows, por lo cual se puede ejecutar desde cualquier terminal.

Su sintaxis es sencilla, en este caso:

$ mpasmwin practica8.asm

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

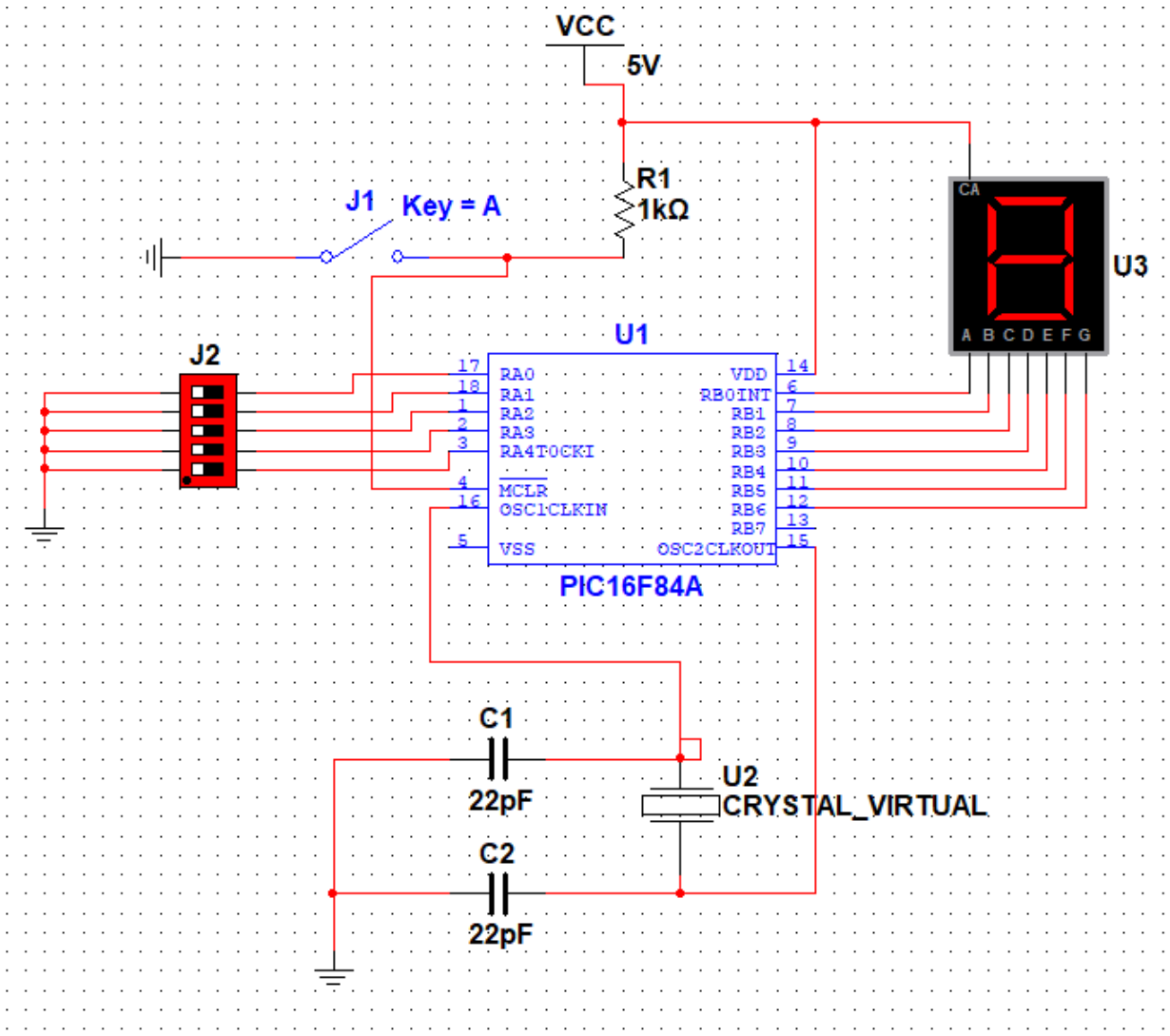
Esta es la salida del código cuando es correcta:



3 Simulación

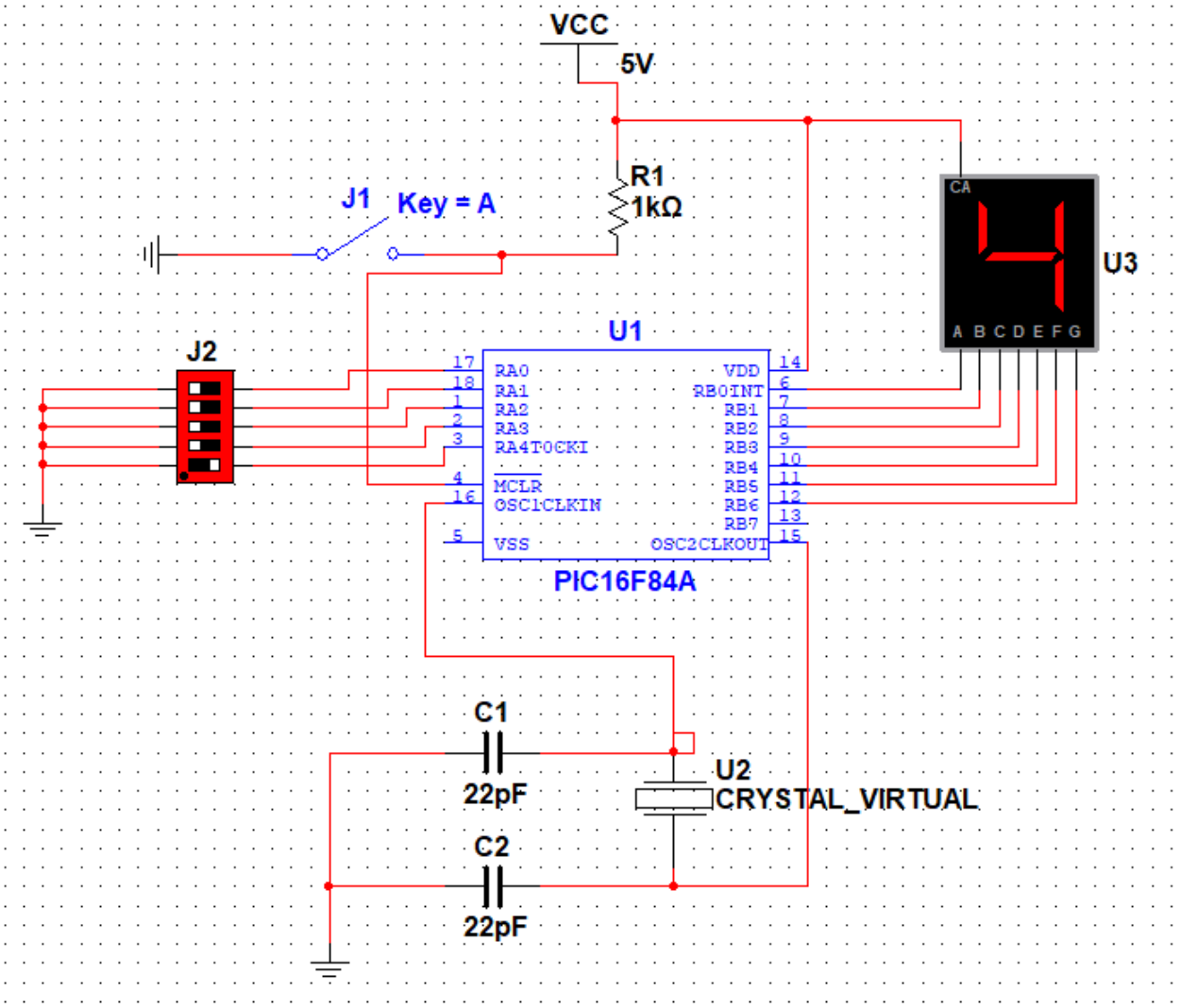
Antes que nada, **veamos el caso base**: todo el switch físico está apagado por lo cual PORTA está completamente apagado. Es como si se le hiciera un CLRF PORTA.

Ahora, vea que se agregó un CLTF PORTB al inicio del programa para resetear el PIC cada vez que se enciende. De otra forma iluminaría el mismo valor con el cual se apagó. Veamos

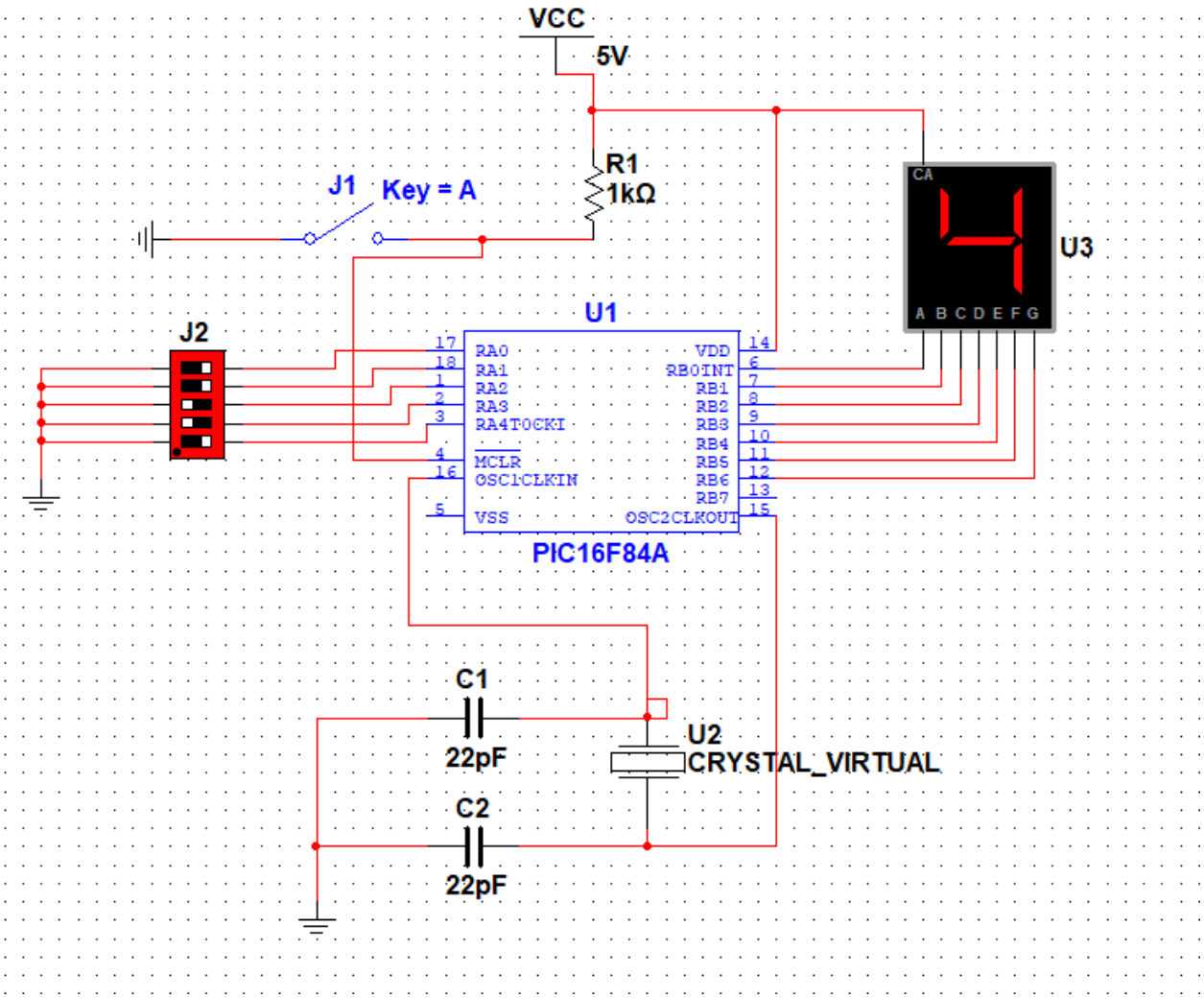


En el display se muestra un 8 dado que PORTB es 0000000, por lo cual todos los segmentos están encendidos.

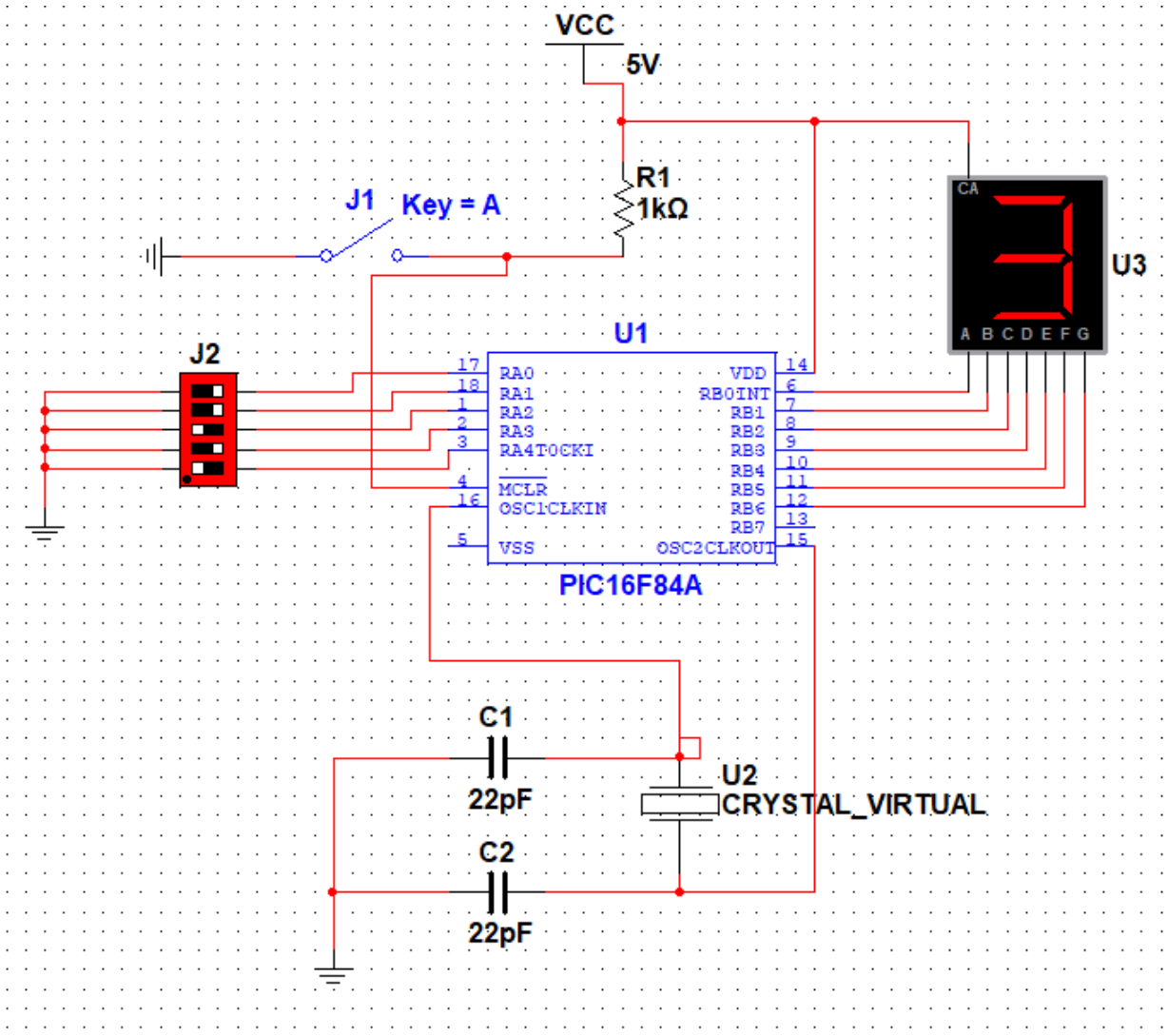
Ahora veamos el 4



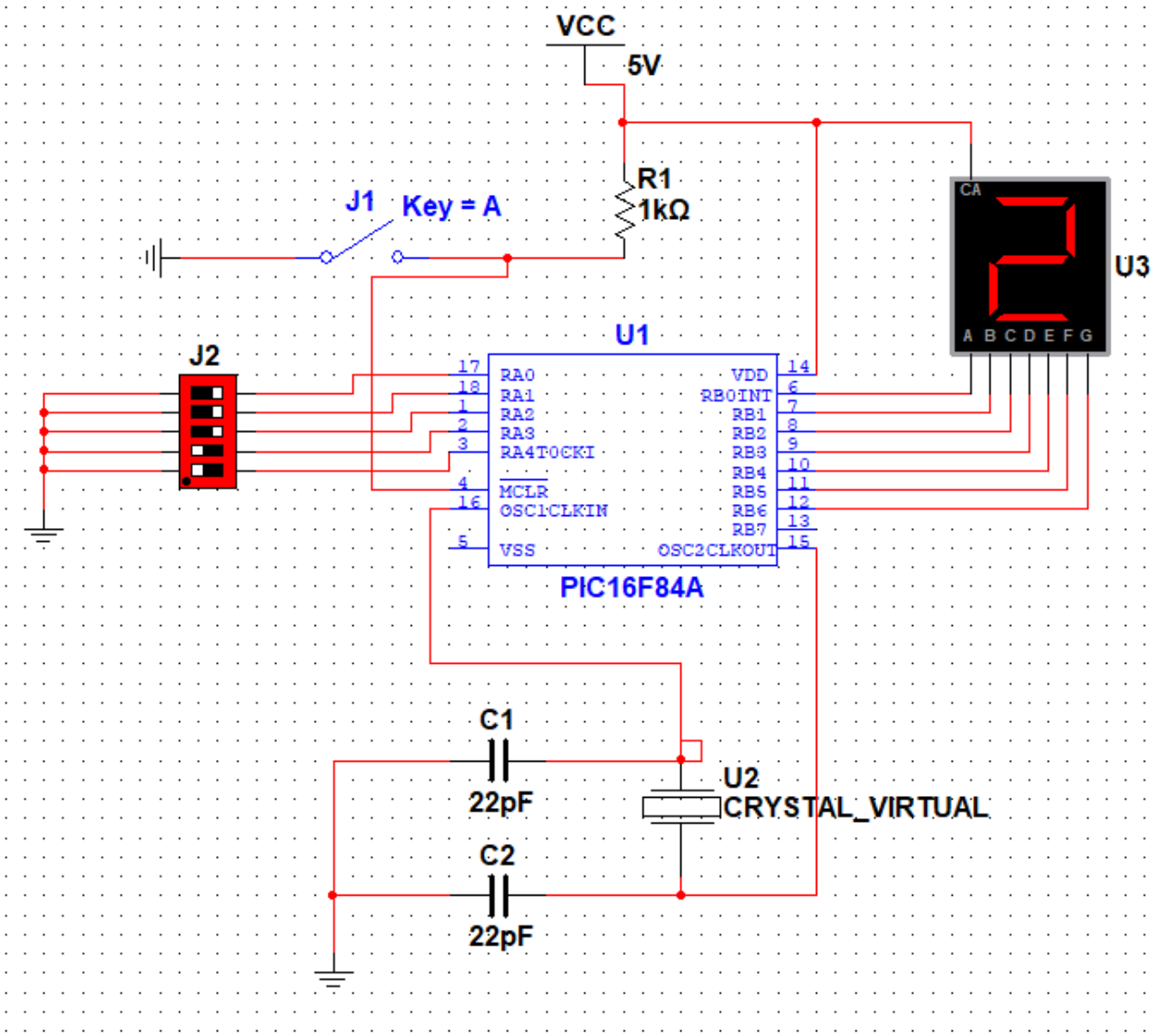
Como se puede ver RA4/T0CKI está activado, este es PORTA<4>. Puede observarse el 4 en pantalla. Ahora veremos que no importa si los demás bits de PORTA están encendidos, el 4 siempre se verá:



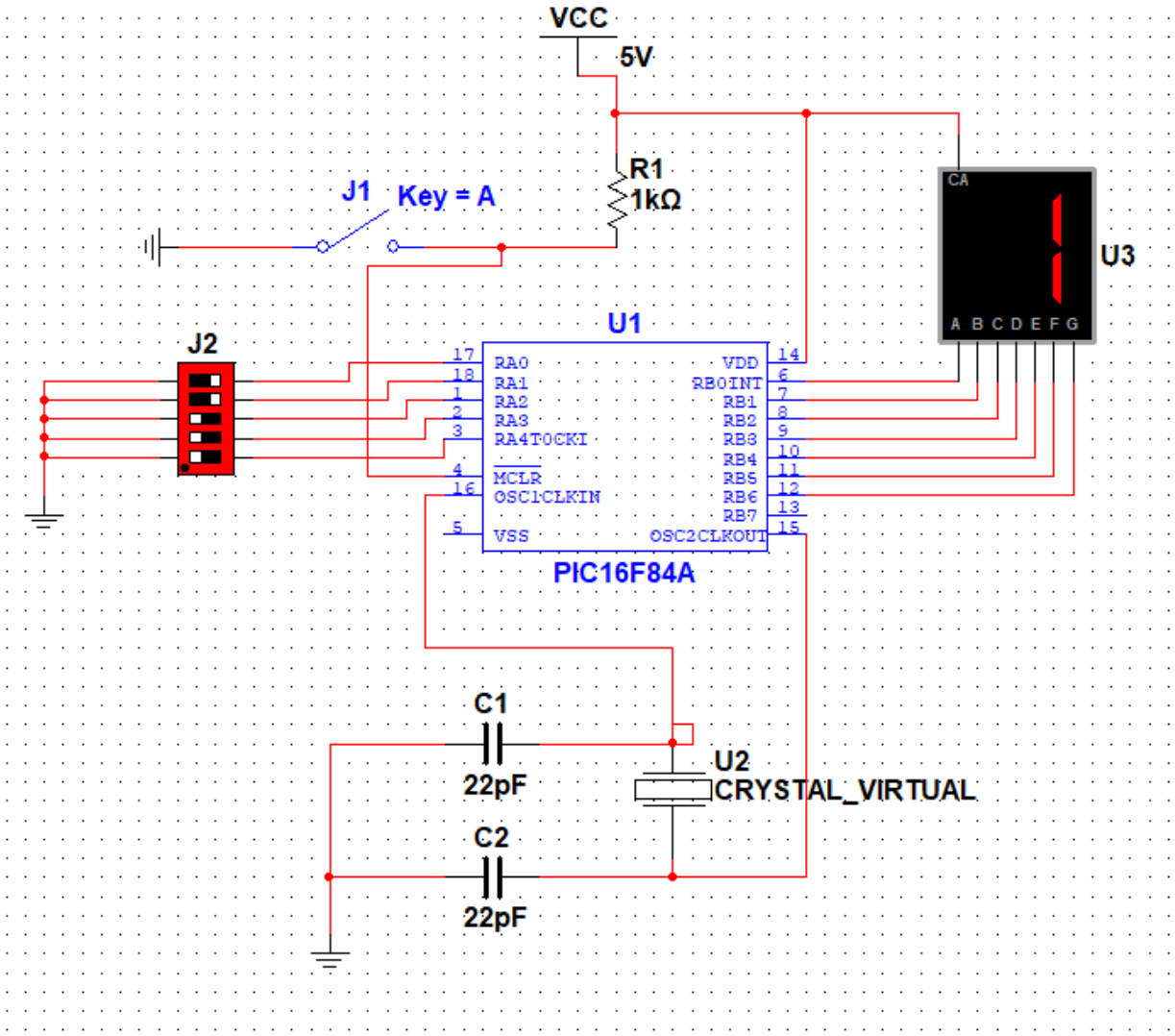
Note como en el switch se están activando PORTA<0> y PORTA<1>, pero no hay efecto sobre el display. Esto sucede porque siempre se salta a la rutina de CUATRO porque PORTA<4> sigue activado.



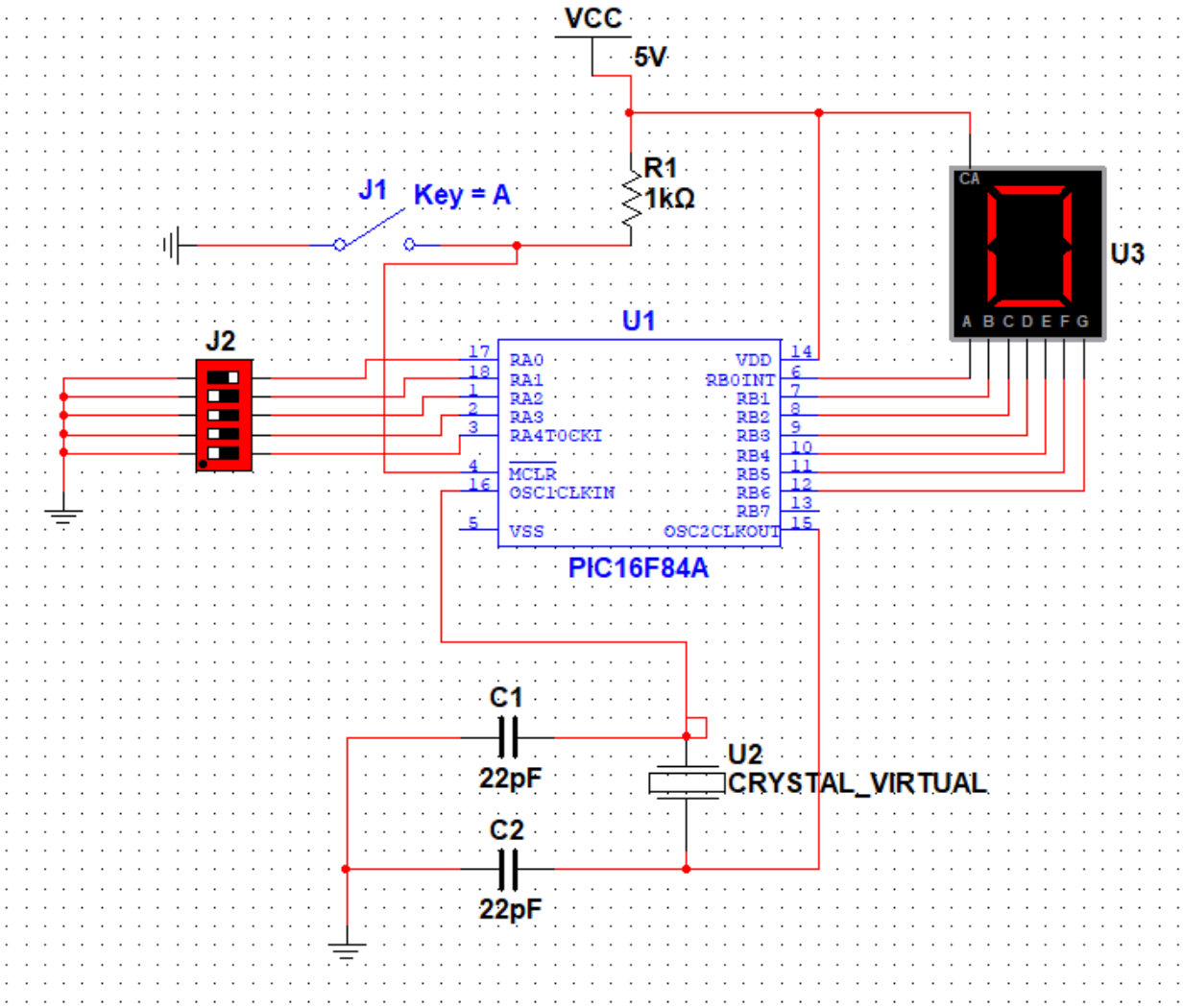
Note ahora que el switch se dejó igual para PORTA<0:1>, pero se apagó para PORTA<4> y se encendió PORTA<3> por lo cual ahora se ve un 3. Esto significa que se brincó a la rutina TRES.



Caso análogo para PORTA<2>.



Caso análogo para PORTA<1>. Se brinca a UNO.



Finalmente, el caso donde se muestra el 0, por lo cual PORTA<0>.

4 CONCLUSIONES

- El PIC posee instrucciones de bajo nivel del control de flujo en un programa
    - Con el uso de Bit Test y GOTO puede implementar un “if”
    - Con el uso de GOTO puede implementar ciclos al estilo “while”.
- Usa en conjunción sus PORTA y PORTB para recibir información y para mostrar información. Estos son configurados al inicio del programa para decidir cómo van a funcionar
- En gran parte de los casos se utiliza PORTA para entrada, dado que es un registro de 5 bits. La mayoría de los componentes electrónicos se benefician de 8 bits, como un display de 7 segmentos, una pantalla LCD, etc. Para esto es mejor PORTB
    - Esto NO quiere decir que PORTB no pueda usarse de entra, solo que es más conveniente usarlo de salida.