



FCEFyN - UNC

Guía de Ejercicios Electrónica Digital II

Año 2020

PRACTICO 1 – Introducción a Sistemas de Numeración y Códigos

Se pide resolver los ejercicios sin usar calculadora.

Ejercicio 1.1

Convertir al sistema binario y al hexadecimal los siguientes números decimales:

- a. 128_D
- b. 2048_D
- c. 555_D

Ejercicio 1.2

Convertir al sistema binario y al decimal los siguientes números hexadecimales:

- a. $7F3_H$
- b. 1560_H
- c. FFF_H

Ejercicio 1.3

Expresar en binario y hexadecimal el resultado de las siguientes operaciones:

- a. $FFFF_H - 3D00_H$
- b. $CA55_H - BB77_H$
- c. $1D99_H + F1C3_H$

Ejercicio 1.4

Utilizando complemento a dos y considerando un ancho de palabra de 1 byte, calcular el valor negativo de los siguientes números. Expresar el resultado en el sistema hexadecimal y en decimal.

- a. $7F_H$
- b. 63_H
- c. 33_H

Ejercicio 1.5

Considerando complemento a dos y un tamaño de palabra de 1 Byte:

- a. Indique cuál es el mayor número positivo que es posible representar.
- b. Indique cuál es el menor número negativo que es posible representar.

Ejercicio 1.6

Idem al ejercicio anterior pero utilizando un ancho de palabra de 2 Bytes (Word).

Ejercicio 1.7

Utilizando el código alfanumérico ASCII extendido expresado en hexadecimal, elaborar el string de números que imprime el siguiente mensaje.

"Esto es una prueba.
EDII"

Ejercicio 1.8

Escribir en código numérico BCD empaquetado y en BCD no empaquetado los siguientes números:

- a. 98_D
- b. 63_D
- c. 33_D

Ejercicio 1.9

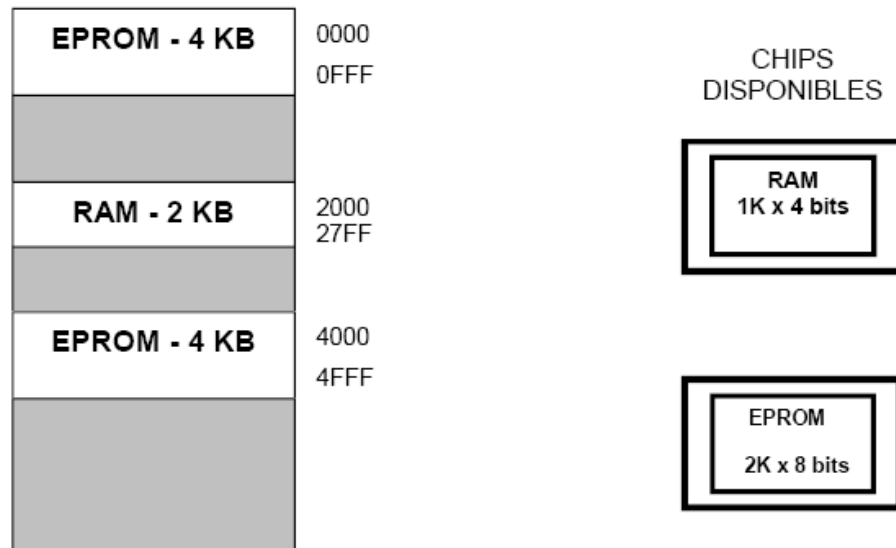
Escribir en código numérico Aiken y en código numérico exceso-3 los siguientes números:

- a. 52_D
- b. 30_D
- c. 74_D

PRACTICO 2 – Mapeo de Memorias

Ejercicio 2.1

Implementar un Banco de Memoria que responda al siguiente mapa:



Ejercicio 2.2

Implementar un Banco de Memoria que responda al siguiente mapa:

a partir de 0000h ⇒ □ 8 KB de EPROM

a partir de 3000h ⇒ □ 4 KB de RAM

a partir de 4000h ⇒ □ 16 KB de EPROM

a partir de A000h ⇒ □ 4 KB de EPROM

Se dispone de memorias RAM de 1K x 4 bits y EPROM de 4K x 8 bits.

Ejercicio 2.3

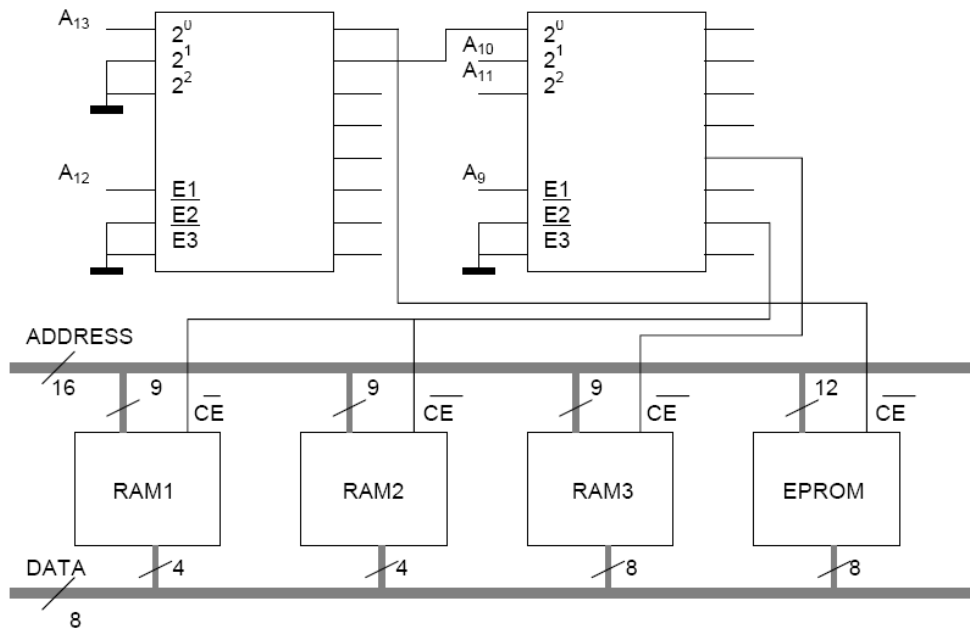
Realizar un mapeo que ubique 4KBytes de EPROM a partir de la posición 0 y 4KBytes de RAM a partir de 1000H. Se dispone de chips EPROMs de 1Kx4 bits y RAMs de 512 Bytes. Se solicita mapeo relativo y mapeo absoluto.

Ejercicio 2.4

Realizar un mapeo que ubique 8KBytes de EPROM a partir de la posición 0 y 8KBytes de RAM a partir de 2000H. Se dispone de chips EPROMs de 2Kx4 bits y RAMs de 1 KBytes.

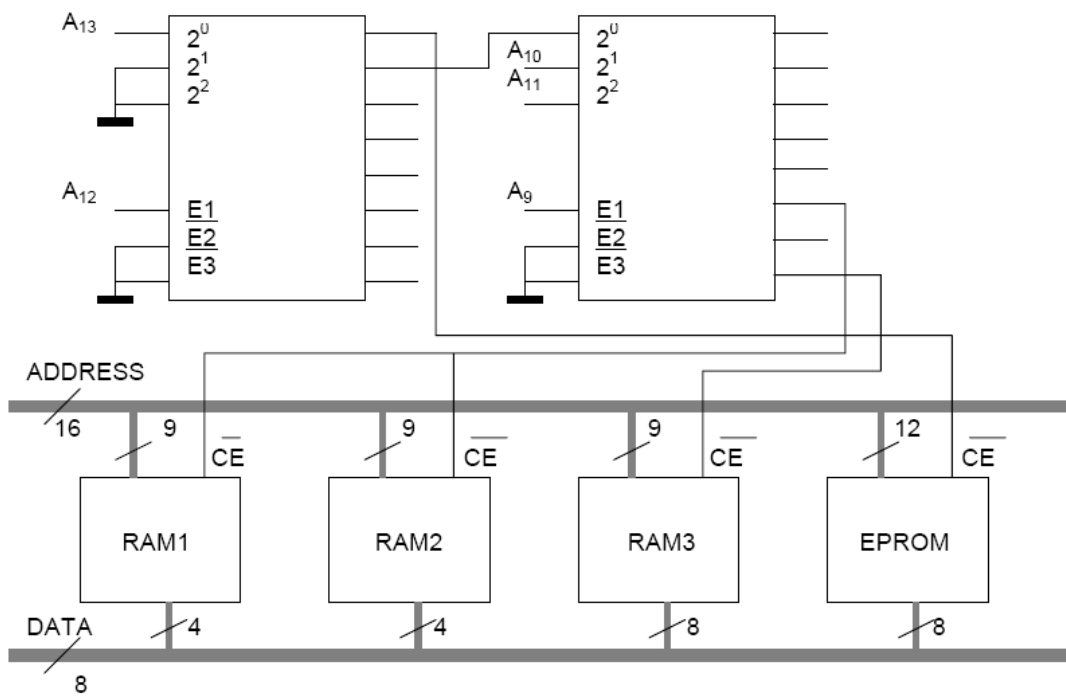
Ejercicio 2.5

Dado el siguiente circuito encuentre el mapeo correspondiente:



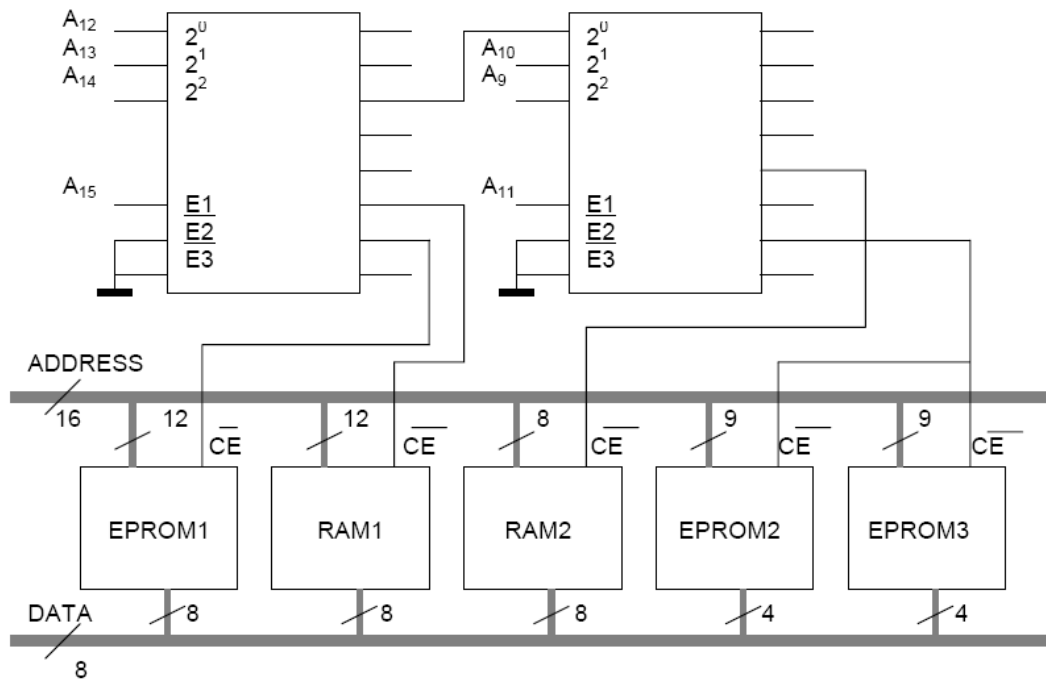
Ejercicio 2.6

Analice que sucede si se pide el mapeo del siguiente circuito. Discuta los motivos por los que no funciona al decodificar posiciones.



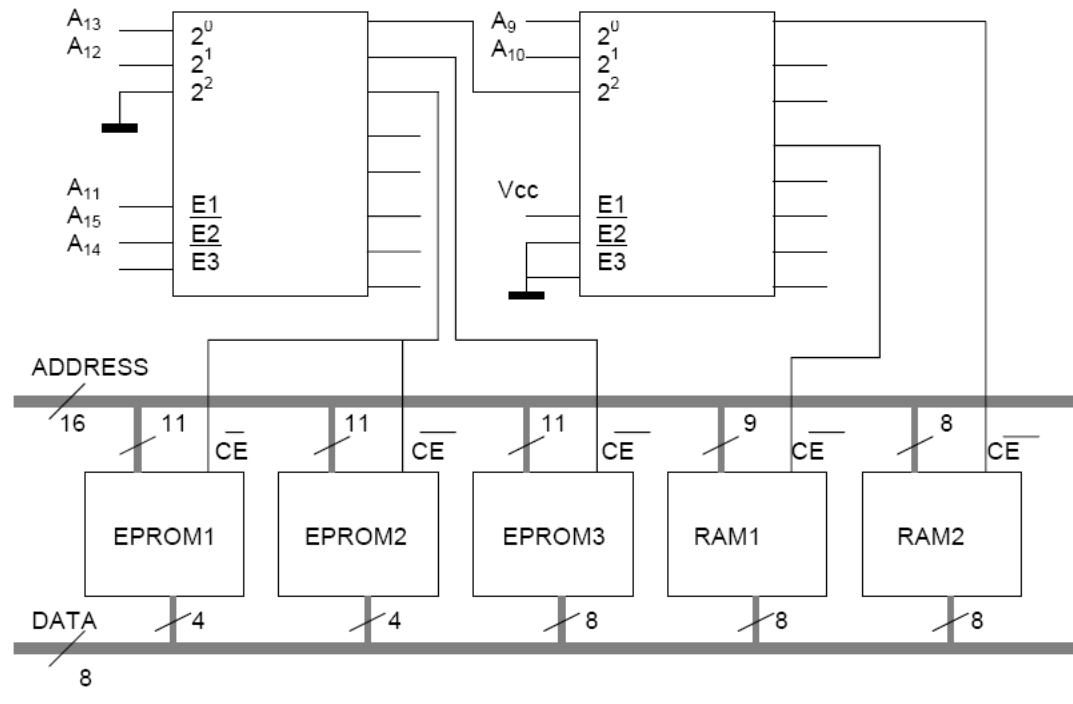
Ejercicio 2.7

Dado el siguiente circuito encuentre el mapeo correspondiente:



Ejercicio 2.8

Dado el siguiente circuito encuentre el mapeo correspondiente:



PRACTICO 3 - Programación Básica

Ejercicio 3.1

Se pide que responda a las siguientes preguntas orientadas a la arquitectura del PIC 16F887.

- a. Analice desde que dirección cada banco de memoria RAM está dedicado a Registros de Funciones Especiales – SFRs – y cuál es la última dirección ocupada por dichos registros.
- b. Analice de donde a dónde van los Registros de Usos Generales en cada banco.
- c. ¿Cuántos bytes tiene disponible para uso propio el programador en total, considerando todos los bancos?
- d. ¿Qué bits y de qué registro se usan para seleccionar cada banco en direccionamiento directo?
- e. ¿Qué bits y de qué registro se usan para seleccionar cada banco en direccionamiento indirecto?

Ejercicio 3.2

Escribir un programa que sume dos valores guardados en los Registros 21H y 22H con resultado en 23H y 24H

Ejercicio 3.3

Escribir un programa que resuelva la ecuación: $(A + B) - C$ (posiciones 21H, 22H y 23H)

Ejercicio 3.4

Escribir un programa que sume dos números de 16 bits A (20H 21H) y B (22H y 23H) y al resultado colocarlo en A.

Ejercicio 3.5

Escribir un programa que su ejecución demore un milisegundo (Cristal de 4MHz).

Ejercicio 3.6

Escribir un programa que su ejecución demore un segundo (Cristal de 4MHz).

Ejercicio 3.7

Escribir un programa que compare dos números A y B.

- Si son iguales, el resultado debe ser 0.
- Si $A > B$, el resultado debe ser la diferencia $A - B$.
- Y si $A < B$ el resultado debe ser la suma $A + B$.

Considere A en posición 30D, B en 31D y R en 32D.

Ejercicio 3.8

Escribir un programa para almacenar el valor 33D en 15 posiciones contiguas de la memoria de datos, empezando en la dirección 0x30.

Ejercicio 3.9

Escribir un programa que convierta un número de 8 bits, escrito en ASCII, en su equivalente en BCD no empaquetado. El número se encuentra en la posición 0x20.

Ejercicio 3.10

Escribir un programa que convierta un conjunto de números decimales codificados ASCII en su equivalente en BCD empaquetado. Los números codificados ASCII están en los Registros 20H a 30H.

Ejercicio 3.11

Escribir un programa que convierta un conjunto de Números Hexadecimales codificados ASCII en su equivalente en Hexadecimal no empaquetado. Los números codificados ASCII son 8 y están almacenados a partir del Registro 21H. Al resultado colocarlo a partir de la posición 31H

Ejercicio 3.12

Realizar un programa en Lenguaje Ensamblador que transforme 10 Bytes que contienen números BCD empaquetados a ASCII. Los números BCD están almacenados empezando en el Registro A0H y el resultado se almacenará a partir del Registro 1A0H.

Ejercicio 3.13

Redactar un programa que multiplique por 4 todos los números contenidos en los Registros que van de 50H a 5FH (ambos inclusive). Estos números tienen su nibble superior en 0 y el inferior contiene un número binario natural. El resultado se guarda en el mismo lugar.

Ejercicio 3.14

El código Gray es un código binario en el que dos números consecutivos difieren solamente en un bit. Escribir un programa que convierta un conjunto de números de 4 bits en su equivalente en código Gray. Estos números tienen su nibble superior en 0 y el inferior contiene un número binario natural. Son 20 números ubicados a partir del Registro 120H. Utilizar tabla.

Ejercicio 3.15

Escribir un programa que convierta un conjunto de números de 4 bits en su equivalente en código Exceso Tres. Estos números tienen su nibble inferior en 0 y el superior contiene un número binario natural BCD. Son 15 números ubicados a partir del Registro 40H y luego de modificarlos se deben guardar en el mismo lugar y con el mismo formato. No utilizar tabla.

PRACTICO 4 - Puertos de E/S Digitales

Ejercicio 4.1

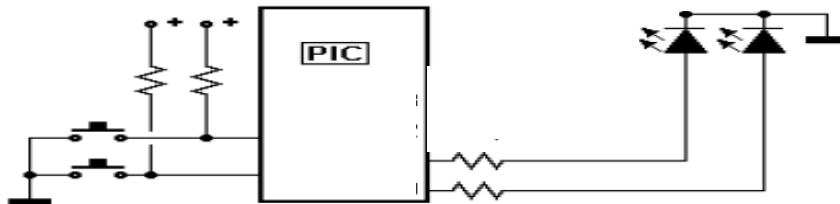
Se pide que responda a las siguientes preguntas orientadas a la arquitectura del PIC 16F887.

- ¿Qué bits y de que registros están dedicados a programar los puertos I/O trabajando en forma digital?
- ¿Cuántos pines de entrada comparten ingreso de señales digitales y analógicas?
- ¿Cuál es la máxima corriente que proporciona un pin de los puertos A, B y E?
- ¿Cuál es la máxima corriente que podemos sacar de cada pin si están entregando corriente simultáneamente todos los bits de los Puertos A, B y E?

Ejercicio 4.2

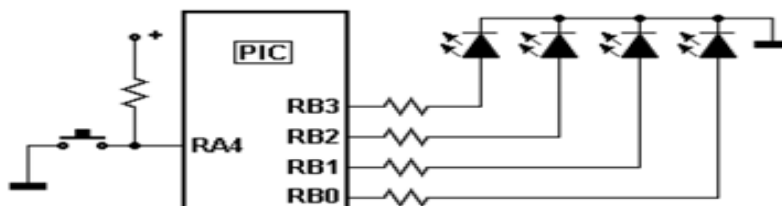
Escribir un programa que lea de dos pulsadores, conectados a RA4 y RB0, el estado lógico y lo muestre en dos de los LEDs, RB2 y RB3. La configuración es como se muestra en el esquema.

Adicionalmente indique los valores de las resistencias colocadas en cada puerto.



Ejercicio 4.3

Escribir un programa que cuente el número de veces que se pulsó la tecla conectada al terminal RA4 y que saque ese valor en binario natural por el Puerto B. Sólo se utilizarán los bits RB0 a RB3 que son los que tienen conectados diodos LED para su observación. Como consecuencia, el contador es de 4 bits: de 0 a 15.



Ejercicio 4.4

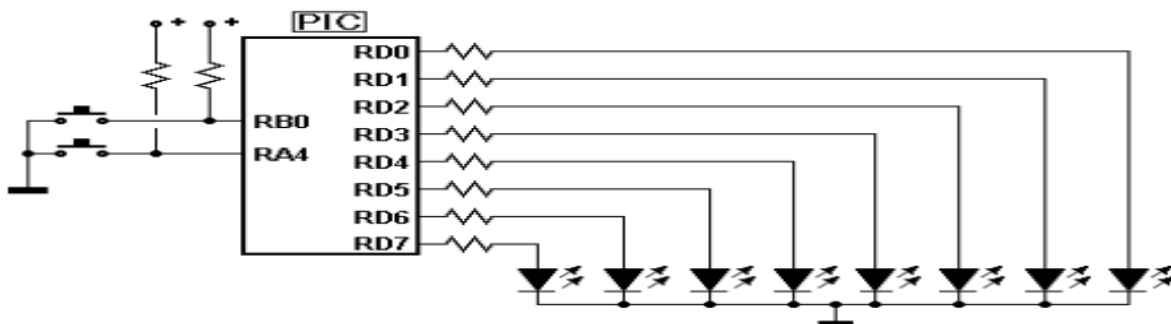
a) Escribir un programa que, dependiendo del estado de dos interruptores conectados a RA4 y RB0, presente en el puerto D diferentes funciones lógicas cuya tabla de verdad es:

RA4	RB0	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
0	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1	0	1
1	0	0	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0	0

b) Diseñe el hardware considerando que en cada pin del Puerto D se encuentra conectado un dispositivo lumínico que, cuando el puerto está en 0, se enciende. Cuando está encendido, cada dispositivo lumínico consume 100 mA.

Ejercicio 4.5

Se desea que al apretar el pulsador conectado a RA4 parpadeen, a una frecuencia de 0.5Hz, los 8 LEDs conectados en cátodo común a los 8 terminales del puerto D de un microcontrolador PIC 16F887. Dicho parpadeo se debe interrumpir durante unos instantes (3 segundos) si se aprieta el pulsador conectado al terminal RB0. Inicialmente, los LEDs están apagados. El oscilador es de 4MHz.



Ejercicio 4.6

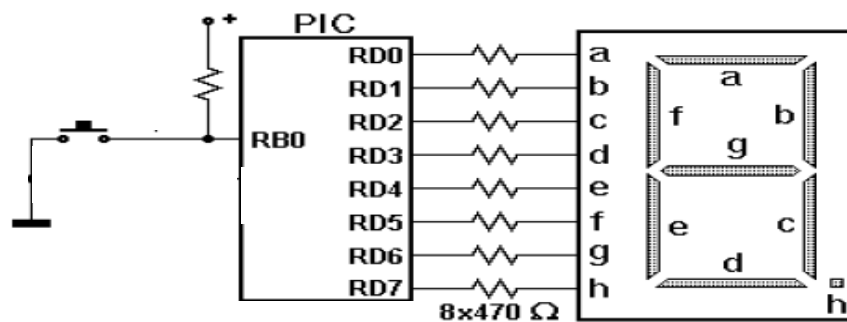
Considerando el mismo hardware del Ejercicio 4.4 escribir un programa que ilumine los LEDs conectados al puerto D según las siguientes especificaciones:

- Inicialmente (al salir del reset) aparecen parpadeando los LEDs.

- Si se aprieta el pulsador conectado al pin RA4 se produce un desplazamiento de derecha a izquierda.
- Si se vuelve a apretar el pulsador RA4 cambia el sentido de desplazamiento. Es decir, si se desplazaba de derecha a izquierda ahora lo hará de izquierda a derecha y viceversa.
- El desplazamiento debe comenzar al soltar el pulsador.
- En cualquier momento, al apretar el pulsador conectado al pin RB0, se vuelve al parpadeo inicial.

Ejercicio 4.7

Realizar un programa en Lenguaje Ensamblador que cuente de 0 a 9 indefinidamente. Cada número permanecerá encendido 1 seg (retardo por software). El conteo iniciará en 0 al apretarse el pulsador y se detendrá al volver a pulsarlo en el valor que esté la cuenta. Se muestra el esquema del hardware sobre el que correrá el programa. El oscilador es de 4MHz.



PRACTICO 5 - Interrupciones

Ejercicio 5.1

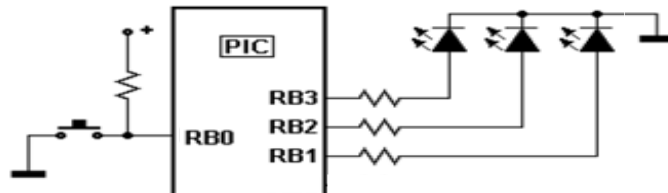
Se pide que responda a las siguientes preguntas orientadas a la arquitectura del PIC 16F887.

- ¿Qué bits y de qué registros están dedicados a programar las interrupciones?
- El micro debe responder a una interrupción por flanco ascendente ¿Puedo conectarla a RB7?
- ¿Qué diferencia hay entre una interrupción por RB0 y otra por RB2?
- ¿Cómo está configurado el stack y para qué se utiliza?

Ejercicio 5.2

Escribir un programa que prenda un LED que se va desplazando cada vez que se pulsa la tecla conectada a RB0. Al pulsar por primera vez la tecla se enciende el LED conectado a RB1 y al llegar a RB3 vuelve a RB1 y así indefinidamente. El programa principal no realiza tarea alguna y todo se desarrolla dentro de la subrutina de interrupción.

¿Cómo reorganizaría el software y el hardware si usara resistencia de pull-up interna en RB0?

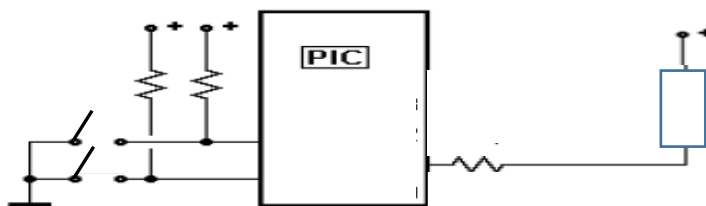


Ejercicio 5.3

Escribir un programa que lea de dos teclas conectadas a RB1 y RB2 y actúe sobre un relé conectado en RB3.

- Si están ambas teclas abiertas el relé está abierto.
- Si RB2 está abierta y RB1 cerrada se activa el relé por 1 minuto.
- Si RB2 está cerrada y RB1 abierta se activa el relé por 2 minutos.
- Si están ambas cerradas se activa el relé por 3 minutos.

Cualquier cambio en las teclas mientras esté activado el relé no debe modificar la salida.

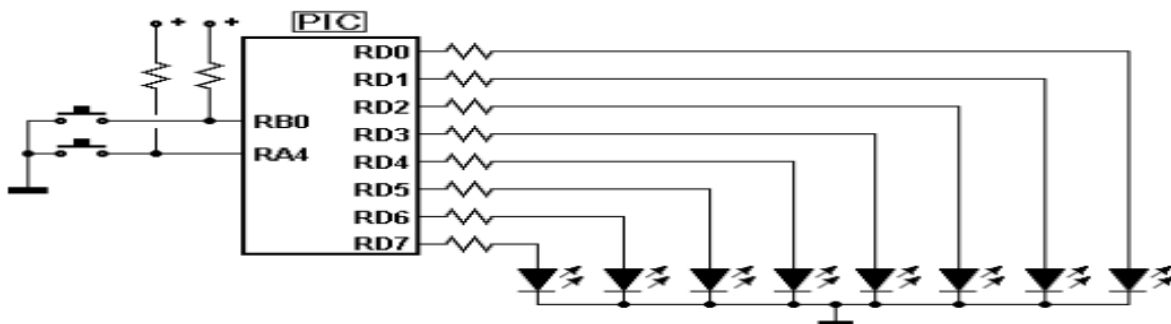


Ejercicio 5.4

Escribir un código en assembler que realice una interrupción por RB cuando se realice un cambio de nivel en cualquiera de los puertos RB4 a RB7. En el servicio a la interrupción (ISR) generar un retardo de 100 mseg e incrementar un contador solo cuando la interrupción es debida a un estado en alto. Suponer un reloj de 4 Mhz.

Ejercicio 5.5

Se desea que al apretar el pulsador conectado a RA4 parpadeen, a una frecuencia de 0.5Hz, los 8 LEDs conectados en cátodo común a los 8 terminales del puerto D de un microcontrolador PIC 16F887. Dicho parpadeo se debe interrumpir durante unos instantes (3 segundos) si se aprieta el pulsador conectado al terminal RB0. Inicialmente, los LEDs están apagados. El oscilador es de 4MHz.



Ejercicio 5.6

Por RB0 ingresa una onda cuadrada como se muestra en la figura y actúa sobre un relé conectado a RB3. Se pide que al salir del reset el relé permanezca apagado por 1 minuto y luego conectado por 2 minutos y así sucesivamente. Utilizar flanco ascendente para activar la interrupción.

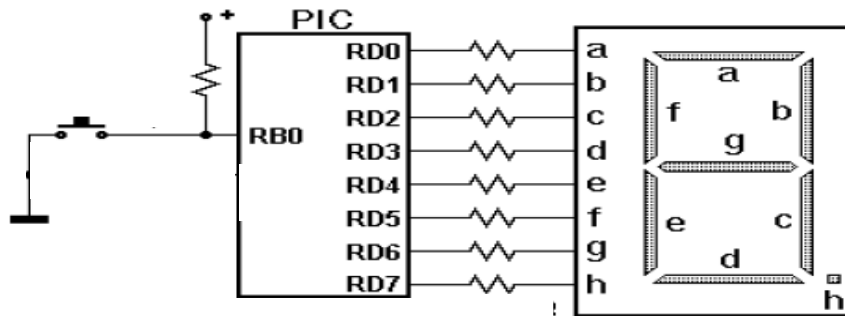


Ejercicio 5.7

Utilizando interrupciones por RB0, muestre mediante un display de 7 segmentos el número de veces que sucedió un flanco descendente.

Se pide:

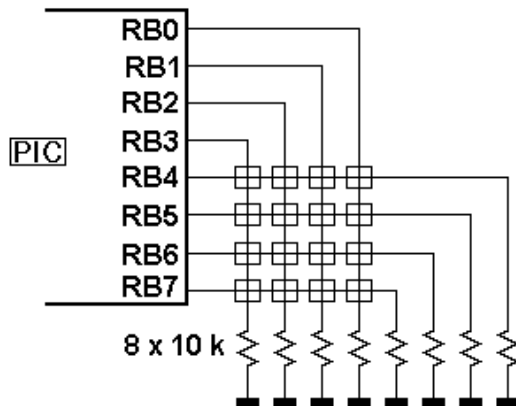
- El programa en assembler del programa principal y de la rutina de interrupción con resistencia de pull-up interna.
- El circuito completo considerando que el display es cátodo común y cada display consume 20 mA.
- El circuito completo considerando ahora que el display es ánodo común.
- Si cada display consume 50 mA, ¿Cómo quedarían los circuitos para los casos b y c?



PRACTICO 6 – Display y Teclado

Ejercicio 6.1

Realizar un programa que obtener el código ASCII de la tecla que se pulsa en un teclado estándar conectado al puerto B de un microcontrolador PIC 16F887, como indica la figura. El valor ASCII de la tecla se guardará en el Registro 30H. Se pide resolución utilizando el método polling.



TECLADO

1	2	3	4
5	6	7	8
9	A	B	C
D	E	F	0

Ejercicio 6.2

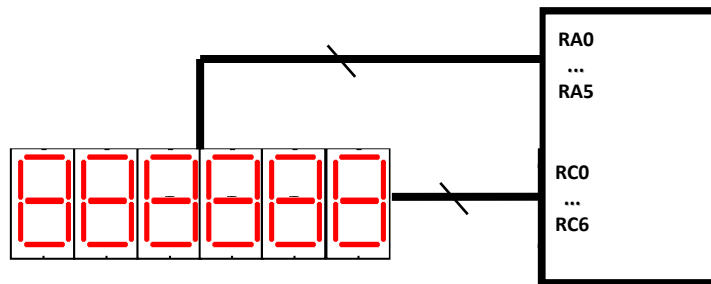
Con la configuración del ejercicio anterior considere ahora que el micro tiene un buffer circular de teclado de 6 teclas a partir de la dirección 30H.

Se pide:

- Que el buffer sea llenado con un byte por tecla que tenga el nibble superior igual a 0 y el nibble inferior sea igual al número que representa la tecla.
- Que el buffer sea llenado con un valor igual a número que representa la tecla para ser mostrada en un display 7-segmento ánodo común y con el b7 igual a 0.

Ejercicio 6.3

Según el esquema que se muestra y utilizando técnicas de multiplexado de display desarrolle un programa que muestre '000005' al iniciarse el programa. El número 5 se desplazará hacia la izquierda una posición cada 1 minuto y su lugar será reemplazado por 0 ('000050', '000500', etc.). Al llegar a '500000' se repetirá la rutina indefinidamente.



Ejercicio 6.4

En el esquema del ejercicio anterior escribir ahora una rutina que muestre los valores de un buffer circular de 6 bytes que inicia en 30H. El Byte de la dirección B0H se muestra en el display extremo de la izquierda. Considere que los valores a mostrar están codificados en ASCII.

Ejercicio 6.5

Realice el hardware necesario para manejar 6 displays multiplexados 7-segmentos, con un consumo por segmento de 20 mA, en cada una de los siguientes casos:

- a) Display ánodo común. Lógica positiva.
- a) Display ánodo común. Lógica negativa.
- a) Display cátodo común. Lógica positiva.
- a) Display cátodo común. Lógica negativa.

Ejercicio 6.6

Idem ejercicio anterior pero considerando que cada uno de los segmentos de los displays consumen 60 mA.

Ejercicio 6.4

Conectar a un PIC un teclado matricial 4x4 y 6 display 7 segmentos. Considere que los displays son ánodo común y los segmentos se activan con 0. Se pide:

- a. Hardware completo.
- b. Subrutina de teclado y de display por interrupción de manera que funcionen independientes del programa principal.

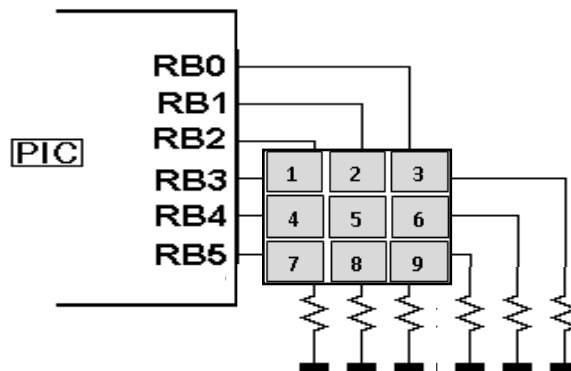
Ejercicio 6.5

Idem ejercicio anterior pero considerando que los displays son cátodo común y los segmentos se activan con 1. Desarrolle sólo lo que cambia con respecto al otro ejercicio.

PRACTICO 7 – Display y Teclado con Interrupciones

Ejercicio 7.1

Realizar un programa que obtenga el valor de la tecla y la guarde en código BCD empaquetado. Debe colocarla en un buffer circular de 32 nibbles en 20H. La resolución de cuál es la tecla apretada y su almacenamiento se resuelve integralmente dentro de la interrupción.



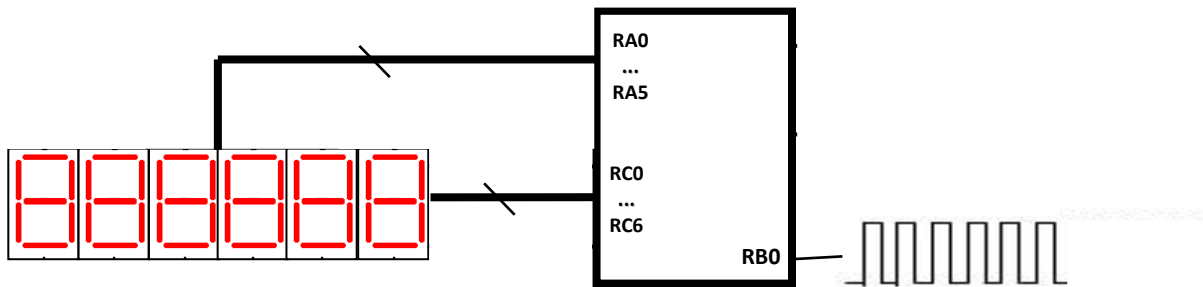
Ejercicio 7.2

Con la configuración del ejercicio anterior considere ahora que el micro tiene dos buffers circulares de teclado de 6 teclas cada uno. El primer buffer inicia en 60H y allí se coloca la tecla codificada en ASCII. El segundo inicia en 70H y se almacena la tecla codificada en 7-segmentos lógica negativa. El teclado funciona por interrupciones.

Ejercicio 7.3

Según el esquema que se muestra y utilizando técnicas de multiplexado de display desarrolle un programa que cuente desde '000000' al iniciarse el programa. El contador avanza una cuenta cada segundo y al llegar a '999999' vuelve a '000000' y así indefinidamente. Los displays y la cuenta se manejan por la interrupción en RB0 que se activa por flanco descendente.

Se pide además calcular la frecuencia a la que se debe colocar el generador de onda cuadrada que ingresa por RB0.



Ejercicio 7.4

Con el esquema del ejercicio anterior y agregando un pulsador al pin RB1 realice ahora un cronómetro. RB1 oficia como start/stop del cronómetro y el valor se muestra en los 6 dígitos con el formato: HHMMSS.

Si el equipo sólo hiciera esto, ¿qué valor de clock seleccionaría?

Ejercicio 7.5

Realice el hardware y el software necesario para manejar simultáneamente 4 displays 7-segmentos cátodo común, con técnica de multiplexado, y un teclado matricial de 3x4 teclas. Utilice la menor cantidad de puertos de E/S posible y coloque un generador de onda cuadrada, con la frecuencia conveniente, para manejar la interrupción que ingresa por RB0.

Ejercicio 7.6

Con el hardware del ejercicio anterior escriba un programa que maneje un cronómetro con la siguiente rutina:

1. El teclado tendrá los números 0 a 9, M y S. M y S se utilizarán para indicar el número de minutos y segundos en el cual el cronómetro parará.
2. Al iniciarse el programa todos los dígitos están en 0.
3. Al apretar S el programa tomará las dos teclas numéricas siguientes como el máximo de minutos hasta el cual llega. Al apretar S sucederá lo mismo pero para los minutos.
4. Al apretar 0 comienza la cuenta que se detiene en el valor preseteado.
5. Al presionar 9, ya sea que esté detenido o contando, el cronómetro se pone en 0.

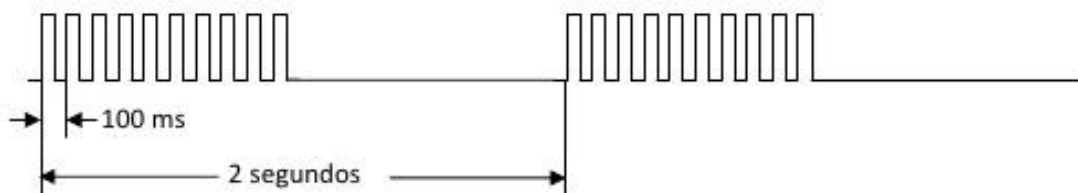
PRACTICO 8 - Módulo Temporizador

Ejercicio 8.1

Reescribir un programa que cumpla la misma función que el pedido en el Ejercicio 4.7 pero TMR0 se utiliza para fijar el tiempo en que cada dígito permanecerá encendido (1 segundo). Considere esta vez que el clock es de 8 MHz.

Ejercicio 8.2

Genere la señal que se muestra utilizando interrupciones y TMR0. Considere que el clock del microprocesador es de 8 MHz.



Ejercicio 8.3

Escribir un programa que cuente eventos aleatorios. Cada evento genera una señal que ingresa convenientemente al PIC. Cuando la cuenta supere los 2048 pulsos dejará de contar y comenzará a parpadear un LED con una frecuencia de 1 Hz. Utilice el TMR1 y considere una frecuencia de clock de 4 MHz.

Ejercicio 8.4

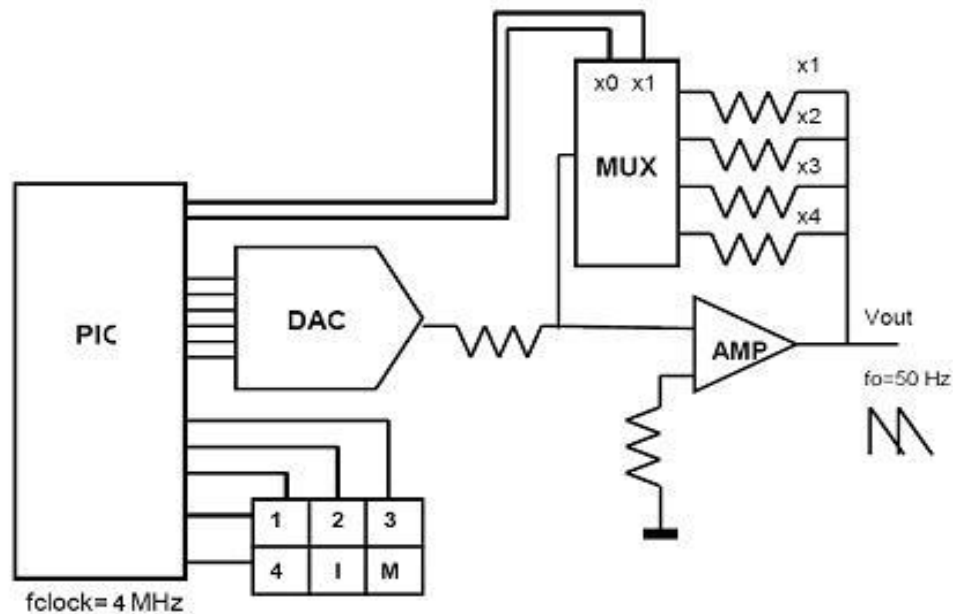
Programa el microcontrolador para que luego de realizar una rutina de programa entre en modo sleep. Saldrá de ese modo cuando ingresen 20 pulsos consecutivos por uno de sus pines.

Ejercicio 8.5

Realice un programa que cuente las botellas que pasan por una cinta de producción. A medida que avanza la cuenta se mostrará su valor en dos displays 7 segmentos. Cada 80 botellas contadas enviará un pulso de 1 segundo a un relé conectado a uno de los pines y volverá a reiniciar la cuenta en 0.

Ejercicio 8.6

Sobre el Hardware que se plantea en el esquema, redactar un programa en assembler que genere las formas de onda que se indican en la tabla.



Los valores de tensión de referencia del conversor son 0V y 4V. Según la tecla que se presione, la salida será:

Tecla	Función
1	Diente Sierra x1
2	Diente Sierra x2
3	Diente Sierra x3
4	Diente Sierra x4
I	0 V
M	4 V

El microprocesador funciona con un clock de 8MHz.

La rutina que debe cumplir es la siguiente:

1. El tipo de salida se preseleccionará con 1, 2, 3, 4 y M.
2. La salida permanecerá en 0 hasta que se presione I.
3. Cuando se esté generando, la salida se pondrá en 0 cuando se presione cualquier tecla, excepto Inicio.
4. Si se presionan 2 teclas, será válida la última.
5. Recién comenzará a generar el nuevo tipo de onda preseleccionado cuando se presione I.

Ejercicio 8.7

Generar un sonido en el pin RB1 en los tonos de 1000Hz, 500Hz, 250Hz y 125Hz. Cada vez que llega una interrupción externa a RB0 cambia de frecuencia comenzando con 1000Hz, 500Hz, 250Hz, 125Hz, 1000Hz, 500Hz, 250Hz, Realizar el hardware y software completo, con los comentarios correspondientes. (Cristal del PIC 8MHz).

PRACTICO 9 - Módulo Conversor A/D

Ejercicio 9.1

Un sensor de temperatura entrega un valor de tensión que va desde 0 a 4 V. El PIC tiene un display 7 segmentos en el que se mostrará "a" hasta que llegue a 2 V, de 2 a 3 V mostrará "b" y de 3 a 4 V se visualizará "c". El conversor debe usar todo su rango dinámico. La muestra se tomará una vez por minuto. Todo el proceso de medir y mostrar display se lleva adelante en la interrupción y el programa principal realiza otras tareas (o sea se mantiene en un bucle sin hacer nada a los fines de la resolución del ejercicio).

Ejercicio 9.2

El PIC maneja un Voltímetro de tensión continua que mide valores de 0 a 5 V. El resultado de la medición se muestra en tres displays de 7 segmentos. El conversor debe entregar la máxima resolución posible.

Se pide:

- a. Hardware completo.
- b. Software completo incluido la subrutina de display.

Ejercicio 9.3

En el ejercicio anterior suponga ahora que la tensión ahora es alterna. ¿Qué modificaciones de hardware debería realizar para que funcione?

Ejercicio 9.4

Un sistema comandado por un PIC 16F887, cuyo oscilador es de 4 MHz, tiene conectado un display de 3 dígitos de 7 segmentos (ánodo común), un pulsador en RB0 que es activo por bajo y un sensor de temperatura de un horno. Este sensor tiene una salida analógica de 0 a 3V que, una vez digitalizada, representa una temperatura de 0 a 511 grados centígrados. El programa leerá la temperatura cada vez que se oprima el pulsador y en el display se mostrará el valor leído en el conversor ADC. El proceso mostrará inicialmente 0 y será cambiado por el valor leído en el ADC. Diseñar software y hardware completo para que cumpla con lo requerido.

PRACTICO 10 - Módulo Transmisión Serie

Ejercicio 10.1

Un PIC recibe bytes por su puerto serie asincrónico a una velocidad de 9.600 baudios y los muestra en un display 7 segmentos de 6 dígitos. Al inicio el display estará apagado y los valores recibidos serán mostrados con lógica FIFO de izquierda a derecha. Los bytes contienen número hexadecimales en ASCII.

Ejercicio 10.2

Un PIC tiene conectado un teclado con la disposición que se muestra en la figura. Se pide que cada dos teclas apretadas empaquete el valor en un byte, o sea, si se apretó la secuencia 4, F, 2, 4, 7 y B se conformen los bytes de información F4, 42 y B7. Cada 2 teclas sucesivas, cuando completa el byte, lo transmite por la EUSART en configuración serie asincrónica. El clock del microcontrolador es de 4 MHz y la velocidad de transmisión de 2400 baudios.

1	2	3	4
5	6	7	8
9	A	B	C
D	E	F	0

Ejercicio 10.3

Considere que tiene una impresora que funciona con una conexión serie RS232 conectada a un PIC. La recepción y transmisión de datos se realiza a una velocidad de 9600 baudios. El PIC tiene un oscilador de 4 MHz. Se pide:

1. Inicialice la EUSART para transmitir y recibir a esa velocidad con el menor error posible.
2. El PIC queda a la espera de recibir el caracter *enquiry* <ENQ> de la impresora para comenzar a enviar el texto a imprimir.
3. Luego de recibir *enquiry* envía el texto **PRINT TEST** a la impresora seguido del caracter *carry return* <CR> seguido *end of text* <EOT>.
4. Si le responden con un *acknowledgement* <ACK>, finaliza el programa.
5. Si le responde con un *no acknowledgement* <NAK> vuelva al principio y retransmita la frase

Ejercicio 10.4

En el Ejercicio 5.4 cada vez que lee el valor del Conversor A/D deberá convertir el valor de cada uno de los tres dígitos a ASCII y transmitirlos de menor a mayor peso del dígito por el puerto serie. La velocidad de transmisión será de 2.400 baudios y el clock de 4 MHz. En cada caso, antes de enviar los 3 bytes de lectura, se debe enviar el caracter # y al finalizar el caracter EOT.