

### BLOQUE 3: PROGRAMACIÓN EN ENSAMBLADOR APLICADA A MAQUETAS DE SIMULACIÓN. Curso 2007/08.

El objetivo de la práctica es la toma de contacto con la máquina MP-88 de la casa ALECOP y la simulación de programas mediante código en ensamblador, utilizando las maquetas de accesorios conectadas al módulo BP-88.

Como complemento al estudio del microprocesador 8088, se ha realizado este manual de usuario donde se incluyen programas de ejemplo y el uso de las tarjetas de simulación de la serie 88.

## 1. INTRODUCCIÓN

MP-88 es una máquina versátil que puede desempeñar numerosas funciones como:

- Ejecutar programas y juegos escritos en código ensamblador
- Cargar datos desde/en cinta
- Examinar el estado del sistema comprobando contenidos de memoria y registros
- Comunicarse con el sistema a través del programa monitor incorporado

Para usar la máquina MP-88 debe conectar el cable de alimentación a la red y pulsar el interruptor general. Si se ha conectado adecuadamente la máquina, se mostrarán en la pantalla los siguientes mensajes:



Fig.1. Sistema principal MP-88 donde se conectará el cable de alimentación, el teclado y el módulo de expansión BP-88

MP-88/87  
Monitor versión 1.2

MP-88/87  
© 1991 ALECOP

MP-88/87  
CHEQUEO DE RAM █

MP-88/87  
CHEQUEO DE ROM █

> █

En este instante, la máquina MP-88 está preparada para recibir instrucciones en ensamblador. En caso contrario, si se señalase algún mensaje de error, sería necesario comprobar que el conexionado realizado es correcto (alimentación o teclado).

## 2. PRIMER PASO: EJEMPLO DE PROGRAMA

En el programa en lenguaje ensamblador a desarrollar como ejemplo, se calcula la suma de todos los números enteros desde el 1 hasta el 100, mostrando el resultado en la pantalla.

A 0080:0 ←

Comando del monitor que solicita entrar en el ensamblador. La dirección 0080:0 le dice al sistema dónde comienza el programa. Hay que confirmar con la pulsación de INTRO.

0080:0000	MOV AX,0
0080:0003	MOV CX,64
0080:0006	ADD AX,CX
0080:0008	LOOP 0006
0080:000A	INT 10
0080:000C	INT 7 ←
0080:000E	←

Cada vez que se introduce una instrucción y se pulsa la tecla INTRO, el ensamblador la traduce a código máquina y le muestra la dirección que tendrá la siguiente instrucción (sombreado en gris).

Las direcciones de memoria dependerán del número de bytes que ocupe cada instrucción junto con sus operandos.

Debido a que no se pueden usar etiquetas directamente, la instrucción LOOP realiza un salto al desplazamiento 0006 del segmento 0080.

La instrucción INT 10 significa que mostrará el resultado de la operación por la pantalla. Para ello se ejecuta una rutina de interrupción almacenada en la memoria ROM, cuya función es escribir el contenido del registro AX sobre la pantalla.

La instrucción INT 7 transfiere el control del sistema desde el programa al monitor, debiendo todo programa terminar con esta instrucción.

> █

Esta línea se visualizará en pantalla siempre que se encienda el sistema o siempre que se haya ejecutado un programa. Representa un estado de espera en el que se pueden introducir los comandos del monitor.

Una vez que el programa está cargado, se deberá ejecutar mediante el siguiente comando a teclear:

> G 80:00 ←

Apareciendo en pantalla el siguiente mensaje:

13BA  
Programa terminado  
> █

Este mensaje nos informa de que el programa ha sido ejecutado y que el resultado es 13BA en código hexadecimal.

Todo el contenido de la memoria RAM (como el programa de ejemplo) desaparece cuando se apaga el sistema MP-88.

### 3. HARDWARE DEL SISTEMA

El MP-88 se compone de la CPU, la memoria y otros dispositivos, donde la comunicación se realiza a través del bus del sistema.

La CPU es el corazón del microordenador que controla todas las operaciones que se realizan. El MP-88 tiene el microprocesador 8088 de INTEL cuya frecuencia de trabajo es de 4,77 MHz (tiempo de ciclo de 0,21 $\mu$ s).

El MP-88 incluye memoria ROM, RAM y un área de expansión para el usuario, desde la dirección 00000h hasta la FFFFFh:

- 2 pastillas ROM de 16 Kbytes y se utilizan para almacenar el software monitor y del ensamblador.
- 3 zócalos de pastillas RAM de 2 Kbytes y se usan como área de trabajo para programas y datos

La pantalla del sistema MP-88 es una matriz de puntos LCD de 2 líneas por 20 caracteres. Sin embargo el software monitor soporta una visualización lógica de 24 líneas, pudiendo mostrar cualquier sección de 2 líneas de las 24 de la pantalla.

El equipo dispone de un teclado estándar compacto para PC. De todas las teclas disponibles, las más utilizadas son:

- Todas las teclas normales como letras números y caracteres especiales
- Combinación de las teclas ALT+A o ALT+Z (flecha arriba o flecha abajo): sirven para desplazar hacia arriba o hacia abajo a través de la pantalla de 24 líneas
- La tecla BORRAR borra el último carácter escrito
- La tecla INTRO define el fin del carácter escrito

El MP-88 lleva en su interior una fuente de alimentación de 5V/1A y  $\pm$ 12V/200mA. Esta fuente será suficiente para alimentar el sistema principal y cualquiera de las tarjetas de la serie-88 a través del módulo de expansión BP-88.



Fig.2. Módulo de expansión BP-88 donde se conectarán las tarjetas de simulación de la serie-88

### 3.1. Módulos de expansión y tarjetas de la serie 88

Los datos de los programas en ensamblador son introducidos a la máquina principal MP-88 a través de un teclado, la cual se conecta mediante su puerto de extensión (posee dos) al módulo de expansión BP-88. Dicho módulo posee 5 zócalos de expansión de 62 pines para conectar las tarjetas de la serie 88. En concreto se utilizará para nuestros propósitos la tarjeta PT-88 de 6 puertos E/S.



Fig.3. Vista completa del conjunto MP-88 con las conexiones al teclado del sistema monitor, módulo de expansión BP-88 y tarjeta de simulación Accesorios-2000 a través de la tarjeta PT-88

La tarjeta PT-88 puede ser conectada a una gran variedad de tarjetas de simulación de la serie 88, entre ellas podemos encontrar:

- ACCESORIOS-2000. Tarjeta multiaccesorio que posee leds, displays y microrruptores.
- ADAPTADOR-2000. Tarjeta para realizar tareas ADC y DAC mediante relés y señales analógicas y digitales.
- SIMULADOR-2000. Tarjeta de simulación de un cruce de carreteras con semáforos mediante diodos leds.
- ALPHA-2000. Tarjeta de visualización de mensajes mediante un display.

Para la comunicación entre la tarjeta PT-88 y las tarjetas de simulación de la serie 88 es necesario configurar correctamente las direcciones de los puertos (PA, PB y PC), cuyas direcciones se seleccionan mediante unos microrruptores. La interfaz paralelo PT-88 contiene dos circuitos PPI-8255 y por defecto, para la realización de las prácticas, se han seleccionado las direcciones 100H para la pastilla 8255 (1) y 120H para la pastilla 8255 (2). De esta forma las direcciones de las puertas quedan según el cuadro siguiente:

	8255 (1)	8255 (2)
<b>Puerta A</b>	100 H	120 H
<b>Puerta B</b>	101 H	121 H
<b>Puerta C</b>	102 H	122 H
<b>Puerta de Control</b>	103 H	123 H

Las puertas se programan como entrada o como salida según su utilización. Esta programación se realiza mediante la palabra de control, definiendo su valor según el formato de la figura 4:

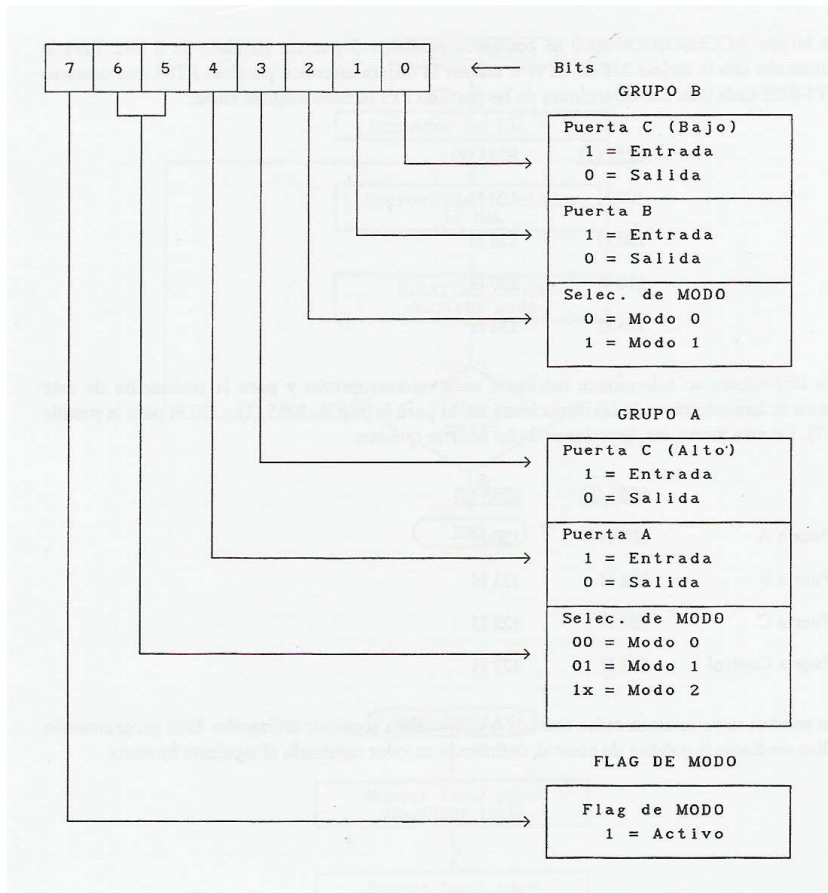


Fig.4. Esquema que indica la palabra que hay que cargar en el puerto de control, según la asignación de bits de la figura

### 3.2. Ejemplo de configuración de la tarjeta PT-88

En el siguiente ejemplo, se utilizará una tarjeta de simulación genérica, cuyas puertas se conectarán a los puertos de la tarjeta PT-88 de la siguiente forma:

8255 (1)	
Puerta A	Salida
Puerta B	Salida
Puerta C (Bajo)	Salida
Puerta C (Alto)	Entrada
8255 (2)	
Puerta A	Salida
Puerta B	Entrada

Como se trabaja en modo 0 por defecto, la palabra de control del 8255 (1) queda:

1	0	0	0	1	0	0	0
---	---	---	---	---	---	---	---

 = 88H

y la palabra de control del 8255 (2) queda:

1	0	0	0	1	0	1	1
---	---	---	---	---	---	---	---

 = 8BH

El código de ejemplo para establecer la configuración de las puertas pertenecientes al dispositivo 8255 (1) sería:

```
MOV DX, 103      ; dirección del puerto de control
MOV AL, 88      ; palabra de control
OUT DX, AL      ; poner la palabra de control en el puerto especificado
```

El código de ejemplo para mandar un dato a una puerta de salida perteneciente al dispositivo 8255 (1) sería:

```
MOV DX, 100     ; dirección de la puerta A
MOV AL, F      ; los 8 bits a 1
OUT DX, AL     ; poner el dato en el puerto especificado
```

El código de ejemplo para leer un dato de una puerta de entrada perteneciente al dispositivo 8255 (2) sería:

```
MOV DX, 121     ; dirección de la puerta B
IN DX, AL      ; poner el dato del puerto especificado en AL
```

#### 4. COMANDOS DEL MONITOR

La ventaja del ensamblador de línea estriba en que, mientras se escribe el programa en lenguaje ensamblador, no se necesita memoria para almacenar el programa fuente.

Cada vez que se introduce una instrucción, el ensamblador traducirá inmediatamente el código fuente a código máquina. Las instrucciones ensambladas que se introducen quedan almacenadas en posiciones sucesivas de memoria, comenzando con la dirección que se especifica en el comando A. Si no se especifica ninguna dirección en el comando A, las instrucciones se ensamblan en posiciones de memoria consecutivas a la dirección 0080:0000.

El ensamblador de línea está capacitado para la sintaxis del lenguaje ensamblador del 8086/88 con las siguientes reglas a tener en cuenta:

- a) Todos los valores numéricos introducidos se suponen hexadecimales y pueden tener una longitud máxima de 4 dígitos.

- b) El pseudocódigo debe introducirse en la línea precedente a la instrucción donde se modifican:

0080:1000	CS:
0080:1001	MOV AL,[1001]

Esto es equivalente a MOV AL, CS:[100]

- c) El ensamblador puede ensamblar automáticamente saltos o llamadas (JUMP o CALL):

0080:00A0	JMP A2
0080:00A2	JMP NEAR A5
0080:00A5	JMP FAR AA

- d) En el ensamblador de línea no se pueden utilizar etiquetas para aludir a una dirección o a una línea a la que se quiere saltar:

0080:0010	ADD AX,CX
0080:0012	LOOP 10

- e) El ensamblador utiliza los operandos entre corchetes para referirse a memoria:

MOV AX, 63	;Cargar AX con 63
MOV AX,[63]	;Cargar AX con el ;contenido de la ;posición de memoria 63

- f) Los códigos de operación DB y DW ensamblarán valores *byte* y *word* directamente en lugares sucesivos de memoria:

DB 10,11,12,13,"MENSAJE"
DW 'ELEGIR LA OPCION DESEADA:'
DB "MWNU DE AYUDA:"
DW 1388,1770,1B58,"FECHA"

#### 4.1. Comando A: El ensamblador de línea

*Nombre:* ENSAMBLADOR

*Sintaxis:* (1) A  
(2) A <dir>

*Objetivo:* Permitir al usuario teclear programas en lenguaje ensamblador del 8088 y ensamblarlos en código máquina línea a línea en la memoria del MP-88. Para salir del ensamblador en línea y volver al símbolo del monitor “> █” pulsar la tecla INTRO sin teclear antes ninguna instrucción.

*dir:* Si no se especifica el parámetro dirección con el comando A, el monitor utilizará el valor actual del registro CS (iniciado con 0080:0000).

#### 4.2. Comando G: Ejecutar un programa

*Nombre:* EJECUCIÓN

*Sintaxis:* (1) G  
(2) G <dir>

*Objetivo:* Ejecuta o hace correr un programa que previamente se ha introducido haciendo uso del comando A.

*dir:* La dirección se puede especificar como una dirección de desplazamiento, o de un segmento más un desplazamiento. Si no se especifica la dirección como en la sintaxis 3.1, el monitor utilizará el contenido actual del registro CS como el valor del segmento y el contenido actual del puntero de instrucción. La ejecución continuará hasta que se encuentre con un punto de ruptura o hasta que termine el programa.

#### 4.3. Comando S: Ejecutar un programa paso a paso

*Nombre:* PASO A PASO

*Sintaxis:* (1) S  
(2) S <n>

*Objetivo:* Ejecuta un programa paso a paso o un número específico de instrucciones visualizando en pantalla el contenido de los registros. La ejecución comienza en la dirección apuntada por CS y el puntero de instrucción.

*n:* El parámetro especifica el número de instrucciones a ejecutar después de que el comando S haya sido introducido. Posteriormente el programa se parará y mostrará el contenido de los registros. Debido al tamaño de la pantalla sólo se visualizarán dos pares de registros o un par de registros y la siguiente instrucción. Use las teclas de desplazamiento para visualizar el resto del contenido.

#### 4.4. Comando B: Poner / mostrar puntos de ruptura

*Nombre:* PUNTO DE RUPTURA

*Sintaxis:* (1) B  
(2) B <n>  
(3) B <n> <dir>

*Objetivo:* Coloca hasta tres puntos de ruptura y visualiza sus posiciones de colocación. Cuando el programa está ejecutándose y entra en la dirección de un punto de ruptura, la ejecución del programa se interrumpirá.

*n:* El parámetro especifica el número del punto de ruptura, cuyo valor está comprendido entre 1 y 3.

*<dir>* Dirección en la que hay que colocar el punto de ruptura. Advierta que los puntos de ruptura no pueden colocarse en direcciones de la memoria ROM. Cada dirección se puede especificar como una dirección de desplazamiento, o de un segmento más un desplazamiento.

*Ejemplos:*

B 1 90:9
----------

 Este comando coloca el punto de ruptura 1 en la dirección 0090:0009

B 2 1B
--------

 Este comando coloca el punto de ruptura 2 en la dirección 0080:001B

B
---

 Este comando muestra todos los puntos de ruptura que se han colocado en sistema, de la forma:

01 0090:0009
02 0080:001B

Utilice este comando para recordar los puntos de ruptura, siempre que haya olvidado dónde los colocó.

Suponga que hay un punto de ruptura en 0080:0004. Si usted comienza a ejecutar desde la dirección 0080:0000 y el programa no encuentra antes una instrucción de bifurcación, observará el siguiente mensaje en pantalla:

```
Break en 0080:0004
> █
```

Se puede pulsar G para reanudar la ejecución (recuerde que CS e IP apuntarán a la siguiente instrucción), S para ejecutar la siguiente instrucción, X para examinar los contenidos de los registros de la memoria o M para cambiarlos.

#### 4.5. Comando C: Cancelar puntos de ruptura

*Nombre:* CANCELAR

*Sintaxis:* (1) C  
(2) C <n>

*Objetivo:* Anula uno o todos los puntos de ruptura colocados previamente.

- (1) Este comando cancelará todos los puntos de ruptura actuales colocados en memoria
- (2) Este comando cancelará el punto de ruptura <n> tras el comando C.

*n:* El parámetro especifica el número del punto de ruptura, cuyo valor está comprendido entre 1 y 3.

#### 4.6. Comando X: Observar / alterar el contenido de los registros

*Nombre:* REGISTRO

*Sintaxis:* (1) X  
(2) X <nombre del registro>

*Objetivo:* Visualiza o cambia el contenido de cualquiera de los registros (AX, BX, CX, DX, SI, DI, IP, SP, FL -banderas- BP, CS, DS, ES y SS)

- (1) Esta opción del parámetro permite visualizar los contenidos de los registros de cuatro en cuatro.
- (2) Esta opción visualiza el contenido de un sólo registro y da la opción de reemplazar su contenido con un nuevo valor.

*Nombre del registro:* Parámetro que se puede utilizar para visualizar o cambiar el contenido de cualquiera de los 14 registros.

```
> X CS
CS = 0080 █
0090 ←
```

#### 4.7. Comando M: Observar / alterar el contenido almacenado en memoria

*Nombre:* MEMORIA

*Sintaxis:* (1) M  
(2) M <dir1>

- (3) M <dir1> <dir2>
- (4) M <dir1> <dir2> / <dato1> /
- (5) M <dir1> <dato1> [dato2]... /
- (6) M <dir1> <dir2> / <dato1> [dato2]... /

**Objetivo:** Las funciones pueden dividirse en tres categorías: examinar o cambiar el contenido de una posición de memoria, de una zona de memoria o cargar posiciones de memoria con valores.

**dir#:** Cada dirección se puede especificar como una dirección de desplazamiento, o de un segmento más un desplazamiento.

**dato#:** Representa cualquier carácter alfanumérico o número hexadecimal. Si utiliza caracteres debe precederse el dato con el símbolo (') tal como 'ABCD.

```
> M ←
0080:
0000 00 00 00 00
0004 00 00 00 00
>
```

El sistema le responderá automáticamente con los contenidos de ocho posiciones de memoria consecutivas, comenzando con la posición 0080:0000 (suponga que estas posiciones de memoria contienen ceros).

```
> M ←
0080:
0008 00 00 00 00
000C 00 00 00 00
>
```

En este punto, el contenido del puntero interno es 0080:0008. Si se introduce el mismo comando otra vez se mostrarán los contenidos de las nuevas posiciones de memoria.

```
> M A
0080:
000A 00 ■
2B ←
```

Para cambiar el contenido de la posición de memoria 0080:000A con 2B se utiliza la expresión (2). El sistema responderá con el valor actual y espera ■ que se introduzca un nuevo valor. Sin embargo si se pulsa la tecla espacio en lugar de la tecla INTRO el sistema mostrará el contenido de la siguiente posición de memoria y esperará su decisión.

```
> M 0 F/61/ ←
o
> M 0 F/'a'/ ←
```

Para llenar una zona de memoria con un valor determinado se utiliza la expresión (4). El sistema almacenará el valor 61 en todas las posiciones de memoria desde 0080:0000 hasta 0080:000F.

```
> M 100:0 100:F / 33 44 /
←
```

Utilizando M se puede almacenar un valor particular o un conjunto de valores en una zona de memoria definida. Si el número de valores es menor que el número de zonas de memoria especificada, el conjunto de valores se repetirá hasta el fin de la zona de memoria especificada.

```
0100:
0000 33 44 33 44
0004 33 44 33 44
0008 33 44 33 44
000C 33 44 33 44
```

## 5. LISTADO DE INTERRUPCIONES

Lo que sigue es un listado de las rutinas de servicio más utilizadas, que pueden ayudar a incrementar la versatilidad de un programa o para comunicarse con dispositivos externos.

INT 7H	Finaliza la ejecución de un programa y devuelve el control al sistema monitor.
INT 8H	Trae un carácter desde el teclado de la consola y lo almacena en el registro AL en forma de código ASCII.
INT 9H	Muestra el código ASCII del carácter almacenado en el registro AL sobre la consola del sistema.
INT A	Trae un valor en código ASCII desde el teclado de la consola y lo introduce en el registro AX en formato hexadecimal. AH debe guardar el número máx. de dígitos a introducir en cada línea de la pantalla, antes de que el cursor deba moverse a la siguiente línea. CL guarda el número de dígitos que se han introducido. ZF=0 indica que se ha terminado de introducir caracteres pulsando la barra de ESPACIO.
INT B	Saca un carácter almacenado en el registro AL a la pantalla LCD del MP-88. CH debe indicar si se muestra el cursor por la pantalla. CL debe indicar si se hará un <i>scroll</i> por la pantalla.
INT C	Realiza un ruido corto <i>beep</i> por el altavoz del sistema
INT D	Saca una cadena de caracteres a la pantalla que están almacenados en memoria. DS debe contener la dirección de segmento. SI debe contener el desplazamiento de comienzo. El último byte de la cadena de caracteres a mostrar debe tener a 1 el bit de signo, indicando el final de la cadena.
INT E	Visualiza en forma hexadecimal el contenido del registro AL en la pantalla de la consola.
INT 18	Hace que el altavoz del MP-88 suene en función de los contenidos de los registros BX (duración) y CX (frecuencia).
INT 21 AH=0	Finaliza la ejecución de un programa visualizando el mensaje "Programa terminado" guardando los valores de los registros de usuario.
INT 21 AH=90	Realiza un retardo variable de N ms especificado en el registro CX. Durante el servicio de esta función no se realiza el testeado del teclado.

## PRÁCTICA 5: TARJETA ACCESORIOS – 2000

Fecha límite de entrega: Semana 15 - del 22/1-26/1

### 1. Objetivo de la práctica

Realizar un programa en ensamblador del 8088 que, ejecutándose en la plataforma MP-88, sea capaz de:

- Leer el estado de los *microrruptores*.
- Iluminar los LEDs correspondientes a los *microrruptores* activados.
- Mostrar en el *display* el número de *microrruptores* activados.

### 2. Descripción de la tarjeta ACCESORIOS-2000

La tarjeta ACCESORIOS-2000 se comunica con la tarjeta de expansión PT-88 mediante 5 puertas paralelo de 8 bits:

- 1 para la lectura del estado de los microrruptores
- 1 para la activación de los LEDs
- 2 para la escritura en el display
- 1 para la lectura del teclado

#### 2.1. Microrruptores

Cuando un *microrruptor* está activado se pone a nivel alto (1) el *pin* correspondiente de la puerta a la que está conectado.

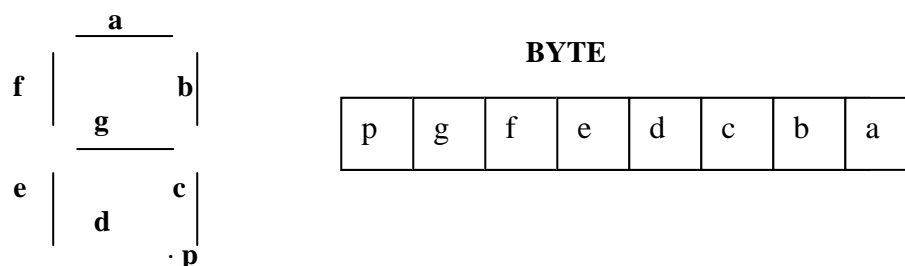
#### 2.2. LEDs

Para encender un LED basta con poner a 1 el *pin* correspondiente de la puerta a la que está conectado

#### 2.3. Displays

Los *displays* se controlan mediante 2 puertas paralelo de 8 bits, una permite seleccionar el *display* en el que queremos mostrar la información de los 4 disponibles, y la segunda muestra la información en el *display* seleccionado en formato “7 segmentos”. Tanto la selección del *display* como los segmentos de éste son activos con un 0.

- **Selección del display.** Sólo se utilizan los cuatro primeros bits de la puerta correspondiente. Poniendo a 0 el bit 0, se activa el primer *display*; poniendo a 0 el bit 1 se activa el segundo *display*; etc. Sólo se puede activar un *display* cada vez.
- **Selección de los segmentos.** La información que se muestra en el *display* se codifica de la siguiente manera:



Por ejemplo, para visualizar '5' se deben encender a,c,d,f y g. Por lo tanto:

1	0	0	1	0	0	1	0
---	---	---	---	---	---	---	---

El valor para visualizar '5' es, por ende, 92H

#### 2.4. Teclado

El teclado es un matriz de 4 filas por 4 columnas:

	Y0	Y1	Y2	Y3
X0	0	1	2	3
X1	4	5	6	7
X2	8	9	A	B
X3	C	D	E	F

Para detectar si una tecla está pulsada se envía un nivel alto (1) a las columnas, una tras otra, y se lee el estado de las filas (todas a la vez). Si alguno de los 4 bits correspondientes a las 4 filas está a 1, significa que la tecla correspondiente a la intersección de la columna seleccionada y la fila activa está pulsada.

## **PRÁCTICA 6: TARJETA ADAPTADOR – 2000**

Fecha límite de entrega: Semana 15 - del 22/1-26/1

### **1. Objetivo de la práctica**

Realizar un programa en ensamblador del 8088 que, ejecutándose en la plataforma MP-88, sea capaz de:

- Leer la salida de los convertidores analógicos-digitales
- Activar los relés correspondientes a los CAD que tengan un nivel bajo (0)
- Encontrar la tensión umbral para la que el CAD interpreta un 1

### **2. Descripción de la tarjeta ADAPTADOR-2000**

La tarjeta ADAPTADOR-2000 se comunica con la tarjeta de expansión PT-88 mediante 2 puertos paralelo de 8 bits:

- 1 para la lectura de la salida de los CADs
- 1 para la activación de los relés

#### **2.1. CADs**

La tarjeta dispone de 8 entradas analógicas (en la parte izquierda) cada una de las cuales ataca a un CAD cuya salida está conectada a uno de los pines de la puerta paralelo de 8 bits. Poniendo una diferencia de tensión alta (5V p.ej.) en una entrada analógica, tendremos un 1 (nivel alto) en el pin correspondiente.

**Importante: la diferencia de tensión en las entradas analógicas no debe ser mayor de 5 V**

#### **2.2. Relés**

La tarjeta dispone de 8 relés conectados a la otra puerta paralelo de 8 bits. Poniendo un 1 o un 0 en cada bit de la puerta, se activa o desactiva el relé correspondiente.

**PRÁCTICA 7a: TARJETA SIMULADOR – 2000**

Fecha límite de entrega: Semana 15 - del 22/1-26/1

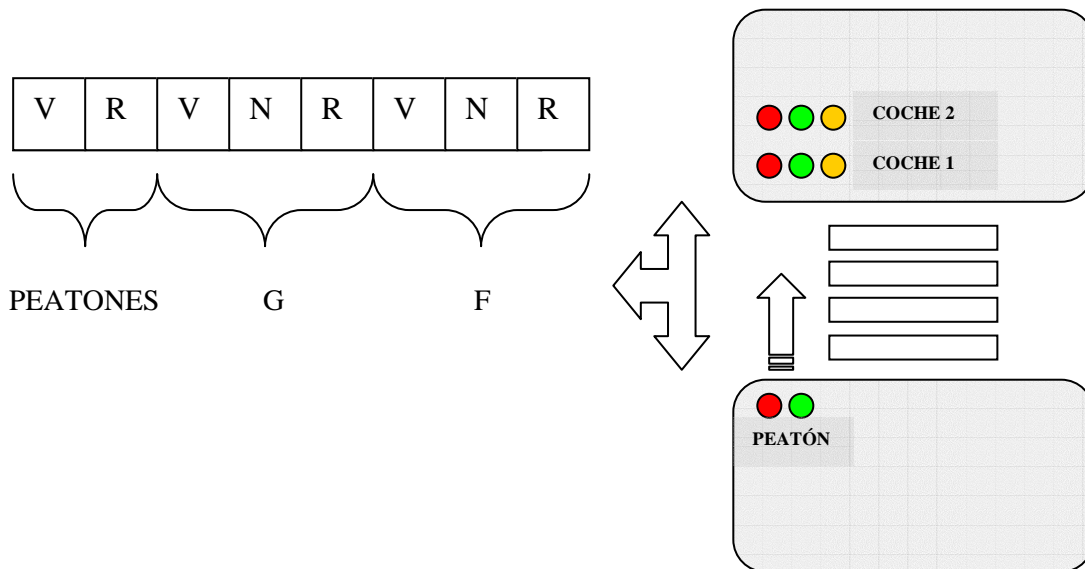
**1. Objetivo de la práctica**

Realizar un programa en ensamblador del 8088 que, ejecutándose en la plataforma MP-88, sea capaz de:

- Simular el comportamiento de los semáforos del cruce activando los LEDs rojo, verde y amarillo correspondientes
- Introducir los retardos adecuados para la correcta iluminación de los semáforos y ejecutar el programa en un bucle continuo

**2. Descripción de la tarjeta SIMULADOR-2000**

La tarjeta SIMULADOR-2000 se comunica con la tarjeta de expansión PT-88 mediante 3 puertos paralelo de 8 bits. Cada puerto está dividido en dos grupos de 3 bits (semáforo de coches) y un grupo de 2 bits (semáforo de peatones) de la siguiente forma:



Para activar el semáforo de forma correcta, se debe pasar la palabra hexadecimal correspondiente a la combinación de 8 bits en el puerto adecuado (PA, PB y PC). En la palabra de 8 bits, un bit a nivel alto (1) hace que en su posición se encienda el color que representa y un nivel bajo (0) que permanezca apagado.

**Importante: la tensión de alimentación en las entradas no debe ser mayor de 5 V**

## **PRÁCTICA 7b: TARJETA ALPHA – 2000**

Fecha límite de entrega: Semana 15 - del 22/1-26/1

### **1. Objetivo de la práctica**

Realizar un programa en ensamblador del 8088 que, ejecutándose en la plataforma MP-88, sea capaz de:

- Visualizar en la matriz de displays la secuencia de teclas que se van pulsando en el teclado y rotar de derecha a izquierda por orden de pulsación.
- Los caracteres que se visualizarán, son los correspondientes a todas aquellas teclas cuyo carácter tenga un valor ASCII comprendido entre los valores 20H y 80H.

### **2. Descripción de la tarjeta ALPHA-2000**

La tarjeta ALPHA-2000 se comunica con la tarjeta de expansión PT-88 mediante 3 puertos paralelo de 8 bits. La pantalla está formada por 8 displays (D0, D1... D7) compuestos por matrices de diodos LEDs de 7 filas x 5 columnas y el signo del punto (columna 0, fila 7).

PUERTO A	Columnas del display
PUERTO B	Selección del display
PUERTO C	Filas del display

El procedimiento a seguir consiste en seleccionar en primer lugar el display a visualizar. Por ejemplo, si quisiéramos seleccionar los displays D0 y D1, la palabra formada sería 3h (00000011). A continuación se debe enviar fila a fila la combinación de bits que se desea iluminar en las 5 columnas de ese display.

**Importante: la tensión de alimentación en las entradas debe ser de 12 V**