

1. Se dispone de una CPU cuyas señales se describen en la Figura 1. Como dirección de E/S se utilizan los 8 bits inferiores del bus de direcciones. Se desea conectar, mediante E/S programada, dicha CPU a un periférico de salida cuyas señales se muestran en la Figura 2 y cuyo protocolo de *handshaking* se muestra en la Figura 3.

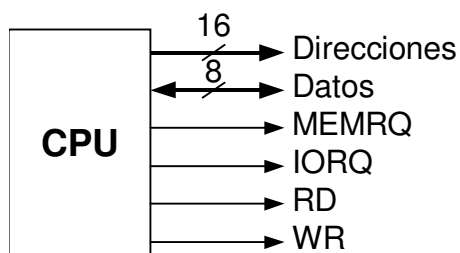


Figura 1. Señales de la CPU.

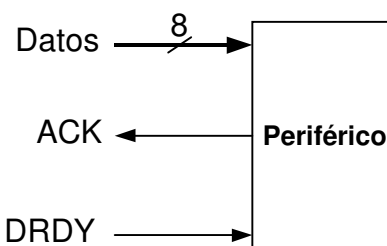


Figura 2. Señales del periférico.

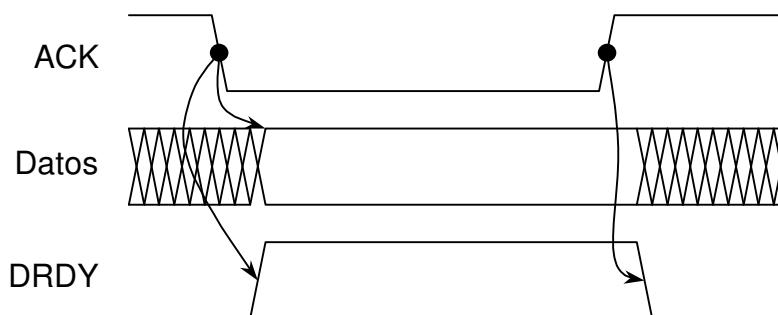


Figura 3. Protocolo de *handshaking* del periférico..

- a) Diseñe una interfaz que permita la conexión entre la CPU y el periférico.
b) Escriba un programa que, de acuerdo al protocolo exigido, envíe un dato al periférico.

2. La CPU de cierto computador dispone para la E/S de las siguientes señales:

A_0-A_7 \equiv 8 líneas de dirección para seleccionar el periférico

D_0-D_7 \equiv 8 líneas de datos para leer o enviar un byte

\overline{RD} \equiv 1 línea para indicar lectura (el procesador toma la información por D_0-D_7)

\overline{WR} \equiv 1 línea para indicar escritura (el procesador envía la información por D_0-D_7)

La CPU dispone de las instrucciones IN *dir* y OUT *dir*, donde *dir* indica la dirección del periférico que intercambia un byte de información con un determinado registro de la CPU. Se desea emplear esta CPU para supervisar 5000 sensores de una fábrica. Cada sensor devuelve 0/1 indicando si una determinada válvula está cerrada / abierta.

- a) Diseñe el sistema de direccionamiento e interfaz necesarios para poder acceder y leer los 5000 sensores.
b) Indique la secuencia de pasos necesaria para leer el estado de un sensor determinado.

3. Se dispone de una CPU, de bloques de memoria, y de periféricos, como se describe en la Figura 4, cuyas características más importantes son las siguientes:

CPU: En los accesos a memoria ($M / \overline{IO} = 1$) son válidos los 16 bits del bus de direcciones y en los accesos a E/S ($M / \overline{IO} = 0$) son válidos únicamente los 4 bits menos significativos de la dirección

(A0-A3).

Memorias: Bloques de 64 K palabras cuya activación se realiza mediante la señal $\overline{CS} = 0$.

Periféricos: La CPU podrá leer de ellos o escribir en ellos datos de 16 bits.

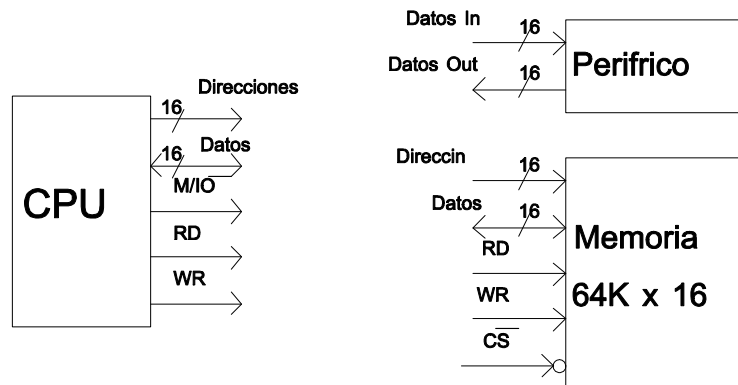


Figura 4. CPU, bloques de memoria y periféricos del problema 3.

Se desea conectar a dicha CPU 15 periféricos de los descritos, y además se desea tener una memoria de 128 K palabras dividida en dos bancos de 64 K palabras.

a) Diseñe una interfaz de E/S simple para cada periférico con dos puertos, uno de entrada y otro de salida, pero ambos con las misma dirección. Utilice un *latch* o registro *buffer* para escritura en el periférico, un registro *buffer* triestado para lectura del periférico (ambos con entradas de selección \overline{CS}), las señales de lectura y de escritura y una señal de selección de periférico CSi.

b) Diseñe la conexión y decodificación tanto de la E/S como de la memoria. Si faltasen líneas de direccionamiento utilice un biestable D (direccionado como puerto de salida y seleccionado para escritura a través de su entrada CK) para indicar a cuál de los dos bancos de memoria se desea realizar los siguientes accesos.

c) Piense si la solución diseñada para el apartado b) puede plantear problemas en la ejecución de programas (en la captación de instrucciones) al cambiar de banco de memoria para acceder a datos. Si es así soluciónelo modificando el diseño (basta añadir un par de puertas) suponiendo que la CPU activa (=1) una salida adicional FETCH siempre que realiza un acceso a memoria correspondiente a una búsqueda de instrucción, y que en el banco 0 se almacenarán datos y programas y en el banco 1 sólo datos.

d) Escriba un programa en lenguaje ensamblador para mover un bloque de 128 palabras desde la posición de memoria 00000h y siguientes hasta la 10000h y siguientes. Utilice instrucciones del siguiente repertorio:

```
MOV RegDestino,DatoInmediato
MOV RegDestino,RegFuente
MOV [RegDestino],RegFuente
MOV RegDestino,[RegFuente]
IN RegDestino,PuertoEntradaInmediato
OUT PuertoSalidaInmediato,RegFuente
INC RegDestino
DEC RegDestino
CMP RegDestino,DatoInmediato
CMP RegDestino,RegFuente
JMP Etiqueta
Jcond Etiqueta
```

donde *cond* es cualquiera de las condiciones conocidas, los corchetes [] indican direccionamiento indirecto a memoria y las instrucciones INC, DEC y CMP afectan a los indicadores. Suponga que la CPU dispone de todos los registros que sean necesarios: R0, R1, R2, R3, R4, ...

4.

Se quiere controlar mediante un microprocesador de 8 bits un gran sistema de alarma que cuenta con 100 sensores de "efecto doppler" (la señal de salida de cada uno será de 5 ó 0 voltios según detecte o no la presencia de un intruso en las inmediaciones del correspondiente sensor).

Diseñe la estructura de E/S del sistema usando circuitos integrados 74244 (Figura 5) y 74138

(Figura 6), y suponiendo E/S aislada o independiente (no mapeada en memoria. Indique las direcciones de puertos usadas.

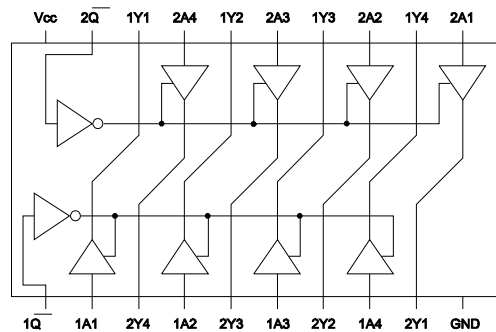
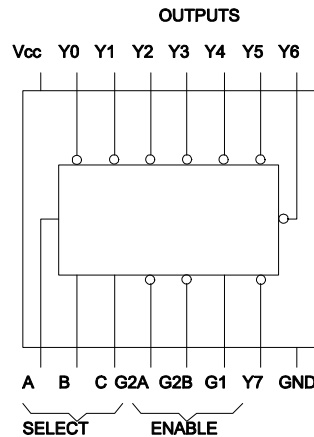


Figura 5. Buffer octal (puerto de entrada de 8 bits) 74244.



Inputs		Outputs							
Enable	Select								
G1 G2*	C B A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X 1	X X X	1	1	1	1	1	1	1	1
0 X	X X X	1	1	1	1	1	1	1	1
1 0	0 0 0	0	1	1	1	1	1	1	1
1 0	0 0 1	1	0	1	1	1	1	1	1
1 0	0 1 0	1	1	0	1	1	1	1	1
1 0	0 1 1	1	1	1	0	1	1	1	1
1 0	1 0 0	1	1	1	1	0	1	1	1
1 0	1 0 1	1	1	1	1	1	0	1	1
1 0	1 1 0	1	1	1	1	1	1	0	1
1 0	1 1 1	1	1	1	1	1	1	1	0

$$* G2 = G2A + G2B$$

Figura 6. Decodificador 3 a 8 74138.

- 5.** Cada una de las 16 teclas del teclado de la Figura 7 consiste en un conmutador como el de la Figura 8. El comportamiento real de este circuito cuando se pulsa una tecla se muestra en la Figura 9.

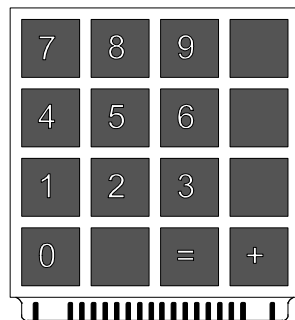


Figura 7. Teclado.

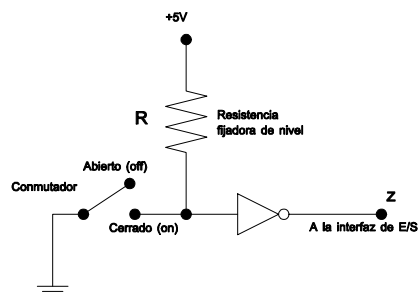


Figura 8. Conmutador.

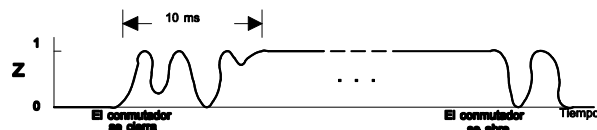


Figura 9. Comportamiento del conmutador.

- Diseñe la interfaz necesaria para conectar el teclado a un 8086.
- Escriba una subrutina READKEY que devuelva en el registro CL el número de tecla pulsada, de 1 a 16, 0 si no hay ninguna tecla pulsada y -1 si se ha pulsado más de una tecla. La rutina no contemplará la eliminación de rebotes.
- Suponga que dispone de una subrutina DELAY que realiza un retardo de 15 ms. ¿Cómo podría aprovechar esa subrutina junto con la del apartado b) para evitar los rebotes?

6. Se desea conectar un sistema basado en 8086 (el bus del sistema se detalla en la Figura 10) al convertidor analógico / digital (A/D) de la Figura 12 mediante un interfase paralelo programable 8255 (Figura 11), que comprende tres puertos de 8 bits (A, B y C) y un registro o puerto de control.

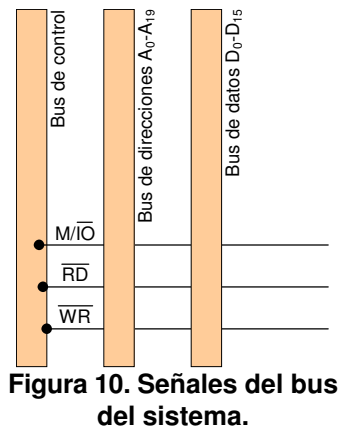


Figura 10. Señales del bus del sistema.

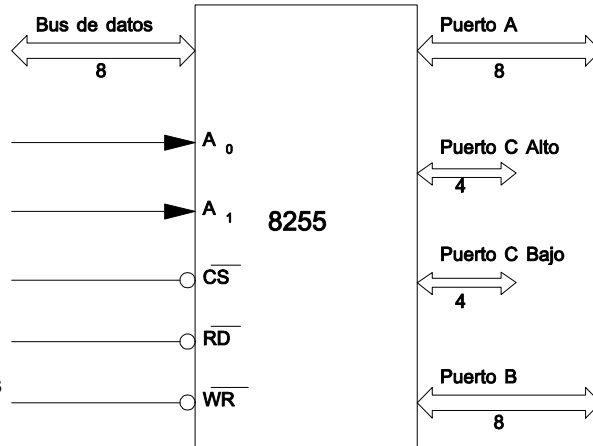


Figura 11. Interfaz paralela programable 8255.

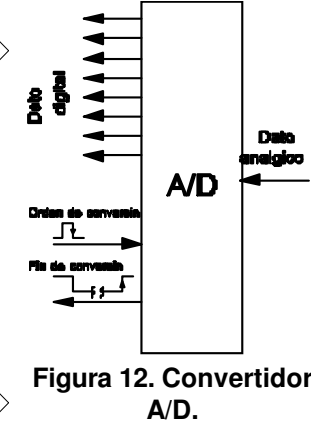


Figura 12. Convertidor A/D.

La selección de un puerto de estos cuatro se realiza mediante los bits de dirección A_0 y A_1 del 8255 (Tabla I). Se pueden seleccionar por software tres modos de funcionamiento del 8255: *modo 0* o E/S básica, *modo 1* o E/S validada, y *modo 2* o E/S validada con bus bidireccional. El modo 0 o E/S básica se usa en operaciones sencillas de E/S con control programado. En ese modo el puerto C funciona como dos puertos independientes de 4 bits, y cada puerto (Puerto A, Puerto B, Puerto C Bajo y Puerto C Alto) puede ser programado independientemente como de entrada o de salida. Cada bit de estos puertos puede ser utilizado para datos o para señales de control o de estado. La configuración de cada puerto como de salida o de entrada, así como el modo de funcionamiento de los puertos A y B puede programarse enviando la palabra de la Figura 13 al registro de control. Para escribir en el puerto C en el modo 0 se puede usar una instrucción normal de escritura en ese puerto, y si se desea escribir un solo bit sin modificar los demás se envía simplemente la palabra de la Figura 14 al registro de control.

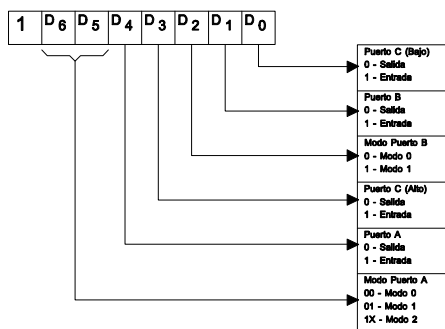


Figura 13. Palabra de control usada para programar la configuración de cada puerto.

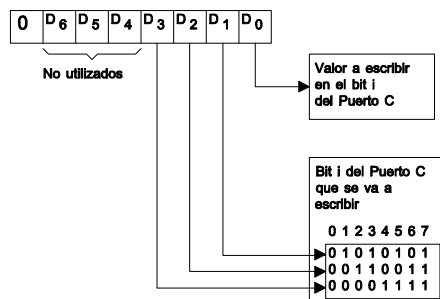


Figura 14. Palabra de control usada para escribir un bit del Puerto C.

Tabla I.
Direccionamiento de puertos.

A_1	A_0	Registro direccionado
0	0	Puerto A
0	1	Puerto B
1	0	Puerto C
1	1	Registro de control

Para efectuar una conversión analógico-digital se debe enviar una orden de conversión (impulso positivo) por la línea *Orden de conversión* del convertidor. El convertidor A/D pone inmediatamente la señal de estado *Fin de conversión* a 0 y procede a realizar la conversión. Cuando la conversión finaliza el convertidor pone *Fin de conversión* a 1.

a) Diseñe un **esquema de conexión** con espacio de E/S separado del espacio de memoria de forma que las direcciones de los puertos / registros de la Tabla I sean 0, 2, 4, y 6, respectivamente, teniendo en cuenta que al 8086 pueden conectarse hasta 64 K puertos de 8 bits (los bits A_{16} - A_{19} no se usan), y que los puertos de 8 bits en direcciones pares se conectan a las líneas D_0 a D_7 .

b) Dibuje el **organigrama** y escriba la correspondiente sección de un **programa en ensamblador** para leer un dato del convertidor, usando únicamente el *modo 0* del 8255.

7. Consideremos una CPU de 8 bits de ancho de palabra, que cuenta con una sola entrada de petición de interrupción **INT** y la correspondiente salida de reconocimiento de interrupción **INTA**.

Se desea dotar a la CPU de 4 niveles de petición de interrupción, en cada uno de los cuales pueden interrumpir 4 periféricos. Para ello se pide:

a) Diseñar la lógica de control de interrupciones que hay que añadir a la CPU de modo que:

- Se resuelvan las prioridades de los distintos niveles.
- Se resuelvan las prioridades de los distintos dispositivos del mismo nivel.
- Se identifique el dispositivo mediante una vector de 8 bits.

b) Describir el funcionamiento de la unidad diseñada.

8. Se dispone de una CPU (Figura 15) con las siguientes características:

- Bus de direcciones de 16 bits y bus de datos de 16 bits.
- Cuando se realizan accesos a puertos de E/S (instrucciones del tipo IN y OUT) se activa la señal IORQ, y cuando se accede a una dirección de memoria se activa la señal MEMRQ.
- La lectura se controla con la señal RD y la escritura con la señal WR.
- Hay una línea de petición de interrupción INT y una salida de reconocimiento de interrupción INTA.
- Las interrupciones son vectorizadas, con vectores de 8 bits.

a) Dibuje el interfaz de E/S que es necesario añadir a un periférico de E/S como el de la Figura 16 para conectarlo a la CPU.

b) Para un sistema con dos niveles de petición de interrupción (uno con mayor prioridad que el otro en el caso de peticiones simultáneas) con cuatro periféricos de E/S en cada nivel conectados mediante *daisy-chain*, describa las conexiones y el esquema de decodificación para los dispositivos de E/S, detallando la conexión con las líneas INT e INTA y la generación de los vectores de interrupción (pero sin detallar la implementación a nivel de puertas / biestables del *daisy-chain*).

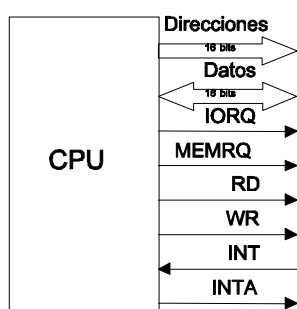


Figura 15. CPU.

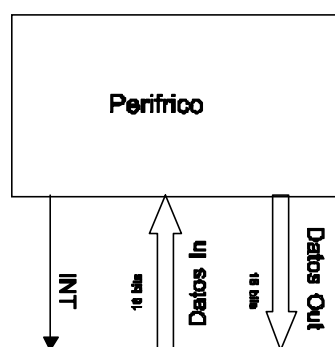


Figura 16. Periférico de E/S.

9. Un microprocesador tiene, entre otras, las siguientes características:

- Bus de datos: 8 bits, D_7-D_0 .
- Bus de direcciones: 16 bits, $A_{15}-A_0$.
- Señales de control:
 - IORQ, indica operación de E/S (también se utiliza en el reconocimiento de interrupciones).
 - MREQ, indica operación de acceso a memoria.
 - WR, operación de escritura.
 - RD, operación de lectura.
 - F0 y F1, indican la fase de ejecución del microprocesador:

F1	F0	Fase
0	0	Ejecución
0	1	FETCH
1	0	Acceso a memoria (datos)
1	1	Reconocimiento de interrupción

- INT3, INT2, INT1 e INT0, líneas de solicitud de interrupción.

Para la ubicación de las rutinas de interrupción se emplea vectorización. Cuando el microprocesador atiende una interrupción activa $F1=1$ y $F0=1$, y coloca en las líneas A_1 y A_0 del bus de

direcciones el valor binario correspondiente al nivel (0, 1, 2 y 3) de la interrupción aceptada. Seguidamente, activa IORQ para validar el contenido del bus de direcciones y leer del bus de datos un vector de 8 bits que debe suministrarle el periférico cuya solicitud de interrupción ha sido atendida.

Se desea conectar a esta CPU cuatro periféricos de sólo lectura, cada uno de los cuales suministra datos de 8 bits y cuenta con una línea de interrupción. El vector generado por cada periférico ha de ser programable desde la CPU (mediante una instrucción de salida).

Diseñe el sistema de E/S, es decir, mapa de E/S, decodificación, gestión de interrupciones. Utilice los decodificadores, registros, buffers triestado, puertas lógicas, etc. que crea convenientes.

10. Considere un paso a nivel con barrera como el de la Figura 17.

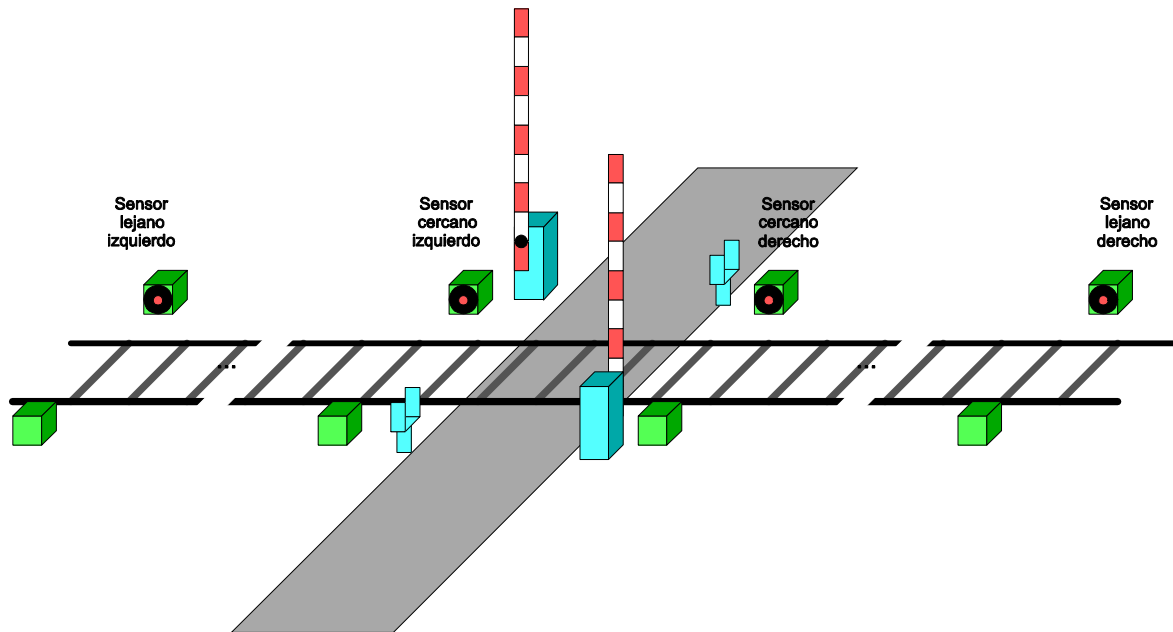


Figura 17. Paso a nivel con barrera.

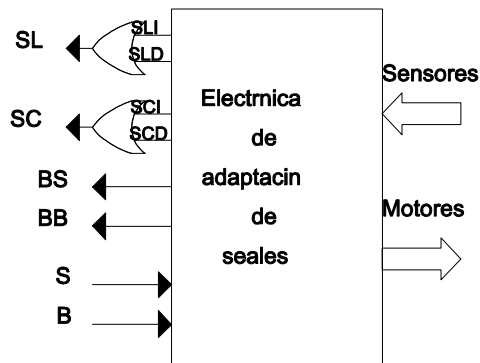


Figura 18. Circuitería de adaptación de señales.

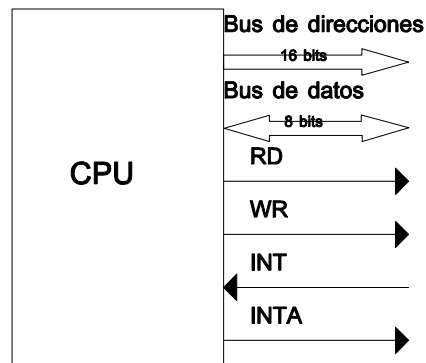


Figura 19. CPU y buses.

En la vía existen dos sensores lejanos situados a una distancia adecuada del paso a nivel, y dos sensores cercanos colocados junto al paso, para detectar el paso del tren. Las barreras disponen de sensores para determinar si se encuentran totalmente subidas y totalmente bajadas.

Para el control de las barreras se dispone de una circuitería de adaptación (Figura 18) que genera y acepta las siguientes señales:

SL (S ensor L ejano)	Se pone a 1 cuando alguno de los sensores lejanos detecta el principio del tren (siempre que el tren se acerca al paso a nivel, nunca cuando se aleja) y se vuelve a poner a 0 cuando el sensor detecta el final del tren.
SC (S ensor C ercano)	Se ponen a 1 cuando alguno de los sensores cercanos detecta el principio del tren y se vuelve a poner a 0 cuando detecta el final del tren.
BS (B arreras S ubidas)	Están a 1 siempre que las dos barreras estén completamente subidas o bajadas, respectivamente.
BB (B arreras B ajadas)	
S (S ubir barreras)	Mientras estén a 1 funciona el motor encargado de subir o bajar, respectivamente, las barreras. Habrá que ponerlos a 0 cuando las barreras lleguen al tope superior o inferior, respectivamente, para no forzar los motores.
B (B ajar barreras)	

Se desea usar la CPU de la Figura 19 para controlar el sistema, esto es, bajar las barreras cuando se acerque un tren y subirlas cuando haya rebasado completamente el paso a nivel. La CPU acepta hasta 256 interrupciones vectorizadas. Los vectores de interrupción se sitúan en la parte inferior de la memoria, en direcciones pares (cada vector contiene una dirección de 16 bits). La CPU acepta una interrupción cuando se detecta un flanco de subida en la señal INT, respondiendo con un pulso INTA, durante el cual se lee el número de vector de interrupción por el bus de datos.

a) Diseñe la lógica de decodificación e interfaz necesaria para conectar las señales anteriores a la CPU, suponiendo que se va a usar únicamente E/S programada.

b) Suponiendo que se desea dedicar la CPU también a otras tareas, modifique el diseño del apartado *a)*, de tal modo que se tengan dos fuentes de interrupción, SL y SC, la primera prioritaria sobre la segunda, que pueden interrumpir a la CPU a través de INT, cada una con un vector de interrupción diferente.

c) Realice los diagramas de flujo para cada una de las dos ISRs correspondientes al apartado *b)*, y escriba el programa en ensamblador de la primera (SL).

11. Un computador consta de una CPU y un dispositivo de E/S conectados a M.P. por medio de un bus común de una palabra. La CPU puede ejecutar, como máximo, 10 MIPS. Una instrucción necesita, por término medio, 5 ciclos máquina y durante tres de ellos utiliza el bus de memoria. Una operación de lectura o escritura en M.P. lleva un ciclo. Estime la velocidad de transferencia de datos (palabras/s) en los dos siguientes supuestos:

a) El computador tiene 5 tareas en ejecución que no realizan E/S por programa. Se arranca una nueva tarea para realizar E/S por programa. Los recursos del ordenador se reparten equitativamente entre las tareas. Cada transferencia de un dato requiere dos instrucciones.

b) Una vez arrancado el periférico, éste transfiere sus datos accediendo directamente a memoria, realizando la transferencia por bloque.

12. a) Dibuje un esquema de un sistema de E/S por DMA que conecte un microprocesador de 8 bits (bus de datos de 8 bits y bus de direcciones de 16 bits) que usa E/S con correspondencia en memoria, la memoria principal y una impresora. El sistema incluirá:

- Una interfaz paralela con un puerto de salida de un byte, que se comunica con la impresora a través de un bus de datos de 8 bits.
- Un controlador de DMA con los registros habituales.

b) Escriba en lenguaje ensamblador la rutina de inicialización del DMA para imprimir un bloque de 200 caracteres (bytes) almacenados a partir de la dirección 0x3000 de memoria.

13. Un computador de 32 bits tiene dos canales selectores y un canal multiplexor. Cada canal selector soporta dos unidades de disco y dos unidades de cinta magnética. El canal multiplexor tiene dos impresoras de líneas, dos lectoras de tarjetas y diez terminales CRT conectados a él. Supongamos que las velocidades de transferencia para estos periféricos son las siguientes:

Disco: 800 KB/s
 Cinta: 200 KB/s
 Impresora: 6.6 KB/s
 Lectora: 1.2 KB/s
 CRT: 1 KB/s

Estime la máxima velocidad de E/S conjunta de este sistema.

14. Se pretende leer y procesar bloques de N bytes desde un dispositivo de entrada orientado a caracteres utilizando una técnica de *dobles buffer*. El dispositivo genera un breve pulso a 1 por una línea de 1 bit cuando dispone de un nuevo dato de 8 bits.

a) Diseñe la interfaz de entrada necesaria para conectar el dispositivo a una CPU con E/S mapeada en memoria, bus de direcciones de 32 bits, bus de datos de 8 bits, señal de control R/W#, y entrada de interrupción INT. Esta interfaz ha de contar adicionalmente con:

- un puerto de estado de 1 bit que proporcione a la CPU la señal de *dato listo* en el caso de que se utilice entrada programada con consulta de estado, y
- un puerto de control de 1 bit que permita habilitar la generación de interrupciones por parte del dispositivo para realizar entrada por interrupciones.

b) Escriba en lenguaje de alto nivel (se recomienda C) un programa y una *ISR* (rutina de servicio de interrupción) que permitan realizar la técnica de *dobles buffer*. Para ello se ha de contar con dos buffers (0 y 1) de N bytes cada uno. El programa inicialmente leerá mediante E/S programada con consulta de estado el bloque 0. A continuación entrará en un bucle infinito en el que se llamará a una función dada `ProcesarBuffer(int NumBuffer)` encargada de procesar un buffer completo ya leído mientras la *ISR* va realizando la transferencia de caracteres a memoria. Debe tenerse en cuenta que el periférico puede ser más lento que `ProcesarBuffer()`, en cuyo caso no se debe entrar de nuevo en esta función hasta que no termine de llenarse el siguiente buffer. También puede preverse el caso contrario, es decir, que `ProcesarBuffer()` termine después de que haya finalizado la transferencia del siguiente buffer, en cuyo caso no se almacenarán nuevos caracteres en un buffer hasta que no termine el procesamiento de ese buffer.

15. Una planta industrial usa varios sensores para monitorizar temperatura, presión y otros factores. La salida de cada sensor consiste en un conmutador ON / OFF, y ocho de estos sensores han de ser conectados al bus de un pequeño ordenador. Diseñe una interfaz apropiada de forma que el estado de los ocho sensores pueda leerse como un único byte en la dirección FE10₁₆. Diseñe también la interfaz apropiada para conectar un indicador de siete segmentos (cuya entrada está codificada en 4 bits) como dispositivo de salida en la dirección FE11₁₆.

16. Sea el sistema de adquisición de datos analógicos de la Figura 20 compuesto por los siguientes módulos:

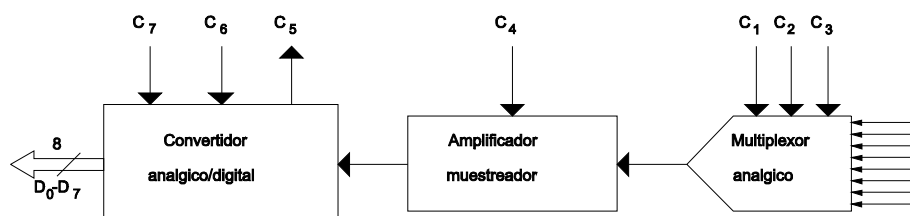


Figura 20. Sistema de adquisición del problema 16.

- Multiplexor analógico. Tiene 8 entradas y una salida analógicas. La salida está gobernada por tres señales de control (C_1 , C_2 y C_3).

- Amplificador muestreador (SH). Toma una muestra de la entrada analógica y la mantiene a su salida. Se gobierna mediante una señal de control (C_4), que normalmente está a 1 y que debe pasar a 0 durante un período de 10 μ s para realizar la toma de la señal, tal y como muestra la Figura 21.



Figura 21. Señal C_4 .

- Convertidor analógico digital de 8 bits. Produce una palabra de 8 bits proporcional a la entrada analógica. Se gobierna mediante la señal de control C_6 , que debe permanecer a 0 durante el tiempo de la conversión, que dura de 10 a 25 μ s. El convertidor contesta a la señal C_6 mediante la señal C_5 , que pasa a 1 cuando el convertidor completa la conversión. La lectura del valor digital se hace mediante la señal C_7 , que activa los 8 buffers triestado (D_0 - D_7) del convertidor y que hace un reset de la señal C_5 .

a) Diseñe un sistema de E/S basado en interrupciones vectorizadas (cada vez que se completa una conversión ha de producirse una interrupción) para gobernar este sistema de adquisición de datos, partiendo de las señales del microprocesador de la Figura 22. Incluya en el diseño la lógica necesaria para la lectura del vector de interrupción.

b) Suponiendo que todas las instrucciones tienen una duración de 1 μ s, escriba la rutina de servicio de interrupción que permita leer el valor convertido (dejando el resultado en el registro R2) y prepare una nueva conversión de la entrada analógica especificada en el registro R1.

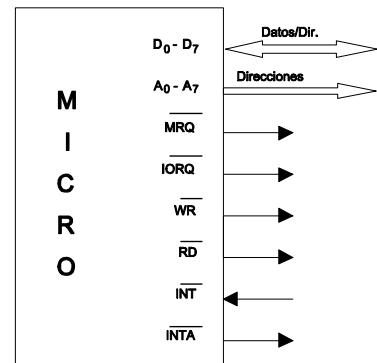


Figura 22. Microprocesador.

17. Un ordenador tiene instrucciones que requieren dos ciclos del bus, uno para captar la instrucción y otro para captar los datos. Cada ciclo del bus toma 250 ns y cada instrucción toma 500 ns (es decir, el procesamiento interno es despreciable). El ordenador tiene un disco con 16 sectores por pista, cada uno de 512 bytes. El tiempo de rotación del disco es 8,192 ms. Cuando se transfieren datos entre disco y memoria por DMA se utiliza robo de ciclo, y el disco transfiere datos a su máxima velocidad. ¿A qué porcentaje de su velocidad normal (sin transferencias con disco) se reduce la velocidad del ordenador durante la transferencia por DMA de un gran bloque de datos entre disco y memoria si cada transferencia simple por DMA tarda un ciclo del bus? Considere dos casos: transferencias de 8 bits y de 16 bits de ancho en el bus.

18. Disponemos de un microprocesador de 8 bits (bus de datos de 8 bits y bus de direcciones de 16 bits) con E/S independiente. Diseñe un sistema de E/S que permita acceder a los siguientes puertos: puerto 0x0020 de entrada y 0x0021 de salida. Utilice lógica de decodificación distribuida. No emplee decodificadores.

19. Un controlador de E/S posee un búfer para el almacenamiento temporal de los datos con una capacidad de 256 KB. En un instante determinado inicia una operación de E/S con una impresora a una velocidad de transferencia de 256 KB/s. Si el controlador de E/S recibe la información que debe enviar a la impresora a una velocidad de 1 MB/s, ¿cuánto tiempo tardará en llenarse por primera vez el búfer suponiendo que inicialmente está vacío, y que recibe y envía información simultáneamente de forma continua?