

## Práctico 9 Repaso general

### Ejercicio 1 (Ex. Set. 94 Pr.2)

Se desea diseñar un sistema basado en un Z80 con un mecanismo que permita, una vez en funcionamiento, redireccionar los dispositivos de entrada/salida por medio de una tabla configurable por software. Dicha tabla deberá convertir la **dirección de E/S del Z80** en la **dirección externa** que seleccione el dispositivo efectivamente accedido en cada caso.

Este mecanismo deberá funcionar de la siguiente forma:

- cada vez que un programa ejecute un acceso a la **dirección de E/S del Z80** NNh, el sistema deberá acceder al dispositivo en la **dirección externa** MMh, dada por la tabla que previamente fue llenada por el programador.
- este mecanismo deberá ser transparente al programa, es decir, el procedimiento anterior se deberá realizar en el tiempo de ejecución del acceso a E/S y sin agregar ninguna instrucción al programa.

Para la implementación de este dispositivo se usará una memoria RAM de 256 bytes con patas de entrada y salida separados. Esta memoria tendrá almacenadas las **direcciones externas** que se utilizarán de manera adecuada por el hardware cada vez que se acceda a un puerto; pero además deberá estar mapeada en el espacio de memoria para poder escribir en ella.

Se pide:

- a) Diseñar el hardware del sistema indicando todas las conexiones y señales necesarias para el funcionamiento del dispositivo. El sistema deberá tener **además** 32kbytes de ROM, 16kbytes de RAM. Equipar al sistema con dos puertos de entrada y dos puertos de salida en las **direcciones externas** 00h y 01h.
- b) Escribir una rutina que modifique el direccionamiento de manera que a través de las **direcciones de E/S del Z80** de 0 a 10d se acceda al puerto en la **dirección externa** 00h .
- c) Dibujar en un diagrama de tiempos todas las señales significativas para la generación del pulso de selección de dispositivo del puerto de entrada en la **dirección externa** 00h. ¿Qué condiciones se deben cumplir para que no sea necesario insertar tiempos de espera TW adicionales en un acceso al puerto mencionado?

Datos: Z84C0004, fclk=4MHz

|  |                                  |
|--|----------------------------------|
| retardo de compuertas:                           | $t_{pmin} < t_p < t_{pmax}$      |
| retardo en decodificadores:                      | $t_{dmin} < t_d < t_{dmax}$      |
| Buffer triestado:                                |                                  |
| retardo de OE a dato válido:                     | $t_{OEmin} < t_{OE} < t_{OEmax}$ |
| RAM 256 bytes:                                   |                                  |
| tiempo de acceso de direcciones a datos válidos: | $t_{AAmin} < t_{AA} < t_{AAmax}$ |

### Ejercicio 2 (Ex. Ag. 94. Pr.2)

Un registrador de perturbaciones recibe periódicamente interrupciones que utiliza para leer muestras de una señal **S** a través de un puerto de entrada de 8 bits. Las muestras se van almacenando en una cola circular en memoria de 256 bytes. Cada vez que es detectada una perturbación, el registrador debe detener la adquisición 128 muestras después, de modo que quede en la cola circular un registro de la señal observada con

128 muestras antes y 128 muestras después de la perturbación. Luego de esto el sistema almacena el registro completo en el dispositivo **disco** de E/S y reinicia la adquisición.

El flanco de subida de la señal periódica **tic** indica que hay una muestra pronta para ser leída y se utiliza para generar la interrupción periódica. Las perturbaciones son detectadas por un circuito externo, que genera un pulso de nivel bajo de corta duración en la señal **pertur** que se utilizará para interrumpir al sistema.

Se pide:

- a) Diseñar el hardware del registrador en base a un microprocesador Z80. Se trabajará en el modo 1 de interrupciones, debiéndose por algún mecanismo recibir las dos fuentes de interrupciones (para leer una muestra y cuando se detecta una perturbación) a través de la entrada INT del procesador. Se dispone de latches y buffers octales, decodificadores y compuertas lógicas. El dispositivo **disco** de E/S y la memoria del sistema se suponen implementados y no se pide su diseño.

- b) Escribir la rutina de atención a la interrupción **INT**.

El programa principal tiene la estructura que se indica en el recuadro. Un valor distinto de cero en la dirección **reg\_completo** le indica al programa principal que hay un registro completo en memoria pronto para ser almacenado. La rutina de atención a la interrupción debe mantener en la dirección **ind\_cola** el índice dentro de la cola de la próxima muestra a adquirir. La cola se encuentra en memoria a partir de la dirección 8000h. La subrutina **salva\_a\_disco** se supone implementada. Recuerde que la rutina debe dar servicio a las dos fuentes de interrupción.

```

ORG 0
inicializ:   IM 1
...
lazo:  LD A, (reg_completo)
OR A
JR Z lazo
CALL salva_a_disco
LD A, 0
LD (reg_completo), A
EI
JR lazo

```

### Ejercicio 3 (Ex. Mar. 94 Pr. 2)

Un dispositivo CONMUTADOR basado en el Z80 recibe datos adquiridos en **N** canales de entrada y debe enrutarlos hacia la misma cantidad de canales de salida. Los datos de los **N** canales de entrada y de los **N** canales de salida están multiplexados en el tiempo en los buses INFOIN e INFOOUT respectivamente, ambos de 8 bits. La señal **sinc\_canal** se utilizan para sincronizar la información. El flanco de subida de **sinc\_canal** indica que se establecieron los datos correspondientes a un nuevo canal en el bus INFOIN.

El dispositivo a diseñar debe sacar por cada uno de los **N** canales de salida los datos del canal de entrada adecuado, de acuerdo a lo especificado por una **tabla de control** almacenada en memoria a partir de la dirección CONTROL. El contenido de la posición **i** de la tabla de control indica cuál es el canal de entrada que debe "conectarse" al canal **i** de salida. Para eso el dispositivo escribe en orden los datos de los canales de entrada a medida que llegan en una **tabla de datos** en memoria a partir de la dirección DATOS, de modo que en la posición **j** de la tabla siempre está almacenado el último dato recibido por el canal **j** de entrada. Con cada flanco de subida de **sinc\_canal** el la rutina de conmutación debe almacenar el dato de entrada en la posición correspondiente de la tabla DATOS y sacar por el bus de salida el dato adecuado indicado por la tabla de control.

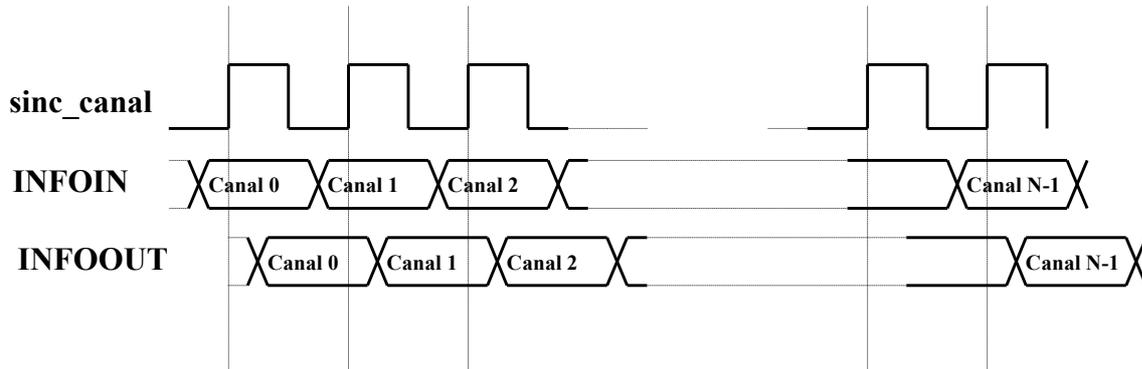
Las direcciones CONTROL y DATOS tienen sus bytes menos significativos en 0.

Para modificar una conexión, un dispositivo que supervisa al CONMUTADOR escribe en los registros DESDE y HACIA de 8 bits cada uno los números de canal de

entrada y salida que deben conectarse, y genera un pulso a 0 en la señal *modif* que debe interrumpir al Z80. La rutina de atención a la interrupción debe modificar la tabla de control de manera adecuada para establecer la conexión solicitada.

Se pide:

- Todos los puertos necesarios con su decodificación correspondiente y la circuitería asociada al pedido de interrupción del Z80.
- Realizar la rutina que efectúe en régimen la conmutación de los datos (loop infinito). Esta rutina será llamada por el programa principal un instante antes del flanco de subida de *sinc\_canal* correspondiente al Canal 0.
- La rutina de atención a la interrupción.



#### Ejercicio 4 (Ex. Dic. 93 Pr.2)

Se debe proveer a un sistema basado en el Z80 de un mecanismo para detectar errores de funcionamiento como se describe más adelante.

Para eso se utiliza un contador de 16 bits que cuenta períodos de una señal **reloj**. El contador tiene una entrada **clear**, activa en nivel bajo, que lleva la cuenta a cero, y una salida **fin** que produce un pulso a cero de duración fija cada vez que la cuenta llega al valor máximo. El valor de la cuenta en cada instante está disponible en las patas de salida **CNT[ 0..15 ]**, que deben poder leerse por programa a través de dos puertos de entrada. El contador demora un tiempo **T** en alcanzar el valor máximo desde que **clear** va nivel alto.

Los programas para este sistema deben escribirse de manera que a intervalos menores que **T** se le de un pulso de nivel bajo en la entrada **clear** a través de algún mecanismo de E/S que debe diseñarse. Si todo funciona correctamente la señal **fin** se mantiene en 1, pero si se produce algún mal funcionamiento (p. ej. por un error de programación, el sistema queda ejecutando un loop eternamente), transcurrirá un tiempo mayor que **T** y la señal **fin** bajará a 0 indicando el error.

La bajada a 0 en **fin** debe provocar una interrupción no enmascarable al microprocesador a través de la entrada NMI. La rutina de servicio a la interrupción debe preservar la dirección de retorno como se explica más adelante y resetear el Z80 por algún mecanismo hardware que deberá diseñarse (tener en cuenta el ancho mínimo del pulso de reset).

Las direcciones de retorno se van almacenando en memoria en forma consecutiva y se conservan entre un reset y otro. Para esto debe utilizarse un puntero que se almacena en la dirección PUNT, que apunta al primer lugar vacío de memoria y que debe ser incrementado en forma adecuada por la rutina de servicio a NMI. Este puntero se inicializa por algún mecanismo que se supone implementado.

Se encuentran libres las 16 direcciones más bajas de E/S para utilizarse como se desee.

- a) Dar un diagrama detallado del hardware de este sistema incluyendo el circuito de reset. Este, además de resetear al Z80, debe borrar el contador para que el sistema funcione correctamente. No incluya la decodificación de RAM y ROM que se supondrá existente.
- b) Escribir el trozo de código que debe ejecutarse a intervalos menores que  $T$  a efectos de mantener la señal **fin** en nivel alto.
- c) Escribir la rutina de atención a la interrupción.

### **Ejercicio 5 (Ex. Set. 93)**

Se quiere diseñar un sistema basado en un microprocesador Z80 que contenga 32Kbytes de ROM, 16Kbytes de RAM y 256 bytes de memoria cache. Esta última es una memoria rápida cuyo contenido es igual al de un bloque de 256 bytes consecutivos que se encuentran en la ROM a partir de una dirección múltiplo de 256.

La forma de funcionamiento del sistema será la siguiente:

- Se usará un puerto de salida para guardar la dirección de comienzo del bloque de bytes que está copiado en la memoria cache ( observar que alcanza con guardar el byte alto de la dirección).
- Cada vez que se acceda a una dirección de ROM cuyo contenido esté copiado en la memoria cache, se deberá leer de la memoria cache.
- En el caso en que no esté copiado en la memoria cache, se deberá leer de la ROM insertando 2 estados de espera, y se generará una interrupción que se encargará de copiar un nuevo bloque de ROM en la memoria cache y de actualizar el registro que contiene la dirección de comienzo del bloque.
- Para escribir en la memoria cache, ésta deberá estar mapeada en el espacio de memoria.
- En el caso de un acceso a RAM, el sistema funcionará normalmente.
- Para el buen funcionamiento del sistema es necesario un mecanismo que inhiba la lectura de la memoria cache, forzando la lectura de ROM, hasta que esté todo inicializado luego del reset. Este mecanismo no se pide resolverlo.

Se pide:

- a) Diseñar el hardware completo del sistema indicando la conexión y decodificación de los chips de memoria y puertos E/S necesarios. Se dispone de chips de RAM de 256 bytes y de 16Kbytes, chips de ROM de 32Kbytes, latches y buffers octales, comparadores de 8 bits y toda clase de compuertas lógicas y flip-flops.
- b) Escribir la rutina de atención a la interrupción que realice lo descrito antes. Para evitar que actúe el mecanismo de cache esta rutina deberá residir en RAM.

Sugerencia: a los efectos de que la rutina de atención sepa la dirección de comienzo del nuevo bloque a copiar, se debe latchear el byte alto de la dirección presente al acceder a la ROM, y luego leerlo a través de un puerto de entrada.

### **Ejercicio 6 (Ex. Feb. 93)**

Se desea diseñar un probador de chips de memorias RAM estáticas en base a un Z80. El sistema probará chips de 64K x 8. El algoritmo de prueba consistirá en escribir y verificar la secuencia 55H en toda la memoria.

El sistema dispone de 1 señal de entrada y dos señales de salida:

#### ENTRADAS

**INICIAR:** va a 1 durante un período del reloj del sistema para indicar que la memoria a probar se ha insertado en el sistema y que puede comenzar la prueba. Se ignorarán los pulsos en INICIAR que pueden aparecer durante la prueba de una memoria.

#### SALIDAS

**PROBANDO:** Vale 1 durante la prueba de una memoria.

**RAMBIEN:** Al finalizar la prueba de una memoria va a 1 si la prueba fue correcta y 0 en caso contrario; manteniendo su valor hasta iniciar una nueva prueba. No importa su valor mientras que PROBANDO = 1.

Se pide:

- a) Dar un diagrama completo del hardware necesario para el sistema incluyendo mapas de memoria y E/S. La memoria a probar deberá disponer de un zócalo (punto de conexión físico) para ella, que se deberá incluir en el diagrama. Se dispone de EPROMs de 8k x 8 con las señales de control CS OE y RAMs de 8K x 8 con las señales de control CS OE y WR. Todas las señales de control de las memorias son activas en nivel bajo. Se dispone además de los dispositivos lógicos usuales (buffers triestado, latches, flip-flops y compuertas). Fundamentar brevemente las distintas partes del diseño elegido.  
Nótese que el espacio de memoria de 64k del Z80 no es suficiente para direccionar la memoria a probar más la memoria del sistema por lo que se deberá diseñar algún mecanismo para resolver ese punto.
- b) Dar toda la programación (que se colocará en EPROM) necesaria para el funcionamiento del sistema.