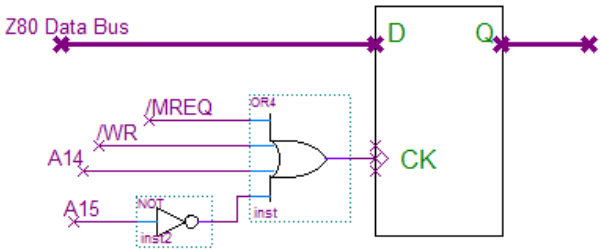
a) Utilice solo un lado de las hojas
b) Incluya un solo problema por hoja.
c) Sea prolijo.
d) Indique Nombre, CI y numere cada hoja.

Ejercicio 1 (20 pts.)

Se dispone de un registro conectado a un Z80 como indica la figura, mapeado en el espacio de memoria. Su hoja de datos especifica un mínimo requerido t_{su_req} para el tiempo de setup de los datos (desde datos estables hasta flanco de subida en CK).



Se pide:

a) Plantear todas las ecuaciones requeridas para determinar si es necesario insertar tiempos Tw. Indicar para cada parámetro, cuando corresponda, si debe considerarse el valor máximo o el mínimo. Los períodos o semiperíodos de reloj T que aparezcan en las ecuaciones se deben anotar indicando a qué período corresponden dentro del ciclo (p. ej. T1, $T2_{LOW}$, etc.). Los tiempos de los demás componentes deben referirse como: $t_{not,}$ t_{OR}.

b) Si se utiliza un Z84C0020 con un reloj de 20 MHz (Tck: 50nS) (ciclo de trabajo 50/50) y se tienen los siguientes tiempos: t_{su_req} = 50 ns, $5 \leq t_{OR} \leq 20$ ns, $0 \leq t_{not} \leq 10$ ns, indicar cuántos Tw es necesario insertar.

Ejercicio 2 (30 pts.)

Se tienen dos señales SEN1 y SEN2 en las que se generan pulsos de muy corta duración en forma aleatoria. Se agregará a un sistema existente basado en T80 la capacidad de informar en cada momento la diferencia entre la cantidad de pulsos que se han generado en ambas señales y de detectar si el valor absoluto de dicha diferencia supera un valor prefijado.

El sistema trabaja en modo 2 de interrupciones. Se agregarán dos controladores de interrupciones a los que llegarán SEN1 y SEN2 (SEN1 deberá tener mayor prioridad). Luego la rutina de atención a las interrupciones generadas por SEN1 incrementará la variable en memoria DIFERENCIA y la rutina que atiende a las interrupciones de SEN2 la decrementará. Se supondrá que este valor siempre estará en el rango -100 a 100.

Luego de la inicialización el programa principal esperará por un pulso de corta duración en la entrada START. Cuando llegue este pulso llevará a 0 la variable DIFERENCIA y quedará en loop copiando el valor de la diferencia al puerto de salida pDIF hasta que el valor absoluto de la diferencia supere un valor prefijado UMBRAL. A partir de ese momento no se modifica el puerto pDIF (aunque DIFERENCIA puede seguir cambiando si llegan más pulsos SEN1 o SEN2) y se vuelve a esperar un nuevo pulso de START para recomenzar el ciclo.

El sistema ya dispone de periféricos que hacen uso de las direcciones desde 0x00 hasta 0x3F y otros controladores de interrupciones que utilizan los vectores de interrupción 0x00, 0x02 y 0x04. Además se sabe que está equipado con una memoria ROM de 32K y una RAM de 32K.

Se pide:

a) Controladores de interrupciones, puertos y todo lo necesario (incluyendo la decodificación) para conectar SEN1, SEN2, START y pDIF.

b) Inicialización de todo lo necesario para el correcto funcionamiento del sistema luego de reset. La subrutina INI_OTROS inicializa los demás periféricos y controladores de interrupciones. Esta rutina espera que la tabla de interrupciones esté a partir de la dirección 0x9000.

c) Rutinas de atención a las interrupciones.

d) Programa principal y directivas de reserva de memoria para la tabla de interrupciones y las variables usadas.

Ejercicio 3 (50 pts.)

Se desea llevar un registro de las llamadas a subrutina realizadas en un sistema con Z80. En cada llamada a subrutina se desea saber en qué dirección comienza la subrutina invocada y a cuál dirección va a retornar (es decir, la dirección de la instrucción siguiente al CALL).

Para esto se debe idear un mecanismo hardware que genere una interrupción modo 1 cada vez que el procesador ejecuta un CALL. Luego, la rutina de atención a la interrupción, valiéndose de la información almacenada en el stack, puede obtener la dirección a la que se retornará luego del call y la dirección de comienzo de la subrutina llamada. Tener en cuenta cómo se utiliza el stack en una instrucción CALL y en interrupciones.

Estas dos direcciones deberán enviarse a una impresora, primero la dirección de la subrutina y luego la dirección de retorno del CALL. La impresora tiene una entrada DATA de 16 bits desde donde leerá el número a imprimir cada vez que reciba un flanco de subida en su entrada PRINT de un bit. La salida BUSY sube a 1 con el flanco de PRINT y vuelve a 0 un tiempo después para indicar que la impresora se encuentra disponible para recibir un nuevo dato. Se debe verificar que BUSY valga 0 antes de escribir un dato. Si BUSY = 1 el flanco en PRINT es ignorado.

Se pide:

a) Conectar al Z80 mediante puertos de entrada y salida y su correcta decodificación, la impresora anteriormente descrita. Se encuentran libres los puertos de 0x40 a 0x60. Realizar el hardware que genere una interrupción cuando el Z80 lee el código de operación de una instrucción CALL. Notar que para esto basta detectar la presencia en el bus de datos del opcode de la instrucción CALL en un ciclo M1.

b) Para el código del recuadro, completar el siguiente diagrama de tiempo que indica la secuencia de ciclos de máquina ejecutados entre la búsqueda del código de operación del CALL y el fin de la ejecución de la primera instrucción de la rutina de atención a interrupciones (que supondremos es PUSH AF). Etiquetar los ciclos como M1, Rdm (lectura memoria) y Wrm (escritura en memoria), INTA, etc.. No olvidar los ciclos M que se producen en la secuencia de atención a la interrupción.

```
           inst_1
dir_call1:CALL sub_rut1
           inst_3
           .....
           .....
sub_rut1:  inst_2
           .....
           .....
           RET
```

Para los ciclos de escritura a memoria destinada al stack, indicar cuál es el dato que se guarda.

Ciclo	M1	RDm	RDm								
Al stack	--	--	--								

c) Indicar cuál será el contenido del stack luego de ejecutar la instrucción PUSH AF mencionada en la parte anterior.

d) Escribir la rutina de atención a la interrupción que envíe a la impresora primero la dirección de la subrutina y luego la dirección de retorno del CALL. Antes de enviar cada uno de los datos, debe verificarse que la impresora esté lista.
Indicar mediante directiva ORG su ubicación en memoria.

Nota: recordar que no hay una instrucción que copie el registro SP a otro registro pero puede usarse para eso la suma o la transferencia de 16 bits a memoria.