Sistemas Digitales II

2º curso del Grado en Ingeniería Electrónica Industrial

Convocatoria de Junio (10-06-2026)

APELLIDOS: _____

NOMBRE: _____ DNI Nº: _____

---

1 (3.5 puntos). Diseñar un mapa de memoria para un procesador con un espacio de direccionamiento de 16K y bus de datos de 16 bits. Los bloques necesarios son:

- Un bloque de ROM de 4K que comienza en la dirección 0.
- Un bloque de EPROM a continuación de la ROM.
- Un bloque de RAM a continuación de la EPROM.
- Un módulo de E/S (E/S-1) que ocupa 256 posiciones.
- Un hueco libre de 768 posiciones.
- Un módulo de E/S (E/S-2) que ocupa 256 posiciones.

Circuitos integrados disponibles (todos con habilitación CS activa a nivel bajo, excepto los módulos de E/S que tienen dos líneas de habilitación CS1 y CS2 cada uno de ellos, activas a nivel alto):

- Memorias ROM de 4K × 8.
- Memorias EPROM de 1K × 16.
- Memorias RAM de 4K × 4.
- Dos módulos de E/S de 256 posiciones cada uno.

Se pide:

a) Demostrar que caben todos los bloques que se piden en el espacio de direccionamiento disponible. Si es así, calcular de forma clara los integrados necesarios en cada bloque. *(0.5 puntos)*

b) Diseñar el mapa de memoria del sistema. *(0.75 puntos)*

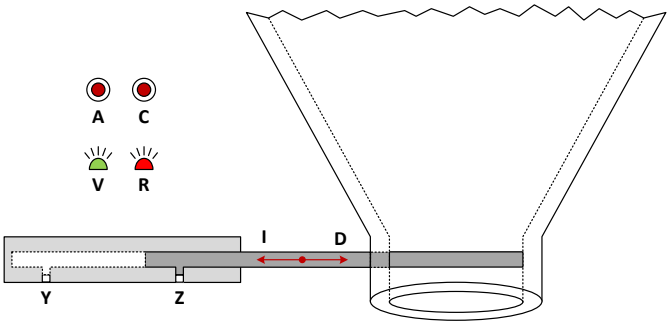
c) Obtener la tabla de direcciones inicial y final de cada circuito en binario y hexadecimal. *(0.75 puntos)*

d) Definir con qué combinación de líneas se realiza la decodificación y porqué. *(0.25 puntos)*

e) Dibujar el diagrama lógico completo detallado. *(1 punto)*

f) Definir un módulo VHDL para realizar la decodificación con un dispositivo programable. *(0.75 puntos)*

2 (3.5 puntos).- Se desea automatizar el proceso de apertura y cierre de una trampilla que se ha colocado en la salida de una tolva, como se muestra en la figura siguiente.



Para ello, se han instalado los siguientes elementos:

- Dos pulsadores **A** y **C** que generan niveles altos al ser accionados por el usuario de la tolva. ***Se supone que el usuario nunca accionará estos dos botones a la vez***.
- Dos leds, uno verde y otro rojo, que se iluminan al aplicar niveles altos a las señales **V** y **R**, respectivamente.
- Un cilindro con dos líneas de control **I** y **D** que provocan el desplazamiento de su émbolo hacia la izquierda y hacia la derecha, respectivamente. El desplazamiento horizontal de este émbolo provoca a su vez la apertura y el cierre de la trampilla.
- Dos finales de carrera **Y** y **Z** que adoptan niveles altos cuando el émbolo del cilindro está totalmente recogido y totalmente extendido, respectivamente.

El funcionamiento del sistema debe ser el siguiente:

- Inicialmente hay que cerrar la trampilla si no está en esa posición.
- Si la trampilla se encuentra totalmente cerrada y el usuario pulsa el botón A, ésta comenzará a abrirse 5

SEGUNDOS DESPUÉS DE SOLTAR ESE PULSADOR.

- Si la trampilla se está abriendo y antes de estar totalmente abierta el usuario pulsa el botón C, ésta comenzará a cerrarse, PERO HARÁ UNA PAUSA PREVIA DE 6 SEGUNDOS DESDE QUE SE SUELTA C.
- Si la trampilla se encuentra totalmente abierta y el usuario pulsa el botón C, ésta comenzará a cerrarse 5 SEGUNDOS DESPUÉS DE SOLTAR EL PULSADOR.
- Si la trampilla se está cerrando y antes de estar totalmente cerrada el usuario pulsa el botón A ésta comenzará a abrirse, PERO HARÁ UNA PAUSA PREVIA DE 3 SEGUNDOS DESDE QUE SE SUELTA A.
- Si la trampilla en su movimiento alcanza alguna de sus posiciones límites (izquierda o derecha), ésta se detendrá automáticamente.
- Mientras la trampilla esté totalmente cerrada permanecerá encendido el led rojo y mientras esté totalmente abierta se encenderá el led verde.

Realizar el sistema de control de la trampilla mediante un diagrama ASM. Se pide:

a) Cálculos previos detallados necesarios. El reloj disponible es de 25 MHz. ***(0.25 puntos)***
b) El diagrama ASM, indicando cuales son los bloques ASM obtenidos. ***(2 puntos)***
c) Módulo VHDL que represente el funcionamiento del diagrama ASM obtenido, respetando los nombres asignados a los bloques ASM. ***(1.25 puntos)***

3 (3 puntos). Diseñar un sistema basado en un microcontrolador *Picoblaze* con las siguientes características: dispone de 4 puertos de entrada numerados como 14, 16, 29 y 30. Dispone de 3 puertos de salida numerados como 127, 220 y 223. El sistema debe resolver estas operaciones:

a) Obtendrá los valores de entrada de los 4 puertos de entrada y guardará estos valores en las mismas posiciones de la RAM.
b) Recuperará el primer dato (puerto 14) y si es igual a 30 o a 40, entonces le sumará 5 y lo enviará al puerto de salida 127. Si no es así, enviará un '1' a ese puerto de salida.
c) Obtendrá desde la RAM los dos datos siguientes (puertos 16 y 29) y los sumará. Si se produce acarreo, entonces restará 50 al resultado y lo enviará al puerto de salida 220. Si no hay acarreo, colocará un '0' en ese puerto de salida.
d) Recuperará el valor almacenado del puerto 30, y si el bit más significativo es un '1', entonces pondrá a uno todas las salidas del puerto 223, y si no es así, sumará 10 al valor del puerto 30 y lo colocará en el puerto 223.

Si se produce una interrupción, entonces debe borrar todos los puertos de salida y toda la RAM, y volverá a empezar desde el principio. Si no se produce una interrupción, entonces realizará todos los pasos de forma continua, desde el a) hasta el d). Se pide:

a) Diagrama lógico DETALLADO, incluyendo la memoria de programa. ***(1 punto)***
b) Programa en ensamblador debidamente comentado. ***(2 puntos)***

Instrucción				Comentario
ADD sX, sY	ADD sX, kk	ADDC sX, sY	ADDC sX, kk	Operaciones sin y con acarreo (finalizadas en C)
SUB sX, sY	SUB sX, kk	SUBC sX, sY	SUBC sX, kk	
SL0 sX	SL1 sX	SLX sX	SLA sX	Desplazamientos introduciendo un 0, 1, el último elemento y el bit de acarreo
SR0 sX	SR1 sX	SRX sX	SRA sX	
RL sX	RR sX			Rotación
AND sX, sY	AND sX, kk			Instrucciones lógicas
OR sX, sY	OR sX, kk			
XOR sX, sY	XOR sX, kk			
COMP sX, sY	COMP sX, kk			Instrucciones de verificación
TEST sX, sY	TEST sX, kk			
LOAD sX, sY	LOAD sX, kk			Movimiento de registros
IN sX, (sY)	IN sX, kk			Operaciones con puertos
OUT sX, (sY)	OUT sX, kk			
FETCH sX, (sY)	FETCH sX, kk			Operaciones con memoria
STORE sX, (sY)	STORE sX, kk			
JUMP etiq	JUMP (N)Z, etiq	JUMP (N)C, etiq		Instrucciones de saltos
CALL etiq	CALL (N)Z, etiq	CALL (N)C, etiq		
RET	RETI			
EINT				Habilitación Y deshabilitación de interrupciones
DINT				