GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL

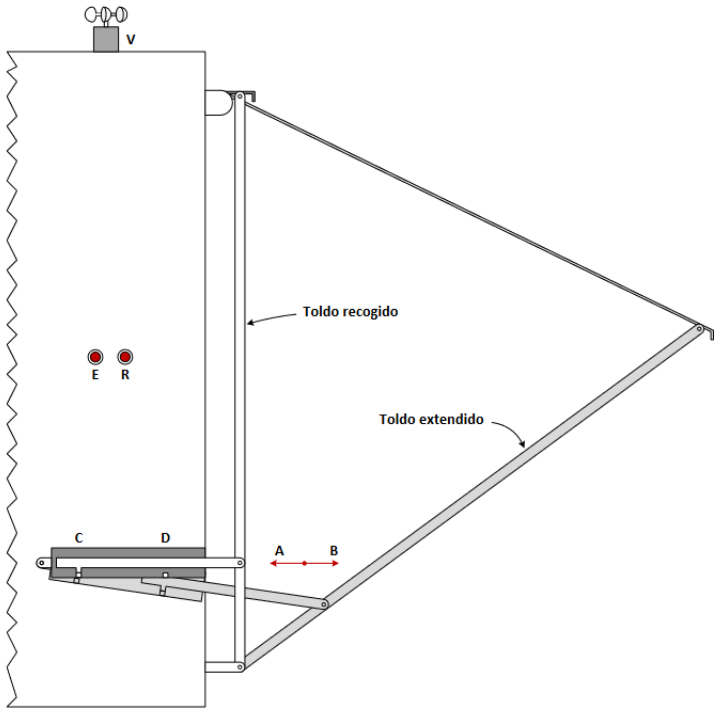
SISTEMAS DIGITALES II (segundo curso)

Examen de Junio (13/06/2024)

APELLIDOS:_____

NOMBRE: _____ DNI Nº_____

1. ***(3,5 puntos)*** Se desea automatizar el funcionamiento del toldo cuya vista lateral se muestra en la figura:



Para ello se han instalado los siguientes elementos:

- Dos pulsadores **E** (extender) y **R** (recoger) que proporcionan niveles altos al ser accionados por el usuario.
- Un anemómetro cuya salida **V** adopta un nivel alto mientras la velocidad del viento supera un determinado valor.
- Un cilindro con dos líneas de control **A** y **B** que provocan la recogida y la salida del émbolo, respectivamente, mientras se esté aplicando a las mismas un nivel alto.
- Dos finales de carrera **C** y **D** que adoptan niveles altos cuando el émbolo del cilindro está totalmente recogido y totalmente extendido, respectivamente.
- Un reloj de impulsos de 25 MHz de frecuencia.

El modo de funcionamiento del toldo debe ser el siguiente:

- Al poner en marcha el sistema el toldo se encontrará ya recogido.
- Si estando el toldo recogido y en ausencia de viento, se activa el pulsador **E**, éste comenzará a extenderse, y lo hará totalmente si antes no se produce ninguna incidencia.
- Si, estando el toldo extendido se activa el pulsador **R**, éste se recogerá totalmente.
- En cualquier momento, si mientras el toldo se está extendiendo se detecta una racha de viento, se debe parar el toldo. El sistema esperará 5 segundos, comprobará si sigue la racha de viento y, en caso de que ésta se mantenga, entonces el toldo se recogerá automáticamente por completo. Si no hay viento ya, el toldo seguirá extendiéndose.
- En cualquier momento, si mientras el toldo está totalmente extendido se detecta una racha de viento el sistema esperará 3 segundos, comprobará si sigue la racha de viento y, en caso de que ésta se mantenga, entonces el toldo se recogerá automáticamente por completo. Si no sigue el viento, el toldo permanecerá extendido.

Se pide:

a) Diagrama ASM de control *(2,5 puntos).*
b) Describir el circuito mediante VHDL *(1 punto).*

2. ***(3 puntos)*** Diseñar un sistema basado en el microcontrolador *Picoblaze* que permita resolver el circuito anterior. Se dispone de los siguientes elementos hardware:

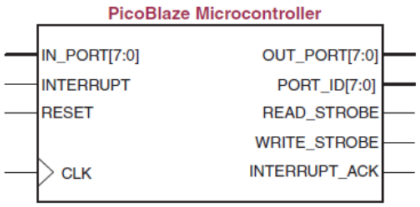
- Un módulo contador **CTR5** que cuenta 5 segundos con las siguientes entradas y salidas:
  - Una entrada **RESET** para ponerlo a cero cuando recibe un pulso positivo (0 -> 1 -> 0).
  - Una entrada **INICIO_5s** que cuando recibe un flanco de bajada empieza la cuenta de 5 segundos.

- Una salida **FIN_CUENTA_5s** que toma el valor ALTO cuando el contador acaba la cuenta de 5 segundos. Toma automáticamente un valor BAJO cuando se resetea el módulo.

- Un módulo contador **CTR3** que cuenta 3 segundos con las siguientes entradas y salidas:
  - Una entrada **RESET** para ponerlo a cero cuando recibe un pulso positivo (0 -> 1 -> 0).
  - Una entrada **INICIO_3s** que cuando recibe un flanco de bajada empieza la cuenta de 3 segundos.
  - Una salida **FIN_CUENTA_3s** que toma el valor ALTO cuando el contador acaba la cuenta de 3 segundos. Toma automáticamente un valor BAJO cuando se resetea el módulo.

  Se pide:
  - el diagrama lógico detallado *(1 punto)*
  - la programación en ensamblador COMENTADA equivalente a los primeros 7 bloques ASM obtenidos **en el apartado "a" del problema 1**. (*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				



3. **(3,5 puntos)** La siguiente figura muestra la Unidad de Procesado de Datos de un microprocesador con arquitectura von Neumann. Como puede observarse, el bus de datos (D7-D0) es de 8 bits y el de direcciones (A15-A0) de 16 bits. Los registros visibles son los siguientes:

- **PC** (Program Counter): es de 16 bits y está formado por la concatenación de dos registros de 8 bits (PCH y PCL).
- **AC** (Acumulador): es de 8 bits.
- Además, la unidad de procesado dispone de los siguientes registros ocultos:
- **MAR** (Memory Address Register): de 16 bits, está formado por la concatenación de dos registros de 8 bits: MARH y MARL.
- **MDR** (Memory Data Register): de 8 bits, su finalidad es servir como registro intermedio entre el procesador y la memoria externa. Todo dato que entre o salga del procesador deberá ser almacenado previamente en MDR.
- **IR** (Instruction Register): es de 8 bits.
- **RT** (Registro Temporal): es de 8 bits y es utilizado para el cálculo de operaciones intermedias.
- **SUB** (Subroutine register): Es de 16 bits y sirve para guardar el valor del PC cuando se hace una llamada a una subrutina, de forma que el procesador pueda volver a la siguiente posición del programa desde donde se llamó a la subrutina.

Las líneas ENx de cada elemento permiten habilitar el elemento. Si no está habilitado, el elemento se encuentra en estado de alta impedancia.

Las posiciones de memoria tienen un ancho de 8 bits, y cada instrucción puede ocupar hasta tres posiciones consecutivas: la primera indica el código de operación, después vendrá o ningún operando, o un dato de 8 bits, o una dirección de memoria de 16 bits dividida en dos partes de 8 bits (primero la parte alta y luego la parte baja de la dirección). Se pide:

- Indicar la secuencia de microoperaciones necesarias para realizar las instrucciones de la tabla que se presenta a continuación e indicar cuales corresponden al ciclo de carga y decodificación y cuales al ciclo de ejecución. *(2.25 puntos)*
- Diseñar la unidad de control (arquitectura multiplexor – memoria – contador) y conectarla a la unidad operativa. *(0,75 puntos)*
- Codificar en binario las primeras 6 microinstrucciones *(0,5 puntos).*

INSTRUCCIÓN	CODIGO DE OPERACIÓN	FUNCIÓN
LDAC $dir	00001000	Carga en el acumulador (AC) el dato que se encuentra en la dirección $dir de la memoria.
ADD (dato)	00001001	Carga en el acumulador (AC) el resultado de sumar su contenido con el dato que se indica en la instrucción.
MUL3	00001010	Multiplica por tres el valor del acumulador (AC)
SUBR $dir	00001011	Salta a la subrutina que se encuentra en la dirección $dir de la memoria.
RETS	00001100	Vuelve desde subrutina.
JUMP Z $dir	00001101	Salta a la dirección $dir si la última operación dio como resultado 0 en la última operación de la ALU.
STAC $dir	00001110	Guarda en la dirección $dir el contenido del AC.
STOP	00001111	Detiene el programa.

C1 C0	*Operación de la ALU*
0 0	*L + R*
0 1	*L - R*
1 0	*L*
1 1	*R + 1*

C10 C9	*MDR*
1 0	*Carga datos desde el bus interior y se configura como salida hacia el exterior*
0 1	*Carga datos desde el bus de datos y se configura como entrada desde el exterior*

C14 C13	*PC*
0 1	*Incrementa el conjunto PCH PCL*
1 0	*Resetea el conjunto PCH PCL*

