Universidad de Huelva

ETSI ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL

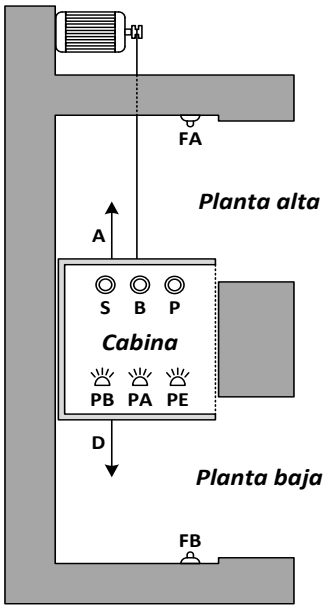
SISTEMAS DIGITALES II

Examen de Junio (problemas) (19/6/2025)

APELLIDOS: _____

NOMBRE: _____ DNI Nº: _____

1 (3 puntos).- Se desea automatizar el funcionamiento de un montacargas para trasladar materiales entre las dos plantas de una nave industrial, como se muestra en la siguiente figura.



Para ello se han instalado los siguientes elementos:

- Tres pulsadores **S**, **B** y **P** que al ser accionados por el usuario producen niveles altos para generar las órdenes de subida, de bajada y de parada de emergencia, respectivamente. ***El usuario nunca activará varios de estos pulsadores simultáneamente***.
- Dos finales de carrera **FB** y **FA** que adoptan niveles altos cuando la cabina llega a la planta baja y a la planta alta, respectivamente.
- Un sistema de tracción dotado de dos señales de control, **A** (ascender) y **D** (descender), que al ser activadas provocan el movimiento de la cabina en el sentido correspondiente.
- Tres indicadores luminosos **PB**, **PA** y **PE**, que se activan al aplicarles niveles altos.

El funcionamiento del sistema debe ser el siguiente:

- Si la cabina está parada, al activar los pulsadores **S** o **B**, ésta subirá o bajará hasta alcanzar la planta alta o la planta baja, respectivamente.
- Si la cabina se encuentra en una posición intermedia y se activa el pulsador P se producirá una parada de emergencia. A partir de ahí, la cabina permanecerá parada **7 segundos**. Una vez transcurrido este tiempo, el usuario podrá pulsar **S** o **B** para continuar el movimiento. Se dispone de un reloj de 25 MHz.
- Si la cabina se encuentra en la planta baja deberá activarse **PB**, si está en la planta alta **PA** y si está parada en una posición intermedia, **PE**.

Realizar el sistema de control del proceso. Se pide:

a) Diagrama ASM de funcionamiento del sistema de control *(1,5 puntos). Adjuntar cálculos que hayan sido necesarios de forma detallada.*

b) Módulo VHDL que represente el diagrama ASM diseñado *(1,5 puntos).*

2. **(2,5 puntos)** Se desea diseñar un sistema hardware que incluya un procesador embebido *picoBlaze* y que tenga las siguientes características:

- Ocho puertos de entrada (puertos número 1, 2, 3, 4, 5, 6, 7 y 8).
- 4 puertos de salida (puertos 1, 2, 3 y 4).

Se pide:

- Diagrama hardware del sistema de control diseñado con el microcontrolador *picoBlaze*, detallando **bien** las conexiones, las entradas y salidas, así como la memoria de programa. *(1 punto).*
- *(1,5 puntos)* Escribir un programa en ensamblador que lea los datos presentes en los 8 puertos de entrada y los guarde en las primeras 8 posiciones de la memoria interna RAM. Después, el programa realizará las siguientes tareas con los datos obtenidos:
  - I. Restará los datos que se obtuvieron desde los puertos de entrada 1 y 2 (dato puerto 1 menos dato puerto 2) y depositará el resultado en el puerto de salida 1.
  - II. Realizará la operación OR de los datos obtenidos desde los puertos 3 y 4 y depositará el resultado en el puerto 2.
  - III. Sumará los datos obtenidos desde los puertos de entrada 5 y 6. Si el resultado proporciona un acarreo, entonces poner a cero el puerto de salida 3. Si no hay acarreo, enviar el resultado de la operación AND de los puertos 5 y 6 al puerto 3.
  - IV. Finalmente realizar la operación OR exclusiva de los puertos 7 y 8, y enviar el resultado al puerto 4 si el resultado es distinto de cero. En caso contrario pondrá todos los bits de puerto 4 a '1'.
  - V. Volver al paso I.

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. **(2 puntos)** Diseñar un mapa de memoria para una CPU con bus de datos de 16 bits y con un espacio de direccionamiento máximo de 1 Mega posiciones que sitúe los siguientes módulos:

- Un bloque PROM de 128K posiciones que termina al final del mapa de memoria (entradas de control CE, activa a nivel alto).
- Un bloque RAM de 256K posiciones que empiece en la posición 20000 del mapa de memoria (entradas de control CE activa a nivel alto y WE activa a nivel bajo).
- Un circuito de E/S (E/S-1) que ocupa 64K posiciones de memoria a partir de la posición 80000 (entradas de control CE activa a nivel bajo y WE activa a nivel bajo).

- Un circuito de E/S (E/S-2) que ocupa 32K posiciones de memoria y comienza al principio del mapa de memoria (entradas de control CE activa a nivel bajo y WE activa a nivel bajo).
  Se pide:
  - La tabla del mapa de memoria, dirección inicial y final de cada módulo en hexadecimal.
  - Diagrama lógico simplificado del circuito.

  Los bloques PROM disponibles son de 32Kx16. Los bloques RAM disponibles son de 128Kx8.

4. **(2,5 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.

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:

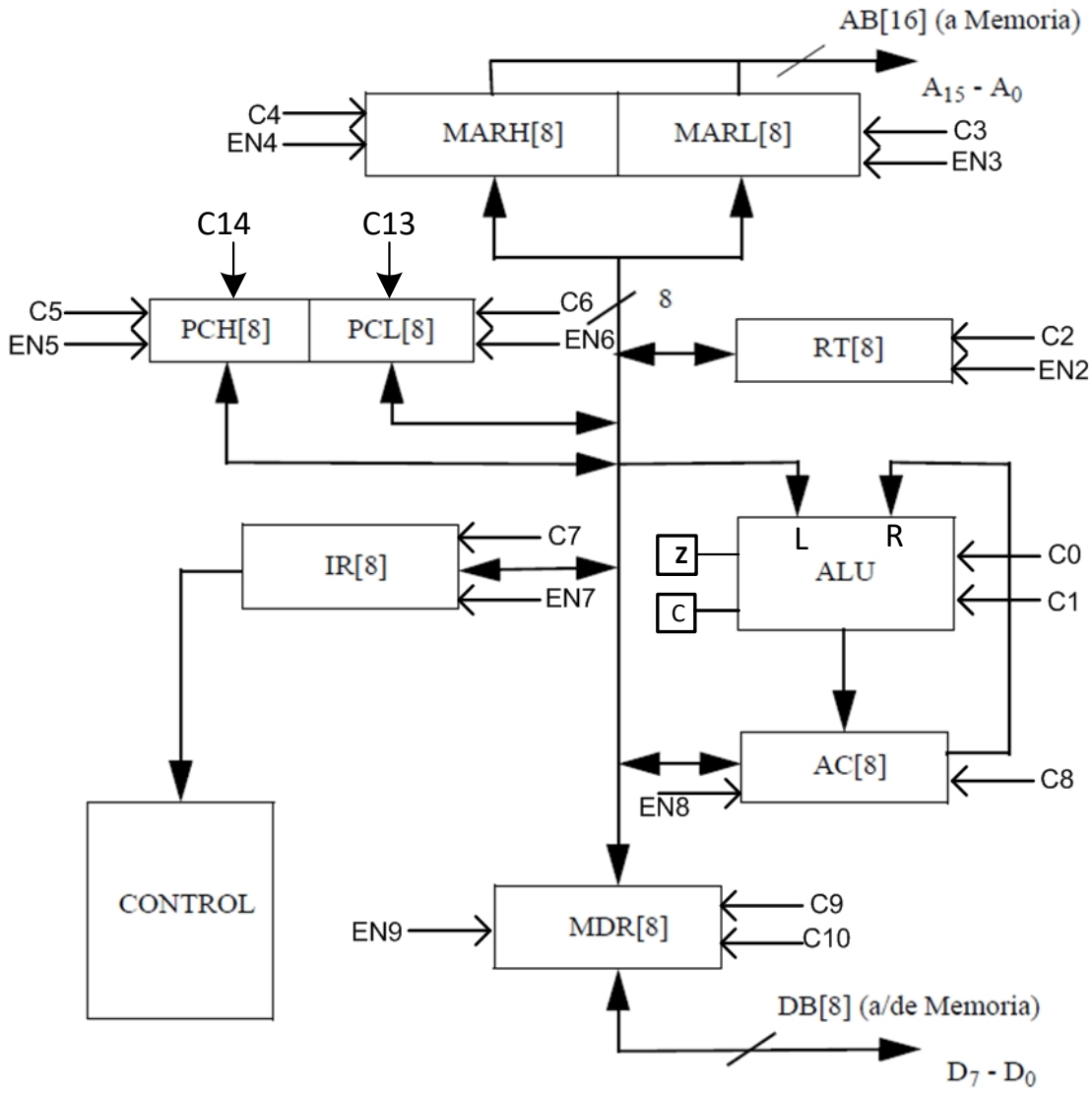
- Indicar la secuencia de microoperaciones necesarias para realizar las instrucciones de la tabla que se presenta a continuación e indicar cuales corresponden al ciclo de carga y decodificación y cuales al ciclo de ejecución. *(2 puntos)*
- Diseñar la unidad de control (arquitectura multiplexor – memoria – contador) y conectarla a la unidad operativa. *(0,5 puntos)*

INSTRUCCIÓN	CODIGO DE OPERACIÓN	FUNCIÓN
LDAC <dato>	00000001	Carga en el acumulador (AC) el dato que se indica en la instrucción.
SUB $dir	00000011	Carga en el acumulador (AC) el resultado de restar su contenido con el dato que se encuentra en la dirección indicada por la instrucción.
NAND <dato>	00000100	Realiza la operación NAND entre el AC y el dato indicado, cargando el resultado en AC
JUMP C $dir	00000101	Salta a la dirección $dir si la última operación dio como resultado 0 en la última operación de la ALU.
JUMP $dir	00000010	Salta a la dirección $dir
STAC $dir	00000110	Guarda en la dirección $dir el contenido del AC.
STOP	00000111	Detiene el programa.

C1 C0	*Operación de la ALU*
0 0	*NOT L*
0 1	*L - R*
1 0	*L*
1 1	*L AND R*

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*


AB[16] (a Memoria)
A15 - A0
C4
EN4
MARH[8]
MARL[8]
C3
EN3
C14
C13
C5
EN5
PCH[8]
PCL[8]
C6
8
EN6
RT[8]
C2
EN2
IR[8]
C7
EN7
Z
C
L
R
ALU
C0
C1
AC[8]
C8
EN8
CONTROL
EN9
MDR[8]
C9
C10
DB[8] (a/de Memoria)
D7 - D0