

Examen final de IC

Curso 2001-2002 Q2

3 de Julio de 2002

- **Duración del examen: 3 horas**
- **Hay que resolver cada problema en HOJAS SEPARADAS**
- **El problema número 3 se entrega en la misma hoja del enunciado**
- **No se puede usar calculadora ni teléfono móvil**
- **Sólo se pueden tener los chuletarios de puertas y de la Máquina Rudimentaria**
- **Habrà una solución del examen disponible en la página web de la asignatura a partir de mañana**
- **Las notas se publicarán el 8 de Julio de 2002 por la tarde**
- **La revisión del examen será el 9 de Julio de 2002 a las 15:00 en la sala D6-004**
- **Las notas definitivas se publicarán el 10 de Julio de 2002**

Problema 1 (3.5 puntos)

a) Se desea diseñar un Circuito Lógico Combinacional que calcule la operación:

$$C = 3*(A - B)$$

Sabiendo que A y B son números **enteros** representados en **Complemento a dos** con **dos bits** cada uno, y C es un **entero** representado en **Complemento a dos** con **tres bits**.

El sistema también debe generar como salida **un bit de OVERFLOW** que se activa a 1 en caso de que el resultado no pueda representarse en los 3 bits de C. En este caso, el valor de C no importa.

a.1) Construir la Tabla de Verdad de este sistema. **(Objetivos 5.5 y 6.1)**

(Puede ser de ayuda construir una tabla adicional en la que las variables se representen en decimal).

a.2) Sintetizar las funciones lógicas de:

- C_0 utilizando un Multiplexor del tamaño adecuado
- C_1 con un Decodificador de tamaño adecuado y puertas lógicas
- C_2 a dos niveles de forma mínima

(Objetivo 6.4.a)

b) Considerar ahora la operación: **$D = 3*(A-B) \bmod 16$**

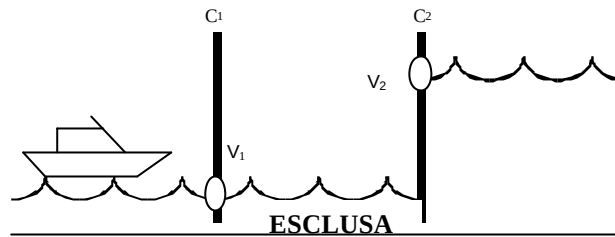
suponiendo que A y B son números **enteros** de **8 bits** cada uno.

b.1) Indicar claramente el número de bits que debe tener la salida del circuito para que **no se produzca Overflow** en ningún caso.

b.2) Diseñar un Circuito Lógico Combinacional que realice esta operación. Realizar el diseño a nivel de bloques, utilizando únicamente elementos del chuletero. Indicar claramente el tamaño de cada uno de los bloques utilizados. **(Objetivo 6.4)**

Problema 2 (3.5 puntos)

Se desea diseñar el sistema para controlar la esclusa de un río. Las esclusas permiten navegar en ríos que tienen desniveles en su cauce. El sistema de funcionamiento se basa en mantener dos compuertas independientes, C_1 y C_2 , de forma que a un lado de cada compuerta hay un nivel diferente de río. La figura muestra un ejemplo.



Existe un sistema diferente de control de la esclusa para cada sentido de movimiento (ascendente y descendente), pero **únicamente diseñaremos uno de ellos: El ascendente**. El sistema de control de esclusas que hay que diseñar consta de los siguientes elementos:

- Dos compuertas, C_1 y C_2 , controladas respectivamente por las señales de control q_1 y q_2 . Si $q_1=1$ la compuerta C_1 se abre y el barco puede entrar o salir de la esclusa. En caso contrario, la compuerta está cerrada. La señal q_2 funciona de forma análoga.
- Una válvula en cada compuerta, V_1 y V_2 , controladas respectivamente por las señales v_1 y v_2 . Estas válvulas permiten llenar o vaciar la esclusa, habilitando así la navegación por el río. Las válvulas son las que permiten el paso del agua desde el río hacia la esclusa y viceversa. Las compuertas **SOLAMENTE** se abren cuando el nivel de la esclusa es adecuado para dejar pasar el barco. Por ejemplo, la compuerta C_1 solamente se abrirá cuando el nivel de la esclusa sea igual al nivel inferior del río.
- Cuando $v_1=1$, el agua sale de la esclusa hacia la zona de menor altura. Cuando $v_1=0$, la válvula está cerrada y no hay flujo de agua de la esclusa hacia el exterior.
- Cuando $v_2=1$, la esclusa se inunda con agua procedente de la zona de mayor altura.
- Una entrada de petición de movimiento del barco, *petición*, que vale 1 cuando un barco se acerca a la esclusa y quiere atravesarla, y 0 en caso contrario. Esta señal permanece a 1 mientras el barco está atravesando la esclusa.
- Una entrada, *hay_barco*, que le indica si el barco está dentro (*hay_barco*=1) o fuera de la esclusa (*hay_barco*=0). Esta entrada funciona de la siguiente manera: Cuando el barco está fuera, vale 0, y se mantiene a 0 mientras está entrando. Cuando está completamente dentro, *hay_barco* se pone a 1. Cuando el barco comienza a salir, la señal *hay_barco* permanece a 1 hasta que el barco ha salido completamente. En ese momento, pasa a valer 0.
- Una entrada, *esclusa_llena*, que le indica si el agua en el interior de la esclusa ha alcanzado el nivel alto del río (*esclusa_llena*=1) o no (*esclusa_llena*=0).

Si un barco desea ir de la zona de menos altura a la de mas altura, se parte de **un estado inicial** en la que compuertas están cerradas y la esclusa está al nivel bajo del río o se está vaciando. Cuando hay un barco atravesando la esclusa, **un sistema externo se encarga de que no lleguen nuevas peticiones hasta que el barco haya salido y la esclusa esté vacía**.

El funcionamiento del sistema es el siguiente:

- Se abre la compuerta C_1 y el barco entra en la esclusa.
- Se cierra la compuerta C_1 cuando el barco ha entrado en la esclusa. Como el cierre se produce muy rápidamente, desde que comienza a cerrarse la compuerta se empieza a inundar la esclusa hasta que el agua llega al nivel alto del río. En ese momento, la compuerta C_1 siempre está completamente cerrada.
- Se abre la compuerta C_2 y el barco avanza hasta salir completamente de la esclusa.
- Se cierra la compuerta C_2 y comienza a vaciarse la esclusa para regresar a la situación inicial de partida.

- Indicar claramente cuales son las entradas y salidas del circuito de control de la esclusa.
- Dibujar el grafo de estados del circuito, indicando claramente cuál es el significado de cada bit. **(Objetivo 6.5).**
- Suponiendo que la señal *petición* se comporta como una señal de inicio **asíncrona**, que se mantiene a 1 durante todo el tiempo que el barco invierte en atravesar la esclusa, dibujar la tabla de transiciones entre estados y la tabla de salidas del circuito. Indicar claramente la codificación escogida para los estados. **(Objetivo 6.6)**
- Implementar el sistema usando **un decodificador para las salidas** y **una memoria ROM** del tamaño adecuado y biestables D **para las transiciones entre estados**. Indicar claramente el tamaño y contenido de la ROM. No os olvidéis de incluir en el circuito la señal *petición*. **(Objetivo 6.7).**

Nombre: _____ DNI: _____
 Apellidos: _____
 Grupo: _____

Problema 3 (3 puntos)

- a. El siguiente cuadro describe un programa escrito en lenguaje ensamblador y máquina de la MR. Completa los cuadros que se encuentran en blanco: **(Objetivo 7.3)**

Dirección	Instrucción en LM	M[@]: LM	M[@] en hexadecimal
00h		0000 0000 0110 0111	
01h		0000 0000 0000 0000	
02h		0000 0000 0000 0000	
03h	LOAD 0(R0),R1		
04h	BNE 07h		
05h			5176h
06h		0000 1101 0011 0101	
07h	BLE 05h		
08h	ADD R1,R1,R1		
09h	STORE R0,0(R1)		
0Ah	BR 06h		
0Bh	STORE R3,2(R0)		

- b. El cuadro siguiente ilustra el comportamiento de la Unidad de Proceso de la MR, ciclo a ciclo, durante la ejecución de las primeras instrucciones del programa anterior. Rellena los cuadros blancos del cronograma indicando en cada ciclo el estado de la UC simplificada de la MR. En caso que un valor no pueda ser determinado, utiliza una X para así indicarlo. Considera que la primera instrucción a ejecutar está en la dirección 03h. **(Objetivo 8.1)**

Estado	FETCH							
M@	03h	X						08h
Mout	0800h	X						C924h
Min	X							X
Din	X	X						X
PC	03h							08h
IR	X							9805h
R@	X	X						
RA	X	X						
RZ	X	X						0
RN	X	X						0
R1	X	X						