

APELLIDOS : \_\_\_\_\_

NOMBRE: \_\_\_\_\_ D.N.I.: \_\_\_\_\_

- **LEER ATENTAMENTE EL ENUNCIADO VARIAS VECES.**
- RAZONAR CADA UNA DE LAS DECISIONES ADOPTADAS TANTO EN EL HW COMO EN EL SW.
- CUIDAR LA PRESENTACIÓN EVITANDO ESQUEMAS ININTELIGIBLES, LETRA INDESCIFRABLE Y EXPRESIONES MALSONANTES.
- UTILIZAR COMENTARIOS EN LA ELABORACIÓN DEL CÓDIGO SW.

P1. (5 puntos)

Diseñar una máquina secuencial síncrona que opere de acuerdo con las siguientes características:

- Dos entradas de datos binarios en serie A y B, independientes y sincronizadas con la señal de reloj.
- Dos salidas E (error) y S (signo)
- Tendrá el siguiente funcionamiento: El autómata deberá controlar en todo momento la diferencia entre las cantidades de unos recibidas por las entradas A y B y el signo de esa diferencia. De forma que si por A se han recibido más unos que por B el signo será positivo ( $S = 1$ ), si se han recibido igual o menos será negativo ( $S = 0$ ). En cualquier caso, si la diferencia alcanza el valor 3 (en cualquiera de los dos sentidos), deberá activarse la señal de error ( $E = 1$ ) y volver al estado inicial.
- El autómata se realizará utilizando únicamente una memoria ROM del tamaño adecuado y biestables D.

En el proceso de diseño deberán aparecer claramente las siguientes fases, en el orden que se considere oportuno:

- Definición y codificación de entradas, salidas y estados.
- Representación del diagrama de flujo.
- Tabla de transiciones o estados.
- Mapa de memoria y tamaño en bits de la memoria mínima necesaria.
- Implementación del circuito (esquema lógico).

Nota: cualquier interpretación del enunciado debe justificarse claramente.

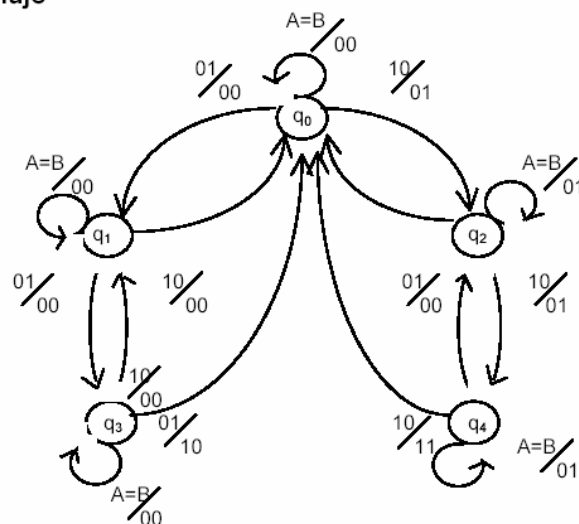
Solución:

ENTRADAS	A	B	Significado
	0	0	Valores correspondientes a ambas entradas
	0	1	i.d.
	1	0	i.d.
	1	1	Id.

SALIDAS	E	S	Significado
	0	0	Diferencia 0 o favorable a B sin alcanzar 3
	0	1	Diferencia favorable a A sin alcanzar 3
	1	0	B ha recibido tres uno más que A
	1	1	A ha recibido tres uno más que B

ESTADOS	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Significado
	0	0	0	q <sub>0</sub> ⇒ A = B
	0	0	1	q <sub>1</sub> ⇒ 1 B > A
	0	1	0	q <sub>2</sub> ⇒ 1 A > B
	0	1	1	q <sub>3</sub> ⇒ 2 B > A
	1	0	0	q <sub>4</sub> ⇒ 2 A > B

Diagrama de flujo



- Tabla de estados

		ENTRADAS AB			
		00	01	10	11
	q <sub>0</sub>	q <sub>0</sub> /00	q <sub>1</sub> /00	q <sub>2</sub> /01	q <sub>0</sub> /00
	q <sub>1</sub>	q <sub>1</sub> /00	q <sub>3</sub> /00	q <sub>0</sub> /00	q <sub>1</sub> /00
	q <sub>2</sub>	q <sub>2</sub> /01	q <sub>0</sub> /00	q <sub>4</sub> /01	q <sub>2</sub> /01
	q <sub>3</sub>	q <sub>3</sub> /00	q <sub>0</sub> /10	q <sub>1</sub> /00	q <sub>3</sub> /00
	q <sub>4</sub>	q <sub>4</sub> /01	q <sub>2</sub> /01	q <sub>0</sub> /11	q <sub>4</sub> /01

▪ Tabla de transiciones

A	B	(t)			(t+1)			E	S
		Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>		
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	1	0	1	1	0	0
0	0	1	0	0	1	0	0	0	1
0	0	1	0	1	X	X	X	X	X
0	0	1	1	0	X	X	X	X	X
0	0	1	1	1	X	X	X	X	X
0	1	0	0	0	0	0	1	0	0
0	1	0	0	1	0	1	1	0	0
0	1	0	1	0	0	0	0	0	0
0	1	0	1	1	0	0	0	1	0
0	1	1	0	0	0	1	0	0	1
0	1	1	0	1	X	X	X	X	X
0	1	1	1	0	X	X	X	X	X
0	1	1	1	1	X	X	X	X	X
1	0	0	0	0	0	1	0	0	1
1	0	0	0	1	0	0	1	0	0
1	0	0	1	0	1	0	0	0	1
1	0	0	1	1	0	0	1	0	0
1	0	1	0	0	0	0	0	1	1
1	0	1	0	1	X	X	X	X	X
1	0	1	1	0	X	X	X	X	X
1	0	1	1	1	X	X	X	X	X
1	1	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	1	0	0
1	1	0	1	0	0	1	0	0	1
1	1	0	1	1	0	1	1	0	0
1	1	1	0	0	1	0	0	0	1
1	1	1	0	1	X	X	X	X	X
1	1	1	1	0	X	X	X	X	X
1	1	1	1	1	X	X	X	X	X
m <sub>4</sub>	m <sub>3</sub>	m <sub>2</sub>	m <sub>1</sub>	m <sub>0</sub>	n <sub>4</sub>	n <sub>3</sub>	n <sub>2</sub>	n <sub>1</sub>	n <sub>0</sub>

P2. (5 puntos)

a) Teniendo en cuenta la interfaz con el 8051, diseñar la conexión de un registro de solo escritura y otro de solo lectura, ambos en la dirección 8FF0H del mapa de memoria externa. Utilizar registros LS374, decodificador LS138 y puertas lógicas. No es necesario decodificar totalmente el bus de direcciones, la zona libre de memoria de datos externa (xdata) comienza en 8000H y termina en FFFFH. Especificar razonadamente el mapa de memoria resultante tras la implementación anterior para la zona de datos externa.

## Solución:

La codificación de la dirección solicitada en binario es: 1000 1111 1111 0000, es decir, 16 bits. Si la zona libre comienza en 8000H, se puede usar toda la zona superior de 32K para el registro, con lo que la decodificación incompleta quedaría reducida al empleo de la línea A15 para el control de los registros. Cuando A15=1 y /RD=0, se activaría el registro de solo lectura. Cuando A15=1 y /WR=0, se activaría el registro de solo escritura.

El mapa de memoria resultante es que toda la zona superior de 32K estaría ocupada por ambos registros.

b) Dado el siguiente código en lenguaje C de un sistema de conversión A/D por aproximaciones sucesivas donde P3.4 es la salida del comparador que se pone a 1 cuando la medida es menor que la salida del convertidor D/A conectado al puerto P1. Recordar que el algoritmo de aproximaciones sucesivas va colocando valores en el convertidor D/A para que un comparador analógico compare este valor con la muestra a medir, de forma que se va reduciendo el rango de medida siguiendo los bits hasta que se calcula el valor medido.

```
/*
Subrutina CONV de aproximaciones sucesivas
*/
#include <reg51.h>
#include <stdio.h>
#define uchar unsigned char

sbit MENOR = P3^4; /* salida del comparador */

uchar conv(void) {
    char i;
    uchar tmp;

    P1=tmp=0;
    for (i=7;i>=0;i--) {
        tmp=tmp|(1<<i);
        P1=tmp; /* escribo el bit del D/A */
        if (MENOR) {
            tmp=tmp&(~(1<<i));
            P1=tmp;
        }
    }
    return tmp;
}

void main(void) {

    SCON = 0x52; /* Configura comunicación serie en modo 1 */
    TCON = 0x40; /* TCON */
    TMOD = 0x20; /* Timer 1 en Modo 2, GATE=0 y C/T=0 */
    TH1 = 0x0FD; /* Valor para 9600 baudios */

    /* Bucle sin fin */
    for (;;) printf ("La conversion es %u.\n", (unsigned)conv());
}
```

Analizar el código fuente y explicar el funcionamiento del sistema, dar una especificación textual empleando “situaciones tipo” del sistema y/o diagramas de flujo. ¿Para que se emplea la variable tmp?

## Solución:

Un bucle definido, donde se va testeando cada bit uno por uno hasta formar la palabra de ocho bits de salida que es precisamente el resultado de la conversión. Si se detecta que el valor de la conversión es menor, se mantiene a uno el bit correspondiente, en caso contrario se pone a cero y se intenta el siguiente de menor peso hasta terminar con el LSB (bit menos significativo). Habrá que desarrollar e indicar el diagrama de flujo.

c) Modificar la rutina conv() anterior para leer una tabla de 10 valores analógicos en memoria utilizando el mismo procedimiento de aproximaciones sucesivas. De forma que la rutina coloca en esta tabla la media de dos muestras de entrada obtenidas con sendos comparadores conectados a las patillas P3.4 (como antes) y un nuevo comparador para la otra señal conectado a la patilla P3.3. La rutina tiene el siguiente prototipo: void conv\_media(unsigned char \*tabla);

```
sbit MENOR_2=P3^3;

void conv_media(unsigned char *tabla)
{
    char i,j;
    uchar tmp,muestral,muestra2,media;

    /* Se repite 10 veces el bucle para conseguir las 10 medidas de media */
    for (j=0;j<10;j++) {
        /* ¡cuidado! El calculo de cada muestra es independiente */
        /* calculo primera muestra */
        P1=tmp=0;
        for (i=7;i>=0;i--) {
            tmp=tmp|(1<<i);
            P1=tmp; /* escribo el bit del D/A */
            if (MENOR) {
                tmp=tmp&(~(1<<i));
                P1=tmp;
            }
        }
        muestral=tmp;
        /* calculo segunda muestra */
        P1=tmp=0;
        for (i=7;i>=0;i--) {
            tmp=tmp|(1<<i);
            P1=tmp; /* escribo el bit del D/A */
            if (MENOR_2) {
                tmp=tmp&(~(1<<i));
                P1=tmp;
            }
        }
        muestra2=tmp;
        /* calculamos la media ¡cuidado! Puede superar 255 */
        media=((int)muestral+muestra2)>>1;
        *(tabla+j)=media; /* introducimos valor en la tabla */
    }
}
```