

**Problema - Septiembre :**

El siguiente algoritmo describe una determinada operación de un sistema digital.

- A) (2 puntos) Diseñar la Unidad de Procesamiento que permita realizar este algoritmo utilizando los módulos de la Figura 1: registros de desplazamiento de 8 bits, una UAL con dos entradas de 8 bits cada una, un contador módulo-8 y circuitos triestado de conexión unidireccional con control de 8 bits; además de puertas lógicas y los módulos combinacionales (MUX, DMUX, codificadores y decodificadores) que considere necesarios. Debe tener en cuenta que al bus vuelcan datos múltiples dispositivos.
- B) (2 puntos) Diseñar la Unidad de Control que ejecute este algoritmo con la Unidad de Procesamiento diseñada en el apartado A) empleando un elemento de memoria tipo D por estado. **Detalle y explique claramente** todos y cada uno de los pasos seguidos hasta obtener la solución.

```

1: Declaración: A[8], B[8], Cont[3];
2:   A ← Bus;
3:   B ← 0;
4:   for Cont = 0 to 7 do
5:     if A[0] es 1 then
6:       B ← B + 1;
7:     endif;
8:     Despl.CerradoDcha(A);
9:   endfor
10:  Bus ← B;
11:  Parar;
    
```

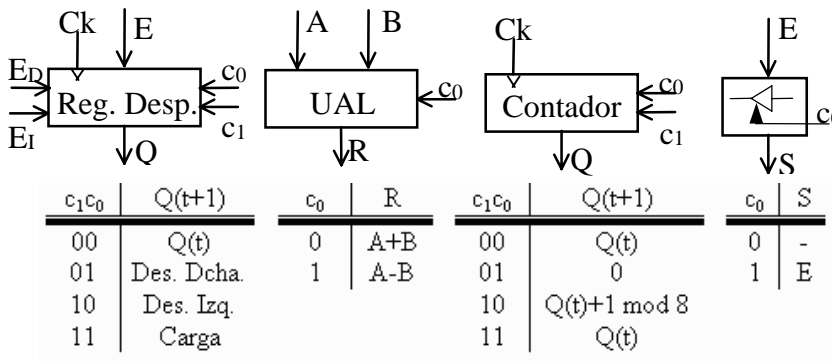


Figura 1: Módulos del problema con sus tablas de funcionamiento

**Solución**

a) En primer lugar se debe analizar las operaciones que debe realizar el algoritmo para ver qué recursos se necesitan y si éstos se encuentran entre los que se ofrecen en el enunciado:

- Un registro A que debe tener la capacidad de carga en paralelo y que sea capaz de realizar un desplazamiento **cerrado** a la derecha. En este caso se dispone de un registro de desplazamiento simple que permite realizar

operaciones de carga y desplazamiento (a la derecha y a la izquierda). Para conseguir que estas operaciones de desplazamiento sean cerradas se debe conectar el bit más o menos significativo ( $A[7]$  o  $A[0]$ ) con la entrada serie que le corresponda ( $E_D$  o  $E_I$ ). En este caso la operación es hacia la derecha con lo que en la ruta de datos se debe conectar el bit  $A[0]$  con la entrada  $E_I$  del registro A.

- Un registro **B** que debe tener la capacidad de carga en paralelo y ser capaz de inicializarse a “0”. En este caso el registro no dispone de una señal de *reset* pero se puede conseguir este efecto cargando el valor “0” (8 bits).
- Una **UAL** capaz de sumar 1 a un número de ocho bits. Como la propuesta **UAL** puede realizar las operaciones  $A+B$  y  $A-B$ , fijando una de las entradas al valor constante 1 se puede realizar la operación. Además, puesto que siempre se va a realizar la operación de suma se puede dejar el valor de la entrada de control  $c_0$  de la **UAL** siempre al valor lógico “0”.
- Un **multiplexor** de dos entradas para seleccionar el valor a cargar en el registro B, ya que se debe poder cargar desde la salida de la **UAL** o de un valor fijo igual a 0.
- Un contador con capacidad de cuenta hacia arriba para controlar las iteraciones del bucle *for* del algoritmo. Puesto que la cuenta es de 0 a 7 es necesario que dicho contador sea de tres bits.
- Un circuito de detección del estado 7 del contador que genere la señal de condición para el fin de bucle. Para ello se puede usar una puerta **AND** de tres entradas conectadas a los bits de estado del contador  $Q_2, Q_1$  y  $Q_0$ .
- Una **puerta triestado** que conecte la salida del registro **B** con el bus de datos.

Del análisis se desprende que es posible implementar la ruta de datos teniendo en cuenta que se deben generar las señales de condición adecuadas hacia la unidad de control:

- $s_0$ , señal de condición correspondiente a  $A[0] = 1$ . Se genera conectando la salida  $A[0]$  del registro **A** a la entrada de condición  $s_0$  de la unidad de control.
- $s_1$ , señal de condición de fin de bucle. Se genera conectando la salida de la puerta **AND** a la entrada de condición  $s_1$  de la unidad de control.

Por otra parte, la Unidad de Control recibe una señal externa que arranca el algoritmo:

- **I**, señal de condición de inicio de algoritmo.

En la Figura 1 se muestra el diagrama de bloques correspondiente a la unidad de procesamiento o ruta de datos con los elementos definidos anteriormente y que cumple todos los requisitos del algoritmo planteado. En la Tabla 1 se describen todas las señales de control utilizadas.

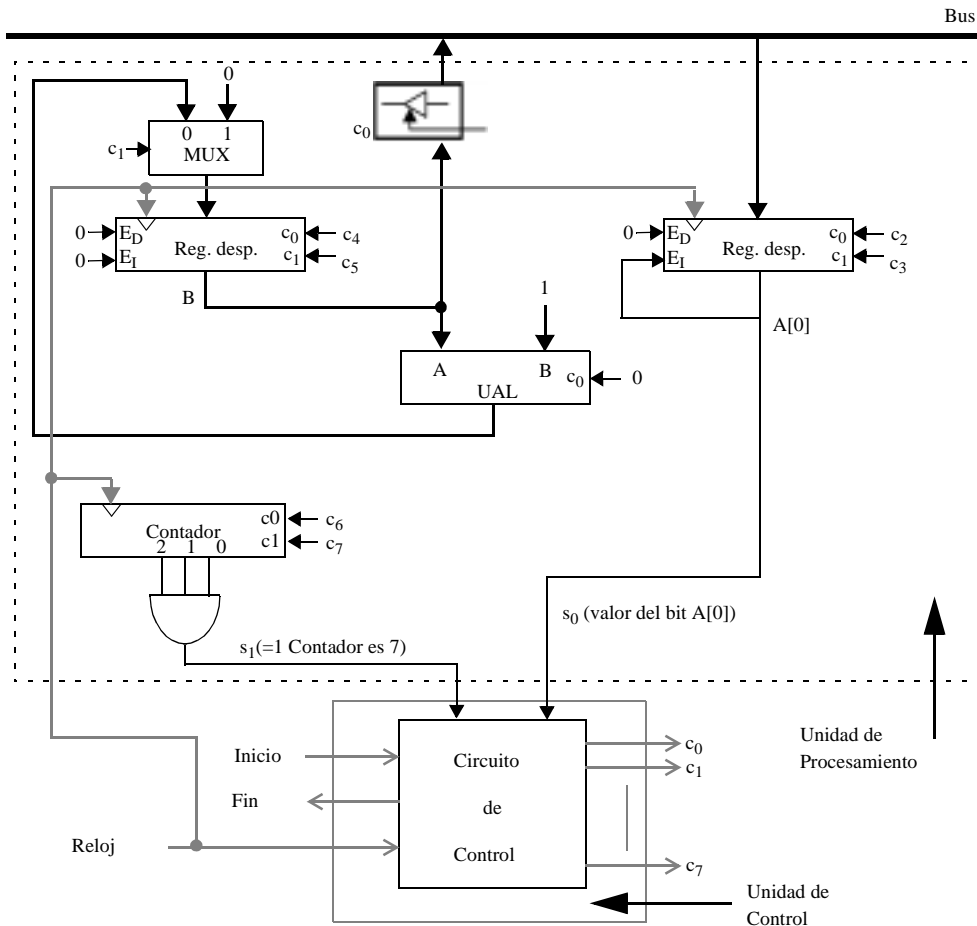


Figura 2: Diseño de la Unidad de Procesamiento

Señal de Control	Operación controlada
$c_0$	Puerta triestado que conecta la salida de B con el Bus: Nada (0) y Conexión (1)
$c_1$	Selección MUX entrada de B: UAL (0) y valor 0 (1)
$c_3$ y $c_2$	Control registro A: Nada (00), Des. Dcha. (01), Des. Izq. (10) y Carga (11)
$c_5$ y $c_4$	Control registro B: Nada (00), Des. Dcha. (01), Des. Izq. (10) y Carga (11)
$c_7$ y $c_6$	Contador: Nada (00), Reset (01), Incrementar (10) y Nada (11)

Tabla 1: Descripción de la señales de control de la Unidad de Procesamiento

b) Para describir el funcionamiento de la Unidad de Control solicitada en este apartado, se diseña el diagrama de transición de estados de la Figura 3, donde el significado detallado de cada uno de los estados propuestos viene dado por la Tabla 2.

Este diagrama de transición de estados cumple todos los requisitos para ejecutar el algoritmo propuesto en el enunciado del problema utilizando la Unidad de Procesamiento diseñada en el apartado anterior. Se puede comprobar, por ejemplo, que ejecuta el bucle *for* 8 veces ya que en el estado  $S_4$  consulta por la condición de finalización y, en caso de no verificarse, incrementa el contador y realiza otra iteración. En total el bucle se realiza para todos los valores del contador, desde el 0 hasta el 7.

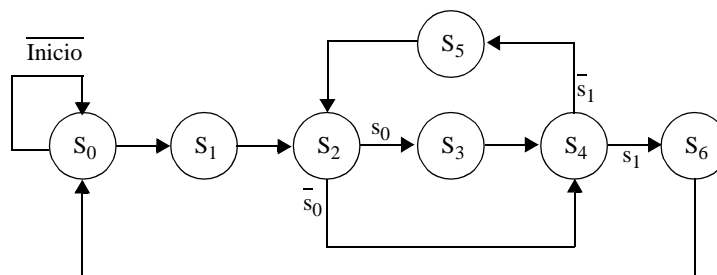
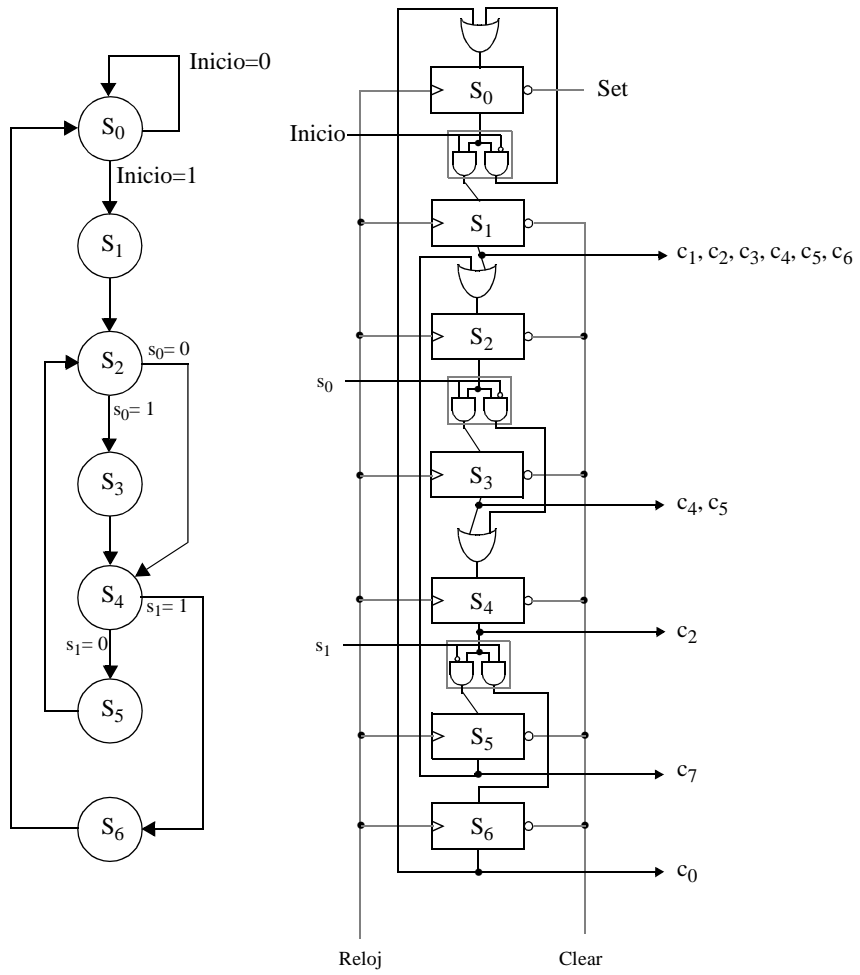


Figura 3: Diagrama de estados de la Unidad de Control

Estado de la Unidad de Control	Microoperaciones efectuadas	Señales de control a activar
$S_0$	Ninguna	Ninguna
$S_1$	$A \leftarrow \text{Bus}$ $B \leftarrow 0$ Reset Cont	$c_3, c_2$ $c_1, c_5, c_4$ $(c_7=0), c_6$
$S_2$	Ninguna	Ninguna
$S_3$	$B \leftarrow B + 1$	$(c_1=0), c_5, c_4$
$S_4$	Despl.CerradoDcha(A)	$(c_3=0), c_2$
$S_5$	Incrementar Cont	$c_7, (c_6=0)$
$S_6$	$\text{Bus} \leftarrow B$	$c_0$

Tabla 2: Acciones tomadas por la Unidad de Control en cada estado

Para implementar esta Unidad de Control vamos a emplear el método de los elementos de retardo. El diseño correspondiente se muestra en la Figura 4. En la parte izquierda de la Figura 4 se ha vuelto a colocar el diagrama de transición de estados de forma que sirva de ayuda y complemento al diseño lógico de la Unidad de Control.



**Figura 4:** Unidad de Control