

```

4:   C ← Bus
5:   for Contador = 0 to 7 do
6:       begin
7:           if C(0) ≠ C(1) then A ← A / B
8:           else A ← A + B endif
9:           Desplazar derecha (C)
10:        end
11:   Parar;

```

Utilizando los módulos dibujados abajo: registros de desplazamiento de 8 bits, una UAL con dos entradas de 8 bits cada una y un contador módulo-8; además de biestables tipo D, puertas lógicas y los módulos combinacionales que considere necesarios:

- A) Diseñar la Unidad de Procesamiento que permita realizar este algoritmo.
- B) Diseñar la Unidad de Control, utilizando la técnica de elementos de retardo, que ejecute este algoritmo con la Unidad de Procesamiento diseñada en el apartado A).

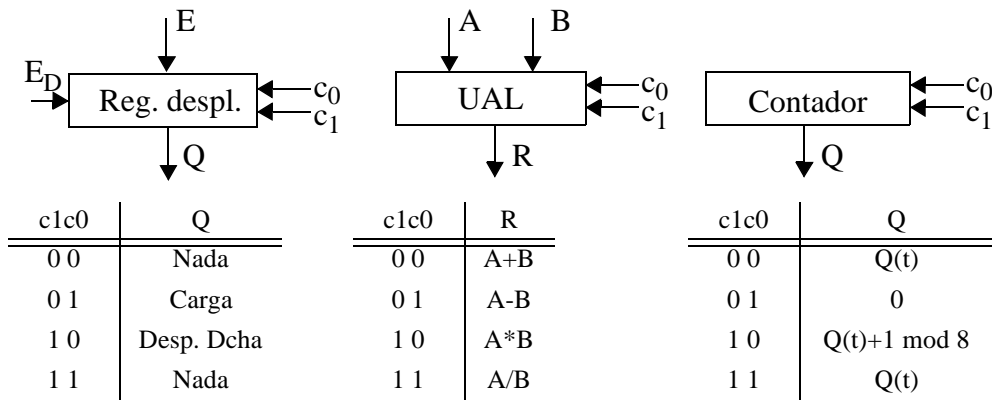


Figura 96-3-1 Módulos del problema y debajo su tabla de funcionamiento

### Solución

A) Antes de proponer un diseño para la Unidad de Procesamiento hagamos unas consideraciones previas sobre los requerimientos del algoritmo. La Unidad de Procesamiento tiene que resolver cada uno de estos

---

requerimientos.

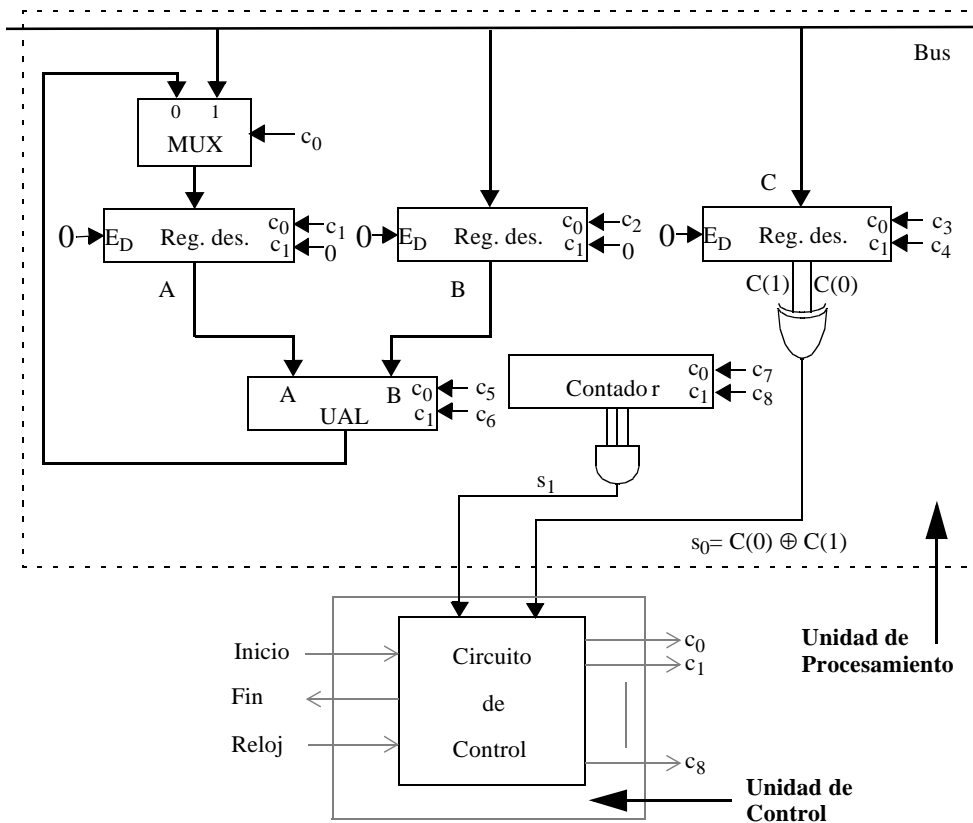
- El registro A tiene que poder recibir información desde el bus y desde la salida de la UAL. Al llegarle información desde dos lugares distintos es necesario colocar un multiplexor en la entrada para seleccionar cuál de los dos debe ser cargado en el registro. También el registro A debe entregar su contenido a la UAL.
- Al registro B le llega información del Bus y entrega su contenido a la UAL.
- El registro C recibe información desde el Bus y el algoritmo necesita consultar el valor de sus dos bits menos significativos para comprobar si son iguales o distintos. Esta condición la generaremos mediante la función lógica  $C(0) \oplus C(1)$ .
- En el algoritmo hay un bucle que se ejecutará un total de 8 veces. Para controlar esto se utilizamos el contador módulo 8 y para detectar el final de la cuenta hacemos el producto lógico (AND) de sus tres salidas.
- Los registros de desplazamiento del enunciado son los únicos elementos de almacenamiento disponibles, por lo que se utilizarán para los tres registros: A, B y C. En el caso de A y B no se hará uso de la capacidad de estos registros para realizar desplazamientos. A la hora de hacer desplazamientos en el registro C hay que introducir un 0, tal y como exige el enunciado del problema, por lo que habrá que fijar su entrada ED a 0. Con los registros A y B no existe este problema, no se van a hacer desplazamientos con ellos, y su entrada ED se fija arbitrariamente a un valor determinado, por ejemplo 0.

Un posible diseño para la Unidad de Procesamiento pedida, teniendo en cuenta todas estas consideraciones iniciales, puede ser el mostrado en la Figura 96-3-2.

Sobre este diseño conviene hacer una observación. Los registros A y B sólo van a utilizar dos valores de sus entradas de control

$c_1c_0$	Q
0 0	Nada
0 1	Carga

por lo que se puede fijar una de ellas, la más significativa  $c_1$ , directamente a 0.



**Figura 96-3-2** Diagrama de bloques, puntos de control ( $c_0$  a  $c_9$ ) y señales de condición ( $s_0$  y  $s_1$ ) de la Unidad de Procesamiento solicitada

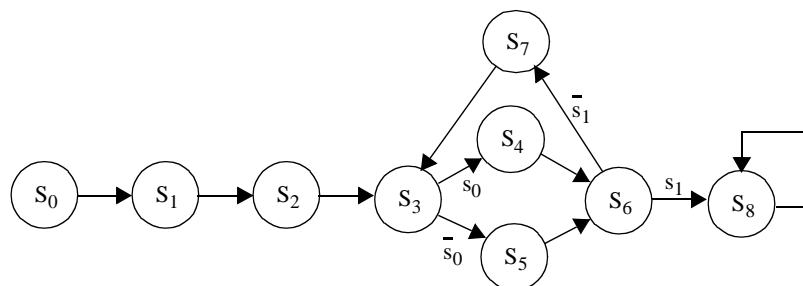
Para completar este diseño, es necesario hacer una breve descripción del significado de cada una de las entradas de control que gobiernan su funcionamiento (ver Tabla 96-3-1).

Señal de Control	Operación controlada
$c_0$	Entrada de selección del multiplexor
$c_1$	Cargar A desde la salida del multiplexor
$c_2$	Cargar B desde el Bus
$c_4$ y $c_3$	Cargar C desde el Bus (01), desplazamiento derecha de C (10) o nada (00)
$c_6$ y $c_5$	Control de la UAL: suma (00), resta (01), producto (10) o división (11)
$c_8$ y $c_7$	Control del contador: nada (00 y 11), reset (01) o incremento (10)

**Tabla 96-3-1** Descripción de la señales de control de la Unidad de Procesamiento

B) Un posible diagrama de estados para la Unidad de Control que gobierne el funcionamiento de la Unidad de Procesamiento de la Figura 96-3-2, es el mostrado en la Figura 96-3-3. La Tabla 96-3-2 da una descripción detallada de cada uno de los estados junto con las señales de control a activar en cada uno de ellos.

Una de las dificultades que tiene este algoritmo es que el bucle debe ejecutarse necesariamente ocho veces. Para conseguirlo se ha optado por comprobar la condición generada por el contador en el estado S6 y después, si no se hubiera llegado al final de la cuenta ( $s_1=1$ ), incrementar el contador.



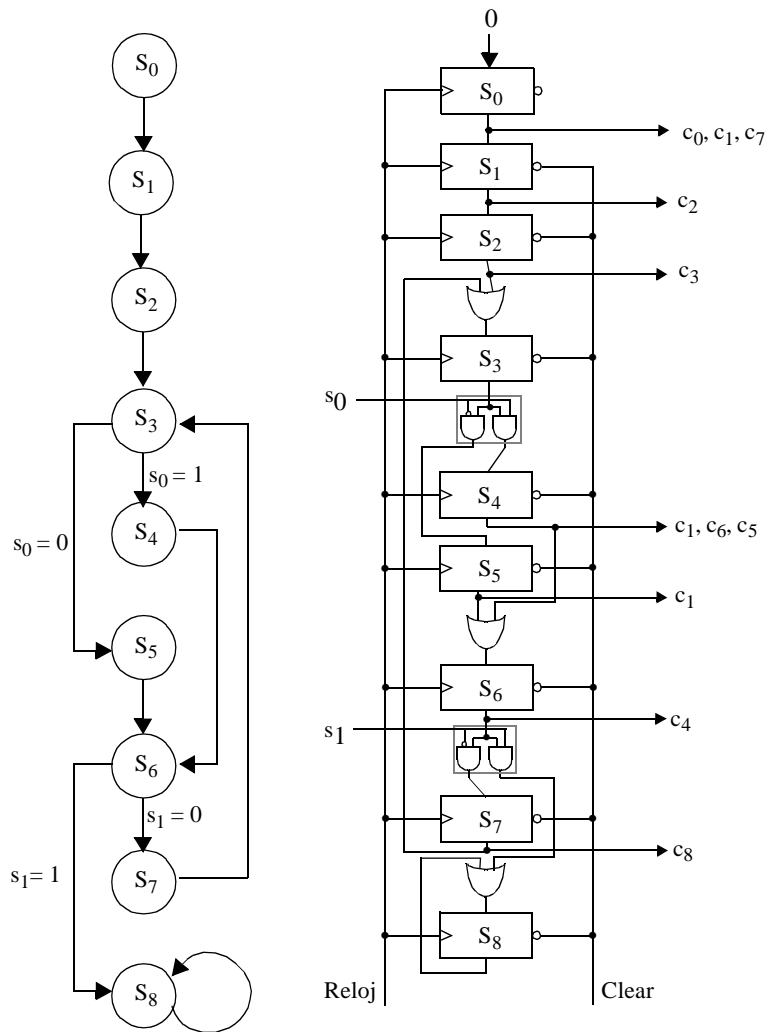
**Figura 96-3-3** Diagrama de estados de la Unidad de Control

---

Estado de la Unidad de Control	Microoperaciones efectuadas	Señales de control a activar
S <sub>0</sub>	A ← Bus contador ← 0	c <sub>0</sub> , c <sub>1</sub> , c <sub>7</sub>
S <sub>1</sub>	B ← Bus	c <sub>2</sub>
S <sub>2</sub>	C ← Bus	c <sub>3</sub>
S <sub>3</sub>	Ninguna	Ninguna
S <sub>4</sub>	A ← A / B	c <sub>1</sub> , c <sub>6</sub> , c <sub>7</sub>
S <sub>5</sub>	A ← A + B	c <sub>1</sub>
S <sub>6</sub>	Desplazar derecha C	c <sub>4</sub>
S <sub>7</sub>	Incrementar contador	c <sub>8</sub>
S <sub>8</sub>	Ninguna	Ninguna

**Tabla 96-3-2** Acciones tomadas por la Unidad de Control en cada estado

El método de diseño utilizando elementos de retardo es prácticamente una transcripción directa del diagrama de estados. En la Figura 96-3-4 se muestra la Unidad de Control diseñada utilizando este método, junto con el diagrama de estados de la Figura 96-3-3, redibujado en vertical, para servir de apoyo en el diseño.



**Figura 96-3-4** Unidad de Control