

```

7:          if C(3) = C(5) then A ← A * B
8:          else A ← A - B endif;
9:          Desplazar derecha (C);
10:         end;
11:         Bus ← A;
12:         Parar;

```

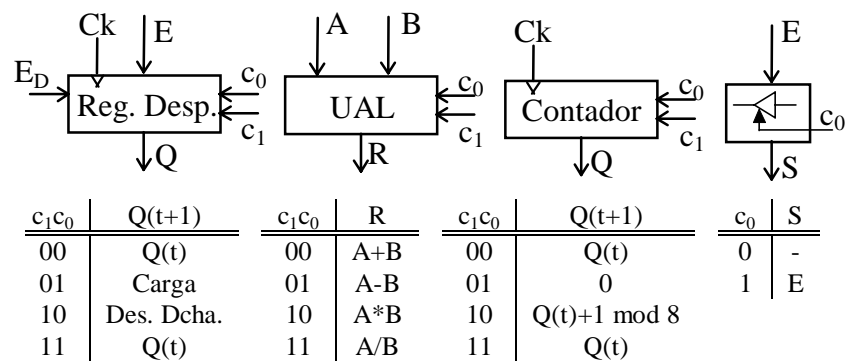


Figura 97-2-1 Módulos del problema y debajo su tabla de funcionamiento

Solución

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

- Los registros de desplazamiento del enunciado son los únicos elementos de almacenamiento disponibles, por lo que se utilizarán necesariamente 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, pero sí en C (paso 9 del algoritmo).
- A la hora de hacer desplazamientos en el registro C hay que introducir un 0, tal y como exige el apartado A) del problema, por lo que habrá que fijar su entrada ED a 0. Con los registros A y B no existe este problema ya que no se van a hacer desplazamientos con ellos, y su entrada ED se fija arbitrariamente a un valor determinado, por ejemplo 0.
- El registro A tiene que poder recibir información desde el bus (paso 2 del algoritmo) y desde la salida de la UAL (pasos 7 y 8 del algoritmo). Al

llevarle 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.

- El registro A debe entregar su contenido a la UAL y al bus. Su salida tiene que estar conectada con ambos, pero para **la conexión al bus es necesario colocar un circuito triestado** para evitar que este registro envíe continuamente su salida al bus. Si no se colocara este circuito triestado se producirían continuos conflictos eléctricos en el bus que lo harían totalmente inservible (ver Capítulo 1 del libro de teoría).
- 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 se genera directamente mediante la función lógica $C(3) \oplus C(5)$.
- En el algoritmo hay un bucle que se ejecutará un total de 4 veces, desde 0 hasta 3. Para controlar esto se utilizamos el contador módulo 8. La forma de detectar el final de la cuenta depende del diagrama de estados que se emplee. En esta solución se propone incrementar el contador nada más entrar en el bucle y comprobar la condición de finalización en el mismo estado en el que se desplaza el registro C. Si se hace así, la Unidad de Control tiene que comprobar cuándo el contador ha llegado a 4, es decir, simplemente tiene que saber si el bit más significativo del contador es 1 o es 0. Si es 1 el bucle ha llegado al final y se salta al estado S6 en el diagrama de estados de la Figura 97-2-3.
- Los registros de desplazamiento de 8 bits y el contador módulo-8 tienen una entrada de reloj Ck. En el diseño de la Unidad de Procesamiento habrá que conectar todas las entradas de reloj con la que llega a la Unidad de Control para conseguir una sincronización correcta en todas las acciones de ambas.

Un posible diseño para la Unidad de Procesamiento pedida, teniendo en cuenta todos estos requisitos, puede ser el mostrado en la Figura 97-2-2.

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	$Q(t+1)$
0 0	$Q(t)$
0 1	Carga

por lo que se puede fijar su entrada de control c_1 directamente a 0.

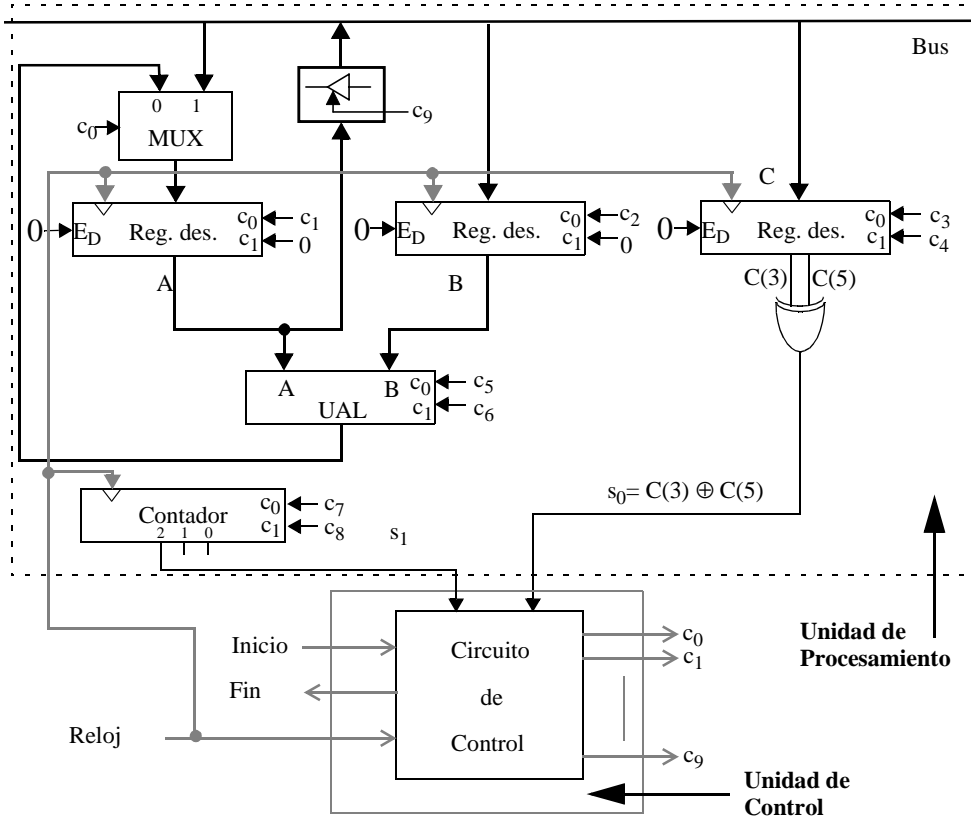


Figura 97-2-2 Diagrama de bloques, puntos de control (c_0 a c_9) y señales de condición (s_0 y s_1) de la Unidad de Procesamiento solicitada

Para completar este diseño, es necesario hacer una breve descripción del significado de cada una de las entradas de control que gobiernan el funcionamiento de la Unidad de Procesamiento (ver Tabla 97-2-1).

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)
c_9	Control del circuito triestado: Bus \leftarrow A (1), A aislado del bus (0)

Tabla 97-2-1 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 97-2-2, es el mostrado en la Figura 97-2-3. La Tabla 97-2-2 da una descripción detallada de cada uno de los estados junto con las señales de control a activar en cada uno de ellos, y la codificación de los estados empleada es la mostrada en la Tabla 97-2-3.

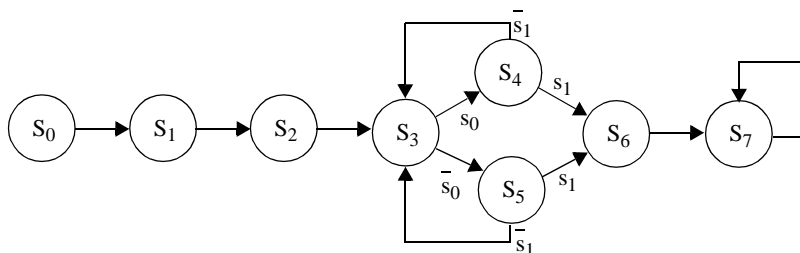


Figura 97-2-3 Diagrama de estados de la Unidad de Control

Estado de la Unidad de Control	Microoperaciones efectuadas	Señales de control a activar
S ₀	A ← Bus contador ← 0	c ₀ , c ₁ , c ₇
S ₁	B ← Bus	c ₂
S ₂	C ← Bus	c ₃
S ₃	Ninguna	Ninguna
S ₄	A ← A - B Desplazar derecha C Incrementar contador	c ₁ , c ₅ , c ₄ , c ₈
S ₅	A ← A * B Desplazar derecha C Incrementar contador	c ₁ , c ₆ , c ₄ , c ₈
S ₆	Bus ← A	c ₉
S ₇	Ninguna	Ninguna

Tabla 97-2-2 Acciones tomadas por la Unidad de Control en cada estado

A la hora de diseñar la Unidad de Control utilizando memoria ROM el enunciado indica que se valorará la utilización de una memoria de tamaño mínimo. Para conseguir este objetivo hay que darse cuenta que en cada estado se consulta, como máximo, una condición, por lo que sería posible reducir el número de palabras de la ROM a la mitad si se selecciona previamente la condición utilizando un multiplexor. Las entradas de selección de este multiplexor pueden ser el mismo estado (selección por estado) o un bit adicional almacenado en la memoria ROM (selección por campo). Como esta segunda solución (selección por campo) requiere de una memoria ROM de tamaño mayor al tener que almacenar ese bit adicional por estado, se opta por la primera (selección por estado) aunque requiere del multiplexor mencionado.

Estado	Q ₂ Q ₁ Q ₀
S0	0 0 0
S1	0 0 1
S2	0 1 0
S3	0 1 1
S4	1 0 0
S5	1 0 1
S6	1 1 0
S7	1 1 1

Tabla 97-2-3 Codificación de los estados

La Figura 97-2-4 muestra la estructura de la Unidad de Control pedida en el enunciado del problema. Tiene 4 líneas de direcciones (1 para la condición y las otras 3 para el valor del estado actual) y una longitud de palabra de 13 bits (3 para el estado siguiente y 10 para las señales de control), por tanto tiene un tamaño de 2^4 palabras x 13 bits por palabra.

NOTA

Un error muy frecuente a la hora de resolver el problema es la utilización de puntos de control en vez de las entradas de control de los módulos empleados (registros de desplazamiento, contadores, etc). Tal y como se indica en las páginas 321 a 323 del libro de teoría, “*Los puntos de control se realizan mediante circuitos cuya naturaleza depende de las características de los dispositivos conectados a los caminos de datos sobre los que actúa* ”.

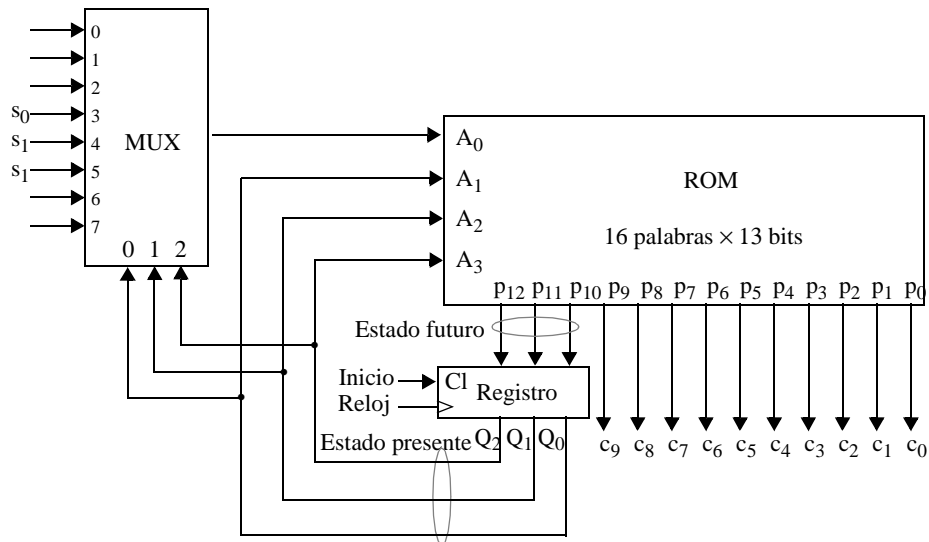


Figura 97-2-4 Unidad de Control diseñada con una memoria ROM

Estado Presente	Dirección de la ROM				Contenido de la ROM													
	A ₃	A ₂	A ₁	A ₀	P ₁₂	P ₁₁	P ₁₀	P ₉	P ₈	P ₇	P ₆	P ₅	P ₄	P ₃	P ₂	P ₁	P ₀	
	Estado presente + Cond. Q ₂ Q ₁ Q ₀ s _i				Próximo Estado Q ₂ Q ₁ Q ₀			Control c ₉ c ₈ c ₇ c ₆ c ₅ c ₄ c ₃ c ₂ c ₁ c ₀										
S ₀	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	1
	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	1	1
S ₁	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0
S ₂	0	1	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0
	0	1	0	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0
S ₃	0	1	1	0 (si s ₀ = 0)	1	0	1	0	0	0	0	0	0	0	0	0	0	0
	0	1	1	1 (si s ₀ = 1)	1	0	0	0	0	0	0	0	0	0	0	0	0	0
S ₄	1	0	0	0 (si s ₁ = 0)	0	1	1	0	1	0	0	1	1	0	0	1	0	0
	1	0	0	1 (si s ₁ = 1)	1	1	0	0	1	0	0	1	1	0	0	1	0	0
S ₅	1	0	1	0 (si s ₁ = 0)	0	1	1	0	1	0	1	0	1	0	0	1	0	0
	1	0	1	1 (si s ₁ = 1)	1	1	0	0	1	0	1	0	1	0	0	1	0	0
S ₆	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
S ₇	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Tabla 97-3-4 Contenido de la memoria ROM