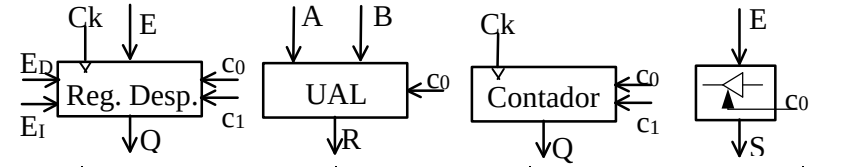
dispositivos y evitar en su diseño posibles conflictos eléctricos entre ellos.

B) (2 puntos) Diseñar la Unidad de Control que ejecute este algoritmo con la Unidad de Procesamiento diseñada en el apartado A) empleando la técnica de los elementos de retardo. **Detalle y explique claramente** todos y cada uno de los pasos seguidos hasta obtener la solución.

```
1: Declaración: A[8], B[8], Cont[4]; Bus[8]
2:      A ← Bus;
3:      B ← Bus, Cont = 0;
4:      while Cont ≠ 14
5:           if A es multiplo de 4 then
6:                A ← A - B, Cont = (Cont +2) mod 16;
7:           else
8:                B ← B + A;
9:           endif;
10:     endwhile;
11:     Bus ← B;
12:     Bus ← A;
13:     Parar;
```



c_1c_0	Q(t+1)
00	Q(t)
01	D. Dcha.
10	D. Izq.
11	Carga

c_0	R
0	A+B
1	A-B

c_1c_0	Q(t+1)
00	Q(t)
01	Q(t)-1 mod 16
10	Q(t)+1 mod 16
11	0

c_0	S
0	-
1	E

Figura 2: Módulos secuenciales del problema con sus tablas de funcionamiento

Solución

a) En primer lugar se deben analizar las operaciones a realizar para ver qué recursos se necesitan y si éstos se encuentran disponibles:

- Sendos registros **A** y **B** de 8 bits que deben poder cargarse desde el bus y volcar su contenido al bus. Así mismo, deben poder cargarse desde la salida de la UAL. Como registros nos ofrece el enunciado registros de desplazamiento de 8 bits con capacidad de carga en paralelo. Éstos son los registros que habrá que utilizar, aunque su capacidad de desplazamiento no se utilizará.
- Para seleccionar el origen de datos de los registros se utilizarán multiplexores.

- Los registros volcarán su contenido al bus a través de sendas **puertas triestado**, para evitar problemas eléctricos.
- Un **contador** módulo-16 (de 4 bits de longitud de palabra por tanto). Aunque en el algoritmo propuesto la cuenta se ha de incrementar en 2 en cada ocasión y el contador de que se dispone sólo puede contar de 1 en 1, esto no supone nungún problema ya que bastará con incrementar 1 dos veces seguidas.

Del análisis del algoritmo se sigue que harán falta dos señales de condición:

- s_0, que indique si el valor del registro A es múltiplo de 4. Para ello veamos cómo son los números múltiplos de 4:

	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
4:	0	0	0	0	0	1	0	0
8:	0	0	0	0	1	0	0	0
12:	0	0	0	0	1	1	0	0
16:	0	0	0	1	0	0	0	0
20:	0	0	0	1	0	1	0	0

Tabla 1: Los múltiplos de 4

Es decir, son todos ellos números acabados en 00. Si se admite el 0 como múltiplo de 4 (este problema se resolverá aquí adoptando este criterio) entonces se tiene que $s_0 = \overline{A}_1\,\overline{A}_0$. Si se considera que 0 no es múltiplo de 4 entonces alguno de los bits más significativos ha de ser distinto de 0: $s_0 = (A_7 + A_6 + A_5 + A_4 + A_3 + A_2)\,\overline{A}_1\,\overline{A}_0$.

Otra manera de ver cómo son los números múltiplos de 4 es la siguiente: Un número binario $a_n a_{n-1} a_{n-2} \ldots a_1 a_0$ tiene el valor decimal $a_n \times 2^n + a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \ldots + a_1 \times 2^1 + a_0 \times 2^0$. Para dividir este número entre 4 lo multiplicamos por 2^{-2}, con lo que resulta: $a_n \times 2^{n-2} + a_{n-1} \times 2^{n-3} + a_{n-2} \times 2^{n-4} + \ldots + a_2 \times 2^0 + a_1 \times 2^{-1} + a_0 \times 2^{-2}$. Si el número inicial es divisible entre 4, la parte decimal del cociente de la división debe ser 0. En este caso, la parte fraccionaria es $a_1 \times 2^{-1} + a_0 \times 2^{-2}$, pues $2^{-1} = 0.5$ y $2^{-2} = 0.25$. Ya que a_1 y a_0 sólo pueden tomar los valores 0 ó 1 (≥ 0), la única forma de que la parte fraccionaria sea 0 es que $a_1 = a_0 = 0$.

- s_1, que indica si el contador vale 14. Con cuatro dígitos binarios, 14 se expresa como 1 1 1 0, luego esta señal de condición se realizará fácilmente con una puerta AND y un inversor.

Un posible diseño para la Unidad de Procesamiento, que cumple con todos estos requisitos, es el que se muestra en la Figura 3.

Ésta no es la única Unidad de Procesamiento capaz de realizar el algoritmo pedido. Por ejemplo, dado que los registros A y B no se van a cargar simultáneamente, es posible hacer uso de un único multiplexor. Así mismo, dado

que la capacidad de desplazamiento de los registros no se va a utilizar, y ya que 11 implica carga en paralelo y 00 implica no-operación, es posible gobernar cada uno de los registros con una única señal de control conectada simultáneamente a sus entradas c_0 y c_1. La Figura 4 muestra cómo se realizarían estas modificaciones. (Este diseño, al hacer uso de menos componentes y menos señales de control, es más económico.) En lo que sigue, el problema se resolverá haciendo uso del diseño de la Figura 3.

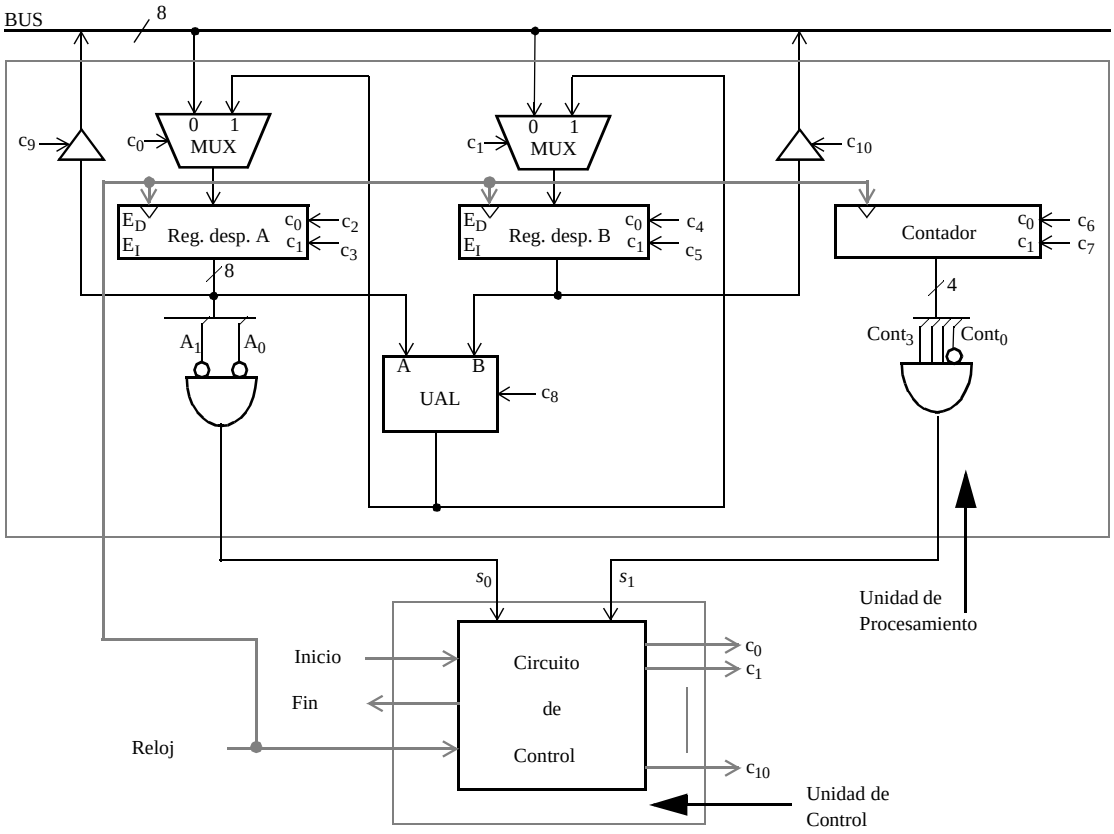


Figura 3: Diseño de la Unidad de Procesamiento

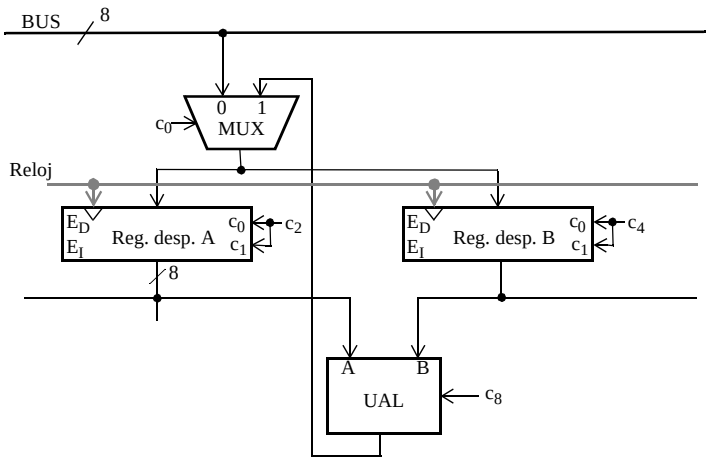


Figura 4: Fragmento de la Unidad de Procesamiento, en el que se muestran algunas posibles modificaciones en el diseño

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 5, donde el significado detallado de cada uno de los estados propuestos viene dado por la Tabla 2. Obsérvese que esta Unidad de Control se ha diseñado como una máquina de Moore. (Dada la sencillez del algoritmo propuesto, se ha obtenido directamente el diagrama de estados a partir de éste, sin necesidad de pasar por el diagrama ASM.)

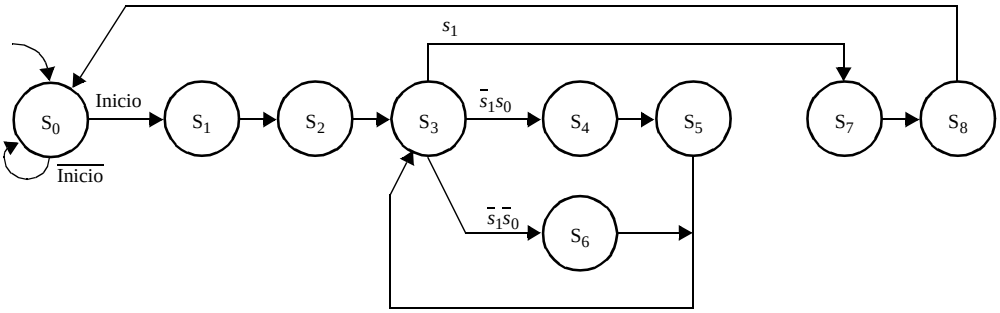


Figura 5: Diagrama de estados de la Unidad de Control

Estado de la Unidad de Control	Microoperaciones efectuadas	Señales de control a activar
S_0	Ninguna	Ninguna
S_1	A ← Bus	(c_0=0), c_2, c_3
S_2	B ← Bus Cont ← 0	(c_1=0), c_4, c_5 c_6, c_7
S_3	Ninguna	Ninguna
S_4	A ← A - B Cont ← (Cont + 1) mod 16	c_0, c_2, c_3, c_8 (c_6=0), c_7
S_5	Cont ← (Cont + 1) mod 16	(c_6=0), c_7
S_6	B ← B + A	c_1, c_4, c_5, (c_8=0)
S_7	Bus ← B	c_{10}
S_8	Bus ← A	c_9

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

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 *while* mientras el valor del contador sea distinto de 14, ya que de verificarse esta condición (s_1 = 1) del estado S_3 se salta al estado S_7. En caso contrario, en S_3 se comprueba si A es múltiplo de 4 (s_0) o no ($\overline{s_0}$) y se salta al estado correspondiente. Obsérvese que, como se comentó en el apartado anterior, el contador no se puede incrementar en 2 en un único estado, por lo que ha habido que incrementarlo en una unidad dos veces consecutivas, en los estados S_4 y S_5.

La Unidad de Control se realiza utilizando la técnica de los *elementos de retardo*, según pide el enunciado, asignando un biestable tipo D a cada estado, tal como se muestra en la Figura 6. Para evaluar las condiciones en S_3, en lugar de utilizar un demultiplexor de dos entradas de control (s_1 y s_0) y cuatro salidas se ha preferido utilizar, de manera equivalente, dos multiplexores colocados en cascada, cada uno de ellos con una única entrada de control y dos salidas. También se muestra cómo se forman las distintas señales de control a partir de las salidas de los biestables tipo D [ver las páginas 304 a 307 del texto base de teoría]. Obsérvese que los bloques de decisión corresponden a demultiplexores, según se muestra en la Figura 7 [ver las páginas 304 a 307 del texto base de teoría].

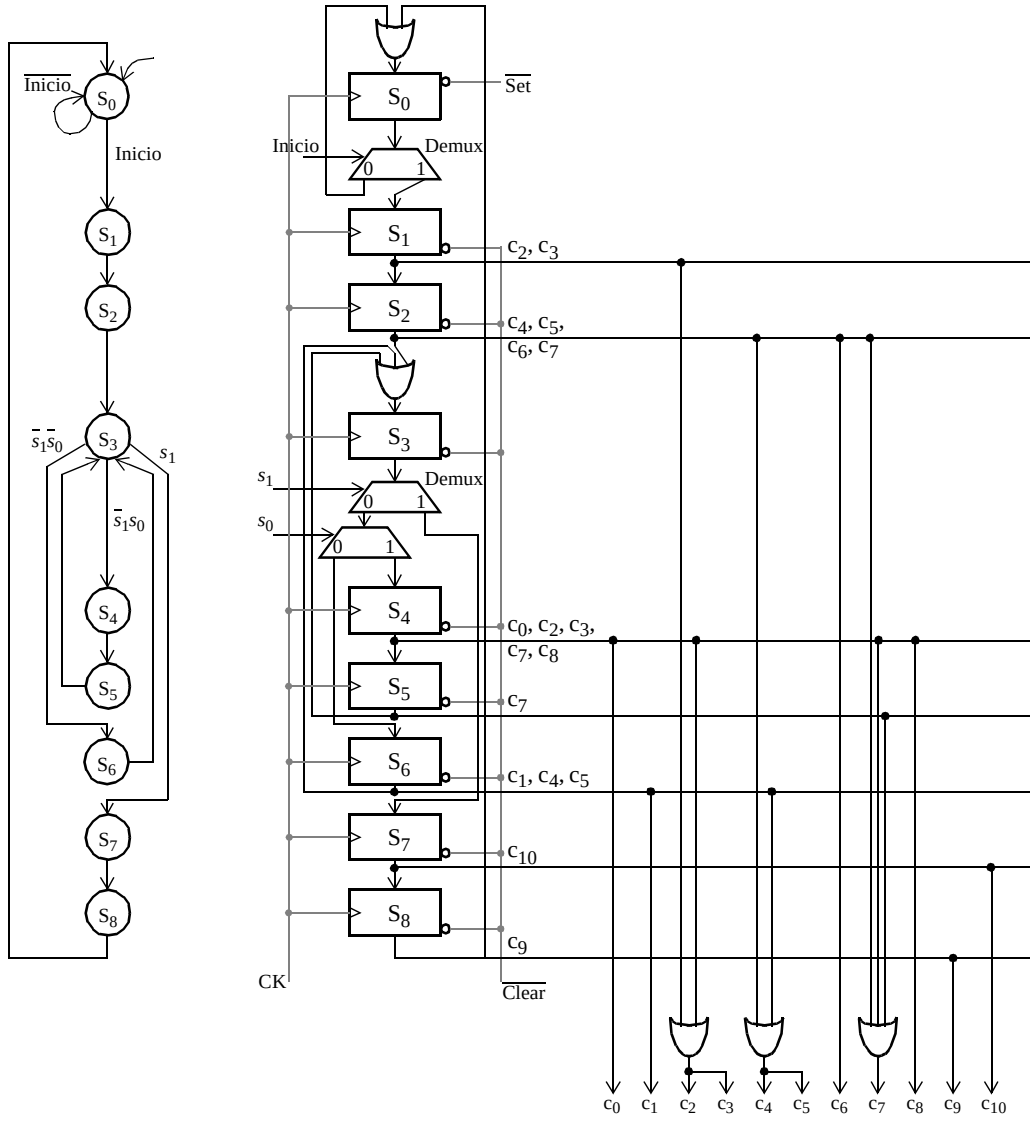


Figura 6: Unidad de Control mediante *elementos de retardo* (derecha); se muestra nuevamente el diagrama de estados (izquierda) para ilustrar el alto grado de paralelismo que existe entre éste y la realización mediante elementos de retardo

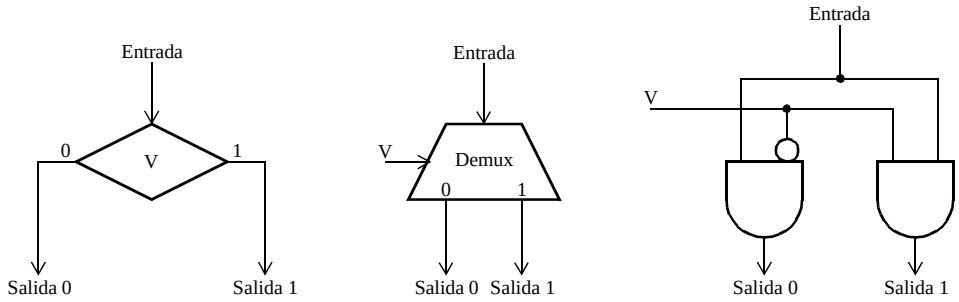


Figura 7: Bloque de decisión (izquierda) visto como un demultiplexor (centro) y su realización equivalente con puertas lógicas (derecha)