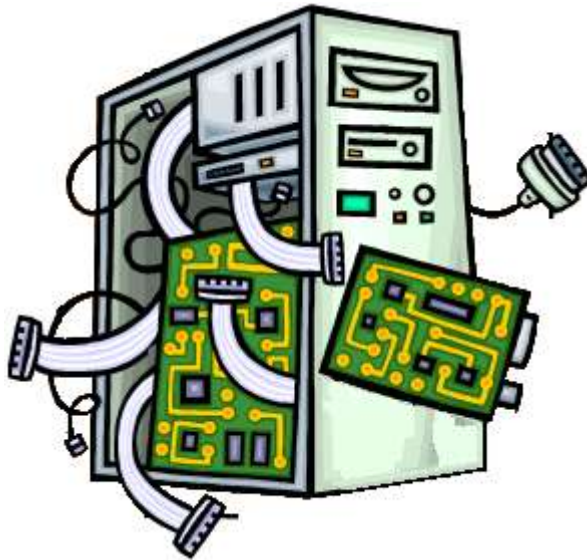
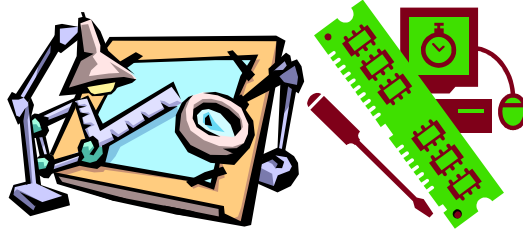
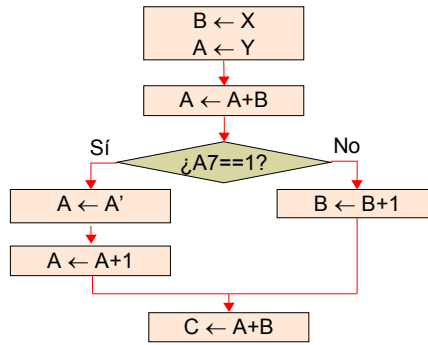


Ejercicios  
de  
Arquitectura de Computadoras



José Garzía

En la figura se representa el diagrama de flujo de un algoritmo.



Los registros  $A$ ,  $B$  y  $C$  tienen una longitud de 8 bits.  $A_7$  representa el bit más significativo del registro  $A$ .  $A'$  es el complemento a 1 de  $A$ . En las operaciones de suma se ignoran los arrastres del bit más significativo. Diseñe la arquitectura de un sistema digital (Sección de Procesamiento + Sección de Control) que realice este algoritmo. Se supone que la sección de procesamiento dispone de los recursos de cálculo necesarios en el algoritmo. Para la sección de control especifique únicamente su diagrama de estados y las acciones que tiene que efectuar cuando se encuentra en cada uno de los mismos (no es necesario diseñar el circuito secuencial correspondiente a la sección de control).

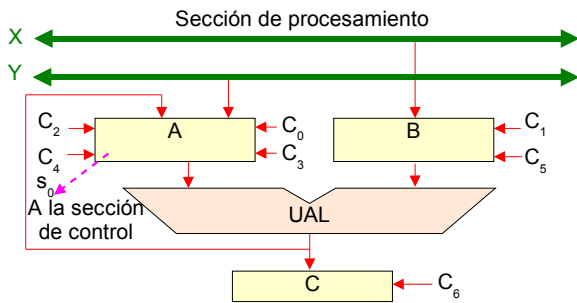
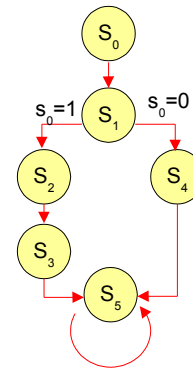


Diagrama de estados de la sección de control



Señal de control	Operación controlada
$C_0$	$A \leftarrow Y$
$C_1$	$B \leftarrow X$
$C_2$	$A \leftarrow A'$
$C_3$	$A \leftarrow A+1$
$C_4$	$A \leftarrow (UAL)$
$C_5$	$B \leftarrow B+1$
$C_6$	$C \leftarrow (UAL)$

Estado de la sección de control	Microoperaciones efectuadas	Señales de control activas
$S_0$	$B \leftarrow X$ $A \leftarrow Y$	$C_0, C_1$
$S_1$	$A \leftarrow A+B$	$C_4$
$S_2$	$A \leftarrow A'$	$C_2$
$S_3$	$A \leftarrow A+1$	$C_3$
$S_4$	$B \leftarrow B+1$	$C_5$
$S_5$	$C \leftarrow (UAL)$	$C_6$

Sea el siguiente algoritmo:

```

1: Declaración de registros : A[5], B[3], R[5];
2: Inicio: A ← 00001, B ← 100; (números binarios)
3: Bucle: A ← A * B;
4: B ← B - 1;
5: if B <> 1 then go to Bucle;
6: R ← A;
7: Parar;

```

- a) ¿Cuál es el valor de R al finalizar? ¿Qué operación realiza este algoritmo?.
- b) Diseñe la arquitectura de un sistema digital (**Sección de Procesamiento + Sección de Control**) que realice este algoritmo. Se supone que la sección de procesamiento dispone de un recurso de cálculo que efectúa la multiplicación. Para la sección de control especifique sólo su diagrama de estados y las acciones que tiene que efectuar cuando se encuentra en cada uno de los mismos (no es necesario diseñar el circuito secuencial correspondiente a la sección de control).

a) Este algoritmo calcula el factorial del número inicialmente almacenado en el registro B. Mientras este contenido va decrementándose en una unidad cada pasada por el bucle, los productos parciales se van acumulando en el registro A. Finalmente, el resultado se almacena en el registro R.

b) Diagrama de bloques de la sección de procesamiento

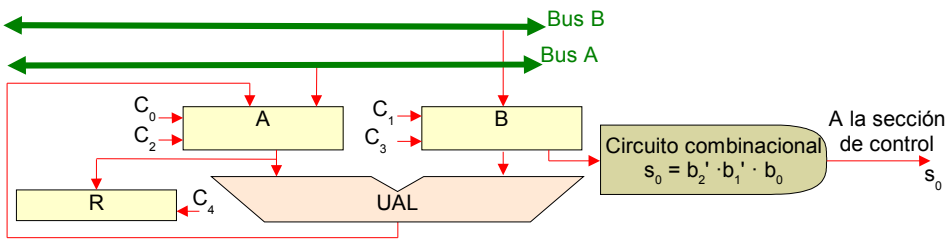
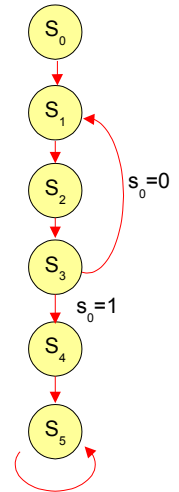


Diagrama de estados de la sección de control



Señal de control	Operación controlada
C <sub>0</sub>	A ← Bus A
C <sub>1</sub>	B ← Bus B
C <sub>2</sub>	A ← (UAL)
C <sub>3</sub>	B ← B-1
C <sub>4</sub>	R ← A

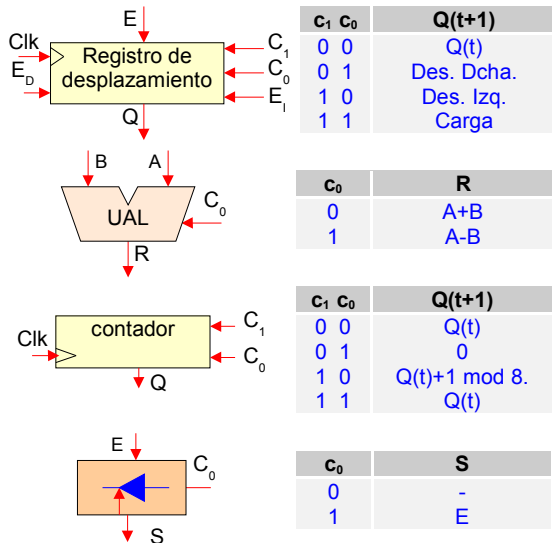
Estado de la sección de control	Microoperaciones efectuadas	Señales de control activas
S <sub>0</sub>	A ← Bus A B ← Bus B	C <sub>0</sub> , C <sub>1</sub>
S <sub>1</sub>	A ← A * B	C <sub>2</sub>
S <sub>2</sub>	B ← B - 1	C <sub>3</sub>
S <sub>3</sub>	ninguna	ninguna
S <sub>4</sub>	R ← A	C <sub>4</sub>
S <sub>5</sub>	ninguna	ninguna

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

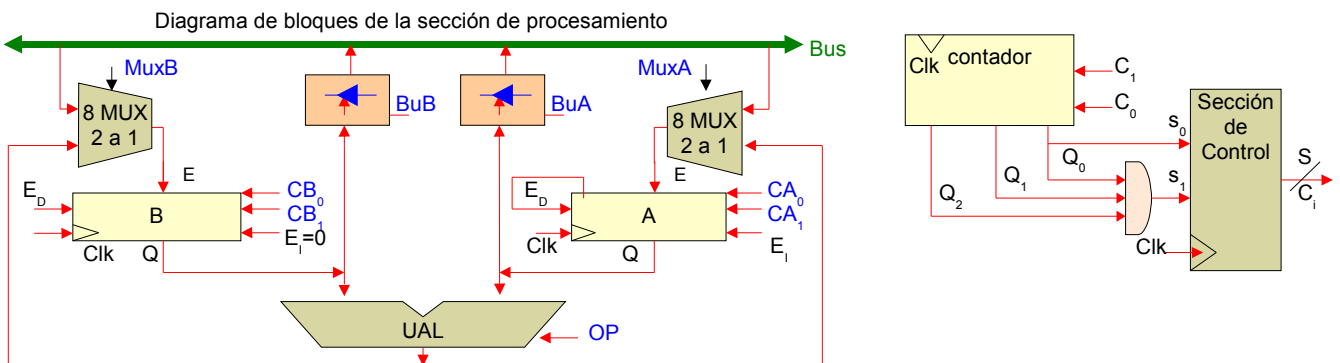
```

1: Declaración: A[8], B[8], Cont[3];
2: A ← Bus;
3: B ← Bus;
4: for Cont = 0 to 7 do
5:   if Cont es par then
6:     A ← A + B; A ← A / 2
7:   else
8:     B ← B - A; B ← B * 2
9:   endif;
10: endfor;
11: Bus ← A;
12: Bus ← B;
13: Parar;
  
```

- a) Diseñe la **sección de procesamiento** que permita realizar este algoritmo utilizando los módulos dibujados abajo: 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. (En concreto dos, el registro A y el registro B).
- b) Diseñe la **sección de control** que ejecute este algoritmo con la sección de procesamiento diseñada en el apartado a), empleando una memoria ROM.

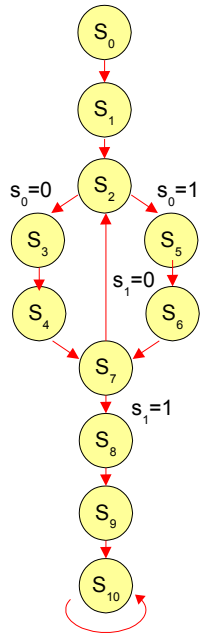


- a) Antes de proponer un diseño para la sección de procesamiento hagamos unas consideraciones previas sobre los requerimientos del algoritmo. La sección de procesamiento tiene que resolver cada uno de estos requerimientos:
- ✓ Registro A. Puede ser cargado desde el bus y desde la salida de la ALU. Por ello se precisan ocho multiplexores 2 a 1 en su entrada para seleccionar cuál es el origen de la carga. Puede ser fuente para el bus y para la entrada de la ALU. Por ello, su salida debe estar conectada a estos dos destinos.
  - ✓ Registro B. Ídem.
  - ✓ Bucle for. Utiliza el registro Cont como índice de bucle. Para el registro Cont utilizaremos el contador disponible. El final de la cuenta lo detecta un circuito combinacional. El caso de Cont par se detecta cuando el bit menos significativo del contador es '0'. El caso de Cont impar se detecta cuando el bit menos significativo del contador es '1'.
  - ✓ Bus. Puede recibir datos de A y de B. Debe estar aislado mediante circuitos triestado.
  - ✓ Multiplicaciones y divisiones por 2. Las divisiones en A con desplazamientos hacia la derecha. Por si el contenido en A fuera negativo, por la izquierda entra otra vez el bit más significativo. Las multiplicaciones en B con desplazamiento hacia la izquierda. Por la derecha entra siempre '0'.



b)

Diagrama de estados de la sección de control



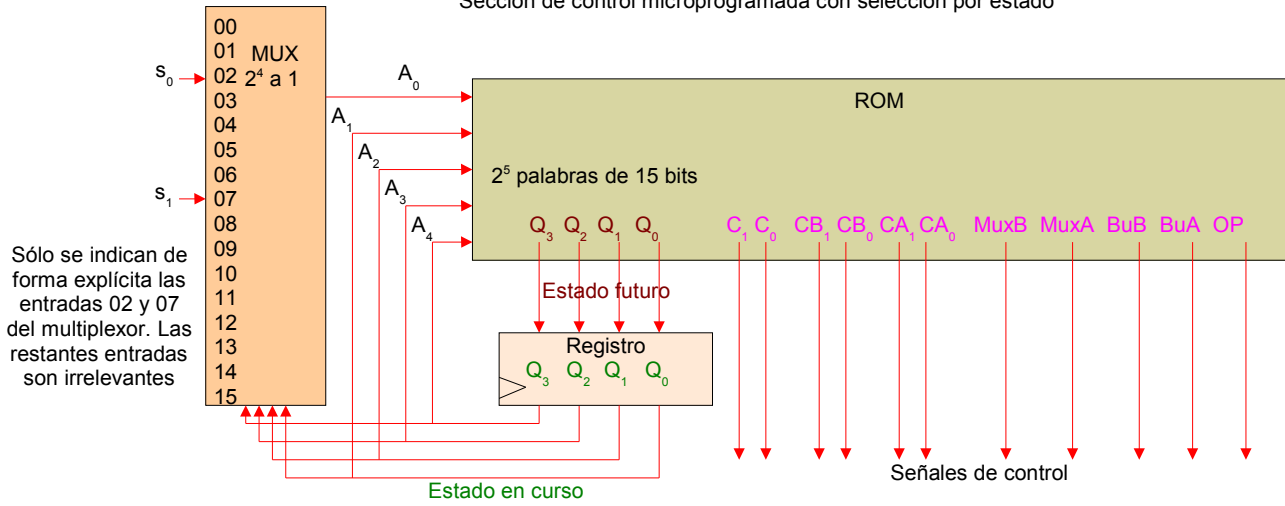
Señal de control	Operación controlada
$C_0$ y $C_1$	Resetean o incrementan el contador
$CA_0$ y $CA_1$	Cargas y desplazamientos en A
$CB_0$ y $CB_1$	Cargas y desplazamientos en B
MuxA	Selecciona la entrada en A
MuxB	Selecciona la entrada en B
BuA	Abre al bus la salida de A
BuB	Abre al bus la salida de B
OP	Selecciona el tipo de operación

Estado actual	Estado siguiente	Microoperaciones efectuadas	Señales de control activas
$S_0$	$S_1$	$A \leftarrow \text{Bus}$	MuxA=1, $CA_1$ $CA_0=11$
$S_1$	$S_2$	$B \leftarrow \text{Bus}$ , Cont $\leftarrow 0$	MuxB=1, $CB_1$ $CB_0=11$ , $C_1C_0=01$
$S_2$	$S_3$ si $s_0=0$ $S_5$ si $s_0=1$	ninguna	ninguna
$S_3$	$S_4$	$A \leftarrow A+B$	$CA_1$ $CA_0=11$
$S_4$	$S_7$	$A \leftarrow A/2$ , Cont $\leftarrow \text{Cont}+1$	$CA_1$ $CA_0=01$ , $C_1C_0=10$
$S_5$	$S_6$	$B \leftarrow B-A$	$CB_1$ $CB_0=11$ , OP=1
$S_6$	$S_6$	$B \leftarrow B*2$ , Cont $\leftarrow \text{Cont}+1$	$CB_1$ $CB_0=10$ , $C_1C_0=10$
$S_7$	$S_2$ si $s_1=0$ $S_8$ si $s_1=1$	ninguna	ninguna
$S_8$	$S_9$	Bus $\leftarrow A$	BuA=1
$S_9$	$S_{10}$	Bus $\leftarrow B$	BuB=1
$S_{10}$	$S_{10}$	ninguna	ninguna

Señal de condición	significado
$s_0$	$s_0=0$ cuando Cont es par y $s_0=1$ cuando Cont es impar
$s_1$	$s_1=0$ cuando $\text{Cont} \neq 7$ y $s_1=1$ cuando $\text{Cont}=7$

Estado actual	Codificación asignada			
	$Q_3$	$Q_2$	$Q_1$	$Q_0$
$S_0$	0	0	0	0
$S_1$	0	0	0	1
$S_2$	0	0	1	0
$S_3$	0	0	1	1
$S_4$	0	1	0	0
$S_5$	0	1	0	1
$S_6$	0	1	1	0
$S_7$	0	1	1	1
$S_8$	1	0	0	0
$S_9$	1	0	0	1
$S_{10}$	1	0	1	0

Sección de control microprogramada con selección por estado



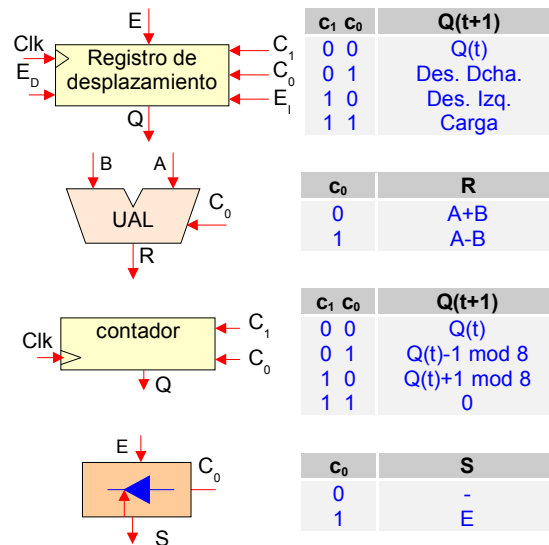
Estado en curso	Dirección de la ROM		Contenido de la ROM										
	Estado en curso Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>	Condición s <sub>1</sub> ó s <sub>0</sub>	Estado futuro Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>	Señales de control									
				C <sub>1</sub> C <sub>0</sub>	CB <sub>1</sub> CB <sub>0</sub>	CA <sub>1</sub> CA <sub>0</sub>	MuxB	MuxA	BuB	BuA	OP		
S <sub>0</sub>	0 0 0 0	0	0 0 0 1	0 0 0 0	0 0 0 0	1 1	0	1	0	0	0	0	0
	0 0 0 0	1	0 0 0 1	0 0 0 0	0 0 0 0	1 1	0	1	0	0	0	0	0
S <sub>1</sub>	0 0 0 1	0	0 0 1 0	0 0 0 0	0 0 0 0	0 0	0	0	0	0	0	0	0
	0 0 0 1	1	0 0 1 0	0 0 0 0	0 0 0 0	0 0	0	0	0	0	0	0	0
S <sub>2</sub>	0 0 1 0	0 si s <sub>0</sub> =0	0 0 1 1	0 0 0 0	0 0 0 0	0 0	0	0	0	0	0	0	0
	0 0 1 0	1 si s <sub>0</sub> =1	0 1 0 1	0 0 0 0	0 0 0 0	0 0	0	0	0	0	0	0	0
S <sub>3</sub>	0 0 1 1	0	0 1 0 0	0 0 0 0	0 0 0 0	1 1	0	0	0	0	0	0	0
	0 0 1 1	1	0 1 0 0	0 0 0 0	0 0 0 0	1 1	0	0	0	0	0	0	0
S <sub>4</sub>	0 1 0 0	0	0 1 1 1	1 0 0 0	0 0 0 0	0 1	0	0	0	0	0	0	0
	0 1 0 0	1	0 1 1 1	1 0 0 0	0 0 0 0	0 1	0	0	0	0	0	0	0
S <sub>5</sub>	0 1 0 1	0	0 1 1 0	0 0 1 1	0 0 0 0	0 0	0	0	0	0	0	1	0
	0 1 0 1	1	0 1 1 0	0 0 1 1	0 0 0 0	0 0	0	0	0	0	0	0	1
S <sub>6</sub>	0 1 1 0	0	0 1 1 1	1 0 1 0	0 0 0 0	0 0	0	0	0	0	0	0	0
	0 1 1 0	1	0 1 1 1	1 0 1 0	0 0 0 0	0 0	0	0	0	0	0	0	0
S <sub>7</sub>	0 1 1 1	0 si s <sub>1</sub> =0	0 0 1 0	0 0 0 0	0 0 0 0	0 0	0	0	0	0	0	0	0
	0 1 1 1	1 si s <sub>1</sub> =1	1 0 0 0	0 0 0 0	0 0 0 0	0 0	0	0	0	0	0	0	0
S <sub>8</sub>	1 0 0 0	0	1 0 0 1	0 0 0 0	0 0 0 0	0 0	0	0	0	0	0	1	0
	1 0 0 0	1	1 0 0 1	0 0 0 0	0 0 0 0	0 0	0	0	0	0	0	1	0
S <sub>9</sub>	1 0 0 1	0	1 0 1 0	0 0 0 0	0 0 0 0	0 0	0	0	0	0	0	1	0
	1 0 0 1	1	1 0 1 0	0 0 0 0	0 0 0 0	0 0	0	0	0	0	0	1	0
S <sub>10</sub>	1 0 1 0	0	1 0 1 0	0 0 0 0	0 0 0 0	0 0	0	0	0	0	0	0	0
	1 0 1 0	1	1 0 1 0	0 0 0 0	0 0 0 0	0 0	0	0	0	0	0	0	0

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

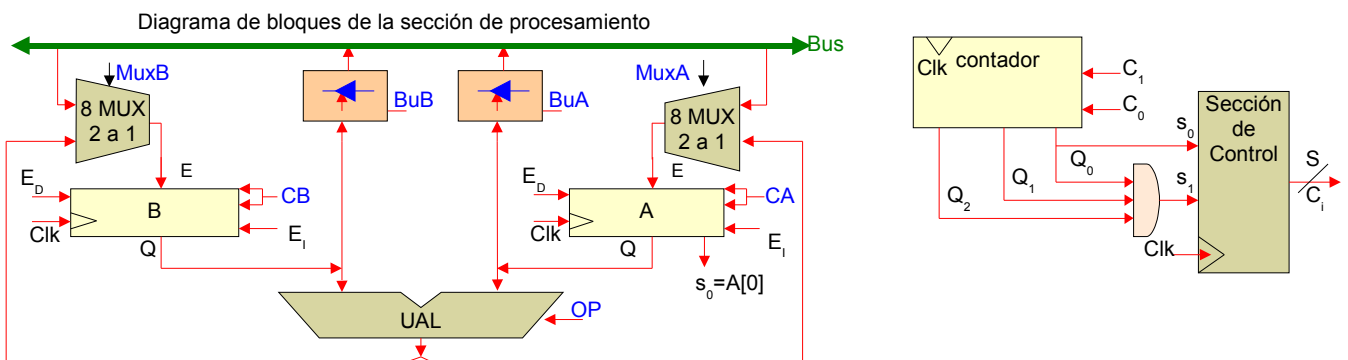
```

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

- a) Diseñe la **sección de procesamiento** que permita realizar este algoritmo utilizando los módulos dibujados abajo: 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 (multiplexores, demultiplexores, codificadores, decodificadores, etc.) que considere necesarios. Debe tener en cuenta que al bus vuelcan datos múltiples dispositivos. (en concreto dos, el registro *A* y el registro *B*).
- b) Diseñe la **sección de control** que ejecute este algoritmo con la sección de procesamiento diseñada en el apartado a) empleando una memoria ROM.

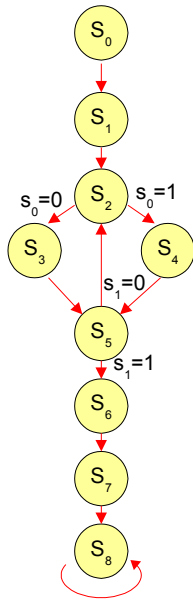


- a) Antes de proponer un diseño para la sección de procesamiento hagamos unas consideraciones previas sobre los requerimientos del algoritmo. La sección de procesamiento tiene que resolver cada uno de estos requerimientos:
- ✓ Registro *A*. Puede ser cargado desde el bus y desde la salida de la ALU. Por ello se precisan ocho multiplexores 2 a 1 en su entrada para seleccionar cuál es el origen de la carga. Puede ser fuente para el bus y para la entrada de la ALU. Por ello, su salida debe estar conectada a estos dos destinos. Su capacidad de desplazamiento no será utilizada. Esto tiene dos consecuencias:
    - 1ª Las entradas de control sólo tendrán valores  $c_1c_0=00$  ó  $c_1c_0=11$ . Como en todo caso tienen ambas el mismo valor, pueden estar unidas, así sólo se necesita una señal de control para ambas.
    - 2ª Las entradas  $E_D$  y  $E_I$  pueden estar conectadas a cualquier valor irrelevante.
  - ✓ Registro *B*. Ídem.
  - ✓ Bucle *while*. Utiliza el registro *Cont* como índice de bucle. Para el registro *Cont* utilizaremos el contador disponible. El final de la cuenta lo detecta un circuito combinacional. El caso de *A* par se detecta cuando el bit menos significativo de *A* es '0'. El caso de *A* impar se detecta cuando el bit menos significativo de *A* es '1'.
  - ✓ Bus. Puede recibir datos de *A* y de *B*. Debe estar aislado mediante circuitos triestado.



b)

Diagrama de estados de la sección de control



Señal de control	Operación controlada
$C_0$ y $C_1$	Resetean o incrementan el contador
CA	Carga en paralelo en A
CB	Carga en paralelo en B
MuxA	Selecciona la entrada en A
MuxB	Selecciona la entrada en B
BuA	Abre al bus la salida de A
BuB	Abre al bus la salida de B
OP	Selecciona el tipo de operación

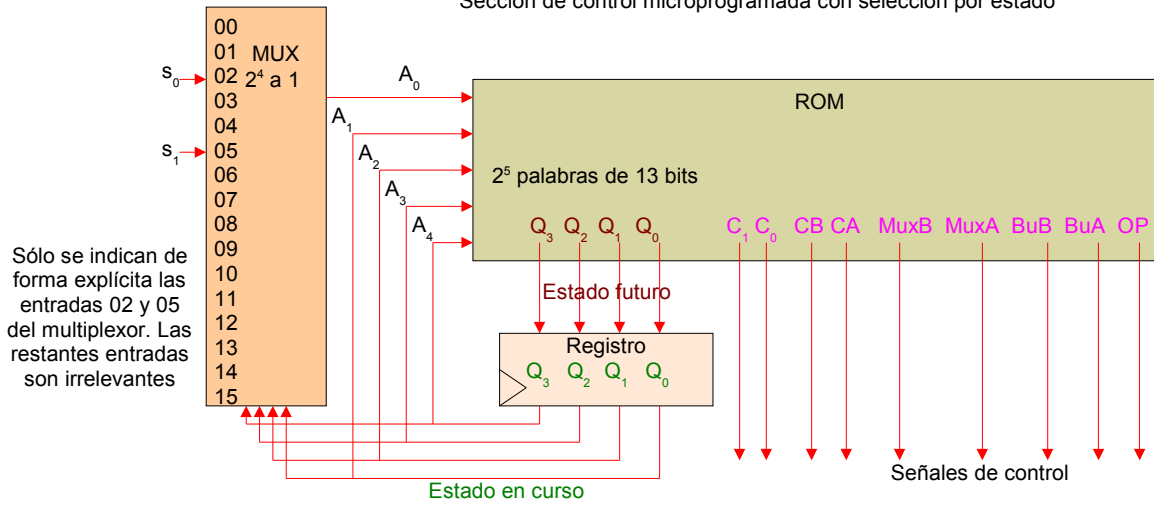
Estado actual	Estado siguiente	Microoperaciones efectuadas	Señales de control activas
$S_0$	$S_1$	$A \leftarrow \text{Bus}$	MuxA=1, CA=1
$S_1$	$S_2$	$B \leftarrow \text{Bus}$ , Cont $\leftarrow 0$	MuxB=1, CB=1, $C_1C_0=11$
$S_2$	$S_3$ si $s_0=0$ $S_4$ si $s_0=1$	ninguna	ninguna
$S_3$	$S_4$	$A \leftarrow A-B$ , Cont $\leftarrow \text{Cont}+1$	CA=1, $C_1C_0=10$ , OP=1
$S_4$	$S_5$	$B \leftarrow B+A$ , Cont $\leftarrow \text{Cont}-1$	CB=1, $C_1C_0=01$
$S_5$	$S_2$ si $s_1=0$ $S_6$ si $s_1=1$	ninguna	ninguna
$S_6$	$S_7$	Bus $\leftarrow A$	BuA=1
$S_7$	$S_8$	Bus $\leftarrow B$	BuB=1
$S_8$	$S_8$	ninguna	ninguna

Señal de condición	significado
$s_0$	$s_0=0$ cuando A es par y $s_0=1$ cuando A es impar
$s_1$	$s_1=0$ cuando Cont $\neq 7$ y $s_1=1$ cuando Cont=7

Estado actual	Codificación asignada			
	$Q_3$	$Q_2$	$Q_1$	$Q_0$
$S_0$	0	0	0	0
$S_1$	0	0	0	1
$S_2$	0	0	1	0
$S_3$	0	0	1	1
$S_4$	0	1	0	0
$S_5$	0	1	0	1
$S_6$	0	1	1	0
$S_7$	0	1	1	1
$S_8$	1	0	0	0



Sección de control microprogramada con selección por estado



Estado en curso	Dirección de la ROM		Contenido de la ROM								
	Estado en curso Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>	condición s <sub>1</sub> ó s <sub>0</sub>	Estado futuro Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>	C <sub>1</sub> C <sub>0</sub>	Señales de control CB CA	MuxB	MuxA	BuB	BuA	OP	
S <sub>0</sub>	0 0 0 0	0	0 0 0 1	0 0	0 1	0	1	0	0	0	
	0 0 0 0	1	0 0 0 1	0 0	0 1	0	1	0	0	0	
S <sub>1</sub>	0 0 0 1	0	0 0 1 0	1 1	1 0	1	0	0	0	0	
	0 0 0 1	1	0 0 1 0	1 1	1 0	1	0	0	0	0	
S <sub>2</sub>	0 0 1 0	0 si s <sub>0</sub> =0	0 0 1 1	0 0	0 0	0	0	0	0	0	
	0 0 1 0	1 si s <sub>0</sub> =1	0 1 0 0	0 0	0 0	0	0	0	0	0	
S <sub>3</sub>	0 0 1 1	0	0 1 0 1	1 0	0 1	0	0	0	0	1	
	0 0 1 1	1	0 1 0 1	1 0	0 1	0	0	0	0	1	
S <sub>4</sub>	0 1 0 0	0	0 1 0 1	0 1	1 0	0	0	0	0	0	
	0 1 0 0	1	0 1 0 1	0 1	1 0	0	0	0	0	0	
S <sub>5</sub>	0 1 0 1	0 si s <sub>1</sub> =0	0 0 1 0	0 0	0 0	0	0	0	0	0	
	0 1 0 1	1 si s <sub>1</sub> =1	0 1 1 0	0 0	0 0	0	0	0	0	0	
S <sub>6</sub>	0 1 1 0	0	0 1 1 1	0 0	0 0	0	0	0	1	0	
	0 1 1 0	1	0 1 1 1	0 0	0 0	0	0	0	1	0	
S <sub>7</sub>	0 1 1 1	0	1 0 0 0	0 0	0 0	0	0	1	0	0	
	0 1 1 1	1	1 0 0 0	0 0	0 0	0	0	1	0	0	
S <sub>8</sub>	1 0 0 0	0	1 0 0 0	0 0	0 0	0	0	0	0	0	
	1 0 0 0	1	1 0 0 0	0 0	0 0	0	0	0	0	0	

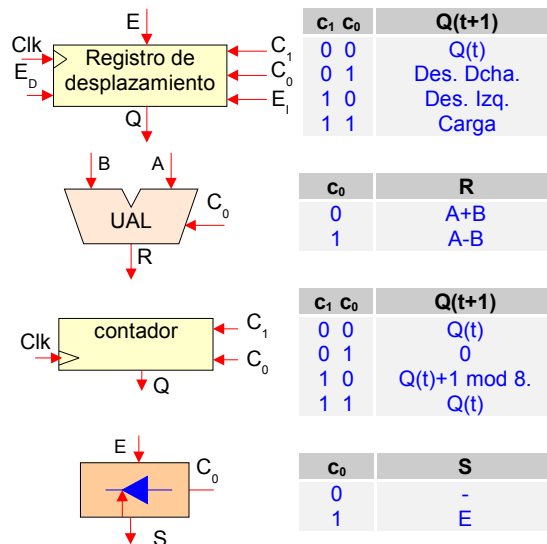
El siguiente algoritmo describe una determinada operación de un sistema digital.

```

1: Declaración: A[8], B[8], Cont[3]; Bus[8]
2: A ← Bus;
3: B ← Bus, Cont ← 0;
4: while A < 128
5:     if Cont < 4 then
6:         A ← A + Cont;
7:     else
8:         A ← B + A;
9:     endif;
10:    Cont = (Cont + 1) mod 8;
11: endwhile;
12: Bus ← A;
13: Bus ← B;
14: Parar;

```

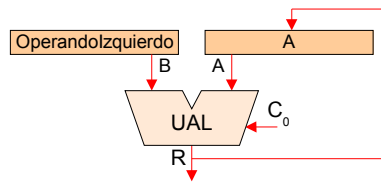
- a) Diseñe la **sección de procesamiento** que permita realizar este algoritmo utilizando los módulos dibujados abajo: registros de desplazamiento de 8 bits, una UAL con dos entradas de 8 bits cada una, un contador módulo-8 y circuitos triestado de conexión unidireccional con control de 8 bits; además de puertas lógicas y los módulos combinacionales (multiplexores, demultiplexores, codificadores, decodificadores, etc.) que considere necesarios. Debe tener en cuenta que al bus vuelcan datos múltiples dispositivos.
- b) Diseñe la **sección de control** que ejecute este algoritmo con la sección de procesamiento diseñada en el apartado a) empleando una memoria ROM.



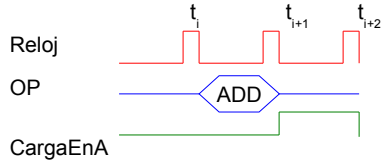
- a) Antes de proponer un diseño para la sección de procesamiento hagamos unas consideraciones previas sobre los requerimientos del algoritmo. La sección de procesamiento tiene que resolver cada uno de estos requerimientos:
- ✓ **Registro A.** Puede ser cargado desde el bus y desde la salida de la ALU. Por ello se precisan ocho multiplexores 2 a 1 en su entrada para seleccionar cuál es el origen de la carga. Puede ser fuente para el bus y para la entrada de la ALU. Por ello, su salida debe estar conectada a estos dos destinos. Su capacidad de desplazamiento no será utilizada. Esto tiene dos consecuencias:
    - 1ª Las entradas de control sólo tendrán valores  $C_1C_0=00$  ó  $C_1C_0=11$ . Como en todo caso tienen ambas el mismo valor, pueden estar unidas, así sólo se necesita una señal de control para ambas.
    - 2ª Las entradas  $E_0$  y  $E_1$  pueden estar conectadas a cualquier valor irrelevante.
  - ✓ **Registro B.** Sólo puede ser cargado desde el bus. Su capacidad de desplazamiento no será utilizada. Esto tiene dos consecuencias:
    - 1ª Las entradas de control sólo tendrán valores  $C_1C_0=00$  ó  $C_1C_0=11$ . Como en todo caso tienen ambas el mismo valor, pueden estar unidas, así sólo se necesita una señal de control para ambas.
    - 2ª Las entradas  $E_0$  y  $E_1$  pueden estar conectadas a cualquier valor irrelevante.
  - ✓ **UAL.** Su entrada izquierda puede proceder del registro B o del contador. Por ello se precisan ocho multiplexores 2 a 1 en su entrada izquierda para seleccionar cuál es el origen del operando izquierdo. Su salida sólo se conecta al registro A. Sólo se utiliza para sumar. Por tanto, su entrada de control puede estar permanentemente a 0'. No necesitamos señal de control para la operación de la UAL.

- ✓ Operación  $A \leftarrow A + \text{OperandoIzquierdo}$ ;  
Pueden estar activas simultáneamente las señales de selección de suma en la ALU y de carga en el registro A.

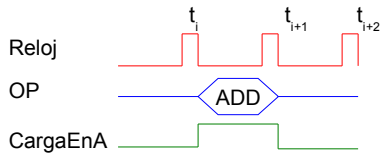
Para evitar que uno de los operandos cambie antes de acabar la operación:



No es necesario:



Podemos aprovechar que la carga en A es en el flanco negativo de la señal correspondiente; cuando ya se ha terminado suma.



- ✓ Bucle *while*. Utiliza el registro *Cont* como índice de bucle. Para el registro *Cont* utilizaremos el contador disponible. Son necesarios dos circuitos combinacionales para detectar:

1º El final del bucle ( $A \geq 128$ ).

Si el registro tuviera más de ocho bits, deberían ser tenidos en cuenta estos dos detalles:

1.1 Siempre es cierta la implicación  $A_7=1 \Rightarrow A \geq 128$ . Es decir:  $\dots 1xxxxxxx_b \Rightarrow A \geq 128$ .

1.2 No siempre es cierta la implicación  $A_7=0 \Rightarrow A < 128$ . Pues podría ser  $A_7=0$  y haber algún bit más significativo  $A_n=1$ .

Combinando ambos detalles, en general es falsa la equivalencia:  $A_7=1 \Leftrightarrow A \geq 128$ .

Pero en nuestro caso, al tratarse de un registro de ocho bits, sí es cierta la equivalencia.

Por tanto, la señal utilizada para discernir si se cumple ( $A \geq 128$ ) será simplemente el bit  $A_7$  del registro A.

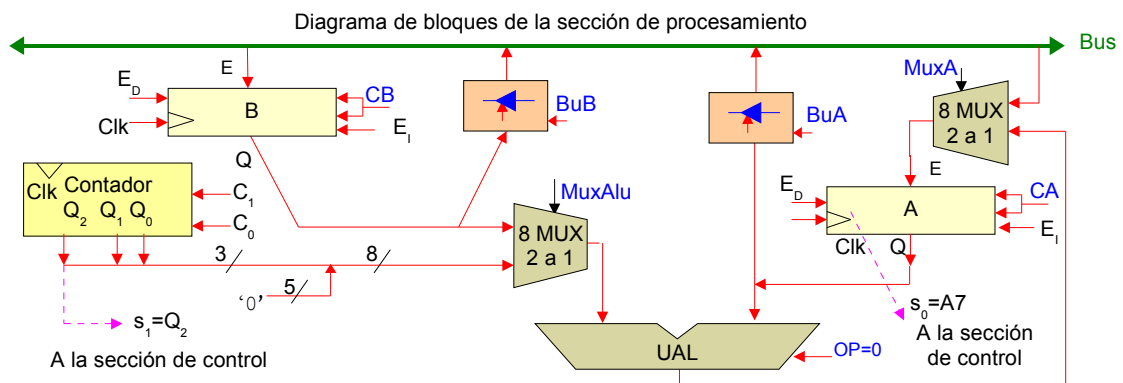
2º  $Cont < 4$

Para  $Cont < 4$  ( $100_b$ ),  $Q_2 = 0$

Para  $Cont \geq 4$  ( $100_b$ ),  $Q_2 = 1$

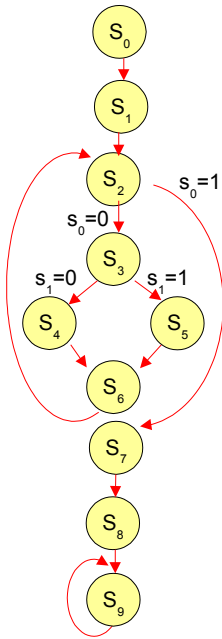
La señal para saber si *Cont* es menor que 4 será simplemente el bit  $Q_2$  del registro *Cont*.

- ✓ Bus. Puede recibir datos de A y de B. Debe estar aislado mediante circuitos triestado.



b)

Diagrama de estados de la sección de control



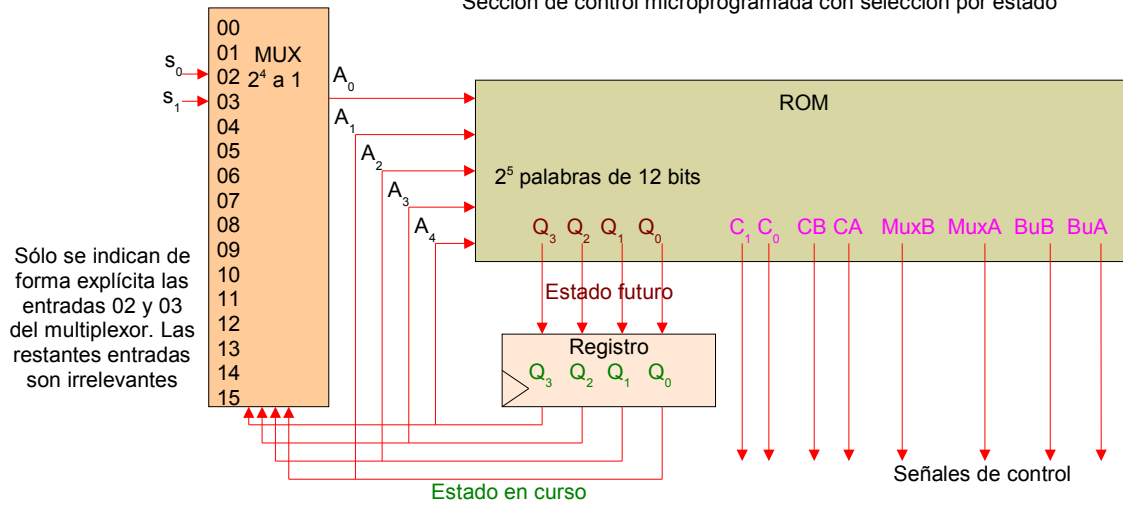
Señal de control	Operación controlada
$C_0$ y $C_1$	Resetean o incrementan el contador
CA	Carga en paralelo en A
CB	Carga en paralelo en B
MuxA	Selecciona la entrada en A
MuxAlu	Selecciona la entrada izquierda de la UAL
BuA	Abre al bus la salida de A
BuB	Abre al bus la salida de B

Estado actual	Estado siguiente	Microoperaciones efectuadas	Señales de control activas
S <sub>0</sub>	S <sub>1</sub>	A ← Bus	MuxA=1, CA=1
S <sub>1</sub>	S <sub>2</sub>	B ← Bus, Cont ← 0	CB=1, C <sub>1</sub> C <sub>0</sub> =01
S <sub>2</sub>	S <sub>3</sub> si s <sub>0</sub> =0 S <sub>7</sub> si s <sub>0</sub> =1	ninguna	ninguna
S <sub>3</sub>	S <sub>4</sub> si s <sub>1</sub> =0 S <sub>5</sub> si s <sub>1</sub> =1	ninguna	ninguna
S <sub>4</sub>	S <sub>6</sub>	A ← A + Cont	CA=1, MuxAlu=0
S <sub>5</sub>	S <sub>6</sub>	A ← A + B	CA=1, MuxAlu=1
S <sub>6</sub>	S <sub>2</sub>	Cont ← (Cont+1) mod 8	C <sub>1</sub> C <sub>0</sub> =10
S <sub>7</sub>	S <sub>8</sub>	Bus ← A	BuA=1
S <sub>8</sub>	S <sub>9</sub>	Bus ← B	BuB=1
S <sub>9</sub>	S <sub>9</sub>	ninguna	ninguna

Señal de condición	significado
s <sub>0</sub>	s <sub>0</sub> =0 cuando Cont<4 y s <sub>0</sub> =1 cuando Cont≥4
s <sub>1</sub>	s <sub>1</sub> =0 cuando Cont<128 y s <sub>1</sub> =1 cuando Cont≥128

Estado actual	Codificación asignada			
	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
S <sub>0</sub>	0	0	0	0
S <sub>1</sub>	0	0	0	1
S <sub>2</sub>	0	0	1	0
S <sub>3</sub>	0	0	1	1
S <sub>4</sub>	0	1	0	0
S <sub>5</sub>	0	1	0	1
S <sub>6</sub>	0	1	1	0
S <sub>7</sub>	0	1	1	1
S <sub>8</sub>	1	0	0	0
S <sub>9</sub>	1	0	0	1

Sección de control microprogramada con selección por estado



Estado en curso	Dirección de la ROM		Contenido de la ROM							
	Estado en curso Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>	Condición s <sub>1</sub> ó s <sub>0</sub>	Estado futuro Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub>	C <sub>1</sub> C <sub>0</sub>	CB CA	MuxAlu	MuxA	BuB	BuA	
S <sub>0</sub>	0 0 0 0	0	0 0 0 1	0 0	0 1	0	1	0	0	
	0 0 0 0	1	0 0 0 1	0 0	0 1	0	1	0	0	
S <sub>1</sub>	0 0 0 1	0	0 0 1 0	0 1	1 0	0	0	0	0	
	0 0 0 1	1	0 0 1 0	0 1	1 0	0	0	0	0	
S <sub>2</sub>	0 0 1 0	0 si s <sub>0</sub> =0	0 0 1 1	0 0	0 0	0	0	0	0	
	0 0 1 0	1 si s <sub>0</sub> =1	0 1 1 1	0 0	0 0	0	0	0	0	
S <sub>3</sub>	0 0 1 1	0 si s <sub>1</sub> =0	0 1 0 0	0 0	0 0	0	0	0	0	
	0 0 1 1	1 si s <sub>1</sub> =1	0 1 0 1	0 0	0 0	0	0	0	0	
S <sub>4</sub>	0 1 0 0	0	0 1 1 0	0 0	0 1	0	0	0	0	
	0 1 0 0	1	0 1 1 0	0 0	0 1	0	0	0	0	
S <sub>5</sub>	0 1 0 1	0	0 1 1 0	0 0	0 1	1	0	0	0	
	0 1 0 1	1	0 1 1 0	0 0	0 1	1	0	0	0	
S <sub>6</sub>	0 1 1 0	0	0 0 1 0	1 0	0 0	0	0	0	0	
	0 1 1 0	1	0 0 1 0	1 0	0 0	0	0	0	0	
S <sub>7</sub>	0 1 1 1	0	1 0 0 0	0 0	0 0	0	0	0	1	
	0 1 1 1	1	1 0 0 0	0 0	0 0	0	0	0	1	
S <sub>8</sub>	1 0 0 0	0	1 0 0 1	0 0	0 0	0	0	1	0	
	1 0 0 0	1	1 0 0 1	0 0	0 0	0	0	1	0	
S <sub>9</sub>	1 0 0 1	0	1 0 0 1	0 0	0 0	0	0	0	0	
	1 0 0 1	1	1 0 0 1	0 0	0 0	0	0	0	0	