Divisores Binarios.

La operación de **división** es algo más compleja que la multiplicación, pero también se realiza en la mayoría de computadores mediante un circuito sumador/restador y algún algoritmo adecuado.

Dado dos operandos, el dividendo **D** y el divisor **d**, el objetivo de la división es calcular el cociente **Q** y el resto **R** tal que

$$\mathbf{D = d * Q + R}$$

con la condición de que el resto sea menor que el divisor, es decir $0 \leq R \leq d$.

Los circuitos que realizan la multiplicación y la división son análogos, pues el producto se puede realizar por sumas sucesivas y el cociente se puede realizar mediante restas sucesivas. Vamos a ver primero el método de lápiz y papel para los números binarios positivos. Para ello seguiremos el siguiente algoritmo:

1. Examinar los bits del dividendo de izquierda a derecha hasta encontrar una cadena mayor que el divisor.
2. Se coloca un 1 en el cociente y se procede a restar el divisor al dividendo.
3. Ahora empieza unas acciones cíclicas: al resto se le añade una cifra del dividendo, si no es mayor que el divisor se añade un 0 al cociente y se baja otra cifra; así hasta que el nuevo resto sea mayor que el divisor y entonces se añade un 1 al cociente y se procede a restar el divisor del resto actual.
4. Este proceso se repite hasta que se acaban todos los bits del dividendo.

Ejemplo:

```
D = 39 = 1 0 0 1 1 1
d =  6 =       1 1 0
```

```
                          1 0 0 1 1 1         | 1 1 0
                          1 1 0   no resta -->  0 1 1 0     Cociente
        resto parcial     1 0 0 1
                            1 1 0    resta
        resto parcial     0 0 1 1 1
                              1 1 0   resta
        resto parcial     0 0 0 0 1 1
                              1 1 0  no resta
        Resto             0 0 0 0 1 1

        Cociente = 6 = 1 1 0
        Resto    = 3 =   1 1
```

El algoritmo de la división se basa en prueba y error. Al igual que con los números en decimal, la división binaria busca el número que multiplicado por el divisor nos da el mayor número que se puede restar al dividendo sin que nos dé un valor negativo. En la división binaria los valores a probar son siempre o el uno o el cero, primero probamos con el uno esto nos hará restar al dividendo el divisor, eligiendo los bits adecuados, si la resta es negativa en vez de un uno cambiamos por un cero y bajamos una nueva cifra, si el resultado fue positivo dejamos el valor de la resta al cual se le añade una nueva cifra del dividendo y seguimos con la operación.

Pero a la hora de realizar un circuito digital que realice la división es mejor cambiar un poco el método y en vez de desplazar el divisor a la derecha, desplazaremos el resto parcial a la izquierda (en la práctica es como multiplicarlo por 2) y operamos con el divisor fijo. Veamos como realizaríamos la anterior división con esta variante al método propuesto.

Ejemplo:
```
D = 39 = 1 0 0 1 1 1
d =  6 =       1 1 0
```

```
             1 0 0 1 1 1              | 1 1 0
             1 1 0   no resta        0 1 1 0     Cociente
   R1        1 0 0 1 1 1
   2*R1    1 0 0 1 1 1 0
              1 1 0        resta
   R2        0 1 1 1 1 0
   2*R2    0 1 1 1 1 0 0
             1 1 0          resta
   R3        0 0 1 1 1 0
   2*R3    0 0 1 1 1 0 0
             1 1 0   no resta
  Resto      0 1 1
```

```
Cociente = 6 = 1 1 0
Resto    = 3 =   1 1
```

El problema es que el último resto parcial Rn no contiene exactamente el resto, pues el valor verdadero se calcula como

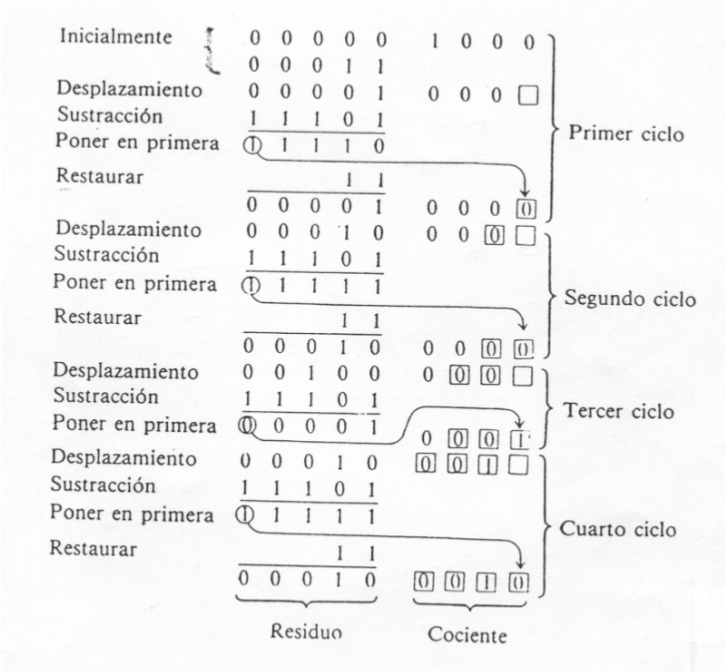
$$R = Rn * 2^{-n}$$

Está claro que es más difícil de automatizar la división debido al proceso de ensayo. Las tareas a realizar por el circuito son: acomodar metódicamente el divisor con relación al dividendo y realizar una sustracción, en complemento a dos. Si el resultado es cero o positivo, se pone el bit cociente como 1, el resultado de la resta se amplía con otro dígito del dividendo y el divisor se acomoda para otra sustracción. Esta técnica se realiza utilizando una estructura de registro similar a la que se utilizó para realizar la multiplicación, y se muestra en el circuito de la hoja siguiente.

Ejemplo: Realizar segun el método de la división con restauración el siguiente cociente: D/d.

D = 8 (1000) d = 3 (0011) con (-3) = (1101)

	Ac	D	d
Inicialmente:	0000	1000	0011
Finalmente:	0010	0010	0011
	resto	cociente	

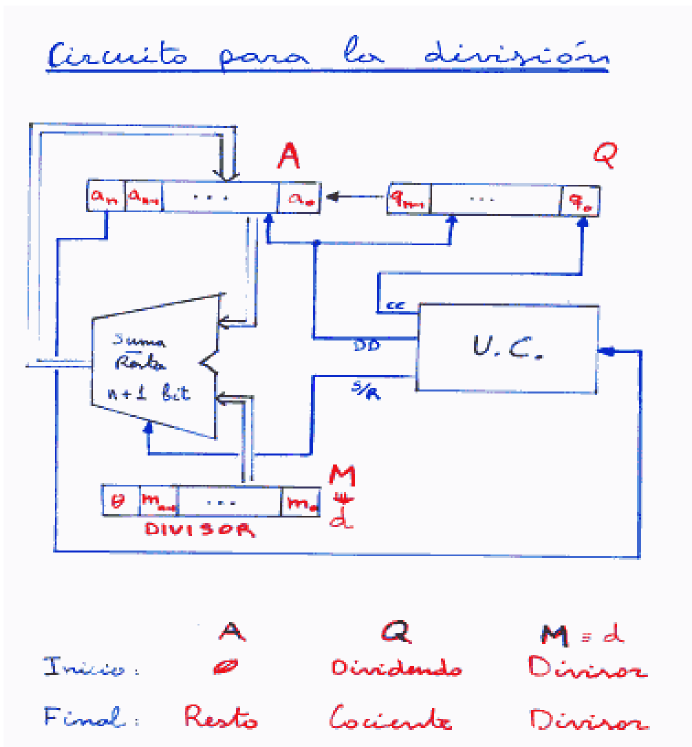


División por el método de restauración.

Tal como se ha indicado, para evitar la utilización de circuitos comparadores de elevado coste, la comparación se realiza entre el dividendo y el divisor se realiza mediante una resta.

Al realizar la resta, una respuesta positiva indica que el divisor es más pequeño, y se coloca un 1 en el cociente. Una respuesta negativa indica que el divisor es mayor y por tanto que la resta no era necesaria, por lo que hay que volver a sumar el divisor al dividendo. A esta operación se le llama **restaurar** el valor original del dividendo, dándole nombre al método.

Todo este proceso se puede realizar sobre la estructura de registros vistos anteriormente y utilizando el algoritmo descrito a continuación:



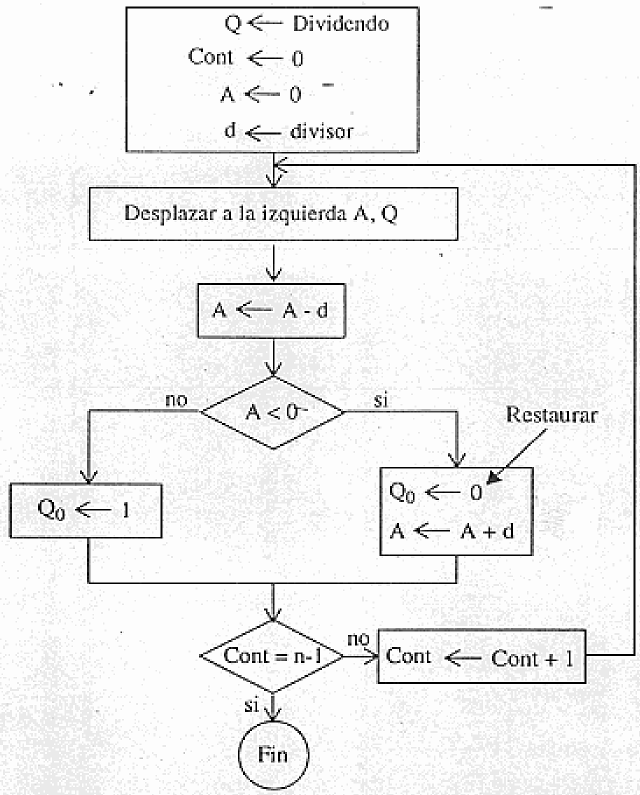


Figura 4-53 Diagrama de flujo del método de restauración

Paso	Acción		A	Q	d	Cont
0	Inicializar registros	0	0000	1011	0101	0
1	Desplazar A,Q a izqda	0	0001	0110		
	$A \leftarrow A - d = A + C2(d)$	1	1100	0110		
	$A < 0 \Rightarrow Q_0 = 0$	1	1100	0110		
	$A \leftarrow A + d$ (Restaurar)	0	0001	0110		
	$Cont \leftarrow Cont + 1$					1
2	Desplazar A,Q a izqda	0	0010	1100		
	$A \leftarrow A - d = A + C2(d)$	1	1101	1100		
	$A < 0 \Rightarrow Q_0 = 0$	1	1101	1100		
	$A \leftarrow A + d$ (Restaurar)	0	0010	1100		
	$Cont \leftarrow Cont + 1$					2
3	Desplazar A,Q a izqda	0	0101	1000		
	$A \leftarrow A - d = A + C2(d)$	0	0000	1000		
	$A \geq 0 \Rightarrow Q_0 = 1$	0	0000	1001		
	$Cont \leftarrow Cont + 1$					3
4	Desplazar A,Q a izqda	0	0001	**0010**		
	$A \leftarrow A - d = A + C2(d)$	1	1100	**0010**		
	$A < 0 \Rightarrow Q_0 = 0$	1	1100	**0010**		
	$A \leftarrow A + d$ (Restaurar)	0	**0001**	**0010**		
	Fin		Resto	Cociente		

Tabla 4- 11: Ejemplo de aplicación del método de restauración

División sin restauración.

Es posible acelerar este algoritmo mediante la eliminación de la restauración que implica una suma para volver a recobrar un dato. Para ello debemos darnos cuenta de que si la restauración la escribimos como

$$(R_j)_A \leftarrow (R_j)_A + d \qquad (1)$$

esta va seguida siempre de un desplazamiento (*2) y una resta

$$(R_{j+1})_A \leftarrow (2*R_j)_A - d \qquad (2)$$

donde 2* corresponde a un desplazamiento a izquierda de los registros A y Q. Estas dos ecuaciones pueden combinarse en una sola para la obtención de los sucesivos restos parciales sobre el registro R:

$$(R_{j+1})_A \leftarrow (2*R_j)_A - d = 2 * ((R_j)_A + d) - d \qquad (3)$$

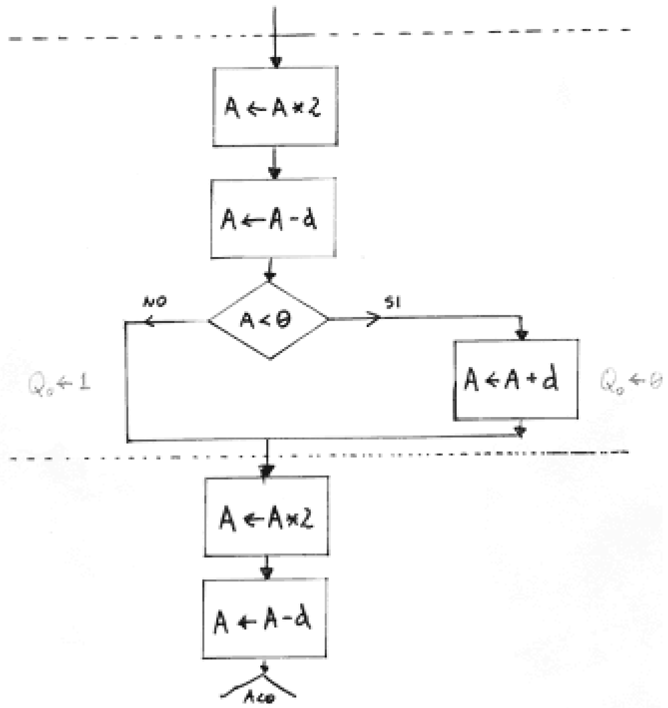
En esta idea se basa el método de la no restauración, en que si el bit

$Q_{n-i} = 1$ el resto parcial se evalúa segun la ecuación (2) y

$Q_{n-i} = 0$ el resto parcial se evalúa segun la ecuación (3).

En la figura siguiente se muestra el algoritmo de método de no restauración.

DIVISION SIN RESTAURACION

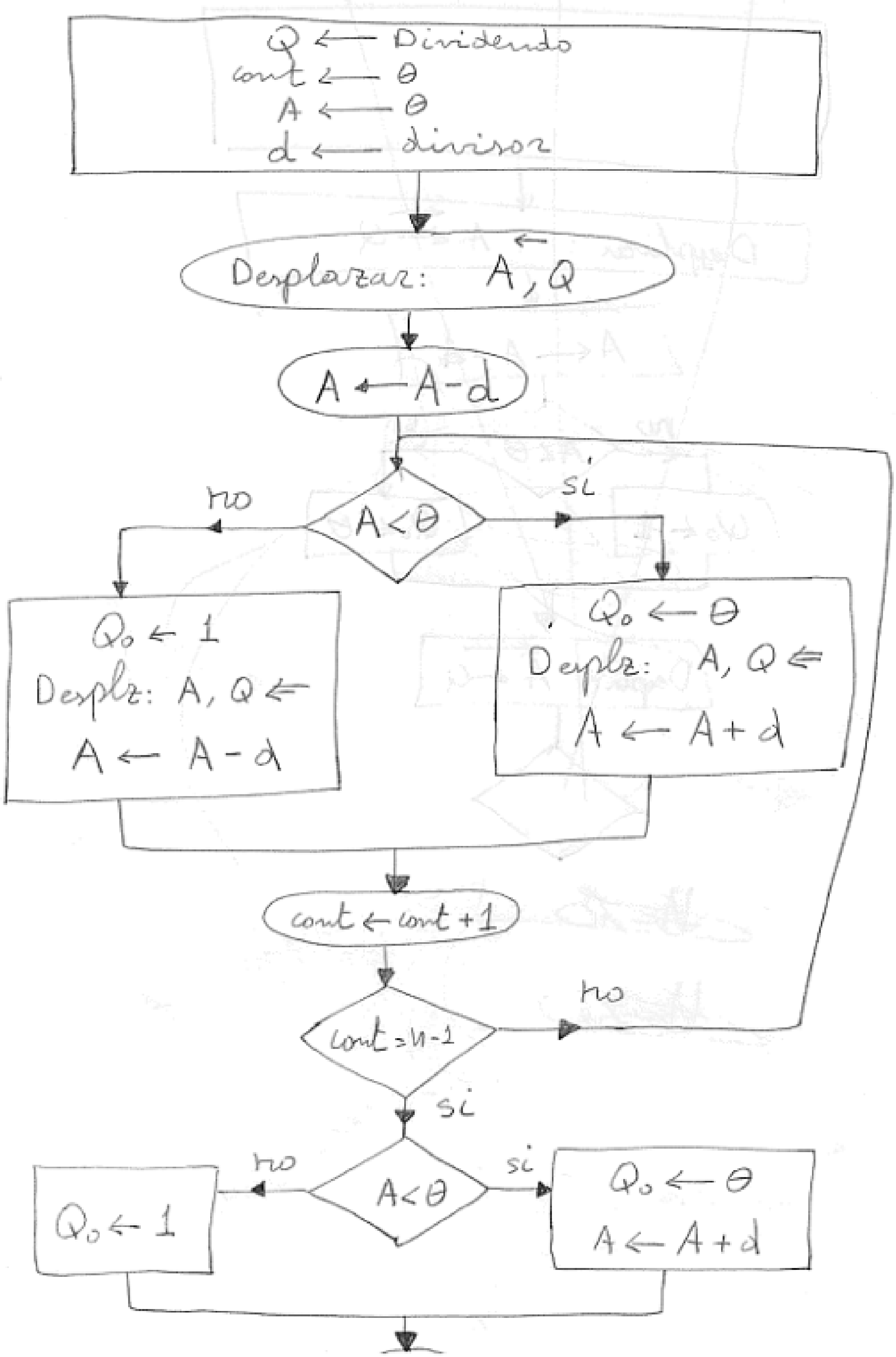


Cuando $A<0$:

$$A \leftarrow [2 * (\underline{A+d})] - d$$

$$A \leftarrow [2 * A + 2*d - d]$$

$$A \leftarrow \underline{2 * A} + d$$


Q ← Dividendo
cont ← 0
A ← 0
d ← divisor
Desplazar: A, Q ←
A ← A-d
A<0
no
si
Q0 ← 1
Desplz: A, Q ⇐
A ← A-d
Q0 ← 0
Desplz: A, Q ⇐
A ← A+d
cont ← cont+1
cont=n-1
no
si
A<0
no
si
Q0 ← 1
Q0 ← 0
A ← A+d

Ejemplo: Realizar segun el método de la división <u>sin-restauración</u> el siguiente cociente: D/d.

D = 8 (1000) d = 3 (0011) con (-3) = (1101)

	Ac	D	d
Inicialmente:	0000	1000	0011
Finalmente:	0010	0010	0011
	resto	cociente	

Inicialmente	0 0 0 0 0	1 0 0 0	Primer ciclo
	0 0 0 1 1		
Corrimiento	0 0 0 0 1	0 0 0 $\boxed{}$	
Sustracción	1 1 1 0 1		
Poner en primera	1 1 1 1 0	0 0 0 $\boxed{0}$	
Desplazamiento	1 1 1 0 0	0 0 $\boxed{0}$ $\boxed{}$	Segundo ciclo
Suma	0 0 0 1 1		
Poner en primera	1 1 1 1 1	0 0 $\boxed{0}$ $\boxed{0}$	
Desplazamiento	1 1 1 1 0	0 $\boxed{0}$ $\boxed{0}$ $\boxed{}$	Tercer ciclo
Suma	0 0 0 1 1		
Poner en primera	0 0 0 0 1	0 $\boxed{0}$ $\boxed{0}$ $\boxed{1}$	
Desplazamiento	0 0 0 1 0	$\boxed{0}$ $\boxed{0}$ $\boxed{1}$ $\boxed{}$	Cuarto ciclo
Sustracción	1 1 1 0 1		
Poner en primera	1 1 1 1 1	$\boxed{0}$ $\boxed{0}$ $\boxed{1}$ $\boxed{0}$	
		Cociente	
Suma	1 1 1 1 1		Restauración del residuo
	0 0 0 1 1		
	0 0 0 1 0		
	Residuo		

Por otra parte hay que destacar que no existen algoritmos simples que efectúen la división de números con signo y que se puedan comparar con la multiplicación. Se necesitará un procesado previo de los operandos y un procesamiento posterior de los resultados. Aunque siempre se podrán transformar los operandos en valores positivos, utilizar uno de los algoritmos antes analizados y transformar los resultados en los valores correctos con el signo que se necesite.