

Ejercicio 1

Se tienen dos vectores A y B de 30 elementos cada uno, cuyas direcciones base se encuentran en los registros \$s0 y \$s1 respectivamente. Implementar un programa mediante instrucciones MIPS que realice lo siguiente:

$$A[0] = B[29], A[1] = B[28], A[2] = B[27], \dots, A[28] = B[1], A[29] = B[0]$$

(2 puntos)

Ejercicio 2

En un sistema de memoria virtual paginado el espacio de direcciones es de 128KB, mientras que el espacio de memoria es de 32KB. El tamaño de página es de 8KB. Indicar:

- a) Formato de la dirección virtual y formato de la dirección física.

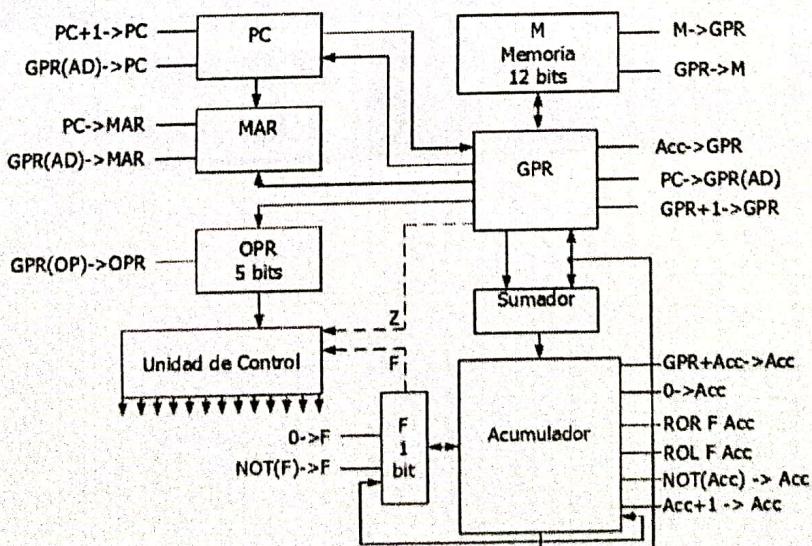
Imagínese que se comienza con la memoria principal vacía y que el algoritmo de sustitución que se posee es el LRU. Si se llaman a las siguientes páginas en el orden: 13, 8, 9, 8, 4, 13, 2, 8, 9. ¿Cuál será el contenido de la tabla de páginas después de la última llamada?

- b) A partir de la tabla de páginas anterior, determinar los rangos de direcciones virtuales que producirán un fallo de página en caso de ser accedidas por la CPU.
- c) Dirección física que se corresponde con las siguientes direcciones virtuales del proceso, en caso de que la traducción sea posible:
- a.1) 03A8F.
 - a.2) 13A8F.

(3 puntos)

Ejercicio 3

RIZ m: Calcular el número de ceros que se encuentren a la derecha del 1 menos significativo del ACC, guardando dicho resultado en la dirección de memoria m. Por ejemplo, si el ACC tuviera el contenido 110011110000, la instrucción guardaría un 4 en la dirección m. En el caso particular en el que el ACC estuviera inicialmente a cero, la instrucción debe guardar el valor 111111111111 en la dirección m (es decir, los 12 bits a 1)



Ejercicio 4

Se tiene un vector A cuya dirección base se encuentra en el registro \$s0, un vector B cuya dirección se encuentra en el registro \$s1 y un vector C en \$s2, cuyo número de elementos N se encuentra en el registro \$s3. Realizar un programa mediante instrucciones MIPS que:

- Inserte en la posición C[i] un 0 si el valor de A es mayor que B.
- Inserte en la posición C[i] un 1 si el valor de A es menor que B.
- Inserte en la posición C[i] el valor de A si A y B son iguales

Ejercicio 5

Una computadora tiene una memoria principal de 256Kx16 y una memoria caché asociativa por conjuntos de 3 vías, que utiliza bloques de 4 palabras y que puede almacenar 6K palabras de memoria principal.

- Determinar la organización de la memoria caché.
- Determinar el tamaño de la memoria caché.
- Teniendo en cuenta que en memoria principal se encuentran dos vectores de elementos de 16 bits, almacenados en las direcciones representadas por el esquema siguiente:

Vector A. Dirección de comienzo: 3B018(hex)											
Contenido	-4	22	53	102	-1	0	67	521	-13	42	

Vector B. Dirección de comienzo: 03818(hex)											
Contenido	15	-12	23	7	40	32	-18	98	11	3	

Representar gráficamente el contenido de la memoria caché después de 3 iteraciones del bucle de código que se presenta a continuación:

```

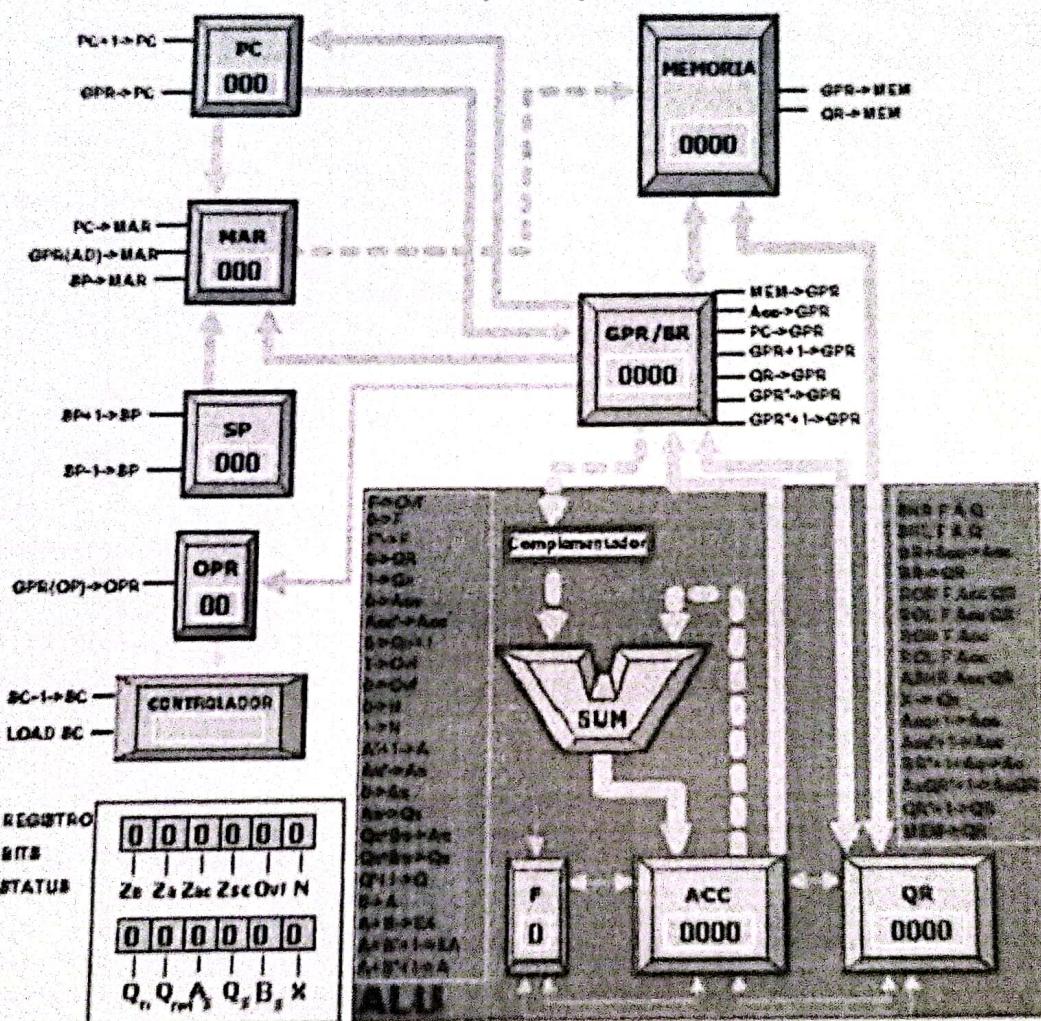
1: int sp;
2:
3: sp = 0;
4: for (int i=0; i<n; ++i)
5:     sp += A[i]*B[i];

```

(Representar únicamente las líneas de caché que han sido modificadas por el programa en ejecución).

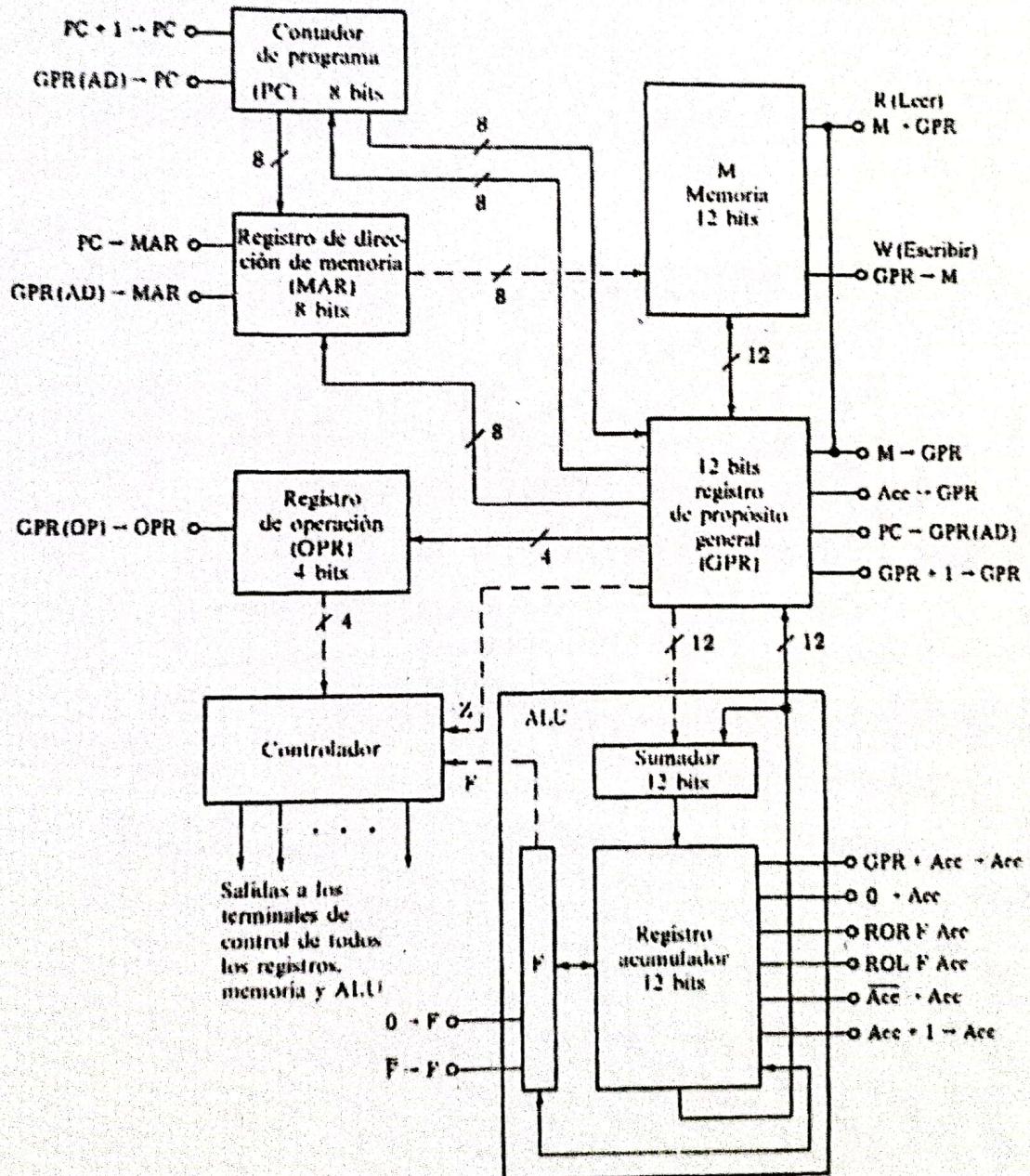
Ejercicio 4

Tomando como punto de partida la Computadora Mejorada de SiCoMe, indicar las instrucciones necesarias para realizar tanto *push* como *pop* en el tope de pila, teniendo en cuenta las cuatro configuraciones de esta.



Ejercicio 1

Tomando como punto de partida la estructura de la Computadora Mejorada y completándola con las microoperaciones que sean necesarias, implementar mediante control microprogramado la instrucción "RZM m" que debe procesar el contenido de la posición de memoria "m" para eliminar todos los ceros que se encuentren a la izquierda del número, rellenando con ceros por la derecha. Ejemplo: si el valor almacenado en la posición de memoria "m" es 000001110110, debe quedar como resultado 111011000000. Incluir también el ciclo de búsqueda. El único bit de status existente es el F. Únicamente se pide la tabla de la CROM en formato RTL y la tabla de la Lógica de Control de Bifurcación.



Ejercicio 2

Realizar un programa en MIPS que ejecute el siguiente código:

```
for(int i=0; i<n-1; i++)  
    for(int j=0; j<n-1; j++)  
    {  
        aux=menor(v[j], v[j+1]);  
        v[j]=v[j+1];  
        v[j+1] = aux;  
    }
```

Donde la dirección de memoria del vector v se encuentra en la dirección \$S0 y de la variable n en la dirección \$S1;

Ejercicio 3

En un sistema de memoria virtual paginado el espacio de direcciones es de 2 MB, mientras que el espacio de memoria es de 64 KB. El tamaño de página es de 1KB. Determinar:

- Longitud en bits de una dirección virtual y de una dirección física.
- Número de páginas y bloques del sistema de memoria.
- Dimensiones de las tablas de páginas, según se implementen mediante una memoria de acceso aleatorio o a través de una memoria asociativa.
- Un programa tiene una extensión de 12 páginas consecutivas en memoria virtual. Su primera página comienza en la dirección virtual 002C00 (en hexadecimal). El contenido de la tabla de páginas del programa es el siguiente:

Página	Bloque
0	30
1	24
3	2

Página	Bloque
4	15
6	56
9	39

Página	Bloque
10	25
11	7

Realizar la traducción de las siguientes direcciones virtuales a direcciones físicas:

- 4.1. 002C35
- 4.2. 0034D1
- 4.3. 012127
- 4.4. 005959

Se dispone de un sistema de memoria que utiliza direcciones de 32 bits. La unidad mínima direccionable es el byte. En memoria se encuentran dos arrays de elementos de 8 bits, almacenados en las direcciones representadas por el esquema siguiente:

Array A. Dirección de comienzo: 00101AF0												
Contenido	12	4	67	81	231	89	106	198	206	11	39	162

Array B. Dirección de comienzo: 3C50BAF0												
Contenido	102	44	54	241	163	25	175	102	39	89	156	250

Suponiendo que el sistema de memoria incorpora una memoria caché con mapeo directo que permite almacenar 32 K palabras de memoria principal y que el tamaño de bloque es de 8 palabras, representar gráficamente el contenido de la memoria caché después de 3 iteraciones del bucle de código que se presenta a continuación:

```
1: int sp;
2:
3: sp = 0;
4: for (int i=0; i<n; ++i)
5:     sp += A[i]*B[i];
```

Nota: Representar únicamente las líneas de caché que han sido modificadas por el programa en ejecución.

Ejercicio 5

Un sistema de memoria de 256×8 incorpora una memoria caché de mapeo directo que puede almacenar 16 palabras de la memoria principal. El tamaño de bloque es de 4 palabras. En la memoria principal se encuentra una tabla A de 24 elementos, almacenada como indica la siguiente figura:

08	A[0,0]	A[0,1]	A[0,2]	A[0,3]
0C	A[1,0]	A[1,1]	A[1,2]	A[1,3]
.....				
24	A[7,0]	A[7,1]	A[7,2]	A[7,3]

Se diseñan dos programas para sumar todos los elementos de la tabla, cuyo código es el siguiente:

Programa 1	Programa 2
1: int s; 2: 3: s = 0; 4: for (int i=0; i<8; ++i) 5: for (int j=0; j<4; ++j) 6: s += A[i][j];	1: int s; 2: 3: s = 0; 4: for (int j=0; j<4; ++j) 5: for (int i=0; i<8; ++i) 6: s += A[i][j];

Determinar qué programa se ejecutará más rápido.

Se dispone de un sistema de memoria de $64K \times 8$. La unidad mínima direccionable es el byte. En memoria se encuentran dos arrays de elementos de 8 bits, almacenados en las direcciones representadas por el esquema siguiente:

Array A. Dirección de comienzo: 18F0												
Contenido	12	4	67	81	231	89	106	198	206	11	39	162

Array B. Dirección de comienzo: B2D0												
Contenido	102	44	54	241	163	25	175	102	39	89	156	250

Suponiendo que el sistema de memoria incorpora una memoria caché con mapeo directo que permite almacenar 1K palabras de memoria principal y que el tamaño de bloque es de 4 palabras, representar gráficamente el contenido de la memoria caché después de 3 iteraciones del bucle de código que se presenta a continuación:

```
1: int sp;
2:
3: sp = 0;
4: for (int i=0; i<n; i+=2)
5:     sp += A[i]*B[i];
```

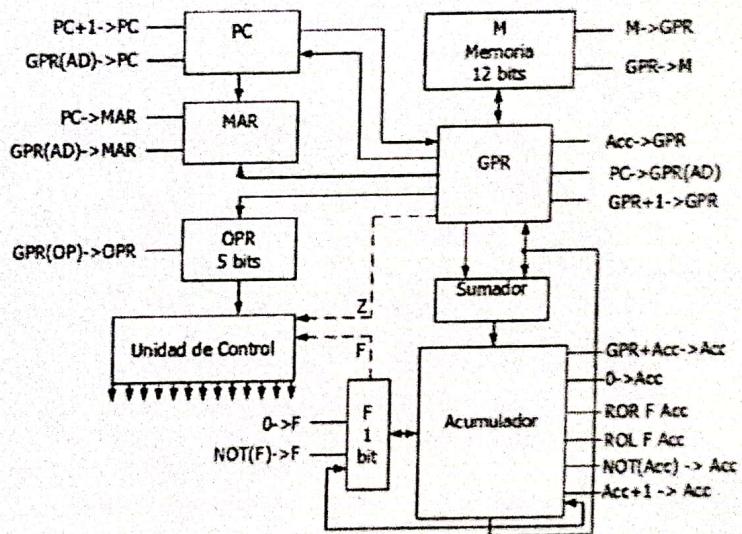
Nota: Representar únicamente las líneas de caché que han sido modificadas por el programa en ejecución.

1. La organización de la pila explicada en clase seguía un protocolo para escribir y recuperar datos en la pila de modo que SP siempre apuntaba a la última posición ocupada (de la pila). Imaginense que se quiere cambiar este protocolo para que SP pase a apuntar siempre a la primera posición libre (de la pila). Por tanto:
- Escribir el nuevo protocolo para guardar algo en el tope de pila.
 - Escribir el nuevo protocolo para recuperar algo del tope de pila.
 - Si antes se inicializaba el SP con la dirección 00(hex) (suponiendo direcciones de 8 bits), ¿con cuál dirección debería inicializarse ahora?

(1 puntos)

3. Supongamos que, en el siguiente esquema de la computadora mejorada, se modifica el registro OPR pasando de 4 a 5 bits. Indicar las siguientes cuestiones, justificando brevemente su respuesta:
- Máximo número de instrucciones en el repertorio máquina.
 - Tamaño de la memoria principal expresado en número de palabras por ancho de palabra.
 - Señalar en el esquema, junto al nombre de cada registro, el número de bits que almacena.

(1 punto)



6. Consideremos que, a la estructura de la Computadora Mejorada mostrada en la figura del ejercicio 3, se le añade:
- Contador *SC* en la unidad de control para la realización de bucles, con su bit asociado *Zsc*.
 - Registro *QR* conectado con la memoria principal y con el registro *GPR*. A su vez, está conectado con el *ACC*, de modo que se permiten desplazamientos a derecha e izquierda con el conjunto formado por *F-ACC-QR*.

Implemente la instrucción *MLTB m*, encargada de realizar la multiplicación en complemento a dos, utilizando el algoritmo de Booth, del contenido del registro *QR* por el contenido de la posición de memoria "*m*" (que deberá colocarse en el registro *GPR*). Realícelo:

- Mediante control cableado con decodificadores de tiempo y de instrucción.

Incluya también el ciclo de búsqueda.

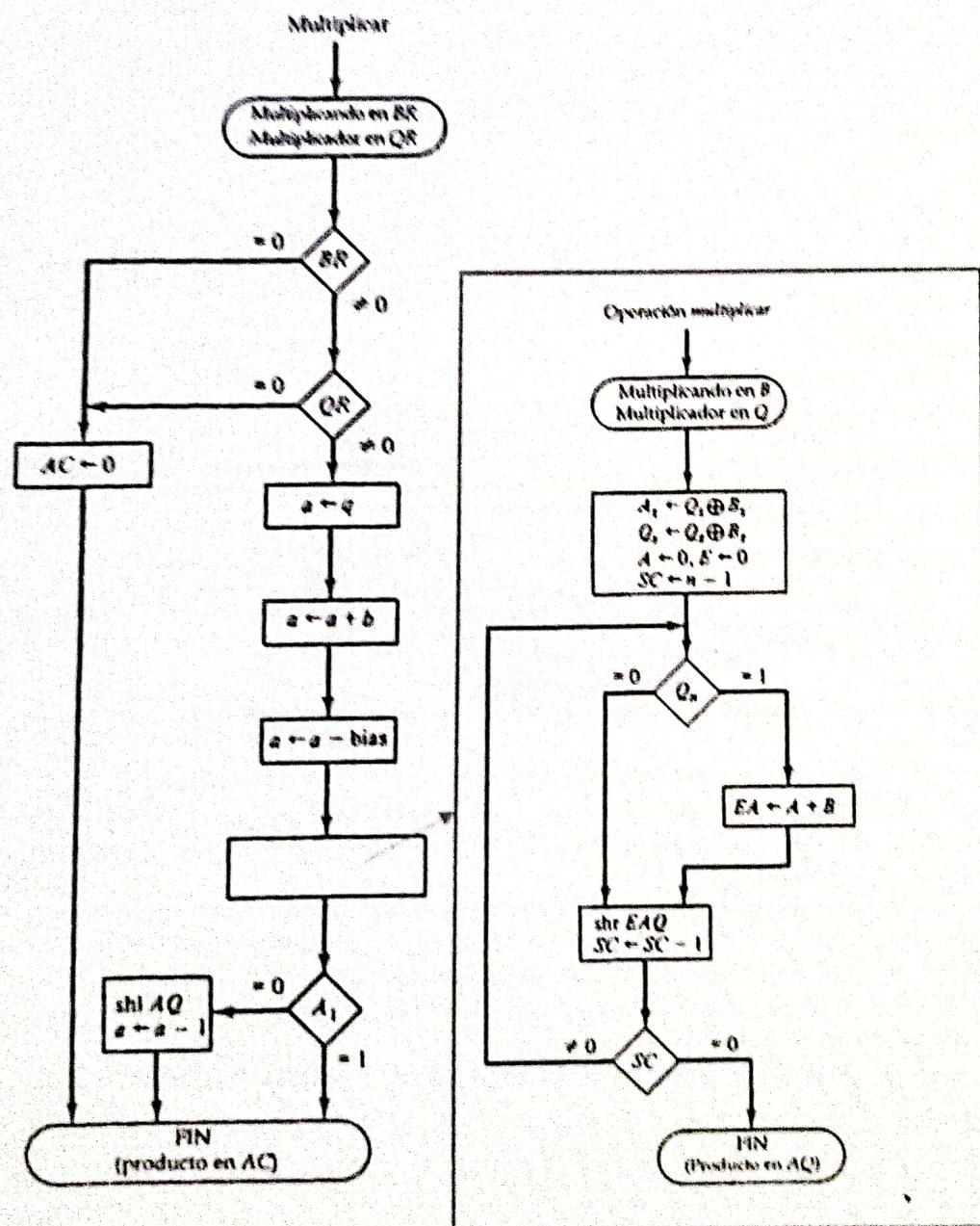
Nota: Incluya los terminales de control y bits de estado que crea necesarios para implementar la instrucción.

(2 puntos)

Ejercicio 4

Tomando como punto de partida la estructura de la Computadora Mejorada y completandola con las microoperaciones que sean necesarias implementar, mediante microprogramado con Enable, la instrucción que a continuación se relaciona. A todos los efectos, el registro GPR de la Computadora Mejorada actuará como registro B/BR en los algoritmos referidos. Incluir también el ciclo de búsqueda.

"MPF m": Realiza la multiplicación en punto flotante del contenido del registro QR por el contenido de la posición de memoria "m" (que deberá colocarse en el registro BR).



EJERCICIOS DE EXAMEN

Academia Universitaria Main()

Ejercicio 1

Implementar un programa mediante instrucciones MIPS que guarde en una variable llamada mayor(\$S2) la dirección del menor elemento de un vector A(\$S0). El vector posee n(\$S1) elementos.

Ejercicio 2

Suponer una computadora que incluye una memoria pila construida como parte de la memoria principal. Disponemos de un puntero de pila SP de 8 bits para poder direccionar hasta 256 palabras diferentes y que es inicializado a FF Hex. Si las posiciones de la pila en la memoria principal comienza en la dirección FF Hex en orden decreciente de direcciones. Definir las secuencias de microoperaciones para realizar una operación de escritura y otra de lectura en la memoria pila.

Ejercicio 3

Suponed que hacemos las siguientes modificaciones en la computadora mejorada: 12 bits para definir el formato de dato y 8 bits para el de instrucciones, el repertorio de instrucciones se reduce a 8. ¿Qué modificaciones estructurales hay que realizar en la computadora mejorada? Razona tu respuesta.

Ejercicio 4

Determinar la secuencia de microoperaciones necesarias para implementar la instrucción "C2M m" en la Computadora Mejorada tanto en control microprogramado como cableado. La instrucción "C2M m" debe realizar el complemento a dos del contenido de la posición de memoria "m" y, si el resultado es un número negativo, saltarse la siguiente instrucción. Es necesario implementar el ciclo de búsqueda.

Ejercicio 5

Se dispone de un sistema de memoria de 1Mx8. En memoria se encuentran dos arrays de elementos de 8 bits almacenados en las direcciones indicadas y con el contenido representado en el siguiente esquema:

Array A. Dirección de comienzo: 15A08												
Contenido	2	56	21	3	185	13	243	1	60	107	139	87
Array B. Dirección de comienzo: D9A08												
Contenido	10	10	10	10	10	10	10	10	10	10	10	10

El sistema de memoria incorpora una memoria caché asociativa por conjuntos de 2 vías que permite almacenar 32K palabras de memoria principal. El tamaño de bloque es de 4 palabras.

Representar el estado de la memoria caché después de 3 iteraciones del siguiente código:

```
1: for (int i=0; i<10; i+=2)
2:     B[i] = A[i] + A[i+1]
```

Ejercicio 6

A partir de la estructura de la computadora mejorada implementada en SiCoMe determinar la secuencia de microoperaciones necesarias para implementar las siguientes instrucciones. Diseñar la tabla de control de bifurcación que permita ejecutar las instrucciones. Diseñar mediante control microprogramado y cableado las instrucciones necesarias para realizar:

- a) Ciclo de búsqueda.
- b) CAL m → llamada a la rutina que está en la posición m.
- c) ABS m → calcula el valor absoluto del contenido de la dirección m y lo guarda en la misma dirección. El dato se almacena en complemento 2.
- d) PUSH_CCA m → guarda en el tope de pila el número de ceros consecutivos hasta encontrar el primer 1 más significativo.

Ejercicio 7

Implementar el siguiente programa en MIPS:

```
for (i=0; i<5; ++i) {  
    F[i] = rst2(4*i,i);  
  
}  
  
int rst2(int a1, int a2) {  
    return 2*(a1-a2);  
  
}
```

Donde F comienza en la dirección de memoria \$S0.

Ejercicio 8

Implementar un programa en MIPS:

$$f = g + A[B[h] + 1]$$

f → \$S0

g → \$S1

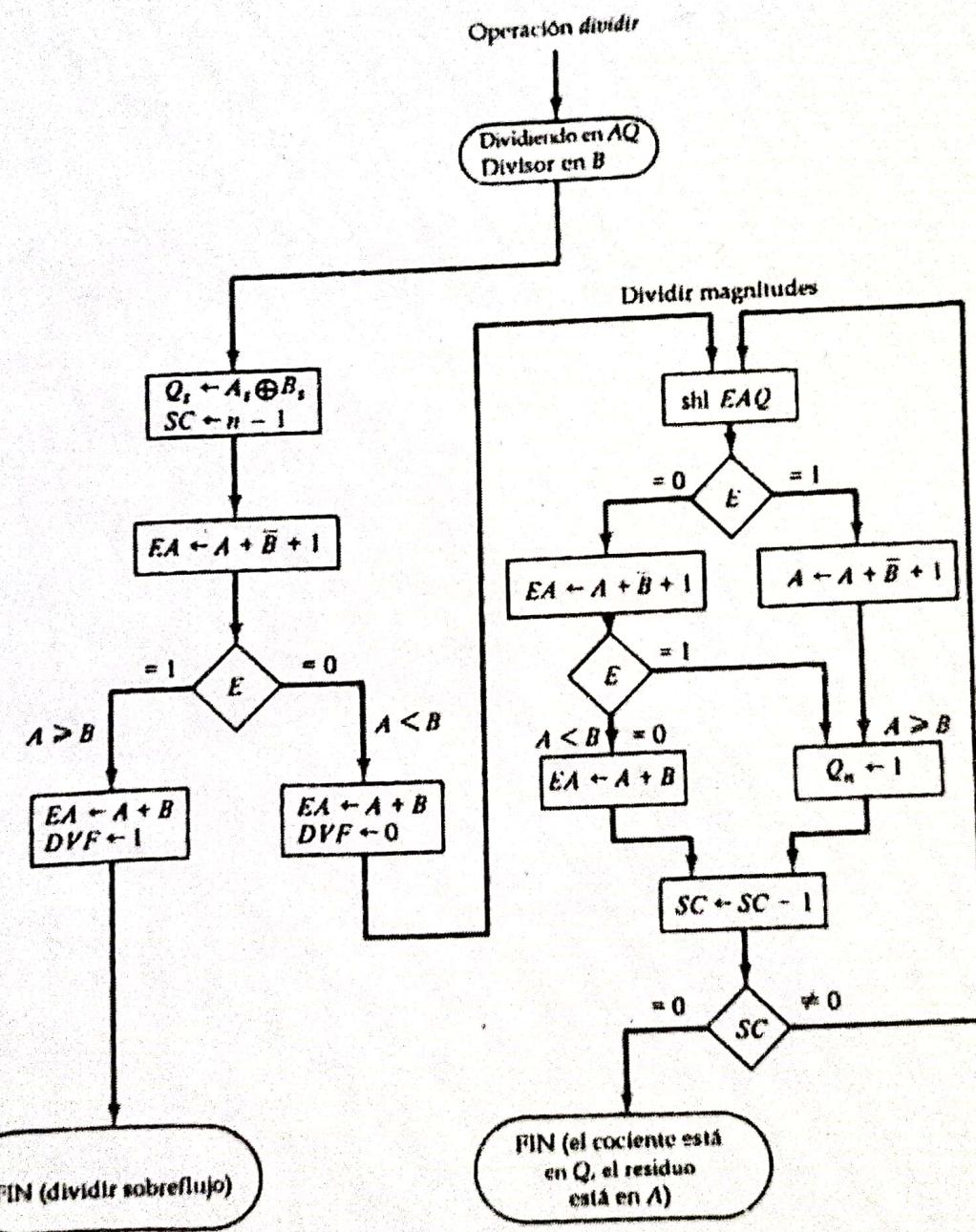
A comienza en la dirección \$S2

B comienza en la dirección \$S3

h → \$S4

División en signo-magnitud

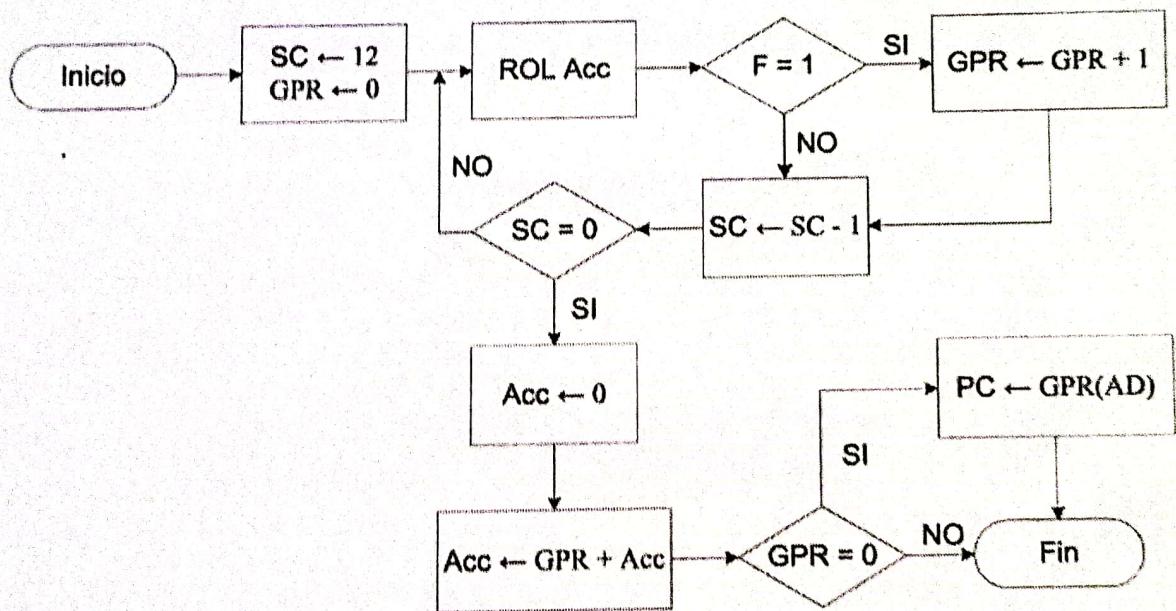
“DSM m”: Realiza la división con restauración en signo-magnitud del contenido del registro AQ entre el contenido de la posición de memoria “m” (que deberá colocarse en el registro B).



División en signo-magnitud

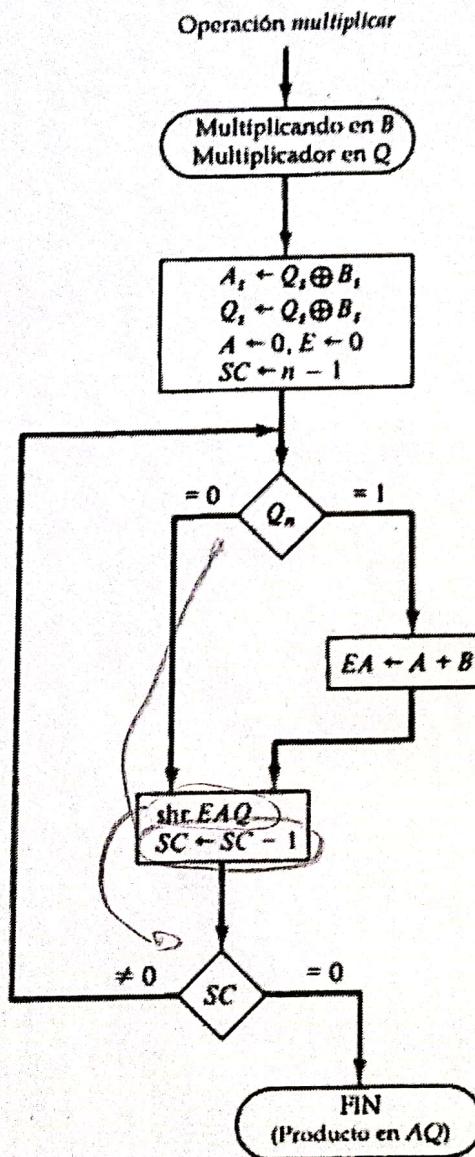
CUA

Tomando como punto de partida la estructura de la Computadora Mejorada y completándola con las microoperaciones que sean necesarias implementar, mediante control microprogramado y cableado, la instrucción CUA. Dicha instrucción reemplazará el contenido del acumulador por un número que indica la cantidad de unos que contenía (el acumulador) antes de ejecutarse la instrucción. En caso de que todos los bits del acumulador fueran cero, debe realizar además un salto a la posición de memoria 0. Incluir también el ciclo de búsqueda. El diagrama de flujo de la instrucción CUA se muestra a continuación.



Multiplicación signo-magnitud

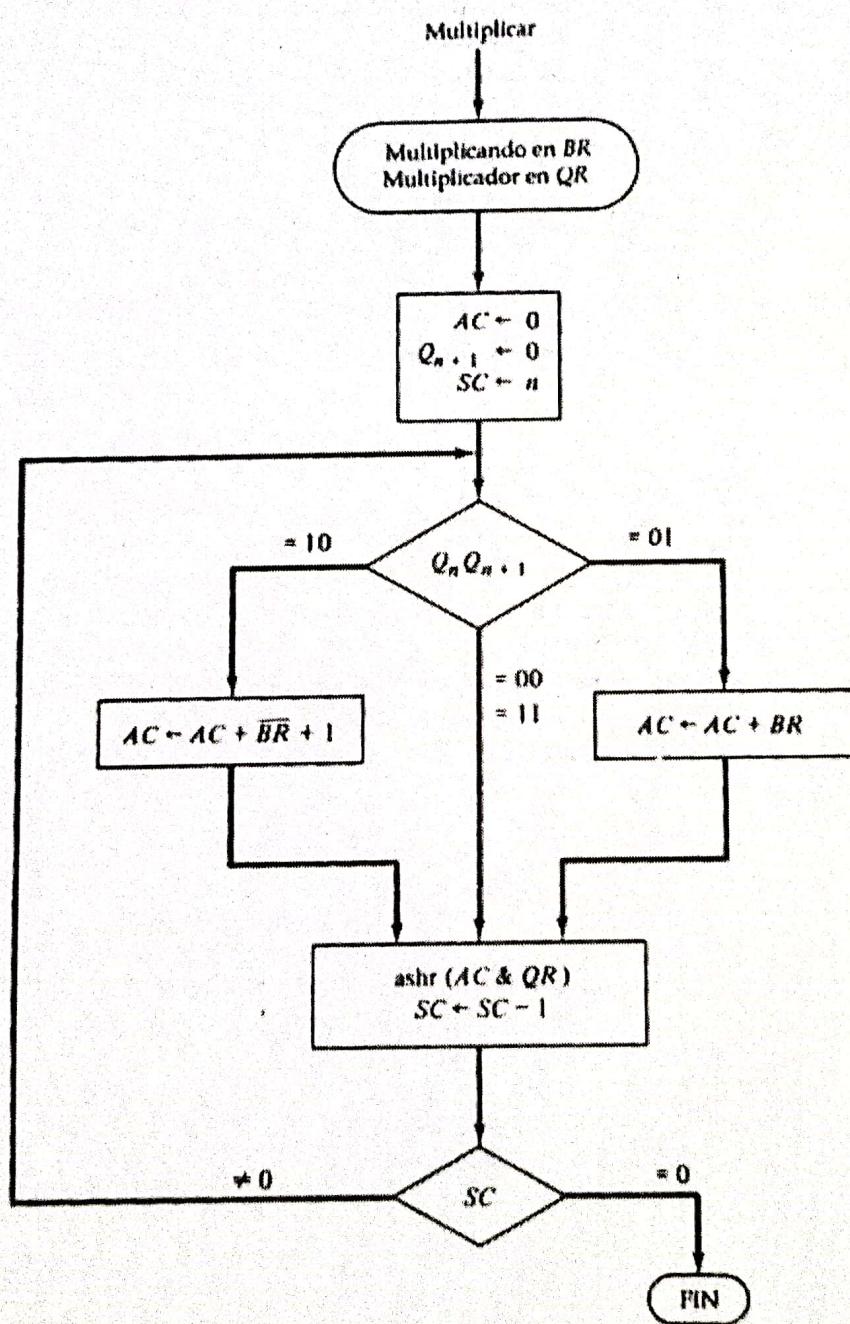
“MSM m”: Realiza la multiplicación en signo-magnitud del contenido del registro Q por el contenido de la posición de memoria “m” (que deberá colocarse en el registro B)



Multiplicación en signo-magnitud

Multiplicación en complemento a dos (Algoritmo de Booth)

“MC2 m”: Realiza la multiplicación en complemento a dos, utilizando el Algoritmo de Booth, del contenido del registro QR por el contenido de la posición de memoria “m” (que deberá colocarse en el registro BR).

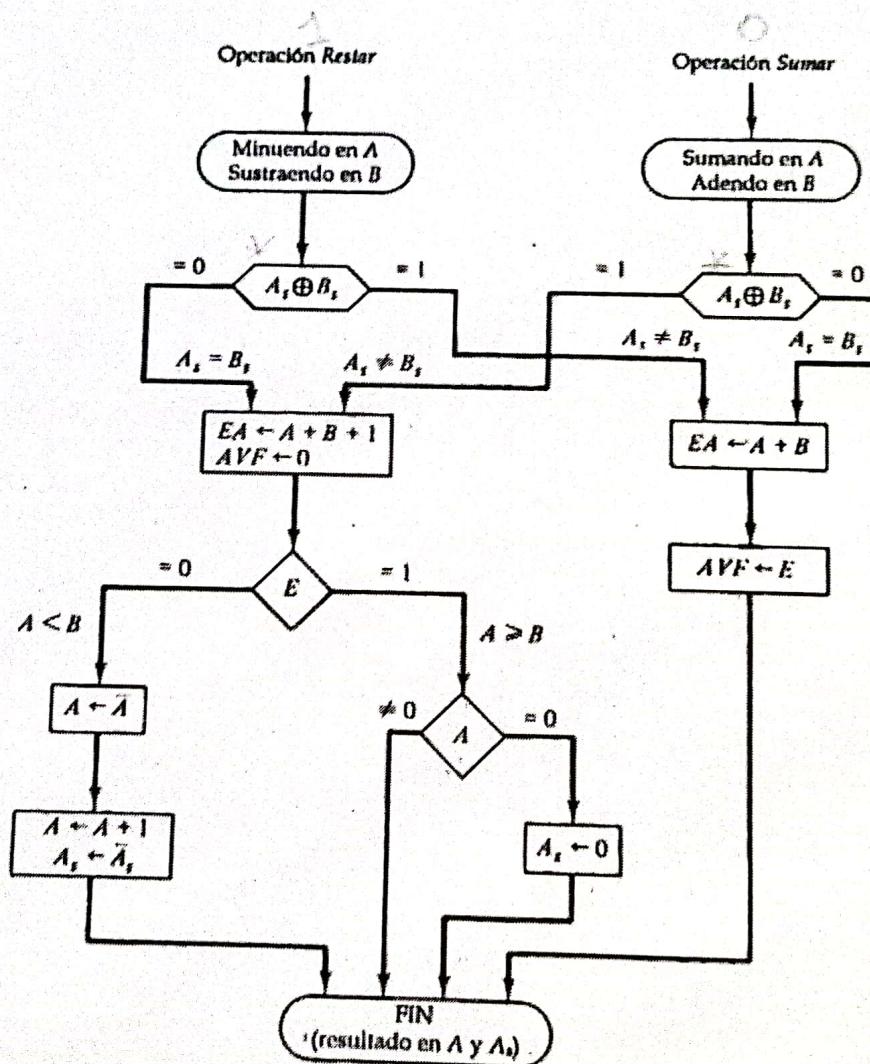


Multiplicación en complemento a dos (Algoritmo de Booth)

Suma/resta en signo-magnitud

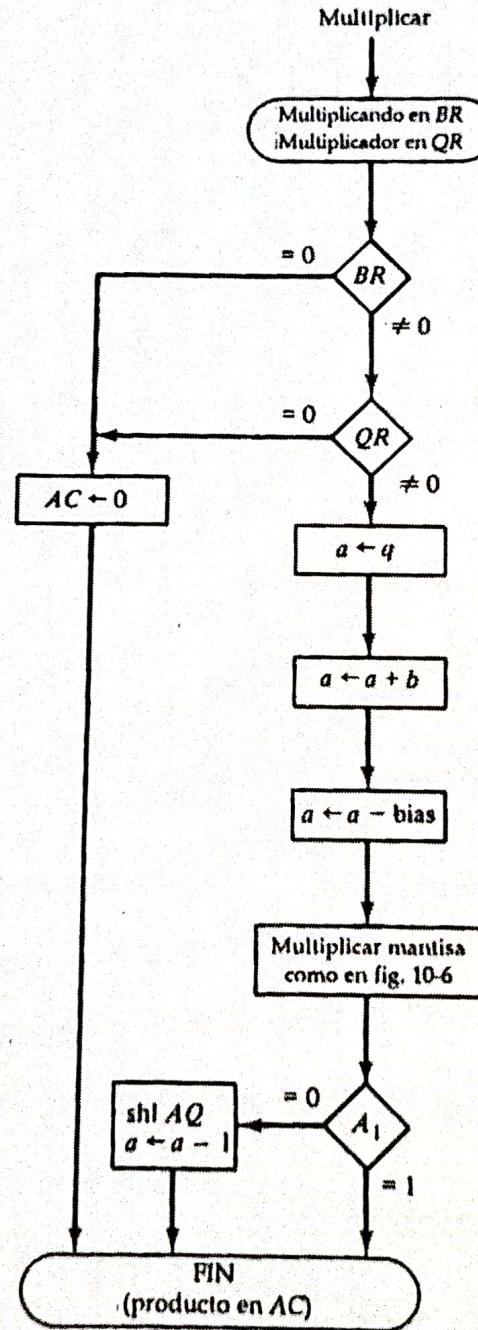
Suma
Resta

"SRSM m": Realiza la suma/resta en signo-magnitud del contenido del registro A con el contenido de la posición de memoria "m" (que deberá colocarse en el registro B).



Suma/resta en signo-magnitud

“MPF m”: Realiza la multiplicación en punto flotante del contenido del registro QR por el contenido de la posición de memoria “m” (que deberá colocarse en el registro BR).



Multiplicación en punto flotante