

Algoritmos_Explicados.pdf



TEAM_GETPPID__



Arquitectura de Computadores



2º Grado en Ingeniería Informática



**Escuela Politécnica Superior de Córdoba
Universidad de Córdoba**



[Accede al documento original](#)



Escuela de
Organización
Industrial

Contigo que evolucionas.
Contigo que lideras. Contigo que transformas.

**Esto es EOI.
Mismo propósito,
nueva energía.**



Descubre más aquí



EOI Escuela de
Organización
Industrial

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo
espacio



Necesito
concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH

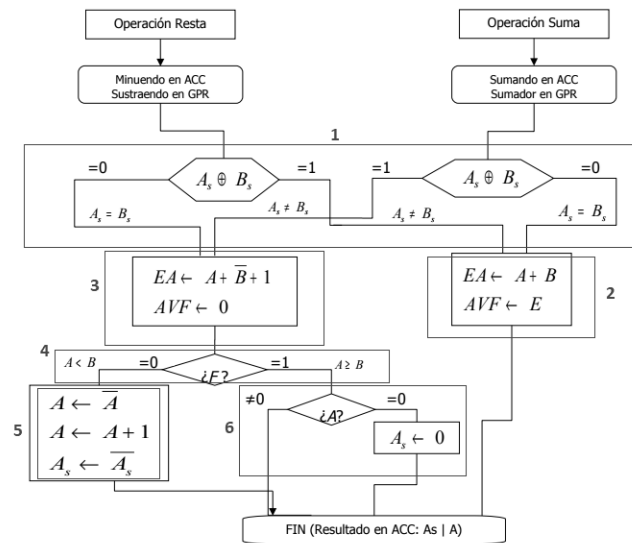


ARQUITECTURA DE COMPUTADORES ALGORITMOS.

Jose Luis Gordillo Relañó

WUOLAH

Algoritmo Suma-Resta Signo-Magnitud: SM/SR



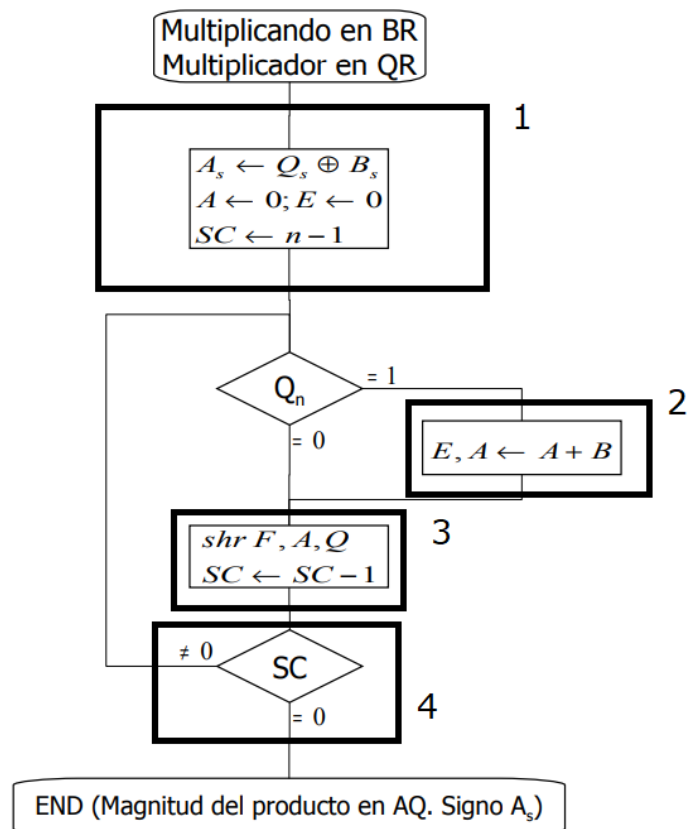
Explicación:

Necesitamos dos términos que usaremos para sumar/restar, uno estará ya situado en el Acumulador, el otro estará en el GPR.

Vamos a suponer que tenemos en Acc → +2 y en GPR → +5 y queremos realizar la suma.

- En las puertas XOR, dependiendo de la operación, realizará una u otra, en este caso activamos la de la suma, al comparar signos obtenemos que son iguales por tanto nos da un 0.
- En este caso se activa la sección 2, en el que mete en Acumulador la suma de Acc+Gpr, y en E el bit de acarreo.
Posteriormente mete en AVF (registro de sobreflujo) el bit de acarreo → Redirige a Fetch.
- Suponiendo que son de signo distinto Acc → -2 y GPR → +5 y los sumamos obtendríamos en la puerta XOR un 1, por tanto, entramos a la sección 3.
 $-2 = 1010$ $+5 = 0101$
 $EA \leftarrow 010 (2) + 101 (5) \rightarrow 5-2 = 101-010 = 011$ con signo (-) → 0011
 A su vez guardamos el signo de la operación en el bit F(E).
 Guarda en el bit E el acarreo, y mete en el registro de sobreflujo un 0, ya que en **restas nunca tendremos sobreflujo**.
- Pasamos a comprobar el bit del Acumulador **F**, el cual nos dirá el signo del resultado, como resultado nos dará 0 si el Acc es menor que GPR.
En este caso el signo será positivo 0.
- En este registro nos indica que hemos hecho la operación inversa a la deseada, por tanto, únicamente debemos complementar y cambiar el signo de la operación.
 $A \leftarrow 5 = 101 \rightarrow 010 + 1 = 011 (3)$
 $A_s \leftarrow$ guardamos el signo (+).
- En caso contrario que el Acc sea mayor que Gpr, comprobamos si el resultado de la resta es 0, obligamos a que el signo del 0 sea positivo.
Si no fuesen iguales, vamos al Fetch.

Multiplicación SM:



Algoritmo que realizará la multiplicación de dos números (sumas sucesivas) en signo-magnitud.

Para ello partimos con que el multiplicando esta en GPR y el multiplicador estará en el QR.

Comprobaremos en Q_n (bit menos significativo del QR):

-Si vale 1: Sumaremos las magnitudes $A+B$ (PASO 2)

-Si vale 0: Ejecutaremos el Paso 3.

El bucle terminará desplazando un bit a la derecha el conjunto de E, A y Q.

Tras $n-1$ iteraciones (**recordemos 1 bit de signo, de ahí obtenemos $n-1$**), el resultado estará en ACC.

PASO 1: $A_s \leftarrow Q_s \text{ XOR } B_s$ (XOR para obtener el signo del ACC).

Inicializamos la cuenta del número de bits a $N-1$.

Ponemos el ACC y la F (E) a cero.

PASO 2: Si Q_n vale 1, realizará la suma de las magnitudes (**SOLO MAGNITUD**)

PASO 3: Si Q_n vale 0, realizará el desplazamiento completo del ACC, así como restará una iteración.

PASO 4: Toma de decisión para saber cuando ha finalizado el número de iteraciones.

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo espacio

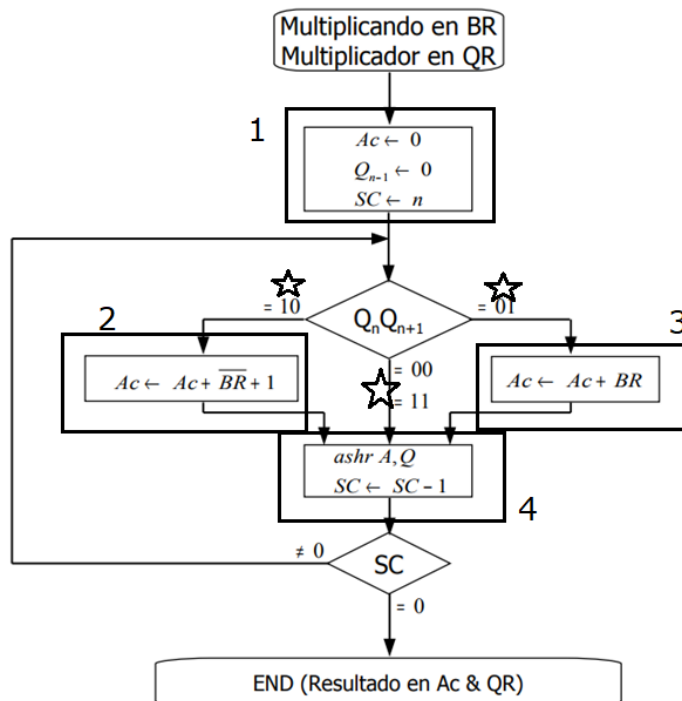


Necesito concentración

ali ali oohh
esto con 1 coin me
lo quito yo...

WUOLAH

Multiplicación Booth:



Algoritmo basado en búsqueda de cadenas de unos.

PASO 1: Meto un 0 en Acumulador y en Q_{n+1} .

Inicio la cuenta de SC igual a N.

PASO 2: Realizamos la resta del Acc con el GPR y lo dejamos en Acc.

PASO 3: Sumamos los contenidos del GPR + Acc

PASO 4: Desplazamiento aritmético del acumulador y del QR.

Restamos 1 a la cuenta de SC.

10 → Si Q_n vale 0 y Q_{n+1} vale 1, estoy en el inicio de una cadena de 1.

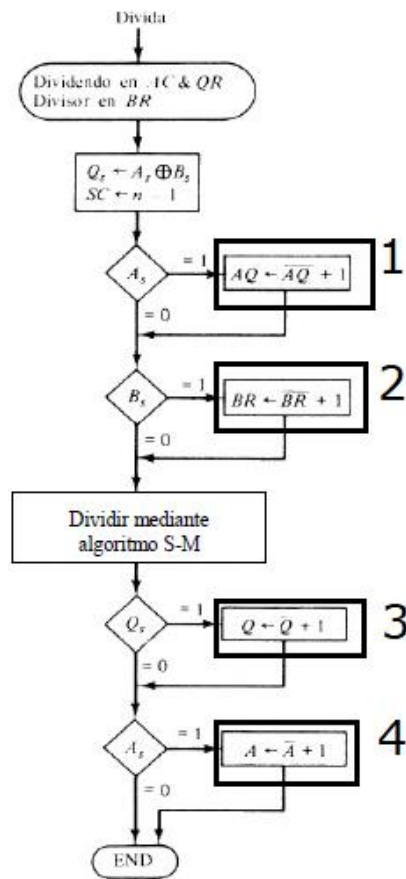
01 → Si Q_n vale 1 y Q_{n+1} vale 0, nos indica que hemos alcanzado el fin de la cadena de bits (cadena de 1s).

00 → Indica que no se ha encontrado aun la cadena, por tanto, sigue realizando desplazamientos.

11 → Indica que continua la iteración de la cadena de bits, por tanto, sigue desplazando.

WUOLAH

Algoritmo División Complemento a 2:



ACLARACIONES PREVIAS:

El paso 1 y el paso 2, se realizarán para convertir los dos números en positivos.

El paso 3 y paso 4, se realizará la división mediante el algoritmo de SM, si el resultado es negativo, lo complementamos.

PASO PREVIO:

Metemos en Qs el signo resultante del Signo del divisor y del dividendo.

Metemos en SC el número de bits(MAGNITUD) -1 (SIGNO).

PASO 1: Comprobamos el signo del dividendo, si es negativo, le cambio el signo al número global.

PASO 2: Compruebo el signo del divisor, si es negativo le cambio el signo al número global.

PASO INTERMEDIO: Ahora únicamente con los números positivos realizamos la división de Signo Magnitud.

PASO 3: Si el cociente tenía que ser negativo, le cambio el signo.

PASO 4: Compruebo el signo que tenía inicialmente el dividendo, y se lo cambio en caso de ser negativo.

Imagínate aprobando el examen

Necesitas tiempo y concentración

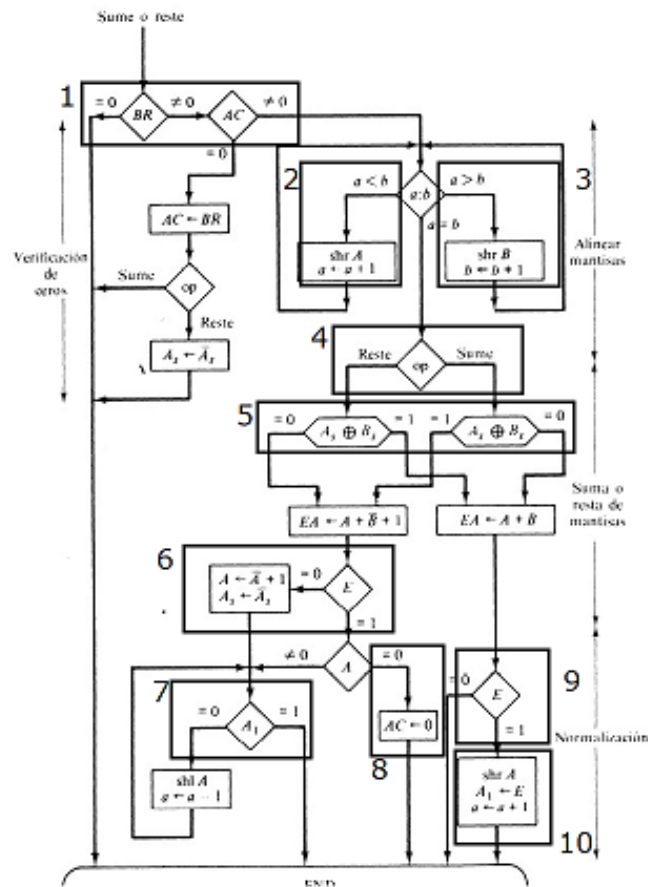
Planes	 PLAN TURBO	 PLAN PRO	 PLAN PRO+
 Descargas sin publi al mes	10 	40 	80 
 Elimina el video entre descargas			
 Descarga carpetas			
 Descarga archivos grandes			
 Visualiza apuntes online sin publi			
 Elimina toda la publi web			
 Precios Anual <input type="checkbox"/>	0,99 € / mes	3,99 € / mes	7,99 € / mes

Ahora que puedes conseguirlo,
¿Qué nota vas a sacar?



WUOLAH

Suma-Resta en Punto flotante:



La parte superior izquierda corresponde a comprobar que no se divida por 0 ni entre 0

PASO 1: Comprobamos si BR y AC es cero, en caso de que alguno sea cero, realiza la suma o resta y finaliza la instrucción.

PASO 2: Si el exponente de ACC es menor que el de BR, desplazamos a la derecha el Acc e incremento exponente, se compara hasta que sean iguales los exponentes.

PASO 3: Si el exponente de ACC es mayor que el de BR, desplazamos a la derecha el ACC e incremento el exponente, se compara hasta que sean iguales los exponentes.

PASO 4: Dependiendo del código de operación realizaremos la suma o la resta.

PASO 5: Realizamos una XOR de signos entre los operandos.

PASO 6: Si F vale 0, indica que la mantisa de B era mayor, por tanto, la invierto e incremento, así como le cambio el signo.

PASO 7: Comprobamos si el número está o no normalizado, comprobando el bit más significativo de la mantisa.

-Si vale 1 = finalizamos.

-Si vale 0 = rotamos a la izquierda y decrementamos los exponentes.

Es decir, pasaríamos de 0,000001 a 0,1e-5

PASO 8: Si la mantisa vale cero, pondremos todo el registro a cero.

PASO 9: Realizamos la comprobación de F, ya que puede existir desbordamiento de suma.

PASO 10: Si F=1, tenemos un 1 a la izquierda de la coma, por tanto, desplazamos a la derecha e incrementamos.

Es decir: 1,0e-5 \rightarrow 0,1e-4

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins?

Plan Turbo: barato

Planes pro: más coins

pierdo espacio

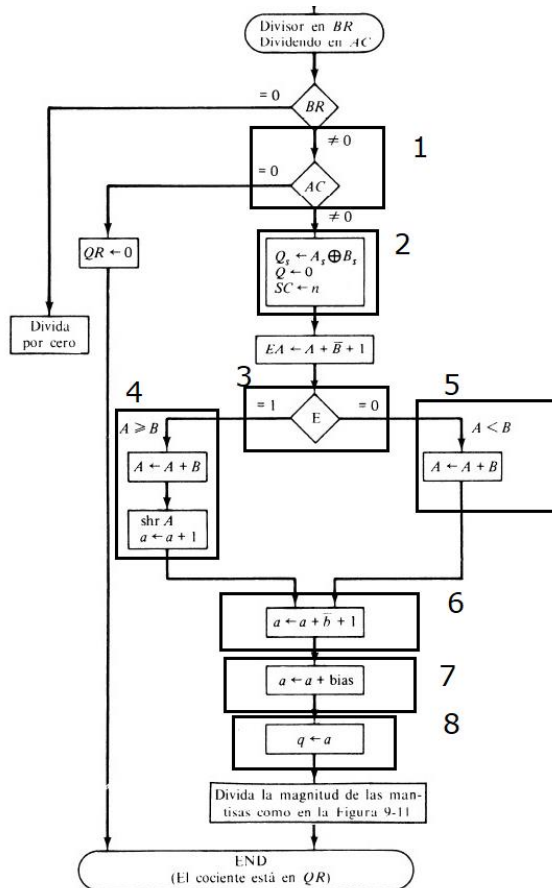


Necesito concentración

ali ali oohh
esto con 1 coin me
lo quito yo...

WUOLAH

División en Punto flotante:



PASO 1: Comprobaciones previas de división (dividir por cero, o dividir al cero).

PASO 2: Realizamos XOR de signos, guardamos un 0 en Q, e iniciamos el contador a N.

PASO 3: Comprobamos si el divisor entra en el dividendo, para ello restamos y lo comprobamos.

PASO 4: Movemos a la derecha el dividendo y ya podremos realizar la división, así como aumentar el exponente en 1 ($a \leftarrow a + 1$), cuando $E=1$.

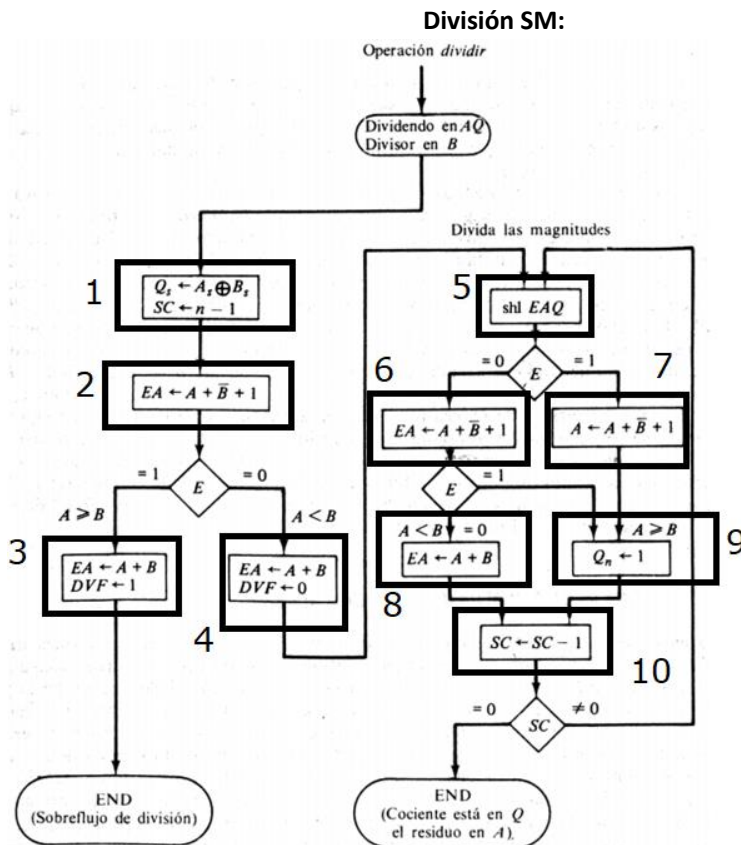
PASO 5: Si $E=0$, indica que el divisor no cabe.

PASO 6: Realizamos la resta de exponentes.

PASO 7: Realizamos la suma del sesgo y exponente.

PASO 8: Movemos el exponente a Q y realizamos la división mediante algoritmo SM.

WUOLAH



PASO 1: Partimos con el divisor en GPR y el dividendo en el ACC.

Partimos con repetir N-1 veces el bucle asignado en $SC \leftarrow N-1$.

Determinamos si los signos del divisor y dividendo son iguales o distintos con la XOR.

PASO 2: Realizamos la resta mediante complemento a 2, si el resto es negativo, la división se ha completado (PASO 3).

PASO 3: Se restaura el valor del Resto sumándole el

PASO 4: Si no cabe, desplazamos el cociente un bit a la izquierda, poniendo su bit menos significativo a 0.

PASO 5: Rotamos el dividendo y comprobamos el signo de F, si vale cero (PASO 6), si vale 1 (PASO 7).

PASO 6: Se comprueba mediante resta si el numero cabe en el divisor, para ello posteriormente comprueba el bit F y determina si puede dividir (PASO 9) o no puede dividir y requiere de hacer otra rotación (previa restauración, paso 8).

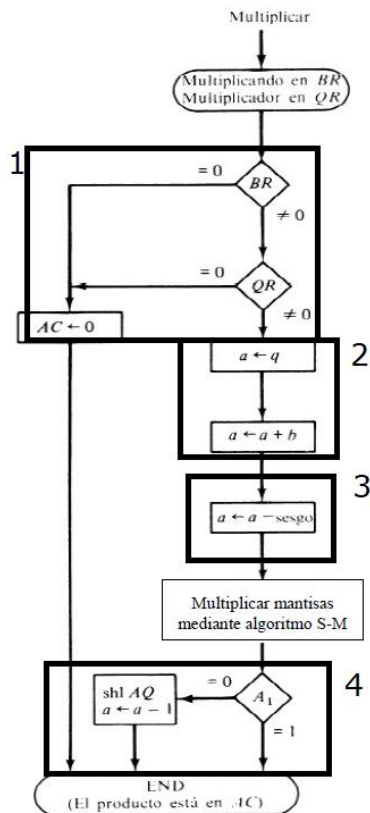
PASO 7: Realizamos la resta en Complemento a 2 y dejamos el resultado en Acc.

PASO 8: Restauramos el número del acumulador ya que no ha sido posible dividir.

PASO 9: Metemos un 1 en el bit menos significativo de QR (para al finalizar tener el cociente en QR)

PASO 10: Decrementamos SC para hacer el bucle y rotar todos los bits.

Producto Punto Flotante:



Paso 1: Comprobamos si algún registro (GPR/QR) vale cero, si valen cero, ponemos un cero en el acumulador y finalizamos.

Paso 2: Pasamos el exponente del QR al ACC y sumamos los exponentes del ACC y del GPR.

Paso 3: Restamos el sesgo al exponente del acumulador.

Paso 4: Comprobamos el bit mas significativo del ACC, si vale 1 finalizamos, pero si vale cero, rotamos hacia la izquierda para normalizar el número.