1.- Sistemas de representación

1.1 **(a)** Indicar el vector de pesos en un sistema posicional para representación de números naturales, en base $r = 16$ y cuatro dígitos, $n = 4$. Dar la respuesta en base 10.

(b) Sea un sistema de representación de números naturales con $n = 4$ y $r = 2$. ¿Cuáles son el mayor y el menor número representables? ¿Se puede representar el número +16?

(c) Indicar el valor explícito utilizado para representar los siguientes números en complemento a 2, considerando 8 bits para la representación: –45, +53, –6 ¿Y si se utiliza complemento a 1? ¿Y por exceso?

(d) Representar los números +41,15 y –6,21 en signo-magnitud, complemento a la base y complemento restringido a la base, con $r = 2$, $n = 8$ y $k = 4$. ¿Es exacta la representación? En caso negativo, ¿cuál es el error cometido?

(e) Dado un sistema posicional de base 7, calcular el complemento a la base del número 4563_7.

(f) Representa el número –136, con $r = 2$ y $n = 10$, en: (a) signo-magnitud, (b) complemento a la base, (c) complemento restringido a la base y (d) exceso. Efectúa la operación cambio de signo en los casos anteriores.

1.2 Representar el número negativo –1302,125 en base 4 y utilizando el mínimo número de dígitos necesario en:

- Signo-magnitud
- Complemento a la base
- Complemento restringido a la base
- Exceso

1.3 **(a)** Obtener el valor del número representado por los siguientes vectores de dígitos:

a.1) En signo-magnitud	1000 1011
a.2) En complemento a 2	1101 0111
a.3) En complemento a 1	1001 0101
a.4) En exceso	0001 0111

Indicar el vector de dígitos de cada número en las otras representaciones.

(b) Dado el vector de dígitos $X = (1,0,0,1,1,1)_r$

b.1) Calcular el valor explícito, X_e, para los casos de $r = 2$, 4, 8 y 10.

b.2) ¿Cuál es el valor implícito del número, siendo $r = 2$, en signo-magnitud, complemento a 2 y complemento a 1?

b.3) ¿Cuál es el mayor valor explícito X_e que puede representarse mediante un vector de 6 dígitos si $r = 2$, 10 y 16?

1.4 Completar la siguiente tabla, en los cuatro casos que se indican:

	Valor implícito	Valor explícito	Vector de dígitos
(a)	-37_{10}		
(b)		205_{10}	
(c)			11011
(d)	+9		

a) $r = 4$, $n = 6$, complemento a la base
b) $r = 2$, $n = 8$, complemento restringido a la base
c) $r = 2$, $n = 5$, exceso
d) $r = 2$, $n = 5$, signo-magnitud

1.5 Dado el número real –68,356, representarlo en coma fija con $r = 2$ y 12 bits, 8 para la parte entera y 4 para la fraccionaria, en los siguientes sistemas:

- Signo magnitud
- Complemento a la base
- Complemento restringido a la base.

¿Es exacta la representación? Si no lo es, ¿cuál es el error cometido?

2.- Suma y resta de números enteros

2.1 Se realizan las siguientes operaciones (sumas y restas) en un sumador de propagación de la llevada (RCA, CPA) para números naturales:

a) 100110 + 011101
b) 100101 + 101010
c) 01110100 – 01011010
d) 00010110 – 11000110

Considerando los retardos ya vistos, ¿qué tiempo se necesita en cada operación para obtener un resultado estable? ¿cuánto tardará la operación en cada caso?

2.2 Los números $X = 0111$ e $Y = 1011$ están representados en complemento a 1. Realizar la suma de los mismos en dicha representación.

2.3 **(a)** Considerando el sumador-restador del apartado 2.7 de los apuntes, y suponiendo que los datos están ya cargados en los registros de operandos, indicar las funciones a realizar y las señales de control necesarias para efectuar las siguientes operaciones:

Operación		Función	Señales de control
SUMA	$s = x + y$	S:=ADD(X,Y,0)	G1, LdF, LdE
RESTA	$s = x - y$		
SUMA con llevada	$s = x + y + C0$		
INCREMENTO	$s = x + 1$		
COMPARACIÓN	(x,y)		

(b) Sean X = 0110011011 e Y = 1110100111, dos números representados en complemento a 2. Indicar el contenido de los registros de datos e indicadores al realizarse las siguientes operaciones: (b.1) $X + Y$; (b.2) $X - Y$.

¿Cuál es el periodo mínimo de reloj que puede utilizarse si Δ = 5 ns y el retardo de los multiplexores es 10 ns? No tomar en consideración el tiempo de carga de los registros.

Para el caso de estos dos números, indicar la longitud de la máxima propagación de la llevada. ¿A partir de qué momento es estable el resultado?

2.4 En un sumador de propagación de la llevada (RCA, CPA) se suman los dos números naturales siguientes:

$$\begin{array}{r} 0\,0\,1\,0\,1\,0\,1\,1 \\ 0\,0\,1\,1\,0\,1\,0\,1 \\ \hline 0\,1\,1\,0\,0\,0\,0\,0 \end{array}$$

Rellena la tabla siguiente indicando los resultados parciales que se obtienen cada 2Δ. Para este caso particular, ¿cuál es el retardo del sumador?

	S7 S6 S5 S4 S3 S2 S1 S0	**C8 C7 C6 C5 C4 C3 C2 C1 C0**
2Δ	0 0 0 1 1 1 1 0	0 0 1 0 0 0 0 1 **0**
4Δ		
6Δ		
8Δ		
10Δ		
12Δ		
14Δ		
16Δ		

2.5 Sean los números A=10110011 y B=10100101. Realiza la operación A+B para los siguientes casos:
(a) considerando que se trata de números naturales;
(b) considerándolos enteros en complemento a 2.
¿Ha habido desbordamiento? ¿Por qué?
Repite lo anterior para $A - B$.

2.6 Realiza las siguientes operaciones en signo magnitud:

(a) 11001010 + 0010111 y (b) 11001110 – 1011000

2.7 Se realiza la siguiente suma en un sumador de propagación de la llevada (RCA, CPA) para números naturales:

$$\begin{array}{r} 0\,1\,1\,1\,1\,0\,1\,1 \\ 0\,0\,1\,1\,0\,1\,0\,1 \\ \hline 1\,0\,1\,1\,0\,0\,0\,0 \end{array}$$

Rellenar la tabla siguiente con los resultados parciales que se van generando cada 2Δ (retardo del sumador de un bit). Deducir así el retardo final en este caso particular.

	S7 S6 S5 S4 S3 S2 S1 S0	C8 C7 C6 C5 C4 C3 C2 C1 C0
2Δ	0 1 0 0 1 1 1 0	0 0 1 1 0 0 0 1 0
4Δ		
6Δ		
8Δ		
10Δ		
12Δ		
14Δ		
16Δ		

3.- Sumadores rápidos

3.1 Queremos sumar los dos números siguientes: $N1 = 2954_H$ y $N2 = 9F78_H$.

(a) Dibuja un sumador de 16 bits basándote en bloques de 4 bits (FA o CLA)

(b) Indica los valores de las funciones G y P en todos los niveles y el valor de la suma.

(c) ¿Cuánto tiempo se necesita para obtener la suma si los retardos de los circuitos CLA son los vistos en clase? ¿Qué cambia si los circuitos CLA son de 2 bits?

3.2 La figura adjunta representa un sumador de dos números de 4 bits:

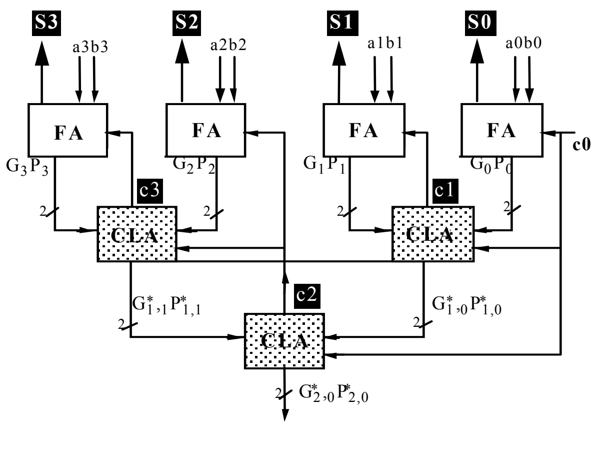
a) ¿A qué tipo de sumador rápido corresponde dicho esquema?

b) Explica brevemente cómo se realiza la suma, esto es:

- Qué operación realizan los dos tipos de bloques representados: FA y CLA.
- Cuál es el significado de cada una de sus señales de salida (funciones Si, Gi, Pi, Gi*, Pi*, Ci).
- Calcula, en consecuencia, el máximo retardo asociado a este sumador.

c) Realiza la suma para el caso particular siguiente: A = 0110 y B = 0111. Detalla el valor de las funciones intermedias.

d) Propón un esquema alternativo usando una estructura tipo *Carry Select Adder* en base a módulos de dos bits. ¿El retardo máximo es mayor o menor que en el caso anterior?



3.3 En un sumador "*Carry-Select*" de 20 bits, se han utilizado bloques de 4 bits para obtener la suma.

(a) Dibuja el esquema hardware de ese sumador.

(b) Si se suman los siguientes números $N1 = 3895A_H$ y $N2 = 7BF43_H$, indica todos los resultados parciales y la suma final (indícalos en la figura que hayas realizado). Calcula el tiempo de ejecución así como el de peor caso.

(c) El tiempo de respuesta de estos sumadores se puede optimizar si el número de bits de los bloques es diferente. Por ejemplo, repite los dos apartados anteriores utilizando los siguientes tamaños de bloque: 6 - 5 - 4 - 3 - 2 (el de 6 bits para los bits de mayor peso). Intenta explicar por qué se obtiene ahora un tiempo de respuesta menor para el peor caso.

3.4 En un sumador "*Carry-Skip*" de 20 bits, se han utilizado bloques de 4 bits para obtener la suma.

(a) Dibuja el esquema hardware de ese sumador.

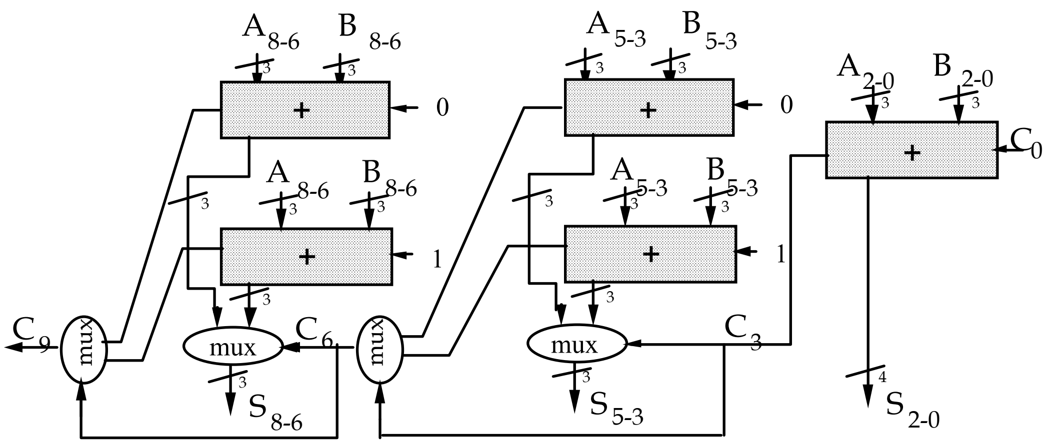
(b) Si se suman los siguientes números $N1 = E34A1_H$ y $N2 = 2CE5F_H$, indica todos los resultados parciales y la suma final (indícalos en la figura que hayas realizado). Calcula el tiempo de ejecución así como el de peor caso.

(c) El tiempo de respuesta de estos sumadores se puede optimizar si el número de bits de los bloques es diferente. Por ejemplo, repite los dos apartados anteriores utilizando los siguientes tamaños de bloque: 2 - 5 - 6 - 5 - 2. Intenta explicar por qué se obtiene ahora un tiempo de respuesta menor para el peor caso.

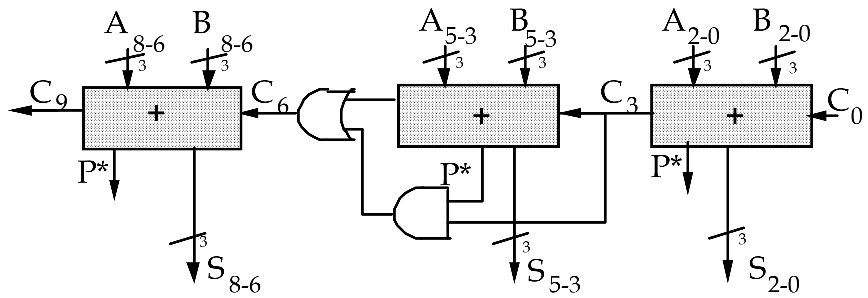
3.5 Dibuja el esquema de un sumador *carry-skip* para números de 18 bits con sumadores serie de 6 bits. Indica cuál es el tiempo que tardará en calcularse el resultado en el peor de los casos de acuerdo a los retardos habitualmente utilizados. Explica la respuesta.

3.6 Dados los siguientes esquemas de sumadores para operar con números de 9 bits, construidos a partir de módulos de 3 bits ¿cuál es el retardo de cada uno de ellos considerando el peor de los casos? Razona tu respuesta.

Sumador 1



Sumador 2

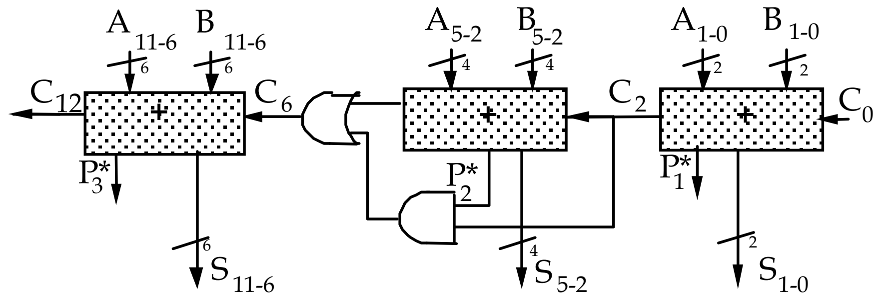


3.7 Contamos con el sumador de la figura siguiente para números de 12 bits. Según puedes apreciar el esquema corresponde a un sumador tipo *carry-skip*, en el que los bloques RCA utilizados no son todos iguales: el RCA de menor peso es de 2 bits, el siguiente es de 4 bits y el último es de 6 bits.

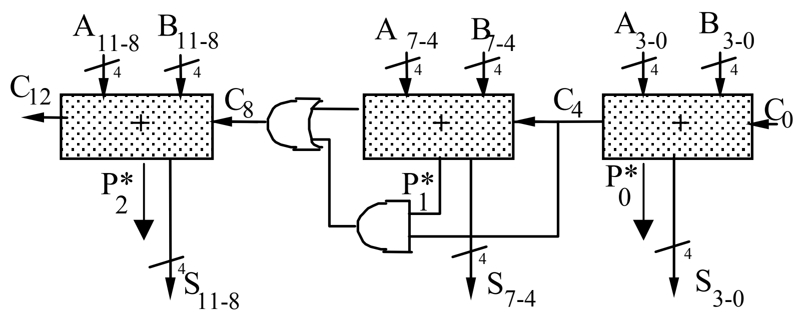
a) Explica cómo se calcula la función P_2* y cuál es su significado.

b) Tenemos que sumar los números: 9B3 y E4D. Explica de acuerdo a este ejemplo el valor que tomarán las salidas C2, C6, P_2* y $S_{11\text{-}6}$ y di cuándo serán estables. ¿Se trata de un ejemplo del peor caso posible? Razona tu respuesta.

c) Explica el tiempo de respuesta máximo que corresponde a este sumador.



3.8 Dado el sumador *Carry-Skip* de 12 bits visto en clase cuyo esquema tienes a continuación y tomando como datos de entrada A=401H y B=EFFH



a) Completa la tabla adjunta indicando claramente los cambios que se producen en las salidas del sumador cada 2Δ. Suponemos todas las señales inicializadas a cero. Explica **de manera concisa** el motivo de los cambios que se van produciendo en las salidas. ¿Cuánto tiempo tarda en estabilizarse el resultado?

b) ¿Cuánto tiempo tardará este sumador en sumar dos números en el peor caso? Pon un ejemplo de dicho caso justificándolo razonadamente.

	C_{12}	P^*_2	S_{11-8}	C_8	P^*_1	S_{7-4}	C_4	P^*_0	S_{3-0}
0Δ	0	0	0 0 0 0	0	0	0 0 0 0	0	0	0 0 0 0
2Δ									
4Δ									
6Δ									
8Δ									
10Δ									
12Δ									
14Δ									
16Δ									
18Δ									
20Δ									
22Δ									

3.9 Se quieren sumar tres números naturales de 8 bits: N1, N2 y N3 y obtener su resultado también en 8 bits. Se dispone para ello de módulos sumadores de 4 bits de tipo RCA y CSA. Basándote en estos módulos dibuja para cada una de las estructuras siguientes el esquema del sumador y justifica su retardo:

1. Estructura tipo Carry-Select Adder.
2. Estructura tipo Carry-Save Adder (CSA). En este caso indica los resultados parciales y la suma final del esquema propuesto en el siguiente caso particular: N1=0001 1101, N2=0010 1010, y N3=0111 1100.

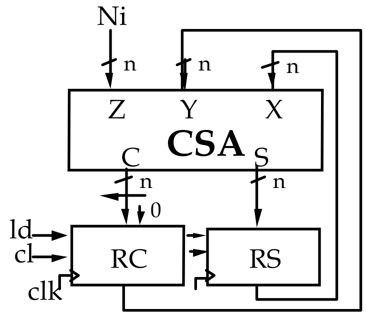
3.10 Se quieren sumar tres números de 16 bits. Disponemos de sumadores RCA y CSA de 4 bits, sumadores CLA de 8 bits (con circuitos CLA internos de 2 bits) y cicuitos CLA de 2 bits. Indica, para los casos siguientes, cuál será el tiempo de respuesta y cuántos bloques se utilizarán:

a) utilizando sólo bloques RCA.
b) sumadores CLA en niveles unidos por circuitos CLA.
c) utilizando CSA y RCA.

3.11 Dado el sumador CSA de la figura, realizar la suma de:

A = 3 B = 1
C = 2 D = 5
E = 6

si n = 5 y r = 2.
¿Hay desbordamiento?



3.12 Construir un árbol de Wallace con máximo paralelismo y sumar los siguientes números (sólo magnitud) calculando los resultados parciales en cada CSA:

N1 = 000110 (6)
N2 = 000001 (1)
N3 = 001010 (10)
N4 = 010100 (20)
N5 = 001000 (8)
N6 = 000101 (5)
N7 = 001001 (9)

3.13 Se ha construido un árbol de Wallace para sumar 20 números de 16 bits. La última suma se realiza con un sumador en serie.

(a) Calcula el tiempo de respuesta de dicha estructura.

(b) ¿Cuántas sumas se pueden realizar por segundo si entre dos sumas consecutivas de 20 números hay que esperar 15 ns y $\Delta = 4$?

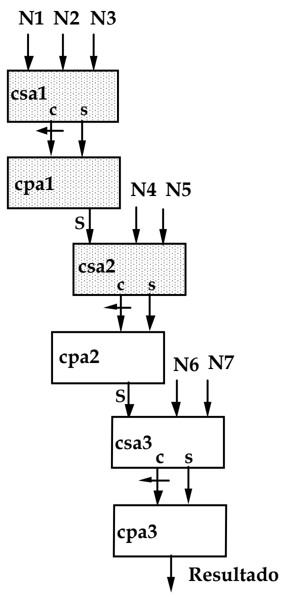
(c) El sumador se ha realizado con bloques básicos (CSA y RCA) de 4 bits. ¿Cuál es el coste del sumador, si el coste de cada bloque es x?

Repite el ejercicio para los dos casos siguientes: (1) sumador CSA de un único nivel, y (2) sumador de 3 niveles de CSA.
(el retardo de los multiplexores y la carga de los registros es de 4 ns).

3.14 Construye un árbol de Wallace para poder sumar en paralelo 6 números naturales de 5 bits. A continuación, indica los resultados parciales a lo largo del árbol, incluido el último de ellos, para el siguiente caso: N1 = 00011, N2 = 00010, N3 = 01100, N4 = 10000, N5 = 10111 y N6 = 00100.
¿Se produce desbordamiento en esta operación? ¿Dónde? En cualquier caso, añade al sumador el hardware necesario para detectar el desbordamiento.
Calcula el tiempo de ejecución de esta operación si $\Delta = 5$ ns.

3.15 Diseña un sumador CSA multinivel con máximo paralelismo para sumar 6 números naturales de 6 bits utilizando para la suma final un sumador RCA. Los módulos CSA y RCA son también de 6 bits.
N1= 001100, N2 = 010101, N3 = 000011, N4 = 100001, N5 = 010111, N6 = 000100
Indica los resultados parciales que se obtienen en cada sumador y el resultado final. ¿Ha ocurrido desbordamiento?, ¿Dónde?, ¿Por qué?
Calcula el tiempo de respuesta del sistema siendo $\Delta = 2$ ns.

3.16 Dado el sumador de la figura para sumar 7 números naturales de 6 bits.



- Sean **N1** = 000100, **N2** = 000110, **N3** = 000111, **N4** = 001001 y **N5** = 001010
  Indica sobre el papel la operación que realizan los tres primeros sumadores (csa1, cpa1 y csa2).
- Calcula el tiempo necesario para obtener la suma de los 7 números siendo $\Delta = 2$ ns.
- Reestructura el sumador construyendo otra estructura que utilice los mismos módulos pero tenga un tiempo de respuesta mínimo. ¿Cuál es ese tiempo?
- ¿De cuántos bits tienen que ser los csa y cpa para asegurar que no se produce desbordamiento en el resultado?

3.17 Hay que construir un sumador para sumar 7 números naturales de 5 bits. Se escogen dos esquemas diferentes para el mismo:

1. Un árbol de Wallace con máximo paralelismo, con un RCA para la suma final.
2. Un sumador CSA mixto con 3 niveles (un único CSA por nivel) y un RCA para la suma final.

-- Dibuja con precisión la estructura de los sumadores correspondientes a cada uno de los dos esquemas anteriores y calcula sus respectivos tiempos de respuesta, de acuerdo a los retardos estándar habitualmente utilizados. Considera un retardo Δ para el resto de circuitos (registros etc).

-- Dados los números N1=3, N2=1, N3=4, N4=6, N5= 7, N6=3, N7=8, indica el resultado de los sumadores que has colocado en los dos primeros niveles del árbol de Wallace que has construido.

4.- Multiplicación

4.1 Realiza los siguiente productos utilizando un sumador CSA para las sumas intermedias y un sumador corriente para la suma final (RCA):

a) 14×11, mediante el algoritmo de "suma y desplazamiento"
b) $12 \times (-5)$, mediante el algoritmo de Booth en base 2
c) $(-15) \times (-10)$, mediante el algoritmo de Booth en base 2

4.2 Haz las siguientes multiplicaciones mediante el algoritmo "suma o salta". ¿Cuántos bits se necesitan para el resultado?

a) $X = 10010$, $Y = 11011$ complemento a 2
b) $A = 001110$, $B = 110110$ complemento a 2
c) $(-7) \times (-5)$

4.3 Realiza el producto $35 \times (-13)$ utilizando el algoritmo de Booth en base 4 para recodificar el multiplicador.

4.4 Calcula el **tiempo de respuesta** de las multiplicaciones $4 \times (-6)$ y $(-6) \times 4$ con cada uno de los algoritmos indicados a continuación, considerando los tiempos: desplazamiento = 5 ns (sea de 1 o 2 bits); suma/resta = 25 ns. (El resto de tiempos los consideraremos cero.)

a) Suma y desplazamiento
b) Suma o salta
c) Algoritmo de Booth en base 2

¿De cuántos bits serán las sumas en cada caso? ¿Cuántos bits tendrá el registro del resultado?

4.5 Sea el siguiente número de 8 bits: 11110110. Recodifícalo aplicando el algoritmo de Booth en base 2 y base 4 para los dos casos siguientes:

a) si se trata de un número entero en C2
b) si se trata de un número natural

¿Cuántos dígitos se necesitan para representar el nuevo número?

4.6 Se desea realizar el producto de dos números enteros $A \times B$, donde A=1001 0100 está representado en signo magnitud y B=100-2 ha sido recodificado según Booth base 4.

a) Representar dichos números en complemento a dos.
b) Calcular el producto utilizando el Algoritmo de Booth base 2.

4.7 Considera un multiplicador secuencial común para datos de 12 bits, con el que queremos multiplicar los dos números enteros siguientes:

N1 = 0000 0001 1110 y N2 = 1111 1101 1001

Indica paso a paso cómo se realiza este producto en los dos casos siguientes:

a) Aplicando el algoritmo de Booth en base 2
b) Aplicando el algoritmo de Booth en base 4

Suponiendo que el tiempo de respuesta de las operaciones suma/resta es 3T y que los desplazamientos requieren de un tiempo T (sean de 1 o 2 bits), ¿cuánto tiempo se necesita para realizar la multiplicación? ¿qué porcentaje de tiempo se ahorra con el segundo algoritmo? ¿se ha producido desbordamiento?

4.8 El algoritmo de Booth presenta un pequeño problema. Si en el multiplicador a recodificar aparece la secuencia <... 1 1 0 1 1 ...> o <... 0 0 1 0 0 ...>, el número de unos en la nueva codificación es mayor que en la original; por tanto, obtendremos una codificación que aumenta el número de operaciones para hacer la multiplicación.

Por ejemplo,	multiplicador	0 1 **0** 1 1 0 0 0 **1** 0 0 (0)	4 dígitos a 1
	tras recodificar	1 **-1 1** 0 -1 0 0 **1 -1** 0 0	6 dígitos a 1

¿Por qué ocurre esto? Porque al recodificar el número estamos <u>suponiendo</u> que una cadena de unos comienza cuando aparece un 1 (o que finaliza cuando nos encontramos con un 0). Si no se trata de eso, sino que nos encontramos con un 1 o un 0 aislado dentro de una cadena, el resultado de la recodificación es peor que el que teníamos; en otras palabras, se ha hecho una apuesta pero con poca información (sólo se tiene en cuenta un bit), y cuando no se acierta hay que pagar una penalización.

¿Cuál puede ser la solución? No se deben recodificar cadenas de un sólo componente (1 o 0), ya que si lo hacemos obtenemos la pareja 1 –1 (o –1 1). Por lo tanto al hacer la recodificación habrá que **analizar** como mínimo **dos bits**, el que se está recodificando y el siguiente, con el fin de detectar con mayor seguridad el comienzo o el final de una cadena. A esta mejora del algoritmo de Booth se le conoce con el nombre de **recodificación canónica**. El objetivo sigue siendo hacer el menor número de sumas/restas posible. Para la recodificación se utiliza la siguiente tabla en la que C es un bit de control que indica si estamos dentro de una cadena de unos (inicialmente vale 0). El dígito que se está recodificando es d_i y d_{i+1} es el siguiente (ojo con el bit de más peso, porque hay que ampliar la representación del número 1 bit más para tener d_{i+1}.). El resultado de la recodificación es d_i'.

d_{i+1}	d_i	C	d_i'	C´ (nuevo)
0	0	0	0	0
0	1	0	1	0
1	0	0	0	0
1	1	0	–1	1
0	0	1	1	0
0	1	1	0	1
1	0	1	–1	1
1	1	1	0	1

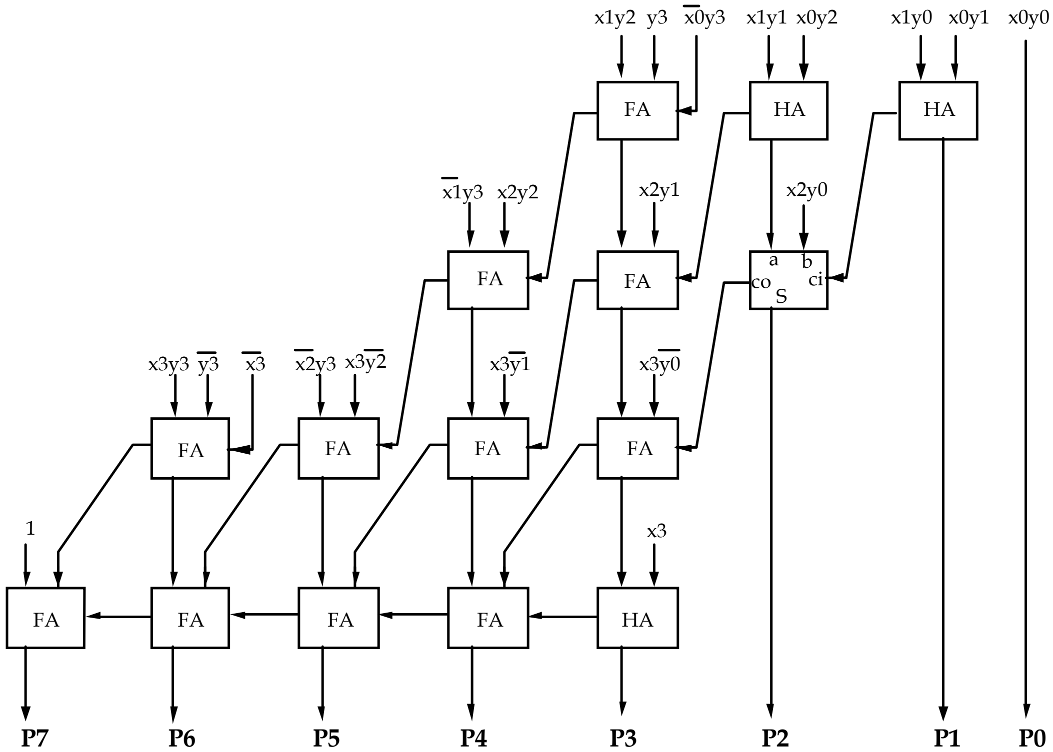
• Recodifica los números siguientes mediante el algoritmo de Booth y mediante el nuevo algoritmo, verificando las diferencias entre ambos:

a) (0) 0 0 0 1 0 0 1 ($C_{inicial} = 0$)
b) (0) 0 1 1 0 1 1 1 0 0 ($C_{inicial} = 0$)

• Realiza el producto 14 × (–22) codificando los operandos según ambos algoritmos. Calcula el tiempo de respuesta en cada caso considerando los siguientes retardos: desplazamiento T; suma/resta 3T (el resto de tiempos cero). ¿Qué porcentaje de tiempo se ahorra con el segundo algoritmo comparándolo con el de Booth?

4.9 Diseñar un multiplicador en *array* para multiplicar dos números de 8 bits según las siguientes estructuras: (a) *Carry save* iterativo y (b) *Carry save* en árbol. Calcula el tiempo de respuesta en cada caso.

4.10 Considera el circuito de la figura siguiente. El conjunto de sumadores FA (*full adder*) y HA (*half adder*) realiza la multiplicación de dos números de 4 bits en **complemento a 2**. Verifica el funcionamiento del circuito multiplicando los dos números siguientes: X=1001 e Y=1011. Indica el resultado parcial que se genera en cada elemento.



Si el tiempo de respuesta en los sumadores es de 2Δ para el bit de llevada y la suma, calcula el tiempo de respuesta del multiplicador si Δ = 4 ns (no consideres el retardo de las puertas AND). ¿Cuál es el *throughput* máximo de este sumador, es decir, el número máximo de multiplicaciones que puede realizar por segundo?

4.11 Construir un multiplicador para multiplicar dos números de 8 bits, utilizando multiplicadores de 4 bits. Intenta después el de 12 bits.

4.12 Se quiere diseñar un circuito aritmético para calcular z = (k**2) + 2 × k teniendo en cuenta que k es un número natural menor que 2^{12}. El nº de bits del resultado debe asegurar que no se produce OVF.
Disponemos de los siguientes componentes para el diseño: multiplicadores de números naturales de 4 bits (resultado 8 bits), módulos CSA de 4 bits, sumadores de un bit, y puertas básicas de cualquier tipo. El precio de cada componente (P) y el retardo (T) son los siguientes:

multiplicadores:	T = 20 y P = 10	sumadores de 1 bit:	T = 3 y P = 0,5
sumadores CSA:	T = 3 y P = 2	puertas simples:	T = 1 y P = 0,2

a) Diseña detalladamente el circuito de menor coste, reflejando con claridad el hardware y las conexiones.
b) Calcula el tiempo de respuesta del sistema diseñado.

4.13 Dados dos números: *A* = 101101 (codificado en complemento a 2) y *B* = 0 –1 –2 (recodificado según Booth base 4), se pide:
a) Indica de qué valores se trata (en base 10).
b) Codifica *B* en complemento a 2 y recodifícalo según Booth base 2.
c) Realiza el producto A × B utilizando un sumador CSA para las sumas intermedias y un sumador serie para la suma final.

5.- División

5.1 Realiza la división x/y, siendo $x = 8$ e $y = 7$, mediante los siguientes algoritmos:
a) División con restauración
b) División sin restauración

5.2 Realiza la división 14/3 mediante el algoritmo de división sin restauración explicando lo más claramente posible todos los pasos realizados, así como el cociente y el resto. $n = 5$, $r = 2$.

5.3 Dado el siguiente cociente, 110101, responde a las siguientes cuestiones:
a) Si se ha utilizado el algoritmo de división con restauración, ¿cuántas restas se han realizado, es decir, cuántas veces no se ha necesitado la restauración?
b) Si se ha utilizado el algoritmo de división sin restauración, ¿cuántas sumas y restas se han realizado? Explícalo concisamente.

5.4 Se quiere calcular el valor $A = C \times D - B/E$ en un caso concreto. Explica el procedimiento a seguir así como el tiempo de respuesta de acuerdo con las siguientes características

- B=011010, C=000011, D=111000 y E=000101 en complemento a 2
- el resultado ha de conseguirse en signo magnitud (6 bits)
- tiempo para sumas y restas 2T, desplazamientos T (uno o más bits), y el resto de tiempos 0
- todas las operaciones son secuenciales
- algoritmo para la multiplicación: Booth
- algoritmo de división sin restauración
- para las sumas: sumador RCA

5.5 Sean X=16 e Y=3 (números naturales de 5 bits). Realiza la división X/Y (16/3) en binario mediante el algoritmo de división sin restauración. Obtén el cociente con un bit decimal (por ejemplo: 11001,1). ¿Cuál es el resto?

6.- Coma flotante

6.1 Dados los formatos estándar de IEEE, calcula para ambos el mayor y el menor número representable. Calcula también la menor y mayor distancia entre dos números consecutivos.

6.2 Dado el siguiente formato en coma flotante:

s	mantisa (6 bits)	exp.(4 bits)

- mantisa normalizada ($1 \leq |m| < 2$) y bit oculto
- exponente representado en exceso 7
- exponente 1111 reservado para infinito y caracteres NAN
- el 0 se representa mediante una cadena toda a 0

a) ¿Cuántos números diferentes se pueden representar?
b) ¿Cuál es el intervalo de la representación?
c) Representa los números 8,4056 y –8,4056 utilizando las siguientes aproximaciones: inmediato inferior, inmediato superior, truncamiento y redondeo. Indica el error cometido en cada caso.

6.3 Dado el siguiente formato en coma flotante, con base 2:

- 1 bit para el signo
- 7 bits para la mantisa (normalizada, $1 \leq m < 2$) y bit oculto
- 4 bits de exponente, representado en "exceso 7"

(a) ¿A qué conjunto de números reales representa el número DBB_H (formato: signo/mantisa/exponente) utilizando como aproximaciones *redondeo* e *inmediato inferior*?
(b) ¿Cuál es el mayor error que se puede cometer en cada caso?

6.4 Sean A = 1 01100111 0011 y B = 0 00010110 0110 dos números en coma flotante, representados en un formato estándar (s/m/e, $1 \leq m < 2$, exp. "exceso 7"). Realiza la operación $A+B$. ¿Cuál es el resultado? ¿Y el error producido?

6.5 Realizar la división entre los dos números siguientes, representados en coma flotante: dividendo = 1 10101 1010, divisor = 1 00011 0011. Para la división de las mantisas se utiliza el algoritmo de división sin restauración. ¿Cuál es el resultado del cociente? ¿Se produce desbordamiento? Formato de los números: s/m/e, $1 \leq$ m <2, e "exceso $r^n/2 - 1$", bit oculto.

6.6 Dado el siguiente formato de 12 bits en coma flotante: **S**igno (bit de más peso) / **M**antisa: 7 bits, bit oculto, 1≤m<2 / **E**xponente: 4 bits, representación en exceso 7 (bits de menos peso). Infinito (o cero): cadena de 1 (o 0) en la mantisa y en el exponente; caracteres NAN: exponente = 0000

(a) ¿Cuál es el número que se utiliza para representar el infinito (en decimal)? ¿Cuáles son en valor absoluto el mayor y el menor número representables en este formato? Tal y como sabes, la diferencia entre números consecutivos no es constante en este modo de representación. ¿Cuál es la menor y la mayor diferencia entre dos números consecutivos? Da un ejemplo.

(b) Sean los dos números siguientes en el mencionado formato: 409_H y $C0A_H$. Indica el procedimiento para realizar la multiplicación de ambos y cuál es el resultado de la misma (en hexadecimal). ¿Se produce algún error al realizar la operación?

6.7 Dados los números: $9C_H$ y EE_H, formato de coma flotante s/m/e, con las siguientes características:

signo --> 1 bit;
mantisa --> 4 bits, bit oculto, 1≤m<2;
exponente --> 3 bits, exceso 3.

Haz la suma en coma flotante y calcula el error relativo cometido. Indica las distintas fuentes de error que haya. Da los resultados también en base 10. Justifica todos los pasos.

6.8 Representa los dos números a continuación, **+18** y **+3,9**, en el siguiente formato de coma flotante, s/m/e:

signo --> 1 bit;
mantisa --> 4 bits, bit oculto, 1≤m<2;
exponente --> 3 bits, exceso 3;
aproximación = redondeo

A continuación, **haz la resta en coma flotante**. Indica el resultado en el mismo formato, y da su valor en base 10. Justifica todos los pasos. Indica las fuentes de error, así como el error relativo que se produce.

6.9 Sea el siguiente formato: $r = 10$, 6 dígitos para la mantisa normalizada ($0 \leq$ m <1) y 2 dígitos para el exponente representado en "exceso 50". Representa, en base 10, los números $A = 345155$ y $B = 000125{,}5$. Realiza después la suma y la división entre ambos, indicando claramente los pasos seguidos.

6.10 Dado el siguiente formato de 10 bits en coma flotante: *signo* (bit de más peso), *mantisa*: 6 bits, bit oculto $1<m<2$, *exponente* 3 bits en exceso 3 (bits de menos peso), *infinito (o cero)* cadena de 1 (o 0) en la mantisa y en el exponente, caracteres NAN, exponente 111.

a) ¿A qué números o caracteres especiales corresponden los siguientes códigos? (basta indicar los números en notación exponencial)

A	1111110110
B	1011011011
C	1000000000
D	0110001101
E	0100000000
F	1000000111

b) Si se ha utilizado la aproximación de inmediato superior, ¿qué rango de números reales representan los códigos *B* y *D*?

c) Realizar la suma de *B* +*D* expresando claramente las distintas fuentes de error.