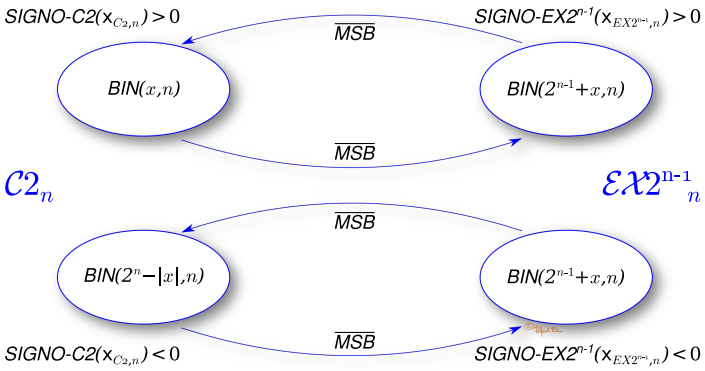
PROBLEMAS: La ruta de datos

9. Especifique qué conversión ha de realizar un operador de cambio de sistema de representación para pasar de complemento a dos sobre 8 bits a exceso a 2^{n-1} también sobre 8 bits. Demuestre su propuesta.

SOLUCIÓN:

El esquema siguiente muestra las transformaciones que hay que realizar para pasar la representación de un número en complemento a dos a su representación en exceso a 2^{n-1} y viceversa.

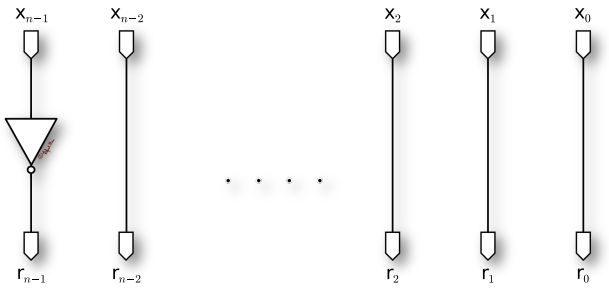


A continuación se demuestra formalmente cómo se pasa de complemento a dos a exceso:

$$x_{C2,n} + BIN(2^{n-1}, n) = BIN(2^n + x, n) + BIN(2^{n-1}, n) =$$
$$BIN(2^{n-1} + x, n) = EX\text{-}2^{n-1}(x, n) = x_{EX2^{n-1},n}$$

Y cómo se pasa de exceso a complemento a dos:

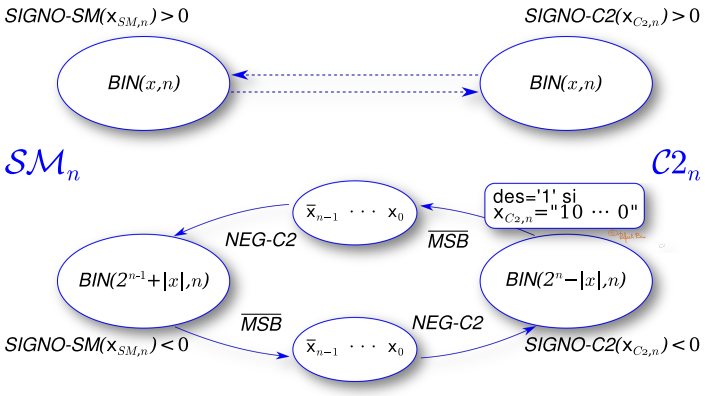
$$x_{EX2^{n-1},n} + BIN(2^{n-1}, n) = BIN(2^{n-1} + x, n) + BIN(2^{n-1}, n) =$$
$$BIN(2^n + x, n) = C2(x, n) = x_{C2,n}$$



10. Se desea construir un operador *hardware* de 16 bits que convierta números representados en complemento a dos a números representados en signo-magnitud. Especifique qué transformaciones hay que realizar y proponga una implementación suponiendo que dispone de todos los operadores y puertas lógicas que necesite.

SOLUCIÓN:

El esquema siguiente muestra las transformaciones que hay que realizar para pasar la representación de un número en signo-magnitud a su representación en complemento a dos y viceversa.



Como vemos, los positivos no requieren transformación mientras que los negativos deben pasar dos: el complemento del MSB ($\overline{MSB}$) y el negativo en complemento a dos (*NEG-C2*). Además, en el caso de pasar de la representación en complemento a dos a la de signo-magnitud hay que tener en cuenta que se puede producir desbordamiento (des='1' si $x_{C2,n}$="10 ⋯ 0").

Aunque este problema nos pide un operador que realice las traducciones en un sentido solamente (de C2 a SM), vamos a considerar que tenemos una señal de control que permita traducir en ambos sentidos ($\overline{\text{SM\_a\_C2}}$/C2_a_SM) dando solución, así, al problema siguiente también.

El *hardware* tiene que combinar un inversor condicional para hacer el complemento del MSB ($\overline{MSB}$) y un operador de negativo en complemento a dos (*NEG-C2*) que entrarán en funcionamiento exclusivamente cuando la representación de entrada sea negativa (x_{n-1}='1').

La tabla de verdad que recopila todas las especificaciones tanto para el MSB del resultado (r_{n-1}) como para el desbordamiento (des) es la siguiente:

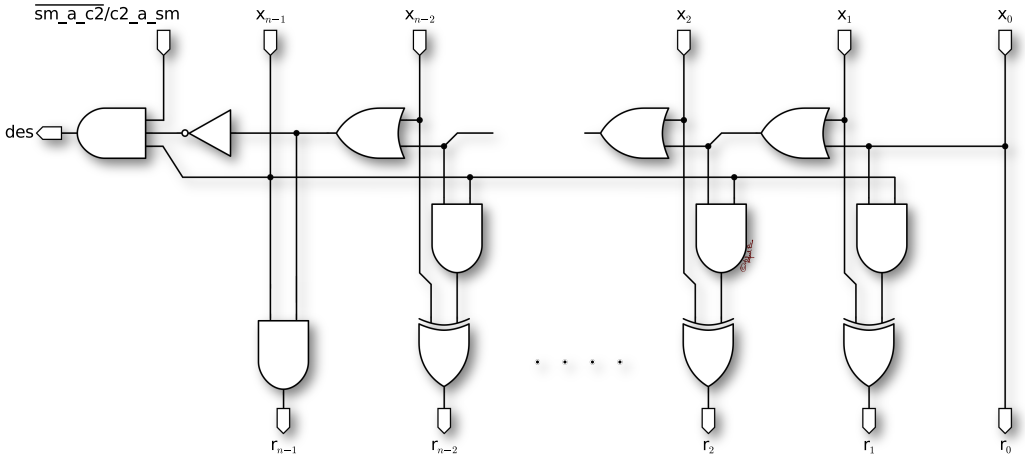
$\overline{\text{SM\_a\_C2}}$/C2_a_SM	x_{n-1}	cero	r_{n-1}	des
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0

La señal denominada cero corresponde a la propagación del detector de cero que entrega el peso 2^{n-2}. A las vista de las tablas de verdad concluimos en las siguientes ecuaciones lógicas:

$$r_{n-1} = \text{cero} \cdot x_{n-1}$$

$$\text{des} = \overline{\text{SM\_a\_C2}}/\text{C2\_a\_SM} \cdot \overline{\text{cero}} \cdot x_{n-1}$$

De donde el mejor circuito posible será el siguiente:



Este circuito sirve tanto para traducciones de signo-magnitud a complemento a dos como en sentido contrario.

11. Complete el operador del problema anterior con los elementos necesarios para realizar la conversión en sentido contrario.

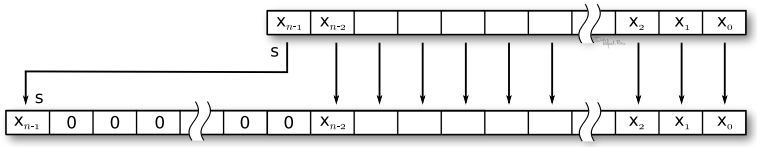
SOLUCIÓN:

La solución del problema anterior ya ofrece lo pedido en este puesto que sólo hay que añadir el detector de desbordamiento.

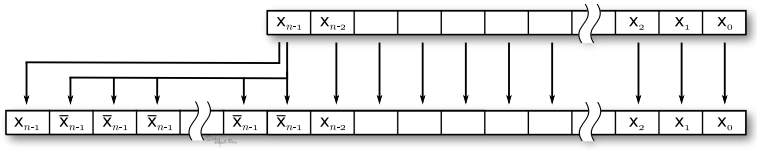
12. La unidad de coma flotante de un procesador solamente trabaja en doble precisión. Cuando se le pasan datos en simple precisión realiza automáticamente las extensiones de signo de exponente y mantisa antes de operar. Explique cómo se hacen dichas extensiones sabiendo que los exponentes se codifican en exceso a 2^{n-1} y las mantisas en signo-magnitud. El formato de simple precisión asigna 8 bits al exponente y 24 a la mantisa mientas que el formato de doble precisión asigna 11 y 53 respectivamente.

SOLUCIÓN:

A continuación se ilustra cómo se hacen las extensiones de signo en signo-magnitud:



Y en exceso a 2^{n-1}:



DEMOSTRACIÓN:

Sea $x_{EX2^{n-1},n}$ la representación en exceso a 2^{n-1} de un valor x sobre n bits. La representación del mismo valor en exceso a 2^{m-1}sobre m bits con $m > n$ será por definición:

$$x_{EX2^{m-1},m} = BIN(2^{m-1} + VAL\text{-}EX2^{n-1}(x_{EX2^{n-1},n}), m) = BIN(2^{m-1} + \sum_{i=0}^{n-2} x_i \cdot 2^i - \overline{x}_{n-1} \cdot 2^{n-1})$$

Si $\mathsf{x}_{n-1} = 1$ tenemos:

$$2^{m-1} + \sum_{i=0}^{n-2} x_i \cdot 2^i - \overline{x}_{n-1} \cdot 2^{n-1} = 2^{m-1} + \sum_{i=0}^{n-2} x_i \cdot 2^i$$

Lo cual implica que $\mathsf{x}_{m-1} = 1$ y $\mathsf{x}_i = 0$ para todo $i \in [n-1, m-2]$.

Si $\mathsf{x}_{n-1} = 0$ tenemos:

$$2^{m-1} + \sum_{i=0}^{n-2} x_i \cdot 2^i - \overline{x}_{n-1} \cdot 2^{n-1} = 2^{m-1} + \sum_{i=0}^{n-2} x_i \cdot 2^i + 2^{n-1}$$

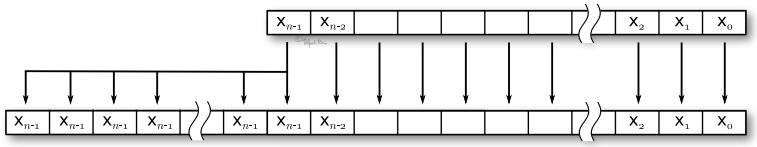
Lo cual implica que $\mathsf{x}_{m-1} = 0$ y $\mathsf{x}_i = 1$ para todo $i \in [n-1, m-2]$.

Ya que $2^{m-1} - 2^{n-1}$ es una cadena de unos desde el peso 2^{n-1} hasta el peso 2^{m-2}.

13. Diseñe un operador de extensión de signo en complemento a dos que extienda representaciones de 8 bits a 16 bits.

SOLUCIÓN:

A continuación se ilustra cómo se hacen las extensiones de signo en complemento a dos o complemento a uno:

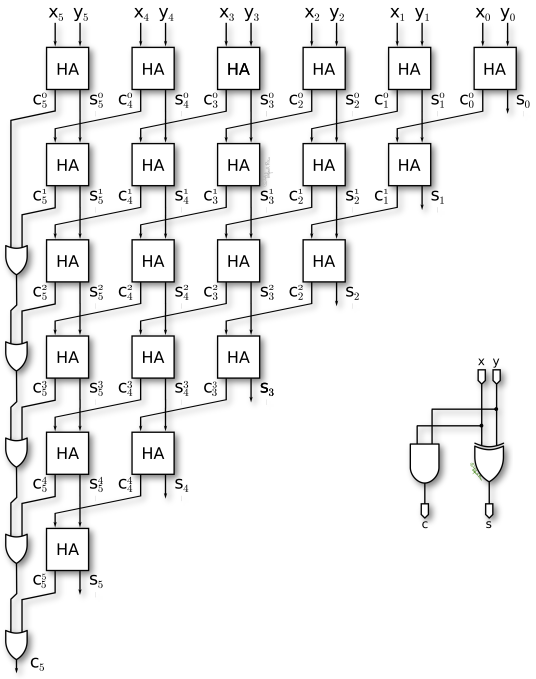


La demostración en cada caso es análoga a la presentada en el problema anterior.

14. Diseñe un sumador de 6 bits basado en mediosumadores. Indique la complejidad $\mathcal{O}(g(x))$ del operador tanto en área como en tiempo. Suponiendo que las ecuaciones lógicas del semisumador son $\mathsf{s}_i = \mathsf{x}_i \oplus \mathsf{y}_i$ y $\mathsf{c}_i = \mathsf{x}_i \cdot \mathsf{y}_i$ indique el retardo temporal medido en niveles lógicos.

SOLUCIÓN:

La figura siguiente muestra el esquema *hardware* de un sumador basado en mediosumadores. La suma finaliza cuando estamos seguros de que no queda ningún acarreo por sumar y eso sucede a los n pasos siendo n el tamaño de los sumandos. De los acarreos salientes en cada paso, solamente uno puede ser '1' de manera que basta con una cadena de propagación que detecte el primer '1' si es que lo hay.



El retardo del circuito es igual a nr_g siendo r_g el retardo de una puerta lógica ya que cada semisumador trabaja como un operador de bit.

15. Podemos realizar un sumador binario de n bits basado en mediosumadores (HA) o en sumadores completos (FA). Comente el modo de funcionamiento de cada uno. Discuta sus ventajas e inconvenientes y sus posibles áreas de aplicación.

SOLUCIÓN:

El sumador basado en HA obtiene la suma en una sucesión de n sumas de vectores de suma parcial y acarreo parcial mientras que el sumador basado en FA propaga acarreos verdaderos de LSB a MSB.

El tiempo de retardo es similar en diseños basados en HA o en FA pero el coste es menor en el caso de utilizar FA.

No obstante, el diseño basado en HA puede ser conveniente para operaciones de suma de más de dos operandos de entrada (ver sumador CSA o *Carry-Save Adder*).

16. Usando un restador en complemento a dos y las puertas oportunas, realice un circuito comparador de 2 enteros de 4 bits en complemento a 2 ($\mathsf{x}_{C2,4}$ e $\mathsf{y}_{C2,4}$) que proporcione las señales IGUAL y MAYORQUE de manera que IGUAL='1' si $x = y$ y MAYORQUE='1' si $x > y$.

SOLUCIÓN:

La señal IGUAL la obtenemos a partir de un detector de cero situado a la salida del restador aplicado a la ristra de bits del resultado.

Para obtener la señal MAYORQUE hemos de usar el restador. La tabla siguiente muestra los diferentes casos que podemos encontrar y cómo se detectan.

Sean $x, y \in \text{rango}(\mathcal{C}2_n)$. $C2(x-y,n)$ en función de BIN:

operandos	resultado	módulos	r_{n-1}	c_{n-1}	des	$>$
$x, y > 0$	$BIN(2^n + x - y, n)$	$x > y$	0	1	0	1
		$x < y$	1	0	0	0
$x > 0, y < 0$	$BIN(x + \|y\|, n)$	$x + \|y\| \leq 2^{n-1}-1$	0	0	0	1
		$x + \|y\| > 2^{n-1}-1$	1	0	1	1
$x < 0, y > 0$	$BIN(2^n - (\|x\| + y), n)$	$\|x\| + y \leq 2^{n-1}$	1	1	0	0
		$\|x\| + y > 2^{n-1}$	0	1	1	0
$x, y < 0$	$BIN(2^n - \|x\| + \|y\|, n)$	$\|x\| < \|y\|$	0	1	0	1
		$\|x\| > \|y\|$	1	0	0	0

A la vista de la tabla anterior podemos sacar las siguientes ecuaciones:

$$MAYORQUE\text{-}C2(\mathsf{x}, \mathsf{y}, n) = \overline{(\mathsf{x}_{n-1} \oplus \mathsf{y}_{n-1})} \cdot \overline{\mathsf{r}_{n-1}} + \overline{\mathsf{x}_{n-1}} \cdot \mathsf{y}_{n-1}$$

$$MAYORQUE\text{-}C2(\mathsf{x}, \mathsf{y}, n) = \overline{(\mathsf{x}_{n-1} \oplus \mathsf{y}_{n-1})} \cdot \mathsf{c}_{n-1} + \overline{\mathsf{x}_{n-1}} \cdot \mathsf{y}_{n-1}$$

No obstante, esta ecuaciones requieren de la detección del signo de operandos y resultado (o acarreo de salida). Existe una opción más sencilla.

Observamos que siempre que $\mathsf{x}_{C2,n}$ es mayor que $\mathsf{y}_{C2,n}$ se cumple que r_{n-1} es igual a des (incluso en los casos de desbordamiento del resultado). En consecuencia:

$$MAYORQUE\text{-}C2(\mathsf{x}, \mathsf{y}, n) = \overline{\mathsf{r}_{n-1} \oplus \mathsf{des}}$$

Puesto que $SIGNO\text{-}C2(\mathsf{r}, n) = \mathsf{r}_{n-1}$:

$$MAYORQUE\text{-}C2(\mathsf{x}, \mathsf{y}, n) = \overline{SIGNO\text{-}C2(\mathsf{r}, n) \oplus \mathsf{des}}$$

17. Diseñe un comparador para números enteros representados en signo-magnitud que proporcione las señales IGUAL y MAYORQUE.

SOLUCIÓN:

Un comparador para números enteros representados en signo-maginitud trabaja por una parte sobre los signos y por otra con las magnitudes. Dado que las magnitudes son valores absolutos podemos usar un comparador basado en resta o un comparador basado en una cadena de comparadores de 1 bit.

Supongamos que tenemos una cadena de comparadores de 1 bit que nos ofrece la señal IGUAL y la señal ENCIMA y la tenemos aplicada a las magnitudes. La tabla de verdad siguiente especifica cómo se detecta qué operado en mayor que el otro.

x_{n-1}	y_{n-1}	*ENCIMA*	*MAYORQUE*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

La síntesis nos da la siguiente ecuación lógica:

$$MAYORQUE\text{-}SM(\mathsf{x}, \mathsf{y}, n) = \overline{ENCIMA} \cdot \mathsf{y}_{n-1} + ENCIMA \cdot \overline{\mathsf{x}_{n-1}}$$

18. Se desea realizar un sumador entero de n bits basado en el selector de acarreo. Los bloques propagadores son todos del mismo tamaño k. ¿Qué fórmula nos da el tiempo de retardo en función de n y k medido en niveles lógicos? Determine el valor de k que minimiza el retardo para un tamaño n dado.

SOLUCIÓN:

El retardo de un sumador de n bits selector de acarreo homogéneo, es decir, con todos los bloques propagadores del mismo tamaño k es:

$r = 2k + 2\left(\frac{n}{k} - 2\right) + r_{mux}$

El mejor valor de k para un tamaño n dado es un problema de optimización que se resuelve derivando respecto a k e igualando a 0.

$\frac{\partial r}{\partial k} = 0$, es decir, $2 - 2\frac{n}{k^2} = 0$ que nos da: $k = \sqrt{n}$

19. Se desea diseñar un sumador con selección de acarreo de 16 bits. Podemos usar bloques propagadores de 2, 4 y 8 bits. Construya el mejor sumador posible suponiendo que los conmutadores tienen un retardo equivalente a $2r_g$ (retardo de una puerta lógica genérica).

SOLUCIÓN:

Como hemos visto en el problema anterior, el valor óptimo de k es:

$$k = \sqrt{n}$$

Lo que en este caso significa que $k = \sqrt{16} = 4$ bits.