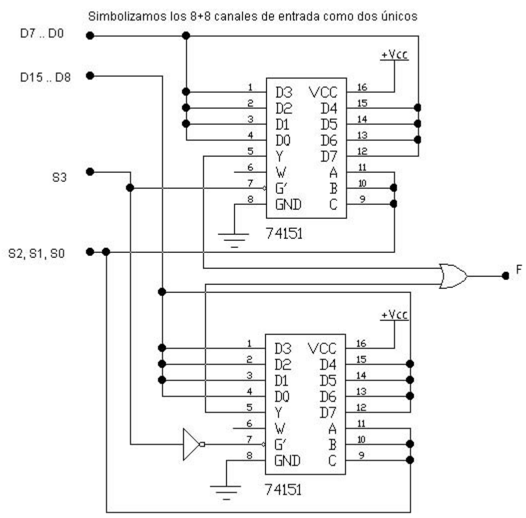
# PDSD - Tema 3: Resolución de los Ejercicios

1. **Ejercicio 1**: Utiliza multiplexores 74151 y cualquier otra lógica necesaria para multiplexar 16 líneas de datos en una única línea de salida.

   Un mux. 74151 presenta 8E/1S y 3 bits de Serlección (8 a 1).




2. **Ejercicio 2**: Utilizando un MUX 4 a 1, implementa

   $f(a,b,c) = ab + ac.$

   Acudiendo al mapa de Karnaugh, la función canónica es:
   $f(a,b,c) = abc + a\bar{b}c + ab\bar{c}.$

   Sobre el mapa de Karnaugh, de la Figura 1, se muestra cuales han sido las simplificaciones de los minter para obtener la función a implementar.

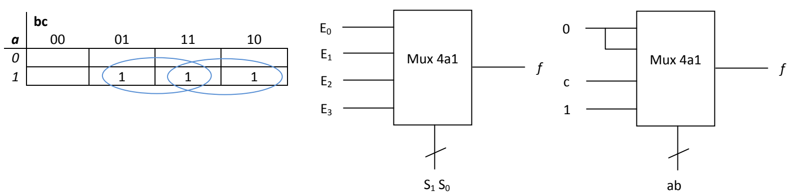



Figura 1. Mapa de Karnaugh - Mux 4a1 genérico - Función implementada

Sabiendo que la función $f$ del Mux 4a1 se expresa de la siguiente forma:

$f = E_0\overline{S}_1\overline{S}_0 + E_1\overline{S}_1S_0 + E_2S_1\overline{S}_0 + E_3S_1S_0,$

sobre la Tabla 1 siguiente se muestra la selección de variables para la implementación, escogiendo $a, b$ como $S_1S_0$ y $c$ entrada al Mux.:

Tabla 1. Mux 4a1

| $a$ | $b$ | $c$ | $f$ | Valor $E_i$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $E_0 = 0$ |
| 0 | 0 | 1 | 0 | |
| 0 | 1 | 0 | 0 | $E_1 = 0$ |
| 0 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 0 | $E_2 = c$ |
| 1 | 0 | 1 | 1 | |
| 1 | 1 | 0 | 1 | $E_3 = 1$ |
| 1 | 1 | 1 | 1 | |

Finalmente, la Figura 1 muestra la conexión de un Mux 4a1 para implementar la función propuesta.

3. **Ejercicio 3**: Utilizando un MUX de 8 a 1, implementa la función:

$f = \Sigma m(1, 2, 5, 6, 7, 8, 10, 12, 13, 15).$

Teniendo en cuenta que es una función de 4 variables ($A_3A_2A_1A_0$), se necesitará un Mux 8a1, dedicando 3 variables a la selección y la cuarta variable como posible entrada al Mux. La Tabla 2 muestra las diferentes agrupaciones y la Figura 2 muestra las conexiones a realizar en el Mux 8a1 para realizar la función propuesta:

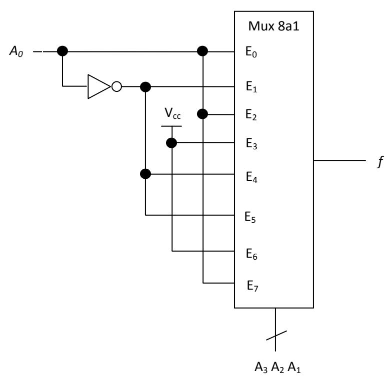



Figura 2. Función implementada con Mux 8a1

Tabla 2. Mux 8a1

| $A_3$ | $A_2$ | $A_1$ | $A_0$ | $f$ | Valor $E_i$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | $E_0 = A_0$ |
| 0 | 0 | 0 | 1 | 1 | |
| 0 | 0 | 1 | 0 | 1 | $E_1 = \overline{A}_0$ |
| 0 | 0 | 1 | 1 | 0 | |
| 0 | 1 | 0 | 0 | 0 | $E_2 = A_0$ |
| 0 | 1 | 0 | 1 | 1 | |
| 0 | 1 | 1 | 0 | 1 | $E_3 = 1$ |
| 0 | 1 | 1 | 1 | 1 | |
| 1 | 0 | 0 | 0 | 1 | $E_4 = \overline{A}_0$ |
| 1 | 0 | 0 | 1 | 0 | |
| 1 | 0 | 1 | 0 | 1 | $E_5 = \overline{A}_0$ |
| 1 | 0 | 1 | 1 | 0 | |
| 1 | 1 | 0 | 0 | 1 | $E_6 = 1$ |
| 1 | 1 | 0 | 1 | 1 | |
| 1 | 1 | 1 | 0 | 0 | $E_7 = A_0$ |
| 1 | 1 | 1 | 1 | 1 | |

4. **Ejercicio 4**: Utilizando el decodificador 74154, implementa un circuito para decodificar un número de 5 bits. Determina la salida que se activa al introducir el código binario de entrada 10101.

   Hay que encontrar la función $f$ que se active a 1 cuando se dé la combinación de entradas 10101. Elaborando la tabla de verdad, según se muestra en la Tabla 3, pasamos a realizar las conexiones adecuadas en el dispositivo 74154 decodificador 4 a 16 que se muestran en la Figura 3.

Tabla 3. Decodificador 4a16

| $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | $f$ | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | |
| | | .... | | | | |
| | | .... | | | | |
| 0 | 1 | 1 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 0 | 0 | |
| | | .... | | | | |
| 1 | 0 | 1 | 0 | 1 | 1 | (5) |
| | | .... | | | | |
| 1 | 1 | 1 | 1 | 1 | 0 | |

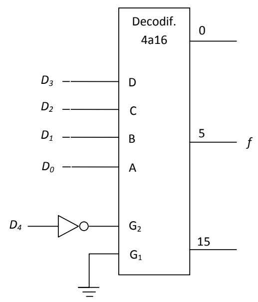



Figura 3. Función implementada con Decodificador 4a16

5. **Ejercicio 5**: Utiliza el 74151 para implementar la función:

   $f = \Sigma m(0,2,4,6).$

   La Tabla 4 muestra la función activa en los minterm requeridos y la Figura 4 muestra las conexiones en el CI 74151.

Tabla 4. Tabla de verdad de la función requerida

| $S_2$ | $S_1$ | $S_0$ | $f$ | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | $D_0 = 1$ |
| 0 | 0 | 1 | 0 | $D_1 = 0$ |
| 0 | 1 | 0 | 1 | $D_2 = 1$ |
| 0 | 1 | 1 | 0 | $D_3 = 0$ |
| 1 | 0 | 0 | 1 | $D_4 = 1$ |
| 1 | 0 | 1 | 0 | $D_5 = 0$ |
| 1 | 1 | 0 | 1 | $D_6 = 1$ |
| 1 | 1 | 1 | 0 | $D_7 = 0$ |

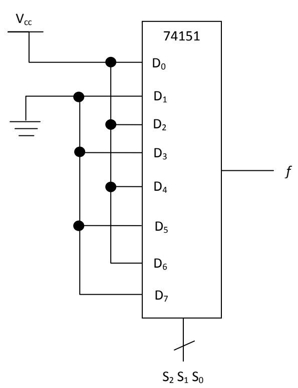



Figura 4. Conexiones del CI 74151 para la función $f = \Sigma m(0,2,4,6)$.

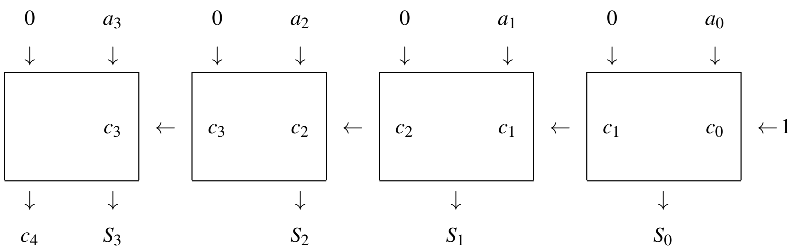
6. **Ejercicio 6**: Utiliza el 74151 para implementar la función:

   $f = \Sigma m(0,1,2,3,4,9,13,14,15)$.

   Sigue el mismo procedimiento que el empleado en el ejercicio 3.

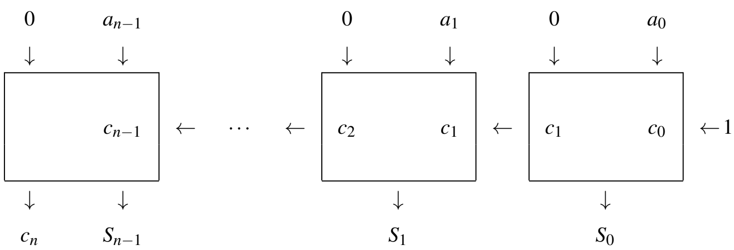
7. **Ejercicio 7**: Dado un numero de 4 bit, diseña el circuito que sume "1" a este circuito.

   Escogemos 4 bloques full-adder y realizamos el montaje de bloques, introduciendo "1" en el $c_0$ (carry inicial) del primer bloque. Un operando será el nº $A = a_3a_2a_1a_0$ y el otro operando será "0" en todos sus bits.




   Podemos plantearlo de manera genérica con $n$ bits: $A = a_{n-1}a_{n-2} \cdots a_1a_0$.

   Escogemos los $n$ bloques full-adder y realizamos el montaje de bloques, del mismo modo que con 4 bits, introduciendo "1" en el $c_0$ del primer bloque. Un operando será el nº $A$ y el otro operando será "0" en todos sus bits.




8. **Ejercicio 8**: Dadas dos variables de 4 bit positivas, diseña un restador $(A-B)$ tal que la resta esté realizada en complemento a 2. El resultado puede ser positivo o negativo. Representa el resultado en valor absoluto y el signo se mostrara sobre un diodo led. Si el resultado es negativo, el diodo led estará encendido.

   La primera consideración es que al ser números de 4 bits, y trabajar en complemento a 2, es necesario añadir un quinto bit (bit de signo) para poder realizar el complemento a 2 correctamente. Veamos un ejemplo:

   Sean $A$ y $B$ las variables a operar, $R = A - B$, si $A = 0111$, es decir $A = 7$, y $B = 1111$, es decir, $B = 15$, el complemento a 1 de $B$ es 0000 y el complemento a 2 es 0001, por lo que $R = A - B = 0111 + 0001 = 1000$, es decir, $R = 8$, cuando debería ser $R = -8$.

   Debemos añadir un bit más para la operación (bit de signo), tendremos $A = 00111$ y $B = 01111$, por lo que el complemento a 2 de $B$, será 10001. Realizando la operación de suma, tenemos que $R = 00111 + 10001 = 11000$. como el bit de signo es 1, se trata de un número negativo, por lo que realizando el complemento a 2 tendrémos el valor absoluto e indicaremos que es negativo: $11000 \rightarrow 01000$, por lo tanto $R = -8$.

   Una vez obtenido el valor de $R$, hay que determinar el signo con el fin de mostrar el valor absoluto de la operación. Si es negativo ($Signo = 1$) hay que volver a realizar el complemento a 2. El resultado final se obtendrá a traves de un multiplexor 2a1 de 4 bits por variable de entrada, siendo la señal de control el signo.

El esquema viene reflejado en la Figura 5:

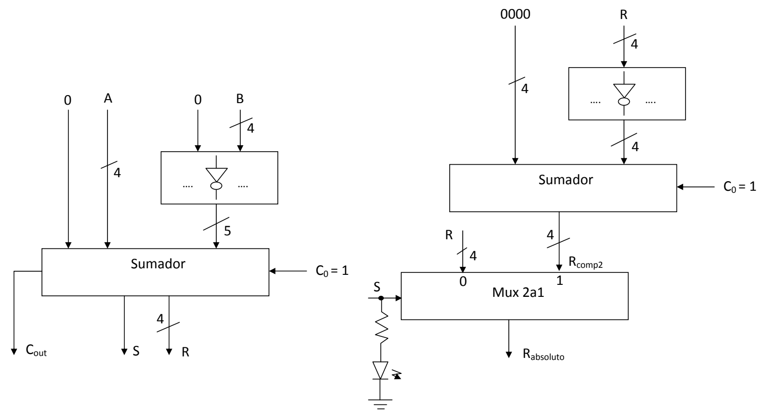



Figura 5. Restador del ejercicio 8

9. **Ejercicio 9**: Dado un numero de 4 bit, obtener el resultado de su multiplicación por 2. Obtener, ademas, el resultado de su división por 2. Observando el *modus operandi* de la multiplicación por 2, efectúa las multiplicaciones por 3, 4 y 5.

   Las operaciones a realizar son las siguientes:
   $N = A \times 2$, $N = A \div 2$, $N = A \times 3$, $N = A \times 4$, y $N = A \times 5$.

   - $N = A \times 2$
     $N = (a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0 2^0) \times 2 = a_3 2^4 + a_2 2^3 + a_1 2^2 + a_0 2^1$, observese que el nº se ha desplazado hacia la izquierda.
   - $N = A \div 2$
     $N = (a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0 2^0) \times 2^{-1} = a_3 2^{3-1} + a_2 2^{2-1} + a_1 2^{1-1} + a_0 2^{0-1}$
     $N = a_3 2^2 + a_2 2^1 + a_1 2^0 + a_0 2^{-1}$, observese que el nº se ha desplazado hacia la derecha, desapareciendo el término $a_0$ ya que estamos trabajando con números enteros.
   - $N = A \times 3$
     $N = (a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0 2^0) \times (2^1 + 2^0) = (a_3 2^4 + a_2 2^3 + a_1 2^2 + a_0 2^1) + (a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0 2^0) =$
     $a_3 2^4 + (a_3 + a_2) 2^3 + (a_1 + a_2) 2^2 + (a_0 + a_1) 2^1 + a_0 2^0$
   - $N = A \times 4$ y $N = A \times 5$: Siguen el mismo procedimiento, es decir, $4 = 2^2$ y $5 = 2^2 + 2^0$.

   Podemos plantearlo de manera genérica con $n$ bits: $A = a_{n-1} a_{n-2} \cdots a_1 a_0$.

   - $N = A \times 2$
     $N = (a_{n-1} 2^{n-1} + a_{n-2} 2^{n-2} + \cdots + a_1 2^1 + a_0 2^0) \times 2 = a_{n-1} 2^n + a_{n-2} 2^{n-1} + \cdots + a_1 2^2 + a_0 2^1$, observese que el nº se ha desplazado hacia la izquierda.
   - $N = A \div 2$
     $N = (a_{n-1} 2^{n-1} + a_{n-2} 2^{n-2} + \cdots + a_1 2^1 + a_0 2^0) \times 2^{-1} = a_{n-1} 2^{n-2} + a_{n-2} 2^{n-3} + \cdots + a_2 2^{2-1} + a_1 2^{1-1} + a_0 2^{0-1}$
     $N = a_{n-1} 2^{n-2} + a_{n-2} 2^{n-3} + \cdots + a_2 2^1 + a_1 2^0 + a_0 2^{-1}$, observese que el nº se ha desplazado hacia la derecha, desapareciendo el término $a_0$ ya que estamos trabajando con números enteros.

- $N = A \times 3$
  $N = (a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \cdots + a_1 2^1 a_0 2^0) \times (2^1 + 2^0) = (a_{n-1}2^n + a_{n-2}2^{n-1} + \cdots + a_1 2^2 + a_0 2^1) + (a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \cdots + a_1 2^1 + a_0 2^0) =$
  $a_{n-1}2^n + (a_{n-1} + a_{n-2})2^{n-1} + \cdots + (a_1 + a_2)2^2 + (a_0 + a_1)2^1 + a_0 2^0$

10. **Ejercicio 10**: Dado un sistema de computación que realiza sus operaciones aritméticas de suma en binario natural, expresados los operandos en 4 bit, el resultado puede no ser directamente comprensible por un usuario. Para facilitar la labor de comprensión se deberá trasladar el resultado (binario natural) a BCD.

    El planteamiento es realizar la suma en binario natural y posteriormente realizar la transformación a BCD.

    Sean las variables $A$ y $B$, la suma será $S = A + B$. Sabemos que el menor valor de la suma es $S = 0 + 0 = 0$ y el mayor valor es $S = 15 + 15 = 30$, es decir 11110, ya que la suma se realiza en binario natural. Estos resultados deben presentar una expresión final en BCD ($S_{BCD}$).

    La casuística de la suma $S = A + B$ se presenta en la Tabla 5:

Tabla 5. Suma binaria y transformación a suma en BCD

| $S$ | | $S_{BCD}$ | | | *Factor* |
|---|---|---|---|---|---|
| 0 | 00000 | 00 | 0000 | 0 | +0 |
| 1 | 00001 | 00 | 0001 | 1 | +0 |
| … | … | … | … | … | … |
| 9 | 01001 | 00 | 1001 | 9 | +0 |
| 10 | 01010 | 01 | 0000 | 16 | +6 |
| … | … | … | … | … | … |
| 19 | 10011 | 01 | 1001 | 25 | +6 |
| 20 | 10100 | 10 | 0000 | 32 | +12 |
| … | … | … | … | … | … |
| 29 | 11101 | 10 | 1001 | 41 | +12 |
| 30 | 11110 | 11 | 0000 | 48 | +18 |

El término *Factor* muestra el valor que hay que sumar a la suma $S$ para obtener el resultado en BCD ($S_{BCD} = S + Factor$). De esta Tabla, se extrae la siguiente casuística:

- Si $S \leq 9$, no hay que realizar ninguna transformación del resultado ($Factor = 0$) $\Rightarrow S_{BCD} = S + 0$.
- Si $10 \leq S \leq 19$, hay que transformar el resultado ($Factor = 6$) $\Rightarrow S_{BCD} = S + 6$.
- Si $20 \leq S \leq 29$, hay que transformar el resultado ($Factor = 12$) $\Rightarrow S_{BCD} = S + 12$.
- Si $S = 30$, hay que transformar el resultado ($Factor = 18$) $\Rightarrow S_{BCD} = S + 18$.

Veamos tres casos significativos:

| | | | | | |
|---|---|---|---|---|---|
| $10 \leq S \leq 19$ | 01010 | $\rightarrow$ | 01 | 0000 | : Se observa que sumando 6 (110) a |
| | +110 | | | | 10 (01010) |
| | 01 0000 | | | | se obtiene el equivalente BCD |
| $20 \leq S \leq 29$ | 10100 | $\rightarrow$ | 10 | 0000 | : Se observa que sumando 12 (1100) a |
| | +1100 | | | | 20 (10100) |
| | 10 0000 | | | | se obtiene el equivalente BCD |
| $S = 30$ | 11110 | $\rightarrow$ | 11 | 0000 | : Se observa que sumando 18 (10010) a |
| | +10010 | | | | 30 (11110) |
| | 11 0000 | | | | se obtiene el equivalente BCD |

Para poder generar estos factores, hay que detectar qué números son $S \leq 9$, $10 \leq S \leq 19$, $20 \leq S \leq 29$, $S = 30$. Para ello podemos reflejar sobre un mapa de Karnaugh, de 5 variables, todos los posibles resultados que cumplan estos requisitos, como minterms. El resultado de la suma será $S_{BCD} = S_{BCD4}S_{BCD3}S_{BCD2}S_{BCD1}S_{BCD0}$ y un bit de carry $C_{BCD}$ que es $S_{BCD5}$. Por cada caso se obtendrá un $Factor$ que llamaremos $F_a$, $F_b$, $F_c$ y $F_d$.

*a*) Si $S \leq 9 \Rightarrow F_a$

*b*) Si $10 \leq S \leq 19 \Rightarrow F_b$

*c*) Si $20 \leq S \leq 29 \Rightarrow F_c$

*d*) Si $S = 30 \Rightarrow F_d$

Sea $S = CS_3S_2S_1S_0$.

| $S_3S_2$ \ $S_1S_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0001 | 0001 | 0001 | 0001 |
| 01 | 0001 | 0001 | 0001 | 0001 |
| 11 | 0010 | 0010 | 0010 | 0010 |
| 10 | 0001 | 0001 | 0010 | 0010 |

$C = 0$

| $S_3S_2$ \ $S_1S_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0010 | 0010 | 0010 | 0010 |
| 01 | 0100 | 0100 | 0100 | 0100 |
| 11 | 0100 | 0100 | X000 | 1000 |
| 10 | 0100 | 0100 | 0100 | 0100 |

$C = 1$

En vez de confeccionar 4 tablas de Karnaugh, introducimos en la misma los minterm relativos a los 4 factores (funciones). Los minterm reflejados guardan la siguiente relación: $F_dF_cF_bF_a$, por lo tanto, después de realizar las agrupaciones de simplificación, por cada factor, tendremos que:

$F_a = \overline{C}\bar{S}_3 + \overline{C}\bar{S}_2\bar{S}_1 = \overline{C}(\bar{S}_3 + \bar{S}_2\bar{S}_1)$

- Es decir, si $S \leq 9$ entonces: $F_a = 1$ si nó $F_a = 0$.

$F_b = \overline{C}S_3S_2 + \overline{C}S_3S_1 + C\bar{S}_3\bar{S}_2 = CS_3(S_2 + S_1) + C\bar{S}_3\bar{S}_2$

- Es decir, si $10 \leq S \leq 19$ entonces: $F_b = 1$ si nó $F_b = 0$.

$F_c = C\bar{S}_3S_2 + CS_2\bar{S}_1 + CS_3\bar{S}_2 = C(S_3\bar{S}_2 + S_2S_1 + S_3\bar{S}_2)$

- Es decir, si $20 \leq S \leq 29$ entonces: $F_c = 1$ si nó $F_c = 0$.

$F_d = CS_3S_2S_1$

- Es decir, si $S = 30$ entonces: $F_d = 1$ si nó $F_d = 0$.

De esta forma, al presentar dos valores la función $F_i$, podemos establecer que:

- Si $F_a = 1 \Rightarrow +0 \Rightarrow +00000 \Rightarrow 00000$
- Si $F_b = 1 \Rightarrow +6 \Rightarrow +00110 \Rightarrow 00F_bF_b0$
- Si $F_c = 1 \Rightarrow +12 \Rightarrow +01100 \Rightarrow 0F_cF_c00$
- Si $F_d = 1 \Rightarrow +18 \Rightarrow +10010 \Rightarrow F_d00F_d0$

Teniendo en cuenta estas agrupaciones y que cuando un factor esté activo, el resto no estará activo, al resultado del sumador se le sumará la función lógica $F = F_dF_c(F_b + F_c)(F_b + F_d)0$. La Figura 6 muestra el esquema de bloques del sumador binario natural a BCD.

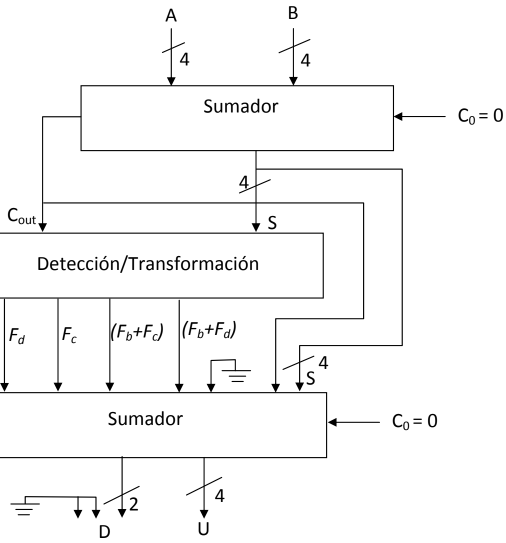



Figura 6. Sumador binario natural a BCD

El bloque Detección/Transformación, contiene las ecuaciones lógicas correspondientes a $F_dF_cF_bF_a$.