Introducción a los Sistemas Lógicos y Digitales

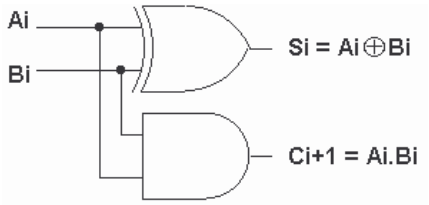
EJERCICIOS RESUELTOS

T.P. Nº 4: CIRCUITOS ARITMÉTICOS

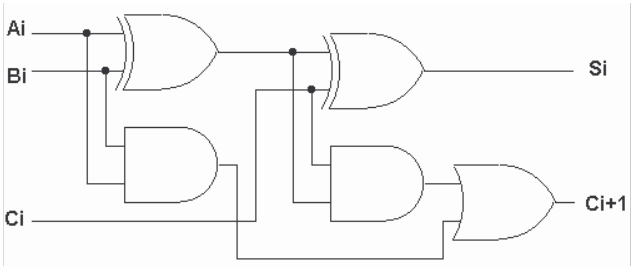
SUMADORES:

1) Cada sumador completo se puede construir usando 2 semisumadores:

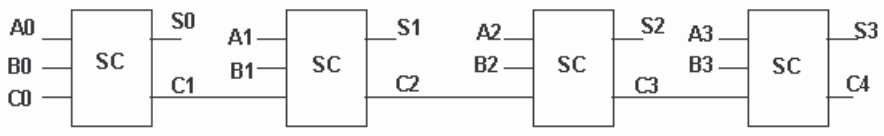
Semisumador de 1 bit:



Sumador completo (full-adder) de 1 bit:



Sumador en cascada de 4 bits con sumadores completos (SC):



Retardos:

S0 se estabiliza en 4 tpd	C1 en 4 tpd
S1 se estabiliza en 6 tpd	C2 en 6 tpd
S2 se estabiliza en 8 tpd	C3 en 8 tpd
S3 se estabiliza en 10 tpd	C4 en 10 tpd

Por lo tanto, para un sumador en cascada de n bits:

S_{n-1} y Cn se estabilizan en $(2^{n-1} + 2)$ tpd

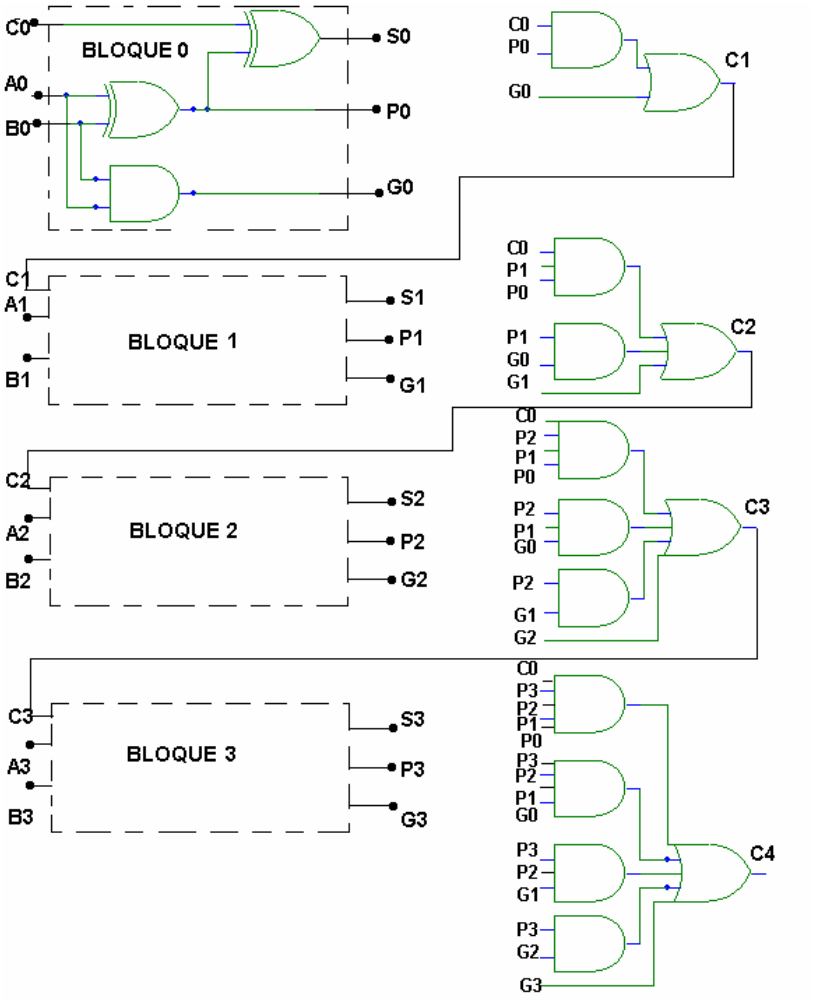
$Gi = Ai \cdot Bi \qquad Pi = Ai \oplus Bi$

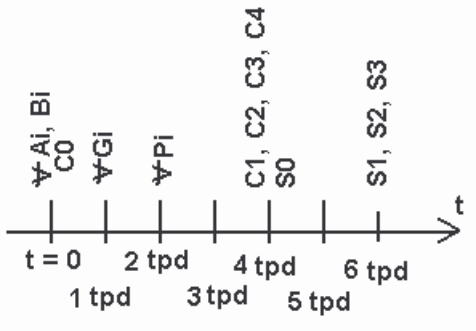
$C1 = G0 + C0\ P0$

$C2 = G1 + C1\ P1 = G1 + G0\ P1 + C0\ P0\ P1$

$C3 = G2 + C2\ P2 = G2 + G1\ P2 + G0\ P1\ P2 + C0\ P0\ P1\ P2$

$C4 = G3 + C3\ P3 = G3 + G2\ P3 + G1\ P2\ P3 + G0\ P1\ P2\ P3 + C0\ P1\ P2\ P3\ P0$



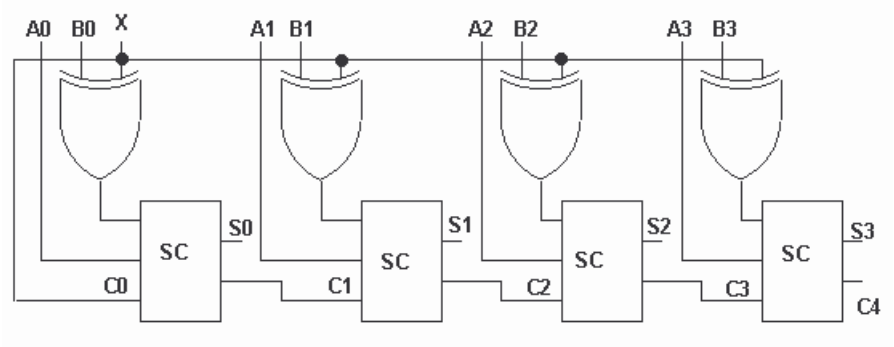


S0, C1, C2, C3 y C4 se estabilizan en 4 tpd.
S1, S2 y S3 se estabilizan en 6 tpd.

Como se ve, el sumador de arrastre anticipado es más rápido que el sumador en cascada debido a la lógica que emplea para predecir los acarreos que permite adelantar la generación de las salidas al no tener que esperar el carry de la etapa precedente para obtener las sumas respectivas.

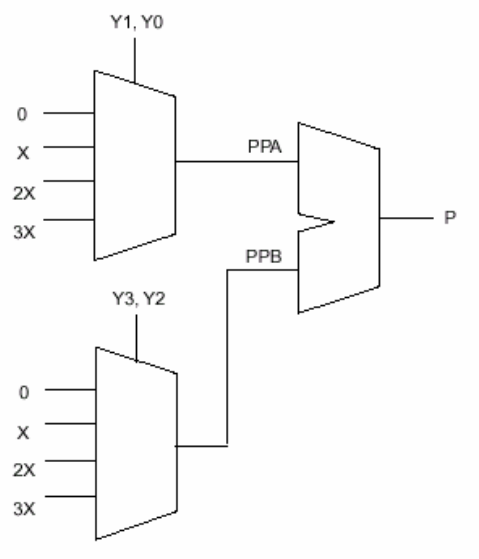
2) Sintetizar un sumador de 4 bits en complemento a 2.

Se utilizan 4 bloques sumadores completos en cascada para realizar la suma de dos números de 4 bits "A" y "B". Las entradas "Bi" pasan a través de compuertas OR exclusivo, junto con una entrada "X" común a todas. Cuando X= 0, la OR exclusivo no altera el valor de los "Bi", pero cuando X= 1 los invierte, obteniéndose el complemento a 1 del número B. Simultáneamente suma 1 a través de C0. En definitiva cuando X=1 se suma el complemento a 2 del número B.



3) b)

						X3	**X2**	**X1**	**X0**
Only 2 LSB used	>				x			**Y1**	**Y0**
if Y1=0,Y0=0	|	0	0	0	0	0	0	0	0
if Y1=0, Y0=1	|	0	0	0	0	X3	X2	X1	X0
if Y1=1, Y0=0	|	0	0	0	X3	X2	X1	X0	0
if Y1=1, Y0=1	|	0	0	0	X3	X3+X2	X2+X1	X1+X0	0
Multiplexer A result	>	0	0	PPA5	PPA4	PPA3	PPA2	PPA1	PPA0
						X3	**X2**	**X1**	**X0**
Only 2 MSB used	>				x	**Y3**	**Y2**		
if Y3=0,Y2=0	|	0	0	0	0	0	0	0	0
if Y3=0, Y2=1	|	0	0	X3	X2	X1	X0	0	0
if Y3=1, Y2=0	|	0	X3	X2	X1	X0	0	0	0
if Y2=1, Y2=1	|	0	X3	X3+X2	X2+X1	X1+X0	0	0	0
Multiplexer B result	>	PPB7	PPB6	PPB5	PPB4	PPB3	PPB2	0	0
		0	0	PPA5	PPA4	PPA3	PPA2	PPA1	PPA0
	+	PPB7	PPB6	PPB5	PPB4	PPB3	PPB2	0	0
Final product	>	P7	P6	P5	P4	P3	P2	P1	P0



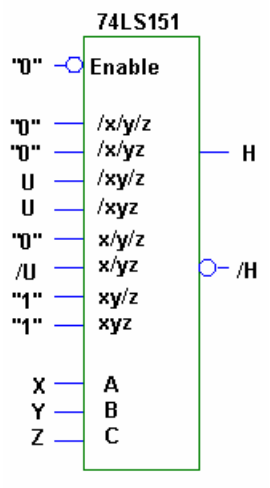
4) Implementar la función usando multiplexores:

$$H = X \cdot Y + \overline{X} \cdot Y \cdot U + X \cdot \overline{Y} \cdot Z \cdot \overline{U}$$

ZU / XY	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	1	1	1	1
10	0	0	0	1

$\overline{Z}$ Z

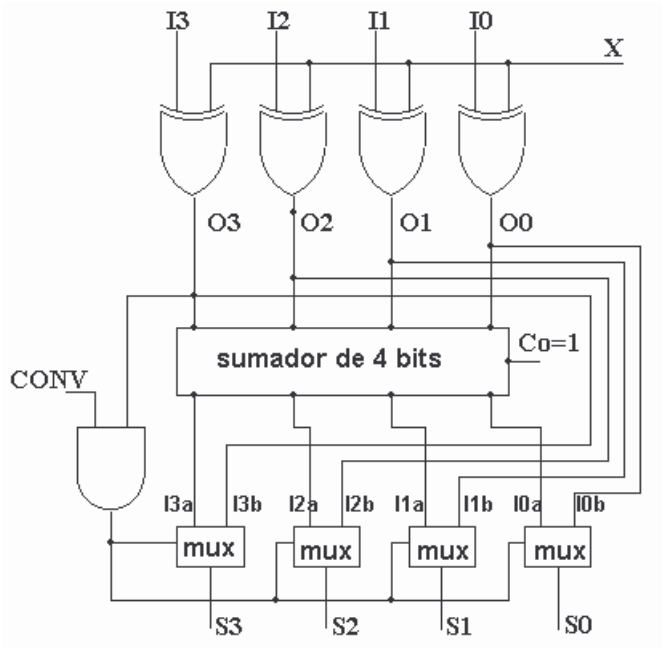
En este caso se tomaron X, Y y Z como las variables de selección del multiplexor y U como variable de residuo (cada par de celdas marcados en el Karnaugh, corresponde a una de las 8 combinaciones posibles de las variables de selección y los valores de cada celda del par definen el valor de la variable de residuo U). Por lo tanto U irá conectada a las entradas del mux (con el valor que corresponda según el diagrama de Karnaugh).



EJERCICIOS RESUELTOS ADICIONALES:

1) Diseñar un circuito en base a **sumadores y compuertas** OR-EXCLUSIVA tal que cumpla las siguientes especificaciones:

- 4 bits de entrada para un número binario en formato complemento a 1.
- 4 bits de salida.
- Entrada de control **X**. Si **X = 0** no altera los 4 bits de salida.
  Si **X = 1** hace el Ca1 del dato de entrada y lo pone a la salida.
- Entrada de control **CONV**. Si **CONV = 0** los 4 bits de salida se representan en Ca1.
  Si **CONV = 1** los 4 bits de salida se representan en Ca2.



La entrada CONV junto con la salida del OR exclusivo O3 controlan la salida según la siguiente tabla de verdad:

CONV	O3	control mux's	salidas
0	0	0	I ib
0	1	0	I ib
1	0	0	I ib
1	1	1	I ia

Nota: sólo cuando O3 es "1" el número O3 O2 O1 O0 es negativo por lo que hay que sumar el carry = "1" para obtener el complemento a 2.

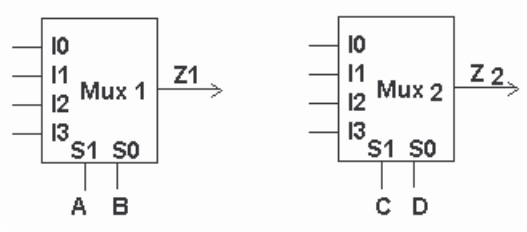
2) Implementar usando sólo 2 multiplexores 4:1 la función:

$$F = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot B \cdot \overline{C} \cdot \overline{D} + C \cdot D$$

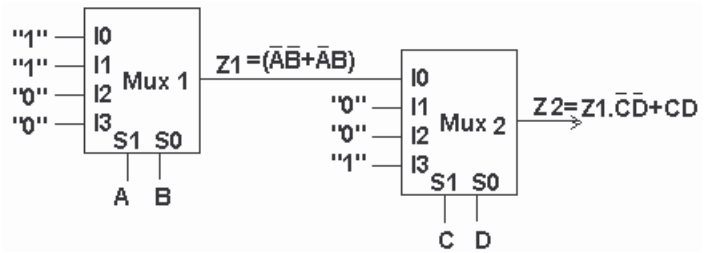
Representamos la función en un Karnaugh de 4 variables:

CD / AB	00	01	11	10
00	1		1	
01	1		1	
11			1	
10			1	

Como los mux son de 4:1, tendré 2 variables de selección:



La función F se puede escribir también como: $F = (\overline{A} \cdot \overline{B} + \overline{A} \cdot B) \cdot \overline{C} \cdot \overline{D} + C \cdot D$



donde Z2 es la función F buscada.