Andersson David Sánchez Méndez

Taller de bits, qubits y compuertas clásicas

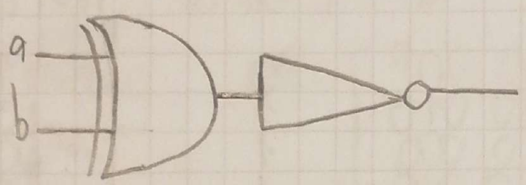
Operaciones / Identidades

1) a NXOR b (NXOR es la negación del XOR)

Tabla de verdad

a	b	a XOR b	a NXOR b
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Circuito



Matriz Para 2 entradas y 1 salida, M_{4x4}

$$\begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \end{array} \qquad NXOR \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Derivación

$$XOR = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad NOT = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

NOT al bit de salida (último bit) $\longrightarrow \mathbb{I} \otimes NOT$

$$\mathbb{I} \otimes NOT = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad NXOR = (\mathbb{I} \otimes NOT) \cdot XOR$$

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & [illegible] \\ 0 & 1 & 1 & 0 \end{bmatrix} [illegible] \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

2) (NOT P) AND (NOT Q) = NOT(P OR Q)

Tablas de verdad

Lado izquierdo (NOT P) AND (NOT Q)

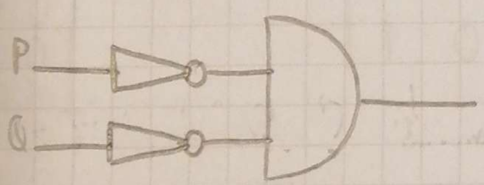
P	Q	NOT P	NOT Q	(NOT P) AND (NOT Q)
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Lado derecho NOT (P OR Q)

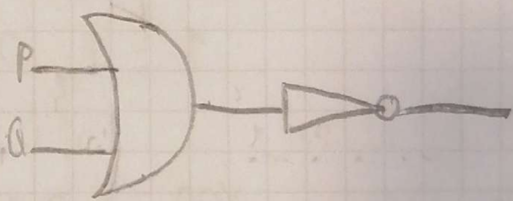
P	Q	P OR Q	NOT (P OR Q)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Circuitos

Lado izquierdo



Lado derecho



Matrices

$$\begin{array}{c|cccc} & 00 & 01 & 10 & 11 \\ \hline 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{array}$$

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

M izquierda = M derecha

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Derivaciones

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \end{bmatrix}$$

Lado izquierdo

$\downarrow$

AND $\cdot$ (NOT $\otimes$ NOT)

NOT $\otimes$ NOT : $\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$

AND : $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

Lado derecho

$\downarrow$

$(\mathbb{I} \otimes \text{NOT}) \cdot \text{OR}$

OR: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

3) P AND (Q OR R) = (P AND Q) OR (P AND R)

Tablas de verdad

Lado izquierdo P AND (Q OR R)

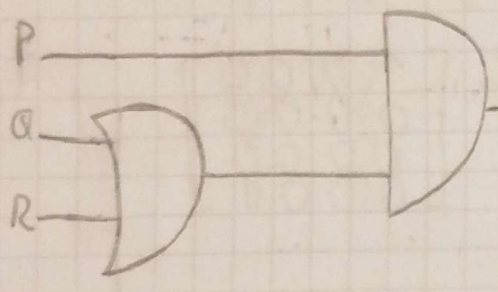
P	Q	R	Q OR R	P AND (Q OR R)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Lado derecho (P AND Q) OR (P AND R)

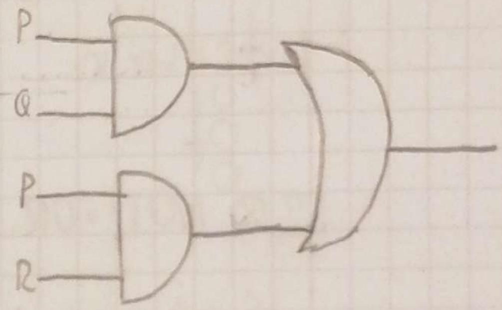
P	Q	R	P AND Q	P AND R	(P AND Q) OR (P AND R)
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

Circuitos

Lado izquierdo



Lado derecho



Matrices $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Salidas 1 están en las posiciones 101, 110, 111

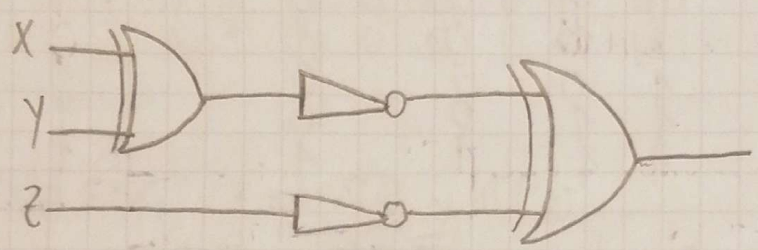
Derivaciones Mediante composición es compleja con 3 bits. Se requiere:
- Para lado izquierdo: $I \otimes$ OR seguido de AND aplicado correctamente
- Para lado derecho: dos AND en paralelo, luego OR

4) (NOT (X XOR Y)) XOR (NOT Z)

Tabla de verdad

X	Y	Z	X XOR Y	NOT(X XOR Y)	NOT Z	Result
0	0	0	0	1	1	0
0	0	1	0	1	0	1
0	1	0	1	0	1	1
0	1	1	1	0	0	0
1	0	0	1	0	1	1
1	0	1	1	0	0	0
1	1	0	0	1	1	0
1	1	1	0	1	0	1

Circuito



Matriz

$$M = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \end{bmatrix}$$

Derivación Mediante composición

Pasos a tener en cuenta
1) XOR sobre X, Y: matriz M_1
2) NOT sobre resultado: $I \otimes I \otimes NOT \cdot M_1$
3) NOT sobre Z: aplicar previamente
4) XOR final entre los dos resultados.

(X XOR Y) AND NOT Z

$$\begin{bmatrix} 1&0&0&1 \\ 0&1&1&0 \end{bmatrix} \otimes \begin{bmatrix} 0&1 \\ 1&0 \end{bmatrix}$$

$$\|$$

$$\begin{bmatrix} \begin{pmatrix} 0&1\\1&0 \end{pmatrix} & \begin{pmatrix} 0&0\\0&0 \end{pmatrix} & \begin{pmatrix} 0&0\\0&0 \end{pmatrix} & \begin{pmatrix} 0&1\\1&0 \end{pmatrix} \\ \begin{pmatrix} 0&0\\0&0 \end{pmatrix} & \begin{pmatrix} 0&1\\1&0 \end{pmatrix} & \begin{pmatrix} 0&1\\1&0 \end{pmatrix} & \begin{pmatrix} 0&0\\0&0 \end{pmatrix} \end{bmatrix} \times \begin{bmatrix} 1&1&1&0 \\ 0&0&0&1 \end{bmatrix} = \begin{bmatrix} 1&1&0&1&0&1&1&1 \\ 0&0&1&0&1&0&0&0 \end{bmatrix}$$

$$\text{NOT (X XOR Y) AND NOT Z} = \begin{bmatrix} 0&0&1&0&1&0&0&0 \\ 1&1&0&1&0&1&1&1 \end{bmatrix}$$

5) Adicionador de un bit con acarreo, es decir, tres entradas y dos salidas. Modelar solo el circuito para la suma, es decir, ignore el acarreo. Mostrar la matriz y circuito que obtuvo.

Entradas: x, y, c (carry-in)

Salidas: z (suma), c' (carry-out)

Tabla de verdad

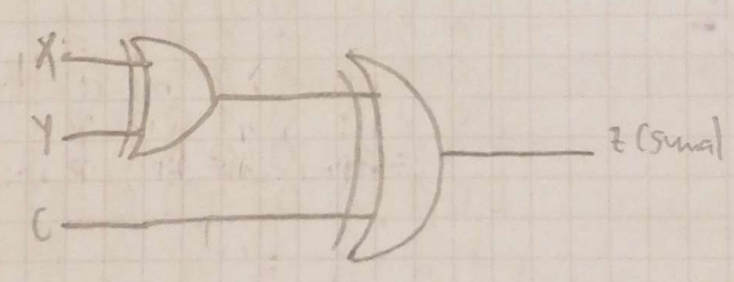
X	Y	C	z (suma)	c' (carry-out)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fórmulas $z = X \text{ XOR } y \text{ XOR } C$

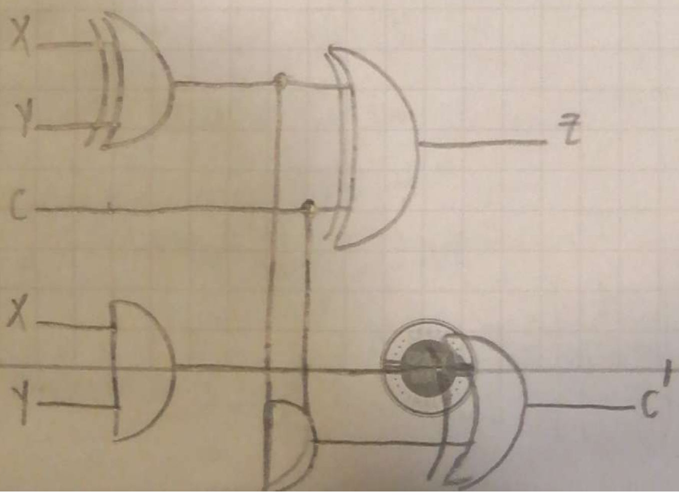
$c' = (X \text{ AND } Y) \text{ OR } (c \text{ AND } (X \text{ XOR } Y))$

Circuitos

• Suma (z)



• Completo (z y c')



Matrices

- Adicionador completo: Para 3 entradas y 2 salidas necesitamos una representación que mapee estados de 3 bits a estados de 5 bits (3 entradas + 2 salidas). Esto resulta en una matriz 32 x 8 que es compleja de derivar por composición.

- Suma (z): Ignorando el carry-out, consideramos $z = X \text{ XOR } Y \text{ XOR } C$

$$M_{suma} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Derivación Solo suma $z = X \text{ XOR } Y \text{ XOR } C$

Pasos: 1) Aplicar XOR a X y Y: Primera compuerta XOR en los dos primeros bits (dejando C intacto)

$$XOR_1 = XOR \otimes I$$

2) Aplicar XOR al resultado con C: El resultado temporal del paso 1 está en el primer bit, C en el tercero. Así que, se necesita una segunda XOR.

$$M_{suma} = XOR_2 \cdot XOR_1$$

Donde XOR_2 opera sobre el bit resultante y C