

ÁLGEBRA DE BOOLE Y CÓDIGOS NUMÉRICOS

Figura 1.1
Funciones lógicas de dos variables

A	0	0	1	1	Función
B	0	1	0	1	
F ₀	0	0	0	0	$F=0$
F ₁	0	0	0	1	$F=AB$ (AND)
F ₂	0	0	1	0	$F = A \supset B$
F ₃	0	0	1	1	$F = A$
F ₄	0	1	0	0	$F = B \supset A$
F ₅	0	1	0	1	$F = B$
F ₆	0	1	1	0	$F = A \oplus B$ (OR-exclusiva)
F ₇	0	1	1	1	$F = A+B$ (OR)
F ₈	1	0	0	0	$F = \overline{A+B} = A \downarrow B$ (NOR)
F ₉	1	0	0	1	$F = \overline{A \oplus B}$ (NOR exclusiva)
F ₁₀	1	0	1	0	$F = \overline{B}$
F ₁₁	1	0	1	1	$F = B \supset A$ (B implica A)
F ₁₂	1	1	0	0	$F = \overline{A}$
F ₁₃	1	1	0	1	$F = A \supset B$ (A implica B)
F ₁₄	1	1	1	0	$F = \overline{AB} = A \uparrow B$ (NAND)
F ₁₅	1	1	1	1	$F = 1$

Figura 1.2.
Relaciones entre las funciones lógicas de dos variables

$$\begin{array}{lll}
 F_0 = A\overline{A} = B\overline{B} = 0 & F_1 = AB & F_2 = \overline{AB} \\
 F_3 = A & F_4 = \overline{AB} & F_5 = B \\
 F_6 = \overline{AB} + \overline{A}B & F_7 = A+B & F_8 = \overline{A+B} = \overline{A} \overline{B} \\
 F_9 = \overline{A} \overline{B} + AB & F_{10} = \overline{B} & F_{11} = A+\overline{B} \\
 F_{12} = \overline{A} & F_{13} = \overline{A}+B & F_{14} = \overline{AB} = \overline{A} + \overline{B} \\
 F_{15} = A + \overline{A} = B + \overline{B} = 1 & &
 \end{array}$$

Suficiencia AND-NOT – Toda función lógica de n variables puede escribirse analíticamente como una expresión en la que aparezcan exclusivamente funciones AND y NOT.

Demostración:

$$A + B = \overline{\overline{A+B}} = \overline{\overline{A} \overline{B}}$$

Suficiencia OR-NOT – De un modo equivalente se puede demostrar la suficiencia OR-NOT.

Demostración:

$$AB = \overline{\overline{AB}} = \overline{\overline{A} + \overline{B}}$$

Y por último también existen las **suficiencias NAND y NOR**. Como conclusión, basta con ser capaces de fabricar puertas lógicas de tipo NAND o de tipo NOR para poder implementar cualquier función lógica de n variables. La demostración de las suficiencias NAND y NOR es similar a las anteriores.

Términos canónicos (mintérminos y maxtérminos) - Forma canónica de una función lógica

Objetivo – Mostrar procedimientos sistemáticos para efectuar simplificaciones y obtener funciones lógicas expresadas de la manera más sencilla posible. Obtener la expresión algebraica de una función lógica a partir de su tabla de verdad.

Término canónico – se llama término canónico de una función lógica a todo producto (mintérmino) o suma (maxtérmino) en el cual aparecen todas las variables de las que depende, ya sea en su forma directa o complementada.

Ejemplo: Z(A, B, C, D). Un mintérmino sería $\overline{A}\overline{B}\overline{C}\overline{D}$ y un maxtérmino $\overline{A} + \overline{B} + C + D$

Forma canónica de una función – Cuando una función se expresa como una suma de mintérminos o un producto de maxtérminos se dice que está expresada en su forma canónica.

Teorema – Toda función lógica puede expresarse como suma de todos los mintérminos afectados de un factor igual al valor que toma la función al sustituir cada variable por “1” o “0” lógicos según aparezca en el mintérmino en forma directa o complementada respectivamente.

Y su dual:

Toda función lógica puede expresarse como producto de todos los maxtérminos afectados por un sumando igual al valor que toma la función al sustituir cada variable por “0” o “1” según aparezca en el maxtérmino en forma directa o complementada respectivamente.

Ejemplo

Dec	A	B	C	f
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

$$F(ABC) = \overline{ABC} + \overline{A}BC + A\overline{B}C$$

$$F(ABC) = (A+B+\overline{C})(A+\overline{B}+C)$$

$$(A+\overline{B}+\overline{C})(\overline{A}+B+\overline{C})(A+\overline{B}+\overline{C})$$

Vemos que para expresar una función en suma de mintérminos, basta con coger los términos para los cuales la función vale “1” y sustituir los unos por la variable correspondiente y los ceros por la variable complementada.

Del mismo modo para expresar una función como producto de maxtérminos basta con coger los términos para los cuales la función vale “0” y sustituir los ceros por la variable correspondiente y los unos por la variable complementada.

Figura 1.3.
Diagramas de Karnaugh

Diagrama de Karnaugh para 3 variables

		C B A 2 ² 2 ¹ 2 ⁰			
A\CB	00	01	11	10	
0	0	2	6	4	
1	1	3	7	5	

Diagrama de Karnaugh para 4 variables

		D C B A 2 ³ 2 ² 2 ¹ 2 ⁰			
BA\DC	00	01	11	10	
00	0	4	12	8	
01	1	5	13	9	
11	3	7	15	11	
10	2	6	14	10	

Diagrama de Karnaugh para 5 variables

		E D C B A 2 ⁴ 2 ³ 2 ² 2 ¹ 2 ⁰			
BA\DC	00	01	11	10	
00	0	4	12	8	
01	1	5	13	9	
11	3	7	15	11	
10	2	6	14	10	

E = 0

BA\DC	00	01	11	10
00	16	20	28	24
01	17	21	29	25
11	19	23	31	27
10	18	22	30	26

E = 1

Figura 1.4.
Códigos numéricos

Tabla de equivalencias decimal-binario natural-Gray-Johnson-BCD

Decimal	Binario natural	Gray o Reflejado	Johnson	BCD
0	0000	0000	0000	0000
1	0001	0001	0001	0001
2	0010	0011	0011	0010
3	0011	0010	0111	0011
4	0100	0110	1111	0100
5	0101	0111	1110	0101
6	0110	0101	1100	0110
7	0111	0100	1000	0111
8	1000	1100		1000
9	1001	1101		1001
10	1010	1111		
11	1011	1110		
12	1100	1010		
13	1101	1011		
14	1110	1001		
15	1111	1000		

Representación de números en punto flotante

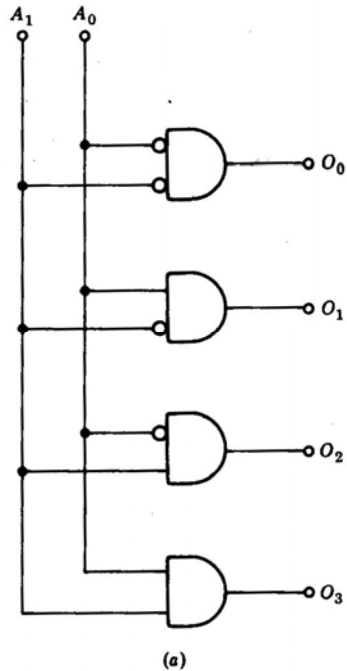
Representación de números negativos (complemento a dos)

Códigos Alfanuméricos – Código ASCII

CIRCUITOS COMBINACIONALES

Figura 2.1.
Decodificador

Circuito combinacional compuesto por n entradas y 2ⁿ salidas. Para cada combinación de entradas una sola de las salidas numerada con el subíndice correspondiente a la combinación binaria de las entradas se distingue del resto de salidas.

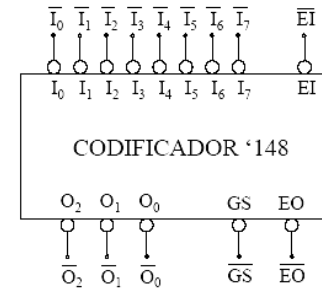


(b)

A ₁	A ₀	O ₀	O ₁	O ₂	O ₃
0	0	1			
0	1		1		
1	0			1	
1	1				1

Figura 2.2.
Codificador

Circuito combinacional compuesto por 2ⁿ entradas y n salidas. Realiza la función opuesta al decodificador. En la salida aparece la combinación binaria correspondiente a la entrada activa de mayor peso (con prioridad)

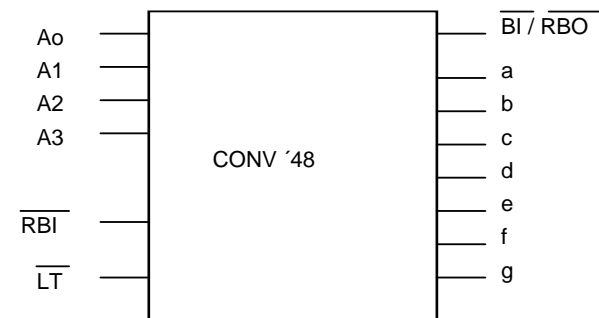


EI	I ₀	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	O ₂	O ₁	O ₀	EO	GS
0	1	1	1	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0
0	*	0	1	1	1	1	1	1	1	1	0	1	0
0	*	*	0	1	1	1	1	1	1	0	1	0	0
0	*	*	*	0	1	1	1	1	1	0	0	0	1
0	*	*	*	*	0	1	1	1	1	0	1	0	1
0	*	*	*	*	*	0	1	1	1	0	0	1	1
0	*	*	*	*	*	*	0	1	1	0	0	1	1
0	*	*	*	*	*	*	*	0	0	0	0	1	0
1	*	*	*	*	*	*	*	*	1	1	1	1	1

Figura 2.3.
Convertidor de Código

Circuito combinacional que permite traducir el número que aparece en su entrada al código numérico de la salida.

Ejemplo: convertidor 48 – convierte un número en BCD al código que utiliza un display visualizador de 7 segmentos.



A₀ A₁ A₂ A₃ - Entradas del número en BCD.
a,b,...g – Salidas hacia el display.

Figura 2.4.
Multiplexor

Circuito combiancional compuesto por 2^n entradas, n bits de selección (entradas) y una salida. Actua a modo de conmutador. En la salida aparece el valor de la entrada seleccionada por los bits de selección en cada momento.

Aplicaciones:

- Generación de funciones lógicas. Con un mux de n bits de selección se puede implementar cualquier función lógica de n variables. Si además contamos con un inversor, se puede implementar cualquier función lógica de n+1 variables.
- Acceso de varias fuentes a un único bus o linea de transmisión

Multiplexores implícitos - puertas triestado

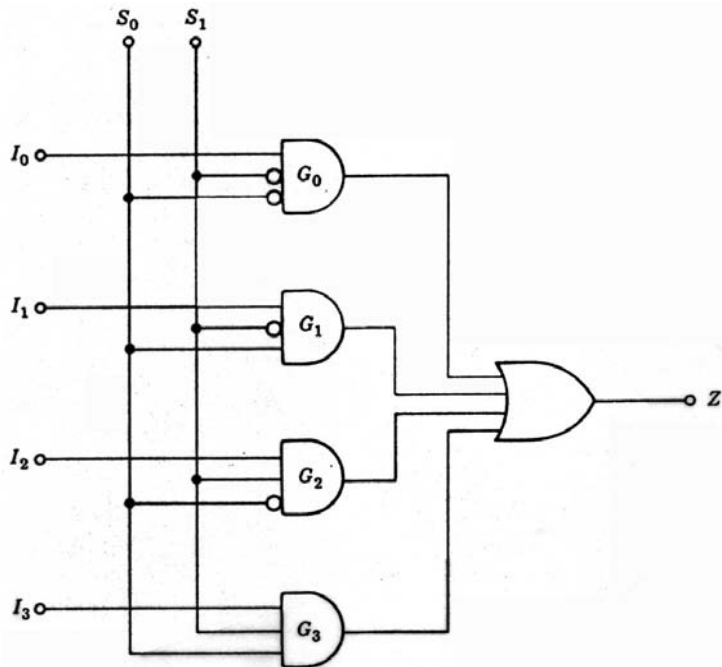


Figura 2.5
Demultiplexor

Circuito combiancional compuesto por una única entrada, 2^n salidas y n bits de selección (entradas). Realiza la función complementaria al multiplexor. En la salida seleccionada por los bits de selección aparece el mismo valor que en ese momento tiene la entrada.

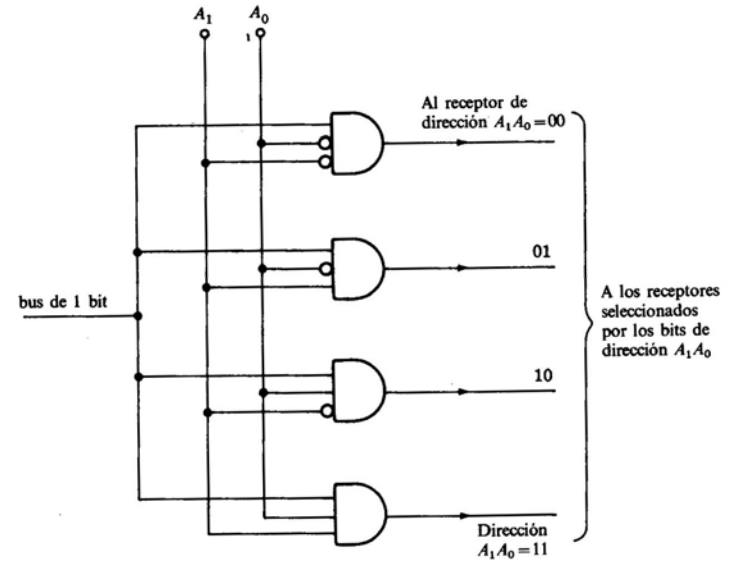
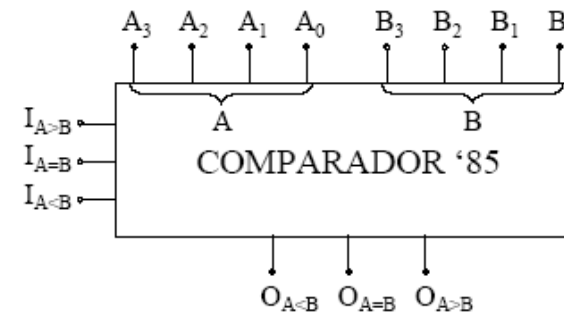


Figura 2.6.
Comparador

Circuito combiancional que compara dos números binarios e indica cuál de ellos es mayor.

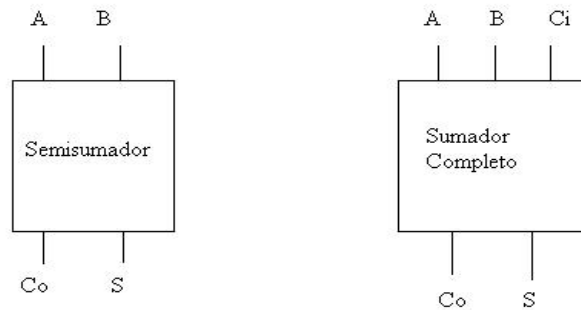
Ejemplo – Comparador '85



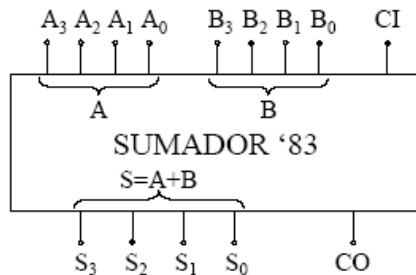
ENTRADAS		CONEXION EN CASCADA	SALIDAS
A	B	$I_{A>B} \cdot I_{A<B} \cdot I_{A=B}$	$O_{A>B} \cdot O_{A<B} \cdot O_{A=B}$
A > B		+ + +	1 0 0
A < B		+ + +	0 1 0
A = B		1 0 0	1 0 0
A = B		0 1 0	0 1 0
A = B		+ + 1	0 0 1
A = B		0 0 0	1 1 0
A = B		1 1 0	0 0 0

Figura 2.7.
Sumador

Circuito combinacional cuya salida es la suma algebraica de los dos numeros binarios en sus entradas.



Ejemplo – sumador ‘83



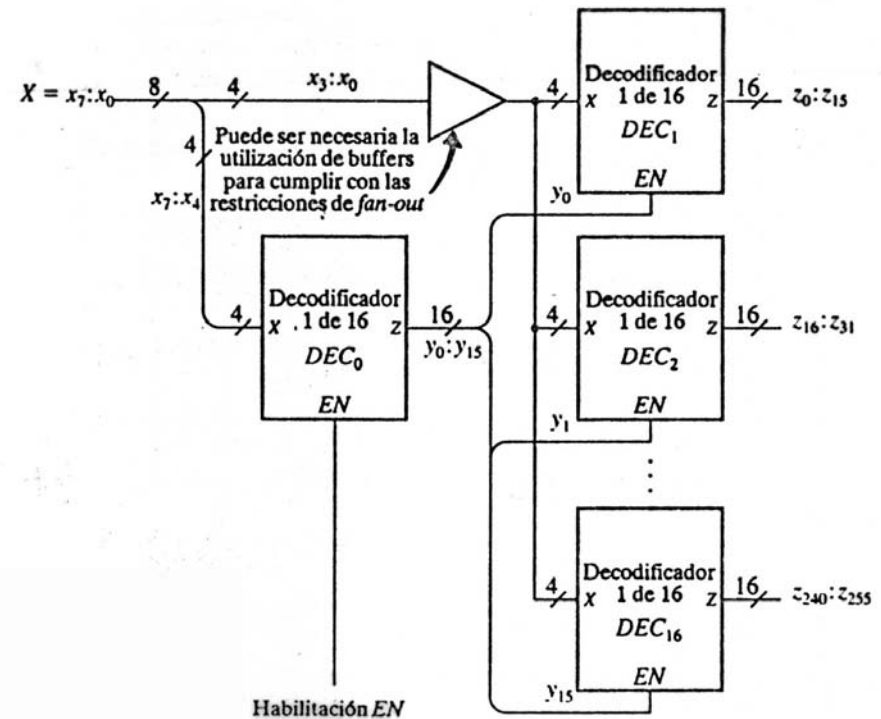
A_3, A_2, A_1, A_0 : Entrada A.
 B_3, B_2, B_1, B_0 : Entrada B.
 CI: Acarreo de entrada.
 S_3, S_2, S_1, S_0 : Salida S (suma aritmética de las entradas).
 CO: Acarreo de salida.

Figura 2.8.

Números (4 Bits) En Representación Complemento A Dos (con signo)

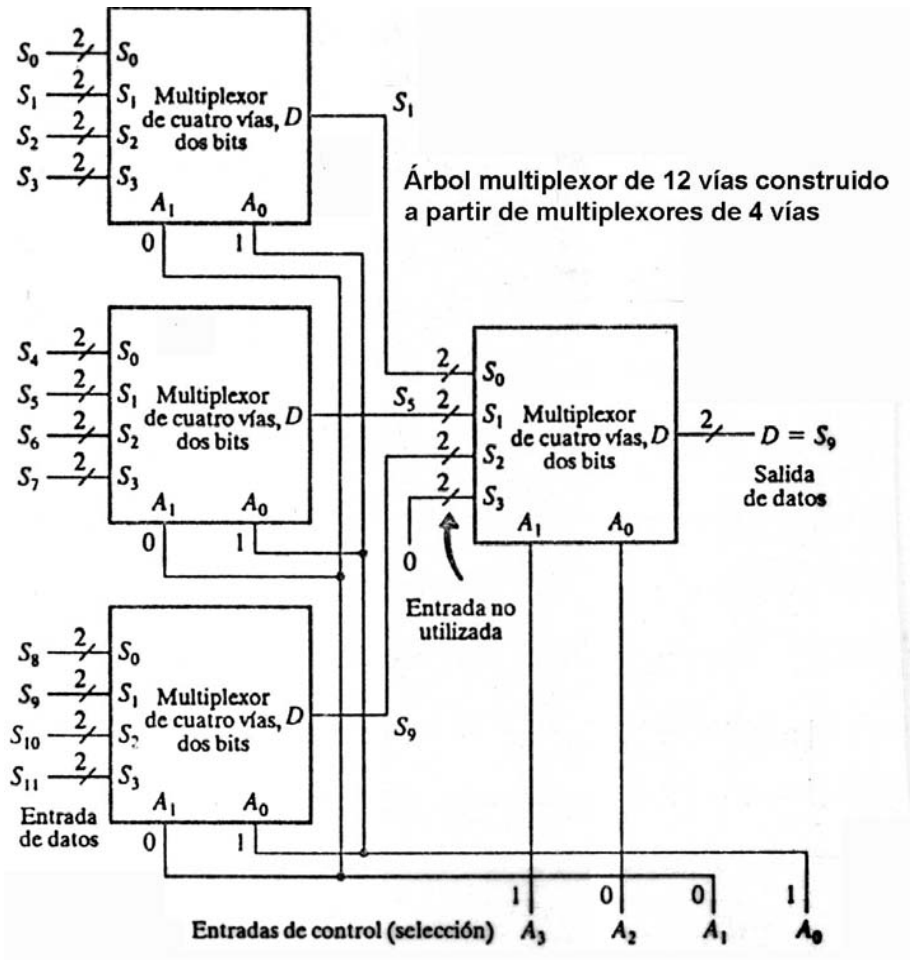
0: 0 0 0 0	-1: 1 1 1 1
1: 0 0 0 1	-2: 1 1 1 0
2: 0 0 1 0	-3: 1 1 0 1
3: 0 0 1 1	-4: 1 1 0 0
4: 0 1 0 0	-5: 1 0 1 1
5: 0 1 0 1	-6: 1 0 1 0
6: 0 1 1 0	-7: 1 0 0 1
7: 0 1 1 1	-8: 1 0 0 0

Figura 2.9.
Decodificador de 256 líneas



Árbol decodificador 1 de 256, construido a partir de decodificadores 1 de 16

Figura 2.10.
Multiplexor de 12 vías de 2 bits



LATCHES Y FLIP-FLOPS: REGISTROS

Figura 3.1.
Esquema lógico de un biestable

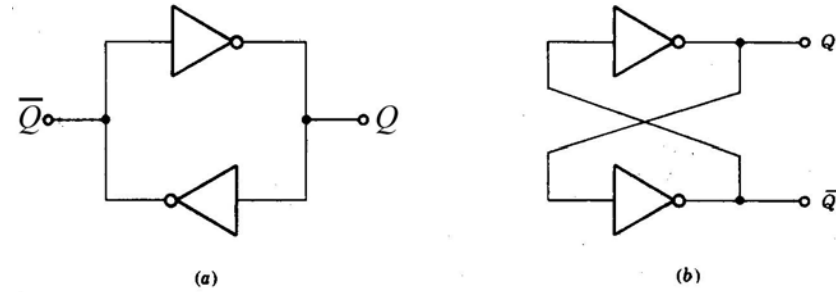


Figura 3.2.
Esquema lógico de un biestable R-S o cerrojo estático

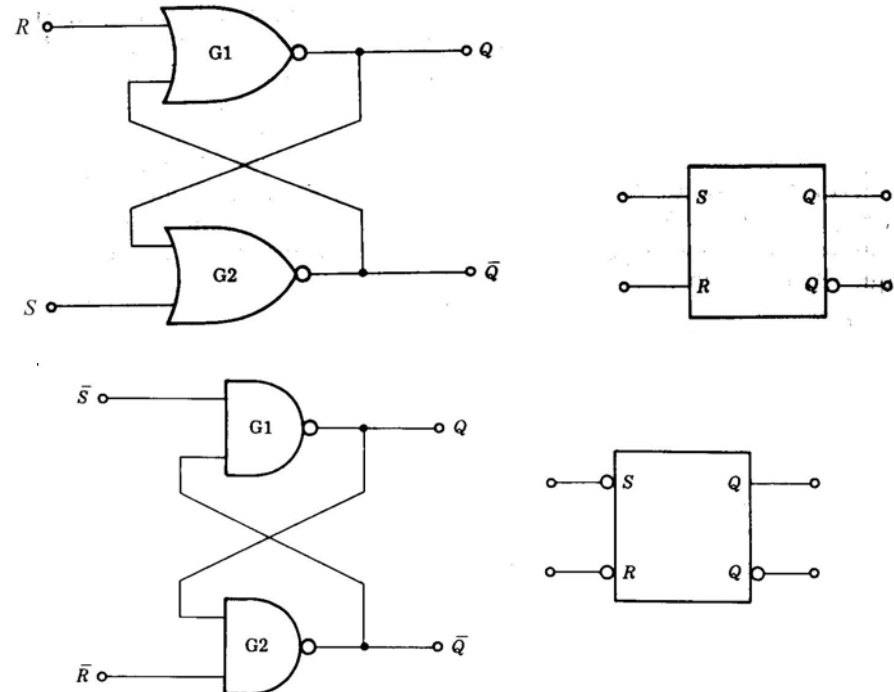


Figura 3.3.
Esquema lógico del cerrojo dinámico tipo R-S

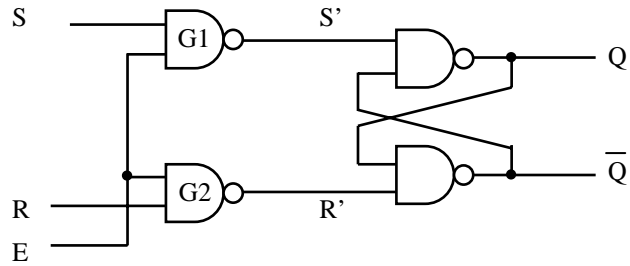


Figura 3.4.
Esquema lógico del cerrojo dinámico tipo D

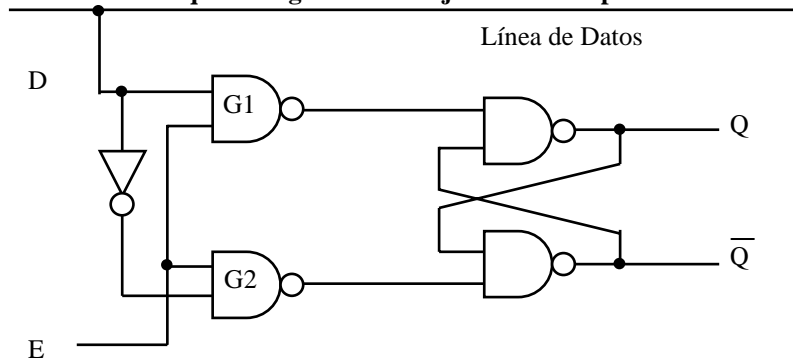


Figura 3.5.
Señales ilustrativas de la operación de un cerrojo transparente

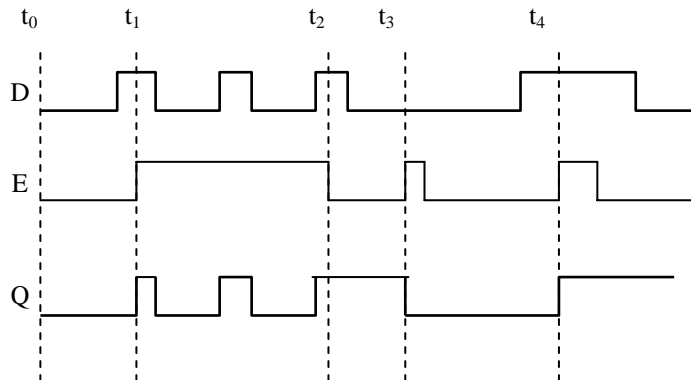


Figura 3.6.
Flip-Flor R-S Maestro-Esclavo

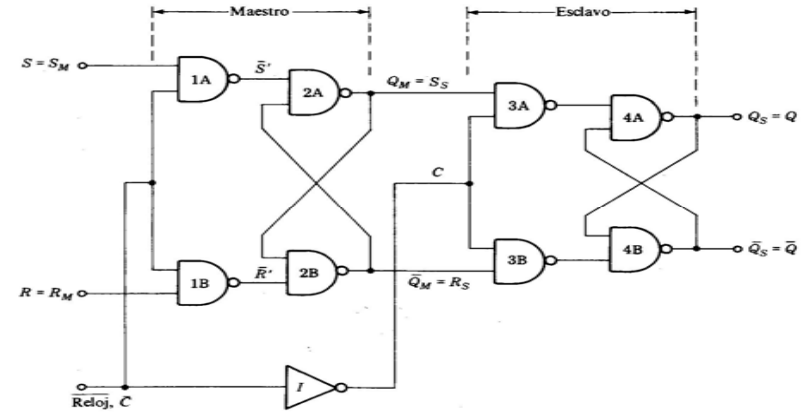


Figura 3.7.
Diagrama lógico simplificado de un flip-flop R-S disparado por flanco positivo

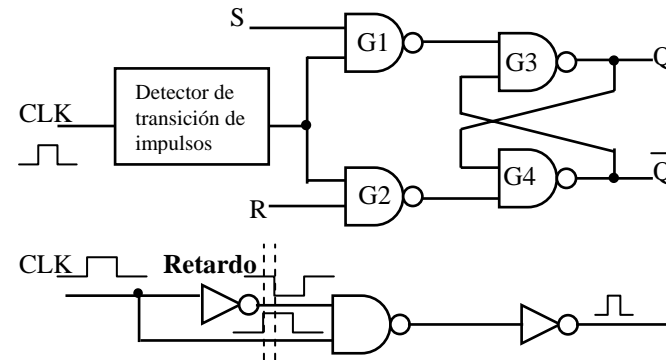


Figura 3.8.
Tipos de flip-flops básicos

R^n	S^n	Q^{n+1}	\bar{Q}^{n+1}
0	0	Q^n	\bar{Q}^n
0	1	1	0
1	0	0	1
1	1	No util.	

J^n	K^n	Q^{n+1}	\bar{Q}^{n+1}
0	0	Q^n	\bar{Q}^n
0	1	0	1
1	0	1	0
1	1	Q^n	\bar{Q}^n

D^n	Q^{n+1}	\bar{Q}^{n+1}
0	0	1
1	1	0

T^n	Q^{n+1}	\bar{Q}^{n+1}
0	Q^n	\bar{Q}^n
1	\bar{Q}^n	Q^n

Figura 3.9.

Tabla de especificaciones del integrado 74LS107 (flip-flop activados por flanco de bajada con entrada directa asínrona de Reset) (Texas Instruments 1988b)

Recommended operating conditions

	SN54LS107A			SN54LS107A			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V _{CC} , Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V _{IH} , High-level input voltage	2			2			V
V _{IL} , Low-level input voltage	0.7			0.8			V
I _{OH} , High-level output current	-0.4			-0.4			mA
I _{OL} , Low-level output current	4			8			mA
f _{clock} , Clock frequency	0		30	0		30	MHz
t _w , Pulse duration	CLK low	20		20		ns	
	CLR low	25		25			
t _{su} , Setup time before CLK ↓	data high or low	20		20		ns	
	CLR inactive	25		25			
t _h , Hold time-data after CLK ↓	0		0	0		ns	
Operating free-air temperature, T _A	-55		125	0		70	°C

switching characteristics, V_{CC} = 5V, T_A = 25°C

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
f _{max}			C _L = 15 pF, R _L = 2 kΩ	30	45		MHz
t _{PLH}		Q or \bar{Q}		15	20		ns
t _{PHL}	CLR or CLK	Q or \bar{Q}		15	20		ns

Figura 3.10.

Registros de desplazamiento

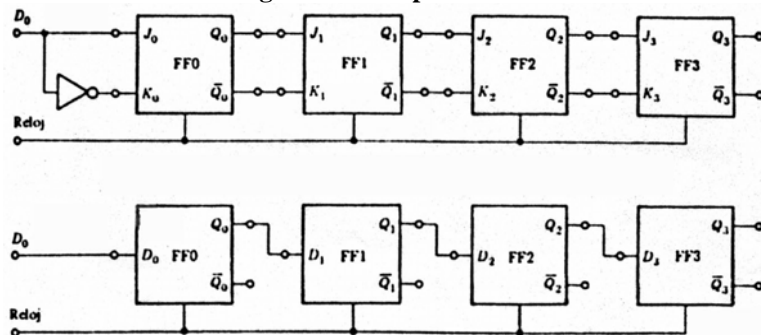


Figura 3.11.

Celda i de un registro que realiza las 5 operaciones que aparecen en la tabla activando en cada caso el terminal de control correspondiente

Órdenes a los cuales responde registro

Orden	Simbolo
1. Escribe en el registro la palabra del bus	W
2. Lee en el bus la palabra del registro	R
3. Incrementa el registro	I
4. Complementa el registro	C
5. Borra el registro para que sean cero todas las Q	Z

