Álgebra Booleana y Diseño Lógico

Circuitos Digitales,

2º de Ingeniero de Telecomunicación.

EITE — ULPGC.

Índice

Propiedades Algebraicas

- Un conjunto es una colección de objetos con una propiedad común o varias
- *x* es miembro de *S* :

$$x \in S$$

- **Axioma**: propiedad que se asume como cierta sin necesidad de probarse

Propiedades Algebraicas

- **Cierre**: Un conjunto *S* se dice cerrado respecto a la operación • si, y sólo si,

$$\forall x, y \in S \quad x \bullet y \in S$$

- **Elemento identidad**: *e* es el elemento identidad respecto a la operación • definida en *S* si

$$\forall x \in S \quad e \bullet x = x \bullet e = x$$

Propiedades Algebraicas

- **Conmutativa**: Una operación • es conmutativa si, y sólo si,

$$\forall x, y \in S \quad x \bullet y = y \bullet x \in S$$

- **Elemento inverso**: en un conjunto S existe el elemento inverso si

$$\forall x \in S \quad \exists y \in S / x \bullet y = e$$

En este caso, y es el elemento inverso de x

Propiedades Algebraicas

- **Distributiva**: Siendo • y □ operadores en *S*, • es distributiva respecto a □ si, y sólo si,

$$\forall x, y, z \in S \quad x \bullet (y \square z) = (x \bullet y) \square (x \bullet z)$$

Definición axiomática de Álgebra booleana

- Un conjunto *B* con las operaciones + y · es álgebra booleana si cumple las siguientes propiedades:
  - **Cierre**
    - B es cerrado respecto a +
    - B es cerrado respecto a ·
  - **Elemento identidad**
    - Existe elemento identidad para + (0)
    - Existe elemento identidad para · (1)

Definición axiomática de Álgebra booleana

- **Propiedad conmutativa**
  - + es conmutativa
  - · es conmutativa
- **Propiedad distributiva**
  - · es distributiva respecto a +
  - + es distributiva respecto a ·
- **Elemento complemento**

$$\forall x \in B, \exists x' \in B / \begin{cases} (1) & x + x' = 1 \\ (2) & x \cdot x' = 0 \end{cases}$$

Definición axiomática de Álgebra booleana

- **Cardinalidad acotada**
  - Al menos existen dos elementos, x e y, tales que $x \neq y$

Definición axiomática de Álgebra booleana

- Se define de forma implícita la operación de complemento (x' ó $\overline{x}$)
  - También se llama negación o inversión

Definición axiomática de Álgebra booleana

- **Diferencias con el álgebra *ordinaria***
  - En el *álgebra ordinaria* la + no es distributiva con respecto a la ·
  - El álgebra booleana no tiene inversos para las operaciones + y ·
    - No existen operaciones equivalentes a la resta y la división
  - Existe el complemento en el álgebra booleana pero no en el ordinaria

Definición axiomática de Álgebra booleana

- **Diferencias con el álgebra *ordinaria***
  - El álgebra booleana se aplica a un conjunto finito de elementos: el ordinario a un conjunto infinito
  - No se incluye la asociatividad como axioma en el álgebra de Boole: se puede derivar de los establecidos.

Álgebra booleana bivaluada (con dos valores)

- *B* posee dos elementos: 0 y 1
- Tiene dos operadores básicos: la *y lógica (AND o producto lógico)* y la *o lógica (OR o suma lógica)*

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

Álgebra booleana bivaluada (con dos valores)

- Cumple los axiomas para ser un álgebra de Boole (axiomas de Hungtington)
  - **Cierre**: tanto la *o lógica* como la *y lógica* dan como resultado un elemento de *B*
  - **Elemento identidad:**
    - **0 para + y 1 para ·**
      - $0 + 0 = 0$ y $1 + 0 = 0 + 1 = 1$
      - $1 \cdot 1 = 1$ y $1 \cdot 0 = 0 \cdot 1 = 0$

Álgebra booleana bivaluada (con dos valores)

- **Propiedad distributiva**
  - La operación + es distributiva respecto a la ·
  - La operación · es distributiva respecto a la +
- **Elemento complemento**

$$\forall x \in B, \exists x' \in B / \begin{cases} (1) & x + x' = 1 \\ (2) & x \cdot x' = 0 \end{cases}$$

Álgebra booleana bivaluada (con dos valores)

- **Propiedad conmutativa**
  - La operación + es conmutativa en B
  - La operación · es conmutativa en B
  - Se puede deducir a partir de la simetría de las tablas que definen + y ·
    - En +, si cualquiera de los operandos es 1, el resultado es 1
    - En ·, si cualquiera de los operandos es 0, el resultado es 0

Álgebra booleana bivaluada (con dos valores)

- **Propiedad distributiva**

x	y	z	$y+z$	$x(y+z)$	xy	xz	$(xy)+(xz)$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Prueba de que $x(y+z) = (x\,y) + (x\,z)$

Álgebra booleana bivaluada (con dos valores)

- **Propiedad distributiva**

x	y	z	yz	$x+(yz)$	$x+y$	$x+z$	$(x+y)(x+z)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Prueba de que $x + (y\,z) = (x + y)\,(x + z)$

Álgebra booleana bivaluada (con dos valores)

- **Complemento**
  - El 0 y el 1 son complementos el uno del otro:
    - $0 + 0' = 0 + 1 = 1$ y
      $1 + 1' = 1 + 0 = 1$
    - $0 \cdot 0' = 0 \cdot 1 = 0$ y
      $1 \cdot 1' = 1 \cdot 0 = 0$
- **Cardinalidad acotada**
  - Existen al menos dos elementos, representados por 0 y 1, tales que $0 \neq 1$

Procedimiento para evaluación de operadores

- El orden de prioridades para evaluar los operadores es:
  - ( )
  - NOT
  - AND
  - OR

Ejemplo: $(x + x\,y)'$ para x = 0 e y = 1:

$$(0 + 0 \cdot 1)' = (0 + 0)' = (0)' = 0' = 1$$

Principio de Dualidad

- Si una expresión es válida en el Álgebra booleana, su dual también lo es
  - La expresión dual de una se obtiene cambiando en una expresión...
    - AND por OR
    - OR por AND
    - 0 por 1
    - 1 por 0

Principio de Dualidad

Si $x + 1 = 1$ Si $x + x' = 1$

$x \cdot 0 = 0$ $x \cdot x' = 0$

Demostraciones de teoremas del álgebra booleana

- Propiedad de la Idempotencia

$$x + x = x$$

$$\begin{aligned} x+x &= (x+x)\cdot 1 && \text{por la propiedad de Identidad} \\ &= (x+x)\cdot(x+x') && \text{por la propiedad de Complemento} \\ &= x+x\cdot x' && \text{por la propiedad Distributiva} \\ &= x+0 && \text{por la propiedad de Identidad} \\ &= x && \text{por la propiedad de Complemento} \end{aligned}$$

  - Por dualidad, $x \cdot x = x$

Demostraciones de teoremas del álgebra booleana

- Ley de D'Morgan:

  $$(x+y)' = x'y'$$

x	y	$x+y$	$(x+y)'$	x'	y'	$x'y'$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

  - Su dual, $(xy)' = x' + y'$

Funciones booleanas

- Se puede definir como:
  - Una expresión algebraica formada con variables binarias y las funciones AND, OR y NOT

$$F_1 = xy + xy'z + x'yz$$

Funciones booleanas

- Significado de una expresión booleana:
  - *Sólo* hay que leerla

$$F_1 = xy + xy'z + x'yz$$

F_1 vale 1 cuando x vale 1 e y vale 1 o cuando x vale 1, y vale 0 y z vale 1 o cuando x vale 0, y vale 1 y z vale 1

Funciones booleanas

- Se puede definir como:
  - Una tabla de verdad que indica el valor de la función para todas y cada una de las combinaciones de los valores de las variables binarias que forman parte de la función

Variables

Valores de las variables

x	y	z	F_1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Funciones booleanas

- Complemento de una función
  - Si está definida con una tabla de verdad, se obtiene cambiando los unos (1) por ceros (0) y los ceros por unos

x	y	z	F_1	F_1'
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

Funciones booleanas

- Complemento de una función
  - Si está definida de forma algebraica, se aplican los Teoremas de D'Morgan

$$
\begin{aligned}
F_1' &= (xy + xy'z + x'yz)' \\
&= (xy)'(xy'z)'(x'yz)' \\
&= (x' + y')(x' + y + z')(x + y' + z')
\end{aligned}
$$

Funciones booleanas

- Teorema de D'Morgan generalizado:

$$\text{Siendo } F = \text{expr}(x_0, x_1, \ldots, x_{n-1})$$
$$\longrightarrow F' = \text{expr}_d(x'_0, x'_1, \ldots, x'_{n-1})$$

Equivalencia de expresiones

- Por manipulaciones algebraicas

$$
\begin{aligned}
xy+xy'z+x'yz &= xy+xyz+xy'z+x'yz && \text{por absorción}\\
&= xy+x(y+y')z+x'yz && \text{por prop. distributiva}\\
&= xy+x1z+x'yz && \text{por prop. del complemento}\\
&= xy+xz+x'yz && \text{por prop. del elemento identidad}\\
&= xy+xyz+xz+x'yz && \text{por absorción}\\
&= xy+xz+(x+x')yz && \text{por prop. distributiva}\\
&= xy+xz+1yz && \text{por prop. del complemento}\\
&= xy+xz+yz && \text{por prop. del elemento identidad}
\end{aligned}
$$

Minterms

- Son funciones que valen 1 para una única combinación de valores de sus variables
- Su expresión algebraica es un producto donde aparecen todas las variables

Minterms

x	y	z	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Minterms

x	y	z	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Minterms

$$m_0 = x'y'z' \qquad m_4 = xy'z'$$

$$m_1 = x'y'z \qquad m_5 = xy'z$$

$$m_2 = x'yz' \qquad m_6 = xyz'$$

$$m_3 = x'yz \qquad m_7 = xyz$$

Maxterms

- Son funciones que valen 0 para una única combinación de valores de sus variables
- Su expresión algebraica es una suma donde aparecen todas las variables

Maxterms

x	y	z	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

Maxterms

x	y	z	m_3	M_3
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

$$\begin{aligned} M_3 &= m_3' \\ &= (x'yz)' \\ &= x + y' + z' \end{aligned}$$

$$M_i = m_i'$$

Formas canónicas

- Son expresiones del tipo suma de productos o producto de sumas
  - En la forma canónica de suma de productos...
    - en cada uno de los términos producto que se suman aparecen todas las variables de la función: es la expresión de un minterm
  - En la forma canónica de producto de sumas...
    - en cada uno de los términos suma que se multiplican aparecen todas las variables de la función: es la expresión de un maxterm
  - En ningún caso aparecen términos repetidos

Formas canónicas

- Los minterms que aparecen en la forma canónica de una función se llaman "minterms 1 de la función"
- Los maxterms que aparecen en la forma canónica de una función se llaman "maxterms 0 de la función"
- Las formas canónicas son expresiones *únicas* de la función

Formas canónicas

- Los minterms 1 de una función son los que habría que sumar para construir la función

x	y	z	F_1	m_3	m_5	m_6	m_7
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	1	1	1	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	0	1	0	0
1	1	0	1	0	0	1	0
1	1	1	1	0	0	0	1

Formas canónicas

- La expresión de un minterm

Siendo $i = b_{n-1} \ldots b_0$ y representando b_j el valor de la variable x_j para un minterm de las variables $x_{n-1}, \ldots, x_0$, la expresión algebraica del minterm es

$$m_i(x_{n-1}, \ldots, x_0) = y_{n-1} \ldots y_0$$

siendo

$$y_k = \begin{cases} x_k & \text{si } b_k = 1 \\ x'_k & \text{si } b_k = 0 \end{cases}$$

Formas canónicas

- Por tanto...

x	y	z	F_1	m_3	m_5	m_6	m_7
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	1	1	1	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	0	1	0	0
1	1	0	1	0	0	1	0
1	1	1	1	0	0	0	1

$$\begin{aligned} F_1 &= m_3 + m_5 + m_6 + m_7 \\ &= x'yz + xy'z + xyz' + xyz \end{aligned}$$

Formas canónicas

- Los maxterms 0 de una función son los que habría que multiplicar para construir la función
- Se puede ver que sus índices corresponden a los minterms que no están entre los minterms 1 de la función

Formas canónicas

x	y	z	F_1	M_0	M_1	M_2	M_4
0	0	0	0	0	1	1	1
0	0	1	0	1	0	1	1
0	1	0	0	1	1	0	1
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	0
1	0	1	1	1	1	1	1
1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1

$$\begin{aligned} F_1 &= M_0 M_1 M_2 M_4 \\ &= (x+y+z)(x+y+z')(x+y'+z)(x'+y+z) \end{aligned}$$

Formas canónicas

- ¿Cómo obtener una forma canónica de una función?
  - A partir de la tabla de verdad, o
  - Mediante manipulaciones algebraicas

Formas canónicas

- ¿Cómo pasar de una forma canónica a otra de una función?
  - Para realizarse de forma más *cómoda* se utiliza otra notación para definir las funciones:

$$
\begin{aligned}
F &= \sum(\text{índices de los minterms 1 de la función}) \\
&= \prod(\text{índices de los maxterms 0 de la función})
\end{aligned}
$$

Formas canónicas

- Los índices que aparecen en una forma son los que faltan en la otra

x	y	z	F_2
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$\begin{aligned} F_2(x,y,z) &= \sum(0,1,4,7) \\ &= \prod(2,3,5,6) \end{aligned}$$

$$\begin{aligned} F_2(x,y,z) &= x'y'z' + x'y'z + xy'z' + xyz \\ &= (x+y'+z)(x+y'+z')(x'+y+z')(x'+y'+z) \end{aligned}$$

Formas canónicas

- Obtención de formas canónicas mediante manipulaciones algebraicas
  - La idea es ingeniárselas para que en los distintos términos que se sumen (o multipliquen) aparezcan todas las variables

Formas canónicas

$$
\begin{aligned}
F_3 &= x + yz \\
&= x(y + y')(z + z') + (x + x')yz \\
&= xy(z + z') + xy'(z + z') + xyz + x'yz \\
&= xyz + xyz' + xy'z + xy'z' + xyz + x'yz \\
&= x'yz + xy'z' + xy'z + xyz' + xyz
\end{aligned}
$$

Formas canónicas

$$
\begin{aligned}
F_3 &= x + yz \\
&= (x+y)(x+z) \\
&= (x+y+zz')(x+yy'+z) \\
&= (x+y+z)(x+y+z')(x+y+z)(x+y'+z) \\
&= (x+y'+z)(x+y+z')(x+y+z)
\end{aligned}
$$

Formas canónicas

- Se propone:
  - Obtener la tabla de verdad de la función anterior y obtener las formas canónicas a partir de dicha tabla
  - ¿Qué debería dar?

Formas normalizadas

- Son *formas* que responden al esquema de *suma de productos* o *producto de sumas*
  - Suelen tener menor número de operaciones que las formas canónicas
    - Para una función algebraica concreta, es de menos operaciones siguiendo esos mismos esquemas
  - Pueden existir varias formas normalizadas para una misma función

Formas normalizadas

- Literal: unidad que se refiere a una variable o su invertida
- Suma de productos:
  - Es una suma de distintos términos, donde en todos ellos se realiza *exclusivamente* el producto de distintos literales

$$F_1 = xy + x'yz + xy'z$$

Formas normalizadas

- Producto de sumas:
  - Es un producto de distintos términos, donde en cada uno de ellos se realiza *exclusivamente* la suma de distintos literales

$$F_2 = (x' + y')(x + y' + z')(x' + y + z')$$

Formas normalizadas

- Se pueden obtener...
  - Tomando como referencia una forma canónica y combinando términos que se distingan en un único literal

$$
\begin{aligned}
F_1 &= xyz + xyz' + xy'z + x'yz \\
&= xyz + xyz' + xyz + xy'z + xyz + x'yz \\
&= xy(z + z') + x(y + y')z + (x + x')yz \\
&= xy + xz + yz
\end{aligned}
$$

Formas no normalizadas

- Las que se suelen emplear son las derivadas de las normalizadas realizando *factorización*
  - Generalmente necesitan menos operaciones que las normalizadas

$$
\begin{aligned}
F_1 &= xy + xz + yz \\
&= x(y+z) + yz \\
&= xy + z(x+y) \\
&= (x+z)y + xz
\end{aligned}
$$

Operaciones lógicas binarias

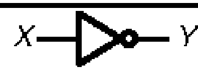
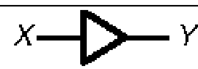
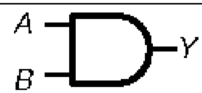
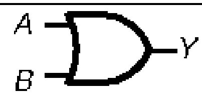
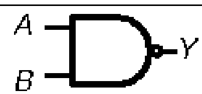
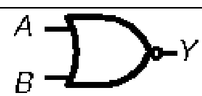
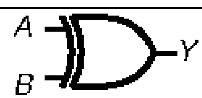
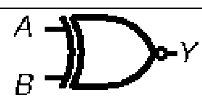
- Para *n* variables binarias existen 2^{2^n} funciones booleanas posibles
- Para 2 variables binarias existen, por tanto, 16 funciones booleanas posibles

Operaciones lógicas binarias

Nombre	Símbolo del operador	Valores x, y funcionales 00	01	10	11	Expresiones algebraicas	Comentarios
Cero		0	0	0	0	$F_0 = 0$	Constante binaria 0
AND	$x \cdot y$	0	0	0	1	$F_1 = xy$	x e y, función *y lógica*
Inhibición	x/y	0	0	1	0	$F_2 = xy'$	x pero no y
Transferencia		0	0	1	1	$F_3 = x$	x
Inhibición	y/x	0	1	0	0	$F_4 = x'y$	y pero no x
Transferencia		0	1	0	1	$F_5 = y$	y
XOR	$x \oplus y$	0	1	1	0	$F_6 = xy' + x'y$	x o y pero no ambos
OR	$x + y$	0	1	1	1	$F_7 = x + y$	x o y, función *o lógica*
NOR	$x \downarrow y$	1	0	0	0	$F_8 = (x + y)'$	OR negada (Not-OR)
Equivalencia	$x \odot y$	1	0	0	1	$F_9 = xy + x'y'$	x igual a y, XOR negada, XNOR
Complemento	y'	1	0	1	0	$F_{10} = y'$	Función *no*, negación, *not* y
Implicación	$x \subset y$	1	0	1	1	$F_{11} = x + y'$	Si y entonces x
Complemento	x'	1	1	0	0	$F_{12} = x'$	Función *no*, negación, *not* x
Implicación	$x \supset y$	1	1	0	1	$F_{13} = x' + y$	Si x entonces y
NAND	$x \uparrow y$	1	1	1	0	$F_{14} = (xy)'$	AND negada (Not-AND)
Uno		1	1	1	1	$F_{15} = 1$	Constante binaria 1

Puertas lógicas digitales

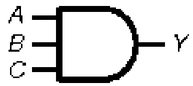
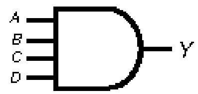
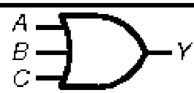
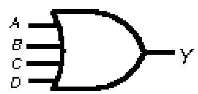
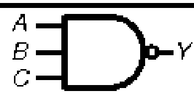
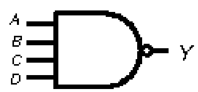
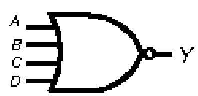
Biblioteca Lógica Básica

Nombre	Símbolo gráfico	Expresión funcional	Coste en número de transistores	Retardo de la puerta en ns
Inversor		$F = x'$	2	1
Driver o *buffer*		$F = x$	4	2
AND		$F = xy$	6	2.4
OR		$F = x + y$	6	2.4
NAND		$F = (xy)'$	4	1.4
NOR		$F = (x + y)'$	4	1.4
XOR		$F = x \oplus y$	14	4.2
XNOR		$F = x \odot y$	12	3.2

(Implementaciones en tecnología CMOS)

Puertas lógicas digitales

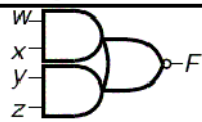
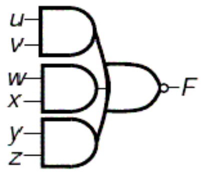
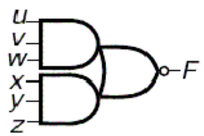
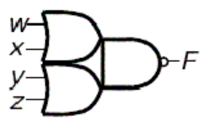
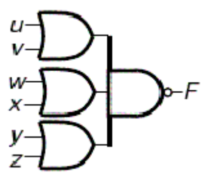
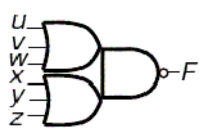
Puertas lógicas de varias entradas

Nombre	Símbolo gráfico	Expresión funcional	Coste en número de transistores	Retardo de la puerta en ns
AND de 3 entradas		$F = xyz$	8	2.8
AND de 4 entradas		$F = xyzw$	10	3.2
OR de 3 entradas		$F = x+y+z$	8	2.8
OR de 4 entradas		$F = x+y+z+w$	10	3.2
NAND de 3 entradas		$F = (xyz)'$	8	1.8
NAND de 4 entradas		$F = (xyzw)'$	10	2.2
NOR de 3 entradas		$F = (x+y+z)'$	8	1.8
NOR de 4 entradas		$F = (x+y+z+w)'$	10	2.2

(Implementaciones en tecnología CMOS)

Puertas lógicas digitales

Puertas lógicas normalizadas multi-operador

Nombre	Símbolo gráfico	Expresión funcional	Coste en número de transistores	Retardo de la puerta en ns
AOI de 2 de ancho con 2 entradas		$F = (wx + yz)'$	8	2.0
AOI de 3 de ancho con 2 entradas		$F = (uv + wx + yz)'$	12	2.4
AOI de 2 de ancho con 3 entradas		$F = (uvw + xyz)'$	12	2.2
OAI de 2 de ancho con 2 entradas		$F = ((w + x)(y + z))'$	8	2.0
OAI de 3 de ancho con 2 entradas		$F = ((u + v)(w + x)(y + z))'$	12	2.2
OAI de 2 de ancho con 3 entradas		$F = ((u + v + w)(x + y + z))'$	12	2.4

(Implementaciones en tecnología CMOS)

Tecnología VLSI

- *Small Scale of Integration* (SSI)
  - Hasta 10 puertas/circuito integrado
- *Medium Scale of Integration* (MSI)
  - 10–100 puertas/circuito integrado
- *Large Scale of Integration* (LSI)
  - 100–1000 puertas/circuito integrado
- *Very Large Scale of Integration* (VLSI)
  - > 1000 puertas/circuito integrado

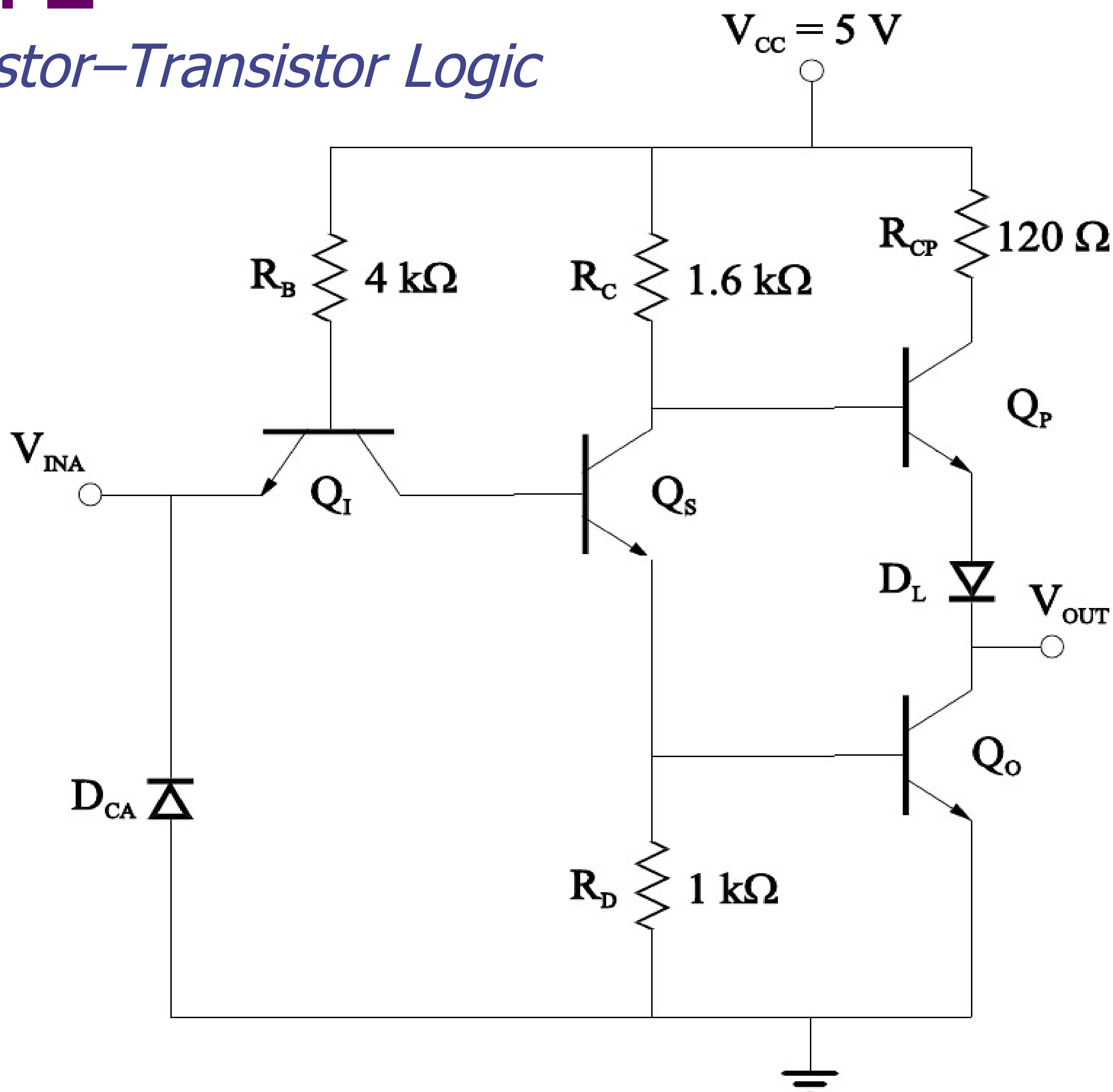
Implementaciones de puertas

- Una *familia lógica* es un conjunto de circuitos elaborados con dispositivos analógicos que realizan distintas funciones en los circuitos digitales
  - Comparten ciertas características comunes en cuanto a estructura y propiedades eléctricas

Familia TTL

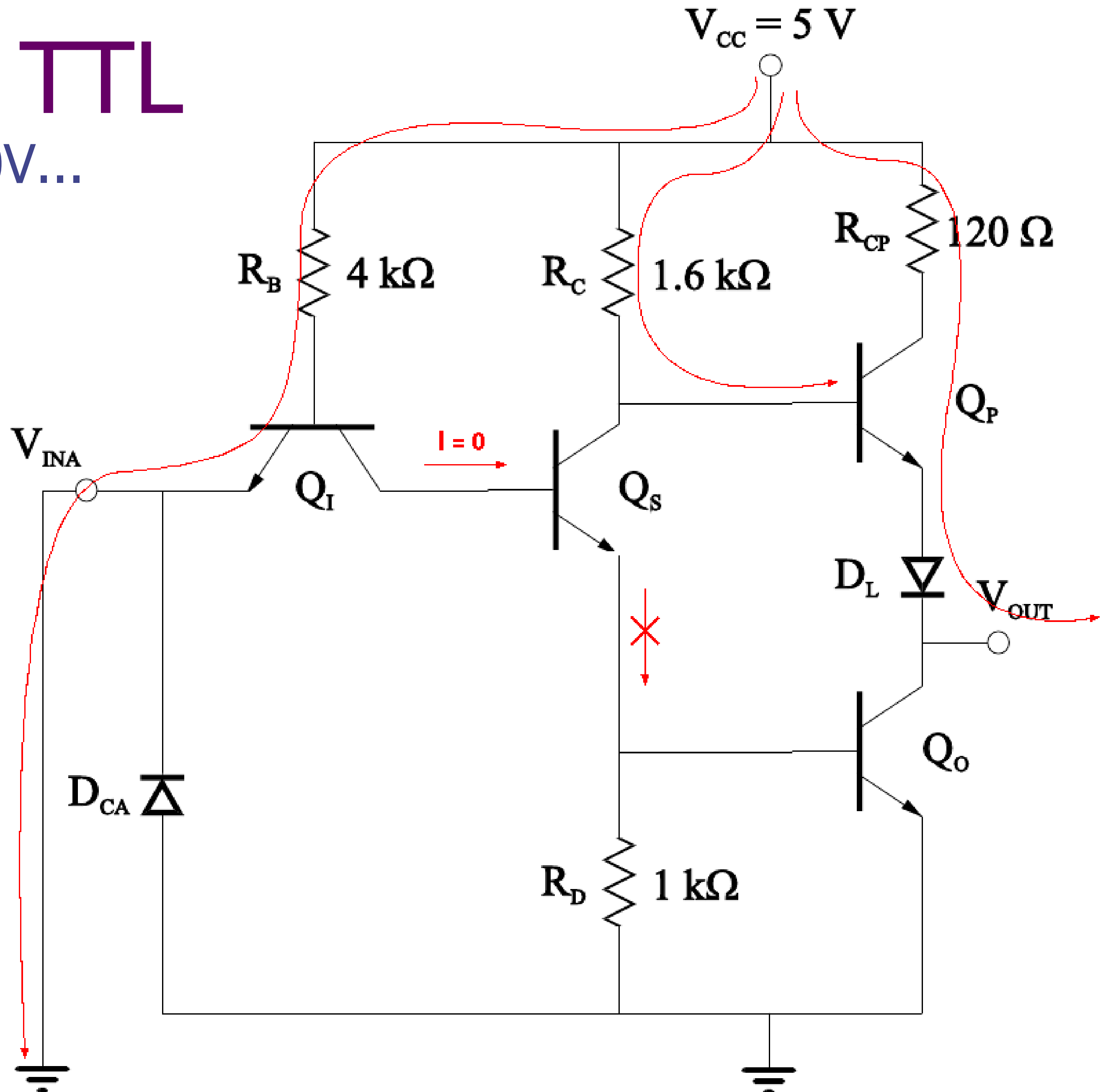
- TTL = *Transistor–Transistor Logic*

Inversor



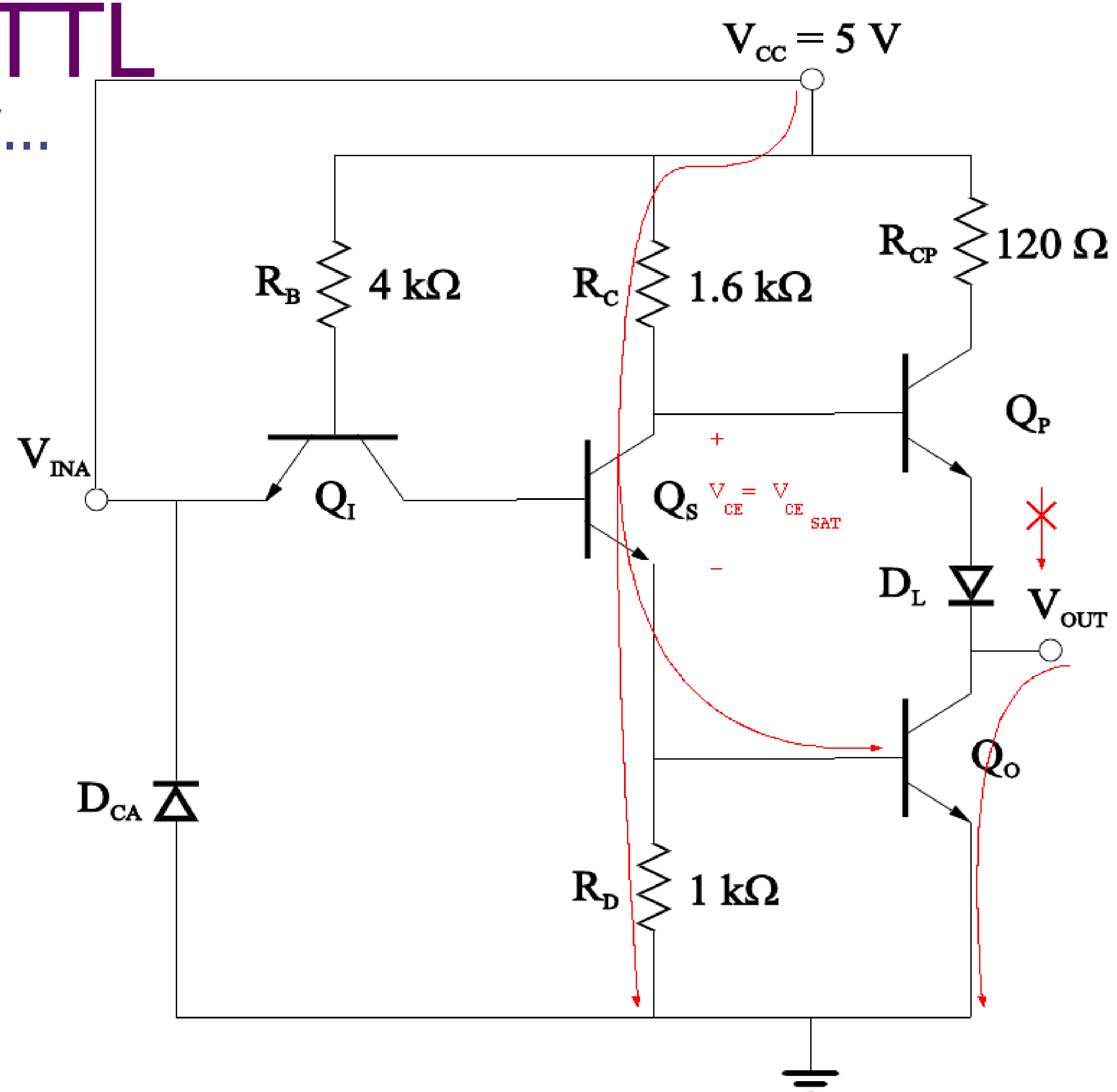
Familia TTL

- Si V_{INA} = 0V...



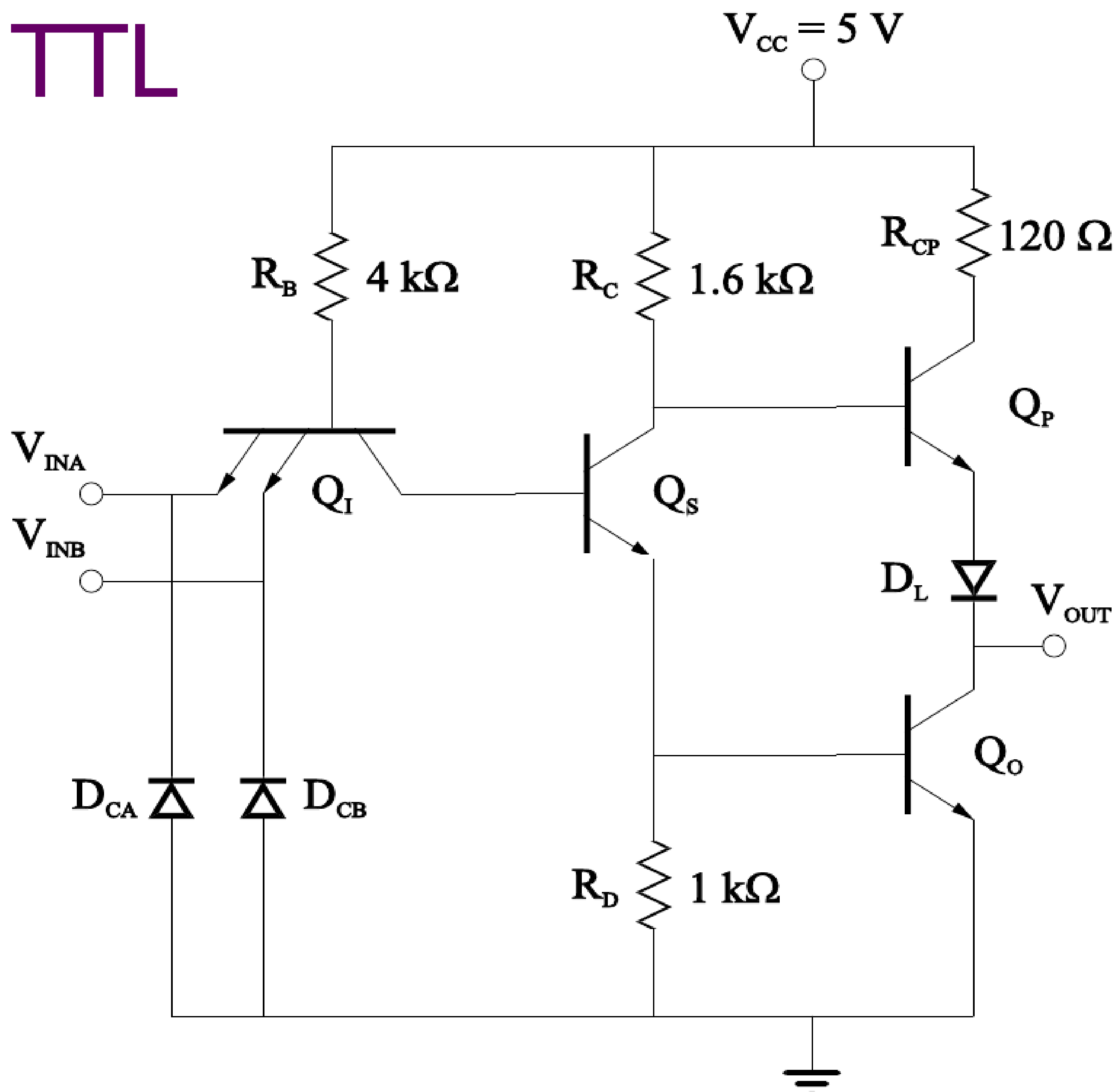
Familia TTL

- Si V_{INA} = 5V…



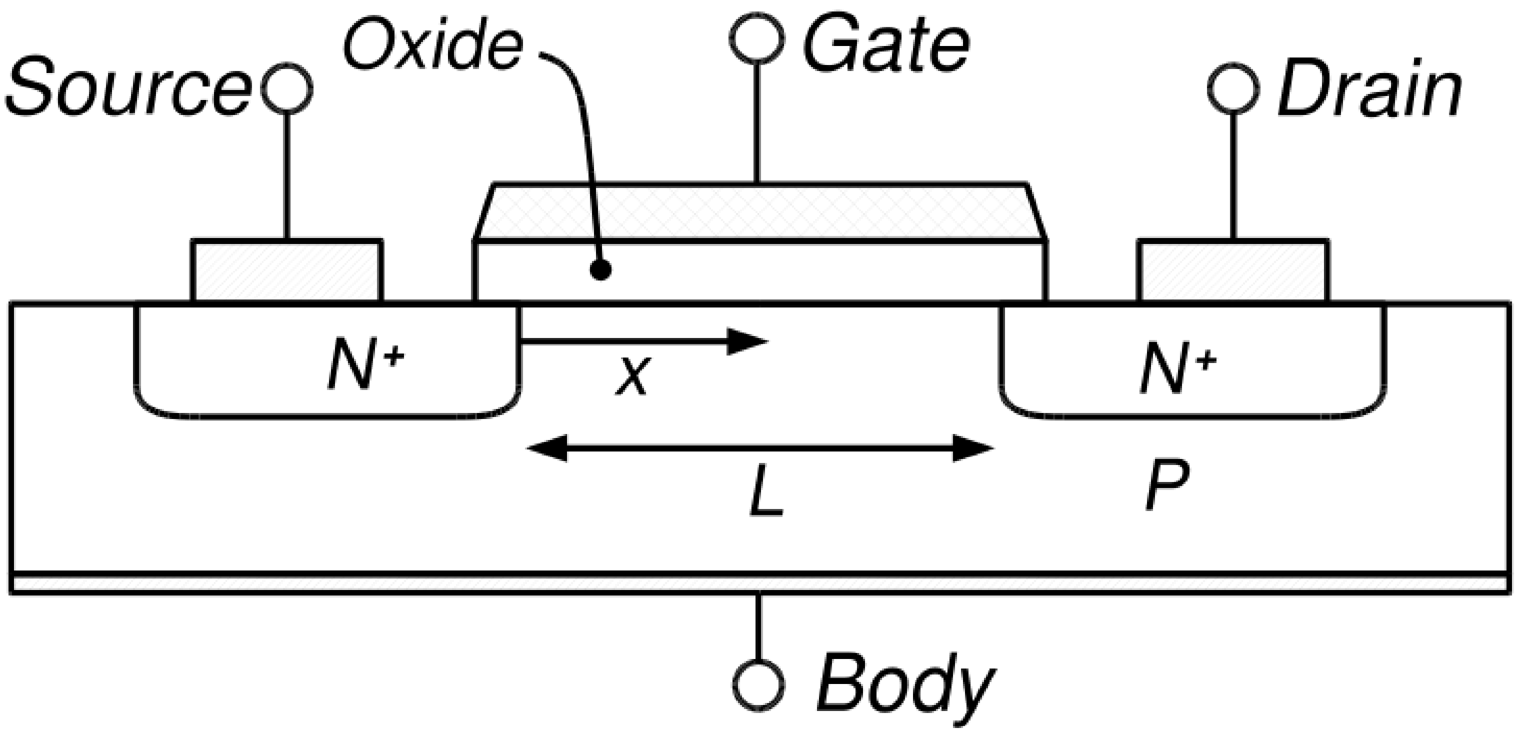
Familia TTL

NAND



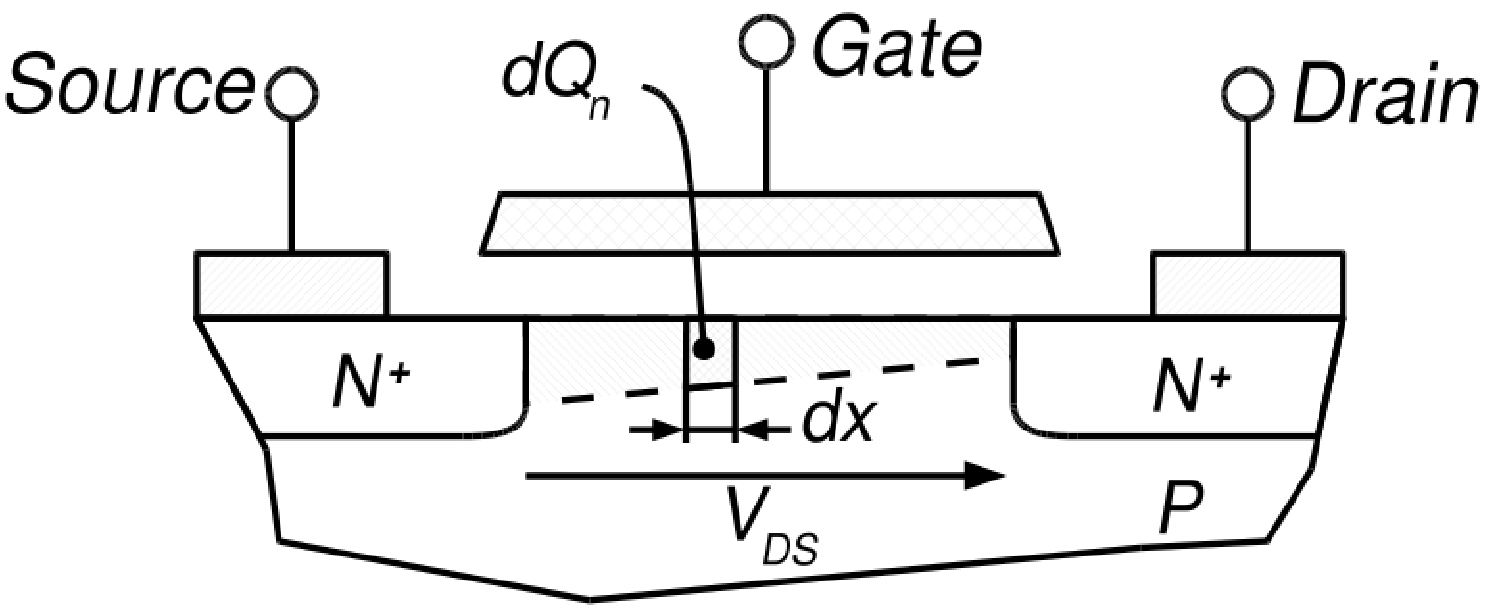
Familia CMOS

- Emplea transistores MOSFET de empobrecimiento complementarios (de canal *n* y *p*)



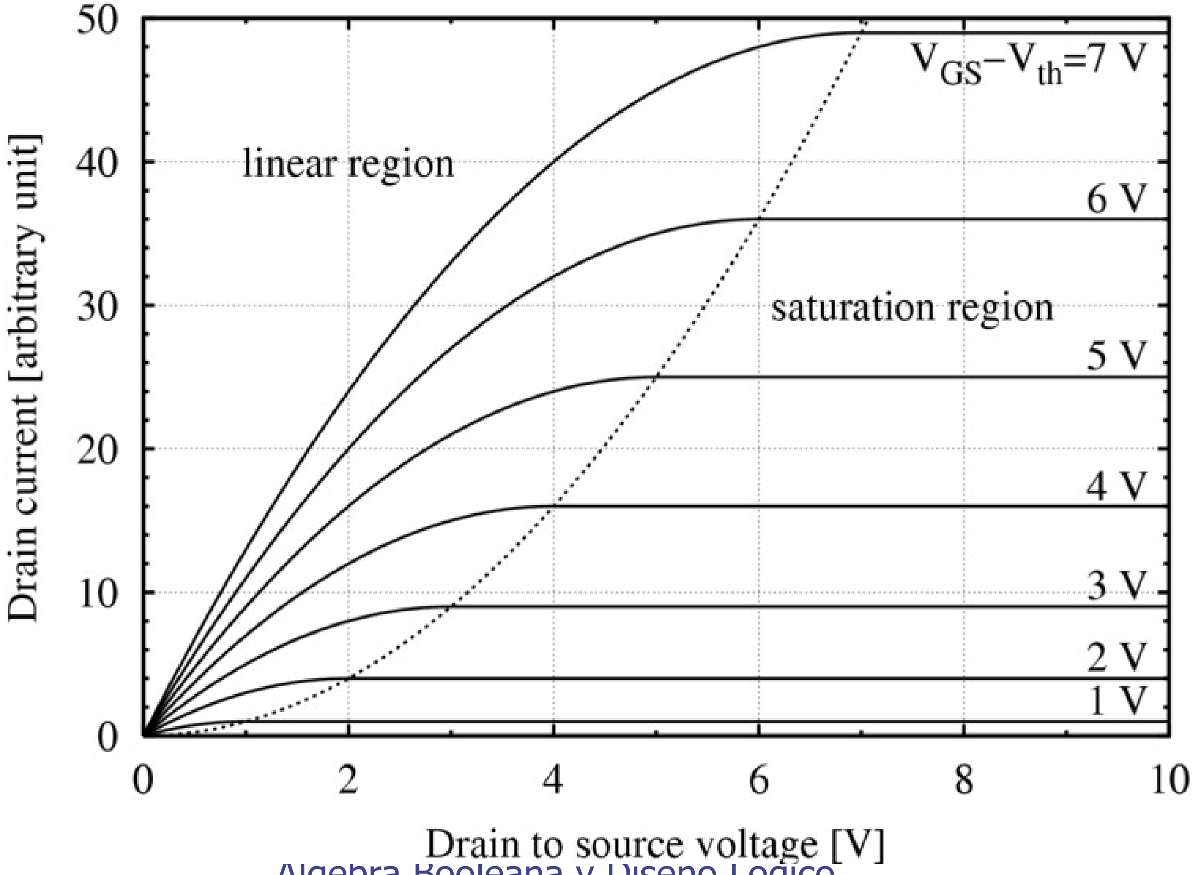
Familia CMOS

- En el transistor MOS de empobrecimiento de canal *n*, con $V_{GS} > V_{th}$ ...



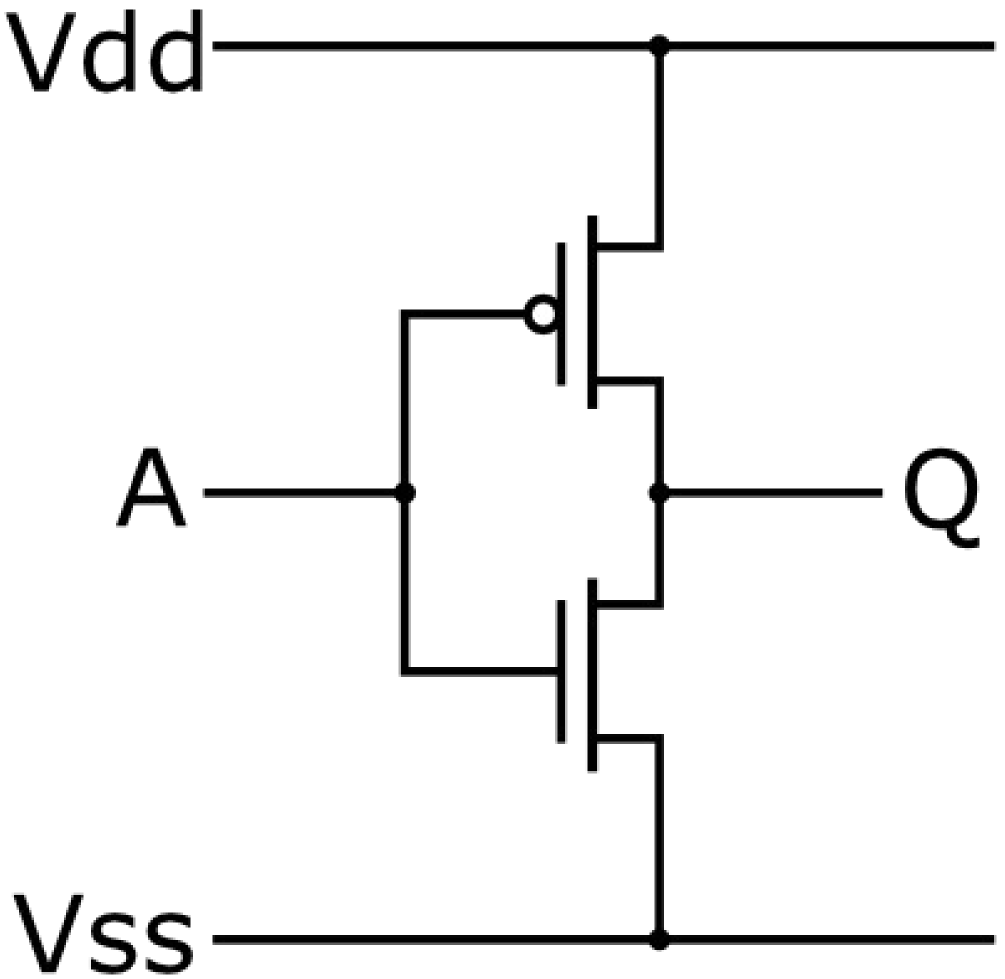
Familia CMOS

- En el transistor MOS de empobrecimiento de canal *n*, con $V_{GS}>V_{th}$

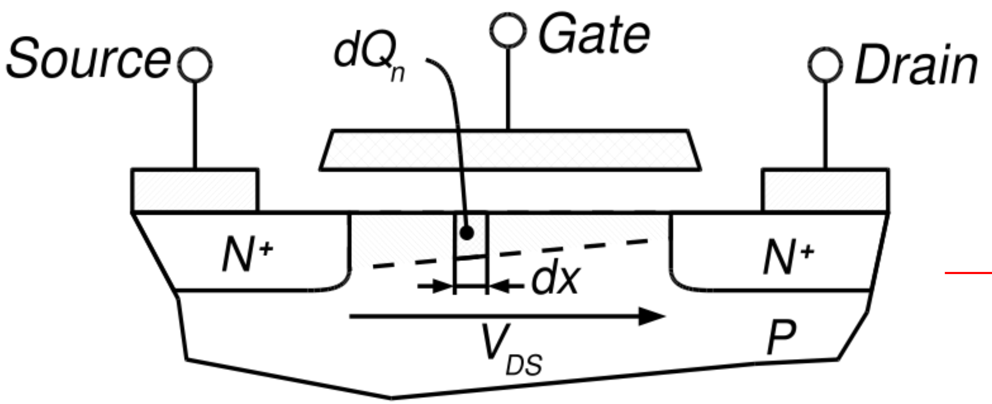


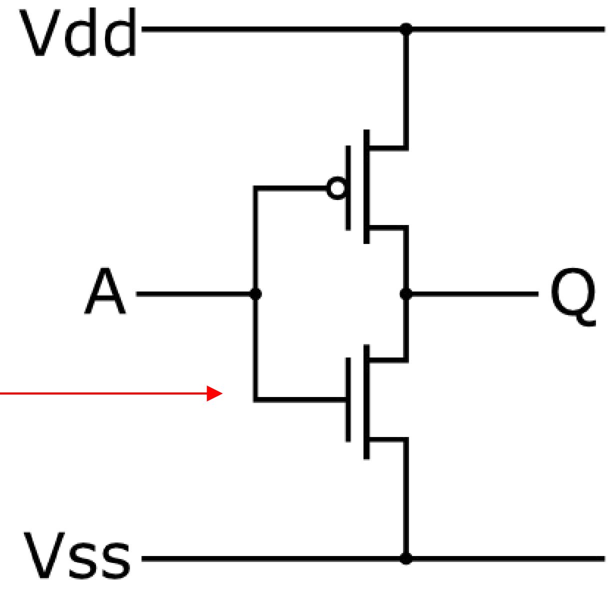
Familia CMOS

Inversor

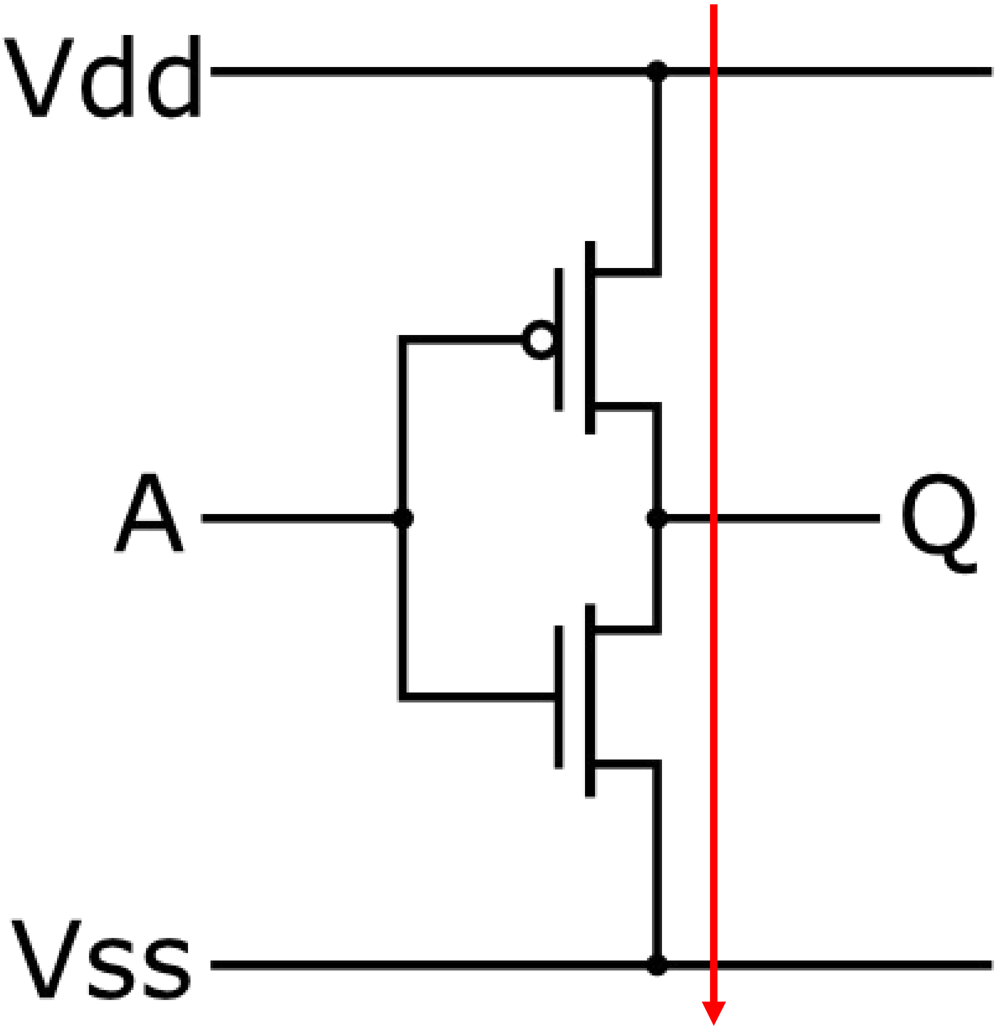


Familia CMOS



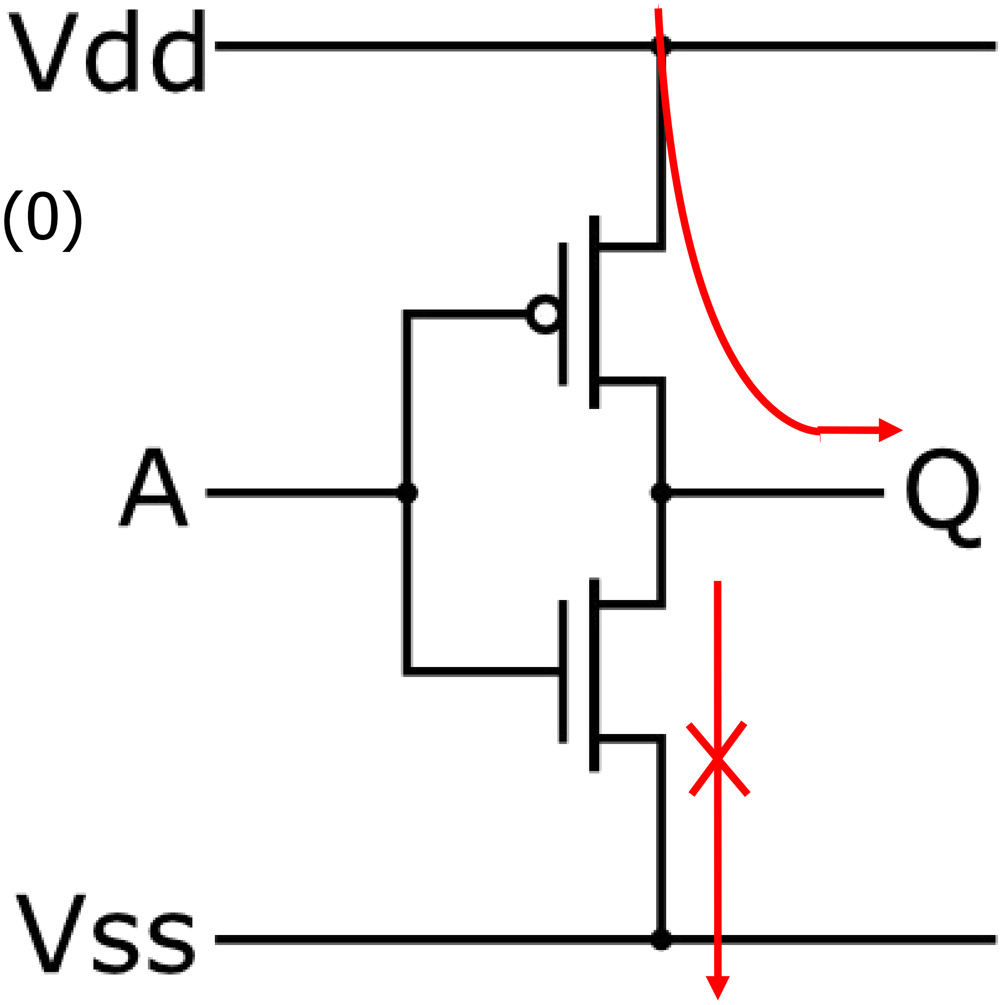


Familia CMOS



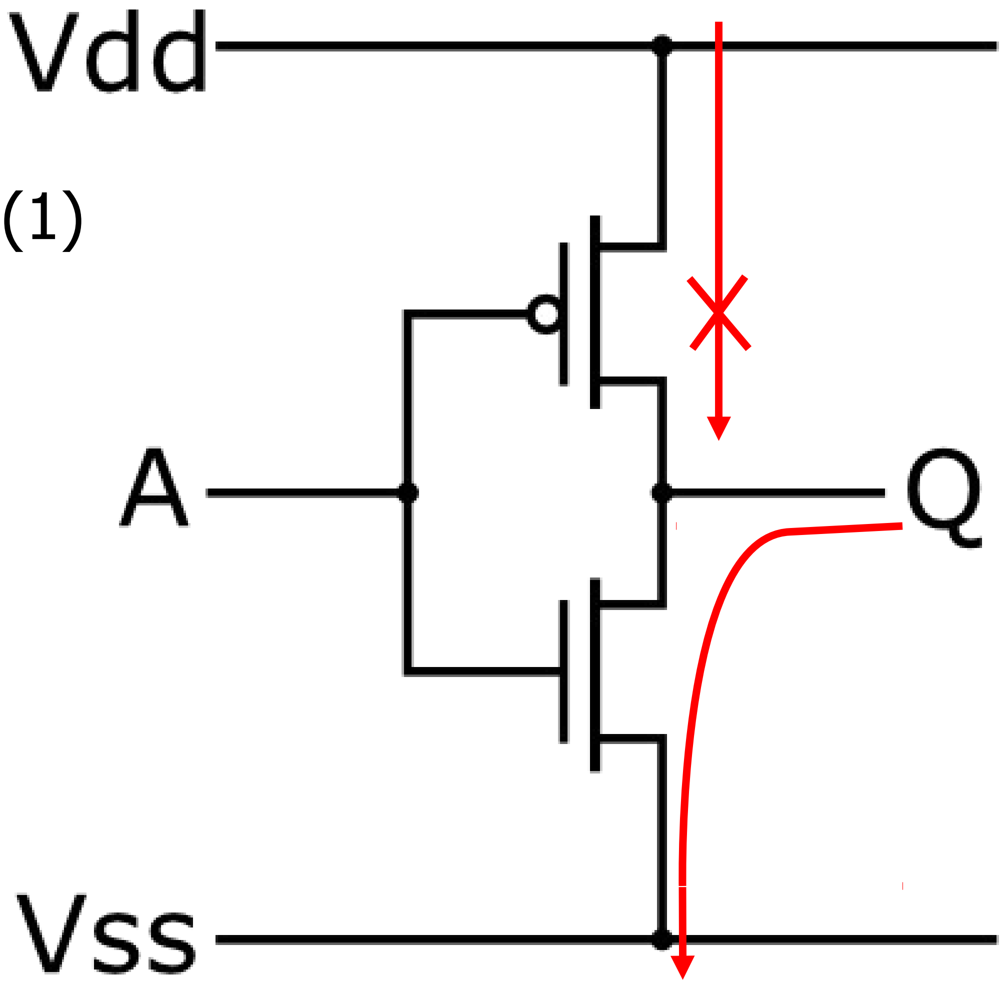
Familia CMOS

Si $V_A = V_{SS}$ (0)



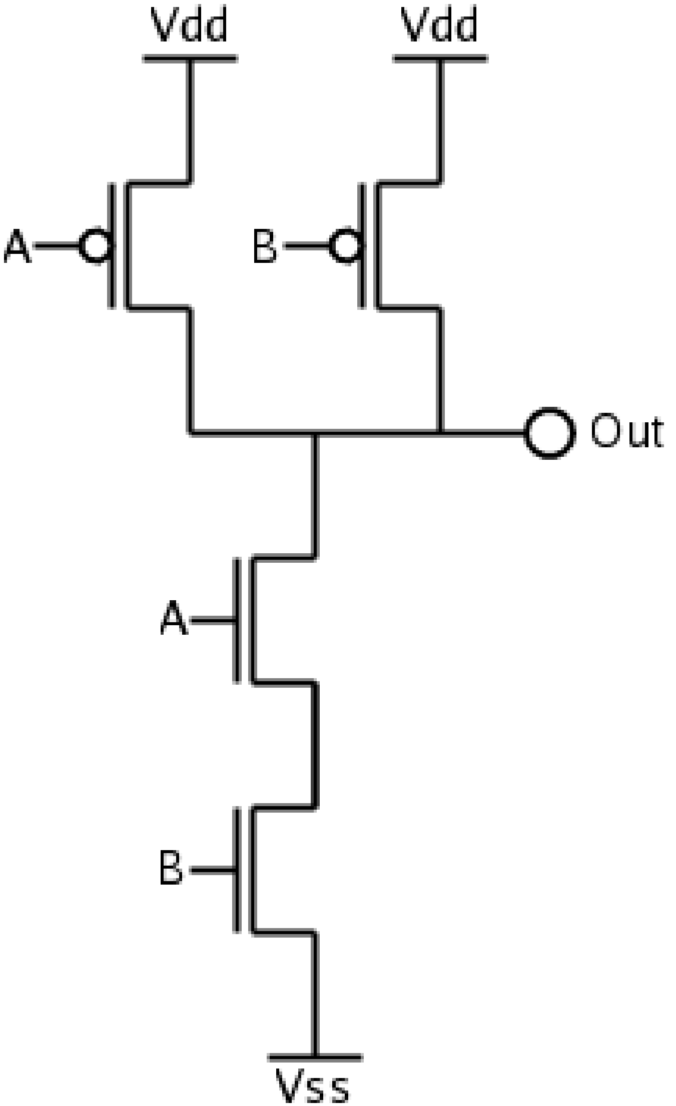
Familia CMOS

Si $V_A = V_{dd}$ (1)



Familia CMOS

NAND



Propiedades de las puertas

- Margen de ruido, niveles lógicos
- Tiempo de retardo: t_{pHL} y t_{pLH}
- Conectividad de salida (*fan-out*)
- Conectividad de entrada (*fan-in*)
- Curva de transferencia / Característica de entrada−salida
  - Estrictamente, es más apropiado hablar de “familia de curvas de transferencia”
- (...)

Propiedades de las puertas

SN5400, SN7400
QUADRUPLE 2-INPUT POSITIVE-NAND GATES

recommended operating conditions

		SN5400 MIN	SN5400 NOM	SN5400 MAX	SN7400 MIN	SN7400 NOM	SN7400 MAX	UNIT
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			0.8			0.8	V
I_{OH}	High-level output current			– 0.4			– 0.4	mA
I_{OL}	Low-level output current			16			16	mA
T_A	Operating free-air temperature	– 55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

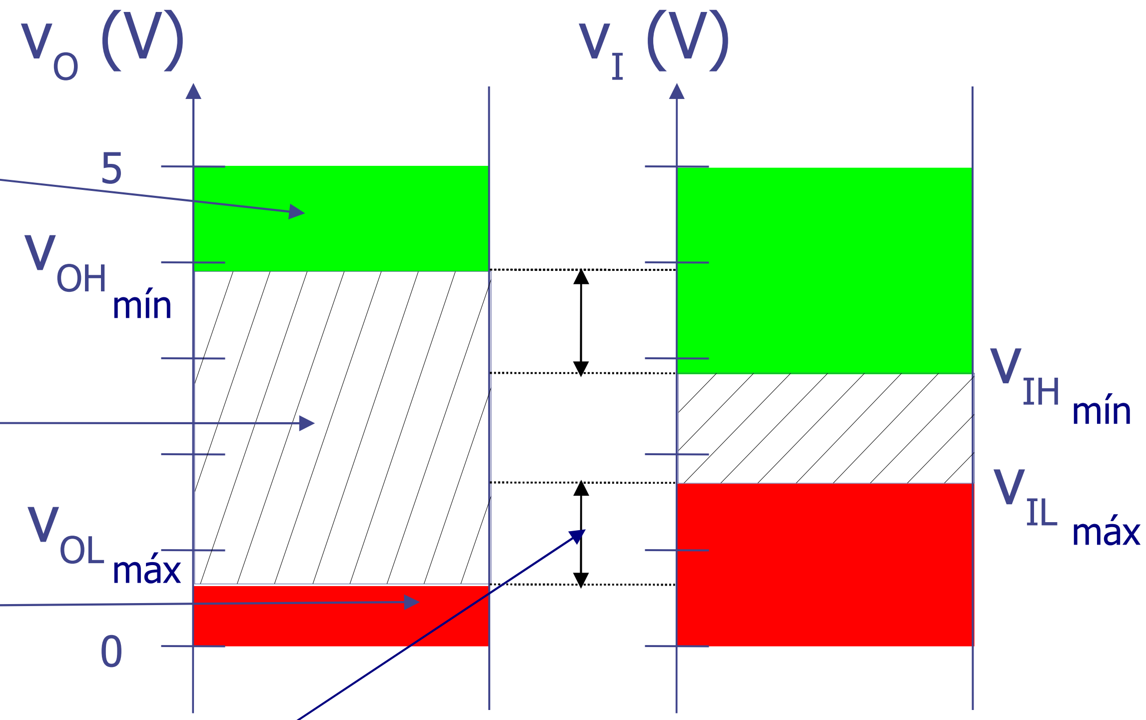
PARAMETER	TEST CONDITIONS †	SN5400 MIN	SN5400 TYP‡	SN5400 MAX	SN7400 MIN	SN7400 TYP‡	SN7400 MAX	UNIT
V_{IK}	V_{CC} = MIN, I_I = – 12 mA			– 1.5			– 1.5	V
V_{OH}	V_{CC} = MIN, V_{IL} = 0.8 V, I_{OH} = – 0.4 mA	2.4	3.4		2.4	3.4		V
V_{OL}	V_{CC} = MIN, V_{IH} = 2 V, I_{OL} = 16 mA		0.2	0.4		0.2	0.4	V
I_I	V_{CC} = MAX, V_I = 5.5 V			1			1	mA
I_{IH}	V_{CC} = MAX, V_I = 2.4 V			40			40	µA
I_{IL}	V_{CC} = MAX, V_I = 0.4 V			– 1.6			– 1.6	mA
I_{OS}§	V_{CC} = MAX	– 20		– 55	– 18		– 55	mA
I_{CCH}	V_{CC} = MAX, V_I = 0 V		4	8		4	8	mA
I_{CCL}	V_{CC} = MAX, V_I = 4.5 V		12	22		12	22	mA

Propiedades de las puertas

Valores de tensión que representan un ‘1’

Niveles de tensión no válidos

Niveles de tensión que representan un ‘0’



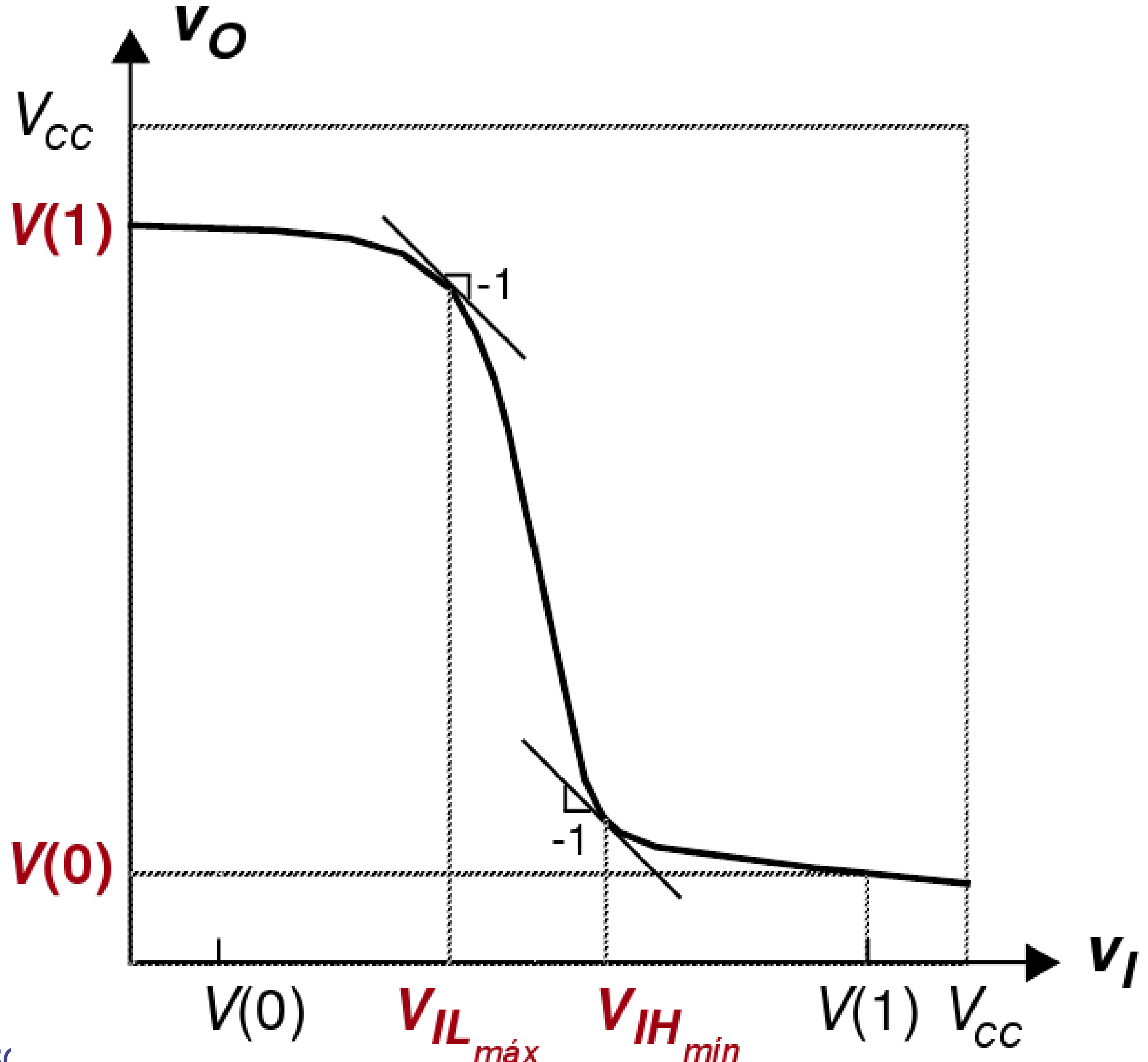
Margen de Ruido a nivel bajo:

$$NM_L = V_{IL_{máx}} - V_{OL_{máx}}$$

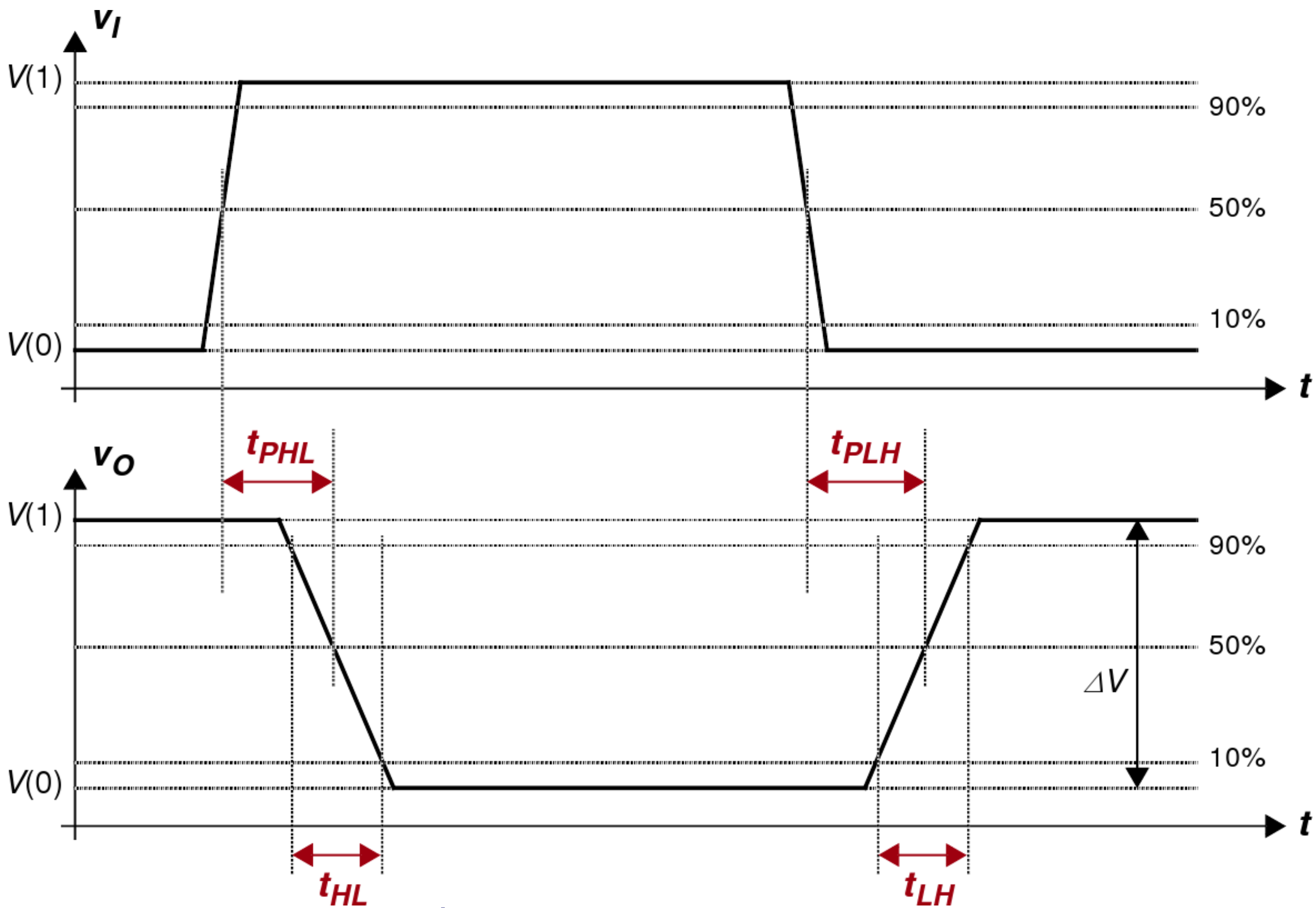
Margen de Ruido a nivel alto:

$$NM_H = V_{OH_{mín}} - V_{IH_{mín}}$$

Característica de entrada–salida



Tiempo de propagación (retardo)



Tiempo de propagación (retardo)