

Apéndice 4

Circuitos de Conmutación

Circuitos Lógicos.

Los dispositivos lógicos pueden implementarse en base a componentes electrónicas, eléctricas, mecánicas, neumáticas, etc. Entre las componentes eléctricas tenemos interruptores y relés. Entre las electrónicas pueden mencionarse circuitos en base a diodos, transistores bipolares y de efecto de campo.

A4.2 Interruptores manuales.

A4.2.1 Circuito and.

La Figura A4.3 ilustra la implementación de la función and de dos variables, mediante interruptores manuales de dos posiciones. La ampolleta se prende cuando ambos interruptores están cerrados. Se muestra el interruptor 2 en posición cerrado.

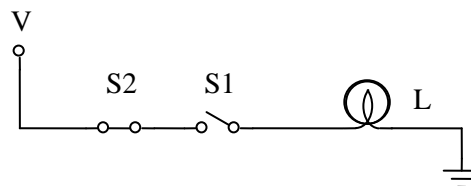


Figura A4.3 And con interruptores

Si a interruptor cerrado se asocia un valor lógico verdadero; y a interruptor abierto el valor lógico falso; y si además a la variable ampolleta L, se le asigna valor lógico verdadero cuando está encendida y falso si está apagada, se tendrá:

S1	S2	L
0	0	0
0	1	0
1	0	0
1	1	1

Interruptor cerrado: 1
Ampolleta encendida: 1

Figura A4.4 Asignación de valores lógicos 1.

De la tabla de verdad se tiene que: $L = \text{and}(S1, S2)$

Para representar la situación mediante el algebra de Boole, las variables deben tomar solamente dos valores. Estos valores físicos deben ser asociados o mapeados a los valores lógicos 0 ó 1. Puede comprobarse que si al valor físico, interruptor cerrado se asocia el valor lógico 0; y a interruptor abierto se asocia un 1 lógico, manteniendo la asignación para la variable ampolleta, se tendrá ahora la siguiente tabla, que resume el comportamiento del sistema anterior:

S1	S2	L
0	0	1
0	1	0
1	0	0
1	1	0

Interruptor cerrado: 0
Ampolleta encendida: 1

Figura A4.5 Asignación de valores lógicos 2.

Es decir: $L = \text{nor}(S1, S2)$

Si además se cambia la asignación lógica de la variable ampolleta por: ampolleta encendida asociada a 0 lógico; ampolleta apagada implica 1 lógico, se tendrá la siguiente representación tabular:

S1	S2	L
0	0	0
0	1	1
1	0	1
1	1	1

Interruptor cerrado: 0
Ampolleta encendida: 0

Figura A4.6 Asignación de valores lógicos 3.

Es decir: $L = \text{or}(S1, S2)$

Lo cual ilustra que las expresiones booleanas dependen de la forma en que se efectúe la asignación de valores lógicos relativa a los valores físicos.

A4.2.2 Circuito or.

El siguiente diagrama ilustra la función or de dos variables, empleando interruptores manuales.

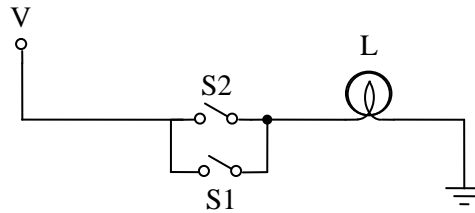


Figura A4.7 Red or con interruptores.

S1	S2	L
0	0	0
0	1	1
1	0	1
1	1	1

Ampolleta encendida = 1
 Interruptor cerrado = 1

Figura A4.7 Red or con interruptores.

De la tabla se tiene: $L = \text{or}(S1, S2)$

Puede estudiarse como cambia la ecuación lógica si se cambia la lógica a ampolleta encendida con valor lógico igual a 0.

A4.2.3 Circuito escalera.

La Figura A4.8 representa el circuito que se emplea en las escaleras para encender y apagar, desde dos lugares diferentes, la misma ampolleta.

En este caso se emplean interruptores con tres terminales, uno de ellos se denomina común. Si cuando el común se conecta con el terminal izquierdo decimos que el interruptor está en valor lógico verdadero, y falso en caso contrario, tendremos:

S1	S2	L
0	0	1
0	1	0
1	0	0
1	1	1

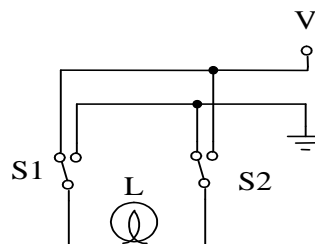


Figura A4.8 Circuito escalera.

Se tiene: $L = \text{xnor}(S1, S2)$

Se implementa la función xnor. Nótese que no importando la posición de uno de los interruptores, cuando se alterna la posición del otro interruptor se logra conmutar la condición lógica de la ampolleta.

A4.3 Relés.

En un relé electromagnético cuando circula corriente a través de la bobina se producen fuerzas que causarán que la conexión entre el contacto nc (normalmente cerrado, closed) y el común se abra, y se efectúe la conexión entre el común y el terminal denominado no (normalmente abierto, open). Cuando no circula corriente por la bobina, existe un resorte mecánico que retorna el contacto móvil a la posición nc. Se dibujan los contactos en su posición normal, con bobina desenergizada.

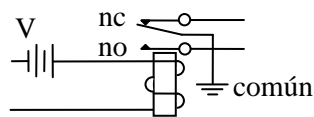


Figura A4.9 Diagrama de un relé.

Para construir sistemas en base a relés se conectan las salidas de contactos en las entradas del próximo elemento, el circuito con las bobinas. Suele emplearse la siguiente asignación lógica: bobina energizada equivale a 1 lógico, contacto cerrado equivale a 1 lógico.

Una alternativa es colocar la fuente en el lado de los contactos. De este modo el contacto normalmente cerrado coloca la tensión de fuente en la línea.

Existe un diagrama simplificado que muestra solamente los contactos, no las bobinas. A continuación se ilustra la simbología que suele emplearse en redes de conmutación, en base a relés:



Figura A4.10 Diagrama de contactos de un relé.

A4.3.1. Función de transmisión.

El siguiente esquema ilustra la función: $f(x,y,z) = x' y z$

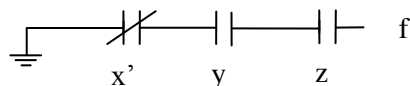


Figura A4.11 Función lógica implementada con relés.

Cuando la bobina asociada al contacto x' esté desenergizada, y cuando circule corriente por las bobinas que activan los contactos, normalmente abiertos, y y z , se tendrá que f será una baja impedancia, que asumimos como valor lógico verdadero. La salida f tomará valor falso cuando no circule corriente, o en la vía se tenga alta impedancia. Se dice que f es un función de transmisión.

Una representación esquemática más simple aún, es la que muestra la Figura A4.12:

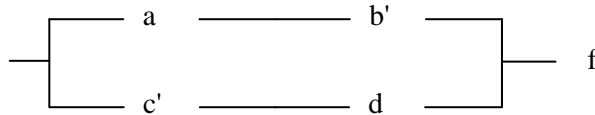


Figura A4.12 Esquema simplificado empleando relés.

Donde la ecuación lógica para f se obtiene como la suma lógica de todas las vías posibles de conexión. En la Figura A4.12 existe una vía de transmisión si ab' toma valor 1 lógico; esto ocurre cuando el contacto a está cerrado y cuando el contacto b' está cerrado; es decir b abierto. También existe una vía alternativa de transmisión si $c'd$ toma valor 1 lógico.

Entonces: $f = ab' + c'd$

A4.3.2. Función de apertura.

También puede obtenerse la ecuación lógica para f , como el producto lógico de todas las posibles aperturas de la vía.

Para la Figura A4.12 existe una apertura si $(a + c')$ toma valor lógico 0. Esto ocurre si a está abierto y c' esta abierto; es decir si c está cerrado.

Entonces: $f = (a+c') (b' +d) (a +d)(b' +c')$

Puede notarse que cada una de las aperturas está formada por los elementos que constituyen un conjunto de corte.

Ejemplo A4.1

En caso de relés, pueden efectuarse conexiones como la siguiente:

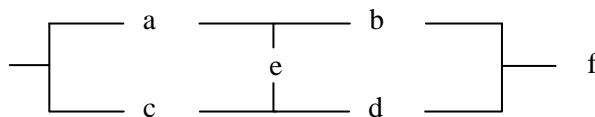


Figura A4.13 Conexión puente empleando relés.

De la Figura A4.13 si se consideran las sumas de las transmisiones posibles, se obtiene:

$$f = ab + cd + aed + ceb$$

Para los productos de las aperturas posibles se obtiene:

$$f = (a + c)(b + d)(c + e + b)(a + e + d)$$

A4.3.3. Controladores lógicos programables(PLC)

En la actualidad no se diseñan sistemas digitales basados en relés, se emplean controladores lógicos programables (PLC) que cumplen las mismas funciones.

En este ambiente suelen emplearse los siguientes esquemas:

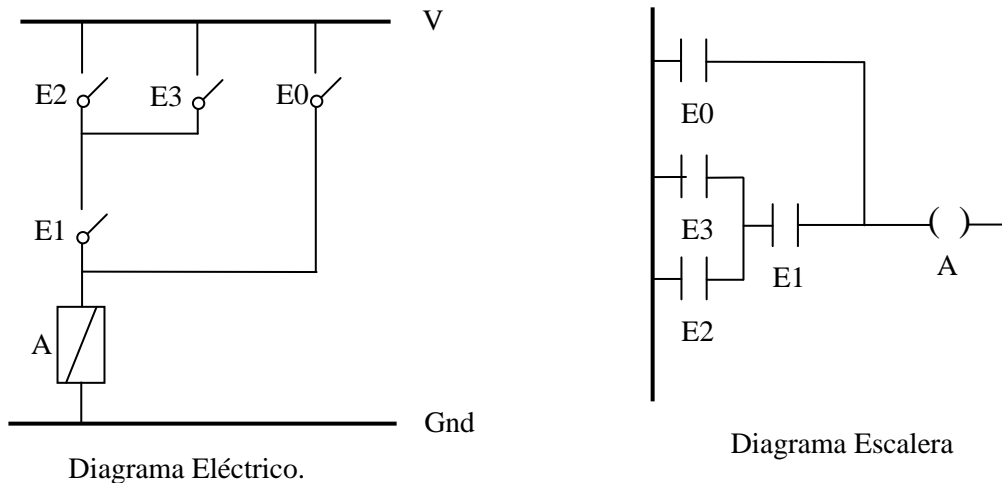


Figura A4.14 Diagramas lógicos para PLC.

En ambos diagramas, se cumple que: $A = E0 + (E1 (E2 + E3))$

A4.4 Circuitos Lógicos empleando diodos.

Un diodo es un dispositivo semiconductor que permite la circulación de corriente desde el ánodo hacia el cátodo si el voltaje entre ánodo y cátodo es mayor que 0,7[V], en diodos de silicio.

Pueden construirse compuertas lógicas empleando diodos y resistencias.

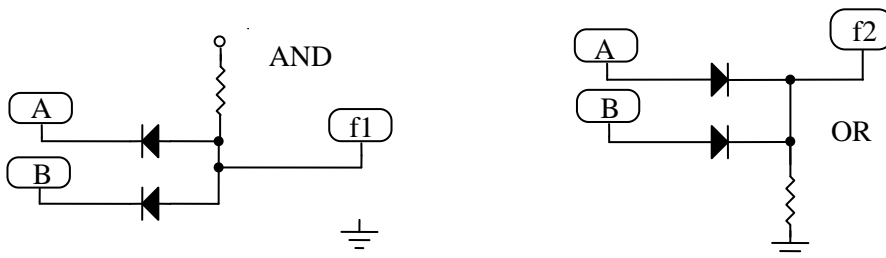


Figura A4.15 Compuertas empleando diodos.

En la red a la izquierda: si la entrada a un diodo está a tierra, fluye corriente a través de él y la salida f1 estará a un voltaje 0,7 (bajo). Si ambos voltajes en A y B son tales que los diodos

estén cortados (no conduzcan), $f1$ queda con una tensión alta (cercana a $+V$). La resistencia se denomina pull-up (tira para arriba) y fija el valor del 1 lógico. El circuito implementa la función and de las entradas.

En la red de la derecha basta que una de las entradas esté en un voltaje alto, para que el diodo conduzca y fije el voltaje de salida en el voltaje de la entrada menos 0,7. Si las dos entradas están a tierra, los diodos no conducen y el voltaje en $f2$ será bajo. La red implementa la función lógica or de las entradas.

De acuerdo a la electrónica empleada pueden definirse rangos de valores de voltaje que pueden ser considerados voltajes bajos (0 lógico) o voltajes altos (1 lógico).

Si las salidas de estas redes se conectan en las entradas de otras similares, las caídas de voltajes a través de los diodos van degradando significativamente los niveles de los voltajes en las salidas. Si se conectan, por ejemplo, varias compuertas and, en cascada, y los diodos conducen, la salida de la etapa final tendrá un voltaje cada vez mayor (no podrá asociarse con un cero lógico). Una solución es aumentar el voltaje de las fuentes y redefinir los rangos de voltajes para el cero y uno lógico, esto además de aumentar el consumo de energía no soluciona el problema de conectar varias componentes en cascada. Adicionalmente no puede implementarse un inversor, empleando diodos y resistencias solamente. Por esta razón se emplean componentes activas, en el diseño de las compuertas lógicas.

A4.5 Transistor bipolar.

En el diagrama se ilustra un transistor bipolar, éstos fueron ampliamente usados para implementar compuertas lógicas entre 1970 y hasta comienzos de la década de 1980.

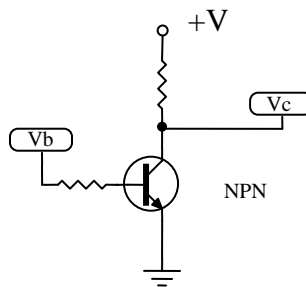


Figura A4.16 Transistor bipolar.

Un voltaje alto en la base satura al transistor, y el voltaje del colector queda en una tensión cercana a 0,2 (voltaje bajo). Cuando se aplica un voltaje bajo en la base, el transistor se corta y la salida adquiere un valor alto a través de la resistencia de pull-up. El circuito opera como un inversor lógico.

Nuevamente pueden definirse rangos de voltajes que pueden ser considerados 0 y 1 lógicos.

A4.6 Compuerta Nand DTL. (Diode transistor logic).

Combinando el circuito and en base a diodos, con el inversor en base a un transistor, se logra un circuito NAND. La familia DTL, fue empleada hasta comienzos de la década de 1970, en

base a chips integrados en pequeña escala con 14 pines. El diagrama de la Figura A4.17 ilustra una compuerta Nand de dos entradas.

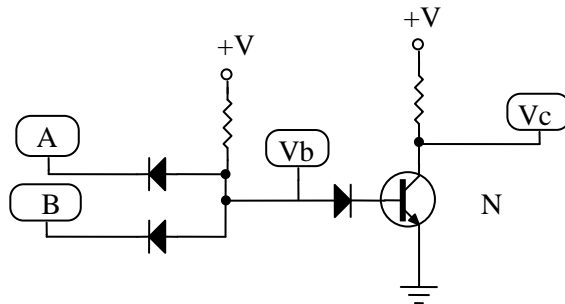


Figura A4.17 Compuerta NAND DTL.

En esta configuración un 0 lógico está representado por un voltaje menor que $0,7[V]$; el 1 lógico está sobre $4[V]$.

Para que el transistor conduzca, el voltaje V_b debe ser mayor que $1,4[V]$. Si es menor, el transistor estará cortado y la salida tomará valor alto (cercano al voltaje de polarización a través de la resistencia de pull-up).

Fan-out.

Si la salida baja de un nand está conectada a varias entradas, éstas le inyectan corriente a través de los diodos que conducen, lo cual podría sacar de saturación al transistor. Se denomina fan-out, al máximo número de entradas que pueden conectarse a una salida, manteniendo los voltajes dentro de los rangos asignados a la lógica.

La implementación de compuertas nand y nor mediante lógica DTL es sencilla de realizar. Y debido a que éstas son funcionalmente completas pueden diseñarse todas las funciones lógicas solamente empleado nands o solamente nor.

Wired-and.

Una ventaja adicional de la configuración es que permite conectar directamente las salidas, lográndose mediante la conexión la función lógica and (se denomina wired-and; es decir, un and debido al alambrado.)

Si suponemos dos salidas f_1 y f_2 conectadas, se tendrá que basta que una de ellas esté en 0 (es decir, que tenga el transistor asociado saturado) para que la salida f (el punto de conexión) también esté en voltaje bajo. Sólo cuando ambas salidas están altas (ambos transistores cortados) la salida f será alta.

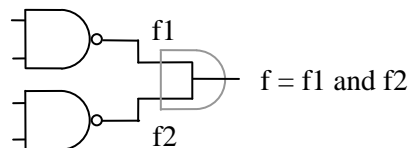


Figura A4.18 AND alambrado.

El circuito AND, en la Figura A4.18, simboliza la función lógica que se logra a través de la conexión.

A4.7 Compuertas TTL.

Pueden emplearse transistores, en lugar de diodos, para construir compuertas. Empleando, de esta forma: solamente transistores y resistencias. Estos diseños se conocen como lógica TTL (transistor-transistor-logic). Los diodos que forman el and (del diseño DTL) pueden ser reemplazados por un transistor que tiene múltiples emisores. Esto considerando que un transistor está formado por dos diodos que apuntan desde la base (en caso de npn).

El transistor con dos emisores, reemplaza a los tres diodos del circuito NAND DTL. El circuito simplificado se ilustra a continuación:

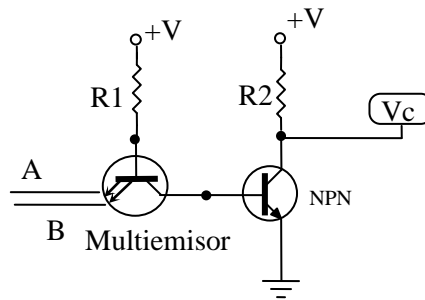


Figura A4.19 NAND TTL.

Cuando una de las entradas A o B está baja, la corriente que circula a través de R1 hacia el colector no alcanza para hacer conducir al transistor de salida, y éste estará cortado; con lo cual la salida estará alta a través de R2, la resistencia que fija el valor alto. Cuando ambas entradas están altas circula corriente (en modo activo inverso) hacia la base del transistor de salida saturándolo. De este modo el voltaje de salida será tipo 0,2 (valor bajo).

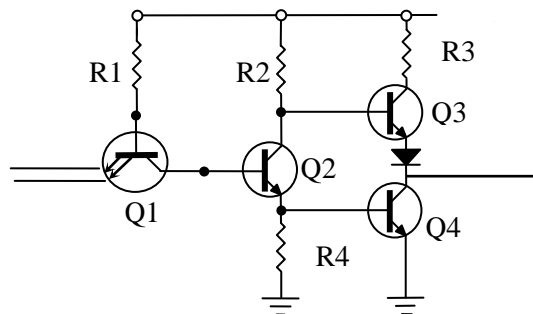


Figura A4.20 Salida totem-pole TTL.

La Figura A4.20 muestra un Nand TTL de dos entradas. El transistor Q3 y Q4 forman una configuración denominada totem, Q3 introduce una constante de tiempo menor para producir la salida alta.

Algunas componentes de la familia, denominadas de colector abierto, no tienen la resistencia R_3 , el transistor Q_3 y el diodo; estas compuertas requieren una resistencia externa conectada a la fuente para proveer el nivel alto de salida. La resistencia se calcula para suministrar la corriente a las entradas que estén conectadas a dicha salida. Además un conjunto de salidas de colector abierto pueden conectarse formando un and alambrado, similar al posible en la familia DTL.

Niveles lógicos TTL

Los valores de las componentes se diseñan de tal modo que se cumplan las especificaciones siguientes:

$$V_{OH} = 2,4[V] ; V_{OL} = 0,4[V] ; V_{IH} = 2[V] ; V_{IL} = 0,8[V]$$

Donde V_{IH} es el mínimo voltaje de entrada que puede ser considerado como 1 lógico; V_{IL} es el máximo voltaje de entrada que es reconocido como 0 lógico.

Similarmente: V_{OH} es el mínimo voltaje de salida que se reconoce como 1 lógico; y V_{OL} es el máximo voltaje que es considerado un 0 lógico.

Si una salida está en V_{OH} y se conecta a una entrada, en ésta se permite un V_{IH} que es 0,4 volts menor; es decir la señal de salida puede degradarse en nivel y aún ser reconocida como un valor lógico 1.

Si una salida está en V_{OL} y se conecta a una entrada, en ésta se permite un V_{IL} que es 0,4 volts mayor; es decir la señal de salida puede contaminarse con ruido externo, y aún ser reconocida como un valor lógico cero. Las diferencias entre dichos niveles se denominan márgenes de ruido de la familia.

A4.8 Compuertas CMOS

Interruptores MOS.

El transistor Metal-Oxide Semiconductor (MOS), actúa como interruptor controlado por voltaje. Casi todas las componentes digitales actuales se diseñan en esta tecnología. Suelen emplearse en pares complementarios, por esto se conoce como tecnología CMOS.

Funciona como interruptor si el voltaje de compuerta (gate) es levemente mayor o menor que el voltaje del terminal fuente (source): Se establece una vía conductora entre la fuente y el sumidero (drain).



Figura A4.21 Canal n y p.

Un transistor MOSFET (Metal Oxide Semiconductor Field Effect Transistor) es un dispositivo de tres terminales que actúa como un interruptor (switch). Estudiaremos el de tipo enriquecimiento.

El voltaje de compuerta (gate, no confundir con compuerta lógica, se emplea el mismo nombre) controla si la trayectoria desde el sumidero (drain) hacia la fuente (source) será un circuito abierto (off) o un camino resistivo (on). Un circuito abierto puede entenderse como un camino de alta impedancia. La compuerta (el terminal) se dibuja como un condensador.

Transistor nfet.

Existen dos clases de MOSFET:

- Un MOSFET de tipo n (nfet) conduce ($I_{DS} > 0$) para voltaje de compuerta alto ($V_{GS} > V_{GS}(\text{umbral}) = V_T > 0$); y no conduce en caso contrario.

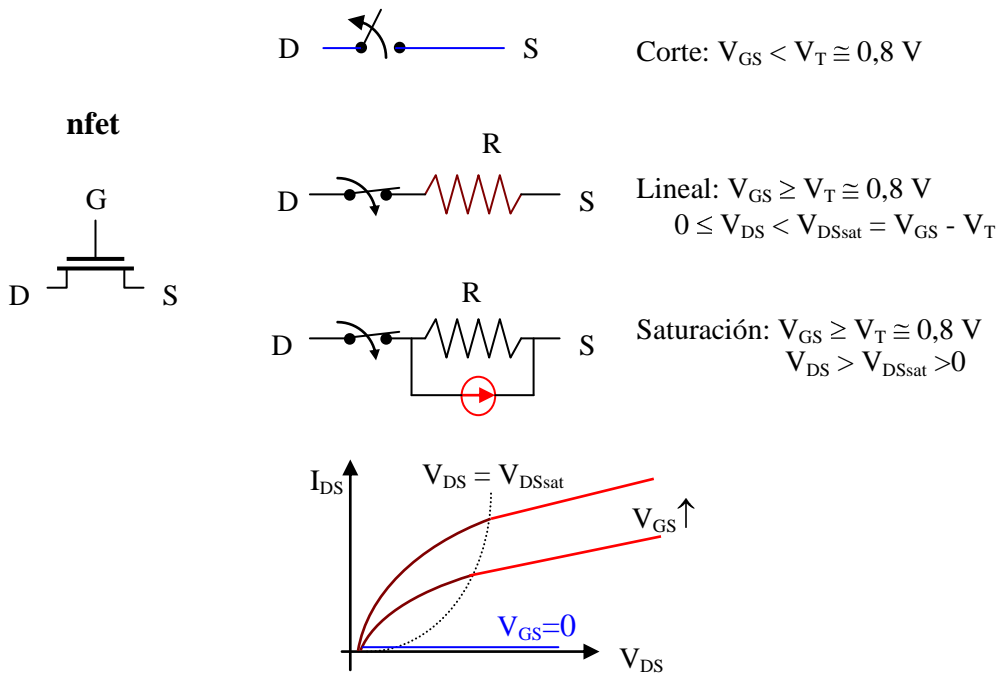


Figura A4.22 Transistor nfet.

Cuando conduce un nfet deja pasar limpiamente un nivel cero. El sustrato está conectado siempre a tierra.

Transistor pfet.

- Un MOSFET de tipo p (pfet) conduce ($I_{SD} > 0$) para voltaje de compuerta bajo ($V_{GS} > -|V_{GS(umbral)}| < 0$); y no conduce en caso contrario.

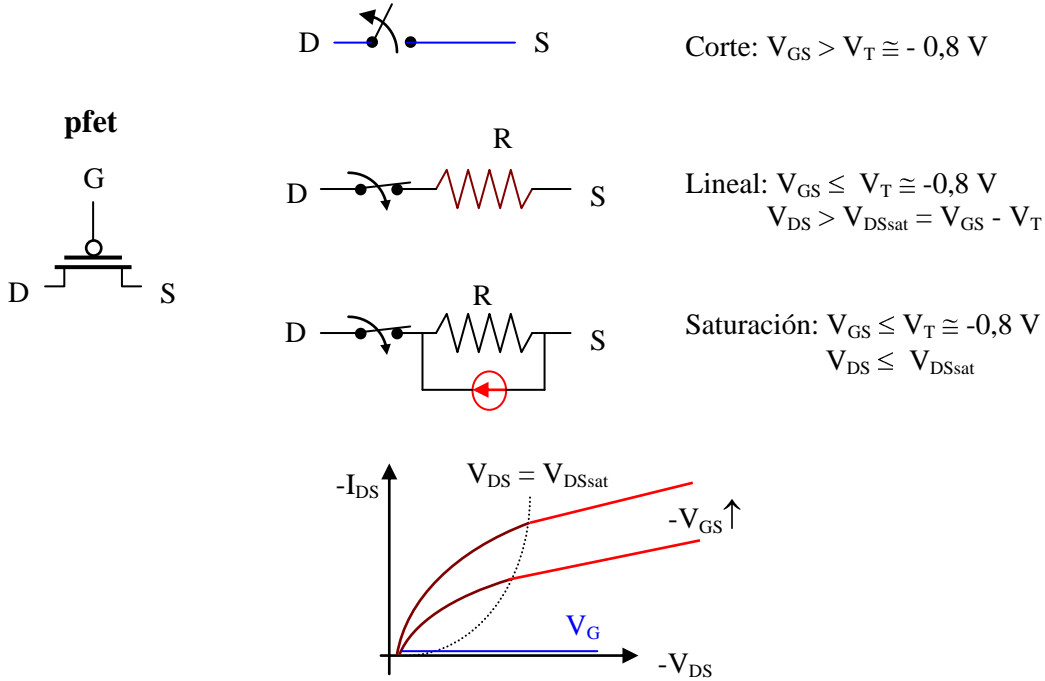


Figura A4.23 Transistor pfet.

Cuando conduce un pfet deja pasar limpiamente un nivel uno. El sustrato está conectado siempre a V_{DD} .

Construcción nfet.

Una sección de un transistor CMOS se ilustra en el siguiente diagrama.

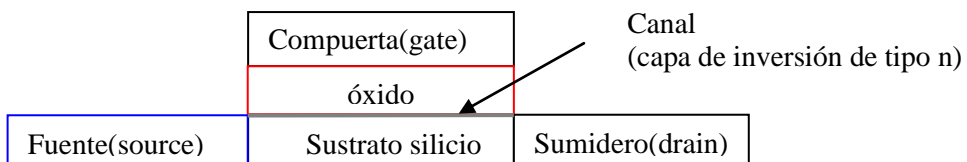


Figura A4.24 Diagrama construcción transistor pfet.

En un nfet: El material de la lámina del fondo es de silicio, y está separada de la capa superior (denominada gate) por una capa de óxido aislante. Originalmente la capa superior era metálica (de esto quedó el nombre del transistor), luego se empleó silicio policristalino. La fuente y el sumidero son de silicio con gran exceso de electrones (material n) y están separadas por el sustrato de silicio cargado fuertemente positivo (material p). La fuente y el sumidero son las regiones de difusión, denominadas así por el proceso químico empleado para su fabricación, en el cual iones cargados negativamente (átomos con electrones extra de valencia) son colocados en la superficie de silicio y difundidos, mediante el calentamiento de éste. Los materiales de la fuente y el sumidero son idénticos. Por convenio la fuente es el terminal con el menor de los dos potenciales en ambos extremos del canal.

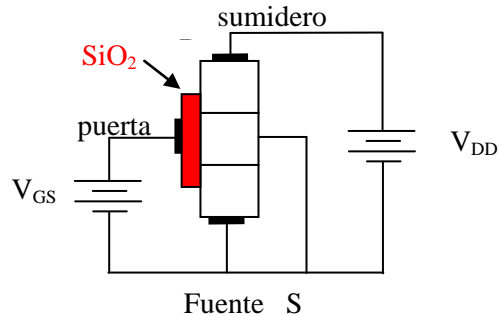


Figura A4.25 Diagrama físico transistor nfet.

Operación.

La conducta del transistor nfet, es como sigue: Cuando se aplica un voltaje positivo en la compuerta los electrones son atraídos hacia el canal (la región, inicialmente no conductora, entre la fuente y el sumidero y muy cercana a la superficie del oxido de silicio). Los electrones libres se recombinan con los huecos del material p y comienzan a circular electrones desde la fuente al sumidero. Cuando el voltaje de compuerta es suficientemente alto muchos electrones pasan al canal, y se establece conducción entre la fuente y el sumidero. Se crea una delgada capa de material tipo n próxima al dióxido de silicio, que se denomina capa de inversión tipo n (es el canal). La tensión mínima que crea el canal se denomina V_{GS} (umbral), que dependiendo de los materiales empleados puede estar entre 1 y 5 volts.

Si el voltaje de compuerta es muy bajo, prácticamente no hay electrones en el canal (salvo los electrones libres producidos, en el material p, por activación térmica) y puede considerarse que no existe flujo apreciable de corriente entre la fuente y el sumidero, se dice que el canal no conduce. Este comportamiento es diferente al fet de juntura o al de empobrecimiento.

Este dispositivo puede estar en corte, o conducir. En caso de hacerlo se comporta como una resistencia (similar a la zona de saturación en un transistor bipolar) o como fuente de corriente (similar a la zona activa de un bipolar). Debido a la existencia de un voltaje umbral y su alta impedancia de entrada son dispositivos ideales para implementar sistemas lógicos.

En un pfet, las regiones de difusión están positivamente cargadas, y el sustrato de silicio está negativamente cargado. En éste, cuando se aplica un voltaje bajo, el canal conduce y cuando se aplica un voltaje alto el canal se considera que no conduce, o que está abierto.

Debido a que la capa de dióxido de silicio se mantiene lo más delgada que sea posible, para que el campo sea más intenso y se pueda controlar mejor el canal, una tensión elevada (tipo 30 Volts) de puerta hasta la fuente puede perforar la capa aislante, dañando el transistor. Estas tensiones pueden producirse fácilmente en forma estática, por esta razón debe cuidarse su manipulación.

Los símbolos para los tipos de transistores recuerdan el modo de operación, el pequeño círculo del pMOS recuerda que el transistor conduce cuando la compuerta es activada por lógica negativa.

Inversor MOS RTL.

La siguiente red ilustra un inversor, implementado con una resistencia y un nfet. Con una tensión menor que la umbral, se asume que la entrada es baja. Si la entrada es baja, el transistor está cortado, y la salida estará en alto a través de la resistencia y la fuente. Cuando la tensión de entrada es mayor que el umbral del nfet, éste conduce, y debe diseñarse tal que la resistencia externa sea mucho mayor que la del fet en conducción, ya que en este caso la tensión de salida será baja.

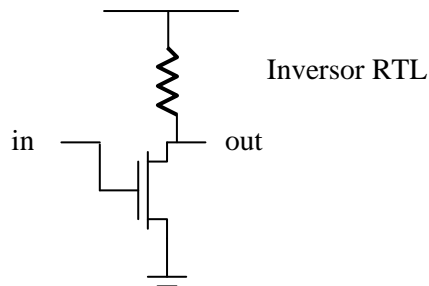


Figura A4.26 Inversor MOS RTL.

Inversor CMOS.

El inconveniente de este inversor es que el tamaño físico de la resistencia externa es mucho mayor que el espacio requerido para el transistor. Además tiene alto consumo de energía, en la resistencia; aumento de la constante de tiempo de subida.

En lugar de emplear la resistencia se puede usar un transistor complementario como resistencia de carga activa. Esta idea permitió reducir apreciablemente el espacio ocupado por un inversor.

Una compuerta típica CMOS estática está construida con dos redes complementarias:

- Una red que fija el cero (pull-down, tira para abajo) compuesta de nfets, con fuentes conectadas a tierra. Esto lleva a $V_{OL} = 0$.
- Una red que fija el uno (pull-up, tira para arriba) compuesta de pfets, con fuentes conectadas a V_{DD} . Esto lleva a $V_{OH} = V_{DD}$.

Solamente una de estas redes puede estar conduciendo en un determinado tiempo. La salida será conectada a tierra o a V_{DD} ; pero no a ambas. Si ambas redes conducen simultáneamente,

la trayectoria eléctrica desde VDD hacia GND causará excesivo flujo de corriente y puede dañar el circuito. (Esto puede suceder si las entradas a una compuerta CMOS son de tercer estado y están flotando.)

La más simple de las compuertas CMOS, es el inversor, formado por un nfet y un pfet. Como se verá otros tipos de compuertas requieren más transistores.

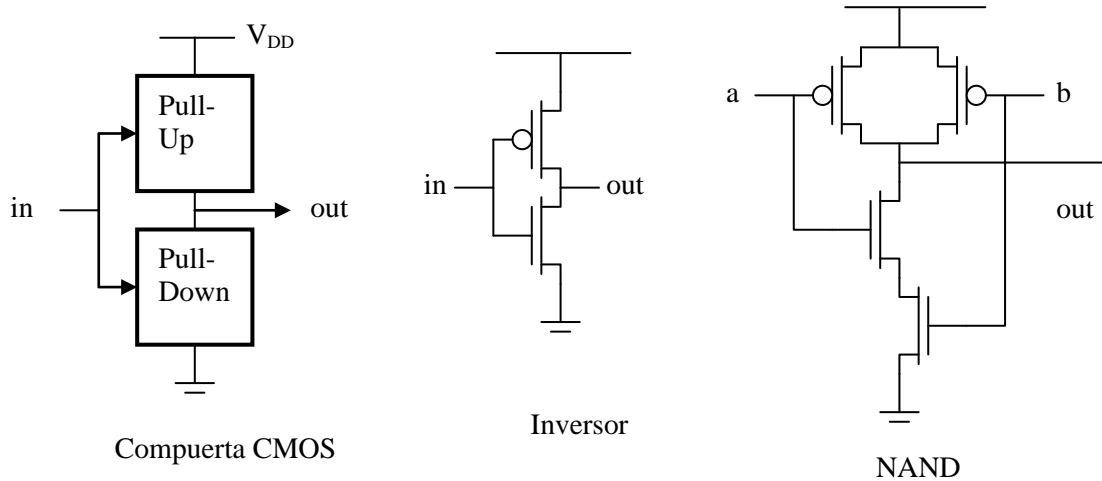


Figura A4.27 Compuertas CMOS.

Las redes pull-up y pull-down son siempre duales. Para construir una red dual, se efectúan las siguientes transformaciones:

- Intercambiar nfets por pfets (y vice versa)
- Intercambiar conexiones serie por conexiones paralelo (y vice versa)

La transformación serie/paralelo requiere observar la estructura jerárquica de la red.

Por ejemplo, el dual de un par paralelo de pares en serie (4 transistores) es un par serie de pares paralelos. Con esta metodología para obtener duales, el diseño de una compuerta CMOS requiere diseñar solamente una de las redes (la de pull-up o la de pull-down), la otra se obtiene como la red dual.

La red de pull-up representa el cálculo de una función con salida alta; y la de pull-down calcula salidas bajas. Un pfet en la red de pull-up es activado por entradas bajas, mientras que un nfet en la red de pull-down es activado por entradas altas. Entonces, deben colocarse transistores en serie para representar condiciones and; y en paralelo, para representar condiciones or.

Por ejemplo, en el nand que se mostró antes, que calcula $(ab)'$, la salida es cero cuando las entradas a y b están ambas altas; por lo tanto la red de pull-down está formada por una conexión serie. En forma alternativa (dual) la función de salida es alta cuando a o b están bajas, entonces la red de pull-down es una conexión paralelo.

Retardo de conmutación.

El *retardo* de una compuerta estática CMOS se mide viendo cuan rápido puede conmutar su salida. En un modelo simplificado se asume que todos los transistores cambian de estado on a off cuando sus compuertas pasan por $V_{DD}/2$. El retardo de una compuerta es medido desde el instante en que el voltaje de compuerta pasa por $V_{DD}/2$ hasta el instante en que la salida pasa por el valor $V_{DD}/2$.

Veremos cómo una compuerta conmuta su salida. En cualquier instante la red de pull-up o de pull-down está conduciendo, creando con esto una trayectoria eléctrica desde V_{DD} o GND (respectivamente) hacia la salida. Esta trayectoria es una fuente de corriente que carga o descarga la capacitancia de carga conectada a la salida. Es decir la capacidad de los cables o de las entradas conectadas a esa salida. Debe considerarse que las entradas de compuertas CMOS estáticas son las compuertas (gates) de los transistores que pueden tratarse como condensadores. La conmutación de la salida de una compuerta puede ser modelada por una red RC. La resistencia es el transistor conduciendo, y el condensador proviene de las compuertas que están conectadas a esa salida.

El retardo es proporcional a RC (la constante de tiempo), a mayor R y mayor C, mayor es el retardo. La configuración de los transistores determina la resistencia total. Los condensadores en paralelo, suman su capacidad; de tal modo que la capacidad total, depende de cuántas compuertas estén conectadas a esa salida. Esto implica que una salida que alimenta a una sola compuerta es n veces más rápida que una salida que tiene n compuertas conectadas a dicha salida; ya que en este último caso la capacitancia de carga será n veces mayor.

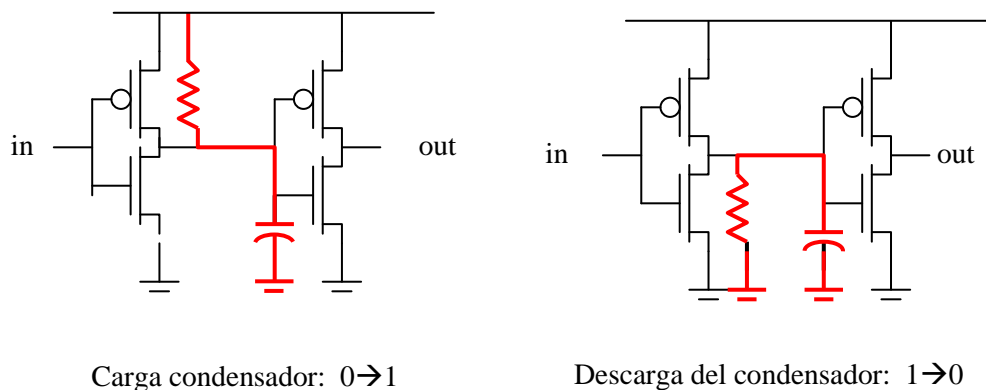


Figura A4.28 Redes RC en compuertas CMOS.

Compuertas CMOS de transmisión.

Debe notarse que el diseño de compuertas CMOS (mostrado anteriormente), es bastante diferente del diseño de circuitos en base a compuertas de transmisión.

Una *compuerta de transmisión* está formada por un transistor nfet en paralelo con un pfet, controlados por señales de compuerta complementarias. Una compuerta de transmisión deja

pasar tanto un 0 como un 1; por esta razón puede emplearse como un interruptor (switch). No se emplean redes de pull-up y pull-down como redes duales.

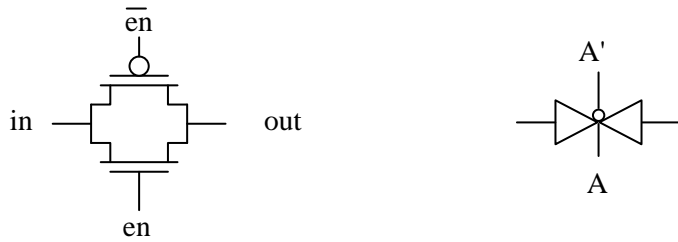


Figura A4.29 Compuerta CMOS de transmisión

in	enable	V _{GS} nmos	enable'	V _{GS} pmos	nmos	pmos
0	0	0	1	1	off	off
1	0	negativo	1	0	off	off
0	1	1	0	0	on	off
1	1	0	0	negativo	off	on

Figura A4.30 Valores lógicos en compuertas de transmisión.

Operación de la compuerta de transmisión:

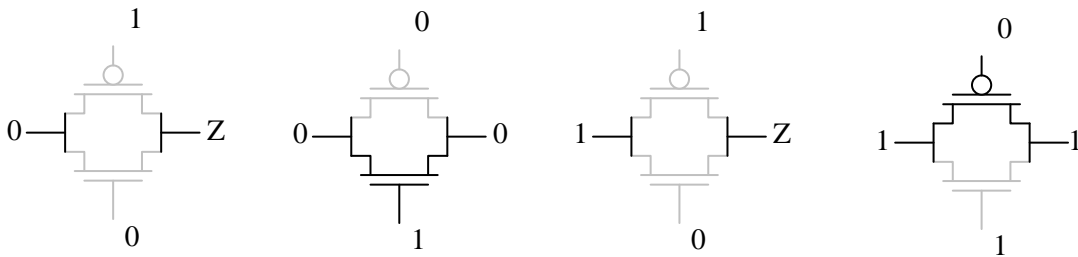


Figura A4.31 Operación de compuerta de transmisión CMOS.

El transistor pMOS funciona bien transmitiendo un voltaje alto, sin pérdida de señal, pero cuando se aplica un nivel 0, en el otro extremo se tendrá un nivel un poco mayor que cero. El nMOS, funciona bien dejando pasar el nivel 0, pero degrada el 1.

En los circuitos lógicos, vistos antes, un pMOS deja pasar los unos en la red de pull-up; y los nMOS en las redes de pull-down, dejan pasar los ceros.

La compuerta de transmisión deja pasar el nivel bajo y el nivel alto.

En diagramas suele emplearse el símbolo que se muestra en la figura A4.28 a la derecha.

Operación de compuertas CMOS.

Inversor CMOS.

En el siguiente diagrama, que emplea dos transistores MOS complementarios: Cuando X es 0V (valor lógico 0), el transistor p conduce, y el transistor n está abierto. Cuando X es 3V (valor lógico 1), el transistor p está abierto, el n conduce.

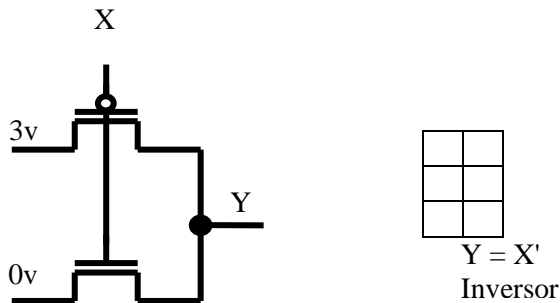


Figura A4.32 Operación de inversor CMOS.

Los siguientes diagramas ilustran la operación del inversor.

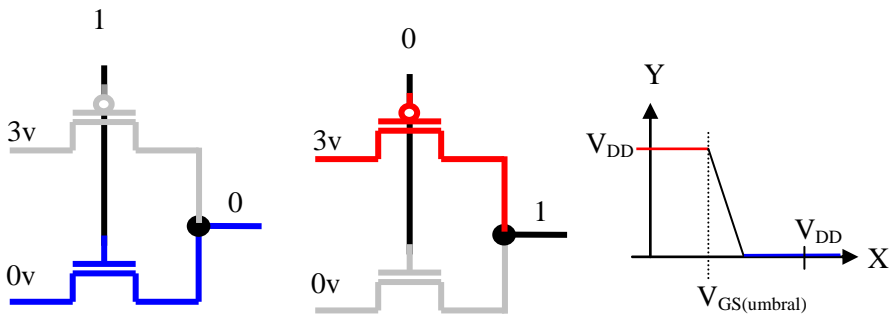


Figura A4.33 Operación de inversor CMOS.

Diseño electrónico inversor CMOS

Si se redibuja el inversor, identificando las fuentes y sumideros para cada transistor, se logra, empleando subíndices u por pull-up y d por pull-down:

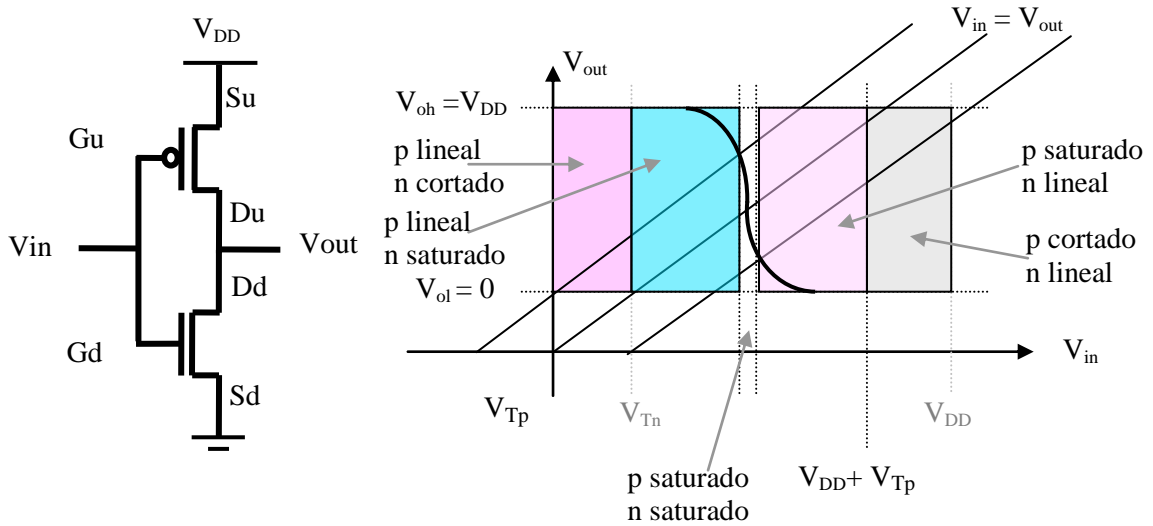


Figura A4.34 Operación de inversor CMOS.

Considerando las siguientes definiciones:

$$\begin{aligned} V_{GSp} &= V_{in} - V_{DD} & V_{DSp} &= V_{out} - V_{DD} \\ V_{GSn} &= V_{in} & V_{DSn} &= V_{out} \end{aligned}$$

Se determina el estado de cada transistor.

De la ecuación, en estado estacionario: $I_{DSd}(V_{in}, V_{out}) = -I_{DSu}(V_{in}-V_{DD}, V_{out}-V_{DD})$ pueden diseñarse los detalles de la característica de transferencia.

NAND CMOS

En el siguiente circuito cuando a e b están altos (uno lógico) ambos transistores n conducen, dejando en nivel bajo a la salida f (0 lógico). Se implementa un nand de dos entradas.

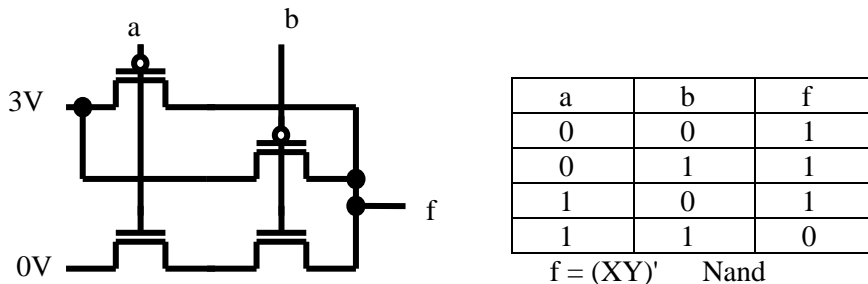


Figura A4.35 Diseño NAND CMOS.

Los siguientes diagramas ilustran la operación de un NAND.

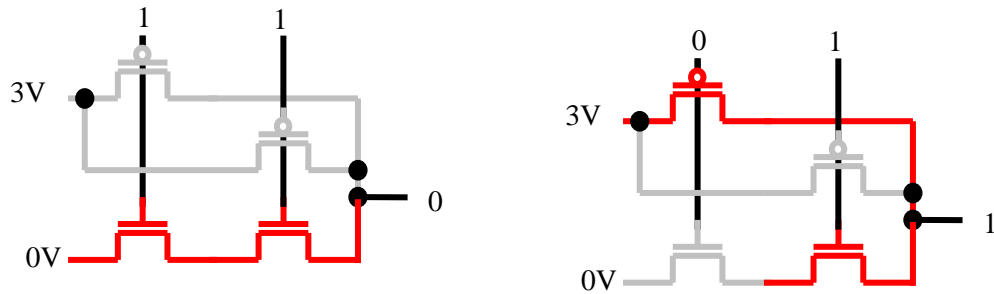
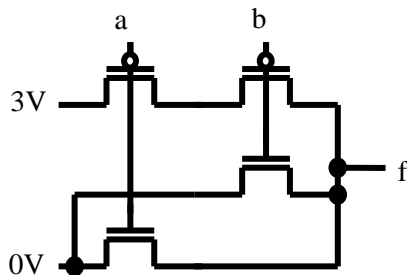


Figura A4.36 Operación NAND CMOS.

NOR CMOS.

Cuando a e b son ceros, ambos transistores p conducen estableciendo un voltaje alto en la salida f.



a	b	f
0	0	1
0	1	0
1	0	0
1	1	0

$$f = (X+Y)' \text{ Nor}$$

Figura A4.37 Estructura NOR CMOS.

Los siguientes diagramas ilustran la operación de un nor.

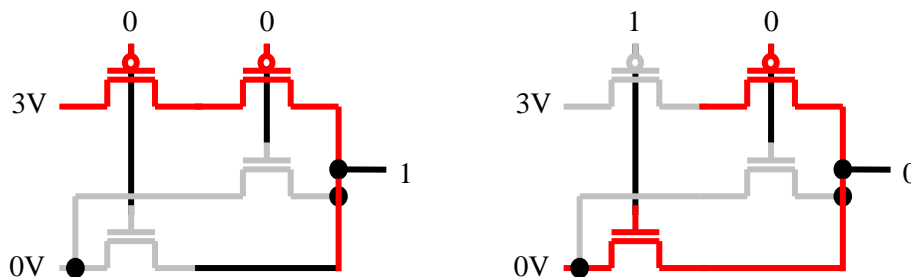


Figura A4.38 Operación NOR CMOS.

Nótese que el inversor requiere 2 interruptores, el nor y el nand requieren 4 interruptores. Puede comprobarse que el or y el and requieren 6 interruptores, y el xor puede diseñarse empleando 12. Por esta razón los diseños suelen estar basados en circuitos NOR o en circuitos NAND.

Ejemplo A4.2.

Diseñar en CMOS la compuerta $F = (A(B+C))'$.

La red de pull-down se logra con el and (en serie) de A con (B+C).
El or se logra con la conexión paralela de los transistores:

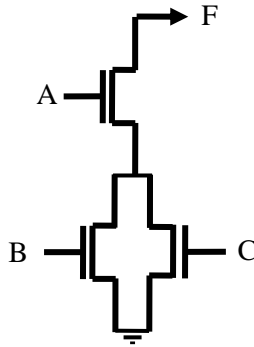


Figura A4.39 Diseño del pull-down.

La red de pull-up, es simplemente el dual de la anterior:

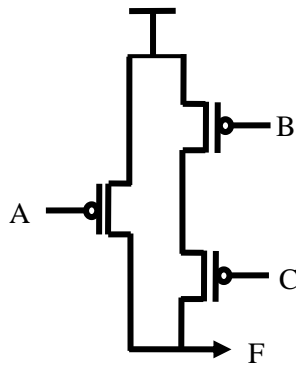


Figura A4.40 Diseño del pull-up.

Luego se combinan las redes de pull-up y pull-down, para formar la compuerta CMOS (complementaria).

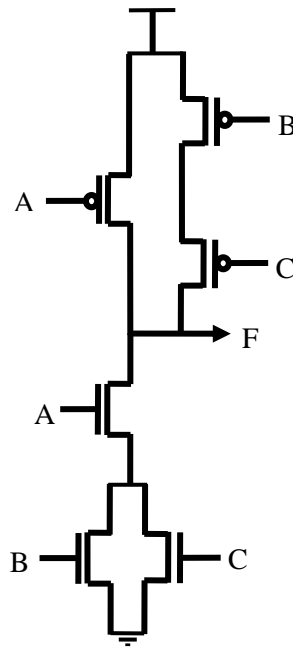


Figura A4.41 Integración de las partes del diseño.

Buffer inversor de tercer estado, en base a transistores CMOS.

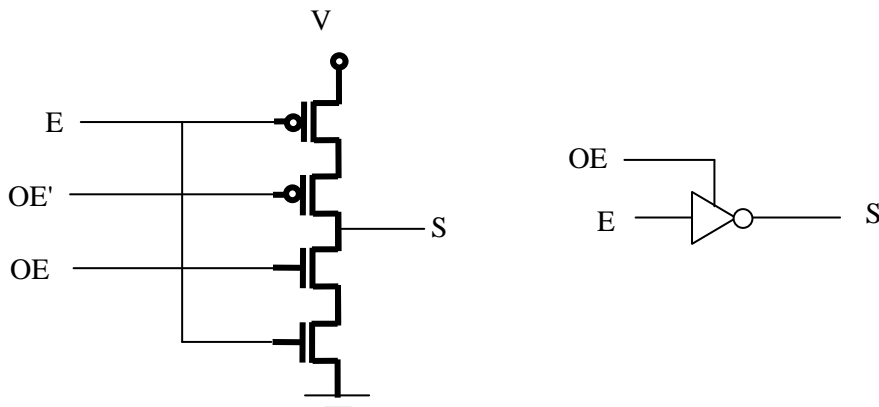


Figura A4.42. Buffer inversor mediante compuertas CMOS.

Cuando la salida no está habilitada, ambos transistores ubicados en la salida quedan desconectados, formando una salida en alta impedancia. Esto se muestra en el esquemático a la izquierda. El de la derecha ilustra que con salida habilitada ambos transistores, ubicados al lado de la salida conducen; el pMOS deja pasar un valor alto cuando E es un valor bajo, en este caso el nMOS conectado a tierra queda abierto.

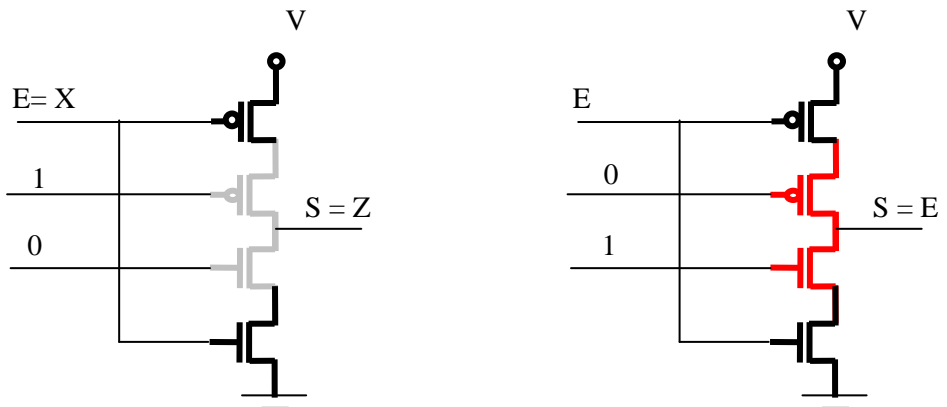


Figura A4.43. Operación buffer tercer estado.

Ejemplos de buffers de tercer estado también se encuentran en la familia TTL, ver por ejemplo los buffers 74240 y 74244.

Mux con compuertas de transmisión.

Para el diseño electrónico de multiplexores y demultiplexores resulta más eficiente emplear compuertas de transmisión CMOS, ya que se reduce el número de transistores necesarios. Este tipo de electrónica no está disponible mediante compuertas TTL. Un esquema de la compuerta de transmisión se muestra a continuación:

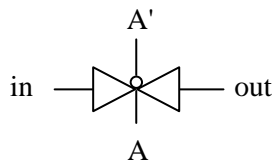


Figura A4.44. Compuerta de transmisión.

Para un multiplexor de dos vías a uno, se puede diseñar el siguiente circuito:

Cuando $A = 0$ conduce la compuerta de transmisión superior; la inferior no lo hace. En este tipo de diseño las señales de control deben activar una y sólo una vía de transmisión entre la entrada y la salida.

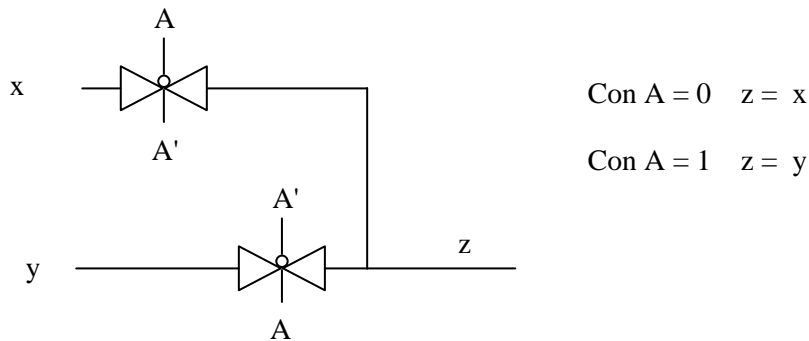


Figura A4.45. Multiplexor de 2 vías a una.

Un demultiplexer, puede diseñarse según:

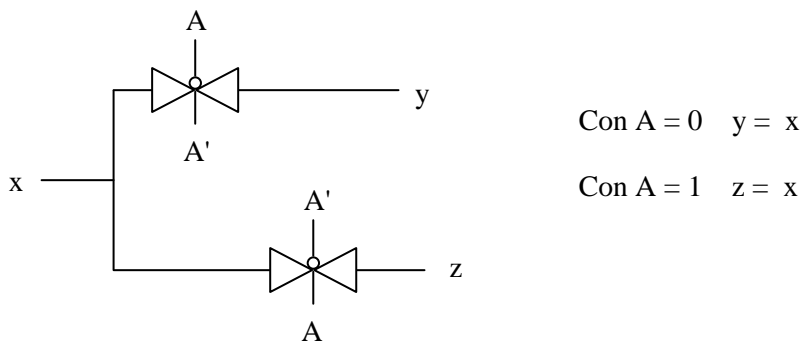


Figura A4.46. Demultiplexor de 1 a dos vías.

La dificultad del diseño anterior, es el valor que toma z cuando A es cero (z tomaría un valor que no es uno ni cero; sería una entrada en alta impedancia).

Para evitar esto: se especifica que con $A = 0$, se tiene que $z = 0$; y con $A = 1$, se tiene $y = 0$.

Para resolver esta dificultad, el diseño debe modificarse según:

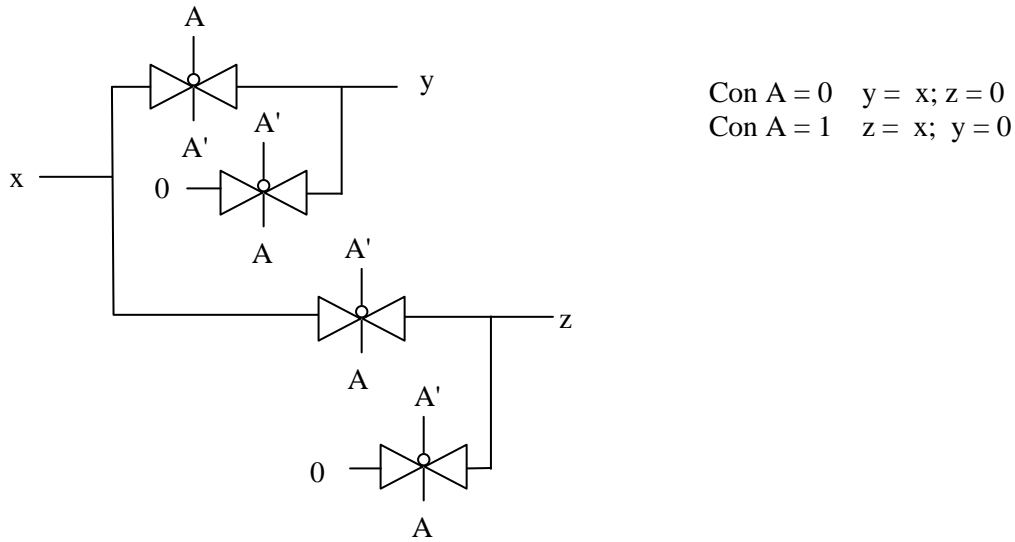


Figura A4.47. Revisión del diseño del demultiplexor.

Matriz de diodos con Mosfet.

Las explicaciones simplificadas del funcionamiento de los arreglos, en base a diodos, son con fines ilustrativos de conceptos solamente. La tecnología implementa los fusibles programables (diodos) empleando MOSFET. Como se ilustra a continuación.

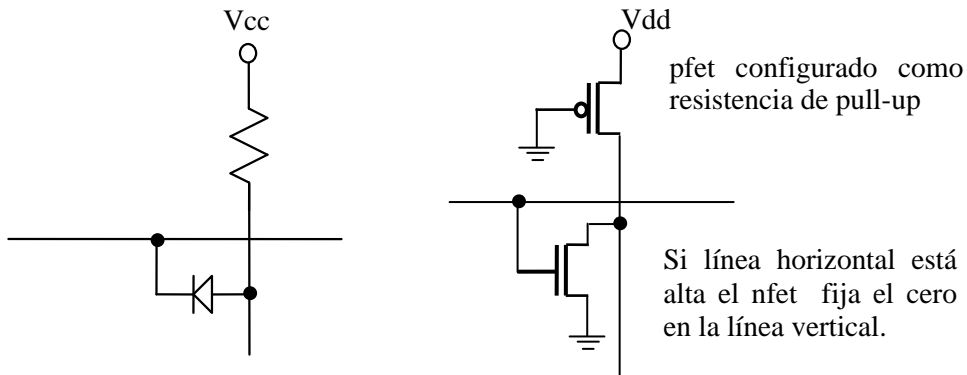


Figura A4.48 Fusibles programables con MOSFET.

La estructura de las PROM y PLA, en base a MOSFET son similares. En la PROM el arreglo de AND es un decodificador de direcciones (las líneas horizontales son las líneas de palabra, y sólo una puede estar a tierra para una determinada entrada); en una PLA cada línea horizontal es una línea de producto. El número de líneas verticales da el ancho de la palabra en una PROM; en una PLA genera el OR de los productos.

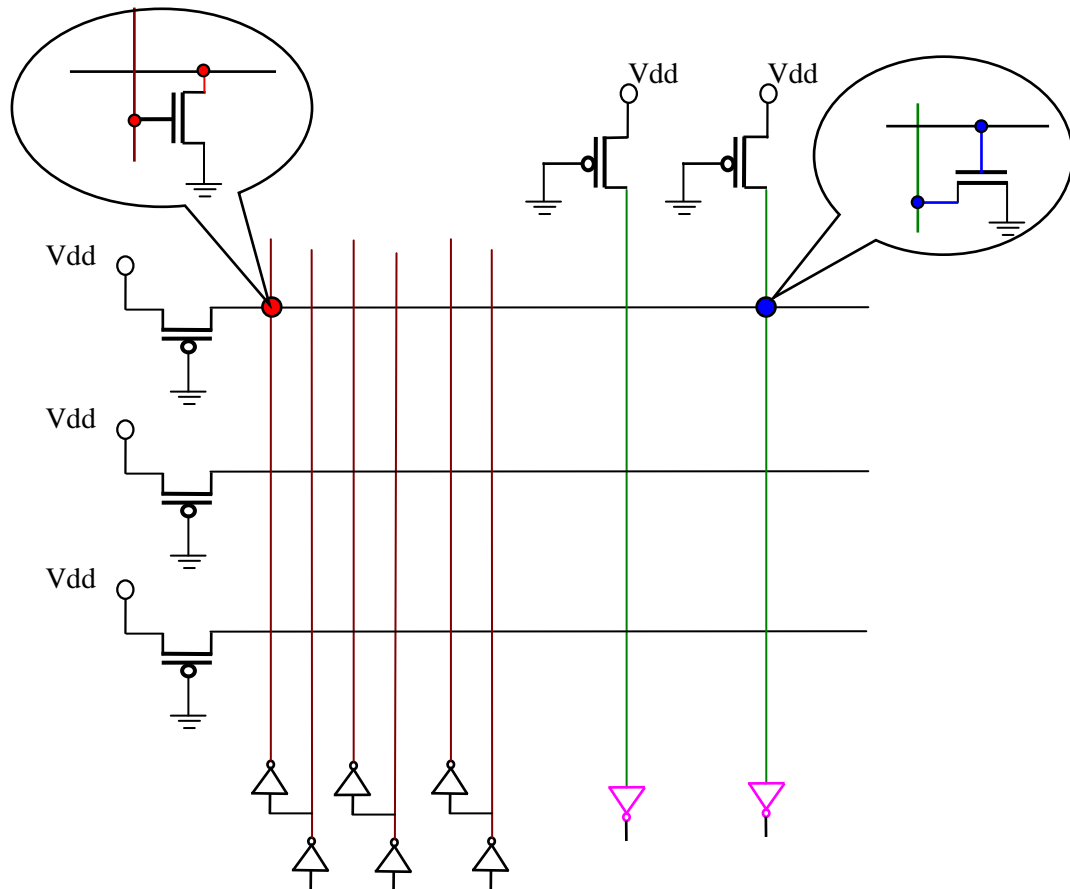


Figura A4.49 PLA y PROM mediante CMOS.

Si una variable de entrada está baja, la línea vertical está alta, el **nmos** fija la línea horizontal en bajo. Todas las líneas de entrada deben estar altas (todos los nmos conectados están abiertos) para que la línea horizontal esté alta.

Basta que una línea horizontal esté alta para que la **salida** esté baja, ya que el **nmos** conduce. Si todas las líneas horizontales están bajas, la tensión de **salida** estará alta. Esto implementa la función nor, por esta razón se agrega un **inversor**, para generar el or.

A4.9. Formato JEDEC para dispositivos programables.

Veremos a través de un ejemplo simple, la información estandarizada para describir el estado de los fusibles en una arquitectura programable. Se describirá empleando la arquitectura del dispositivo GAL22V10.

La Figura A4.50, describe dos funciones booleanas de tres variables. Realizando un diseño convencional, se obtienen las ecuaciones minimizadas para el diseño.

¡Error! No se pueden crear objetos modificando códigos de campo.

Figura A4.50 Funciones D y E de tres variables.

Resultan las ecuaciones:

$$D = A' B' + A B + C'$$

$$E = A B + B' C'$$

Relación entre la información que suministra el compilador y la arquitectura de la GAL.

En la Figura A4.51, se muestra sólo la parte necesaria de la estructura.

El dispositivo tiene 11 entradas y 10 salidas realimentadas que son las columnas de la matriz. Sólo se muestran los pines asignados a señales de entrada y el tipo de salida.

Como se tiene la señal y su complemento, existen 44 columnas. En el archivo Jedec se numeran las líneas de productos, con el número del fusible inicial de la línea. Estos comienzan en cero (el producto de reset); luego en 44 comienza el producto que habilita la primera macrocelda y luego, la segunda macrocelda en 88 y así sucesivamente. En 5784 está el producto de preset. En 5808 comienzan los fusibles que programan la salida combinacional o de registro. Del 5828 a 5891 los para la firma electrónica. Se marcan en el arreglo de fusibles los diodos que quedan en las líneas de productos.

El archivo jedec, que contiene el diseño y generado por un programa, es como sigue:

Type: PAL22V10

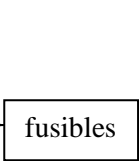
*

QP24* QF5828* QV1024*

F0*

```

L0044 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *
L0088 11 11 10 11 10 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *
L0132 11 11 01 11 01 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *
L0176 11 11 11 11 11 11 10 11 11 11 11 11 11 11 11 11 11 11 11 11 *
L0440 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *
L0484 11 11 01 11 01 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *
L0528 11 11 11 11 10 11 10 11 11 11 11 11 11 11 11 11 11 11 11 11 *
L5808 11 11 11 11 11 11 11 11 11 11 *
C2770*
```



Pin	Signal name	Column	Rows			Activity

			Beg	Avail	Used	
1.	-	0	-	-	-	(Clock)
2.	A	4	-	-	-	High
3.	B	8	-	-	-	High
4.	C	12	-	-	-	High
22.	E	6	10	11	2	High (Three-state)
23.	D	2	1	9	3	High (Three-state)

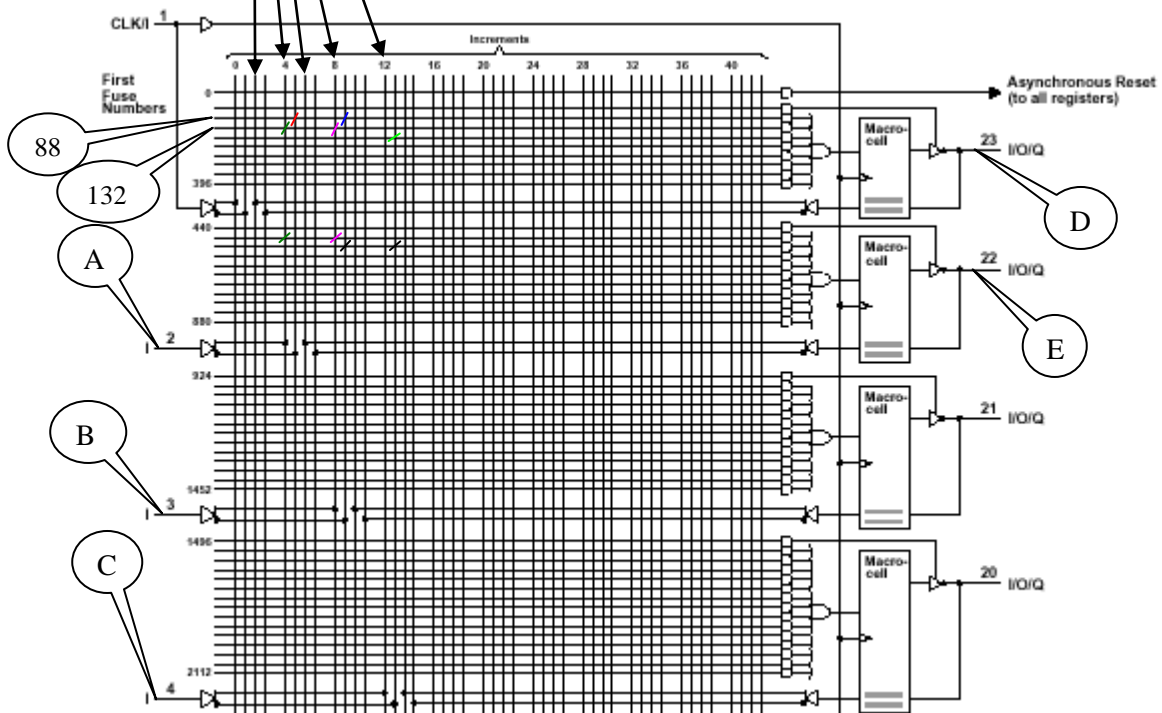


Figura A4.51 Esquema GAL22V10.

La siguiente información muestra los diferentes productos

Señal	Renglón	Productos
D	2	A' B' queda asociado al producto que comienza en 88.
	3	A B producto que comienza en fusible 132.
	4	C' producto que comienza en fusible 176
E	11	A B producto que comienza en fusible 484
	12	B' C' producto que comienza en fusible 528

Se destacan los fusibles que quedan con un cero, que identifican las variables en cada producto.

L0088 11 11 10 11 10 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *

L0132 11 11 01 11 01 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *

L0176 11 11 11 11 11 11 11 10 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *

L0484 11 11 01 11 01 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *

L0528 11 11 11 11 11 10 11 10 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *

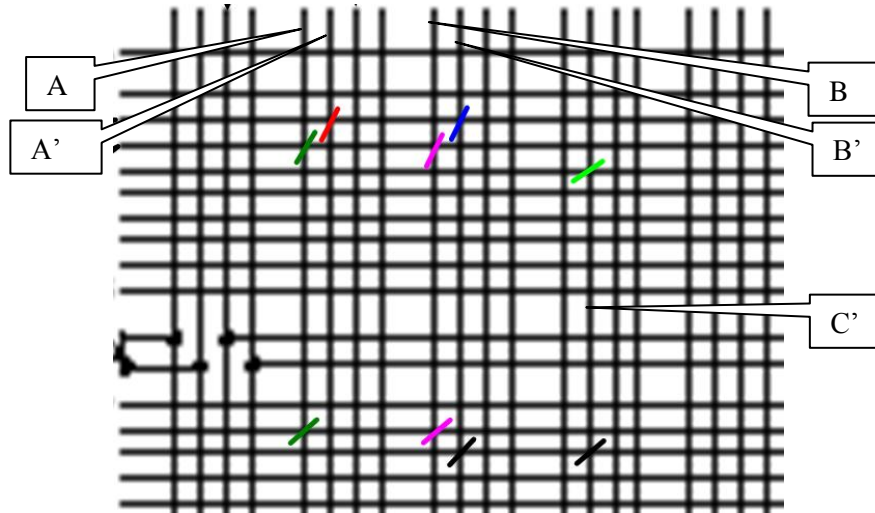


Figura A4.52 Detalle de fusibles en productos.

Los productos que habilitan el **tercer estado** están todos abiertos (L0044 y L0440)

L0044 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *

L0440 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *

Los fusibles que controlan si la macrocelda tendrá **salida combinacional o registrada** comienzan en la línea de fusibles L5808.

L5808 11 11 11 11 11 11 11 11 11 11 11 *

S0 y S1 para la macrocelda con salida por el pin 23 son el 5808 y 5809.

S0 y S1 para la macrocelda con salida por el pin 22 son el 5810 y 5811.

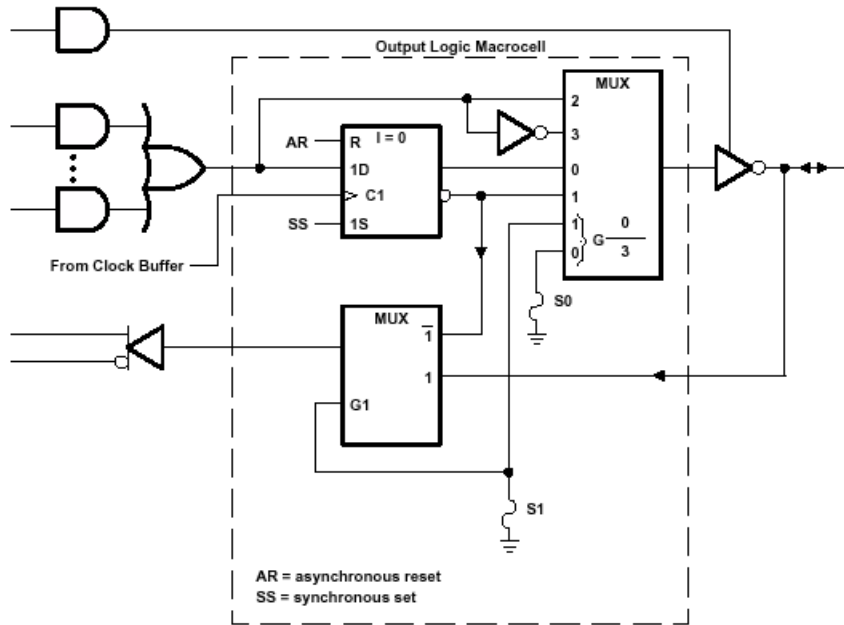


Figura A4.52 Programación de macrocelda de GAL22V10.

Para verificar el diseño y su grabación en el dispositivo, es necesario generar vectores de prueba. Es decir, para cada combinación de valores de las entradas se establece el valor que debe tener la salida.

Mediante esta información es posible verificar funcionalmente la grabación de la GAL. Esto se efectúa usando un programa que viene con el grabador de GAL; el cual pide ingresar los vectores, que suelen agregarse en el mismo archivo jedec en que se establece el estado de los fusibles. Se efectúa una prueba para cada uno de los vectores que se hayan generado.

Para el caso del ejemplo, la estructura del vector tiene 24 caracteres por línea. Uno por cada pin del dispositivo.

Vector: "0",A,B,C,"0000000N000000000", (E,D)L,"N"

Comenzando en 1. Tierra y VCC no tienen especificación y se anotan con una N.

Donde no se efectúa pruebas ni se mide las salidas se marca con 0.

Luego el nombre lógico de las entradas y salidas. Éstas últimas se marcan con L, para indicar que en estos pines debe medirse un valor. En las salidas en los vectores se coloca L (low) y H (high). En las entradas se colocan 0 ó 1s. El compilador calcula los valores lógicos que deben tener las salidas, de acuerdo a las ecuaciones del diseño y coloca L o H según corresponda. En las entradas coloca unos o ceros de acuerdo con los valores que vayan generando para la secuencia de entrada.

Un vector cualquiera, por ejemplo el 5
V00005 0100000000N000000000HHN*

Se interpreta así:

Vector: "0",A,B,C,"0000000N000000000", (E,D)L,"N"

V00005 0 1 0 0 0000000N000000000 HH N*

Es decir se aplica 5 volts en pin 2 (A), 0 en pines 3 y 4 (B y C) y deben medirse voltajes altos en pines 22 y 23 (E y D). En caso que lo medido sea distinto de lo especificado en el vector se genera una condición de error.

Para el caso del ejemplo, se generan todas las combinaciones posibles para tres entradas, y se establecen los valores que deben tener las salidas en cada caso.

```
V00001 00000000000N000000000HHN*
V00002 00010000000N000000000LHN*
V00003 00100000000N000000000LHN*
V00004 00110000000N000000000LLN*
V00005 01000000000N000000000HHN*
V00006 01010000000N000000000LLN*
V00007 01100000000N000000000HHN*
V00008 01110000000N000000000HHN*
A289
```

vectores

A4.10 Uso de manipulador de expresiones simbólicas. Maple.

En el programa Maple (versiones anteriores a 6) existe una biblioteca lógica para manipular expresiones booleanas. Para emplearla debe usarse `with(logic)`; antes de invocar a las funciones.

Se dispone de los siguientes operadores: `&and`, `&or`, `¬`, `&ifff`, `&nor`, `&nand`, `&xor`, y `&implies`.

Para obtener información detallada sobre la biblioteca, puede invocarse `Help`, y luego: `Introduction`. Estando abierta la ventana, en el primer nivel se elige `Programming`, en el segundo: `logic`. En el tercer nivel del árbol puede seleccionarse `Boolean`, o `Logic package`. Luego ver las funciones disponibles.

Para invocar ayuda específica se escribe: `?logic`, función.

Las expresiones booleanas se escriben precediendo los operadores con el signo `&`.

Se dispone de las siguientes funciones:

`Distrib(b)`; expande en suma de productos la expresión `b`.

`Bsimp(b)`; retorna una suma mínima de productos para la expresión `b`.

`Dual(b)`; obtiene la expresión dual de la expresión `b`.

`Bequal(b1, b2)`; retorna valor verdadero si dos expresiones lógicas `b1` y `b2` son equivalentes.

`canon(b, alpha, forma)` retorna la forma canónica de la expresión `b`, con los valores de la lista de variables; y si el argumento `forma` es `CNF` (retorna la forma conjuntiva) y si es `DNF` retorna la forma disyuntiva.

`convert(b, forma)`; cambia la forma de despliegue de la expresión `b`. Si `forma` es `frominert` representa los operadores sin el ampersand (forma en que se está más habituado a escribir expresiones booleanas). Si la forma es `toinert`, se representan operadores precedidos del símbolo `&`.

`tautology(b)`; retorna verdadero si la expresión `b` es una tautología.

A continuación se ilustra una sesión, donde se ilustran las funciones:

```
> with(logic); [bequal, bsimp, canon, convert/frominert, convert/toinert, distrib, dual,
tautology]
> bsimp(a &or (a &and b));
      a
> bsimp((a &and b) &or (a &and (&not b)));
      a
> distrib(&not(a &and b));
      &not(b) &or &not(a)
> distrib(&and(a,b &or c));
      (a &and b) &or (a &and c)
> dual(a &and (&not a) = false);
      a &or &not(a) = true
> dual(a &implies b);
      a &implies b
> bequal(&not(a &and b),(&not a) &or (&not b));
      true
> canon(a &xor b,{a,b},CNF);
      (&not(a) &or &not(b)) &and (a &or b)
> canon(a &xor b,{a,b},DNF);
      (a &and &not(b)) &or (b &and &not(a))
> convert(&and(a,b,c) &or b,frominert);
      a and b and c or b
> convert(&or(a,b,c) &and b,frominert);
      (a or b or c) and b
> convert(a and b or c,toinert);
      c &or (a &and b)
> tautology(&and(a,b) &or (&not a) &or (&not b));
      true
```


Índice general.

APÉNDICE 4 1

CIRCUITOS DE CONMUTACIÓN 1

CIRCUITOS LÓGICOS..... 1

A4.2 INTERRUPTORES MANUALES..... 1

 A4.2.1 Circuito and. 1

 A4.2.2 Circuito or. 2

 A4.2.3 Circuito escalera. 3

A4.3 RELÉS..... 4

 A4.3.1. Función de transmisión. 4

 A4.3.2. Función de apertura. 5

 Ejemplo A4.1.....5

 A4.3.3. Controladores lógicos programables(PLC)..... 6

A4.4 CIRCUITOS LÓGICOS EMPLEANDO DIODOS. 6

A4.5 TRANSISTOR BIPOLAR. 7

A4.6 COMPUERTA NAND DTL. (DIODE TRANSISTOR LOGIC). 7

 Fan-out. 8

A4.7 COMPUERTAS TTL..... 9

 Niveles lógicos TTL 10

A4.8 COMPUERTAS CMOS 10

 Interruptores MOS..... 10

 Transistor nfet..... 11

 Transistor pfet..... 12

 Construcción nfet..... 12

 Operación 13

 Inversor MOS RTL..... 14

 Inversor CMOS..... 14

 Retardo de conmutación..... 16

 Compuertas CMOS de transmisión. 16

 Operación de compuertas CMOS..... 18

 Inversor CMOS..... 18

 Diseño electrónico inversor CMOS 18

 NAND CMOS 19

 NOR CMOS..... 20

 Ejemplo A4.2. 21

 Buffer inversor de tercer estado, en base a transistores CMOS..... 22

 Mux con compuertas de transmisión. 23

 Matriz de diodos con Mosfet..... 25

A4.9. FORMATO JEDEC PARA DISPOSITIVOS PROGRAMABLES. 26

 Relación entre la información que suministra el compilador y la arquitectura de la GAL..... 27

A4.10 USO DE MANIPULADOR DE EXPRESIONES SIMBÓLICAS. MAPLE..... 31

ÍNDICE GENERAL..... 33

ÍNDICE DE FIGURAS..... 34

Índice de Figuras

Figura A4.3 And con interruptores.....	1
Figura A4.4 Asignación de valores lógicos 1.....	1
Figura A4.5 Asignación de valores lógicos 2.....	2
Figura A4.6 Asignación de valores lógicos 3.....	2
Figura A4.7 Red or con interruptores.....	3
Figura A4.7 Red or con interruptores.....	3
Figura A4.8 Circuito escalera.....	3
Figura A4.9 Diagrama de un relé.....	4
Figura A4.10 Diagrama de contactos de un relé.....	4
Figura A4.11 Función lógica implementada con relés.....	4
Figura A4.12 Esquema simplificado empleando relés.....	5
Figura A4.13 Conexión puente empleando relés.....	5
Figura A4.14 Diagramas lógicos para PLC.....	6
Figura A4.15 Compuertas empleando diodos.....	6
Figura A4.16 Transistor bipolar.....	7
Figura A4.17 Compuerta NAND DTL.....	8
Figura A4.18 AND alambrado.....	8
Figura A4.19 NAND TTL.....	9
Figura A4.20 Salida totem-pole TTL.....	9
Figura A4.21 Canal n y p.....	10
Figura A4.22 Transistor nfet.....	11
Figura A4.23 Transistor pfet.....	12
Figura A4.24 Diagrama construcción transistor pfet.....	12
Figura A4.25 Diagrama físico transistor nfet.....	13
Figura A4.26 Inversor MOS RTL.....	14
Figura A4.27 Compuertas CMOS.....	15
Figura A4.28 Redes RC en compuertas CMOS.....	16
Figura A4.29 Compuerta CMOS de transmisión.....	17
Figura A4.30 Valores lógicos en compuertas de transmisión.....	17
Figura A4.31 Operación de compuerta de transmisión CMOS.....	17
Figura A4.32 Operación de inversor CMOS.....	18
Figura A4.33 Operación de inversor CMOS.....	18
Figura A4.34 Operación de inversor CMOS.....	19
Figura A4.35 Diseño NAND CMOS.....	19
Figura A4.36 Operación NAND CMOS.....	20
Figura A4.37 Estructura NOR CMOS.....	20
Figura A4.38 Operación NOR CMOS.....	20
Figura A4.39 Diseño del pull-down.....	21
Figura A4.40 Diseño del pull-up.....	21
Figura A4.41 Integración de las partes del diseño.....	22
Figura A4.42 Buffer inversor mediante compuertas CMOS.....	22
Figura A4.43. Operación buffer tercer estado.....	23
Figura A4.44. Compuerta de transmisión.....	23

Figura A4.45. Multiplexor de 2 vías a una.....	24
Figura A4.46. Demultiplexor de 1 a dos vías.....	24
Figura A4.47. Revisión del diseño del demultiplexor.....	25
Figura A4.48 Fusibles programables con MOSFET.....	25
Figura A4.49 PLA y PROM mediante CMOS.....	26
Figura A4.50 Funciones D y E de tres variables.....	27
Figura A4.51 Esquema GAL22V10.....	28
Figura A4.52 Detalle de fusibles en productos.....	29
Figura A4.52 Programación de macrocelda de GAL22V10.....	30