

LÓGICA COMBINACIONAL

Los sistemas digitales son construidos con circuitos o dispositivos de conmutación de dos estados. Los circuitos de conmutación se pueden clasificar en circuitos combinacionales y circuitos secuenciales.

Como se ha visto, el uso del álgebra de Boole en el análisis y diseño de circuitos de conmutación algunas veces es denominado teoría o álgebra de conmutación, y los circuitos de conmutación o combinacionales son también denominados circuitos lógicos.

La lógica combinacional trata del análisis, diseño y construcción de circuitos cuyas salidas son función lógica de las entradas.

3.1 CIRCUITOS COMBINATORIOS.

La concepción de circuitos combinacionales implica varias etapas, tres de estas están bien diferenciadas y son:

- La generación de la expresión booleana correspondiente a la función lógica deseada que satisface las condiciones salida / entrada o tabla de la verdad.
- La simplificación de la expresión booleana, a fin de obtener el circuito lógico más sencillo posible.
- Si es deseable, la búsqueda de una expresión que permita realizar el circuito con un conjunto específico de compuertas u operadores.

Actualmente, la tercera etapa es llevada a cabo fácilmente con la ayuda de sistemas CAD, los cuales permiten cumplir con esta restricción impuesta por la utilización de ciertos circuitos integrados.

A continuación, se presentarán algunas consideraciones para el diseño o realización de circuitos lógicos.

3.1.1 EXPRESIÓN DE UN CIRCUITO LÓGICO

La función lógica de un circuito de conmutación puede ser expresada en álgebra de Boole. En la figura 3-1 se muestra un circuito de activación de un solenoide o relé, es claro que los interruptores poseen dos posiciones o valores (cerrado o abierto) y el relé puede asumir uno de dos estados, activado o desactivado.

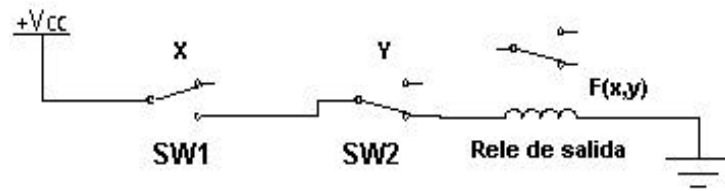


Figura 3-1 Circuito de conmutación

Por tanto podemos definir los parámetros del circuito de la forma siguiente:

- Interruptor SW1= variable X. Si el interruptor SW1 está cerrado: $X=1$, si el interruptor SW1 está abierto: $X=0$.
- Interruptor SW2 = variable Y. Si el interruptor SW2 está cerrado: $Y=0$, si el interruptor SW2 está abierto: $Y=1$
- Solenoide = F , solenoide desactivada: $F=0$, solenoide activada: $F=1$

La definición de las variables, arbitraria en este caso, nos permite definir la función $F(x,y)$ del estado de la solenoide como:

$$F(x,y) = X' + Y$$

Por supuesto, la forma de la expresión dependerá de la definición de las variables.

3.1.2 NIVEL DE ACTIVACIÓN

El uso adecuado de los símbolos alternativos de compuertas puede hacer mucho más clara la operación de un circuito cuando se tiene en cuenta el estado de activación de una entrada o salida.

Esto puede ilustrarse con la función F, tal que:

$$F = A' + B' = (AB)'$$

Esta función, tal como se representa en la figura 3-2, puede expresarse literalmente como: "La salida será ALTA (uno) sólo cuando cualquier de sus entradas sea BAJA (cero)". En este caso el circuito se dice que tiene la salida activa en alto y las entradas activas en bajo.

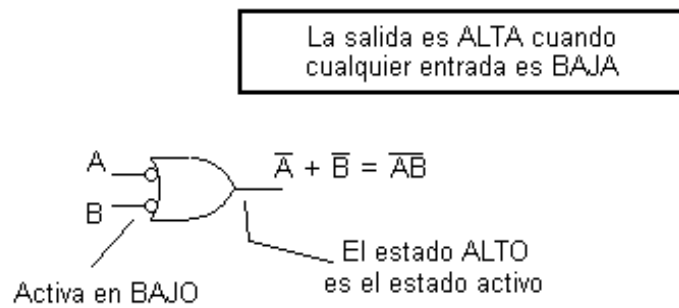


Figura 3-2 Representación de $F = A' + B'$

Otra forma de entender la función es: "La salida será BAJA sólo cuando las entradas sean ALTAS". En este caso el circuito más conveniente tiene la salida activa en bajo y las entradas activas en alto tal como se observa en la figura 3-3.

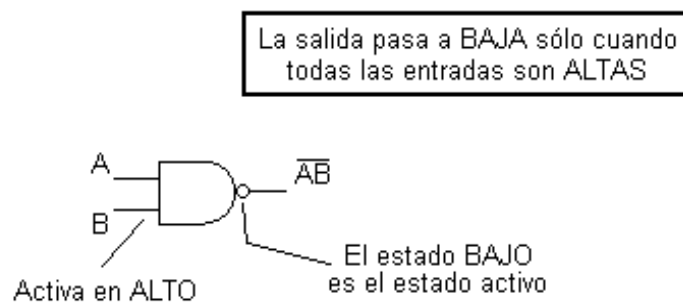


Figura 3-3 Representación de $F = (AB)'$

Cada una de estas representaciones es adecuada para su expresión literal o asociación correspondiente.

3.1.3 CIRCUITOS AND-OR Y OR-AND

Las expresiones en forma de suma de productos, con al menos dos términos de dos variables cada uno, se realizan directamente con circuitos denominados **AND-OR**. Esto en el supuesto de que existan las variables complementadas, caso contrario serán necesarias también compuertas **NOT**. Primeramente se realizan los productos de las variables utilizando compuertas AND, luego el resultado de los productos se suman usando compuertas OR.

Similarmente, las expresiones en producto de sumas se dan lugar a circuitos **OR-AND**.

EJEMPLO 3-1

Sea la función:

$$F(Z,Y,X) = \sum m_i = \sum m(2,3,6)$$

La función puede expresarse en términos canónicos o suma de productos estándar como:

$$F(Z,Y,X) = Z'YX' + ZYX' + Z'YX$$

Simplificando:

$$F(Z,Y,X) = YX' + YZ'$$

También, F puede expresarse como producto de sumas estándar:

$$F(Z,Y,X) = (Z+Y+X) (Z+Y+X') (Z'+Y+X) (Z'+Y+X') (Z'+Y'+X)'$$

Simplificando:

$$F(Z,Y,X) = Y(Z'+X')$$

En la figura 3-4 se muestran los circuitos AND-OR y OR-AND respectivos, con compuertas de dos entradas.

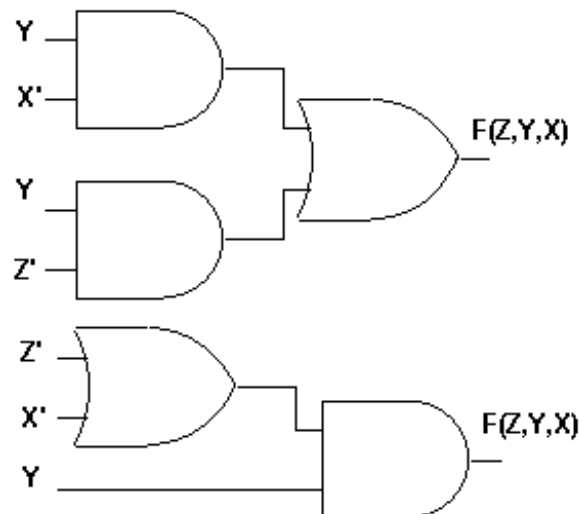


Figura 3-4 Representación de $F(Z,Y,X) = \sum m(2,3,6)$

Las dos expresiones obtenidas para $F(Z,Y,X)$ son equivalentes. Observando los circuitos de las expresiones simplificadas, se nota que el producto de sumas simplificado permite un circuito mínimo.

Para obtener el circuito con el menor número de compuertas básicas es necesario obtener los mínimos circuitos AND-OR y OR-AND y compararlos.

3.1.4 CIRCUITOS DE DOS O MÁS NIVELES

Las expresiones lógicas en la forma suma de productos o producto de sumas tienen la ventaja de poder ser llevadas a circuitos de compuertas básicas directamente. En el diseño de circuitos lógicos cuando se presupone disponible una variable de entrada, también se supone accesible directamente su complemento.

En los métodos de reducción también se asume esta condición, aún cuando en la implementación se genere el complemento de las variables utilizando compuertas NOT. Esto quiere decir, que los métodos de simplificación reducen los términos y las literales en una expresión pero no implican la reducción propiamente de las variables.

En la figura 3-4 se presentaron dos circuitos para la misma función, ambos son circuitos de dos niveles puesto que la trayectoria de las señales, desde la entrada hasta la salida, afecta o pasa por al menos dos compuertas.

En la figura 3-5 la expresión $A'B + B'C'D$ se construye con compuertas AND-OR; este circuito es de dos niveles, sin embargo, en caso de utilizar solo compuertas de dos entradas el circuito poseerá más de dos niveles.

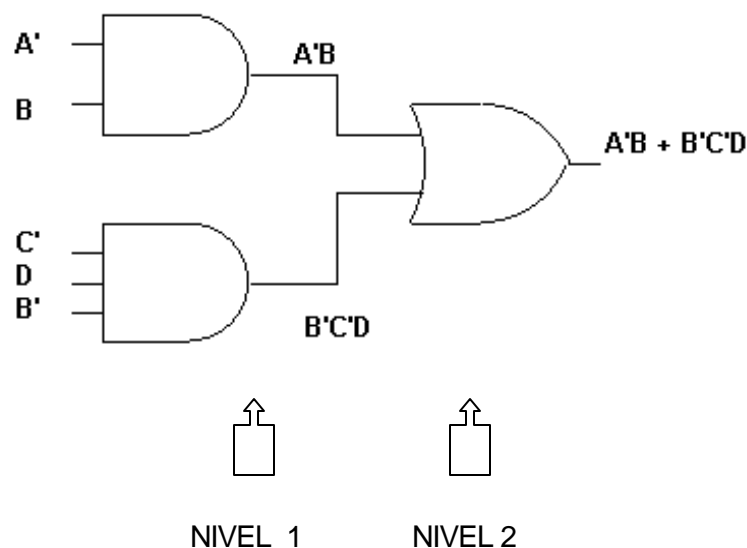


Figura 3-5 Circuito de dos niveles

Es de notar que si no se tiene disponible el complemento de una variable será necesario la incorporación de una compuerta NOT y el circuito ya no será de dos niveles.

3.1.5 SUMA DE PRODUCTOS Y CIRCUITOS NAND

Las compuertas Nand se usan ampliamente debido a su propiedad de universalidad. Las equivalencias de las compuertas Nand nos permite convertir un circuito de compuertas AND, OR y NOT en un circuito de solamente compuertas Nand.

El análisis de un circuito de dos niveles nos permite deducir un procedimiento para convertir de manera fácil circuitos de dos niveles AND-OR (S. P) en circuitos NAND-NAND y viceversa.

Sea la función:

$$G(edcba) = edc' + cb' + a$$

Cada término de la expresión es el producto de literales (compuerta AND) y la función G es la suma (compuerta OR) de estos términos.

Aplicando doble complemento a la expresión y los teoremas de De Morgan se tiene:

$$G(edcba) = [(edc' + cb' + a)']'$$

$$G(edcba) = [(edc')' (cb')' (a)']'$$

Cada uno de los términos dentro de los corchetes puede ser visto como el complemento de los productos de los literales originales (Compuerta Nand en lugar de AND) y la función G es el complemento del producto (compuerta Nand en lugar de OR) de estos términos.

Por tanto el procedimiento para diseño de circuitos utilizando solo compuertas Nand o para la conversión de circuitos AND-OR en NAND-NAND será:

- Diseñar el circuito en dos niveles AND-OR (suma de productos) y luego convertir el circuito sustituyendo las compuertas AND y OR por NAND de igual cantidad de entradas y escribiendo las mismas variables de entrada.
- En el caso de un término de una sola variable en la S. P; es decir una variable conectada directamente a la compuerta OR, ésta se debe complementar. Tal como la variable “a” del tercer término de la función G .

Siguiendo estas reglas, es fácil la conversión de dos niveles AND-OR / NAND-NAND, tal como se ilustra en la figura 3-6.

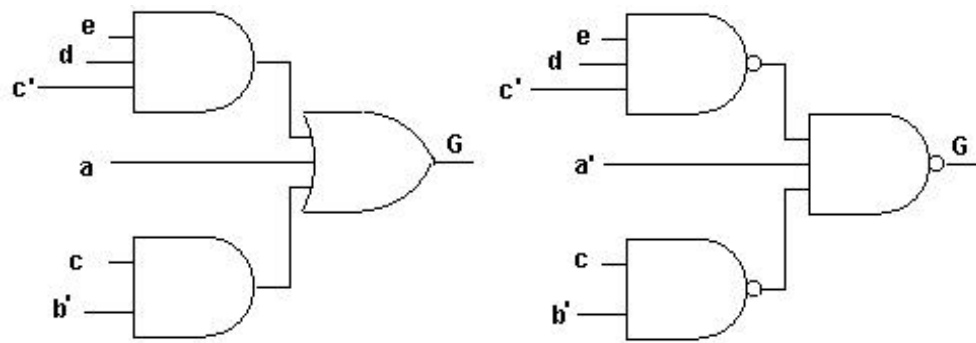


Figura 3-6 Circuito de dos niveles con NAND

3.1.6 PRODUCTO DE SUMAS Y CIRCUITOS NOR

El procedimiento expuesto para circuitos S.P. se puede extender para convertir circuitos de dos niveles OR-AND o Producto Sumas en compuertas Nor.

Sea la función:

$$H(edcba) = (e+d+c') (c+b') (a)$$

Cada término de la expresión es la suma de literales (compuerta OR), y la función H es el producto (compuerta AND) de estos términos.

Aplicando doble complemento a la expresión y luego los teoremas de De Morgan se tiene:

$$H(edcba) = \{ [(e+d+c') (c+b') (a)]' \}'$$

$$H(edcba) = \{ (e+d+c')' + (c+b')' + (a)' \}'$$

Cada uno de los términos dentro de las llaves puede ser visto como el complemento de la suma de los literales originales (Compuerta NOR en lugar de OR) y la función H es el complemento de la suma (compuerta NOR en lugar de AND) de estos términos.

Por tanto, un procedimiento para diseño de circuitos utilizando solo compuertas NOR es la conversión de circuitos OR-AND en NOR- NOR así:

- Diseñar el circuito en dos niveles OR-AND (Producto de Sumas) y luego volver a dibujar el circuito sustituyendo las compuertas OR y AND por compuertas NOR con igual cantidad de entradas y las mismas variables de entrada.

- En el caso de un término de una sola variable, es decir una variable conectada directamente a la compuerta OR, esta se debe complementar. Este es el caso de la variable "a" del tercer término de la función H.

En la figura 3-7 muestra la conversión OR-AND en NOR- NOR.

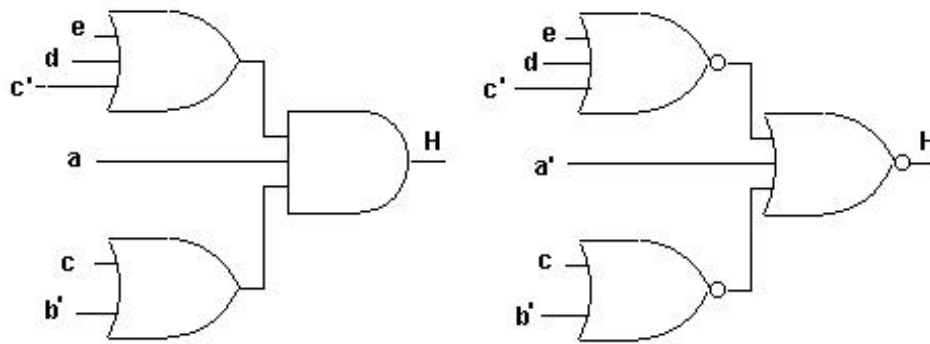


Figura 3-7 Circuito de dos niveles con NOR

3.2 SIMULACIÓN DE CIRCUITOS

En el mercado se dispone de múltiples programas especializados en realizar simulaciones de circuitos digitales, la mayoría realiza también simulación de circuitos analógicos. Entre los más utilizados tenemos:

- MicroSim PSpice A/D versión de evaluación 8.0.
- Electronics Workbench EDA 5.0.
- CircuitMaker.

En seguida se describirá brevemente las facilidades del MicroSim Pspice (versión Evaluación 8.0) y del Electronics Workbench (EWB versión 5.0), los cuales se utilizan para desarrollar ejercicios y proyectos en el presente texto. [3] [4]

3.2.1 MICROSIM PSPICE

Este muy conocido programa de simulación, permite simular circuitos análogos, digitales y mixtos, en el caso de circuitos digitales permite conocer la respuesta de los circuitos en forma de diagramas de tiempo. Esta aplicación se basa en el programa Spice creado en la Universidad de Berkeley en 1972, el cual se ha convertido en un estándar en la comunidad de la electrónica. A partir de este se han desarrollado diferentes versiones comerciales y del dominio publico. [3]

En realidad el MicroSim Pspice es un conjunto de programas de la empresa MicroSim que se deriva del Spice2G.6 (versión para BM). Los principales programas del MicroSim Pspice son:

- MicroSim Schematics: Realiza los diagramas de los circuitos para simular con Pspice A/D.
- MicroSim Stimulus Editor: Crea diferentes señales digitales.
- MicroSim PSpice A/D: Realiza las Simulaciones de los circuitos.
- MicroSim Probe: Permite visualizar los resultados obtenidos.

Schematics

□ Construcción del circuito digital

Para llevar a cabo la simulación de los circuitos, en primer lugar se dibuja el circuito en **Schematics**, obteniendo los dispositivos o C. Integrados a través el menú **Draw, Get New Part**. El Schematics proporciona los circuitos 74XX, como compuertas individuales ligadas al mismo integrado, pudiendo cambiar la selección de la compuerta a utilizar al hacer doble clic en el **Partname** del componente (generalmente U1A, U2A, etc.) y seleccionar en **Gate** la compuerta de interés (A,B,C o D). Al montar y alambrear el circuito en el Schematics, se debe aplicar la(s) entrada(s) al circuito. El modelo que posee Pspice proporciona operación en dos modos, analógico y digital.

El tipo de análisis y otros parámetros de simulación se definen en el menú **Análisis Setup**. Para realizar la simulación debe seleccionarse un análisis transitorio.

Después de construir el circuito, se ejecuta el comando **Simulate** el cual automáticamente ejecuta otros subprogramas (Electrical Rule Check y Create Netlist) para verificar la correcta conexión de las partes y componentes y crear el archivo **nombre.cir**. Luego de crear el archivo **.cir** (si no hay errores), se ejecuta automáticamente el **Pspice A/D** el cual simula el circuito bajo las condiciones especificadas y construye el archivo **nombre.dat** que puede ser usado por el programa **Probe** para realizar los gráficos de las señales eléctricas del circuito.

Por último, se ejecuta automáticamente Probe, si esto a sido seleccionado a través del menú **Analysis y Probe Setup**, pudiendose realizar los gráficos de las señales eléctricas que se deseen.

❑ Fuentes de señales digitales

Para realizar los diagramas de tiempo es necesario contar con un conjunto de señales digitales de entrada (variables binarias) que den todas las combinaciones posibles para caracterizar el circuito digital. Existen dos formas directas de crear señales digitales (source) **DigStim** y **DigClock**.

La fuente de señales **DigStim** puede crear cualquier tipo de señal digital por medio del programa Stimulus Editor (en la versión de evaluación solo se permite tener señales de sincronización o reloj). DigClock solo genera señales de reloj, indicando el periodo de la señal y el tiempo en activo (Duty Cycle).

Si se selecciona **DigStim**, al hacer doble click sobre este componente, el programa pedirá un nombre para la señal de entrada, lo cual abre la pantalla el editor de estímulos **StimEd**. Se muestra un tren de pulsos cuadrados con niveles de ALTO y BAJO, y aparece un cuadro de diálogo donde se puede seleccionar la frecuencia y el ciclo de servicio de la señal. Además, puede seleccionarse si se quiere que la señal comience en 0 ó 1. De esta manera puede seleccionarse la señal digital a aplicar en la entrada del circuito. Al cerrar el **StimEd**, se crea una librería de estímulos que debe salvarse para permitir el análisis de PSpice.

Si se utiliza el componente **DigClock**, al hacer doble click sobre el mismo, se abre un cuadro de diálogo donde se seleccionan las características de la señal:

- DELAY: Retardo de tiempo inicial en segundos.
- ONTIME: Tiempo (en segundos) durante el cual la señal permanece en alto.
- OFFTIME: Tiempo (en segundos) durante el cual la señal permanece en bajo.
- STARTVAL: Valor lógico de comienzo (cero o uno).
- OPPVAL: Valor lógico de final (cero o uno).

Así, el período de la señal es $ONTIME + OFFTIME$. Si se quiere un tren de pulsos, se selecciona STARTVAL y OPPVAL con valores lógicos opuestos. Si se quiere sólo un cero o uno lógico a la entrada durante todo el tiempo, simplemente se selecciona $STARTVAL = OPPVAL$.

Además de este tipo de señales de pulso digitales, puede generarse un ALTO o BAJO a través de los componentes **HI** y **LO**, que son dos puertos con valores lógicos definidos. También existe el puerto **X**, que es una condición “no importa”, esto es, esta señal no tiene un valor lógico definido.

Otra forma de crear las señales digitales consiste en crear un subcircuito generador de señales con “Draw Block”, creando un nuevo símbolo (Simbol) o parte (Part) dentro de una librería en el programa. Este método, tiene la ventaja de que permite utilizar el nuevo símbolo como si fuese un componente o circuito integrado dentro del programa, permitiendo utilizar el botón “Get New Part” o la ventana “Get Recent Part” para pegarlo en el esquemático, como cualquier otro componente.

Probe

En el análisis en modo digital se estimulará el circuito con ceros y unos lógicos, y se observará la salida en términos de valores discretos, obviando niveles de voltaje. El objetivo final de las simulaciones en los circuitos digitales es obtener los diagramas de tiempos.

□ Construcción de los gráficos

Para construir los gráficos de las señales de interés se seleccionan estas a través del menú **Trace, Add de Probe**. Así pueden verse las señales de entrada y salida del circuito.

Pueden obtenerse de manera adecuada los diagramas de tiempo del circuito si se seleccionan las diferentes señales de entrada de modo que la relación de frecuencias sea de dos, con lo que se abarcan todas las posibles combinaciones de las señales de entrada. Estimulando el circuito con trenes de pulso donde la frecuencia de cada entrada dobla a la anterior, se cubren todas las posibilidades de entrada. El barrido transitorio se debe ajustar para observar al menos un período de la señal con menor frecuencia. Esto permite obtener la tabla de verdad de circuitos con múltiples entradas y múltiples salidas a través de Probe.

□ Tabla de la verdad

La tabla de verdad de un circuito lógico no se puede obtener directamente, sin embargo, existe una manera indirecta y práctica de obtener la tabla de verdad del circuito, usando el concepto de **bus**. Con Probe, puede tomarse una cantidad de señales digitales y considerarlas como “bits” independientes cada una, ya que en cada momento estas señales representan un cero o uno lógico. Todos estos “bits” pueden agruparse juntos de manera ordenada y formar una “palabra” o código, constituyendo lo que en Probe se denomina **bus**. Así, el bus no es otra cosa que un conjunto de señales

digitales ordenadas como una palabra binaria natural, de modo que al hacer la gráfica del bus en Probe se puede mostrar el valor de la palabra en binario, decimal, hexadecimal u octal. Al definir un bus, debe teclearse en el cuadro de diálogo del menú **Trace**, **Add** la expresión:

$$\{ S_N S_{N-1} \dots S_1 S_0 \} ; \text{nombre} ; x$$

Donde S_N, S_{N-1}, \dots, S_0 son las señales digitales de Probe ordenadas en forma natural, de modo que S_N constituye el bit más significativo, mientras que S_0 viene a ser el bit menos significativo. El identificador “**nombre**” es un literal con el cual se nombra al bus en el gráfico de Probe, mientras que x es un carácter que identifica la base en la que se expresará el valor del bus: **b** para binario, **d** para decimal, **h** para hexadecimal y **o** para octal.

EJEMPLO 3-2

Si las entradas de un sistema son D, C, B y A, siendo A la entrada menos significativa y D la más significativa, una notación de bus para las entradas sería:

$$\{ D C B A \} ; \text{ENT} ; b$$

Esta expresión hará que Probe muestre las entradas como números binarios de 4 bits en una señal llamada ENT.

Bibliotecas

Para que funcionen correctamente las simulaciones en otras computadoras, se deben instalar las librerías utilizadas, copiando los archivos **.lib** en el directorio de trabajo, y el Schematics del Pspice se debe configurar siguiendo los siguientes pasos:

- En el menú Options, seleccionar Editor Configuración.
- Pulsar el botón Libraries Settings, Browse.
- Seleccionar la librería en el directorio correspondiente.
- Presionar el botón Add* o Add y pulsar OK.

Es importante también configurar algunos parámetros para el análisis de circuitos digitales, esto se hace seleccionado Analysis, luego set up y finalmente digital set up. En esta opción se puede predeterminar los retardos de todos los componentes y el valor inicial de los dispositivos secuenciales.

EJEMPLO 3-3

Utilice Solamente NAND para realizar un circuito que cumpla con el siguiente enunciado:

La salida del circuito será ALTA, siempre que sus entradas A y B o sus entradas C y D sean ALTAS.

¿Cuál es la representación más adecuada del circuito?

Solución:

Para realizar el circuito que se pide, se necesita conocer la expresión booleana que lo caracteriza. Al leer el enunciado se desprende que la expresión lógica es:

$$Z = AB + CD.$$

No obstante, sería conveniente demostrar que esta expresión es la más reducida. Para hacerlo, se debe construir la tabla de la verdad y de allí sintetizar la mínima expresión booleana en forma de suma de productos.

Como la función está en la forma de suma de productos, la implementación en compuertas lógicas es del tipo AND-OR, como se muestra en la figura 3-8.

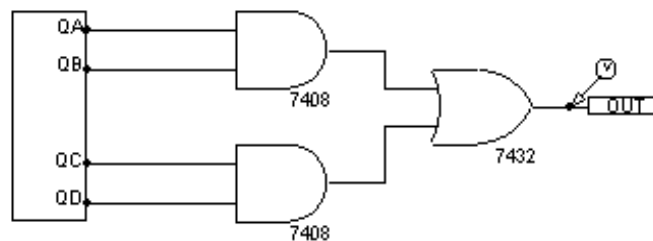


Figura 3-8 Esquemático en Pspice de la función $Z = AB + CD$

En la simulación se utilizó un componente creado con fuentes de señales DigStim de frecuencia 100kHz, 50kHz, 25kHz, 12.5kHz, para las señales QA, QB, QC, QD, respectivamente (figura 3-9).

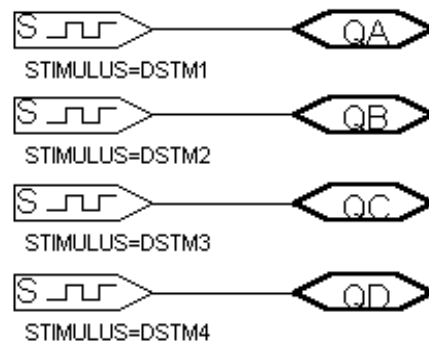


Figura 3-9 Señales de entrada con DigStim de Pspice

El diagrama de tiempos (figuras 3-10 y 3-11) muestra las señales de 4 bits que alimentan el circuito, llamadas QA, QB, QC, QD, siendo QA el bit menos significativo y QD el más significativo. Estas señales, están sincronizadas para que en su conjunto conformen una señal de 4 bits con valor ascendente, es decir, que puedan generar todas las posibles 16 combinaciones de una señal de 4 bits. El valor se puede observar en formato binario.

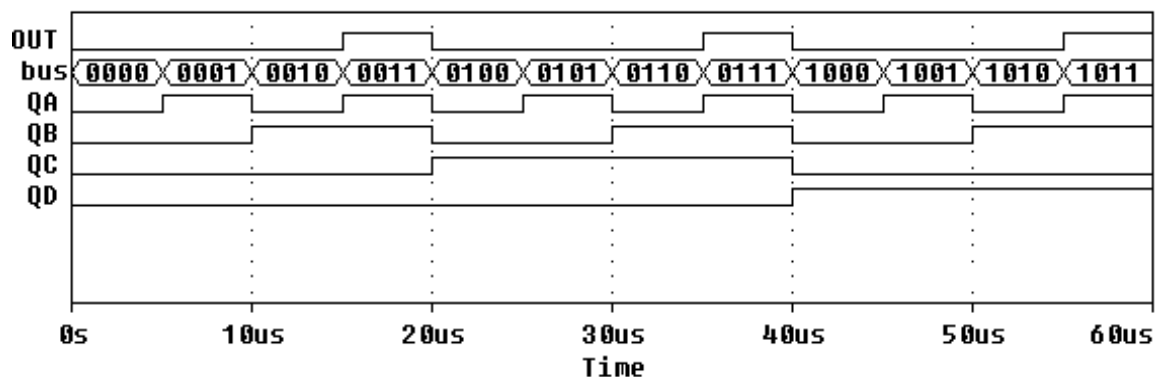


Figura 3-10 Diagrama de tiempos utilizando Probe

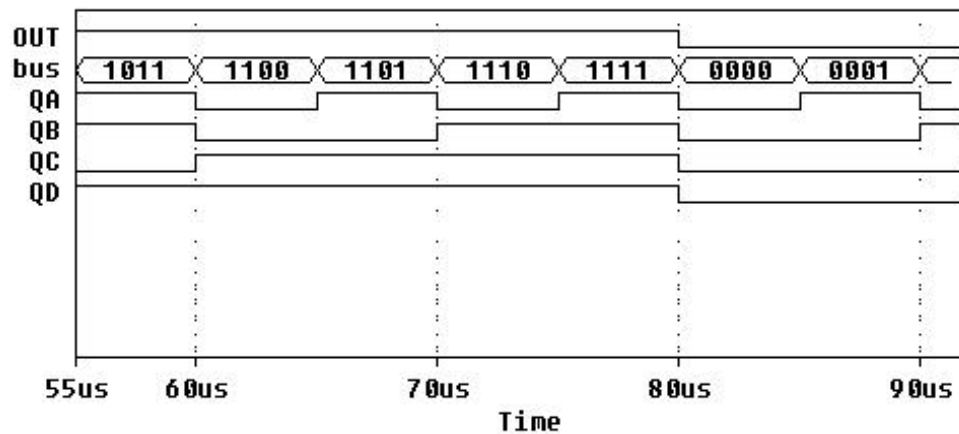


Figura 3-11 Diagrama de tiempos utilizando Probe

Para obtener el circuito NAND se sustituye cada compuerta por el equivalente y se simplifican los inversores en serie, obteniéndose el circuito mostrado en la figura 3-12. Los diagramas de tiempos, tanto del circuito AND-OR como del circuito con compuertas NAND, obtenidos por simulación se muestran en la figura 3-10 y 3-11.

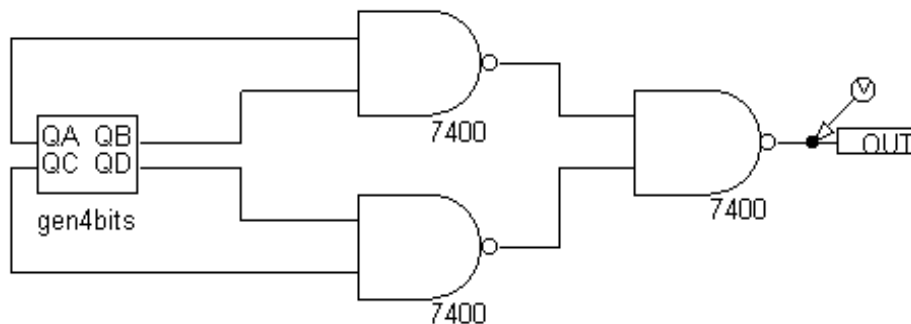


Figura 3-12 Esquemático de Z con NAND

La representación más adecuada del circuito debe poseer la salida activa en alto según el enunciado, tomando en cuenta esto se consigue el circuito con solo cambiar la compuerta NAND de salida por su símbolo alternativo (circuito de la figura 3-13).

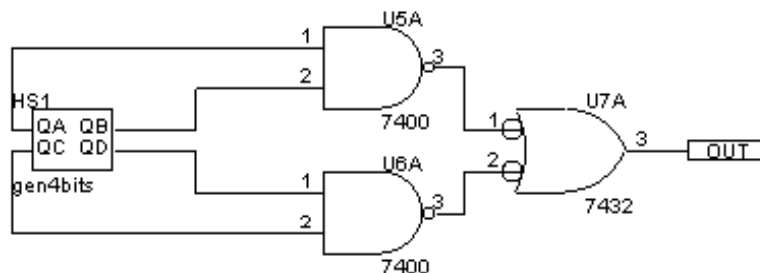


Figura 3-13 Representación adecuada de Z

3.2.2 ELECTRONIC WORK BENCH

Existe también otro software comercial especializado en la simulación de circuitos, se trata del **Electronic Work Bench**, que en su versión 5.0 ofrece herramientas que hacen muy real el trabajo con circuitos digitales.

Lo más importante de este tipo de aplicación resulta de la disponibilidad de componentes para realizar mediciones y generar funciones. Estos componentes se asemejan, en forma y funcionamiento, a los equipos utilizados en laboratorios. [4]

Construcción del circuito

Al entrar en el EWB, puede dibujarse el circuito escogiendo las compuertas necesarias a través del conocido sistema *drag&drop* (arrastrar y soltar). Al hacer doble click sobre cualquiera de las compuertas, puede escogerse el modelo de las mismas: TTL, CMOS o ideal. Si se escoge TTL o CMOS, se dispone de una amplia lista de modelos comerciales.

Generador de palabras

Para generar las señales del circuito, puede utilizarse el **Word Generator** (Generador de palabras) que provee hasta 16 salidas binarias. Se puede ajustar en el generador la frecuencia máxima de salida (la del bit menos significativo) y el patrón de la palabra de salida: conteo progresivo, conteo regresivo, corrimiento a la derecha o corrimiento a la izquierda, todo esto a través del botón **Pattern**.

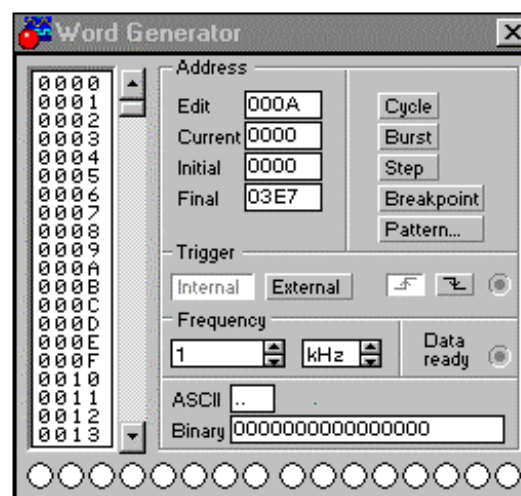


Figura 3-14 Generador de palabras

Si se quiere generar la tabla de verdad del circuito, se ajusta **Pattern** para **Up counter** y la frecuencia al valor máximo deseado. Durante la simulación se realizará un barrido de la salida del generador, de manera indefinida o sólo una vez, esto si la selección es **Cycle** o **Burst**. El barrido puede ajustarse desde un valor inicial a un valor final definidos por el usuario, y expresados en **Initial** y **Final** como palabras hexadecimales de 4 dígitos.

Analizador lógico

Para observar la salida del circuito a través del diagrama de tiempo, puede utilizarse el **Logic Analyzer** (Analizador lógico), conectando la salida del circuito a este dispositivo. En la pantalla se verá (al hacer doble click sobre este componente) el diagrama de tiempo de las señales conectadas, y puede ajustarse la base de tiempo al valor deseado.

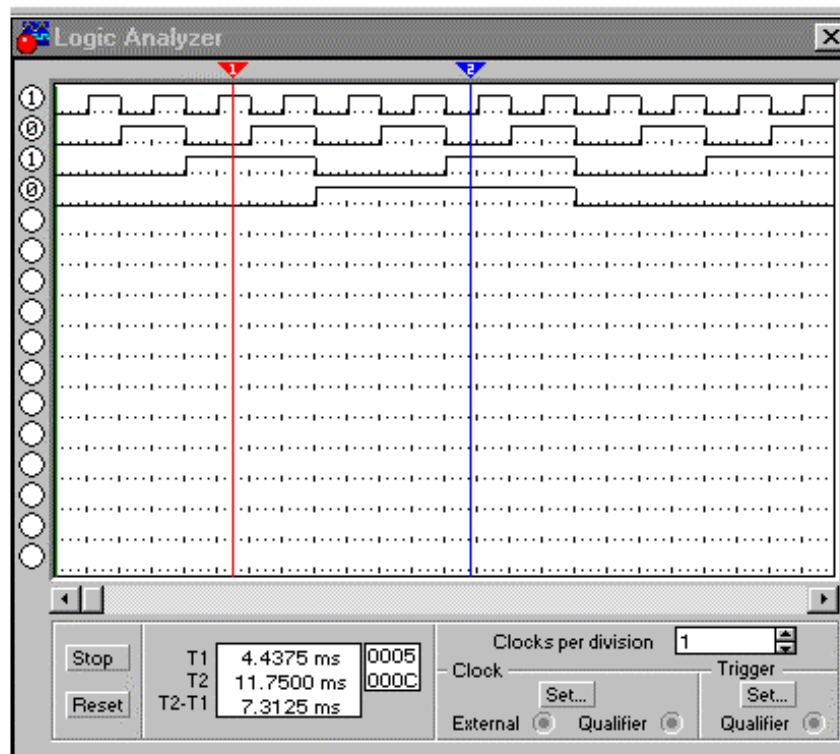


Figura 3-15 Analizador lógico

El convertidor lógico

El EWB posee una herramienta muy útil, el **Logic Converter** (Convertidor lógico) que no tiene similar en el laboratorio. Al conectarse al **Logic Converter** las entradas y la salida de cualquier circuito lógico, este puede elaborar la tabla de verdad del circuito. Al generar las entradas con el Word Generator, el Logic Converter despliega la tabla de verdad del circuito. También posee otras opciones interesantes, como generar la expresión lógica completa o simplificada a partir de la tabla de verdad, generar el diagrama de circuito de la expresión en compuertas AND-OR-NOT o en compuertas NAND. Todos estos procesos de conversión y su inverso solo se realizan para una función de una salida.

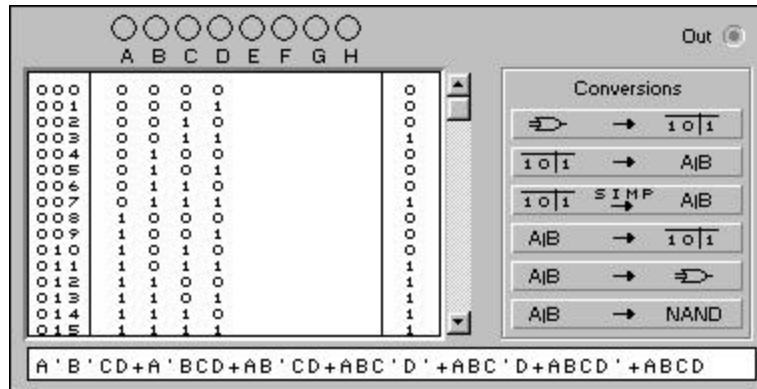


Figura 3-16 Convertidor lógico

Según puede verse, este tipo de simulaciones digitales con compuertas lógicas son más sencillas en EWB, y las herramientas son más amigables. El EWB cuenta con una librería digital más completa que el PSpice. Sin embargo, el núcleo del EWB es una versión modificada del Pspice y por tanto no es un estándar. El Pspice permite la manipulación directa de los parámetros de simulación y hacer gráficos con Probe.

3.3 CIRCUITOS INTEGRADOS

Los circuitos lógicos están disponibles como circuitos integrados (C.I. en forma abreviada), los cuales poseen ventajas en cuanto a costo, tamaño, consumo de potencia y confiabilidad, a la vez que permiten la estandarización. Grandes cantidades de compuertas lógicas pueden ser colocadas en un solo chip o pastilla. Las técnicas de

fabricación de los C.I. es comparable a la de los dispositivos discretos o transistores y su estudio está fuera del alcance del presente texto.

Los circuitos integrados son clasificados de acuerdo a su complejidad. Los integrados denominados SSI o de integración de pequeña escala son aquellos que contienen menos de 20 compuertas y pueden encontrarse en empaques de 14 terminales denominado DIP siglas de Dual Inline Package.

Los circuitos MSI o de integración de media escala contienen menos de 200 compuertas y normalmente contienen un circuito funcional completo. Los circuitos de gran escala de integración LSI contienen el equivalente de 200 a 20000 compuertas o más, estos incluyen memorias, microprocesadores, lógica programable y circuitos de aplicación específica (ASICs).

Los circuitos de integración a muy gran escala o VLSI, son aquellos que poseen más de 400.000 transistores, entre estos tenemos grandes memorias (1Megabit) y microprocesadores.

Todos estos C.I. pueden estar en empaques o encapsulados de diversos tamaños, de plástico o de cerámica (figura 3-17).

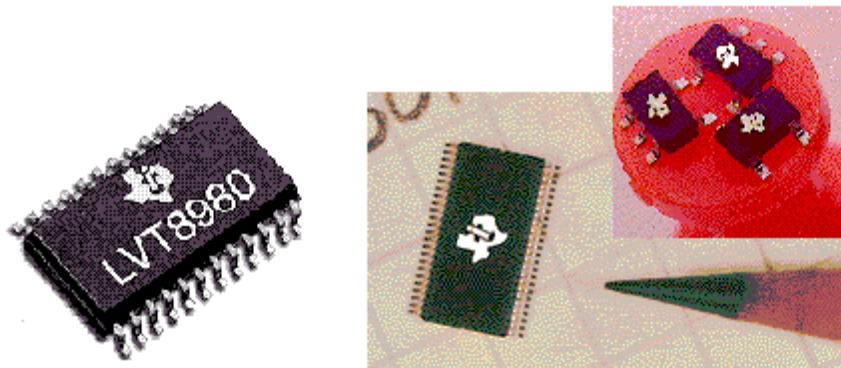


Figura 3-17 Circuitos Integrados *Fuente:* Texas Instruments Inc. Logic Selection Guide and Databook. 2/E, Texas Instruments, Dallas, 1997

3.3.1 FAMILIAS LÓGICAS

Los circuitos integrados se clasifican también por la familia de circuitos lógicos a las cuales pertenecen. Cada familia lógica tiene su propio circuito electrónico básico con el cual se desarrollan circuitos y funciones más complejas. En la tabla 3-1 se muestran algunas de las familias que han alcanzado un amplio uso.

Los C.I. bipolares son aquellos que están fabricados con transistores bipolares de unión (NPN y PNP) como su principal componente de circuito. Los C.I. unipolares son los que emplean transistores unipolares de efecto de campo (MOSFET de canal P o N) como elemento principal.

FAMILIA	CARACTERÍSTICA	COMENTARIO
TTL	Lógica transistor-transistor	Una de las primeras familias.
ECL	Lógica de emisor acoplado	Basada en circuitos de alta velocidad.
MOS	Semiconductor de óxido metálico	Permite circuitos que requieren alta densidad de componentes.
CMOS	Semiconductor complementario de óxido metálico	Posee bajo consumo de potencia y amplio rango de alimentación
I^2L	Lógica de inyección integrada	Circuitos de muy alta densidad de componentes.

Tabla 3-1 Familias lógicas

En la figura 3-18 se muestra la tendencia de las tecnologías o familias de C.I. según el fabricante de componentes TEXAS INSTRUMENTS.

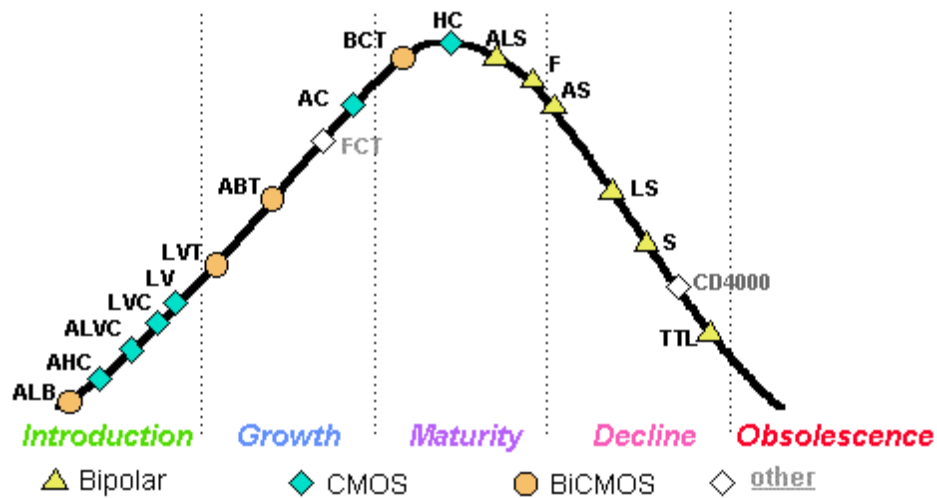


Figura 3-18 Tendencias en la utilización de circuitos integrados *Fuente: Idem*

3.3.2 NIVELES LÓGICOS

Los C.I. digitales tienen dos regiones de voltaje para las entradas y salidas de sus circuitos las cuales representan los valores binarios o estados 1 y 0 del álgebra de conmutación.

La figura 3-19 muestra una típica característica voltaje de salida contra voltaje de entrada para una compuerta bipolar NOT. En esta ilustración se resaltan las regiones de operación para los estados 1 y 0. Dentro de estas regiones los voltajes son reconocidos por los dispositivos digitales como estados 1 ó 0 ; estos son los rangos permitidos puesto que los diferentes fabricantes no garantizan que para otros valores de voltaje los dispositivos respondan adecuadamente cuando se interconectan. [6]

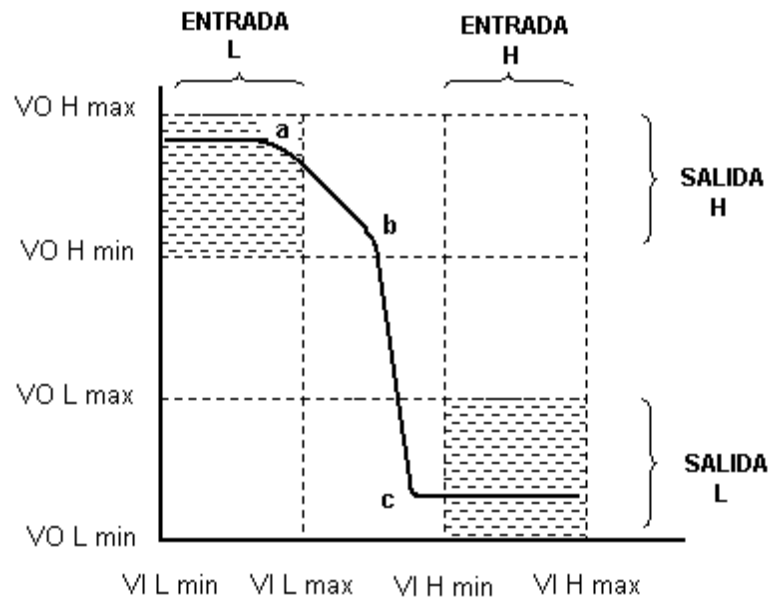


Figura 3-19 Característica Salida vs Entrada de una compuerta bipolar *Fuente: Idem*

Como se indica en la figura 3-20, para una entrada de un dispositivo, un estado 0 corresponde a un voltaje denominado voltaje de entrada del nivel bajo, V_{IL} , entre los valores $V_{IL\ min}$ y $V_{IL\ max}$. Por otra parte, la entrada de un estado 1 corresponde a un voltaje denominado voltaje de entrada del nivel alto V_{IH} , en el intervalo de $V_{IH\ min}$ hasta $V_{IH\ max}$.

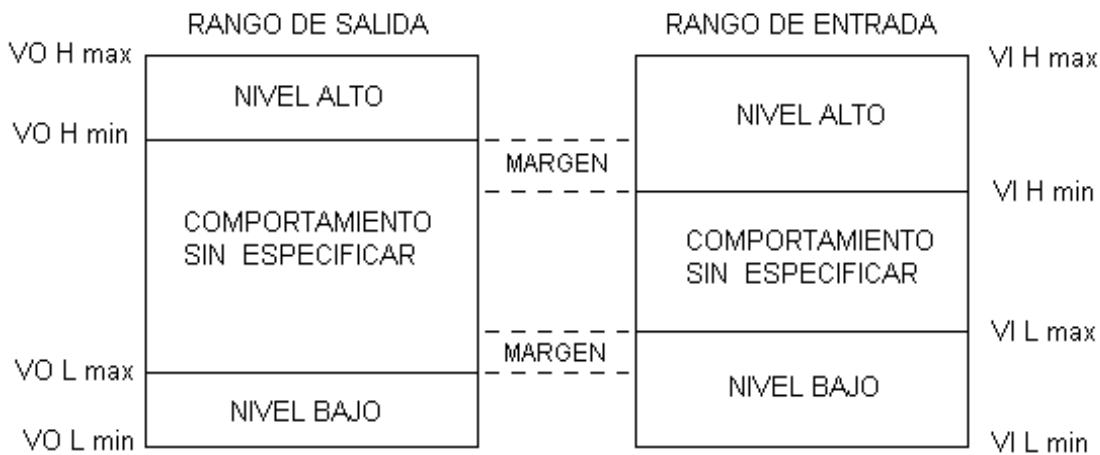


Figura 3-20 Niveles lógicos

De manera similar, una salida en estado 0 consiste en un voltaje V_{OL} entre $V_{OL\ min}$ y $V_{OL\ max}$, y una salida en estado 1 tendrá un valor V_{OH} entre $V_{OH\ min}$ y $V_{OH\ max}$.

Existen diversos procesos y materiales (tecnologías) usados en la fabricación de compuertas en circuitos integrados, de modo que el usuario dispone de diferentes tipos de circuitos lógicos agrupados en las llamadas familias de circuitos digitales. Los niveles lógicos y las características eléctricas en general varían de acuerdo a la tecnología y diseño del circuito integrado.

3.3.3 SERIES DE LA FAMILIA LÓGICA TTL

La familia más utilizada de C.I. digitales bipolares es la familia TTL (Lógica transistor-transistor) que fue introducida comercialmente en 1964 y está constituida por varias sub-familias o series. Actualmente la familia TTL está desapareciendo por la aparición de mejores C.I. realizados en otras tecnologías, sin embargo, por el amplio uso que se le dio en los años 60 y 70 todavía continua su producción.

Las diferentes series TTL se diferencian principalmente en la disipación de potencia, tiempos de retraso y velocidad de conmutación. Los empaques estándar de los circuitos son del tipo DIP y no difieren en la distribución de terminales o en las operaciones lógicas efectuadas por sus circuitos. En la figura 3-21 se ilustra el diagrama de un circuito integrado SSI el cual contiene cuatro compuertas NAND, el tipo de empaque y las características eléctricas son dadas por el fabricante en una hoja de datos.

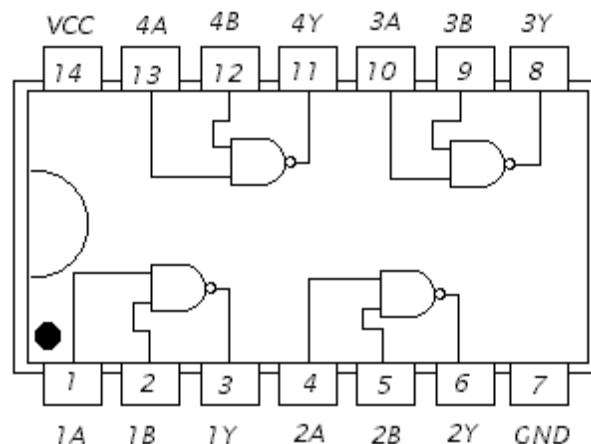


Figura 3-21 Diagrama de un C. Integrado SSI

Los identificadores de los circuitos integrados TTL contienen un prefijo utilizado para especificar a cual serie pertenece. Tal como se indica en la tabla 3-2, los C.I. que forman parte de la serie estándar TTL tienen un número de identificación que comienza con 74 (7400, 7408 etc.); mientras que los de igual función lógica que pertenecen a otra serie como la TTL Schottky (baja potencia) tienen un número de identificación que comienza con 74LS (74LS00, 74LS08 etc.).

SERIE	PREFIJO	COMPUERTA NOT
TTL estándar	74	7404
TTL de alta velocidad	74H	74H04
TTL de baja potencia	74L	74L04
TTL Schottky	74S	74S04
TTL Schottky de baja potencia	74LS	74LS04
TTL Schottky avanzada	74AS	74AS04
TTL Schottky de baja potencia	74ALS	74ALS04

Tabla 3-2 Identificación de la familia TTL

3.3.4 CARACTERÍSTICAS DE LAS SERIES TTL

Los fabricantes de C.I. proporcionan tablas con las características eléctricas. En la tabla 3-3^a y tabla 33b se muestra parte de las especificaciones para la serie 74 estándar, tal como la presenta un fabricante. Los valores mínimo y máximo que se muestran, son para las peores circunstancias de fuente de alimentación, temperatura y condiciones de carga.

	SN7400			UNIT
	MIN	MAX	NOM	
V _{cc} supply voltage	4.75	5.25	5.0	V
V _{IH} high-level input voltage	2.0			V
V _{IL} low-level input voltage		0.8		V
I _{OH} high-level output current		-0.4		mA
I _{OL} low-level output current		16		mA
T _A operating free-air temperature	0	70		°C

Tabla 3-3a Características eléctricas de la serie TTL estándar *Fuente: Texas Instruments, Inc. Logic Selection Guide and Databook. 2/E, Texas Instruments, Dallas,*

PARAMETER	TEST CONDITIONS	SN7400			UNIT
		MIN	TYP®	MAX	
VOH	V _{cc} =Min, IOH=-0.4mA	2.4		3.4	V
VOL	V _{cc} =Min, VIH=2V, IOL=16mA		0.2	0.4	V
I _{ih}	V _{cc} =Max, VI=2.4V			40	μA
I _{il}	V _{cc} =Max, VI=0.4V			-1.6	mA
I _{os}	V _{cc} =Max (one output)		-18	-55	mA
I _{cch}	V _{cc} =Max, VI=0V		4.0	8.0	mA
I _{ccl}	V _{cc} =Max, VI=4.5V		12	22	mA

Tabla 3-3b Características eléctricas de la serie TTL estándar Fuente: Idem

Las tablas 3-4 y 3-5 corresponden a las principales características de algunas series de circuitos integrados.

NÚMERO DE C.I. COMPUERTA CARACTERÍSTICA	74 L 02 NOR			74 LS 00 NAND			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V _{cc}	4.75	5.0	5.25	4.75	5.0	5.25	V
VIH	2.0			2.0			V
VIL			0.7			0.8	V
VOH (IOH max)	2.4	3.2		2.7	3.4		V
VOL (IOL max)		0.2	0.4		0.25	0.5	V
IOH			-0.2			-0.4	mA
IOL			3.6			8	mA

Tabla 3-4 Características de la serie TTL 74 L y 74 LS Fuente: Ibid

NÚMERO DE PARTE: COMPUERTA: CARACTERÍSTICA	74 32 OR			74 H 08 AND			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V _{cc}	4.75	5.0	5.25	4.75	5.0	5.25	V
VIH	2.0			2.0			V
VIL			0.8			0.8	V
VOH (IOH max)	2.4	3.4		2.7	3.4		V
VOL (IOL max)		0.2	0.4		0.15	0.3	V
IOH			-0.8			-0.5	mA
IOL			16			20	mA

Tabla 3-5 Características de la serie TTL 74 y 74 H Fuente: Ibid

Niveles lógicos TTL

La mayoría de los C.I. de la familia TTL funciona con una sola fuente de alimentación (VCC= + 5 v) y las tolerancias varían de acuerdo al fabricante (5% típico).

Los rangos de voltajes aceptables de la serie TTL estandar para los niveles lógicos alto (1) y bajo (0) son los que se indican en la figura 3-22.

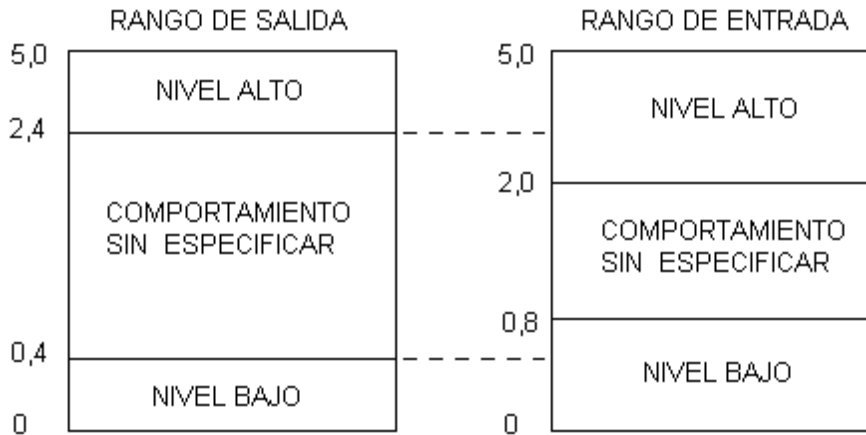


Figura 3-22 Niveles lógicos de la serie TTL

Corrientes de salida y factor de carga (fan – out)

El factor de carga (fan-out) de una compuerta se define como el número máximo de entradas lógicas estándar que una salida puede manejar confiablemente.

A medida que aumenta la carga o corriente, en la salida de una compuerta, el voltaje de la salida para el nivel alto (bajo) disminuye (aumenta). Si el fan-out es excedido la salida puede estar fuera del intervalo de los niveles lógicos, es decir, en este caso no se garantiza que los voltajes del nivel lógico de salida sean los correctos.

Para la serie TTL estándar el fabricante garantiza que la corriente de entrada necesaria para un nivel bajo es $I_{IL} < 1,6 \text{ mA}$ (tabla 3-2b), mientras que para un nivel alto se necesita $I_{IH} < 40 \text{ uA}$. El fabricante a diseñado la etapa de salida para una corriente de 16 mA en el rango de voltajes del nivel bajo, esto quiere decir que el $V_{OL \text{ max}}$ corresponde a una carga equivalente a 16 mA, más allá de este valor de corriente el voltaje de la salida aumenta y estará fuera del rango del nivel bajo.

Al observar las corrientes de entrada y salida para el nivel bajo se deduce que una salida puede manejar 10 entradas. Un análisis similar para el nivel alto da que la máxima carga es de 100 entradas. Por tanto el Factor de carga de la compuerta está garantizado para 10 entradas.

El Fan-out o factor de carga de los circuitos lógicos es por tanto:

$$FC(\text{bajo}) = I_{OL}(\text{max}) / I_{IL}(\text{max})$$

$$FC(\text{alto}) = I_{OH}(\text{max}) / I_{IH}(\text{max})$$

Si estos factores de carga no son iguales se debe escoger el menor valor .

Margen de ruido DC

El nivel lógico bajo (0) a la salida de la compuerta se garantiza para un máximo de 400mv el cual es 400mv menor que el voltaje de entrada para el 0 lógico (800mv). Para la salida en nivel lógico alto (1) el voltaje mínimo garantizado es 2,4v lo cual es 400mv mayor que el correspondiente mínimo valor para la entrada (2,0v). Este margen de seguridad para el 0 y 1 lógico se conoce como margen garantizado de ruido de corriente directa. En la figura 3-23 y figura 3-24 se muestra a modo de ilustración el margen de ruido de otras familias lógicas. [5]

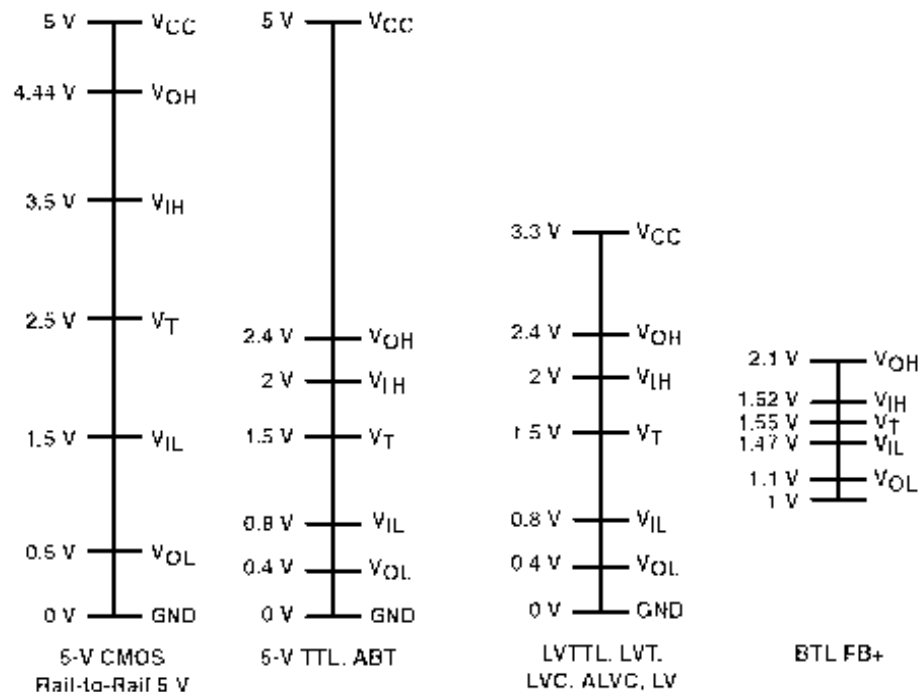


Figura 3-23 Niveles lógicos vs tecnologías

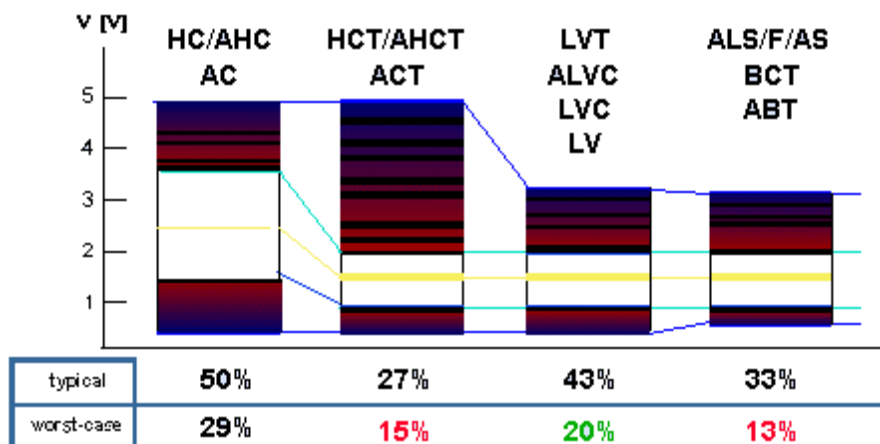


Figura 3-24 Márgenes de ruido versus tecnologías *Fuente: Idem*

Compuerta básica de la serie estándar TTL

La figura 3-25 muestra el circuito de la compuerta NAND básica de la familia TTL estándar, todas las funciones lógicas de la familia se basan en esta compuerta.

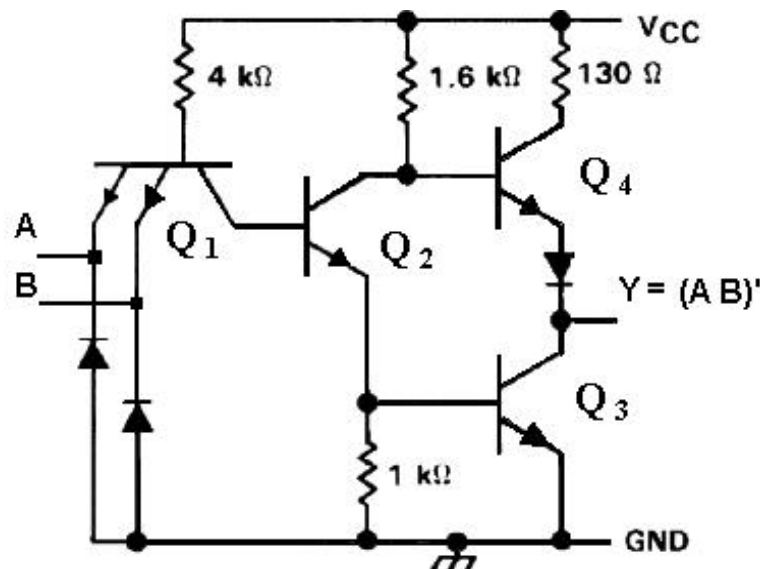


Figura 3-25 Compuerta NAND TTL

Para comprender el funcionamiento del circuito es conveniente referirse también a la función de transferencia de la figura 3-19, cuando el voltaje de entrada V_I de uno o más de los emisores de entrada es menor que el voltaje en el punto "a" (V_a), los transistores Q_2 y Q_3 entran en corte, colocando en conducción a Q_4 y logrando un alto en la salida cuyo valor aproximado es:

$$V_{OH} = V_{CC} - V_{BE} - V_F - V_{130}$$

Donde V_F es el voltaje en el diodo.

Con el voltaje de entrada menor que V_a , la corriente fluye hacia fuera de los emisores de entrada de Q1 debido al voltaje de alimentación V_{CC} y a través de la resistencia de $4K\Omega$. El transistor Q4 actúa como seguidor de emisor colocando un nivel alto en la salida cuyo voltaje depende de la carga que se conecte.

Cuando el voltaje de entrada V_I en todas sus entradas es mayor que el valor en el punto "c", los transistores Q2 y Q3 entran en conducción; Q4 pasa a corte logrando un nivel bajo (0) a la salida. El nivel lógico bajo para una carga específica, está determinado por la resistencia de saturación de Q3 y la corriente que este debe drenar a tierra.

Se puede entender la región de transición a,b,c , suponiendo que todas las entradas se unen físicamente y analizando el comportamiento del circuito ante el voltaje de entrada. Al aumentar el valor del voltaje de entrada desde cero, la corriente de base de Q1 se desviará gradualmente del emisor causando que Q2 entre en conducción, lo cual ocurre para $V_I = V_a = -0.7v$. El transistor Q2 pasa a operar en su región lineal y debido a que Q4 permanece conduciendo, la salida sigue las características de ganancia de Q2 que está determinada por la relación entre sus resistencias de colector y emisor la cual es de 1,6. En el punto "b" la entrada es suficientemente alta para causar que Q3 entre en conducción y reduzca la impedancia del emisor de Q2 , lográndose con esto un aumento de la ganancia y por tanto una pendiente más pronunciada entre los puntos b y c .

En el punto c , $V_C = 1,3v$ aprox., Q4 deja de conducir y la salida toma un valor lógico 0 que depende de la carga y las características de Q3 en saturación. [6]

Rangos de voltaje de alimentación y temperatura

La serie 54 y 74 utilizan una fuente de alimentación con un voltaje nominal de 5V. La serie 74 se especifica para el rango de 4.75 a 5.25V, mientras que la serie 54 permite un voltaje de alimentación desde 4.5 hasta 5.5V. La serie 74 está diseñada para temperaturas ambientales que van de 0°C a 70°C.

Requerimientos de potencia

La cantidad de potencia que necesita un C.I. se especifica por lo general en términos de la corriente I_{cc} que consumen las fuentes de alimentación. En el caso de la familia TTL la potencia real es el producto $I_{cc} \times V_{cc}$. Cuando todas las salidas de las compuertas son altas, el consumo de la corriente de alimentación V_{cc} recibe el nombre de I_{cch} , y cuando todas las salidas de las compuertas sean bajas reciben el nombre de I_{ccl} .

Disipación de potencia

En un C.I. TTL con 4 compuertas básicas el consumo de corriente promedio es $I_{ccprom} = 8\text{mA}$ ($I_{cch} = 4\text{mA}$ y $I_{ccl} = 12\text{mA}$), esto implica una potencia de $P_{dprom} = 8\text{mA} \times 5\text{V} = 40\text{mW}$. Por tanto una compuerta NAND TTL estándar disipa una potencia promedio de 10mW .

Por otra parte, la potencia disipada aumenta con la frecuencia de trabajo, esto es debido a los picos de corriente que se producen en las salidas (y por tanto en la alimentación) durante el tiempo en que ambos transistores de salida de la compuerta cambian de estado, esto se ilustra en la figura 3-26 donde se presentan la corriente de la alimentación de dos compuertas de diferente tecnología. Es conveniente comentar que estas variaciones pueden afectar otros circuitos y por ello se deben colocar filtros en la alimentación de los C. Integrados.

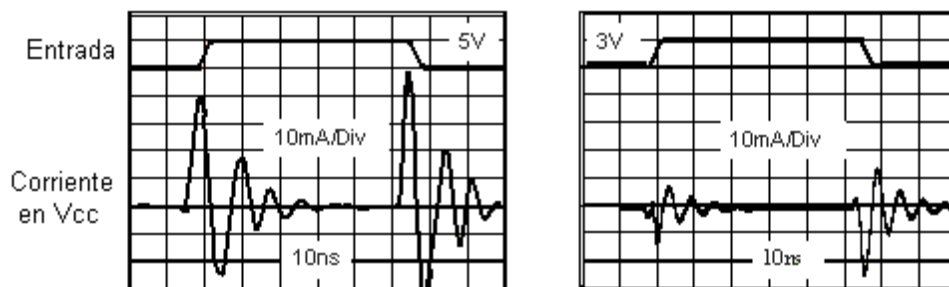


Figura 3-26 Perturbaciones en la alimentación de un C. I. Fuente: Texas Instruments Inc. Logic Selection Guide and Databook. 2/E, Texas Instruments, Dallas, 1997

Retardos en compuertas

La trayectoria o recorrido de una señal es el camino eléctrico desde una entrada particular a una salida particular en un elemento o compuerta lógica. El retraso

por propagación **tpd** en esa trayectoria es la cantidad de tiempo necesaria para que un cambio en la señal de entrada produzca un cambio en la señal de salida.

En la figura 3-27 se muestra como un cambio en el nivel de entrada de una compuerta NOT produce el nivel lógico un tiempo **tpd** después que el nivel a la entrada cambia.

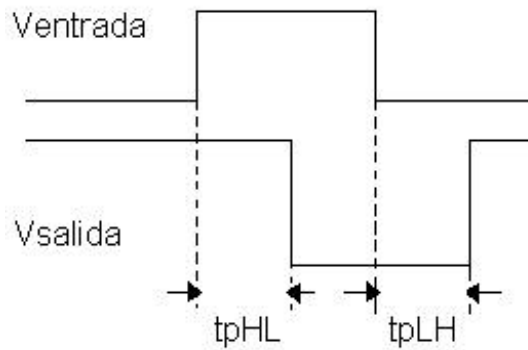


Figura 3-27 Retardos en compuertas lógicas

Se observa que el tiempo de propagación cuando la salida cambia de BAJO a ALTO (t_{PLH}) puede ser diferente del retraso cuando esta cambia de ALTO a BAJO (t_{PHL}). En la figura 3-28 se presentan los valores de retraso típicos para una compuerta TTL tal como los presenta el fabricante. En términos generales t_{PLH} y t_{PHL} varían según las condiciones de carga.

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	RL=400Ω CL=15PF		11	22	NS
t_{PHL}			7	15	NS

Tabla 3-6 Tiempos de transición de compuertas TTL

Diversos factores conducen a tiempos de propagación diferentes de cero aún cuando los tiempos de ascenso y descenso para entrada y salida tiendan a cero como en la figura 3-27 o diferentes de cero como en la figura 3-29.

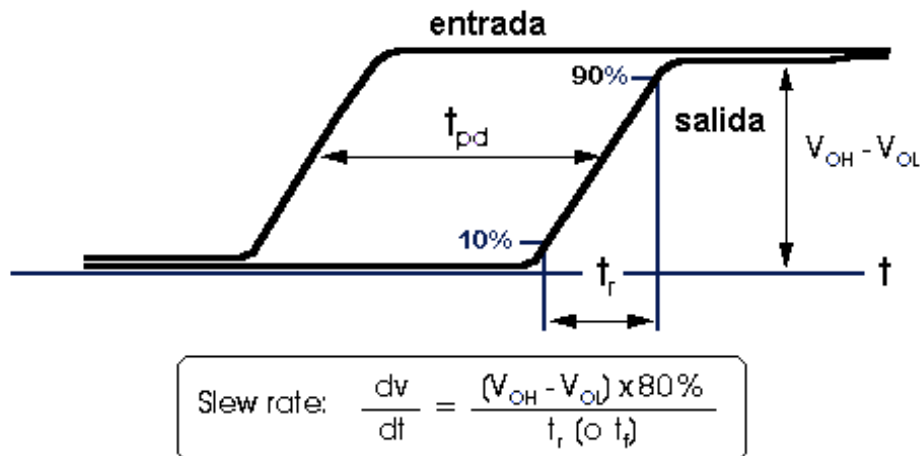


Figura 3-28 Transición y tiempos de ascenso en compuertas

Por ejemplo, en lógica TTL cuando cambia la entrada a un transistor saturado, la salida requiere que varios transistores internos cambien de estado antes que la salida pueda hacerlo.

Un circuito combinacional, dependiendo de la cantidad de niveles (AND-OR, NAND, etc) y de las entradas y salidas tendrá varias trayectorias o rutas y cada una puede tener un tiempo de propagación diferente. Las especificaciones de tiempo para un dispositivo pueden dar un valor mínimo, típico y máximo del tiempo de propagación para cada ruta y dirección de transición.

El peor de los casos de retraso a través de un circuito se calcula como la suma de los peores casos de retraso a través de los componentes individuales, de modo independiente de la dirección de transición y de otras condiciones del circuito. Esto puede dar una visión pesimista del retraso del circuito pero ahorra tiempo en el diseño y garantiza el funcionamiento.

Tiempos de ascenso y descenso

La salida de un dispositivo digital no cambia instantáneamente, sino como se representa en la figura 3-29, cuando la salida cambia de estado lo hace con un tiempo de transición, es decir la salida presenta un frente de onda o flanco tipo rampa.

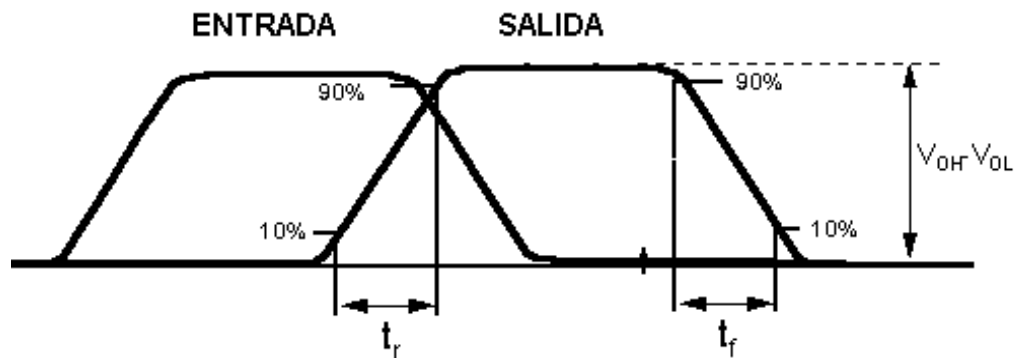


Figura 3-29 Tiempos descenso y descenso en compuertas

Se define como tiempo de ascenso o crecimiento t_r (rise) el tiempo en que la salida tarda en ir desde el 10% al 90% de su excursión máxima ($V_{oh\ MAX} - V_{ol\ MIN}$). De manera similar, se define el tiempo de descenso o caída t_f (fall), como el tiempo en que la salida tarda en ir desde el 90% al 10% de su excursión máxima.

Una señal de entrada lenta (slow) es aquella que posee una pendiente o rata de cambio del flanco de bajo valor en relación con la velocidad o retardos de el dispositivo. Este tipo de entradas puede producir oscilaciones (figura 3-30) tanto en la entrada como en la salida del componente digital debido a las inductancias internas del circuito integrado.

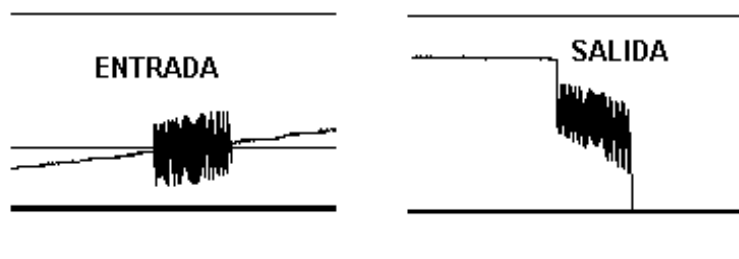


Figura 3-30 Oscilaciones en entradas y salidas de compuertas Fuente: Idem

Producto velocidad potencia

Las familias digitales de C.I. generalmente se comparan tanto en su potencia como por su velocidad, idealmente es deseable tener menores retardos en la propagación y bajos valores de disipación de potencia. La forma de medir y comparar el des-

empeño de una familia de C.I. es el producto velocidad potencia, que se obtiene al multiplicar el retardo de propagación de la compuerta por la potencia que disipa.

Margen de ruido de AC

Tal como se indico anteriormente, una señal lógica siempre experimenta un retardo al recorrer un circuito. Considerando una señal de entrada tipo pulso, conforme las anchuras de los pulsos de señal de entrada decrecen a la región inferior de los nanosegundos, se llega a un límite donde la duración del pulso, es demasiado corta para que el circuito responda. En este punto la duración del pulso tendría que ser incrementada apreciablemente a fin de producir una variación en la salida del circuito.

Por tanto, un circuito lógico puede tolerar una amplitud mayor de ruido, si el pulso de ruido es de una duración muy corta en comparación con el tiempo de respuesta del circuito. Los márgenes de ruido de AC del circuito son por consiguiente mayores que los márgenes de ruido DC.

3.4 DIAGRAMAS DE KARNAUGH

El método del DIAGRAMA DE KARNAUGH permite realizar de manera fácil y rápida la reducción de expresiones de seis o menos variables. Este método lo introdujo inicialmente E. Veitch, con los diagramas de Veitch que luego fueron modificados y presentados como MAPAS (M.K.) por M. Karnaugh.

3.4.1 APLICACIÓN DEL TEOREMA DE ADYACENCIA

Los procedimientos de simplificación de expresiones booleanas, a través de la aplicación de axiomas y teoremas algebraicos, son tediosos y algunas veces intuitivos.

Un método consiste en expandir la expresión hasta la forma canónica (ESTANDAR) y agrupar sistemáticamente utilizando la idempotencia y el teorema de adyacencia para simplificar y obtener la mínima cantidad de términos.

Es conveniente expresar el teorema de adyacencia para términos productos estándar como:

Para dos variables X e Y : $XY + XY' = X$

Para tres variables A,B,C : $AB' C' + AB' C = AB'$

Es decir, la suma de dos términos productos estándar que difieren en la condición de una variable es equivalente al término producto sin esta variable.

En forma más general, la suma de dos minitérminos que difieren en una sola variable es equivalente al término producto consistente de las variables en que no difieren los dos minitérminos.

Es importante notar que los números de dos minitérminos que difieran en una única variable, tal como lo establece el teorema de adyacencia, serán números binarios que difieren en un solo bit.

Veamos esto más detenidamente con un ejemplo:

Sea la función

$$F(A,B,C) = \sum m_i = \sum m(0,1,2,3,5)$$

Por definición de minitérminos tenemos la expresión:

$$F(A,B,C) = m_0 + m_1 + m_2 + m_3 + m_5$$

$$F(a,b,c) = a'b'c' + a'b'c + a'bc' + a'bc + ab'c$$

La función está en términos estándar por lo que no es necesario la expansión de los términos.

Por definición del teorema de adyacencia, este puede ser aplicado entre los minitérminos (0,1) y entre (1,3), (2,3) y (1,5).

Utilizamos la idempotencia para generar otro minitérmino 1 y poder formar los grupos de minitérminos (0,1), (2,3) y (1,5):

$$F(a,b,c) = (a'b'c' + a'b'c) + (a'bc' + a'bc) + (ab'c + a'b'c) \quad \text{T. Idempotencia.}$$

Aplicando T Adyacencia a los grupos de minitérminos (0,1), (2,3) y (1,5) tenemos:

$$F(a,b,c) = (a'b') + (a'b) + (b'c)$$

Observe que no es necesario el grupo de minitérminos (1,3) puesto que los grupos escogidos (0,1), (2,3) y (1,5) incluyen estos minitérminos (teorema de inclusión). De los grupos (0,1) y (2,3) se obtuvieron dos términos (a'b') y (a'b) entre los cuales también se puede aplicar el T. de Adyacencia y obtener así:

$$F(a,b,c) = a' + (b'c)$$

En general, el proceso de reducción consiste en:

Primero, utilizar la idempotencia para generar los términos necesarios y formar todos los grupos de minitérminos que difieran en una variable: (0,1), (1,3), (2,3) y (1,5):

Segundo, escoger la mínima cantidad de grupos que incluyan todos los minitérminos de la función: (0,1), (2,3) y (1,5).

$$F(a,b,c) = (a'b'c' + a'bc') + (a'b'c + a'bc) + (a'b'c + ab'c)$$

Reducirlos por el teorema de adyacencia:

$$F(a,b,c) = (a'c') + (a'c) + (b'c)$$

Tercero, si es posible, aplicar el teorema de adyacencia al resultado. En este caso, de los grupos originales (0,2) y (1,3) se obtuvieron dos términos $(a'c')$ y $(a'c)$ a los cuales se les puede aplicar el T. de Adyacencia:

$$F(a,b,c) = a' + (b'c)$$

Este ultimo paso depende de la correcta selección de los grupos de minitérminos (segundo paso).

3.4.2 DIAGRAMAS DE KARNAUGH

El método del diagrama de KARNAUGH permite de manera fácil y rápida realizar la reducción de expresiones de seis o menos variables utilizando el procedimiento anterior, es decir el uso intensivo del Teorema. de Adyacencia.

El diagrama de Karnaugh es una forma modificada de la tabla de la verdad donde las combinaciones de los valores de las variables de la función han sido reorganizados. El mapa de Karnaugh (M.K) de tres variables se muestra en la tabla 3-7.

Para n variables, el mapa de Karnaugh muestra los 2^n diferentes términos producto que se pueden formar; cada uno de ellos en una celda diferente. Se coloca "1" en cualquier celda que corresponda a los valores de variables para la cual la función tome el valor de "1", y se coloca "0" en las celdas donde la función toma ese valor.

		B C						B C			
A		00	01	11	10	A		00	01	11	10
0		0	1	3	2	0		$A'B'C'$	$A'B'C$	$A'BC$	$A'BC'$
1		4	5	7	6	1		$AB'C'$	$AB'C$	ABC	ABC'

Tabla 3-7 Mapa de Karnaugh de tres variables

Por tanto, cada una de las 2^n celdas en un mapa corresponde a uno de los posibles minitérminos de la función de n variables.

En la tabla 3-7 observamos que las celdas (o minitérminos) están localizados de tal forma que cualquier par de celdas inmediatamente adyacentes, horizontal o verticalmente entre sí, corresponden a un par de minitérminos que son lógicamente adya-

centes; es decir, cada par de celdas adyacentes difieren en la condición de una única variable (complementada o no).

El diagrama de Karnaugh es un medio de visualización de los valores de una función de manera gráfica. Debido a la forma en que están dispuestas las celdas al utilizar el código Gray y no el binario para la realización de los mapas, los conjuntos de minitérminos que se puedan combinar o reducir por el teorema de Adyacencia aparecerán adyacentes o en patrones simétricos en el mapa.

En las tabla 3-7 y tabla 3-8 se muestran los diagramas para tres y cuatro variables con la ubicación y la numeración de cada cuadro o celda según el minitérmino y de acuerdo con la elección de los pesos de las variables.

En las tablas 3-9, 3-10 y 3-11 se presentan los diagramas más usados para simplificar expresiones de cinco y seis variables respectivamente.

Frecuentemente, el M.K. para cinco variables consiste de dos mapas de cuatro variables y para seis variables en cuatro mapas de cuatro variables.

		C D						C D			
AB		00	01	11	10	AB		00	01	11	10
00		0	1	3	2	00		A'B'C'D'	A'B'C'D	A'B'CD	A'B'CD'
01		4	5	7	6	01		A'BC'D'	A'BC'D	A'BCD	A'BCD'
11		12	13	15	14	11		ABC'D'	ABC'D	ABCD	ABCD'
10		8	9	11	10	10		AB'C'D'	AB'C'D	AB'CD	AB'CD'

Tabla 3-8 Mapa de Karnaugh de cuatro variables

		D E						D E			
BC		00	01	11	10	BC		00	01	11	10
00		0	1	3	2	00		16	17	19	18
01		4	5	7	6	01		20	21	23	22
11		12	13	15	14	11		28	29	31	30
10		8	9	11	10	10		24	25	27	26

A = 0

A = 1

Tabla 3-9 Mapa de Karnaugh de cinco variables

		E	F	
CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

A = 0 B = 0

		E	F	
CD	00	01	11	10
00	16	17	19	18
01	20	21	23	22
11	28	29	31	30
10	24	25	27	26

A = 0 B = 1

		E	F	
CD	00	01	11	10
00	32	33	35	34
01	36	37	39	38
11	44	45	47	46
10	40	41	43	42

A = 1 B = 0

		E	F	
CD	00	01	11	10
00	48	49	51	50
01	52	53	55	54
11	60	61	63	62
10	56	57	59	58

A = 1 B = 1

Tabla 3-10 Mapa de Karnaugh de seis variables

		D E F						
ABC	000	001	011	010	110	111	101	100
000	0	1	3	2	6	7	5	4
001	8	9	11	10	14	15	13	12
011	24	25	27	26	30	31	29	28
010	16	17	19	18	22	23	21	20
110	48	49	51	50	54	55	53	52
111	56	57	59	57	62	63	61	60
101	40	41	43	42	46	47	45	44
100	32	33	35	34	38	39	37	36

Tabla 3-11 Otra forma del Mapa de Karnaugh de seis variables

3.4.3 CUBRIMIENTO DE UNA FUNCIÓN

Subcubos

Un subcubo de minitérminos es el conjunto o grupo de 2^m celdas, adyacentes entre sí, que contienen 1 lógicos en un mapa (la denominación "subcubo" proviene de los diagramas de Venn).

Un subcubo 0 ($m = 0$) consiste en una única celda y por tanto representa el minitérmino correspondiente.

Un subcubo 1 ($m = 1$) consiste en un grupo de dos celdas adyacentes y puesto que las dos celdas presentan una diferencia en solo una de sus variables, el grupo o

subcubo corresponde o equivale al término producto de las variables en que no difieren los dos minitérminos de las celdas (T. de Adyacencia). Es decir, el término producto de las variables que no difieren en el grupo o subcubo 1, cubrirá o incluirá los minitérminos de las dos celdas. Esto puede ser también interpretado como que el subcubo equivale a un término producto que incluye las variable que no varían en su presentación o condición.

En general, un grupo o subcubo de 2^m celdas adyacentes, en un mapa de n variables, corresponde al término producto de las variables en que no difieren los minitérminos de las celdas del subcubo o grupo. Este término tendrá $(n - m)$ variables que serán las mismas en todas las celdas o minitérminos, esto se puede deducir fácilmente al observar los códigos binarios de las celdas del grupo y determinar que las variables en que no difieren las celdas, son aquellas cuyo valor binario no cambia en ninguna de las celdas del grupo.

EJEMPLO 3-4

Para ilustrar el procedimiento de reducción a través de mapas simplifiquemos la función del ejemplo anterior utilizando M.K.

Sea la función: $F(A,B,C) = \sum m_i = \sum m(0,1,2,3,5)$

En la tabla 3-12 se presenta el mapa de tres variables para esta función, siendo la variable C la menos significativa y A la más significativa.

A	B C				A	B C			
	00	01	11	10		00	01	11	10
0	1	1	1	1	0	$A'B'C'$	$A'B'C$	$A'BC$	$A'BC'$
1	0	1	0	0	1	$AB'C'$	$AB'C$	ABC	ABC'

Tabla 3-12 Mapa de Karnaugh de $F(A,B,C) = \sum m(0,1,2,3,5)$

Existen varios subcubos o grupos de "1" que pueden cubrir todos los minitérminos de la función. Como se observa en la figura 3-31, los grupos que cubren la función y producen la menor cantidad de variables por término producto son el subcubo $m = 2$ correspondiente a los minitérminos (0,1,2,3), y el subcubo $m = 1$ para los minitérminos 1,5.

BC	00	01	11	10
A				
0	1	1	1	1
1	0	1	0	0

A'

B' C

Figura 3-31 Agrupación en el M. K. de S m (0,1,2,3,5)

El subcubo $m = 2$ o grupo de cuatro unos corresponde al término producto de las variables en que no difieren los minitérminos de las celdas del grupo, es decir A' , esto se deduce al observar los códigos binarios de las celdas del grupo y determinar que las variables en que no difieren las celdas son aquellas cuyo valor binario no cambia en ninguna de las celdas del grupo. El grupo de dos unos corresponde por el mismo principio a $B' C$.

En la figura 3-31 se resaltan los grupos y se indica el término producto correspondiente a cada uno de ellos. Tal como se dedujo en el procedimiento algebraico, la función puede ser expresada como:

$$F(A,B,C) = \sum m(0,1,2,3,5) = A' + B' C$$

3.4.4 SIMPLIFICACIÓN POR KARNAUGH

La reducción de una función consiste en encontrar la expresión mínima que cubre o incluye todos sus minitérminos. Es evidente que si un minitérmino puede cubrirse con más de un subcubo, debe utilizarse el mayor subcubo (subcubo máximo para ese minitérmino) para así generar un término producto con la menor cantidad de variables, el cual lo incluya o cubra. El término producto correspondiente al grupo máximo se denomina **implicante primo**.

Adicionalmente, la expresión mínima de una función debe tener la menor cantidad de términos posibles.

Concretando, las reglas para encontrar la expresión mínima de una función son:

- Cada minitérmino debe ser cubierto por un subcubo máximo.
- La función debe ser cubierta por la menor cantidad de subcubos máximos.

- La expresión mínima será la suma de los implicantes primos resultantes.

Es importante entender que un minitérmino puede estar incluido en más de un grupo con el objetivo de crear un subcubo máximo para otro u otros minitérminos.

También debe ser claro que un subcubo máximo en el que todos sus minitérminos estén incluidos en otros grupos máximos es redundante.

EJEMPLO 3-5

Sea la función: $F(A,B,C,D) = \sum m_i = \sum m(4,6,7,12,14)$

El mapa de cuatro variables para esta función y los subcubos que lo cubren es mostrado en la figura 3-32. La expresión mínima de la función es:

$$F(A,B,C,D) = BD' + A'BC$$

AB \ CD	00	01	11	10
00				
01	1		1	1
11	1			1
10				

Figura 3-32 Mapa K. de $F(A,B,C,D) = \sum m(4,6,7,12,14)$

EJEMPLO 3-6

Sea la función: $G(A,B,C,D) = \sum m_i = \sum m(0,1,4,5,11,13,15)$

La figura 3-33 presenta el mapa para esta función y un posible agrupamiento en el mapa, las celdas vacías se suponen con cero.

AB \ CD	00	01	11	10
00	1	1		
01	1	1		
11		1	1	
10			1	

Figura 3-33 Mapa K. de $G(A,B,C,D) = \sum m(0,1,4,5,11,13,15)$

La expresión mínima de la función según esta agrupación es:

$$G(A,B,C,D) = A' C' + BC' D + ACD$$

Otra opción de agrupamiento es mostrada en la figura 3-34. La expresión mínima de la función para esta selección de subcubos máximos es:

$$G(A,B,C,D) = A' C' + ABD + ACD$$

A B	C D	0 0	0 1	1 1	1 0
0 0		1	1		
0 1		1	1		
1 1			1	1	
1 0				1	

Figura 3-34 Mapa K. de $G(A,B,C,D)$ (segunda opción)

EJEMPLO 3-7

Sea la función: $H(A,B,C,D) = \sum m_i = \sum m (0,1,2,5,7,10,11)$

En la figura 3-35 se muestra uno de los posibles agrupamientos el cual genera la mínima expresión. El cubrimiento u obtención de los subcubos se debe comenzar por los minitérminos 7 y 11, puesto que no existe más alternativas que el subcubo máximo marcado. Para los otros minitérminos se toman en cuenta las reglas mencionadas obteniendo: $H(A,B,C,D) = A' B' C' + A' BD + B' CD' + AB' C$.

A B	C D	0 0	0 1	1 1	1 0
0 0		1	1		1
0 1			1	1	
1 1					
1 0				1	1

Figura 3-35 Mapa K. de $H(A,B,C,D) = \sum m (0,1,2,5,7,10,11)$

EJEMPLO 3-8

Sea la función: $K(A,B,C,D,E) = \sum m_i = \sum m (4,5,9,10,14,20,21,25,26,30)$

En este caso se tiene una expresión de cinco variables cuyo mapa se muestra en la figura 3-36. Este mapa consiste de dos submapas de cuatro variables.

En el mapa se indican las agrupaciones y sus respectivos términos productos. Es obvio que celdas o grupo de celdas similares con igual posición relativa en ambos submapas, solo difieren en la variable A; lo que permite realizar subcubos o grupos que incluyan celdas de ambos diagramas.

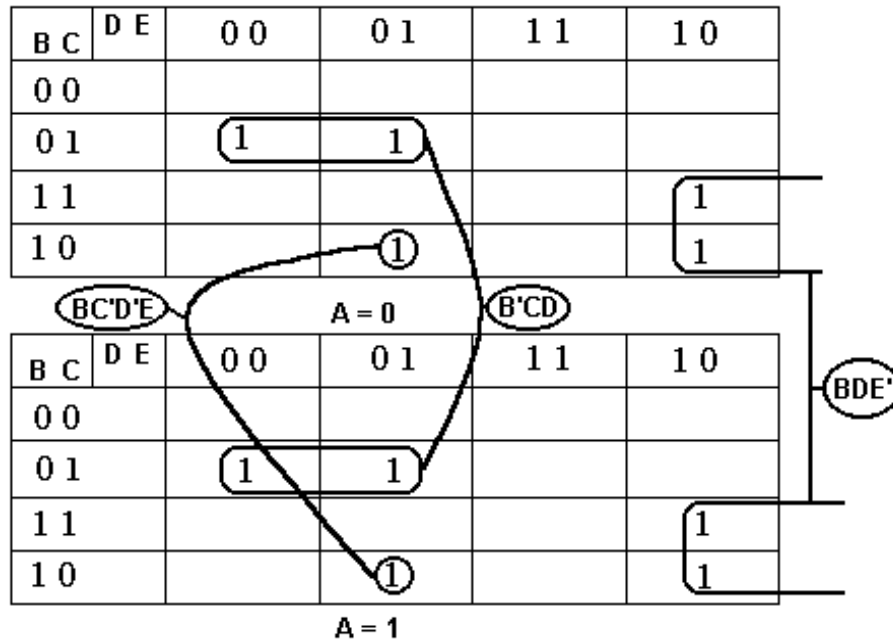


Figura 3-36 Mapa K. de $K(A,B,C,D,E) = \Sigma m(4,5,9,10,14,20,21,25,26,30)$

Los grupos seleccionados en la figura 3-36 dan como resultado:

$$K(A,B,C,D,E) = B'CD + BC'D'E + DE'$$

Es importante observar que se escoge por conveniencia a la variable más significativa **A** como aquella en que difieren los dos submapas.

3.4.5 RIESGOS

Debido a los retrasos en los circuitos, el comportamiento transitorio de un circuito lógico puede diferir de lo que predice el análisis de estado estacionario. En particular, la salida de un circuito puede producir un pulso corto, a menudo llamado falla o spike, cuando el análisis de estado estacionario predice que la salida no debe cambiar. Se dice que existe un riesgo cuando un circuito tiene la posibilidad de producir tal falla. Como los retrasos son difíciles de controlar en la producción de circuitos, un diseñador lógico debe estar preparado para eliminar los riesgos (posible falla) inclusive cuando esta pueda ocurrir en las condiciones lógicas eléctricas menos probable.

Riesgo ó peligro estático uno

Por definición es la posibilidad de que una salida de un circuito produzca una falla de 0 (cero) cuando esperamos que la salida permanezca en un 1 (uno) estacionario. El circuito de la figura 3-37 presenta este tipo de riesgo.

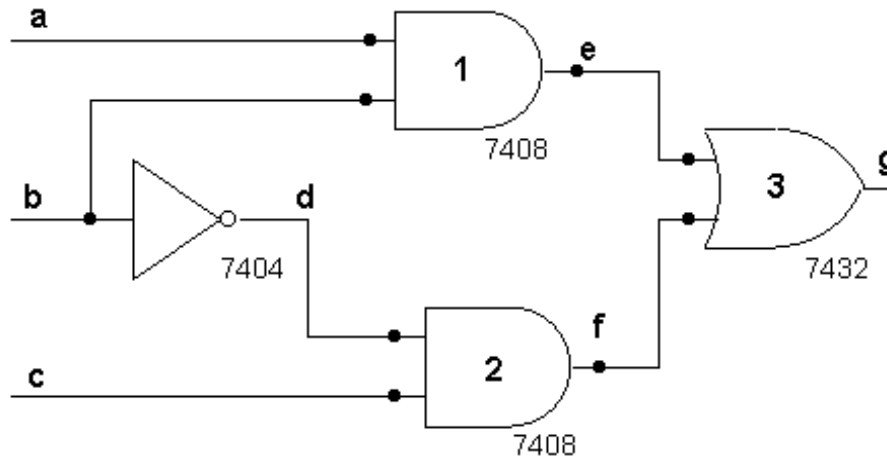


Figura 3-37 Circuito con riesgo estático uno

Esto se ilustra en el diagrama de tiempo de la figura 3-38, donde al inicio las tres entradas son iguales a 1 y por tanto la salida del circuito es igual a 1. Se observa que un cambio de la entrada b desde 1 a 0 no debería producir cambio alguno en la salida puesto que la salida de la compuerta 1 cambia a 0 y la de la compuerta 2 cambia a 1. Sin embargo, en forma momentánea la salida puede pasar a 0 si el retardo de propagación del inversor no es despreciable.

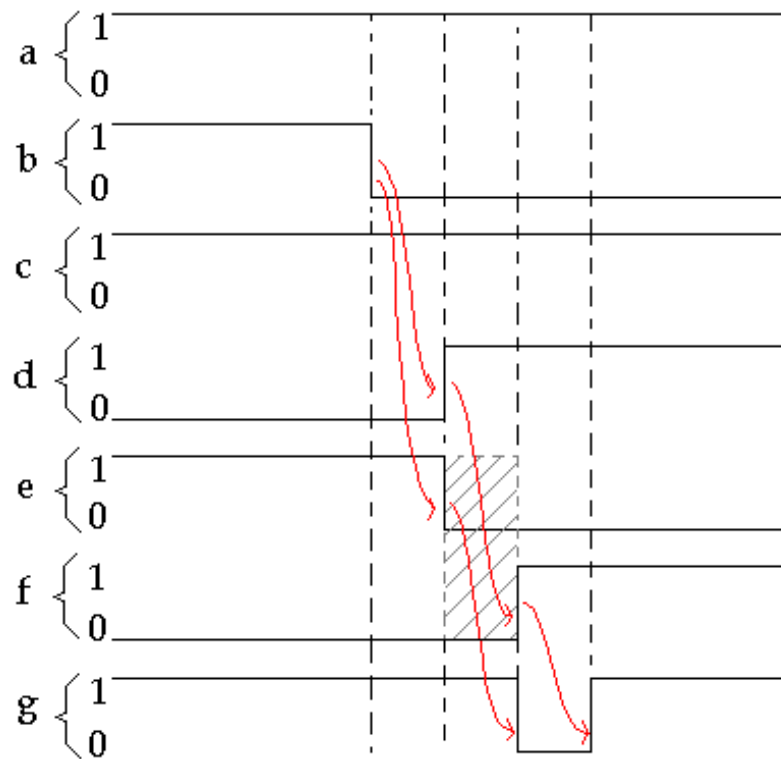


Figura 3-38 Diagrama de tiempos de un riesgo estático uno

Riesgo ó peligro estático cero

Es el dual del riesgo estático 1, y se define como la posibilidad de una falla 1 cuando esperamos que el circuito tenga una salida estable de cero.

Riesgos dinámicos

Este peligro o riesgo causa que la salida cambie tres o más veces cuando debe cambiar una sola vez (1 a 0 o 0 a 1). Esto puede ocurrir si hay múltiples rutas con diferentes retrasos desde la entrada cambiante a la salida.

Determinación de riesgos en un circuito

Los riesgos dinámicos al igual que los estáticos pueden ser eliminados fácilmente por el uso de herramientas de diseño CAD especiales. Lo importante es que el ingeniero sea consciente de su posibilidad de existencia y de las consecuencias de estos.

Por medio del mapa de Karnaugh se pueden detectar riesgos estáticos en un circuito de SP o de PS de dos niveles.

En relación al circuito de la figura 3-37, se observa de su mapa (figura 3-39) que el cambio indeseable en la salida ocurre porque para la entrada $b = 1$ la salida es 1 debido a un término producto o grupo (salida de la compuerta AND 1), mientras que para la entrada $b = 0$ lo es por otro término producto o grupo (salida de la compuerta AND 2).

$a \backslash b \ c$	00	01	11	10
0		1		
1		1	1	1

Figura 3-39 Mapa K indicando el cambio de una entrada

Siempre que una señal cambie la salida de un término producto a otro, hay la posibilidad de un intervalo momentáneo cuando ningún término es igual a 1, dando lugar a una salida indeseable 0.

Circuitos libres de riesgos.

El método para eliminar un riesgo es unir los dos minitérminos separados con otro término producto que conecte ambos agrupamientos. Los minitérminos que causan el peligro se deben combinar en un término producto como en la figura 3-40.

$a \backslash b \ c$	00	01	11	10
0		1		
1		1	1	1

Figura 3-40 Eliminación de riesgo por redundancia

El circuito libre de riesgo obtenido por esta configuración se muestra en la figura 3-41 y corresponde a la expresión: $g = a.b + a.c + b'.c$

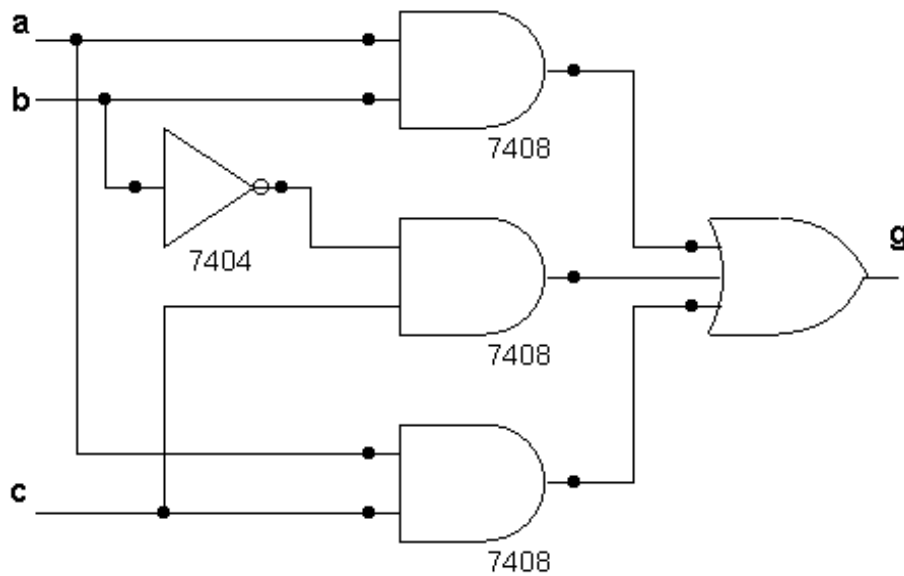


Figura 3-41 Circuito libre de riesgos

Notese que la remoción de peligros requiere la adición en el circuito de compuertas redundantes.

3.4.6 CONDICIONES NO IMPORTA

Las condiciones no importa (don't care) se dan en funciones incompletamente especificadas. Los valores de una función para todas las posibles combinaciones de las variables permiten construir la tabla de la verdad y el mapa de Karnaugh donde se listan las condiciones bajo las cuales la función es 1 y 0; pero, hay aplicaciones donde la función para ciertas combinaciones de las variables de entrada no está definida explícitamente.

Por ejemplo, para el código BCD de cuatro bits existen seis combinaciones (del 10 al 15) que no existen para este código. Por tanto, cualquier circuito digital que utiliza este código puede operar bajo la suposición de que estas combinaciones nunca ocurrirán, mientras el sistema trabaje apropiadamente. Como resultado, para esas combinaciones de las variables de entrada, no importa que valor se especifique (0 o 1) para la función, ya que estas no deberían ocurrir.

En general, puede ocurrir que para algunas condiciones de las variables el valor de la función lógica no se pueda inferir del funcionamiento o requerimientos del circuito.

Estas condiciones “*no importa*” pueden usarse en un mapa para proporcionar una simplificación adicional de la función.

Para distinguir en el mapa las condiciones no especificadas, se estila utilizar una equis (**X**). Cuando se escogen cuadros adyacentes para simplificar la función en el mapa puede suponerse que las condiciones (**X**) son ceros o unos, según convenga para hacer la expresión lo más simple.

Notación

Para poder expresar una función con condiciones incompletamente especificadas es común expresarla en forma de una suma (producto) de minitérminos (maxitérminos) y los términos no importa.

$$G (w,x,y,z) = \sum (1, 2, 3, 4, 8) + \text{No Importa} (12, 13, 14)$$

$$F (w,x,y,z) = \sum (1, 2, 4, 10) + \text{don't care} (11, 12, 13, 14, 15)$$

$$F (w,x,y,z) = \sum (1, 2, 4, 10) + d (11, 12, 13, 14, 15)$$

EJEMPLO 3-9

Diseñe un circuito que indique cuando una entrada de 4 bits (un nibble) en código BCD es equivalente a los números decimales 1,2,4 y 8.

Solución:

Se procede a resolver el problema usando el método del mapa. El código BCD incluye los números del 0 al 9 (0000 al 1001). Las celdas del 10 al 15 también son posibles, pero no pertenecen al código BCD, puesto que no se indica como debe actuar el circuito ante estas combinaciones de entrada se tratarán como condiciones no especificadas (no importa).

Eligiendo el valor de 1 para la salida F del circuito cuando el código BCD es equivalente a los números decimales 1,2,4 y 8 se tiene que:

$$F (w,x,y,z) = \sum (1, 2, 4, 8) + d (10, 11, 12, 13, 14, 15)$$

La figura 3-42 presenta el mapa de Karnaugh de la función y las condiciones No Importa. Escogiendo adecuadamente el valor para dichas condiciones se obtiene el mapa de la figura 3-43 del cual se deducen cuatro cubos o grupos.

- $m(1) = w'x'y'z$ $m(2, 10) = x'yz'$
- $m(4, 12) = xy'z'$ $m(8, 10, 12, 14) = wz'$

wx	yz	0 0	0 1	1 1	1 0
0 0		0 ₀	1 ₁	0 ₃	1 ₂
0 1		1 ₄	0 ₅	0 ₇	0 ₆
1 1		X ₁₂	X ₁₃	X ₁₅	X ₁₄
1 0		1 ₈	0 ₉	X ₁₁	X ₁₀

Figura 3-42 Mapa K de F y condiciones No Importa

wx	yz	0 0	0 1	1 1	1 0
0 0		0 ₀	1 ₁	0 ₃	1 ₂
0 1		1 ₄	0 ₅	0 ₇	0 ₆
1 1		X ₁₂	X ₁₃	X ₁₅	X ₁₄
1 0		1 ₈	0 ₉	X ₁₁	X ₁₀

Figura 3-43 Agrupamiento con condiciones No Importa

El agrupamiento adecuado para obtener la mínima expresión da como resultado:

$$F = w'x'y'z + x'yz' + xy'z' + wz'$$