



UNIDAD 9:INTRODUCCIÓN A FALLAS EN CIRCUITOS LÓGICOS.

(86:44) Técnica Digital Avanzada- Unidad 9.
Profesor: Ing. Miguel Antonio Martínez.

Fallas en circuitos lógicos.

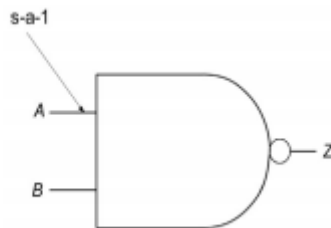
Se llama **falla** la ocurrencia de un hecho que haga desviar al circuito lógico de su comportamiento específico. En general, una falla tiene origen en un defecto físico de un circuito. Por ejemplo, un corto entre dos líneas de señal o un circuito abierto en un camino de señal son defectos físicos. Un **error** es usualmente la manifestación de una falla en el circuito. Por ejemplo, una falla en un circuito puede hacer cambiar el valor lógico de una variable, pasar de un valor correcto en "0" a uno erróneo en "1" o viceversa. No siempre una falla causa un error, en ese caso a la falla se la llama **latente**.

Una falla se caracteriza por su naturaleza, valor, extensión y duración. La naturaleza de una falla se clasifica en **lógicas y no lógicas**. Una falla lógica causa que un valor lógico en un punto del circuito cambie en oposición al valor correcto. Una falla lógica puede ser por un mal funcionamiento de una línea de clock, por problemas en la fuente de alimentación, etc. El valor de una falla indica si en un punto del circuito se fijan valores erróneos fijos o variables. La extensión de una falla indica cuando el efecto de la misma es localizado o distribuido. La duración de una falla indica si el resultado de la misma es temporario o permanente.

Stuck at Fault.

El más común modelo usado para fallas lógicas es el llamado **stuck at fault**. Generalmente se traduce como **atascado**. El método consiste en asumir que una falla en una compuerta lógica resulta en una de sus entradas o salidas las señales quedan fijas o atascadas. Pueden fijarse en 0 (stuck at 0) o en 1 (stuck at 1). Estas fallas son abreviadas de la siguiente manera, a stuck at 0 se la llama **s-a-0** ya a stuck at 1 como **s-a-1**.

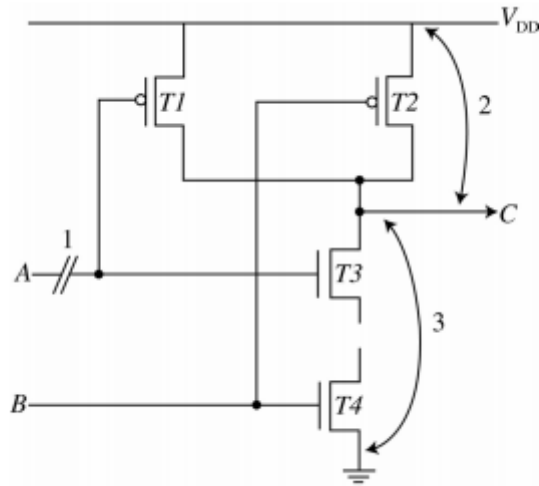
Veamos la siguiente figura:



Asumimos que la entrada A de la compuerta NAND está atascada a 1, o sea s-a-1. La compuerta percibe que en la entrada A hay un 1 independientemente del valor lógico colocado en esa entrada. Por ejemplo, la salida de la compuerta es 1 para la combinación de entrada A = 0 y B = 1, pero como la entrada A está bloqueada en 1, la salida dará un 0. En ausencia de falla, para esta combinación de entrada, la salida vale 1. Quiere decir que la combinación A = 0 y B = 1 puede considerarse como un **test** para detectar **s-a-1** en la entrada A, ya que hay diferencia entre la compuerta en buen estado y defectuosa.

Este modelo de stuck at fault se suele llamar **modelo de falla clásica** y ofrece una buena representación para los defectos más comunes como ser cortocircuitos o circuitos abiertos en la tecnología CMOS.

En la siguiente figura vemos el esquema circuital de una compuerta NAND de dos entradas.



El numero 1 indica un circuito abierto, en cambio los números 2 y 3 representan un cortocircuito entre el nodo de salida y V_{DD} y entre el nodo de salida y masa respectivamente. Un cortocircuito en un dispositivo CMOS indica que no se ha removido el suficiente metal en el proceso de fotolitografía, mientras que la extracción excesiva de metal produce circuitos abiertos. La falla 1 produce la desconexión de la entrada A con los transistores T₁ y T₃. Sabemos que en esta situación un transistor puede conducir y el otro estar en condición de no conducción. Entonces, esta falla puede representarse como un atascamiento de la entrada A. Si A es s-a-0, T₁ conducirá y T₃ no, y si A es s-a-1, T₁ estará en OFF y T₃ en ON. La falla 2 fuerza la salida al valor de V_{DD}, por lo tanto, la falla puede ser considerada como s-a-1. De la misma forma la falla 3 fuerza la salida a masa, por lo tanto, puede considerarse como s-a-0.

Este modelo también se usa para representar varias fallas de bloqueo o atascamiento. En este caso se supone que varias líneas del circuito están bloqueadas a 0 lógico o a 1 lógico. O sea, hay varias fallas atascadas en el circuito a la vez. Una variación de las fallas múltiples es la que se llama **falla unidireccional**. Este tipo de fallas es aquella que todas las fallas constituyentes de la misma son s-a-0 o s-a-1, pero no ambas simultáneamente.

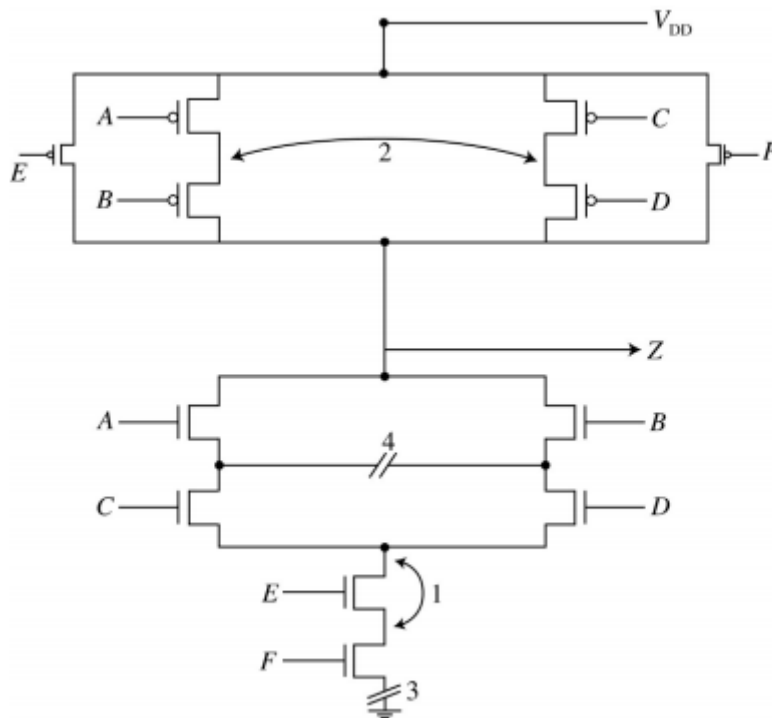
Este modelo ha ganado mucho terreno en el pasado con circuitos de baja escala de integración. Sin embargo, no es bueno para detectar todas las fallas de un circuito de alta escala de integración (VLSI), principalmente de tecnología CMOS. Las fallas en los circuitos CMOS no necesariamente presentan fallas lógicas que puedan ser representadas por este modelo de stuck at fault.

En la figura siguiente vemos el circuito CMOS que implementa la función:

$$Z = \overline{(A + B)(C + D)} \cdot EF.$$

En el circuito representamos dos fallas llamadas 1 y 2 que representan cortocircuitos y dos fallas 3 y 4 que representan circuitos abiertos. El corto 1 puede ser

representado por s-a-1 de la entrada E, el circuito abierto 3 puede representar un s-a-0 en la entrada E, en la entrada F o en ambas.



Por otro lado, el corto 2 y el circuito abierto 4 no pueden ser modelizados por el método de stuck at fault. Esto se debe que estas dos fallas producen un cambio en la topología del circuito. Por ejemplo, en el caso de presentarse el corto 2, la función lógica del circuito pasa a ser:

$$Z = \overline{(A + C)(B + D)} \cdot EF,$$

Y un circuito abierto representado por 4 hace que la función cambie a:

$$Z = \overline{(AC) + (BD)} \cdot EF.$$

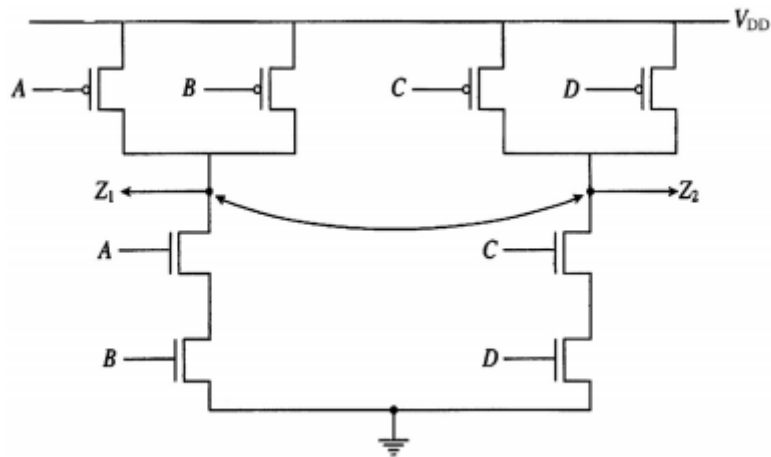
Por esa razón, un corto entre las salidas de las dos compuertas no puede ser modelado por este método de stuck at fault. Sin la existencia de ese corto la salida de las compuertas es:

$$Z_1 = \overline{AB} \quad \text{and} \quad Z_2 = \overline{CD},$$

Mientras que con el corto la funciones quedan:

$$Z_1 = Z_2 = \overline{AB} + \overline{CD}.$$

El circuito representante de estas últimas expresiones (con y sin corto) se ve en la siguiente figura. Ver cómo ha cambiado la topología y la función lógica del mismo.



Fallas de puentes.

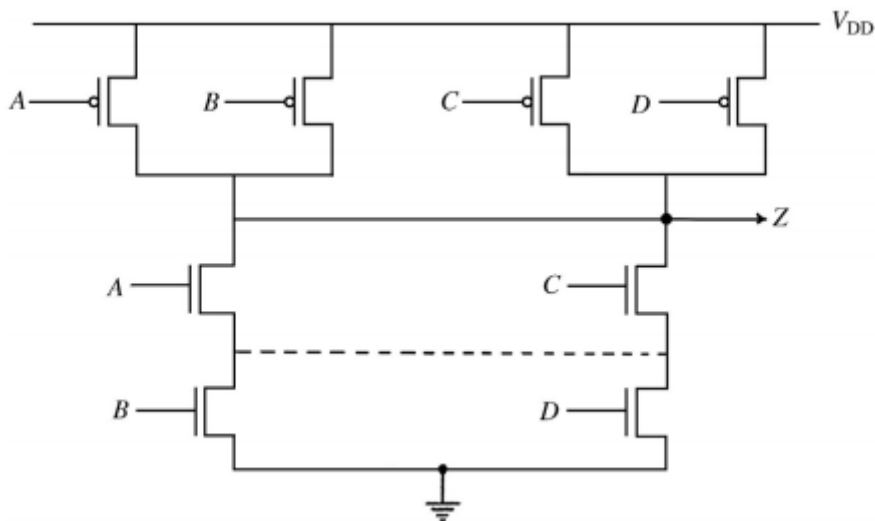
Las fallas en forma de puente son una clase importante de fallas permanentes que no pueden ser modelizadas por el método de Stuck at Fault. Se dice que hay una falla en forma de puente cuando dos o más líneas de señal en un circuito son conectadas entre sí. El estudio anterior se centraba en el cortocircuito a nivel de compuertas, esto daba como resultado una especie de **lógica cableada (wired logic)**.

Las fallas de puente a nivel de compuertas pueden ser clasificados en dos tipos: **los puentes a la entrada de las compuertas y los puentes que producen realimentación**. Las primeras corresponden a cortos en cierto número de líneas de entrada. En cambio, las de realimentación corresponden a un corto entre la salida y alguna de las líneas de entrada. Estas últimas fallas pueden producir que el circuito oscile o se convierta en secuencial.

Una falla de puente a nivel de transistores puede ocurrir entre los terminales del transistor o entre dos o más líneas de señal.

En la siguiente figura vemos un circuito que realiza la siguiente función booleana:

$$Z_1 = Z_2 = \overline{AB} + \overline{CD}$$



La falla de puente entre dos líneas está graficada con la línea punteada, lo cual hace que cambie la función del circuito.

El efecto de un puente entre terminales de los transistores depende de la tecnología. Por ejemplo, en CMOS estas se manifiestan como fallas atascadas o como circuitos abiertos dependiendo de la ubicación física y de la resistencia óhmica que presente el puente.

Fallas de retardo.

Como hablamos anteriormente no todos los defectos de fabricación de los circuitos VLSI pueden representarse con el modelo de stuck at fault. La extensión de una falla determinará si este defecto afectará la función lógica del circuito. Las fallas producidas por cortocircuitos o circuitos abiertos tienen más probabilidad de aparición debido a la forma de fabricación.

Las fallas de retardo son fallas que no cambian la función lógica. Este tipo de fallas no cambia la función, pero no cumple con las especificaciones de tiempo previstas. Por ejemplo, puede haber un retraso en la transición de una línea de 0 a 1 o viceversa. Este tipo de mal funcionamiento se llaman **fallas de retraso**. Este tipo de fallas se pueden dividir en fallas de retardo de compuertas o en retardo de líneas.

Por ejemplo, si una compuerta tiene un tiempo de propagación de x unidades, y en realidad tiene $x + \Delta x$ unidades, se dice que hay una falla de retardo de Δx .

Este tipo de modelo de fallas sirve para modelar fallas aisladas o no distribuidas. Un ejemplo de esto último serían varios retrasos pequeños de la señal. En el caso del retardo de línea es si el tiempo de propagación de una línea supera el especificado.

Fallas en circuitos CMOS.

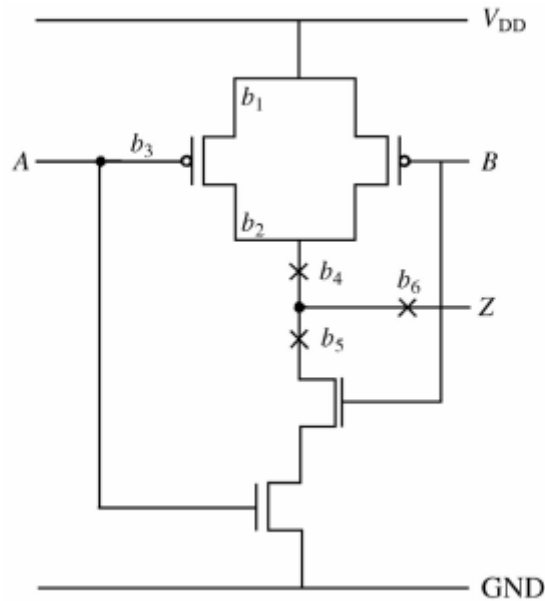
Como discutimos anteriormente, no todos los defectos en circuitos CMOS de gran escala pueden ser representados con el modelo de stuck at fault. Se ha demostrado que los transistores abiertos o en cortocircuito como así también las fallas de puenteo pueden pasar desapercibidas si usamos este método.

Los circuitos abiertos en CMOS generalmente se deben a la falta de material conductor en el proceso de implantación de los transistores. Como vimos antes, estas fallas pueden ser de dos tipos, los circuitos abiertos **intracompuerta** o los circuitos abiertos en las **líneas de señal**.

Como lo dice la palabra, un circuito abierto intracompuerta es un corte en la continuidad dentro de una compuerta, generalmente por la desconexión del source, del drain o del gate en un transistor CMOS. Esto está expresado en la siguiente figura por las letras b_1 , b_2 y b_3 respectivamente.

La presencia de b_3 no cambia el funcionamiento de la lógica del circuito, pero si cambia el retardo de propagación, con lo cual un circuito abierto en b_3 representará un tipo de falla de retardo. Similarmente un corte en b_1 también produce un incremento del retardo de la compuerta sin cambiar la función lógica del circuito. Contrariamente,

un corte en b_2 hará el transistor canal p no conduzca, por lo tanto, se puede interpretar al transistor como bloqueado abierto.



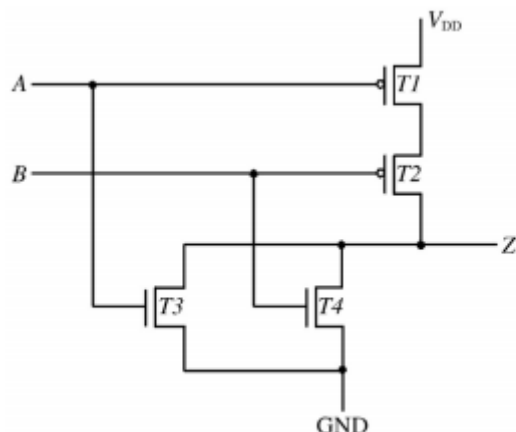
Un corte intracompuerta también puede desconectar una rama del circuito, como lo indican los cortes b_4 , b_5 y b_6 . La presencia de b_4 o b_5 hará que el nodo de salida quede bloqueado a 0 o a 1, o sea s-a-0 o s-a-1 respectivamente. En presencia de b_6 , el voltaje de salida tendrá un valor intermitente entre s-a-0 y s-a-1. Esto se debe a que uno de los transistores conduce en un periodo de tiempo impredecible.

Hay fallas llamadas de transistores bloqueados o transistores abiertos, un transistor bloqueado significa que tiene cerrado el paso entre el terminal de source y el terminal de drain. Aunque el transistor bloqueado se comporta en la práctica como uno atascado cerrado, hay una pequeña diferencia. Mientras un transistor bloqueado tiene la misma resistencia de drenaje que uno sin fallas, la resistencia que presenta uno bloqueado cerrado presente una resistencia mucho menor. En el caso de un transistor bloqueado cerrado, el cortocircuito entre el terminal de source y el de drain es casi perfecto, y esto no es cierto para un transistor bloqueado. Un transistor bloqueado significa implica la apertura permanente de la conexión entre el source y el drain. Pero no alarmarse, ya que las fallas producidas por un transistor bloqueado y bloqueado abierto se consideran equivalentes.

Una falla en un transistor que se comporte como bloqueado abierto puede representar una realimentación que haga funcionar un circuito combinacional como secuencial.

En la figura siguiente vemos el circuito de una compuerta NOR de dos entradas en tecnología CMOS. Una falla bloqueada abierta hace que la salida no se conecte ni a VDD ni a GND. Por ejemplo, si el transistor T_2 es un circuito abierto, cuando las entradas sean $AB = 00$, el circuito de pull-up no se activa y no habrá cambio en la tensión de salida. O sea, la salida conserva el valor lógico anterior, aunque el tiempo de duración de la misma dependerá de la corriente de fuga del nodo de salida.

El circuito que hacemos referencia es el siguiente:



En la tabla mostrada a continuación vemos el comportamiento de este circuito ante diferentes fallas:

A	B	Z	$Z(A_{s-op})$	$Z(B_{s-op})$	$Z(V_{DDs-op})$
0	0	1	1	1	Z_t
0	1	0	0	Z_t	0
1	0	0	Z_t	0	0
1	1	0	0	0	0

El circuito sin ningún tipo de falla se muestra la columna Z . Las tres columnas de la derecha muestran el circuito con fallas del tipo bloqueadas abiertas. La primera (A_{s-op}) es causada por cualquier entrada, drain o source desconectados del pull-down que representa el FET T_3 . La columna (B_{s-op}) muestra una falla causada por cualquier entrada, drain o source desconectados del pull-down que representa el transistor FET T_4 . La última columna (V_{DDs-op}) muestra una falla producida por la apertura de cualquiera de los transistores de pull-up conectados a VDD. Los símbolos Z_i indican que la salida retiene el valor lógico que tenía anteriormente.

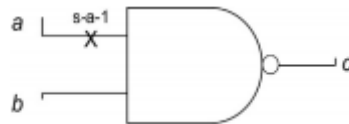
Conceptos básicos de la detección de fallas.

La detección de fallas en un circuito lógico consiste en la aplicación de una secuencia de pruebas y observando el valor de la salida del mismo. Una **prueba o test** es una combinación de las variables de entrada que especifica la respuesta esperada que debe producir el circuito. Si la respuesta observada es diferente a la esperada, hay una **falla** presente en el circuito.

El objetivo de las pruebas a nivel de compuertas es verificar que las mismas funciones correctamente y la conexiones entre las mismas estén en buenas condiciones.

Se supone que en el circuito existe **una sola falla**, entonces la premisa es generar un numero de pruebas que la pueda detectar utilizando solo las entradas y las salidas del mismo.

Como indicamos antes, la **prueba o test** detecta una falla si y solo sí, la salida producida por el circuito en presencia de la falla es **distinta** a la salida que presenta el circuito en ausencia de dicha falla. Como ejemplo veremos la siguiente figura donde se observa una compuerta NAND.

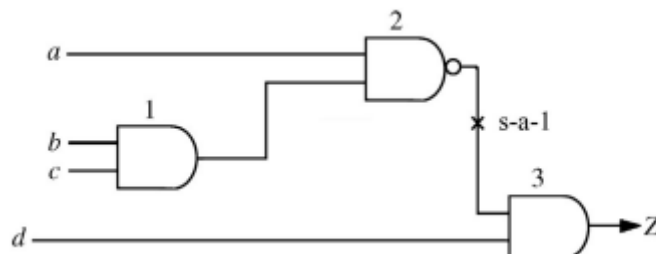


Asumimos que la entrada **a** de la compuerta está bloqueada a 1, o sea s-a-1. La tabla de verdad de la compuerta con y sin esta falla se ve en la siguiente figura:

TABLE 1.2: Output response of the NAND gate			
INPUT		OUTPUT	
<i>a</i>	<i>b</i>	<i>c</i> (FAULT-FREE)	<i>c</i> (FAULT-PRESENT)
0	0	1	1
0	1	1	0
1	0	1	1
1	1	0	0

Como se ve en la tabla, hay una sola combinación de las variables de entrada que da distinta la salida con y sin falla, la misma es la combinación **ab = 01** cuando se produce una falla en la entrada **a** del tipo **s-a-1**.

O sea, para detectar una falla en un circuito, lo primero es **excitar la falla**, o sea buscar cual es la combinación de las variables de entrada que hacen aparecer la falla. Luego la falla hay que **sensibilizarla**, o sea hacer que la falla fluya dentro del circuito hasta alguna salida del mismo donde este error sea observado. Por ejemplo, veamos el siguiente circuito:



La función lógica que realiza este circuito es:

$$F = (\overline{a \cdot b \cdot c}) \cdot d$$

Si escribimos la tabla de verdad de este circuito con y sin falla nos queda lo siguiente:

a b c d	sin falla	con falla
0 0 0 0	0	0
0 0 0 1	1	1
0 0 1 0	0	0
0 0 1 1	1	1
0 1 0 0	0	0
0 1 0 1	1	1
0 1 1 0	0	0
0 1 1 1	1	1
1 0 0 0	0	0
1 0 0 1	1	1
1 0 1 0	0	0
1 0 1 1	1	1
1 1 0 0	0	0
1 1 0 1	1	1
1 1 1 0	0	0
1 1 1 1	0	1

La combinación de entrada que debemos usar para **excitar la falla es a b c = 1 1 1** y para **sensibilizar** la falla se debe cumplir que **d = 1**. Por lo tanto, el **vector de prueba** para detectar un **s-a-1** en ese punto del circuito debe ser **a b c d = 1 1 1 1**.

Controlabilidad y Observabilidad.

La habilidad de aplicar un patrón de entrada a las entradas del circuito para configurar un valor lógico determinado en la ubicación deseada dentro del circuito se conoce como **controlabilidad**.

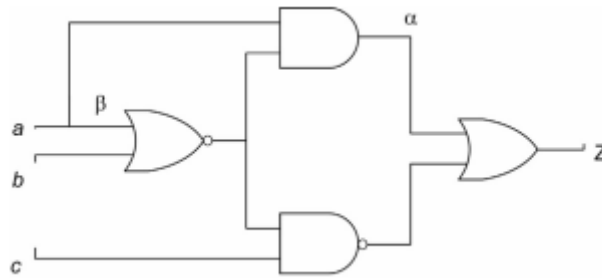
La parte de sensibilización del proceso de generación de pruebas requiere la aplicación de apropiados valores de entrada para que el efecto de la falla sea observable en las salidas. Por ejemplo, en el circuito de la figura anterior, el efecto de s-a-1 puede ser observable si la entrada "d" está seteada en 1, si d = 0 la falla se ve enmascarada (no se detecta la falla).

La habilidad de observar una respuesta de una falla de un nodo interno en una salida de un circuito se conoce como **observabilidad**.

Fallas indetectables.

Una falla se denomina **indetectable** si no es posible activar la falla o no se puede sensibilizar el efecto y observarlo en una salida. En otras palabras, no existe ningún a ni

prueba que pueda detectar la falla. Para ver un ejemplo de esto, veremos el siguiente circuito:



Demostraremos que las fallas en el punto α s-a-0 y la falla en el punto β s-a-0 son indetectables. Para ello si vemos el circuito y después de algunas manipulaciones de lógica booleana se llega a que la función de salida z tiene la siguiente expresión:

$$Z = a + b + \bar{c}$$

Al igual que en el ejemplo anterior hacemos la tabla de verdad con la salida sin fallas y las salidas suponiendo las fallas en los puntos α y β .

a b c	Z	α s-a-0	β s-a-0
0 0 0	1	1	1
0 0 1	0	0	0
0 1 0	1	1	1
0 1 1	1	1	1
1 0 0	1	1	1
1 0 1	1	1	0
1 1 0	1	1	1
1 1 1	1	1	1

Primero consideramos la falla α s-a-0, la misma es indetectable ya que no se puede verificar que cambie la salida con falla de la salida sin falla, lo vemos en la tabla de verdad.

Respecto de la falla β s-a-0 también es indetectable. Aunque se ve que la falla se detectaría con la combinación $ab = 10$, pero no hay un camino para sensibilizar la misma, y además si β está bloqueada en cero es lo mismo que $a = 0$ y no se observa efecto en la salida de dicha falla.

Los circuitos que presentan alguna falla indetectable se denominan **redundantes**.

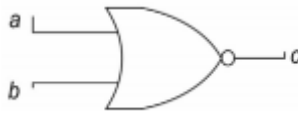
Un conjunto de pruebas en un circuito se basa en la suposición que existe una sola falla. Por lo tanto, la presencia de una falla indetectable y una detectable viola esta hipótesis.

Además, la presencia de una falla indetectable puede evitar detectar otra falla detectable.

Fallas equivalentes.

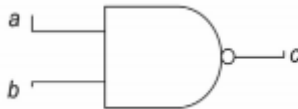
Una prueba o test, en general, puede detectar más de una falla. Y un conjunto de muchas pruebas pueden detectar la misma falla. Los subconjuntos de fallas detectados por una prueba no son disjuntos. Por lo tanto, el objetivo principal de la generación de pruebas es reducir el número total de fallas a ser consideradas agrupándolas en subconjuntos equivalentes. Entonces es suficiente con probar una falla de cada subconjunto para cubrir todas las fallas del conjunto, evitando así las redundancias en el proceso de generación de pruebas.

En una compuerta de m entradas, puede haber $2(m + 1)$ fallas del tipo bloqueo (stuck at fault). Por ejemplo, la siguiente compuerta que es una NOR de dos entradas:



Aplicando la formula vista antes tenemos **2 entradas**, por lo tanto, hay **2(2 + 1)** bloqueos posibles, ellos son a s-a-0, b s-a-0, a s-a-1, b s-a-1, c s-a-0 y c s-a-1. Sin embargo, un bloqueo en una entrada puede ser indistinguible con un bloqueo en una salida. Por ejemplo: en la compuerta NOR presentada una falla en cualquier entrada del tipo s-a-1 es indistinguible con una falla a la salida s-a-0.

De la misma manera en la siguiente compuerta NAND.



Cualquier falla en la entrada del tipo s-a-0 es indistinguible con una salida s-a-1.

Dos fallas se consideran **equivalentes** si cualquier test para una también detecta la otra. En la compuerta NOR, una falla s-a-1 en cualquiera de las entradas "a" o "b" es equivalente a una falla en la salida "c" del tipo s-a-0. Por lo tanto, las tres deben ser parte del mismo **conjunto de equivalencia**. Una prueba en cualquiera de las tres también detectará las otras dos. Los conjuntos de equivalencias para la compuerta NOR mostrada son:

$$\{ a \text{ s-a-1, } b \text{ s-a-1, } c \text{ s-a-0} \}$$

$$\{ a \text{ s-a-0, } c \text{ s-a-1} \}$$

$$\{ b \text{ s-a-0, } c \text{ s-a-1} \}$$

Y los conjuntos de equivalencias para la compuerta NAND son:

$$\{ a \text{ s-a-0, } b \text{ s-a-0, } c \text{ s-a-1} \}$$

$$\{ a \text{ s-a-1, } c \text{ s-a-0} \}$$

$$\{ b \text{ s-a-1, } c \text{ s-a-0} \}$$

Debido a que cada compuerta tiene tres conjuntos de equivalencias, se necesitan solo tres pruebas para detectar estos problemas, o sea una prueba por cada conjunto de equivalencias.

En general, una compuerta de **m** entradas puede tener **(m + 2)** fallas lógicamente distintas. Sin embargo, solo deben chequearse **(m + 1)** conjuntos equivalentes.

Fallas temporales.

Como dijimos anteriormente, un error es la manifestación de una falla. Una falla temporal es aquella que es intermitente o un error transitorio. Los errores transitorios son la mayor fuente de fallas en los chips VLSI. No se pueden reparar porque no hay daños físicos en el hardware. La tecnología submicrónica ha permitido empaquetar millones de transistores en un chip VLSI reduciendo las dimensiones de los mismos. Aparejado a la disminución del tamaño también se reduce sus **márgenes de ruido**. Como resultado son más vulnerables al ruido, el cross-talk, etc. Lo que resulta en errores transitorios. Sumado a esto, son más vulnerables a la radiación terrestre.

Este tipo de fallas intermitentes son recurrentes, ya que aparecen cada cierto periodo de tiempo. Tales fallas pueden ocurrir por conexiones deficientes, componentes parcialmente defectuosos o errores de diseño. Algunas fallas transitorias, debido al envejecimiento de los componentes, pueden transformarse en permanentes. Las fallas intermitentes también ocurren debido a condiciones ambientales, como ser temperatura, humedad o vibraciones.

La probabilidad de dichas fallas también está influenciada por cómo está protegido el hardware respecto de su entorno, por ejemplo, con blindajes, refrigeración filtros, etc.

Una falla intermitente solo causa problemas en un circuito si está activa. Si está inactiva, el circuito funciona normalmente.

Como las fallas intermitentes son aleatorias, solo se pueden modelizar usando métodos matemáticas probabilísticos.

