Revista de la
Universidad Politécnica
de Puerto Rico

Publicado semestralmente por la Universidad Politécnica de Puerto Rico para difundir los hallazgos de la investigación científica que en ella se hace.

VOL. 2 NUM. 1

Junio 1992

Simulación digital para verificación de diseño y sus aplicaciones en el desarrollo de pruebas automatizadas

Rafael A. Figueroa
Rodolfo E. Jordán
Candidatos a graduación

Sinopsis

En este artículo se presenta el simulador digital como la herramienta para el análisis computadorizado de circuitos de lógica digital. Se discute el comportamiento de los circuitos digitales, el desarrollo de vectores de pruebas para equipo automatizado, el principio de fallas para probar circuitos digitales y el uso de modelos para simular la lógica digital. Se menciona especialmente el simulador de lógica excitado por eventos y se detalla el procedimiento para implantar un simulador de lógica digital.

Abstract

This paper presents the digital simulator as the tool for the computerized analysis of digital logic circuits. Among the topics discussed in the study are the behaviour of digital circuits, the development of vectors to test automated equipment, the fault principle to test digital circuits and the use of models to simulate the digital logic. The logic simulator driven by events is studied in detail and some advices on how to implant a logical simulator are given.

Introducción

La necesidad de integrar los procesos de diseño y fabricación de los productos basados en circuitos lógicos ha impulsado el uso del simulador digital en aplicaciones para verificar el diseño y el desarrollo de pruebas para equipos automatizados. Con el avance de la tecnología y el énfasis en la integración, cada vez aparecen circuitos digitales de mayor complejidad. Debido a la gran cantidad de compuertas lógicas que estos circuitos digitales poseen, se requiere mayor planificación para diseñarlos, analizarlos y probarlos.

Se puede decir que una prueba es un ejercicio diseñado para confirmar o rechazar una hipótesis o para distinguir entre dos o más hipótesis. Cuando el dispositivo que se somete a prueba es un elemento lógico digital, el estímulo aplicado a sus entradas se conoce como patrón o vector de prueba. Las respuestas a las pruebas se observan normalmente en las compuertas de salida, aunque algunas configuraciones permiten cierto grado de visibilidad en las etapas intermedias. La evaluación de las respuestas se logra comparando los resultados obtenidos contra los resultados esperados.

Si la respuesta obtenida difiere de la esperada, entonces se dice que ha ocurrido un error o una falla. Debido a que un circuito digital es una fuente potencial de un número significativo de fallas, la respuesta errónea es sólo el primer paso. Es necesario realizar pruebas adicionales para determinar cuál de todas las fallas es la causante del error. Este procedimiento se conoce como diagnóstico. Por otro lado, si el dispositivo responde favorablemente al estímulo aplicado, no necesariamente implica que el dispositivo esté libre de errores, sino que el circuito no contiene alguno de los errores que la prueba pretende detectar.

Los vectores de verificación de diseño deben ser suficientes como para confirmar que el diseño satisface el comportamiento expresado en las especificaciones del producto. El desarrollo de estímulos efectivos para las

pruebas es un proceso altamente iterativo, en el cual las discrepancias entre los resultados y lo esperado señalan la necesidad de la creación de vectores adicionales.

Por lo tanto, la etapa de verificación del diseño ocurre bajo el marco o escenario de las pruebas. De esta forma, antes de que el diseño pase a la etapa de producción, el ingeniero de diseño realiza una labor experimental intensa y verifica que el circuito funcione bajo las especificaciones del diseño, tomando en consideración las estadísticas de retraso de cada uno de los componentes del circuito. En el pasado se realizaban pruebas similares en los bancos de trabajo con prototipos del circuito producidos a partir del diagrama esquemático. Estas pruebas consumían largas horas de tediosa labor. Al presente el ingeniero de diseño o pruebas usa un simulador digital y puede generar los vectores de estímulo a su circuito, a la vez que observa la funcionalidad del circuito como respuesta a un estímulo aplicado ("software breadboarding"). Este tipo de análisis es el primer paso hacia el proceso de verificación de diseño automatizado.

Consideraciones de tiempo

El comportamiento de los circuitos digitales se caracteriza por su función lógica, la cual puede expresarse en términos del álgebra booleana y por la relación de tiempo entre sus entradas y salidas. Si el circuito es estrictamente combinacional, la relación entre la entrada y la salida está dada por el tiempo de propagación acumulativo del paso de la señal o el tiempo de retraso de la salida del circuito con respecto a la entrada que ocasiona un cambio de estado. Este tiempo de retraso puede explicarse observando el comportamiento de los portadores de carga minoritarios en las uniones "p-n" de los dispositivos semiconductores, cuyas densidades están dadas por la siguiente ecuación:

$$p_n(0) = p_{no} \exp^{(\frac{V}{V_T})} \qquad (1)$$

en la cual:

V_T = T/11,600
T = temperatura (Kelvin)
V = voltaje en la unión
p_{no} = concentración de portadores minoritarios de carga distantes de la unión, para material tipo-n y X >> 0.
$p_n(0)$ = concentración de portadores minoritarios de carga en la unión, para material tipo-n y X = 0

El tiempo de propagación, según se ilustra a continuación, es la media del tiempo de carga $t_{pd}(1)$ y el tiempo de descarga $t_{pd}(0)$ de la unión "p-n". En la expresión, 1 y 0 representan los estados discretos del voltaje aplicado a la unión "p-n" de saturación y corte ("cutoff"), respectivamente:

$$t_{p_d} = \frac{[t_{p_d}(0) + t_{p_d}(1)]}{2} \qquad (2)$$

La descripción del comportamiento de los circuitos lógicos es, por lo tanto, un problema de dos dimensiones: tiempo y estado (nivel lógico discreto). En el caso de los circuitos puramente combinacionales, el no tomar en cuenta la dimensión del tiempo no afecta la respuesta en el campo del estado si el resultado se observa luego que los estados transitorios del circuito se hayan estabilizado. En los circuitos secuenciales la situación es completamente diferente, ya que los circuitos secuenciales dependen críticamente del tiempo. El comportamiento correcto de estos circuitos depende de que los elementos de las memorias se actualicen con los valores correctos y que la señales de control aparezcan en el orden de tiempo adecuado.

Hay dos aplicaciones de circuitos secuenciales: los sensitivos a niveles lógicos y los disparados por transición ("edge-triggered"). Un circuito secuencial sensitivo a niveles lógicos responde a los niveles de "0" ó "1" de la señal del reloj, mientras que un circuito excitado por transición responde al filo ascendente o descendente de la señal del reloj o la base del tiempo.

A los dispositivos secuenciales se les especifican dos parámetros claves: el tiempo de preparación ("set-up time") y el tiempo de aguante ("hold time"). El tiempo de preparación se refiere al intérvalo durante el cual una señal debe mantenerse estable en el terminal de entrada antes de una transición activa de la señal de la base del tiempo. El tiempo de aguante se refiere al intérvalo durante el cual una señal debe mantenerse estable en un terminal de entrada después de haber ocurrido una transición activa de la señal de la base del tiempo. Es fácil notar que los circuitos secuenciales representan un reto adicional en la consideración del tiempo. No sólo son sensitivos a las fallas del estado lógico, sino a las fallas que afectan su ejecución como resultado de la velocidad de propagación y el tiempo relativo entre sus señales de entrada. La violación de cualquiera de estas especificaciones se debe detectar con la ayuda de un simulador digital y el diseño debe corregirse en la etapa de verificación.

Métodos de pruebas

Al desarrollar una prueba es importante crear o generar estímulos o vectores de prueba efectivos. Los primeros acercamientos en este aspecto cubrían desde la aplicación de todas las posibles combinaciones en las entradas del dispositivo bajo prueba hasta la aplicación de estímulos destinados a verificar solamente las funciones específicas del circuito. Por supuesto, la aplicación de 2^n vectores de pruebas a un dispositivo de **n** entradas era efectivo mientras **n** fuese pequeño. Sin embargo, puesto que el número de vectores crece exponencialmente, ya que se duplica por cada entrada del circuito, este método no es efectivo para dispositivos con un número considerable de entradas.

No obstante, el procedimiento de pruebas más efectivo fue el originado por R.D. Eldred [1]. Este método enfatiza la creación de pruebas para detectar fallas específicas. Bajo este concepto sólo las fallas que ocurren con más frecuencia se modelan. El modelo de fallas es un modelo computadorizado del circuito que se modifica para adaptarse a las premisas o conjeturas de los posibles defectos reales. Para probar un circuito digital usando el principio de fallas es requisito entender los tipos de errores que pudieran acontecer y sus consecuencias.

En el circuito que presenta la figura 1 se presume que R_4 está en corto circuito, entonces la base de Q_3 se mantiene a cero voltios y Q_3 no conduce. Q_4 no puede conducir, pues su base no puede hacerse más positiva que su terminal emisor. Sin embargo, Q_5 sí puede conducir, dependiendo del voltaje aplicado a su emisor por Q_6 que depende de la entrada Z.

Cuando la entrada Z tiene un valor lógico de 1, el voltaje positivo en la base de Q_6 lo hace conducir hasta saturación, con el efecto de aparentar un corto circuito del colector o del emisor. Bajo esta condición la base de Q_5 es más positiva que su emisor, lo que causa la conducción de Q_5; la salida F viene a ser 0 lógico. Si Z es 0, tanto Q_5 como Q_4 estarían en estado de no conducción y la salida F sería un 1 lógico. Vemos que como resultado de la falla, la salida en F es el complemento de Z y es totalmente independiente de las entradas X_1, X_2, Y_1 y Y_2.

---

[1] Eldred, R. D., "Test Routines Based on Symbolic Logic Statements, *J. ACM,* Vol. 6, Núm. 1, Enero, páginas 33-36.

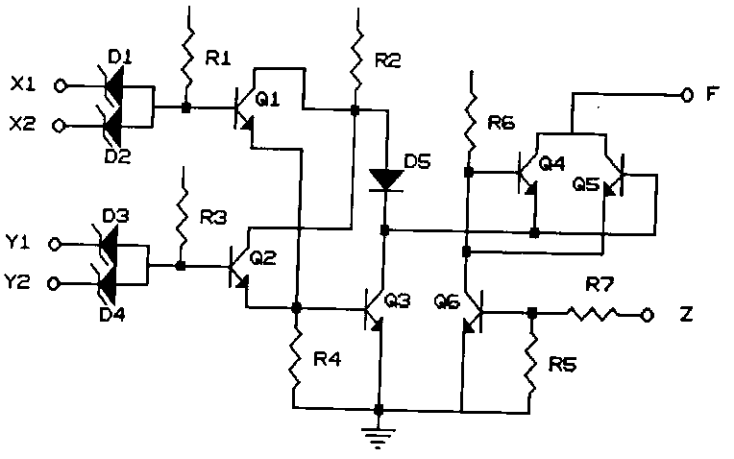

Figura 1. Ejemplo de un circuito para análisis

Una vez se sabe el comportamiento del circuito cuando la falla está presente, se procede a determinar cómo derivar una prueba que certifique que la falla está presente usando el punto F como el único punto de observación en el circuito. Esta restricción significa que la única manera de detectar la falla es generando un estímulo para el cual la respuesta sea una función de la ausencia o la presencia de la falla. La respuesta de un circuito defectuoso es lo opuesto a la respuesta del circuito libre de falla. Por consiguiente, se comienza investigando el estímulo necesario para lograr una condición contraria a la condición de defecto en el área de la falla. En este caso se requiere definir el estímulo necesario para que el estado de la base de Q_3 sea 1, presumiendo una condición de no falla. Este propósito se logra condicionando las entradas X_1 y X_2 o Y_1 y Y_2 a un 1 lógico. Cuando esto sucede, la salida F asume el valor de Z si no hay fallas y el negado de Z si la falla está presente.

La mayor parte de las fallas en los componentes se pueden representar por entradas o salidas de compuertas lógicas que se atan o permanecen indefinidamente en un estado lógico determinado, 1 ("stuck at 1") ó 0 ("stuck at 0"). El modelo de fallas SA-X, en el cual $X \in \{0,1\}$ se ha convertido en el modelo universal y tiene la ventaja de permitir la enumeración de las fallas. Para una compuerta lógica de n entradas es posible asignar un número determinado de fallas y definir específicamente las fallas y su efecto en el

comportamiento de la compuerta.

El modelo de fallas SA-X se basa en la premisa de fallas simples y fallas por estados lógicos permanentes (SA-0 y SA-1) en las compuertas lógicas básicas. Cuando se lleva este concepto a la práctica, lo que corresponde al modelo de fallas para circuitos, entonces el modelo de fallas es la combinación de los errores de todos los modelos primitivos que componen el circuito. Puesto que para el propósito de este análisis no hay acceso inmediato a todas las compuertas lógicas, es indispensable justificar las condiciones necesarias en las entradas primarias del circuito. De esta forma se condicionan las entradas de la compuerta bajo prueba y se hacen visibles los efectos de una falla en una salida observable por el sistema de pruebas.

Para lograr esta finalidad se establece un paso sensible, originado en la falla hasta una salida observable por el sistema de pruebas. El paso sensibilizado de una falla f es el paso de la señal que se origina en la falla f, cuyo valor a lo largo del paso es funcionalmente dependiente de la presencia o la ausencia de la falla. Si el paso sensibilizado termina en un nodo observable por el equipo de prueba, entonces la falla es detectable. El proceso de crear un paso sensibilizado se conoce como propagación.

En 1966 J.P. Roth[2] introdujo un método para mecanizar el proceso de generación de vectores de pruebas garantizando el máximo de eficiencia en las pruebas a los circuitos. Este método usa la técnica de propagación del paso de sensibilidad a través de todos los pasos posibles y se conoce como el algoritmo-$D_{iscrepancia}$. Este algoritmo depende del uso de cubos D, los cuales son esencialmente tablas de comparación del efecto de las fallas a través de las compuertas consideradas.

El algoritmo-$D_{iscrepancia}$ usa los arreglos de las tablas del estado, las cuales se manipulan con operaciones que las separan en cubiertas llamadas cubos (o cubos primos), dependiendo del estado de la función de salida. Es decir, las condiciones que producen que el estado de la función del sistema sea 1 componen el cubo primo 1-punto y las que ocasionan 0 en la salida componen el cubo primo 0-punto. Cada elemento del cubo, cuya expresión incluye el estado de las entradas y las salidas del sistema, se conoce como un

---

[2] Roth, J. P., 1966, "Diagnosis on Automatic Failures: A Calculus and a Method, *IBM J. Res. Dev.*, Vol. 10, Núm. 4, July, páginas 268-271.

vértice. La intersección de cubos singulares es el más pequeño cubo singular que contiene todos los vértices comunes entre sí. El algoritmo incluye una tabla de intersecciones y un procedimiento de manejo que halla los vectores de prueba que exponen, propagan y justifican una falla dada a través del paso de propagación hacia la salida observable por el sistema. Una vez se ejecuta, el resultado es un vector de prueba capaz de hacer visible una falla simple de SA-X, donde X puede asumir los valores discretos de {0,1}. La habilidad para perseguir una falla a través de todos los pasos de propagación posibles asegura la consecución de un patrón de prueba para la falla, si este patrón existiera.

Simulación de la lógica digital

La simulación es un proceso imitativo que se usa para estudiar las relaciones entre los parámetros que interaccionan en un sistema. En ciertos casos la simulación puede señalar los errores de un diseño que causarían que el circuito fuera inoperable o inservible. También permite la optimización de un diseño para máxima ejecución y economía en su operación o construcción. Hay casos en los cuales el sistema puede ser tan complejo que la simulación es la única manera de estudiar y controlar las variables del sistema que afectan el diseño. Por consiguiente, la simulación digital provee un medio para el análisis computadorizado de los circuitos de lógica y puede ser la base para la automatización de los procesos de diseño, manufactura y pruebas.

Para poder imitar el comportamiento de un sistema la simulación emplea modelos. Un modelo es una réplica imperfecta del comportamiento de circuitos lógicos digitales en términos de las variables del tiempo y el estado. El modelo debe ser tan exacto como sea posible para imitar el comportamiento de las variables de interés. Mientras más detallado y exacto el modelo, más detallado y exacto es el resultado de la simulación, lo que envuelve un compromiso en la necesidad de recursos computacionales.

Hay una vasta demanda por el uso de la simulación para la verificación de diseño. Por mucho tiempo los diseñadores de lógica han dependido de la construcción de prototipos de los sistemas diseñados. La aplicación de la simulación como herramienta en el análisis de fallas se ha popularizado por la revolución de la automatización de pruebas. Algunas ventajas que provee el método de la simulación por computadora son:

- les permite a los diseñadores evaluar simultáneamente diferentes partes del diseño, de manera sencilla y con un mínimo de tiempo de espera

- permite la evaluación de una pieza antes de fabricarla y facilita el estudio de los efectos de los parámetros de tiempos estadísticos (la medida de tales efectos es simple cuando se compara con los requisitos de la medida del mundo real)

- permite ver dentro de los circuitos integrados (ICs) y automatizar procesos de fabricación y pruebas.

Simulador excitado por eventos

El simulador lógico excitado por eventos toma ventaja del hecho de que no toda actividad de cambio en estímulo causa un cambio de estado en la salida de un elemento. En otras palabras, no toda actividad se propaga a través de un circuito. De hecho, la verdad es que la actividad dentro de un circuito es muchas veces mínima y puede terminar abruptamente. La estrategia en este caso es la de simular solamente los elementos que están envueltos en los cambios de estado en el paso de propagación de la señal. Esto requiere que el simulador opere en unas tablas de información que incluyen una lista de los elementos del circuito, su función, y sus interconexiones. La lista de interconexiones incluye una tabla de la relación de entradas-salidas para cada dispositivo del circuito. Estas relaciones se usan para encadenar los elementos con el propósito de pasar los resultados de la simulación entre los elementos y facilitar la programación ("scheduling") de los eventos a simularse.

Cuando ocurre un cambio en la señal de una entrada primaria o en la salida de cualquier elemento del circuito, se dice que ha ocurrido un evento en el nodo manejado por la entrada primaria o elemento. Cuando el evento ocurre en un nodo, entonces se simulan todos los elementos manejados por dicho nodo. Si el cambio en la señal en la entrada de un dispositivo no ocasiona actividad en la salida, la simulación a través de este paso de propagación se interrumpe.

Hay tres versiones del simulador excitado por eventos:

a. Simulador de retraso cero
   Simula la función lógica del elemento. No toma en cuenta los tiempos de retraso entre los elementos de lógica.

b. Simulador de retraso por unidad
   Similar al anterior, excepto que analiza los tiempos de retraso a base de un retraso fijo o unitario.

c. Simulador de retraso nominal

Asigna valores de retrasos a los elementos lógicos basados en las especificaciones del fabricante o mediante mediciones con instrumentos de precisión.

El simulador de retraso nominal puede hacer una simulación precisa a costa del tiempo computacional. La contraparte de éste, el simulador de retraso cero, usualmente corre a mayor velocidad pero sin la información del tiempo cuando los eventos ocurren. El simulador de retraso unitario descansa entre ambos, pero los mecanismos para la programación de los eventos en el tiempo futuro son más sencillos que en el caso de la simulación de la base del tiempo nominal. Entre todas las versiones, el simulador de retraso cero constituye el peor caso porque no toma en cuenta la dimensión del tiempo y colapsa todas las computaciones a tiempo cero.

La simulación de retraso nominal hace uso de los retrasos inherentes de los elementos lógicos. Para programar la secuencia en que los elementos lógicos se simulan se usa una lista lineal encadenada ("linear linked list"). Esto es necesario debido a que los eventos tienen que procesarse en el orden en que aparecen.

Los eventos que se producen como resultado de la actividad y que no se encuentren en la secuencia de procesamiento tienen que insertarse al tiempo en que dicho evento ocurre. Para este propósito se usa una lista lineal encadenada que permita la creación, búsqueda, adición, inserción y eliminación de los elementos, los cuales, en este caso, son estructuras de eventos. Los elementos de la lista son estructuras conocidas como noticias de eventos y se usan para describir la actividad que debe realizarse y el tiempo preciso en que se debe ejecutar.

Cuando se va a simular un evento, primeramente es necesario encontrar la posición cronológica del evento en la lista encadenada. Si la actividad en la entrada de una compuerta produce un cambio en la salida, entonces el evento se programa o se planifica para el futuro a un tiempo $t_0 + t_p$, en el que t_p representa el tiempo de propagación de la compuerta. El paso de la simulación continúa a medida que la actividad se propaga a través del circuito mientras los eventos se programan al tiempo apropiado dependiendo de los tiempos de retraso de las compuertas.

Concurrentemente, el simulador usa la celda de descripción para almacenar información perteneciente al circuito: interconexiones, función lógica, estado presente y condición del marcador de actividad. Cada celda

corresponde a un elemento del circuito que se simula y los nodos que interconectan los elementos del circuito se representan en el modelo por listas encadenadas que hacen su camino a través de las celdas de descripción (figuras 2,3 y 4).

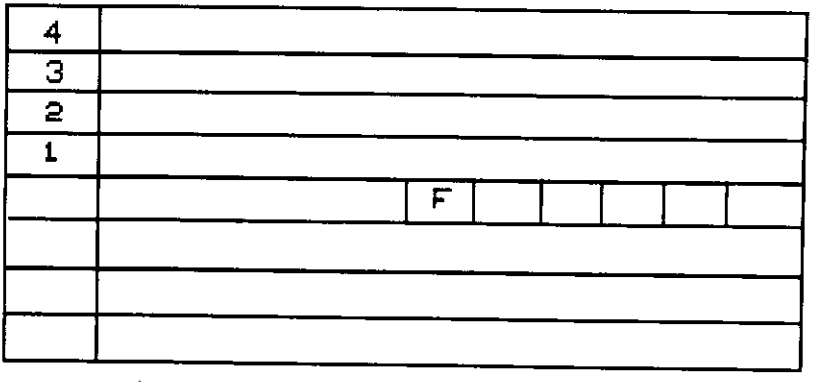

Figura 2. Celda de descripción

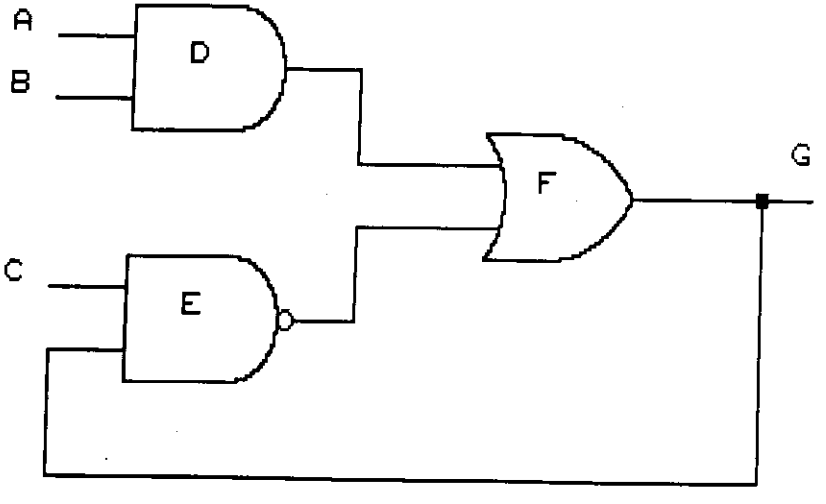

Figura 3. Ejemplo de celdas descriptivas

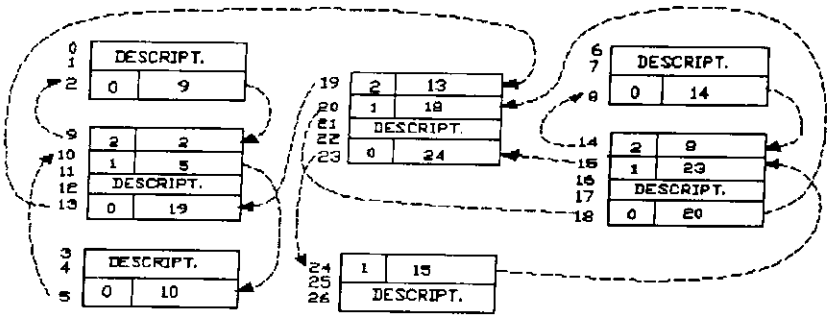

Figura 4. Celdas descriptivas del circuito anterior

El uso de los simuladores de lógica se prolifera en un mercado competitivo que crece tecnológicamente en forma desenfrenada. El adelanto tecnológico trae consigo la automatización de los procesos de diseño y pruebas en el área de la fabricación. Los equipos automáticos de pruebas cumplen con la necesidad de reducir los costos de las pruebas, lo que mejora la productividad y la calidad del producto.

El simulador de lógica digital sirve de plataforma para la automatización del proceso de verificación del diseño. El simulador de fallas analiza la calidad del estímulo o vectores de pruebas aplicados al circuito bajo análisis. El simulador de lógica predice el comportamiento del circuito ante el estímulo que el ingeniero de diseño usa para las pruebas y advierte de los peligros de las violaciones del tiempo y de las características de carga de los dispositivos.

Generalmente el ingeniero de diseño crea un sinnúmero de vectores de pruebas para descubrir la mayoría de las posibles fallas y violaciones de las especificaciones de función, tiempo y carga. Por otro lado, a nivel de fabricación al ingeniero de pruebas sólo le interesan los vectores que sirvan para descubrir fallas de estado funcional, presumiendo que el diseño está libre de fallas de propagación o de carga, verificables por el proceso de simulación nominal. Por lo tanto, aunque el uso del simulador de lógica digital es posible en ambos casos, los métodos de pruebas son muy diferentes y particulares ante la necesidad de las fallas a considerar. Sin embargo, es posible la integración de ambos métodos en el sentido de que el ingeniero de pruebas puede utilizar los vectores que descubran las fallas del estado entre los vectores de verificación de diseño.

Otro detalle que se puede notar es la posibilidad de emigrar del ambiente de simulación a equipo para pruebas automáticas (ATE, por sus siglas en inglés). En la mayoría de los casos, esta emigración es muy sencilla ya que estos equipos entienden el lenguaje del tiempo y el estado que usa el simulador. Con raras excepciones, las especificaciones del estímulo para controlar el equipo del subsistema digital de pruebas consiste de comandos de estados con restricciones de tiempo específicas o programables (fases), limitados solamente por las características del equipo. Su contraparte es la definición de la respuesta, en la que se define un evento del estado que ocurre dentro de una ventana de tiempo. Para lograr la emigración de la simulación a ATE, se requiere que los métodos de procesamiento posterior y simulación tomen en cuenta las limitaciones del sistema de pruebas en cuanto a sus capacidades de velocidad de ejecución, fases y ventanas de tiempo.

La manera más sencilla de implantar un simulador de lógica sugiere el uso de la programación sobre un sistema de base de datos capaz de crear y mantener los índices de los récores en orden de clave, en la que la clave es el campo de la estructura que establece el orden de los récores. Para un simulador de lógica excitado por eventos, las estructuras están relacionadas con la cadena de eventos y la celda de descripción. Este método de análisis de los eventos parece ser el más simple y eficaz. Una vez se sabe operarlo, el programador o ingeniero puede mejorar la eficiencia del método. Una forma de lograrlo es controlando las señales de la base del tiempo. Una señal de la base del tiempo corriendo libremente significa un sinnúmero de eventos ocurriendo al azar. Por lo tanto, para una simulación eficiente se debe observar extremo cuidado sobre las señales de control y la base del tiempo.

El proceso de la creación de modelos de comportamiento lógico es el más complicado y crítico. Estos modelos describen el comportamiento y las características de los componentes del circuito. Los componentes de alta integración (LSI, VLSI) con capacidades de programación presentan estructuras secuenciales internas muy complejas que deben modelarse, lo cual representa el mayor reto en este aspecto. En el caso anterior, el comportamiento funcional no se puede modelar en una expresión de lógica booleana, más bien requiere un subprograma que pueda tomar decisiones basándose en los estados de los registros programables. En este caso, cada uno de estos dispositivos requiere un subprograma de esta índole, a lo cual hay que añadir las especificaciones del tiempo de propagación, preparación y aguante.