



10  
28j

**UNIVERSIDAD NACIONAL  
AUTONOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES  
CUAUTITLAN**

**SIMULACION INTERACTIVA DE CIRCUITOS  
DIGITALES AUXILIADA POR COMPUTADORA**

**T E S I S**  
**QUE PARA OBTENER EL TITULO DE  
INGENIERO MECANICO ELECTRICISTA  
PRESENTAN  
ROGELIO CISNEROS GUADARRAMA  
OSCAR DAVILA MONROY**



Universidad Nacional  
Autónoma de México



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## I N D I C E

---

INDICE . . . . .	1
PROLOGO . . . . .	5

### I. ANTECEDENTES Y GENERALIDADES.

---

I.1. Introducción . . . . .	7
I.2. Historia de la Simulación de Sistemas . . . . .	9
I.3. Definiciones de Simulación . . . . .	11
I.4. Ejemplos de Simuladores . . . . .	12
I.5. Ventajas de la Simulación . . . . .	13
I.6. Elementos que intervienen en un Proceso de Simulación. . . . .	14
I.7. Sistemas . . . . .	15
I.8. Modelos . . . . .	15
I.8.1. Clasificación de Modelos . . . . .	16
I.8.2. Consideraciones para formular Modelos . . . . .	17
I.9. Validación y Verificación . . . . .	18
I.9.1. Validación . . . . .	18
I.9.2. Verificación . . . . .	19
I.10. Pasos para Realizar un Sistema de Simulación . . . . .	19

## II. SIMULACION DE SISTEMAS DIGITALES POR COMPUTADORA.

---

II.1. Introducción . . . . .	22
II.2. Modelado de un Simulador de Sistemas Digitales (SSD) . . . . .	23
II.3. Estructura Básica de un SSD . . . . .	24
II.3.1. Módulo Preprocesador . . . . .	26
II.3.2. Módulo de Estructura . . . . .	28
II.3.3. Módulo Simulador . . . . .	28
II.3.4. Módulo de Salida . . . . .	31
II.4. Retardo . . . . .	31
II.4.1. Retardo de Transporte . . . . .	32
II.4.2. Retardo por Ambigüedad . . . . .	32
II.4.3. Retardo de Subida y Bajada . . . . .	32
II.4.4. Retardo Inercial . . . . .	33
II.5. Sistemas de Lógica Multivalor . . . . .	34
II.5.1. Lógica de Tres Estados . . . . .	35
II.5.2. Lógica de Cinco Estados . . . . .	35
II.5.3. Lógica de Ocho Estados . . . . .	36
II.6. Tipos Básicos de SSD's . . . . .	38
II.6.1. Simulación por Código Compilado . . . . .	38
II.6.2. Simulación por Tabla Estructurada . . . . .	41
II.7. Evaluación de Elementos . . . . .	47
II.7.1. Evaluación por Subrutina . . . . .	48
II.7.2. Evaluación por Conteo de Entradas . . . . .	48
II.7.3. Evaluación por Tablas de Verdad . . . . .	49
II.7.4. Evaluación por Tablas Zoom . . . . .	49

### III. DISEÑO DEL SIMULADOR INTERACTIVO (SIM/CD).

---

III.1. Introducción . . . . .	51
III.2. Consideraciones de Diseño . . . . .	53
III.2.1. Consideraciones del Sistema a Simular . . . . .	54
III.2.2. Consideraciones del Modelo Simulador . . . . .	56
III.2.3. Consideraciones de la Computadora . . . . .	61
III.3. Algoritmo de SIM/CD . . . . .	63
III.4. Módulo Preprocesador . . . . .	66
III.4.1. Inicialización de Variables . . . . .	67
III.4.2. Lectura de Descripción Lógica . . . . .	67
III.4.3. Lectura de Estimulo Externo . . . . .	68
III.4.4. Lectura de Puntos a Monitorear . . . . .	69
III.4.5. Chequeo de Sintaxis . . . . .	69
III.5. Módulo de Estructura . . . . .	70
III.5.1. Descripción de la Tabla de Elementos . . . . .	70
III.5.2. Interrelación de los datos . . . . .	73
III.6. Módulo Simulador . . . . .	74
III.7. Módulo de Salida . . . . .	78

### IV. EXTENSION A LAS CAPACIDADES DEL SIMULADOR.

---

IV.1. Introducción . . . . .	80
IV.2. Entrada de Datos utilizando Gráficos . . . . .	81
IV.3. Pantallas de Captura . . . . .	84
IV.4. Trazo Selectivo . . . . .	86
IV.5. Macroelementos . . . . .	88

**V. MANUAL DE USUARIO.**

-----

V.1. Introducción . . . . .	92
V.2. Operación de SIM/CD Versión 1.0 . . . . .	93
V.2.1 Descripción Lógica del Circuito a Simular. . . . .	93
V.2.2 Descripción de Estímulos Externos . . . . .	96
V.2.3 Descripción de las Señales a Monitorear. . . . .	98
V.2.4 Proceso de Simulación . . . . .	99
V.2.5 Mensajes de Error . . . . .	102
V.3 Operación de SIM/CD Versión 2.0 . . . . .	103
V.3.1 Descripción de Elementos Lógicos . . . . .	104
V.3.2 Descripción de Estímulos Externos . . . . .	105
V.3.3 Descripción de Nodos a Monitorear . . . . .	106
V.4 Alternativas . . . . .	108
V.4.1 Cambio de Símbolos Impresos . . . . .	108
V.4.2 Almacenamiento de un Circuito dentro del Programa . . . . .	109
V.4.3 Aumento o Disminución de la Capacidad de SIM/CD . . . . .	110
CONCLUSIONES . . . . .	111
APENDICE A. LISTADO DEL PROGRAMA SIM/CD 1.0 . . . . .	113
APENDICE B. LISTADO DEL PROGRAMA SIM/CD 2.0 . . . . .	118
APENDICE C. EJEMPLOS . . . . .	130
REFERENCIAS . . . . .	154
BIBLIOGRAFIA . . . . .	155

## PROLOGO

-----

El presente trabajo trata sobre el diseño e implementación de un Simulador de Circuitos Digitales. El simulador construido consiste en un programa en lenguaje BASIC que puede ser utilizado tanto por instituciones educativas (universidades y escuelas técnicas) como por la industria. Debido a sus características interactivas, el SIMulador de Circuitos Digitales (SIM/CD) proporciona una herramienta invaluable al diseñador de circuitos, permitiéndole verificar la operación lógica de sus diseños en un tiempo muy reducido.

La Tesis consta de cinco Capítulos y tres Apéndices:

En el Capítulo I se da una breve reseña histórica de los simuladores y se definen los conceptos básicos como son Modelos, Sistemas reales, y Computadoras.

En el Capítulo II se presentan las bases teóricas de la Simulación de Circuitos Digitales: Tipos de simuladores, Partes constitutivas (Módulos), y Características consideradas (retardo, lógica multivalor, etc.).

En el Capítulo III se discuten las consideraciones de diseño y se describen cada uno de los Módulos del simulador, tal como fueron implementados en la versión básica del programa (SIM/CD 1.0).

En el Capítulo IV se describen algunas características adicionales tales como mejoras en la Captura de Datos utilizando

gráficos, Trazo Selectivo y Macroelementos, que fueron implementados en su mayoría en la versión avanzada del programa (SIM/CD 2.0).

El Capítulo V constituye un Manual de Usuario que contiene las indicaciones y los pasos necesarios para la correcta operación del simulador.

Finalmente, el Apéndice A contiene el listado fuente de la versión básica del programa; el Apéndice B contiene el listado fuente de la versión avanzada; y el Apéndice C proporciona algunos ejemplos de la utilización del simulador.

Es nuestro deseo que el presente trabajo no solamente sirva como una invaluable herramienta para estudiantes y diseñadores de circuitos, sino que constituya un punto de partida para futuros trabajos de Tesis.

RCG / ODM

Septiembre, 1987



ANTECEDENTES Y GENERALIDADES

I.1 INTRODUCCION.

Los diseñadores de circuitos de generaciones pasadas han visto la necesidad de construir prototipos de prueba para verificar el correcto funcionamiento de los circuitos. En los últimos años, la complejidad de los circuitos a diseñar se ha incrementado a un ritmo acelerado. Este incremento en complejidad es debido a los avances en tecnología así como a la necesidad de cumplir con un número cada vez mayor de especificaciones. Lo anterior ha provocado que los diseñadores utilicen las computadoras para satisfacer las necesidades de diseño y prueba.

Debido a la rápida evolución en la tecnología de los procesadores, es posible utilizar la computadora no sólo como un medio para simular un sistema particular sino también para modelar los efectos de variaciones estadísticas en los procesos de manufactura, para optimizar los parámetros en un diseño dado, así como para explorar varias alternativas de diseño y determinar la mejor solución cuando se tienen varios objetivos simultáneamente.

El término "Simulación de Sistemas" se refiere a la duplicación de las características de un modelo, de tal modo que

permita que un investigador estudie y trabaje con él. La representación matemática o simbólica de un sistema se llama modelo. Este método para resolver problemas no es nuevo. Los investigadores frecuentemente usan modelos o representaciones con el fin de aprender, probar ideas y predecir el efecto de cambios introducidos. Sin embargo las computadoras han hecho posible los estudios de métodos que anteriormente eran imprácticos. Aún en la simulación de sistemas medianos se requiere por lo general un número mayor de cálculos que los que se pueden hacer eficientemente a mano.

Este método se usa en las diferentes ciencias con el fin de simular los procesos físicos, tales como el funcionamiento de los controles y sistemas de un avión, el funcionamiento de un sistema digital, y en los negocios para simular los sistemas utilizados. La solución por simulación requiere una descripción cuantitativa del sistema y la programación del modelo en una computadora. El modelo permite reducir el tiempo real de operación, en tal forma que varios años de operación se pueden estudiar en unos cuantos minutos. O el proceso se puede hacer lento para estudiar determinadas áreas del problema en detalle. La simulación ofrece una base para las decisiones con respecto a un nuevo sistema sin las dificultades y los costos que acarrearía construir un prototipo.

## I.2 HISTORIA DE LA SIMULACION DE SISTEMAS.

---

La simulación de sistemas puede remontarse desde Lord Kelvin, quien construyó el primer analizador diferencial o computadora analógica a base de conectar algunos integradores mecánicos que habían sido inventados por su hermano James Thomson. La simulación con computadoras electrónicas analógicas tuvo su mayor apogeo durante la Segunda Guerra Mundial con el uso de circuitos electrónicos para estudiar fenómenos como dinámica de aviones, bombas guiadas y sistemas de control de proyectiles. La computadora digital posee algunas ventajas sobre la analógica en la simulación de sistemas. Evita la mayoría de los problemas inherentes en computación analógica: precisión limitada, la tendencia de los amplificadores a variar con el voltaje, y el hecho que el tamaño del sistema a ser simulado está limitado por el tamaño de la computadora analógica. No obstante las ventajas que las computadoras digitales ofrecen para la simulación de sistemas, no fueron muy utilizadas con este propósito por varios años debido a la gran dificultad de programar en ellas modelos de simulación. La ayuda llegó en forma de programas conocidos hoy como Programas Digitales de Simulación Analógica. Estos programas fueron diseñados para facilitar el uso de la computadora digital en estudios de simulación a base de permitir al usuario programar su modelo en una forma muy similar a como se hace en una máquina analógica.

El primero de estos programas fue producido por R.G. Selfridge para la IBM 701 en 1955. El primero que recibió mucha aceptación fue MIDAS (Modified Integration Digital Analog

Simulator). Fué desarrollado en la base aérea Wright-Patterson de los Estados Unidos en 1963 para la IBM 7090/7094 por H.E. Peterson y F.J. Sansom. Otro simulador digital-analógico que tuvo éxito fue PACTOLUS, programado para la IBM 1620 por Brennan y Sano en 1964, el cual trajo los beneficios de programas digitales de simulación analógica al usuario de computadoras pequeñas. PACTOLUS permitía al usuario modificar la configuración, parámetros y condiciones iniciales a base de usar un teclado y conmutadores de sensibilidad.

Los lenguajes de simulación pronto tomaron una nueva dirección. Fueron diseñados para aprovechar la capacidad de la computadora digital y para utilizar los poderosos sistemas de programación como FORTRAN y ALGOL que ya existían.

La tabla 1.1 da un breve resumen de los principales lenguajes de simulación.

Lenguaje	Desarrollado en:	Autores	Utilizado en Computadoras	Ref.
Gasp	U.S. Steel, Arizona	Kiviat y Pritsker	Cualquiera con compilador FORTRAN IV	[1]
GPSS	IBM	Gordon	IBM 7090, 7094, 7044, S 360/370; UNIVAC 1107/1108/1110; CDC serie 6000; Honeywell series 600 y 6000.	[2]
Simsript	RAND	Markowitz, Hausner, y Karr	IBM 7090, 7094, 7040, 7044, 360/370; CDC 3600, 3800, 6400, 6600, 7600; UNIVAC 474, 1107, 1108, 1110	[3]
Simula	Centro de Computo Noruego	Dahl y Nygaard	UNIVAC 1107, 1108, 1110; CDC 6400, 6600, 6700, 7600; Burroughs B5000, B6000 y B7000	[4]

TABLA 1.1 Principales Lenguajes de Simulación.

### I.3 DEFINICIONES DE SIMULACION.

Se han propuesto varias definiciones de simulación. John McLeod [5] dice: "Simulación es el desarrollo y uso de modelos para el estudio de ideas, sistemas y situaciones". En su libro, Gordon [6] dice "La Simulación de Sistemas es una técnica para resolver problemas a base de seguir los cambios a través del tiempo en un modelo dinámico de un sistema". Naylor [7], en su libro que tiene que ver primordialmente con el estudio de sistemas comerciales y económicos, aunque mucho de éste puede aplicarse a

sistemas generales, dice: "Simulación es una técnica numérica para conducir experimentos en una computadora digital, que incluye ciertos tipos de modelos lógicos y matemáticos que describen el comportamiento de un sistema económico o comercial (o de alguno de sus componentes) a través de periodos de tiempo real". Finalmente, una definición tomada de una publicación especializada [8] menciona que "La simulación es el proceso de diseñar y construir un modelo de un sistema real para conducir experimentos con éste, con el fin de entender las cualidades del sistema y evaluar las estrategias a seguir en la operación del mismo".

En el presente trabajo, se define la simulación digital como "el proceso de construir y ejercitar un modelo de un circuito digital en una computadora". Por ejercitar se entiende la evaluación de los valores de las señales en el circuito modelado para alguna secuencia de entradas aplicadas.

#### I.4 EJEMPLOS DE SIMULADORES.

---

Un ejemplo específico del uso de la computadora como herramienta podría ser el diseño de un circuito integrado VLSI (integración a muy alta escala). Se podría utilizar un programa de computadora para simular el proceso de fabricación, tomando en cuenta las variaciones en dicho proceso. Un simulador de circuitos digitales permitiría la verificación de las operaciones lógicas del diseño. Finalmente, un programa de layout facilitaría la colocación de los miles de transistores del circuito VLSI en

un chip.

Otro ejemplo de simulador es el Simulador de Combate de la Fuerza Aerea Norteamericana llamado TAC THUNDER. Este programa simula las posiciones americanas y soviéticas en una guerra convencional en Europa, considerando el aspecto aéreo y terrestre, así como el reabastecimiento. Este es un simulador por computadora escrito en aproximadamente 40,000 líneas de lenguaje SIMSCRIPT II.5 (lenguaje de propósitos especiales), lo que es equivalente a 100,000 líneas de FORTRAN.

En el momento de simular un combate, se pudo observar que en caso de un ataque soviético no había suficiente tiempo para activar el plan de defensa debido a que la alerta llegaba cuando el ataque era inminente. Como resultado, la estrategia de defensa aérea fue rediseñada.

#### I.5 VENTAJAS DE LA SIMULACION.

-----

Entre las ventajas de la simulación, se pueden citar las siguientes:

- Eliminar riesgos al no tener que llevar a cabo experimentos peligrosos para conocer el comportamiento de un sistema.
- Reducir los costos de diseño al suprimir la necesidad de construir un prototipo para verificar el funcionamiento del sistema.
- Eliminar tareas repetitivas y por lo tanto reducir las posibilidades de error humano.
- Disminuir el tiempo de verificación de funcionamiento de un sistema diseñado.

## I.6 ELEMENTOS QUE INTERVIENEN EN UN PROCESO DE SIMULACION.

En un proceso de Simulación por computadora intervienen tres elementos principales:

1. Sistema real
2. Modelo
3. Computadora

Estos elementos se vinculan como lo muestra la Fig. 1.1.

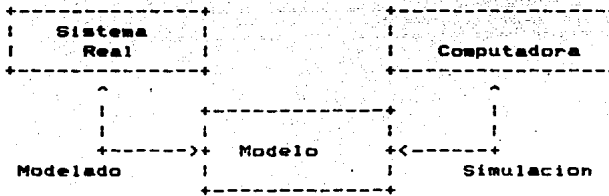


Fig. 1.1 Elementos Principales en la Simulación por Computadora.

Modelado es la relación entre el sistema real y el modelo.

Simulación relaciona la computadora con el modelo.

El sistema bajo estudio se representa con un modelo en una forma capaz de ejercitarse con una computadora. El modelo de simulación se ejercita en la computadora para obtener información sobre el sistema representado.



## **I.7 SISTEMAS.**

Por sistema se entiende una colección de entidades relacionadas, cada una de las cuales se caracteriza por atributos o características que pueden estar relacionados entre si.

Todo sistema consta de tres características: Tiene fronteras, existe dentro de un medio ambiente y tiene subsistemas.

El sistema puede ser:

- natural o artificial.
- existente o planeado para el futuro
- adaptable o inadaptable
- abierto o cerrado

## **I.8 MODELOS.**

Por modelo se entiende la abstracción de un sistema real con el proposito de control y/o predicción de comportamiento. El objetivo del modelo es permitir el cambio o la introducción de variables en el sistema modelado con el fin de observar como afectan éstas a otras variables del sistema o a la totalidad del mismo.

Para que un modelo sea útil debe de reunir dos atributos: realismo y simplicidad. Por un lado, el modelo ha de servir como una aproximación razonable al sistema real y debe incorporar la mayor parte de los aspectos importantes de éste; por otro, no es

conveniente que el modelo sea tan complejo que resulte imposible entenderlo o manipularlo.

#### I.8.1 Clasificación de Modelos:

-----

Los modelos se clasifican de acuerdo a distintos criterios:

##### a) Según el tipo de sistema:

-----

- Modelo de sistema continuo.

Las variables matemáticas que describen el estado del sistema varían continuamente en el tiempo.

- Modelo de sistema discreto.

El estado del sistema cambia únicamente de una manera discreta durante ciertos instantes de tiempo y entre esos instantes el estado del sistema está fijo.

##### b) Según el tipo de variables usadas:

-----

- Modelo determinístico.

En un modelo determinístico aparecen variables determinísticas, es decir, no-aleatorias.

- Modelo probabilístico.

En un modelo probabilístico o estocástico intervienen variables aleatorias.

c) Según la invariancia en el tiempo:

---

- Modelo invariante en el tiempo.

Un modelo es invariante en el tiempo si las reglas de interacción están especificadas completamente en términos de los valores que las variables pueden asumir.

- Modelo variante en el tiempo.

En un modelo variante en el tiempo, las reglas de interacción incluyen explícitamente al tiempo como argumento. Esto significa que los resultados serán diferentes en tiempos distintos.

d) Según el tipo de computadora:

---

- Modelo en Computadora Analógica
- Modelo en Computadora Digital
- Modelo en Computadora Híbrida

I.8.2 Consideraciones para formular modelos.

---

Al formular modelos para simulación por computadora, se deben de tomar en cuenta las siguientes consideraciones:

- Se debe saber cuantas variables se van a incluir en el modelo ya que existe un límite superior en el número de variables posibles a investigar en un sólo experimento de simulación, el tamaño de la computadora disponible para el investigador impondrá ciertas limitaciones relativas a este aspecto.

- Se debe considerar la complejidad de los modelos. La formulación de modelos debe producir descripciones o predicciones razonablemente exactas, referentes al comportamiento de un sistema y reducir a la vez el tiempo de computación y programación.
- Se debe considerar la validez o cantidad de realismo incorporado a los modelos. El modelo debe describir adecuadamente al sistema de interés, y proporcionar buenas predicciones de comportamientos futuros.
- Se debe considerar la finalidad de la simulación, es decir, si el simulador será utilizado para diseñar circuitos, para comprobarlos o bien para observar su comportamiento.

## I.9 VALIDACION Y VERIFICACION.

-----

Dos conceptos importantes en la simulación de sistemas son la validación y la verificación. A continuación se definen cada uno de ellos.

### I.9.1 Validación:

-----

Validez es una medida de la similitud entre el sistema real y el modelo. El modelo es válido replicativamente si concuerda con los datos obtenidos con anterioridad en el sistema real. Es válido en predicción si puede obtener datos correctos antes de ser adquiridos por el sistema real.

### I.9.2 Verificación:

---

El programa debe ser verificado para asegurar que es correcto, ya que no se deben confundir las fallas del programa, con las fallas del modelo.

Validación ----> Modelo  
Verificación ----> Programa

### I.10 PASOS PARA REALIZAR UN SISTEMA DE SIMULACION.

---

Los pasos necesarios para realizar un sistema de simulación se muestran en la Figura 1.2, y se describen a continuación:

1. Definir el problema. La formulación debe incluir las preguntas que se van a contestar, las hipótesis que se van a probar, los cálculos que se van a hacer, etc.
2. Contruir un modelo matemático. El modelo debe abstraer las características esenciales del sistema y hacer posible la predicción de estados futuros del sistema basado en varios estados iniciales.
3. Validación. Si el modelo no representa al sistema lo suficientemente bien para los propósitos del estudio, regresar al paso anterior.
4. Construir el Programa. Escribir un programa del modelo que permita introducir datos, fijar condiciones iniciales, corregir errores y generar soluciones.
5. Verificación. Si el programa no representa correctamente el modelo, regresar al paso 4.

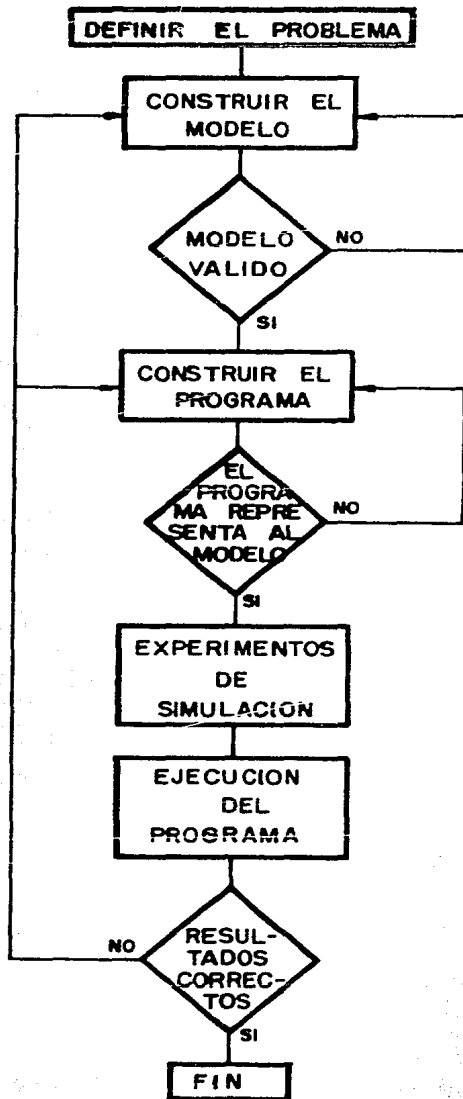


Fig. 1.2 Pasos para realizar un Sistema de Simulación.

6. Planear los experimentos de simulación. Formular un plan cuidadoso de los experimentos que se van a efectuar con el modelo para entregar la información requerida en el paso 1. El plan debe incluir los datos y condiciones iniciales que se van a usar y el tipo de resultados deseados.
7. Efectuar las corridas de simulación. Ejecutar el programa de acuerdo con el paso anterior para producir los resultados.
8. Analizar los resultados. Si los resultados obtenidos son los esperados, terminar. En caso contrario, regresar a los puntos 2 y 4 según convenga.

El desarrollo de un Sistema de Simulación requiere un análisis detallado del problema así como la realización de una serie de actividades bien definidas como se menciona en éste capítulo. No obstante, los beneficios que presentan los sistemas simuladores en la ingeniería (disminución de costos de diseño, rápida verificación de funcionamiento, etc.) justifican ampliamente el costo y tiempo de su realización.

SIMULACION DE SISTEMAS DIGITALES POR COMPUTADORA

II.1. INTRODUCCION.

Existen dos aplicaciones principales para un simulador de sistemas digitales. La primera de ellas es la evaluación de un nuevo diseño, donde al diseñador le interesa comprobar que la lógica utilizada es la correcta, el comportamiento en el tiempo, y las características de propagación de las señales utilizadas.

La segunda aplicación es en el análisis de fallas de los sistemas lógicos. En ésta, al diseñador le interesa saber que fallas pueden ser detectadas si se sugiere una "secuencia de prueba", o cual es el comportamiento del sistema frente a cierto tipo de fallas.

A continuación se resumen las aplicaciones anteriormente descritas:

1. Verificación del diseño Hardware

a. Verificación de la lógica

b. Análisis de tiempos

Retardos

Análisis de oscilación

c. Análisis de inicialización



## 2. Análisis de Fallas

- a. Tipos de fallas
- b. Análisis de tiempos bajo condiciones de falla
- c. Inicialización bajo condiciones de falla
- d. Evaluación de efectividad en los puntos de prueba

La utilidad de un sistema simulador de circuitos digitales puede ser evaluada con respecto a la gama de circuitos lógicos que puede manejar: compuertas (NAND, NOR, AND, OR, XOR, XNOR e inversor), flip-flops (latch, maestro-esclavo, etc), así como RAMs, ROMs, multivibradores, codificadores, decodificadores, etc.

La precisión de un simulador lógico depende de la fidelidad con que se representan las características de retardo en la propagación de señales.

## II.2 MODELADO DE UN SIMULADOR DE SISTEMAS DIGITALES (SSD).

---

Modelado es el proceso de describir un circuito en términos de las primitivas del sistema. El objetivo es construir un modelo, de tal forma que los resultados del simulador correspondan a los valores de las señales resultantes en el circuito real.

De acuerdo con lo expuesto en el Capítulo I, el modelo de un sistema digital tiene las siguientes características: es un modelo de sistema discreto, debido a que el estado del sistema

cambia solamente durante ciertos instantes de tiempo; es un modelo determinístico, porque no aparecen variables aleatorias; y es un modelo invariante en el tiempo ya que los resultados deben ser los mismos en tiempos distintos.

En la construcción del modelo intervienen cuatro factores principales: los tipos de elementos en los circuitos originales; las primitivas del sistema; el algoritmo básico; y el lenguaje de programación utilizado. En el Capítulo III (Diseño del Simulador) se describen estos factores para el simulador desarrollado.

### **II.3 ESTRUCTURA BASICA DE UN SIMULADOR DE SISTEMAS DIGITALES.**

---

La estructura de un SSD es la representada en la figura 2.1:

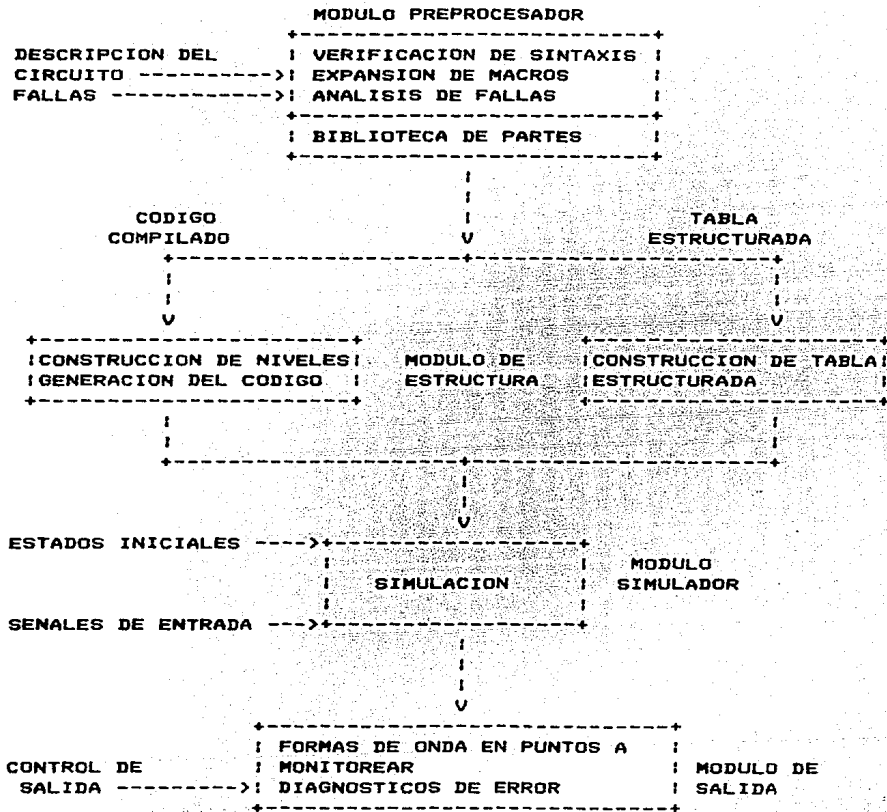


Fig 2.1 Estructura General de un Simulador de Sistemas Digitales

### II.3.1 Módulo Preprocesador.

-----

La función principal del módulo preprocesador es la de leer la descripción de los circuitos introducida por el usuario y reemplazar esta por su circuito equivalente en términos de las primitivas del sistema. Este proceso incluye la substitución de macro-elementos por sus primitivas correspondientes. El módulo preprocesador debe, asimismo, detectar errores de sintaxis en los datos de entrada.

La información requerida a la entrada de un simulador de sistemas digitales consiste en:

1. Descripción de los circuitos a ser simulados
2. Datos de entrada a simular
3. Valores iniciales de los elementos
4. Fallas a simular, si existen
5. Señales a monitorear

Existen dos métodos para indicar la topología (descripción de las interconexiones) del circuito: descripción orientada a nodos y descripción orientada al elemento.

La descripción orientada a nodos especifica las terminales de los elementos asociadas con una señal en particular.

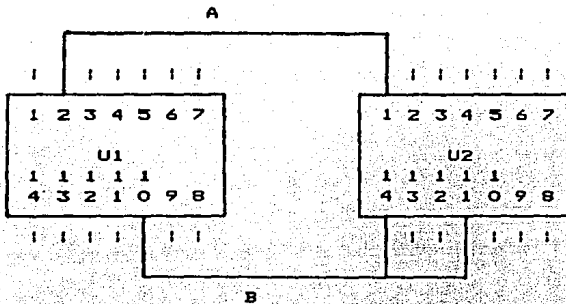
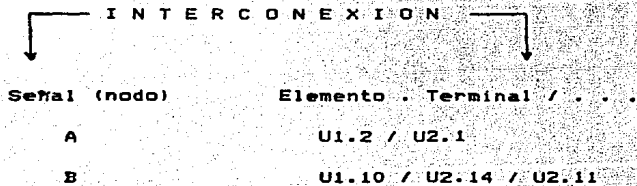
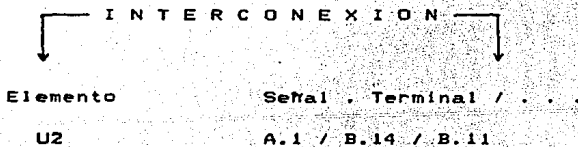


Fig 2.2 Ejemplo de Interconexión de un Circuito Digital.

Para ilustrar esto, se representan las conexiones del circuito de la Fig. 2.2 de la siguiente manera:



La descripción orientada al elemento especifica las señales asociadas con cada elemento. Así, el circuito anterior se representaría de la siguiente manera:



### II.3.2 Módulo de Estructura.

---

Existen dos formas típicas de representar un circuito durante la simulación. Los sistemas más antiguos utilizaban Código Compilado, es decir, el comportamiento del circuito a simular era representado por una secuencia de instrucciones de computadora. Los simuladores modernos operan en base a una Tabla Estructurada (arreglo), que describe la topología (interconexión de elementos) del circuito así como datos tales como tipo de elemento, retardo, etc.

La función del módulo de estructura es preparar las estructuras de datos requeridas internamente para lograr una simulación eficiente. Para los simuladores de Código Compilado este módulo debe producir una secuencia de instrucciones de máquina ejecutables, mientras que para un simulador de Tabla Estructurada, el módulo de estructura debe de producir una serie de tablas describiendo el circuito a simular.

### II.3.3 Módulo Simulador.

---

Un concepto de importancia en la simulación de sistemas digitales (SSD) es el Trazo Selectivo o Actividad Direccional. En la simulación de un sistema digital solamente un porcentaje reducido de las líneas cambian de valor (entre 2% y 10%). Debido a este hecho resulta ineficiente simular todos los elementos del sistema. La simulación por Trazo Selectivo considera lo anterior al efectuar solamente la simulación de los elementos donde ocurrió un evento (cambio de estado a la entrada).

Cuando ocurre un evento, cada uno de los elementos afectados se convierte en potencialmente activo. Al simular todos los elementos potencialmente activos se estarán simulando, en realidad, algunos elementos activos (que provocaron un evento) y otros inactivos. Al simular solamente los elementos potencialmente activos se obtiene una reducción apreciable en el tiempo de simulación.

El diagrama general de un módulo simulador se muestra en la figura 2.3.

El bloque A se refiere a la inicialización de los estados del circuito. Cuando se simula un circuito, es necesario especificar el valor inicial de los elementos que determinan su estado. A menudo éste estado inicial es desconocido. Por ejemplo, cuando un circuito es inicialmente energizado, el valor inicial de los flip-flops es impredecible. La manera mas común de resolver este problema es fijar el estado inicial de cada elemento a un valor indeterminado. Así, al aplicar una secuencia de pulsos, el simulador debe de calcular el estado final correcto del circuito.

En el bloque B un nuevo vector de entrada es leído. Solamente los cambios en las líneas deben ser procesados ya que solamente ellos definen actividad. En el bloque C se realiza un análisis de oscilación. En el bloque D se actualizan todos los valores de los circuitos que han cambiado debido a una nueva entrada.

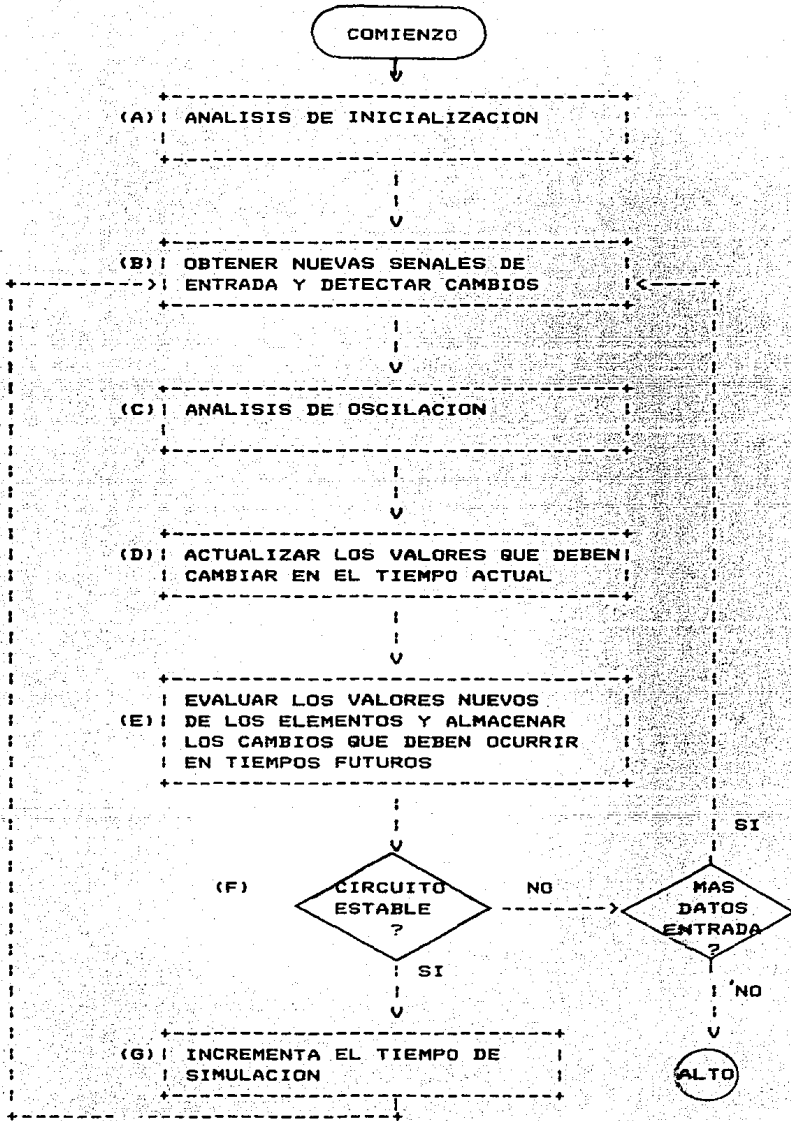


Fig 2.3 Diagrama a Bloques del Módulo Simulador.



En el bloque E se simulan todos los elementos afectados por la nueva entrada, si alguno de estos elementos cambia de estado es preparado para modificar su estado en un tiempo futuro. En el siguiente bloque (F), se determina la estabilidad del circuito. Si el circuito es inestable, se cuestiona la existencia de mas datos de entrada. Si los hay, se regresa al bloque B a leer el nuevo dato. En el caso de que el circuito sea estable, se incrementa el tiempo de simulación (bloque G) en una unidad.

#### II.3.4 Módulo de Salida.

-----

El Módulo de Salida debe producir una representación gráfica de los estados lógicos en los puntos de interés. Asimismo, éste módulo puede reportar la sustitución de macro-elementos por sus primitivas y producir tablas que contengan la descripción y localización de las fallas detectadas. El módulo de salida proporciona la información requerida por el usuario y va en función de las cualidades del sistema utilizado.

#### II.4 RETARDO.

-----

Debido a que la simulación digital se basa en el cálculo del valor lógico de cada señal como una función en el tiempo, los retardos dentro del circuito son sumamente importantes y deben ser considerados. En general, los retardos en un circuito pueden ser caracterizados por uno o mas de los siguientes modelos:

#### II.4.1 Retardo de Transporte

---

Este tipo de retardo implica que cada elemento provoca un retardo unitario en la propagación de la señal. Dicho de otra manera, si un cambio a la entrada de un elemento ocurre en un tiempo  $t$ , los efectos de este cambio estarán presentes en el tiempo  $t+1$  a la salida.

#### II.4.2 Retardo por Ambigüedad

---

El retardo de transporte de un dispositivo es difícil de determinar con exactitud. Los fabricantes de circuitos integrados generalmente especifican el retardo entre entradas y salidas del dispositivo. Sin embargo, el retardo que ocurre internamente en cada uno de los elementos no se especifica.

Para resolver éste problema se pueden asignar un par de valores de retardo a un elemento dado. Sea  $\Delta_m$  y  $\Delta_M$  los retardos mínimo y máximo a través del elemento. Estos retardos definen una región ambigua de duración  $\Delta_M - \Delta_m$ , durante la cual pueden ocurrir cambios en el valor de la señal. Esto se ilustra en la figura 2.4 (a).

#### II.4.3 Retardo de Subida y Bajada

---

Para algunos elementos, los tiempos de respuesta en la subida (de 0 a 1) y en la bajada (de 1 a 0) son diferentes debido a varios parámetros eléctricos tales como la capacitancia. En éste tipo de elementos se deben asignar dos retardos  $\Delta_s$  y  $\Delta_b$ .

El resultado de tener  $\Delta s \neq \Delta b$ , produce el efecto de modificar el ancho del pulso que se propaga. Esto se ilustra en la figura 2.4 (b). En esta figura, se tiene  $\Delta s = 1$  y  $\Delta b = 2$ , lo que produce que el ancho del pulso se incremente.

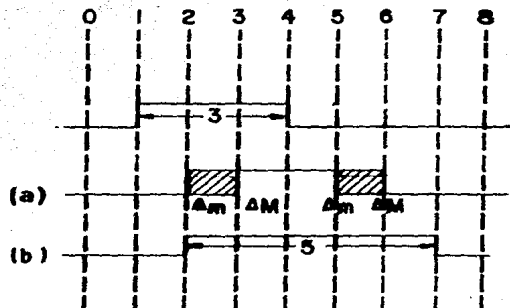


Fig. 2.4 (a) Retardo por ambigüedad  
(b) Retardo de Subida y Bajada

#### II.4.4 Retardo Inercial.

Un elemento requiere de cierta energía para cambiar de estado. Esta energía está en función de la amplitud y duración del pulso. El retardo inercial es la mínima duración de un pulso de entrada para lograr que un elemento cambie de estado. Este tipo de retardo se considera generalmente a la entrada del elemento.

Una combinación de los tipos de retardo anteriores puede ser utilizada en un simulador de circuitos digitales, dependiendo de las necesidades del modelo. Sin embargo, mientras mas tipos de retardo se utilicen, mas lento y complejo será el simulador.

Desde el punto de vista de la complejidad, el tipo de retardo mas sencillo de implementar es el Retardo de Transporte, también llamado Retardo Nominal. Si a cada elemento se le asigna el mismo valor de retardo  $\Delta t$ , entonces se puede tener un simulador de Retardo Unitario ( $\Delta t = 1$ ).

La simulación con Retardo de Subida y Bajada es menos sencilla que la anterior, y la simulación con Retardo por Ambigüedad es aún mas complicada, debido a la necesidad de tener mas estados lógicos. Finalmente, el Retardo Inercial es el mas difícil de simular debido a que si un pulso de entrada es mas corto que el Retardo Inercial, se requiere de un análisis complicado para efectuar los cambios de estado a la salida.

## II.5 SISTEMAS DE LOGICA MULTI-VALOR.

-----

En muchas aplicaciones de simulación no son suficientes dos valores lógicos (0, 1). Como se vió anteriormente el retardo por ambigüedad introduce una región de tiempo durante la cual la señal se encuentra en un estado de transición. Asimismo, cuando el circuito es encendido inicialmente o si existen oscilaciones durante su funcionamiento el valor de una señal puede ser indeterminado. Debido a éstas razones, la mayoría de los simuladores lógicos emplea un sistema con lógica multi-valor.

### II.5.1 Lógica de Tres Estados.

Cuando una señal cambia valores entra en un período de transición que no puede ser considerado como 1 o 0, por lo que se usa un tercer valor: indeterminado (u). El uso de un estado indeterminado implica el desarrollo de una lógica de tres estados. La figura 2.5 muestra algunas compuertas con este tipo de lógica.

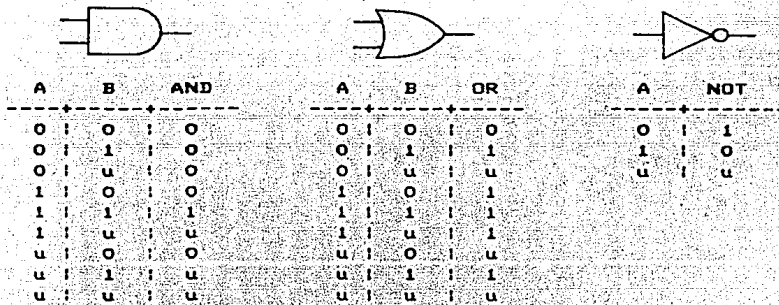


Figura 2.5 Lógica de Tres Estados.

### II.5.2 Lógica de Cinco Estados.

La lógica de cinco estados es útil cuando un simulador utiliza un retardo por ambigüedad debido a que este tipo de retardo considera los estados de transición 0/1 y 1/0. Los cinco estados son los siguientes:

Valor Lógico	Símbolo
0	0
1	1
transición 0 a 1	0/1
transición 1 a 0	1/0
indeterminado	u

La tabla de verdad para una compuerta NAND utilizando lógica de cinco estados se muestra en la figura 2.6.



NAND	0	1	0/1	1/0	u
0	1	1	1	1	1
1	1	0	1/0	0/1	u
0/1	1	1/0	1/0	u	u
1/0	1	0/1	u	0/1	u
u	1	u	u	u	u

Figura 2.6 Lógica de Cinco Estados.

### II.5.3 Lógica de Ocho Estados

Una de las deficiencias de la lógica de tres estados es que no es posible predecir un impulso provocado por ruido. Para resolver este problema, se desarrolló la lógica de ocho estados, que se muestra en la figura 2.7.

Valores Lógicos	Explicación	Secuencia Lógica
00	cero estático	
11	uno estático	
01	transición de 0 a 1	
10	transición de 1 a 0	
00*	cero estático con ruido	
11*	uno estático con ruido	
01*	transición 0 a 1 con ruido	
10*	transición 1 a 0 con ruido	

Fig. 2.7 Lógica de Ocho Estados.

El asterisco (\*) indica la presencia de un impulso provocado por ruido entre dos estados estables.

## II.6 TIPOS BASICOS DE SSD's.

-----

Como se mencionó en la Sección II.3, existen dos formas de representar un circuito en un simulador: Código Compilado y Tabla Estructurada. En esta sección se describe con mas detalle cada una de ellas.

### II.6.1 Simulación por Código Compilado.

-----

La primera fase en la simulación por código compilado es llevada a cabo por el módulo de estructura, que traduce la descripción del circuito a código de máquina.

Antes de que sea generado el código, los elementos del circuito se ordenan por niveles. Este proceso se efectúa de la siguiente manera:

1. Se asigna el nivel cero a las líneas de entrada.
2. Se asigna un número consecutivo de nivel a cada uno de los elementos y líneas de señal. A la señal de salida de un elemento se le asigna el mismo nivel que al propio elemento.
3. Se ordenan los niveles en forma ascendente, agrupando los elementos de cada nivel.

Un ejemplo del proceso anterior se muestra en la figura 2.8.



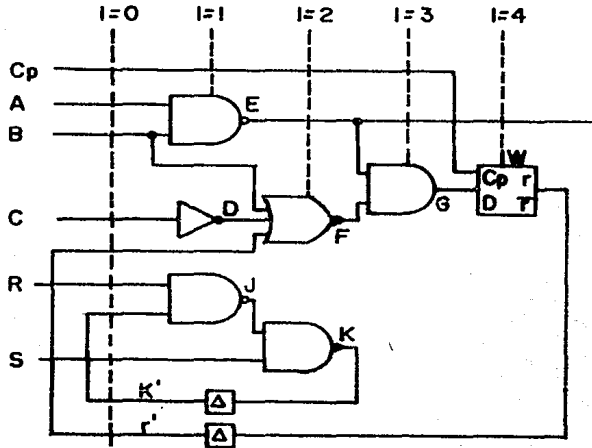


Fig 2.8 Simulación por Código Compilado.

Después de evaluar un elemento de un nivel  $k$  por medio del proceso anterior queda determinado el valor de las entradas en el nivel  $k + 1$ .

Una vez que los elementos han sido ordenados, se procede a generar el código. Generalmente se utiliza una palabra de computadora para especificar el valor de cada una de las señales del circuito, aunque en rigor sería suficiente un solo bit para un simulador con lógica de dos estados.

Para ilustrar la forma en que se genera el código, considérese una compuerta AND con tres entradas (A, B y C) y una salida (D). El código generado en instrucciones de un procesador hipotético sería el siguiente:

<u>Código</u>	<u>Comentarios</u>
CAL A	Colocar A en acumulador
ANA B	Ejecutar A AND B
ANA C	Ejecutar (A AND B) AND C
SLV D	Almacenar el resultado en D

El código que resulta se almacena comunmente en un archivo secuencial.

La segunda fase en la simulación por código compilado es llevada a cabo por el módulo de simulación, que ejecuta las instrucciones de código compilado de acuerdo al siguiente algoritmo:

1. Leer el valor inicial de cada entrada.
2. Leer los nuevos valores de entrada y actualizar los estados de los elementos. Si no hay valores de entrada entonces terminar.
3. Ejecutar el código compilado; es decir, simular la lógica de cada elemento.
4. Salida de resultados. Regresar al paso 2.

## II.6.2 Simulación por Tabla Estructurada

---

En un simulador por tabla estructurada el circuito se representa por un conjunto de tablas (vectores o matrices) que contienen los datos más importantes de cada elemento:

1. Valor lógico de las terminales de salida.
2. Tipo de elemento.
3. Retardo asociado.
4. Nodos asociados a las terminales de entrada (lista fanin).
5. Elementos conectados a las terminales de salida (lista fanout).
6. Estado del elemento, en caso de tener memoria.

Existen dos tipos básicos de estructuras de datos para representar un circuito: Tabla de Elementos y Listas Encadenadas.

En la Tabla de Elementos se asume que cada elemento tiene una sola salida por lo que puede asignarse el nombre del elemento al nombre de su señal de salida. Cada señal y elemento tiene un nombre alfanumérico (NOMBRE) así como un nombre interno (INDICE). El índice se utiliza para direccionar un dato deseado. La descripción que sigue se refiere a la figura 2.9, cuya estructura de datos se muestra en la tabla 2.1.

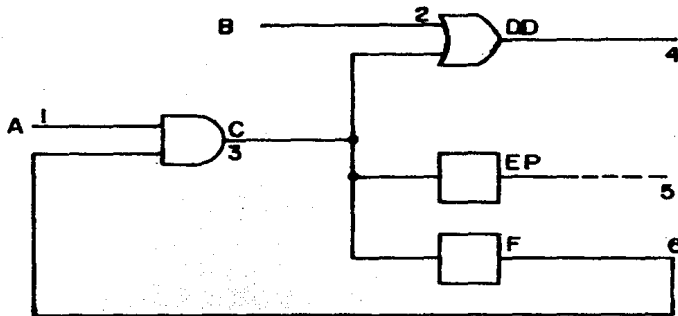


Fig. 2.9 Ejemplo de un Circuito para el Análisis de la Estructura de Datos por Tabla de Elementos.

INDICE	NOMBRE	VALOR	NFO	NFI	FIL	FOL	TIPO	RETARDO	ESTADO	
									INTERNO	E/S
1	A									
2	B									→ 4
3	C	0	3	2			AND			→ 5
4	DD						OR			→ 6
5	EP									
6	F									
7	.									→ 1
8	.									→ 6

(a)

(b)

Tabla 2.1 Tabla de Elementos Correspondiente al Circuito Mostrado en la Fig. 2.9.

La estructura de datos consiste en dos tablas: la matriz de Elementos (a) y el vector Lista Entradas-Salidas (b). La matriz de Elementos contiene las siguientes columnas:

1. NOMBRE (i) indica el nombre del i-ésimo elemento.
2. VALOR (i) es el valor lógico actual del i-ésimo elemento.
3. NFO (i) es el número de elementos conectados a la salida del i-ésimo elemento (fanout).
4. NFI (i) es el número de elementos conectados a la entrada del i-ésimo elemento (fanin).
5. FIL (i) es un apuntador que indica el primer elemento conectado a la entrada del i-ésimo elemento dentro de la Lista Entrada-Salida.
6. FOL (i) es un apuntador que indica el primer elemento conectado a la salida del i-ésimo elemento dentro de la Lista Entrada-Salida.
7. TIPO (i) especifica el tipo de compuerta del i-ésimo elemento.
8. RETARDO (i) indica el retardo asociado al i-ésimo elemento.

Por ejemplo, la compuerta de NOMBRE C (INDICE = 3) con un VALOR = 0, tiene un fanin de dos (señales A y F), un fanout de tres (compuertas F, EP y DD) y es de TIPO AND. El módulo simulador utiliza la estructura de datos para simular cada compuerta de la siguiente manera: FIL (3) indica el primer elemento (INDICE = 1) conectado a la entrada de la compuerta C, cuyo VALOR es consultado en el renglón del elemento A. El valor de las NFI - 1 entradas restantes sería calculado en forma similar. Con todas las entradas conocidas y sabiendo que TIPO (3)

= AND, el módulo simulador calcula el VALOR (3), afectando a los elementos indicados por FOL (3) y se repite el proceso para cada uno de estos elementos.

El segundo tipo de estructura de datos, Listas Encadenadas, utiliza una estructura para definir completamente cada elemento. A la estructura de un elemento se le denomina Descriptor de Elemento y tiene la forma general mostrada en la figura 2.10.

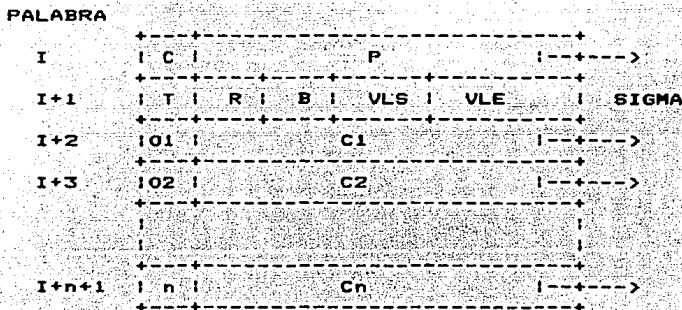


Fig. 2.10 Estructura de Datos por Cadena de Elementos.

La primera palabra de la estructura de un elemento se denomina Cabecera, identificada por la bandera C, que contiene al campo P cuya función es apuntar al primer elemento en el fanout.



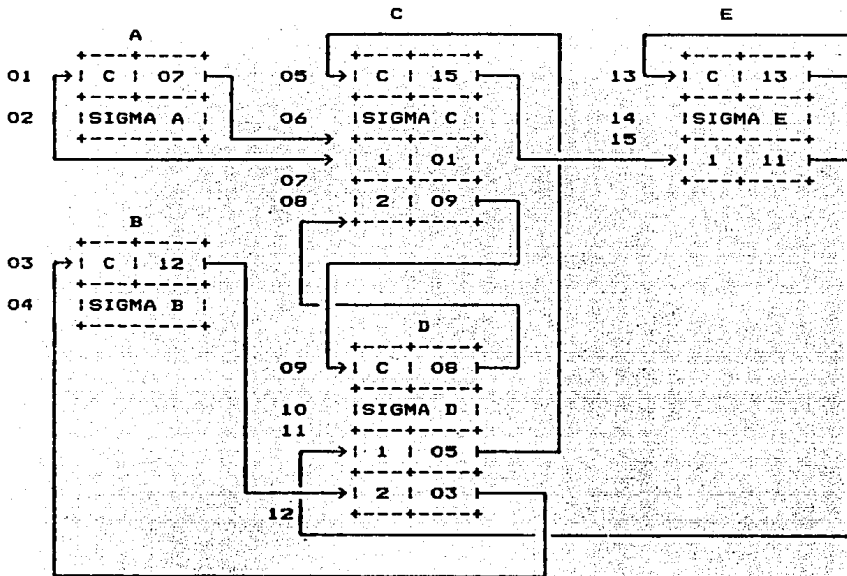


Fig. 2.12 Lista Encadenada del Circuito de la Fig. 2.11.

Los elementos A y B son entradas primarias por lo que no poseen fanin, y el elemento E es una salida primaria ya que carece de fanout. Cada elemento comienza con su cabecera. La primera palabra del elemento C apunta a la palabra 15, que es el primer elemento conectado a la entrada de E así como el primer elemento conectado a la salida de C. El contenido de la palabra 15 apunta al segundo elemento en el fanout de C, que es también el primer elemento en el fanin de D. Finalmente el contenido de la palabra 11 apunta a la cabecera del elemento que originó la cadena.



La estructura de datos de Listas Encadenadas puede ser usada de la siguiente manera: asumiendo que el elemento C cambia de estado, todos los elementos en su lista de fanout deben ser recalculados. El índice  $I = 1$  en la palabra 15 nos indica que la palabra  $15 - I = 14$  contiene a sigma, que especifica entre otras cosas, el tipo de elemento. El primer dato VLE de E es modificado al nuevo valor VLS de C. Con la información anterior es posible simular el elemento E. Se continúa en forma similar para los elementos restantes de la cadena.

## II.7 EVALUACION DE ELEMENTOS.

-----

Al proceso de evaluar la salida (o próximo estado) de un elemento, dadas sus entradas y su estado, se le denomina Evaluación de Elementos. Esta evaluación puede ser realizada utilizando cualquiera de las siguientes técnicas: evaluación por subrutina, evaluación por conteo de entradas, evaluación por tablas de verdad, y evaluación por tablas zoom.

Antes de discutir cada una las técnicas de evaluación, es importante introducir el concepto de Simulación en Paralelo. Considérese una compuerta AND con dos entradas A y B. Si el valor de A y B es almacenado en el  $i$ -ésimo bit de las palabras de computadora A y B respectivamente, entonces, al efectuar la operación AND de estas dos palabras, se obtiene el valor de salida de la compuerta en cuestión en el  $i$ -ésimo bit de la palabra resultante. Si la palabra de computadora consta de W bits es posible manejar W problemas diferentes simultáneamente (en paralelo).

En un simulador por código compilado es posible evaluar  $W$  elementos en paralelo en aproximadamente el mismo tiempo que tomaría evaluar un solo elemento. La desventaja de la simulación en paralelo es que toma tiempo el empaquetar y desempacar bits en una palabra.

En un simulador por tabla estructurada con trazo selectivo también es posible utilizar la simulación en paralelo. Por ejemplo, si se compara el vector  $V(i)$  con  $V'(i)$ , entonces es posible detectar un evento ocurrido cuando dos bits de la misma posición difieren en valor.

#### II.7.1 Evaluación por Subrutina.

-----

Esta técnica considera el desarrollo de una subrutina para cada tipo de elemento. Los parámetros de la subrutina deben incluir los nombres de las señales de entrada y salida para que la subrutina pueda calcular los nuevos valores a la salida. Generalmente la subrutina consiste en las operaciones lógicas del sistema de cómputo utilizado, o bien es escrita partiendo de las reglas de Boole en un lenguaje de alto nivel.

#### II.7.2 Evaluación por Conteo de Entradas.

-----

Para los tipos más comunes de compuertas, AND, OR, NAND y NOR es suficiente conocer el número de entradas con valores 0 y 1 para determinar la salida. Por ejemplo, si se considera una compuerta OR, entonces un CONTADOR puede registrar el número de entradas que tienen el valor uno lógico. Para un evento que

provoque un cambio de 0 a 1 en cualquiera de las entradas, el CONTADOR deberá ser incrementado. Para un evento que provoque un cambio de uno a cero, el CONTADOR deberá ser decrementado. Después de todos los eventos, si el contador es igual a uno, la salida será uno, y si el contador es igual a cero, la salida será cero. En cualquier otro caso, la salida permanece sin cambio respecto a su estado anterior. Esta técnica es, en consecuencia, sumamente rápida.

### II.7.3 Evaluación por Tablas de Verdad.

-----

Esta técnica considera la utilización de un vector de entradas como dirección de una tabla de verdad donde el valor de la tabla corresponde al valor de salida de la compuerta. Si se compara el valor obtenido de la tabla con el valor actual, es posible determinar la existencia de un evento. Este procedimiento puede ser utilizado para simular compuertas, flip-flops, codificadores y multiplexores.

### II.7.4 Evaluación por Tablas Zoom.

-----

Esta técnica es una generalización de la técnica de evaluación por tabla de verdad. Utiliza la palabra Sigma de la estructura de datos por Listas Encadenadas y considera que cada uno de los datos contenidos en esta palabra es un apuntador hacia ciertas direcciones en una tabla zoom, la cual contiene las acciones que pueden ocurrir dependiendo de la información contenida en Sigma. Estas acciones pueden ser:

1. Transición 0 a 1
2. Transición 1 a 0
3. Ninguna Acción
4. Producir un retardo inercial, etc.

Estas tablas pueden ser muy extensas, pero son sumamente rápidas al determinar el efecto que produce un evento a la entrada de un elemento.

Los conceptos descritos en este capítulo son básicos en el diseño de cualquier Simulador de Sistemas Digitales. Sin embargo, existen simuladores con aplicaciones muy específicas que requieren de otros conceptos que caen fuera del propósito del presente trabajo.

En el diseño de un SSD deben considerarse los temas contemplados en el presente capítulo. En el capítulo siguiente se describe el Modelo del Sistema y la Estructura Básica del Simulador desarrollado. Asimismo, se selecciona el tipo de retardo, la lógica multi-valor, y el tipo de simulador de acuerdo a los conceptos revisados anteriormente.

DISEÑO DEL SIMULADOR INTERACTIVO (SIM/CD)

---

III.1 INTRODUCCION.

---

Un simulador de circuitos digitales puede considerarse como una tableta de pruebas basada en software que permite la simulación de circuitos digitales sin el uso de componentes físicos. Este tipo de sistemas es un auxiliar para los diseñadores experimentales y constituye una poderosa y económica herramienta en la enseñanza. La existencia del mismo a niveles escolares proporciona al estudiante una gran experiencia en la operación y el entendimiento de los dispositivos digitales, así como en el diseño por computadora, que es la tendencia actual de la ingeniería.

El uso de un simulador lógico es igual al uso de una tableta de pruebas: se interconectan las compuertas lógicas de acuerdo a las funciones requeridas; se energiza el circuito y se prueba aplicando algún tipo de estímulo a la entrada y observando el resultado a la salida. Si el resultado es el esperado, se implementa el diseño en su presentación final (circuito impreso, chasis, etc). Sin embargo, es muy frecuente que la salida no sea la deseada, debido a fallas en la implementación lógica, en el

alambrado o bien por la omisión de algún componente. En este punto se tiene que analizar el sistema para descubrir la falla, remediarla e iniciar el proceso de prueba nuevamente.

La tableta de pruebas basada en software se utiliza de manera análoga y ofrece ventajas significativas. Se proporciona al simulador una descripción de la interconexión de las compuertas del diseño propuesto. El tipo y la cantidad de compuertas está limitado por la capacidad del programa y no por la cantidad de circuitos integrados disponibles en el laboratorio. Posteriormente se le indica al simulador el tipo de estímulo necesario para verificar el diseño. La complejidad e interrelación de los estímulos no está limitada por la existencia de generadores e interfaces síncronas. El hecho de que el simulador se base en software proporciona al diseñador el equivalente de generadores de señales programables.

El uso de una pantalla de video permite el monitoreo de varias señales de acuerdo a la capacidad de la pantalla, y en consecuencia, es comparable con un osciloscopio multitrazo o un analizador lógico.

La mayor desventaja de la tableta de pruebas basada en software, es la necesidad del diagnóstico de la respuesta en tiempo real. Sin embargo, se debe tener presente que se trata de un simulador y no de un emulador. Si se acepta como propósito primario la validación de la operación lógica del diseño y si se pospone el análisis de tiempo hasta que la lógica es verificada, el impacto de esta limitación se reduce al mínimo.

El Simulador de Sistemas Digitales realizado en el presente trabajo se denomina SIM/CD (SIMulador de Circuitos Digitales). En este Capítulo se describen sus características y elementos constitutivos.

### III.2 CONSIDERACIONES DE DISEÑO.

-----

En el capítulo anterior se hizo mención de la existencia de dos aplicaciones principales para un simulador de sistemas digitales: Evaluación de un nuevo diseño y Análisis de fallas.

El presente trabajo contempla la elaboración de un SSD, que sea fácil de manejar, fácil de entender, que sea rápido y eficiente.

SIM/CD sirve para la primera aplicación mencionada (evaluación de un nuevo diseño), permitiendo al diseñador probar que la lógica utilizada, el comportamiento en el tiempo y las características de propagación de las señales utilizadas, sean correctas. Cae fuera del alcance del trabajo realizado la segunda aplicación, que enfoca su atención al análisis de fallas de los sistemas lógicos.

Los elementos que intervienen en un proceso de simulación son: Sistema Real, Modelo Simulador, y Computadora utilizada. Con estos elementos, el sistema bajo estudio se representa a través de un modelo en forma tal que sea capaz de simularse con una computadora y posteriormente el modelo se ejercita en la computadora para obtener información acerca del sistema

representado. A continuación se analizan y consideran éstos elementos.

### III.2.1 Consideraciones del Sistema a Simular.

---

SIM/CD incorpora los aspectos más importantes del sistema real con el objeto de que el modelo del mismo contenga realismo y simplicidad (Sec. I.8).

El sistema a simular es un dispositivo digital (Circuito digital) al cual se le aplican diversos estímulos externos con la finalidad de obtener una respuesta determinada, dependiendo de las cualidades del circuito y de los estímulos aplicados.

Al circuito digital se le aplican los estímulos provenientes de un generador de señales. A SIM/CD se le indica la conformación de los estímulos externos mediante una serie de instrucciones; el hecho de que SIM/CD se base en software es equivalente a tener generadores de señales programables, lo cual permite elaborar el estímulo externo deseado sin ninguna limitación.

Un circuito digital está constituido por elementos básicos llamados compuertas, las que a su vez conforman sistemas más complejos. Estos elementos están interconectados de tal forma que al aplicar un estímulo externo se presenta a la salida del circuito una respuesta determinada. Ahora bien SIM/CD contempla la inclusión de seis elementos básicos constitutivos de un circuito digital. Los elementos básicos (primitivas) utilizados



son las compuertas AND, OR, XOR, NAND, NOR, y XNOR. La evaluación de estos elementos es a través de subrutinas para cada función. Para evaluar un inversor lógico (NOT), es posible utilizar la primitiva NAND con una sola entrada.

Para que el circuito digital se comporte como se desea, estos elementos se deben de conectar de una manera determinada. De igual forma, el usuario debe de indicar a SIM/CD la interconexión de los elementos por medio de una serie de instrucciones (de manera análoga a la descripción de los estímulos externos), que son almacenadas por el simulador en una estructura de datos la cual, además de indicar el tipo de compuerta utilizado, indica los nodos asociados a la compuerta, lo que en conjunto permite conocer la interconexión del circuito digital.

El sistema real se energiza, se le aplican los estímulos y genera un resultado. A éste resultado se llega gracias al comportamiento interno de las compuertas, la interconexión y los estímulos externos. Con la finalidad de que el sistema real se describa con máximo detalle, en SIM/CD se toman en cuenta los aspectos de retardo, inicialización, lógica multivalor y tipo de evaluación de elementos, los cuales se describen ampliamente en la Sec. III.2.2.

En cuanto a las respuestas, al sistema real se le pueden conectar un gran número de transductores --generalmente instrumentos muy costosos-- , los cuales interpretan los resultados. El propósito de SIM/CD es la verificación de que la

lógica utilizada en un diseño sea correcta, por lo que la utilización de una pantalla de video es adecuada para la interpretación de los resultados del proceso de simulación.

### III.2.2. Consideraciones del Modelo Simulador.

-----

Por modelo se entiende la abstracción de un sistema real con el propósito de control y/o predicción de comportamiento. El objetivo del modelo es permitir el cambio o la introducción de variables en el sistema modelado con el fin de observar como afectan éstas a otras variables del sistema o a la totalidad del mismo.

Un modelo debe de representar las principales cualidades del sistema real. Por tal motivo, en un modelo de un circuito lógico es necesario hacer hincapié en el tipo de retardo utilizado, la lógica multivalor, la inicialización, y la evaluación de elementos. Estas y otras cualidades se describen a continuación.

#### Retardo

-----

Debido a que la simulación digital se basa en el cálculo del valor lógico de cada señal como una función en el tiempo, los retardos dentro del circuito son muy importantes y deben ser considerados.

Como se mencionó en el Capítulo II, existen cuatro tipos de retardo:

- Retardo de Transporte
- Retardo por Ambigüedad
- Retardo de Subida y Bajada
- Retardo Inercial

SIM/CD utiliza el retardo de transporte, donde si existe un cambio en la entrada de alguno de los elementos en el tiempo  $t$ , el efecto de este cambio está presente a la salida en el tiempo  $t + 1$ .

A cada una de las primitivas se le ha asignado el mismo retardo  $\Delta t$ , por lo que el simulador se denomina de retardo unitario  $t = 1$ .

Este tipo de retardo presenta las siguientes ventajas: es fácil de entender, es sencillo de simular, se acerca al comportamiento real de los dispositivos lógicos, y por último es el tipo de retardo que consume menos tiempo de máquina.

El retardo unitario se implementa en SIM/CD por medio del uso de dos vectores, que contienen los estados lógicos para el tiempo de simulación  $t$  y  $t+1$  respectivamente.

#### Lógica Multivalor

---

Cuando una señal cambia de valor, entra en un periodo de transición que no puede ser considerado como 1 o como 0, por lo que se hace necesaria la utilización de un tercer valor lógico conocido como indeterminado. En el presente trabajo se toma en cuenta esta consideración y se utiliza una lógica de tres estados

(1,0, INDETERMINADO), suficiente para evaluar los estados en que se pueda encontrar el dispositivo lógico analizado. Caen fuera del propósito del presente trabajo la evaluación de los estados de transición de 0/1 y 1/0, así como la predicción de impulsos provocados por ruido, por lo que no se hace necesaria la utilización de lógica de 5 u 8 estados. Si se desea mayor información en el empleo de lógica de 5 y 8 estados, consúltese la Sec. II.5.2 y II.5.3 respectivamente.

Cada uno de los estados lógicos del sistema tiene una representación interna dentro del programa con el fin de identificar el valor lógico en el que se encuentran los elementos. Así, la representación interna del estado lógico bajo es 0, el estado lógico alto es 1, y el estado indeterminado es 3.

Asimismo, estos tres estados tienen una representación gráfica para su utilización en la presentación de resultados (Sec. III.7). La representación gráfica del nivel bajo es "\_", la del nivel alto es "X" y la del estado indeterminado es "?".

#### Inicialización de Nodos.

Cuando se simula un circuito, es necesario especificar el valor inicial de los elementos. A menudo este estado inicial es desconocido y una forma común de resolver este problema es la de fijar el estado inicial de cada elemento a un valor indeterminado (?).

SIM/CD contiene una subrutina de inicialización de elementos para asegurar que todos los nodos del circuito se encuentren en

un estado indeterminado al iniciar el proceso de simulación.

#### Primitivas

SIM/CD parte de la utilización de seis primitivas o elementos constitutivos básicos que son las compuertas AND, OR, y XOR, NAND, NOR, y XNOR. La razón principal es que los estudios sobre dispositivos lógicos parten del análisis de estas compuertas para la construcción de cualquier tipo de circuito lógico.

Cada una de las primitivas contempla la utilización de hasta cinco entradas y dos salidas. Se consideran cinco entradas ya que la mayoría de los chips comerciales contienen compuertas con dos a cinco entradas; y se consideran dos salidas, una para la salida normal y otra salida negada que es utilizada para simular las compuertas NAND, NOR, y XNOR.

#### Tipo de SSD

Como se mencionó en la sección II.3, existen dos tipos básicos para representar un circuito en un simulador: Código Compilado, donde el comportamiento del circuito a simular se representa por medio de una secuencia de instrucciones de computadora; y Tabla Estructurada, donde el circuito se representa por un conjunto de tablas que describen la topología del circuito, así como datos tales como el tipo de elemento, retardo, etc. SIM/CD utiliza la representación por Tabla Estructurada, por ser el método que marcha a la vanguardia en el

campo de la simulación, por ser un método eficiente, versátil, rápido, fácil de realizar y fácil de entender.

En la representación por Tabla Estructurada (Sec. II.6.2) existen dos tipos básicos de estructuras de datos para representar un circuito: Tabla de Elementos y Lista Encadenada.

SIM/CD utiliza la estructura Tabla de Elementos donde además de describir el valor lógico de las terminales de entrada y salida de los dispositivos, tipo de elemento, retardo, etc., se utilizan las cualidades de la misma, como son la asignación de índices internos a los elementos y sobre todo la facilidad que representa el construir éste tipo de estructura de datos. SIM/CD utiliza una tabla para describir la lógica del circuito, otra en donde se describen las señales a aplicar al circuito y además se utilizan dos tablas en donde se acomodan los cambios en el tiempo de los estados lógicos en los nodos.

#### Evaluación de Elementos.

-----

Al proceso de evaluar la salida (o próximo estado) de un elemento dadas sus entradas y el tipo de elemento se le denomina Evaluación de Elementos. Esta evaluación se puede realizar con cualquiera de las siguientes técnicas:

- Evaluación por subrutina.
- Evaluación por conteo de entradas.
- Evaluación por tablas de verdad.
- Evaluación por tablas zoom.

SIM/CD utiliza la evaluación por subrutina, donde a través de algoritmos simples se describen las funciones lógicas de las compuertas utilizadas. También se toman en cuenta algunos aspectos lógicos para producir una evaluación de elementos mas rápida. Así, por ejemplo, si en la evaluación de una compuerta AND aparece un cero en alguna de sus entradas, se fija su salida a cero y no se hace necesario revisar los valores de las entradas restantes.

### III.2.3. Consideraciones de la Computadora.

---

El proceso de simulación relaciona a la computadora con el modelo realizado, por lo que es necesario considerar algunas cualidades que debe reunir la computadora utilizada.

#### Lenguaje de Programación Utilizado.

---

Como se mencionó en el Capítulo I, después de la utilización de programas digitales de simulación analógica, se procedió a utilizar lenguajes muy poderosos de propósito general como el FORTRAN, ALGOL, BASIC, etc.

El lenguaje BASIC representa un lenguaje de alto nivel, de propósito general muy difundido y aceptado porque es fácil de aprender y de aplicar aun para el usuario que no esté familiarizado con la ciencia de la computación.

El propósito del presente trabajo es que SIM/CD sea interactivo de tal forma que aunque el conocimiento del usuario

respecto a sistemas computacionales sea muy pobre, le sea posible utilizarlo exitosamente, ya que éste lo guía paso a paso para su operación. No obstante este propósito tangible, se persigue otro igualmente importante, que se refiere a que el usuario interesado en el diseño lógico pueda analizar fácilmente el programa de SIM/CD, pueda modificarlo, y sobre todo introducirle nuevas cualidades. Esto se logra mediante la utilización de un lenguaje tan versátil como lo es el BASIC, y ésta es la razón para la utilización del mismo en el simulador realizado.

#### Consideraciones del Hardware Utilizado.

-----

Otra de las finalidades que persigue SIM/CD es que pueda funcionar en una gran variedad de equipos que pueden ir desde las microcomputadoras personales hasta computadoras de grandes dimensiones sin necesidad de modificar el programa substancialmente.

La industria de la computación actualmente persigue el mismo fin, y la solución que se le está dando en forma general es que los sistemas desarrollados sean compatibles con sistemas IBM. Por lo tanto, SIM/CD es un sistema compatible con la gama de microcomputadoras IBM-PC.

La cantidad de memoria requerida por el programa es sumamente pequeña. El programa ocupa alrededor de 8Kb para la versión 1.0 y menos de 21Kb para la versión 2.0. , por lo tanto, el simulador de sistemas digitales puede ser implementado en prácticamente cualquier tipo de computadora.



### III.3. ALGORITMO DE SIM/CD.

-----

Una vez considerados los aspectos más importantes del sistema real, del modelo simulador y de la computadora a utilizar, el siguiente paso en el desarrollo de SIM/CD es la construcción del diagrama a bloques que incluye todas las funciones necesarias para llevar a cabo la simulación de un circuito.

Como se describe en la Sección II.3, un simulador de sistemas digitales consta de cuatro módulos: Módulo Preprocesador, Módulo de Estructura, Módulo Simulador y Módulo de Salida. Estos módulos se muestran en la Figura 3.1 para SIM/CD.

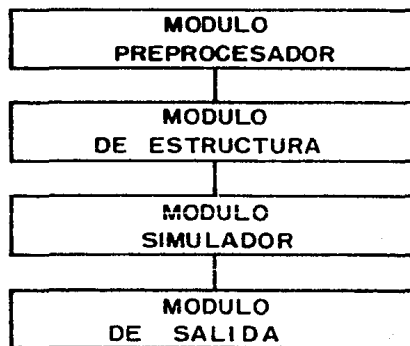


Fig. 3.1 Módulos de SIM/CD

---

En base a los módulos anteriores, se construye el algoritmo de SIM/CD, mostrado en la Figura 3.2.

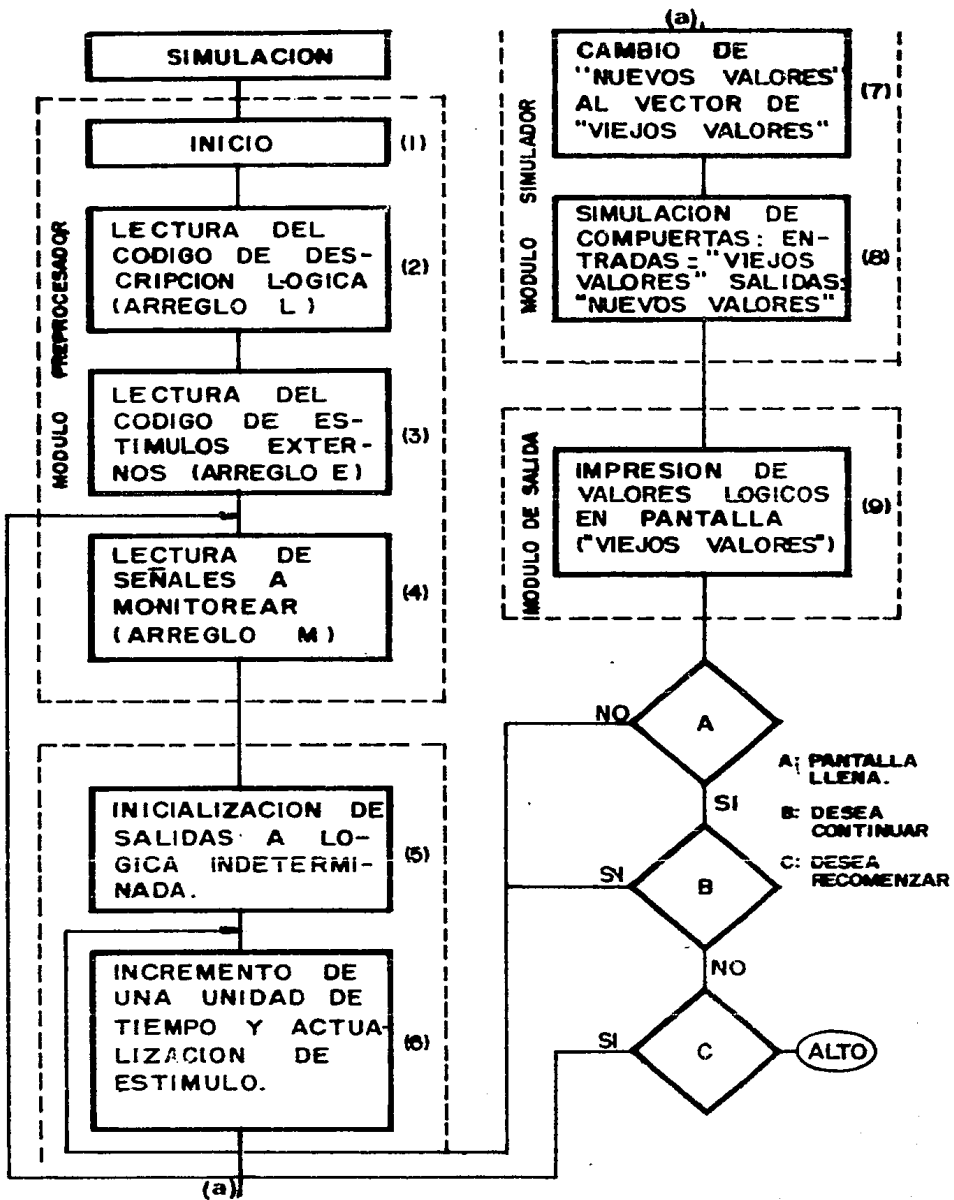


Fig. 3.2. Algoritmo de SIM/CD

## Descripción del Algoritmo.

-----

El Bloque 1 "Inicialización de variables" define las variables utilizadas y las coloca en su valor inicial. El Bloque 2 "Lectura de la Descripción Lógica" efectúa la lectura de los elementos constitutivos del circuito a simular. El Bloque 3 "Lectura de Estímulos Externos" efectúa la lectura de las señales externas aplicadas al circuito. El Bloque 4 "Lectura de Señales a Monitorear" efectúa la lectura de los puntos que el usuario desea observar. Estos cuatro bloques (1-4) forman parte del Módulo Preprocesador, que se describe en la Sec. III.4.

El Bloque 5 "Inicialización de los Nodos" Asigna a cada uno de los nodos del circuito el valor lógico indeterminado en los vectores Antiguo y Nuevo. El Bloque 6 "Incremento del Tiempo de Simulación y Actualización de Estímulo" incrementa el contador de tiempo de simulación en una unidad y actualiza el vector Nuevo con el estímulo externo correspondiente al tiempo de simulación actual. El Bloque 7 "Colocación de los Valores del Vector Nuevo en el Vector Antiguo" transfiere los estados lógicos del vector N al vector A. El Bloque 8 "Evaluación de Elementos" realiza las funciones lógicas para cada compuerta a simular con la información contenida en el vector Antiguo como entrada y colocando el resultado en el vector Nuevo. Los bloques 5, 6, 7 y 8 forman parte del Módulo Simulador, que se describe en la Sec. III.4.

El Bloque 9 "Impresión de los Estados Lógicos de los Nodos a Monitorear" imprime en la pantalla los estados lógicos de los

nodos que el usuario desea monitorear tomados del vector Antiguo. Este bloque constituye el Módulo de Salida (Sec. III.7).

Al final del Algoritmo, existen tres bloques de toma de decisión, donde se realiza una bifurcación a distintos bloques del diagrama. El primer bloque de decisión investiga si la pantalla puede o no incluir mas resultados. El segundo bloque de bifurcación cuestiona al usuario si desea continuar con el proceso de simulación después de que se ha llenado la pantalla. Finalmente, el tercer bloque pregunta al usuario si desea recomenzar el proceso a partir de la lectura de los puntos a monitorear.

El Módulo de Estructura, se construye internamente a partir de los datos introducidos por el usuario en el Módulo Preprocesador. El Módulo de Estructura se describe en la Sec. III.5.

En las siguientes secciones se describe cada uno de los Módulos con detalle.

#### III.4. MODULO PREPROCESADOR.

El Módulo Preprocesador efectua la lectura de todos los datos introducidos por el usuario. Estos datos deben describir las compuertas utilizadas, la interconexión de estas, las señales de entrada, y los puntos del circuito que se desean monitorear.

Asimismo, el módulo preprocesador tiene las funciones de inicialización de variables y la verificación de que los datos introducidos por el usuario sean válidos de acuerdo a las reglas de sintaxis predefinidas.

#### III.4.1. Inicialización de Variables.

-----

Como todo programa de computadora, SIM/CD contiene una sección de inicialización de variables. En ésta sección se definen las variables utilizadas asignando a cada una de ellas su valor inicial. Se debe mencionar que algunas de las variables inicializadas no cambian de valor en el transcurso de todo el programa, por lo que pueden considerarse como constantes.

#### III.4.2. Lectura de Descripción Lógica.

-----

Para describir cada uno de los elementos del circuito es necesario proveer la siguiente información :

- Tipo de elemento lógico (AND , OR , XOR)
- Nodos conectados a las entradas
- Nodos conectados a las salidas

Se consideran 5 entradas para cada una de las compuertas utilizadas. Las entradas no utilizadas se dejan en el estado de no conexión por medio de un número de nodo nulo.

En el caso de las salidas, se consideran dos salidas para cada compuerta: La primera corresponde a la salida normal de la compuerta, mientras que la segunda se utiliza como salida negada para las compuertas NAND, NOR y XNOR.

Los datos anteriores son introducidos por medio de una sola instrucción de INPUT. Primero se lee el tipo de compuerta y se almacena en la variable L\$. Enseguida se introducen las entradas almacenándose en los primeros cinco elementos de la matriz (arreglo bidimensional) L de Descripción Lógica, descrita en la Sección III.5.1. Finalmente, son leídas las salidas que son almacenadas en el sexto y séptimo elementos de la matriz L.

Las compuertas del circuito son leídas en la forma descrita anteriormente hasta que el usuario introduzca la palabra FIN en el lugar correspondiente al tipo de elemento lógico.

#### III.4.3. Lectura de Estímulo Externo.

-----

Para describir los estímulos externos es necesario proveer la siguiente información:

- Número de Nodo donde se aplica el estímulo externo
- Valor Lógico inicial
- Tiempo inicial
- Tiempo en el que ocurre el primer cambio de estado
- Tiempo en el que ocurren los cinco cambios de estado subsecuentes

Los datos anteriores son introducidos por medio de una sola instrucción de INPUT almacenándose en la matriz E (Estímulo Externo), que se describe con detalle en la Sección III.5.1.

#### III.4.4. Lectura de Puntos a Monitorear.

-----

Una vez conocida la descripción lógica y los estímulos externos aplicados al circuito, el módulo preprocesador investiga cuales son los nodos que el usuario desea monitorear. En principio, cualquier número de nodos puede ser monitoreado por SIM/CD. Sin embargo, el programa actual puede manejar un máximo de diez puntos a monitorear debido a que es mas fácil para el usuario interpretar un menor número de nodos a la vez.

El número de los nodos a monitorear se almacena en el vector (arreglo unidimensional) M o Monitor.

#### III.4.5. Chequeo de Sintaxis.

-----

Al introducir los datos, el usuario puede cometer errores o introducir información que no está en el formato adecuado. Por tal motivo, es necesario verificar que los datos introducidos tengan la sintaxis adecuada. Para este fin, se han introducido subrutinas de chequeo de errores en datos en las secciones de lectura.

Es importante aclarar que el programa no verifica que la lógica de los datos introducidos sea correcta, sino que solamente verifica que dichos datos tengan el formato adecuado. Es decir,

si el usuario introduce una descripción de un circuito por medio de los comandos correctos, pero con una lógica de interconexiones incorrecta, entonces el programa no detecta esta inconsistencia y los resultados son impredecibles.

Una de las verificaciones que se realizan es la del tipo de compuerta. Si el usuario introduce un tipo de compuerta no valido, éste recibirá un mensaje de error. Asimismo, existe un chequeo del número de nodos contenidos en el circuito, del número de elementos introducidos por el usuario, y del número de estímulos externos aplicados. Si alguno de ellos rebasa el límite establecido, se envía un mensaje de error al usuario.

### III.5. MODULO DE ESTRUCTURA

-----

Como se mencionó en la sección II.3.2, la finalidad del Módulo de Estructura es preparar las estructuras de datos que se requieren para una simulación eficiente. Debido a que el simulador desarrollado en el presente trabajo es del tipo de "Tabla Estructurada", el módulo de estructura produce un conjunto de tablas (vectores y matrices) que contienen los datos mas importantes de cada elemento a simular. Estos datos son introducidos por el usuario mediante el Módulo Preprocesador.

#### III.5.1. Descripción de la Tabla de Elementos.

-----

La Tabla Estructurada que utiliza SIM/CD es una Tabla de Elementos compuesta por una Matriz de Elementos Lógicos, una



Matriz de Estimulos Externos y dos Vectores que contienen los valores lógicos de todos los nodos del circuito. Esta tabla estructurada se muestra en la fig. 3.3.

INDICE	TIPO	E1	E2	E3	E4	E5	S1	S2
1								
2								
...								

MATRIZ DE ELEMENTOS

INDICE	NODO	V.I.	T0	T1	T2	T3	T4	T5
1								
2								
...								

MATRIZ DE ESTIMULOS EXTERNOS

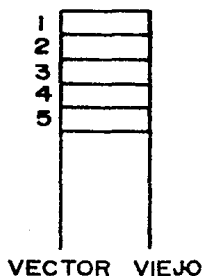
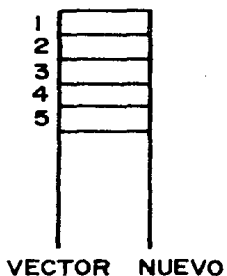


Fig. 3.3. Tabla de Elementos de SIM/CD

La matriz de elementos lógicos "L" contiene las siguientes columnas:

1. TIPO Tipo de Compuerta
2. E1 Nodo conectado a la Entrada 1
3. E2 Nodo conectado a la Entrada 2
4. E3 Nodo conectado a la Entrada 3
5. E4 Nodo conectado a la Entrada 4
6. E5 Nodo conectado a la Entrada 5
7. S1 Nodo conectado a la Salida
8. S2 Nodo conectado a la Salida Negada

Como puede observarse, la matriz L contiene los números de los nodos conectados a las entradas y a las salidas de cada elemento; es decir, esta matriz provee al simulador con la información de las interconexiones de las compuertas del circuito. El máximo número de renglones permisible para esta matriz es de 100 (i.e. cien compuertas máximo).

La matriz de Estímulos Externos "E" contiene las siguientes columnas:

1. NODO Nodo donde se aplica la señal
2. V.I. Valor Lógico Inicial
3. T0 Tiempo Inicial del Estimulo Externo
4. T1 Tiempo en el que ocurre el primer cambio de estado
5. T2 Tiempo en el que ocurre el segundo cambio de estado
6. T3 Tiempo en el que ocurre el tercer cambio de estado
7. T4 Tiempo en el que ocurre el cuarto cambio de estado
8. T5 Tiempo en el que ocurre el quinto cambio de estado

La información contenida en la matriz E provee al simulador los tiempos de simulación en los que ocurre un cambio en las señales externas aplicadas en los nodos de la primera columna. El máximo número de renglones (señales externas) permisibles en esta matriz es de 30.

Los vectores Antiguo (A) y Nuevo (N) contienen una sola columna. Cada elemento de estos vectores almacena el valor lógico de cada uno de los nodos del circuito. Así, A(5) contiene el estado lógico Antiguo del nodo número cinco; N(5) contiene el estado lógico Nuevo del mismo nodo. El uso de estos dos vectores implica un retardo de tipo unitario; es decir, cada estado en el vector Antiguo aparece un ciclo de tiempo después de que éste aparece en el vector Nuevo.

El máximo número de elementos de los vectores A y N es de 100 (cien nodos en el circuito). Sin embargo, si se requiere aumentar el número de nodos, o bien el número de señales externas o de elementos lógicos, se puede realizar fácilmente éste cambio, re-dimensionando los arreglos L, E, N, y A.

### III.5.2. Interrelación de los datos.

Los valores de las tablas y vectores descritos están interrelacionados de la siguiente manera.

Los vectores Antiguo y Nuevo tienen valores lógicos indeterminados para cada uno de los nodos al inicio de la simulación. Las señales contenidas en la tabla E proporcionan la

fuerza primaria de información para actualizar los estados lógicos en el vector Nuevo.

El vector Nuevo con la información actualizada de los estímulos externos se copia al vector Antiguo. La tabla L de Elementos Lógicos recurre a éste vector (A), para obtener los valores lógicos de las entradas de cada compuerta, para ser utilizados por el Módulo Simulador (Sec. III.6) al ejecutar las funciones lógicas. El resultado de la función lógica es almacenado en el nodo apuntado por la salida de la tabla E en el vector Nuevo.

#### III.6. MODULO SIMULADOR.

El Módulo Simulador está formado por una serie de subrutinas que son llamadas desde la "Malla Principal de Simulación". La malla utiliza un contador de tiempo de simulación "T", que da inicio con  $T=0$  e inmediatamente la subrutina "Actualizar Estímulo" es llamada.

La función de la subrutina Actualizar Estímulo es verificar si el tiempo T analizado es el tiempo inicial de algún estímulo externo. En este caso se coloca en el vector Nuevo el valor indicado en el arreglo E para el nodo correspondiente. Posteriormente se verifica si en el tiempo T analizado existe un cambio de algún estímulo externo (llamado marca de tiempo), de ser así, el valor del nodo analizado se trae del vector Nuevo, se invierte y se almacena nuevamente en el vector Nuevo. Esta operación se repite para cada uno de los estímulos externos, para

cada uno de los cambios de tiempo introducidos en el arreglo E.

El siguiente paso en la Malla Principal de Simulación es la colocación de todos los valores del vector Nuevo en el vector Antiguo, para pasar posteriormente a la subrutina de "Funciones Lógicas" .

La subrutina "Funciones Lógicas" tiene la función de identificar, para cada una de las compuertas introducidas en el arreglo L, el tipo de compuerta de que se trata, de acuerdo a la letra A, O y X introducida, para enviar el control del programa a la sección de evaluación de elementos correspondiente.

La operación de las subrutinas de evaluación de elementos se basa en lógica elemental; así para la compuerta AND se fija la variable Y a un valor lógico uno, para posteriormente identificar los nodos que sirven de entrada a la compuerta analizada. Se verifica en el arreglo A si alguna de estas entradas es cero, en cuyo caso Y se hace igual a cero y se pasa inmediatamente a la subrutina anidada en la de "Funciones Lógicas" llamada "Fijado de Salidas". En caso de no encontrar ningún cero, se verifica si el valor es indeterminado. En este caso la variable Y se fija al valor indeterminado y se revisa el valor del siguiente nodo considerado como entrada a la compuerta. Como se puede ver en caso de no existir valores lógicos cero o indeterminado, la variable Y continúa con el valor uno.

La subrutina de evaluación de la compuerta OR funciona de manera muy semejante. Se fija la variable Y inicialmente a cero,

se identifican las entradas de la compuerta, se verifica si alguna de estas entradas es igual a uno lógico, lo que provoca que la variable Y cambie su valor a uno y se procede a la subrutina de "Fijado de Salidas"; a continuación, se verifica en el vector Antiguo si el valor del nodo es indeterminado, en cuyo caso Y será igual a indeterminado y también se procede a la subrutina ya mencionada. En caso de no existir estas condiciones, la salida Y conserva su valor anterior. En esta secuencia se analizan las tres opciones posibles para cada una de las entradas de las compuertas del arreglo L.

La evaluación de la compuerta XOR se realiza de la siguiente manera. Se coloca la variable Y en valor indeterminado, se identifican los nodos correspondientes a las entradas de la compuerta; se identifica si alguno de ellos tiene valor de indeterminado, en cuyo caso se procede a la subrutina de "Fijado de Salidas" con Y en el valor inicial. En caso contrario se hace la operación aritmética de sumar el valor interno de la primera entrada al valor interno de la segunda entrada; si el resultado es igual a uno (en caso de que solo una entrada sea uno lógico), la variable Y se hará igual a uno lógico. En cualquier otro caso Y será igual a cero y se procede con la siguiente subrutina.

Como se puede observar en cada una de las compuertas se pasa a la subrutina "Fijado de Salidas" con la variable Y conteniendo el valor de salida de la compuerta evaluada. En esta subrutina se detecta si la salida de la compuerta evaluada es de tipo normal

(compuertas AND, OR, XOR), o es del tipo negada (compuertas NAND, NOR, XNOR), y el resultado se fija en el vector Nuevo de acuerdo al nodo indicado en el arreglo L.

La operación de identificación de compuerta, evaluación de ésta y fijado de salidas en el vector Nuevo, se realiza para cada una de las compuertas del arreglo L en el tiempo T analizado.

Posteriormente se procede con la subrutina "Impresión de Salidas", que imprime en la pantalla el tiempo inicial de simulación, el tiempo final de simulación, los nodos monitoreados y el valor lógico vigente de éstos nodos en el arreglo Antigo. Para facilitar la interpretación de resultados, la pantalla se divide en columnas de diez unidades de tiempo cada una, con su respectivo encabezado. En esta subrutina se verifica si la pantalla está llena; es decir, si se concluyó el análisis de setenta unidades de tiempo, y en caso negativo se procede a incrementar el contador de Tiempo de Simulación (T), para repetir el proceso de la Malla Principal de Simulación en su totalidad. En caso de que la pantalla esté llena, el sistema cuestiona al usuario si desea continuar o recomenzar con la corrida del programa. En caso de que se desee continuar, el programa toma como tiempo inicial al tiempo indicado en la corrida anterior como tiempo final de simulación más uno y continúa con el proceso. Si no se desea continuar, se imprime el mensaje "FIN DEL PROCESO DE SIMULACION" y el proceso se detiene. En caso de que se desee recomenzar, el sistema cuestiona nuevamente por los puntos a monitorear e inicia en este punto el proceso nuevamente.

### III.7. MODULO DE SALIDA.

-----

El módulo de salida proporciona los resultados obtenidos en el proceso de simulación de tal forma que sean fácilmente analizados y comprendidos por el usuario.

El uso de una pantalla de video permite monitorear varios nodos simultáneamente; sin embargo, el número de nodos que es posible monitorear, está limitado a diez debido al dimensionamiento del vector M. Si se desea aumentar este número, refiérase a la Sec. V.4. Con esta misma finalidad se ha limitado el análisis de tiempo a 70 unidades, cantidad que también puede modificarse manipulando la variable P en la subrutina de impresión de salidas.

Con el fin de facilitar la observación del resultado, la pantalla se ha dividido en columnas de diez unidades de tiempo cada una con su respectivo encabezado.

Con respecto a los caracteres impresos, éstos son:

- "\_" Valor Lógico Cero
- "X" Valor Lógico Uno
- "?" Valor Lógico Indeterminado

Cada uno de éstos se puede cambiar dependiendo de las necesidades del usuario. Así por ejemplo, si se desea obtener la salida con los símbolos utilizados en las tablas de Verdad, se cambiarían los caracteres de impresión por "0", "1" y "?" respectivamente. Este cambio se realiza en la sección del



programa denominada "Variables Usadas".

En cuanto a las leyendas que aparecen en el formato de salida éstas son: "TIEMPO INICIAL DE SIMULACION", que indica el tiempo en el que se inició el análisis; "SENALES", que proporciona el número de nodo monitoreado; "TIEMPO FINAL DE SIMULACION", que es el tiempo actual en el proceso de simulación. Al finalizar aparece la opción: "QUIERE CONTINUAR? (S)I, (N)O, (R)ECOMENZAR". Si la respuesta es "S" continúa con el análisis de tiempo a partir del tiempo final de simulación anterior. Si la respuesta es "N" imprime la leyenda "FIN DEL PROCESO DE SIMULACION" y por último si la respuesta es "R" se reinicia el proceso a partir de la petición de señales a monitorear.

El simulador básico implementado en el presente capítulo se denomina SIM/CD versión 1.0. Las instrucciones para su uso se encuentran en el Manual de Usuario (Sec. V.2), y el listado del programa BASIC en el Apéndice A.

En el Capítulo siguiente se implementan algunos conceptos que aumentan la capacidad de SIM/CD.

EXTENSION A LAS CAPACIDADES DEL SIMULADOR

IV.1. INTRODUCCION.

El Simulador descrito en el Capitulo anterior (SIM/CD versión 1.0), es un sistema completo y eficiente. Sin embargo, es posible aumentar sus capacidades para hacerlo mas fácil de usar, mas rápido y mas versátil.

Para facilitar su uso, es necesario mejorar la entrada de datos; es decir, hacer mas sencillo para el usuario la introducción de los datos descriptivos del circuito a simular. Una buena forma de lograr lo anterior es mediante gráficos para describir los elementos lógicos utilizados, y usando "Pantallas de Captura" para describir los estímulos externos y los nodos a monitorear.

Para aumentar la rapidez del proceso de simulación, es necesario implementar el concepto de "Trazo Selectivo", descrito en la Sec. II.3.3. Simulando únicamente los elementos que tuvieron un cambio de estado en sus entradas se incrementa la rapidez del simulador de dos a diez veces.

Para simular elementos lógicos más complejos, aumentando la versatilidad del sistema, es posible incorporar circuitos predefinidos o "Macroelementos". Un Macroelemento representa a un circuito complejo tal como un Flip-Flop, un Registro de Corrimiento o un Multiplexor, en términos de las primitivas del simulador.

En el presente Capítulo se incorporan al simulador básico, la entrada de datos utilizando gráficos, pantallas de captura, y trazo selectivo, dando por resultado a SIM/CD en su versión 2.0. Se describe también en este Capítulo la utilización de Macroelementos (Sec. IV.5) para ser implementados por el lector interesado.

#### IV.2. ENTRADA DE DATOS UTILIZANDO GRAFICOS.

Una de las características que ayudan apreciablemente la comunicación hombre-máquina es el uso de gráficos, que son utilizados ampliamente en los sistemas por microcomputadora debido a la facilidad de interpretación de información.

En el presente trabajo se han empleado gráficos para facilitar la captura de la descripción lógica del circuito a simular. Cada uno de los elementos manejados por SIM/CD se representa en forma gráfica en la pantalla de video. De esta manera, es muy sencillo para el usuario seleccionar el tipo de compuerta que desea simular.

Como se describió en el Capítulo anterior, en la versión

básica de SIM/CD (Versión 1.0), se describe un elemento por medio de una letra que identifica el tipo de compuerta, seguida de los números de los nodos conectados a las 5 entradas posibles separados por comas, y finalmente los nodos conectados a las salidas normal y negada también separados por comas.

En la Versión 2.0 de SIM/CD se hace uso de gráficos para la descripción lógica de la siguiente manera :

Se presentan en la pantalla los gráficos correspondientes a todos los elementos manejados por el simulador. Debajo de cada gráfico aparece un campo de captura en forma de corchete donde el usuario puede introducir un número indicando las entradas que contiene el elemento a describir. El cursor aparece en el primer gráfico (esquina superior izquierda de la pantalla). Para seleccionar alguno de los elementos que aparecen en la pantalla, el usuario simplemente debe posicionar el cursor en el corchete correspondiente utilizando las flechas en el teclado.

Una vez seleccionado el elemento, aparece una nueva pantalla con el gráfico correspondiente y campos de captura para los nodos conectados a las entradas y a las salidas. Al finalizar la captura, se cuestiona al usuario si los datos han sido introducidos en forma correcta. Si la respuesta es afirmativa, se almacenan los datos correspondientes en el arreglo L. En caso negativo, se ignoran los datos introducidos y se presenta la pantalla de gráficos nuevamente.

Para dar por terminada la fase de descripción lógica, basta con oprimir la tecla ESC (escape), y el sistema continuará con la

"Pantalla de Captura" de descripción de estímulos externos (Sec. IV.3 ).

Es importante señalar que la subrutina de descripción lógica contiene instrucciones que detectan posibles datos incorrectos. Así por ejemplo, los corchetes de selección de elemento no aceptan otro dato que no sea un número del 1 al 5, para el caso de las compuertas lógicas, y un 1 para el inversor.

La Figura 4.1 muestra la pantalla de "Descripción de Elementos Lógicos", que contiene los gráficos de cada tipo de elemento manejado por SIM/CD. La Figura 4.2 muestra lo que aparece en la pantalla después de haber seleccionado una compuerta NAND de 3 entradas.

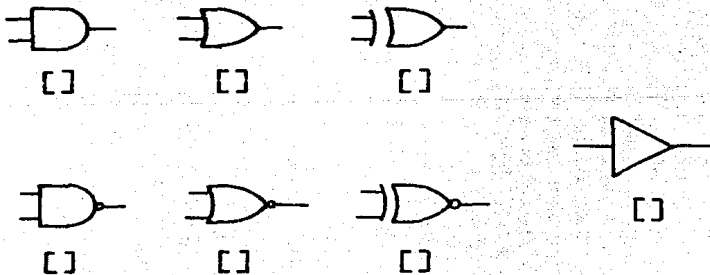


Fig. 4.1 Descripción de los Elementos Lógicos en SIM/CD 2.0



E 1 : \_\_\_

SAL : \_\_\_

E 2 : \_\_\_

E 3 : \_\_\_

---

DATOS OK? (S)I , (N)O ---> \_\_\_

---

Fig. 4.2 Entrada de Datos para una Compuerta NAND con 3 Entradas

---

#### IV.3. PANTALLAS DE CAPTURA.

-----

Una pantalla de captura crea un diálogo entre el simulador y el usuario: el simulador (programa) solicita al usuario la introducción de un dato por medio de un mensaje comprensible en la pantalla; y el usuario responde introduciendo el dato requerido por medio del teclado.

El uso de pantallas de captura para describir los Estímulos

Externos y los Nodos a Monitorear, hacen de SIM/CD Versión 2.0 un sistema mas interactivo .

Las Figuras 4.3 y 4.4 muestran las pantallas de captura para la descripción de señales y la selección de nodos a monitorear respectivamente.

DESCRIPCION DE ESTIMULOS EXTERNOS

-----  
INTRODUZCA O PARA TERMINAR CAPTURA

SEÑAL 1 .

NUMERO DE NODO: \_\_\_\_\_

VALOR INICIAL: \_\_\_\_\_

TIEMPO INICIAL: \_\_\_\_\_

TIEMPO 1er CAMBIO: \_\_\_\_\_

TIEMPO 2o CAMBIO: \_\_\_\_\_

TIEMPO 3er CAMBIO: \_\_\_\_\_

TIEMPO 4o CAMBIO: \_\_\_\_\_

TIEMPO 5o CAMBIO: \_\_\_\_\_

-----  
DATOS OK? (S)I, (N)O \_\_\_\_\_

Fig. 4.3 Pantalla de Captura de Estímulos Externos

DESCRIPCION DE NODOS A MONITOREAR

-----  
INTRODUZCA O PARA TERMINAR CAPTURA

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

-----  
DATOS OK? (S)I, (N)O \_\_\_\_

Fig. 4.4 Pantalla de Captura de Nodos a Monitorear

IV.4. TRAZO SELECTIVO.  
-----

Otra característica importante incorporada en la versión 2.0 de SIM/CD, es la utilización de trazo selectivo o actividad direccionada. Como se describió en la Sección II.3.3, en la simulación de un sistema digital solamente cambian de estado de un 2% a un 10% de los nodos.



Tomando en cuenta que la salida de un elemento lógico no cambia de estado a menos que aparezca un cambio en alguna de sus entradas, es posible optimizar el tiempo de corrida efectuando solamente la simulación de aquellos elementos que tuvieron un cambio en sus entradas. En términos de la simulación global, ningún elemento debe de ser simulado si no ha sido aplicada ninguna señal externa.

La optimización global puede ser implementada por medio de una Bandera de Cambio Global BCG, que se mantiene apagada mientras no se detecte algún cambio de estado en los nodos. La detección puede realizarse al transferir los nodos del vector N al vector A. De esta manera, el proceso de simulación puede ser omitido completamente (para el tiempo de simulación actual) si BCG se encuentra apagada.

La implementación del trazo selectivo en SIM/CD incluye el uso de tres vectores. El vector TS1 contiene una bandera que indica, para cada elemento lógico, si alguna de sus entradas ha sufrido un cambio de estado. El vector TS2 se encuentra ordenado por número de señal, es decir un nodo determinado tiene su posición correspondiente en el vector. Los elementos de TS2 contienen un apuntador al tercer vector TS3, que tiene la lista de los elementos manejados por la señal correspondiente.

Los tres vectores anteriores son construidos por medio de la subrutina "Construcción de la Tabla de Trazo Selectivo". Esta tabla es similar a una Lista Encadenada.

Durante la simulación, mientras se transfieren las señales del vector Antiguo al vector Nuevo, se checa si ha ocurrido un cambio en una señal, en cuyo caso se levanta la bandera de cambio global (BCG), así como las banderas locales (TSI) de los elementos manejados por la señal en cuestión. Al efectuar las funciones lógicas, se prueba el estado de la bandera del elemento a simular. Si la bandera es 1, entonces ha ocurrido un cambio de estado en alguna de las entradas del elemento, por lo que éste debe ser simulado. Si la bandera es cero, ningún cambio de estado ocurrió en sus entradas, y la simulación es inhibida.

La simulación con trazo selectivo puede incrementar de dos a diez veces la velocidad de simulación.

#### IV.5. MACROELEMENTOS.

-----

La utilización de Macroelementos incrementa la capacidad de un sistema simulador de sistemas digitales, haciendo posible la simulación de circuitos mas complejos. La versión 2.0 de SIM/CD no incorpora el uso de Macroelementos; sin embargo, en esta sección se describe como puede implementarse esta facilidad.

Un Macroelemento puede considerarse como una caja negra con una o varias entradas y salidas, como se muestra en la Figura 4.5



Fig. 4.5 Macroelemento visto como Caja Negra

Un Macroelemento --circuito digital complejo-- puede ser descrito por medio de las primitivas del simulador. Como se mencionó en la Sección III.2.1, SIM/CD contiene solamente seis primitivas: AND, OR, XOR, NAND, NOR y XNOR; sin embargo, con estas compuertas es posible construir cualquier circuito, por mas complejo que pueda ser. Algunos de los circuitos comerciales mas comúnmente utilizados que pueden ser implementados como Macroelementos son: Flip-Flops, Contadores, Registros de Corrimiento, y Multiplexores.

Para definir un Macroelemento en términos de las primitivas, es necesario proveer al simulador la siguiente información:

1. Compuertas Lógicas requeridas (Primitivas).
2. Conexiones Internas de las Primitivas.
3. Conexiones Externas del Macroelemento.

Las compuertas lógicas requeridas pueden ser descritas de la misma manera como se hace para un elemento convencional en SIM/CD: Tipo de Compuerta, cinco entradas y dos salidas.

Las conexiones internas de las primitivas no corresponden a ninguna de las terminales del Macroelemento, por lo que deben de tener números de nodos "internos", que sean transparentes para el usuario.

Las conexiones externas del Macroelemento corresponden a las terminales de entrada y salida de éste y deben de tener los mismos números de nodos proporcionados por el usuario.

Debido a que un Macroelemento constituye un circuito digital complejo que puede ser utilizado extensamente en un diseño dado, es necesario que sea definido en forma genérica como una Subrutina en el Simulador. De esta manera, cada vez que el usuario introduzca un Macroelemento, éste puede ser "expandido" en términos de las primitivas por medio de la subrutina correspondiente.

Como ejemplo de la definición de un Macroelemento típico, en la Figura 4.6 se muestra un Flip-Flop tipo JK.

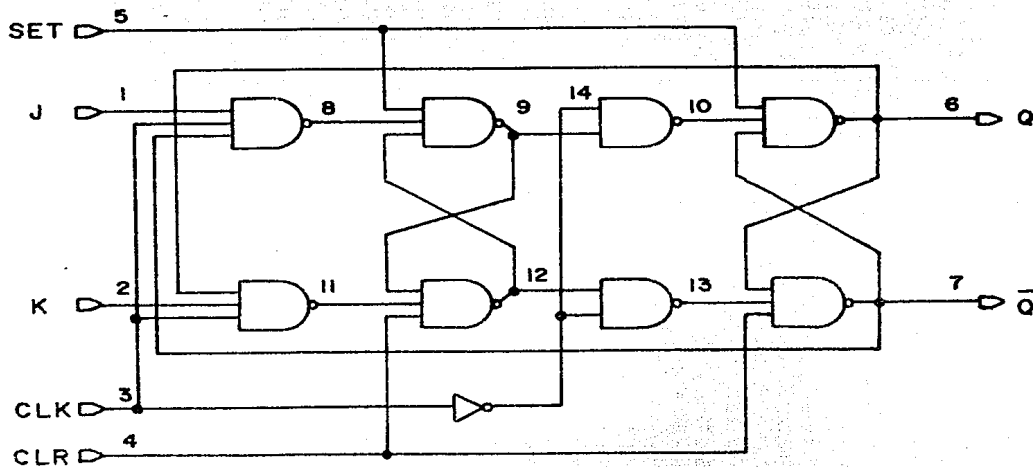


Fig 4.6 (a) Diagrama de un Flip-Flop JK

```

FFJK,1,2,3,4,5,6,7
NAND,1,3,7,0,0,0,8
NAND,2,3,6,0,0,0,11
NAND,5,8,12,0,0,0,9
NAND,4,11,9,0,0,0,12
NAND,9,14,0,0,0,0,10
NAND,12,14,0,0,0,0,13
NAND,10,7,5,0,0,0,6
NAND,13,6,4,0,0,0,7
INV,3,0,0,0,0,0,14
FIN,0,0,0,0,0,0,0

```

Fig. 4.6 (b) Definición de un Macroelemento Flip-Flop JK

La expansión de Macroelementos puede ser realizada por una subrutina de "Expansión de Macros", la cual debe efectuar las siguientes funciones principales:

1. Traducir las entradas y salidas del Macroelemento genérico por los números de nodos proporcionados por el usuario.
2. Generar números de nodos internos de manera tal que no existan duplicados con nodos internos de otros Macroelementos o bien con nodos introducidos por el usuario.
3. Introducir en la Tabla de Descripción Lógica (arreglo L) las primitivas resultantes de la expansión, conteniendo los números de nodos asignados en los puntos 1 y 2.

Los conceptos implementados en el presente capítulo facilitan la operación del simulador de circuitos digitales SIM/CD, y mejoran algunas de sus características tal como la rapidez del proceso de simulación. En la Sección V.4 se describen otras modificaciones que el usuario puede realizar fácilmente para adaptar el programa a sus necesidades.

MANUAL DE USUARIO

V.1. INTRODUCCION.

La Tesis persigue la finalidad de que SIM/CD sea interactivo; esto es, que permita establecer una relación amable entre el usuario y la máquina. Dicho en otras palabras, el objetivo es que el usuario sea guiado por el sistema para la operación del mismo.

En el presente Capítulo se dan las instrucciones necesarias para la operación de SIM/CD en su versión básica (v. 1.0) y avanzada (v. 2.0); y se mencionan algunas alternativas para hacer los cambios más comúnmente deseados.

Para la operación de SIM/CD el usuario debe: conocer la computadora a utilizar; tener presente que el sistema operativo utilizado es MS-DOS; y que el simulador está escrito en BASIC.

Para una fácil manipulación de la presente sección, la secuencia de instrucciones a teclear aparece subrayada y la leyenda "comentario" implica un mensaje en la pantalla, o bien como se debe de responder a una interrogante del sistema. Para

facilitar la comprensión del manual, se describe el ejemplo de la simulación de un "sumador completo" paso a paso.

## V.2. OPERACION DE SIM/CD VERSION 1.0.

-----

En esta sección se describe el procedimiento para introducir los datos necesarios para una corrida en el simulador básico.

### V.2.1. Descripción Lógica del Circuito a Simular.

-----

En primer lugar, se deben numerar cada uno de los nodos constitutivos del circuito lógico a simular, partiendo del número 1, en orden ascendente, y cuidando de no repetir ningún número como se aprecia en la Figura 5.1.

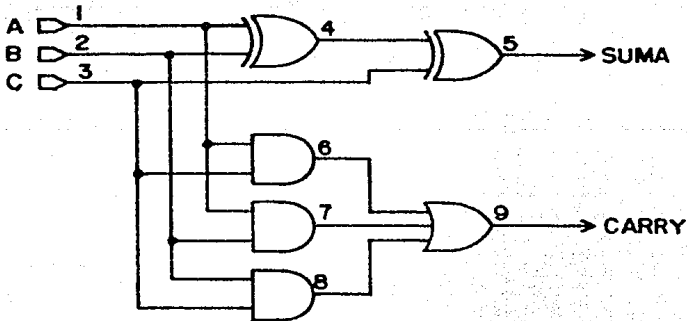


Fig. 5.1 Diagrama Lógico de un Sumador Completo.

Una vez numerados los nodos del circuito, se procede a cargar el simulador en la memoria de la computadora de la siguiente manera:

1. Insertar Diskette conteniendo los siguientes archivos:
  - Sistema Operativo MS-DOS
  - Intérprete BASIC
  - Sistema Simulador SIM/CD1
2. Cargar el sistema operativo e introducir la fecha y hora en el formato indicado en la pantalla.
3. Cargar el intérprete BASIC tecleando el siguiente comando:

```
BASIC <return>
-----
```
4. Cargar el programa simulador con el comando:

```
LOAD "SIM/CD1" <return>
-----
```
5. Ejecutar el programa tecleando:

```
RUN <return>
---
```

Comentario: Aparece en la pantalla el siguiente mensaje:

```
SISTEMA BASICO DE SIMULACION LOGICA
INTRODUZCA LA DESCRIPCION LOGICA DEL CIRCUITO
?
```

Comentario: Frente al signo de interrogación se deben describir cada una de las compuertas de acuerdo al siguiente formato:

```
TIPO,ENT1,ENT2,ENT3,ENT4,ENT5,SAL,SALNEG
```



En TIPO se debe introducir una de las siguientes letras:

A , O , X

que definen a las compuertas AND, OR, y XOR respectivamente.

En los campos comprendidos entre ENT1 y ENT5, se deben de introducir los números de los nodos conectados a las entradas de la compuerta que se está describiendo. En caso de que no se ocupen todas las entradas, los espacios deben de permanecer libres.

En el campo de SAL (Salida) se coloca el número correspondiente al nodo conectado a la salida de la compuerta descrita. En el campo de SALNEG (Salida Negada) se coloca el número de nodo correspondiente a la salida de la compuerta descrita siempre y cuando esta sea del tipo NAND, NOR y XNOR.

Como se observa, solo uno de los campos reservados para la salida será ocupado, mientras que el otro debe de permanecer libre.

Los elementos descriptivos de la compuerta deben de ir separados entre sí por medio de comas (,).

Al finalizar la descripción de una compuerta, automáticamente aparece otro signo de interrogación, en donde debe describirse la siguiente compuerta en la misma forma. En caso de haberse descrito todas las compuertas, se debe teclear la siguiente secuencia:

FIN,,,,,  
-----

Esto da origen a que se continúe la ejecución del programa

con la aparición de un nuevo mensaje.

En la Figura 5.2 se observa la descripción lógica del circuito de la Figura 5.1.

? X,1,2,,,4,  
-----

? X,3,4,,,5,  
-----

? A,1,3,,,6,  
-----

? A,1,2,,,7,  
-----

? A,2,3,,,8,  
-----

? 0,6,7,8,,,9,  
-----

? FIN,,,,,  
-----

Fig. 5.2 Descripción Lógica del Sumador Completo mostrado en la figura 5.1

---

#### V.2.2. Descripción de Estímulos Externos.

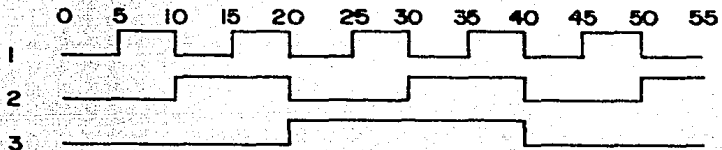
Después de describir las compuertas del circuito a simular, aparece en la pantalla el siguiente mensaje:

DESCRIBA LAS SEÑALES DE ENTRADA AL CIRCUITO  
?

Comentario: Frente al signo de interrogación se debe describir el diagrama de tiempo de la señal de entrada de acuerdo al siguiente



En el ejemplo 5.3 se muestra un diagrama de tiempo de la señal de entrada aplicada al circuito sumador completo y la manera en que debe ser descrita.



DESCRIBA LAS SEÑALES DE ENTRADA AL CIRCUITO

? 1, 0, 0, 5, 10, 15, 20, 25

? 1, 0, 30, 35, 40, 45, 50, 55

? 2, 0, 0, 10, 20, 30, 40, 50

? 3, 0, 0, 20, 40, 60, 80, 100

? 0, , , , , ,

Fig. 5.3. Diagrama de Tiempo y Descripción de las Señales de Entrada

### V.2.3. Descripción de las Señales a Monitorear.

Después de describir los estímulos externos, aparece en la pantalla el siguiente mensaje:

INTRODUZCA LOS NODOS A MONITOREAR

?

Comentario: Frente al signo de interrogación se debe de colocar el número de nodo del cual se desea conocer su comportamiento en el tiempo. Al introducir un número, automáticamente da origen a la aparición de un nuevo signo de interrogación; si no se desean introducir mas nodos, se teclea un 0, que da por terminada la captura de puntos a monitorear, dando inicio al proceso de simulación.

En la Figura 5.4 se muestra como se deben de introducir los nodos a monitorear para el circuito sumador completo de la Figura 5.1.

INTRODUZCA LOS NODOS A MONITOREAR

? 1

-

? 2

-

? 3

-

? 5

-

? 9

-

? 0

-

Fig. 5.4. Descripción de los Nodos a Monitorear.

---

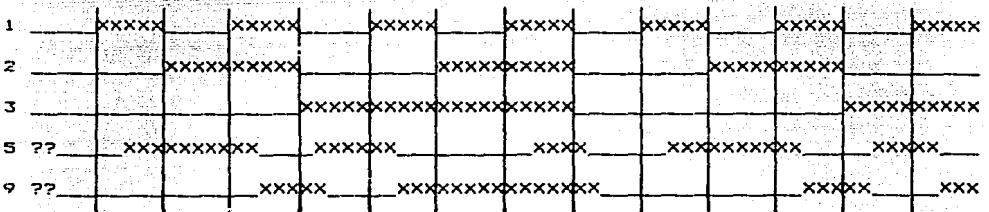
V.2.4. Proceso de Simulación.

Al finalizar la captura de datos (Descripción Lógica, Señales de Entrada, y Nodos a Monitorear), se da inicio

automáticamente al proceso de simulación, apareciendo en pantalla lo siguiente:

TIEMPO FINAL DE SIMULACION 70  
 TIEMPO INICIAL DE SIMULACION 0

SENALES



QUIERE CONTINUAR? (S)I, (N)O, (R)ECOMENZAR.

? N

FIN DEL PROCESO DE SIMULACION

Ready

El comportamiento real de un Sumador Completo se ilustra en el diagrama de tiempo de la figura 5.5

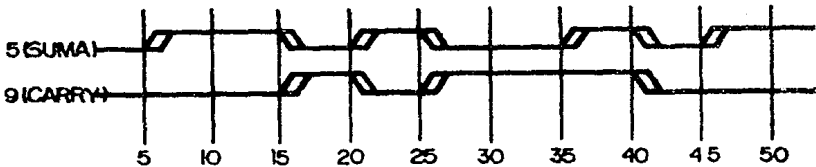


Fig. 5.5 Diagrama de Tiempo de un Sumador Completo

Como puede observarse, los resultados obtenidos por el Simulador son los mismos a los esperados en un sistema real, lo cual indica que SIM/CD cumple satisfactoriamente con los objetivos por los que fué diseñado.

Comentario: Existen tres posibles respuestas a la interrogante QUIERE CONTINUAR? , que aparece al completar una pantalla con los resultados del proceso simulador. La primera posible respuesta es S (si), que permite continuar con el proceso simulador a partir del último tiempo de simulación analizado, hasta evaluar las siguientes 70 unidades de tiempo. La segunda posibilidad es N (no) que da por concluido el proceso de simulación apareciendo en pantalla el mensaje FIN DEL PROCESO DE SIMULACION, y la microcomputadora queda bajo el control del intérprete BASIC, lo cual es indicado por medio del mensaje Ready. La tercera posible respuesta es R (recomenzar) que permite la introducción de nuevos puntos a monitorear utilizando el mismo circuito descrito.

Es conveniente señalar que es posible detener el programa simulador en cualquier momento por medio de la tecla BREAK o en su defecto oprimiendo simultáneamente las teclas CTRL (control) C. Esto interrumpe cualquier instrucción que se esté ejecutando en ese instante, sin afectar en manera alguna al Simulador. Sin embargo, una interrupción de este tipo hace necesario volver a correr (RUN) SIM/CD y comenzar desde la Descripción Lógica.

## V.2.5 Mensajes de Error.

---

En cuanto a los mensajes de error, resulta conveniente señalar la diferencia entre los mensajes producidos por SIM/CD y los mensajes originados por el intérprete BASIC. Los primeros son mensajes que facilitan la operación de SIM/CD y los últimos son generados por el intérprete BASIC cuando no se cumple con el formato preestablecido para la introducción de datos. Los mensajes de ambos tipos pueden aparecer por separado o en combinación.

### Mensajes generados por SIM/CDI.

---

#### 1. DEMASIDOS ELEMENTOS LOGICOS.

El número de elementos lógicos que SIM/CDI puede manejar fué excedido. Si se desea manejar un número mayor de elementos, refiérase a la Sección V.4.

#### 2. NUMERO DE NODO NO VALIDO.

El número máximo de nodos excede el límite definido en el simulador. Si se requiere manejar un número mayor de nodos refiérase a la sección V.4.

#### 3. TIPO DE COMPUERTA NO VALIDO.

Un caracter extraño, distinto a los tres tipos permitidos por el formato, fué introducido en la Descripción Lógica. Al aparecer este mensaje automáticamente se pide la descripción de la compuerta que se describió en forma errónea.



#### 4. DEMASIADOS ESTIMULOS EXTERNOS.

El número de estímulos externos (señales) que es posible manejar por SIM/CD, fué excedido . Si se desea introducir un mayor número de señales refiérase a la Sección V.4.

Mensaje generado por el intérprete BASIC.

-----

El siguiente mensaje aparece cuando un dato no cumple con el formato esperado por el intérprete BASIC:

Redo from start

Al aparecer este mensaje, el dato introducido se ignora y es necesario introducirlo nuevamente.

Los formatos correctos para la descripción de datos se indican en las Secciones V.2.1, V.2.2 y V.2.3.

#### V.3 OPERACION DE SIM/CD VERSION 2.0

-----

Como se mencionó en el capítulo anterior la necesidad de facilitar la captura de datos, así como la necesidad de optimizar el proceso simulador, dan origen a la versión SIM/CD 2.0.

Para facilitar la introducción de datos se hace uso de gráficos y pantallas de captura. En cuanto a la optimización del proceso de simulación, se utiliza el trazo selectivo.

En esta sección se describe la utilización de la versión avanzada de SIM/CD en cuanto a la captura de datos, ya que la

utilización del Trazo Selectivo se efectúa en forma interna.

### V.3.1 Descripción de Elementos Lógicos.

---

En la Versión 2.0 de SIM/CD se hace uso de gráficos para la descripción lógica de la siguiente manera :

1. Se presentan en la pantalla los gráficos correspondientes a todos los elementos manejados por el simulador.
2. Debajo de cada gráfico aparece un campo de captura en forma de corchete donde el usuario puede introducir un número indicando las entradas que contiene el elemento a describir.
3. El cursor aparece en el primer gráfico (esquina superior izquierda de la pantalla). Para seleccionar alguno de los elementos que aparecen en la pantalla, el usuario simplemente debe posicionar el cursor en el corchete correspondiente utilizando las flechas en el teclado.
4. Una vez seleccionado el elemento, aparece una nueva pantalla con el gráfico correspondiente y campos de captura para los nodos conectados a las entradas y a las salidas.
5. Al finalizar la captura, se cuestiona al usuario si los datos han sido introducidos en forma correcta. En caso negativo, se ignoran los datos y se regresa al punto 1.
6. Para dar por terminada la fase de descripción lógica, basta con oprimir la tecla ESC (escape), y el sistema continuará con la "Pantalla de Captura" de descripción de estímulos externos.

Es importante señalar que la subrutina de descripción lógica contiene instrucciones que detectan posibles datos incorrectos. Así por ejemplo, los corchetes de selección de elemento no aceptan otro dato que no sea un número del 1 al 5, para el caso de las compuertas lógicas, y un 1 para el inversor.

### V.3.2 Descripción de Estímulos Externos.

---

Para describir los estímulos externos, SIM/CD provee de una pantalla de captura como la mostrada en la Fig. 5.6

Como puede observarse, la pantalla de captura cuestiona al usuario la introducción de cada uno de los datos por medio de mensajes descriptivos de las señales externas:

Esta pantalla de captura contiene los mismos datos introducidos en la Descripción de Estímulos Externos de la versión 1.0 (Sec. V.2.2), y además, estos se encuentran en el mismo orden. En cada campo de la pantalla de captura debe de escribirse el dato solicitado por el mensaje. El mensaje DATOS OK? debe ser respondido con S (sí) si los datos introducidos son correctos y con N (no) si se ha cometido algún error y se desea volver a introducir el mismo dato.

## DESCRIPCION DE ESTIMULOS EXTERNOS

INTRODUZCA O PARA TERMINAR CAPTURA

SEÑAL 1 .

NUMERO DE NODO: \_\_\_\_\_

VALOR INICIAL: \_\_\_\_\_

TIEMPO INICIAL: \_\_\_\_\_

TIEMPO 1er CAMBIO: \_\_\_\_\_

TIEMPO 2o CAMBIO: \_\_\_\_\_

TIEMPO 3er CAMBIO: \_\_\_\_\_

TIEMPO 4o CAMBIO: \_\_\_\_\_

TIEMPO 5o CAMBIO: \_\_\_\_\_

---

DATOS OK? (S)I, (N)O \_\_\_\_\_

Fig. 5.6 Pantalla de Captura de Estimulos Externos

### V.3.3 Descripción de Nodos a Monitorear.

---

Para describir los nodos a monitorear, SIM/CD provee una pantalla de captura como la mostrada en la Fig. 5.7

DESCRIPCION DE NODOS A MONITOREAR

-----  
INTRODUZCA O PARA TERMINAR CAPTURA

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

NODO NUMERO: \_\_\_\_

-----  
DATOS OK? (S)I , (N)O \_\_\_\_

Fig. 5.7 Pantalla de Captura de Nodos a Monitorear

Como puede observarse, la secuencia de los mensajes en esta pantalla de captura es la misma que en la Descripción de Nodos a Monitorear de la versión 1.0. La forma de introducir los nodos a monitorear en SIM/CD 2.0, es escribiendo el número deseado a continuación del mensaje NODO NUMERO: , o bien escribiendo O para dar por concluida la captura. El mensaje DATOS OK? debe ser respondido con S (sí) en caso de que los datos introducidos sean correctos; y con N (no) si los datos no son correctos, en cuyo

caso estos son ignorados y deben ser re-introducidos.

#### V.4 ALTERNATIVAS.

-----

En esta sección se describen algunas modificaciones para que el usuario adecúe SIM/CD a sus necesidades.

Estas modificaciones facilitan la interpretación de resultados y el manejo de una gran cantidad de compuertas para un circuito lógico muy extenso, permitiendo dimensionar el número de elementos lógicos y el número de estímulos externos de acuerdo a las necesidades del usuario y a la capacidad de la máquina utilizada.

Para realizar los cambios aquí descritos se pretende que el usuario tenga conocimientos del lenguaje BASIC.

##### V.4.1 Cambio de Símbolos Impresos.

-----

Para determinados fines, por ejemplo, si se desea que el conteo de estados lógicos sea rápido, o que sea más legible la presentación, o sencillamente que sea más estética, se hace necesario el cambio de símbolos impresos para los estados lógicos. Por esta razón se sugiere la utilización de los siguientes símbolos para los estados lógicos:

Simbolo Impreso

Estado Lógico

■ , X , 1

UNO

— , " " , .

CERO

? , u

INDETERMINADO

Sin embargo la utilización de los símbolos va de acuerdo a las necesidades y a la imaginación del usuario. Estos cambios se logran de una manera muy sencilla; En la sección de "Variables Usadas" se define el símbolo impreso para los valores lógicos, la única acción necesaria es intercambiar estos por los símbolos deseados.

V.4.2 Almacenamiento de un circuito dentro del programa.

Cuando se efectúa la simulación de un circuito muy extenso resulta muy laboriosa la introducción de los datos del circuito. Mas aún, si el análisis del circuito no se realiza en una sola sesión, resulta por demás impráctico realizar la captura de datos cada que sea necesario.

La solución consiste en un cambio de la instrucción INPUT por la instrucción READ, en las secciones "Lectura de la Descripción Lógica", "Lectura de Estimulo Externo" y "Lectura de Nodos a Monitorear". Los datos se colocan al final del programa, precedidos por la instrucción DATA, en la secuencia normal en la que se introducen para una corrida convencional y en la cantidad de líneas que sean necesarias. Con esta operación es posible efectuar el análisis de un circuito muy extenso en varias sesiones sin la necesidad de introducir los datos en cada una de

ellas.

#### V.4.3 Aumento o Disminución de la capacidad de SIM/CD.

-----

Como se mencionó anteriormente la capacidad actual de SIM/CD es la simulación de circuitos que contengan como máximo 100 compuertas, 100 nodos, 30 estímulos externos y 10 nodos a monitorear. Sin embargo puede ser necesario, de acuerdo a las necesidades del usuario o a la capacidad de la máquina utilizada, aumentar o disminuir la capacidad del simulador.

Resulta conveniente señalar que éste cambio en la capacidad del simulador no afecta el tiempo de máquina utilizado para simular un circuito, ya que la optimización del tiempo de máquina se efectúa gracias a la utilización de las variables ULTELE, ULTEXT y MAXNOD; Este cambio en la capacidad de SIM/CD únicamente optimiza la cantidad de memoria disponible en la máquina utilizada.

Para realizar un cambio en el máximo número de nodos a simular, o en el máximo número de elementos a simular es necesario cambiar la constante MAXNOD; en caso de que se desee modificar el número máximo de estímulos externos es necesario modificar la constante MAXEXT, y por último si se desea modificar el número de nodos a monitorear es necesario modificar el dimensionamiento del arreglo M.

Los tres cambios descritos deben efectuarse en la sección "Estructura de Datos" y son suficientes para aumentar o disminuir la capacidad de simulación de SIM/CD.



## CONCLUSIONES

-----

En el presente trabajo se dan las bases teóricas del diseño y funcionamiento de los Simuladores de Sistemas Digitales, y se implementan dos versiones de Simuladores SIM/CD que pueden ser utilizados tanto en las Universidades como en la Industria, debido a la facilidad de operación y a la confiabilidad de los resultados.

Las aplicaciones de SIM/CD se enfocan a la evaluación de la operación lógica en el diseño de un nuevo circuito digital, reduciendo las dificultades y los costos para construir un prototipo.

SIM/CD puede ser fácilmente utilizado por estudiantes y diseñadores de circuitos, aún cuando estos carezcan de conocimientos en computación. Lo anterior es posible debido a que el simulador es de tipo interactivo puesto que permite establecer un diálogo entre el usuario y la computadora para su operación. Además, el simulador está estructurado de tal forma que permite al usuario con conocimientos en computación, entender el programa, modificarlo, y mejorarlo.

El modelo en el que se basa SIM/CD considera los elementos mas importantes de un sistema digital real, y es válido en predicción puesto que permite obtener con fidelidad los resultados que se obtendrían utilizando un prototipo.

La existencia de SIM/CD proporciona al estudiante y al diseñador experimental gran experiencia en la operación y el entendimiento de los dispositivos y sistemas digitales, así como en el diseño por computadora que es la tendencia actual en la ingeniería.

APENDICE A. LISTADO DE SIM/CD VERSION 1.0

```

100 REM *****
110 REM *           SIMULADOR DE CIRCUITOS DIGITALES   SIM/CD   VERSION 1.0   *
120 REM *
130 REM * ELABORADO POR:
140 REM *           OSCAR DAVILA MONROY
150 REM *           ROGELIO CISNEROS GUADARRAMA
160 REM *
170 REM *****
180 REM
190 REM *****
200 REM *           UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
210 REM *           FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
220 REM *
230 REM * TESIS: SIMULACION INTERACTIVA DE CIRCUITOS DIGITALES
240 REM *           AUXILIADA POR COMPUTADORA.
250 REM *
260 REM * FECHA: 21/AGO/87
270 REM *****
280 REM
290 DEFINT A-Z
300 CLS
310 PRINT "SISTEMA BASICO DE SIMULACION LOGICA"
320 REM
330 REM ----- ESTRUCTURAS DE DATOS -----
340 REM
350 MAXNOD=100           ' MAXIMO NUMERO DE NODOS
360 MAXEXT= 30          ' MAXIMO NUMERO DE ESTIMULOS EXTERNOS
370 DIM L(7,MAXNOD)    ' ARREGLO DE ELEMENTOS LOGICOS
380 DIM E(7,MAXEXT)    ' ARREGLO DE ESTIMULO EXTERNO
390 DIM M(10)          ' ARREGLO DE SENAL A MONITOREAR
400 DIM A(MAXNOD)      ' ARREGLO DE LA SENAL ANTIGUA
410 DIM N(MAXNOD)      ' ARREGLO DE LA SENAL NUEVA
420 REM
430 REM ----- VARIABLES USADAS -----
440 REM
450 LO = 0              ' REPRESENTACION INTERNA DE CERO LOGICO
460 L1 = 1              ' REPRESENTACION INTERNA DE UNO LOGICO
470 LX=3                ' REPRESENTACION INTERNA DE INDETERMINACION
480 I(LO)=L1           ' COMPLEMENTO DE CERO LOGICO
490 I(L1)=LO           ' COMPLEMENTO DE UNO LOGICO
500 I(LX)=LX           ' COMPLEMENTO DE INDETERMINACION LOGICA
510 S$(LO)="_"         ' SIMBOLO IMPRESO PARA CERO LOGICO
520 S$(L1)="X"        ' SIMBOLO IMPRESO PARA UNO LOGICO
530 S$(LX)="?"        ' SIMBOLO IMPRESO PARA INDETERMINACION LOGICA
540 FP=0                ' BANDERA DE FIN DE PAGINA
550 P=0                 ' IMPRESION DE UNA LINEA
560                     ' ULTELE=ULTIMO ELEMENTO,ULTEXT=ULTIMO ESTIMULO EXTERNO
570                     ' ULTNOD=NUMERO MAS ALTO DE NODO

```

```

580                               'L$=CODIGO LOGICO
590 REM
600 REM ----- LECTURA DE DESCRIPCION LOGICA -----
610 REM
620 I=-1
630 REM ENTRADA DE LA MATRIZ L(*,0) A L(*,ULTELE)
640 PRINT "INTRODUZCA LA DESCRIPCION LOGICA DEL CIRCUITO"
650 I=I+1
660 IF I>MAXNOD THEN GOTO 2340:      'DEMASIADAS COMPUERTAS
670 INPUT L$,L(1,I),L(2,I),L(3,I),L(4,I),L(5,I),L(6,I),L(7,I)
680 IF L$="FIN" THEN GOTO 730 :      'CHEQUEO DE FINALIZACION
690 GOSUB 2230
700 IF L(0,I)=0 THEN GOSUB 2400
710 IF ULTNOD>MAXNOD THEN GOSUB 2360
720 GOTO 650
730 ULTELE=I-1
740 REM
750 REM ----- LECTURA DE ESTIMULO EXTERNO -----
760 REM
770 I=-1
780 PRINT "DESCRIBA LAS SEÑALES DE ENTRADA AL CIRCUITO"
790 I=I+1
800 IF I>MAXEXT THEN GOTO 2430:      'DEMASIADOS ESTIMULOS EXTERNOS
810 INPUT E(0,I),E1,E(2,I),E(3,I),E(4,I),E(5,I),E(6,I),E(7,I)
820 REM CAMBIO DE VALOR INICIAL A FORMATO INTERNO
830 E(1,I)=LX
840 IF E1=0 THEN E(1,I)=L0
850 IF E1=1 THEN E(1,I)=L1
860 IF E(0,I)<>0 THEN GOTO 790 :      'CHEQUEO DE FINALIZACION
870 ULTEXT=I-1
880 REM
890 REM ----- LECTURA DE PUNTOS A MONITOREAR -----
900 REM
910 PRINT "INTRODUZCA LOS NODOS A MONITOREAR"
920 PRINT "(HASTA 10, 0 PARA TERMINAR)"
930 FOR I=1 TO 10
940     INPUT M(I)
950     IF M(I)=0 THEN GOTO 980
960 NEXT I
970 REM
980 REM ----- INICIALIZADO (COLOCACION DE NODOS EN LOGICA INDETERM) -----
990 REM
1000 FOR I=0 TO MAXNOD
1010     A(I)=LX      'ARREGLO DEL ANTIGUO VALOR
1020     N(I)=LX      'ARREGLO DEL NUEVO VALOR
1030 NEXT I
1040 T=-1
1050 REM
1060 REM =====
1070 REM MALLA PRINCIPAL DE SIMULACION
1080 REM =====
1090 REM
1100 T=T+1      : ' INCREMENTAR TIEMPO DE SIMULACION
1110 GOSUB 1340 : ' ACTUALIZAR ESTIMULO
1120 REM TRANSFERIR VECTOR NUEVO EN VECTOR ANTIGUO

```

```

1130 FOR I=1 TO ULTNOD
1140   A(I)=N(I)
1150 NEXT I
1160 GOSUB 1480      : ' FUNCIONES LOGICAS
1170 GOSUB 1880      : ' IMPRESION DE SALIDAS
1180 IF FP<>1 THEN GOTO 1100
1190 REM FIN DE LA PANTALLA
1200 LOCATE 25,1: PRINT "QUIERE CONTINUAR? (S)I, (N)O, (R)ECOMENZAR"
1210 INPUT A*
1220 IF A*="S" THEN GOTO 1100
1230 IF A*="R" THEN GOTO 910
1240 GOTO 2450
1250 REM
1260 REM =====
1270 REM              FIN DE MALLA PRINCIPAL DE SIMULACION
1280 REM =====
1290 REM
1300 REM ***** SUBROUTINAS *****
1310 REM
1320 REM ----- ACTUALIZAR ESTIMULO -----
1330 REM
1340 FOR I=0 TO ULTEXT
1350   FOR J=2 TO 7
1360     IF T<>E(J,I) THEN GOTO 1420
1370     ' MARCA DE TIEMPO, OBTENGA EL NUMERO DE SENAL
1380     X=E(O,I)
1390     XI=N(X)
1400     N(X)=I(XI)          : ' CAMBIO DE ESTADO EN VECTOR NUEVO
1410     IF J=2 THEN N(X)=E(1,I) : ' TIEMPO INICIAL
1420   NEXT J
1430 NEXT I
1440 RETURN
1450 REM
1460 REM ----- FUNCIONES LOGICAS -----
1470 REM
1480 FOR I=0 TO ULTELE
1490   ON L(O,I) GOSUB 1530,1620,1710 : ' AND, OR, XOR
1500   GOSUB 1800 : 'FIJADO DE SALIDAS
1510 NEXT I
1520 RETURN
1530 Y=L1      : '*****COMPUERTA AND*****
1540 FOR J=1 TO 5
1550   X=L(J,I)
1560   IF X=0 THEN GOTO 1590
1570   IF A(X)=L0 THEN Y=L0: GOTO 1600
1580   IF A(X)=LX THEN Y=LX
1590 NEXT J
1600 RETURN
1610 REM      *****COMPUERTA OR*****
1620 Y=L0
1630 FOR J=1 TO 5
1640   X=L(J,I)
1650   IF X=0 THEN GOTO 1680
1660   IF A(X)=L1 THEN Y=L1: GOTO 1690
1670   IF A(X)=LX THEN Y=LX

```

```

1680 NEXT J
1690 RETURN
1700 REM          *****COMPUERTA XOR*****
1710 Y=LX
1720 X1=L(1,I)
1730 X2=L(2,I)
1740 IF (A(X1)=LX OR A(X2)=LX) THEN GOTO 1760
1750 IF (A(X1)+A(X2)=1) THEN Y=1 ELSE Y=0
1760 RETURN
1770 REM
1780 REM ----- FIJADO DE SALIDAS -----
1790 REM
1800 X=L(6,I)
1810 N(X)=Y
1820 X=L(7,I)
1830 N(X)=I(Y)
1840 RETURN
1850 REM
1860 REM ----- IMPRESION DE SALIDAS -----
1870 REM
1880 IF P<>0 THEN GOTO 2060 : 'REQUIERE ENCABEZADO?
1890 REM IMPRESION DE ENCABEZADO, Y NUMERO DE LAS SENALES
1900 CLS
1910 PRINT
1920 PRINT "TIEMPO INICIAL DE SIMULACION";T
1930 PRINT "SENALES"
1940 PRINT "      1          10          20          30          40          50          60
70"
1950 FOR I=1 TO 15
1960     FOR JJ=1 TO 15
1970         PRINT "      ";
1980     NEXT JJ
1990 PRINT " "
2000 NEXT I
2010 FOR I=1 TO 10
2020     X=M(I)
2030     IF X<>0 THEN LOCATE 5+I*2,1: PRINT X
2040 NEXT I
2050 REM IMPRESION DE LOS VALORES DE LA SENAL
2060 LOCATE 1,1: PRINT "TIEMPO FINAL DE SIMULACION";T;
2070 FOR I=1 TO 10
2080     X=M(I)
2090     IF X=0 THEN GOTO 2130
2100     Y=A(X)
2110     LOCATE 5+I*2,5+P: PRINT S*(Y);
2120 NEXT I
2130 REM CHEQUEO DE LLENADO DE PANTALLA
2140 FP=0
2150 P=P+1
2160 IF P<71 THEN RETURN
2170 P=0
2180 FP=1
2190 RETURN
2200 REM
2210 REM ----- OBTENCION DEL TIPO DE COMPUERTA -----
2220 REM

```

```

2230 L(O,I)=0
2240 IF L$="A" THEN L(O,I)=1      'COMPUERTA AND
2250 IF L$="O" THEN L(O,I)=2      'COMPUERTA OR
2260 IF L$="X" THEN L(O,I)=3      'COMPUERTA XOR
2270 FOR J=1 TO 7
2280     IF L(J,I)>ULTNOD THEN ULTNOD=L(J,I)
2290 NEXT J
2300 RETURN
2310 REM
2320 REM ----- PROCESADO DE ERROR -----
2330 REM
2340 PRINT "DEMASIADOS ELEMENTOS LOGICOS (;I;)"
2350 STOP
2360 PRINT "NUMERO DE NODO NO VALIDO (;ULTNOD;)"
2370 PRINT "DATO IGNORADO. REPITA LA LINEA COMPLETA ...."
2380 I=I-1
2390 RETURN
2400 PRINT "TIPO DE COMPUERTA NO VALIDO (;L$;)      DATO IGNORADO"
2410 I=I-1
2420 RETURN
2430 PRINT "DEMASIADOS ESTIMULOS EXTERNOS (;I;)"
2440 STOP
2450 PRINT "FIN DEL PROCESO DE SIMULACION"

```

APENDICE B. LISTADO DE SIMCD VERSION 2.0

```

100 REM *****
110 REM *           SIMULADOR DE CIRCUITOS DIGITALES   SIM/CD   VERSION 2.0           *
120 REM *
130 REM * ELABORADO POR:
140 REM *           OSCAR DAVILA MONROY
150 REM *           ROGELIO CISNEROS GUADARRAMA
160 REM *
170 REM *****
180 REM
190 REM *****
200 REM *           UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
210 REM *           FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
220 REM *
230 REM * TESIS: SIMULACION INTERACTIVA DE CIRCUITOS DIGITALES
240 REM *           AUXILIADA POR COMPUTADORA.
250 REM *
260 REM * FECHA: 20/SEP/87
270 REM *****
280 CLS
290 REM
300 REM ----- ESTRUCTURAS DE DATOS -----
310 REM
320 MAXNOD=100           * MAXIMO NUMERO DE NODOS
330 MAXEXT= 30          * MAXIMO NUMERO DE ESTIMULOS EXTERNOS
340 DIM L(7,MAXNOD)    * ARREGLO DE ELEMENTOS LOGICOS
350 DIM E(7,MAXEXT)    * ARREGLO DE ESTIMULO EXTERNO
360 DIM M(10)          * VECTOR DE SENAL A MONITOREAR
370 DIM A(MAXNOD)      * VECTOR DE LA SENAL ANTIGUA
380 DIM N(MAXNOD)      * VECTOR DE LA SENAL NUEVA
390 DIM TS1(MAXNOD)    * VECTOR DE BANDERAS DE CAMBIO LOCAL (TRAZO SEL)
400 DIM TS2(MAXNOD)    * VECTOR DE APUNTADES (TRAZO SELECTIVO)
410 DIM TS3(MAXNOD+50) * VECTOR DE LISTA DE ELEMENTOS (TRAZO SELECTIVO)
420 REM
430 REM ----- VARIABLES USADAS -----
440 REM
450 LO = 0              'REPRESENTACION INTERNA DE CERO LOGICO
460 LI = 1              'REPRESENTACION INTERNA DE UNO LOGICO
470 LX=3                'REPRESENTACION INTERNA DE INDETERMINACION
480 I(L0)=L1           'COMPLEMENTO DE CERO LOGICO
490 I(L1)=L0           'COMPLEMENTO DE UNO LOGICO
500 I(LX)=LX           'COMPLEMENTO DE INDETERMINACION LOGICA
510 S$(L0)="_"         'SIMBOLO IMPRESO PARA CERO LOGICO
520 S$(L1)="X"         'SIMBOLO IMPRESO PARA UNO LOGICO
530 S$(LX)="?"         'SIMBOLO IMPRESO PARA INDETERMINACION LOGICA
540 FP=0                'BANDERA DE FIN DE PAGINA
560 P=0                 'IMPRESION DE UNA LINEA
570                     'ULTELE=ULTIMO ELEMENTO,ULTEXT=ULTIMO ESTIMULO EXTERNO
580                     'ULTNOD=NUMERO MAS ALTO DE NODO

```



```

590          'L$=CODIGO LOGICO
600 REM
610 GOSUB 2950          : ' GENERACION DE GRAFICOS
620 REM
630 REM ----- LECTURA DE DESCRIPCION LOGICA -----
640 REM
650 I=-1
660 GOSUB 5420          : ' DESCRIPCION DE ELEMENTOS LOGICOS
670 ULTELE=I
680 REM
690 REM ----- LECTURA DE ESTIMULO EXTERNO -----
700 REM
710 I=-1
720 CLS
730 PRINT TAB(18);"DESCRIPCION DE ESTIMULOS EXTERNOS"
740 PRINT STRING$(79,"=")
750 PRINT: PRINT "INTRODUZCA O PARA TERMINAR CAPTURA"
760 I=I+1
770 PRINT: PRINT "SENAL ";I+1;".": PRINT: PRINT
780 IF I>MAXEXT THEN GOTO 2620 : ' DEMASIADOS ESTIMULOS EXTERNOS
790 INPUT "NUMERO DE NODO: ";E(0,I)
800 IF E(0,I)=0 THEN GOTO 970 : 'CHEQUEO DE FINALIZACION
810 INPUT "VALOR INICIAL: ";E1
820 INPUT "TIEMPO INICIAL: ";E(2,I)
830 INPUT "TIEMPO 1er CAMBIO: ";E(3,I)
840 INPUT "TIEMPO 2o CAMBIO: ";E(4,I)
850 INPUT "TIEMPO 3er CAMBIO: ";E(5,I)
860 INPUT "TIEMPO 4o CAMBIO: ";E(6,I)
870 INPUT "TIEMPO 5o CAMBIO: ";E(7,I)
880 LOCATE 21,1: PRINT STRING$(79,"_"): PRINT
890 PRINT TAB(10);"DATOS OK? (S)I , (N)O ";
900 INPUT A$
910 IF A$(">S") THEN I=I-1: GOTO 720
920 REM CAMBIO DE VALOR INICIAL A FORMATO INTERNO
930 E(1,I)=LX
940 IF E1=0 THEN E(1,I)=L0
950 IF E1=1 THEN E(1,I)=L1
960 GOTO 720
970 ULTEXT=I-1
980 REM
990 REM ----- LECTURA DE PUNTOS A MONITOREAR -----
1000 REM
1010 CLS
1020 PRINT TAB(18);"DESCRIPCION DE NODOS A MONITOREAR"
1030 PRINT STRING$(79,"=")
1040 PRINT: PRINT "INTRODUZCA O PARA TERMINAR CAPTURA": PRINT
1050 FOR I=1 TO 10
1060     INPUT "NODO NUMERO: ";M(I)
1070     IF M(I)=0 THEN GOTO 1090
1080 NEXT I
1090 LOCATE 21,1: PRINT STRING$(79,"_"): PRINT
1100 PRINT TAB(10);"DATOS OK? (S)I , (N)O ";
1110 INPUT A$
1120 IF A$(">S") THEN GOTO 1010
1140 GOSUB 2640          : ' CONSTRUCCION DE LA TABLA DE TRAZO SELECTIVO

```

```

1150 REM
1160 REM ----- INICIALIZADO (COLOCACION DE NODOS EN LOGICA INDETERM -----
1170 REM
1180 FOR I=0 TO MAXNOD
1190     A(I)=LX           'ARREGLO DEL ANTIGUO VALOR
1200     N(I)=LX           'ARREGLO DEL NUEVO VALOR
1210 NEXT I
1220 T=-1
1230 REM
1240 REM -----
1250 REM           MALLA PRINCIPAL DE SIMULACION
1260 REM -----
1270 REM
1280 T=T+1                :' INCREMENTAR TIEMPO DE SIMULACION
1290 GOSUB 1540           :' ACTUALIZAR ESTIMULO
1300 REM                 TRANSFERIR VECTOR NUEVO EN VECTOR ANTIGUO
1310 GOSUB 2770           :' UTILIZAR TRAZO SELECTIVO EN LA TRANSFERENCIA N-->A
1320 REM
1330 IF BCG=1 THEN GOSUB 1710  :' FUNCIONES LOGICAS SOLO SI HAY CAMBIO GLOBAL
1340 REM
1350 GOSUB 2130           :' IMPRESION DE SALIDAS
1360 IF FP<>1 THEN GOTO 1280
1370 REM FIN DE LA PANTALLA
1380 LOCATE 25,1: PRINT "SELECCIONE (1)CONTINUAR, (2)NODOS, (3)FIN";
1390 INPUT A$
1400 IF A$="1" THEN GOTO 1280
1410 IF A$="2" THEN GOTO 990
1430 IF A$="3" THEN GOTO 6440
1440 GOTO 1380
1450 REM
1460 REM -----
1470 REM           FIN DE MALLA PRINCIPAL DE SIMULACION
1480 REM -----
1490 REM
1500 REM ***** SUBROUTINAS *****
*
1510 REM
1520 REM ----- ACTUALIZAR ESTIMULO -----
1530 REM
1540 FOR I=0 TO ULTEXT
1550     FOR J=2 TO 7
1570         IF T<>E(J,I) THEN GOTO 1650
1580             ' MARCA DE TIEMPO, OBTENGA EL NUMERO DE SENAL
1590             X=E(O,I)
1610             XI=N(X)
1620             N(X)=I(XI)                :' CAMBIO DE ESTADO EN VECTOR NUEVO
1640             IF J=2 THEN N(X)=E(1,I)    :' TIEMPO INICIAL
1650     NEXT J
1660 NEXT I
1670 RETURN
1680 REM
1690 REM ----- FUNCIONES LOGICAS -----
1700 REM
1710 FOR I=0 TO ULTELE
1720     IF TSI(I)=0 THEN GOTO 1760  :' INHIBIR SIMULACION POR TRAZO SELECTIVO
1730     ON L(O,I) GOSUB 1780,1870,1960,1780,1780,1870,1960

```

```

1740     REM          AND, OR, XOR, INV, NAND,NOR, XNOR
1750     GOSUB 2050 : 'FIJADO DE SALIDAS
1760     NEXT I
1770     RETURN
1780     Y=L1       : '****COMPUERTA AND****
1790     FOR J=1 TO 5
1800         X=L(J,I)
1810         IF X=0 THEN GOTO 1840
1820         IF A(X)=L0 THEN Y=L0: GOTO 1850
1830         IF A(X)=LX THEN Y=LX
1840     NEXT J
1850     RETURN
1860     REM          ****COMPUERTA OR****
1870     Y=L0
1880     FOR J=1 TO 5
1890         X=L(J,I)
1900         IF X=0 THEN GOTO 1930
1910         IF A(X)=L1 THEN Y=L1: GOTO 1940
1920         IF A(X)=LX THEN Y=LX
1930     NEXT J
1940     RETURN
1950     REM          ****COMPUERTA XOR****
1960     Y=LX
1970     X1=L(1,I)
1980     X2=L(2,I)
1990     IF (A(X1)=LX OR A(X2)=LX) THEN GOTO 2010
2000     IF (A(X1)+A(X2)=1) THEN Y=1 ELSE Y=0
2010     RETURN
2020     REM
2030     REM ----- FIJADO DE SALIDAS -----
2040     REM
2050     X=L(6,I)
2060     N(X)=Y
2070     X=L(7,I)
2080     N(X)=I(Y)
2090     RETURN
2100     REM
2110     REM ----- IMPRESION DE SALIDAS -----
2120     REM
2130     IF P<>0 THEN GOTO 2310 : 'REQUIERE ENCABEZADO?
2140     REM IMPRESION DE ENCABEZADO, Y NUMERO DE LAS SENALES
2150     CLS
2160     PRINT
2170     PRINT "TIEMPO INICIAL DE SIMULACION";T
2180     PRINT "SENALES"
2190     PRINT "      1      10      20      30      40      50      60
2200     FOR I=1 TO 15
2210         FOR JJ=1 TO 15
2220             PRINT "      I";
2230         NEXT JJ
2240     PRINT " "
2250     NEXT I
2260     FOR I=1 TO 10
2270         X=M(I)
2280     IF X<>0 THEN LOCATE 5+I#2,1: PRINT X

```

```

2290 NEXT I
2300 REM IMPRESION DE LOS VALORES DE LA SENAL
2310 LOCATE 1,1: PRINT "TIEMPO FINAL DE SIMULACION";
2320 FOR I=1 TO 10
2330     X=M(I)
2340     IF X=0 THEN GOTO 2380
2350     Y=A(X)
2360     LOCATE 5+I*2,5+P: PRINT S*(Y);
2370 NEXT I
2380 REM CHEGUEO DE LLENADO DE PANTALLA
2390 FP=0
2400 P=P+1
2410 IF P<71 THEN RETURN
2420 P=0
2430 FP=1
2440 RETURN
2450 REM
2460 REM ----- OBTENCION DEL ULTIMO NODO -----
2470 REM
2480 FOR J=1 TO 7
2490     IF L(J,I)>ULTNOD THEN ULTNOD=L(J,I)
2500 NEXT J
2510 RETURN
2520 REM
2530 REM ----- PROCESADO DE ERROR -----
2540 REM
2550 PRINT "DEMASIADOS ELEMENTOS LOGICOS (;I;)"
2560 STOP
2570 PRINT "NUMERO DE NODO NO VALIDO (;ULTNOD;)"
2580 RETURN
2620 PRINT "DEMASIADOS ESTIMULOS EXTERNOS (;I;)"
2630 STOP
2640 REM ----- CONSTRUCCION DE LA TABLA DE TRAZO SELECTIVO -----
2650 AP=0 : ' APUNTADOR A TS3
2660 FOR I=1 TO ULTNOD : ' PARA TODOS LOS NODOS
2670     TS2(I)=AP
2680     FOR J=0 TO ULTELE : ' BUSCAR EN TODOS LOS ELEMENTOS
2690         FOR K=1 TO 5 : ' PARA CADA TERMINAL DE ENTRADA
2700             IF L(K,J)=I THEN TS3(AP)=J: AP=AP+1
2710         NEXT K
2720     NEXT J
2730     TS3(AP)=-1 : ' MARCA FIN DE BLOQUE
2740     AP=AP+1
2750 NEXT I
2760 RETURN
2770 REM ----- MOVER VECTOR NUEVO EN VECTOR ANTIGUO -----
2780 BCG=0 : ' BANDERA DE CAMBIO GLOBAL APAGADA
2790 FOR II=0 TO ULTELE : ' APAGAR TODAS LAS BANDERAS DE TRAZO SEL.
2800     TS1(II)=0
2810 NEXT II
2820 FOR I=1 TO ULTNOD : ' REVISAR TODOS LOS NODOS
2830     IF N(I) (<> A(I)) THEN GOSUB 2860
2840 NEXT I
2850 RETURN
2860 REM ----- TRAZO SELECTIVO: MARCAS DE CAMBIO DE ESTADO -----

```

```

2870 BCG=1                                :' ENCENDER BANDERA DE CAMBIO GLOBAL
2880 A(I)=N(I)                             :' MOVER VECTOR NUEVO EN ANTIGUO
2890 II=TS2(I)                             :' OBTENER APUNTADOR A TS3
2900 J= TS3(II)                            :' ELEMENTO CONECTADO AL NODO QUE CAMBIO
2910 IF J=-1 THEN RETURN                   :' FIN DEL BLOQUE
2920 TS1(J)=1                               :' ENCENDER BANDERA DE CAMBIO LOCAL
2930 II=II+1
2940 GOTO 2900
2950 REM
2960 REM          DESCRIPCION LOGICA POR MEDIO DE GRAFICOS
2970 REM
2980 DIM PANEL(8,2)
2990 PANEL(1,1)=10: PANEL(1,2)=8
3000 PANEL(2,1)=28: PANEL(2,2)=8
3010 PANEL(3,1)=46: PANEL(3,2)=8
3020 PANEL(4,1)=64: PANEL(4,2)=8
3030 PANEL(5,1)=10: PANEL(5,2)=17
3040 PANEL(6,1)=28: PANEL(6,2)=17
3050 PANEL(7,1)=46: PANEL(7,2)=17
3060 REM ----- GENERACION DE GRAFICOS -----
3070 CLS
3080 LOCATE 15,20,0
3090 REM      ***** GRAFICO INVERSOR *****
3100 DIM INV(150)
3110 LINE (1,8) - (15,8)                   :' LINEA DE ENTRADA
3120 LINE (15,1) - (15,17)                 :' VERTICAL
3130 LINE (15,1) - (58,8)                  :' SUPERIOR
3140 LINE (15,17) - (58,8)                 :' INFERIOR
3150 CIRCLE (62,8),5                        :' CIRCULO NEGADO
3160 GET (1,1) - (68,17),INV
3170 CLS
3180 REM      ***** GRAFICO XOR *****
3190 DIM XORZE(150)
3200 REM LINEAS DE ENTRADA
3210 LINE (1,24) - (18,24)
3220 LINE (1,28) - (18,28)
3230 CIRCLE (1,26),18,.8,.2                 :' VERTICAL 1
3240 CIRCLE (8,26),20,.8,.2                 :' VERTICAL 2
3250 CIRCLE (18,51),68,.75,.87             :' SUPERIOR
3260 CIRCLE (18,1),68,.125,.25            :' INFERIOR
3270 GET (1,16) - (70,36),XORZE
3280 CLS
3290 REM      ***** GRAFICO XNOR *****
3300 DIM XNORZE(170)
3310 REM LINEAS DE ENTRADA
3320 LINE (1,24) - (18,24)
3330 LINE (1,28) - (18,28)
3340 CIRCLE (1,26),18,.8,.2                 :' VERTICAL 1
3350 CIRCLE (8,26),20,.8,.2                 :' VERTICAL 2
3360 CIRCLE (18,51),68,.75,.87             :' SUPERIOR
3370 CIRCLE (18,1),68,.125,.25            :' INFERIOR
3380 CIRCLE (70,26),5                       :' CIRCULO NEGADO
3390 GET (1,16) - (75,36),XNORZE
3400 CLS
3410 REM      ***** GRAFICO NOR DE 5 ENTRADAS *****

```

3420 DIM NORSE(150)  
3430 REM LINEAS DE ENTRADA  
3440 LINE (1,22) - (15,22)  
3450 LINE (1,24) - (18,24)  
3460 LINE (1,26) - (19,26)  
3470 LINE (1,28) - (18,28)  
3480 LINE (1,30) - (15,30)  
3490 CIRCLE (1,26),20,.8,.2 : ' VERTICAL  
3500 CIRCLE (10,51),68,.75,.87 : ' SUPERIOR  
3510 CIRCLE (10,1),68,.125,.25 : ' INFERIOR  
3520 CIRCLE (62,26),5 : ' CIRCULO NEGADO  
3530 GET (1,16) - (70,36),NORSE  
3540 CLS  
3550 REM \*\*\*\*\* GRAFICO NOR DE 4 ENTRADAS \*\*\*\*\*  
3560 DIM NOR4E(150)  
3570 REM LINEAS DE ENTRADA  
3580 LINE (1,22) - (15,22)  
3590 LINE (1,25) - (18,25)  
3600 LINE (1,28) - (18,28)  
3610 LINE (1,31) - (15,31)  
3620 CIRCLE (1,26),20,.8,.2 : ' VERTICAL  
3630 CIRCLE (10,51),68,.75,.87 : ' SUPERIOR  
3640 CIRCLE (10,1),68,.125,.25 : ' INFERIOR  
3650 CIRCLE (62,26),5 : ' CIRCULO NEGADO  
3660 GET (1,16) - (70,36),NOR4E  
3670 CLS  
3680 REM \*\*\*\*\* GRAFICO NOR DE 3 ENTRADAS \*\*\*\*\*  
3690 DIM NOR3E(150)  
3700 REM LINEAS DE ENTRADA  
3710 LINE (1,22) - (15,22)  
3720 LINE (1,26) - (19,26)  
3730 LINE (1,30) - (15,30)  
3740 CIRCLE (1,26),20,.8,.2 : ' VERTICAL  
3750 CIRCLE (10,51),68,.75,.87 : ' SUPERIOR  
3760 CIRCLE (10,1),68,.125,.25 : ' INFERIOR  
3770 CIRCLE (62,26),5 : ' CIRCULO NEGADO  
3780 GET (1,16) - (70,36),NOR3E  
3790 CLS  
3800 REM \*\*\*\*\* GRAFICO NOR DE 2 ENTRADAS \*\*\*\*\*  
3810 DIM NOR2E(150)  
3820 REM LINEAS DE ENTRADA  
3830 LINE (1,22) - (15,22)  
3840 LINE (1,30) - (15,30)  
3850 CIRCLE (1,26),20,.8,.2 : ' VERTICAL  
3860 CIRCLE (10,51),68,.75,.87 : ' SUPERIOR  
3870 CIRCLE (10,1),68,.125,.25 : ' INFERIOR  
3880 CIRCLE (62,26),5 : ' CIRCULO NEGADO  
3890 GET (1,16) - (70,36),NOR2E  
3900 CLS  
3910 REM \*\*\*\*\* GRAFICO OR DE 5 ENTRADAS \*\*\*\*\*  
3920 DIM OR5E(150)  
3930 REM LINEAS DE ENTRADA  
3940 LINE (1,22) - (15,22)  
3950 LINE (1,24) - (18,24)  
3960 LINE (1,26) - (19,26)

```

3970 LINE (1,28) - (18,28)
3980 LINE (1,30) - (15,30)
3990 CIRCLE (1,26),20,.8,.2      :' VERTICAL
4000 CIRCLE (10,51),68,.75,.87  :' SUPERIOR
4010 CIRCLE (10,1),68,.125,.25  :' INFERIOR
4020 GET (1,16) - (60,36),OR5E
4030 CLS
4040 REM      ***** GRAFICO OR DE 4 ENTRADAS *****
4050 DIM OR4E(150)
4060 REM LINEAS DE ENTRADA
4070 LINE (1,22) - (15,22)
4080 LINE (1,25) - (18,25)
4090 LINE (1,28) - (18,28)
4100 LINE (1,31) - (15,31)
4110 CIRCLE (1,26),20,.8,.2      :' VERTICAL
4120 CIRCLE (10,51),68,.75,.87  :' SUPERIOR
4130 CIRCLE (10,1),68,.125,.25  :' INFERIOR
4140 GET (1,16) - (60,36),OR4E
4150 CLS
4160 REM      ***** GRAFICO OR DE 3 ENTRADAS *****
4170 DIM OR3E(150)
4180 REM LINEAS DE ENTRADA
4190 LINE (1,22) - (15,22)
4200 LINE (1,26) - (19,26)
4210 LINE (1,30) - (15,30)
4220 CIRCLE (1,26),20,.8,.2      :' VERTICAL
4230 CIRCLE (10,51),68,.75,.87  :' SUPERIOR
4240 CIRCLE (10,1),68,.125,.25  :' INFERIOR
4250 GET (1,16) - (60,36),OR3E
4260 CLS
4270 REM      ***** GRAFICO OR DE 2 ENTRADAS *****
4280 DIM OR2E(150)
4290 REM LINEAS DE ENTRADA
4300 LINE (1,22) - (15,22)
4310 LINE (1,30) - (15,30)
4320 CIRCLE (1,26),20,.8,.2      :' VERTICAL
4330 CIRCLE (10,51),68,.75,.87  :' SUPERIOR
4340 CIRCLE (10,1),68,.125,.25  :' INFERIOR
4350 GET (1,16) - (60,36),OR2E
4360 CLS
4370 REM      ***** GRAFICO NAND DE 5 ENTRADAS *****
4380 DIM NAND5E(150)
4390 REM LINEAS DE ENTRADA
4400 LINE (1,5) - (15,5)
4410 LINE (1,7) - (15,7)
4420 LINE (1,9) - (15,9)
4430 LINE (1,11) - (15,11)
4440 LINE (1,13) - (15,13)
4450 LINE (15,1) - (15,17)      :' LINEA VERTICAL
4460 LINE (15,1) - (40,1)      :' LINEA SUPERIOR
4470 LINE (15,17) - (40,17)    :' LINEA INFERIOR
4480 CIRCLE (40,9),16,.75,.25  :' SEMICIRCULO
4490 CIRCLE (62,9),5          :' CIRCULO NEGADO
4500 GET (1,1) - (70,17),NAND5E
4510 CLS

```

```

4520 REM      ***** GRAFICO NAND DE 4 ENTRADAS *****
4530 DIM NAND4E(150)
4540 REM LINEAS DE ENTRADA
4550 LINE (1,7) - (15,7)
4560 LINE (1,9) - (15,9)
4570 LINE (1,11) - (15,11)
4580 LINE (1,13) - (15,13)
4590 LINE (15,1) - (15,17)      : ' LINEA VERTICAL
4600 LINE (15,1) - (40,1)      : ' LINEA SUPERIOR
4610 LINE (15,17) - (40,17)    : ' LINEA INFERIOR
4620 CIRCLE (40,9),16,.75,.25  : ' SEMICIRCULO
4630 CIRCLE (62,9),5           : ' CIRCULO NEGADO
4640 GET (1,1) - (70,17),NAND4E
4650 CLS
4660 REM      ***** GRAFICO NAND DE 3 ENTRADAS *****
4670 DIM NAND3E(150)
4680 REM LINEAS DE ENTRADA
4690 LINE (1,5) - (15,5)
4700 LINE (1,9) - (15,9)
4710 LINE (1,13) - (15,13)
4720 LINE (15,1) - (15,17)      : ' LINEA VERTICAL
4730 LINE (15,1) - (40,1)      : ' LINEA SUPERIOR
4740 LINE (15,17) - (40,17)    : ' LINEA INFERIOR
4750 CIRCLE (40,9),16,.75,.25  : ' SEMICIRCULO
4760 CIRCLE (62,9),5           : ' CIRCULO NEGADO
4770 GET (1,1) - (70,17),NAND3E
4780 CLS
4790 REM      ***** GRAFICO NAND DE 2 ENTRADAS *****
4800 DIM NAND2E(150)
4810 REM LINEAS DE ENTRADA
4820 LINE (1,5) - (15,5)
4830 LINE (1,13) - (15,13)
4840 LINE (15,1) - (15,17)      : ' LINEA VERTICAL
4850 LINE (15,1) - (40,1)      : ' LINEA SUPERIOR
4860 LINE (15,17) - (40,17)    : ' LINEA INFERIOR
4870 CIRCLE (40,9),16,.75,.25  : ' SEMICIRCULO
4880 CIRCLE (62,9),5           : ' CIRCULO NEGADO
4890 GET (1,1) - (70,17),NAND2E
4900 CLS
4910 REM      ***** GRAFICO AND DE 5 ENTRADAS *****
4920 DIM AND5E(150)
4930 REM LINEAS DE ENTRADA
4940 LINE (1,5) - (15,5)
4950 LINE (1,7) - (15,7)
4960 LINE (1,9) - (15,9)
4970 LINE (1,11) - (15,11)
4980 LINE (1,13) - (15,13)
4990 LINE (15,1) - (15,17)      : ' LINEA VERTICAL
5000 LINE (15,1) - (40,1)      : ' LINEA SUPERIOR
5010 LINE (15,17) - (40,17)    : ' LINEA INFERIOR
5020 CIRCLE (40,9),16,.75,.25  : ' SEMICIRCULO
5030 GET (1,1) - (70,17),AND5E
5040 CLS
5050 REM      ***** GRAFICO AND DE 4 ENTRADAS *****
5060 DIM AND4E(1000)

```



```

5070 REM LINEAS DE ENTRADA
5080 LINE (1,7) - (15,7)
5090 LINE (1,9) - (15,9)
5100 LINE (1,11) - (15,11)
5110 LINE (1,13) - (15,13)
5120 LINE (15,1) - (15,17) : LINEA VERTICAL
5130 LINE (15,1) - (40,1) : LINEA SUPERIOR
5140 LINE (15,17) - (40,17) : LINEA INFERIOR
5150 CIRCLE (40,9),16,.75,.25 : SEMICIRCULO
5160 GET (1,1) - (70,17),AND4E
5170 CLS
5180 REM ***** GRAFICO AND DE 3 ENTRADAS *****
5190 DIM AND3E(150)
5200 REM LINEAS DE ENTRADA
5210 LINE (1,5) - (15,5)
5220 LINE (1,9) - (15,9)
5230 LINE (1,13) - (15,13)
5240 LINE (15,1) - (15,17) : LINEA VERTICAL
5250 LINE (15,1) - (40,1) : LINEA SUPERIOR
5260 LINE (15,17) - (40,17) : LINEA INFERIOR
5270 CIRCLE (40,9),16,.75,.25 : SEMICIRCULO
5280 GET (1,1) - (70,17),AND3E
5290 CLS
5300 REM ***** GRAFICO AND DE 2 ENTRADAS *****
5310 DIM AND2E(150)
5320 REM LINEAS DE ENTRADA
5330 LINE (1,7) - (15,7)
5340 LINE (1,11) - (15,11)
5350 LINE (15,1) - (15,17) : LINEA VERTICAL
5360 LINE (15,1) - (40,1) : LINEA SUPERIOR
5370 LINE (15,17) - (40,17) : LINEA INFERIOR
5380 CIRCLE (40,9),16,.75,.25 : SEMICIRCULO
5390 GET (1,1) - (70,17),AND2E
5400 CLS
5410 RETURN
5420 REM ----- PANEL DE COMPUERTAS -----
5430 CLS
5440 PRINT " CAPTURE DE DESCRIPCION LOGICA"
5450 PRINT STRING$(79,"=");
5460 PUT (50,25),AND5E
5470 PUT (200,25),OR5E
5480 PUT (350,25),XOR2E
5490 PUT (500,25),INV
5500 PUT (50,100),NAND3E
5510 PUT (200,100),NOR3E
5520 PUT (350,100),XNOR2E
5530 LINE (1,80) - (639,80)
5540 LINE (1,160) - (639,160)
5550 LOCATE 22,1
5560 PRINT "1. SELECCIONE UN ELEMENTO UTILIZANDO LAS FLECHAS DEL TECLADO"
5570 PRINT "2. INTRODUCZA EL NUMERO DE ENTRADAS DESEADAS EN EL CORCHETE"
5580 PRINT "3. PRESIONE LA TECLA ESC PARA TERMINAR LA CAPTURE"
5590 REM *****
5600 FOR TIPO=1 TO 7
5610 X=PANEL(TIPO,1)

```

```

5620     Y=PANEL(TIPO,2)
5630     LOCATE Y,X-1 : PRINT "[*";
5640     LOCATE Y,X+1 : PRINT "]*";
5650 NEXT TIPO
5660 REM     ***** SELECCION DEL NUMERO DE ENTRADAS *****
5670 TIPO=1
5680 LOCATE PANEL(TIPO,2),PANEL(TIPO,1),1
5690 A$=INPUT$(1)
5700 IF (VAL(A$)>0 AND VAL(A$)<=5 AND (TIPO=1 OR TIPO=2 OR TIPO=5 OR TIPO=6))
THEN GOTO 5790
5710 IF (VAL(A$)=2 AND (TIPO=3 OR TIPO=7)) THEN GOTO 5790
5720 IF (VAL(A$)=1 AND (TIPO=4)) THEN GOTO 5790
5730 IF (ASC(A$)=28 AND TIPO>1) THEN TIPO=TIPO-1
5740 IF (ASC(A$)=29 AND TIPO<7) THEN TIPO=TIPO+1
5750 IF (ASC(A$)=30 AND TIPO>4) THEN TIPO=TIPO-4
5760 IF (ASC(A$)=31 AND TIPO<4) THEN TIPO=TIPO+4
5770 IF (ASC(A$)=27) THEN GOTO 6430
5780 GOTO 5680
5790 NUMENT=VAL(A$)
5800 I=I+1
5810 IF I>MAXNOD THEN GOTO 2550           : ' DEMASIADOS ELEMENTOS LOGICOS
5820 L(O,I)=TIPO
5830 PRINT A$
5840 FOR JJ=1 TO 500: NEXT JJ
5850 CLS
5860 REM     ***** CAPTURA DE LAS ENTRADAS Y SALIDAS *****
5870 ON TIPO GOTO 5890,5940,5990,6000,6010,6060,6110,6120
5880 REM     AND, OR, XOR, INV, NAND,NOR,XNOR
5890 ON NUMENT GOTO 5900,5900,5910,5920,5930
5900 PUT (250,50),AND2E: GOTO 6130
5910 PUT (250,50),AND3E: GOTO 6130
5920 PUT (250,50),AND4E: GOTO 6130
5930 PUT (250,50),AND5E: GOTO 6130
5940 ON NUMENT GOTO 5950,5950,5960,5970,5980
5950 PUT (250,50),OR2E: GOTO 6130
5960 PUT (250,50),OR3E: GOTO 6130
5970 PUT (250,50),OR4E: GOTO 6130
5980 PUT (250,50),OR5E: GOTO 6130
5990 PUT (250,50),XOR2E:GOTO 6130
6000 PUT (250,50),INV: GOTO 6130
6010 ON NUMENT GOTO 6020,6020,6030,6040,6050
6020 PUT (250,50),NAND2E: GOTO 6130
6030 PUT (250,50),NAND3E: GOTO 6130
6040 PUT (250,50),NAND4E: GOTO 6130
6050 PUT (250,50),NAND5E: GOTO 6130
6060 ON NUMENT GOTO 6070,6070,6080,6090,6100
6070 PUT (250,50),NOR2E: GOTO 6130
6080 PUT (250,50),NOR3E: GOTO 6130
6090 PUT (250,50),NOR4E: GOTO 6130
6100 PUT (250,50),NOR5E: GOTO 6130
6110 PUT (250,50),XNOR2E:GOTO 6130
6120 PRINT "ELEMENTO INVALIDO": STOP
6130 FOR JJ=1 TO NUMENT
6140     LOCATE 15+JJ,10
6150     PRINT "E";JJ;" ";
6160 NEXT JJ

```

```

6170 LOCATE 15,60
6180 PRINT "SAL:";
6190 FOR JJ=1 TO NUMENT
6200     LOCATE 15+JJ,15
6210     INPUT L(JJ,I)
6220 NEXT JJ
6230 LOCATE 15,65
6240 IF (TIPO=1 OR TIPO=2 OR TIPO=3) THEN INPUT L(6,I)
6250 IF (TIPO=4 OR TIPO=5 OR TIPO=6 OR TIPO=7) THEN INPUT L(7,I)
6260 REM --- PREGUNTAR SI DATOS OK ----
6270 LOCATE 21,1
6280 PRINT STRING$(79,"_"):PRINT
6290 PRINT TAB(10);"DATOS OK? (S)I, (N)O ----> ";
6300 A$=INPUT$(1)
6310 IF A$="S" THEN GOSUB 2460: GOTO 5430      : ' OBTENCION DEL ULTNOD: CAPTURADA
6320 IF A$(">N" THEN 6270
6330 PRINT "NO":PRINT
6340 PRINT TAB(10);"DATO IGNORADO"
6350 FOR II=1 TO 300:NEXT II
6360 I=I-1
6370 GOTO 5430
6380 REM DATOS OK
6390 GOSUB 2460                               : ' OBTENCION DEL ULTIMO NODO
6400 IF ULTNOD<=MAXNOD THEN GOTO 5430        : ' REGRESA A CAPTURADA SIG. ELEMENTO
6410 GOSUB 2570                               : ' NUMERO DE NODO NO VALIDO
6420 GOTO 6340                               : ' DATO IGNORADO
6430 RETURN
6440 PRINT "FIN DEL PROCESO DE SIMULACION"

```

APENDICE C. EJEMPLOS.

EJEMPLO 1.

Diagrama del Circuito:



Tabla de Verdad:

1	2	3
0	0	1
0	1	1
1	0	1
1	1	0

Descripción Lógica:

A, 1, 2, , , , 3  
FIN, , , , , ,

Descripción de Estímulos Externos:

1, 0, 0, 10, 20, 30, 40, 50  
1, 0, 60, 70, 80, 90, 100, 110  
2, 0, 0, 5, 10, 15, 20, 25  
2, 0, 30, 35, 40, 45, 50, 55  
2, 0, 60, 65, 70, 75, 80, 85  
0, , , , , ,

Descripción de las Señales a Monitorear:

1  
2  
3  
0

Resultado de la Simulación:

TIEMPO FINAL DE SIMULACION: 70  
TIEMPO INICIAL DE SIMULACION: 0  
SENALES

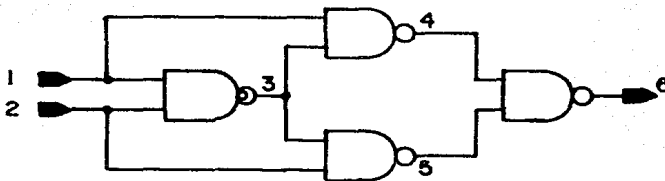
	1	10	20	30	40	50	60	70
1	_____	XXXXXXXXXX	_____	XXXXXXXXXX	_____	XXXXXXXXXX	_____	X
2	_____	XXXXXX	_____	XXXXXX	_____	XXXXXX	_____	XXXXXX
3	XXXXXXXXXXXXXXXXXX	_____	XXXXXXXXXXXXXXXXXX	_____	XXXXXXXXXXXXXXXXXX	_____	XXXXXXXXXX	_____

QUIERE CONTINUAR? (S)I, (N)O, (R)ECOMENZAR ? N

FIN DEL PROCESO DE SIMULACION

**EJEMPLO 2.**

**Diagrama del Circuito:**



**Tabla de Verdad:**

1	2	6
0	0	0
0	1	1
1	0	1
1	1	0

**Descripción Lógica:**

A, 1, 2, ..., 3  
 A, 1, 3, ..., 4  
 A, 2, 3, ..., 5  
 A, 4, 5, ..., 6  
 FIN, ..., ,

**Descripción de Estímulos Externos:**

1, 0, 0, 10, 20, 30, 40, 50  
 1, 0, 60, 70, 80, 90, 100, 110  
 2, 0, 0, 5, 10, 15, 20, 25  
 2, 0, 30, 35, 40, 45, 50, 55  
 2, 0, 60, 65, 70, 75, 80, 85  
 0, ..., ,

Descripción de las Señales a Monitorear:

1  
2  
6  
0

Resultado de la Simulación:

TIEMPO FINAL DE SIMULACION: 70  
TIEMPO INICIAL DE SIMULACION: 0  
SENALES

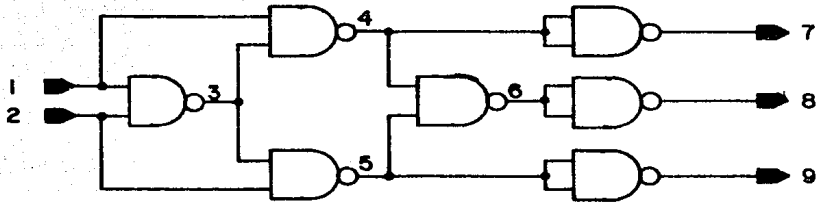
	1	10	20	30	40	50	60	70
1	_____XXXXXXXXXX_____	XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXXXX	X
2	_____XXXXX_____	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX
6	??	XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXXXX	XXXX

QUIERE CONTINUAR? (S)I, (N)O, (R)ECOMENZAR ? N

FIN DEL PROCESO DE SIMULACION

**EJEMPLO 3.**

-----  
**Diagrama del Circuito:**



**Tabla de Verdad:**

1	2	8	7	9
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

**Descripción Lógica:**

A,1,2,,,,,3  
 A,1,3,,,,,4  
 A,2,3,,,,,5  
 A,4,5,,,,,6  
 A,4,4,,,,,7  
 A,6,6,,,,,8  
 A,5,5,,,,,9  
 FIN,,,,,.

**Descripción de Estímulos Externos:**

1, 0, 0, 10, 20, 30, 40, 50  
 1, 0, 60, 70, 80, 90, 100, 110  
 2, 0, 0, 5, 10, 15, 20, 25  
 2, 0, 30, 35, 40, 45, 50, 55  
 2, 0, 60, 65, 70, 75, 80, 85  
 0,,,,,.



Descripción de Señales a Monitorear:

- 1
- 2
- 7
- 8
- 9
- 0

Resultado de la Simulación:

TIEMPO FINAL DE SIMULACION: 70

TIEMPO INICIAL DE SIMULACION: 0

SENALES

	1	10	20	30	40	50	60	70			
1	_____	XXXXXXXXXX	_____	XXXXXXXXXX	_____	XXXXXXXXXX	_____	X			
2	_____	XXXXXX	_____	XXXXXX	_____	XXXXXX	_____	XXXXXX			
7	??	_____	XXXXXX	_____	XXXXXX	_____	XXXXXX	_____			
8	???	XXXXXX	_____	XXXXXXXXXX	_____	XXXXXXXXXX	_____	XXXXXXXXXX			
9	??	_____	XXXXXX	_____	X	_____	XXXXXX	_____	X	_____	XXXX

QUIERE CONTINUAR? (S)I, (N)O, (R)ECOMENZAR ? N

FIN DEL PROCESO DE SIMULACION

**EJEMPLO 4.**

Diagrama del Circuito:

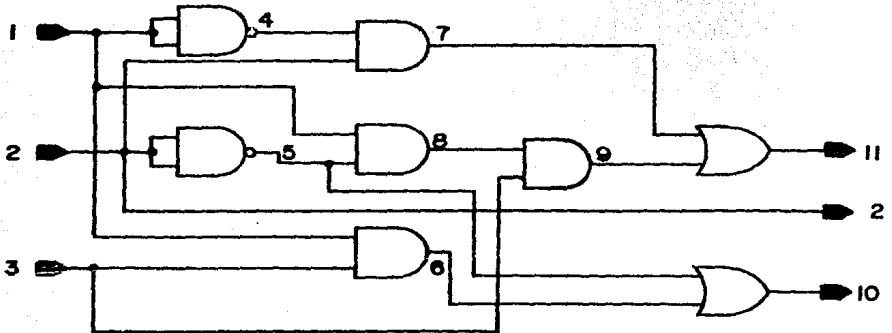


Tabla de Verdad:

3	2	1	10	2	11
0	0	0	1	0	0
0	0	1	1	0	0
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	0	1	1
1	1	1	1	1	0

Descripción Lógica:

A, 1, 2, ..., 4  
 A, 2, 2, ..., 5  
 A, 1, 3, ..., 6,  
 A, 4, 2, ..., 7,  
 A, 1, 5, ..., 8,  
 A, 8, 3, ..., 9,  
 O, 5, 6, ..., 10,  
 O, 7, 9, ..., 11,  
 FIN, ..., ,

**Descripción de Estímulos Externos:**

1, 0, 0, 5, 10, 15, 20, 25  
 1, 0, 30, 35, 40, 45, 50, 55  
 1, 0, 60, 65, 70, 75, 80, 85  
 2, 0, 0, 10, 20, 30, 40, 50  
 2, 0, 60, 70, 80, 90, 100, 110  
 3, 0, 0, 20, 40, 60, 80, 100  
 0, , , , , , ,

**Descripción de las Señales a Monitorear:**

1  
 2  
 3  
 11  
 2  
 10  
 0

**Resultado de la Simulación:**

TIEMPO FINAL DE SIMULACION: 70  
 TIEMPO INICIAL DE SIMULACION: 0  
 SENALES

	1	10	20	30	40	50	60	70
1	_____	_____	_____	_____	_____	_____	_____	_____
2	_____	_____	_____	_____	_____	_____	_____	_____
3	_____	_____	_____	_____	_____	_____	_____	_____
11	??	_____	_____	_____	_____	_____	_____	_____
2	_____	_____	_____	_____	_____	_____	_____	_____
10	??	_____	_____	_____	_____	_____	_____	_____

QUIERE CONTINUAR? (S)I, (N)O, (R)ECOMENZAR ? N

FIN DEL PROCESO DE SIMULACION

EJEMPLO 5.

Diagrama del Circuito:

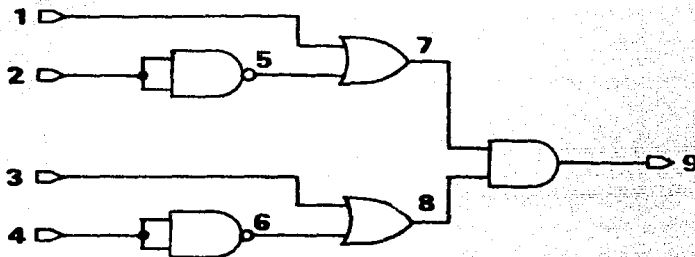


Tabla de Verdad:

4	2	3	1	9
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

**Descripción Lógica:**

A,2,2,,,,5  
0,1,5,,,,7,  
A,4,4,,,,6  
0,3,6,,,,8,  
A,7,8,,,,9,  
FIN,,,,,

**Descripción de Estímulos Externos:**

1, 0, 0, 5, 10, 15, 20, 25  
1, 0, 30, 35, 40, 45, 50, 55  
1, 0, 60, 65, 70, 75, 80, 85  
2, 0, 0, 20, 40, 60, 80, 100  
3, 0, 0, 10, 20, 30, 40, 50  
3, 0, 60, 70, 80, 90, 100, 110  
4, 0, 0, 40, 80, 120, 160, 200  
0,,,,,

**Descripción de las Señales a Monitorear:**

1  
2  
3  
4  
9  
0

Resultado de la Simulación:

TIEMPO FINAL DE SIMULACION: 70  
TIEMPO INICIAL DE SIMULACION: 0  
SENALES

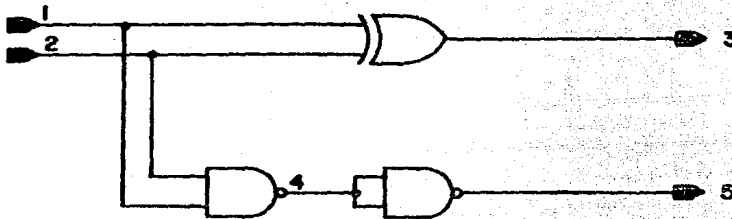
	1	10	20	30	40	50	60	70
1	_____	XXXXX	_____	XXXXX	_____	XXXXX	_____	XXXXX
2	_____	_____	XXXXXXXXXXXXXXXXXXXX	_____	_____	_____	XXXXXXXXXXXX	_____
3	_____	XXXXXXXXXX	_____	XXXXXXXXXX	_____	XXXXXXXXXX	_____	X
4	_____	_____	_____	_____	XXXXXXXXXXXXXXXXXXXX	_____	XXXXXXXXXXXX	_____
9	???	XXXXXXXXXXXXXXXXXXXX	_____	XXXXX	_____	XXXXX	_____	XXXXXXXXXX

QUIERE CONTINUAR? (S)I, (N)O, (R)ECOMENZAR ? N

FIN DEL PROCESO DE SIMULACION

**EJEMPLO 6.**

**Diagrama del Circuito:**



**Tabla de Verdad:**

1	2	3	5
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

**Descripción Lógica:**

X, 1, 2, ..., 3,  
 A, 1, 2, ..., 4  
 A, 4, 4, ..., 5  
 FIN, ..., ...

**Descripción de Estímulos Externos:**

1, 0, 0, 10, 20, 30, 40, 50  
 1, 0, 60, 70, 80, 90, 100, 110  
 2, 0, 0, 5, 10, 15, 20, 25  
 2, 0, 30, 35, 40, 45, 50, 55  
 2, 0, 60, 65, 70, 75, 80, 85  
 0, ..., ...

Descripción de las Señales a Monitorear:

1  
2  
3  
5  
0

Resultado de la Simulación:

TIEMPO FINAL DE SIMULACION: 70  
TIEMPO INICIAL DE SIMULACION: 0  
SENALES

	1	10	20	30	40	50	60	70
1		XXXXXXXXXX		XXXXXXXXXX		XXXXXXXXXX		X
2	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX
3	? XXXXXXXXXXXX		XXXXXXXXXXXX		XXXXXXXXXXXX		XXXXXX	XXXXXX
5	??	XXXXX		XXXXX		XXXXX		XXXXX

QUIERE CONTINUAR? (S)I, (N)O, (R)ECOMENZAR ? N

FIN DEL PROCESO DE SIMULACION



**EJEMPLO 7.**

Diagrama del Circuito:

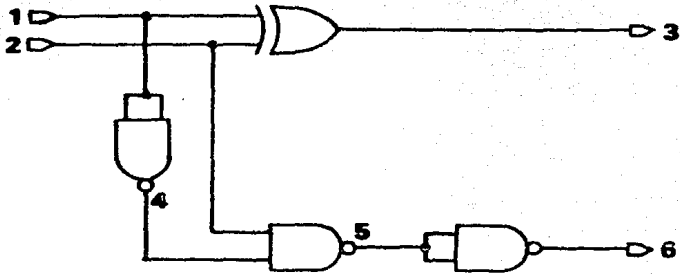


Tabla de Verdad:

1	2	3	6
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Descripción Lógica:

X, 1, 2, ..., 3,  
A, 1, 1, ..., 4  
A, 2, 4, ..., 5  
A, 3, 5, ..., 6  
FIN, ..., ..

**Descripción de Estimulos Externos:**

1, 0, 0, 10, 20, 30, 40, 50  
1, 0, 60, 70, 80, 90, 100, 110  
2, 0, 0, 5, 10, 15, 20, 25  
2, 0, 30, 35, 40, 45, 50, 55  
2, 0, 60, 65, 70, 75, 80, 85  
0, , , , , , , ,

**Descripción de las Señales a Monitorear:**

1  
2  
3  
6  
0

**Resultado de la Simulación:**

TIEMPO FINAL DE SIMULACION: 70  
TIEMPO INICIAL DE SIMULACION: 0  
SENALES

	1	10	20	30	40	50	60	70
1	_____	XXXXXXXXXX	_____	XXXXXXXXXX	_____	XXXXXXXXXX	_____	X
2	_____	XXXXX	_____	XXXXX	_____	XXXXX	_____	XXXXX
3	?	XXXXXXXXXX	_____	XXXXXXXXXX	_____	XXXXXXXXXX	_____	XXXXX
4	??	XXXXX	_____	XXXXX	_____	XXXXX	_____	XXXX

QUIERE CONTINUAR? (S)I, (N)O, (R)ECOMENZAR ? N

FIN DEL PROCESO DE SIMULACION

EJEMPLO 8.

Diagrama del Circuito:

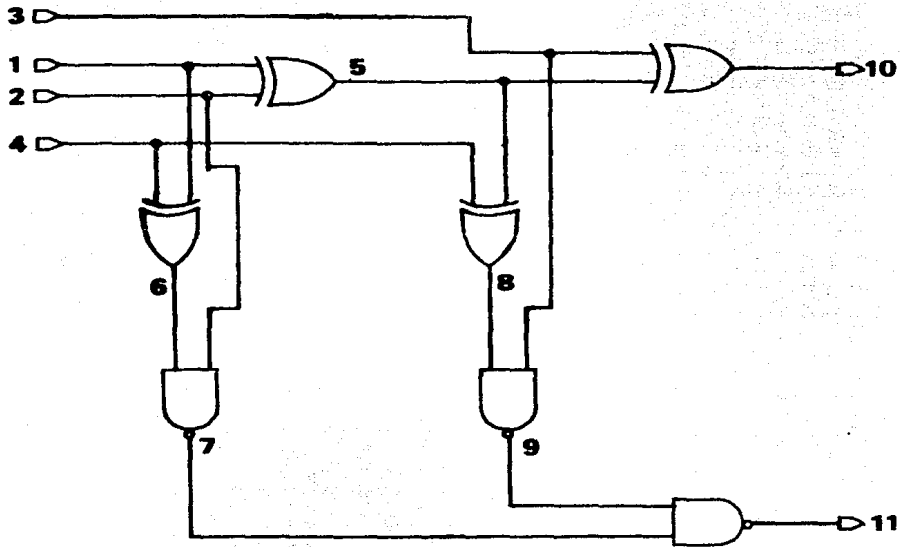


Tabla de Verdad:

1	2	3	4	10	11
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	0	1	0
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	0	1	1

**Descripción Lógica:**

X,1,2,,,,5,  
X,1,4,,,,6,  
A,6,2,,,,7  
X,5,4,,,,8,  
A,3,8,,,,9  
X,3,5,,,,10,  
A,9,7,,,,11  
FIN,,,,,

**Descripción de Estímulos Externos:**

1, 0, 0, 20, 40, 60, 80, 100  
2, 0, 0, 10, 20, 30, 40, 50  
2, 0, 60, 70, 80, 90, 100, 110  
3, 0, 0, 5, 10, 15, 20, 25  
3, 0, 30, 35, 40, 45, 50, 55  
3, 0, 60, 65, 70, 75, 80, 85  
4, 0, 0, 0, 0, 0, 0, 0  
0,,,,,

**Descripción de las Señales a Monitorear:**

1  
2  
3  
4  
10  
11  
0

Resultado de la Simulacion:

TIEMPO FINAL DE SIMULACION: 70  
TIEMPO INICIAL DE SIMULACION: 0  
SENALES

	1	10	20	30	40	50	60	70	
1	XXXXXXXXXXXXXXXXXXXXXXXX							XXXXXXXXXXXX	
2	XXXXXXXXXXXX			XXXXXXXXXXXX		XXXXXXXXXXXX			X
3	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	
4	_____								
10	??	XXXXXX	XXXXXX	X	XXXXXX	XXXXXX	XXXXXX	XXXXXX	
11	??	XXXXXXXXXXXXXXXXXXXX				XXXXXX	XXXXXX	XXXXXX	

QUIERE CONTINUAR? (S)I, (N)O, (R)ECOMENZAR ? N

FIN DEL PROCESO DE SIMULACION

EJEMPLO 9.

Diagrama del Circuito:

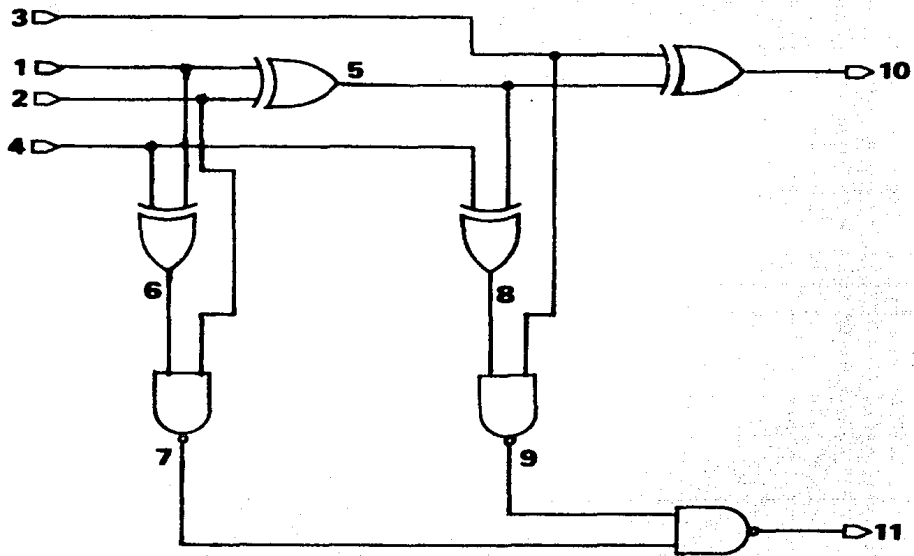


Tabla de Verdad:

1	2	3	4	10	11
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	1	0	1
1	0	0	1	1	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	1	1	1

**Descripción Lógica:**

X,1,2,,,,5,  
X,1,4,,,,6,  
A,6,2,,,,7,  
X,5,4,,,,8,  
A,3,8,,,,9,  
X,3,5,,,,10,  
A,9,7,,,,11  
FIN,,,,,

**Descripción de Estímulos Externos:**

1, 0, 0, 20, 40, 60, 80, 100  
2, 0, 0, 10, 20, 30, 40, 50  
2, 0, 60, 70, 80, 90, 100, 110  
3, 0, 0, 5, 10, 15, 20, 25  
3, 0, 30, 35, 40, 45, 50, 55  
3, 0, 60, 65, 70, 75, 80, 85  
4, 1, 0, 0, 0, 0, 0, 0  
0,,,,,

**Descripción de las Señales a Monitorear:**

1  
2  
3  
4  
10  
11  
0

Resultado de la Simulación:

TIEMPO FINAL DE SIMULACION: 70  
 TIEMPO INICIAL DE SIMULACION: 0  
 SENALES

	1	10	20	30	40	50	60	70
1	XX							
2	XXXXXXXXXX		XXXXXXXXXX		XXXXXXXXXX		XXXXXXXXXX	X
3	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX
4	XX							
10	??	XXXXXX	XXXXXX	X	XXXXXX	XXXXXX	XXXXXX	XXXXXX
11	??	XXXXXXXXXXXXXXXXXXXX			XXXXXX	XXXXXXXXXXXXXXXXXXXX		

QUIERE CONTINUAR? (S)I, (N)O, (R)ECOMENZAR ? N

FIN DEL PROCESO DE SIMULACION



EJEMPLO 10.

Diagrama del Circuito:

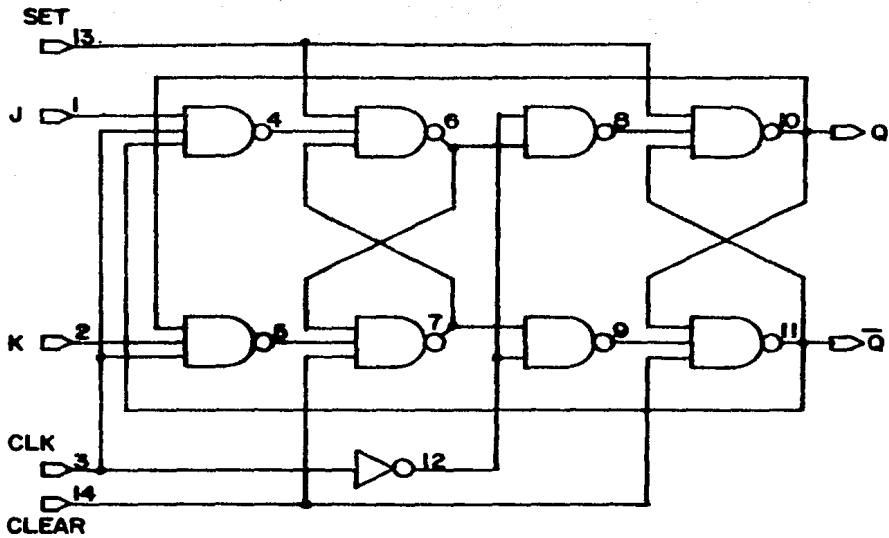


Tabla de Verdad:

J	K	$Q_n$	$Q_{n+1}$
0	X	0	0
1	X	0	1
X	1	1	0
X	0	1	1

**Descripción Lógica:**

A,11,1,3,,,,4  
A,3,2,10,,,,5  
A,4,7,13,,,,6  
A,6,5,14,,,,7  
A,6,12,,,,,8  
A,12,7,,,,,9  
A,8,11,13,,,,10  
A,9,10,14,,,,11  
A,3,,,,,12  
FIN,,,,,:

**Descripción de Estímulos Externos:**

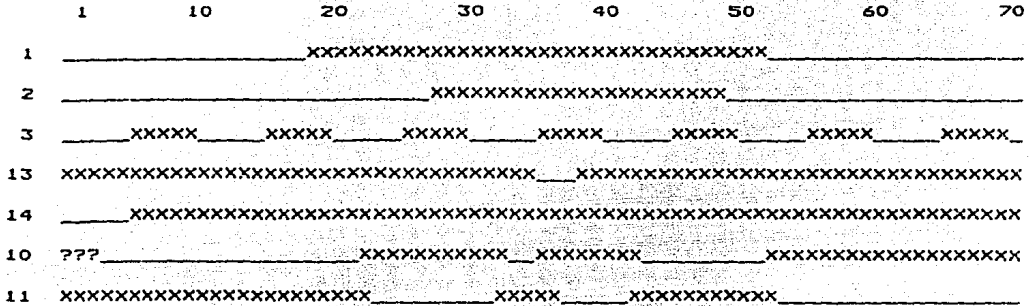
1, 0, 0, 18, 52, 999, 999, 999  
2, 0, 0, 27, 48, 999, 999, 999  
3, 0, 0, 5, 10, 15, 20, 25  
3, 0, 30, 35, 40, 45, 50, 55  
3, 0, 60, 65, 70, 75, 80, 85  
13, 1, 0, 35, 38, 999, 999, 999  
14, 0, 0, 5, 999, 999, 999, 999  
0,,,,,:

**Descripción de las Señales a Monitorear:**

1  
2  
3  
13  
14  
10  
11  
0

**Resultado de la Simulacio'n:**

TIEMPO FINAL DE SIMULACION: 70  
TIEMPO INICIAL DE SIMULACION: 0  
SENALES



QUIERE CONTINUAR? (S)I, (N)O, (R)ECOMENZAR ? N

FIN DEL PROCESO DE SIMULACION

REFERENCIAS:

- 
1. Kiviat y Pritsker  
Simulation with GASP II  
Prentice Hall, 1967;  
Pritsker  
The GASP IV Simulation Language  
Wiley, 1974
  2. IBM Corporation  
General Purpose Simulation System/360 CS Version 2  
Manual de Usuario SH20-0694-0
  3. Markowitz, Hausner y Karr  
Simsript, a Simulation Programming Language  
Prentice Hall, 1963
  4. Dahl y Nigaard  
Simula- A Language for Programming and Description of Discrete  
Event Systems; Manual de Usuario  
Centro de Computo Noruego, 1965
  5. John McLeod, Proceedings IBM Scientific Computing  
Symposium, Digital Simulation of Continuous Systems, 1967
  6. Geoffrey Gordon, System Simulation, Prentice Hall, 1969
  7. Thomas H. Naylor, Joseph L. Balintfy, Donald S. Burdick y  
Kong Chu, Computer Simulation Techniques, John Wiley, 1966
  8. Richard Bronson, Computer Simulation, BYTE, Marzo, 1984

**BIBLIOGRAFIA:**

1. Calahan D. A.  
Computer Aided Network Design  
Mc. Graw Hill  
N. Y. 1972
2. Chua L. O. / Ipen Min Lin  
Computer Aided Analysis of Electronic Circuits  
Prentice Hall  
1975
3. Preston E.J. / Crawsord G.W. / Coticchia M.E.  
CAD-CAM System Justification, Implementation, Productivity,  
Measurement.  
Marcel Dekker  
1984
4. Bruer M. A.  
Technics for the Simulation of Computer Project  
1974
5. George S. Fishman  
Conceptos y Metodos en la Simulacion Digital de Eventos  
Discretos.  
Limusa  
1978
6. Bryton R.K. / Chua L.O. / Rhodes J.D.  
CAD Systems and their Function  
Georgi Publishing  
1978, Suiza
7. Revista Communications ACM  
Julio, 1964  
pp. 443-446
8. Revista IEEE  
Transactions on Computer Volumen CXXI  
Abril, 1972  
pp. 399-402

9. Anthony Ralston, Editor  
Encyclopedia of Computer Science  
Van Nostrand Reinhold Company  
1976, Nueva York
  
10. Donald G. Fink, Editor  
Electronics Engineers' Handbook  
Mc Graw-Hill Book Company  
1982, Nueva York
  
11. Roger L. Tokheim  
Digital Electronics  
Mc Graw-Hill Book Company  
1979
  
12. TTL Logic Data Manual  
Signetics Corporation  
1982