

21 MICROELECTRÓNICA: CIRCUITOS INTEGRADOS ESPECIFICADOS POR EL USUARIO

- 21.1. Circuitos integrados programados por el usuario: tipos
- 21.2. Circuitos integrados diseñados por el usuario: tipos
- 21.3. Herramientas de diseño
- 21.4. Validación de circuitos integrados: simulación, análisis de tiempos y test

La palabra «microelectrónica» alude a la «electrónica» en tamaño «micro», es decir, a la configuración miniaturizada de los circuitos electrónicos. Ahora bien, añadir sin más la referencia al reducido tamaño de los circuitos electrónicos no deja de ser una redundancia: el desarrollo de la electrónica se encuentra marcado, desde los años 50, por un continuado proceso de miniaturización, del cual deriva su propia funcionalidad (su capacidad de abordar eficazmente funciones complejas) y su viabilidad (mediante equipos de tamaño, consumo y coste reducido).

A partir de los años 80, algunos autores han aprovechado, en forma relativamente oportunista y confusa, el título de «microelectrónica» en textos de electrónica realizada con circuitos integrados estándar, en cuyo contexto el prefijo «micro» no aporta ninguna precisión porque no hay «otro tipo de electrónica»: carece de sentido práctico (y estaría fuera de contexto tecnológico) un diseño que no utilice los circuitos integrados disponibles.

El término «microelectrónica», para que resulte significativo y conceptualmente relevante, debe ser entendido como la configuración del circuito electrónico completo, resultante de un diseño específico, en el interior de un solo circuito integrado. Ello supone pasar de la interconexión de circuitos integrados estándar a la configuración de un circuito integrado específico: se trata de insertar nuestro propio diseño completo en un integrado.

Es decir, en lugar del ensamblaje de «microcircuitos» genéricos (de catálogo) para configurar nuestro diseño, la microelectrónica trata de la realización de dicho diseño específico en un «microcircuito» particular; para lo cual disponemos de dos alternativas:

- programar nuestro diseño sobre un circuito integrado programable (CPLD, FPGA)
- o, fabricar dicho diseño como circuito integrado específico (ASIC).

Ambos casos (programación o fabricación del diseño) utilizan las mismas tecnologías de fabricación de los circuitos integrados (proceso de integración CMOS), el mismo esquema de desarrollo del proceso de diseño, el mismo tipo de herramientas y las mismas consideraciones metodológicas a tener en cuenta.

El presente capítulo describe las diversas posibilidades existentes en cuanto a circuitos integrados programables (PLDs, basados en la estructura PAL, y FPGAs, con estructura PGA) y respecto a la fabricación del propio diseño (ASIC); presenta las diversas herramientas necesarias para desarrollar un diseño en sus diferentes fases y opciones y reflexiona sobre tres aspectos diferenciados de la comprobación del diseño.

21.1. Circuitos integrados programados por el usuario: tipos

La finalidad de la microelectrónica es *configurar un circuito digital específico en el interior de un circuito integrado* (es decir, «insertar nuestro diseño en un integrado») y los circuitos integrados programables ofrecen una posibilidad de realizar este objetivo.

Un circuito integrado programable, adecuadamente grande, permite configurar, a través de su programación, un diseño digital específico y ello puede ser realizado directamente por el propio usuario.

En tal sentido, en la construcción de prototipos y en la fabricación de series limitadas la *lógica programada* sobre un único circuito integrado ha desplazado a las placas de múltiples circuitos integrados estándar. La utilización de integrados estándar, aún cuando puede resultar útil en laboratorio (para casos de diseños relativamente simples), ha sido sustituida en la práctica por el empleo de circuitos integrados programables, de forma que un solo circuito recoge el conjunto completo de funciones booleanas necesarias, evitando el cableado entre los diversos bloques digitales.

Los circuitos integrados programables presentan siempre una configuración interna modular: una *celda tipo* repetida múltiples veces; cada celda cuenta con una parte combinacional, sobre la que puede programarse una función booleana, seguida de un biestable. La parte combinacional de la celda o módulo básico permite establecer una división de los dispositivos programables en dos grandes grupos:

- dispositivos de tipo PAL: funciones expresadas en forma de suma de productos;
- dispositivos basados en multiplexores (funciones LUT), organizados en forma de matriz de módulos programables, tipo PGA.

Los dispositivos de tipo PAL están conformados por macroceldas (que incluyen las funciones como suma de productos seguidas de sendos biestables) y fueron estudiados en el apartado 4 del capítulo 13 (volumen 2). Las macroceldas han de contener sumas de términos producto de muchas variables; son módulos de «granularidad gruesa»: funciones de muchas variables de entrada (pudiendo cada módulo realizar una función compleja o bien desaprovechándose muchas de sus conexiones programables).

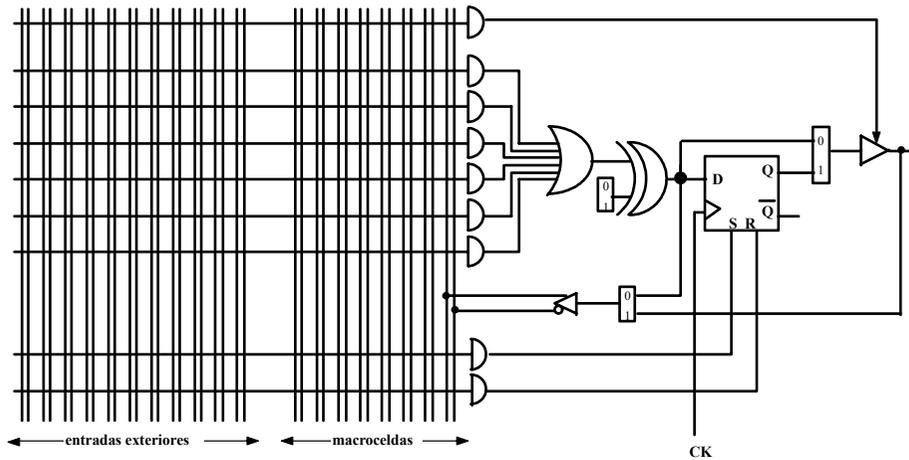
Los dispositivos de tipo PGA (*Programmable Gate Array*) corresponden a una «granularidad más fina», con funciones internas de pequeño número de entradas (4, 5 ó 6) y, en cambio, ofrecen mayor potencialidad en las conexiones entre módulos; están compuestos por una matriz de «celdas lógicas» (de reducido tamaño), separadas por canales de conexión verticales y horizontales, junto con «celdas de entrada/salida» situadas en la periferia, que conectan con los terminales del circuito.

Los módulos de tipo PAL son más potentes (y, por ello, más susceptibles de ser desaprovechados) en cuanto a su parte combinacional interna y se presentan en estructuras de interconexión (arquitecturas) relativamente directa: lineal o ramificada. Los dispositivos PGA contienen módulos más simples (y, por lo tanto, de limitada capacidad booleana) en una organización matricial con canales verticales y horizontales de interconexión, con una gran capacidad en cuanto a conexiones programables.

La manera más habitual de expresar una función booleana es su forma algebraica como suma de productos $\sum p$; la configuración reticular de varias funciones booleanas expresadas en forma de suma de productos recibe el nombre de **PAL**: cada función ocupa un módulo separado, conformado por una puerta "o" que recibe las salidas de varias puertas "y" sobre las que se programan los términos producto de la función.

Para poder programar sistemas secuenciales es necesario añadir a la configuración **PAL** la capacidad de memoria que proporcionan los biestables: la inclusión de un biestable en cada módulo, cuyos términos producto han de admitir como variables, además de las propias variables de entrada, las salidas de los biestables, para permitir la realimentación de las variables de estado.

El módulo básico resultante, «suma de productos + biestable», ha sido ampliado con diversos «selectores» programables, así como con la posibilidad de salida «tri-estado», y recibe el nombre genérico de *macrocela*.



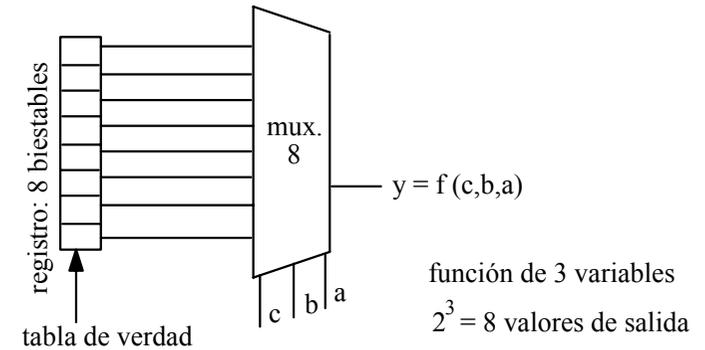
Los circuitos integrados programables que utilizan macroceldas suelen ser aludidos con las siglas **PLD** (dispositivo lógico programable) y **CPLD** (PLD complejo, cuando el número de módulos o macroceldas que contiene es alto).

En los **PLDs** pequeños cada macrocela está conectada a todas las demás (recibe la realimentación de las otras macroceldas). En los **CPLDs** las macroceldas se organizan en bloques, existiendo un canal de interconexión (líneas con conexiones programables para conectar los bloques entre sí y para comunicarlos con las entradas exteriores).

[Véase una descripción más detallada de los CPLDs en el apartado 13.4 del volumen 2 y en el siguiente capítulo, apartado 22.1, que incluye un repaso de dicha descripción.]

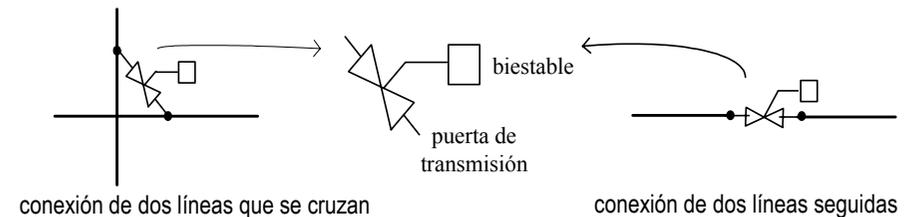
Matrices de celdas lógicas configurables: FPGAs

Una función booleana puede expresarse, también, a través de su tabla funcional y puede construirse mediante un multiplexor que «toma valores» sobre dicha tabla: configuración **LUT** (*look-up-table*).

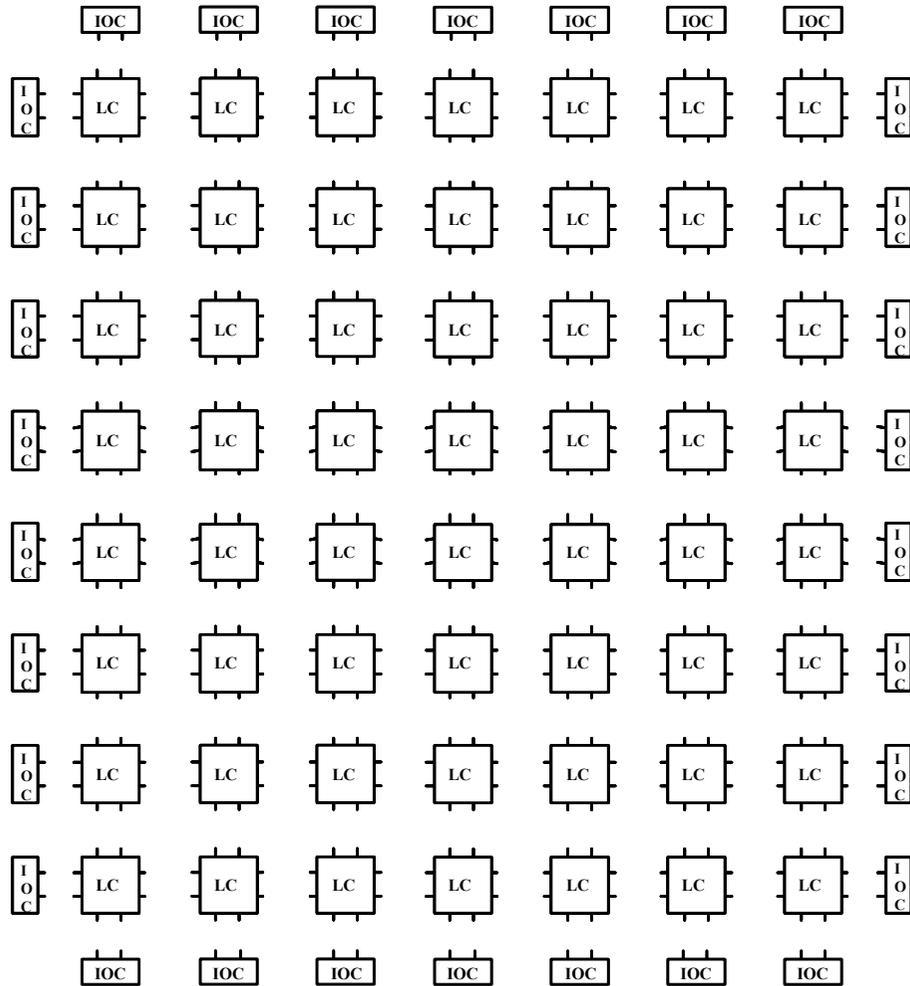


De esta forma, las funciones se configuran mediante su tabla de verdad que se memoriza en un «registro»: un multiplexor cuyas entradas son las salidas del registro selecciona (por «búsqueda en la tabla», **LUT**) la salida de la función.

Asimismo, las conexiones entre funciones (o entre módulos) pueden ser controladas mediante biestables: una puerta de transmisión entre dos líneas permite su conexión cuando la puerta conduce o su separación en caso de que se encuentre en corte; la situación de conducción o corte de la puerta de transmisión puede ser determinada por el valor booleano almacenado en un biestable. Téngase en cuenta que las puertas de transmisión, en tecnología CMOS, son bidireccionales: la conexión que establecen entre dos líneas puede actuar en ambas direcciones.



Los circuitos programables resultantes de esta configuración (funciones LUT y conexiones por puertas de transmisión) disponen de un conjunto de celdas lógicas (LC), organizadas matricialmente en dos dimensiones; a través de las franjas (verticales y horizontales) de separación entre ellas se establecen canales (conjuntos de líneas) para la interconexión de las celdas y, además, en la periferia se disponen celdas específicas de adaptación de las entradas y salidas (IOC).



Estos circuitos programables reciben el nombre de PGA (*Programmable Gate Array*): matrices de puertas programables, entendiendo el término puerta (*G gate*) en su acepción amplia de módulo lógico (como celda que contiene una función booleana de pocas variables). Su denominación habitual es FPGA (*Field Programmable Gate Array*), donde la F inicial (*field*: in situ), viene a indicar que la programación por el usuario puede hacerse en la propia placa circuital de aplicación, sin tener que extraer la FPGA del circuito de que forma parte.

Las matrices de celdas programables FPGAs son dispositivos de «granularidad fina», compuestos por una amplia matriz de celdas lógicas de reducido tamaño (y limitada capacidad booleana), separadas por canales de conexión verticales y horizontales, y por celdas de entrada/salida situadas en la periferia, que conectan con los terminales del dispositivo. El calificativo «programable» adjetiva tanto a las celdas (de ambos tipos: lógicas y de entrada/salida) como a la matriz de conexiones, es decir, tiene una doble repercusión: son programables tanto las celdas internamente (su función booleana y su configuración) como las conexiones de las celdas entre sí.

El próximo capítulo (cap. 22) se dedica a los circuitos integrados programables, repasando brevemente la configuración y características de los CPLDs y describiendo en detalle las FPGAs.

21.2. Circuitos integrados diseñados por el usuario: tipos

El extraordinario desarrollo alcanzado por las tecnologías de integración MOS y la continuada reducción de costes de los circuitos fabricados con ellas, junto con la sistematización de los procesos de diseño y la disponibilidad de eficaces herramientas CAD, han permitido que el diseño de los circuitos integrados sea realizado por los propios usuarios de los mismos.

De esta manera, cualquier diseñador de sistemas digitales puede abordar el diseño de su propio circuito integrado y, por tanto, cualquier empresa puede incorporar a los productos que fabrica circuitos integrados propios, diseñados específicamente para el producto en cuestión y pensados desde la perspectiva de mejorar, ampliar y personalizar las prestaciones de dicho producto.

La utilización de circuitos integrados específicos no sólo resulta viable, sino también económicamente rentable; aun más, contribuye en gran medida a aumentar la competitividad de los productos fabricados en serie, por su repercusión en las prestaciones y en el precio de los mismos.

Un ASIC (*application specific integrated circuit*) es un circuito integrado, no disponible en catálogo, diseñado para una aplicación concreta, de forma que sus prestaciones se ajustan específicamente a dicha aplicación.

En realidad, el término ASIC puede aplicarse tanto a los circuitos integrados programados por el usuario como a los fabricados según el diseño del mismo; en ambos casos, tales circuitos integrados resultan configurados en forma «no estándar», para aplicaciones específicas. Ahora bien, suele ser habitual reservar la denominación de ASICs para los circuitos fabricados con diseño propio y, en tal sentido, lo utilizaremos en este texto.

Un **ASIC** permite sustituir una o varias placas de circuitos integrados estándar, reduciendo drásticamente su coste, su tamaño y su consumo y aumentando su fiabilidad y su velocidad así como las prestaciones del circuito global; además un **ASIC** personaliza el circuito electrónico, asegurando en gran medida su confidencialidad, es decir, la dificultad de reproducción del mismo mediante duplicación de las placas circuitales o mediante el rediseño de las mismas con ligeras modificaciones.

En relación con una placa de circuitos integrados estándar, un **ASIC** presenta:

↑ menor coste	↑ mayor velocidad	↓ mayor inversión inicial NRE
↑ menor consumo	↑ mayores prestaciones	↓ necesidad de un n° mínimo
↑ menor tamaño	↑ mayor fiabilidad	↓ menor flexibilidad al cambio
↑ menor peso	↑ mayor confidencialidad	↓ test más complejo

pero, también, supone:

La fabricación de circuitos integrados específicos ofrece un coste por ejemplar sumamente inferior al coste global de la placa (o placas) de circuitos integrados estándar a los que sustituye y muy inferior, también, al coste de un circuito integrado programable equivalente.

Se requiere, en cambio, una inversión inicial (**NRE**: *non recursive engineering*) relativamente elevada, referente al diseño, desarrollo y construcción de las máscaras de integración; inversión que es independiente del número de ejemplares a fabricar. La repercusión de la inversión **NRE**, en el coste de cada unidad, es pequeña cuando el número de unidades que posteriormente se integran es de varios miles; de otro modo, el coste de fabricación individual se verá fuertemente incrementado por la amortización de dicha inversión inicial.

La última etapa del diseño a nivel físico de un circuito integrado, la que conecta con el proceso de fabricación, tiene un carácter netamente geométrico, para culminar en el dibujo de las máscaras que determinan, en cada una de las etapas del proceso de integración, la distinción entre unas y otras zonas superficiales de la oblea; respetando, además, las reglas de diseño que impone la tecnología de integración.

Existen tres posibilidades para abordar el diseño y la fabricación de un circuito integrado, según se expresa en el siguiente diagrama:

	<i>fabricación completa</i>	<i>partir de una matriz de puertas prefabricadas</i>
<i>diseñar completamente cada una de sus partes</i>	diseño físico completo FULL CUSTOM	
<i>utilizar una librería de celdas prediseñadas</i>	celdas prediseñadas STANDARD CELL	puertas prefabricadas GATE ARRAY

Un *diseño físico completo (full custom)* incluye el dibujo de las máscaras por parte del propio diseñador, ayudado por las correspondientes herramientas **CAD**. Este tipo de diseño completo puede resultar sumamente complejo; en cambio, permite, si el diseñador es suficientemente hábil, un óptimo aprovechamiento de la superficie de la oblea de silicio. De esta forma se diseñan los circuitos integrados estándar, cuyo alto volumen de producción justifica el esfuerzo de un diseño físico completo, en orden a minimizar la superficie de silicio de cada integrado y, con ello, su precio.

Sin embargo, el diseño de un **ASIC** no suele descender hasta el nivel de las máscaras de fabricación, sino que aprovecha librerías de celdas lógicas prediseñadas (puertas, biestables, pequeños bloques combinatoriales, reducidos registros o contadores,...). El circuito específico se describe a través de la correspondiente interconexión de tales celdas y, una vez capturado su esquema, el compilador lo traduce automáticamente a las correspondientes máscaras físicas (utilizando para ello potentes algoritmos de colocación óptima de las celdas, *placement*, y de conexionado de ellas, *routing*).

A partir del diseño con *celdas prediseñadas (standard cell)* pueden obtenerse todas las máscaras necesarias para fabricar el correspondiente circuito integrado, personalizadas todas ellas para cada **ASIC**, y, consiguientemente, puede efectuarse el proceso completo de fabricación del mismo.

Otra posibilidad consiste en disponer de circuitos integrados parcialmente prefabricados, con un gran número de puertas en su interior, y configurar el circuito digital específico mediante el conexionado sobre dichas puertas. Este tipo de fabricación que utiliza una *matriz de puertas prefabricadas (gate array* o, también, *sea of gates)* simplifica el proceso de fabricación de cada **ASIC** individual, pues solamente las máscaras correspondientes al conexionado son personalizadas para cada **ASIC**.

En ambos casos (*standard cell* y *gate array*) el diseño se realiza utilizando una librería de celdas estándar y no es muy diferente del diseño clásico con circuitos integrados estándar, seleccionándolos en el correspondiente catálogo y realizando la correspondiente interconexión entre ellos.

La *fabricación completa de celdas prediseñadas (standard cell)* permite un buen aprovechamiento de la superficie de la oblea de silicio, por cuanto que se incorporan solamente las celdas funcionalmente necesarias, ya optimizadas en su diseño, y tales celdas se posicionan (mediante algoritmos de *placement* y *routing*) de forma que su interconexión sea lo más directa posible; por tanto permite fabricar un mayor número de unidades en cada oblea.

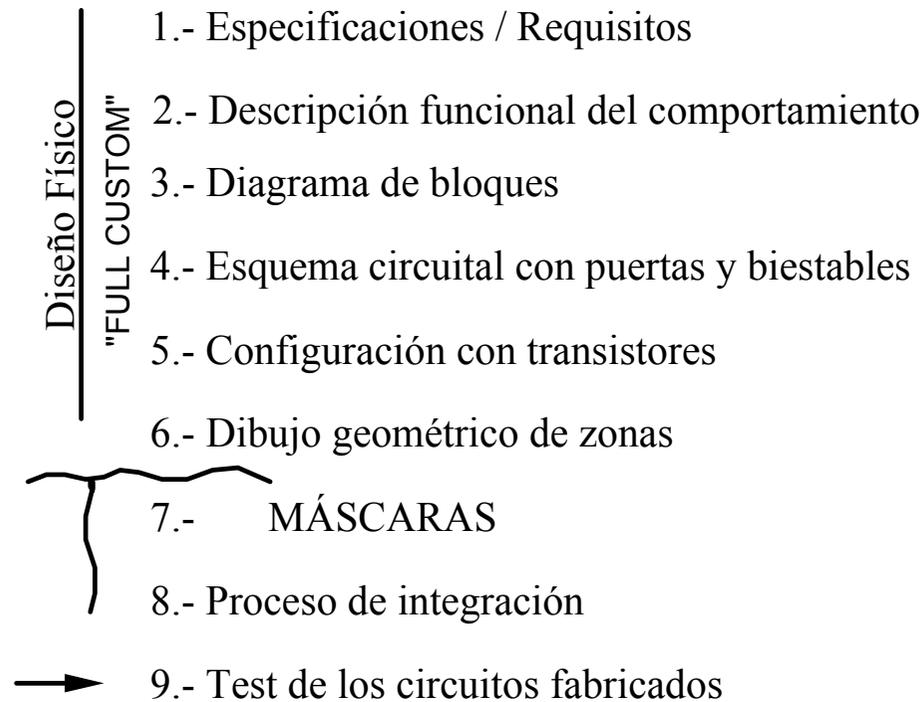
En cambio la utilización de *matrices de puertas prefabricadas (gate array)* requiere mayor área (no es posible utilizar todas las puertas prefabricadas, ni tampoco conseguir un óptimo aprovechamiento de las mismas), pero disminuye fuertemente el coste de la inversión inicial **NRE**, ya que solamente es necesario personalizar las máscaras y las etapas de fabricación correspondientes al conexionado (es decir, a las líneas de metal).

La elección entre uno u otro método de fabricación de un **ASIC** (*standard cell* o *gate array*) depende del número de ejemplares a integrar, es decir, al término de coste que predomina: el del proceso de fabricación concreta de cada ejemplar o el de la inversión inicial **NRE** distribuida entre todos los ejemplares fabricados.

En todo caso, el proceso de diseño incluirá la descripción del circuito, su simulación funcional (comprobar que funciona conforme se desea) y la generación de los vectores de test que permitan la comprobación física de cada circuito una vez fabricado; todo ello sobre computador, utilizando potentes herramientas CAD.

Diseño físico completo

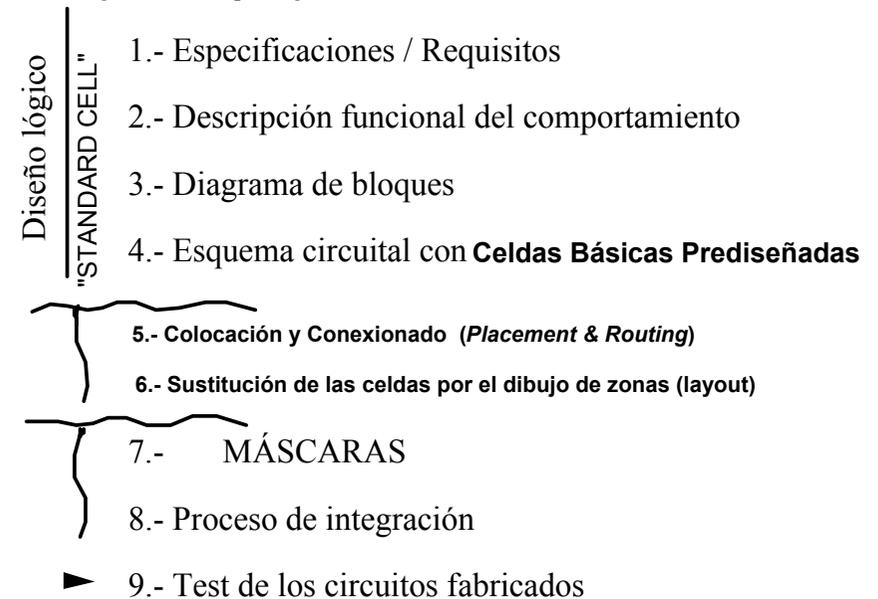
Los procesos de diseño completo (*full custom*) de un circuito integrado, junto con su fabricación, abarcan las siguientes etapas:



En un diseño completo, la tarea del diseñador parte de las especificaciones o requisitos que debe cumplir el circuito hasta llegar a su dibujo geométrico (*layout*), que dará lugar a las máscaras que el fabricante utilizará en el proceso de fabricación. También, es responsabilidad del diseñador la obtención de los vectores de test que el fabricante ha de aplicar para verificar cada uno de los circuitos resultantes del proceso de fabricación.

Diseño con celdas prediseñadas

Otra posibilidad de diseño consiste en utilizar una librería de celdas prediseñadas (*standard cell*) por el fabricante, de forma que el diseñador no ha de descender al nivel físico de los transistores que configurarán el circuito, sino que efectúa el diseño en el nivel lógico (booleano). Los procesos de diseño y fabricación utilizando una librería de celdas seguirán las etapas siguientes:



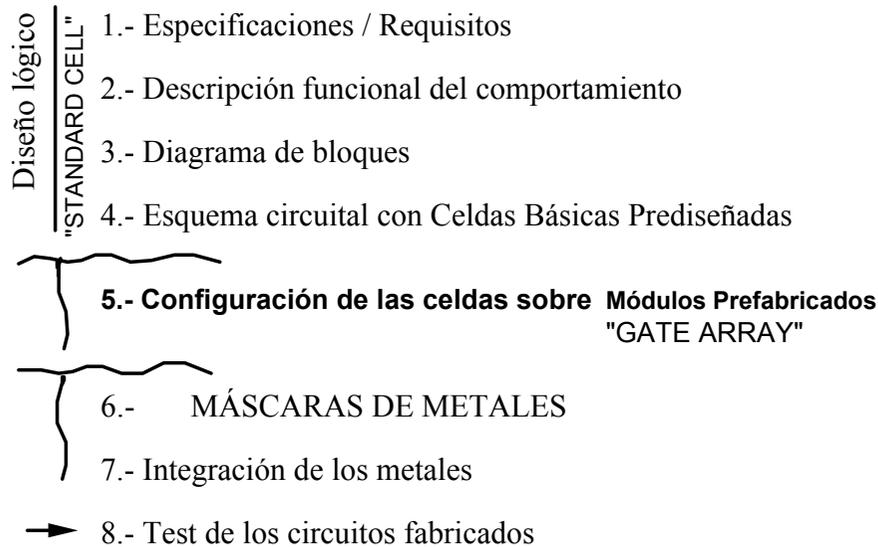
Las celdas básicas incluidas en una librería tienen dimensiones geométricas ajustadas entre sí, para facilitar el encaje de unas con otras. Las tareas de ubicación de las celdas y conexinado de las mismas (*placement and routing*) son ejecutadas con la ayuda de potentes herramientas informáticas y suelen ser realizadas por el diseñador, en contacto y colaboración directa con el fabricante. La sustitución de las celdas por su *layout* es realizada directamente por el fabricante.

La descripción del circuito puede ser gráfica (captura de esquemas) o textual (lenguajes de descripción circuital o descripción de *hardware*) o mezcla de ambas. A medida que aumenta la complejidad de los circuitos integrados resultan cada vez más útiles los lenguajes de descripción circuital (VHDL, Verilog,...), que permiten describir un circuito a través de su funcionamiento, sin descender a su configuración (puertas, biestables o bloques); de construir tal configuración se encarga el correspondiente compilador.

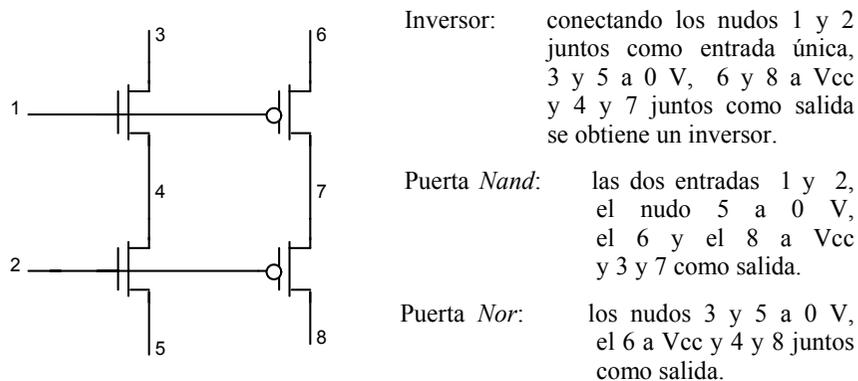
Fabricación con matrices de puertas prefabricadas

En cuanto al proceso de fabricación, una posibilidad de reducir costes consiste en utilizar circuitos integrados ya fabricados a nivel de puertas (*gate array*) o de módulos equivalentes, sobre los cuales se configurarán (por conexión mediante las capas de metales) las celdas de la librería.

De esta forma, el circuito particularizado sólo requiere las máscaras correspondientes a los metales y a sus contactos:



La configuración típica de una puerta o módulo básico *gate array* suele ser del tipo representado en la figura siguiente:



21.3. Herramientas de diseño

El proceso de diseño de un circuito integrado digital (en sus diferentes modalidades) se desarrolla sobre computador, en forma de tareas de tipo CAD, para cuya ejecución se dispone de eficaces y potentes herramientas informáticas. Las herramientas que se utilizan para el diseño digital con circuitos programables y para el diseño de ASICs digitales son básicamente las mismas y se emplean de la misma forma, diferenciándose luego en la forma de compilar el diseño y en los resultados de tal compilación.

El diseño de un circuito integrado digital es básicamente un proceso de descripción del mismo, junto con la simulación de su comportamiento para verificar que el diseño es correcto, conforme a las especificaciones o requisitos del mismo. La descripción digital puede hacerse a diferentes niveles, en particular, a nivel geométrico (*layout*), a nivel lógico (esquemático) o a nivel funcional («programa»).

El nivel geométrico (descripción física del circuito integrado) no tiene sentido en el diseño sobre circuitos integrados programables (pues ya están fabricados) y es utilizado en forma muy limitada en el diseño de ASICs con librería, es decir en el diseño *standard cell* o *gate array* (solamente se usan algoritmos de colocación y conexionado de las celdas de la librería, que ya están diseñadas).

Los niveles lógico y funcional ofrecen dos alternativas de diseño: en forma gráfica de esquema de puertas y biestables, adecuadamente conectadas, o en forma de texto, describiendo el comportamiento del circuito; actualmente, y cada vez más, predomina la descripción en texto, utilizando un lenguaje de descripción circuital (HDL, *Hardware Description Language*).

Existen, pues, tres niveles de descripción de los diseños digitales:

- nivel geométrico (*layout*): descripción gráfica de las diferentes regiones físicas que conforman el circuito integrado (difusiones, polisilicio, metal, contactos,...), o sea, el dibujo que corresponde a las máscaras con las que se fabricará el integrado; este nivel es propio y casi exclusivo del diseño completo (*full custom*),
- nivel lógico (esquemático): descripción, en términos de puertas lógicas y biestables; es el nivel que corresponde directamente al diseño con librería,
- nivel funcional (texto): descripción del comportamiento del circuito en un lenguaje de descripción circuital (como puede ser VHDL o Verilog).

El tipo de tareas que permiten abordar las herramientas informáticas, según los tres niveles anteriores, está resumido en la tabla de la página siguiente.

1 Nivel Funcional**→ Lenguajes de Descripción Funcional**

- Edición y análisis de la descripción.
- Simulación funcional.
- Compilación lógica.

2 Nivel Lógico**→ Descripción**

- Captura de esquemas.
- Generación de celdas matriciales (ROM, PLA, RAM, ...).

→ Simuladores Lógico-temporales**→ Análisis Temporal**

- Verificación de tiempos de anticipación y de mantenimiento.
- Cálculo de retardos y detección de caminos críticos.

→ Test

- Simulación de fallos.
- Cobertura de test.
- Generación automática de test: ATPG.

3 Nivel Geométrico**→ Descripción**

- Colocación y conexionado de celdas: *Placement and Routing*.
- Edición gráfica.

→ Simulación

- Extractores de parámetros: *back-annotation*.
- Simuladores Eléctricos.

→ Comprobación Lógica

- Extractores lógicos.
- Verificación de correspondencia.

→ Verificadores de reglas de diseño: DRC.

En todos los niveles es preciso comprobar que el diseño es correcto, a través de la simulación de su comportamiento:

- en el nivel geométrico, se utilizan simuladores eléctricos, en particular el simulador SPICE, extrayendo previamente, sobre el *layout*, los transistores y los componentes capacitivos y resistivos presentes;
- en el nivel lógico, se emplean simuladores lógico-temporales que actúan en términos de funciones booleanas y de retrasos temporales (tiempos de propagación);
- en el nivel funcional, los lenguajes de descripción circuital ofrecen simuladores del comportamiento del circuito, tal como queda descrito en dichos lenguajes.

Además, en el nivel lógico se aplica el análisis de tiempos para verificar que se respetan los tiempos de propagación, de anticipación y de mantenimiento (y, en el caso de ASICs, se obtiene el conjunto de vectores de test necesarios en el proceso de integración para comprobar que cada ejemplar fabricado responde efectivamente al diseño).

En el nivel funcional (lenguajes de descripción circuital), se dispone de compiladores que trasladan la descripción al nivel lógico. En dicho nivel lógico, los compiladores para circuitos integrados programables, traducen el diseño en el «mapa de fusibles» que corresponde a su programación; en el caso de ASICs (*standard cell* y *gate array*), del nivel lógico se pasa al nivel geométrico mediante algoritmos de ubicación y conexionado que sustituyen la descripción lógica por la conexión de las correspondientes celdas de la librería, adecuadamente situadas.

El compilador para CPLDs trasladará la descripción (esquemática o en texto) a funciones booleanas, expresadas en suma de productos, y biestables tipo D y tratará de encajar dichas funciones y biestables en las macroceldas que conforman el dispositivo programable. El resultado será un «mapa de fusibles» que describe la forma en que deben quedar los diversos términos producto y, en su caso, las conexiones entre macroceldas; un programador ejecutará dicho mapa de fusibles sobre el correspondiente circuito integrado programable.

El compilador para FPGAs pasará a funciones booleanas de un número de variables limitado, descomponiendo cada función en varias si es preciso, para insertarlas en las celdas lógicas que componen el dispositivo; el «mapa de valores» (denominación que es más adecuada que la de «mapa de fusibles», habida cuenta de que predomina en este caso la programación de tipo SRAM) describirá las funciones booleanas, mediante su «tabla de verdad», la configuración de las celdas y las conexiones entre ellas.

En el caso de ASICs, la descripción funcional se compila utilizando la librería de celdas disponible para pasarla a un esquemático de celdas (nivel lógico) y, posteriormente, se traslada a nivel geométrico, mediante los algoritmos de colocación y conexionado de las celdas (cuya dimensión geométrica ya se conoce pues su *layout* se ha diseñado previamente).

21.4. Validación de circuitos integrados: simulación, análisis de tiempos y test

En el proceso de diseño y fabricación de un circuito integrado se utilizan tres tipos de procedimientos de simulación, verificación y supervisión que tienen objetivos y metodologías de ejecución totalmente diferentes:

- la *simulación funcional*
- el *análisis de tiempos*
- y el *test del circuito fabricado*.

Estos tres procedimientos se refieren, respectivamente, a la verificación del comportamiento del circuito resultante del proceso de diseño, a la verificación de los tiempos implicados en tal circuito y a la verificación de cada circuito integrado obtenido tras el proceso de fabricación:

1 verificación del comportamiento

- **Simulación** → **Funcional**
- **Lógica-temporal**
- **Eléctrica**

2 verificación de tiempos

- **Análisis de tiempos** → **Tiempos de anticipación y de mantenimiento**
- **Tiempos de propagación y caminos críticos**

3 verificación del circuito obtenido

- **Test** → **Simulación de fallos**
- **Cobertura de fallos**

La *simulación funcional* trata de comprobar si el diseño realizado funciona adecuadamente, es decir, si se ajusta a las especificaciones o requisitos que se pretendían alcanzar. Consiste en reproducir el funcionamiento real del circuito para verificar que es correcto, incluyendo las más diversas situaciones posibles de sus entradas y comprobando la ausencia de errores en la respuesta del circuito. Es un estudio estímulo/respuesta (entradas/salidas) en relación con el funcionamiento normal del circuito; se realiza con simuladores lógico-temporales que evalúan la respuesta booleana a los vectores de entrada y calculan también el tiempo de respuesta: cada celda básica es sustituida por la correspondiente función booleana y por un tiempo de propagación que tiene en cuenta la carga (*fan out* y capacidad equivalente) que soporta su salida.

El *análisis de tiempos* trata de comprobar que se respetan las restricciones temporales internas (compatibilidad entre los tiempos de propagación y los tiempos de anticipación *set-up* y de mantenimiento *hold*) y de calcular la velocidad máxima de trabajo del circuito; asimismo, aporta información sobre los caminos en que se violan las restricciones temporales y sobre los caminos que resultan críticos respecto a la velocidad máxima de funcionamiento del circuito.

El análisis de tiempos se efectúa por caminos: un camino es el recorrido existente entre dos biestables, se inicia en la salida de un biestable y finaliza en la entrada de otro (sin atravesar ningún biestable intermedio). Se realiza mediante un cálculo directo de tiempos entre la entrada de cada biestable y las salidas de aquellos biestables que influyen sobre ella: cada celda básica combinacional es sustituida por sus tiempos de propagación (teniendo en cuenta en ellos la situación real de carga *-fan out* y capacidad equivalente) que les afecta) y para cada biestable se tienen en cuenta sus tiempos de propagación (biestable origen) y los de anticipación y de mantenimiento (biestable final del camino).

El análisis de tiempos no tiene en cuenta, en cambio, la función concreta que el circuito va a realizar: no utiliza vectores de entrada y de salida, ni efectúa la simulación del circuito sino que meramente calcula tiempos entre biestables; supone de partida que el diseño es síncrono y los tiempos se calculan con referencia al flanco activo del reloj.

Ambos procedimientos de comprobación/verificación (*simulación funcional* y *análisis de tiempos*) se ejecutan en las diferentes etapas del diseño: tanto en el nivel lógico, sobre el esquemático del circuito, como en el nivel geométrico, sobre las máscaras que conectan con el proceso de fabricación. Cuanto más cercana es la información que utilizan al nivel físico definitivo del circuito más precisa es la simulación temporal y el análisis de tiempos: cuanto más próximos nos situemos a la configuración física real del circuito integrado, mayor precisión puede obtenerse en los aspectos de carga efectiva que soportan las salidas de las celdas básicas y en los retardos debidos a los efectos resistivos y capacitivos que introducen las conexiones (es decir, en aquellas cuestiones que afectan a los cálculos temporales).

Obviamente, el análisis de tiempos es la verificación propiamente «temporal», pero también la simulación funcional tiene en cuenta el tiempo ya que involucra dos aspectos: el lógico relativo a las funciones booleanas que ejecutan las celdas y el temporal correspondiente a sus tiempos de propagación; es una simulación *lógico-temporal*.

La *simulación funcional* se efectúa «globalmente» (calculando las salidas que produce el circuito en función de las ondas de entrada que recibe) y *suministra información sobre la validez del diseño* (sobre si es correcto o no); en su caso, obliga a rediseñar parcial o totalmente el circuito para que su funcionamiento coincida con el que se pretende.

El análisis de tiempos se efectúa por caminos (calcula tiempos entre la entrada de un biestable y cada una de las salidas de los biestables que le afectan), ayuda a aumentar la velocidad de trabajo del circuito (detecta los caminos críticos, aquellos que presentan mayor tiempo de propagación entre biestables) y proporciona información sobre las violaciones de los tiempos de mantenimiento (*hold*) de los biestables. Los tiempos de anticipación (*set-up*) de los biestables intervienen en la velocidad máxima del circuito sumándose a los tiempos de propagación en los diversos caminos; asimismo, aun no existiendo problemas de velocidad, han de respetarse los tiempos de mantenimiento.

La violación de un tiempo de *set-up* requiere reducir los tiempos de propagación intermedios (disminuyendo el número de puertas intermedias o utilizando puertas más rápidas) o bien admitir una velocidad de trabajo inferior (mayor período de reloj, es decir, mayor intervalo temporal para formar el nuevo dato y presentarlo en la entrada del biestable). La violación de un tiempo de *hold* requiere aumentar el tiempo de permanencia del dato en la entrada del biestable respecto del flanco activo del reloj, es decir, aumentar el tiempo de propagación a partir de la salida del biestable anterior (incluyendo un par de inversores para retrasar la señal o utilizando puertas de mayor tiempo de propagación). Obviamente puede suceder que la corrección de una violación de *hold* produzca una violación de *set-up* y, consiguientemente, exija una velocidad de trabajo menor.

Tanto la simulación funcional como el análisis de tiempos son comprobaciones que se ejecutan en el proceso de diseño y pueden determinar una vuelta atrás en el mismo para corregir lo que «no funciona bien». En cambio, el test de un circuito integrado es la comprobación, posterior a la fabricación del mismo, que garantiza que el circuito se ha fabricado bien, es decir, que todas las celdas que configuran el circuito han sido integradas correctamente de forma que el resultado físico coincide con el esquema circuital resultante del diseño.

El test no equivale, ni mucho menos, a la simulación funcional: no se trata de verificar que el funcionamiento «normal» del circuito es el deseado sino de comprobar que todos y cada uno de los transistores han sido fabricados y conectados correctamente. Téngase en cuenta que, si el circuito es relativamente complejo, su funcionamiento «total» no puede ser simulado: requeriría una secuencia enormemente alta de vectores de test.

El test es, por tanto, mucho más exigente que la simulación funcional: que el diseño es correcto queda garantizado, en la misma etapa de diseño, por la simulación funcional y, complementariamente, por el análisis de tiempos en lo que se refiere a restricciones temporales; con el test de lo que se trata es de garantizar, para cada uno de los circuitos integrados, que tal diseño ha sido fabricado correctamente, es decir, que cada uno de los circuitos físicos obtenidos (*chips*) corresponde exactamente al diseño.

El test va más allá de una comprobación funcional de los casos de interés: los vectores de test pretenden verificar que todos los nudos booleanos presentes en el circuito integrado son capaces de actuar correctamente, lo cual asegura, en gran medida, que el circuito físico responde efectivamente al diseño efectuado y no contiene errores debidos al proceso de integración. En la práctica, es imposible la comprobación funcional completa de un circuito digital complejo, recorriendo todos los casos posibles; el test permite efectuar una comprobación «completa» de todos los nudos booleanos del mismo, conforme a un modelo conceptual establecido para tal finalidad.

El test se efectúa, también, mediante un estudio estímulo/respuesta (entradas/salidas) utilizando una secuencia apropiada de vectores test (que no coincide con la secuencia propia de la simulación funcional, aunque ésta puede formar parte del conjunto global de vectores de test). Los vectores de test han de permitir «controlar», desde las entradas del circuito integrado, cada uno de los «nudos booleanos» internos del mismo (pudiendo situarlos a valor 0 y a valor 1) y «observar», desde las salidas del circuito integrado, el estado booleano de cada uno de dichos nudos; de esta forma, podremos comprobar que cada celda básica está en su sitio y que actúa correctamente.

Aun cuando la aplicación del test es posterior a la fabricación, afecta directamente al diseño por cuanto que el circuito ha de ser «testeable» (no todos lo son) y, además, la obtención de los vectores de test forma parte del diseño del circuito. Es decir, todo el proceso de diseño ha de estar orientado al test; no basta un diseño funcional sino que hay que efectuar, a la vez, un diseño para el test, siendo así que, en muchas ocasiones, los aspectos relativos al test resultan más difíciles y complejos que el propio diseño funcional. [El capítulo 25 se dedica íntegramente al test de circuitos integrados digitales.]