

En este capítulo se abordan los distintos tipos de Dispositivos Lógicos Programables, incluyendo su definición, su clasificación y una breve explicación de cada tipo de PLD.

### **2.1 Definiciones**

Un Dispositivo Lógico Programable (PLD) es un componente electrónico usado para construir circuitos digitales reconfigurables. A diferencia de una compuerta lógica que tiene una función fija, los PLDs salen de fábrica sin una función en específico, por lo tanto necesitan ser programados o reconfigurados antes de poder ser usados.

Los PLDs tienen varias ventajas. La primera es la habilidad de integración, que permite integrar una gran cantidad de funcionalidad en un solo chip. Los PLDs eliminan el uso de múltiples chips así como la inconveniencia y desconfianza de usar cableado externo. La segunda ventaja es el hecho de poder cambiar el diseño. Muchos PLDs permiten ser reprogramados o reconfigurados.

Existen dos ramas principales dentro de los dispositivos lógicos programables, la lógica programable de campo y la de fábrica. El término campo en este contexto implica que los dispositivos puedan ser programados en el “campo” del usuario, mientras que la lógica de fábrica puede ser programada en la misma fábrica donde se construyen, de acuerdo a los requerimientos del cliente. En este caso, la tecnología de programación usa procesos irreversibles, por lo que solo es posible hacerlo una vez.

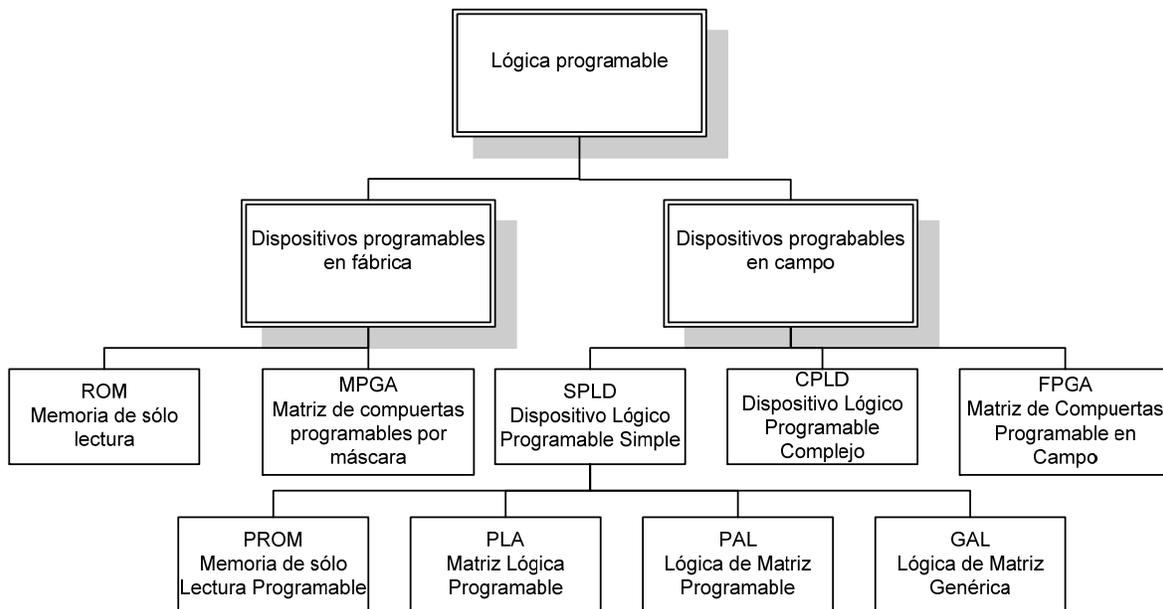


Figura 2.1 Árbol de clasificación de PLDs.

Algunos ejemplos de lógica programable de fábrica son los MPGAs y memorias de sólo lectura (ROMs). Las primeras generaciones de muchos dispositivos programables también fueron programados únicamente en fábrica. Las ROMs son consideradas como lógica programable porque, aunque fueron concebidas como unidades de memoria, también sirven para implementar cualquier circuitería combinatorial. Los MPGAs son arreglos de compuertas tradicionales que requieren una máscara para ser diseñados. Los MPGAs son también llamados simplemente gate arrays y han sido la tecnología popular para crear ASICs (*Application Specific Integrated Circuits*).

La lógica programable por el usuario basada en compuertas AND y OR fue desarrollada al inicio de la década de los '70s. Para 1972-73 ya estaban disponibles arreglos lógicos programables una sola vez que permitían personalizaciones instantáneas para diseñadores. Algunos se refieren a estos circuitos como FPLAs (*Field Programmable Logic Arrays*).

Monolithic Memories Inc. (MMI), una compañía comprada por Advanced Micro Devices (AMD) creó circuitos integrados llamados PLAs (*Programmable Logic Arrays*) que podían tener el mismo rendimiento y funcionalidad que 5 a 20 chips comerciales. Un dispositivo similar es el PAL (*Programmable Array Logic*).

PALs y PLAs contienen arreglos de compuertas. En el PLA se tiene un arreglo de compuertas AND programable y un arreglo de compuertas OR programable, permitiendo a los usuarios implementar funciones combinatoriales en dos niveles de compuertas. El PAL es un caso especial del PLA, ya que el arreglo de ORs es fijo y el único arreglo programable es el de compuertas AND. Muchas PALs también contienen flip flops.

Los primeros dispositivos programables permitían ser programados una sola vez. El siguiente avance tecnológico fue el poder borrar los dispositivos, lo cual al inicio se hacía por medio de luz ultravioleta, que significaba remover el dispositivo del circuito para ponerlo en un ambiente ultravioleta. Este proceso era lento (de 10 a 15 minutos) y no permitía borrar la información en el circuito. El siguiente avance fue la tecnología de borrado eléctrico, que permitió la creación de dispositivos que se podían borrar rápida y fácilmente así como ser reprogramados sin necesidad de removerlo del circuito.

## **2.2 SPLDs**

Los PLAs, PALs, GALs y ROMs son llamados SPLDs (*Simple Programmable Logic Devices*) a partir del surgimiento de los CPLDs (*Complex Programmable Logic Devices*) los cuales básicamente contienen múltiples PLDs en el mismo chip.

En esta sección se hablará brevemente sobre los principales SPLDs.

### 2.2.1 ROM

Una ROM consiste en un arreglo de dispositivos semiconductores que están interconectados para almacenar de datos binarios. Una vez almacenada la información, puede ser leída cuando se requiera, pero no puede ser modificada bajo condiciones normales de operación.

Las ROMs tienen combinaciones de entradas, que generalmente son llamadas direcciones, y patrones de salidas, llamadas palabras. Una ROM que tiene  $n$  líneas de entrada y  $m$  líneas de salida contiene un arreglo de  $2^n$  palabras, cada una de  $m$  bits de longitud. La dirección sirve para seleccionar una de las  $2^n$  palabras, por lo que cuando una combinación de entrada es aplicada a la ROM, el patrón de ceros y unos almacenados en la palabra correspondiente aparece en las líneas de salida.

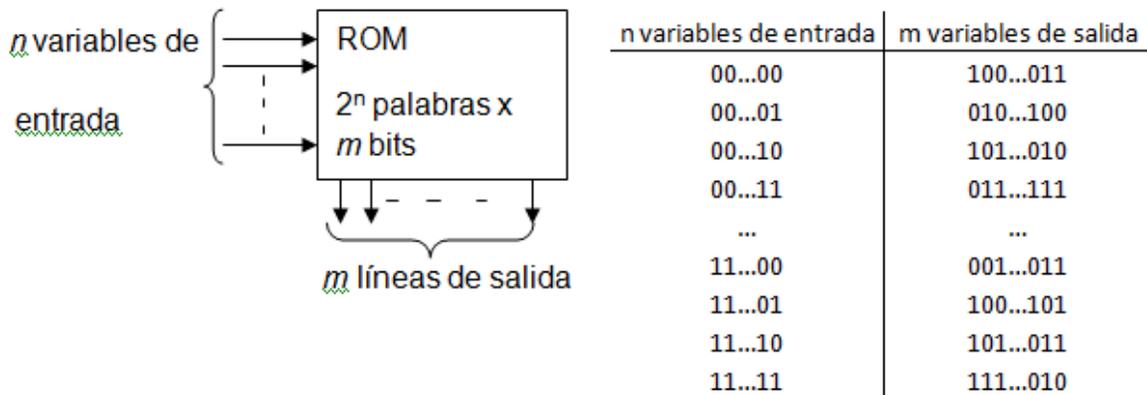


Figura 2.2 Ejemplo de una ROM y su tabla de verdad.

Una ROM consiste básicamente de un decodificador y un arreglo de memoria. Cuando un patrón de entrada se aplica a las entradas del decodificador, una de las  $2^n$  salidas de dicho decodificador se activa, seleccionando una de las palabras almacenadas en la memoria y se transfiere a las líneas de salida.

Los tipos básicos de ROM incluyen ROMs programables por máscara, ROMs programables por el usuario (PROMs), ROMs programables borrables (EPROMs), ROMs programables borrables eléctricamente (EEPROMs) y memorias flash.

En las ROM programables por máscara, el arreglo de datos se almacena permanentemente a la hora de la manufactura. Este proceso resulta caro debido a que se requiere preparar una máscara especial, usada en la fabricación del dispositivo.

Las EPROMs se programan mediante un programador que provee pulsos de voltaje apropiados para almacenar cargas eléctricas en los lugares de memoria y se borran generalmente usando luz ultravioleta, mientras que las EEPROMs se borran por medio de pulsos eléctricos. Las EEPROMs tienen un número limitado de veces que pueden ser borradas y reprogramadas, típicamente entre 100 y 1000 veces. Las memorias flash son similares a las EEPROMs salvo que usan un mecanismo diferente de carga y almacenaje.

Una ROM puede implementar cualquier circuito combinacional. Si las salidas de todas las combinaciones de entradas son almacenadas en la ROM, pueden ser buscadas (“looked up” en inglés) en la tabla de verdad almacenada. Por esto, el método que emplea una ROM es también conocido como *look-up table* (LUT).

### **2.2.2 PLAs**

Un arreglo lógico programable (PLA) realiza la misma función que una ROM. Un PLA con  $n$  entradas y  $m$  salidas puede realizar  $m$  funciones de  $n$  variables. La organización interna del PLA difiere de la de la ROM, el decodificador se reemplaza por un arreglo de ANDs que realiza los términos producto seleccionados de las variables de entrada. El arreglo de ORs realiza la operación OR a los términos producto necesarios para formar las funciones de salida.

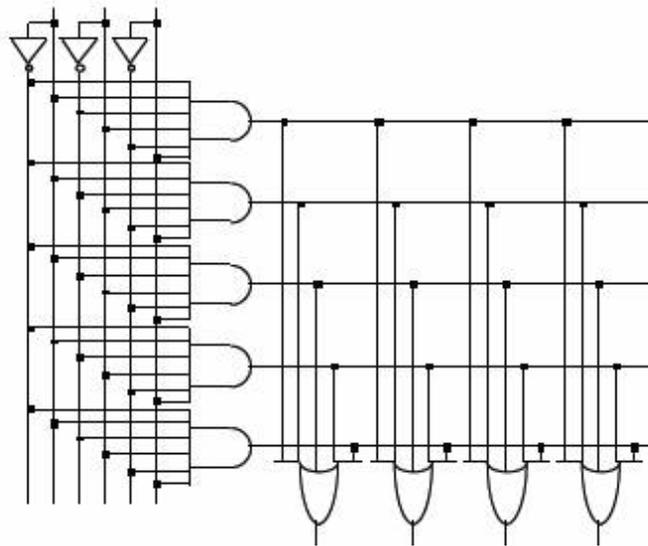


Figura 2.3 Diagrama de un PLA.

Para determinar la información que se graba en un PLA se realiza una tabla para PLA, que es diferente a una tabla de verdad para una ROM. En una tabla de verdad cada fila representa un mintermino, por lo tanto exactamente una fila se selecciona por cada combinación de valores de entrada, mientras que en cada fila de una tabla para PLA representa un término producto general. Por lo tanto cero, una o más filas se seleccionan por cada combinación de valores de entrada. Para determinar el valor de la función para cierta combinación de entrada, a los valores de la función en las filas seleccionadas de la tabla para PLA se les debe aplicar la operación OR.

término producto	entradas			salidas			
	A	B	C	F0	F1	F2	F3
A'B'	0	0	–	1	0	1	0
AC'	1	–	0	1	1	0	0
B	–	1	–	0	1	0	1
BC'	–	1	0	0	0	1	0
AC	1	–	1	0	0	0	1

Figura 2.4 Ejemplo de una tabla PLA.

### 2.2.3 PALs

El PAL (Programmable Array Logic) es un caso especial del PLA en el que el arreglo de ANDs es programable y el de ORs es fijo. Sus estructuras son iguales, pero el hecho de que únicamente el arreglo de ANDs sea programable hace más barato y fácil de programar el PAL en comparación con el PLA.

Cuando se diseña con PALs se deben simplificar las ecuaciones lógicas para que quepan en uno (o más) de los PALs existentes. Los términos AND no se pueden compartir entre dos o más compuertas OR, por lo tanto cada función puede ser simplificada por sí misma sin importar los otros términos. En cualquier PAL el número de términos AND que alimentan cada compuerta OR es fijo y limitado. Los PALs también pueden contener flip flops D con sus entradas provenientes del arreglo combinacional. Estos se llaman PALs secuenciales. Los PALs fueron desapareciendo con el desarrollo de otros dispositivos, como GALs, CPLDs y FPGAs.

### 2.2.4 Dispositivos lógicos programables/ Generic Array Logic

Conforme avanzaba la tecnología de circuitos integrados, una gran variedad de dispositivos lógicos programables aparecieron. Los PALs tradicionales no son reprogramables, sin embargo existen ahora PALs borrables y reprogramables con tecnología flash. A veces, a éstos se les llama PLDs.

El 22CEV10 es un PLD con tecnología CMOS borrable eléctricamente que puede ser usado para hacer tanto circuitos combinacionales como secuenciales.

Además de los arreglos AND y OR, la mayoría de los PLDs tienen algún tipo de macrobloque que contiene multiplexores y otros bloques programables adicionales. Estos PLDs se llaman de acuerdo a sus capacidades de entrada/salida. Por ejemplo, el 22CEV10 tiene 12 pines de entrada más 10 pines que se pueden programar como entrada o salida (22 en total). Contiene también 10 flip flops D y 10 compuertas OR. Cada compuerta OR dirige una macrocelda lógica de salida. Cada macrocelda contiene uno de los 10 *flip flops* D, los cuales comparten un reloj común, un *reset* asíncrono de entrada, y un *preset* síncrono de entrada.

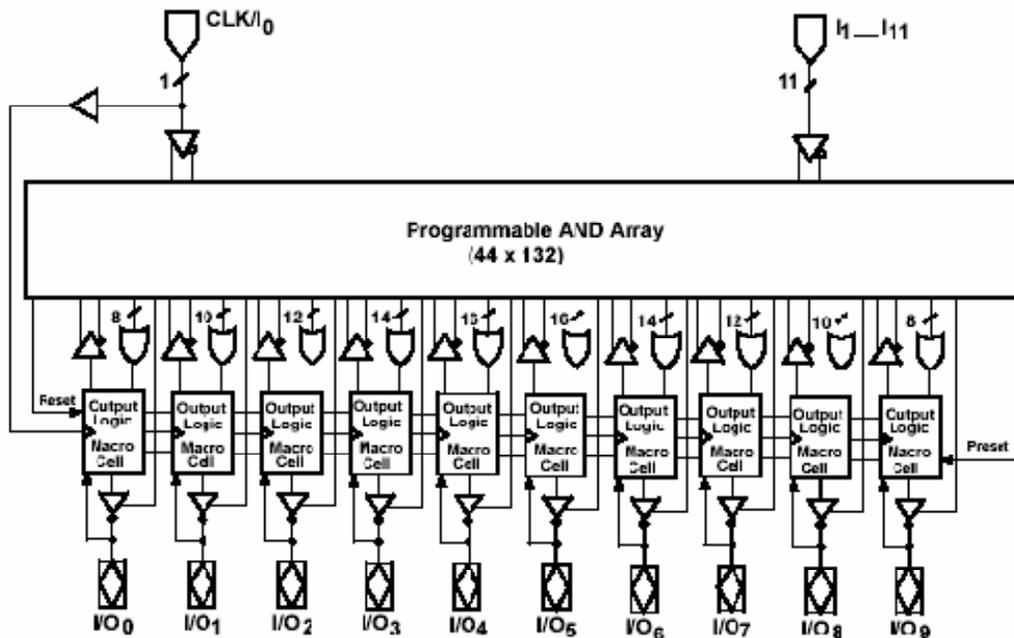


Figura 2.5 Esquema de una GAL 22V10.

La compañía Lattice Semiconductor creó dispositivos similares que tienen la capacidad de ser programados dentro del circuito (*in-circuit programming*) y lo llamó *Generic Array Logic* (GAL). Las GALs son perfectas para implementar pequeñas cantidades de lógica de interfaz. La mayoría de los PLDs, como PALCE22V10, PALCE20V8 entre otros, tienen sus equivalentes en GAL, llamados GAL22V10, GAL20V8, etc.

Existen programas CAD (*Computer Aided Design*) para PALs y PLDs. Estos programas aceptan ecuaciones lógicas, tablas de verdad, gráficas de estados y demás como entrada para generar automáticamente los patrones de bits que se requieren. Posteriormente, un programador puede descargar dichos patrones a los dispositivos para crear las conexiones necesarias. PALASM y ABEL son ejemplos de lenguajes que fueron populares como lenguaje de diseño para PALs y PLDs, aunque en estos días es posible hacer diseños para GALs en lenguajes como VHDL y Verilog.

### **2.3 CPLDs**

Los avances en tecnología han hecho posible la creación de circuitos integrados programables equivalentes a varios PLDs en el mismo chip. A estos circuitos integrados se les llaman dispositivos lógicos programables complejos (CPLDs por sus siglas en inglés).

Un CPLD es un circuito integrado que consiste en un número de bloques lógicos parecidos a un PAL, incluyendo además una matriz programable de interconexiones entre estos bloques.

Algunos CPLDs se basan en la arquitectura del PAL, en cuyo caso cada macrocelda contiene un flip flop y una compuerta OR, cuyas entradas están asociadas a un arreglo de compuertas AND fijo, mientras que los CPLDs que se

basan en PLAs cada salida de compuertas AND en un bloque se puede conectar a la entrada de cualquier compuerta OR en ese bloque.

Los más grandes fabricantes de CPLDs hoy en día son Xilinx, Altera, Lattice Semiconductor, Cypress y Atmel. Algunos de estos vendedores especifican sus productos en términos de cantidad de compuertas, mientras que otros lo hacen en términos de elementos lógicos.

Por ejemplo, Altera vende tres series de CPLDs, las cuales son MAX II, MAX 3000 Y MAX 7000. Cada una de éstas tiene especificaciones en general y los dispositivos dentro de cada serie se diferencian de acuerdo a sus capacidades lógicas y el número de pines de entrada/ salida.

Dentro de la serie MAX 7000 existen dispositivos que van de las 600 compuertas (32 macroceldas) hasta 5000 compuertas (256 macroceldas). Su arquitectura esta basada en módulos de arreglos lógicos, llamados Logia Array Blocks (LABs) que consisten en arreglos de 16 macroceldas. Los LABs se conectan por medio del *Programmable Interconnect Array* (PIA) alimentado por todas las entradas, pines de entrada/salida así como por las macroceldas.

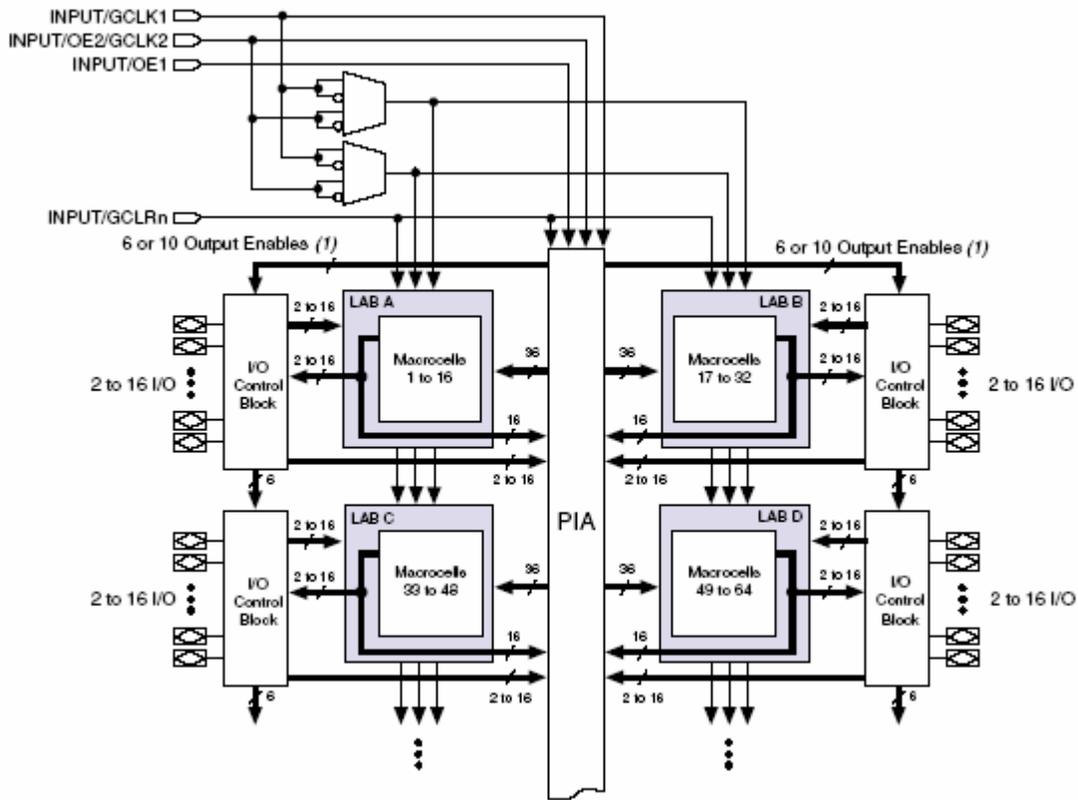


Figura 2.6 Diagrama de bloques del MAX 7000.

Las macroceldas de la serie MAX 7000 pueden ser configuradas individualmente para operar con lógica combinacional o secuencial. Tienen tres bloques funcionales: el arreglo lógico, la matriz de selección de términos producto y el registro programable.

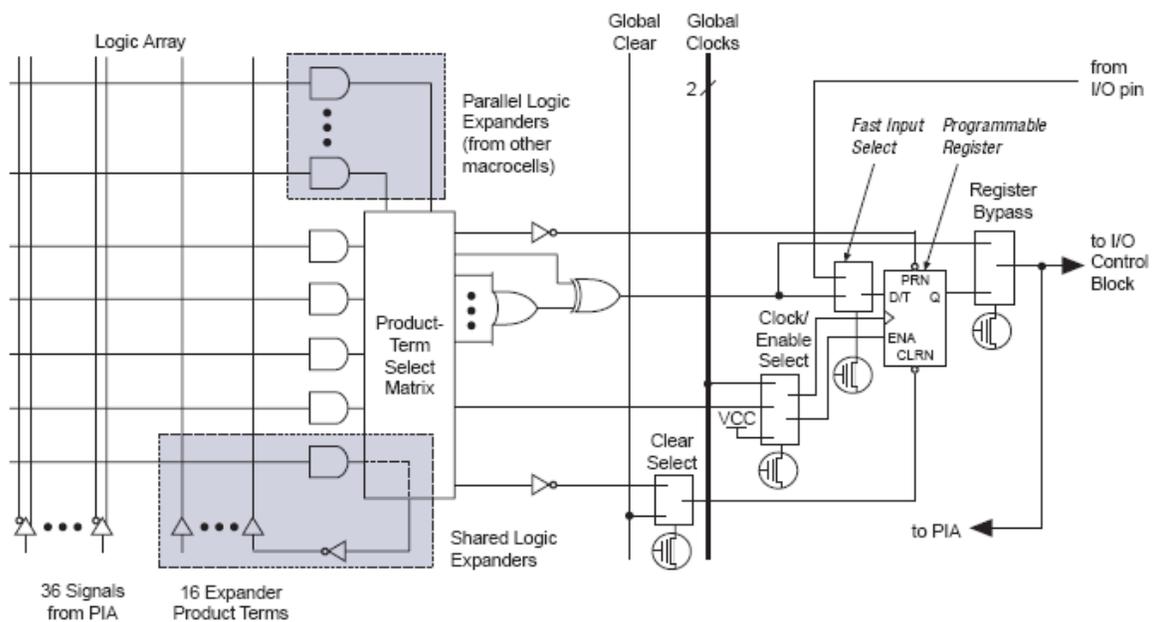


Figura 2.7 Diagrama de una macrocelda de la serie MAX 7000.

La lógica combinacional se implementa en el arreglo lógico que provee cinco términos producto por cada macrocelda. El registro de cada macrocelda se puede configurar para trabajar como *flip flop* D, T, JK o SR con un control de reloj programable. En el caso de tener únicamente lógica combinacional, el *flip flop* se puede ignorar.

## 2.4 FPGAs

Los FPGAs son circuitos integrados que contienen un arreglo de bloques lógicos idénticos con interconexiones programables, en los que el usuario puede programar tanto las funciones realizadas por cada bloque lógico como las conexiones entre bloques.

Los FPGAs tienen varias ventajas con respecto a MPGAs. Un arreglo de compuertas tradicional puede ser usado para implementar cualquier circuito, pero

sólo se puede programar en fábrica ya que se requiere hacer una máscara específica para un circuito en particular y el tiempo de diseño para un circuito integrado basado en arreglo de compuertas es de algunos meses. Por otro lado, los FPGAs son productos comerciales, el tiempo de manufactura se puede reducir de meses a algunas horas cambiando de MPGAs a FPGAs. De la misma forma, se vuelve más fácil y más barato corregir errores en los diseños. A volúmenes no tan altos, los FPGAs son más baratos que los MPGAs.

El interior de los FPGAs contiene típicamente tres elementos programables: los bloques lógicos, los bloques de entrada/ salida y las interconexiones. Se considera que los bloques de entrada/ salida se encuentran en la periferia del circuito integrado, éstos conectan las señales lógicas a los pines del chip. Los bloques lógicos se encuentran distribuidos dentro del FPGA y el espacio entre ellos se usa para mandar conexiones entre bloques.

La programabilidad de campo se logra por los elementos que pueden ser reconfigurables por el usuario. Los bloques lógicos se crean usando multiplexores, look-up tables y arreglos de compuertas AND-OR o NAND-NAND, y cualquiera de estas cosas puede ser programada (o configurada) por el usuario.

Lo que diferencia un FPGA de un CPLD es la interconexión flexible de propósito general. En un CPLD la interconexión es bastante restringida, mientras que en un FPGA es muy flexible, lo cual a veces puede resultar ser una desventaja ya que mandar una conexión de una parte del chip a otra muy alejada puede hacer el diseño más lento.

Los bloques lógicos entre FPGAs varían en los componentes básicos que emplean. Algunos FPGAs usan bloques basados en LUTs, mientras que otros usan multiplexores y compuertas lógicas. Existen también bloques lógicos que simplemente consistían en pares de transistores. En los primeros FPGAs de Altera, los bloques consistían en PLDs.

Los bloques lógicos también varían en el tamaño. Algunos FPGAs usan bloques básicos grandes, capaces de implementar varias funciones de cuatro o cinco variables, con algunos flip flops. En contraste, también existen FPGAs con bloques que sólo permiten una función de tres variables y un flip flop en cada bloque. Los distintos fabricantes usan nombres diferentes para sus bloques, por ejemplo, en Xilinx un bloque lógico programable se llama *Configurable Logic Block*; Altera los llama *Logic Elements* (LEs) y una colección de ocho o diez de ellos se llama *Logic Array Block* (LAB).

Un elemento importante en los FPGAs es la interconexión programable entre bloques lógicos. Existen diferentes tipos de conexiones en FPGAs comerciales. Algunos usan matrices de *switches*, en las que hay un *switch* en cada intersección de “cables”. Una matriz de *switches* soporta cualquier conexión entre cable y cable, pero resulta muy cara esta tecnología, además de que no todas las conexiones pueden existir al mismo tiempo.

Otros FPGAs usan conexiones especiales entre bloques lógicos adyacentes. Este tipo de conexiones son rápidas porque no necesitan pasar por una matriz de ruteo. En este tipo de FPGAs las interconexiones directas se dan hacia los cuatro bloques vecinos (arriba, abajo, izquierda y derecha), en otros casos se dan hacia los ocho vecinos, incluyendo así a los diagonales.

Los pines de un FPGA están conectados a bloques programables de entrada/salida que facilitan conectar las señales de los bloques lógicos al mundo externo.

Cada bloque de entrada salida tiene un número de opciones. El pin puede ser configurado para ser entrada o salida mediante un buffer triestado. El bloque contiene *flip flops* para guardar los valores de entrada o de salida. La señal de salida puede ser invertida si se desea, mediante una compuerta XOR.

Los FPGAs recientes tienen también bloques especializados. Dentro de estos bloques existen los de memoria RAM, que van de 16k a 10M bits los cuales pueden servir para almacenar datos necesarios en un proceso. El vendedor puede incluir también bloques de procesamiento digital de señales con *hardware* para realizar transformadas rápidas de Fourier, filtros FIR e IIR, entre otras cosas. Existen también bloques embebidos de procesadores dentro de los FPGAs modernos, como el MicroBlaze de Xilinx y el Nios de Altera.