

## UNIDAD 1

### Tecnología de Dispositivos Programables

#### 1.1 Estado del Arte en el Diseño Digital Moderno.

En la década de 1970, cuando los circuitos lógicos Transistor-Transistor (TTL), se introducen al mercado, se convierten en los elementos básicos para los diseñadores de sistemas digitales. Por muchos años, las estructuras de estos circuitos se mantuvieron, cambiando solamente su tecnología de proceso para hacerlos más versátiles y rápidos, como fueron los casos de *Low Power Schottky (LS)*, *Advanced Low Power Schottky (ALS)*, *Standart Schottky (S)*, *Advanced Standart Schottky (AS)*, *Fast (F)*, *HCMOS*, *Advanced CMOS Logic (ACL)*, *BiCMOS (BCT)* y *Advanced BiCMOS Technology (ABT)*.

A mediados de esta misma década, se enfocaron esfuerzos en desarrollar dispositivos que fueran capaces de reemplazar a esta tecnología para reducir espacio en las tarjetas de circuito impreso. La Memoria de Sólo Lectura Programable por el usuario una sola vez (*Programmable Read Only Memory, PROM*), desarrollada por *Harris y Monolithics Memories Inc.*, fue el primer dispositivos “programable” que prometía ser un elemento lógico universal y dispositivo de propósito general para tales aplicaciones. Tuvo aplicaciones como decodificador de direcciones de memoria y creció hasta tamaños de 32 palabras por 8 bits. *National Semiconductors* fue el pionero en el desarrollo de Arreglos Lógicos Programables (*Programmable Logic Arrays, PLA's*) en forma de un DIP grande de 24 pines o terminales (0.6 pulgadas de ancho) con 96 términos productos, 14 entradas, 8 salidas.

*Intersil* hizo el primer *Arreglo Lógico Programable en el Campo (Field Programmable Logic Array, FPLA)*, pero con la mitad de capacidad del dispositivo creado por *National Semiconductor*. *Signetics* incremento el empaque en número de pines a 28, el FPLA 82S10 de 16 entradas, 8 salidas, y 48 términos producto. *Monolithics Memories* fue la primer compañía que tomó ventaja de la tecnología bipolar de los fusibles de interconexión de las PROM's para crear algunas FPLA's pequeñas y veloces. También fue la primera en imponer como estándar industrial, los DIP's delgados de 20 pines (0.3 pulgadas).

La lógica programable fue desarrollada como una alternativa entre los siguientes métodos de diseño: Los dispositivos de *función fija* (o de *Lógica de Catálogo*), y Los circuitos integrados elaborados *ex profeso (CUSTOM)*. Cada uno de éstos tiene sus ventajas y desventajas que determinan su uso apropiado. La ventaja de utilizar estos dispositivos en el diseño digital radica en que todo el desarrollo se lleva a cabo en un solo ambiente de trabajo, esto es, el diseñador propone la función lógica a realizar y en base a métodos de descripción define los parámetros de su problema, esto se hace por medio de programación. Una vez acotado el problema, se optimiza su representación lógica y eléctricamente la misma, se selecciona el dispositivo que mejor se adapte a las condiciones de nuestro problema según criterios de velocidad, potencia, costo, etc., y finalmente se programa y verifica en el mismo lugar el o los dispositivos seleccionados. Esto ha sido posible no sólo por el avance tecnológico propiamente, sino también a la creación de herramientas de software que permiten una valiosa ayuda en la representación de los diseños. Estas ventajas se resumen en los siguientes puntos:

- Permite la minimización del número de componentes en un diseño, con esto se reducen los gastos de inventario, inspección y prueba, así como el número de fallas a nivel circuito impreso, incluyendo el área utilizada por el mismo.

- Reemplazo de lógica catálogo si un producto está siendo rediseñado para ahorrar espacio. Una medida de la eficiencia de un PLD se expresa mediante el número de dispositivos de función fija que pueden reemplazarse.
- Reducción en el tiempo de diseño, lo cual a su vez reduce el tiempo y los costos de desarrollo, no solo de nuevos productos, sino también, de aquellos que requieren modificaciones (rediseño). Esto último, se debe a que los cambios en el diseño son realizados mediante la simple programación de un nuevo dispositivo.
- Uso de una gran variedad de herramientas de Ingeniería Asistida por Computadora (Computer-Aided Engineering, CAE) disponibles actualmente en el mercado.
- Los PLDs son excelentes para realizar prototipos de diseño de circuitos integrados elaborados para tal efecto (diseños “FULL CUSTOM” y “STANDARD CELLS”) debido a:

a) Muchas de las herramientas de CAE utilizadas en el diseño de circuitos integrados elaborados lo son también para los PLDs.

b) Los PLD's son programables, de aquí que los cambios en el diseño pueden ser realizados inmediatamente, lo cual es tan opuesto a ordenar la fabricación de un nuevo conjunto de máscaras (circuitos integrados elaborados).

## 1.2 Escalas de Integración de los Circuitos Electrónicos Integrados.

La creciente popularidad de los *circuitos integrados (CI)* se debe, en parte, a la disponibilidad de éstos a bajos precios. Los fabricantes han desarrollado muchas familias de circuitos digitales integrados –grupos que pueden ser utilizados juntos para construir un sistema digital-. Los CI de una familia se dice que son compatibles, y pueden conectarse fácilmente entre sí.

Los CI digitales pueden dividirse en bipolares y unipolares. Los CI digitales bipolares se fabrican a partir de elementos comparables a los transistores bipolares, diodos y resistores discretos. La familia TTL es la más popular de los CI que usan tecnología bipolar. Los CI integrados unipolares se fabrican a partir de elementos comparables a los transistores de efecto de campo de puerta aislada (IGFET). La familia CMOS (metal-óxido semiconductor complementario) es un grupo de CI ampliamente usado basado en la tecnología metal-óxido semiconductor (MOS).

Los fabricantes, a veces, agrupan los CI por su complejidad. La complejidad de éstos se define como sigue:

### 1. SSI (pequeña escala de integración):

Número de compuertas: menos de 12

Dispositivos digitales típicos: compuertas y flip-flops

### 2. MSI (mediana escala de integración):

Número de compuertas: de 12 a 99

Dispositivos digitales típicos: sumadores, contadores, decodificadores, codificadores, multiplexores, demultiplexores y registros

### 3. LSI (alta escala de integración):

Número de compuertas: de 100 a 9,999

Dispositivos digitales típicos: relojes, chips pequeños de memoria, calculadoras

### 4. VLSI (muy alta escala de integración):

Número de compuertas: de 10,000 a 99,999  
Dispositivos digitales típicos: microprocesadores, memorias, calculadoras avanzadas.

#### **5. ULSI (ultra alta escala de integración):**

Número de compuertas: sobre 100,000  
Dispositivos digitales típicos: microprocesadores avanzados.

El diseñador de circuitos digitales dispone de muchas familias CI digitales, algunas de ellas se mencionan a continuación:

##### **1. Familias bipolares:**

RTL	lógica de resistencia-transistor
DTL	lógica de diodo-transistor
TTL	lógica de transistor-transistor
DTL	lógica de emisor acoplados
HTL	lógica de alto umbral
IIL	lógica de inyección integrada

##### **2. Familias CMOS:**

PMOS	metal-óxido canal P
NMOS	metal-óxido canal N
CMOS	metal-óxido semiconductor complementario.

Las tecnologías TTL y CMOS son utilizadas comúnmente para fabricar circuitos integrados SSI y MSI. Los dispositivos MOS (PMOS, NMOS y CMOS) dominan en la fabricación de dispositivos LSI y VLSI. NMOS es especialmente popular en los microprocesadores y en las memorias. CMOS es popular en aplicaciones de muy baja potencia tales como calculadoras, relojes de pulsera y computadoras alimentadas por baterías.

### **1.3 Generalidades de las Tecnologías Programables.**

Los Dispositivos de Lógica Programable incluyen a los circuitos de integración SSI y MSI que pueden ser configurados para emular más circuitos lógicos discretos, igual que las interconexiones complejas de estos circuitos. Debido a estas características, muchos profetas predijeron alguna vez que la época de la lógica discreta había terminado debido a que todos los diseñadores la reemplazarían agresivamente con lógica programable. Tal decisión puede reducir inventarios y proporcionar un medio ambiente más flexible de diseño.

Aunque tal razonamiento era adecuado, el mercado multimillonario del dólar aún mantuvo el interés por los circuitos de lógica discreta demostrando que esta profecía era falsa. Sin embargo, la falla de esta predicción pudo ser debido a la ignorancia del mercado del diseño digital, al igual que por otros factores.

El aspecto más significativo de este concepto, del cual toma ventaja es que estos circuitos pueden ser configurados para reemplazar a más de un circuito de lógica discreta. Por ejemplo, si asumimos que un diseño digital de un circuito de lógica combinatoria genera 10 funciones lógicas, cada una minimizada con variables de 8 minitérminos de un conjunto de 12 entradas diferentes, tal diseño podría ser implementado con compuertas de lógica discreta. Un mínimo de 10 compuertas OR de ocho entradas tendrían que asumirse para los ocho minitérminos, cada uno creada por una compuerta AND de 12 entradas. Esto requeriría en total 96 compuertas AND de 12 entradas. Este problema puede resolverse fácilmente usando un solo dispositivo de lógica programable que esta comercialmente disponible. Además, los dispositivos disponibles contienen flip flops que son parte del circuito secuencial común.

Al desarrollar diseños de máquinas de estado, descubriremos que los dispositivos de lógica programable son particularmente más adaptables para estas aplicaciones. Una máquina de estados compleja moderadamente, puede requerir mucha lógica discreta por lo que construirla puede ser poco práctico. Sin embargo, el mismo diseño puede requerir solo uno o dos dispositivos de lógica programable.

## UNIDAD 2

### Arquitecturas Básicas de los PLDs

#### 2.1 Arquitectura PLD Clásica.

El gran desarrollo tecnológico que se ha experimentado en las últimas décadas sobre todo en la microelectrónica, ha permitido integrar millones de elementos activos en un cristal de silicio, mediante la *tecnología planar*. Lo anterior ha provocado que viejas arquitecturas de dispositivos lógicos resurjan con más fuerza y complejidad que antes, tal es el caso de los *Arreglos Lógicos Programables (PLAs)* y en general los *Dispositivos Lógicos Programables (PLDs)*. Estas arquitecturas, datan de 1974; en aquel entonces, las limitantes eran la complejidad, las rígidas arquitecturas, el escaso avance de la tecnología y los altos costos, los hizo poco competitivos contra los dispositivos de línea estándares, por ejemplo la familia TTL.

Sin embargo, la necesidad de diseñar sistemas digitales compactos que respondan a los requerimientos de los diseñadores, con tiempos de diseño cortos y de costo accesible, utilizando las funciones de los dispositivos lógicos discretos de baja escala de integración (*SSI, Short Scale Integration*) y los de mediana escala de integración (*MSI, Medium Scale Integration*), tales como compuertas TTL y CMOS, dio lugar a los PLDs que son dispositivos que pueden ser programados por el usuario, personalizando de esta manera sus diseños. Aunque el término anterior es aplicable a cualquier dispositivo lógico programable, actualmente consiste en un arreglo de compuertas lógicas, elementos de memoria (Flip-Flop's), y registros los cuales pueden ser interconectados de múltiples maneras para conformar circuitos lógicos de alta complejidad, para un número ilimitado de diseños lógicos. Los tipos de PLDs básicos incluyen las *Memorias de Sólo Lectura (PROMs, Programmable Read Only Memorys)*, que son la forma más vieja de PLDs, pero no fueron diseñadas para aplicaciones lógicas, los *Arreglos Lógicos Programables (PLAs, Programmable Logic Arrays)*, la *Lógica de Arreglos Programables (PALs, Programmable Array Logic)*, la *Lógica Programable y Borrable Eléctricamente (PEELs, Programmable Electrically Erasable Logic)*, los *Arreglos Lógicos Genéricos (GALs, Generic Array Logic)*, y los *PLDs Complejos (CPLD, Complex PLD)*.

##### 2.1.1 Memoria de sólo lectura programable (PROM)

La *memoria de sólo lectura programable (PROM)*, desarrollada por *Harris y Monolithic Memories Inc.*, fue el primer dispositivo “programable” que prometía ser un elemento lógico universal y de propósito general para tales aplicaciones. Dos versiones básicas de las PROM's fueron desarrolladas: las que son programadas solo por el fabricante, y las que pueden ser programadas por el usuario final. El primer tipo es llamado programable por máscara y el segundo programable en campo. En el contexto de

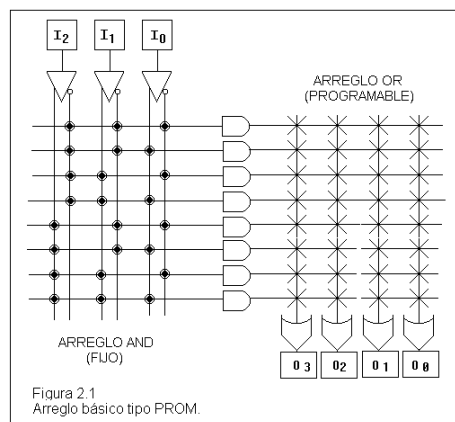
implementación de circuitos lógicos, el dispositivo de mayor velocidad es el programable por máscara, porque las conexiones dentro de él se alambran durante su fabricación. En contraste, las conexiones programables en campo siempre involucran algunos interruptores (switches) programables (como un fusible), lo que es inherentemente más lento que una conexión alambrada. Sin embargo, un dispositivo programable en campo ofrece ventajas que muchas veces sobrepasan sus deficiencias de velocidad de ejecución:

- Estos dispositivos son menos caros en volúmenes bajos que los programables por máscara, porque son partes estándar de fabricación. Las instalaciones para la fabricación de Circuitos Integrados (CI) deben equiparse para iniciar la producción de dispositivos programados por máscara, lo cual genera grandes gastos.
- Pueden ser programados inmediatamente, en minutos, mientras que los programados por máscara deben ser fabricados, lo cual toma semanas o meses.

Dos variantes de la PROM son: la *Memoria de Sólo Lectura y Borrable por luz ultravioleta (EPROM)* y la *Memoria de Solo Lectura y Borrable Eléctricamente (EEPROM)*, las que ofrecen una ventaja adicional, ya que pueden ser borradas y reprogramadas varias veces. En algunas veces, y en particular durante las primeras etapas del diseño de un circuito, la reprogramación es una atractiva característica.

Este tipo de memoria es la forma más antigua de los PLD's, pero inicialmente no fueron diseñadas para aplicaciones lógicas, sino para almacenar datos, tales como programas de inicialización (bootstrap), e instrucciones de microcódigo. También fueron usadas extensamente para funciones de decodificación y máquinas de estado simples ( cuando se usan con registros de almacenamiento externo). La estructura de las PROM puede emplearse para circuitos lógicos de propósito general, puesto que incorpora una suma de productos en un arreglo lógico. El arreglo básico se muestra en la **figura 2.1**, donde están indicadas todas las localidades de los fusibles programables con **Xs**. Para programar este arreglo, uno decide que puntos de interconexión (cada intersección de renglón-columna) son preservados, y por consiguiente deshabilitadas las conexiones restantes.

La memoria ROM es un circuito de *baja escala de integración (LSI, Low Scale Integration)*, la cual consiste de un arreglo de dispositivos semiconductores (diodos, transistores bipolares o de efecto de campo), los cuales son interconectados para almacenar un arreglo de datos binarios. Sólo datos binarios son almacenados en este tipo de memoria, la cual puede ser leída cuando se desee pero no modificada en condiciones de operación normal. El contenido de está memoria es usualmente especificada por una tabla de verdad.

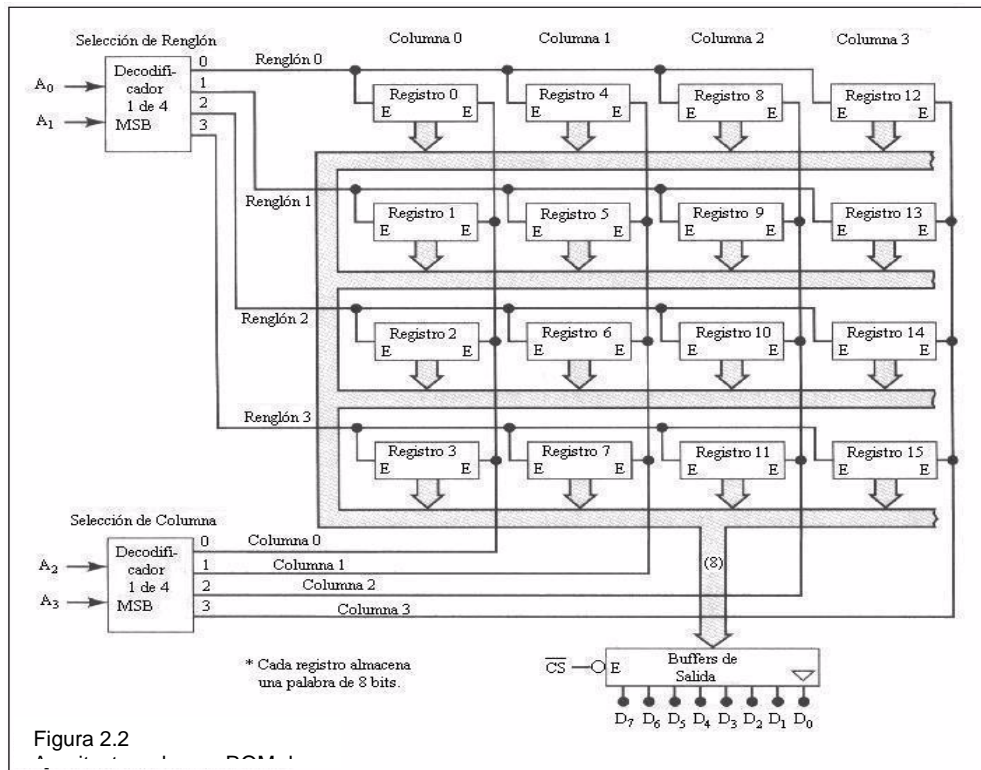


La arquitectura interna (estructura) de un CI ROM es muy complejo, pero no necesitamos familiarizarnos con todos esos detalles. Un diagrama simplificado de está arquitectura se muestra en la

**figura 2.2** para una ROM de 16 X 8 (16 palabras de 8 bits por palabra). En esta figura tenemos cuatro partes básicas: arreglo de registros, decodificador de renglón, decodificador de columna, y buffers de salida.

Los arreglos de registros almacenan los datos que han sido programados en la ROM. Cada registro contiene un número de celdas de memoria igual al tamaño de palabra. En este caso, cada registro almacena una palabra de ocho bits. Los registros son ordenados en una matriz cuadrada, que es lo común en la mayoría de los dispositivos de memoria. Se puede especificar la posición de cada registro, dando el renglón y la columna. Por ejemplo, el registro 0 esta dado por el renglón 0 columna 0, y el registro 9 por el renglón 1, columna 2. La salida de los ocho datos de salida de cada registro son conectados a un bus de datos interno que va a través de todo el circuito. Cada registro tiene dos entradas de habilitación (E, Enable); ambas deben de estar en un nivel ALTO para que los datos del registro sean puestos en el bus.

*Decodificadores de direcciones.* El código de direcciones  $A_3A_2A_1A_0$  aplicado, determina cuál registro en el arreglo será habilitado para que su palabra de ocho bits sea puesta en el bus. Los bits de direcciones  $A_1A_0$  son puestos a un decodificador 1 de 4, el cual activa una línea del renglón seleccionado, y los bits de direcciones  $A_3A_2$  son puestos a un segundo decodificador 1 de 4, el cual activa una línea de la columna seleccionada. Sólo un registro será seleccionado por las entradas de direcciones de renglón y columna.



*Buffers de salida.* Los registros que son habilitados por las entradas de direcciones serán puestos sus datos sobre el bus. Estos datos son puestos en los buffers de salida, los cuales pasan el dato a la salida externa siempre y cuando la señal /CS este a un nivel BAJO. Si /CS es ALTO, los buffers de salida estarán en un estado de alta impedancia, y  $D_7$  a  $D_0$  estarán flotando.

La arquitectura mostrada en la **figura 2.2** es similar a cualquier CI ROM. Dependiendo del número de palabras de datos almacenados, los registros en algunas ROMs no serán arreglos cuadrados. Por ejemplo, la memoria ROM MOS de Intel 2764 que almacena 8192 palabras de 8 bits, sus registros

están ordenados en arreglos de 256 renglones X 32 registros. Las capacidades de los arreglos de las ROMs van desde 256 X 4 hasta 8M X 8.

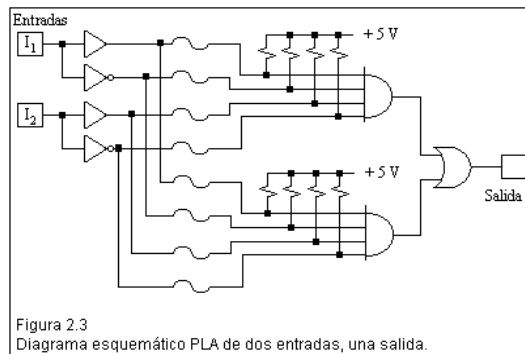
### 2.1.2 Arreglos Lógicos Programables (PLAs).

La arquitectura de un PLD afecta las aplicaciones lógicas para las cuales el dispositivo puede ser usado. Esto es, al hablar de la arquitectura hacemos referencia a los atributos que afectan la construcción lógica del dispositivo (configuración de terminales, el arreglo y tamaño de los arreglos programables, y la configuración de la lógica de interface de entrada/salida).

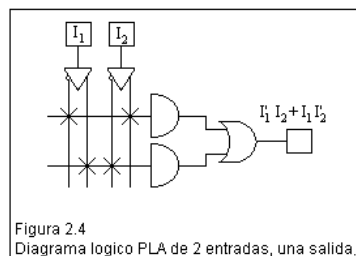
Un PLD normalmente esta compuesto de un número específico de líneas de entrada conectadas a través de un arreglo fijo o programable para un conjunto de compuertas AND, las cuales están conectadas a un arreglo fijo o programable de compuertas OR. Las compuertas OR proporcionan las señales de salida del arreglo lógico.

Un PLD simplificado esta compuesto de dos entradas y una salida como se muestra en la **figura 2.3**. Como se puede observar, para proporcionar todas las posibles combinaciones de entrada, cada una es enrutada al arreglo en ambas formas; verdadera y complementada. Estas entradas están conectadas a compuertas AND vía puntos de interconexión programables. Es un producto de arreglos OR, los cuales forman una suma de productos de arreglos lógicos. Como un PLD tiene más entradas y salidas que el arreglo simple mostrado anteriormente, se usa una notación especial llamado diagrama lógico, este se emplea para describir gráficamente las estructuras complejas PLA asociadas con estos dispositivos. Cuando el arreglo es programado para implementar una función lógica en particular, la interconexión deseada puede ser indicada sobre el diagrama con **X's**. La **figura 2.4**, muestra el arreglo lógico de dos entradas y una salida, trazado en forma de diagrama lógico y programado con una simple función lógica.

En algunos dispositivos, la estructura AND/OR es remplazada por una estructura NAND/NAND ó NOR/NOR, pero el resultado es el mismo: con un arreglo bastante grande, cualquier función puede ser implementada.



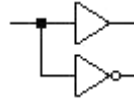
La programación de los dispositivos es realizada por habilitación o deshabilitación de las interconexiones en el arreglo programable del dispositivo. Las conexiones actuales pueden ser a condición de una variedad de caminos dependiendo de la tecnología del dispositivo.



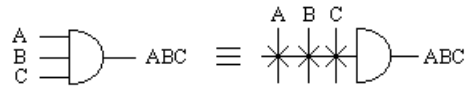
La **figura 2.4** representa un segmento de un PAL programado. El símbolo



representa un buffer de entrada, el cual es equivalente lógico a



Un buffer es usado para cada entrada del PAL y maneja cualquier entrada a la compuerta AND. Cuando el arreglo es programado, los fusibles que unen son seleccionados para fundirse de acuerdo a la conexión deseada a las entradas de la compuerta AND. Por ejemplo, conectando las entradas a la compuerta AND en una PAL, quedaría como sigue



La mayoría de las funciones lógicas de  $n$  variables pueden implementarse con menos de  $2^n$  términos producto. Para mayor eficiencia del mapa de las funciones lógicas con un gran número de entradas, se desarrollaron los dispositivos PLA y PAL. La estructura PLA virtualmente es la base para todos los PLDs en la actualidad. Esta estructura proporciona una gran flexibilidad ya que los términos producto se localizan en las compuertas OR y sus salidas asociadas. La estructura PLA, como se muestra en la **figura 2.5**, caracteriza a un arreglo programable AND, y un arreglo programable OR. Para una PLA con  $n$  entradas, cualquier variable de entrada (o su complemento) puede ser una entrada a cualquier entrada AND. Por lo tanto, cualquier compuerta AND en el arreglo puede configurarse para implementar cualquiera de  $3^n$  posibles términos productos (por que cada entrada a una compuerta tiene tres posibles valores –verdadero, complemento y, no conectado). El diseño del PLA permite que cualquier término producto en el arreglo esté conectado a la compuerta OR de cualquier salida. Esta característica algunas veces es referida como términos producto compartido.

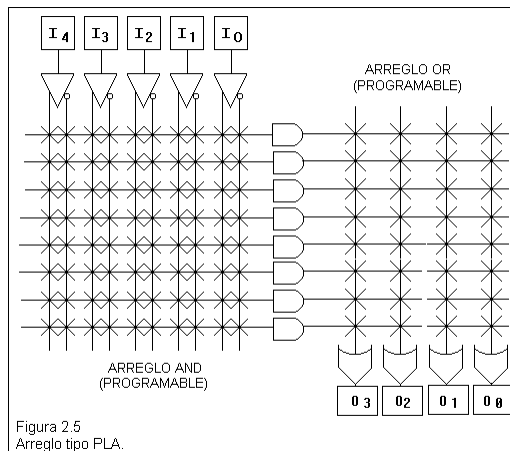


Figura 2.5  
Arreglo tipo PLA.

El PLA proporciona la mayor flexibilidad para implementar circuitos lógicos, particularmente para grandes diseños en los cuales los elementos de lógica común pueden compartirse entre las salidas del circuito. Los dispositivos tipo PLA generalmente son más lentos en operación que las PROMs y PALs, ya que tiene dos arreglos programables a través de los cuales las señales tienen que propagarse.



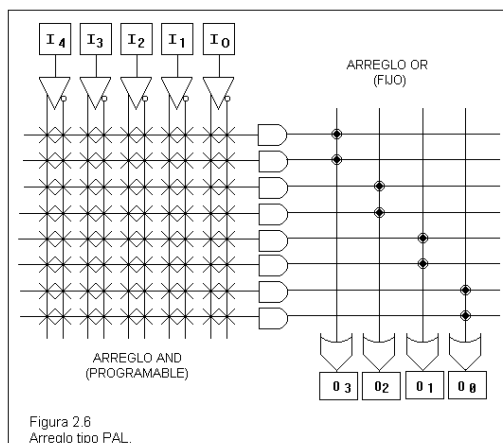
### 2.1.3 Lógica de Arreglos Programables (PALs) y Lógica de Arreglos Genéricos (GALs).

Otro tipo de dispositivo programable típico diseñado específicamente para implementar circuitos lógicos es el PAL, que es similar al PLA, pero consiste de un plano programable AND seguido por un plano fijo OR. Las salidas de las compuertas OR son sumas de productos que pueden ser opcionalmente registradas por un flip-flop en la mayoría de los dispositivos. Los PALs también ofrecen la ventaja de ser programables en campo. Cada salida en un dispositivo tipo PAL estándar tiene una compuerta OR, que es única a esa salida. Este no proporciona términos producto compartidos. Como un número limitado de términos producto se abastecen para cada salida del PAL, las técnicas de minimización lógica son importantes para implementar circuitos lógicos en este tipo de dispositivo.

El primer circuito PAL fue el 16L8, de 20 terminales (pines), con soporte para 8 salidas y hasta 16 entradas con tecnología Bipolar. Con este tipo de dispositivos pueden simularse arreglos del tipo de Productos Lógicos, o en su caso realizar decodificación de direcciones. Sin embargo, la mayor aportación de los dispositivos PAL fue generar aplicaciones específicas en muy corto tiempo. La operación de estos dispositivos inicio manejando velocidades de 4.7 Mhz para la IBM PC hasta 33 Mhz, para posteriormente alcanzar los 50 Mhz; hoy se pueden encontrar dispositivos que operan con un retraso de propagación de la señal del orden de 5ns.

El PLA es una versión más flexible en términos de arquitectura que el PAL, ya que el número de términos no es fijo, es decir, por los arreglos AND y OR que son programables. Esto hace que las PLAs sean mejor utilizadas para aplicaciones de máquinas de estado complejas que para aplicaciones de decodificación como las PALs.

Con su estructura simple de dos niveles, como se muestra en la **figura 2.6**, este tipo de PLDs descritos anteriormente permite la implementación de circuitos lógicos de alta velocidad. Sin embargo, su estructura simple también es su principal desventaja, ya que solo se pueden implementar pequeños circuitos lógicos representados con un modesto número de términos producto, porque su estructura de interconexión esta fija en cuanto a que no se puede configurar la salida si se requiere que sea activo alto, activo bajo, con o sin registro.



Los dispositivos GALs, surgen como respuesta a la necesidad de incrementar las interconexiones internas de los dispositivos, teniendo sus salidas conectadas a macroceldas, las cuales pueden ser configuradas a sistemas combinatorios o de lógica con registros. Un dispositivo de amplio uso es el 16V8, el cual reemplaza a 24 tipos de PALs. Las GALs están construidas con arreglos EEPROM (tecnología CMOS), por lo cual pueden ser programadas varias veces, presentando consumos de potencia menores en comparación a los de la tecnología de las PALs (BIPOLAR, las cuales son

programables una sola vez), y con velocidades comparables. El problema con las GALs es la velocidad de disparo, que puede producir grandes cantidades de ruido en el circuito.

En el GAL 16V8, ciertas localidades en el arreglo de memoria son diseñadas para controlar las conexiones programables a la matriz de los términos de entrada, cada bit en esta matriz representa una conexión programable entre un renglón y una columna (ver archivo .JED, más adelante). Afortunadamente, no es necesario buscar dentro de las direcciones de cada localidad de bit en la matriz. El software de programación cuida de estos detalles en una manera amigable. El diagrama lógico completo del GAL 16V8 se muestra en la **figura 2.7**. Este dispositivo tiene ocho terminales de entrada dedicadas (terminales 2-9), dos con funciones especiales (1 y 11), y ocho (12-19) que pueden ser usadas como entradas o salidas.

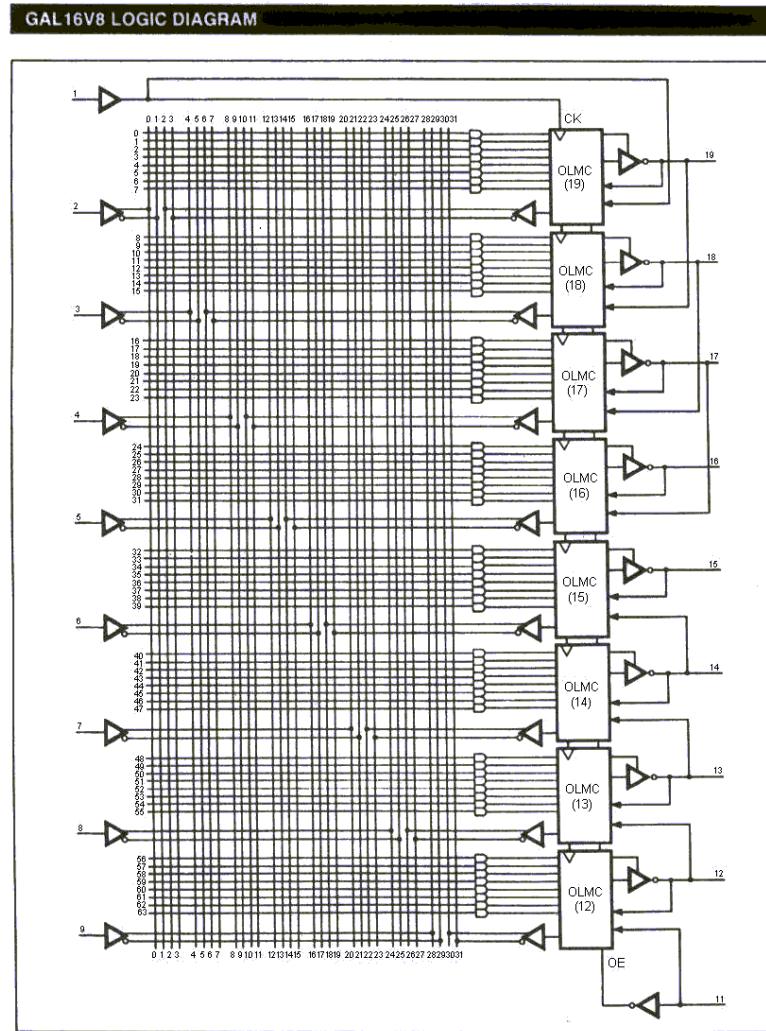
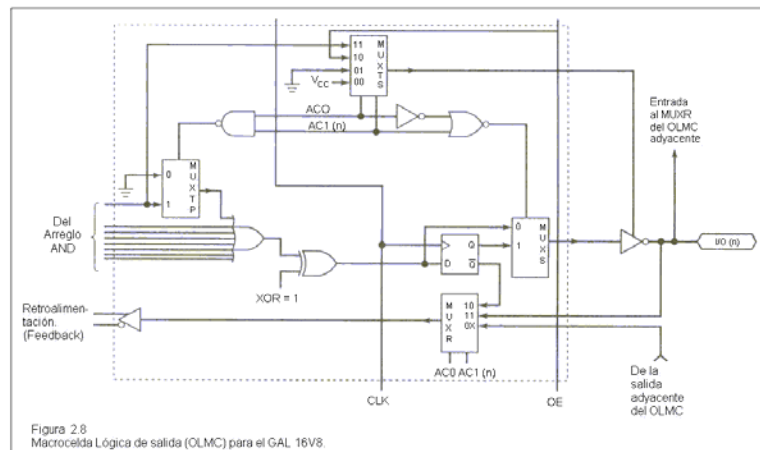


Figura 2.7.- Diagrama Lógico GAL 16V8.

La mayoría de los componentes de las GALs son; la matriz de los términos de entrada, las compuertas AND (las cuales generan los términos producto de entrada), y las macroceldas lógicas de salida (OLMCs, Output Logic Macro Cells). Hay que notar que las 8 terminales de entrada son conectados directamente a una columna de la matriz de términos de entrada, y así cada complemento de entrada es conectado a está. Estas terminales siempre serán especificadas como entradas cuando se programa el dispositivo. Un nivel lógico y su complemento son siempre retroalimentados de cada macro a una columna de la matriz de entrada. La fuente de estos niveles lógicos es determinada por la

configuración de cada macro. Con esto, da un total de 32 variables de entrada (columnas en la matriz de entrada) que pueden ser programadas como conexiones a las 64 compuertas AND de entradas múltiples. En la matriz de entrada, cualquier columna puede ser conectada a un renglón de entrada durante el proceso de programación.

La flexibilidad del GAL 16V8 consiste en su *macrocelda lógica de salida* programable (ver **figura 2.8**). Ocho productos diferentes (salidas de las compuertas AND) son aplicadas como entradas a cada una de las ocho macroceldas. Dentro de cada macro los productos son enviados a una compuerta OR al mismo tiempo para generar la suma de productos (SOP, sum of products). Recordando, cualquier función puede ser expresada como una forma de SOP. Dentro de la macro, la salida del SOP puede ser enrutado a la terminal de salida para implementar un circuito combinacional, o como entrada a un flip-flop tipo D para implementar un circuito con salida registrada.



## 2.2 Métodos de Programación de los PLDs.

Obviamente, existe un mejor método para especificar una configuración para un PLD que escribir **X's** y **O's** sobre un diagrama de matriz. Para entender y apreciar las reglas de software de desarrollo, vamos a examinar algunos detalles en el proceso de programación. Varios recursos son necesarios para diseñar y construir circuitos usando PLDs:

- Una Computadora Personal (PC)
- Software de desarrollo de PLDs
- Accesorios de programación
- Software para manejar el equipo de programación
- Dispositivos Lógicos Programables.

La **figura 2.9**, muestra un equipo de programación típico. El diseño inicial es cargado en la PC en uno de varios caminos posibles, el cual será discutido en una forma sencilla. En la PC corre un software de desarrollo, traslada el diseño a un archivo llamado *Diagrama de Fusibles*. Este archivo es como un mapa que muestra los fusibles que serán fundidos y los que permanecerán intactos en un dispositivo programable. Este diagrama es entonces trasladado a un archivo de salida con formato apropiado para enviarlo al equipo de programación.

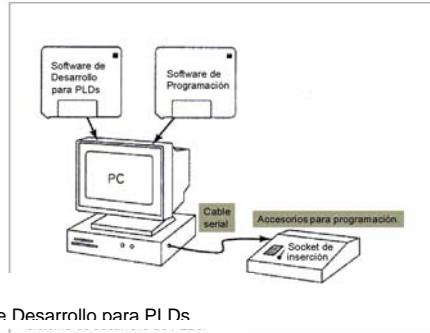


Figura 2.9  
Sistema de Desarrollo para PLDs

El software de programación que se comunica con el equipo de programación es entonces invocado en la PC. Esto permite que el usuario informe al programador el tipo de dispositivo que será programado. El archivo de salida es entonces enviado sobre un cable al equipo programador. Este proceso es llamado “transferir el archivo al programador” (*downloading*). Finalmente, el PLD es insertado en la base (socket) del programador, y el comando es enviado desde la PC para programarlo. Normalmente los programadores tienen una base especial para permitir que se inserte el chip, y entonces se cierran los contactos sobre las terminales. Esto es llamado un socket de cero esfuerzo de inserción (ZIF, Zero Insertion Force). Los equipos de programación universal, pueden programar cualquier tipo de dispositivos, incluyendo PROMs, EPROMs, microcomputadoras en un solo chip, y una gran cantidad de PLDs que están disponibles de numerosos fabricantes.

Afortunadamente, como las piezas programables empezaron a proliferar, los fabricantes tuvieron la necesidad de estandarizar la asignación de terminales, y los métodos de programación. Como un resultado, fue formado el *Consejo Mixto de Ingeniería de Dispositivos Electrónicos (JEDEC, Joint Electronic Device Engineering Council)*. Uno de los resultados fue el estándar 3 JEDEC, que es un formato estándar para transferir datos de programación para PLDs, independiente del fabricante o software de programación. La asignación de terminales para varios empaques de circuitos integrados (CI), fue también estandarizado, haciendo programadores universales menos complicados. Consecuentemente, los equipos de programación son capaces de programar numerosos tipos de PLDs. El software que permite diseñar específicamente una configuración para un PLD, simplemente necesita producir un archivo de salida que conforme el estándar JEDEC, este archivo puede ser cargado dentro de cualquier programador de PLD compatible con JEDEC, y es capaz de programar el tipo deseado de PLD.

## Bibliografía

1. - Mano, M. Morris y Kime, Charles. “*Fundamentos de Diseño Lógico y Computadoras*”, Ed. Prentice - Hall, 1999.
2. - Tokheim, Roger. “*Principios Digitales*”, Ed. McGraw Hill, 1995.
- 3.-. Horenstein, Mark. “*Microelectrónica: Circuitos y Dispositivos*”, Ed. Prentice - Hall, 1997.
4. - Bolton, Martin. “*Digital System Design with Programmable Logic*”, Ed. Addison - Wesley, 1997.
5. - Pallerin, Davis y Holley, Michael. “*Practical Design Using Programmable Logic*”, Ed. Prentice - Hall, 1997.
6. - Digiacommo, Joshep. “*Designing with High Performance ASIC's*”, Ed. Prentice - Hall, 1998.
7. - Smith, Michael J. S. “*Application – Specific Integrated Circuits*”, Ed. Addison - Wesley, 1999.