

# 1 Generalidades

El nombre de Dispositivos lógicos Programables ó PLD (Programmable Logic Device) es una acepción genérica establecida para cualquier sistema digital cuyo funcionamiento está determinado por el usuario, después que dicho dispositivo fuera fabricado. Al referirnos a usuario, generalmente nos dirigimos a un profesional del diseño lógico, aunque pudiera ser realmente el cliente final. Al hablar aquí de circuitos digitales en forma general, nos referimos a todos los sistemas digitales incluyendo por supuesto a los combinacionales y secuenciales.

Actualmente se los utiliza para realizar todo tipo de circuitos digitales, desde los más sencillos a los más complejos; que en el pasado se realizaba con lógica cableada a través del uso de una gran cantidad de elementos lógicos convencionales SSI ó MSI; utilizados en forma discreta. Pero la diferencia fundamental es que estos dispositivos están armónicamente estructurados, tanto en su hardware como en las herramientas de software utilizadas para el diseño lógico.

Debido a los avances de la tecnología en la realización de nuevas estructuras y el incremento de la integración, logrando cada vez circuitos en menor tamaño, ó mayor cantidad de elementos lógicos en el mismo, es que cualquier diseño digital, que hasta hace poco tiempo se realizaba en forma discreta, se pueda configurar ó realizar con estos circuitos lógicos programables. Podemos decir entonces, en forma generalizada que estos dispositivos permiten programar todo tipo de componente de la lógica booleana, desde las compuertas más elementales hasta los secuenciales más complejos, pasando por decodificadores, multiplexores, contadores, registros, procesadores digitales, sistemas de transmisión de datos, etc.

Estos circuitos PLD conforman una estructura básica compacta, perfectamente estructurada y por lo general realizada en un solo circuito integrado (aunque a veces puede ser más de uno) fabricados en alta (LSI), muy alta (VLSI) ó ultra alta (ULSI) escala de integración. Poseen diferentes características según el fabricante, en cuanto a su capacidad de programación, tecnología, forma exterior, etc. Al igual que las memorias programables, están aquellos que permiten una única programación, los que pueden ser borrados con luz ultravioleta y aquellos borrables y programables eléctricamente.

Los sistemas secuenciales síncronos de control microprogramados, también llamados microcontroladores, constituyen una especie de dispositivos lógicos programables, pero como en general poseen una arquitectura perfectamente definida y lo único que se programa es una memoria PROM no se incluyen en el estudio de este tipo de circuitos. Algunos autores suelen incluirlos, junto a los PLD propiamente dichos dentro del estudio y clasificación de las memorias PROM.

Debido a la gran diversidad de oferta de estos dispositivos programables por los diferentes fabricantes, y que con diferentes nombres se suelen presentar elementos muy parecidos, es que antes de estudiar en detalle el funcionamiento, estructura interna, etc. de los mismos, vamos a realizar una clasificación lo más completa y simple posible.

## **CLASIFICACION GENERAL**

Esta clasificación se presenta muy compleja pues se establece mucha confusión debido a que cada fabricante establece su nomenclatura particular, con la finalidad de lograr su propio espacio de poder en la comercialización de dichos dispositivos. Cada empresa fabricante de circuitos integrados digitales establece un nombre, aunque dicho producto ya exista en el mercado con otra denominación, aun más correcta. Generalmente estas ideas se aclaran estudiando detenidamente el significado de dichas siglas.

Por lo antes, indicado realizaremos familias de PLD y ahora las analizaremos brevemente, para luego dedicarnos de lleno al estudio pormenorizado de cada una de ellas.

1. **Redes Lógicas Programables, PAL** (Programmable Array Logic) ó **PLA** (Programmable Logic Array). Se las puede dividir en tres grandes subfamilias, según el tipo de matriz programable que incluya en su estructura lógica:
  - a) **Matriz Y programable – matriz O programable: PLA.**
  - b) **Matriz Y fija – matriz O programable: PROM**
  - c) **Matriz Y programable – matriz Y fija: PAL** propiamente dichas, y que según el tipo de dispositivo lógico que incluya en su estructura interna, pueden ser:
    - **PAL combinacionales** ó PAL simples.
    - **PAL secuenciales.** ó FPLS.

Desde el punto de vista de la tecnología de programación, se las divide en dos tipos:

- a) **PAL Bipolares.**
  - b) **PAL CMOS.**
- 
2. **EPLD** (Erasable Programmable Logic Device) Son circuitos lógicos programables eléctricamente y borrables con luz ultravioleta, que permiten desarrollar un diseño, borrando y regrabando hasta sacarle todos los problemas lógicos ó eléctricos. Tenemos las siguientes familias:
    - **EPLD Simples ó clásicas:** Familia de serie EPXXXX.
    - **EPLD Complejas ó CPLD (Complex Programmable Logic Device):**
      - Familia MAX ó de la serie EPMXXXX. Múltiple Array matriz.
      - Familia FLEX ó Flexible Logic Element Matrix.
      - Familia APEX ó Advanced Programmable Element Matrix.
    - **EPLD Específicas:** Familia de serie EPB  
Familia de serie EPS

3. **GAL** (Generic Array Logic) Son equivalente a las memorias EEPROM, es decir que permiten ser borradas y grabadas eléctricamente. Se pueden establecer las siguientes subfamilias:

**a) Tipo PAL:**

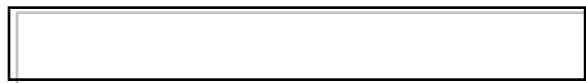
**Matriz Y / Macrocelda programable de salida:**

- Borrables y Programables fuera del circuito.
- Borrables y Programables en circuito.

**b) TIPO FPLA :**

**Matriz Y / Matriz O / Macrocelda programable de salida.** Se introduce un nuevo concepto más general de redes/macroceldas programables.

4. **ASIC:** Son circuitos digitales completos, de uso directo por los diseñadores, sin necesidad de programación posterior. Aparecieron al mismo tiempo que las PAL, pero como son programables por máscaras en fábrica sólo se aplica a grandes producciones en serie, pues su costo es elevado y su amortización se justifica solo en esos casos.
5. **LCA** (Logic Cell Array) ó **FPGA** (Field Programmable Gate Array) ó red de celdas lógicas. Podemos decir que es un producto similar a las ASIC, pero programables por el usuario. Se caracteriza por poseer una gran cantidad de bloques funcionales básicos, que en el estadio de diseño e implementación el usuario interconecta para obtener las funciones lógicas deseadas.
6. **FPGA de antifusibles:** se trata de redes de compuertas lógicas muy similares a las LCA, pero con una técnica de programación de las uniones totalmente diferentes, cuya denominación es la que identifica a estos circuitos, es decir antifusibles.

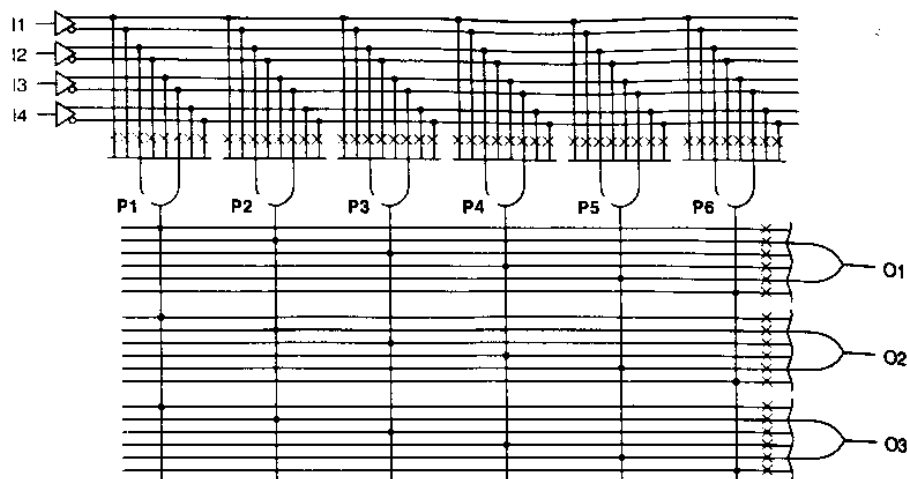


## 2 Redes Lógicas Programables

A pesar, de que en la sección anterior se ha realizado una clasificación de éstas redes programables, cabe indicar que ellas surgieron a comienzos de la década del '70 con el nombre de PAL. Este concepto fue introducido, y podemos decir inventado por la desaparecida empresa MMI (Monolithic Memories Inc.), actualmente absorbida por AMD (American Micro Devices). Cabe recordar que MMI fue pionera y primer fabricante de memorias PROM de fusibles; que no son ni más ni menos que PAL con matriz Y fija y O programable. Por ello, para ampliar el rango de uso de dispositivos programables introduce el concepto de matriz Y programable y matriz O fija. Evidentemente esta idea es original y simple, es decir, el uso de la tecnología de componentes electrónicos que se “queman” para obtener circuitos lógicos variados. Esto es, manipulados desde el exterior del circuito integrado, obtener redes lógicas cuyo conexionado interno se varía a voluntad del usuario.

### Concepto y Notación:

Generalizando, estas estructuras lógicas programables tienen una configuración interna de  $p$  productos lógicos y  $s$  sumas lógicas, con una cantidad  $n$  de entradas y  $m$  salidas (igual a  $s$ ). Por ello, con este tipo de dispositivo se pueden obtener funciones lógicas “sumas de productos”. No tienen la posibilidad de realizar todos los términos canónicos para tal cantidad de entradas. La forma de nominarlas es “ $n \times m$  con  $p$  productos”. En la figura N° 1 vemos una red lógica de “ $4 \times 3$  con 6 minterms”



**Fig N°1.1: Red lógica programable de “ $4 \times 3$  con 6 minterms”**

Vemos que esta representación, si bien es muy completa, no se utiliza en la práctica. Esto se debe a que ocupa demasiado espacio, haciendo muy complejos los esquemas representativos; especialmente considerando que las redes lógicas más usadas tienen decenas de compuertas lógicas.

En la industria se ha adoptado otra notación esquemática, para lo cual vemos en la fig N° 2 toda la evolución lógica justificando dicha representación.

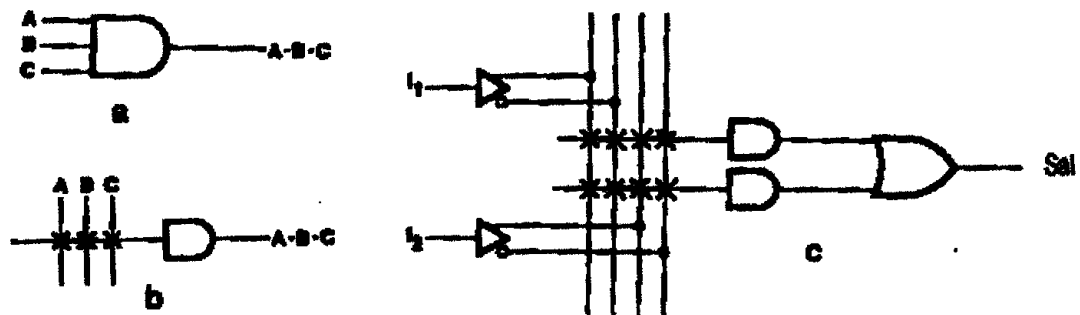


Fig. Nº 1.2: Origen y evolución de la notación para redes lógicas programables

En la fig. Nº 3 vemos la notación simplificada de la red lógica de “4 x 3 con 6 minterms” de la fig Nº1.

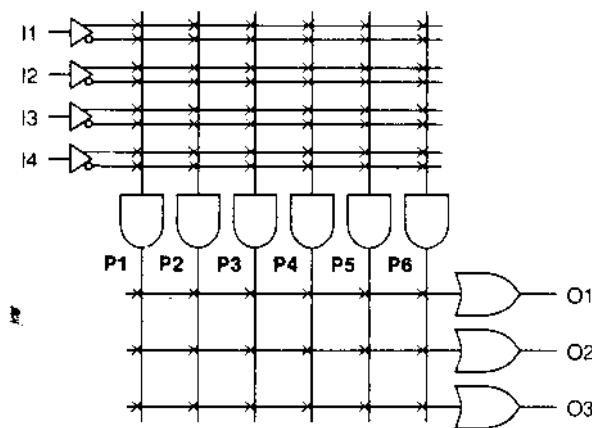


Fig. Nº 1.3: Notación simplificada de una red lógica programable de de 4 x 3.

En la fig. Nº 4 vemos la realización de las siguientes funciones de 4 variables.

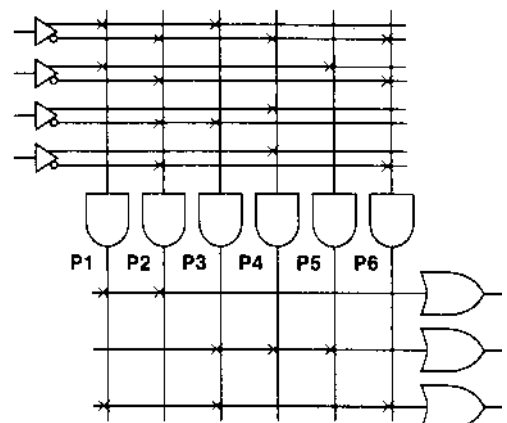
$$f_1(d,c,b,a) = \sum (0,3,7,11,15) = b \cdot a + d \cdot c \cdot b \cdot a$$

$$f_2(d,c,b,a) = \sum (1,2,3,6,7,9,10,11,12,14,15) = c \cdot a + d \cdot c \cdot a + b$$

$$f_3(d,c,b,a) = \sum (0,1,3,4,7,9,11,15) = b \cdot a + c \cdot a + d \cdot b \cdot a$$

El lugar donde no hay cruces significa que se han “quemado” los fusibles.

Fig. N 1.4: Implementación de las funciones f1, f2, y f3 indicadas.



## Clasificación:

Según se estableció oportunamente, las redes lógicas programables se trata de un arreglo estructurado de compuertas lógicas, agrupadas en dos matrices bien diferenciadas: una de compuertas AND y otra de compuertas OR, y según su programabilidad se las puede dividir en tres grandes subfamilias, según cual sea la matriz programable dentro dispositivo lógico:

- a) **Matriz Y programable / Matriz O Programable: PLA ó FPLA.**
- b) **Matriz O programable / Matriz Y Fija: PROM.**
- c) **Matriz Y programable / Matriz O Fija: PAL**

**a) PLA ó FPLA.** (Considerando Programmable Logic Array ó Field Programmable Logic Array)  
Se trata de las redes lógicas programables más completas, por cuanto contienen una “matriz Y y una matriz O programable”. Está representado por circuitos, como los ya descriptos, y vemos en la siguiente figura:

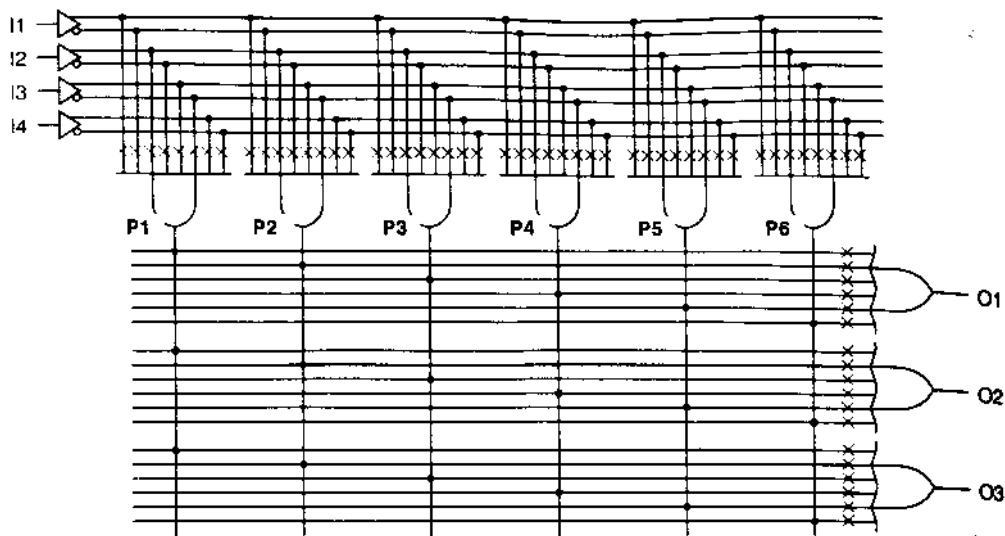


Fig Nº 1.5: Red lógica programable tipo PLA ó FPLA: matriz O y matriz Y programable

**b) Arreglos lógicos programables completos:** Lo cual se puede lograr a través de una “una matriz O programable y “matriz Y fija”. En otras palabras se dice que es un arreglo lógico completo, porque la matriz de puertas Y con una estructura fija e invariable constituye la zona de direccionamiento ó de decodificación de direcciones. Y la zona de compuertas O constituye el mapa de celdas de memoria propiamente dicho. Es decir que esta red lógica, no es más que una memoria programable de solo lectura ó PROM. En los primeros desarrollos constituyeron las típicas memorias PROM bipolar de fusibles. En la fig Nº 6 puede verse un diagrama estructural.

Podemos decir que antes de programar esta red lógica, las funciones de salida  $O_i$  son:

$$O_i(D;C;B;A) = \sum (0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)$$

**Direccionamiento**

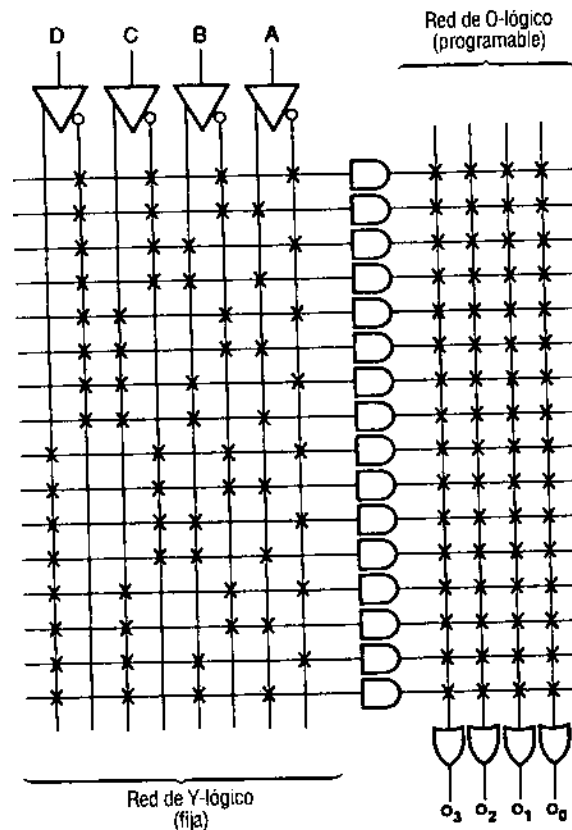


Fig Nº 1.6: Red lógica programable tipo PROM: matriz Y fija / matriz O programable

c) **PAL verdaderas** ó simplemente **PAL** ( Considerando **Programmable Array Logic**, el concepto introducido, por su creador MMI ): Tiene una estructura parecida a la anterior, pero aquí, por el contrario la “ la matriz O de salida es fija y matriz Y es programable “. Y podríamos aventurarnos a decir que la flexibilidad de uso de una PAL es similar a la de una PROM, ya que sólo se advierte una conmutación de las zonas programables. Pero como veremos luego, esto no es correcto, ya que los procedimientos son muy diferentes. Según el tipo de dispositivo lógico que incluya en su estructura interna, pueden ser del tipo combinacional ó secuencial; pero para poder analizarlas en detalle es preciso antes conocer algunos detalles constructivos y de nomenclatura nominativa, que veremos a continuación.

## **PAL: Estructura interna general**

En este punto examinaremos los componentes generales de estos dispositivos, externos e internos. Es decir, aquellos que puede acceder directamente el usuario, y a los que no lo hace directamente; pero ambos igualmente programables.

Todas las PAL contienen entradas, salidas, matriz programable arreglos lógicos combinacionales y celdas de salida. Estas celdas de salida pueden existir ó no, y en el primer caso pueden ser simples ó complejas, como vemos en una estructura típica y que describiremos a continuación:

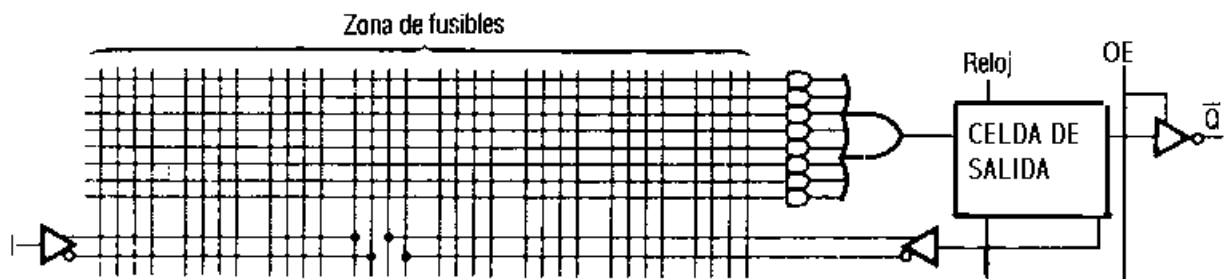


Fig N° 1.7: Estructura interna de una PAL en forma genérica

**Entradas:** Todas ellas poseen un cierto número, que invariablemente terminan en forma directa y complementada sobre la matriz de programación.

**Salidas:** A diferencia de las anteriores pueden tomar varias configuraciones. Siempre se realiza a través de un buffer de tres estados, directo ó inversor. Este buffer es controlado por dos posibles tipos de señales: desde la zona de fusibles, ó desde una entrada dedicada especialmente a tal función: habilitación de salida (**O**uput **E**nable). Por ello, esto implica que el estado de salida, lógico ó de alta impedancia, puede ser manejado por variables del sistema ó por señales dedicadas. Puede tratarse de un terminal de salida exclusivamente, ó de un terminal bidireccional, es decir que cumpla con dos funciones alternativamente, ó sea entrada/salida.

En las PAL secuenciales, la salida no tiene realimentación, y por lo general es controlada por una entrada dedicada a esa función, para todas las salidas en conjunto a través de una línea común llamada "habilitación de salidas" (**O**uput **E**nable).

**Realimentación:** La salida se reinyecta hacia la zona de fusibles programables a través de un dispositivo inversor-no inversor como los de entrada, con la intención de lograr:

- Una genuina realimentación de la salida hacia la zona de entrada.
- Utilizar la pata de salida, como una entrada alternativa cuando el buffer de salida se polariza en alta impedancia.

Este montaje es muy utilizado en todas las PAL, por ser de mucha utilidad práctica.

En el caso de arreglos combinacionales se realimenta directamente desde la salida.

En el caso de arreglos secuenciales no se realimenta desde la salida, sino desde la salida negada del correspondiente biestable.

**Arreglo lógico:** Aquí se establece la típica suma lógica de compuertas Y, es siempre del tipo combinacional, como se ve en la fig N° 7.

**Celda de salida:** depende del tipo específico de PAL.

En el caso de dispositivos combinacionales simples, esta celda no existe, y el arreglo lógico se conecta directamente con el buffer triestado de salida, como puede verse en la fig. N° 8.

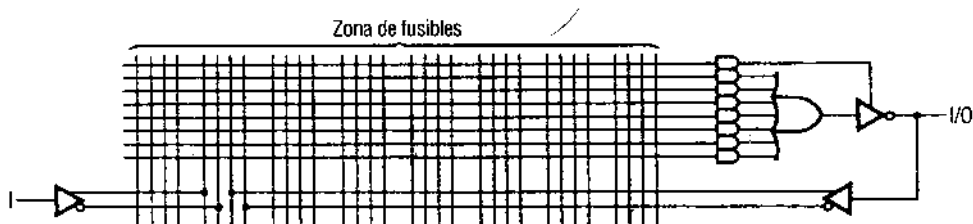
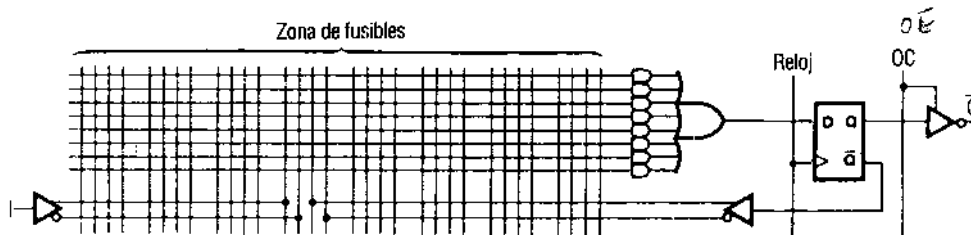


Fig N° 1.8: Estructura básica de una PAL combinacional simple.



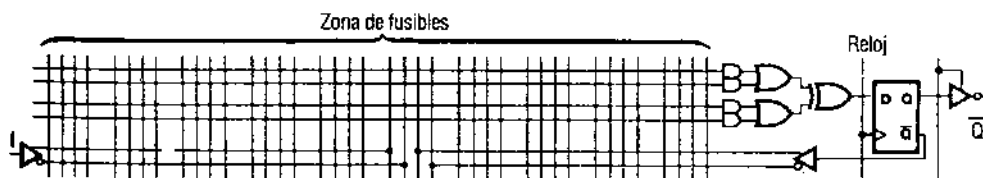
En el caso de PAL secuenciales simples, la celda de salida está realizada por un biestable, por lo general del tipo D, como se ve en la fig. N° 9. En estos casos los controles del reloj de los biestables son comunes a todas las salidas y manejados por una línea de específica llamada CLOCK.



**Fig N° 1.9: Estructura básica de una PAL secuencial simple**

En el caso de PAL complejas, la celda de salida está compuesta por varios elementos digitales, en general programables, como se verá luego, lográndose estructuras programables mucho más versátiles que las analizadas. Son ejemplos de este tipo las PAL Versátiles ó VPAL y las PAL Asíncrona de Registro. En estos casos los controles del reloj también son totalmente programables, en cuanto a la función que maneja su lógica, como el flanco que dispara al biestable correspondiente.

También existe un caso especial de arreglo lógico secuencial, que si bien no es muy generalizado, existe y es el caso de las PAL O-Exclusivas, mostradas en la figura N° 10. Son muy parecidas a las PAL secuenciales simples, sólo que incluyen una función O-exclusiva a nivel de la salida de la zona de fusibles. Esto permite la realización la función de memorización, especialmente en contadores y secuenciadores.



**Fig N° 1.10: Estructura básica de una PAL O-exclusiva.**

## **PAL: Nomenclatura**

Si bien existe una gran disparidad de referencias en el momento de catalogarlas, algunos fabricantes han tratado de establecer una normalización básica, cuyos lineamientos generales se indicará a continuación. Veremos la nomenclatura adoptada por AMD y Texas Instruments, que asemejan bastante sus principios.

**AMD                    T ( P )   X X   O   Y Y   P   V   DEF**

Referencias:

- **T** : Tipo de circuito programable, en este caso cabe siempre la sigla **PAL**
- **P**: Tipo de PAL . ( Nada ) PAL Bipolar

## CE PAL CMOS

- **XX** Número de entradas disponibles. Esta referencia se encuentra en todas las marcas de una u otra forma.
- **O** Estructura de salida. Consta de una ó dos letras según la siguiente tabla:

<b>Letra(s) Código(s)</b>	<b>Estructura de Salida</b>
L	<i>Combinacional activo bajo</i>
H	<i>Combinacional activo alto</i>
R	<i>Registro</i>
RA	<i>Registro asíncrono</i>
X	<i>OR-exclusiva+Registro</i>
V	<i>V-PAL ó Versátil</i>

Casi todas las marcas usan la misma nomenclatura.

- **YY** Número de salidas disponibles. Pudiendo ser salidas simples ó Entrada/Salidas También lo usan todos los fabricantes.
- **P** Potencia consumida: Normal, Bajo ó muy Bajo.
- **V** Velocidad de la PAL. Indicado a través de un N° que muestra el tiempo de propagación típico y expresado en nanosegundos (nseg).
- **DEF** Un conjunto de letras que en general describen varios parámetros del dispositivo, como: Tipo de encapsulado, rango de temperatura de trabajo, etc.

**Texas Instruments      TI F ( P ) XX O YY -V DEF**

Referencias:

- **F** : Familia tecnológica de **PAL**:  
**B: Bipolar**  
**C: CMOS**  
**E: ECL**
- **P**: Tipo de PAL . Según una clasificación propia de Texas.
- **XX** Número de entradas disponibles.
- **O** Estructura de salida. Idem a lo mencionado para AMD
- **YY** Número de salidas disponibles.
- **P** Potencia consumida: Normal, Bajo ó muy Bajo.
- **V** Velocidad de la PAL. Indicado a través de un Guión seguido de un número que indica el tiempo de propagación típico por puerta.
- **DEF** Un conjunto de letras que en general describen varios parámetros del dispositivo, como: Tipo de encapsulado, rango de temperatura de trabajo, etc.

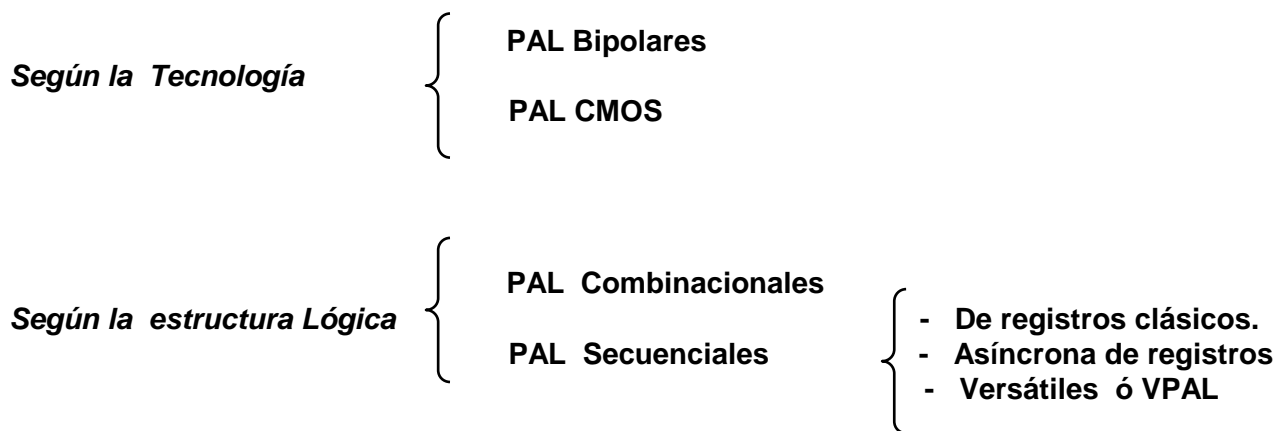
Ejemplos:

**PAL16R8H15PC** : Indica que se trata de una PAL Bipolar, que tiene 16 entradas y que es una PAL de Registros. Dispone de 8 salidas, es un modelo de mediana potencia , tiene encapsulado plástico DIL, y trabaja en el rango de temperaturas normales.

**TIBPAL16R8-15CN** : Indica que se trata de una PAL de texas, y es equivalente a la descrita para AMD en el ejemplo anterior.

## Clasificación:

Según la característica que se analice, se las puede clasificar de dos formas diferentes; como se ve a continuación:



**a) Considerando la tecnología de fabricación,** se las divide en dos tipos:

- **PAL Bipolares.** Son programables por fusibles, como las memorias PROM. Al igual que sus parientes, las PROM de fusibles, aunque son realmente de muy bajo costo, sólo permiten la programación por una única vez, pues lo que se realiza es la ruptura física de una conexión. La retención de la información es infinita, por cuanto dicha unión no puede volverse a recomponer. Por lo manifestado, vemos que no existe ninguna posibilidad de ser borradas, y las únicas modificaciones que se pueden realizar sobre una PAL programada es destruir los fusibles que aún están intactos. Como las PAL bipolares no pueden ser verificadas al 100 % en el proceso de fabricación, cuando se produce una partida con problemas de programabilidad verificables, las fabricas de los mismos las recambian en forma gratuita. Durante el período de implementación y prueba del prototipo y hasta tanto se ajuste el diseño, se realiza una gran quema de dispositivos, aumentando considerablemente el costo de desarrollo. Pero luego en el proceso de fabricación, una vez que el prototipo está “depurado” resultan con una mejor relación costo/performance. Los tiempo de propagación duplican a las PAL CMOS, y actualmente están en el orden de 4 a 5 nseg. En la actualidad, salvo casos muy especiales, han caído en desuso por ser obsoletas frente a otros productos, con mejores prestaciones.
- **PAL CMOS:** Para solucionar el problema presentado por las PAL Bipolares , se desarrollaron dispositivos programables eléctricamente y borrables eléctricamente ó por

radiación de luz ultravioleta. Por ello en la actualidad existen dos familias de PAL CMOS: UV-CmosPAL y EE-CmosPAL. Es un dispositivo idéntico a las GAL, pero como Lattice Semiconductor fue quien las puso primero en el mercado y obtuvo la patente correspondiente. Los nuevos fabricantes, como AMD y Cypres, las incluyeron dentro de las PAL, pero con el nombre de PAL CMOS. Considerando el concepto de borrado, las UVCmosPAL se parecen a los EPLD, y las EECmosPAL a las GAL.

Como mencionamos en el ítem anterior, la aparición en el mercado de las PAL CMOS (ó los circuitos programables equivalentes como GAL ó EPLD) ha posibilitado mejorar la fase de puesta a punto de un producto, por cuanto la posibilidad de borrado reduce notablemente los costos. Debido a la reducción considerable que han sufrido los costos de estos productos, es que también se los utiliza ampliamente en la fase de producción.

Existen reemplazos directos, de las Bipolares, no sólo de la estructura lógica sino pata a pata. Y como algunas familias han incorporado importantes innovaciones en su arquitectura interna, se pueden reemplazar todas las PAL bipolares por CMOS, apelando a una adecuada programación de las celdas de salida.

Actualmente, se han logrado tiempos de propagación muy buenos, en el orden de los 10 nseg; por lo cual la enorme ventaja de ser borrables, las hace un producto cada vez más usado. Resultando muy pocos los casos dónde la velocidad es el parámetro crítico, deba usarse otro componente programable.

Todas las PAL, tanto Bipolares como CMOS se programan con equipos apropiados, los cuales pueden ser autónomos ó simplemente placas para ser colocadas en computadores personales. También, en ambos casos, se puede realizar un desarrollo manual ó apelar a poderosas herramientas de diseño, con software dedicado a tal fin.

**b) Considerando la estructura lógica**, se las divide en dos tipos:

**1) PAL combinacionales** ó PAL simples. Constituyen los primeros circuitos lógicos desarrollados, cuyo aspecto programable está exclusivamente en la matriz pasiva.

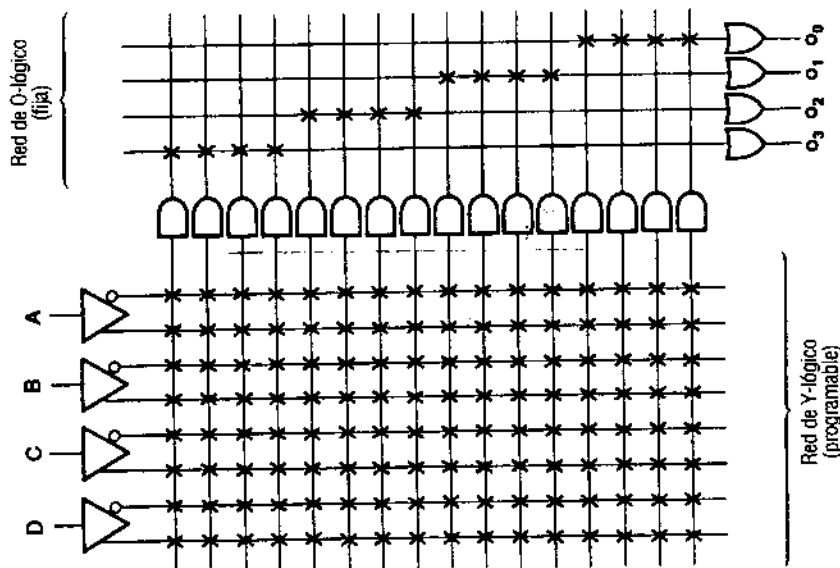


Fig Nº 1.11: Estructura típica de una PAL combinacional.

En el diagrama anterior, las funciones obtenidas son las siguientes:

$$O_i (D;C;B;A) = \sum (4 \text{ términos canónicos programados})$$

Podemos mencionar numerosos ejemplos, con arquitecturas internas diversas, pero siguiendo los lineamientos ya desarrollados. Permitiendo variar el número de entradas y de salidas, como también la configuración de ellas, de acuerdo a lo indicado. En estas últimas familias, en general la mayoría de los fabricantes las suministran con encapsulado DIL en circuitos integrados de 20 ó 24 patitas. Por ello, se las conoce prácticamente como la familia de las PAL de 20 patas ó de las familias de las PAL de 24 patas. Esto se puede verificar ampliamente en los manuales de cada fabricante. A título de ejemplo vemos a continuación la clásica PAL 16L8, encapsulada en un CI de 20 patas. Dicha nomenclatura define una PAL combinacional con salida activa en bajo, 16 entradas y 8 salidas.

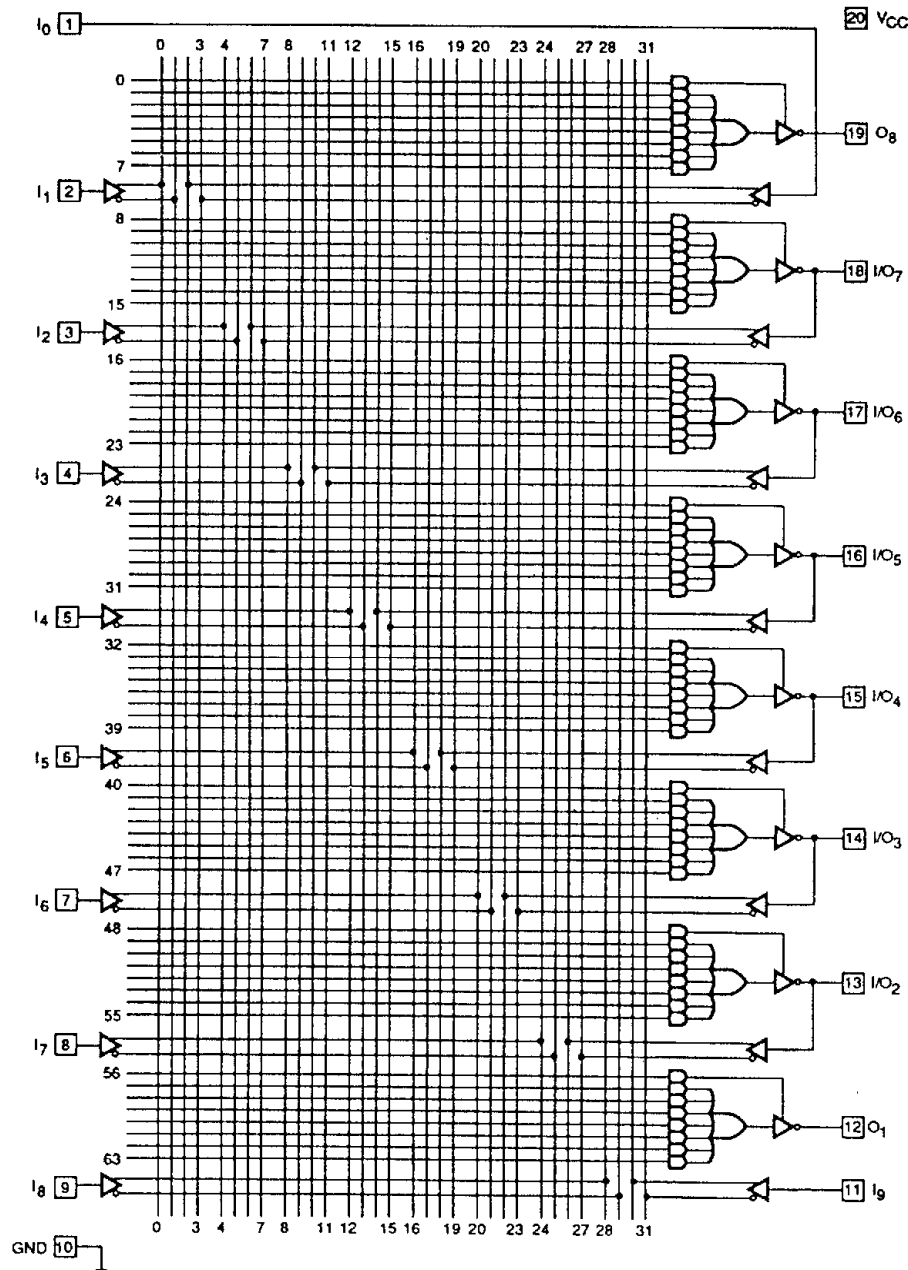


Fig N° 1.12:  
interna de

16L8.

Estructura  
una PAL

Siendo en este caso la función de salida :

$$O_i (i18,i17, ..., i2,i1) = OE^* \cdot \sum (7 \text{ términos canónicos programados})$$

Analizando la figura anterior, podemos ver que se trata de una configuración con 10 entradas específicas ( I ), 2 salidas específicas ( O ) y 6 entradas/salidas (I/O). Además en la siguiente tabla se puede apreciar una idea aproximada de los principales parámetros de esta PAL 16L8 de AMD.

Símbolo	Descripción	Mín.	Max.	Unid.
$t_{PD}$	Entrada ó realimentación a la salida combinacional	3	7	nseg
$t_{EA}$	Entrada a salida permitida usando <i>term. control</i>	3	10	nseg
$t_{ER}$	Entrada a salida no permitida usando <i>term. control</i>	3	10	nseg

**Tabla Nº 1.1: Principales parámetros de conmutación de la PAL 16L8.**

En la actualidad prácticamente han dejado de fabricarse, pues han sido desplazadas por circuitos más versátiles y complejos.

**2) PAL secuenciales** ó FPLS (Field Programmable Logic Sequence). Como su nombre lo indica, es un circuito lógico, que además de contener una matriz programable, incluye dispositivos secuenciales, como biestables y registros. Por lo tanto, esta arquitectura posibilita el diseño y realización de sistemas digitales más complejos, por cuanto permite la introducción de la variable tiempo en el diseño.

Se las puede clasificar en:

- a) De registros clásicos.**
- b) PAL asíncrona de registros.**
- c) PAL Versátiles ó VPAL.**

**a) De registros clásicos:**

Podemos decir lo mismo de lo enunciado para las PAL combinacionales, pues siguen los mismos lineamientos conceptuales: la única parte programable que tiene, es la matriz pasiva. Luego posee una red combinacional, y finalmente su salida es siempre a través de un biestable. Esta celda secuencial básica por lo general del tipo D activados por flancos.

En la figura Nº 13 vemos la clásica PAL 16R8 de AMD, la cual tiene:

- Ocho ( 8 ) entradas externas.
- Ocho ( 8 ) entradas internas (resultado de una realimentación secuencial).
- un ( 1 ) reloj ( **CLoK** ).
- una ( 1 ) habilitación de salida (**Ouput Enable**) de manejo externo.
- Ocho ( 8 ) salidas secuenciales con habilitación de tercer estado externo.

Existen algunas versiones, como la PAL 16R6 ó la PAL 16R4, que combinan en el mismo circuito integrado, un sector de PAL de registros y otro de PAL combinacional.



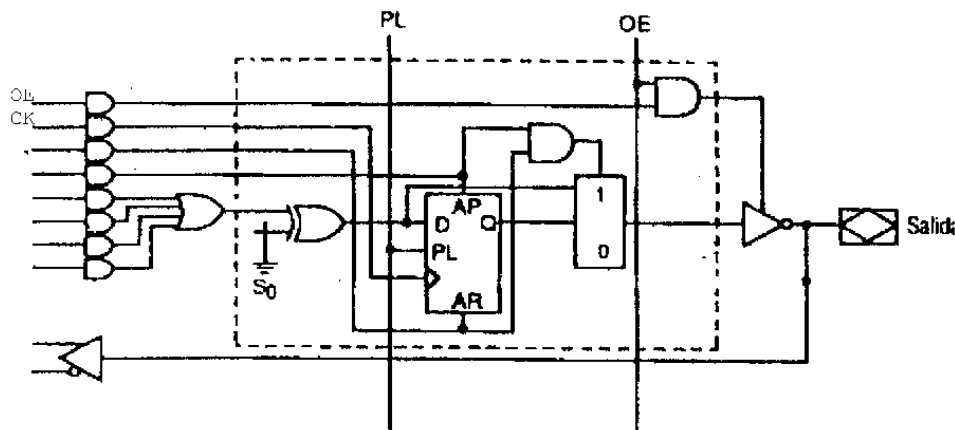


Fig N° 1.14: Celda de salida de una PAL asíncrona de Registros. (doc., AMD).

Como se puede apreciar, esta familia tiene la estructura de una PAL de registros, pero con una diferencia fundamental a las originales; y es que tal como mencionamos en párrafos anteriores, ofrecen una programabilidad propia.

Podemos diferenciar en esta arquitectura, dos tipos de formas de programarlas:

- **Interna:** Establecida por la programación de la compuerta Or-exclusiva. Pues, según como se programe el fusible  $S_0$ , la señal lógica proveniente del circuito combinacional programable ingresará al biestable en forma acertada ó complementada.
- **Externa:** Por programación de la matriz de fusibles permite configurar la forma de funcionamiento de esta macrocelda. Además al ser controlable, permite variar dichas configuraciones. Tenemos los siguientes casos:
  - a) **Preinicialización:** Hay dos que líneas permiten configurar la inicialización del biestable AR (Asynchronous Reset) y AP (Asynchronous Preset), desde la matriz.
  - b) **Tipo de funcionamiento:** Combinacional ó secuencial según se maneje el multiplexor de salida OMUX, a través de AP y AR. Pues seteando ambas en estado lógico alto, la macrocelda pasa a ser combinacional. Y en este caso se dice que el flip-flop está inhibido ó transparente.
  - c) **Manejo del Reloj.** Desde la matriz de fusibles.
  - d) **Habilitación de la salida:** Que está bajo el control de dos señales externas, pero de origen bastante diferente. Una señal global común a todas las salidas de la PAL, manejable directamente por el usuario, y otra local para cada macrocelda, desde la matriz de fusibles.

De acuerdo a lo descripto, una PAL Asíncrona de registros se puede configurar con una estructura combinacional según se puede apreciar en las figuras N° 15 y 16 .

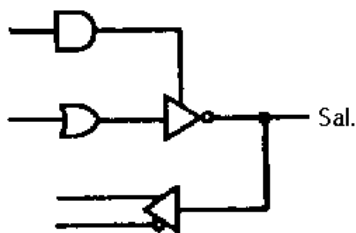


Fig N° 1.15: Combinacionales activa en bajo

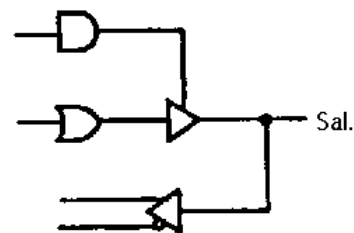


Fig N° 1.16: Combinacionales activa en alto

Y con una estructura secuencial según se puede apreciar en las figuras N° 17 y 18.



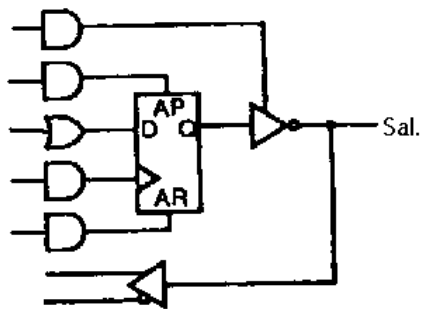


Fig N° 1.17: Secuencial activo en bajo

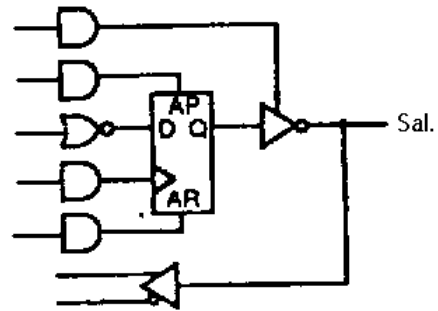


Fig N° 1.18: Secuencial activo en alto

A continuación vemos el diagrama interno de la PAL asíncrona de registros de AMD 16RA8, el cual nos permite inferir que una PAL de este tipo permite reemplazar cualquiera de las descritas anteriormente.

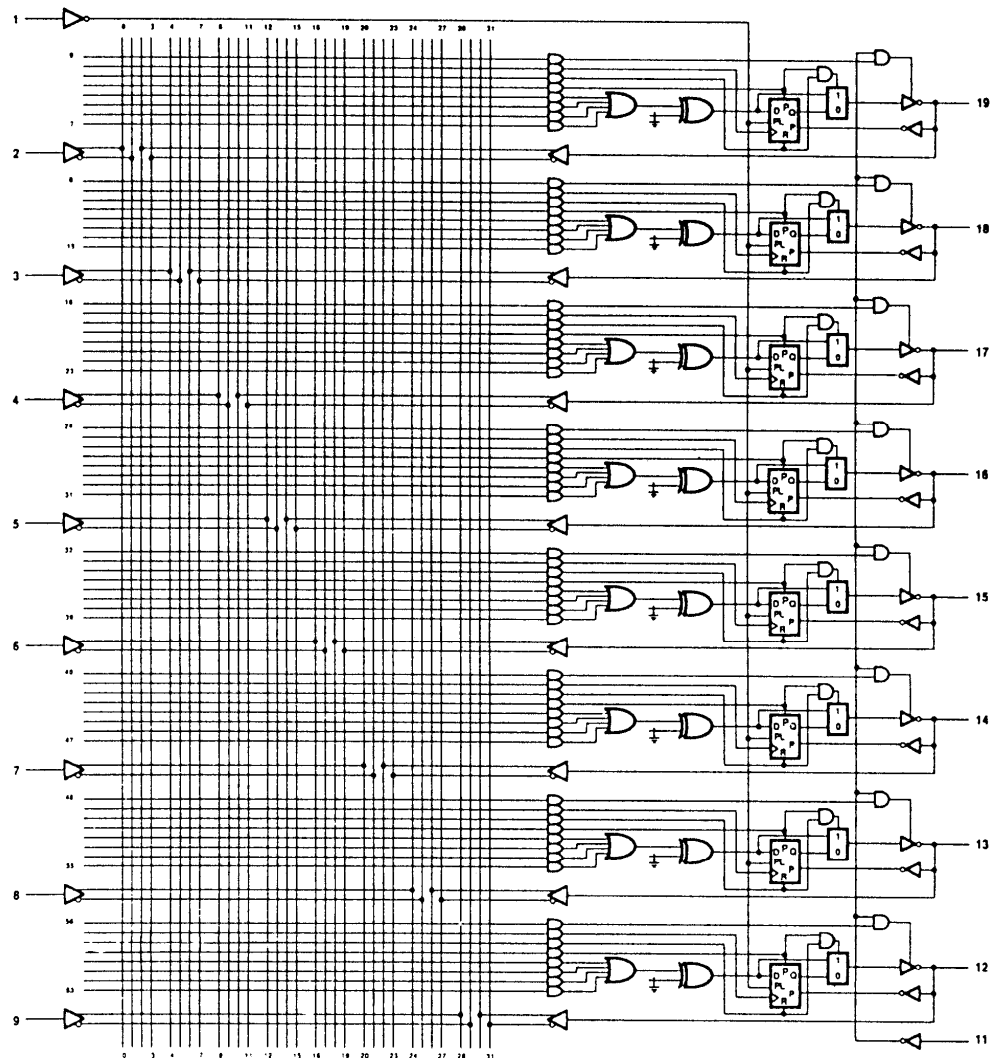


Fig N° 1.19: Configuración interna de una PAL 16RA8. (doc. AMD).

### c) PAL versátiles ó VPAL:

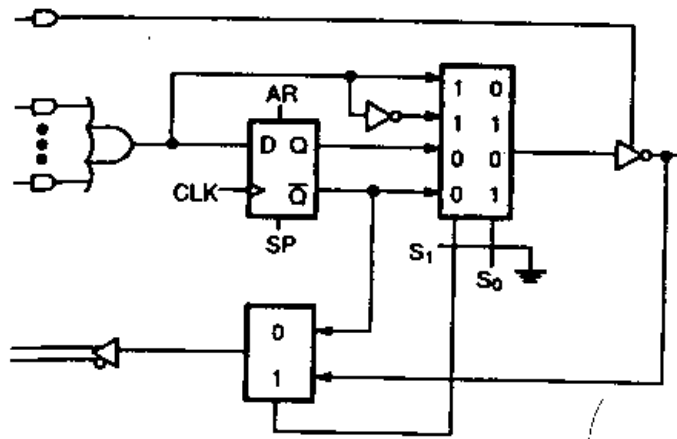
Esta familia de PAL evoluciona aún más hacia los circuitos programables de más alto nivel; pues la arquitectura del circuito lógico de salida introduce una

macrocelda aún más compleja en su estructura programable. Existen dos tipos de VPAL, caracterizada por la estructura de la celda de salida, pero fundamentalmente por el tipo de tecnología con el cual han sido fabricadas:

- PAL versátiles Bipolares o VPAL Bipolares .
- PAL Versátiles CMOS o VPAL CMOS.

a) **VPAL Bipolares**.

La celda de salida presenta amplias posibilidades de programación, como se ve en la fig. 20.

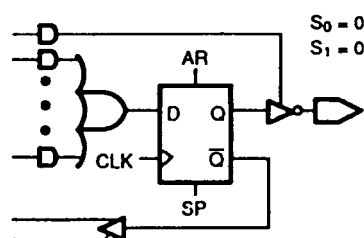


**Fig N° 1.20: Diagrama de la macrocelda VPAL Bipolar**

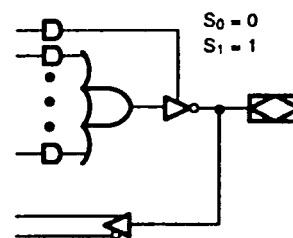
La arquitectura de esta macrocelda ronda alrededor de un biestable y dos multiplexores programables que sirven para configurarla. Esta celda se puede configurar de cuatro modos diferentes. Ello se logra programando los fusibles  $S_0$  y  $S_1$ , según la sig. tabla:

$S_1$	$S_2$	Configuración Salida
0	0	Registro. Activo en bajo
0	1	Registro. Activo en alto
1	0	Combinacional, activo en bajo
1	1	Combinacional, activo en alto

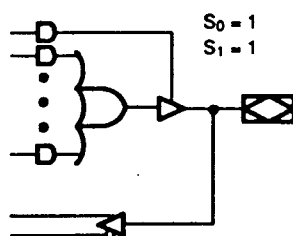
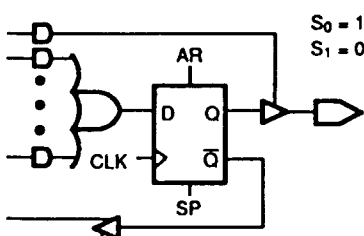
Bajo estas condiciones, la celda toma alguna de las siguientes configuraciones; dónde podrá observarse que tenemos todas las estructuras de las PAL simples.



**Registro, activo en bajo**



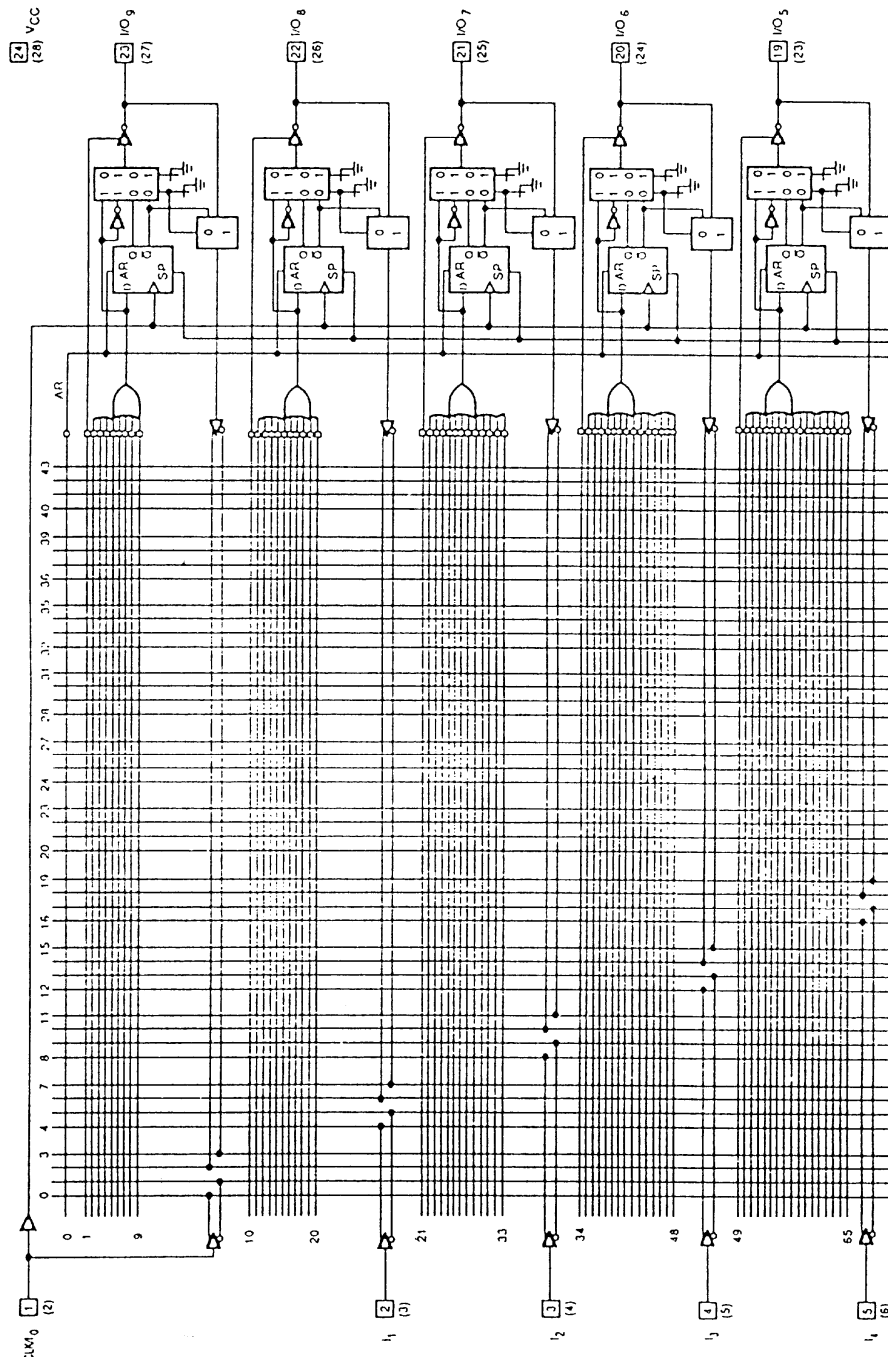
**Combinacional, activo en bajo**



**Registro, activo en alto**

**Combinacional, activo en alto**

A continuación, en la fig. N° 21 se presenta la estructura interna de la PAL 22V10 (sólo en forma parcial), que es del tipo VPAL Bipolar.



**Fig. n°**

**1.21: Configuración interna parcial de una PAL 22V10 (doc. AMD)**

En ella puede observarse que los biestables que conforman las celdas de salida pueden “setearse” ó “resetearse”, pero en forma global. Pues la puesta a uno y a cero son comunes a todos los biestables del circuito, y no independiente por cada una de ellas, como en el caso de las PAL asíncronas de registros. Además, no son accesibles directamente al usuario, sino sólo a través de la matriz de fusibles.

Además, por lo general se incluye una circuitería extra de preinicialización automática; con el fin de que todos los biestables lleven su salida a cero en el momento de dar tensión al dispositivo.

La PAL 22V10 es uno de los pocos circuitos de tecnología bipolar que aún existe en el mercado. Y esto se debe a varios motivos:

- Con esta única estructura es posible reemplazar cualquiera de las otras PAL bipolares.
- Aparecieron al mismo tiempo que las PAL CMOS; y como se verá a continuación, estas son algo más versátiles, desalentando por lo tanto cualquier nuevo desarrollo de esta tecnología.
- Es fabricada por varias empresas simultáneamente, como Texas Inst., AMD, etc.

## b) VPAL CMOS:

La empresa AMD es la mayor fabricante de VPAL CMOS actual del mercado, del cual tiene disponibles 4 familias. La base estructural de estas es muy parecido al mencionado en el ítem anterior para la 22V10, pero con una arquitectura de macrocelda de salida algo diferente, por supuesto más compleja y por lo tanto más flexible a las posibles programaciones. Esto se debe a que es más sencillo realizar integraciones en la tecnología MOS, y por lo tanto permite circuitos más complejos en pastillas parecidas.

A continuación presentamos la celda básica de una VPAL CMOS, y que permanece inalterada en las cuatro familias actuales.

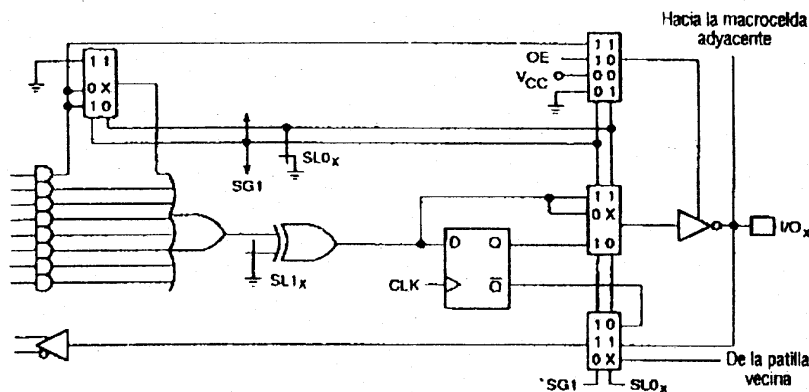
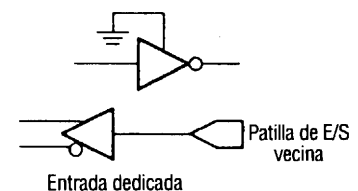


Fig. nº 1.21: Macrocelda de una PAL Versátil CMOS ó VPAL CMOS(doc. AMD)

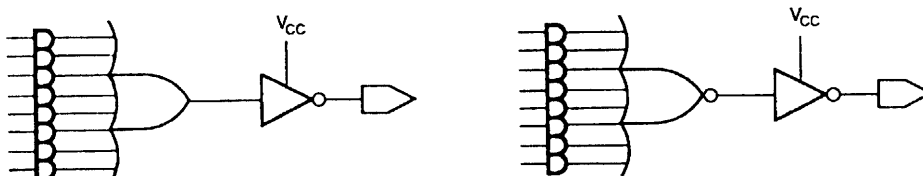
Vemos un biestable y tres multiplexores que permiten una gran cantidad de configuraciones. El biestable no posee ningún tipo de preinicialización, pero según se programen los fusibles  $SG_1$ ,  $SL_0$  y  $SL_1$  se logra una gran cantidad de salidas diferentes; como se ve a continuación:

$SG_1$	$SL_0$	$SL_1$	Configuración
0	0	0	Salida Combinacional activo en bajo.
0	0	1	Salida Combinacional activo en alto.
0	1	x	Salida anulada/Entrada dedicada.
1	0	0	Salida con registro activo en bajo.
1	0	1	Salida con registro activo en alto.
1	1	0	Entrada/Salida combinacional activo en bajo.
1	1	1	Entrada/Salida combinacional activo en alto.

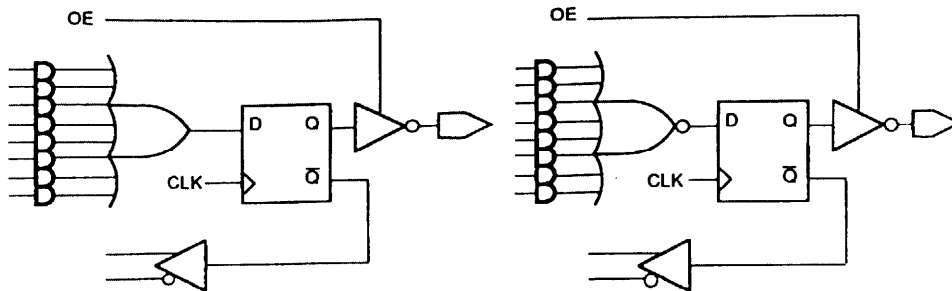
Tabla Nº 1.2: Programación de la macrocelda VPAL CMOS.



01x Sal. anulada/Ent. dedicada



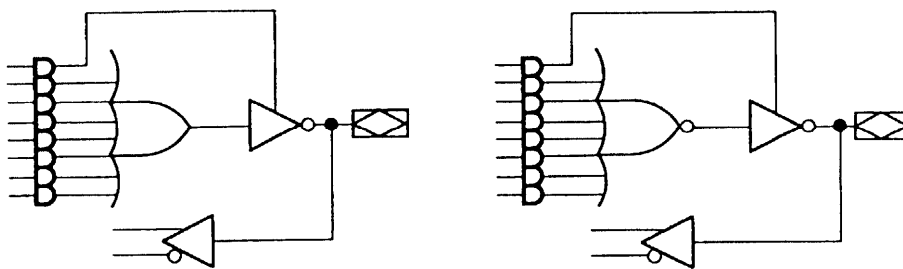
000 Salida combinacional activo en bajo    001 Salida combinacional activo en alto



a registro activo en bajo

101 Salida a registro activo en alto

100 Salida



110 Entrada/Salida combinacional activo en bajo    111 Entra/Salida combinacional activo en alto

En los manuales de este tipo de VPAL CMOS aparecen tablas, que indican según la programación realizada, el tipo de PAL combinacional ó a registros emulada. Así se pueden obtener reemplazos directos, no solo de la funciones lógicas, sino de la distribución física de patas en el Circuito Integrado que se desea seleccionar.

ooo O ooo

## 3 Redes Lógicas Genéricas

Son circuitos lógicos programables, cuyo nombre original es **Generic Logic Array** ó simplemente GAL. Son en esencia muy parecidos a las redes lógicas programables PAL, sólo que al ser introducidas por otro fabricante, les cambió el nombre. Esta empresa es Lattice Semiconductor, quien registró la marca en 1984.

Como aparecen después de más de una década que las PAL estuvieran en el mercado, logran reunir todas las mejoras introducidas por aquellas. Básicamente son muy parecidas a las PAL CMOS programables y borrables eléctricamente. Cuando Lattice introduce las GAL, las PAL CMOS aún no existían en el mercado, pero en la actualidad se han ganado un espacio importante dentro de los Dispositivos Programables.

### Generalidades:

Cuando Lattice se propone desarrollar un circuito lógico programable para competir con las PAL Bipolares, en auge en aquel momento, establece cinco objetivos básicos:

1. Lograr un circuito de reemplazo directo, no sólo lógico y funcional, sino físico pata a pata en cualquier aplicación. Premisa fundamental de competencia.
2. Contar con mayor flexibilidad de configuración y programabilidad de entradas/salidas.
3. Ofrecer un producto fabricado en una tecnología que permita velocidades comparables.
4. Permitir la verificabilidad 100 % de la zona de programación durante la fase de fabricación. Característica imposible de cumplimentar por las PAL Bipolares, pues ello equivaldría a hacer saltar los fusibles en dicha fase.
5. Ofrecer un consumo más reducido, para circuitos de complejidad equivalente.

Luego de obtener este tipo de circuito, el mismo fue bautizado como Generic Logic Array (GAL) en clara contienda con las tradicionales PAL de MMI. Luego AMD (ex MMI) produce circuitos PAL similares bajo la tecnología CMOS.

Existen en la actualidad varias familias de GAL, básicamente reunidas en dos diferentes tipos:

- 1) **Tipo PAL: Matriz Y / Macrocelda programable de salida.** Muy parecidas a las PAL asíncronas de registros y PAL Versátiles. Podemos encontrar dos familias bien diferenciadas, a saber:

- a) Borrables y Programables fuera del circuito.
- b) Borrables y Programables en circuito.

- 2) **TIPO FPLA : Matriz Y / Matriz O / Macrocelda programable de salida.** Se introduce un nuevo concepto más general de redes/macroceldas programables.

Esto se ve reflejado en la siguiente tabla, dónde se presentan alguna familias de GAL.

Tipo	Nomenclatura	Descripción	t <sub>PD</sub> (nseg)	I <sub>CC</sub> (mA)	Nº patas
------	--------------	-------------	------------------------	----------------------	----------

PAL osp	GAL16V8A	E <sup>2</sup> CMOS PAL Versatil Genérica	10, 15, 25	55, 115	20
	GAL 20V8A	E <sup>2</sup> CMOS PAL Versatil Genérica	10, 15, 25	55, 115	24
	GAL18V10	E <sup>2</sup> CMOS PAL Versatil universal	15,20	115	20
	GAL22V10	E <sup>2</sup> CMOS PAL Versatil universal	15,20,25	130	24
	GAL20RA10	E <sup>2</sup> CMOS PAL Asíncrona Reg universal	15,20	115	24
PAL isp	ispGAL16z8	E <sup>2</sup> CMOS PAL programable en circuito	20, 25	90	24
FPLA	GAL60001	E <sup>2</sup> CMOS FPLA	20, 25	90	24

**Tabla 2.1: Familias GAL**

Analizaremos a continuación en detalle cada componente de esta clasificación. Como hemos visto en este tipo de circuito programable ya no aparecen los equivalentes a PAL simples, combinacionales ó a registros.

## 1) Tipo PAL: Matriz Y / Macrocelda programable de salida.

a) **Borrables y Programables fuera del circuito.** Existen dos opciones, muy parecidas a las PAL más complejas. En ambos casos la matriz Y programable es del mismo tipo y se diferencia de aquellas que la misma no es de fusibles, sino CMOS.

- **a1) GAL Versátiles:**

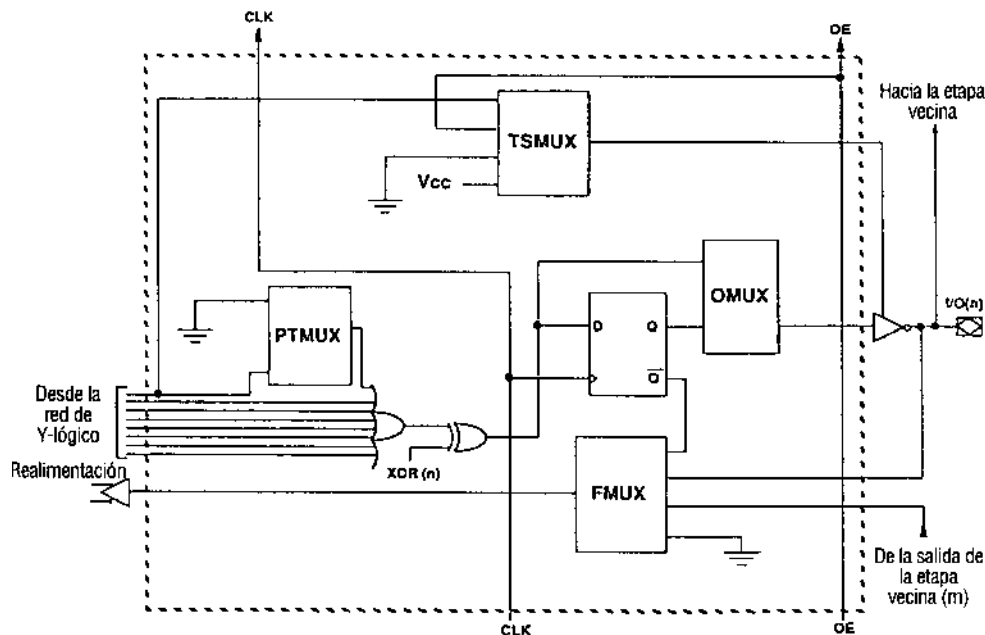
También llamadas GAL del tipo **XXVYY**, pues poseen una nomenclatura idéntica a las PAL V, dónde tenemos:

XX entradas;

V cada salida a través de celdas programables

YY salidas.

A continuación vemos la celda de salida de todas las GAL tipo V, a las cual Lattice las identifica con el nombre **OLMC** ( **O**uput **L**ogic **M**acro**C**ell ). Es muy parecida, salvo condiciones de programación a las celdas de salidas de las VPAL desarrolladas por AMD en tecnología PAL EE CMOS.



**Fig. Nº 2.1:**

**de salida de las GAL tipo V**

**Celda**

Esta celda de salida tiene cuatro multiplexores para lograr la máxima cantidad de configuraciones:

- **Tipo de salida:** Permite configurar si la celda será combinacional ó secuencial.
- **Habilitación de salida:** Permite seleccionar que variable manejará la salida.
- **Entrada de información:** Indica la señal que ingresa desde la red Y-programable.
- **Realimentación:** Permite seleccionar que señal se realimentará a la matriz Y.

Al igual que las PAL V disponen de la circuitería necesaria para realizar un reset automático al dar tensión al dispositivo lógico, de modo que los biestables de todas las macroceldas vayan a un estado definido (  $Q = 0$  ). En la siguiente figura se puede observar la arquitectura interna de la GAL 16V8A.

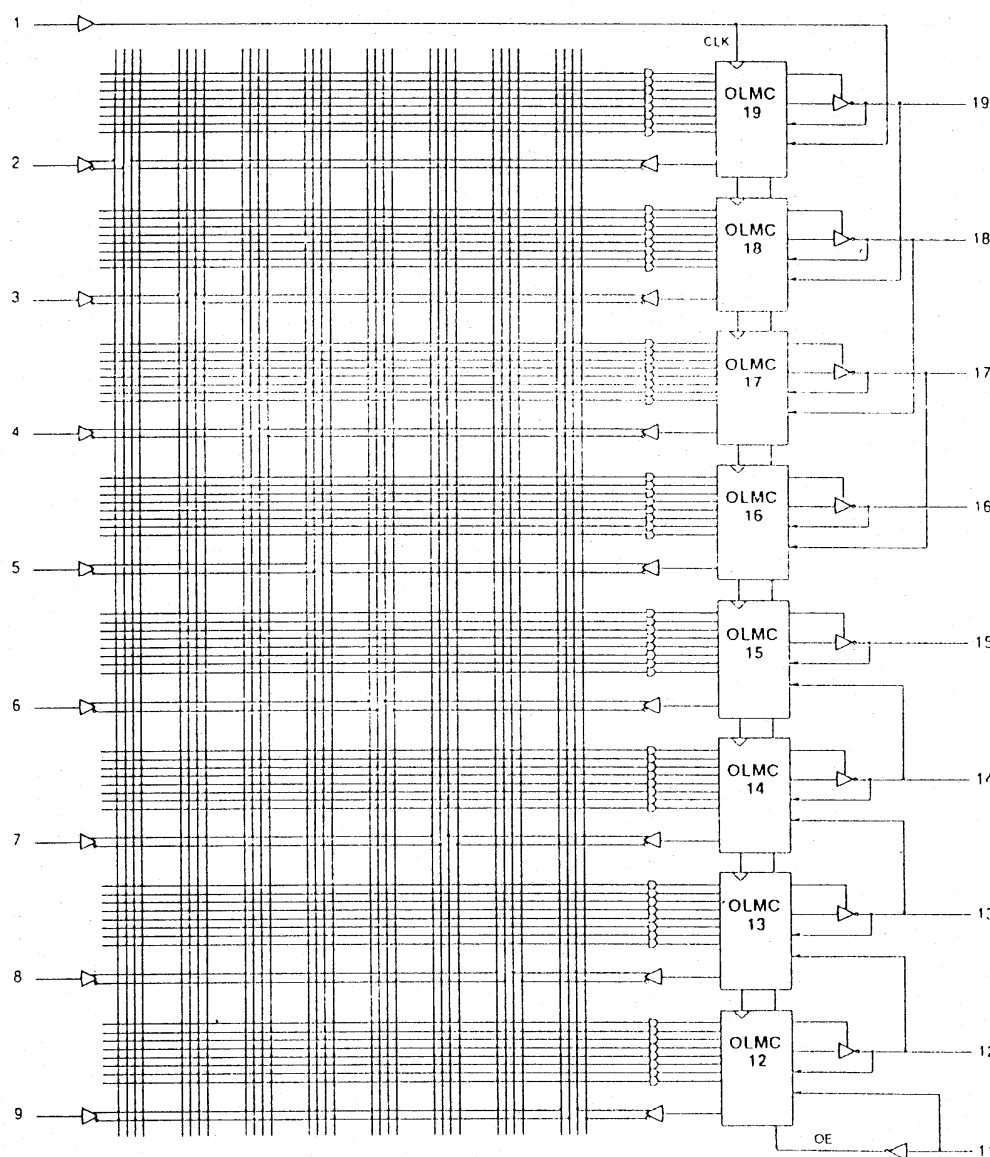


Fig. N°

GAL 16V8A: Configuración interna ( doc. Lattice )

2.2 :

A continuación vemos las principales características de conmutación de la GAL presentada, y que nos habla del parecido con las PAL equivalentes.



Parámetro	Desde	Hasta	Descripción	Mínimo	Máximo	Unidad
$t_{PD}$	I, I/O	O	Tiempo de propagación combinacional	3	10	nseg
$t_{CO}$	CLK	Q	Tiempo de Reloj a Salida	2	7	nseg
$t_{EN}$	I, I/O	O	Tiempo Salida permitida, Z a O.	-	10	nseg
$t_{EN}$	OE*	Q	Tiempo Salida registro permitida, Z a Q.	-	10	nseg
$t_{DIS}$	I, I/O	O	Tiempo Salida NO permitida, O a Z.	-	10	nseg
$t_{EN}$	OE*	Q	Tiempo Salida NO permitida, Q a Z.	-	10	nseg
$t_{SU}$	-	-	Establecer tiempo, entrada ó realim, antes Clk	10	-	nseg
$t_h$	-	-	Retener tiempo, entrada ó realim, antes Clk	0	-	nseg
$t_W$	-	-	Duración impulso reloj, Alto / Bajo	8 / 8	-	nseg
$f_{clk}$	-	-	Frec. reloj sin realimentación = $1/(t_{WH} + t_W)$	-	62.5	Mhz
$f_{clk}$	-	-	Frec. reloj con realimentación = $1/(t_{SU} + t_{CO})$	-	58,8	Mhz

**Tabla Nº 2.2: Características de conmutación de la GAL 16V8A (principales).**

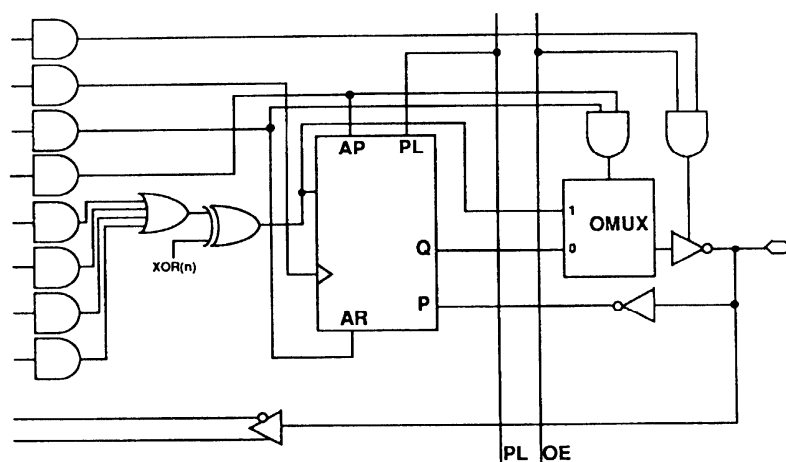
Vemos que tiene una estructura interna muy parecida a una PAL equivalente, con la única diferencia y ventaja que los bloques OLMC tienen mayor flexibilidad de programación que las celdas de salida de las VPAL, y además son borrables eléctricamente.

También podemos apreciar que son circuitos bastante rápidos a pesar de ser tecnología CMOS, pues sus parámetros de conmutación se asemejan a los presentados en tecnología Bipolar. El consumo resulta bastante menor que el de sus PAL Bipolares similares. Sabemos que en todo circuito CMOS la potencia disipada depende de la frecuencia, siendo un consumo típico a 25 MHZ para la 16V8A 75 mA con alimentación de 5 volts.

- **a2) GAL Asíncronas de Registros:**

Este tipo de GAL, si bien sigue los lineamientos generales descriptos para ellas, como su nombre lo indica fue realizado fundamentalmente para competir con el mercado idéntico de PAL; por lo tanto su macrocelda de salida es diferente a las del tipo V.

A continuación vemos la celda típica de una GAL asíncrona de registros, y que es idéntica a una PAL ; con la única diferencia que ésta puede ser borrada y grabada eléctricamente.



**Fig. Nº 2.3 : Celda de salida de las GAL asíncrona de Registros (Incluida en la GAL 20RA10)**

**b) Borrables y Programables en el circuito ó ispGAL(in system programming):**

Podemos decir que son idénticas a las GAL de tipo V, pero con la ventaja que pueden ser programadas y reprogramadas en cualquier instante en el propio circuito

sobre el cual están montadas, sin necesidad de ser extraídas. Esta programación usa la tensión de alimentación normal (5 volts), lo que da aún más ventajas a la instalación.

Con el objeto de realizar la programación mencionada, sin interferir con el funcionamiento lógico de las mismas, presenta 4 patas especiales:

- **Modo ( Mode )**: Permite la configuración en los diferentes modos de trabajo.
- **Reloj de Datos (Data Clock) DCLK**: Es el reloj de validación en la fase de programación ó validación.
- **Datos de entrada Serie**: (Serial Data Input) **SDI**: Datos de entrada de programación.
- **Datos de salida Serie**: (Serial Data Output) **SDO**: Datos de entrada de programación

A continuación vemos la estructura interna de una GAL de este tipo, la ispGAL16Z8.

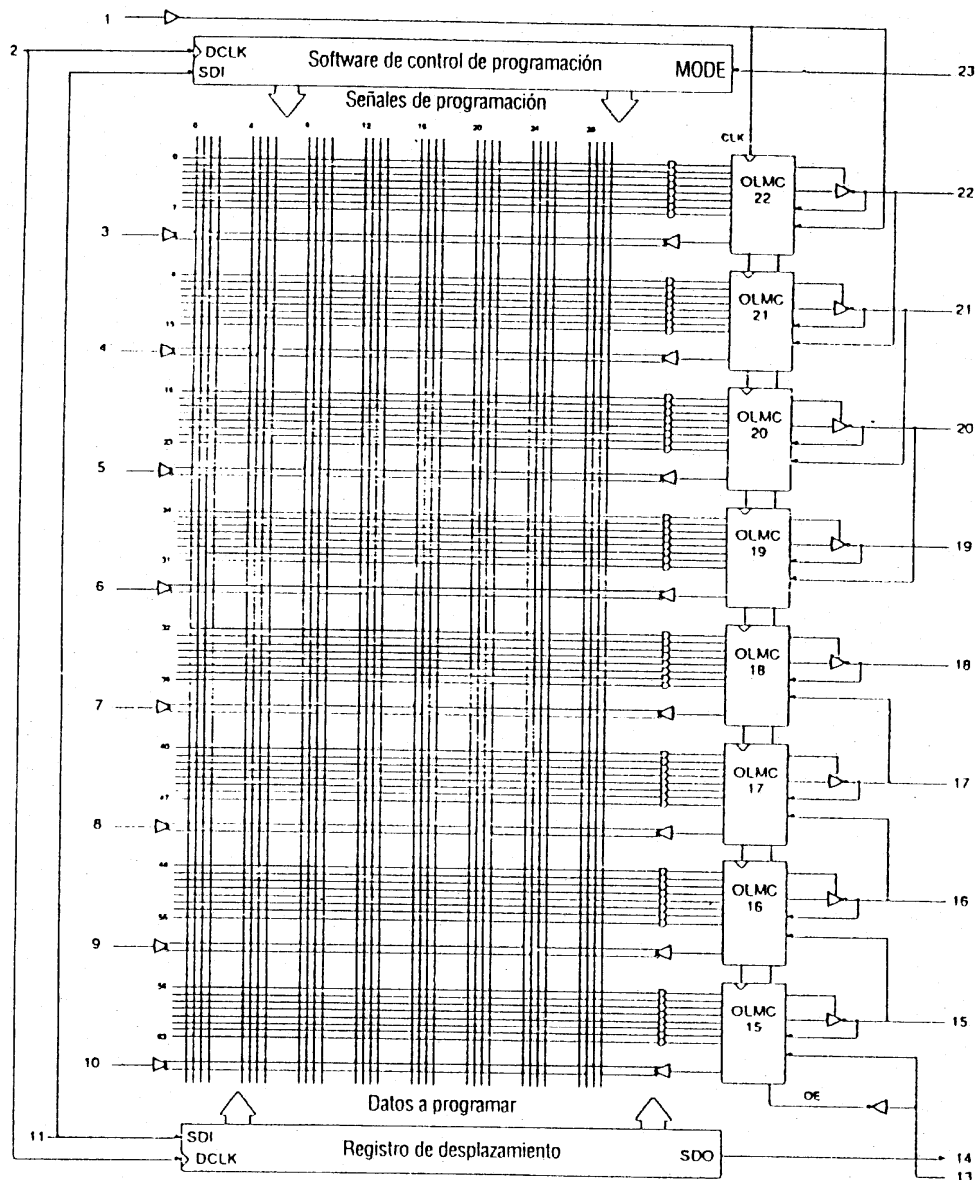


Fig. Nº

#### Configuración interna dela ispGAL 16Z8A ( doc. Lattice )

Podemos ver que la estructura interna ó estructura lógica funcional es idéntica a la de su equivalente programable fuera del circuito GAL16V8. Inclusive las macroceldas de salida OLMC son iguales. Pero aparecen dos circuitos de control especiales, que utilizan las variables descriptas, para realizar la programación y reprogramación " in circuit " de las mismas. Luego que la GAL ha sido programad, estos complejos procesadores de programación no interfieren para nada con la lógica de trabajo establecida.

2.4 :

Como se observa en el siguiente esquema, para simplificar el desarrollo de las aplicaciones que utilizan estos circuitos, los mismos pueden conectarse en cascada, a través del uso de las señales de control definidas precedentemente.

Este circuito cuenta con una lógica de control que puede programar varios circuitos simultáneamente. La misma también sirve para leerlos con el objeto de verificar su correcta programación. Esta lógica está realizada a través de un sistema secuencial síncrono de proceso discreto ó integrado, como en caso de un microprocesador ó microcomputador.

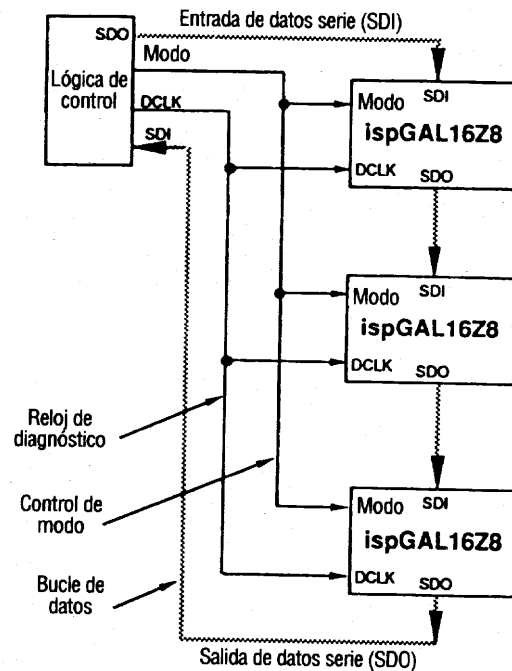
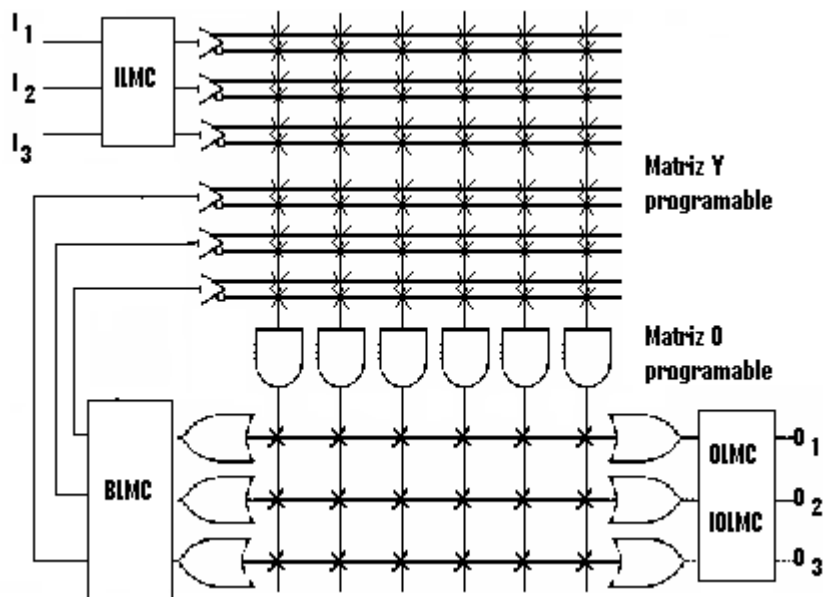


Fig. Nº 2.5 : ispGAL 16Z8A - Configuración para programación en cascada ( doc. Lattice )

## 2) Tipo FPLA : Matriz Y / Matriz O / Macrocelda programable de salida.

Este tipo de GAL se diferencia de las anteriores pues su estructura cambia radicalmente, siendo muy compleja y en la cual tenemos los siguientes componentes lógicos:



### 1. Matriz Y de entrada programable.

### 2. Matriz O de salida programable.

En ambos caso idénticos a los ya descritos, pero con la salvedad que se tratan de celdas CMOS programables, y no de fusibles. La aparente complejidad de la estructura interna de una GAL de este tipo está dada por la doble estructura ("Y" y "O") programable. Que como se verá luego en el esquema correspondiente, en el caso de la estructura interna de la GAL 6001, ocupa la mayor parte del mismo.

### 3. ILMC: ( Input Logic MacroCell )Macroceldas de entrada.

### 4. IOLMC: (Input Output Logic MacroCell ) Macrocelda de entrada/salida.

A continuación se ve la estructura típica de este tipo de celda programable. Además se presentan las tres configuraciones que pueden llegar a adoptar luego de programadas:

- **Entrada pasiva:** En el cual puede verse una entrada sin ningún tipo de dispositivo activo, sólo una conexión directa a la matriz de entrada. En este caso la señal lógica tiene dos posibles caminos de entrada, en forma directa ó inversa, como en una PAL clásica.
- **Entrada a registro D activo por flancos:** En este caso la señal lógica es aplicada a la entrada de un biestable tipo D. La señal de reloj de este flip-flop es común a todas las macroceldas de entrada y se la indica como ICLK ( Input ClOCK ).
- **Entrada registro D latch:** Vemos que la representación es casi idéntica a la anterior, pero tiene la diferencia funcional de los biestables involucrados. En este tipo, cuando la señal de reloj está activa (en este caso nivel alto), la celda se comporta como una línea pasiva, es decir toda señal presente en la entrada pas directamente a la salida. Pero cuando se inactiva, se cierra el flip-flop, enclavando en la salida el estado inmediatamente anterior. Aunque ahora la señal cambie.

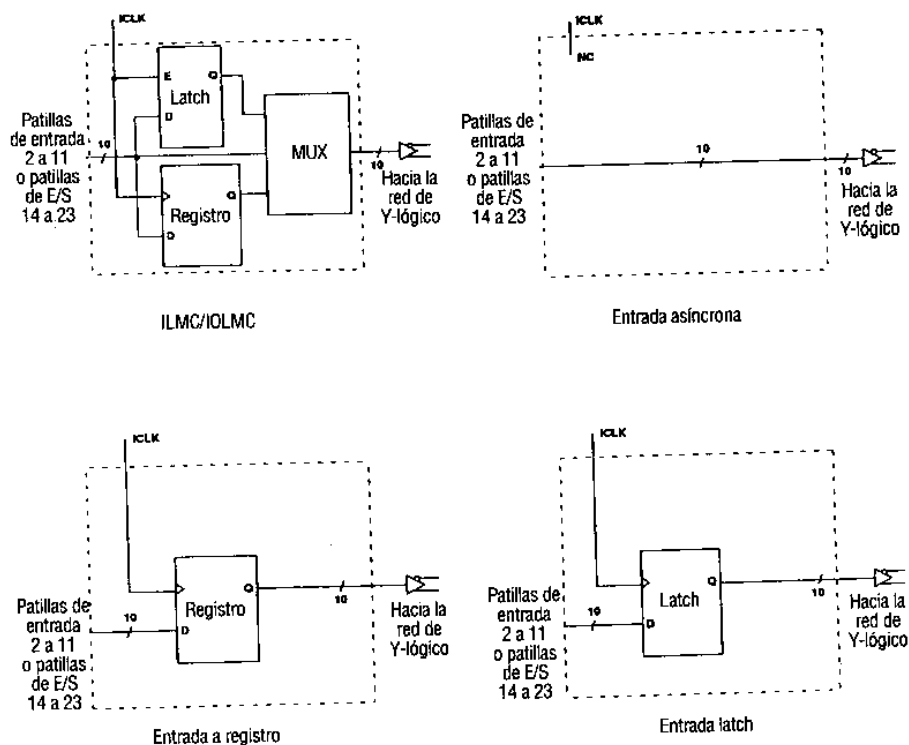


Fig.

**GAL Tipo FPLA: Macroceldas de entrada y entrada/salida, en sus distintas configuraciones**

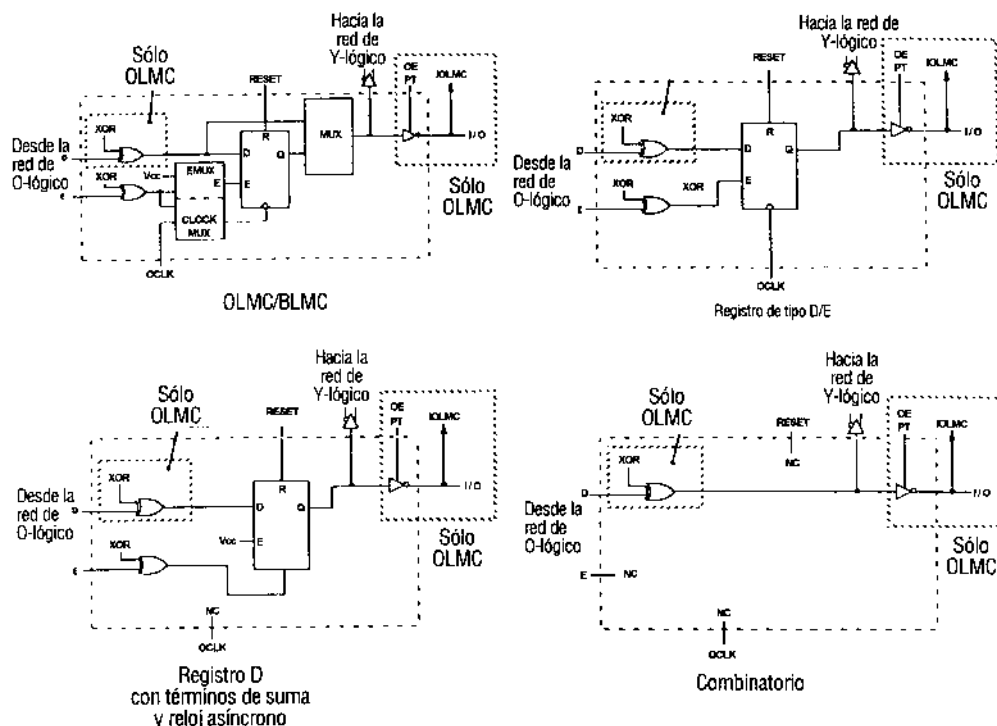
Nº 2.6 :

5. **OLMC: ( Output Logic MacroCell ) Macrocelada de salida.** El concepto de esta macrocelda es igual al de todas las PAL y GAL vistas .
6. **BLMC: ( Blank Logic MacroCell ) Macroceladas ocultas.** Esta macrocelda introduce un nuevo concepto, y es que recibe la señal lógica de la matriz “ O “ programable de salida y la reinyectan a la matriz “ Y “ de entrada. Debido a su ubicación estratégica en el circuito, junto a su capacidad de programación permite relaizar iteraciones internas de alta complejidad y muy flexibles.

Como vemos en el siguiente esquema, estas macroceldas son muy parecidas y pueden programarse de tres modos diferentes:

- **Modo Combinacional:** En el cual todos los biestables y registros están eliminados circuitalmente.
- **Entrada a registro D, con reloj independiente:** Poseee una entrada de inhibición E, que maneja la actividad de dicho biestable, desde la red lógica “O” programable. Tiene un control de reloj común a todas las macroceldas de salida y ocultas, pero diferente a las de entrada. Proviene directamente de una pata del encapsulado, llamada OCLK ( Output ClOCK ).
- **Entrada a registro con reloj programable:** Vemos que la representación es casi idéntica a la anterior, pero tiene la diferencia funcional de los biestables involucrados. Cuando la señal de reloj está activa (en este caso nivel alto), la celda se comporta como una línea pasiva, es decir toda señal presente en la entrada pasa directamente a la salida. Pero cuando se inactiva, se cierra el flip-flop, enclavando en la salida el estado inmediatamente anterior. Y aunque ahora la señal cambie permanece dicho nivel.

En los tres modos de programación , en el caso de las celdas OLMC, tiene una entrada OR-Exclusiva que permite seleccionar la polaridad de la señal de activación del biestable, como puede verse en la sig figura.

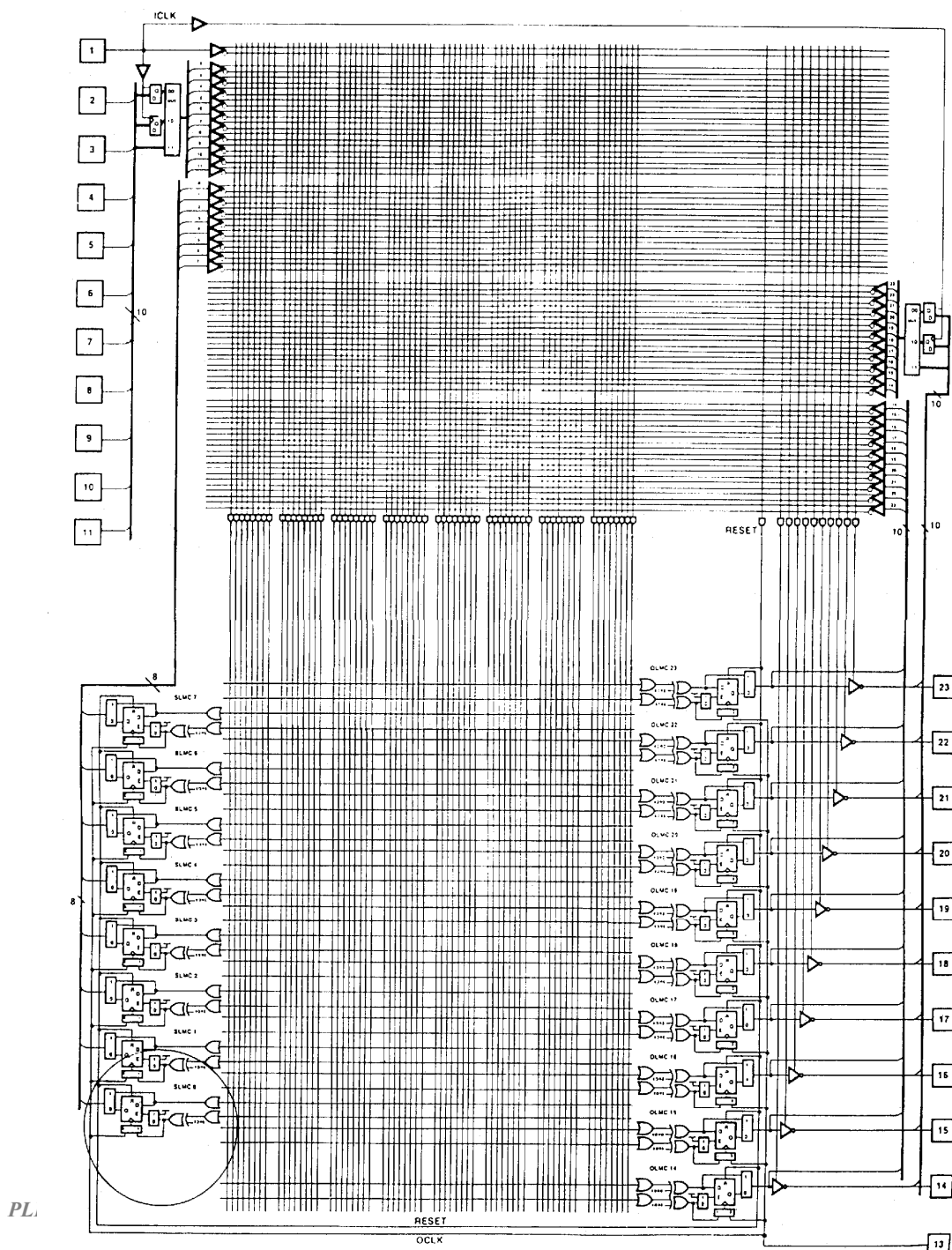


**Fig. Nº 2.7 : GAL Tipo FPLA: Macroceldas de salida y ocultas, en sus distintas configuraciones.**

La familia GAL 6001 de Lattice es un representante típica de esta tecnología. Es mucho más flexible que las PAL ó GAL clásicas. Permitiendo configuraciones muy complejas, aptas en el desarrollo e implementación de sistemas secuenciales de control, máquinas de estado ó autómatas programables.

A continuación realizaremos algunos comentarios sobre la estructura interna, pues tiene algunas características muy particulares, para presentar luego dicho el diagrama. Podemos decir que posee:

- Diez (10) Macroceldas ILMC
- Diez (10) Macroceldas IOLMC
- Diez (10) Macroceldas OLMC.
- Ocho (8) Macroceldas BLMC. En el diagrama se ve una de ellas rodeada con un círculo.



**Fig. Nº 2.8: GAL 6001: Estructura interna ( doc. Lattice )**

Además la matriz de entrada y salida son excepcionales, por ello analizamos a continuación su estructura:

- Matriz Y de entrada: Está organizada en 78 líneas de entrada por 75 de salida. Estas corresponden a: 10 ILMC, 10 IOLMC, 8 retornos de las BLMC, 10 de retorno de las OLMC y una (1) línea de ICLK, las cuales suman 39, pero todas tiene su par directo y complemento alcanzando el valor indicado.
- Matriz O de salida: Está organizada en 64 líneas de entrada por 36 salidas, otorgando al circuito una programabilidad espectacular.

Al igual que las PAL y GAL CMOS, posee un circuito especial dedicado al reset automático cuando se aplica tensión al mismo; posibilitando partir de un estado inicial perfectamente conocido.



## *Dispositivos Lógicos Programables*

### **4 - EPLD**

Son circuitos lógicos programables, cuya característica más importantes, es que se pueden borrar y volver a programar. De allí nace su nombre ***Erasable Programmable Logic Device*** ó sus siglas ***EPLD***. Son básicamente circuitos lógicos programables eléctricamente y borrables por la exposición a radiación ultravioleta (UV), de una cierta longitud de onda y potencia. Fueron desarrollados por la empresa Altera, hacia fines del año 1984 e introducidos en el mercado de los circuitos lógicos programables para competir con las PAL, y las GAL de reciente aparición.

Por lo enunciado hasta ahora, respecto de los principios de programación y borrado ultravioleta, se puede llegar a pensar que sólo son una variedad más de PAL CMOS borrables. Pero no es así, ya que la arquitectura interna, los tipos de macroceldas y redes de interconexión son totalmente diferentes a las vistas para las PAL y GAL. También los procesos físicos de fabricación difieren bastante, ya que los niveles de integración actualmente realizados en los EPLD son mucho mayores que las PAL CMOS con funciones similares. En efecto, los EPLD actualmente comercializados integran desde varios miles ( 7000 a 8000) compuertas en los dispositivos de las familias clásicas, a varios millones en las familias de alta performance y prestación. Podemos decir entonces, que este tipo de circuito puede alojar comparativamente, esquemas lógicos que incluyan desde algunas décadas de PAL clásicas hasta varios cientos de ellas en el mismo encapsulado. También podemos concluir en que alojarán desde cientos a miles de funciones TTL típicas.

Cuando Altera entra en el mercado de este tipo de productos, se propone como premisa fundamental llegar a una gran masa de desarrolladores de sistemas digitales. Por lo tanto tiene la correcta visión de comercializar, (u obsequiar) herramientas de desarrollo y programación simples y de aprendizaje sencillo. Y esto lo consigue a través de un hardware y software adaptables a cualquier computador personal. Estas herramientas son de muy fácil manejo y de una gran compatibilidad con cualquier procesador actual, de tal manera que permiten a cualquier diseñador, desde los novatos a los más expertos, obtener una EPLD programada en pocas horas de desarrollo y con un bajo nivel de aprendizaje. Debido a lo reducido de estos costos, permite que tales dispositivos sean utilizados por cualquier pequeño laboratorio ó empresa electrónica, alcanzando en poco tiempo gran éxito comercial.



Con la comercialización de los EPLD, Altera buscaba tres objetivos básicos, a saber:

- **Gran Densidad de integración**, muy por encima de la que las PAL ofrecían en aquella época y muy cercana a lo que ofrecían las redes de compuertas programables (FPGA).
- **Alta velocidad de trabajo**, lo más cercana a las PAL Bipolares y muy por encima de las redes de lógicas tradicionales.
- **Posibilidad de borrado**, ya que en aquel momento habían muy pocos dispositivos que permitían tal característica.

Estas premisas, le aseguran una gran diversidad de productos; por lo que tienen la posibilidad de atacar todos los nichos del mercado de redes lógicas programables, como se ve en el siguiente esquema.

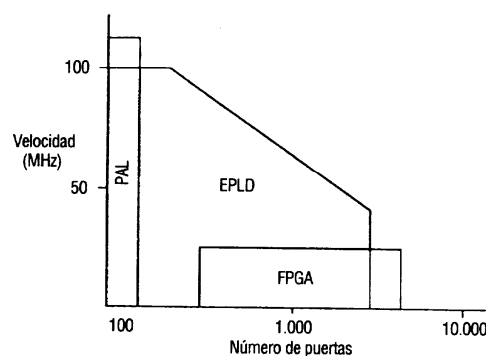


Fig. Nº 4.1: Campo abarcado por los EPLD.

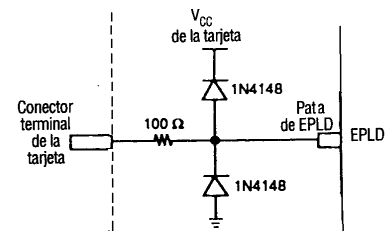
Se trata de circuitos fabricados en tecnología MOS, y más específicamente en la configuración CMOS; la cual presenta algunas ventajas como es el reducido consumo de energía, lo cual permite altísimas densidades de integración. Pero, por otro lado este consumo aumenta con la frecuencia de trabajo, pues es necesaria mayor cantidad de energía para cumplimentar con la carga y descarga de las capacidades parásitas, más cantidad de veces por segundo. Para salvar este importante inconveniente Altera propone para sus circuitos dos modos de funcionamiento:

- **Zero-Standby:** Modo de trabajo de consumo extra reducido de potencia. Esto se logra a través de no consumir cuando el circuito está en reposo. Para poder lograrlo cada entrada posee un detector de transición, de tal forma que mientras no exista actividad en la misma, es decir no se detecte información válida, el consumo se reduce a micro ó nanoamperes. Pero cuando se establece una actividad válida, y el circuito detecta una transición, el circuito se activa ó “despierta” y se pone a funcionar normalmente. A través de este artilugio se logra reducir bastante el consumo; pero el tiempo de propagación de entrada/salida se incrementa en un 30% a un 40%. Este modo puede determinarse en el mismo momento de la programación.
- **Alta velocidad:** Este modo permite trabajar en la máxima velocidad que permite la arquitectura del dispositivo y la tecnología constructiva. Es el que aparece por defecto (“default”) al programar el circuito, pues al modo anterior hay que activarlo expresamente. Altera, le llama específicamente a este modo “Turbo”.

Una característica importante a tener en cuenta son las protecciones que posee el EPLD para ciertas aplicaciones de alta performance, como lo es el hecho que se puedan sacar y colocar las placas en funcionamiento, es decir energizadas. Muy usado en el caso de sistemas de funcionamiento ininterrumpido; como puede ser el caso de computadores servidores de red. Podemos decir que los circuitos bipolares aceptan bastante bien este procedimiento, pero para los dispositivos CMOS es “mortal”;

ya que no aceptan bajo ningún motivo señal en sus entradas si dicho circuito no está alimentado.

Por lo tanto si el EPLD no aclara específicamente que se pueda realizar tal operación, recomendamos realizar el circuito mostrado en la Fig N° 4.2 en toda conexión al exterior de la placa correspondiente.



**Fig. N° 4.2: Protección para conexión exterior de un EPLD CMOS**

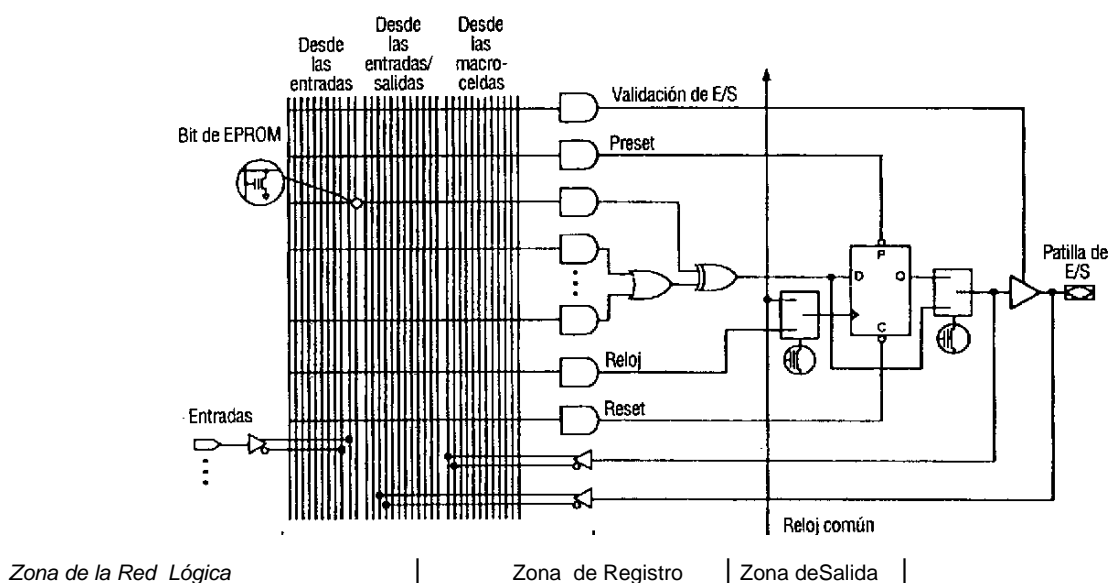
Puesto que se trata de circuitos programables y con la posibilidad de ser borrados por agentes exteriores, como radiaciones, ya sea aplicada para tal fin ó no. Resulta de especial interés el tiempo de retención de la información. Cuando tales circuitos se utilizan en las fases de desarrollo para la implementación de prototipos, tal característica no es muy importante. Pero para productos finales es de vital importancia, puesto que si varía la programación es como si desapareciera el circuito. Altera establece pruebas de funcionamiento en condiciones extremas, como 28 días a 250 °C, llamados *test de burn-in*, que equivale a un funcionamiento a 70°C continuado durante 100000 años sin pérdida de programación.

Como las EPLD son borrables por ultravioleta, su encapsulado debe incluir la ventana de exposición. Esto implica que tal encapsulado tenga que ser cerámico, pues es imposible de realizar en plástico. Sin embargo, para abaratar costos, ha aparecido una versión de EPLD sin ventana, llamada OTP, ó One Time Programmable, que se destina a los productos ya desarrollados y que no necesitarán ser borrados.

Al igual que todos los circuitos programables estudiados hasta ahora, los EPLD tienen una celda de seguridad, que una vez programada no permite la lectura interna del dispositivo; impidiendo toda copia del mismo. Esta celda no se puede borrar aisladamente, sino que se debe borrar toda la programación del EPLD.

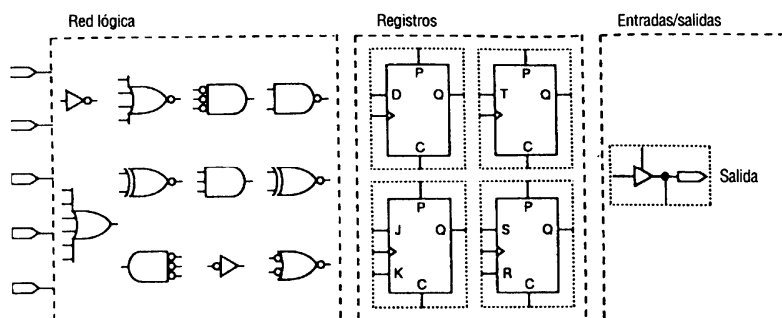
## Macroelda de EPLD

Al igual que en las GAL y PAL CMOS, en las EPLD también existen las celdas programables. A continuación vemos una macroelda típica.



**Fig. N° 4.3: Macroelda típica de un EPLD**

En el esquema presentado en la Fig. 4.1 podemos ver tres zonas bien diferenciadas: Zona de la red lógica, zona del registro y de salida. Estas Zonas pueden tomar diferentes configuraciones según la programación realizada, llegando a obtener estructuras genéricas como las mostradas en la fig. 4.3 .



**Fig. Nº 4.4: Macrocelda EPLD: Posible unidades lógicas programables**

Podemos entonces analizar estas tres zonas de la siguiente forma:

1. **Zona de la red lógica:** Evidentemente una parte importante de ella está constituida por la matriz de conexiones programables; y en la cual se presenta la misma estructura de uniones vistas en las PAL clásicas. Se observa una particularidad, y es que las señales utilizables están agrupadas en tres bloques según el origen de dicha señal, a saber: señales de entrada propiamente dichas provenientes de las patas dedicadas a tal fin; señales de entrada/salida y señales dirigidas desde otras macroceldas. Como vemos las posibilidades de conexión son muy grandes y diversas.

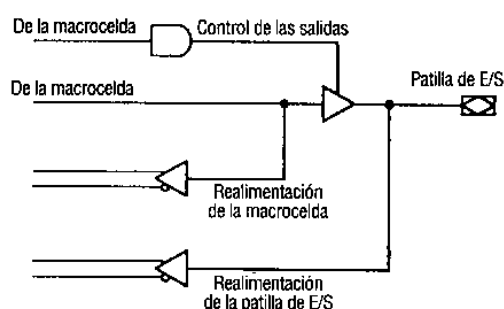
2. **Zona de registros:** Vemos aquí un biestable genérico, y que según la programación puede tomar la característica de flip-flop D, T, R-S e incluso J-K. Esta zona puede eliminarse por programación del multiplexor adecuado como se verá al analizar detalladamente la misma.

El reloj puede provenir de la zona de la red lógica, y en cuyo caso puede ser activo con flanco ascendente ó descendente según la programación. También puede venir de la línea común a todos los biestables y en este caso sólo se activará con el frente de subida.. En este último caso se puede garantizar exactamente los tiempos de propagación y sincronismo. También aquí se logra esto por programación específica del multiplexor adecuado, como puede verse en la figura de la macrocelda típica.

La dato aplicado al biestable ó disponible a la salida de una celda combinacional (si dicho registro es deshabilitado) es una función lógica de variables proveniente de la red lógica. Acá aparece, como caso especial, una OR-exclusiva como confirmación ó completación de la señal obtenida.

El biestable puede ser “preseteado” ó “reseteado” a voluntad como función lógica desde la matriz programable.

3. **Zona de salida:** La señal de salida se maneja a través de un buffer de tercer estado por una señal originada en la zona programable; también es realimentada hacia la misma matriz con el objeto de permitir la realización de lazos. Esto también posibilita el envío hacia otras macroceldas. Esta pata puede ser de



entrada/salida como se ve en la fig. N° 4.5 .  
No ocurre lo mismo con las patas de  
entradas dedicadas.

Fig. N° 4.5: Detalle de la zona de salida de un EPLD clásico.

## Tipos de EPLD

Desde la introducción de las primeras EPLD en el mercado, Altera se ha esforzado por cumplir con las premisas que se propuso, y esto se ve reflejado en la gran variedad de productos para usos generales que posee actualmente: PLD Clásicos (Classic™ PLD), MAX 3000, MAX 5000, MAX 7000, MAX 9000, FLEX 6000, FLEX 8000, FLEX 10K, Y APEX 20K. La familia clásica y MAX ó Multiple Array Matrix utilizan la arquitectura de sumatoria de productos con una matriz Y-Programable y una O-Fija. La familia FLEX ó Flexible Logic Element Matrix utiliza la arquitectura de generador de funciones combinacionales ó LUT (Look-up table) junto con elementos secuenciales. Y finalmente la familia APEX ó Advanced Programmable Logic Matrix combina las dos arquitecturas indicadas.

En la fig. 4.6 se esquematiza una síntesis de la evolución de la arquitectura de los EPLD de Altera, para mantener las premisas especificadas con anterioridad.

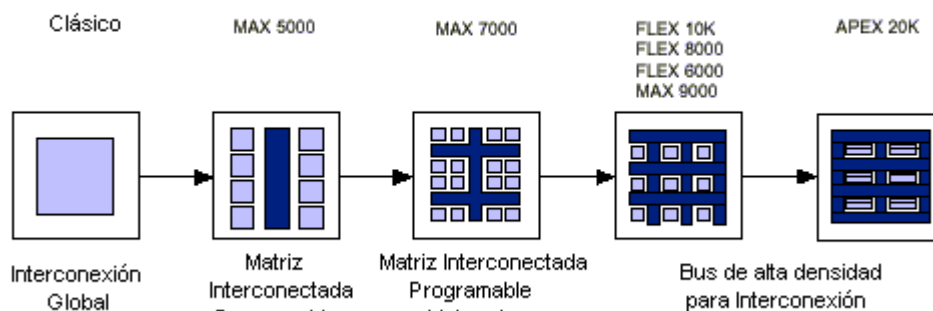
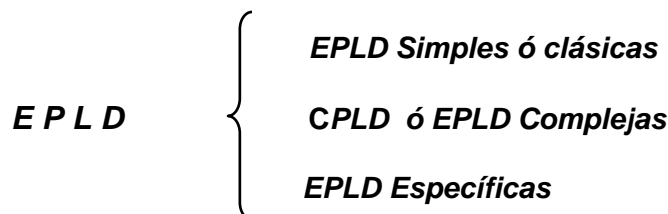


Fig. N° 4.6: Evolución de la arquitectura de Altera.

De lo comentado podemos hacer una rápida agrupación de familias para obtener la clasificación que vemos a continuación, y que desarrollaremos luego:



### 1) EPLD Simples ó clásicas

Se trata de los primeros circuitos programables de este tipo, incorporados inicialmente por Altera, aunque luego fueron también fabricados por otras empresas.

Su estructura interna está compuesta por arreglos lógicos interconectados en forma global; agrupándose en macroceldas que toman señal lógica desde las entradas dedicadas, entradas/salidas, y realimentaciones varias en forma directa y complementada, conformando una serie de buses adecuados para la distribución de esta información. Las mencionadas macroceldas están conformadas por redes de suma de productos y registros programables.

Son del tipo UV-EPROM, es decir grabables eléctricamente y borrables por ultravioleta. En los encapsulados con ventana de borrado pueden ser el 100 % testeados en fábrica.

Presentan densidades de integración de alrededor de 300 a 900 compuertas utilizables. Se los fabrica en circuitos integrados de 20 a 68 patas, presentados en una gran variedad de encapsulados, tales como: CerDIP (Ceramic Dual In-line Package), PDIP (Plastic Dual In-Line Package), PLCC (Plastic JLead Cip Carrier), JLCC (Ceramic JLead Cip Carrier), PGA (Pin-Grid Array), y SOIC (Small Out-Line Integrated Circuit). ( Ver detalle de los tipos de encapsulado en Apéndice N° 1.)

Esta familia ofrece una solución de bajo costo para aplicaciones de propósitos generales de baja densidad. Su uso principal es reemplazar lógica cableada realizada con algunos dispositivos PAL e incluso algunas decenas de circuitos integrados de compuertas TTL clásicas. Solo posibilita el trabajo en modo “zero stanby”, lo que la hace ideal para aplicaciones de bajo consumo.

Conforman esta familia la serie EPXXXX, cuyos componentes son bastante parecidos unos a otros, pero con variaciones en el número de macroceldas, número de entradas salidas, y el tiempo de propagación típico y cuyas principales características vemos en la tabla N° 4.1 y figura comparativa N° 4.5.

Características	EP610 EP610I	EP910 EP910I	EP1810 EP1830
Compuertas utilizables	300	450	900
Macrocelas	16	24	48
Máximo número de patas	22	38	64
t <sub>PD</sub> (nseg)	10	12	20
f <sub>CNT</sub> (Mhz)	100	76.9	50

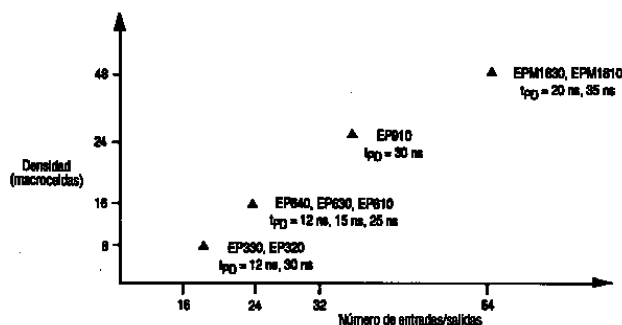
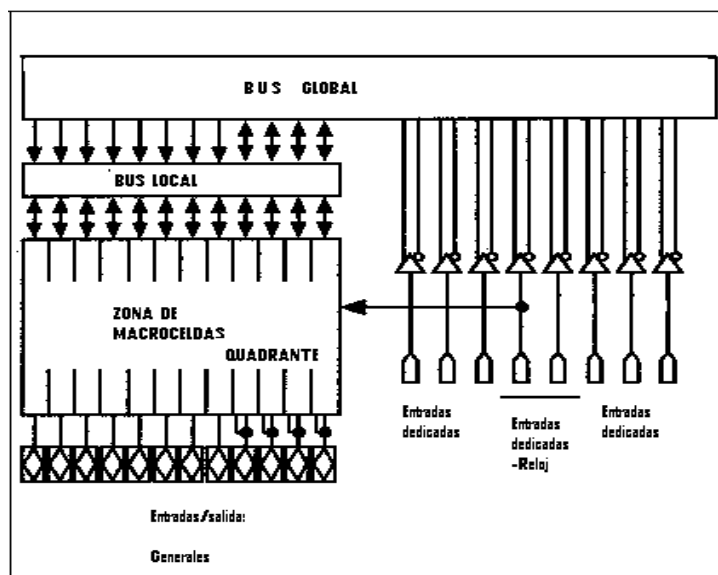


Tabla N° 4.1: Principales características comparativas de la serie EP - Fig N° 4.5:

## Arquitectura de la familia de EPLD clásica:

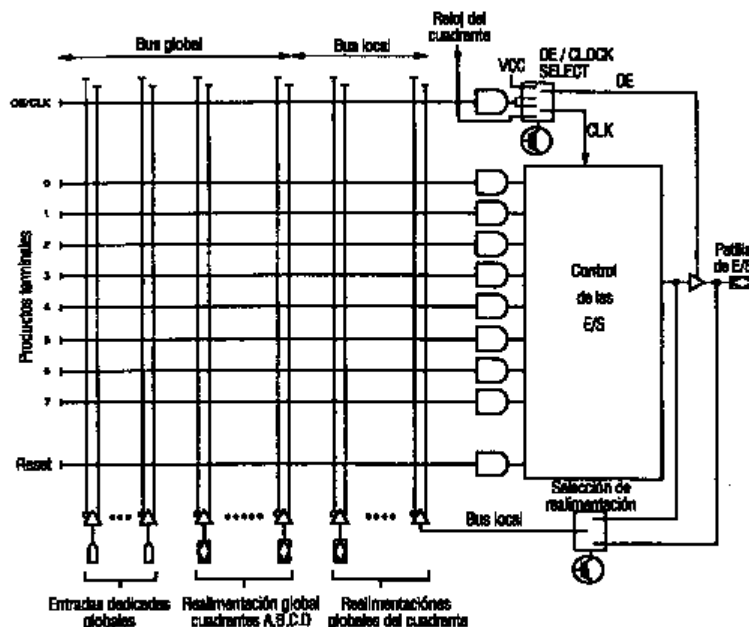
En la fig. N° 4.7 puede verse un diagrama en blocks general de la estructura interna de esta familia, desarrollándose luego una breve descripción de cada uno de dichos elementos.



**Fig N° 4.7: Arquitectura interna general de un EPLD clásico.**

En la fig anterior podemos ver los siguientes elementos básicos:

- **Entradas dedicadas:** Señales lógicas que ingresan al dispositivo directamente a la matriz programable.
- **Entradas/salidas generales:** Señales lógicas que con característica bidireccional identifican a cada macrocelda en particular.
- **Bus Global:** Bus que incorpora todas las señales lógicas de entradas dedicadas, entradas/salidas, realimentación de macroceldas y reloj en su forma acertada ó complementada.
- **Bus Local:** Bus que incorpora algunas señales lógicas de entradas dedicadas, entradas/salidas, realimentación de macroceldas y reloj en su forma acertada ó complementada.
- **Relojes:** Entradas específicas de reloj
- **Cuadrantes:** Agrupación de macroceldas excitadas con un reloj común. Puede incluir macroceldas de dos tipos: *Locales* y *Globales*. Esta diferencia se debe sólo a las opciones de conectividad: *Las locales* sólo pueden vincularse en forma bidireccional con las macroceldas que integran el cuadrante; sin embargo *las globales* lo pueden hacer con todas las del dispositivo. Como vemos en la fig. N° 4.8 las macroceldas locales tienen acceso a las entradas dedicadas, a los retornos de todas las macroceldas del cuadrante al cual pertenecen, y a los retornos globales de algunas macroceldas de los otros cuadrantes. Sus señales de salida sólo las puede volcar en el bus local.



**Fig N° 4.8:**

**Macrocelas Locales: Posibilidades de conexión.**

Las macroceldas globales tienen acceso a los mismos recursos que los indicados, pero además pueden reinyectar su señal en el bus local, como se ve en la fig. N° 4.9.

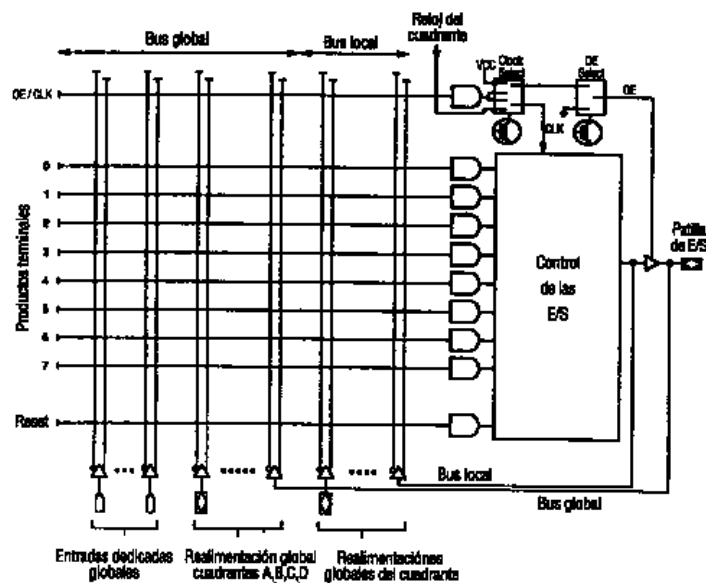


Fig Nº 4.8: Macroceldas Locales: Posibilidades de conexión

Cada componente de la familia incorpora algunas ó todas las características enunciadas, como se ve en la siguiente tabla y se verá luego cuando se describan algunos circuitos de la familia clásica:

	EP320 EP330	EP 610 EP630	EP910 EP910I	EP1810 EP1830
Entradas dedicadas	10*	4	12	16*
Entradas/Salidas	8	16	24	48
Relojes	1*	2	2	4*
Bus Global	1	1	1	1
Bus Local	0	0	0	4
Cuadrantes	1	2	2	4
Macroceldas por cuadrante	8	8	12	12

\* Los relojes están multiplexados con una entrada dedicada cada uno.

Las macroceldas clásicas incorporan los componentes mostrados en la fig. Nº 4.9 :

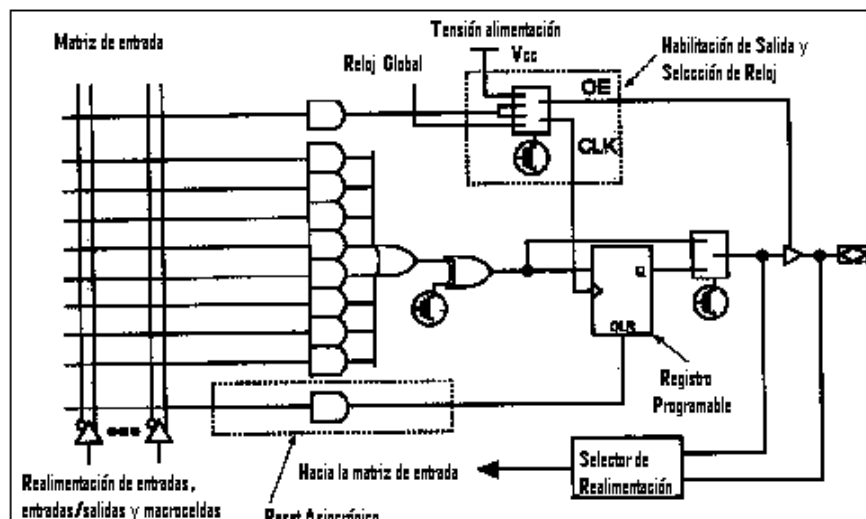
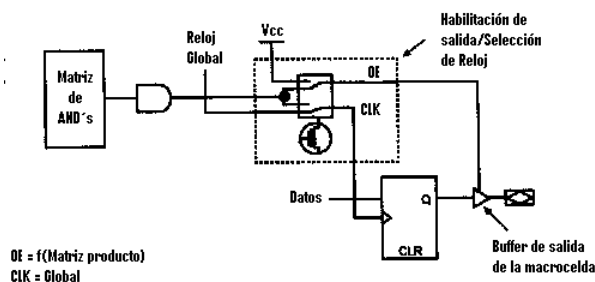


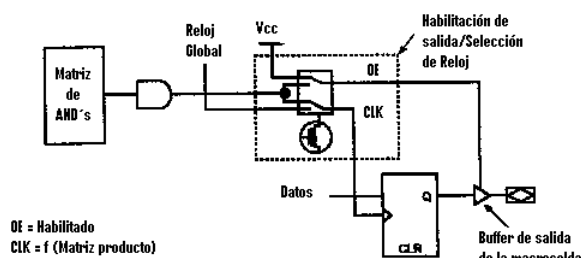
Fig. N° 4.9: Macrocelda de la familia clásica de Altera

1. **Redes combinacionales de suma de productos.** Estos arreglos están estructurados como las clásicas PAL con una matriz Y-programable y una matriz O-fija. Cada macrocelda incorpora una suma lógica de ocho (8) términos producto programables. La salida de la compuerta OR de 8 entradas alimenta una OR-Exclusiva. La otra entrada de esta compuerta incluye un bit programable, que permite usarse como inversor controlado; obteniendo así señales activas en bajo ó en alto. Además se incorporan dos (2) productos lógicos adicionales, uno para la implementación de un reset asíncrono del registro programable y el otro para dos funciones diferentes: habilitación de salida y generación de reloj.  
La información para estas macroceldas proviene de señales acertadas ó complementadas de entradas dedicadas, realimentación de entradas-salidas, ó realimentación de otras macroceldas. Como puede verse en el esquema general de estos dispositivos, las entradas dedicadas entran directamente al bus global y se distribuyen directamente a todas las macroceldas. Las otras señales tienen aplicaciones más específicas.
2. **Registros programables:** Para implementar funciones secuenciales, cada macrocelda posee un registro que puede programarse individualmente como biestable SR, D, T e inclusive JK. También puede bypassarse para funciones exclusivamente combinacionales. Durante la programación, el software selecciona el funcionamiento más conveniente para cada registro.  
Los biestables son preseleccionados automáticamente cuando se le da alimentación a estos dispositivos. También tienen un reset funcional a través de una compuerta AND dedicada para tal fin desde la matriz de entrada. Cada reloj de registro puede ser excitado en forma global ó realimentaciones adecuadas a través de la matriz de entrada. La señal de salida de estos registros pueden dirigirse directamente al exterior del circuito integrado ó realimentados de varias formas realizando las programaciones adecuadas.
3. **Selección de reloj y habilitación de salida:** Los EPLD de esta familia tienen dos modos de operación, a través del adecuado uso del selector OE/CLK ( Fig. N° 4.9). Este es controlado por la programación de un único bit y puede ser configurado en forma individual para cada macrocelda.

**Modo 0:** El buffer de alta impedancia (tri-state) de salida es controlado por uno de los términos producto. Y en este caso, si la habilitación está en alto; el buffer está habilitado, pero por el contrario si esta señal está en nivel cero, la salida toma una alta impedancia (HiZ). En este modo el biestable es excitado por un reloj global de entrada.



**Modo 1:** En este modo el buffer de salida está siempre habilitado y el reloj del biestable correspondiente es excitado por el resultado de un producto lógico desde la matriz de entrada. Por lo tanto este modo permite manejar el reloj desde cualquier señal proveniente de la mencionada matriz. Y como además estas variables pueden ser acertadas ó complementadas podemos lograr una programación con el flanco de subida ó de bajada.





4. **Selección de realimentación:** Esta característica es controlado por un multiplexor específico, llamado precisamente “ Feedback multiplexer” . Se puede realimentar la salida específica de la macrocelda desde la salida Q del biestable ó la pata de entrada/salida correspondiente a la misma macrocelda; pero luego del buffer controlado. Además se puede realimentar al bus global ó al bus local adecuadamente; y en dispositivos más avanzados, a ambos a la vez con un doble multiplexor de realimentación.

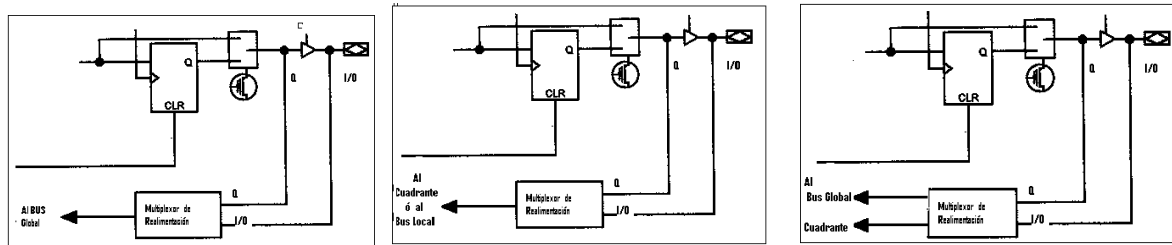


Fig. Nº 4.10: Configuraciones clásicas del multiplexor de realimentación

### Estrategias de seguridad:

Estos dispositivos clásicos, poseen un bit de seguridad que controla el acceso a todos los datos programados en el dispositivo. Cuando este bit es programado, el diseño implementado en tal circuito no puede ser copiado ó recuperado. Esta característica le confiere un alto grado de seguridad, ya que la información almacenada es invisible. El bit de seguridad que controla esta función y otros datos de configuración solo puede resetearse cuando el dispositivo es borrado.

### Tiempos de propagación:

Como hay muchas entradas y numerosas formas de realimentación, el tiempo final dependerá del camino realmente seguido por la señal lógica. Por ello, cuando es necesario conocer y ajustar al detalle el valor de los mismos, podemos decir, que existen numerosas herramientas de análisis de retardos y simuladores que nos permiten obtener detalles de los mismos. Además, cada fabricante provee *modelos típicos de temporización*, que desglosando los mismos en bloques característicos permiten determinar los retardos propios de cada camino recorrido por la información. Y de esta forma se puede obtener al detalle cada tiempo de retardo sumando los valores parciales. Esto puede verse en la fig. 4.11 junto a un detalle general de las formas de onda de estos dispositivos clásicos.

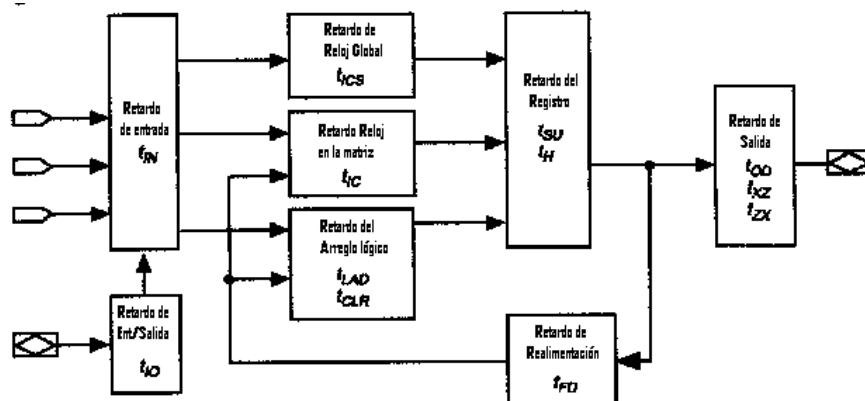
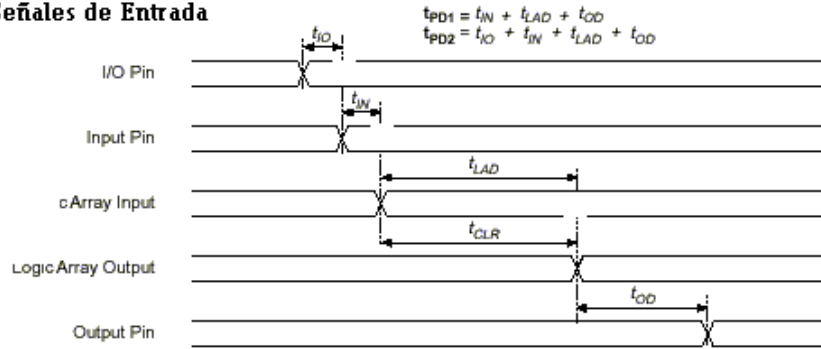
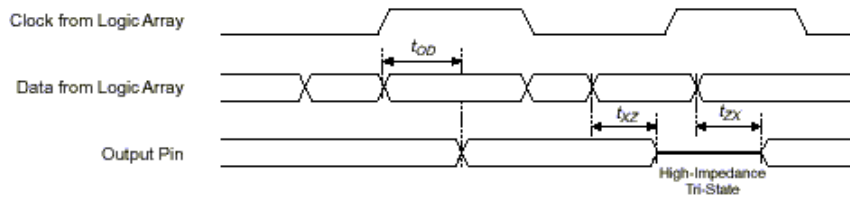


Fig. Nº 4.11: Modelo de temporización de la familia EPLD Clásica de Altera.

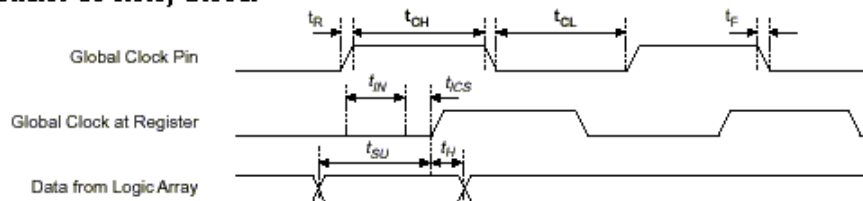
### Señales de Entrada



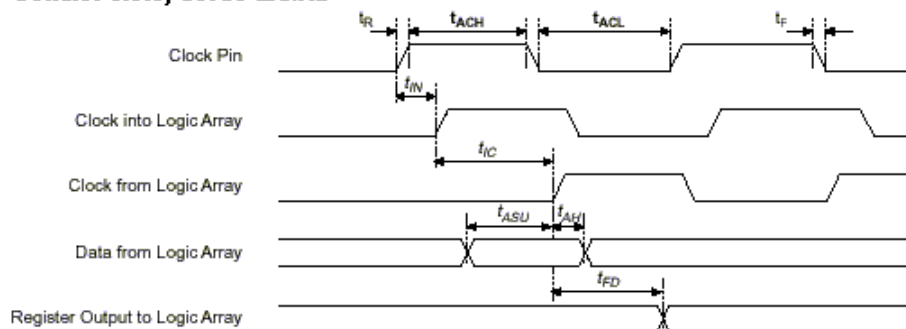
### Señales de Salida



### Señales de Reloj Global



### Señales Reloj desde Matriz

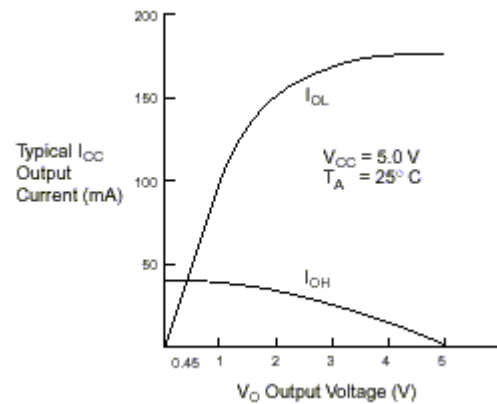
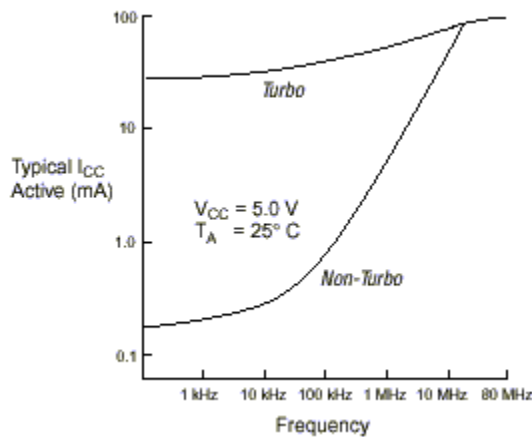


### Condiciones de Operación:

Como en todo circuito electrónico existen condiciones eléctricas básicas de operación a tener en cuenta; de las cuales las más importantes son:

- $V_{CC}$  : Tensión de alimentación.
- $I_{CC}$  : Corriente de alimentación.
- $V_{IH}$  : Tensión de entrada en nivel alto.
- $V_{IL}$  : Tensión de entrada en nivel bajo.
- $V_{OH}$  : Tensión de salida en nivel alto.
- $V_{OL}$  : Tensión de salida en nivel bajo.
- $T_A$  : Temperatura ambiente.

Y básicamente es importante apreciar la variación de la corriente de alimentación ( ó consumo) con la frecuencia y con el valor de tensión de salida, y cuya forma general podemos apreciar en las siguientes curvas:



Curva típica  $I_{CC} = F (v_o)$

### Comentarios sobre programabilidad:

Si bien el emplazamiento de las funciones lógicas en el circuito, la realiza el software del sistema de desarrollo, es necesario tener bien presente que la eficiencia del mismo está directamente relacionado a la distribución adecuada de los recursos. Lo expuesto resulta de vital importancia, por cuanto con la experiencia de desarrollo en este tipo de circuitos, se comprueba que es imposible alcanzar una utilización del 100% de los recursos internos, debido a la falta de lugar para hacer algunas interconexiones. Hay que ser conscientes que un porcentaje de ocupación del 60 a 80 % resulta ser bueno a excelente. Ahora por otro lado si el mismo resultara del 20 al 30 % deberíamos preocuparnos, por cuanto nos indica de una mala elección del EPLD. Siempre es necesario realizar este tipo de evaluaciones, por cuanto es mejor utilizar varios EPLD pequeños, del tipo EP330, utilizados al 80 % que un solo EP 1810 utilizado al 30 %. Por lo general, los compiladores de los programas de desarrollo, indican en forma muy precisa el porcentaje de ocupación de los recursos internos de los diferentes circuitos EPLD.

**Principales circuitos de la familia clásica:** Para tener una idea general y poder realizar un análisis comparativo presentamos los circuitos más representativos de esta familia clásica de la empresa Altera:

**a) EP330 :** Fue uno de los primeros circuitos EPLD,

- PLD de media performance, 8 macroceldas del tipo clásica.
- Velocidad combinacional mejor que 12 nseg.
- Frecuencias de contadores a 100 Mhz.
- Manejo de datos continuos a 125 Mhz.
- Arquitectura de entrada/salida hasta 18 entradas, 8 salidas ó 1 reloj.
- Macrocelas programables individualmente para operación combinacional ó secuencial con biestables D, T, JK ó SR.
- Disponible en los siguientes encapsulados: (Ver fig. N° 4.10)
  - SOIC 20 patas (Small Outline Integrated Circuit)
  - PDIP 20 patas (Plastic Dual In Line Package).
  - PLCC 20 patas (Plastic J-Lead Chip Carrier).

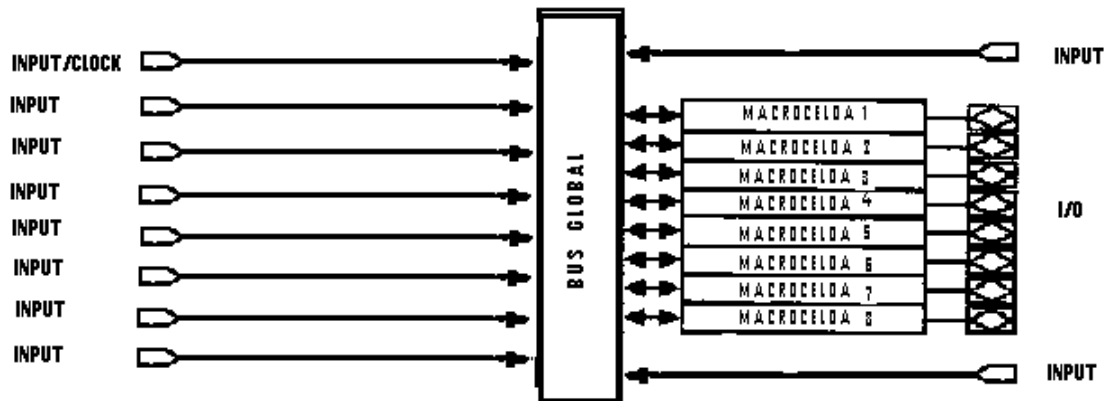


Fig. N° 4.12 : Arquitectura interna del EP 330

b) EP610 :

- PLD de alta performance, 16 macroceldas del tipo clásica.
- Velocidad combinacional mejor que 10 nseg.
- Frecuencias de contadores a 100 Mhz.
- Manejo de datos continuos a 125 Mhz.
- Arquitectura de entrada/salida hasta 20 entradas o 16 salidas y 2 relojes.
- Reloj programable para temporización independiente de todos los biestables.
- Macroceldas programables individualmente para operación combinacional ó secuencial con biestables D, T, JK ó SR.
- Disponible en los siguientes encapsulados: (Ver fig. N° 4.10)
  - SOIC 24 patas (Small Outline Integrated Circuit)
  - CerDIP y PDIP 24 patas (Ceramic and Plastic Dual In Line Package).
  - PLCC 28 patas (Plastic J-Lead Chip Carrier).

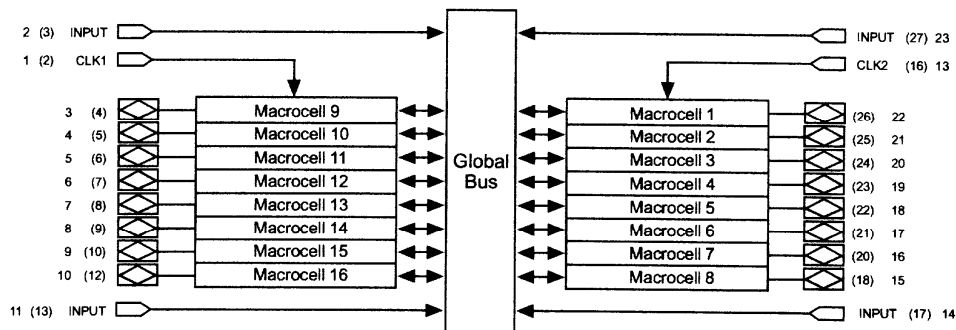


Fig. N° 4.13 : Arquitectura interna del EP 610

c) EP910 :

- PLD de alta performance, 24 macroceldas del tipo clásica.
- Velocidad combinacional mejor que 12 nseg.
- Frecuencias de contadores a 76,9 Mhz.
- Manejo de datos continuos a 125 Mhz.
- Arquitectura de entrada/salida programable hasta 36 entradas o 24 salidas.
- Reloj programable para temporización independiente de todos los biestables.
- Macroceldas programables individualmente para operación combinacional ó secuencial con biestables D, T, JK ó SR.
- Disponible en los siguientes encapsulados: (Ver fig. N° 4.10)
  - CerDIP y PDIP 40 patas (Ceramic and Plastic Dual In Line Package).
  - PLCC 40 patas (Plastic J-Lead Chip Carrier).

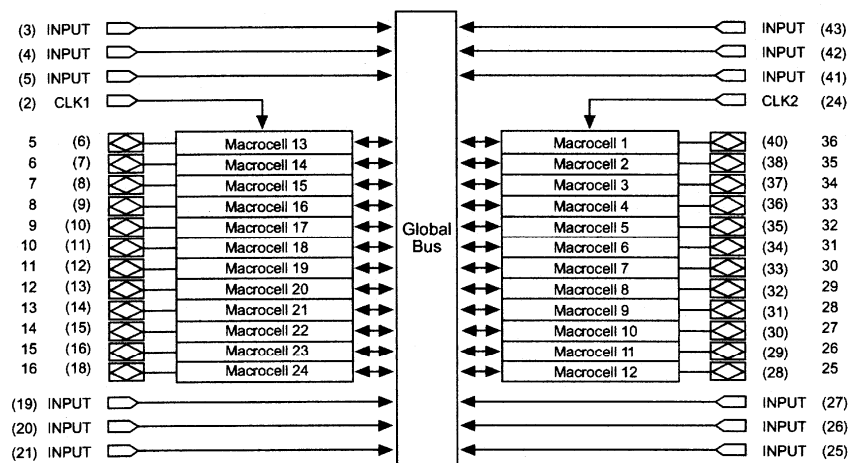


Fig. N° 4.14 : Arquitectura interna del EP 910

d) EP1810 :

- PLD de alta performance, 48 macroceldas del tipo clásica.
- Velocidad combinacional mejor que 20 nseg.
- Frecuencias de contadores a 50 Mhz.
- Manejo de datos continuos a 62,5 Mhz.
- Arquitectura de entrada/salida programable hasta 24 entradas o 48 salidas.
- Reloj programable para temporización independiente de todos los biestables.
- Macrocelas programables individualmente para operación combinacional ó secuencial con biestables D, T, JK ó SR.
- Disponible en los siguientes encapsulados: (Ver fig. N° 4.10)
  - PGA 68 patas (Ceramic Pin Grid Array).
  - PLCC 68 patas (Plastic J-Lead Chip Carrier).

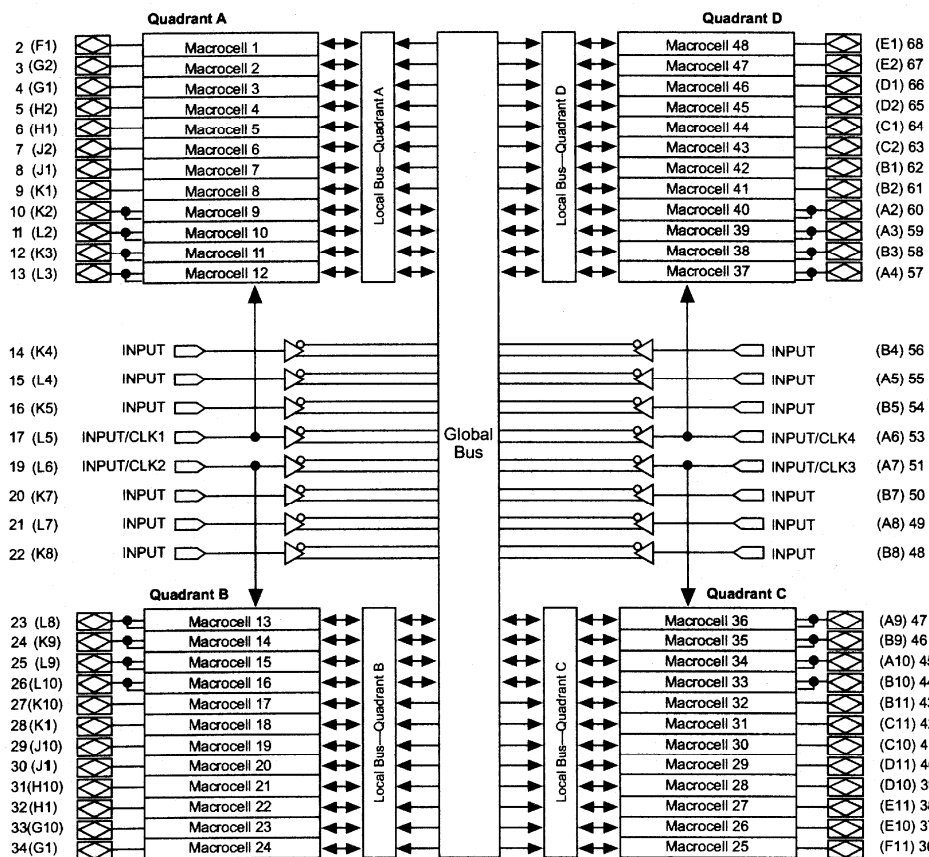


Fig. N° 4.15 : Arquitectura interna del EP 1810

## **2).EPLD Complejas ó CPLD (Complex Programmable Logic Device):**

Se trata de circuitos programables complejos, fundamentalmente caracterizados por una nueva etapa en la densidad de integración, pues como hemos mencionado mientras un EP1810 de la serie clásica tiene alrededor 1000 compuertas lógicas utilizables, en uno de la serie compleja puede superar ampliamente las 20000. Este incremento en la densidad de integración, trae aparejado cambios y mejoras de arquitectura y tecnología.

Si bien se trata de una serie compleja, su uso está orientado a aplicaciones de propósitos generales, como la serie EP; pero con mayor densidad de integración. Está orientada a reemplazar lógica realizada con varias decenas de dispositivos PAL ó cientos de componentes TTL normales. Vienen encapsulados en circuitos Integrados de 28 a 780 patas, en DIP , *flat-pack*, PLCC, PGA, tc. con densidades de integración de 600 a 2.500.000 compuertas efectivamente utilizables. Las velocidades de trabajo típicas están en el orden de 3 a 4 nseg, y con versiones cuyo funcionamiento puede llegar a los 500 Mhz.

Existen en la actualidad 3 familias, en las cuales se han realizado mejoras estructurales para lograr una mayor densidad de interconexiones que las EPLD con interconexión clásico, y tenemos:

- a) Familia MAX ó de la serie EPMXXXX.ó *Múltiple Array matrix*., Actualmente Altera posee varios productos, como las MAX 5000, MAX 7000 y MAX9000.
- b) Familia FLEX ó *Flexible Logic Element Matrix*. Tenemos las subfamilias FLEX 6000 y FLEX 8000.
- c) Familia APEX ó *Advanced Programmable Element Matrix*. Tenemos las subfamilias APEX 10K y APEX 20K. ( año 2000)

### **Familia MAX ó de la serie EPMXXXX :**

Por ahora estudiaremos sólo esta familia dentro de las CPLD, la cual incorpora una innovación en la arquitectura interna llamada MAX ó *Múltiple Array matrix*. Con lo cual se pretende mejorar la estructura de interconexiones; ya que que ha medida que los PLD se hacen más complejos este resulta ser el problema puntual en busca de diseños efectivos y con buena performance. Sin embargo, a pesar de la introducción de estas innovaciones en su arquitectura, esta subfamilia sigue usando la arquitectura de producto de términos lógicos de matriz Y programable y matriz O fija.

En los circuitos de la serie MAX siguen existiendo las macroceldas, presentadas y ampliamente usadas en la serie clásica, pero ahora integrada en un paquete lógico más complejo llamado LAB ó Logical Array Block. Tal como puede verse en la fig. 4.16 este bloque lógico no es más ni menos que una ampliación y reagrupación de los componentes de una macrocelda típica de los EPLD clásicos de Altera.

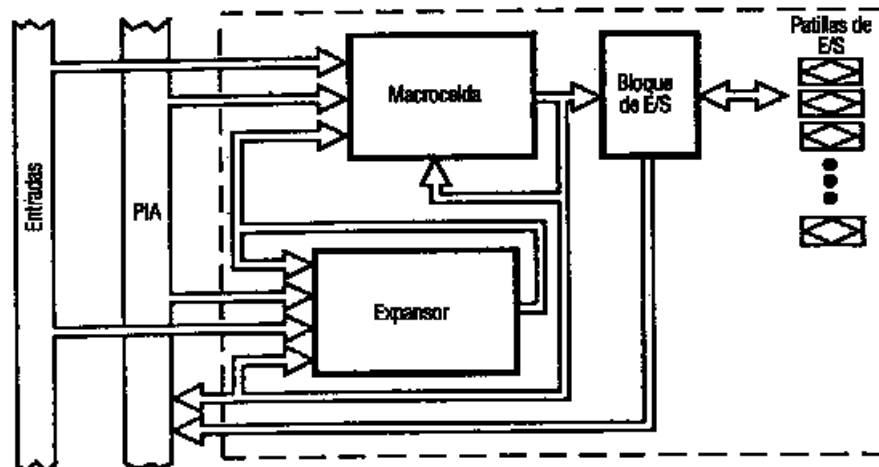
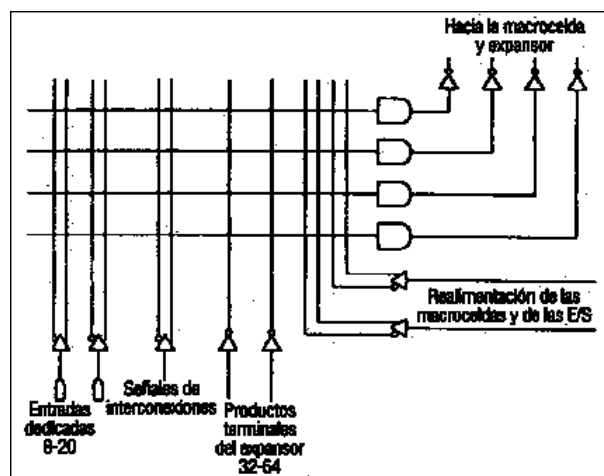


Fig. Nº 4.16: LAB ó Logical Array Block: estructura interna.

De acuerdo a lo comentado podemos ver los siguientes tres componentes:

- **Macrocelda:** Componente lógico que contiene específicamente los elementos programables secuenciales( Fig. 4.3 y 4.7 ).
- **Expansor:** Bloque de lógica combinatorial clásica. Presentado esquemáticamente en la fig Nº 4.17, muestra que realiza un reagrupamiento lógico de señales, según varias funciones combinatoriales, con objetivo claro de tratar de disminuir al máximo la cantidad de interconexiones globales necesarias.
- **Bloque E/S:** Interfaz adecuada con el exterior del EPLD ( fig. 4.5 ).



Fig, Nº 4.17: LAB: Principio del expansor

En líneas generales podemos decir que se maneja dos tipos de información y según esto es el acceso lógico que tienen, a saber:

- **Entradas dedicadas:** Sólo tienen acceso a la macrocelda y al expansor correspondiente a través de la matriz programable clásica.
- **Entradas/Salidas e información interna:** Esta última, generada por las salidas del expansor y de la macrocelda ingresan a una matriz programable nueva y diferente, llamada PIA ó Programmable Interconnect Array. Esta zona permite en su realimentación ser conectada nuevamente a la entrada de las macroceldas y los expansores, otorgando una flexibilidad de programación muy poderosa.

A título de ejemplo, vemos en la fig N° 4.18 la estructura interna de una EPM 5192, dónde se puede apreciar 12 LAB que agrupan un total de 192 macroceldas, (Recordemos que el EP1810 de la serie clásica sólo tenía 48 macroceldas) conectadas todas a una PIA ó zona de interconexión interna gigantesca. Y tal como dijimos, en la parte superior de las LAB se puede apreciar que las entradas dedicadas sólo llegan directamente a ellas.

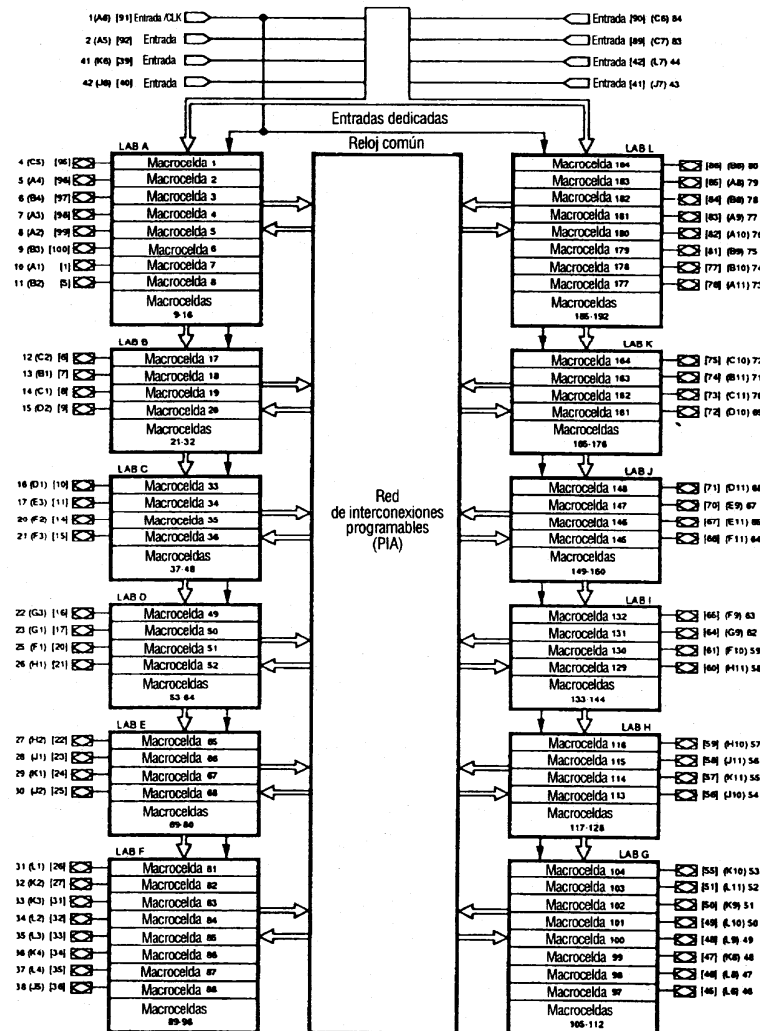


Fig. N° 4.18: EPM 5192: Diagrama interno.

### 3) EPLD Específicas ó SEPLD (Specific Erasable Programmable Logic Device):

Se trata de circuitos programables orientados a una determinada aplicación, y por lo tanto se ha introducido una mejora interna orientada a tal fin.

**Uso:** Propósitos especiales.

**Tipo:** a) Familia de serie EPB X Orientados al Bus. Incorpora toda la lógica necesaria para interfacear con el bus adecuado. Nació para el Micro Channel Bus.

b) Familia de serie EPS X Orientados al diseño de secuenciadores.

**Función:** Reemplazar lógica de uso general, orientadola a circuitos específicos.

**Densidad de Integración:** Circuitos Integrados de 20 a 68 patas, en DIP , *flat-pack*, PLCC, PGA, etc. con densidades de integración de 300 a 900 compuertas.

**Velocidades de trabajo:** Bajas, pues en general no es el objetivo fundamental.



## 5 FPGA

El término FPGA, corresponde a *Field Programmable Gate Array*, y su traducción sería Red de puertas programable en campo, es decir “in situ” ó en el lugar; también podría decirse por el usuario. Este tipo de dispositivo lógico programable constituye el grupo actual con más alta densidad de integración, superando ampliamente el millón de puertas lógicas. Esto justifica el nombre, lo cual nos dice que circuitos tan altamente sofisticados, con extrema densidad de integración, aún siguen siendo programables por el usuario final con medios técnicos y económicos al alcance de cualquier pequeña empresa, ámbito educativo ó laboratorio de desarrollo.

Podemos decir que en la actualidad existen dos tipos básicos de FPGA, y sobre las cuales desarrollaremos el presente capítulo. Podemos decir que el fin es más ó menos el mismo, permitiendo al usuario desarrollar proyectos que involucren la necesidad de redes de compuertas programables muy densas, pero con herramientas y métodos totalmente diferentes. Ellos son:

1. **LCA ó FPGA tipo RAM:** Circuitos programables y borrables eléctricamente.
2. **FPGA de antifusibles:** Circuitos programables eléctricamente, no borrables.

### 5-1 L. C. A. ó FPGA tipo RAM

Estos dispositivos programables fueron introducidos hacia los años 1984-85 con el nombre original de LCA (Logical Cell Array) ó Red de celdas lógicas, por la empresa norteamericana Xilinx.

Esta empresa determinó, luego de una serie de estudios, que los circuitos programables que existían en aquella época tales como PAL, GAL y EPLD no eran usados en su totalidad, y además que se derrochaba una gran cantidad de recursos. Pues, aunque tales circuitos integren una gran cantidad de puertas lógicas, si estas no están utilizadas en la matriz de entrada, pierden su efectividad en el resto del circuito como puede verse en la fig Nº 5.1.1 El hecho principal de este derroche parece ser la estructura fija que une la red de puertas de entrada y la macrocelda correspondiente. Pues, si bien se establece una estructura programable de entrada muy versátil, al bloquearse la misma, se inutiliza todo el resto por la estructura rígida de la salida. Es decir, si la macrocelda no es necesaria para la realización de una función determinada, se desperdicia el bloque completo.

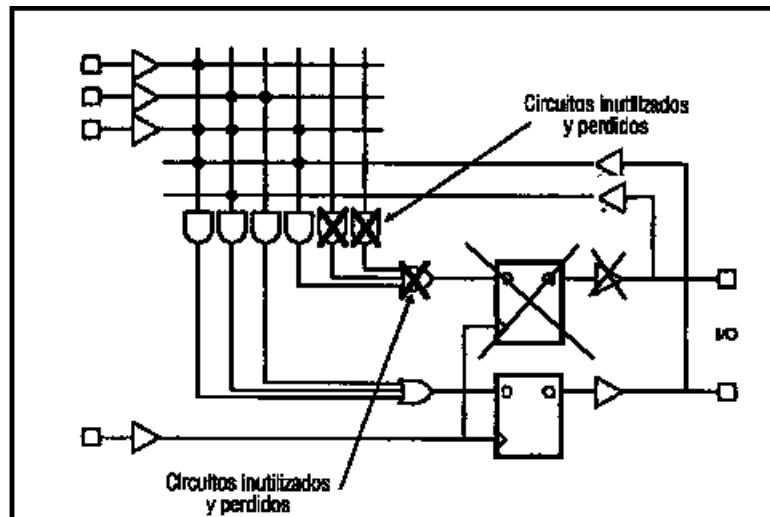


Fig. Nº 5.1.1: Desperdicio de recursos en un PLD clásico.

Se estableció que la solución ideal era la realización de un bloque central conformado por una gran matriz de compuertas elementales y rodeado por bloques altamente sofisticados y programables de entrada/salida, como puede verse en la fig. Nº 5.1.2. Pero inicialmente esta red sólo era posible de realizar a través de la interconexión de redes programables por máscara, las cuales sólo lo son únicamente por el fabricante, sobre la pastilla de silicio. Es decir que no existen ningún tipo de interconexión preestablecida, sino que se realizan solamente las necesarias en la fase de fabricación. De esta forma, se puede lograr un índice de utilización de recursos muy alto; pero con los inconvenientes propios de esta forma de realización, como ser el costo, demora de realización, cantidad mínima etc.

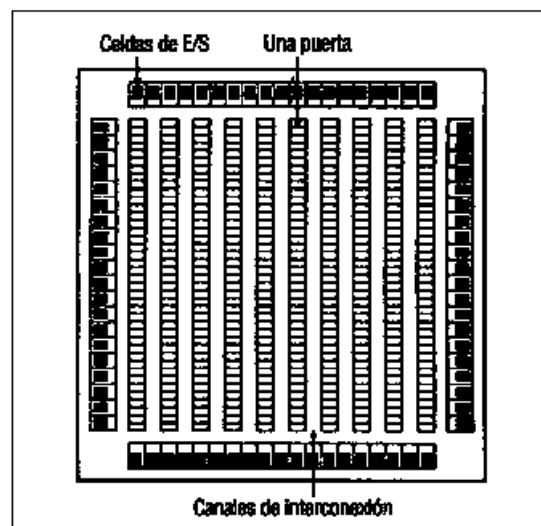


Fig. Nº 5.1.2: Red de puertas programable por máscaras

La estrategia comercial de Xilinx fue introducir en el mercado un producto que resolviera este problema, tratando de conservar la idea de la estructura ideal, pero con la versatilidad de la programación por el usuario final. Así la estructura de las Logical Cell Array combina la programabilidad de las viejas estructuras de PLD, con la nueva idea de las redes de puertas programables por máscaras.

Este objetivo, presenta algunos inconvenientes muy complejos de resolver, como lo es el *gran número de interconexiones* entre compuertas básicas y *como realizarlas* efectivamente para que alcance la calidad de programable. La resolución de estos temas se basa la originalidad innovadora de esta empresa:

- Para reducir la enorme cantidad de interconexiones que conlleva tener puertas elementales, para conformar funciones más complejas, Xilinx reestructura la idea original. La conformación del bloque central deja de ser una red de este tipo, implantando en dicho lugar estructuras más elaboradas, parecidas a las conocidas macroceldas y que denomina CLB ó Configurable Logic Block. La inclusión de estos CLB más complejos permite reducir ampliamente el número de interconexiones. El resto de la pastilla sigue teniendo la misma arquitectura, es decir la periferia la ocupa una gran cantidad de bloques de entrada/salida programables. En la fig. 5.1.3 se ve la estructura general, dónde la arquitectura típica se parece a las redes de compuertas programables por máscaras.

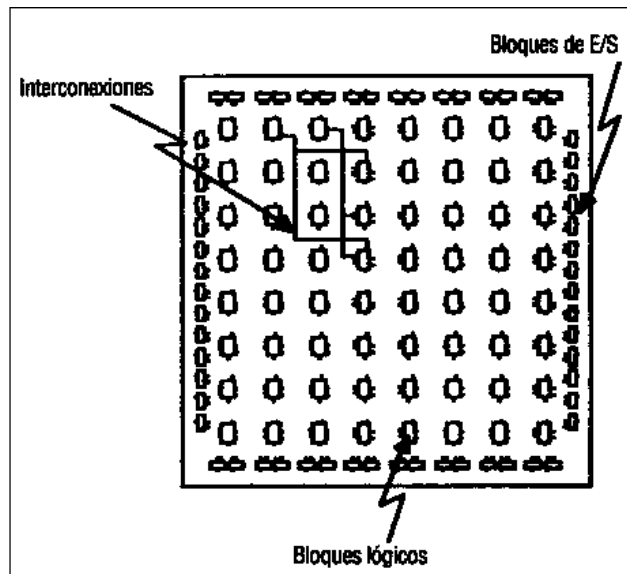


Fig Nº 5.1.3: LCA: Estructura adoptada por Xilinx.

- El problema de la programabilidad del interconexionado es resuelto por Xilinx de una manera muy original y flexible: la LCA está recorrida por una amplia matriz de líneas verticales y horizontales que circundan todos los CLB que ocupan el bloque central. La conexión a dicha red se realiza con transistores CMOS, cuya habilitación es controlada por celdas de memoria RAM. Podemos decir entonces, que "bajo" dicha matriz hay una memoria RAM dónde se programa la configuración de las conexiones a realizar, según se puede ver en la fig. Nº 5.1.4.

Como las memorias RAM pueden escribirse y reescribirse ilimitadamente, de una forma muy rápida, es decir direccionandolás y considerando sólo el tiempo de acceso a la misma, se puede inferir que las LCA heredan dichas características que las hace borrables y programables eléctricamente en el circuito, sin la necesidad de programadores especiales que insumen importantes tiempos de programación. Tampoco necesitan tensiones de programación altas ó especiales, siendo suficiente sólo la alimentación propia del circuito integrado. Por otro lado presentan un problema adicional, pues como se sabe, las RAM pierden su información cuando se les corta la energía. Por lo tanto, cada vez que ocurra esto, se perderá la información de programación almacenada. Para evitar este problema se plantean, y adoptan, según el caso, dos posibles soluciones, a saber:

1. Tratar de mantener alimentada la RAM, aún luego de la pérdida de energía. Para lo cual se incluye una pila de litio que mantiene la LCA alimentada siempre. Como el consumo de la zona de RAM es insignificante, la duración de la misma puede ser de varios años. Inclusive se puede colocar una pila recargable, con su correspondiente cargador, evitándose el problema de posible descarga de la misma.

2. Esta solución utiliza una memoria PROM, para almacenar los datos de programación iniciales ó de configuración, con un circuito secuencial anexo que frente al corte y recuperación de la energía, vuelve a copiar los datos en la RAM reconfigurándola inmediatamente cada vez que se aplica tensión. Esta es quizás la mejor solución, por cuanto posee un circuito sencillo, y no tiene problema de pérdida de información ante una posible descarga de la batería.

Una alternativa de esta solución es que exista pequeño sistema de microprocesador ó microcontrolador que guarde en su memoria la información de configuración de la LCA, y frente a una eventual pérdida de datos, se la transfiera haciendo uso de sus recursos.

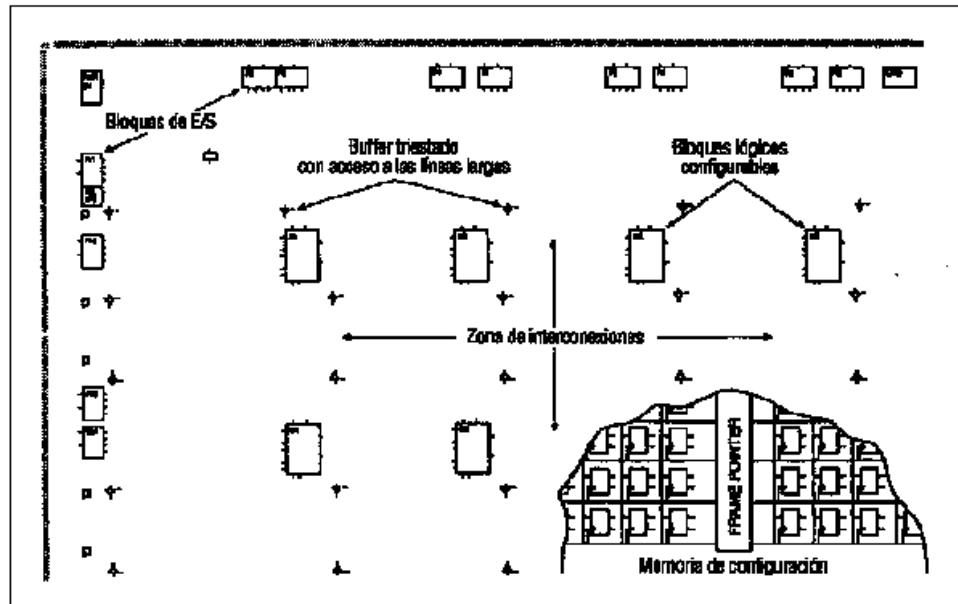


Fig Nº 5.1.4: LCA: RAM oculta para configurar conexiones.

Por todo lo enunciado hasta ahora, vemos que la utilización de una RAM para la configuración del dispositivo programable, cambia conceptualmente todo lo referente a ellos, en especial la puesta en funcionamiento. Esto también proporciona a los diseñadores nuevas e interesantes perspectivas, por cuanto el hecho de que una memoria RAM pueda ser leída y reescrita en cualquier momento, posibilita modificar la configuración de la lógica, inclusive durante el funcionamiento de la aplicación.

### Arquitectura de las LCA:

De acuerdo a lo indicado hasta ahora, la estructura interna toda LCA está compuesto por dos componentes básicos: Celdas lógicas e interconexiónado entre ellas. Y estos, además tienen varias formas, que podemos resumir en el siguiente cuadro:

- **Celdas Lógicas**
  - 1. Celdas Lógicas generales ó CLB Configurable Logic Block
  - 2. Celdas Lógicas de entrada-salida ó IOB Input Output Block.
- **Interconexiones**
  - a - De propósitos generales
  - b - Directas
  - c - Líneas largas

### • Celdas Básicas de las LCA:

Como se ha podido apreciar en la presentación general, podemos decir que hay dos tipos de celdas básicas:

- Celdas Lógicas propiamente dichas, llamadas **CLB** ó Configurable Logic Block.
- Celdas de Entrada/Salida, llamadas **IOB** ó Input/Output Block.

Analizaremos cada una de ellas, presentando las principales características y configuraciones:

1. **Celdas lógicas ó CLB:** Este bloque es muy funcional y ampliamente configurable, posibilitando la programación para cualquier tipo de aplicación lógica. En la figura 5.1.5 vemos su estructura interna, la cual puede configurarse de acuerdo a las necesidades específicas, ya que posee: dos (2) biestables del tipo D, una celda de lógica combinacional pura y varios multiplexores programables, que permiten seleccionar los diferentes modos funcionales del block. Puede programarse como puramente combinacional, secuencial, ó un circuito complejo.

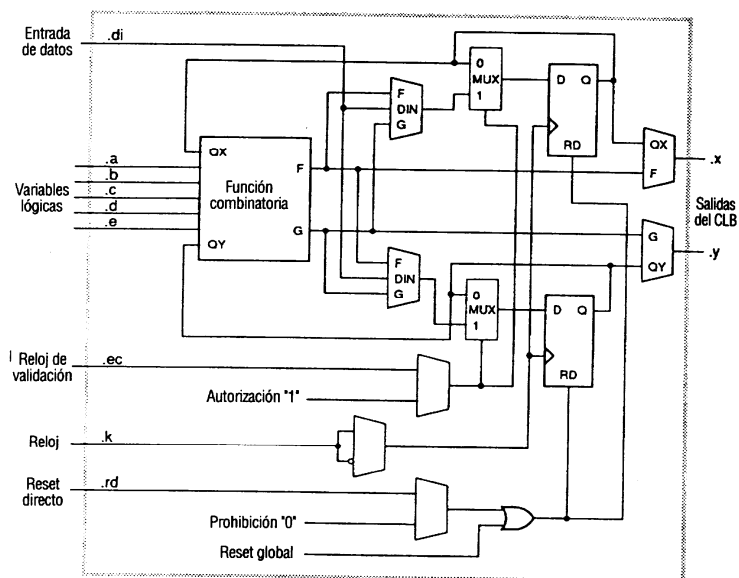


Fig. Nº 5.1.5: CLB:

Estructura interna.

Ahora analizaremos cada una de estas partes en detalle:

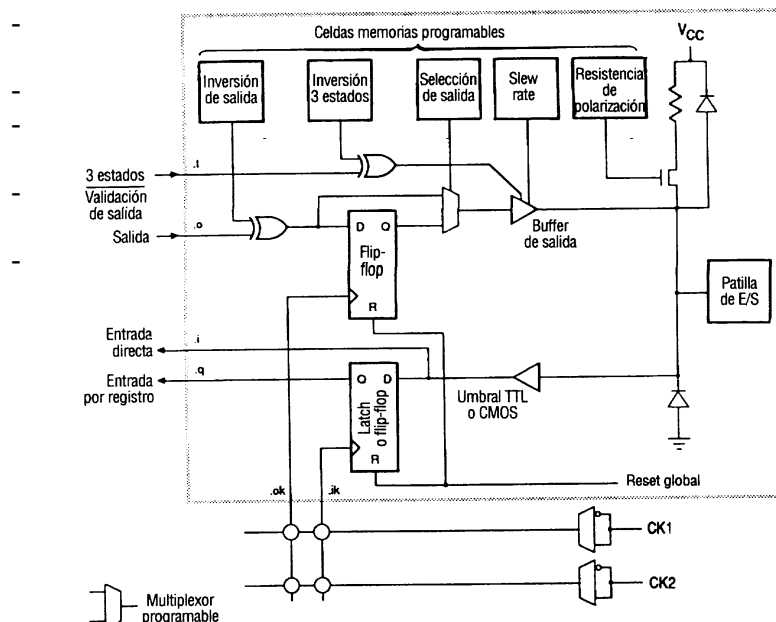
**a) Bloque lógico combinacional**, llamado también LUT (Look-up tables). Este es un elemento nuevo para los dispositivos lógicos programables, y por ello es el que más interrogantes plantea al usuario de las LCA.

Posee cinco (5) entradas lógicas directas, desde la *a* hasta la *e*, dos entradas provenientes de la realimentación de la salida de los biestables, *QX* y *QY*; y dos (2) salidas. Es decir se trata de un block que realiza una multifunción combinacional de 7 variables. Analizando un poco más en detalle esta celda, podemos decir que trabaja en tres modos, como se puede ver en la fig. Nº 5.6 y se describe ahora:

- **Modo FG:** se generan dos funciones combinacionales independientes de 4 variables cada una. La variable  $a$  debe ser común a ambas funciones, la segunda y tercera variable puede elegirse entre  $b$ ,  $c$ ,  $QX$  y  $QY$ ; y la cuarta se puede seleccionar entre  $d$  ó  $e$ .
- **Modo F:** Permite realizar cualquier función lógica combinacional de 5 variables cada una. Las variables  $a$ ,  $d$ , y  $e$  forman parte de la función como entradas fijas, y las dos restantes se las puede elegir a partir de  $b$ ,  $c$ ,  $QX$  y  $QY$ .
- **Modo FGM:** Es muy parecido al modo FG, pues se generan dos funciones combinacionales independientes de 4 variables cada una, donde las entradas fijas son  $a$  y  $d$ , la tercera y cuarta variable puede elegirse entre  $b$ ,  $c$ ,  $QX$  y  $QY$ ; y la variable  $e$  actúa sobre un multiplexor posibilitando seleccionar una de las salidas de ambas funciones .

**b) Bloque lógico secuencial:** Conformado por dos biestables D activado por flancos, los cuales pueden ser activados con flanco de subida ó de bajada, según la programación escogida. Pueden tener un reset propio proveniente de la zona de interconexiones, ó uno global, común con las celdas IOB. La señal de reloj, y la de validación de reloj también proviene de la zona de interconexiones.

En esta estructura podemos diferenciar algunas partes, a saber:



**Fig Nº 5.1.6: LCA: Estructura interna de un Input-Output Block (IOB)**

Como vimos, la salida puede pasar ó no a través de un biestable tipo D . Este posee un reset de accionamiento individual ó global, de tal forma que al dar energía al sistema provoca una puesta a cero general de todos los elementos secuenciales, para comenzar el funcionamiento desde un estado perfectamente definido. También puede ser activado por una pata especial del circuito integrado.

El reloj de los biestables que conforman este block de entrada-salida proviene de la zona de interconexiones independientemente elegido. Estos biestables pueden funcionar activados por nivel ó flancos. La activación de los mismos puede realizarse con un flanco ascendente ó uno descendente, según la programación elegida.

La señal de entrada puede ser del tipo secuencial ó simplemente combinacional según se programe que pase ó no por el biestable correspondiente. Puede tener todas las características descritas para el biestable de salida, según sea su programación.

### • **Interconexiones Básicas entre LCA:**

Hasta ahora hemos descripto y analizado las celdas lógicas elementales, pero falta un componente importante y es la conectividad entre las mismas, pues de nada sirven dichas celdas si no están interconectadas o si la misma es deficiente. Esta conectividad Xilinx la realiza a través de tres formas diferentes, de acuerdo a la longitud y el destino de las uniones, a saber:

- Interconexiones de propósito general.
- Interconexiones directas .
- Interconexiones largas (long lines).

Por supuesto, que considerando el fundamento básico de las LCA, todos ellos están basados en el mismo concepto de interconexión, a través del establecimientos de uniones con transistores MOS controlados por la memoria RAM. Todos estos caminos de interconexión se realizan a nivel de transistores MOS de conmutación , a través de lo que se

conoce como PIP ó *Programmable Interconnect Points*, los cuales están distribuidos a través de todo la superficie de la pastilla. En la fig. N° 5.1.7, se pueden ver algunas de estas conexiones.

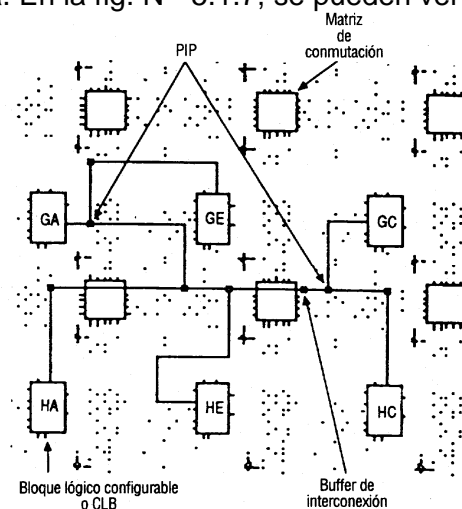


Fig. N° 5.1.7. PIP: Muestra de algunas uniones.

### 1. Interconexiones de propósito general:

Consiste en una matriz formada por buses verticales y horizontales, formados físicamente cada uno de ellos por una rejilla de 5 segmentos metálicos colocados entre las filas y columnas de los CLB e IOB, tal como se ve en la fig 5.1.8. En las intercepciones de los buses verticales con los horizontales se encuentra un dispositivo llamado *matriz de conmutación*, el cual permite distribuir la señal que viaja por los segmentos según las necesidades de cada diseño en particular.

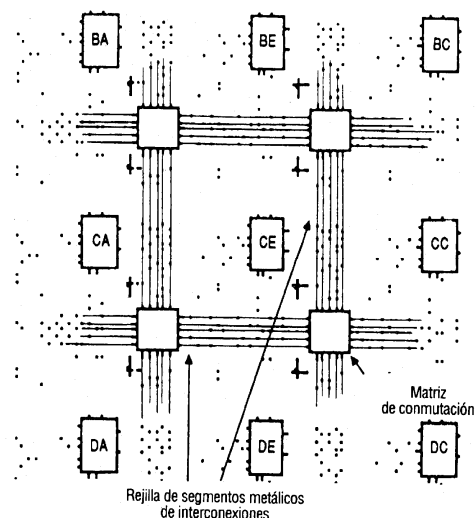
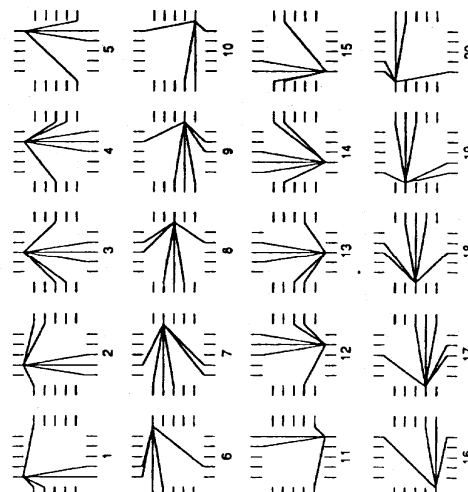


Fig. N° 5.1.8. LCA: Distribución de interconexiones y matrices de conmutación.

Esta distribución no es el intercambio de segmentos entre sí; pero, la fig 5.9, existe una gran posibilidad una cierta uniones.



total, ya que no posibilita información de todos los tal como puede verse en cantidad de ellas; lo cual flexibilidad a dichas



**Fig. Nº 5.1.9. Matrices de conmutación: Distribución de señales.**

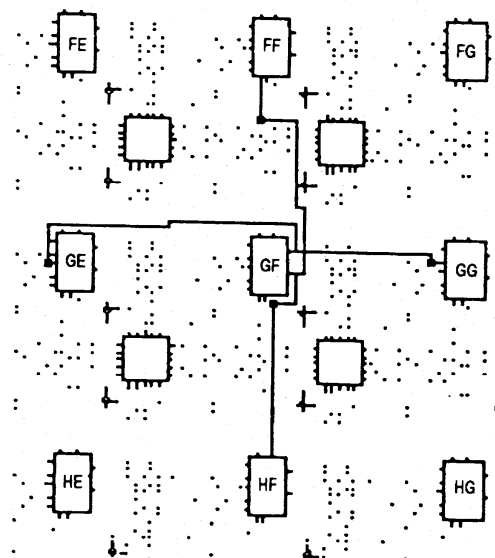
Junto a cada matriz de conmutación se emplazan regularmente buffers. Estos tienen la función de reconstituir las señales que recorren grandes distancias a través de las líneas que recorren la pastilla. Estos buffers, tal como puede verse en la fig 5.1.7, se distribuyen por encima y a la derecha de todas estas matrices.

## 2. Interconexiones directas:

Este tipo de unión permite realizar uniones entre los CLB e IOB con un máximo de eficacia considerando la velocidad y ocupación de la pastilla de C.I., como se ve en la fig. 5.1.10. Esto se puede hacer, puesto que de acuerdo a la geometría de las celdas es posible conectar en forma directa ciertas salidas de celdas a las entradas de otras sin la necesidad de otros recursos de interconexión. Las uniones que se pueden realizar de esta forma son las siguientes:

- **Entre CLB:** Se puede conectar la salida X directamente a la entrada B del CLB ubicado inmediatamente a su derecha y a la entrada C del CLB de su izquierda. En forma similar la salida Y puede conectarse directamente a la entrada D del CLB superior ó a la entrada A del CLB inferior.
- **Entre CLB e IOB** adyacente: se permite la conexión directa por medio de las entradas I y las salidas O respectivas, según su posicionamiento en la pastilla.

**Fig. Nº 5.1.10 : Interconexiones directas**



## 3. Líneas largas ó “ long lines “:

Este tipo de conexión es bastante diferente a los mostrados anteriormente, ya que no utilizan las matrices de conmutación para su realización. Se utilizan conexiones tipo bus que atraviesan la pastilla de extremo a extremo: de arriba abajo y de derecha a izquierda. Tienen una prioridad de utilización cuando las señales lógicas deben transmitirse con el mínimo de retardo entre los diferentes elementos, con el propósito de lograr un correcto y eficaz sincronismo de funcionamiento.

Su distribución está mostrada en la fig. 5.1.11; y en la cual se puede ver tres líneas verticales entre las columnas de los CLB y dos horizontales entre las

líneas de CLB a ambos lados de las matrices de conmutación. Y en el caso de las matrices más complejas se colocan dos líneas largas adicionales. Inclusive a veces se las puede dividir en dos partes, formando semilíneas largas.

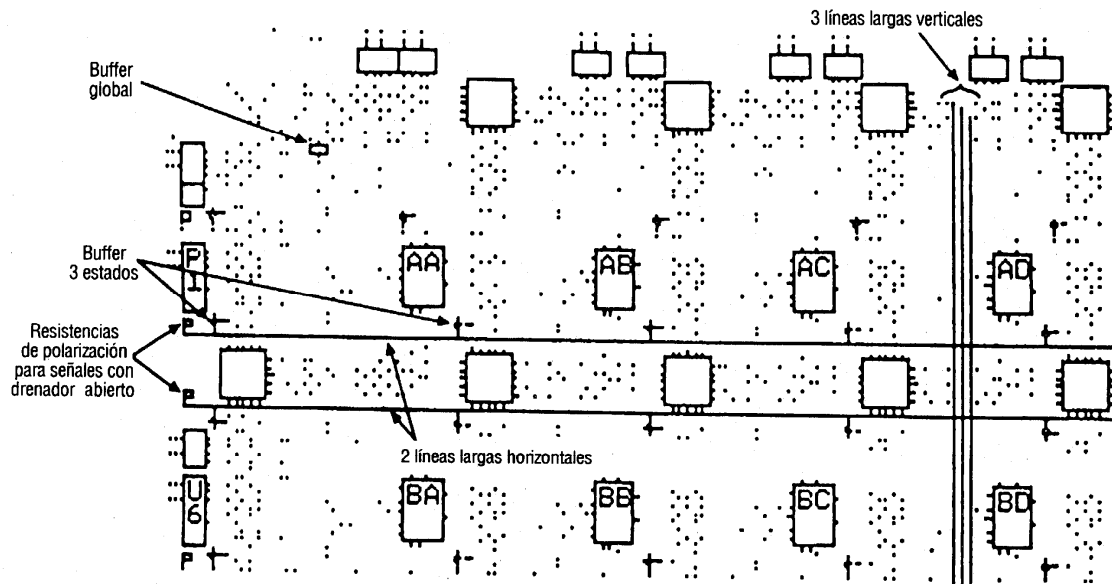


Fig. Nº 5.1.11. Líneas Largas: Distribución en la pastilla.

La transmisión de información digital por estas “long lines” puede ser manejada por la salida de un CLB ó un IOB en forma tradicional mediante el uso de un buffer disponible de la matriz; ó caso contrario como si fuera un bus; utilizando buffers tiestate, también disponibles sobre la pastilla.

### **Características propias de las LCA: Oscilador de cuarzo**

Puesto que todos estos dispositivos programables están orientados, es general, a la realización de sistemas digitales síncronos, y que los mismos poseen una señal de sincronismo general ó reloj, Xilinx agregó en la pastilla de las LCA un amplificador inversor de alta velocidad, que puede ser adecuado en la fase de programación como un oscilador de cuarzo; con el objetivo claro de simplificar cualquier desarrollo que requiera una señal de sincronismo genérica.

Como se puede apreciar en la fig 5.1.12, este oscilador se realiza utilizando dos IOB contiguos.

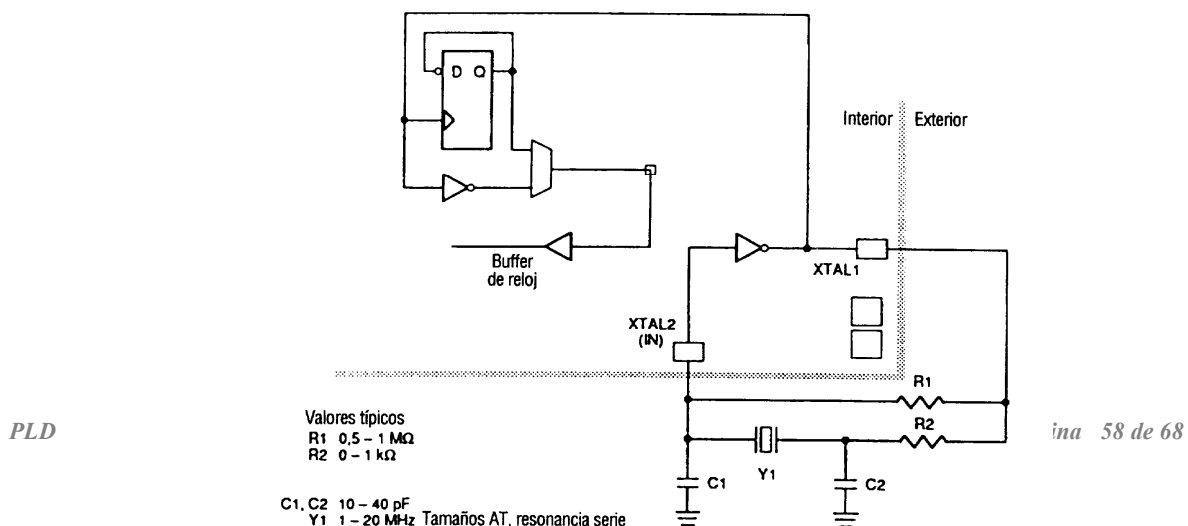


Fig. Nº 5.1.12. LCA: Realización de un oscilador de cuarzo.

Respecto de la realización de este oscilador hay que tener dos consideraciones importantes:

- a) El mismo puede realizarse sólo en el ángulo que la pastilla integre el amplificador de alta velocidad; y este depende exclusivamente del encapsulado utilizado.
- b) Si no se utiliza el oscilador, los IOB adyacentes, se pueden usar normalmente, como otro recurso más de la LCA correspondiente.

## **Programación:**

Como ya se ha mencionado, las LCA incorporan como característica distintiva, que su programación se realiza copiando en la RAM interna la configuración de conexiones que se desea realizar. Esta información, en cuanto a los datos propiamente dichos y al formato que los mismos ocuparán en la RAM es muy técnico y privativo del fabricante, por lo tanto no presenta interés para el diseñador propiamente dicho, ya que se genera automáticamente por el software de desarrollo en forma de un paquete de datos que es suministrado a las LCA. Pero, por el contrario, lo que si se debe saber es *como* enviar dicho paquete de datos y tener presente como será el resguardo en dicha LCA al perder la alimentación.

Básicamente las LCA poseen tres modos para cargar los datos luego de aplicarle tensión, según el siguiente detalle:

- Modo Maestro
- Modo Esclavo
- Modo tipo Periférico.

### **1.Modos Maestro :**

Este modo presenta varias opciones:

**a) Datos Serie ó propietario:** Su nombre se debe a que los datos son transmitidos desde la memoria a la LCA en serie, según una estructura y componentes propios de Xilinx; es decir la LCA correspondiente y una memoria de acceso secuencial ó serie, fueron desarrollados y comercializados por esta empresa. Esta memoria EPROM del tipo OTP está programada con el mismo sistema de desarrollo utilizado para la LCA, y sólo sirve para reprogramarla cada vez que sea necesario. Es la opción más sencilla, desde el punto de vista del hardware, ya que es la que menos lugar ocupa en el circuito.

La LCA a configurar genera todas las señales del protocolo de hardware, para que la PROM vuelque sus datos a ella. Cuando se culmina la programación de la RAM interna, la misma LCA genera una señal indicando que el proceso ya está cumplido ( DONE ). Cuando el circuito posee muchas LCA, se puede utilizar una sola PROM para programarlas a todas, conectadas en cascada, tal como se ve en la figura 5.1.13.

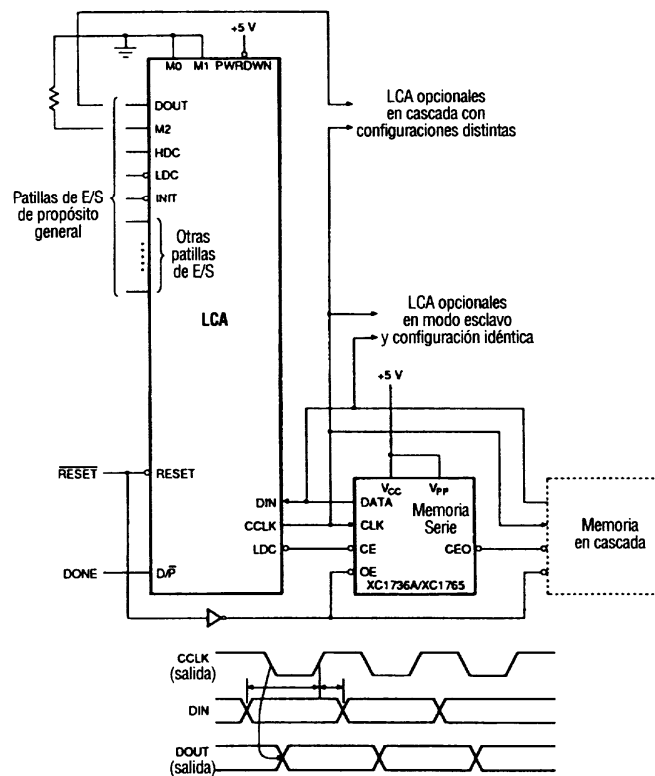


Fig. Nº 5.1.13.

LCA: Esquema en modo maestro serie.

Programación de

**a) Datos paralelo ó general:** Este modo es muy parecido al anterior en cuanto a la funcionalidad, pero su diferencia está en que los datos de reprogramación, son suministrados por la memoria PROM en forma paralelo. En este caso se puede utilizar una memoria PROM clásica y de cualquier fabricante (UV-EPROM ó EEPROM ). El conexionado básico entre ellos se ve en la fig. 5.1.14. En este caso, evidentemente se necesitan mayor cantidad de líneas de conexionado, pues necesitamos el conexionado al bus de direcciones y de datos, dependiendo en general de la capacidad de información a programar en la LCA correspondiente. En la figura mencionada se puede observar un direccionamiento de 10 bits y 8 de datos, es decir corresponde a una memoria de 2 K x 8.

En este caso, también se puede realizar programación en cascada de las LCA. Igual al caso anterior existe una señal que indica cuando finalizó la reprogramación ( DONE ).

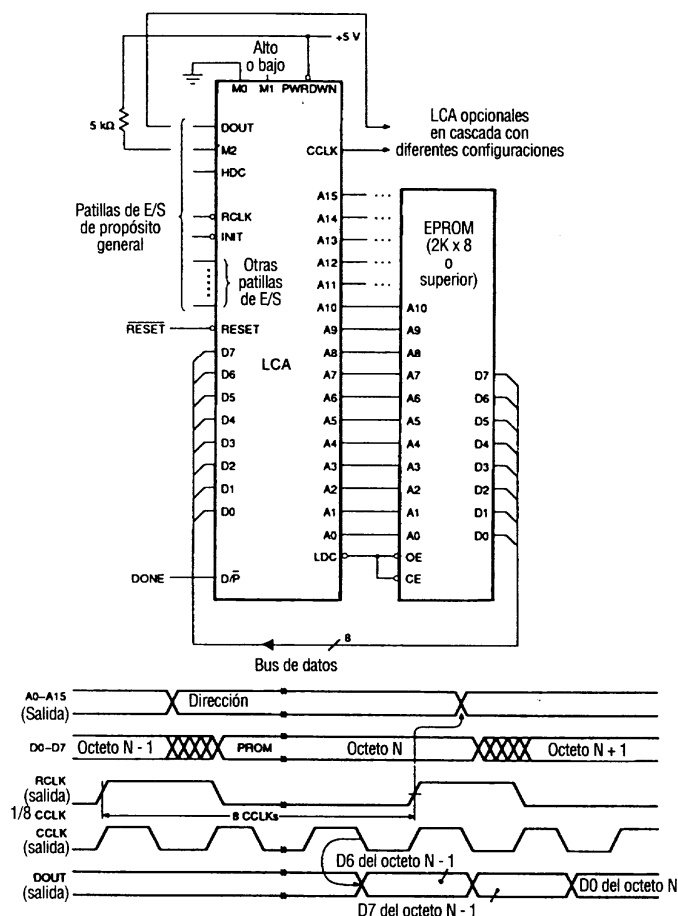


Fig. Nº 5.1.14. Programación de LCA: Esquema en modo maestro paralelo.

Cabe hacer una consideración importante y es que las líneas de programación mencionadas, luego de cumplir con este propósito vuelven a tomar su uso normal dentro de la operatividad normal de entrada/salida lógica de la LCA. Es decir, no existen patas diferenciadas para programación; estas son las mismas para ambos propósitos. Por lo tanto en este modo de reconfiguración, es importante tener mucho cuidados en la fase de diseño del circuito, para evitar interferencias de datos en ambas operaciones.

Vemos que este modo permite la utilización de una memoria normal y es bastante más rápido, pero presenta algunas complicaciones, como son el hecho de que ocupa más lugar en el circuito impreso y exige un diseño muy cuidadoso de la circuitería anexa.

Existen dos formas de concretar este tipo de reprogramación, y es el sentido de evolución de las direcciones, en un caso parten de 0000<sub>H</sub> y van aumentando mientras que en el caso restante parten de FFFF<sub>H</sub> y van disminuyendo. Estas posibilidades se realizan para lograr compatibilidad con los microprocesadores de diferentes marcas, pues algunos comienzan a ejecutar los programas a partir de las direcciones altas y otros de las bajas.

## 2. Modo Esclavo:

En este caso la LCA recibe los datos desde una LCA maestra, como se mencionó en el ítem anterior. La primera de la cascada se llama maestra y las demás esclavas. Esta LCA maestra origina el ciclo de reloj (CLCK) para toda la cascada. Los datos de programación, proveniente de la PROM, ingresan en la LCA por la entrada serie (DIN) y se

transmiten a las otras LCA a través de la salida serie ( DOUT ). Este conexionado se ve en la fig 5.1.15.

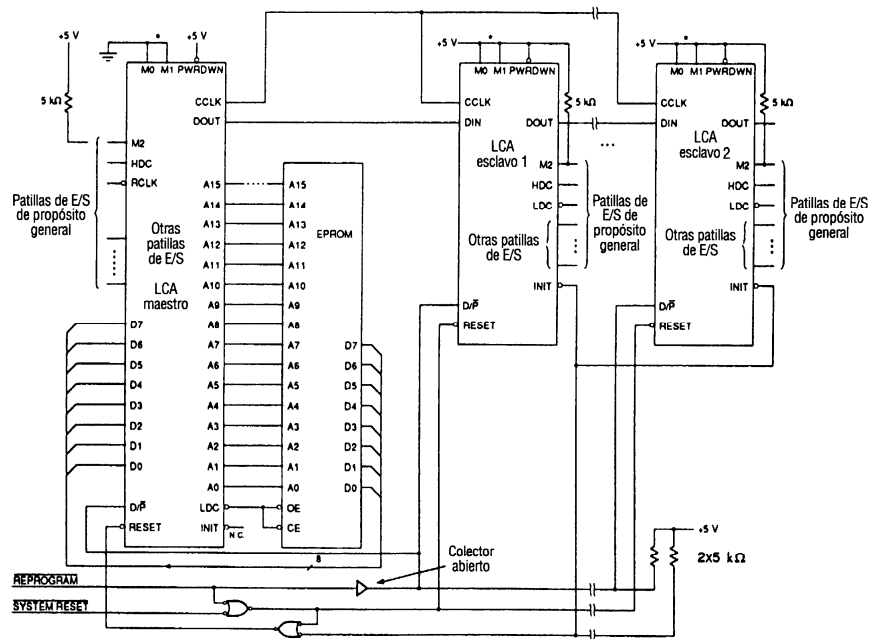


Fig. Nº 5.1.15. Programación de LCA: Conexionado general maestro-esclavo.

La LCA maestra puede estar conectada en cualquiera de los otros modos, sin que se modifique el formato de cableado de las esclavas. Este modo es muy utilizado cuando el diseño posee muchas LCA, pues el cableado a nivel de las esclavas resulta mínimo.

### 3. Modo Periférico:

También llamado modo pasivo, pues se ha estructurado para que la LCA sea configurada ó reprogramada a partir de un microprocesador cualquiera, como si fuera un simple periférico, como se puede ver en la fig. 5.1.16.

Como puede verse, el bus de datos se aplica a la entrada de datos de la LCA, el bus de direcciones es decodificado para ubicarla en el mapa de memoria del microprocesador y algunas otras líneas de control que se usan en el proceso de programación.

Frente a la inicialización de la LCA ó reprogramación de la misma, por cualquier motivo, el microprocesador toma los datos que tiene almacenados en su memoria ROM ó PROM y los transfiere a la LCA, de acuerdo a un protocolo adecuadamente estructurado. Esta configuración es muy útil para aplicaciones de tiempo real, es decir para aplicaciones en las cuales la programación de la LCA se va modificando instantáneamente.

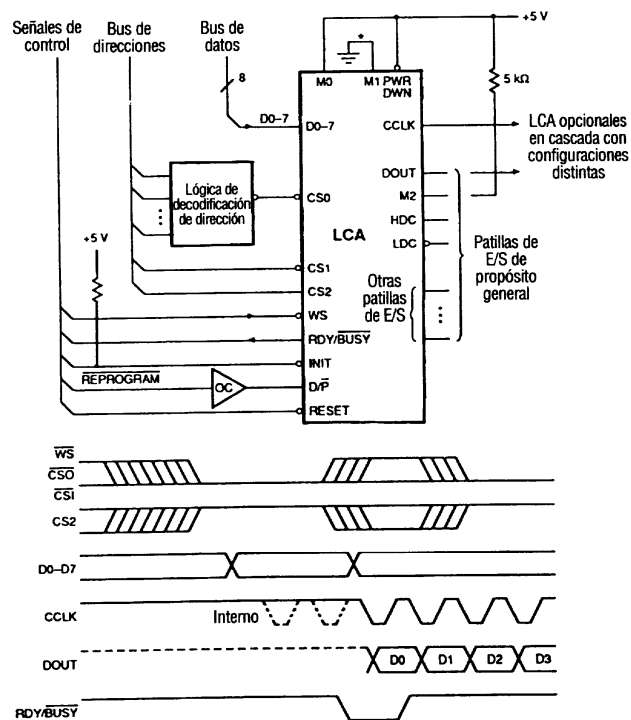


Fig. Nº 5.1.16. Programación de LCA: Conexionado modo periférico.



Estos dispositivos programables fueron introducidos hacia los años 1991-92 con lo que podemos decir que se trata de los últimos circuitos de este tipo aparecidos en el mercado, a través de dos empresas muy diferentes, Texas Instruments, una de las más grandes fabricante de circuitos integrados y la modesta Plus Logic, para aquellos años de reciente aparición.

Al igual que las LCA de Xilinx, las FPGA de antifusibles tienen como premisa fundamental, la optimización de los recursos dispuestos en la pastilla por medio de un mejor uso de las posibilidades de interconexión entre los diferentes bloques lógicos. Pero, si bien la arquitectura es similar a la de las LCA, como se ve en la fig. 5.2.1. pues incluye una gran cantidad de bloques lógicos en el centro de la matriz, rodeado por un cierto número de celdas lógicas de entrada/salida, y un bus adecuado de interconexión, el método utilizado para alcanzar este resultado es completamente distinto al usado en dichos circuitos programables.

La diferencia fundamental, es que en este tipo de FPGA, si bien estamos frente a circuitos con programación eléctrica, no permiten el borrado y posterior reprogramación ya que la grabación es definitiva, como pasa con las PAL de fusibles ó las memorias PROM. Quienes defienden este tipo de circuito, basan su postura en que gracias ha esto se pueden lograr altas densidades de integración, ya que las interconexiones se realizan a través de los llamados *antifusibles* que ocupan estructuralmente muy poco espacio sobre la pastilla del circuito integrado.

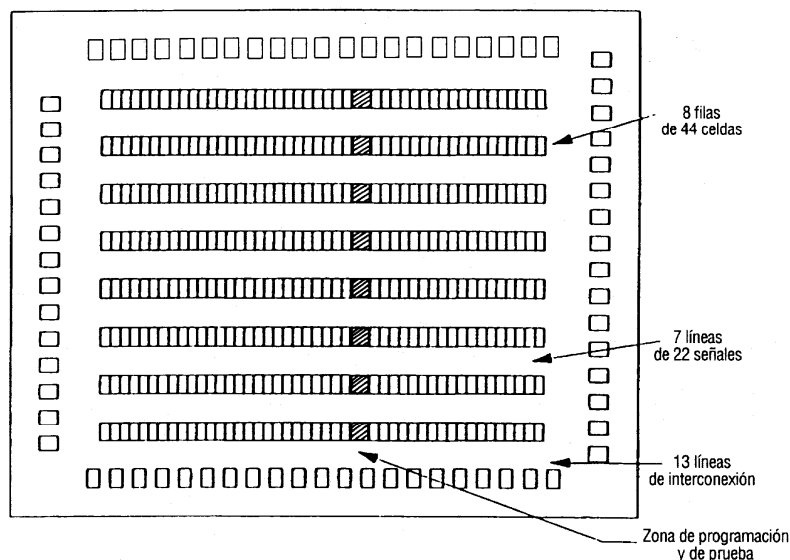


Fig. Nº 5.2.1. FPGA de Antifusibles: Estructura interna.

### Principio del antifusible:

En la fig. 5.2.2 puede observarse un esquema de la estructura típica de un antifusible: Consta de dos líneas metálicas, colocadas en dos niveles diferentes y que permanecen aisladas por una fina capa de óxido. Al aplicar un impulso de alta tensión (entre 20 y 24 volts) durante un tiempo especificado (menor a 5 mseg) la capa de óxido es perforada y



las dos líneas metálicas se unen estableciendo la unión-antifusible. Vemos que realmente tiene el efecto opuesto, ya que mientras en las *PAL de fusibles* se queman las uniones, en las *FPGA de antifusibles* se generan dichas uniones. Algunos autores mencionan que para programar las PAL se hacen saltar los fusibles, con el fin de establecer las uniones buscadas, y en caso de las FPGA se hace saltar “*alguna cosa*” para establecer dichas uniones. Y entonces han bautizado a esta “*alguna cosa*” como antifusible.

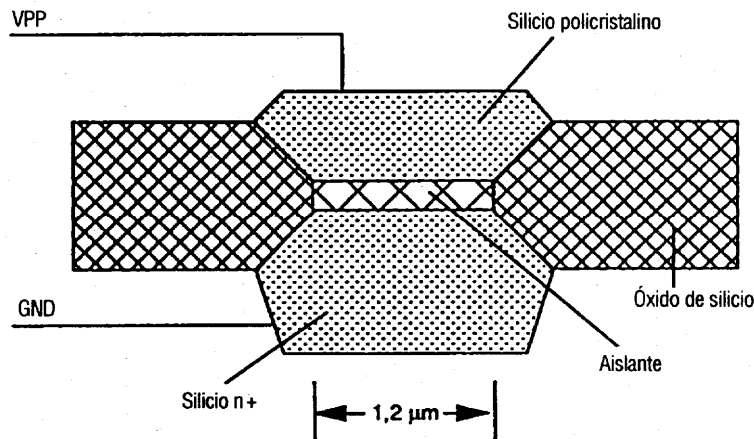


Fig. Nº 5.2.2. Antifusible: Estructura típica.

### Celdas lógica básica:

A diferencia de las LCA, la celda lógica de la FPGA de antifusibles es muy sencilla y en general del tipo combinacional pura, por lo que si es necesario realizar un circuito secuencial, hace falta interconectar por lo menos dos celdas. Esto lo podemos apreciar en la fig. 5.2.3.

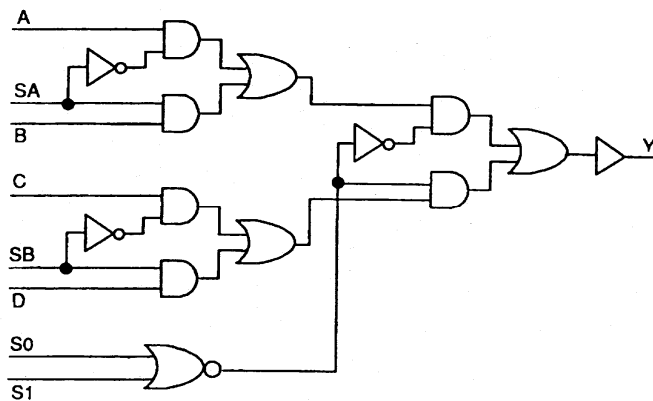


Fig. Nº 5.2.3. FPGA de Antifusible: Celda básica (doc. Texas)

Por lo visto cualquier diseño puede resultar sumamente complejo, y con la posibilidad cierta de no obtener el circuito más eficaz. Por esto las marcas que realizan este tipo de dispositivos suministran programas de desarrollo capaces de alcanzar los circuitos más efectivos y simples. Este software por lo general incluye pequeños programas, macros, ó rutinas que facilitan tal tarea. Estas pueden ser de dos tipos:

- *Hard macros*: Funciones lógicas básicas que utilizan una ó dos celdas pre-encaminadas para lograr la máxima eficiencia.

- **Soft macros:** Funciones lógicas complejas, que utilizan muchas celdas básicas (90 ó más), para la realización de contadores, registros, registros de desplazamiento, etc. sin un pre-encaminado, y que por lo tanto se puede implantar como se desee en cualquier diseño de circuito lógico complejo.

Las celdas de entrada/salida se pueden definir como entradas ó salidas. Y en ambos casos como normales ó triestado. Además son compatibles con las tecnologías más usadas, como TTL y CMOS.

Estas celdas son bastante simples, por lo que permiten una gran flexibilidad de interconexión; a pesar de que ello implica la necesidad de utilizar un gran número de ellas para desarrollar cualquier diseño medianamente complejo.

### **Modos de Trabajo:**

Las FPGA presentan dos modos básicos de funcionamiento:

- **Modo Normal:** En este modo el circuito permite ser usado para la función para la cual fue desarrollado.
- **Modo Probe:** En este modo se deshabilita para el funcionamiento normal y permite al usuario ó desarrollador chequear el funcionamiento interno a través del uso de un hardware y software especial para este modo. En el caso de Texas, sus FPGA tienen circuitos internos especiales llamados action probe. Estos permiten, a través de patas específicas, actuar un software y hardware llamado igual: *ActionProbe* que permite revisar internamente la funcionalidad de la FPGA, a través de facilitar el análisis de cualquier punto interno de conexión del circuito. Esto se ve en la fig. 5.2.4

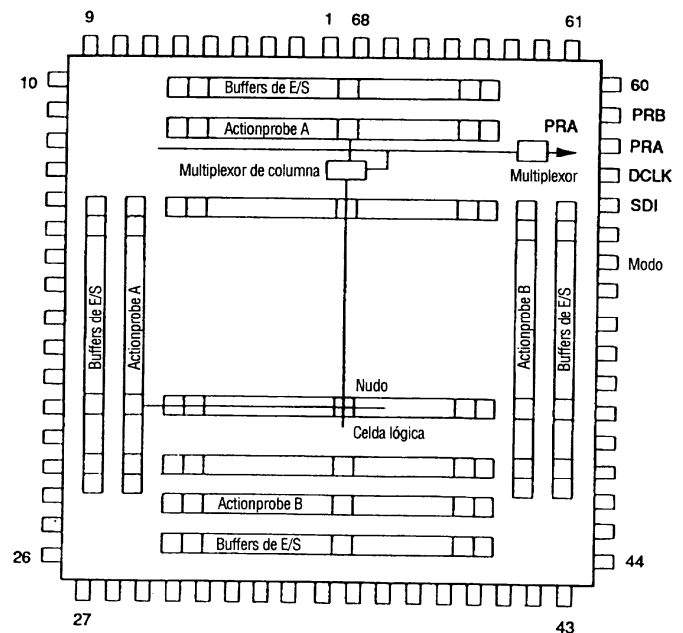


Fig. Nº 5.2.4. ActionProbe: Principio de funcionalidad.

### **Interconexiones:**

Las FPGA contienen entre las celdas básicas un gran número de segmentos metálicos horizontales y verticales conformando verdaderas mallas ó matrices de interconexión. Existen dos posibles caminos de interconexión: Donde se produce el cruce de tales buses, se aíslan a través de los antifusibles vírgenes permitiendo una futura unión. Además tales segmentos metálicos son regularmente interrumpidos por transistores MOS, a los cuales se los puede hacer conductores ó aisladores, según la polarización. Ambas posibilidades permiten una gran flexibilidad de interconexión, fundamentalmente considerando que sobre la pastilla se implementa un gran número de antifusibles. Podemos mencionar el TPC 1010 de TEXAS que contienen apenas 1200 compuertas y 120000 antifusibles. Esto lo vemos en la fig 5.2.5.

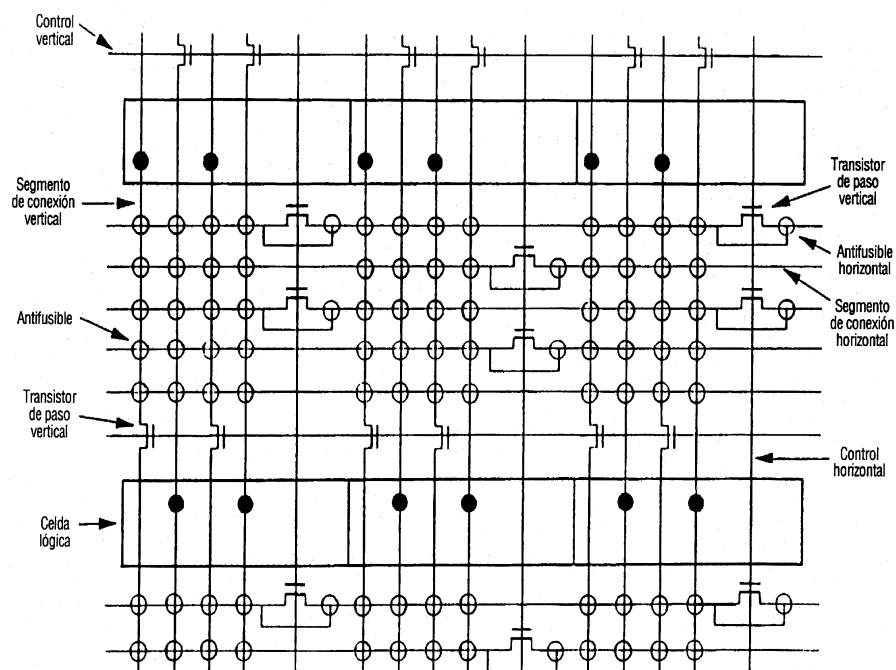


Fig. Nº 5.2.5. FPGA de Antifusible: Estructura de interconexiones

### Familias de FPGA de antifusibles:

Algunas de las primeras familias de estos circuitos fueron introducidas por Texas con la familia TPC. Presentan características muy modestas como vemos en la sig. tabla.

	TPC 1010A	TPC 1020A	TPC 1240	TPC 1280
Número de compuertas	1200	2000	4000	8000
Proces. CMOS	1.2	1.2	1.2	1.2
Número de celdas.	295	147	684	1232
Número máximo de biestables.	147	273	565	994
Número de antifusibles	112000	186000	400000	750000
Número de líneas de conexión por canal.	22	22	36	36
Número de patas de E/S	57	69	104	140
Frecuencia máxima	20 Mhz	20 Mhz	30 Mhz	30 Mhz
Encapsulado	68 PLCC	68 PLCC 84 PLCC	132 CPGA	176 CPGA

### Herramientas de desarrollo:

Para estos circuitos, es muy importante el sistema de desarrollo y programación utilizado, ya que en un EPLD ó una LCA se permiten errores de diseño, ya que permiten ser borrados; pero en este caso cuando se programa, si fue mal realizado se pierde el circuito.

Los sistemas de desarrollo están realizados por cada empresa en particular para sus circuitos, pero en general cuentan con fases de operatividad más ó menos parecidas, a saber:

- Fase de entrada ó trazado esquemático del circuito:** Generalmente puede ser reemplazado por cualquier programa de entrada standard ó de uso genérico como el Orcad, View Logic, Futurenet, Exemplar Logic, Mentor Graphics, etc.

- b) **Fase de simulación lógica:** Con el fin de asegurar con el mínimo detalle la correcta funcionalidad lógica del circuito que se está desarrollando.
- c) **Fase de asignación de patas.**
- d) **Fase de encaminamiento.**
- e) **Fase de simulación temporal.**

- Si las fases anteriores han sido correctas viene la concreción del circuito –

#### f) Programación

Actualmente varias empresas ofrecen sistemas de desarrollo para ser usados con computadores personales, para los cuales vienen placas especiales que se conectan en el bus ISA ó PCI y se conectan a una placa dónde se coloca el dispositivo programable a ser configurado.

Cuando luego de programado el circuito presenta una falla grave de funcionamiento, en general se puede analizar colocándolo en modo *Probe*. En el caso específico de Texas, como ya mencionamos oportunamente, se ofrece un depurador de circuitos ( debugger) llamado *ActionProbe* que permite analizar el funcionamiento de las FPGA e incluso visualizar ciertos nudos internos.

