A German Shepherd dog is sitting on a wooden table, looking to the right with its tongue hanging out. A green ball is on the table next to its front paws. The background shows a brick wall and a window with white curtains.

Diseño de Sistemas con FPGA

**Patricia Borensztein
Primer Cuatrimestre
2012**

Objetivo del curso

- Adquirir conocimientos elementales de la arquitectura de las FPGA actuales
- Aprender a utilizar las herramientas de programación y realizar dos proyectos sencillos “que anden” (uno combinacional y otro secuencial) y finalmente un proyecto algo mas interesante a acordar entre todos.

Temario y Planificación

- Introducción las FPGA (1)
- Modelado de Sistemas Combinacionales. Verilog. Entorno de desarrollo de xilinx. (5)
- Modelado de Sistemas Secuenciales. (5)
- Co-Diseño software hardware con Picoblaze (5)
- Proyectos (11)

Bibliografía del curso (la que yo uso...)

- FPGA Prototyping by Verilog Examples. Pong P. Chu. Wiley InterScience. (Verilog e ISE)
- FPGA-Based System Design. Wayne Wolf (tecnología)
- The Design Warrior's Guide for FPGAs. Clive Maxfield (arquitecturas)
- Digital Arithmetic. Ercegovac, y Thomas Lang (algoritmos aritméticos)
- Synthesis of Arithmetic Circuits. Deschamps, Gioul, and Sutter. (algoritmos y codificación en VHDL para FPGA)
- Writing Testbenches: Functional Verification of HDL Models. Janick Bergeron. Kluwer Academic Publishers.
- Hojas de Datos.... (de FPGA's...)
- Y todo lo que ustedes encuentren en la web

Hardware y Software

- La cátedra cuenta con los siguientes placas de desarrollo:
 - Spartan-3E Starter Kit (3 unidades, Digilent) donada por XUP (Xilinx University Program)
 - Spartan-3 Starter Board (Digilent) donada por XUP
 - Xilinx Spartan-3A EVALUATION KIT (Avnet) (propio)
 - Virtex-II Pro Development System (Xilinx University Program) (Digilent)
 - Virtex4 FX12 Evaluation Board (Avnet) con Audio/Visual Card (Avnet) (Propiedad del Grupo de Investigación: Embebidos)
 - Virtex-5 OpenSPARC Evaluation Platform (University Program OpenSparc)
 - NetFPGA con VirtexII Pro 50 FPGA. Donado por empresa proveedora de Internet, interesada en el prototipo.

Hardware y Software

- El software que utilizamos es :
 - ISE Webpack 11.1 (xilinx.com)
 - ModelSim PE Student Edition (model.com)
- Pueden bajarla directamente de xilinx, pues es gratuita, ojo! Bajar el webpack, ya que la herramienta completa (ISE Design Suite) NO es gratuita. (y pesa mucho mas)

Cuestiones Administrativas

- Usaremos 27 clases, teórico-prácticas. Comenzamos hoy, 28 de marzo y terminamos el 6 de julio.
- No se puede faltar mas de cinco veces.

Cuestiones Administrativas

- Los docentes de la materia somos:
 - Patricia Borensztejn (Profesora): patriciamiriam@gmail.com
 - Matias López : matiaslopez@gmail.com
- La página de la materia es:
<http://www.dc.uba.ar/materias/disfpga/2012/c1>
- El material de la página se irá subiendo de clase en clase, versiones anteriores en:
<http://www.dc.uba.ar/materias/disfpga/2011/c2>

Empecemos....

Introducción

- Sistemas Embebidos
- ¿Que son los FPGA?
- ¿En que se diferencian de un microprocesador?
- Origen de los FPGA: PLD's
- ASIC's
- FPGA

¿Donde estamos?



En el mundo de los sistemas
embebidos

Sistemas Embebidos

- El mundo está lleno de ellos.
- Se pueden definir como todo sistema que NO es una PC de escritorio , ni un servidor, ni una workstation, ni una supercomputadora, ni un cluster distribuido, en fin, NO es un sistema programable de propósito general.

Sistemas Embebidos

- El mercado de los productos embebidos es enormemente mas grande que el mercado de las PC's que lidera Intel, por ejemplo.
- Jim Turley, asesor de la industria de los embebidos, dice que aproximadamente el 0% de los microprocesadores que se fabrican, se usan en el mercado de las PC de escritorio. El 100% restante, se usan en el mercado de los embebidos.
- En los últimos años, se vendieron alrededor de 500 millones de microprocesadores para el mercado de las PC's y 10 mil millones para el mercado de los embebidos.

Embedded Processors by the numbers, 1999.

- <http://vault.embedded.com/1999/9905/9905turley.htm>
- Es este artículo de Jim Turley, asesor de la industria de los embebidos, escrito en el año 1999, el decía:

About zero percent of the world's microprocessors are used in computers. Yup. Every PC, Macintosh, engineering workstation, Cray supercomputer, and all the other general-purpose computers put together account for less than 1% of all the microprocessors sold every year. **If you round off the fractions, embedded systems consume 100% of the worldwide production of microprocessors.**

Jim Turley

- En 1999 se vendieron para el mercado de embebidos:
 - 250 millones de microprocesadores de 32 bits
 - 1000 millones de microprocesadores de 16 bits
 - 1000 millones de microprocesadores de 8 bits
 - 1000 millones de microprocesadores de 4 bits
- Contra 100 millones para el mercado de computadoras (PCs, MACs, workstations y supercomputadoras)
- **Hoy, 10 años después, se vendieron 10 mil millones de procesadores para el mercado de los embebidos!**

Sistemas Embebidos

- Son sistemas de hardware y software, de propósito específico embebidos en algún producto del mercado. Por ejemplo:
 - Embebidos de la industria automotriz : navegador GPS, de la aviación: control de aterrizaje automático
 - Embebidos de la industria de las telecomunicaciones: routers, modems
 - Embebidos de la industria de las comunicaciones: teléfonos celulares, palms, e-books
 - Embebidos de la industria del hogar: control de heladeras, microondas, robots
 - Etc, pero muchos etcéteras.

El mundo de los embebidos

- Aplicaciones específicas
- Optimizadas en consumo y área.
- Se trabaja en hardware y en software
- Se programa en C , ensamblador y/o HDL
- ¡¡¡Costosísima la parte de verificación y testing!!!

El mundo de los embebidos


- Mucha variedad en las implementaciones
- Si un ingeniero necesita desarrollar un producto nuevo, puede elegir entre diversas alternativas:
 - Procesadores de propósito general, adaptados para placas embebidas
 - DSP (procesadores específicos para procesamiento de señales)
 - Microcontroladores
 - ASIC's
 - FPGA's

El mundo de los embebidos ...

...es muy codiciado....

2010: Intel introduce una FPGA en el Atom Processor

Introducing the Intel® Atom™ Processor E600C Series
A Configurable Intel Processor



INTEL® ATOM™ E600C PROCESSOR SERIES

ALTERA FIELD PROGRAMMABLE GATE ARRAY

Flexibility

- Configurable with application-specific or proprietary I/O and algorithms
- Single Design supports multiple product derivatives
- Multiple functions combine into a single package for smaller form factor needs

Simplicity

- **Single Package** reduces footprint and helps lower costs
- **Ease of Design** with few chips and simplified inventory
- **Power of One** with one supplier, one package, and one support call

intel
Embedded

Y ¿que es el procesador Atom de Intel?

- Es el procesador que está ejecutando este código en mi portátil...
- Es el procesador mas pequeño de Intel, fabricado con tecnología de 45 nm (nm=mil millonésima parte del metro) y próximamente de 32nm.
- Pensados para una amplia gama de dispositivos: netbooks, PC básicas, tablets, laptops, smartphones, dispositivos electrónicos de consumo y otros dispositivos complementarios.
- Características del procesador:
 - Arquitectura de ahorro de consumo de energía, optimizada para su uso en los productos embebidos

¿Y que es una FPGA?

- Es... lógica programable organizada de alguna manera... dentro de un chip.
- Se usa para:
 - En los 80, cuando recién arrancaban, para implementar “lógica para unir” (glue logic), lógica para interfacear componentes mas grandes (porque sus antepasados son los PLD’s)
 - En los 90, telecomunicaciones y redes
 - 2000 ... todo tipo de aplicaciones del mercado de los embebidos...
 - 2010 ... ¿para hacer llegar fibra óptica a cada hogar?
 - 2020 ... todo tipo de aplicaciones, hoy no muy imaginables... ¿alguna idea?

Recordemos que ...

- Intel es líder en el mercado de procesadores para la industria de las PC básicas.
- ...pero no en el mercado de los embebidos... (donde reina ARM, especialmente en el mercado de los dispositivos móviles)
- Hoy, el mercado de los productos embebidos alcanza los 10 billones (miles de millones) de unidades vendidas incluyendo microcontroladores, procesadores, DSP, FPGA, ASICs.

¿Porque reina ARM en el mercado de los dispositivos móviles?

- El secreto de su éxito:
 - ARM no fabrica micros, solo los diseña y luego los licencia a fabricantes de electrónica que los integran en sus chips
 - La clave es que los microchips que finalmente salen al mercado son algo mas que microprocesadores; por ejemplo, un fabricante de teléfonos móviles fabricará procesadores que además del núcleo ARM contendrán el transmisor 3G y el controlador de pantalla y teclado; de esta forma reducimos el número de chips necesarios para montar el teléfono, con la consiguiente reducción de coste, tamaño y consumo.



Procesador para router ADSL basado en ARM

Idea de Intel para entrar en el mercado de las cosas pequeñas...

- Intel no licencia sus procesadores, por lo tanto si queremos hacer un teléfono móvil basado en Intel, tendríamos que montar un conjunto extra de chips...y eso trae problemas y deja a Intel fuera del mercado de las cosas pequeñas...
- Solución de Intel para entrar en ese mercado: hacer configurables a sus procesadores → poner una FPGA.
- La FPGA es un chip que puede convertirse en una tarjeta gráfica, un sintonizador wifi, un controlador de puertos, ... Dicho con otras palabras, una FPGA proporciona a los fabricantes una capacidad para personalizar el chip similar a la que proporciona ARM.

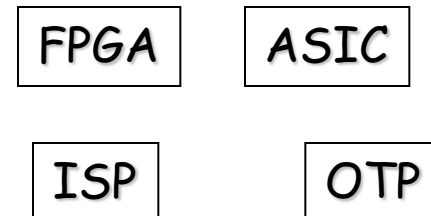
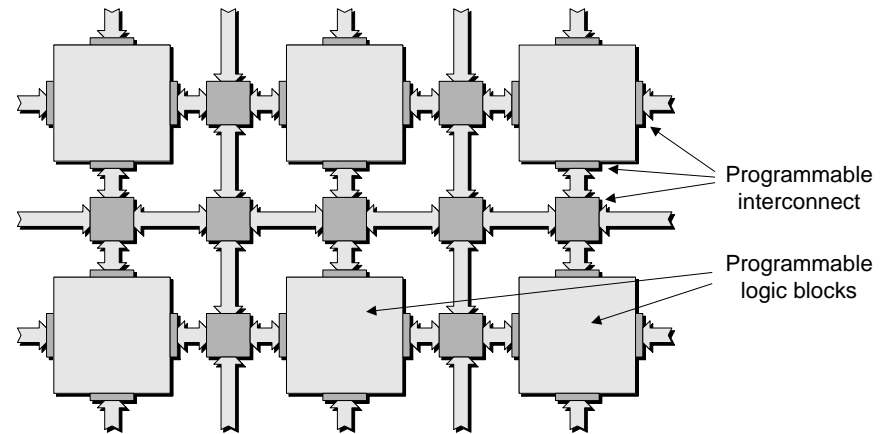
El mercado de los embebidos

- Es un muy buen mercado.... ¿no les parece?
- A Intel le pareció una muy buena idea irrumpir en ese mercado adosando al costado de su micro procesador un producto, la FPGA, claramente usable en el mercado de los embebidos para hacer aplicaciones ad-hoc.
- ¿Y que tal si nosotros también nos introducimos en el mundo de los embebidos?

¿Que son los FPGA?

Fiel Programmable Gate Array

- Arreglos de Puertas Programables en el Campo : (nunca mas lo traduzco)
 - Circuitos Integrados que contienen **bloques configurables de lógica junto con conexiones configurables entre esos bloques.**
 - Para que se configuran? Para realizar distintas tareas. Algunas FPGA permiten ser programados una sola vez(OTP one time programmable), o bien una y otra y otra vez...
 - Los FPGA se programan “in the field”, o sea, no los programa el fabricante, sino que lo puede programar el desarrollador “en su campo”, o sea, en su laboratorio. Si un dispositivo puede ser programado mientras está residente, o embebido en un sistema mayor, se dice que es ISP (In System Programmable)
 - ¿Porqué Gate Array? Es el nombre de un tipo de tecnología de fabricación de los ASIC (Application Specific Integrated Circuit)



Microprocesadores, FPGA, ASIC's

Microprocesadores

- Las funciones se realizan en software
- No son “custom parts” por lo tanto no son tan buenos como un chip dedicado (ASIC)

FPGA

- Las funciones se realizan en hardware.
- No son “custom parts”. Son mas lentos que los ASIC's, consumen mas energía.

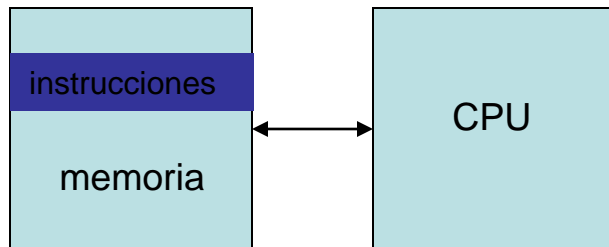
ASIC (Application Specific IC)

- Diseñado para implementar una función lógica particular. Son “hechos a medida”.
- Son más rápidos que los FPGA y manufacturados en gran escala son mas baratos.
- Consumen menos.

Microprocesadores y FPGA

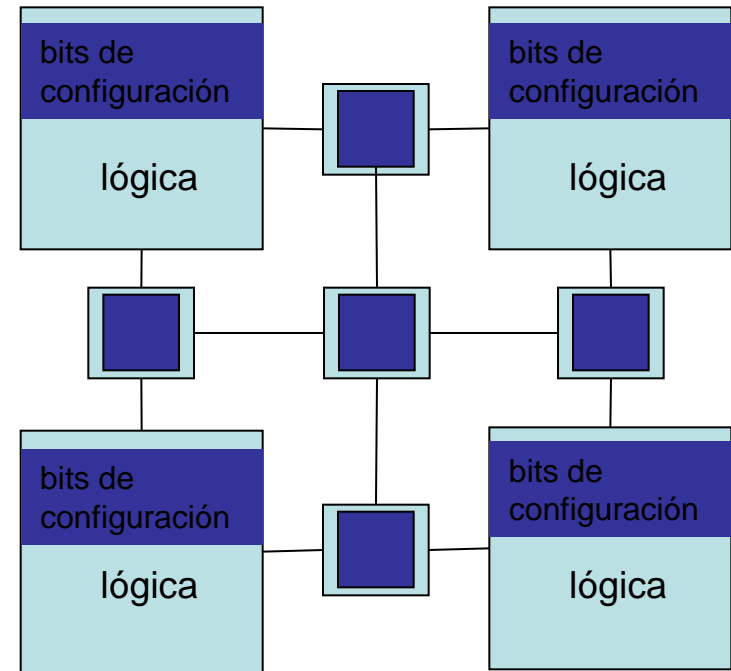
- Ninguno de los dos son circuitos dedicados. (no full custom) No están diseñados para una función específica, sino que el usuario los programa.
- Los microprocesadores implementan funciones por software. Los FPGA lo hacen por hardware.
- (pero... un procesador es, desde el punto de vista del diseñador de procesadores (Intel, por ejemplo), un full custom ASIC!!)

Microprocesadores y FPGA



Programación CPU:

- Las instrucciones se obtienen de la memoria
- Las instrucciones seleccionan operaciones complejas



Configuración FPGA:

- Bits permanecen en el dispositivo que programan
- Un bit de configuración controla un switch o un bit de lógica

Tecnologías de Programación

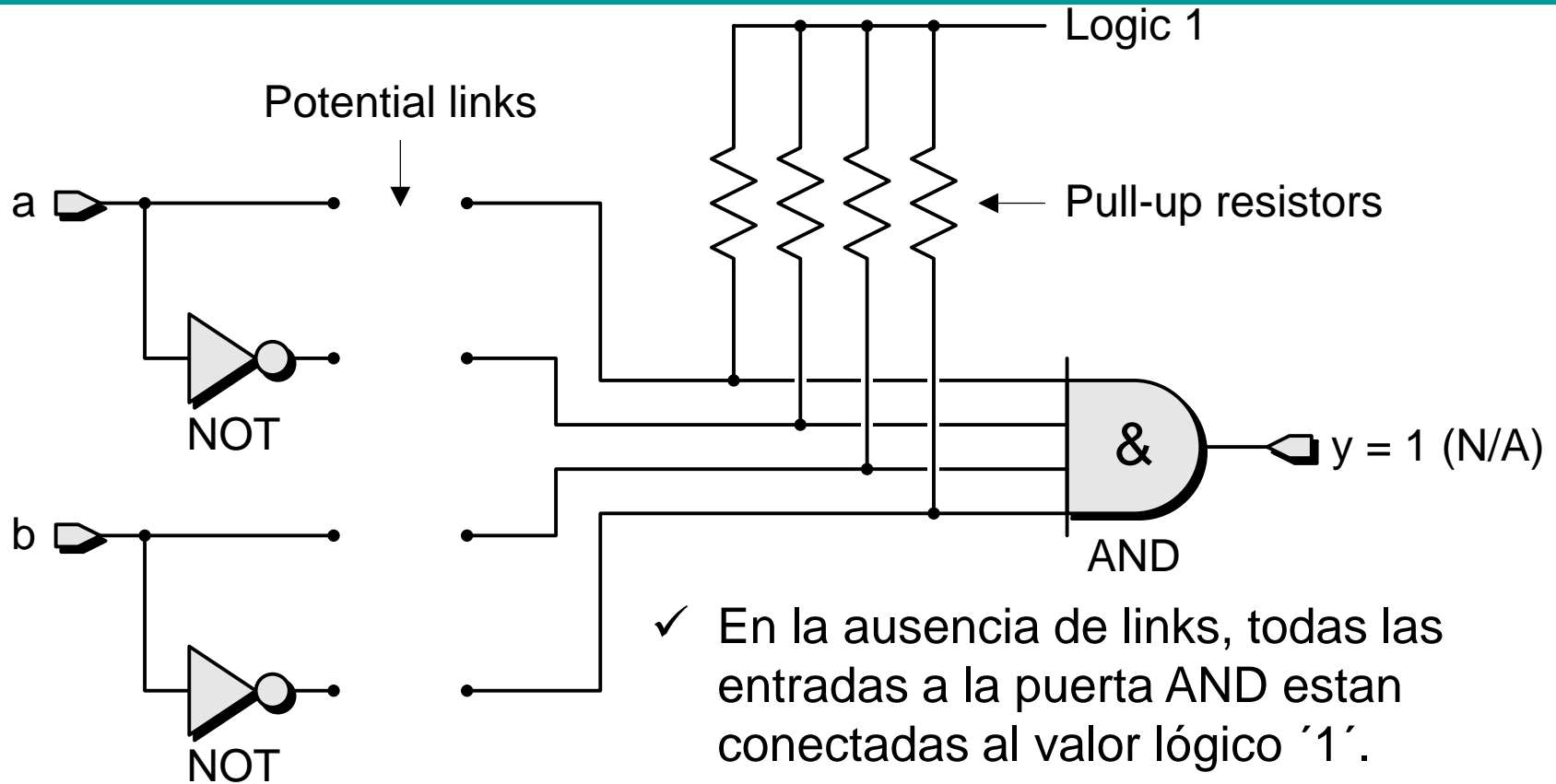
Field *Programmable* Gate Array

- Lo que caracteriza o distingue a un FPGA de un ASIC es un aspecto crucial que justamente reside en su nombre:

Field **Programmable** Gate Array

- Para poder programarlos necesitamos algún mecanismo que nos permita programar o configurar un chip de silicio preconstruido.
- Vamos a ver entonces las tecnologías de fabricación que nos van a permitir programar (configurar) estos dispositivos.

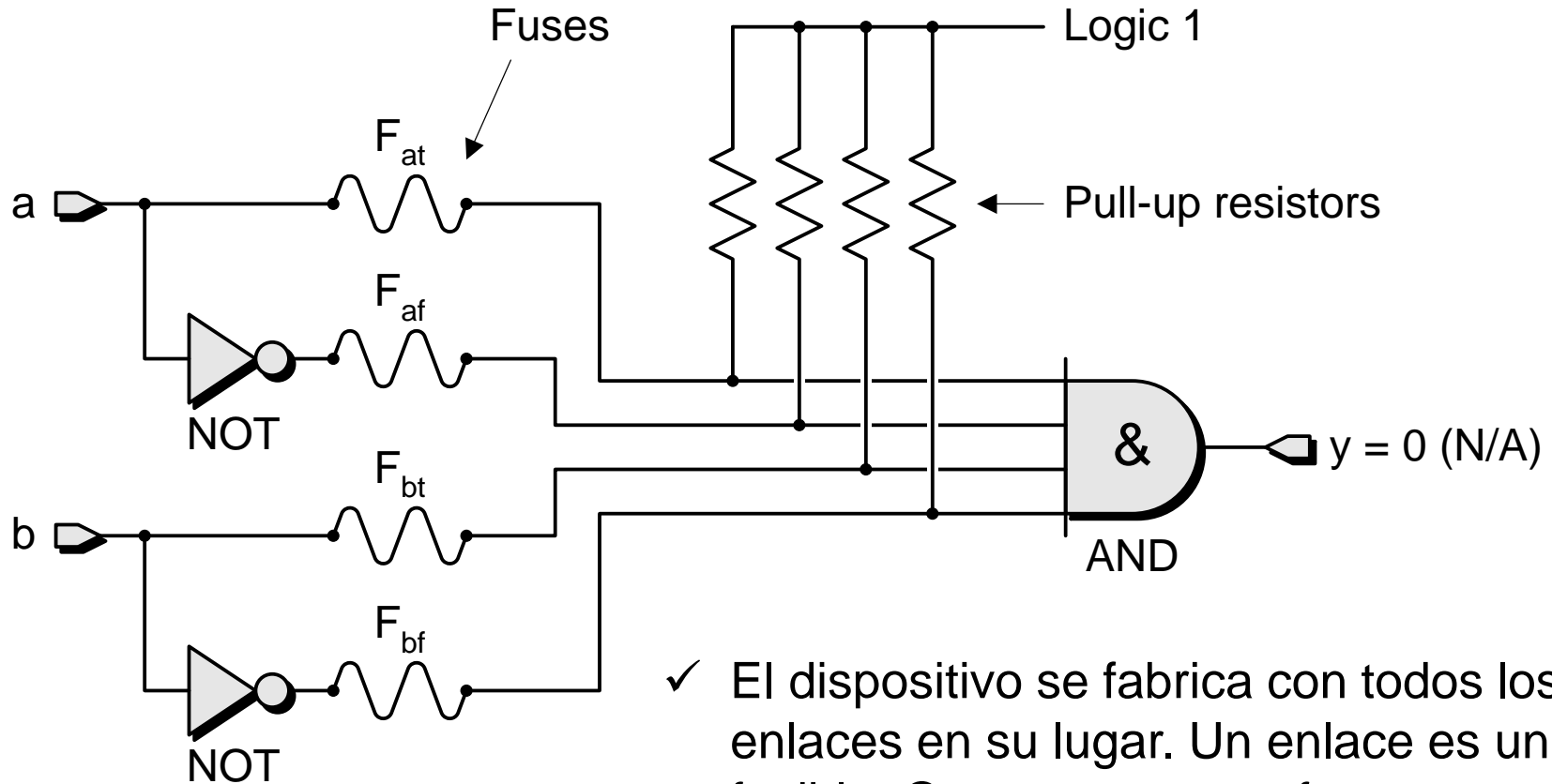
Tecnologías de Programación



- ✓ En la ausencia de links, todas las entradas a la puerta AND están conectadas al valor lógico '1'.
- ✓ Los pull-up resistors mantienen débilmente el valor lógico '1'.
- ✓ Para realizar una función hay que buscar un mecanismo que permita establecer uno o mas links.

Tecnologías de Programación:

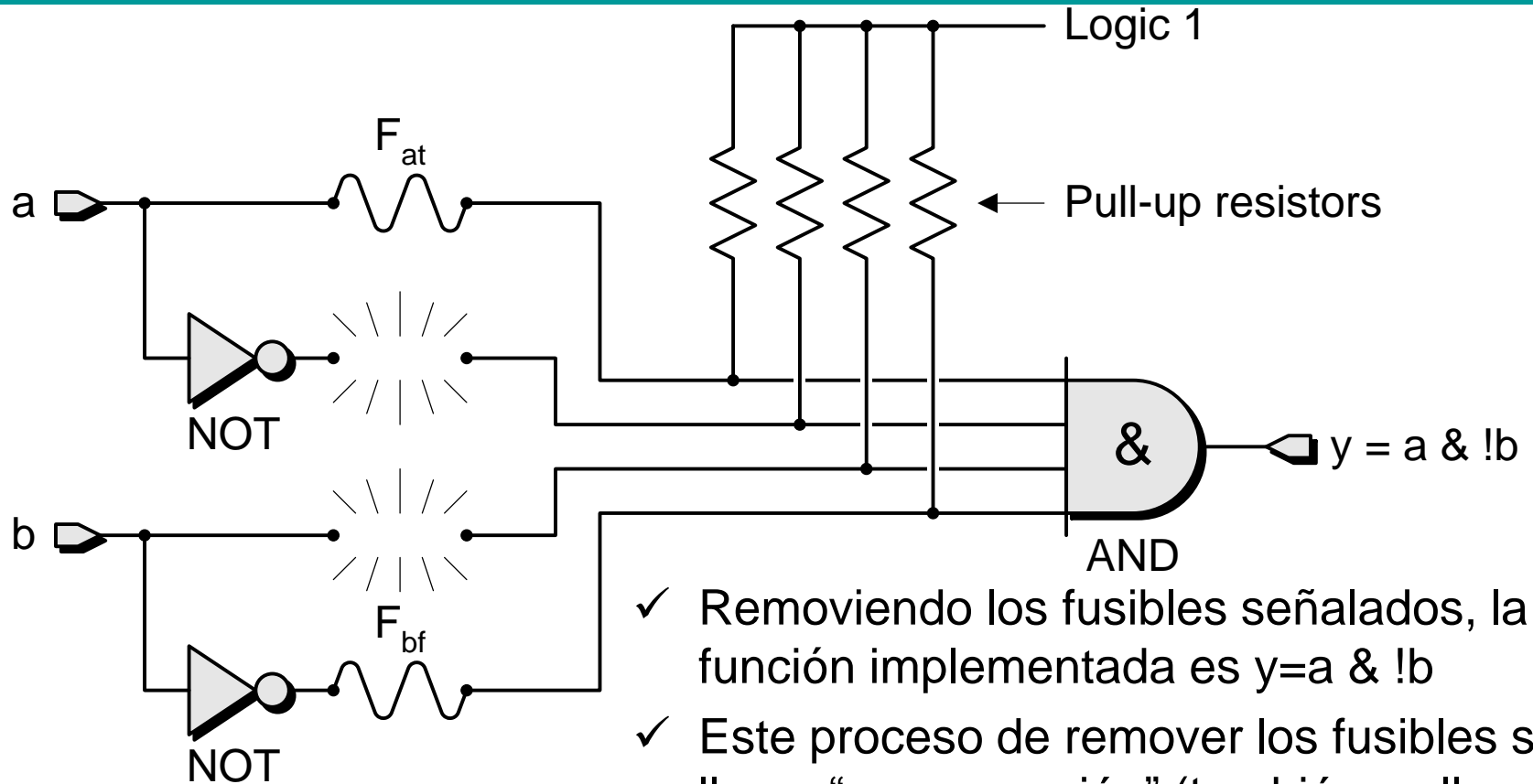
(1) Fusible Link



- ✓ El dispositivo se fabrica con todos los enlaces en su lugar. Un enlace es un fusible. O sea que, en su forma no programada, la función valdrá siempre '0'.
- ✓ Para remover los fusibles se aplican pulsos de un voltaje alto a las entradas.

Tecnologías de Programación:

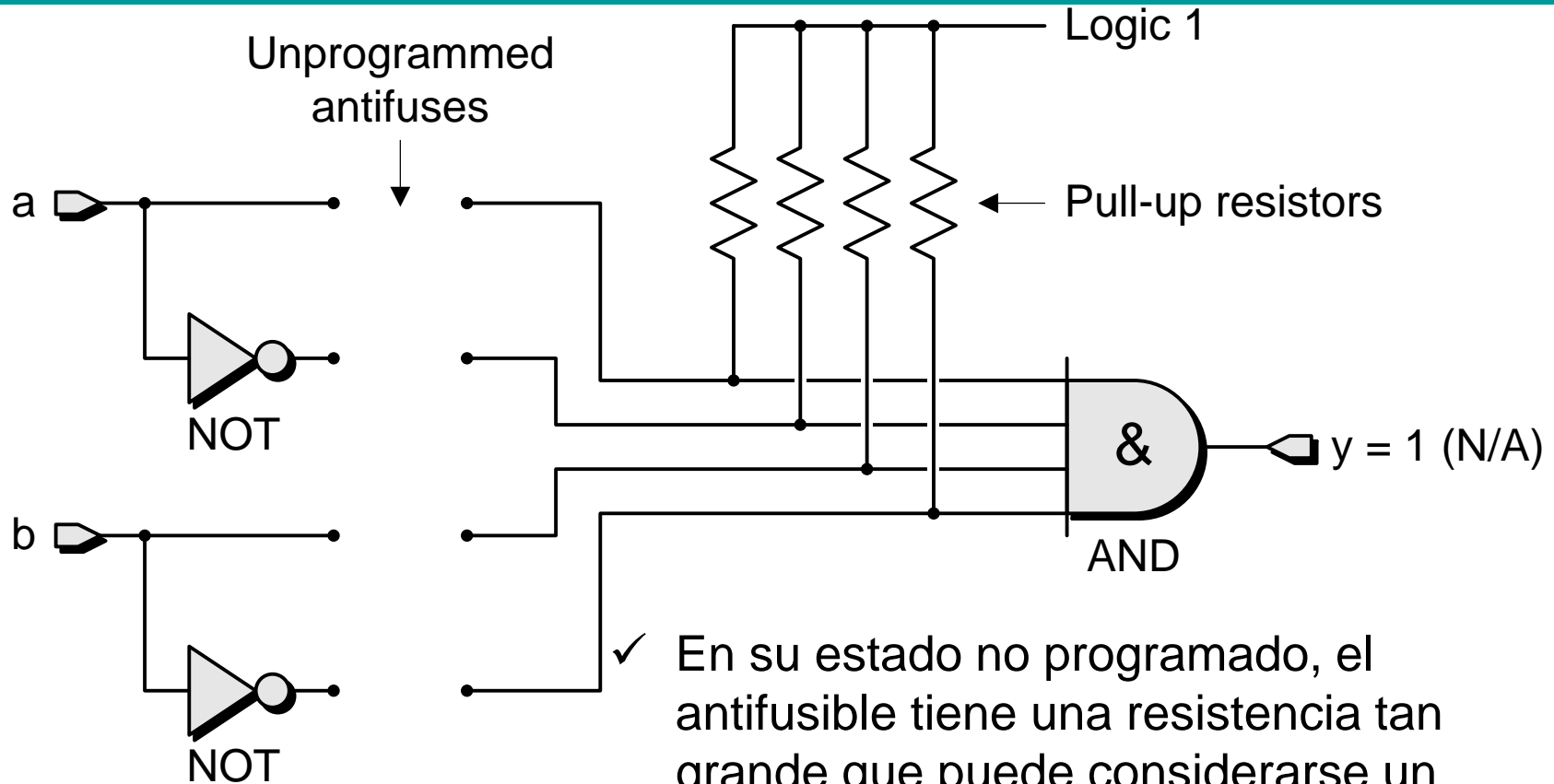
(1) Fusible Link



- ✓ Removiendo los fusibles señalados, la función implementada es $y = a \& !b$
- ✓ Este proceso de remover los fusibles se llama “programación” (también se llama “blowing” o “burning”)
- ✓ Los dispositivos son OTP, porque el fusible no puede recuperarse después de haberse quemado.

Tecnologías de Programación:

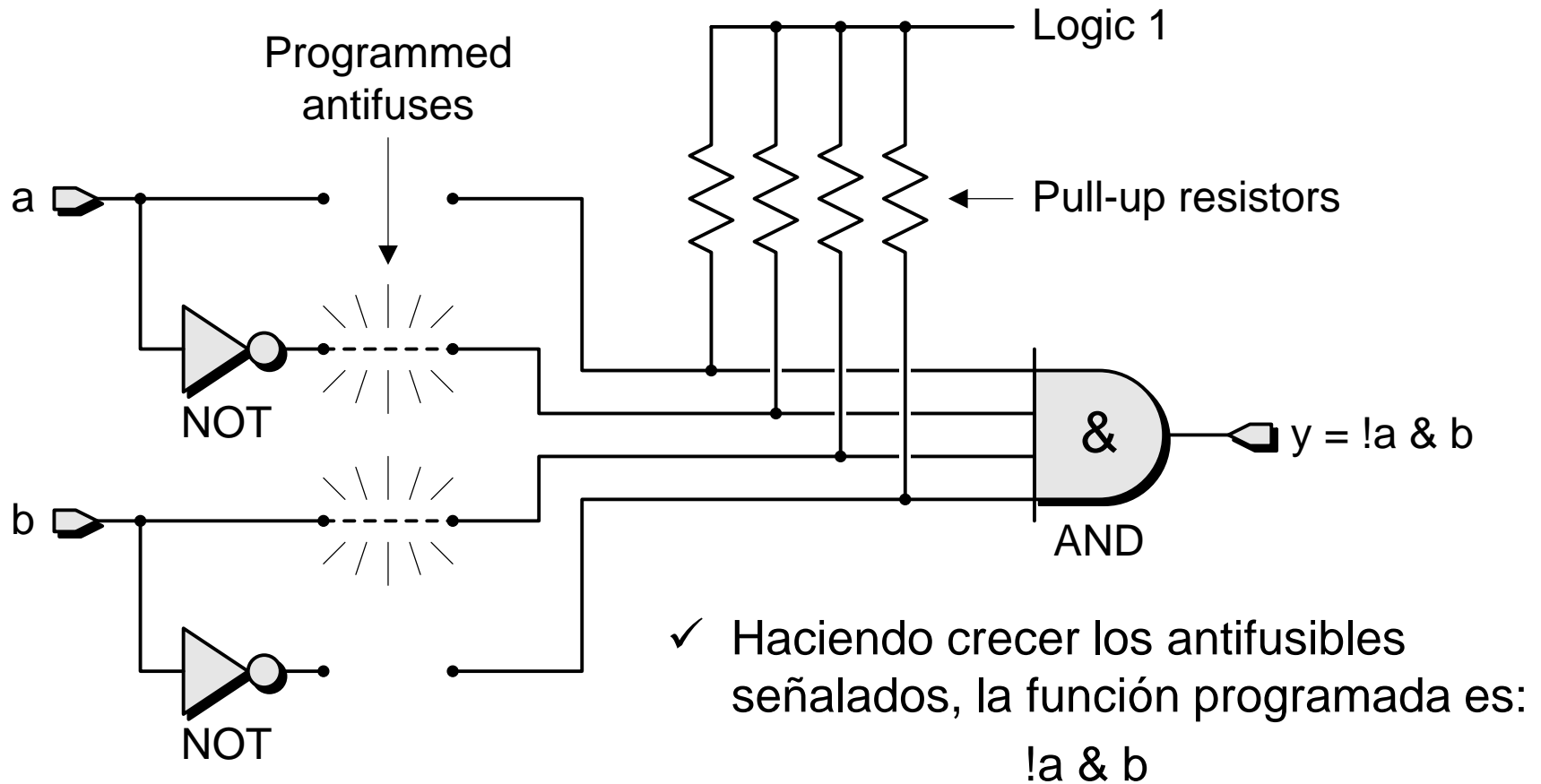
(2) Antifuse



- ✓ En su estado no programado, el antifusible tiene una resistencia tan grande que puede considerarse un circuito abierto.
- ✓ Cuando se programa, (se dice que ha sido crecido (grown)) , aplicando pulsos de alto voltaje y corriente a las entradas del dispositivo.

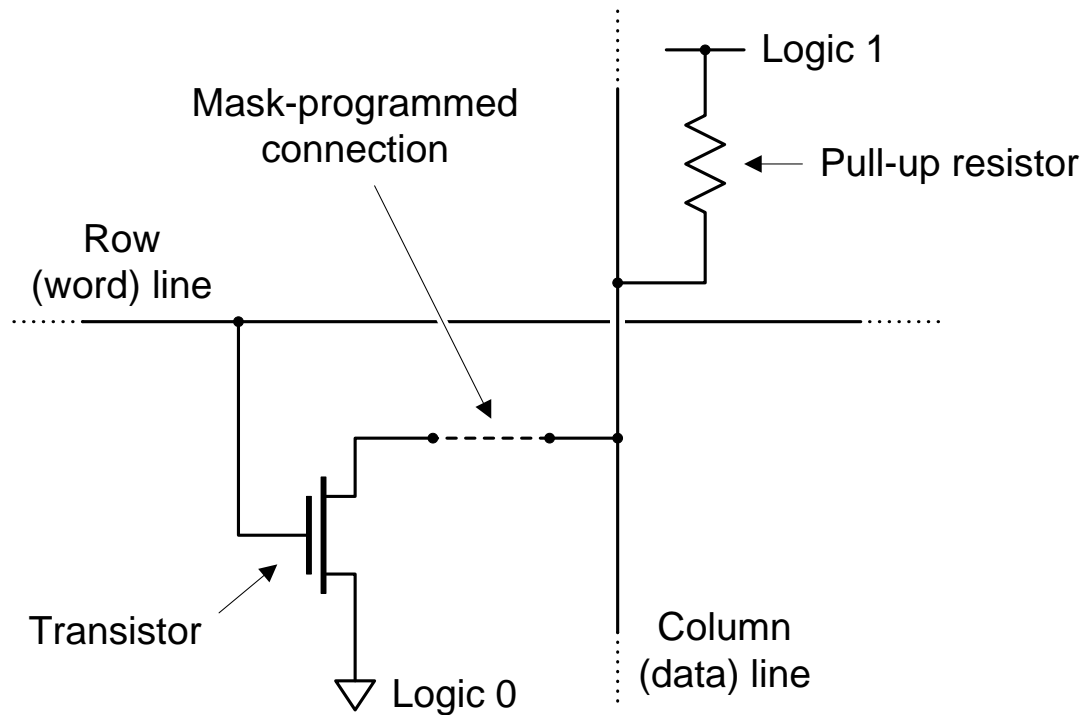
Tecnologías de Programación:

(2) Antifuse



Tecnologías de Programación:

(3) ROM

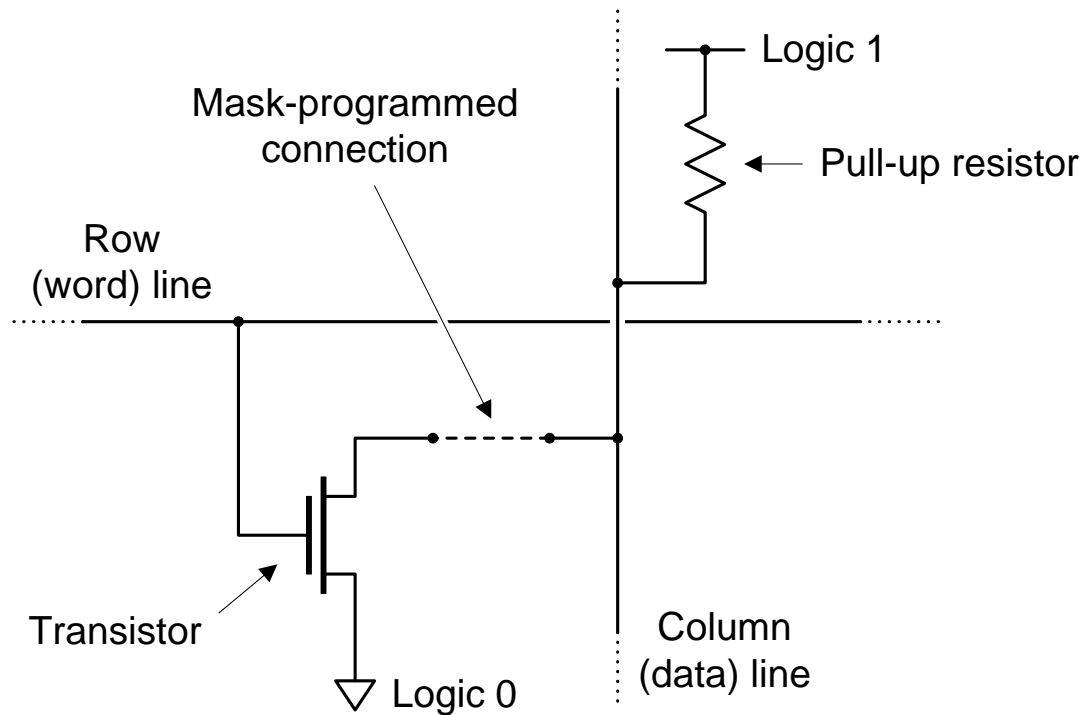


Celda de una memoria ROM

- ✓ Consiste de array de filas (row) y columnas
- ✓ Cada columna tiene un único pull-up que intenta mantener a “1” esa columna
- ✓ Cada intersección fila/columna tiene un transistor y una “conexión” potencial
- ✓ La ROM se preconstruye y la misma arquitectura puede usarse para múltiples clientes.

Tecnologías de Programación:

(3) Mediante Máscara (ROM)

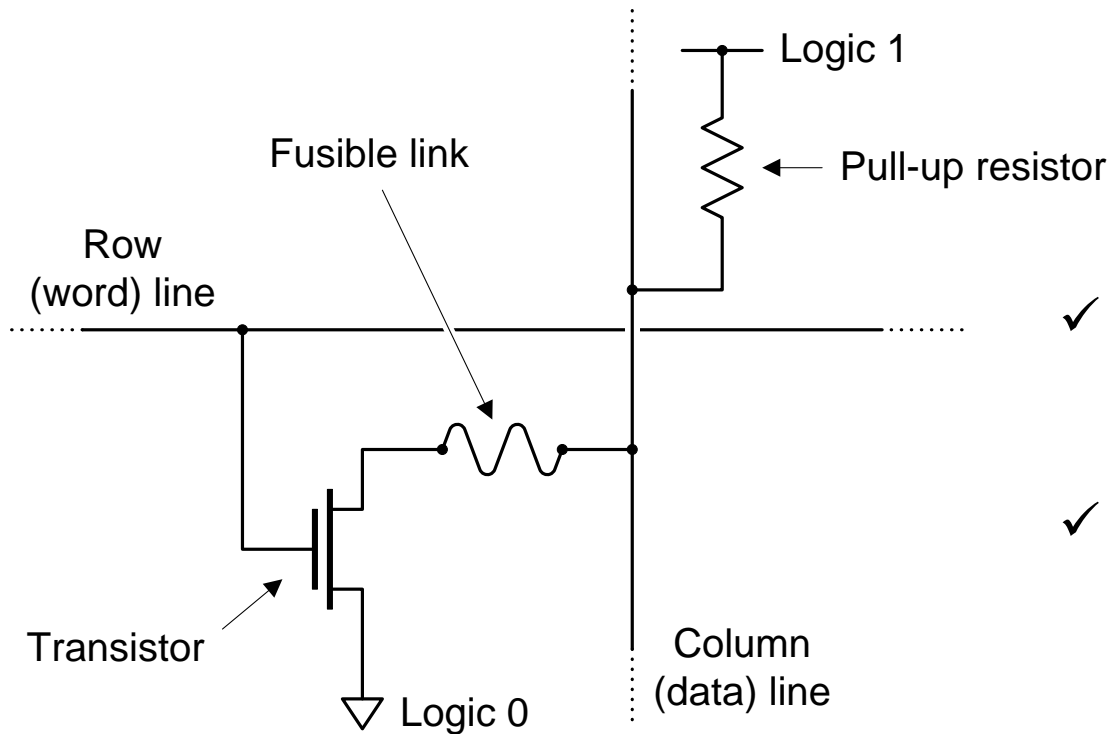


Celda de una memoria ROM

- ✓ Se pre-construyen y, para adaptarlas a los requerimientos del cliente se utiliza una máscara fotográfica para definir cuales celdas tendrán o no una conexión programada.
- ✓ Si la línea de fila se activa, el transistor se activa y :
 - ✓ Si hay conexión, en la columna aparece el valor lógico 0
 - ✓ Si no hay conexión, en la columna sigue el valor '1' del pullup.

Tecnologías de Programación:

(4) PROM

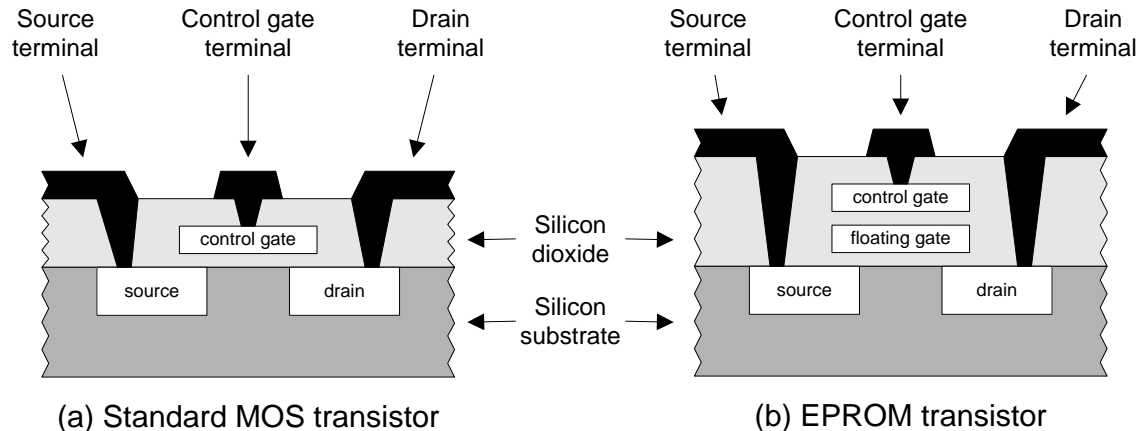


Celda de una memoria PROM

- ✓ Problema de los dispositivos programados con máscara: son caros! Se hacen en la fábrica y solo salen a cuenta si son muchísimos
- ✓ Programmable ROM (1970) están basados en la tecnología de fusible link.
- ✓ En su estado no programado, tal como se compra, todos los enlaces están presentes. O sea, si la línea se activa, la columna conduce '0'.
- ✓ La programación al remover los enlaces, hace que la celda almacene un '1'.

Tecnologías de Programación:

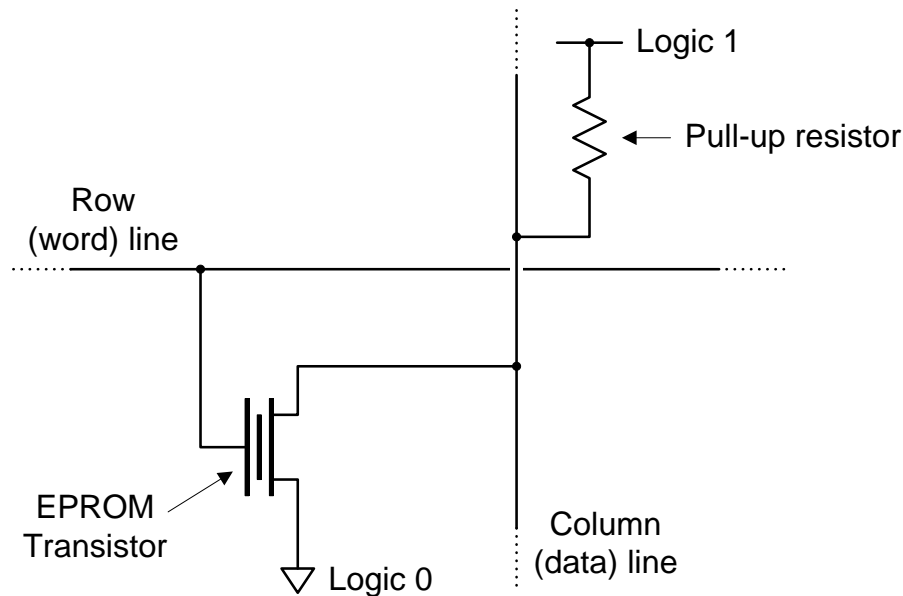
(5) EPROM



- ✓ Problema con las tecnologías basadas en fusible links y máscaras
➔ son OTP.
- ✓ Erasable Programmable EPROM (1971) : los transistores tienen una puerta adicional de polisilicio : puerta flotante
- ✓ En su estado no programado, la puerta flotante no está cargada y no afecta el normal funcionamiento del transistor.
- ✓ Al programar el transistor, se carga la puerta flotante, inhibiendo la normal operación del transistor, y distinguiendo aquellas celdas que han sido programadas, de las que no lo han sido.

Tecnologías de Programación:

(5) EPROM

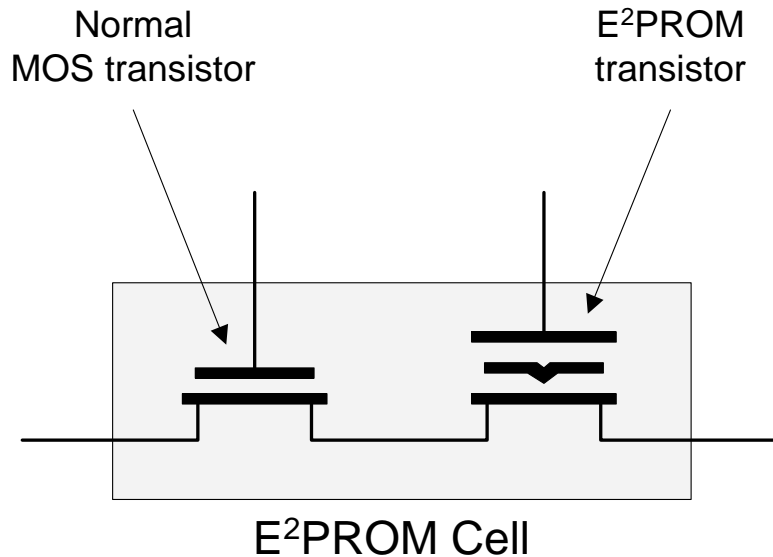


- ✓ Para borrar la EPROM hay que quitarla del circuito.
- ✓ Problemas: mucho tiempo para ser borradas (20'). Cuanta mas integración, se necesita mas radiación → mas tiempo de exposición.

- ✓ En este caso, no es necesario el fusible.
- ✓ En su estado no programado, tal como se compra, todas las puertas flotantes están descargadas. O sea, si la línea se activa, se activa el transistor y la columna conduce '0'.
- ✓ La programación, al cargar la puerta flotante, inhibe la operación del transistor, por lo tanto la columna conduce '1'.
- ✓ Para descargar esa puerta, se utiliza radiación ultravioleta.

Tecnologías de Programación:

(6) EEPROM y FLASH



- ✓ Electrically Erasable Programmable ROM
- ✓ Necesita dos transistores, el normal se utiliza para el borrado.
- ✓ Son 2,5 veces mas grandes que los EPROM.

- ✓ FLASH: borran mas rápido que EPROM.
- ✓ Usan diversas arquitecturas, pero todas permiten ser borradas eléctricamente. Estas arquitecturas con similares de las EEPROM.

Tecnologías de Programación:

(7) SRAM

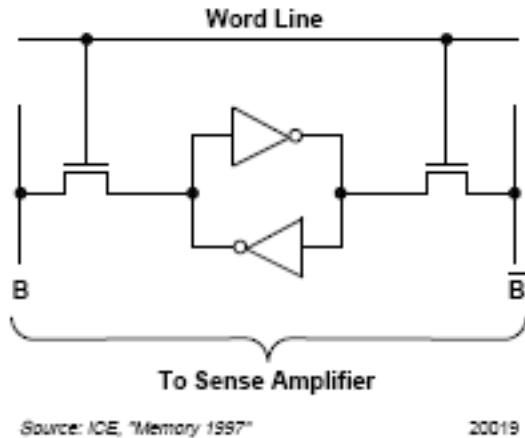
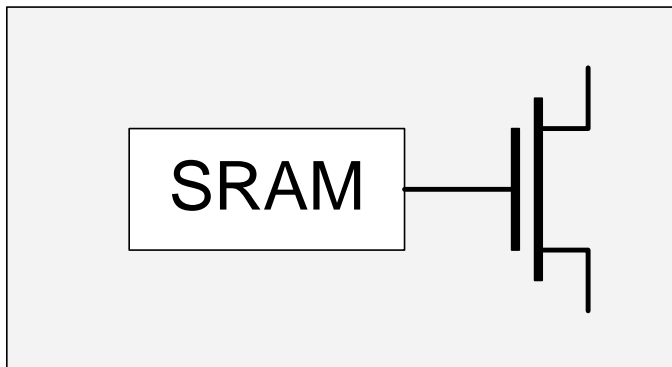


Figure 8-3. SRAM Cell



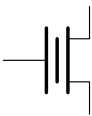
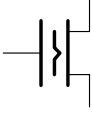
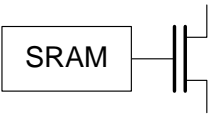


- ✓ Es un multitransistor formado por 4 a 6 transistores configurados como un latch. Dos de los seis transistores controlan el acceso al latch.
- ✓ Cuando la celda no se direcciona, los dos transistores de control están cerrados y los datos se mantienen dentro del latch.
- ✓ Consumen mucha área
- ✓ Pierden la información cuando dejan de ser alimentados.
- ✓ Pueden ser reprogramados rápidamente y repetidamente.
- ✓ Para escribir un "1", se pone $B=1; \bar{B}=0$ y se activa WordLine

Tecnologías de Programación: (8)

La tecnología avanza... próximamente....
MRAM.... (magnetic RAM) quien sabe...

Tecnologías de Programación

Technology	Symbol	Predominantly associated with ...
Fusible-link		SPLDs
Antifuse		FPGAs
EPROM		SPLDs and CPLDs
E ² PROM/ FLASH		SPLDs and CPLDs (some FPGAs)
SRAM		FPGAs (some CPLDs)

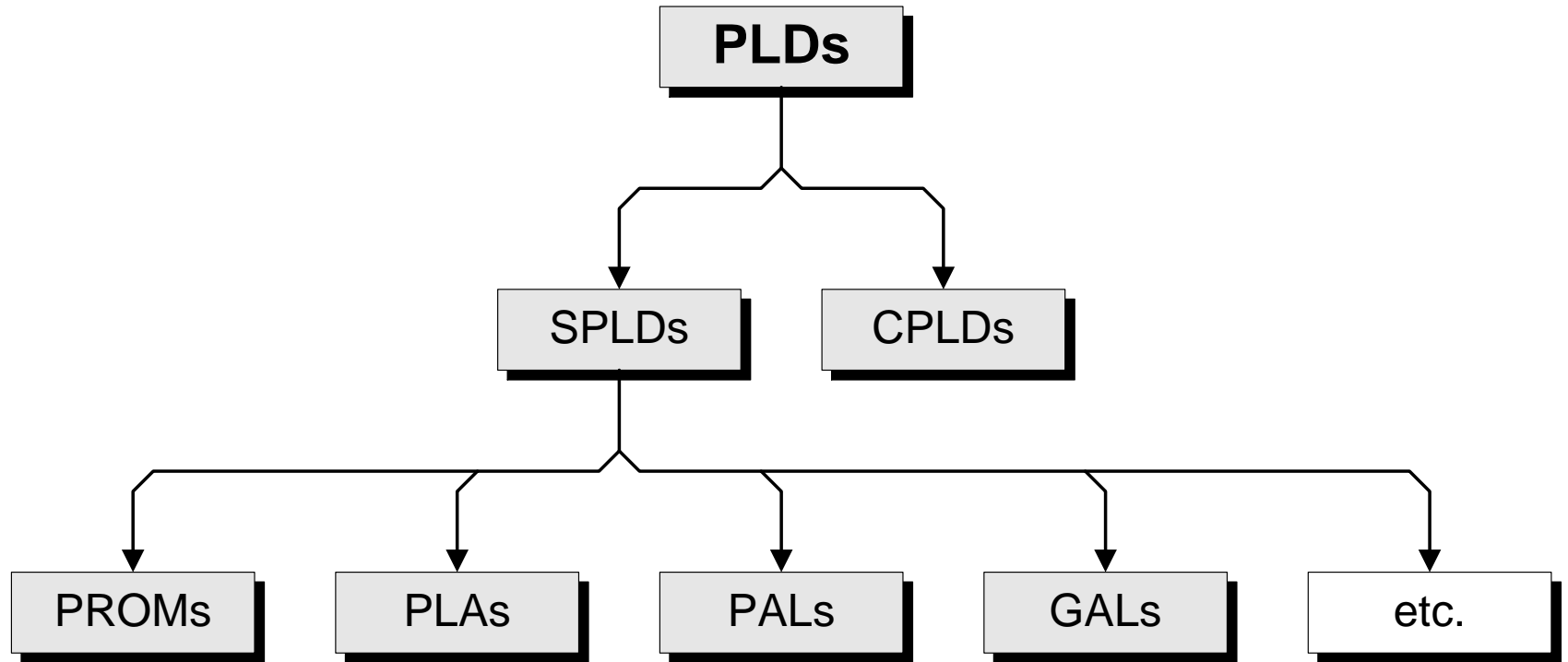
Tecnologías de Programación

Feature	SRAM	Antifuse	E2PROM / FLASH
Technology node	State-of-the-art	One or more generations behind	One or more generations behind
Reprogrammable	Yes (in system)	No	Yes (in-system or offline)
Reprogramming speed (inc. erasing)	Fast	----	3x slower than SRAM
Volatile (must be programmed on power-up)	Yes	No	No (but can be if required)
Requires external configuration file	Yes	No	No
Good for prototyping	Yes (very good)	No	Yes (reasonable)
Instant-on	No	Yes	Yes
IP Security	Acceptable (especially when using bitstream encryption)	Very Good	Very Good
Size of configuration cell	Large (six transistors)	Very small	Medium-small (two transistors)
Power consumption	Medium	Low	Medium
Rad Hard	No	Yes	Not really

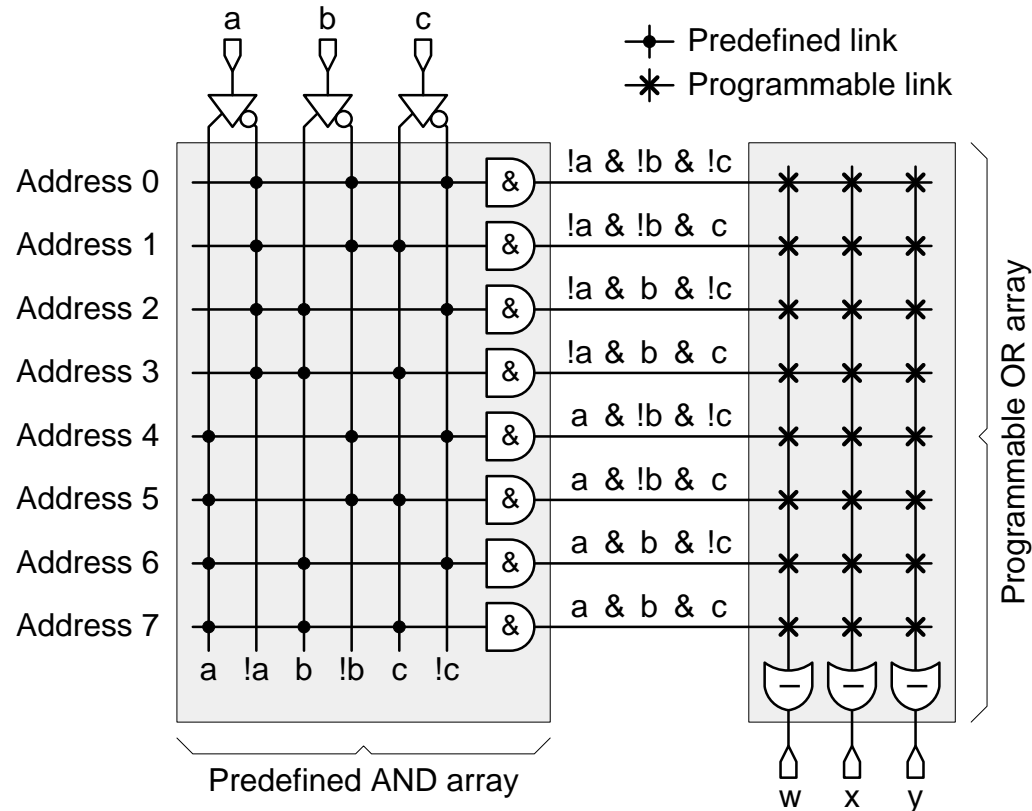
Origen de los FPGA PLD's

Programmable Logic Device

- Dispositivos cuya arquitectura interna está predeterminada por el fabricante, pero pueden ser configurados por los ingenieros “en el campo” para realizar una variedad de funciones.
- Pocas puertas lógicas → funciones simples

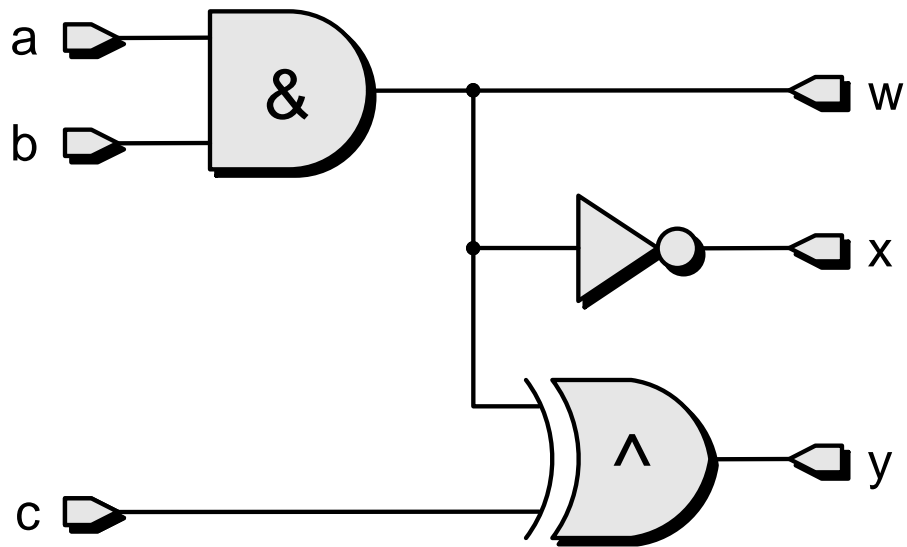


PROMs (1970)



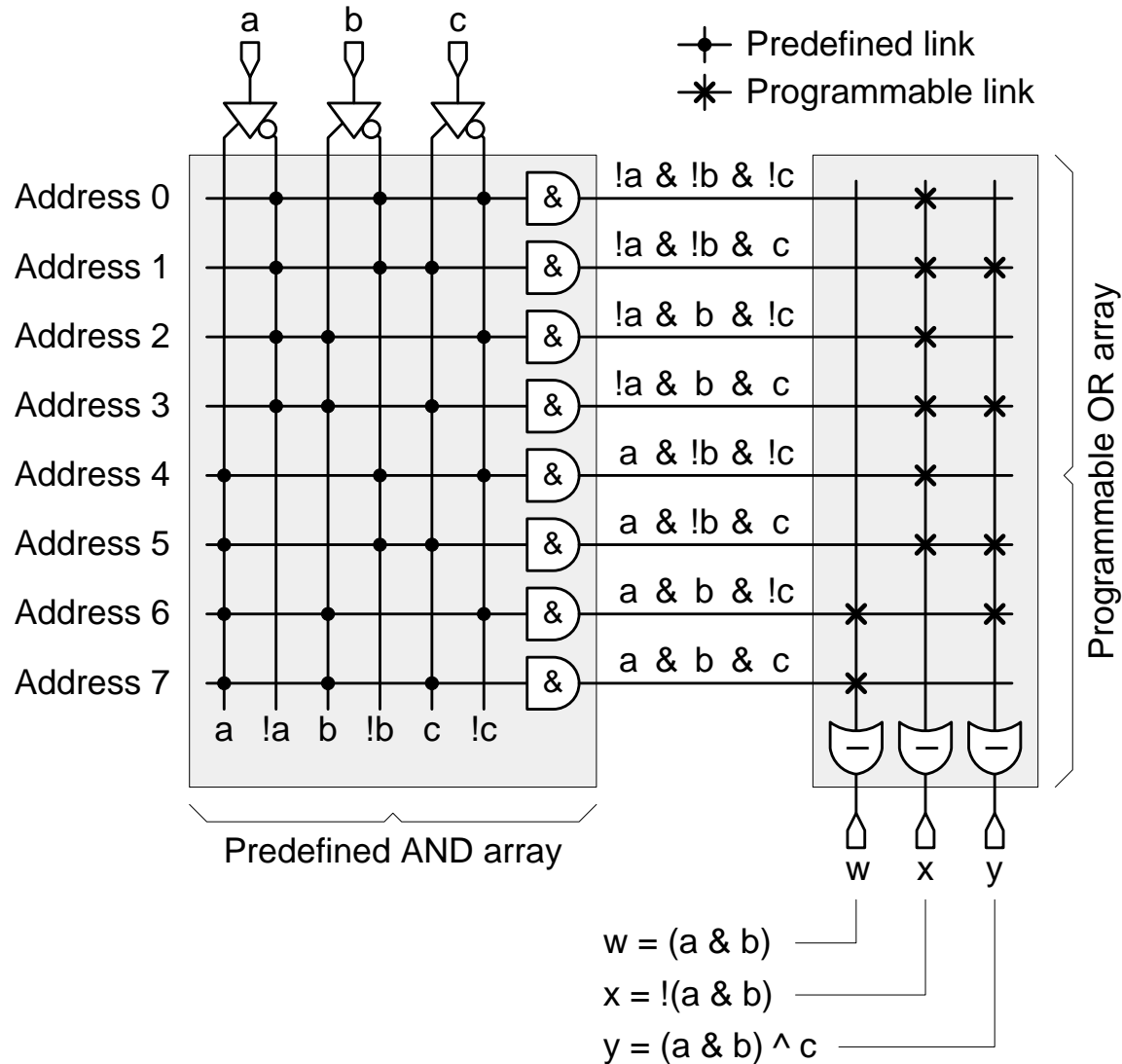
- ✓ La programación se puede realizar con cualquiera de las tecnologías vistas (fusibles, EPROM, EEPROM).
- ✓ Solo es configurable la matriz OR.
- ✓ Útiles para ecuaciones con pocas entradas y muchos términos producto.

Implementación de una función lógica combinacional

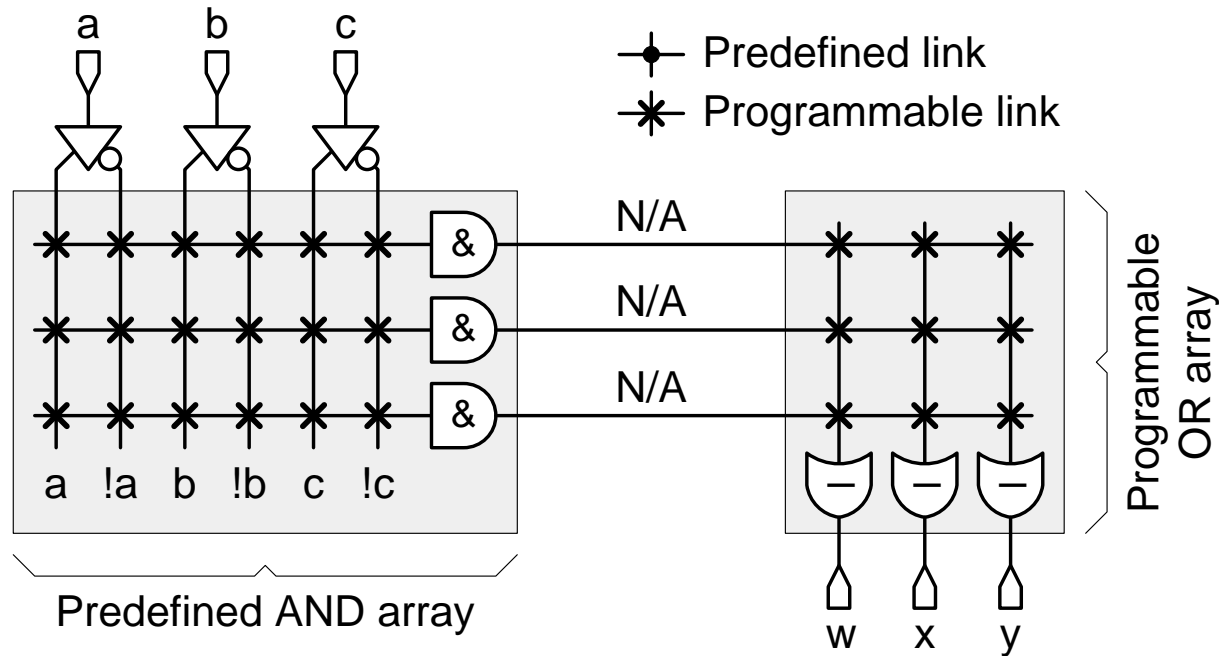


<i>a</i>	<i>b</i>	<i>c</i>	<i>w</i>	<i>x</i>	<i>y</i>
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Implementación con PROM

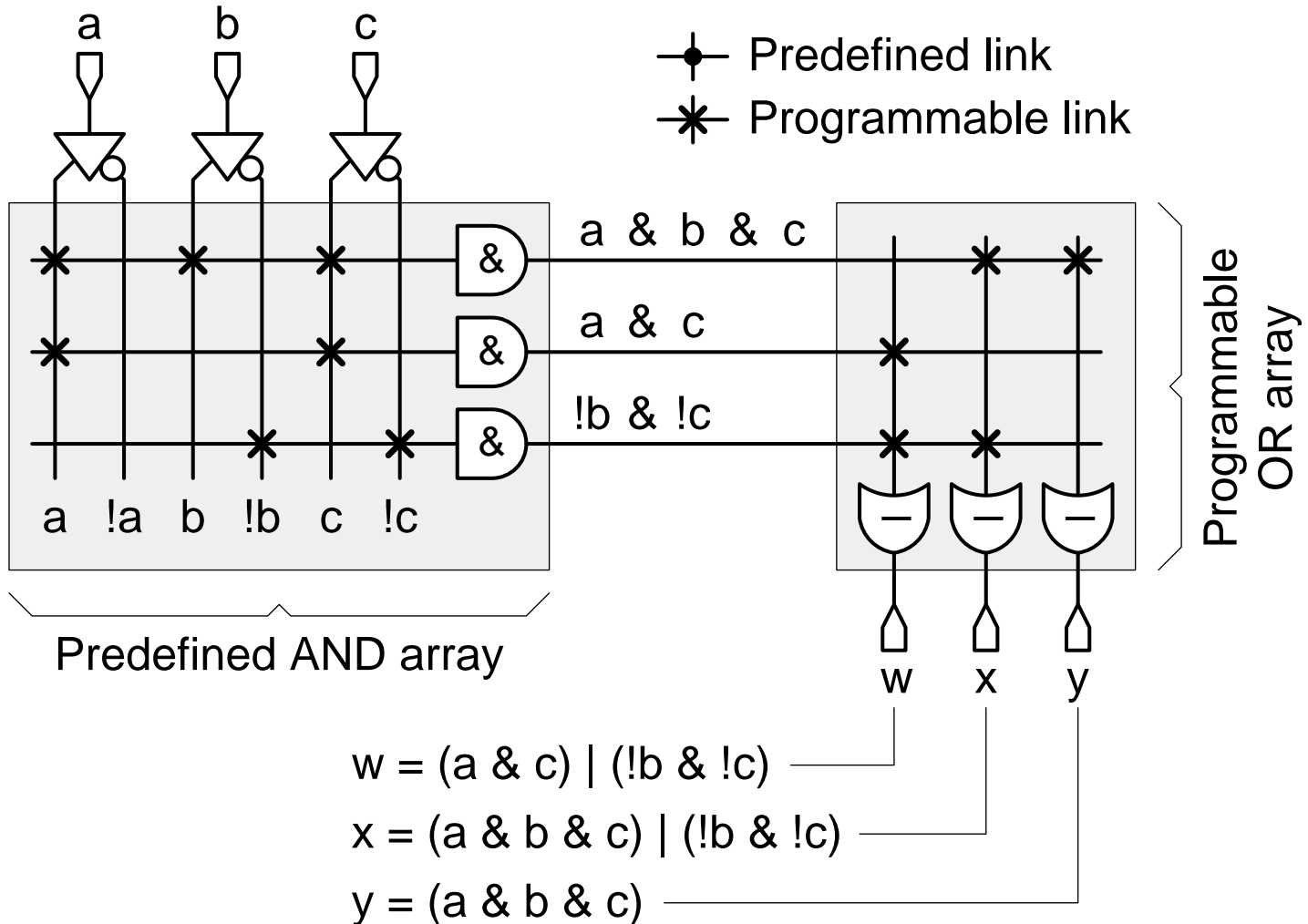


PLA (Programmable Logic Array)

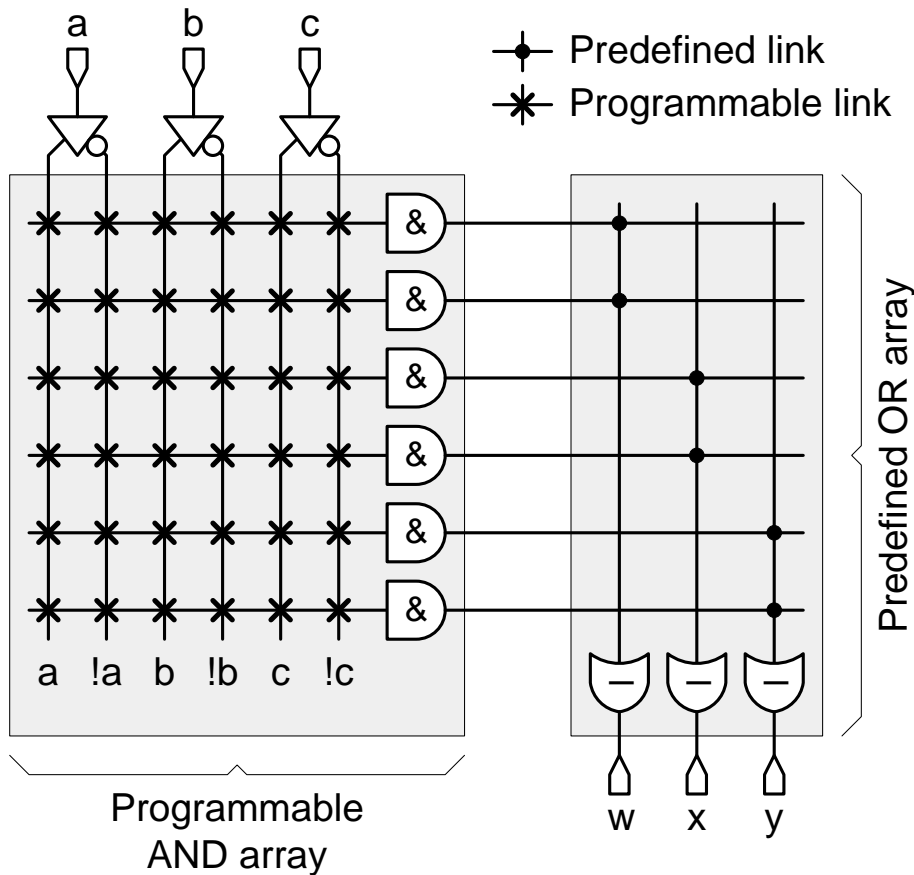


- ✓ Disponible a partir de 1975, se pueden programar los dos arrays.
- ✓ Se hicieron algunas variantes: arrays AND con arrays NOR. No mucho éxito en el mercado
- ✓ Son útiles cuando diversas funciones usan o comparten términos producto.
- ✓ Son mas lentas que las PROMS

Implementación con PLA

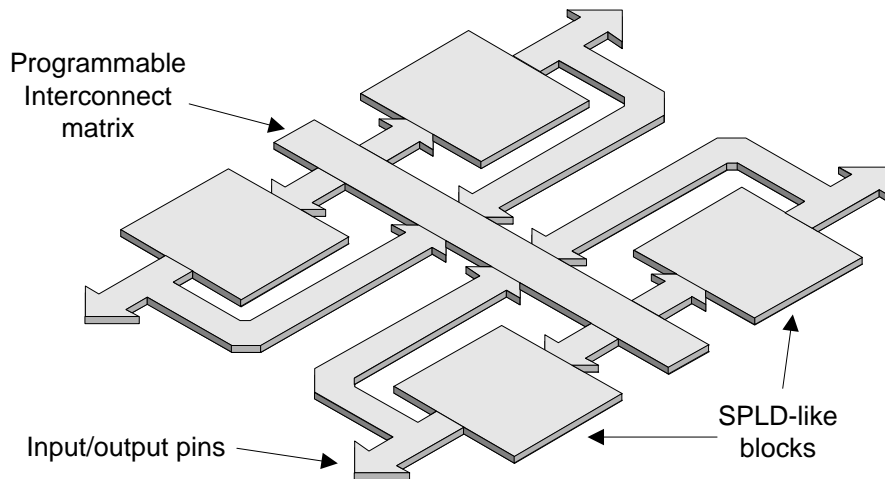


PAL (Programmable Array Logic)



- ✓ Al revés de las PROM, la parte programable es la matriz AND
- ✓ Las GAL (Generic Array Logic) son variaciones de las PAL, mas sofisticados (EE)
- ✓ Todos estos dispositivos, aparecen en el mercado con una variedad de opciones: inversión de las salidas, salidas triestado, salidas registradas, etc. Además de tener un número mas grande de entradas y salidas.

CPLD's (vamos llegando)



- A finales de los 70, los inventores de la PAL, introducen el Mega-Pal, dispositivo con 4 Standard Pals interconectadas de alguna manera. No funcionó. Consumía mucho.
- 1984: Altera (nueva empresa) introduce el CPLD basado en tecnología CMOS y EPROM.
- Las conexiones entre los bloques se programan mediante la matriz de interconexión.

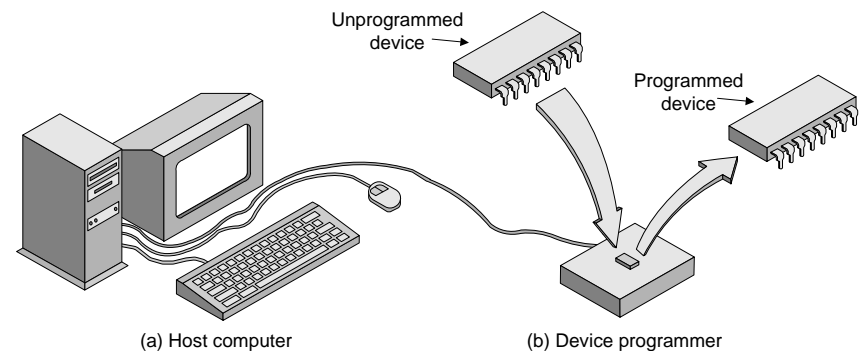
Proceso de Diseño

- ¿Como se diseñaba un circuito usando PLD's?

Se realizaba un esquemático o bien un diagrama de estados, todo en papel porque no existían las herramientas que hoy conocemos.

El esquemático se convertía del papel a una tabla (con un formato especial, requerido por el dispositivo programador). Esta tabla indicaba que conexiones se debían programar. La tabla se tipeaba en un computadora y se bajaba luego al dispositivo programador.

Cada vendedor de PLD's, obviamente, había desarrollado su propio formato para el archivo, que solo servía para sus dispositivos.



Ensambladores y Herramientas de Diseño

- 1980: JEDEC (Joint Electron Device Engineering Council) propone un formato standard para los archivos de configuración de los PLD's.
- John Birkner, creador de las PAL, crea PALASM el primer ensamblador para PAL. Es un primitivo HDL (Hardware Description Language) y además una aplicación software. Es usado para trasladar expresiones booleanas y grafos de estados a una tabla para PAL. PALASM solo es para PAL's fabricadas por MMI (Monolithic Memories Inc). No realiza minimizaciones o optimaciones.
- 1983: ABEL (Advanced Boolean Expression Language) y CUPL (Common Universal tool for Programmable Logic). Ambas herramientas trabajan con diversos tipos de PLD's y de fabricantes.
- Estas herramientas y HDL son las precursoras de VHDL y Verilog, lenguajes HDL de alto nivel y herramientas que son usadas hoy para los ASIC's y FPGA.

PALASM

[illegible]

PALASM

opcional

```
TITLE Example
PATTERN Simple
REVISION A
AUTHOR ECEN 220
COMPANY BYU
DATE March 1, 1995
```

obligado

```
CHIP Example PAL20L8
;PINS
; 1  2  3  4  5  6  7  8  9  10 11 12
  B  A  C  D  NC NC  NC NC  NC NC  NC GND

; 13 14 15 16 17 18 19 20 21 22 23 24
  NC NC NC  NC NC  NC NC  NC  Z  /X NC VCC
```

Equaciones booleanas

```
EQUATIONS
X=/A*B + A*B + A*/B + /A*/B*C
Z=/A*B + A*B + A*/B + /A*/B*C
```

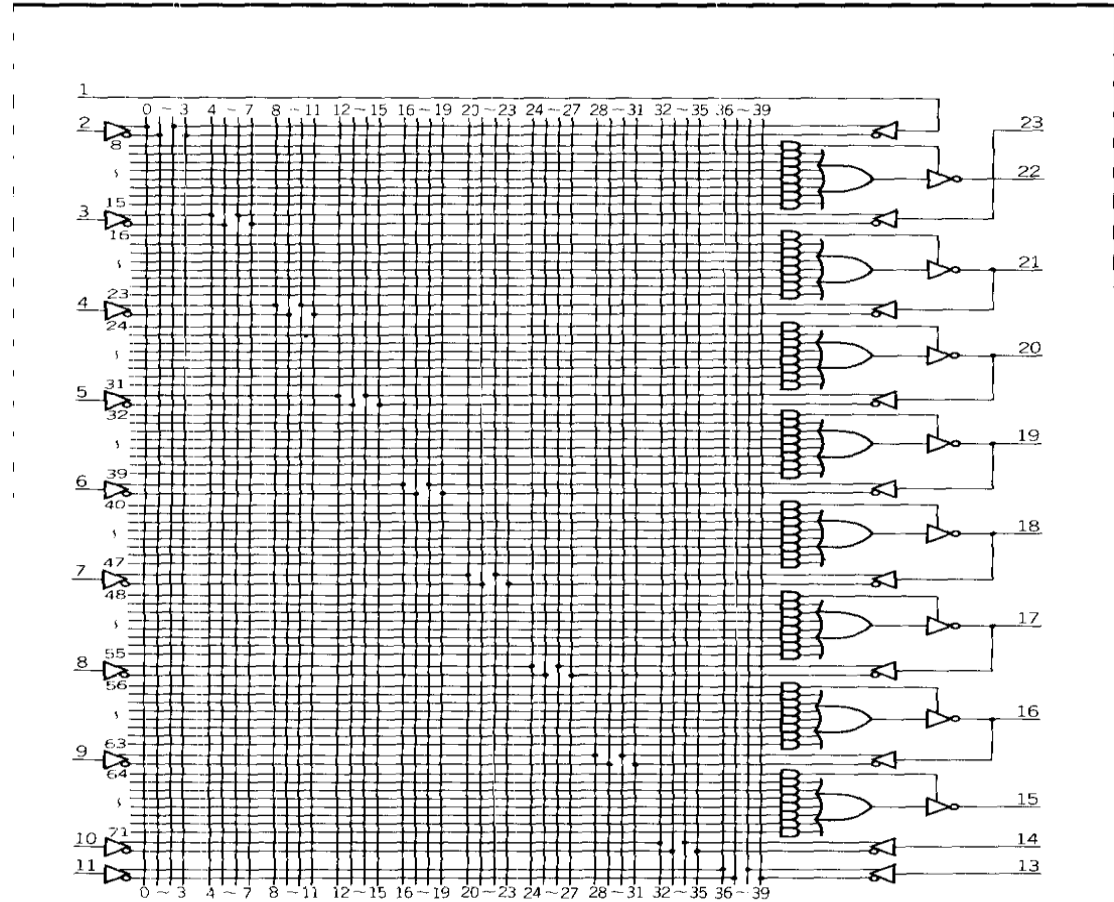
- Trace define que señales mostrará el simulador.
- SETF define el valor de las variables en cada simulación.

```
SIMULATION
TRACE-ON A B C /X Z
SETF /A /B /C
SETF /A /B C
SETF /A B /C
SETF /A B C
SETF A /B /C
SETF A /B C
SETF A B /C
SETF A B C
TRACE-OFF
```

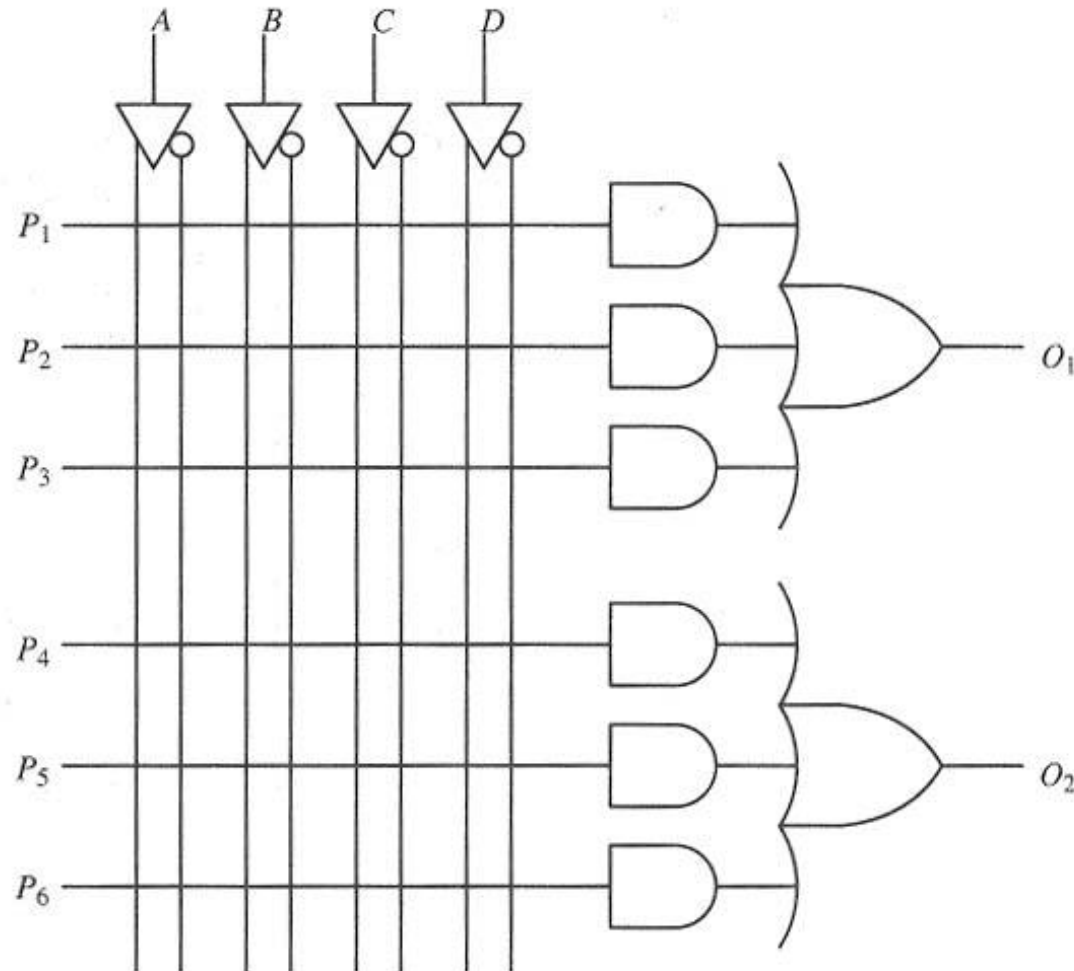
PAL20L8

- Entradas (20) :
1..11; 13-14; 16..23
- Salidas (8): 15..22
- Algunas salidas (16..22) son triestado. Es decir, pueden ser configuradas tanto como entradas como salidas.
- Todas las salidas son negadas.
- Pin 12 es ground, Pin 24 es Vcc.

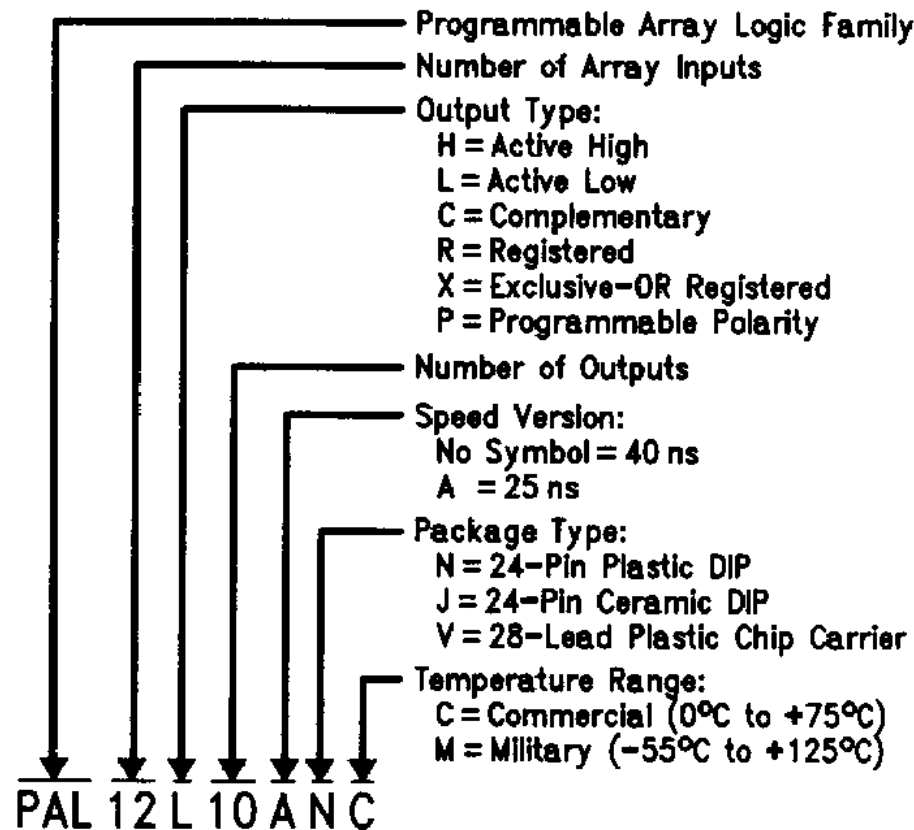
PAL20L8



Esquema de la PAL

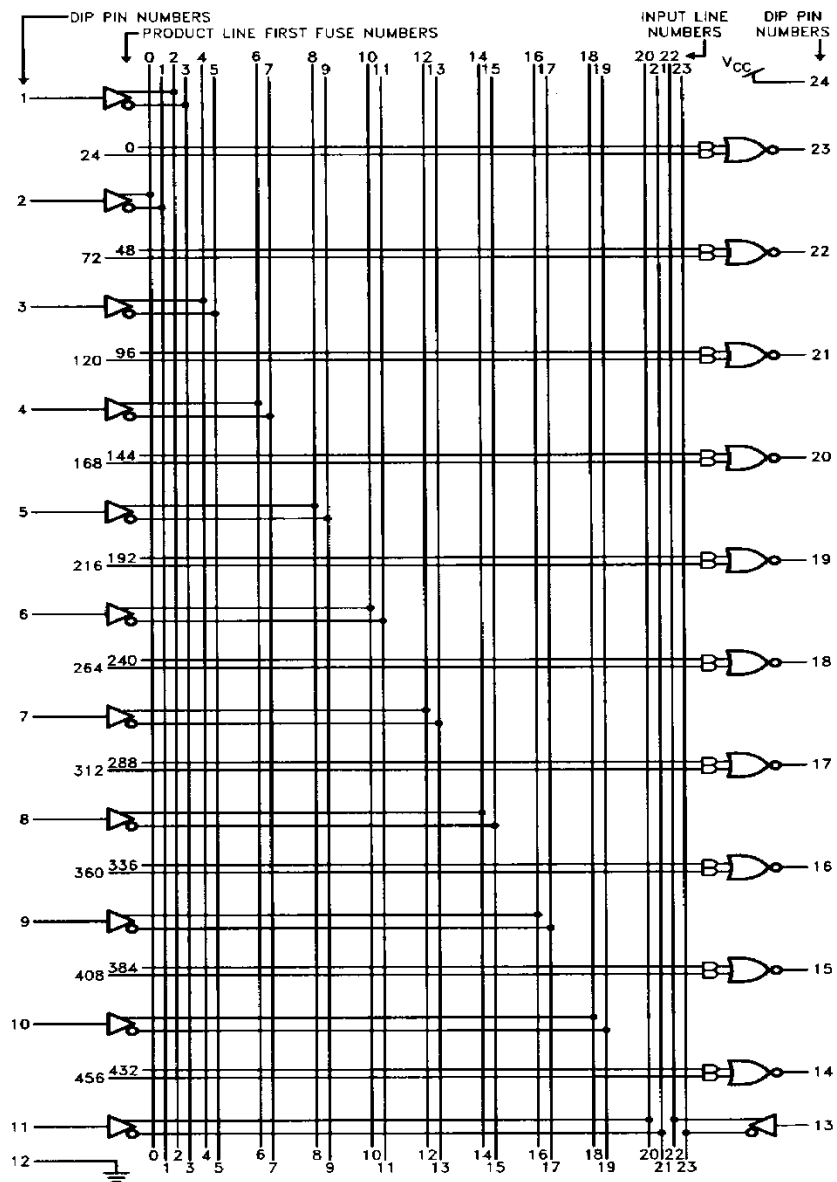


24-Pin Small PAL Family



PAL 20 L 8

Logic Diagram PAL12L10



Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

TL/L/9997-16

PALASM: archivo .xpt

- Después de creado el archivo extensión .pds se ensambla. El ensamblador genera dos archivos:
 - .xpt: contiene el fuse map
 - .trf es el archivo para simulación
- x se usa para denotar un contacto quemado.
- Columnas 0,1,2,3,4 5 corresponden a las entradas y sus complementarios.
- Filas 0,1,2,3 corresponden a la función \bar{x} (sus términos producto y su habilitación)
- Filas 8 y 9 corresponden a la función Z. Sus términos producto y su habilitación.
- Fijense que cada OR puede admitir hasta 7 términos producto (pues uno de ellos se reserva para la habilitación)

```

PALASM90 version 1.1 PAL ASSEMBLER      - MARKET RELEASE (8-27-90)
(C) - COPYRIGHT ADVANCED MICRO DEVICES INC., 1990

TITLE      :Example                      AUTHOR :ECEn 220
PATTERN    :Simple                      COMPANY:BYU
REVISION:A                                DATE   :March 1, 1995

PAL20L8
EXAMPLE
;
;      --  -
;      AAB  CC
;
;      11  1111  1111  2222  2222  2233  3333  3333
;      0123 4567 8901 2345 6789 0123 4567 8901 2345 6789
;
0  ----  ----  ----  ----  ----  ----  ----  ----  ----
1  X----  ----  ----  ----  ----  ----  ----  ----
2  ----  X----  ----  ----  ----  ----  ----  ----
3  --X-  ----  ----  ----  ----  ----  ----  ----
;
8  ----  ----  ----  ----  ----  ----  ----  ----
9  -X-X  -X--  ----  ----  ----  ----  ----  ----
;
SUMMARY
-----
TOTAL FUSES BLOWN      = 234

```

PALASM: archivo .xpt

- Interpretación de los términos producto.

- La función $X = \neg A \cdot B + A \cdot \neg B + A \cdot B \cdot C$

Simplificada es :

$$\neg X = A + B + C$$

Comprobarlo!!!! (Pista: Usar Mapas de Karnaugh o bien deducirlo de la Tabla de Verdad).

- La función $Z = \neg A \cdot B + A \cdot \neg B + \neg A \cdot B \cdot C$

Es idéntica a la anterior, salvo en el valor de la variable dependiente Z, que ahora la usamos sin negar.

Pero, las salidas son negadas en el dispositivo.... Así que el ensamblador tiene que complementar la expresión

$$Z = A + B + C$$

$$\neg Z = \text{???????} \text{ (aquí se aplica Morgan)}$$

PALASM: archivo .trf (simulación)

```
PALASM90 version 1.1 PLDSIM - MARKET RELEASE (8-27-90)
(C) - COPYRIGHT ADVANCED MICRO DEVICES INC., 1990

PALASM SIMULATION SELECTIVE TRACE LISTING

Title      : Example                      Author   : ECEn 220
Pattern    : Simple                      Company  : BYU
Revision   : A                          Date     : March 1, 1995

PAL20L8
Page : 1

      88888888
A  LLLLHHHH
B  LLHHLLHH
C  LHLHLHLH
/X HLLLLLLL
Z  LHHHHHHH
```

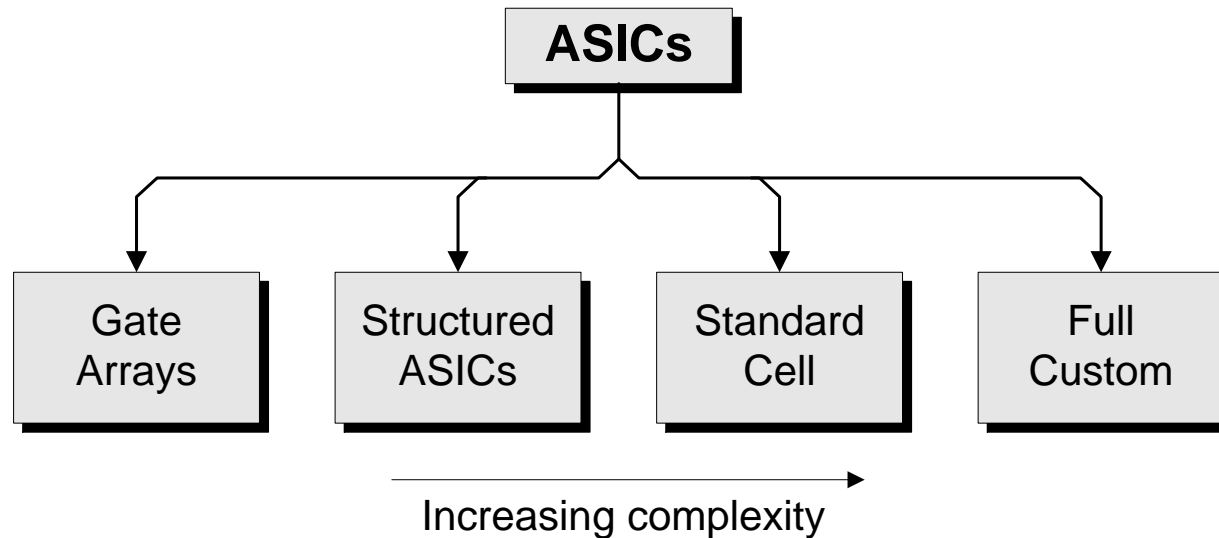
ASIC

- Application Specific IC
 - Diseñado para una función específica.
 - Contienen cientos de millones de puertas lógicas y pueden ser usados para crear funciones complejas.
 - El proceso de diseño y construcción de un ASIC es largo y caro, y finaliza en su realización en silicio.
 - No puede ser usado ni testeado antes de su fabricación.

Fabricación de un IC

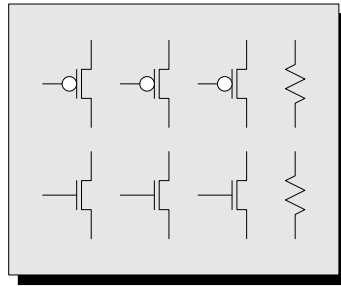
- Los transistores y sus conexiones se construyen mediante muchas capas (typical 10 to 15 in CMOS) puestas unas sobre las otra
- Cada capa tiene una forma especial definida por una máscara. Algunas de las capas o niveles forman transistores, otras los planos de conexión.
- Un aspecto importante de un IC es el tamaño del mas pequeño transistor que puede ser fabricado:
 - Este es medido en micrones (μm , 10^{-6} meter)
 - Por ejemplo, decimos que un IC está construido con un proceso de $0.50 \mu\text{m}$
 - Tal cual como profetizó Moore, el proceso continúa mejorando, o sea haciendose mas pequeño....
 - En este momento, el proceso de miniaturización es menor que $0.1 \mu\text{m}$ (deep sub-micron)

ASICs

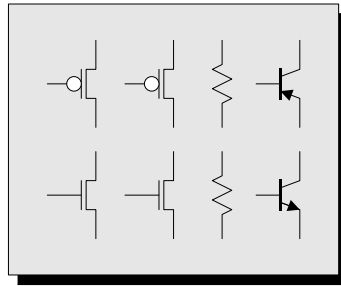


- ASIC: Es un chip (circuito integrado, IC) diseñado para una determinada aplicación y para una determinada compañía.
- Full custom: hecho enteramente por encargo (a medida): desde componentes pequeños, a microprocesadores diseñados y fabricados para una compañía específica.

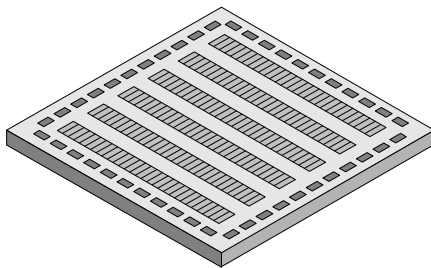
Gate Array



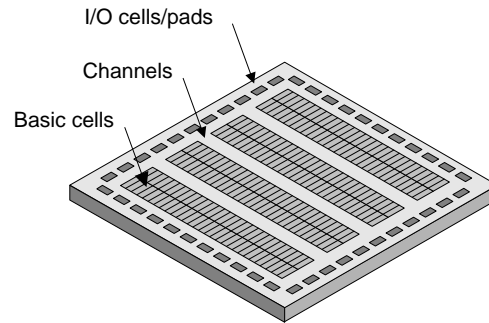
(a) Pure CMOS basic cell



(b) BiCMOS basic cell



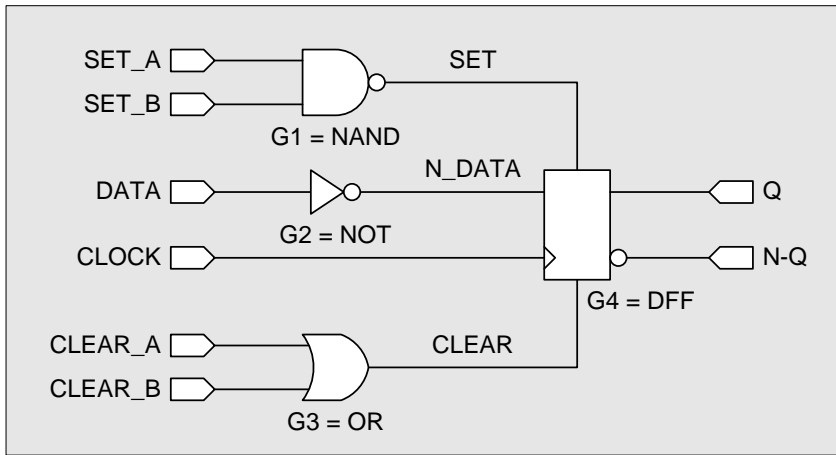
(a) Single-column arrays



(b) Dual-column arrays

- Gate Array: (1975) basado en la idea de celdas básicas formadas por transistores y resistencias sin conexión.
- Cada fabricante de ASIC determina que incluir en una celda básica, y construye chips presiliconados formados por arrays de celdas. (sea of cells).
- Los fabricantes definen una librería funciones lógicas (puertas primitivas, multiplexores, y registros) que son los que usan los ingenieros de la aplicación.
- Los ingenieros diseñan hasta llegar a nivel de netlist. Luego se hace el mapeo, ubicación y routing con las herramientas provistas por el fabricante.
- El resultado de este proceso son las máscaras con las cuales se crean los niveles de metalización que unirán las celdas básicas entre sí, así como los componentes dentro de las celdas básicas.

Schematic. Gate Level. Netlist.



```
BEGIN CIRCUIT=TEST
  INPUT  SET_A, SET_B, DATA, CLOCK, CLEAR_A, CLEAR_B;
  OUTPUT Q, N_Q;
  WIRE   SET, N_DATA, CLEAR;

  GATE G1=NAND (IN1=SET_A, IN2=SET_B, OUT1=SET);
  GATE G2=NOT  (IN1=DATA, OUT1=N_DATA);
  GATE G3=OR   (IN1=CLEAR_A, IN2=CLEAR_B, OUT1=CLEAR);
  GATE G4=DFF  (IN1=SET, IN2=N_DATA, IN3=CLOCK,
                IN4=CLEAR, OUT1=Q, OUT2=N_Q);

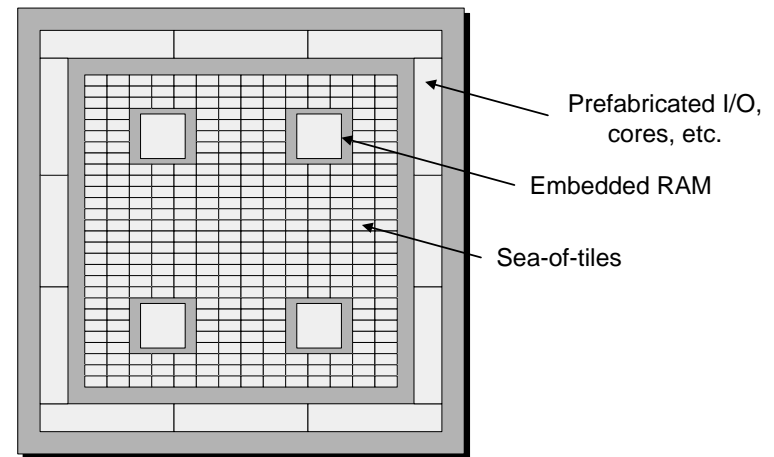
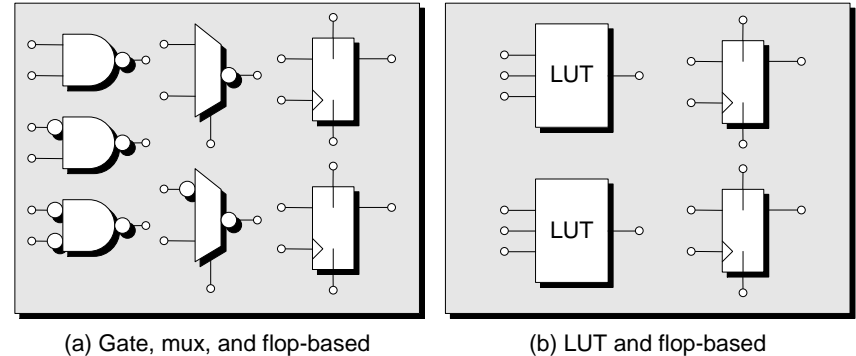
END CIRCUIT=TEST;
```

- Gate Level: el diseño se representa como una colección de puertas lógicas y funciones y conexiones entre ellas.
- Netlist: para poder usar el simulador, el diseño debe describirse de forma textual (archivos de texto)

Structured ASIC

- (2002) Cada vendedor tiene su arquitectura.
- Cada dispositivo está formado por elemento básico llamado módulo (tiles) que contiene una mezcla de lógica prefabricada (multiplexores, puertas, lookup table) junto con uno o mas registros y posiblemente algo de RAM
- Un array (sea) de estos elementos se prefabrica sobre la superficie del chip. Además, en los bordes de este mar de “tiles” (tejas, baldosas, ladrillo) hay bloques de RAM, generadores de reloj, etc.
- Cada dispositivo se “particulariza” mediante niveles de metalización, aunque muchos de estos niveles ya están también predefinidos. Solo 2 o 3 niveles se deben aplicar. Reducción de costos.
- Consumen mas que un standard cell. Tambien ocupan mas. (dos o tres veces mas).

Structured ASIC tiles



Generic structured ASIC

Standard Cell

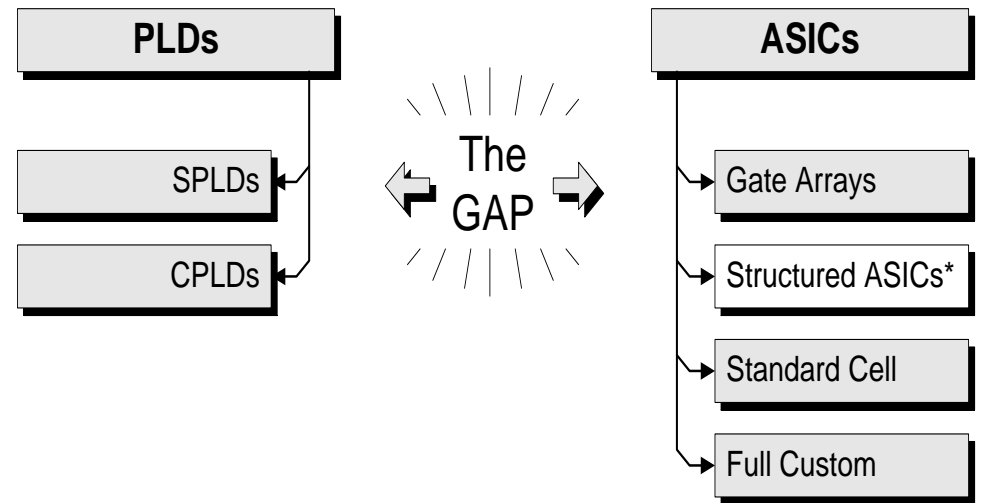
- Como en el caso de Gate Arrays, el fabricante define un conjunto de bloques básicos (multiplexores, registros, puertas, etc) que ofrece al ingeniero en forma de librerías.
- También ofrece librerías que pueden incluir microprocesadores, elementos de comunicación, funciones de ROM y RAM.
- Además hay IP que los ingenieros pueden reusar.
- IP: Intellectual Property: bloques funcionales creados por algun otro. Se compran.
- Los ingenieros, con todos esos elementos hacen el diseño hasta llegar a nivel de netlist, que describe las puertas lógicas que usarán y sus conexiones. Las herramientas de diseño son, incluso hoy, muy sofisticadas)
- La diferencia con las Gate Arrays es que no hay nada prefabricado. Cada función se crea con el mínimo número de transistores necesarios, sin componentes redundantes.
- Mas eficiente uso del silicio que Gate Arrays.

Circuitos Dedicados (full custom)

- Los ingenieros tienen el control completo sobre cada una de las máscaras usadas para fabricar el chip.
- El vendedor del ASIC no prefabrica ningún componente en el silicio y no provee ninguna librería ni puertas predefinidas.
- Por medio de las herramientas apropiadas, los ingenieros pueden modelar “a mano” las dimensiones de los transistores y pueden crear sus propias funciones basados en estos transistores. Incluso, las propias herramientas con las que ellos hacen estas cosas son diseñadas por ellos.
- El proceso es altamente complejo, y consume mucho tiempo, pero el chip resultante contiene la máxima cantidad de lógica con el mínimo desperdicio de silicio.

FPGA

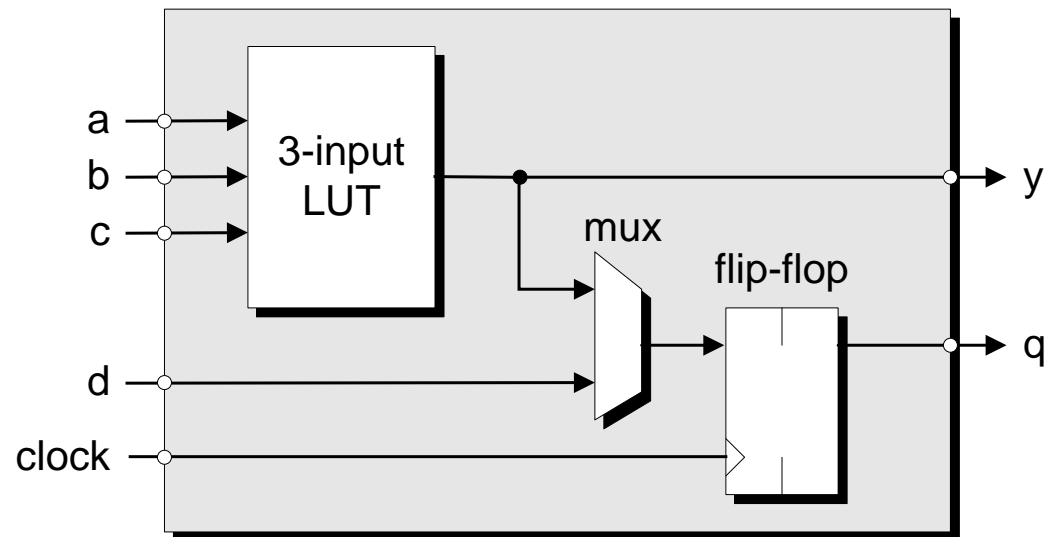
- Hacia 1980, es evidente que hay un GAP entre el mundo de los IC.
- Por un lado, los dispositivos programables, muy sencillos pero muy configurables. Por el otro, los ASIC's, soportando funciones complejas, pero muy caros, y muy costosos en tiempo de diseñar. Además, una vez el diseño estaba hecho, quedaba congelado en el silicio.
- Para salvar ese GAP, Xilinx lanza al mercado en 1984, una nueva clase de IC: FPGA.



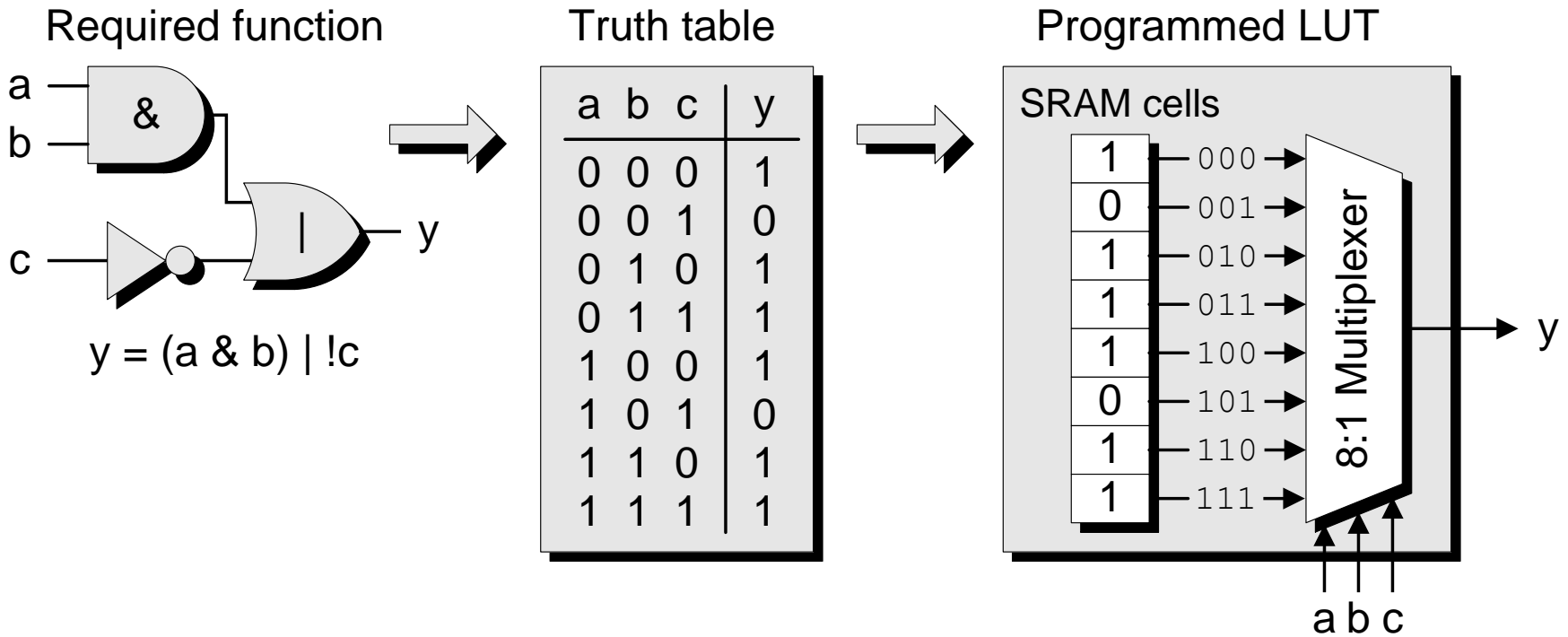
*Not available circa early 1980s

FPGA

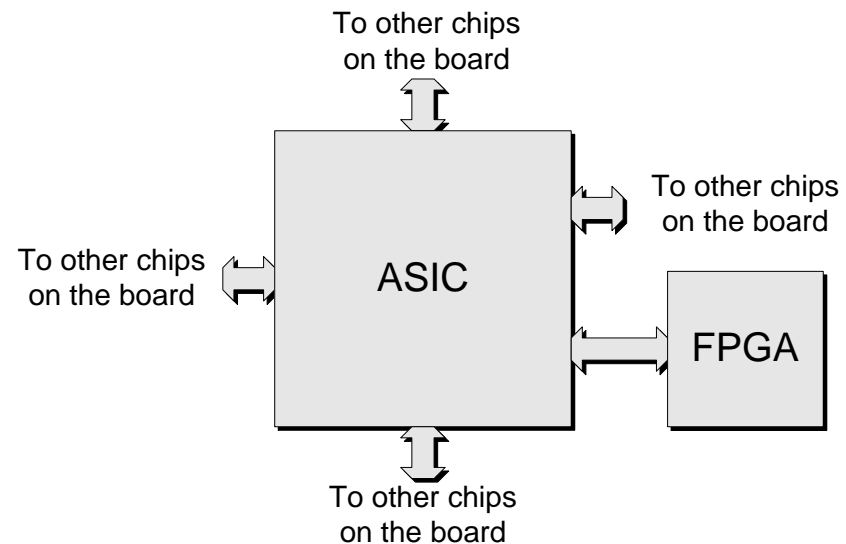
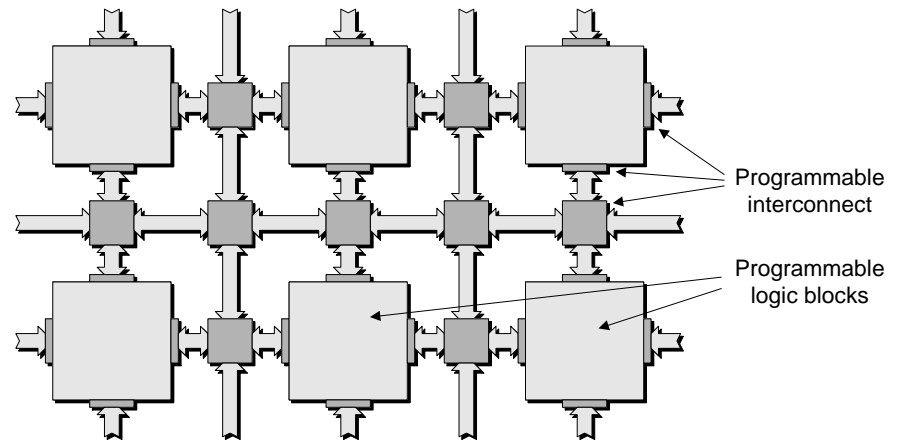
- Basados en el concepto de bloque lógico programable, que contiene una LUT, un registro y un multiplexor.
- Cada FPGA contendrá un número grande de estos bloques.
- Utilizando SRAM todos los bloques se pueden programar para que realicen una función diferente.



FPGA



- Un FPGA está compuesto por un gran número de bloques lógicos “islas” rodeados por un “mar” de interconexiones programables.
- Además, suele haber interconexiones globales de alta velocidad que atraviesan todo el chip.
- Y, entradas/salidas.
- Los FPGA son el puente entre los ASIC's y los PLD's porque son altamente configurables, y, pueden ser utilizados para implementar funciones complejas.
- Pueden ser utilizados en diseños híbridos, junto con ASIC's.
- Los FPGA son arquitecturas de grano medio porque utilizan bloques lógicos, a diferencia de los ASIC que son grano fino porque se programan a nivel de transistor



Anexos

- Rendimiento y Consumo en microprocesadores y ASICs
- Moore
- High End Reconfigurable Computing

Consumo y Rendimiento

Comparación entre procesadores y ASIC's

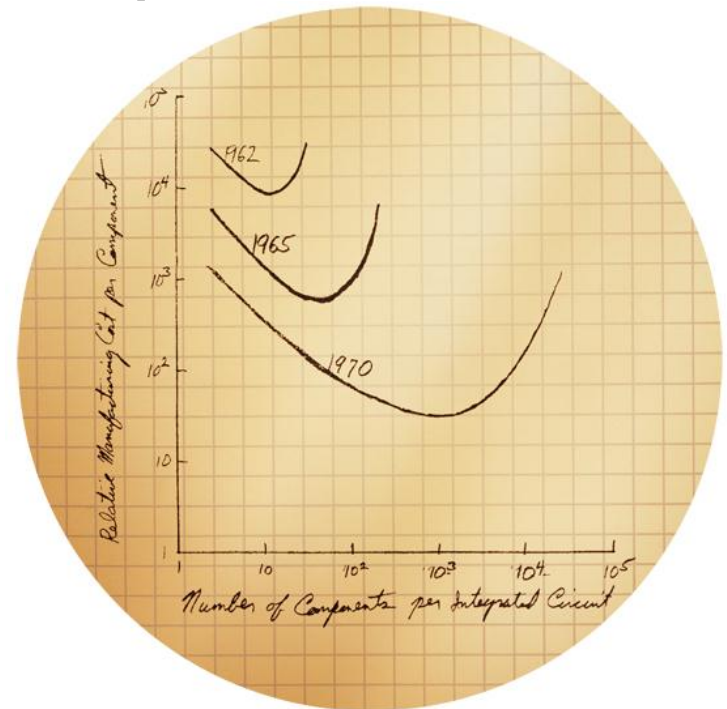
- “Efficient Embedded Computing” .Artículo de Stanford University, está en la página. Computer. Julio 2008
- Algunos números interesantes de rendimiento:
 - Dispositivos electrónicos (teléfonos celulares, cámaras de video y televisión digital) realizan más operaciones que la más rápida computadora, y con mucho menor consumo que una computadora personal.
 - Ejemplo: 3G mobile phone receiver requiere 35 a 40 GOPS (Giga(mil millones) Operaciones por segundo) de rendimiento para soportar un canal de 14.4 Mbps (para un canal de 100Mbps (OFDM) requerirían entre 210 y 290 GOPS.
 - En contraste, el rendimiento de pico de una computadora típica es de algunos GOPS, pero mantiene menos de un GOPs en la mayoría de las aplicaciones.

Consumo y rendimiento

- Consumo:
 - Eficiencia requerida para un embebido: 25pJ/op (para el receptor 3G)
 - Un ASIC , diseñado cuidadosamente, puede conseguir una eficiencia de 5pJ/op en la tecnología CMOS de 90-nm.
 - Un procesador embebido muy eficiente o un DSP requieren 250pJ/op
 - Un laptop requiere 20nJ/op (4000 veces la de un ASIC)
 - Conclusión: las eficiencias de los procesadores programables son inadecuadas para las aplicaciones embebidas, obligando a los ingenieros a usar hardware. (caro, muy caro!!! En dinero y en tiempo de desarrollo)

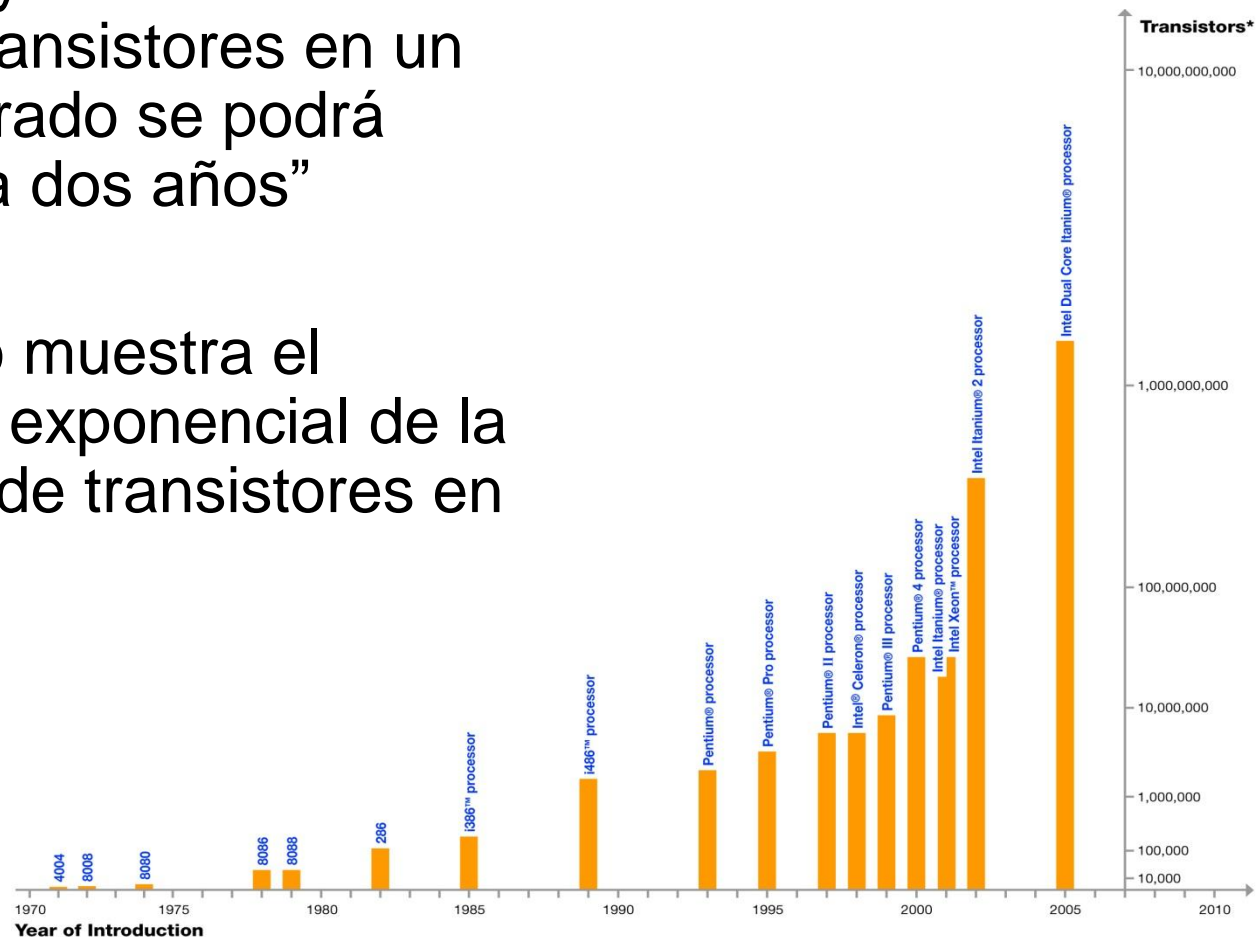
Ley de Moore

- Moore dió un discurso en el año 1965, donde predijo que en el año 1970 (cinco años mas adelante) el costo por componente sería la décima parte del actual costo.
- Y además predijo que esa tendencia se mantendría por 10 años.



40 años de Ley de Moore

- Una de las maneras en que se enunció la ley de Moore fué : “ El número de transistores en un circuito integrado se podrá duplicar cada dos años”
- Este cuadro muestra el crecimiento exponencial de la integración de transistores en un chip.



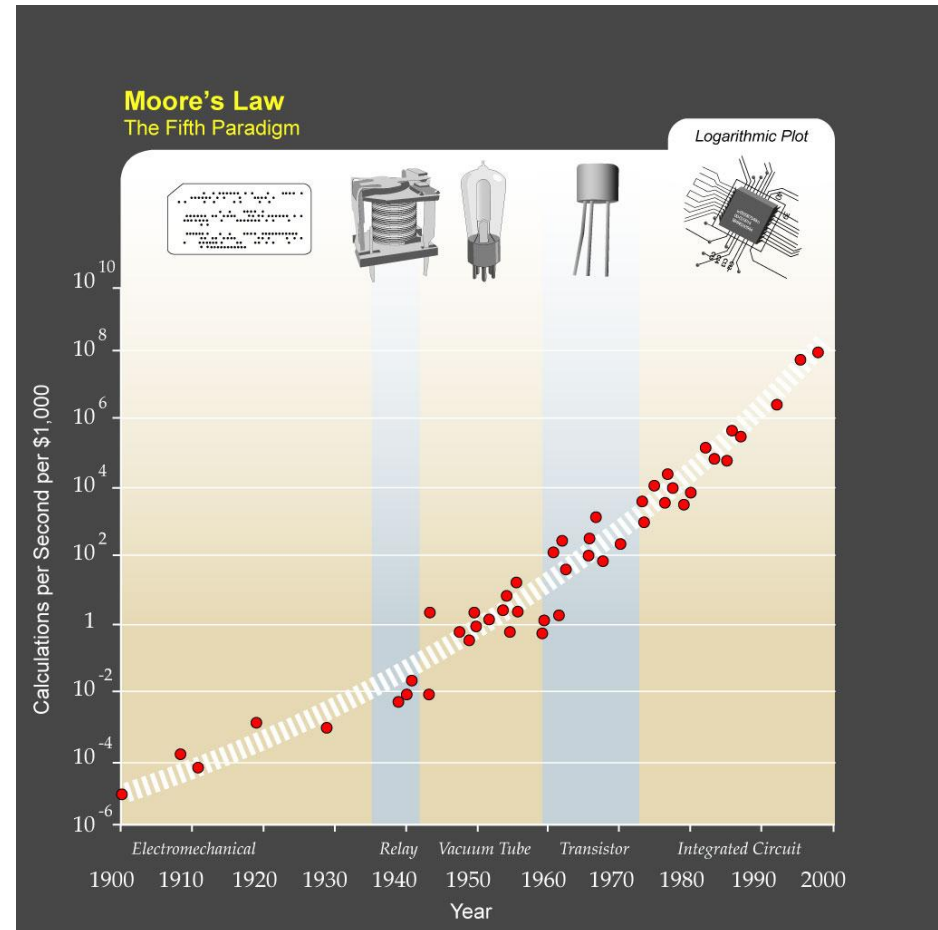
*Note: Vertical scale of chart not proportional to actual Transistor count.

Es interesante también extrapolar hacia atrás...

...podemos ver como la tecnología hace que el poder de cómputo crezca exponencialmente con el tiempo.

En rigor, el autor de este cuadro (“The Law of Accelerating Returns “ by [Ray Kurzweil](#)) dice que lo que no nos damos cuenta es que hay un doble crecimiento exponencial, de lo contrario no estaríamos obteniendo una curva sino una línea recta en el cuadro de la derecha).

Están representadas 49 máquinas. Desde las primeras calculadoras electromecánicas que calcularon el censo de EEUU entre 1890 y 1899, la máquina de Robinson que craqueó el código secreto de los nazis en 1940, la máquina CBS de tubos de vacío que predijo la elección de Eisenhower, las computadoras que se usaron en los primeros lanzamientos al espacio.... Y.... Aquellas en las que escribimos y mostramos esta transparencia....



Sin embargo ...

- Miremos este cuadro. Muestra que si tomamos como medida la “densidad computacional” que relaciona el número de operaciones por segundo, la frecuencia de operación y el área ocupada, vemos que ésta ha ido descendiendo con los años...
- Esto muestra que los microprocesadores no están siendo capaces de traducir en un aumento equivalente en el rendimiento (MOPS) al aumentar la frecuencia y el área ocupada.
- Sin embargo, en el mismo período de tiempo, los FPGA, han sobrepasado la Ley de Moore.

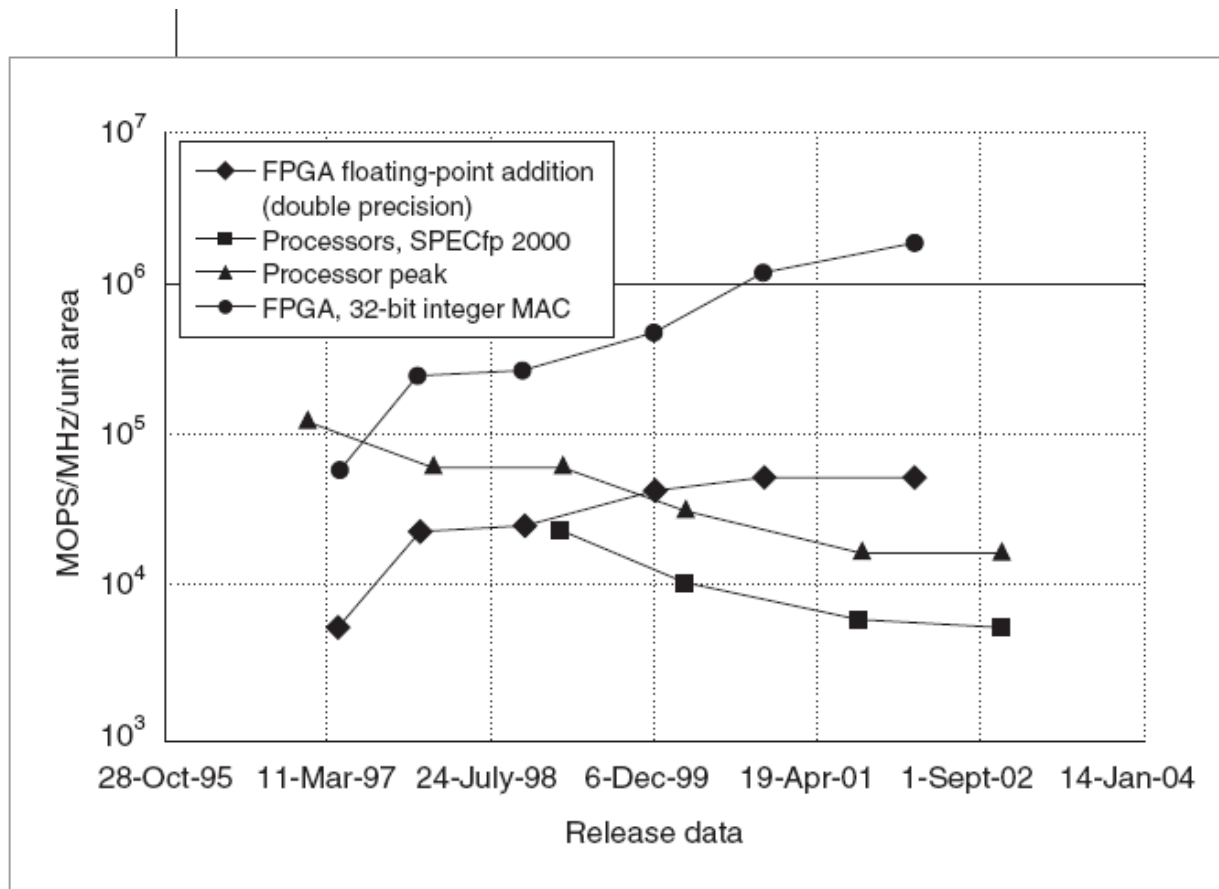


Figure 1. Computational density of FPGAs and Intel processors.

Referencia:

BEE2: A High-End Reconfigurable Computing System. Chen Chang, John Wawrzynek, and Robert W. Brodersen. University of California, Berkeley. IEEE Design & Test of Computers, March–April 2005.

...el rendimiento de un solo procesador no aumenta acorde a la tecnología

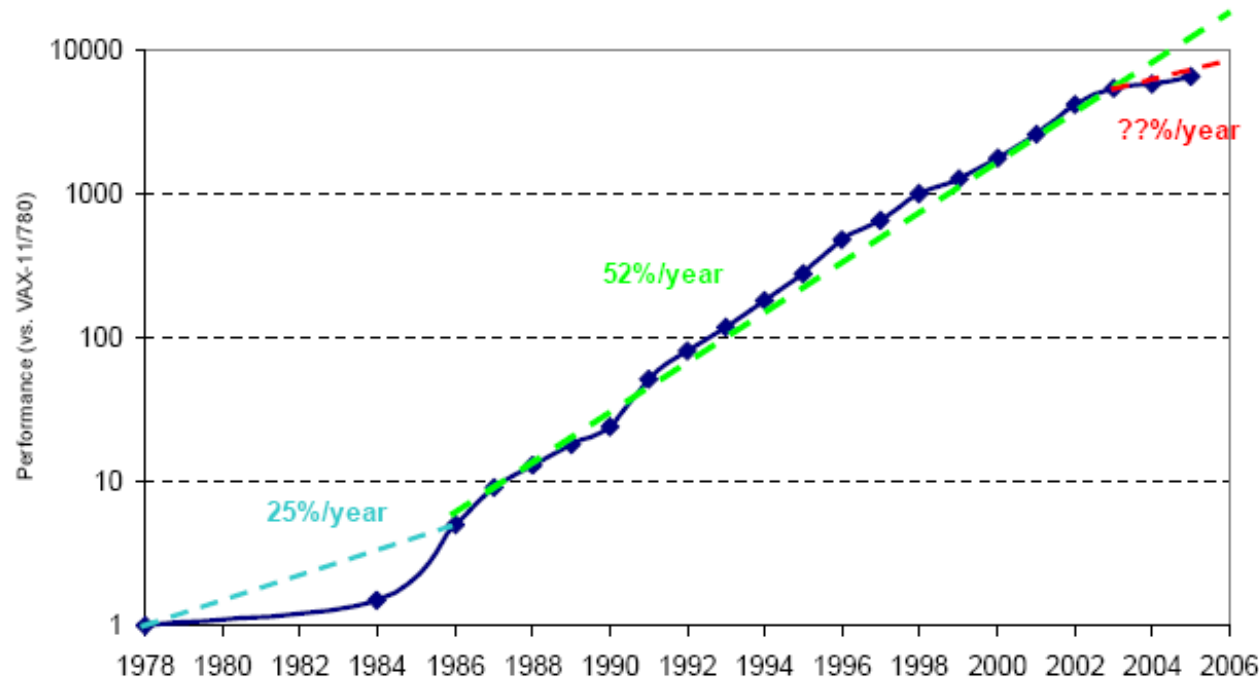


Figure 2. Processor performance improvement between 1978 and 2006 using integer SPEC [SPEC 2006] programs. RISCs helped inspire performance to improve by 52% per year between 1986 and 2002, which was much faster than the VAX minicomputer improved between 1978 and 1986. Since 2002, performance has improved less than 20% per year. By 2006, processors will be a factor of three slower than if progress had continued at 52% per year. This figure is Figure 1.1 in [Hennessy and Patterson 2007].

¿Como aumentar el rendimiento? (¿Como hacer que la Ley de Moore continúe?)

- Si un solo procesador ya no puede ser capaz de obtener un aumento exponencial del rendimiento conforme la tecnología avanza, entonces el aumento del rendimiento debe provenir de multiplicar los núcleos de procesador (arquitecturas multicore) dentro del chip. Es el camino que siguen en este momento IBM (PowerPc), Sun e Intel.
- ¡Esto quiere decir que toda la computación se vuelve paralela!

High Performance Computing

- En el mundo de las Supercomputadoras, rendimientos entre los 100Gflops y 10Teraflops se obtienen en base a sistemas contruidos como una colección de procesadores del mercado (componentes “off-the-shelf”).
- Como se producen tan pocos sistemas de este tipo, no se justifica utilizar procesadores dedicados al cómputo de alto rendimiento (Todas las veces que se intentó, la empresa quebró.)

High End Reconfigurable Computing

- Son sistemas de super cómputo basados en FPGA's.



- RAMP Blue: contiene
(<http://ramp.eecs.berkeley.edu/index.php?index>)
 - 21 sistemas llamados BEE2.
 - Cada uno de los sistemas BEE2 contiene 48 cores de Microblaze (core de procesador implementado en una FPGA)
 - O sea, en total el sistema tiene 1008 procesadores

Referencias

- BEE2: A High-End Reconfigurable Computing System
Chen Chang, John Wawrzynek, and Robert W. Brodersen. University of California, Berkeley
IEEE Design & Test of Computers, 0740-7475/05/\$20.00
© 2005 IEEE
- *“The Law of Accelerating Returns “ by Ray Kurzweil*
- The Landscape of Parallel Computing Research: A View from Berkeley. David Paterson et al.
- <http://www.embedded.com/1999/9905/9905turley.htm>. Embedded processors by the numbers. Jim Turley