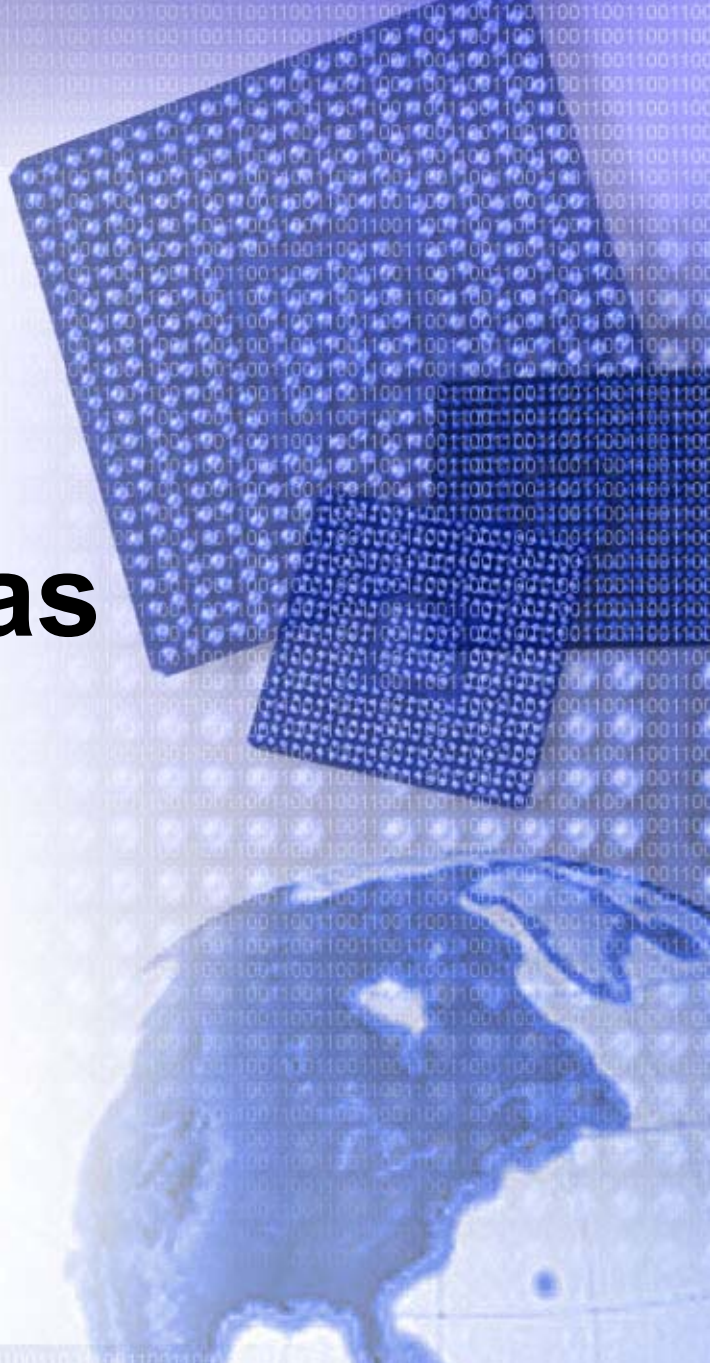
XILINX®

Una inspección a las tecnologías de implementación



Tecnologías de Implementación

- Small scale and medium scale integration.
  - Up to about 200 gates per device
  - Most common is 74xx type devices
    - Gates, flip flops, latches.
    - Decoders, registers, counters, and other functional building blocks.

Tecnologías de Implementación

- Large scale integration.
  - Ranging from 200 to 200,000 gates per device.
  - Small memories, programmable logic devices, custom designs.
- Very large scale integration.
  - Above 200,000 gates per device.
  - Often “gate count” is replaced by transistor count because these large designs have integrated memories, etc.

Tecnologías de Implementación

- Survey of small and medium scale components by browsing data books.
  - Different functional classes.
  - Generally used as “glue” logic now, to help interface larger scale components.
  - Back in the day, large designs were done using this technology.

Tecnologías de Implementación

- Advantages of small and medium scale, particularly with regard to 74xx stuff.
  - Easy to understand functions.
  - Exceptional signal visibility.
- Disadvantages.
  - Low logic density means big boards or small designs only.
  - Higher power consumption.
  - Cost per function, failure concerns.

Tecnologías de Implementación

- Survey of large scale components, for logic design, particularly programmable logic devices in this density.
  - Many different flavors of devices; most draw on basic device types.
    - ROM, PLA, PAL = PLDs.
    - CPLDs
  - Can be used as glue logic but have enough available logic to implement significant designs in larger parts.

Tecnologías de Implementación

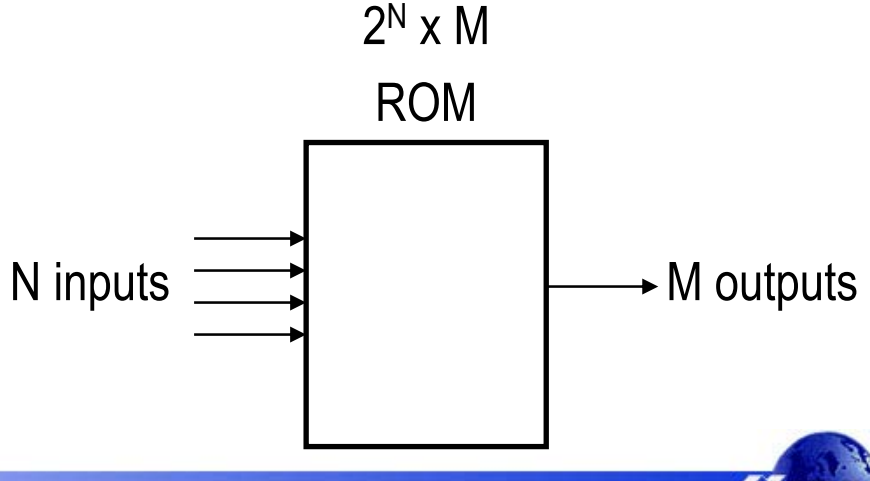
- Advantages of large scale integration.
  - Higher logic density means smaller boards or larger designs.
  - Many devices can be programmed and reprogrammed, saving expense when changes are made.
- Disadvantages.
  - Need to learn how to use and program.
  - Signal visibility is reduced.

Tecnologías de Implementación

- What is a ROM? How can I use it?
- What is a PLA? How can I use it?
- What is a PAL? How can I use it?
- How are all these things related?
- What, then, is a CPLD?

Tecnologías de Implementación

- A ROM is a SOP logic device with a fixed AND array and a programmable OR array.
- You can implement M functions of N inputs in this ROM.

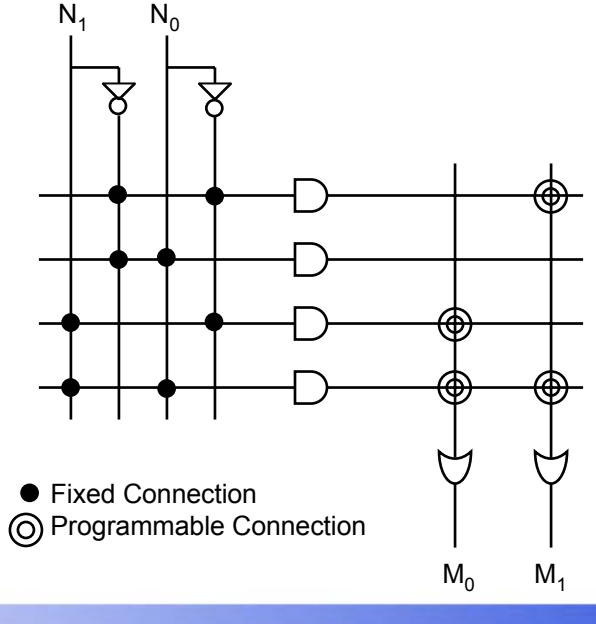


Tecnologías de Implementación

- You basically specify a truth table of the functions when you program the ROM.
- There is no advantage to simplifying the function when you are using a ROM since you need to specify the entire list of minterms anyway…

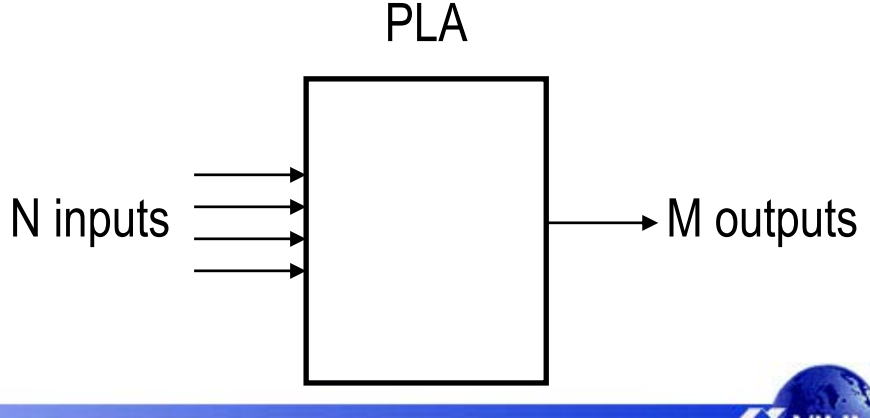
Tecnologías de Implementación

- ROM of 2^N by M; N = 2, M=2
- M0 = N1•N0 + N1•N0’
- M1 = N1•N0 + N1’ •N0’



Tecnologías de Implementación

- A PLA is a SOP logic device with a programmable AND array (fewer pt's than a ROM) and a programmable OR array.
- You can implement functions using the available minterms, which may be shared between functions.

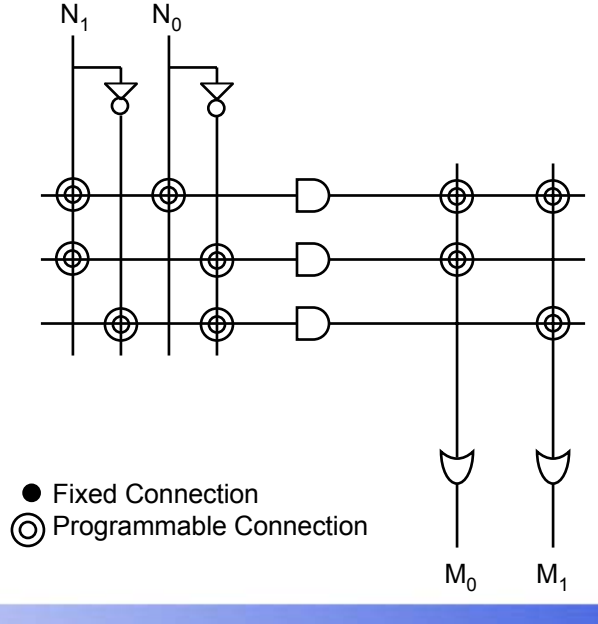


Tecnologías de Implementación

- You have to reduce your design to a sum of products which will hopefully be realizable with the available minterms.
- Computer aided design tools are available to do optimization for product term sharing.

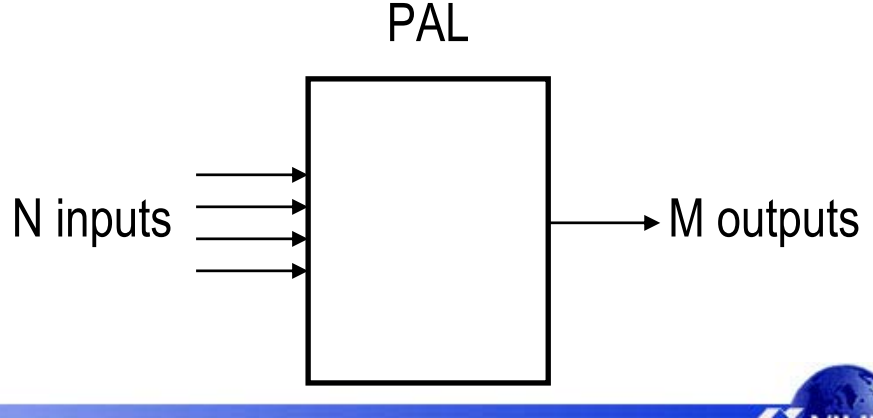
Tecnologías de Implementación

- PLA of N inputs and M out; N = 2, M=2
- M0 = N1•N0 + N1•N0’
- M1 = N1•N0 + N1’ •N0’



Tecnologías de Implementación

- A PAL is a SOP logic device with a programmable AND array and a fixed OR array.
- You can implement functions using the available minterms for each output function (no pt sharing).

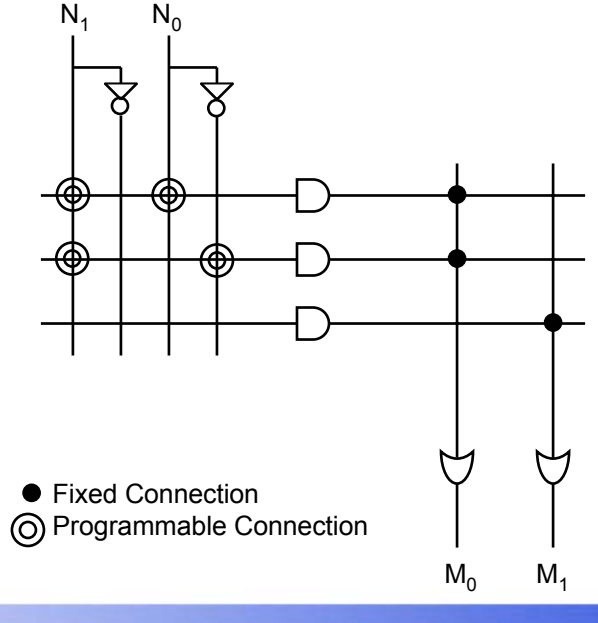


Tecnologías de Implementación

- Again, the design has to be reduced if possible.
- No product term sharing, and note that in real devices, each output function may have access to a different number of product terms.

Tecnologías de Implementación

- PAL of N inputs and M out; N = 2, M=2
- M0 = N1•N0 + N1•N0’
- M1 = N1•N0 + N1’ •N0’ insufficient minterms



Tecnologías de Implementación

- A CPLD is a complex programmable logic device that essentially consists of a number of programmable logic blocks (such as a PLA, PAL, and less commonly, ROM) connected by a programmable interconnect array.
- Why has CPLD density stagnated?

Tecnologías de Implementación

- Full Custom Logic.
- Standard Cell Design.
- Gate Array Design.
- Field Programmable Logic.

Tecnologías de Implementación

- Full Custom Logic.
  - Each primitive logic function or transistor is manually designed and optimized.
  - Most compact chip design, highest possible speed, lowest power consumption.
  - Non recurring engineering cost (NRE) is the highest for obvious reasons.
  - Rarely used today due to high engineering cost and low productivity.

Tecnologías de Implementación

- Standard Cell Design.
  - Predefined logic blocks (a la 74xx style) are made available to the designer in a cell library; the design is built with these.
  - Done with schematic capture or HDL.
  - Automated tools place and route the cells.
  - Cells are often standard dimensions to facilitate automated place and route.
  - Substantially shorter design time than custom.

Tecnologías de Implementación

- Gate Array Design.
  - Full custom and standard cell require custom masks to produce wafers (read as “expensive”).
  - Instead, create base wafers using common masks; base wafers have an “array” of gates which are not committed and not wired.
  - Designers specify connectivity and the top metal masks are created to connect the gates on the base wafers.
  - Low wafer cost, fast turnaround, area penalty.

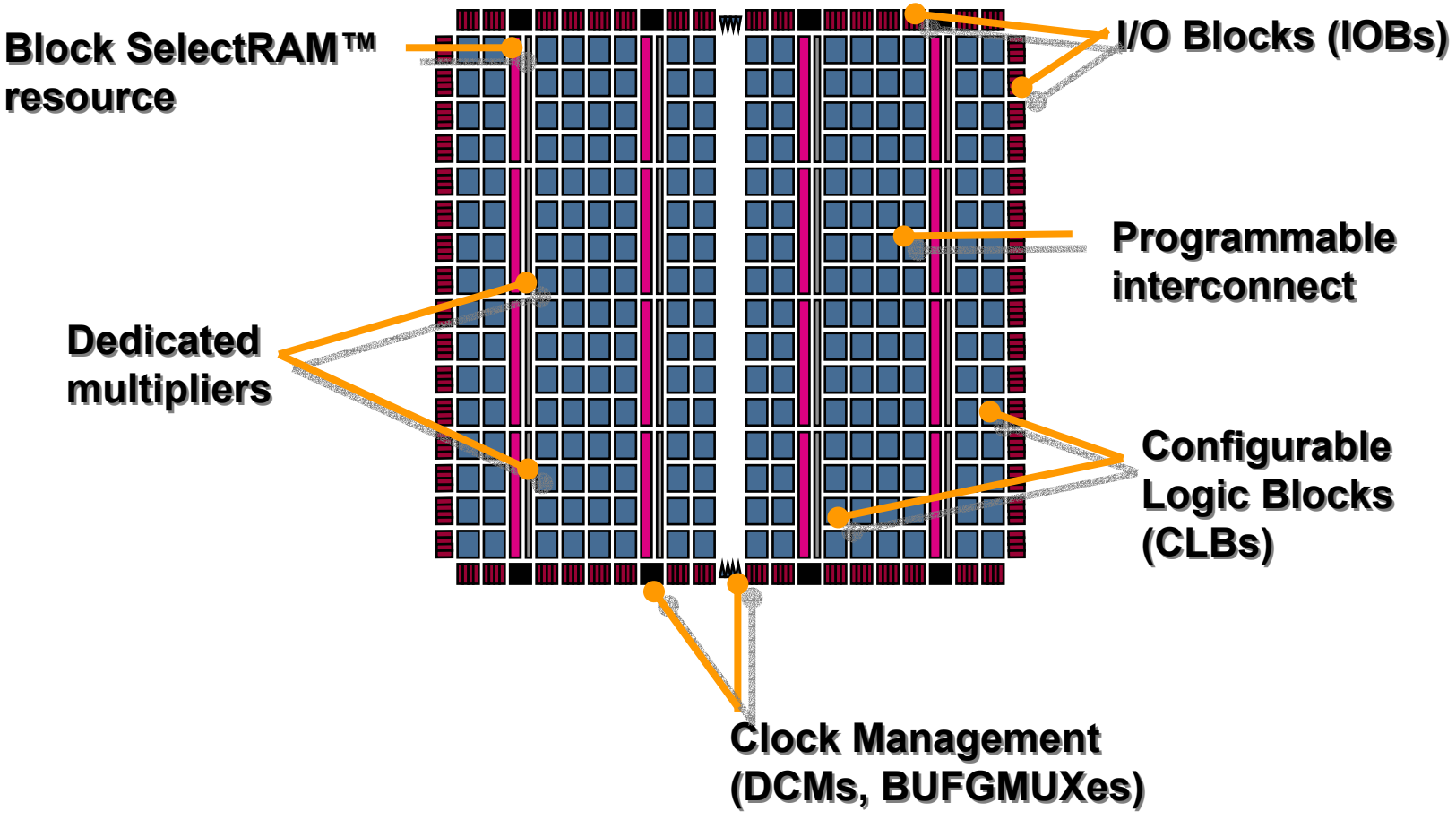
Tecnologías de Implementación

- Field Programmable Logic.
  - We have already discussed several types of programmable logic.
    - ROM, PLA, PAL.
    - CPLD.
  - The other main type of programmable logic is the field programmable gate array, or “FPGA”.

Field Programmable Gate Arrays

- FPGA devices are an improvement in gate array technology which offer improved time to market and reduced prototyping cost.
- Types of FPGA devices:
  - Non volatile, one time programmable (anti-fuse).
  - Non volatile, re-programmable (flash).
  - Volatile (sram).
- An FPGA is much more than an array of gates...

Arquitectura de la Spartan 3



Field Programmable Gate Arrays

- The programmable routing is of particular significance because this is the main improvement over a standard gate array.
- An FPGA is really some programmable logic with a whole bunch of programmable wires!!!
- Various array sizes are available from the vendor.

Field Programmable Gate Arrays

- We will discuss the architectural details of the Xilinx Spartan-3 family of FPGAs in this class.
- You should be aware that Xilinx has other architectures, and that other companies have competing architectures.

Xilinx Spartan-3 Family

- The Spartan product is a cost reduced, high volume FPGA. Most Spartan devices are a close relative to another Xilinx product.
- There are several Spartan FPGA families:
  - Spartan-II, Spartan-IIE (similar to Virtex).
  - Spartan-3, Spartan-3E (similar to Virtex-4).

Xilinx Spartan-3 Family

- The course currently uses the Spartan-3 FPGA family on a prototyping platform from Xilinx / Digilent.
  - High volume, 1.2 volt FPGA devices.
  - Pinout compatibility between devices.
  - On-chip memories and clock management.
  - Up to 5,000,000 system gates.

Spartan-3 Product Matrix

100X Density Range

Device	XC3S50	XC3S200	XC3S400	XC3S1000	XC3S1500	XC3S2000	XC3S4000	XC3S5000
System Gates	50K	200K	400K	1000K	1500K	2000K	4000K	5000K
Logic Cells	1,728	4,320	8,064	17,280	29,952	46,080	62,208	74,880
Dedicated Multipliers	4	12	16	24	32	40	96	104
Block RAM Blocks	4	12	16	24	32	40	96	104
Block RAM Bits	72K	216K	288K	432K	576K	720K	1,728K	1,872K
Distributed RAM Bits	12K	30K	56K	120K	208K	320K	432K	520K
DCMs	2	4	4	4	4	4	4	4
I/O Standards	24	24	24	24	24	24	24	24
Max Single Ended I/O	124	173	264	391	487	565	712	784

50,000 to 5,000,000 System Gates

Choice of Packages