FPGA based System Design

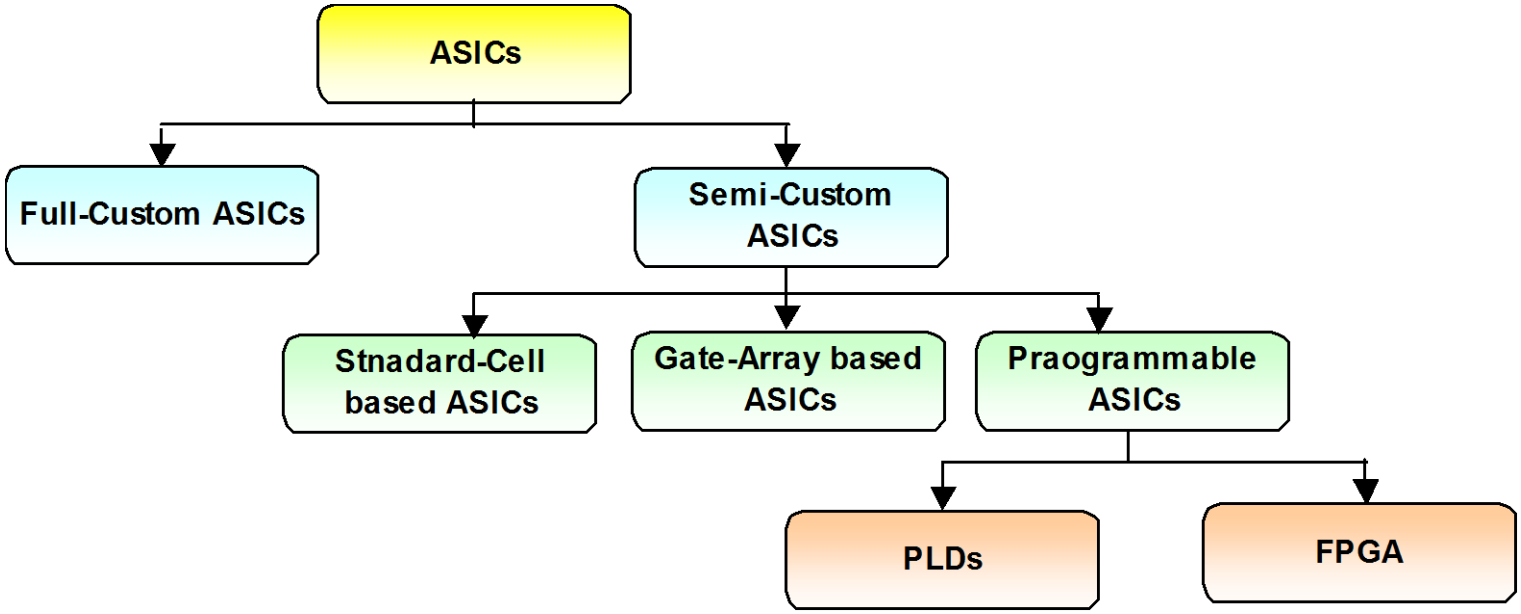
ASIC vs Standard IC

- **Standard ICs –** ICs sold as Standard Parts
  - SSI/LSI/ MSI IC such as MUX, Encoder, Memory Chips, or Microprocessor IC
- **Application Specific Integrated Circuits (ASIC) –** A Chip for Toy Bear, Auto-Mobile Control Chip, Different Communication Chips [GRoT: ICs not Found in Data Book]
  - Concept Started in 1980s
  - An IC Customized to a Particular System or Application – Custom ICs
  - Digital Designs Became a Matter of Placing of Fewer CICs or ASICs plus Some Glue Logic
  - Reduced Cost and Improved Reliability
- **Application Specific Standard Parts (ASSP) –** Controller Chip for PC or a Modem

Types of ASICs

- **Full-Custom ICs/Fixed ASICs and Programmable ASICs**
  - **Wafer :** A circular piece of pure silicon (10-15 cm in dia, but wafers of 30 cm dia are expected soon-IEEE micro-Sep/Oct. 1999, pp 34-43)
  - **Wafer Lot:** 5 ~ 30 wafers, each containing hundreds of chips(dies) depending upon size of the die
  - **Die:** A rectangular piece of silicon that contains one IC design
  - **Mask Layers:** Each IC is manufactured with successive mask layers(10 – 15 layers)
    - First half-dozen or so layers define transistors
    - Other half-dozen or so define Interconnect

Types of ASICs

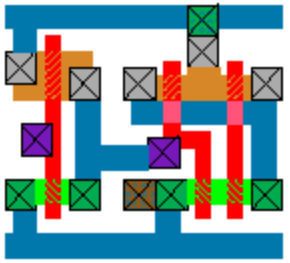


PLD: Programmable Logic Device

- Full-Custom ASICs: Possibly all logic cells and all mask layers customized
- Semi-Custom ASICs: all logic cells are pre-designed and some (possibly all) mask layers customized

Types of ASICs

❑Full-Custom ASICs



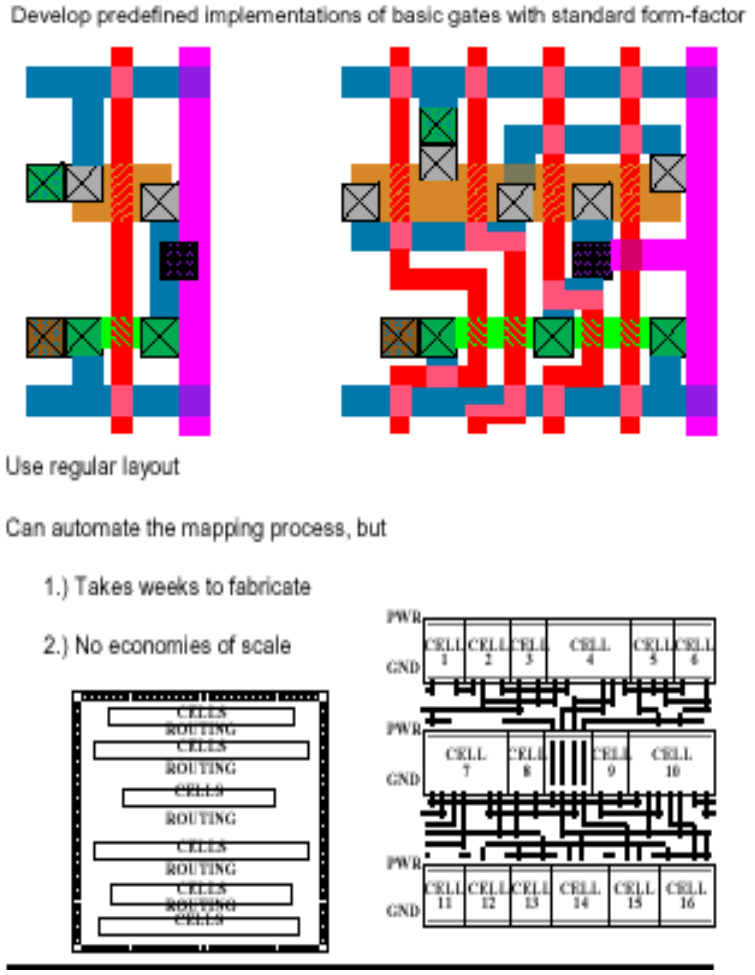
- ❖ Include some (possibly all) customized logic cells
- ❖ Have all their mask layers customized
- ❖ Full-custom ASIC design makes sense only
  - ✓ *When no suitable existing libraries exist or*
  - ✓ *Existing library cells are not fast enough or*
  - ✓ *The available pre-designed/pre-tested cells consume too much power that a design can allow or*
  - ✓ *The available logic cells are not compact enough to fit or*
  - ✓ *ASIC technology is new or/and so special that no cell library exits.*
- ❖ Offer highest performance and lowest cost (smallest die size) but at the expense of increased design time, complexity, higher design cost and higher risk.
- ❖ **Some Examples:** High-Voltage Automobile Control Chips, Ana-Digi Communication Chips, Sensors and Actuators

Types of ASICs

❑ Semi-Custom ASICs

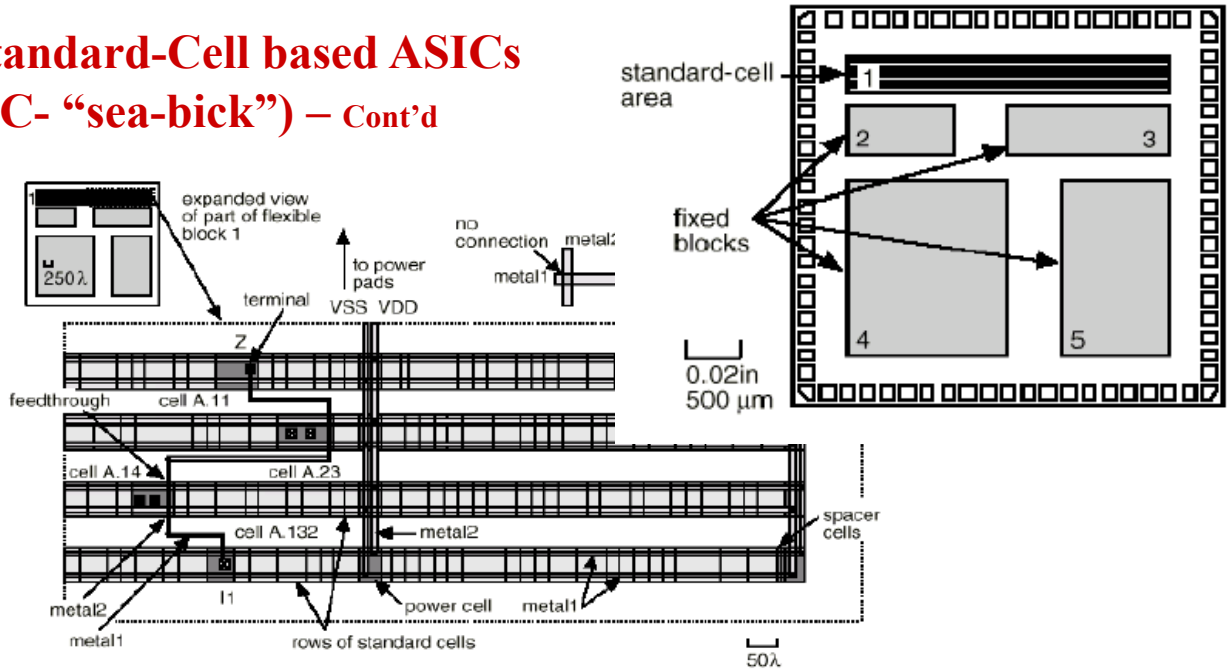
❖ Standard-Cell based ASICs (CBIC- "sea-bick")

- ✓ Use logic blocks from standard cell libraries, other mega-cells, full-custom blocks, system-level macros(SLMs), functional standard blocks (FSBs), cores etc.
- ✓ Get all mask layers customized- transistors and interconnect
- ✓ Manufacturing lead time is around 8 weeks
- ✓ Less efficient in size and performance but lower in design cost



Types of ASICs

- **Semi-Custom ASICs** – Cont'd
  - **Standard-Cell based ASICs (CBIC- "sea-bick") – Cont'd**



- Routing a CBIC (cell-based IC)
  - A "wall" of standard cells forms a **flexible block**
  - **metal2** may be used in a **feedthrough cell** to cross over cell rows that use **metal1** for wiring
  - Other wiring cells: **spacer cells**, **row-end cells**, and **power cells**

Types of ASICs

❑ Semi-Custom ASICs – Cont'd

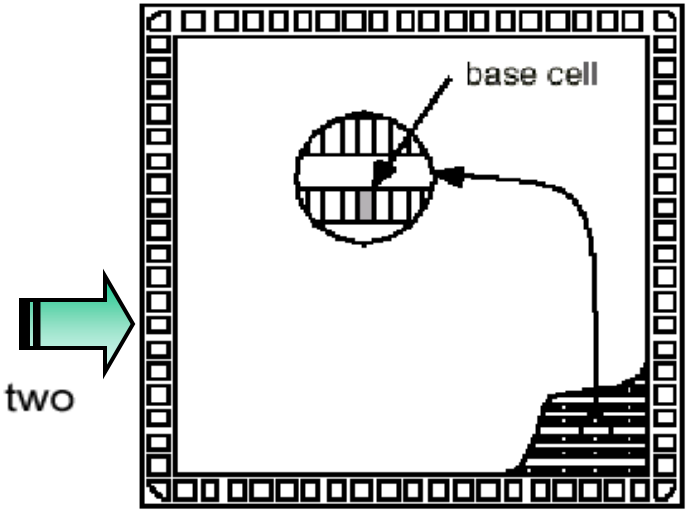
❖ Gate Array based ASICs

A **gate array**, **masked gate array**, **MGA**, or **prediffused array** uses **macros** (**books**) to reduce **turnaround time** and comprises a **base array** made from a **base cell** or **primitive cell**. There are three types:

- Channeled gate arrays
- Channelless gate arrays
- Structured gate arrays

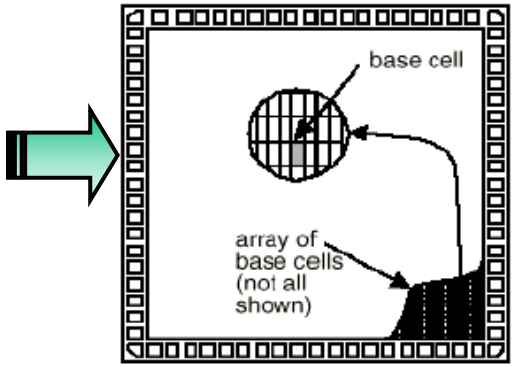
A **channeled gate array**

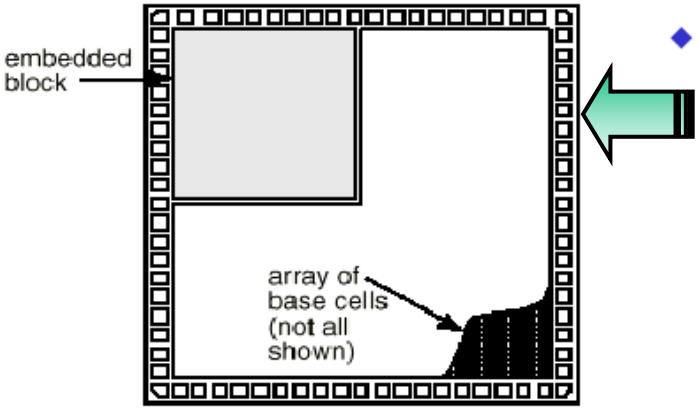
- Only the interconnect is customized
- The interconnect uses predefined spaces between rows of base cells
- Manufacturing lead time is between two days and two weeks



Types of ASICs

- **Semi-Custom ASICs –** Cont'd
  - **Gate Array based ASICs** - Cont'd
    - A **channelless gate array** (**channel-free gate array**, **sea-of-gates array**, or **SOG** array)
      - Only some (the top few) mask layers are customized — the interconnect
      - Manufacturing lead time is between two days and two weeks.





- An **embedded gate array** or **structured gate array** (**masterslice** or **masterimage**)
  - Only the interconnect is customized
  - Custom blocks (the same for each design) can be embedded
  - Manufacturing lead time is between two days and two weeks.

Types of ASICs

❑ Semi-Custom ASICs – Cont'd

❖ Programmable ASICs

✓ **PLDs -** PLDs are low-density devices which contain 1k – 10 k gates and are available both in bipolar and CMOS technologies **[PLA, PAL or GAL]**

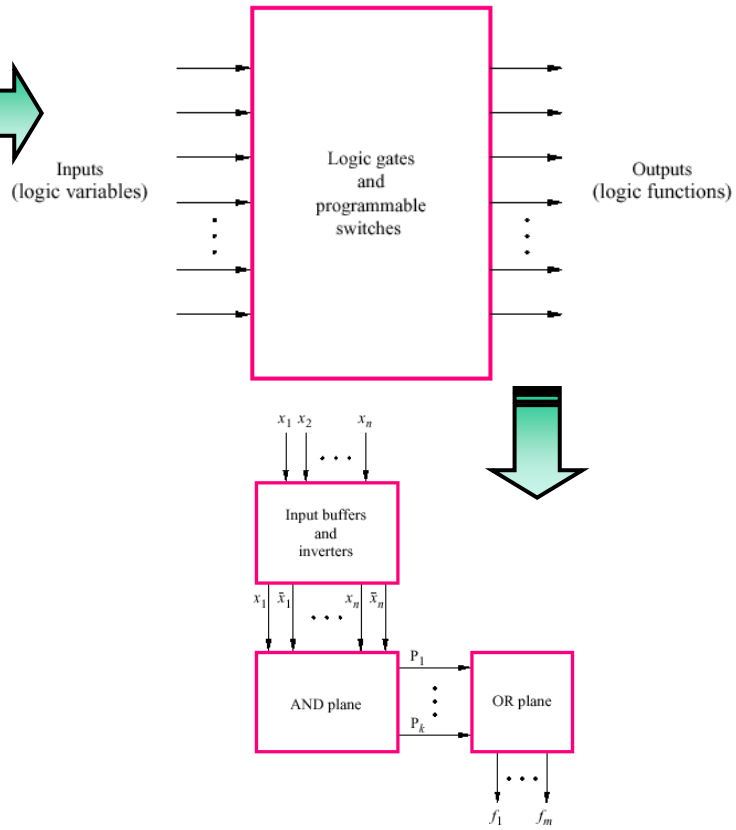
✓ **CPLDs or FPLDs or FPGAs -** FPGAs combine architecture of gate arrays with programmability of PLDs.

✓**User Configurable**

✓ **Contain Regular Structures -** circuit elements such as AND, OR, NAND/NOR gates, FFs, Mux, RAMs,

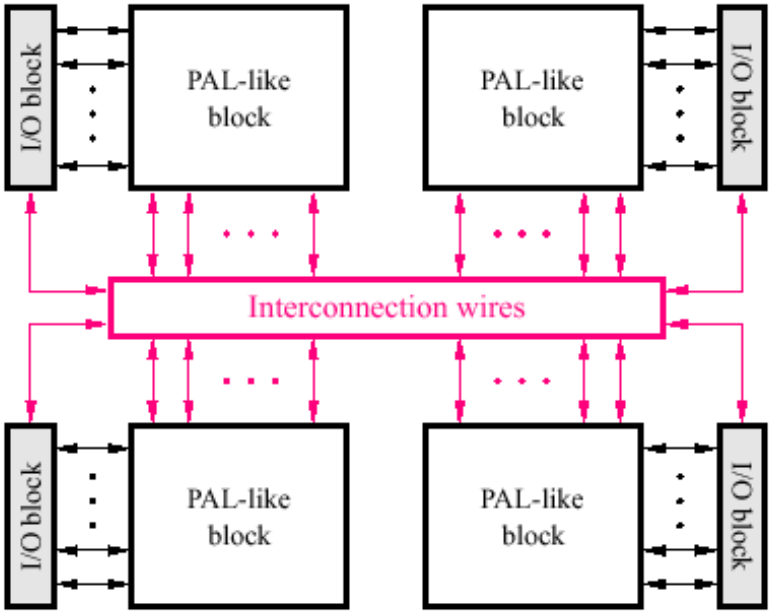
✓**Allow Different Programming Technologies**

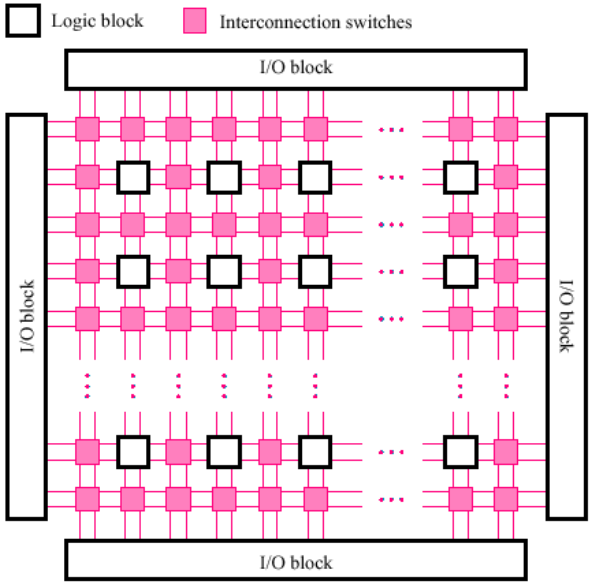
✓ **Allow both Matrix and Row-based Architectures**



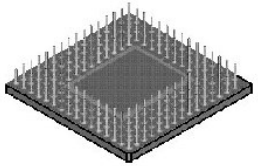
Types of ASICs

- **Semi-Custom ASICs** – Cont'd
  - **Programmable ASICs** - Cont'd
    - **Structure of a CPLD / FPGA**





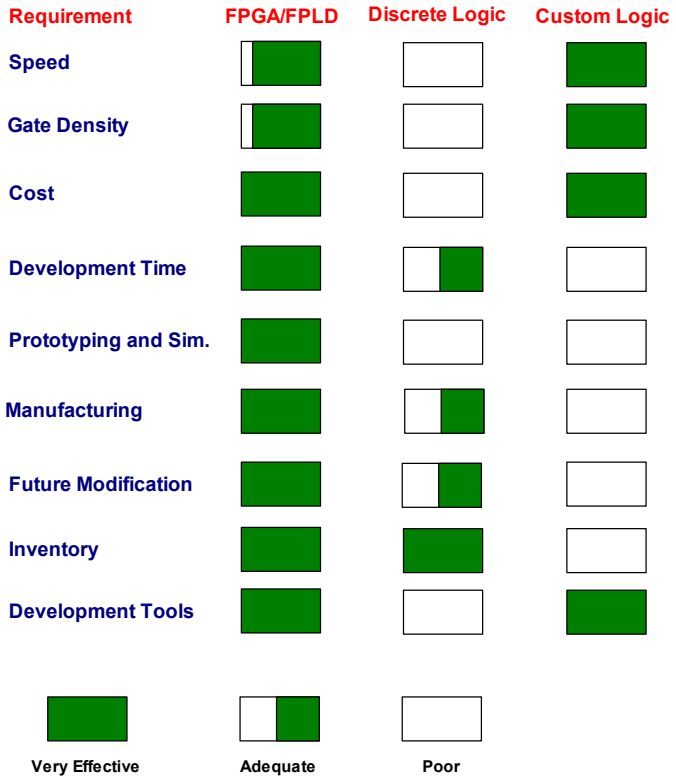
(a) General structure of an FPGA



(b) Pin grid array (PGA) package (bottom view)

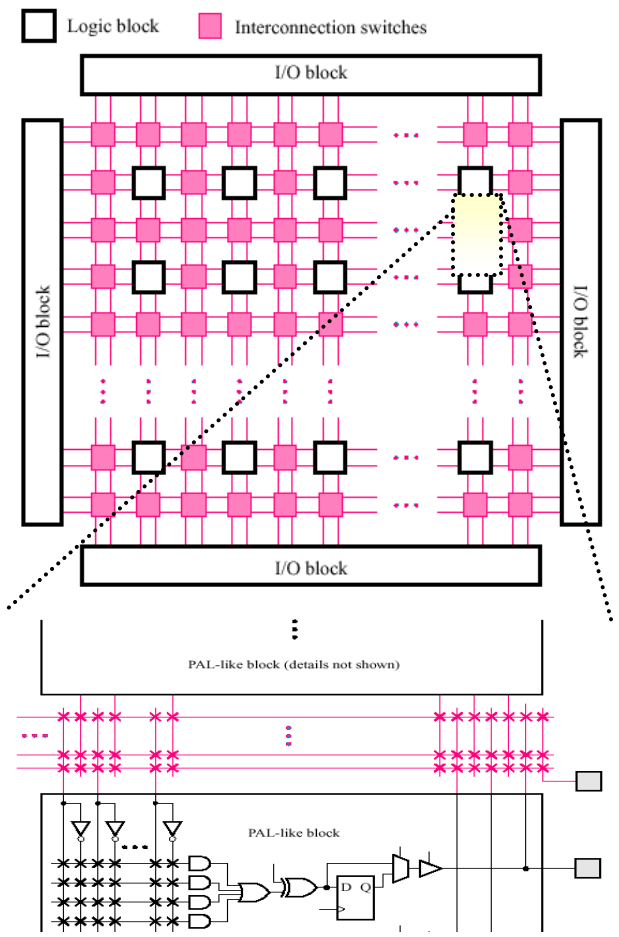
Why FPGA-based ASIC Design?

- **Choice is based on Many Factors ;**
  - **Speed**
  - **Gate Density**
  - **Development Time**
  - **Prototyping and Simulation Time**
  - **Manufacturing Lead Time**
  - **Future Modifications**
  - **Inventory Risk**
  - **Cost**



Different Categorizations of FPGAs

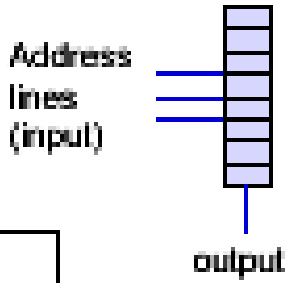
- **Based on Functional Unit/Logic Cell Structure**
  - Transistor Pairs
  - Basic Logic Gates: NAND/NOR
  - MUX
  - Look –up Tables (LUT)
  - Wide-Fan-In AND-OR Gates
- **Programming Technology**
  - Anti-Fuse Technology
  - SRAM Technology
  - EPROM Technology
- **Gate Density**
- **Chip Architecture** (Routing Style)

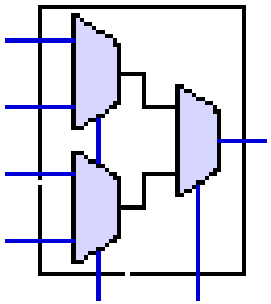


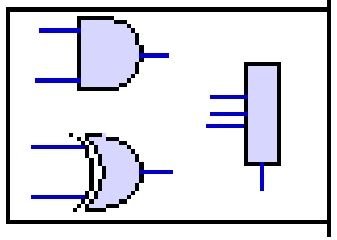
Different Types of Logic Cells

FPGA Architecture: Functional Units

- Functional units
  - RAM blocks (Xilinx): implement function truth table
  - Multiplexers (Actel): build Boolean functions using muxes
  - Logic gates, flip-flops: Such as carry chains. Used for high-performance computations

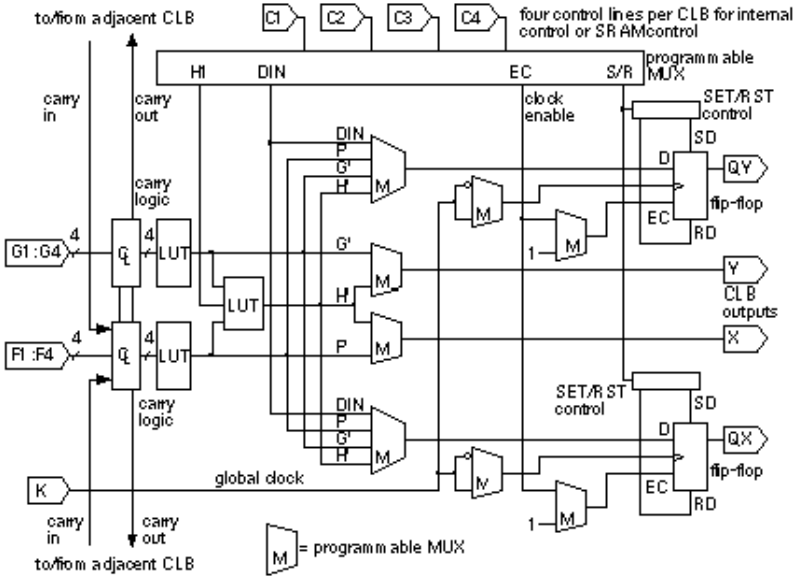


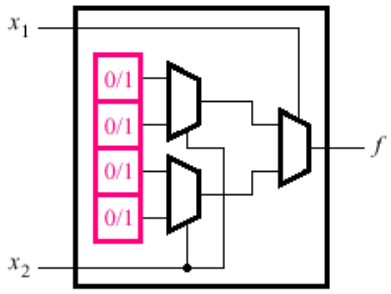




Different Types of Logic Cells

❑ Xilinx XC4000 CLB Structure

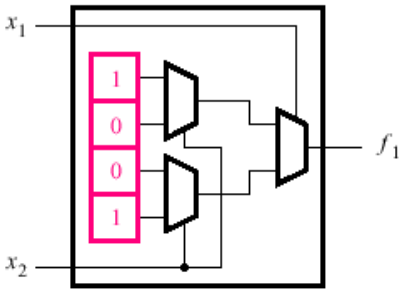




(a) Circuit for a two-input LUT

x_1	x_2	f_1
0	0	1
0	1	0
1	0	0
1	1	1

(b) $f_1 = \bar{x}_1\bar{x}_2 + x_1x_2$

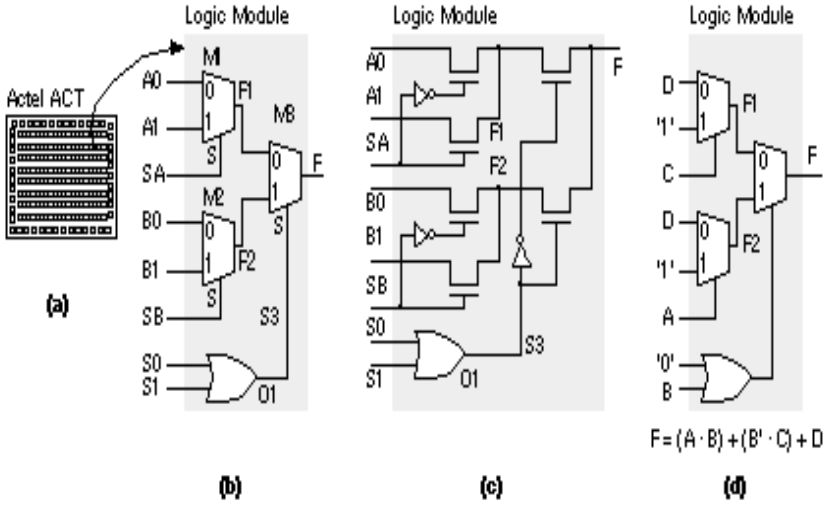


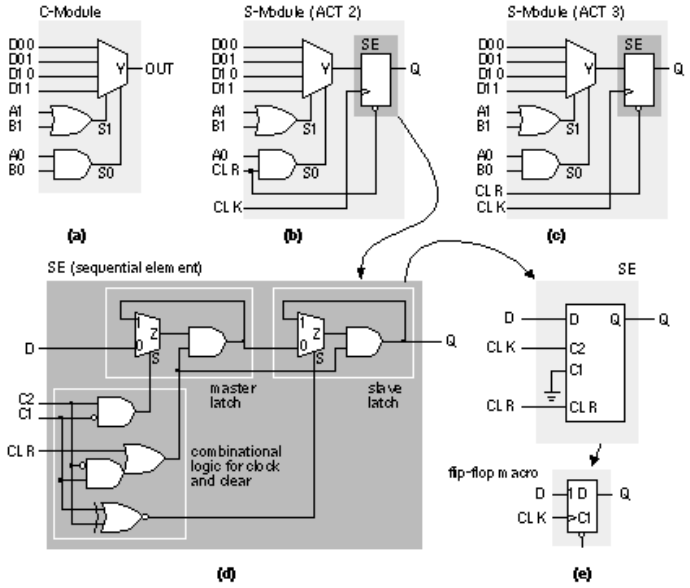
(c) Storage cell contents in the LUT

Different Types of Logic Cells

❑ Actel Act Logic Module Structure

- ❖ Use Antifuse Programming Tech.
- ❖ Based on Channeled GA Architecture
- ❖ Logic Cell is MUX which can be configured as multi-input logic gates



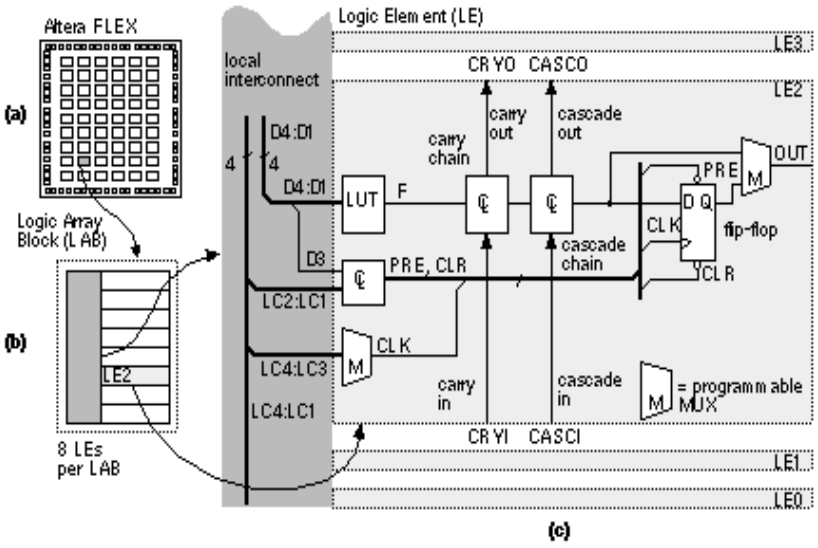


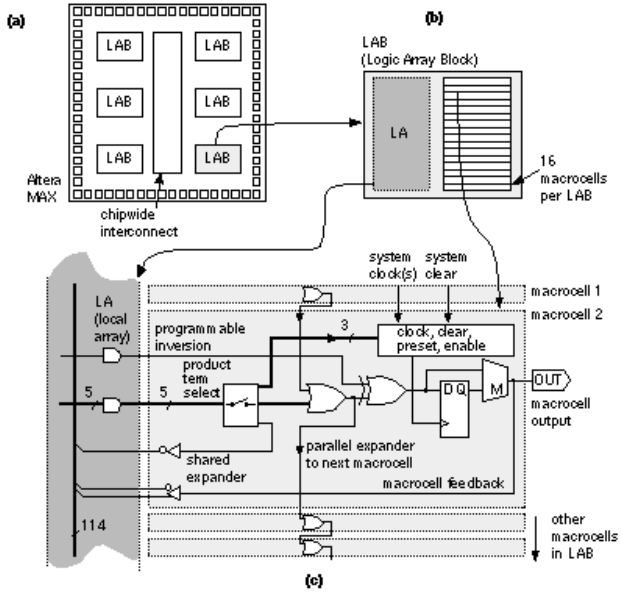
The Actel ACT 2 and ACT 3 Logic Modules. (a) The C-Module for combinational logic. (b) The ACT 2 S-Module. (c) The ACT 3 S-Module. (d) The equivalent circuit (without buffering) of the SE (sequential element). (e) The sequential element configured as a positive-edge–triggered D flip-flop. (Source: Actel.)

Different Types of Logic Cells

❑ Altera Flex / Max Logic Element Structure

❖ **Flex 8k/10k Devices – SRAM Based LUTs, Logic Elements (LEs) are similar to those used in XC5200 FPGA**





The Altera MAX architecture. (a) Organization of logic and interconnect. (b) A MAX family LAB (Logic Array Block). (c) A MAX family macrocell. The macrocell details vary between the MAX families—the functions shown here are closest to those of the MAX 9000 family

Different Types of Logic Cells

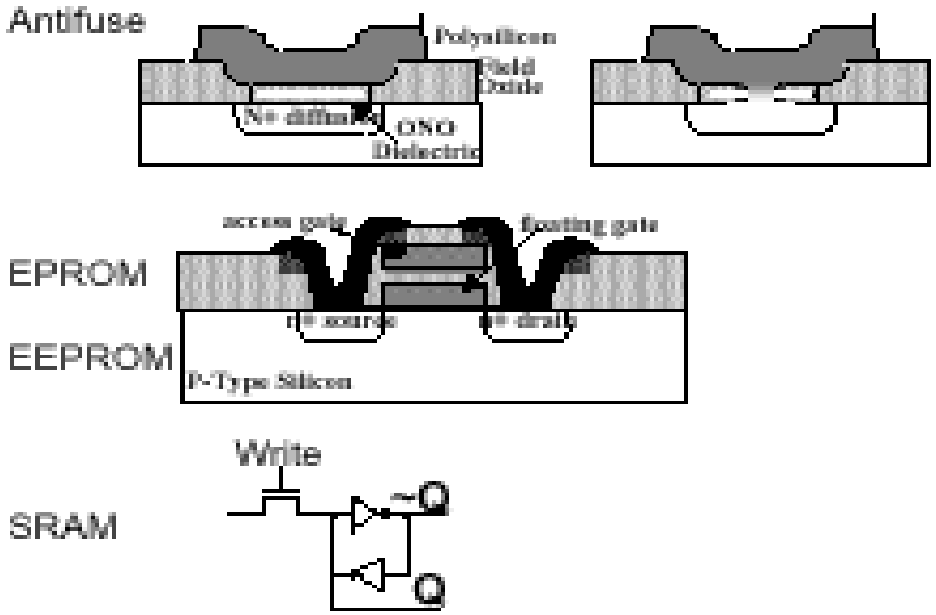
To SUMMARIZE, FPGAs from various vendors differ in their

- **Architecture** **(Row Based or Matrix Based Routing Mechanism)**
- **Gate Density** **(Cap. In Equiv. 2- Input NAND Gates)**
- **Basic Cell Structure**
- **Programming Technology**

Vendor/ Product		Architechture	Capacity	Basic Cell	Programming Technology
Actel		Gate Array	2-8 k	MUX	Antifuse
QuickLogic		Matrix	1.2-1.8 k	MUX	Antifuse
Xilinx		Matrix	2-10 k	RAM Block	SRAM
Altera		Extended PLA	1- 5 k	PLA	EPROM
Concurrent		Matrix	3-5 k	XOR, AND	SRAM
Plessy		SOG	2-40 k	NAND	SRAM

Programming Technologies

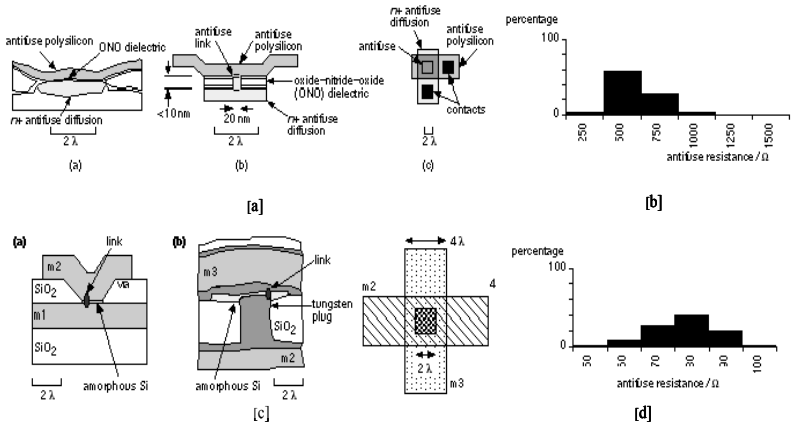
- Three Programming Technologies
  - The Antifuse Technology
  - Static RAM Technology
  - EPROM and EEPROM Technology



Programming Technologies

The Antifuse Technology

- Invented at Stanford and developed by **Actel**
- Opposite to regular fuse Technology
- Normally an open circuit until a programming current (about 5 mA) is forced through it
- Two Types:
  - **Actel's PLICE** [Programmable Low-Impedance Circuit Element]- A High-Resistance Poly-Diffusion Antifuse
  - **QuickLogic's** Low-Resistance metal-metal antifuse [**ViaLink**] technology
    - Direct metal-2-metal connections
    - Higher programming currents reduce antifuse resistance
- Disadvantages:
  - Unwanted Long Delay
  - **OTP** Technology



Actel Antifuse [b] Actel Antifuse Resistance [c] QuickLogic Antifuse [d] QL Antifuse Resistance

Programming Technologies

❑ Static RAM Technology

❖ SRAM cells are used for

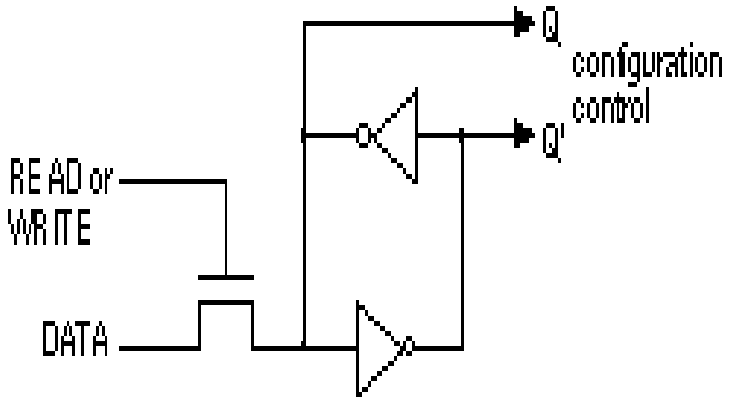
- ✓As Look-Up Tables (LUT) to implement logic (as Truth Tables)
- ✓As embedded RAM blocks (for buffer storage etc.)
- ✓As control to routing and configuration switches

❖ Advantages

- ✓Allows In-System Programming (ISP)
- ✓Suitable for Reconfigurable HW

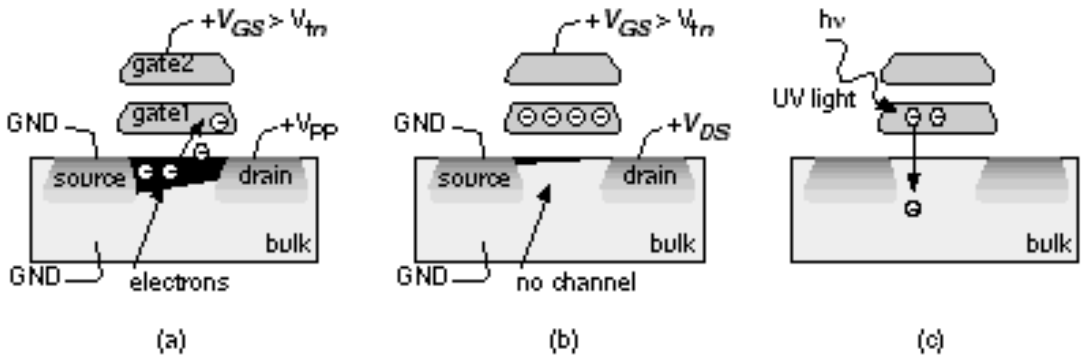
❖ Disadvantages

- ✓Volatile – needs power all the time / use PROM to download configuration data



Programming Technologies

❑ EPROM and EEPROM Technology-



- ❖ **EPROM Cell is almost as small as Antifuse**
- ❖ **Floating-Gate Avalanche MOS (FAMOS) Tech.**
  - ✓Under normal voltage, transistor is on
  - ✓With Programming Voltage applied, we can turn it off (configuration) to implement our logic
  - ✓Exposure to UV lamp (one hour) we can erase the programming
  - ✓Use EEPROM for quick reconfiguration, also, ISP possible

Programming Technologies

❑ Summary Sheet

Programmable ASIC technologies

	Actel	Xilinx LCA[1]	Altera EPLD	Xilinx EPLD
Programming technology	Poly–diffusion antifuse, PLICE	Erasable SRAM ISP	UV-erasable EPROM (MAX 5k) EEPROM (MAX 7/9k)	UV-erasable EPROM
Size of programming element	Small but requires contacts to metal	Two inverters plus pass and switch devices. Largest.	One n-channel EPROM device. Medium.	One n-channel EPROM device. Medium.
Process	Special: CMOS plus three extra masks.	Standard CMOS	Standard EPROM and EEPROM	Standard EPROM
Program-ming method	Special hardware	PC card, PROM, or serial port	ISP (MAX 9k) or EPROM program-mer	EPROM program-mer

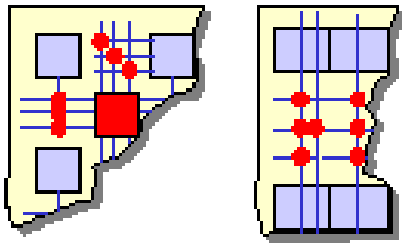
	QuickLogic	Crosspoint	Atmel	Altera FLEX
Programming technology	Metal–metal antifuse, ViaLink	Metal–polysilicon antifuse	Erasable SRAM. ISP.	Erasable SRAM. ISP.
Size of programming element	Smallest	Small	Two inverters plus pass and switch devices. Largest.	Two inverters plus pass and switch devices. Largest.
Process	Special, CMOS plus ViaLink	Special, CMOS plus antifuse	Standard CMOS	Standard CMOS
Program-ming method	Special hardware	Special hardware	PC card, PROM, or serial port	PC card, PROM, or serial port

[1]Lucent (formerly AT&T) FPGAs have almost identical properties to the Xilinx LCA family

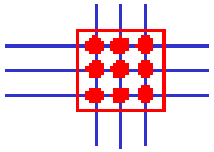
Chip Architecture or Routing Style

Programmable Switch Elements

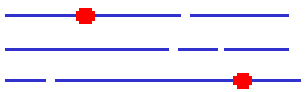
- Used in connecting:
  - The I/O of functional units to the wires



  - A horizontal wire to a vertical wire

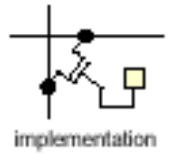


  - Two wire segments to form a longer wire segment



Programmable Switch Elements: Implementation

- SRAM connected to the gate of a transistor (Xilinx)

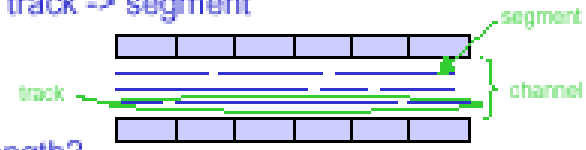


- Fuse / Anti Fuse (Actel)

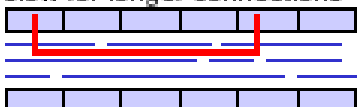
Chip Architecture or Routing Style

Routing Channels

- Note: fixed channel widths (tracks)
- Should "predict" all possible connectivity requirements when designing the FPGA chip
- Channel -> track -> segment

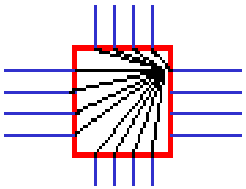


- Segment length?
  - Long: carry the signal longer, less "concatenation" switches, but might waste track
  - Short: local connections, slow for longer connections

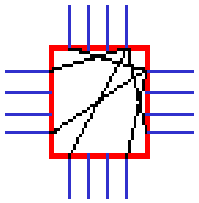


Switch Boxes

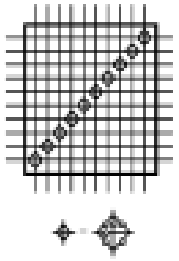
- Ideally, provide switches for all possible connections



- Trade-off:
  - Too many switches:
    - Large area
    - Complex to program
  - Too few switches:
    - Cannot route signals



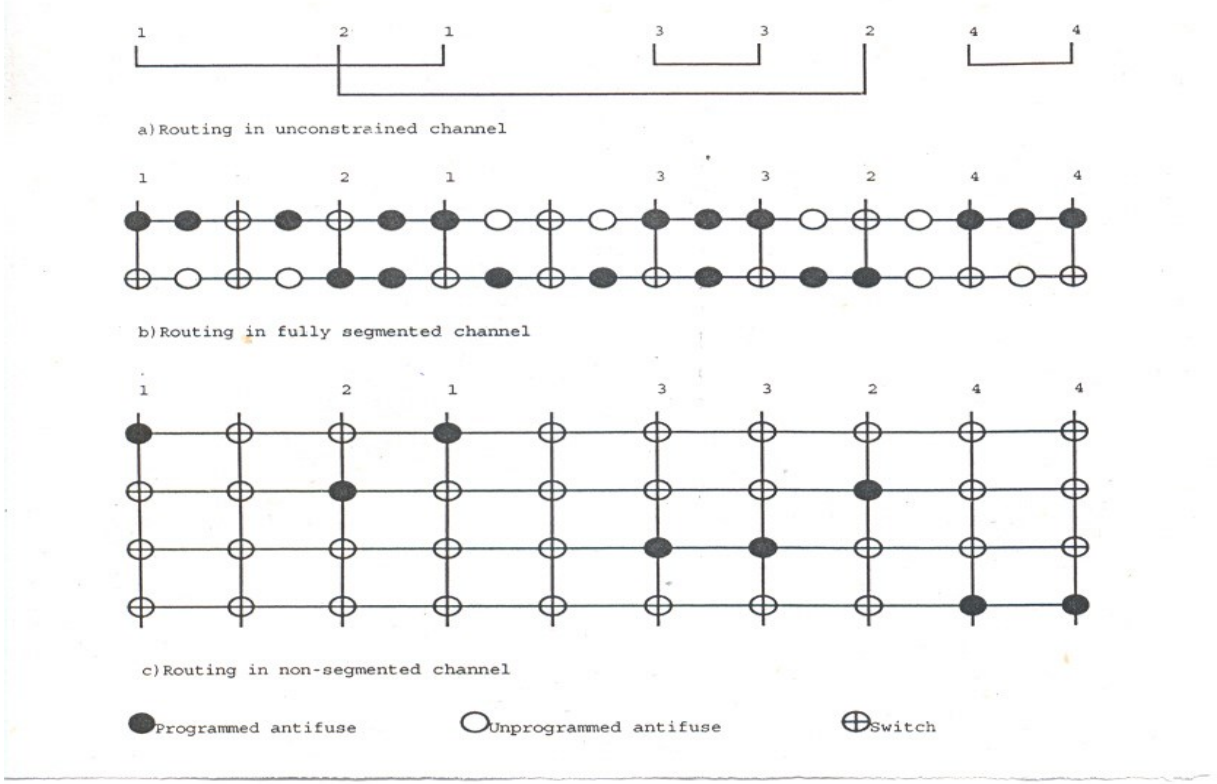
One possible solution



Xilinx 4000

Chip Architecture or Routing Style

- ***Trade-off between Longer and Shorter Tracks Explained Through Example***



ASIC Design Process

S-1 Design Entry: Schematic entry or HDL description

S-2: Logic Synthesis: Using Verilog HDL or VHDL and Synthesis tool, produce ***a netlist***-logic cells and their interconnect detail

S-3 System Partitioning: Divide a large system into ASIC sized pieces

S-4 Pre-Layout Simulation: Check design functionality

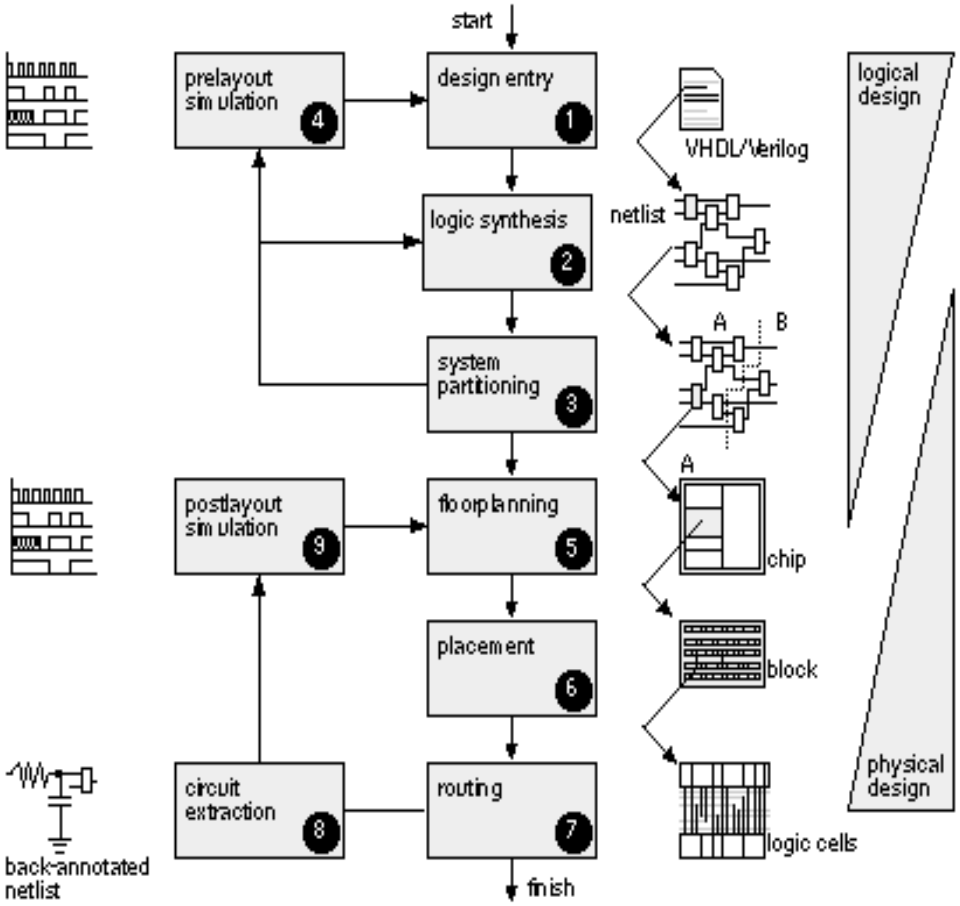
S-5 Floorplanning: Arrange netlist blocks on the chip

S-6 Placement: Fix cell locations in a block

S-7 Routing: Make the cell and block interconnections

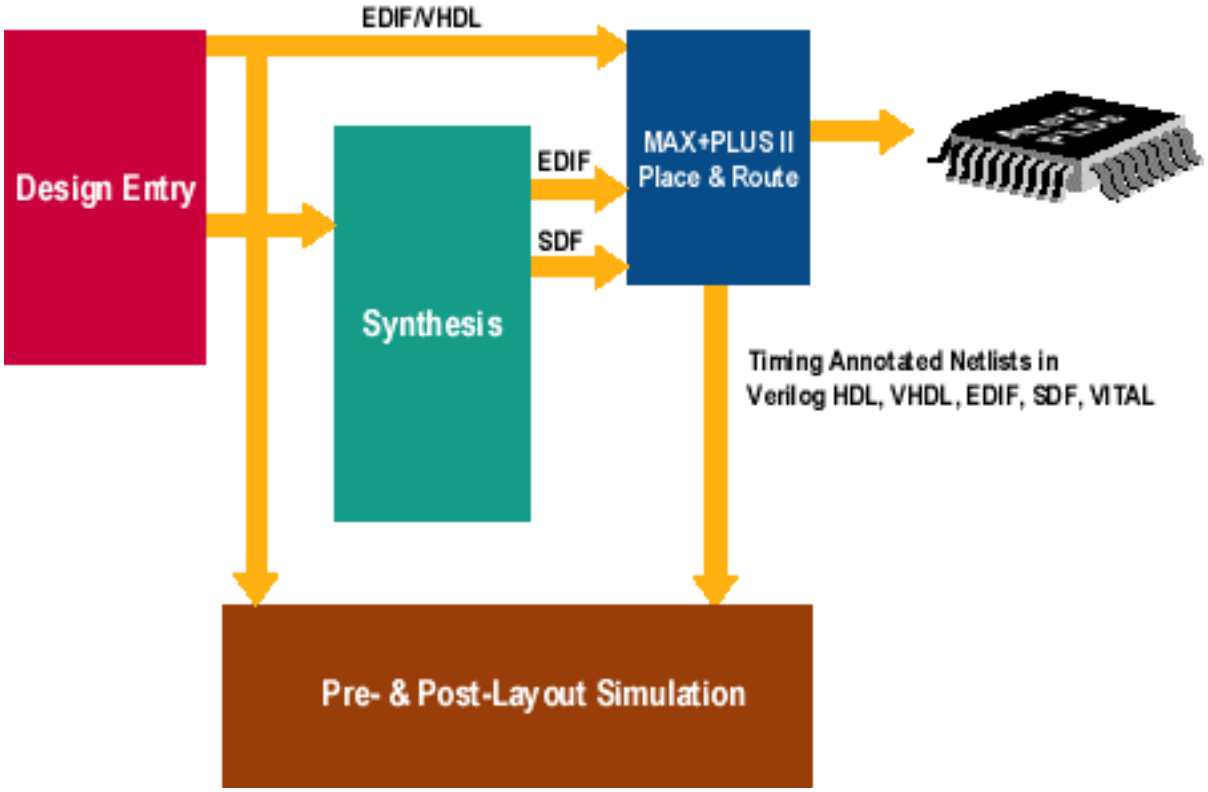
S-8 Extraction: Measure the interconnect R/C cost

S-9 Post-Layout Simulation



ASIC Design Process

❑ **Altera FPGA Design Flow –** **A Self-Contained System that does all from Design Entry, Simulation, Synthesis, and Programming of Altera Devices**



ASIC Design Process

- **Xilinx FPGA Design Flow –** **Allows Third-Party Design Entry SW, Accepts their generated netlist file as an input**
  - **Use Pin2xnf and wir2xnf SW to convert the netlist file to .XNF**
  - **xnfmap and xnfmerge programs convert .xnf files to create a unified netlist file (Nand/Nor Gates) .MAP file are generated**
  - **map2lca program does fitters job, produces un-routed .LCA file**
  - **apr or ppr SW does the routing job, post-layout netlist generated**
  - **makebits SW generates BIT files**