ADVANCED FPGA BASED SYSTEM DESIGN

Dr. Tayab Din Memon

tayabuddin.memon@faculty.muet.edu.pk

Lecture 3 & 4

Books

- Recommended Books:
- Text Book: FPGA Based System Design by Wayne Wolf

Overview

- Why VLSI?
- Moore’s Law.
- Why FPGAs?
- The VLSI and system design process.

Why VLSI?

- Integration improves the design:
  - lower parasitics = higher speed;
  - lower power;
  - physically smaller.
- Integration reduces manufacturing cost-(almost) no manual assembly.

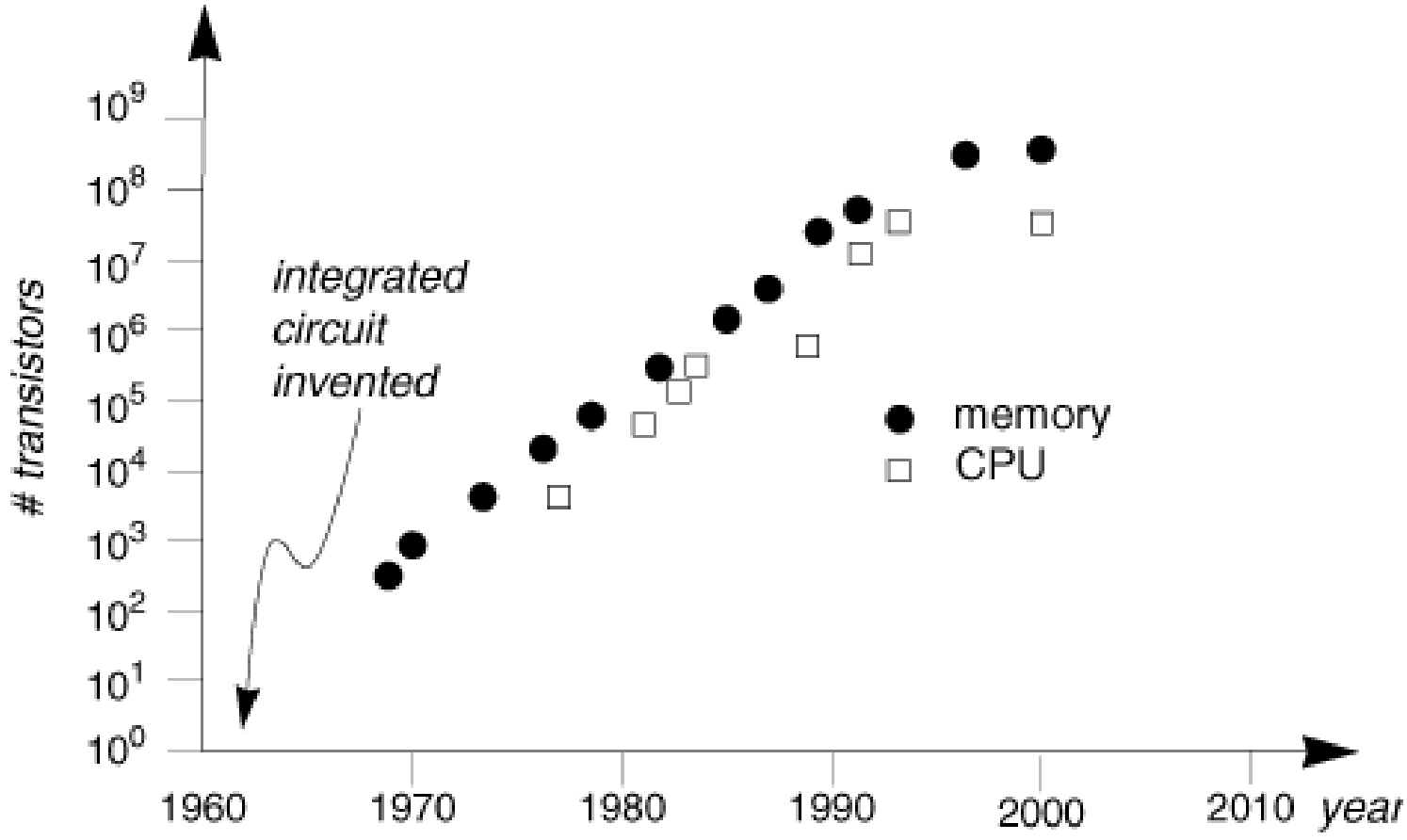
VLSI and you

- Microprocessors:
  - personal computers;
  - microcontrollers.
- DRAM/SRAM/flash.
- Audio/video and other consumer systems.
- Telecommunications.
- FPGAs, ASICS, ASIP, and etc.

Moore's Law

- Gordon Moore: co-founder of Intel.
- Predicted that number of transistors per chip would grow exponentially (double every 18 months).
- Exponential improvement in technology is a natural trend: steam engines, dynamos, automobiles.

Moore's Law plot



The cost of fabrication

- Current cost: $2-3 billion.
- Typical fab line occupies about 1 city block, employs a few hundred people.
- New fabrication processes require 6-8 month turnaround.
- Most profitable period is first 18 months-2 years.

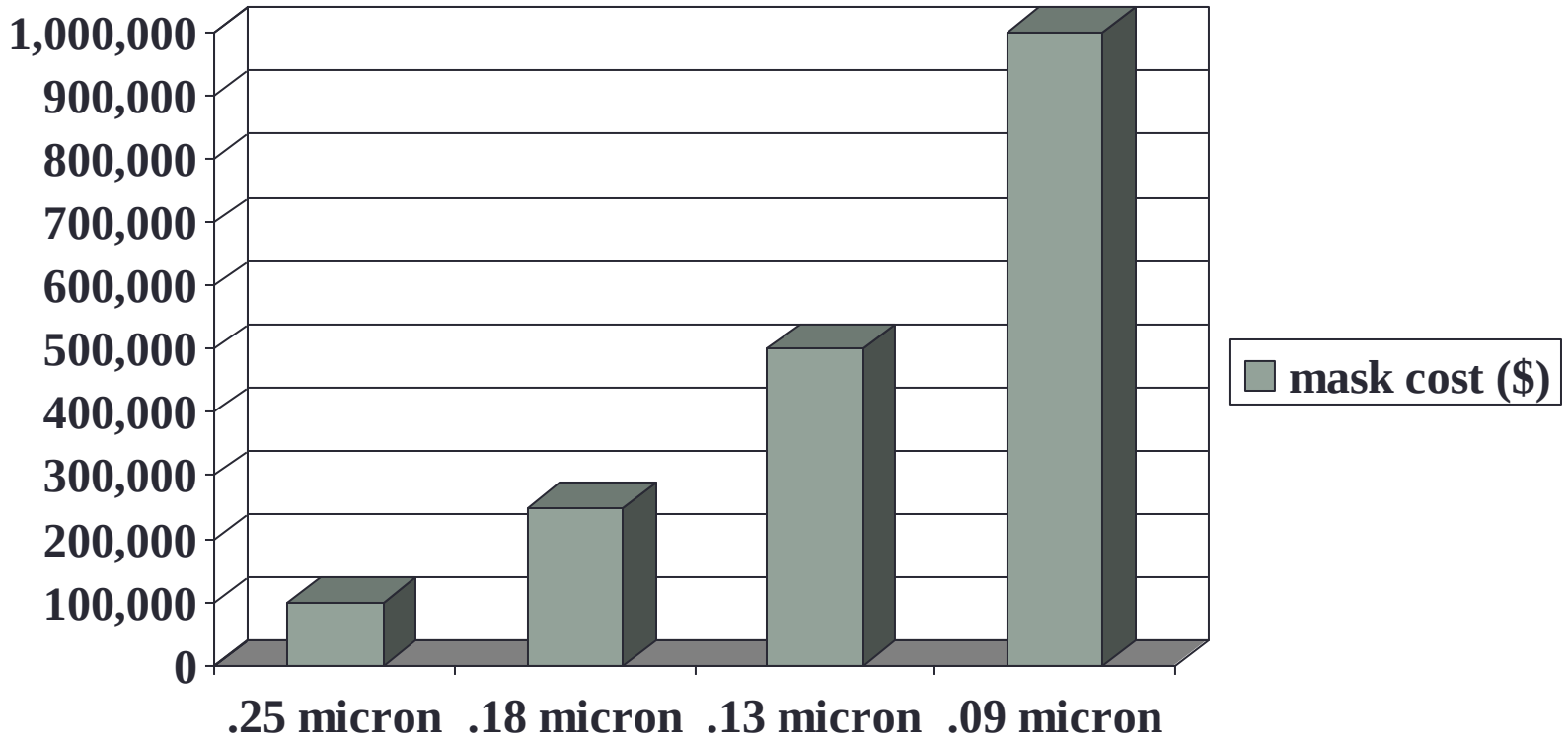
Cost factors in ICs

- For large-volume ICs:
  - packaging is largest cost;
  - testing is second-largest cost.
- For low-volume ICs, design costs may swamp all manufacturing costs.
  - $10 million-$20 million.

Packaging

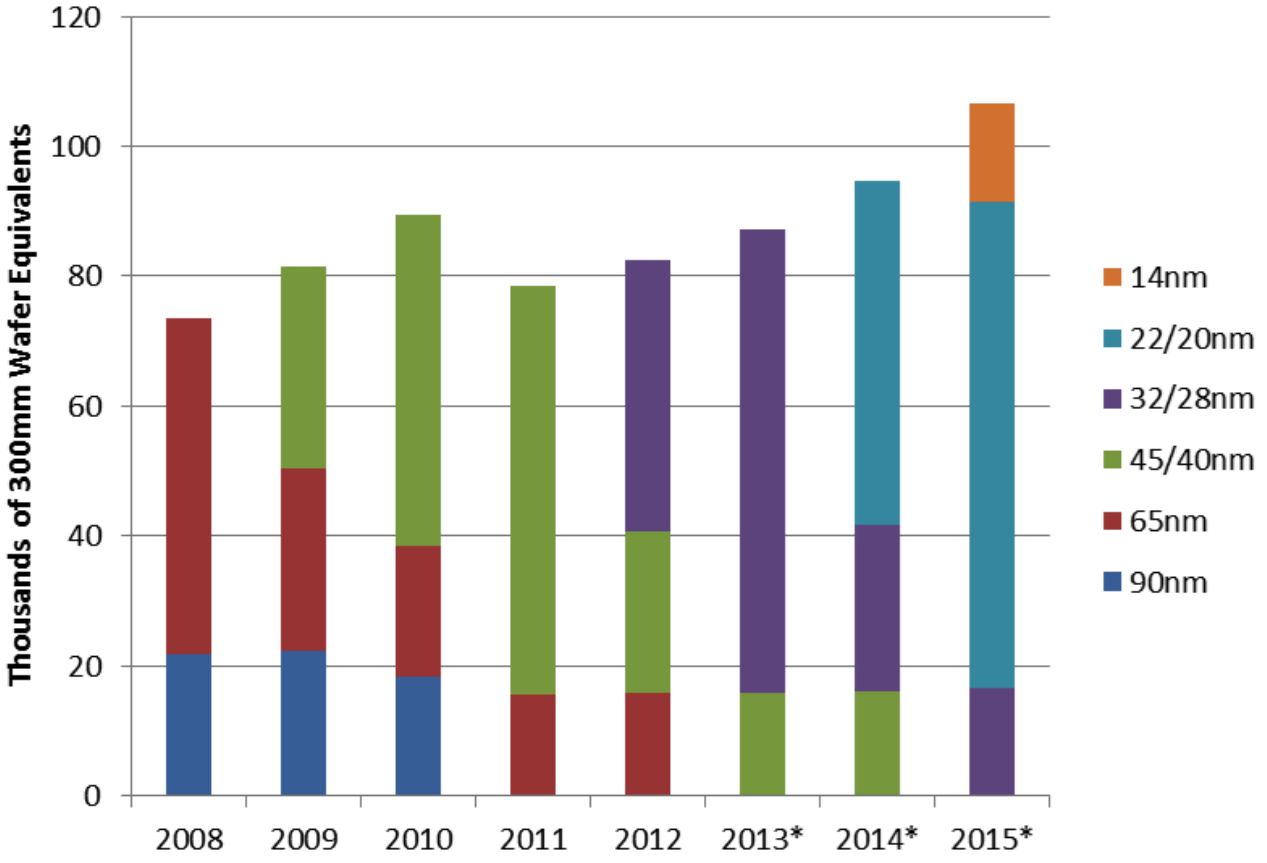
- In electronics manufacturing, **integrated circuit packaging** is the final stage of semiconductor device fabrication, in which the tiny block of semiconducting material is encased in a supporting case that prevents physical damage and corrosion. The case, known as a "package", supports the electrical contacts which connect the device to a circuit board.
- In the integrated circuit industry it is called simply **packaging** and sometimes **semiconductor device assembly**, or simply **assembly**. Sometimes it is called **encapsulation** or **seal**. The packaging stage is followed by testing of the integrated circuit [***Source: Wikipedia***].

Mask cost vs. line width



Beyond Moore's Law

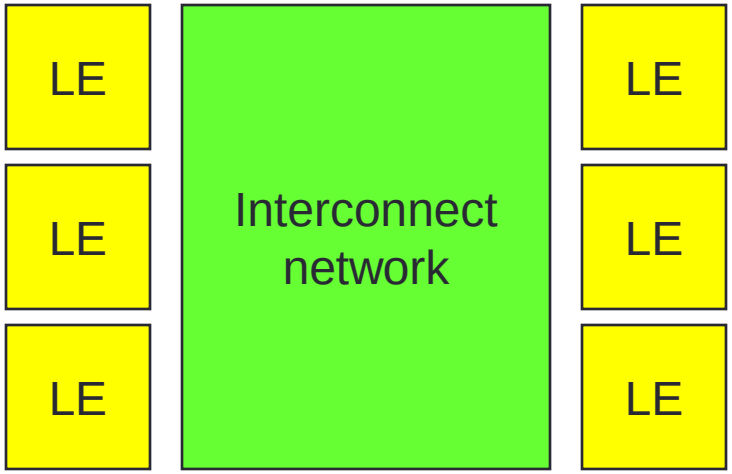
Beyond Moore's Law: 20nm FPGA



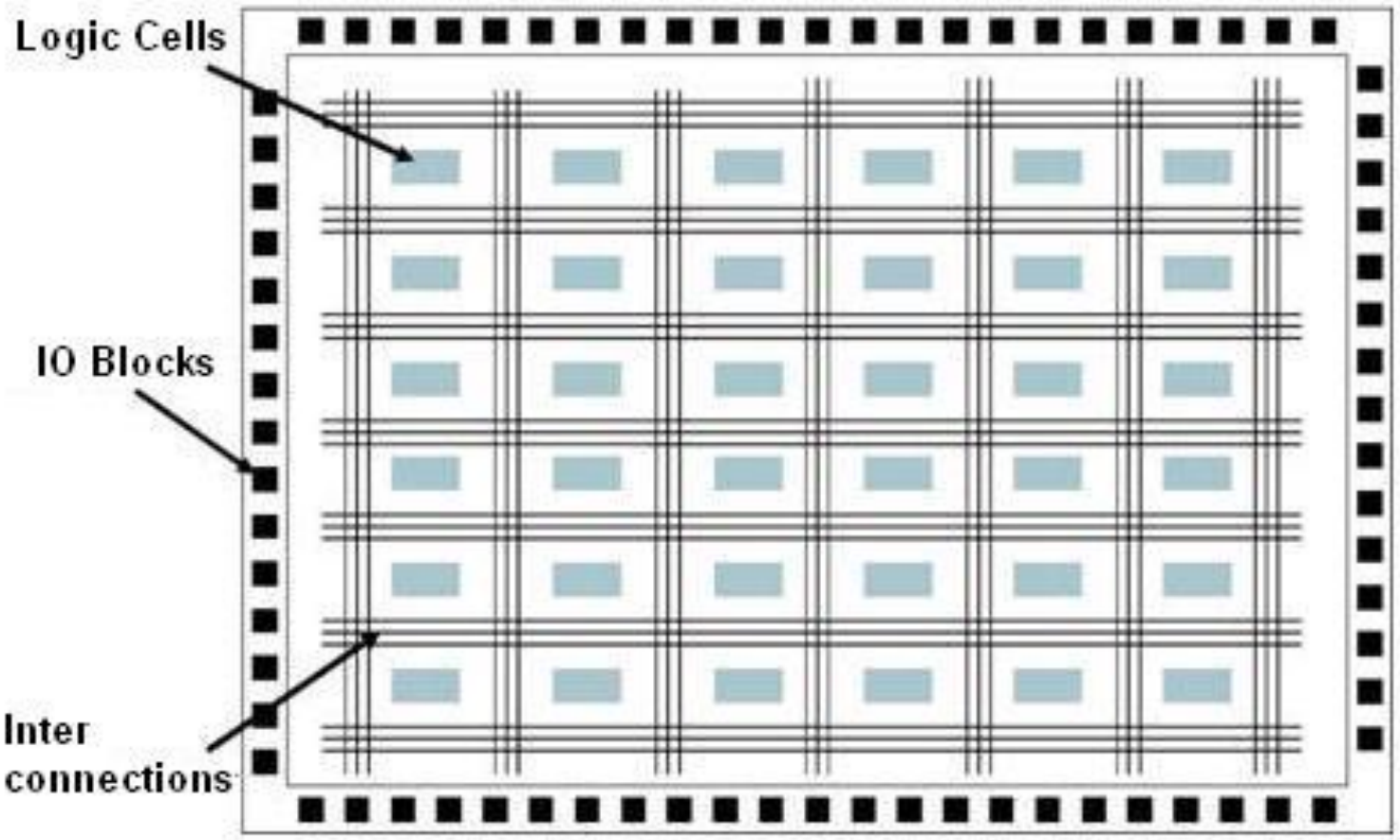
Source: Semico MAP Model

Field-programmable gate arrays

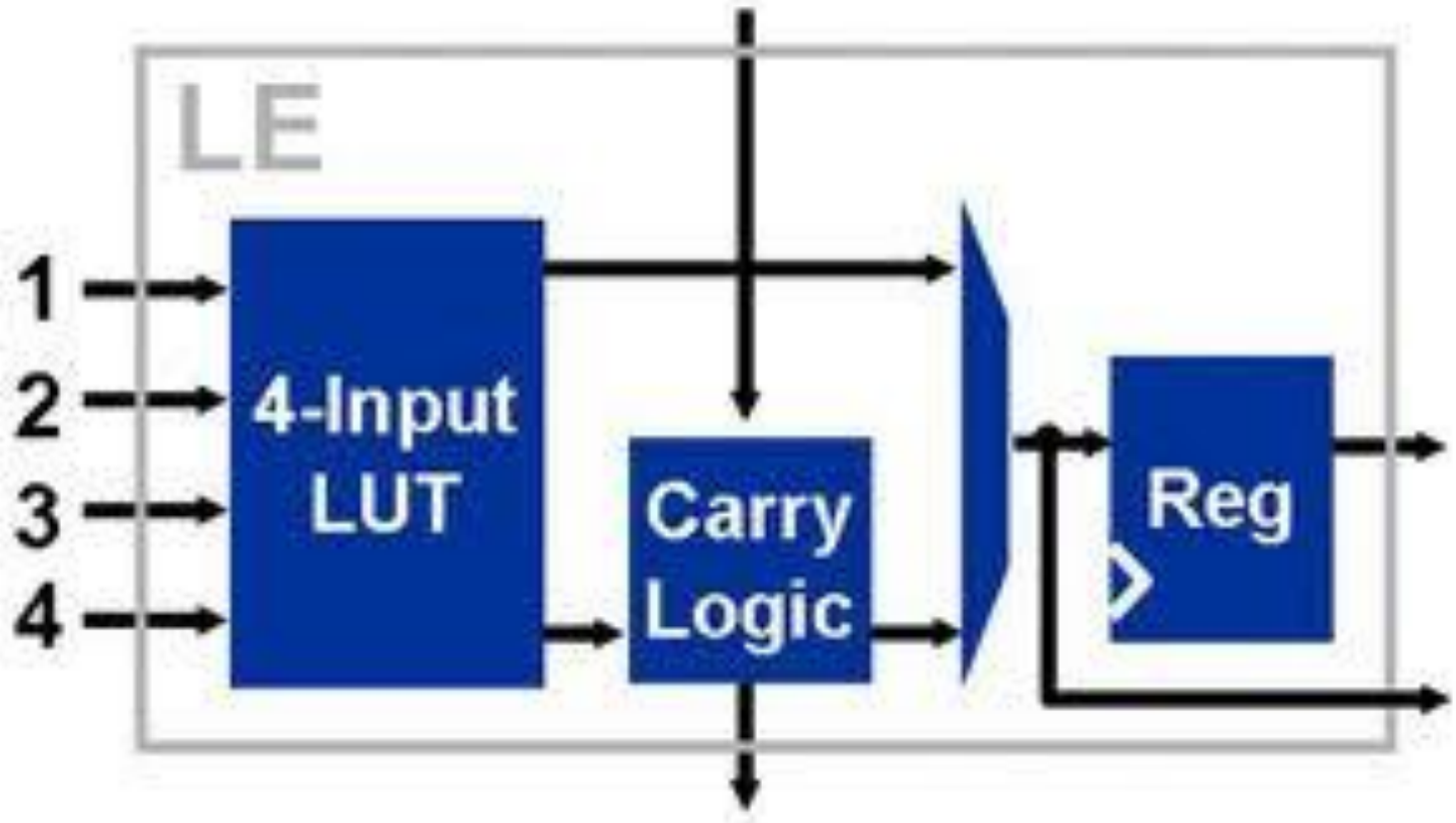
- FPGAs are programmable logic devices:
  - Logic elements + interconnect.
  - Provide multi-level logic.



FPGA Architecture in a Glance



Logic Element



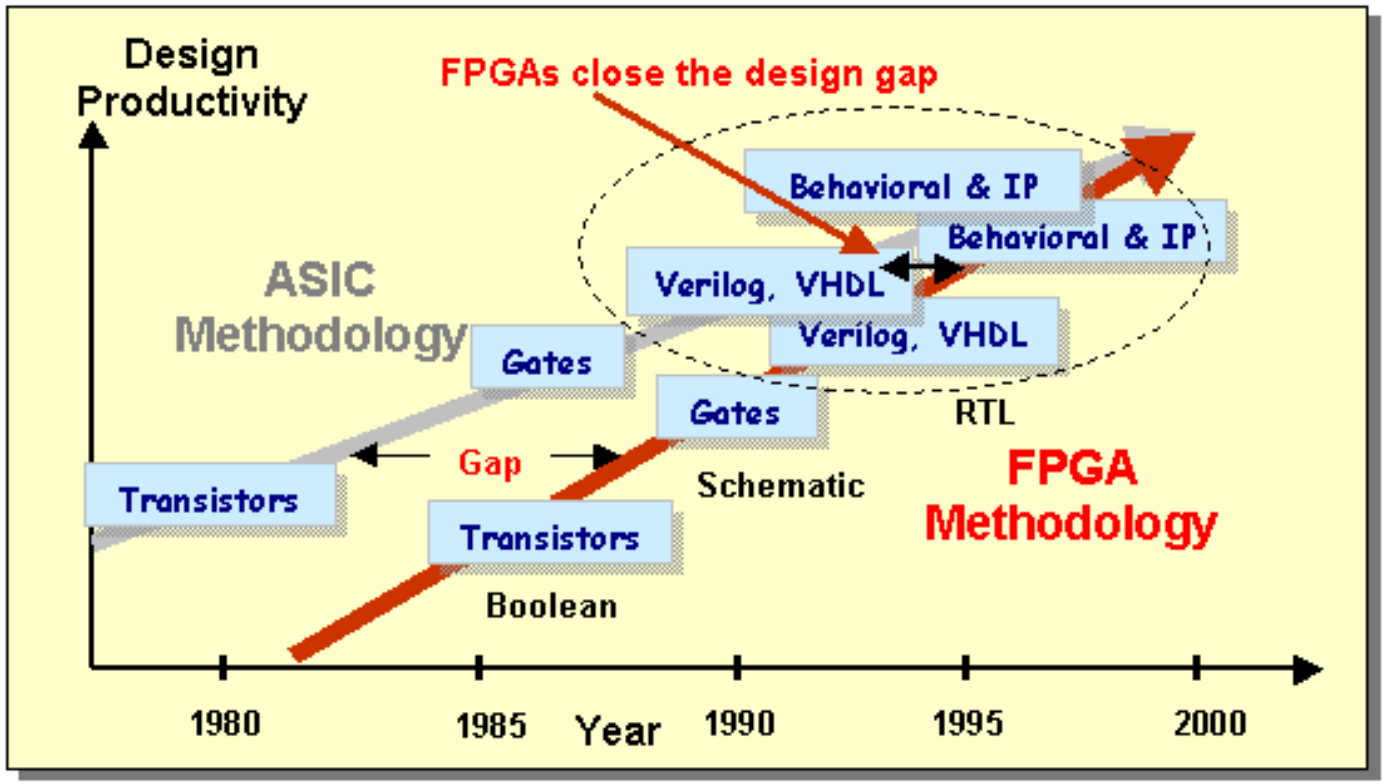
FPGAs and ASIC

- FPGAs are standard parts:
  - Pre-manufactured.
  - Don’t worry (much) about physical design.
- Custom silicon:
  - Tailored to your application.
  - Generally lower power consumption.
- Revisit:
  - FPGAs use a grid of logic gates, and once stored, the data doesn't change, similar to that of an ordinary gate array.
  - The term "*field-programmable*" means the device is programmed by the customer, not the manufacturer [Source: Wikipedia].

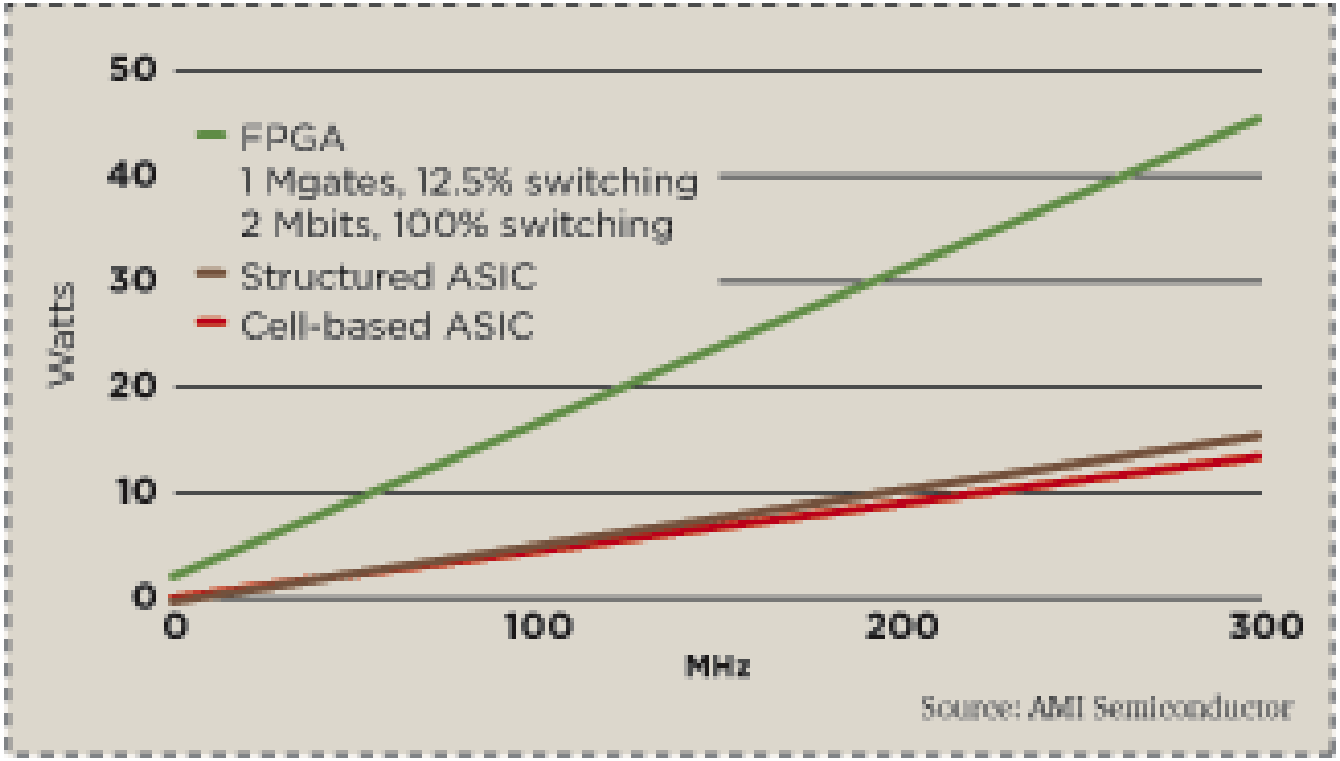
Standard parts vs. custom

- Do you build your system with an FPGA or with custom silicon?
  - FPGAs have shorter design cycle.
  - FPGAs are great for prototyping and low-volume production
  - FPGAs have no manufacturing delay.
  - FPGAs reduce inventory.
  - FPGAs are slower, larger, more power-hungry.
  - however, that any relatively complex mid- to high-volume design for which power consumption, component cost and size are important issues requires another solution for mass production, where ASIC comes in place (i.e., Custom Silicon)

Standard parts vs. custom

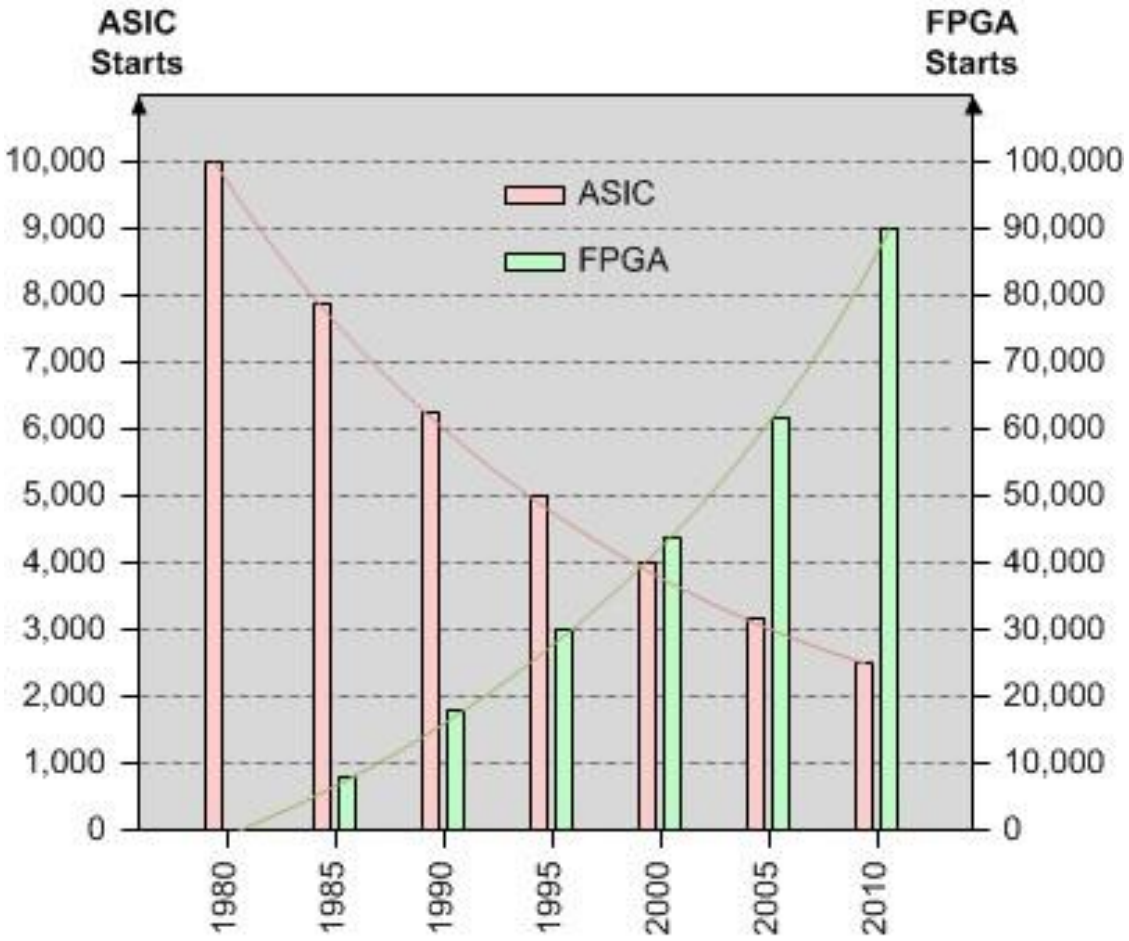


Standard parts vs. custom



Considering all of the power-saving options, it is common for a structured ASIC design to use 20 to 50 percent less power than the same design implemented in an FPGA.

Standard parts vs. custom



ASIC versus FPGA design starts

(This is totally fictitious data of no value whatsoever)

FPGA Based System Design (Goals and Techniques)

- **Performance** – The logic must run at a required rate. Performance may be measured in throughput and latency
- **Power/energy** – The chip must often run within an energy or power-budget. Too critical in battery-powered systems
- **Design time** – you can't take forever to design the system. FPGAs, because they are standard parts, have several advantages in design time. They can be used as prototype, can be programmed quickly, and they can be used as parts in final design

FPGA Based System Design (Goals and Techniques)

- **Design Cost** – Design time is one important component of design cost, but other factors, such s the required support tools, may be a consideration. FPGA tools are often less expensive than custom VLSI tools.

- **Manufacturing cost** – manufacturing cost of FPGAs (i.e., replicating the system many times) is generally more expensive than ASICs. However, the fact they are standard parts helps to reduce their cost.

Programming technologies

- SRAM.
  - Can be programmed many times.
  - Must be programmed at power-up.
- Antifuse.
  - Programmed once.
- Flash.
  - Similar to SRAM but using flash memory.

Challenges in system design

- Multiple levels of abstraction: starting from specification to architecture, and logic design.
- Multiple and conflicting constraints: low cost and high performance are often at odds.
- Short design time: Late products are often irrelevant.

The system design process

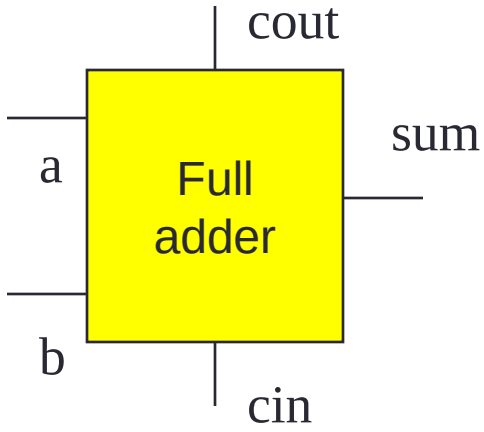
- May be part of larger product design.
- Major levels of abstraction:
  - specification;
  - architecture;
  - logic design;
  - circuit design;
  - layout.

Dealing with complexity

- Divide-and-conquer: limit the number of components you deal with at any one time.
- Group several components into larger components:
  - transistors form gates;
  - gates form functional units;
  - functional units form processing elements;
  - etc.

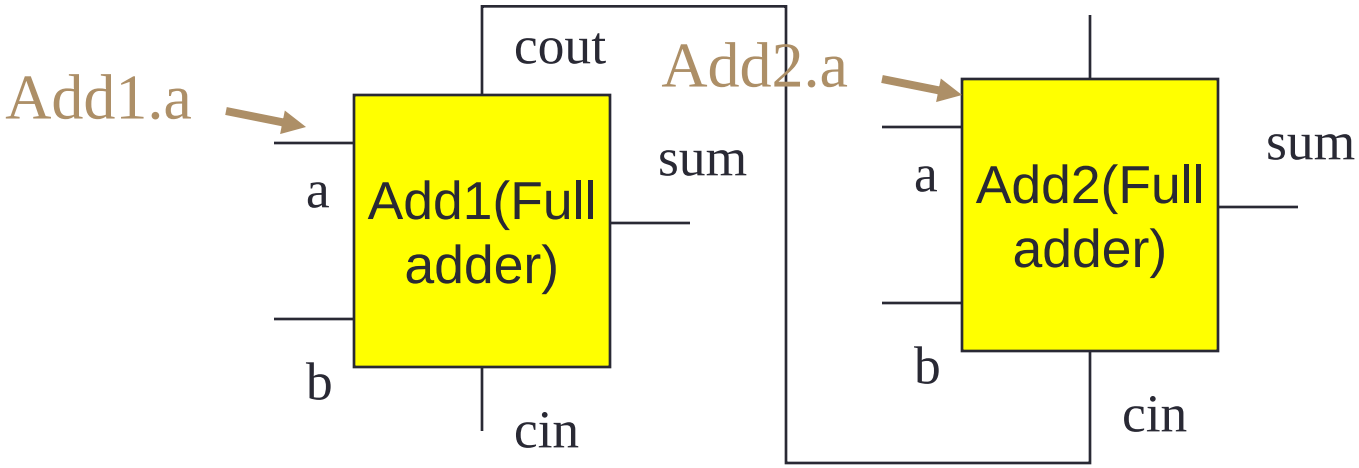
Hierarchical name

- Interior view of a component:
  - components and wires that make it up.
- Exterior view of a component = type:
  - body;
  - pins.

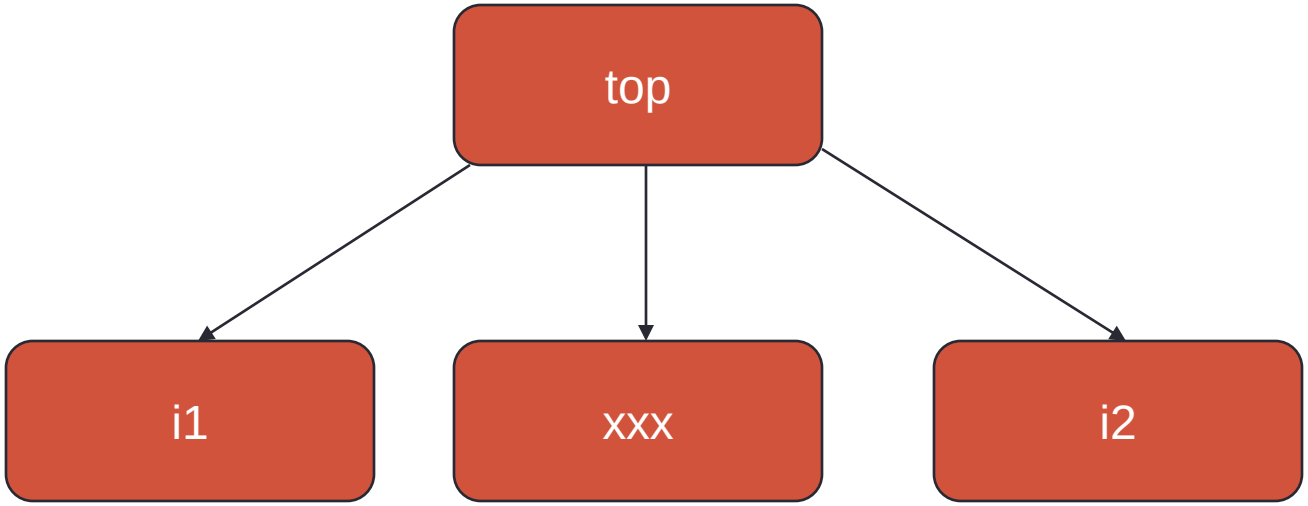


Instantiating component types

- Each instance has its own name:
  - add1 (type full adder)
  - add2 (type full adder).
- Each instance is a separate copy of the type:

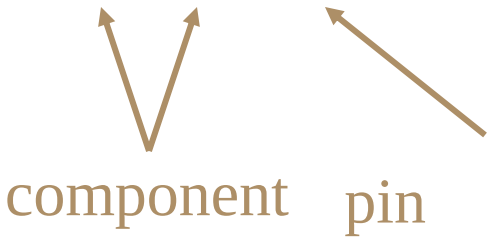


Component hierarchy



Hierarchical names

- Typical hierarchical name:
  - top/i1.foo

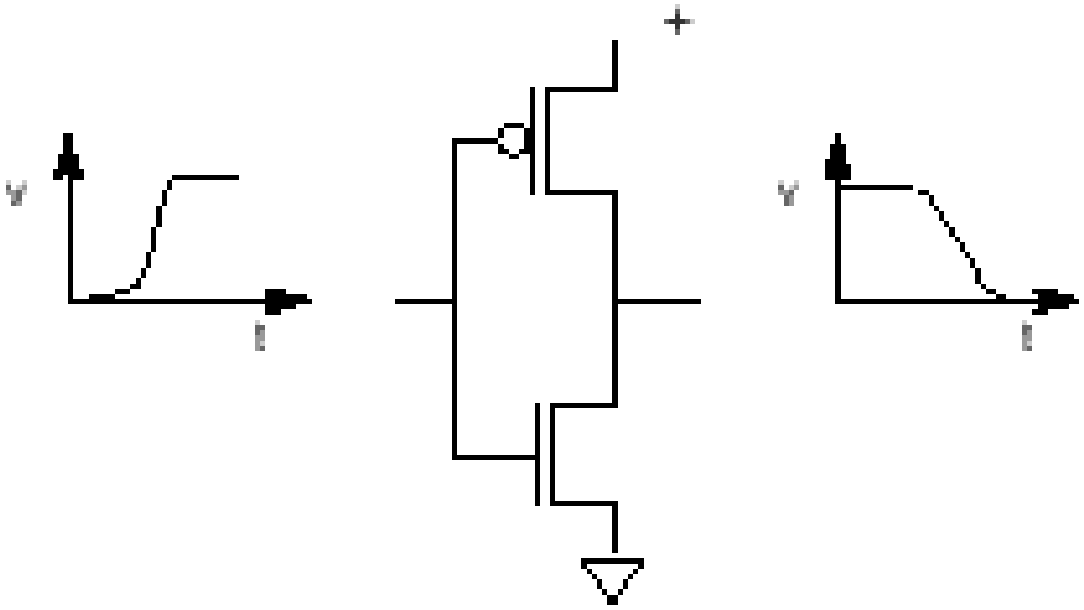


Levels of abstraction

- Specification: function, cost, etc.
- Architecture: large blocks.
- Logic: gates + registers.
- Circuits: transistor sizes for speed, power.
- Layout: determines parasitics.

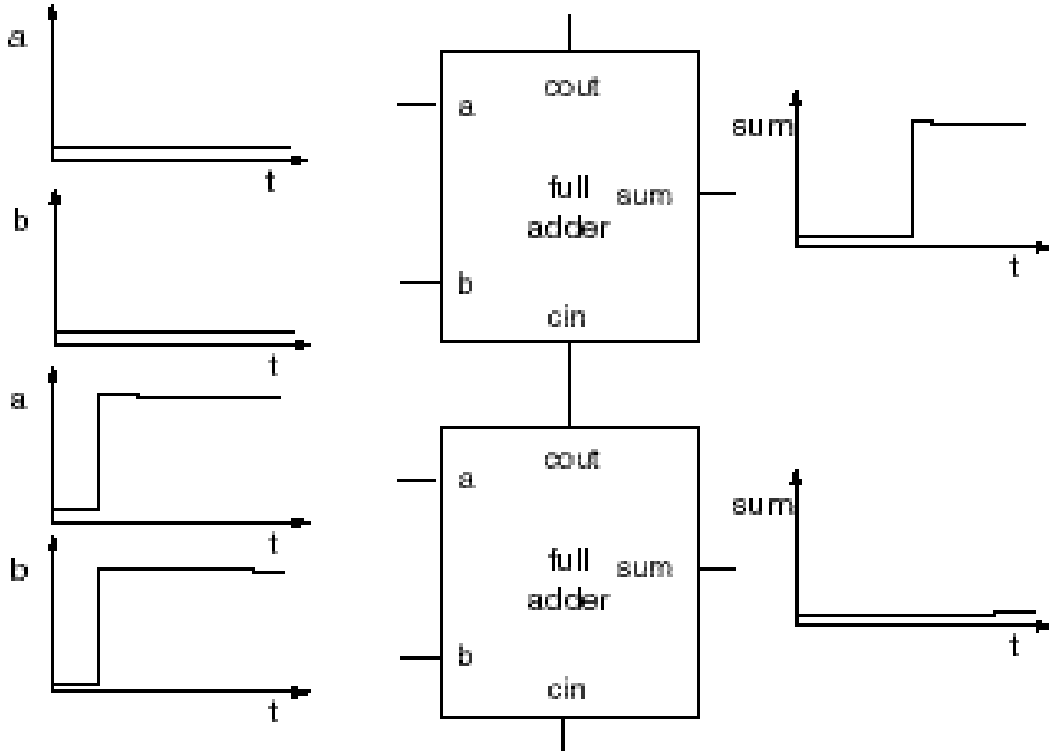
Circuit abstraction

- Continuous voltages and time:



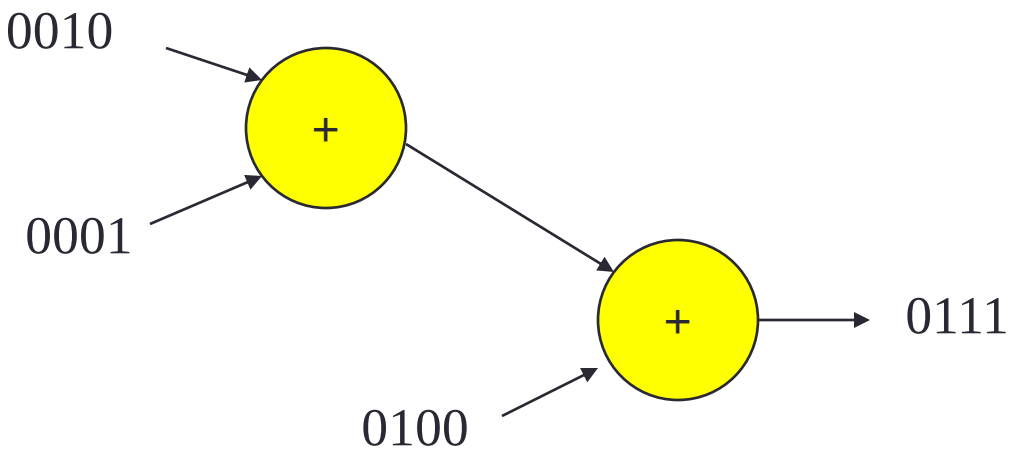
Digital abstraction

- Discrete levels, discrete time:



Register-transfer abstraction

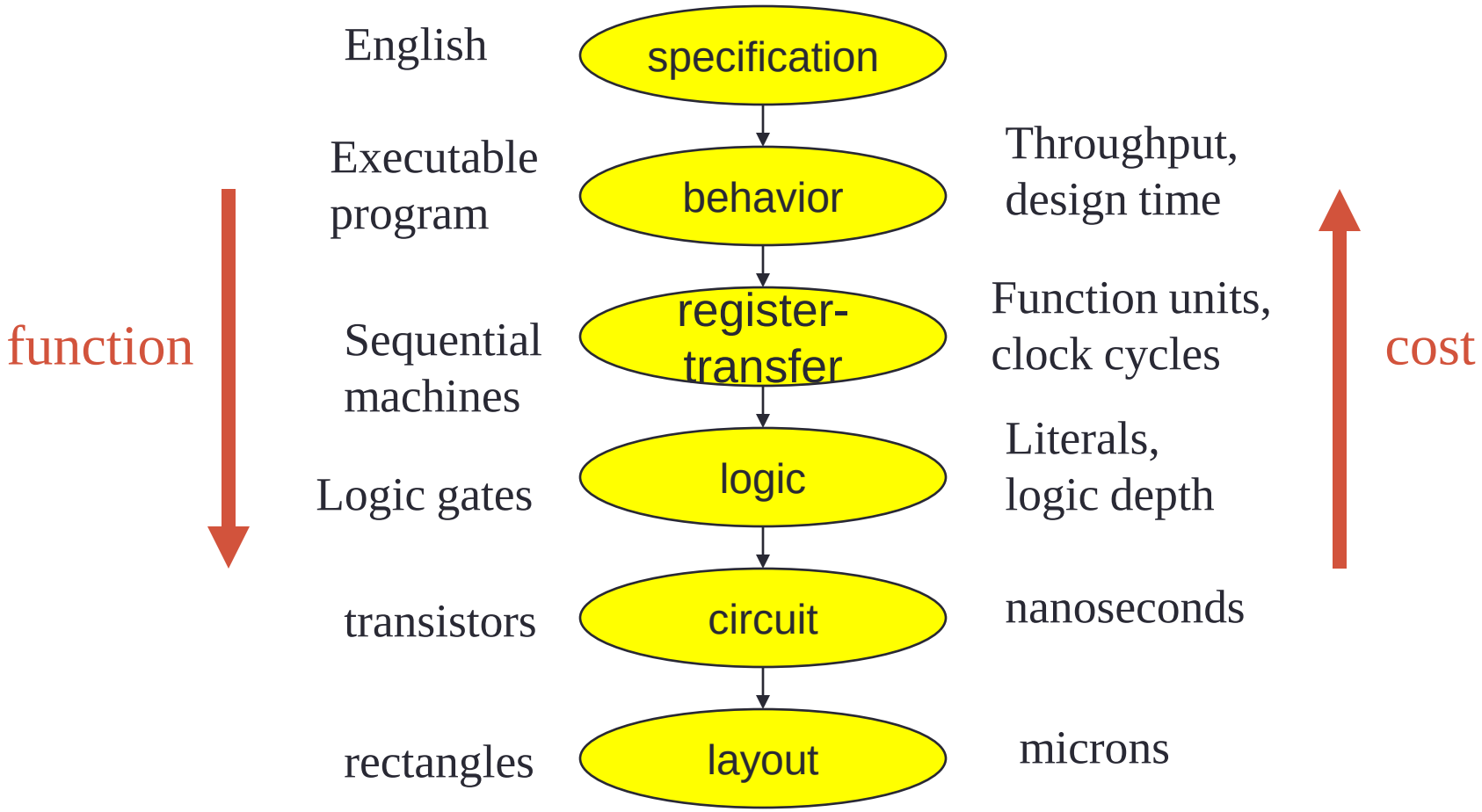
- Abstract components, abstract data types:



Top-down vs. bottom-up design

- Top-down design adds functional detail.
  - Create lower levels of abstraction from upper levels.
- Bottom-up design creates abstractions from low-level behavior.
- Good design needs both top-down and bottom-up efforts.

Design abstractions



Explanation

- **Behavior** – A detailed, executable description of what the chip should do, but not how it should do it
- **Register-transfer** – The system's time behavior is fully-specified – we know the allowed input and output values on every clock cycle – but the logic isn't specified as gates
- **Logic** – the system designed in terms of Boolean logic gates, latches, and flip-flops.
- **Configuration** – the logic must be placed into logic elements around the FPGA and the proper connections must be made between those logic elements. Placement and routing perform these important steps.

END OF THE LECTURE 3 & 4