Chapter 1
Overview of Digital Systems Design
SKEE2263 Digital Systems

Mun'im Zabidi {munim@utm.my}
Ismahani Ismail {ismahani@fke.utm.my}
Izam Kamisian {e-izam@utm.my}

Faculty of Electrical Engineering, Universiti Teknologi Malaysia

February 11, 2018

Why Digital Design?

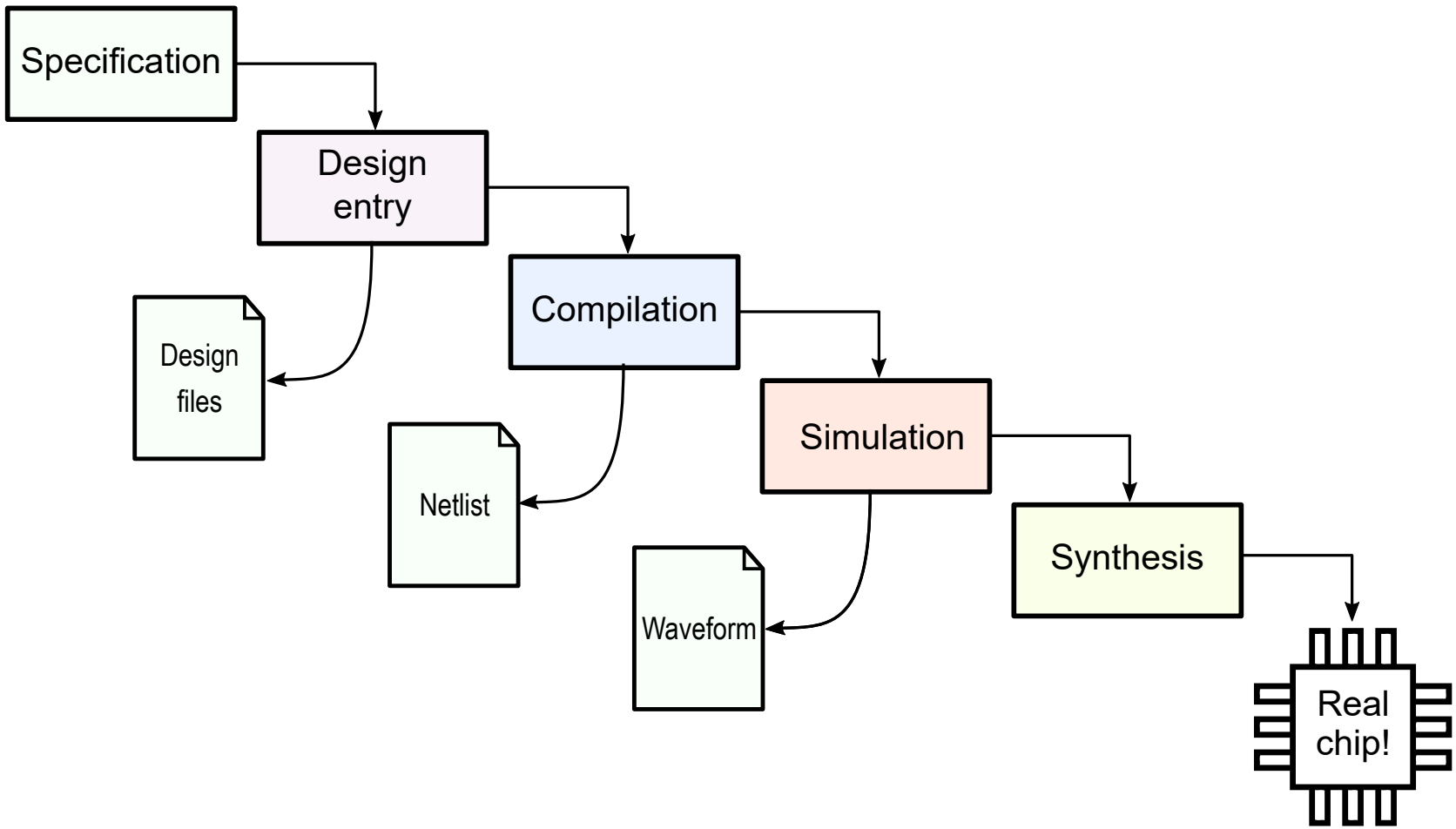
Many times, microcontrollers are not powerful enough

- For solving many engineering problems, microcontrollers are cheap and sufficient
- Need to know the best solution for the problem at hand

Design options for digital systems

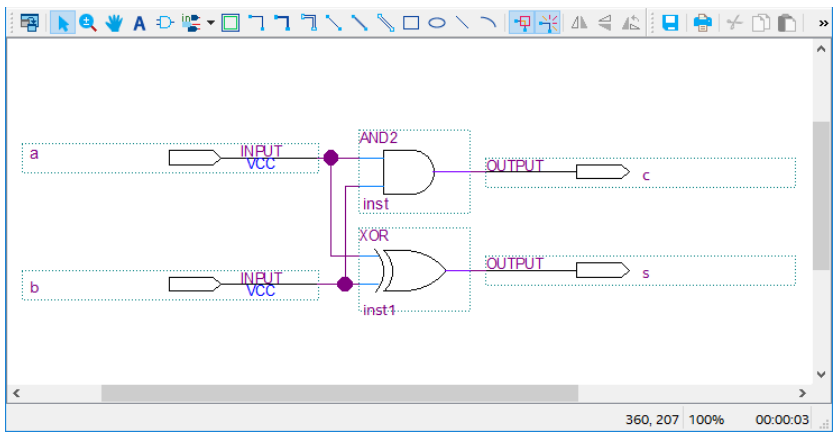
Device	Advantages	Shortcomings
Desktop processor	Familiar	Big and uses much power
Microcontroller	Low cost	May not have enough performance
Custom digital circuit	Highest performance	Require specialized design

Digital Design Process



Design Entry

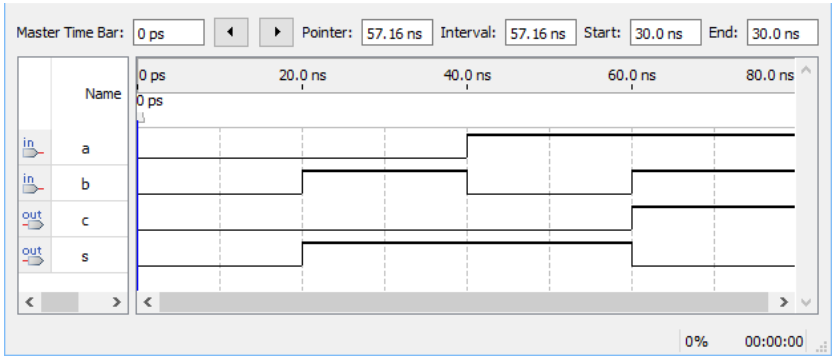
- Schematic capture
  - Graphical based
  - Only suitable for small designs



- Hardware Description language (HDL)
  - Text-based
  - Just like computer programming

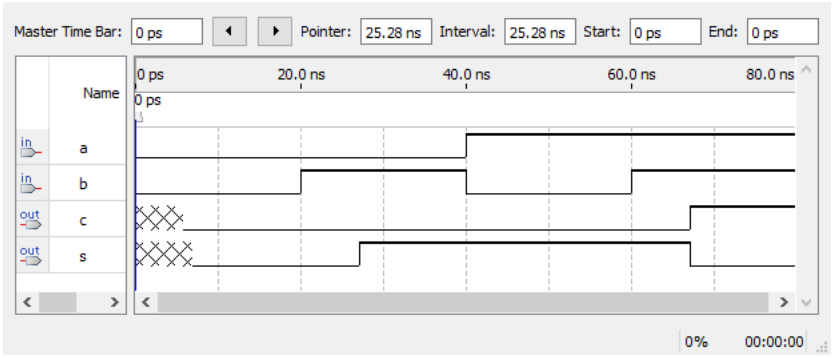
```
module ha(a,b,s,c);
    input a,b;
    output s,c;

    xor u1(s,a,b);
    and u2(c,a,b);
endmodule
```

Simulation



Functional simulation

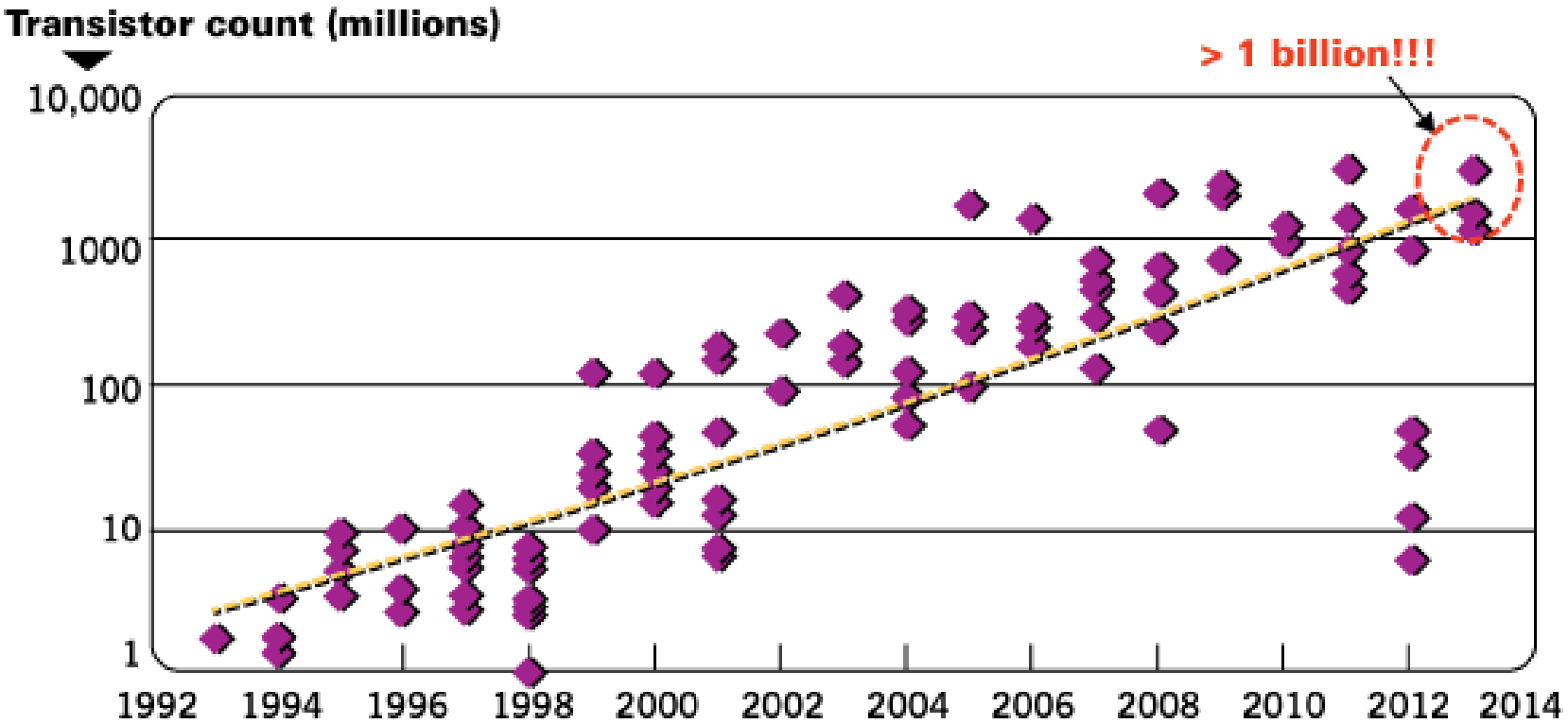
Faster
Device-independent



Timing simulation

Delays are simulated accurately
Must be “fitted” to target chip

Moore's Law

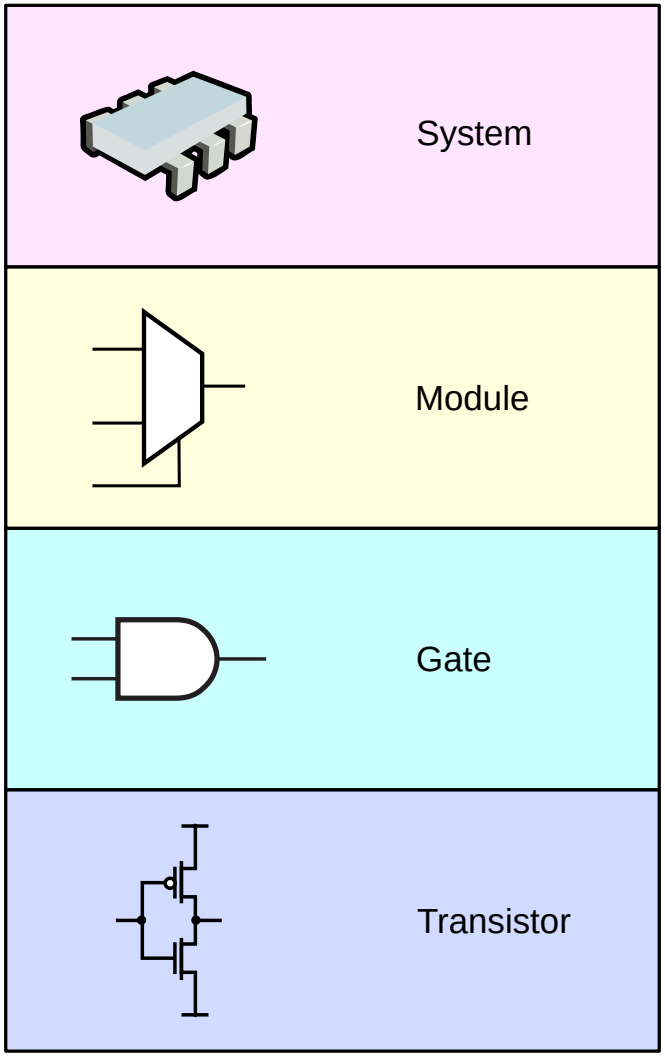


Source: IEEE international Solid-State Circuits Conference (ISSCC) 2013.

Level of Abstraction

Levels	Physical objects	Behavioral forms
System	Processors, controllers, memories, SoC	Executable programs
Register	Adder, comparators, registers, counters	Instructions, RTL, flowcharts
Gate	Gates, flip-flops	Boolean equations, finite state machines
Transistor	Transistors, resistors capacitors	Current-voltage equations

Level of Abstraction



The Three Y's

- **Hierarchy** involves dividing a system into modules, then further subdividing each of these module until the pieces are manageable.
- **Modularity** means that modules have well-defined functions and interfaces, so can connect together without surprises.
- **Regularity** seeks uniformity among modules. Common modules reused often, reducing number of distinct modules that must be designed.

Hardware Description languages

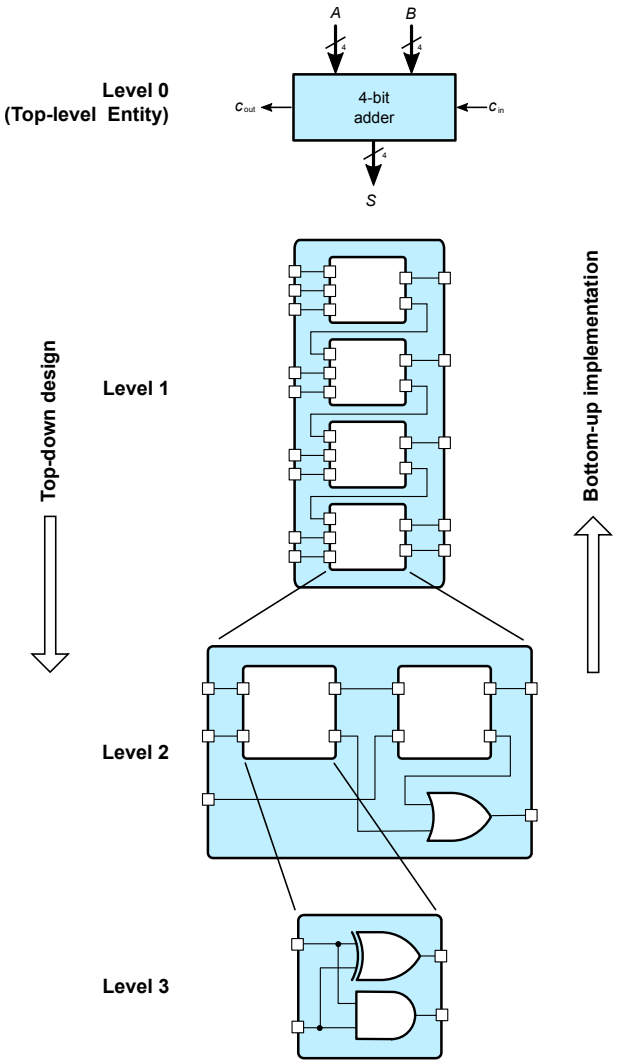
Why?

- Design exploration: describe & simulate a circuit
- Logic synthesis: compile code to create hardware
- More productive compared to schematic entry

Most common HDLs:

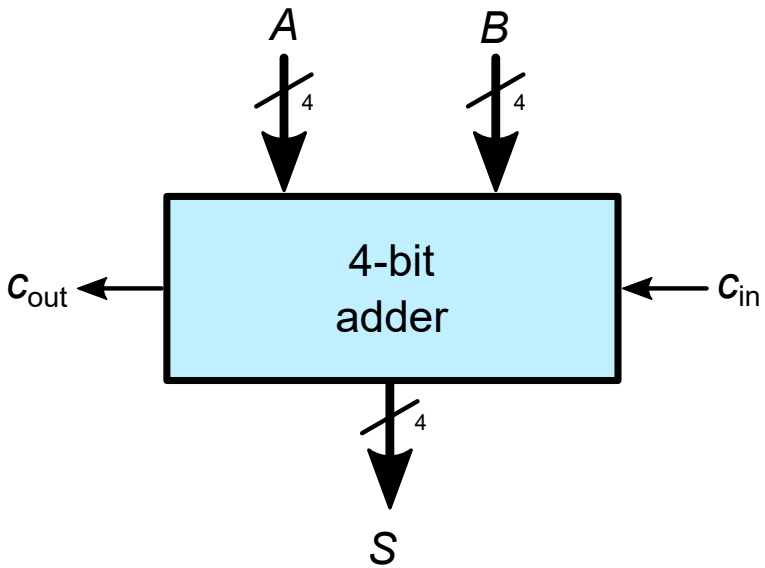
- VHDL
  - Based on Ada language
  - Somewhat more suitable for very complex design
- Verilog
  - Based on C language
  - Usually simpler code compared to VHDL

Top-Down Design: Level 0

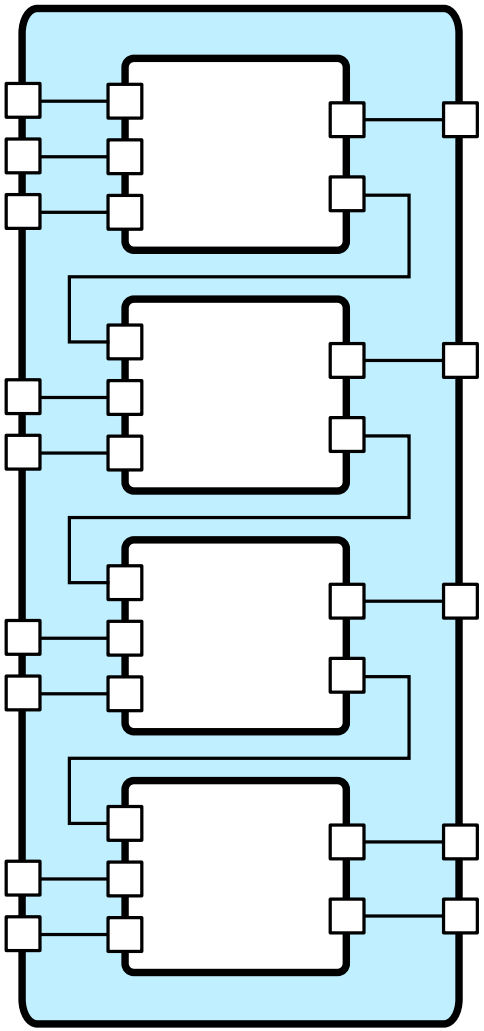


a.k.a. Entity Level

- Start at design from the top.
- Define the inputs, outputs and function



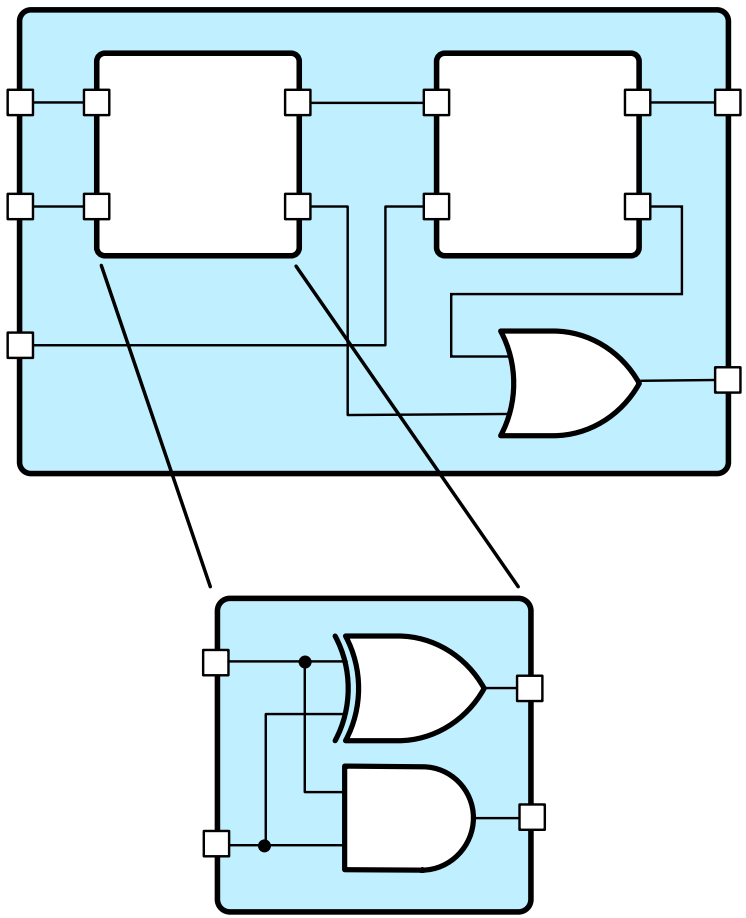
Top-Down Design: Level 1



Major blocks

- Next draw and connect the major blocks.
- Define inputs, output and function of each major block
- “Specify but don’t implement”

Top-Down Design: Level 2 & 3



Refine major blocks

- Define inputs, output and function of the blocks, going into further detail
- Top-down design is complete when a block is built of primitives (gates & flip-flops)

Bottom-Up Implementation

- Build the components from the primitive level first
- Test, test & test
- When no errors, store component in library module
- Then build the the higher level component using library modules
- Keep adding modules to library
- Stop when you reach Entity Level

ASIC

- Application-Specific Integrated Circuit
- A custom-made chip
- Take a lot of engineering effort to design but for some applications, it's worth the trouble
- Microprocessors & microcontrollers must be ASIC to achieve maximum performance & lowest power
- In Malaysia, ASIC design is done in Intel, Altera, Silterra
- Must be good in VLSI design

FPGA

- Field-Programmable Gate Array
- Programmable hardware
- A blank chip is programmed to implement logic functions
- Mainly used to accelerate operations which are too slow on microcontrollers yet want to maintain small size & low power
  - Remember, PC is also power but big & power-hungry
- FPGA design is easier than ASIC design
- Top two FPGA companies are Xilinx and Altera

FPGA is very useful in:

- Robotics & Automation
  - Control of robots using image processing
  - Low power high speed controllers
- Instrumentation
  - Increase data acquisition speed
  - Increase measurement accuracy
- Communications
  - Process more data packets per second
  - Implement software-defined radio
- Power
  - Control a motor more precisely
  - More efficient renewable energy

System-on-Chip (SoC) and Embedded Systems

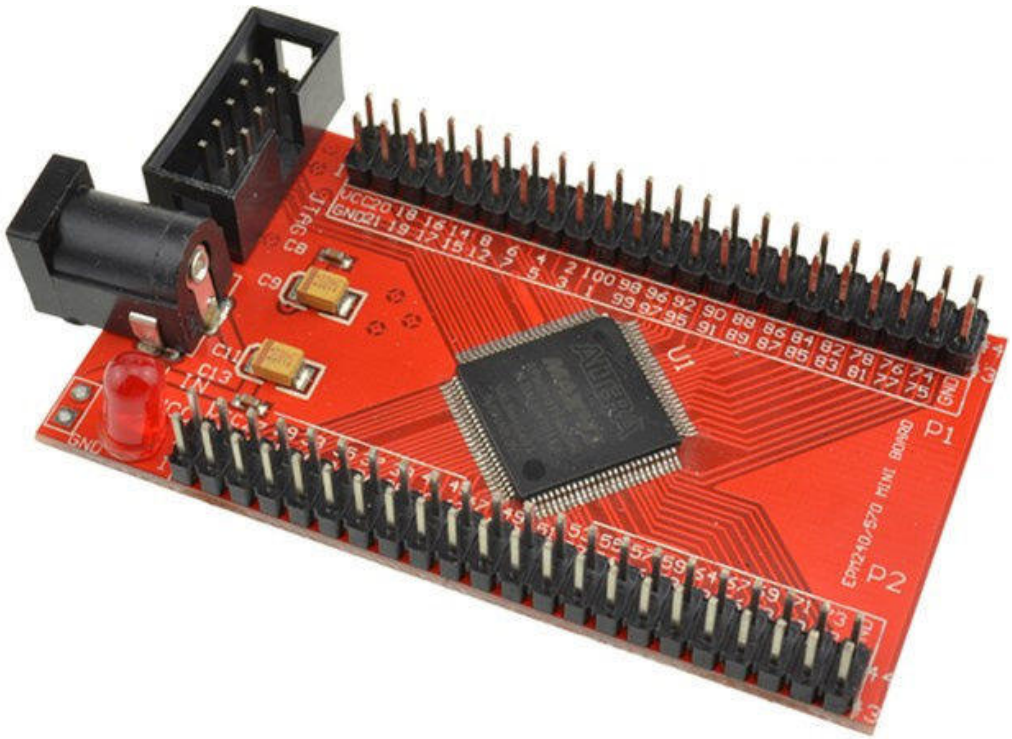
SoC:

- A complex chip that integrates major functions of a complete end-product into a single chip or chipset
- Examples: Apple A5 which contains ARM processor & most hardware required for iPad
- May be ASIC or FPGA

Embedded systems:

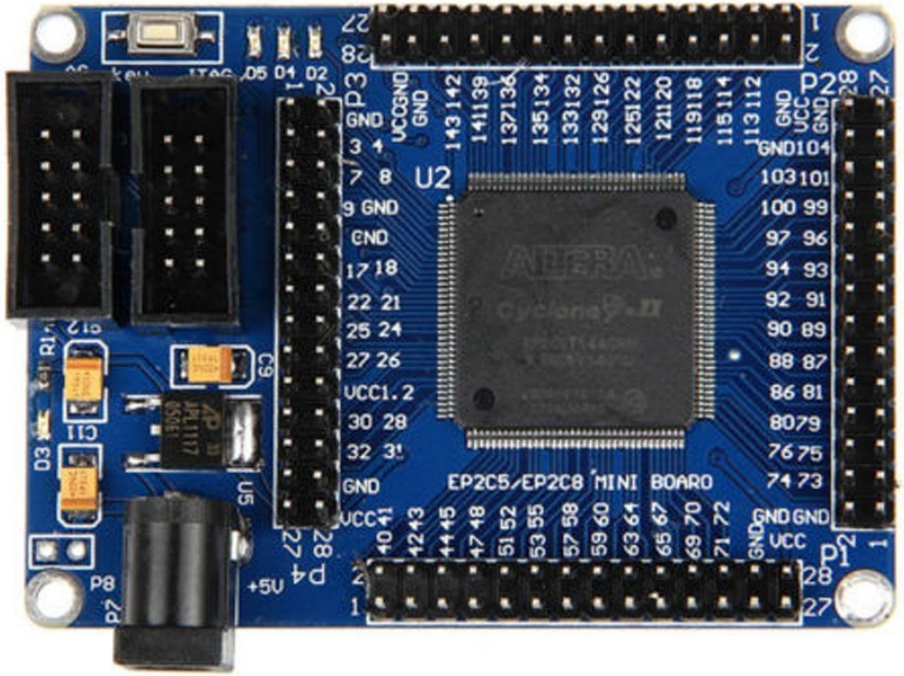
- A processor that runs at something that doesn't look like a computer
- Examples: computer mouse, USB drive, printer, camera, TV, traffic light.... the list is endless
- Nice place to apply what you will learn in this subject

Altera EPM240 CPLD



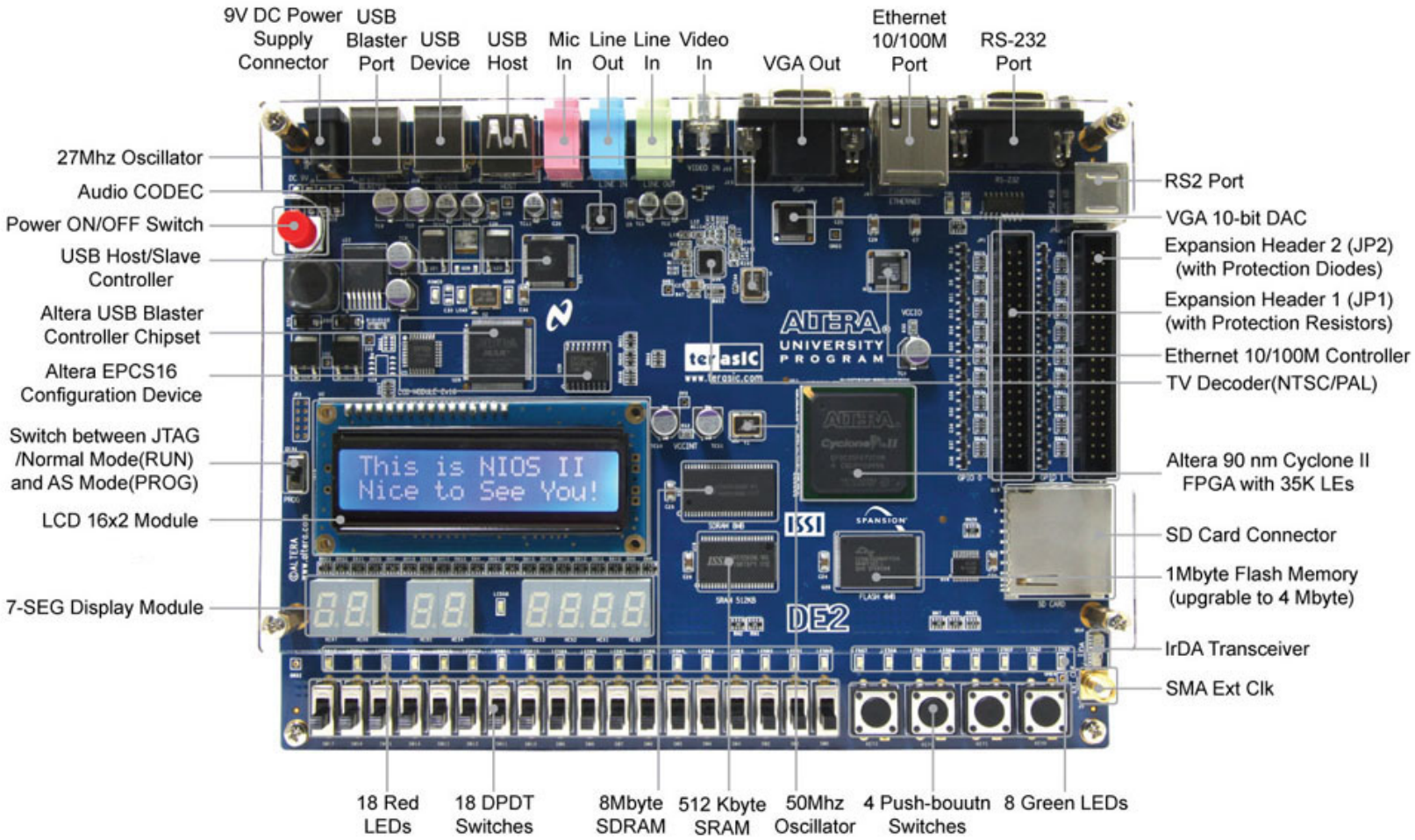
- Altera MAX II Family
- Lowest cost programmable device
- The one you'll use this sem
- Number of logic elements: 240
- Max freq: 300 MHz
- On board 50 MHz oscillator

Altera EP2C5T144 FPGA



- Altera Cyclone II Family
- Lowest cost FPGA
- Number of logic elements: 4068
- 13 multipliers
- Max freq: 300 MHz
- On board 50 MHz oscillator
- Supports on board Nios II CPU

Altera DE2 FPGA Board



Available in our lab

National Instruments myRIO

"myRIO provides educators with an embedded, WiFi-enabled solution to deliver an engaging approach to learning controls, investigating mechatronics, and designing imaginative capstone projects."

- Available in our lab
- With Xilinx Zynq-7010 SoC
- Zynq contains dual ARM Cortex-A9 processors + programmable logic (28K cells)
- Programmed using LabVIEW language

Summary

- Abstraction allows a design to be tackled at a high level by ignoring the details.
- Simple digital circuits may be entered using schematic capture but very complex systems require writing code in HDL.
- The concept of hierarchy, modularity and regularity maintains productivity.
- Contemporary digital systems are implemented on CPLD or FPGA for rapid prototyping. Designs are implemented on ASIC when cost and performance targets are justified.