CO503 - 2020

Advanced Embedded Systems

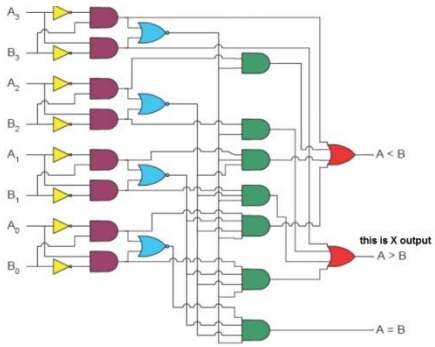
System-on-Chip Design & Prototyping using FPGA Tools

ISURU NAWINNE

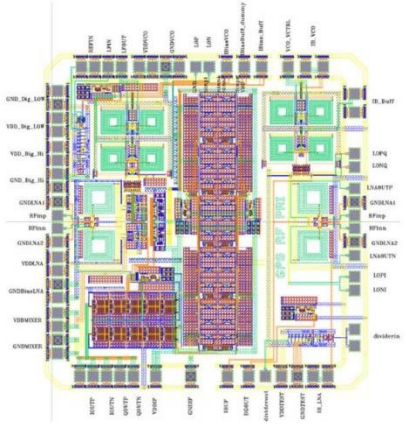
FIELD-PROGRAMMABLE GATE ARRAYS

What is an FPGA?

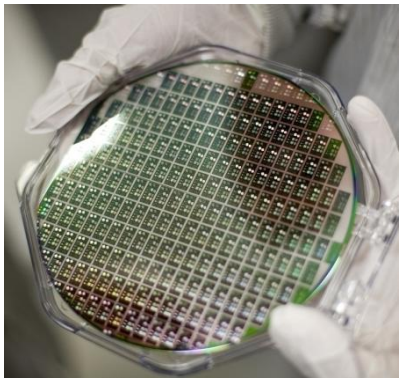
- Working with traditional Integrated Circuits (ICs)



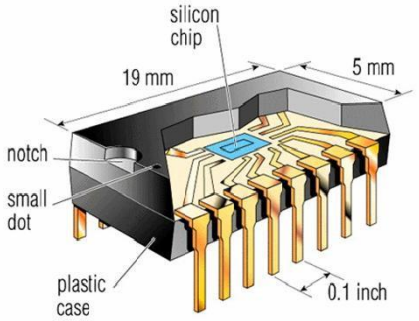
Design



Synthesize



Fabricate



Package

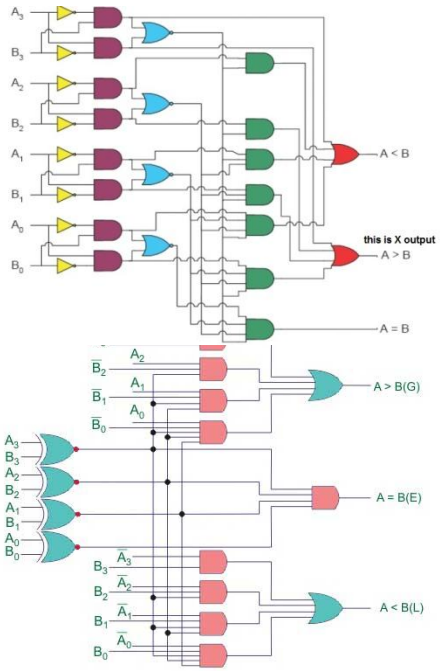
IC

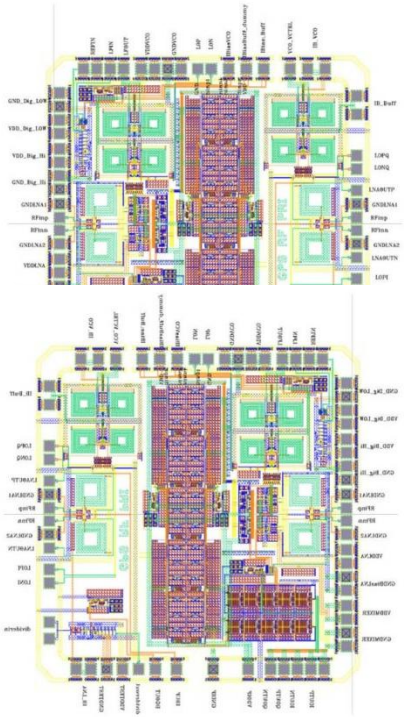
Single Function!

FIELD-PROGRAMMABLE GATE ARRAYS

What is an FPGA?

- Working with FPGAs





Program

Program

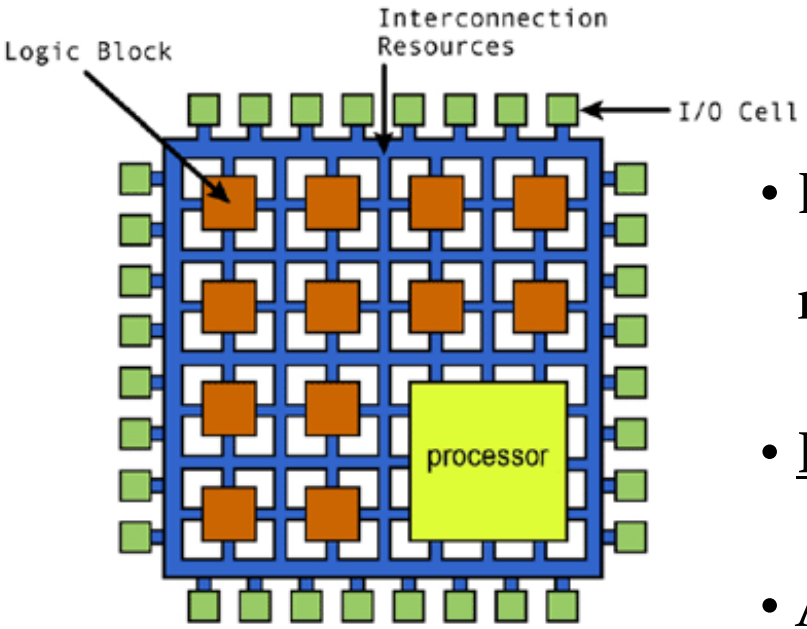
Program

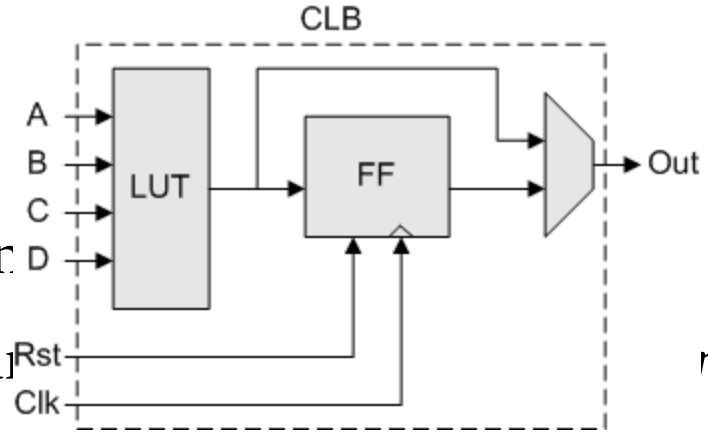
Buy FPGA
Flexible!

Allows upgradability
Great for **prototyping**

FIELD-PROGRAMMABLE GATE ARRAYS

What is an FPGA?





- Fur
- ma *mable"*
- **Reconfigurable** logic fabric
  - Configure the logic blocks
  - Route the logic blocks
- Allows designers to implement various digital logic circuits on the fabric

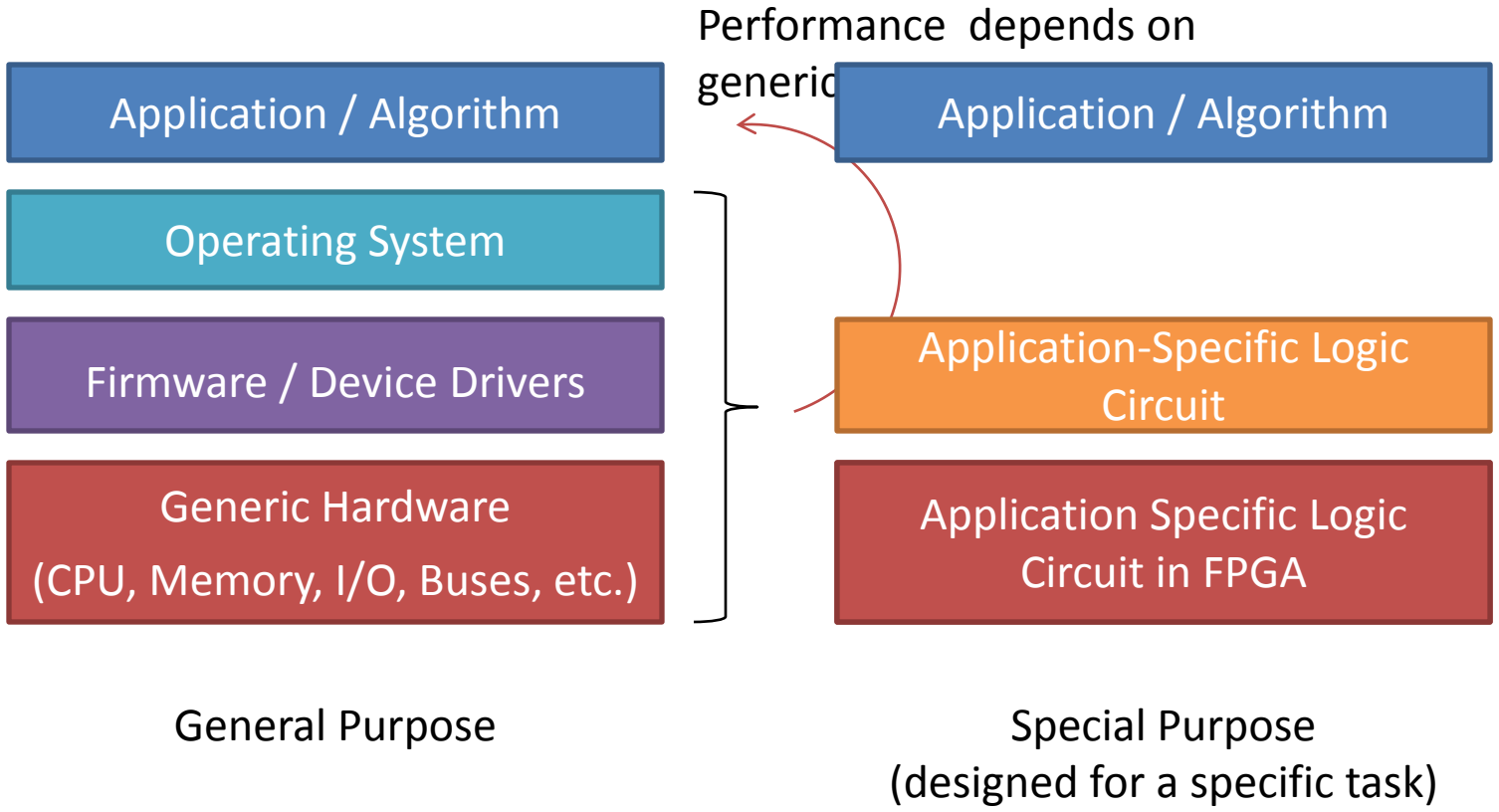
(just like we can run different programs on a CPU)

Logic blocks :

- Simple (gates)
- Complex (cells, LUTs, etc.)

How can an FPGA improve performance?

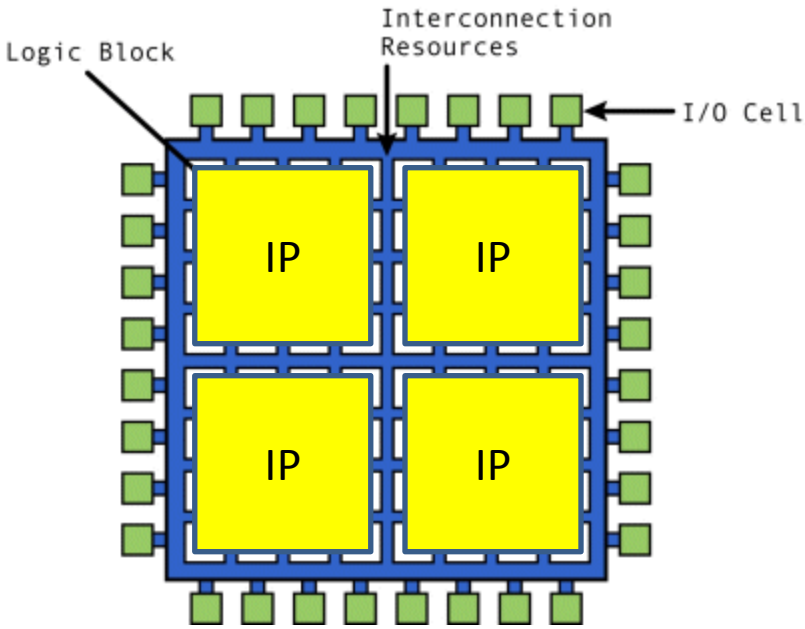
1) Through *application-specific* hardware design:



FIELD-PROGRAMMABLE GATE ARRAYS

How can an FPGA improve performance?

2) By exploiting *parallelism*:

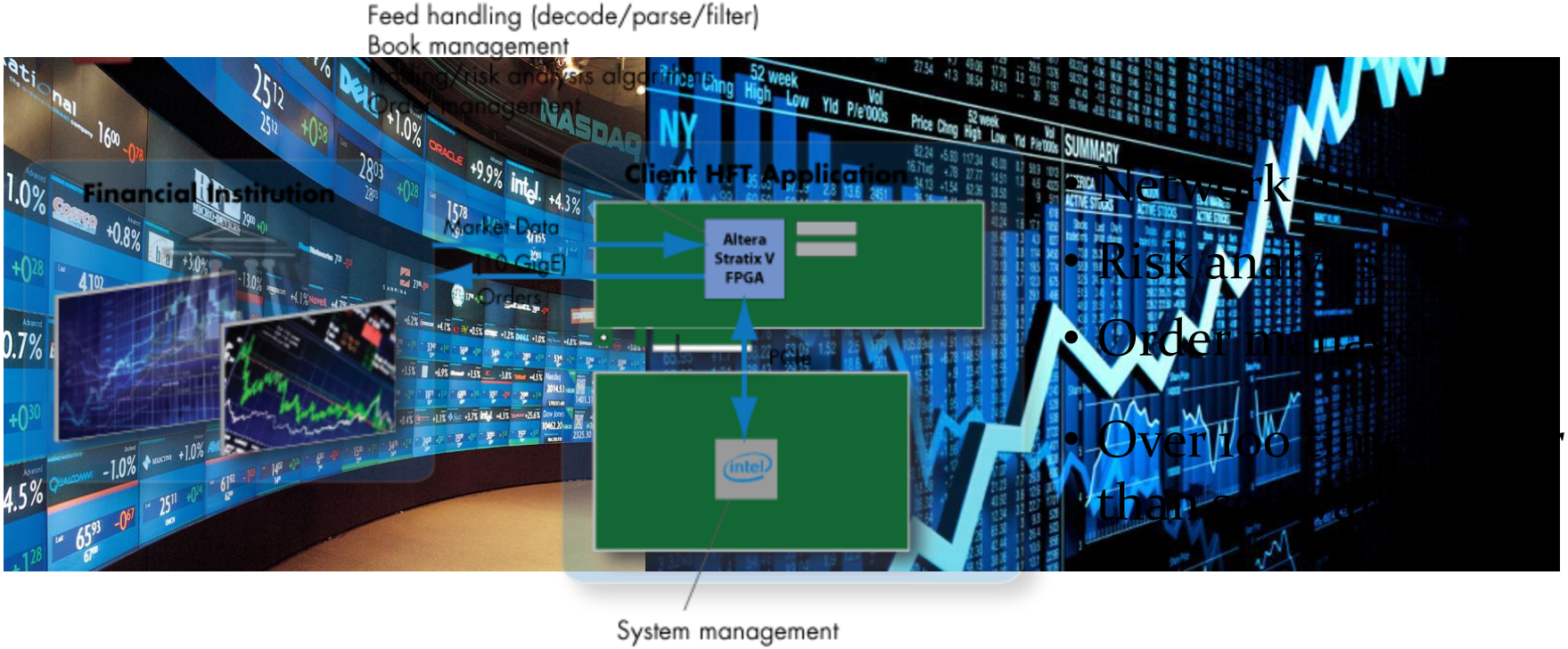


- Multiple circuits working in parallel
- SIMD approach, similar to GPUs (identical circuits, different data)

OR

- Different circuits (heterogeneous)
- Depends on accelerated task/application
- Multiple FPGAs connected together can host massive designs!

Example: High Frequency Trading (HFT)



- Stocks/shares
- Forex (currency)

} Property held for less than half a second

Buy/sell decision made in sub-millisecond scale

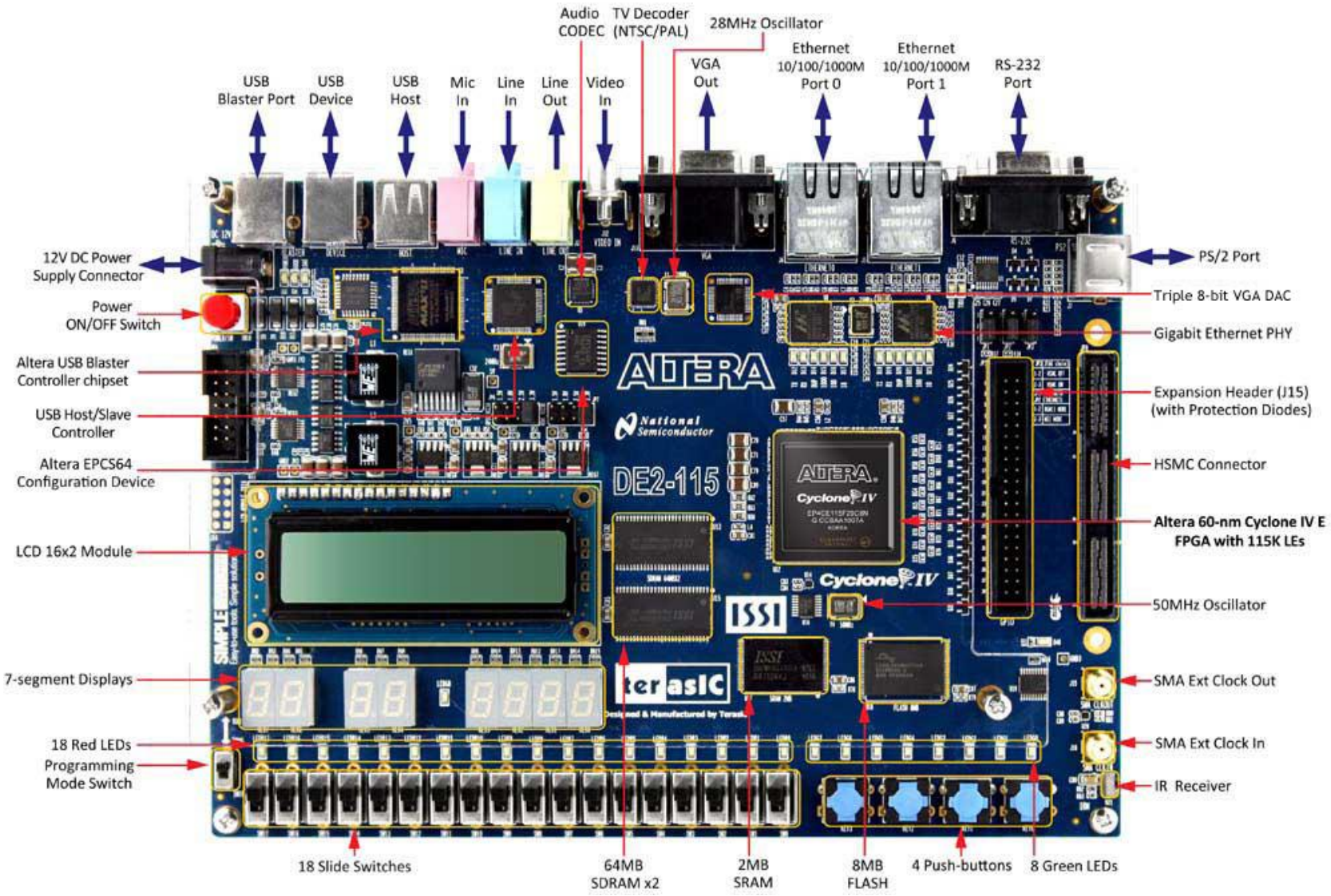
FPGA ACCELERATION

Other Example Applications…

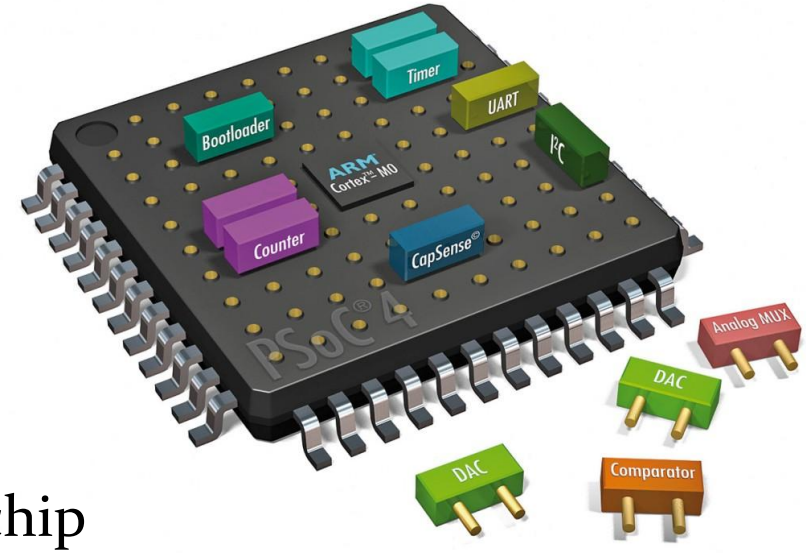
- Accelerating Search Engines: Microsoft's *Bing* (page ranking algo)
- Accelerating Cloud Services: Microsoft's *Azure* (service requests)
- Hybrid Computing: CPU+FPGA chips (targeting datacenters and cloud servers)
    - *SRC Computers' MAP* hybrid processor
    - *Intel*'s acquisition of *Altera* (for $16.7bn !)

FPGA's are not new (been around since 1980's), but they're making a come back as CPU performance reaches limits…

FIELD-PROGRAMMABLE GATE ARRAYS



What is a SoC ?

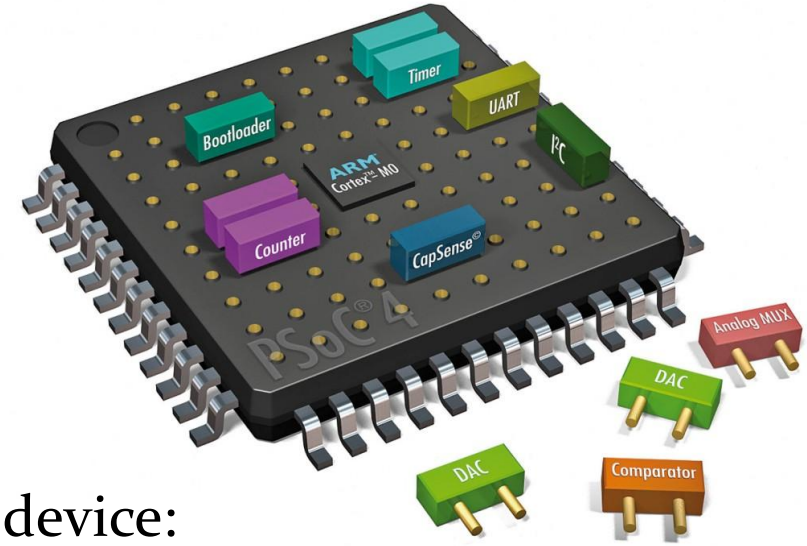


A system-on-a-chip (SoC) is a microchip

with all the necessary electronic circuits and parts for a given system

- CPU
- Memory (cache, DRAM controller, etc.)
- GPU
- I/O (UART, I²C, CAN, DAC, etc.)

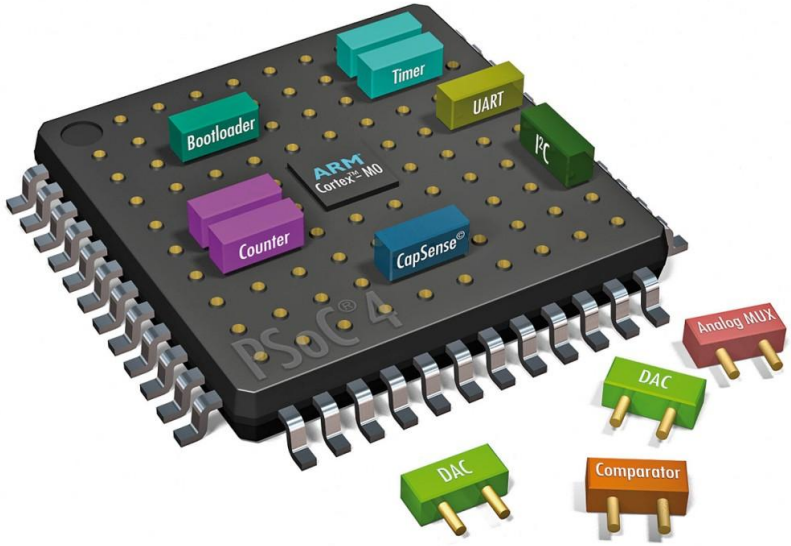
What is a SoC ?



Example - SoC for a sound-detecting device:

- ADC
- CPU / DSP
- Memory
- General Purpose I/O

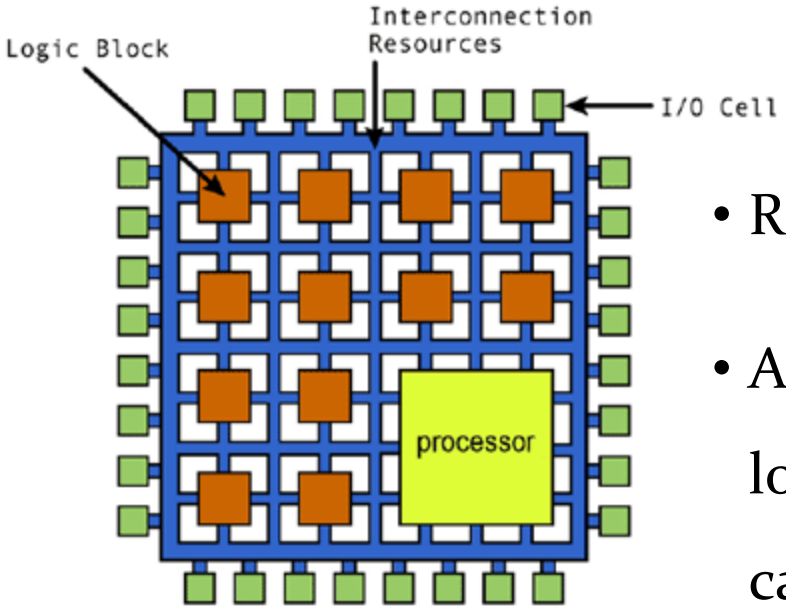
What is a SoC ?



Benefits of SoCs :

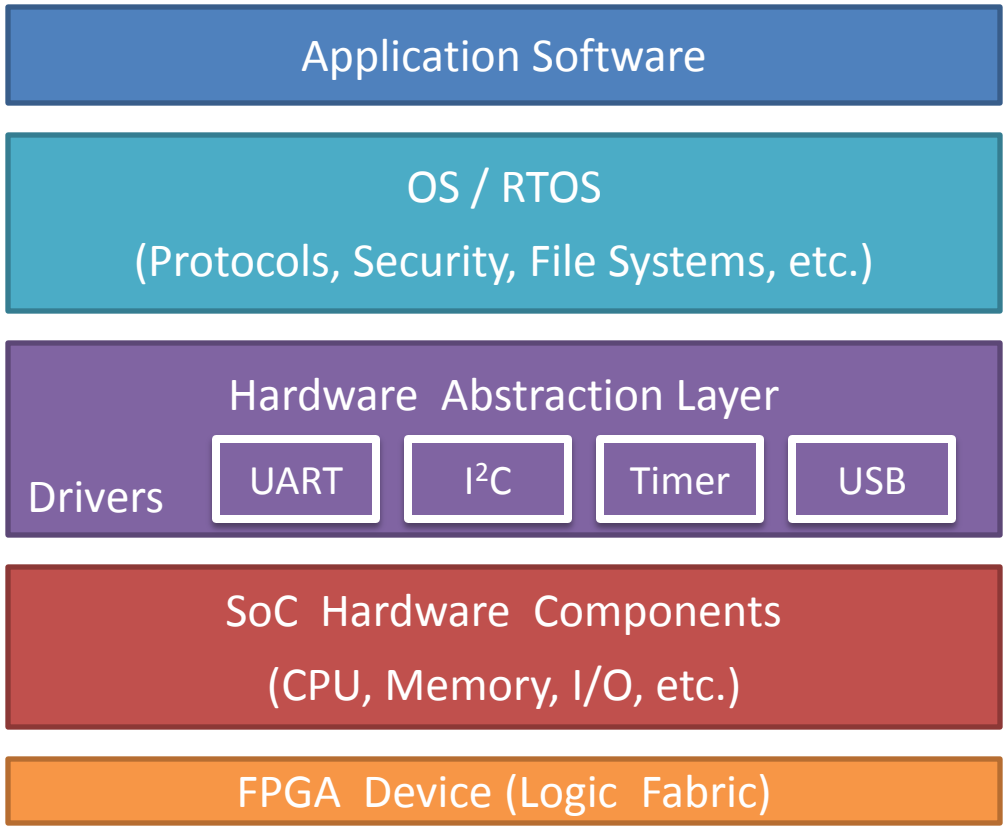
- Increase performance (low communication delays)
- Low power consumption
- Smaller size
- Reduced overall cost

SoC on FPGA (Field Programmable Gate Arrays)

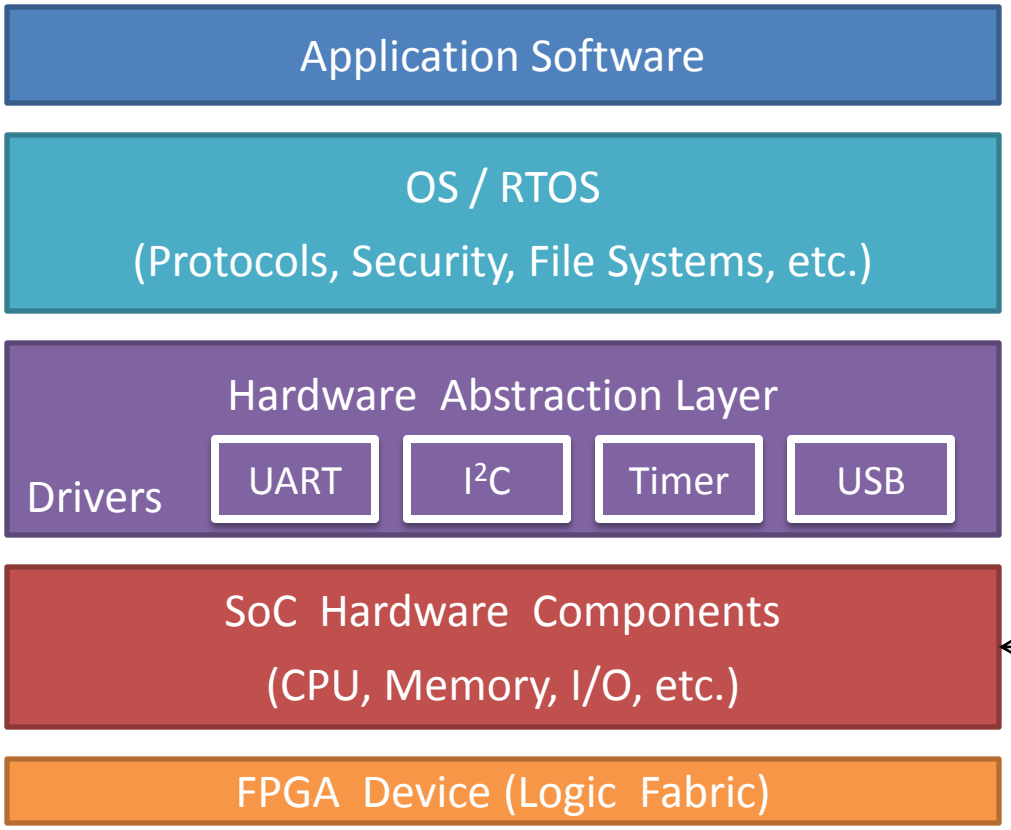


- Reconfigurable logic fabric
- Allows designers to implement digital logic circuits on the fabric (just like we can run different programs on a CPU)
- SoC components can be implemented on the FPGA

How does a complete System look like?



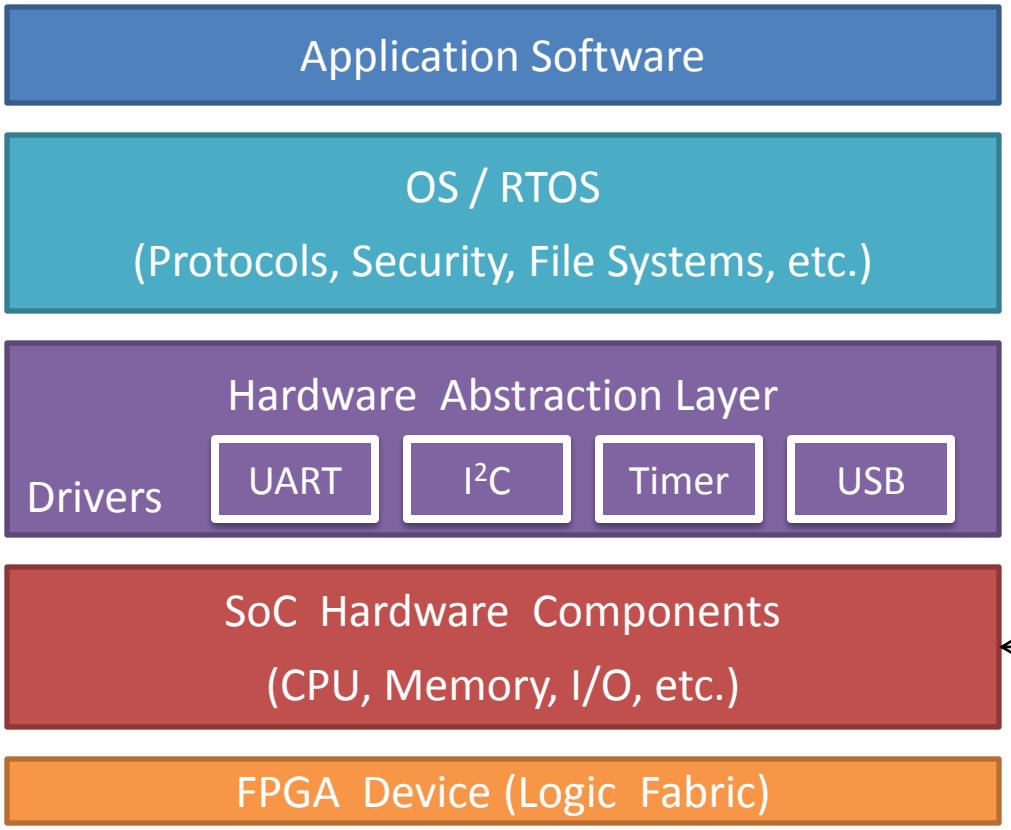
Design Steps



1. Designing the SoC with components

- HDL – VHDL, Verilog
- Block Diagram / Schematic
- State machine

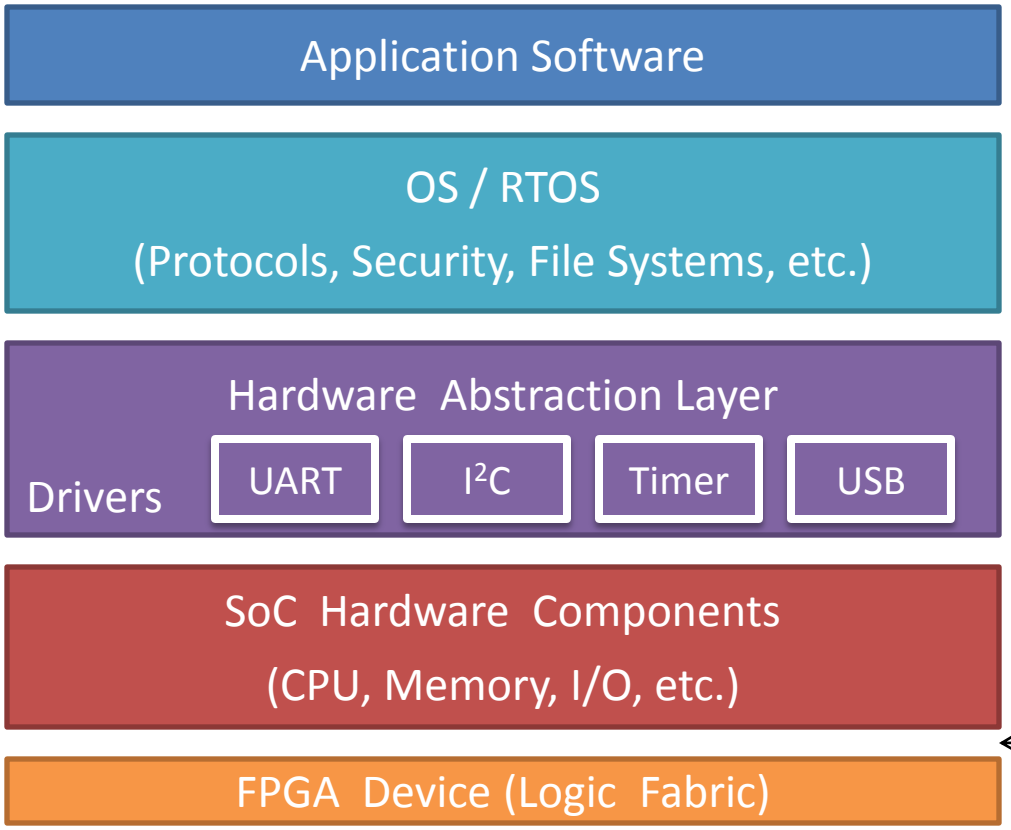
Design Steps



2. Analysis & Elaboration

- Check for Syntax Errors
- Check for Semantic Errors

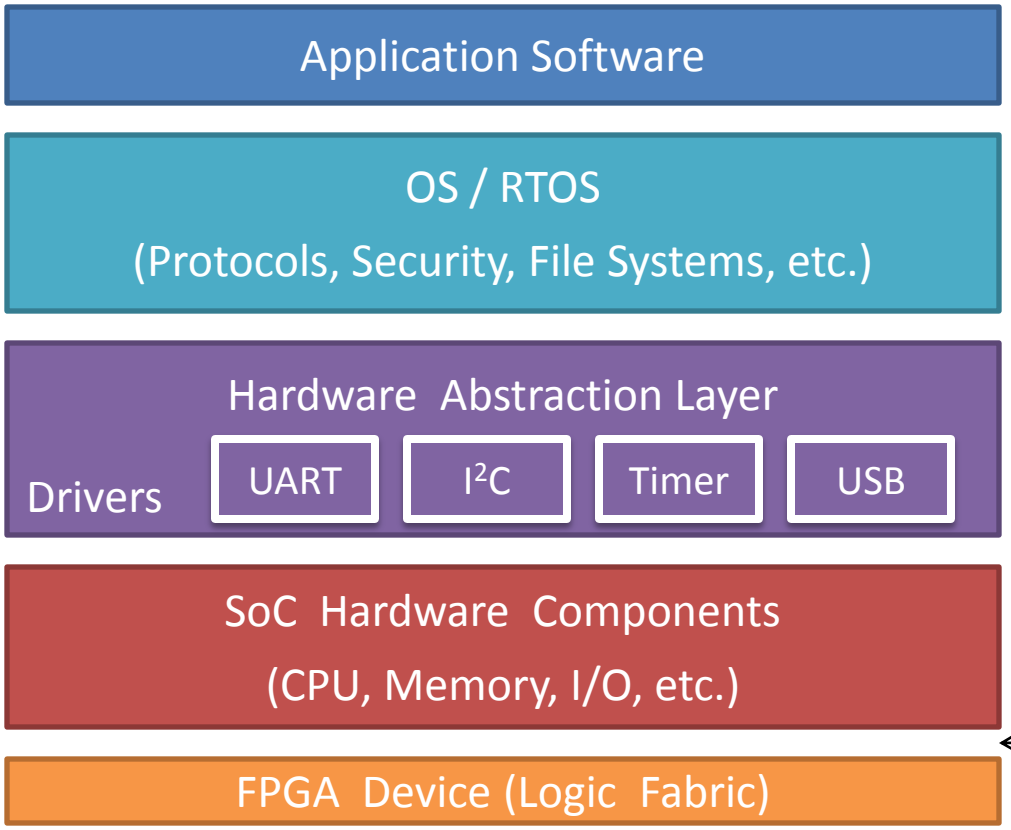
Design Steps



3. Pin Planning

- Map I/O wires of the SoC to actual pins on the FPGA device

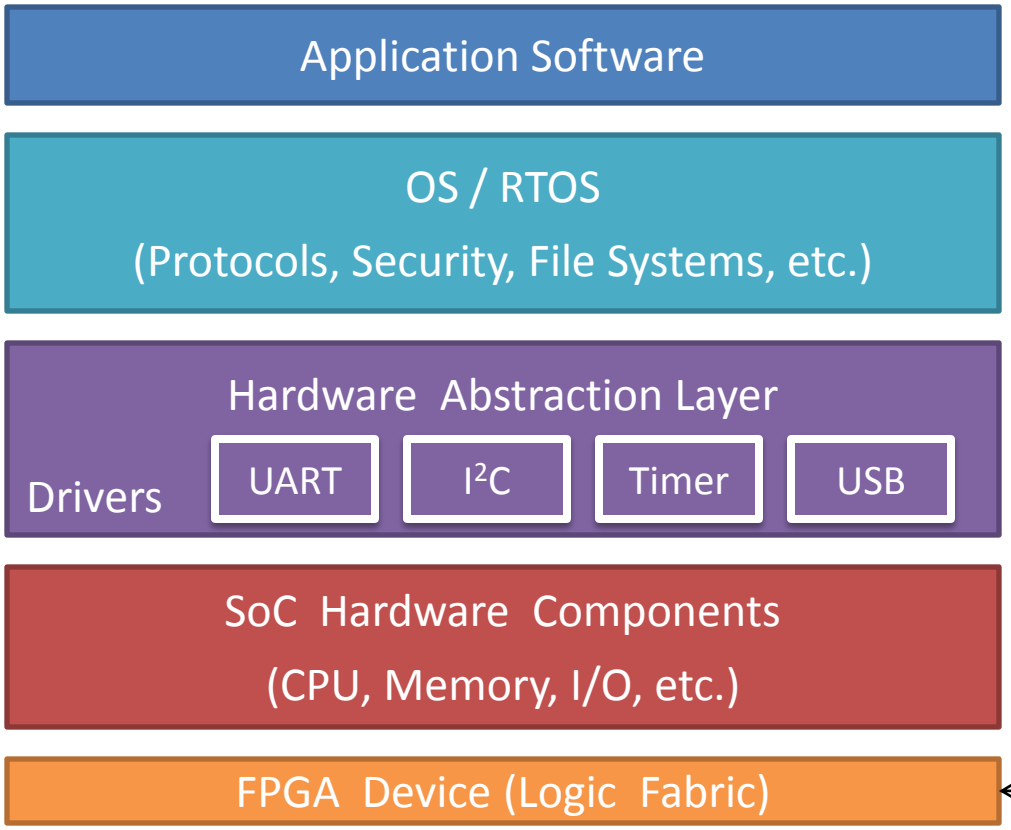
Design Steps



4. Compilation

- Synthesis
  - Technology mapping
  - Optimize logic usage
- Place & Route
- Generate Assembler (for software)
- Timing Analysis

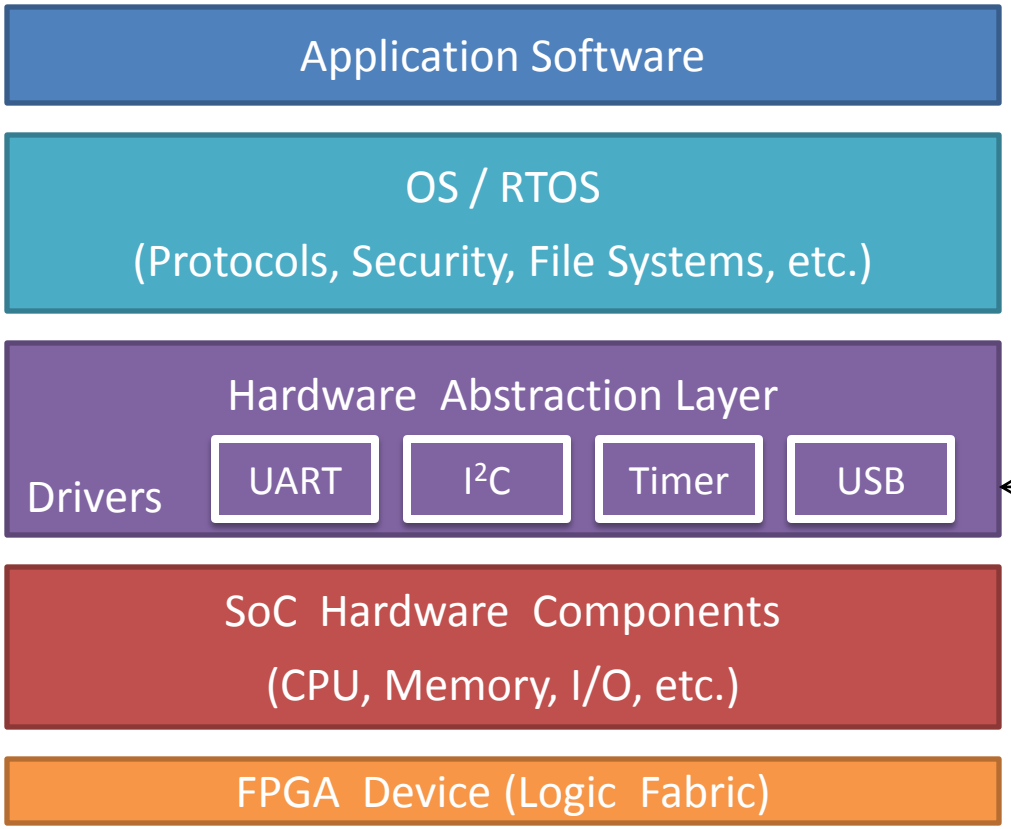
Design Steps



5. Program FPGA device

- Download the compiled design
- Configure the logic fabric

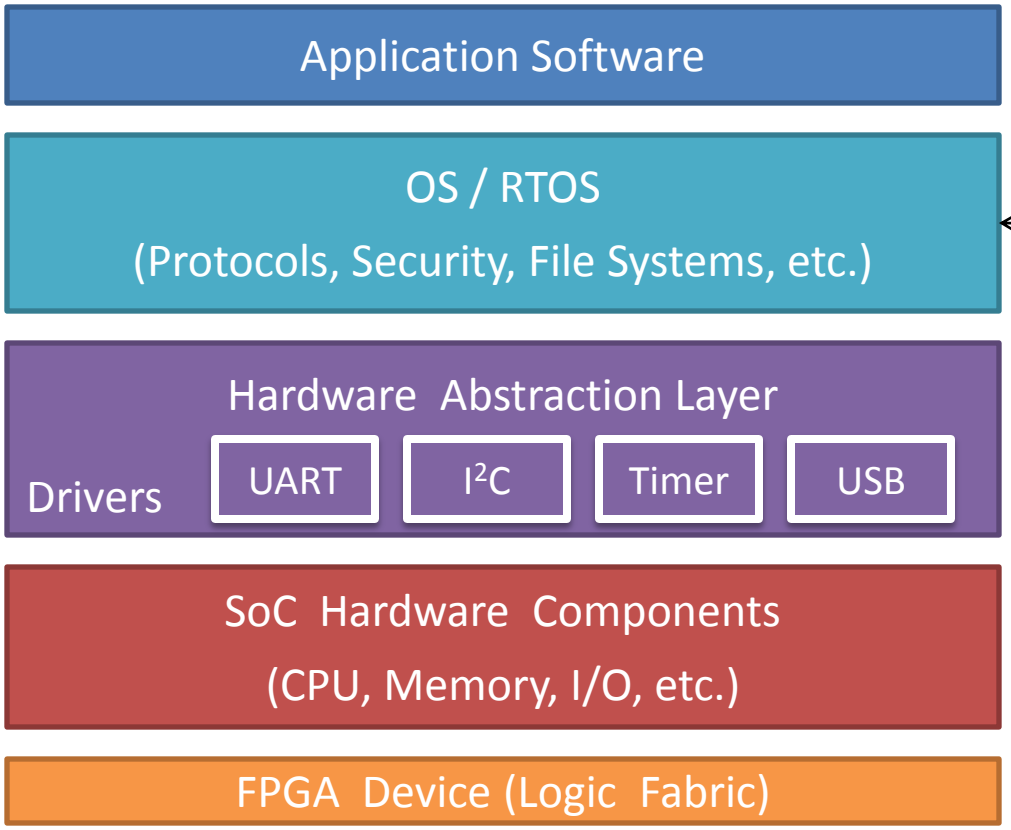
Design Steps



6. Generate HAL

- Low level code to support software
- Include required device drivers
- Define memory layout
- Generate compiler support for application software

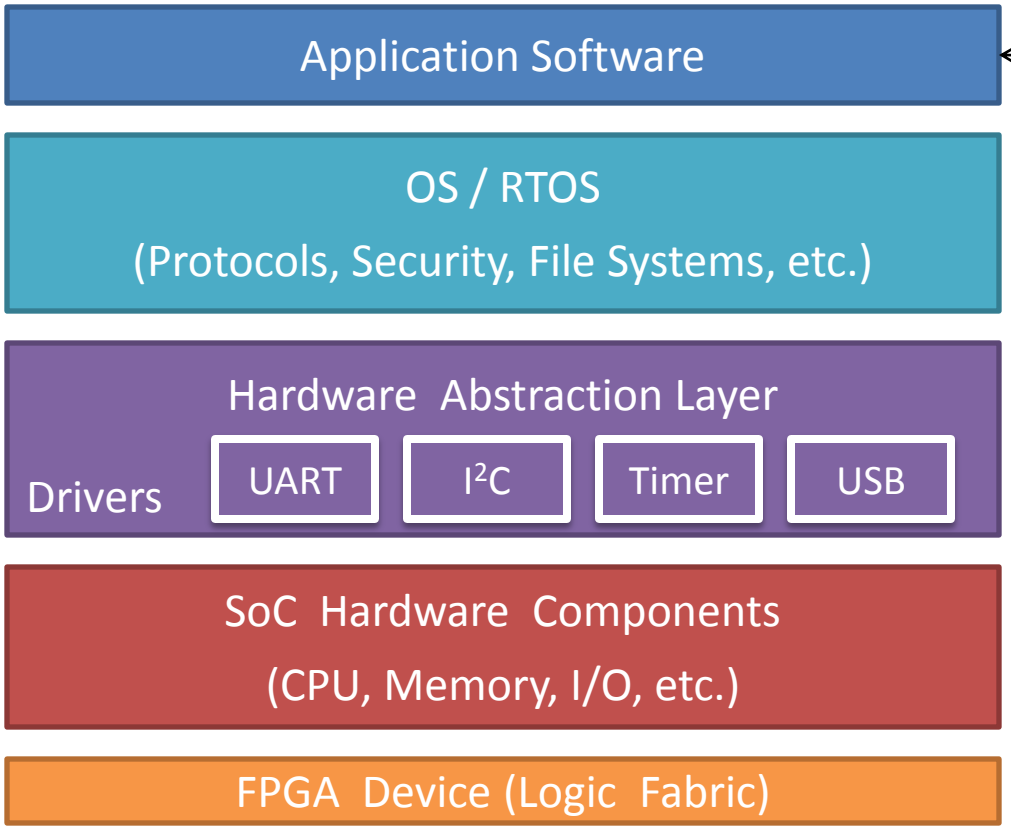
Design Steps



7. Operating System (optional)

- ex: microc/os-ii, Linux
- depending on requirement
- provides advanced resource management and security.

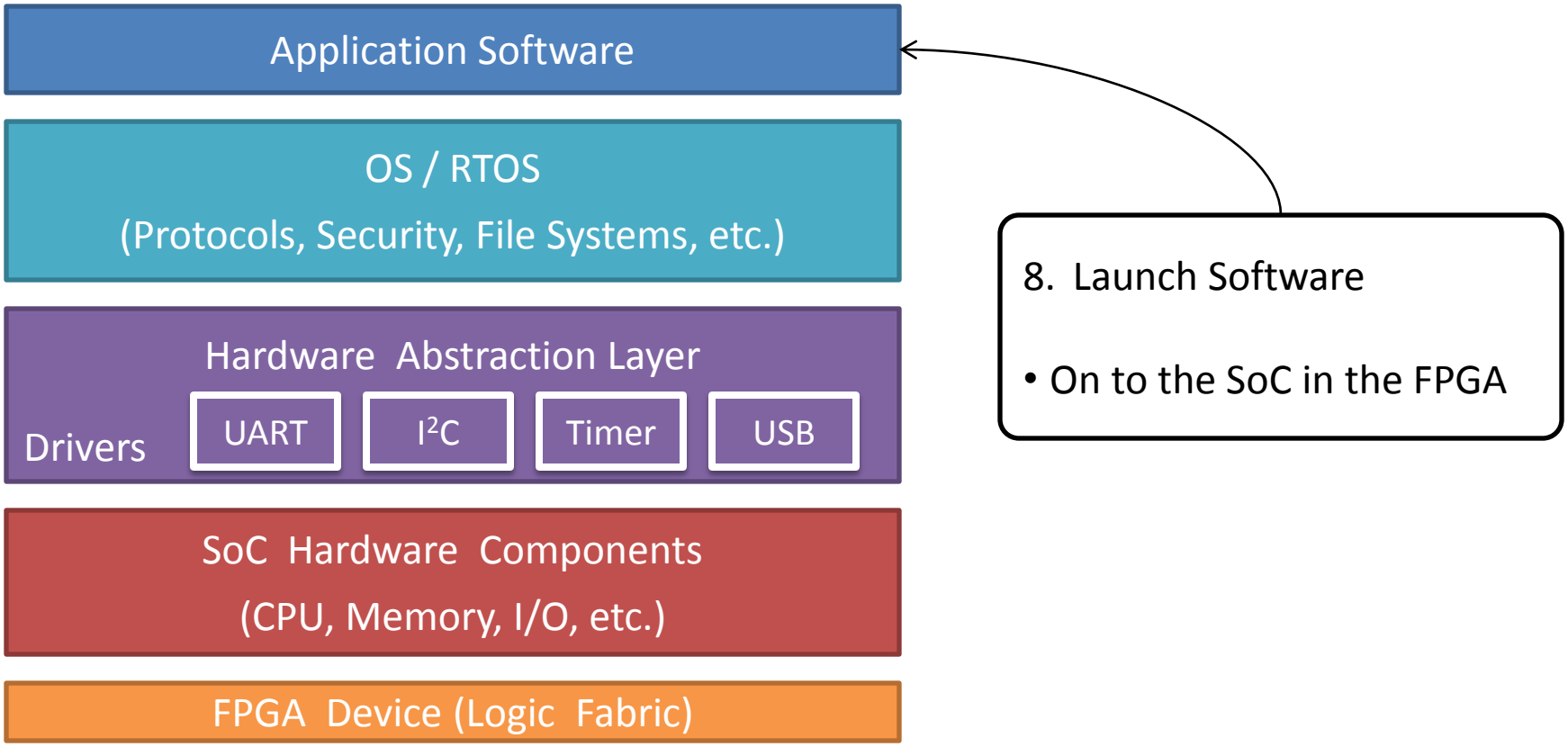
Design Steps



8. Build Software Programs

- Using the functionality of designed hardware, through the firmware in HAL

Design Steps



In the lab...

Build and test a SoC on FPGA.

Learning Objectives:

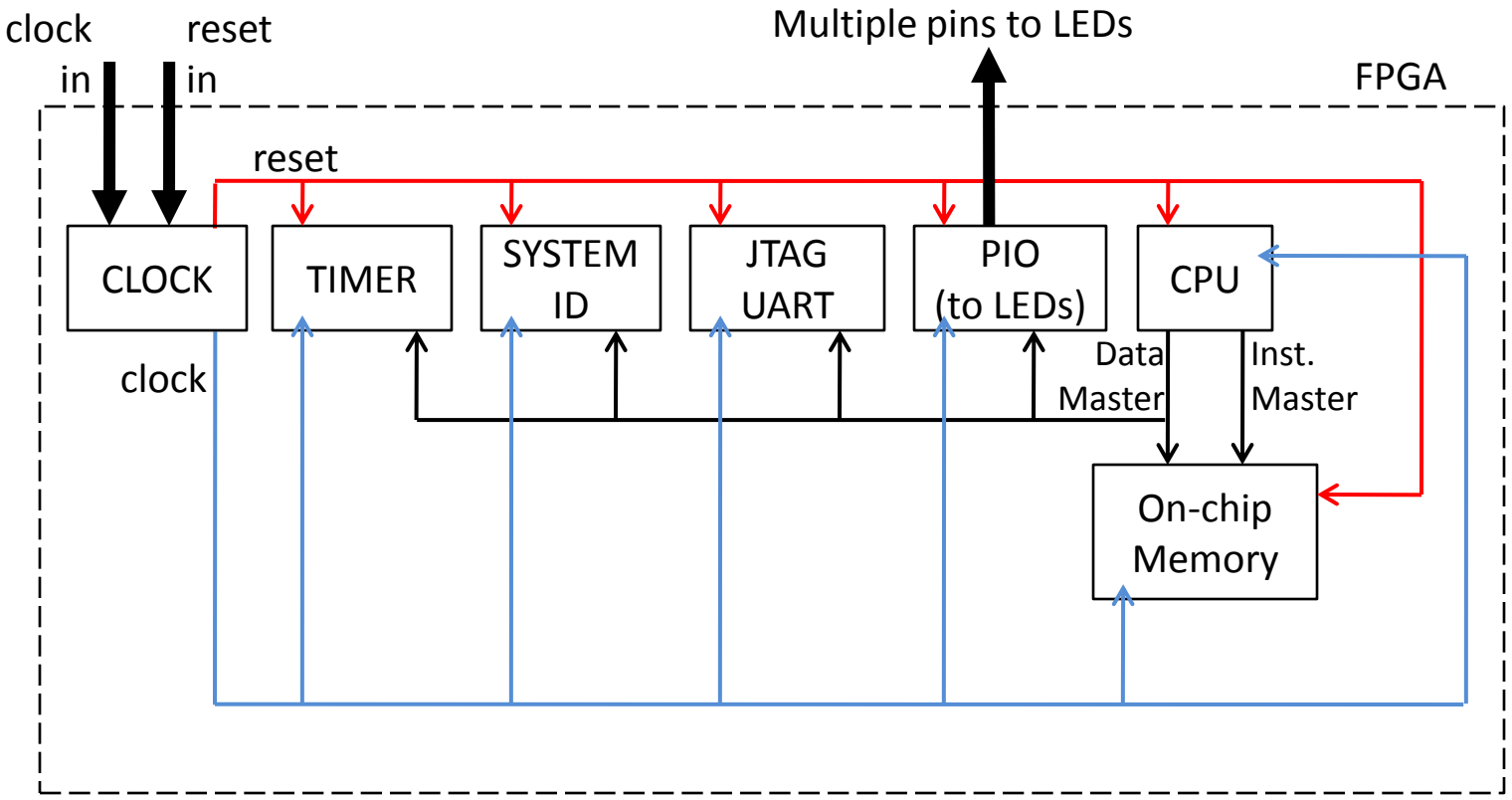
- Design and synthesize System-on-Chip hardware.
- Co-design and develop hardware and software .
- Use FPGA-based design and prototyping tools.

Tools:

- **Altera Quartus II 12.1 / 13.1** design suite (install)
- **Altera DE2-115** FPGA board (get one from our TO)

In the lab...

Part 1 : Getting started – Simple LED counter. (Step-by-step guide)



In the lab...

Part 2 : JPEG Encoder SoC