TEC de Monterrey®

DEL SISTEMA TECNOLÓGICO DE MONTERREY

An Overview of Embedded Systems

Luis F. González P.

ITESM GDA

What´s an Embedded System?

- An embedded system is a computer that doesn´t look like a computer ☺
- Wikipedia says an embedded system is "a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints"
- A more formal definition…
  - An embedded system is a collection of programmable parts surrounded by application specific integrated circuits (ASICs) and other standard components that interact with the environment through sensors and actuators to perform a specific task
  - The collection can be physically a set of chips on a board (off the shelf) or a set of modules on an IC (SoC – System on Chip)

Typical Characteristics

- Embedded systems differ from traditional digital systems in which most of the system functionality is implemented in hardware
- In an embedded system, many of these functionalities (tasks) are implemented in software running on one or more processors.
- Dedicated hardware is only used for increased performance and/or reduced power consumption (generally DSP)

Typical Characteristics

- They perform in reactive and time-constrained environments
  - Reactive: responds to events in the environment
  - Time-constrained: must respond within a specific time interval, i.e., ES have deadlines (hard or soft)
- They might contain one or more processors, memories and additional logic
- They are similar in structure to a general-purpose computer. Where is the key difference?
  - The **software** (functionality) in the embedded system is part of the **system specification**
- **Embedded software** is a key part of the system design and is, in general, unchanged after shipped to the end user. That is why embedded software is AKA **firmware**

Embedded vs Hardware Systems

- Migration to embedded systems has been driven by the demands of **increasingly complicated system features**, lower system cost and shorter product development cycles
- This can be met with **software programmable solutions** (application specific code running on existing processors) rather than application specific logic
- Software is **easier to develop and more flexible** than hardware
- By using different versions of the software, a family of products based on the same hardware can be developed to target different market segments
- This spreads hardware design cost and reduces design time
- Software allows to **enhance the system features quickly** so as to suit the end users' changing requirements and differentiate the product from that of the competitors

Embedded Systems Requirements

- Embedded systems are usually **not very flexible** and always perform the same task
- Embedded systems need to be designed in an extremely **short** time to meet their **time-to-market**.
  - Only a few months should elapse from the conception of a consumer product to the first working prototype
  - If deadlines are not met, the result is a concurrent increase in design cost and decrease of the profits, because fewer items will be sold
  - Delays in the design cycle may make a huge difference between a succesfull product and an unsuccesful one
- **Cost, reliability and safety are often more important** criteria than performance

Embedded Systems Requirements

- Embedded systems generally interact with the outside world
  - They measure sensors
  - Control actuators
  - Communicate with other systems
  - Interact with users
- Many of these tasks must be performed at precise times
- An embedded system with such timing constraints is called a **real-time system**
- For real-time systems, the correctness depends not only on the logical results of computations (its functionality), but also on the time at which the results are produced

Embedded Systems Requirements

- Real-time systems are further classified as
  - Hard real-time systems
  - Soft real-time systems
- **Hard real-time systems** cannot tolerate any missed timing deadlines
  - Example: automotive apps
- **Soft real-time systems** have less stringent timing requirements. An occasionally missing of timing deadline can be tolerated
  - Example: cellular phone

Embedded Systems Requirements

- Real-time systems **MUST NOT NECESSARILY** be a fast response system
- A real- time system is one with deterministic timing behavior, i.e., the completion of a task is guaranteed to be within a specified time interval
- The specification of this time interval requires determining tight bounds on the performance of the embedded software
- This is not an easy problem…

Challenges in Designing Embedded Systems

- There exist many challenges…
- Embedded systems have a set of tasks to execute. These tasks can be implemented in hardware (ASICs or FPGAs), or in software running on one or more processors
- Embedded system design includes HW and SW. It involves:
  - Processor selection
  - Partition of tasks between HW and SW
  - Synthesis of logic components
  - Program code
  - Performance and cost evaluation
  - Fulfill all real-time constraints

Challenges in Designing Embedded Systems

- Other important parameters:
  - Cost
  - Reliability
  - Power consumption
  - Size
  - Time to market
- Another, very important, aspect in the design of ES is the scheduling of software tasks on the available processors
- An important prerequisite of HW/SW codesign methods and scheduling algorithms is that the extreme case (best and worst cases) execution time of a single software task when running on a given processor be known

Current Applications

- Embedded systems are everywhere, spanning practically all aspects of modern life
  - Consumer electronics
  - Communication systems
  - Biomedical devices
  - Peripheral controllers
  - Industrial instrumentation
  - Automotive
  - Scientific

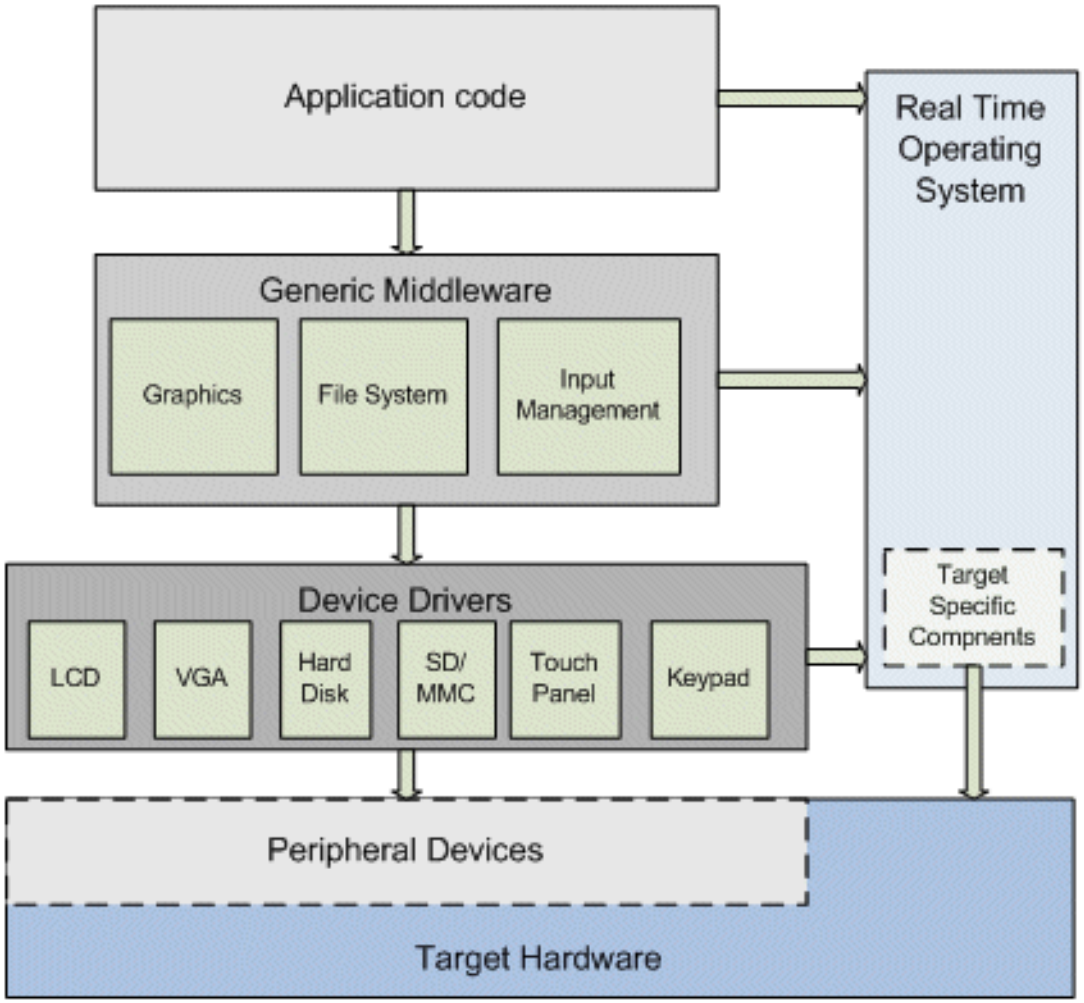
Embedded Systems Components

- Every embedded system consists of **custom hardware** built **around a Central Processing Unit** (CPU)
- The hardware contains memory circuitry onto which the software is loaded
- Depending on the overall system complexity and functionality, **software can be** written **specific** for the application **or** it would be more efficient to have **an operating system (OS) running above the hardware**
- OS serves as a manager of hardware resources and provides services for software applications scheduling tasks for efficient use of the embedded system
- OS also acts as arbiter between programs and hardware

Embedded Systems Components

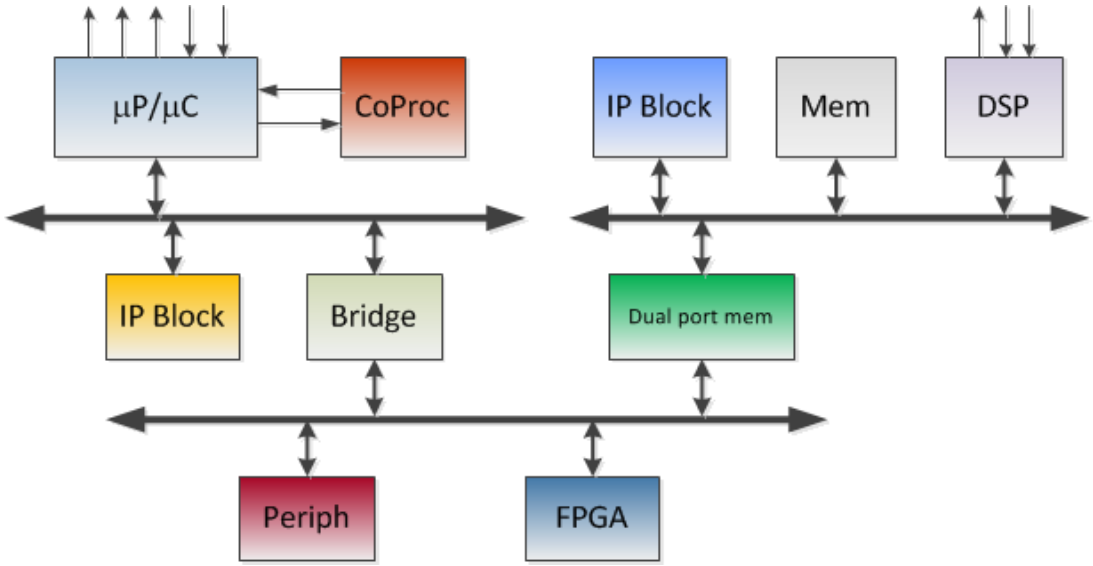
- In case an OS is employed, the final component is the application software
- For small appliances there is no need for an OS
- In addition to the OS and the application software, there is a vital component in an embedded system… device drivers
- Device drivers are low-level software that controls a particular type of device that is attached to the system.
  - It is specific to the device (modem, USB port, LCD screen, ADCs, DACs, codecs, etc.)
  - It simplifies system development by acting as a **translator** between the device and the application code or OS

Embedded Systems Components



Hardware Building Blocks in Embedded Systems

- All embedded systems have at least the following building blocks
  - Processor
  - Memory (ROM, RAM)
  - Input devices
  - Output devices
  - Communication interfaces
  - Application specific circuitry



Eembedded systems are very limited in resources, specially memory.
They are power constrained (battery powered) and need to be highly reliable.
And in addition… they must operate in extreme environmental conditions

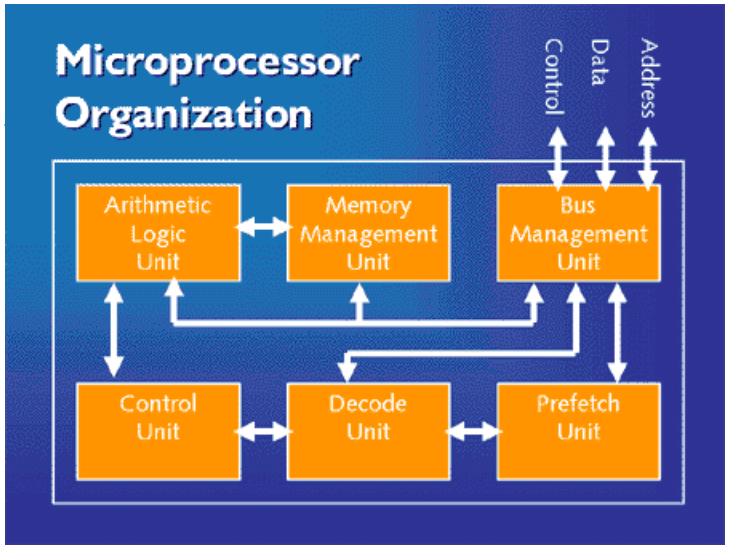
Processor

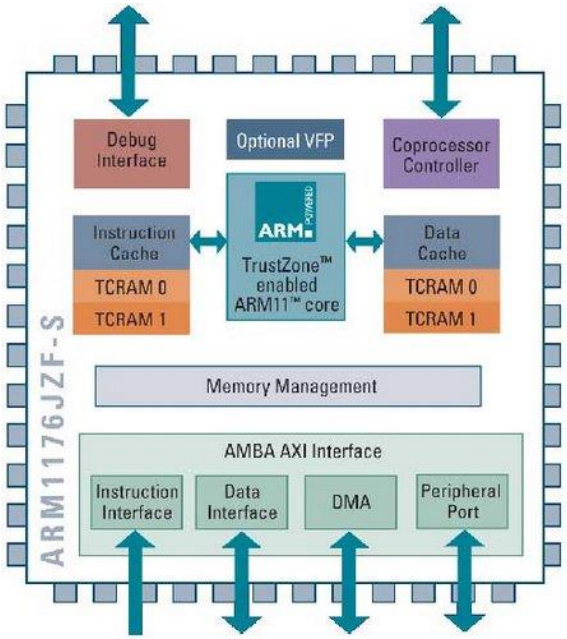
- Three main types of processors are used in embedded systems
  - Microcontrollers
  - Microprocessor
  - Digital Signal Processors

- They are specified by clock speed, data word length (8-bit, 16-bit, 32-bit)
- The higher the clock speed, the faster the processor
- Bigger data word-length leads to higher precision and higher memory addressing

Microprocessor

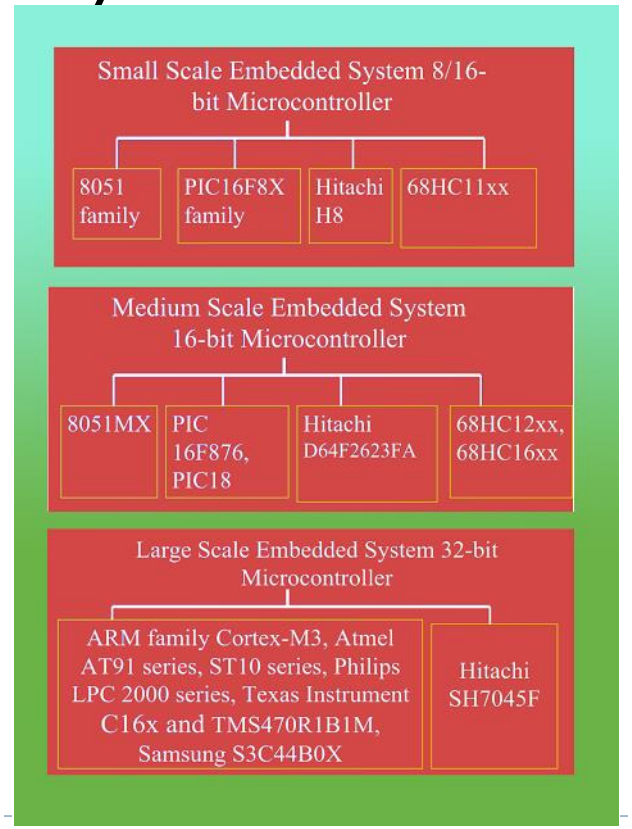
- µP is an IC which has only the Central Processing Unit inside them
- They don't have RAM, ROM and other peripheral chips
- The system designer has to add them externally
- They find application in areas where tasks are unspecific
- They need high amount of resources
- Clock rate is much higher than microcontrollers
- Not well suited for real time apps
  - Execution times are highly unpredictable because of intensive resource sharing and dynamic decisions

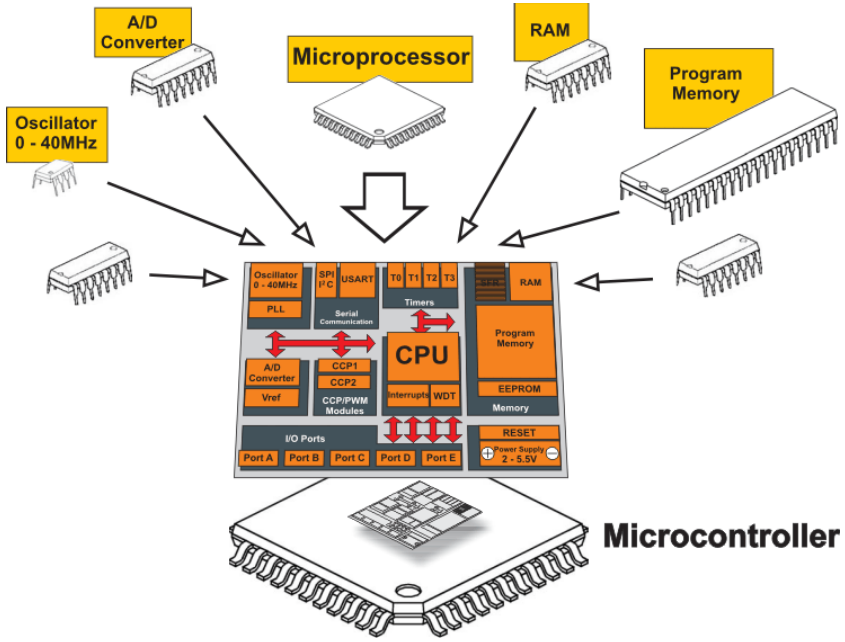




Microcontroller

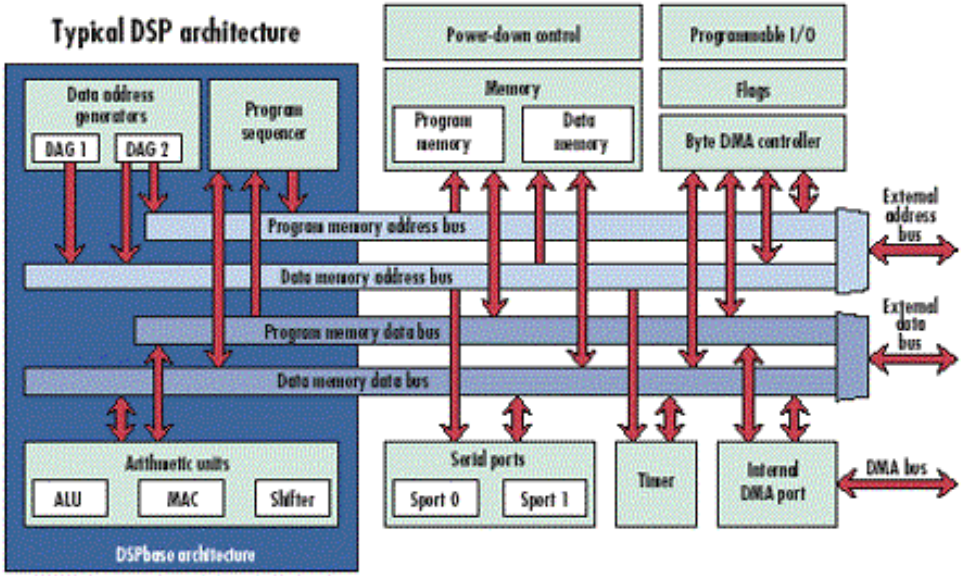
- μC has a CPU and fixed amount of RAM, ROM and other peripherals
- They are fabricated to specific application domains





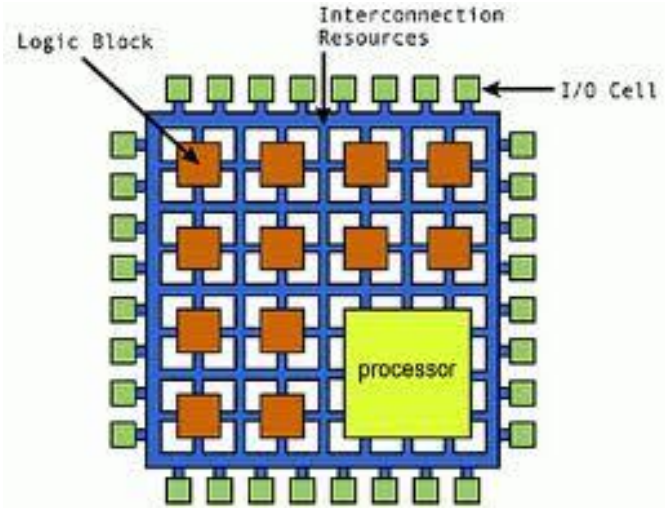
Digital Signal Processor

- A DSP is a specialized processor
- Specialized means that its internal hardware architecture is optimized for the typical mathematical operations needed in digital signal processing

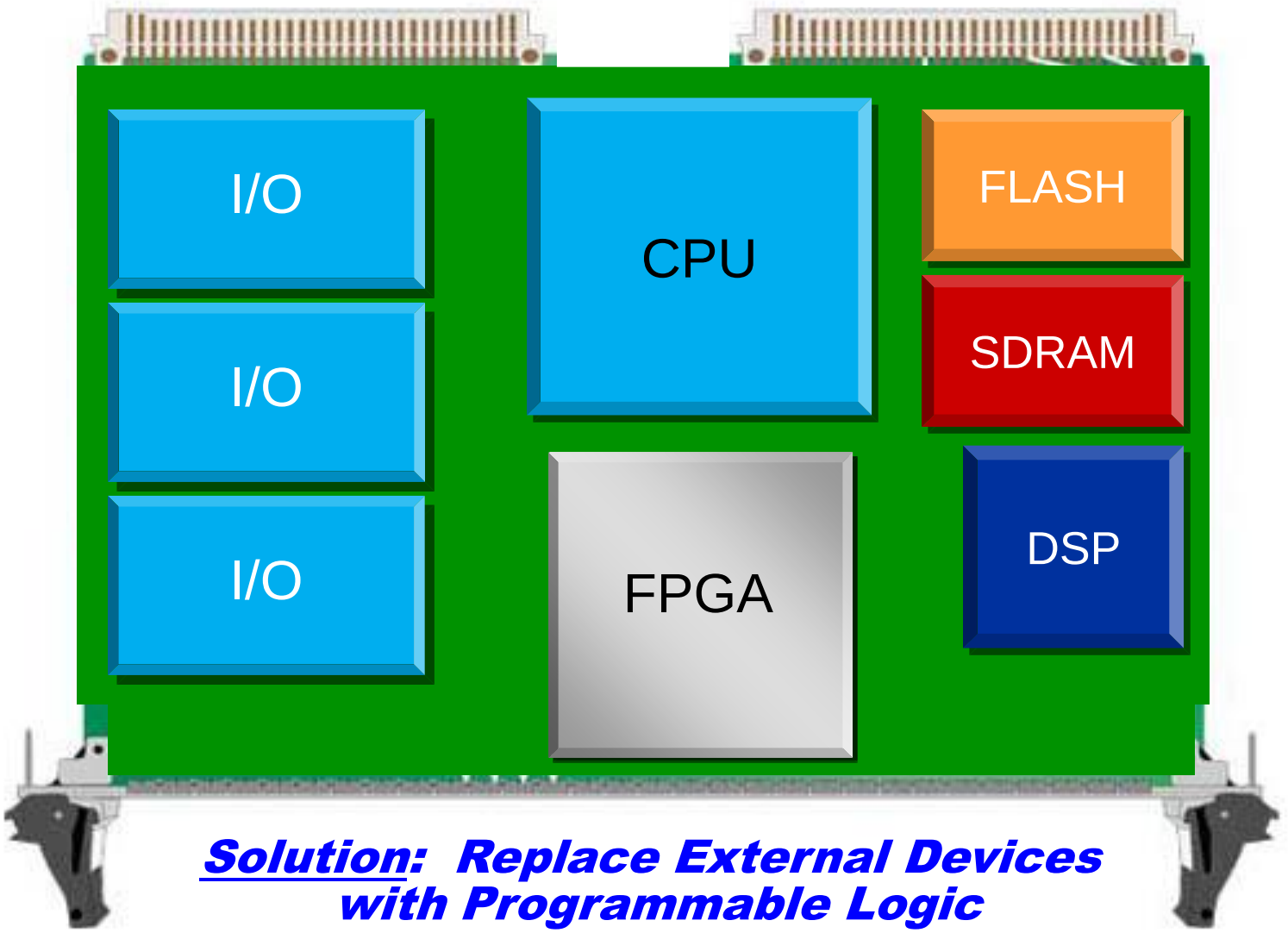


Other processor alternatives

- Soft-core processor or soft processor
  - It is a hardware description language (HDL) model of a specific processor that can be customized for a given application and synthsized for an ASIC or FPGA target
  - Advantages
    - Reduced cost
    - Flexibility
    - Platform independence
    - Greater immunity to obsolecence

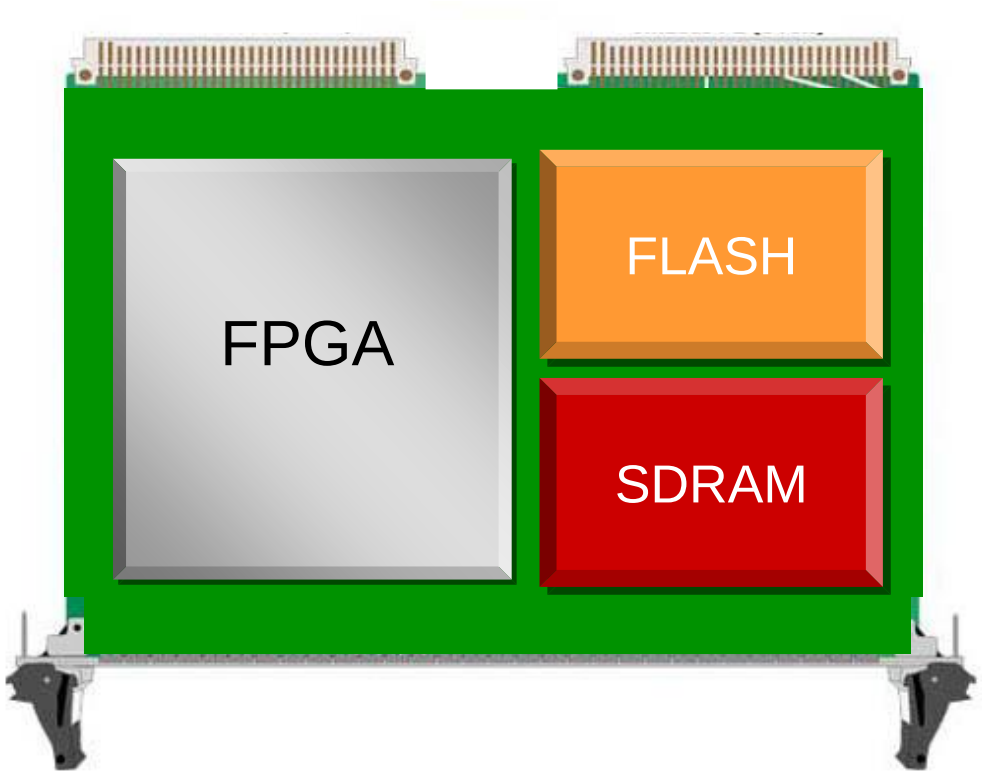


Problem: Reduce Cost, Complexity, and Power



Solution: Replace External Devices with Programmable Logic

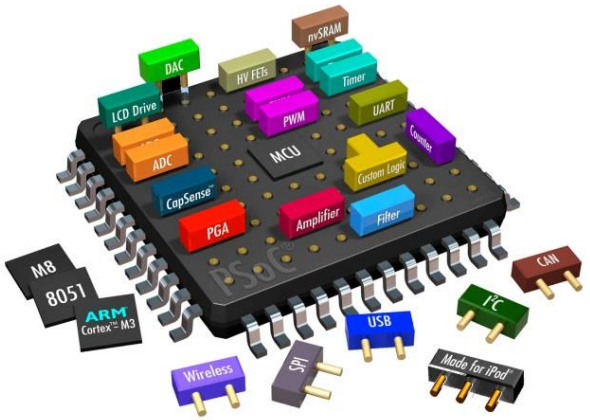
System on Programmable Chip

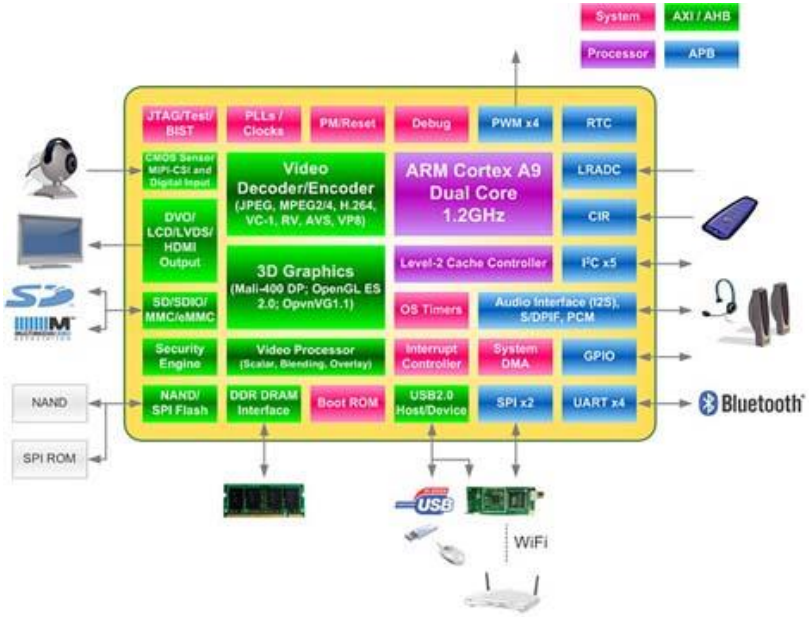


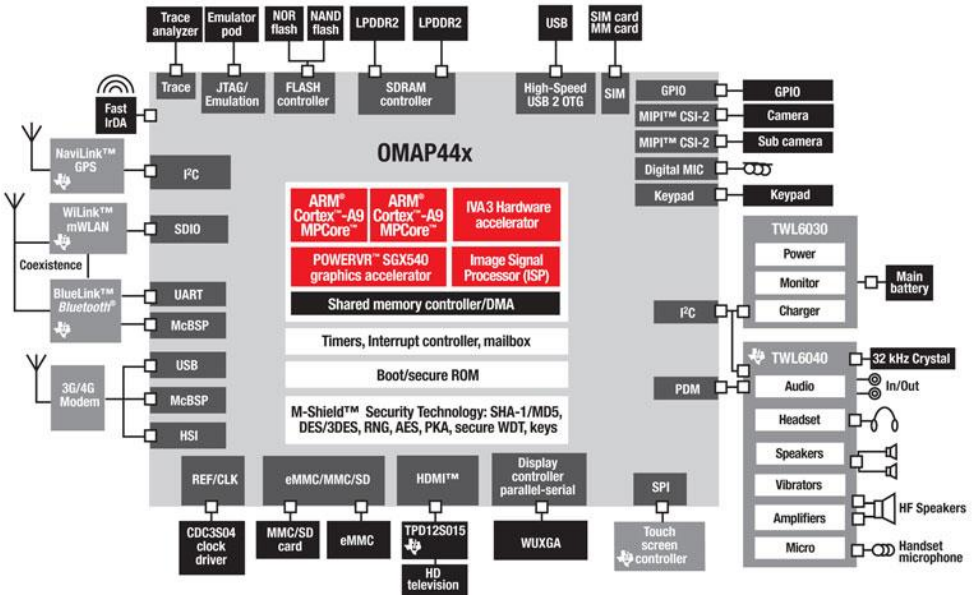
CPU is a Critical Control Function Required for System-Level Integration

Other processor alternatives

- System on Chip
  - An IC that integrates all electronics components necessary to implement a complet system

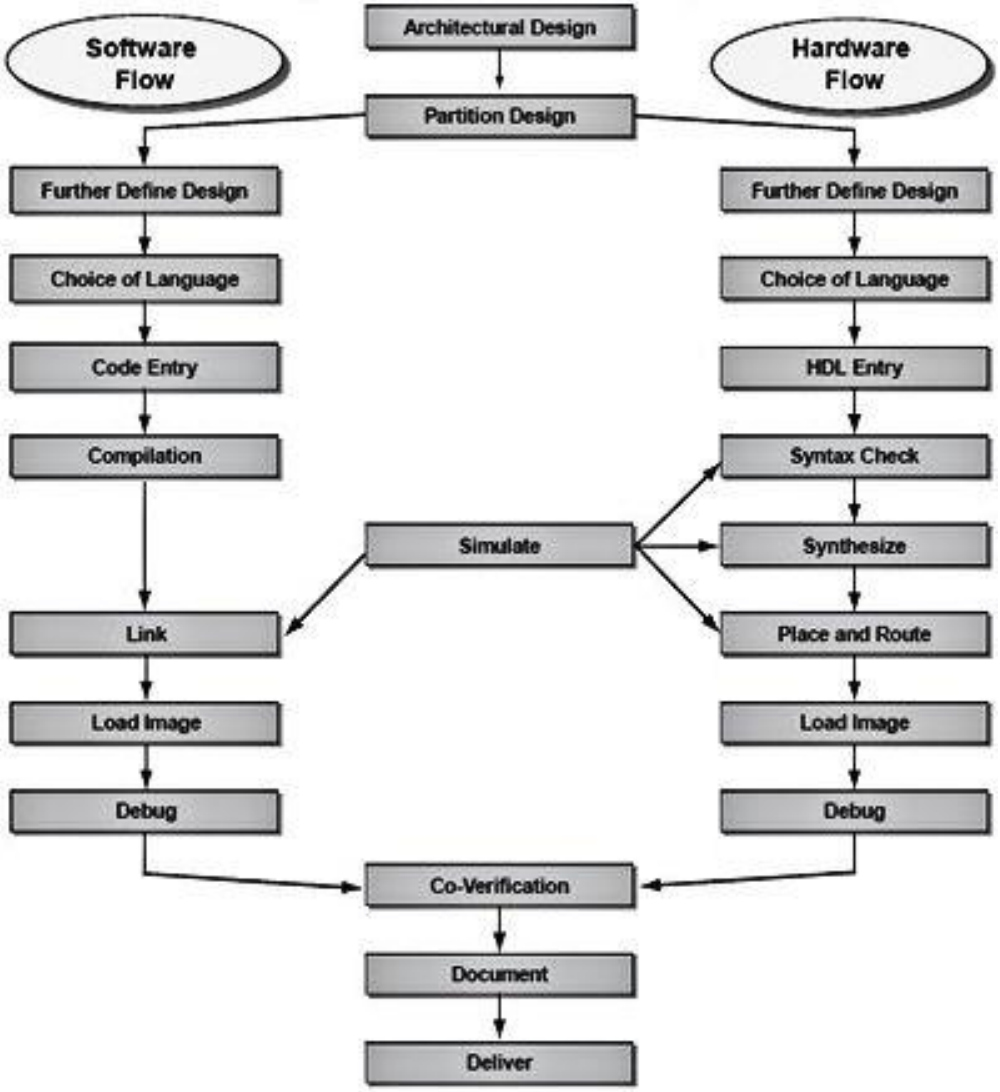




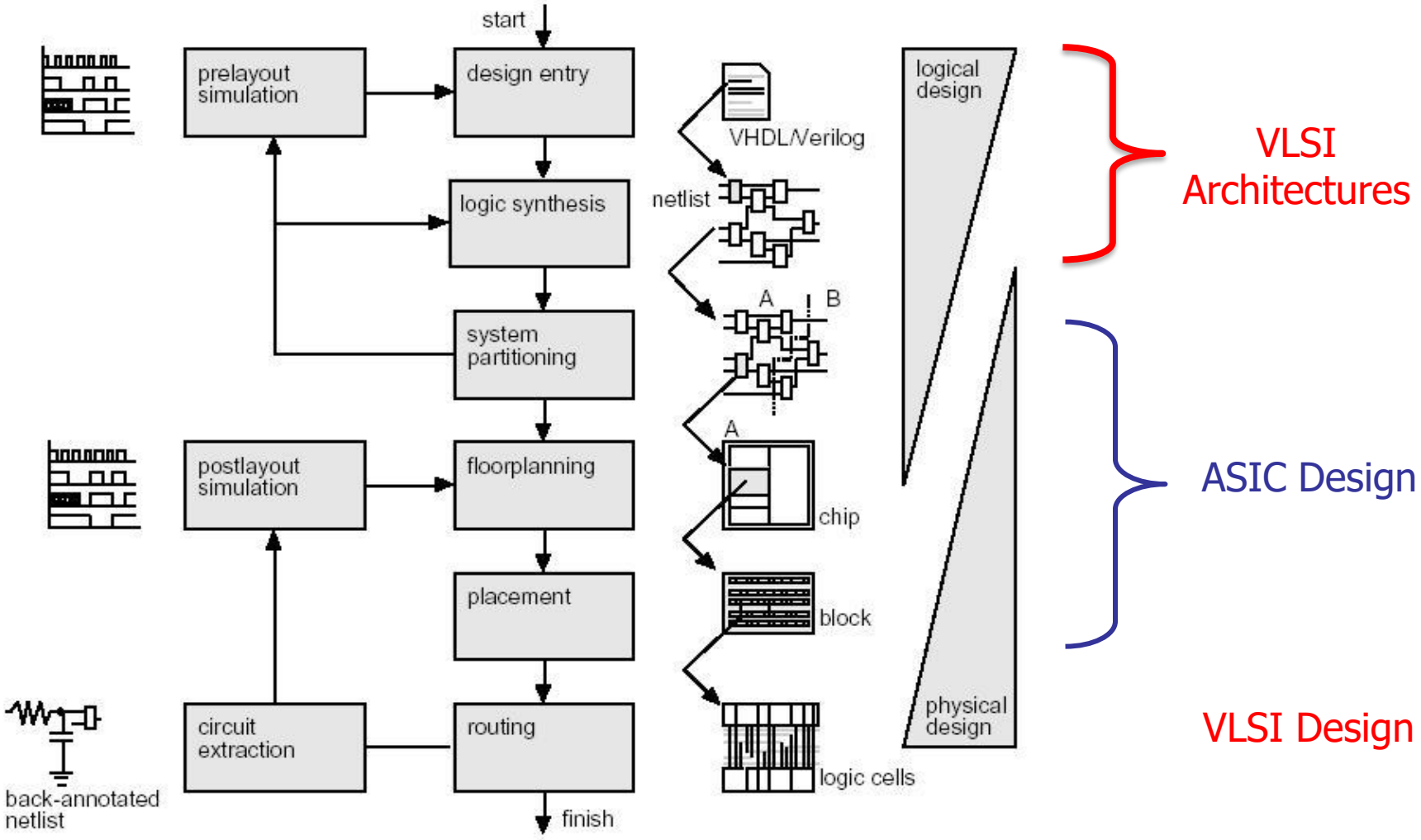


Embedded System Design Flow

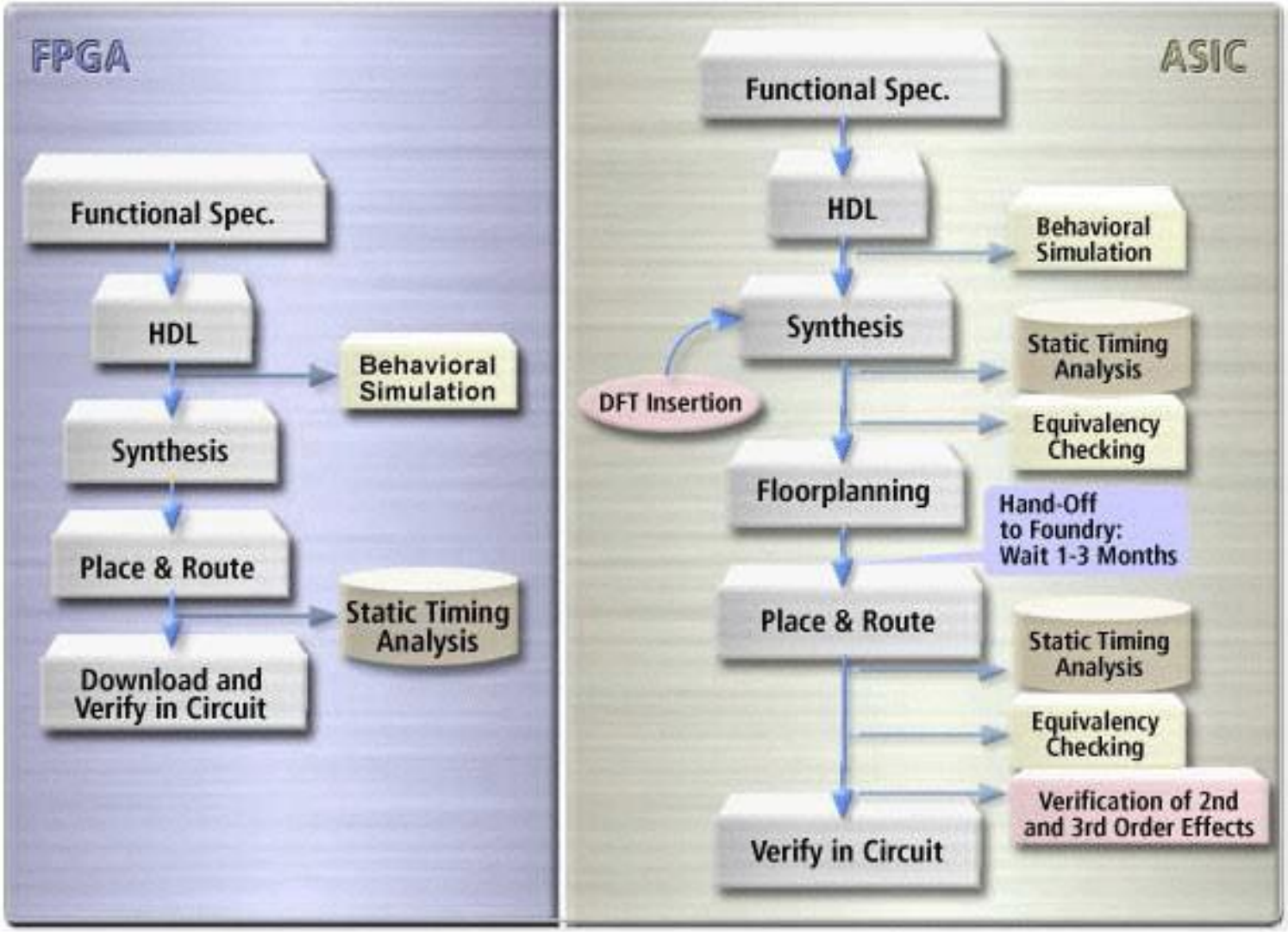
- Embedded system design involves HW & SW codesign
- As a result, embeded systems design teams follow two design flow
  - HW design flow
  - SW design flow
  - … and ID/ME design flow!



HW design flow

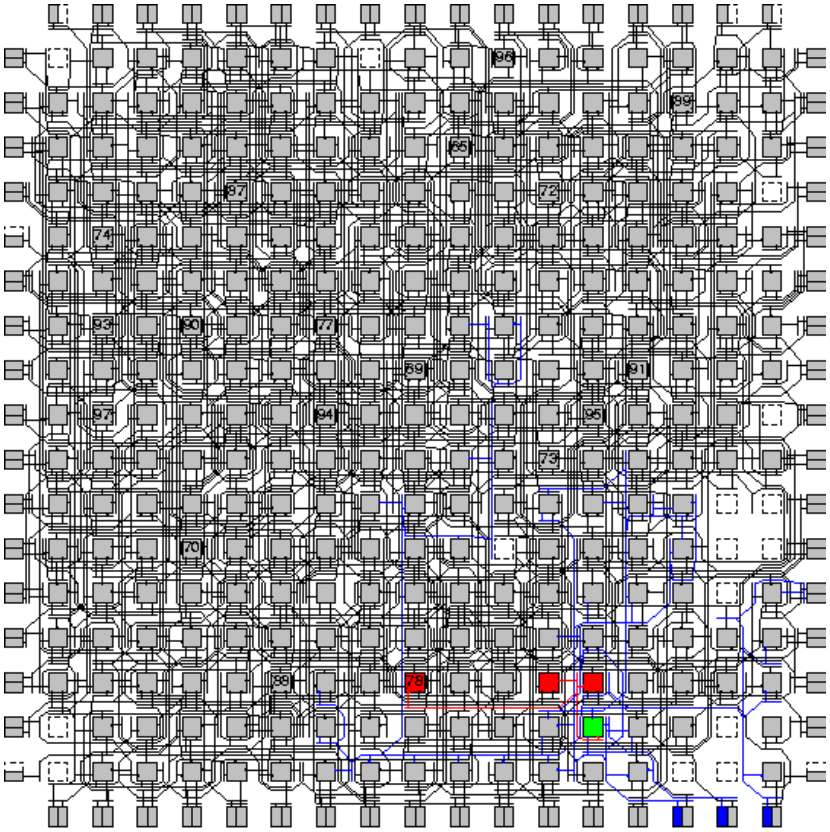


HW design flow



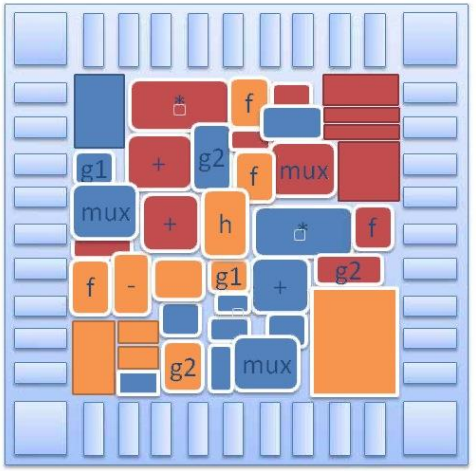
Place & Route in FPGA & ASIC

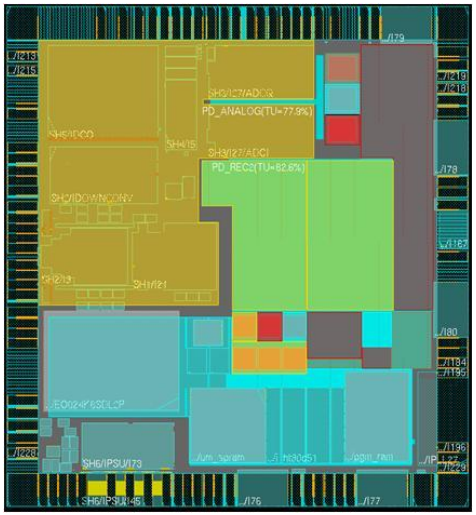
- FPGA



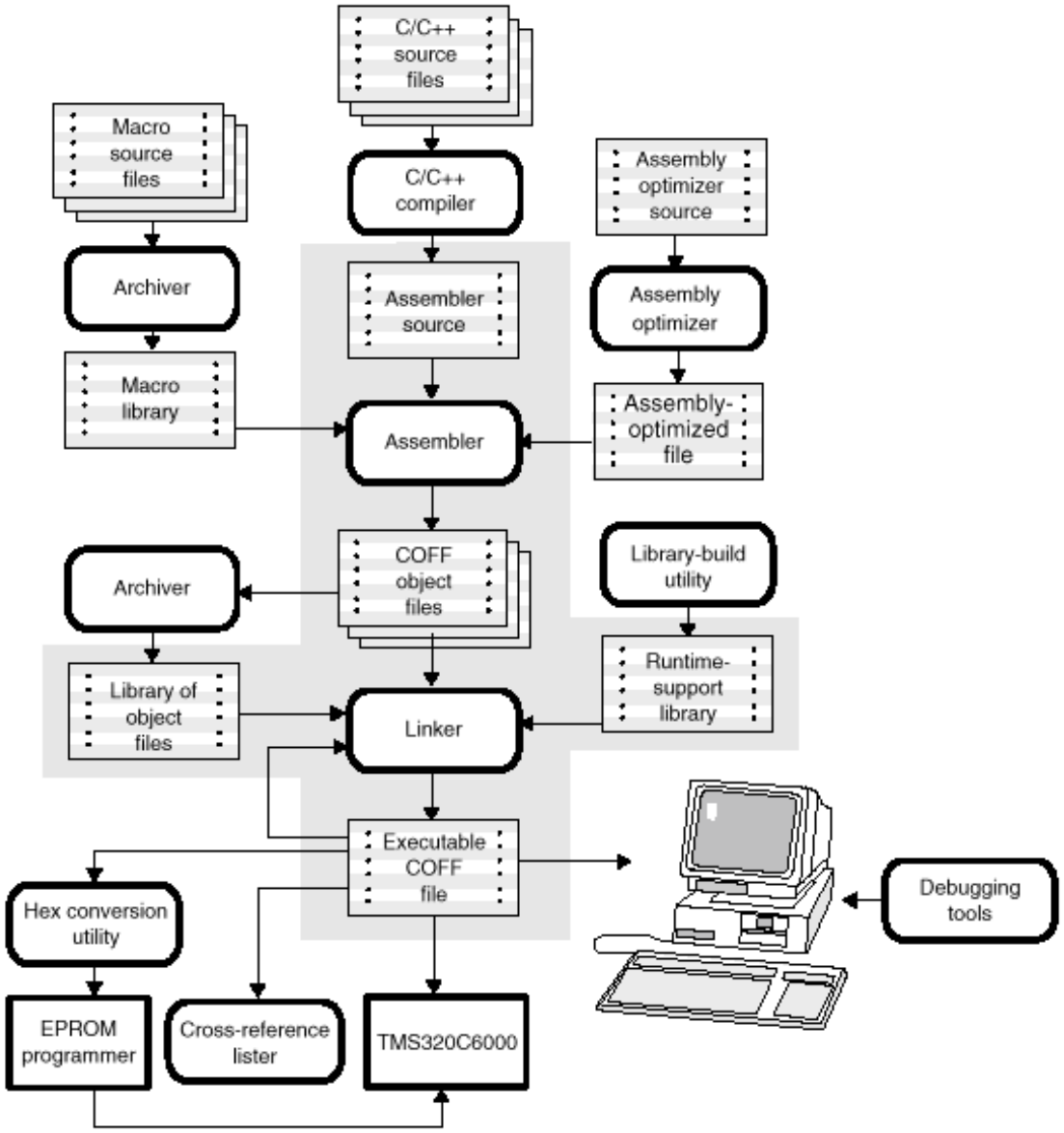
Routing succeeded with a channel width factor of 7.

- ASIC





SW design flow



Summary

- Embedded systems have evolved in a breathtaking pace in the last years
- Semiconductor technology guarantees that this trend will continue. At presente time, the spectrum of technology targets are broad
- Embedded system is the area where hardware design, software development, signal processing and ID/ME gather…
- The main feature of embedded systems is that they have strict performance (functionality and time execution) and cost requirements