SISTEMI EMBEDDED

Embedded Systems
SOPC Design Flow

Federico Baronti

Last version: 20160229
Definition(s) of Embedded Systems

- Systems with embedded processors

- One or more computers are used as components

- A physical system that employs computer control for a specific purpose, rather than for general-purpose computation

- Device that includes a programmable computer but is not itself intended to be a general-purpose computer
Embedded Systems (1)

• Not easy to find a precise definition. But we can identify some prominent features:
  – Contain a computer which is **not** general-purpose
  – Are part of a larger device/equipment/plant that they control
  – Are application-specific and single functioned; functions are a-priori known and executed repeatedly
  – Need to be optimized for cost, size, energy consumption,…
  – May have Real time and safety requirements
  – Typically have minimal user interface
Embedded Systems (2)

• Are very pervasive in our daily life:
  – Household appliances, cars (ABS, EPS,...), ATMs, printers, scanners, cameras, videogames, TVs, smart watches, alarm systems, “smartphones”, “tablets”,...

• Every year billion of new embedded computers
  – 100 embedded computers every one general-purpose computer
Application Examples (1)
Application Examples (2)

Microwave Oven

Camera
Embedded System Block Diagram

Embedded Computer

- Processor
- Memory
- Peripherals

Embedded System

- DRIVER
  - Actuator
  - Actuator
  - Indicator

Device/Equipment to be controlled
An **embedded system** is a computer system that is **not** general-purpose like a personal computer.

Main building blocks:

- **Processor(s)**
- **Memories**: FLASH, EEPROM, SRAM, SDRAM, DDRAM
- **Peripherals**: GPIO, Timer/Counter, PWM, Communication interfaces (SPI, I²C, CAN,...), A/D, D/A,...
- ...
- and definitely **some “specific” SOFTWARE**
An embedded system is a computer system that is not general-purpose like a personal computer.

Main design constraints:

- Development cost (*Non-Recurring Engineering cost*)
- Production cost
- Power consumption
- Physical size

- Their relative importance varies from one embedded system to another.
An **embedded system** is a computer system that is **not** general-purpose like a personal computer.
System-on-Programmable-Chip

• Configure soft-core processor:
  – Core configuration
    • Core version (E.g. economy, standard, fast for Altera Nios II)
    • Instruction/Data Cache, Pipeline Stages, JTAG Debug Modules, Custom Instructions, etc.
  – Peripheral configuration (what and where)
    • Peripheral selection
      – Standard peripherals from Altera and third-party vendors: GPIOs, Timers, Serial Communication Interfaces, Memory Interfaces, etc.
      – Custom peripherals
    • Address mapping
SOPC Design Flow

Altera’s CAD tools

- Logic Design: **Quartus II**
  - Nios II Configuration and **Computer** integration: **Qsys**
  - **Qsys** generates the HDL description of the **Computer**
- Logic Simulation: **ModelSim- Altera**
- Software Development: **Nios II Embedded Design Suite (EDS)** – Eclipse
- DE2: Development & Education board
  - Cyclone II EP2C35F672C6 (33216 LE; 105 M4K)

<table>
<thead>
<tr>
<th></th>
<th>Nios II/e</th>
<th>Nios II/s</th>
<th>Nios II/f</th>
</tr>
</thead>
<tbody>
<tr>
<td>#LE</td>
<td>600-700</td>
<td>1200-1400</td>
<td>1400-1800</td>
</tr>
<tr>
<td>#M4K</td>
<td>2</td>
<td>2 + cache</td>
<td>3 + cache</td>
</tr>
</tbody>
</table>

**Nios II versions:**
- economy
- standard
- fast
Nios II HW/SW Design Flow

Hardware System
Generated
in SOPC Builder

Hardware System
Generated
in Qsys

Software Project
Created
in Nios II IDE

Software Project
Created
in Nios II SBT
(Eclipse or Command Line)
Nios II SBT Design Flow

• Creating a project
  – Nios II Application and BSP from Template
    • Target hardware information (.sopcinfo, CPU)
    • Project template
    • Board Support Package (BSP)

• Code editing (.c, .h)

• Building the Project (.elf)

• Configuring the FPGA
  – Quartus II programmer (.sof)

• Running/ Debugging the Project on Nios II
  – Run/Debug configurations
Board Support Package (BSP)

• Library and header files (*e.g.* `system.h`) specific to the target processor
• Automatically generated through `.sopcinfo` and CPU
• Hides memory map, available devices, device implementation and processor configuration
  – Device drivers
  – Hardware Abstraction Layer (HAL)
  – RTOS: Micrium MicroC/OS-II