EEL 4783: Hardware/Software Co-design with FPGAs

Lecture 1: Introduction*

Prof. Mingjie Lin

* Partial material taken from NSU CS4722 slides
Overview

• What is an embedded system?
• Why HW/SW Co-design?
• Why take this course?
• Class mechanics
  – Administrative issues
  – Lecture topics
  – Assignment and projects
What is an Embedded System?

• ES from 10,000 Feet Above
  – a computer system designed for specific control functions within a larger system
  – often with real-time computing constraints
  – embedded as part of a complete device often including hardware and mechanical parts

• By contrast, a general-purpose computer
  – designed to be flexible and to meet a wide range of end-user needs
Embedded Computing Systems

• Obvious examples:
  – HDTV
  – Washing Machines
  – Microwave
  – Controllers for other household devices such as A/C
  – Digital watches
  – MP3 players

• Not-so-obvious examples:
  – Automobiles
  – Avionics / Flight control
  – Nuclear Power Plants
  – Medical devices
In-Depth Example: Mobile phones

- Multiprocessor
  - 8-bit/32-bit for UI
  - DSP for signals
  - 32-bit in IR port
  - 32-bit in Bluetooth
- 8-100 MB of memory
- All custom chips
- Power consumption & battery life depends on software
But …

• Dual-core A5 chip
  – package on package (PoP) system-on-a-chip (SoC)
  – 45 nm Dual core GPU PowerVR SGX543MP2 clocked at 200 MHz

• 8MP camera and optics
• iOS 5 and iCloud
• Siri
In-Depth Example: Cars

• Multiple processors
  – Upto100
  – Networked

• Multiple networks
  – Body
  – Engine
  – Telematics
  – Media
  – Safety
Cars

• Function diversity
  – ABS: Anti-lock braking systems
  – Airbags
  – Efficient automatic gearboxes
  – Theft prevention with smart keys
  – Blind-angle alert systems

• Device diversity
  – 8-bit – door locks, lights, etc.
  – 16-bit – most functions
  – 32-bit – engine control, airbags
Little-Known Facts about Cars

• Car electronics is an increasingly important market, requiring new design flows
  – Software is important for value addition

• Comments by major manufacturers
  – Daimler Chrysler: More than 90% of the innovation is from the car electronics (and not from the mechanical parts!)
  – BMW: More than 30% of the manufacturing cost of a car is from the electronic components!

• Reliable/robust ES design flows needed!
ES Design Challenges

• Real-time and/or Reactive
  – Often combines hard and soft real-time
  – Timing constraints on the response

• Low power budget
  – Novel architectures etc.

• High code density
  – Aggressive Code compression possible

• Profile driven development all important
Hardware/Software Design Methodology

• System Modeling
  – Irrespective of which parts are implemented in hardware and which parts in software
  – various choices of Models of Computation for reactive real-time systems

• HW/SW Partitioning
  – HW: Can be reconfigurable (FPGA)
    • Soft core or hard core
    • Function blocks
  – SW: Run on micro-controllers or more complex processors.
    • Further allocation needed if multiple processing elements (PEs) are available.
Hardware/Software Design Methodology

- **Compute Scheduling**
  - After allocation of tasks to PEs
  - Determines order in which tasks allocated to the same PE will be invoked so that
    - Performance constraints (deadlines) are met
    - Any dependencies between tasks are preserved
    - Communication/context-switch overheads in execution are minimized if possible

- **Communication synthesis**
  - Simple: Replace shared var. names by appropriate locations
  - Complex: Design interfaces to enable communication among design components
Why This Course?

- Because it is FUN intellectually!
- Because HS-Codesign become increasingly more critical

Aerospace/Defense

Broadcast

Consumer

HPC and storage

Industrial/Scientific/Medical

Wired communication

Wireless communication

Automobile
Class goal

- Learn about basic concept and techniques of hardware/software co-design, ...

- Hands-on class projects
  - Complete FPGA design flow to implement a “real” embedded computing system
  - Improve your HDL programming skills
  - Improve your Software programming skill
  - Learning by doing
Administrative issues

• Fill out the student info sheet
  – Name, status, reason of taking this class, expectations, prior knowledge, …

• Pre-requisites
  – EEL 3342: Digital Logic Design
  – Course self-contained, but logic design and computer architecture knowledge helpful (EEL 4768: Computer Architecture)
  – Willingness to work hard

• Information distribution
Lecture schedule

See Webpage:
www.eecs.ucf.edu/~mingjie/EEL4783_2012
Final issues

• Please fill out the student info sheet before leaving

• Come by my office hours (right after class)

• Any questions or concerns?