EE382V: System-on-a-Chip (SoC) Design

Lecture 0 – Class Overview

Jacob Abraham
Department of Electrical and Computer Engineering
The University of Texas at Austin
jaa@cerc.utexas.edu

Lecture 0: Outline

• Introduction
  • Systems-on-Chip (SoCs)
  • Design flow

• Course information
  • Overview, goals, topics
  • Administration
  • Labs and project

• Digital radio class project
  • DRM software receiver
  • DRM SoC implementation
Industrial Structure Shift

• Ubiquitous, embedded computing
  • From personal to dedicated computers

Source: SONY Corp & Market Estimates

System-on-Chip (SoC) Era

Source: SONY Corp & Market Estimates
**Integrated Circuits and Systems**

- **Wireless Communications**
- **Mixed-Signal System Design**
- **EE382V: RF-IC Design**
- **EE382M-14: Analog-1**

**System: HW & SW**

- **EE382V: Embedded System Design**
- **EE382N-4: Embedded Arch**
- **EE382M-1: VLSI Testing**
- **EE382M-11: Verification**
- **EE382M-8: VLSI-II**
- **EE382M-7: VLSI-I**

**Macro (RTL) Components**

**Cells**

---

**SoC Methodology Flow Chart**

1. **Start**
2. **MRD**
3. **PRD**
4. **Map Model & Scenarios in VHDL or ModelSim or C++ or C**
5. **Mapping to Platforms or Components Complete?**
6. **Analyze results**
7. **Functionality Met?**
8. **System BOM Costs Met?**
9. **Power Req. Met?**
10. **Schedule Req. Met?**
11. **Platform Req. Met?**
12. **Return**

**MRD Met?**

**PRD Met?**

---

(c) 2010 A. Gerstlauer
Course Overview

• Provide an understanding of the concepts, issues, and process of designing highly integrated SoCs
  • Systematic hardware/software co-design & co-verification

➢ Class labs and project: Software-defined radio SoC
  • DRM (Digital Radio Mondiale) system
    – Hardware/software co-design
  • State-of-the-art synthesis and verification tools and flows
    – High-level hardware synthesis from C++ to RTL
    – Virtual ARM platform modeling and simulation in SystemC
  • ARM-based FPGA prototyping platform
    – ARM processor, I/O devices, memory components, hardware accelerators

Course Goals

• Course is designed to learn about:
  • High-level system modeling and specification.
  • Early functional and nonfunctional performance analysis to support design decisions.
  • Analysis and optimization of hardware/software tradeoffs, algorithms, and architectures based on requirements and implementation constraints.
  • Architectures for control-dominated and data-dominated systems and real-time systems.
  • Hardware, software, and interface synthesis.
  • Interface design.
  • Co-simulation to validate system functionality.
  • Examples of applications and systems developed using a co-design approach.
  • Intellectual property, reuse, and verification issues.
Course Topics

- **Likely to be covered in class:**
  - System-level and SoC design methodologies and tools;
  - HW/SW Co-design: analysis, partitioning, real-time scheduling, hardware acceleration;
  - Virtual platform models, co-simulation and FPGAs for prototyping of HW/SW systems;
  - Transaction-Level Modeling (TLM) and Electronic System-Level (ESL) languages: SystemC;
  - High-Level Synthesis (HLS): allocation, scheduling, binding, resource sharing, pipelining;
  - SoC and IP integration, verification and test.

Course Administration

- **Instructors**
  - Course Coordinator: Jacob Abraham <jaa@cerc.utexas.edu>
    - Office Hours: ACE 6.124, MW 2:30 – 3:30 or by appt.
  - Guest lecturers from industry and academia:
    - Jacob Abraham, Xtreme EDA, Mark McDermott, Steven Smith
  - TA: Hyungman Park <hpark@cerc.utexas.edu>
    - Office hours: ENS 113A, TBD

- **Information**
  - Web: http://www.cerc.utexas.edu/~jaa/soc/
    - Lecture notes, homework and lab exercises.
  - Blackboard
    - Announcements, assignments, grades.

- **Dates (tentative)**
  - Exam: November 12
  - Final project presentations: December 3
Course Policies

- **Grading:**
  - Homework: 15%
  - Lab assignments: 30%
  - Exam: 20%
  - Project: 35%

- **Late penalties:**
  - 20% per day (24 hours)

- **Academic dishonesty**
  - Homeworks
    - Ok to discuss questions and problems with others
    - But turn in own, independent solution
  - Labs and project
    - In teams, one report and presentation
    - Collaboration encouraged and desired
  - Plagiarism
    - Use of any outside source of information without quoting or referencing is cheating

Labs (tentative)

- **Lab 1: DRM on ARM (3 weeks, due Sept. 25)**
  - Profiling of DRM code on Linux host and ARM simulator
    - Identify time consuming bottlenecks to optimize
  - Float to fixed point conversion
    - Improve performance without loss in accuracy (SNR)

- **Lab 2: DRM system architecture (due October 16)**
  - ARM plus hardware acceleration
    - Identify and partition DRM code into hardware and software
  - Virtual platform modeling and simulation
    - Isolate and interface hardware as SystemC module
    - Co-simulation w/ software and firmware on ARM-Linux using OVPsim

- **Lab 3: DRM hardware synthesis (due Nov. 6)**
  - High-level synthesis of Viterbi decoder from C++ to RTL
    - Mentor Catapult-C synthesis tool
  - Verification of synthesis results
    - Testbench around standalone C++ vs. RTL hardware module
Class Project

- **DRM implementation on prototyping board**
  - Synthesizing hardware into the FPGA
    - DRM modules, at minimum Viterbi decoder
    - Synthesis and download using Xilinx software
  - C/C++ program on the ARM host processor
    - Cross-compilation and download to run under Linux on the ARM
  - Hardware/software interfacing and communication
    - Software drivers and hardware bus interfaces
  - Analysis and validation of product metrics
    - Estimate timing, area and power consumption

  ➢ **Low-power SoC implementation of DRM receiver**
  - Reference design of DRM ASIC
    - FPGA-based implementation as prototype of ASIC design
  - To be incorporated into MP3 players or cell phones
    - Satisfy market and product requirements

Software Radio for DRM

- **Digital Radio Mondiale (DRM)**
  - World standard for digital broadcasting in the AM radio bands below 30 MHz.
  - DRM-capable software radio developed by Fraunhofer (FhG) Institut für Integrierte Schaltungen
    - Push DRM technology and adoption thereof
    - Early availability and an easy way to reproduce the radio.
Commerially Available DRM Receivers

**Coding Technologies**

- Himalaya
- iGear

**TI DRM Chip**

- Texas Instruments TMS320DRM300/350

![Diagram of TI DRM Chip]
### DRM Broadcast Schedule Examples

<table>
<thead>
<tr>
<th>UTC</th>
<th>Days</th>
<th>kHz</th>
<th>Beam</th>
<th>Target</th>
<th>Power</th>
<th>Programme</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0059 daily</td>
<td>1431</td>
<td>ND</td>
<td>Canberra</td>
<td>0.05</td>
<td>MCS</td>
<td>English</td>
<td></td>
</tr>
<tr>
<td>0000-0059 daily</td>
<td>9790</td>
<td>227</td>
<td>NE USA</td>
<td>70</td>
<td>TD Radio</td>
<td>Dance Music</td>
<td></td>
</tr>
<tr>
<td>0000-0200 daily</td>
<td>177</td>
<td>ND</td>
<td>Germany</td>
<td>150</td>
<td>DLR Kultur</td>
<td>German</td>
<td></td>
</tr>
<tr>
<td>0000-2400 daily</td>
<td>1386</td>
<td>ND</td>
<td>AUS-NSW</td>
<td>3</td>
<td>ABC</td>
<td>English</td>
<td></td>
</tr>
<tr>
<td>0000-2400 daily</td>
<td>1006</td>
<td>ND</td>
<td>Prov. Hunan</td>
<td>4</td>
<td>Economic Ch.</td>
<td>Chinese</td>
<td></td>
</tr>
<tr>
<td>0000-2400 daily</td>
<td>999</td>
<td>ND</td>
<td>Paris</td>
<td>8</td>
<td>DRM test</td>
<td>French</td>
<td></td>
</tr>
<tr>
<td>0000-2400 daily</td>
<td>25775</td>
<td>ND</td>
<td>Rennes</td>
<td>0.1</td>
<td>TDF Radio</td>
<td>French</td>
<td></td>
</tr>
<tr>
<td>0000-2400 daily</td>
<td>25775</td>
<td>ND</td>
<td>Côte d’Azur</td>
<td>0.7</td>
<td>AGORA</td>
<td>French</td>
<td></td>
</tr>
<tr>
<td>0000-2400 daily</td>
<td>59500</td>
<td>ND</td>
<td>Rennes</td>
<td>0.15</td>
<td>TDF</td>
<td>French</td>
<td></td>
</tr>
</tbody>
</table>

### Dream DRM Receiver

- **Software implementation of DRM Receiver**
- Open-source alternative to commercial FhG software