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

Lecture 1 – Project Overview

Andreas Gerstlauer
Electrical and Computer Engineering
University of Texas at Austin
gerstl@ece.utexas.edu

Lecture 1: Outline

• Marketing Requirements Document (MRD)
  • Market focus
  • Product description
  • Cost metrics
  • Product features
  • References

• Project description
  • Overview
  • Hardware and software development tasks
Market Focus

- **Cellphones that receive digital radio transmissions**
  - Estimated market size is billion of units per year
  - Receive FM and DRM/DAB/HD Radio transmissions

➤ **What problem are we trying to solve?**
  - There is a need to receive digital radio and data using various standards
  - Flexible receiver that can be adapted to different markets, changing environments, etc.
  - Support for other wireless communications (WiFi) in one chip?
  ➤ Software-defined radio (SDR)

Competition

- **Texas Instruments TMS320DRM300/350**
Lecture 1: Outline

• Marketing Requirements Document (MRD)
  ✓ Market focus
  • Product description
  • Cost metrics
  • Product features
  • References

• Project description
  • Overview
  • Hardware and software development tasks
  • TLL5000 Prototyping board

Product Description

• SDR SoC to integrate into cell phone
  • The hardware intellectual property will be delivered in a SystemC environment. This will include synthesizable RTL for all components which are not available in the standard library, such as accelerators, special I/O devices, etc.

• DRM benefits
  • Ability to receive digital music and data
    – Using existing long-, medium- and short-wave transmission systems
    – Providing near-FM quality sound and available to markets worldwide.
  • Small bandwidth of less than 20 kHz
    – Easy to handle with current generation of embedded computing devices.
  • Excellent audio quality
    – Significant improvement upon analog AM
    – Range of audio content, including multi-lingual speech and music
  • Capacity to integrate data and text
    – Additional content can be displayed to enhance the listening experience.
  • Use existing AM broadcast frequency bands
    – Designed to fit in with the existing AM broadcast band plan
    – Signals of 9 kHz or 10kHz bandwidth
    – Modes requiring as little as 4.5kHz or 5kHz bandwidth, plus modes that can take advantage of wider bandwidths, such as 18 or 20kHz.
Cost Metrics

- **Performance**
  - Utilize no more than 25% of a dual-core ARM Cortex A9 running at 667Mhz

- **Additional die size cost**
  - Accelerators < 0.5 mm²
  - On board memory – TBD

- **Advanced system and power management**
  - Additional system power for accelerators < 8 mW

Product Features

- **Flexible and scalable platform based architecture**
  - **Standard architecture** for a wide range of devices supporting a wide range of services
  - **Flexibility** to dynamically re-program different digital radio standards tailored to particular scenarios
  - **Portability** to host third party designs on multiple independent platforms
    - Potential for significant life-cycle cost reduction
    - Over the air downloads of patches, new features & services
    - Significant improvement in flexibility, portability and interoperability between different users
Product Features (cont’d)

• DRM technical features
  • Frequency coverage: 0-32 MHz
  • Mode reception: USB, LSB, CW, AM, synchronous AM, NFM, DATA
  • Advanced IP3 greater than +35 dBm
  • Very high dynamic range
    – >100 dB in AM mode with 7 kHz filter
    – >105 dB in SSB mode with 2.2 kHz filter
    – >110 dB in CW mode with 500 Hz filter
  • Passband tuning: +/-5 kHz
  • Audio pitch tune in CW & DATA

DRM References

• DRM consortium
  • http://drm.org

• Commercial DRM software radio (Frauenhofer)
  • http://drmrx.org

• Receiver hardware
  • http://winradio.com

➢ Open-source DRM software (DREAM)
  • http://drm.sourceforge.net
Lecture 1: Outline

- Marketing Requirements Document (MRD)
  - Market focus
  - Product description
  - Cost metrics
  - Product features
  - References

- Project description
  - Overview
  - Hardware and software development tasks

Project Description

- HW/SW co-design of an embedded SoC
  - Low-power DRM implementation
  - ARM-based target platform
    - ARM A9 processor, memory components, I/O devices
    - Custom hardware accelerators
    - Interconnected via standard system bus
  - Virtual and physical prototyping
    - SystemC TLM-based virtual platform model (QEMU ARM simulator)
    - ARM- and Xilinx FPGA-based prototyping board (Zynq-7000)

  ➢ Lab and project teams
**Project Objectives and Activities**

- **Project objective:**
  - Implement the DRM C++ code on a ARM based SoC while meeting the performance, area and power metrics.

- **Project activities:**
  - Profile the DRM C++ software implementation to determine performance bottlenecks
  - Optimize the DRM C++ software (fixed point operation)
  - Partition the software into components which will run on the ARM processor and on hardware accelerators
  - Synthesize accelerators into Verilog for gate level implementation
  - Co-simulate and prototype the HW/SW implementation
  - Estimate timing, area and power metrics and validate against product requirements

**PC-Based DRM System Architecture**

- DRM reference code is designed to run on a desktop computer
DRAM Software Overview

- Sound card interface
- Frequency acquisition
- Sample rate correction
- Rob. mode detection
timing acquisition
  useful part extraction
- Frequency correction
  OFDM demodulation
- Resample, freq.
  offset tracking
  frame sync.
- Channel estimation
timing tracking
- Source decoders,
  channel decoders,
  OFDM demux

DRM Software Architecture

- Sound card interface
detection
- Resampling
equation
- Frequency sync. acquisition
  Frequency offset correction
  Channel estimation
- Time sync acquisition
  Source channel removal
  Timing estimation
- OFDM demodulation
- Sync string addition
  Source channel removal
  Timing estimation
- Channel estimation:
  Time sync tracking
  Resampling
- OFDM set demapping
  (IEEE1011 demapping)
Development Tasks

- **Hardware development on FPGA**
  - Hardware accelerators (using synthesized code)
  - Interface to ARM board and on-chip bus
  - Interrupt logic
  - Clocking & reset
  - Optional memory controller (for external DRAM)
  - Diagnostics

- **ARM software development**
  - Compile and profile DRM on ARM simulator
  - Convert floating-point to fixed-point code and check SNR
  - Compile and profile fixed-point DRM on ARM board
  - Parallelize software on dual-core platform
  - Develop hardware abstraction layer (HAL) and I/O handler
  - Develop interrupt handler & hardware drivers