Lecture 2 – DRM Project Overview

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

Lecture 2: 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
Market Focus

• MP3 players that receive digital radio transmissions
  • Estimated market size is approximately 5-7 million units per year
• It is anticipated that the next generation cell phones may be configured to receive FM and DRM/DAB transmissions. If so…
  • The potential market size is approximately 35 million units per year

➤ What problem are we trying to solve?
  • There is a need to transmit and receive digital music and data using existing AM bands. Transmitters in these wavelengths are accessible world wide.
  • Need to provide near-FM quality sound and the capacity to integrate data and text.

Competition

• Texas Instruments TMS320DRM300/350
Lecture 2: 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

• DRM SoC to integrate into MP3 player or 4G 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 75 MHz of an ARM 926-EJS running at 256 MHz

- **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)

- **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](http://drm.org)

- **Commercial DRM software radio (Frauenhofer)**
  - [http://drmr.org](http://drmr.org)

- **Receiver hardware**
  - [http://winradio.com](http://winradio.com)

- **Open-source DRM software (DREAM)**
  - [http://drm.sourceforge.net](http://drm.sourceforge.net)
Lecture 2: 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

Project Description

• HW/SW co-design of an embedded SoC
  • Low-power DRM implementation
  • ARM-based target platform
    – ARM9 processor, memory components, I/O devices
    – Custom hardware accelerators
    – Interconnected via standard system bus
  • Virtual and physical prototyping
    – SystemC TLM-based virtual platform model (OVPsim ARM simulator)
    – ARM- and Xilinx FPGA-based prototyping board (TLL6219-TTL5000)

✓ Lab and project teams
Project Objectives and Activities

• Project objective:
  • Implement the DRM C++ code on a ARM based platform 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 for fixed point operation
  • Partition the software into components which will run on the ARM processor and on the hardware accelerators
  • Synthesize time-critical functions 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
DRM Software Overview

- sound card interface
- frequency acquisition
- sample rate correction
- OFDM demodulation
- resample, freq., offset tracking frame sync.
- channel estimation timing tracking
- DRM Software Architecture
- sample rate detection
- useful part extraction
- acquisition
- timing
- detection
- acquisition
- useful part
- extraction
- source decoders, channel decoders, OFDM demux

DRM Software Architecture

- Sound card interface
- Resampling
- Frequency synchronization
- OFDM demodulation
- Sync using pilot
- OFDM VLS demodulation
- Channel estimation, Time and Tracking frame sync.
- DRM Software Architecture
- Frequency synchronization
- OFDM demodulation
- Sync using pilot
- OFDM VLS demodulation
- Channel estimation, Time and Tracking frame sync.

CB: Cyclic buffer
SB: Single buffer
- Time domain
- Frequency domain
- QAM
- Bit stream
High-Level Hardware Architecture

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 SDRAM)
  - 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
  - Develop hardware abstraction layer (HAL) and I/O handler
  - Develop interrupt handler
Lecture 2: 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

TLL5000

• Prototyping platform
  • Base board
TLL5000 Architecture

TLL5000 Block Diagram
Xilinx Spartan 3 FPGA

<table>
<thead>
<tr>
<th>Device</th>
<th>System Gates</th>
<th>Equivalent Logic Cells</th>
<th>CLB Array (One CLB = Four Slices)</th>
<th>Distributed RAM Bits</th>
<th>Block RAM Bits</th>
<th>Dedicated Multipliers</th>
<th>DCI</th>
<th>Maximum User IO</th>
<th>Maximum Differential I/O Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>XC3S50</td>
<td>580K</td>
<td>12</td>
<td>120</td>
<td>12K</td>
<td>23K</td>
<td>22</td>
<td>2</td>
<td>124</td>
<td>66</td>
</tr>
<tr>
<td>XC3S20</td>
<td>280K</td>
<td>24</td>
<td>24</td>
<td>21K</td>
<td>33K</td>
<td>22</td>
<td>3</td>
<td>124</td>
<td>66</td>
</tr>
<tr>
<td>XC3S400</td>
<td>800K</td>
<td>32</td>
<td>32</td>
<td>32K</td>
<td>56K</td>
<td>24</td>
<td>4</td>
<td>264</td>
<td>116</td>
</tr>
<tr>
<td>XC3S4000</td>
<td>1000K</td>
<td>40</td>
<td>40</td>
<td>40K</td>
<td>32K</td>
<td>24</td>
<td>4</td>
<td>264</td>
<td>116</td>
</tr>
<tr>
<td>XC3S5000</td>
<td>2000K</td>
<td>60</td>
<td>60</td>
<td>70K</td>
<td>32K</td>
<td>24</td>
<td>4</td>
<td>264</td>
<td>116</td>
</tr>
<tr>
<td>XC3S6000</td>
<td>3000K</td>
<td>90</td>
<td>85</td>
<td>65K</td>
<td>50K</td>
<td>24</td>
<td>4</td>
<td>264</td>
<td>116</td>
</tr>
</tbody>
</table>

TLL6219 ARM Processor Board
### TLL6219 Block Diagram

- **USB 1.1**
- **RS232**
- **Ethernet**
- **ARM-9 Embedded Processor (iMX21)**
- **CPLD**
- **Address Buffers**
- **Data Buffers**
- **JTAG Header**
- **Expansion Port**
- **Flash Memory**
- **SDRAM Memory**
- **Data**
- **Address**
- **Control**

### i.MX21 Features

#### i.MX21 Applications Processor Block Diagram

- **Connectivity**
  - Internal: 3 x CSI, 2 x SSI, PC, Audio Mux
  - External: 4 x UART, USB-OTG Host, 1-Wire, iDA
  - Expansion: 2 x MMC/SD, PCMCI/CF

- **MC9328MX21**
  - ARM926EJ-S™
  - I-Cache, D-Cache, MMU
  - Internal Control, Bus Control, Memory Control

- **MEMORY INTERFACE**
  - SDRAMC
  - EIM/BMI
  - NAND Controller

- **ENHANCED MULTIMEDIA ACCELERATOR (eMMA)**
  - Pre and Post Processing
  - Video Accelerator

- **System Control**
  - JTAG/ICE™
  - Bootstrapping
  - Clock Management

- **Standard System I/O**
  - 3 x Timer
  - PWM
  - WD Timer
  - RTC
  - GPIO
  - DMA

- **Human Interface**
  - SLCD Controller
  - Keypad

- **Multimedia Interface**
  - Camera Interface
There are eight 512MB partitions

<table>
<thead>
<tr>
<th>Address</th>
<th>Size</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>512 Mbyte</td>
<td>ROM, Primary AHB Slaves, and Peripherals</td>
</tr>
<tr>
<td>0x20000000</td>
<td>512 Mbyte</td>
<td>Reserved</td>
</tr>
<tr>
<td>0x40000000</td>
<td>512 Mbyte</td>
<td>Reserved</td>
</tr>
<tr>
<td>0x60000000</td>
<td>512 Mbyte</td>
<td>Reserved</td>
</tr>
<tr>
<td>0x80000000</td>
<td>512 Mbyte</td>
<td>Secondary AHB Slave Port 1</td>
</tr>
<tr>
<td>0xA0000000</td>
<td>512 Mbyte</td>
<td>Secondary AHB Slave Port 2</td>
</tr>
<tr>
<td>0xC0000000</td>
<td>512 Mbyte</td>
<td>Secondary AHB Slave Port 3</td>
</tr>
<tr>
<td>0xE0000000</td>
<td>512 Mbyte</td>
<td>Primary AHB (RAM)</td>
</tr>
</tbody>
</table>
### TLL6219 ARM926EJ-S Board

- **External interfaces**
  - RS-232 serial port
  - Ethernet
  - USB-OTG (Linux host driver for flash disk)
  - Graphic LCD panel
- **TLL5000 Interface**
  - External memory interface
  - /CS1, /CS5 memory regions
  - D[31:0], A[23:0], control signals (thru CPLD)
  - Connections to TLL6219 CPLD

- **Interface from ARM to hardware**
  - Exclusively through Chip Select 1 & 5 memory regions
  - All TLL5000 peripherals must be accessed through the FPGA
  - The only direct connection to the ARM9 is
    - LCD, RS-232, USB, Ethernet

---

### System Block Diagram
TLL6219 CPLD Connections

- CPLD_INT connects to PF[16]
- MISC[xxxxx] signals defined by CPLD
- /DTACK for cycle timing
Connector B

TLL6219 CPLD Overview

- The CPLD generates read and write strobes for accesses in the /CS1 and /CS5 spaces (combinational logic)
  - \( cs1_{rs}_b = \neg (\neg cs1_b \& \neg oe_b); \)
  - \( cs1_{ws}_b = \neg (\neg cs1_b \& \neg (&eb) \& \neg rw_b); \)
  - \( cs5_{rs}_b = \neg (\neg cs5_b \& \neg oe_b); \)
  - \( cs5_{ws}_b = \neg (\neg cs5_b \& \neg (&eb) \& \neg rw_b); \)
- /DTACK is synchronized in the CPLD
  - Single flip-flop synchronizer
- Transceiver control
  - The NFIO4 jumper controls data transceiver operation when the ARM is not accessing /CS1 or /CS5 space
    - If the jumper is NOT installed, the data transceivers are disabled
    - If the jumper IS installed, the data transceivers are enabled toward the FPGA to permit snooping bus activity not in the /CS1 or /CS5 spaces
TLL6219 CPLD_MISC[] Pins

mz_cpld_misc[0] = cs1_rs_b; //CS1 read strobe (active-low)
mz_cpld_misc[1] = cs1_ws_b; //CS1 write strobe (active-low)
mz_cpld_misc[2] = cs5_rs_b; //CS5 read strobe (active-low)
mz_cpld_misc[3] = cs5_ws_b; //CS5 write strobe (active-low)
mz_cpld_misc[4] = oe_b; from ARM96
mz_cpld_misc[5] = cs0_b; from ARM926 (flash memory)
mz_cpld_misc[6] = cs1_b; from ARM926 (FPGA access)
mz_cpld_misc[7] = cs2_b; from ARM926 (SDRAM)
mz_cpld_misc[8] = cs3_b; from ARM926 (Ethernet)
mz_cpld_misc[9] = cs5_b; from ARM926 (FPGA access)
mz_cpld_misc[10] = nfio4; TLL6219 jumper
mz_cpld_misc[11] = nfio5; TLL6219 jumper
mz_cpld_misc[12] = data_dir; TLL6219 transceiver control
mz_cpld_misc[13] = data_oe; TLL6219 transceiver control
mz_cpld_misc[14] = fpga_interrupt; FPGA IRQ to ARM926 PF[16]

iMX21 External Interface Module (EIM)

• The EIM permits fine-grained control of the bus interface
  • Bus width
  • Timing of /CSx assertion/negation
  • Timing of /OE, /WE assertion/negation
  • Dead cycles between transfers
  • DTACK sensitivity and sampling
  • Byte enable behavior
  • Burst mode


**iMX21 EIM Timing Example**

![Diagram of EIM Timing Example](image)

*Figure 77. DTACK Edge Triggered Read Access, WSC=3F, OEA=8, OEN=5, AGE=1.*

**EIM Configuration in Boot Monitor**

- **Chip Select 1 & 5 Upper Register settings in uMon**
  - CS1U,CS5U = 0x00000480
    - DCT = 0, at least 2 HCLK before /DTACK checked
    - RWA = 0, R/W asserted when address valid
    - WSC = 4 wait states (minimum cycle = 6 HCLK)
    - EW = 1, level sensitive /DTACK

- **Chip Select 1 & 5 Lower Register settings in uMon**
  - CS1L,CS5L = 0x22220E01
    - WEA = 2, byte enables asserted 2 half-clocks after start of access
    - WEN = 2, byte enables negated 2 half-clocks before end of access
    - OEA, OEN = 2, similar for /OE on reads
    - CSA = 0, /CS asserted when write starts
    - CSN = 0, /CS negated when write ends
    - EBC = 1, byte enables during writes only
    - DSZ = 6, 32-bit bus width
    - CSEN = 1, /CS enabled
Bus Timing

- Default uMon settings

*minimum read cycle length = 7 HCLK *

*minimum write cycle length = 7 HCLK *

with DCT = 0, WSC = 4 xcvr disabled Dx is hi-Z during SNOOP if NFI04 jumper removed