DRM Class Project

EE382V-SoC
Fall 2009

Andreas Gerstlauer
Mark McDermott

Project Description

The class will be assigned to teams to do the various components of the design. The intent of the project is to do a HW/SW co-design of an embedded SOC. The design is a low power SOC implementation of the public domain DRM software implementation. We will use both a virtual platform (ARM) and a hardware platform (TLL5000-TLL6219) to simulate the design. These platforms consist of an ARM processor, I/O devices, memory components, hardware accelerators interconnected via a standard bus.
**Project Objectives and Activities**

- **The objectives of the project are as follows:**
  1. Implement the DRM C++ code on a ARM based platform while meeting the performance, area and power metrics.

- **The project activities include:**
  - Optimize the DRM C++ software for fixed point operation
  - Profile the updated DRM C++ software implementation to determine performance bottlenecks
  - Convert time critical functions to pseudo ANSI-C/SystemC code. Synthesize C code to Verilog for gate level implementation
  - Partition the software into components which will run on the ARM processor and on the hardware accelerators
  - Co-simulate and co-verify the HW/SW implementation
  - Develop a high level power model of the system

**DRM PC Based System Architecture**

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

DRM SW System Architecture
High Level HW Architecture

- USB 1.1
- RS232
- Ethernet
- ARM-9 Embedded Processor (iMX21)
- Flash Memory
- SDRAM Memory
- Buffer
- Configurable Logic (FPGA)
- SDRAM Memory
- TLL5000
- TLL6219

HW Development Tasks

- Develop the following FPGA modules:
  - Memory Controller
  - Interface to ARM Board and on chip bus
  - Hardware Accelerators (using Catapult code)
  - Clocking & Reset
  - Interrupt logic
  - Diagnostics
Software based tasks

- Compile FLP DRM and get it running on ARM board
  - Profile code on ARM board
- Convert FLP -> FXP in DRM code
  - Run SNR checks
  - Modify conversion as needed
- Compile FXP DRM and get it running on ARM board
  - Profile code on ARM board
- Develop streaming I/O handler
- Develop interrupt handler
- Develop HAL

TLL500 Prototyping Board
**Xilinx Spartan 3 FPGA**

- Spartan 3 XC3S1500 Discussion
  - Overview
  - Logic Resources
  - Memory Resources
  - Clock Resources
  - Input/Output Resources

### Spartan 3 Overview

<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>DCUs</th>
<th>Maximum User I/O</th>
<th>Maximum Differential I/O Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>XC3S00</td>
<td>50K</td>
<td>1,728</td>
<td>16</td>
<td>192</td>
<td>768</td>
<td>12K</td>
<td>72K</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>XC3S300</td>
<td>200K</td>
<td>4,320</td>
<td>24</td>
<td>480</td>
<td>1,920</td>
<td>30K</td>
<td>216K</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>XC3S600</td>
<td>600K</td>
<td>9,964</td>
<td>32</td>
<td>966</td>
<td>3,554</td>
<td>50K</td>
<td>256K</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>XC3S1000</td>
<td>1,000K</td>
<td>17,272</td>
<td>48</td>
<td>1,290</td>
<td>7,680</td>
<td>128K</td>
<td>832K</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>XC3S1300</td>
<td>1,500K</td>
<td>29,982</td>
<td>64</td>
<td>1,628</td>
<td>13,112</td>
<td>204K</td>
<td>1,078K</td>
<td>32</td>
<td>4</td>
</tr>
<tr>
<td>XC3S4000</td>
<td>4,000K</td>
<td>62,208</td>
<td>96</td>
<td>6,492</td>
<td>27,648</td>
<td>432K</td>
<td>1,728K</td>
<td>96</td>
<td>4</td>
</tr>
<tr>
<td>XC3S8000</td>
<td>8,000K</td>
<td>74,800</td>
<td>104</td>
<td>8,320</td>
<td>33,280</td>
<td>520K</td>
<td>3,072K</td>
<td>104</td>
<td>4</td>
</tr>
</tbody>
</table>

#### All Possible I/O Pins by Type

<table>
<thead>
<tr>
<th>Edge</th>
<th>IO Rank</th>
<th>Maximum I/O</th>
<th>I/O</th>
<th>DUAL</th>
<th>DCE</th>
<th>VBFF</th>
<th>GSFF</th>
<th>GCLK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top</td>
<td>0</td>
<td>02</td>
<td>52</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>01</td>
<td>51</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Right</td>
<td>2</td>
<td>00</td>
<td>52</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>00</td>
<td>52</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bottom</td>
<td>4</td>
<td>00</td>
<td>51</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>01</td>
<td>52</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Left</td>
<td>6</td>
<td>00</td>
<td>52</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>00</td>
<td>52</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Interconnect Matrix

Channels contain various connection resources

Channel Interconnects

- Horizontal and Vertical Long Lines (horizontal channel shown as an example)

- Horizontal and Vertical Short Lines (horizontal channel shown as an example)

- Horizontal and Vertical Double Lines (horizontal channel shown as an example)

- Direct Connections
CLBs & Slices

- Each slice has 2 LUTs & 2 storage elements

CLB Slice

- 2 FFs or latches
- 2 LUTs for combinational logic
- Some LUTs can be used as distributed RAM or ROM, or shift registers
- Carry look-ahead
- Dedicated muxes
Multipliers

- 2's-complement multipliers distributed across fabric
  - Primarily to support DSP
    - Cascade connections
  - Pipeline registers on inputs and outputs
  - Can do 2 small width op
- Can also be used for
  - Barrel shifters
  - Data storage
- Some shared connections with adjacent block RAM

Block RAM

There is no dedicated monitor to arbitrate the result of identical addresses on both ports. The application must time the two clocks appropriately. However, conflicting simultaneous writes to the same location never cause any physical damage.
Digital Clock Manager (DCM)

- **DCM functions**
  - Eliminate clock skew using Delay-Locked Loop (DLL)
    - Monitors clock skew on output and corrects
    - Performs frequency doubling
    - Creates multiphase clocks
  - Fractional Digital Frequency Synthesizer (DFS)
    - $f_{OUT} = M/N f_{IN}$
  - Clock conditioning
  - Clock buffering and signal translation
Digital Clock Manager (DCM)

- PSINCDEC
- PSEN
- PSCLK
- CLKN
- CLKFB
- RST
- Phase Shifter
- Delay Flaps
- Output Stage
- DLL
- DFS
- PSDONE
- CLKO
- CLK90
- CLK180
- CLK270
- CLK2X
- CLK2X180
- CLKDV
- Clock Distribution Delay
- Locked
- Status [7:0]

Input/Output Block (IOB)

- Slew rate and drive strength control
- Pull-up, pull-down and keeper
- DDR signals
- Controlled-Z input/output
- Boundary scan
## I/O Standards

### Table 7-8: Single-Ended I/O Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>V_{CC}</th>
<th>Drive Class</th>
<th>Spartan-3 FPGA</th>
<th>Spartan-6 FPGA</th>
<th>Spartan-20M/XA DSP FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVCMOS</td>
<td>1.5V</td>
<td>H</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>1.8V</td>
<td>H</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>2.5V</td>
<td>H</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>3.3V</td>
<td>H</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>5.0V</td>
<td>H</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>LVTL</td>
<td>3.3V</td>
<td>-</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>5.0V</td>
<td>-</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PC15</td>
<td>3.3V</td>
<td>-</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>5.0V</td>
<td>-</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>FDX</td>
<td>3.3V</td>
<td>-</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>5.0V</td>
<td>-</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SSTL</td>
<td>1.8V</td>
<td>I</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>2.5V</td>
<td>I</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

### Table 1-6: Differential I/O Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>V_{CC}</th>
<th>Spartan-3 FPGA</th>
<th>Spartan-6 FPGA</th>
<th>Spartan-20M/XA DSP FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVDS</td>
<td>3.3V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>5.0V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>BLDD</td>
<td>2.5V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>3.3V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>LVPECL</td>
<td>3.3V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>5.0V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>RS423</td>
<td>2.5V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>3.3V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>LVDS</td>
<td>2.5V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>3.3V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>LVDS</td>
<td>3.3V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>5.0V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>LVTTL</td>
<td>2.5V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>3.3V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>LVDS</td>
<td>3.3V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>5.0V</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

### TLL6219 ARM Processor Board

The TLL6219 ARM Processor Board is a development board designed for teaching and learning embedded systems, based on the TLL6219 ARM processor. It comes with a set of I/O standards and interfaces to facilitate various educational and project-based tasks. The board supports both single-ended and differential I/O configurations, making it versatile for different applications.
TLL6219 ARM Processor Board

**TLL6219 Block Diagram**

- USB 1.1
- RS232
- Ethernet
- JTAG Header
- ARM-9 Embedded Processor (iMX21)
- CPLD
- Data Buffers
- Address Buffers
- Mezzanine Connectors
- Expansion Port
- Flash Memory
- SDRAM Memory
- Control
- JTAG
- Control
- DATA
- ADDRESS
- DATA
- ADDRESS
- RS-232, GPIO
- DATA
- ADDRESS
i.MX21 Features

![i.MX21 Features Diagram]

Block Diagram

![Block Diagram]
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 MB</td>
<td>ROM, Primary AHB Slaves, and Peripherals</td>
</tr>
<tr>
<td>0x20000000</td>
<td>512 MB</td>
<td>Reserved</td>
</tr>
<tr>
<td>0x40000000</td>
<td>512 MB</td>
<td>Reserved</td>
</tr>
<tr>
<td>0x60000000</td>
<td>512 MB</td>
<td>Reserved</td>
</tr>
<tr>
<td>0x80000000</td>
<td>512 MB</td>
<td>Secondary AHB Slave Port 1</td>
</tr>
<tr>
<td>0xA0000000</td>
<td>512 MB</td>
<td>Secondary AHB Slave Port 2</td>
</tr>
<tr>
<td>0xC0000000</td>
<td>512 MB</td>
<td>Secondary AHB Slave Port 3</td>
</tr>
<tr>
<td>0xE0000000</td>
<td>512 MB</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 FPGA**

- All access from the ARM9 to hardware will be through the Chip Select 1 & 5 memory regions
- All TLL5000 peripherals must be accessed through the FPGA
- The only direct connection to the ARM9 is
  - QVGA LCD with touchscreen
  - RS-232
  - Ethernet

**System Block Diagram**

[Diagram showing the system block diagram with TLL5000, TLL5000 FPGA, QVGA LCD, TLL6219 ARM920E-J-S, Ethernet, RS-232, and other connections]
TLL6219 CPLD Connections

- CPLD_INT connects to PF[16]
- MISC[xxxxx] signal functions defined by CPLD
- /DTACK for cycle timing

TLL6219 CPLD Overview

- The CPLD generates read and write strobes for accesses in the /CS1 and /CS5 spaces (combinational logic)
  - cs1_rs_b = ~(~cs1_b & ~oe_b);
  - cs1_ws_b = ~(~cs1_b & ~(&eb) & ~rw_b);
  - cs5_rs_b = ~(~cs5_b & ~oe_b);
  - cs5_ws_b = ~(~cs5_b & ~(&eb) & ~rw_b);
- /DTACK is synchronized in the CPLD
  - Single flip-flop synchronizer
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 ARM926
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]

TLL6219 Transceiver Control

- The NFIO4 jumper is used to control data transceiver operation when the ARM926 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 that is not in the /CS1 or /CS5 spaces
iMX21 External Interface Module (EIM)

- The EIM permits very 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

Figure 77. DTACK Edge Triggered Read Access, WSC=3F, OEA=8, OEN=5, AGE=1.
Chip Select configuration set uMon

- 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 configuration set uMon

- 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

TLL5000 Block Diagram