ECE260C
Hardware-Software Verification

Sujit Dey
ECE, UCSD
Hardware-Software System Verification Challenges

• **Verification goals**
  – *functionality*, timing, performance, power, physical

• **Design complexity**
  – MPUs, MCUs, DSPs, interface, telecom, multimedia

• **Diversity of blocks (IPs/Cores)**
  – different vendors
  – soft, firm, hard
  – digital, analog, synchronous, asynchronous
  – different modeling and description languages - C, Verilog, VHDL
  – software, firmware, hardware

• **Different phases in system design flow**
  – specification validation, algorithmic, architectural, hw/sw,
    full timing, prototype
System Validation Techniques

- Hardware Simulation
- Prototype Validation: Emulation, ICE, Debug Monitor
- System Verification Environments
- Simulation Models
- Instruction-Set Simulation
- Hardware-Software Co-Simulation
Prototype Validation

Firmware Design
- Algorithm
- Source Code
- Object Code

Hardware Design
- Behavior
- RTL
- GATE

Integration Test
- In Circuit Emulator (ICE)
Emulation

Emulation: Imitation of all or parts of the target system by another system, the target system performance achieved primarily by hardware implementation

In-Circuit Emulator (ICE): A box of hardware that can emulate the processor in the target system. The ICE can execute code in the target system’s memory or a code that is down loaded to emulator.

- ICE also can be fabricated as silicon within the processor-core: provides interface between a source level debugger and a processor embedded within an ASIC.
  - Provides Realtime emulation.
  - Functions supported such as Breakpoint setting, Single step execution, Trace display and Performance analysis.
  - Provide C-source debugger.

Example: embeddedICE macrocell in ARM7TDM1
Debugging environment for CPU core

In-circuit Emulator

Create by using standard LSI, FPGA & G/A

Bread board for emulator

Target system

Source: NECEL
Embedded ICE Macrocell

- ARM Core
- EmbeddedICE Macrocell
- Data bus scan chain
- Control
- Data ARM7TDM Addr
- TAP
- 5 pin JTAG Interface

Source: ARM
Embedded ICE in ARM7TDMI Core
Problem of Validation after HW/SW Completion

Typical embedded system project schedule

<table>
<thead>
<tr>
<th>System Design</th>
<th>Hardware Design</th>
<th>Prototype Build</th>
<th>Hardware Debug</th>
<th>Software Design</th>
<th>Software Coding</th>
<th>Software Debug</th>
<th>Project Complete</th>
</tr>
</thead>
</table>

Diagram illustrating the typical embedded system project schedule.
Need to move Validation Early in the Design Cycle
Simulation Models

• Bus-functional model
• Instruction-Set simulation (ISS) model
  – instruction accurate
  – cycle accurate
• Full-timing gate-level model
  – encrypted to protect IP
Bus Functional Model

- Idea is to remove the application code and the target processor from the hardware simulation environment
- Performance gains by using the host processor’s capabilities rather than simulating same operation happening on target processor
- Varying degrees of use of host processor leads to different models
- Bus functional model
  - only models the interface circuitry (bus), no internal functionality
  - usually driven by commands, like read, write, interrupt, ..
  - bus-transaction commands converted into a timed sequence of signal transitions: fed as events to traditional hardware simulator
- **Bus Functional model emulates:**
  - Read/Write Cycles (single/burst transfers)
  - Interrupts
Compiled Code Simulation

- Host code not equal to Target code
- Low-level debugging not possible
  - *eg.* observing processor internal registers
- Measurements may be inaccurate
  - *eg.* cycle counts
Instruction Set Simulation

• Full functional accuracy of the processor as viewed from pins
• Operations of CPU modeled at the register/instruction level
  – registers as program variables
  – instructions by program functions which operate on register values
• Instructions define relationships between registers, internal memory, and external memory
• Data Path that connects the registers abstracted out
• Allows both high level and assembly code to be debugged
• Instruction Accurate
  – accurate at instruction boundaries only
  – correct bus operations, and total number of cycles, but no guarantee of state of CPU at each clock cycle; inaccuracy due to bus contention
• Cycle Accurate
  – guarantees the state of the CPU at every clock cycle
  – guarantees exact bus behavior
  – slower than instruction-accurate
Instruction-Set Simulation: Example

- Example system: Questa Codelink™
- ISS model 100,000 times faster than RTL models
- software debug: source code debugging, register and memory views
Hardware-Software Co-Simulation

- Most of the bus cycles are Instruction or Data fetches

- High Activity
  - 700-1000 instructions for each I/O bus cycle

- Low Activity
  - Only during processor I/O cycles
Hardware-Software Co-Simulation: Implementation

Application Program (Assembly) → ISS Processor model → Host memory → HDL model

- Bus functional model
- Memory model

Memory & Signal Synchronization
**Questa Codelink CVE™: Comprehensive System Wide Analysis & Debug**

Source: Mentor Graphics
### Comparison of Validation Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Proc.</th>
<th>UDL</th>
<th>Silicon</th>
<th>Cost</th>
<th>Debug Cap.</th>
<th>Speed</th>
<th>Timing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HW Prototype</strong></td>
<td>Real</td>
<td>Real</td>
<td>Yes</td>
<td>High</td>
<td>Low</td>
<td>Fast</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>ICE</strong></td>
<td>Proc. w/ICE</td>
<td>Real</td>
<td>Yes</td>
<td>High</td>
<td>Med.</td>
<td>Fast</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>HW Emulator</strong></td>
<td>Real</td>
<td>FPGAs</td>
<td>No</td>
<td>High</td>
<td>Low</td>
<td>Fast</td>
<td>Yes (but FPGAs)</td>
</tr>
<tr>
<td><strong>HW Simulator</strong></td>
<td>HDL mode (timing)</td>
<td>HDL</td>
<td>No</td>
<td>Low</td>
<td>High</td>
<td>Very Slow</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>HW model</strong></td>
<td>HDL</td>
<td>No</td>
<td>Med.</td>
<td>High</td>
<td>Slow</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td><strong>Bus functional</strong></td>
<td>HDL</td>
<td>No</td>
<td>Low</td>
<td>Only HW</td>
<td>Med.</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td><strong>ISS</strong></td>
<td>ISS model</td>
<td>HDL</td>
<td>No</td>
<td>Low</td>
<td>Only SW</td>
<td>Med.</td>
<td>Cycle accurate</td>
</tr>
<tr>
<td><strong>Co-simulation</strong></td>
<td>ISS model</td>
<td>HDL</td>
<td>No</td>
<td>Med.</td>
<td>High</td>
<td>Med.</td>
<td>Cycle accurate</td>
</tr>
</tbody>
</table>