Prototyping and Emulation

Siamak Mohammadi
University of Tehran

Overview

- Introduction
- Prototyping and Emulation Techniques
- Prototyping and Emulation Environment
- Ref: Chapter 3 of HW/SW Codesign

Introduction

- Issues:
  - Increasing complexity of ASICs
  - Errors of Specification, design and implementation
- Need for validation methods and tools for “first-time-right-silicon”
  - High level abstraction specification and automatic synthesis
  - Simulation
  - Formal verification
  - Prototyping and emulation

High Level Synthesis Advantages

- High level synthesis is very important for HW/SW Codesign
- Investigating different HW partitions must be done as fast as SW compilation into machine code
  - High level synthesis is therefore the enabling technology for HW/SW Codesign
- High level of abstraction made possible → shorter formal specification time
- Supports investigation of many different design alternatives → higher quality designs
Simulation?
- Simulation is the standard and most common way of validating designs
  - But it could be very time consuming and slow requiring an explicit testbench
- Development of new systems requires simulation of both HW and SW and of peripheral devices
  - Difficult to integrate interface components
  - Slow cycle-based simulation compared to real-time execution
- Formal verification is an alternative but is limited in many ways
  - High complexity
  - Not suitable for time-related constraints

Prototyping and Emulation?
- Two problems for validation:
  - Appropriate means for HW/SW validation needed
  - These means must be combined for integrated system validation
- Software validation methods are well-known
- More efforts on HW validation methods
- For HW/SW codesign:
  - Low-level validation not useful
    - Like writing C++ code and debugging the generated assembly code
  - HW/SW implies high-level synthesis for HW parts

Prototyping and Emulation?
- Great advantage of P&E is higher speed
  - Only 100 times slower than real time
  - Easier integration of the environment
- Main disadvantage of emulation is that timing errors are hard or impossible to detect
  - Slow compilation once the circuit changes
  - High expense
  - Mostly serves as functional testing

P & E Techniques
- HDL-ICE System allows relating the probed signals to a register-transfer specification
  - Quickturn
- Cycle-based emulation of an RT-level design
  - Synopsys: Arkos system
  - Quickturn: Cobalt system
- Classical gate level emulation techniques based on FPGA
  - Mentor
- Special user programmable switches for interconnect
P & E Techniques
- Specially developed VLSI custom switch circuits
- Processor based emulator with an instruction set whose instructions are optimized for execution of logic operators
  - Circuit design is translated into Boolean equations and compiled into machine code for processors

P & E Environments
- Many approaches for prototyping of hardware but not enough for embedded systems
  - Difficult for the emulator’s synthesis SW to route buses introduced by the integration of µP
  - Hardware partitioning too inefficient leading to inadequate emulator’s HW resource utilization
  - Debugging facilities not sufficient
- Recent FPGAs with soft CPUs permits better emulation of HW/SW codesign

Principle of Emulation
- Logic design is mapped on an emulation board with multiple FPGAs
- Different interconnection schemes exist
  - Programmable gate arrays for interconnection structure
  - Special interconnection switches
  - Custom interconnect chips

Prototyping and Emulation
What is the distinction between P & E?
- The dividing line is blurred
- Prototyping:
  - Refers to a special architecture cut to fit a specific application
  - Often FPGA-based
  - Achieves higher clock rate due to specific cutting
    - Best elements selected for µcontroller, RAMs, special peripherals
  - New expenditure for the development of such a special prototype occurs for every new prototype
**P & E Systems**

- Some systems provide a compromise solution between general purpose emulation systems and special prototypes
  - Use hardwired regular interconnection scheme
    - Fewer signals routed through programmable devices
    - Better performance
  - Bus module provided for interconnecting modules
  - Debugger with tracing signals at run time
  - Configure FPGA at run time without affecting other parts of the architecture
  - OS with mini Kernel to run software

**Future Developments in P & E**

- Design synthesis from higher level of abstractions are enabled due to advances in design automation
- Interactive debugging becomes problematic
  - Requires mapping between the description and the emulation environment to reconstruct the transformations implemented by synthesis system
  - SLE: source level emulation: closes gap between description and hardware emulation
    - Retain relation between hardware elements and the source program (e.g. VHDL)

---

**Steps in High Level Synthesis**

1. Algorithm
2. Dataflowgraph
3. DFG annotated with a set of HW-component types
4. Operations mapped to a HW components type and a time slot
5. Structural circuit comprising HW-component, regs, muxes and a controller
6. Controller-Synthesis, Assignment