System-on-Chip: An EDA Perspective

CSE 291E / ECE 260C
Spring 2002
SoC Design

- e.g. ARM core and AMBA bus

- Analogous to Pentium, Windows/Linux, PCI, memory, graphics card, sound card, Ethernet card ...
- Need standardized bus & driver models (e.g. AMBA bus instead of PCI)
- Need stripped down OS (e.g. WindRiver instead of Windows/Linux) to host software and “drivers” (communication channel) to hardware
Current SoC Interconnect Approaches

- Standardized busses

- Point-to-point wires

- Shared FIFO’s or shared register files

Diagram:
- Standardized busses
- Point-to-point wires
- Shared FIFO’s or shared register files
EDA for SoC Design: Co-Simulation

- Co-simulation = Cycle-true instruction-set simulators + cycle-based Verilog simulators
- Can monitor registers & stack values on processor side
- Can monitor "wire" values on the hardware/HDL side
- Bus models & other macro models available in Verilog
EDA for SoC Design: Design and Synthesis

- WindRiver real-time OS to multi-thread embedded software apps
- Drive API on software side: enable programmers to connect to hardware w/o knowing gory details of bus protocol
- “PCI-like controller logic” in Verilog as “stub-code” on Verilog side: enable Verilog designers to “connect-to-the-bus” w/o knowing gory details of bus protocol
EDA for SoC Design: Design and Synthesis

• Higher level abstractions viewed design as communicating processes
  • e.g. socket/queue-pair style communication abstractions (analogous to MPI/PVM)
  • C-API on software side, stub verilog code on hardware side
  • High level modeling and synthesis tools map to multiprocessor architecture using shared busses, point-to-point wires, and shared buffers

• So far, approach most successful for DSP and digital communications applications
EDA for SoC Design: Design and Synthesis

• Verilog/VHDL
  • Models hardware at the clock-cycle to clock-cycle level (called RTL = register transfer level)

• SystemC
  • C syntax
  • Minus memory and pointer references
  • Minus generic function calls (inline expansion only)
  • Synthesize to RTL Verilog
EDA for SoC Design: Design and Synthesis

• SystemC (cont’d)
  • Synthesis process similar to code generation for Tensilica (Tensilica borrowed many ideas from EDA and vice versa)
  • Analyzes code for ILP (instruction-level parallelism) and “allocates” ALU’s, registers ... to exploit ILP
  • Generates accelerator datapaths (with user “hints”)
  • Usually maps resulting code sequence into a hard coded FSM (rather than a micro-sequencer that needs to fetch instructions)
  • Based on research from “high-level synthesis”
• C syntax enables “syntactical” unification of HW/SW co-design
Questions?