

Co-Design

- Goals – Why do it?
 - Accelerate SW OS/Driver development
 - Increase HW confidence pre-silicon
 - Simplify HW/SW integration
 - Optimize schedule and resources

Co-Design Approaches

- Multiple approaches exist for Co-design
 - Fully custom, stand-alone environment
 - Subset of the full DV environment
 - Integrated, modular verification environment
 - Others...

Modular DV Environment

- Advantages
 - Abstraction level can “float” to suit application
 - Can be SW or HW-centric, depending on level
 - Can start out with abstract models, refine to bit-accuracy as project progresses/design solidifies
 - No significant development hit to schedule if planned up-front.

SystemC as a solution

- SystemC lends itself to creating integrated, modular simulation/verification environments
- Easy to use for both SW and HW
- Can run independently of HW simulator if no RTL involved in a given session (saves money in license cost)
- Standard debug/code checking tools can be used – gdb, purify etc...

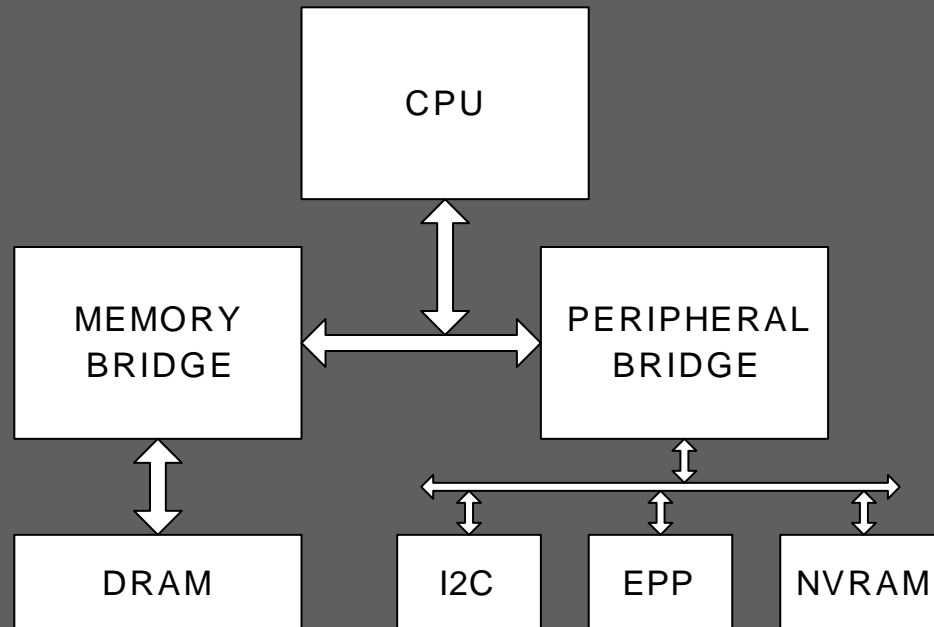
Abstraction

- Ability to declare a given section of a design at a particular level of abstraction is key to allowing multiple uses and to solving multiple problems with a single HW development environment.

Example Design Flow

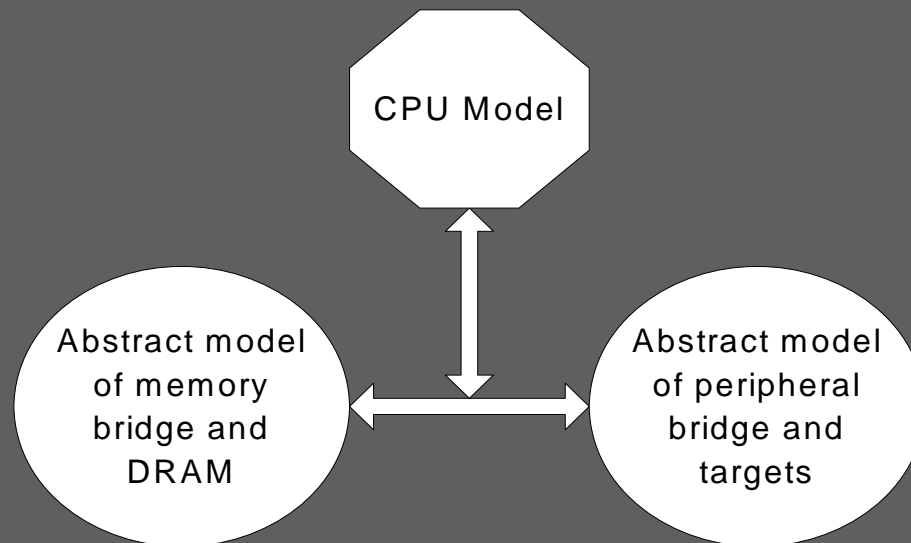
- We will consider a generic system design with the following objectives:
 - Illustrate the value of co-design at different points in the design flow
 - Illustrate the value of modular system design/verification infrastructure

Example System



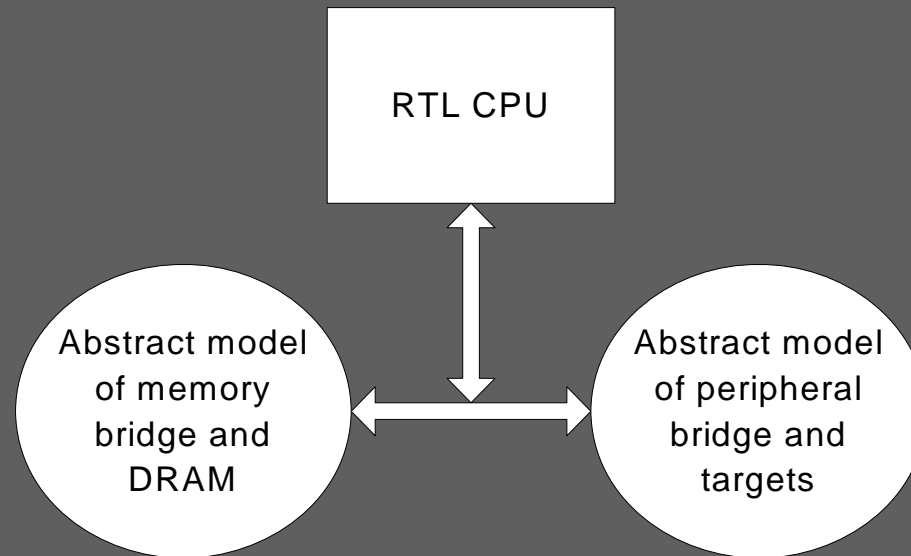
Example System (cont...)

- Architectural



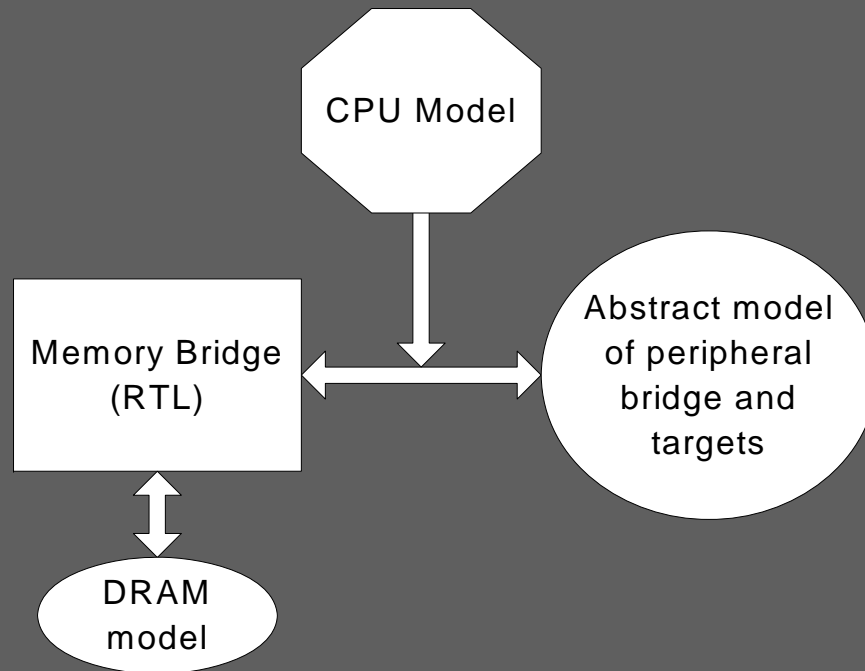
Example System (cont...)

- RTL CPU, abstract peripherals



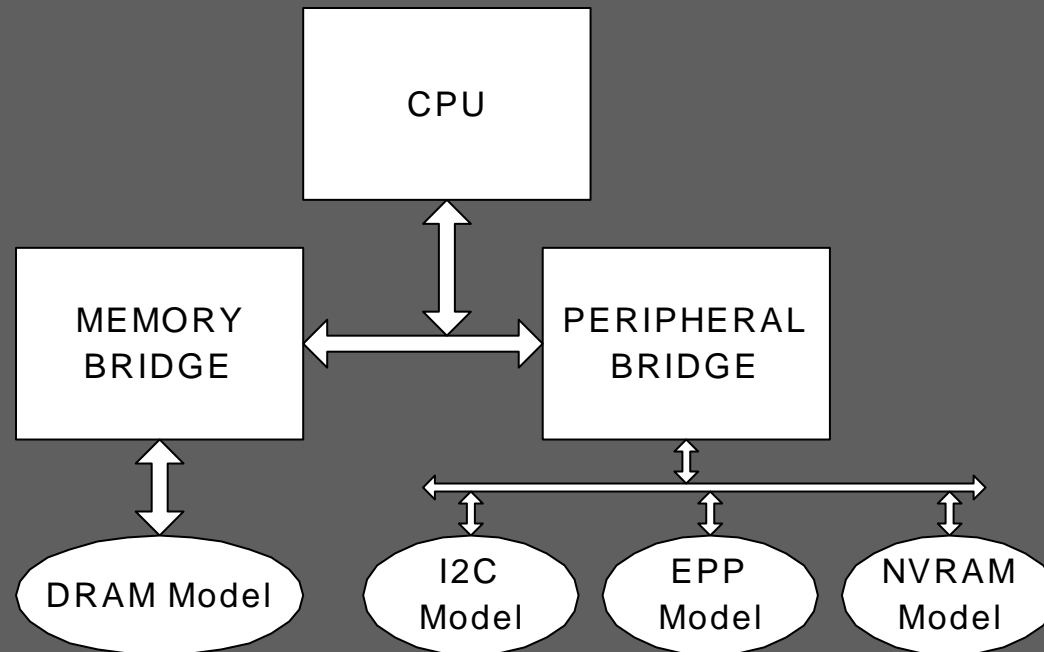
Example System (cont...)

- RTL memory bridge, abstract peripherals



Example System (cont...)

- RTL system, abstract peripheral models



Summary

- Modular, unified SystemC environment brings the following benefits:
 - Architecture confidence earlier
 - RTL tested against real SW sequences
 - DV test suite started before RTL complete
 - SW sequences can port back from lab to DV
 - Drivers/OS proven out before power-on

Recommendations

- Modular, SystemC-based environments can offer significant advantages to a project.
- EDA Companies need to allow for flexibility on the part of the user – not “one size fits all”
- Consider adding utility suite to simulators
 - Messaging, data manipulation, event handling, etc...

BACKUP SLIDES

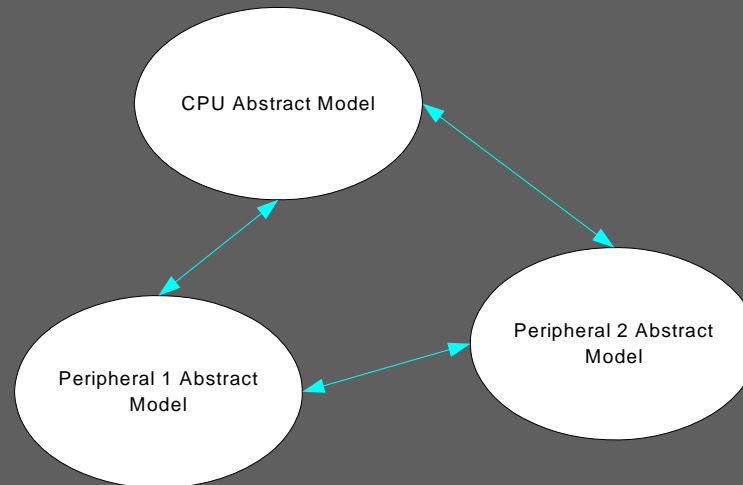
--

Modularity

- SystemC can easily mimic HW-style modularity
- Still retains the ability to collapse structure into abstract behavioral functions

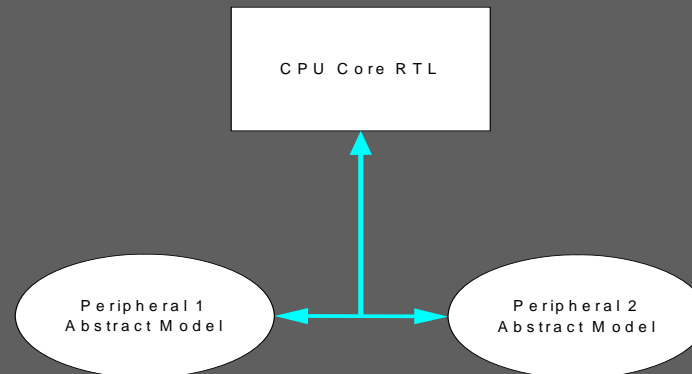
Example Environment

- Totally abstract architectural environment



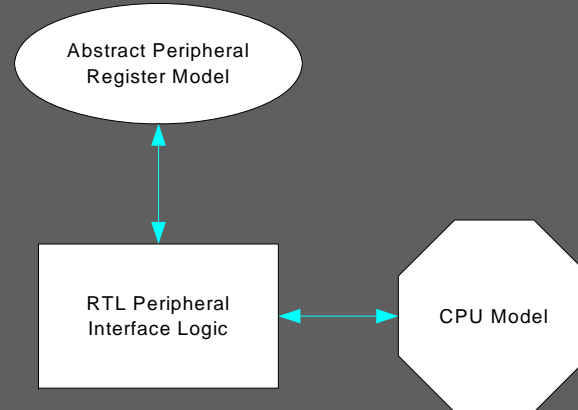
Example Environment (cont...)

- RTL CPU core, abstract SoC



Example Environment (Cont...)

- Abstract core, RTL module, abstract peripherals



Fully Custom Environment

- Pros:
 - Optimal speed (pure SW models)
 - Often most abstract system model
 - SW-friendly
- Cons:
 - Inaccurate representation – details get missed
 - Hard to integrate with HW/DV

Subset of DV Environment

- Pros:
 - Part of DV/design flow – lower schedule hit
 - Accurate (RTL representation of modules)
 - HW-friendly
- Cons:
 - Slow (RTL/gate-level simulation)
 - Can be gated by RTL availability
 - Require HDL simulator + license