Research Accelerator for Multiple Processors

Architecture, Language and Compiler

Presented 9/28/2006
GSRC Symposium,
San Jose, CA

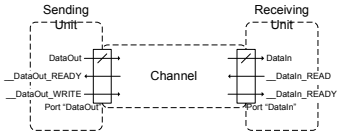
Greg Gibeling (gdgib@berkeley.edu)
http://ramp.eecs.berkeley.edu
Thanks to Krste Asanovic, John Wawrzynek,
Andrew Schultz and Nathan Burkhart

Introducing RAMP

- **Research Accelerator for Multiple Processors.**
  - Originally envisioned as a cross platform architectural simulator.
  - Designed to foster community research (Share & verify results).
- **A distributed event simulation & message passing system framework.**
  - Orders of magnitude faster than existing solutions.
  - Eases component re-use and integration.
- **A modeling language (RDL) is a key step in the realization of RAMP.**
  - The "Target System," the system being emulated, is captured in RDL and emulated on the "Host System" (an FPGA or CPU).
  - Allows virtualization of simulation time to provide cycle accurate simulation results.

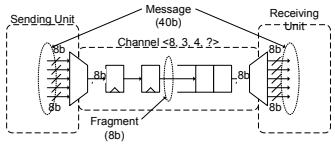
The "Target" Model

Basic "Target" Model



Hardware Signals for Unit Ports

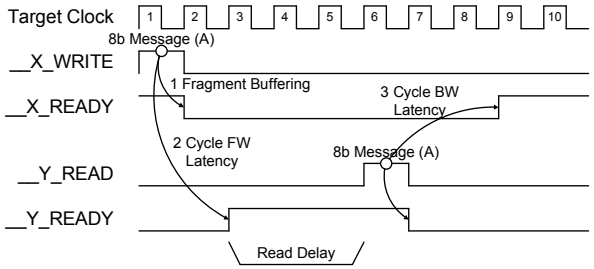
- **Units communicate over point-to-point, unidirectional channels**
  - A unit would be ~10,000 gates (Processor + L1 cache)
  - Units are implemented in the "host" language, eg. Verilog
  - Existing message passing hardware/software can be ported easily
- **Channels include a delay model**
  - Allows timing simulations
  - Staticly typed, variable size messages
  - Bitwidth (Fragment)
  - Latency
  - Buffering



Message Fragmentation

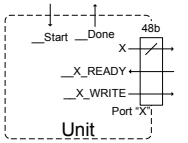
Channel Timing Parameters

- **Bitwidth**
  - Size of fragments
  - Bits/Fragment
- **FW Latency**
  - Latency for data
  - Target Cycles
- **Buffering**
  - Channel capacity
  - # of Frements
- **BW Latency**
  - Latency for ACK
  - Target Cycles



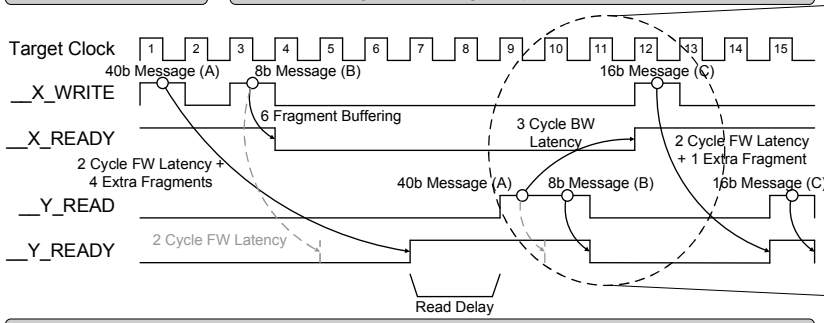
Target Level Timing Example 1 - Channel <8, 2, 1, 3>

Virtualization of Time

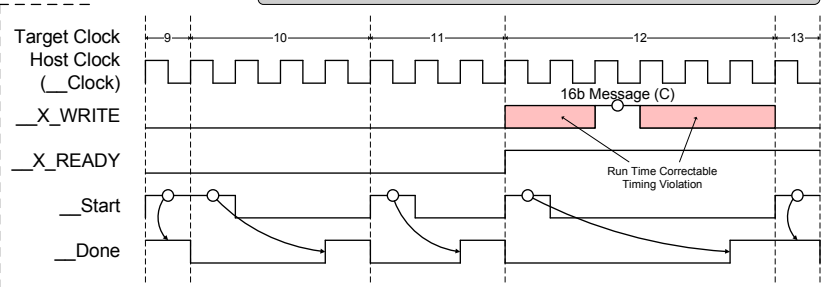


Sending Unit

- **Start & Done**
  - Regulate the length of a target cycle, virtual, with respect to the host, physical, clock. Start is triggered according to distributed event simulation rules (previous unit completed a target cycle). Done is triggered by the unit completing a target cycle worth of computation
- **Simulation of Timing & Latency**
  - Automatic "token" message fragments could be used to indicate of a target cycle. Injection of such "tokens" at startup leads to simulation of latency. Such tokens would never be visible to units.
  - This is merely an example, not a specification.



Target Level Timing Example 2 - Channel <8, 2, 6, 3>



Host Level Timing Example