Research Accelerator for Multiple Processors

Architecture, Language & Compiler

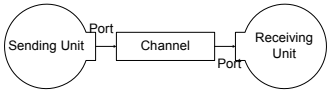
RAMP Archiecture, Language & Compiler
Presented 1/10/2006
BWRC Winter Retreat, Monterrey, CA

Greg Gibeling (gdgib@berkeley.edu)
http://ramp.eecs.berkeley.edu
Special thanks to the entire
UC Berkeley RAMP Gateware Group

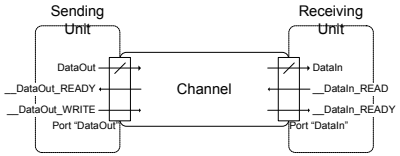
Introducing RAMP

- **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)**

The "Target System" Model

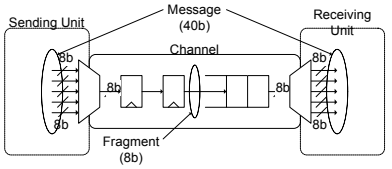


Basic "Target System" Model



Hardware Signals for Unit Ports

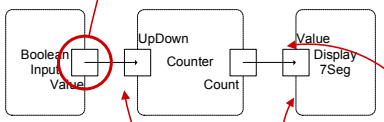
- **Units communicate over point-to-point, unidirectional channels**
  - **A unit would be ~10,000 gates (Processor + L1 cache)**
  - **Units are implemented in the "host" language, eg. Verilog**
  - **Existing message passing hardware/software can be ported easily**
- **Channels include a delay model**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



Message Fragmentation

The "Counter" Example

```
unit {
   input     bit[1]              UpDown;
   output    bit[32]             Count;
}                                Counter;

unit {
   output    bit[1]              Value;
   attribute CaLinx2             "input:1:SW_";
   attribute CaLinx2             "input:1:BTN_";
}                                IO::BooleanInput;
   Platform specific attributes
   (For external connections)
platform {
   language                      "verilog";
   default link                  "SerialLink";
}                                CaLinx2;

map {
   platform  CaLinx2             BasePlatformInst;
   unit      CounterExample      BaseUnitInst;
}                                CaLinx2Counter;
```



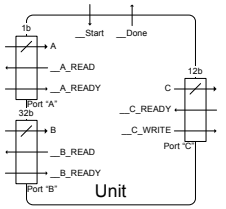

```
unit {
   instance  IO::BooleanInput    BooleanInput;
   instance  Counter             Counter;
   instance  IO::Display7Seg     Display7Seg;

   channel   fifopipe[1, 1, 1]   InChannel
     { BooleanInput.Value -> Counter.UpDown };
   channel   fifopipe[32, 1, 1]  OutChannel
     { Counter.Count -> Display7Seg.Value };
}                                CounterExample;
```

RDL Compiler Toolflow

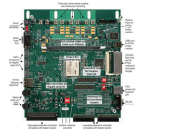
- **A simple unit**
  - **Three ports**
  - **Unstructured messages**
- **No functionality in RDL**

```
unit {
   input     bit[1]    A;
   input     bit[32]   B;
   output    bit[12]   C;
}                      Unit;
```

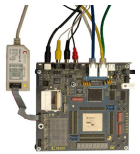
RDLC (Shell)



- **Unit shell, ready for implementation**
- **Verilog, Java, etc...**



Xilinx XUP

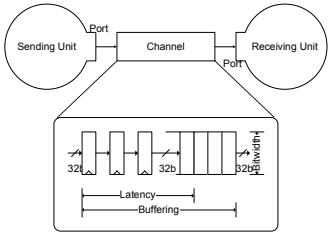


Berkeley CaLinx2

Standard Compiler

Any computer

- **RDL Hierarchical Netlist**
- **Include channel model**

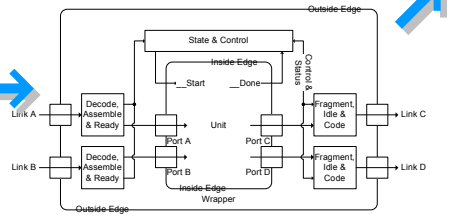


- **Flexible, extensible back end**
- **Support for multiple target languages**
- **Automatic cross platform implementation**

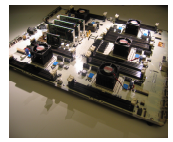
RDLC (Map)

<XML> Config

.java Plugins



- **Complete host implementation**



BEE2