Bluespec Runtime Debugging
New Capabilities and Future Direction

August 13, 2007
Bluespec Runtime Debugging - Goals

- **Transaction-Level Debugging View**
  - Match transaction-level design view
  - Provide symbolic representations of signals/data structures
    - Allow viewing through $display
    - Allow viewing as waves

- **BSV Source-Level Debugging View**
  - Debugging view relates signals to BSV source code
  - Provides view of elaborated BSV Hierarchy
  - Rules Are Special

- **Simulator/Flow Agnostic**
  - Works with Bluesim
  - Works with Verilog Simulation
Added Bluespec Features/Capabilities: **Fmt** Primitive Type (Beta, 08-2007 Release)

- Represents list of arguments to `$display` family of system tasks
- Created with new `$format` BSV system task
  - Takes same arguments as `$display` etc
  - Returns object of type **Fmt**
- Can be concatenated
- Valid argument to `$display` family of system tasks (including itself)
- Ideal building block for dynamic *show* class
  - Generates **ASCII** representation of objects at runtime
Added Bluespec Features/Capabilities:

**Fmt** Primitive Type (Beta, 08-2007 Release)

```verbatim
function Fmt fshow (TLMResponse#(`TLM_TYPES) op);
  return ($format("<TRESP [%2d] ", op.transaction_id)
          +
          fshow(op.status)
          +
          $fomat(" %8h>", op.data));
endfunction

rule show Resp;
  resp0 <- rx.get;
  $display("[%5d] Response is: ", $time, fshow(resp0));
endrule
```
Symbolic $display$ output created using Fmt Primitive Type

> [ 15] Request is: <TDESC [ 0] WRITE STREAM ( 5) A:6306b97b D:0d46df99>
> [ 25] Request is: <TDATA [ 0] 18484d60>
> [ 35] Request is: <TDATA [ 0] 30573870>
> [ 35] Response is: <TRESP [ 0] SUCCESS 0000000d>
> [ 45] Request is: <TDATA [ 0] da9a7d65>
> [ 45] Response is: <TRESP [ 0] SUCCESS 0000000d>
> [ 55] Request is: <TDATA [ 0] 30509650>
> [ 55] Response is: <TRESP [ 0] SUCCESS 00000000>
> [ 65] Response is: <TRESP [ 0] SUCCESS 0000000d>
> [ 65] Response is: <TRESP [ 0] SUCCESS 00000000>
>
Added Bluespec Features/Capabilities: $swrite family of system tasks (08-2007 Release)

- Part Of Verilog 2001 LRM
- Analogous to $fwrite system tasks
  - Except ASCII output is now written to variable or state element
  - In BSV a function call of type Action
  - ActionValue versions available as well
- Allows viewing of Fmt objects as waves

```verilog
case #show_resp;
  when #endrule
  resp0 <- ifc0.get;
  let ascii0 <- $swriteAV(fshow(resp0));
endrule
```
Symbolic waveforms created using **Fmt** primitive type and **$swrite**
Added Bluespec Features/Capabilities: BSV Source-Level Debugging View (Beta, 4Q 2007)

- Generates Navigation Tree Based on BSV Hierarchy
  - Both Synthesized and Non-Synthesized Modules
  - Includes Rules in hierarchy as well
- Hyper-Links To BSV Source Code
  - Modules Instantiations/Rules
- Rule Firing Visualization
- Waveform Viewer Integration
  - Synthesized Module Ports
  - Rule WILL_FIRE/CAN_FIRE Signals
- Works For Both Bluesim and Verilog Flows
  - Links to Verilog For Verilog Flow
BSV Source Code Debugging View:
Navigation Tree and BSV Source

```verilog
gerule every_count <= count + 1;
if (count == 1000) $finish;
endrule
endmodule

(* synthesize *)
module mkSource#(Maybe#(TLMCommand) m_command)
(TLMSendIFC#(TLM_STD_TYPES))

Reg#(Bool) initialized <- mkReg(False);
FIFO#(TLMResponseStd) response_fifo <- mkFIFO;
Randomize#(TLMRequestStd) gen <- mkTLMRandomizer(m_command);
Probe#(TLMRequestStd) request <- dbgProbe;
Probe#(TLMResponseStd) response <- dbgProbe;

rule start (!initialized);
gen.cntrl.init;
initialized <= True;
endrule

rule grab_responses;
let value = response_fifo.first;
response_fifo.deq;
```
Added Bluespec Features/Capabilities: Automatic Derivation Of Dynamic Show Class

- Automatic Derivation of **Fmt** Representation
  - Implemented as **Format** typeclass
- Allows User To Write Their own As Well
- Eliminates Need For Special **provisos**

```haskell
typedef Format#(type t) function Fmt format(t value);
endtypeclass
```
Bluespec Runtime Debugging Enhancements Roadmap

- Add $swrite family of system tasks
  - 08-2007 Release
- Add Fmt primitive type
  - Beta – 08/2007 Release
  - 4Q 2007
- BSV Source-Level Debugging View
  - Beta - 4Q 2007
  - Integrated with Bluespec Workstation – 1Q 2008
- Automatic Deriving Of Dynamic Show Class
  - 4Q 2007