Preview of Bluespec Debug Workstation

Ed Czeck
13 August 2007
Overview

Bluespec Workstation integrating development and debug with Bluespec SystemVerilog

Features:
- Single environment for compile, debug, simulate
- Supports standard tools (editors, simulator, waves)
- Adds Bluespec specific analysis

Releases:
- Beta, Fall 2007
- General, Q1 2008
Bluespec Analysis Features

- Provide source level view of high-level data
  - Details hierarchical view of advanced types (structures, unions, enumeration, interfaces)
  - Maps complex types into Verilog bits
  - Links structures to wave viewing
- Provides graphical tools to analyze and verify rule schedule and method use
- Allows users to develop scripts for custom analyses or reports
- Beta, Fall 2007
Workstation Architecture

- bsv Source
- IP & Libraries
- Packages
- Modules
- Verilog
- Sim

Analysis Windows:
- Package Browser
- Type Browser
- Module Browser
- Schedule Analysis
- Reports
Generating Verilog...
/workspace/hambersxum/Bluespec-2007.07.18_betawish/lib/bin/bsc -u -verilog -g mkTestAx
checking package dependencies
compiling ./Randomizable.bsv

#include 'TLM_defines'

module mkAxIrSlaveStd#(function Bool addr_match(AxiAddr偶(TMSTD_TYPES) addr)
(AxiIrSlaveXActorIFC偶(TMSTD_TYPES));
let _ifc <- mkAxIrSlaveSynthStd;

interface TLMSendIFC tlm = _ifc tlm;
interface AxIrSlave bus = addrAddrMatch(addr_match, _ifc.bus);
endmodule

module mkAxIrSlaveSynthStd (AxIrSlaveXActorIFC偶(TMSTD_TYPES));

29,1
Package Browser

- Pre-elaboration view of design
- View package contents including:
  - Interfaces & types with type parameters
  - Modules & functions with argument type
- Links to source
Package Browser
Type Browser

- Interactive type analysis
- Show details of any type
  - polymorphic or concrete
  - recursively expand interfaces, structs and unions
  - show fields and mapping to Verilog bits

Properties of selected type
- Definition
- Encoding to Verilog bits
- Source
Module Browser

- Post-Elaboration Viewer
- Show module hierarchy and composition
  - instance name, types, and doc
  - rule predicates and method calls

<table>
<thead>
<tr>
<th>Module</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td>mkArbiter</td>
<td></td>
</tr>
<tr>
<td>mkAxiDefines</td>
<td></td>
</tr>
<tr>
<td>mkAxiDMA</td>
<td></td>
</tr>
<tr>
<td>mkAxiMaster</td>
<td></td>
</tr>
<tr>
<td>mkAxiRam</td>
<td></td>
</tr>
<tr>
<td>mkAxiRdBus</td>
<td></td>
</tr>
<tr>
<td>mkAxiSlave</td>
<td></td>
</tr>
<tr>
<td>mkAxiWrBus</td>
<td></td>
</tr>
<tr>
<td>mkBusSwitch</td>
<td></td>
</tr>
<tr>
<td>mkBusUtils</td>
<td></td>
</tr>
<tr>
<td>mkCBus</td>
<td></td>
</tr>
<tr>
<td>mkControl</td>
<td></td>
</tr>
<tr>
<td>mkDMAConfigRegs</td>
<td></td>
</tr>
<tr>
<td>mkDMADefines</td>
<td></td>
</tr>
</tbody>
</table>

Module contents
- Interface
- Instances
- rules

Detailed view
- Types
- Doc string
- Rule predicates
- Source
Schedule Analysis

- View and Analyze Schedule
  - Rule execution order
  - Instance/method call use table
  - Blocking rule
  - Messages from scheduling
  - Relation between any pairs of rule

- Possible to add custom report
Schedule Analysis

Warning: TestSupport.bsv, line 65, column 17: (G0023)
    The condition for rule fsm_update_from_action
    Removing...
Warning: TestSupport.bsv, line 65, column 48: (G0023)
    The condition for rule fsm_update_from_action
    false. Removing...
Phase II Features

- Extend to Dynamic (simulation time) analysis
- Tight integration with Bluesim simulation: e.g. rule execution, rule condition analysis, interactive control, breakpoints, etc
- Design analysis, e.g., design structure, flop count, path delay estimates
- Others
- Scheduled Spring 2008
Feedback

Would love to hear suggestions and feedback:

Ed Czeck
support@bluespec.com