Reusable UVM_REG Backdoor Automation

Balasubramanian G, Design Services, PMC-Sierra India Pvt Ltd.
Agenda

• Verification challenges
• What is IP-XACT?
• Limitations of IP-XACT and vendor tool automation.
• Custom Automation Solution
• Testing and Statistics
• Summary
Verification challenges

TYPICAL VERIFICATION ENVIRONMENT

AXI/APB uVC (Master) ➔ regmodel

UVM_REG sequence ➔ Frontdoor

AXI/APB Slave Interface

bit1 ➔ bit2 ➔ bit3 ➔ … ➔ bitn ➔ RTL

BACKDOOR
BACKDOOR
BACKDOOR
BACKDOOR

© Accellera Systems Initiative
IP-XACT and Register definitions

- Programmable registers are captured in IP-XACT XML format.
- irezGen vendor toolset parses the IP-XACT XML and generates UVM_REG register definitions.
- Backdoor access forces or probes a RTL net directly.

IPXACT XML

```xml
<spirit:register>
  <spirit:name>SAMPLE_REG_1</spirit:name>
  <spirit:addressOffset>0x0</spirit:addressOffset>
  <spirit:size>32</spirit:size>
  <spirit:field>
    <spirit:name>MULTIBITRW0</spirit:name>
    <spirit:bitOffset>27</spirit:bitOffset>
    <spirit:bitWidth>2</spirit:bitWidth>
    <spirit:access>read-write</spirit:access>
  </spirit:field>
</spirit:register>
```

UVM REGISTER

```cpp
class SAMPLE_REG_1 extends uvm_reg;
  rand uvm_reg_field MULTIBITRW0;
  virtual function void build();
      MULTIBITRW0 = uvm_reg_field::type_id::create("MULRW0");
      MULTIBITRW0.configure(this, 2, 27, "RW", 0, 0);
  endfunction
endclass
```
IP-XACT and vendor tool limitations

- RTL implementation of a register field could not be captured properly.
IP-XACT and vendor tool limitations

- RTL implementation of a register or an array of registers could not be captured.

```
IPXACT XML

HDLPATH IN RTL

REG_1[n].FLD1
REG_1[1].FLD1
REG_1[0].FLD1
REG_2[n].FLD1
REG_2[1].FLD1
REG_2[0].FLD1
REG_2[n][m].FLD1
REG_2[1][m].FLD1
REG_2[0][m].FLD1
REG_2[n][0].FLD1
REG_2[1][0].FLD1
REG_2[0][0].FLD1

some_path.xxx_0X0_reg1_fld1.dout
some_path.xxx_0X4_reg1_fld1.dout
some_path.xxx_0XN_reg1_fldn.dout
some_path.xxx_0X100_reg2_fld1.dout
some_path.xxx_0X104_reg2_fld1.dout
some_path.xxx_0X180_reg2_fld1.dout
some_path.xxx_0X184_reg2_fld1.dout
```
IP-XACT and vendor tool limitations

- Capturing combination of sub-fields and arrays in IP-XACT is challenging.
- Handling or splitting HDL_PATH instantiation hierarchy.
- Setting HDL_PATHs to composite registers.
Custom Backdoor Automation solution

• Register declarations captured in IP-XACT XML format.
• register_hdl_paths captured in separate backdoor XML file.
Custom Backdoor Automation solution

• script parses the backdoor file and...
• Creates a derived class that inherits the top-level class of the front door register file generated by iregGen toolset
• Extends build method to setup hdlpaths to a single bit and instance of an array as follows:

```
REG_BLK.SAMPLE_REG1.add_hdl_path_slice("xcbi_nregs_m0.singlebitr1_0x0.dout", 29, 1, "RTL");
REG_BLK.rsvd.SAMPLE_REG2[1].add_hdl_path_slice("xcbi_nregs_m0.multibitrw_0x104", 0, 32, "RTL");
```

• Use add_hdl_path() configuration to add additional hdllpaths to the hierarchy.

```
REG_BLK.add_hdl_path("top_testbench.dut.block_inst1", "RTL");
```
## Results and statistics

<table>
<thead>
<tr>
<th>Number of register access</th>
<th>CPU time (sec) taken for frontdoor access</th>
<th>CPU time taken (in seconds) for backdoor access</th>
<th>Acceleration</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>5</td>
<td>1</td>
<td>5x</td>
</tr>
<tr>
<td>1000</td>
<td>50</td>
<td>3</td>
<td>17x</td>
</tr>
<tr>
<td>10000</td>
<td>418</td>
<td>19</td>
<td>22x</td>
</tr>
<tr>
<td>100000</td>
<td>4567</td>
<td>208</td>
<td>22x</td>
</tr>
</tbody>
</table>
Results and statistics
Results interpretation

• Front door access typically happens through AXI protocol (in our test environment).

• Results indicate that backdoor is 20+ times faster than the front door access.
  – In practice, the acceleration will always be better than reported here: The DUT used for testing was very simplistic.

• Backdoor access reduces simulation configuration time of an SOC.
Summary

• HDLPATHs can be incrementally setup based on hierarchy to enable reuse backdoor information

• Backdoor automation can drastically reduce simulation time.
Thank you...
Questions???