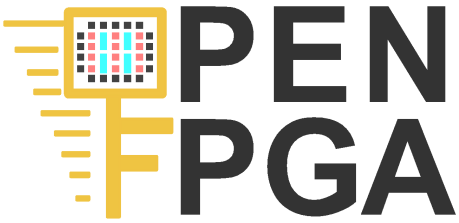
Tutorial 103: Functional Verification of the FPGA Fabric



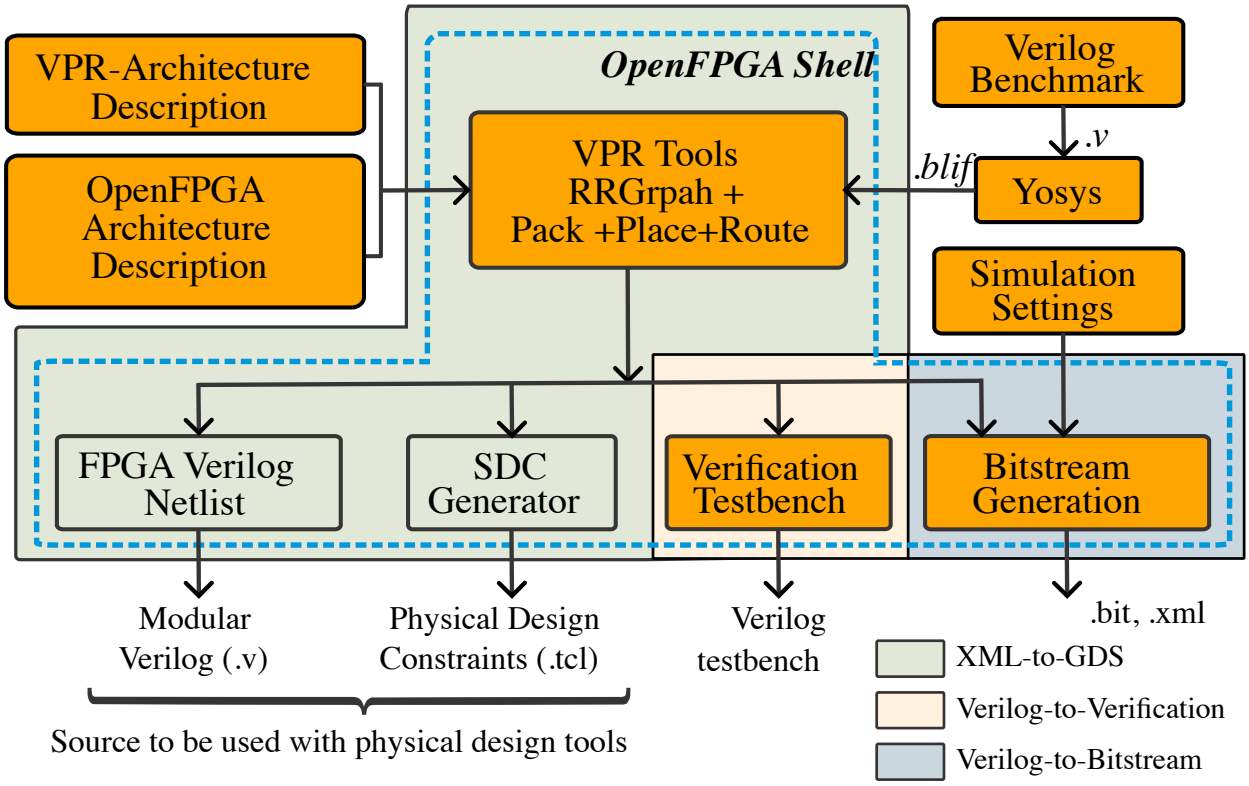
OpenFPGA Tutorial

Tutorial 103: Functional Verification of the FPGA Fabric

ORGANIZERS

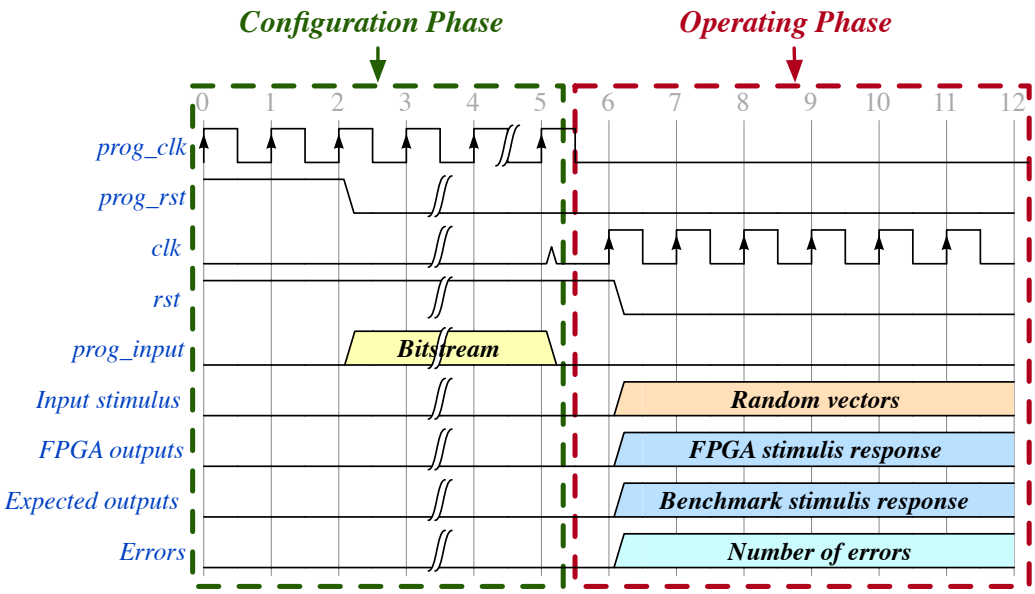
Tutorial 103: Functional Verification of the FPGA Fabric

FPGA Fabric Functional Verification



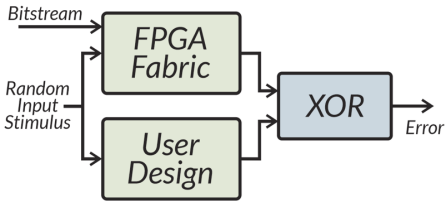
Tutorial 103: Functional Verification of the FPGA Fabric

FPGA Simulation Phases

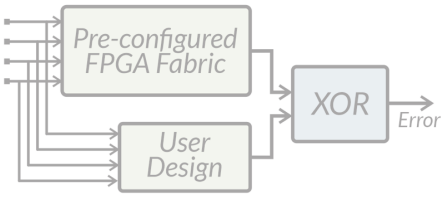


Tutorial 103: Functional Verification of the FPGA Fabric

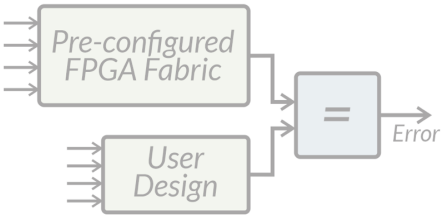
Customizable Testbenches



Random vector testbench generation



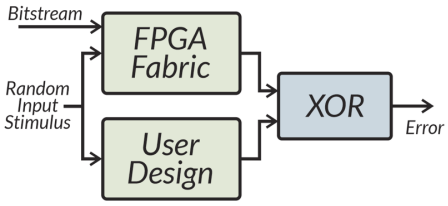
Pre-configured fabric verification



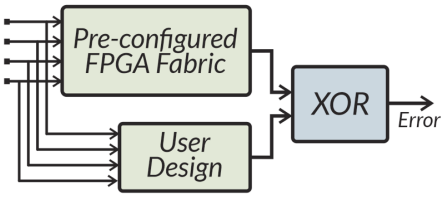
Formal verification

Tutorial 103: Functional Verification of the FPGA Fabric

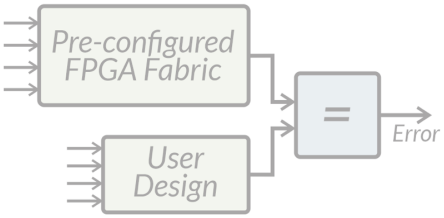
Customizable Testbenches



Random vector testbench generation



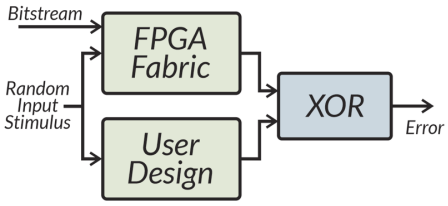
Pre-configured fabric verification



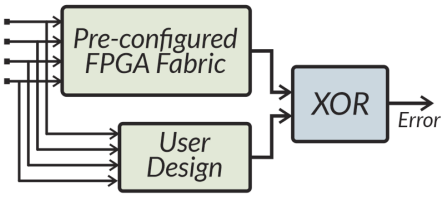
Formal verification

Tutorial 103: Functional Verification of the FPGA Fabric

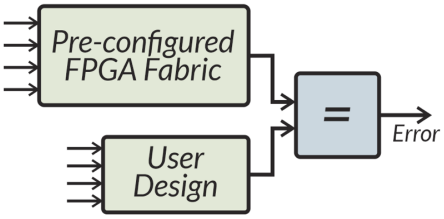
Customizable Testbenches



Random vector testbench generation



Pre-configured fabric verification



Formal verification

Tutorial 103: Functional Verification of the FPGA Fabric

Create OpenFPGA Task

```
create-task lab3 template_tasks/fabric_verification_template
run-task lab3
```

Changes in the task file

```
[SCRIPT_PARAM_MIN_ROUTE_CHAN_WIDTH]
end_flow_with_test=
vpr_fpga_verilog_formal_verification_top_netlist=
```

Command to Write Bitstream

```
# Build the fabric-independent bitstream and output in XML format
build_architecture_bitstream --verbose \
  --write_file fabric_independent_bitstream.xml

# Build fabric-dependent bitstream
build_fabric_bitstream --verbose

# Write fabric-dependent bitstream
# Supported format plain_text or xml
write_fabric_bitstream --file fabric_bitstream.bit --format plain_text
```

Tutorial 103: Functional Verification of the FPGA Fabric

Command to Write Testbench

(Configuration phase + Operating phase)

```
write_full_testbench \
      --file ./SRC \
      --reference_benchmark_file_path ${REFERENCE_VERILOG_TESTBENCH} \
      --explicit_port_mapping \
      --include_signal_init \
      --bitstream fabric_bitstream.bit
```

Write preconfigured testbench

```
write_preconfigured_fabric_wrapper \
    --embed_bitstream iverilog \
    --file ./SRC  \
    --explicit_port_mapping
write_preconfigured_testbench \
        --file ./SRC \
        --reference_benchmark_file_path ${REFERENCE_VERILOG_TESTBENCH} \
        --explicit_port_mapping
```

Tutorial 103: Functional Verification of the FPGA Fabric

Bitstream Formats

```
<bitstream_block name="lut6_DFFR_mem" hierarchy_level="6">
  <hierarchy>
    <instance level="0" name="fpga_top"/>
    <instance level="1" name="grid_clb_1__1_"/>
    <instance level="2" name="logical_tile_clb_mode_clb__0"/>
  </hierarchy>
  <bitstream>
    <bit memory_port="mem_out[0]" value="0"/>
    <bit memory_port="mem_out[1]" value="0"/>
    <bit memory_port="mem_out[2]" value="0"/>
    <bit memory_port="mem_out[3]" value="0"/>
    <bit memory_port="mem_out[4]" value="0"/>
  </bitstream>
</bitstream_block>
```

fabric-independent

```
0
0
0
0
1
0
0
1
0
0
1
0
1
0
0
1
```

.txt

```
<fabric_bitstream>
  <region id="0">
<bit id="2179" value="0" p
    </bit>
    <bit id="2178" value="
    </bit>
    <bit id="2177" value="
    </bit>
    <bit id="2176" value="
    </bit>
    <bit id="2175" value="
    </bit>
    <bit id="2174" value="
    </bit>
    ....
  </region>
</fabric_bitstream>
```

.xml

Analyse Results

- Check the bitstream generated in the `lab3/`

Exercise

1. Configure the OpenFPGA task for generating testbench for sequential design (`8-bit Counter`) and verify using iVerilog
2. Update task to simulate complete configuration and operating phase of the FPGA for `and2` design