

Tutorial 102: FPGA Fabric Netlist Generation



OpenFPGA Tutorial

Tutorial 102: FPGA Fabric Netlist Generation

ORGANIZERS



Tutorial 102: FPGA Fabric Netlist Generation

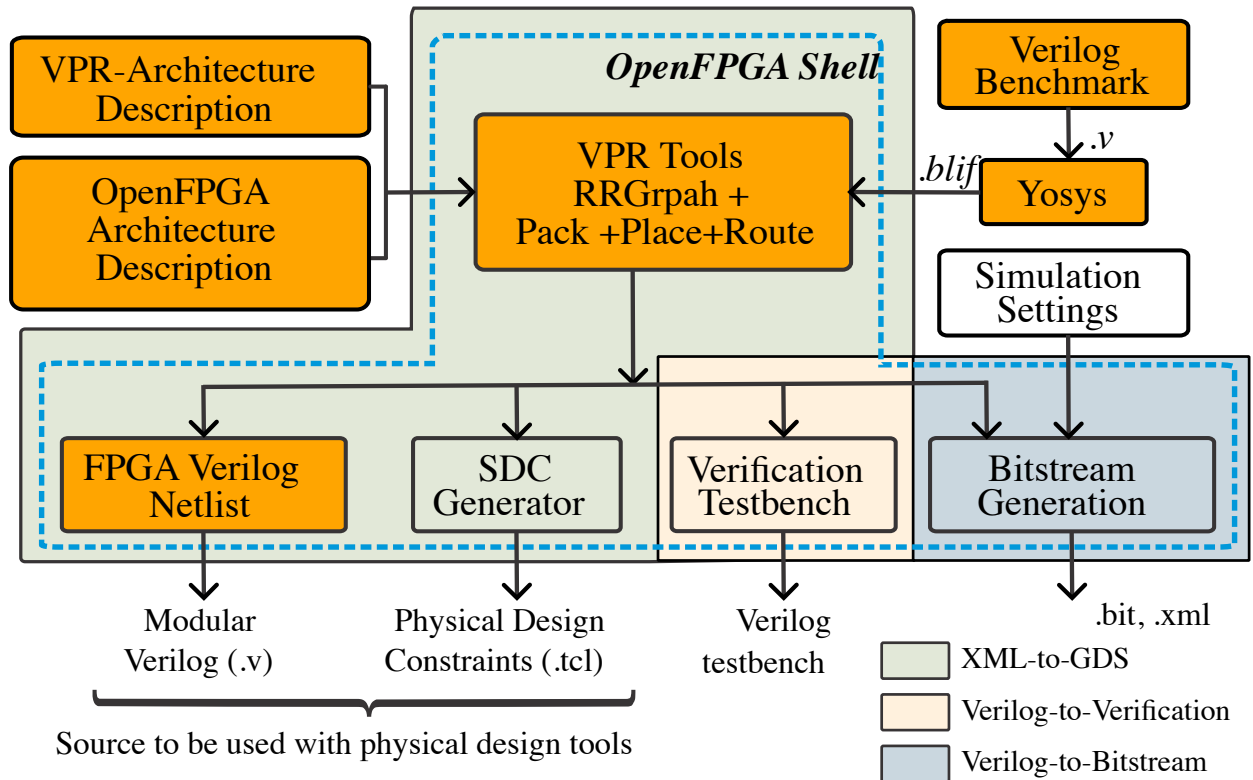
Objective

Generate **verilog netlist** of the **customized FPGA architect**
using OpenFPGA

(Note: This is a Verilog netlist of the FPGA fabric itself)

Tutorial 102: FPGA Fabric Netlist Generation

FPGA Fabric Netlist Generation



Tutorial 102: FPGA Fabric Netlist Generation

Create OpenFPGA Task

```
> create-task lab2 template_tasks/fabric_netlist_gen_template
Creating task      lab2
Template project   template_tasks/fabric_netlist_gen_template
```

Run Task

```
> run-task lab2
```

- **task.conf** file contains only one architecture and benchmark
- ***.openfpga** contains an additional command to map the **openfpga.xml** architecture file, and commands generate Verilog netlist

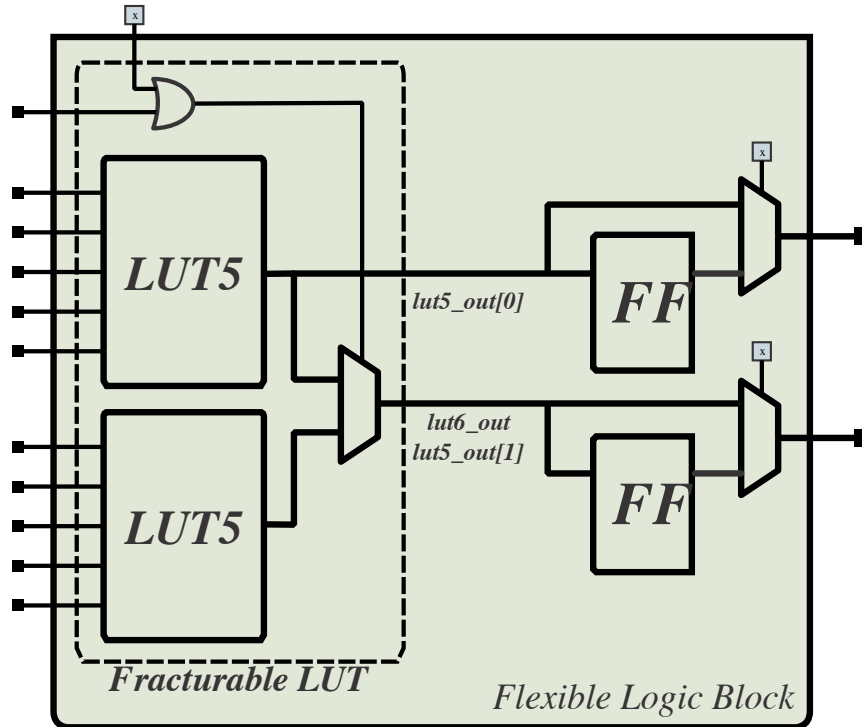
Tutorial 102: FPGA Fabric Netlist Generation

OpenFPGA Architecture File

```
<pb_type_annotations>
  <pb_type name="clb">
    <interconnect name="crossbar" circuit_model_name="mux_2level"/>
  </pb_type>
  <pb_type name="clb.fle" physical_mode_name="physical"/>
  <pb_type name="clb.fle[physical].ble6.lut6"
    circuit_model_name="lut6"/>
  <pb_type name="clb.fle[physical].ble6.ff"
    circuit_model_name="DFFSRQ"/>
  <pb_type name="clb.fle[n1_lut6].ble6.lut6"
    physical_pb_type_name="clb.fle[physical].ble6.lut6"/>
  <pb_type name="clb.fle[n1_lut6].ble6.ff"
    physical_pb_type_name="clb.fle[physical].ble6.ff"/>
  .....
</pb_type_annotations>
```

Tutorial 102: FPGA Fabric Netlist Generation

Physical mode of pb_type



CLB Physical Mode

Tutorial 102: FPGA Fabric Netlist Generation

OpenFPGA Shell Script

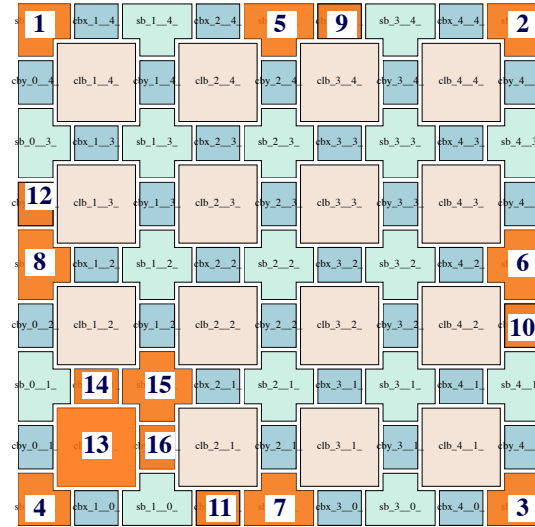
```
# Read OpenFPGA architecture definition
read_openfpga_arch -f ${OPENFPGA_ARCH_FILE}

# Annotate the OpenFPGA architecture to VPR data base
link_openfpga_arch --activity_file ${ACTIVITY_FILE} \
    --sort_gsb_chan_node_in_edges
.
....
..
# Write the Verilog netlist for FPGA fabric
write_fabric_verilog --file ./SRC --explicit_port_mapping \
    --include_timing --print_user_defined_template --verbose

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

Tutorial 102: FPGA Fabric Netlist Generation

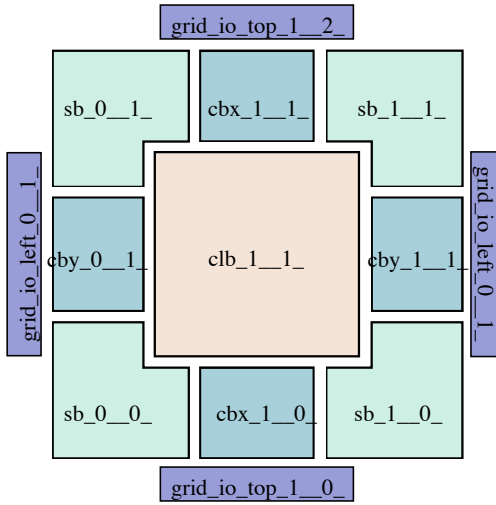
Homogeneous FPGA Tiles



**4 Corner + 4 Sides CB +
4 Sides SB + 4 tile**

Tutorial 102: FPGA Fabric Netlist Generation

FPGA Tiles (in this tutorial)



Tutorial 102: FPGA Fabric Netlist Generation

Source Directory Structure

```
SRC
├── sub_module
│   ├── luts.v
│   ├── .....
│   └── muxes.v
├── lb
│   ├── grid_clb.v
│   ├── .....
│   └── grid_io_bottom.v
├── routing
│   ├── cbx_1__0_.v
│   ├── .....
│   └── sb_1__1_.v
└── fpga_top.v
```

Tutorial 102: FPGA Fabric Netlist Generation

fpga_top.v Content

```
module fpga_top(pReset, prog_clk, set, reset, clk,
  gfpga_pad_GPIO_PAD, ccff_head, ccff_tail);

  inout [0:31] gfpga_pad_GPIO_PAD; // FPGA IO
  ....

  // modules instantiations
  grid_io_top      grid_io_top_1__2_ { ..... }
  grid_io_right    grid_io_right_2__1_ { ..... }
  grid_io_bottom   grid_io_bottom_1__0_ { ..... }
  grid_io_left     grid_io_left_0__1_ { ..... }
  grid_clb         grid_clb_1__1_ { ..... }

  ....
endmodule
```

fpga-top netlist

```
1
2 fpga_top:
3   - grid_io_top:
4   - grid_io_right:
5   - grid_io_bottom:
6   - grid_io_left:
7   - grid_clb:
8   - sb_0__0_:
9   - sb_0__1_:
10  - sb_1__0_:
11  - sb_1__1_:
12  - cbx_1__0_:
13  - cbx_1__1_:
14  - cby_0__1_:
15  - cby_1__1_:
16
```

fpga-top hierarchy

Tutorial 102: FPGA Fabric Netlist Generation

Switch box hierarchy

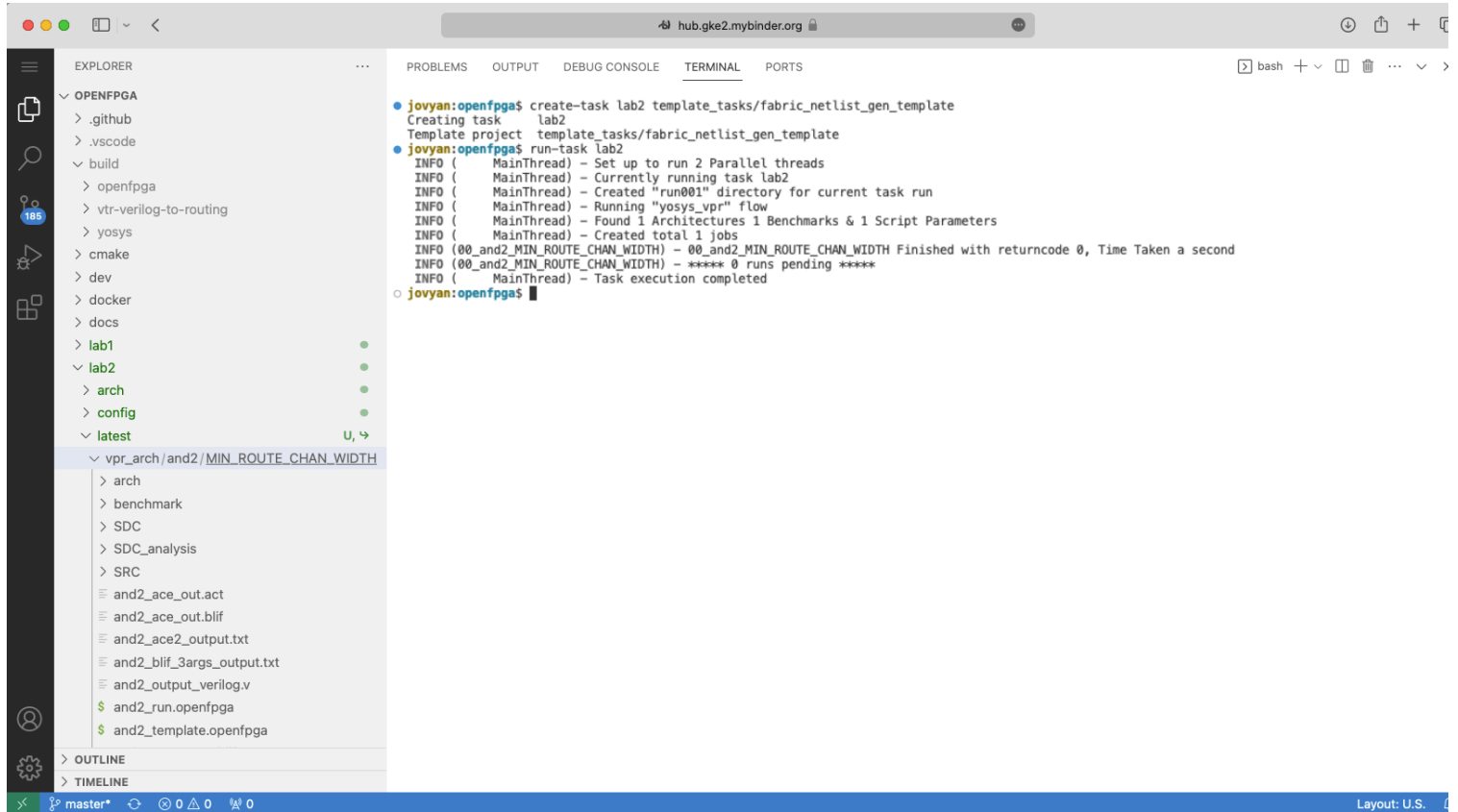
```
- sb_1__1_:
  - mux_2level_tapbuf_size4:
  - mux_2level_tapbuf_size4_mem:
  - mux_2level_tapbuf_size3:
  - mux_2level_tapbuf_size3_mem:
```

grid-clb hierarchy

```
- grid_clb:
  - logical_tile_clb_mode_clb_:
    - logical_tile_clb_mode_default__fle:
      - logical_tile_clb_mode_default__fle_mode_physical__ble6:
        - logical_tile_clb_mode_default__fle_mode_physical__ble6_mode_default__
        - logical_tile_clb_mode_default__fle_mode_physical__ble6_mode_default__
        - mux_1level_tapbuf_size2:
        - mux_1level_tapbuf_size2_mem:
      - mux_2level_size50:
      - mux_2level_size50_mem:
```

Tutorial 102: FPGA Fabric Netlist Generation

Task Complete



```
hub.gke2.mybinder.org

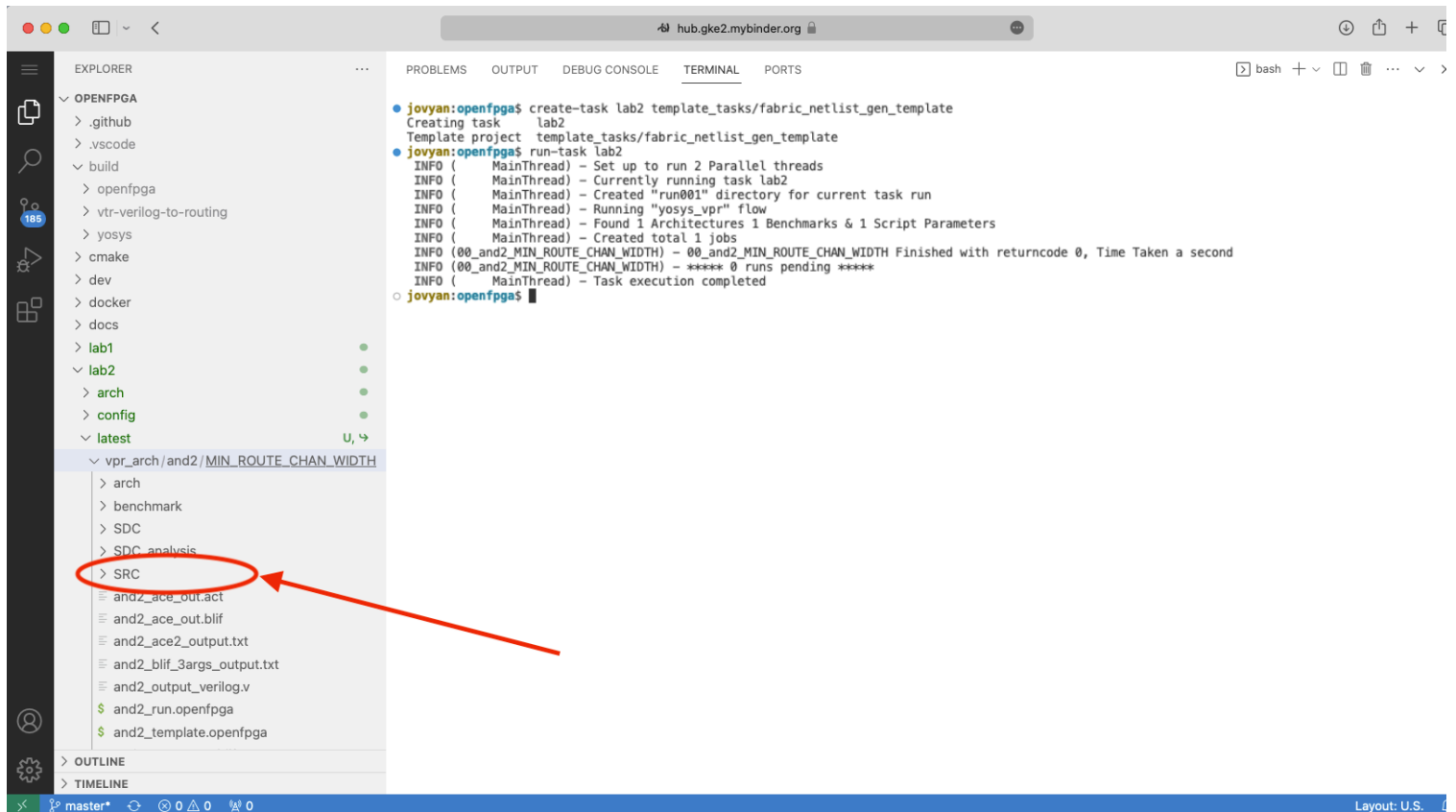
EXPLORER
└─ OPENFPGA
  ├── .github
  ├── .vscode
  ├── build
  │   ├── openfpga
  │   ├── vtr-verilog-to-routing
  │   ├── yosys
  │   ├── cmake
  │   ├── dev
  │   ├── docker
  │   ├── docs
  │   ├── lab1
  │   ├── lab2
  │   ├── arch
  │   ├── config
  │   └── latest
  │       ├── vpr_arch/and2/MIN_ROUTE_CHAN_WIDTH
  │       ├── arch
  │       ├── benchmark
  │       ├── SDC
  │       ├── SDC_analysis
  │       ├── SRC
  │       ├── and2_ace_out.act
  │       ├── and2_ace_out.blif
  │       ├── and2_ace2_output.txt
  │       ├── and2_blif_3args_output.txt
  │       ├── and2_output_verilog.v
  │       ├── and2_run.openfpga
  │       └── and2_template.openfpga
  ├── OUTLINE
  └── TIMELINE

PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

jovyan:openfpga$ create-task lab2 template_tasks/fabric_netlist_gen_template
Creating task lab2
Template project template_tasks/fabric_netlist_gen_template
jovyan:openfpga$ run-task lab2
INFO (MainThread) - Set up to run 2 Parallel threads
INFO (MainThread) - Currently running task lab2
INFO (MainThread) - Created "run001" directory for current task run
INFO (MainThread) - Running "yosys_vpr" flow
INFO (MainThread) - Found 1 Architectures 1 Benchmarks & 1 Script Parameters
INFO (MainThread) - Created total 1 jobs
INFO (00_and2_MIN_ROUTE_CHAN_WIDTH) - 00_and2_MIN_ROUTE_CHAN_WIDTH Finished with returncode 0, Time Taken a second
INFO (00_and2_MIN_ROUTE_CHAN_WIDTH) - ***** 0 runs pending *****
INFO (MainThread) - Task execution completed
jovyan:openfpga$
```

Tutorial 102: FPGA Fabric Netlist Generation

Location of Results



Netlist located: **lab2/latest/<architecture>/<benchmark>/<chan_width>/SRC**

Tutorial 102: FPGA Fabric Netlist Generation

Exercise

1. Identify the number of **global signals** and their connector with top-level instance
2. Generate an FPGA netlist for **4×4 homogeneous FPGA Fabric** and identify a unique number of modules