PUF OVER FPGA

Episode 03: Building PUF with VHDL and Verilog – sub 01

AGENDA

Part 1 - Course intro

Part 2 - What is PUF and discussion of the project structure

Part 3 - HDL Development of PUF

- **sub 01**

Part 4 - Building a MicroBlaze-based soft processor system

Part 5 - Connect PUF to MicroBlaze and assign FPGA pins

Part 6 – Using pblock for separate PUF and MicroBlaze placement

Part 7 - Introduction to TCL and placing PUF on FPGA

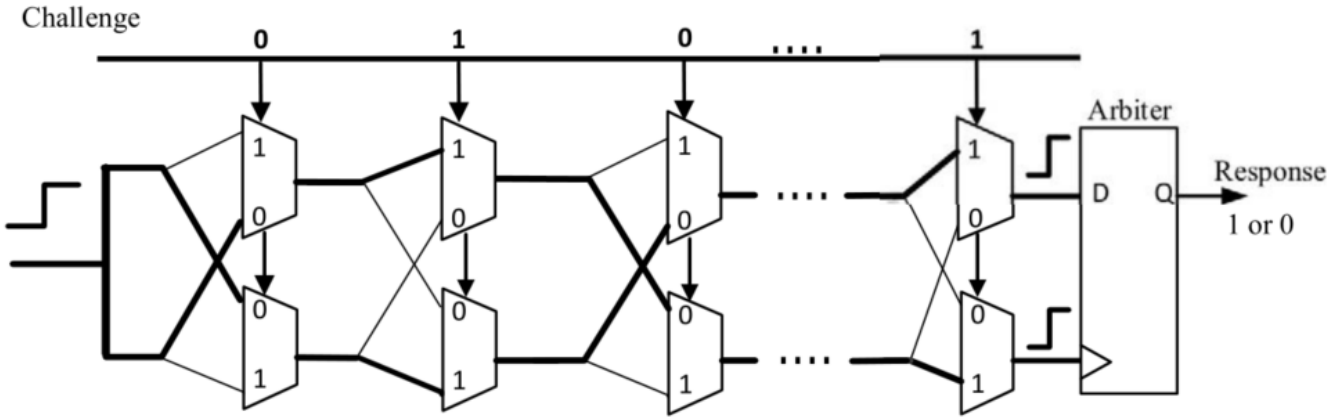
Part 8 - Writing code for the processor system

Part 9 - Debugging and running PUF

**Number of episodes could be change in future*

FPGA community

ARBITER PUF



Arbiter PUF

The main blocks of Arbiter PUF are:

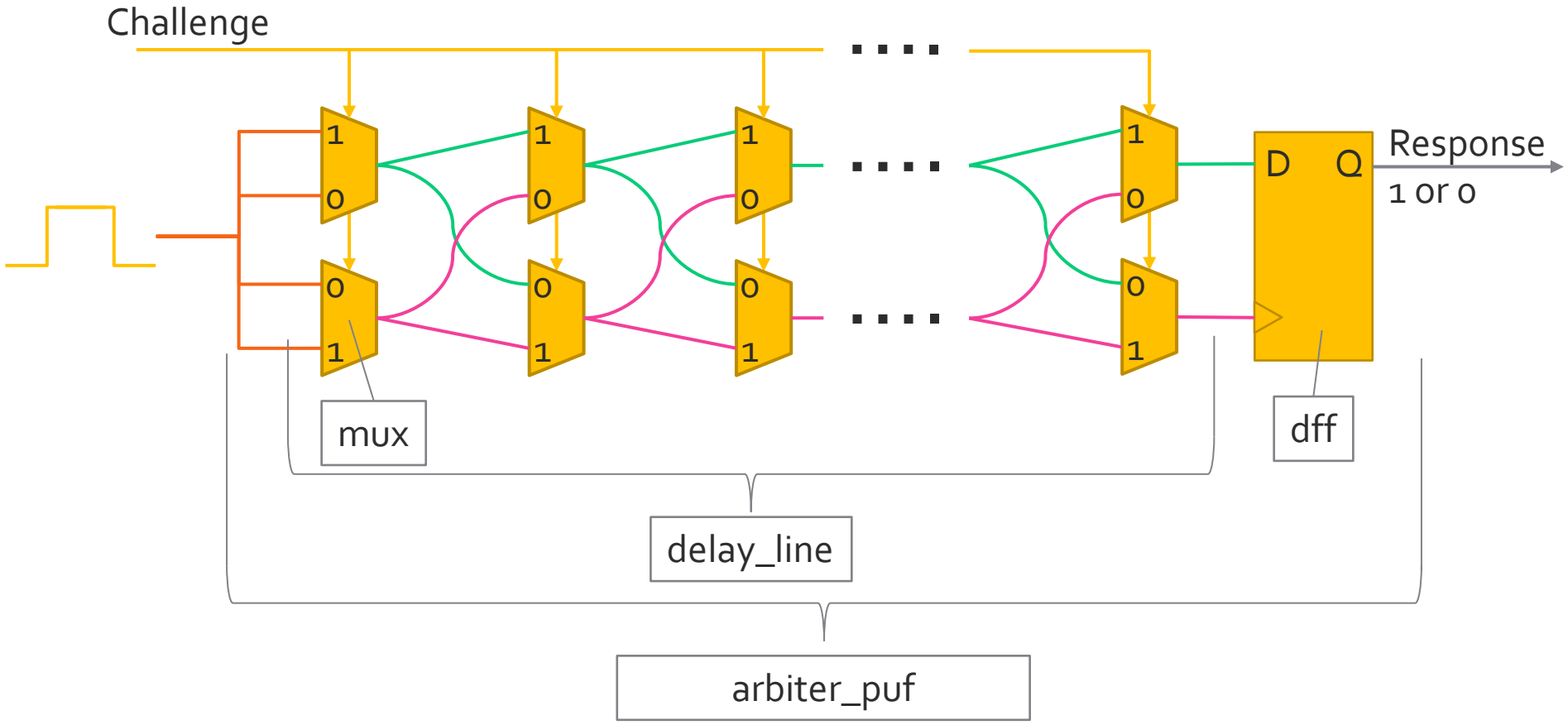
- MUX-pair based delay line
- D-flip-flop with input and clock connected to out of delay line

To Do:

- Create multiplexer (MUX)
- Build MUX-based delay line
- Describe D-flip-flop
- Combine all to Arbiter PUF

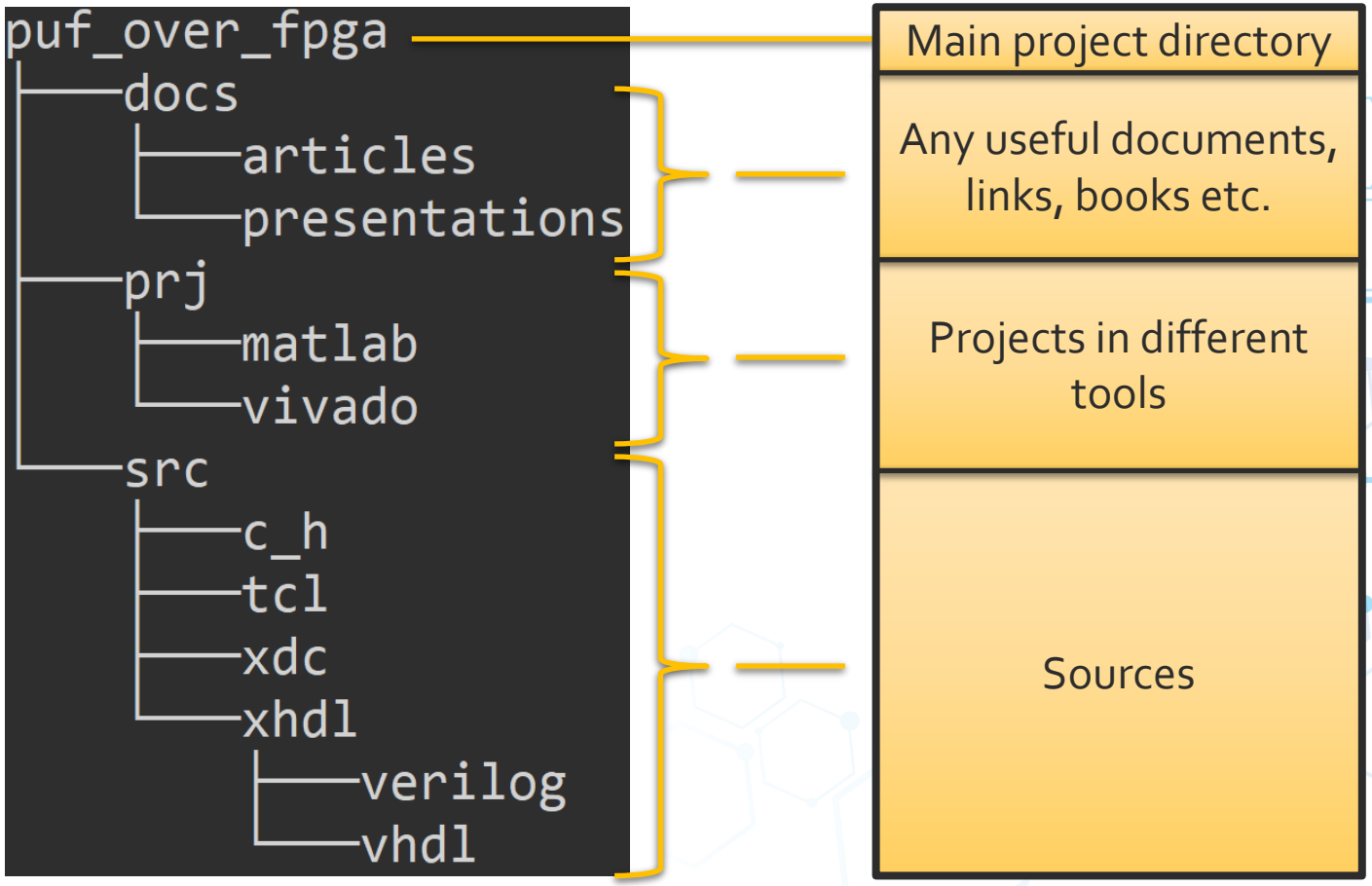
*Source link

PROJECT HIERARCHY



*Source [link]

PROJECT DIRECTORIES



*Source [link]

LEARNED TODAY HOW TO:

1. Create Vivado project
2. Add source files
3. Write behavioral description of multiplexer on VHDL and Verilog
4. Elaborate design
5. Running design synthesis
6. Review of post-synthesis results

HOMEWORK

1. Try describe multiplexer using other statements, like:
   - `process` or `always`
   - `if`
   - `case`
   - any other (just for a little practice)

2. It is very good practice to use synthesis guide for your tool to provide a correct description of modules. For Vivado it's UG901 - Synthesis

HAVE A QUESTIONS?

Telegram

@fpgacommunity

E-mail

admin@fpgacommunity.com

Find more

github

Support

Patreon

Donation Alerts