PUF OVER FPGA

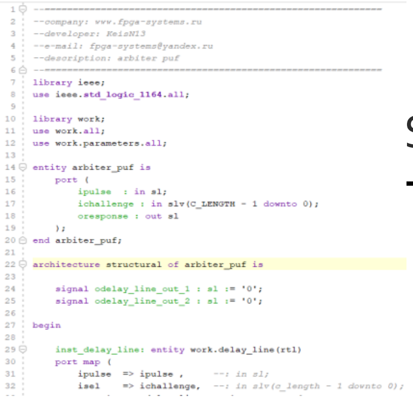
Episode 01: Course intro

FPGA community

COURSE REQUIREMENTS

1. Minimal knowledge of VHDL or Verilog
2. Xilinx Vitis / Vivado software
3. Common understanding of FPGA design flow

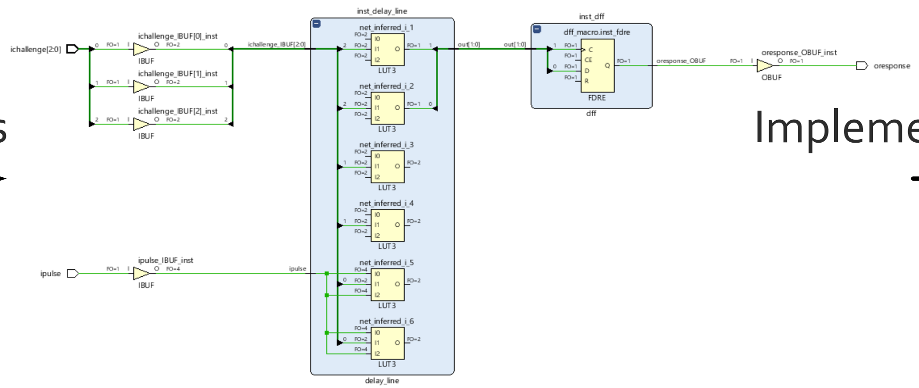
HDL



```
--=================================================================
--company: www.fpga-systems.ru
--developer: KeisN13
--e-mail: fpga-systems@yandex.ru
--description: arbiter puf
--=================================================================
library ieee;
use ieee.std_logic_1164.all;

library work;
use work.all;
use work.parameters.all;

entity arbiter_puf is
    port (
        ipulse  : in sl;
        ichallenge : in slv(C_LENGTH - 1 downto 0);
        oresponse : out sl
    );
end arbiter_puf;

architecture structural of arbiter_puf is

    signal odelay_line_out_1 : sl := '0';
    signal odelay_line_out_2 : sl := '0';

begin

    inst_delay_line: entity work.delay_line(rtl)
    port map (
        ipulse  => ipulse ,      --: in sl;
        isel    => ichallenge,  --: in slv(c_length - 1 downto 0);
```

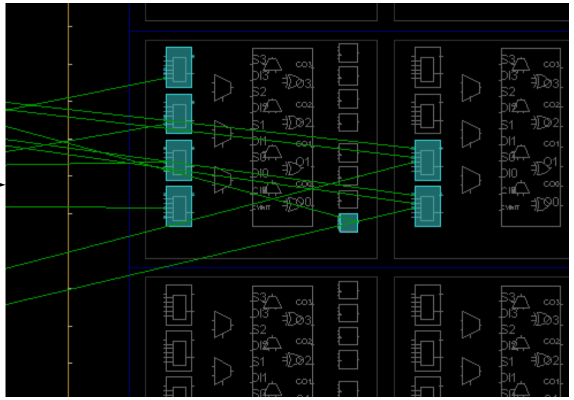
Synthesis

Netlist



Implementation

Placed and routed design

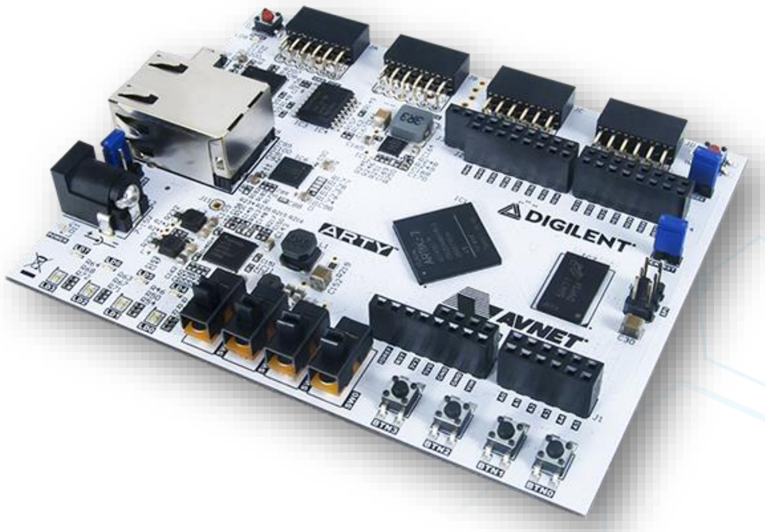


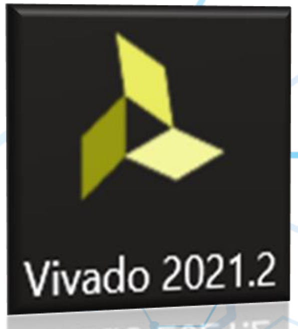
WHAT WILL WE LEARN?

1. Writing VHDL and Verilog code with two types of component descriptions.
2. Manage synthesis settings using VHDL and Verilog language attributes.
3. Building processor system based on the MicroBlaze soft processor.
4. Using pblock and elements placement inside the FPGA (LOC)
5. Writing code for a processor system on C language.
6. Control custom logic using MicroBlaze.
7. Environment management using TCL.
8. Debugging using Intagrated Logic Analyzer.
9. Relevant user manuals.

WHAT IS THE PROJECT, BOARD AND TOOL?

1. Our project is implementing PUF (Physical Unclonable Function) in FPGA
2. Board could be any with Xilinx FPGA **BUT NOT earlier than Xilinx 7-series** FPGA – I’am will use Arty-A35T evaluation board from Digilent
3. Xilinx Vitis + Xilinx Vivado (v.2021.2)





WHY PUF? WHY NOT UART OR LED?

1. There are a lot of UART and led blink tutorials – lets try something unusual to start – it will simple but very complex design.
2. This project will not require very much knowledge of FPGA design from you, but it will affect many aspects of it.
3. There are several non-obvious and curious, from the point of view of design limitations, moments, ignorance of which can easily doom the developer to several “happy” days of searching the solution of problem.
4. PUF is a very good thesis research topic for your bachelor, master or PhD degree.
5. Each PUF generates a unique data set, so your implementation will be unique. And it’s also possible to connect PUF to LEDs and I think you will be able to do that by yourself.

AGENDA

➡ **Part 1 - Course intro**

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

Part 3 - HDL Development of PUF

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*

HAVE A QUESTIONS?

Telegram

@fpgacommunity

E-mail

admin@fpgacommunity.com

Find more

github

Support

Patreon

Donation Alerts