ECE505 Project RISC-V Processor - Common Questions

-

Q0 AMD Xilinx Licence Issue

- All the desktops in **AK317** and the Vivado Software are pre-installed and ready to use.

If you want to Download the Vivado locally (Option 1)

- Register an AMD Xilinx Education account through your WPI account on the official website.

Q1 Change top-level problem:

"boost::filesystem::remove: The process cannot access the file because another process "

Q2 What and how to use the *locked* signal in MMCM:

What is the *locked signal and how to use it?*

Q3 Some part of the demo code and testbench design is different from the lecture or book.

- > These codes are provided to help students who are not familiar with Verilog. If you have taken ***ECE2029,*** and ***ECE3829*** before, please feel free to design on your own.
  > The demo codes' port definition and structure are slightly different from the book and instruction. You do not need to follow the specific port or code structure design, feel free to realize the design in your own way

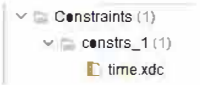
Q4 How to Synthesis and Implementation in ~~Xilinx~~ AMD Vivado

1. Create a New Project

1. **Open Vivado**: Start Vivado and select "Create New Project" on the welcome page.
2. **Project Name and Location**: Choose a project name and location. Avoid spaces and special characters in the file path.
3. **Project Type**: Select "RTL Project" and check "Do not specify sources at this time" if you want to add sources later.
4. **Select Target**: Choose the FPGA device or family you are targeting. (**default** or **xc7vx485tffg1157-1**)

2. Add Sources

1. **Create or Add Source Files**: You can create new HDL files or add existing ones. This includes your main design files (.v or .vhdl) and any constraints files (.xdc).
2. **Add Constraints**: If you have an XDC (Xilinx Design Constraints) file, add it to your project. This file specifies the pin assignments and other constraints. **(Time Constraint for 20MHz is provided)**



3. Synthesis

1. **Run Synthesis**: In the Flow Navigator, click on “Run Synthesis”. This process converts your HDL code into a gate-level representation. (Warning in the report is acceptable, but **NO critical Warnings** and **Errors**)
2. **Review Synthesis Report**: Check for any errors or warnings that need to be addressed.

4. Implementation

1. **Run Implementation**: After successful synthesis, run implementation. This step includes translation, mapping, placing, and routing your design onto the FPGA architecture.
2. **Review Implementation Reports**: Like with synthesis, check for any issues that need resolving.

Q5 How to set the time constraint (20MHz)

- you can add the "time.xdc" to the project from the supportive files. And run the implementation

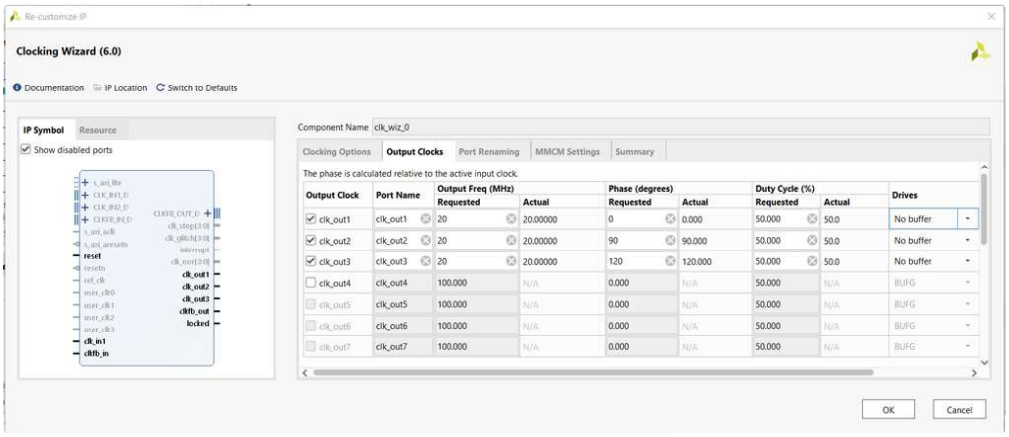
```
create_clock -add -name CLOCK_20 -period 50.00 [get_ports CLOCK_20]
```

(Keep the (* DONT_TOUCH = "TRUE" *) comment)

Q6 How to optimize the design if you fail to reach the time constraint

- Option1: Change the phase difference larger
  e.g.: change from **(0,45,90)** to **(0,90,120)**

- Option2: Change the MMCM buffer drivers from (BUFG) to (No buffer) as follows (Huge Delays in Datapath)



In this case, you need to add the feedback clock port when you instantiate the MMCM:

```
// ================== MMCM ============================
wire clk_fb;
clk_wiz_0 clkWiz(
    .clkfb_out(clk_fb),
    .clk_out1(clk_0),
    .clk_out2(clk_1),
    .clk_out3(clk_2),
    .reset(1'b0),
    .locked(locked),
    .clk_in1(CLOCK_20),
    .clkfb_in(clk_fb)
);
```