Assignment "Final Project Microprocessor"

Due date: June 7 (Friday)

(PRELIMINARY) STUDENT VERSION – "FIRST MICROPROCESSOR"

The final project is implemented on your own. Please start working on the following tasks (100 pts total). In addition, you *must* consider the following:

- Vivado creates a default template including a comment block with:
  - "Company" – put some creative company name here
  - "Engineer" – put only your name here
  - "Design Name" – based on task description (*do not leave empty*)
  - "Project Name" – Final Project
  - "Description" – based on task description (*do not leave empty*)
- To prepare your assignment for submission through Canvas, you must use (in Vivado): `File > Project > Archive`
- Preparing your HDL project through this process is especially important when working with IP cores such as DSPs and BRAMs!

We will be working on this project together through Zoom as a lab. Please try to follow along and ask questions if you get stuck somewhere. Your project includes all tasks mentioned here (even those, that you manage to complete during our joint lab time).

1 Task

For the final project, you want to implement your own microprocessor. This follows exactly the design as presented in class. Please use the following basic structure for your microprocessor:

```
TOP
|-EX_MODULE
|-IF_MODULE
|
(optional part)
|-COPROCESSOR
```

Use the following part for your assignment: XC7A100T-1CSG324C. In the Vivado project manager, you need to select xc7a100tcsg324-1. Please make sure to select this part as we want to compare your synthesis results (e.g., if we manage to do this, we could do an anonymous scoreboard of the fastest design with a small prize for the best implementation?).

The grading will follow the structure as discussed in class (with a total of 45%):

- LaTeX-documentation - 5%
- Testbenches - 10%
- TOP module - 10%
- EX module - 10%
  - ALU submodule will be 2.5%
  - SRU submodule will be 2.5%
  - EX module "glue logic" 5%
- IF module - 10%

Should you decide to implement the COPROCESSOR, then you can earn an additional 10% (e.g., in case you missed submitting a previous assignment). With the documentation, a total of 105% can be earned in this course. If the coprocessor is implemented additionally, a total of 115% can be earned in this course. This grading follows the structure as discussed in class.

Full points on the TOP module can only be earned if your design can be synthesized, runs above 1 MHz, and works. The port description of the TOP module as given in the course materials should allow you synthesize the design.

2 Output

We are mainly looking at the following project outputs:

- Implementation
- Testbenches
- LaTeX documentation

Each 'major' module should be complemented with its own testbench as you do see fit. The TOP module should test your microprocessor and have the following test categories:

- TEST category one: `LDI` operation(s)
- TEST category two: `ALU/SRU` operation(s)
- TEST category three: memory operation(s)
- TEST category four: `JMPD`

- TEST category five: JMPS
- TEST category six: RETS
- TEST category seven: returning to program memory address 0 via JMPC

Please note that these tests are the result of executing your program and not test inputs through your test bench! In your TOP test bench, you only start your microprocessor using the START signal. **Please note: it is sufficient for the TOP module to simply check the output through the OPRESULT signal.** OPRESULT is the value you would use as writeback (e.g., the result of an ALU operation before writing it back). There is no need to check the individual results in registers and such (so as to avoid writing too many lines of code).

3 Hints

Good luck! =)