

CPU Design Project: Part 3 Datapath components and control unit implementation

Report Due Wednesday, 10/23/13

Develop and verify the VHDL model for *each unique component* within your datapath. Also, design and test a VHDL “behavioral” model of the control unit to realize the behavior described in your control signal table in part 2, including any instruction decoding. Thoroughly simulate the control unit and each component individually. **Guaranteeing the correct functioning of each component is very important** since the control unit and all components will be connected in the next part to form a top-level component. Any failing component will induce the malfunctioning of the top-level module.

To facilitate debugging on the Altera FPGA board, the datapath needs **an additional 4-bit input named “inr”** used with a multiplexer to select one of your CPU registers, with a **16-bit output named “outvalue”** to display the value of the selected register. These two ports will be part of the primary inputs and primary outputs of the final CPU. These two ports are added to help us correctly implement our CPU all the way through this project. In the simulation phase, they are used to help us decide whether we get the correct results saved in the correct register. In the experiment phase, these two ports are used to display the content of target register on the FPGA board to demonstrate the correct execution of your program. **In real CPU, we don’t have these two ports.** They are only used to help us implement this project.

For single-cycle and pipeline datapath, the lecture slides show separate instruction memory and data memory. If your datapath uses a separate data and instruction memory, you can actually write a VHDL model of your own instruction memory and leave the data memory to be generated from Altera’s Megafunction library in part 5. In this way, it will be easier for you to simulate a small program in part 4. Of course, you can also leave both memories to be generated in part 5. In this way you will have to manually supply each instruction to the datapath in part 4. If your design is multi-cycle datapath, you can leave your memory to be generated in part 5, or write a VHDL model of a memory to be used to help test part 4.

Notes:

1. Refer the [*LeonardoSpectrum for Altera HDL Synthesis Guide*](#) to write the VHDL code according to the synthesis guidelines so that in the final stage, your design is synthesized correctly by the FPGA.
2. For every sequential component within your datapath, you will need an input “reset” signal to establish initial states.
3. The VHDL model is to be a register-transfer-level (RTL) design **(not gate level)**.
4. Design and **test** VHDL models of **each unique component**.
5. Submit all VHDL code. Submit all simulation results and clearly annotate each simulation results.