Project 2   Register File of 16-bit Processor

Objectives

The project requires you to design the register file module of the Verilog processor started in Project 1. It may be helpful to refresh your memory by referring to the DOCS for that project.

In this assignment you will design the sequential logic components, in the form of a register file module, and ensure that it works together with the **decoder** and **alu** modules implemented in Project 1.

The document **processorComponents** shows the input/output ports of the ***regfile*** module that you must implement in this project. Identify where the input signals come from – either directly from the **decoder** or **alu** modules, or from the testbench module that combines the **decoder**, **alu**, and **regfile** modules together. In the latter case, a few signals (**clk**, **clk_en** and **reset**) are generated in the testbench module, but the rest are obtained by combining the **alu** or **decoder** outputs to generate a signal for the register file. It is important for you to understand the data flow between modules.

> **regfile**: the module maintains the set of general purpose processor registers $0 to $7, and two 1-bit flags that store the latest carry and borrow outputs generated by the **alu**.
>
> The module must write to the specified register when directed, and must update the carry and borrow flags based on the input port values. At its output ports, it must return the values in the specified source registers, the data for a store instruction, and the current values of the carry and borrow flags.
>
> The comments in the stub for this module in file **regfile.v** provide additional details.
>
> Unlike the three modules of Project 1, which were purely combinational, this module uses clocked sequential logic. Updates to a register in the register file and to the carry/borrow flags occur only at the positive edge of a clock signal supplied to the module. There are additional conditions on when exactly a write is to be performed as specified in the comments in the code stub.

Assignment

A Verilog stub for the **regfile** module to be implemented, a testbench module, and a results file can be found in the tar file. To test your register file design, you will need to compile your solution together with your **alu** and **decode** modules from Project 1 and the testbench module provided. Check your answers by comparing with the results file provided.

For submission upload your completed **regfile.v** file. Please do not change any file names. Within the skeleton module provided do not change any of the existing declarations (input/output ports in particular) in any way.

Although this assignment only requires behavioral Verilog design, think about the actual hardware necessary. How is the register implemented using Flip Flops and combinational logic?

Due Date: Friday, November 6, 11:59 pm