EC 413 – Computer Organization
Project Phrase 1 Report
Hok Yin Shum

ALU

The Arithmetic Logic Unit is the core of the execution processor of the single cycle MIPS processor that I create for this project. The ALU takes two 32bits operand inputs: operand_A and operand_B and perform logic or arithmetic operations based on the 4bits OP-code ALU control. The ALU implemented can have the follow functions: ADD, OR, ADD, SUB, < , and negation of bit wise OR. The output is named ALU_result which also has 32bits. Lastly, there is a zero output to indicate whether the ALU_result is 0.
The module is designed to be used for addition of any two 32bits operands and can be used for both R and I type instructions for operating two register values or a register value with an immediate value that is 32bits.
The learning goal of this part is to understand the behavior of the ALU more thoroughly and to be able to implement and test its behavior in Verilog.

The Register Files

This module is the used to construct the registers file for the MIPS processor. There are 32 registers of 32bits with clock, reset as input. There are also two read register addresses inputs: read_sel1, and read_sel2. They are 5bits each used to indicate the address of register for read_data1 and read_data2 32bits output correspondingly. Similarly, the register file also features one input port write_data for 32bits writeback and its address is indicated by write_sel. Lastly, there is a write wire to indicate whether the processor is writing anything to the register from ALU or MEM.
The learning goal of this part is to understand the behavior of the register file and understand the implementation of the writing data needs to be done at positive edges.

Memory

The memory module is a general module that has 8bits address of 32bits word addresses in the memory. There is 1bit clock input, 1bit MemRead, and 1bit MemWrite input. The output is a 32bit output ReadData for reading data in memory; the input is a 32bit input writeData for writing into the memory. The design is single cycle and the memory module can be used by the CPU unit to create instance of Data memory and Instruction Memory.
The learning goal of this part is to solidify our understanding of the memory module and its functionality through implementation of it in Verilog.