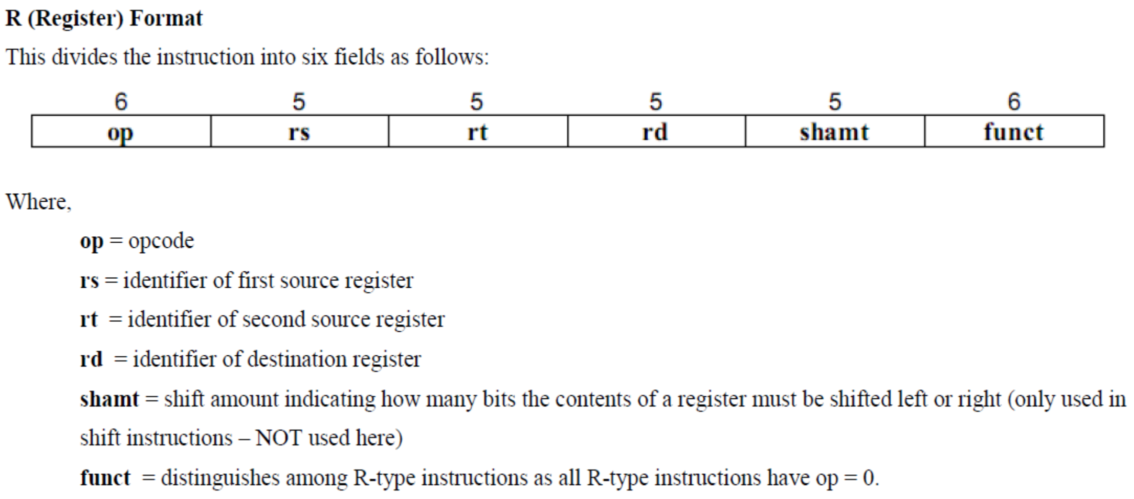
CSE 331 - Computer Organization
Project-2 R-Type MIPS

Due to November 10 Monday–17:00

In this project, you will use Altera Quartus II with Verilog. You will design a part of the 32-bit MIPS processor. The block that you will design will get an 32-bit instruction as its input and compute the resultant value and stores it to the destination register given by the instruction. The supported instructions in your design will be **add, sub, and, or, sra, srl, sll, sllv** and **slt** instructions. The input of your top-module will be the **instruction only**. The output of your top-module will be the output of your ALU to follow and check its computations during the simulation. You will write the memory contents before and after the execution of instructions using `writememh` in your testbench verilog code. You will initialize memory contents using `readmemh`.



R (Register) Format

This divides the instruction into six fields as follows:

6	5	5	5	5	6
op	**rs**	**rt**	**rd**	**shamt**	**funct**

Where,

op = opcode

rs = identifier of first source register

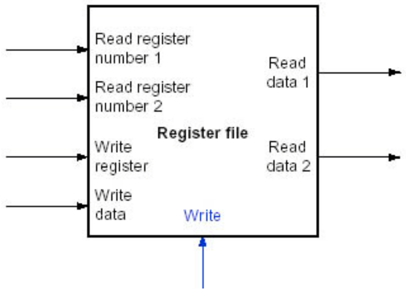
rt = identifier of second source register

rd = identifier of destination register

shamt = shift amount indicating how many bits the contents of a register must be shifted left or right (only used in shift instructions – NOT used here)

funct = distinguishes among R-type instructions as all R-type instructions have op = 0.

1. You will design **ALU module using only structural Verilog with assign** statement. (So you cannot use **always, initial, if, case** statements.)
2. You will design **Register module** using **behavioral Verilog.** The **register** has **32 registers** each containing **32-bit number**.



3. You will write a **top-level Verilog module** to connect ALU and Register modules accurately to finish the project.
4. You will write a **working testbench and simulate your design by ModelSim** as you learnt in the PS.

The project will be explained in detail in the PS. But do not wait until the PS to start the project. If you did not understand – sorry, if you understood but want to understand better☺ – feel free to ask. Attend this PS, it is a must.

Please be sure that your design simulates correctly. Designs that are not even simulating can get at most 25 points.

Submit your Altera Project folder as a zip file to Moodle. We will simulate your design using not only your testbench but also our testbench to see whether all instructions are executing correctly or not.

No late submissions even if it is 5 minutes. No medical reports. No excuses. No cry. So start early.

Any cheating attempt with the previous years' projects or with your friends or Internet will result in at least -100 and at most -300. No matter you gave or take the code. Protect your code.

The "best" projects will have an opportunity to upload their designs to a FPGA board and we will execute it in class. What we mean by "best" will be explained in this week's PS.