

Lab# 11 THE 16-BIT REGISTERS FILE

Instructor: I Putu Danu Raharja.

Objectives:

Learn to implement a simple 16-bit single-stage Registers File and an Instruction memory.

Method:

Design the circuit for a 16-bit register file and an instruction memory and verify it's operation using the simulator.

Preparation:

File To Use:

11.1 OVERVIEW:

Suppose we would like to design a simple 16-bit MIPS-like processor with seven 16-bit general-purpose registers: R1 through R7. R0 is hardwired to zero and cannot be written, so we are left with seven registers. There is also one special-purpose 16-bit register, which is the program counter (PC). All instructions are also 16 bits. There are three instruction formats, R-type, I-type, and J-type as shown below:

R-type format:

4-bit opcode (Op), 3-bit register numbers (Rs, Rt, and Rd), and 3-bit function field (funct)

Op(4)	Rs(3)	Rt(3)	Rd(3)	funct(3)
-------	-------	-------	-------	----------

I-type format:

4-bit opcode (Op), 3-bit register number (Rs and Rt), and 6-bit immediate constant

Op(4)	Rs(3)	Rt(3)	Imm(6)
-------	-------	-------	--------

J-type format:

4-bit opcode (Op) and 12-bit immediate constant

Op(4)	Imm(12)
-------	---------

For R-type instructions, Rs and Rt specify the two source register numbers, and Rd specifies the destination register number. The function field can specify at most eight functions for a given opcode. We will reserve opcode 0 and opcode 1 for R-type instructions.

For I-type instructions, Rs specifies a source register number, and Rt can be a second source or a destination register number. The immediate constant is only 6 bits because of

the fixed size nature of the instruction. The size of the immediate constant is suitable for our uses. The 6-bit immediate constant is signed (and sign-extended) for all I-type instructions.

For J-type, a 12-bit immediate constant is used for instructions such as J (jump), JAL (jump-and-link), and LUI (load upper immediate) instructions.

11.2 LAB EXERCISE

Based on the above requirement, design the following components:

- a. ***Instruction Memory decoding scheme***: This component takes an address from PC as an input, then read the instruction from the RAM at that address and separate the content into seven parts (outputs): Opcode (4 bits), Rs (3 bits), Rt (3 bits), Rd (3 bits), funct (3 bits), Imm6 (6 bits), and Imm12 (12 bits).
- b. ***Registers File***: This component consists of eight 16-bits registers. This component has inputs Rs (3 bits), Rt (3 bits), Rd (3 bits), Read/Write control signal (1 bit), Input-Data (16 bits), and 2 Output-Data (each is 16 bits).