North South University

Department of Electrical & Computer Engineering

LAB REPORT

Course Name: **CSE332L- Computer Organization and Architecture Lab**

Experiment Number: 05

Experiment Name: **Design of a Register File**

Experiment Date: 27/07/2022

Report Submission Date: 02/08/2022

Section: 02

Group Number: 01

Student Name: **Md. Baker**	Score
Student ID: **1911672642**	
Remarks:	

Exp Name: Design of a Register File

Objectives:

- We have understood how register file works
- We have understood how data fetching works
- We can Design the interfacing for reading data from any of those registers.
- We can Design the interfacing for writing data to any of those registers

Equipment List:

A 16-bit ISA. We need to design a register file for this ISA.

Theory:

A register file is a means of memory storage within a computer's central processing unit (CPU). The computer's register files contain bits of data and mapping locations. These locations specify certain addresses that are input components of a register file. Other inputs include data, a read and write, write enable and clock. There are decoder and multiplexer also to specify targeted single storage and get data from targeted storage.

The internal registers of a microprocessor characterize its architecture. For example, a 32-bit microprocessor has (mostly) 32-bit registers internally. Moving data among these registers is the single most frequent operation that takes place in a computer. In this session, we will construct a 4x4 "register file" comprising registers R0, R1, R2 and R3 to demonstrate the concept of register transfer logic.
We will also implement a simple arithmetic and logic unit (ALU) and then combine the ALU and the register file to construct a simple computer Datapath.

When we give command or instructions to a computer, it always do two things- 1. Instruction fetch, 2. Instruction execute. After fetching an instruction then it executes it. In Instruction fetch-The Control Unit generates the control signals that copy an instruction byte from the memory into the Instruction Register, IR. The address of this instruction is in the Program Counter, PC. In the instruction fetch unit, there is a counter for track the instruction.

Circuit Diagrams:

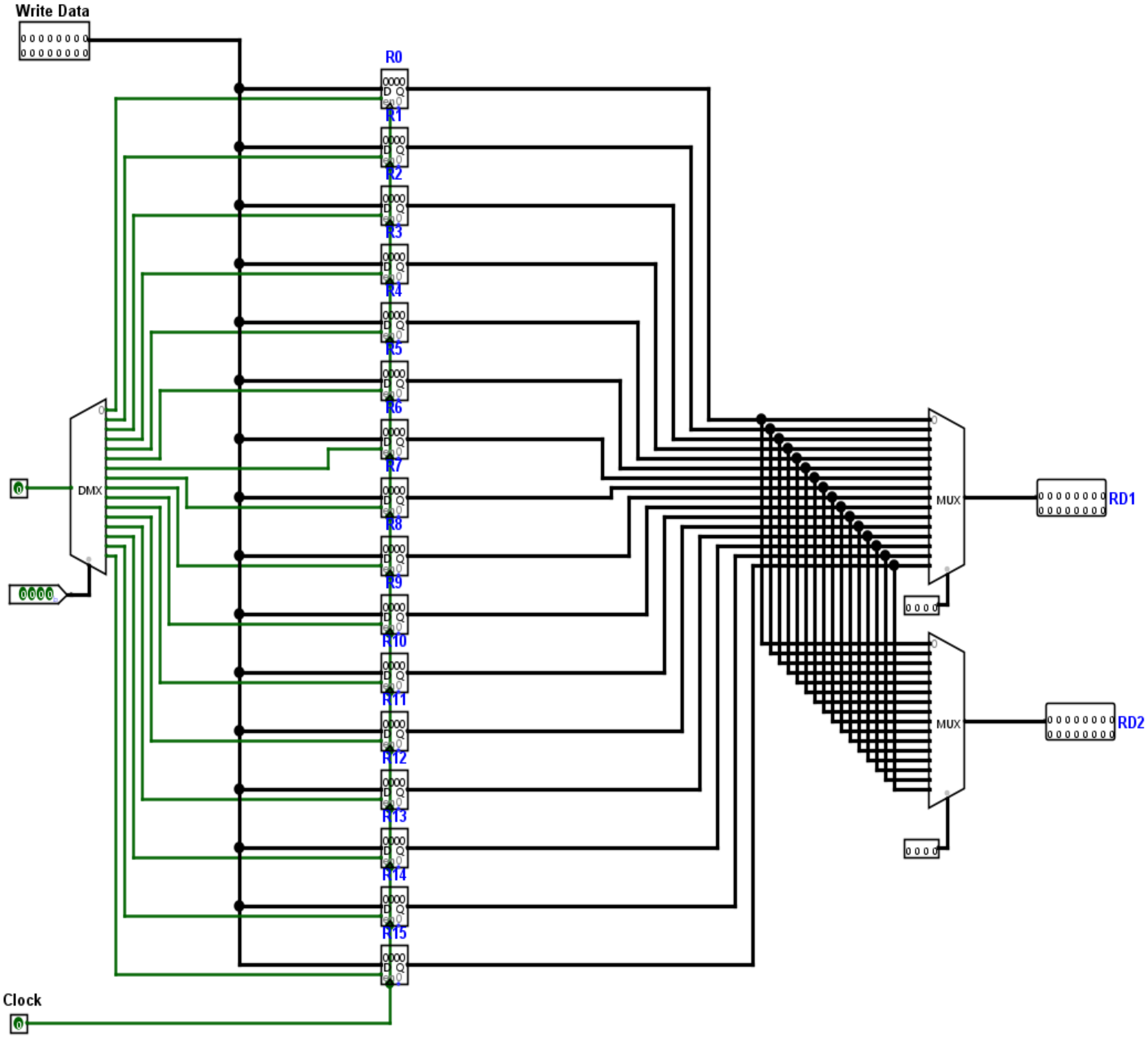

Figure: 16-Bit Register File

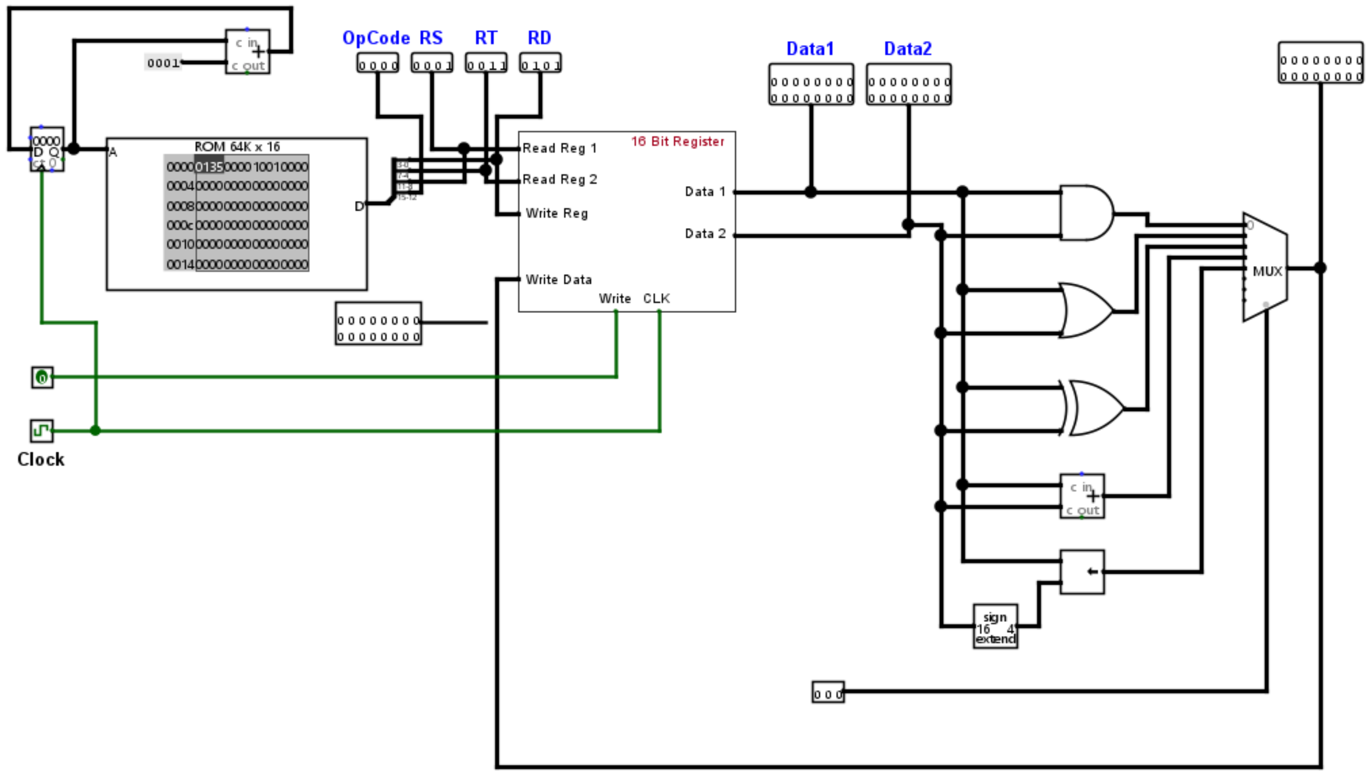

Figure: R-Type Datapath

Discussion:

In lab, we have designed a register file and an interaction fetching R-type data path. First, we have made a register file IC then the data fetching unit.

In register file- we have 16 registers named R0 - R15 and each register is 16-bit. A common clock has been connected all the registers. To update a particular register, we have used a decoder to specify the register. Then we have a 16-bit input labeled as writing data will be stored in that particular register. We have used an input labeled as Enable with decoders which controls the writing mode. Decoder indicates a specific register and if the writing enable input is true, then we can update the register. Then we have two 16 to 1 Mux to read two data (RS, RT). Mux outputs are 16-bit long and shows the stored data in particular registers by 4-bit selector.

In R-type data path - we have a single register in the beginning which will compute the instructions count. And next we have a ROM storage for storing multiple instructions. Then we have split the instruction into OP-code, RS, RT, RD and connected them to the register file inputs sequentially. Then we have other inputs connected to register file like clock, writing enable, writing data all. Then we have connected two 16-bit outputs to the register file labeled as Data1, Data2.

We have used 4 extra outputs for RS, RT, RD, OP-code Data Testing. While testing we have used the R-type data Path circuit made sure the register files are empty. Thus Data-1, Data-2 is showing 16-bit zero. So, we have to write into the register file and then we can read the data. If we want to write 1010 into the 5th register, we have to set Writing data to 1010 and made the Enable input 1. In ROM- let's write in first cell-1234. Here Op-code is 1, RS is 2, RT is 3, RD is 4. Writing data 1010 will write into the RD- 5th register means R4 register. For check, we can see 0,1,2,3 into RS, RT, RD, Op-code sequentially. We will give a clock-pulse to the it and, 1010 will be written into the 5th register. Now ROM address is indicating into the second row. And we can also see that instruction count is showing 1, means one instruction executed. Now if we want to read data from RD to RS and RT, we have to select ROM address to 0440.After that if we give clock pulse and Enable input is set 0, we will see that Data-1 and Data-2 is showing 1010 input.

If we want to read data to data1 only we will select the ROM address as 0400 and give clock pulse. The data will be read by data1 only.

This is how we have learned to create the register file and how it works also how to fetch instruction using R-type Datapath.