North South University

Department of Electrical & Computer Engineering

Lab Report

Experiment No:	6
Experiment Title:	Building a Single Cycle Data path
Course Code:	**CSE332L**
Course Name:	**Computer Organization & Architecture Lab**
Name & ID:	Md. Rifat Ahmed ~ 1931725042
Date of Experiment:	7/12/2021
Date of Submission:	8/12/2021

Objectives:

* Our objective in this experiment is to build the instruction fetch unit then add the pre-viously built Register file and ALU with it and then execute some R-type commands.

Equipment List:

* Logisim Tools

Block Diagram:

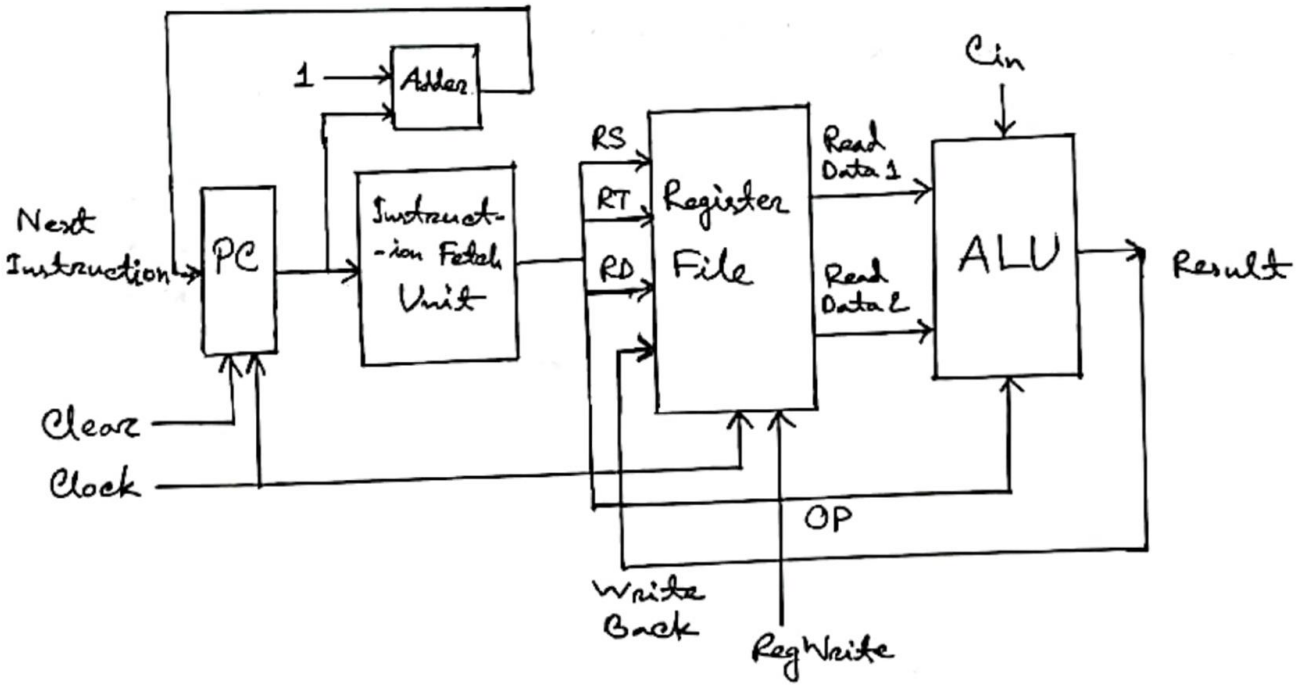

Figure-1: Single Cycle Datapath

Tables:

Our Register file is of 16-bit ISA with the following fields where the formats of the R-type instruction are:

OP (4-bit)	Rs (4-bit)	Rt (4-bit)	Rd (4-bit)

Register Table:

Register Serial	Register No. (In Binary)	Register Name
$0	0000	$zero
$1	0001	$at
$2	0010	$v0
$3	0011	$a0
$4	0100	$a1
$5	0101	$t0
$6	0110	$t1
$7	0111	$t2
$8	1000	$t3
$9	1001	$s0
$10	1010	$s1
$11	1011	$s2
$12	1100	$s3
$13	1101	$k0
$14	1110	$gp
$15	1111	$ra

OP Code Table:

S1	S0	Cin	Microopertion
0	0	0	**Add**
0	0	1	**Add with Carry**
0	1	0	**Subtract with Borrow**
0	1	1	**Subtract**
1	0	0	**Transfer A** A1 A0 + 0 0 + 0 = Transfer A
1	0	1	**Increment A** A1 A0 + 0 0 + 1 = Increment A
1	1	0	**Decrement A** A1 A0 + 1 1 + 0 = Decrement A
1	1	1	**Transfer A** A1 A0 + 1 1 + 1 = Transfer A

Circuit Diagrams:

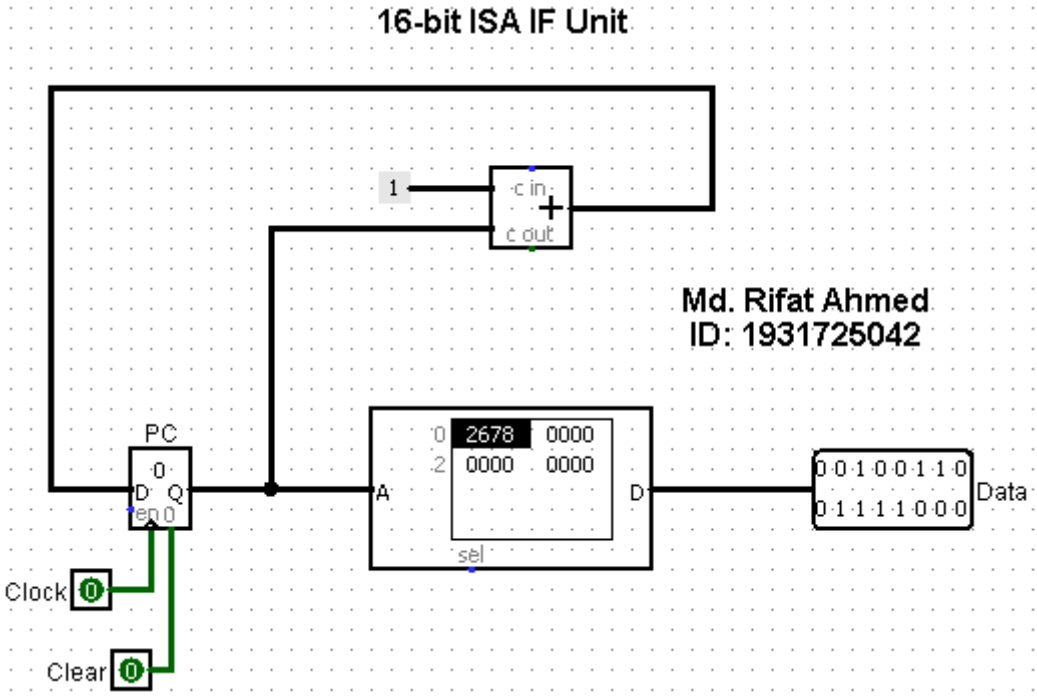

Figure – 2: 16-bit ISA Instruction Fetch (IF) Unit

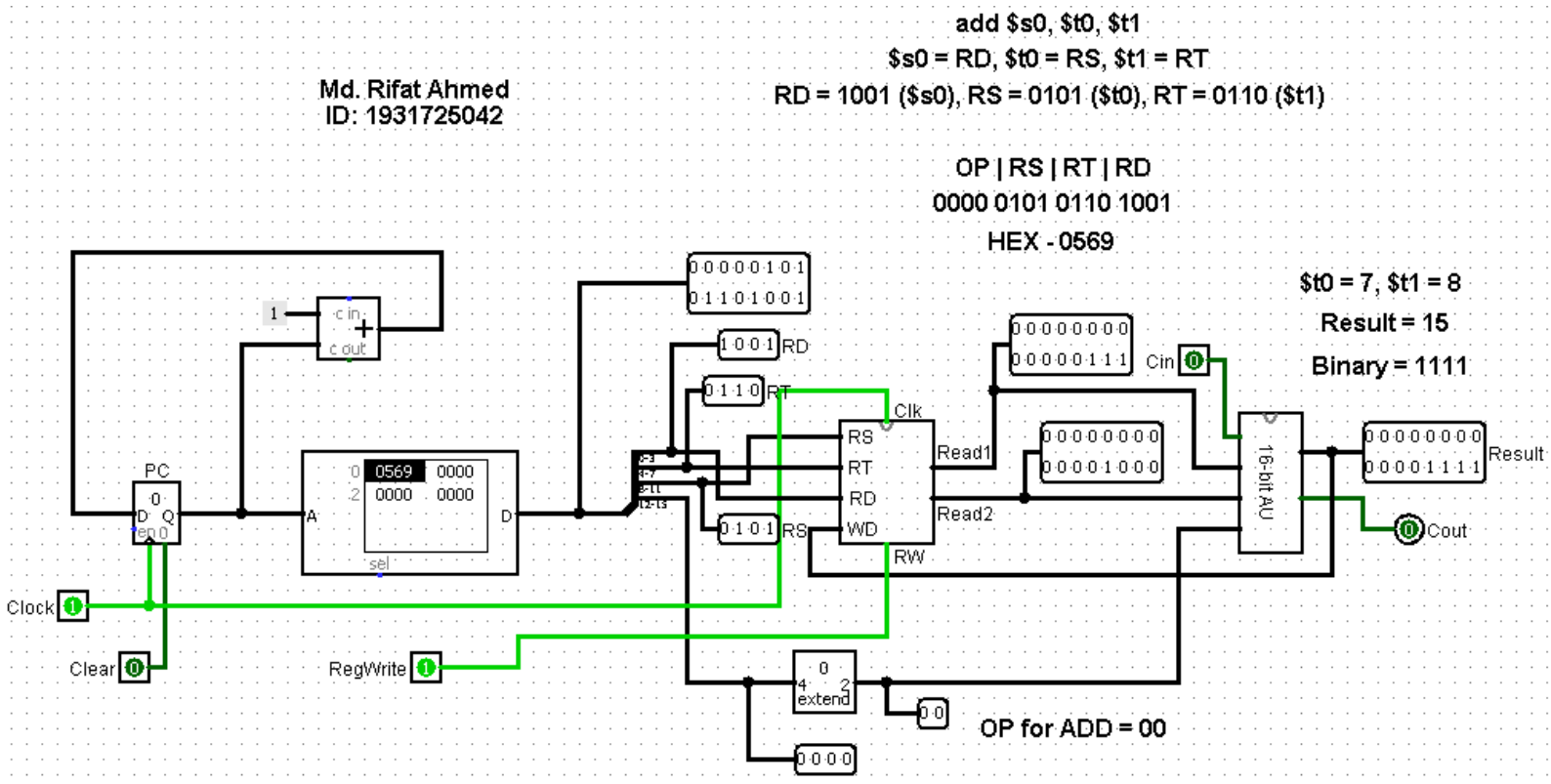

Figure – 3: IF, ID, ExE for add $s0, $t0, $t1 where $t0 = 7 & $t1 = 8

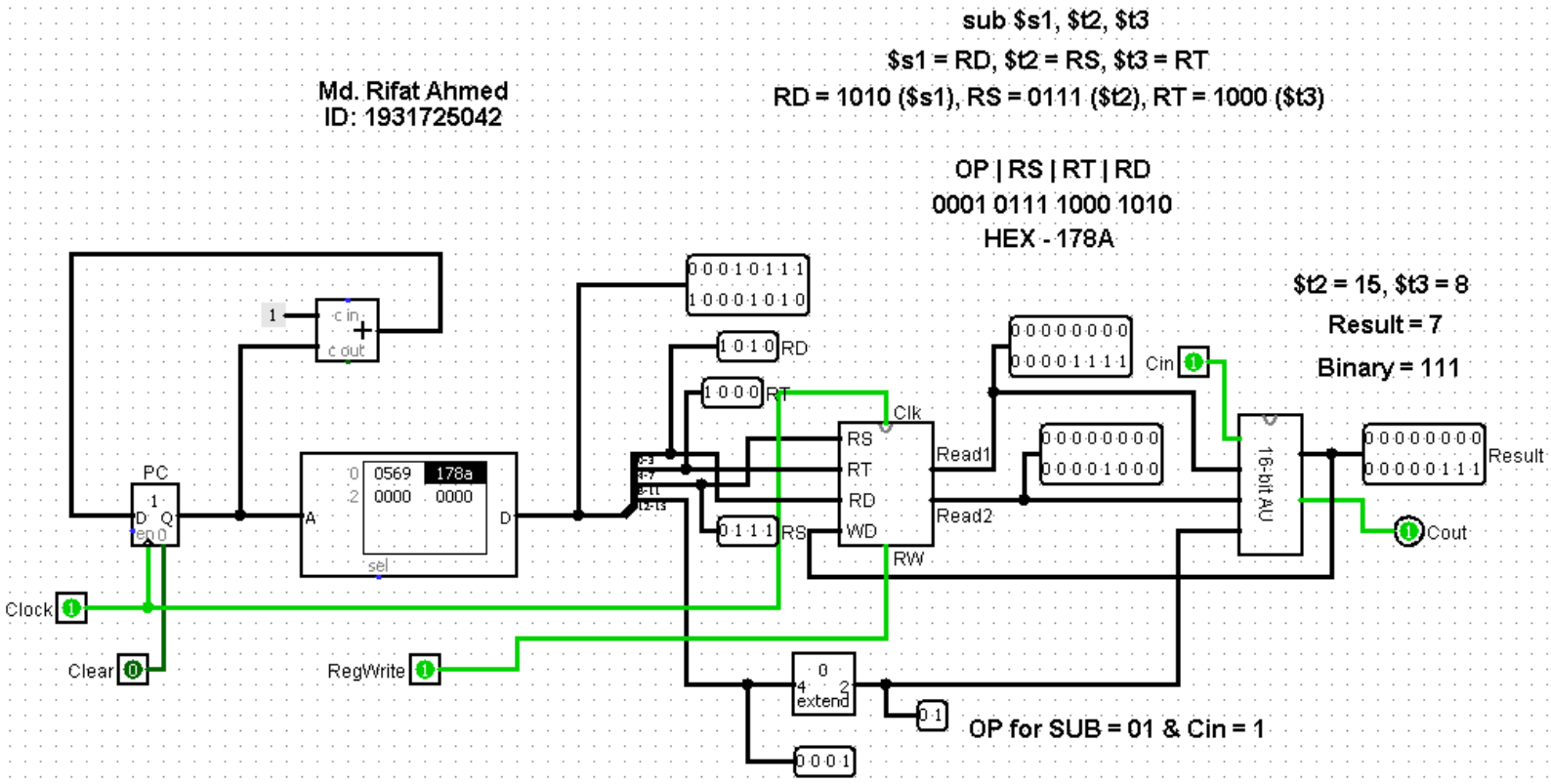

Figure – 4: IF, ID, ExE for sub $s1, $t2, $t3 where $t2 = 15 & $t3 = 8

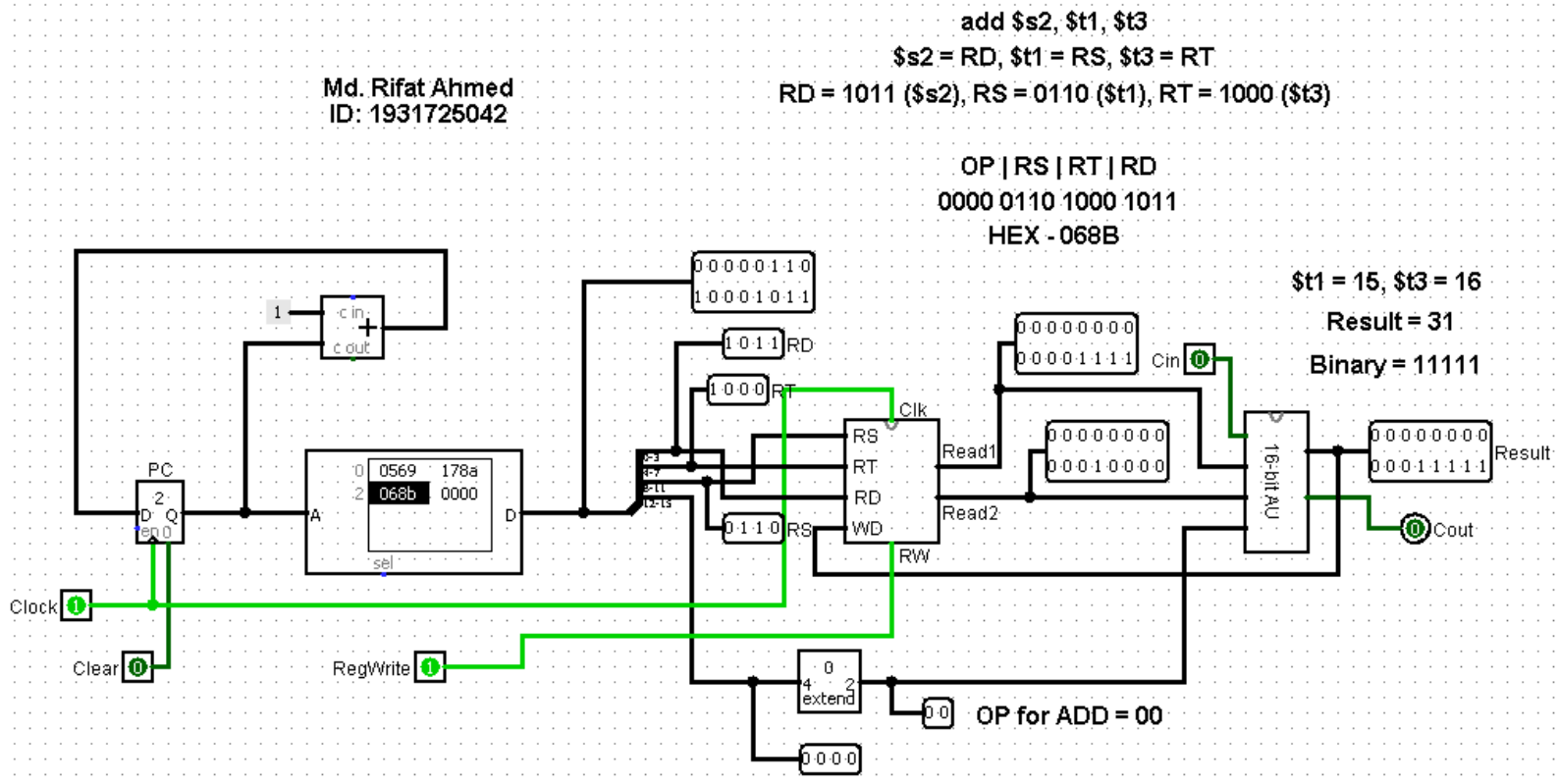

Figure – 5: IF, ID, ExE for add $s2, $t1, $t3 where $t1 = 15 & $t3 = 16

Discussion:

In this experiment we learned how to build a single cycle datapath using Instruction Fetch Unit, Register File and ALU. At first we took a ROM adjusted its Address bits and data bits width. Then took a register to use as the Program Counter and added it to the input of the ROM. Then also took the output of the PC to an adder and added it with 1 and then took the output of the adder into the input of the PC, this'll work to cycle between instructions and thus our IF unit is ready. Then we added the output of the ROM and splitted it into RS, RT, RD and OP then we took an ALU added it's 2 input from the Read data 1 and Read data 2 of the Register file. Then we took the output result of the ALU and added it into the Write Data pin of the Register file. Then we added clock to the PC and Register file added Cin to the ALU. But here we had to use use a bit extender to turn the OP from IF to 2 bits from 4-bits as our ALU OP is of 2-bits. And finally we got our single cycle datapath.