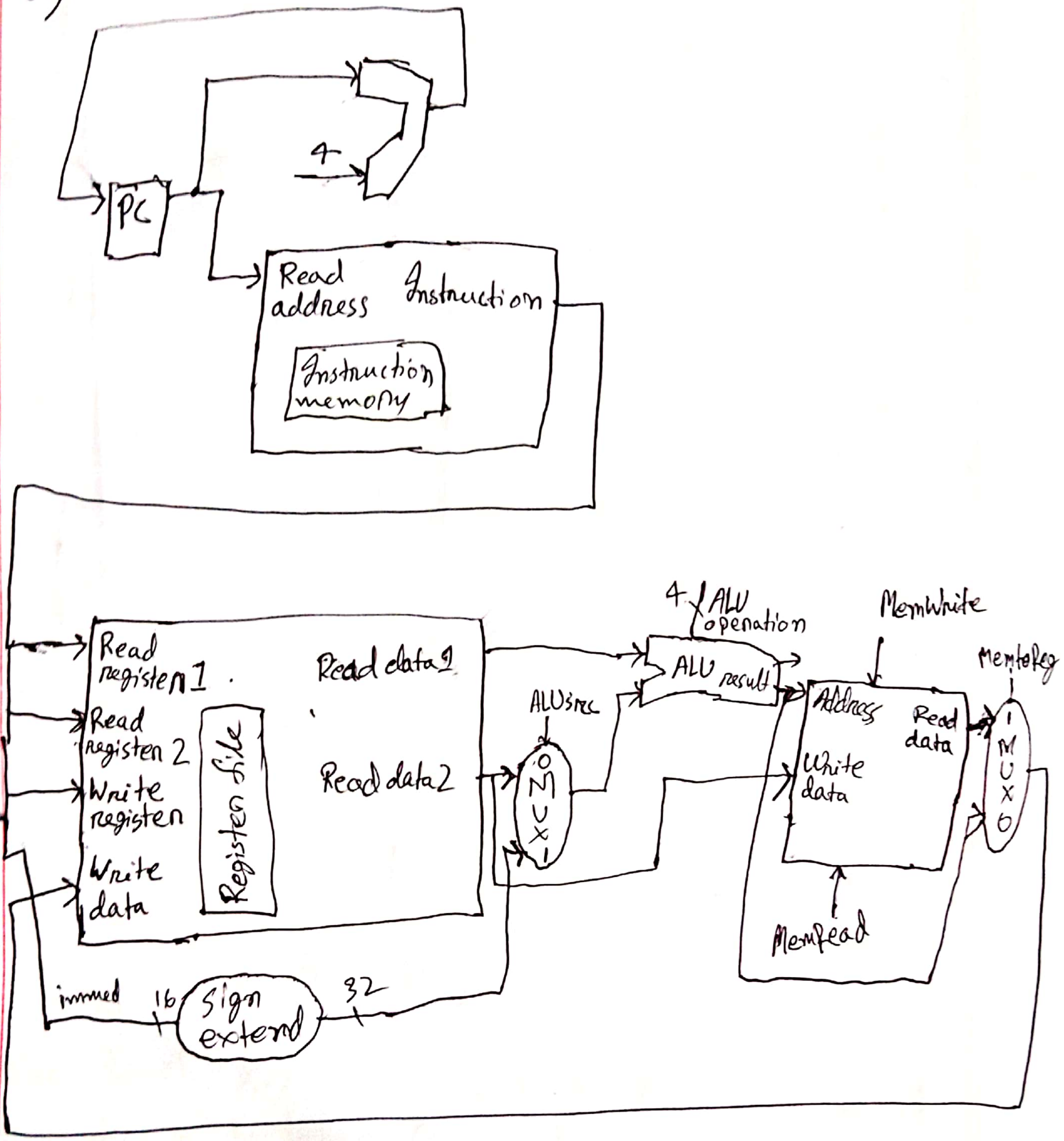
Name: Rashik Rahman
ID : 17201012

ii

a)



~~A+R~~ R[i] = R[i]+Y

MIPS instruction:

~~(i) lw `$t0`, 48(`$s0`) [for i = 12]~~

Here,
i=12
R=`$s0`
Y=`$s1`

`$s0` & `$s1` number is 16 & 17 respectively. `$t0` number is 8
So mips instruction is,

```
(1) lw $t0, 48($s0)
(2) add $t0, $t0, $s1
(3) sw $t0, 48($s0)
```

Explaination:

For 1st instruction:

It's an I type instruction, a load/Read instruction. Here the number of `$s0` that is 16 will be passed to read register 1. Thus the content of `$s0` that is the base address will come out.

from read data 1 and will goto ALU. Now the offset is 18 this ~~is~~ in 16 bit by using sign extend we convert it to 32 bit by extending the MSB. To indicate it's an Itype instruction ALUsrc is SET. So the offset will be passed through the MUX and will goto ALU. ALU will add base address to offset and output a physical address throug ALU result. This address will goto address point of Data memory and MemRead & MemtoReg will be set. So ~~data me~~ data corresponding ~~th~~ to that address will fetched and MUX will pass ~~return~~ the data to write data point of the register file. To save it to $t0, $t0's number that is 8 will be passed to write register. Thus the data would be loaded on $t0

for 2nd instruction

This ~~da~~ is a Rtype instruction. To indicate this Rtype ~~int~~ instruction ALUsrc MUX will be CLEAR. ~~Thus th~~ It will work like, $t0 & $s1 register numbers 8 & 17 ~~w~~ respectively will goto read register 1 & read registe 2 respectively. So the ~~data~~ content of these two registers will come out through Redd data 1 and read data 2 respectively and as the ALUsrc is CLEAR so these two contents will be passed to ALU. ALU will ~~add~~ do add operation ~~in~~ and the result will be directly pass throug MUX and goto write data point in the register file as the MemWrite, MemRead & MemtoReg all are CLEAR. To save the result in $t0, $t0's number that is 8 would be passed to write register. Thus the result of the add operation is saved in $t0.

for the 3rd instruction:

Again it's an Itype instruction but ~~a~~ its now a save instruction. This will work like,

The number of ~~t_0 & S_0 will be p~~ S_0 & t_0 ~~will be passed~~ that is 16 & 8 would be passed to Read register 1 and read register 2 respectively. The content of ~~the~~ S_0 ~~will co~~ that is the base address will come out of read register 1 and would goto ALU. ALUsrc is SET so the offset will be extended to 32 bit and as ALUsrc is ~~SET s~~ SET so it would ~~be passed thro~~ pass through the MUX and will goto ALU. ALU will add base address with offset and thus ~~we~~ we get physical address from ALU result.

The physical address now will goto address point in data memory and the result of the add operation will come ~~of~~ out of read data 2 and will directly will goto write data point in data memory. Now only MemWrite is SET so it'll write the received data to that physical address. This is how data is stored in memory.