

# CS-321

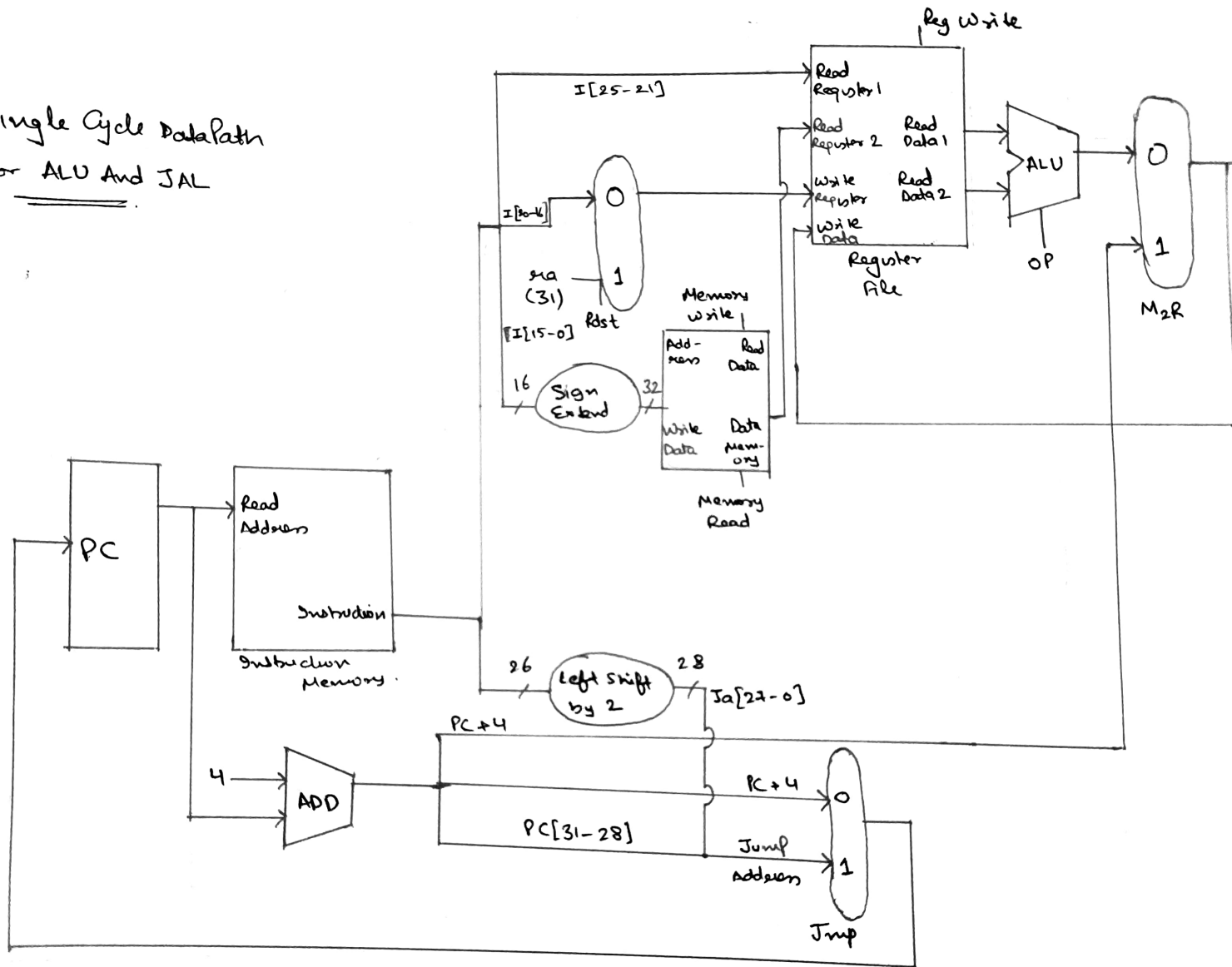
## MID - SEMESTER

- TARUSI MITTAL

- 1901CS65

- Panur Mittal

# Single Cycle DataPath for ALU And JAL



(i)  $op, \$R_1, \$R_2, [M]$

Ans:  $\Rightarrow R_1 = R_2 \text{ op } [M]$

$\rightarrow R_1 = 5 \text{ bit}, R_2 = 5 \text{ bit}, \text{ op code} = 6 \text{ bit}, M = 16 \text{ bit}$

Now when the above instruction will be received the following steps will be there.

a) As  $R_1 = R_2 \text{ op } [M]$

$\Rightarrow R_1$  is our final destination register. Also both  $R_1$  and  $R_2$  can be accessed from the Register file directly without any extra operation.

b) In the ALU, unit, we need to select the operation that needs to be performed which will be decided by the op code control line.

c) Now the data needs to be extracted from the Data Memory Unit. And as per the given instruction  $[M]$  is our memory address.

$\Rightarrow$  During the instruction, the op code control line in MU will define the operation performed in MU.

Also; Memory write = 0

Memory read = 1

Rdst = 0

Mrk = 0

Reg Write = 1

Jump = 0.

(ii) JAL Off.

Ans: Off = 26 bit, op code = 6 bit.

For implementing JAL the following steps need to be performed.

- a) First of all, as the offset is initially 26 bits we need to get a 32 bit PC from it. So, first the 26 bit offset will be extracted and 28 bits will be created out of it by shifting it to left by 2 bits. Now the top 4 bits of the current PC will be merged with it.

At the end of this step we will have our 32 bit PC.

- b) Now  $ra(31)$  will be hard coded and will be added to the Rdst MUX. This step will permit its selection. Parallely the return address which is now  $PC+4$  will be connected to the write data port in the Register File.
- c) After this we need to reach to Jump address and also it should be equal to PC. So for this the Jump MUX will be equal to 1.

⇒ During this op control line in the ALU has no function

Also:

Memory write = 0

Memory read = 0

Rdst = 1

Mzk = 1

Reg Write = 1

Jump = 1