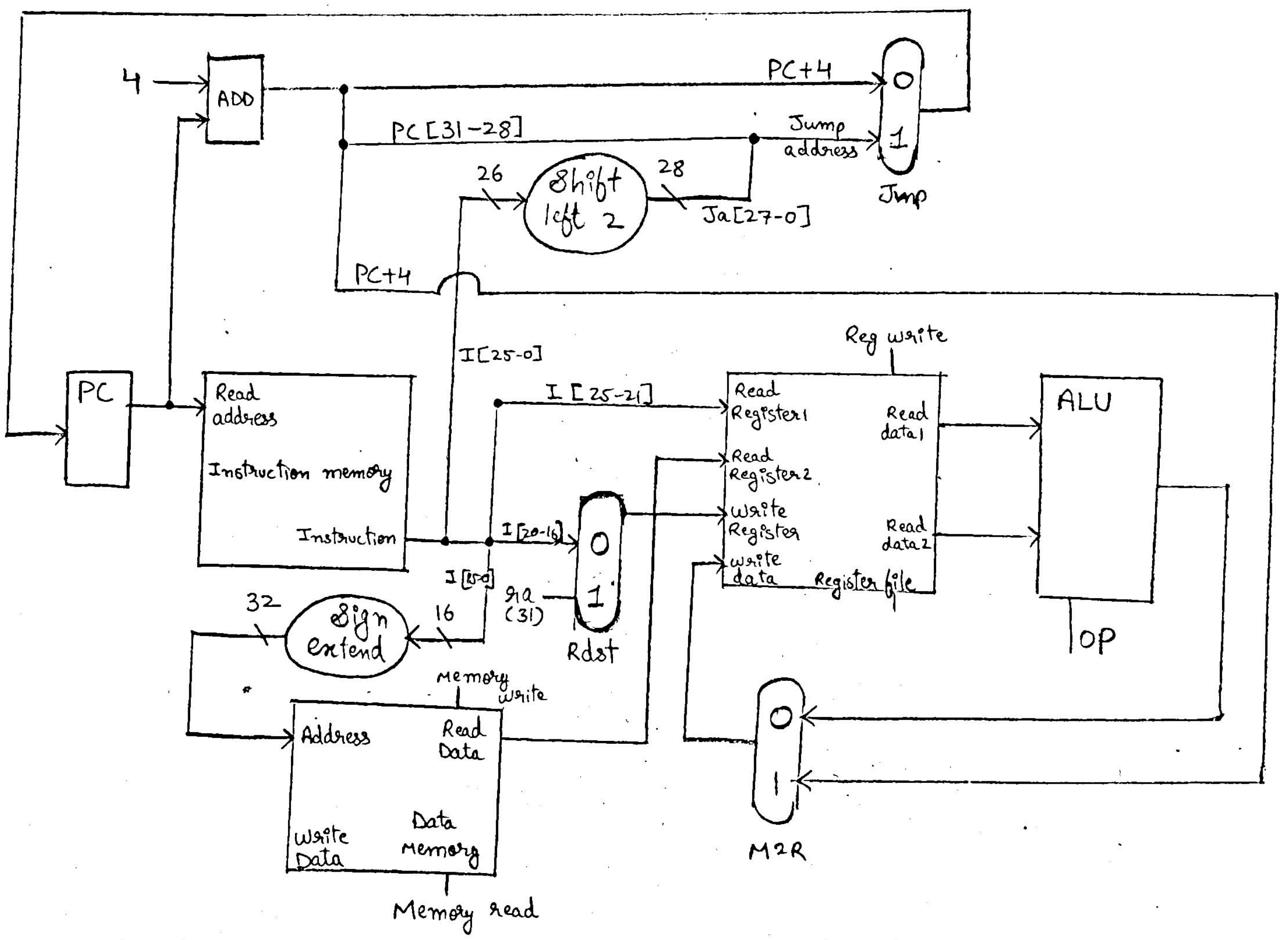
CS321 TANISHQ MALU 1901CS63



Single cycle datapath for AlU & JAL

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

Format of instruction $R_1 = R_2$ op [M]

Op code ⇒ 6 bit
R_1 ⇒ 5 bit
R_2 ⇒ 5 bit
M ⇒ 16 bit

Whenever this instruction will be received:-

(i) The opcode control line in main ALU unit will determine the ALU operation to be used

(ii) R_1 and R_2 can be directly accessed from Registers file. R_1 will be our destination register.

(iii) [M] is memory address (as given). Thus first the data needs to be fetched from Data memory unit.

Control line during this instruction ⇒

Memory read = 1	Memory write = 0
Rdst = 0	Reg write = 1
M2R = 0	Jmp = 0

Op line in Alu will define Alu operation

(ii) Jal off ⇒ op ⇒ 6 bit off ⇒ 26 bit

(i) To implement Jal, firstly the 26 bit offset needs to be extracted and shifted to the left by 2 bits to Create 28 bits. To get a 32 bit PC, Top 4 bits of current Pc are merged with it.

(ii) Now, the return address (Pc+4) needs to be Connected to write data port of register file. However, ra(31) needs to be "hard-coded" and added in Rdst mux to permit its selection.

(iii) Further the jmp ~~regsel~~ mux would be 1, to ~~m~~ reach to jump address and make pc equal to it.

<u>control lines</u>

Memory read = 0 Memory write = 0

Rdst = 1 Reg write = 1

M2R = 1 Jmp = 1

op line for AlU - doesn't matter