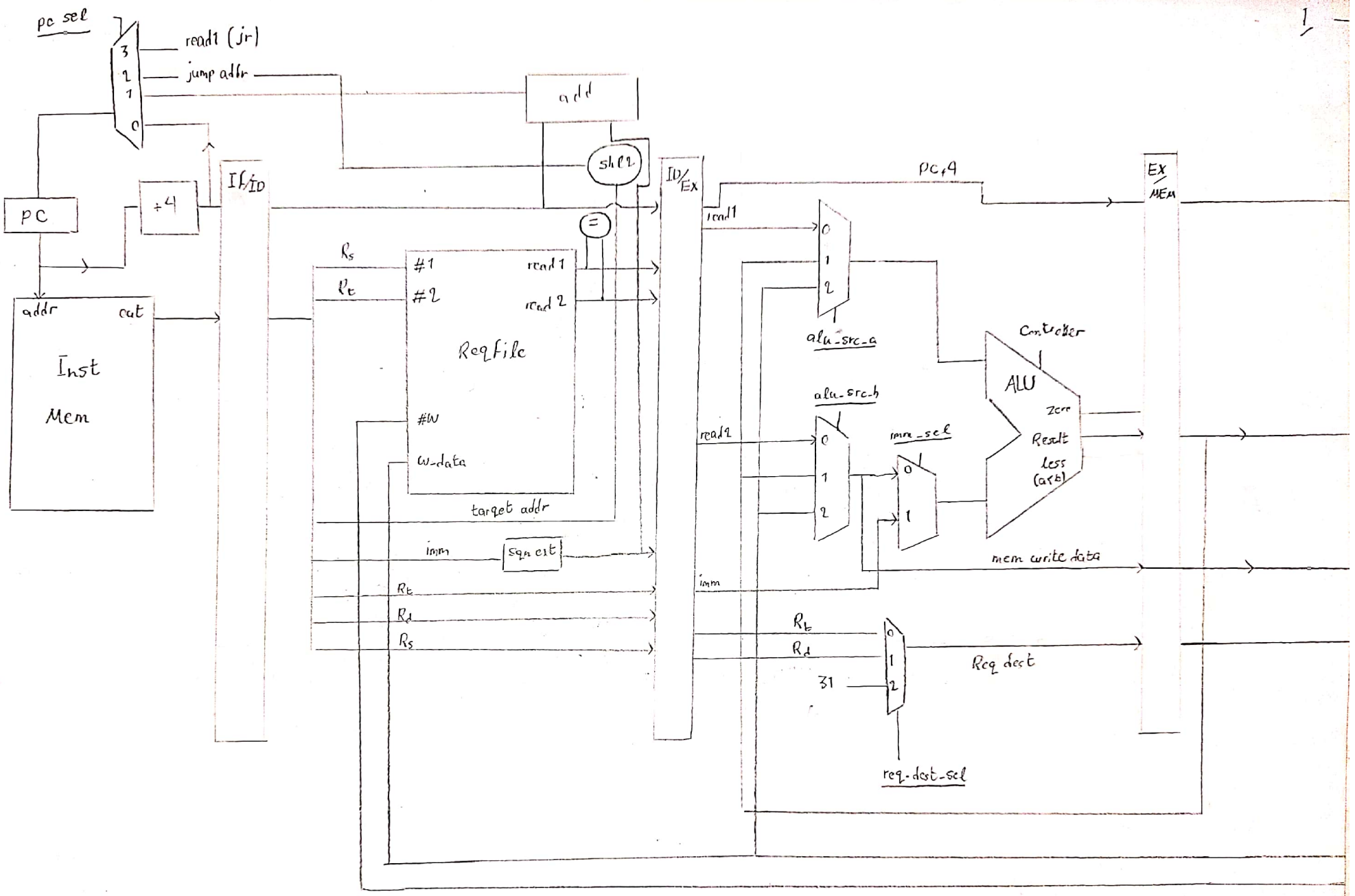
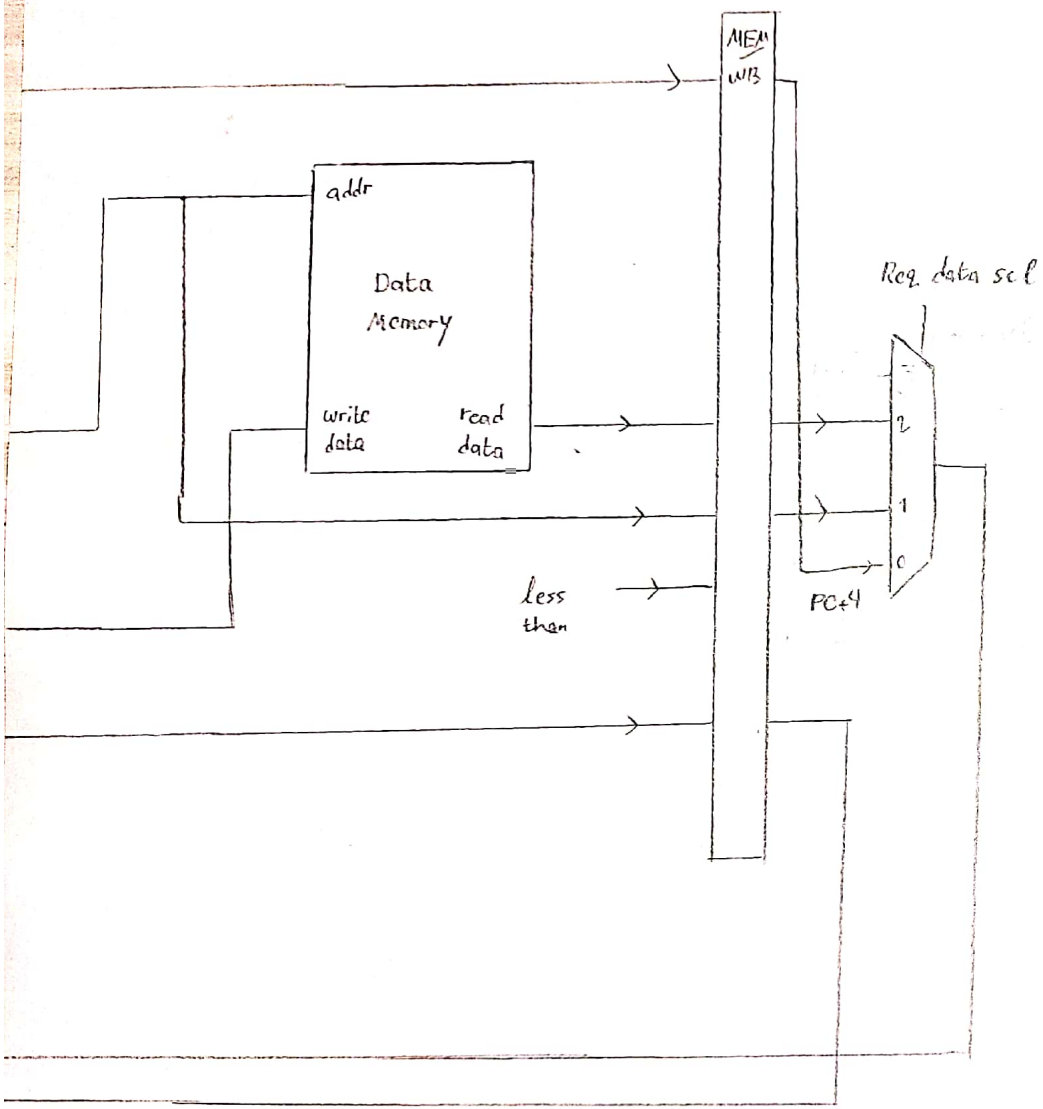
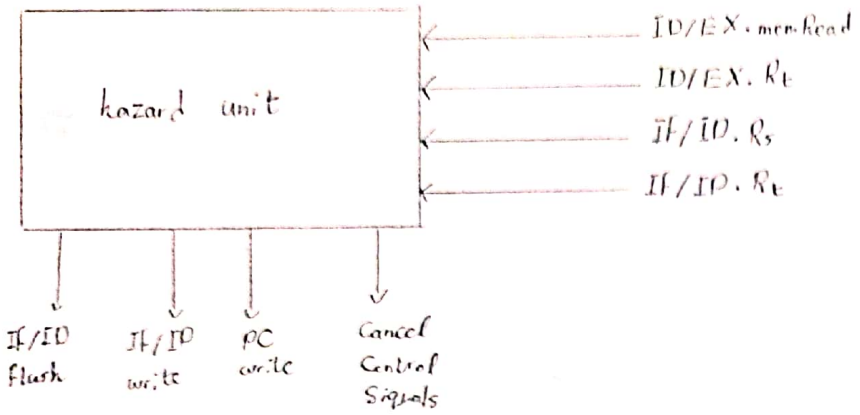
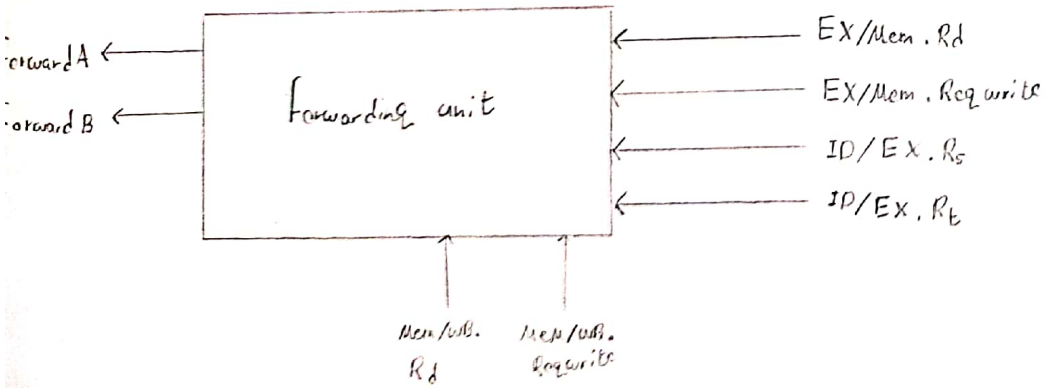






If((ID/EX.RegisterRt = IF/ID.Rs) and ID/EX.memRead)
make all ID/EX controll signals zero
PC write = 0
IF/IDwrite = 0

If(inst = branch eq) and (read1 = read2)
IF/ID flush = 1



second_forwardA = (Mem/WB.Regwrite=1) and (Mem/WB.Rd = ID/EX.Rs) and (Mem/WB.Rd ≠ 0)

third_forwardA = (EX/Mem.Regwrite=1) and (EX/Mem.Rd = ID/EX.Rs) and (EX/Mem.Rd ≠ 0) and ¬second_forward_A

if(second_forwardA) alu_src_a = 1

if(third_forwardA) alu_src_a = 2

second_forwardB = (Mem/WB.Regwrite=1) and (Mem/WB.Rd = ID/EX.Rt) and (Mem/WB.Rd ≠ 0)

third_forwardB = (EX/Mem.Regwrite=1) and (EX/Mem.Rd = ID/EX.Rt) and (EX/Mem.Rd ≠ 0) and ¬second_forwarB

IF/ID reg: { PC+4, inst_mem }

ID/EX reg: { WB Control signals, Mem Control signals, Ex Control signals, reqFile read1, reqfile read2, imm, r_t, r_d, r_s }

EX/Mem reg: { WB Control signals, Mem Control signals, ALU result, mem write data, Req dest }

Mem/WB reg: { WB Control signals, memory read data, ALU result, Req dest }