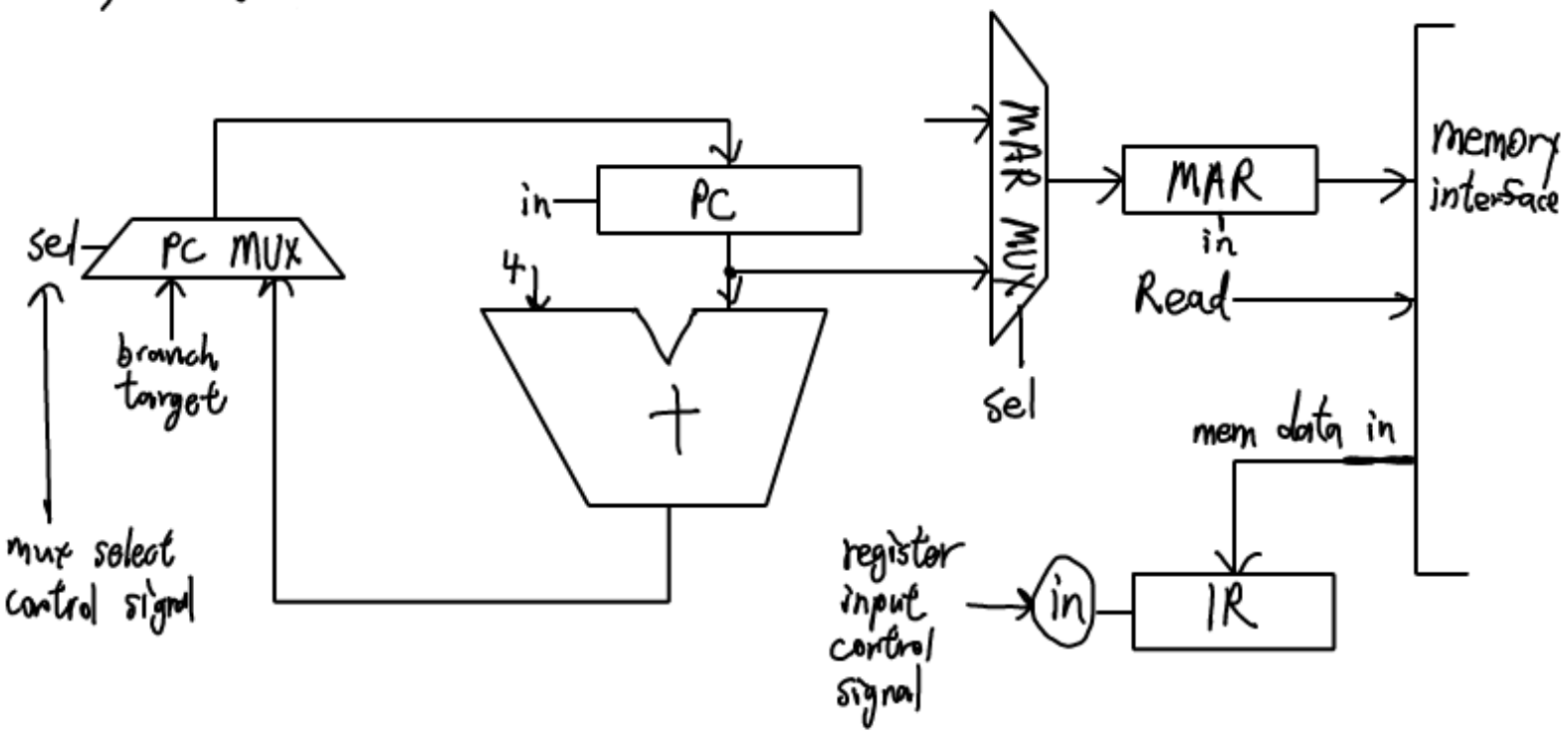
Processor Design

1) Execution Stages

stage \ type	Data Processing	Load	Store	Branch
1. fetch	ADD r2,r0,r1	LDR r1,[r0,#4]	STR r1,[r0,#4]	BEQ [PC,#16]
2. reg. read	r0, r1	r0	r0, r1	(PC)
3. use ALU	[r0]+[r1]	[r0]+4	[r0]+4	[PC]+16
4. use MEM	—	read	write	—
5. reg. write	r2←	r1←	—	if(zflag==1) PC←

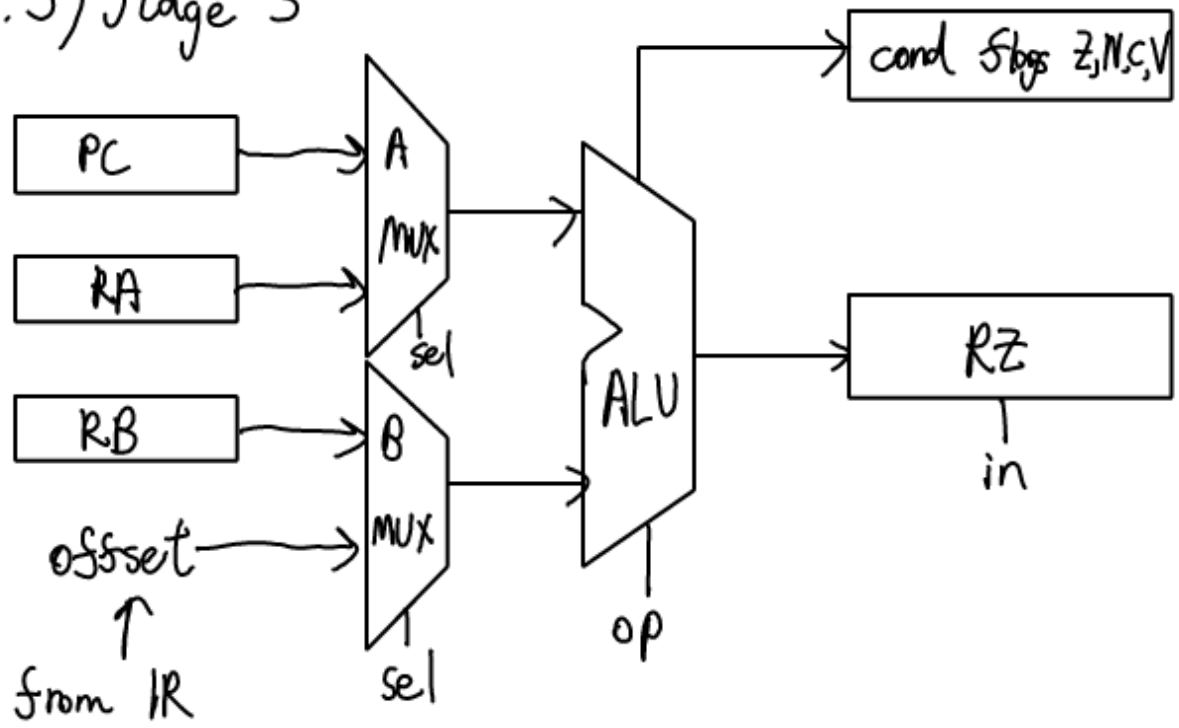
2) Datapath

2.1) Stage 1

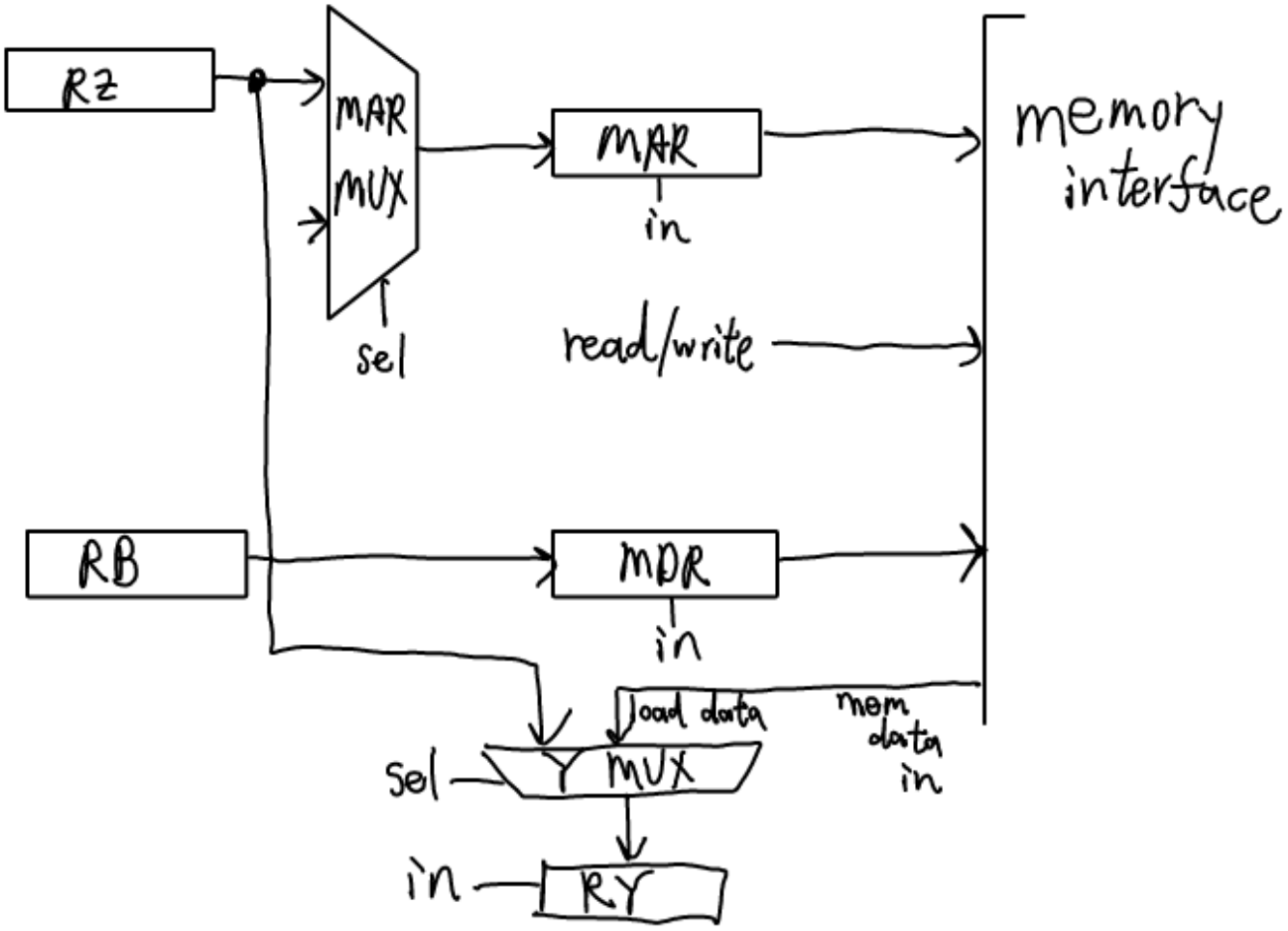


2.2) Stage 2

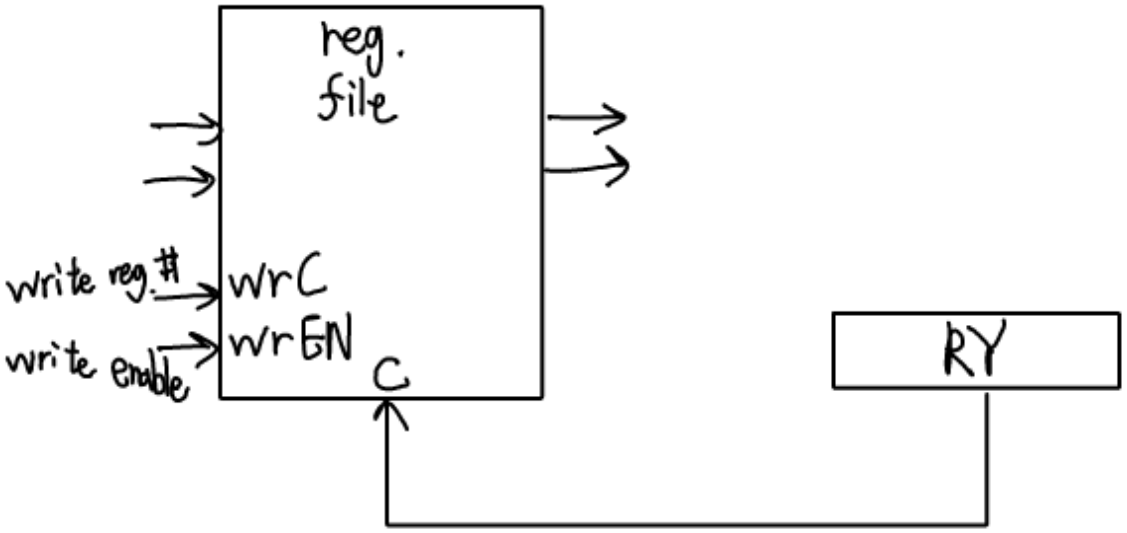
2.3) Stage 3



2.4) Stage 4



2.5) Stage 5



2.6) Combined Stages
- handout: Processor Datapath

3) Control
- registers are clocked; output is always enabled
- instructions typically spend 1 clock cycle per stage (multi-cycle processor)
- data flow is controlled by asserting register input and multiplexor select signals.
- if a control signal is omitted, it is assumed deasserted.

3.1) data processing e.g. ADD r2, r0, r1

Stage	Action	Control Signals Asserted
1. fetch	MAR ← [PC]	MAP_{sel} = PC, MAR_{in}
	read memory	read
	IR ← mem data in	IR_{in}
	PC ← [PC] + 4	PC_{sel} = inc, PC_{in}

Stage	Action	Control Signals Asserted
2. reg read	RA ← [r0] RB ← [r1]	(rdA = 0), RAin (rdB = 1), RBin
3. ALU	RZ ← [RA] + [RB]	Asel = A, Bsel = B, ALUin = add, Zin
4. mem	RY ← [RZ]	Ysel = Z, RYin
5. reg wr.	r2 ← [RY]	(wrC = 2), wrEn