

Multicycle mMIPS

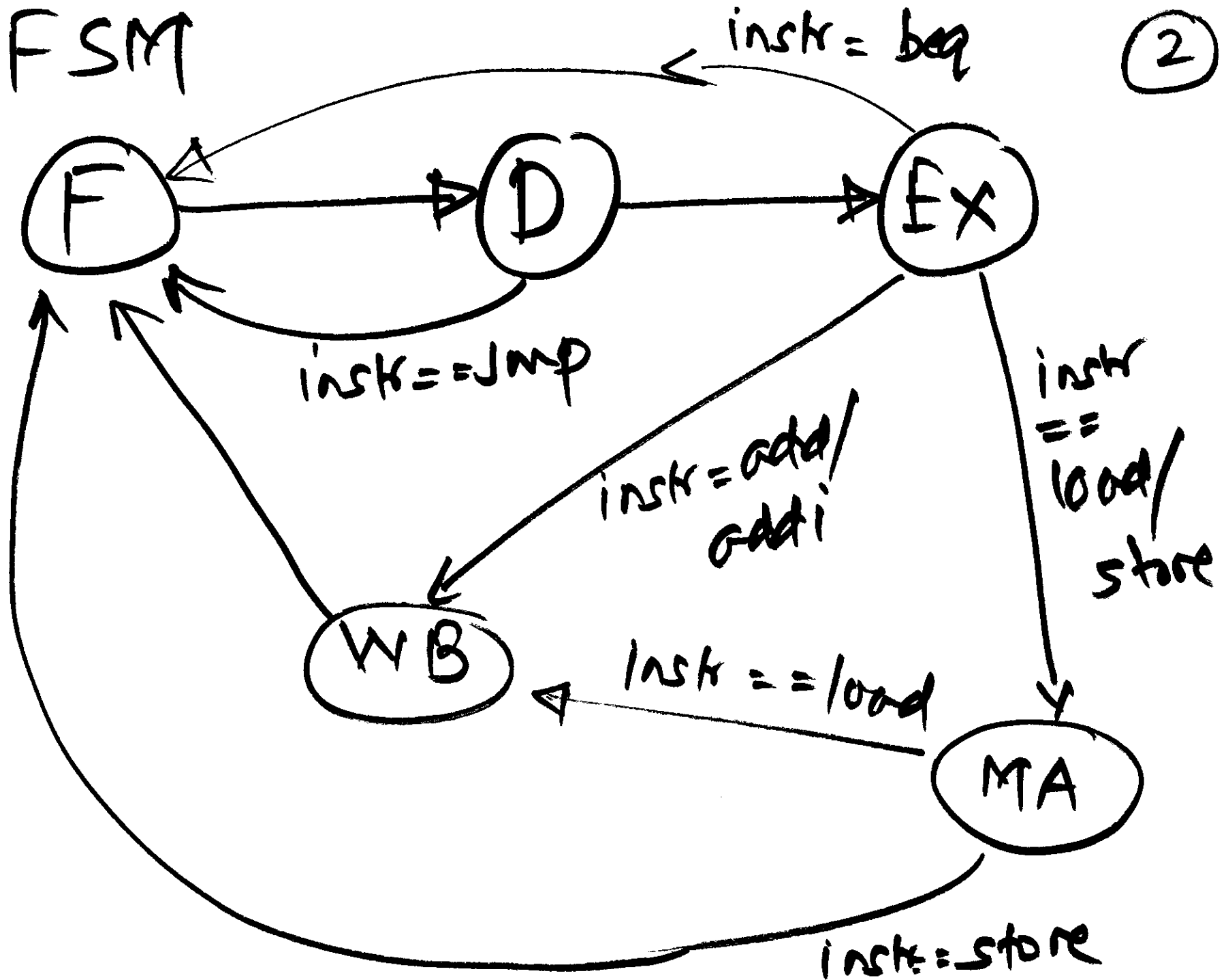
- details

- FSM

+ Datapath

FSM

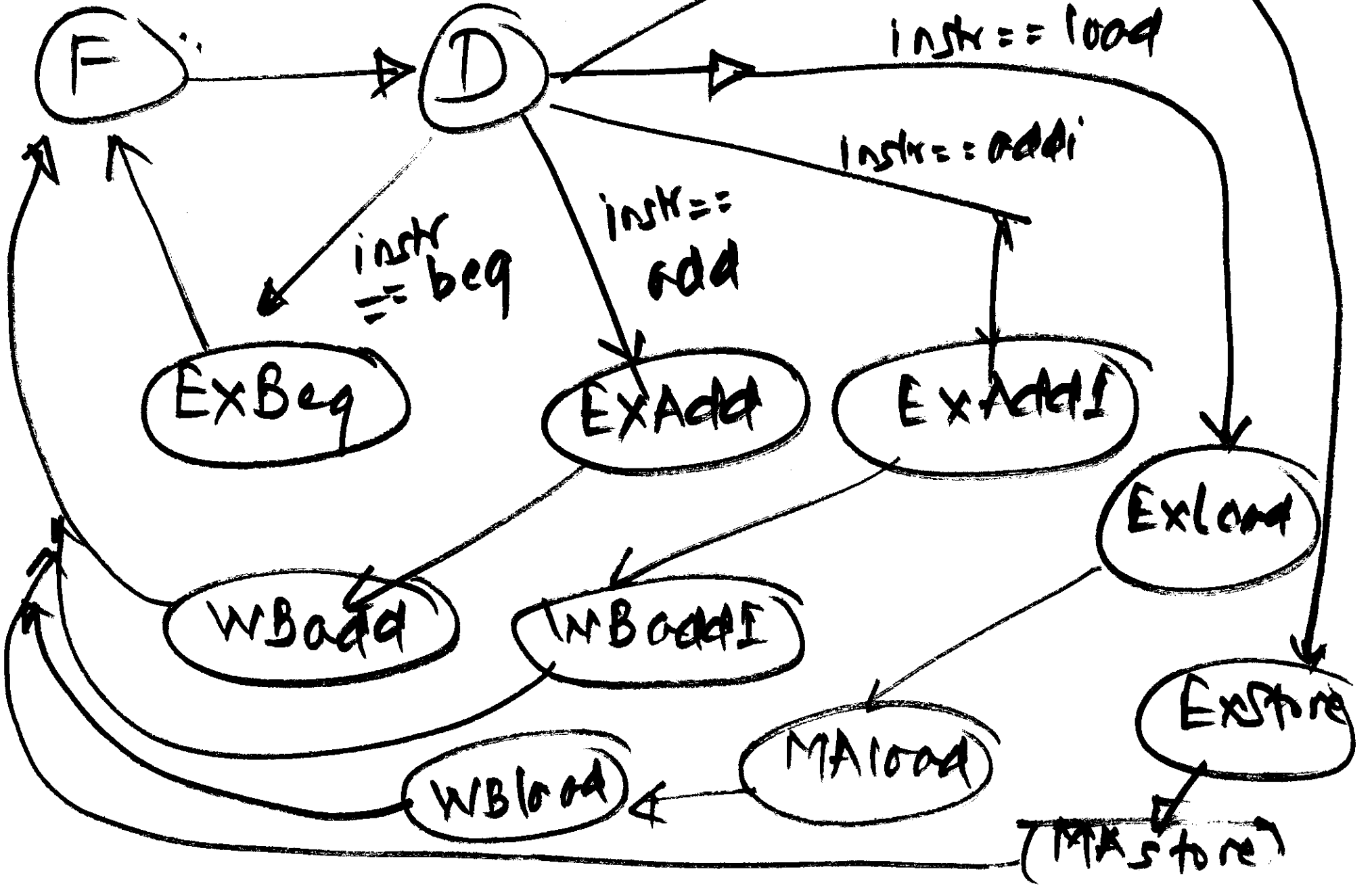
(2)



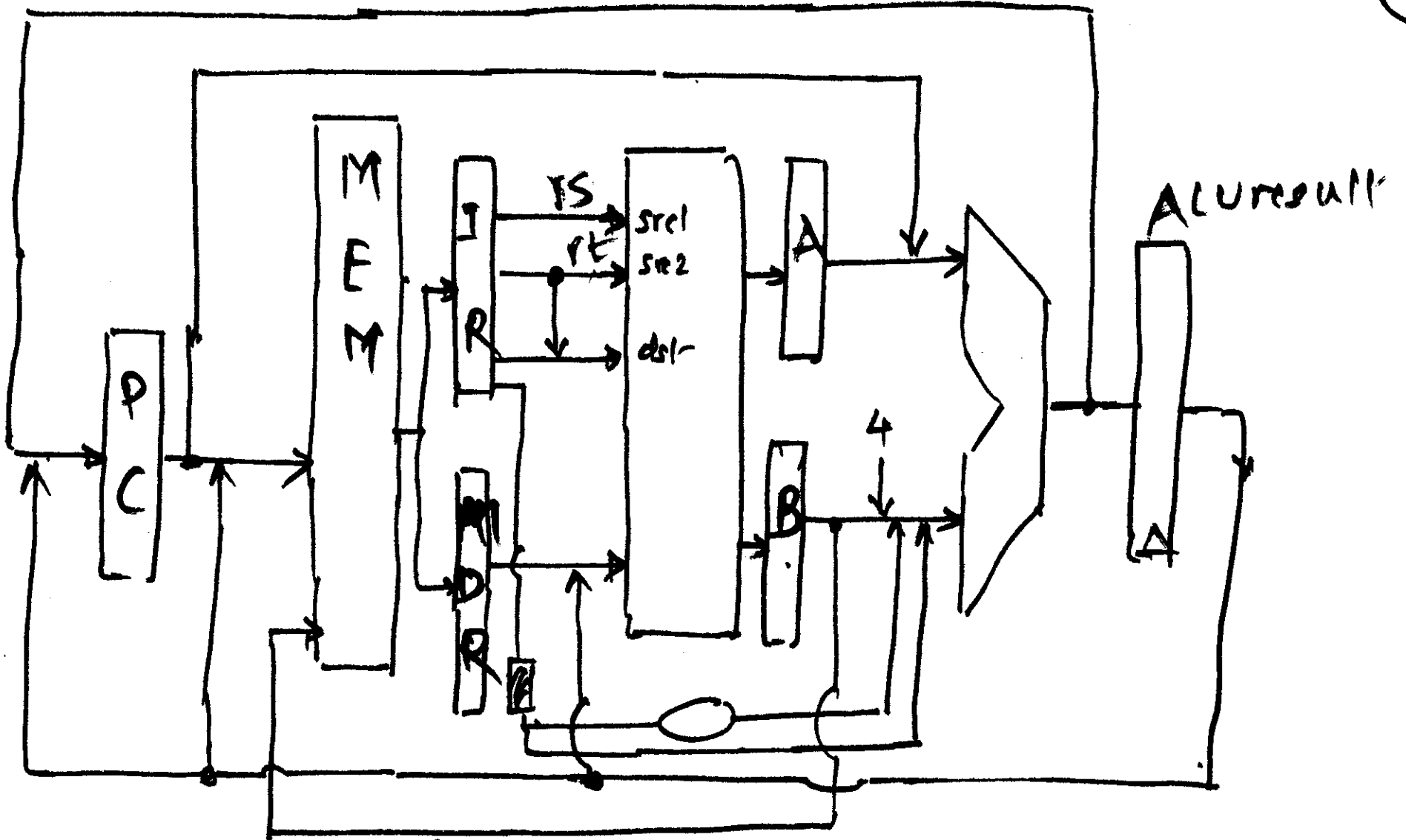
Refined FSM

instr = store

3



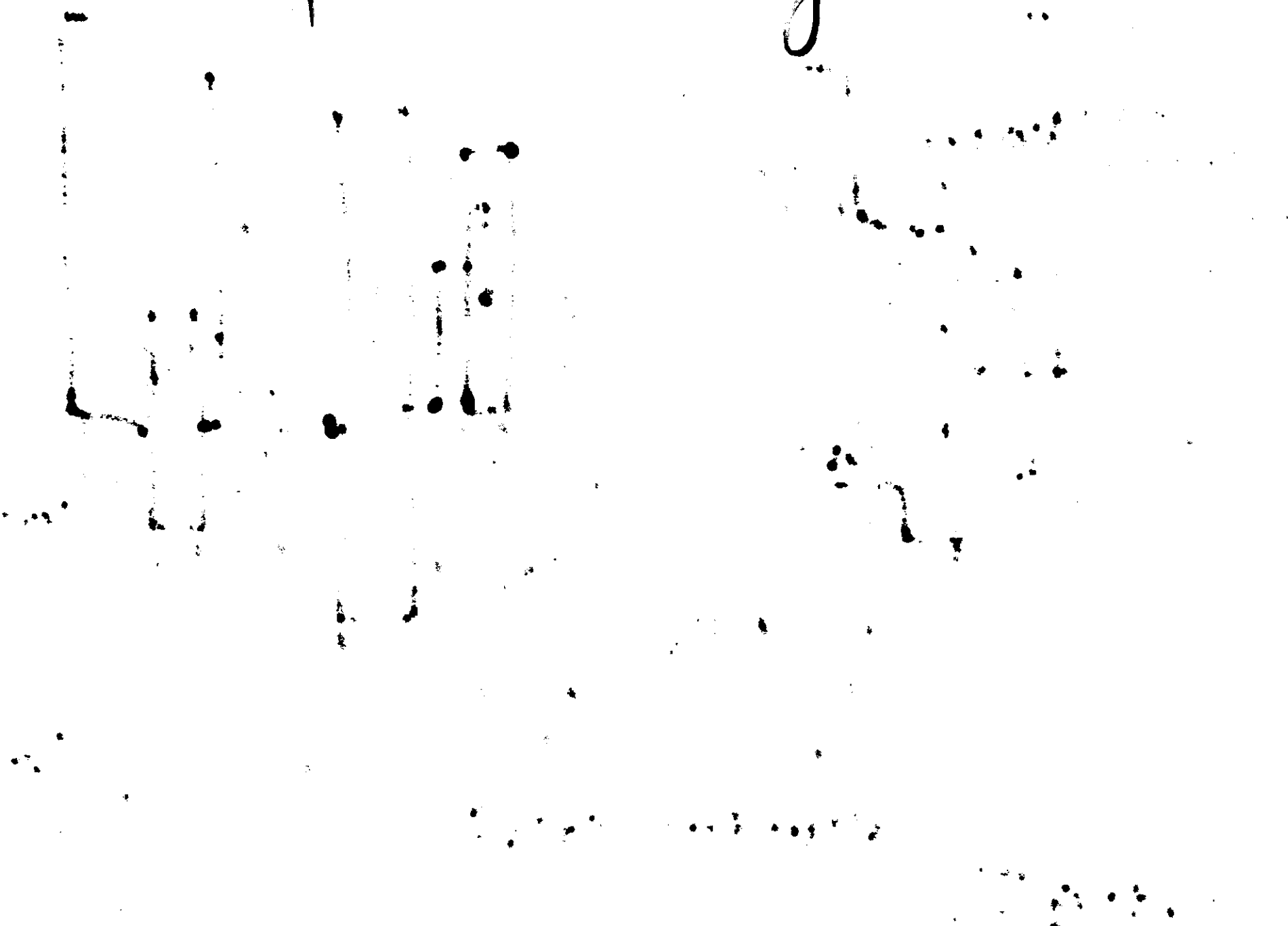
4



MAstore subcycle

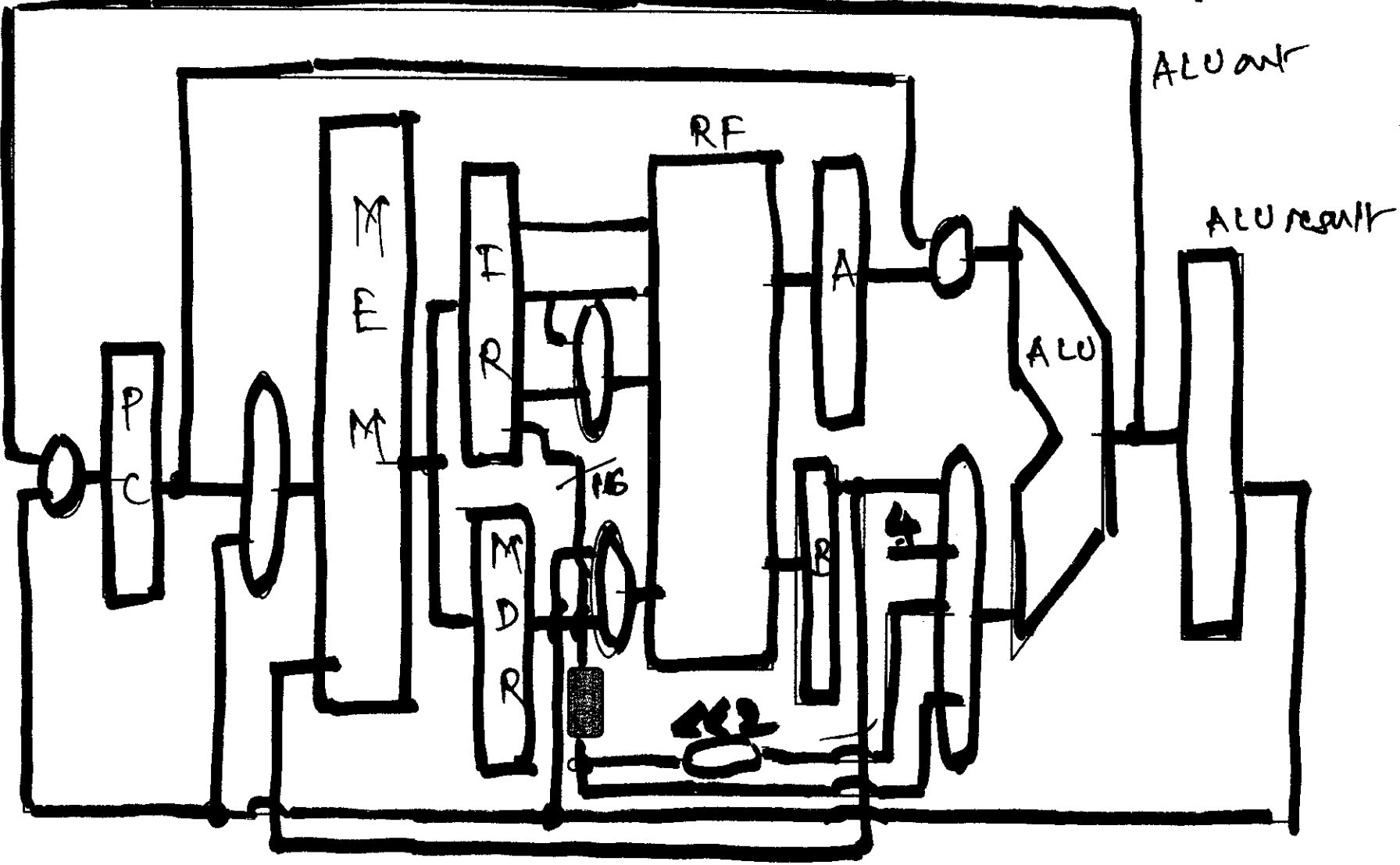
Datapath control signals

(5)



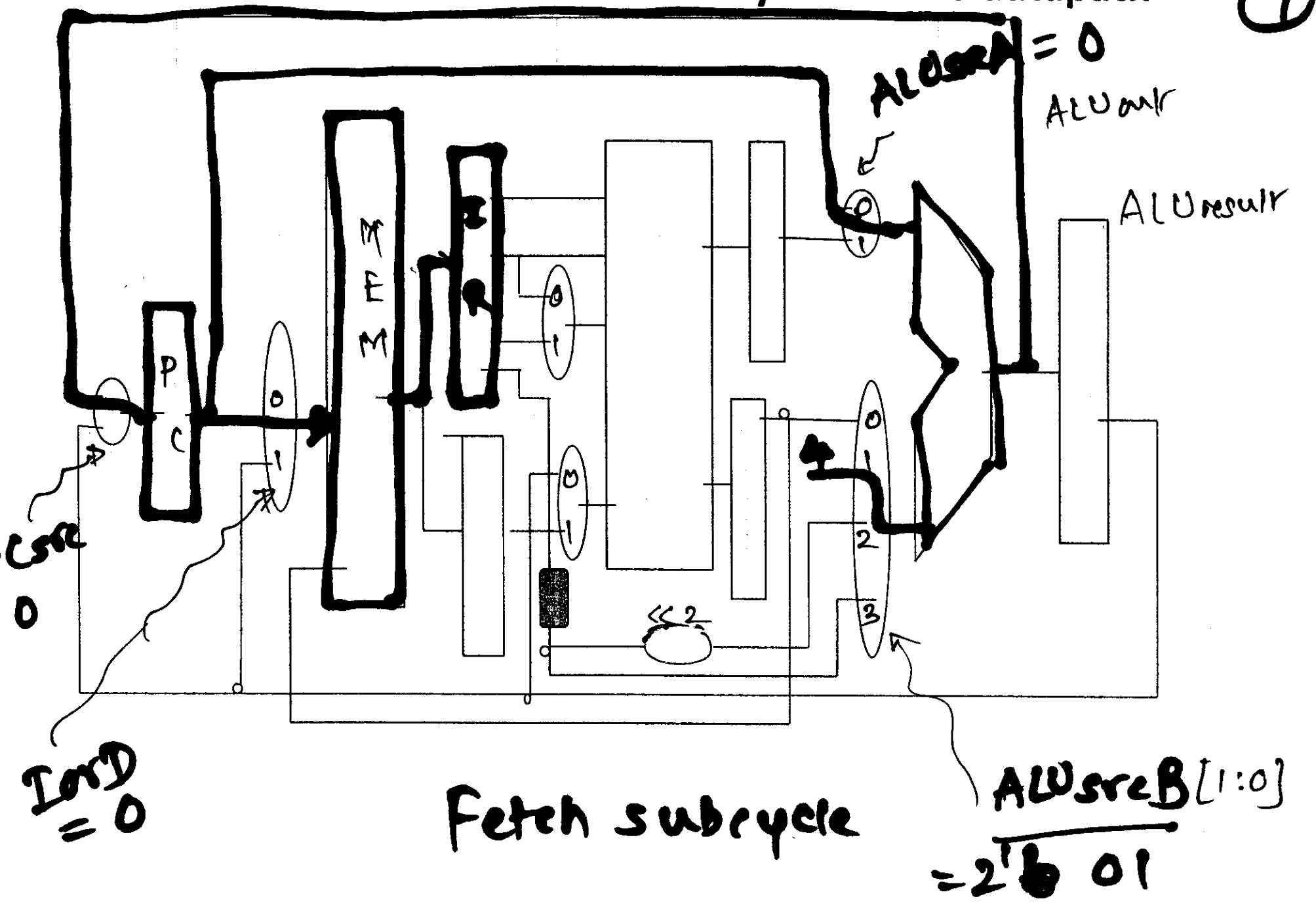
6

A Multi-cycle mMIPS datapath



A Multi-cycle mMIPS datapath

⑦



$ALUSrcA = 0$

Fetch subcycle

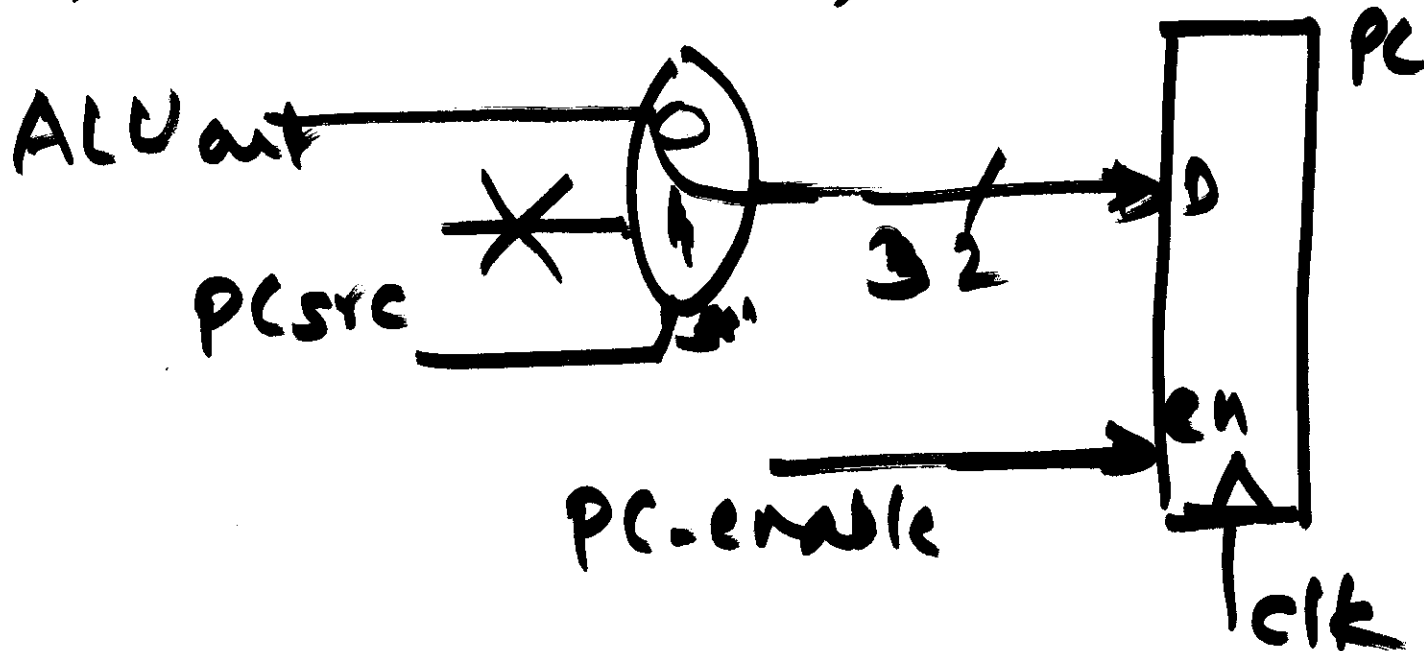
$ALUSrcB[1:0] = 2'b\ 01$

8

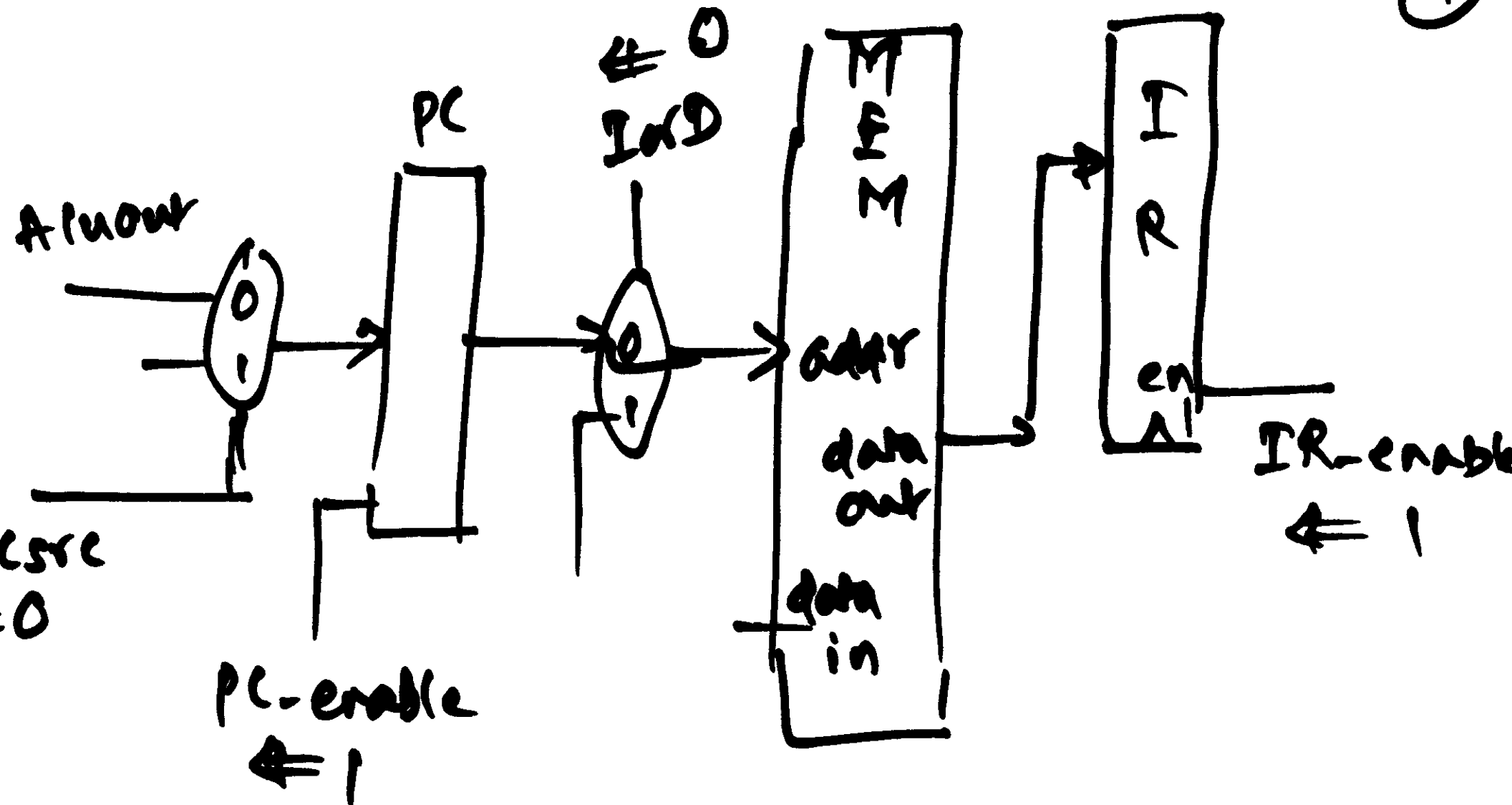
Fetch subcycle

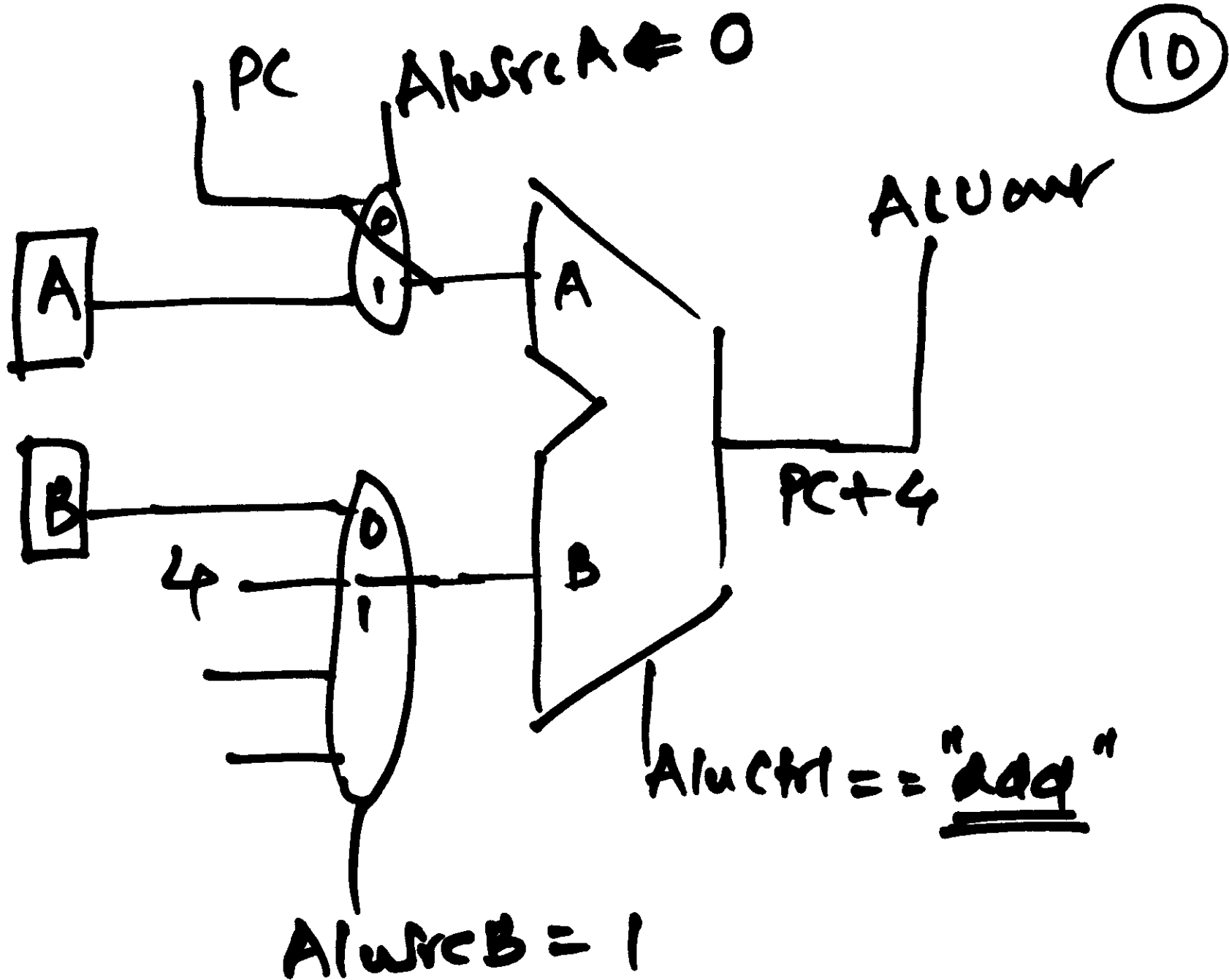
Control Signals

```
PCsrc <= 0 ; // ALU out  
           // which is PC + 4  
PC.enable <= 1 ;
```

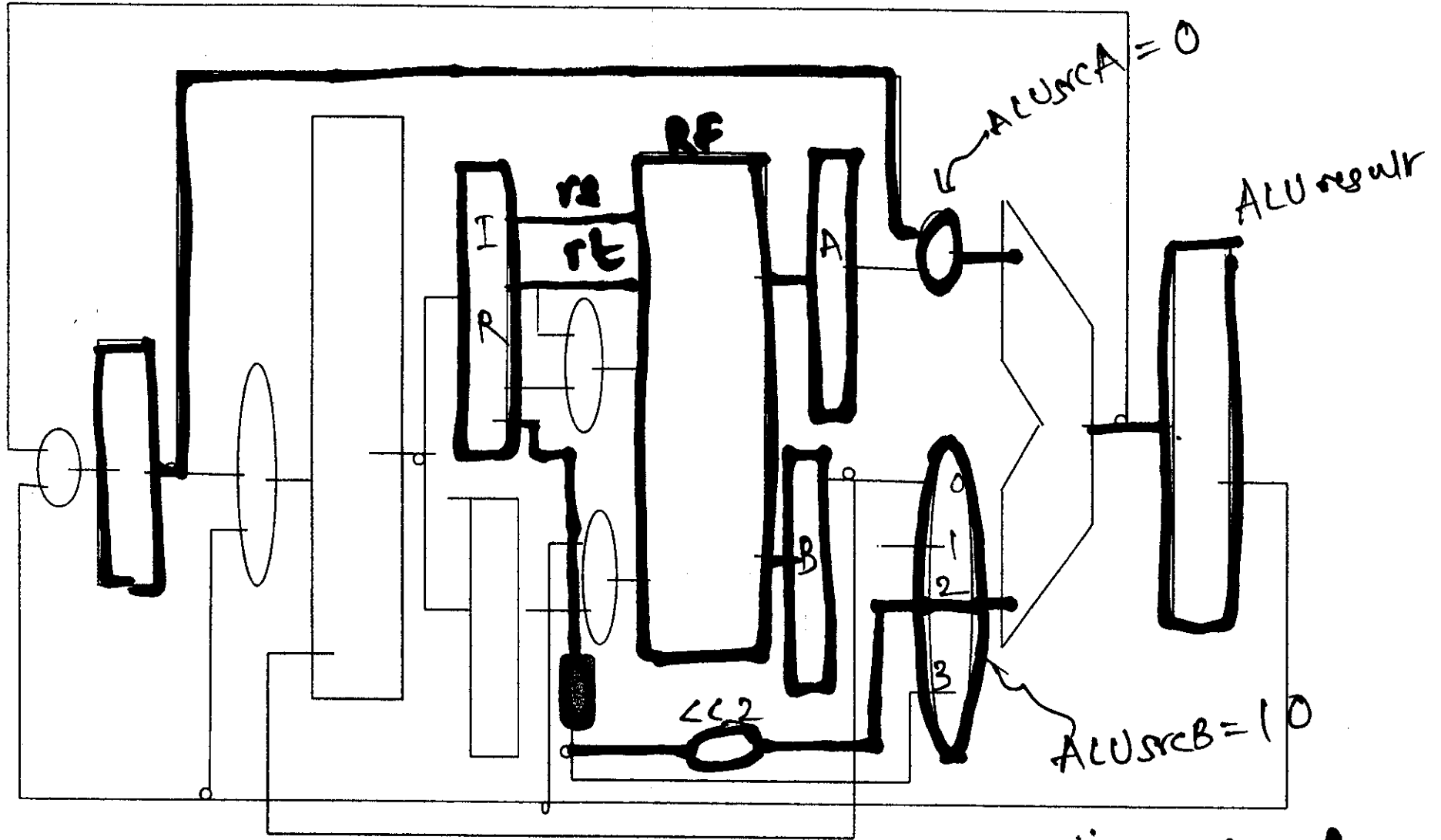


④





A Multi-cycle mMIPS datapath



Decode ~~stage~~

subcycle

- inputs to FSM are now avail in IR

Control signals for
Decode sub cycle

12

ALUsrcA \leftarrow 0

"imm"

sign extend

$\ll 2$

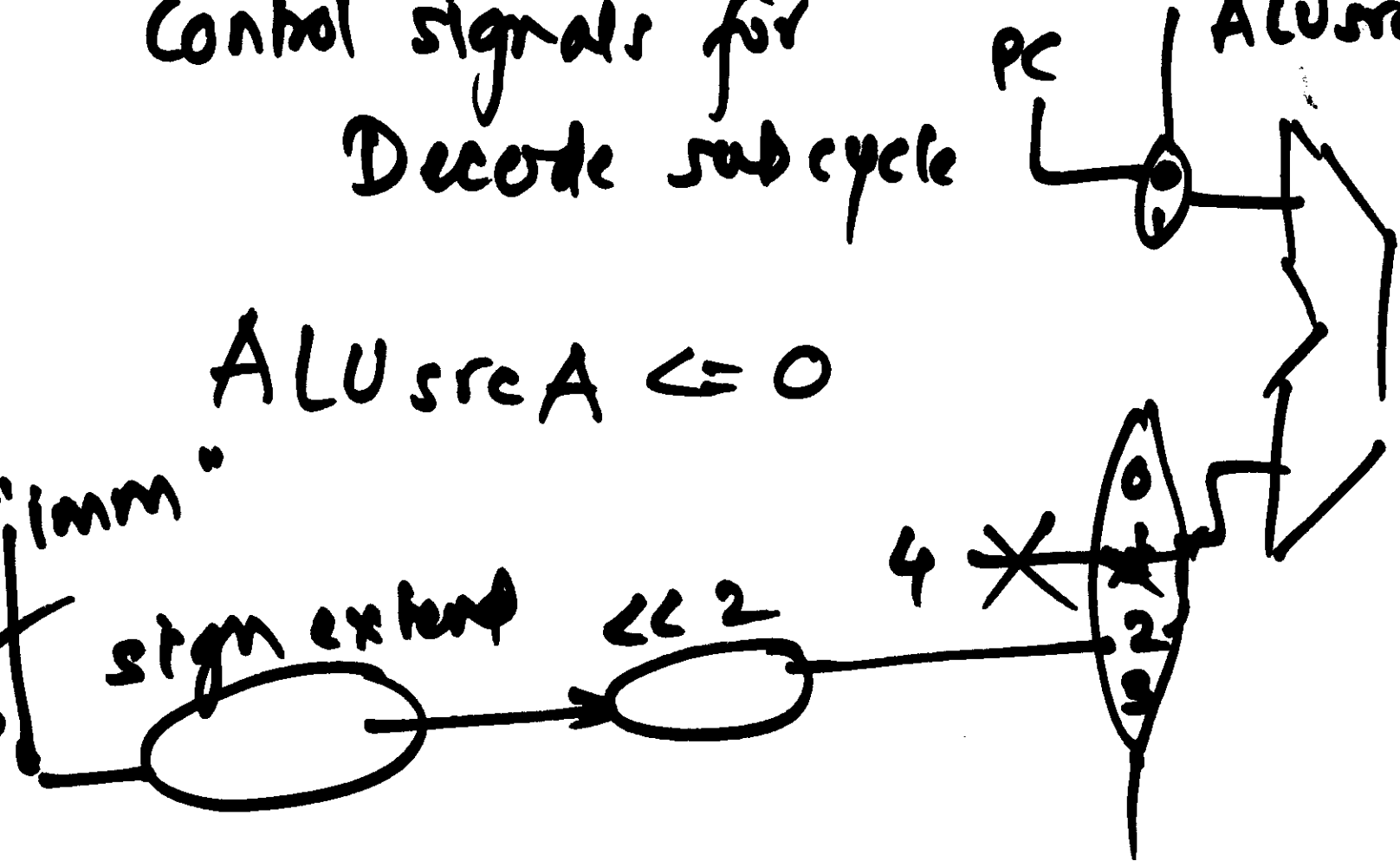
4

ALUsrcB \leftarrow ~~imm~~
2'b10

ALUsrcB \leftarrow 2

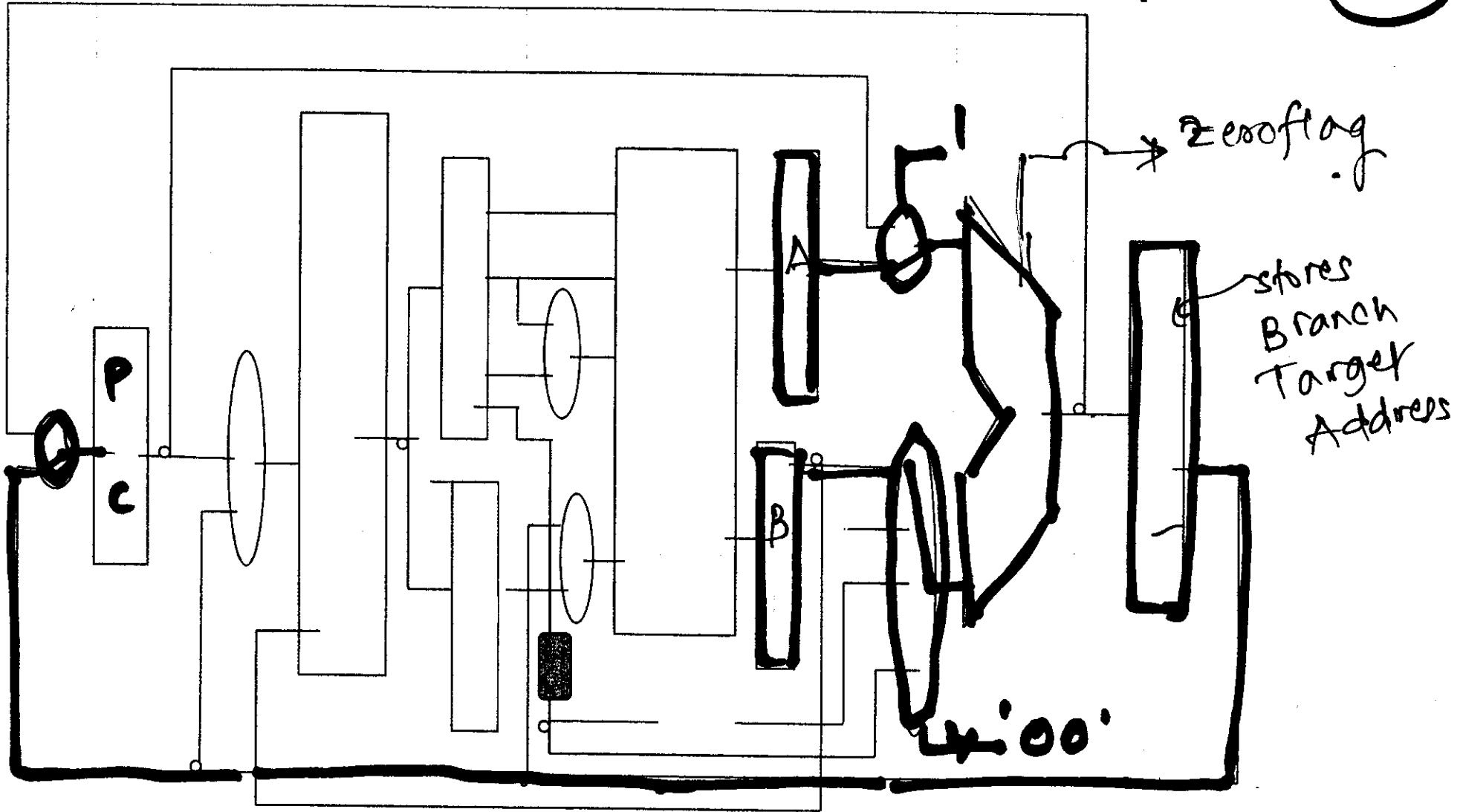
ALUenable \leftarrow HIGH;

PC ALUsrcA \neq 0



A Multi-cycle mMIPS datapath

13



ExBeq subcycle

ExBeq subcycle

(14)

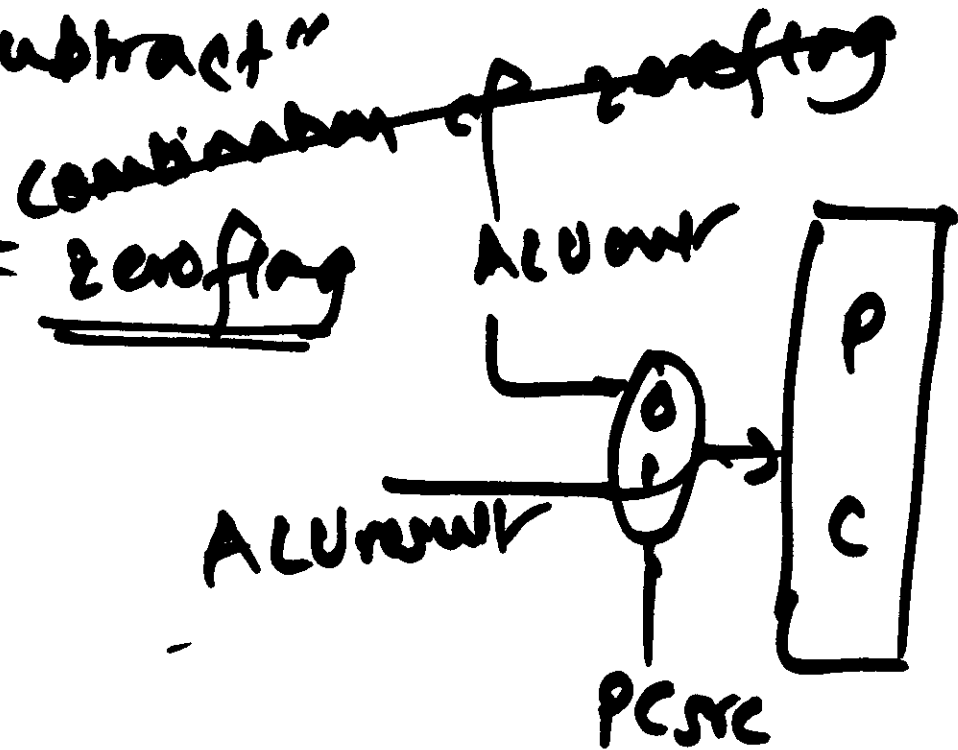
ALUsrcA \leftarrow 1

ALUsrcB \leftarrow 00

ALUctrl \leftarrow "subtract"

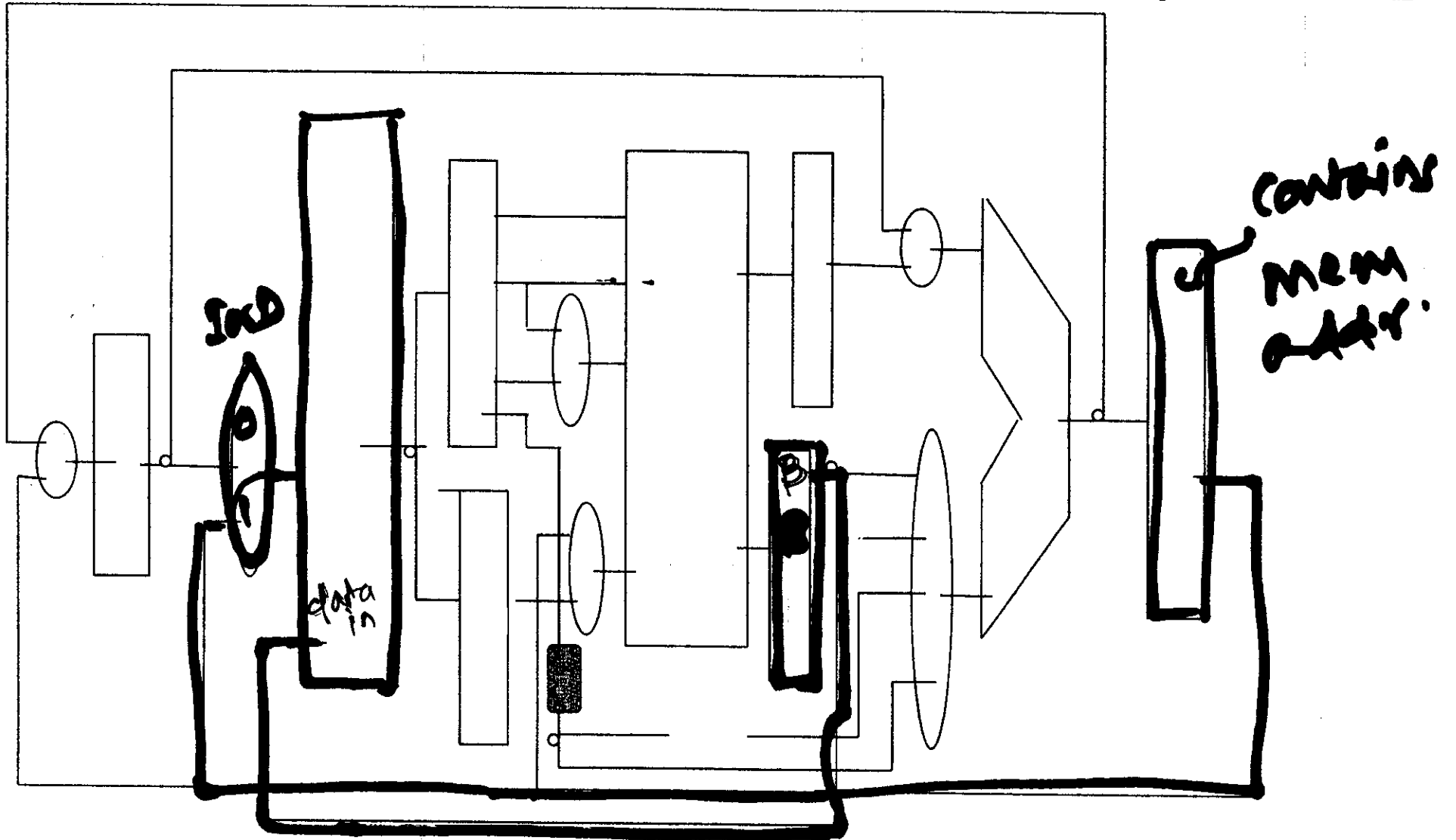
PC-enable \leftarrow zeroing

PCsrc \leftarrow 1



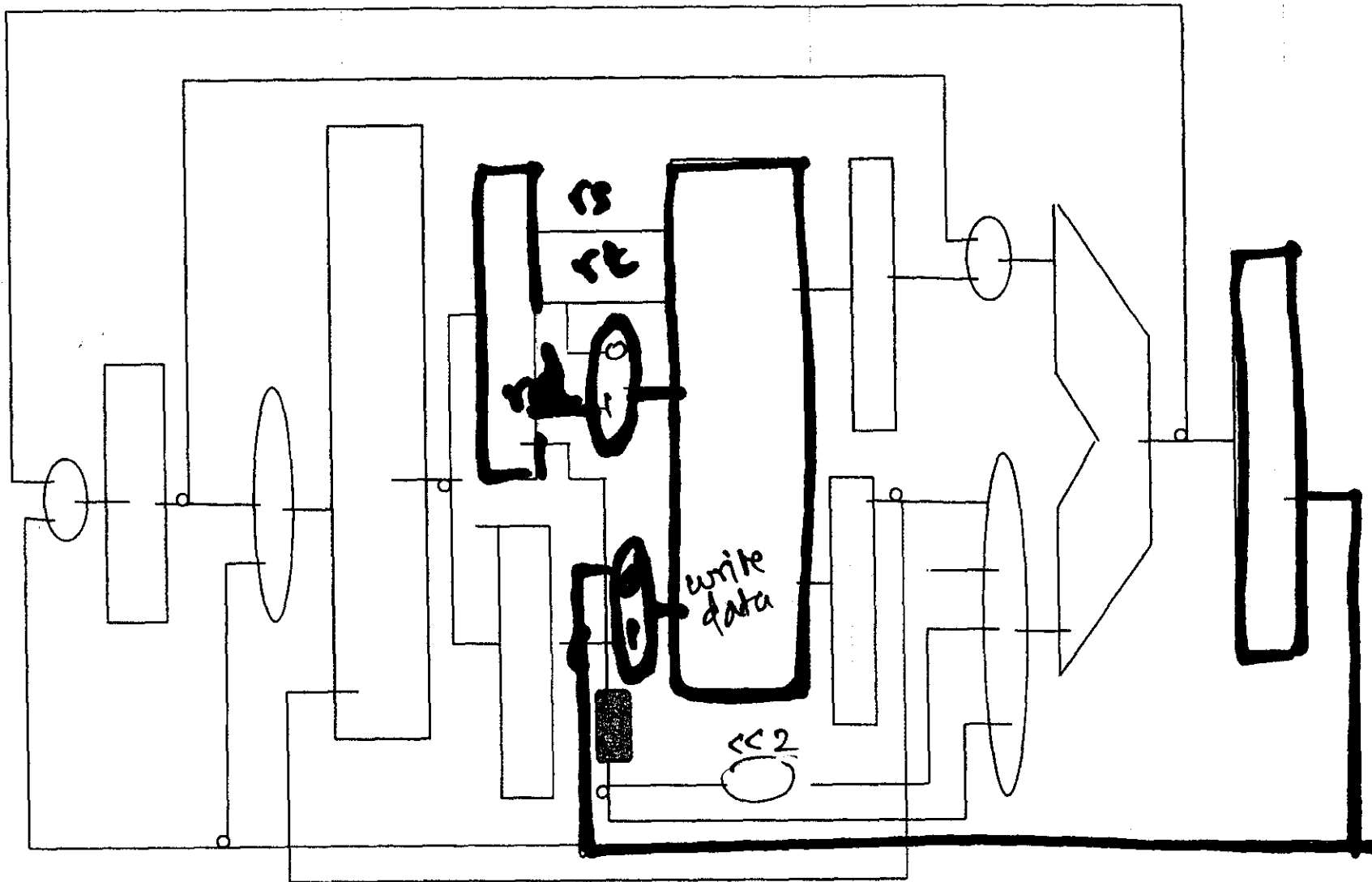
A Multi-cycle mMIPS datapath

15



MA store subcycle

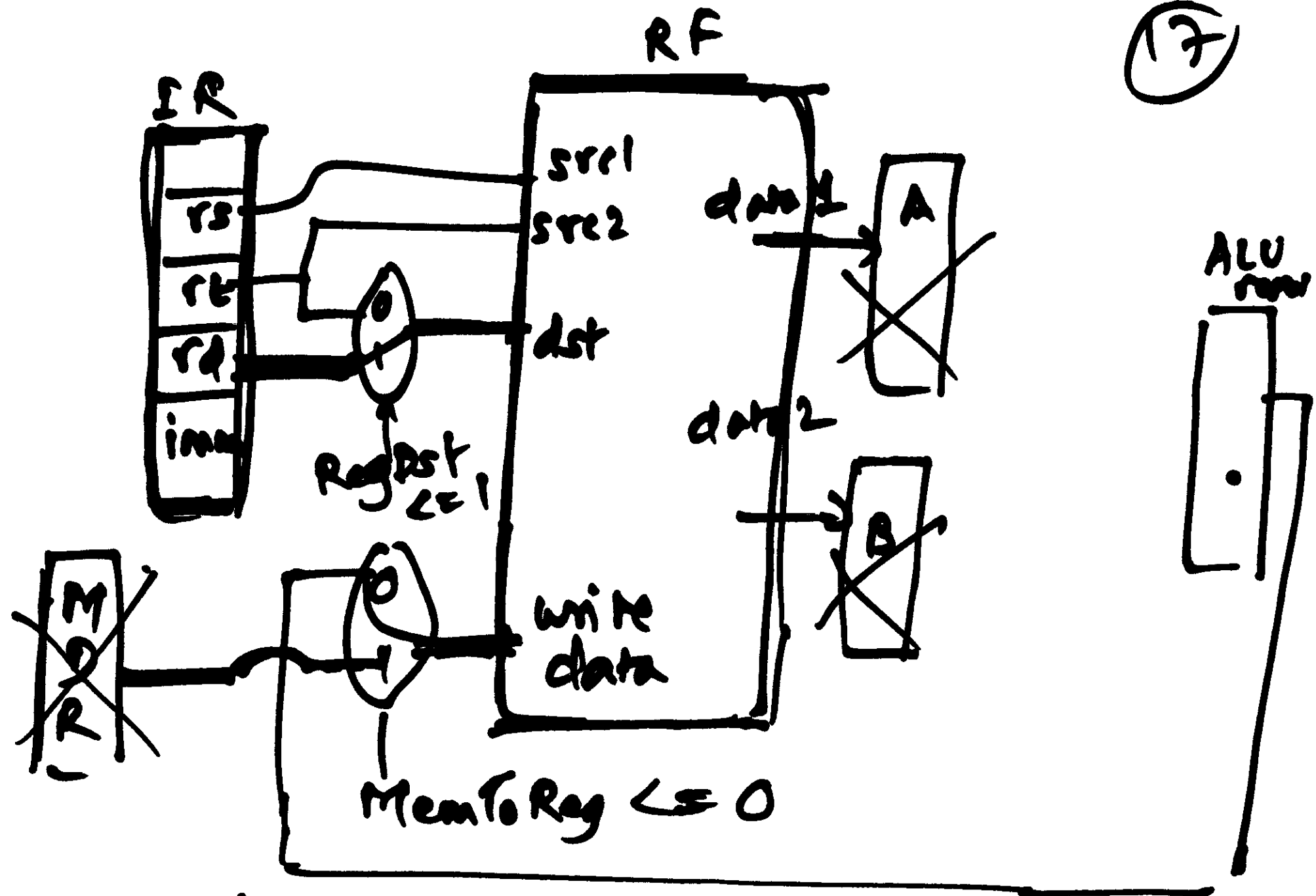
A Multi-cycle mMIPS datapath



WB add subcycle

MEMTOREG
MEMTOREG

17



WB add subcycle