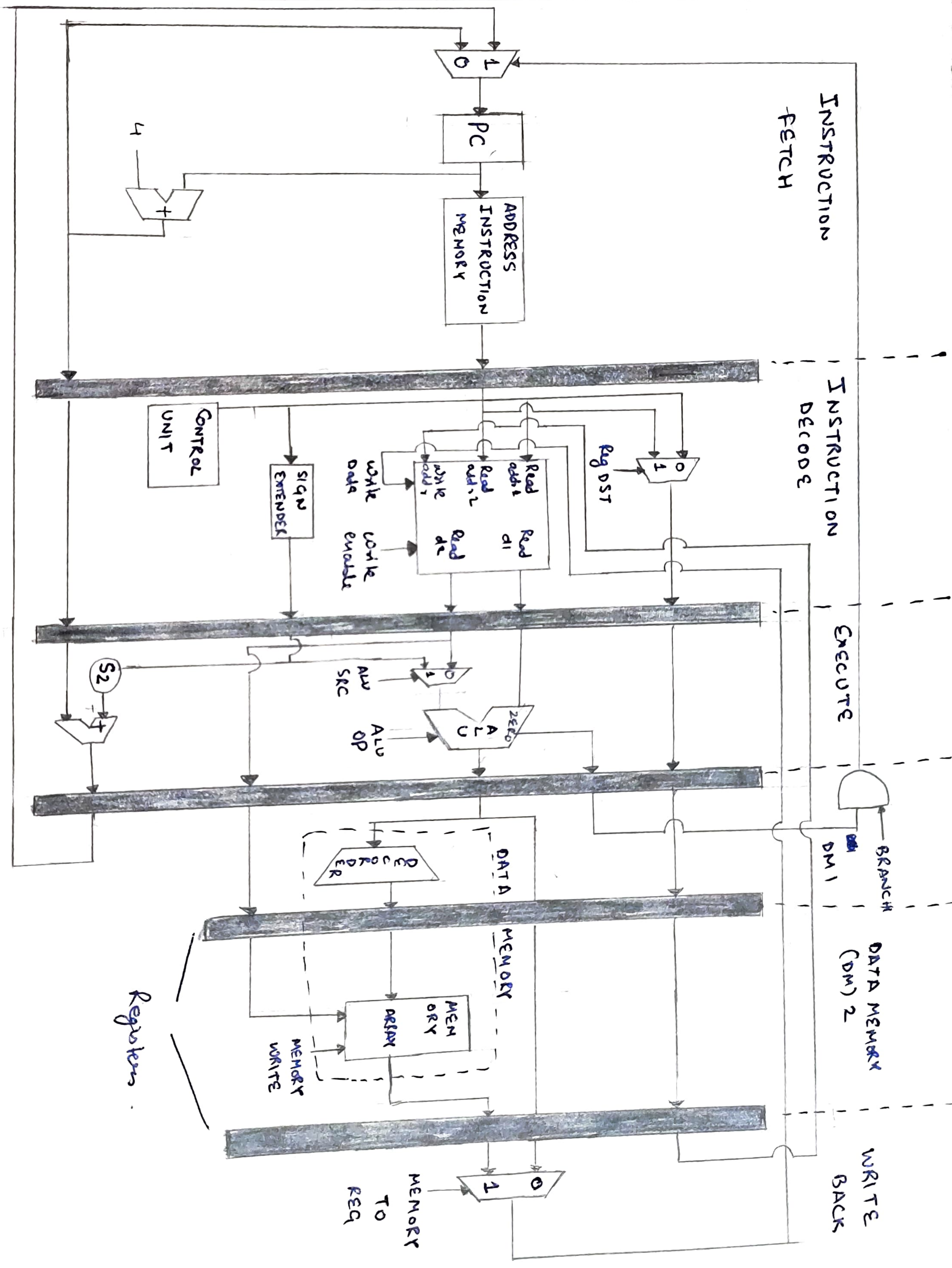
CS-321
ESE PART - II (2021)

TARUSI MITTAL
1901CS65

6-STAGE PIPELINE ARCHITECTURE



Que 1 (a) The above drawn figure represents a six-stage pipelined datapath, in which the data memory operational stage is divided into 2 stages (DM1, DM2).

The 6 total stages are : (IF, ID, EX, DM1, DM2, WB) as shown above.

The basic idea of pipelining is that in this multiple instructions are overlapped during execution.

(b) Yes, on dividing DM into 2 stages the performance may increase. provided that the memory stage is the major time defining step

For our circuit we will take the CPI = 1 (for large no of instruction). and ~~we~~ as for the full pipeline, in one cycle one instruction is executed.

Now, if we have a set of instructions. then total time would be, no of cycles x time period. = $(x+k-1)\times T \approx xT$; x = no of cycles.

As we know when we calculate time, we take the largest time. So, if our DM stage is the most time consuming step then our time for the circuit will be determined by it.

Now, when we divide it into 2 parts!

DM 1 → It will only decode and provide address to memory arrays where the data is stored.

DM2 → It will read and output the data.

If orginally our step was taking time T and DM_1 was taking t_1 then now DM_2 will take $T-t_1$

And total time now will be max $(t_1, T-t_1)$.

Lets take $t_1 = 50$ and T to be 100

So orginally Time $= 100x$ (x = no of cycles)

But now its $= \max(50, 50)x$

$= 50x$

Now, as $50x < 100x$

Therefore our performance time increases when Data Memory is divided.

ue 2:- **Instruction Set**

1. lw $\boxed{\$3}$, 0($7) ⟶ Step 1
2. add $7 , ($3) , $5 ⟶ Step 2
3. sub $8 , $5, ($3) ⟶ Step 3.

Now it is clear that for step 2 and step 3 we need an updated value of $3 . and we have to make sure that whenever step 2 and step 3 are performed step 1 is completed and value of stored in $3 is the updated value.

Now if we try to do it normally.

IF	ID	EX	DM1	DM2	WB			Step 1
	IF	ID	EX	DM1	DM2	WB		Step 2
		IF	ID	EX	DM1	DM2	WB	Step 3

Here the steps 2 and step 3 are using the value before it is updated. as the updated value of $3 has to be written back first or forwarded.

(a) No forwarding

IF	ID	EX	DM1	DM2	WB						Step 1
	IF	ID	ID	ID	ID	EX	DM1	DM2	WB		Step 2
		IF	IF	IF	IF	ID	EX	DM1	DM2	WB	Step 3
1	2	3	4	5	6	7	8	9	10	11	⟶ No of cycles.

With No Forwarding No. of Cycles = 11

As we can see that the step 2 waits until the value of $3 in step 1 gets written back and hence step 3 also have to wait until step 2 will not move to further ~~[illegible]~~ stage

(b) With forwarding:

IF	ID	EX	DM_1	DM_2	WB					step 1
	IF	ID	ID	ID	EX	DM_1	DM_2	WB		step 2
		IF	IF	IF	ID	EX	DM_1	DM_2	WB	Step 3
1	2	3	4	5	6	7	8	9	10 $\rightarrow$	No of cycles

With forwarding, No. of cycles = 10.

If we compare we can see the forwarding takes 1 less cycle than from no forwarding.

In this the step 2 will be continued. by accessing data from. DM 2 state of step 1 via forwarding.

Therefore we need not to wait for the value of #3 to be written back.

—x —x —x —x —x —x —x —x —x —x