



# **University of Asia Pacific (UAP)**

## **Department of Computer Science & Engineering**

**Program: B.Sc. in Computer Science and Engineering**  
**Final Examination, Fall 2020**

**Course Code : CSE 317**

**Course Title : Computer Architecture**

**Name: Sudip Ghose**

**Reg. ID: 18101094**

**Semester: 3rd Year 2nd Semester**

**Section: B**

**Date: 05-05-21**

Ams to the qus no. 4 (or)

(a)

Four question about cache design.

1. where can a block be placed in upper level? (Block placement  
→ mapping function)
2. How is a block found if it is in the upper level? (Block identification)
3. which block should be replaced on a miss? (Block replacement)
4. what happens on a write?

(write strategy)

Ans to the qus no. 4(or) b

Here, my last two digit of id = 94

$$\therefore i = 94$$

$$\therefore A[i] = B + A[i+11] - C$$

$$\Rightarrow A[94] = B + A[105] - C$$

Now, Instructions are:

$$A \rightarrow \$S_0(16)$$

$$B \rightarrow \$S_1(17)$$

$$C \rightarrow \$S_2(18)$$

$$\text{temp} \rightarrow \$t_0(8)$$

$S_0$ ,

i) lw \$t\_0, 420(\$\$S\_0)

ii) add \$t\_0, \$t\_0, \$\$S\_1

iii) sub \$t\_0, \$t\_0, \$\$S\_2

iv) sw \$t\_0, 376(\$\$S\_0)

machine code:

(1)

100011	10000	01000	0000000001011100
OP	rs	rt	offset

(II)

000000	01000	10001	01000	00000	100000
OP	rs	rt	rd	shamt	funct

(III)

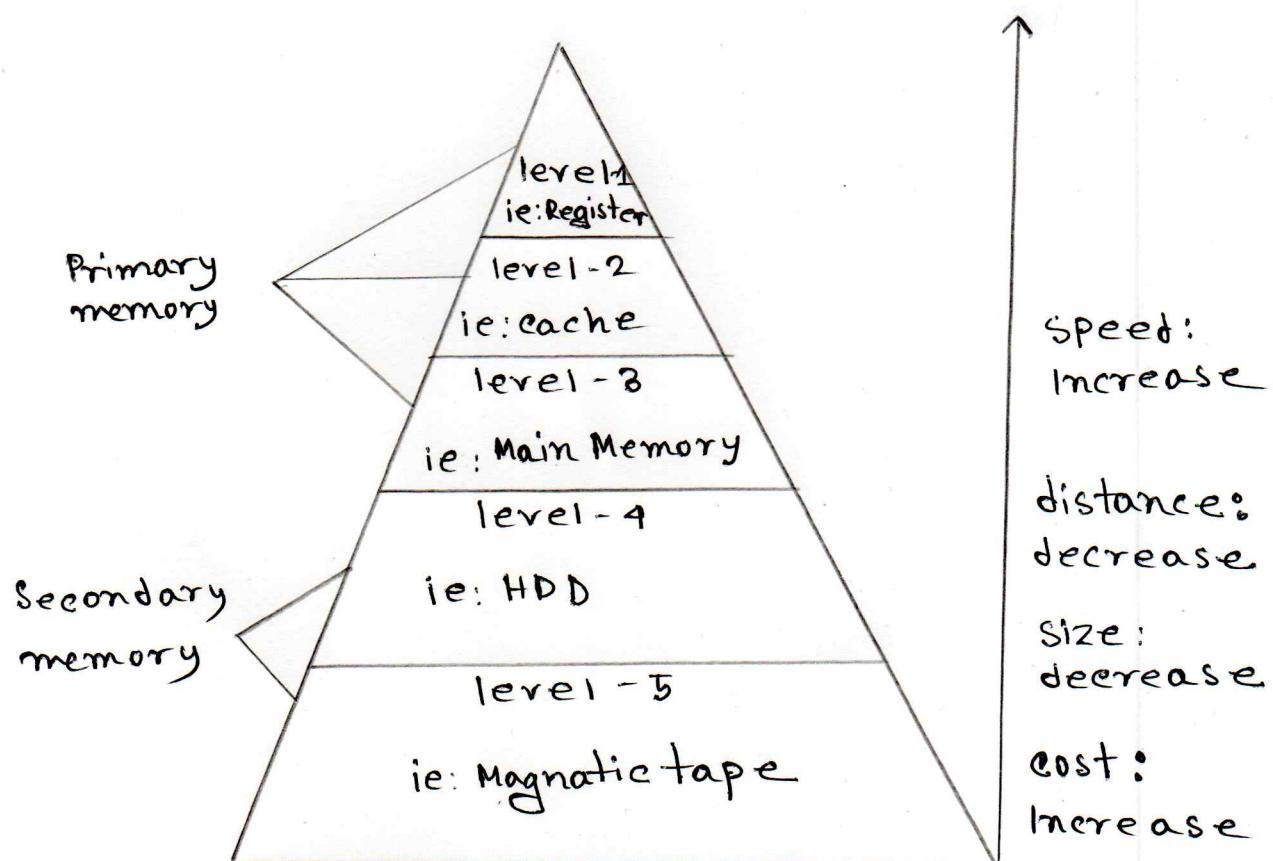
000001	01000	10010	01000	00000	100010
OP	rs	rt	rd	shamt	funct

(IV)  
~~1010~~

101011	10000	01000	0000 0000 00110000
OP	rs	rt	offset

### Ans to the qus no. 4(or) e

memory hierarchy is an enhancement to organize the memory such as it can minimize the access time. the memory hierarchy was developed based on a program behavior known as locality of reference.



Here,  $x = 04$

## Block Address :

(94+15), (94+25), (94+12)

= 109, 119, 106

## 1. Directed mapped:

$$\text{No of hit} = 0$$
$$\text{No of miss} = 3$$

Request Address	hit miss	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	
109	miss	MEM[109]	
119	miss	MEM[119]	
106	miss	MEM[106]	

## ⑪ 2 way associative

$$\frac{\text{no of block}}{2} = \frac{16}{2} = 8 \text{ set} \quad : \text{No of hit} = 0 \\ \text{No of miss} = 3$$

(iii) 4 way associative,

$$\frac{16}{4} = 4 \text{ sets}$$

No of hit - 0  
No of miss - 3

Requested address	Set 0				Set 1				Set 2				Set 3			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
100	hit	miss														
110																
101																
111																
100																
101																
110																
111																

[61] main

[60] main

② 8 way associative

$$\frac{16}{8} = 2 \text{ step}$$

No of hits = 0  
No of misses = 0

(v) 16 way / Full associative:

Ans to the Ques no. 2(b)

① Here,

$$\text{cache size} = 32 \text{ word}$$

$$\text{memory size} = 512 \text{ word}$$

$$\text{Block size} = 4 \text{ word}$$

$$\text{memory block} = \frac{512}{4} = 128 \text{ block}$$

$$\text{Cache line} = \frac{32}{4} = 8 \text{ block}$$

$$\text{total bit for physical address} = \log_2(512)$$

$$= 9 \text{ bit}$$

$$\text{Index} = \log_2(\text{Cache line}) = \log_2(8) = 3 \text{ bit}$$

$$\text{Offset} = \log_2(\text{block size}) = \log_2(4) = 2 \text{ bit}$$

$$\text{tag bit} = 9 - 3 - 2 = 4 \text{ bit.}$$

tag	index	offset
4 bit	3 bit	2 bit

(11)

cache size = 16 word

memory size = 64 word

block size = 4 word

memory block =  $64/4 = 16$  blockcache line =  $16/4 = 4$  blocktotal bit =  $\log_2(64) = 6$  bitIndex =  $\log_2(4) = 2$  bitoffset =  $\log_2(4) = 2$  bittag bit =  $6 - 2 - 2 = 2$  bit.

tag	index	offset
2 bit	2 bit	2 bit

(iii)

cache size = 16 word

main memory size = 128 word

block size = 4 word

memory block =  $128 / 4 = 32$  blockcache line =  $16 / 4 = 4$  blocks.

~~total~~  
~~total~~ bit =  $7 - 2 - 2 = 3$  bit.  
 tag

tag	index	offset
3 bit	2 bit	2 bit

$$\text{Index} = \log_2 (4) = 2 \text{ bit}$$

$$\text{offset} = \log_2 (4) = 2 \text{ bit}$$

$$\text{tag bit} = 7 - 2 - 2 = 3 \text{ bit}$$

(v)

$$\text{cache size} = 32 \text{ word}$$

$$\text{memory size} = 1024 \text{ word}$$

$$\text{block size} = 9 \text{ word}$$

$$\text{memory block} = 1024 / 9 = 256 \text{ block}$$

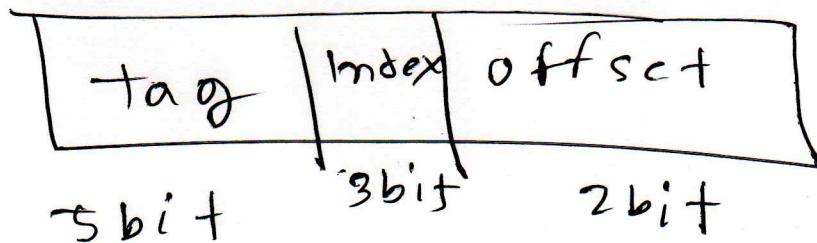
$$\text{cache line} = 32 / 9 = 8 \text{ block}$$

$$\text{total bits} = \log_2(1024) = 10 \text{ bit}$$

$$\text{index} = \log_2(8) = 3 \text{ bit}$$

$$\text{offset} = \log_2(9) = 2 \text{ bit}$$

$$\text{tag} = 10 - 3 - 2 = 5 \text{ bit}$$

Ans

⑦

cache size = 8 word

memory size = 16 word

block size = 4 word

memory block =  $16/4 = 4$  blockcache line =  $8/4 = 2$  blocktotal bit =  $\log_2(16) = 4$  bitindex =  $\log_2(2) = 1$  bitoffset =  $\log_2(4) = 2$  bittag =  $4 - 1 - 2 = 1$  bit

tag	index	offset
1 bit	1 bit	2 bit

Ans to the qu no. 3(a)

$$\text{Instruction} = x + \bar{x} = 1 + 1 = 1, \quad \text{stage} = 5$$

S <sub>1</sub>	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>	T <sub>7</sub>	T <sub>8</sub>	T <sub>9</sub>	T <sub>10</sub>	T <sub>11</sub>
S <sub>2</sub>	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>	T <sub>7</sub>	T <sub>8</sub>	T <sub>9</sub>	T <sub>10</sub>	111
S <sub>3</sub>	11	12	13	14	15	16	17	18	19	10	111
S <sub>4</sub>	11	12	13	14	15	16	17	18	19	10	111
S <sub>5</sub>	11	12	13	14	15	16	17	18	19	10	111

Here  $\kappa = 5$ ,  $m = 11$ ,

total time for pipeline =  $k+m+1 = 5+11+1 = 17$  clock cycle

total time for non-pipeline =  $m \times \kappa = 5 \times 11 = 55$  clock cycles.

$$\text{Speedup} =$$

$$\frac{\text{clock cycle of non-pipeline}}{\text{clock cycle of pipeline}} = \frac{15}{55} = 0.2727$$

$$\text{Utilization} = \frac{\text{used block}}{\text{total block}} = \frac{5 \times 11}{55 + 20} = \frac{55}{75} = 0.7333$$

$$= 73.33\%$$

Ans to the ques no. 3 (b)

Non pipeline

$$\text{Instruction latency} = \sum (\text{stage time})$$

$$= (30 + 25 + 29 + 12) + 20 \\ = 96 \text{ ns}$$

$$\text{instructions} = (99 + 1000) = 1099$$

$$\therefore \text{execution time} = \cancel{96} \times (1099 \times 111) \text{ ns} \\ = \cancel{90559} \text{ ns} \\ = 121434 \text{ ns}$$

For pipeline:

$$\text{Instruction latency} = \max \text{ time of stage} \\ \times \text{no of stages} \\ = \cancel{20} \times (30 \times 5) = 150 \text{ ns}$$

$$\text{execution time} = 150 + (30 \times 1099) \\ = 32970 \text{ ns}$$

$$\text{so, speed up} = \frac{\text{Non-pipeline time}}{\text{Pipeline time}} \\ = \frac{121439}{32970} \\ = 3.68$$

Ans

Ans to the Ques no. 3(c)

For pipeline:

$S_1$	$I_1$	$I_2$	$I_3$					
$S_2$		$I_1$	$I_2$	$I_3$				
$S_3$			$I_1$	$I_2$	$I_3$			
$S_q$				$I_1$	$I_2$	$I_3$		
$S_5$					$I_1$	$I_2$	$I_3$	

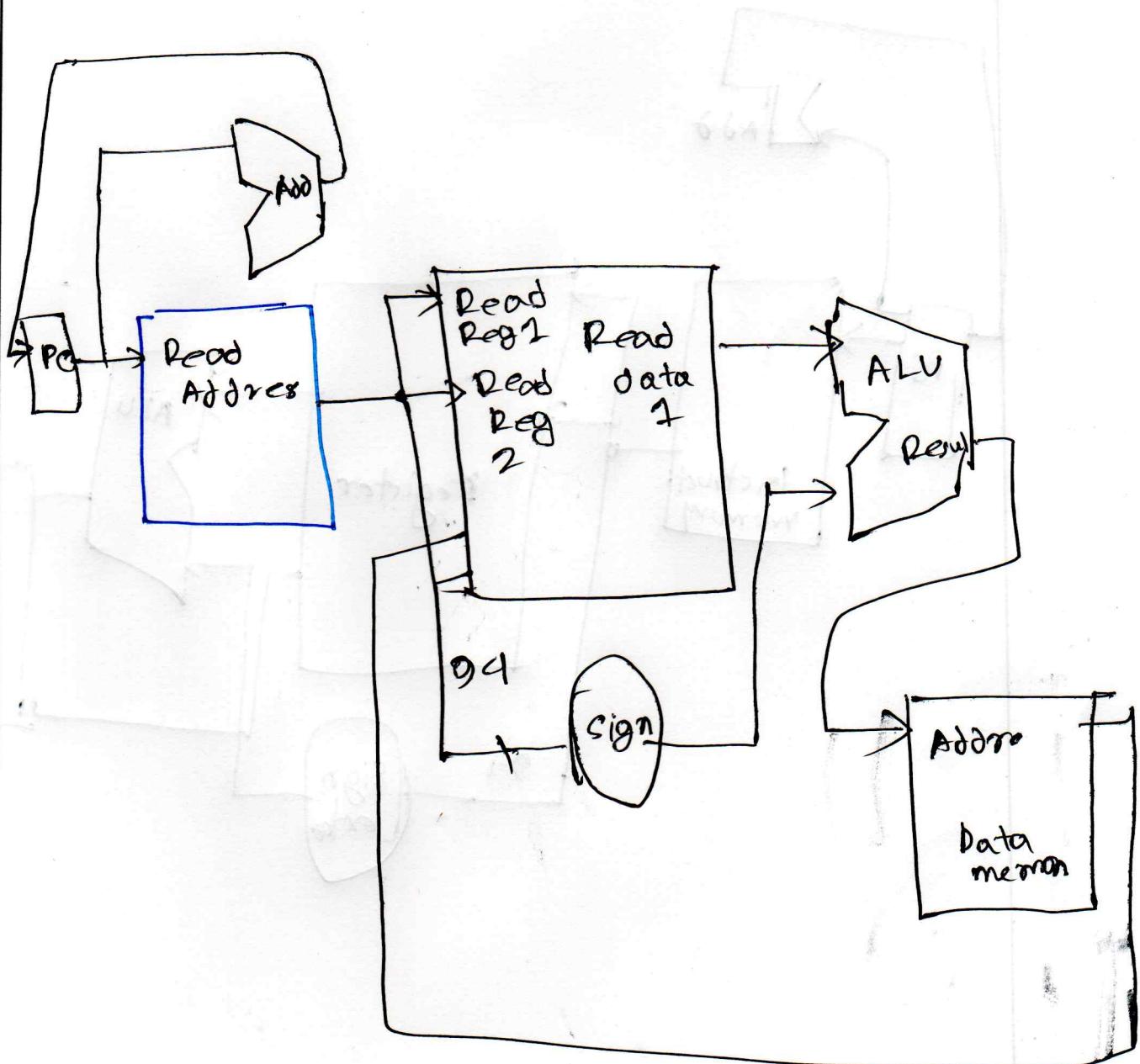
There would be hazard for  $S_2$  of instruction 2 is loaded to pipeline before instruction 1 store new values to  $S_2$ . Same goes for instruction 3 it also cause hazard.

For non-pipeline.

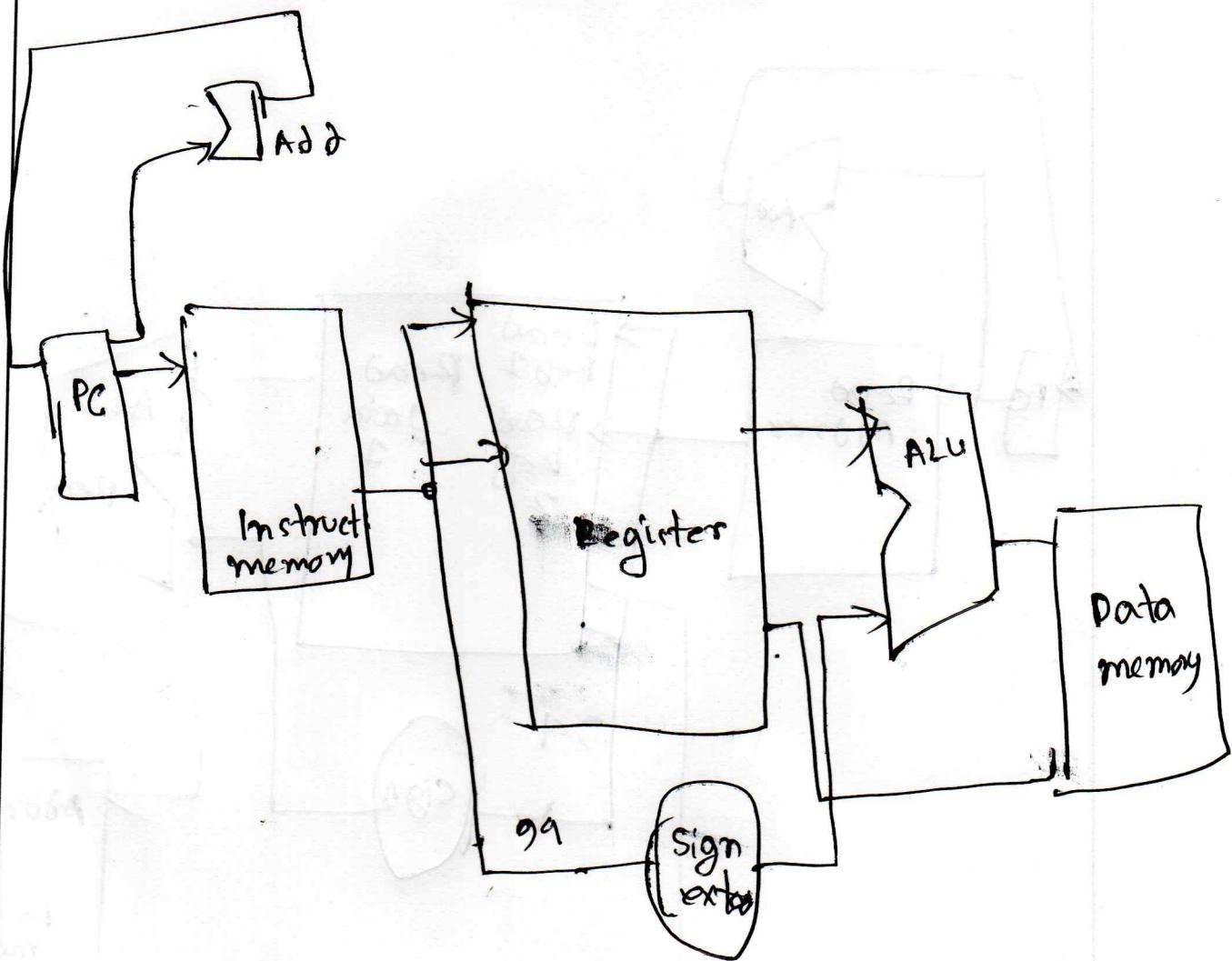
$S_1$	$I_1$	$I_2$	$I_3$					
$S_2$		$I_1$	$I_2$	$I_3$				
$S_3$				$I_1$	$I_2$	$I_3$		
$S_q$					$I_1$	$I_2$	$I_3$	
$S_5$						$I_1$	$I_2$	$I_3$

Ans to the qus no.1

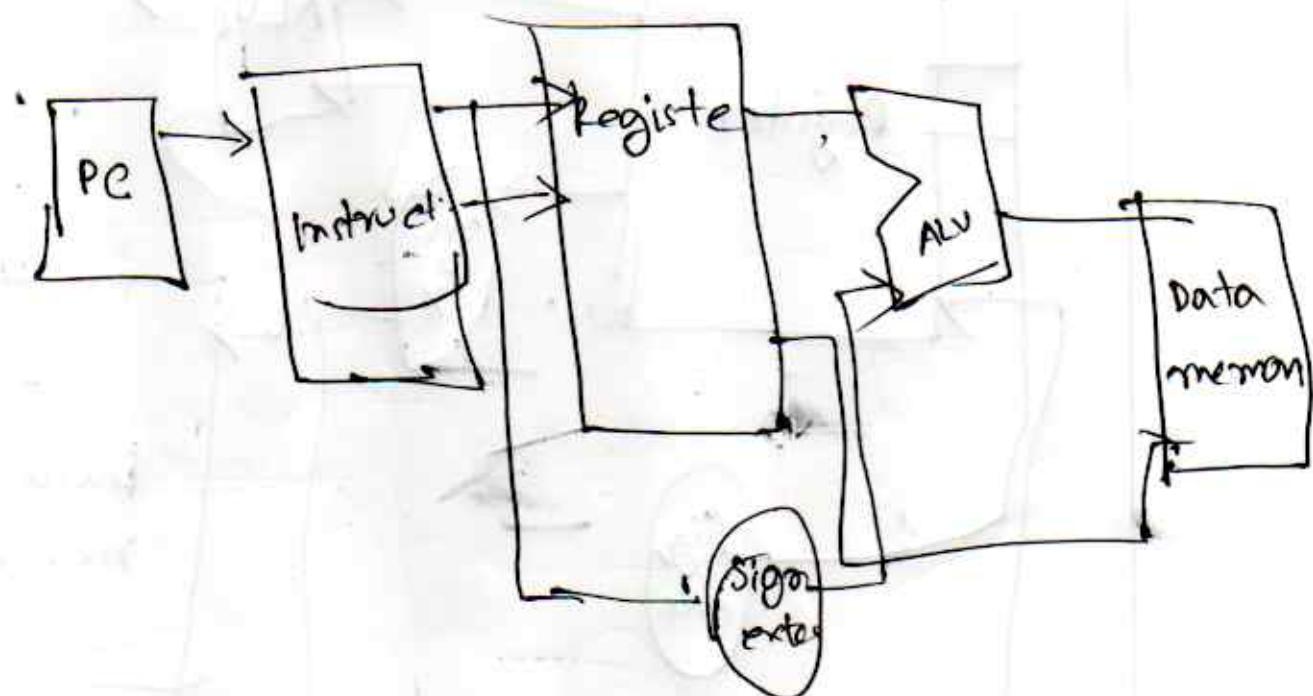
①  $lw \$t0, \$04(\$s0)$



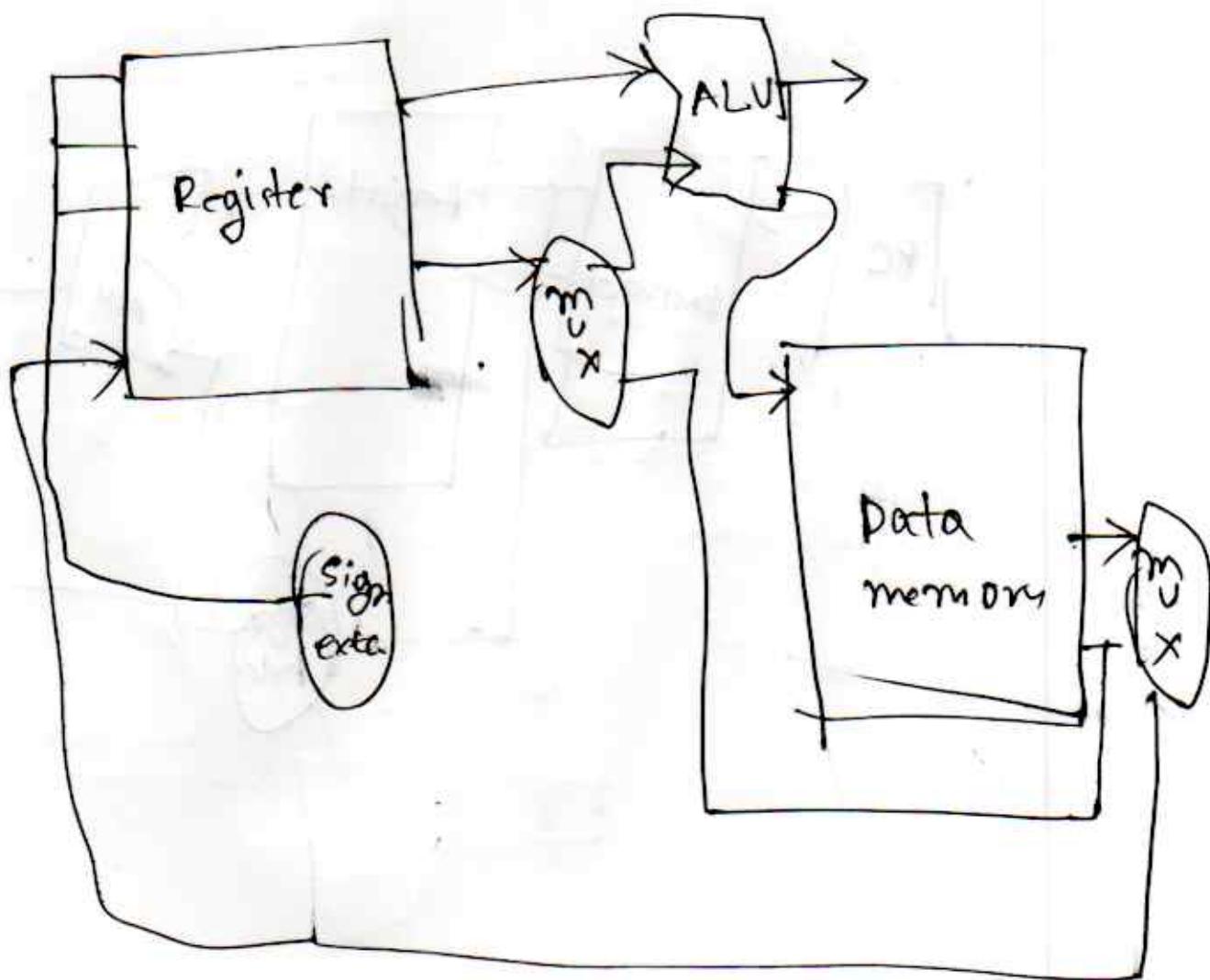
(i)  $sw \$t6, 09 (\$s0)$



(iii) Add \$ to \$s<sub>1</sub>, s<sub>2</sub>



(iv)  $A[99] = Y + A[102]$



iv) For the 4th instruction:

$$A[x] = y + A[x+8]$$

Here,

$x$  = Last two digit of ID

$$\therefore x = 94$$

$$\therefore A[94] = y + A[94+8] = y + A[102]$$

Now,

$$A \rightarrow \$s_0 (16)$$

$$Y \rightarrow \$s_1 (1x)$$

$$\text{temp} \rightarrow \$t_0 (8)$$

so, the instruction are,

a) lw \$t\_0, 408(\$s\_0)

b) add \$t\_0, \$t\_0, \$s\_1

c) sw \$t\_0, 3x6(\$s\_0)

a)  $lw \#0, 408 (\$S_0)$

This is I-type instruction, so ALU src is SET. Number of that is  $\$S_0$  is passed to read register 1. And we'll get its content that is the base address from read data 1. 16 bit offset is converted to 32 bit using sign extend. base address and offset will go to ALU and ALU will add them thus we'll get the physical address from ALU result. This physical address goes to data memory fetches data from that address and pass that data to write register. Regwrite is SET.

b) add \$t<sub>0</sub>, \$t<sub>0</sub>, \$s<sub>1</sub>

This instruction is R-type instruction, so ALUSrc is CLEAR. The number of \$t<sub>0</sub>, \$s<sub>1</sub> is 8818 is passed to read register1 & read register2. Then content of the two from read data1 & read data2. These two data is passed to ALU. ALU adds them thus we get the result from ALU. The result is passed to mem-to-Reg Mux but as it is CLEAR so the mux passes the result to write of \$t<sub>0</sub> that is passed to write register and Regwrite is SET.

c) sw \$t<sub>0</sub>, 37 6 (\$s<sub>0</sub>)

As this is an I-type instruction so ALU Src is SET. The number of \$s<sub>0</sub> that is 16 is passed to read register 1 and the number of \$t<sub>0</sub> that is 8 is passed to read register 2. So, the content of that is the base address from read data 1 and content of \$s<sub>0</sub> that is the data to be stored from data 2. 16 bit offset is converted to 32 bit. Base address and offset is passed to ALU thus we get physical from ALU result. This physical address and data to be stored is passed to data memory. This is the procedure of this instruction.