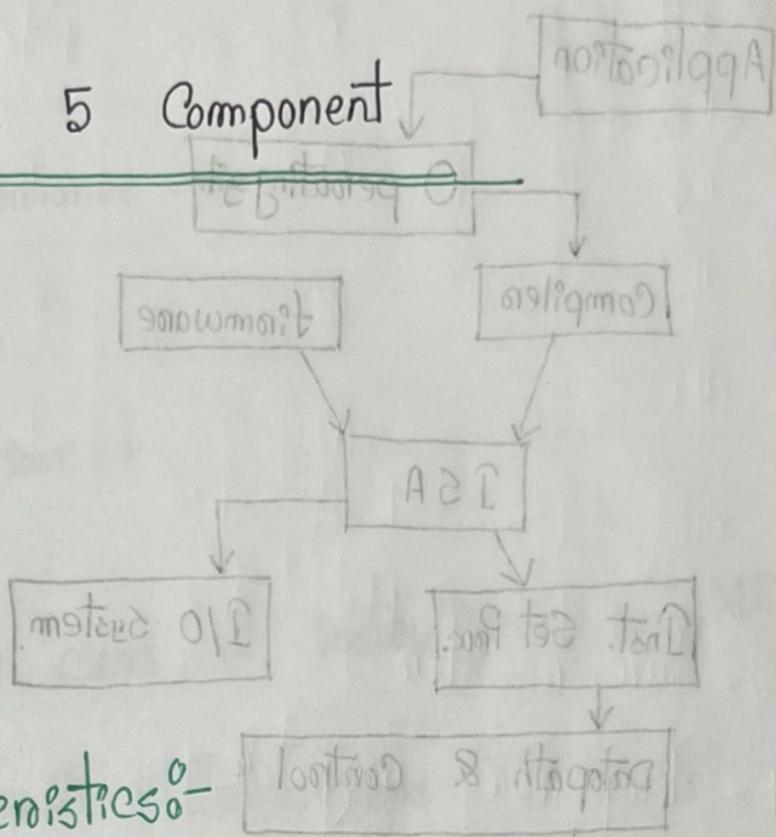


~~connection to input~~

Lecture-1

Computers have 5 Component

- ① Control
- ② Datapath
- ③ Memory
- ④ Input
- ⑤ Output

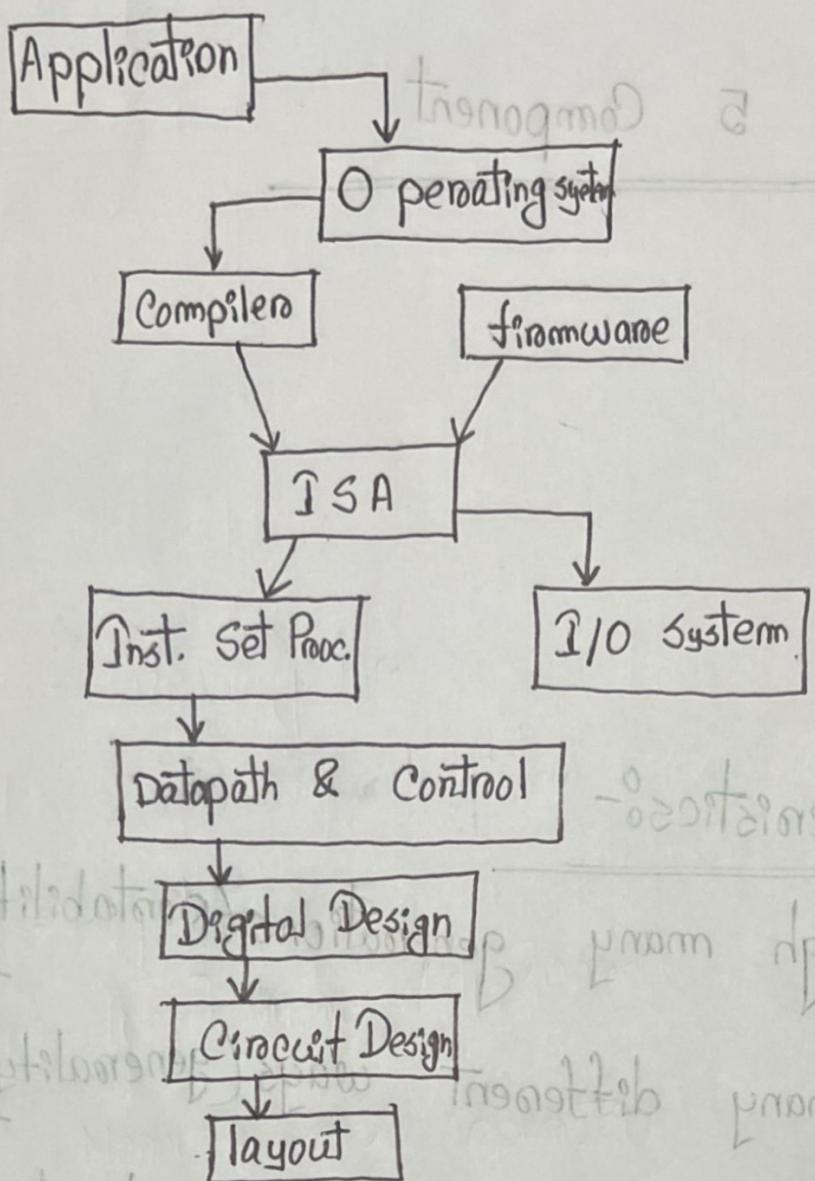


ISA characteristics:-

- ① Lasts through many generations (portability)
- ② Used in many different ways (generality)
- ③ Provides convenient functionality to higher level
- ④ Permits an efficient implementation at lower level

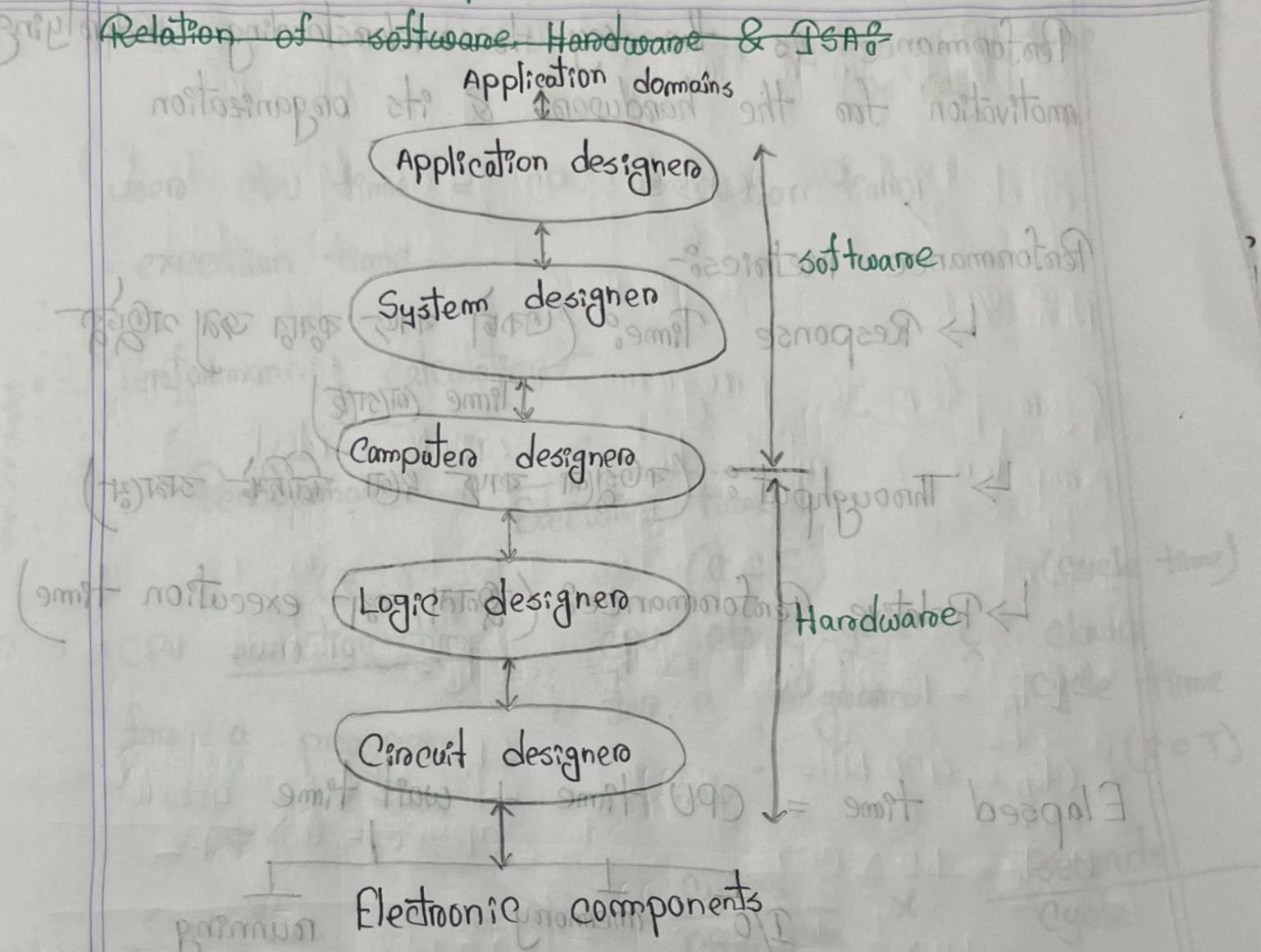
~~Layers of Architecture~~

Relation between Software, hardware & ISA



Lec-1

Layers of Computer Architecture



Lecture-2 Computer Architecture to understand

Performance is the key to understanding underlying motivation for the hardware & its organization

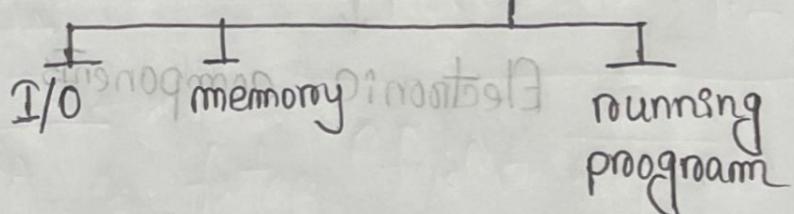
Performance Metrics:-

↳ Response Time: (একটি কার্ড ক্ষেত্র জন্য মাত্রিক
Time (নেগেচ))

↳ Throughput: (কর্তৃপক্ষে কার্ড থেকে নির্দিষ্ট অর্থ (প্র))

↳ Relative Performance: (ট্রুলি ক্ষেত্র execution time)

$$\text{Elapsed time} = \text{CPU time} + \text{wait time}$$



CPU time = user CPU time + system CPU time
(OS call)

User CPU time = CPU execution time
= execution time

Performance calculation:-

$$\text{Performance} = \frac{1}{\text{Execution time}}$$

$$\text{CPU execution time} = \text{CPU clock cycles for a program} \times \text{clock cycle time}$$

$$\frac{\text{seconds}}{\text{program}} = \frac{\text{cycles}}{\text{program}} \times \frac{\text{seconds}}{\text{cycle}}$$

$$1 \text{ Hz} = 1 \frac{\text{Cycle}}{\text{sec}} = 1 \text{ MHz} = 10^6 \text{ cycle/sec}$$

$$\text{clock rate} = \frac{1}{\text{clock cycle time}} = \frac{\text{cycle}}{\text{seconds}}$$

CPU Execution time = $\frac{\text{CPU clock cycles for a program}}{\text{clock rate}}$

clock rate - cycles per second. (always in Hz)

Math #1

Our favorite program runs in 10 seconds on computer A, which has a 400 MHz clock.

Ans:

$$ET_A = 10 \text{ seconds}$$

$$CR_A = 400 \text{ MHz}$$

$$ET_B = 1.2 \times CCT_A$$

$$= 1.2 \times \frac{1}{CR_A}$$

$$= \frac{1.2}{400 \times 10^6}$$

$$\therefore CR_B = \frac{1.2}{400 \times 10^6 \times 1.2} = \frac{1}{400 \times 10^6}$$

$$ET_B = 6 \text{ seconds}$$

$$CR_B = ?$$

$$CPUCC = 6 \times \frac{400 \times 10^6}{1.2}$$

$$CCT_A = \frac{1}{400 \times 10^6} = 2.5 \times 10^{-9}$$

$$CCT_B = 1.2 \times 2.5 \times 10^{-9}$$

$$= 3 \times 10^{-9}$$

$$CR_B = \frac{1}{3 \times 10^{-9}}$$

$$ET_A = 10 \text{ seconds}$$

$$CR_A = 400 \text{ MHz}$$

$$ET_B = 6 \text{ seconds}$$

$$CR_B = ?$$

$$CCC_B = 1.2 \times CCC_A$$

Now,

$$CCT_A = \frac{1}{CR_A} = \frac{1}{400 \times 10^6} = 2.5 \times 10^{-9}$$

$$CCC_A = \frac{ET_A}{CCT_A} = \frac{10}{2.5 \times 10^{-9}} = 4 \times 10^9$$

Then,

$$\begin{aligned} CCC_B &= 1.2 \times 4 \times 10^9 \\ &= 4.8 \times 10^9 \end{aligned}$$

$$\cancel{CCC_B} \cdot CCT_B = CCC_B \times ET_B$$

$$CCT_B = \frac{ET_B}{CCC_B} = \frac{6}{4.8 \times 10^9} = 1.25 \times 10^{-9}$$

$$CR_B = \frac{1}{CCT_B} = \frac{1}{1.25 \times 10^{-9}} = 800 \times 10^6 \text{ Hz}$$

$$= 800 \text{ MHz}$$

$CPI = \text{Cycles per Instruction}$

Performance equation II:

CPU Execution time for a program = $\frac{\text{Instruction count for a program}}{CPI} \times \frac{\text{Average clock cycle time}}{10^{-9}}$

Example 10-

Suppose we have two implementations of the same instruction set architecture (ISA) for some program:

- machine A has a clock cycle time of 10 ns. and a CPI of 2.0
- machine B has a clock cycle time of 20 ns. and a CPI of 1.2

a) which machine is faster for this program, and by how much?

b) If two machines have the same ISA, which of our quantities (e.g. clock rate, CPI, execution time of instruction, MIPS) will always be identical?

Ans^o-

$$CCT_A = 10 \text{ ns}$$

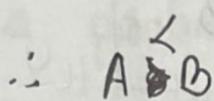
$$CCT_B = 20 \text{ ns}$$

$$CPIT_A = 2.0$$

$$CPIT_B = 1.2$$

$$\begin{aligned} ET_A &= 10 \times 2.0 \times 1 \\ &= 20 \text{ I} \end{aligned}$$

$$\begin{aligned} ET_B &= 20 \times 1.2 \times 1 \\ &= 24 \text{ I} \end{aligned}$$



Machine B is $\frac{24}{20}$ or 1.2 times faster than machine A.

Math 2 :-

A compiler design is trying to decide between two code sequence for a particular computer. The hardware designers have supplied the following facts:

	A	B	C
CPI	1	2	3

now CPI for this instruction class

Code Sequence	A	B	C
1	2	1	2
2	4	1	1

Instruction Counts for instruct class

- ① Which code sequence executes the most instructions?
- ② which will be faster?
- ③ What is the CPI for each sequence?

Ans :-

① Sequence 1 executes $(2+1+2) = 5$ instructions
" " " " $(4+1+1) = 6$ "

So, sequence 2 executes most instructions.

② We know,

CPU clock cycles for a program = CPI \times Instruction count

Now,

CPU clock cycles for Code sequence 1 is
 $= (2 \times 1) + (1 \times 2) + (3 \times 2)$
 $= 10$ cycles

CPU clock cycles for code sequence 2 is
 $= (4 \times 1) + (1 \times 2) + (1 \times 3)$
 $= 9$ cycles

∴ Sequence 2 will be faster.

③ For each sequence,

$$CPI_1 = \frac{10}{5} = 2$$

$$CPI_2 = \frac{9}{6} = 1.5$$

MIPS = Million Instructions Per Second

$$MIPS = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6}$$

Math #3

Instruction count (in billion) for each instruction class

Codeform	A	B	C
Compiler 1	5	1	1
Compiler 2	10	1	4

Assume that the computer's clock rate is 4GHz

① Which code sequence will execute faster according to MIPS?

② ----- " according to Execution time?

Ans:-

① We know,

$$\text{MIPS} = \frac{\text{Instruction Count}}{\text{Execution time} \times 10^6}$$

$$\text{MIPS}_1 = \frac{(5+1+1) \times 10^9}{2.5 \times 10^6} = 2800$$

$$\text{MIPS}_2 = \frac{(10+1+1) \times 10^9}{3.75 \times 10^6} = 3200$$

According to MIPS, compiler 1 is faster due to higher rating of 2.

(ii) We know, $\text{CPU clock Cycles} = \sum (\text{CPI}_i \times C_i)$

$$\text{CPU clock cycles}_1 = ((5 \times 1) + (1 \times 2) + (1 \times 3)) \times 10^9$$

Time of execution of program = 10×10^9 cycles

$$\text{CPU clock cycles}_2 = ((10 \times 1) + (1 \times 2) + (1 \times 3)) \times 10^9$$
$$= 15 \times 10^9 \text{ cycles}$$

$$\text{Execution time } 1 = \frac{10 \times 10^9}{4 \times 10^9} = 2.5 \text{ seconds}$$

$$\text{Execution time } 2 = \frac{15 \times 10^9}{4 \times 10^9 (1+1+0)} = 3.75 \text{ seconds}$$

According to execution time, compiler 1
faster program = $\frac{10 \times (1+1+0)}{15 \times 10^9} = 2.92 \text{ M}$

so compiler 1 is faster than compiler 2

Lecture-3

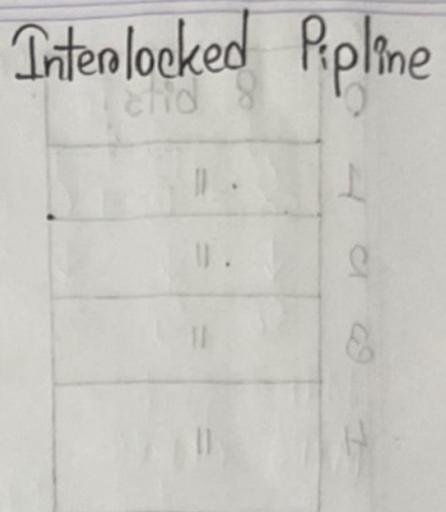
MIPS → Microcomputers without

Stages.

→ performance ↑

→ cost ↓

→ design time ↓



There are 3 class of instructions in MIPS:

① R-type:- Arithmetic Instructions

② I-type:- Memory refer instructions

③ J-type:- Jump instructions

There are 32 registers (0-31) in MIPS. MIPS

uses word addressing (4 bytes / 32 bits at a time)

Memory ← Register

	8 bits
0	"
1	"
2	"
3	"
4	"

0	32	bits	of	data
4	"	"	"	"
8	"	"	"	"
12	"	"	"	"

$s =$ save register

$t =$ temporary register

$\$ =$ register

$lw =$ load word (op word)

(Memory read operation)

Memory \rightarrow Registers

$sw =$ store word (op code)

(Memory write operation)

Registers \rightarrow Memory

At time we can only use 3 variables.

Exp:-

add \$50, \$51, \$52 ✓

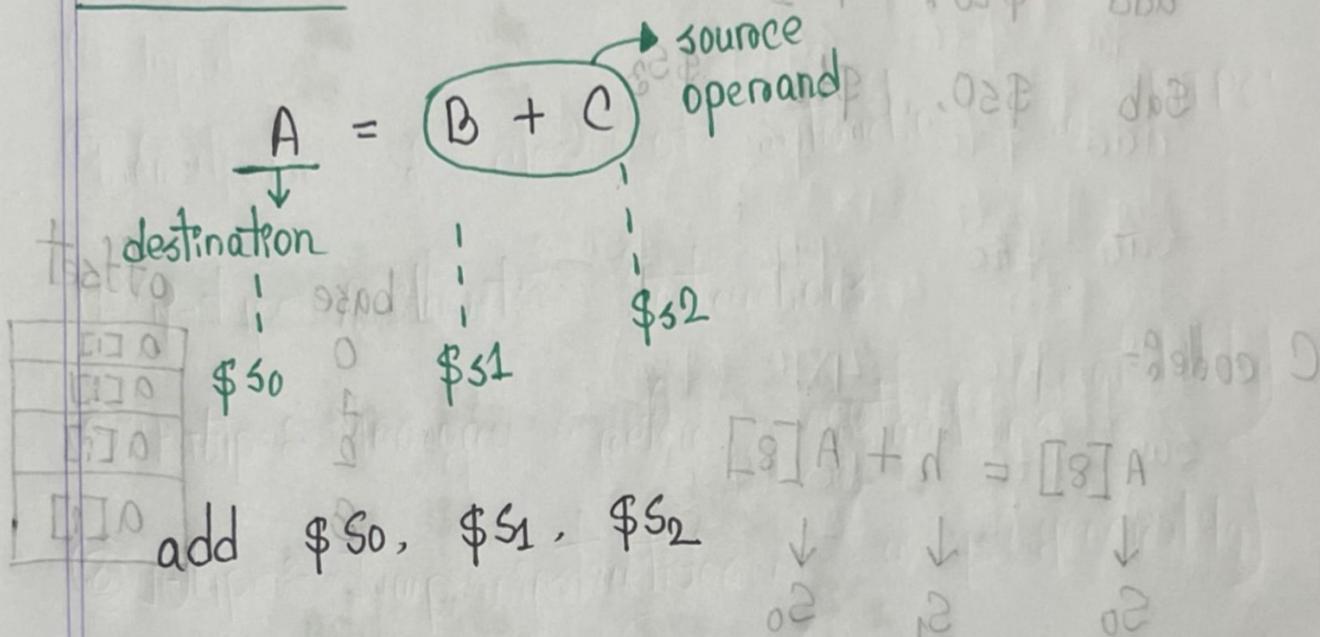
$$d - 5 + 8 = x$$

↓ ↓ ↓ ↓
5 5 8 0

add \$51, \$50, \$53, \$54 ✗

MIPS CODE :-

C code:-



MIPS CODE: —↑

C code:-

$$X = \boxed{B + C} - D$$

\downarrow \downarrow \downarrow \downarrow
 S_0 S_1 S_2 S_3

Ex:-

MIPS CODE :-

add \$t0, \$S_1, \$S_2

Sub \$S_0, \$t0, \$S_3

C Code :-

$$(Q + D) = \frac{A}{\downarrow}$$

base | offset
 0 1 2 3 4

C code:-

$$A[8] = h + A[8]$$

$$\downarrow \quad \downarrow \quad \downarrow$$

S_0 S_1 S_0

a[i]
a[i]
a[i]
a[i]

MIPS Code :-

MIPS Code :-

lw \$t0, 8x4(\$S0)

add \$t0, \$S1, \$t0

sw \$t0, 32(\$S0)

G Codes

lw → destination first

D - [] X + A = [] X

sw → destination last

↓ ↓ ↓ ↓
D D D D

C Code:-

$$A[12] = h + B[2]$$

↓ ↓
~~h~~
S₀ S₁

MIPS Code:-

lw \$t0, 4x2(\$S₂)

lw \$t0, 8\$S₂

add \$t0, \$S₁, \$S₂

sw \$t0, 12x4 (\$S₀)

sw \$t0, 48 (\$S₀)

C. Code :-

$$x[14] = A + x[16] - C$$

↓ ↓ ↓ ↓
 s_0 s_1 s_0 s_2

for init position1 < w/
for init position2 < w/

C Code :-

MIPS Code :-

lw \$t0, 4*x16(\$s0)
lw \$t0, 64(\$s0)
add \$t0, \$s1, \$s0
sub \$t0, \$t0, \$s2

$$[14] \oplus A + d = [16] A$$

↓ ↓
 t_0 t_0

MIPS Code :-

sw \$t0, 4*x14(\$s0)
sw \$t0, 56(\$s0)
add \$t0, \$t0, \$t0
sub \$t0, \$t0, \$t0

.otf w/
 .otf w/
 .otf bbs
 (otp) Hx14 .otf w/
 (otp) 8H .otf w/

C Code:-

$$x[7] = x[9] - A + C$$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 $s_0 \quad s_0 \quad s_1 \quad s_2 \quad 0c \quad 0c \quad 0c$

MIPS Code:-

lw \$t0, 4*x(\$s0)

lw \$t0, 36 \$s0

sub \$t0, \$s0, \$s1

add \$t0, \$t0, \$s2

sw \$t0, 4*x(\$s0)

sw \$t0, 28 \$s0

C code:-

$$A[8] = A[1] + A[2] + A[3] \times [5] \times$$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 $s_0 \quad s_0 \quad s_0 \quad s_0 \quad s_0 \quad s_0 \quad s_0$

MIPS Code:-

lw \$t0, 4\times 2(\$s0)

lw \$t0, 8(\$s0)

lw \$t0, 4\times 1(\$s0)

lw \$t0, 4(\$s0)

add \$t0, \$t0, \$t1

sw \$t0, 32(\$s0)

C code:-

$$A = B + C + D$$

$$\begin{matrix} \downarrow & \downarrow & \downarrow & \downarrow \\ s_0 & s_1 & s_2 & s_3 \end{matrix}$$

$$[e]A + [s]B = [s]A$$

$$\begin{matrix} \downarrow & \downarrow & \downarrow \\ o_0 & o_1 & o_2 \end{matrix}$$

MIPS Code:-

~~add~~ \$t0, \$s1, \$s2

add \$s0, \$t0, (\$s3) ~~exp~~ ofp wi

ofp ofp ofp wi

C code:-

$$E = F - A$$

$$\begin{matrix} \downarrow & \downarrow & \downarrow \\ s_0 & s_1 & s_2 \end{matrix}$$

$$(o_2 p) SxH \quad ofp o_0$$

$$ofp ofp ofp wi$$

MIPS Code:-

Sub \$t0, \$s1, \$s2, [s]A ~~+ [s]B = [s]A~~

~~Sub~~ \$t0, \$s1, \$s2, [s]A ~~+ [s]B = [s]A~~

~~ofp ofp ofp wi~~

~~ofp ofp ofp wi~~

C code:-

$$A[8] = B[6] + A[9]$$

$\downarrow \quad \downarrow \quad \downarrow$
 $S_0 \quad S_1 \quad S_0$

$$D + C + B = A$$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 $D \quad C \quad B \quad A$

MIPS Code :-

MIPS Code:-

lw \$t0, 4x9(\$S0)

lw \$t0, 36\$S0

add \$t0, \$t0, \$S1

sw \$t0, 4x8(\$S0)

sw \$t0, 32(\$S0)

$$A - 7 = E$$

$\downarrow \quad \downarrow \quad \downarrow$
 $S_0 \quad S_1 \quad S_0$

C code:-

$$A[i] = B[5] + C[i+3]$$

$$A[15] = B[5] + C[18]$$

$$\begin{array}{ccc} \downarrow & \downarrow & \downarrow \\ S_0 & S_1 & S_{12} \\ \hline t_1 & + & t_0 = t_0 \end{array}$$

$$i = 15$$

MIPS Code :-

MIPS Code:-

lw \$t₀, 4x18 (\$\$₂)

lw \$t₀, 72 \$\$₃

lw \$t₁, 4x5 (\$\$₁)

lw \$t₁, 20 \$\$₁

add, \$t₀, \$t₀, \$t₁

sw, \$t₂, 4x15 (\$\$₀)

sw, \$t₂, 60 \$\$₀

~~add \$\$₀ = \$\$₁ + \$\$₂~~

first 6 bit ৩৫০ রেখা then ২৫১ R-type

|| 6 " " 35 " " " T "

|| 6 " " 2 " " " J "

shift amount will always be zero.
shift

Last 6 bit ৩৫০ কি কোনো arithmetic function
কোনো

1	0	2	4	8	16
0	0	0	0	1	0
1	0	1	0	0	1

Example 0-

C Code 0-

$$A = B + C$$

$$\downarrow \quad \downarrow \quad \downarrow$$

$$S_0 \quad S_1 \quad S_2 \rightarrow \text{Registers}$$

$$16 \quad 17 \quad 18$$

add \$S_0, \$S_1, \$S_2

op	rs	rt	rd	shamt	func
0000000	17 \$P	18 \$2	16 \$0	00000	, 32

6 bits 5b 5b 5b 5b 6b

Example 2 0-

$$X = B + C - D$$

$$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$$

$$S_0 \quad S_1 \quad S_2 \quad S_3$$

add \$t0, \$S1, \$S2
 sub \$S0, \$t0, \$S3

add

OP	ns	nt	nd	shant	func
0	17	18	8	0	32

0000 0000 0000 0000 0000 0000
6bit 5bit 5bit 5bit 5bit 6bit

Sub

OP	ns	nt	nd	shant	func
0	8	19	16	0	42

0000 0000 0000 0000 0000 0000
6bit 5bit 5bit 5bit 5bit 6bit

Example 3:-

$$f = C - A$$

$$\downarrow \quad \downarrow \quad \downarrow$$

$$S_0 \quad S_1 \quad S_2$$

$$C + D = A$$

$$\downarrow \quad \downarrow \quad \downarrow$$

$$000 \quad 100 \quad 000$$

Sub \$S_0, \$S₁, \$S₂

OP	ns	nt	nd	shant	func
0	17	18	16	0	42

6bit 5bit 5bit 5bit 5bit 6bit

Identify the following MIPS instructions and write the corresponding MIPS assembly code.

000000 10001100101000000000100000

op	rs	rt	rd	shamt	func
000000	10001	17 s_1	18 s_2	16 s_0	32 1000000

6bit 5bit 5bit 5bit 5bit 6bit

$$A = B + C$$

$$\downarrow \quad \downarrow \quad \downarrow$$

$$s_0 \quad s_1 \quad s_2$$

add \$s0, \$s1, \$s2

$$A - C = ?$$

$$\downarrow \quad \downarrow \quad \downarrow$$

$$s_0 \quad s_1 \quad s_2$$

0	0	1	1	1	0
0	1	1	1	1	0

6bit 5bit 5bit 5bit 5bit 6bit

I-type & R-type mixed

lw \$t0 destination

sw \$t1 source

" " "

not

01

8

51

33

Exp:

C code:

$$A[5] = B[10] - A[2]$$

$$S_0/16 \quad S_1/17 \quad S_2/16$$

Assembly code

lw \$t0, ~~10~~ 2x4 (\$S0)

lw \$t1, 10x4 (\$S1)

sub \$t0, \$t0, \$t1

sw \$t0, 5x4 (\$S0)

bottom part of 2nd part

Machine Code :-

1st line

35	17	8	40
----	----	---	----

op

6 bit

rs

5b

rt

5b

offset

16 b

2nd line :-

$$[1]A - [0]A = [0]A$$

01/02 11/12 01/02

35	16	9	8
----	----	---	---

(02) H x 01

02 H x 01

3rd line

1tP 0tP

0tP 0tP

0	8	9	8	0	42
---	---	---	---	---	----

op

rs

rt

rd

shamt

func

4th line

43	16	8	20
----	----	---	----

op

rs

rt

offset

C.Code :-

$$X[2] = Y - X[4] + Z$$

↓ ↓ ↓ ↓
 \$0 \$1 \$0 \$2

Assembly Code :-

```

    lal $t0, 4x4($$0)
    sub $t0, $$1, $t0
    add $t0, $t0, $$2
    sw $t0, 2x4($$0)
  
```

Machine code :-

1st Line :-

35	16	8	16
op	rs	rt	offset

2nd Line :-

0	8	17	8	0	42
op	rs	rt	offset rd	shamt	fun

3rd line

0	8	18	18	0	32
op	res	rat	rad	shamt	func

4th line

	op	ros	rat	-offset top
43	16	(op) $\times 8$	8	8 <u>code</u> <u>main</u>

C. code :-

$$A[10] = B[1] + B[4] - A[9]$$

↓ ↓ ↓ ↓
 s_0 s_1 s_1 s_0

Assembly code :-

lw \$t_0, 9x4 (\$s_0)

lw \$t_1, 4x4 (\$s_1)

lw \$t_2, 1x4 (\$s_1)

add \$t_0, \$t_1, \$t_2

sub \$t_0, \$t_1, \$t_0

sw \$t_0, 10x4 (\$s_0)

Machine Code :-

1st line

op	ns	at	offset
35	16	8	36

6 bit 5b 5b 8b

2nd line

35	17	9	16
----	----	---	----

3rd line :-

35	17	10	4	A = [0] . A
		↓ 0C	↓ 1C	↓ 0C

4th line :-

op	rs	rt	rd	shamt	fune
0	9	10	9	0	32

6bit 5bit 5bit 5bit 5bit 6bit

5th line :-

op	rs	rt	rd	shamt	fune
0	9	8	8	0	42

6bit 5bit 5bit 5bit 5bit 9bit

6th line :-

op	rs	rt	offset
43	8	16	40

51 8 F1 80

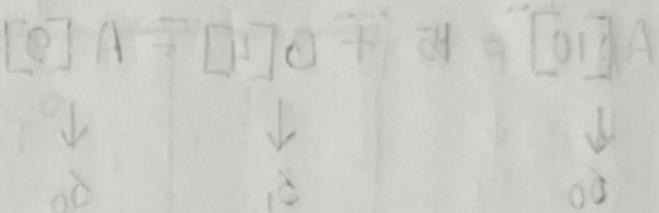
add:

C code:-

$$A = B + 12$$

$$\downarrow \quad \downarrow$$

$$\$0 \quad \$1$$



Assembly code:-

addi, \$50, \$51, 12

Machine code:-

op 6b	5b	5b	16b
op	rs	rt	offset
8	17	16	12

add: এই ~~কোড~~ destination টি আবশ্যিক কার্ড করা নি

00	10	01	00
00	10	01	00
00	10	01	00

C code :-

$$A[10] = 15 + B[4] - A[9]$$

↓ | ↓ | ↓ ; ↓ ↓
 \$0 | \$1 | \$0 ; \$C \$D

assembly code :-

lw \$t0, 9x4 (\$\$0)

lw \$t1, 4x4 (\$\$1)

addi \$t1, 15, \$t1

sub \$t0, \$t1, \$t0

sw \$t0, 10x4 (\$\$0)

Machine code :-

1st line

op

rs

rt

offset

35	16	8	36
6 bit	5 bit	5 bit	16 bit

2nd line

op	rs	rt	offset
35	17	9	16

6bit 5bit 5bit 16 bit

3rd line

op	rs	rt	offset
8	9	9	15

6bit 5bit 5bit 16 bit

4th line

op	rs	rt	rd	shamt	func
0	9	8	8	0	42

6bit 5bit 5bit 5bit 5bit 6 bit

5th line

op	rs	rt	offset
43	16	8	40

6bit 5bit 5bit 16 bit

C code:-

$$A = B - 12$$

↓ ↓

$s_0 \quad s_1$

12 → 00000000000000001100

13 → 11111111111110011
(+) 1

25 → 11111111111110100

Assembly Code:-

addi \$s0, \$s1, -12

Machine Code:-

001000	10001	10000	11111111111000
--------	-------	-------	----------------

OP 0 8 8 0

10000

01

8

01

01

1001

1001

1001

1001

Subi

C code:-

$$Y = X[i+2] + X[i+4] - 12$$

$$Y = X[16] + X[18] - 12$$

$$\begin{matrix} \downarrow & \downarrow & \downarrow \\ \$0 & \$1 & \$1 \end{matrix}$$

Assembly code:-

lw \$t0, 18x4 (\$\$1)

lw \$t0, 16x4 (\$\$1)

add \$t0, \$t0, \$t1

~~subi \$t0, \$t1, 12~~

subi \$s0, \$t0, 12

Machine code:-

1st line

op

rs

offset

42	16	8	72
6 bit	5 bit	5 bit	16 bit

2nd line

op	ns	rot	offset
42	16	9	64

6 bit 5 bit 5 bit 16 bit

3rd line

op	ns	rot	rd	rd	shamt	fun
0	8	9	8	0	32	

6 bit 5 bit 5 bit 5 bit 5 bit 16 bit

4th line

op	ns	rot	offset
34	8	15	120

6 bit 5 bit 5 bit 16 bit

J type:-

beq → branch equal

bne → branch not equal

C code:-

if ($i == j$) then

$$h = i + j$$

$$h = s_0 = 15$$

$$i = s_1 = 16$$

$$j = s_2 = 17$$

else

$$h = i - j$$

Assembly Code:-

beq \$s_1, \$s_2, Poner-Ta

sub \$s_0, \$s_1, \$s_2

Poner-Ta:

add \$s_0, \$s_1, \$s_2

Exit: 100

1st line :-

op	res	rot	loop	total
4	17	16		9

↳ assumed

$$i = p = j$$

$$i = l = j$$

2nd line :-

0	16	17	15	0	42
---	----	----	----	---	----

3rd line :-

0	16	17	15	0	32
---	----	----	----	---	----

4th line :-

2	100
---	-----

001 : fix

C Code :-

if ($a_0 \neq b$) then

$$C[12] = a + b$$

s_0 s_1 s_2

else

$$C[12] = a - b$$

s_0 s_1 s_2

$$s_2 = 17$$

$$s_0 = 15$$

$$s_1 = 16$$

$$t_0 = 8$$

Assembly Code :-

bne, ~~add~~ \$s_1, \$s_2, update

~~sub~~, \$^{t_0}, \$s_1, \$s_2

sw, \$t_0, 4x12 (\$s_0)

update :-

add, \$^{t_1}, \$s_1, \$s_2

sw, \$^{t_0}, 4x12 (\$s_0)

Exit: 100

8H	1	0	8	8H
fid 01	fid 02	fid 04	fid 03	fid 02

32 16 8 4 2 1

Machine code :-

1st line :-

op	rs	rt	offset
4	16	17	9

6 bit 5 bit 5 bit

(d = 10) fi

rd

rdt

d+

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

4th line :-

op	rs	rt	rd	shamt	func
0	16	17	9	0	32

6bit 5bit 5bit 5bit 5bit 6bit

5th line :-

op	rs	rt	offset
48	9	15	48

6bit 5bit 5bit 16 bit

6th line

op	offset
2	100

6bit 26 bit