

## Lecture - 3

Example:

```
int add(int a , int b){
```

```
    int c=20;
```

```
    c = c+a+b;
```

```
    return c; }
```

```
int main(){
```

```
    int a=5,b=10,res;
```

```
    res = add(a,b);
```

```
    res = res+a+b;
```

```
}
```

Here,  $a = \$50$ ,  $b = \$52$ ,  $res = \$53$

Solution:

addi \$50, \$50, 5

addi \$51, \$51, 10

jal Add

add \$52, \$v0, \$zero // \$52 = (v0 + 0)  
[For linking]

add \$52, \$52, \$50

add \$52, \$52, \$51

jr EXIT

Solution (Before):

addi \$sp, \$sp, -4

sw \$54, 0(\$sp)

Solution (Partial):

addi \$59, \$59, 20

add \$59, \$59, \$50

add \$59, \$59, \$51

add \$v0, \$59, \$zero

Solution (After):

lw \$59, 0(\$59)

addi \$SP,\$SP, 9  
jr \$ra

EXIT:

## MIPS to Binary (32-bit) Conversion

\* Chapter 7 MIPS এর ফার্মেট  
Binary তে convert করা হয়।

\* MIPS → একটি Instruction  
আছে অনেকগুলি ধরণের ৩ type-এ<sup>র</sup>  
যায়। :-

R-Type: (যখন Operand 3bit:-

1st destination, 2nd source এবং  
একই নথির R-type টল। কিন্তু  
সম্ভিক্ষণ ৩ টাইপ।  $\rightarrow$  add, sub, and,  
or, nor, sll, srl, slt, jr.

## I-Type: I-Type Constant

on Address 2nd ৰাখলো।

:- lw, sw, andi, ori, addi, beq, bne

## J-Type: Jump register নথে।

except jp.

:- j, jal.

## R-type MIPS Field:

Op	rs	rt	rd	shamt	funct
6bit	5 bit	5 bit	5 bit	5 bit	6 bits

$\hookrightarrow \text{total} = 32 \text{ bit}$

Op: Operation code / op code দৰি 2<sup>6</sup>.

rs: 1st Source register

rt: 2nd Source register

rd: destination register

shamt: shift Amount . কষ্ট শিফট  
বয়ে দেবি

funct: Function code

# ~~MIPS~~ Machine Language জনপ্রিয় ডেকুমেন্ট টেবিল কর্তৃত

২৫৮০১

MIPS machine language

Name	Format	OP	Example				Funct	Comments
add	R	0	18	19	17	0	32	add \$s1,\$s2,\$s3
sub	R	0	18	19	17	0	34	sub \$s1,\$s2,\$s3
lw	I	35	18	17	100			lw \$s1,100(\$s2)
sw	I	43	18	17	100			sw \$s1,100(\$s2)
and	R	0	18	19	17	0	36	and \$s1,\$s2,\$s3
or	R	0	18	19	17	0	37	or \$s1,\$s2,\$s3
nor	R	0	18	19	17	0	39	nor \$s1,\$s2,\$s3
andi	I	12	18	17	100			andi \$s1,\$s2,100
ori	I	13	18	17	100			ori \$s1,\$s2,100
sll	R	0	0	18	17	10	0	sll \$s1,\$s2,10
srl	R	0	0	18	17	10	2	srl \$s1,\$s2,10
beq	I	4	17	18	25			beq \$s1,\$s2,100
bne	I	5	17	18	25			bne \$s1,\$s2,100
slt	R	0	18	19	17	0	42	slt \$s1,\$s2,\$s3
j	J	2	2500					j 10000 (see Section 2.9)
jr	R	0	31	0	0	0	8	jr \$ra
jal	J	3	2500					jal 10000 (see Section 2.9)
Field size		6 bits	5 bits	5 bits	5 bits	5 bits	6 bits	All MIPS instructions 32 bits
R-format	R	op	rs	rt	rd	shamt	funct	Arithmetic Instruction format
I-format	I	op	rs	rt	address			Data transfer, branch format

→ OP and Funct table কোথা থাকে?

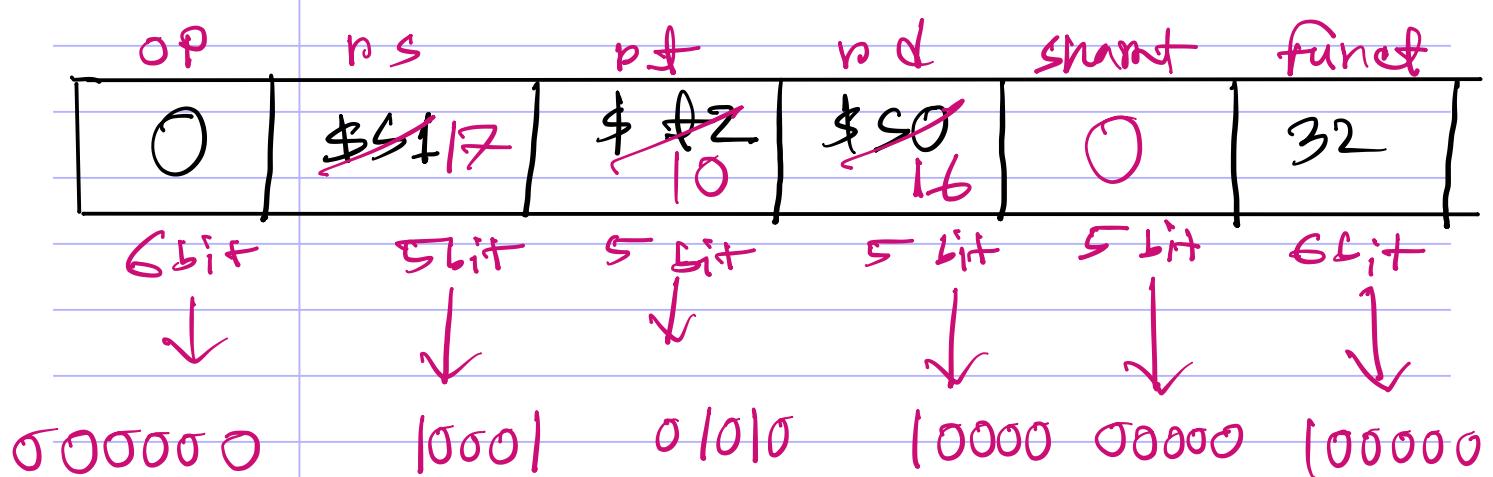
এবং

→ Register এর বাই জনপ্রিয় ডেকুমেন্ট টেবিল  
কোথা থাকে?

Name	Register number	Usage	Name	Register number	Usage
\$zero	0	the constant value 0	\$t8-\$t9	24-25	more temporaries
\$v0-\$v1	2-3	values for results and expression evaluation	\$gp	28	global pointer
\$a0-\$a3	4-7	arguments	\$sp	29	stack pointer
\$t0-\$t7	8-15	temporaries	\$fp	30	frame pointer
\$s0-\$s7	16-23	saved	\$ra	31	return address

Lets solve a problem:-

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



Enter the - Decimal value 108

and then Binary conversion

বাট্টা মাত্রা 17

(X) SLL \$40,\$53,20

OP	rs	rt	rd	shamt	funct
0	\$53 10	X 0	\$40 8	20	0
6 bit	5 bit	5 bit	5 bit	5 bit	6 bit

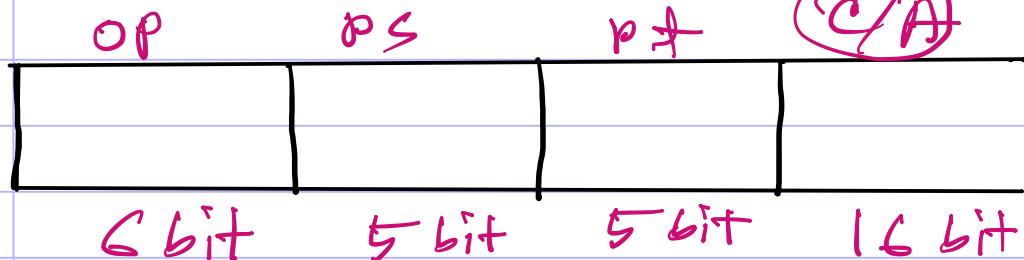
↓      ↓      ↓      ↓      ↓      ↓

000000    10011    0000000    01000    10100    0000000

(\*) JR \$ba

OP	rs	rt	rd	shamt	funct
0	JR 31	X 8	X 0	X 0	8
6 bit	5 bit	5 bit	5 bit	5 bit	6 bit

\* I-type MIFPS Field:



constant/  
Address

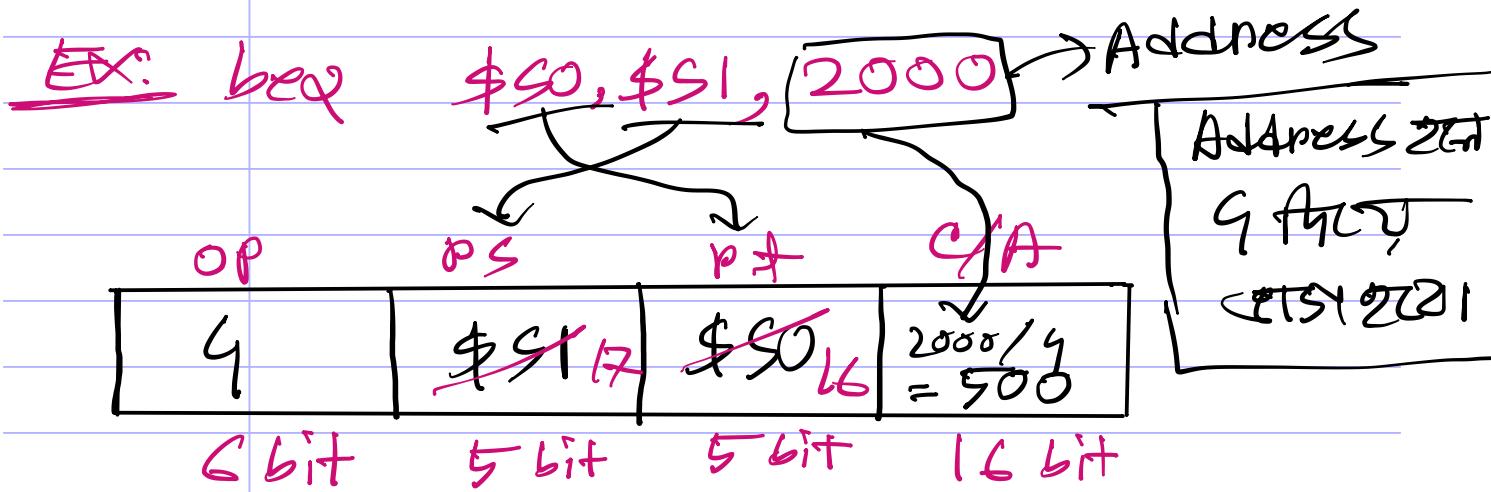
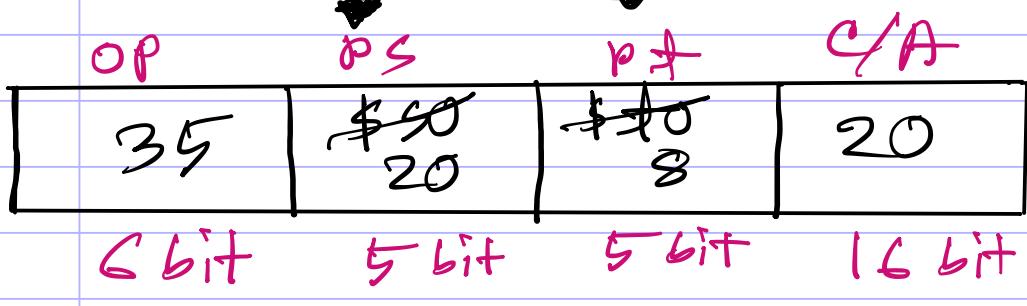
\* Example:

IW \$10, 20(\$50)

Ans:

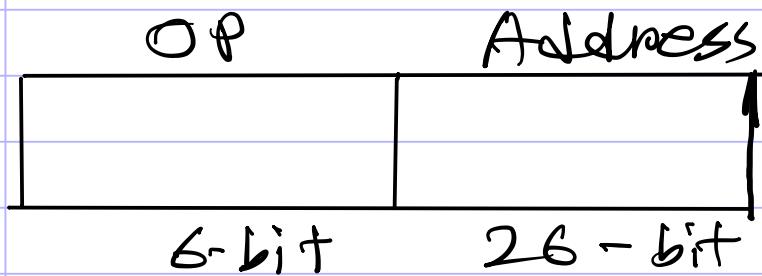
\* \* \*  
NOTE

I-type  
ISRT Always  
constant Address  
connect 2!

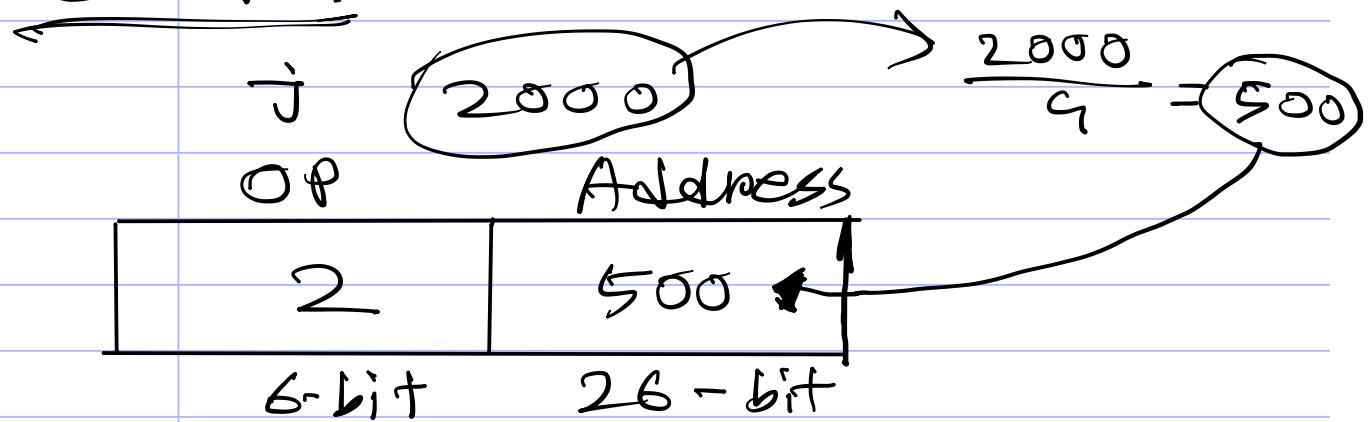




## J-type Field



example:



Maximum  $4 \times (2^{26} - 1)$  - এখন

-Jump করা যায়।