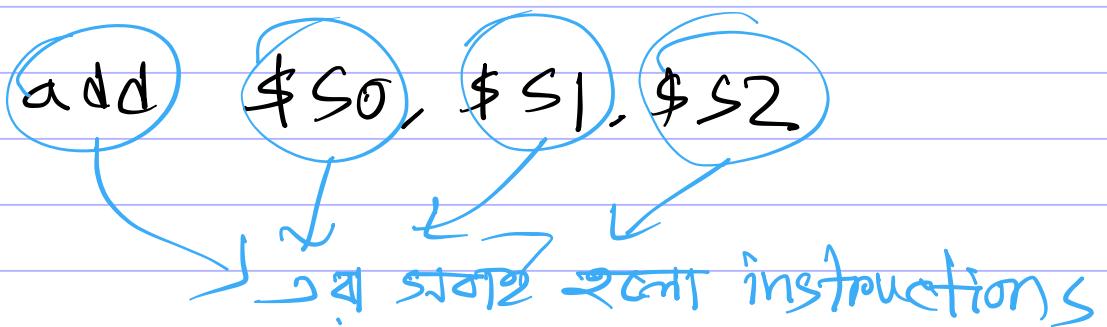


Lecture-2

* Instruction Set:

- MIPS ହେ ଏକ instruction architecture ହେଲା ।
→ ଯେତ୍ୟା କୋଣର କୋଣର code ହାଲିଛି ଲେଖିବା ହେବାରେ, ଯାମକିଛି କାହାଙ୍କିଲୋ instruction-set କୁ ବନ୍ଦ ହେବାରେ MIPS ହେ ।

୭ ।



→ \$0, \$2 ଏକଳ instruction ନିଯେ ଏ ଏକ instruction set ହେଲା
Instruction Set Architecture (ISA) ହେଲା

ISA (Instruction Set Architecture)

*** In this course we will learn
MIPS.

Now we will convert c code to MIPS.

Rules:

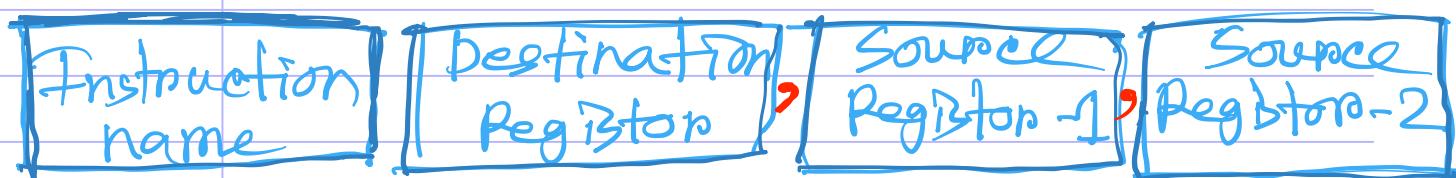
(*) MIPS → Only Instruction first

Instruction Order

(*) MIPS → Register 32 bit

(*) MIPS → Instruction first Order

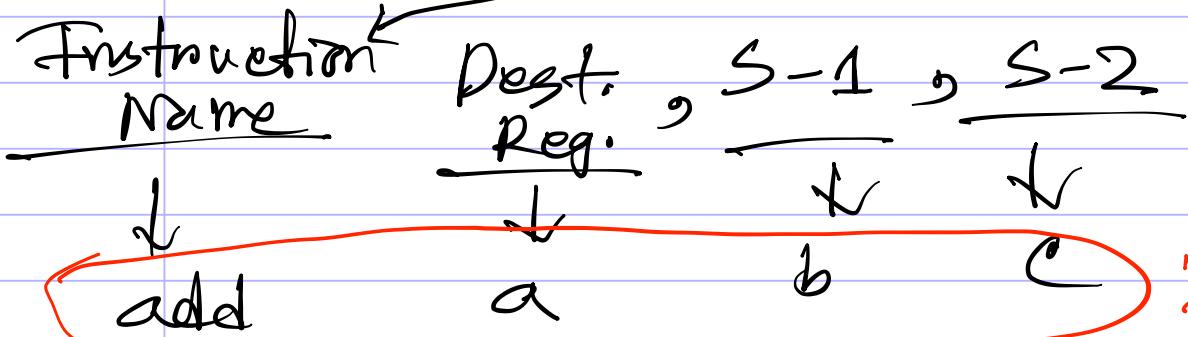
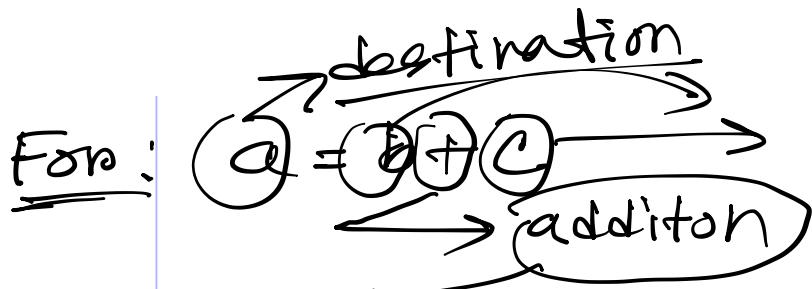
[All instruction has primarily 3 Operands:-



(*) Example: convert: $a = b + c$ this to MIPS.

Soln:

We know,



↳ Solution fe কিন্তু partially

কারণ । কারণ , যেসব প্রোগ্রাম , MIPS
এ Register use করা হচ্ছে । But
যেসব , Destination এর Source নাও
যাইছায় Register এর রূপে
Variable প্রক্রিয়ার করেছি ।

* So Register কার্য্য আছে যেসব
এখন যেকোন বিস্তৃত ক্ষিয়ৎ ।

→ MIPS → Register আছে 32 bit ,
এই instruction format use করি ।

⊕ Register: Register এর যোগ্য
(\\$) sign আছে ।

→ [Dollar Sign]



যোব্দক অন্যাং 3-Catagory \rightarrow 3-Register চিনিব। যার মধ্যে 10 ft Register আছে।:-

① Saved Register: Variable সংরক্ষণ।

সবচেয়ে Always Saved Register কোথায়। MIPS \rightarrow saved Register আছে 8-টি : $\Rightarrow \$S_0, \$S_1, \$S_2, \dots, \S_7 .

[Program-এর Register সংরক্ষণ।

Declare কোথা থাকে তাদের সবচেয়ে
Saved Register প্রদর্শন করা হয়।]

② Temporary Register: MIPS \rightarrow 10 ft

Temporary Register আছে।:-

$\Rightarrow \$t_0, \$t_1, \$t_2, \dots, \t_9

[এর কাম Saved Register এর
মতোই but oft temporary (container)
চিহ্ন প্রদর্শন হয়।]

③ Zero Register: (১৫ অন্তর্ভুক্ত), এটি

zero value Represent করে, \$zero.

Now lets comeback to the question

$$\rightarrow d = b + c$$

Instruction name Dest., S-1, S-2
 Ref.
add ~~d~~ ~~b~~ ~~c~~ ~~\$S1~~ ~~\$S2~~

[Question \rightarrow অধিবর্ণ অ, b, c এর
প্রতি Register কেন ফিল, যেখানে
না হাতে কেন Assume করে নি।]

\rightarrow এখন, Assume করার পদ্ধতি:-

left to Right \rightarrow Assign

করব।]

Ex: $a = b + c + d$ Convert this to MIPS.

Sol:

~~add \$S0, \$S1, \$S2, \$S3~~

↳ এটি কোন কারণ MIPS →

কারণ ৩টিrd register instruction নিয়ে আরও মাত্র।

তেব্বে,

$$a = b + c + d$$

$$\Rightarrow \textcircled{I} \quad a = b + c$$

$$\Rightarrow \textcircled{II} \quad a = a + d$$

] ~~একটা~~ solve

ব্যবহার করে।

অন্যান temporary
register ফ্লেচ দেওয়া
হচ্ছে।

Ans:

add \$S0, \$S1, \$S2 // $a = b + c$

add \$S0, \$S0, \$S3 // $a = a + d$

$$\begin{array}{c}
 \$S9 \\
 \swarrow \quad \searrow \\
 \textcircled{*} \quad \textcircled{P} = (\textcircled{g+h}) - (\textcircled{i+j}) \\
 \downarrow \qquad \qquad \qquad \downarrow \\
 \textcircled{t_0} \quad \textcircled{t_1} \quad \textcircled{t_2} \quad \textcircled{t_3} \\
 \end{array}
 \quad \text{convert this}$$

to MFPS.

Sol'n:

[\rightarrow इन, योक्तव्या Temporary Register
use करें solution करें।]

$$\begin{array}{c}
 P = (\textcircled{g+h}) - (\textcircled{i+j}) \\
 \downarrow \qquad \qquad \qquad \downarrow \\
 \textcircled{t_0} \quad \textcircled{t_1}
 \end{array}$$

add \$t_0, \$S0, \$S1 // $t_0 = g+h$

add \$t_1, \$S2, \$S3 // $t_1 = i+j$

sub \$S9, \$t_0, \$t_1 // $P = t_0 - t_1$

(Ans)

(*) int a,b,c,d,e,F,gh,i,j,k,----;
जिनके लिए temporary Register की ज़रूरत है।

उसके लिए temporary Register की ज़रूरत है।

→ इनमें से, कौनसा Variable कौनसा
अधिक संख्या के लिए इसका उपयोग करेगा।

ପିକ୍ରାନ୍ତ ଯେତେ stack ଦ୍ୱାରା । ଏହି
Register ମାତି ଏବଂ ଓପାଟିନ ମାତି ।

MIPS

* Arithmetic -

→ add , add i
→ sub

add i

କୌଣସି କରନ୍ତୁ

constant
କରନ୍ତୁ

ex. $a = b + 4$

constant ମାତି
କରନ୍ତୁ
1st 2nd
ପାଇଁ ୧୦

$$a = b + 6$$

* Constant କରନ୍ତୁ :-

instruction : dest , S-1 , Constant
reg.

addi

\$50, \$51 , 6

instruction : dest , constant, constant

soft କିମ୍ବା ଅନ୍ତର କରନ୍ତୁ 2ଟି
constant MIPS → Allowed ନାହିଁ ।

Ex: $a = 10$

Soln:

addi \$S0, \$zero, 10 // $a = 0 + 10$

* Array Related →
 → Load word (lw)
 → Store word (sw)

* Logical → and, or , andi, ori
 nor

Ex: $a = b - 2$

Sub: \$S0 - \$S1, 2 → এটি কোন কার্য
মিহি → Sub: মাই
এটি কার্য addi ফর্মে
possible.

addi \$S0, \$S1, -2 (Ans)

* left shift and right shift shift
⇒ ans instruction use ->>

$a = b \ll 3$ [3 ans left shift]

$a = b \gg 4$ [4 bits right shift]

(100

1st shift Right: 0100

2nd u u: 0011

3rd u u: 0001

→ Shift Left logical → sll

→ Shift Right logical → srl

* conditional branch →
 - == beq
 - != bne
 - < slt

* Conditional Jump →
 - Jump Reg.: jr
 - j
 - Jump and Link: jal

[Assign on Lecture 2 videos]

instruction এর মতো কোটির মতো

Slide ▲ Assignment করে !