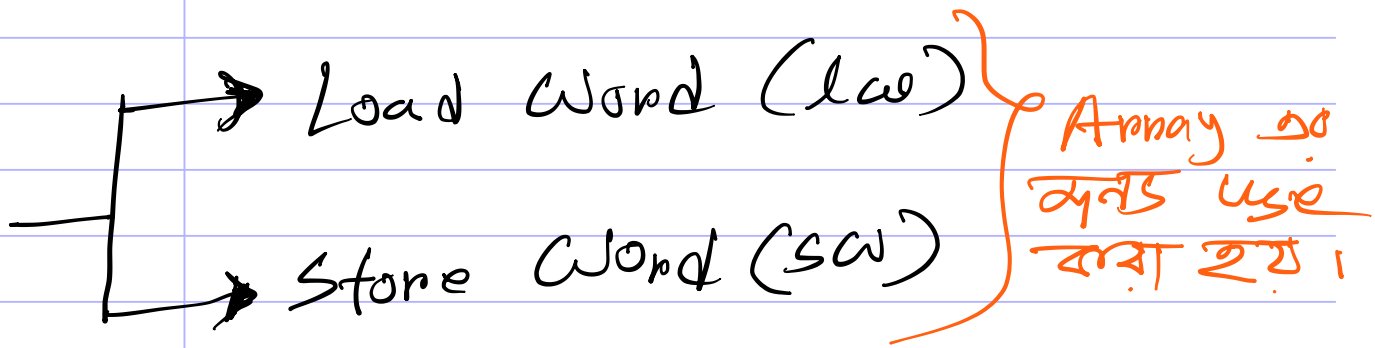
Lec-4.

✳ Return to MIPS Again.

Today we will Learn about Load word and store word.

→ Used For Array.



✳ Spilling Register: আমাদের MIPS এ 32 টি Register আছে। But variable যদি 32 টির বেশি হয় তখন extra variable কোথায় যাবে?
এই জন্যে extra less commonly used variable গুলো Register এ

যাদের কে stack এ রাখা হয়।
এটাই spilling Register.

[The process of putting less commonly used variables (or those needed later) into memory (stack) is called spilling register.]

Array

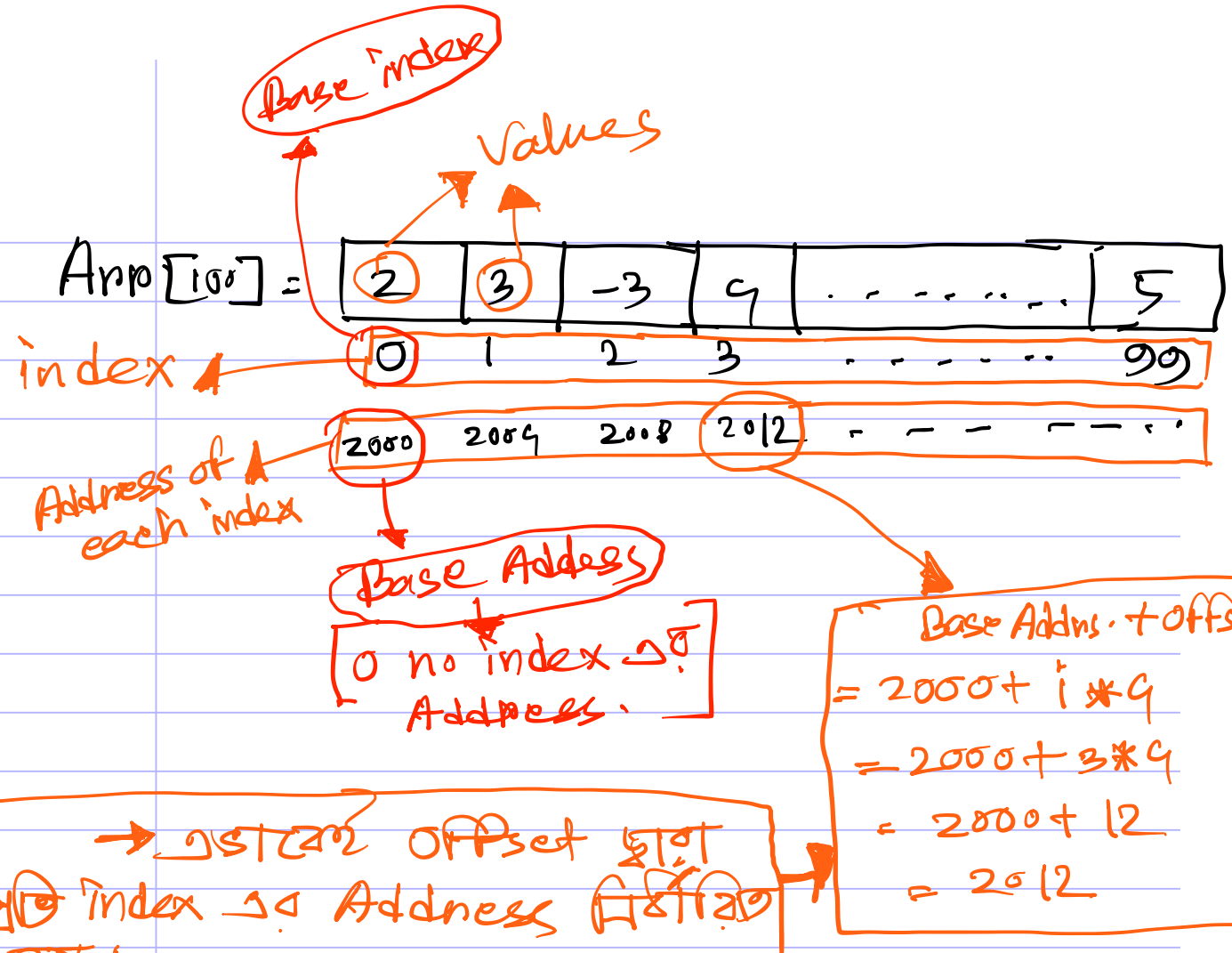
Array তে:- Value index, address এর offset থাকে।

* base Address: 0 no index এর Address
* offset: Array এর 0-no. index থেকে বাকি index গুলোর দূরত্ব।

equation:

Offset of index i = i * 4

4 byte for 32 bit-Register



→ এইভাবে offset দ্বারা যেকোন index এর Address নির্ণয় করা যায়।

⊛ Read = lw [data from memory to register]

⊛ Write = sw [data from register to memory]

A[i] = base Address + offset of i

System:

(lw/sw register, offset (base Address of Array))

register → s_i or t_i

offset → 4 × i

Array এর কাজ:

A[n] → 4 * n ($S0) ← if $S0 - Base Address

(*) A[12] = h + A[8]

Here, base Address is = $S3
h = $S2

↳ এখানে h + A[8] কে read করে sum করে value টাকে A[12] এ write করতেছি।

Soln:

lw $t0, 32($S3) // 4x8 — এখান থেকে read হচ্ছে A[8]
এর মানে, $t0 = A[8] হবে lw এর জন্য

add $t1, $S2, $t0 // এর মানে, $t1 = h + A[8]

sw $t1, 48($S3) // 4x12
এর মানে, A[12] = $t1
অর্থাৎ $t1 এর value A[12] এ Store বা Write হয়েছে।

Example:

```
h = 230;
A[10] = h + B[i]
```

Suppose, Compiler will Assign A(base addrs) into $S0, B(base Addrs) into $S1, h into $S2 and i into $S3.

Soln:

```
addi  $S2, $zero, 230
```

→ এখানে একটা special problem create হয়েছে। problem টি হলো offset related। এখানে (i) কোনো integer নয়। এটি একটি variable তাই, (4*i) এর value আমরা সরাসরি পাবনা। তাহলে, (4*i) এর offset value কিভাবে বের করব?

= এর জন্য আমরা left shift করব।

আমরা জানি যে,

→ binary bit এলোকে 0 বার left shift করলে $2^0 = 1$ পুন হয়

→ 1 বার left shift এ $2^1 = 2$ পুন হয়।

→ 2 " " " $2^2 = 4$ " "

→ n " " " 2^n পুন হয়।

So,

***এভাবে i কে 2 বার left shift করলে (i*4) এর value পেয়ে যাব → কিন্তু সেটাকে একটি saved/temporary register এ রাখব। Then use করব।

So,

```
addi  $s2, $zero, 230 // n=230
sll   $t0, $s3, 2 //
lw    $t1, $t0($s4) //
```

i কে 2 বার left shift করে যে value টি $t0 এ রাখা হচ্ছে।

$t0 = i >> 2

এ: মানে হলো [//$t1 = B[i]] $t0

[$t1 → B[i] read करेगा]

add $t2, $s2, $t1 // $t2 = h + B[i]

⇒ $t2 = h + $t1

sw $t2, 40($s0) // A[10] → $t2 का value store करेगा

A[10] = $t2

⇒ A[10] = h + B[i]

(Ans)