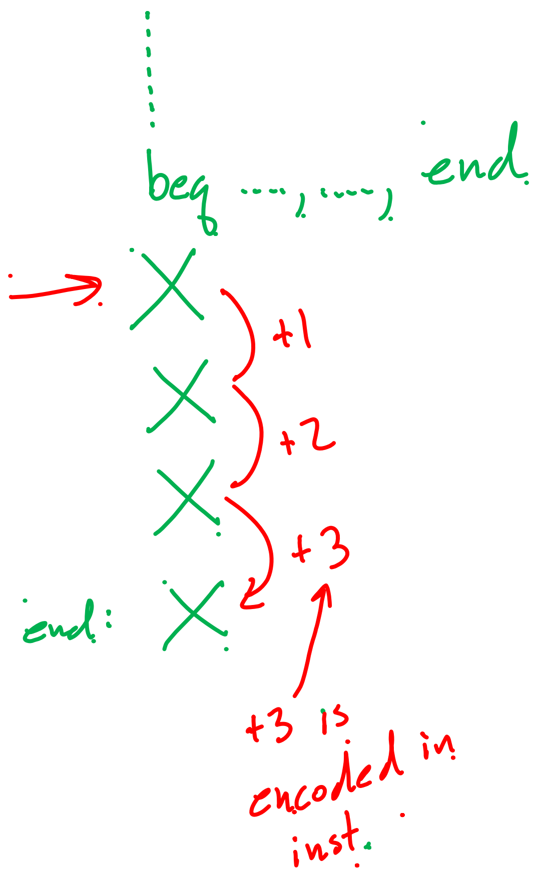
Branches

of. inst to branch target

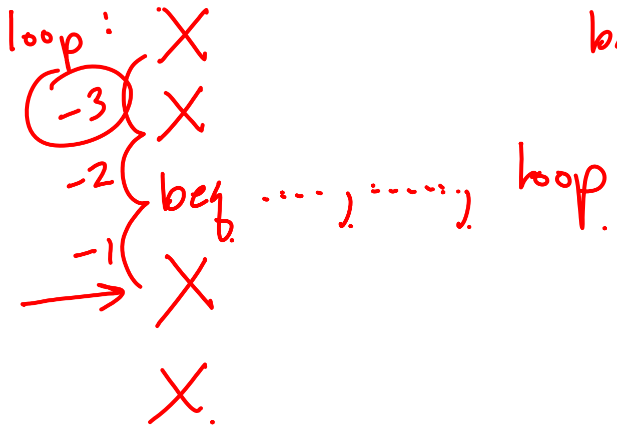
Operation	rs	rt	rd	shamt	funct	# comment
beq	3	2				# if ($2 == $3) goto loop
bne	3	2				# if ($2 != $3) goto loop
jr		3			8	# goto $3
j						# goto loop

Computing Branch Targets



1. start at the inst. after the branch

2. count the number of inst. to branch target (forward in code is positive, backward in code is negative)



MIPS Example 4

$s1 = sum
$s0 = i
$t0 = &A

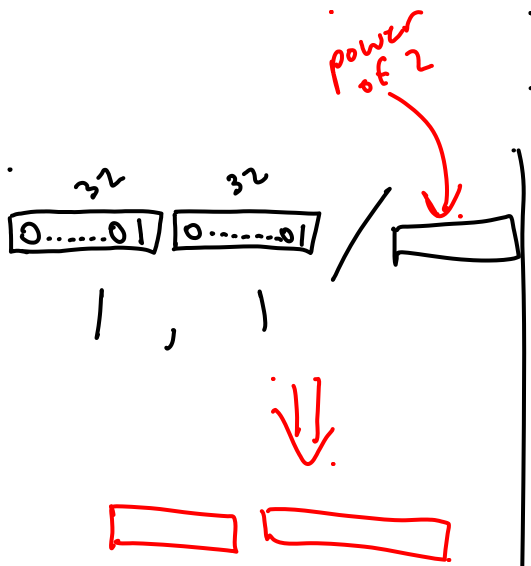
```
sum=0;
for(i=0; i < 100; i++)
      sum += A[i];
-----------------------------------------------
-
      sum = 0;
      i = 0;
loop:     if (  i>=100     ) goto  end
_______;

      sum += A[i];

      i++
      goto  loop ;
end:
```

```
1.  addi $s1, $0, 0
2.  addi $s0, $0, 0

3.  loop: slti $t1, $s0, 100
4.        beq  $t1, $0, end
                    i
5.  sll  $t1, $s0, 2
6.  add  $t1, $t1, $t0      &A
7.  lw   $t2, 0($t1)
8.  add  $s1, $s1, $t2

9.  addi $s0, $s0, 1
10. j loop
end:
```

Translating into machine code

assembly inst	op	rs	rt	rd	shamt	func	comment

MIPS Instructions



Key Points:

1. lw/sw ___ , ___ (___)

must be a reg. — must be an immediate — must be a register

2. constants are limited to 16-bits

3. branch distance is limited