5.14 slt: Set on less than instruction

Set on less than instruction

The ***set on less than*** (***slt***) instruction sets a register to 1 if the value held in the register is less than the value held in anothe otherwise the register is set to 0. Ex: `slt $t1, $t4, $t5` sets $t1 to 1 if $t4's value is less than $t5's value, otherwise $t1 slt instruction is typically used with the beq or bne instructions to branch to an instruction based on the relational comparis registers.

PARTICIPATION ACTIVITY 5.14.1: Set on less than instruction.

Start 2x speed

```
        slt $t2, $t0, $t1          5 < 10
        bne $t2, $zero, BLabel     1 != 0
        ...
BLabel: addi $t5, $t5, 1
```

Register file

$t0	5
$t1	10
$t2	1
$t3	

PARTICIPATION ACTIVITY 5.14.2: Set on less than.

Assume initial register values of:

- $t0: 40
- $t1: 20
- $t2: 55

1) What is $t5 after the instruction below?

```
slt $t5, $t0, $t2
```

- ○ 0
- ○ 1

2) What is $t5 after the instruction below?

```
slt $t5, $t2, $t1
```

- ○ 0
- ○ 1

3) Determine if the branch instruction is taken or not taken.

```
slt $t4, $t0, $t2
beq $t4, $zero, BLabel
```

- ○ Taken
- ○ Not taken

4) Determine if the branch instruction is taken or not taken.

```
slt $t5, $t0, $t1
beq $t5, $zero, BLabel
```

- ○ Taken
- ○ Not taken

5) Determine if the branch instruction is taken or not taken.

```
slt $t6, $t1, $t2
bne $t6, $zero, BLabel
```

- ○ Taken
- ○ Not taken

A brief note from your instructor:

Once again, please do note that BLT, BLE, BGT and BGE are pseudoinstructions and not real MIPS; therefore you will not use them on exams (though Mars does accept them). Table 5.14.1 provides a good example of their equivalent translati MIPS instructions, if you need it to help you study.

Branch pseudoinstructions: blt, ble, bgt, bge

Common comparisons used for branch instructions include less than (<), less than or equal (≤), greater than (>), and greate (≥). However, MIPS natively supports only two branch instructions: beq (branch on equal) and bne (branch on not equal). Ar either a beq or bne can implement any comparison <, ≤, >, ≥. However, the slt + beq/bne approach is non-intuitive to many p Thus, MIPS supports the following pseudoinstructions.

- A ***branch on less than*** (***blt***) instruction branches if the first register is less than the second.
- A ***branch on less than or equal*** (***ble***) instruction branches if the first register is less than or equal to the second.
- A ***branch on greater than*** (***bgt***) instruction branches if the first register is greater than the second.
- A ***branch on greater than or equal*** (***bge***) instruction branches if the first register is greater than or equal to the seconc

A MIPS assembler will convert each pseudoinstruction into the indicated two native instructions. The assembler uses a ter to store the result of an slt instruction, which is used in the following beq or bne instruction. The ***$at*** (or ***assembler tempora*** reserved for use by the assembler to hold temporary values needed to implement pseudoinstructions.

Table 5.14.1: Branch pseudoinstructions and equivalent native instructions.

Branch pseudoinstruction	Native instructions
`blt $t0, $t1, BLabel`	`slt $at, $t0, $t1` `bne $at, $zero, BLabel`
`ble $t0, $t1, BLabel`	`slt $at, $t1, $t0` `beq $at, $zero, BLabel`
`bgt $t0, $t1, BLabel`	`slt $at, $t1, $t0` `bne $at, $zero, BLabel`
`bge $t0, $t1, BLabel`	`slt $at, $t0, $t1` `beq $at, $zero, BLabel`

PARTICIPATION ACTIVITY 5.14.3: Branch pseudoinstructions.

Refer to the above table.

1) beq is a pseudoinstruction.
   - ○ True
   - ○ False

2) slt is a native instruction.
   - ○ True
   - ○ False

3) A blt $t0, $t1 is replaced by an slt $at,

$t0, $t1 followed by a bne.

- ○ True
- ○ False

4) A bgt $t0, $t1 is replaced by an slt $at, $t0, $t1 followed by a bne.

- ○ True
- ○ False

5) A bge $t0, $t1 is replaced by an slt $at, $t0, $t1 followed by a beq.

- ○ True
- ○ False

6) A ble $t0, $t1 is replaced by an slt $at, $t1, $t0 followed by a beq.

- ○ True
- ○ False

Table 5.14.2: Instruction summary: slt.

Instruction	Format	Description	Example
slt	`slt $a, $b, $c`	Set on less than: Write 1 to register $a if value held in register $b is less than value held in register $c, and otherwise writes 0.	`slt $t1, $t5, $t6`

CHALLENGE

ACTIVITY 5.14.1: Set on less than, and branch pseudoinstructions.

Start

1

Implement blt to branch when $t0 < $t1.

2

```
slt ▼  $t0 ▼ , $t0 ▼ , $t0 ▼
bne ▼  $at ▼ , $zero ▼ , L1 ▼
add ▼  $t2 ▼ , $t2 ▼ , $t2 ▼
L1:
```

Registers

$t0	1
$t1	2
$t2	5

3

1 2 3

Check Next

Provide feedback on this section