7.2 Jump/branch immediates

Most parts of an assembly instruction are straightforwardly translatable to a machine instruction field. An operation like ad opcode like 00100, each register, like $t1, has a specific encoding like 01001, etc. However, the immediate field for a jump c instruction is more involved.

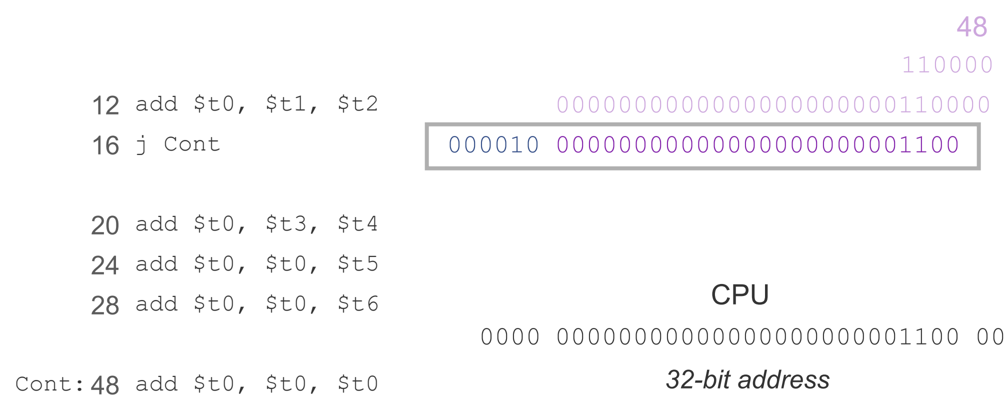
Jump immediates

A jump assembly instruction (j or jal) jumps to a label representing an address in instruction memory. An address is 32 bits machine instruction only has a 26-bit immediate field. Because an address is always a multiple of 4, the rightmost two bits and thus are omitted, meaning those 26 bits represent 28 bits. To form a 32-bit address, the processor copies the uppermc jump instruction's address. Thus, a jump instruction can only jump to addresses whose uppermost 4 bits are the same as t jump instruction.

Determining a label's address is discussed in another section on assemblers.

PARTICIPATION ACTIVITY 7.2.1: Determining the immediate field for a jump instruction.

Start 2x speed



PARTICIPATION ACTIVITY 7.2.2: Jump instruction immediate.

In the binary addresses below, the .. means enough 0's to form the indicated number of bits.

1) For instruction `j Label`, the address of Label is determined to be 40. What is the machine instruction's immediate field? (Each choice is 26 bits)
   - ○ 0..0101000
   - ○ 0..01010
   - ○ 0..0101

2) For instruction `j Label`, the address of Label is determined to be 2004. What is the machine instruction's immediate field? (Each choice is 26 bits) (Hint: 2004 is 0..011111010100 in binary).
   - ○ 0..011111010100
   - ○ 0..0111110101
   - ○ 0..01111101010000

3) Instruction `j Label` is at address 0000..1000 (32 bits). Given an immediate field of 0..0111 (26 bits), what address will the CPU construct?
   - ○ 0000..0111 (32 bits)
   - ○ 0000..011100 (26 bits)

- 0000..011100 (32 bits)

4) Instruction `j Label` is at address 0110..1000 (32 bits). Given an immediate field of 0..0111 (26 bits), what address will the CPU construct?
   - 0000..011100 (26 bits)
   - 0000..011100 (32 bits)
   - 0110..011100 (32 bits)

Branch immediates

A branch assembly instruction (beq or bne) also branches to a label representing an address, but a branch's machine instru 16-bit immediate field. 16 bits is too few to specify an actual address. Thus, that 16-bit field instead indicates an offset fror instruction's address. Offsets must be a multiple of 4, so the rightmost two bits are always 00, and thus omitted. Branches to addresses reachable by an 18-bit offset from the current address, meaning a range of -2^{17} (-131,072) to $+2^{17}-4$ (131,068 range is OK because branches are usually to nearby addresses.

PARTICIPATION ACTIVITY 7.2.3: Determining the immediate field for a branch instruction.

Start 2x speed

```
12 add $t0, $t1, $t2
16 beq $t1, $t2, Cont

20 add $t0, $t3, $t4
24 add $t0, $t0, $t5
28 add $t0, $t0, $t6

Cont: 48 add $t0, $t0, $t0
```

48 - 20 = 28

11100

00000000000011100

```
000100 01001 01010  0000000000000111
```

CPU

```
  00000000000011100   28
+ 00000000000000000000000000010100   20
  00000000000000000000000000110000   48
```

32-bit address

Why is the offset from the next instruction's address?

The processor's hardware keeps track of the current instruction's address using a hardware component called a program counter (PC). When fetching the current machine instruction from memory, the processor immediately adds 4 to the PC, in preparation for fetching the next instruction. If the current instruction branches, the offset is added to that PC + 4 value.

An address specified as an offset to the PC is called a ***PC-relative address***

PARTICIPATION ACTIVITY 7.2.4: Branch instruction immediate.

In binary addresses below, the .. means enough 0's to form the indicated number of bits.

1) For instruction `beq $t0, $t1, Label`, beq's next instruction's address is 40, and Label's address is 60. What is the offset in decimal, before any adjustments for filling the 16-bit immediate field?
   - ○ 20
   - ○ 40
   - ○ 60

2) For instruction `beq $t0, $t1, Label`, the beq instruction's address is 40, and

Label's address is 60. What is the offset in decimal, before any adjustments for filling the 16-bit immediate field?

- 16
- 20
- 24

3) For a branch instruction, the offset is determined to be 32 (before adjustments). What will the 16-bit immediate field be?

- 0..010
- 0..01000
- 0..0100000

4) For instruction `beq $t0, $t1, Label`, the beq instruction's address is 40, and Label's address is 60. What is the machine instructions 16-bit immediate field?

- 0..010000
- 0..0100
- 0..0101

5) For instruction `beq $t0, $t1, Label`, the beq instruction's address is 40, and Label's address is 32. What is the offset in decimal, before any adjustments for filling the 16-bit field?

- 4
-

-8

-12

CHALLENGE ACTIVITY 7.2.1: Jump/branch immediates.

Start

For instruction: j Label
The address of Label is determined to be 8.

Enter the last 4 bits of the 26-bit machine instruction's immediate field.

00000000 00000000 000000 Ex: 1001

1	2	3	4	5	6

Check Next

1 2 3 4 5 6

Provide feedback on this section