MITx 6.004.2x

Computation Structures 2: Computer Architecture

Help

Course Progress Dates Discussion

Course / 9. Designing an Instruction Set / Lecture Videos (52:28)

< Previous ✓ ✓ ✓ ✓ ✓ Next >

LE9.5

Bookmark this page

Calculator

LE9.5.1: Branch Instructions

1/1 point (ungraded)

- Summary of Instruction Formats (PDF)
- Beta Documentation (PDF)

Consider the execution of a short program that loops to sum the elements of an array with 4 elements. The first element of the array is stored at location 0×2000.

```
  . = 0               // first instruction is at location 0
  ADDC(r31,array,r0)  // r0 = pointer to next array element
  ADDC(r31,4,r1)      // r1 = number of array elements remaining
  ADDC(r31,0,r2)      // r2 = accumulated sum
loop:
  LD(r0,0,r3)         // load next value from array
  ADD(r3,r2,r2)       // add to sum
  ADDC(r0,4,r0)       // increment pointer to next word
  SUBC(r1,1,r1)       // decrement counter
  BNE(r1,loop,r31)    // loop if more elements to go
  ST(r2,result,R31)   // write result to memory
  // execution stops here

  . = 0x2000
array:
  LONG(1)             // array[0] = 1
  LONG(2)             // array[1] = 2
  LONG(3)             // array[2] = 3
  LONG(4)             // array[3] = 4
result:
  LONG(0)             // where result will be stored
```

Program execution starts with the first instruction and halts after execution of the ST instruction.

(A) What value does the assembler give the label "loop"? 0×000C ✓ Answer: 0xc

(B) After execution, number of times LD is executed? 4 ✓ Answer: 4

(C) After execution, value left in r0? 0×2010 ✓ Answer: 0×2010

(D) After execution, value left in r1? 0×0 ✓ Answer: 0

(E) After execution, value left in r2? 0×0A ✓ Answer: 10

(F) After execution, value left in r3? 0×04 ✓ Answer: 4

Explanation
"loop:" labels the fourth instruction of the program and the first instruction is at location 0. So the value of the symbol "loop" is 0xC.
At the end of the loop, the loop counter (R1) is decremented and if it's still non-zero, we branch back for another loop iteration. The initial value for R1 was 4, so the LD instruction is executed 4 times.
The array pointer (R0) is incremented by 4 on each loop iteration. The initial value of the pointer was 0×2000 and there were four loop iterations, so the value left in R0 is 0×2010.
The loop only terminates when the value of the loop counter (R1) is 0. So at the end of execution, the value left in R1 is 0.
R2 is the accumulated sum of the array values = 1 + 2 + 3 + 4 = 10.
R3 is used to hold the value of the array element accessed during each iteration. The last array element accessed had the value 4.

The encoding for the OPCODE, RC and RA fields of a branch instruction is just like the encodings for other E

Calculator

instructions. Figuring out the value for the 16-bit constant field takes a little more work. The offset value is the number of words between the instruction following the branch (ST in this example) to target instruction (LD in this example). Positive values indicate a forward branch to a subsequent location with a higher address; negative offset values indicate a backward branch to a location with a lower address.

In this example, we'd start counting instructions backwards from the store instruction until we reached the LD instruction. Since it's a backwards branch, we'd encode the count as a negative number in the 16-bit constant field of the BNE instruction.

(G) What is the binary encoding for `BNE(r1,loop,r31)`? 0b01110111111000011111 ✓ Answer: 0×77E1FFFB

Explanation
Here's a diagram showing how the offset of -5 is calculated:

```
offset    instruction
             . = 0               // first instruction is at location 0
             ADDC(r31,array,r0)  // r0 = pointer to next array element
             ADDC(r31,4,r1)      // r1 = number of array elements remaining
             ADDC(r31,0,r2)      // r2 = accumulated sum
          loop:
-5           LD(r0,0,r3)         // load next value from array
-4           ADD(r3,r2,r2)       // add to sum
-3           ADDC(r0,4,r0)       // increment pointer to next word
-2           SUBC(r1,1,r1)       // decrement counter
-1           BNE(r1,loop,r31)    // loop if more elements to go
 0           ST(r2,result,R31)   // write result to memory
```

The 32-bit binary encoding of the BNE instruction is

```
       opcode  RC    RA    constant (= -5)
       011101 11111 00001 1111 1111 1111 1011
```

Submit

Answers are displayed within the problem

Discussion

Hide Discussion

Topic: 9. Designing an Instruction Set / LE9.5

Add a Post

Show all posts by recent activity

[Staff] paid for verified, showing as honor code
I paid to take this class in verified mode, but it shows as honor code. Neil Berry
8

Value of R31
I thought I understood that the value of R31 was hardwired as all 0's, but it seems to be all 1's in some of the exercises. How do you...
6

R31?
Dear Staff.. In earlier discussion it is clearly mention that R31 is hardwired to zero but in BNE(r1,loop,r31) we are assigning PC value ...
5

Reg[Rc] ← PC
Hi, I'd like to ask about the branch BNE(r1,loop,r31), in wich RC is R31, as far as I understood R31 is read only, it can't store data so t...
3

No video with FIrefox
Please help. I am using Windows XP, SP3 and Firefox 42. I cannot get the lecture videos to play on my computer. What do I need to ...
12

Reg[Rc] ←Mem[EA]
"The location in memory specified by EA is read into register Rc" Does this mean: Rc register now has memory address stored or th...
2

Value of RC in this branch
Dear Staff, The following is given as the explanation to the encoding of binary encoding for BNE(r1,loop,r31) opcode RC RA constan...
6

BNE

Calculator

BNE instruction is supposed to work like that NPC = 0×1c+4 = 0×20 if (REG[R1]!=0) pc = NPC + 4*offset if the condition is true, the...

hallo
Ich bin mohammed hany elemam .
2

reversing the bits in r0
Hi everybody, In the assembly program provided in the " Introduction to Bsim" section what does it mean to reverse the bits in r0 ?
2

[STAFF] Xseries Status
When I took the first part of this course, it was listed as part of the "Foundations of Computer Science X series" however I see no m...
8

Alternative video lectures
If like me you find the video lectures for *Programmable Machines* too fast to cope with at the first pass, and find the 0.5x slow-do...
4

Hey
Hows everyone doing my name is Tim from San Diego California
1

< Previous Next >

edX

About
Affiliates
edX for Business
Open edX
Careers
News

Legal

Terms of Service & Honor Code
Privacy Policy
Accessibility Policy
Trademark Policy
Sitemap
Cookie Policy
Your Privacy Choices

Connect

Idea Hub
Contact Us
Help Center
Security
Media Kit

GET IT ON Google play
Download on the App Store

Calculator

Calculator