<u>Help</u>

selfpoised >

<u>Course</u> <u>Progress</u> <u>Dates</u> <u>Discussion</u>

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





⊞ Calculator

LE9.3.1: ALUC Instructions

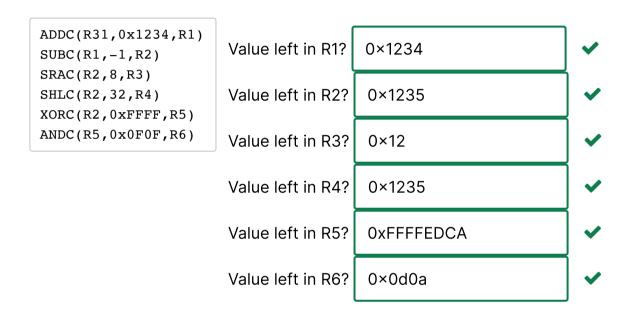
1.0/1.0 point (ungraded)

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

For the Beta instruction sequence shown below, indicate the 32-bit two's complement values of the specified registers after the sequence has been executed by the Beta. The effect of the instructions is cummulative, later instructions use the values stored by earlier instructions.

You can find detailed descriptions of each Beta instruction in the "Beta Documentation" handout -- see link above. Remember that register values and the ALU use a 32-bit two's complement representation.

Hint: You can enter answers in hex by specifying a "0x" prefix, *e.g.*, 17 could be entered as "0x11". Usually one would enter addresses, values in memory, etc. using hex. You can also use a "0b" prefix to enter a binary value, e.g., "0b10001".



Submit

LE9.3.2: Do we need SUBC?

1.0/1.0 point (ungraded)

A 6.004 student here at MIT has proposed that all SUBC instructions of the form

SUBC(Rx,const,Ry)

can be replaced by an ADDC that uses the negative of the constant:

ADDC(Rx,-const,Ry)

Is the student on to something? Will this transformation work for all SUBC instructions?

Yes, the transformation will work for all SUBC instructions

No, there are SUBC instructions with no ADDC equivalent

Submit

LE9.3.3: ALUC Instruction Encoding?

1.0/1.0 point (ungraded)



Please give the 32-bit binary encoding for each of the ALUC instruction shown below.

Note that when translating the symbolic form of the ALUC instruction into binary, the program (called the "assembler") we run will only use the low-order 16 bits of the value provided by the second operand, even if that value requires more than 16 bits to represent correctly. Our assembler is dumb:

One other note: we've been using the symbol R0 to represent register 0. The assembler knows to treat the Rx symbols as shorthand for the decimal constant x, e.g., R17 is a more readable alternative to specifying the constant 17.

(A) 32-bit encoding for ADDC(R31,0x1234,R1)?
(B) 32-bit encoding for xORC(R1,-17,R0)?
(C) 32-bit encoding for CMPLEC(R12,40000,R0)?
(Db1101100000001100100

(D) 32-bit encoding for ADDC(R1,R2,R3)?

0b110000000110000100

Submit

Discussion

Topic: 9. Designing an Instruction Set / LE9.3

Hide Discussion

Add a Post

Show all posts

✓ by recent activity ✓

< Previous

Next >

⊞ Calculator



edX

About

Affiliates

edX for Business

Open edX

Careers

News

Legal

Terms of Service & Honor Code

Privacy Policy

Accessibility Policy

<u>Trademark Policy</u>

<u>Sitemap</u>

Connect

<u>Blog</u>

Contact Us

Help Center

Media Kit

Donate













© 2021 edX Inc. All rights reserved.

深圳市恒宇博科技有限公司 <u>粤ICP备17044299号-2</u>