

**Video explanation of solution is provided below the problem.**

**For all Beta related questions, you should make use of the [Beta documentation](#), the [Beta Instruction Summary](#), and the [Beta Diagram](#).**

## Beta Assembly

4/7 points (ungraded)

For each of the Beta instruction sequences shown below, indicate the values of the specified quantities after the sequence has been executed. Consider each sequence separately and assume that execution begins at location 0 and halts when the HALT() instruction is about to be executed. Also assume that all registers have been initialized to 0 before execution begins. Remember that even though the Beta loads and stores 32-bit words from memory, all addresses are *byte addresses*, i.e., the addresses of successive words in memory differ by 4.

Fill in requested values left after execution of each segment, or "CAN'T TELL" where appropriate.

```
1.      . = 0
        LD(R31, c, R0)
        ADDC(R0, b, R0)

        HALT()

        . = 0x200
a:      LONG(0x100)
b:      LONG(0x200)
c:      LONG(0x300)
```

**Value left in R0 (HEX): 0x**



**Value assembler assigns to the symbol "c": 0x**



2.

```
. = 0  
BR(. + 4, R0)  
HALT()
```

**Value left in R0: 0x**

3.

```
. = 0  
LD(R31, x, R0)  
CMOVE(0, R1)  
  
loop: ANDC(R0, 1, R3)  
      ADD(R3, R1, R1)  
      SHRC(R0, 1, R0)  
      BNE(R0, loop)  
      HALT()  
  
x:    LONG(0x0FACE0FF)
```

**Value left in R0: 0x****Value left in R1: 0x**

4.

```
. = 0  
CMOVE(0x1000, SP)  
PUSH(SP)  
  
HALT()
```

**Value left in SP (HEX): 0x****Value pushed onto stack (HEX): 0x**



Submit

\* Partially correct (4/7 points)

## Beta Assembly

[Start of transcript. Skip to the end.](#)



As presented in lecture, in this course, we use a simple 32-bit processor called the Beta.

The Beta works on 32-bit instruction and data words.

However, the addresses in memory are specified in bytes.

## Video

[Download video file](#)

## Transcripts

[Download SubRip \(.srt\) file](#)

[Download Text \(.txt\) file](#)

## Discussion

Hide Discussion

**Topic:** 10. Assembly Language, Models of Computation / WE10.1

Add a Post

Show all posts ▼	by recent activity ▼
<input checked="" type="checkbox"/> <a href="#">BEQ definition - edit?</a>	4 ▼
<input checked="" type="checkbox"/> <a href="#">WE10 - Runtime error when attempting 'PUSH(SP)'</a> <a href="#">When I try to run these instructions in BSim: . = 0 CMOVE(0×1000, SP) PUSH(SP) HALT() I get the f...</a>	7 ▼
<input checked="" type="checkbox"/> <a href="#">Words layout in memory.</a> <a href="#">Hi. Suppose we have following instruction &gt; x: LONG(0×0FACE0FF) It's memory layout (Little Endia...</a>	2 ▼
<input type="checkbox"/> <a href="#">Value Left in R0</a> <a href="#">I'm confused with the first conclusion that R0 contains 0×300 after the first instruction. My unders...</a>	3 ▼
<input checked="" type="checkbox"/> <a href="#">Where did this 8 come from?</a>	8 ▼