

Final

110321054

12/26/2022

陳學綾

(You may answer in Chinese or English.)

1. 20%

Write an assembly program to compute the average of an array. The starting address of the array is stored in ARRAY, and its size is stored in SIZE. Assume  $SIZE > 0$ . Assume DIV instruction is available.

DIV Ra, Rb, Rc =====  $Ra = Rb / Rc$

.orig x3000

# code for computing the average of ARRAY[]

# store the answer in AVG

# ASSUME DIV Ra, Rb, Rc is available

# i.e.  $Ra = Rb / Rc$

HALT

~~ARRAY~~ .BLKW 1 # address of array ← array 起始位置

~~SIZE~~ .BLKW 1 # size of array ← 有幾個元素  $R_1 + 2 + 3 + 4$

AVG .BLKW 1 # the average

.END

最後答案存這  $R_2$   
 $R_3 + 4$

2. 20%

There are four characters, 'ABCD', which are to be pushed in and popped out from a stack. So the sequence "push push push push pop pop pop pop" will convert ABCD to DCBA if the first 'push' picks up 'A' and the second picks up 'B',... etc. Write the sequence to convert from

a) ABCD to ADCB

b) ABCD to BCAD

用 push, pop 組成, 做不到就是不可能

(a) push pop u u u, pop u u u pop pop  
A B C

3. 10%

The mechanism of invoking an TRAP service routine is actually invoking a function through a table. What is the benefit of such in-direct invoking?

為什麼間接呼叫 TRAP

TRAP 23

DCBA

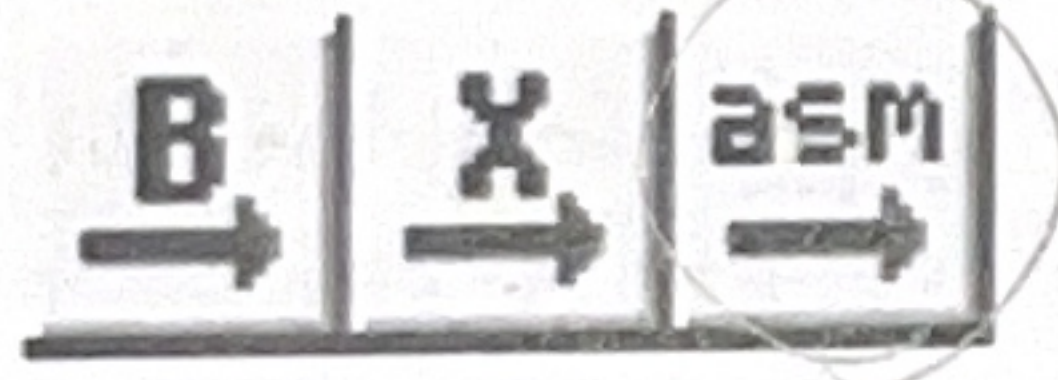
BCA  
D  
B



4. 25%

To use the LC-3 simulator, we first need to use the LC3edit to write and compile your program to get the object file for the LC-3 simulator to execute. Please answer the following questions.

- a) (5%) When the program is written in assembly code, which button shown below should be used to generate the object code?



用書白子

- b) (20%) A snapshot of the simulator is shown below. What information the "PC" register keep? Which button should be pressed for executing the program all the way? Which button should be pressed for step execution? How do you move the arrow from x3000 to x3005?

**C LC3 Simulator**

File Execute Simulate Help

1 2 3 4 Jump to: x3000

R0	x0000	0	R4	x0000	0	PC	x3000	12288
R1	x0000	0	R5	x0000	0	IR	x0000	0
R2	x0000	0	R6	x0000	0	PSR	x8002	-3276
R3	x0000	0	R7	x0000	0	CC	Z	

→ x3000	0000000000000000	x0000	NOP
x3001	0000000000000000	x0000	NOP
x3002	0000000000000000	x0000	NOP
x3003	0000000000000000	x0000	NOP
x3004	0000000000000000	x0000	NOP
x3005	0000000000000000	x0000	NOP
x3006	0000000000000000	x0000	NOP
x3007	0000000000000000	x0000	NOP
x3008	0000000000000000	x0000	NOP
x3009	0000000000000000	x0000	NOP
x300A	0000000000000000	x0000	NOP
x300B	0000000000000000	x0000	NOP
x300C	0000000000000000	x0000	NOP
x300D	0000000000000000	x0000	NOP
x300E	0000000000000000	x0000	NOP
x300F	0000000000000000	x0000	NOP

Ready 0 instructions executed Idle



5. 15%

Given the following assembly code and the data stored in memory addresses as shown below, what is the result (in hexadecimal) of R1 after executing the program?

```
.ORIG x3000
AND R1, R1, x0
AND R4, R4, x0
ADD R4, R4, x2
LEA R2, x3100
LOOP LDR R3, R2, x0
ADD R2, R2, x1
ADD R1, R1, R3
ADD R4, R4, x-1
BRp LOOP
HALT
.END
```

Address	data
x3100	x00A0 160
x3101	x00AF 175

R1 = 0  
R4 = 0  
R4 = 2  
R2 = x00A0  
R3 = x00A0

16進制 -1

A  
B  
C  
D  
E  
F

R1 00A0  
00A1  
00A2  
00A3  
00A4  
00A5  
00A6  
00A7  
00A8  
00A9  
00AA  
00AB  
00AC  
00AD  
00AE  
00AF  
00B0  
00B1  
00B2  
00B3  
00B4  
00B5  
00B6  
00B7  
00B8  
00B9  
00BA  
00BB  
00BC  
00BD  
00BE  
00BF  
00C0  
00C1  
00C2  
00C3  
00C4  
00C5  
00C6  
00C7  
00C8  
00C9  
00CA  
00CB  
00CC  
00CD  
00CE  
00CF  
00D0  
00D1  
00D2  
00D3  
00D4  
00D5  
00D6  
00D7  
00D8  
00D9  
00DA  
00DB  
00DC  
00DD  
00DE  
00DF  
00E0  
00E1  
00E2  
00E3  
00E4  
00E5  
00E6  
00E7  
00E8  
00E9  
00EA  
00EB  
00EC  
00ED  
00EE  
00EF  
00F0  
00F1  
00F2  
00F3  
00F4  
00F5  
00F6  
00F7  
00F8  
00F9  
00FA  
00FB  
00FC  
00FD  
00FE  
00FF

00A0 H  
00BD  
00CA H  
00D0 H  
00E0  
00F0  
00F1  
00F2  
00F3  
00F4  
00F5  
00F6  
00F7  
00F8  
00F9  
00FA  
00FB  
00FC  
00FD  
00FE  
00FF

6. 20%

True or false.

- In an operation system, a privileged program also has high priority for execution.
- A user program cannot access memory address space reserved for privileged programs.
- With memory-mapped I/O, each operation of an I/O device needs a designated I/O instruction for CPU to use to communication with the I/O device.
- With interrupt I/O, CPU does not have to wait for an I/O device to complete its execution. The I/O device sends a signal to CPU to inform its completion.

R4 = 2

R2 = x00A0 2

R3 = x00A0

R1 = 00A0  
00A1

10  
15  
15 H  
0  
1  
2  
3  
4