

LC-3 Assembly Programming Examples & Tips

Assembly Errors

Example:

```
ADD R6, R2, #1 ;increment index reg by 1.  
Op Dest Src1 Src2 Comment
```

Potential Compiler errors:

- Immediate argument too large
➢ E.g. ADD R1,R2,#1023
- Improper number or type of arguments
➢ E.g. ADD R1,R2
ADD R3,R3,NUMBER
NOT R1,#7

CIT 593

2/9

Assembly Errors (contd ..)

Examples:

```
LD R6, NUMBER ;R6 <- value at location named NUMBER.  
Op Dest LABEL Comment
```

Note: Using labels eliminates calculating offset

Potential Errors

- Label Number is not declared
- Address (associated with label) more than 256 from instruction
➢ Can't use PC-relative addressing mode
- Address is in range that user code cannot access

CIT 593

3/9

Load Operation

For the code snippet below what is value of R1, R2, R3 ?

```
.ORIG x3000  
LD R1, A  
LEA R2, A  
LDI R3, A  
HALT ; give back control to OS  
A .FILL x0005 ;Initialize a location name A with value 5  
.END
```

R1 = 5

R2 = address that A symbolically represents

R3 = Contents of location 5 i.e. M[x0005]

➢ However, this illegal as in LC3 user code cannot access location 5

CIT 593

4/9

Example 1

```
.ORIG x3000
LEA R1, VAL1
LDR R2, R1, #0
LDR R3 R1, #1
NOT R4, R3
ADD R4, R4, #1
ADD R5, R2, R4
BRzp L1
STR R3, R1, #2
BRnzp DONE
L1: STR R2, R1, 2
DONE: HALT
VAL1: .FILL #25
VAL2: .FILL x000D
.END
```

Recognize the problem ?

➤ Example 7 from hw 2

What will the symbol table look like for this program ?

CIT 593

5/9

How do compare two values are same?

Example 1: R1 == -1

```
ADD R2, R2, #1
NOT R2, R2
ADD R2, R2, #1 ;2's comp of R2
ADD R1, R1, R2
BRz GOTO
```

GOTO: ...

...

Example 1: R1 == 256 ?

```
LD R2, VAL
NOT R2, R2 ;!1
ADD R2, R2, #1 ;!2
ADD R1, R1, R2
BRz GOTO
```

...

...

GOTO: ...

...

VAL .FILL x0100

Can we avoid !1 & !2 ?

CIT 593

6/9

Example 2: 7.15 from Text

Assume integers are stored starting at x4000 and sequence terminates with the value x0000

What does this program do ?

```
.ORIG x3000
LD R0, NUM
LD R2, MASK
LOOP: LDR R1, R0, #0
BRz DONE
AND R5, R1, R2
BRz L1
BRnzp NEXT
L1: ADD R1, R1, R1
STR R1, R0, #0
NEXT: ADD R0, R0, #1
BRnzp LOOP
DONE: HALT
NUM: .FILL x4000
MASK: .FILL x8000
.END
```

CIT 593

7/9

NEXT Time

How I do input – output (I/O) ?

I/O is another electronic device that interface with the computer

- By computer we mean
 - Processor: ALU, Control
 - Memory

Require specialized knowledge and protection

- Knowledge of I/O device and how to use them
 - Programmers don't want to know this!
- Protection for shared I/O resources e.g. Hard- Disk
 - Want some sort of isolation

Solution: *service routines* or *system calls*

- Low-level, privileged operations performed by operating system
- Used by almost all applications and hence made a routine (regular course of procedure)
- Example: keyboard polling routine in Chap 8

CIT 593

8/9

Character input/output using Trap Codes

Code	Equivalent	Description
IN	TRAP x23	Print prompt on console, read (and echo) one character from keyboard. Character stored in R0[7:0].
OUT	TRAP x21	Write one character (in R0[7:0]) to console.

Example:

```
ADD R0, R0, #10
```

```
OUT          ;displays value in R0 to the monitor
```

Question: What will be printed to the Monitor given LC-3 uses ASCII codes for characters?

CIT 593

9/9

Things to consider

If TRAP instruction is essentially executing different piece of code

- Is it using the same resources as the user program is doing
- If yes, then what must be done from user side or service providers side
- All this and much more in Chp 9

CIT 593

10/9

Next Time

Reading

- Chapter 7

Lecture

- Chap 8

Homework

- Homework 3 coming up (10/7 or 10/8)

CIT 593

11/9