

Computer Science 210 s1c  
**Computer Systems 1**  
2008 Semester 1  
Lecture Notes

Lecture 14, 7Apr08:  
**The LC-3 ISA; Assembly Language**

*James Goodman*



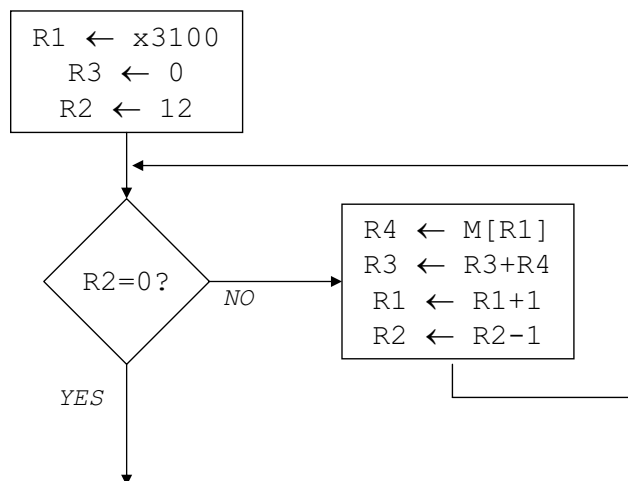
Credits: Slides prepared by Gregory T. Byrd, North Carolina State University

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

## Using Branch Instructions

Compute sum of 12 integers.

Numbers start at location x3100. Program starts at location x3000.



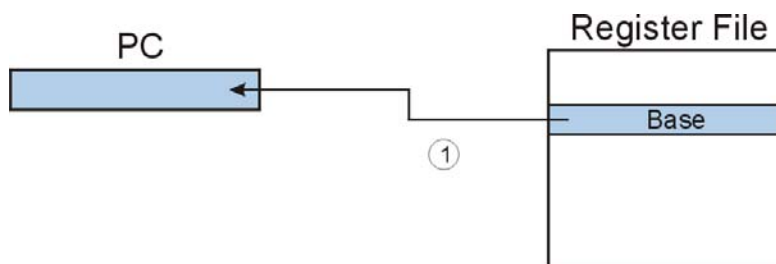
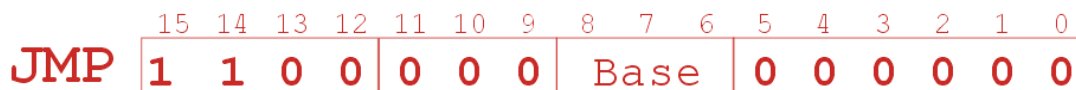
## Sample Program

Address	Instruction																Comments
x3000	1	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1	<b><i>R1 ← x3100 (PC+0xFF)</i></b>
x3001	0	1	0	1	0	1	1	0	1	1	1	0	0	0	0	0	<b><i>R3 ← 0</i></b>
x3002	0	1	0	1	0	1	0	0	1	0	1	0	0	0	0	0	<b><i>R2 ← 0</i></b>
x3003	0	0	0	1	0	1	0	0	1	0	1	0	1	1	0	0	<b><i>R2 ← 12</i></b>
x3004	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	<b><i>If Z, goto x300A (PC+5)</i></b>
x3005	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	<b><i>Load next value to R4</i></b>
x3006	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	1	<b><i>Add to R3</i></b>
x3007	0	0	0	1	0	0	1	0	0	1	1	0	0	0	0	1	<b><i>Increment R1 (pointer)</i></b>
x3008	0	0	0	1	0	1	0	0	1	0	1	1	1	1	1	1	<b><i>Decrement R2 (counter)</i></b>
x3009	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	<b><i>Goto x3004 (PC-6)</i></b>

## JMP (Register)

Jump is an unconditional branch -- *always* taken.

- Target address is the contents of a register.
- Allows any target address.



## TRAP



Calls a **service routine**, identified by 8-bit “trap vector.”

<i>vector</i>	<i>routine</i>
x23	input a character from the keyboard
x21	output a character to the monitor
x25	halt the program

When routine is done,  
PC is set to the instruction following TRAP.  
(We’ll talk about how this works later.)

## Another Example

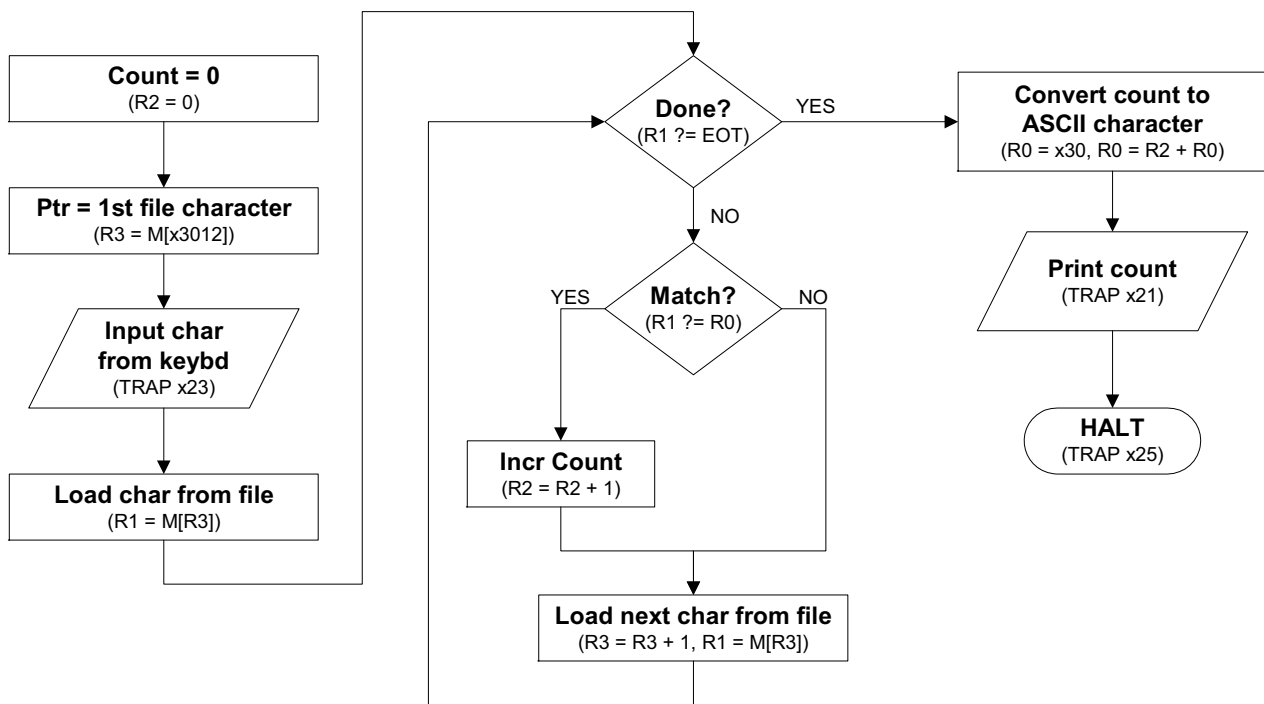
### Count the occurrences of a character in a file

- **Program begins at location x3000**
- **Read character from keyboard**
- **Load each character from a “file”**
  - File is a sequence of memory locations
  - Starting address of file is stored in the memory location immediately after the program
- **If file character equals input character, increment counter**
- **End of file is indicated by a special ASCII value: EOT (x04)**
- **At the end, print the number of characters and halt**  
(assume there will be less than 10 occurrences of the character)

**A special character used to indicate the end of a sequence is often called a **sentinel**.**

- Useful when you don’t know ahead of time how many times to execute a loop.

## Flow Chart



7-Apr-08

CS210

250

## Program (1 of 2)

Address	Instruction	Comments
x3000	0 1 0 1 0 1 0 0 1 0 1 0 0 0 0 0	<b><math>R2 \leftarrow 0</math> (counter)</b>
x3001	0 0 1 0 0 1 1 0 0 0 0 1 0 0 0 0	<b><math>R3 \leftarrow M[x3102]</math> (ptr)</b>
x3002	1 1 1 1 0 0 0 0 0 1 0 0 0 1 1	<b>Input to R0 (TRAP x23)</b>
x3003	0 1 1 0 0 0 1 0 1 1 0 0 0 0 0	<b><math>R1 \leftarrow M[R3]</math></b>
x3004	0 0 0 1 1 0 0 0 0 1 1 1 1 1 0	<b><math>R4 \leftarrow R1 - 4</math> (EOT)</b>
x3005	0 0 0 0 0 1 0 0 0 0 0 1 0 0 0	<b>If Z, goto x300E</b>
x3006	1 0 0 1 0 0 1 0 0 1 1 1 1 1 1	<b><math>R1 \leftarrow \text{NOT } R1</math></b>
x3007	0 0 0 1 0 0 1 1 0 0 0 0 0 1	<b><math>R1 \leftarrow R1 + 1</math></b>
x3008	0 0 0 1 0 0 1 0 0 0 0 0 0 0	<b><math>R1 \leftarrow R1 + R0</math></b>
x3009	0 0 0 0 1 0 1 0 0 0 0 0 0 0 1	<b>If N or P, goto x300B</b>

7-Apr-08

CS210

251

## Program (2 of 2)

Address		Instruction	Comments
x300A	0 0 0 1	0 1 0 0 1 0 1 0 0 0 0 0 1	<b><math>R_2 \leftarrow R_2 + 1</math></b>
x300B	0 0 0 1	0 1 1 0 1 1 1 0 0 0 0 0 1	<b><math>R_3 \leftarrow R_3 + 1</math></b>
x300C	0 1 1 0	0 0 1 0 1 1 0 0 0 0 0 0 0	<b><math>R_1 \leftarrow M[R_3]</math></b>
x300D	0 0 0 0	1 1 1 1 1 1 1 1 0 1 1 0	<b><i>Goto x3004</i></b>
x300E	0 0 1 0	0 0 0 0 0 0 0 0 0 1 0 0	<b><math>R_0 \leftarrow M[x3013]</math></b>
x300F	0 0 0 1	0 0 0 0 0 0 0 0 0 0 1 0	<b><math>R_0 \leftarrow R_0 + R_2</math></b>
x3010	1 1 1 1	0 0 0 0 0 0 1 0 0 0 0 0 1	<b><i>Print R<sub>0</sub> (TRAP x21)</i></b>
x3011	1 1 1 1	0 0 0 0 0 0 1 0 0 1 0 1	<b><i>HALT (TRAP x25)</i></b>
X3012	Starting Address of File		
x3013	0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0		<b><i>ASCII x30 ('o')</i></b>

7-Apr-08

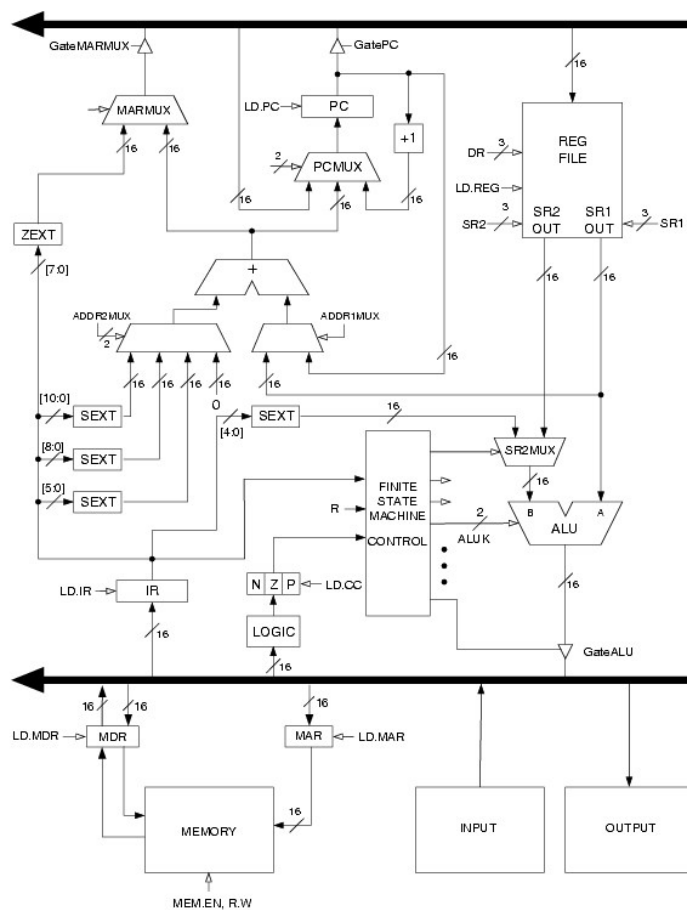
CS210

252

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

## LC-3 Data Path Revisited

**Filled arrow**  
= info to be processed.  
**Unfilled arrow**  
= control signal.



7-Apr-08

253