CS/ECE 252: Introduction to Computer Engineering
Computer Sciences Department
University of Wisconsin, Madison
Midterm III

Professor David Wood

Friday, April 10th 2009

This exam is closed book. There is to be nothing used during the exam. There are 7 pages in this exam not counting this cover sheet. See the back page for the LC-3 instruction set.

Last Name: ______________________

First Name: ______________________

Section: ______________________

Student ID: ______________________

Question	Points	Score
1	10	
2	10	
3	12	
4	20	
5	24	
6	24	
Total:	100	

1. (10 points) Machine Language to Pseudo Code:

   Translate the binary value to pseudo code to and give the value of R3 in binary after execution of the code fragment. The first line is filled in for you.

Binary Value	Pseudo Code
0101 0100 1010 0000	R2 ← R2 AND 0
0001 0100 1000 0010	
1001 0110 1011 1111	
0010 0010 0000 0000	
1100 0000 0100 0000	

R3 =

Solution:

Binary Value	Pseudo Code
0101 0100 1010 0000	R2 ← R2 AND 0
0001 0100 1000 0010	R2 ← R2 AND R2
1001 0110 1011 1111	R3 ← NOT R2
0010 0010 0000 0000	R1 ← MEM[PC + 0]
1100 0000 0100 0000	PC ← R1

R3 = 1111 1111 1111 1111

2. (10 points) Pseudo Code to Machine Language:

   Translate the pseudo code to binary machine language and give the value of R4 in binary after execution of the code fragment. The first line is filled in for you.

Binary Value	Pseudo Code
0101011011100000	R3 ← R3 AND 0
	R4 ← R3 + 11
	R5 ← PC + -1
	MEM[PC + 5] ← R4
	R4 ← R4 + R3

R4 =

Solution:

Binary Value	Pseudo Code
0101011011100000	R3 $\leftarrow$ R3 AND 0
0001100011101011	R4 $\leftarrow$ R3 + 11
1110101111111111	R5 $\leftarrow$ PC + -1
0011100000000101	MEM[PC + 5] $\leftarrow$ R4
0001100011000100	R4 $\leftarrow$ R4 + R3

R4 = 0000 0000 0000 1011

3. (12 points) Note that there is no OR instruction in the LC-3 ISA. Complete the code so that the following 4 instruction sequence stores the result of R1 OR R2 in the register R3.

   (1):
   (2): 1001 1000 0111 1111
   (3): 0101 1101 0000 0101
   (4):

   Solution:

   (1): 1001 1010 1011 1111; R5 $\leftarrow$ NOT R2
   (2): 1001 1000 0111 1111; R4 $\leftarrow$ NOT R1
   (3): 0101 1101 0000 0101; R6 $\leftarrow$ R4 AND R5
   (4): 1001 0111 1011 1111; R3 $\leftarrow$ NOT R6

4. (20 points) Addressing:

Let R0,R1,R2, and R3 be initialized to 0. The PC initially has value x3000. What are the values of R0,R1,R2, and R3 when we terminate.

Address	Value	Translation
x3000	x2009	R0 $\leftarrow$ MEM[PC + 9]
x3001	xA20B	R1 $\leftarrow$ MEM[MEM[PC + 11]]
x3002	xC000	PC $\leftarrow$ R0
x3003	x6448	R2 $\leftarrow$ MEM[R1 + 8]
x3004	x6647	R3 $\leftarrow$ MEM[R1 + 7]
x3005	xF025	HALT
x3006	x3000	MEM[PC + 0] $\leftarrow$ R0
x3007	x3001	MEM[PC + 1] $\leftarrow$ R0
x3008	x3002	MEM[PC + 2] $\leftarrow$ R0
x3009	x3003	MEM[PC + 3] $\leftarrow$ R0
x300A	x3004	MEM[PC + 4] $\leftarrow$ R0
x300B	x3005	MEM[PC + 5] $\leftarrow$ R0
x300C	x3006	MEM[PC + 6] $\leftarrow$ R0
x300D	x3007	MEM[PC + 7] $\leftarrow$ R0
x300E	x3008	MEM[PC + 8] $\leftarrow$ R0
x300F	x3009	MEM[PC + 9] $\leftarrow$ R0

Solution: x3000 LD R0, #9; R0 $\leftarrow$ MEM[x3000 + 1 + 9] = x3004
x3001 LDI R1, #11; R1 $\leftarrow$ MEM[MEM[x3001 + 1 + B]] = MEM[x3007] = x3001
x3002 JMP R0
x3003 LDR R2, R1, #8; R2 $\leftarrow$ MEM[R1 + 8] = x3003 (never run)
x3004 LDR R3, R1, #7; R3 $\leftarrow$ MEM[R1 + 7] = x3002
x3005 HALT
x3006 x3000
x3007 x3001
x3008 x3002
x3009 x3003
x300A x3004
x300B x3005
x300C x3006
x300D x3007
x300E x3008
x300F x3009

Therefore R0=x3004, R1=x3000, R2 = 0, R3 = x3003

5. (24 points) Iteration:

Everytime a register is written write the new value update the table. Updating the table consists of finding the registers row, and writing in hex the value that is written. The first 3 entries to the table have been written for you corresponding the the first 3 lines of the program. Complete the partially filled table to match the execution of the program.

Address	Value	Translation
x3000	0101 0110 1110 0000	R3 ← R3 AND 0
x3001	0001 0110 1110 0010	R3 ← R3 + 2
x3002	0101 1011 0110 0000	R5 ← R5 AND 0
x3003	0101 1001 0010 0000	R4 ← R4 AND 0
x3004	0001 1001 0010 0010	R4 ← R4 + 2
x3005	0001 1011 0110 0011	R5 ← R5 + 3
x3006	0001 1011 0100 0011	R5 ← R5 + R3
x3007	0001 1001 0011 1111	R4 ← R4 + -1
x3008	0000 0011 1111 1101	BRp -3
x3009	0001 0110 1111 1111	R3 ← R3 + -1
x300A	0000 0011 1111 1000	BRp -8
x300B	0001 1101 0110 0101	R6 ← R5 + 5
x300C	1111 0000 0010 0101	HALT

Register	1st Val	2nd Val	3rd Val	4th Val	5th Val	6th Val	7th Val	8th Val
R0								
R1								
R2								
R3	0000	0002						
R4								
R5	0000							
R6								
R7								

Solution:

Register	1st Val	2nd Val	3rd Val	4th Val	5th Val	6th Val	7th Val	8th Val
R0								
R1								
R2								
R3	0	2	1	0				
R4	0	2	1	0	0	2	1	0
R5	0	3	5	7	A	B	C	
R6	17							
R7								

6. (24 points) Debugging:

Recall that in homework 6 we wrote a program to compare 2 numbers. The following program does something similar but instead of comparing 2 numbers we compare 2 strings. The two strings are stored at memory locations x4000 and x5000 and are null terminated, that is end with x0000. You may assume that the strings are of the same length. The program was intended to have R1 be 0 if the two strings are equal and 1 otherwise. However, the program has 4 errors and does not behave as expected. Identify and correct the errors in the code, give the address for each error and the correction in Hex, and the pseudo code.

Memory:

Address	Hex Value	Translation
x3000	x5260	R1 ← R1 AND 0
x3001	x240C	R2 ← MEM[PC + 12]
x3002	x260C	R3 ← MEM[PC + 12]
x3003	x6880	R4 ← MEM[R2 + 0]
x3004	x6A80	R5 ← MEM[R2 + 0]
x3005	x0406	BRz 6
x3006	x14A1	R2 ← R2 + 1
x3007	x16E1	R3 ← R3 + 1
x3008	x993F	R4 ← NOT R4
x3009	x1922	R4 ← R4 + 2
x300A	x1905	R4 ← R4 + R5
x300B	x0BF7	BRnp -9
x300C	x1261	R1 ← R1 + 1
x300D	xF025	HALT
x300E	x4000	JSRR R0
x300F	x5000	R0 ← R0 AND R0

Solution:

Address	Hex Value	Translation
x3000	x5260	R1 ← R1 AND 0
x3001	x240C	R2 ← MEM[PC + 12]
x3002	x260C	R3 ← MEM[PC + 12]
x3003	x6880	R4 ← MEM[R2 + 0]
x3004	x6AC0	R5 ← MEM[R3 + 0]
x3005	x0407	BRz 7
x3006	x14A1	R2 ← R2 + 1
x3007	x16E1	R3 ← R3 + 1
x3008	x993F	R4 ← NOT R4
x3009	x1921	R4 ← R4 + 1
x300A	x1905	R4 ← R4 + R5
x300B	x05F7	BRz -9
x300C	x1261	R1 ← R1 + 1
x300D	xF025	HALT
x300E	x4000	JSRR R0
x300F	x5000	R0 ← R0 AND R0

Figure 1: Instruction Set from ItCS 2nd edition

A.3 The Instruction Set 525

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

ADD+	0001	DR	SR1	0	00	SR2
ADD+	0001	DR	SR1	1	imm5	
AND+	0101	DR	SR1	0	00	SR2
AND+	0101	DR	SR1	1	imm5	
BR	0000	n	z	p	PCoffset9	
JMP	1100	000	BaseR	000000		
JSR	0100	1	PCoffset11			
JSRR	0100	0	00	BaseR	000000	
LD+	0010	DR	PCoffset9			
LDI+	1010	DR	PCoffset9			
LDR+	0110	DR	BaseR	offset6		
LEA+	1110	DR	PCoffset9			
NOT+	1001	DR	SR	111111		
RET	1100	000	111	000000		
RTI	1000	000000000000				
ST	0011	SR	PCoffset9			
STI	1011	SR	PCoffset9			
STR	0111	SR	BaseR	offset6		
TRAP	1111	0000	trapvect8			
reserved	1101					

Figure A.2 Format of the entire LC-3 instruction set. **Note:** + indicates instructions that modify condition codes