CS/ECE 252: INTRODUCTION TO COMPUTER ENGINEERING
COMPUTER SCIENCES DEPARTMENT
UNIVERSITY OF WISCONSIN-MADISON

Prof. David A. Wood
TAs Spyros Blanas, Priyananda Shenoy, Shengnan Wang

Midterm Examination 4
In Class (50 minutes)
Friday, May 9, 2008
Weight: 15%

CLOSED BOOK, NOTE, CALCULATOR, PHONE, & COMPUTER.

The exam in two-sided and has **TWELVE** pages, including two blank pages and a copy of the ***Standard ASCII Table***, some ***Trap Service Routines*** description and the ***LC-3 Instruction Set handout*** on the final page (please feel free to detach this final page, but insert it into your exam when you turn it in).

You are **required** to present a valid UW-Madison student ID card or other government-issued photo ID to one of the teaching assistants who are proctoring this exam before leaving the room. **If you fail to do so, we cannot grade your exam.**

Plan your time carefully, since some problems are longer than others.

NAME: _____KEY_____

SECTION:_____

ID# _____

"Green"

Problem Number	Maximum Points	Graded By
1	12	SW
2	8	SB
3	20	SB
4	25	PS
5	26	SW
6	9	PS
Total	100	

Problem 1 (12 points): Short Answers

a. The LC-3 assembly process is done in two complete passes through the entire assembly language program. What is the objective of the second pass?

Generates machine code for each instruction

b. What single instruction is equivalent to the following two LC-3 instructions?

```
LD      R0, FooBar
LDR     R0, R0, #0
```

LDI R0, FooBar

c. What single instruction is equivalent to the following one LC-3 instruction?

```
RET
```

JMP R7

d. What is the purpose of .BLKW pseudo-op?

Allocates a block of memory

Problem 2 (8 points): Memory-Mapped I/O

a) An LC-3 instruction loads from the address xFE02. How does the LC-3 know whether to load from KBDR or from memory location xFE02?

All addresses in the range xFE00-xFFFF are reserved for I/O. The Address Control Logic knows that the location xFE02 maps to the KBDR.

b) How are the bits in the KBSR defined?

KBSR[15] = is there a new character pressed.

KBSR[14-1] = 0

Problem 3 (20 points): Two-Pass Assembly Process

An assembly language LC-3 program is given below:

```
1        .ORIG x3000
2
3   MAIN
4        LEA R0, MSG
5        PUTS
6        JSR RL
7        HALT
8
9   RL
10       ST R7, RL_RETURN
11       LD R3, ENTER         ; initialize R3 to 'enter char'
12       AND R1, R1, #0
13       ADD R1, R1, BUFFER   ; initialize R1 to point to the
14                            ; start of buffer
15
16       LD  R0, PROMPT
17       OUT                  ; show prompt
18
19   RL_START
20       GETC
21       OUT                  ; read input and echo it back
22
23       NOT R4, R3
24       ADD R4, R4, #1
25       ADD R4, R0, R4
26       BRZ RL_END           ; leave if user hits enter
27
28       STR R0, R1, #0
29       ADD R1, R1, #1
30       BR  RL_START         ; write char, increment pointer,
31                            ; read next char
32
33   RL_END
34       RET
35
36   BUFFER          .BLKW x000F
37   RL_RETURN       .FILL x0000
38   PROMPT          .FILL x003E ; '>' character
39   ENTER           .FILL x000A ; 'enter' character
40   MSG             .STRINGZ “Enter input:”
41
42   .END
```

a. Fill in the symbol table for the program:

Symbol	Address
MAIN	x3000
RL	**x3004**
RL_START	**x300A**
RL_END	**x3013**
BUFFER	**x3014**
RL_RETURN	**x3023**
PROMPT	**x3024**
ENTER	**x3025**
MSG	**x3026**

b. Assuming that both passes of the assembler were to execute, write the binary word (machine language instruction) that would be generated by the assembler for the instruction at line 11 of the program.

0010 000 0 0001 1111 = x201F

c. The programmer intended that the RL subroutine reads user input, writes it in BUFFER and returns when user types enter. There are two errors in this subroutine. For each, describe the error and indicate whether it will be detected at assembly time or at run time.

Assembly time error: ADD R1, R1, BUFFER is not valid. It should be LEA R1, BUFFER

Runtime Error: The trap GETC overwrites R7 so subroutine RL doesn't return properly.

Problems 4,5,6 make use of the following program

```
      .ORIG x3000
0                 ST  R0, SAVER0
1                 ST  R1, SAVER1
2                 JSR SUBROUTINE1
3                 LD  R0, SAVER0
4                 LD  R1, SAVER1
5                 HALT

6     SUBROUTINE1 ST  R7, SAVER7
7                 ST  R2, SAVER2
8                 ST  R3, SAVER3
9     CHECKPOINT1 LEA R4, BUFFER
10                LD  R3, DELIM
11                NOT R3, R3
12                ADD R3, R3, #1
13    LOOP_START  JSR SUBROUTINE2
14                ADD R2, R0, R3
15                BRz LOOP_END
16                STR R0, R4, #0
17                ADD R4, R4, #1
18                BR  LOOP_START
19    LOOP_END    AND R0, R0, #0
20                STR R0, R4, #0
21                LD  R2, SAVER2
22                LD  R7, SAVER7
23                LD  R3, SAVER3
24                RET

25    SUBROUTINE2 LDI R1, KBSR
26                BRzp SUBROUTINE2
27                LDI R0, KBDR
28    CHECKPOINT2 RET

29    SAVER1      .FILL x0000
30    SAVER2      .FILL x0000
31    SAVER7      .FILL x0000
32    SAVER0      .FILL x0000
33    SAVER3      .FILL x0000
34    DELIM       .FILL x003B
35    KBSR        .FILL xFE00
36    KBDR        .FILL xFE02
37    BUFFER      .BLKW x0030
      .END
```

Problem 4 (20 points): Traps and Subroutines

a) In the program in page 6, what registers are callee-saved, and what registers are caller-saved?

Caller Saved: R0,R1
Callee Saved: R7,R2,R3

b) Is there a register which cannot be callee-saved? If yes, why not?

R7. There is no point in saving R7 in the callee, since R7 gets overwritten by the JSR instruction.

c)What will be the value in R7:

1. If you put a breakpoint at Checkpoint1?

   x3003

2. If you put a breakpoint at Checkpoint2?

   x300E

d) Can interrupts use R7 to hold the return address? If no, why not?

No. Interrupts can occur at any time, so the programmer cannot save-restore values as could be done in the case of subroutines.

Problem 5 (26 points): Input/Output

a) In the program in page 6, what does the subroutine SUBROUTINE2 do?

Polls the keyboard until it gets a character

b) When does the loop in SUBROUTINE1 terminate?

When the key pressed is ';'

c) What does the subroutine SUBROUTINE1 do?

Reads characters from keyboard and copies it into a buffer, terminates when a ';' is pressed

d) What does this program do?

Reads characters from keyboard and copies it into a buffer

e) Assume that the label BUFFER points to address x3037. If the user types the following sequence:

A B C ; K M \ +

What would be the contents of the following memory locations

Address	ASCII value
x3037	'A'
x3038	'B'
x3039	'C'
x303A	0

Problem 6 (9 points): Input/Output

a) What is the purpose of the Keyboard Status Register?

The keyboard status registers maintains a flag indicating "has the character in KBDR been read?". If it's 0, that means the character has already been read, if it's 1 it means the character is new and has not been read.

b) What problem could occur if the keyboard hardware doesn't check the KBSR before writing to the KBDR?

The previously typed value in KBDR will be lost.

c) Circle the correct combination that describes the program on page 6.

1. Special Opcode for I/O and interrupt driven
2. Special Opcode for I/O and polling
3. Memory mapped and interrupt driven
4. **Memory mapped and polling**

Scratch Sheet 1 (in case you need additional space for some of your answers)

ASCII Table

Character	*Hex*	*Character*	*Hex*	*Character*	*Hex*	*Character*	*Hex*
nul	00	sp	20	@	40	`	60
soh	01	!	21	A	41	a	61
stx	02	"	22	B	42	b	62
etx	03	#	23	C	43	c	63
eot	04	$	24	D	44	d	64
enq	05	%	25	E	45	e	65
ack	06	&	26	F	46	f	66
bel	07	'	27	G	47	g	67
bs	08	(	28	H	48	h	68
ht	09	)	29	I	49	i	69
lf	0A	*	2A	J	4A	j	6A
vt	0B	+	2B	K	4B	k	6B
ff	0C	,	2C	L	4C	l	6C
cr	0D	-	2D	M	4D	m	6D
so	0E	.	2E	N	4E	n	6E
si	0F	/	2F	O	4F	o	6F
dle	10	0	30	P	50	p	70
dc1	11	1	31	Q	51	q	71
dc2	12	2	32	R	52	r	72
dc3	13	3	33	S	53	s	73
dc4	14	4	34	T	54	t	74
nak	15	5	35	U	55	u	75
syn	16	6	36	V	56	v	76
etb	17	7	37	W	57	w	77
can	18	8	38	X	58	x	78
em	19	9	39	Y	59	y	79
sub	1A	:	3A	Z	5A	z	7A
esc	1B	;	3B	[	5B	{	7B
fs	1C	<	3C	\	5C	|	7C
gs	1D	=	3D	]	5D	}	7D
rs	1E	>	3E	^	5E	~	7E
us	1F	?	3F	_	5F	del	7F

Trap Service Routines

Trap Vector	Assembler Name	Description
x20	GETC	Read a single character from the keyboard. The Character is not echoed onto the console. Its ASCII code is copied into R0. The high eight bits of R0 are cleared.
x21	OUT	Write a character in R0[7:0] to the console display.
...	...	...
x25	HALT	Halt execution and print a message on the console.

LC-3 Instruction Set (Entered by Mark D. Hill on 03/14/2007; last update 03/15/2007)

PC': incremented PC. setcc(): set condition codes N, Z, and P. mem[A]:memory contents at address A.
SEXT(immediate): sign-extend immediate to 16 bits. ZEXT(immediate): zero-extend immediate to 16 bits.
Page 2 has an ASCII character table.