CS/ECE 252: INTRODUCTION TO COMPUTER ENGINEERING

UNIVERSITY OF WISCONSIN—MADISON

Prof. Mikko Lipasti & Prof. Gurinder S. Sohi
TAs: Daniel Chang, Felix Loh, Philip Garcia, Sean Franey, Vignyan Kothinti Naresh, Raghu Raman and Newsha Ardalani

Midterm Examination 4
In Class (50 minutes)
Friday, December 17, 2010
Weight: 12.5%

NO: BOOK(S), NOTE(S), CALCULATORS OF ANY SORT.

This exam has 9 pages, including one page for the LC3 Instruction Set and two blank pages at the end. Plan your time carefully, since some problems are longer than others. You must turn in pages 1 to 7.

LAST NAME: _____

FIRST NAME:_____

SECTION: _____

ID# _____

Question	Maximum Point	Points
1	3	
2	5	
3	4	
4	6	
5	7	
Total	25	

1. Assembly Errors (3 Points)

Consider the following assembly code.

```
          .ORIG x3000

.MAIN
          LD R0, ASCII
          LD R1, NEG
LOOP      TRAP x22
          BRzp LOOP
          TRAP x23
          ADD R0, R0, MINUSONE
          ADD R3, R0, R1
          BRp LOOP
HALT      HALT
ASCII     .FILL     x0047
NEG       .FILL     xFFBD
MINUSONE  .FILL     #-1
          .END
```

Briefly explain three assembly errors in the above program (3 points)

2. Two-Pass Assembly Process (5 points)

An assembly language LC-3 program is given below :

```
        .ORIG x3000
        AND R2, R2, #0
        AND R3, R3, #0
        LD R0, M0
        LD R1, M1
LOOP BRz DONE
        ADD R3, R3, #1
        ADD R2, R2, R0
        ADD R1, R1, #-1
        BR LOOP
DONE    ST R2, RESULT
        HALT
RESULT  .FILL x0000
M0      .FILL x0006
M1      .FILL x0011
        .END
```

(a). Fill in the symbol table created by the assembler on the first pass of the above program. (3 points)

Symbol Name	Address

(b) Once the symbol table is created, the assembler then creates a binary version (.obj) of the program as listed below (with 2 missing lines). (2 points)

```
0101 0100 1010 0000 ;AND R2, R2, #0
0101 0110 1110 0000 ;AND R3, R3, #0
0010 0000 0000 1001 ;LD R0, M0
0010 0010 0000 1001 ;LD R1, M1
0000 0100 0000 0100 ;BRz DONE
0001 0110 1110 0001 ;ADD R3, R3, #1
0001 0100 1000 0000 ;ADD R2, R2, R0
____ ____ ____ ____ ;ADD R1, R1, #-1
0000 1111 1111 1011 ;BR LOOP
0011 0100 0000 0001 ;ST R2, RESULT
____ ____ ____ ____ ;HALT (TRAP x25)
0000 0000 0000 0000 ;.FILL x0000
0000 0000 0000 0110 ;.FILL x0006
0000 0000 0001 0001 ;.FILL x0011
```

3. I/O in LC-3 (4 Points)

An LC-3 program is provided below:

```
            .ORIG x3000
            LD    R0, ASCII
            LD   R1, NEG
AGAIN       LDI  R2, DSR
            BRzp AGAIN
            STI   R0, DDR
            ADD  R0, R0, #1
            ADD  R3, R0, R1
            BRn  AGAIN
            HALT

ASCII       .FILL x0041
NEG         .FILL xFFB6
DSR         .FILL xFE04              ; Address of DSR
DDR         .FILL xFE06              ; Address of DDR
            .END
```

a) What does this program do? (3 points)

b) What is the purpose of the Display Status Register (DSR)? (1 points)

4. Subroutines (6 Points)

```
 1   ;CODE TO INPUT AND PRINT 6 CHARACTERS
 2
 3          .ORIG x3000
 4          AND R0, R0, #0              ; Initialise R0, our counter
 5 LOOP
 6          LEA R1, INPSTRING           ; R1 now has base of INPSTRING
 7          ADD R1, R1, R0              ; R1 now has base + offset = R0
 8          ST R0, SAVEREG1             ; SAVE R0
 9          JSR ONECHAR                 ; Call Subroutine
10          LD __, SAVEREG1             ; Restore ??
11          ADD R0, R0, #1              ; Increment R0
12          LD R1, LENGTH               ; Load R1 with minus length
13          ADD R1, R1, R0              ;
14          BRn LOOP                    ; loop till 6 characters are reached
15          LEA R0, INPSTRING           ; Get ready to print
16          PUTS                        ; TRAP X22 And print
17          HALT                        ; We're done
18
19 ONECHAR
20          ST __, SAVEREG2             ; SAVE ??
21          GETC                        ; TRAP X20 Get a character from
                                        ; Keyboard input.
22          LD __, SAVEREG2             ; Restore ??
23          STR R0, R1, #0              ; Save keyboard inp(R0 contains input)
24          RET
25
26 LENGTH  .FILL xFFFA                  ; minus Length (-6)
27 KBSR    .FILL xFE00
28 KBDR    .FILL xFE02
29 SAVEREG1  .FILL x0
30 SAVEREG2  .FILL x0
31 INPSTRING .BLKW 6
32         .END
```

In the code above the Subroutine ONECHAR takes 1 character from the user (keyboard) and saves it into the memory. The assembly code uses ONECHAR in a loop 6 times to input 6 characters and saves it to the memory. Finally it prints the string to the screen.

(a) Line 8 saves R0 before calling the subroutine ONECHAR. Briefly explain why this is necessary. (2 points)

(b) What other register needs to be stored and restored inside the subroutine [Fill in lines 20, 22].

(c) Once the subroutine is done, we will have to restore the registers. Fill in the register restored in line 10 (1 point)

5. General Questions (7 points)

Circle the best answer.

1. A new service routine is defined starting in memory location x3700. After loading a program that calls this subroutine, the user sets memory location x0066 to x3700. Which of the following can be used to call this subroutine?

a. TRAP x66
b. TRAP x67
c. TRAP x3700
d. TRAP x0037

2. JSRR R3 is equivalent to

a. LEA R7, #1
   JMP R3, #0
b. LEA R3, #1
   JMP R7, #0
c. LEA R3, #1
   JMP R3, #0
d. All of the above are equivalent

3. Which of the following pseudo-op tells the assembler where the program ends

a. END
b. .HALT
c. HALT
d. .END

4. Assembling the instruction ADD R1, R1, #55 causes which of the following errors

a. R1 is not initialised
b. ADD instruction takes only 3 register sources (2 sources + 1 destination)
c. Immediate value (55) is out of range
d. The instruction does not cause an error.

5. How many memory locations are used by the following assembly directive :
PALINDROME .STRINGZ "malayalam"

a. 9
b. 8
c. 10
d. 11

6. As discussed in lecture, when faced with a difficult decision in the workplace, it is most useful to separate the issues at hand into the following categories:

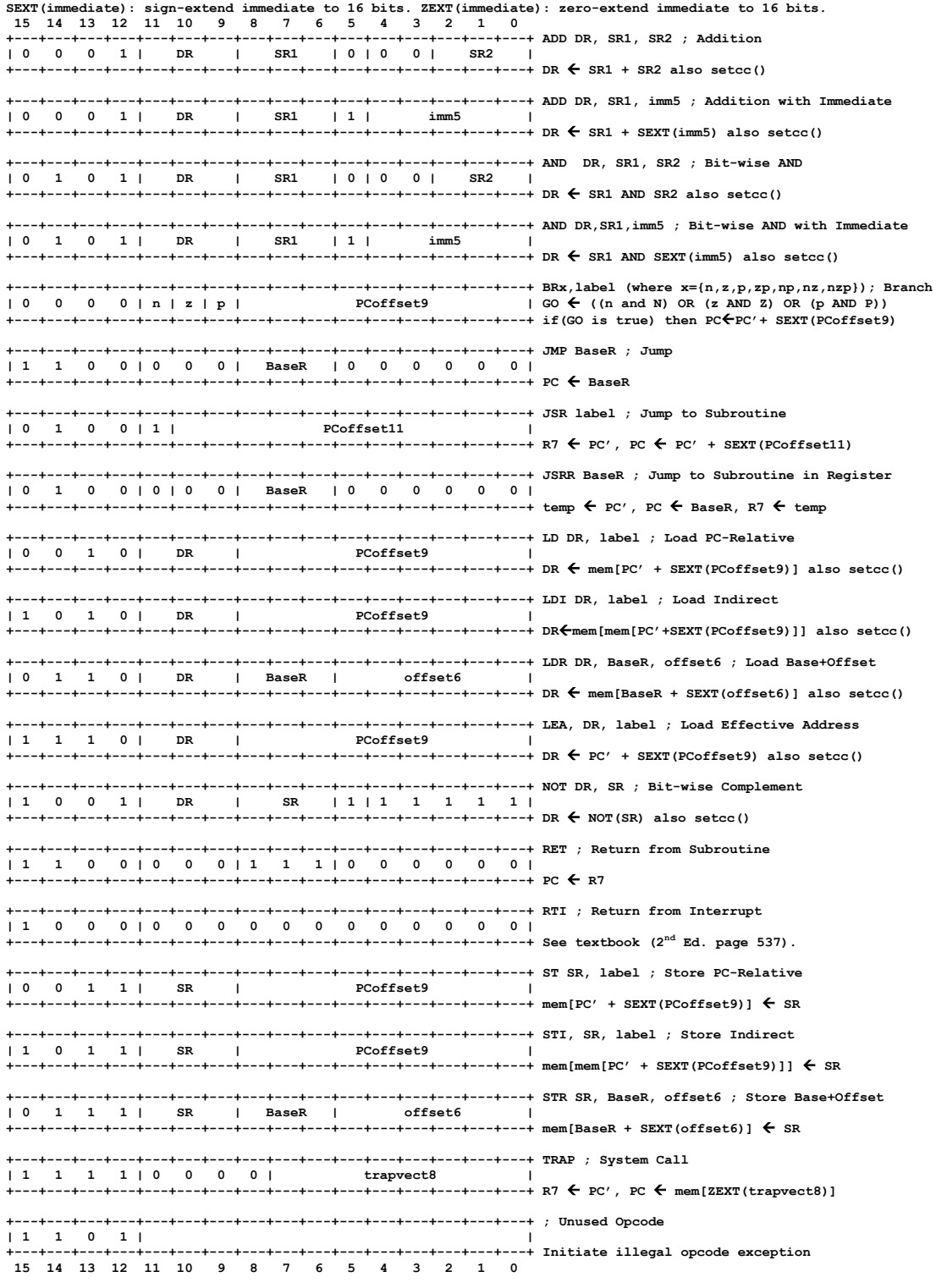
a. legal, moral, and algorithmic
b. immediate, mid-term, and long-term
c. executive, judicial, and legislative
d. factual, conceptual, and ethical

7. Which of the following combinations best describes the way input/output service routines work in the LC-3 processor

a. Special opcode for I/O and interrupt driven
b. Special opcode for I/O and polling
c. Memory mapped I/O and polling
d. All of the above

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.



ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	*[NULL]*	32	20	*[SPACE]*	64	40	@	96	60	`
1	1	*[START OF HEADING]*	33	21	!	65	41	**A**	97	61	**a**
2	2	*[START OF TEXT]*	34	22	"	66	42	**B**	98	62	**b**
3	3	*[END OF TEXT]*	35	23	#	67	43	**C**	99	63	**c**
4	4	*[END OF TRANSMISSION]*	36	24	$	68	44	**D**	100	64	**d**
5	5	*[ENQUIRY]*	37	25	%	69	45	**E**	101	65	**e**
6	6	*[ACKNOWLEDGE]*	38	26	&	70	46	**F**	102	66	**f**
7	7	*[BELL]*	39	27	'	71	47	**G**	103	67	**g**
8	8	*[BACKSPACE]*	40	28	(	72	48	**H**	104	68	**h**
9	9	*[HORIZONTAL TAB]*	41	29	)	73	49	**I**	105	69	**i**
10	A	*[LINE FEED]*	42	2A	*	74	4A	**J**	106	6A	**j**
11	B	*[VERTICAL TAB]*	43	2B	+	75	4B	**K**	107	6B	**k**
12	C	*[FORM FEED]*	44	2C	,	76	4C	**L**	108	6C	**l**
13	D	*[CARRIAGE RETURN]*	45	2D	-	77	4D	**M**	109	6D	**m**
14	E	*[SHIFT OUT]*	46	2E	.	78	4E	**N**	110	6E	**n**
15	F	*[SHIFT IN]*	47	2F	/	79	4F	**O**	111	6F	**o**
16	10	*[DATA LINK ESCAPE]*	48	30	**0**	80	50	**P**	112	70	**p**
17	11	*[DEVICE CONTROL 1]*	49	31	**1**	81	51	**Q**	113	71	**q**
18	12	*[DEVICE CONTROL 2]*	50	32	**2**	82	52	**R**	114	72	**r**
19	13	*[DEVICE CONTROL 3]*	51	33	**3**	83	53	**S**	115	73	**s**
20	14	*[DEVICE CONTROL 4]*	52	34	**4**	84	54	**T**	116	74	**t**
21	15	*[NEGATIVE ACKNOWLEDGE]*	53	35	**5**	85	55	**U**	117	75	**u**
22	16	*[SYNCHRONOUS IDLE]*	54	36	**6**	86	56	**V**	118	76	**v**
23	17	*[ENG OF TRANS. BLOCK]*	55	37	**7**	87	57	**W**	119	77	**w**
24	18	*[CANCEL]*	56	38	**8**	88	58	**X**	120	78	**x**
25	19	*[END OF MEDIUM]*	57	39	**9**	89	59	**Y**	121	79	**y**
26	1A	*[SUBSTITUTE]*	58	3A	:	90	5A	**Z**	122	7A	**z**
27	1B	*[ESCAPE]*	59	3B	;	91	5B	[	123	7B	{
28	1C	*[FILE SEPARATOR]*	60	3C	<	92	5C	\	124	7C	|
29	1D	*[GROUP SEPARATOR]*	61	3D	=	93	5D	]	125	7D	}
30	1E	*[RECORD SEPARATOR]*	62	3E	>	94	5E	^	126	7E	~
31	1F	*[UNIT SEPARATOR]*	63	3F	?	95	5F	_	127	7F	*[DEL]*