CS/ECE 252: INTRODUCTION TO COMPUTER ENGINEERING

UNIVERSITY OF WISCONSIN—MADISON

Prof. Gurindar Sohi

TAs: Pradip Vallathol and Junaid Khalid

Examination 4

In Class (50 minutes)

Wednesday, December 12, 2012

Weight: 17.5%

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

The exam has nine pages. **Circle your final answers**. Plan your time carefully since some problems are longer than others. You **must turn in the pages 1-7**.

LAST NAME: _____

FIRST NAME: _____

ID# _____

Problem	Maximum Points	Points Earned
1	6	
2	6	
3	6	
4	3	
5	3	
6	6	
Total	30	

Problem 1: Assembly Language

(a) Briefly explain the four assembly errors in the following LC-3 program. **(4 Points)**

```
          .ORIG x3000
          LD  R2, INPUT
          AND R0, R0, #0
          ADD R1, R0, #1
          BR NEXT

LOOP      AND R4, R2, R1
          BRz SKIP
          OR  R0, R0, #1

SKIP      ADD R1, R1, R1
          ADD R3, R3, x2F
          LD  R6, SKIP
          NOT R6, R6
          BRzp LOOP

INPUT     .FILL x1997
SKIP      .FILL x1998
  .END
```

i. Label NEXT is not declared.

ii. Duplicate label SKIP

iii. x2F cannot be represented as a signed number in 5 bits

iv. OR is an undefined instruction

(b) Which of the following (if any) of the following pseudo-ops can be used multiple times in a single assembly file. Circle all options that apply. **(2 Points)**

i. .ORIG

ii. .FILL

iii. .BLKW

iv. .STRINGZ

v. .END

Problem 2: Two pass Assembly Process

An LC-3 assembly language program in given below:

```
          .ORIG x3000
          AND R3, R3, #0
          LD  R0, M0
          LD  R1, M1
          LD  R2, M2

LOOP      ADD R3, R3, #1
          ADD R3, R3, R2
          ADD R0, R0, #-1
          BRn LOOP

DONE      ST  R3, RESULT
          HALT
RESULT    .FILL x0000
M0        .BLKW #5
M1        .STRINGZ "CS-ECE-252"
M2        .FILL x0009
          .END
```

(a) A symbol table is created during the first pass by the assembler. Fill in the following symbol table for the above program: **(4 Points)**

Symbol	Address
LOOP	x3004
DONE	x3008
RESULT	x300A
M0	x300B
M1	x3010
M2	x301B

(b) The assembly program is converted into a binary file during the second pass by the assembler. Fill in the binary instructions at the following memory locations: **(2 Points)**

Address	Instructions
x3001	0010 0000 0000 1001
x3007	0000 1001 1111 1100

Problem 3: Traps and Subroutines **(6 Points)**

The following LC-3 assembly program takes a single character as input from the user. If the input character is a digit (0-9), it prints the message "Is a digit" on the display. This process is continued until the user enters the termination character '#', and the program halts. Fill in the missing parts of the program indicated by _____.

```
            .ORIG x3000

GETINPUT    TRAP x20          ; Input a character from the user
                              ; (Do not echo it on the display)

            LD  R1, TERMCHAR  ; termination check
            ADD R1, R0, R1
            BRz END           ; Branch to END on '#'

            JSR CHECKINPUT    ; Call CHECKINPUT subroutine
            BR  GETINPUT

END         HALT

CHECKINPUT
            ST R7, SAVELOC    ; Save something here
            LD  R2, DIGIT0
            ADD R2, R0, R2
            BRn RELOAD

            LD  R2, DIGIT9
            ADD R2, R0, R2
            BRp RELOAD

DISP_IS     LEA R0, STR_IS    ; print a string
            TRAP x22          ; to the display

RELOAD      LD R7, SAVELOC    ; Load something here

            RET

; Data
SAVELOC     .BLKW       #1
STR_IS      .STRINGZ    "Is a digit\n"
STR_NOT     .STRINGZ    "Not a Digit\n"
TERMCHAR    .FILL       0xFFDD      ; negative ASCII value of '#'
DIGIT0      .FILL       0xFFD0      ; negative ASCII value of '0'
DIGIT9      .FILL       0xFFC7      ; negative ASCII value of '9'

            .END
```

Problem 4: I/O

a) Briefly explain the difference between *interrupt-driven I/O* and *polling based I/O*? **(2 Points)**

Polling: CPU keeps checking status register until new data arrives or device ready for new data.

Interrupt: Device sends a special signal to CPU when new data arrives or device ready for next data.

b) What is the main reason to prefer asynchronous I/O over synchronous I/O in recent microprocessor designs? **(1 Point)**

I/O devices usually operate at speeds very different from that of a microprocessor. The rate at which data is provided or consumed is not predictable and usually not in lockstep with the processor clock.

Problem 5: Trap Handling **(3 Points)**

List the main steps of the TRAP mechanism involved in executing the instruction **`TRAP x67`**.

a. Lookup the starting address of the service routine to execute in the Trap Vector table at location 0x67.

b. Transfer control to service routine (Set PC to contents of the memory location 0x67). Save return address in R7.

c. Return from service routine (JMP R7).

Problem 6: Short Answer Questions

Answer the flowing questions briefly.

a) What important feature does the instruction **JSRR** provide that **JSR** does not? **(1 Point)**

JSRR uses the contents a register as the address to jump to (16 bits), while JSR instruction provides an 11 bit offset to PC. Thus the range of addresses to which a JSRR instruction can jump to is larger than that of the JSR instruction.

b) Explain briefly the problem that the *callee-save* and the *caller-save* approaches are trying to solve. **(2 Point)**

If a register value is "destroyed" by actions of a subroutine or service routine, the value has to be saved before it is modified, and reloaded before it is used again.

c) How many trap service routines can be defined in LC-3? **(1 Point)**

256

d) What is the use of *Comments* in an assembly program? **(1 Point)**

Comments are useful to humans to document or understand programs. They are ignored by the assembler.

e) What happens during the **linking** phase of an assembly program? **(1 Point)**

Linking is the process of resolving symbols between independent object files. The linker will search symbol tables of other modules to resolve symbols and complete code generation before loading.

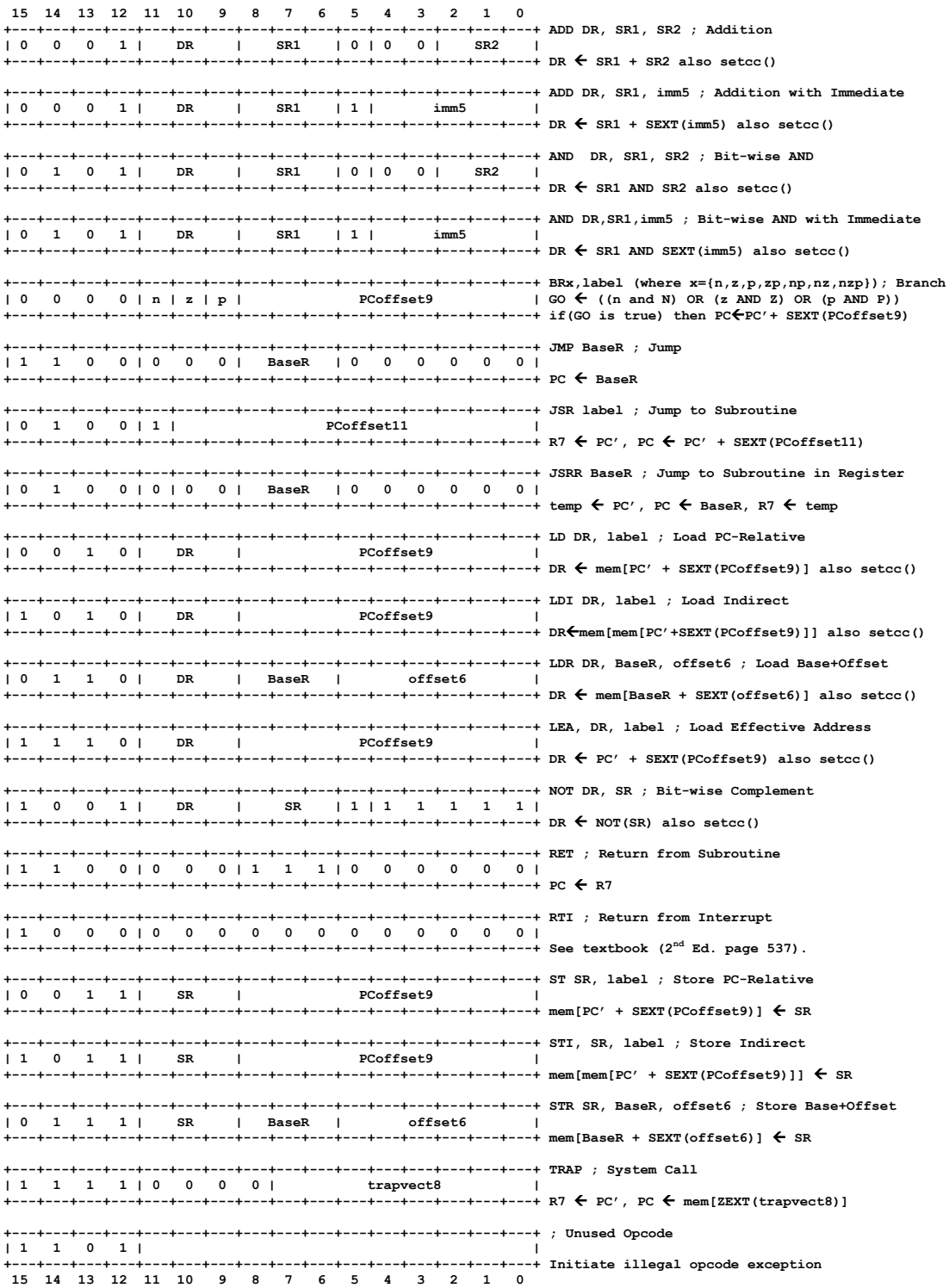
Extra page for hand written work, if needed. This page is not required and will NOT affect your grade. You don't even need to hand this page in.

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	‘ (*Apostr.)*	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	, *(Comma)*	2C	L	4C	l	6C
cr	0D	-	2D	M	4D	m	6D
so	0E	. *(Period)*	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	_ *(Undrscre)*	5F	del	7F

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.



TRAP CODES

Code	*Equivalent*	*Description*
HALT	TRAP x25	Halt execution and print message to console.
IN	TRAP x23	Print prompt on console, read (and echo) one character from keybd. Character stored in R0[7:0].
OUT	TRAP x21	Write one character (in R0[7:0]) to console.
GETC	TRAP x20	Read one character from keyboard. Character stored in R0[7:0].
PUTS	TRAP x22	Write null-terminated string to console. Address of string is in R0.