

# Midterm Exam Study Guide

These are the instructions on page 1 of the test. I am giving them to you here so you can read them **BEFORE** the test. Failure to read and follow these instructions can have dire consequences on your grade.

1. Read these instructions carefully. Failure to follow these instructions exactly will result in loss of points or even a zero on the exam.
2. One 8.5-inch x 11-inch note sheet is allowed with notes front and back. This is a standard sized piece of notebook paper.
3. No calculators are allowed.
4. **PRINT** your first and last name at the top of this page.

**You will lose 5 points if I cannot read your name.**

5. On **SIDE 2** of the Scantron sheet, put each letter of your last name, your first initial, and your middle initial with no spaces and bubble in the letters. Do not include numbers. Simply leave them out. Do not fill in any other sections on SIDE 2.

**You will lose 5 points if you do not enter your name AND bubble in the letters on SIDE 2.**

6. **PRINT** your name on **SIDE 1** of the Scantron sheet on the line provided at the top of the page. Put your section and the letter A after your name if this is exam A. Put the section and the letter B after your name if this is exam B.

**You will lose 5 points if you fail to PRINT your name or fail to put the letter of the exam after your name.**

7. All answers **MUST** be transferred to the Scantron sheet.
8. Scantron sheets must be completed with a #2 pencil. You must erase any incorrect marks completely. If you do not erase completely, that question will be marked wrong.
9. Perform all work on this exam.
10. Work is required for credit. I will assume you are guessing if you show no work and may count the answer wrong.
11. If you fail to turn in this portion of your exam **you will receive a zero for a grade** regardless of your score on the Scantron.
12. **Do not wait until the end to bubble in answers. If you run out of time you WILL NOT be given extra time.**

**This is a guide. Understand that ANYTHING covered in the book, in the notes, or on the assignments and quizzes could be on this exam. The only sure things to ignore are things specifically listed at the end of this guide.**

Quiz questions are a good study place, especially the speed quizzes and LC3 quizzes. The pre-quizzes are a good source of information and understanding, but I rarely use those specific questions directly on the exam.

**THIS IS A LONG TEST. PRACTICE CONVERSIONS OR YOU WILL RUN OUT OF TIME. IF YOU DON'T KNOW AN ANSWER, DON'T SPEND 5 MINUTES SOUL SEARCHING. MARK AN ANSWER AND CONTINUE.**

## Introduction (CH1)

ISA and Microarchitecture - look at quiz questions.

## Number Systems (CH2)

Binary

Hexadecimal

Sign magnitude

1's complement

2's complement

Bits, Powers of 2, Metric Prefixes (KMGT)

Sign/Zero extension

Overflow

Bit operations and masking

Bit vectors

16-bit and 32-bit floating point numbers (no special cases such as infinity or not a number)

## LC3 Assembly Programming

Addressing modes: Immediate, Register, PC-Relative, Indirect, Base + Offset. Know the usage and limitations of each.

Be able to figure out what will be stored in any register or memory location given a list of register values, memory values, and one or more instructions.

Know the differences between all the jump/branch instructions. JSR, JSRR, JMP, BRNZP, RET.

Know how and when to save registers.

Traps. Know what each one does and how to use it in a program. Even the ones we haven't talked about or used in detail.

Assembly language labels. How to use them. Where they go.

Assembly language pseudo-ops and how they work: .orig, .fill, .blkw, .stringz, .end.

Conditional Branches and NZP.

Assembling and simulation techniques: Step, Step Over, Step Out, Continue

Make sure you can write simple LC3 programs. Make sure you can compare numbers and program simple loops.

Know how to print characters and how to print strings. They are different.

Make sure you practice the "LC3 Operations Quizzes." There will be questions exactly like these on the exam.

## **You will be penalized for any of the following on your assembly programs.**

1. You must use assembly. DO NOT TURN IN HEX CODE! You will get zero for not writing assembly.
2. Do not use LDI, JMP, JSRR, IN, or PUTSP in your code.
3. Do not use offsets in your code. You MUST use labels.
4. Do not add more than three times to get a constant. Load using .fill and a label.
5. Do not clear registers that don't need clearing.
6. Do not load a positive value then make it negative.
7. Do not reload constants in a loop.
8. Do not write code that is grossly inefficient.
9. You must use PUTS and LEA to print strings. Do not print one char at a time.
10. Do not use multiple RETs in a subroutine or multiple HALTs in main.
11. If I ask for main code do not write a subroutine.
12. Main code must start with an origin and end with HALT.
13. If I ask for a subroutine, do not write main code.
14. Subroutines must start with a label and end with RET.
15. You must format your code with tabs and use LC3 assembly conventions.
16. For brevity, do not include comments.

### **Stuff that will not be on the test.**

C Programming

RTI – This is the only instruction we didn't discuss. You don't need to know anything about it.  
None of the floating-point special cases (infinity, NAN, etc.)

### **Some other things you should know.**

Make sure you understand hexadecimal numbers. Many of the problems require you to answer in hex or translate bits from hex. If you don't know how to do that conversion you will have trouble on many problems on the test. Hex is easy and you should know it by now.

You should know the common sizes of different data types. These are common, yet C does not define the sizes specifically. You should probably know **8 bits is a byte** also.

**char is 8 bits.**

**short is 16 bits.**

**int is 32 bits.**

**long is 64 bits.**

**float is 32 bits.**

**double is 64 bits.**

## **THE FOLLOWING PAGES WILL BE GIVEN TO YOU ON THE EXAM**

YOU CAN DETACH THIS PAGE IF YOU LIKE.

### LC3 Opcodes - Appendix A

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD <sup>+</sup>	0001				DR			SR1			0	00		SR2		
ADD <sup>+</sup>	0001				DR			SR1			1	imm5				
AND <sup>+</sup>	0101				DR			SR1			0	00		SR2		
AND <sup>+</sup>	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 <sup>+</sup>	0010				DR			PCoffset9								
LDI <sup>+</sup>	1010				DR			PCoffset9								
LDR <sup>+</sup>	0110				DR			BaseR			offset6					
LEA <sup>+</sup>	1110				DR			PCoffset9								
NOT <sup>+</sup>	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															

### Items that can cost you points on your written programs.

1. You must use assembly. DO NOT TURN IN HEX CODE! You will get zero for not writing assembly.
2. Do not use LDI, JMP, JSRR, IN, or PUTSP in your code.
3. Do not use offsets in your code. You MUST use labels.
4. Do not add more than three times to get a constant. Load using .fill and a label.
5. Do not clear registers that don't need clearing.
6. Do not load a positive value then make it negative.
7. Do not reload constants in a loop.
8. Do not write code that is grossly inefficient.
9. You must use PUTS and LEA to print strings. Do not print one char at a time.
10. Do not use multiple RETs in a subroutine or multiple HALTs in main.
11. If I ask for main code do not write a subroutine.
12. Main code must start with an origin and end with HALT.
13. If I ask for a subroutine, do not write main code.
14. Subroutines must start with a label and end with RET.
15. You must format your code with tabs and use LC3 assembly conventions.
16. For brevity, do not include comments.

## TRAPS

GETC x20  
OUT x21  
PUTS x22  
IN x23  
PUTSP x24  
HALT x25

## ASCII CHART

Dec	Hex	Char
0	00	Null
1	01	Start of heading
2	02	Start of text
3	03	End of text
4	04	End of transmission
5	05	Enquiry
6	06	Acknowledge
7	07	Audible bell
8	08	Backspace
9	09	Horizontal tab
10	0A	Line feed
11	0B	Vertical tab
12	0C	Form feed
13	0D	Carriage return
14	0E	Shift out
15	0F	Shift in
16	10	Data link escape
17	11	Device control 1
18	12	Device control 2
19	13	Device control 3
20	14	Device control 4
21	15	Neg. acknowledge
22	16	Synchronous idle
23	17	End tran. Block
24	18	Cancel
25	19	End of medium
26	1A	Substitution
27	1B	Escape
28	1C	File separator
29	1D	Group separator
30	1E	Record separator
31	1F	Unit separator

Dec	Hex	Char
32	20	Space
33	21	!
34	22	"
35	23	#
36	24	\$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?

Dec	Hex	Char
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_

Dec	Hex	Char
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	
125	7D	}
126	7E	~
127	7F	Delete