CMPE 12/L Practice Midterm

November 1, 2016

Instructions:

- This practice exam is based off of last quarters midterm. Due to the rearrangement of the class certain sections were removed. This means the length is not necessarily representative of the given midterm.
- This exam is closed book and closed notes. You may NOT use a calculator.
- Do not remove the staple.
- **The last page is the LC3 instruction and ASCII reference. You may tear it off, but leave the staple.**
- Always show your work in the space provided. If you do not show your work, you will not be given full credit for that problem.
- Do not use extra paper.

1) [pts] Boolean Logic:

a) (pts) Create a logic circuit (gates) for the following truth table:

A	**B**	**C**	**F**
0	0	0	**1**
0	0	1	**1**
0	1	0	**1**
0	1	1	**0**
1	0	0	**0**
1	0	1	**1**
1	1	0	**0**
1	1	1	**1**

b) (pts) Write the truth table for the following boolean circuit:
Assume level of difficulty from the homework

2) [pts] Binary Conversion

a) (pts) Fill in the following table by converting the given number to the other bases. Assume that each number is 8 bits. If number is un-representable by given representation indicate this. **Show your work!**

Decimal	2's complement
	10110110
	11101010
	11111011
	10100111
37	
93	
2	
-8	

b) (pts) convert 431_5 to base 3.
c) (pts) convert 431_6 to base 4.
d) (pts) convert 321_4 to base 2.

3) [pts] Binary Arithmetic

a) (pts) Perform the following arithmetic operations on the unsigned integers. Do **not** convert them to decimal first and **show your "carries"** between digits. Assume variable size is same as digits given. Indicate whether there is **overflow or no overflow**.

```
  0b 1 0 0 1 1 0 1 1
+ 0b 0 0 1 1 0 1 1 0
  0b
```

```
  0x C 1 D 2
+ 0x 2 A A 0
  0x
```

b) (pts) Perform the following arithmetic operations on the 2's complement integers. Do **not** convert them to decimal first and **show your "carries"** between digits. Indicate whether there is **overflow or no overflow**.

```
  0b 1 0 1 0 0 1 0 0
+ 0b 1 1 0 1 1 0 0 1
  0b
```

```
  0b 0 1 0 1 0 1 1 0
+ 0b1 0 0 0 1 1 1 1
  0b
```

4) [pts] LC-3 Architecture

a) (pts) How many memory locations can the LC-3 address?

b) (pts) What is the word size of the LC-3?

c) (pts) How many general purpose registers does the LC-3 have and what are they named?

d) (pts) What two things happen during the FETCH phase of the instruction cycle?

e) (4 pts) List what the PC, IR, MDR, and MAR stand for in the LC-3 architecture and briefly what they do.

5) [pts] Binary Multiplication

Do not convert the numbers to unsigned form and flip the sign back after multiplication
a) (pts) perform -3*-7 in 4 bit 2's complement
a) (pts) perform 2*4 in 4 bit 2's complement
a) (pts) perform -1*6 in 4 bit 2's complement
a) (pts) perform 3*-5 in 4 bit 2's complement

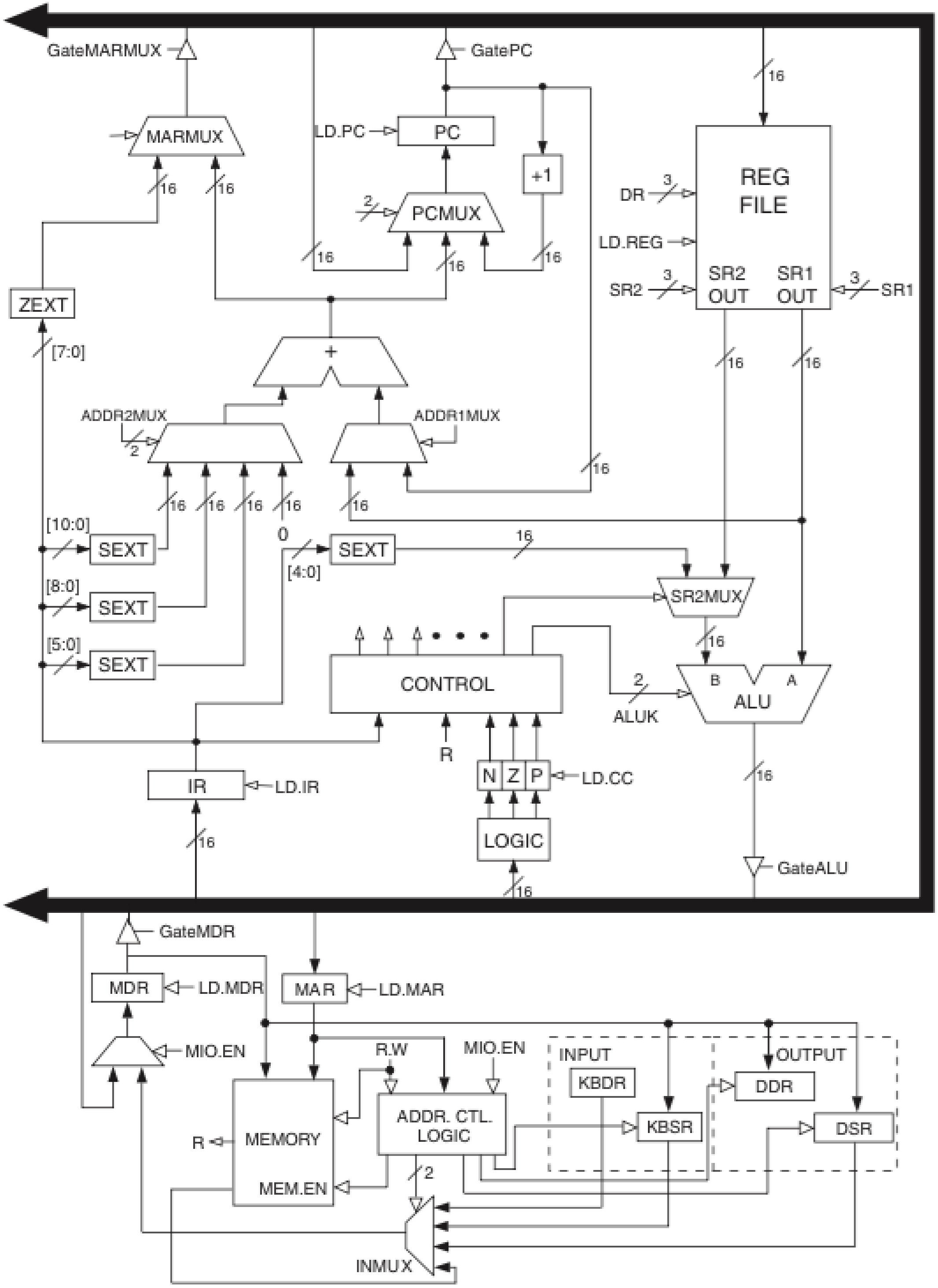
6) [pts] Digital Logic

a) (pts) Draw the gate level diagram for a 1-4 decoder, be sure to label your circuit.

c) (pts) Using only NAND gates, show the implementations for **OR**, **NOT**, and **AND** Logic functions.

7) [pts] LC-3 Datapath

Indicate which data lines are being used and indicate the values on those lines for the instruction `ADD R0, R1, R2`
Assume `R1` contains the value `x1234` and `R2` contains the value `x4`



8) [pts] LC-3 Coding

a) Write LC-3 assembly code that will NAND the values in R1 with R3 and store the result in R0.
b) Write a subroutine for subtraction. Be sure to use safe register calling conventions. Use R0 for the output and R1,R2 for the input

8) [pts] LC-3 Code Running

a) After the following LC-3 code executes what are the ending contents of the registers and memory? Assume some registers/memories have starting values as indicated. If blank, the content is unknown. Remember that both registers and memory locations are 16-bits wide. The memory portion starts at address 0x3200.

```
LEA R1, label0
LDR R2, R1, #0
STR R0, R1, #4
LEA R6, label2
ADD R5, R0, R1
LEA R0, label1
AND R7, R2, R5
NOT R3, R0
STR R7, R6, #-2
STR R2, R1, #1
```

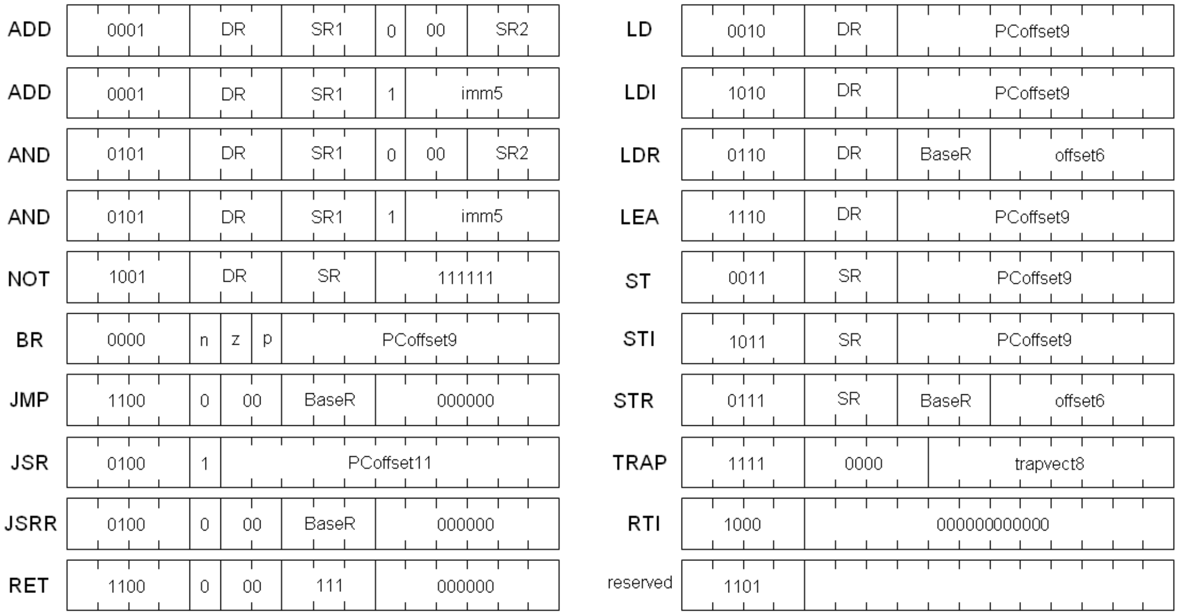
8) [pts] LC-3 Instructions
Decode each instruction and describe the operation it is performing

a) (pts) 0001 0010 1111 1100

b) (pts) 0111 0000 1001 1111

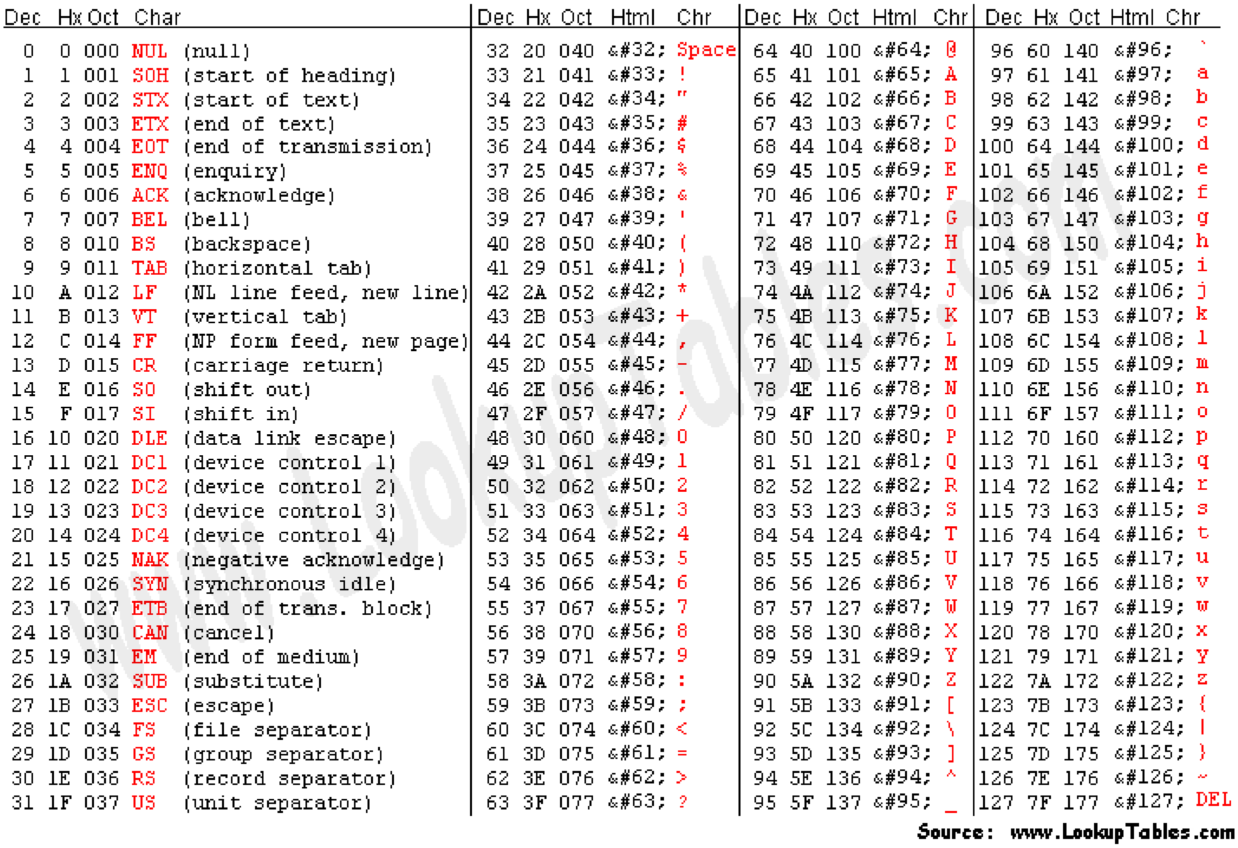
c) (pts) 0000 1011 1111 1001

LC-3 Instructions



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	
NOT	1001	DR	SR	111111		
BR	0000	n	z	p	PCoffset9	
JMP	1100	0	00	BaseR	000000	
JSR	0100	1	PCoffset11			
JSRR	0100	0	00	BaseR	000000	
RET	1100	0	00	111	000000	

LD	0010	DR	PCoffset9	
LDI	1010	DR	PCoffset9	
LDR	0110	DR	BaseR	offset6
LEA	1110	DR	PCoffset9	
ST	0011	SR	PCoffset9	
STI	1011	SR	PCoffset9	
STR	0111	SR	BaseR	offset6
TRAP	1111	0000	trapvect8	
RTI	1000	000000000000		
reserved	1101			



Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	&#32;	Space	64	40	100	&#64;	@	96	60	140	&#96;	`
1	1	001	SOH (start of heading)	33	21	041	&#33;	!	65	41	101	&#65;	A	97	61	141	&#97;	a
2	2	002	STX (start of text)	34	22	042	&#34;	"	66	42	102	&#66;	B	98	62	142	&#98;	b
3	3	003	ETX (end of text)	35	23	043	&#35;	#	67	43	103	&#67;	C	99	63	143	&#99;	c
4	4	004	EOT (end of transmission)	36	24	044	&#36;	$	68	44	104	&#68;	D	100	64	144	&#100;	d
5	5	005	ENQ (enquiry)	37	25	045	&#37;	%	69	45	105	&#69;	E	101	65	145	&#101;	e
6	6	006	ACK (acknowledge)	38	26	046	&#38;	&	70	46	106	&#70;	F	102	66	146	&#102;	f
7	7	007	BEL (bell)	39	27	047	&#39;	'	71	47	107	&#71;	G	103	67	147	&#103;	g
8	8	010	BS (backspace)	40	28	050	&#40;	(	72	48	110	&#72;	H	104	68	150	&#104;	h
9	9	011	TAB (horizontal tab)	41	29	051	&#41;	)	73	49	111	&#73;	I	105	69	151	&#105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	&#42;	*	74	4A	112	&#74;	J	106	6A	152	&#106;	j
11	B	013	VT (vertical tab)	43	2B	053	&#43;	+	75	4B	113	&#75;	K	107	6B	153	&#107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	&#44;	,	76	4C	114	&#76;	L	108	6C	154	&#108;	l
13	D	015	CR (carriage return)	45	2D	055	&#45;	-	77	4D	115	&#77;	M	109	6D	155	&#109;	m
14	E	016	SO (shift out)	46	2E	056	&#46;	.	78	4E	116	&#78;	N	110	6E	156	&#110;	n
15	F	017	SI (shift in)	47	2F	057	&#47;	/	79	4F	117	&#79;	O	111	6F	157	&#111;	o
16	10	020	DLE (data link escape)	48	30	060	&#48;	0	80	50	120	&#80;	P	112	70	160	&#112;	p
17	11	021	DC1 (device control 1)	49	31	061	&#49;	1	81	51	121	&#81;	Q	113	71	161	&#113;	q
18	12	022	DC2 (device control 2)	50	32	062	&#50;	2	82	52	122	&#82;	R	114	72	162	&#114;	r
19	13	023	DC3 (device control 3)	51	33	063	&#51;	3	83	53	123	&#83;	S	115	73	163	&#115;	s
20	14	024	DC4 (device control 4)	52	34	064	&#52;	4	84	54	124	&#84;	T	116	74	164	&#116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	&#53;	5	85	55	125	&#85;	U	117	75	165	&#117;	u
22	16	026	SYN (synchronous idle)	54	36	066	&#54;	6	86	56	126	&#86;	V	118	76	166	&#118;	v
23	17	027	ETB (end of trans. block)	55	37	067	&#55;	7	87	57	127	&#87;	W	119	77	167	&#119;	w
24	18	030	CAN (cancel)	56	38	070	&#56;	8	88	58	130	&#88;	X	120	78	170	&#120;	x
25	19	031	EM (end of medium)	57	39	071	&#57;	9	89	59	131	&#89;	Y	121	79	171	&#121;	y
26	1A	032	SUB (substitute)	58	3A	072	&#58;	:	90	5A	132	&#90;	Z	122	7A	172	&#122;	z
27	1B	033	ESC (escape)	59	3B	073	&#59;	;	91	5B	133	&#91;	[	123	7B	173	&#123;	{
28	1C	034	FS (file separator)	60	3C	074	&#60;	<	92	5C	134	&#92;	\	124	7C	174	&#124;	|
29	1D	035	GS (group separator)	61	3D	075	&#61;	=	93	5D	135	&#93;	]	125	7D	175	&#125;	}
30	1E	036	RS (record separator)	62	3E	076	&#62;	>	94	5E	136	&#94;	^	126	7E	176	&#126;	~
31	1F	037	US (unit separator)	63	3F	077	&#63;	?	95	5F	137	&#95;	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com