PART A(25 points) Short Answers

A1 (3 points) Add the two hexadecimal 2's complement integers below:

x9A6
\+ x675E

A2 (5 points) What is the positive range represented by the IEEE single-precision floating point representation (normalization form, exponent cannot be all 1 or all 0)?

Please use the IEEE single-precision floating point representation to represent 42.375.

A3 (3 points) Considering a memory of 8G Bytes, assuming the size of instruction or data stored in the memory is 16-bit, what are the address space and the addressability of the memory, respectively?

A4 (5 points) Is the pair of functions, f1(A,B) and f2(A,B), together logically complete? Justify your answer.

A	B	f1(A,B)	f2(A,B)
0	0	1	0
0	1	0	1
1	0	1	0
1	1	0	0

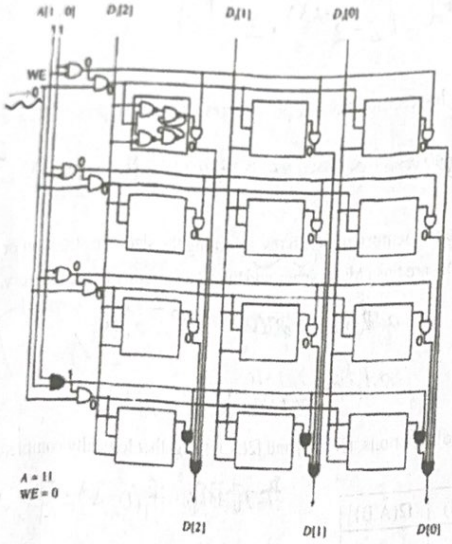
A5 4 points) Consider the following LC-3 instruction:

LDI R5, x100 ; the address of the instruction is X4000

Assuming the address of the instruction is X4000, what is the value of R5 after running this instruction? Some contents of the memory are shown below:

Address	X2786	X2787	X2788	X3F00	X3F01	X3902	X4100	X4101	X4102
Content	X4100	X4101	X4102	X2786	X2787	X2788	X3900	X3901	X3902

A6 (5 points) The following diagram is a gate circuit diagram of 2^2-3-bit memory. Memory access requires decoding the address, which means the decoder decodes input A [1:0] into 4 output lines. As shown in the figure, each word line in memory contains 3 bits (i.e. 1 word), which is the origin of the term "word line". When reading memory, as long as the address value A [1:0] is set, the corresponding word line is selected for output. Each bit in memory is **AND**ed with its corresponding word line, and **OR**ed with the same bit on other word lines. Since only one word line is selected at any time, this is actually a bit selection multiplexer switch. Three such bit-level multiplexers are connected together to form a word selector switch, which reads one word at a time.



We already know the values of memory locations:

$D_i[j]$ i	$D_i[2]$	$D_i[1]$	$D_i[0]$
0	1	0	1
1	0	1	y
2	0	x	0
3	1	1	0

1) (3 points) If A [0] = 1 and A [1] = 0, we will get D1 as the output. If A [0] = 0 and A [1] = 1, we will get D2 as the output. If A [0] = 0 and A [1] = 0, we will get D3 as the output. Assuming D1+D2=D3, what are the values of x and y, respectively?

2) (2 points) What would be the impact if the value where the arrow points becomes 1?

PART B(26 points) Digital Logic Structures

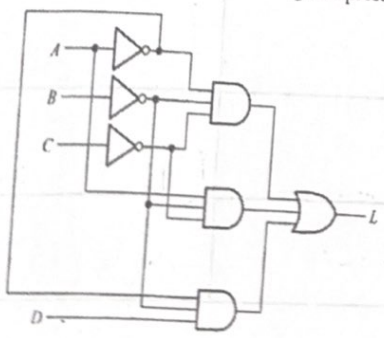
B1(6 points) Please implement a **transistor-level** circuit for the following truth table, where A, B and C are input signals, and L is an output signal.

Hint: you can first implement some parts of the circuit as sub-modules, and then build the entire circuit based on these sub-modules.

A	B	C	L
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

B2(7 points)

1) (2 points) Please write the logic expression for the circuit below.



2) (3 points) Re-write the expression using only NOT and NAND.

3) (2 points) Draw the logic diagram implemented using only NOT gates and NAND gates.

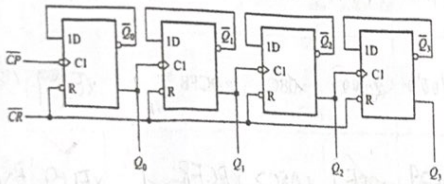
B3(3 points) Use only NOT gates and 2-input NOR gates to draw a logic circuit diagram for expression $L = A\bar{B} + \bar{A}C$.

B4: (10 points)

1) (2 points) Please explain why state machines must be implemented using sequential circuits and cannot be realized with only combinational circuits.

2) (2 points) D-trigger, is a type of digital storage element in digital circuits. It has three inputs: a data input (D), clear input (R) and a clock input (CLK). When the clear signal (R) is asserted, it forces the output (Q) to be '0'. The clock input determines when the data input should be sampled. When clock signal rising edge occurs, it will make Q = D.

Your job: Describe what the following sequential circuit does in **one sentence**. Assume that the output $Q_0Q_1Q_2Q_3$ is initially 0000.



3) (6 points) Design a Finite State Machine (FSM) that recognizes patterns of "000" and "010" in an input string and outputs 1 when either pattern is detected.

For example, giving a string 00001000, it will produce 00110101

PART C: The Von Neumann Model (20 points)

For the following questions, refer to the LC-3 Von Neumann Model below. The current PC, register file and part of the memory is provided, and the LC-3 computer is about to perform a **FETCH**

Address	Instruction / Data	Note
x3060	0110 001 111 000000	R1←M[R7]
x3061	0110 010 111 000001	R2←M[R7+1]
x3062	1001 011 010 111111	R3←~R2
x3063		
x3064	0101 110 110 1 00000	R6←0
x3065	0001 110 110 1 00001	R6←R6+1
x3066	0101 100 010 0 00 110	R4←R2 & R6
x3067	0000 010 000000001	
x3068	0001 000 000 000 001	R0 ←R0 + R1
x3069	0001 001 001 000 001	R1 << 1
x306a	0001 110 110 000 110	R6 << 1
x306b	0001 101 110 000 011	R5←R6 + R3
x306c	0000 110 111111001	
x306d		

Component	R0	R1	R2	R3	R4	R5	R6	R7
Value(Hex)	x0006	x04D7	x9B1F	x1145	xAE86	x8848	x0C5B	X3100

Component	IR	PC
Value(Hex)	---	x3060

C1 (4 points) After the **execution** of the **instruction at x306d**, the partial results in memory are as follows.

Address	Instruction / Data	Note
x3100	0000 0000 0001 0010	
x3101	0000 0000 0000 1111	
x3102	0000 0001 0000 1110	
x3103	0000 0001 0001 0100	
x3104	0000 0001 0010 0000	

Complete the missing data between x3060 and x306d based on the provided information. (The "note" section in the memory table is only for your convenience and will not be considered in the grading. Some notes have already been provided and ensured to be correct.)

C2 *(12 points) When the machine* ***completes the execution of an instruction*** *in memory address x3060 to x306d, the status of some registers of the machine is shown in the table below. Please fill in the blanks in the following three tables according to the given information.*

1) *(6 points) Register File*

Component	R0	R1	R2	R3	R4	R5	R6	R7
Value(Hex)			x000F	xFFF1		xFFF9	x0008	x3100

2) *(2 points) IR and PC*

Component	IR	PC
Value(Hex)		x3066

3) *(4 points) MAR and MDR*

Component	MAR	MDR
Value(Hex)		

C3*(4 points) Describe the function of the instructions between x3060 and x306d.* **(Within 30 words.** *You can use specific addresses, registers, and data for description.*

PART D: The LC-3 (28 points)

D1. (16 points) The PC is initially loaded with x3000 and some instructions in x3000~x300D are shown in the Table D.1. We've hidden the middle 7 instructions, but you can restore them by following the tips in Table 2 below. Table D.2 contains the contents of some registers after executing each of the 7 instructions, when they are executed for the first time.

Your job is to complete the Table D.2

Table D.1:

Address	Instruction / Data	Note
x3000	0101 010 010 1 00000	AND R2, R2, #0
x3001	1001 001 001 1 11111	NOT R1, R1
x3002	0001 001 001 1 00001	ADD R1, R1, #1
....	....	....
x300A	0101 010 010 1 00000	AND R2, R2, #0
x300B	0001 010 010 1 11111	ADD R2, R2, #-1
x300C	0011 010 0 0000 0101	ST R2, #5
x300D	1111 0000 0010 0101	HALT

...nction in memory address
...the table below. Please fill
...tion.

5	R6	R7
F9	x0008	x3100

...06d. **(Within 30**

...000~x300D
...estore them
...isters after

Table D.2:

	PC	MAR	MDR	IR	R0	R1	R2	R3
After executing x3002	x3003	/	/	/	x0461	xDCF8	0	0
After executing x3003	x3004	x3003			x0461	xDCF8	0	xFFF0
After executing x3004	x3005	x3004	x02	x0)				
After executing x3005	x3006	x3005	x1	x1___			0	xE159
After executing x3006	x3007	x3006	x0	x0				
After executing x3007	x3008	x3007			x0461	xDCF8	1	
After executing x3008	x3009	x3008			x08C2	xDCF8	1	
After executing x3009			x0F	x0F__				

D2(4 points). What does this program do ?

D3 (8 points). How to do right shift and compare? Complete the following program.

Hints:

We fix the shift bits as 3 and we store 1/0 as the compare result.

Suppose all the registers are initialized to 0.

Address	Instruction / Data
X3000	X9__
X3001	X1261
X3002	X1DA1
X3003	X1___
X3004	X1__
X3005	X58___
X3006	X0 01
X3007	X1B43
X3008	X16C3
X3009	X1482
X300A	X0 FA
X300B	X1___
X300C	X0401
X300D	X1D___
X300E	X3C05
X300F	XF025