Introduction to Computing Systems (2021 Fall)

Mid-term Exam

Saturday, November 27th, 2021

Student_ID: _____ Name: _____

Organization	*Score*
Signature (1 point)	
A.Short Answers (20 points)	
B.Digital Logic Structures(29 points)	
C.Von Neumann Model (20 points)	
D.LC-3 Programming (30 points)	
Total (100 points)	

I will not cheat on this exam.

_____signature (1 point)

PART A: Short answers (20 points)

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

x985
+ x4123

F985
4123
3AA8

A2 (3 points): The result of adding two floating point numbers x7F800000 and xFF800000.

NaN

A3 (3 points): I need a memory that contains a total of 4G Bytes. If the size of instruction or the data stored in each memory location is 32-bit, what are the address space and the addressability of the memory?

addressability 4 bytes
address space 1G

A4 (3 points): What's the largest positive normalized number that ***can be represented*** using the 32-bit IEEE floating point number?

0 1111 1110 1111 ··· 1 $1.1\cdots1 \times 2^{127}$

A5 (3 points):What's the smallest positive integer that a ***normalized*** floating point number with n-bit fraction ***cannot represent***? Assume the number of bit of the ***exponent*** is large enough.

A6 (5 points): Consider the following LC-3 instruction (x3500 is the address where the instruction is located):

x3500 LD R5,_____ ; we want to load register R5 with the value x2BFF

Where could the value x2BFF be stored in memory? Please give the memory address range.

9 bit PCoffset [-x100 + x3500+1 , xFF + x3500+1]

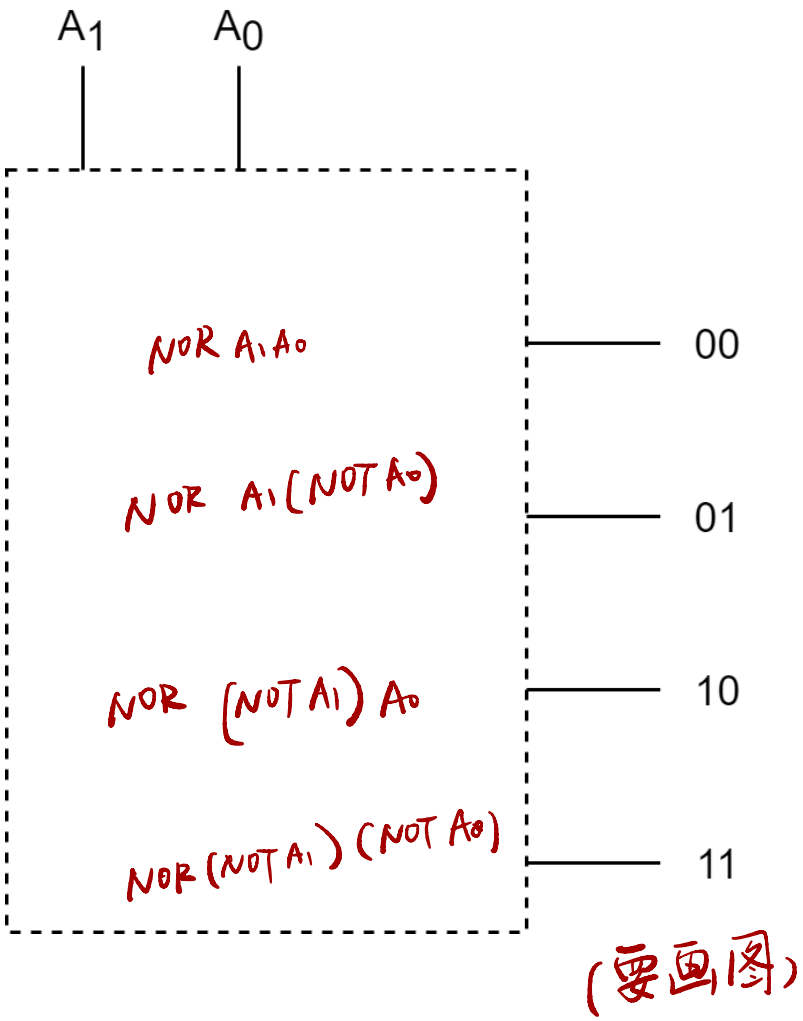
= [x3401, x3600]

PART B: Digital Logic Structures (29 points)

B1

i) (4 points) Prove that NOR is logically complete

ii) (4 points) Implement 2-4 decoder use only NOR and NOT gates.



B2 (10 points): Rock/paper/scissors (石头/布/剪刀) is a two-person game which most of you played in your childhood. During each round of the game, the two players simultaneously form one of three SHAPES (rock, paper, or scissors) with an outstretched arm. If one player shows rock and the other shows scissors, the player showing rock wins (since rock crushes scissors). Similarly paper covers rock, and scissors cuts paper. If both players choose the same shape, the round ends in a draw(平局). The players continue to play rounds as long as they wish to play the game.

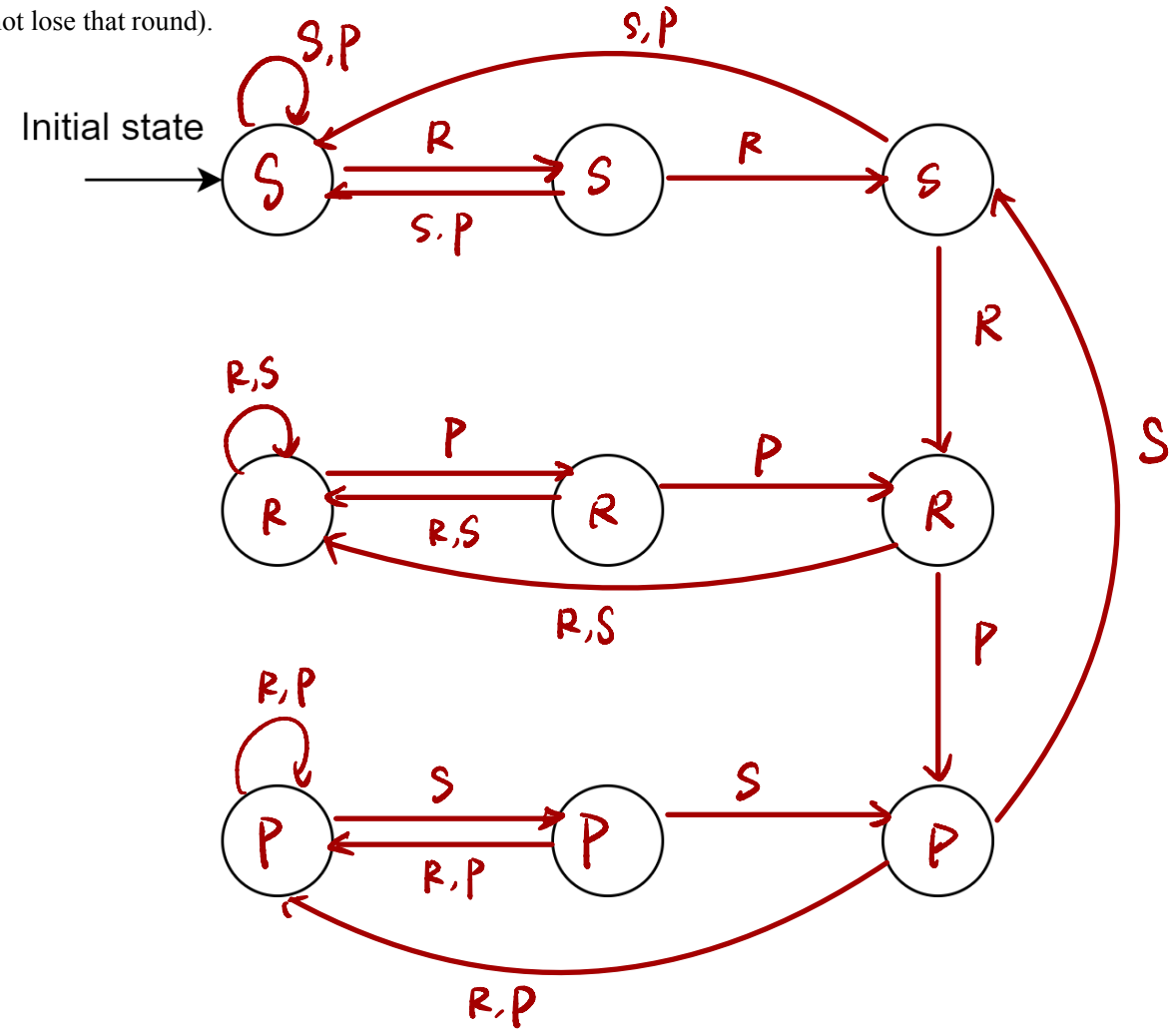
Zhang San has devised a strategy for rock/paper/scissors. The strategy is as follow:

- He always plays "scissors" in the first round.
- He plays the same shape again unless he lost the last three rounds. In that case, he plays the shape his opponent played in the previous round.

Your job: Construct a finite state machine for what Zhang San should play in the current round. The input is the shape that his opponent plays in the current round. On your finite state machine designate "rock" as "R", "paper" as "P", and "scissors" as "S".

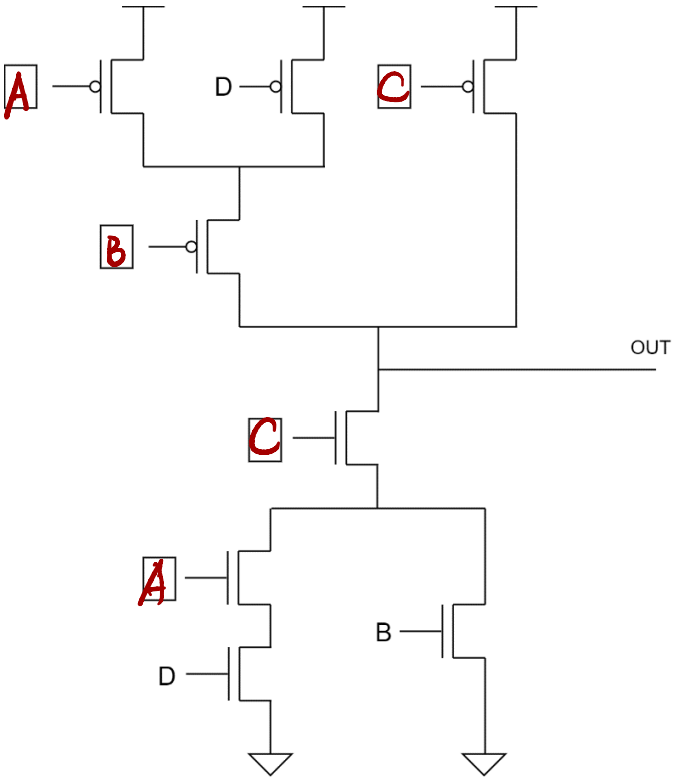
Note: Your finite state machine will not provide any signals to designate the winner or loser of a round. It will also not designate when the players stop playing the game.

Hint: If Zhang San plays the same shape as his opponent, the round ends in a draw (i.e., Zhang San did not lose that round).



B3 (11 points) Shown below is a transistor circuit, having four inputs (A,B,C,D) and one output (out). Also shown is the truth table for this circuit. The gates of some of the transisors are not labeled, and the outputs of some of the input combinations in the truth table are not shown.

Your job: Complete the transistor diagram by labeling the missing inputs to the gates, and by adding the missing outputs to the truth table. Every input combination produces an ouput of either 0 or 1. The result will be a transistor diagram and the truth table describing its behavior.



A	B	C	D	OUT
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	1
0	1	0	0	
0	1	0	1	
0	1	1	0	0
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	1
1	0	1	1	0
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

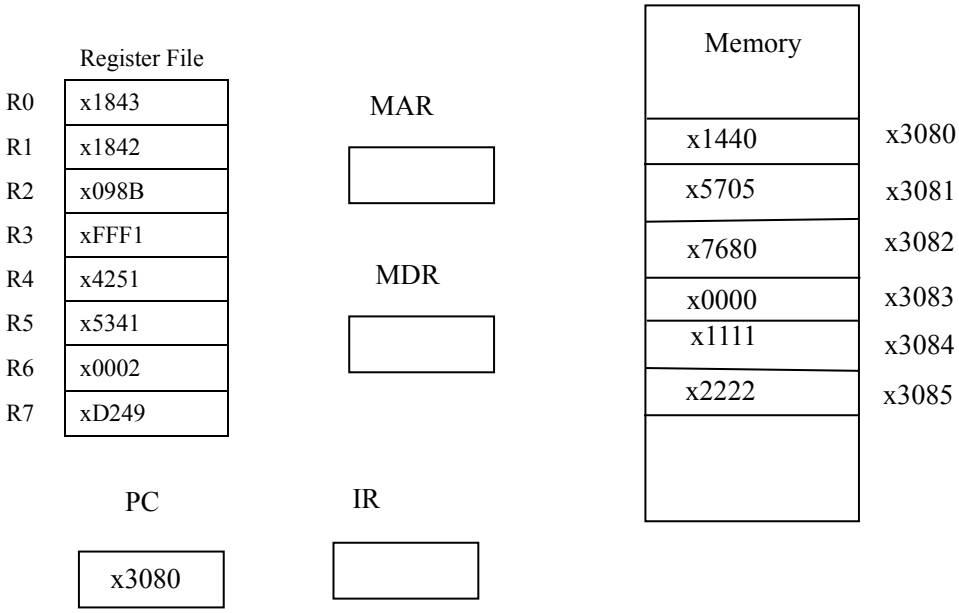
PART C: The Von Neumann Model (20 points)

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

C1 (4 points) The data in memory locations x3080, x3081 and x3082 are indicated in hexadecimal on the diagram. For each data word shown in memory, translate them into LC-3 assembly language.

Address	LC-3
x3080	ADD R2, R1, R0
x3081	AND R3, R0, R5
x3082	STR R3, R2, #0

C2 (16 points) In the diagram below, the LC-3 is about to start the instruction cycle for the instruction at x3080. Update the diagram to reflect the state of the machine after the execution of **three** full instruction cycles(before fetch the forth instruction). Clearly denote the final values in the following components within the diagram below.



i) (4 points) Register File

Component	R0	R1	R2	R3	R4	R5	R6	R7
Value(Hex)	..	..	x3085	x4241	..	..	..	..

ii) (4 points) IR and PC

Component	IR	PC
Value(Hex)	x7680	x3083

打..的表示不变

iii) (4 points) MAR and MDR

Component	MAR	MDR
Value(Hex)	x3085	x4241

iiii) (4 points) Memory

Address	x3080	x3081	x3082	x3083	x3084	x3085
Value(Hex)						X4241

PART D: The LC-3 (30 points)

D1. (5 points) The code below is a program to multiply two positive integers in R1 and R2 respectively. The result is written into R0.

x3000	0101 0000 0010 0000	AND R0 ← R0 , #0
x3001	0001 0010 0111 1111	ADD R1 ← R1 , #-1
x3002	0000 1000 0000 0010	BRn x3005
x3003	0001 0000 1000 0000	ADD R0 ← R2 , R0
x3004	0000 1111 1111 1100	BRnzp x3001
x3005	1111 0000 0010 0101	HALT

What results will be stored in R0 if we replace the instruction in x3003 with ***ADD R0 ← R0, R1***? Use R1 or R2 to represent your answer.

$$R_1 - 1 + \cdots + 0 = \frac{R_1(R_1-1)}{2}$$

D2. (5 points) The LC-3 ISA contains an instruction LDR. After the instruction is decoded, the following operations (called micro-instructions) are carried out to complete the processing of the LDR instruction:

MAR ← BaseR + SEXT(Offset6) ; set up the memory address
MDR ← Memory[MAR] ; read mem at BaseR + offset
DR ← MDR ; load DR

We want to add a new instruction ***MOVE DR, SR*** that would copy ***the data*** in memory location whose address is in SR and store it into the memory location whose address is in DR. What sequence of microinstructions of ***MOVE***, following the decode operation, would be?

MAR ← SR
MDR ← M[MAR]
MAR ← DR
M[MAR] ← MDR

D3. (20 points) The program starts at x3000 and compares two numbers that are stored in x3100 and x3101 respectively, and it puts the larger number in R1. If the numbers are equal, then R1 is set equal to 0. The program only uses R1, R2, R3, and R4, and all these registers are initialized with 0.

i) (10 points) Fill in the remaining entries in the program

Address	Instruction	
x3000	0010 0100 1111 1111	LD R2 x3100
x3001	0010 0110 1111 1111	LD R3 x3101
x3002	1001 1000 1111 1111	NOT R4 R3
x3003	0001 1001 0010 0001	ADD R4, R4, #1
x3004	0001 0011 0000 0010	ADD R1 R4, R2
x3005	0000 0100 0000 0010	BRz x3009 / BRz x3011
x3006	0000 1000 0000 0001	BRn x3008
X3007	0000 0010 0000 0010	BRp x3010
X3008	0101 0010 1100 0011	AND R1, R2, R3 / ADD R1, R3, #0
X3009	0000 1110 0000 0001	BRnzp x3011
X3010	0001 0010 0100 0011	ADD R1, R1, R3 / ADD R1, R2, #0 / AND R1, R3, R2
X3011	1111 0000 0010 0101	TRAP x25

ii) (10 points) Assume the value in address x3100 is x1001 and the value in x3101 is x0090. Please fill in the remaining entries in the following table.

	IR	**MAR**	**MDR**	**R1**	**R2**	**R3**	**R4**
Initially	----	----	----	x0000	x0000	x0000	x0000
After instruction at x3000	x24FF	x3100	x1001	x0000	x1001	x0000	x0000
After instruction at x3001	x26FF	x3101	x0090	x0000	x1001	x0090	x0000
After instruction at x3002	x98FF	x3002	x98FF	x0000	x1001	x0090	xFF6F
After instruction at x3003	x1921	x3003	x1921	x0000			
After instruction at x3004	x1302	x3004	x1302				