Homework #3 **COMPUTER ARCHITECTURE**

CSc 4210/6210

Due Friday, March 12th, but can be submitted by Thursday, March 18th without penalty at 11:59 pm

Late Deadline: Saturday, March 20th at 11:59 pm

For any questions regarding this assignment, please reach out to TA Babatunde Bello at bbello1@student.gsu.edu

Part I : For all students

1- Consider the instruction formats of the basic computer in Fig. 5-5 and the list of instructions in Table 5-2. For each of the following 16-bit instructions, give the equivalent 16 bit binary code, write the equivalent RTL statement and find the final binary value of the destination when the instruction is executed. All values are given in hexadecimals and M[AR]=memory.

a. Complete the following tables using the instruction: 4450, where M[450]= 0990

Instruction code (Hexadecimal)	Instruction(binary)	Instruction RTL
4450	0100 0100 0101 0000	PC ← AR

Final value (Hexadecimal)	Final value (binary)
0990	0000 1001 1001 0000

b. Complete the following tables using the instruction: C450, where, M[450]=0890, and M[890]= AF02.

Instruction code (Hexadecimal)	Instruction(binary)	Instruction RTL
C450	1100 0100 0101 0000	PC ← AR

Final value (Hexadecimal)	Final value (binary)
AF02	1010 1111 0000 0010

c. Complete the following table with the instruction; 7040, where AC=5007 and E=0

Instruction code (Hexadecimal)	Instruction(binary)	Instruction RTL
7040	0111 0000 0100 0000	AC(15:1) ← AC(14:0), AC(0) ← E, E ← AC(15)

Final value (Hexadecimal)	Final value (binary)
E080	1110 0000 1000 0000

2- The following microoperations can't be executed during a single clock pulse in the system shown in Fig. 5-4. Specify the sequence of microoperations that will perform the operation.

a-

Single microoperation	Sequence of microoperations
AC← TR	DR ← TR
	AC ← DR

b-

Single microoperation	Sequence of microoperations
AC ← DR + M[AR]	AC ← M[AR]
	AC ← AC + DR

c-

Single microoperation	Sequence of microoperations
M[PC]← DR	AR ← PC
	M[AR] ← DR

d-

Single microoperation	Sequence of microoperations
AC ← M[AR]	DR ← M[AR]
	AC ← DR

e-

Single microoperation	Sequence of microoperations
M[AR]← M[AR]+1	DR ← M[AR]
	DR ← DR + 1
	M[AR] ← DR

3- Complete the following table (all values are in hexadecimal).

Control condition	RTL	I	AC	DR	IR	PC	AR	M[AR]
	Initial Values	0	CD22	F0E0	1EA1	A09	BA8	0061
		0	CD22	F0E0	1EA1	A09	061	02FA
T0:	AR← PC	0	CD22	F0E0	1EA1	A09	A09	9BA8
T1:	IR← M[AR] , PC ← PC+1	0	CD22	F0E0	9BA8	A10	A09	9BA8
T2:	AR←IR(0-11), I← IR(15), D_0:D_7←IR(14:12)	1	CD22	F0E0	9BA8	A10	BA8	0061
T3:D7'IT3:	AR← M[AR]	1	CD22	F0E0	9BA8	A10	061	02FA
T4	DR <- M[AR]	1	CD22	02FA	9BA8	A10	061	02FA

T5	AC <- AC + DR	1	D01C	02FA	9BA8	A10	061	02FA

4- The following register transfers are to be executed in the system of Fig. 5-4. for each transfer, use the following table to specify:

(1) the binary value that must be applied to the bus select inputs S2, S1, and S0;
(2) the register whose LD control input must be active (if any);
(3) a memory control read or write operation (if needed).

a. T5:TR ← PC
b. T6:M[AR] ← IR
c. T7:AC ← DR, PC ← AC (done simultaneously)

i- Complete the following table;

	RTL	s_2	s_1	s_0	Load(LD)	Memory control
a	T5:TR← PC	0	1	0	TR	N/A
b	T6:M[AR]← IR	1	0	1	N/A	Write
c	T7:AC←DR, PC←AC	1	0	0	AC, PC	N/A

ii- Write a Boolean expression in terms of T for s_0, s_1 and s_2.

S_0= T6

S_1= T5

S_2= T6 + T7

5- Refer to Table 5-6 for this problem. However, assume that the following changes are applied to the table:

Instead of PC being incremented with control logic R'T1, it is incremented during the decode stage with control logic R'T2. Also, assume we have a new instruction added in that will automatically skip the next instruction. The control logic and microoperation corresponding to this new SKP instruction is D7IT3B5: PC ← PC + 1.

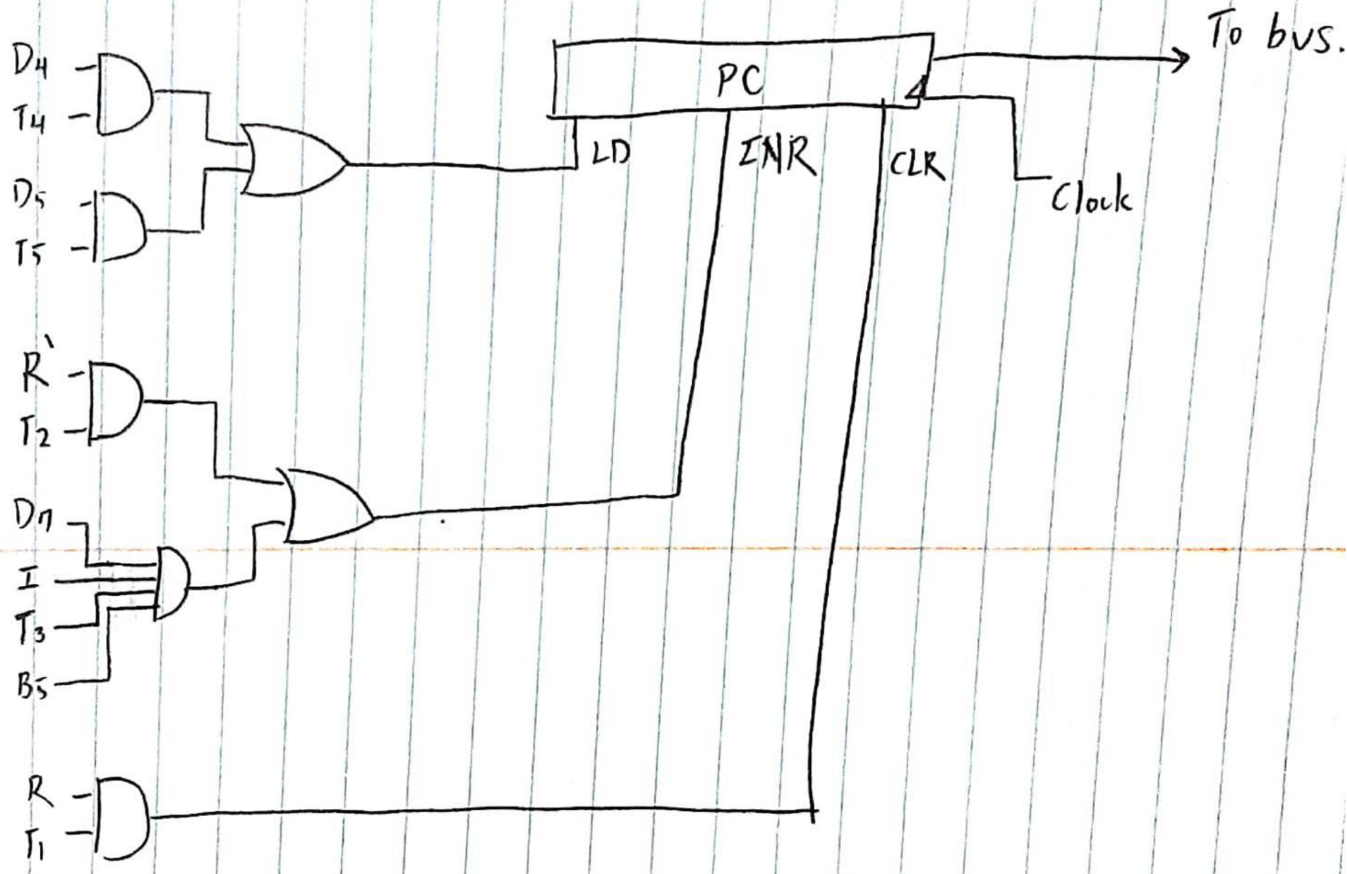
Considering these two changes to the table, derive the control gates associated with the program counter PC for the Mano basic computer model.

Since the PC incremented is moved to R'T2, and we also add a SKP instruction at D7IT3B5.

We end up having following microoperations that include PC:

Load: D4T4, D5T5
Clear: RT1
Increment: R'T2, D7IT3B5

The control gate will be LD(PC) = (D4T4+D5T5), CLR(PC) = RT1, INR(PC) = (R'T2+D7IT3B5)


D_4
T_4
D_5
T_5
R'
T_2
D_7
I
T_3
B_5
R
T_1
PC
LD
INR
CLR
Clock
To bus.