EEL3701 – Dr. Gugel **Last Name**______________________________ First Name ___________________
Fall 2018
Final Quiz Open book/open notes, **90-minute** exam. *** ***No electronic devices permitted!*** ***

Page 1) 10 pt. __________________ Page 2) 14 pt. _________________

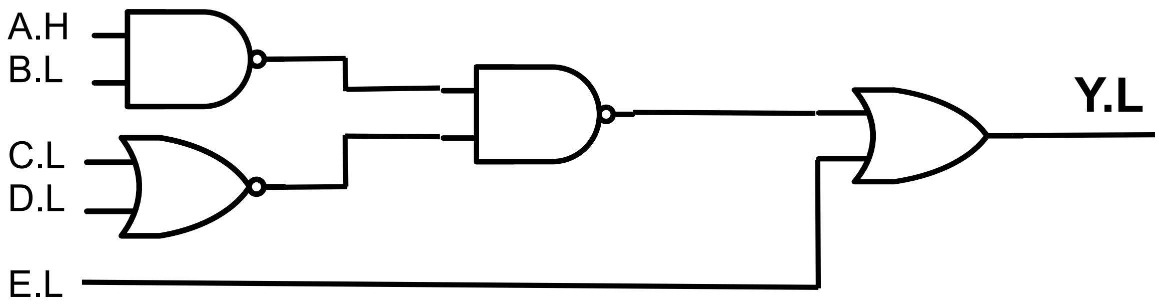
Page 3) 14 pt. __________________ Page 4) 13 pt. _________________

TOTAL _____________________ (51)

1. Implement the logic equation below **using only 3 Input NOR gates**. (5 pt.)

$$Y = \overline{(A*B)} * (D + \overline{E}) \quad ; A.H, B.L, D.L, E.L, Y.H$$

2. For the circuit below, derive **Y.L**. **DO NOT SIMPLIFY!** (5 pt.)



Y.L = ____

Pg. Score =

3A. For the simple CPU in **Appendix A** and the following ROM contents, fill out the Cycle Table below for instructions stored in ROM. ***Assume all registers are initially reset to zero***. Use 'X' to indicate a "don't care" condition and show **all answers in HEX**. (9 pts.)

Address	=>	0	1	2	3
Data (Hex)	=>	F7	F1	F7	F2

Show ALL VALUES IN HEX!

Cycle	State	Input3:0	IR	PC	Reg. A3:0	MSA1:0	MSC2:0
1 (reset)	0		0	0	0		
2							
3							
4							
5							
6							
7							
8							

3B. After the instructions above are executed, what should the contents of Register A:30 be in the 9th cycle?

____ HEX (1 pt.)

4. In your simple CPU used in **Lab #9**, what ***HARDWARE modifications*** and ***new HARDWARE*** is required to add the new instruction below? (4 pt.)

STAA Addr ;store register A to memory specified by the **8 bit address** operand "**Addr**"

____

____

____

____

____

5. See **Appendix B** to write the required program below. Write your code in the left column first and then wrap around to the right column for extra lines/space. (8 pt.)

______________________________	______________________________
______________________________	______________________________
______________________________	______________________________
______________________________	______________________________
______________________________	______________________________
______________________________	______________________________
______________________________	______________________________
______________________________	______________________________
______________________________	______________________________
______________________________	______________________________
______________________________	______________________________

6. See the attached G-CPU program in Appendix C. Assuming that the code was placed in **ROM** starting at address **0x0**. Answer the questions below:

6A. Based on the program execution, what size SRAM must exist in the G-CPU memory map?

RAM size ____ (1 pt.)

6B. What is the -RAM_CE Logic Equation? ______________________________ (1 pt.)

6C. If the **clock is 50 MHz**, how many seconds does it take to execute the **STAB 0xFFFF** instruction? (1 pt.)

STAB 0xFFFF Execution Time = ______________________ **seconds**

6D. Assuming that the loop for the code in Appendix C is executed **twice**, show all SRAM addresses/data modified by the loop. (3 pt.)

ADDRESS (Hex)	DATA (Hex)

Page Score =

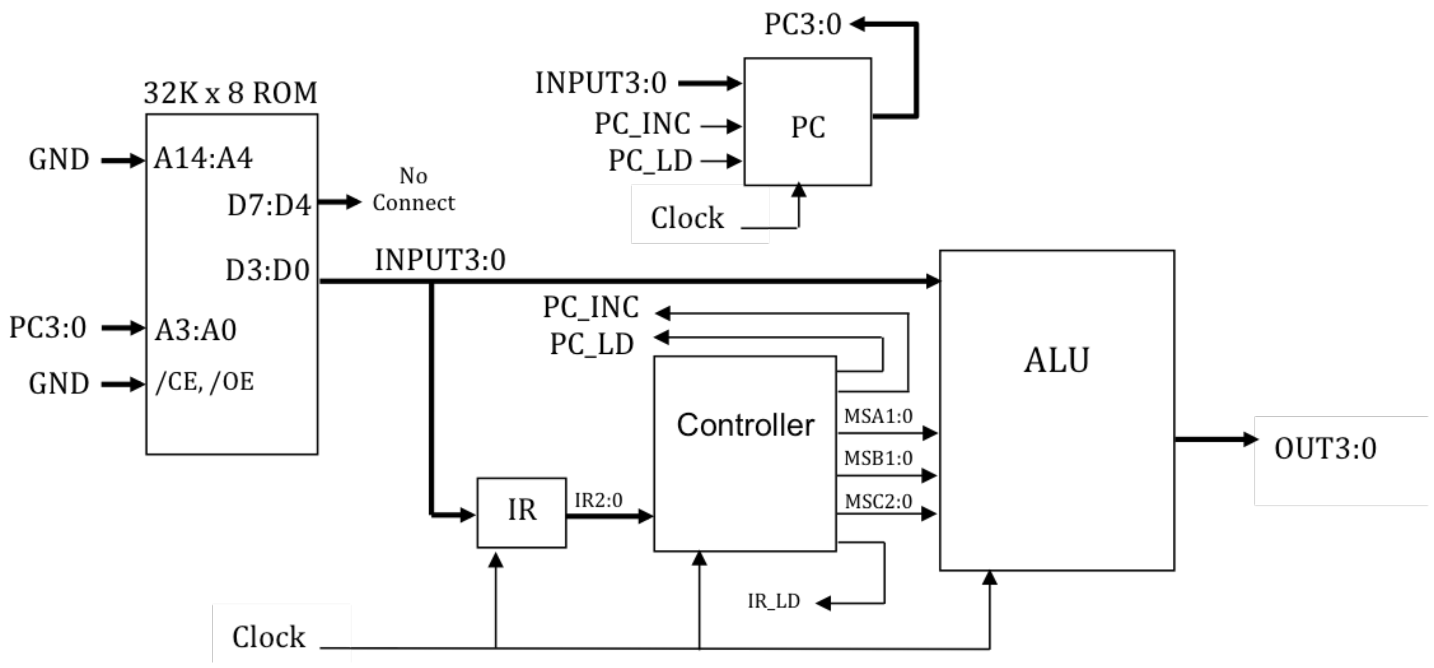
6E. Fill out the cycle diagram below for execution of the **LDAA 5,Y** and **STAB 0,X** instructions in the **SECOND PASS** of the loop. **Show ALL VALUES IN HEX.** (12 pt.)

Cycle	State	Addr Bus	Data Bus	Reg Driving Addr Bus	Device Driving Data Bus	IR	PC	X Reg	Y Reg	R/-W
1		**0012**								
2										
3										
4										
5										
6										
7										
8										

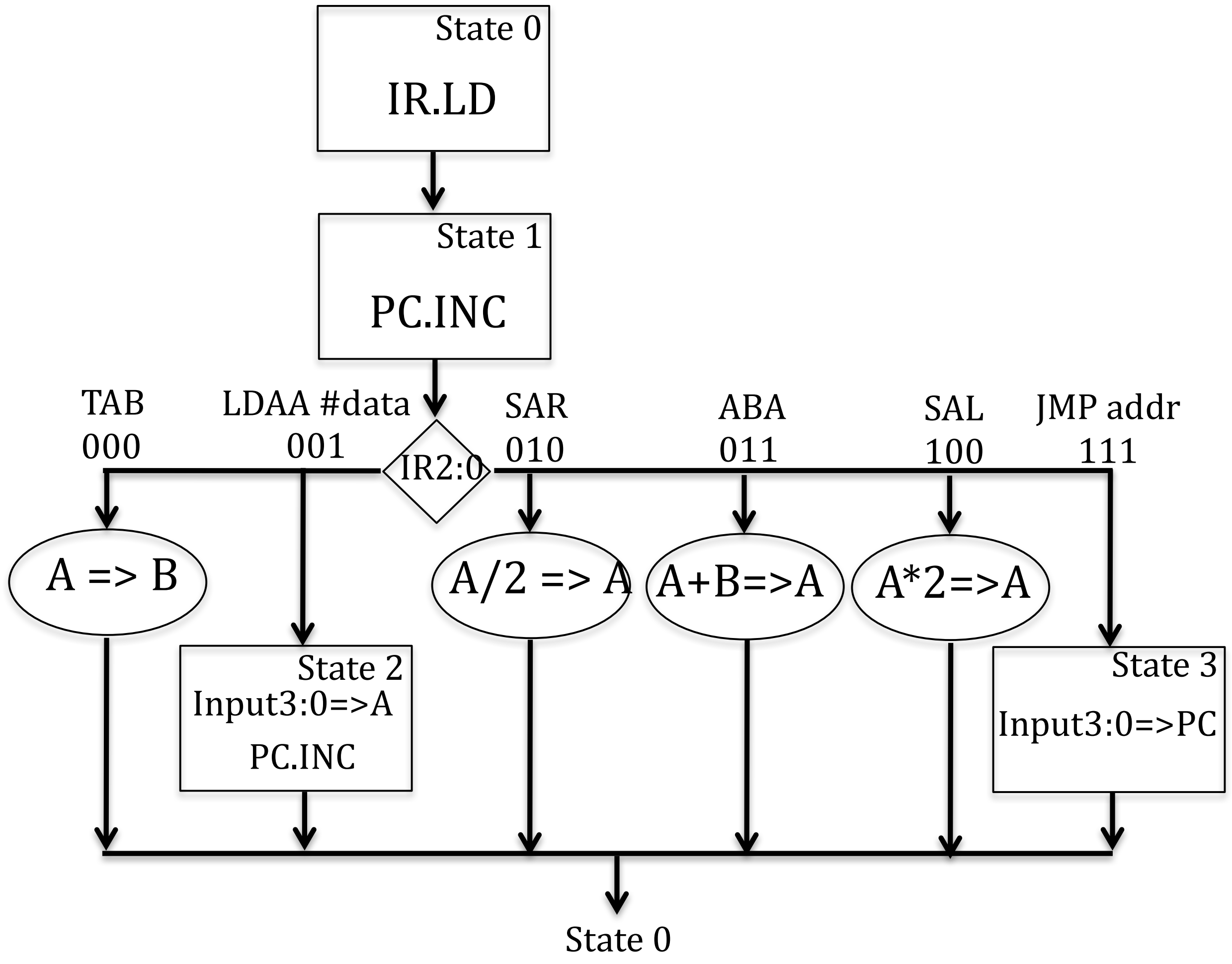
7. ***Extra Credit.*** How many times does the loop run in **HEX**? (1 pt.) ____________________________ **Hex**

Page Score =

Appendix A. *Elementary CPU (Lab #9/Problem #3), Controller ASM & ROM Contents*



Controller ASM:



Page Score =

Appendix B. *Programming Problem #4*

Write a G-CPU assembly program to sum the number of **-127** values stored in RAM starting at address **0x4081** and ***terminating with a zero*** (last value in the array). The final "**SUM**" *should be* ***stored at 0x4080*** and should be used as temporary storage for the **SUM** during your program execution. Use the **Y register** as your **input pointer** and register **B** to hold the required **constant**.

Assume the following **new instructions** are available:

DECA, decrement Register A (A-1 => A), **INCA, increment Register A (A+1 => A).**

Appendix C. G-CPU Code for **Problems 5 - 10:**

```
        ORG     0x0   ;this tells the assembler to start the program at addr zero
        LDAB    #0xFE
        STAB    0xFFFF
        LDAB    #100
        STAB    0xFFFE
        LDY     #0x0
        LDX     #0xC000
Top:    LDAB    3,Y
        LDAA    5,Y
        STAB    0,X
        STAA    1,X
        INY
        INX
        LDAB    0xFFFF
        LDAA    0xFFFE
        SUM_BA
        STAA    0xFFFE
        BNE     Top
End1:   BEQ     END1
```