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

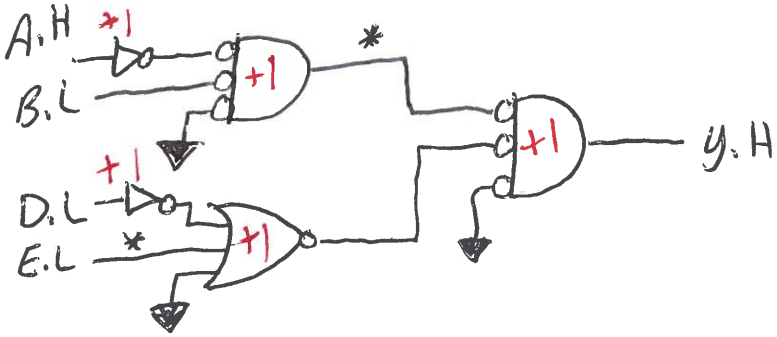
Page 1) 10 pt. Steph + Tomasz ① Marquez, Dan O. ② Page 2) 14 pt. 3A/3B Yikai 4 Lysny, Danny B

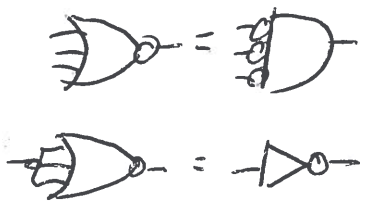
Page 3) 14 pt. ~~C~~ A, D ~~Daniel~~ ~~Kevin~~ Daniel C. Page 4) 13 pt. Su + Patrick #5 Alex B, Kevin, Tom TOTAL _____ (51)

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

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

Key





Ground error -0.5

Floating -1

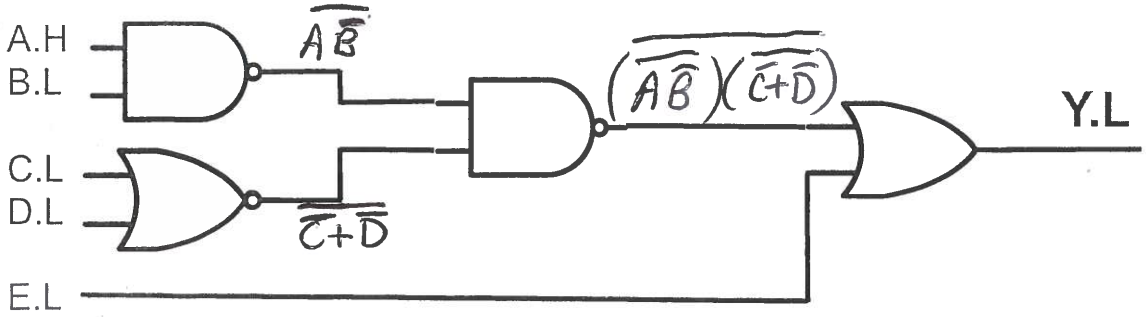
No Key -0.5

2 Input NORs -1

Missing inverter -1 ea.

Extra Inverter -0.5 or 1 depending on situation

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



+1 +1

$$Y.L = \underbrace{(\overline{AB})(\overline{C+D}) + \overline{E}}_{+1 \quad +1 \quad +1} = \overline{\overline{(\overline{A}+B)(CD)} + \overline{E}} = (\overline{A}+B)(CD) + E$$

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 **Show ALL VALUES IN HEX!**
Data (Hex) => F7 F1 F7 F2

Jmp 1 } memory
Jmp 2

Cycle	State	Input3:0	IR	PC	Reg. A3:0	MSA1:0	MSC2:0	
1 (reset)	0	7	0	0	0	01	XXX	Execution
2	1	7	7	0	0	01	XXX	Jmp 1
3	3	1	7	1	0	01	XXX	
4	0	1	7	1	0	01	XXX	Ldaa #7
5	1	1	1	1	0	01	XXX	
6	2	7	1	2	0	00	XXX	
7	0	2	1	3	7	01	XXX	SAR
8	1	2	2	3	7	11	110	
		+1.5	+1.5	+1.5	+1.5	+1	+1.5	+0.5

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

3 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"

IR2:0 → IR3:0 } +1
PC3:0 → PC7:0 }

+1 MAR or Temp reg to hold operand (address)

+1 Addr Mux to select between PC and MAR

+1 Tri-states (buffers) on output bus to connect to input bus

New control signals: R/W, MAR_LD, Addr_sel

Page Score =

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.)

```
-127 + 127 = 0   constant = 0x7F

+1  Ldaa #0;   ;clear SUM
    staa 0x4040
+1  Ldab #0x7F ; constant
+1  Ldy  #0x4080 ; input ptr

Top: Ldaa 0,y   ; last value?
+1   BEQ  DONE
+1   Sum_ba     ; = -127?
     BNE  SKIP
```

```
                    Ldaa 0x4080 ; SUM=
+1                  INCA        ; SUM+1
                    staa 0x4080
SKIP:  +1 INY
       +1 BNE Top

DONE:  BEQ DONE
```

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 C000-FFFF 16K x 8 (1 pt.)

6B. What is the -RAM_CE Logic Equation? -RAM_CE = A15*A14 (1 pt.)

or = A15*A14*D5.L

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 = $5 * 1/50 \times 10^6 = 10^{-7}$ **seconds** 0.1 μsec

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)
C000	FF +0.5
C001	+0.5 03 ← FF ; 2nd Pass
C002	64 +0.5
~~C003~~	+0.5
~~FFFE~~	60 ~~optional~~

+1

Page Score =

Best

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	0	0x12	OD	PC	Rom	OF	12	C001	0001	1
2	.5 1	0012	OD .5	PC	Rom	OD	12			1
3	1E	0013	05	PC	Rom	OD	13			1
4	1F	0006	64	Y BLK	Rom	OD	14			1
5	0	0014	12	PC	Rom	OD	14			1
6	.5 1	0014	12 .5	PC	Rom	12	14			1
7	28	0015	00	PC	Rom	12	15			1
8	29	C001	FF	X BLK	CPU	12	16	C001	0001	0
	+0 1	+2	+3	+1	+1	+1	+1	+0.5	+0.5	+1

7. ***Extra Credit.*** How many times does the loop run in **HEX**? (1 pt.) 32 **Hex**

+1 each 64

+1 each FF

0.5 each

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  0,1
        STAB    0xFFFF 2,3,4
        LDAB    #100   5,6
        STAB    0xFFFE 7,8,9
        LDY     #0x0   A,B,C
        LDX     #0xC000 D,E,F
Top:    LDAB    3,Y  } 1st pass y=0
  *     LDAA    5,Y  } 2nd pass =1
  *     STAB    0,X  } 1st pass X= C000
        STAA    1,X  }          = C001
        INY
        INX
        LDAB    0xFFFF
        LDAA    0xFFFE
        SUM_BA
        STAA    0xFFFE
        BNE     Top
End1:   BEQ     END1
```

	Addr	Data
LDAB	0	03
	1	FE
STAB	2	07
	3	FF
	4	FF
LDAB	5	03
	6	64
STAB	7	07
	8	FE
	9	FF
LDY	A	09
	B	00
	C	00
LDX	D	08
	E	00
	F	C0
LDAB	10	
	11	

FFFF [FE]
FFFE [64]

100 = 0x64

16*5=80
16*6=96

0110 0100
+ 1111 1110
0 1100010 62

2nd Pass → 60

Page Score =