EEL3701 – Dr. Gugel Last Name_____ First Name _____
Spring 2017
Final Quiz UF ID#_____

Open book/open notes, **90-minute** exam. ***No electronic devices permitted.***

Point System: Page 1 => 4 points _____ Page 2 => 10 points _____

Page 3 => 6 points _____ Page 4 => 10 points_____

Page 5 => 15 points _____ Page 6 => 5 points_____

TOTAL => _____ (50 pt.)

Grade Review Information: (*NOTE: deadline of request for grade review is the day the exam is returned*.)

_____

You are given a **1 KHz** clock for the following problem. Design a state machine that creates an output (**OUT.H**) based on three input variables: **A.L**, **B.L** and **C.L** ***Assume ONE and ONLY one of the three INPUT variables will be TRUE at any given time.***

If **A = T**, output a **500 Hz 50% Duty Cycle** square wave. Else If **B = T**, output a **333.3 Hz 33.3% Duty Cycle** square wave. Else if **C = T** output a **250 Hz 25% Duty Cycle** square wave. Recall: **Frequency = 1/Period** and **Duty Cycle = 100% * (Time High/Period).** Best Design = Most pts.

1. Draw the **Flow Chart** below for the State Machine. Don't forget to ***number your states!*** (4 pt.).

2. Draw the **Functional Block Diagram** for the system assuming that **logic gates** & **D flip-flops** will be used in the implementation. ***Label all signals***. (3 pt.)

3. Fill in the **Next State Table** below and use only the Present and Next States required in your design (7 pt.)

Note: You may assume that we have created a **reset circuit** to always **start** in the first state '**State 0'**.

A.L	B.L	C.L	Q3.H	Q2.H	Q1.H	Q0.H	D4.H	D3.H	D2.H	D1.H	D0.H	OUT.H

Page Score =

4. Create a **Logic Equation (SOP is OK!)** & **circuit** for **OUT.H** using inputs **A.L, B.L, C.L** and **Qx:Q0.** (3 pt.)

5. Suppose we desired OUT.H to run at twice it's current frequency, to do this we would need to create a clock that is twice the original 1KHz clock. Show how to ***create a 2 KHz clock from the original 1 KHz clock***.
Hint: Use the asynchronous clear and/or set mechanism in a D flip-flop. (3 pt.)

Page Score =

5. A vector is stored in 4Kx8 RAM at address **$1000.** It consists of **37_{10} signed numbers that are each two bytes in length**. Write a GCPU program to count the number of positive numbers in the vector. Write your ***final sum to the address immediately after the vector*** stored in RAM. Use **X as a pointer to the data** and **use address $1FFF for your loop counter**. (10 pt.)

Page Score =

For the next set of questions, consult the G-CPU code in **Appendix A.**

6. What is the effective address for the LDY #$0 instruction? _____ **Hex** (1 pt.)

7. What is the effective address for the STAA 0,X instruction the 1st time it is executed?

_____ **Hex** (1 pt.)

8. How many RAM memory locations are modified by this program? _____ **Decimal** (2 pt.)

9. What are the values in RAM when the program has completed? (2 pt.)

$1000 = _____ Hex $1001 = _____ Hex

10. In Appendix A, we are now using a ***SLOW ROM*** to fetch code. This ROM takes ***2 cycles per memory fetch***. i.e. ***2 clock periods instead of 1 to fetch a byte out of memory.*** Fill in the cycle table below for the ***1st loop pass*** for the lines of code shown below assuming that ***2 cycles instead of one will be required to fetch a memory byte out of ROM***. RAM still has 1 cycle R/W to/from memory. Label simply "**NEW**" for new states in the controller ASM below. (9 pt.)

OR_BA ;From Appendix A, show execution of the 1st pass of this code
STAA 0,X

Cycle	Controller State	Addr Bus	Data Bus	IR	R/-W	A Reg	Address Mux Sel1:0	Dev Outputting To Data Bus
1		000F						
2								
3								
4								
5								
6								
7								
8								
9								

Page Score =

11. Using only decoders, flip-flops, muxes, full adders and other elementary digital components, ***show the new hardware and modifications to the controller*** required to create a JMP ADDR instruction where ADDR is a 16 bit operand that corresponds to the jump address. Use **BUS labeling** when possible. (5 pt.)

Page Score =

Appendix A. G-CPU Code for **Problems 6 - 10:**

Note1: **8K ROM starting address $0** and **8K RAM starting at address $1000**.

Note2: **ROM** is a **slow device** that requires **TWO cycles** to **read** in a **value** from it.

```
        ORG     $0
        LDAB    #$F3
        STAB    $2FFF
        LDY     #$0
        LDX     #$1000
T1:     LDAA    0,Y
        LDAB    #$88
        OR_BA
        STAA    0,X
        INX
        INY
        LDAA    $2FFF
        LDAB    #1
        SUM_BA
        STAA    #2FFF
        BNE     T1
END1    BEQ     END1
```

Note: Use the space below to hand assemble instructions as needed in 6 - 10:

Page Score =