EEL3701 – Dr. Gugel Last Name_____ First Name _____
Spring 2015
Final Quiz UF ID#_____

Open book/open notes, **90-minute** exam to be done in **non-red** pen. ***No electronic devices permitted.***

Point System: Page 1 => 8 points _____ Page 2 => 12 points _____

Page 3 => 15 points _____ Page 4 => 15 points_____

TOTAL => _____ (50 pt.)

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

1. Implement the logic equation below using 2 Input NAND Gates only. (6 pt.)

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

2. The G GPU has been redesigned to initially boot from **8000 Hex**. i.e. Fetch it's first instruction from **8000H** in memory. Assuming that there is a **low true Reset** signal, explain what needs to be modified in the current design to facilitate booting from **8000 Hex**. (2 pt.) Best Answer = Most Points!

_____

_____

_____

3. A student has interfaced ***5K x 8 ROM*** with the G-CPU ***starting at address 8000 Hex***. The student also has interfaced and placed ***4K x 8 RAM in the highest 4K of the G-CPU memory map***. What is the memory ranges for the ROM and RAM devices? (2 pt.)

ROM Memory Range 8000 Hex to _____ (Hex)

RAM Memory Range _____ (Hex) to FFFF Hex

4. A vector stored in RAM consists of **signed numbers that are one byte in length**. Write the code to convert all the negative numbers to positive numbers. Assume **B contains the length** of the vector and **Y contains the starting address** of the vector. Also, if you need a counter in memory, use address **FFFF Hex**. (10 pt.)

Page Score =

For the next set of questions, consult the G-CPU code in **Appendix A** and assume the following initial conditions: **A = 5**, **X = $FF00**, Memory location ***$FF00 = $00***, **$FF01 = $11**, **$FF02 = $22 .... $FF08 = $88**.

5. What is the effective address for the STAB instruction at $8006? _____ Hex (2 pt.)

6. What is the effective address for the LDAB instruction at $8009? _____ Hex (2 pt.)

7. How many times is the LDAA 0,X at address $801C executed at run time? _____ Hex (2 pt.)

8. What values in SRAM memory below when the PC is fetching $8028 (Program is DONE)? (3 pt.)

$FF00 = _____ Hex $FF01 = _____ Hex $FF02 = _____ Hex

$FF03 = _____ Hex $FF04 = _____ Hex $FF05 = _____ Hex

9. What is the function of this short program? _____

_____ (2 pt,)

10. Using only 2:1 decoders, flip-flops, 2:1 muxes, single bit full adders and any other elementary digital components, design the hardware required to implement the SUM_AY instruction. Label all buses and explain any new signals you'll need in the ALU, Controller, IR and/or Address Mux. Best design = Most points. (4 pt.)

11. For the G CPU code shown in Appendix A, fill in the cycle table below for the ***first pass*** for the lines of code shown below. ***Use the initial conditions shown on the previous page and show all answers in Hex.*** (9 pt.)

```
LDAB    $FFFE    ; first pass of this instruction
DECA             ; assume this takes 2 cycles to execute
```

Cycle	Controller State	Addr Bus	Data Bus	IR	R/-W	B Reg	Dev driving Addr Bus	Dev Driving Data Bus
1								
2								
3								
4								
5								
6								
7								

12. The G CPU Next State Table and Controller can be found in **Appendix B**. The Controller was created via an **8K x 32 ROM**. For the very ***first*** and ***last*** cycles of the **LDAB $FFFE** instruction above, show the addresses and corresponding data that **need to be programmed in the ROM**. (4 pt.)

First Cycle Executed **(Upper 3 bytes Only)**

ROM Addresses = _____ Hex Data = _____ Hex

Last Cycle Executed **(Lower 3 bytes Only)**

ROM Addresses = _____ Hex Data = _____ Hex

14. For the new **SUM_AY** instruction in **Appendix A**, show below what needs to be **added** to the Controller **ASM Flow Chart** assuming that this instruction will **follow the INY instruction** in the ASM Chart. (2 pt.)

Appendix A. G-CPU Code (address of instruction in memory shown on left):

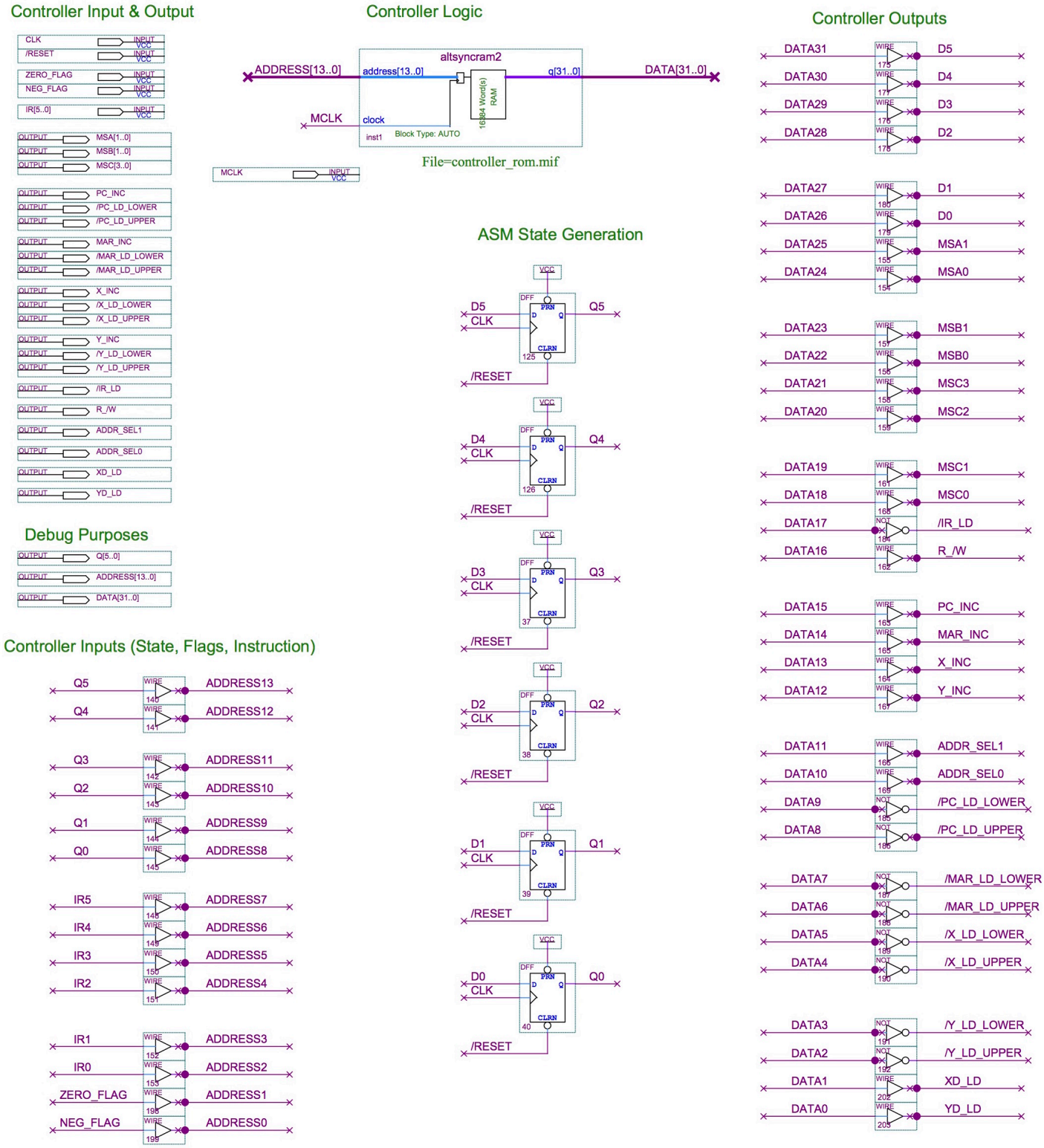
Register ***A = Vector Length***, Register ***X = Starting Address of Vector***

```
                ORG     $8000
8000    Compute: TXY                    ;new instruction transfer X => Y
8001            SUM_AY                  ;new instruction A + Y => Y
8002            DEY                     ;new instruction Decrement Y
8003            SHFA_R
8004            TAB
8005            COMB
8006            STAB    $FFFE
8009            LDAB    #1
800B            SUM_BA
800C            STAA    $FFFF

800F    Loop:   LDAA    $FFFF
8012            LDAB    $FFFE
8015            DECA                    ;new instruction Decrement A
8016            TAB
8017            BEQ     Done
8019            STAA    $FFFF
801C            LDAA    0,X
801E            LDAB    0,Y
8020            STAA    0,Y
8022            STAB    0,X
8024            INX
8025            DEY                     ;new instruction, Decrement Y
8026            BRA     Loop            ;new instruction, unconditional branch
8028    Done:   BRA     Done            ;new instruction, unconditional branch
```


Page Score =

Appendix B. G-CPU Controller & Next State Table:



Page Score =

G-CPU Controller Next State Table

Pres State	Opcode	Flags	Next State	Mux Select			Control		REG INC	ADDR SEL	PC	MAR	X,Y Loading		Disp Regs	
Q[5..0]	IR[5..0]	Z N	D[5..0]	MSA [1..0]	MSB [1..0]	MSC [3..0]	IR LD	R /W	PC MAR X Y	ADDR SEL [1..0]	PC LD L/U	MAR LD L/U	X LD L/U	Y LD L/U	XD_LD YD_LD	Present State Function
000000	XXXXXX	XX	000001	01	10	0000	1	1	0000	00	00	00	00	00	00	generic instruction fetch
000001	000000	XX	000000	01	01	0000	0	1	1000	00	00	00	00	00	00	Transfer A to B (TAB)
000001	000001	XX	000000	10	10	0000	0	1	1000	00	00	00	00	00	00	Transfer B to A (TBA)
000001	000010	XX	000010	01	10	0000	0	1	1000	00	00	00	00	00	00	LDAA #data, state 1
000010	XXXXXX	XX	000000	00	10	0000	0	1	1000	00	00	00	00	00	00	LDAA #data, state 2
000001	000011	XX	000011	01	10	0000	0	1	1000	00	00	00	00	00	00	LDAB #data, state 1
000011	XXXXXX	XX	000000	01	00	0001	0	1	1000	00	00	00	00	00	00	LDAB #data, state 3
000001	000100	XX	000100	01	10	0000	0	1	1000	00	00	00	00	00	00	LDAA addr, state 1
000100	XXXXXX	XX	000101	01	10	0000	0	1	1000	00	00	10	00	00	00	LDAA addr, state 4
000101	XXXXXX	XX	000110	01	10	0000	0	1	1000	00	00	01	00	00	00	LDAA addr, state 5
000110	XXXXXX	XX	000000	00	10	0000	0	1	0000	01	00	00	00	00	00	LDAA addr, state 6
000001	000101	XX	000111	01	10	0000	0	1	1000	00	00	00	00	00	00	LDAB addr, state 1
000111	XXXXXX	XX	001000	01	10	0000	0	1	1000	00	00	10	00	00	00	LDAB addr, state 7
001000	XXXXXX	XX	001001	01	10	0000	0	1	1000	00	00	01	00	00	00	LDAB addr, state 8
001001	XXXXXX	XX	000000	01	00	0000	0	1	0000	01	00	00	00	00	00	LDAB addr, state 9
000001	000110	XX	001010	01	10	0000	0	1	1000	00	00	00	00	00	00	STAA addr, state 1
001010	XXXXXX	XX	001011	01	10	0000	0	1	1000	00	00	10	00	00	00	STAA addr, state A
001011	XXXXXX	XX	001100	01	10	0000	0	1	1000	00	00	01	00	00	00	STAA addr, state B
001100	XXXXXX	XX	000000	01	10	0000	0	0	0000	01	00	00	00	00	00	STAA addr, state C
000001	000111	XX	001101	01	10	0000	0	1	1000	00	00	00	00	00	00	STAB addr, state 1
001101	XXXXXX	XX	001110	01	10	0000	0	1	1000	00	00	10	00	00	00	STAB addr, state D
001110	XXXXXX	XX	001111	01	10	0000	0	1	1000	00	00	01	00	00	00	STAB addr, state E
001111	XXXXXX	XX	000000	01	10	0001	0	0	0000	01	00	00	00	00	00	STAB addr, state F
000001	001000	XX	010000	01	10	0000	0	1	1000	00	00	00	00	00	00	LDX #data, state 1
010000	XXXXXX	XX	010001	01	10	0000	0	1	1000	00	00	00	10	00	00	LDX #data, state 10
010001	XXXXXX	XX	000000	01	10	0000	0	1	1000	00	00	00	01	00	00	LDX #data, state 11
000001	001001	XX	010010	01	10	0000	0	1	1000	00	00	00	00	00	00	LDY #data, state 1
010010	XXXXXX	XX	010011	01	10	0000	0	1	1000	00	00	00	00	10	00	LDY #data, state 12
010011	XXXXXX	XX	000000	01	10	0000	0	1	1000	00	00	00	00	01	00	LDY #data, state 13
000001	001010	XX	010100	01	10	0000	0	1	1000	00	00	00	00	00	00	LDX addr, state 1
010100	XXXXXX	XX	010101	01	10	0000	0	1	1000	00	00	10	00	00	00	LDX addr, state 14
010101	XXXXXX	XX	010110	01	10	0000	0	1	1000	00	00	01	00	00	00	LDX addr, state 15
010110	XXXXXX	XX	010111	01	10	0000	0	1	0100	01	00	00	10	00	00	LDX addr, state 16
010111	XXXXXX	XX	000000	01	10	0000	0	1	0000	01	00	00	01	00	00	LDX addr, state 17
000001	001011	XX	011000	01	10	0000	0	1	1000	00	00	00	00	00	00	LDY addr, state 1

Page Score =