Exam 2

May the Schwartz be with you!

Last Name, First Name

UF's NaviGator AMS team.

Please read carefully.

Instructions:

- *Turn off all **cell phones** and other **noise making devices** and put away **all electronics**.*
- *Show **all** work on the **front** of the test papers. **Box** each answer. If you need more room, make a clearly indicated note on the front of the page, "**MORE ON BACK**", and use the back. The back of the page will **not** be graded without an indication on the front.*
- *This exam counts for 20% of your total grade.*
- *Read each question **carefully** and **follow the instructions**.*
- *You may **not** use any notes, HW, labs, other books, or calculators.*
- *The point values for problems may be changed at prof's discretion.*
- *You must pledge and sign this page in order for a grade to be assigned.*
- ***CLEARLY** write your name at the top of **this** test page (and, if you remove the staple, all others). Be sure your exam consists of **11** distinct pages. Sign your name and add the date below. (If we struggle to read your name, you will lose points.)*
- *Failure to follow the below rules will result in **NO** partial credit*
  - *The **base** (radix) of all number should be indicated with a **subscript** or **prefix**.*
  - *Truth tables, voltage tables, and timing simulations must be in **counting** order.*
  - *Label the inputs and outputs of each circuit with activation-levels.*
  - *For each mixed-logic circuit diagram, label inputs of **each** gate with the appropriate logic equations.*
  - *For K-maps, label **each** grouping with the appropriate equation.*
  - *Labels inside of parts must be provided whenever it can be confusing, e.g., they **MUST** be specified for MUXes and Decoders, but not for NANDs and ORs.*
  - *For each circuit design, equations must **not** be used as replacements for circuit elements.*
  - *Boolean expression answers must be in **lexical order**, (i.e., /A before A, A before B, & D_3 before D_2).*

Good luck & Go Gators!!!

PLEDGE: On my honor as a University of Florida student, I certify that I have neither given nor received any aid on this examination, nor I have seen anyone else do so.

SIGN YOUR NAME _____ DATE (9 November 2016) _____

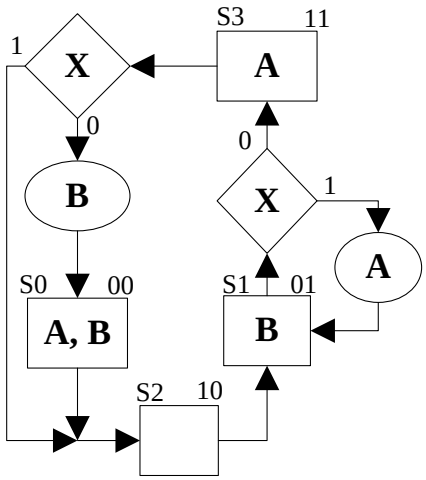
Regrade comments below: Give page # **and** problem # and reason for the petition.
.
.
.
.
.
.
.
.

Page	Available	Points
2-3	17	
4	14	
5-6	22	
7	15	
8-10	20	
11	12	
TOTAL	100	

Exam 2

Last Name, First Name

[17%] 3 min 1. In this problem you will ultimately design a circuit to implement this ASM. Use a **T-FF** for the **least significant state bit**, a **JK-FF** for the **most significant state bit**, and **D-FF**'s for any **other** FF's that you might need. The signals **X** and **B** are **active-low** and **A** is **active-high**. An active-high signal **Start** should **asynchronously** move the machine to state **S2**.



(7%) 7 min a) Complete the next-state **truth** table. You may **not** need all the rows and/or columns. Use wild cards (*) where possible.

(3%) 5 min b) Find the required VHDL equations (**no need** to simplify) **for determining B and the JK-FF inputs**. (You do **NOT** have to find the other equations.)

Exam 2

Last Name, First Name

(5%) 3 min 1. c) Design the complete mixed-logic circuit diagram, using a JK-FF and T-FF (and D-FF(s), if necessary) as described previously. Use a block diagram for combinatorial logic (like you did in Lab 5 when you created a graphical symbol in Quartus); label all inputs and outputs as Quartus would when a graphical symbol is created. All **inputs** and **outputs** of your circuit design should be **clearly indicated <u>coming into or out of</u>** the below box. Your design must include the circuitry necessary to **asynchronously** cause the system to go to state **S2** when the **Start** (active-high) signal goes true.

5 min **<u>Inputs</u>** **<u>Outputs</u>**

(2%) 2 min d) What changes are necessary if X and B are active-high instead of active-low? Describe any changes necessary in part b and write any new required VHDL below.

Exam 2

Last Name, First Name

[6%] 2. Answer the following questions about a state machine designed with one EEPROM, one J-K flip-flop, one T flip-flop, and any other necessary flip-flops of type D.

(3%) 3 min a) What is the size of the EEPROM (addresses × data bits) if the state machine has 3 inputs, 4 outputs, and 5 states? Provide numbers only, i.e, 37×9, **NOT** expressions like $3^7 \times \sqrt[4]{37}$.

(3%) 3 min b) What is the size of the EEPROM (addresses × data bits) if the state machine has 5 inputs, 4 outputs, and 13 states? Provide numbers only, i.e, 37×9, **NOT** expressions like $3^7 \times \sqrt[4]{37}$.

[4%] 5 min 3. Draw a complete **mixed-logic circuit diagram** to generate the below equation. **The circuit must include three switch circuits**, one for an **active-high** input signal **W(H),** one for an **active-low** input **I(L)**, and one other for input **N**.

$$Cubs = \overline{W + I * \bar{N}}$$

Use the **minimum number of gates** on only a **single** SSI **chip** (i.e., a single 74HCxxx). Draw the switches in their **true** positions. Draw an **LED circuit** for the **active-high** output, **Cubs(H)**. Do **NOT** draw a layout.

[4%] 4. Two of the four **modes** in VHDL are called `in` and `out`.

(2%) 1 min a) What does the mode `inout` mean, and for what type of device is it used?

(2%) 1 min b) What does the mode `buffer` mean, and for what type of device is it used?

Exam 2

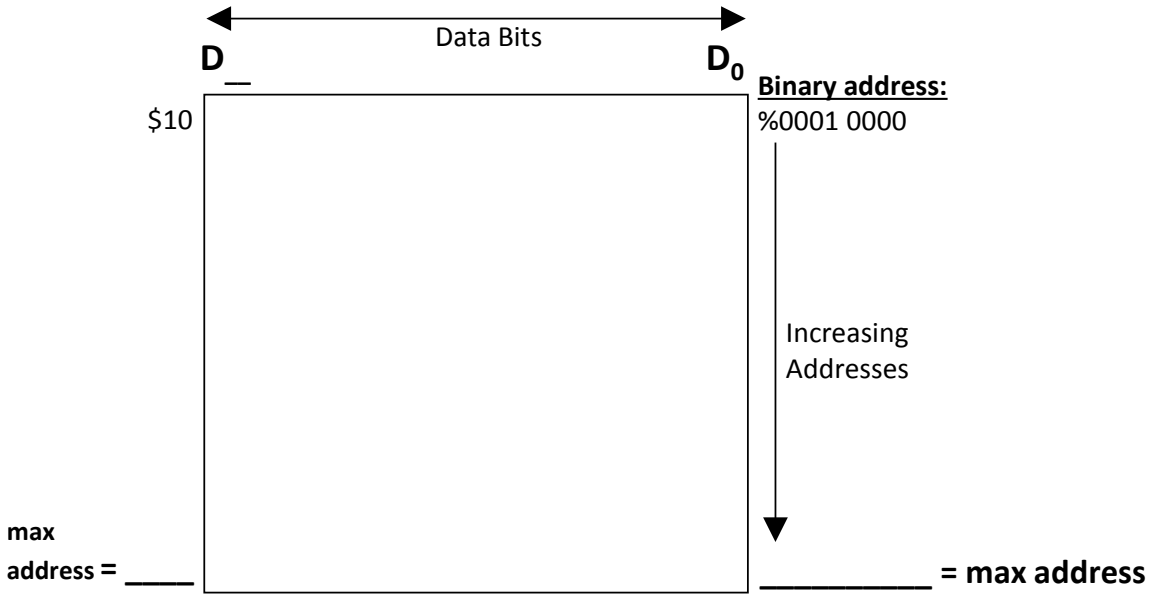
Last Name, First Name

[22%] 5. Given as many of each of the following as needed, design the memory module described below, with an **active-low** chip enable, **CE(L)**: **32×4 SRAMs**, **16×8 EEPROMs**, and **16×4 FLASH EEPROMs**, The memory module should begin at address **$10 = 10_{16}** with **32x4** of **Flash** at the **same addresses** as **32x4 SRAM** followed immediately by **16×8** of **EEPROM**. Add the **minimum number** of memory devices required. Addresses outside of those required (but within **$0** to **$FF**) must be **unaffected**.

3 min

(%) a) Draw vertical and horizontal lines in the box below and label each resulting box with the memory type and size, using only the defined types and sizes given above. Also, fill in the subscript on the D (top left) and the maximum address (bottom left and right).

6 min



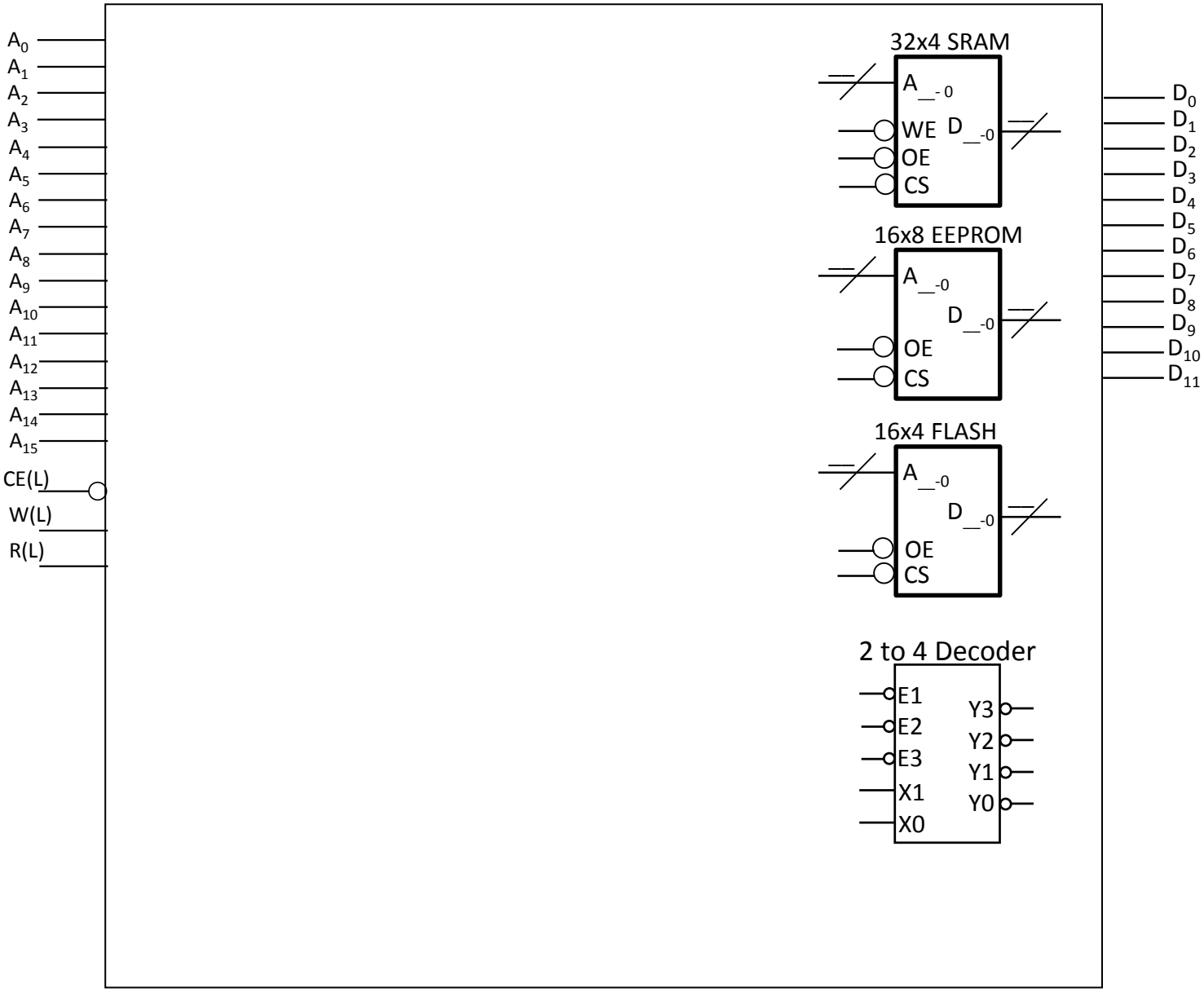
(%) b) What are the address and data ranges for **each of the** memory components drawn above (in **binary** and in **hex**)?

5 min

Exam 2

Last Name, First Name

(%) 5. c) 8 min Design the required memory device circuit diagram below. Use the single decoder shown (bottom right) and add the **minimum number** of **additional** memory components and SSI **gates** necessary (but **no** other MSI gates, LSI gates, or PLDs). **Add** memory components and address and data subscripts as needed; **cross out** unneeded address and data pins. **Use labels instead of wires for the design**. **Also**, **write the equations** for each CS at the bottom of the page. Show **all connections** with either labels or wires (**labels are preferred**), just as in Quartus. Don't forget the system's **CE(L), R(L), and W(L)**. Note: R(L) and W(L) are never true at the same time.



Equations:

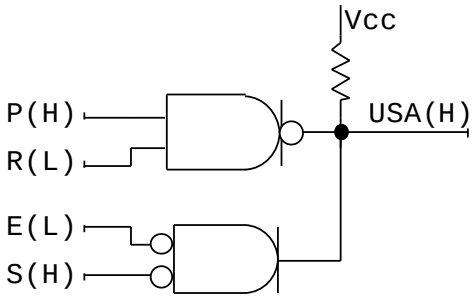
Exam 2

Last Name, First Name

[6%] 6. What are the MSOP or MPOS equations for the following circuits? Please look at each circuit carefully.

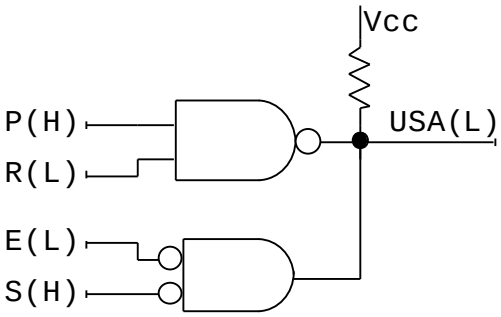
(%)
4 min

a)



(%)
4 min

b)



[9%] 7. Design an **8-input MUX with a tri-state enable** using only 2-input MUXes with single tri-state enables and a **one more single** device. This other device can be any **one** of the following MSI devices: DeMUX, Decoder, Encoder, Open-Collector Gate, or Tri-State Buffer. Draw a functional block diagram and then a circuit diagram. (If you can **not** solve the problem as stated, i.e., with **one** additional MSI device, then use SSI gates in addition to the 2-input MUXes.)

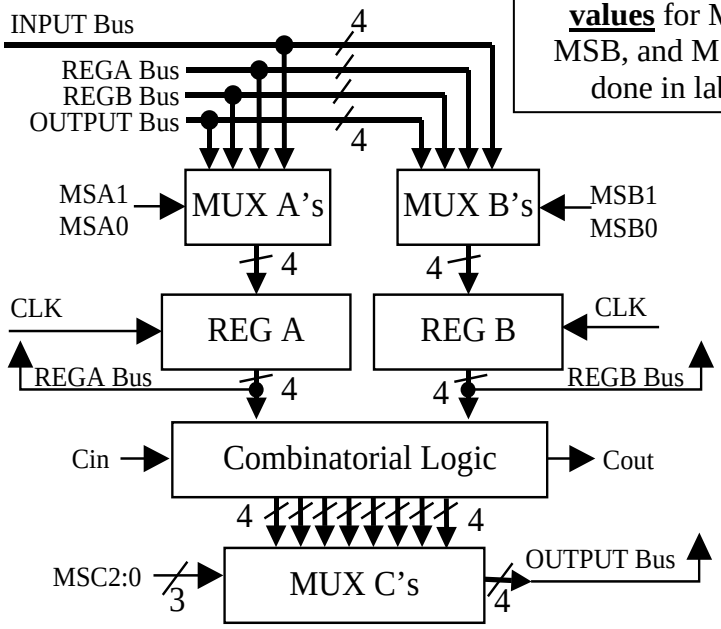
10 min

Exam 2

Last Name, First Name

[20%] 8. A block diagram and tables from your Lab 4 are shown below.

3 min



Use **proper default values** for MSA, MSB, and MSC, as done in lab 4.

MSA1:0/ MSB1:0	Bus Selected as Input to Combinatorial Logic
00	INPUT Bus
01	REG A Bus
10	REG B Bus
11	Output Bus

MSC	Action
000	REGA Bus to OUTPUT Bus
001	REGB Bus to OUTPUT Bus
010	complement of REGA Bus to OUTPUT Bus
011	bit wise AND REGA/REGB to OUTPUT Bus
100	bit wise OR REGA/REGB Bus to OUTPUT Bus
101	sum of REGA Bus & REGB Bus to OUTPUT Bus
110	shift REGA Bus left one bit to OUTPUT Bus
111	shift REGA Bus right one bit to OUTPUT Bus **without** sign extension

(4%) 5 min

a) Assume that you can add a small additional circuits to determine when REGA is equal to zero and another small circuit to determine when REGB is (2's complement) negative. Design a circuit for each of these active-high outputs (**Aeq0** and **Bneg, respectively**). Show **ALL** work.

Exam 2

Last Name, First Name

(5%) 8. c) Calculate **$D + $C*5**, putting the result in **RegA** (where + is addition and * is multiplication). Describe your algorithm (plan) and then what is accomplished in each step. Use the minimum number of clock cycles. Give appropriate values for **ALL** signals. (Your algorithm should still work if all of the numbers are changed.)

5 min

#	MSA	MSB	MSC	Input	Cin	RegA	RegB	Output	Cout	RegA+	RegB+	Output+	Cout+
1						1011	1111						
2													
3													
4													
5													
6													

Plan: _____

_____

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____

(5%) 8. b) Subtract 7 from 3, i.e., X = 3 - 7; put the resulting difference in RegB. Describe your algorithm (plan) and then what is accomplished in each step. Use the minimum number of clock cycles. Give appropriate values for **ALL** signals. (Your algorithm should still work even if the two numbers, i.e., the 7 and the 3 are changed.)

5 min

#	MSA	MSB	MSC	Input	Cin	RegA	RegB	Output	Cout	RegA+	RegB+	Output+	Cout+
1						0101	1010						
2													
3													
4													
5													
6													

Plan: _____

_____

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____

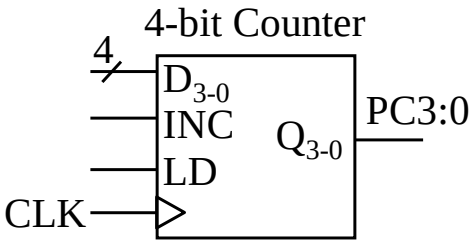
Exam 2

Last Name, First Name

(6%) 7 min

8. d) When discussing Lab 6 in class, I have shown that we create assembly language instructions by adding a controller to the Lab 4 hardware. A JMP instruction (e.g., **`JMP Addr3:0`**) causes the processor to jump to the specified address (Addr3:0). All microprocessors have instructions (often known as branch instructions) that ask a question that will result in either a **jump to another address** or **continuation to the next address**. A "branch if RegA=0" instruction, (i.e., **`BrAeq0 Addr3:0`**) will cause the CPU to jump to address **`Addr3:0`** if **RegA = 0** and to increment the program counter (i.e., PC3:0 ← PC3:0 + 1) if **RegA ≠ 0**.

Design a circuit, using the below 4-bit counter, to either load the counter with a 4-bit address or increment the counter, depending on the value of **Aeq0** (determined in part a).



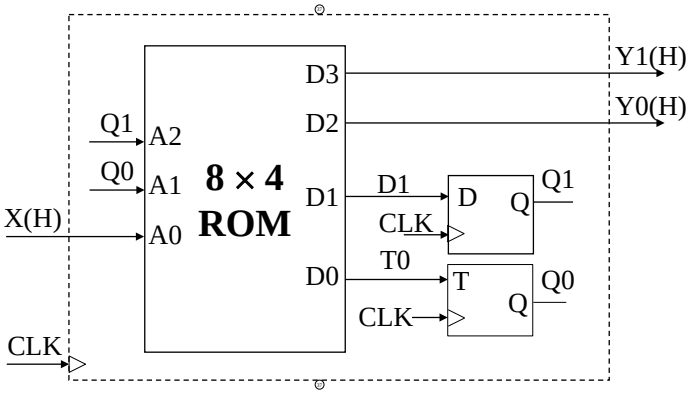
Exam 2

Last Name, First Name

[12%] 9. Use the below circuit diagram and the **ROM contents table** below to solve this problem. Note that the addresses and data in the table are in **hexadecimal** [base 16]. (This problem is very similar to a problem in Homework 8, which was also solved in class.) Note that a D-FF and a T-FF are utilized in the below circuit.

14 min

Derive the ASM chart for this circuit. Show **ALL** work, i.e., use at least part of the below table. Draw your ASM in the space provided on this page.



				Contents of ROM (Hex)											
				Addr	Data										
				0	E										
				1	9										
				2	3										
				3	8										
				4	F										
				5	F										
				6	5										
				7	5										