Exam 2

Last Name, First Name

May the Schwartz be with you!

UF's potential entry for 2016 Maritime RobotX.

Please read carefully.

Good luck & Go Gators!!!

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.*
- *Put 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.*
- *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, & D3 before D2).*

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 (12 November 2015) _____

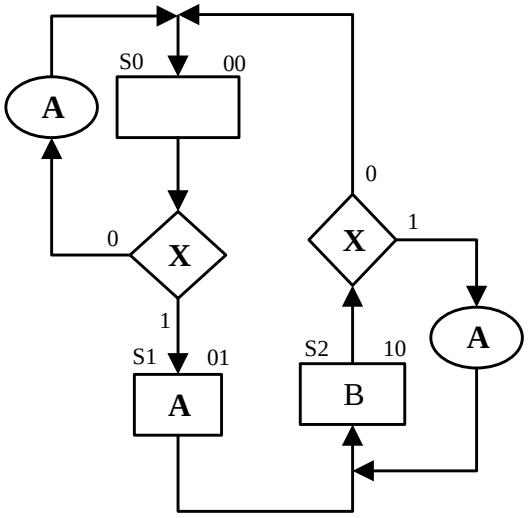
Regrade comments below: Give page # **and** problem # and reason for the petition.
.
.
.
.
.
.
.
.
.

Page	Available	Points
2-3	15	
4	18	
5-6	22	
7-8	20	
9-11	25	
TOTAL	100	

Exam 2

Last Name, First Name

[15%] 1. In this problem you will ultimately design a circuit to implement this ASM. Use only SSI gates, a **T-FF** for the **most significant state bit**, a **JK-FF** for the **least significant state bit**, and **D-FF**'s for any **other** FF's that you might need. The signals **X** and **B** are **active- high** and **A** is **active-low**. An active-low signal **Go** should **asynchronously** move the machine to state **S2**.



3 min

(%) a) Complete the next-state **truth** table. You may **not** need all the rows and/or columns.

5 min

(%) b) Find the required **simplified** (MSOP or MPOS) equations **for determining the next state** and **for the outputs**.

5 min

Exam 2

Last Name, First Name

(%) 1. c) Design the complete mixed-logic circuit diagram, **minimizing** the total number of components, but using the T-FF and JK-FF (and D-FF(s), if necessary) as described previously. All **inputs** and **outputs** of the circuit should be **clearly indicated coming into or out of** the below box. Your design must include the circuitry necessary to **asynchronously** cause the system to go to state S2 when the **Go** (active-low) signal goes true. For debugging purposes, **also output the active-low** versions of the **state bits**.

5 min

Inputs

Outputs

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 2 inputs, 7 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, 2 outputs, and 9 states? Provide numbers only, i.e, 37×9, **NOT** expressions like $3^7 \times \sqrt[4]{37}$.

[4%] 4 min 3. Draw a complete **mixed-logic circuit diagram including two switch circuits**, one for an **active-high** input signal, **U(H)** and one for an **active-low** input, **F(L)**, to generate the equation $\boldsymbol{Go = \overline{U * \overline{\overline{F}}}}$. Draw the switches in their **true** positions. Draw an **LED circuit** for the **active-high** output, **Go(H)**. Do **NOT** draw a layout.

[8%] 4. Answer the following short questions.

(2%) 1 min a) What is VHDL?

(2%) 1 min b) In VHDL, what is an entity?

(2%) 2 min c) If Q1 and Q0 represents state bits, and X, Y, and Z are inputs, write possible **VHDL** equations (**NOT** simplified) for a Moore output (**Mo**) and a Mealy output (**Me**). Use proper VHDL syntax.

Mo

Me

(2%) 2 min d) What is a "Top-Level Entity" in Quartus?

Exam 2

Last Name, First Name

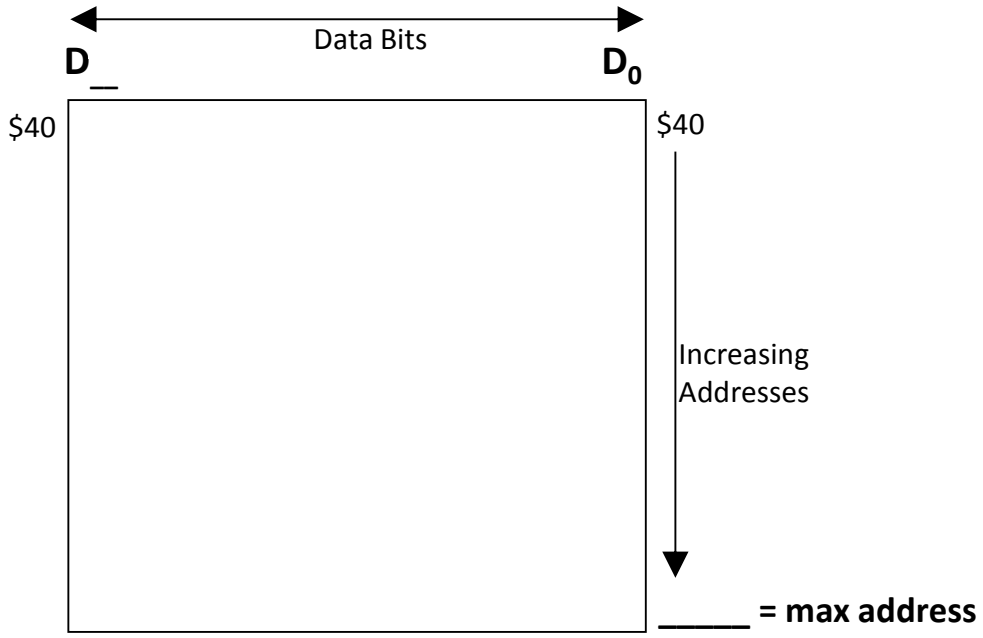
[22%] 5. Given as many of each of the following as needed, design the memory module described below, with an **active-low** chip enable, **CE(L)**: **16×4 SRAMs**, **32×4 EEPROMs**, and **32×8 FLASH EEPROMs**, The device should begin at address **$40 = 40_{16}** with **32x4** of **SRAM** at the **same addresses** as **32x4 EEPROM**, followed immediately by **32×8** of **FLASH**. Add the **minimum number** of memory devices and the **minimum number** of additional **SSI** components required, i.e., other than the memory, **no** other MSI or LSI components are allowed.)

3 min

(%)

5 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 at the top left and the maximum address at the bottom right.



(%)

6 min

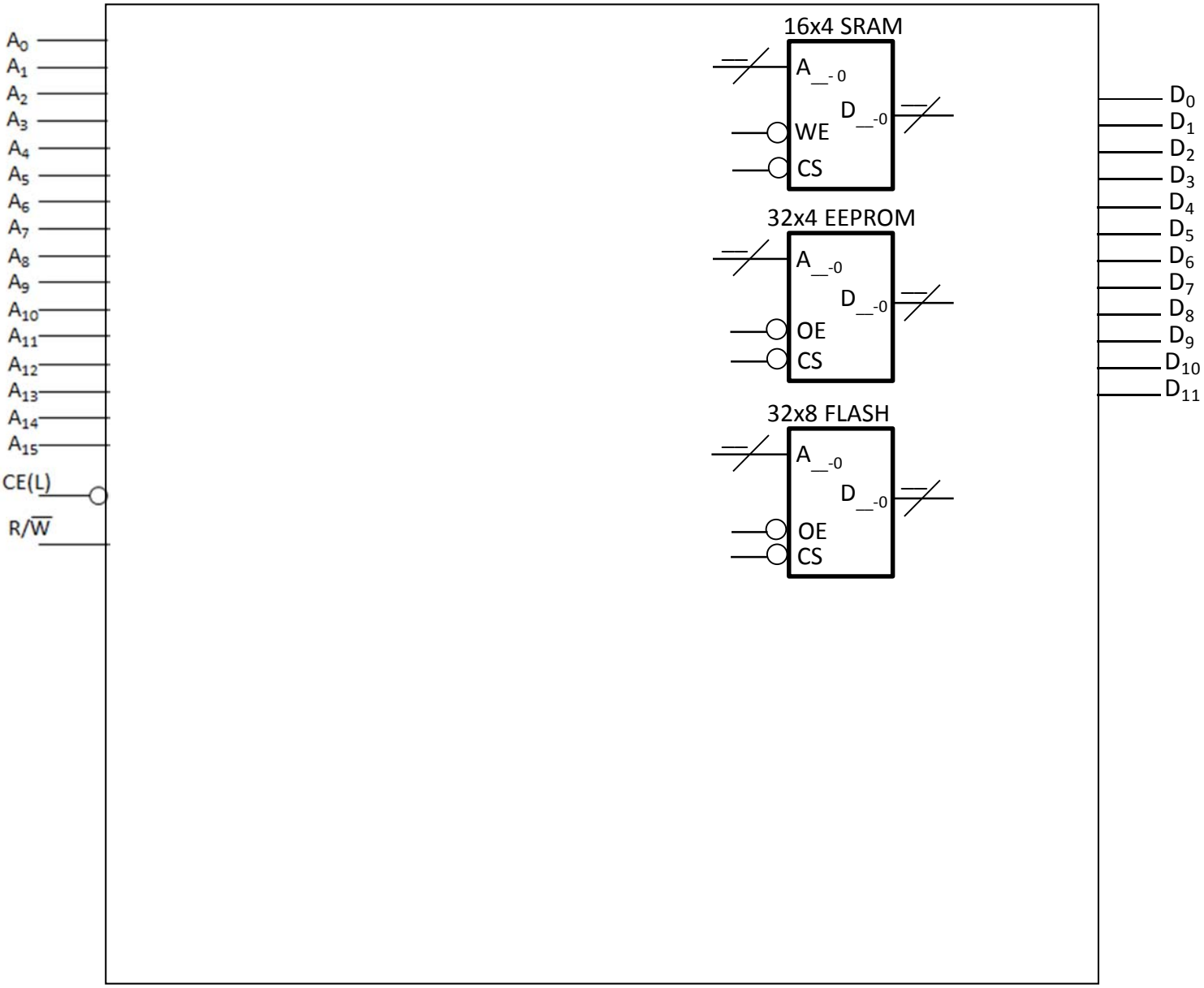
b) What are the address and data ranges for **each** **of the** memory components drawn above (in **binary** and in **hex**)?

Exam 2

Last Name, First Name

(%) 5. c) Design the required memory device circuit diagram below. Add the **minimum number** of **additional** memory components and SSI **gates** necessary (**no** 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 **active-low** chip enable, **CE,** and **R/~W** [pronounced "read, write not," with R(H) and W(L)].

8 min



Equations:

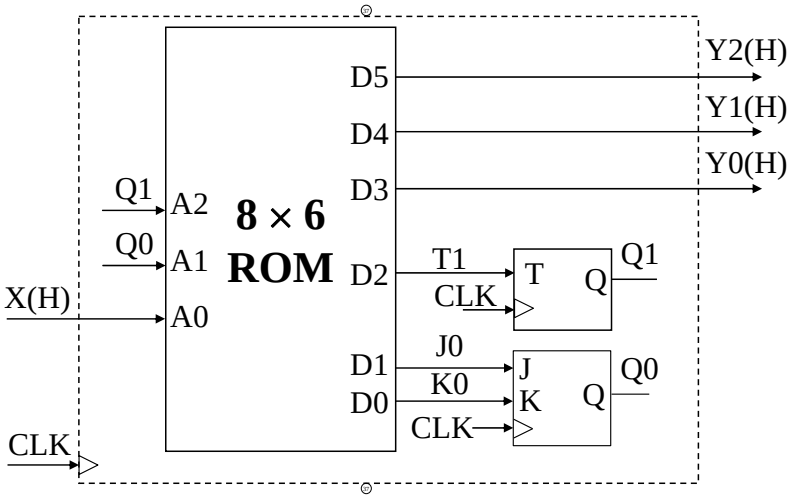
Exam 2

Last Name, First Name

[20%] 6. 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 that was also done in class.) Note that a T-FF and a JK-FF are utilized in the below circuit.

(17%) 17 min

a) **Derive** the ASM chart for this circuit. Show **ALL** work, i.e., use at least part of the below table. You can draw your ASM on this page or on the next page.



				Contents of ROM (Hex)												
				Addr	**Data**											
				0	**14**											
				1	**14**											
				2	**26**											
				3	**26**											
				4	**0A**											
				5	**2A**											
				6	**12**											
				7	**02**											

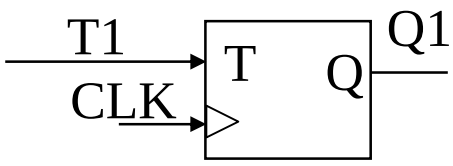
Exam 2

Last Name, First Name

(17%) 6. a) (**Repeated** from previous page.) **Derive** the ASM chart for this circuit. Show **ALL** work, i.e., use at least part of the above blank table. You can draw your ASM on this page or on the previous page.

(3%) 3 min

b) If the D2 pin of the ROM is broken, determine the necessary equation and circuit to input to the T-FF, i.e., create the proper input circuit for the T-FF.

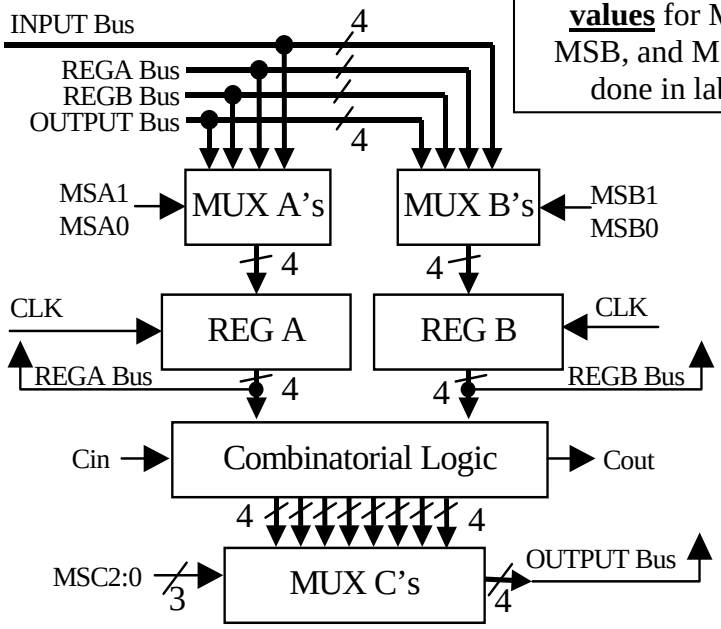


Exam 2

Last Name, First Name

[25%] 7. A block diagram of your lab 6 is shown here, along with two tables from the same handout.

3 min



Use **proper default values** for MSA, MSB, and MSC, as done in lab 6.

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

(6%) 6 min

a) Assume that you can add two small additional circuits to the circuit described above. The purpose of these additional circuits is (1) to know when there would be an **overflow** when addition is used (output is **V**) and (2) to know when RegA is even (output is **A_Even**). Hint: Zero is even. Show **ALL** work.

Exam 2

Last Name, First Name

(5%) 7. b) Get the average of the values in RegA and the number 3; put the resulting average in RegB. Use the values presently in RegA, i.e., 7. 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 in A and the number 3 are changed.)

5 min

#	MSA	MSB	MSC	Input	Cin	RegA	RegB	Output	Cout	RegA+	RegB+	Output+	Cout+
1						0111	0000						
2													
3													
4													
5													
6													

Plan: _____

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

(6%) c) Multiply 3 by 5. (Your algorithm should still work even if the **number 3** is changed, but **not** if the number 5 is changed.) Store the result in **RegA**. 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.

5 min

#	MSA	MSB	MSC	Input	Cin	RegA	RegB	Output	Cout	RegA+	RegB+	Output+	Cout+
1						1010	1011						
2													
3													
4													
5													
6													

Plan: _____

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

Exam 2

Last Name, First Name

(8%) 7 min

7. d) With the present design, you **cannot** add multiple nibbles (e.g., $37 + $5B) because there is no way to remember the carry output between the nibbles. You will fix that issue in this problem. Design a circuit that can be added the existing circuit to "remember" the last carry output (**Cout**) any time addition has occurred. Call the remembered value **R_Cout**. Also design a circuit so that the remembered carry output, **R_Cout**, can be used as the carry input (**Cin**), when a new input **LongAdd** is true, but use the regular **Switch_Cin** switch for the carry input (**Cin**) when **LongAdd** is false.