# Exam 2

_____
Last Name, First Name

*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*
  - *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.*
  - *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 $D_2$).*


***PropaGator 2***: UF MIL's potential 2nd robot boat.

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 (6 November 2013)

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

| Page | Available | Points |
|---|---|---|
| 2-3 | 15 | |
| 4 | 17 | |
| 5-6 | 22 | |
| 7-8 | 20 | |
| 9-11 | 26 | |
| TOTAL | 100 | |

# Exam 2

Last Name, First Name

[15%] 1. Design a system that sequences through the following outputs: $C_{16}$, $8_{16}$, $7_{16}$, $C_{16}$, $8_{16}$, $7_{16}$, etc., with each of **these output bits active-low**. The system must **asynchronously** reset to output the "**$C_{16}$**" when **Start** (**active-low**) goes true. When the sequence output is **$8_{16}$**, the **active-high** output **Z** should be true. Use a T-FF for the **most** significant bit of the design, a JK-FF for the **least** significant bit, and a D-FF for **any other** bits you might need. Note: All the given FFs have **active-low asynchronous** clear and set inputs. Use the **minimum number of flip-flops** and **then** try to **minimize the number** of SSI gates necessary to solve this problem. (Other than the flip-flops, no MSI or LSI gates, PLDs, or ROMs allowed.)

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.

5 min

# Exam 2

Last Name, First Name

( %) 1. c) 5 min 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 **asynchronous** re-start the system at output "$\mathbf{C_{16}}$" when the **Start** (active-low) signal goes true and show the active-high output **Z** (which is true when the output is $\mathbf{8_{16}}$).

**Inputs** **Outputs**

# Exam 2

Last Name, First Name

[4%] 4 min

2. Draw a complete mixed-logic circuit diagram including two switch circuits, one for an **active-low** input signal, **X(L)** and one for an **active-high** input, **Y(H)**. to generate the equation W = X * /Y, with **W(L)**. Draw the switches in their **true** positions. Draw an LED circuit for the active-low output, **W(L)**. Do **NOT** draw a layout.

[6%]

3. Answer the following questions about a state machine designed with one EEPROM, one T flip-flop, one J-K 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, 2 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 1 input, 3 output, and 13 states? Provide numbers only, i.e, 37×9, **NOT** expressions like $3^7 \times \sqrt[4]{37}$.

[3%] 2 min

4. Write the complete **VHDL** equation (one line) for the following equation. Do **NOT** simplify.

$$G = A + \bar{T} * \overline{\overline{O * R} + S}$$

[4%]

5. Answer the following questions.

(2%) 1 min

a) When is a Mealy output required? _____

(2%) 2 min

b) As alternative to using a Mealy part 5a, describe (if possible) how to use a Moore output to accomplish the required task. What is the major disadvantage of this solution?

# Exam 2

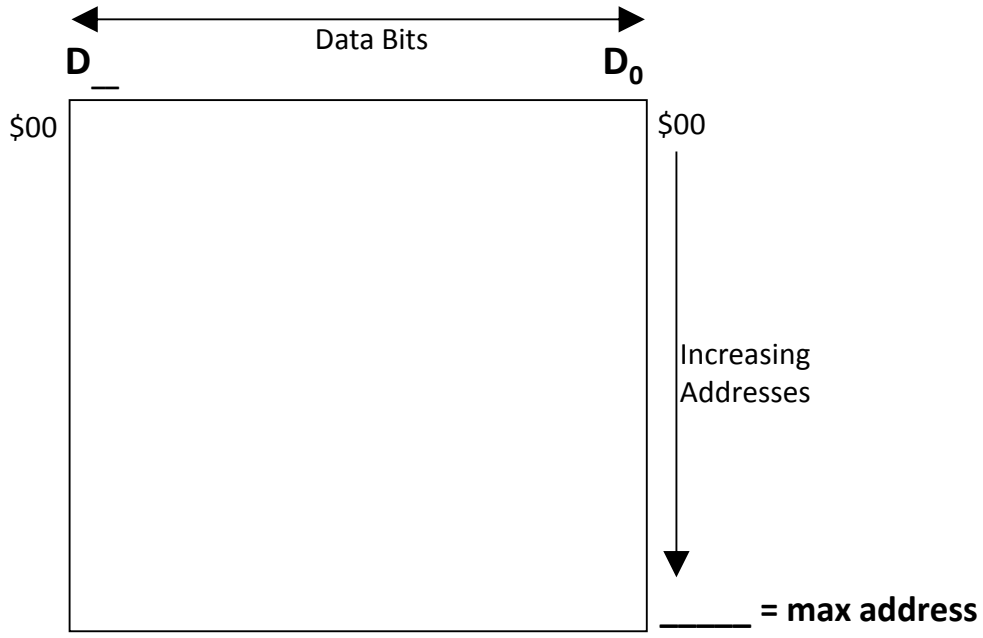
Last Name, First Name

[22%] 6. Given as many **64×4 SRAMs** as needed, up to three **32×4 EEPROMs**, and up to one **128×8 FLASH EEPROM**, design the memory module described below, with an **active-low** chip enable, **CE(L)**. The device should begin at address **zero** with a 64x8 memory block made up of **64×4** of **SRAM AND 64×4 EEPROM**. Immediate follow this memory block with **128×8 FLASH EEPROM**. Add the **minimum number** of memory devices and the **minimum number** of additional **SSI** components required. (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**)?

32x4 EEPROM(s):

64x4 SRAM(s):

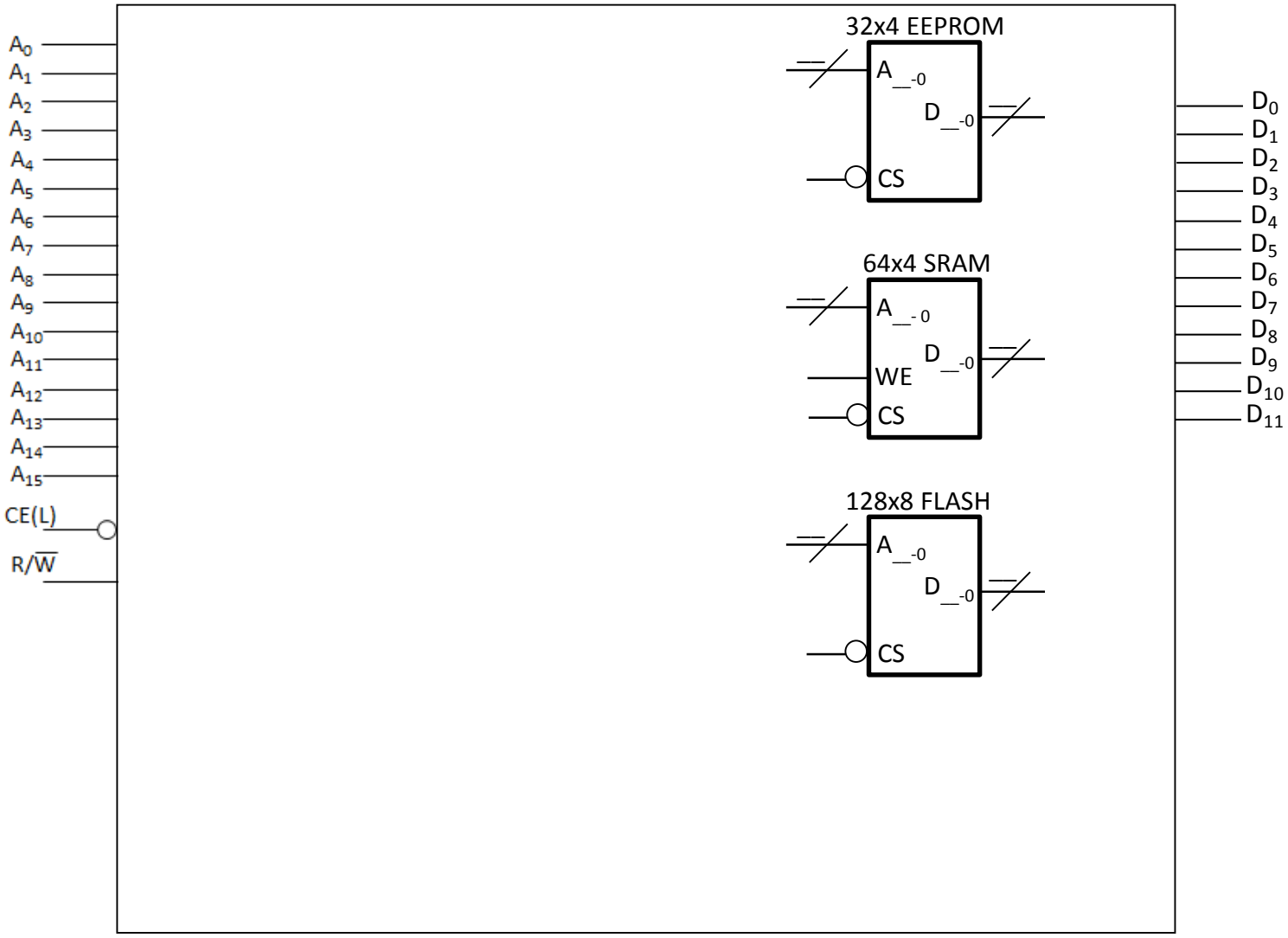
128x8 FLASH(s):

# Exam 2

Last Name, First Name

( %) 6. 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



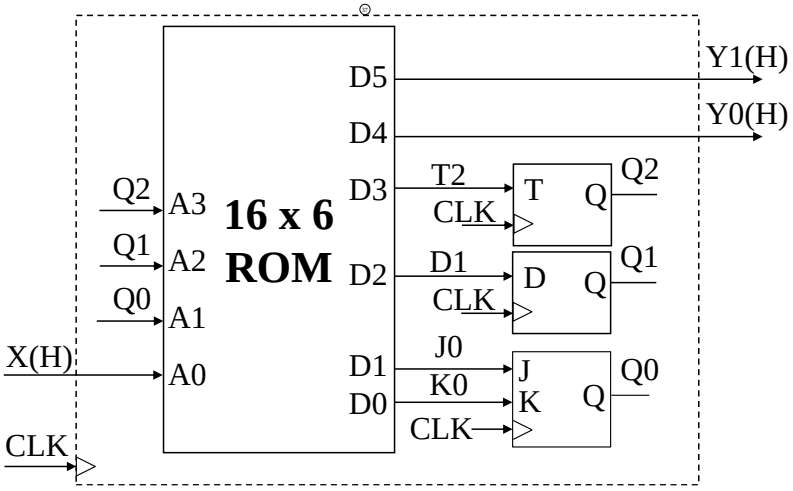

## Equations:

# Exam 2

Last Name, First Name

[20%] 3 min 7. 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, a D-FF, and a JK-FF are utilized in the below circuit.

(16%) 15 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** | **23** | | | | | | | | | | |
| | | | | | **1** | **23** | | | | | | | | | | |
| | | | | | **2** | **34** | | | | | | | | | | |
| | | | | | **3** | **15** | | | | | | | | | | |
| | | | | | **4** | **14** | | | | | | | | | | |
| | | | | | **5** | **04** | | | | | | | | | | |
| | | | | | **6** | **09** | | | | | | | | | | |
| | | | | | **7** | **09** | | | | | | | | | | |
| | | | | | **8** | **1E** | | | | | | | | | | |
| | | | | | **9** | **3A** | | | | | | | | | | |
| | | | | | **A** | **07** | | | | | | | | | | |
| | | | | | **B** | **07** | | | | | | | | | | |
| | | | | | **C** | **06** | | | | | | | | | | |
| | | | | | **D** | **06** | | | | | | | | | | |
| | | | | | **E** | **05** | | | | | | | | | | |
| | | | | | **F** | **05** | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |

# Exam 2

Last Name, First Name

7. 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.

(2%) 3 min

b) If the input X is active-low, explain how this will change the ASM (if at all). Do **NOT** redraw the ASM.

(2%) 4 min

c) If the output Y0 is active-low, **redraw the first two states of the ASM**. Do **NOT** redraw the **entire** ASM, just redraw the first two states.

# Exam 2

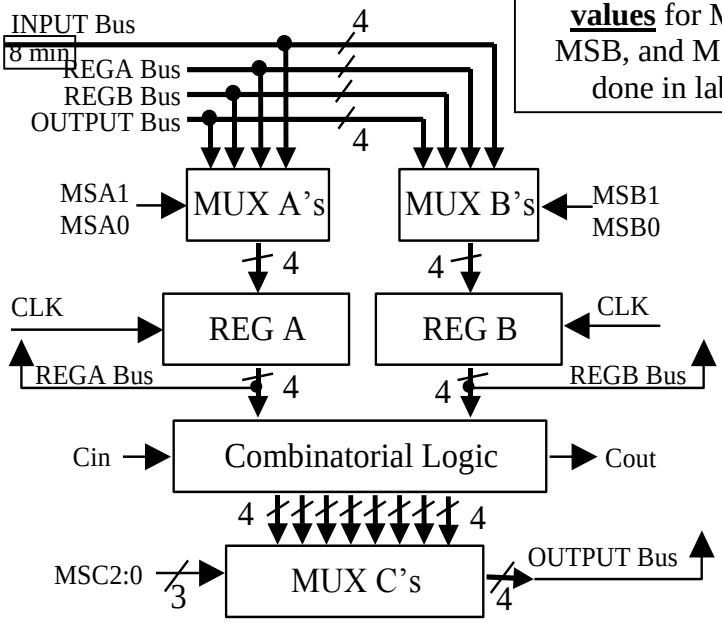
Last Name, First Name

3 min

[26%] 8. A block diagram of your lab 6 is shown here, along with two tables from the same handout, one of which is modified.

8 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 **Cout = SUM_Cout from adder** |
| 110 | **Cout = 1 if REGA ≥ 8; else Cout = 0** |
| 111 | **Cout = 1 if REGB ≥ 8; else Cout = 0** |

(10%) a) Assume that you can add a small additional circuit to calculate the value of Cout that behaves as follows and as described in the *MSC Action* table above. The value of Cout should be 0 if its value is **not** described in the above table. Determine the equation for Cout. You do **NOT** need to simplify the equation.

# Exam 2

Last Name, First Name

( %) 8. b) Exchange the values in registers A and B, i.e., if A = 3 and B = 7 before the below operations, then afterwards A = 7 and B = 3. (Your algorithm should still work even if the two numbers are changed.) If this is not possible, then say so. Describe 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 | | | | | | 0011 | 0111 | | | | | | |
| 2 | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | |

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

( %) c) Use the table below to add two numbers, 3 and 7, and put the results in register A. (Your algorithm should still work even if the two numbers are changed.) Describe 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 | | | | | | 1111 | 0110 | | | | | | |
| 2 | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | |

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

# Exam 2

Last Name, First Name

( %) 8. d) Use the table below to **SUBTRACT** $5 from $E. (Your algorithm should still work even if the two numbers are changed.) Store your solution in register B. Describe 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 | | | | | | 1111 | 1111 | | | | | | |
| 2 | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | |

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

(6%) e) Design a circuit that can be added to your circuit design to "remember" the last carry output, the **modified** Cout from part a. "Remember" the Cout any time one of the three functions that can set the value (the last three in the *MUXC Action* table) are executed. Call the remembered output **R_Cout**. Specify all the inputs and outputs for this circuit.

5 min

.