Exam 1

________________________, ________________

Last Name , First Name

Instructions:

UF's NaviGator champion in Hawaii.

- *Turn off **cell phones**, **beepers** and other **noise making devices.***
- *Show **all** work on the **front** of the test papers. 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.*
- *You may use any of your XMEGA documents with **limited** added material; highlighting and tagging is permissible. You may **not** use any notes (mine or yours), examples, homework, labs, books, calculators, computer, electronic devices, etc.*
- ***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.)*
- *The space provided does **not** necessarily represent the amount of writing needed.*
- *You must pledge and sign this page in order for a grade to be assigned.*
- *In programs, the use of comments results in **more** partial credit.*
- ***Read** each question **carefully** and **follow the instructions**.*
- *The point values for problems may be changed at prof's discretion.*
- *Part of your grade on tests, quizzes, labs, etc. is based not only on solving the problem you are presented with, but the manner in which you solve it. For example, there is a difference between two programs that meet the given specifications, where one is an elegant, extensible 20-line solution, while the other is an obfuscated 100-line program (that also meets the specifications, but would be difficult to extend later). Just as your future employer would value the latter program less than the first, so will I in grading your assignments.*
- *This exam counts for **20.7-24.3**% of your total grade.*
- *Unless otherwise stated assume the following:*
  - *The oscillator frequency is precisely 2 MHz.*
  - *The code should run on an **ATxmega128A1U** as configured on the* Out of the Box uPAD and uPAD Base Board ***without** any additional peripherals.*
  - *You can assume the standard bit equates that I have used in class examples (e.g.,* BIT0 = 0b0000 0001, BIT76 = 0b1100 0000b, INV76 = 0b0011 1111*) have already been done for you.*

May the Schwartz be with you!

Good luck and 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.

______________________ ______________________ ______________

PRINT YOUR NAME | SIGN YOUR NAME | DATE (*22 Feb 17*)

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

Pages	Available	Points
2-4	40	
5	8	
6	11	
7	11	
8-9	15	
10-11	15	
TOTAL	100	

Exam 1

______________________, ______________
Last Name , First Name

[40%] 1. Design a complete expansion to a XMEGA board (like your uPAD) by adding a **ROM**, an **SRAM**, an **input port**, an **output port**, and a **keypad**. Assume that each of the first four devices above can use **only** the address space from **0x03 2000** and **0x03 FFFF**. Complete the figure to the right and the table below for each of parts a-c.

2 min

(%) a) Add a **32K (32k x 8) ROM** in the given address range above. Use **only** **CS0** (this is the highest priority) if possible; if not, use as little additional logic gates (or PLDs) as necessary. As a second priority, place the ROM at the **lowest possible** addresses. Configure CS0 by specifying the below values. Add to the figure to the right and the table below.

6 min

Port/Memory Blocks

EBI_CTRL =
CS0_CTRLA =
CS0_BASEADDRH =
CS0_BASEADDRL =

(%) b) Also add an **8K (8k x 8) SRAM** somewhere in the given address range above. Use **only** **CS1** (this is the highest priority) if possible; if not, use as little additional logic gates (or PLDs) as necessary. As a second priority, place the SRAM at the **lowest possible** addresses. Add to the above right figure and the table below. (It is **not** necessary to calculate the CS1 control register values.)

4 min

(%) c) Also add an **input port** (In7-0) and an **output port** (Out7-0, at the same addresses as the input port) somewhere in the given address range above, filling up **all** the possible available address space. Use **only** **CS2** (this is the highest priority) if possible; if not, use as little additional logic (or PLDs) as necessary. As a second priority, place the ports at the **lowest possible** addresses. Add to the above right figure and the table below. (It is **not** necessary to calculate the CS2 control register values.)

4 min

32K (32k x 8) ROM Addr Range:
0x________ - 0x________ = 0b________________________________ - 0b________________________________

8K (8k x 8) SRAM Addr Range
0x________ - 0x________ = 0b________________________________ - 0b________________________________

Port Addr Range:
0x________ - 0x________ = 0b________________________________ - 0b________________________________

Exam 1

______________________, ______________
Last Name , First Name

(%) 1. d) Derive the **equations for the address part of address decoding**, i.e., X_{Device} = f(Addresses only). These equations should include **ALL** of the addresses (and only addresses) for a particular chip enable or device control (even if you used one of the built-in CS_X pins). In part h you will design the necessary circuits.

4 min

(%) e) Derive the **enable/control equations** that you will need to control the ports and memories, e.g., $X_{Device\text{-}Ctrl}$ = f(X_{Device}, RE, WE, ~~**Reset**~~, …). Leave **Reset** out of your equations and circuits in **this** problem, but in general, Reset should be included. In part h you will design the necessary circuits (and again leave out **Reset**).

3 min

(%) f) Write a new port address equation (similar to part d) that uses CS2, but limits the size of the used address to only the lowest possible 128 addresses, i.e., 128 addresses for the ports, e.g., PORT_Addr = f(CS2, Addresses).

4 min

(%) g) Design the keypad circuit on the bottom left of the next page; use **only external** (i.e., not internal) pull-**down** resistors, **bits 2-0** of **your** input port (In2 – In0 from part c), and **bits 7-4** of **your** output port (Out7 – Out4 from part c). Explicitly show the resistors on your circuit design, i.e., do **NOT** use internal pull-down resistors.

5 min

Write a program fragment here to jump to **Found7** if the **7** key is pressed and **No7** if the **7** key is **not** pressed. Assume that X already points to the input and output port address.

Labels	Instructions

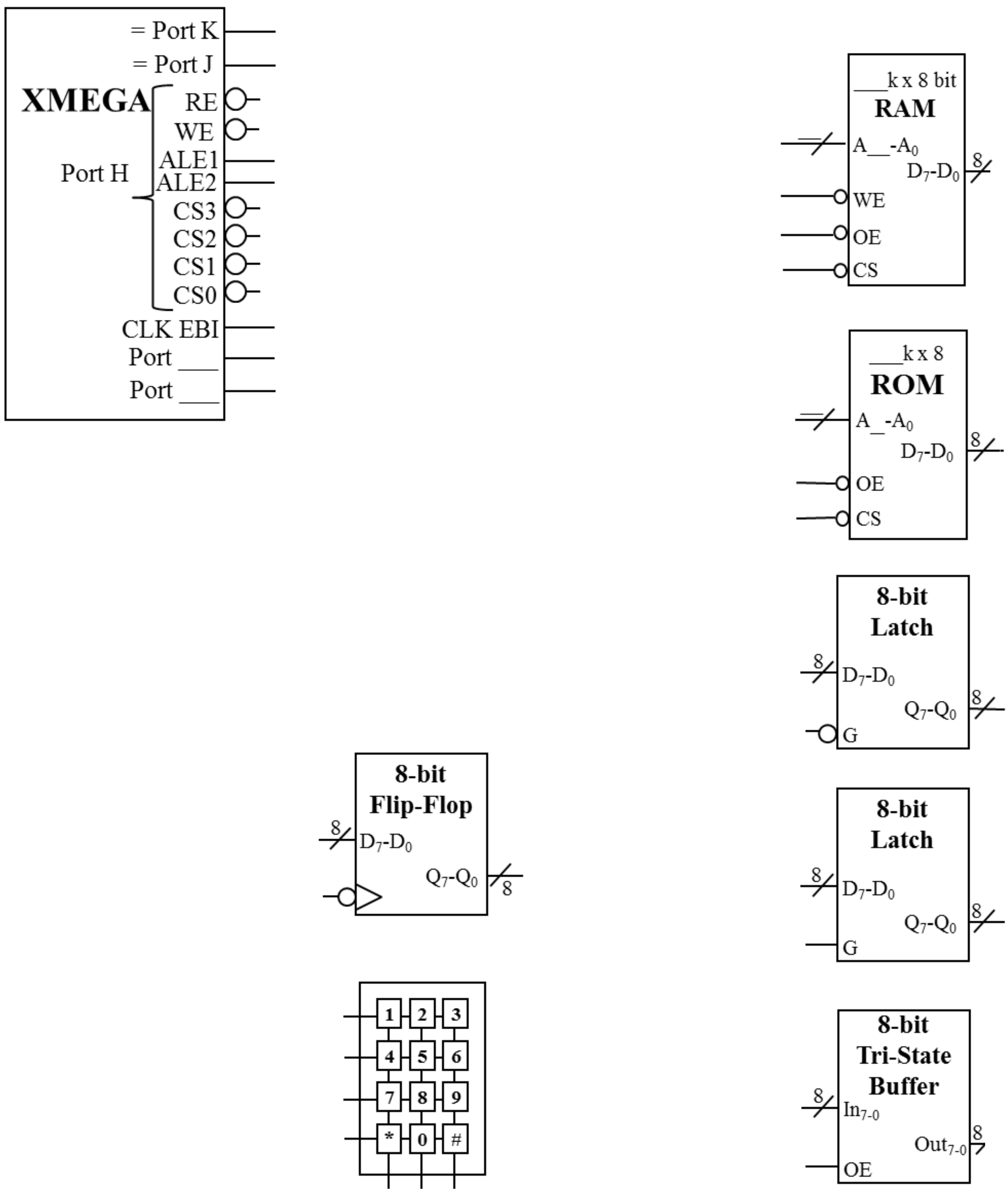
Exam 1

________________________, __________________

Last Name , First Name

(%) 1. h) Complete the circuit diagram below as specified in part a-e and g. **Please USE LABELS instead of wires!** **Please USE LABELS instead of wires!** Add **additional** components **only** if necessary (but only resistors and SSI gates, e.g., ANDs, NORs, NOTs, etc.).

12 min



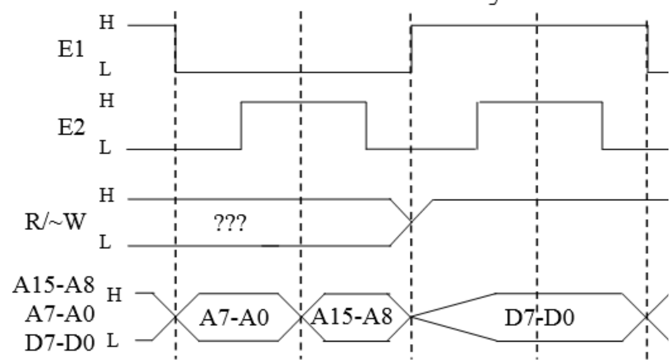
Exam 1

[8%] 2. The **UF3744-µC** is a new 8-bit microcontroller. The timing diagrams for reading and writing are shown here. Like many of the processors that we discussed in class, there is a time-multiplexed bus, in this case, for address and data. Note the two control signals on this device, E1 and E2.

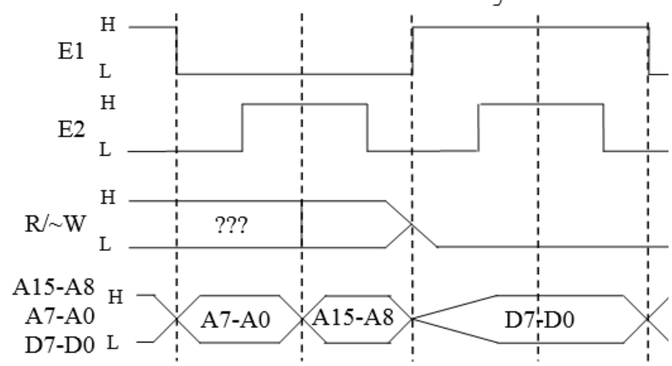
10 min

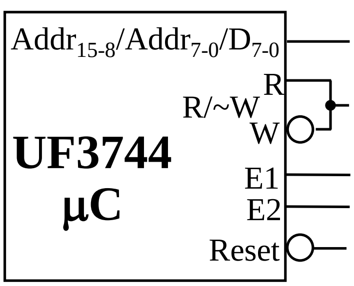
Create a **complete circuit diagram** to add a 16K (16k × 8) SRAM to the UF3744-µC, starting at address 0x4000.

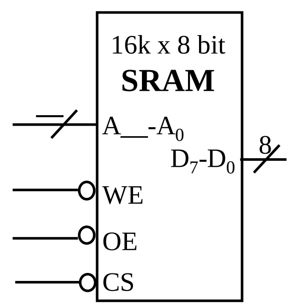
UF3744 Read Cycle



UF3744 Write Cycle







Exam 1

______________________, ______________
Last Name , First Name

[11%] 3. Answer or solve each of the following short questions.

(3%) 3 min

a) Write a program fragment with the minimum number of instructions to initialize the following PORTF pins **without affecting** the other pins: make pins 3 and 7 inputs and pins 4 and 2 outputs.

Labels	Instructions

(2%) 3 min

b) Why does the LDI instruction only work with registers r16 through r31 and not r0 through r15?

____

____

____

____

____

(4%) 4 min

c) When loading from program memory, why do we shift the Z register left by 1 bit?

____

____

____

____

____

____

____

____

(2%) 3 min

d) Write an efficient assembly language program fragment that implements the following pseudo-code:

```
if r16 = 37
     then r17 = 0xE9
     else r17 = 0x42
```

Labels	Instructions

Exam 1

_______________________, _______________
Last Name , First Name

[11%] 4. Answer or solve each of the following short questions.

(2%) 2 min

a) Explain when and why the NOP was needed in the keypad lab.

____

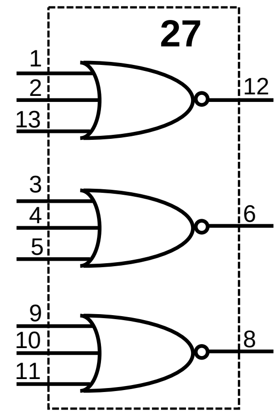
____

____

____

(6%) 5 min

b) Draw a complete **mixed-logic** circuit diagram to directly implement the below equation, i.e., do **NOT** simplify the equation. Include pin numbers on your diagram and use appropriate switch circuits and an LED circuit. Use switches, resistors and LED's as needed. The only IC (chip) you can use is the 74'27 (shown), but draw a **logic diagram** not a layout. Pick and label appropriate activation levels for A, B and Y. Show the switches in input's **true** positions. A block diagram of a 74'27 is shown to the right.



$$Y = /(A * /B)$$

(3%) 4 min

c) Explain how you might use an interrupt with the keypad and why this would be a better solution than what you did in lab. Note: You can also assume that an interrupting timer exists for whatever time you need.

____

____

____

____

____

____

____

____

____

Exam 1

_______________________, _______________
Last Name , First Name

[30%] 5. In this problem you will write an assembly language program that adds two 24-bit numbers.

(15%) a) Write a subroutine (**ADD_24bits**) that adds two 24-bit numbers stored in data memory. Prior to calling the subroutine, the **starting address** of each of the two 24-bit numbers is stored on the stack. Return the 24-bit result **at the same address** as the first of the two 24-bit numbers. You can assume that the subroutine does **NOT** have to preserve any registers. (In the next part of the problem, you will write the main routine.)

15 min

Labels	**Instructions**	**Comments**

Exam 1

_______________________, _______________
Last Name , First Name

(15%) 5. a) (Continued.) **Repeated problem:** Write a subroutine (**ADD_24bits**) that adds two 24-bit numbers stored in data memory. Prior to calling the subroutine, the **starting address** of each of the two 24-bit numbers is stored on the stack. Return the 24-bit result **at the same address** as the first of the two 24-bit numbers. You can assume that the subroutine does **NOT** have to preserve any registers. (In the next part of the problem, you will write the main routine.)

Labels	Instructions	Comments

EEL 3744—Spring 2017
22 February 2017

Exam 1

______________________, ______________
Last Name , First Name

(15%) 5. b) Write the rest of a **complete** program, i.e., the main routine, necessary to test your subroutine. The main routine must initialize **address 0x05AB37** with the value 0x123456 and **address 0x00E192** with the value 0xABEFCD. These are the values and locations that the subroutine must add. Do **ALL** necessary initializations. Don't do anything after the call of subroutine **Add_24bits**.

15 min

Labels	Instructions	Comments

Exam 1

______________________, ______________
Last Name , First Name

(15%) 5. b) (Continued.) **Repeated problem:** Write the rest of a **complete** program, i.e., the main routine, necessary to test your subroutine. The main routine must initialize **address 0x05AB37** with the value 0x123456 and **address 0x00E192** with the value 0xABEFCD. These are the values and locations that the subroutine must add. Do **ALL** necessary initializations. Don't do anything after the call of subroutine **Add_24bits**.

Labels	Instructions	Comments