Exam 1

_____, _____
Last Name , First Name

Instructions:

- *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 **13** 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, but 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 **27**% of your total grade.*
- *Unless otherwise stated assume the following:*
  - *The oscillator frequency is precisely **32 MHz**.*
  - *The code should run on an **ATxmega128A1U** as configured on the* Out of the Box uPAD (and, when specified, uPAD back backs) ***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 0000, INV76 = 0b0011 1111*) have already been done for you.*

May the Schwartz be with you!

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 (*27 June 17*)

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

Problem	Available	Points
1, 2	18	
3	28	
4	34	
5	8	
6	12	
TOTAL	100	

Exam 1

_____, _____
Last Name , First Name

[5%] 4 min 1. Assume that you have a subroutine called **DELAY_37ms** that takes precisely 37 ms from the call to the time the next instruction is ready to run. Write a subroutine (**DELAY_X**) to delay approximate X times 37 ms, where X is passed into the subroutine in register r16. Preserve **ALL** registers. If the value passed to the subroutine is zero, the subroutine should delay as close to zero as possible, i.e., **much less** than 37 ms.

Labels	Assembly Instructions	Comments (including timing info)

[13%] 2. Answer or solve each of the following short problems.

(1%) 2 min a) If the system clock frequency is 32 MHz, what is the clock **period** (a rational number, not an expression) in nanoseconds, microseconds, or milliseconds ($1\ ns = 10^{-9}\ s$, $1\ \mu s = 10^{-6}\ s$, $1\ ms = 10^{-3}\ s$)?

(1%) 2 min b) If the clock source is set to 32 MHz, what is the **slowest clock frequency** possible using only the techniques **and system** utilized in Lab 2. Show the answer (a rational number, not an expression) in GHz, MHz, or kHz.

Exam 1

Last Name , First Name

2. (continued)

(1%) 2 min c) It is obvious why we would want to run a microprocessor at a fast speed. There is one primary reason for running a processor at a **very slow speed**. What is that reason?

(1%) 2 min d) For even parity, what is the parity bit for the binary number 00110111. Why?

(3%) 3 min e) What is the purpose of the start and stop bits in asynchronous serial communication? How are these bits utilized?

(2%) 3 min f) Several XMEGA registers are "protected." From what are they protected and why? How are they protected?

(4%) 4 min g) If you need to send 1,000 blocks of 6-bit data using asynchronous serial communication at 20 kHz, precisely how long should this take? This serial communication will utilize one parity bit (even parity), one start bit, and two stop bits. Justify your solution and show all work. Suggestion: **Check your arithmetic!**

Exam 1

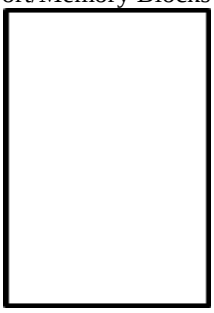
_____, _____
Last Name , First Name

[28%] 3. Design an expansion to a XMEGA board (like your uPAD, with the **SRAM 3-PORT ALE1** EBI mode) by adding an **input port**, an **output port**, an **SRAM**, a **ROM**, and a **keypad**. Complete the figure to the right and the table below for each of parts a-c. You will need a 4K SRAM, an 8K ROM, an input port, and an output port. The ports must be **at the same addresses**. The addresses for your expansion are all between 0 and 0xFFFF. Assume that address bits A_{23} through A_{16} are not needed in this problem. Do **NOT** use XMEGA chip selects for this problem. Assume that you will use a decoder (see part j) and other external circuits to create the necessary address decoding.

4 min

(18% for a-c,j)

Port/Memory Blocks



(%) 4 min

a) Add a **4K (4k x 8) SRAM** starting at address **0x4000**. Add to the above right figure and the table below. Derive the **equations for the address part of the SRAM's address decoding**, i.e., X_{SRAM} = f(Addresses only).

(%) 4 min

b) Add an **8K (8k x 8) ROM** immediately following the SRAM. Add to the above right figure and the table below. Derive the **equations for the address part of the ROM's address decoding**, i.e., X_{ROM} = f(Addresses only).

(%) 4 min

c) Add an **input port** and an **output port** immediately following the ROM. Use partial address decoding for the ports. See also the decoder in part j to determine the **most efficient** address(es) for the ports. Add to the above right figure and the table below. Derive the **equations for the address part of the PORTs' address decoding**, i.e., X_{PORTS} = f(Addresses only).

4K (4k × 8) SRAM Address Range:
0x_____ - 0x_____ = 0b_____ - 0b_____

8K (8k × 8) ROM Address Range:
0x_____ - 0x_____ = 0b_____ - 0b_____

Port Address Range
0x_____ - 0x_____ = 0b_____ - 0b_____

Exam 1

_____, _____
Last Name , First Name

(2%) 3. d) Derive the **enable/control equations** that you will need to control the SRAM, ROM, input port, and output port, e.g., $X_{Device\text{-}Ctrl} = f(X_{Device}, RE, WE, \sout{Reset}, \ldots)$. Leave **Reset** out of your equations and circuits in **this** problem, but in general, Reset should be included. In part j you will design the necessary circuits (and again leave out **Reset**).
3 min

(1%) e) How many addresses can you use for **writing** to your **output** port? _____
1 min

(1%) f) How many addresses can you use for **reading** from your **input** port? _____
1 min

(1%) g) How many addresses can you use to **write** to the first memory location in your **RAM**? _____
1 min

(1%) h) How many addresses can you use to **write** to the last memory location in your **ROM**? _____
1 min

(4%) i) Design the keypad circuit on the bottom of the next page. Use pull-**up** resistors, **bits 7-4** of **your** input port (In7 – In4 from part c), and **bits 2-0** of **your** output port (Out2 – Out0 from part c). Explicitly show the resistors on your circuit design, i.e., do **NOT** use the XMEGA's internal pull-up resistors.
5 min

Write a program fragment here to jump to **Found3** if the **3** key is pressed and **No3** if the **3** key is **not** pressed. Assume that X already points to the input and output port address.

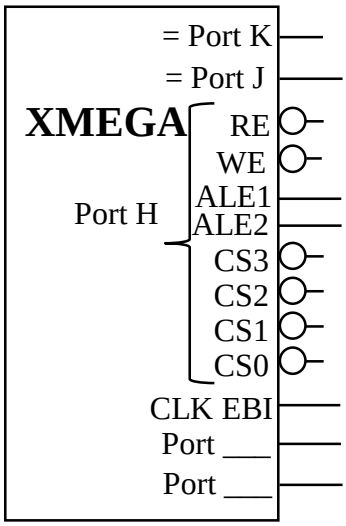
Labels	Instructions

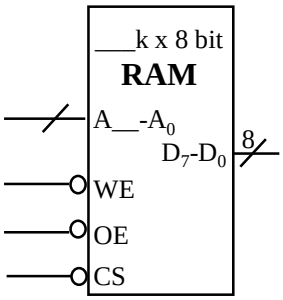
Exam 1

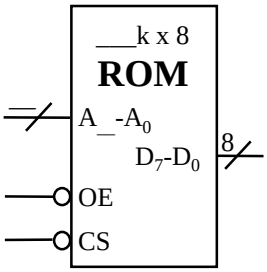
_____, _____
Last Name , First Name

(18% for a-c,j)
(%) 3. j) 12 min

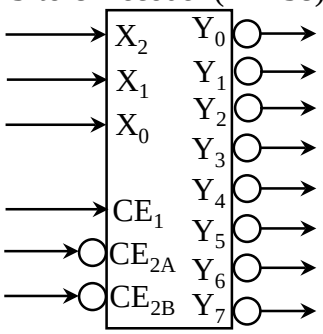
Complete the circuit diagram below as specified in parts a through d and i. You **MUST** use the **decoder** in a **non**-trivial way. **Please USE LABELS instead of wires!** **Please USE LABELS instead of wires!** Add **additional** components **only** if **necessary** (but only resistors and SSI gates, e.g., ANDs, NORs, NOTs, etc.).

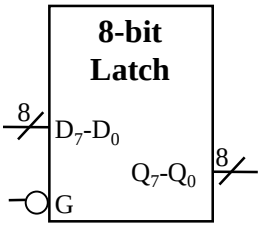


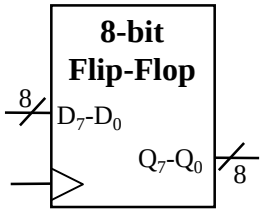


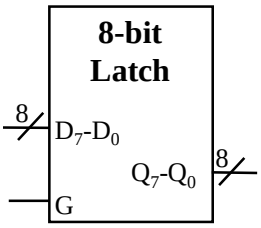


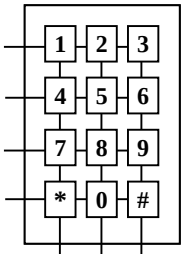
3-to-8 Decoder (74'138)

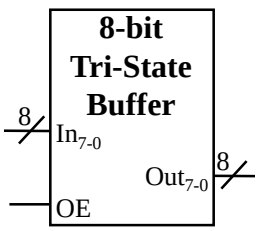












Exam 1

_____, _____
Last Name , First Name

[34%] 4. In this problem, you will ultimately write a complete program using the following systems, a switch, eight **active-high** LEDs on **Port C**, an external interrupt, and an XMEGA **timer**.

(1%) 2 min a) Set up an interrupt pin on Port F, bit 2. The interrupt should be triggered by any change on this pin. When this interrupt occurs, the direction of counting on the eight LEDs should reverse (as described further in part b). For this problem, assume that this switch was somehow debounced with hardware, i.e., it does **not** bounce.

i) Write the necessary assembly code to set up an interrupt vector for this interrupt. Name the interrupt service routine (ISR), **PORTF2_ISR**. The ISR should be executed when the **PF2** pin changes its value.

Labels	Assembly Instructions	Comments

(6%) 7 min ii) Write an ISR initialization (**INIT_PF2_ISR)** subroutine (called shortly after the start of the main routine) to appropriately initialize this external interrupt. The interrupt must be ready to execute upon returning from this subroutine, so configure all necessary registers.

Labels	Assembly Instructions	Comments

Exam 1

_____, _____
Last Name , First Name

(1%) 2 min 4. b) Set up a Port C timer to interrupt every 0.5 s. Every 0.5 s, the value at the 8-LEDs should be incremented or decremented; the direction of counting should be changed every time the switch is pressed. Remember that your assembler **can do arithmetic**!

i) Write the necessary assembly code to set up an interrupt vector for this interrupt. Name the interrupt service routine **T_PortC_ISR**.

Labels	Assembly Instructions	Comments

(9%) 7 min ii) Write an ISR initialization (**INIT_T_PortC)** subroutine (called shortly after the start of the main routine) to appropriately initialize this interrupt system. The interrupt must be ready to execute upon returning from this subroutine, so configure all necessary registers. Remember that your assembler **can do arithmetic**!

Labels	Assembly Instructions	Comments

Exam 1

_____, _____
Last Name , First Name

(5%) 5 min 4. c) Write the **PORTF2_ISR** described in part a. (In the next part, you will the **T_PortC_ISR**; in the final part, you will write the rest of the program including the main routine.)

Labels	Assembly Instructions	Comments

(5%) 5 min 4. d) Write the **T_PortC_ISR** described in part b.

Labels	Assembly Instructions	Comments

Exam 1

______________________, ______________
Last Name , First Name

(7%) 7 min 4. e) Write the rest of the program including the main routine and all required initializations not already done.

Labels	Assembly Instructions	Comments

Exam 1

________________________, ________________

Last Name , First Name

[8%] 8 min

5. Assume that a single byte is placed (passed) **on the stack** immediately before calling the subroutine **ADD_IT**. Write the **complete ADD_IT subroutine**, to add the passed value to each of the 37 successive bytes starting at address **TABLE** in data memory space (defined elsewhere, and between 0 and 0x3F FFFF). Replace the old value with the new one. Other than r16, r17, and X, preserve all registers.

Labels	Assembly Instructions	Comments

Exam 1

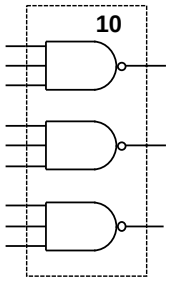
_____, _____
Last Name , First Name

[12%] 6. Answer or solve each of the following short problems.

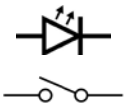
(2%) 2 min a) What is the difference between a simulator and an emulator (in our course)?

(2%) 2 min b) Assume that a small program has 37 instructions. Approximately how many **bytes** of memory are necessary? Explain.

(5%) 5 min c) Draw a complete **mixed-logic** circuit diagram to implement the below equation. (Do not "simplify" the equation.) The circuit should do nothing else. Use the minimum number of required logic gates. Include appropriate switch circuits and an LED circuit on your diagram. Use switches, resistors and LED's as needed. The only IC (chip) you can use is the 74'10 (shown); draw a **logic diagram** not a layout. Pick and label appropriate activation levels for UF, G, and O. Show the switches in their **true** positions. An LED circuit design symbol and an switch circuit design symbol are shown on the right.



$$UF = \overline{G + \bar{O}}$$



–

+

(3%) 3 min d) When a branch instruction condition is true, how does the processor determine the correct destination?

Exam 1

_____, _____

Last Name , First Name