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.
- Put your name at the top of each test page and be sure your exam consists of 12 distinct pages.
- 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.
- 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 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 configure on the UF uTinkerer Development Board **without** any additional peripherals.

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.

<table>
<thead>
<tr>
<th>PRINT YOUR NAME</th>
<th>SIGN YOUR NAME</th>
<th>DATE (18 Feb 14)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Pages</th>
<th>Available</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-5</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>8-9</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>11-12</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
1. In this problem you will design an expansion to a XMEGA board (like your uTinkerer) by adding an input port, an output port, an SRAM, a ROM, and a keypad.

(a & c- e = 22%) a) Assume that no external components have yet been added to your XMEGA board. In parts a-e of this problem you will design the circuits to add an 8-bit input port, an 8-bit output port, a 16k x 8 SRAM, a 32k x 8 ROM, and a keypad. The 16k x 8 SRAM should be fully address decoded and start at 0x13 3000. The 32k x 8 ROM should be fully address decoded and start at 0x13 8000. The two ports should be accessible from all addresses in between the SRAM and ROM. You will also add a keypad, using only Port D on the XMEGA and resistors, if necessary. Fill in the addresses below and draw in port/memory blocks to the right to help you solve this problem. Note that the keypad circuit should be made to recognize one OR MORE keys in the SAME row.

(b) If you need an XMEGA CS, you can use only CS3. If necessary, configure CS3 for this problem.

10 min

16K (16k x 8) SRAM Addr Range:

0x13 3000 - 0x_______ = 0b_________________________ - 0b_________________________

Port Addr Range:

0x_______ - 0x_______ = 0b_________________________ - 0b_________________________

32K (32k x 8) ROM Addr Range:

0x_______ - 0x_______ = 0b_________________________ - 0b_________________________

(4%) b) If you need an XMEGA CS, you can use only CS3. If necessary, configure CS3 for this problem.

4 min

EBI_CTRL =
CS3_BASEADDRH =
CS3_BASEADDRL =
CS3_CTRLA =
c) Derive any necessary **equations for the address part of address decoding**, i.e., $X_{\text{Addr}} = f(\text{Addresses only})$. In part e you will design the necessary circuits.

\[
(\%)
\]

\[4 \text{ min}\]

\[
\]

\[
\]

d) Derive the **control equations** that you will need to control the ports and memories, e.g., $X_{\text{ctrl}} = f(X_{\text{Addr}}, \text{RE}, \text{WE}, \text{Reset}, \ldots)$. In part e you will design the necessary circuits. Be sure to disable your devices when Reset is **true** and to protect your hardware from bad software.

\[
(\%)
\]

\[4 \text{ min}\]
(a & c – e = 22%)

1. e) Complete the circuit diagram below as specified in part a. **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.).

- **XMEGA**
  - Port H
  - Port A
  - Port B
  - Port ___
  - Port ___

- **8-bit Flip-Flop**
  - D7-D0
  - Q7-Q0

- **8-bit Latch**
  - D7-D0
  - Q7-Q0

- **74HC154 Decoder**
  - X3, X2, X1, X0
  - Y0, Y1, Y2, Y3

- **16k x 8 bit RAM**
  - A – A0
  - D7-D0

- **32k x 8 ROM**
  - A – A0
  - D7-D0

- **8-bit Tri-State**
  - In: 0
  - Out: 0

- **8-bit Tri-State**
  - In: 0
  - Out: 0
Exam 1

1. (2%) f) How many different addresses can be used to write to the input port? Show your work. You may leave your answer in the form of an expression, e.g., 37k or 2^7.

2. (3%) g) If the ROM was made with four 16k×4 ROMs instead of a single 32k×8 ROM, draw the complete circuit diagram for the ROM design.

3. (3%) h) Using your circuit design for the keypad, describe how you can detect if the * and # key are pressed simultaneously.
2. With the XMEGA on our uTinkerer, we use shared pins in the External Bus Interface mode, namely A15:8 are shared with A7:0.

(a) Draw a timing diagram for a write cycle for our processor if the shared pins were instead A15:8 and D7:0. Be sure to show all relevant signals.

(b) Draw the necessary component(s) and create a circuit diagram to separate the busses in part a, i.e., A15:8 and D7:0. One set of inputs (others may be necessary) to your circuit is A15:8/D7:0 and the two sets of outputs to your circuit are A15:8 and D7:0. Clearly show these input(s) and outputs going into and out the box with your circuit diagram inside.
3. Write the two related subroutines described below.

- **a)** Write a subroutine, `INIT_PORTE`, to initialize PortE, bits 3 and 7 for input and bits 2 and 6 for output. Do **NOT** affect any of the other pins on PortE.
- **b)** Write a subroutine, `READ_WRITE`, to read the values on PortE, bits 3 and 7. If bit 3 is set, toggle the value in bit 2; otherwise leave bit 2 unchanged. If bit 7 is set, toggle the value in bit 6; otherwise leave bit 6 unchanged. Do **NOT** affect any of the other pins on PortE.
4. Write a complete program to do the following. Assume that a table of numbers are already in program memory at TAB1 (a label that you should use). Copy all but the first number into a new table in data memory at TAB2 (a label that you should use), except that for every number in the table less than 0x37, add 0x37 to that number before copying. The first number in TAB1 is the number of items to copy. Use of assembler directives is strongly encouraged, and often required. Provide your own TAB1 table to test your program. There is more room on the next page.
4. (continued)

<table>
<thead>
<tr>
<th>Labels</th>
<th>Instructions</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
[14%] 5. Assume an XMEGA subroutine named Sub_Exam1 is located at address 0x37AB. Write the instruction at address 0x36E0 for appropriately initiating subroutine execution in the box below left. If possible, use a rcall rather than a call. Write the stack pointer initialization instruction(s) in the box below right. Initialize the stack pointer at an appropriate location in internal XMEGA SRAM. Assume R27 = 0xC3.

<table>
<thead>
<tr>
<th>Address</th>
<th>Subroutine execute instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x36DF</td>
<td>push R27</td>
</tr>
<tr>
<td>0x36E0</td>
<td>pop R23</td>
</tr>
<tr>
<td>0x36__</td>
<td>pop R23</td>
</tr>
</tbody>
</table>

Stack pointer initialization instructions

Describe the stack on the XMEGA as it relates to a subroutine.

For each of the below descriptions, draw the stack. In each case, show an arrow (→) at the location of the stack pointer. On the far left, show (valid) addresses for each of the used stack locations. Also provide the values requested at the bottom of these tables.

1) after the stack is initialized
2) after push R27 (R27 = 0xC3)
3) at the start of the subroutine execution
4) after a push R16 (R16 = 0x25)
5) after a pop R18 instruction
6) after the return from the subroutine
7) after pop R23

Addresses | 1-After Init | 2-After push R27 | 3- at start of subr | 4- after push R16 | 5-After pop R18 | 6-After return | 7-After pop R23 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SP = ___________ SP = ___________ SP = ___________ SP = ___________ SP = ___________ SP = ___________ SP = ___________
PC = ___________ R18 = ___________ PC = ___________ R23 = ___________
6. [15%] Answer the following short questions.

a) How does the microprocessor know if you are using unsigned or two’s complement numbers?

b) List two things that the DAD (Digilent Analog Discovery) board could have helped you accomplish as a 3701 student that could NOT be accomplished with what was available.

c) If an XMEGA chip select is used for address decoding for an input port, do we need a CPLD (or similar device)? Why or why not?

d) If we use an XMEGA chip select for address decoding for an SRAM or ROM, do we need a CPLD (or similar device)? Why or why not?
6. (continued)

(2%) Draw an LED circuit diagram for a microprocessor output that will light up an LED when the active-low signal, X(L) is true. The circuit should do nothing else. (The LED circuit design symbol is shown below.)

(2%) Draw a switch circuit diagram for a microprocessor input, Y(L). (The switch circuit design symbol is shown below.) Use the shown switch position as the true position.

(2%) What is the default data direction for GPIO pins on all microprocessors and why?

(1%) What is the purpose of a bypass capacitor?