CSCE 236 Embedded Systems, Spring 2016
Exam 1

Thursday, February 18, 2016

Instructions: You will have the full class period to complete this test. **Make sure to show your work to ensure you receive partial credit if your final answer is incorrect.** This is a closed book quiz, no computers, textbooks, notes, etc. are allowed.

Unless otherwise specified, assume that questions are referring to the Arduinos/Atmel processors we have been using in class.

Name:

Problem 1. *Warmups (Circle* **all** *answers that apply).*

a). *(2 pts) What clock speed does our Arduino operate at?*

(a) 8MHz

(b) 12MHz

(c) 16MHz

(d) 20MHz

b). *(2 pts) If an LED connected to our Arduino drops 2.5V and there is a 100 ohm resistor connected in series with it, what current(s) will be flowing through it?*

(a) 10mA

(b) 25mA

(c) 250mA

(d) 500mA

c). *(2 pts) On our Arduino, which memory is non-volatile?*

(a) SRAM

(b) EEPROM

(c) Flash

(d) SNAM

d). *(2 pts) How could you fix a button that "bounces" so that it does not bounce?*

(a) Add a capacitor

(b) Add a pullup resistor

(c) Read the pin in analog mode

(d) Add a delay after your digital read

e). *(2 pts) If an operation with an 32-bit variable takes 100 clock cycles, approximately how many clock cycles would it take with a 8-bit variable?*

(a) 10

(b) 20

(c) 50

(d) 25

f). *(2 pts) An Instruction Set Architecture defines which of the following for a processor:*

(a) The memory architecture

(b) The number of CPU registers

(c) The number of clock cycles an operation will take

(d) The amount of SRAM

Problem 2. *Hex and bit operations (all references to bit locations are zero referenced). For each bit operation subproblem write a* **single** *line of* `C` *code to achieve the desired result.*

a). *(4 pts) What is the value of* `((7<<3) - (1<<4))` *in hex?*

b). *(4 pts) What is the value of* `(((3<<4) & (1<<5)) + 2)` *in hex?*

c). *(4 pts) Set bit 5 in the variable* `var` *to 1.*

d). *(4 pts) Clear bit 7 in the variable* `var`*.*

e). *(4 pts) Set bits 3-5 (inclusive) in the variable* `var` *to bits 9-11 in the 16-bit variable* `config`*. As with all of these problems, do this in a single line of code (e.g.* `var=...`*). The bit references are zero indexed.*

Problem 3. *Memory operations. Refer to the following code example for this question:*

```
uint8_t data[] = {0x2, 0x3, 0x6, 0x02, 0x18, 0x20, 0x16};
uint8_t *globalPtr;
uint8_t finalResult;

void function(void){
  static uint32_t counter = 0;
  counter++;
  globalPtr++;
}

int8_t main(void){
  uint16_t var = data[4] + (data[5] << 8);
  uint8_t *p1 = data - 3;

  globalPtr = data;

  function();
  p1[1] = 5;
  *globalPtr = 2;
  globalPtr = 0x10;
  *(globalPtr + 2) = 9;

  function();
  finalResult = *globalPtr + var;

  //Draw Memory Map Here
  return 0;
}
```

a). *(10 pts) Fill in the below memory map for the above code after execution has reached the statement* `Draw Memory Map Here`*. Assumptions you should make: 1) the compiler allocates the memory in the order the statements appear; 2) global variables are allocated starting with the high address; 3) the stack starts at zero; 4) each memory location stores 8 bits; and 4) the memory is little endian (least significant byte stored at lowest address).*

Address	Variable	Value
0x14		
0x13		
0x12		
0x11		
0x10		
0xF		
0xE		
0xD		
0xC		
0xB		
0xA		
0x9		
0x8		
0x7		
0x6		
0x5		
0x4		
0x3		
0x2		
0x1		
0x0		

b). *(5 pts) Where does* `malloc` *allocate memory? Describe why you should avoid using* `malloc` *on an embedded system.*

Problem 4. *Debugging*

a). *(5 pts) Below is assembly code that represents three lines in the corresponding C program. What operations do the three lines of code perform (just give the type of operation, no need for specific values)? What data types are used? Label each line of the assembly code to indicate what is going on.*

```
000000da <loop>:
  da:   80 91 10 01     lds     r24, 0x0110
  de:   8d 5f           subi    r24, 0xFD       ; 253
  e0:   80 93 10 01     sts     0x0110, r24
  e4:   60 91 11 01     lds     r22, 0x0111
  e8:   70 91 12 01     lds     r23, 0x0112
  ec:   6a 51           subi    r22, 0x1A       ; 26
  ee:   7f 4f           sbci    r23, 0xFF       ; 255
  f0:   70 93 12 01     sts     0x0112, r23
  f4:   60 93 11 01     sts     0x0111, r22
  f8:   0e 94 60 00     call    0xc0    ; 0xc0 <_Z3runhj>
```

b). *(5 pts) In the disassembly there sections called* `.data`*,* `.text`*, and* `.bss`*. What do these represent and which are stored in flash and which are stored in SRAM?*

c). *(5 pts) Give two debugging capabilities that are enabled by using a JTAG. Explain a situation where using a JTAG would not help in debugging.*

Problem 5. *Schematics: For these problems refer to the schematic for the Arduino.*

a). *(5 pts) What port and pin number does the pin labeled 9 on the Arduino correspond to on the Atmel processor?*

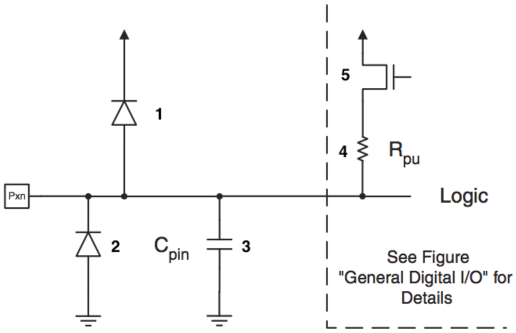
b). *(5 pts) What is the part number of the regulator that supplies 3.3V for the Arduino?*

c). *(5 pts) What is the purpose of part* `U5A`*? Make sure to describe how it works.*

Problem 6. *Digital I/O*

a). *(5 pts) In the following digital I/O schematic for the Atmel processor pins, label the five circuit elements (labeled 1-5 in image) and* **describe** *what they are used for.*

Figure 14-1. I/O Pin Equivalent Schematic



b). *(5 pts) Write the* `C` *code to configure pin* `PB3` *as an output and set it low by directly writing to the processor registers (recall the I/O registers have the general names* `DDRx`*,* `PORTx`*,* `PINx`*).*

c). *(5 pts) Write the Arduino code to set pin* `3` *to an input pin with the internal pullup resistor enabled.*

d). *(5 pts) Draw a signal that shows what happens when a button bounces. Make sure to label the diagram to make it clear when and why bouncing is occurring.*