Do NOT write near this QR Code

EE109: Intro to Embedded Systems
Fall 2025 – Quiz 1
09/30/25, 7PM – 8:15PM

[Complete all the information in the box below.]

Name:______________________________

Student ID: ____

Email: ___________________@usc.edu

Lecture section (Circle One):

	Redekopp	Puvvada	Weber
	11 a.m.	12:30 p.m.	2 p.m.

- **Erase and fill in checkboxes completely (e.g. □ => ■)**
- **All work MUST be on the FRONT (not back) of EXAM PAGES.**
- **No Scratch work will be graded or viewed.**
- **Do NOT write in the upper-right corner of the page with the QR code.**

Ques.	**Your score**	**Max score**	**Recommended Time**
1		8	8 min.
2		10	10 min.
3		12	15 min.
4		6	7 min.
5		14	35 min.
Total		50	75 min.

Calculators are ONLY allowed on Question 3 – Analog/Resistive Circuits.

Do NOT write near this QR Code

1. **(8 pts.) Number Systems**

1.1. Convert **10111001** binary to unsigned **decimal**: ______________________

1.2. Convert **105 decimal** to **(unsigned) binary**
(use exactly 8 bits): 0b________________________

1.3. Convert **110.01011101** unsigned **binary** to **hexadecimal**:
□ C.B5 □ 5.BA □ 6.BA □ C.5B □ 6.5D □ None of these

Suppose, **Club** A has **9** members, **club** B has **5** members, and **club** C has **2** members

1.4. If we combined the members of ALL 3 clubs and assigned each one a unique binary number, we would need to use a minimum of ______ bits.
□ 3 bits □ 4 bits □ 5 bits □ 6 bits □ 64 bits □ None of these

1.5. If we represented each member with two values: a unique **club** ID and a unique **member** ID (unique within that club), **club A's** members would need use a ___ (**□ 1 / □ 2 / □ 3**) bit club ID and a ___ (**□ 2 / □ 3 / □ 4 / □ 5 / □ 9**) bit member ID.

1.6. The value $\mathbf{2^{18}}$ is approximately: **□ 128K □ 256K □ 256M □ 13K □ 23M □ 512K**

2. (**10 pts**.) **Bit Manipulations**. Complete the following **single-line** statements to perform the desired operation stated in the line(s) above the blanks. Assume a standard bit numbering (**bit 7** is the MSB, **bit 0** is the LSB = Least Significant Bit). You may NOT change the structure or code given.

2.1. Turn off (clear to 0), bit **4** of PORTD without affecting other bits of PORTD.

```
PORTD ____________________________;
```

2.2. Turn on (set to 1), bit **6** and **1** of DDRB without affecting other bits of DDRB.

```
DDRB ____________________________;
```

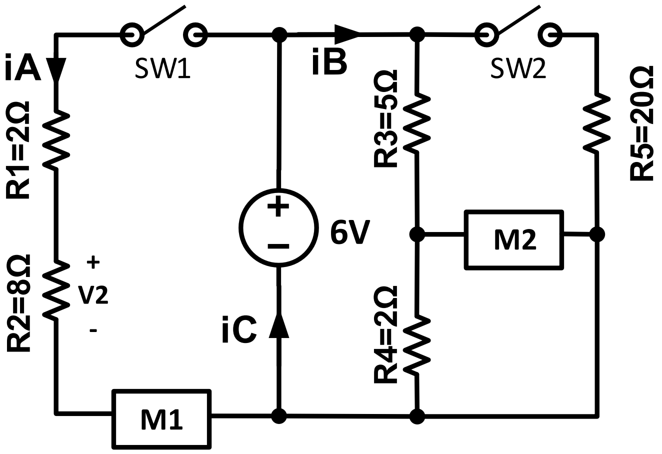
2.3. Assume bit **7, 5, and 2** of Group C are already configured to be inputs.
By accessing a register ONLY once, complete the if statement to be true ONLY if **bit 7 and 5 are logic '1' (high voltage) and bit 2 is logic '0' (low voltage)**.

```
if(_______________________________________     ____      0x_____) {…};
                                         (choose == or !=)
```

3. **(12 pts.) Resistive Circuits.** Examine the circuit below where M1 and M2 are "mystery" (unknown) devices. For **ALL problems**, **assume all switches are OPEN**, and we will only specify which are **closed** in the problem. ***A calculator may be used for this problem only.***

All problems are worth **1.5 pts** unless otherwise specified.

Partial credit will not be given.



3.1. If both switches are **CLOSED**, which of the following are true:

❑ **iA = iB + iC**
❑ **iB = iA + iC**
❑ **iC = iA + iB**
❑ **None of these**

3.2. Suppose it is not known whether **SW1** and **SW2** are open or closed nor what the mystery devices, M1 and M2, are. We can say that **iB** ____ **(is / may be / is not**) zero (0) amps.

❑ **is** ❑ **may be** ❑ **is not**

3.3. If **SW1** is **closed,** and **M1** is a **wire** then **iA** = ________Amps

3.4. If **SW1** is **closed,** and **M1 is a 2Ω resistor** then **V2** = ________ Volts

3.5. If **SW1** is **closed,** and **M1 is an OPEN circuit**, then **V2** = ________ Volts

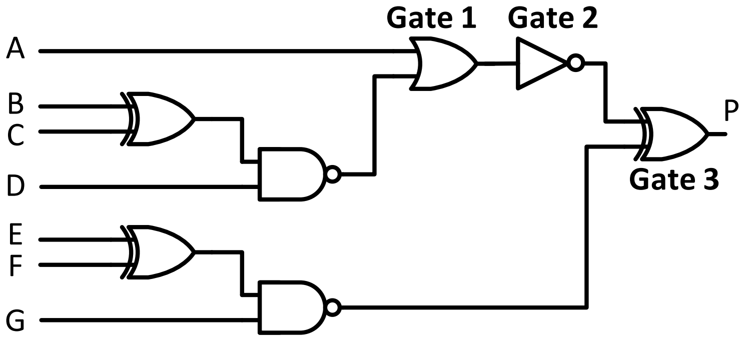
3.6. If **SW2** is **closed,** and **M2 is an OPEN circuit**, then **R3** is in parallel with **R5**. ❑ **T** ❑ **F**

3.7. If **SW2** is **closed,** and **M2 is a wire**, then **R3** is in series with **R4**. ❑ **T** ❑ **F**

3.8. If **SW2** is **closed,** and **M2 is a wire**, then the equivalent resistance of **R3, R4, R5, and M2** (i.e. the right half of the circuit) is:

❑ **1.85 ohms** ❑ **4 ohms** ❑ **5.19 ohms** ❑ **6 ohms** ❑ **27 ohms** ❑ **None of these**

4. (6 pts.) **Logic Circuits**: Consider the circuit shown below and reproduced at the bottom for markup.



Questions:

4.1. An equivalent circuit can be formed by replacing **gate 1 and gate 2** with a single: _____ gate.

□ NOR □ NAND □ XOR □ XNOR

4.2. Assuming **gate 1 and gate 2** were replaced with your answer from part 1, how many **levels of logic** would the resulting circuit be?

□ 2 □ 3 □ 4 □ 5 □ 6 □ None of these

4.3. Given {A,B,C,D,E,F,G} = {0,1,0,1,0,0,1}, the output P is _____. **□ 0** **□ 1**

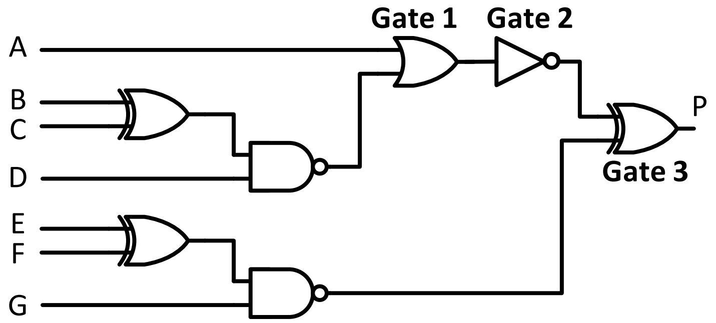
4.4. Given {A,B,C,D,E,F,G} = {0,0,1,0,1,1,1}, the output P is _____. **□ 0** **□ 1**

4.5. If **A=1**, the values of **B, C, & D** have **NO EFFECT** on the output **P**. **□ True** **□ False**

4.6. It is given that **D=0** and **G=0** (but **NO** other inputs are known), we can say that **P=____?**

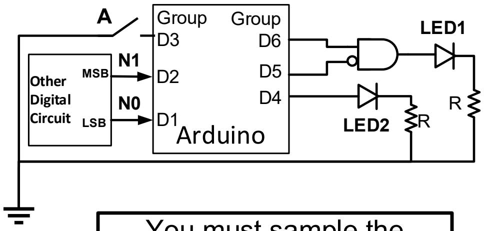
□ 0 □ 1 □ Unable to be determined

Repeated drawings for your own scratch work and annotation (will not be graded):



5. (14 pts.) **READ THE ENTIRE PAGE before solving.** Using the Arduino circuit below (exactly as drawn) with three inputs: **1 button (A) on group D, bit 3, and a 2-bit digital input: (N1 and N0) produced by another digital circuit (not buttons/switches) on group D, bits 2 and 1, respectively**. You should control **LED1, which is connected to an INVERTER and AND gate (shown in the diagram) that connects to Arduino outputs group D, bit 5 and 6, and LED2 on group D, bit 4.** **Complete** the code on the following page that implements the behavior described below.

All buttons must be sampled every **50 ms**. You must implement the different simultaneous behaviors for **LED1** and **LED2** as described in the diagram below.



You must sample the buttons every 50 ms

LED1 Behavior

LED1 should be :

- **ON** when **A is pressed and N1,N0 = 3 dec**.
- **OFF**, otherwise.

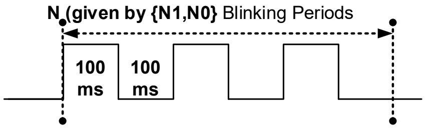
LED2 Behavior

When **A** is **pressed:**

- Sample {N1,N0} which acts as a 2-bit binary number, N, and then Blink LED2 **on** for **100** ms and **off** for **100** ms for the number of periods given by N. So if {N1,N0} = **10**, blink the LED **2** times.

When **A** is **NOT pressed:**

- LED2 should be **OFF**.



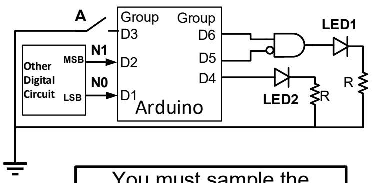
For LED2, the implementation on the next page samples the inputs to produce a variable `n` that corresponds to the value (3, 2, 1, or 0) of the two bits: **N1** and **N0**. It then derives a `count` variable to track time and produce the blinking sequence. Complete the code on the next page by filling in the appropriate blanks.

Important Requirements and Reminders

- The **bit positions and some MASKS for the inputs and outputs are defined at the start of the program**. These can be used wherever needed in the program.
- The program must **check/sample the button inputs every 50 ms**.
- You may **NOT** change the code structure given, but may **ONLY** fill in the given blanks.
- You need not worry about debouncing the button presses.
- **Recall**: All DDR and PORT register bits initialize to 0 on startup.

Complete your code on the page below!

```
#include <avr/io.h>
#include <util/delay.h>       // allows for _delay_ms() function
const int A=3, N1=2, N0=1, LED2=4;
int main() {
  unsigned int n = 0, count = 0;
  unsigned char IMASK1 = 0x0e, IMASK2 = 0x08, IMASK3 = 0x06; // Masks to use below
  unsigned char OMASK4 = 0x70, OMASK5 = 0x60, OMASK6 = 0x10; // Masks to use below
  // Add the appropriate initialization (at most 3 lines)
  DDRD ____  (OMASK__);   // Add an operator and choose between OMASK 4, 5, or 6

  ______ ___  (IMASK2);   // Fill in the register and operation (&=, |=, etc.)
  while(1) {
    // LED1 logic

    if( (_________ & (IMASK___)) == ________________________________) {
      // To do: Choose a register, IMASK 1, 2 or 3 & a correct comparison constant
      PORTD |= (1 << 6);
    } else {
      PORTD _______________________;  // Complete the statement
    }

    // LED2 logic

    if( (_________ & (IMASK___)) == ________________________________) {
      // To do: Choose a register, IMASK 1, 2 or 3 & a correct comparison constant
      n = (PIND & (IMASK__)) >> ___; // To do: Sample and produce number, n
      count = ________;              // To do: Write an expression in terms of n
    }

    if( count > 0 ) {
      count--;
      if(count % ___ == 3) {         // Fill in the appropriate expression
        PORTD |= __________________________;   // Fill in an appropriate mask
      } else if(count % ___ == 1) {
        PORTD _____________________________;   // Complete the statement
      }
    }
    _delay_ms(50);            // Cannot be changed
  } /* end while */
  return 0;
} /* end main */
```



You must sample the buttons every 50 ms

LED1 Behavior

LED1 should be :

- **ON** when **A is pressed and N1,N0 = 3 dec.**
- **OFF**, otherwise.

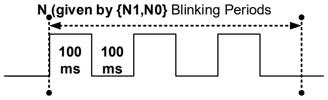
LED2 Behavior

When **A** is **pressed:**

- Sample {N1,N0} which acts as a 2-bit binary number, N, and then Blink LED2 **on** for **100** ms and **off** for **100** ms for the number of periods given by N. So if {N1,N0} = **10**, blink the LED **2** times.

When **A** is **NOT pressed:**

- LED2 should be **OFF**.



Do NOT write near this QR Code

SCRATCH PAPER. NO WORK ON THIS WILL BE GRADED.