Nothing is due tomorrow during lab.

Exam tonight!

Tuesday: 3/12/2019
6:00pm - 8:00pm
LC – A1

3 sections (about 1/3 of the points per section)

- Arduino programming
- HW 1
- HW 2
- Solutions to homeworks in Blackboard

Arduino
- setup()
- loop()

- Blink example (intro to Lab 1)
- Brooklyn Debounce (intro to Lab 2)

```cpp
int val;               // variable for reading the pin status
int val2;             // variable for reading the delayed/debounced status
int buttonState;      // variable to hold the button state

void loop(){
    val = digitalRead(switchPin);      // read input value and store it in val
    delay(10);                         // 10 milliseconds is a good amount of time
    val2 = digitalRead(switchPin);     // read the input again to check for bounces
    if (val == val2) {
        if (val != buttonState) {
            buttonState = val;
        }
    }
}
```

From <http://www.ladyada.net/learn/arduino/lesson5.html>

Homework 1
- two's complement. addition, overflow
- Parallel and Series circuits
- Ohm's Law  \( V = IR \)
  - Series Circuit
    - Total Resistance: \( RT = R_1 + R_2 + R_3 \)
    - Current: constant i.e. the same throughout the circuit
    - Voltage: change as passes through resistors
- Parallel Circuit
  Total Resistance: \(1/RT = 1/R_1 + 1/R_2 + 1/R_3\)
  Current: changes based on resistors in each path
  Voltage: constant

- CMOS gate (lowest level) - Complementary
  - pmos
  - nmos

- Gate Level entity
  NOT, AND, OR, NAND, NOR, XOR, XNOR
- Truth Table

\[
\begin{array}{ccc|c|c|c}
  A & B & C & D & F \\
  0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 1 & 1 \\
  0 & 0 & 1 & 0 & 1 \\
  0 & 0 & 1 & 1 & 0 \\
  0 & 1 & 0 & 0 & 1 \\
  0 & 1 & 0 & 1 & 1 \\
  0 & 1 & 1 & 0 & 0 \\
  0 & 1 & 1 & 1 & 1 \\
  1 & 0 & 0 & 0 & 1 \\
  1 & 0 & 0 & 1 & 1 \\
  1 & 0 & 1 & 0 & 1 \\
  1 & 0 & 1 & 1 & 1 \\
  1 & 1 & 0 & 0 & 0 \\
  1 & 1 & 0 & 1 & 1 \\
  1 & 1 & 1 & 0 & 1 \\
  1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

\[F(A,B,C,D) = BD + A'B'C' + ABC + A'CD + AC'D\]

\[F = ab' + ac' + abc\]

\[F = (a + b')(a + c')(a + b + c)\]
F = \( (a + b') (a + c') (a + b + c) \)

Homework 2
- Boolean Properties and Laws
- FOIL (2 applications of Distribution)
- SUM of PRODUCTS form \( ab + bc \)
- PRODUCT of SUMS form \( b (a + c) \)
- SUM of MINTERMS form each product must use all literals

Simplified form has the fewest literals and the fewest operations
\[ ab + bc \quad == \quad b (a+c) \]

K-Maps
- normally want the largest groupings and the fewest grouping
- beware of "across the edges" groupings

\[ F(A,B,C,D) = BD + A'BC' + ABC + A'CD + AC'D \]

UNIVERAL Gates
- NAND and NOR
- Sum of PRODUCTs => NAND
- Product of SUMS => NOR

\[ F = ab' + ac' + abc \]
F = (a + b') (a + c') (a + b + c)

Rapid Prototyping
- Decoders
- Multiplexors

F(A,B,C,D) = \sum m(0,2,6,7,8,10,11,13)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>