CS240 Laboratory 5
Electrical and Digital Laboratory Concepts

• Basic Concepts of Electricity
• Transistors
• Circuit Equivalence
• Universal Gates
• Integrated Circuits
• Tools and Techniques for Building and Simulating Circuits
Electricity = the movement of electrons in a material

Materials tend to have a net negative or positive charge

Difference of charge between two points = potential difference (V)

Rate at which electrons flow through = current (A).

Ease of conduction, or current flow = resistance (Ω)

Ohm's Law, $V = IR$. 
**Open circuit** = no current

![Diagram of an open circuit](image)

**Short circuit** = infinite current, since \( V/0 = \text{infinite current} \):

![Diagram of a short circuit](image)

**Infinite current** swiftly results in the destruction of the circuit!
### Resistor Color Codes

<table>
<thead>
<tr>
<th>color</th>
<th>digit</th>
<th>multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Brown</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Red</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>Orange</td>
<td>3</td>
<td>1000 (1K)</td>
</tr>
<tr>
<td>Yellow</td>
<td>4</td>
<td>10000 (10K)</td>
</tr>
<tr>
<td>Green</td>
<td>5</td>
<td>100000 (100K)</td>
</tr>
<tr>
<td>Blue</td>
<td>6</td>
<td>1000000 (1M)</td>
</tr>
<tr>
<td>Violet</td>
<td>7</td>
<td>10000000 (10M)</td>
</tr>
<tr>
<td>Gray</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>White</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>
Transistors

**NOT**
Uses one transistor

**NAND**
Uses two transistors

**AND**
Uses three transistors

Voltage levels can be interpreted as 1s and 0s: all possible inputs and outputs of a circuit can be described using the binary number system.

**Positive Logic**
High voltage (2 – 5 volts) = 1
Low voltage (0 – 1 volts) = 0

**Negative Logic** is the opposite assignment

Voltage levels between 1 and 2 volts will cause unpredictable results and must be avoided.
Circuit Equivalence

Two boolean functions with same truth table = equivalent

When there is an equivalent circuit which uses fewer gates, transistors, or chips, it is preferable to use that circuit in the design

Example:
Given: $F = A'B' + A'B$ \quad $Q = A' + A'B + A'B'$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A'B'</th>
<th>A'B</th>
<th>F</th>
<th>A</th>
<th>B</th>
<th>A'</th>
<th>A'B</th>
<th>A' B'</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

F and Q are equivalent because they have the same truth table.
Identities of Boolean Algebra

- Identity law $1A = A \quad 0 + A = A$
- Null law $0A = 0 \quad 1 + A = 1$
- Idempotent law $AA = A \quad A + A = A$
- Inverse law $AA' = 0 \quad A + A' = 1$
- Commutative law $AB = BA \quad A + B = B + A$
- Associative law $(AB)C = A(BC) \quad (A + B) + C = A + (B + C)$
- Distributive law $A + BC = (A + B)(A + C) \quad A(B + C) = AB + AC$
- Absorption law $A(A + B) = A \quad A + AB = A$
- De Morgan's law $(AB)' = A' + B' \quad (A + B)' = A'B'$

Example:

$F = A'B' + A'B$  
$= A'(B' + B)$ distributive  
$= A'(1)$ inverse  
$= A'$ identity

$Q = A' + A'B + A'B'$  
$= A'(1 + B + B')$ distributive  
$= A' (1)$ null  
$= A'$ identity
Universal Gates

Any Boolean function can be constructed with NOT, AND, and OR gates

NAND and NOR = universal gates

DeMorgan’s Law shows how to make AND from NOR (and vice-versa)

\[
AB = (A' + B')' \quad \text{(AND from NOR)}
\]
\[
A + B = (A'B')' \quad \text{(OR from NAND)}
\]

NOT from a NOR

OR from a NOR

To implement a function using only NOR gates:

- apply DeMorgan's Law to each AND in the expression until all ANDs are converted to NORs
- use a NOR gate for any NOT gates, as well.
- remove any redundant gates (NOT NOT, may remove both)

Implementing the circuit using only NAND gates is similar.
Example:  \( Q = (AB)'B' \)

\[
= (A' + B')B' \\
= ((A'+B')' + B)' \quad \text{(NOR gates only, since NOR can be used as a NOT gate)}
\]

**Simplifying Circuits or Proving Equivalency**

General rule to simplify circuits or prove equivalency:

1. Distribute if possible, and if you can’t, apply DeMorgan’s Law so that you can.
2. Apply other identities to remove terms, and repeat step 1.

**EXAMPLE:** Is \((A’B’)(AB)’ + A’B’\) equivalent to \((AB)’\)?

\[
F = (A’B’)(AB)’ + A’B’ \quad -- \text{can’t distribute} \\
= (A + B’) (A’ + B’) + A’B’ \quad \text{DeMorgan’s} \\
= AA’ + AB’ + A’B + B’B’ + A’B’ \quad \text{distributive} \\
= 0 + AB’ + A’B + B’ + A’B’ \quad \text{inverse and idempotent} \\
= AB’ + A’B + A’B’ \quad \text{identity} \\
= B’ (A+ A’) + A’B \quad \text{distributive} \\
= B’(1) + A’B \quad \text{inverse} \\
= B’ + A’B \quad \text{identity} \\
= B’ + (A + B’)’ \quad \text{DeMorgan’s} \\
= (B(A + B’)’) \quad \text{DeMorgan’s} \\
= (AB + BB’)’ \quad \text{distributive} \\
= (AB + 1)’ \quad \text{inverse} \\
= (AB)’ \quad \text{identity}
\]
Integrated Circuits

**Logic diagrams** are not the same as pin-outs! Show information about the logical operation of the device.

Pin-Out (found in TTL Data Book or online) show the physical layout of the pins:

- **Top left** pin is pin 1, always to left of notch in chip, and often marked with a dot
- **Bottom left** pin is almost always connected to ground (0V)
- **Top right** pin is almost always connected to Vcc (+5V)
- The chip will not work if it is not connected to power and ground!
Breadboard for wiring circuits
Tool to create prototype circuits before manufacturing PCB

An array of holes in which wires or component leads can easily be inserted

All holes in a row internally connected (use to tie one point to another in the circuit)

Use .22 gauge wires with 1/4” of insulation stripped from both ends
Insert chips straddling the groove
PB-503 Protoboard
Testing and Debugging

To test, check the output as the inputs are switched through all possible combinations

**Problem:** wrong output for a particular combination of inputs

**Possible causes**

1. Power and/or ground not connected
2. Wrong pin connections
3. Wrong gate/chip
4. Bad gate/chip
5. Faulty design

To avoid problems, use a systematic approach to design and implementation!
Debugging Strategies

Following a systematic process to debug and correct a circuit is preferable (and often much less frustrating) than simply tearing out the circuit and starting again.

**Logic probe:** tool for measuring outputs

1. Check power and ground for each chip
2. Check outputs and trace back to problem
3. If output is incorrect:
   a. Check pin numbers
   b. Check chip number
   c. Bad connection (2 outputs wired together) may need to replace the chips
   d. Bad gate
4. Re-check your design if no problem was found

Once a problem has been corrected, re-test
Circuit Simulation/LogicWorks