Module 2: Coming up...

- Pre-class quiz #3: due Wednesday September 23\textsuperscript{rd} at 19:00.
  - Assigned reading for the quiz:
    - Epp, 5\textsuperscript{th} edition: 2.5
    - Epp, 4\textsuperscript{th} edition: 2.5 plus
      - \textcolor{red}{\url{http://en.wikipedia.org/wiki/Binary_numeral_system}}
      - \textcolor{red}{\url{http://www.ugrad.cs.ubc.ca/~cs121/current/handouts/signed-binary-decimal-conversions.html}}
  - Assignment #1 is due Monday September 28\textsuperscript{th} at 19:00.

Module 2: Coming up...

- Pre-class quiz #4: tentatively due Monday September 28\textsuperscript{th} at 19:00.
  - Assigned reading for the quiz:
    - Epp, 5\textsuperscript{th} or 4\textsuperscript{th} edition: 2.3
    - Epp, 3\textsuperscript{rd} edition: 1.3
    - Rosen, 6\textsuperscript{th} edition: 1.5 up to the bottom of page 69.
    - Rosen, 7\textsuperscript{th} edition: 1.6 up to the bottom of page 75.

Module 2: Coming up...

- Pre-class quiz #2: very well done except for question 3:
  - Which of the following has the same meaning as $p \rightarrow \neg q$?
    - a) Anytime that $p$ is true, $q$ must be false.
    - b) $p$ can not be true unless $q$ is false.
    - c) $q$ can not be false unless $p$ is true.
    - d) if $p$ is true then $q$ is false.
    - e) $q$ and $p$ can never have the same truth value (both true or both false).
Module 2: Conditionals and Logical Equivalences

- By the start of this class you should be able to
  - Translate back and forth between simple natural language statements and propositional logic, now with conditionals and biconditionals.
  - Evaluate the truth of propositional logical statements that include conditionals and biconditionals using truth tables.
  - Given a propositional logic statement and an equivalence rule, apply the rule to create an equivalent statement.

- By the end of this unit, you should be able to
  - Explore alternate forms of propositional logic statements by application of equivalence rules, especially in order to simplify complex statements or massage statements into a desired form.
  - Evaluate propositional logic as a “model of computation” for combinational circuits, including at least one explicit shortfall (e.g., referencing gate delays, fan-out, transistor count, wire length, instabilities, shared sub-circuits, etc.).

Module Outline:

- Logic vs Everyday English
- Logical Equivalence Proofs
- Multiplexers
- More exercises
Module 2.1: Logic vs Everyday English

- Be careful!
- The meaning of if p then q in propositional logic is not quite the same as in normal language.
- Suppose that I state
  
  \[ p: \text{If you rob a bank, then you will go to jail} \]
- You need to distinguish between
  - The truth value of \( p \) (whether or not I lied).
  - The truth value of the conclusion (whether or not you will go to jail).

If you rob a bank, will you go to jail?

a) Yes
b) No
c) Maybe

If you go to jail, have you robbed a bank?

a) Yes
b) No
c) Maybe

Module 2: Conditionals and Logical Equivalences

- Module Outline:
  - Logic vs Everyday English
  - Logical Equivalence Proofs
  - Multiplexers
  - More exercises
Module 2.2: Logical Equivalence Proofs

How do we write a logical equivalence proof?
- We state the theorem we want to prove.
- We indicate the beginning of the proof by **Proof:**
- We start with one side and work towards the other,
  - one step at a time,
  - without forgetting to justify each step
  - usually we will simplify the more complicated proposition, instead of trying to complicate the simpler one.
- We indicate the end of the proof by QED or □

Example: prove that \((\neg a \land b) \lor a \equiv a \lor b\)

**Proof:**
\[
(\neg a \land b) \lor a \equiv a \lor (\neg a \land b) \quad \text{commutative law}
\equiv (a \lor \neg a) \land (a \lor b) \quad \text{distributive law}
\equiv \quad \text{__________} \quad \text{identity law}
\equiv a \lor b
\]

What is missing?
- a) \((a \lor b)\)
- b) \(F \land (a \lor b)\)
- c) \(a \land (a \lor b)\)
- d) Something else
- e) Not enough info to tell

### Worksheet: prove the following
- \(p \land (p \lor r \lor s) \equiv \neg p \rightarrow (p \land r \land s)\)
- \(\neg p \land q \equiv (\neg p \lor q) \land \neg (\neg q \lor p)\)
Module 2: Conditionals and Logical Equivalences

- Module Outline:
  - Logic vs Everyday English
  - Logical Equivalence Proofs
  - Multiplexers
  - More exercises

Module 2.3: Multiplexers

- Propositional Logic is not a perfect model of how gates work.
- To understand why, we will look at a multiplexer.
  - A circuit that chooses between two or more values.
  - In its simplest form, it takes 3 inputs
    - An input $a$, an input $b$, and a control input $select$.
    - It outputs $a$ if $select$ is false, and $b$ if $select$ is true.

Truth table:

<table>
<thead>
<tr>
<th>$a$</th>
<th>$b$</th>
<th>$select$</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
</tr>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
</tbody>
</table>

Here is one possible implementation:

Let us see why this may not work as we expect...
Module 2.3: Multiplexers

- Fact: gates are not instantaneous
  - If we change the input of a gate at time $t = 0$.
  - The output of the gate will only reflect the change some time later.
  - This time gap is called the gate delay.

Suppose $a, b, \text{select}$ are initially $T$
Assume the gate delay is $10\text{ns}$

How long will it take before output reflects any changes in $a, b, \text{select}$?

a) $10\text{ns}$
b) $20\text{ns}$
c) $30\text{ns}$
d) $40\text{ns}$
e) It may never happen.

We switch $\text{select}$ to $F$ at time $0\text{ns}$. At time $5\text{ns}$:
Module 2.3: Multiplexers

- At time 10ns:

- At time 20ns:

  - Note: the output is now F

- At time 30ns:

  - Note: the output is now T again.

- So the output changed from T (old output) to F and then to T (new output).

- This is called an instability.

- The cause of the problem:
  - the information from select travels on two different paths to the output
  - these paths contain different numbers of gates
  - so the shorter path may affect the output until the information on the longer path catches up.
Module 2.3: Multiplexers

- Which one(s) of the following operation may cause an instability?
  
a) Changing \( a \) or \( b \) only
b) Changing \( \text{select} \), when at exactly one of \( a, b \) is F
c) Changing \( \text{select} \), when both \( a, b \) are F
d) Both (a) and (b)
e) None of (a), (b) or (c).

Here is a multiplexer that avoid the instability:

Module 2: Conditionals and Logical Equivalences

- Module Outline:
  
  - Logic vs Everyday English
  - Logical Equivalence Proofs
  - Multiplexers
  - More exercises

Module 2.4: More exercises

- Consider the code:
  
  ```
  if target = value then
    if lean-left-mode = true then
      call the go-left() routine
    else
      call the go-right() routine
  else if target < value then
    call the go-left() routine
  else
    call the go-right routine
  ```

- Let \( gl \) mean “the go-left() routine is called”. Complete the following:

  \( gl \leftrightarrow \)
Module 2.4: More exercises

Consider:
The Java [String] equals() method returns true if and only if the argument is not null and is a String object that represents the same sequence of characters as this object.

- Let
  - n1: the string is null
  - n2: the argument is null
  - nt: the method returns true
  - s: the two objects are strings that represent the same sequence of characters.

Is the sentence logically equivalent to \( nt \leftrightarrow (n1 \land n2) \lor s \)? Why or why not?

Prove:
\[
(a \land \neg b) \lor (\neg a \land b) \equiv (a \lor b) \land \neg (a \land b)
\]