COMP278
Week 2

Joey Lawrance

http://bit.ly/WITDoNow

Answer these at the URL above.

X	Y	1.	2.	3.
0	0	1	0	0
0	1	1	1	0
1	0	1	0	0
1	1	0	0	1

A. XY B. X'Y C. X+Y D. (XY)'

Volunteers?

Who wants to volunteer to record minutes?

It's extra credit…

We need a record of Q&A at:
http://bit.ly/COMP27805Minutes

Boolean expressions

- We can use these basic operations to form more complex expressions:

  f(x,y,z) = (x + y')z + x'

- Some terminology and notation:
  - f is the name of the function.
  - (x,y,z) are the input variables, each representing 1 or 0. Listing the inputs is optional, but sometimes helpful.
  - A literal is any occurrence of an input variable or its complement. The function above has four literals: x, y', z, and x'.
- Precedences are important, but not too difficult.
  - NOT has the highest precedence, followed by AND, and then OR.
  - Fully parenthesized, the function above would be kind of messy:

  f(x,y,z) = (((x +(y'))z) + x')

Truth tables

- **A truth table shows all possible inputs and outputs of a function.**
- **Remember that each input variable represents either 1 or 0.**
  - **Because there are only a finite number of values (1 and 0), truth tables themselves are finite.**
  - **A function with n variables has 2^n possible combinations of inputs.**
- **Inputs are listed in binary order—in this example, from 000 to 111.**

f(x,y,z) = (x + y')z + x'

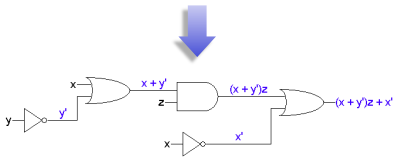
f(0,0,0) = (0 + 1)0 + 1 = 1
f(0,0,1) = (0 + 1)1 + 1 = 1
f(0,1,0) = (0 + 0)0 + 1 = 1
f(0,1,1) = (0 + 0)1 + 1 = 1
f(1,0,0) = (1 + 1)0 + 0 = 0
f(1,0,1) = (1 + 1)1 + 0 = 1
f(1,1,0) = (1 + 0)0 + 0 = 0
f(1,1,1) = (1 + 0)1 + 0 = 1

x	y	z	f(x,y,z)
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Expressions and circuits

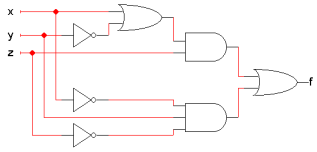
- ***Any* Boolean expression can be converted into a circuit by combining basic gates in a relatively straightforward way.**
- **The diagram below shows the inputs and outputs of each gate.**
- **The precedences are explicit in a circuit. Clearly, we have to make sure that the hardware does operations in the right order!**

(x + y')z + x'



Circuit analysis

- Circuit analysis involves figuring out what some circuit does.
  - Every circuit computes some function, which can be described with Boolean expressions or truth tables.
  - So, the goal is to find an expression or truth table for the circuit.
- The first thing to do is figure out what the inputs and outputs of the overall circuit are.
  - This step is often overlooked!
  - The example circuit here has *three* inputs x, y, z and *one* output f.



http://bit.ly/WITAttendance

Answer these at the URL above.

1. What is 0b1000 in decimal?
   A. 1000 B. 1 C. 4 D. 8 E. 16
2. How many 3-bit numbers are there?
   A. 16 B. 8 C. 6 D. 4 E. 3
3. What is 0b0.1 in decimal?
   A. 0.5 B. 0.2 C. 0.1 D. 0.05 E. 0.02

Volunteers?

Who wants to volunteer to record minutes?

Participation is 15 points, Note taking is 20 points

We need a record of Q&A at:
http://bit.ly/COMP27801Minutes

http://bit.ly/WITDoNow

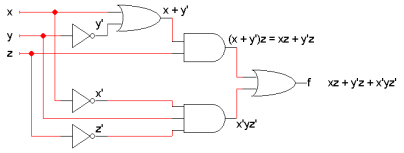
Answer these at the URL above.

Write out the truth table for X XOR Y.

1. X XOR Y is equivalent to which function?
   A. XY+X'Y' B. X'Y+Y'X C. XY'+(XY)' D. X+Y
2. Rotate the truth table counter-clockwise. Which of these is X XOR Y?
   A. 0001 B. 0100 C. 1100 D. 1001 E. 0110
3. There are _____ binary functions of two inputs.
   A. 3 B. 5 C. 16 D. Infinite

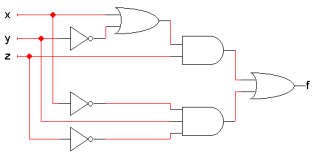
Write algebraic expressions...

- Next, write expressions for the outputs of each individual gate, based on that gate's inputs.
  - Start from the inputs and work towards the outputs.
  - It might help to do some algebraic simplification along the way.
- Here is the example again.
  - We did a little simplification for the top AND gate.
  - You can see the circuit computes f(x,y,z) = xz + y'z + x'yz'



...or make a truth table

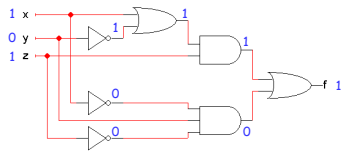
- It's also possible to find a truth table directly from the circuit.
- Once you know the number of inputs and outputs, list all the possible input combinations in your truth table.
  - A circuit with n inputs should have a truth table with 2^n rows.
  - Our example has three inputs, so the truth table will have $2^3 = 8$ rows. All the possible input combinations are shown.



x	y	z	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Simulating the circuit

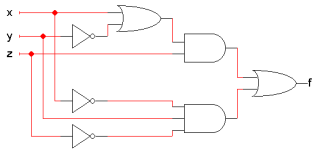
- **Then you can simulate the circuit, either by hand or with a program like LogicWorks, to find the output for each possible combination of inputs.**
- **For example, when xyz = 101, the gate outputs would be as shown below.**
  - **Use truth tables for AND, OR and NOT to find the gate outputs.**
  - **For the final output, we find that f(1,0,1) = 1.**



x	y	z	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	1
1	1	0	
1	1	1	

Finishing the truth table

- **Doing the same thing for all the other input combinations yields the complete truth table.**
- **This is simple, but tedious.**



x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

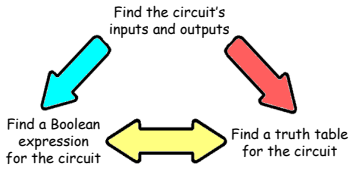
Expressions and truth tables

- Remember that if you already have a Boolean expression, you can use that to easily make a truth table.
- For example, since we already found that the circuit computes the function f (x,y,z) = xz + y'z + x'yz', we can use that to fill in a table:
  - We show intermediate columns for the terms xz, y'z and x'yz'.
  - Then, f is obtained by just OR'ing the intermediate columns.

x	y	z	xz	y'z	x'yz'	f
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	0	0	1	1
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	1	1	0	1
1	1	0	0	0	0	0
1	1	1	1	0	0	1

Circuit analysis summary

- After finding the circuit inputs and outputs, you can come up with either an expression or a truth table to describe what the circuit does.
- You can easily convert between expressions and truth tables.



Expression simplification

- Normal mathematical expressions can be simplified using the laws of algebra
- For binary systems, we can use Boolean algebra, which is superficially similar to regular algebra
- There are many differences, due to
  - having only two values (0 and 1) to work with
  - having a complement operation
  - the OR operation is not the same as addition

Boolean Operator Precedence

- **The order of evaluation in a Boolean expression is:**
  1. Parentheses
  2. NOT
  3. AND
  4. OR
- **Consequence: Parentheses appear around OR expressions**
- **Example:** F = A(B + C)(C + D)

Formal definition of Boolean algebra

- A Boolean algebra requires
  - A set of elements **B**, which needs *at least* two elements (0 and 1)
  - Two binary (two-argument) operations OR and AND
  - A unary (one-argument) operation NOT

Boolean Algebra

1. $X + 0 =$	2. $X \cdot 1 =$	Existence of 0 and 1
3. $X + 1 =$	4. $X \cdot 0 =$	
5. $X + X =$	6. $X \cdot X =$	Idempotence
7. $X + \overline{X} =$	8. $X \cdot \overline{X} =$	Existence of complement
9. $\overline{\overline{X}} =$		Involution
10. $X + Y =$	11. $XY =$	Commutative
12. $(X + Y) + Z =$	13. $(XY)\ Z =$	Associative
14. $X(Y + Z) =$	15. $X + YZ =$	Distributive
16. $\overline{X + Y} =$	17. $\overline{X \cdot Y} =$	DeMorgan's

http://bit.ly/WITAttendance

Answer these at the URL above.

Write the truth table for X' + X.

1. This is equivalent to which function?
   A. 0 B. 1 C. X D. X'
2. Rotate the truth table counter-clockwise. Which of these numbers represents X' + X?
   A. 0 B. 1 C. 2 D. 3
3. There are ___ binary functions with 1 input.
   A. 1 B. 2 C. 3 D. 4

http://bit.ly/WITDoNow

Answer these at the URL above.

1. X NAND Y reads as what from top to bottom?
   A. 1000 B. 1100 C. 1110 D. 0111
2. What is X NAND Y in terms of AND, OR or NOT?
   A. XY+Y B. (XY)' C. (X+Y)' D. X'Y'
3. X NAND X reads as what from top to bottom?
   A. 00 B. 01 C. 10 D. 11