CHAPTER 9

BOOLEAN ALGEBRAS AND COMBINATORIAL CIRCUITS

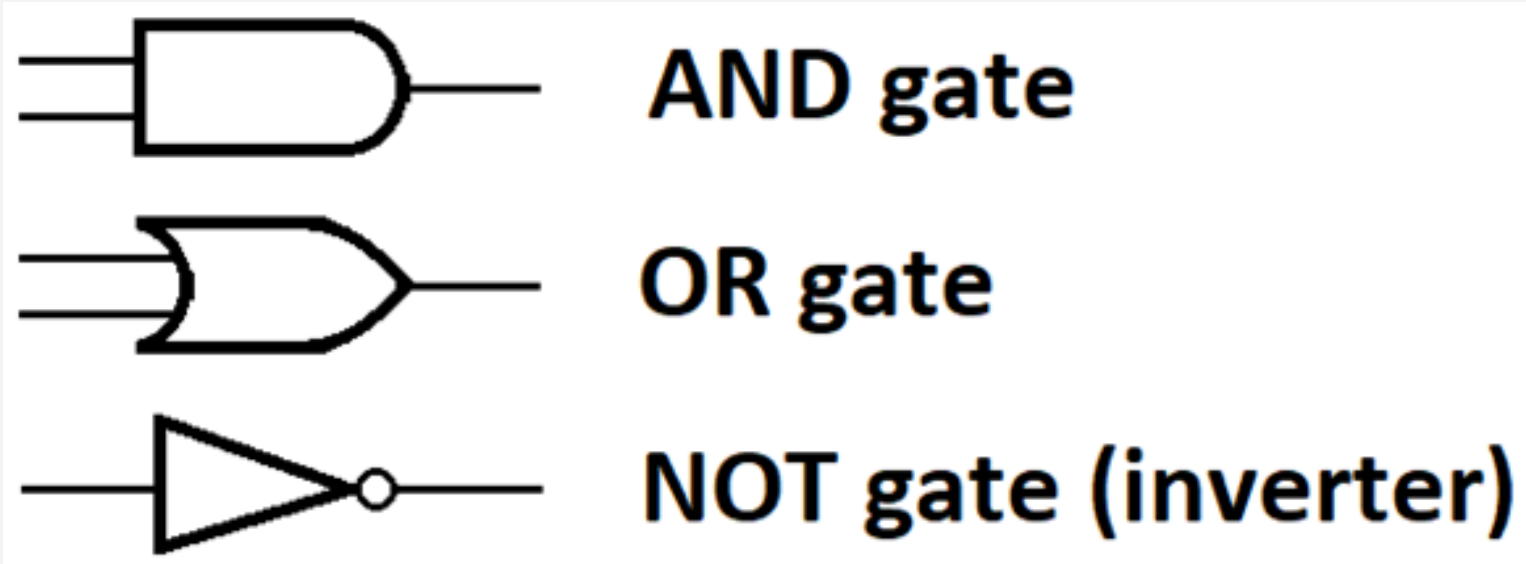
LOGIC (COMBINATORIAL) CIRCUITS

- Now we will view logic from the hardware's viewpoint
- We will also look at other similar ways to design circuits
- Taking the hardware's viewpoint
- From now on, everything is 0 or 1

  0 represents false

  1 represents true

BASIC GATES



LOGIC TABLES

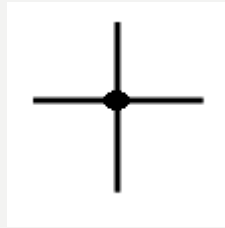
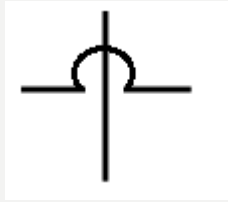
- These gates work like our old logic
- A truth table will now be called a logic table
- We use 0, 1 instead of false, true
- For example, here is the logic table for AND

p	q	P AND q
0	0	0
0	1	0
1	0	0
1	1	1

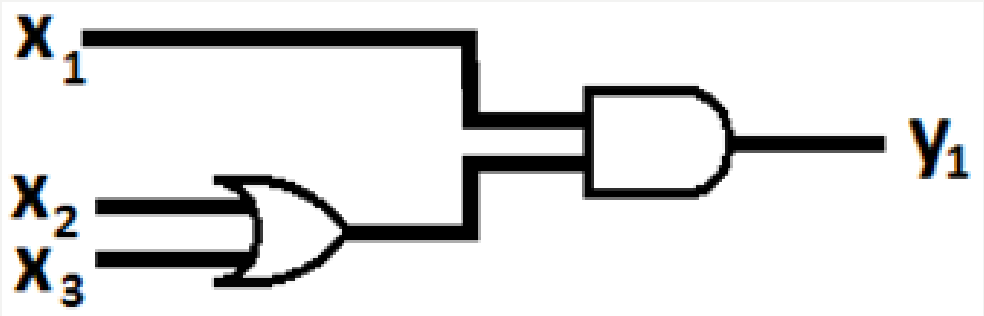
COMBINATORIAL CIRCUITS

- The key kind of circuit we are concerned with is a combinatorial circuit
- This is a circuit in which an output is defined for each combination of inputs
- Combinatorial circuits can be built from AND, OR, NOT

PICTURES WITH WIRES

- Here is a picture of two wires 
- This is ambiguous; are these wires connected or is one just on top of the other?
- If we mean that the wires are connected, we will write it as (with a dot) 
- If we mean that the wires are just one on top of another, we will write it as 

A PICTURE OF A CIRCUIT



LET'S CREATE A LOGIC TABLE FOR IT

- We create a table with columns for each input and each intermediate step
- We then fill it in
- For example, we have columns for X1, X2, X3, X2 OR X3, X1 AND (X2 or X3)
- Then we fill it in

THE LOGIC TABLE FOR THE CIRCUIT

X1	X2	X3	X2 OR X3	X1 AND (X2 OR x3)
0	0	0	0	0
0	0	I	I	0
0	I	0	I	0
0	I	I	I	0
I	0	0	0	0
I	0	I	I	I
I	I	0	I	I
I	I	I	I	I

WRITING A BOOLEAN EXPRESSION

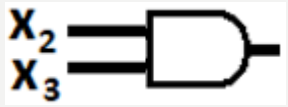
- We could write a Boolean expression for the circuit
- We work backward from the output
- Y1 = _____ AND _____
- Y1 = X1 AND _____
- Y1 = X1 AND (X2 OR X3)

A BOOLEAN EXPRESSION

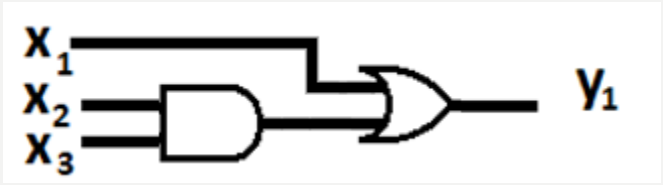
- What is a Boolean expression?
  - I used that term on the previous slide
  - I assumed we would guess at he meaning
- Here is a formal definition
  - 0 and 1 are Boolean expressions
    - X1, X2, …, Xn are Boolean expressions
  - If (BE) is a Boolean expression
    - NOT (BE) is a Boolean expression
    - BE1 AND BE2 is a Boolean expression
    - BE1 OR BE2 is a Boolean expression
- This definition is recursive?

CREATING A CIRCUIT FROM A BOOLEAN EXPRESSION

- To do this, you first work with what's inside parentheses, and then work outward
- Suppose the expression is Y1 = X1 OR(X2 AND X3)
- We draw the circuit
- Start with X2 AND X3



- Then connect in X1 or <that result>



SWITCHING CIRCUITS

- A switching circuit is a circuit made only of switches
- I will usually add a light bulb and a battery to show the point of the circuit
  - We do not draw these in an actual circuit
- In a switching circuit
  - You can use a letter more than once
    - But, if you use A in two different places, that means those two switches move together
    - And, if you use A and -A, that means those two switches move oppositely
- Do #25, p. 422

EQUIVALENT CIRCUITS

- Equivalent circuits are circuits where the same input values give the same outputs
- An example is (X1 OR X2) AND X3 and (X1 AND X2) OR X3
- Are these two circuits equivalent?

LAWS OF CIRCUITS- PART 1

- Associative laws
  - (a OR b) OR c = a OR (b OR c)
  - (a AND? b) AND c = a AND (b AND c)
- Commutative Laws
  - a OR b = b OR a
  - a AND b = b AND a
- Distributive Laws
  - a AND (b OR c) = (a AND b) OR (a AND c)
  - a OR (b AND c) = (a OR b) AND (a OR c)

LAWS OF CIRCUITS- PART 2

- Identity Laws
  - x OR 0 = x
  - x AND 1 = x
  - The identity laws are quite useful in assembly language programming
- Complement Laws
  - x OR x' = 1
  - x AND (x') = 0