CMSC250
Fall 2018
Circuits

Logic == Math?

What calculations can we do with logic?

Add, subtract, multiply?

George Boole – 1800's. Boolean logic

Claude Shannon – 1937. Logic == circuits == math

T = 1 p v q == p+q p v ~q

F = 0 p ^ q == p*q is

~ p == (1-q) p + (1-q)

Find Boolean formula for:

- p, q & r are the variables.

p	q	r	output
1	1	1	1
1	1	0	1
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	0
0	0	1	0

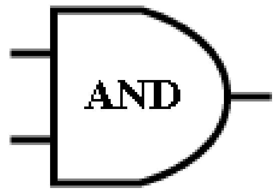
Find Boolean Formula

- For each row with output 1 obtain “mini-formula” which is 1 exactly on that row.
- OR together all of the mini-formulas

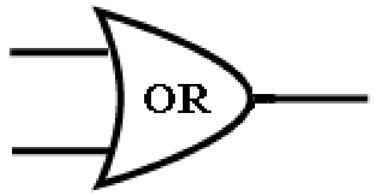
- 111, 110, and 100 all output 1
- p^q^r, p^q^~r, p^~q^~r
- (p^q^r)˅(p^q^~r)˅(p^~q^~r)

Basic logic gates

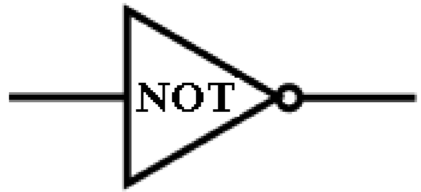
- AND gate:



- OR gate:

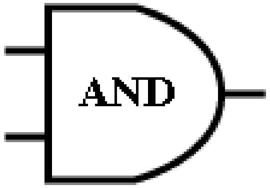


- NOT gate:

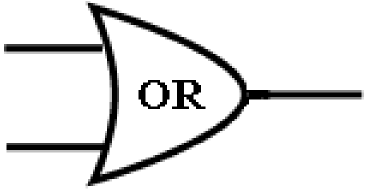


Basic logic gates

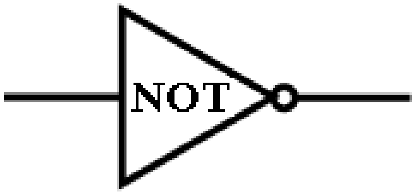
- AND gate:



- OR gate:

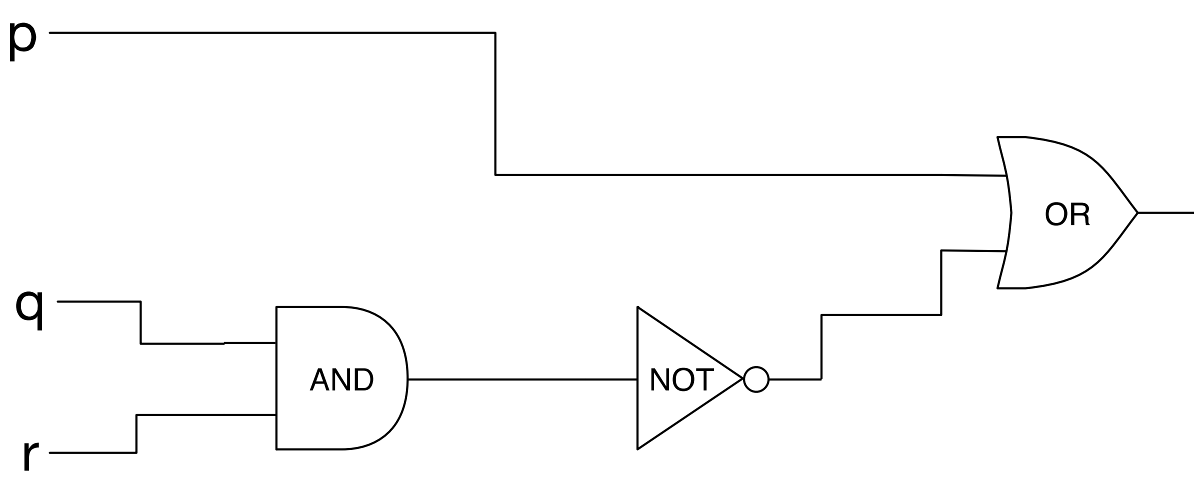


- NOT gate:



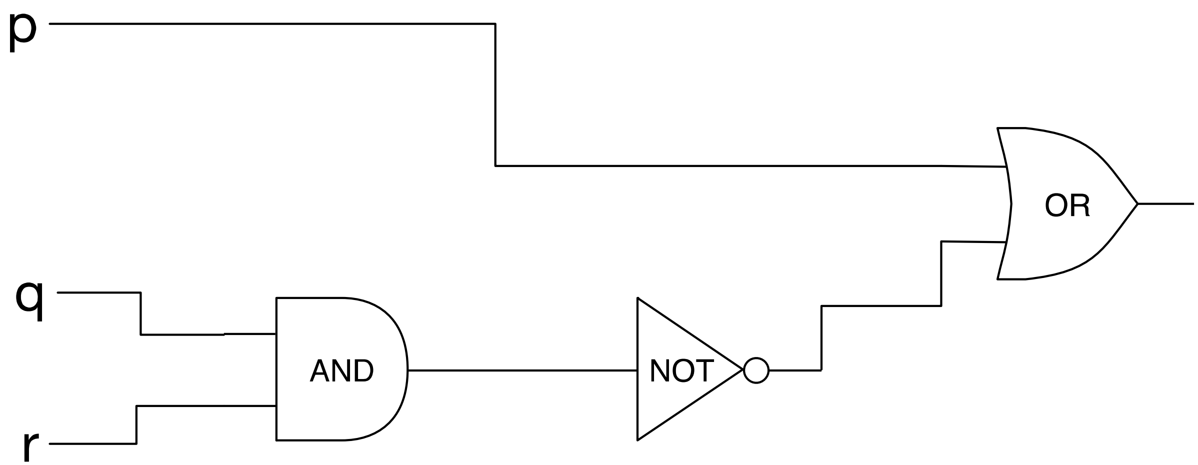
Circuit to Boolean formula

Given this circuit, convert to logic



Circuit to Boolean formula

Given this circuit, convert to logic



p v ~(p ^ r)

Draw a circuit for:

- p, q & r are inputs.
- Simplify before building the circuit.

p	q	r	output
1	1	1	1
1	1	0	1
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	0
0	0	1	0

Number conversions

- Different number system bases are used when convenient
  - some commonly-used bases are 10 (decimal), 2 (binary), 8 (octal), 16 (hexadecimal)
  - the base tells how many different numerals are used
  - the base also determines the value of each place
- Conversions from anything to base 10
  - use the definition of the number system
- Conversions from base 10 to anything
  - use repeated integer division

Addition of binary numbers

- Carry if the number would be too large for the number system- if it is greater than 1

	1001	1001	1011	1101
+	10	11	10	111
	1011	1100	1101	10100

Addition of octal and hexadecimal numbers

- Carry if the number would be too large for the number system (larger than 7 or 15)

$$\begin{array}{r} 723_8 \\ +\quad 12_8 \\ \hline 735_8 \end{array} \qquad \begin{array}{r} 265_8 \\ +\quad 33_8 \\ \hline 320_8 \end{array} \qquad \begin{array}{r} ABC_{16} \\ +\quad 12_{16} \\ \hline ACE_{16} \end{array} \qquad \begin{array}{r} CDE_{16} \\ +\quad ED_{16} \\ \hline DCB_{16} \end{array}$$

Two's complement

- To represent negative values in binary:
  1. Find the binary equivalent of the absolute value.
  2. Pad on the left to completely fill the bits in the specified bit width
  3. Switch all of the 1's to 0's and 0's to 1's.
  4. Add 1 to the result.

- Example: find the 8-bit two's complement representation of -43:
  1. $43_{10} = 101011_2$
  2. 00101011_2
  3. 11010100_2
  4. $11010101_2 = -43_{10}$

Using a circuit for adding two bits

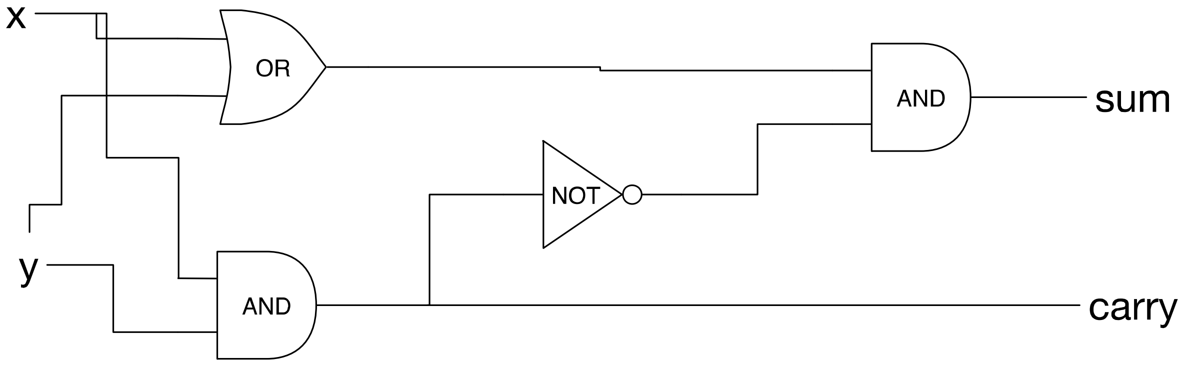
- Write as a logic expression
- Translate to circuits

input		output	
p	q	carry	sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Half adder

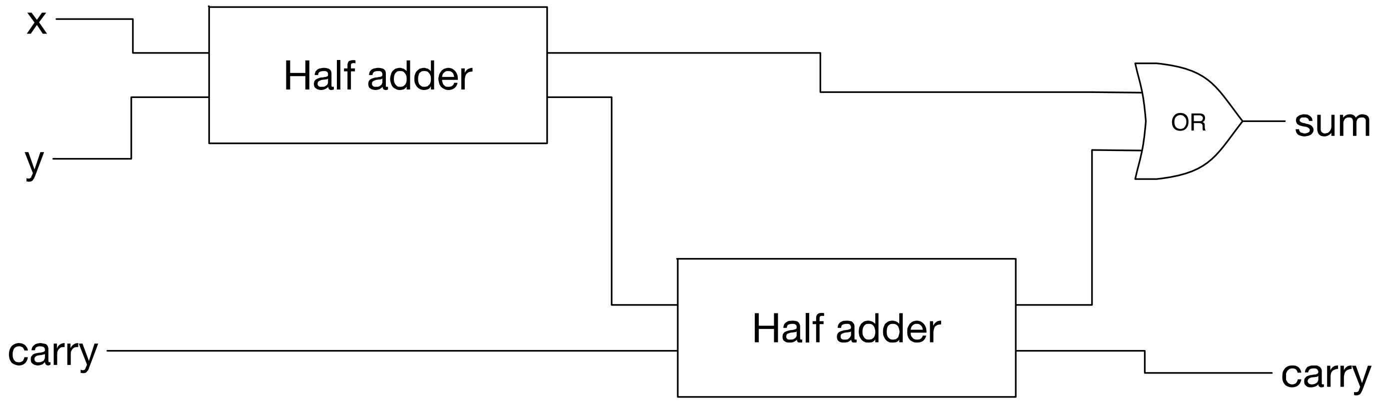
Sum = (x v y) ^ ~ (x ^ y)

Carry = (x ^ y)



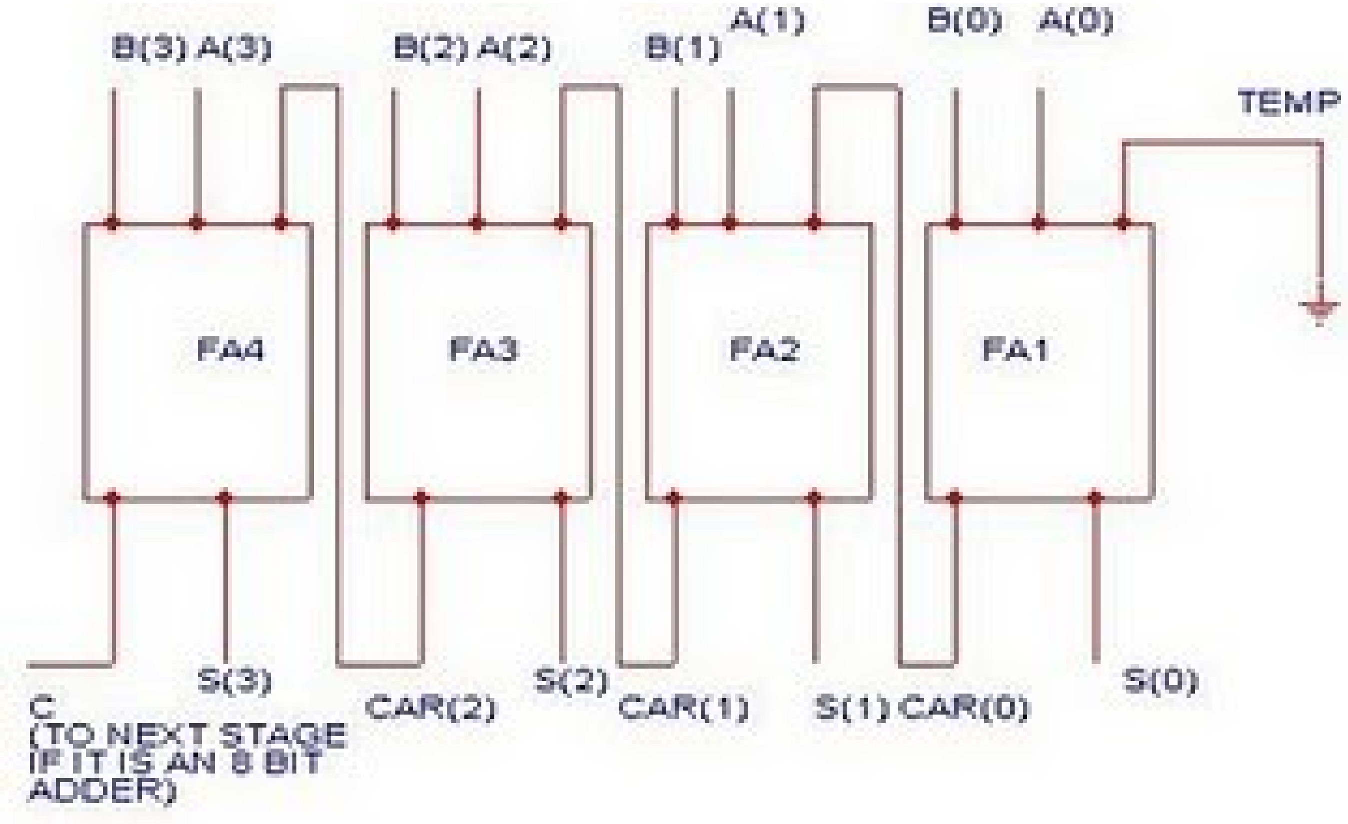
Full adder

Three bits in (x, y, previous carry)



Parallel adders

- Chain these half adders and full adders together for multi-bit addition
- $A_3A_2A_1A_0 + B_3B_2B_1B_0 = S_3S_2S_1S_0$



Topic not covered

Simplifying circuits: there are techniques that exist (which are complex).