CMSC 250
Discrete Structures

Digital Circuits

Number Bases

Definition

What are number bases?

Number Bases

Definition

What are number bases?

How do we convert a number from an arbitrary base into base 10?

Number Bases

Definition

What are number bases?

How do we convert a number from an arbitrary base into base 10?

How do we convert a number from base 10 into an arbitrary base?

Number Bases

Definition

What are number bases?

How do we convert a number from an arbitrary base into base 10?

How do we convert a number from base 10 into an arbitrary base?

We are mostly concerned with base 10(decimal) and base 2 (binary).

Truth Values and Bits

Standard Convention: TRUE is 1, FALSE is 0

Standard Convention: n-inputs viewed as n-bit number.

Truth Values and Bits

Standard Convention: TRUE is 1, FALSE is 0
Standard Convention: n-inputs viewed as n-bit number.

Propositional formula is now *Boolean formula*

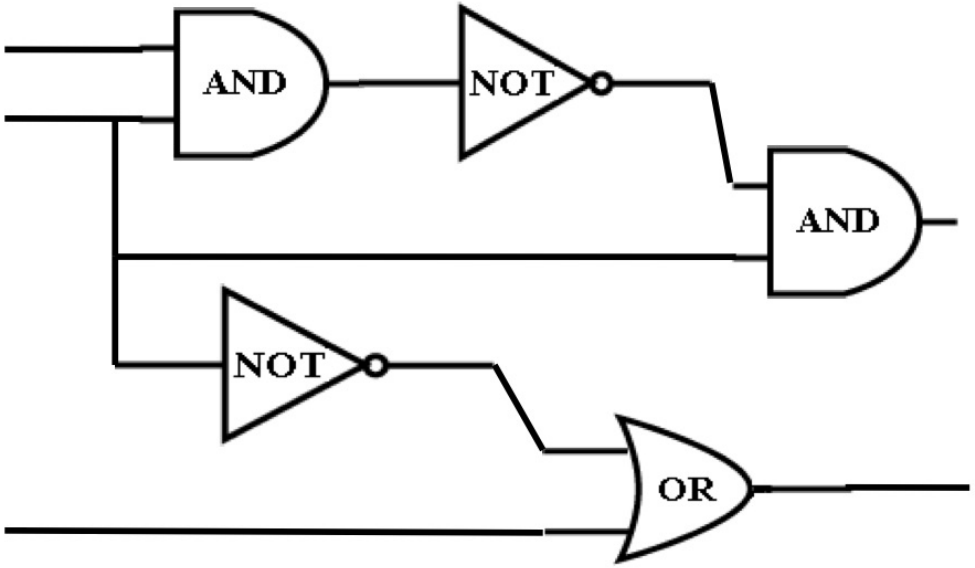
Basic Logic Gates

- An *AND* gate
- An *OR* gate
- A *NOT* gate

Digital Circuits

Definition

A circuit has n Boolean inputs which feed into gates and at the end have one or more outputs.



New problem

- Old Problem: Given a formula, find the truth table for it.

Example

$(p \vee q) \wedge \sim(p \wedge q)$

New problem

- Old Problem: Given a formula, find the truth table for it.

Example

$(p \vee q) \wedge \sim(p \wedge q)$

- New Problem: Given a truth table, find a formula for it

New problem

- Old Problem: Given a formula, find the truth table for it.

Example

$(p \vee q) \wedge \sim(p \wedge q)$

- New Problem: Given a truth table, find a formula for it

- Given a formula find a circuit for it.

Propositional Logic and Circuits

Definition

Each statement of propositional logic can be represented by a circuit with one input for each variable, and a single output bit.

Example

Make circuits for the following:

- $p \vee \sim(q \wedge r)$
- $p \leftrightarrow q$

From Truth Tables to circuits

Example

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
0	0	0	0

Given a TT...

- for each row R of the TT that returns 1 (or T):
  for each variable x_i
  $L_i = x_i$ if Row R has $x_i = 1$;
  $L_i = \sim x_i$ if Row R has $x_i = 0$;
- Write down mini-formula $L_1 \wedge \ldots \wedge L_n$
  KEY: This mini-formula is true IFF that row happens.
- Output the OR of all of the mini formulae

Circuits that Calculate

Circuits can perform math!

Example

- Addition of integers
- Multiplication of integers
- Compute $3x^4+2x^2+7$, where x is an integer
- Approximations of real-valued functions.

Goal

Our goal today will be to build a circuit that can add numbers together:
Inputs: 77 and 49 (in binary)
Output: 126 (in binary)

Brute force: Addition by Truth Table

Adding 2-bit numbers

Example

Adding 2-bit numbers:

	X		+	Y		=	Answer		
3 + 3	1	1	1	1	1	1	0	6
3 + 2	1	1	1	0	1	0	1	5
3 + 1	1	1	0	1	1	0	0	4
3 + 0	1	1	0	0	0	1	1	3
2 + 3	1	0	1	1	1	0	1	5
2 + 2	1	0	1	0	1	0	0	4
2 + 1	1	0	0	1	0	1	1	3
2 + 0	1	0	0	0	0	1	0	2
1 + 3	0	1	1	1	1	0	0	4
1 + 2	0	1	1	0	0	1	1	3
1 + 1	0	1	0	1	0	1	0	2
1 + 0	0	1	0	0	0	0	1	1
0 + 3	0	0	1	1	0	1	1	3
0 + 2	0	0	1	0	0	1	0	2
0 + 1	0	0	0	1	0	0	1	1
0 + 0	0	0	0	0	0	0	0	0

Addition of binary Numbers

Example

$$\begin{array}{r} 1001 \\ +\ 0010 \\ \hline \end{array} \qquad \begin{array}{r} 1001 \\ +\ 0011 \\ \hline \end{array} \qquad \begin{array}{r} 1011 \\ +\ 0010 \\ \hline \end{array} \qquad \begin{array}{r} 1101 \\ +\ 0111 \\ \hline \end{array}$$

Half-Adder

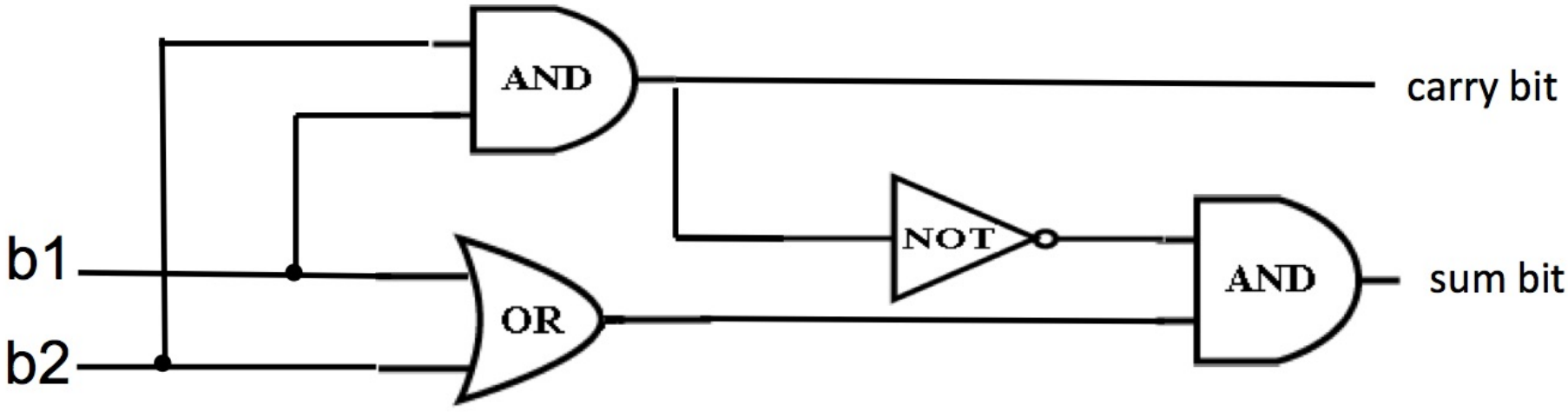
Definition

A *half adder* is a circuit that has 2 boolean inputs, 2 boolean outputs, and outputs the sum and the carry.

Half-Adder

Definition

A *half adder* is a circuit that has 2 boolean inputs, 2 boolean outputs, and outputs the sum and the carry.



Full Adder

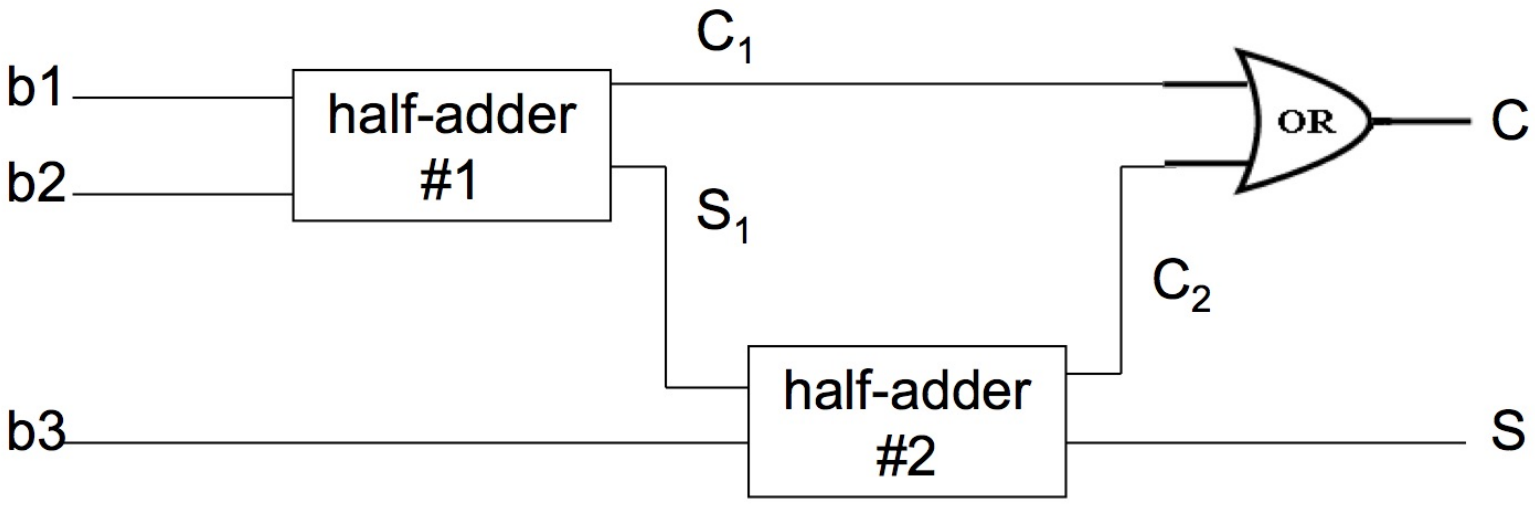
Definition

A *full adder* is a circuit that has 3 boolean inputs, 2 boolean outputs, and outputs the sum and the carry.

Full Adder

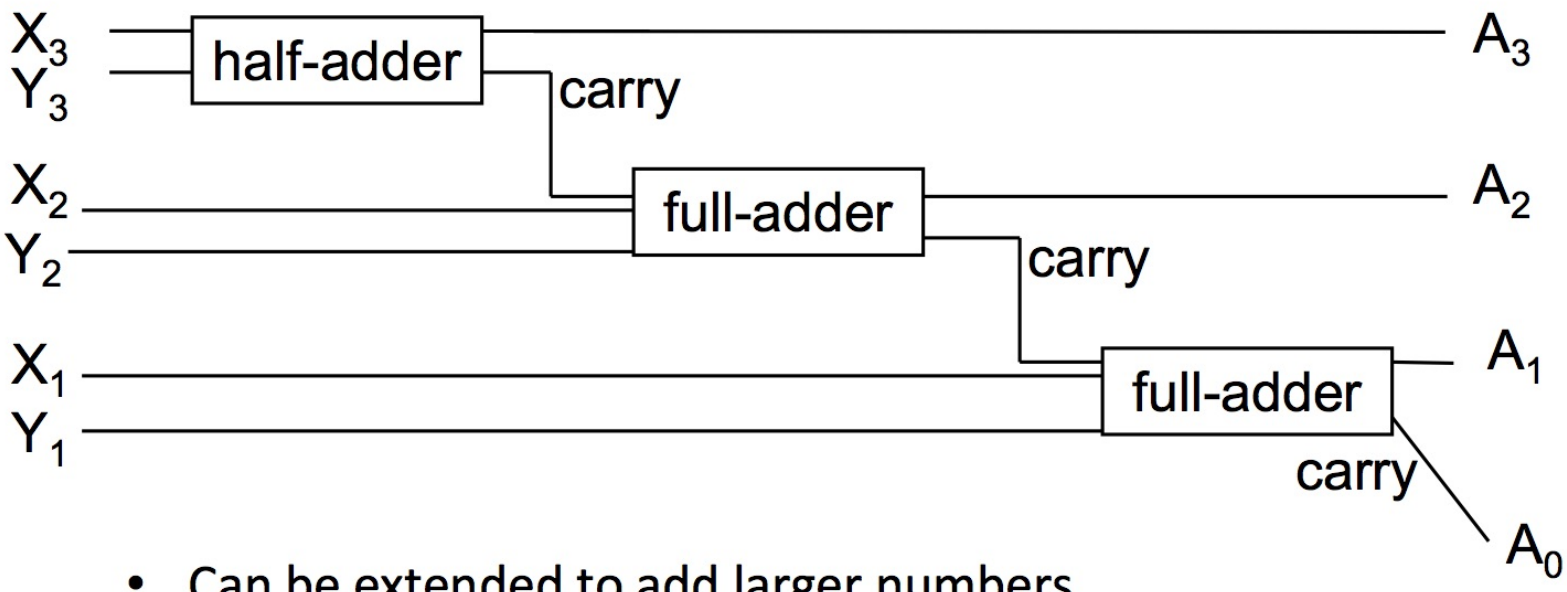
Definition

A *full adder* is a circuit that has 3 boolean inputs, 2 boolean outputs, and outputs the sum and the carry.



Parallel Adder (for 3-bit numbers)

$$\begin{array}{r} X_1X_2X_3 \\ +\ Y_1Y_2Y_3 \\ \hline A_0A_1A_2A_3 \end{array}$$



- Can be extended to add larger numbers