Announcements

Homework 10 (LAST!!) due Wednesday by 5 pm
- Watch TED talk; write essay answering questions

Final Project : Card Game
- Due December 12 – In-class Demos

Intermediate Deadlines
- Wed (11/30): Find project partner
  - Google Doc to find others (email to cs202-tas@cs.wisc.edu)
- Fri (12/2): Project proposal
  - 1 sentence email to cs202-tas@cs.wisc.edu (cc partner)
- Wed (12/7): Project draft to Learn@UW dropbox
  - Whatever you have completed

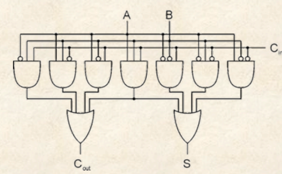
Instructor Office Hours: None today
Tuesday and Thursday -- 1:30 – 4:30

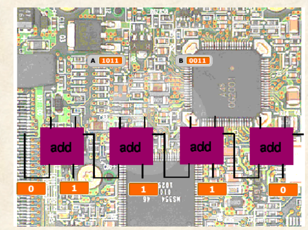
UNIVERSITY of WISCONSIN-MADISON
Computer Sciences Department

CS 202: Introduction to Computation

Professor Andrea Arpaci-Dusseau

How does a computer… do arithmetic?





Previous Lecture: Boolean Logic

Boolean logic: Operates on True (1) and False (0)
- Operators: AND, OR, NOT

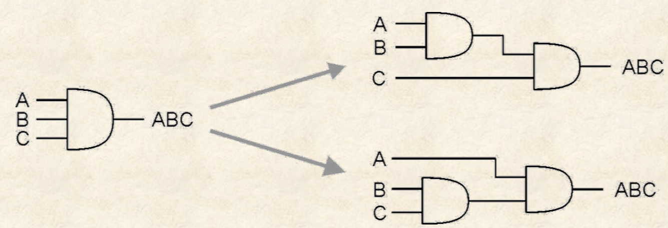
Boolean expression, truth table, combinational circuit
- All three are equivalent

Truth table: Give output for all input combinations
- 1 input variable → how many rows?
- 2 input variables?
- 3 input variables?
- K inputs variables?

2^k rows

Shorthand in Logic Gates

Can draw AND and OR gates with more than two inputs



AND gate: Output is 1 if and only if all inputs are 1

OR gate: Output is 1 if one or more inputs are 1

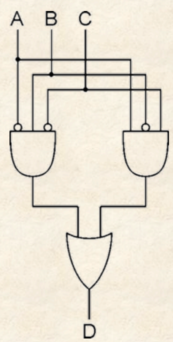
Review: Sum of Products

Can implement ANY truth table with AND, OR, NOT

What is the algorithm? What does circuit look like?

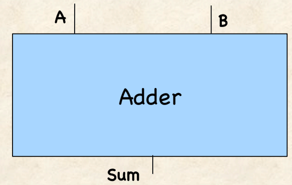
A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

1. AND combinations that yield a "1" in the truth table.
2. OR the results of the AND gates.



Today's Challenge

Can we perform addition using only AND, OR, and NOT gates?



Can compute logic circuit for any function
"Sum = A plus B" is a function of only inputs
Therefore, can create circuit for addition!

Approach #1: Sum of Products

Inputs: Two binary numbers A and B

View each bit of number as an input; 2 bits each

- $A = a_1a_0$
- $B = b_1b_0$

What range of numbers can be added together?

A can be 00, 01, 10, 11 (0, 1, 2, 3)

B can be 00, 01, 10, 11 (0, 1, 2, 3)

Output is a three-bit binary number

- $Sum = s_2s_1s_0$

Are 3 bits enough to represent Sum?

- Largest Sum = 11 + 11 = 3 + 3 = 6 = 100

Approach #1: Sum of Products

View each bit of number as an input; 2 bits each

- $A = a_1a_0$; $B = b_1b_0$

Output is a three-bit binary number

- $Sum = s_2s_1s_0$

Construct truth table of all input combinations

- How many rows?
- 4 bits of input
- $2^4 = 16$ rows of table

Use sum-of-products algorithm for three outputs

- s2, s1, and s0

a_1	a_0	b_1	b_0	s_2	s_1	s_0
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

How to construct addition circuit?

1) Set up Truth Table; Enumerate all 16 input combinations for a1, a0, b1, b0

2) Determine decimal number corresponding to each input

3) Calculate decimal sum as output

4) Translate decimal number to binary

5) Create 3 circuits (Sum-of-products) for s2, s1, s0

Approach #1: Sum of Products

What if 32-bit integers instead of 2-bits??

Number of inputs to circuit?
- 2*32 = 64

How many rows in truth table?
- 2^{64}

Implication:
Need a fundamentally different approach for any real architecture!

Approach #2: Modular Design

Modular Design
- Library of small number of basic components
- Combine together to achieve desired functionality
- Basic principle of modern industrial design

Requires some insight to design component

What algorithm do you use for decimal addition?

$$\begin{array}{r} 6925 \\ +\ 8729 \\ \hline 15654 \end{array}$$

Informal:
- Memorize facts for adding numbers 0..9 to 0..9 + 1 (carry)
- Apply facts to ones position; record units; carry tens
- Repeat for each position (tens, hundred, thousands) w/ carry
- Record final carry out

Algorithm for binary addition?

	1 0 0 1 1	
19	10011	16 + 2 + 1 = 19
+27	11011	16 + 8 + 2 + 1 = 27
46	101110	32 + 8 + 4 + 2 = 46

We know these facts:

0 + 0 + 0 = 00
1 + 0 + 0 = 01
1 + 1 + 0 = 10 (two)
1 + 1 + 1 = 11 (three)

Modular Design for Addition

Notation:

$$
\begin{array}{rcccccl}
 & c_{N-1} & c_{N-2} & \ldots & c_1 & c_0 & \text{Carry bits} \\
 & a_{N-1} & a_{N-2} & \ldots & a_1 & a_0 & \\
+ & b_{N-1} & b_{N-2} & \ldots & b_1 & b_0 & \\
\hline
s_N & s_{N-1} & s_{N-2} & \ldots & s_1 & s_0 &
\end{array}
$$

Repeatedly (N times) do 1-bit full add:
  Take cin, a, b as input
  Compute cout, s as output

Module: 1-bit Full Adder

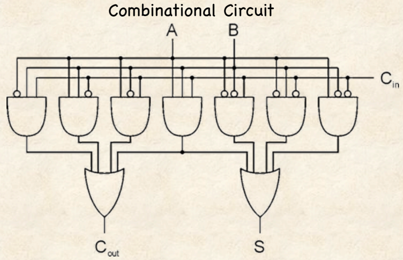
Implement using just AND, OR, NOT

- Add two bits (A, B) and carry-in (C_{in}), for one-bit sum (S) and carry-out (C_{out})

Truth Table

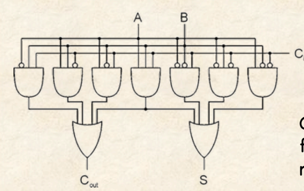
A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Combinational Circuit

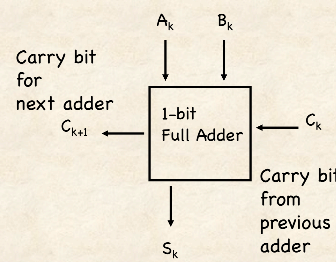


Abstraction: 1-bit Full Adder

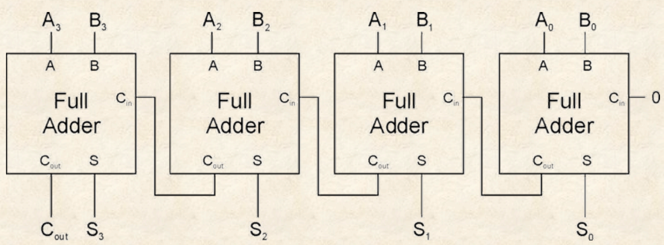
Represent this circuit:



With this diagram:



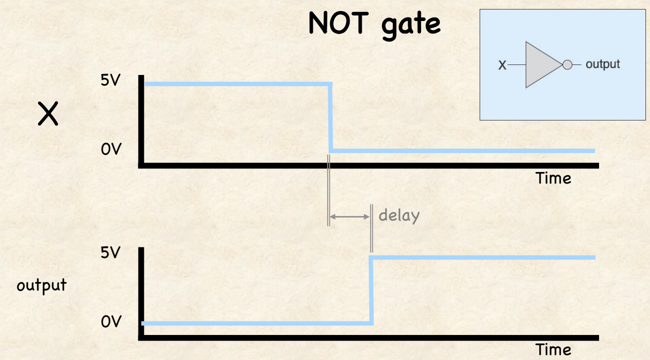
Four-bit Adder



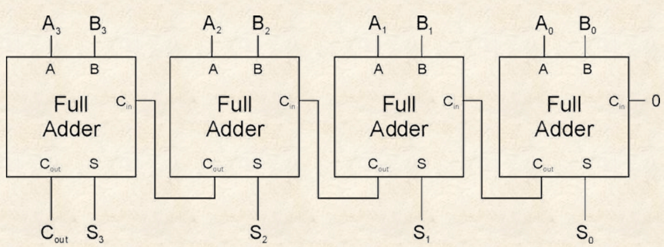
Do you find anything strange or disturbing about this adder?
To compute sum of higher bits need results from lower bits
"Ripple-carry" adder

Timing Diagram

NOT gate



Four-bit Adder



How many gate delays until output settles?
Each 1-bit adder requires 2 gate delays (AND + OR gates)
4 adders * 2 gate delays/adder = 8 gate delays
Influences clock speed of processor

Adder: Example

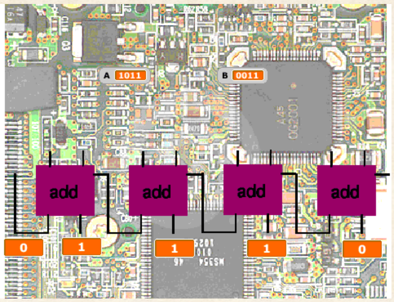
```
        1 1 0 0 1 0
 25      11001
+29      11101
-----------------
```

A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Create Addition Circuit in Scratch!

Add two 4-bit numbers to produce 1 5-bit number

- Only use AND, OR, and NOT!
- 4 nearly identical Sprites



Scripts for 1 Full-bit Adder

Scripts identical across 4 Adder Sprites

Each has private variable for a, b, cin, cout, s

To connect adders, simply set cin to cout of "previous" Sprite

Summary

Today's Topics

- We can do addition with just AND, OR, and NOT!

Homework 10 (LAST!!) due Wednesday by 5 pm

- Watch TED talk; write essay answering questions

Final Project : Card Game

- Due December 12 – In-class Demos

Intermediate Deadlines

- Wed (11/30): Find project partner
  - Google Doc to find others (email to cs202-tas@cs.wisc.edu)
- Fri (12/2): Project proposal
  - 1 sentence email to cs202-tas@cs.wisc.edu (cc partner)
- Wed (12/7): Project draft to Learn@UW dropbox
  - Whatever you have completed

Instructor Office Hours: None today

Tuesday and Thursday -- 1:30 – 4:30