CSE 390 B Spring 2021

Boolean Arithmetic & Time Management

Continuation of Circuit Design, Time Management, Project 2 Overview

Agenda

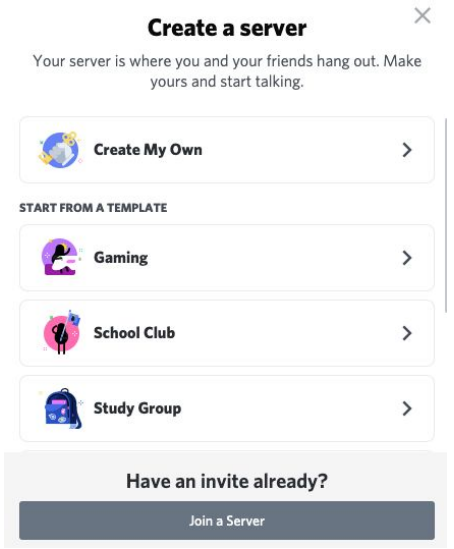
- Connecting w/Other 390B Students
- Introduction to Time Management
- Reading Review and Q&A
- Circuits For Adding Binary Numbers

Agenda

- **Connecting w/Other 390B Students**
- Introduction to Time Management
- Reading Review and Q&A
- Circuits For Adding Binary Numbers

Connect w/Other 390B Students

- Download Discord from https://discord.com/download
- Log in or create an account
- Click on  icon in left-most column to create channel
  - Select “Create My Own”
  - Select “For me and my friends”
  - Give your server a name! (e.g., “CSE 390B”)
- Use the Invite People button to invite peers!
- Create some text and voice channels! Ideas:
  - Text: #projects, #questions, #chill, #off-topic
  - Voice: 🔊 Study Room, 🔊 Lounge
- Feel free to connect via Slack, Messenger, etc. too!



Agenda

- Connecting w/Other 390B Students
- **Introduction to Time Management**
- Reading Review and Q&A
- Circuits For Adding Binary Numbers

Introduction to Time Management

One of your most precious resources is your time.

WHAT TYPICALLY FILLS UP YOUR TIME DURING THE QUARTER?

CLASS LECTURES & QUIZ SECTIONS

WORKING

STUDYING

FAMILY COMMITMENTS

OFFICE HOURS

EXTRACURRICULAR INVOLVEMENT

COMMUTING

HOME CHORES

FRIENDS/PARTNERS

ADMINISTRATIVE WORK

PHYSICAL /MENTAL ACTIVITIES

MORE!

Weekly Time Commitments

Weekly Time Commitments

Time	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
7:30 AM							
8:00 AM							
8:30 AM							
9:00 AM							
9:30 AM							
10:00 AM							
10:30 AM							
11:00 AM							
11:30 AM							
12:00 PM							
12:30 PM							
1:00 PM							
1:30 PM							
2:00 PM							
2:30 PM							
3:00 PM							
3:30 PM							
4:00 PM							
4:30 PM							
5:00 PM							
5:30 PM							
6:00 PM							
6:30 PM							
7:00 PM							
7:30 PM							
8:00 PM							
8:30 PM							
9:00 PM							
9:30 PM							
10:00 PM							
10:30 PM							
11:00 PM							
11:30 PM							

Weekly Time Commitments

- **Class meeting times and quiz sections**
- **Family, friends, community, extracurricular commitments**
- **Meals!**
- **Physical and Mental Activities**
- **Studying for <u>each</u> of your classes**
  - The # of credits for a course reflects the # of hours the class meets. In general, courses require 2 hours of homework for every 1 hour of class.
- **What else is not reflected given your specific situation?**

Agenda

- Connecting w/Other 390B Students
- Introduction to Time Management
- **Reading Review and Q&A**
- Circuits For Adding Binary Numbers

What is Binary?

- A **base n** number system is a system of number representation w/n symbols
  - We are super used to the decimal system, a base 10 number system, with the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  - Increase a number by moving to the next greatest symbol
  - Add another digit when we run out of symbols!

- **Binary** is a base 2 number system
  - Only two symbols (0 and 1)
  - Example values: 1 is the same in decimal and binary, 3 in decimal is 11 in binary

Representing Numbers in Binary

Binary	Decimal
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
. . .	. . .

Representing Numbers in Base 2

- Binary numbers are identical, except "Base 2"
  - Describe a value by specifying multiples of powers of 2
  - For example, a breakdown of 1101 in binary (which is 13 in decimal)

Binary	Power of 2	Decimal
1000	$(1 * 2^3)$	8
100	$(1 * 2^2)$	4
00	$(0 * 2^1)$	0
1	$(1 * 2^0)$	1

Addition

- How do we add two binary numbers?
- As humans, could convert to decimal and then back
  - Example: 0101 + 0010 = ?
  - First convert 0101 to decimal (result is 5)
  - Next convert 0010 to decimal (result is 2)
  - Add the decimal numbers and convert back to binary. 2 + 5 = 7, 7 is 0111 in binary.

- More useful: understand the rules of binary addition so we can teach them to a computer

Binary Addition

- Exactly the same as decimal addition!
  - Right to left (least significant place to most significant place)
  - When a result is more than one digit, **carry** the "overflow"

carry	1	1	1	
x	0	1	1	1
y	0	1	0	1
result	1	0	0	0

Overflow

- What if there's a carry bit in the last column?
- We can't represent it in our fixed-width numbers
  - We are going to "drop" or ignore the extra carry bit

carry	1	1	1	0
x	0	1	1	0
y	1	0	1	0
result	0	0	0	0

Reading Q&A

pollev.com/cse390b

Assuming we are fixing the width of our numbers to be 4-bits, what is the result of adding 1011 and 0110 in binary?

Options:

A. 1101
B. 0001
C. 10001
D. 0111

Agenda

- Connecting w/Other 390B Students
- Introduction to Time Management
- Reading Review and Q&A
- **Circuits For Adding Binary Numbers**

Roadmap: Boolean Arithmetic

- Addition — **Implement**
- Subtraction — **Get it for free!**
- Comparison (<, >, ==) — **Get it for free!**
- Multiplication — **Postpone to software**
- Division — **Postpone to software**

Half Adder

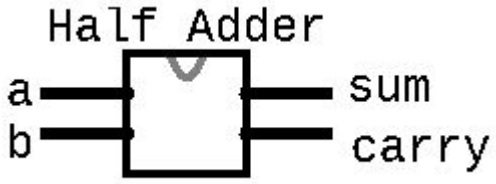
- Circuit that is the logic for adding two bits together
- Takes in two inputs
  - "a" - a bit from the first number being added
  - "b" - a corresponding bit from the second number being added
- Produces two outputs
  - sum - the value to be put for this column in the result
  - carry - the value to be carried over to the next column

```
/**
 * Computes the sum of
 * 2 bits
 */

CHIP HalfAdder {
  IN a, b;
  OUT sum, carry;

  PARTS:
  // Put your code here:

}
```



Half Adder

- Example: 0111 + 0101
- For the first (least significant) column:
  - a = 1
  - b = 1
  - result = 0
  - carry = 1

carry	1	1	1	
x	0	1	1	1
y	0	1	0	1
result	1	0	0	0

Half Adder Group Work

- As a group, come up with the boolean expression that implements the half adder logic
  - First fill in the truth table on the next slide with the correct values for sum and carry-out based on the inputs
  - Develop a boolean expression for sum based on the truth table
  - Develop a boolean expression for carry based on the truth table
- Present your solutions to the course staff (as though we were students learning about half adders)
  - One person give an overview of what the half adder does
  - One person detail how the expression for sum was determined
  - One person detail how the expression for carry was determined

Half Adder Group Work

a	b	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Boolean expressions:
  - sum = a Xor b
  - carry = a And b

Full Adder

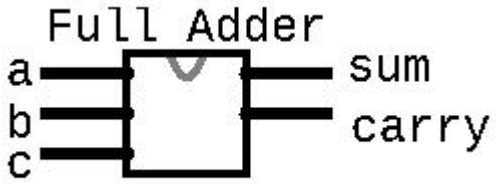
- Circuit that is the logic for adding two bits and the carry from the previous column together
- Takes in three inputs
  - "a" - a bit from the first number being added
  - "b" - a corresponding bit from the second number being added
  - "c" - the carry bit from the previous column
- Produces two outputs
  - sum - the value to be put for this column in the result
  - carry - the value to be carried over to the next column

```
/**
 * Computes the sum of
 * 3 bits
 */

CHIP FullAdder {
  IN a, b, c;
  OUT sum, carry;

  PARTS:
  // Put your code here:

}
```



Full Adder

- Example: 0111 + 0101
- For the second (second least significant) column:
  - a = 1
  - b = 0
  - c = 1
  - result = 0
  - carry = 1

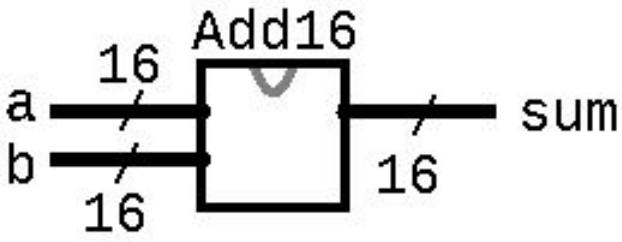
carry	1	1	1	
x	0	1	1	1
y	0	1	0	1
result	1	0	0	0

Full Adder Truth Table

a	b	c	sum	carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Multi-Bit Adder

- Adds two 16-bit numbers
- Connects the full adders for each column together (wires the out carry from one column to the in carry of the next)




```
/**
 * Adds two 16-bit two's complement
 * values. Overflow is ignored.
 */
CHIP Add16 {
  IN a[16], b[16];
  OUT sum[16];

  PARTS:
  // Put your code here:
}
```

Reminders

- **1:1 TA-Student Meetings**
  - First meeting is this week! 1-hour first time; 45-min going forward
- **Office Hours**
  - Porter Wednesdays 4 - 5PM PDT
  - Leslie Tuesdays 4 - 4:30PM PDT
  - Eric & Margot Tuesdays 2 -3PM PDT
- **Project 1: Boolean Logic & Study Skills Inventory**
  - Due Thursday 11:59PM PDT