CSE 390B, Winter 2022

Building Academic Success Through Bottom-Up Computing

Boolean Arithmetic, Time Management

Time Management, Boolean Arithmetic, Adding Binary Numbers, Project 2 Overview

If you can, please have your camera turned on!

Lecture Outline

- **Connect with your CSE 390B peers**
- Introduction to Time Management
  - Weekly Time Commitments
- Reading Review and Q&A
  - Boolean Arithmetic
- Circuits For Adding Binary Numbers
  - Half Adder, Full Adder

Connecting With Your Peers

- 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 

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!

Lecture Outline

- Connect with your CSE 390B peers
- **Introduction to Time Management**
  - **Weekly Time Commitments**
- Reading Review and Q&A
  - Boolean Arithmetic
- Circuits For Adding Binary Numbers
  - Half Adder, Full Adder

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 each 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?

Lecture Outline

- Connect with your CSE 390B peers
- Introduction to Time Management
  - Weekly Time Commitments
- **Reading Review and Q&A**
  - **Boolean Arithmetic**
- Circuits For Adding Binary Numbers
  - Half Adder, Full Adder

What is Binary?

- A **base-n** number system is a system of number representation with n **symbols**

- Decimal system is a base-10 number system
  - Base-10 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
  - Often prefixed with 0b (e.g., 0b1101, 0b10)
  - Base-2 symbols: 0, 1

Representing Numbers in Binary

Binary	Decimal
0b0	0
0b1	1
0b10	2
0b11	3
0b100	4
0b101	5
0b110	6
0b111	7
…	…

Representing Numbers in Base-2

- Binary numbers are identical, except in base-2
  - Describe a value by specifying multiples of powers of 2
  - For example, a breakdown of $0b1101$ in binary (13 in decimal)

Binary	Power of 2	Decimal
$0b1000$	1×2^3	8
$0b100$	1×2^2	4
$0b00$	0×2^1	0
$0b1$	1×2^0	1

Addition

- How do we add two binary numbers?
  - As humans, could convert to decimal and then back
- Example: $0b0101 + 0b0010$
  - First convert $0b0101$ to decimal (result is 5)
  - Next convert $0b0010$ to decimal (result is 2)
  - Add the decimal numbers and convert back to binary
    - $2 + 5 = 7$, which is $0b111$ in binary
- What's more useful is understanding the rules of binary addition so we can teach them to a computer

Case Study: Decimal Addition

- Consider how we perform decimal addition
  - Right to left (least significant place to most significant place)
  - When a result is more than one digit, carry the "overflow"

carry				
a	5	7	8	3
b	2	4	5	6
sum				

Binary Addition

- Binary addition is conceptually 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				
a	0	1	1	1
b	0	1	0	1
sum				

Overflow

- What if there's a carry bit in the last column?

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

Overflow

- What if there's a carry bit in the last column?

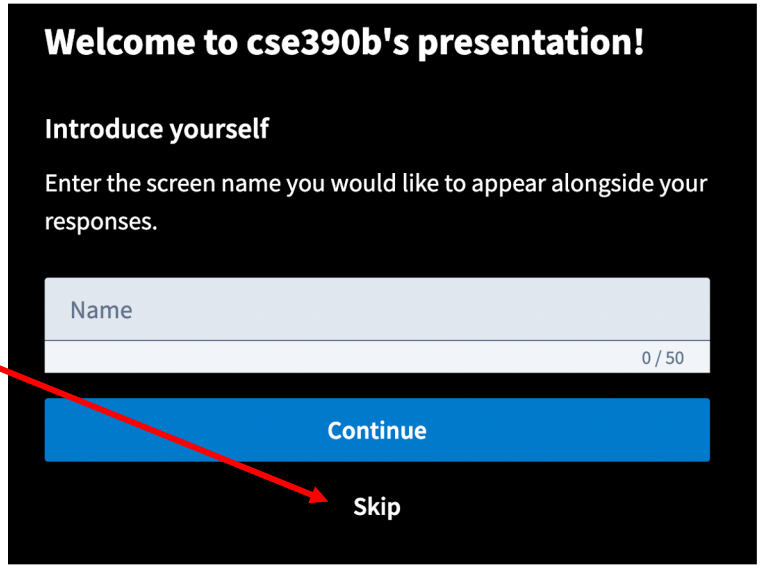
- We can't represent it in our fixed-width numbers
  - We are going to "drop" or ignore the extra carry bit

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

Poll Everywhere

Vote at https://pollev.com/cse390b

- **What is something you learned, were surprised by, or had a question about from today's reading?**
- **What general questions, comments, concerns, or feedback do you have for the course staff from Week 1?**
- You can choose to respond anonymously by not adding your name (click "Skip")



Poll Everywhere

Vote at https://pollev.com/cse390b

- **Assuming we are fixing the width of our numbers to be four bits, what is the result of adding 0b1011 and 0b0110 in binary?**

A. 0b1101
B. 0b0001
C. 0b10001
D. 0b0111
E. We're lost…

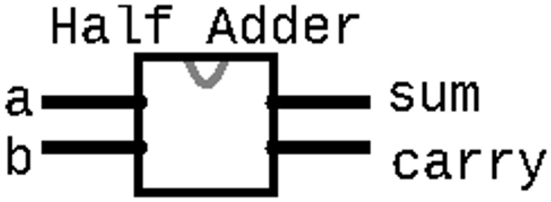
Lecture Outline

- Connect with your CSE 390B peers
- Introduction to Time Management
  - Weekly Time Commitments
- Reading Review and Q&A
  - Boolean Arithmetic
- **Circuits For Adding Binary Numbers**
  - **Half Adder, Full Adder**

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 for adding two bits together
- Takes in two inputs: `a`, `b`
  - `a` is the first bit being added
  - `b` is the corresponding bit to be added
- Produces two outputs: `sum`, `carry`
  - `sum` is the value to be put for this column in the result
  - `carry` is the value to be carried over to the next column

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

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

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

    PARTS:
    // Put your code here:

}
```

Half Adder

- Example: $0b0111 + 0b0101$

- For the first (least significant) column:
  - $a = 1$
  - $b = 1$
  - sum =
  - carry =

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

Half Adder Group Work

- Determine the half adder logical Boolean expression
  - First, fill in the truth table values for sum and carry based on the inputs
  - Then, develop a Boolean expression for sum and carry based on the truth table

- Five-minute breakout rooms, identify one person to share each of the following after coming back:
  - Overview of what the half adder does
  - Thought process for reaching the Boolean expression for sum
  - Thought process for reaching the Boolean expression for carry

Half Adder Group Work

- Boolean expressions:
  - sum =
  - carry =

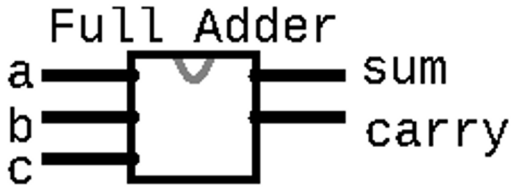
a	b	sum	carry
0	0		
0	1		
1	0		
1	1		

Half Adder Group Work

- Boolean expressions:
  - sum = a XOR b
  - carry = a AND b

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

Full Adder



- Circuit for adding three bits together (two bits *and* carry bit together from previous column)
  - `a` is the first bit being added
  - `b` is the corresponding bit to be added
  - `c` is the carry bit from the right column

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

- Produces two outputs: `sum`, `carry`
  - `sum` is the value to be put for this column in the result
  - `carry` is 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: $0b111 + 0b101$

- For the second (second least significant) column:
  - $a = 1$
  - $b = 0$
  - $c = 1$
  - sum =
  - carry =

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

Full Adder Truth Table

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

Poll Everywhere

Vote at https://pollev.com/cse390b

- **What are the sum and carry bits when a=0, b=1, and c=1?**

A. sum=0, carry=0
B. sum=0, carry=1
C. sum=1, carry=0
D. sum=1, carry=1
E. We're lost…

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

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

Poll Everywhere

Vote at https://pollev.com/cse390b

- **Now that we have a truth table for a full adder, how would we go about determining the Boolean algebra expressions for sum and carry?**

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

$\xrightarrow{?}$ $\text{sum} =$
$\text{carry} =$

Multi-Bit Adder

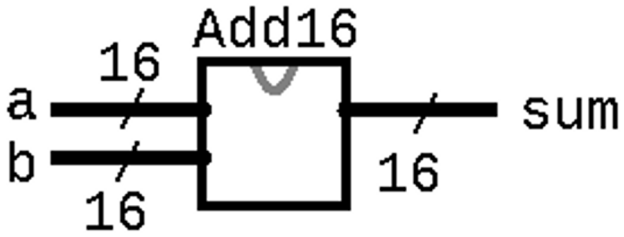
- Adds two 16-bit numbers

- 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:

}
```



Lecture 3 Reminders

- Virtual through the end of Week 2
- 1:1 Student-TA Meetings
  - 1 hour for first meeting, 45 minutes going forward
- Office Hours: Zoom links available via Canvas
  - Eric: Tuesdays and Thursdays, 3-4:00pm at CSE2 152
  - Leslie: Wednesdays, 4:30-5pm at CSE2 174
  - Audrey and Sean: Wednesdays, 1:30-2:30pm at CSE2 152
- Project 1: Boolean Logic & Study Skills Inventory
  - **Due on Thursday (1/13) at 11:59PM PST**