CSE 390B, Autumn 2022

Building Academic Success Through Bottom-Up Computing

Procrastination & Boolean Arithmetic

Combating Procrastination, Overview of Numbers in Binary, Boolean Arithmetic, Circuits for Adding Binary Numbers

Project 2 Check-in

- How has Project 2 been coming along?
- What questions do you have about Project 2?
- Following the plan of action outlined in slide 20 of last Thursday's lecture will work for implementing most chips
- Remember to double check your submission on GitLab
  - Navigate to GitLab, open tags, and verify that the associated commit includes your expected changes

Lecture Outline

- **Combating Procrastination**
  - **Procrastination Reflection and Avoidance Tips**
- Overview of Numbers in Binary
  - Comparison Between Binary and Decimal
- Boolean Arithmetic
  - Addition Operator and Handling Binary Overflow
- Circuits for Adding Binary Numbers
  - Overview of the Half Adder and Full Adder

Let's Talk Procrastination

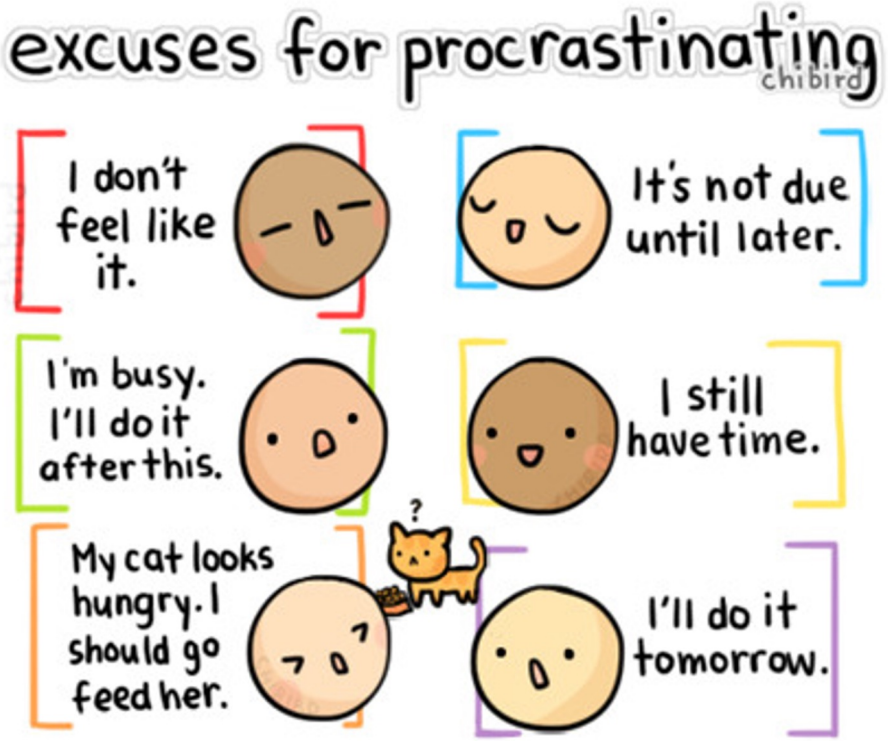
- What is procrastination?
  - Procrastination is the act of putting things off or choosing to do something you prefer to do (or might even need to do) instead of the actual project or chore or work you need to be doing now
  - Common challenge for college students, with about 80-95% of students reporting that they procrastinate (Steel, 2007)

Steel, Piers. "The nature of procrastination: a meta-analytic and theoretical review of quintessential self-regulatory failure." *Psychological Bulletin Journal* 133, no. 1 (2007): 65–94. https://www.researchgate.net/publication/6598646_The_nature_of_procrastination_a_meta-analytic_and_theoretical_review_of_quintessential_self-regulatory_failure_Psychol_Bull_133_65-94

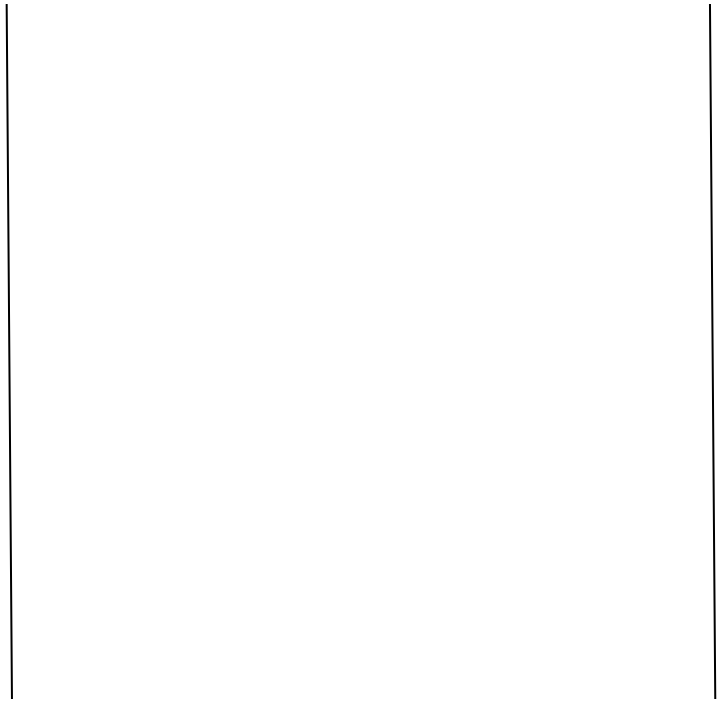
Combating Procrastination

- Identifying why we procrastinate and the internal dialogue we have with ourselves

- Create a proactive strategy to course-correct when you notice you're putting off what needs to be done



Grab a piece of paper!



Grab a piece of paper!

AVOIDANCE AREAS

When you procrastinate, what do you avoid doing?

Identify 3-5 areas

Where does procrastination impact you most?

PERSONAL

- Eating well
- Exercising / Wellness activities
- Getting enough sleep
- Bathing & hygiene
- Health care (i.e. doctor's visit)
- Balancing bank account
- Relaxation & hobbies

SCHOOL/COLLEGE

- Going to class
- Doing class readings
- Studying for tests/exams
- Doing homework/ assignments
- Writing papers
- Starting long-term projects
- Finding a study group
- Talking to an instructor or TA
- Making an advising appointment

SHOPPING/HOME/ MAINTENANCE

- Paying bills
- Getting financial aid taken care of (i.e. FAFSA, forms, etc)
- Doing laundry
- Cleaning
- Grocery shopping
- Doing dishes

SOCIAL/RELATIONSHIPS

- Talking with friends
- Writing email responses
- Socializing
- Calling relatives

WORK/CAREER

- Going to work
- Applying to internships/jobs
- Preparing a resume
- Studying for interviews

Grab a piece of paper!

AVOIDANCE AREAS

When you procrastinate, what do you avoid doing?

Identify 3-5 areas

PROCRASTINATION BEHAVIORS

How do you procrastinate? In other words, what do you do instead of the work that needs to be done?

Identify 3-5 behaviors

Grab a piece of paper!

AVOIDANCE AREAS	PROCRASTINATION BEHAVIORS	PLANNING FOR SUCCESS
When you procrastinate, what do you avoid doing?	*How do you procrastinate? In other words, what do you do instead of the work that needs to be done?*	*What can you do to avoid procrastination? What action can you take to **refocus** yourself on the task you need to complete?*
Identify 3-5 areas	*Identify 3-5 behaviors*	*Identify 3-5 actions*

Tips for Avoiding Procrastination

- Prioritize the tasks that you need to complete
- Plan for the tasks you need to complete
  - Review your to-do list and schedule of upcoming events
- Eliminate distractions that pull you away from focusing on the task at hand
  - Isolate yourself from your phone, close distracting websites, etc.
- Make productive behavior accessible and sources of procrastination harder to access

Lecture Outline

- Combating Procrastination
  - Procrastination Reflection and Avoidance Tips
- **Overview of Numbers in Binary**
  - **Comparison Between Binary and Decimal**
- Boolean Arithmetic
  - Addition Operator and Handling Binary Overflow
- Circuits for Adding Binary Numbers
  - Overview of the Half Adder and 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 (each called a **digit**)
  - 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
  - Base 2 symbols: 0, 1 (each called a **bit**)
  - Often prefixed with 0b (e.g., $0\text{b}1101$, $0\text{b}10$)
  - **Least-significant bit (LSB):** Lowest-order position of a binary value
  - **Most-significant bit (MSB):** Highest-order position of a binary value

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
$0b1000$	1×2^3
$0b0100$	1×2^2
$0b0000$	0×2^1
$0b0001$	1×2^0

Binary vs. Decimal

Binary	Decimal
0b000	0
0b001	1
0b010	2
0b011	3
0b100	4
0b101	5
0b110	6
0b111	7
…	…

Poll Everywhere

Vote at https://pollev.com/cse390b

What is the binary representation of the decimal value 29?

A. 0b011011
B. 0b011101
C. 0b100011
D. 0b100111
E. We're lost…

Lecture Outline

- Combating Procrastination
  - Procrastination Reflection and Avoidance Tips
- Overview of Numbers in Binary
  - Comparison Between Binary and Decimal
- **Boolean Arithmetic**
  - **Addition Operator and Handling Binary Overflow**
- Circuits for Adding Binary Numbers
  - Overview of the Half Adder and 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

Binary Addition

- How do we add two binary numbers?
  - As humans, we could convert to decimal and then back

- Example: $0\text{b}101 + 0\text{b}010$
  - First convert $0\text{b}101$ to decimal (result is 5)
  - Next convert $0\text{b}010$ to decimal (result is 2)
  - Add the decimal numbers and convert back to binary
    - $5 + 2 = 7$, which is $0\text{b}111$ 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 column's result is more than one digit, carry over the digit that overflows

- Example:

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

Binary Addition

- Binary addition conceptually the same as decimal addition
  - Right to left (least significant place to most significant place)
  - When a column's result is more than one digit, carry over the bit that overflows

- Example:

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

Binary Overflow

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

- Example:

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

Binary 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					

Five-minute Break!

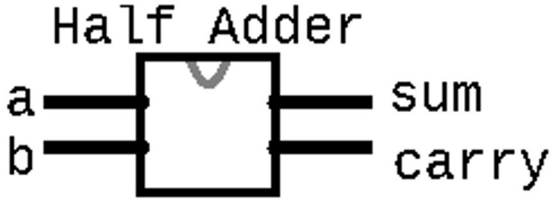
- Feel free to stand up, stretch, use the restroom, drink some water, review your notes, or ask questions

- We'll be back at:

Lecture Outline

- Combating Procrastination
  - Procrastination Reflection and Avoidance Tips
- Overview of Numbers in Binary
  - Comparison Between Binary and Decimal
- Boolean Arithmetic
  - Addition Operator and Handling Binary Overflow
- **Circuits for Adding Binary Numbers**
  - **Overview of the Half Adder and Full Adder**

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

- Example: 0b0111 + 0b0101

- For the right-most (least significant) column:
  - a = 1
  - b = 1
  - sum =
  - carry =

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

Half Adder Example

- Boolean expressions:
  - sum =
  - carry =

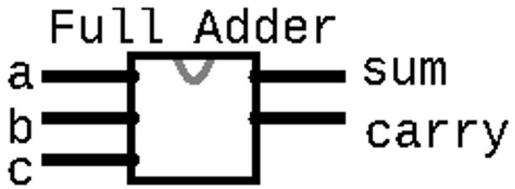
a	b	sum	carry
0	0		
0	1		
1	0		
1	1		

Half Adder Example

- 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: $0b0111 + 0b0101$

- 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	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

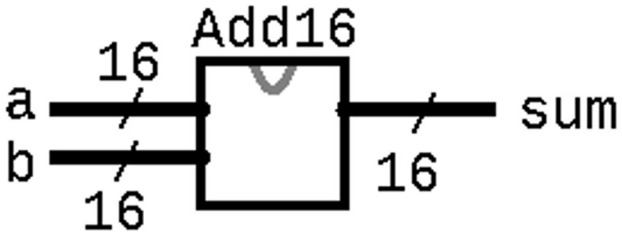
E. We're lost…

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

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:

}
```

```
    … 0 1 1 1 0 0
    … 0 0 1 0 1 0 1
+   … 1 0 1 1 1 0 0
-------------------
    … 1 1 1 0 0 0 1
```

Lecture 4 Wrap-up

- Exciting lecture topics this Thursday!
  - Metacognitive Subject: Growth vs. Fixed Mindset
  - Technical Subject: Binary Number Representations and the ALU

- **Project 2 due this Thursday (10/13) at 11:59pm**

- Preston has office hours after class in CSE2 153
  - Feel free to post your questions on the Ed board as well

- Join the students only CSE 390B Discord channel!
  - https://discord.gg/3ZTCPvgJeJ