CSE 390B, Spring 2022

Building Academic Success Through Bottom-Up Computing

Time Management & Boolean Arithmetic

Time Management, Binary and Boolean Arithmetic, Circuits for Adding Binary Numbers, Making Decisions in Hardware

Connect With Your CSE 390B 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 22sp”)
- Use the 

button to invite peers
- Create some text and voice channels. Some ideas:
  - Text: #projects, #questions, #chill, #random
  - Voice: 🔊 Study Room, 🔊 Lounge
- Feel free to connect via Slack, Messenger, etc. too

Project 2 Check-in

- How has Project 2 been going?
- What questions do you have about Project 2?
  - HardwareSimulator or GitLab setup-related issues?
- You'll discuss with your TA about the Study Skills Inventory and revisit it at the end of the quarter
- Read Part II Tips for guidance on how to begin technical portion

Lecture Outline

- **Time Management**
  - **Weekly Time Commitments**
- Binary and Boolean Arithmetic
  - Addition Operator and Overflow
- Circuits For Adding Binary Numbers
  - Half Adder, Full Adder
- Making Decisions in Hardware
  - Multiplexer (Mux) Logical Gate

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

AND 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
- Physical and Mental Activities
- Studying for each of your classes
  - The number of credits for a course reflects the number of hours the class meets
  - In general, courses require two hours of homework for every one hour of class
- What else is not reflected given your specific situation?

Time Management Group Discussion

- Now that you've filled out your weekly time commitments, discuss in groups of 3-4:
  - What is one thing that is most surprising to you?
  - What is one thing you might change?
  - How might you use this time commitments sheet in the future?

Lecture Outline

- Time Management
  - Weekly Time Commitments
- **Binary and Boolean Arithmetic**
  - **Addition Operator and Overflow**
- Circuits For Adding Binary Numbers
  - Half Adder, Full Adder
- Making Decisions in Hardware
  - Multiplexer (Mux) Logical Gate

Poll Everywhere

Vote at https://pollev.com/cse390b

What is the binary representation of the decimal value of 29?

A. 0b011011

B. 0b011101

C. 0b100011

D. 0b100111

E. We're lost…

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	Decimal	Power of 10
$0b0001$	1×2^0	1	1×10^0
$0b0010$	1×2^1	2	2×10^0
$0b0100$	1×2^2	4	4×10^0
$0b1000$	1×2^3	8	8×10^0

Representing Numbers in Binary

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

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 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				

Binary Overflow

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

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!

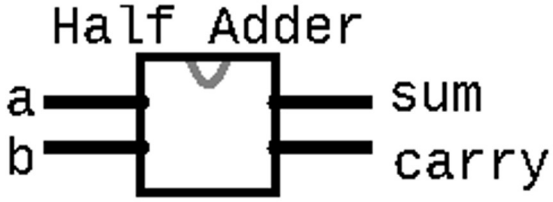
- Feel free to stand up, stretch, use the restroom, drink some water, review your notes, or ask questions
- We'll be back at:
- Any song recommendations? Respond on Poll Everywhere at https://pollev.com/cse390b
- Research shows mid-lecture breaks reduce the decline of attention (Olmsted, 1999)

Olmsted III, John. "The Mid-Lecture Break: When Less Is More." *Journal of Chemical Education* (1999). https://pubs.acs.org/doi/abs/10.1021/ed076p525.

Lecture Outline

- Time Management
  - Weekly Time Commitments
- Binary and Boolean Arithmetic
  - Addition Operator and Overflow
- **Circuits For Adding Binary Numbers**
  - **Half Adder, Full Adder**
- Making Decisions in Hardware
  - Multiplexer (Mux) Logical Gate

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 =

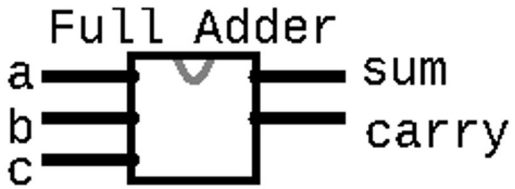
a	b	sum	carry
0	0		
0	1		
1	0		
1	1		

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

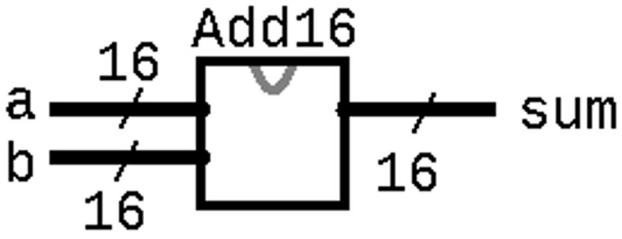
E. We're lost…

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

Lecture Outline

- Time Management
  - Weekly Time Commitments
- Binary and Boolean Arithmetic
  - Addition Operator and Overflow
- Circuits For Adding Binary Numbers
  - Half Adder, Full Adder
- **Making Decisions in Hardware**
  - **Multiplexer (Mux) Logical Gate**

Making Decisions in Hardware

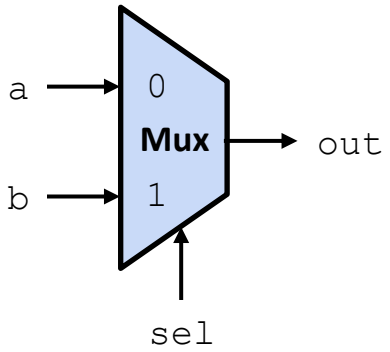
- We write if / else statements in Java with the understanding that *only one* of the branches will run
  - For example, in the following code, we expect to compute one of `a & b` or `a | b` (not both)

```
if (c == 0) {
    out = a & b;
} else {
    out = a | b;
}
```

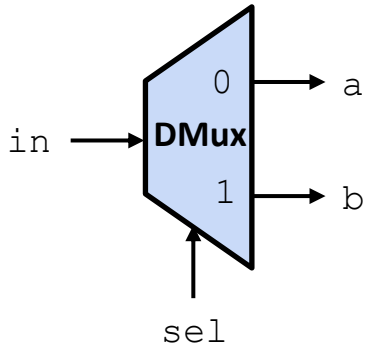
- In hardware, all circuits are always executing
  - We can't "turn off" a circuit based on a condition
  - Instead, we create circuits for different conditions and choose which output based on a condition instead

Making Decisions in Hardware

- We can use a **Multiplexer (Mux) gate** to choose which singular input to output
  - Takes three inputs: `a`, `b`, and `sel`
  - If `sel == 0`, then `out = a`
  - Otherwise, `out = b`



- A **Demultiplexer (DMux) gate** passes one input to one of several outputs and `0` to the rest of the outputs
  - Takes two inputs: `in` and `sel`
  - If `sel == 0`, then `a = in` and `b = 0`
  - Otherwise, `a = 0` and `b = in`



Mux Gate Implementation Example

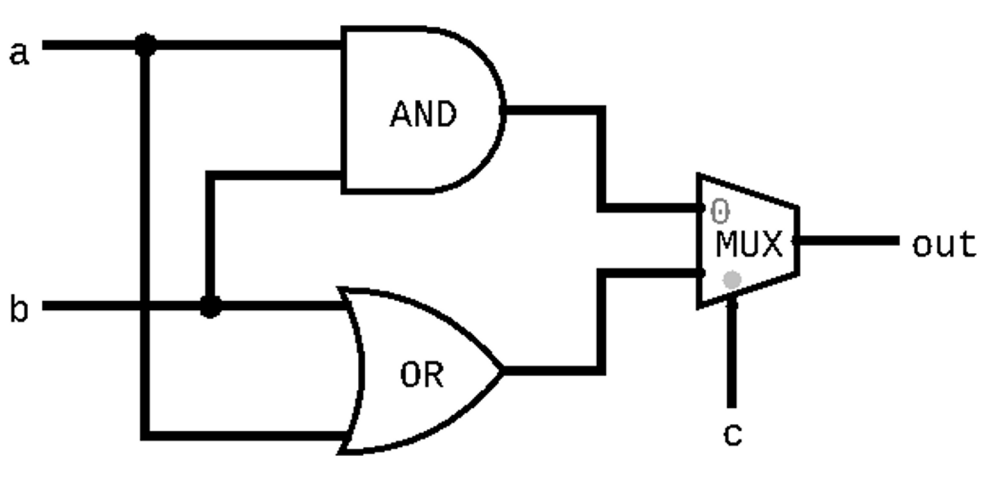
- Example of converting pseudocode into a hardware circuit diagram:

```
if (c == 0) {
    out = a & b;
} else {
    out = a | b;
}
```

Mux Gate Implementation Example

- Example of converting pseudocode into a hardware circuit diagram:

```
if (c == 0) {
    out = a & b;
} else {
    out = a | b;
}
```



Mux Gate Practice Problem 1

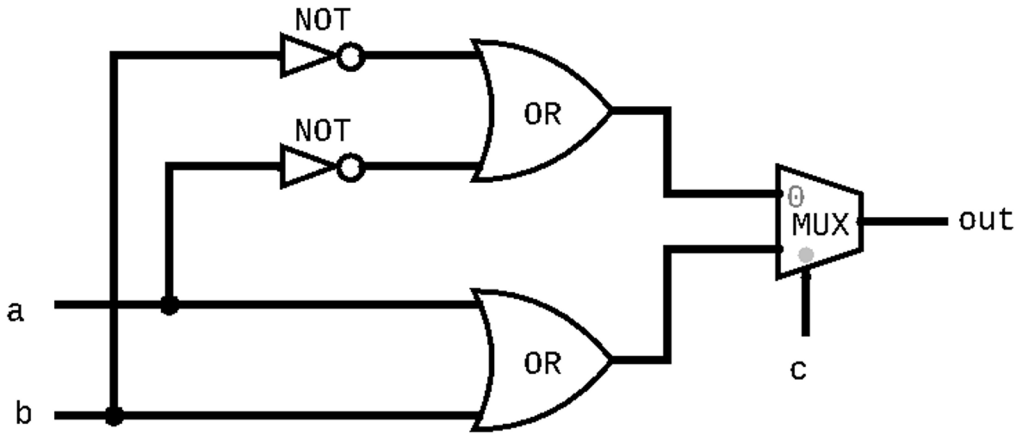
- Draw out the hardware circuit diagram that corresponds to the following pseudocode:

```
if (c == 0) {
    out = ~a | ~b;
} else {
    out = a | b;
}
```

Mux Gate Practice Problem 1

- Draw out the hardware circuit diagram that corresponds to the following pseudocode:

```
if (c == 0) {
    out = ~a | ~b;
} else {
    out = a | b;
}
```



Mux Gate Practice Problem 2

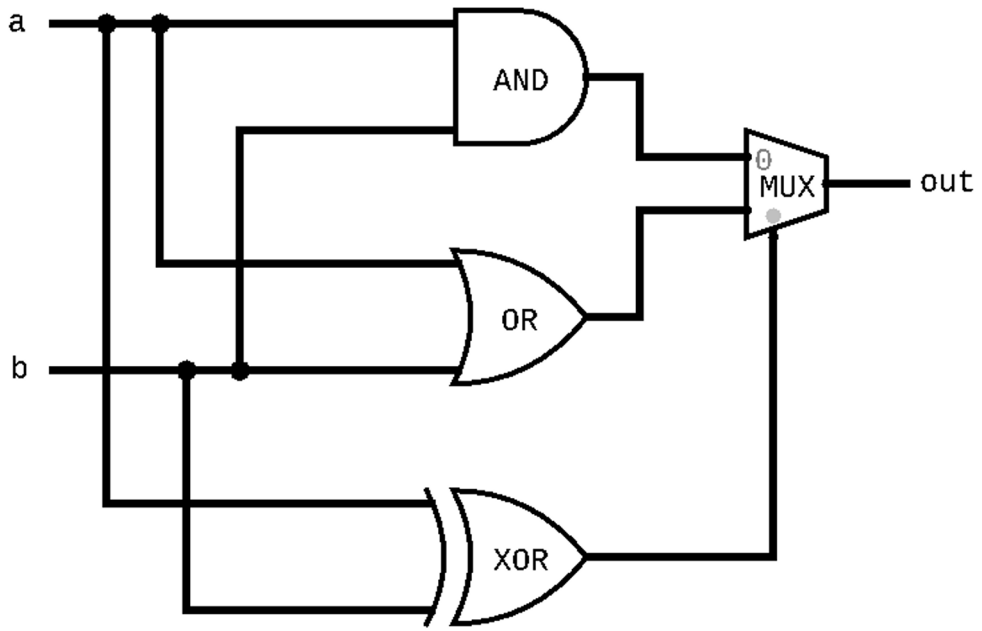
- Draw out the hardware circuit diagram that corresponds to the following pseudocode:

```
if (a == b) {
    out = a & b;
} else {
    out = a | b;
}
```

Mux Gate Practice Problem 2

- Draw out the hardware circuit diagram that corresponds to the following pseudocode:

```
if (a == b) {
    out = a & b;
} else {
    out = a | b;
}
```



Lecture 3 Wrap-up

- Project Reminders
  - Project 1 feedback released on Gradescope
  - **Project 2 due this Thursday (4/7) at 11:59pm PDT**

- Lecture 4 Reading: Negative Numbers in Binary

- Course Staff Support
  - Eric has office hours in CSE2 153 today after lecture until 5pm
  - You can also post your questions on the Ed discussion board