Welcome to ES 4!

Fill out a name tent with the name you prefer to go by.

Please sit toward the front; it's a big room!

On your notecard write:

1. Your name

2. Why you're taking this course

3. What you hope to gain from this course

4. Something you're nervous about

ES 4: Introduction to digital logic circuits

Steven Bell

5 September 2023

Tufts UNIVERSITY

About me

Bachelor's in
Computer Engineering

Oklahoma Christian University

MS/PhD in
Electrical Engineering

Stanford

Two summer-long internships

NVIDIA

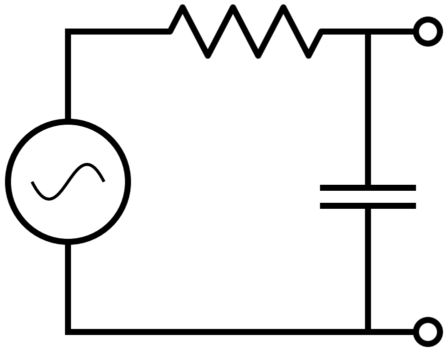



About you

1. Find everyone with the same color name tent
2. Find something you have in common that other groups don't

   Marker color doesn't count!
3. Pose for a picture

What will we learn in ES 4?



Circuits

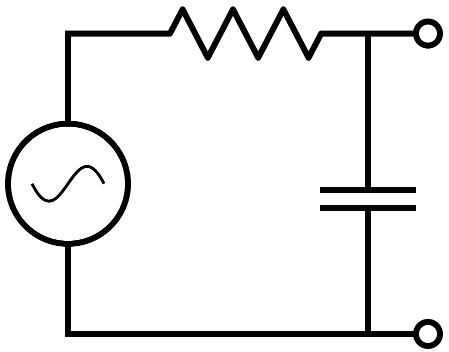
(physics + EE 20)

```c
int main(int argc, char* argv[]){
  for(int i = 0; i < 100; i++){
    printf("i: %d\n", i);
 }
  return(0);
}
```

Programming

(CS 11)

What will we learn in ES 4?



Circuits

(physics + EE 20)

Digital design

(ES 4)

```
int main(int argc, char* argv[]){
  for(int i = 0; i < 100; i++){
    printf("i: %d\n", i);
 }
  return(0);
}
```

Programming

(CS 11)

Who cares?

Who cares?

(one example)

CPU

<100 MHash/sec

0.5 MHash/J

Where does energy go in a CPU?

Figuring out what to do
Moving data around
Keeping track of what's been done

Actually computing the hash

https://en.bitcoin.it/wiki/Mining_hardware_comparison

GPU

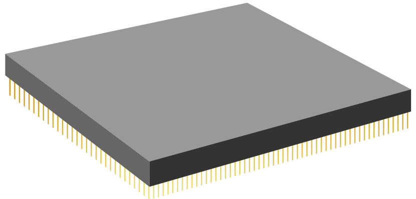
~1 GHash/sec 10x

2 MHash/J 4x

Figuring out what to do
Moving data around
Keeping track of what's been done

Actually computing the hash

https://en.bitcoin.it/wiki/Mining_hardware_comparison



ASIC

~10 THash/sec ~ 10,000x!

10,000 MHash/J ~ 10,000x!

Actually computing the hash

https://en.bitcoin.it/wiki/Mining_hardware_comparison

What other domains are like this?

Neural networks / machine learning

Image processing

Wearables / IoT

Ok, but I don't plan to design hardware,
I'm just gonna write code!

Two ways to think about Computer Engineering

(and more specifically, this course)

1) Designing computer hardware

2) Writing code with knowledge of how the hardware works

Fine, but really I just want a job…

Where are we going in this course?


Introduction to Digital Logic
Computer Programming
Verilog
VHDL
Fields of Englightenment
Sequential Logic
Memory
Process Block Perils
Swamps of Arcane Syntax
State Machines
Flip-flop Pass
Computer Architecture
Plains of Combinational Logic
Desert of "How things used to be done"

Combinational logic

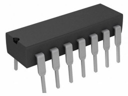
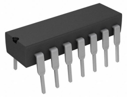
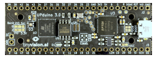
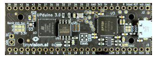
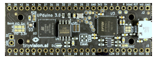
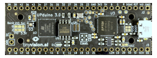
VHDL and FPGAs

Sequential logic

Building a computer

			Topic	Reading	Assignments
Sep 08	Wednesday	1	Welcome and introduction		Lab 1 (blinky lights) released
Sep 13	Monday	2	Boolean equations, truth tables, and circuits	1.5, 2.1-2.2	
Sep 15	Wednesday	3	Manipulating boolean equations	2.3-2.4	L[illegible] (comparator) released
Sep 20	Monday	4	Minimizing logic with Karnaugh maps	2.5-2.7	
Sep 22	Wednesday	5	Multiplexers and FPGAs		Lab 3 (mux adder) released
Sep 27	Monday	6	Timing combinational logic		
Sep 29	Wednesday	7	VHDL for combinational logic	4.1-4.2	
Oct 04	Monday	8	Testing and testbenches		
Oct 06	Wednesday	9	Computer arithmetic	[illegible]	Lab 4 (FP[illegible]rmup) released
Oct 11	Monday		*No class, Indigenous People's Day*		
Oct 13	Wednesday	10	Adders and other combinational circuits	5.1-5.2	Lab 5 ([illegible] + 7-segment) released
Oct 18	Monday	11	**Exam 1**		
Oct 20	Wednesday	12	Latches and flip-flops	3.1-3.3	
Oct 25	Monday	13	Basic sequential circuits	4.4-4.5, 3.[illegible]	
Oct 27	Wednesday	14	State machines, part 1	3.4, 4.6	Lab 6 ([illegible] 7-segment) released
Nov 01	Monday	15	State machines, part 2		
Nov 03	Wednesday	16	Timing sequential logic		
Nov 08	Monday	17	VHDL, FPGAs, and the real world		
Nov 10	Wednesday	18	Memory: registers, RAM, and ROM	5.5	Lab 7 (V[illegible]NES) released
Nov 15	Monday	19	Introduction to final project, designing digital systems		
Nov 17	Wednesday	20	**Exam 2**		
Nov 22	Monday	21	Digital peripherals and protocols	9.1-9.3.4.1	Project proposals due
Nov 24	Wednesday		*No class, Thanksgiving break*		
Nov 29	Monday	22	Testing sequential logic		
Dec 01	Wednesday	23	The ARM instruction set	6.1-6.4	
Dec 06	Monday	24	Components of a microprocessor	7.1-7.3.1	
Dec 08	Wednesday	25	How a microprocessor executes code	7.3.2-7.3.4	Project block diagrams due
Dec 13	Monday	26	Final project demonstrations		
Dec 23	Thursday		Project writeups due		

Labs

- Lab 1 Blinky lights
- Lab 2 Comparator 
- Lab 3 Mux adder 
- Lab 4 FPGA warmup 
- Lab 5 ALU + 7-segment 
- Lab 6 2-digit display 
- Lab 7 Multiple options 
- Final project

I want everyone to succeed in this class!

We're going to cover a lot of ground, and it will be challenging,
but you have what it takes!

We'll evaluate your learning based on what you can do,
not on a curve against each other.

If you're falling behind, change something!

Rarely does "working harder" solve the problem.

A word on stereotypes

The digital abstraction

We handle complexity with abstraction.

And we enable abstractions by making restrictions.

The first fundamental one is the "digital abstraction"

By the end of class today, you should be able to:

- Write a positive decimal number in binary and vice-versa
- Add numbers in binary

0

1

10

11

100

101

110

111

Decimal to binary (one of several ways)

If the number is odd, write a 1 and subtract 1 from the number

If the number is even, write a 0

Divide the number by 2 and repeat (moving one place to the left)

37 → 36

18

9 → 8

4

2

1

32	16	8	4	2	1
1	0	0	1	0	1

32 + 4 + 1 = 37

53 -> 52 110101

26

13 -> 12

6

3 -> 2

1

32 16 8 4 2 1

100110

32 +4 +2 = 38

Binary to decimal

Just add up the place values…

Binary addition

		16	8	4	2	1
		1	1	1		
14		0	1	1	1	0
23		1	0	1	1	1
37	1	0	0	1	0	1

32 + 4 + 1 = 37

Binary addition

	32	16	8	4	2	1
42	1	0	1	0	1	0
36	1	0	0	1	0	0
1	0	0	1	1	1	0

$64 + 8 + 4 + 2 = 78$

What's the largest value you can represent with...

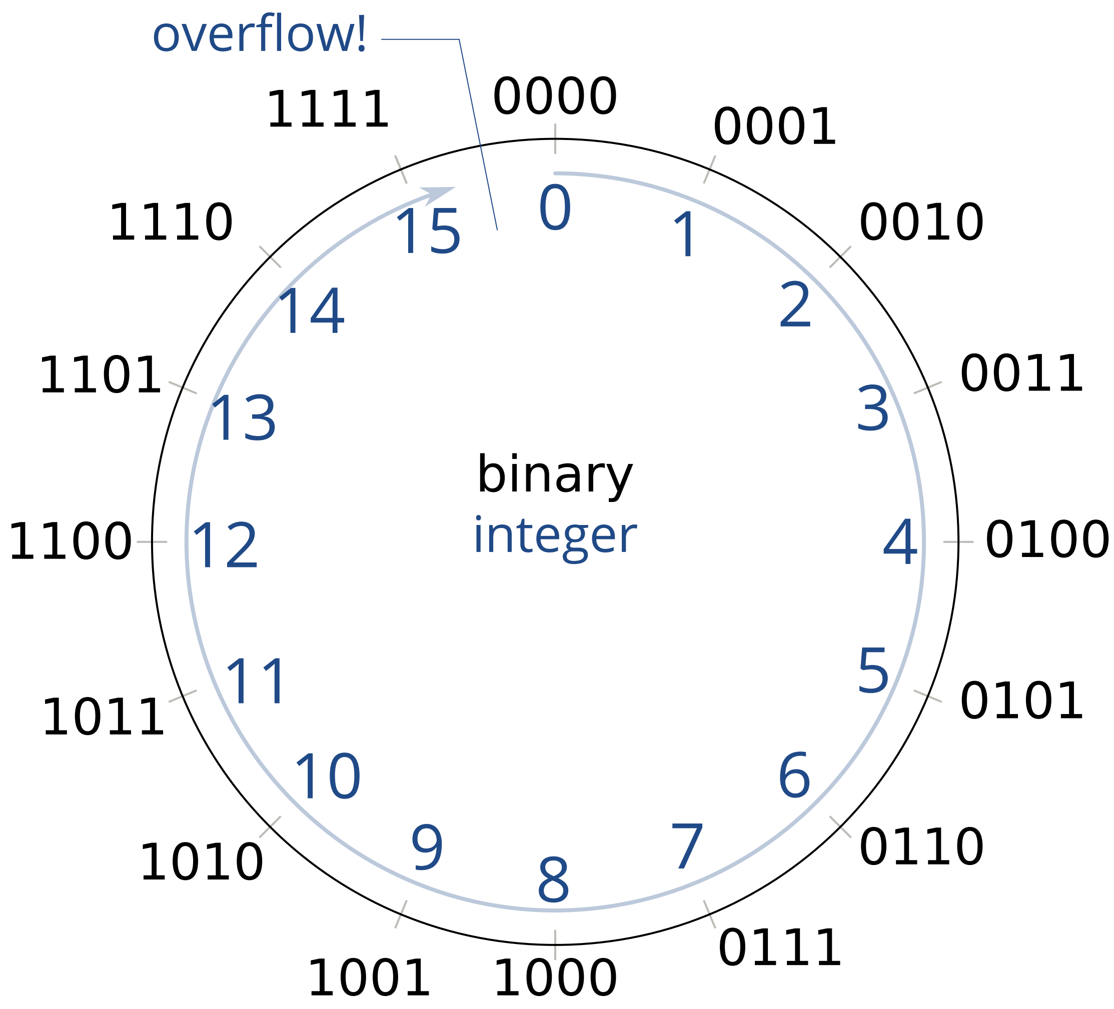
8 bits? $\overset{128}{1}111\ 1111 \to \overset{256 = 2^8}{1\ 0000\ 0000}$

16 bits? $2^{16} - 1$

32 bits? $2^{32} - 1 \to 2^{30} \cdot 2^2 \approx 10^9 \cdot 4$

64 bits?

$$2^{10} = 1024 \approx 1000 = 10^3$$


overflow!
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
binary
integer

For Thursday

Read the book (1.5, 2.1-2.2) and complete the pre-class reading check

Reading check is due at **11AM** the day of class, so I can review it