ECSE-2610
Computer Components & Operations
(CoCO)

Course Overview

Number Systems:

Binary Representation & Arithmetic

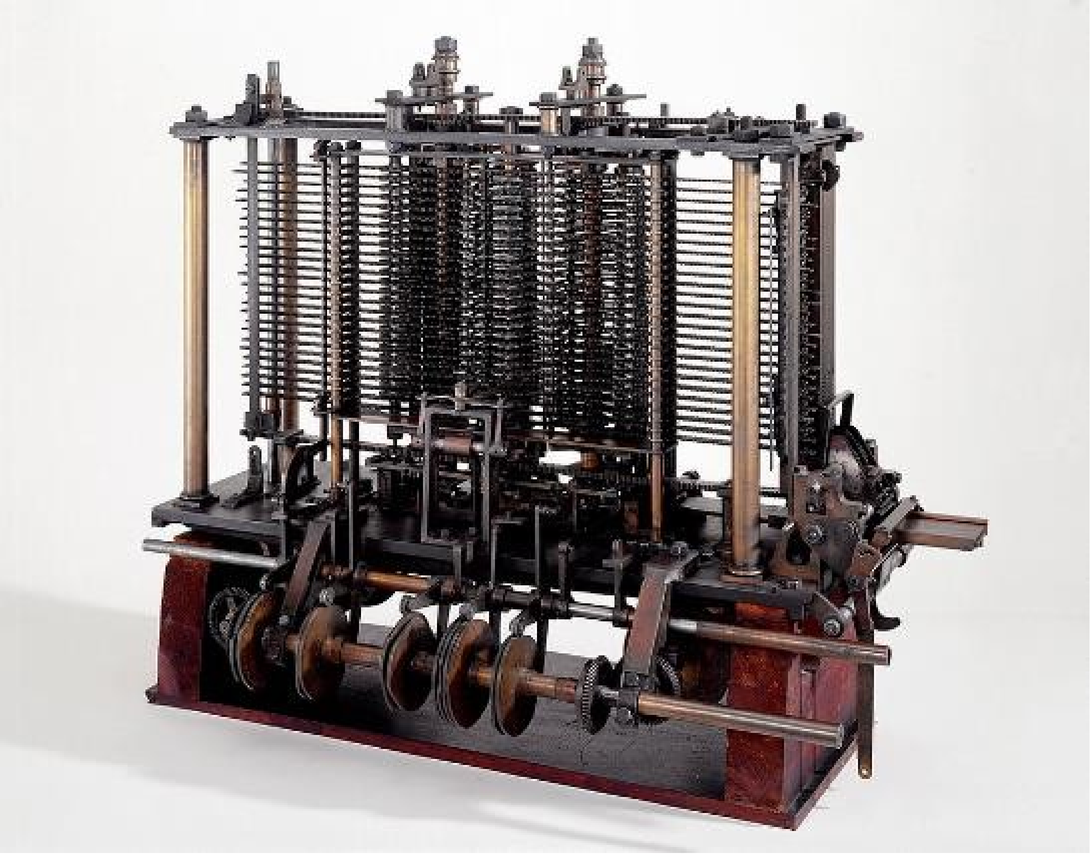
Introduction

Is this a computer? If so, what makes it one?

Apple iMac

Introduction

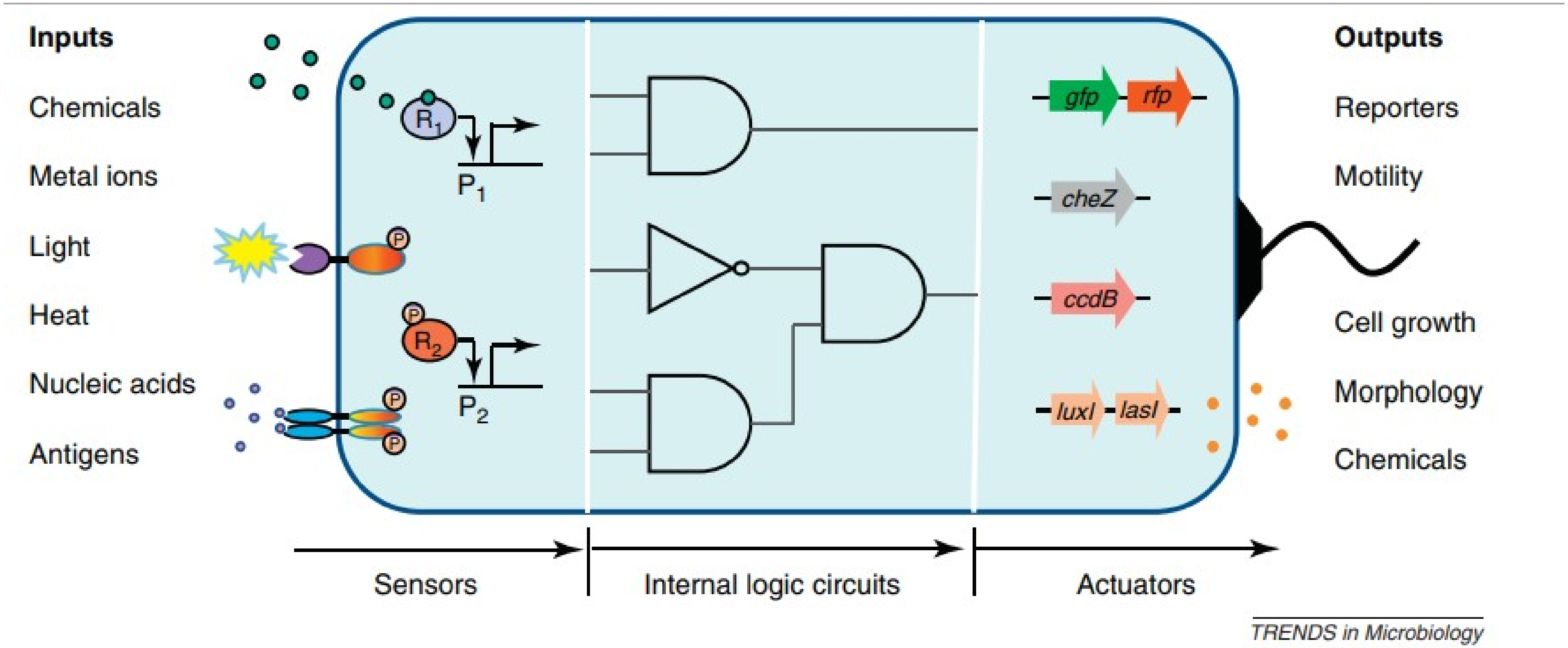
Is this a computer? If so, what makes it one?



Charles Babbage's Analytical Engine (1837)

Introduction

Is this a computer? If so, what makes it one?



[Genetic Circuits / Genetic Logic]

Course Objectives

- Fundamental concepts of digital logic and computer systems
  - Number systems and their representation
  - Operations in binary variables and Boolean algebra

- Basic techniques of analysis and design of digital logic circuits
  - Logic gates: ANDs, ORs, XORs, etc.
  - Combinational circuits (encoders, multiplexers, bus drivers, ALUs)
  - Sequential logic circuits (Flip-Flops, Counters, Finite State Machines)

  - Hands-on experience designing, modeling and building digital logic circuits

Basic Course Logistics

- Course materials will be posted on my website:

https://dylanrees.github.io/coco/

- This includes containing lecture slides + recordings, lab documents, homework solutions, and other necessary info.

- Coursework submission and grading will be handled via Gradescope

Gradescope entry code: **G3DXZW**

Materials

Suggested Textbook

- John F. Wakerly, Digital Design: Principles and Practices, 4th Edition, Prentice Hall. (Book website: http://www.ddpp.com/)
- Suggested readings from Wakerly posted on shared drive

Software

- Xilinx Vivado Design Suite 2016.2 (Installation instructions on shared drive)
- Digital circuit simulation software (we have used LogicWorks in the past. TBA this semester)

Course Components

Lectures (12pm-1:50pm Tue / Fri)

- Will be recorded and saved on the shared drive
- Will sometimes include ungraded group activities

Metacognition Journals

- Reflections on in-class learning. Consist of the class's "attendance"
- Due weekly starting at the end of Week 2

Homework

- You may work on homework together but **you must submit original work**
- Test newly-covered concepts from lecture
- Open-book and open-notes
- Some homework problems will be discussed during lecture

Metacognition Journals

- Metacognition is thinking about thinking, typically to improve or control thought
- Allow you an opportunity to synthesize, organize and apply what you are learning in lecture
- Assigned on Friday as a reflection on the week's lectures; due the following Tuesday (online Gradescope assignment)

Q1
1 Point

Was there something you particularly enjoyed about this week's lectures and/or something you learned that really resonated with you this week? If not, why were this week's lectures uninformative/unexciting? You should reference both lectures in your answer.

Metacognition Journals

Metacognition journal entries do not have a “right answer.” They will be graded for:

- Completeness (address both lectures and engage meaningfully with the lecture content)
- Effort (earnest thought put into each response)
- Clarity of writing

They will NOT be graded for:

- Verbosity
- Spelling/grammar (but do your best)
- Flattery towards the professor

Metacognition Journals

Example of a good metacognition entry:

"On Tuesday and Friday we covered Karnaugh maps. The first lecture was confusing because I had not encountered the concept before. I could have used more example exercises during the first lecture. On Friday, however, we did more examples and it started to click for me. I am still struggling to understand how to implement "don't cares" in the maps."

Metacognition Journals

Example of a bad metacognition entry:

"tbh didn't pay attention much lol"

Metacognition Journals

Example of a bad metacognition entry:

"Tuesday's lecture was awesome.   I learned about Karnaugh maps"

Course Components

- **Studio Sessions (Wednesday)**
  - Section 01 at 12-1:50am, Section 02 at 3-4:50pm, Section 03 at 5-6:50pm
  - Location is JEC 6309
  - Group work involving short write-ups submitted
  - Will cover VHDL/FPGA and digital logic simulation
  - Projects must also be demonstrated to teaching staff
  - You may also use staff office hours to get studio session work checked off
  - Labs use Basys 3 board, but you are **not** required to purchase one
  - First Studio Session is **next week**
  - In the meantime, **begin working on Studio Session 0**

Course Components

- **VHDL Exercises**
  - Will help you to learn VHDL programming
  - Act as a supplement to the lab section
  - Watch a short lecture on YouTube, then complete a questionnaire on Gradescope
  - Easy assignment + easy grading

Omega Labs

Wednesday Lab	
Alpha	Omega
Studio 0 / VHDL Exercise 1	Studio 0 / VHDL Exercise 1
Studio 1	Studio 1
VHDL Exercise 2	VHDL Exercise 2
Studio 2	Studio 2
VHDL Exercise 3	VHDL Exercise 3
Studio A	Studio 3
VHDL Exercise 4	VHDL Exercise 4
-	Studio 4
Spring Break	
Logic Sim Exercise	Logic Sim Exercise
GM Week	
Studio B	Studio 5
-	-
Studio C	Studio 6
-	-
-	-

Omega Labs

- **After Studio 2,** students have a choice between Alpha and Omega Labs (and can switch back and forth between assignments)
- Alpha Labs are more procedural; Omega Labs are more open-ended
- For more detail on Omega Labs, please read the syllabus

Course Components

Exams

- Test your accumulated knowledge of CoCO concepts
- Exams will be open-notes
- Occur in class in the same room as the lecture on specified dates (check course schedule)
- **Every Omega Lab completed will allow you to skip ~10% of the final exam (~30% if all omega labs are completed)**

Grading breakdown

Unit 1 Exam	(20%)
Unit 2 Exam	(20%)
Unit 3 Exam / Final Exam	(20%)
Homework Assignments	(10%)
Studios	(20%)
HDL Assignments	(5%)
Metacognition	(5%)

At the end of the semester, **one homework and one Metacognition entry** will be dropped from each student's average.

I am *considering* implementing some additional attendance score (pop quizzes) if lecture attendance becomes low. If so, this would be averaged into the Metacognition score.

Accommodations

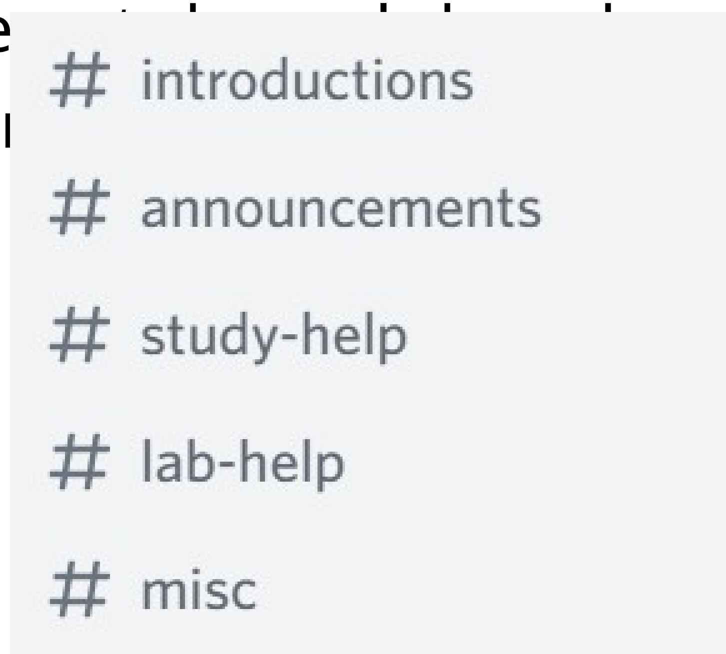
- **Let me know ASAP if you need an accommodation**
- **Contact Dean of Students Office for details**

Course Agreements

- **Must be submitted on Gradescope by each student.**
- **We will review this as a class.**

Discord

- Our primary mode of communication outside the class.
- All teaching staff and students will be on the Discord.
- Several diffe… created for different pu…



Discord

#introductions

- Go here when you first join the server.
- Post your name, pronouns (he/him, she/her, they/them, etc.) email, and section.
- We use this to map your Discord name to your other names/accounts.

Discord

#announcements

- Used to communicate course information.
- Students can’t post here; only the TAs and I can.
- However, students can comment or react to posts.

#study-help

- A place for students to ask questions about homework, lectures, or exams (the non-lab components of the class.)
- Teaching staff will monitor these channels closely during their office hours.

Discord

#lab-help

- A place for students to ask questions about the lab-based components of the class.
- Teaching staff will monitor these channels closely during their office hours.

#misc

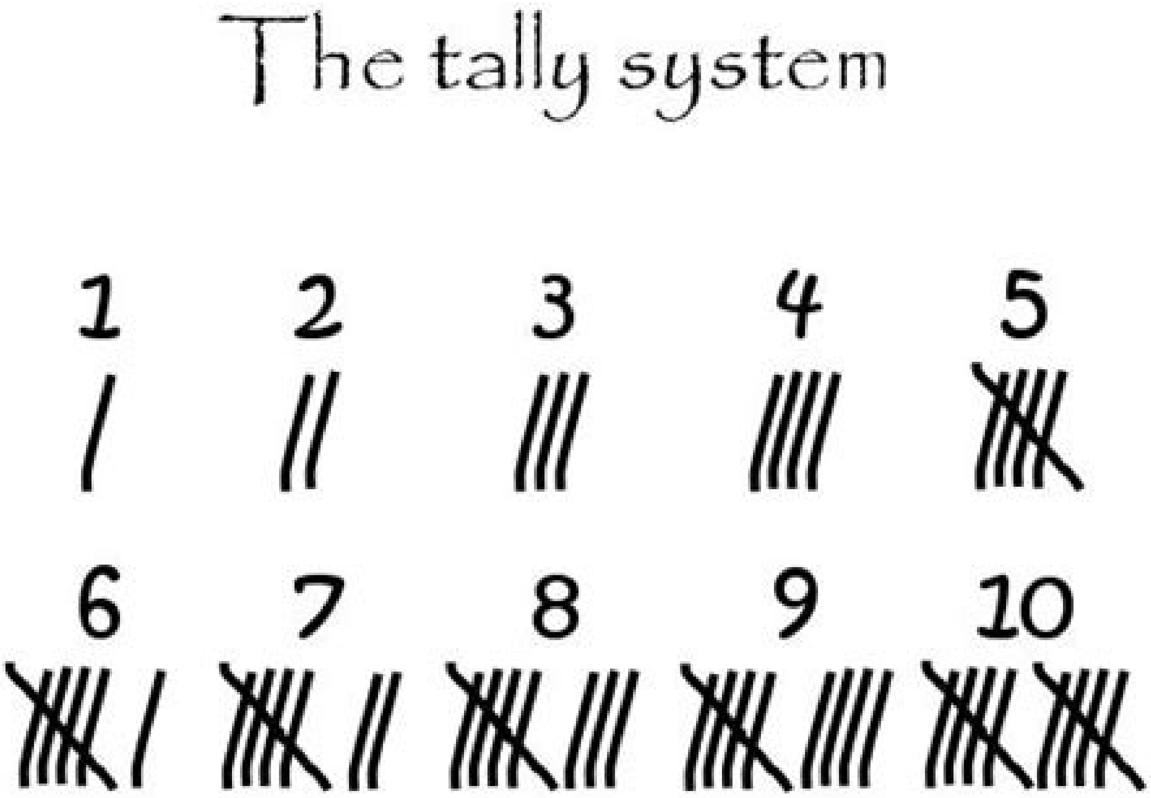
- A place for anything that doesn't fit in another channel or isn't directly related to the course content.

We will now take time to allow students to join the Gradescope and Discord.

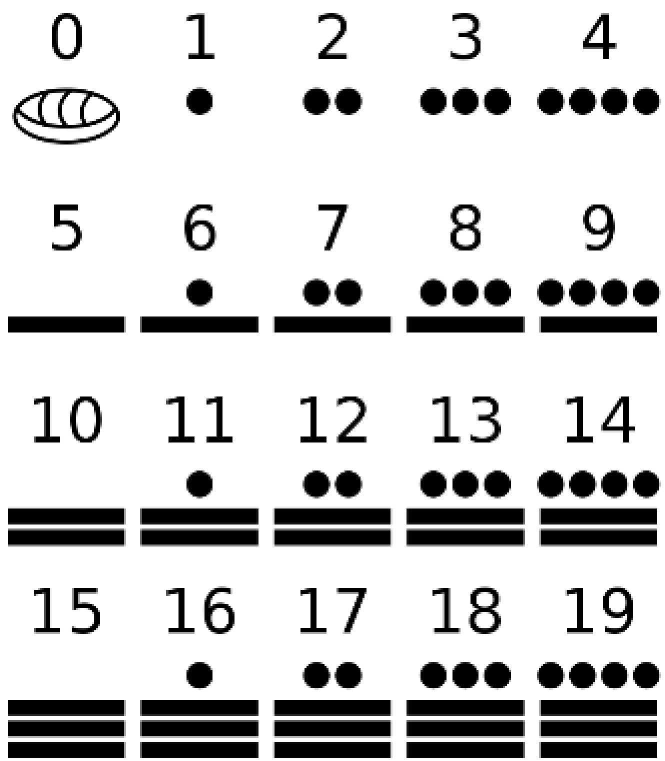
Gradescope entry code:
G3DXZW

Discord link (QR at right): https:
b45J

Number Systems



Number Systems



Mayan numerals

Number Systems

```
00101110 11110011 00101110 00101110 00101110 00101110 00111100
01010111 01100011 00101110 00101110 00101110 00101110 00101110
00101110 00101110 00101110 00101110 00101110 00101110 00101110
01000000 00101110 00101110 00101110 01110011 00101110 00101110
00101110 00101110 01100100 00101110 00101110 01000111 01001000
01100100 00101110 00101110 01010011 00101000 00101110 00101110
00101110 00101110 01110011 00101110 00101110 00101110 00101110
01001000 01100101 01101100 01101100 01101111 00100000 01010111
01101111 01110010 01101100 01100100 01001110 00101000 00101110
00101110 00101110 00101110 00101000 00101110 00101110 00101110
00101110 00101000 00101110 00101110 00101110 00101110 00101000
00101110 00101110 00101110 00101110 01110011 00101110 00101110
00101110 00101110 01101000 01100101 01101100 01101100 01101111
01110111 01101111 01110010 01101100 01100100 00101110 01110000
01111001 01110100 00101110 00101110 00101110 00101110 00111100
01101101 01101111 01100100 01110101 01101100 01100101 00111110
00101110 00101110 00101110 00101110 01110011 00101110 00101110
00101110 00101110
```

Binary machine code

Decimal Representation: 10 digits = {0,1,2,...,9}

Positional Number Notation:

Weight of each digit determined by its position.

Example: 246 = 200 + 40 + 6

$= 2 \times 10^2 + 4 \times 10^1 + 6 \times 10^0$

For Base 10 representation:

$$N = \Sigma N_i \times 10^i$$

where each $N_i \in \{0, 1, 2, . . . , 9\}$ is the weight on the base raised to the exponent i.

Negative exponents yield decimal fractions:

$$0.35 = 3 \times 10^{-1} + 5 \times 10^{-2}$$

Number Systems

Binary Representation: 2 digits = {0, 1}

Positional Number Notation for the Binary Representation

$$N = \Sigma N_i \times 2^i \text{ with each } N_i \in \{0,1\}$$

<u>Example</u>: Converting 111001_2 from binary to decimal

$111001_2 = 100000_2 + 10000_2 + 1000_2 + 000_2 + 00_2 + 1_2$

$= 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$= 1 \times 32 + 1 \times 16 + 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1$

$= 57_{10}$ Easy!

Hexadecimal (Hex, Base 16, Ox)

Representation:

16 digits = {0,1,2,…,9,A,B,C,D,E,F}

Positional Number Notation for the Hexadecimal Representation

Note: A - F represent the decimal values 10 - 15, respectively.

$N = \Sigma N_i \times 16^i$ with each $N_i \in \{0,1,2,...,9,A,B,C,D,E,F\}$

Example: Converting $8A9B_{16}$ from Hex to Decimal representation

$$8A9B_{16} = 8000_{16} + A00_{16} + 90_{16} + B_{16}$$

$$= 8 \times 16^3 + A \times 16^2 + 9 \times 16^1 + B \times 16^0$$

$$= 8 \times 4096 + 10 \times 256 + 9 \times 16 + 11 \times 1$$

$$= 35483_{10}$$

Still easy! (if you remember $16^3 = 2^{12} = 4096$)

Converting Decimal to Binary: Successive Division

Idea: Use remainders from dividing the decimal number by powers of 2.

Example: Convert 57_{10} to Binary

57 / 2 = 28, remainder = 1 (binary number will end with 1)
28 / 2 = 14, remainder = 0
14 / 2 = 7, remainder = 0
7 / 2 = 3, remainder = 1
3 / 2 = 1, remainder = 1
1 / 2 = 0, remainder = 1 (binary number will start with 1)

quotient = 0. Time to stop.

Collecting the remainders (from the bottom up), $57_{10} = 111001_2$

Number Systems

Number Systems

Conversion Among Bases

In general, with positional number notation and the known decimal weights for each position in any arbitrary base, it is easiest to convert other bases to decimal.

This was demonstrated in each previous example where the decimal value was found using the equation for base B:

$$\Sigma\, N_i \times B^i \quad \text{where } N_i \in [0, 1, 2, \ldots, B\text{-}1]$$ and

substituting the equivalent decimal weight for the digits N_i and the decimal value of B^i.

Conversion among the 2^n bases (binary, hexadecimal, octal, etc.) is most easily achieved by first converting to binary, regrouping into n bits (starting at the base point) and then converting the groups of bits into the proper digits.

Converting Between Hexadecimal and Binary

Example: Convert $1110110001110001 01_2$ to hexadecimal.

111011000111000101_2

$0011 \quad 1011 \quad 0001 \quad 1100 \quad 0101_2$

3 B 1 C 5

So $111011000111000101_2 = 3B1C5_{16}$

s table!
versions
decimal
easy.

Hex	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8

7 8 1000

Even easier to go the other way (Hex to Bin)

Wrap Up

To do:

- Complete the Course Agreements on Gradescope (due in 2 weeks)
- Complete Studio 0 (due next Wed)
- Complete HDL Assignment 1 (due next Wed, easier if you complete Studio 0 first)
- Watch Studio 1 introductory video (by next Wed)