



**Computer Systems
from the Ground Up**

Winter 2023

<https://cs107e.github.io>

Who?

Pat



Julie

You!

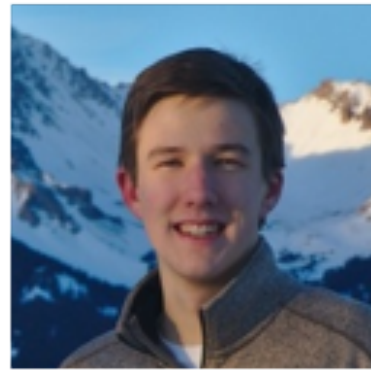
Intrepid young padawans



Maria



Liana



Keegan



Anna

Weekly Cadence

Each week has a focus **topic**

Pair of coordinated **lectures** on Fri and Mon

Lab on Tue/Wed evening

Assignment handed out Wed after lab, YEAH session Thu, due following Tuesday 5pm

Staying on pace leads to best outcomes!

Lectures

Attendance is **necessary**

Content is unique to our course, no textbook

The readings/slides are not a standalone resource

Lectures are not recorded

In-person attendance allows you to participate, ask questions,
stay on schedule

Labs

Attendance is **mandatory**

Guided exercises, work with peers, **check in** with staff

Finish lab **ready** for assignment, esp. experience with tricky parts (hardware/software interface)

Philosophy: lots-of-help, hands-on, collaborative

Lab room: Gates B02

Assignments

7 weekly assignments

Build on each other, complete full system

Assignment specifications

Core (required, tight spec, guided steps)

Extension (optional, opportunity for your exploration/creativity)

Revise and **resubmit** to address issues in core functionality

Project

Design and build **your own system**

Learning community

Stay **connected**

Participate in lecture

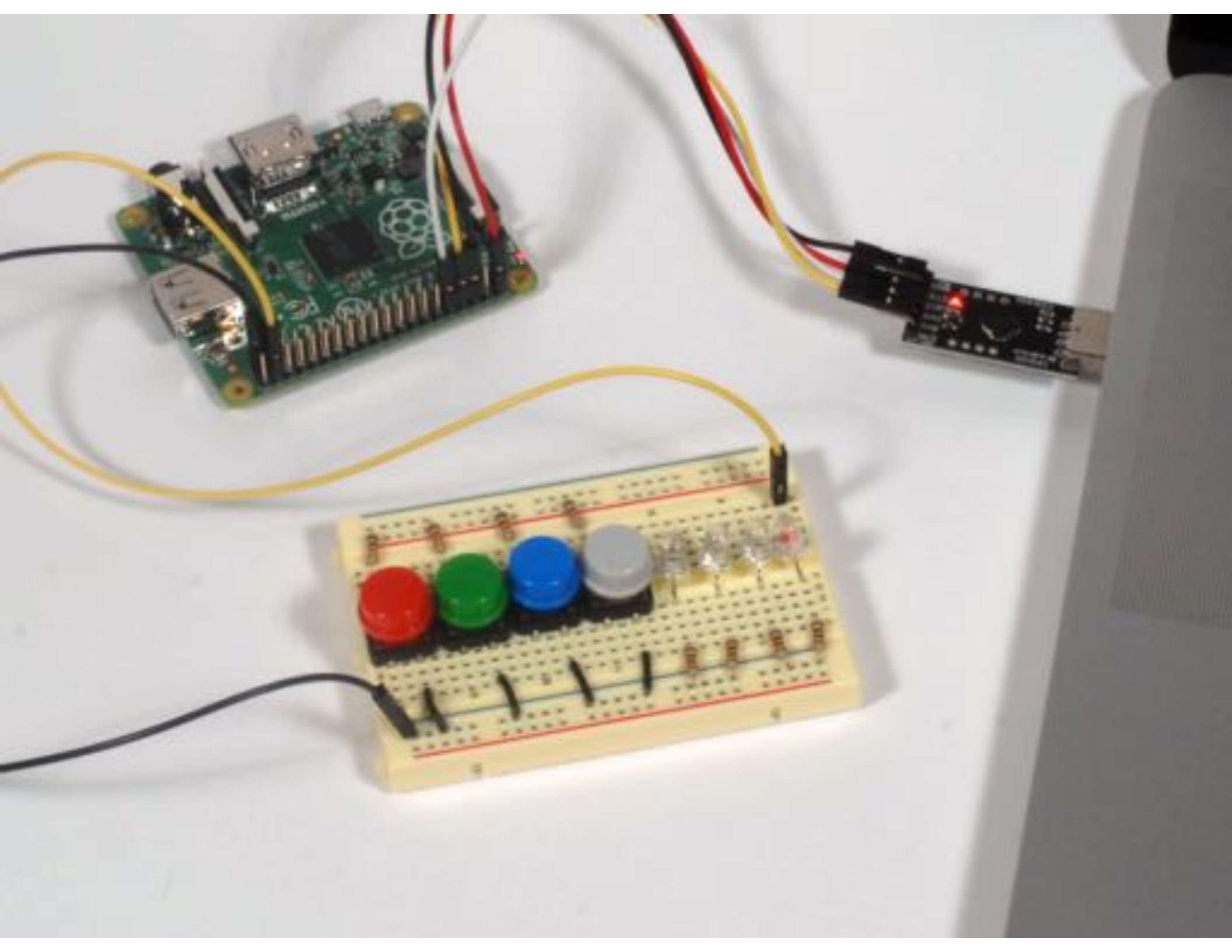
Collaborate in lab

Discuss on Ed forum

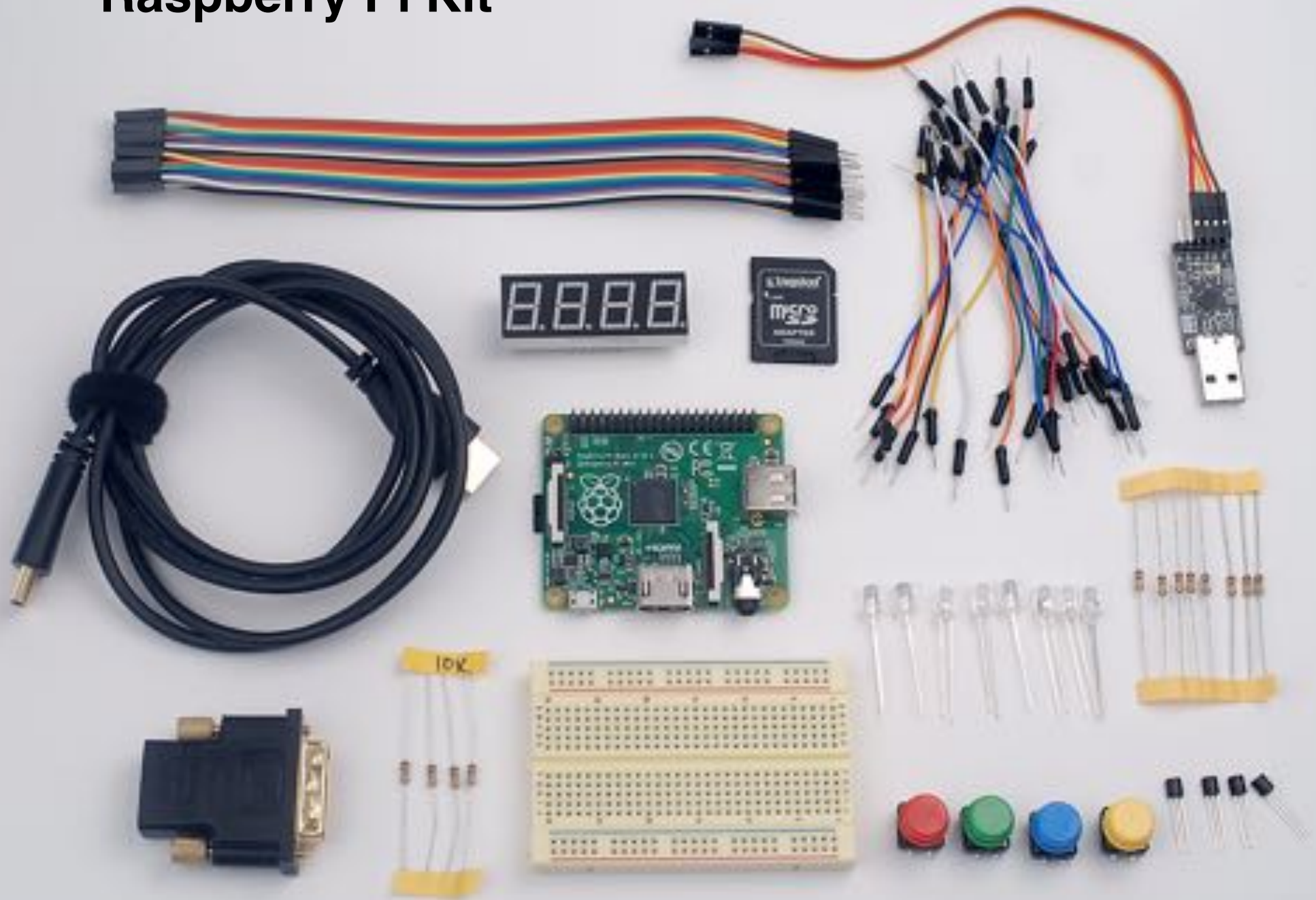
Come to office hours

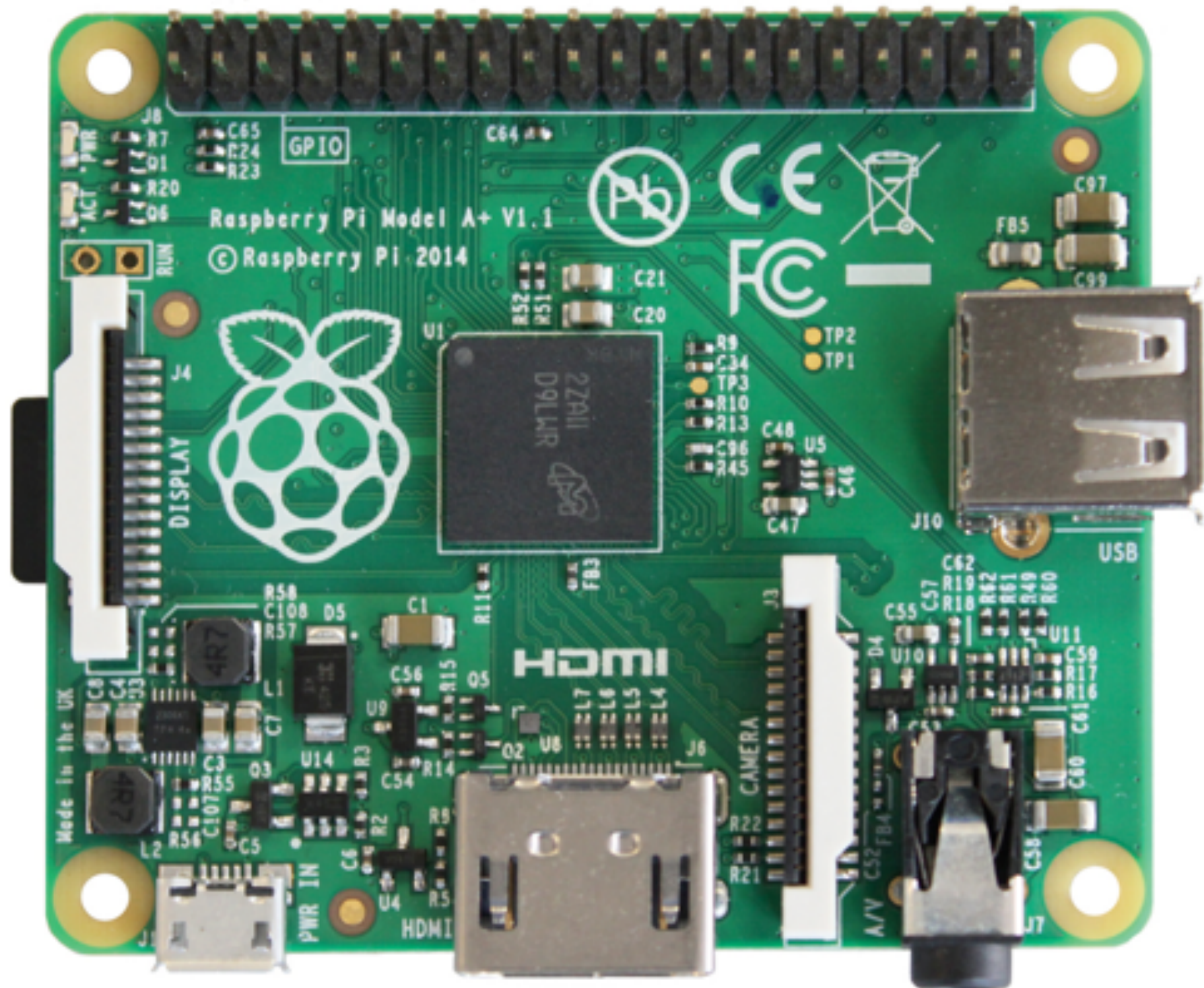
Meet up in lab room

Be **curious**. Learn by **doing**. Ask for and offer **help**.



Raspberry Pi Kit





AWAKE HCl
V-OF3
1439 1-6

27/10

MICRO SD CARD

R12 C6 C17
C39 C36
C94 C95 F82 C69
C51 C49 C18 C37
C9 C9 F81 C14 C25
C13 C12 C35
C45 C29 C30

4090AAC

TRST_N
TDI
TDO
TMS
TCK
GND

F1

First Week

Unix Command Line

Moving around the file system

Creating, moving, and deleting files

Compiling and running programs

Profiles and paths

Guide: <https://cs107e.github.io/guides/unix/>

Note: Watch cs107 UNIX videos!

Essential Tools

git

- **git add/commit/push/pull**

pick/learn an editor

- **vim, emacs, sublime, ...**

Lab 0 on Tue/Wed

Before lab

- Install your development environment (follow steps in install guide <http://cs107e.github.io/guides/install>)
- Read and understand our guides on background topics (electricity, numbers, unix command line)

During lab

- Establish comfort with background topics
- Practice with environment/tools, habits for productivity
- Meet one another!

Basic Electricity

Voltage and current

Ohms Law : $V = I R$

Power : $P = I V$

Driving an LED

Transistor switches

Breadboarding

Guide: <https://cs107e.github.io/guides/electricity/>

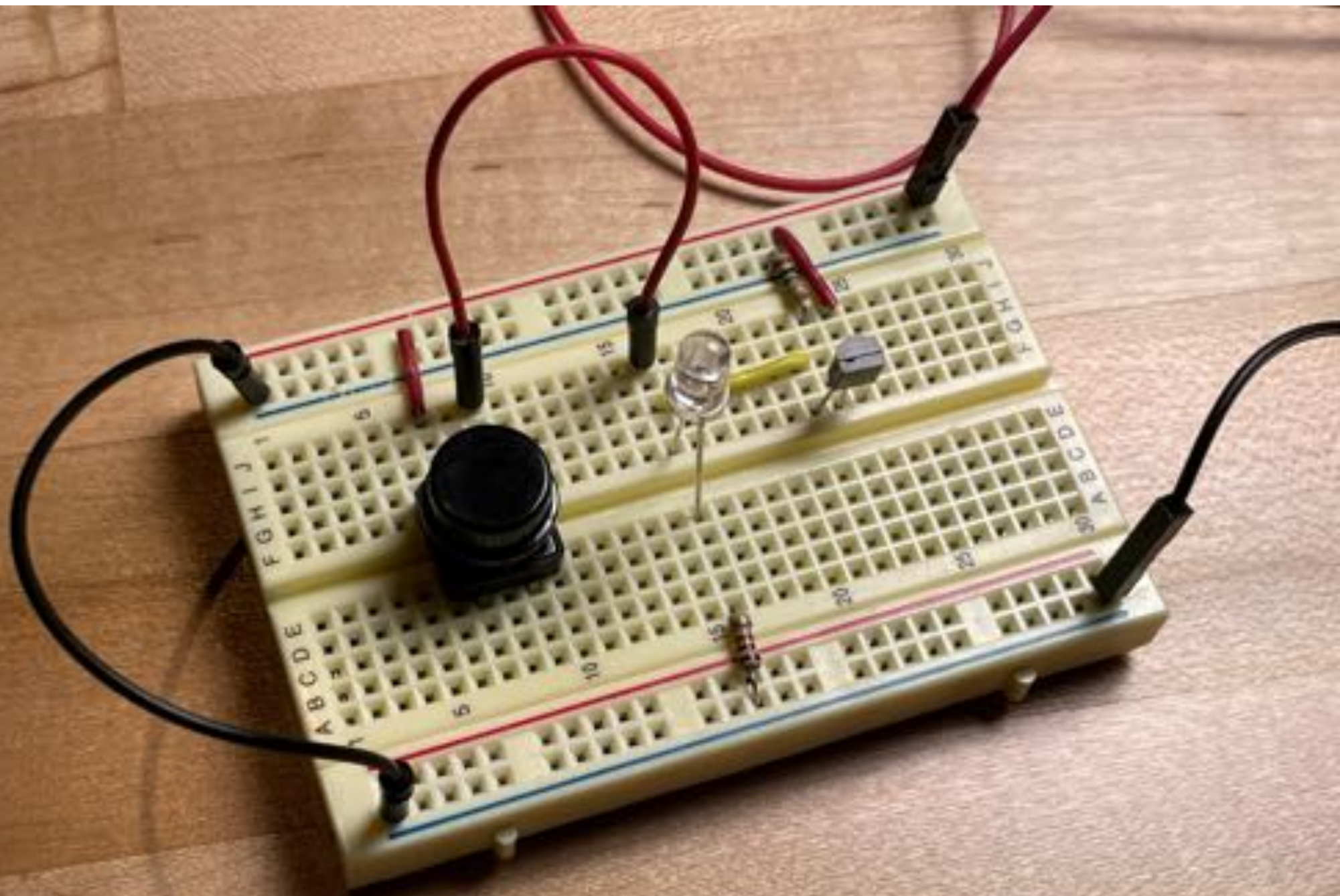
Number Representations

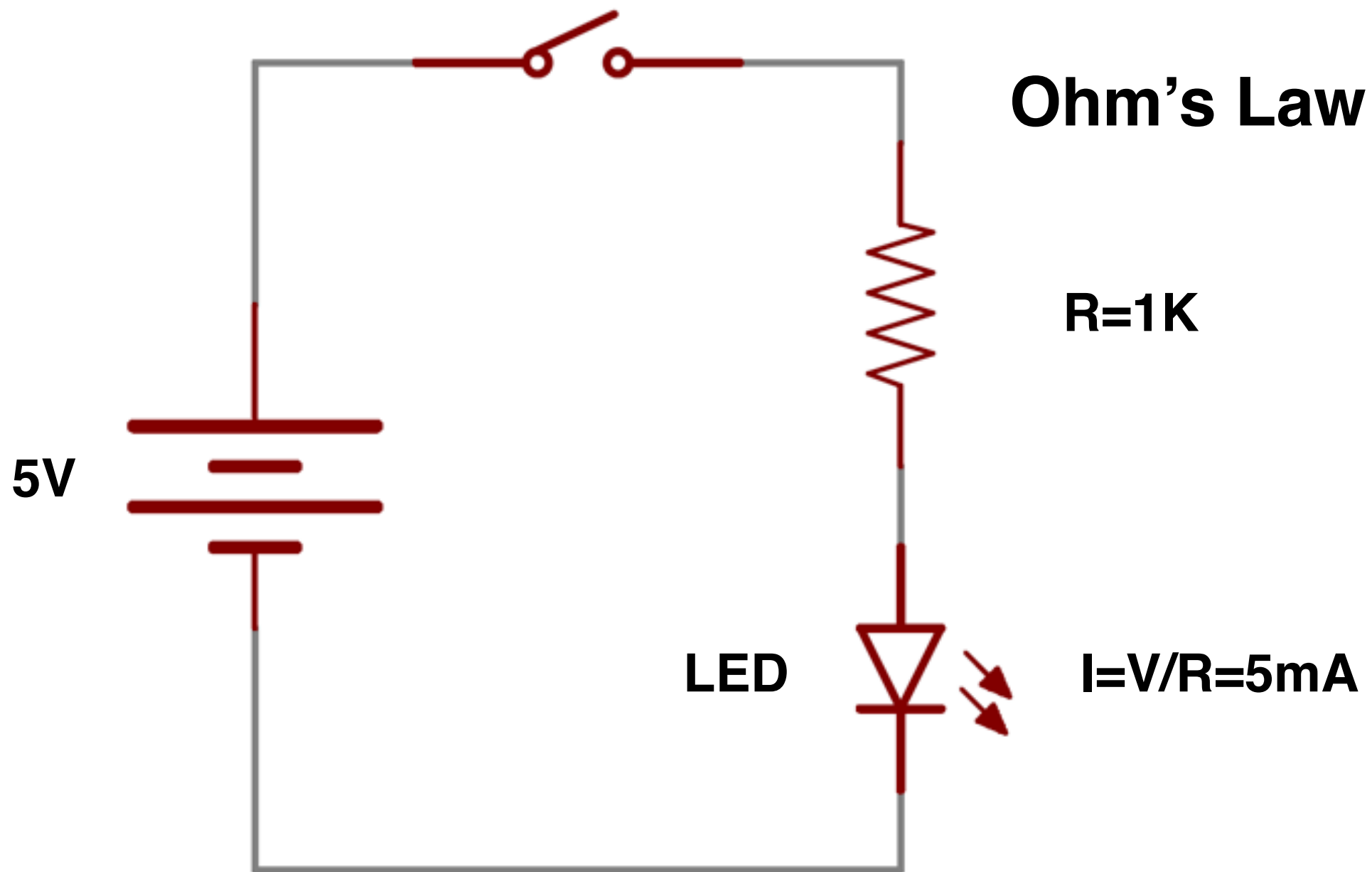
Binary representation

Hexadecimal

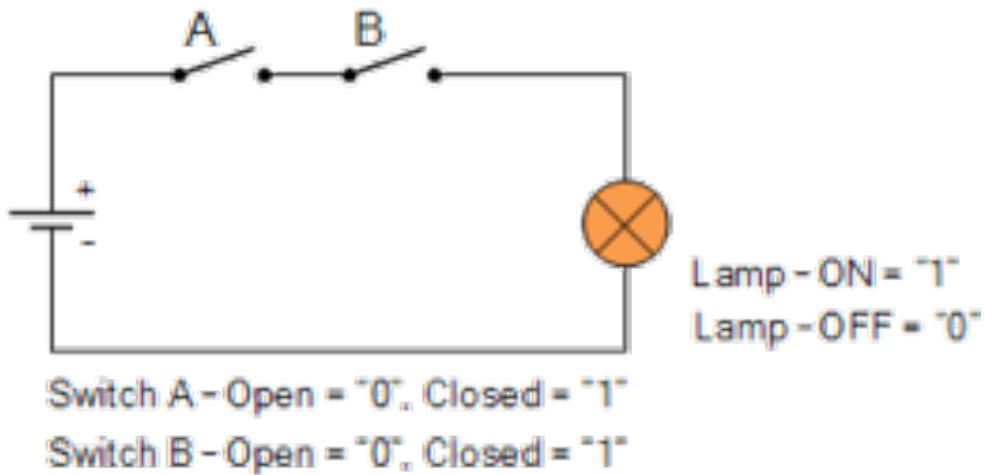
Digital logic and bit operators

Guide: <https://cs107e.github.io/guides/numbers/>

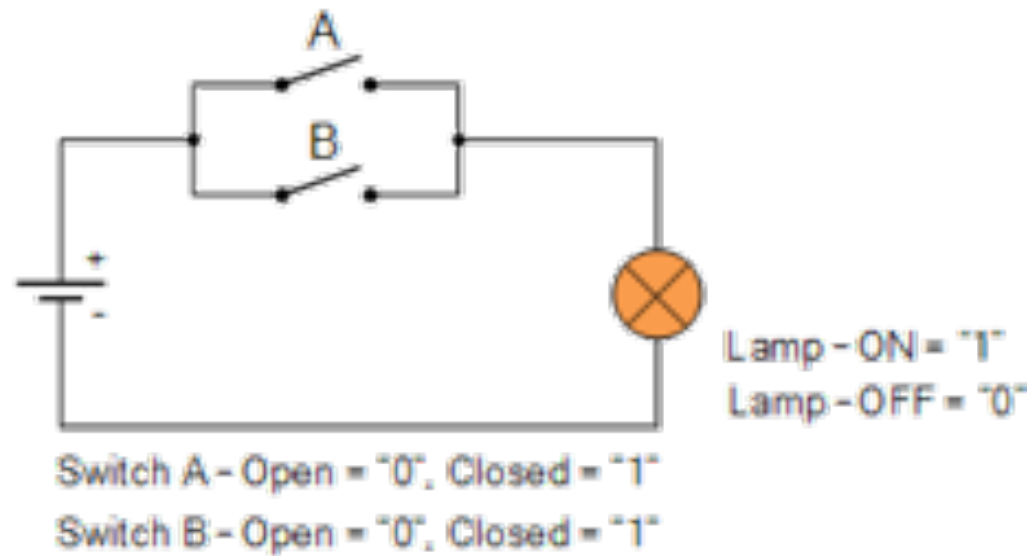




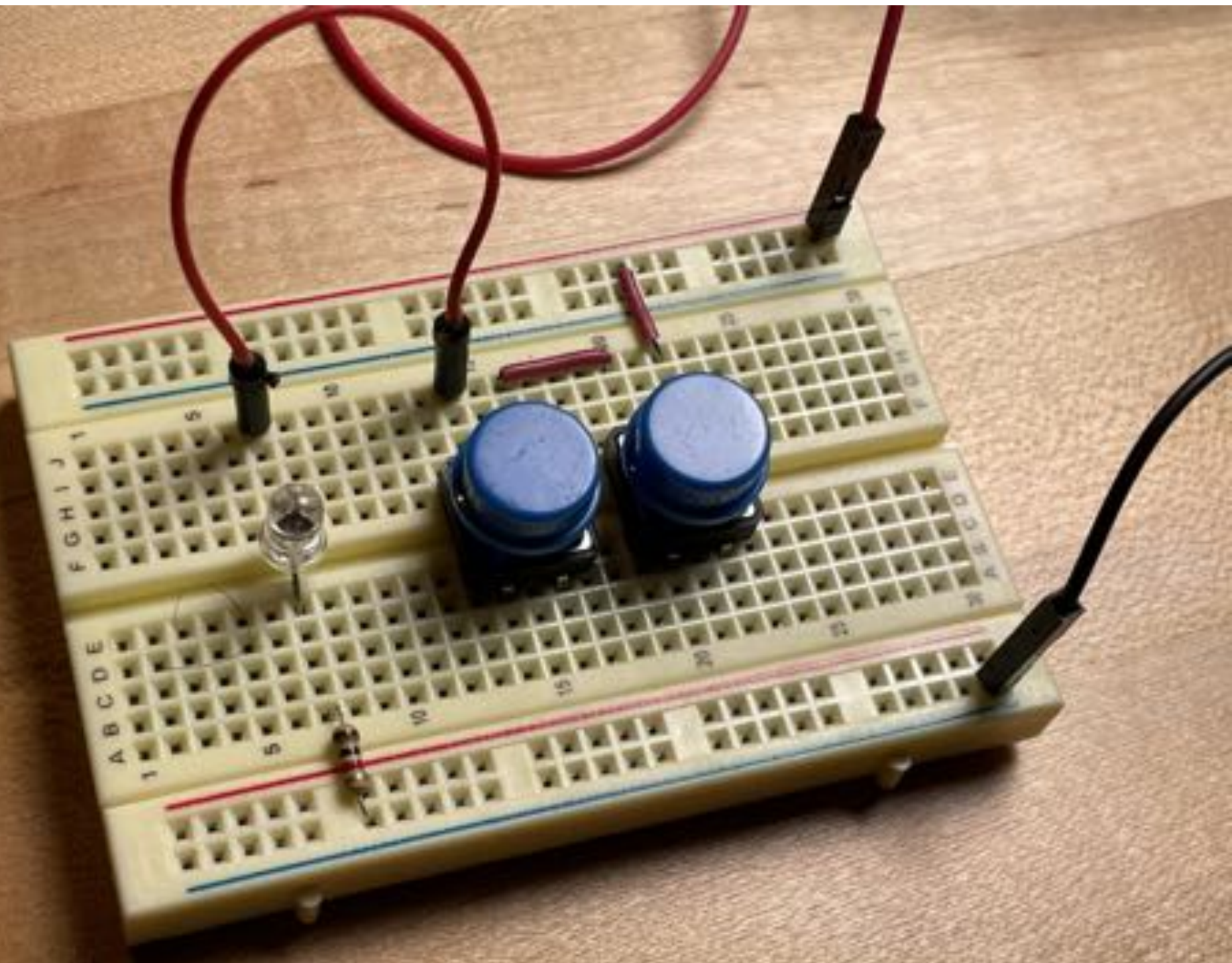
Logic with Switches

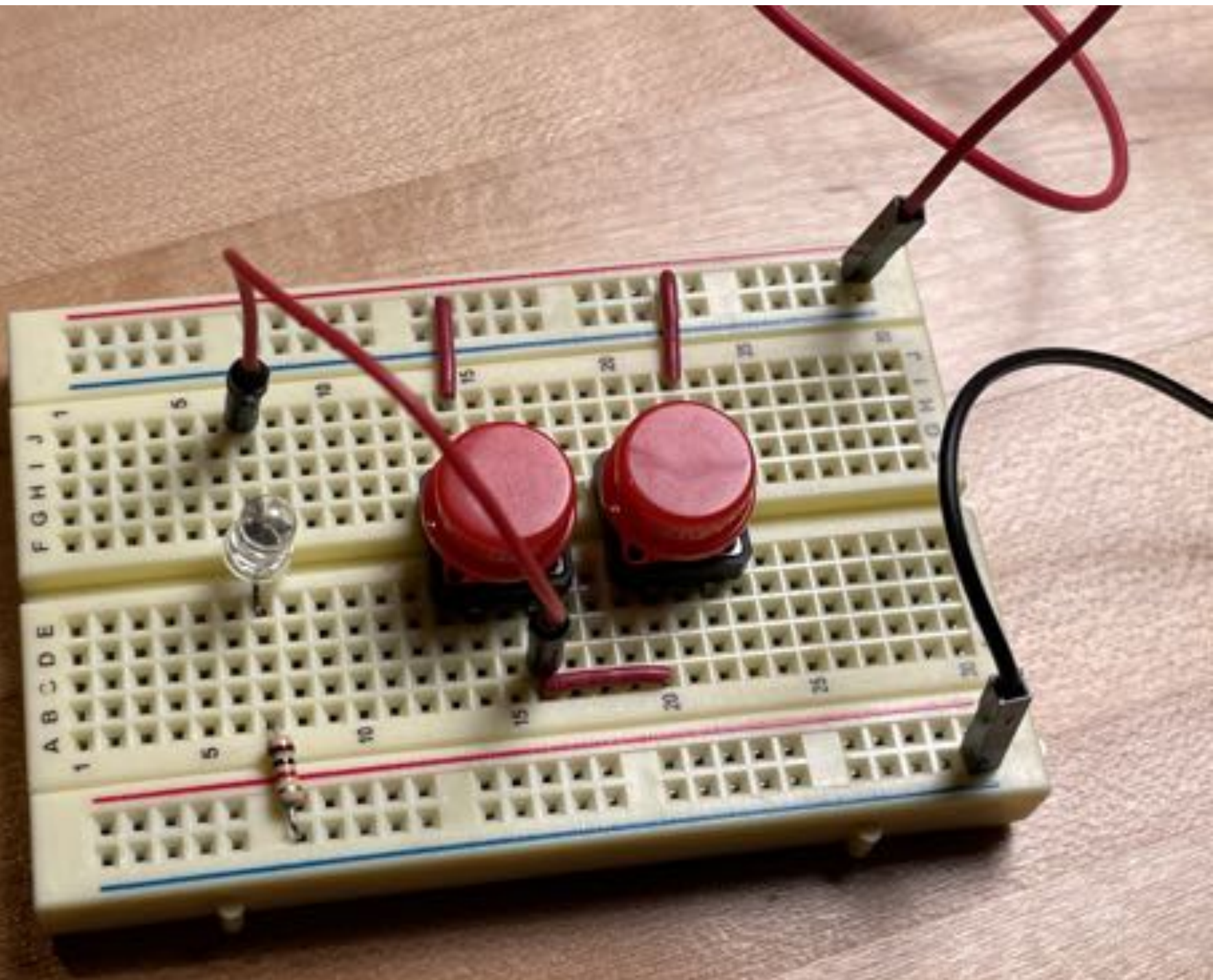


AND







OR





Digital Logic / Boolean Algebra

Name	NOT	AND	NAND	OR																																																			
Alg. Expr.	\overline{A}	AB	\overline{AB}	$A + B$																																																			
Symbol																																																							
Truth Table	<table><tr><th>A</th><th>X</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	X	0	1	1	0	<table><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	B	A	X	0	0	0	0	1	0	1	0	0	1	1	1	<table><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	B	A	X	0	0	1	0	1	1	1	0	1	1	1	0	<table><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	B	A	X	0	0	0	0	1	1	1	0	1	1	1	1
A	X																																																						
0	1																																																						
1	0																																																						
B	A	X																																																					
0	0	0																																																					
0	1	0																																																					
1	0	0																																																					
1	1	1																																																					
B	A	X																																																					
0	0	1																																																					
0	1	1																																																					
1	0	1																																																					
1	1	0																																																					
B	A	X																																																					
0	0	0																																																					
0	1	1																																																					
1	0	1																																																					
1	1	1																																																					

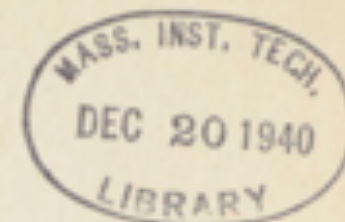
C syntax

$\sim A$

$A \& B$

$\sim(A \& B)$

$A | B$



A SYMBOLIC ANALYSIS
OF
RELAY AND SWITCHING CIRCUITS

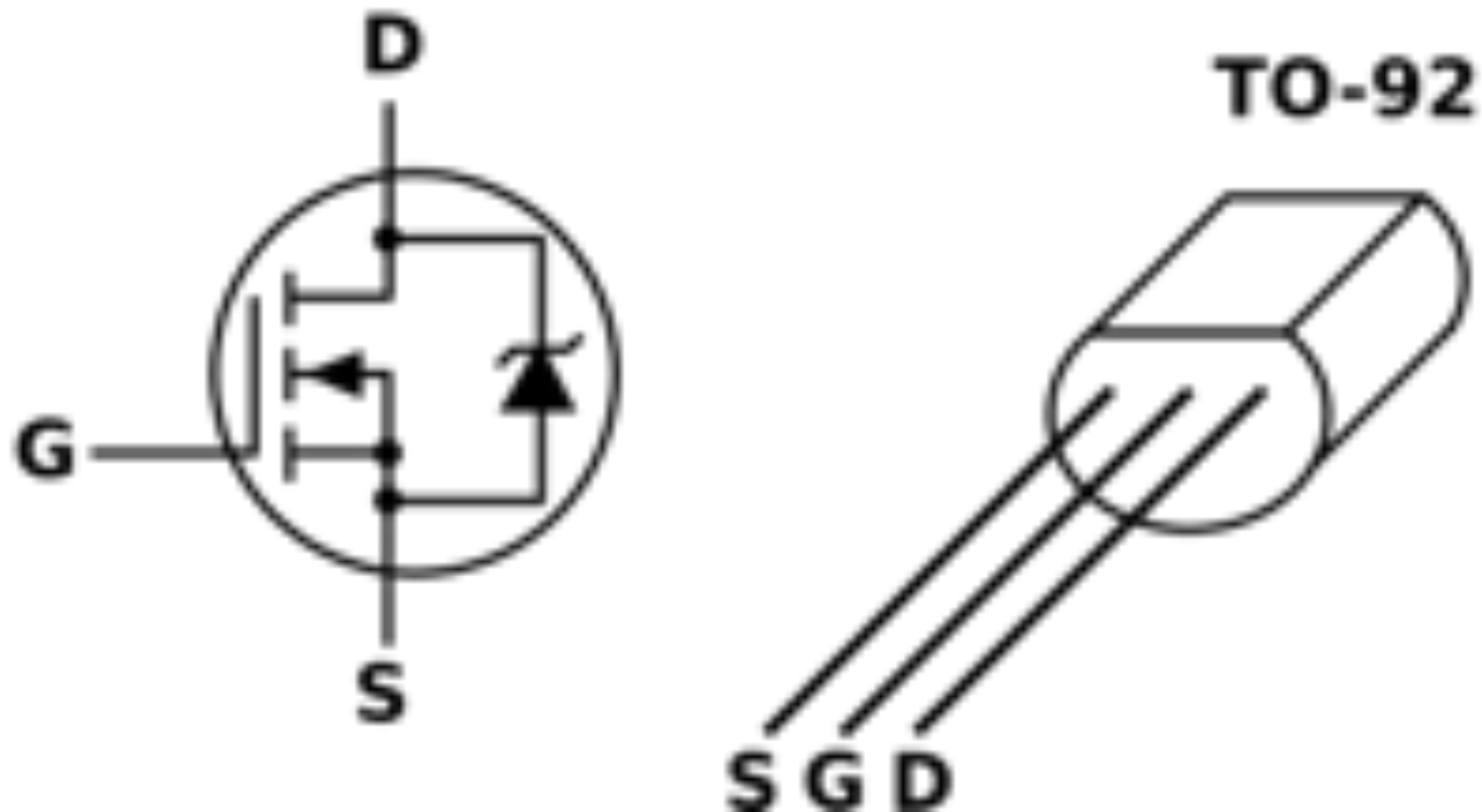
by

Claude Elwood Shannon
B.S., University of Michigan
1936

Submitted in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE

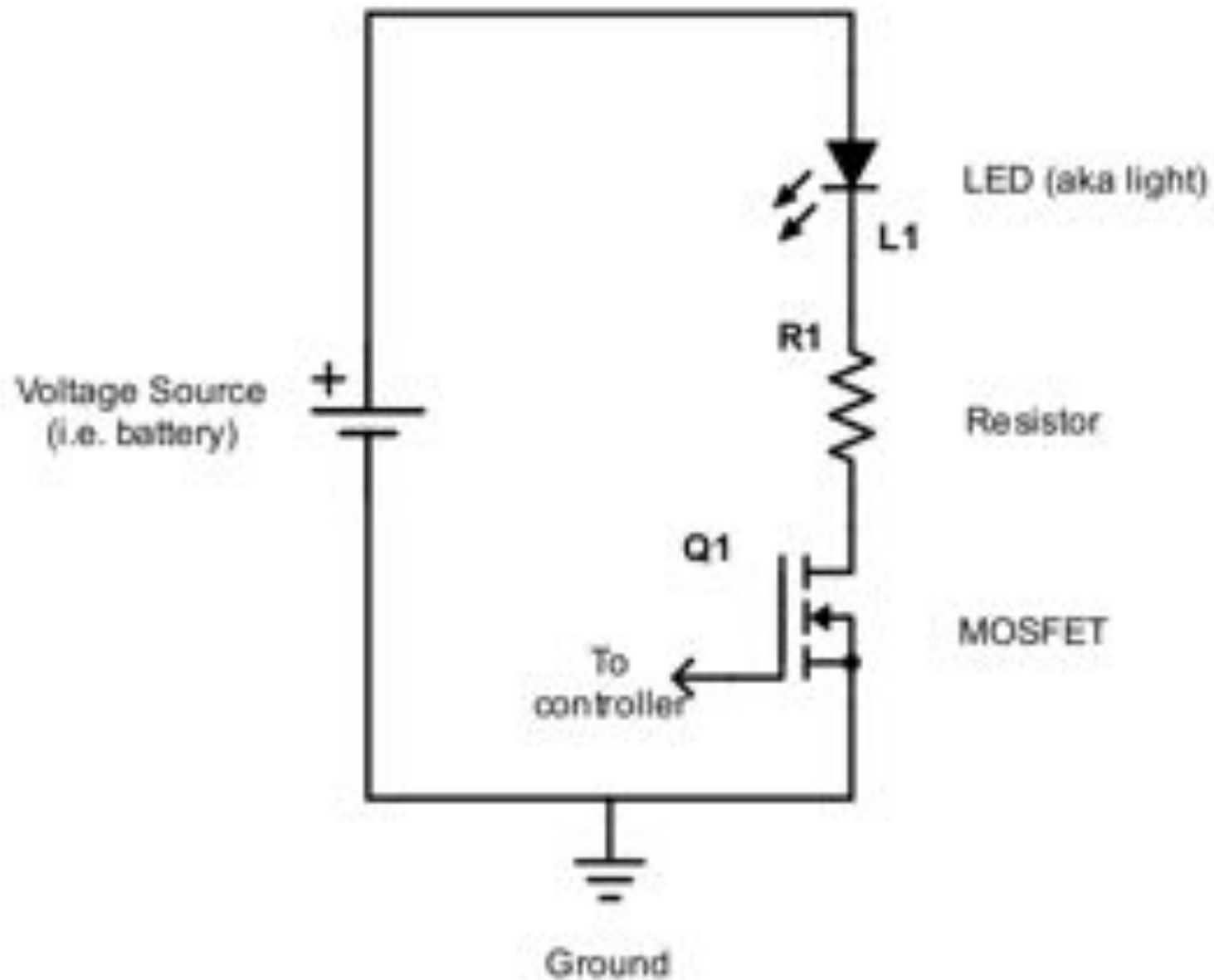
from the
Massachusetts Institute of Technology
1940

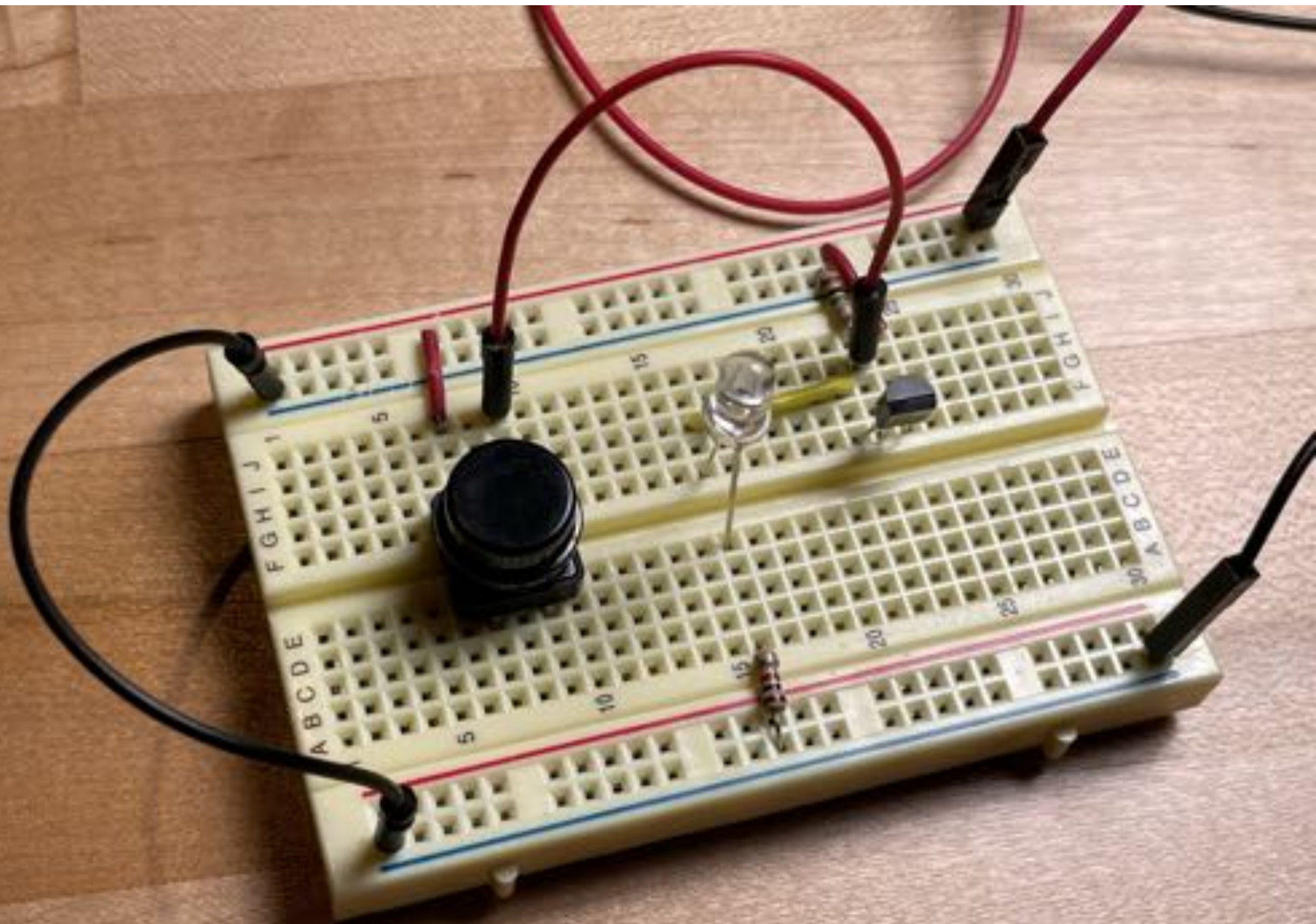
2N7000 N-Channel MOSFET



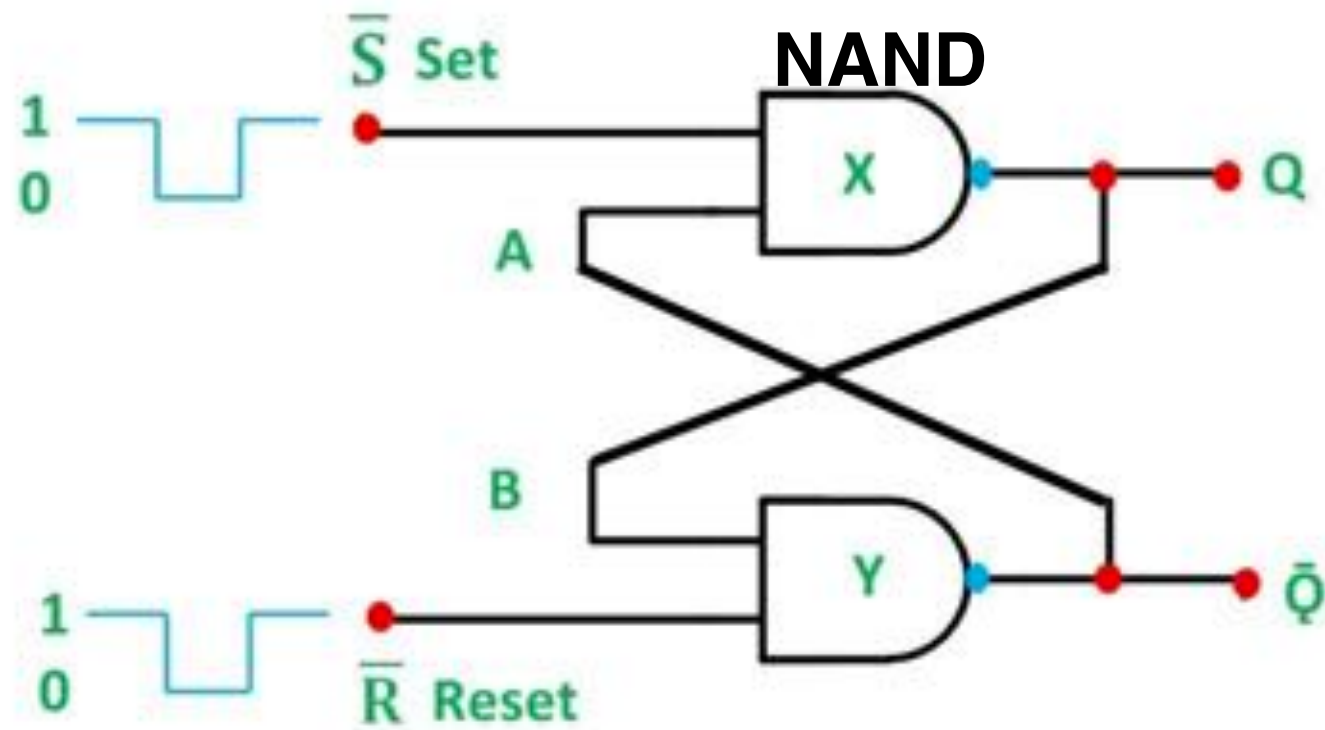
Drain (D), Gate (G), Source (S)

Transistor is a Controllable Switch





SR (Set/Reset) Flip-Flop



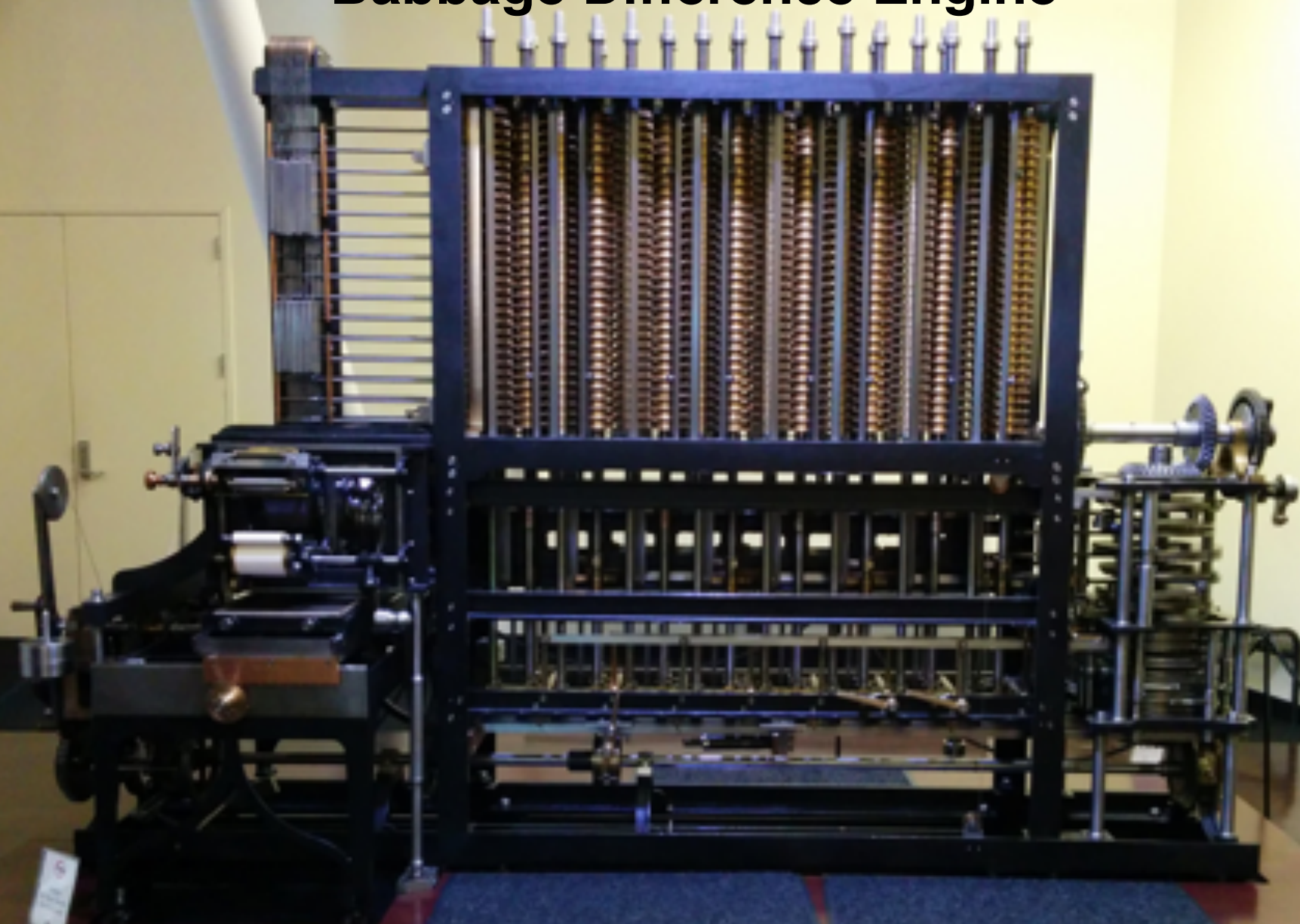
Circuit Globe

A	B	NAND
0	0	1
0	1	1
1	0	1
1	1	0

1-Bit Register/Memory

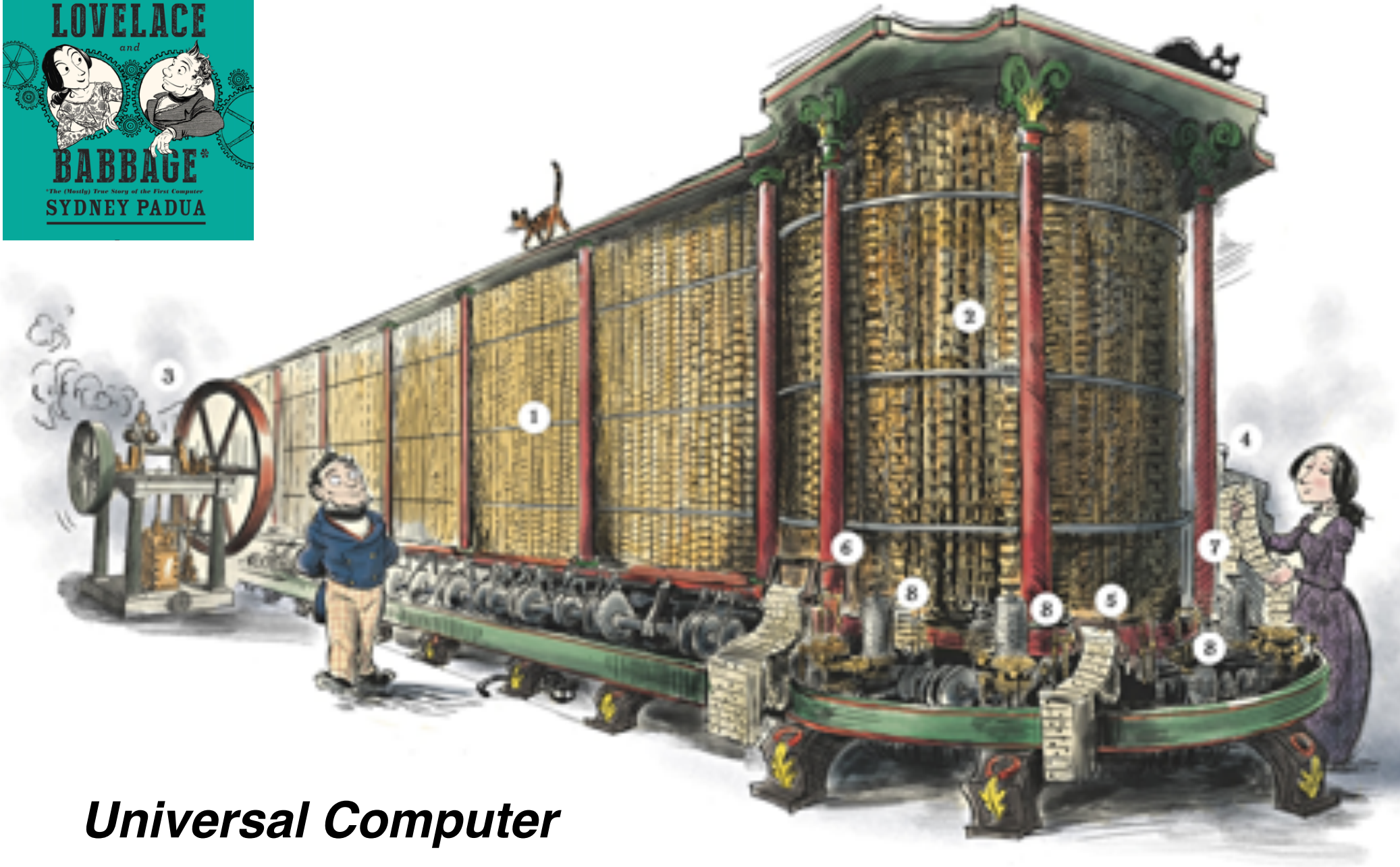
**A NAND gate (for computation)
and a Flip-Flop (for storage)
are all you need to build a computer!**

Babbage Difference Engine





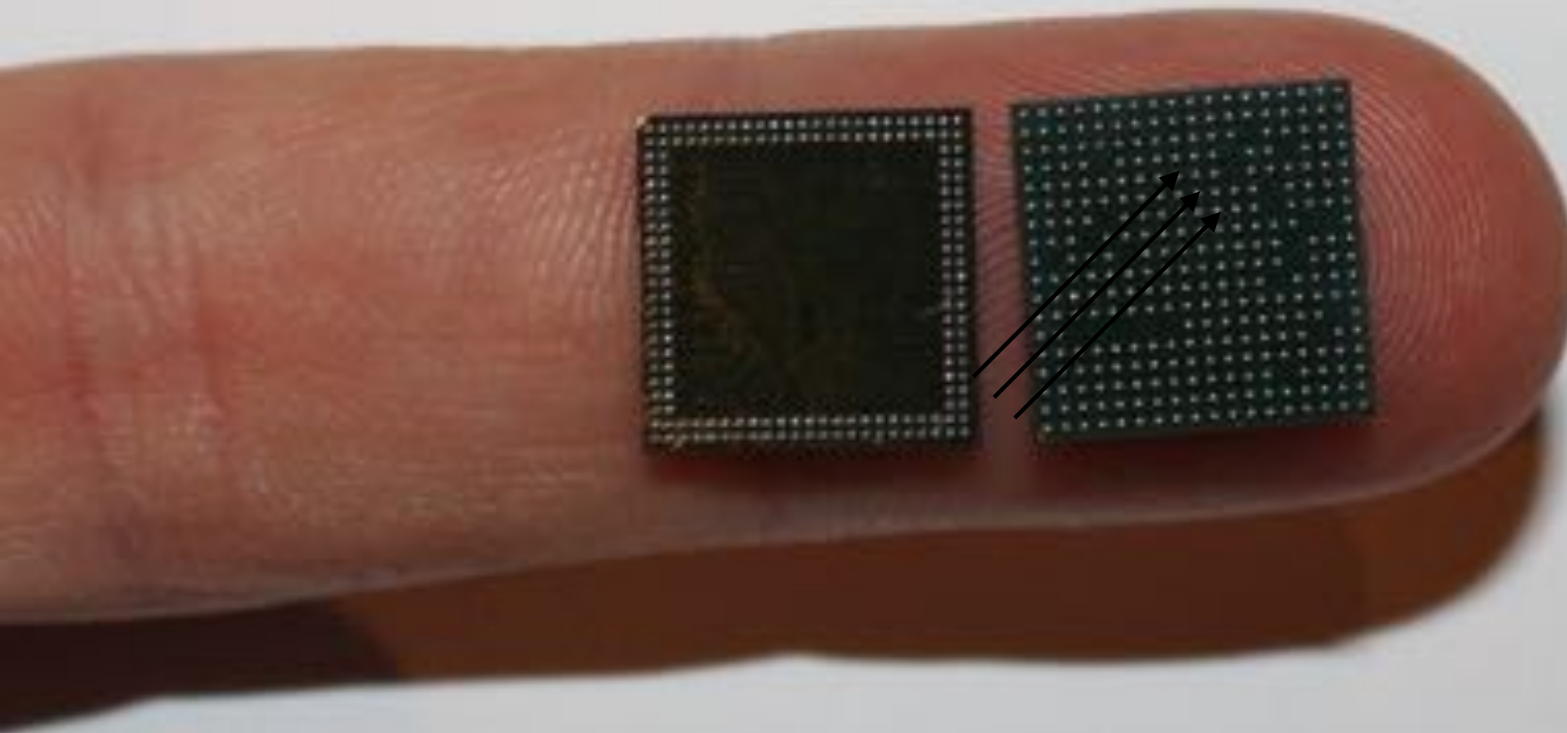
Analytical Engine



Universal Computer

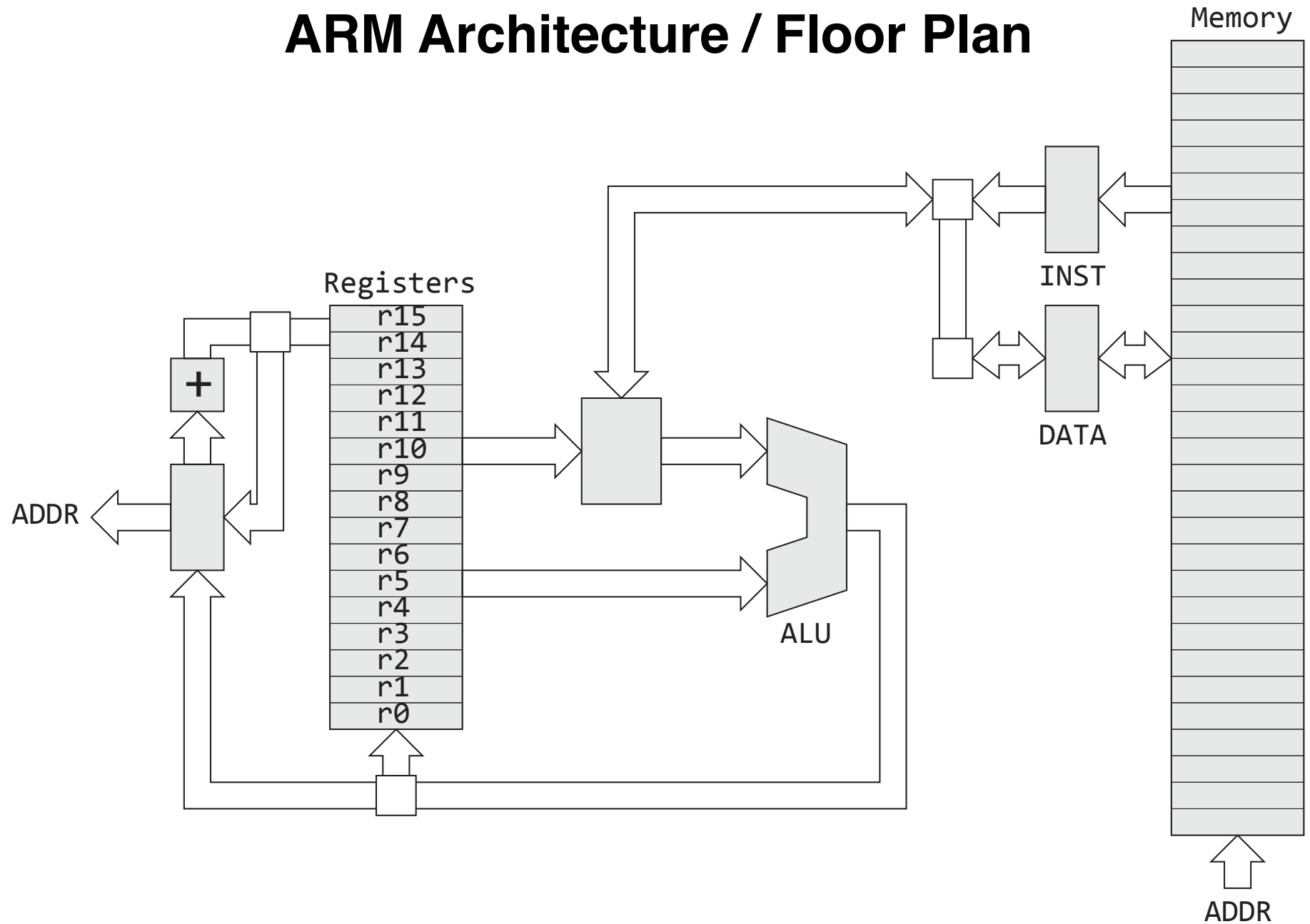
Package on Package

Broadcom 2865 ARM Processor

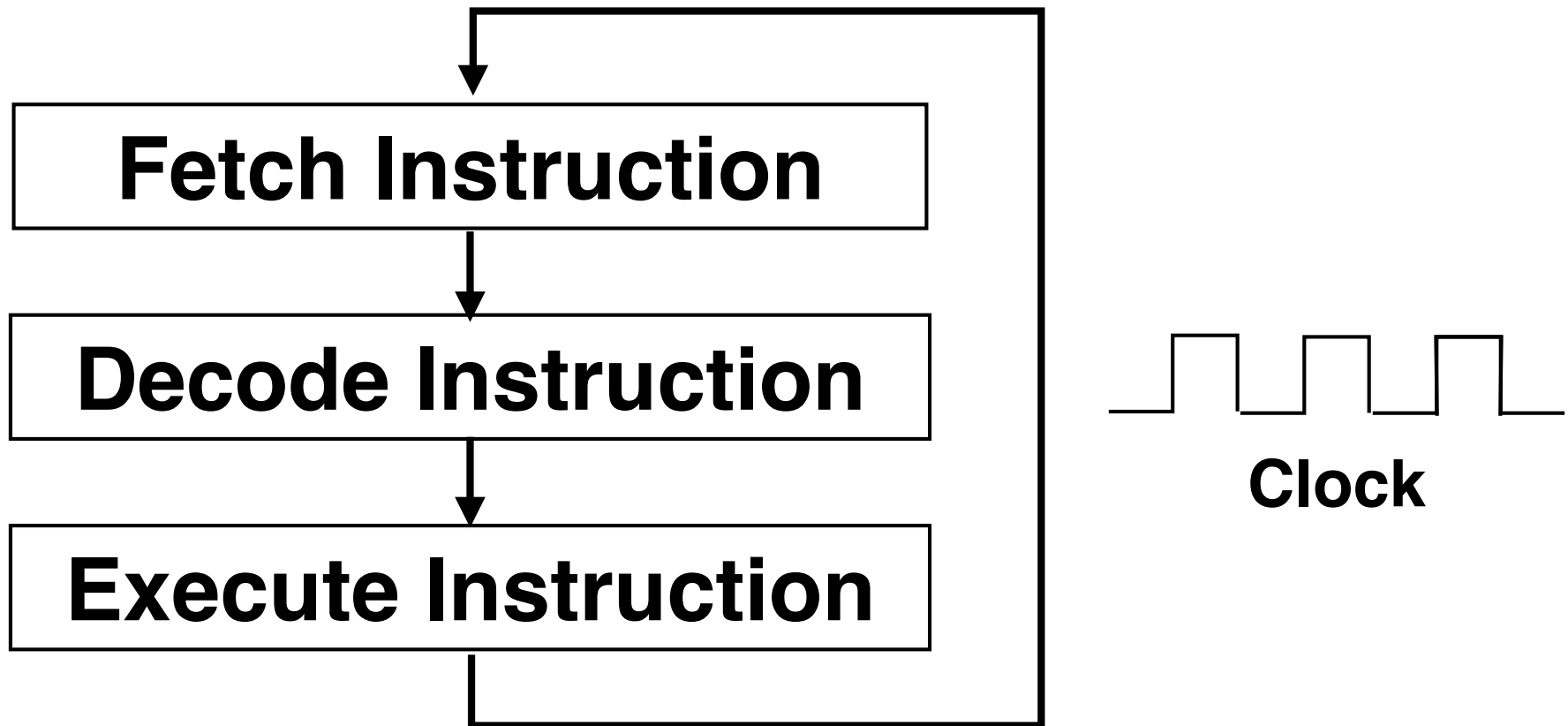


Samsung 4Gb (gigabit) SDRAM

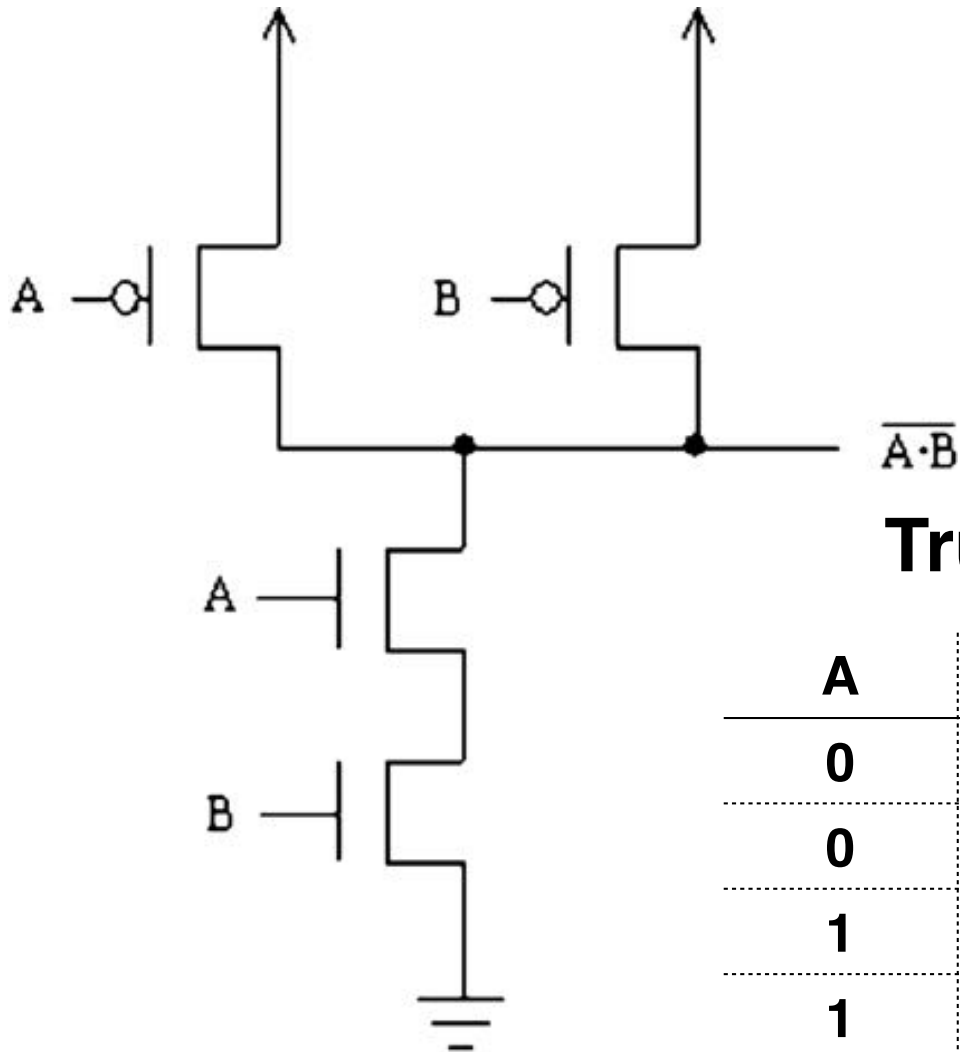
ARM Architecture / Floor Plan



Running a Program



CMOS NAND Gate



Truth Table

A	B	$\sim(A \& B)$
0	0	1
0	1	1
1	0	1
1	1	0

AND, OR, NOT can be built from NAND