6.S188

Build a Digital Clock from the Eighties

Lecture 1A

Overview of Class

- **6.S188 Build a Digital Clock from the Eighties**
- *Level:* U
- *Units:* 1-0-2…PNR credit as far as I understand it
- *Prereqs:* none
- Class Website: **eightiesclock.mit.edu**
- *Instructor:*
  - Adam Hartz (hartz@mit.edu)
  - Joe Steinmeyer (jodalyst@mit.edu)
- *Schedule:* January 5 – January 30
  - Lecture: Tuesdays & Thursdays, 11:00-12:30, room 34-301
  - Open Lab hours: Lab open every weekday 9am-5pm
  - Office hours: See calendar on front page of class, room 38-630

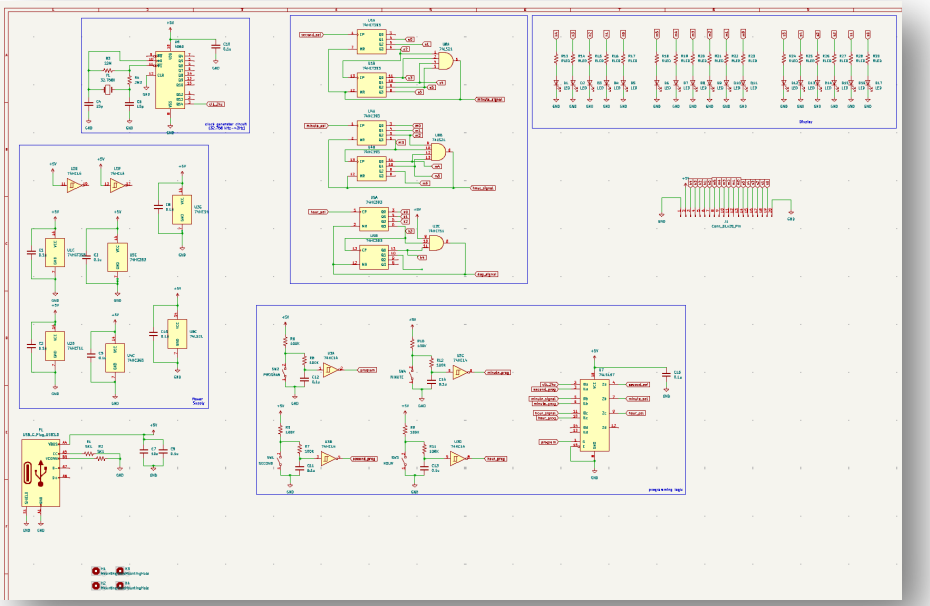
Schedule/ “Syllabus”

- Two lectures every week and a lab or two where you build something.
- Outline:
  - Week 1:
    - Lecture 1A: Origins of Digital Logic
    - Lab 1A: Relay Logic
    - Lecture 1B: Logic Theory and Design
    - Lab 1B: Diode, Transistor Logic, ICs
  - Week 2:
    - Lecture 2A: Combinational Logic
    - Lab 2A: Adder
    - Lecture 2B: Stateful Logic
    - Lab 2B: Counter
  - Week 3:
    - Lecture 3A: Clocks
    - Lab 3A: Digital Clocks (oscillators) and few other things
    - Lab 3B: Start Building Actual Digital Clock From the Eighties
  - Week 4: Continue/Finish Building Clock From the Eighties
    - Build Digital Clock From the Eighties
    - Build Digital Clock From the Eighties
- No Psets, just do the labs, build the stuff
- No Exams

This is a fresh class.
Never run before.
It is going to be a mess.

“Digital Clock From Eighties”?

- Make a Clock that Displays the Time in Binary

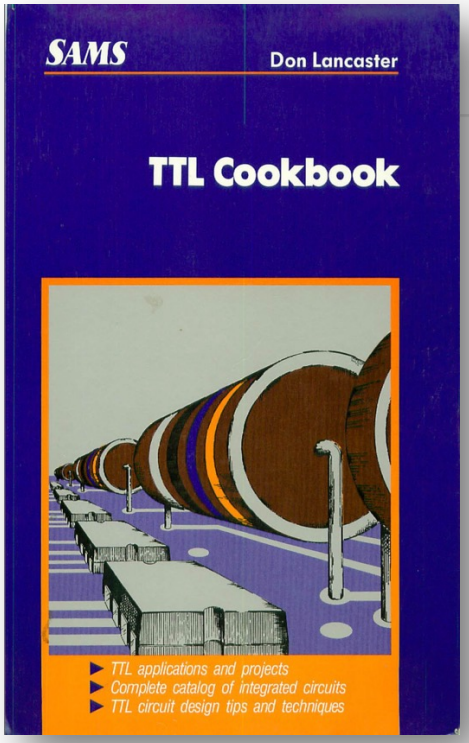


We’ll build up to this

You wanna make it more complicated, be our guest, but we’ll have PCB blanks of one or two version if you want to build that way

EECS Evolves Quickly…

- So quickly it is hard to say this is “eighties” or ”seventies” clock…it is kinda wrong…



One of texts we’ll use as reference

Author reflecting on changes in field from 1972 to 1982….

What has changed explosively, though—in case you have been asleep for the last five years—is the microprocessor revolution. Very simply and very bluntly, if it is electronic and doesn't have a microprocessor in it, it is not worth doing. And that applies to ALL electronics, radio, radar, television, communications, power control, hi-fi, video disks and recorders, appliances, cameras, pagers, satellite receivers, data acquisition, process controls. Everything electronic. You name it.

Very few people are dumb enough to still go out and design an electronic something by taking a bushel basket of TTL or other general purpose logic family and then hand-wiring everything together. Today you do everything you can with changeable software, making any remaining hardware very general and flexible.

Don Lancaster
January, 1982

What we’re doing

Why Do This?

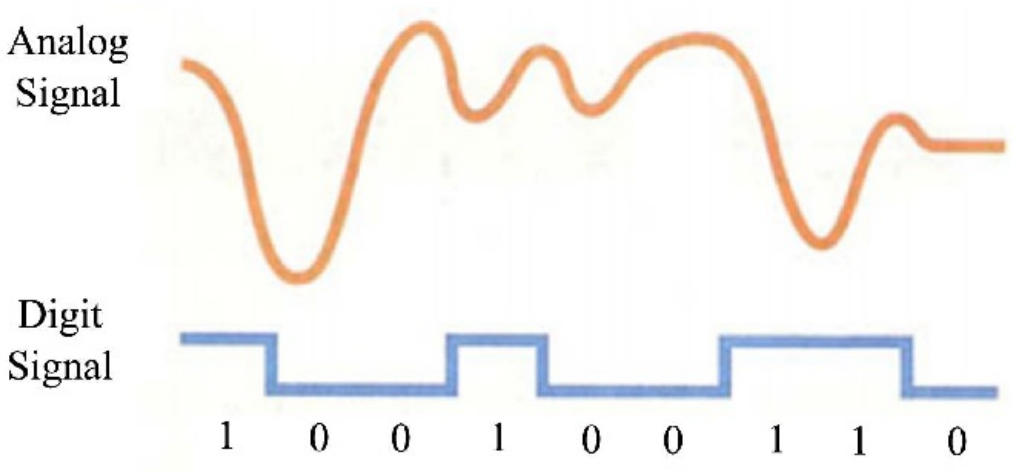
- Iunno we thought it would be fun and a way to "touch some grass" in the field of EECS.

- Designs are so ridiculously complicated at this point that in day-to-day or year-to-year it can be very hard to have a sense of what is actually happening….
  - Send the build up to Vivado or to Cadence or whatever and stuff comes back…maybe it works.

- Why not start at ground up?

Digital Electronics

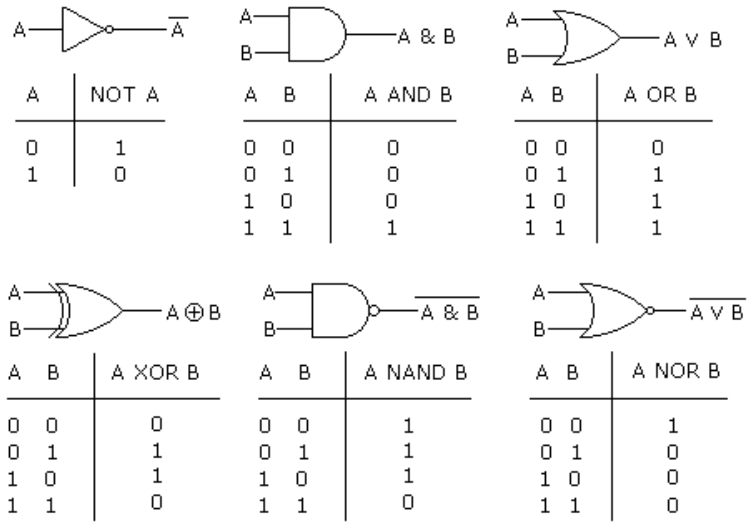
- Thinking of electrical voltages not as continuous signals that can be one of many values (analog)…instead classify all voltages into one of two values.
- It is a choice and it makes designing fundamental units easier



https://www.researchgate.net/figure/Analog-and-digital-signals_fig1_361511253

Simpler worldview…

- Leads to simpler set of foundational circuits.
- These are usually very easy to build and make work reliably.
- This allows us to scale their production massively and make up for (and surpass) the initial limitations of classifying things as 0's and 1's.



A	NOT A
0	1
1	0

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

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

A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

https://cs.lmu.edu/~ray/notes/digitallogic/

Scalability

- Modern digital chips have the equivalent of *billions* of logic gates on them at this point.
- This scale and complexity was only achievable because the designs started with very simple, robust circuits.
- And goodness have we scaled and continued to scale

Moore's Law…

- Original argument/prediction of how we'd scale computation…
- It has been correct so far since we kinda keep redefining what it means so that it keeps getting satisfied.
- It is a little bit of a meme/lie at this point…it represents more of an idea and vague definition of continued progress

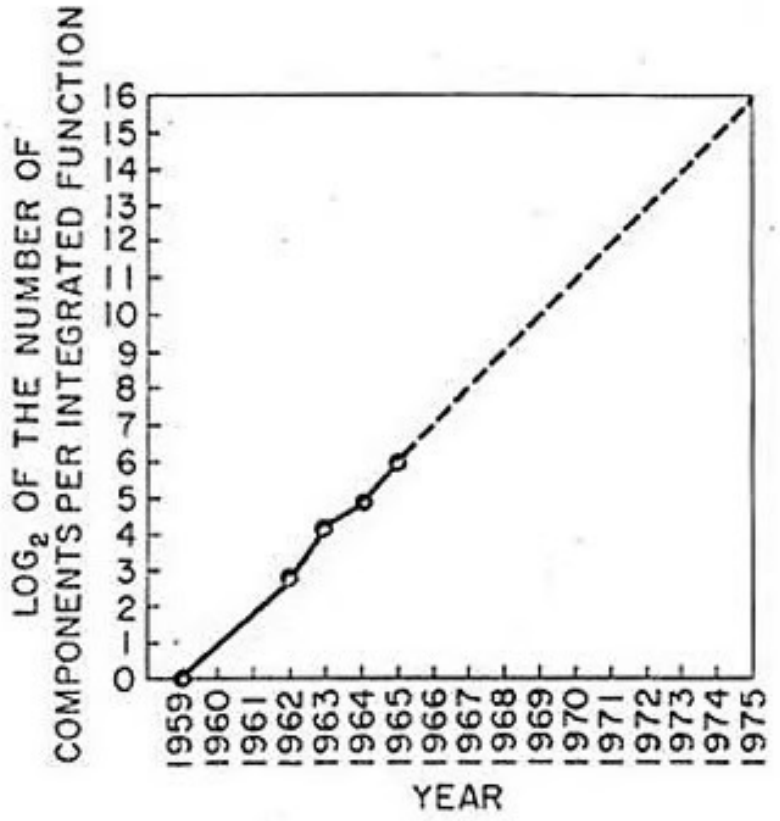
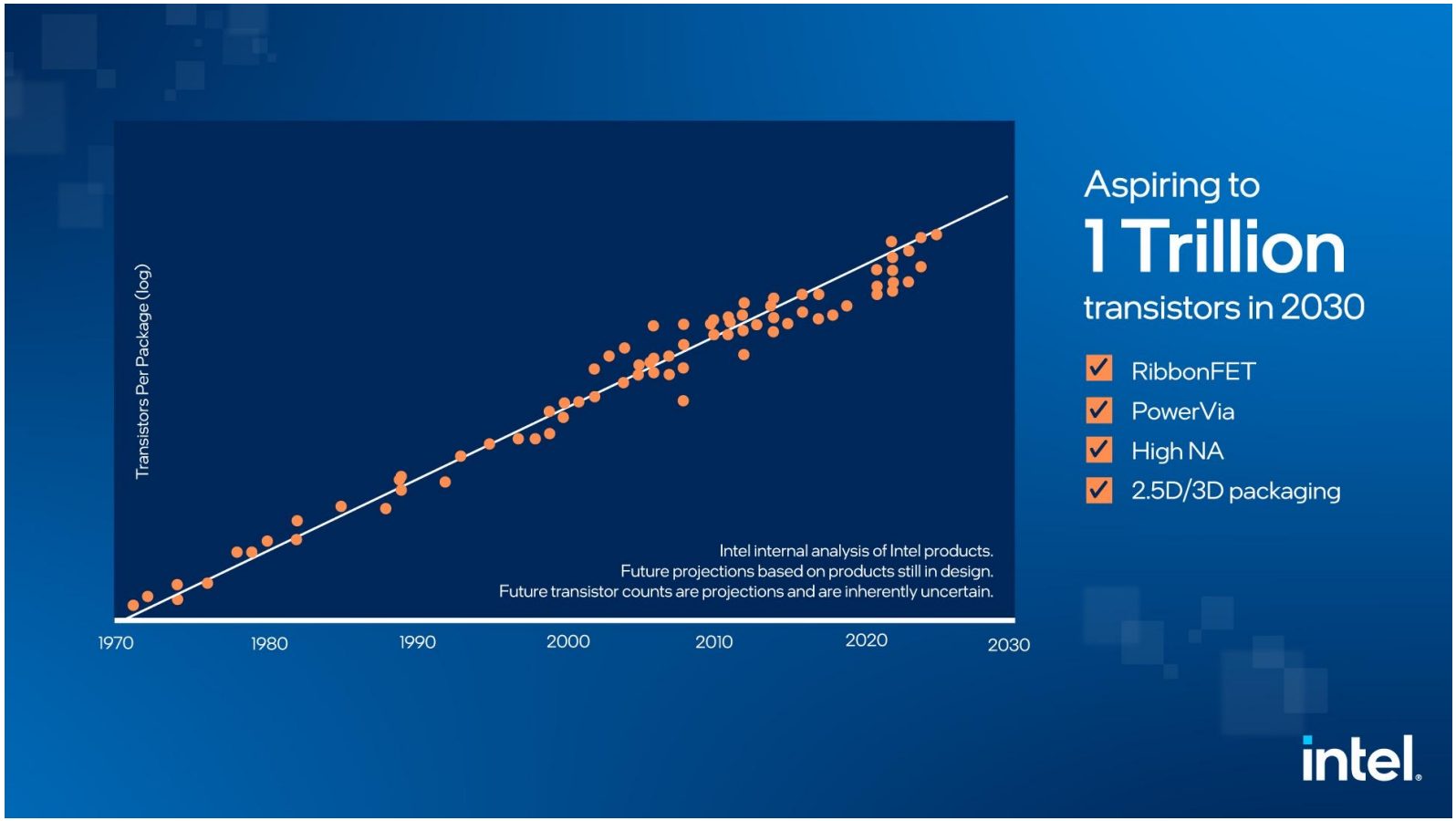


Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

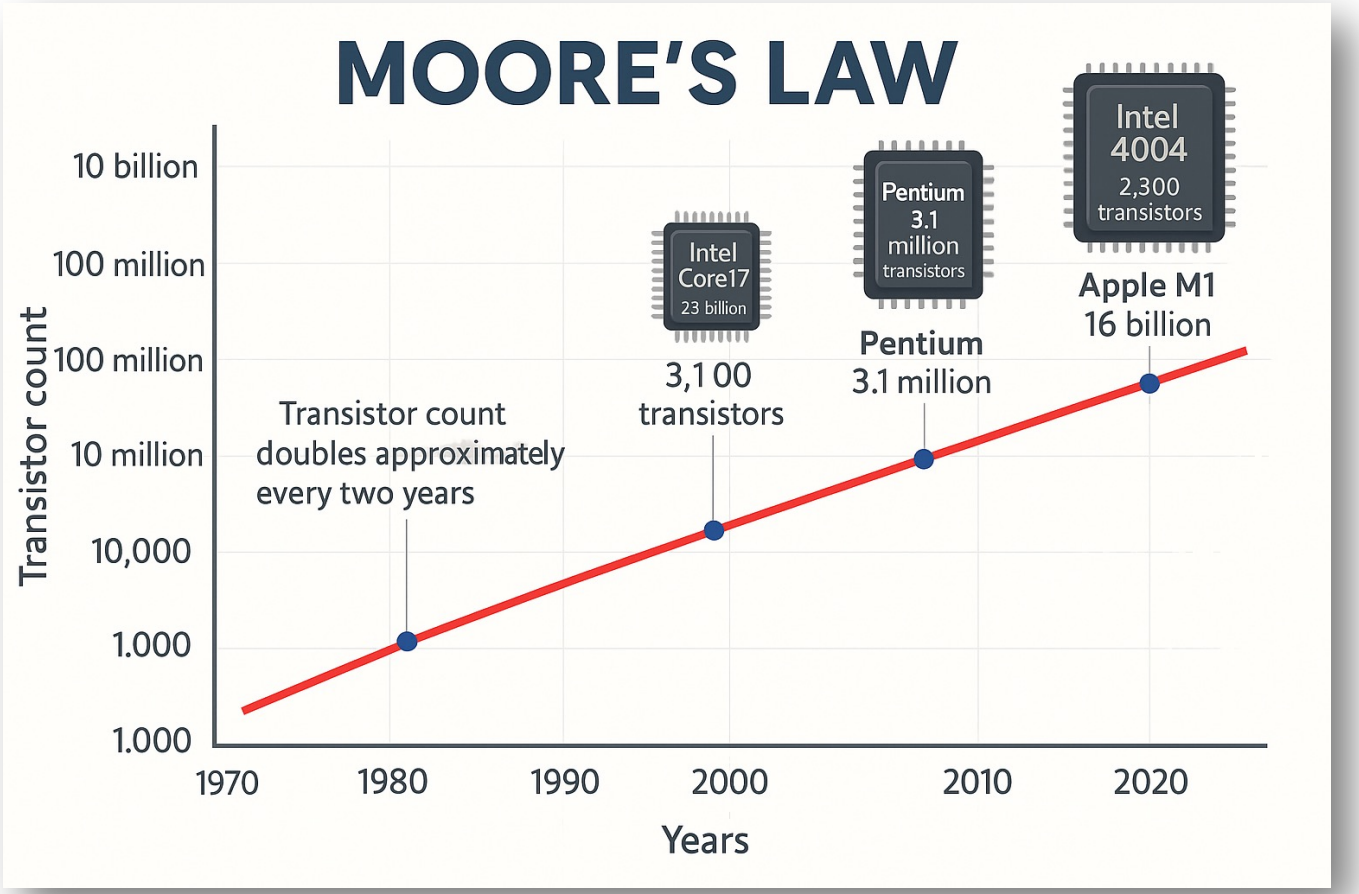
https://ourworldindata.org/moores-law

Intel's Attempt at Explaining Moore's Law



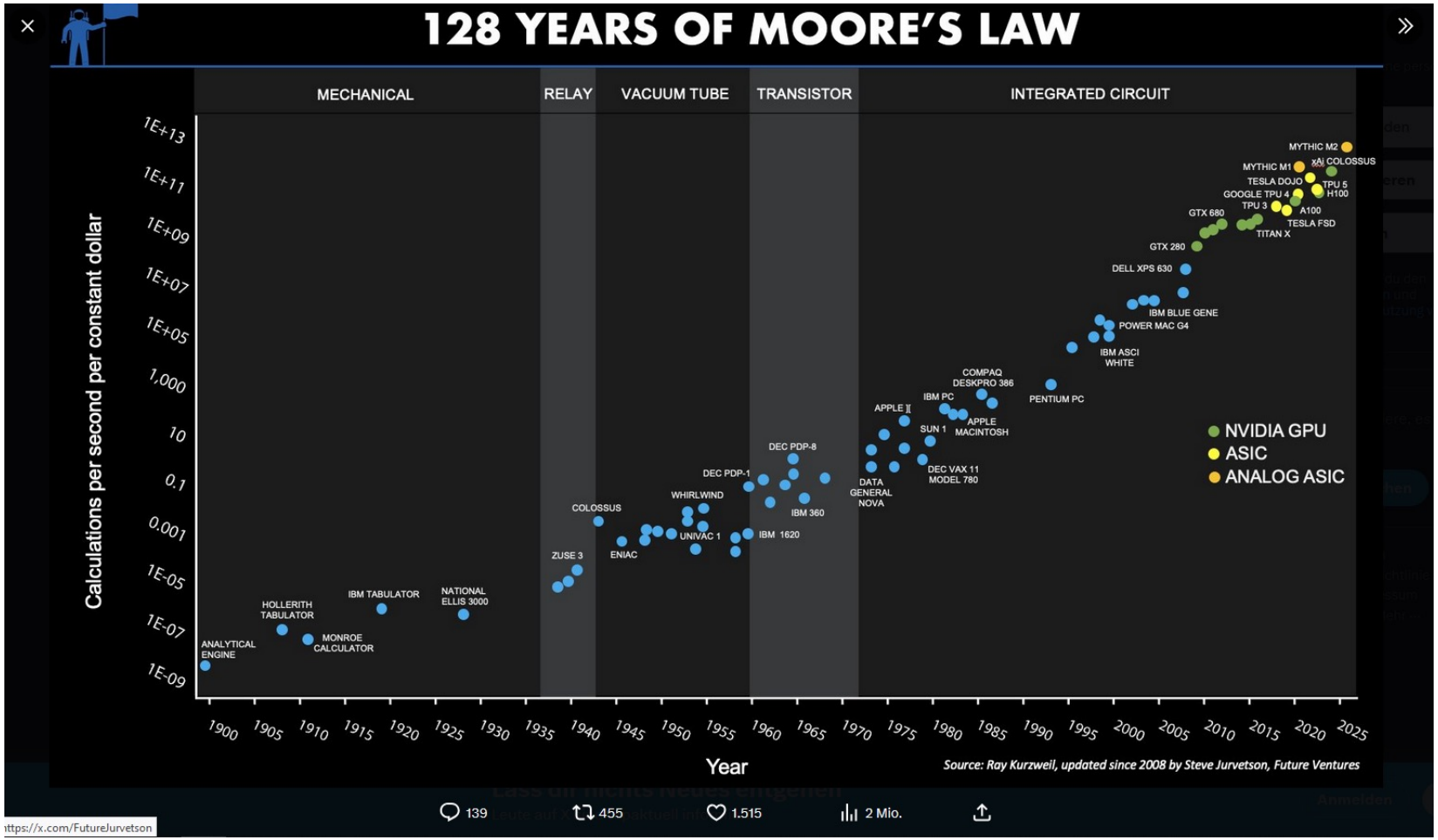
https://newsroom.intel.com/press-kit/moores-law

The AI is not self-aware



https://semiwiki.com/wikis/industry-wikis/moores-law-wiki/

Actually Pretty Good Plot



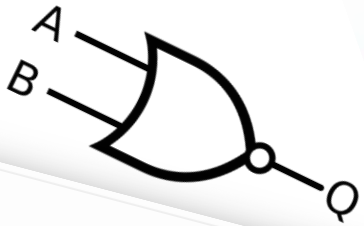
https://www.reddit.com/r/singularity/comments/1h55mxm/moores_law_update/

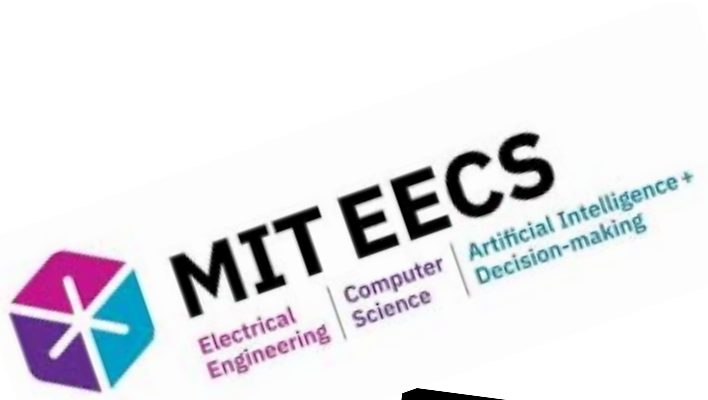
State of the Art is Wild

- I think these are going for $40K-ish
- Each of the 72 Blackwell GPUs has 208 billion transistors on it.
- Digital Electronics are Unfathomable

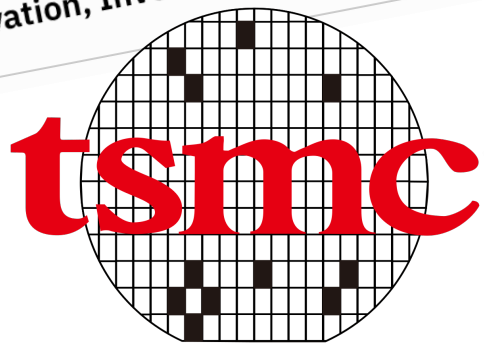
https://futuretimeline.net/blog/2024/03/22-nvidia-ai-chip-208-billion-transistors.htm

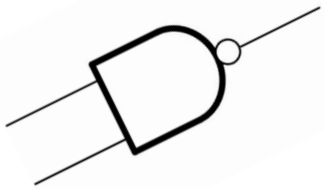
Zeitgeist





6.S187 Deploying Generative AI: Healthcare, Business Analytics, Education, Innovation, Investment §

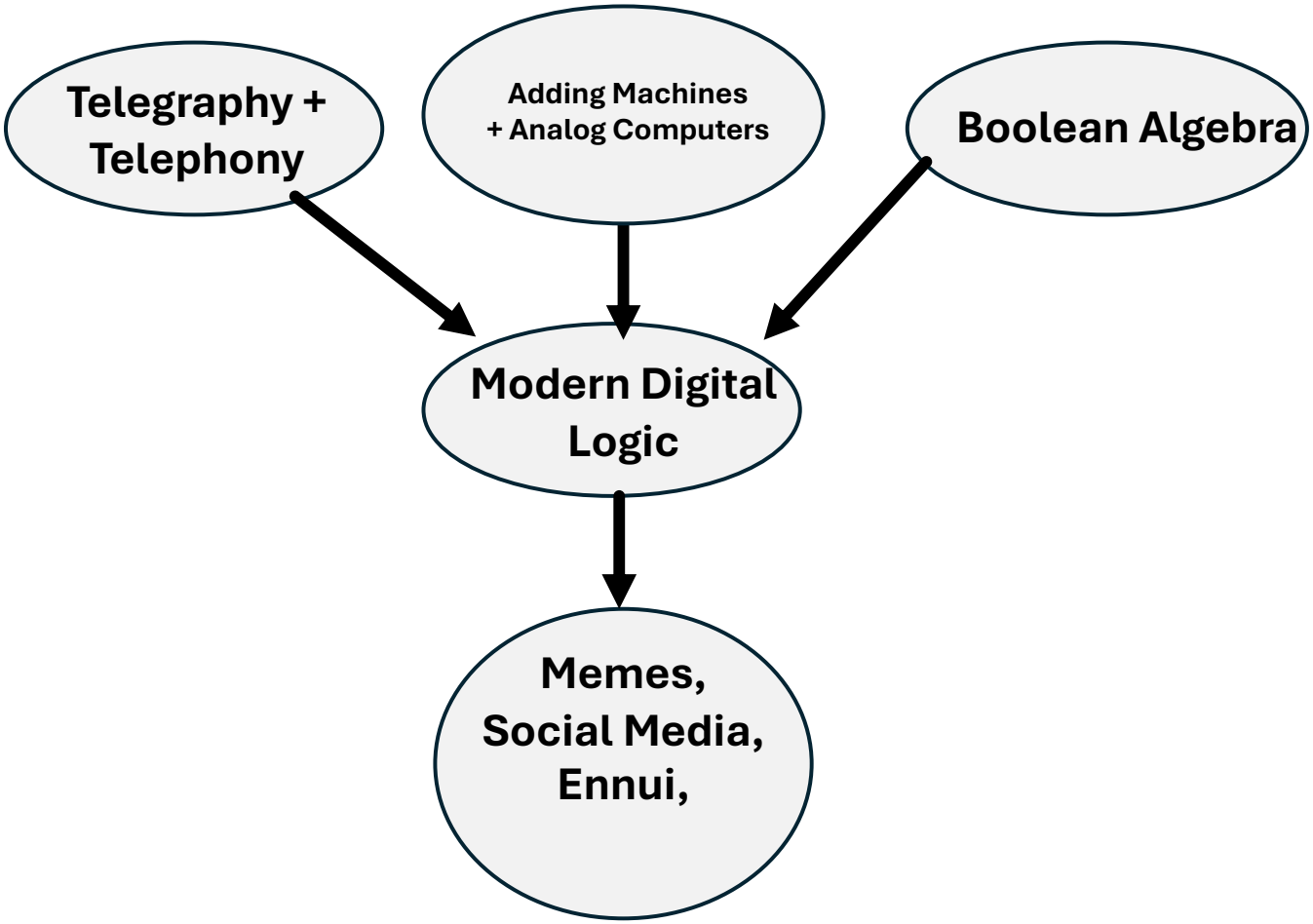


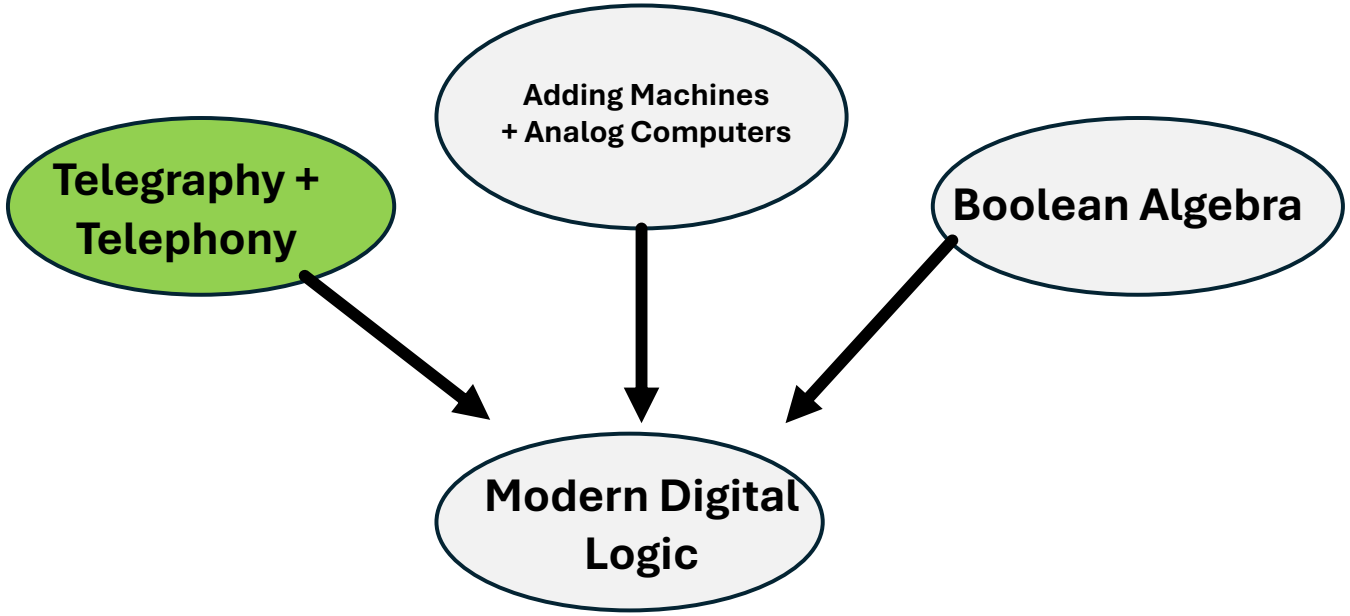


But I guess…where did this come from?

- Great…this is what we're doing now in 2025.
- Where did it come from?
- Why did we go this route?

Stuff Leads to Other Stuff



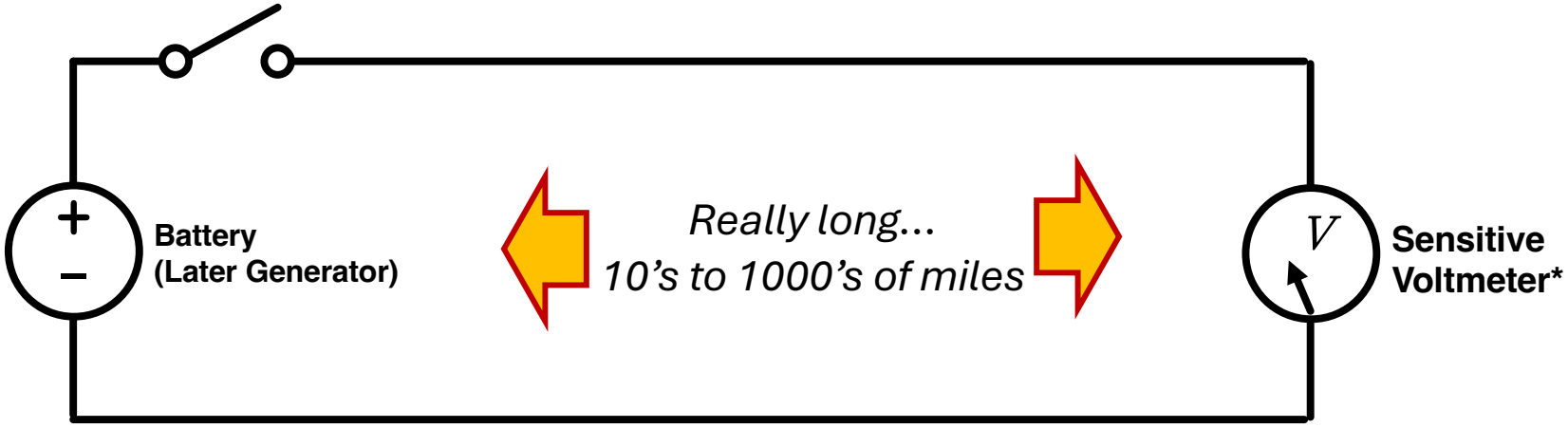


Telegraphy

Digital Communication Before they Knew what Digital Communication Was

Telegraphy/Morse Code (1830s)

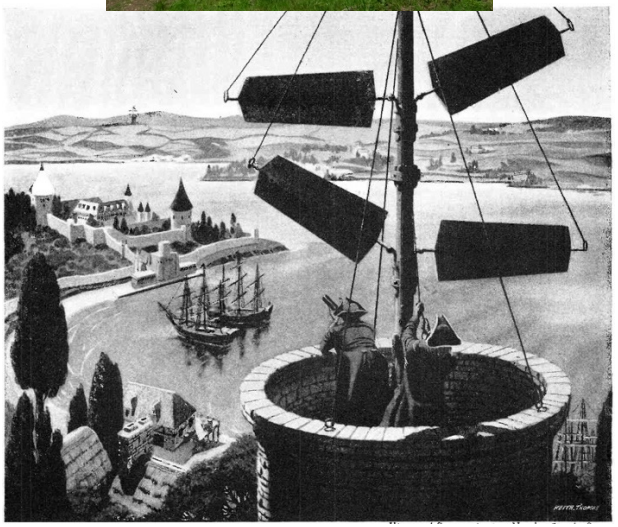
- Samuel Morse (also a ok-ish painter)
- Encode information using ON-OFF values of differing lengths
- Basic digital communication

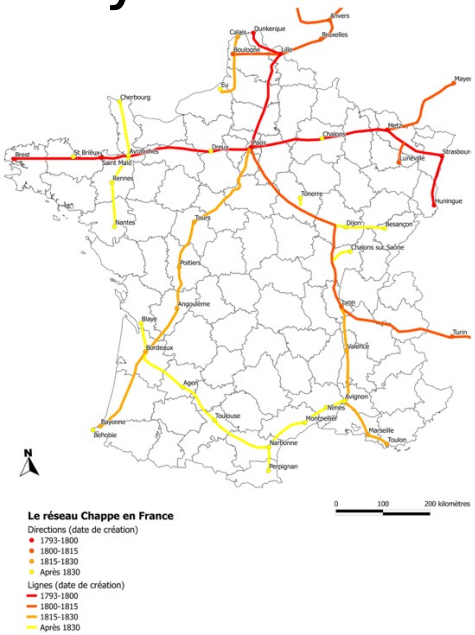


*actually what they used is closer to a current meter

Mechanical Telegraph (Semaphores)

- The original telegraph was made of chains of towers operated by people that would relay signals

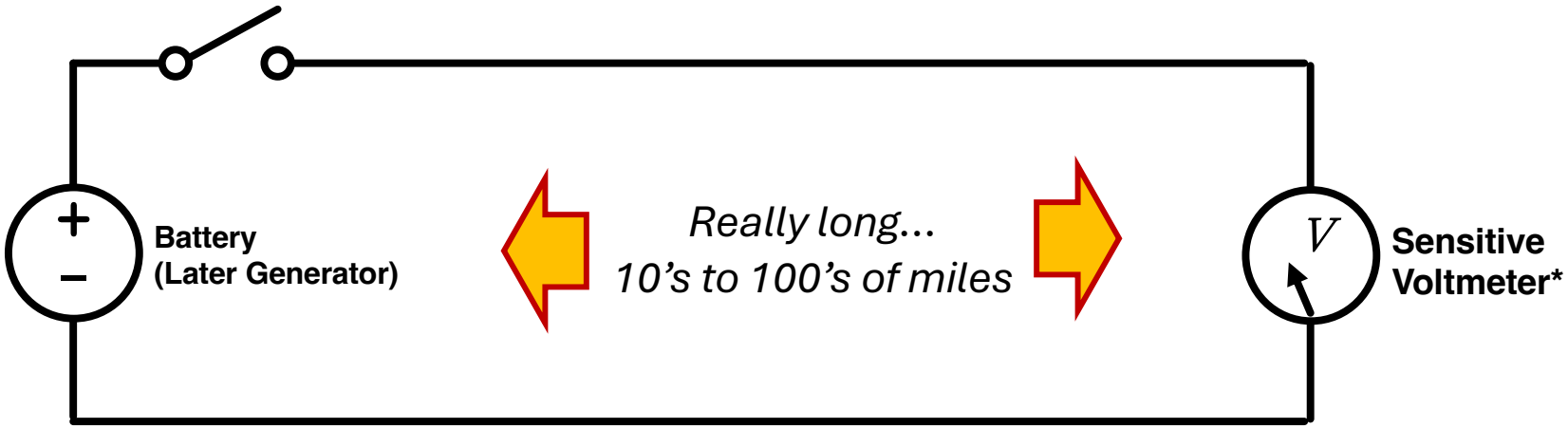




https://en.wikipedia.org/wiki/Optical_telegraph#/media/File:Signaling_by_Napoleonic_semaphore_line.jpg

Electric Telegraphy (1830s)

- Samuel Morse (also a decent painter) and many others developed electric telegraphy
- Encode information using voltage on/off pattern
- Basic digital communication



*actually what they used is closer to a current meter

How to Actually encode information?

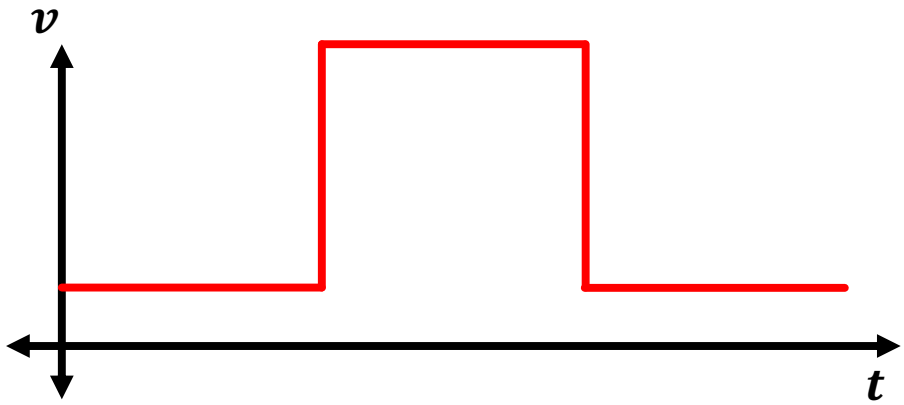
- Same problem we deal with today when we send information.

- Rarely is signal encoded as:
  - Signal = “1”
  - No signal = “0”

- You run into lots of synchronization issues

Synchronization Issues…

- Consider this signal…

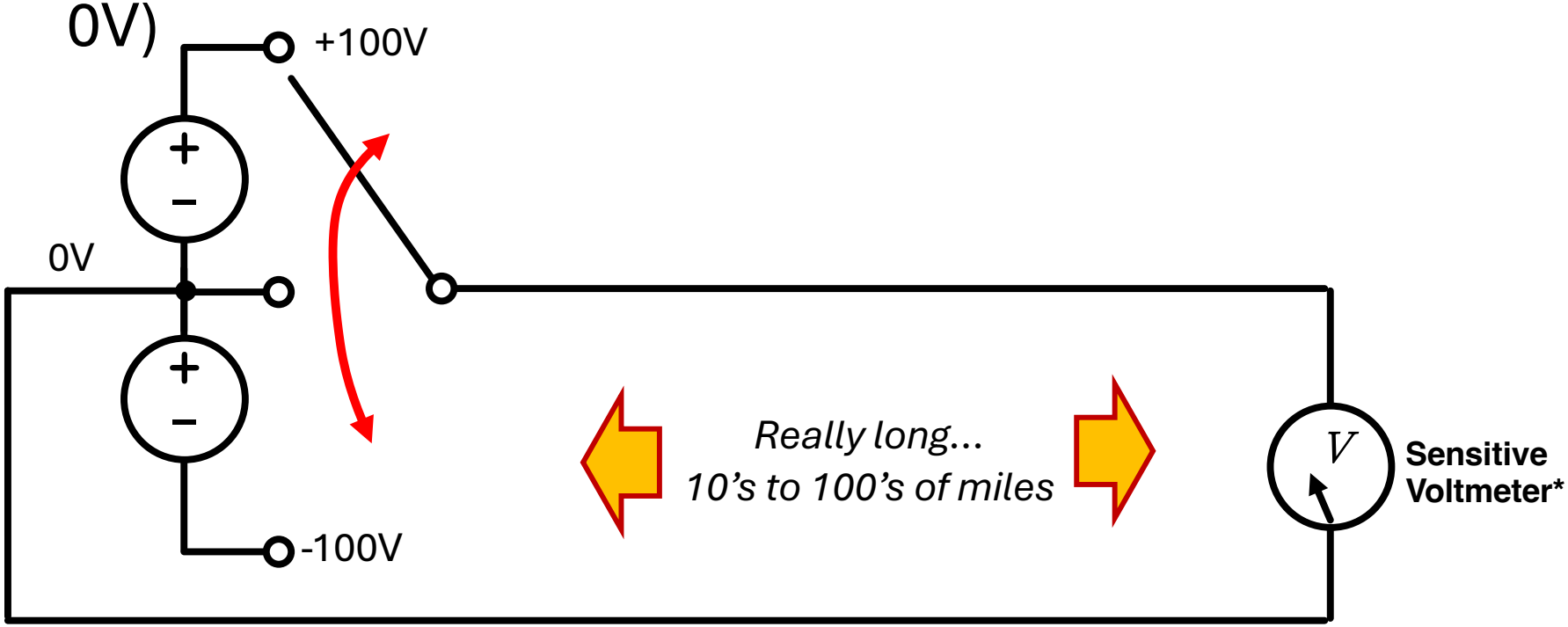


- Is this a 010? Is this a 001100? Is it a 000111000?
- Or is it no signal…followed by a 1 followed by a 0?
- Very confusing…

- In transmitting binary data…you usually need to be able to express ***three*** things…
  - 1
  - 0
  - no signal

Early Variants of Telegraphy

- A lot of European systems actually encoded three levels (large positive, large negative, and 0V)



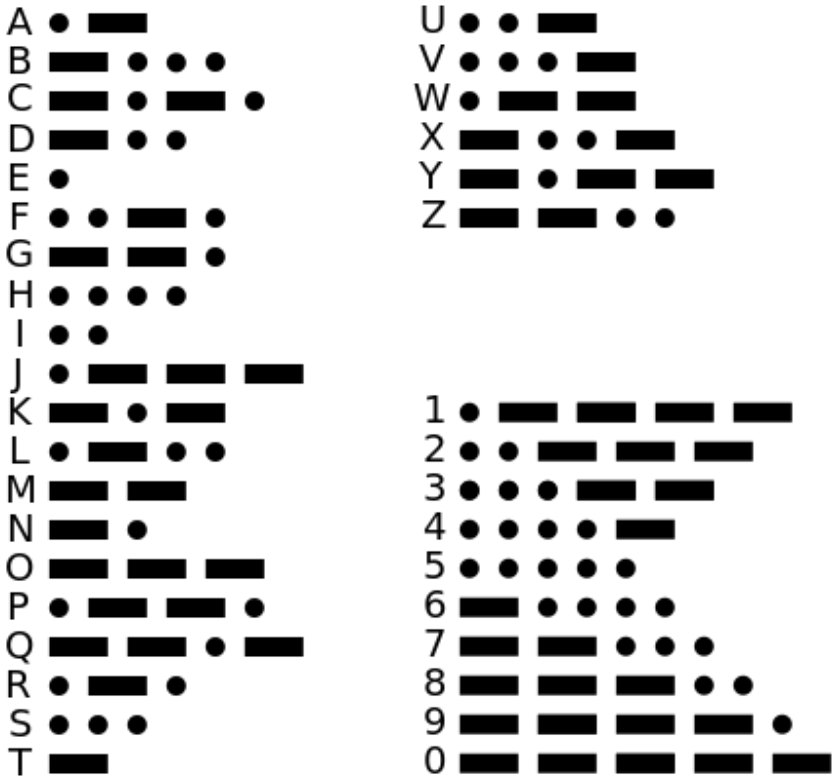
- This actually required significantly more complicated detection equipment than just on/off signals

Morse Code

- What Samuel Morse (and Joseph ***Henry***) and Alfred Vail did was develop an encoding scheme that just involved on-off values
- They then used duration as an additional axis to get more expression
  - Short on pulse is a “1”
  - Long on pulse is a “0”
  - No signal is a no signal

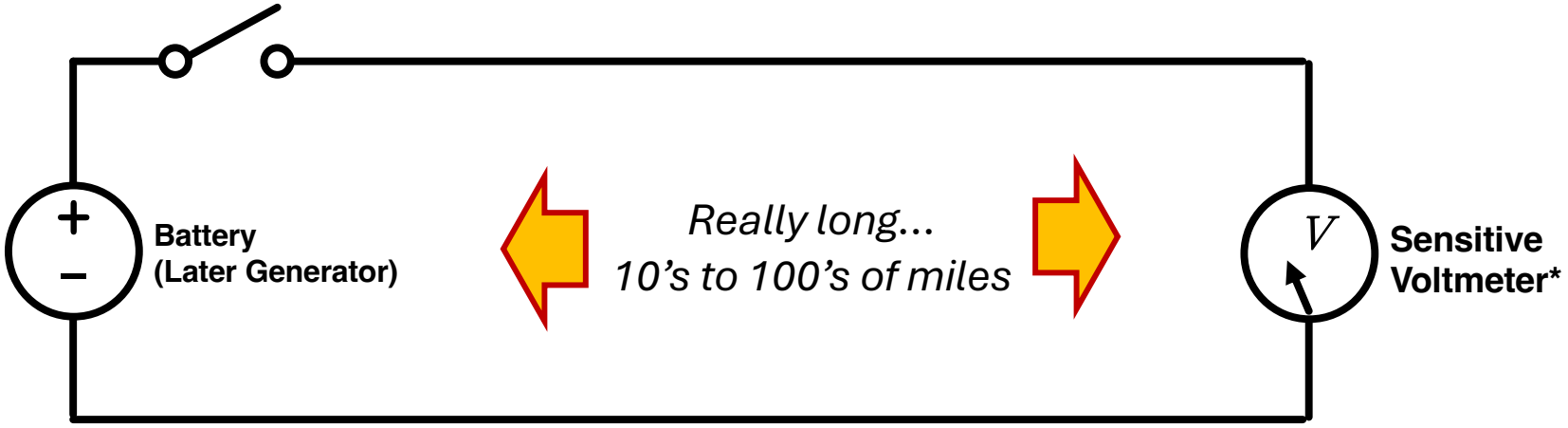
Morse Alphabet (one version anyways)

- This allowed significantly simpler equipment and infrastructure and was actually easier to record, interpret
- It was also easy to sync data rates since silence meant “no signal”
- Proved easy to port to wireless 50 years later



Why digital? Why not analog?

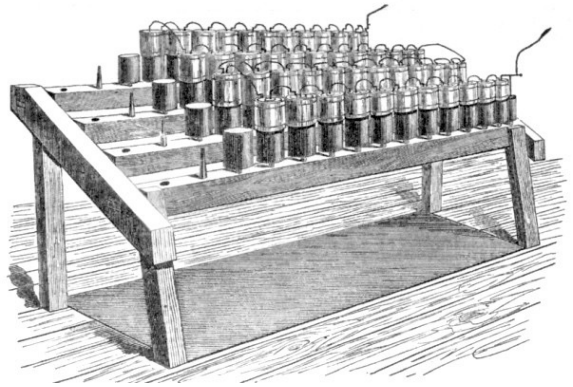
- Well, first we had no meaningful signals in analog format anyways.
- Also losses were so great (and poorly understood) that having anything other than the harsh 1/0 interpretation of signal proved problematic



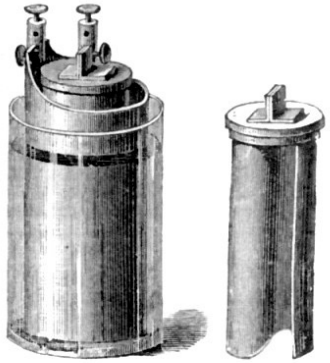
*actually what they used is closer to a current meter

Powering Telegraphy

- How did they do it?
- Generators weren't a thing until 1870s so used giant cell stacks in parallel/series combos (electrical batteries)



Grove Battery

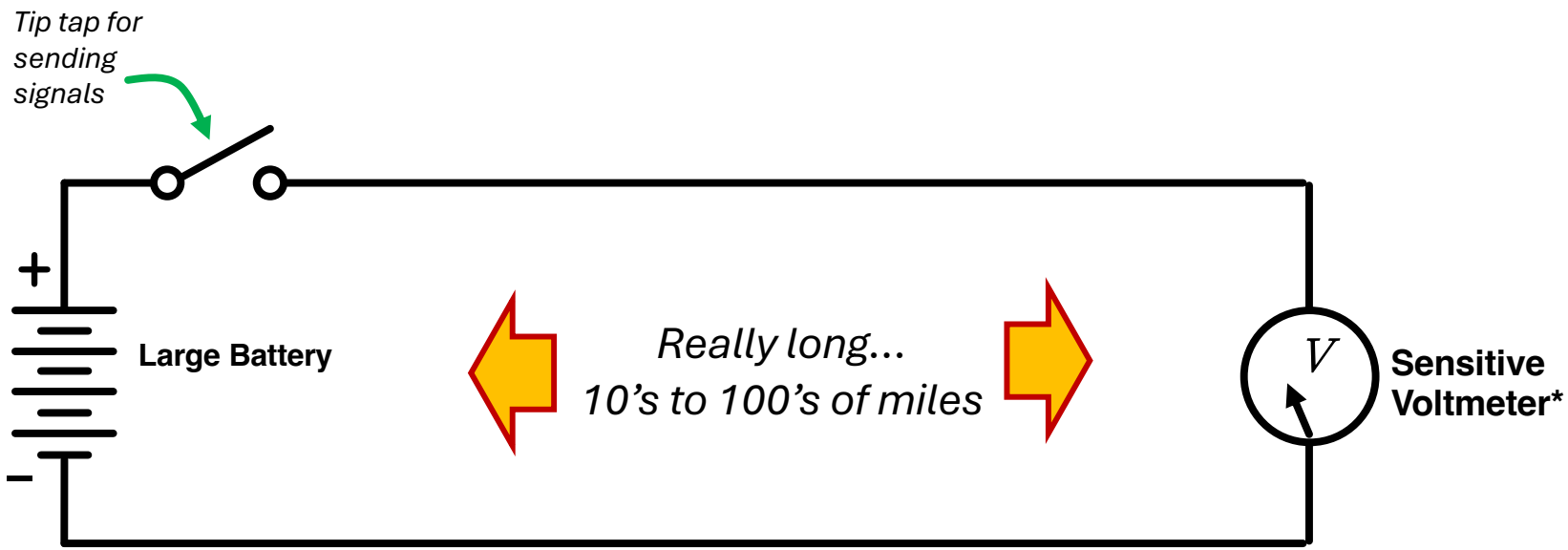


Grove Cell

Zinc, Platinum/Carbon, Sulfuric Acid battery

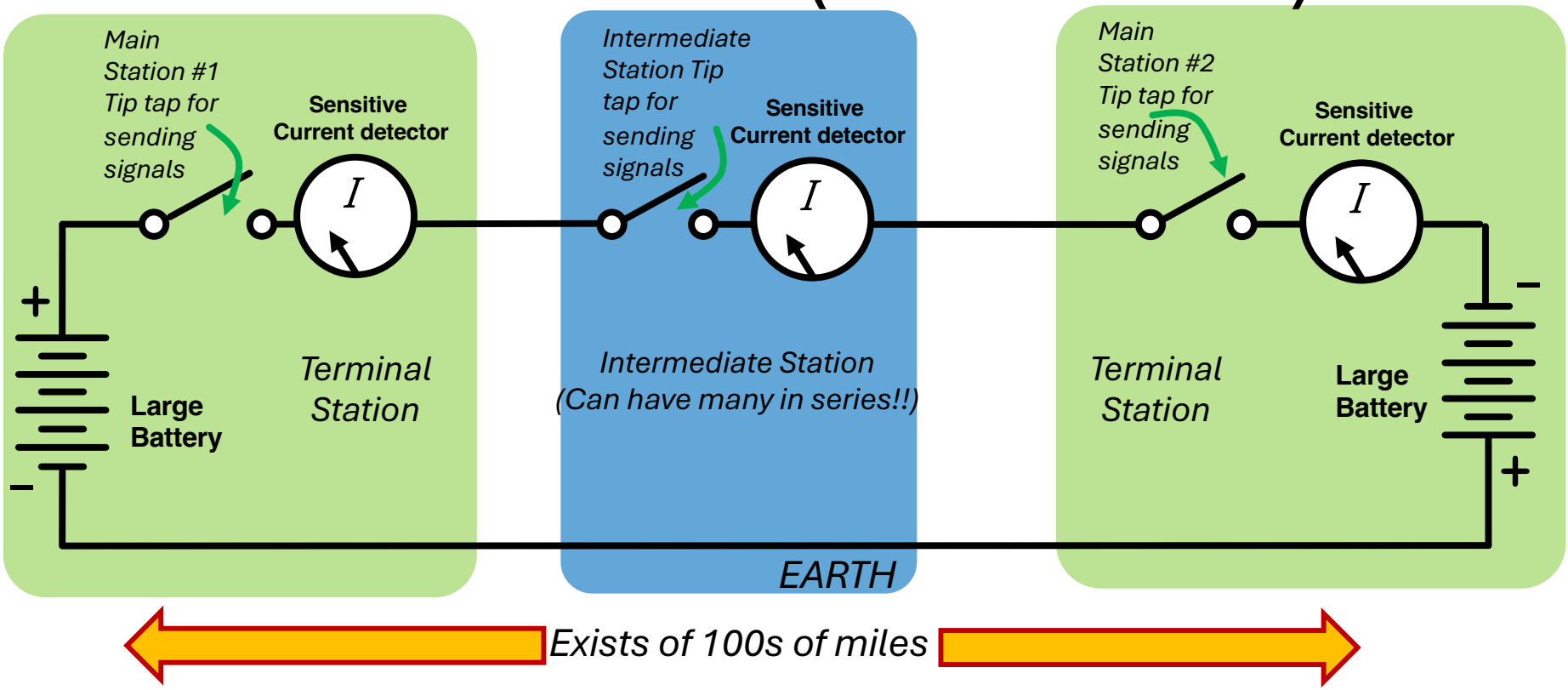
- Sources report you'd need about 2V per 10-to-20 miles of telegraph would work

How Would it Work?



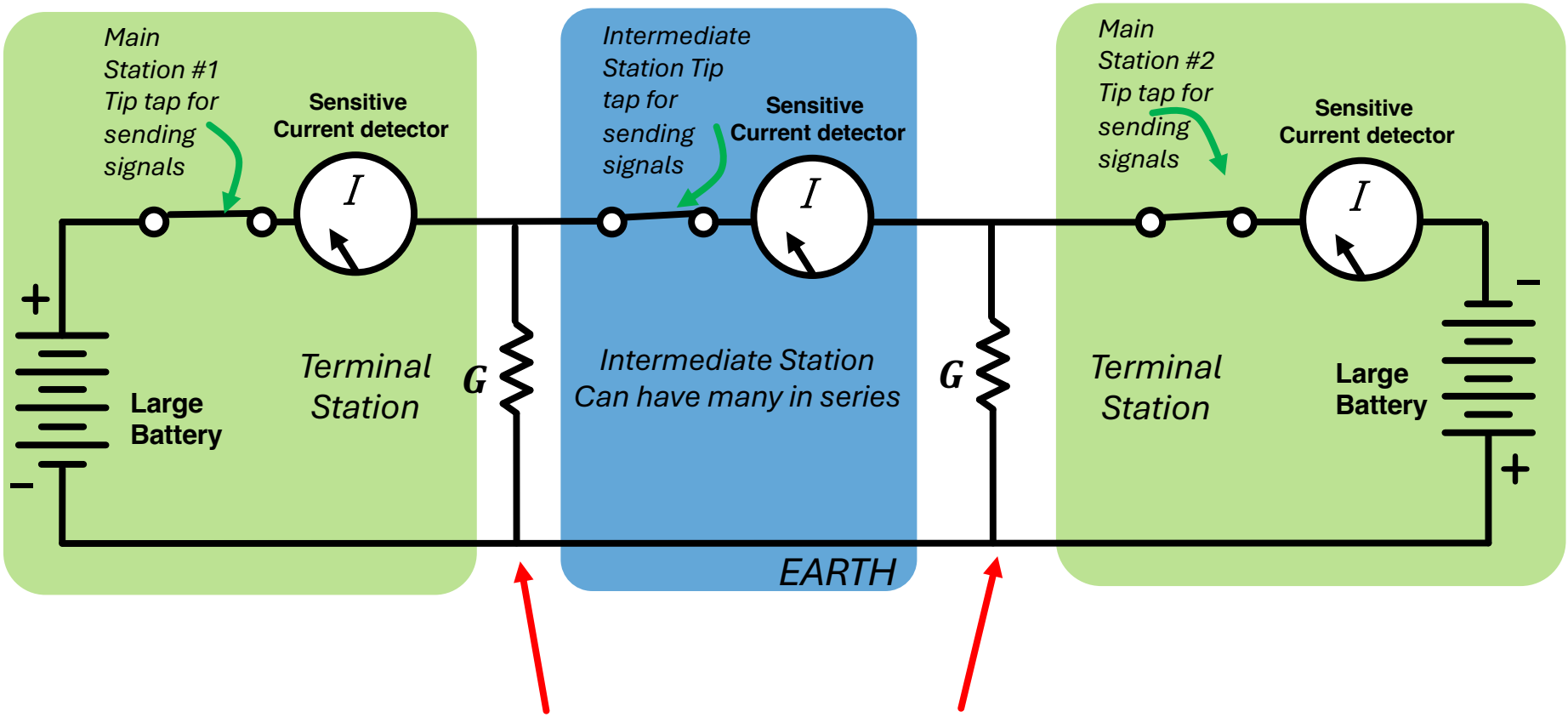
- More Complicated than this…and a bit different…

How Would it Work (More Detail)



- Note opposite battery orientations…why?
- In reality the “return path” was usually Earth…why?
- How Did they make sensitive Current Meters?

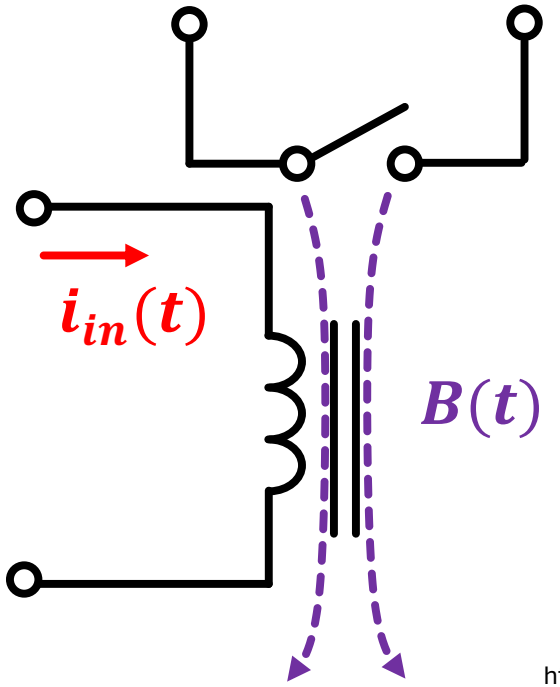
Leakage! Too



Parasitic Leakage all along the line

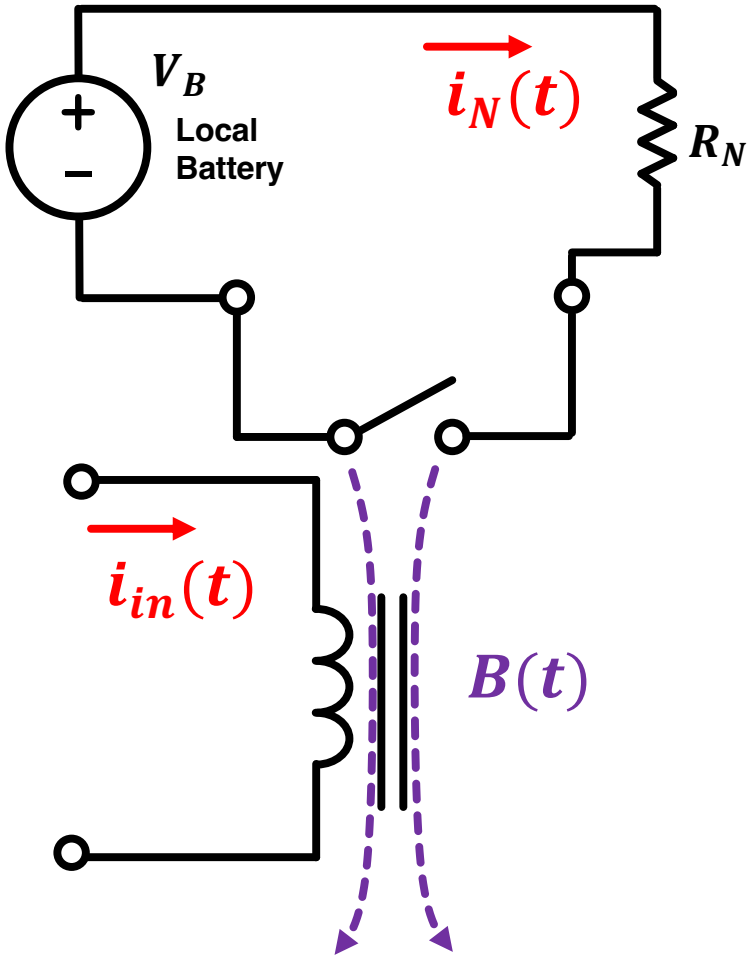
Sense Current Using a Relay…what is a relay?

- An input set of terminals connected to a coil
- Coil can induce magnetic field
- Magnetic Field can influence position of nearby switch
- Incorporate that Switch Into a Local Circuit

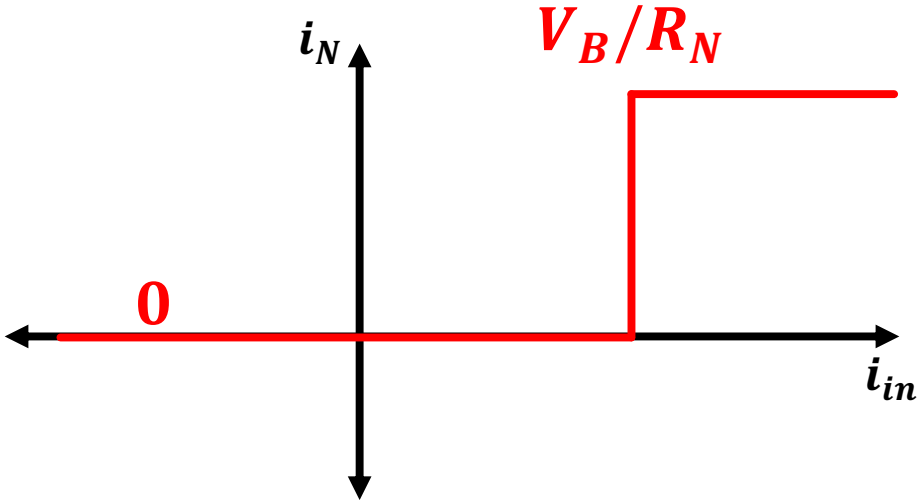


https://www.digikey.com/en/products/detail/omron-electronics-inc-emc-div/G2R-1-E-AC120/36

Incorporate the Relay into a Circuit

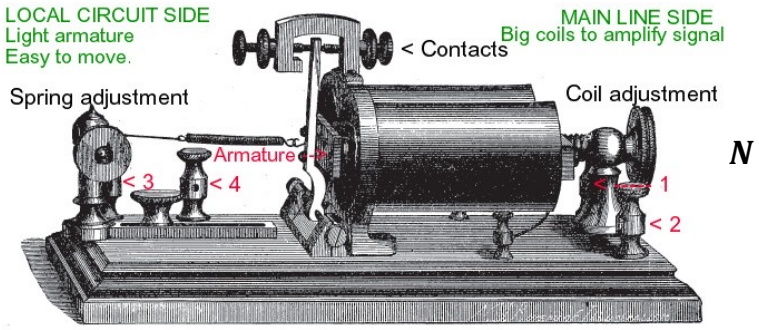


- How will this circuit operate?
- What will its In/Out relationship look like?



- No current in secondary circuit will flow until current in coil passes some threshold

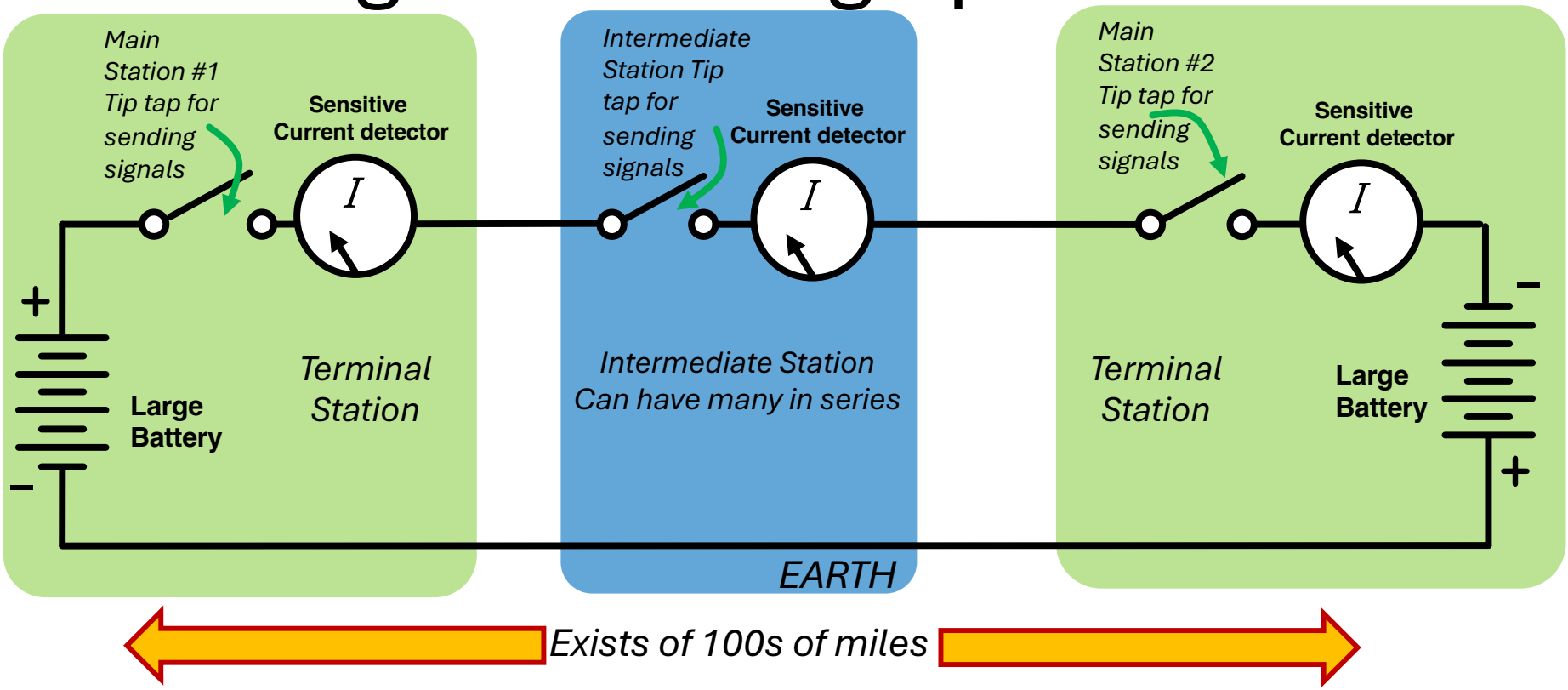
This Input/Output Relationship can be engineered



- A ton of effort went into designing these relays to be very sensitive
- More windings of coil (increase induced field)
- Design a really nice switch
- Make sure it doesn't bounce…
- List goes on

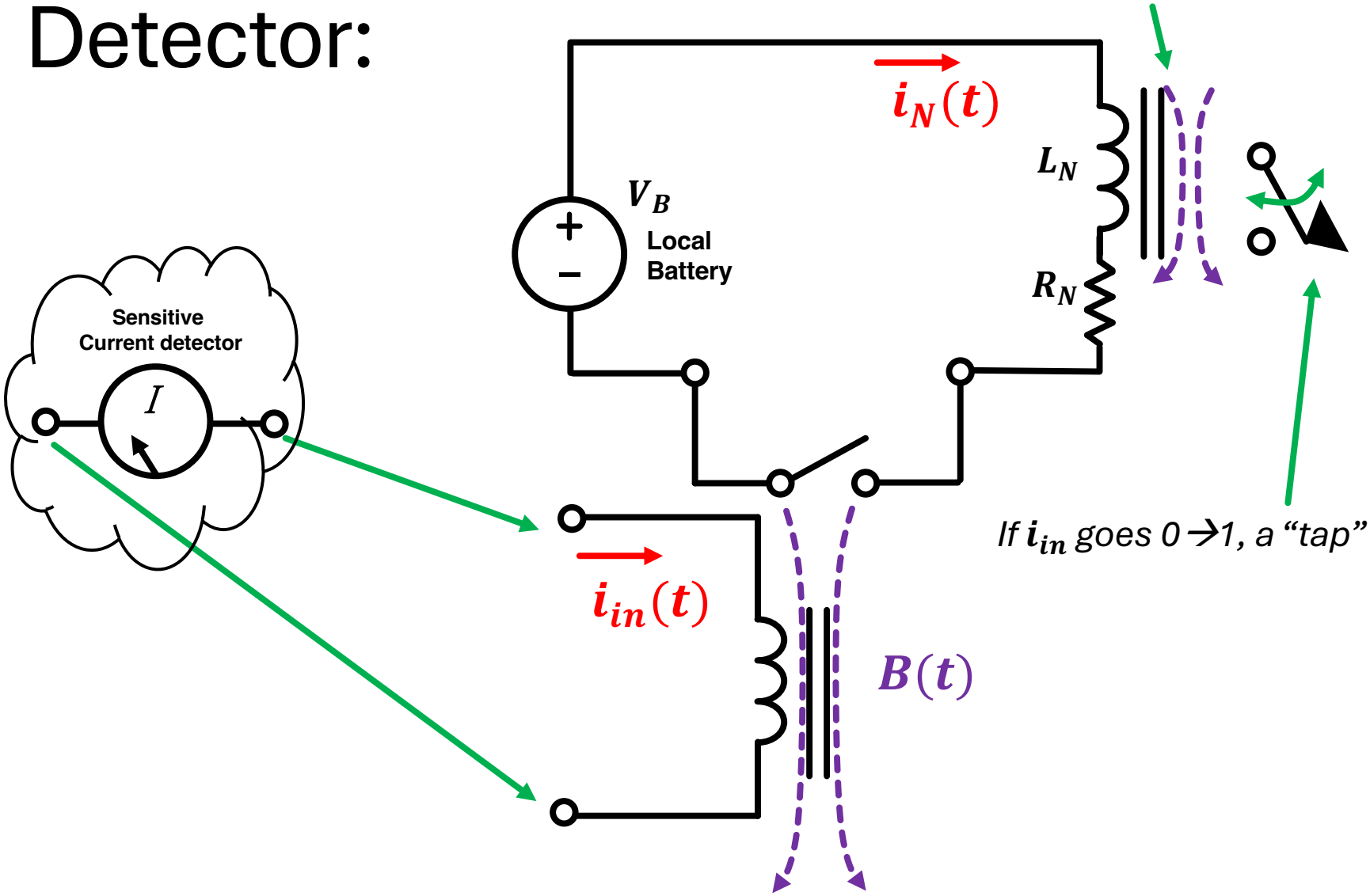
http://members.kos.net/sdgagnon/te5.html

Returning to our Telegraph Circuit



- Each station would have a local sensitive current meter built around a relay and a sounder!

A Relay-based Current Detector:



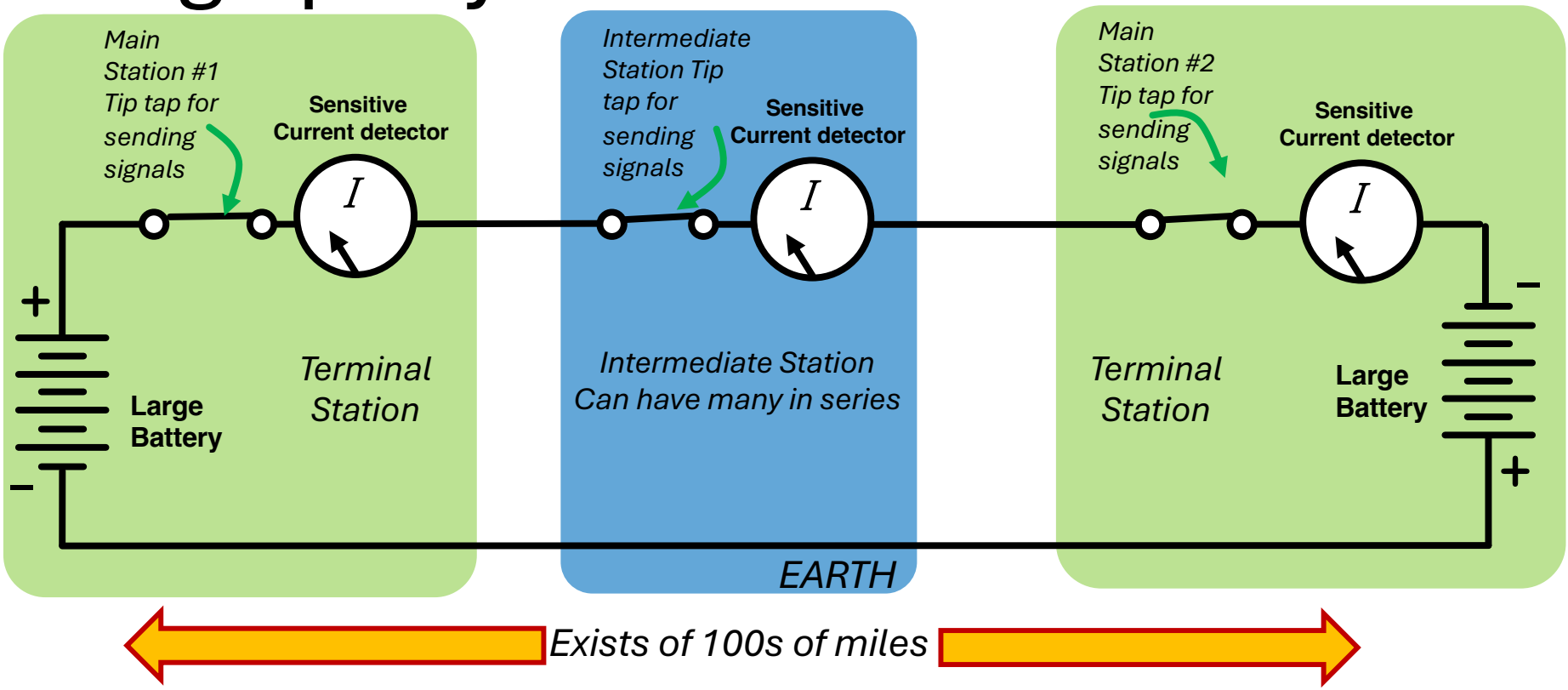
Amplification

- Many miles from the terminal station $\boldsymbol{i_{in}}$ (and $\boldsymbol{v_{in}}$) will be small due to parasitic losses in the line. They would lack the ability to make much signal for a local listener

- A well-designed relay acted as an **amplifying device** for the signal so that the operator could easily hear the message!

- Not a strictly electrical amplifier. (but an electromechanical amplifier)

Telegraph system



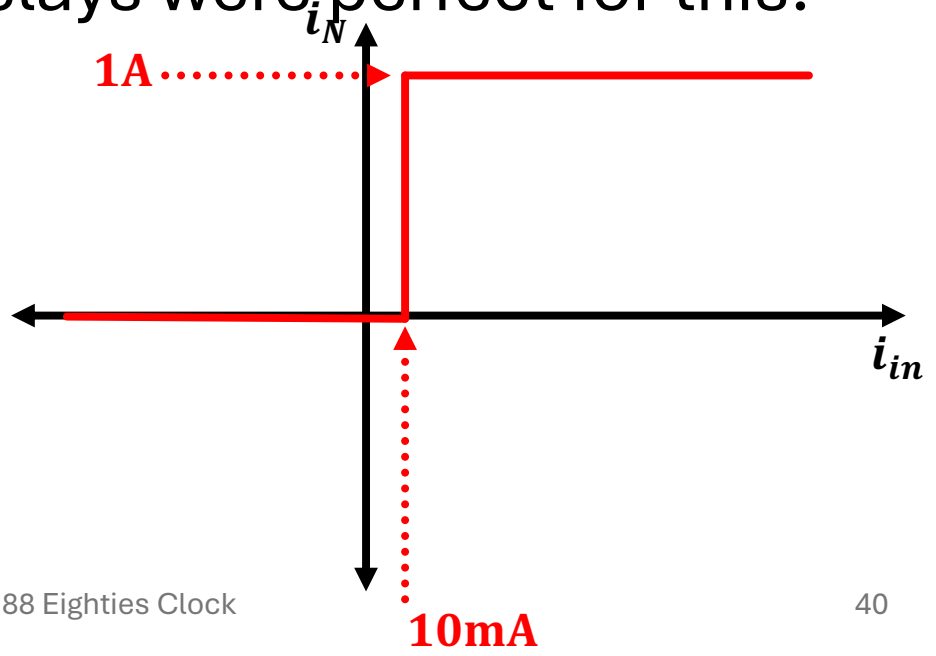
- At rest all stations would have their keys closed in a listen/propagate mode

Telegraph Operation

- Each run of telegraph was a “party line”, meaning all stations on that run could listen in
- At “rest” all stations kept their keying switches closed!
  - Current therefore almost always flowed through the entire system!
  - Wasted power, but actually was beneficial for crappy early batteries
- If a station needed to broadcast, that station would open up their key, breaking entire circuit!
- *All* stations on line would detect this break in current and know to wait for start of message
- Transmitter would then send message

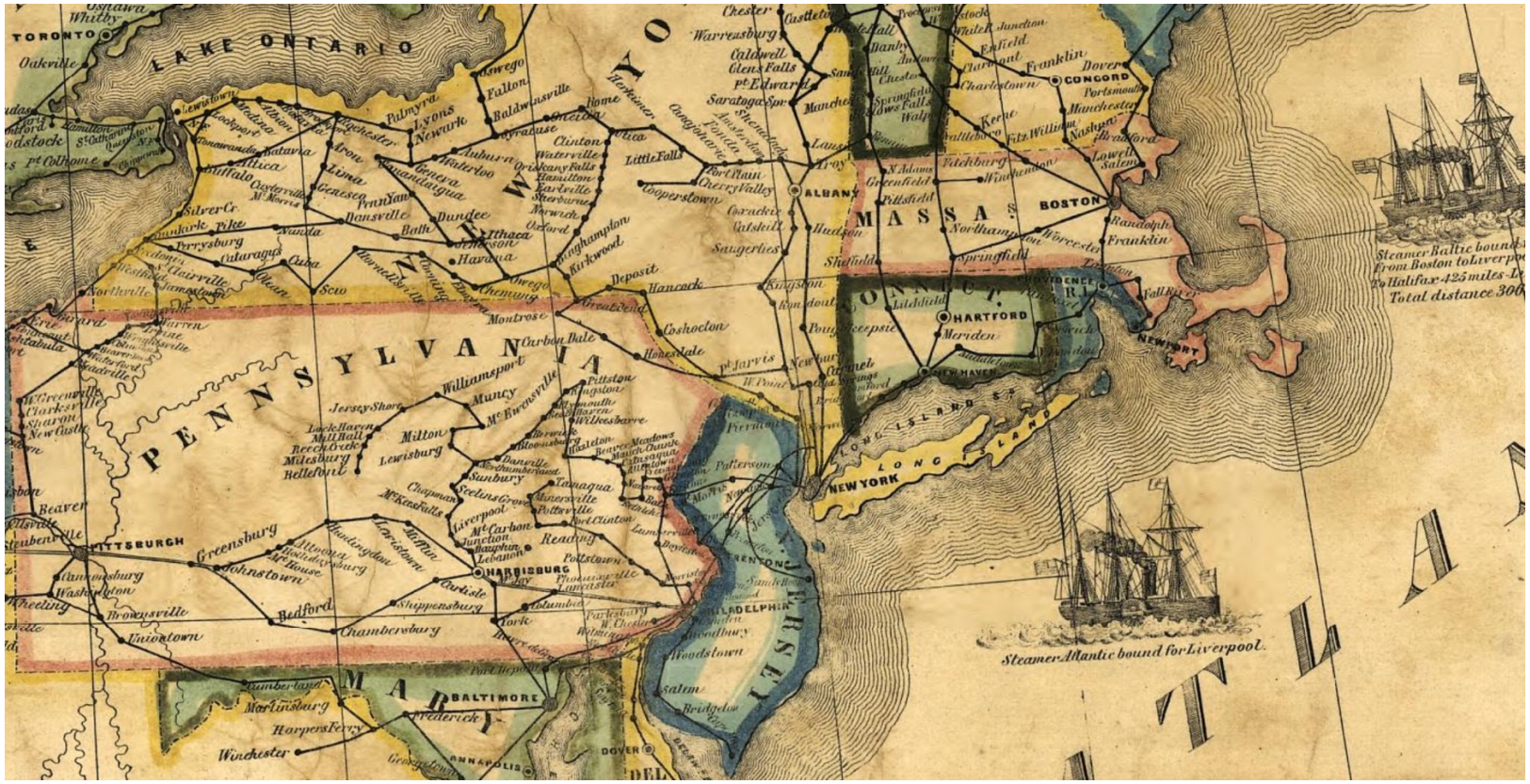
Noise

- Noise in these systems was disgusting!
  - Parasitic leakage
  - Earth Ground
  - Leakage from nearby circuits
- A digital communication scheme was used (on-off signaling) because of all this. Electromechanical relays were perfect for this!



Build Up Network of these Circuits

- North East US telegraph network, 1853

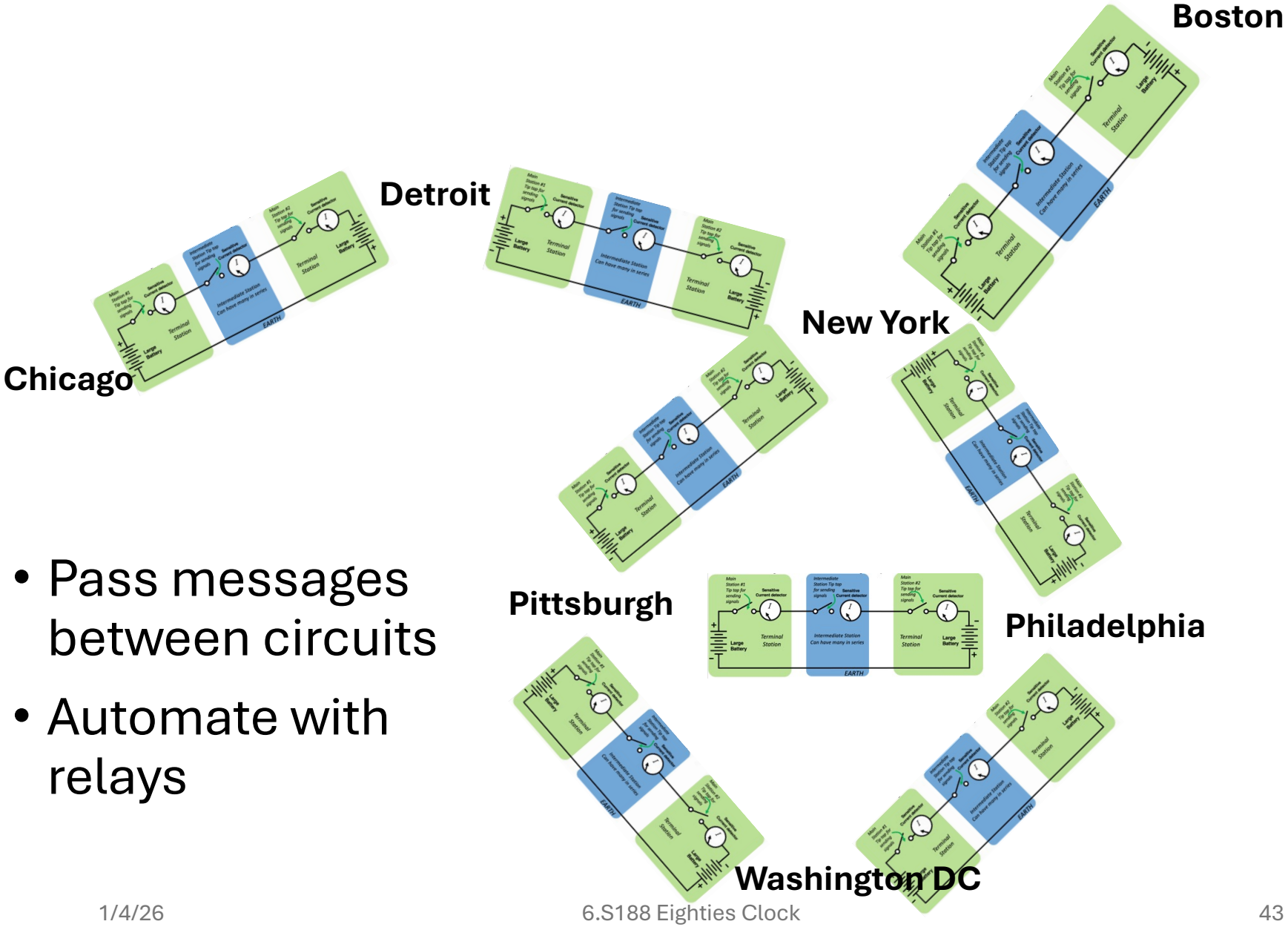


https://www.loc.gov/resource/g3701p.ct000084/?r=0.577,0.18,0.283,0.154,0

Telegraph Network Europe 1856

http://members.kos.net/sdgagnon/te3.html

Build Up Network of these Circuits



- Pass messages between circuits
- Automate with relays

People were intimately exposed to the telegraph (and follow-on telephone)

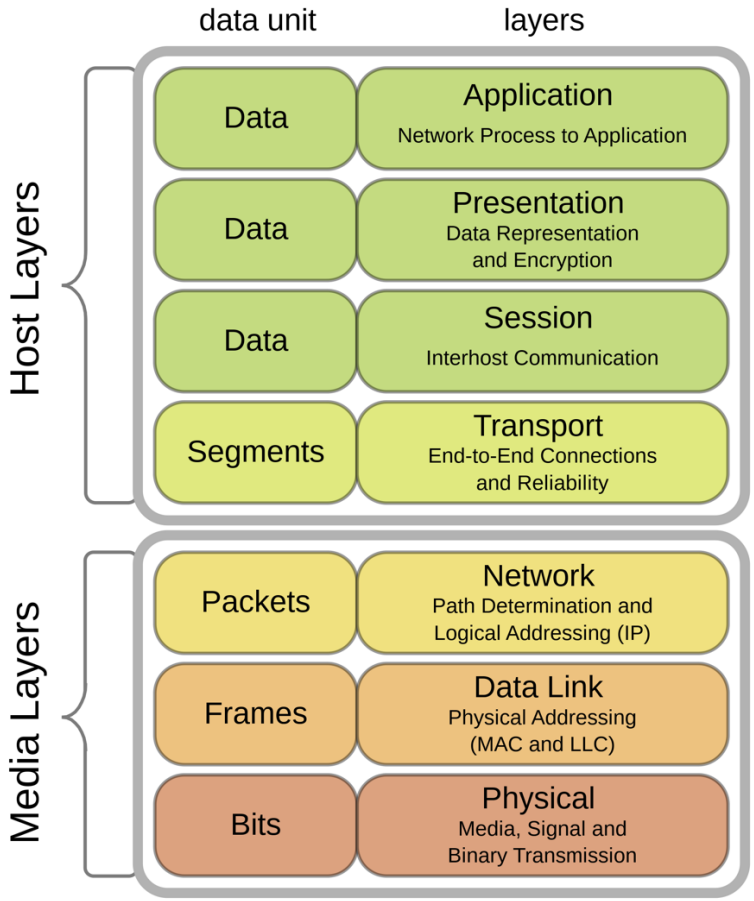
Stockholm Central
Telephone/Telegraph Tower

New York 1887

https://io9.gizmodo.com/photos-from-the-days-when-thousands-of-cables-crowded-t-1629961917

Telegraph Operation

- Had to develop an entire communication protocol!
- Morse was your alphabet (in US), but then you had to have addressing, message-begin/end/qualifiers, forwarding, etc…
- Much of our modern "protocol stack" can be reverse-applied to what they came up with!
- Early telephone and telegraphy had very complicated routing systems

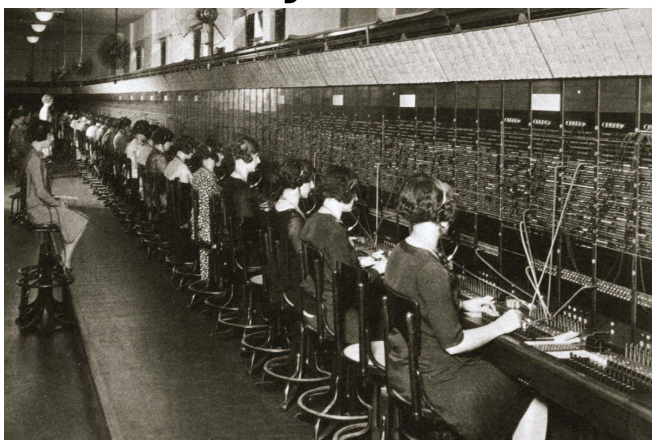


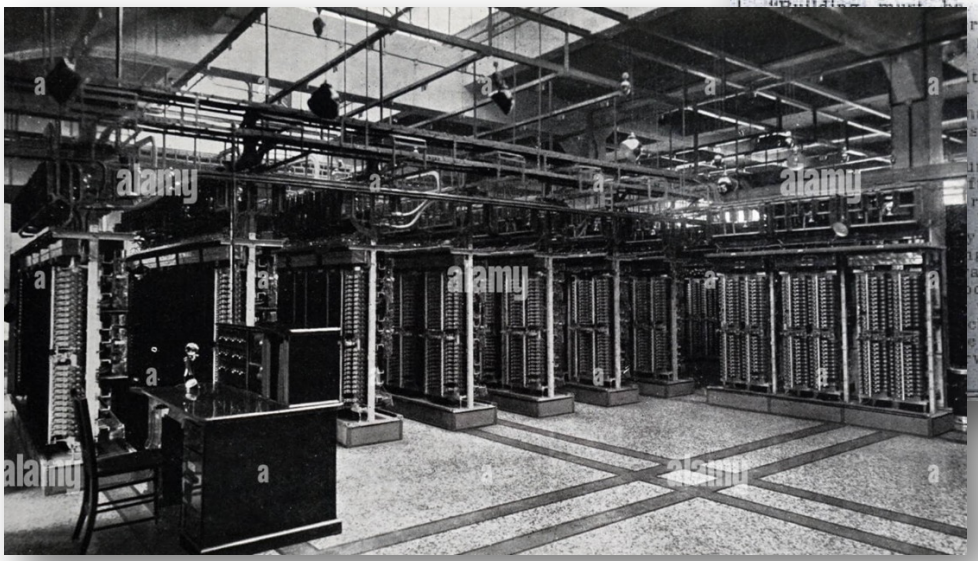
American Telephone and Telegraph

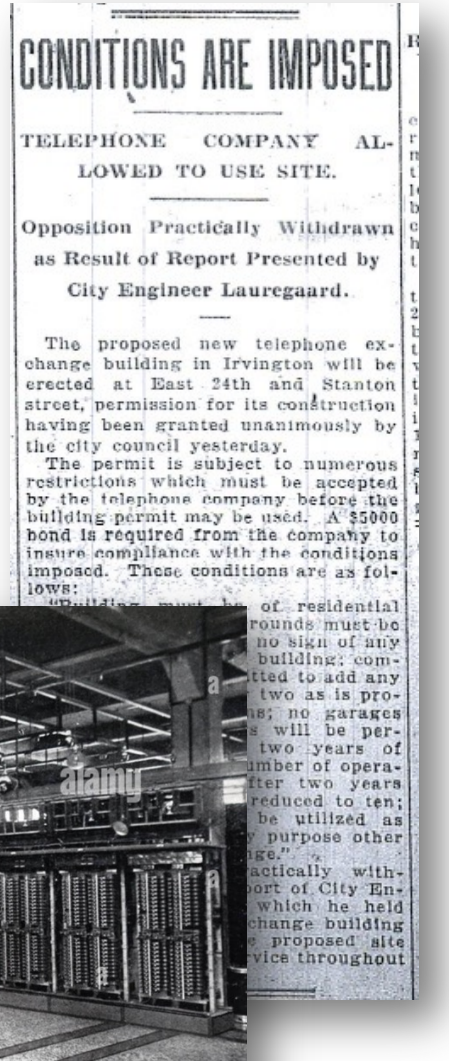
- Known as “Ma Bell” this company developed from the 1880s up through the 1950s before being broken up in anti-trust actions
- Was the largest employer in the US.
- Company rivaled federal government in size and operating costs.
- Poured massive resources into improving their system.
- Spun out Bell Laboratories from which tons of EECS developments and inventions were produced and very liberally licensed

Telephone Relay Stations

- A standard telephone relay station serving 10,000 customers would have about 60,000 relays in it.
- Hundreds of these throughout the country.







CONDITIONS ARE IMPOSED

TELEPHONE COMPANY ALLOWED TO USE SITE.

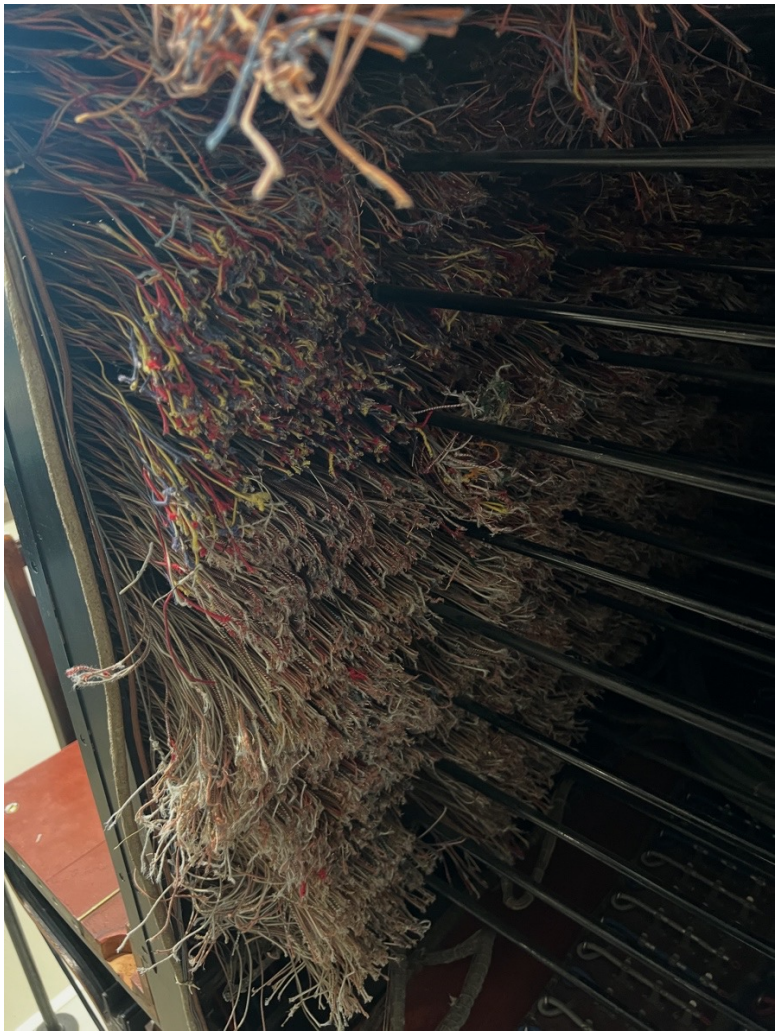
Opposition Practically Withdrawn as Result of Report Presented by City Engineer Lauregaard.

The proposed new telephone exchange building in Irvington will be erected at East 24th and Stanton street, permission for its construction having been granted unanimously by the city council yesterday.

The permit is subject to numerous restrictions which must be accepted by the telephone company before the building permit may be used. A $5000 bond is required from the company to insure compliance with the conditions imposed. These conditions are as follows:

https://www.alamy.com/stock-photo-the-leeds-automatic-telephone-exchange-circa-1920-131277159.html?imageid=AC1DD1CB-8FA5-4119-94A8-E90646857EDB&pn=1&searchId=baae1e609757033cbe6a70eebbebc1d9&searchtype=0

https://alamedahistory.org/2012/02/19/mystery-building-at-ne-24th-and-stanton-has-always-been-a-telephone-exchange/



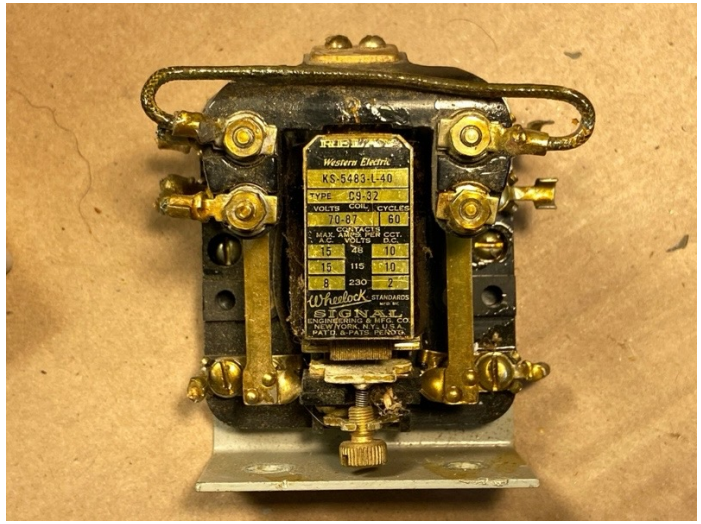
https://www.newscientist.com/article/mg25734260-100-a-message-for-today-from-last-centurys-vast-telephone-exchanges/

Electromechanical Relay

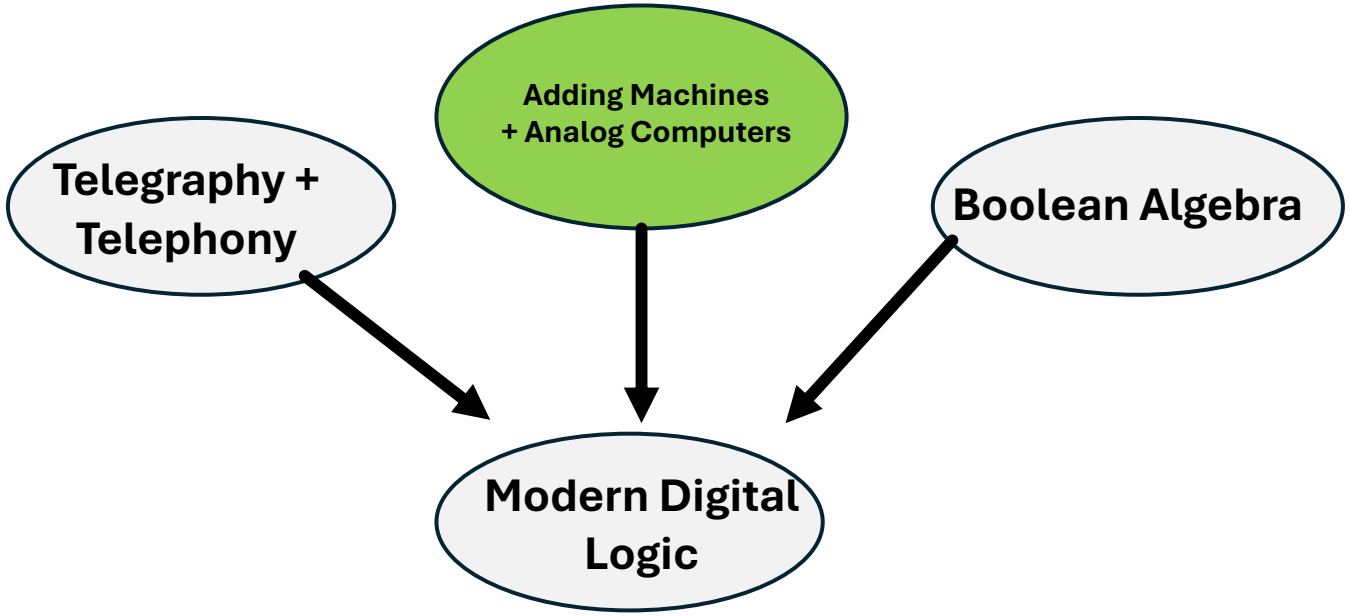
- The entire telephone and telegraph network was built around electromechanical relays that were extremely well-engineered.

- Western Electric (the Bell subsidiary that specialized in manufacturing) produced thousands of types of specialized relays.

- Extremely optimized technology



https://www.ebay.com/itm/114161685726

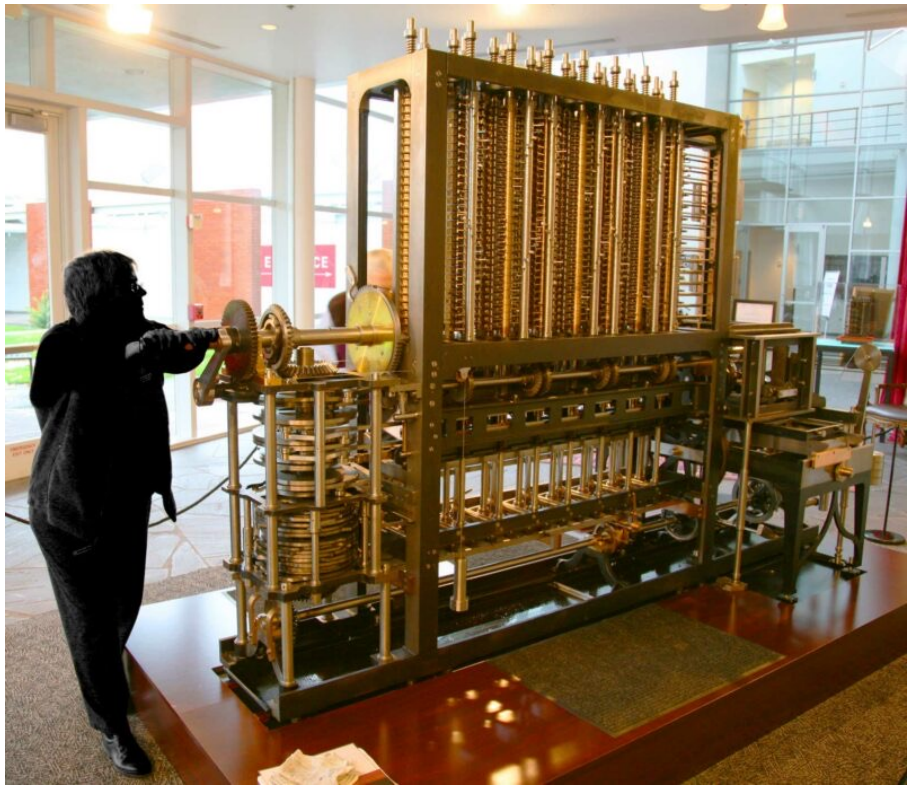


Adding Machines

Doing Math and Maybe Momre Electromechanically

Charles Babbage (1823)

- Designed the Difference Engine (built by Joseph Clement)
- Purely mechanical numerical solver that worked in 10's complement
- Programmed by several people including Ada Lovelace

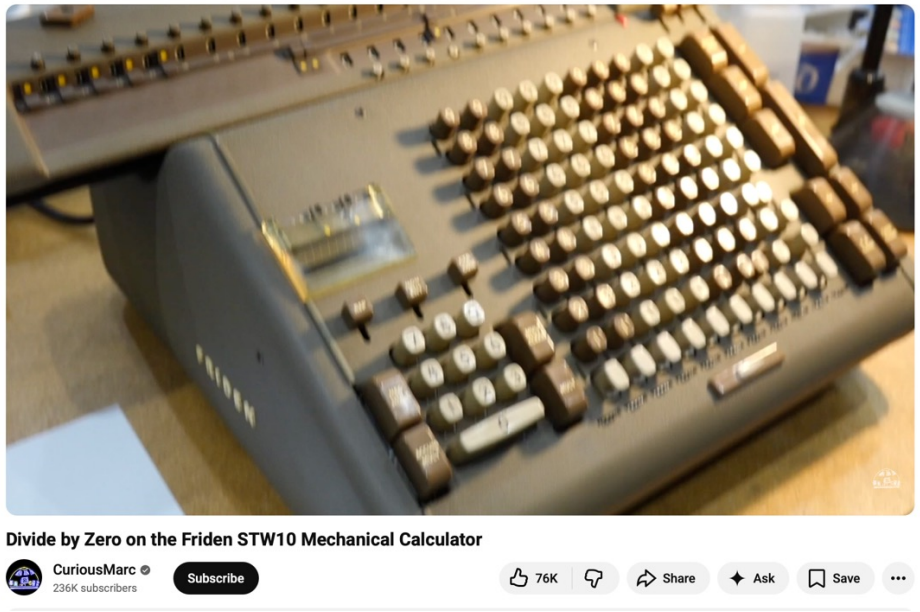


https://www.britannica.com/biography/Charles-Babbage

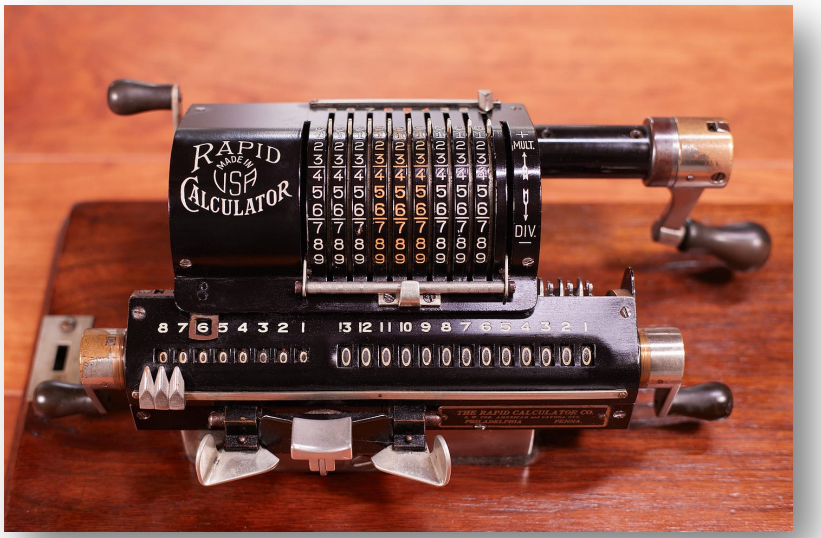
https://cronatec.ch/how-did-the-difference-engine-work/

Adding/Tabulating Machines

- 1870s and onwards featured tons of startups developing machines that could do arithmetic and/or tabulate information using gears and motors.



https://www.youtube.com/watch?v=7Kd3R_RlXgc



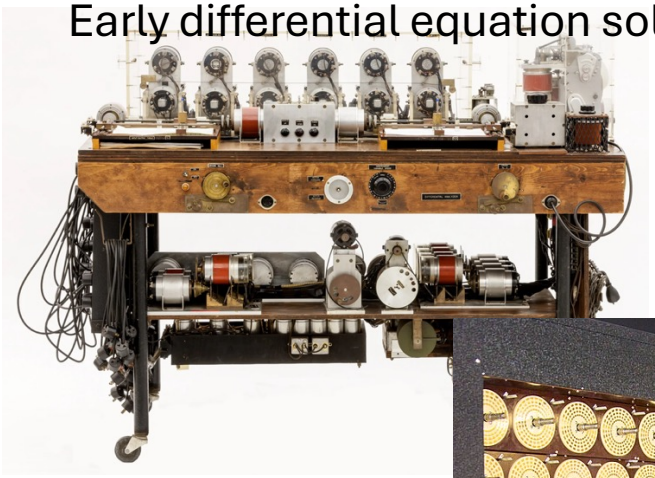
https://lcamtuf.substack.com/p/a-brief-history-of-counting-stuff

Electromechanical “Computing”

- Wasn’t just limited to arithmetic.

- Electromechanical machines were developed to solve differential equations or perform other mathematical operations like

Early differential equation solver



Alan Turing et al’s “Bombe”

https://en.wikipedia.org/wiki/Bombe

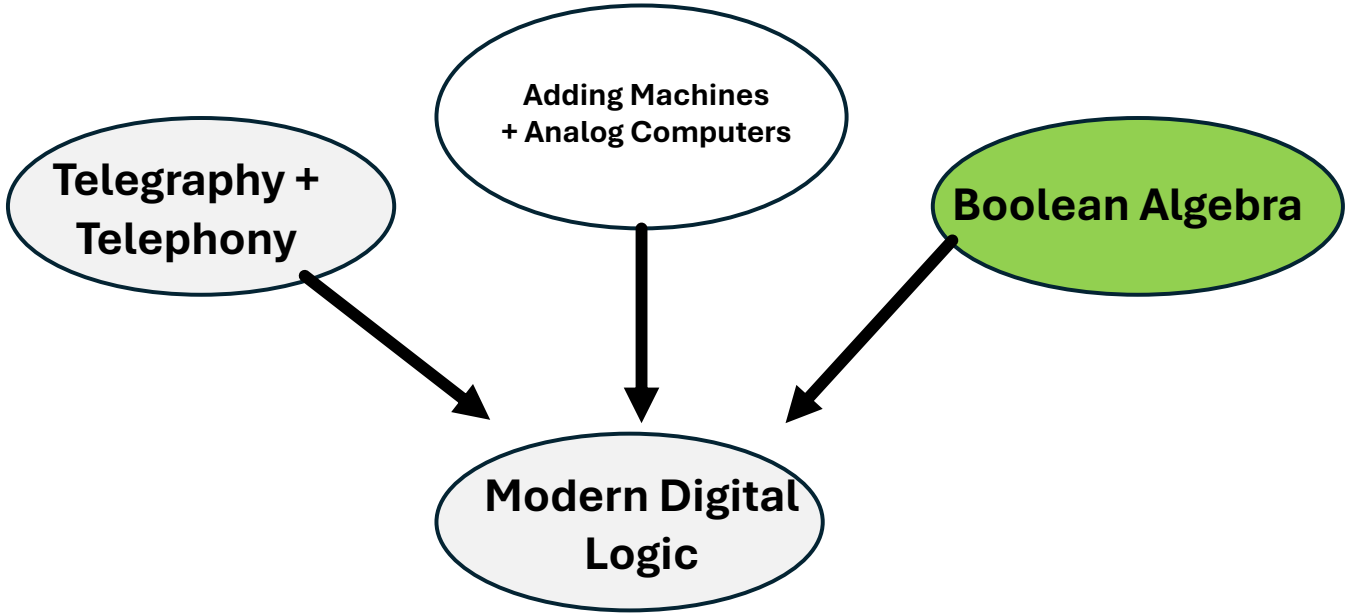
https://spectrum.ieee.org/the-shocking-truth-behind-arnold-nordsiecks-differential-analyzer

Vannevar Bush

- MIT's Vannevar Bush took the electromechanical calculator approach to the extreme.
- The "Differential Analyzer" in 1930s

MIT building 13 was named after him

- Could solve very challenging differential equations purely electromechanicallyy
- One of his grad students was Claude Shannon who's project was to add some improvements onto the Differential Analyzer using relays to route signals….



Boolean Algebra

A different type of math

George Boole (1815-1864)

- Faculty at Queen's College in Cork, Ireland
- Mathematician and philosopher
- Wrote *The Laws of Thought (*1854*)* in which he offered an attempt at mathematical logic.

https://www.ucc.ie/en/heritage/history/people/ucc-staff/professor-george-boole/

Few other folks also contributed

- De Morgan cleaned up and formalized some of Boole's work

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

- William Hamilton also

- The rules/identities/etc…form basis of what we have today..

Boolean Algebra

- This work is a weird text…half math and half philosophy
- It is/was completely removed from what we want to think of it (being applied to computer science)

2nd. Evil of imperfection is not absolute evil.

3rd. Natural evil is either a consequence of evil of imperfection, or it is compensated with greater good, or it is a consequence of moral evil.

4th. That which is either a consequence of evil of imperfection, or is compensated with greater good, is not absolute evil.

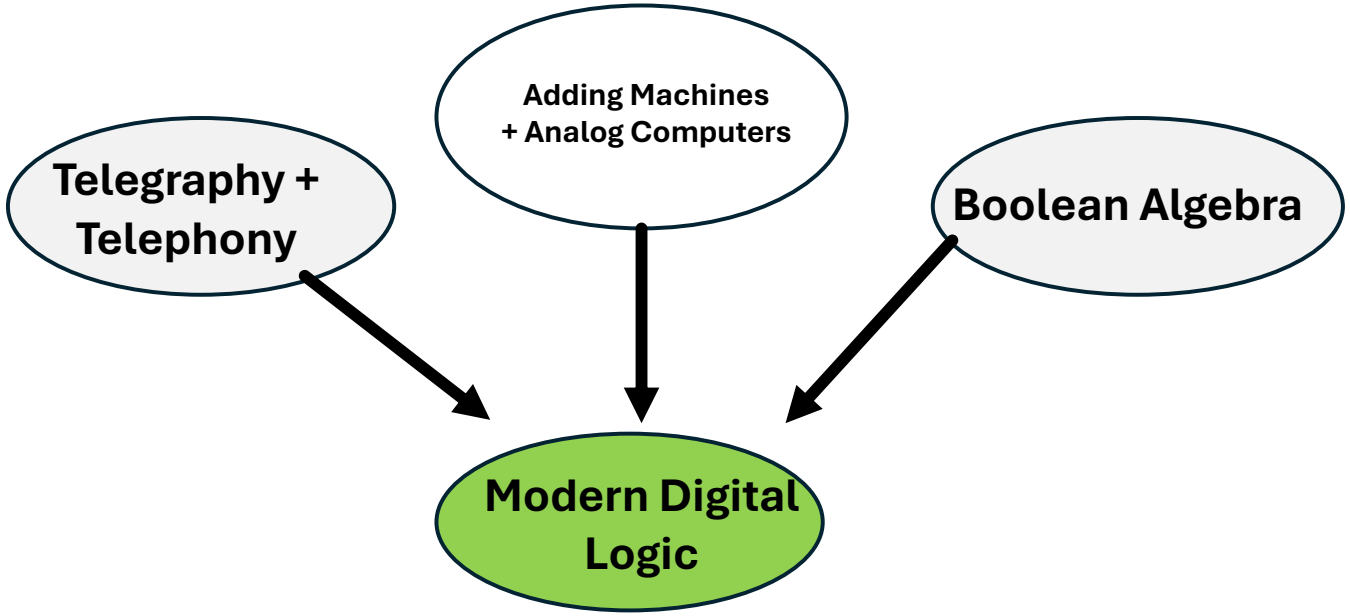
5th. All absolute evils are included in reputed evils.

To express these premises let us assume—

$w =$ reputed evil.
$x =$ evil of imperfection.
$y =$ natural evil.
$z =$ moral evil.
$p =$ consequence of evil of imperfection.
$q =$ compensated with greater good.
$r =$ consequence of moral evil.
$t =$ absolute evil.

Then, regarding the premises as Primary Propositions, of which all the predicates are particular, and the conjunctions *either, or*, as absolutely disjunctive, we have the following equations:

$$w = v\,\{x(1-y)(1-q) + y(1-x)(1-z) + z(1-x)(1-y)\}$$
$$x = v(1-t).$$
$$y = v\,\{p(1-q)(1-r) + q(1-p)(1-r) + r(1-p)(1-q)\}$$
$$p(l-q) + q(l-p) = v(1-t).$$
$$t = vw\,.$$


Adding Machines
+ Analog Computers
Telegraphy +
Telephony
Boolean Algebra
Modern Digital
Logic

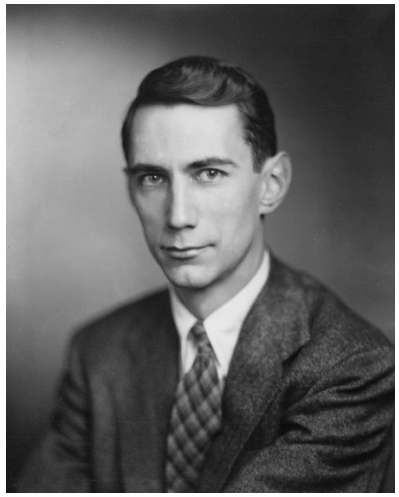
Claude

Claude Shannon

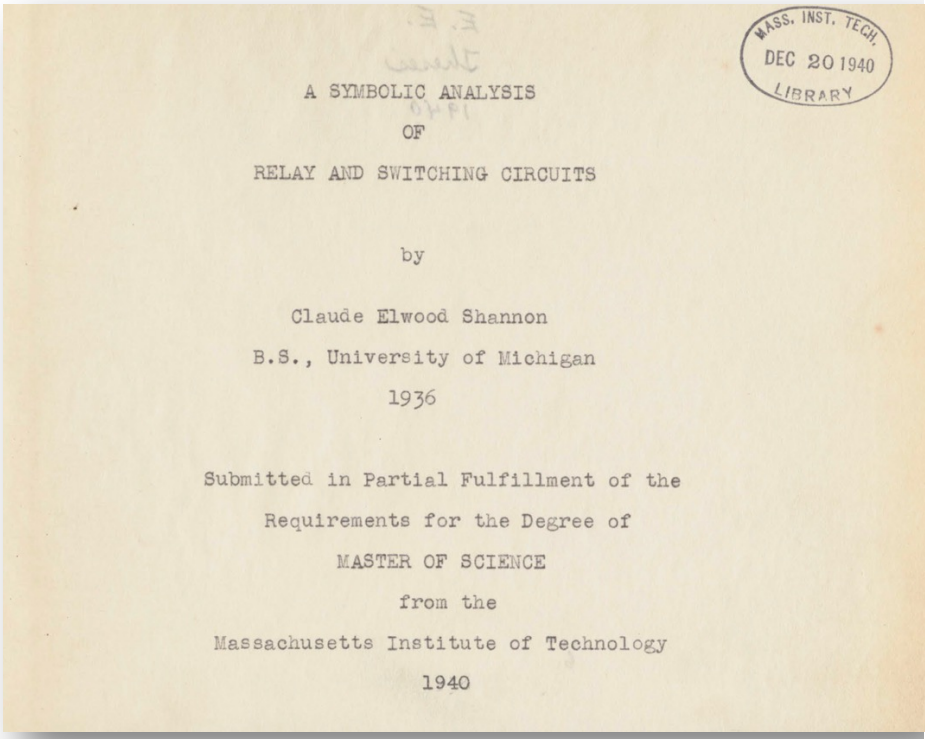
- Undergrad at University of Michigan in EE and Math
- Took a philosophy elective while at Michigan that covered Boole's Algebra one week.
- Came to MIT for grad school.
- Got job working for Vannevar Bush to improve differential analyzer

https://www.lindahall.org/about/news/scientist-of-the-day/claude-shannon/

Claude Shannon

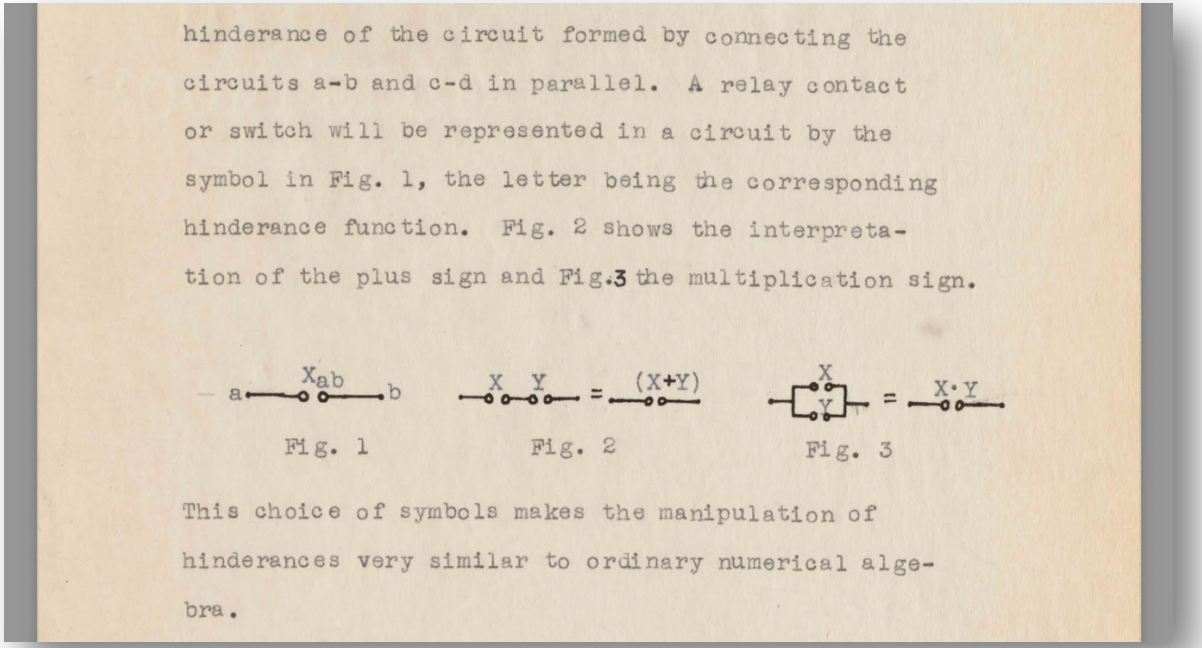


- His PI wanted him to integrate some relays into the differential analyzer.
- He started doing it and kinda realized the Boole's logic stuff from that philosophy class would describe.
- Wrote his M.S. thesis on it.



https://dspace.mit.edu/handle/1721.1/11173

Whole Thesis is Good Reading



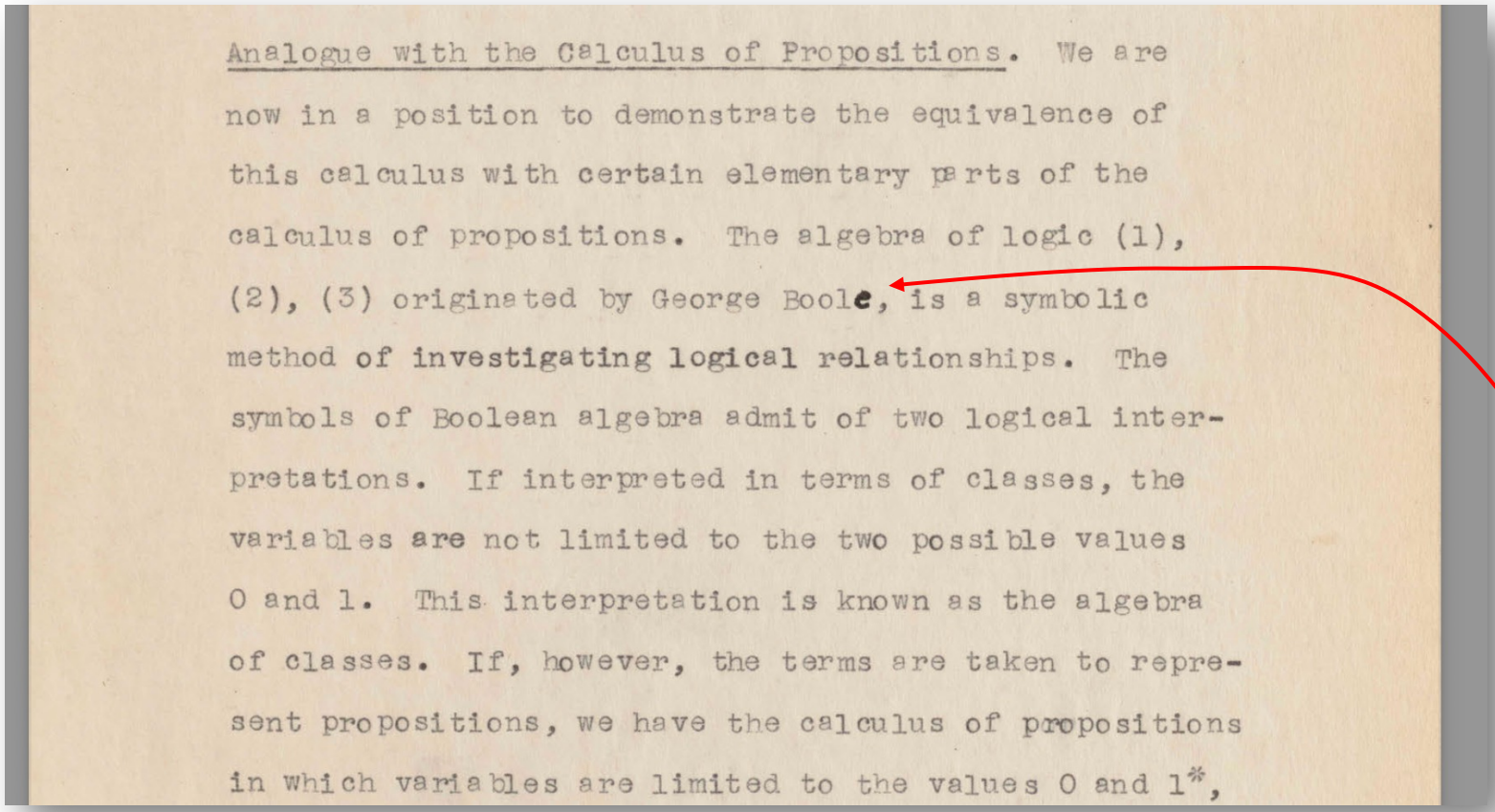
hinderance of the circuit formed by connecting the circuits a-b and c-d in parallel. A relay contact or switch will be represented in a circuit by the symbol in Fig. 1, the letter being the corresponding hinderance function. Fig. 2 shows the interpretation of the plus sign and Fig.3 the multiplication sign.

Fig. 1 Fig. 2 Fig. 3

This choice of symbols makes the manipulation of hinderances very similar to ordinary numerical algebra.

- Many figures are hand-written
- Links switching circuits to Boolean Algebra

Term “hinderance” which you never see again



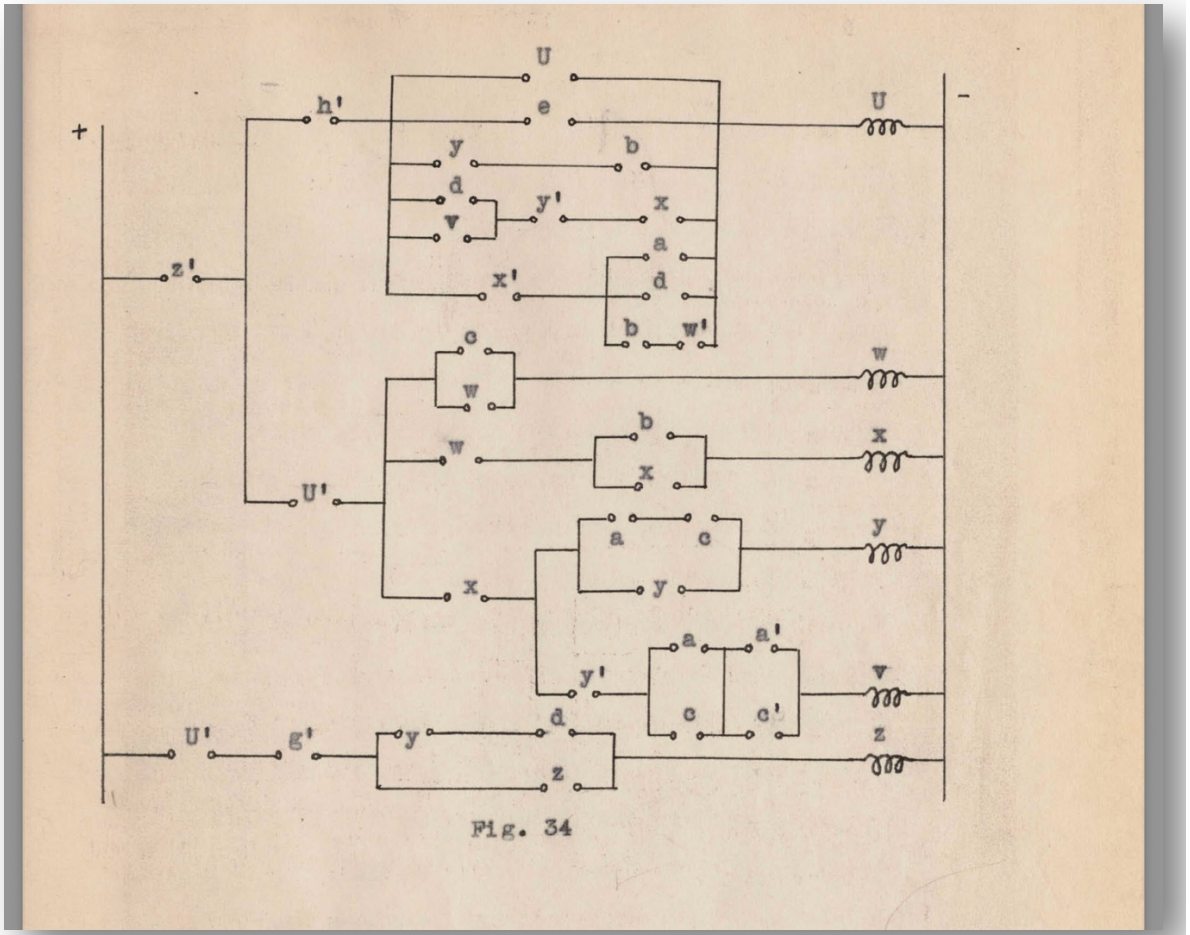
Analogue with the Calculus of Propositions. We are now in a position to demonstrate the equivalence of this calculus with certain elementary parts of the calculus of propositions. The algebra of logic (1), (2), (3) originated by George Boole, is a symbolic method of investigating logical relationships. The symbols of Boolean algebra admit of two logical interpretations. If interpreted in terms of classes, the variables are not limited to the two possible values 0 and 1. This interpretation is known as the algebra of classes. If, however, the terms are taken to represent propositions, we have the calculus of propositions in which variables are limited to the values 0 and 1*,

- Links his circuits to algebra by Boole

Actual typo…must not have had time to retype whole page

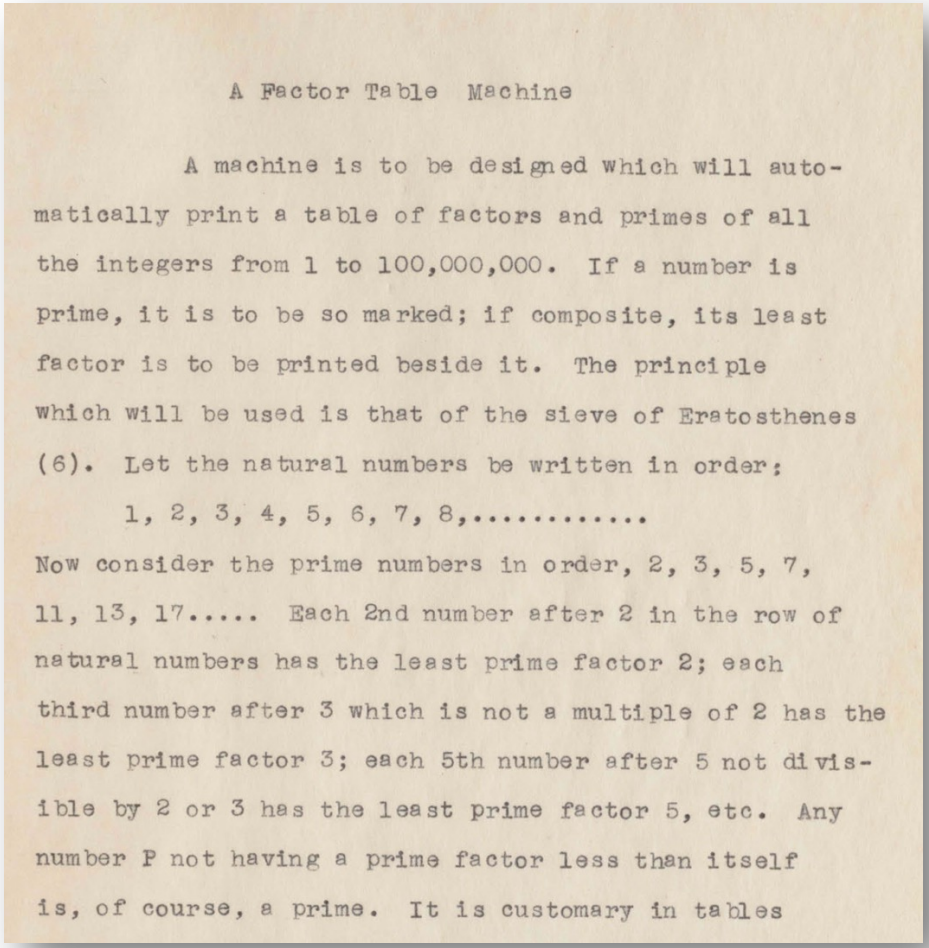
Quite Complicated Examples/Circuits

- Not a modern logic gate symbol to be found!
- Just relay coils and switch on/off locations linked with equations



Claude's Future Work Section...

- Numerous example problems
- Final one is to find prime numbers up to 100,000,000 using the Sieve of Eratosthenes



A Factor Table Machine

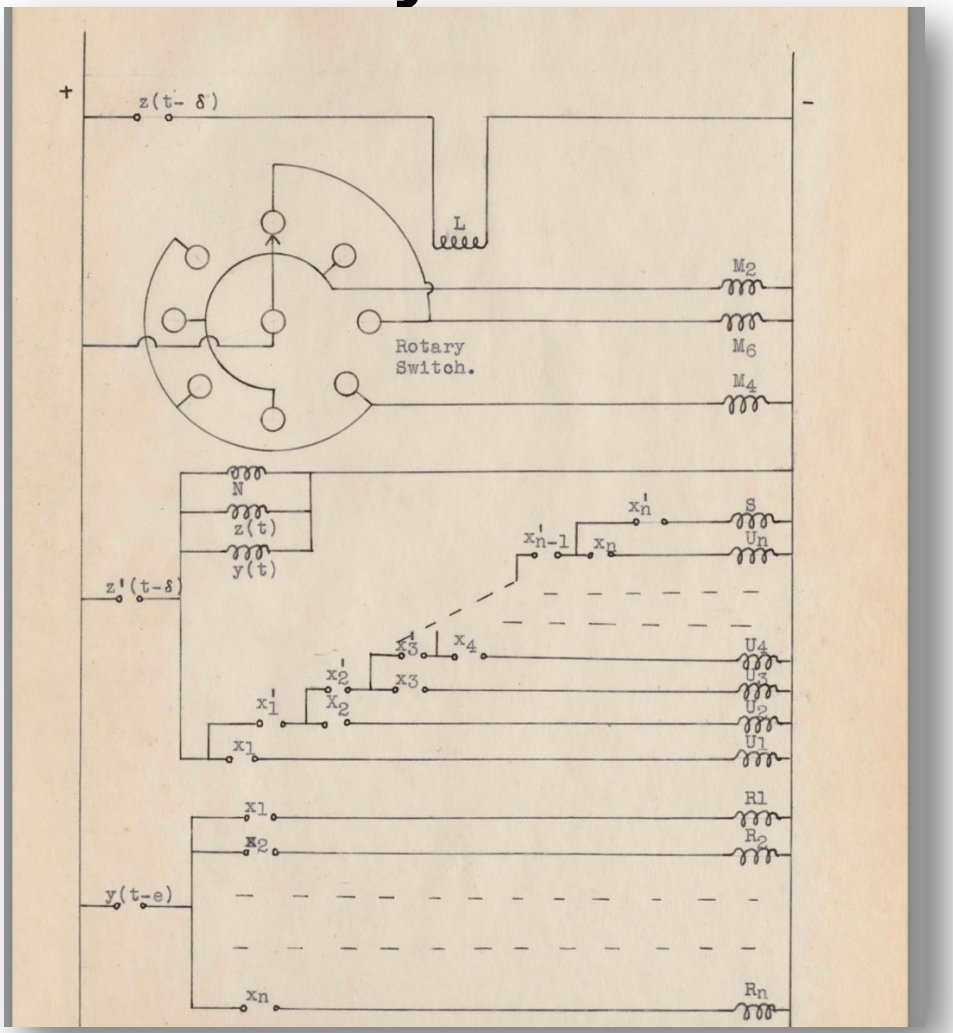
A machine is to be designed which will automatically print a table of factors and primes of all the integers from 1 to 100,000,000. If a number is prime, it is to be so marked; if composite, its least factor is to be printed beside it. The principle which will be used is that of the sieve of Eratosthenes (6). Let the natural numbers be written in order:

1, 2, 3, 4, 5, 6, 7, 8,.....

Now consider the prime numbers in order, 2, 3, 5, 7, 11, 13, 17..... Each 2nd number after 2 in the row of natural numbers has the least prime factor 2; each third number after 3 which is not a multiple of 2 has the least prime factor 3; each 5th number after 5 not divisible by 2 or 3 has the least prime factor 5, etc. Any number P not having a prime factor less than itself is, of course, a prime. It is customary in tables

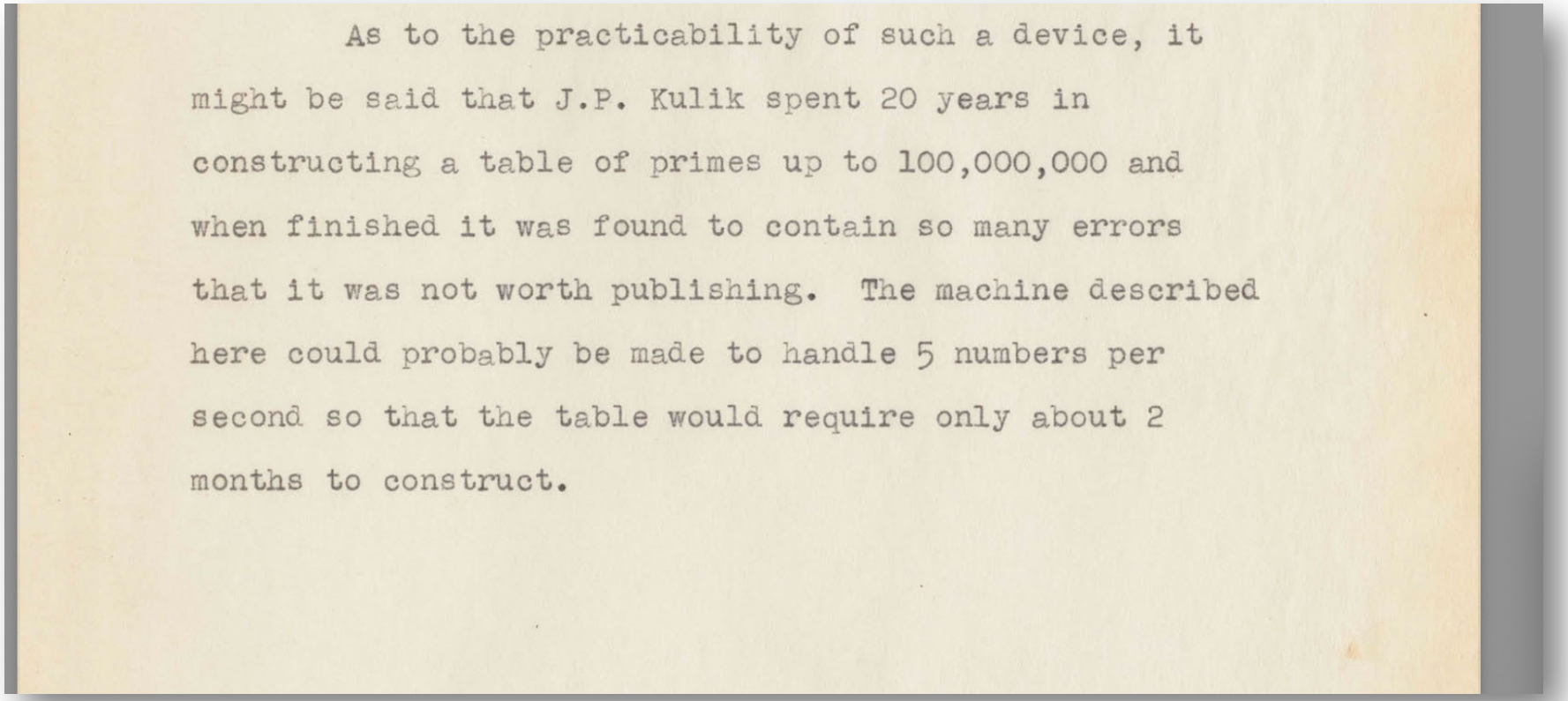
Includes Schematic for System

- It is legit but still a little bit of a thought-experiment at the end



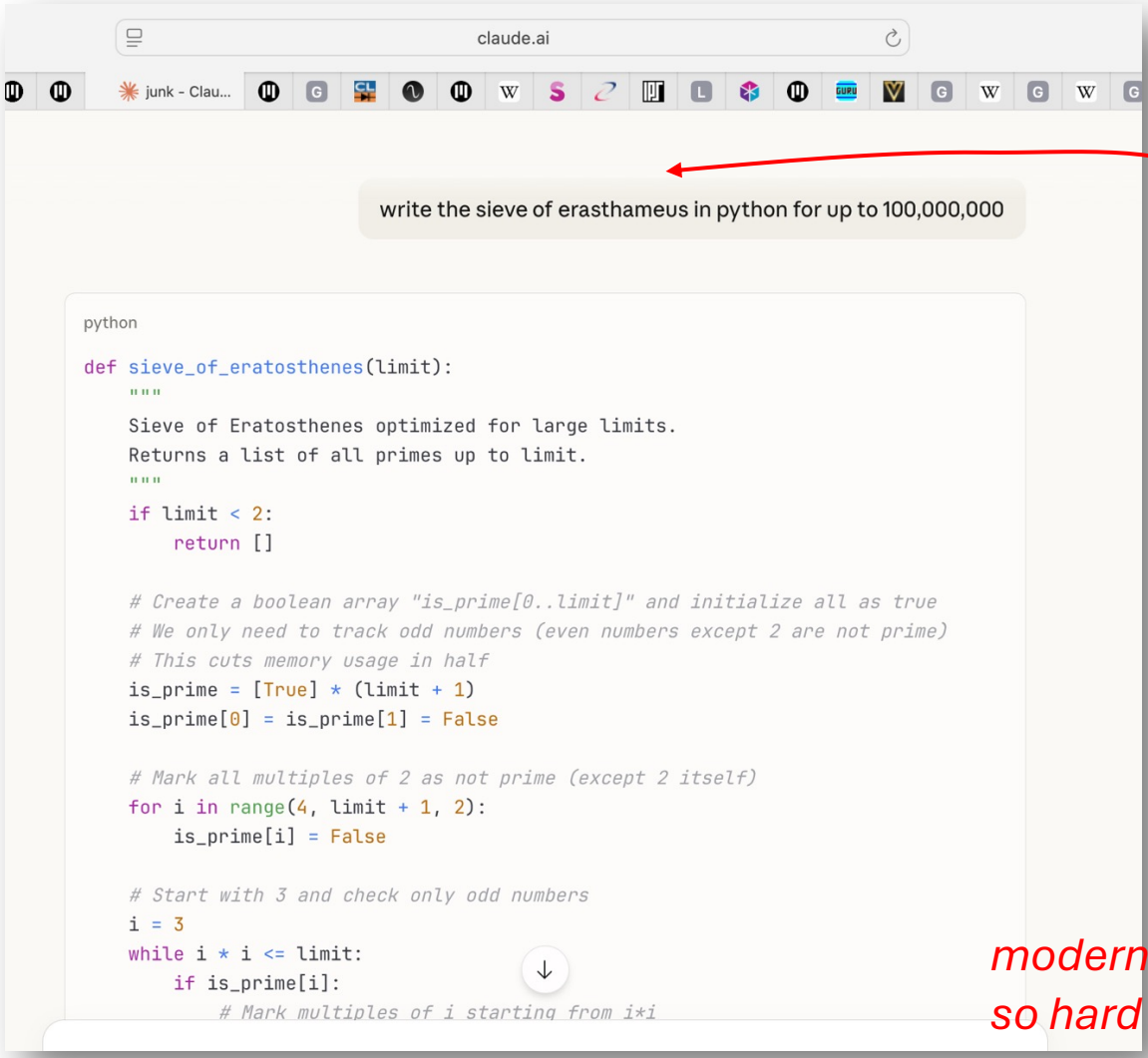
Final Mic Drop

- Conservatively estimates that his circuit could…



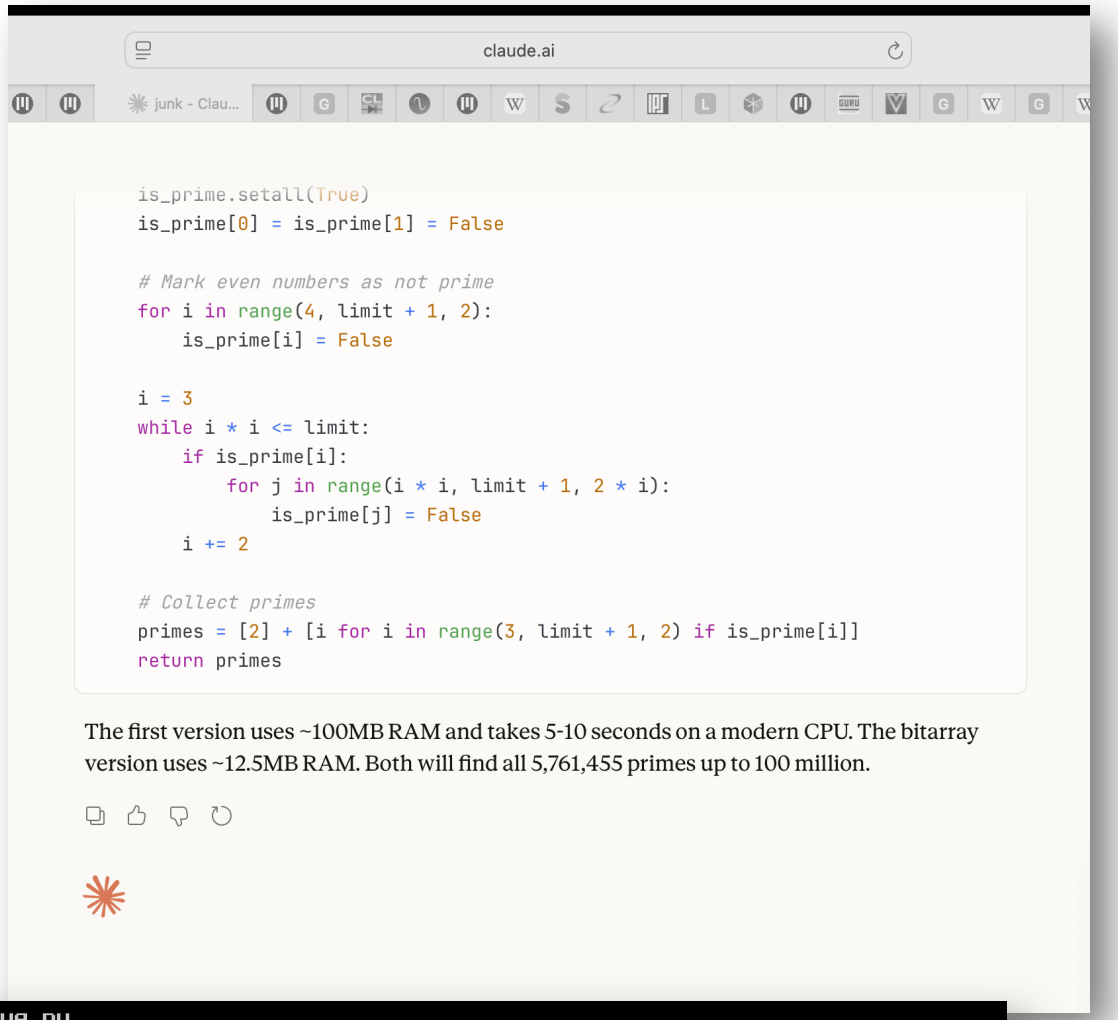
As to the practicability of such a device, it might be said that J.P. Kulik spent 20 years in constructing a table of primes up to 100,000,000 and when finished it was found to contain so many errors that it was not worth publishing. The machine described here could probably be made to handle 5 numbers per second so that the table would require only about 2 months to construct.

Return to Claude



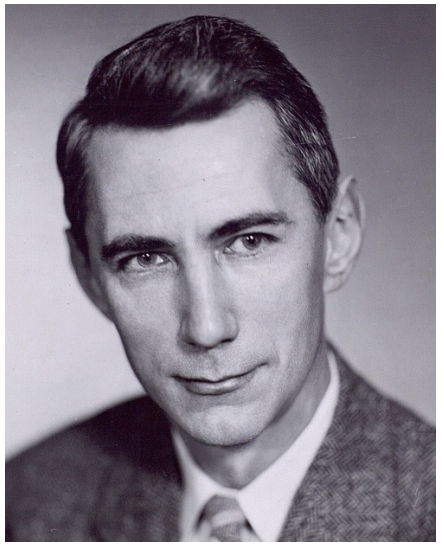
Run it

- Hmmm...




```
jodalyst@jodalysts-MacBook-Air ~ % python3 sieve.py
Finding all primes up to 100,000,000...
Found 5,761,455 primes in 7.05 seconds
First 10 primes: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29]
Last 10 primes: [99999787, 99999821, 99999827, 99999839, 99999847, 99999931, 99999941, 99999959, 99999971, 99999989]
Largest prime: 99,999,989
jodalyst@jodalysts-MacBook-Air ~ %
```

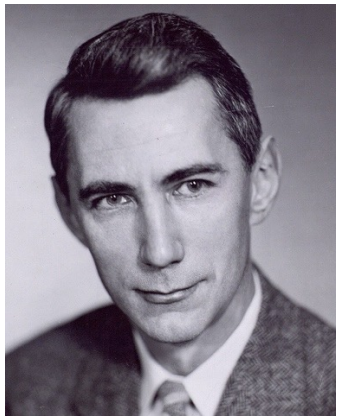
Claude Shannon



- After his MS thesis, the guy never really did anything else. Kinda peaked in college and burned out.

- JK JK jk JK JK

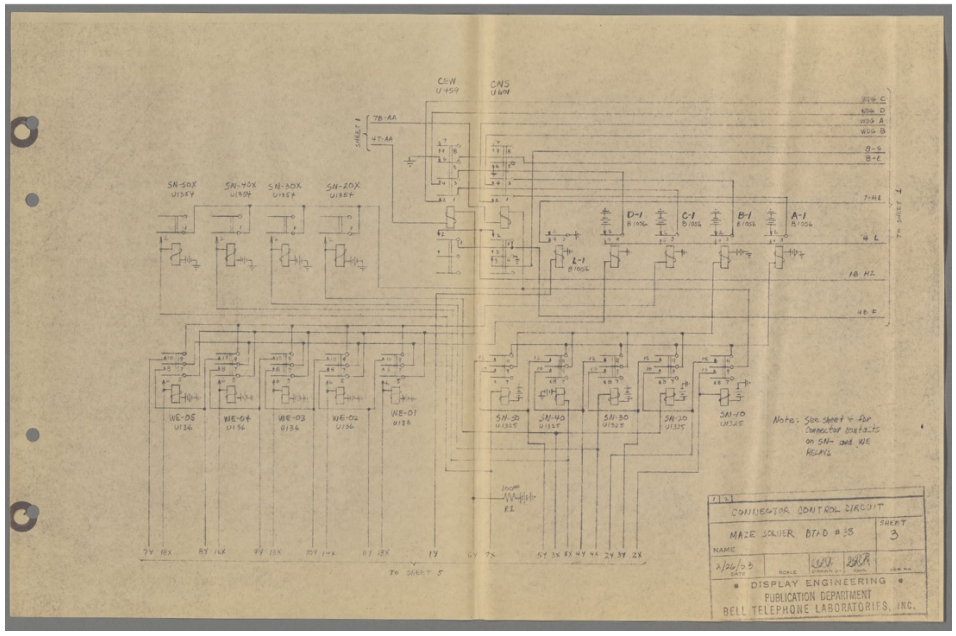
Unbelievable



- Got to work with Einstein during internship at Princeton
- Worked at Bell Labs
- Invented signal-flow graphs
- Proved one-time-pad cryptographic security
- Developed *all* of information theory
- Defined what a bit is
- Foundational work in NLP
- Early implementations of AI and Maze-learning
- Faculty at MIT from 1950s to late 1970s.

Theseus by Claude

- Early maze-solving mouse design built entirely using relay logic

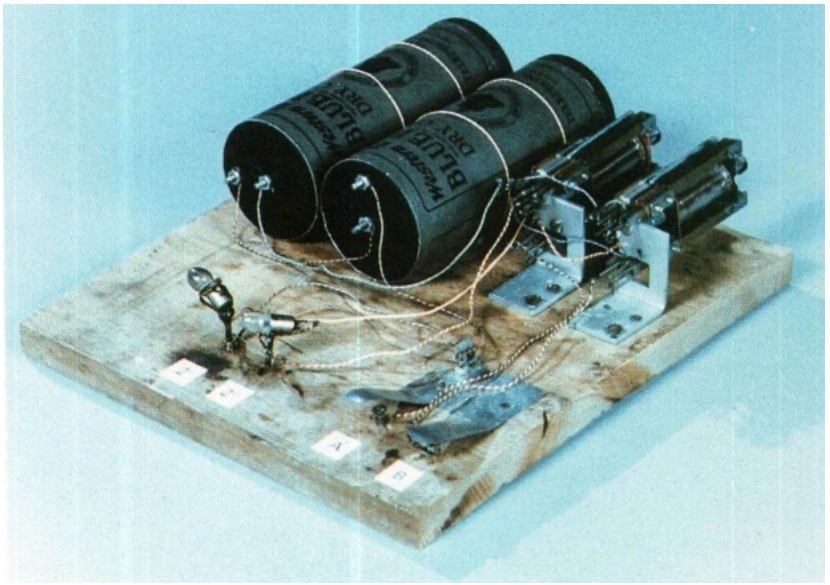


~1953

https://mitmuseum.mit.edu/collections/object/2007.030.001

Relay Logic Really Took Off

- Relays were so ubiquitous and robust thanks to AT&T that coupled with Shannons theory, digital design really expanded rapidly.

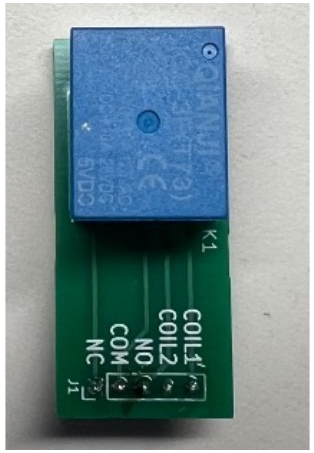


- In 1937, George Stiblitz at Bell Labs took two relays, wired them up in a weird way and could add two base-2 numbers

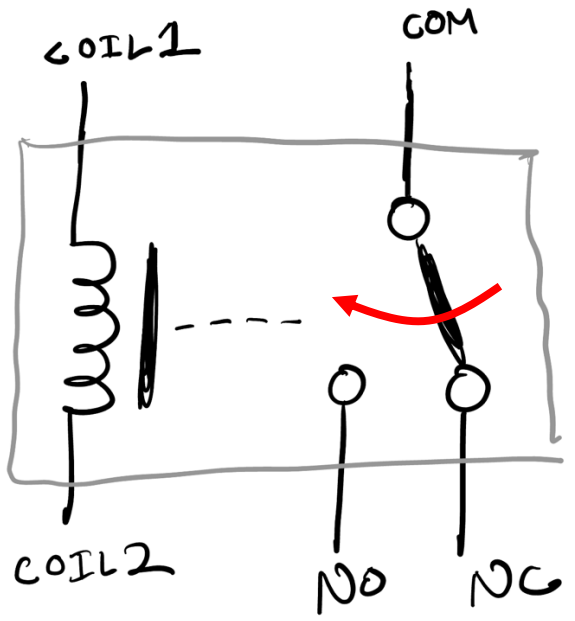
https://technicshistory.com/2017/05/10/lost-generation-the-relay-computers/

Computation with Relays

- You really can do all modern digital logic that we see around us with relays.
- Have some easy to use “modern” relay modules for you to mess with

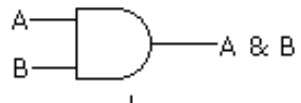


SPDT relay

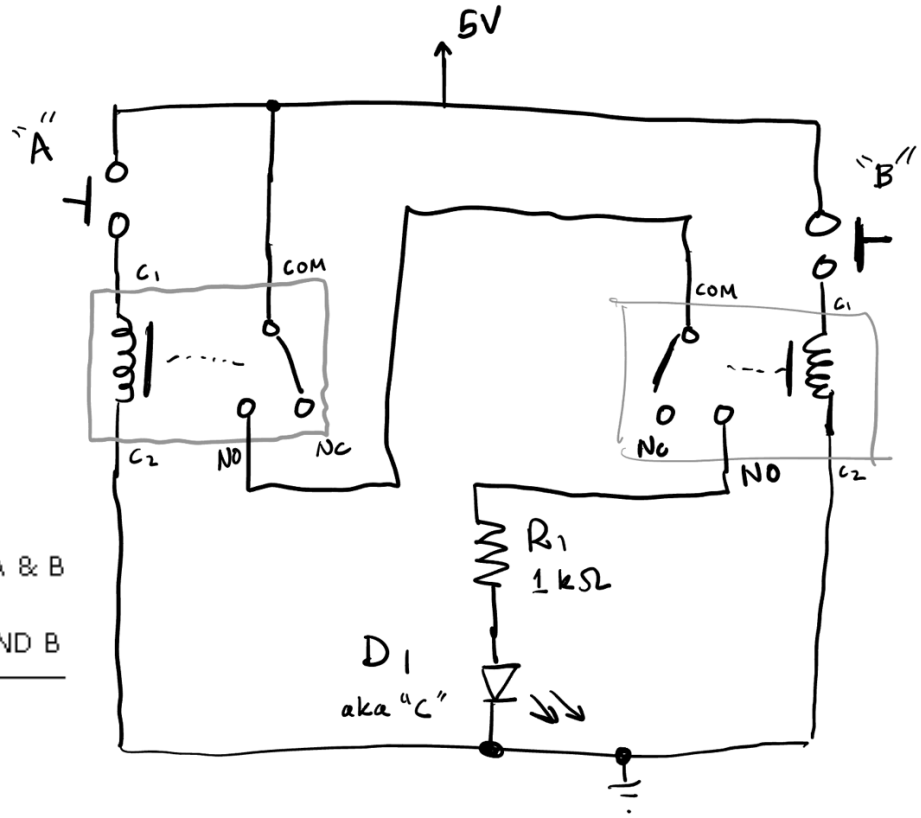


Consider this Circuit

- Inputs are A and B
- C = f(A,B). What is f?
  - f(0,0) = 0
  - f(0,1) = 0
  - f(1,0) = 0
  - f(1,1) = 1
- AND function!

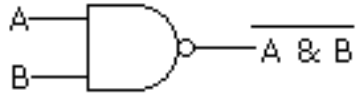


A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

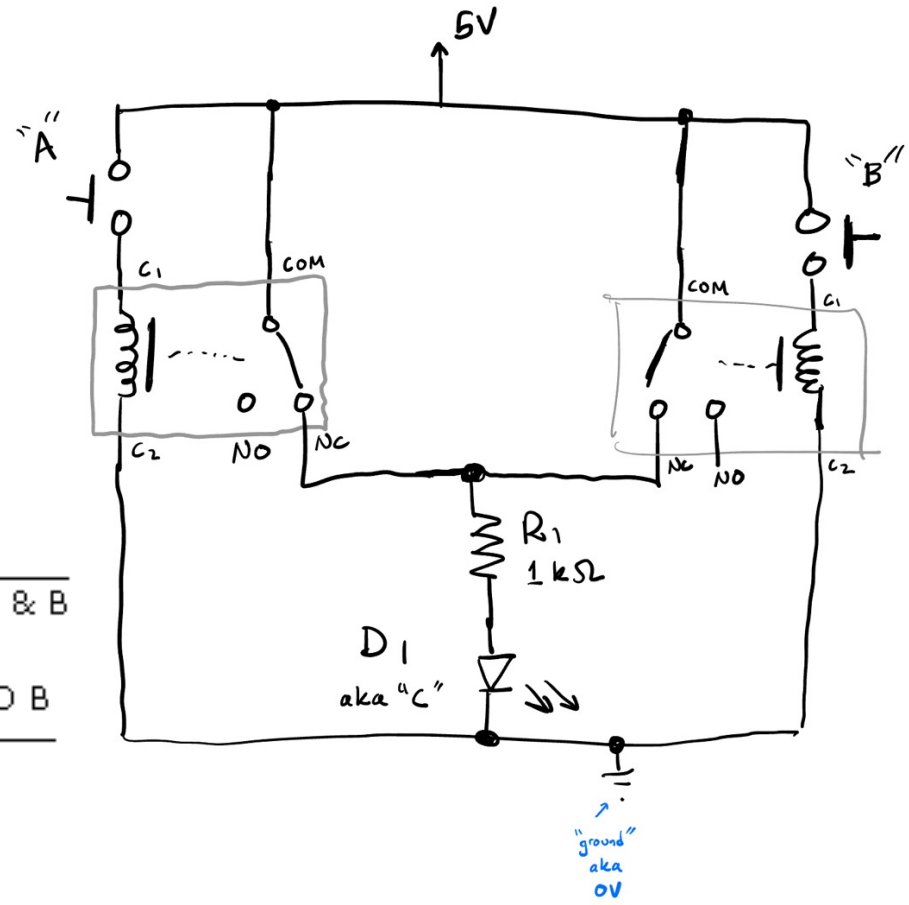


Now this one…

- Inputs are A and B
- C = f(A,B). What is f?
  - f(0,0) = 1
  - f(0,1) = 1
  - f(1,0) = 1
  - f(1,1) = 0
- Not-AND…I function!

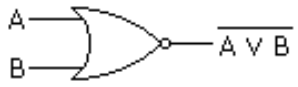


A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

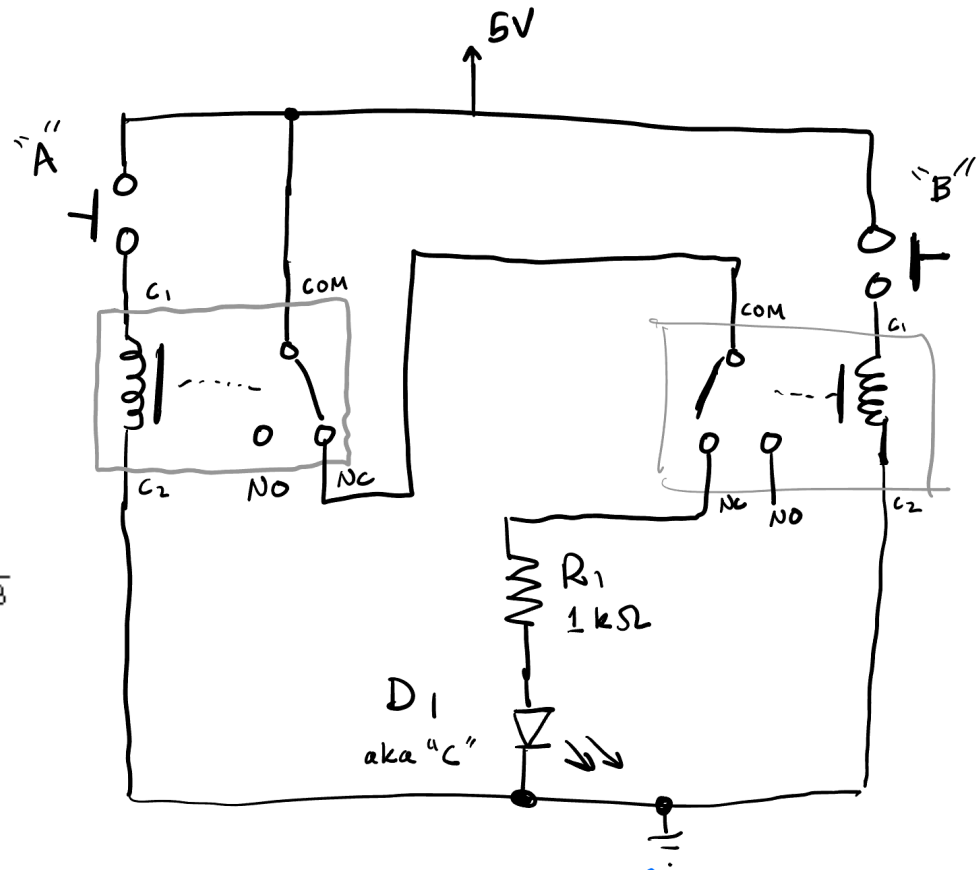


And this one…

- Inputs are A and B
- C = f(A,B). What is f?
  - f(0,0) = 1
  - f(0,1) = 0
  - f(1,0) = 0
  - f(1,1) = 0
- Not-OR…N

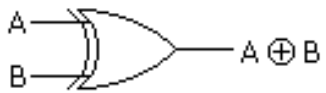


A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

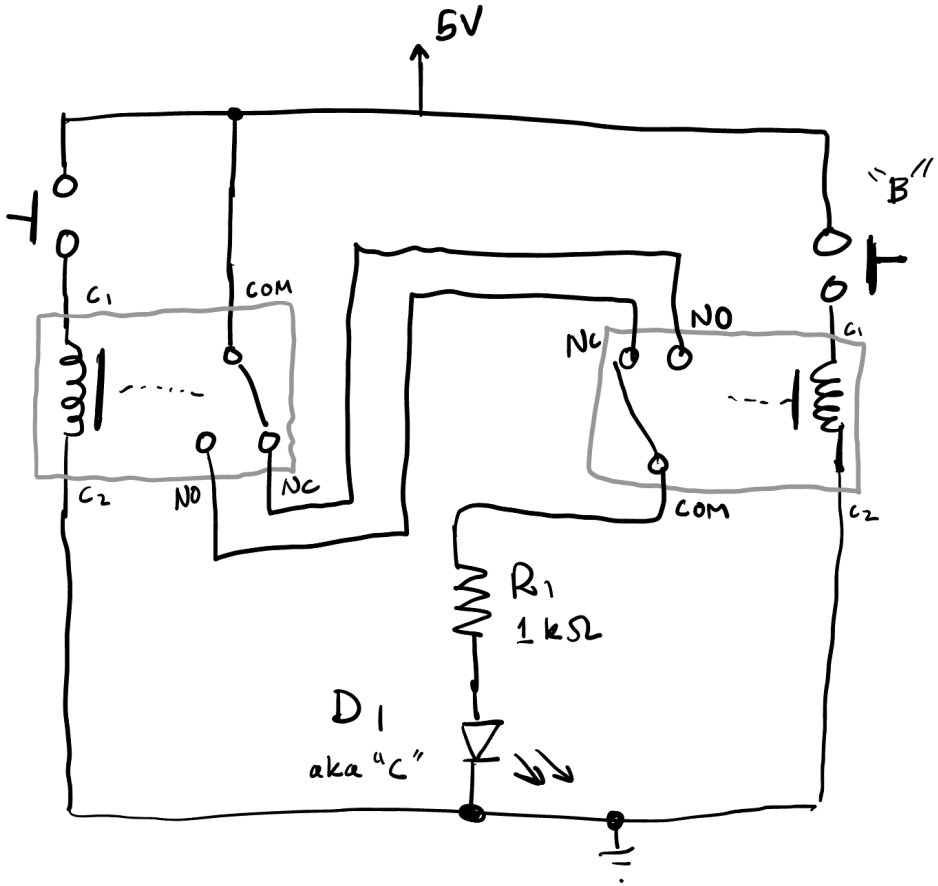


Now this nasty one…

- Inputs are A and B
- C = f(A,B). What is f?
  - f(0,0) = 0
  - f(0,1) = 1
  - f(1,0) = 1
  - f(1,1) = 0
- Either-Or-bu
  more…Exclu
  aka XOR

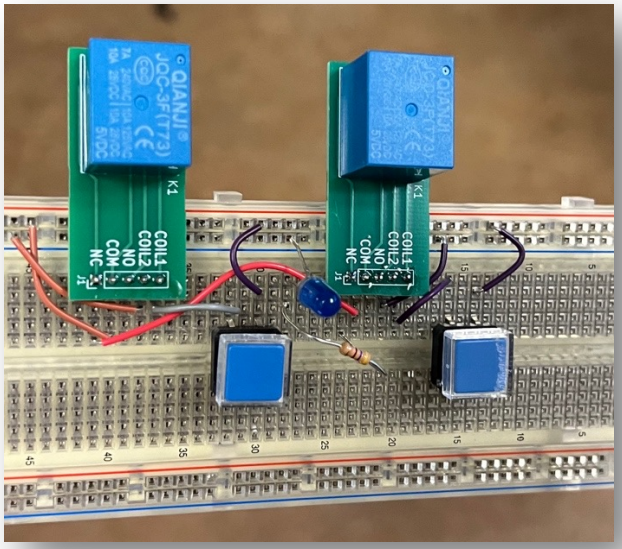


A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

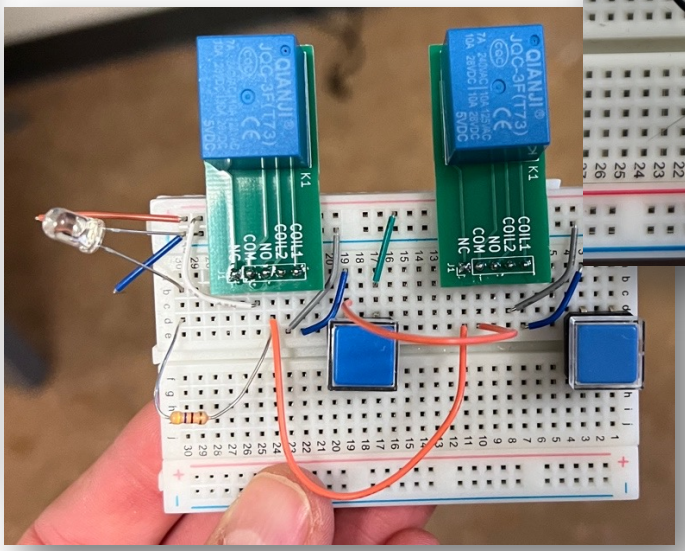


Lab 1A

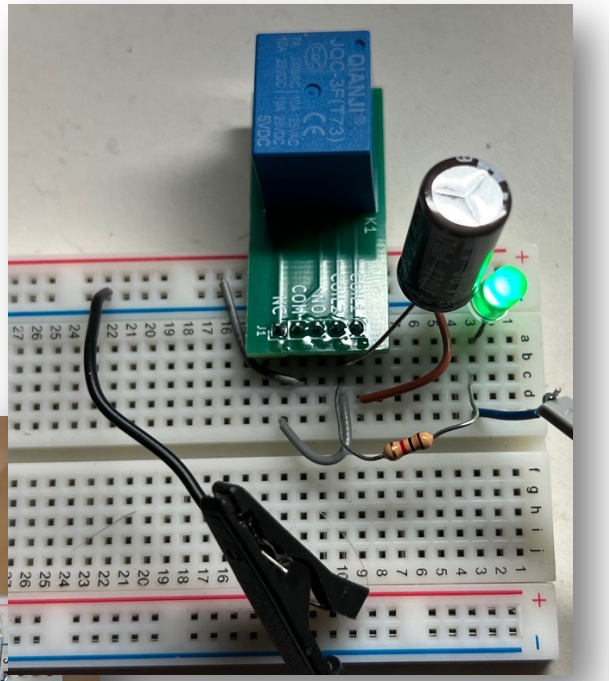
- Electromechanical Digital Logic



Relay-only NAND Gate



Relay-only SR Latch



Relay-only oscillator

Zuse computers

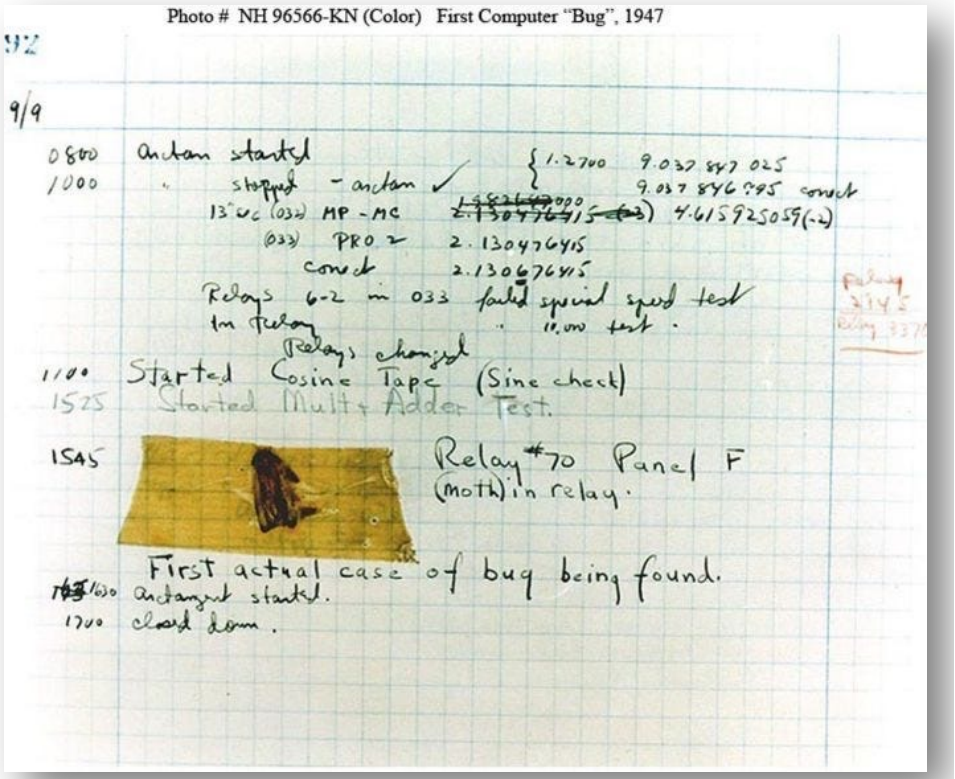
- German engineer who built fully-functioning computers with floating point operations using ONLY relays.

Zuse Z11

https://en.wikipedia.org/wiki/Z11_(computer)

First ”Bug”

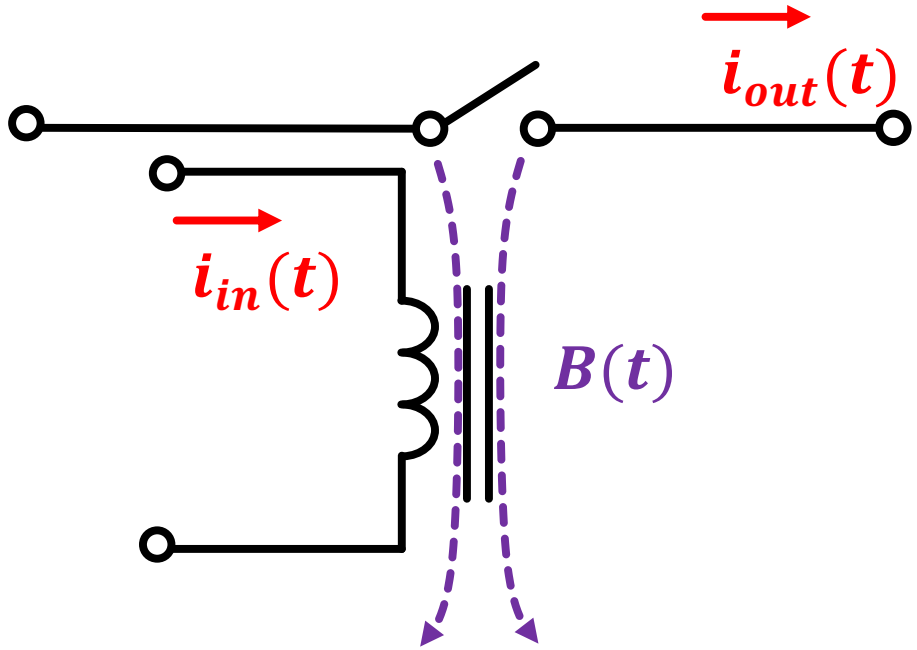
- A moth got crushed under a relay contact in Harvard’s Mark II relay-based computer
- Its guts prevented the switch from closing.
- Often claimed that this was Grace Hopper’s discovery
- Real story is more complicated



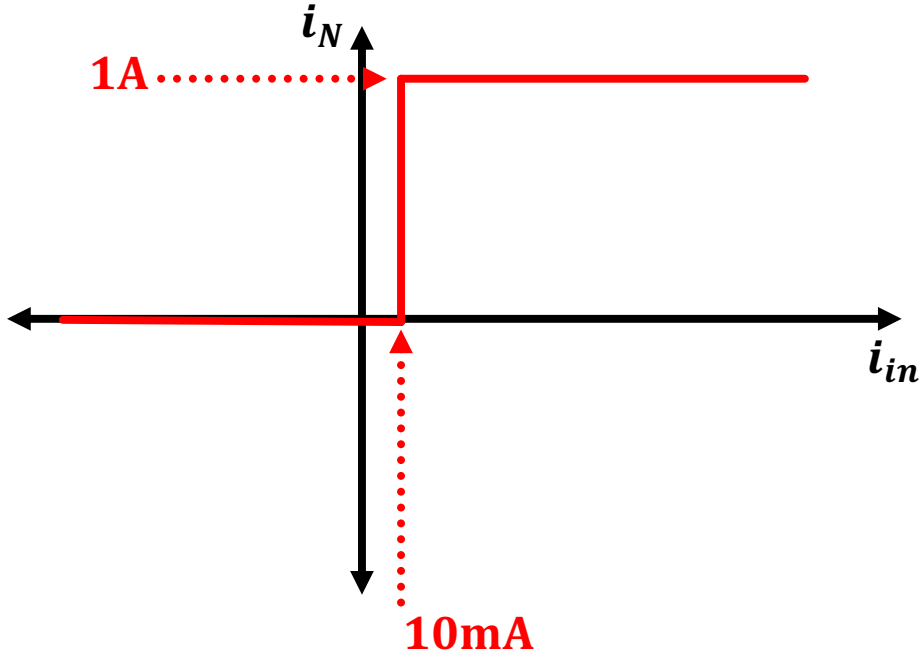
https://lunduke.substack.com/p/the-story-of-the-first-computer-bug

Relays Really Are the *Ideal* Digital Component

- It is *literally* an electrically-controlled switch.



What are some Characteristics of this I/O Relationship?



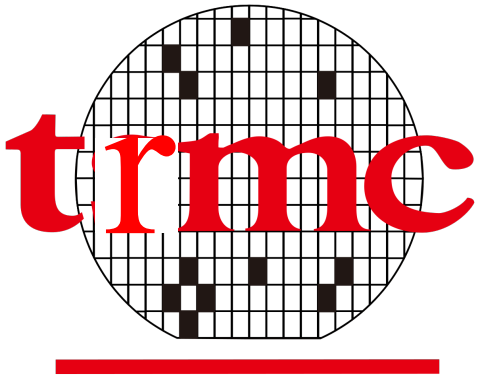
- Reliably and Reproducibly Nonlinear!
- Amplifies!

This is what gives us the Boolean Algebra in circuit form!

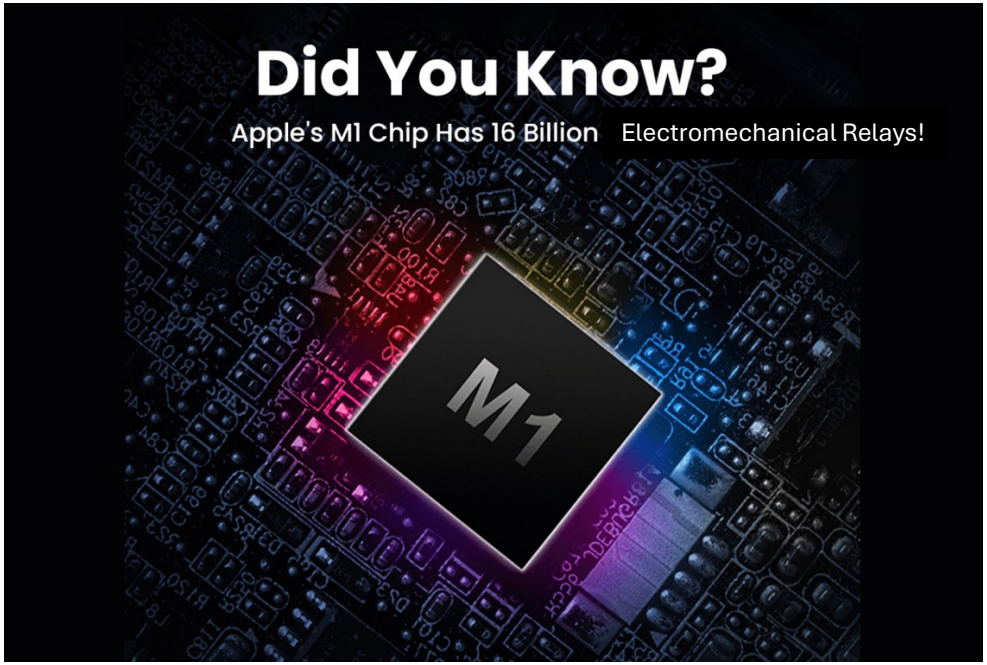
- It is essentially exactly what we want for digital logic.

And so that's how our story ends...

- The next eighty years of engineering was focused on making better and better electromechanical relays.



Taiwan Relay Manufacturing Corporation

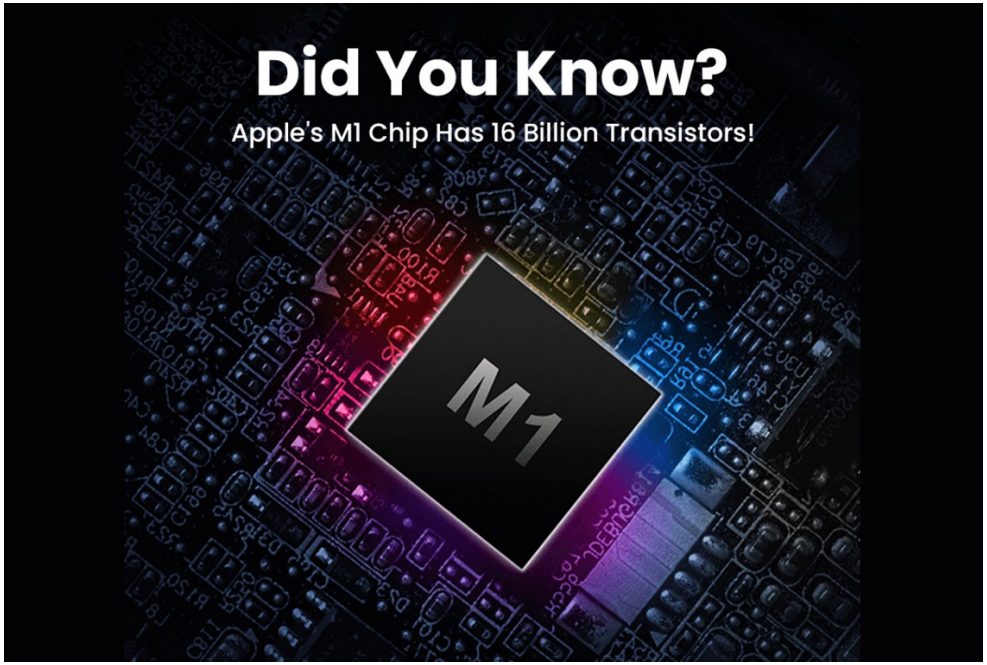


I also remember a couple years back when all the 6-1's got pissed that they removed a mandatory class in electromechanical relays from the curriculum.

JK JK JK JK…wrong timeline

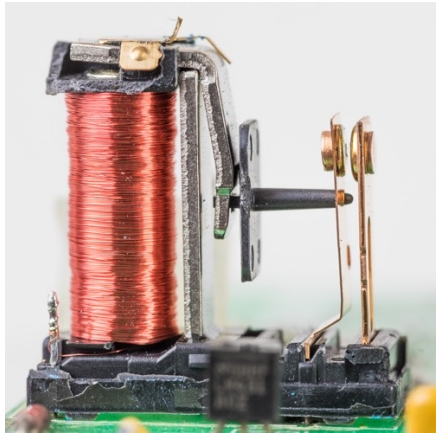
- The next eighty years of engineering was focused on making ~~better and better electromechanical relays.~~

Taiwan Semiconductor Manufacturing Corporation



I also remember a couple years back when all the 6-1's got pissed that they removed a mandatory class in transistors from the curriculum.

Problems with Relays?



- They are electro**mechanical** devices so they are pretty slow (never could operate at GHz…or MHz…and barely kHz)
- Relays can be made to be pretty robust, but still…Course 2 almost always lets us down eventually (mechanical contact wear in extreme cases)
- Use a lot of powers
- Very hard to scale down

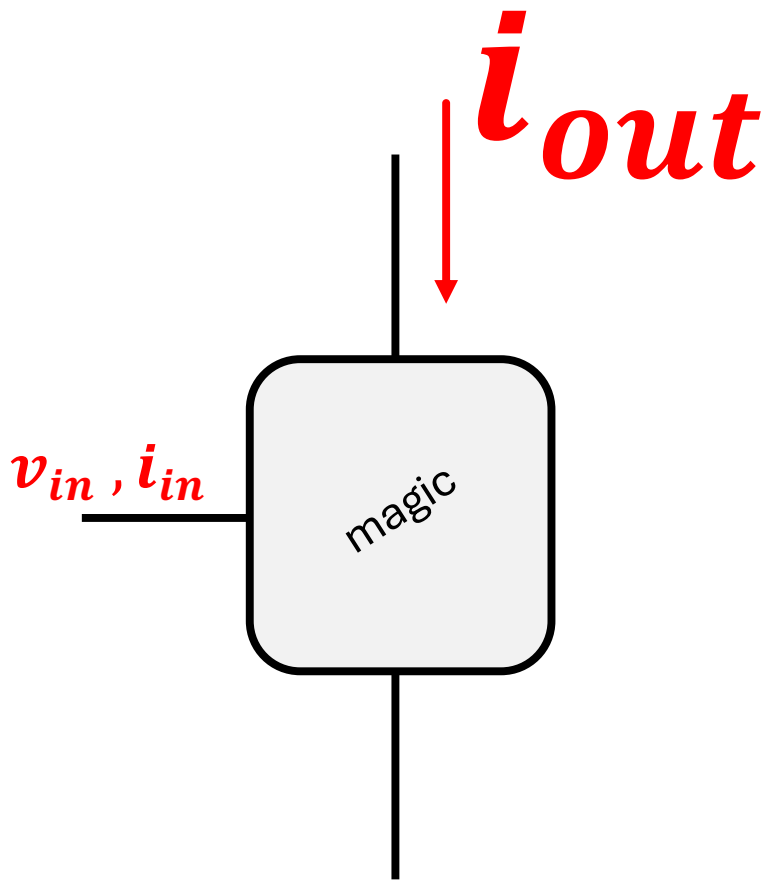
How to Fix?

- Everything in digital circuit design since Relays have really been less ideal*.

- Vacuum Tubes, Diodes, Transistors…
  - All of these technologies aren't actually electrically-controlled switches where the flow of electricity can turn on/off another flow…
  - They are analog devices that can be incorporated into circuits that approximately act that way within certain ranges of operation.

What was wanted…

- Some magical device that could quickly and cheaply and robustly allow a ***small*** electrical signal to modulate a ***BIG*** electrical signal
- Just like a relay but better



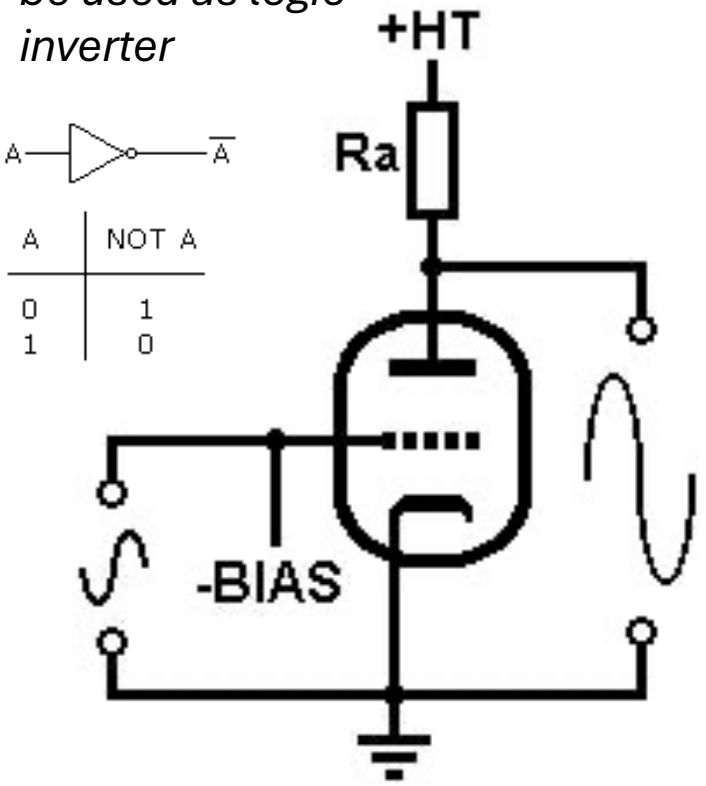
First Attempt: Vacuum Tubes

- Vacuum Tubes were invented around 1906 and were the first purely electronic amplifying device.
- Could be used to operate as electrically controlled switches
- Could be made to operate at higher speeds (MHz) ✅
- On a per-operation basis, were more robust and more power efficient than relays ✅

First Attempt: Vacuum Tubes

- Downsides:
  - Needed hundreds of volts to operate
  - Pretty delicate and could fail regularly
  - Couldn't really scale (easily)
- Also…vacuum tubes weren't actually electrically controlled switches…they were analog devices….they could only kind of act like that in certain situations. Had to build them into circuits that approximated switching behavior

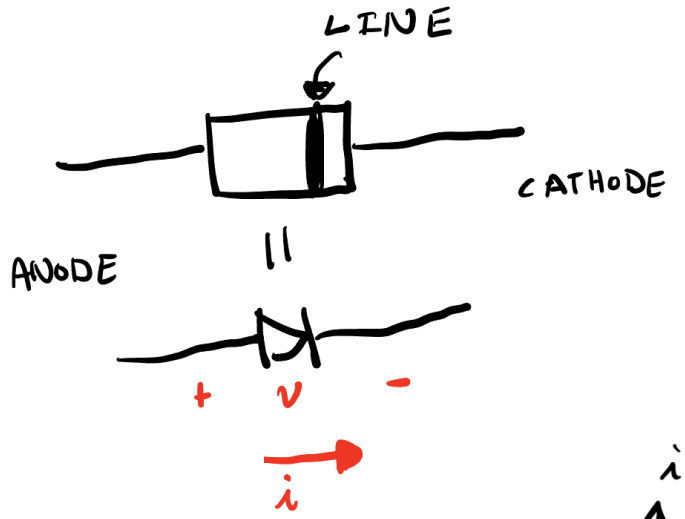
Tube amplifier could be used as logic inverter



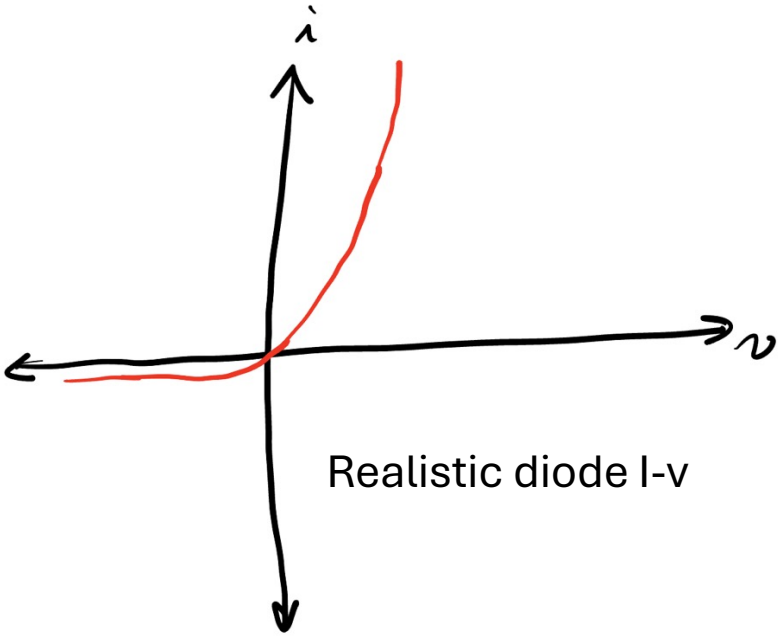
https://www.valvewizard.co.uk/gainstage.html

Second Attempt: Semiconductor Diodes

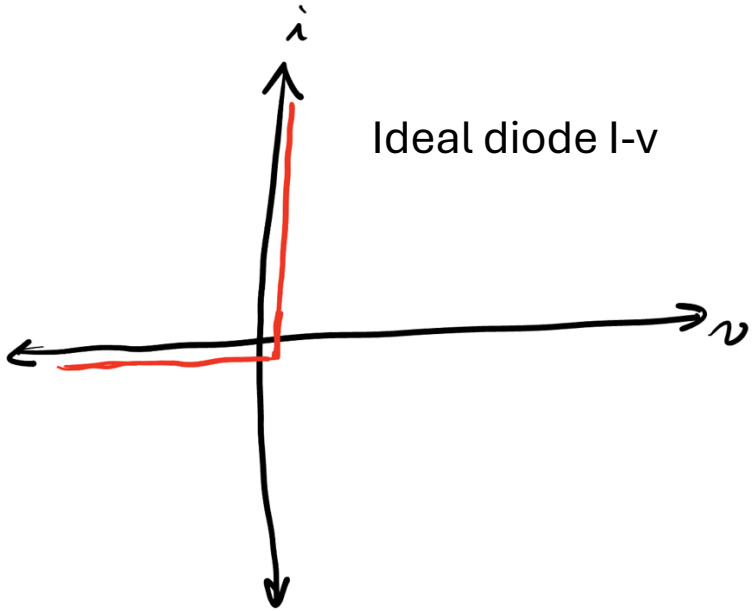
- Early passive semiconductor devices



One Way Electrical Valve



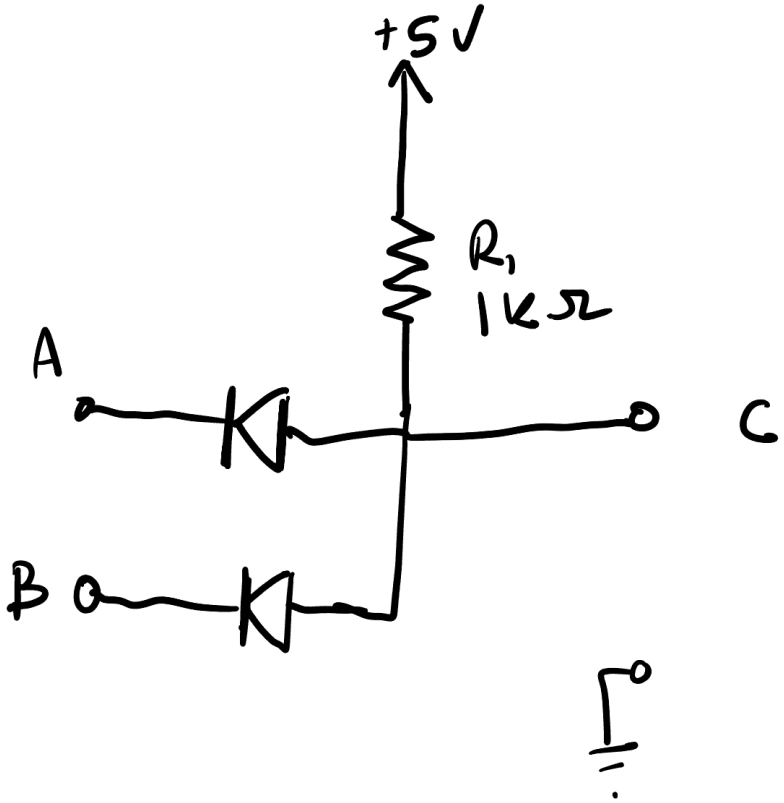
Realistic diode I-v



Ideal diode I-v

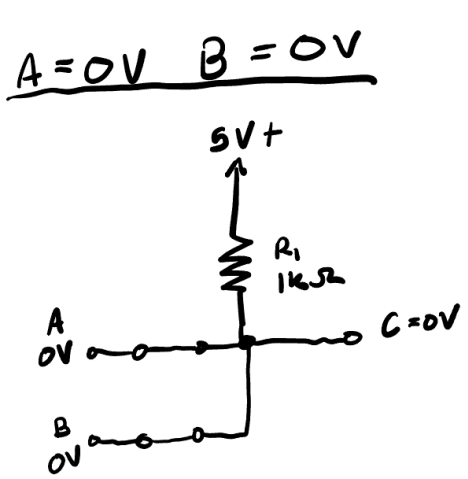
Second Attempt: Semiconductor Diodes

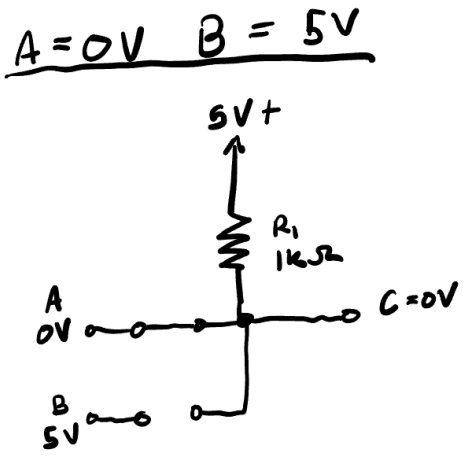
- Could incorporate them into circuits…

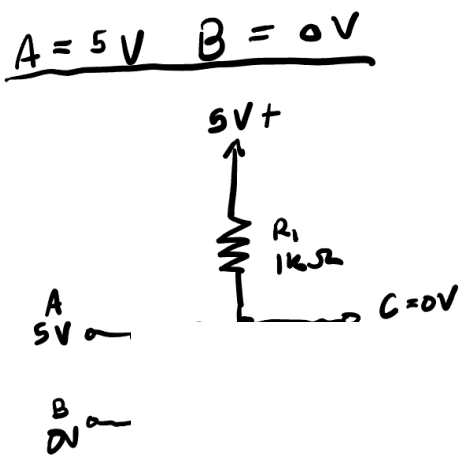


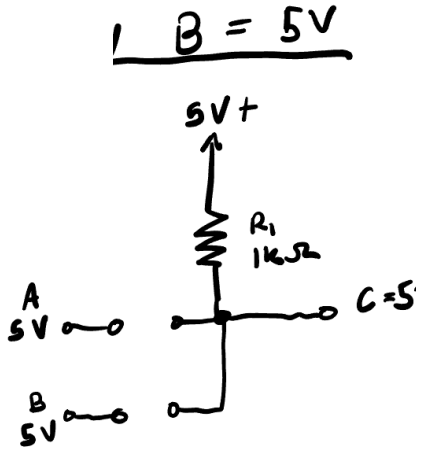
Second Attempt: Semiconductor Diodes

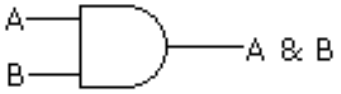
- Analyze using ideal diode behavior











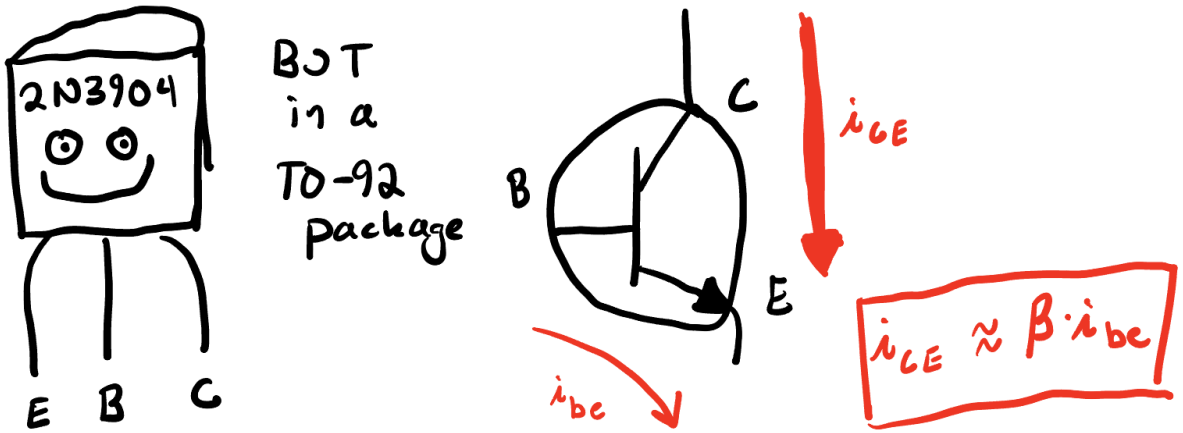
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Second Attempt: Semiconductor Diodes

- Benefits:
  - Small ✅
  - Low voltage ✅
  - Power efficient ✅
  - Fast ✅
- Bad things:
  - Could not amplify…so actually could not do logical inversion (NAND, NOR, NOT)…so…as we'll see…couldn't do all logic. ❌

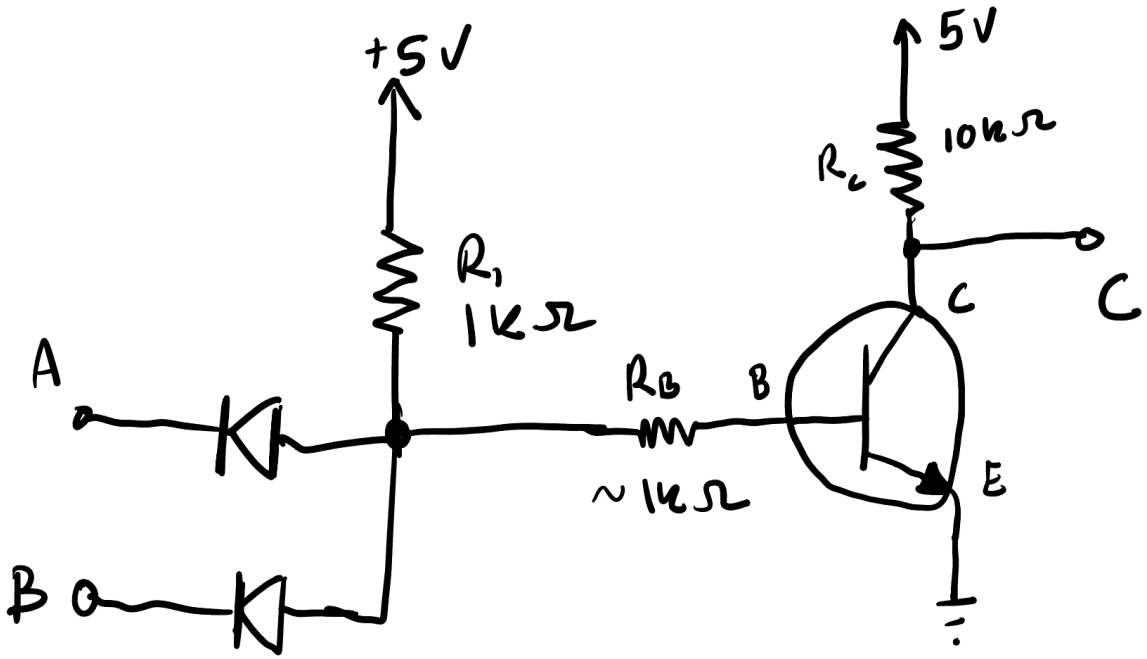
Third Attempt: Transistors (1947)

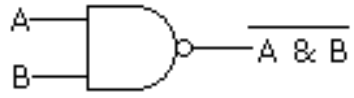
- Like a low-voltage, fast, scalable vacuum tube.
- Transistors on their own were not digital.
- You had to build them into special circuits that could be used in a digital way.



Different Flavors of Transistor Design

- First there was Diode-Transistor Logic

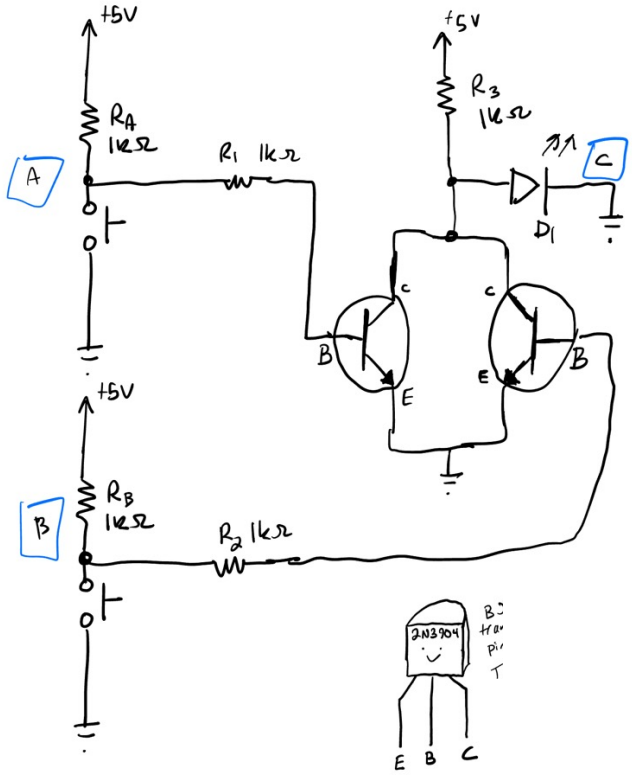


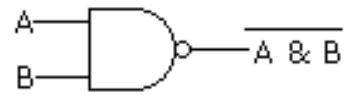


A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

Different Flavors of Transistor Design

- Then there was Resistor-Transistor Logic

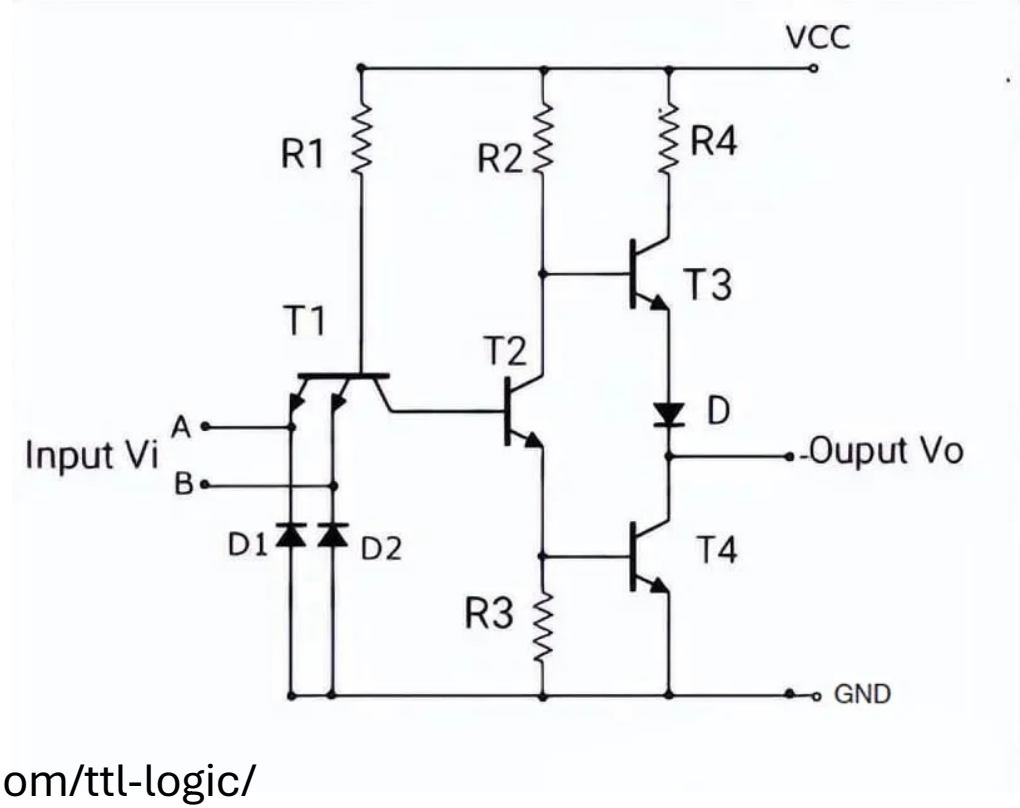




A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

Different Flavors of Transistor Design

- Then there was Transistor-Transistor Logic



https://reversepcb.com/ttl-logic/

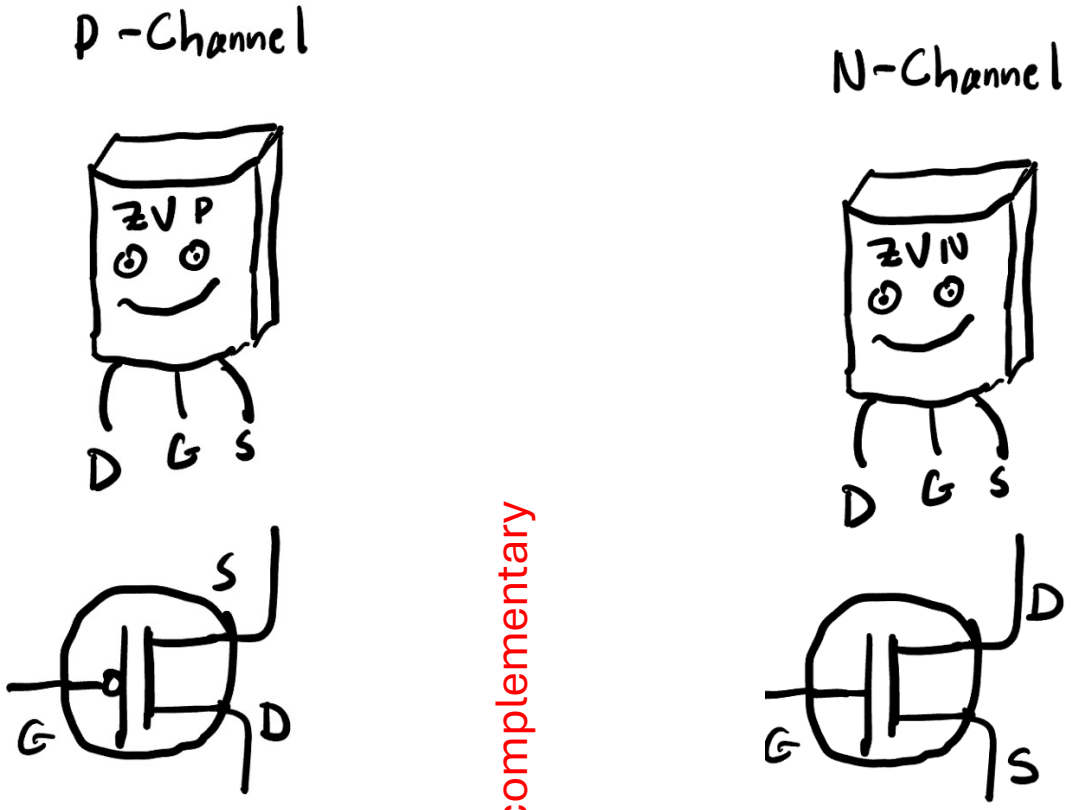
Different Flavors of Transistor Design

- Then there was ***Complementary***-Metal Oxide Semiconductor Circuits (CMOS)

- This was a new thing with transistors! You could have transistors that worked in opposite ways!!!

P and N channel devices

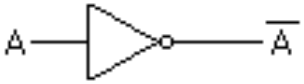
Voltage at Gate (G)…modulates D-S current



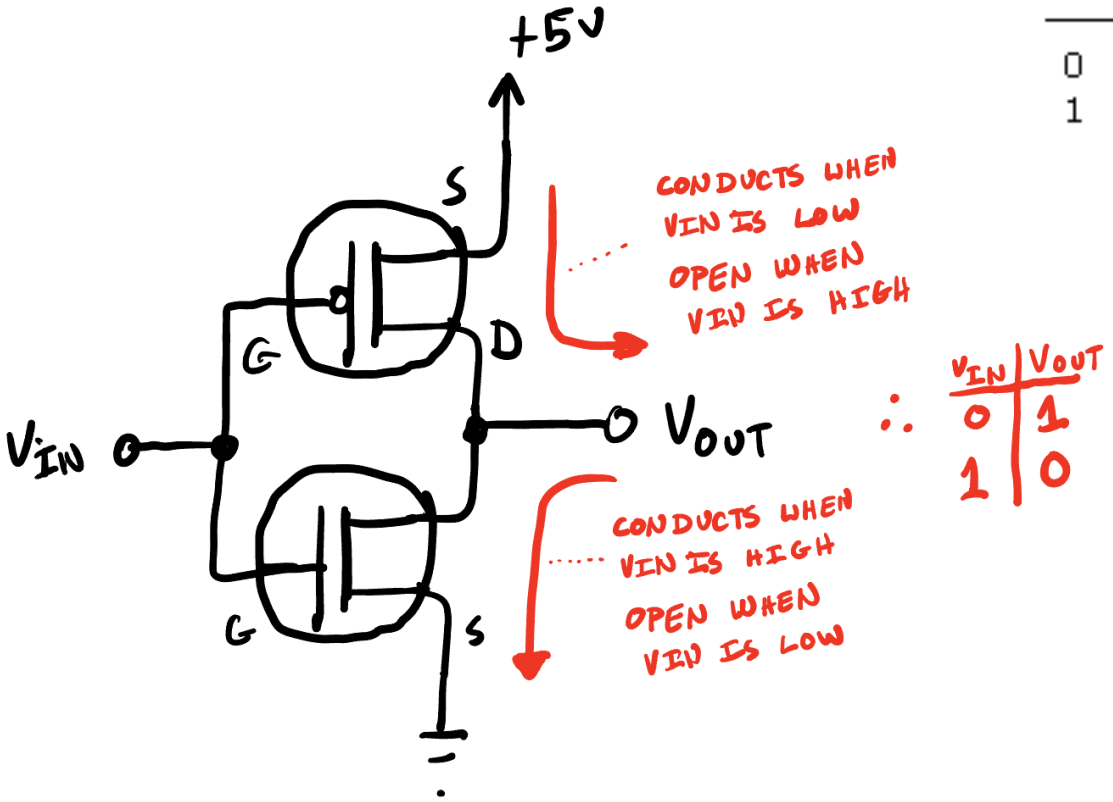
When v_G High, SD no connected
When v_G Low, SD connected

When v_G High, SD connected
When v_G Low, SD no connected

CMOS circuits (Inverter)



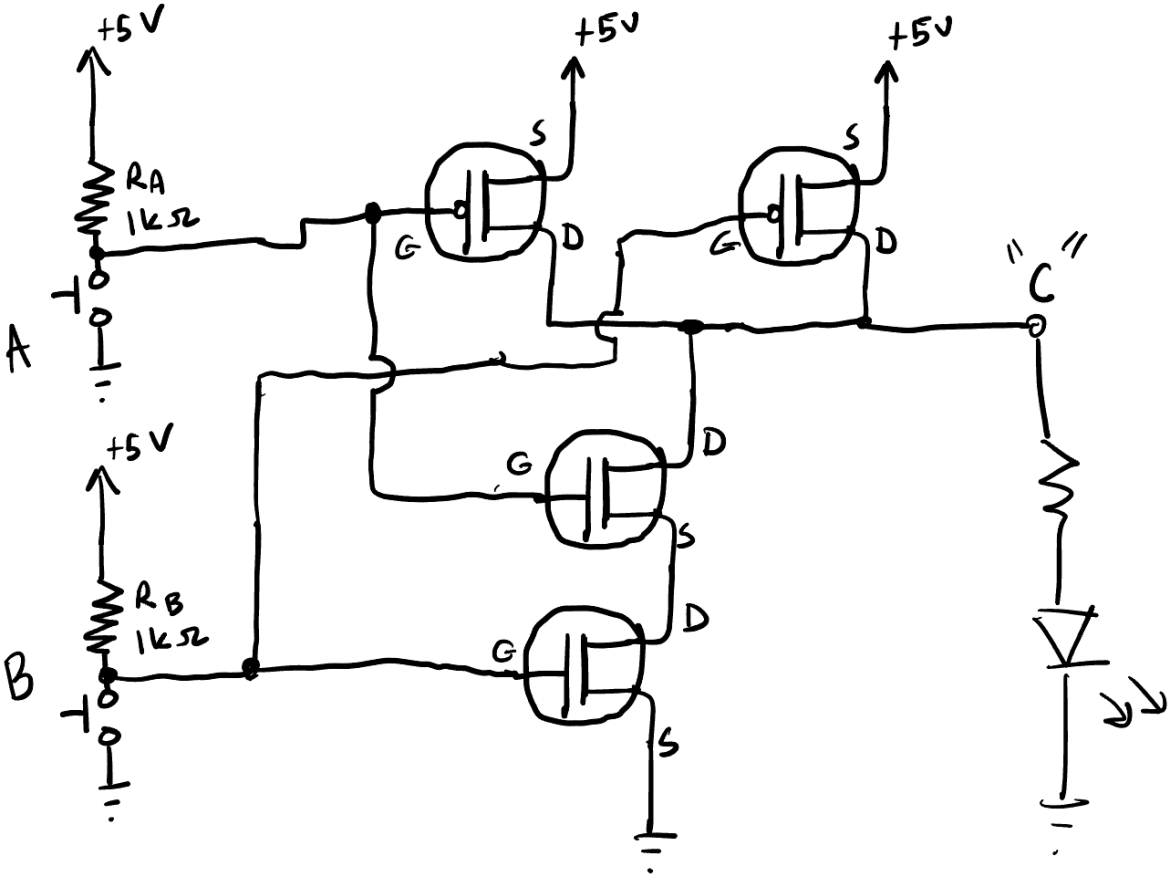
A	NOT A
0	1
1	0

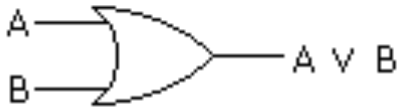


∴

V_{IN}	V_{OUT}
0	1
1	0

CMOS Gate





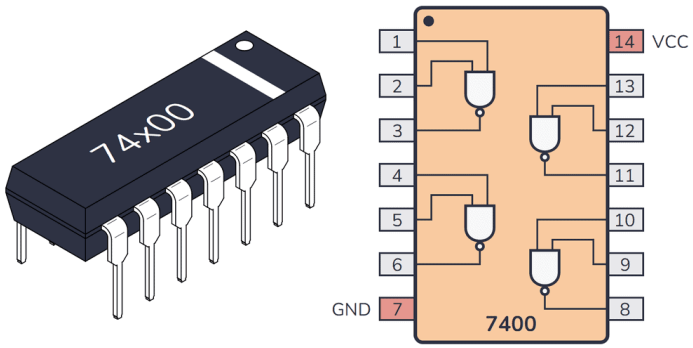
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Transistor (CMOS) Benefits

- Benefits:
  - Low voltage
  - Very Power efficient
  - Fast
  - ***Could be scaled down in size! (exponentially so)***
- Downsides:
  - On their own, they are not digital devices. We have to incorporate them into specialized circuits.
  - Always fighting against their inherent analog nature.

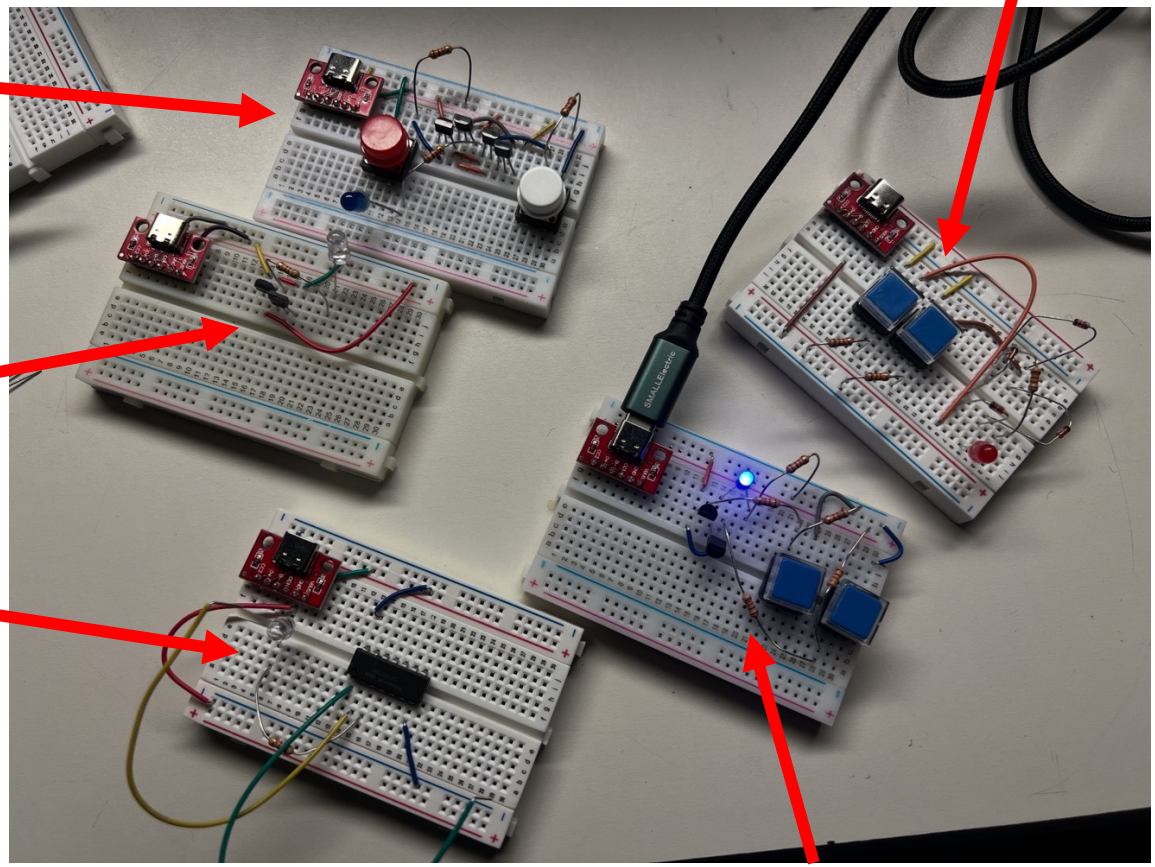
Scaling…

- Transistors could be made so small that by the 1960s they could start to merge multiple transistors into single devices and package them together.
- Didn’t have to build the gates anymore…you could just buy them and stick them together like lego bricks



Lab 1B

- Electrical-only Logic Gates



End

- That’s where we’ll end today
- Next time we’ll start designing with these logical elements