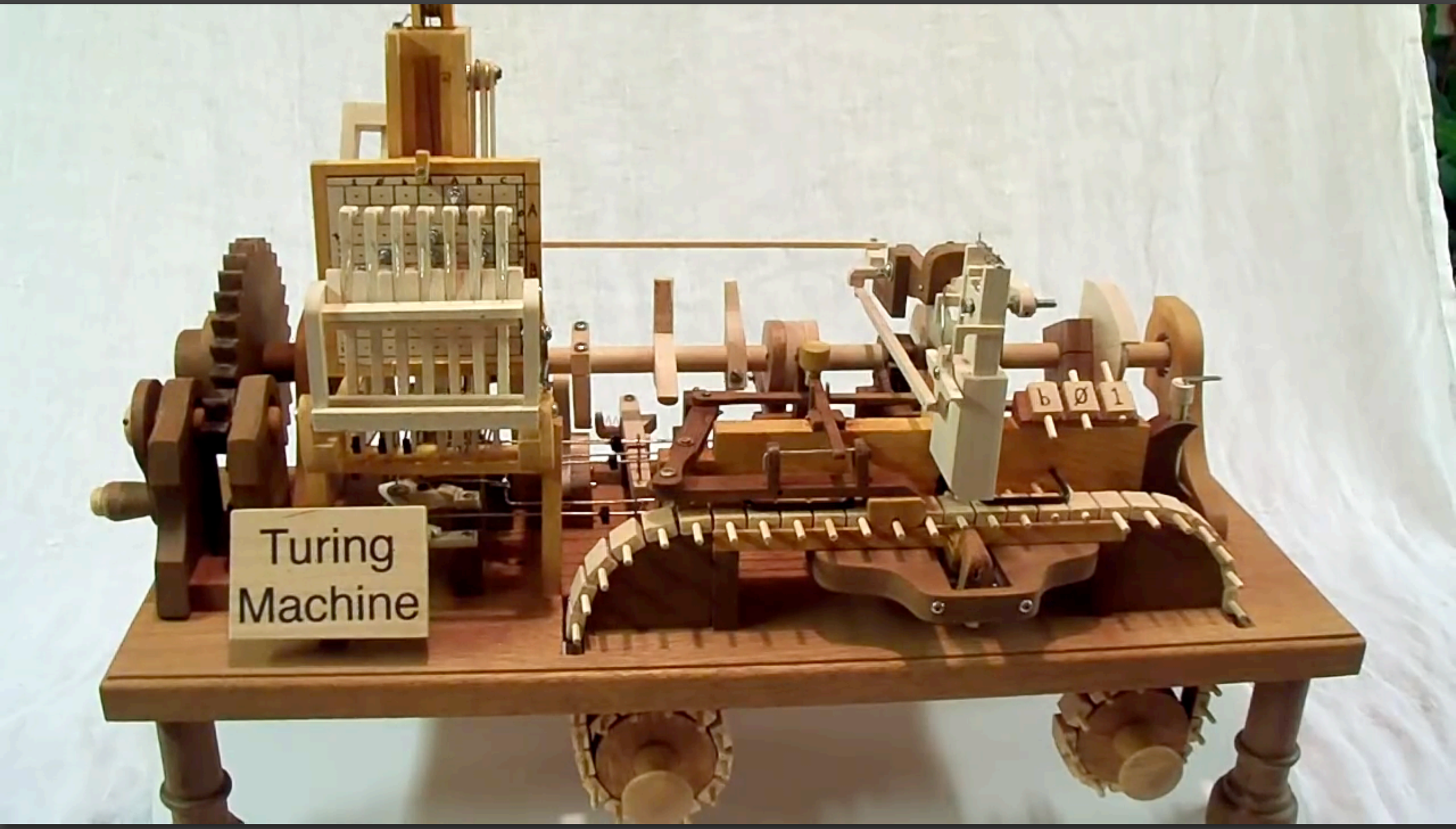
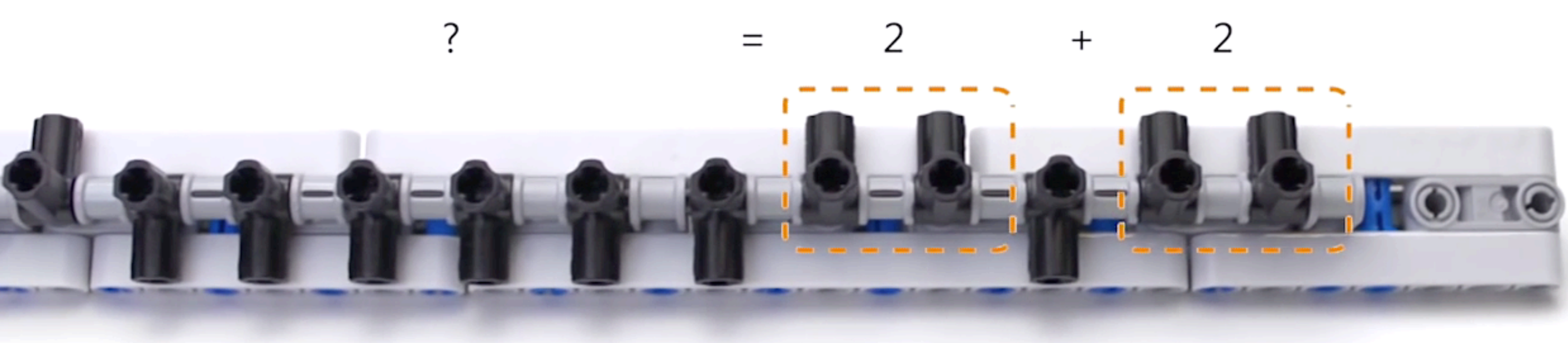


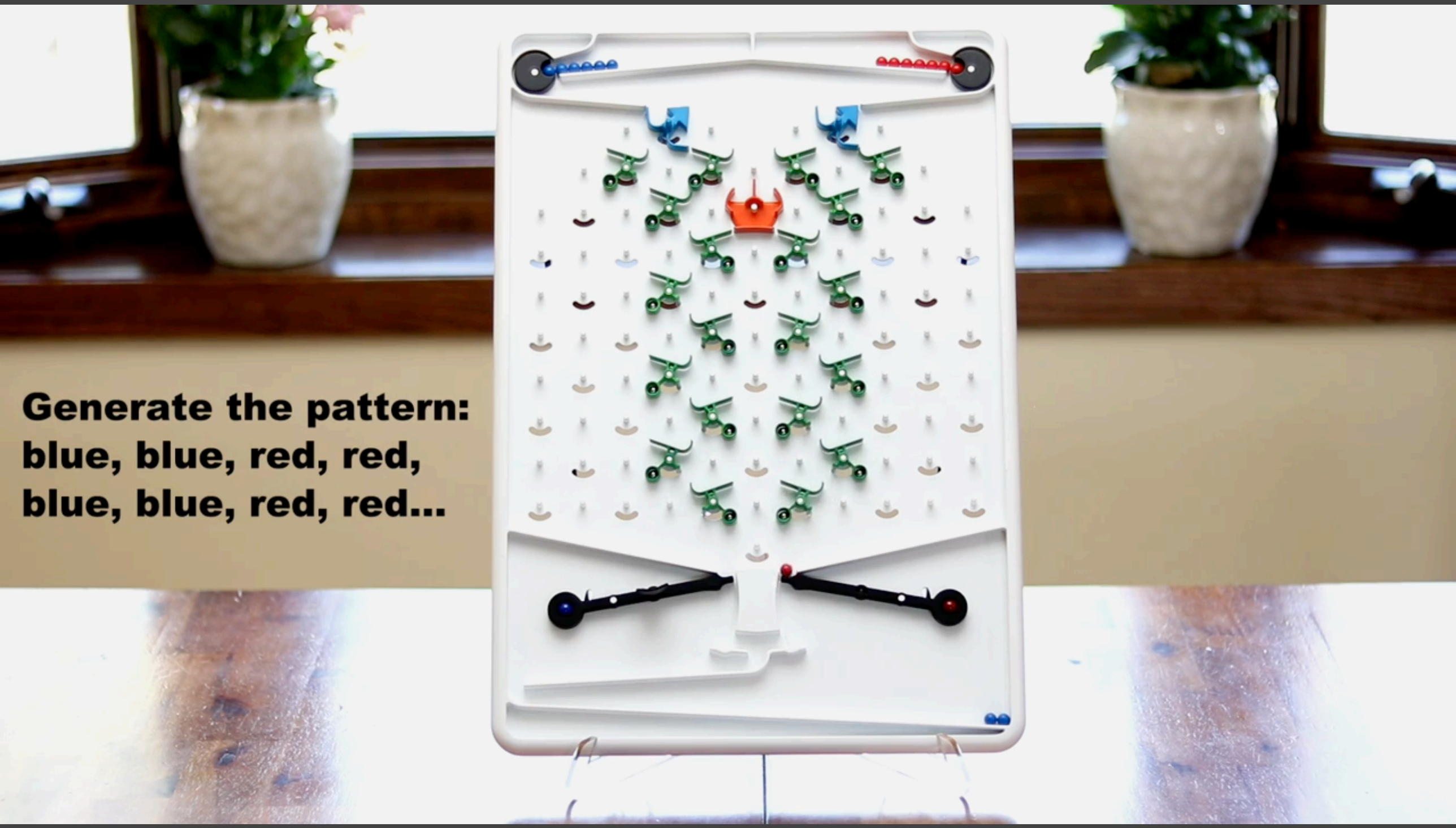
Mechanical Turing Machine in Wood
R. Ridel



LEGO Turing Machine

Built by J. van den Bos & D. Landman

Video by A. Theelen



Turing Tumble

Come visit the one in my office!

What counts as a problem?

Decision problems on finite, bitstring inputs.

What kinds of **problems** can **computers** solve?

Turing Machines can solve more problems than DFAs!
(But not all decision problems)

What counts as a computer?

Programs ≡ Data

How malware finally infected Apple iOS apps: XCodeGhost

Hackers can't easily get malware directly in iOS apps so they're taking a different approach: Modifying the programming environment that Apple provides to make apps.

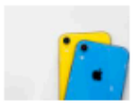
By Kevin Tofel for Mobile Platforms | September 20, 2015 -- 14:41 GMT (07:41 PDT) | Topic: Apple

Google's open approach to Android apps have led to several malware issues over the past few years, while Apple users have remained relatively unscathed. Not any more. At least 39 apps installed by several hundred million iPad and iPhone owners include malware, acccording to Forbes.

The initial issue report came from Palo Alto Networks last week, noting that hackers took a unique approach to inject malware into iOS apps.

Since it's a challenge to get malware past Apple's App Store review team, hackers took a more indirect way: By adding the rogue code into what app developers thought was the official Apple version of XCode, Apple's IDE for creating iOS and Mac OS X apps.

RELATED STORIES



Security
Nasty piece of CSS code crashes and restarts iPhones

Mobility
iPhone XS mocked by Huawei for

People have always computed

八卦 Bāguà—The eight trigrams

乾 Qián	兌 Duì	離 Lí	震 Zhèn	巽 Xùn	坎 Kǎn	艮 Gèn	坤 Kūn
Heaven/Sky	Lake/Marsh	Fire	Thunder	Wind	Water	Mountain	Earth
天 Tiān	澤(泽) Zé	火 Huǒ	雷 Léi	風(风) Fēng	水 Shuǐ	山 Shān	地 Dì

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

https://en.wikipedia.org/wiki/Abacus#/media/File:Boulier1.JPG

https://en.wikipedia.org/wiki/Napier%27s_bones#/media/File:An_18th_century_set_of_Napier%27s_Bones.JPG

88 MEMOIRES DE L'ACADEMIE ROYALE

res Lineaires qu'on lui attribue. Elles reviennent toutes à cette Arithmétique; mais il ſuffit de mettre ici *la Figure de huit Cova* comme on l'appelle, qui paſſe pour fondamentale, & d'y joindre l'explication qui eſt manifeſte, pourvû qu'on remarque premierement qu'une ligne entiere ——— ſignifie l'unité ou 1, & ſecondement qu'une ligne briſée — — ſignifie le zero ou 0.

000 001 010 011 100 101 110 111

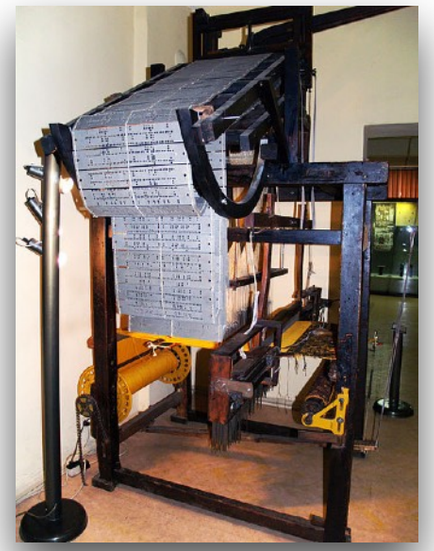
0 1 10 11 100 101 110 111

0 1 2 3 4 5 6 7

Les Chinois ont perdu la ſignification des *Cova* ou Linéations de Fohy, peut-être depuis plus d'un millenaire d'année; & ils ont fait des Commentaires là-deſſus, où ils ont cherché je ne ſçai quels ſens éloignés. De ſorte qu'il a fallu que la vraie explication leur vînt maintenant des Européens: voici comment.

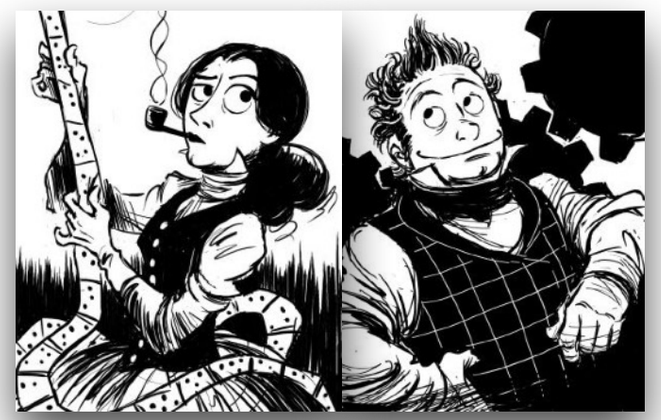
https://hal.archives-ouvertes.fr/ads-00104781

https://en.wikipedia.org/wiki/Ishango_bone#/media/File:Os_d%27Ishango_IRSNB.JPG



https://en.wikipedia.org/wiki/File:Hand-driven-jacquard-loom.jpg

https://en.wikipedia.org/wiki/Computer#/media/File:NAMA_Machine_d%27Anticyth%C3%A8re_1.jpg



http://sydneypadua.com/2dgoggles/cast/

https://www.nasa.gov/sites/default/files/thumbnails/image/youngdorothyvaughan.jpeg

Levels of abstraction

Stored-program computers

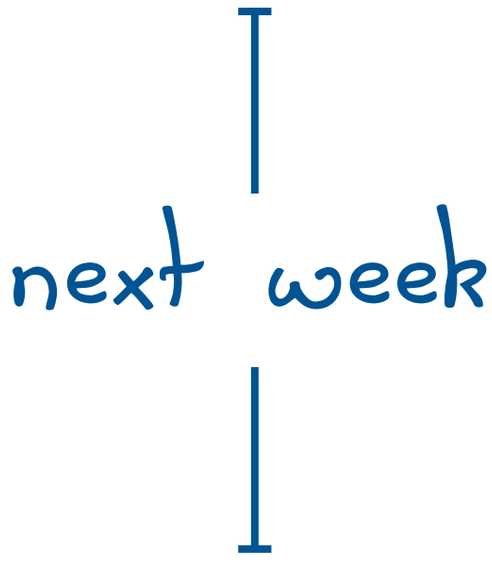
Random-access memory (RAM)

Registers

1-bit memory: latches

Logic gates

Transistors / switches



Boolean functions

A **boolean function** is a function that takes n bits as input and returns 1 bit of output.

This **truth table** defines a boolean function.

input			output
x	y	z	
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

What does the boolean function do?

Can you describe its purpose simply, so that another person could understand?

Firstname Lastname T. 9 / 18

(Your response)

We can also extend the definition of boolean functions so that they return multiple bits of output.

Boolean functions

A **boolean function** is a function that takes n bits as input and returns 1 bit of output.

This **truth table** defines a boolean function.

input			output
x	y	z	
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

What does the boolean function do?

Can you describe its purpose simply, so that another person could understand?

Firstname Lastname T. 9 / 18

Odd parity: Does the input have an odd number of bits whose value is 1?

We can also extend the definition of boolean functions so that they return multiple bits of output.

More boolean functions

input		output
x	y	
0	0	0
0	1	0
1	0	0
1	1	1

input		output
x	y	
0	0	0
0	1	1
1	0	1
1	1	1

input	output
x	
0	1
1	0

Boolean operations as **gates**

*the building blocks of **combinational logic***

AND (∧)

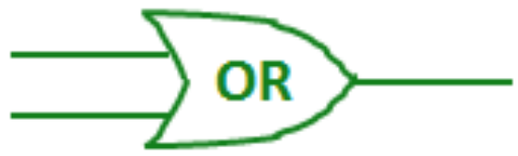
input		output
x	y	
0	0	0
0	1	0
1	0	0
1	1	1

outputs 1 if
all inputs are 1

OR (∨)

input		output
x	y	
0	0	0
0	1	1
1	0	1
1	1	1

outputs 1 if
any inputs is 1



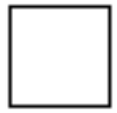
AND & OR gates can also take more than two inputs.

NOT (¬)

input	output
x	
0	1
1	0

inverts its input

Great! How can we build these gates?



Air
(Null)

Switch
(Input)

Block
(Generic)

Torch
(Side of Block)

Torch
(On Ground)

Torch
(Top of Block)

Redstone
(On Ground)

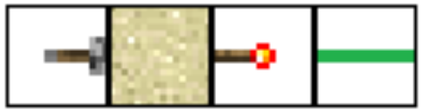
Redstone
(Top of Block)

Redstone
(Output)

Redstone
(Inverse Output)

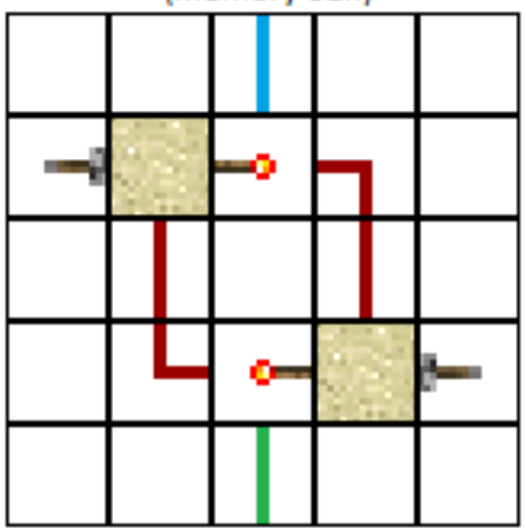
Input/Output Gate

NOT Gate
(Inverter)

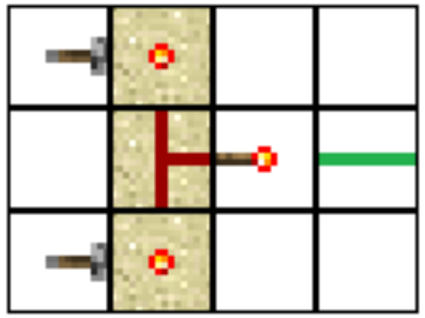


Double NOT
(Repeater)

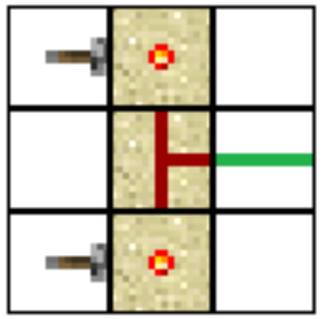
RS NOR Latch
(Memory Cell)



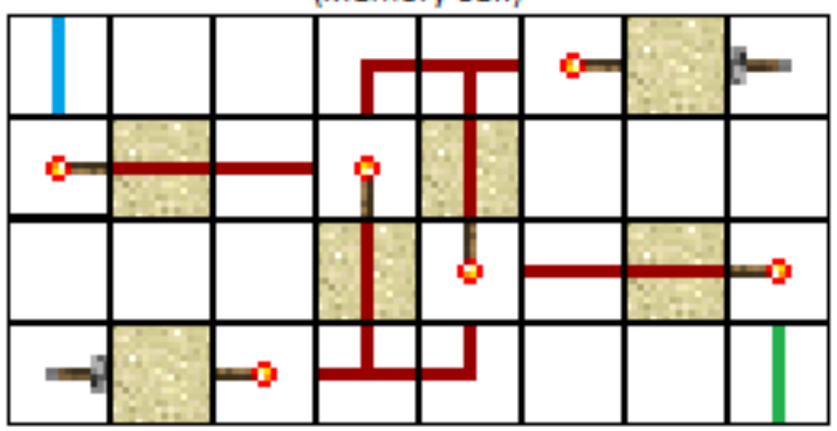
AND Gate



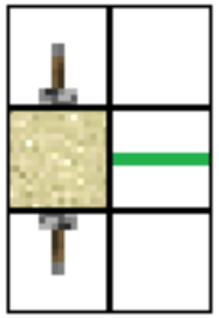
NAND Gate



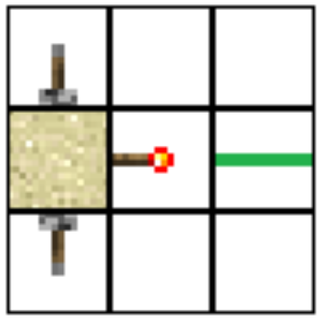
RS NAND Latch
(Memory Cell)



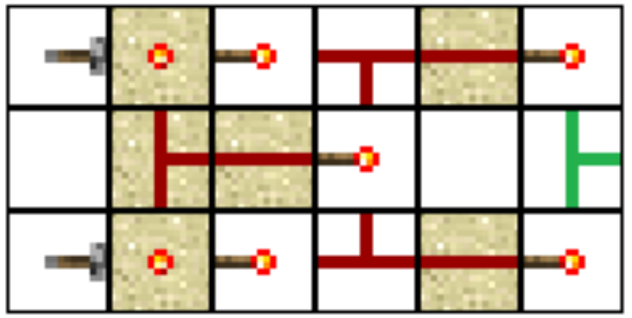
OR Gate



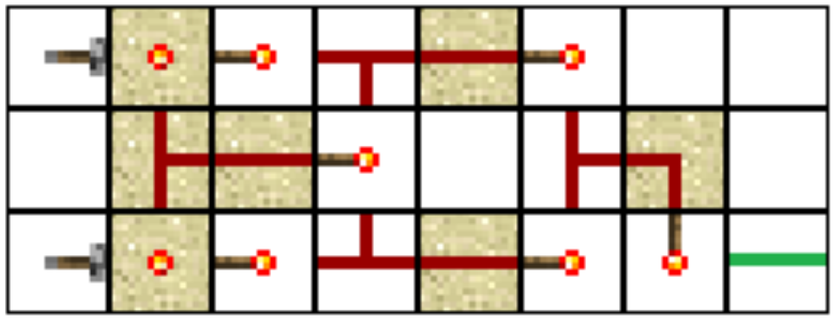
NOR Gate



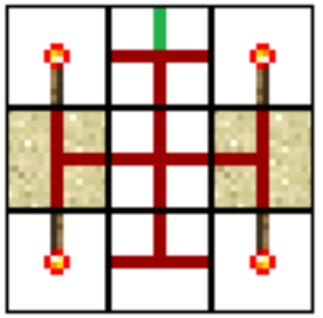
XOR Gate



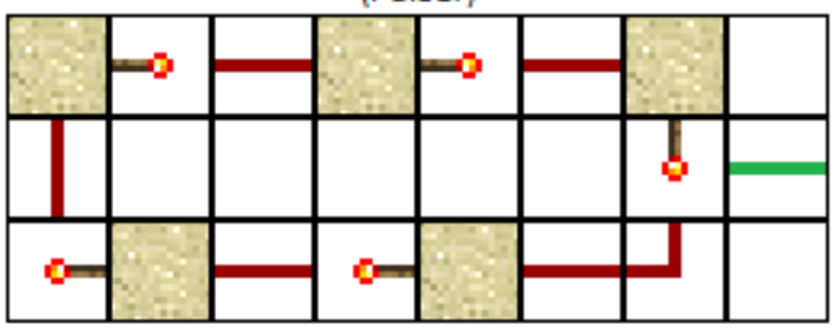
XNOR Gate



Rapid Pulser

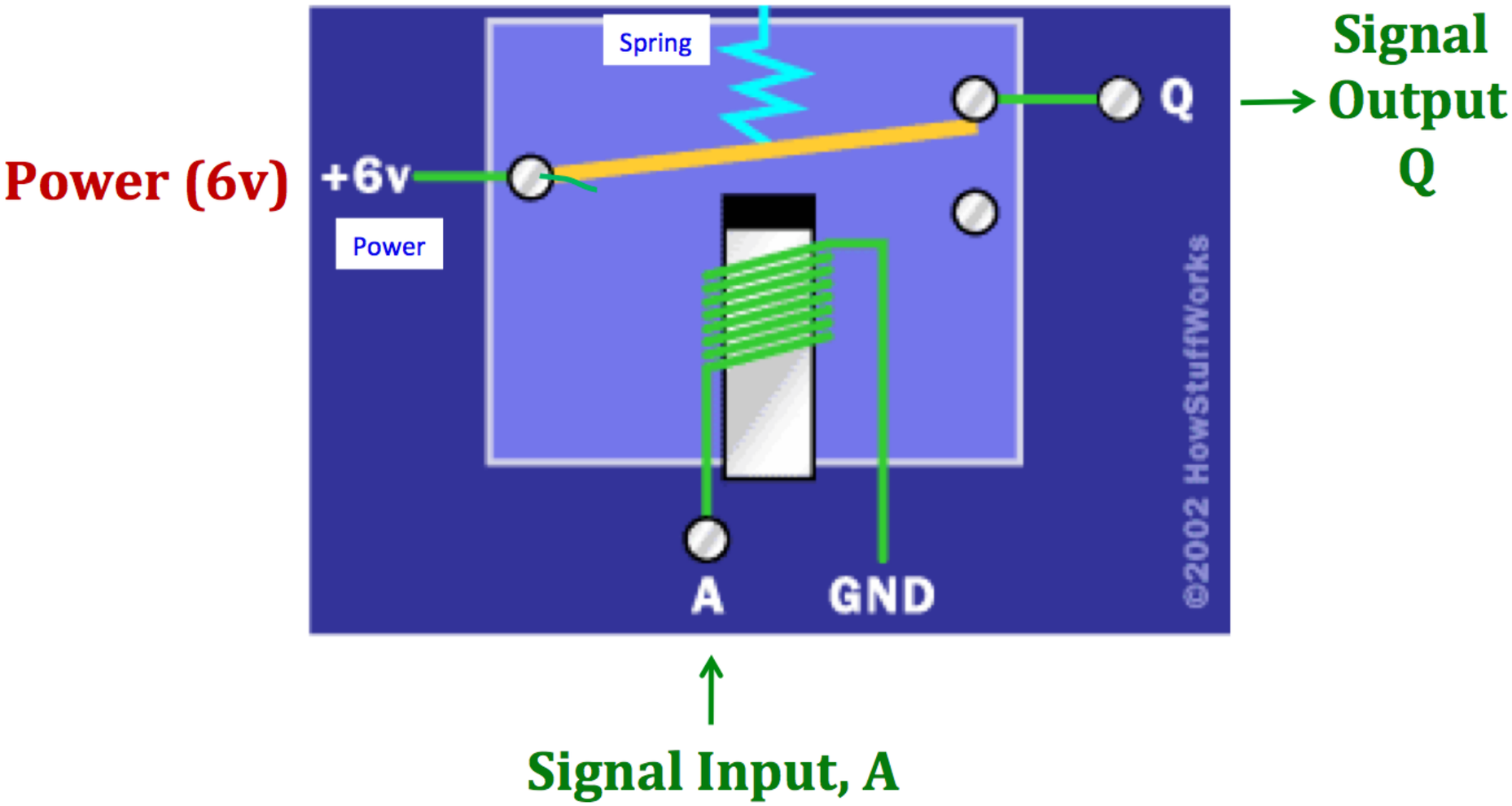


5-Clock
(Pulser)

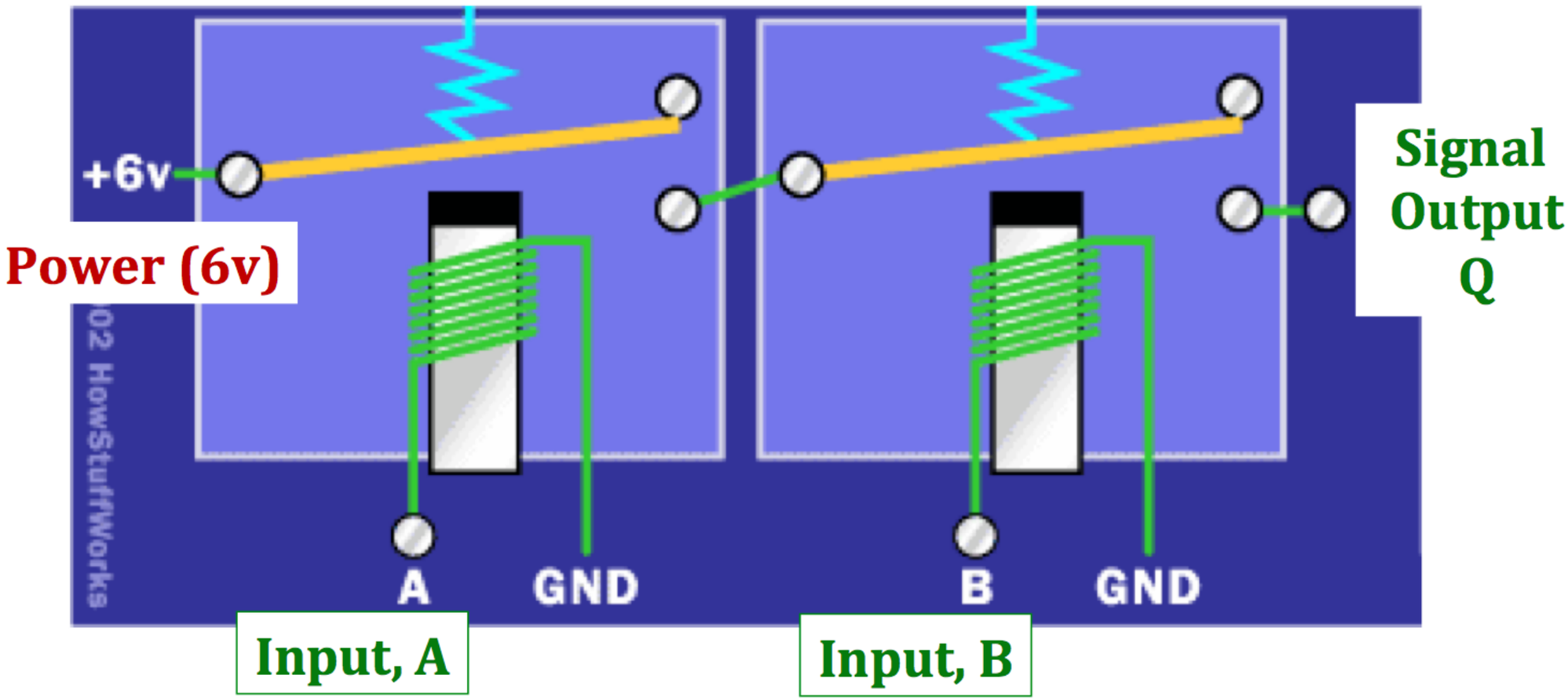


Mechanical relays

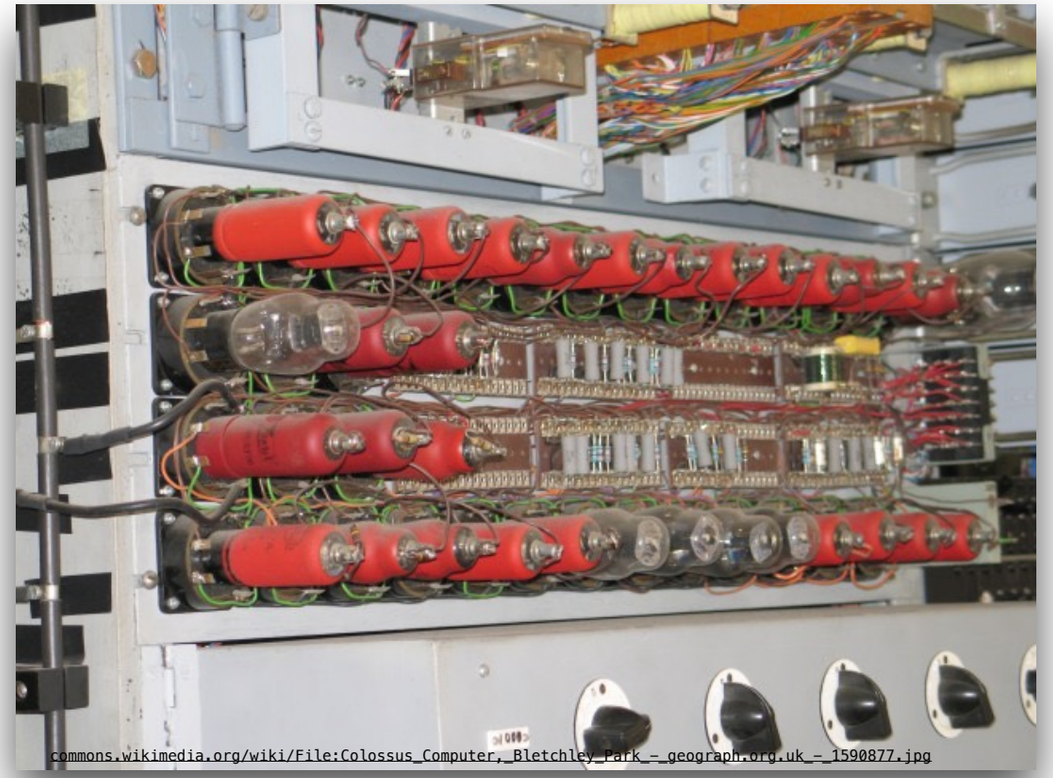
How to build a NOT gate with a magnet



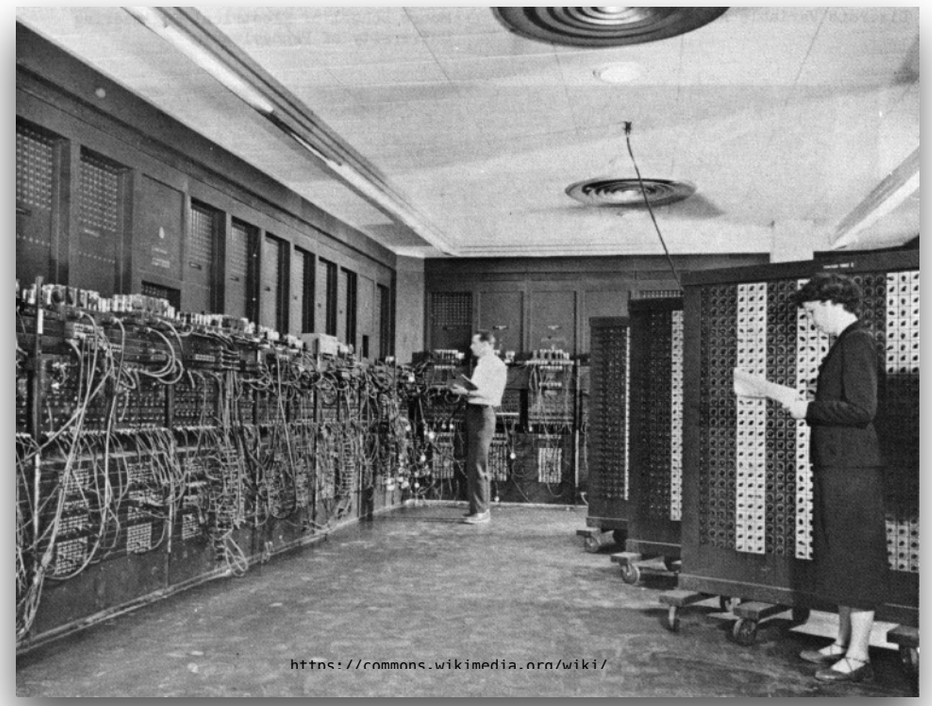
Which gate is this?



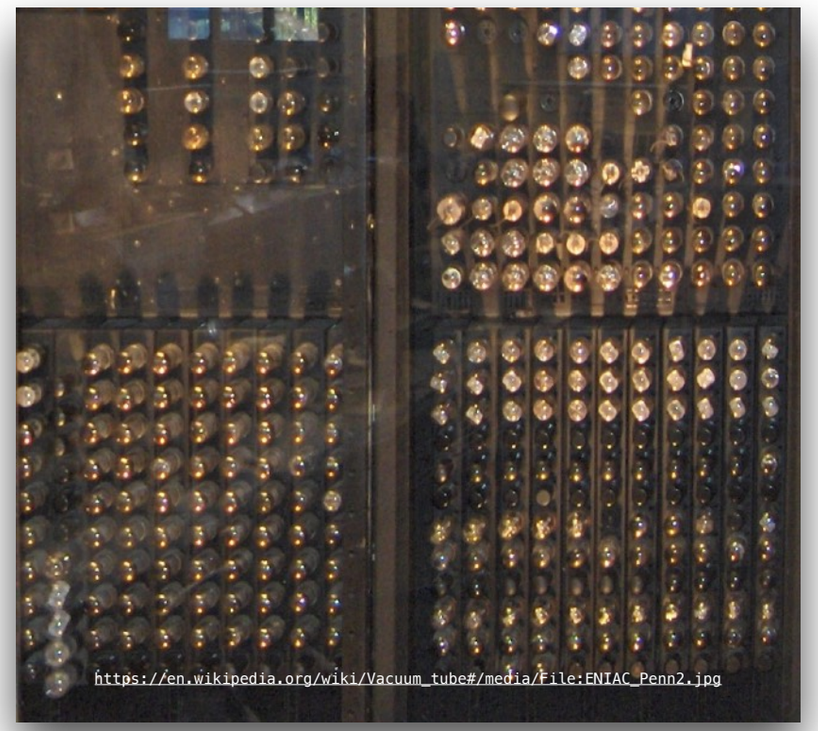
Mechanical systems aren’t always reliable



commons.wikimedia.org/wiki/File:Colossus_Computer,_Bletchley_Park_-_geograph.org.uk_-_1590877.jpg

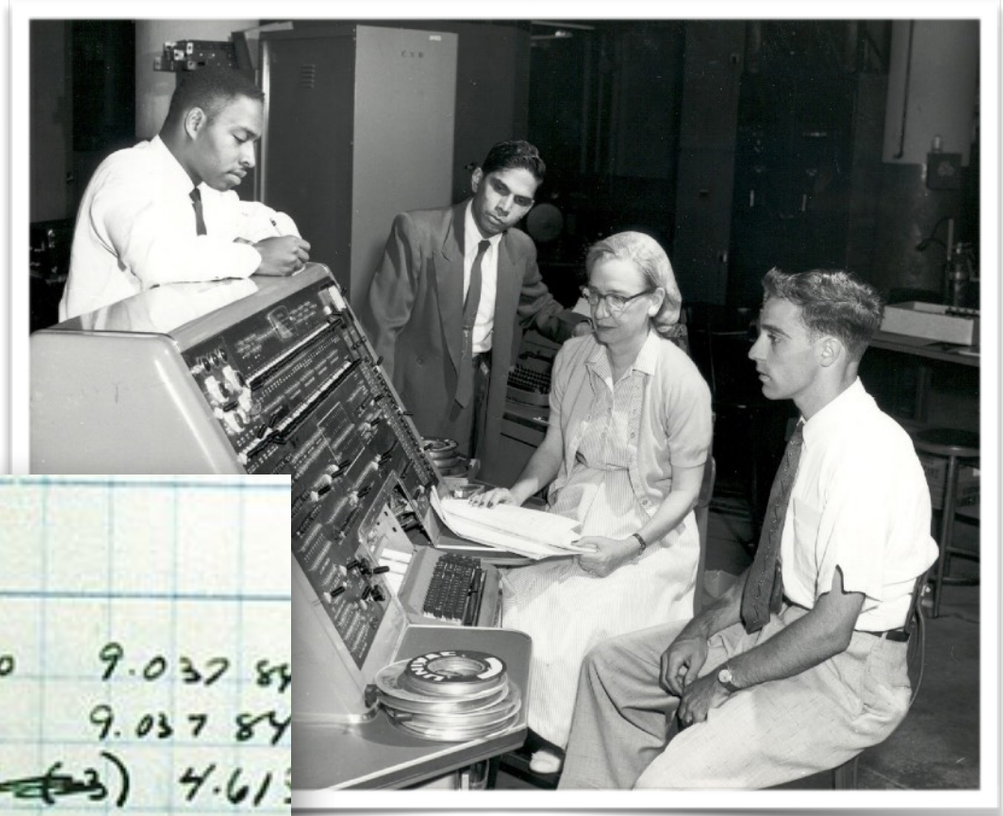


https://commons.wikimedia.org/wiki/



https://en.wikipedia.org/wiki/Vacuum_tube#/media/File:ENIAC_Penn2.jpg

Grace Hopper's bug



http://archive.computerhistory.org/resources/still-image/102741216.03.01.jpg

9/9

0800 Arctan started

1000 " stopped - arctan ✓

13" u/c (032) MP - MC

(033) PRO 2

correct

Relays 6-2 in 033 failed special speed te
In relay " 11.00 test.
Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Multy Adder Test.

1545 Relay #70 Panel
(moth) in relay.

First actual case of bug being found.

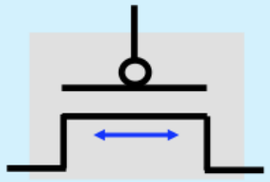
1700 closed down.

https://upload.wikimedia.org/wikipedia/commons/8/8a/H96566k.jpg

Transistors: smaller than most moths

Transistors are current ***switches***:

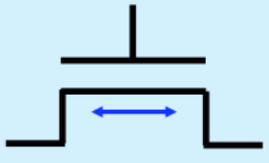
A low voltage here



allows current here

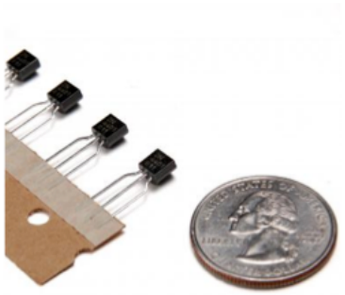
otherwise it's blocked

A high voltage here

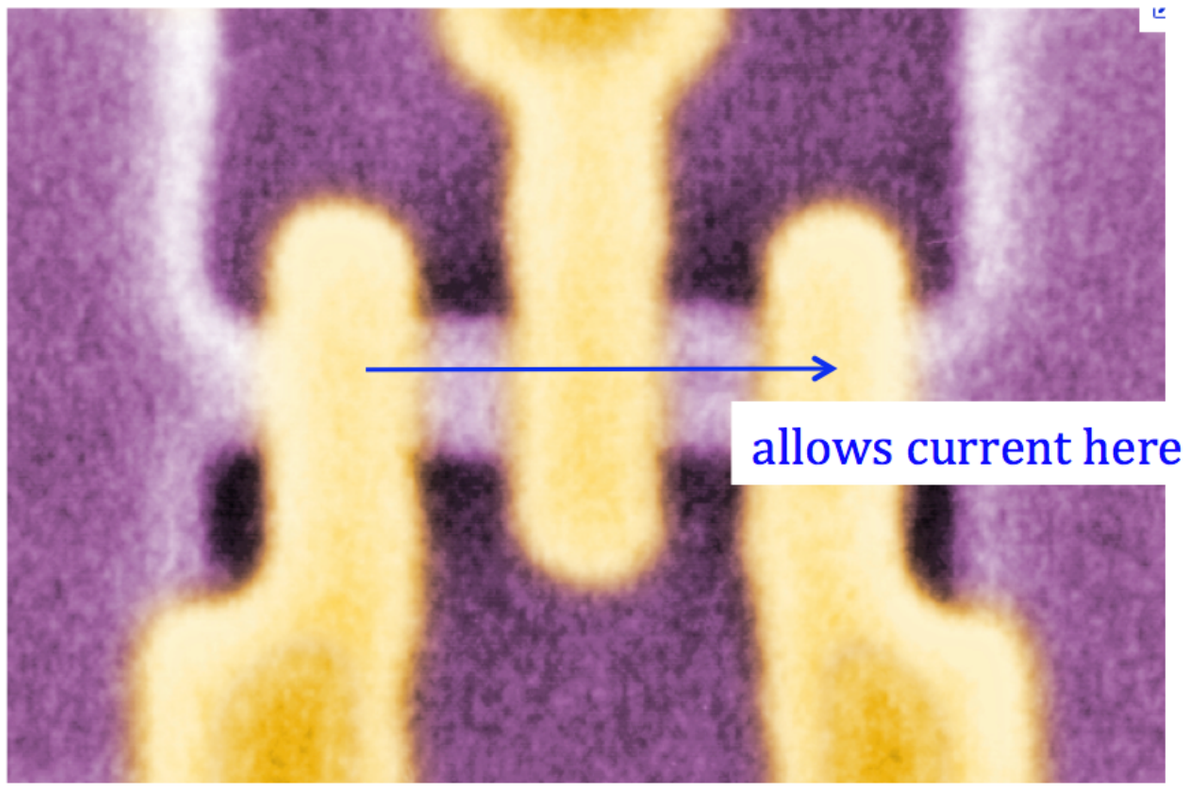


allows current here

otherwise it's blocked



voltage here



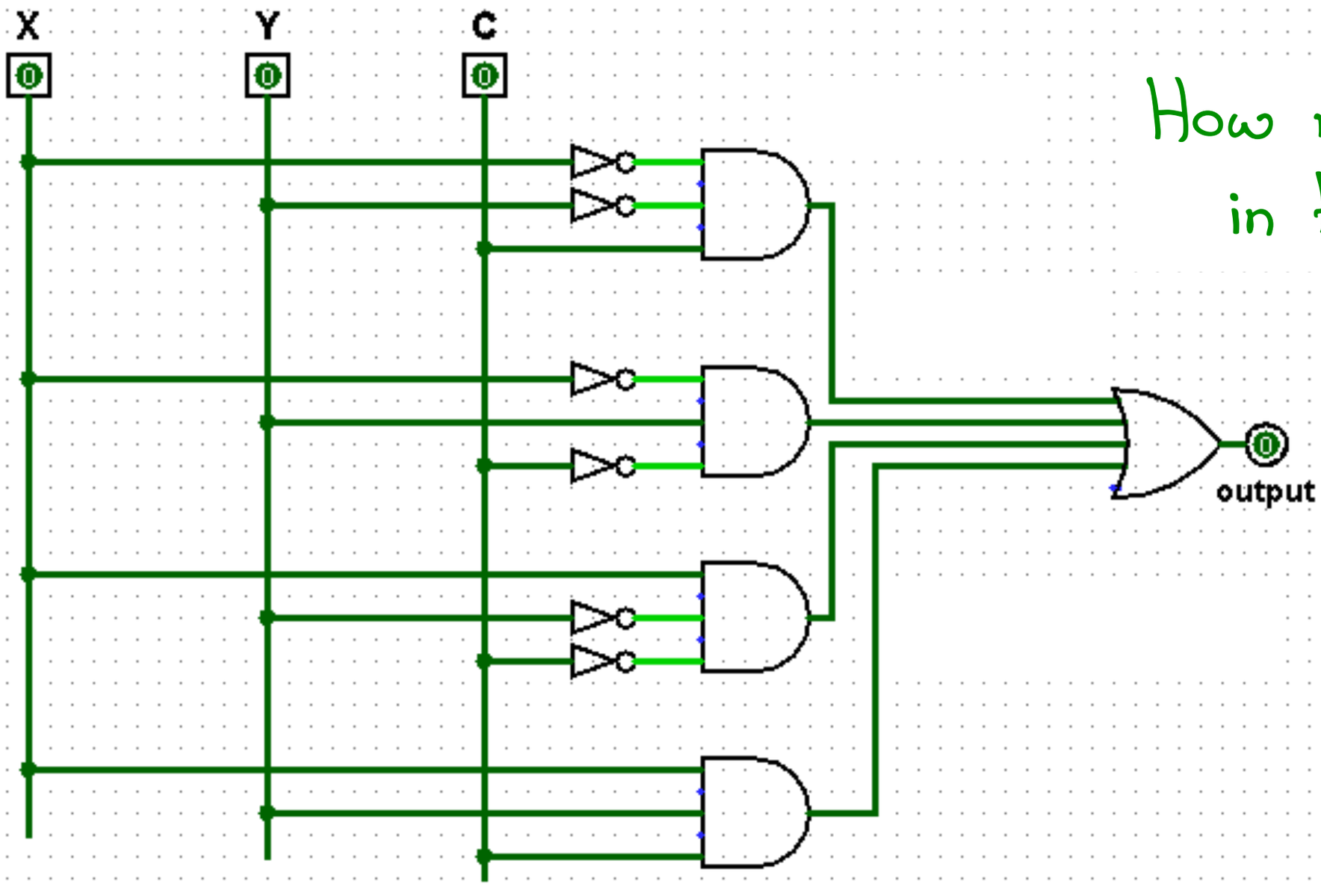
SET transistor

single-electron tunneling

Combinational Logic:

How to convert a truth table to a circuit?

A **minterm** is
an AND gate connected to all input bits
either directly *or* through a NOT gate



How many minterms
in this circuit?

Minterm expansion

How to use gates to build a truth table

Given a truth table:

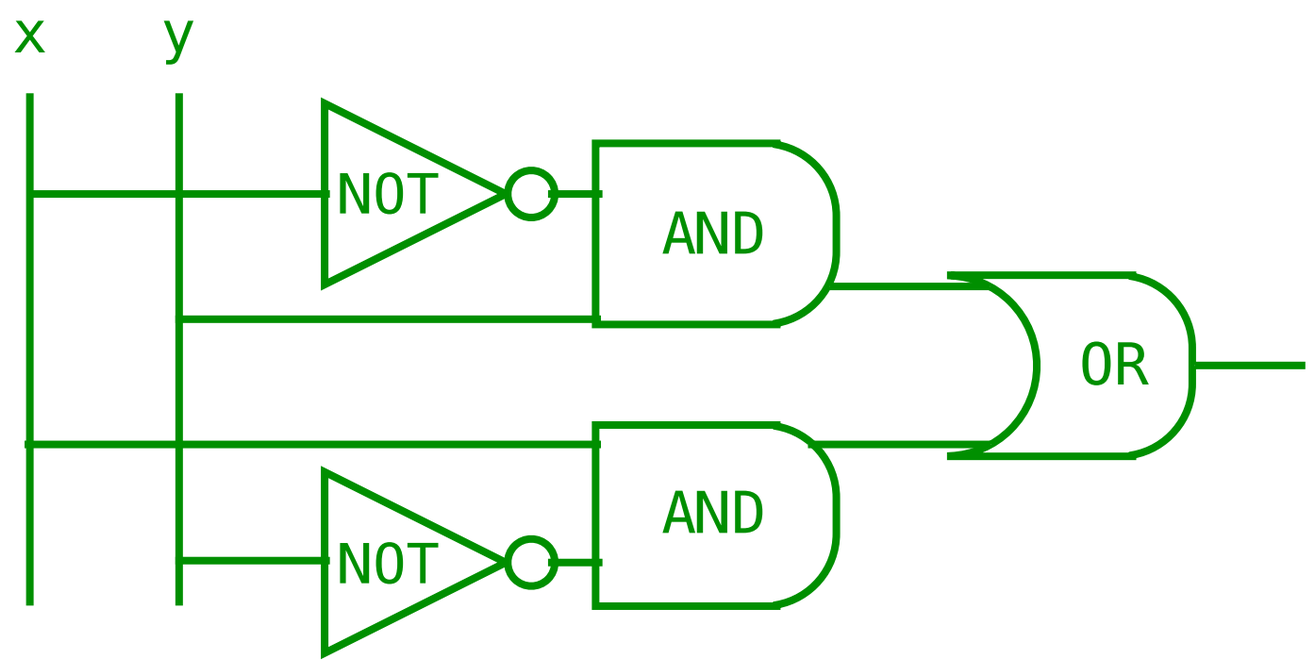
1. Look at all possible combinations of values for the inputs to the function

   For each combination of values that should cause the function to output `1`, build a minterm that outputs `1` *only* for those input values (and `0` for all other input values).

2. OR all the minterms together.

The resulting circuit implements the truth table.

input		output
x	y	
0	0	0
0	1	1
1	0	1
1	1	0



Do the minterm expansion for these functions

Do the minterm expansion version first, then see if you can make the circuit better (for some definition of better)

input			output
x	y	z	
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

input			output
x	y	z	
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Some thought experiments:

What do these two functions do?

Could you build your circuits without using an or gate?

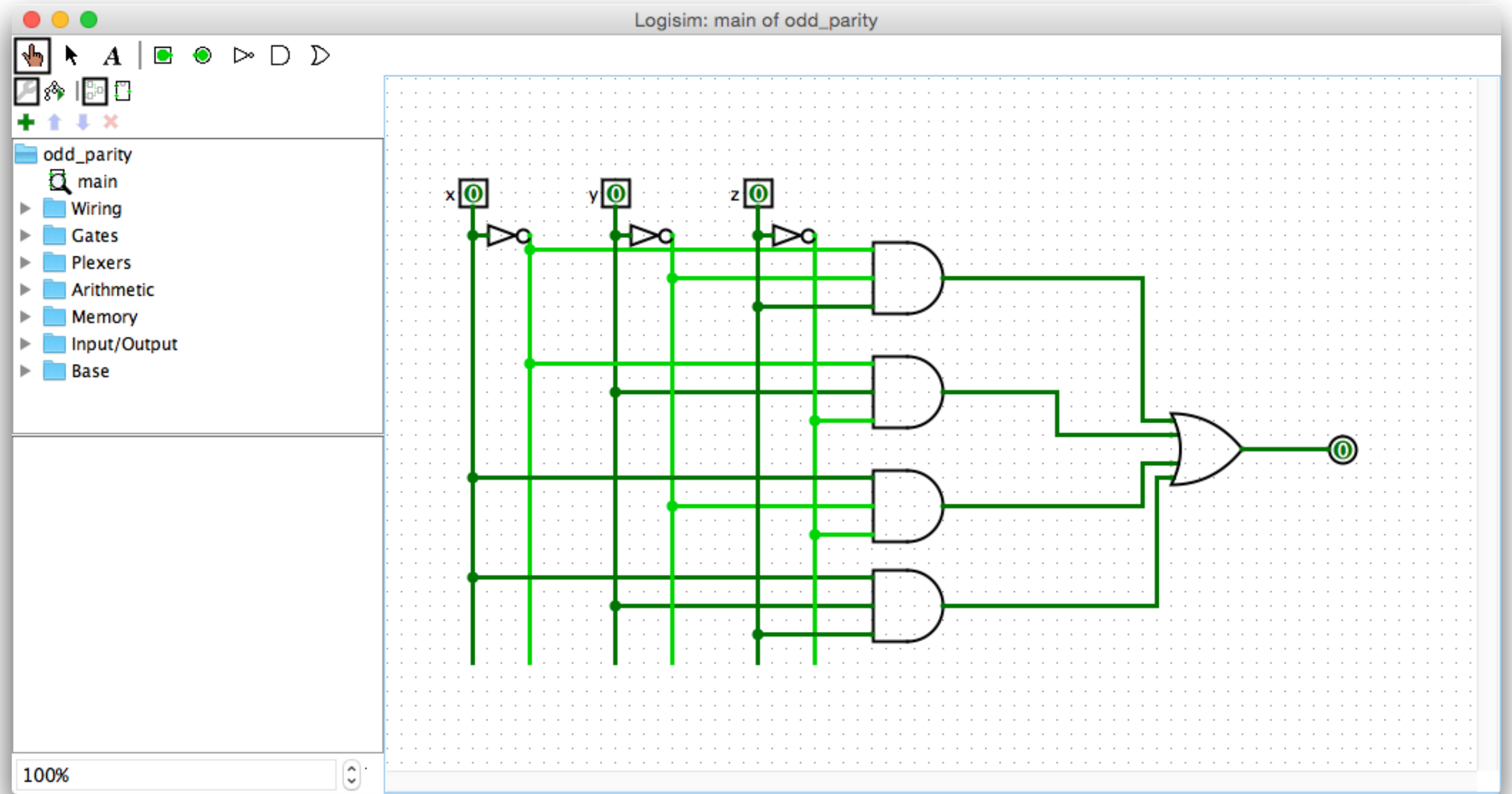
Could you combine these two circuits to compute binary addition?

https://www.youtube.com/watch?v=QNoQvjlmGdk

Logism

Pro tip: use “rails” to lay out your circuit

input			output
x	y	z	
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



A full adder

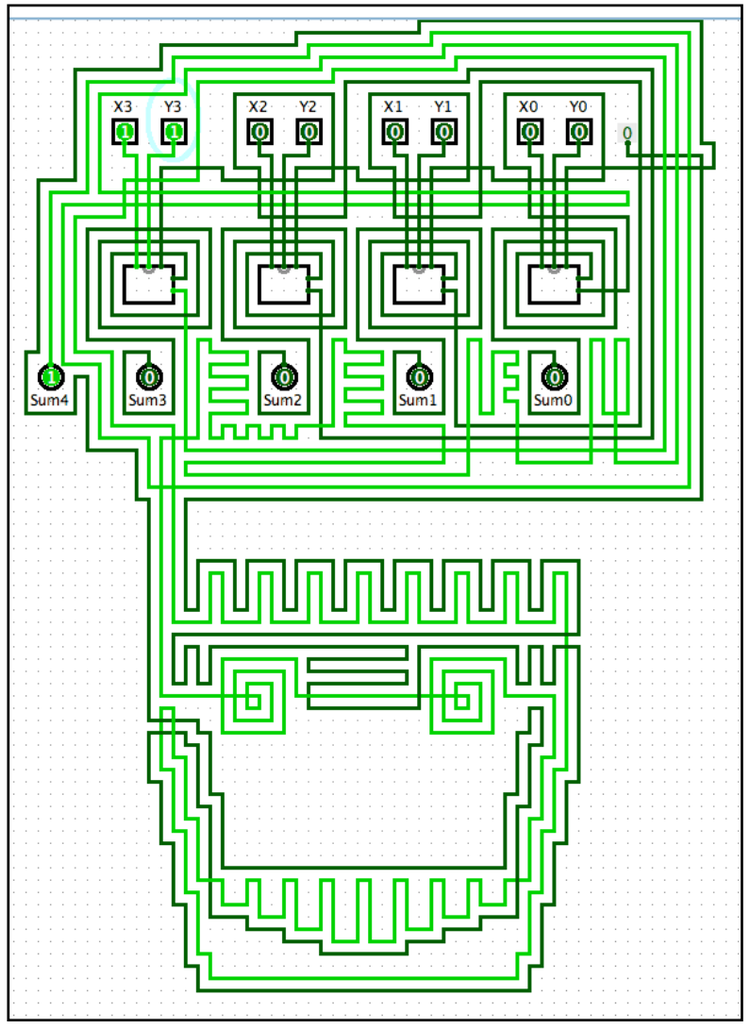
sums three bits of input to create two bits of output

carry bits

input			output	
x	y	c_{in}	c_{out}	sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

This table might look familiar. It's the combination of the two tables we saw earlier...

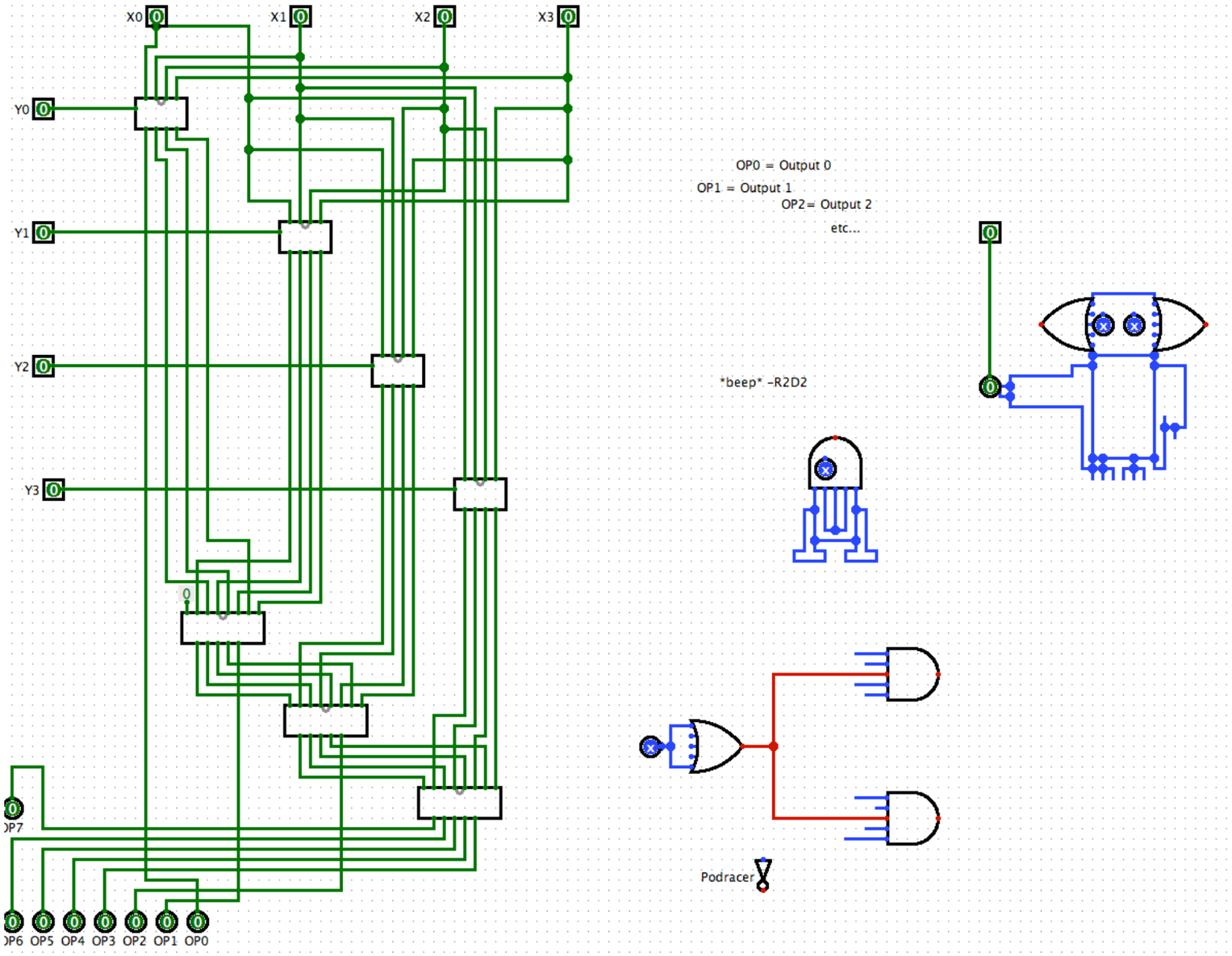
Two adders

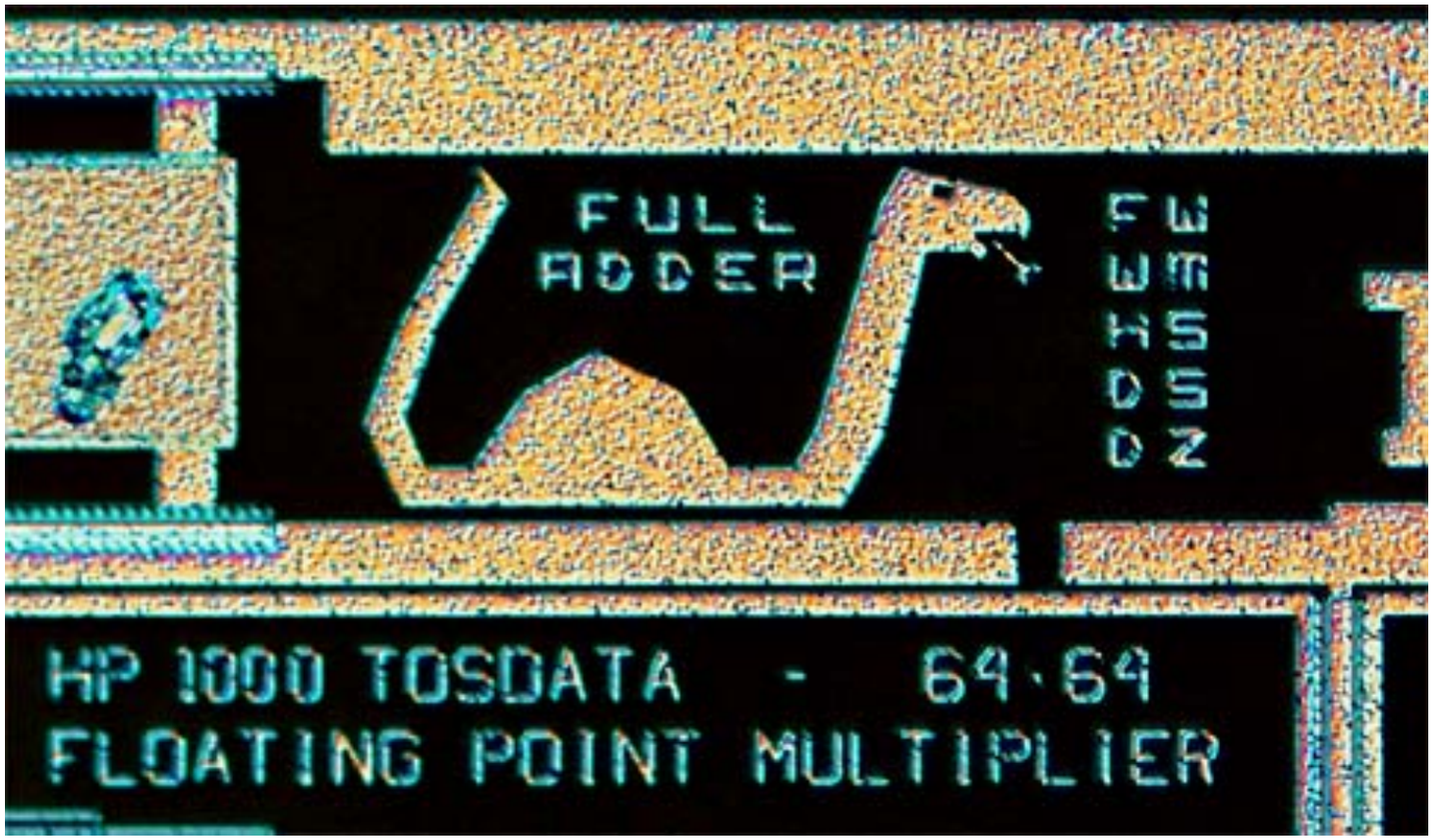


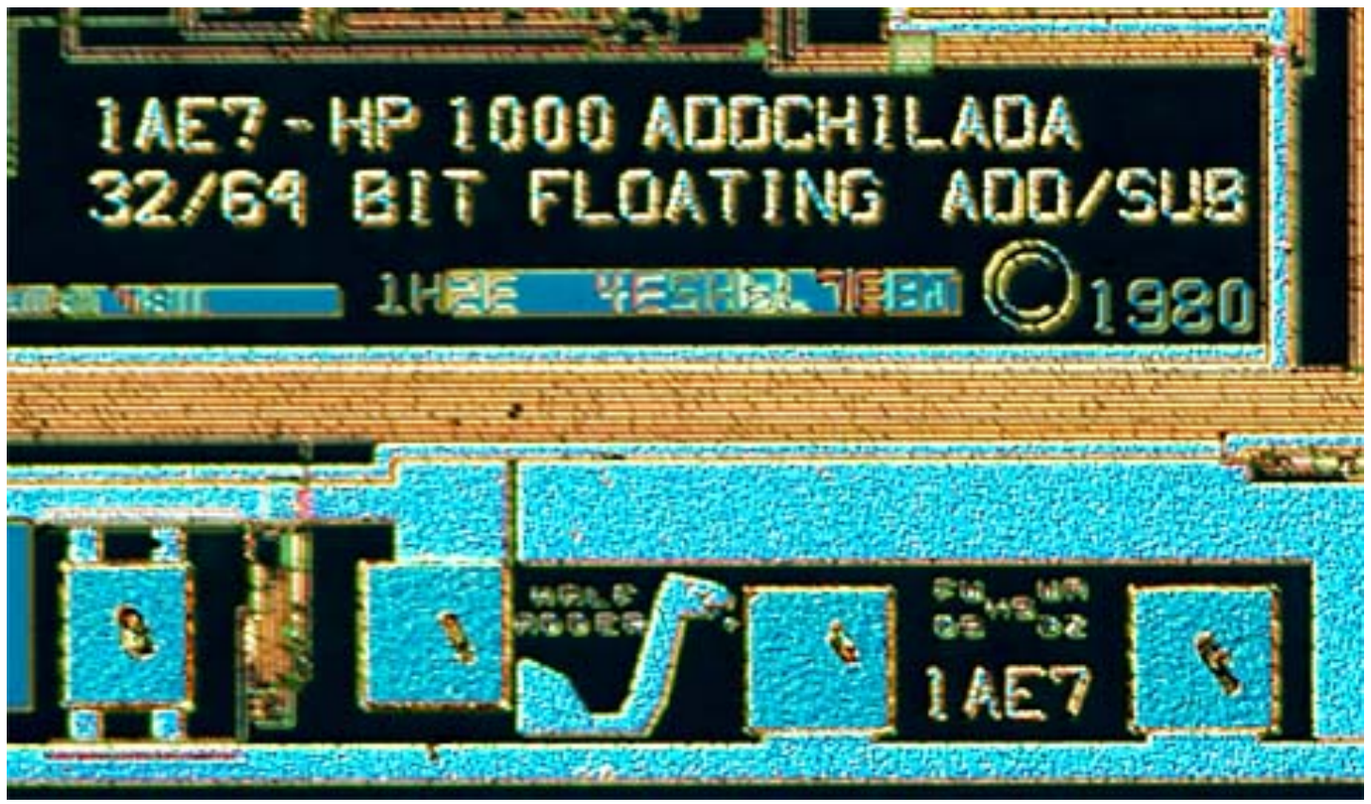
zombie ripple-carry adder

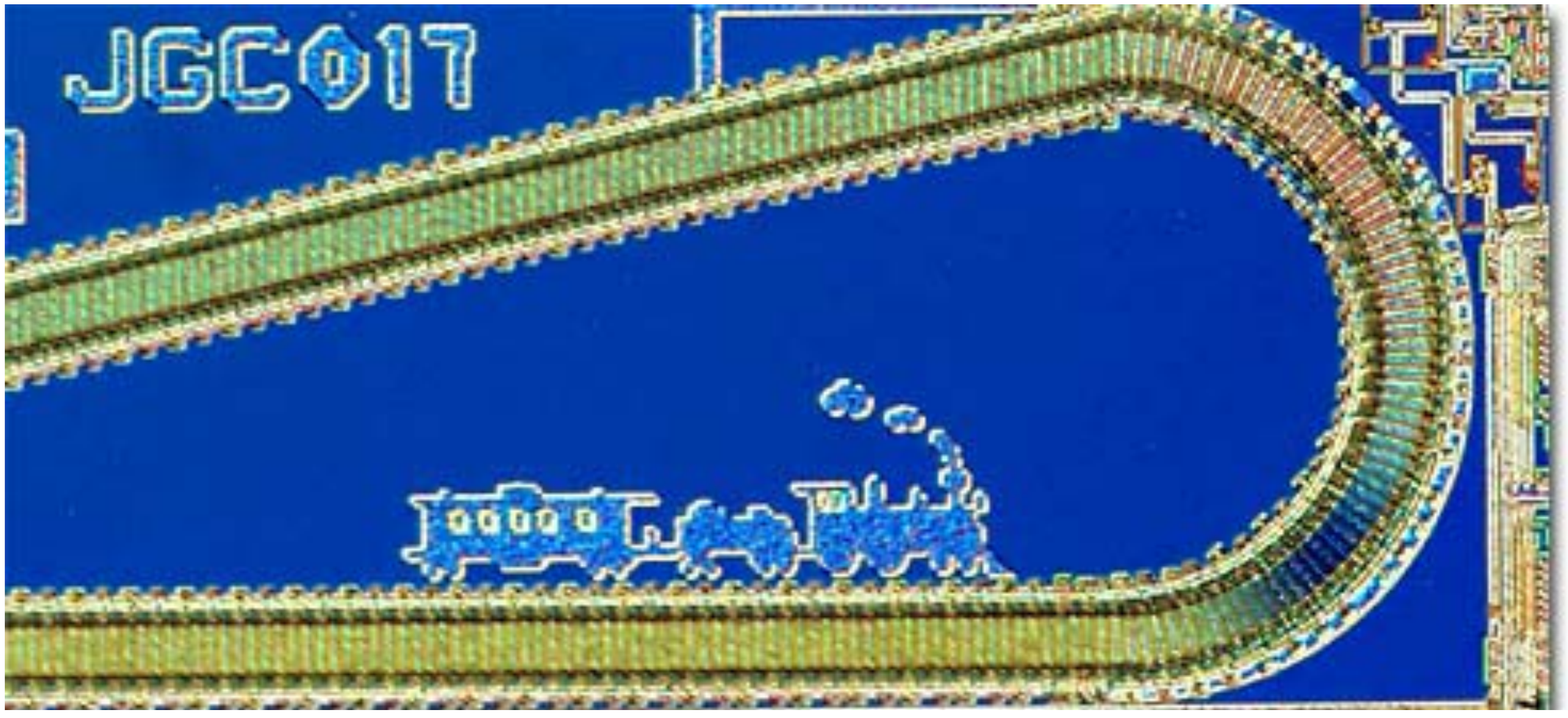
actual adder

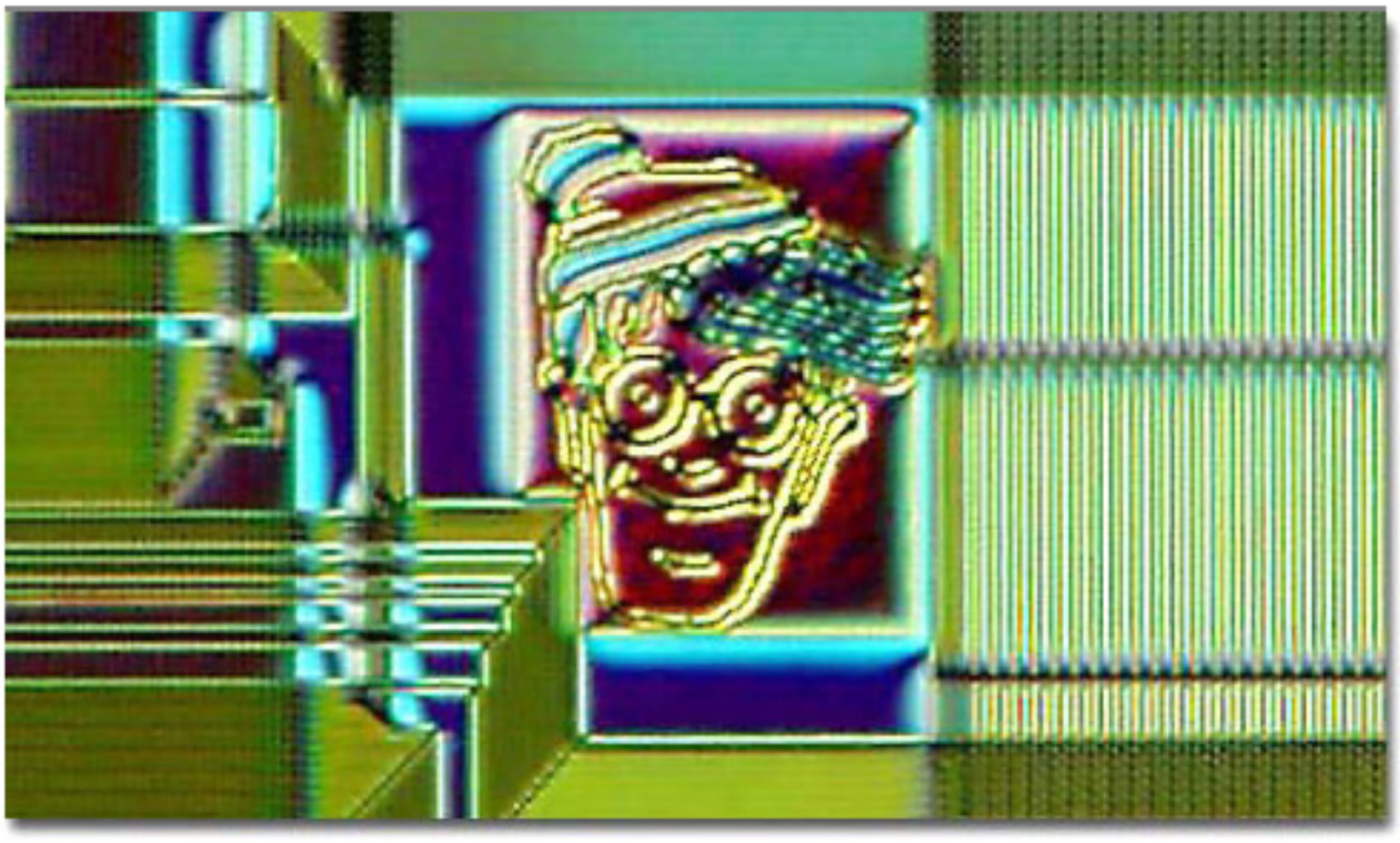
Your canvas


X0
X1
X2
X3
Y0
Y1
Y2
Y3
OP7
OP6
OP5
OP4
OP3
OP2
OP1
OP0
OP0 = Output 0
OP1 = Output 1
OP2= Output 2
etc...
beep -R2D2
Podracer


FULL
ADDER
FW
WM
HS
DS
DZ
HP 1000 TOSDATA - 64·64
FLOATING POINT MULTIPLIER


1AE7-HP 1000 ADDCHILADA
32/64 BIT FLOATING ADD/SUB
1980
1AE7


JGC017

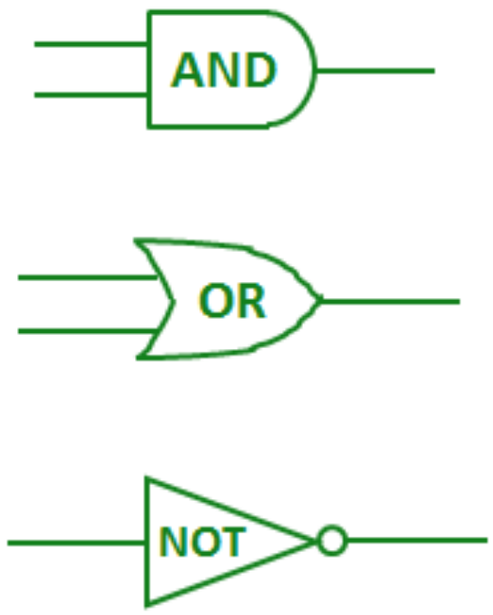


The "silicon zoo": micro.magnet.fsu.edu/creatures/index.html

A remarkable claim!

Functional completeness

We can implement *any* boolean function using only AND, OR, NOT.



Thank you, minterm expansion!

A remarkable claim!

Functional completeness

We can implement *any* boolean function using only `AND`, `NOT`.

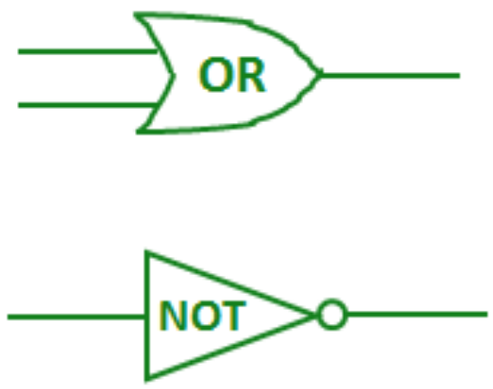
A remarkable claim!

Functional completeness

We can implement *any*

boolean function using only

OR, NOT.



What counts as a problem?

Decision problems on finite, bitstring inputs.

What kinds of **problems** can **computers** solve?

Can NOT + OR + AND solve all the problems that a DFA can? How about a Turing Machine?

What counts as a computer?