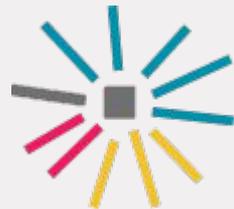


# HARDWARE

PALIMPSEST



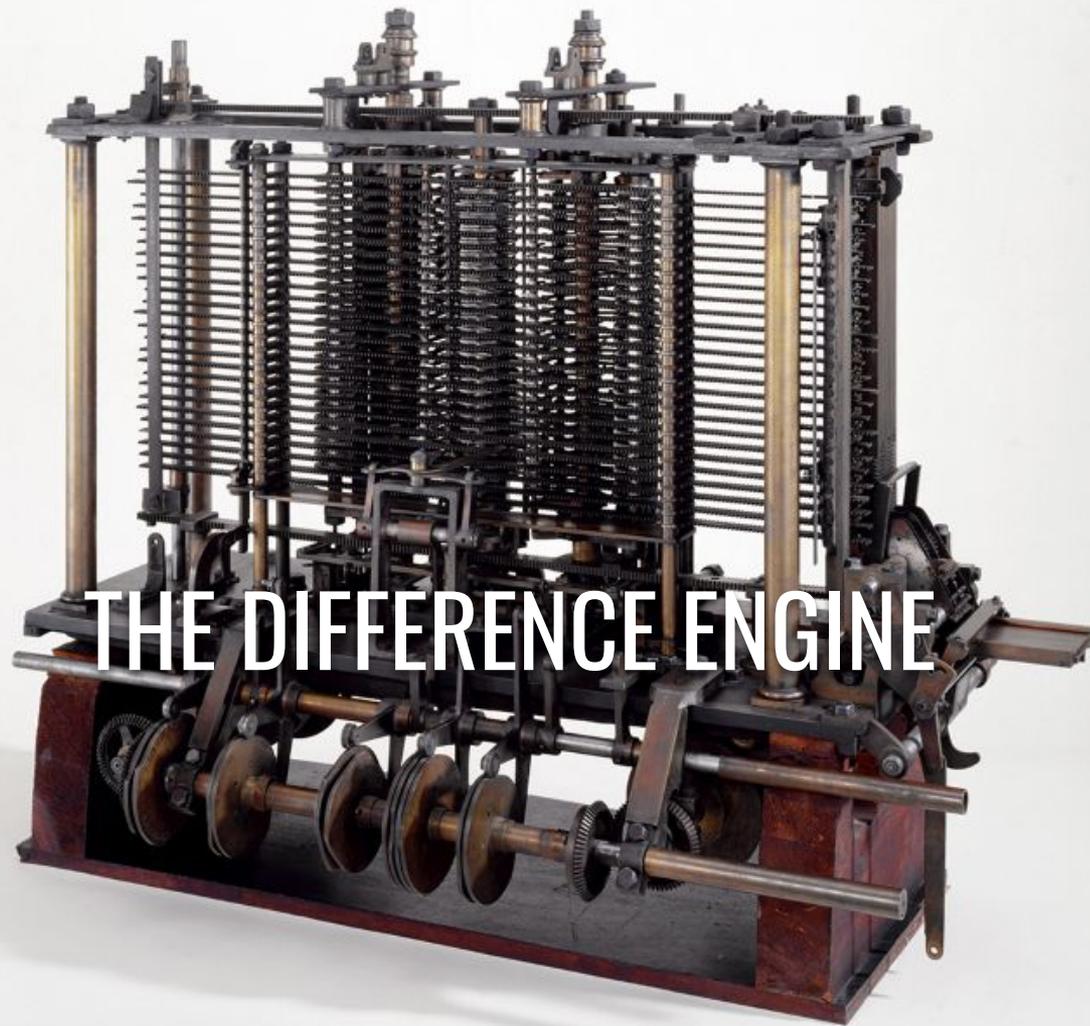
# HARDWARE

---

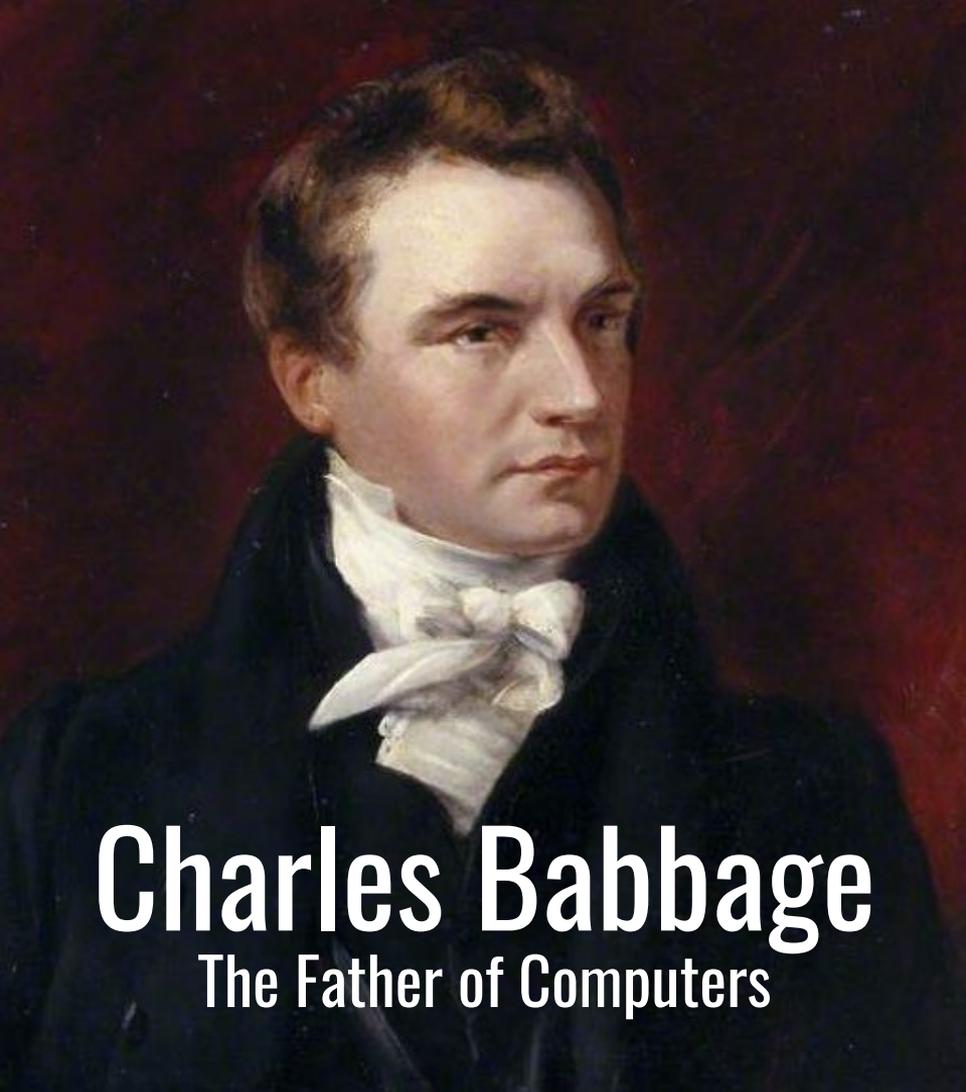
What is a computer even?

- Human/Computer Relationship
- Inputs
- Storage
- Processor
- Outputs
- Binary
- Exercise

**HARDWARE**



THE DIFFERENCE ENGINE

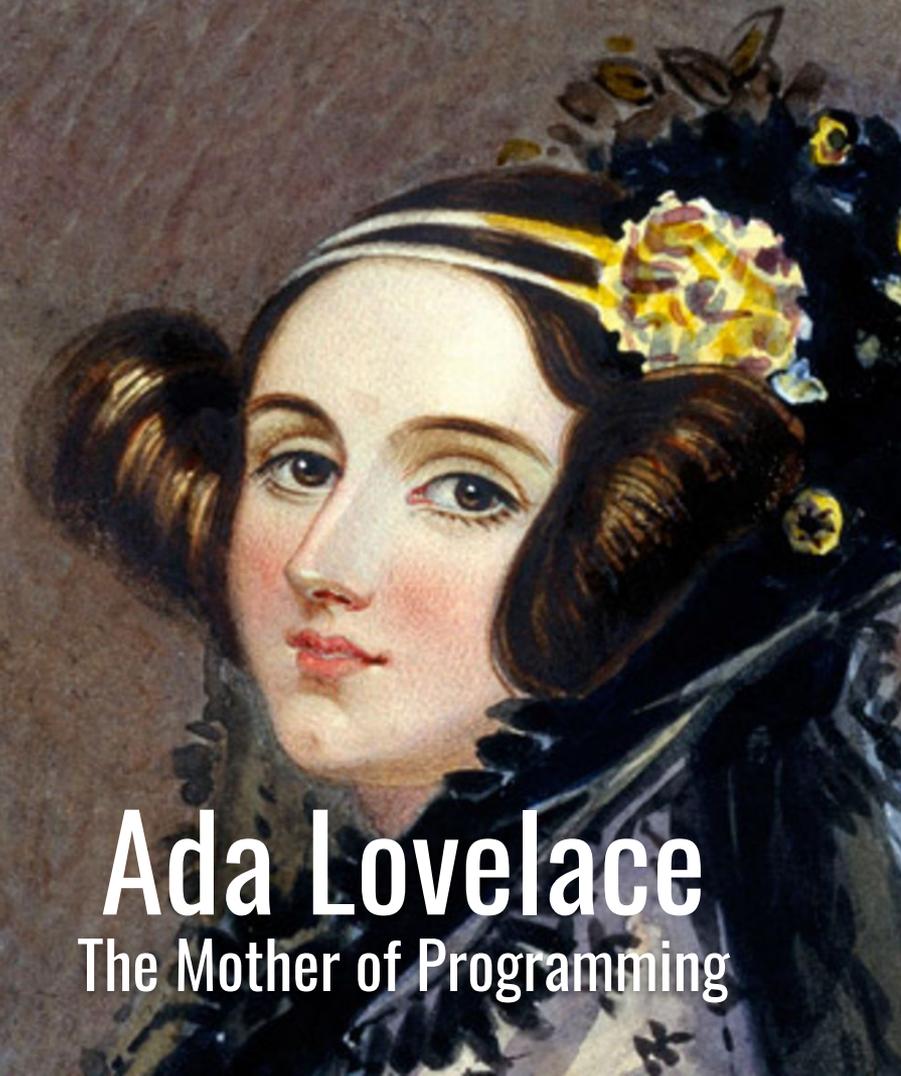


**Charles Babbage**  
The Father of Computers

“As soon as an Analytical Engine exists, it will necessarily guide the future course of science.”

“The Analytical Engine has no pretensions whatever to originate anything. It can do whatever we know how to order it to perform...

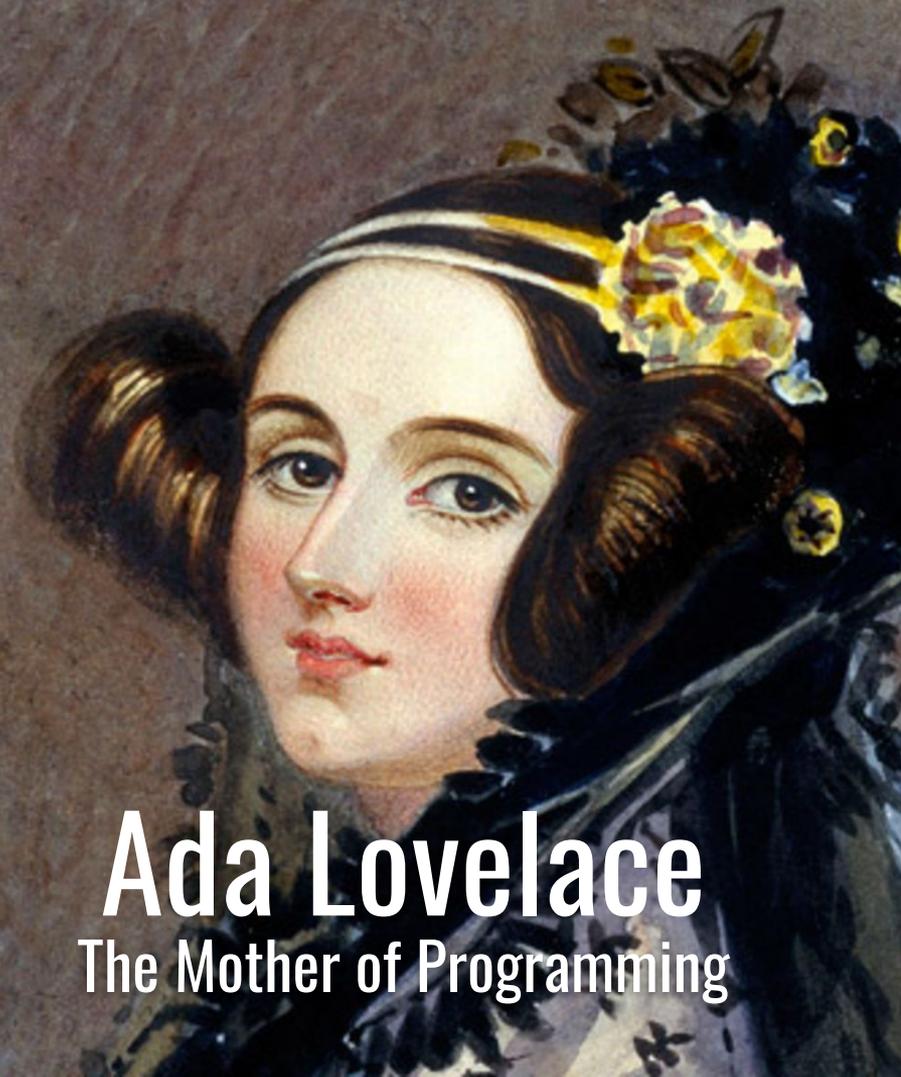
But it is likely to exert an indirect and reciprocal influence on science itself.”



**Ada Lovelace**  
The Mother of Programming

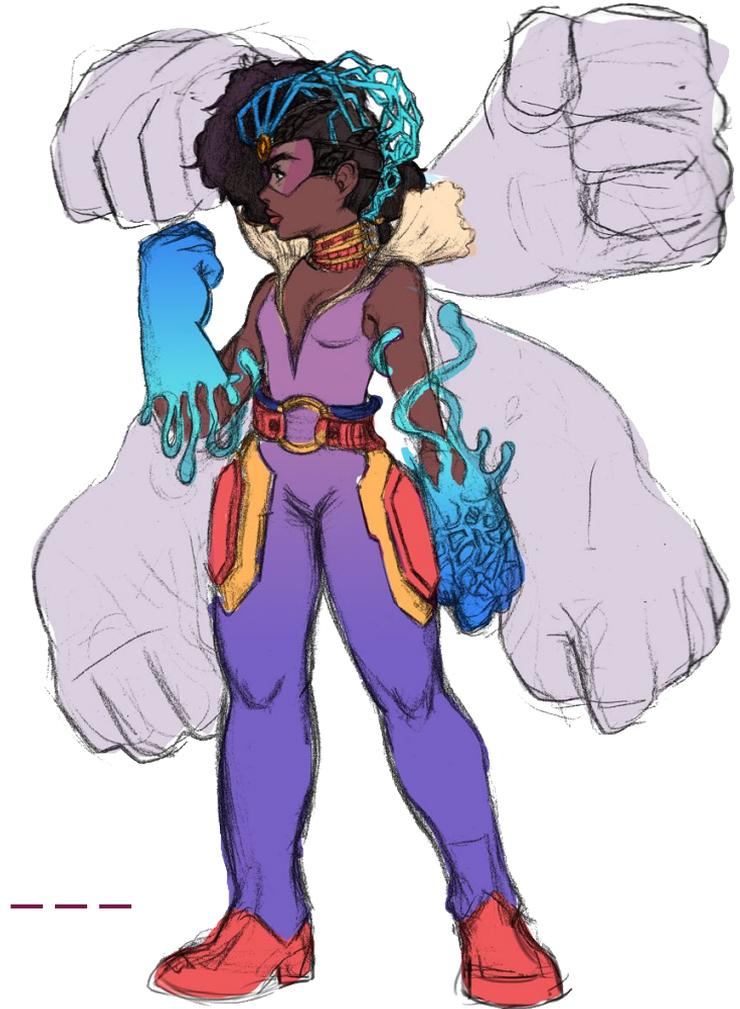
“The Analytical Engine has no pretensions whatever to originate anything. **It can do whatever we know how to order it to perform...**

But it is likely to exert an indirect and reciprocal influence on science itself.”



**Ada Lovelace**  
The Mother of Programming

How do we order  
a computer to do  
anything?



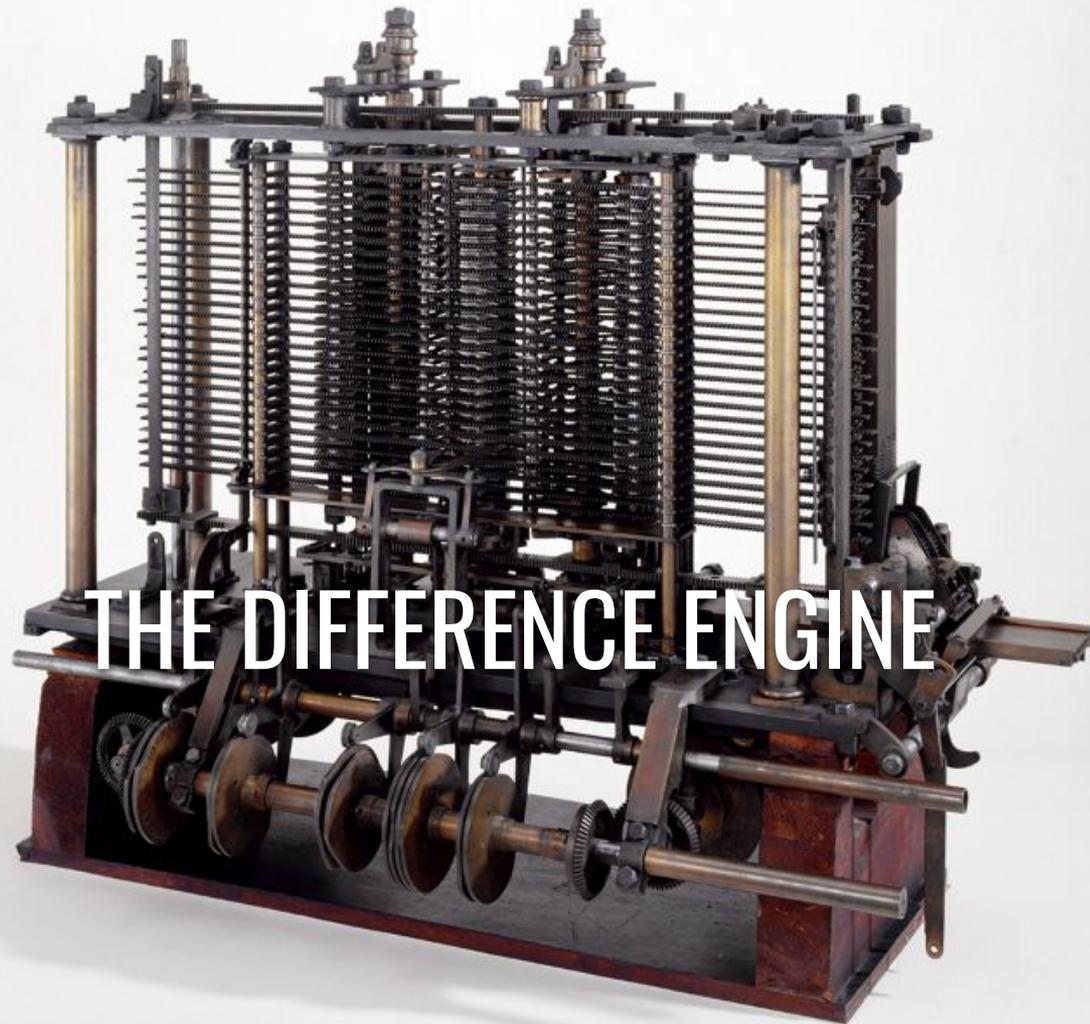


**What language  
does it speak?**

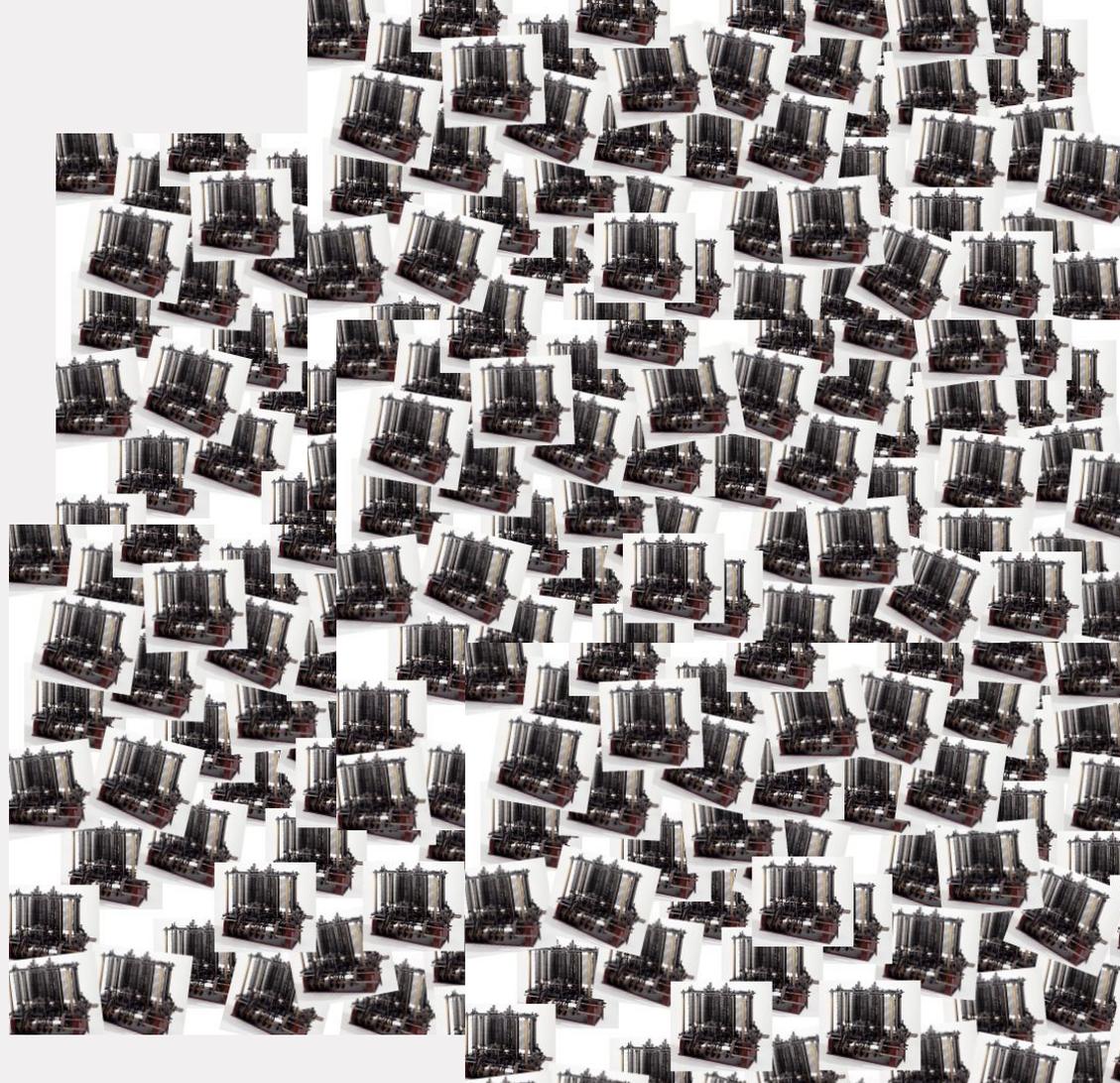
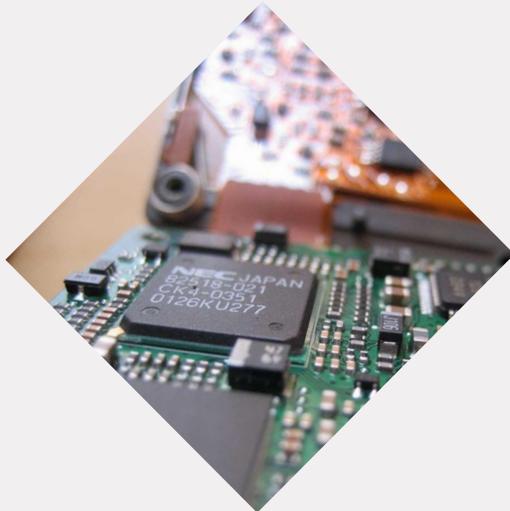
---

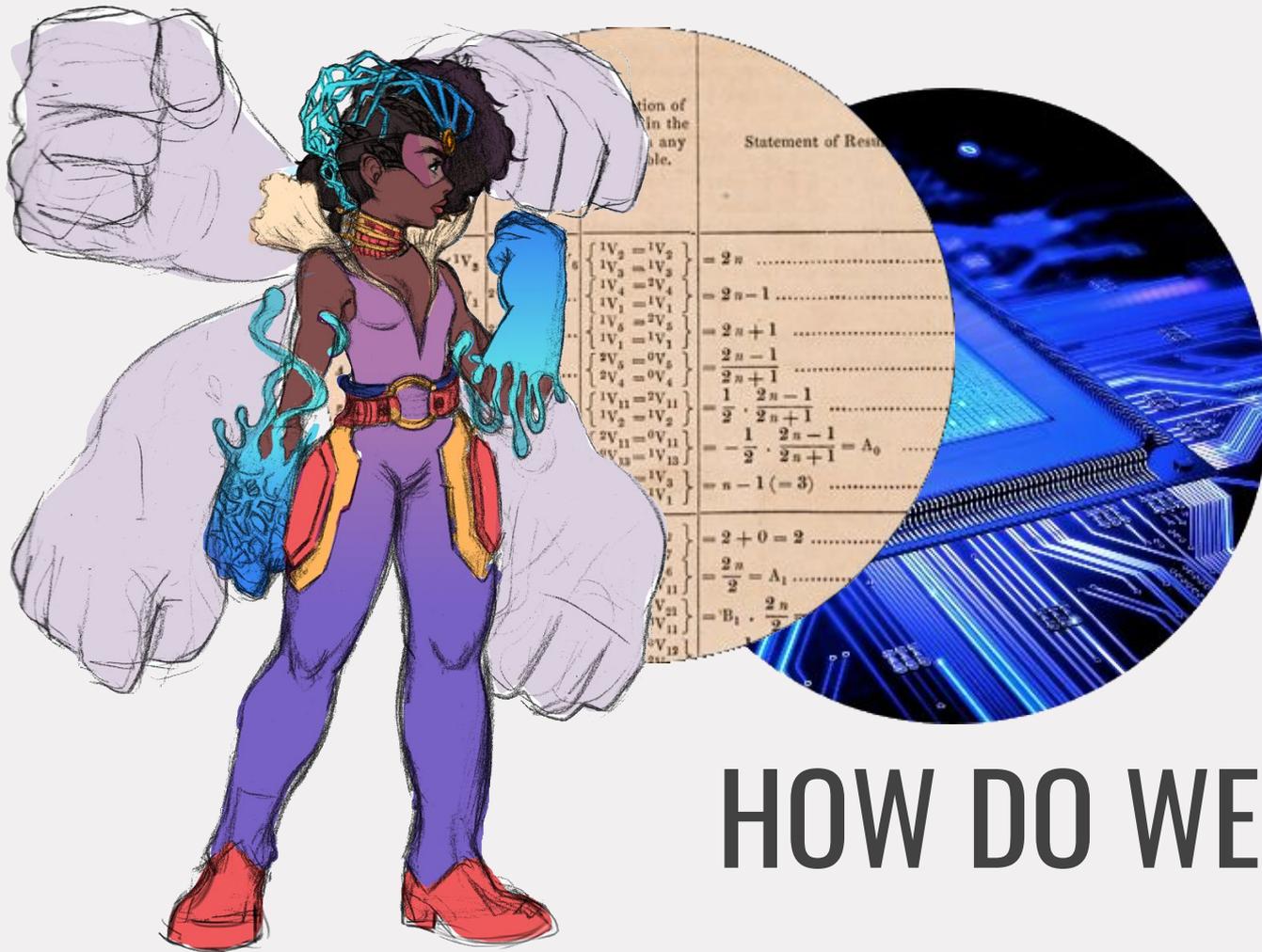
# Computers speak





The





HOW DO WE FIT?



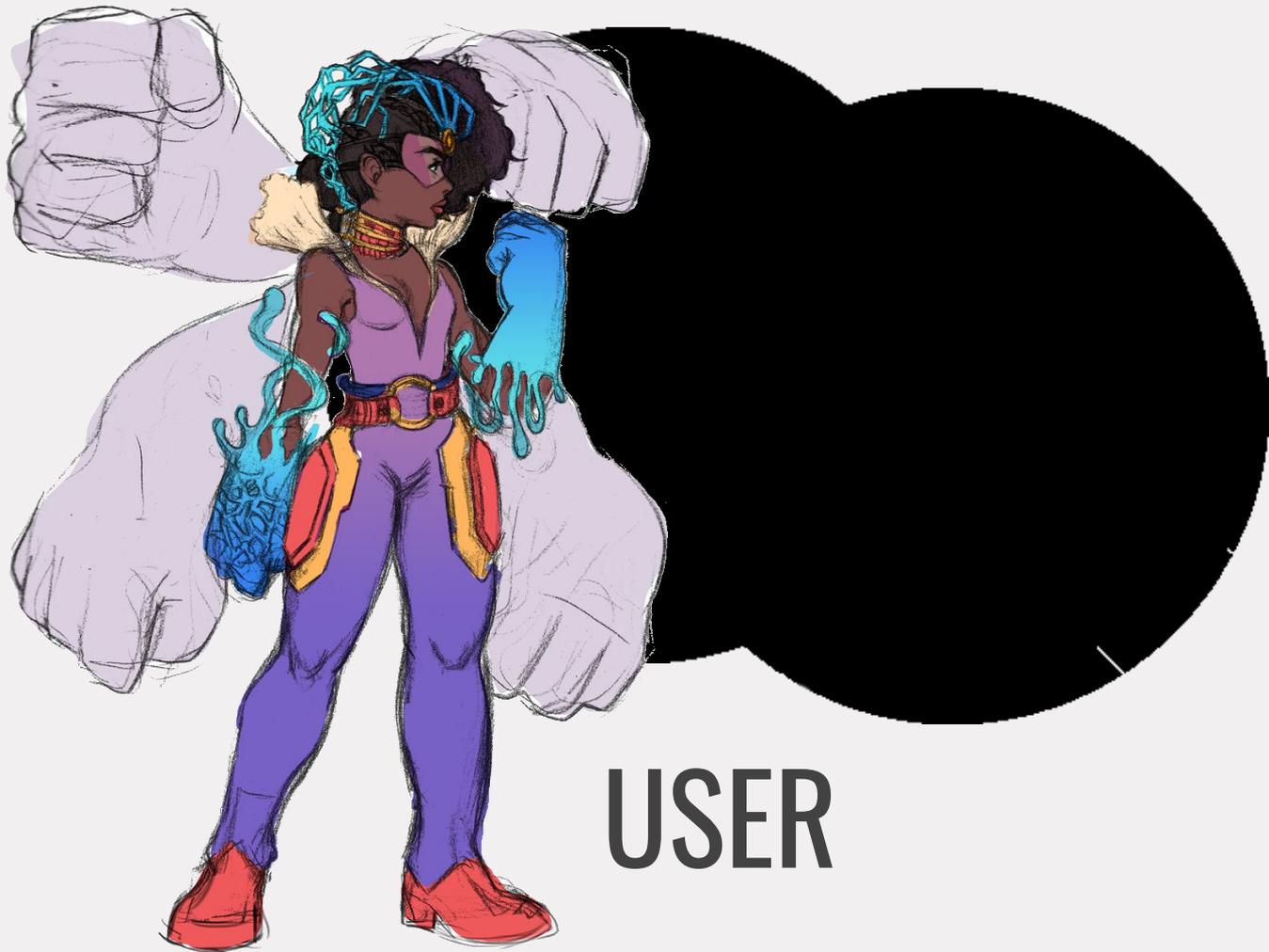
I WANT A THING.



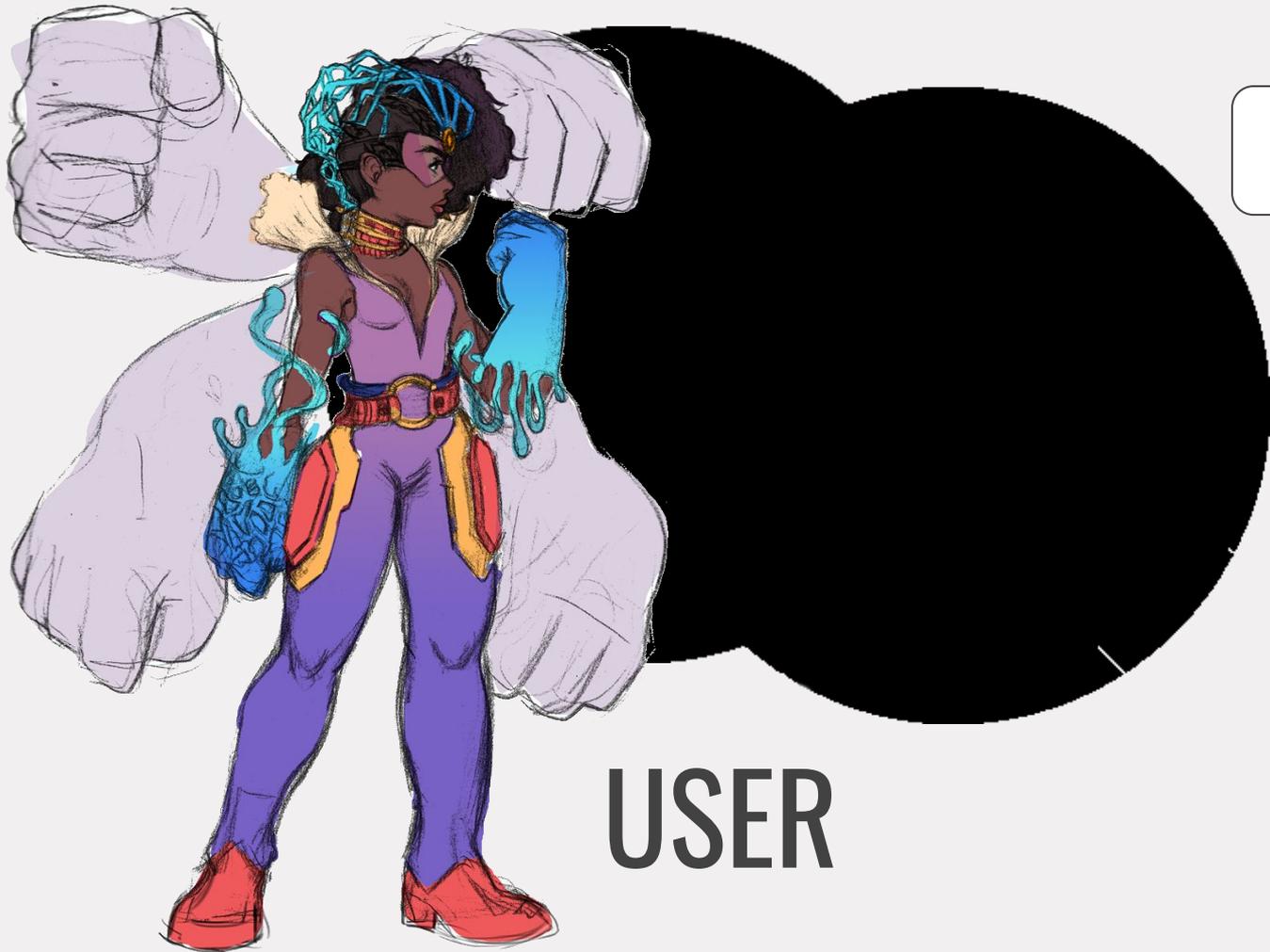
CAN YOU CODE?



NO.

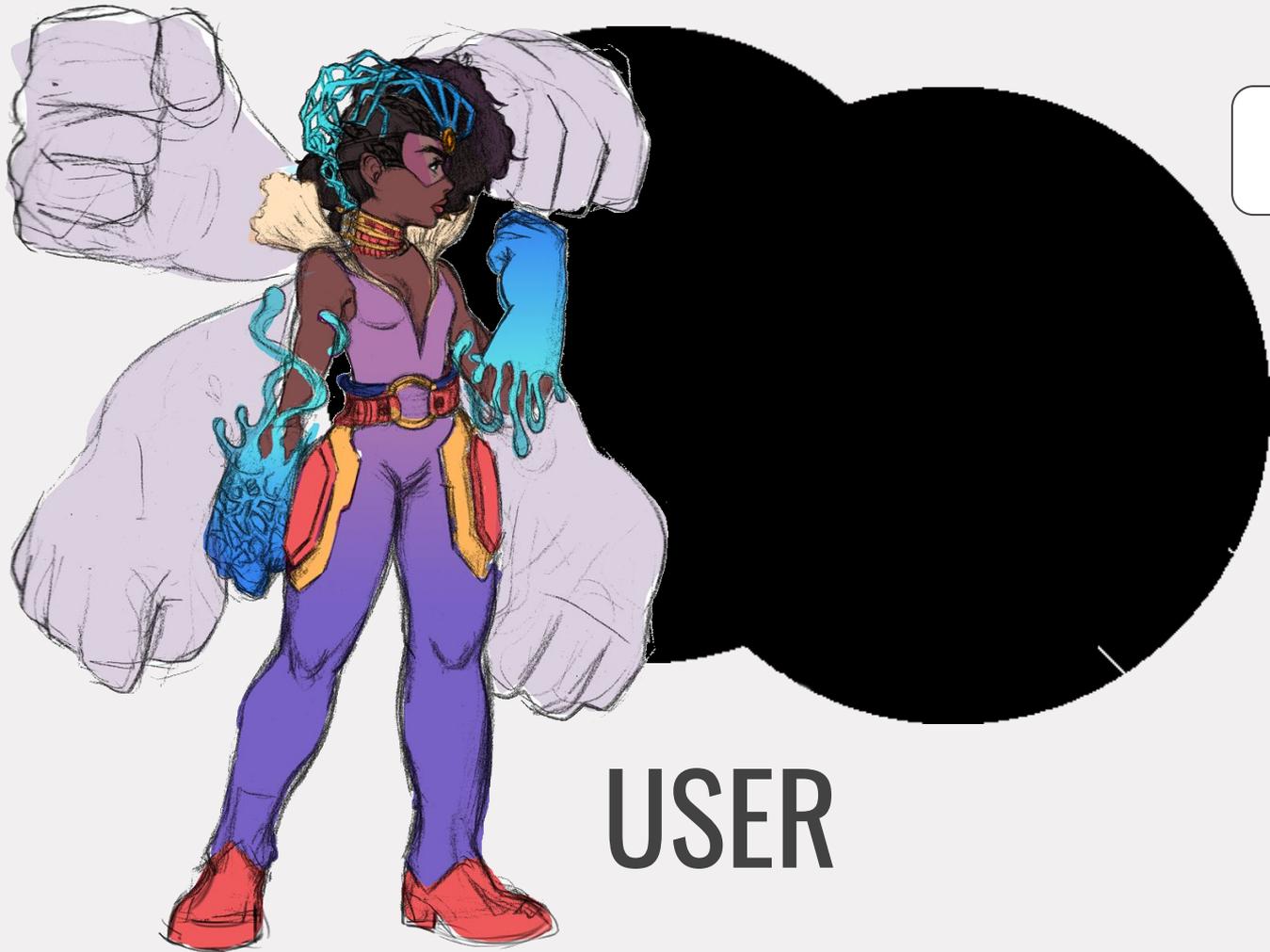


**USER**



GIVE ME MONEY.

**USER**



I WILL MAKE THE  
THING.

**USER**

**OR**



YES.



I WILL MAKE THE  
THING.



**PROGRAMMER**

**BTW**



**USER**

GIVE ME MONEY.

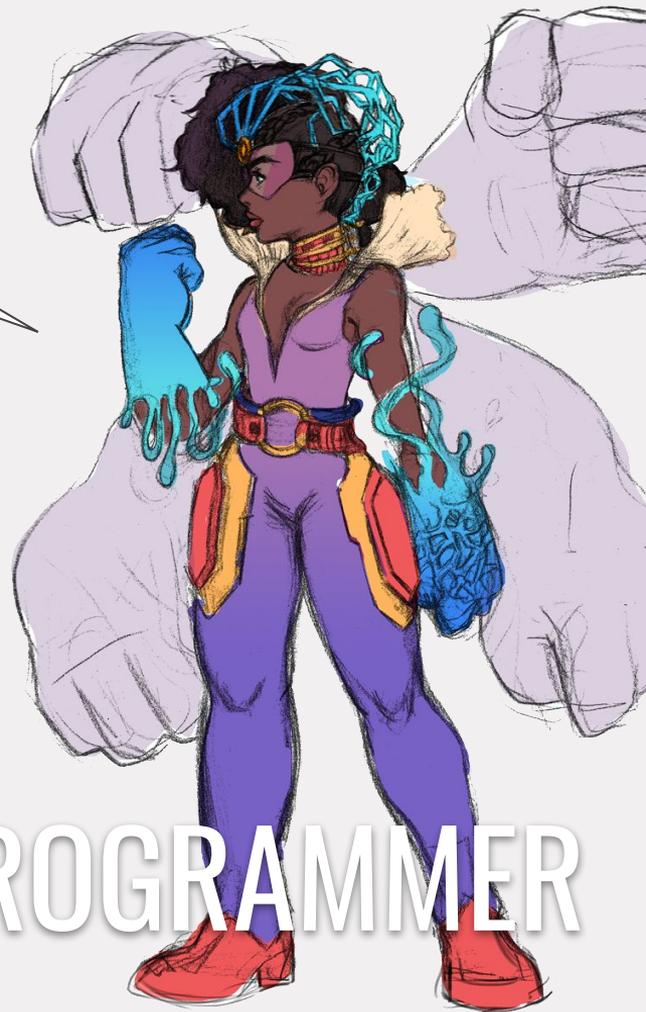


**PROGRAMMER**



**USER**

I WILL MAKE THE  
THING.



**PROGRAMMER**



**PROGRAMMER**



I WILL MAKE THE  
THING WITH CODE.

**PROGRAMMER**



I WILL MAKE THE  
THING WITH CODE.

AT HOME.

# PROGRAMMER



I WILL MAKE THE  
THING WITH CODE.

AT HOME.

LISTENING TO MY  
OWN MUSIC.

**PROGRAMMER**



I WILL MAKE THE  
THING WITH CODE.

AT HOME.

LISTENING TO MY  
OWN MUSIC.

IN MY PJs.

# PROGRAMMER



I WILL MAKE THE THING WITH CODE.

AT HOME.

LISTENING TO MY OWN MUSIC.

IN MY PJs.

RN.

# PROGRAMMER



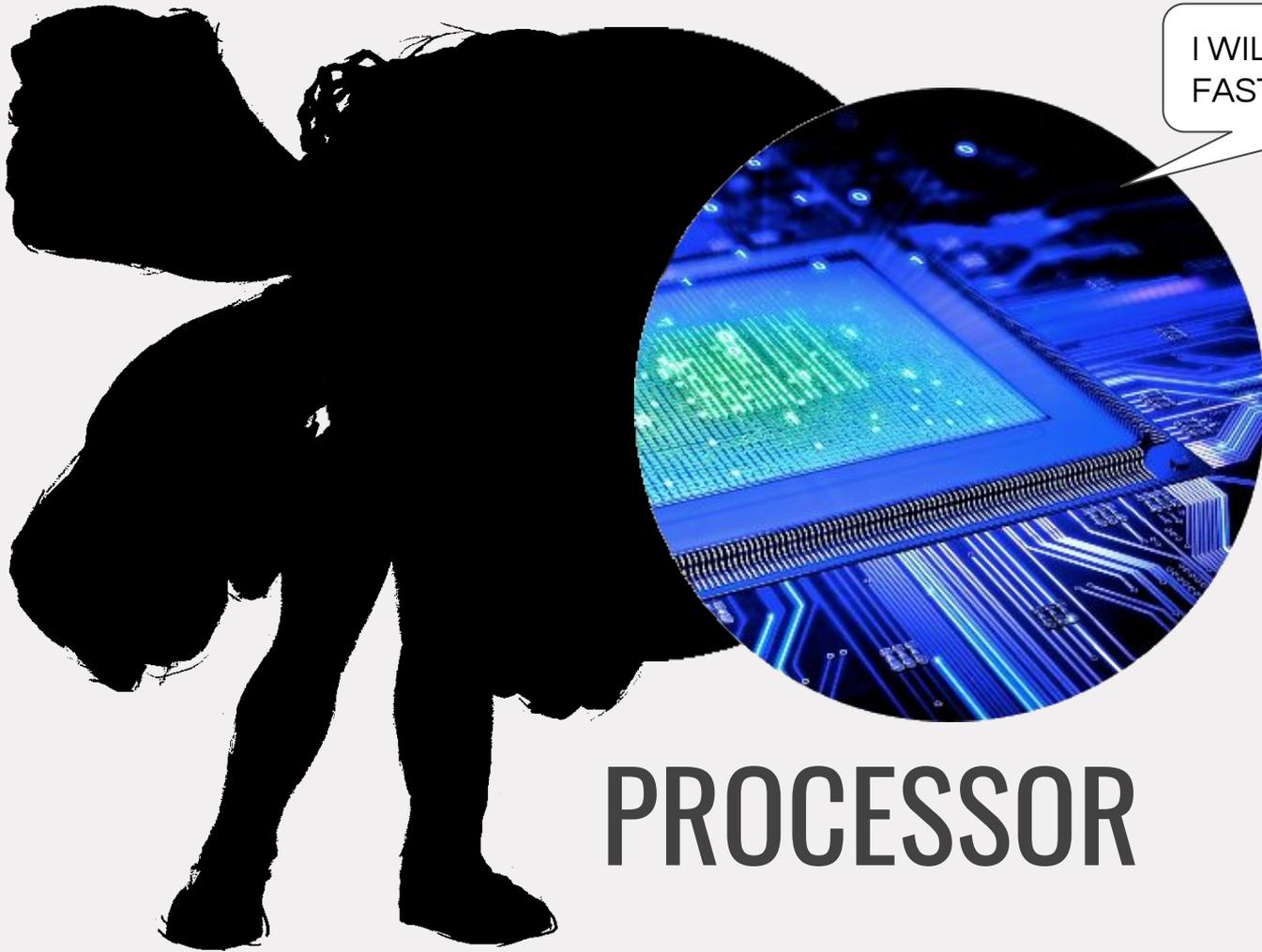
I WILL MAKE THE  
THING WITH CODE.

**PROGRAMMER**

I AM THE THING.

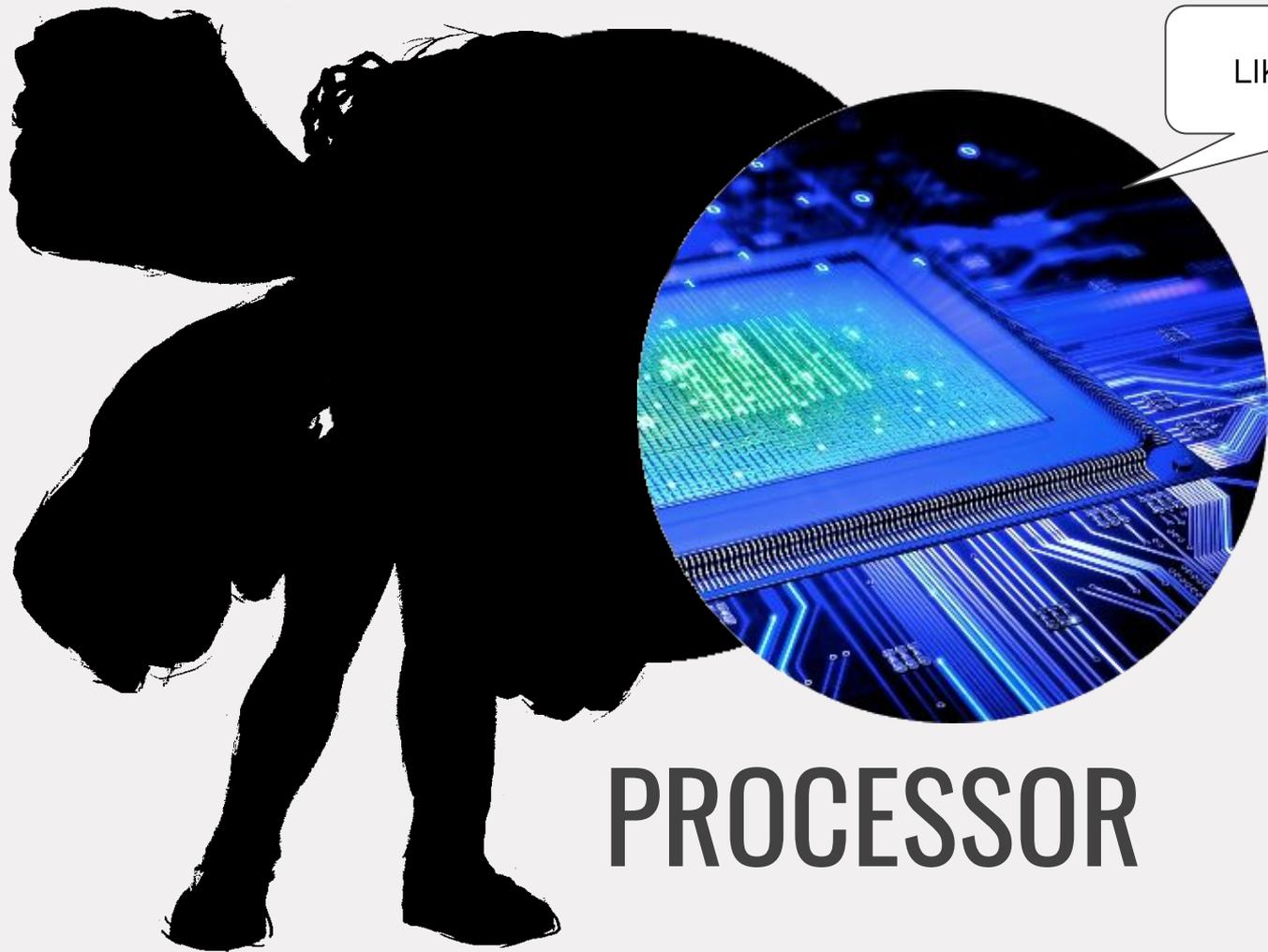
	Nature	Variables acted upon.	Variables receiving results.	Indication of change in the value on any Variable.	Statement of Results
1	$\times$	$1V_2 \times 1V_3$	$1V_4, 1V_5, 1V_6$	$\begin{cases} 1V_2 = 1V_2 \\ 1V_3 = 1V_3 \\ 1V_4 = 2V_4 \\ 1V_5 = 2V_5 \\ 1V_6 = 2V_6 \end{cases}$	$= 2n \dots\dots\dots$
2	$-$	$1V_4 - 1V_1$	$2V_4 \dots\dots\dots$	$\begin{cases} 1V_4 = 2V_4 \\ 1V_1 = 1V_1 \end{cases}$	$= 2n - 1 \dots\dots\dots$
3	$+$	$1V_6 + 1V_1$	$2V_5 \dots\dots\dots$	$\begin{cases} 1V_5 = 2V_5 \\ 1V_1 = 1V_1 \end{cases}$	$= 2n + 1 \dots\dots\dots$
4	$+$	$2V_5 + 2V_4$	$1V_{11} \dots\dots\dots$	$\begin{cases} 2V_5 = 0V_5 \\ 2V_4 = 0V_4 \end{cases}$	$= \frac{2n - 1}{2} \dots\dots\dots$
5	$+$	$1V_{11} + 1V_2$	$2V_{11} \dots\dots\dots$	$\begin{cases} 1V_{11} = 2V_{11} \\ 1V_2 = 1V_2 \end{cases}$	$= \frac{1}{2} \cdot \frac{2n - 1}{2n + 1} \dots\dots\dots$
6	$-$	$0V_{13} - 2V_{11}$	$1V_{13} \dots\dots\dots$	$\begin{cases} 2V_{11} = 0V_{11} \\ 0V_{13} = 1V_{13} \end{cases}$	$= -\frac{1}{2} \cdot \frac{2n - 1}{2n + 1} = A_0 \dots\dots\dots$
	$-$	$1V_3 - 1V_1$	$1V_{10} \dots\dots\dots$	$\begin{cases} 1V_3 = 1V_3 \\ 1V_1 = 1V_1 \end{cases}$	$= n - 1 (= 3) \dots\dots\dots$
		$1V_2 + 0V_7$	$1V_7 \dots\dots\dots$	$\begin{cases} 1V_2 = 1V_2 \\ 0V_7 = 1V_7 \end{cases}$	$= 2 + 0 = 2 \dots\dots\dots$
		$+ 1V_7$	$2V_{11} \dots\dots\dots$	$\begin{cases} 1V_6 = 1V_6 \\ 0V_{11} = 2V_{11} \end{cases}$	$= \frac{2n}{2} = A_1 \dots\dots\dots$
			$1V_{12} \dots\dots\dots$	$\begin{cases} 1V_{21} = 1V_{21} \\ 2V_{11} = 2V_{11} \end{cases}$	$= B_1 \cdot \frac{2n}{2} \dots\dots\dots$
				$\begin{cases} 1V_{12} = 0V_{12} \\ 2V_{12} = 2V_{12} \end{cases}$	$\dots\dots\dots$

PROGRAM



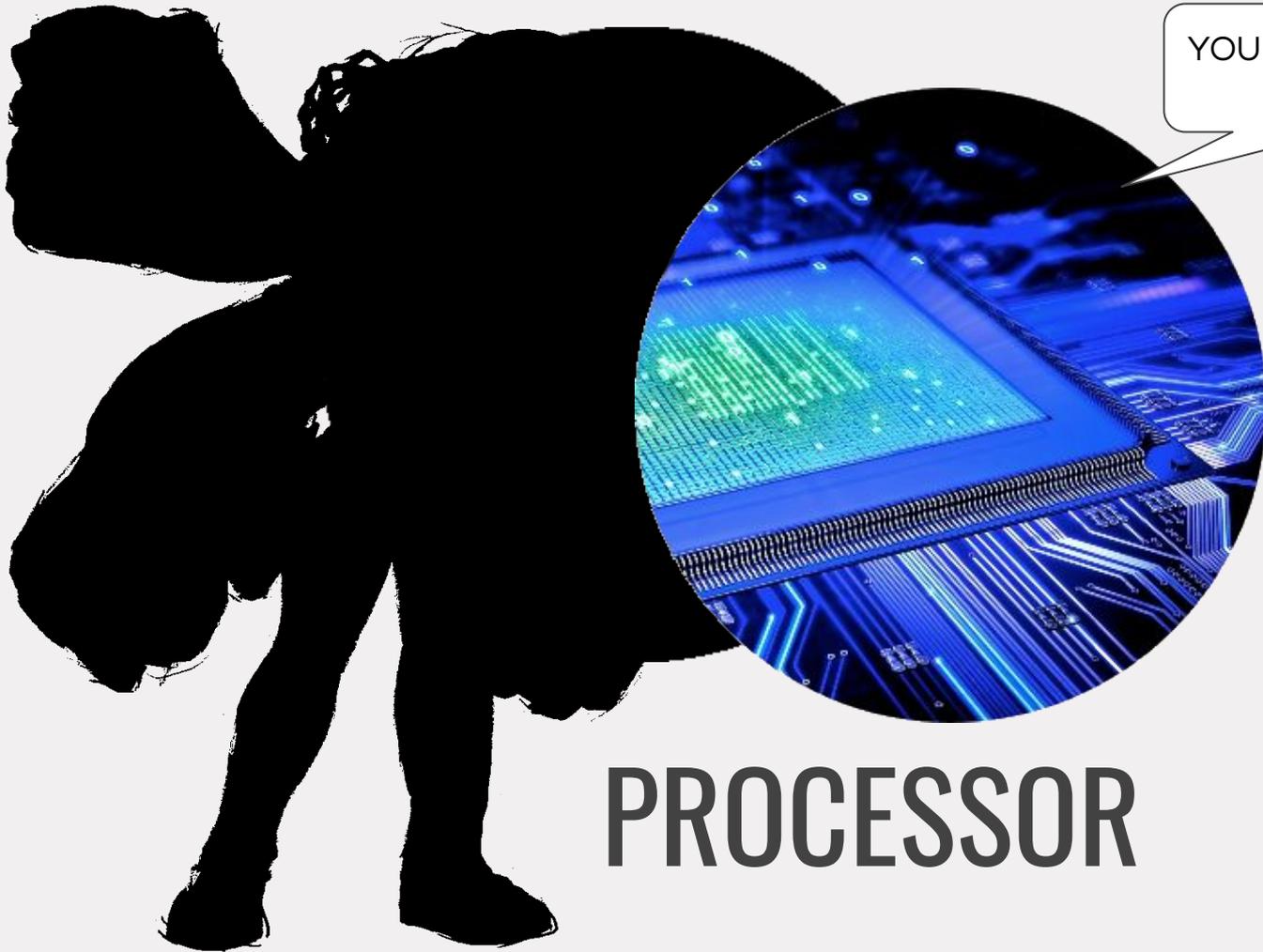
I WILL DO MATH VERY FAST FOR THE THING.

# PROCESSOR



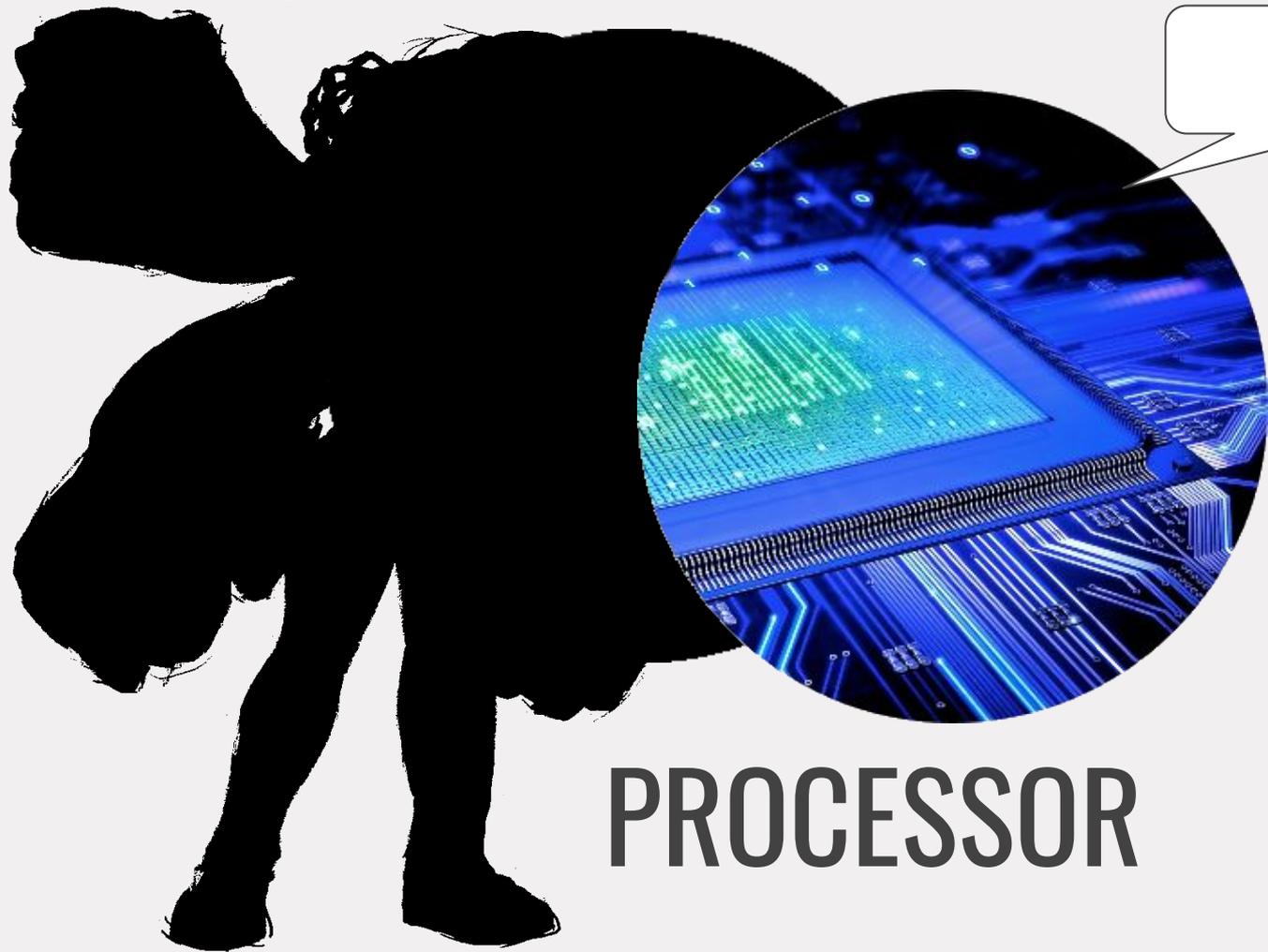
LIKE, VERY FAST.

# PROCESSOR



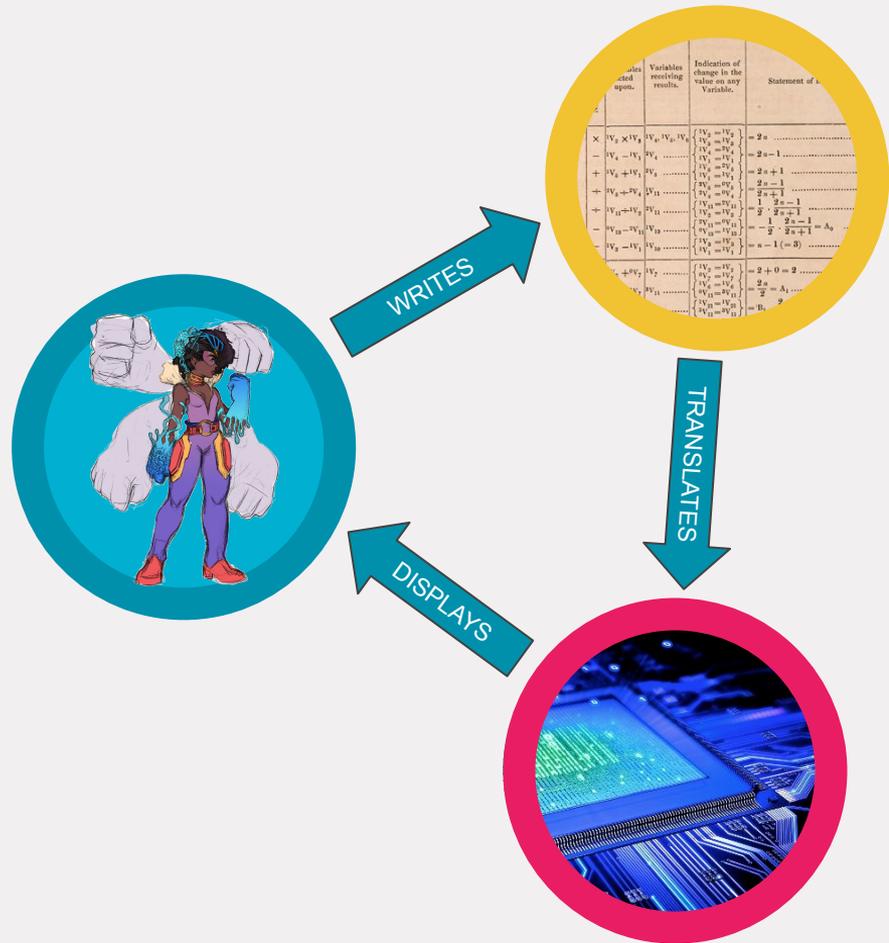
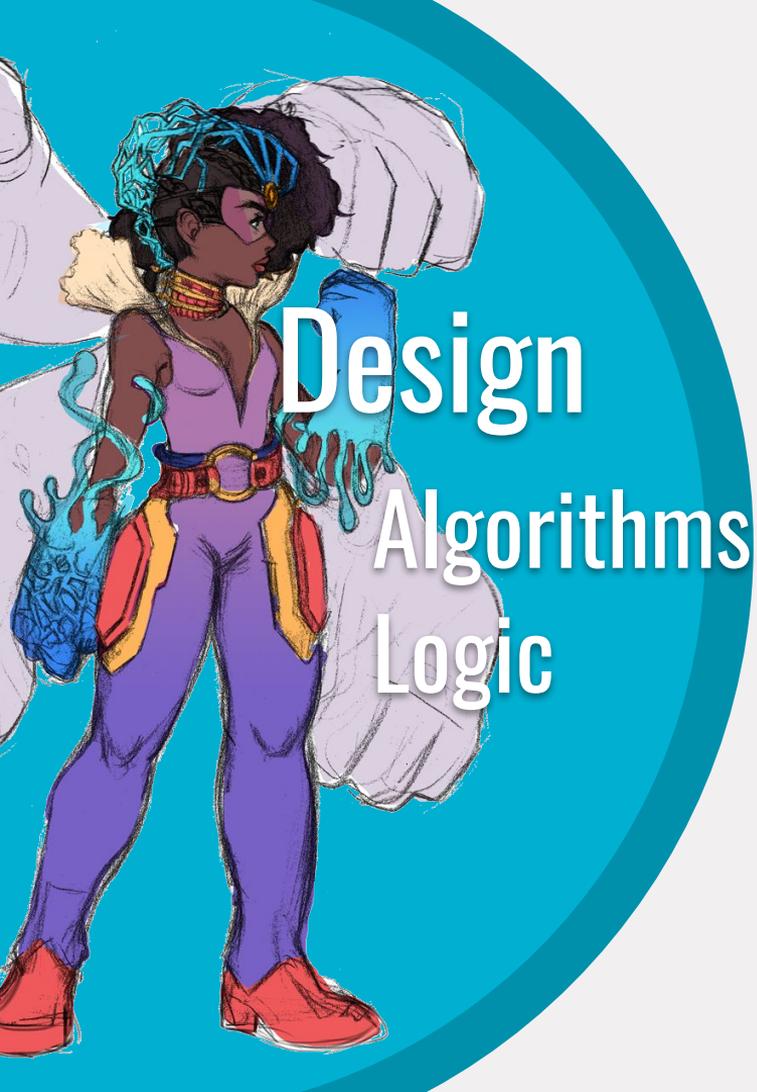
YOU LITERALLY CAN'T  
EVEN.

# PROCESSOR



LITERALLY.

# PROCESSOR





WRITES

	Variables receiving results.	Indication of change in the value on any Variable.	Statement of a
x	$V_2 \times V_3$	$V_2, V_3, V_6$	$V_6 = V_2 \cdot V_3 = 2 \cdot 2 = 4$
-	$V_4 - V_1$	$V_4, V_1$	$V_4 = V_4 - V_1 = 2 - 1 = 1$
+	$V_4 + V_1$	$V_4, V_1$	$V_4 = V_4 + V_1 = 2 + 1 = 3$
+	$V_2 + V_3$	$V_2, V_3$	$V_6 = V_2 + V_3 = 2 + 2 = 4$
+	$V_1 + V_2$	$V_1, V_2$	$V_3 = V_1 + V_2 = 1 + 2 = 3$
-	$V_1 - V_2$	$V_1, V_2$	$V_3 = V_1 - V_2 = 1 - 2 = -1$
-	$V_3 - V_1$	$V_3, V_1$	$V_6 = V_3 - V_1 = 3 - 1 = 2$
-	$V_3 - V_1$	$V_3, V_1$	$V_6 = V_3 - V_1 = 3 - 1 = 2$
+	$V_2$	$V_2$	$V_6 = V_2 = 2$
-	$V_1$	$V_1$	$V_6 = V_6 - V_1 = 4 - 1 = 3$
-	$V_1$	$V_1$	$V_6 = V_6 - V_1 = 4 - 1 = 3$

TRANSLATES



DISPLAYS

# Software Applications System





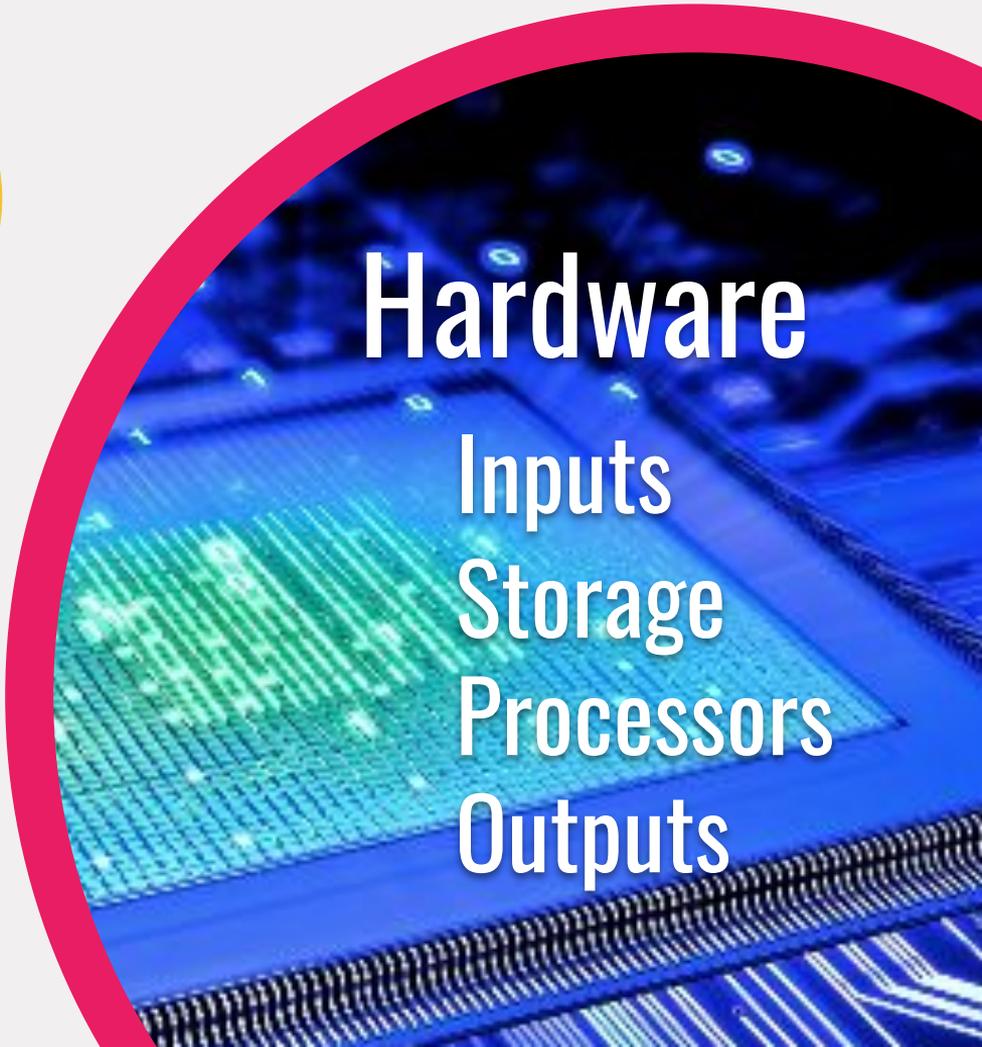
WRITES

	Variables receiving results.	Indication of change in the value on any Variable.	Statement of a
$\times$	$V_2 \times V_3$	$V_2, V_3, V_4$	$V_4 = V_2 \times V_3 = 2 \times 2 = 4$
$-$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = 4 - 2 = 2$
$+$	$V_4 + V_3$	$V_4, V_3$	$V_4 = V_4 + V_3 = 2 + 2 = 4$
$+$	$V_4 + V_2$	$V_4, V_2$	$V_4 = V_4 + V_2 = 4 + 2 = 6$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = 6 - 2 = 4$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = 4 - 2 = 2$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = 2 - 2 = 0$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = 0 - 2 = -2$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -2 - 2 = -4$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -4 - 2 = -6$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -6 - 2 = -8$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -8 - 2 = -10$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -10 - 2 = -12$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -12 - 2 = -14$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -14 - 2 = -16$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -16 - 2 = -18$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -18 - 2 = -20$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -20 - 2 = -22$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -22 - 2 = -24$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -24 - 2 = -26$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -26 - 2 = -28$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -28 - 2 = -30$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -30 - 2 = -32$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -32 - 2 = -34$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -34 - 2 = -36$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -36 - 2 = -38$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -38 - 2 = -40$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -40 - 2 = -42$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -42 - 2 = -44$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -44 - 2 = -46$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -46 - 2 = -48$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -48 - 2 = -50$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -50 - 2 = -52$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -52 - 2 = -54$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -54 - 2 = -56$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -56 - 2 = -58$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -58 - 2 = -60$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -60 - 2 = -62$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -62 - 2 = -64$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -64 - 2 = -66$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -66 - 2 = -68$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -68 - 2 = -70$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -70 - 2 = -72$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -72 - 2 = -74$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -74 - 2 = -76$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -76 - 2 = -78$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -78 - 2 = -80$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -80 - 2 = -82$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -82 - 2 = -84$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -84 - 2 = -86$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -86 - 2 = -88$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -88 - 2 = -90$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -90 - 2 = -92$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -92 - 2 = -94$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -94 - 2 = -96$
$+$	$V_4 - V_3$	$V_4, V_3$	$V_4 = V_4 - V_3 = -96 - 2 = -98$
$+$	$V_4 - V_2$	$V_4, V_2$	$V_4 = V_4 - V_2 = -98 - 2 = -100$

TRANSLATES



DISPLAYS



# Hardware

Inputs

Storage

Processors

Outputs

# A WORKSTATION

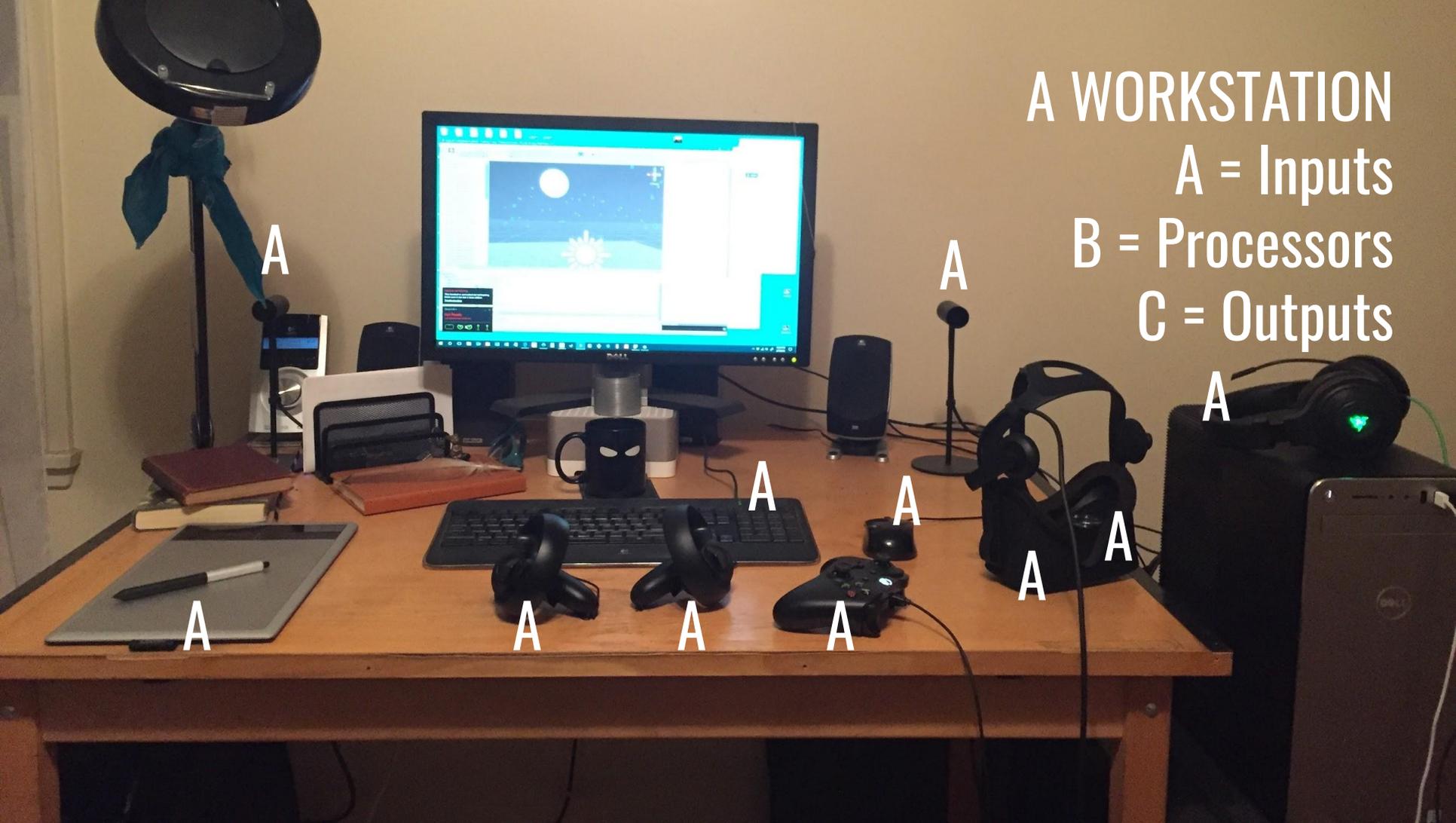


# A WORKSTATION

Inputs  
Processors  
Outputs







# A WORKSTATION

A = Inputs

B = Processors

C = Outputs

A

A

A

A

A

A

A

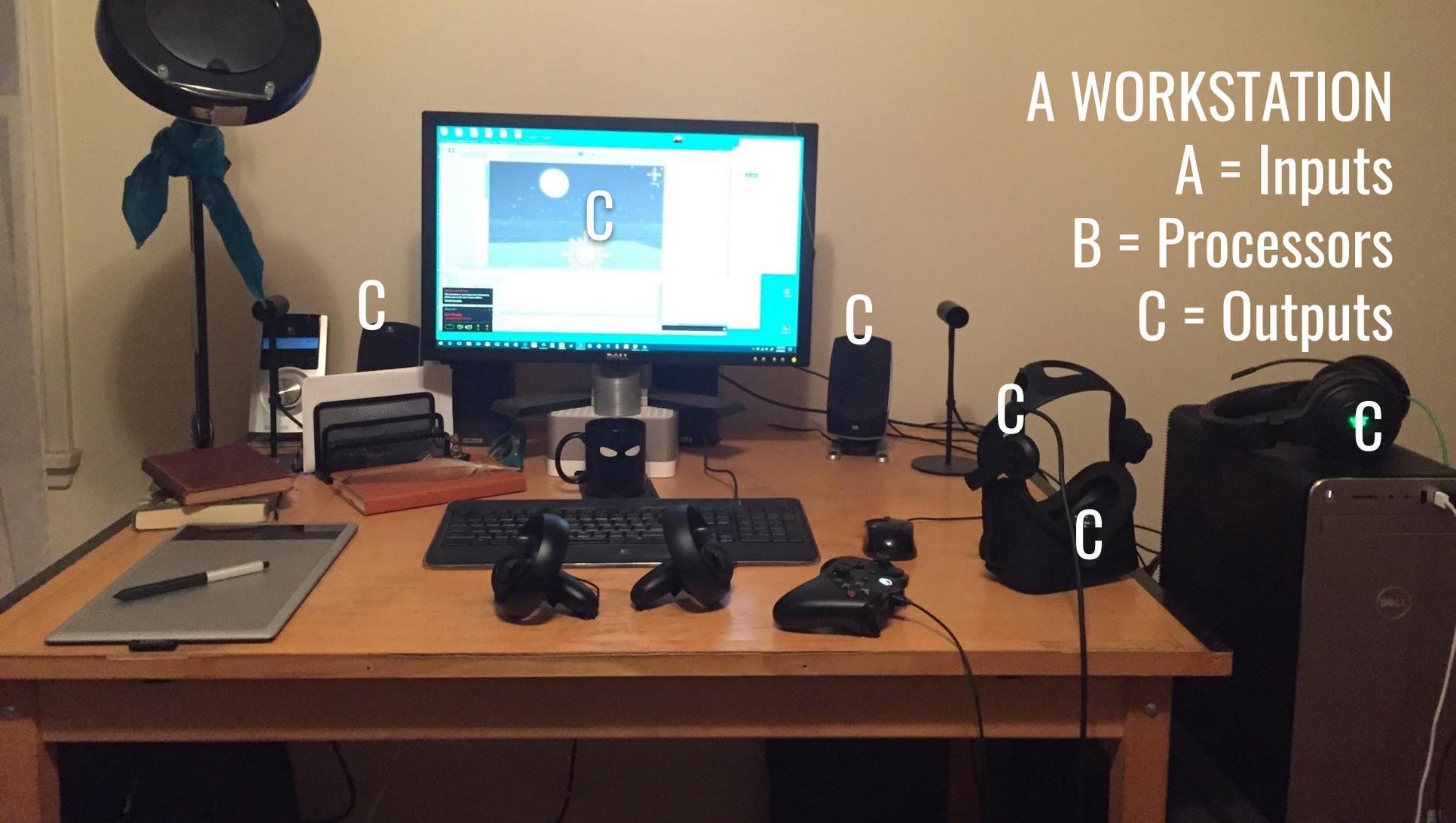
A

A

A

A





# A WORKSTATION

A = Inputs

B = Processors

C = Outputs

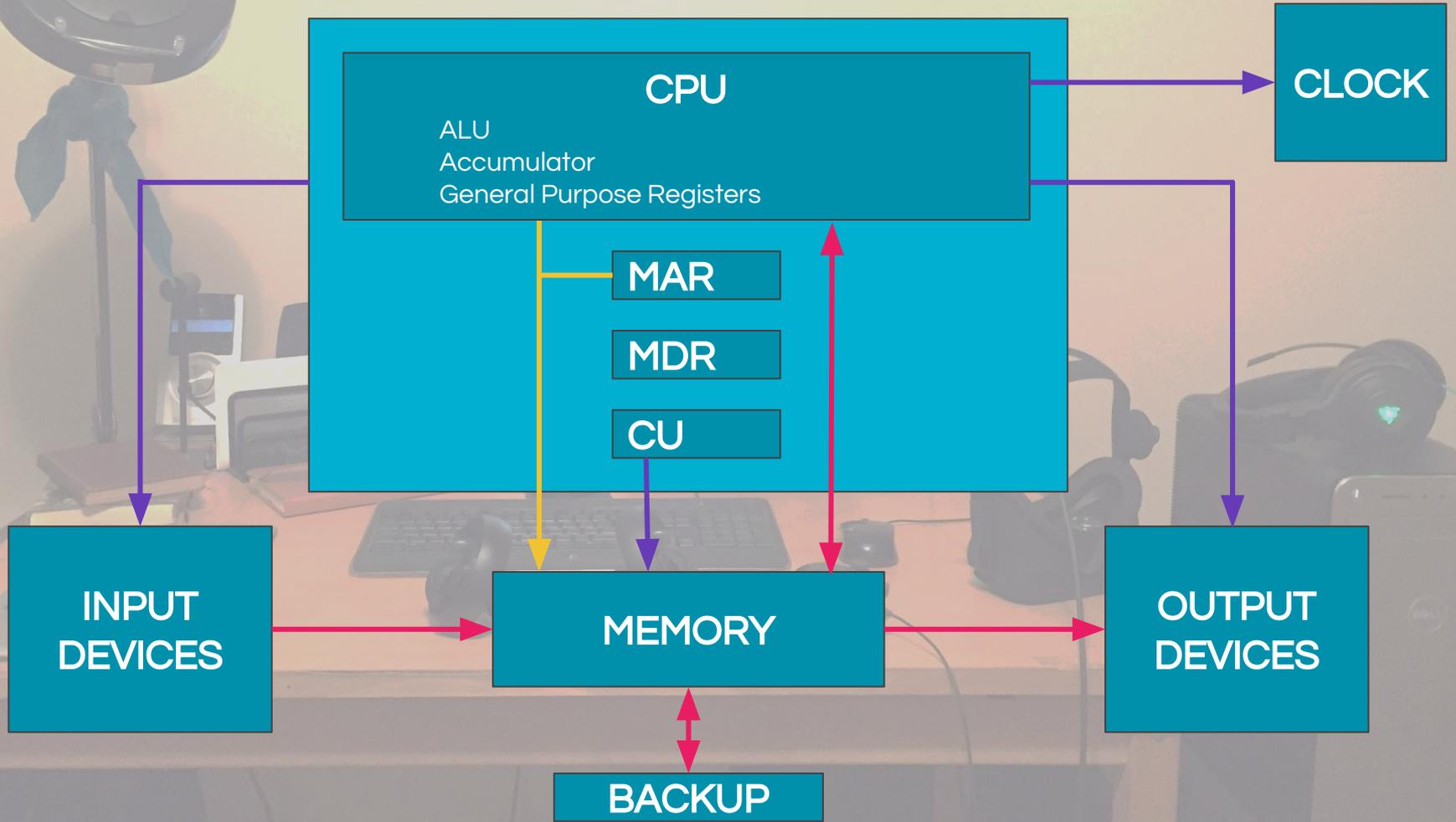
C

C

C

C

C



# Constraints... Let's Break It Down

- You write programs in a **High Level Language**
- the **Compiler** translates that
- into the **Low Level, Assembly Language**
- the **Assembler** translates that
- into **Machine Language** (binary!)
- the **Control Unit** interprets that
- for the **Microarchitecture**
- where the **Microsequencer** interprets the binary
- for the **Logic-Design** at the **Device Level**
- made up of **Semiconductors / Silicon Transistors**

...and **CONSTRAINED** by the properties of **Atoms**,  
**Electrons**, and **Quantum Dynamics!**

Compiler  
Assembler  
Control Unit  
Microsequencer  
Frosting  
More frosting

High Level Language

Low Level Language

Binary

Microarchitecture

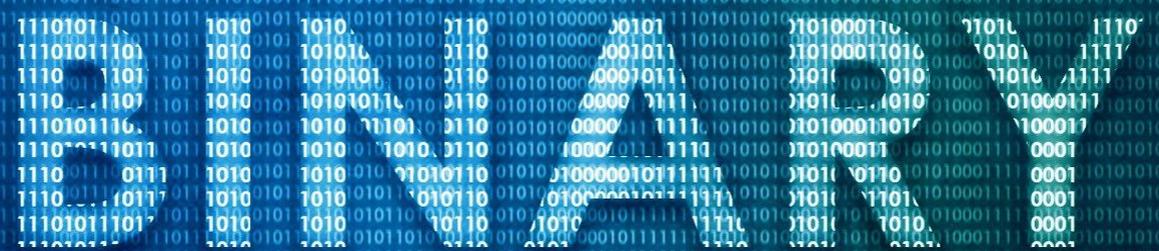
Logic-Design at Device

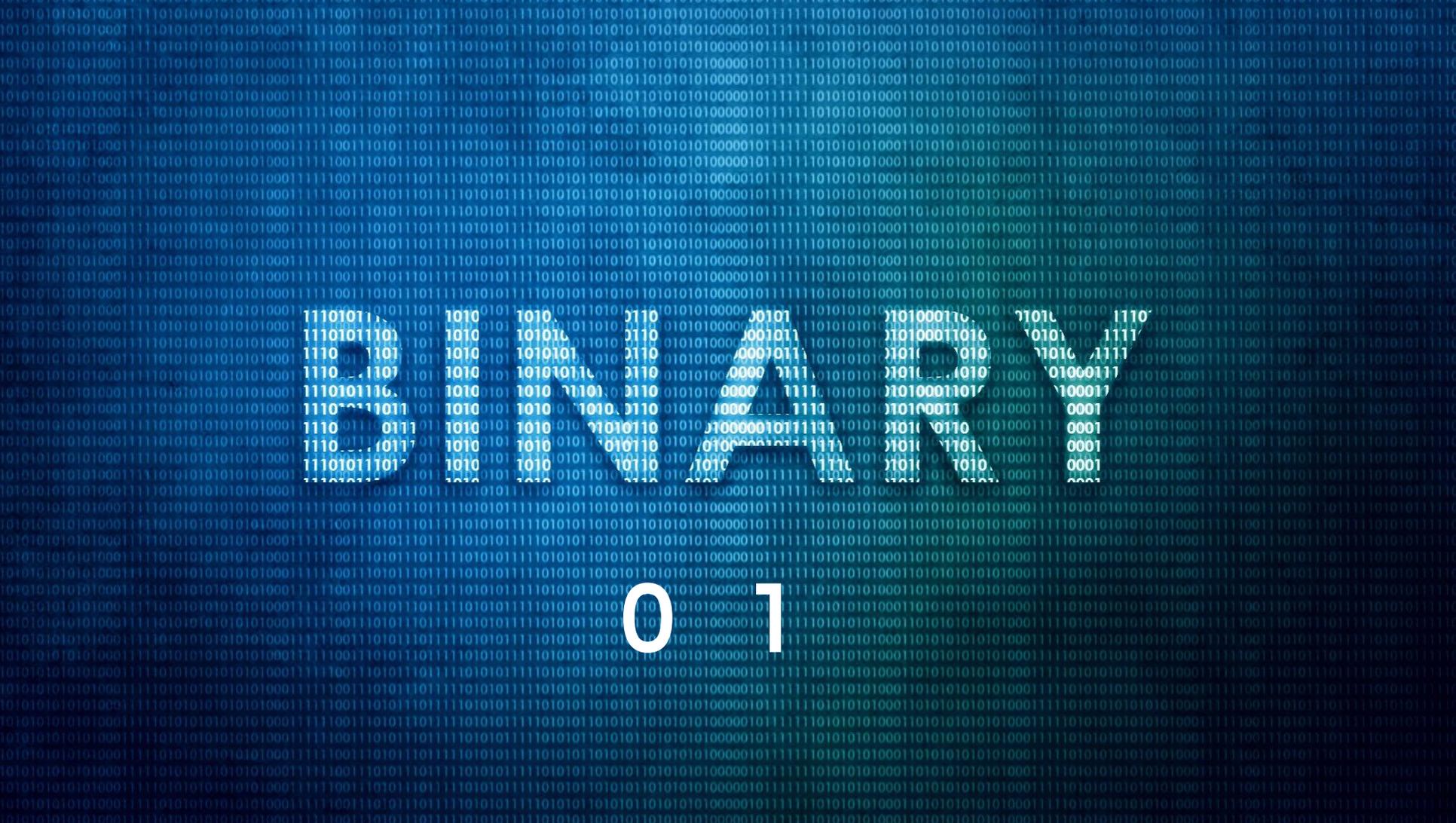
Semiconductors /  
Transistors

Atoms / Electrons,  
Quantum Dynamics



What is





Closed Open





Off On

# A computer is:

*A binary digital machine*

- Basic unit of information is the binary digit, or bit
- Values with more than two states require multiple bits.
  - Two bits have four possible states: 00, 01, 10, 11
  - Three bits have eight possible states: 000, 001, 010, 011, 100, 101, 110, 111
  - A collection of  $n$  bits has  $2^n$  possible states.



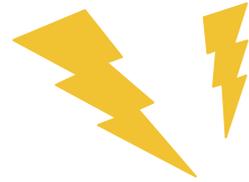
# What do 0 or 1 mean?

- The computer is an electronic machine
- Bits (which can have a value of 0 or 1) are...

---

# What do 0 or 1 mean?

- The computer is an electronic machine
- Bits (which can have a value of 0 or 1) are...



---

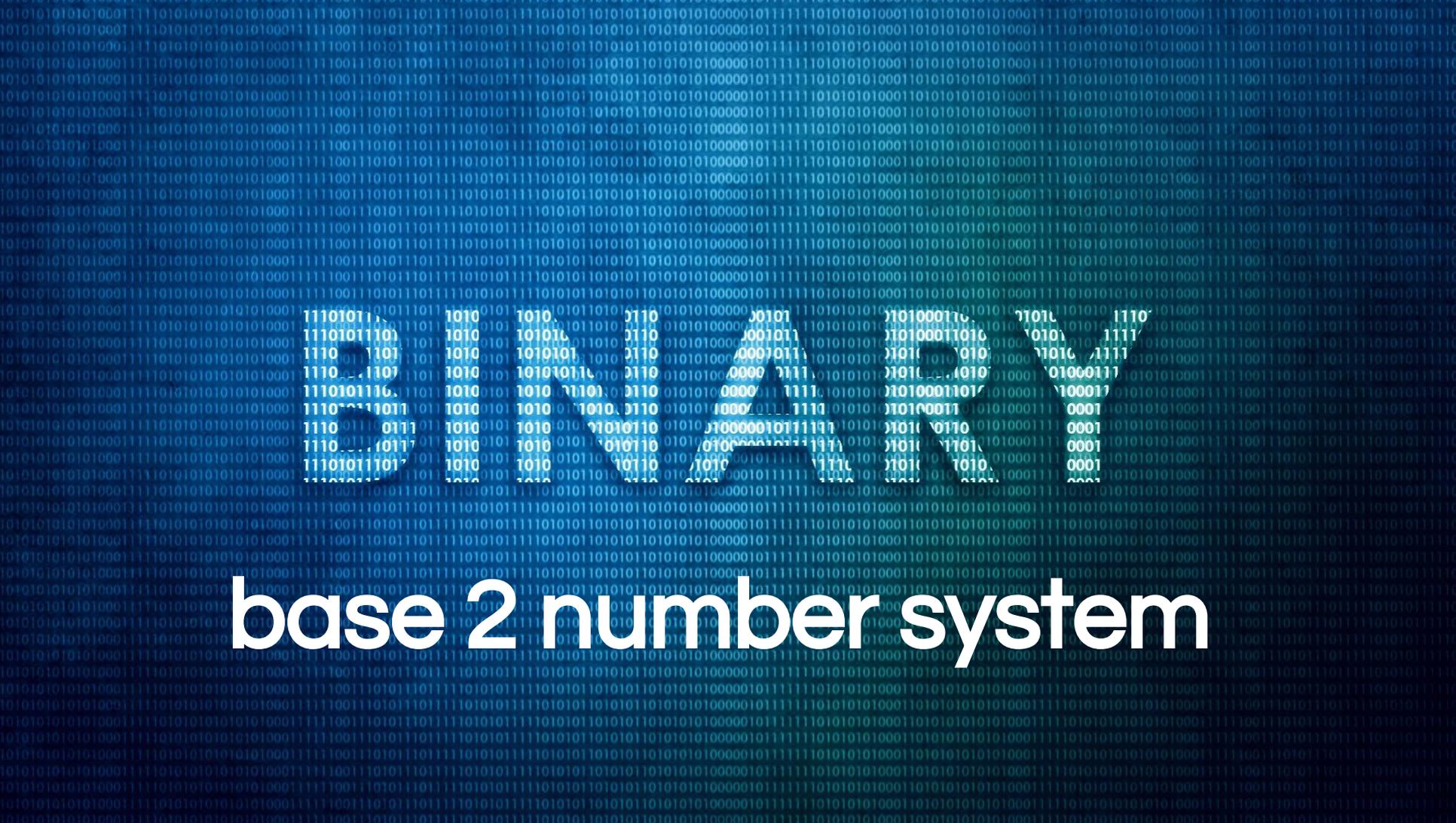
# What do 0 or 1 mean?

- The computer is an electronic machine
- Bits (which can have a value of 0 or 1) are...



- Voltage in one of two states:
  - **presence** of voltage – we call this state “1”
  - **absence** of voltage – we call this state “0”





# base 2 number system



How does it work?

# BINARY ARITHMETIC

- Base-2 Addition
  - just like base-10!
  - add from right to left, carrying

$$\begin{array}{r} 11 \\ + 1 \\ \hline \end{array}$$

BASE 10

$$\begin{array}{r} 11 \\ + 1 \\ \hline \end{array}$$

BASE 2

# BINARY ARITHMETIC

- Base-2 Addition
  - just like base-10!
  - add from right to left, carrying

$$\begin{array}{r} 11 \\ + 1 \\ \hline \end{array}$$

BASE 10

$$\begin{array}{r} 11 \\ + 1 \\ \hline \end{array}$$

BASE 2

# BINARY ARITHMETIC

- Base-2 Addition
  - just like base-10!
  - add from right to left, carrying

$$\begin{array}{r} 10 \\ + 1 \\ \hline 11 \end{array}$$

**BASE 10**

$$\begin{array}{r} 10 \\ + 1 \\ \hline \end{array}$$

BASE 2

# BINARY ARITHMETIC

- Base-2 Addition
  - just like base-10!
  - add from right to left, carrying

$$\begin{array}{r} 11 \\ + 1 \\ \hline 11 \end{array}$$

BASE 10

$$\begin{array}{r} 11 \\ + 1 \\ \hline 100 \end{array}$$

BASE 2

# BINARY ARITHMETIC

- Base-2 Addition

- just like base-10!
- add from right to left, carrying

	$\begin{array}{r} 11 \\ + 1 \\ \hline 11 \end{array}$	$\begin{array}{r} \curvearrowright \curvearrowright \\ 11 \\ + 1 \\ \hline 100 \end{array}$
	BASE 10	BASE 2

# BINARY ARITHMETIC

$$\begin{array}{r} 10010 \\ + 1001 \\ \hline \end{array}$$

$$\begin{array}{r} 10010 \\ + 1011 \\ \hline \end{array}$$

$$\begin{array}{r} 1111 \\ + 1 \\ \hline \end{array}$$

# BINARY ARITHMETIC

$$\begin{array}{r} 10010 \\ + 1001 \\ \hline \end{array}$$

$$\begin{array}{r} 10010 \\ + 1011 \\ \hline \end{array}$$

$$\begin{array}{r} 1111 \\ + 1 \\ \hline \end{array}$$

# BINARY ARITHMETIC

$$\begin{array}{r} 10010 \\ + 1001 \\ \hline 11011 \end{array}$$

$$\begin{array}{r} 10010 \\ + 1011 \\ \hline \end{array}$$

$$\begin{array}{r} 1111 \\ + 1 \\ \hline \end{array}$$

# BINARY ARITHMETIC

$$\begin{array}{r} 10010 \\ + 1001 \\ \hline 11011 \end{array}$$

$$\begin{array}{r} 10010 \\ + 1011 \\ \hline \end{array}$$

$$\begin{array}{r} 1111 \\ + 1 \\ \hline \end{array}$$

# BINARY ARITHMETIC

$$\begin{array}{r} 10010 \\ + 1001 \\ \hline 11011 \end{array}$$

$$\begin{array}{r} 10010 \\ + 1011 \\ \hline 11101 \end{array}$$



$$\begin{array}{r} 1111 \\ + 1 \\ \hline \end{array}$$

# BINARY ARITHMETIC

$$\begin{array}{r} 10010 \\ + 1001 \\ \hline 11011 \end{array}$$

$$\begin{array}{r} 10010 \\ + 1011 \\ \hline 11101 \end{array}$$

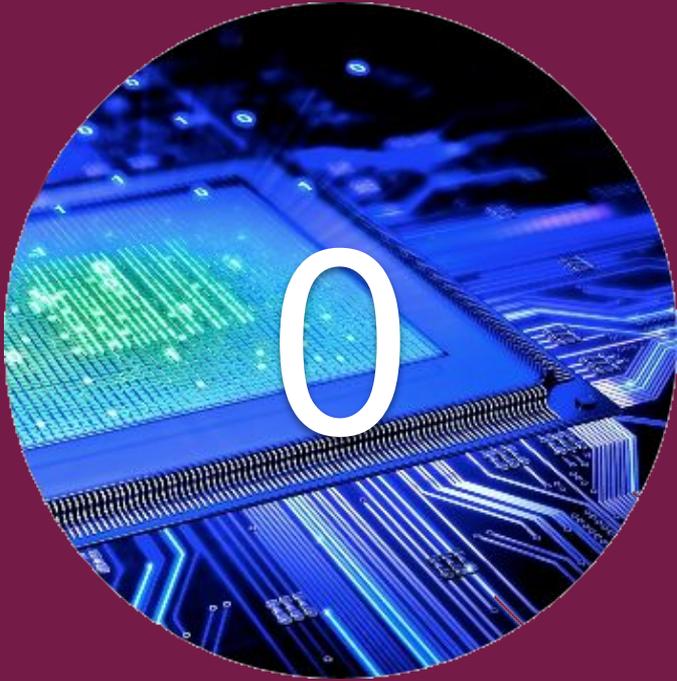
$$\begin{array}{r} 1111 \\ + \quad 1 \\ \hline \end{array}$$

# BINARY ARITHMETIC

$$\begin{array}{r} 10010 \\ + 1001 \\ \hline 11011 \end{array}$$

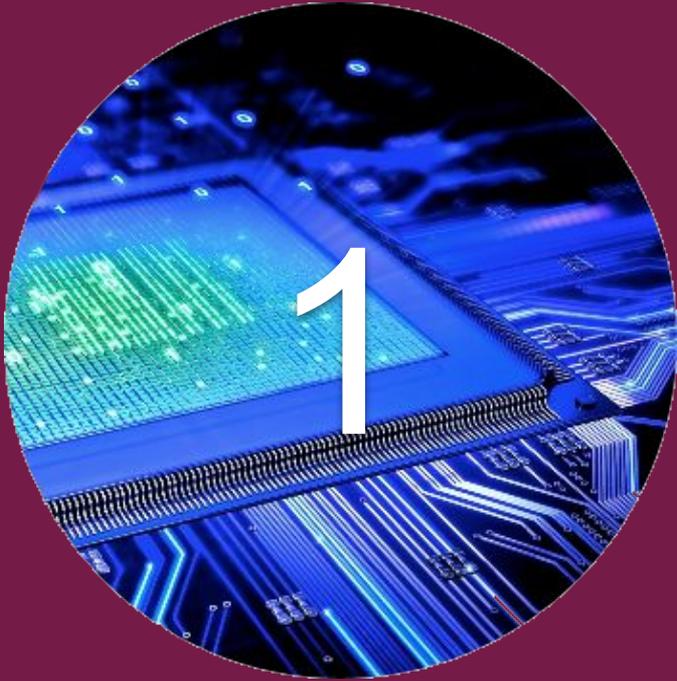
$$\begin{array}{r} 10010 \\ + 1011 \\ \hline 11101 \end{array}$$

$$\begin{array}{r} \overset{\curvearrowright}{\curvearrowright}{\curvearrowright}{\curvearrowright} \\ 1111 \\ + \quad 1 \\ \hline 10000 \end{array}$$



**A BIT**





**A BIT**

---



**A BYTE**

**8 BITS**





# A BYTE

How many variations are possible?

---



32-bit OS

64-bit OS

---

# BINARY NUMBERS 0 to 7

$2^2$	$2^1$	$2^0$	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

# WEIGHTED POSITIONAL NOTATION

*most weight* ← → *least weight*

329

$10^2$   $10^1$   $10^0$

*most weight* ← → *least weight*

101

$2^2$   $2^1$   $2^0$

## Decimal

- "329" the "3" is worth 300 because of its position
- What is the "9" worth?

## Binary

- "101" the first "1" is worth 4 because of its position
- What is the second "1" worth?

# HEXADECIMAL NOTATION

- Also "base 16" or "hex"
- Four bits can represent 16 positions
- $2^4 = 2 \times 2 \times 2 \times 2$

Colors are sometimes represented by hex

#000000 (black) and #FFFFFF (white)

Binary	Hex	Decimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

# HEXADECIMAL NOTATION

#663333

#009999

#FFCC33

- Also "base 16" or "hex"
- Four bits can represent 16 positions
- $2^4 = 2 \times 2 \times 2 \times 2$

Colors are sometimes represented by hex

#000000 (black) and #FFFFFF (white)

Binary	Hex	Decimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15



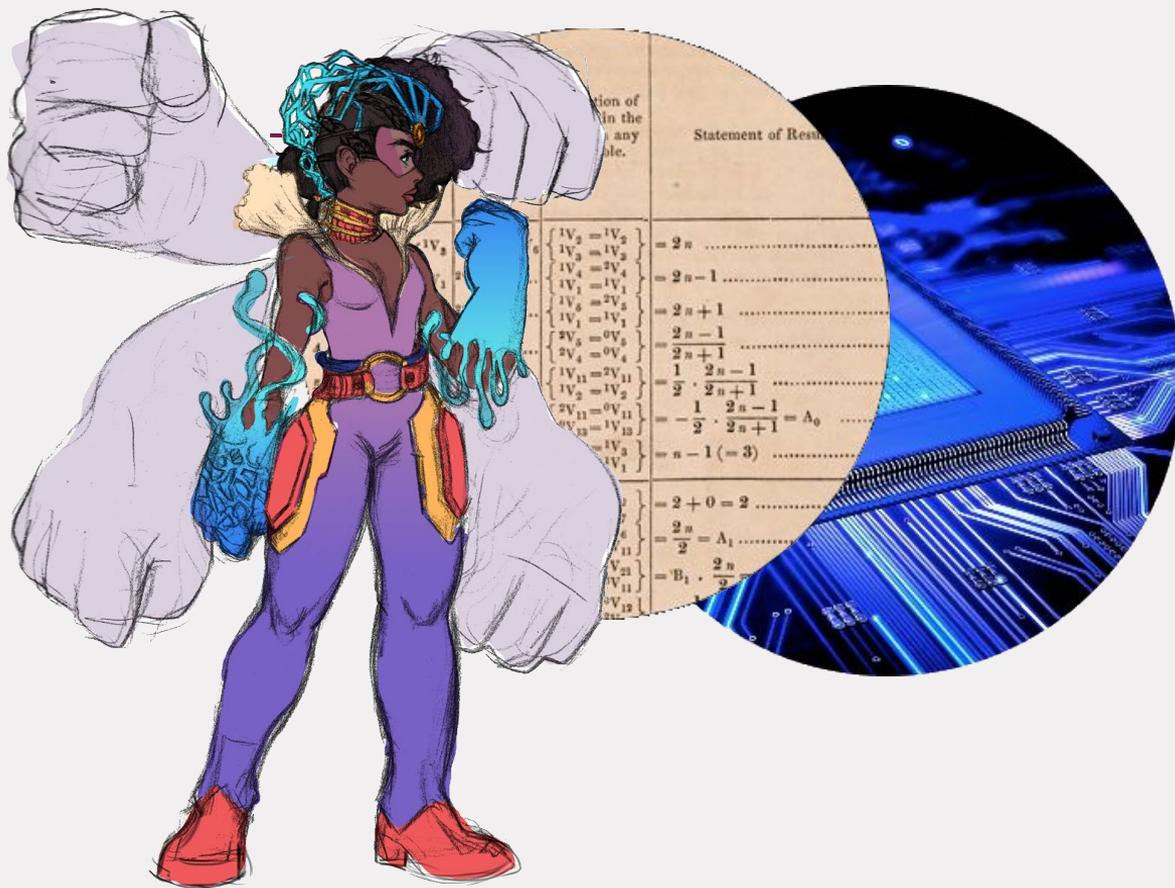
# EXERCISE

$$\begin{array}{r} 10111 \\ + 111 \\ \hline ? \end{array}$$



# EXERCISE

$$\begin{array}{r} 10111 \\ + 00111 \\ \hline \end{array} ?$$



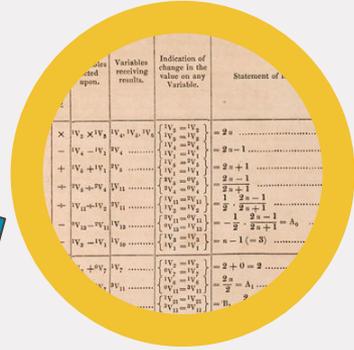
# RELATIONSHIP

- Have idea/problem
- Design a solution
- Write code
  - Debug code
- Solve problem
- Save world - have fun - make \$\$

**Programmer**  
Writes Code



WRITES



**Program**  
Compiles Code  
into Binary

TRANSLATES



**Processor**  
Processes  
and displays  
results

DISPLAYS

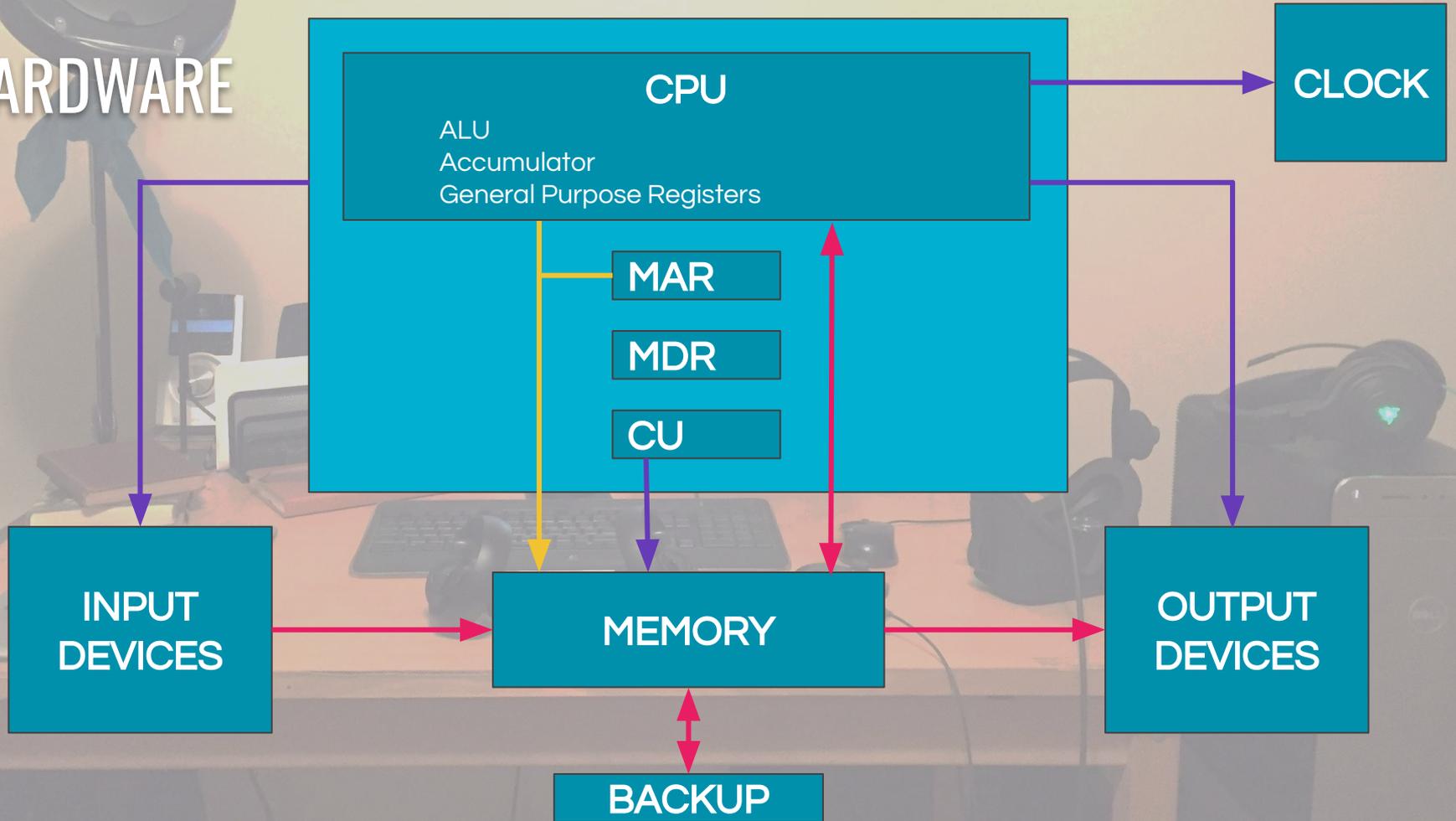
# HARDWARE

---

- Inputs
- Storage
- Processor
- Outputs
  
- Why? → Binary



# HARDWARE



NEXT UP

Programming Languages

