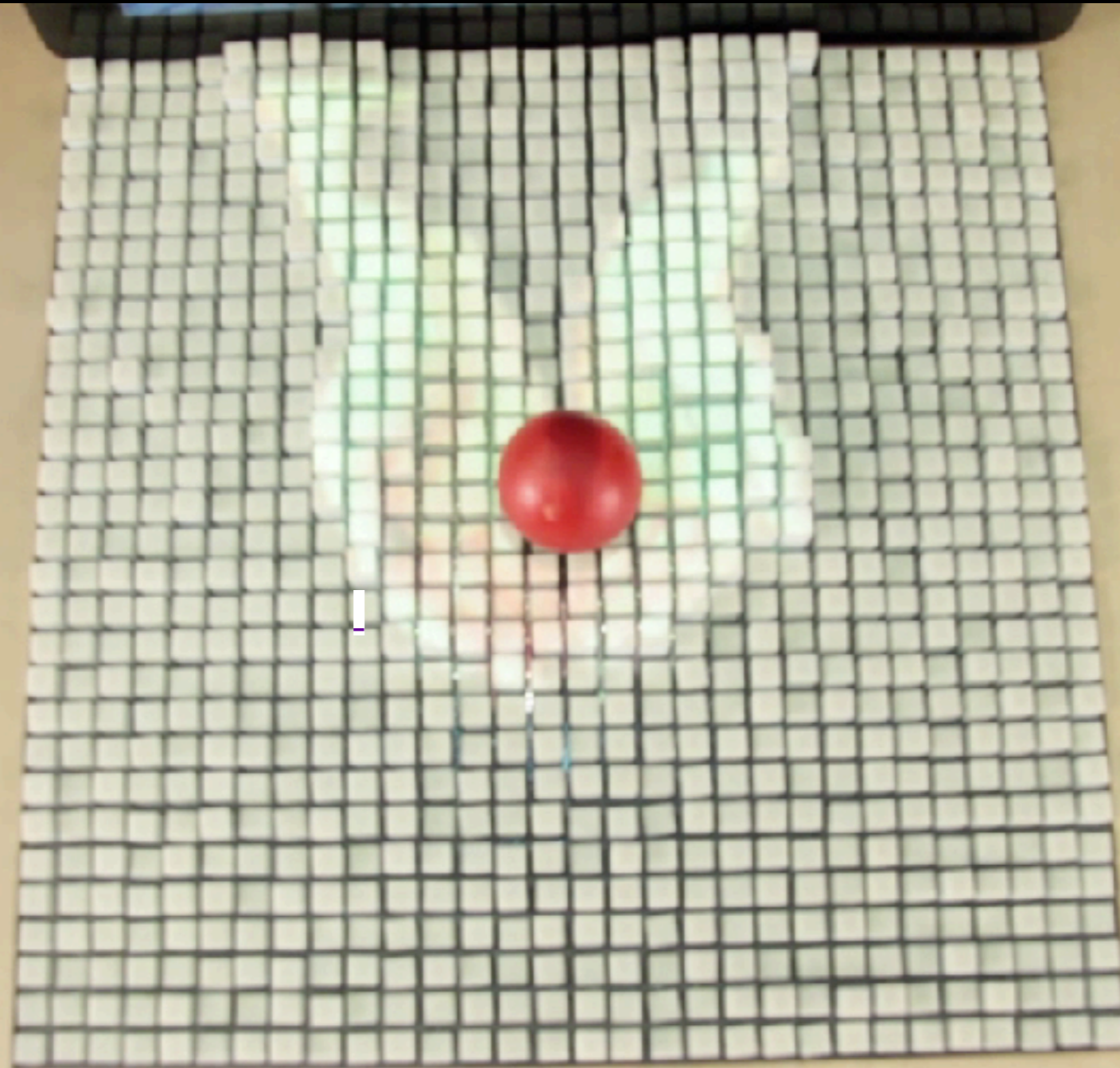




Bits and Atoms

Luke Franzke - 2017

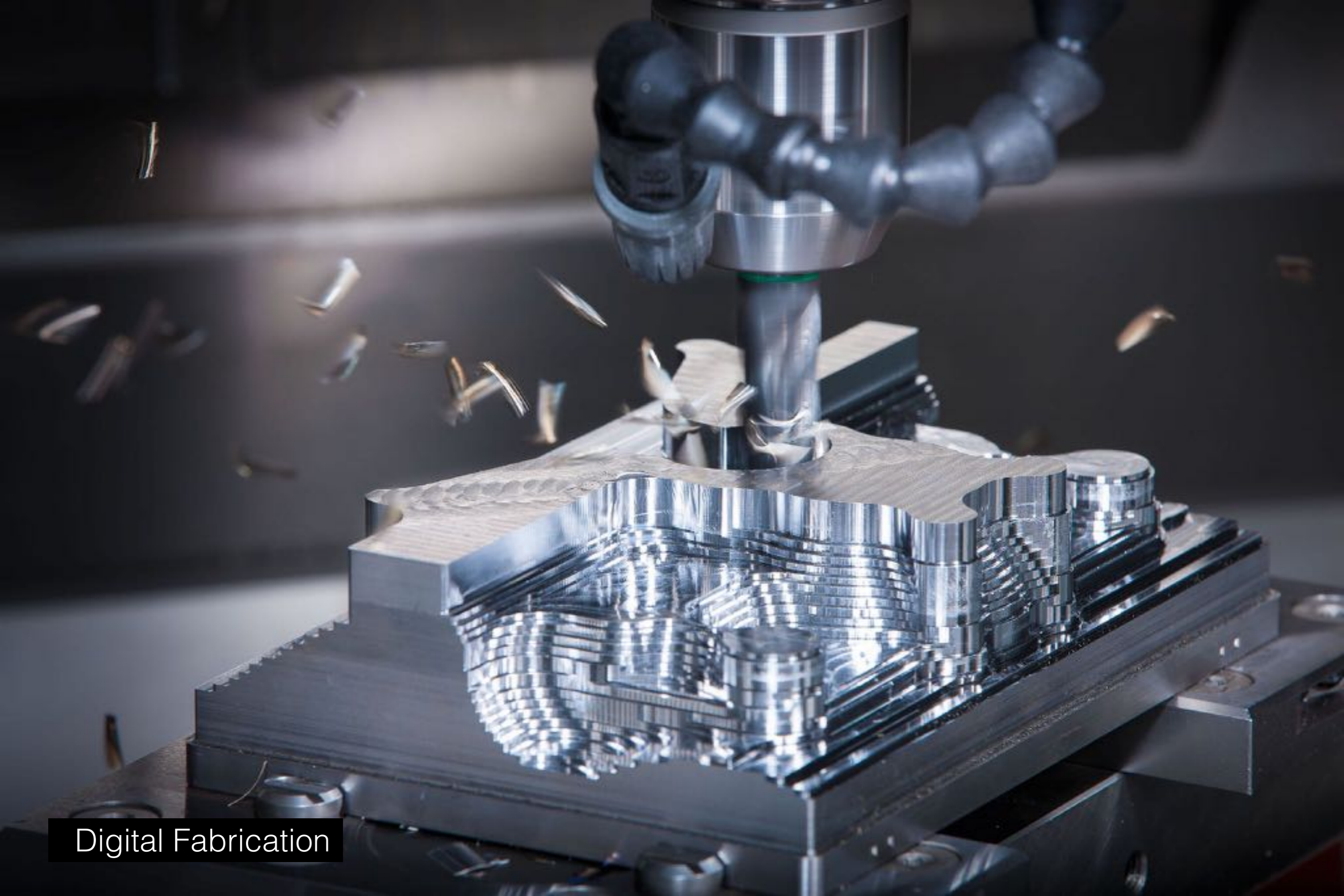
Creative Code





Interactive Experience





Digital Fabrication







Generative Design

Iris van Herpen and Neri Oxman



Generative Design

Nervous System





The impact of code in design



Before Parametric Design

Reproducible curves



Before Parametric Design

Splines: Reproducing curves at different scales

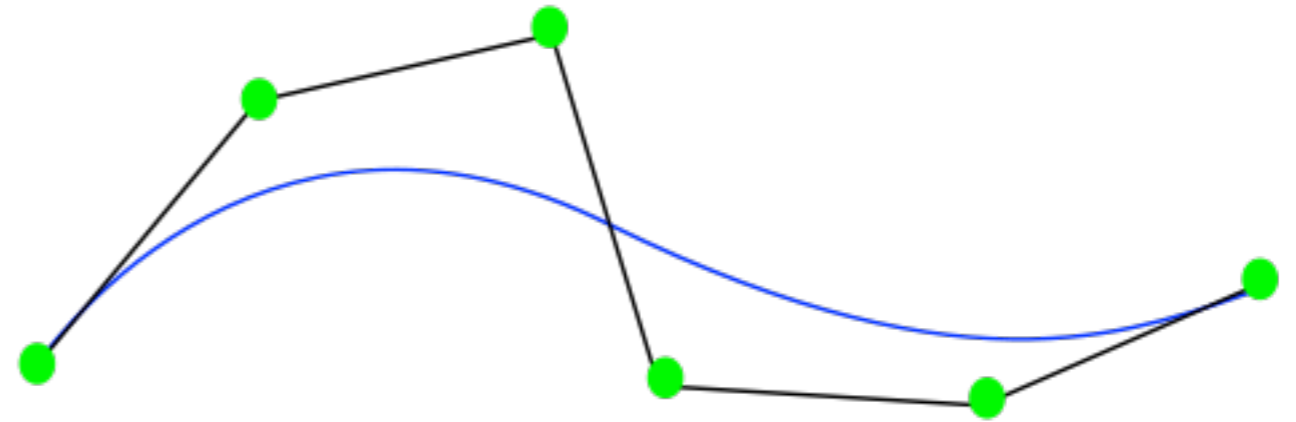
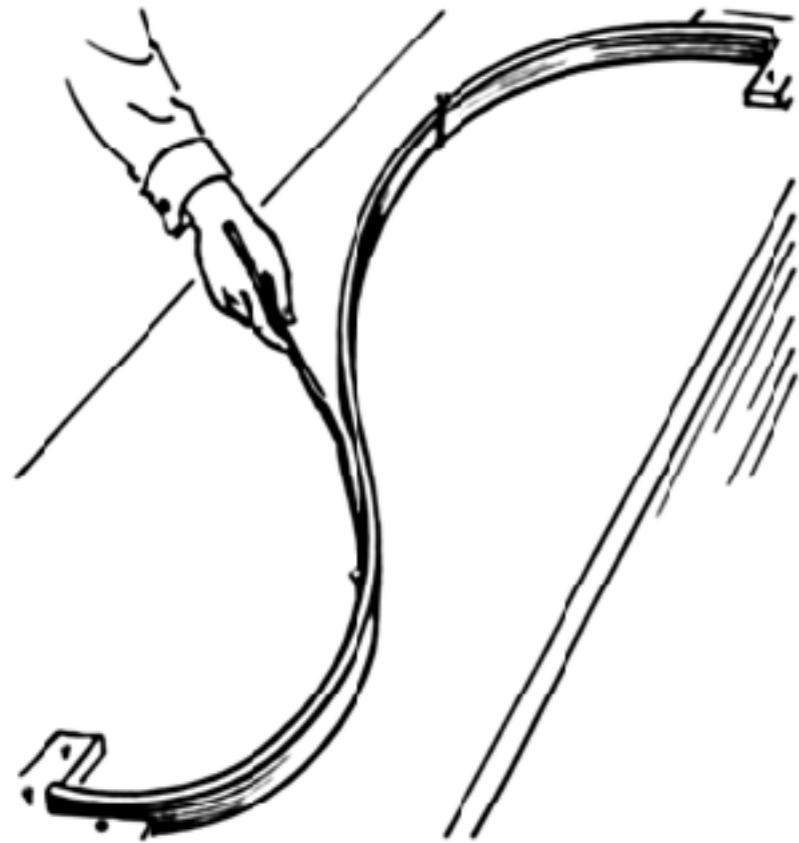
D	Dec 1/82	F	U
C	Dec 20/80	M	R
E	Mar 21/81	K	C
A	Oct 9/79	M	



Before Parametric Design
Jørn Utzon sketches



Before Parametric Design



Splines: Analog to Mathematical Model (Bezier Curve)

Getting to the Basics

- What is Digital?
- What is Analog?

Analog and Digital Signals

- An analog signal is a continuous wave form that changes smoothly over time
- A digital signal is discrete. It can have only a limited number of defined values, often as simple as 1 and 0



a. Analog signal



b. Digital signal



Analog Technology

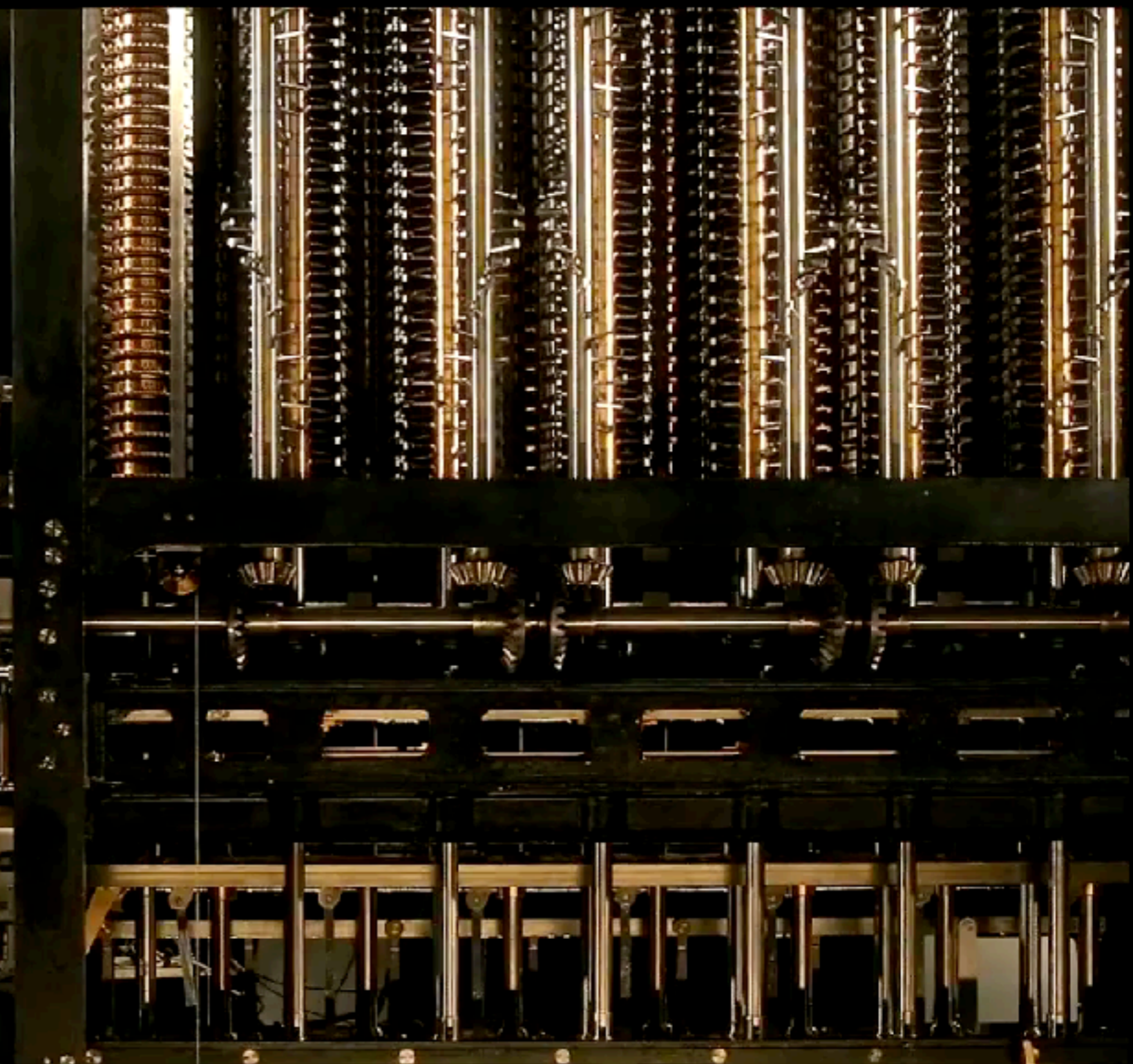
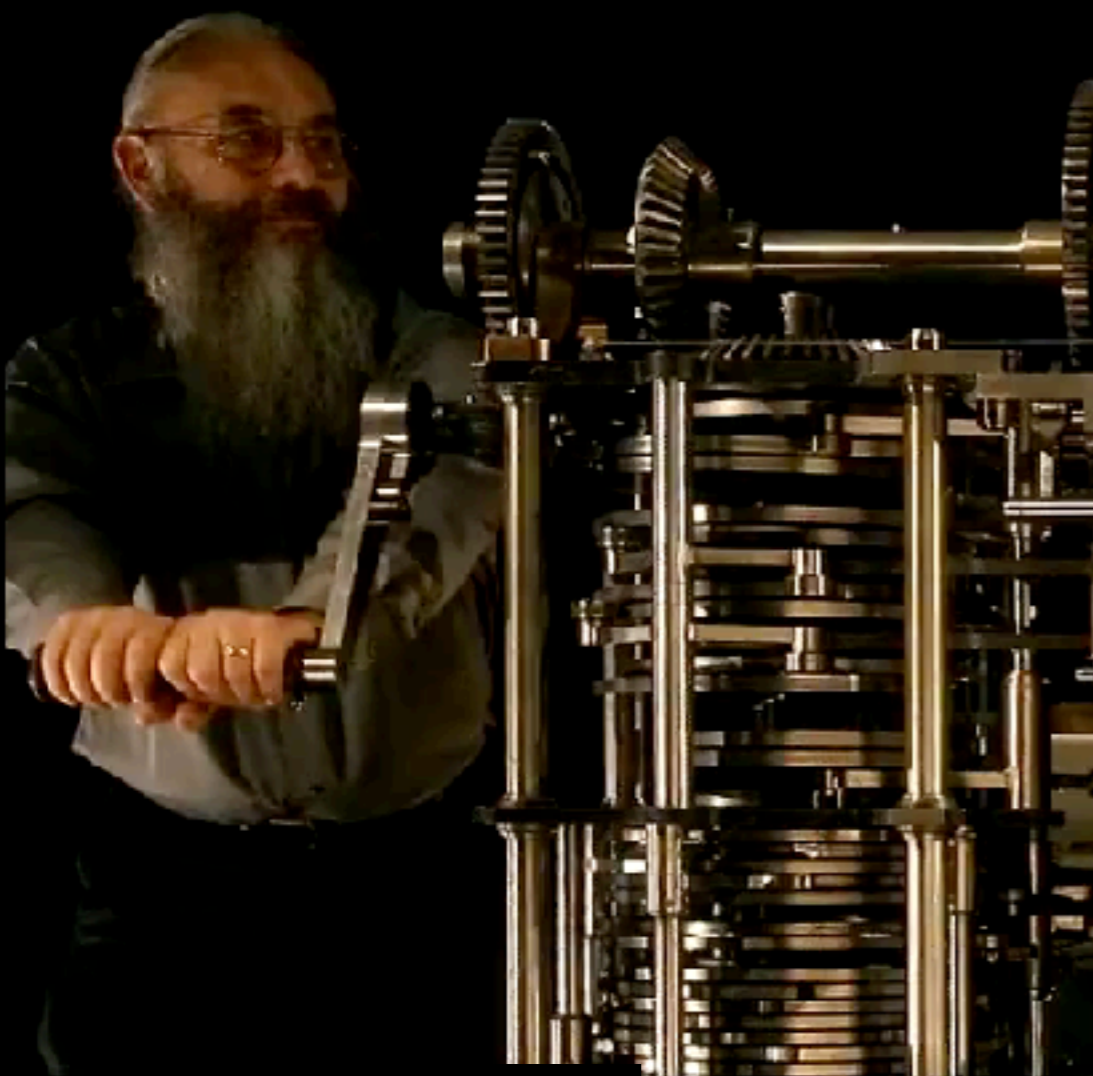


Jacquard Loom



Jacquard Loom

xRez Studio



Babbage engine

Number of Operation.	Nature of Operation.	Variables used, open.	Variables receiving results.	Indication of change in the value of any Variable.	Statement of Results.	Data										Working Variables.									
						1V_1 ○ 9 0 1	1V_2 ○ 9 0 2	1V_3 ○ 9 0 4	1V_4 ○ 9 0 9	1V_5 ○ 9 0 0	1V_6 ○ 9 0 0	1V_7 ○ 9 0 0	1V_8 ○ 9 0 0	1V_9 ○ 9 0 0	${}^1V_{10}$ ○ 9 0 0	${}^1V_{11}$ ○ 9 0 0									
1	×	${}^1V_2 \times {}^1V_3$	${}^1V_{12}$	${}^1V_2 = {}^1V_3$	$-2n$	1	2	n	$2n$	$2n$	$2n$														
2	-	${}^1V_4 - {}^1V_1$	1V_4	${}^1V_4 = {}^1V_1$	$-2n - 1$	1			$2n - 1$																
3	+	${}^1V_5 + {}^1V_1$	1V_5	${}^1V_5 = {}^1V_1$	$-2n + 1$	1				$2n + 1$															
4	+	${}^1V_6 + {}^1V_2$	1V_6	${}^1V_6 = {}^1V_2$	$-\frac{2n-1}{2n+1}$				0	0														$\frac{2n-1}{2n+1}$	
5	+	${}^1V_7 + {}^1V_2$	1V_7	${}^1V_7 = {}^1V_2$	$-\frac{1}{2} \frac{2n-1}{2n+1}$		2																	$\frac{1}{2} \frac{2n-1}{2n+1}$	
6	-	${}^1V_8 - {}^1V_2$	1V_8	${}^1V_8 = {}^1V_2$	$-\frac{1}{2} \frac{2n-1}{2n+1} = A_0$																			0	
7	-	${}^1V_8 - {}^1V_1$	1V_8	${}^1V_8 = {}^1V_1$	$-n - 1 (-3)$	1		n																$n-1$	
8	+	${}^1V_8 + {}^1V_2$	1V_8	${}^1V_8 = {}^1V_2$	$-2 + 0 = -2$		2																		
9	+	${}^1V_8 + {}^1V_7$	1V_8	${}^1V_8 = {}^1V_7$	$-\frac{2n}{2} = A_1$						$2n$	2												$\frac{2n}{2} = A_1$	
10	×	${}^1V_8 \times {}^1V_7$	${}^1V_{12}$	${}^1V_{12} = {}^1V_7$	$-B_1 \cdot \frac{2n}{2} = B_1 A_1$																			$\frac{2n}{2} = A_1$	
11	+	${}^1V_{12} + {}^1V_8$	${}^1V_{12}$	${}^1V_{12} = {}^1V_8$	$-\frac{1}{2} \frac{2n-1}{2n+1} + B_1 \cdot \frac{2n}{2}$																				
12	-	${}^1V_{12} - {}^1V_1$	${}^1V_{12}$	${}^1V_{12} = {}^1V_1$	$-n - 2 (-2)$	1																		$n-2$	
13	-	${}^1V_{12} - {}^1V_1$	1V_4	${}^1V_4 = {}^1V_1$	$-2n - 1$	1					$2n - 1$														
14	+	${}^1V_4 + {}^1V_7$	1V_4	${}^1V_4 = {}^1V_7$	$-3 + 1 = -3$	1																			
15	+	${}^1V_4 + {}^1V_7$	1V_4	${}^1V_4 = {}^1V_7$	$-\frac{2n-1}{3}$						$2n - 1$	3	$\frac{2n-1}{3}$												
16	×	${}^1V_4 \times {}^1V_{12}$	${}^1V_{13}$	${}^1V_{13} = {}^1V_{12}$	$-\frac{2n}{2} \frac{2n-1}{3}$								0											$\frac{2n \cdot 2n-1}{2 \cdot 3}$	
17	-	${}^1V_{13} - {}^1V_1$	${}^1V_{13}$	${}^1V_{13} = {}^1V_1$	$-2n - 2$	1					$2n - 2$														
18	+	${}^1V_{13} + {}^1V_7$	${}^1V_{13}$	${}^1V_{13} = {}^1V_7$	$-3 + 1 = -4$	1																			
19	+	${}^1V_{13} + {}^1V_7$	1V_8	${}^1V_8 = {}^1V_7$	$-\frac{2n-2}{4}$						$2n - 2$	4	$\frac{2n-2}{4}$											$\left\{ \frac{2n \cdot 2n-1}{2} \frac{2n-2}{3} \right\} = A_2$	
20	×	${}^1V_8 \times {}^1V_{13}$	${}^1V_{11}$	${}^1V_{11} = {}^1V_{13}$	$-\frac{2n}{2} \frac{2n-1}{3} \frac{2n-2}{4} = A_2$								0												
21	×	${}^1V_{11} \times {}^1V_8$	${}^1V_{12}$	${}^1V_{12} = {}^1V_8$	$-B_2 \cdot \frac{2n \cdot 2n-1}{3} \frac{2n-2}{3} = B_2 A_2$																			0	
22	+	${}^1V_{12} + {}^1V_{11}$	${}^1V_{13}$	${}^1V_{13} = {}^1V_{11}$	$-A_0 + B_2 A_2 + B_2 A_2$																				
23	-	${}^1V_{13} - {}^1V_1$	${}^1V_{16}$	${}^1V_{16} = {}^1V_1$	$-n - 3 (-1)$	1																		$n-3$	
Here follows a repetition of Operations thirteen to twenty-three.																									
24				${}^1V_8 = {}^1V_8$																					
25				${}^1V_8 = {}^1V_8$ by a Variable-card. ${}^1V_7 = {}^1V_7$ by a Variable-card.	$+1 = 4 + 1 = 5$	1		n+1			0	0													



Ada Lovelace

“

[The Analytical Engine] might act upon other things besides *number*, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations, and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine...Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.

”



Electricity!

20th century and the first electronic computers

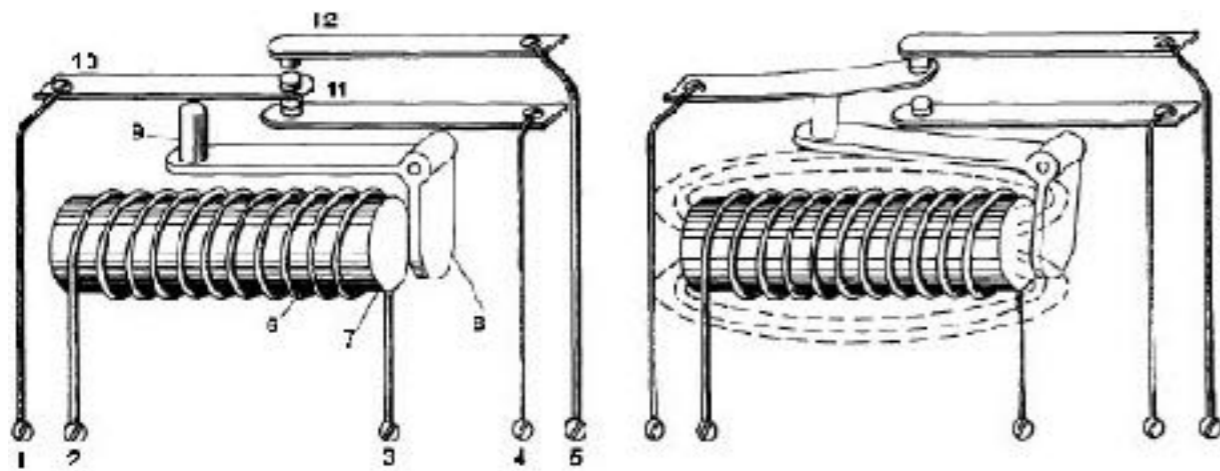


21
8531

22
8532

Electricity!

20th century and the first electronic computers



Relays

9/9

0800 Antan started
 1000 stopped - antan ✓

13'00 (030) MP-AC { 1.2700 9.027 007 025
 2.130476495
 2.130676915 } 9.027 846 095 result
 4.615925039(-2)

030 PRO 2.130476495
 const 2.130676915

Relays 6-2 in 032 failed special speed test
 in relay 11.00 test.

1100 Started Cosine Tape (Sine check)
 1525 Started Multi Adder Test.

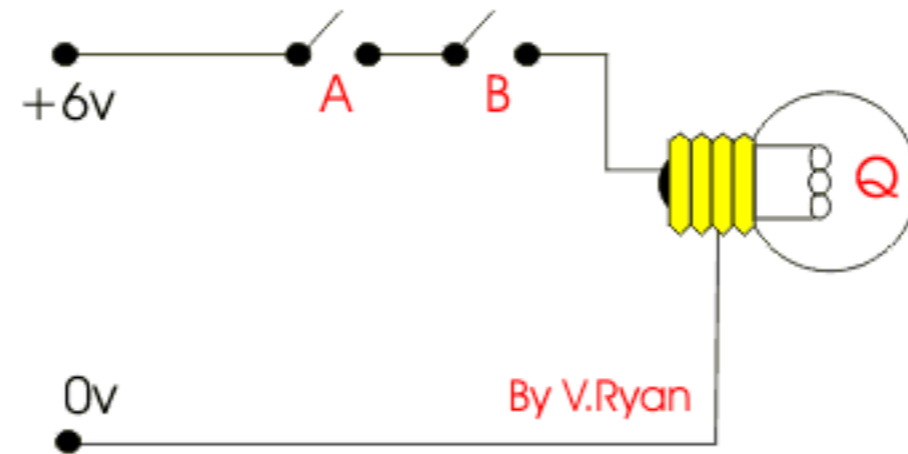
1545 Relay #70 Panel F
 (moth) in relay.

1630 Antan started.
 1700 closed down.

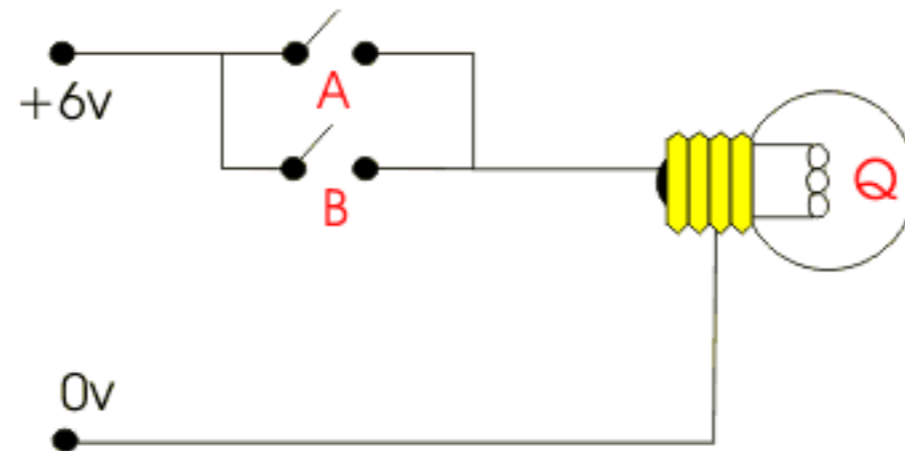
Relay 2142
 Relay 2172

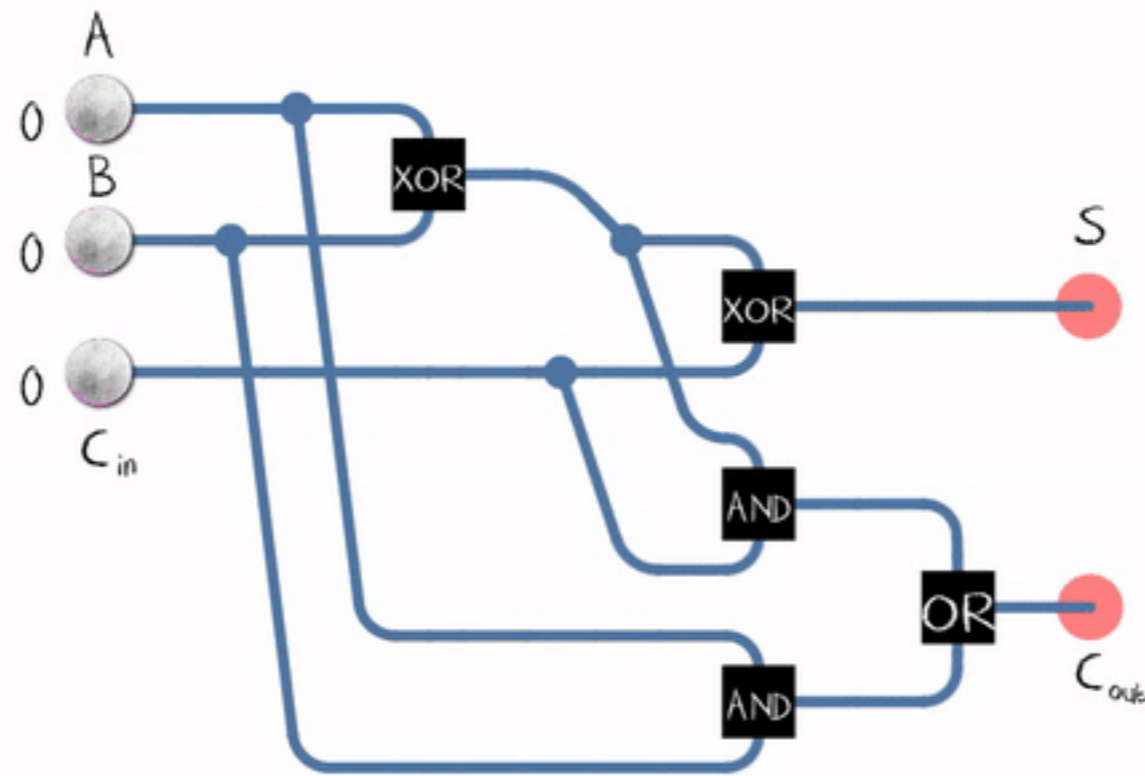
The first computer bugs

AND GATE

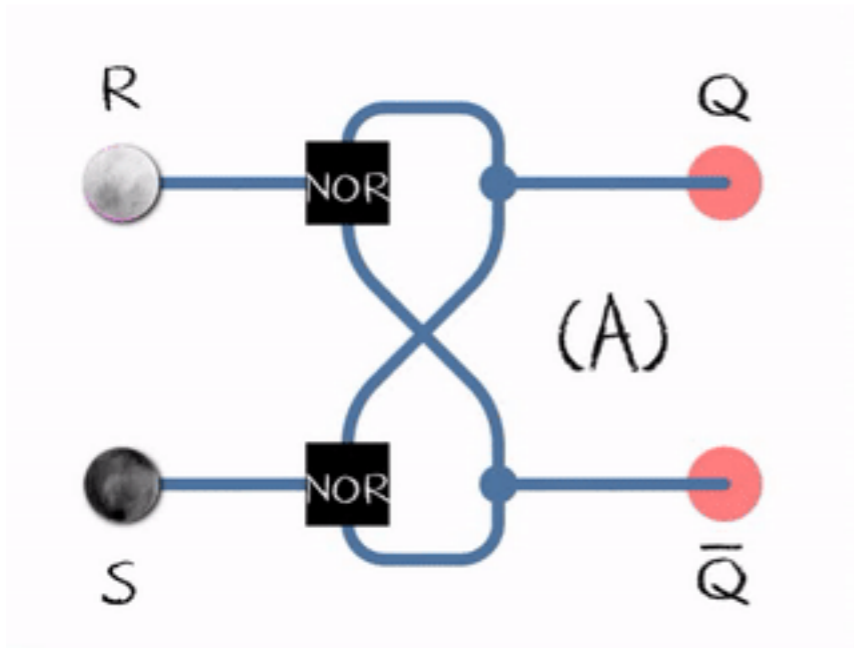


OR GATE





Logic Gates into Binary adders (and Calculators)

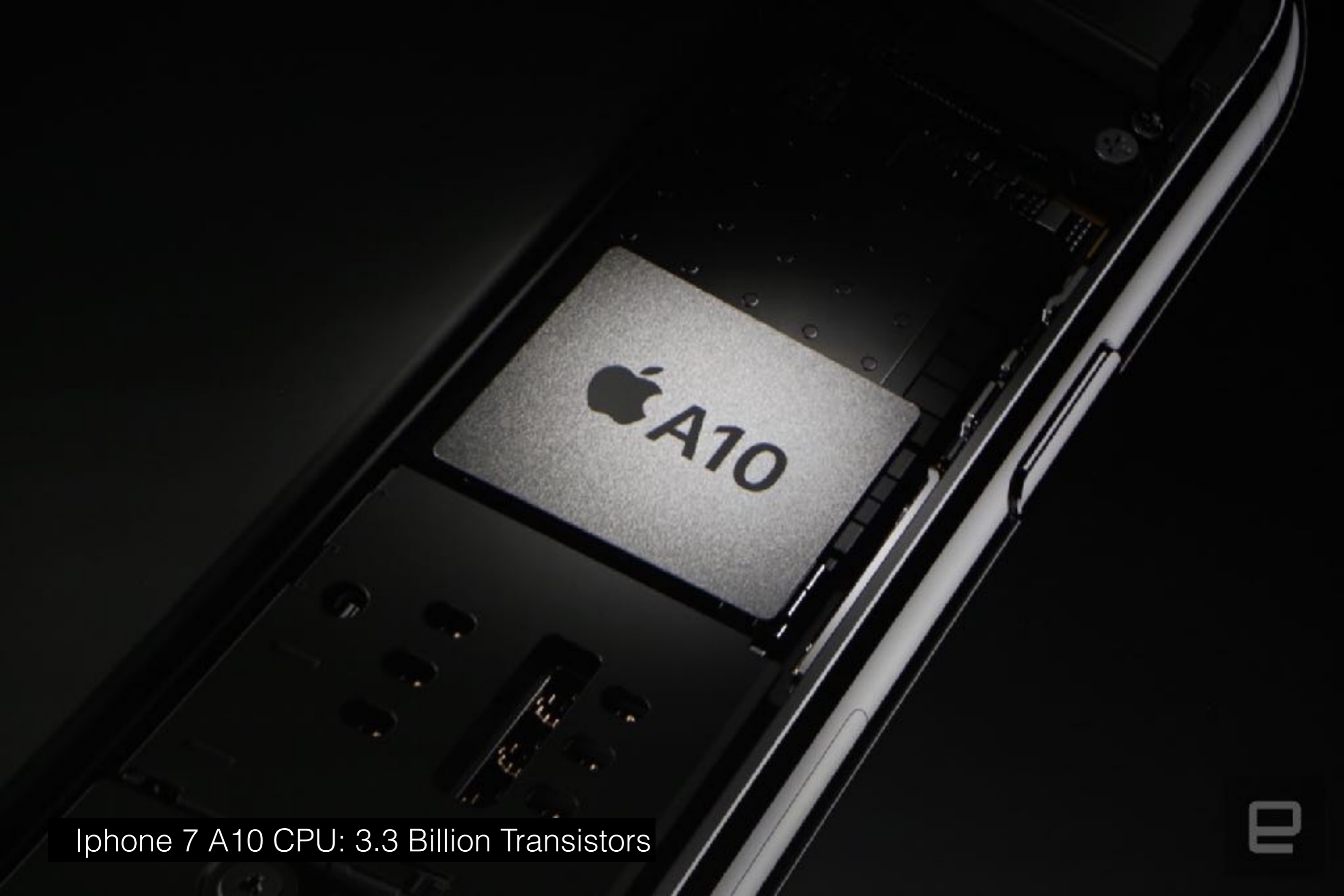


Volatile memory

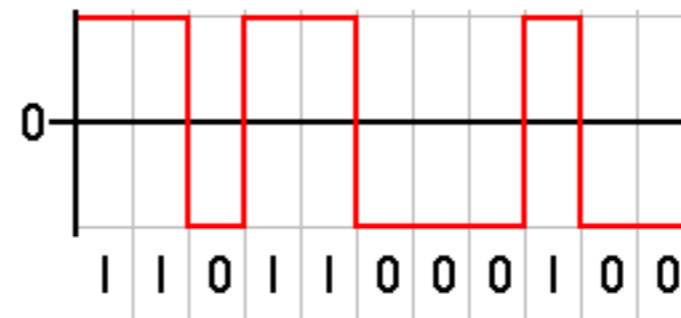
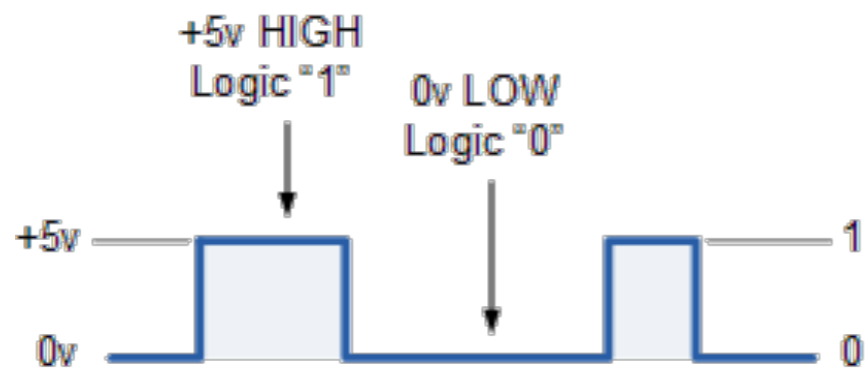


non volatile memory

Data Storage and Memory



Iphone 7 A10 CPU: 3.3 Billion Transistors



Electricity into Binary

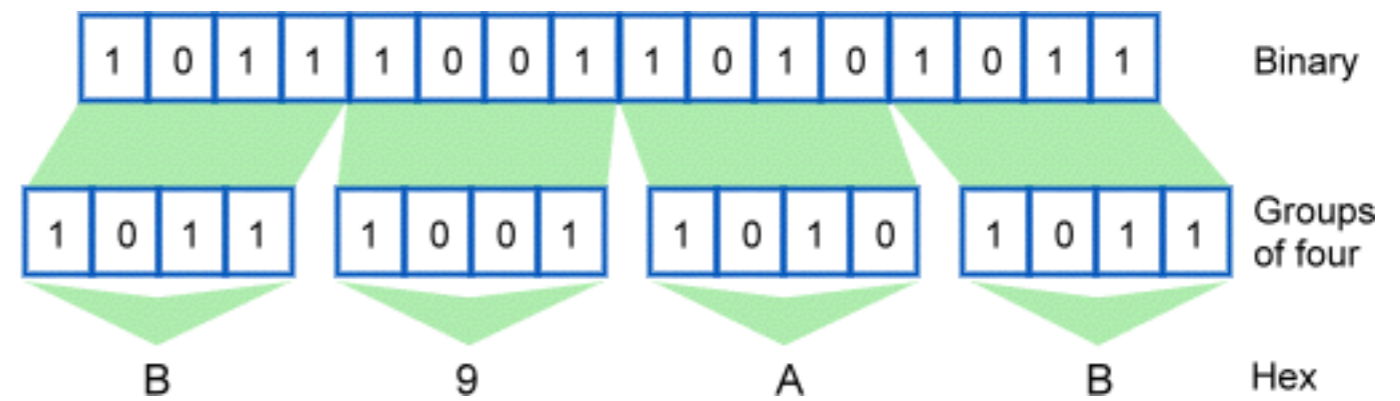
What are Bits all about?

STATE A	STATE B
1	0
TRUE	FALSE
HIGH	LOW
ON	OFF
OPEN	CLOSED
ACTIVE	INACTIVE
UP	DOWN

DECIMAL NUMBER	BINARY EQUIVALENT				PULSE - CODE WAVEFORMS			
	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				
9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

Counting in Binary

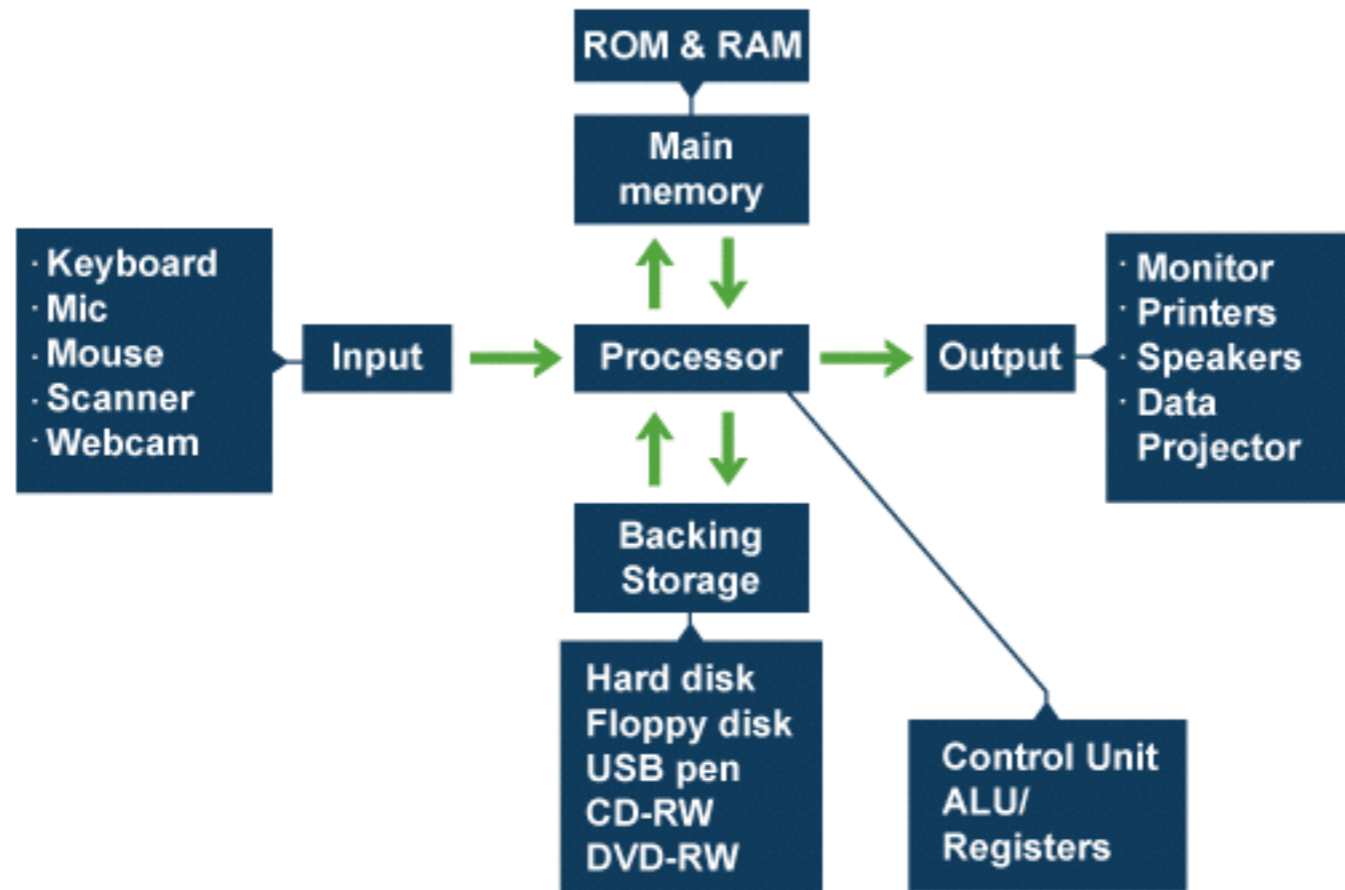
Binary	Decimal	Hexadecimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F



Hexadecimal

#1ABC9C Turquoise	#2ECC71 Emerald	#3498DB Peter River	#9B59B6 Amethyst	#34495E Wet Asphalt
#16A085 Green Sea	#27AE60 Nephritis	#2980B9 Belize Hole	#8E44AD Wisteria	#2C3E50 Green Sea
#F1C40F Sun Flower	#E67E22 Carrot	#E74C3C Alizarin	#ECF0F1 Clouds	#95A5A6 Concrete
#F39C12 Orange	#D35400 Pumpkin	#C0392B Pomegranate	#BDC3C7 Silver	#7F8C8D Asbestos

Hexadecimal Colour



Fitting it all together

```

def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodeName()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print '    %s [label="%s' % (nodename, label),
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '= %s'];' % ast[1]
        else:
            print '"]'
    else:
        print '"];'
        children = []
        for n, child in enumerate(ast[1:]):
            children.append(dotwrite(child))
        print ', ' %s -> {' % nodename
        for n, child in enumerate(children):
            print '%s' % child,

```

Coding Languages

Syntax and instructions

```
age = int (input
  ("Enter your age: ") )

if age < 17:
  print ("You are too young
    to drive")
else:
  print ("You are able to drive")
```

Program is
written in high
level language

Translator program

```
0 1 0 1      1 1 0 1
1 0 1 0      0 1 1 0
0 1 0 1      1 0 0 1
1 1 1 0      0 1 0 0
```

Machine code
is produced

Coding Languages

Compiling and Interpreting



Bits and Atoms

Luke Franzke - 2017