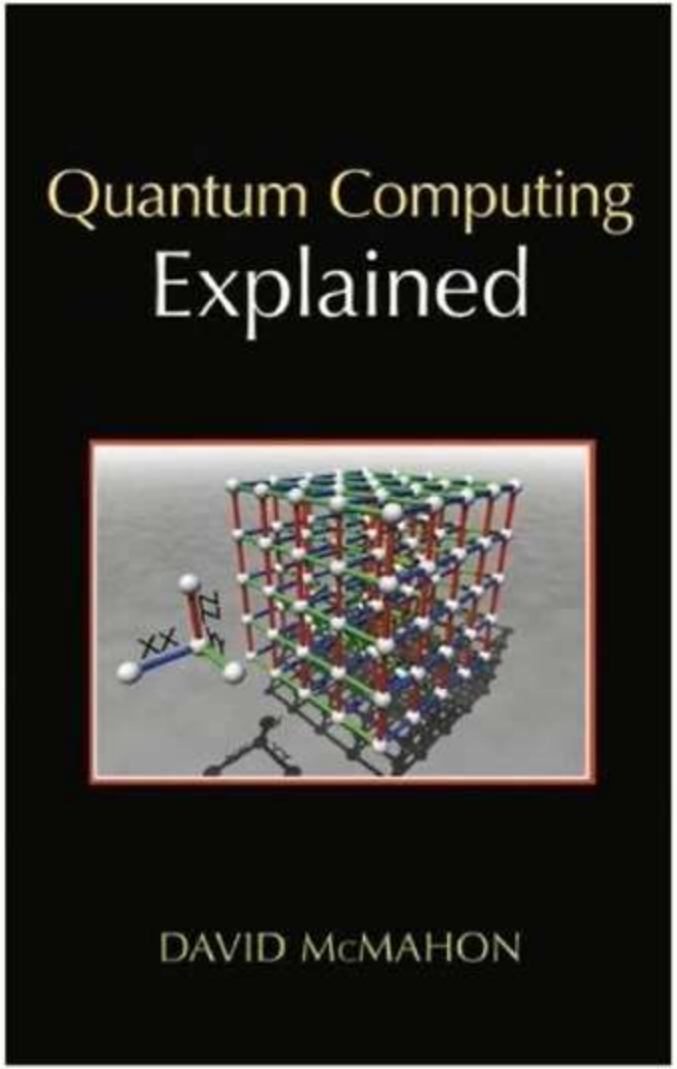
UNIVERSIDAD
NACIONAL
DE COLOMBIA

Computación cuántica

Prof. Alcides Montoya C., Ph.D
Escuela de Física
Universidad Nacional de Colombia Sede Medellín


Quantum Computing
Explained
DAVID McMAHON


colección textos
INTRODUCCIÓN
AL FORMALISMO
DE LA MECÁNICA CUÁNTICA
NO RELATIVISTA
Ma. Carolina Spinel Gómez
UNIVERSIDAD NACIONAL DE COLOMBIA
Facultad de Ciencias

Qiskit

¿Cuál es la diferencia entre la computación clásica y la computación cuántica?

Semillero de computación cuántica - UNAL

UNIVERSIDAD
NACIONAL
DE COLOMBIA

George Boole

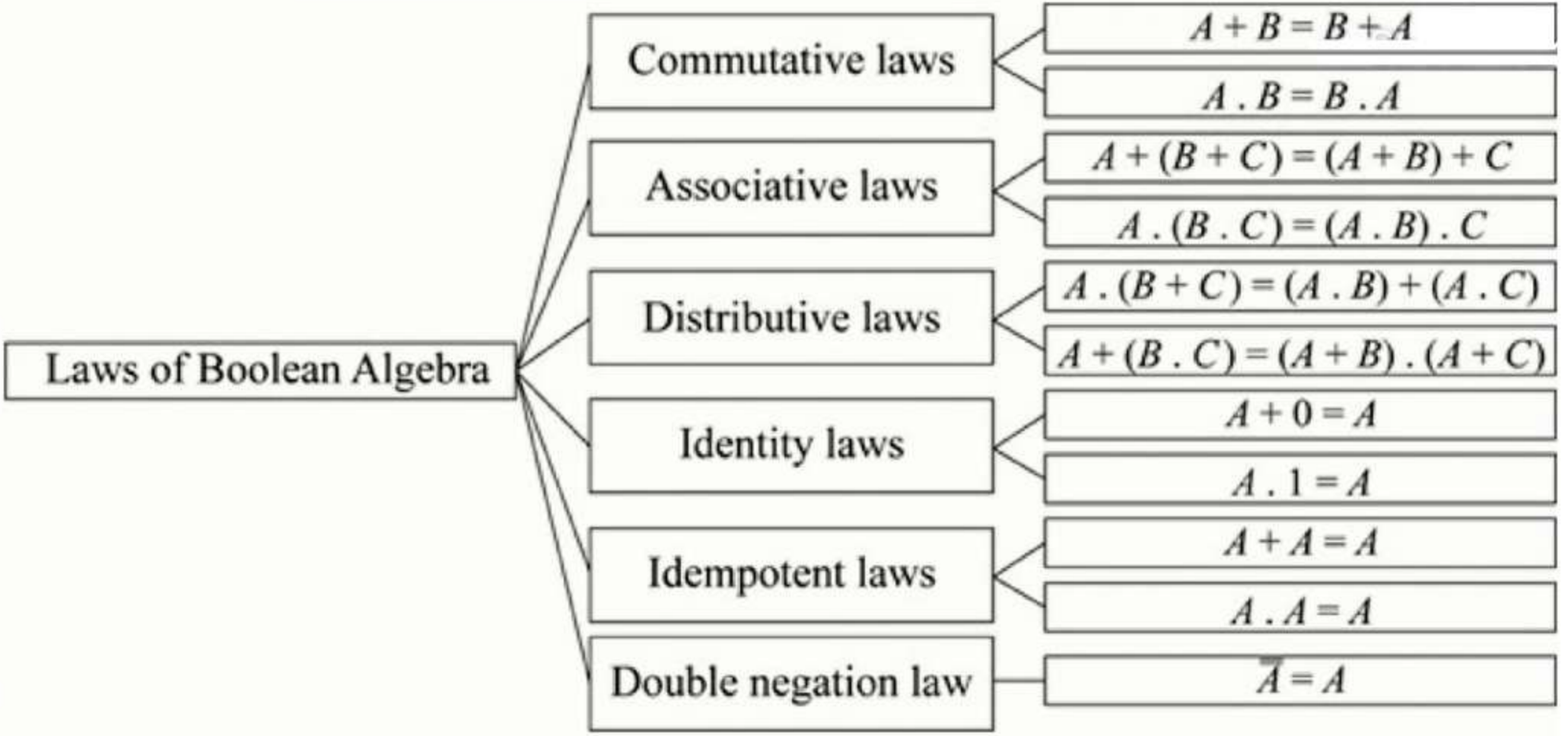
(Lincoln, Reino Unido, 1815 - Ballintemple, actual Irlanda, 1864)
Matemático británico, creador de un nuevo sistema de cálculo lógico que póstumamente sería llamado *Álgebra de Boole*.

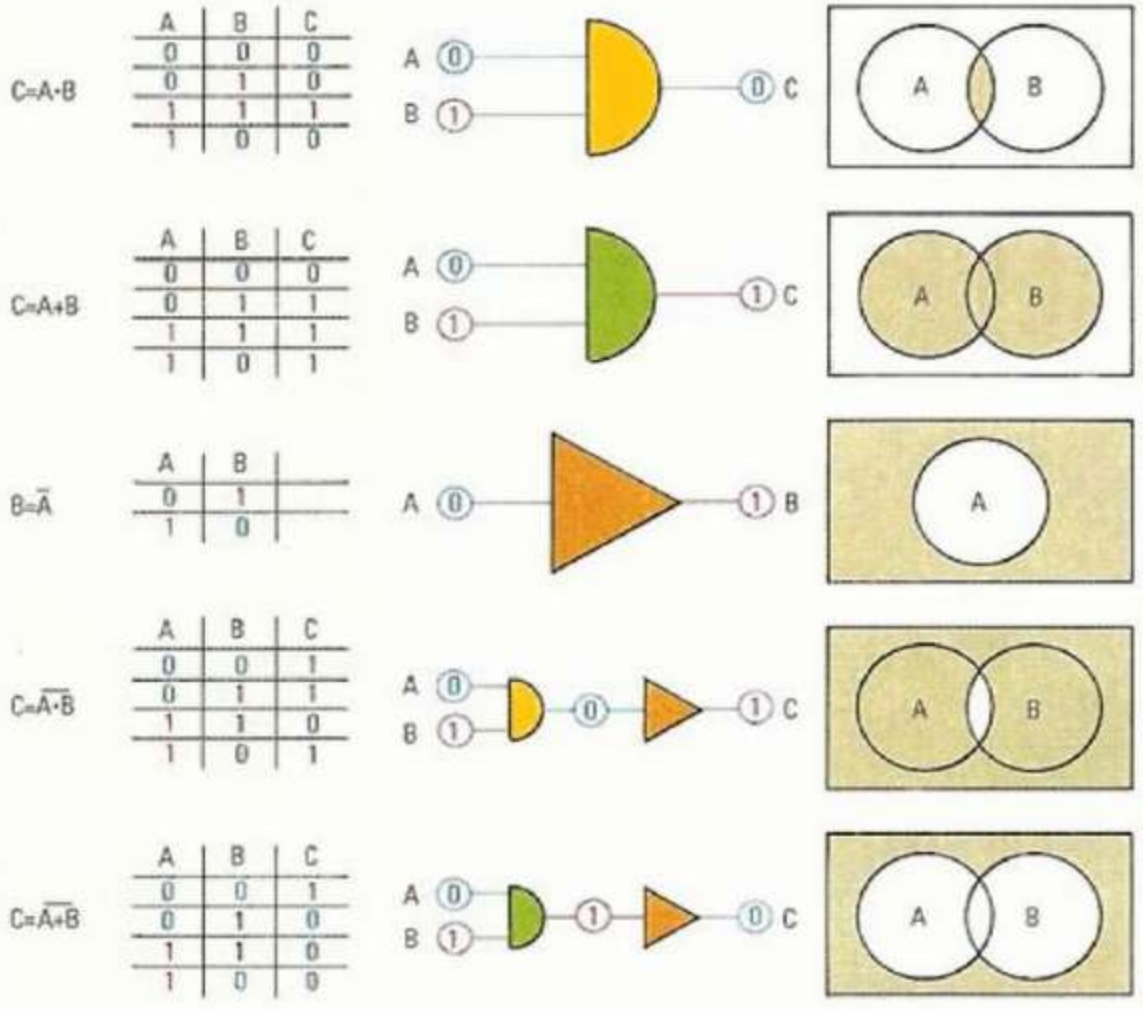
https://blogs.scientificamerican.com/guest-blog/the-bicentennial-of-george-boole-the-man-who-laid-the-foundations-of-the-digital-age/

Basic Boolean Algebraic Identities

Additive	Multiplicative
$A + 0 = A$	$0A = 0$
$A + 1 = 1$	$1A = A$
$A + A = A$	$AA = A$
$A + \overline{A} = 1$	$A\overline{A} = 0$

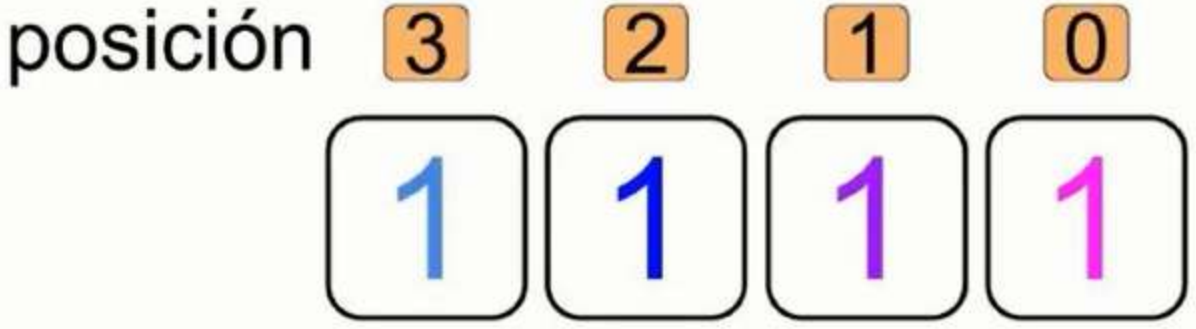
https://www.allaboutcircuits.com/textbook/digital/chpt-7/boolean-algebraic-identities/


Laws of Boolean Algebra
Commutative laws
$A + B = B + A$
$A \cdot B = B \cdot A$
Associative laws
$A + (B + C) = (A + B) + C$
$A \cdot (B \cdot C) = (A \cdot B) \cdot C$
Distributive laws
$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$
$A + (B \cdot C) = (A + B) \cdot (A + C)$
Identity laws
$A + 0 = A$
$A \cdot 1 = A$
Idempotent laws
$A + A = A$
$A \cdot A = A$
Double negation law
$\overline{\overline{A}} = A$



https://www.daviddarling.info/encyclopedia/B/Boolean_algebra.html

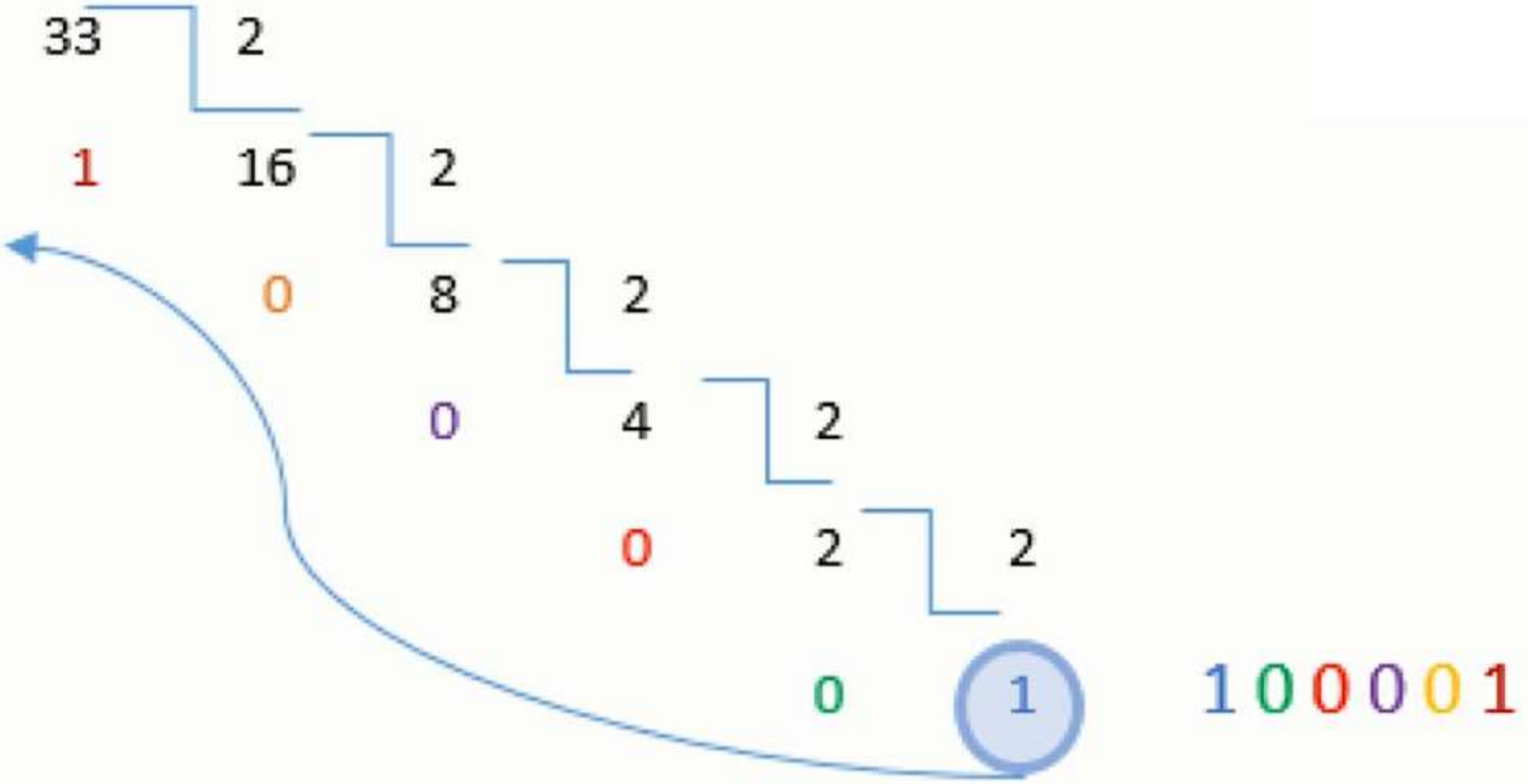
Decimal	Binario	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F



Código binario

$$1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

Código decimal: 8+4+2+1 = 15


33 2
1 16 2
0 8 2
0 4 2
0 2 2
0 1
100001

Decimal = 4156

Division	Quotient	Remainder
4156/16	259	12 – C
259/16	16	3
16/16	1	0
1/16	0	1

Hexadecimal = 103C

https://www.onlinemath4all.com/octal-to-decimal-binary-hexadecimal-converter.html

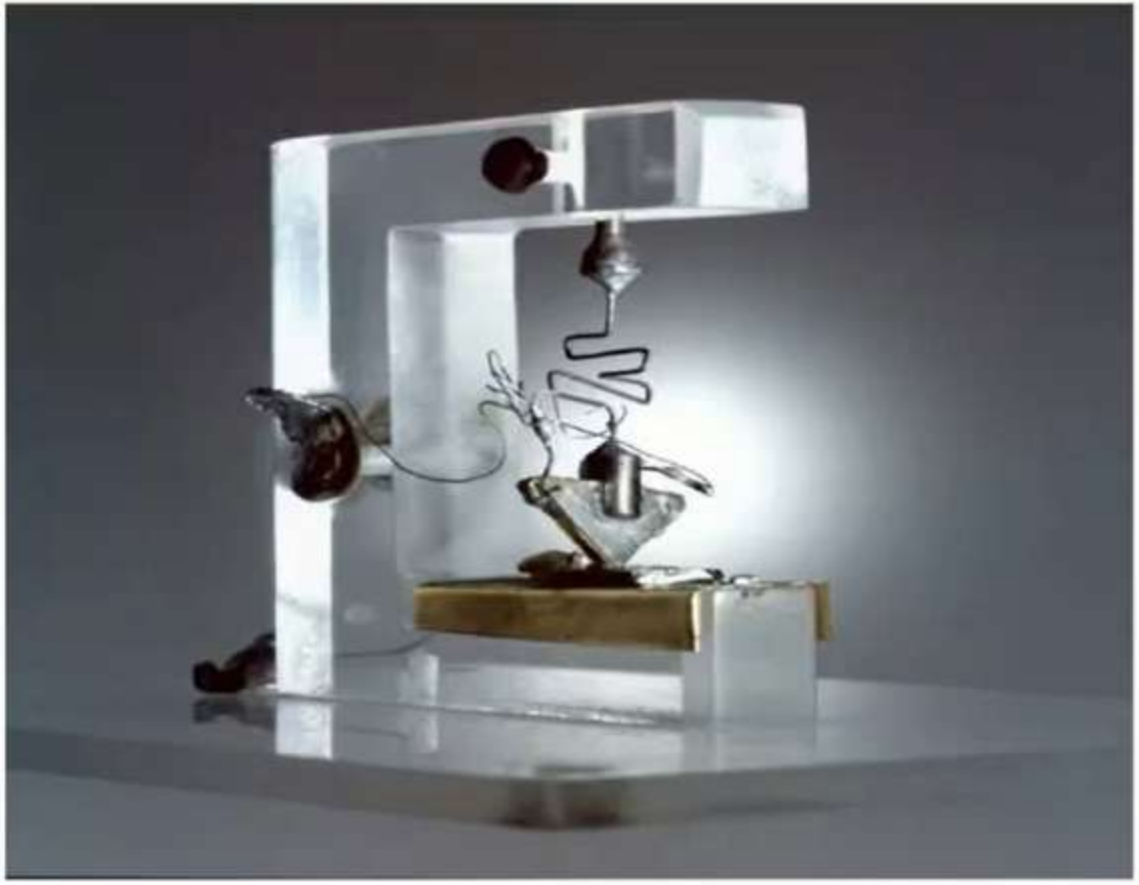
Hexadecimal	Decimal	Binario	Octal
0	0	0000	0
1	1	0001	1
2	2	0010	2
3	3	0011	3
4	4	0100	4
5	5	0101	5
6	6	0110	6
7	7	0111	7
8	8	1000	10
9	9	1001	11
A	10	1010	12
B	11	1011	13
C	12	1100	14
D	13	1101	15
E	14	1110	16
F	15	1111	17

BCD Code

Decimal	8 4 2 1	2 4 2 1	8 4 -2 -1	XS-3
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 0 0 1	0 1 1 1	0 1 0 0
2	0 0 1 0	0 0 1 0	0 1 1 0	0 1 0 1
3	0 0 1 1	0 0 1 1	0 1 0 1	0 1 1 0
4	0 1 0 0	0 1 0 0	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 1 1	1 0 1 1	1 0 0 0
6	0 1 1 0	1 1 0 0	1 0 1 0	1 0 0 1
7	0 1 1 1	1 1 0 1	1 0 0 1	1 0 1 0
8	1 0 0 0	1 1 1 0	1 0 0 0	1 0 1 1
9	1 0 0 1	1 1 1 1	1 1 1 1	1 1 0 0

El primer transistor

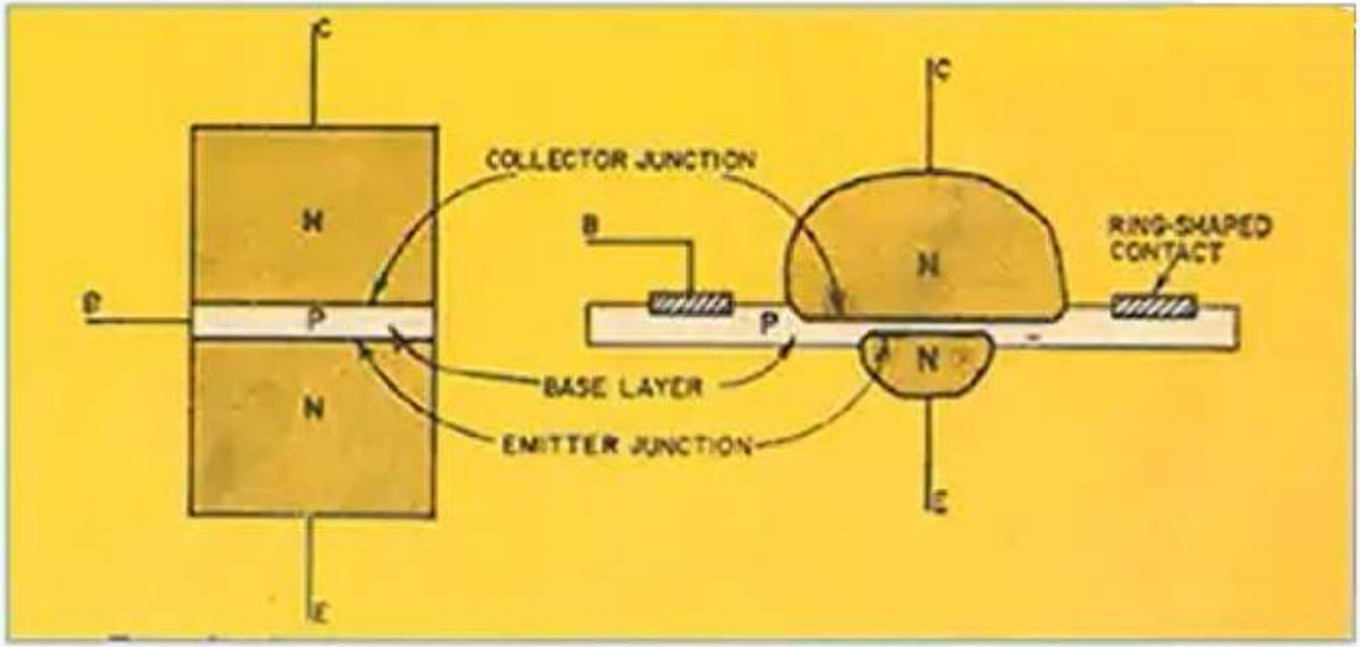
The first transistor was invented by John Bardeen (left) and Walter Brattain (right) and later in 1947 William Shockley (centre) saw the great potential of the device. The same year the first transistor, now an essential part of making smaller electronic devices such as computers and phones, was demonstrated at Bell Laboratories.



https://www.digikey.com/es/maker/blogs/2018/71st-anniversary-of-the-transistor

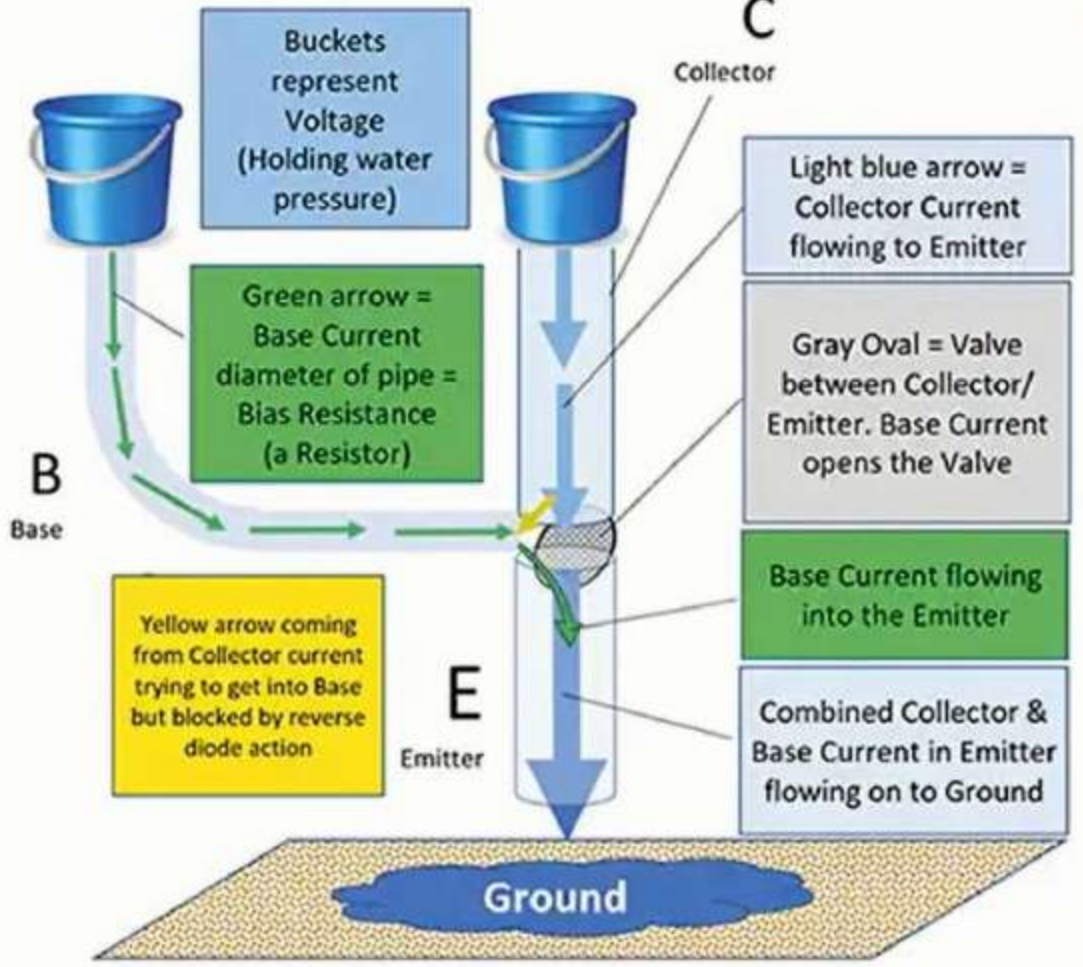
1a)

1b)

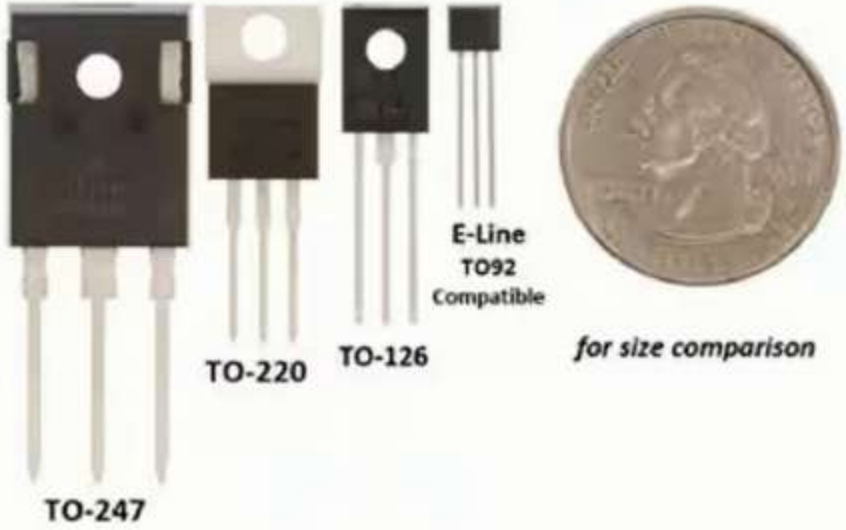


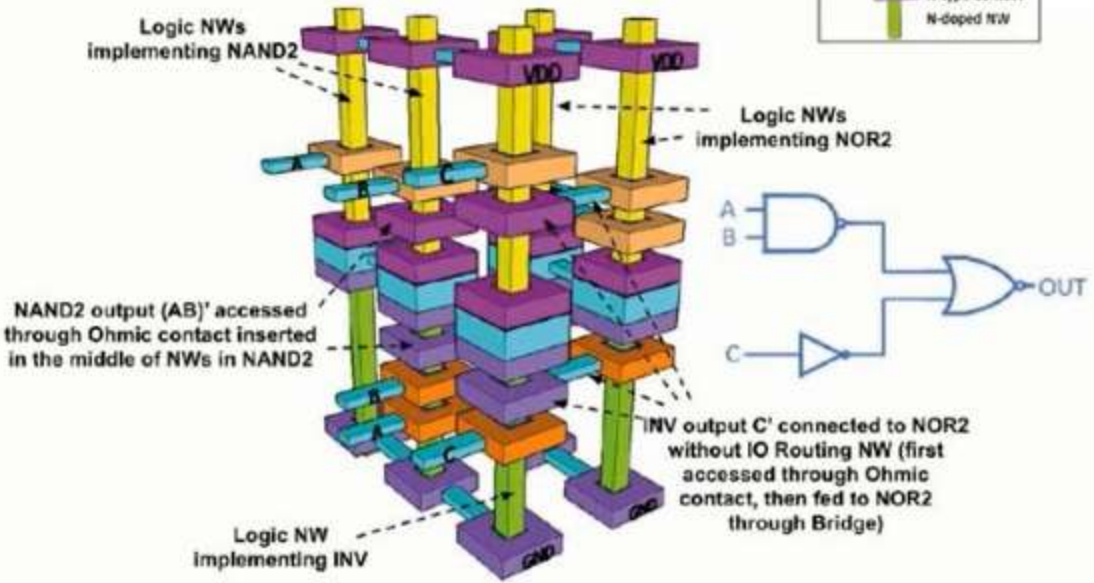
https://www.digikey.com/en/articles/transistor-basics

NPN Transistor Operation



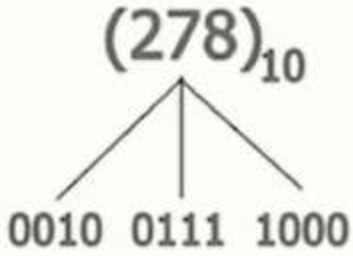
https://www.digikey.com/en/articles/transistor-basics



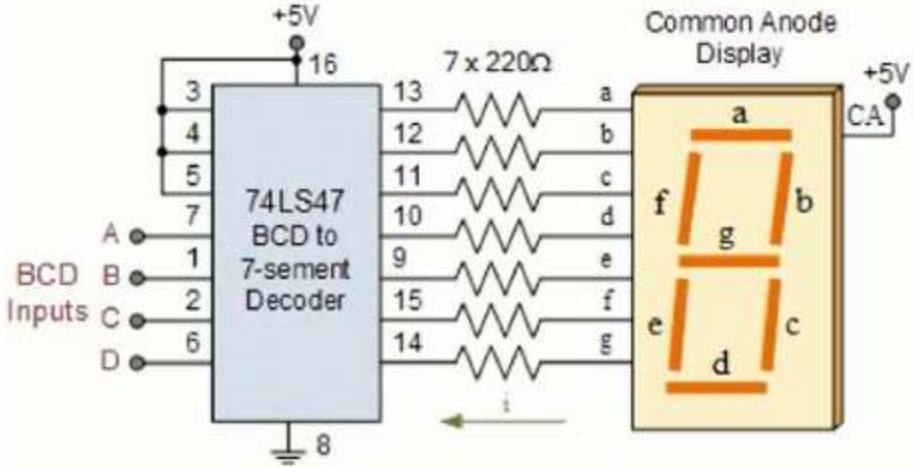


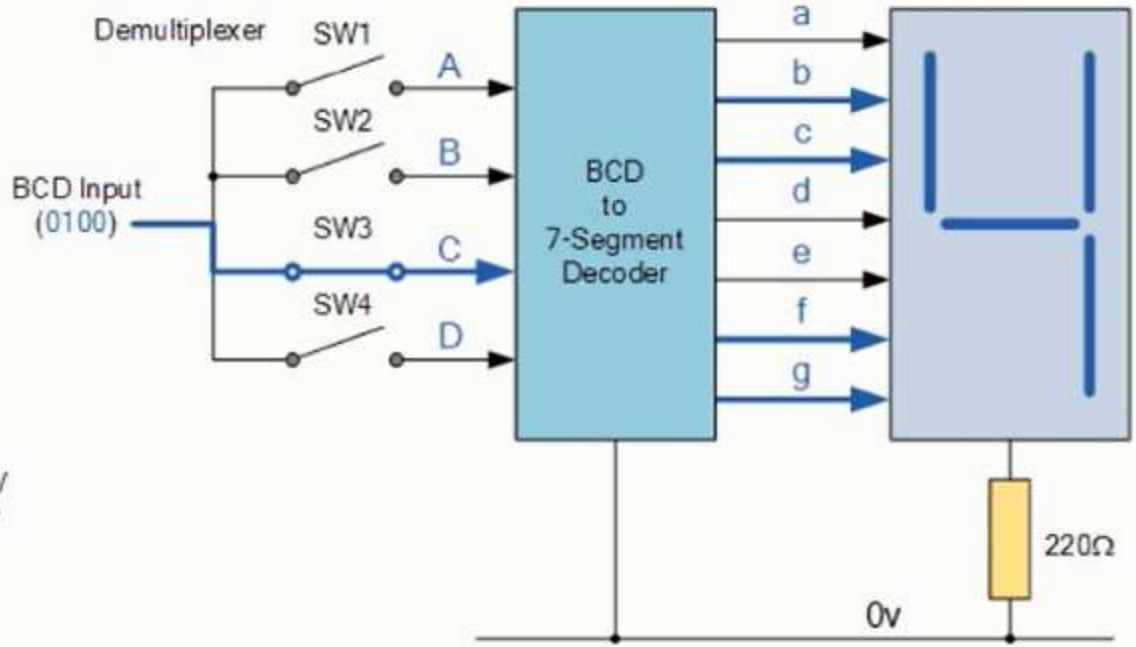
https://researchoutreach.org/articles/engineering-technology/architecting-integrated-circuit-fabrics-nanoscale/

https://www.digikey.com/es/maker/blogs/2018/71st-anniversary-of-the-transistor



Therefore, $(278)_{10}$ = 0010 0111 1000





1999: Intel Pentium III Xeon™

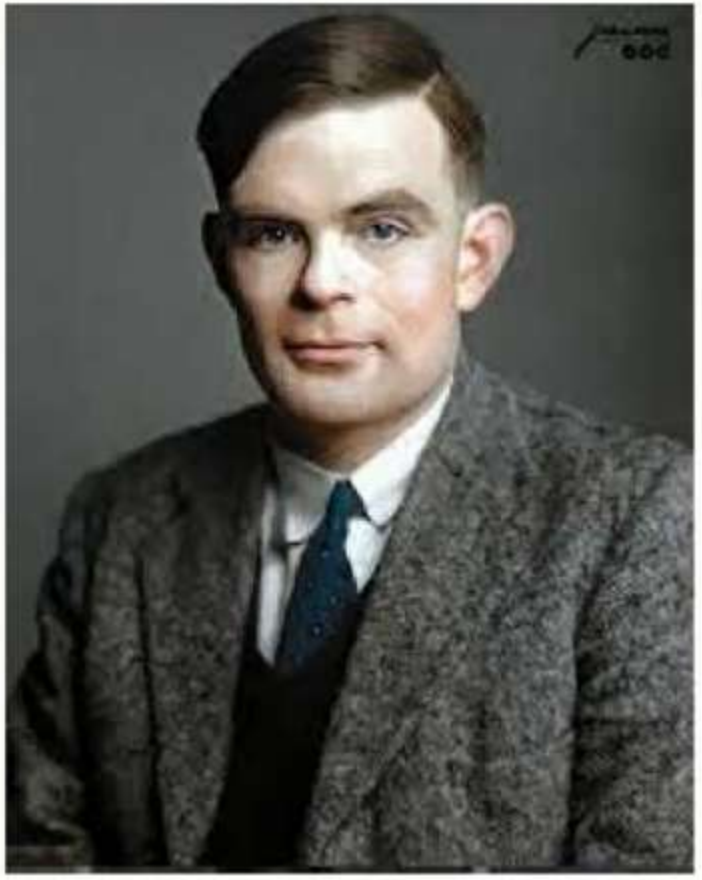
https://www.tayloredge.com/museum/processor/processorhistory.html

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Deci
0	0	[NULL]	32	20	[SPACE]	64
1	1	[START OF HEADING]	33	21	!	65
2	2	[START OF TEXT]	34	22	"	66
3	3	[END OF TEXT]	35	23	#	67
4	4	[END OF TRANSMISSION]	36	24	$	68
5	5	[ENQUIRY]	37	25	%	69
6	6	[ACKNOWLEDGE]	38	26	&	70
7	7	[BELL]	39	27	'	71
8	8	[BACKSPACE]	40	28	(	72
9	9	[HORIZONTAL TAB]	41	29	)	73
10	A	[LINE FEED]	42	2A	*	74
11	B	[VERTICAL TAB]	43	2B	+	75

Alan Turing - 23 June 1912 – 7 June 1954

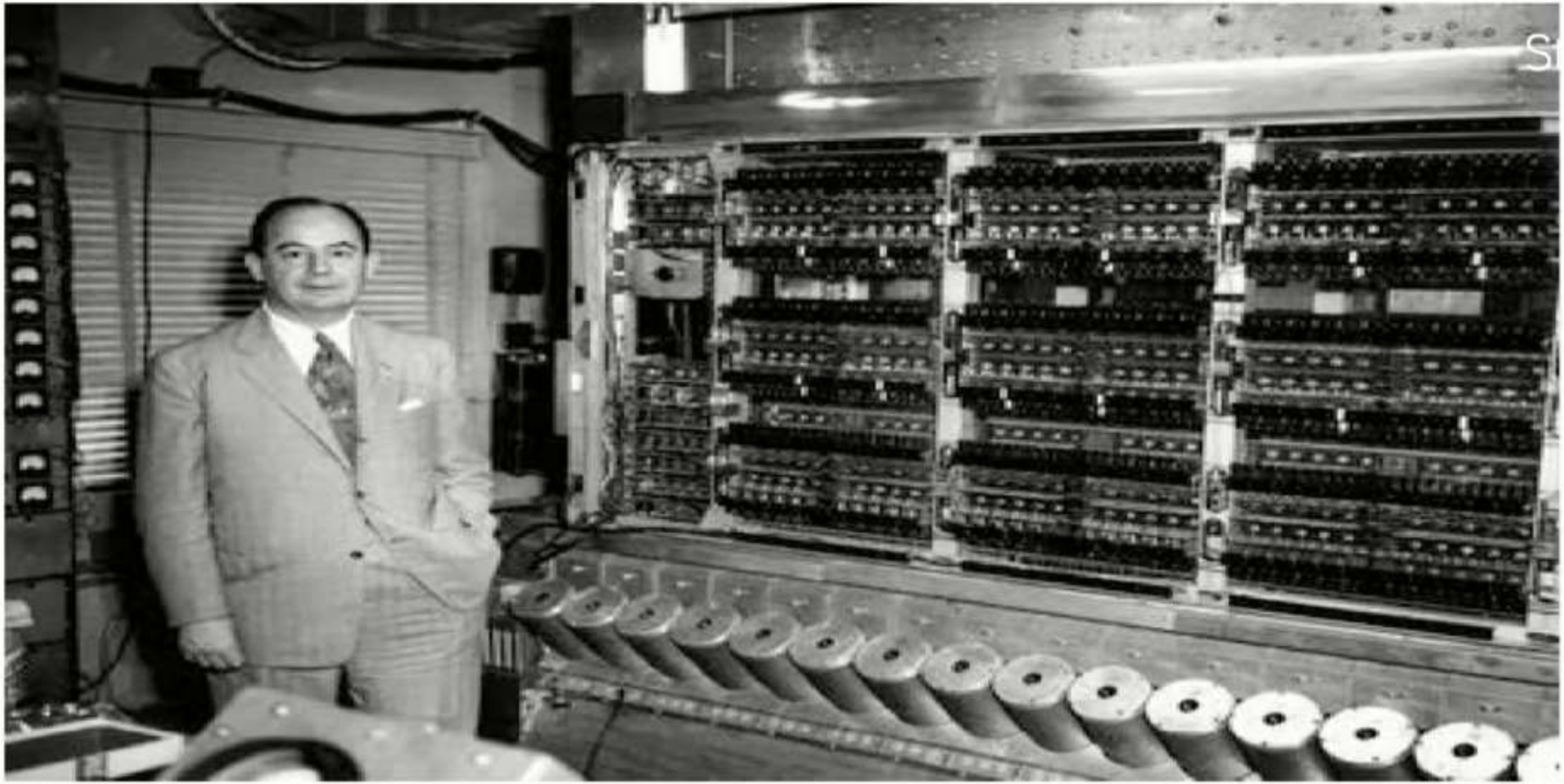
“Turing atacó el problema imaginando una máquina con una cinta infinitamente larga. La cinta está cubierta con símbolos que dan instrucciones a la máquina, diciéndole cómo manipular otros símbolos. Esta máquina de Turing universal, como se la conoce, es un modelo matemático de las computadoras modernas que todos usamos hoy”



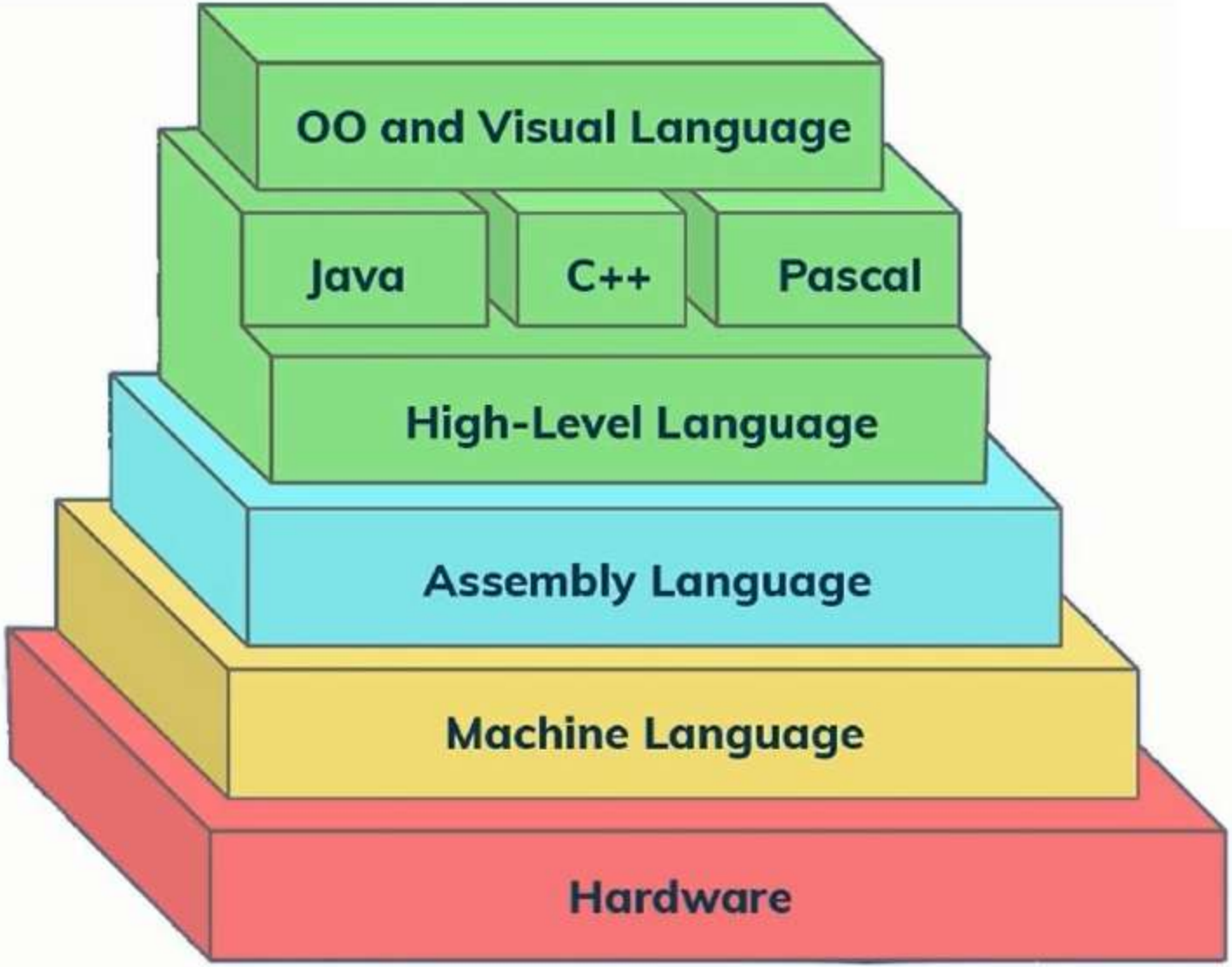

140

John von Neumann, el matemático que diseñó los ordenadores modernos

"El científico húngaro-estadounidense construyó el primer computador que podía realizar diversas tareas, entre otras muchas contribuciones a todo tipo de disciplinas desde la física cuántica a la economía"



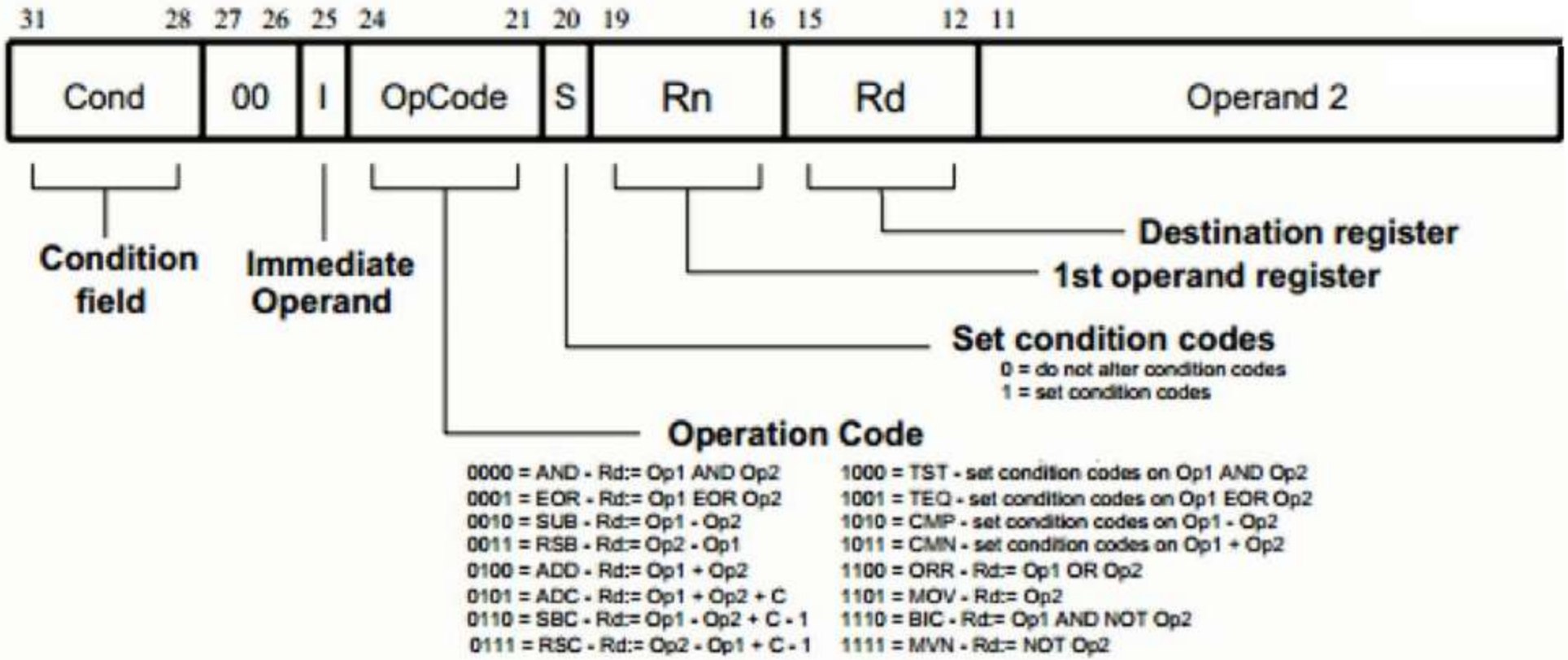
https://elpais.com/ciencia/cafe-y-teoremas/2023-02-23/john-von-neumann-el-matematico-que-diseno-los-ordenadores-modernos.html


OO and Visual Language
Java
C++
Pascal
High-Level Language
Assembly Language
Machine Language
Hardware

Machine code and instruction sets

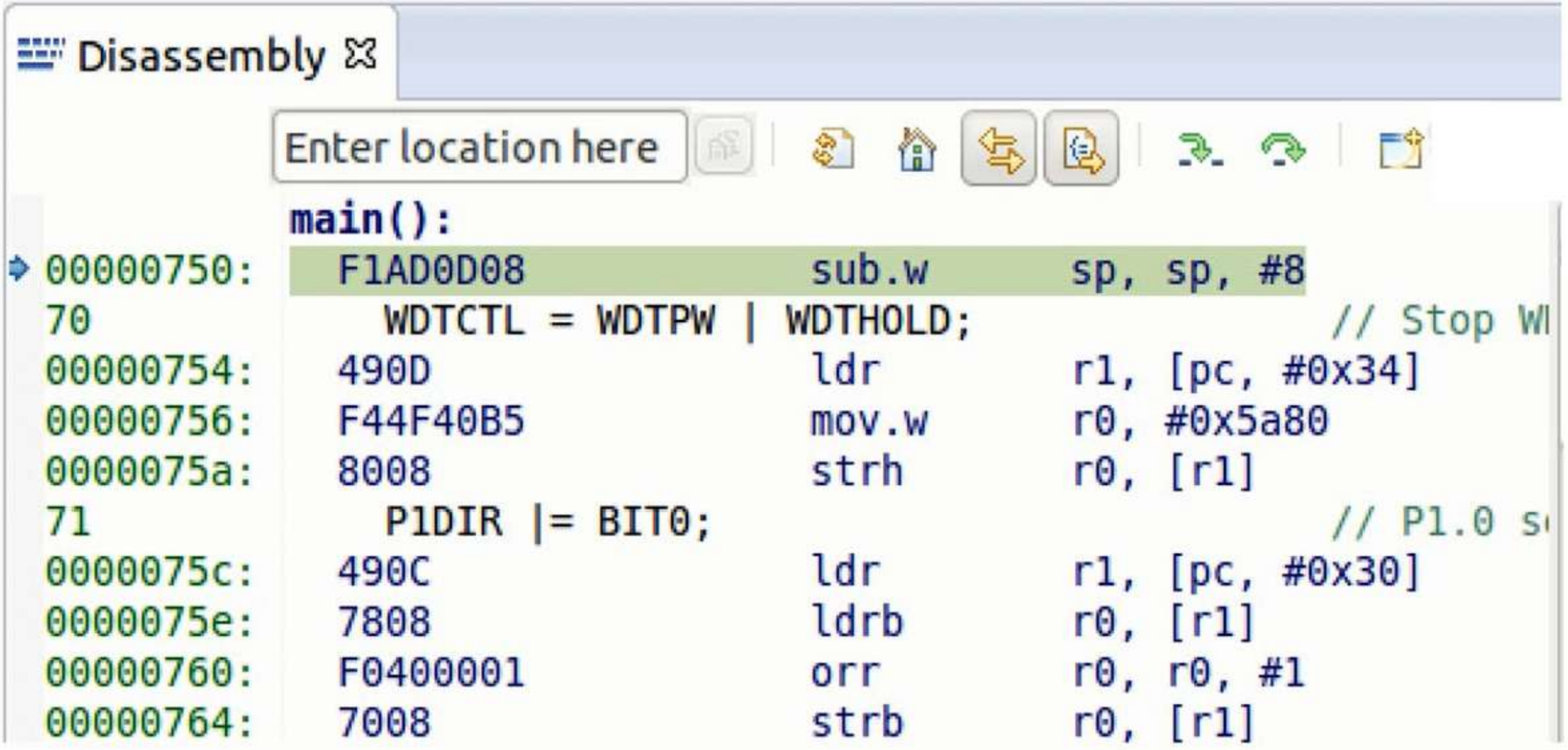
There is no set binary bit pattern for different opcodes in an instruction set. Different processors will use different patterns, but sometimes it might be the case that you are given certain bit patterns that represent different opcodes. You will then be asked to write machine code instructions using them. Below is an example of bit patterns that might represent certain instructions.

Machine code	Instruction	Addressing mode	Example
0000	STORE	Address	STO 12
0001	LOAD	Number	LDA #12
0010	LOAD	Address	LDA 12
0100	ADD	Number	ADD #12
1000	ADD	Address	ADD 12
1111	HALT	None	HALT



ARM data-processing instruction

https://www.allaboutcircuits.com/technical-articles/how-to-write-assembly-basic-assembly-instructions-ARM-instruction-set/


Disassembly
Enter location here
main():
00000750: F1AD0D08 sub.w sp, sp, #8
70 WDTCTL = WDTPW | WDTHOLD; // Stop WI
00000754: 490D ldr r1, [pc, #0x34]
00000756: F44F40B5 mov.w r0, #0x5a80
0000075a: 8008 strh r0, [r1]
71 P1DIR |= BIT0; // P1.0 s
0000075c: 490C ldr r1, [pc, #0x30]
0000075e: 7808 ldrb r0, [r1]
00000760: F0400001 orr r0, r0, #1
00000764: 7008 strb r0, [r1]

High-level Language

```
temp    = v[k];
v[k]    = v[k+1];
v[k+1]  = temp;
```

```
TEMP  = V(K)
V(K)  = V(K+1)
V(K+1) = TEMP
```

C/Java Compiler

Fortran Compiler

Assembly Language

```
lw  $to,   0($2)
lw  $t1,   4($2)
sw  $t1,   0($2)
sw  $t0,   4($2)
```

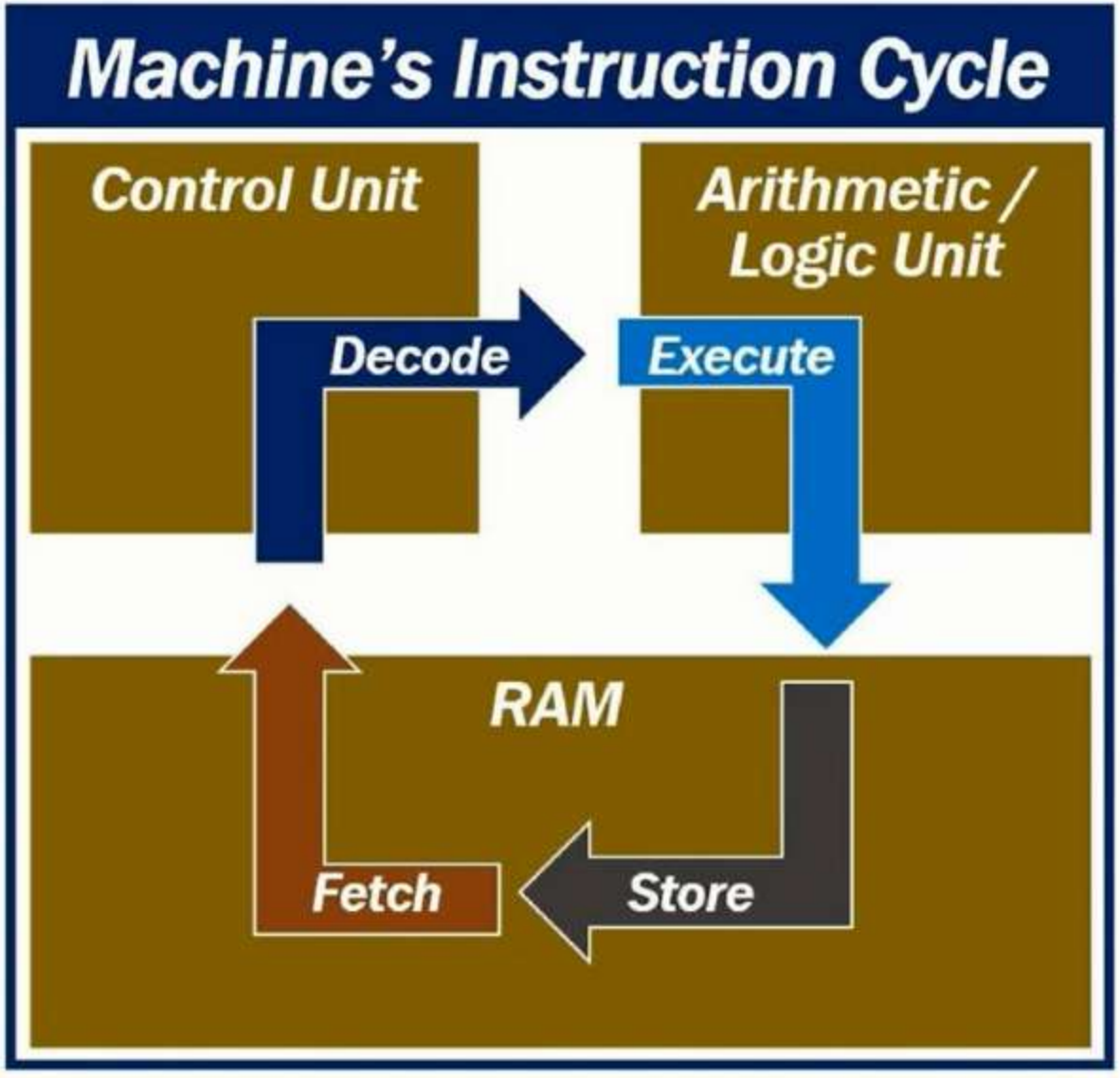
MIPS Assembler

Machine Language

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

Machine language instructions	Assembly language instructions	FORTRAN language instructions
0110 0011 0010 0001	LDA X	$D = X + Y + Z$
0100 0011 0010 0010	ADA Y	
0100 0011 0010 0011	ADA Z	
0111 0011 0010 0100	STA D	

https://youtu.be/Mv2XQgpbTNE


Machine's Instruction Cycle
Control Unit
Arithmetic / Logic Unit
Decode
Execute
RAM
Fetch
Store

Propongo: Leer el capítulo 0 del texto Introducción al formalismo de la mecánica cuántica no relativista