© Jerzy Nawrocki, Introduction to Computing

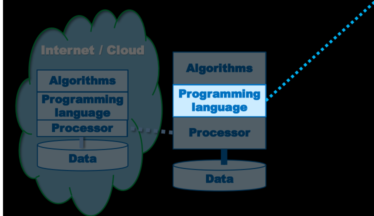
Introduction to Computing

Digital Circuits

Jerzy Nawrocki

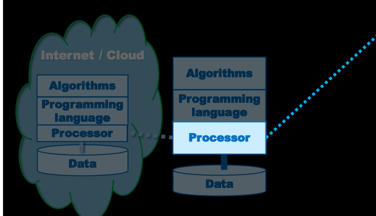
jerzy.nawrocki@put.poznan.pl
Faculty of Computing & Telecom.
Poznan University of Technology

Conceptual map of the lectures

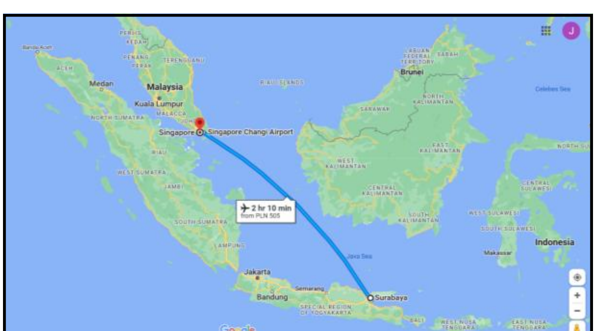


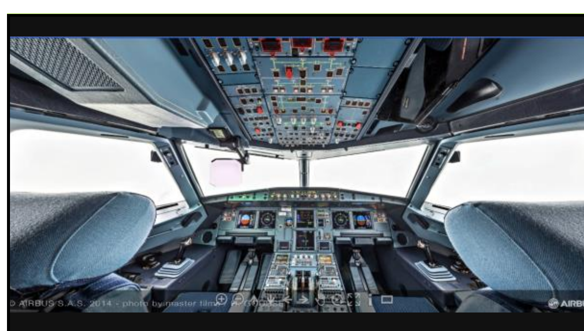
Topic
Imperative Programming
Digital Circuits
Computers
Subprograms
Numerical Methods
Computational Complexity
Object-oriented Programming
Text Processing
Databases and Machine Learning
Parallel Processing
Computer Networks & Cybersec.
Software Engineering
Embedded Systems
Professionalism in Computing

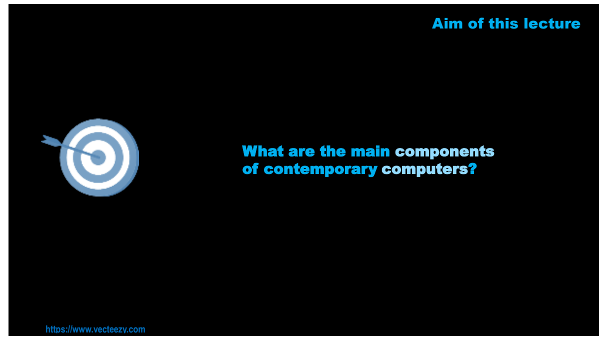
Conceptual map of the lectures

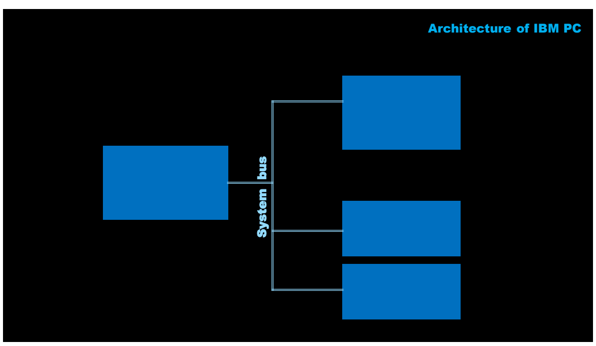


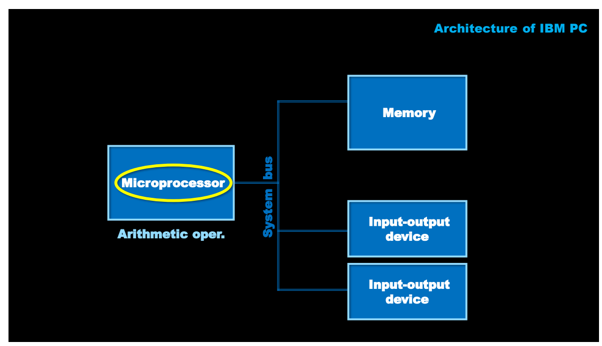
Topic
Imperative Programming
Digital Circuits
Computers
Subprograms
Numerical Methods
Computational Complexity
Object-oriented Programming
Text Processing
Databases and Machine Learning
Parallel Processing
Computer Networks & Cybersec.
Software Engineering
Embedded Systems
Professionalism in Computing

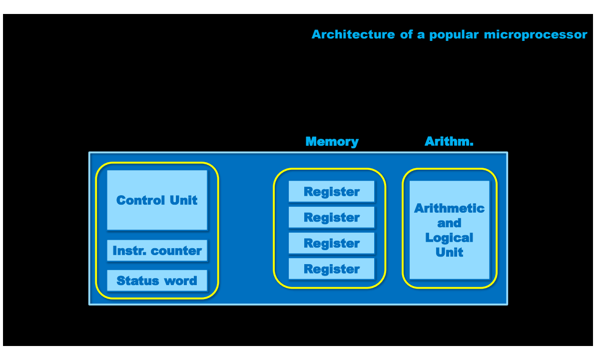


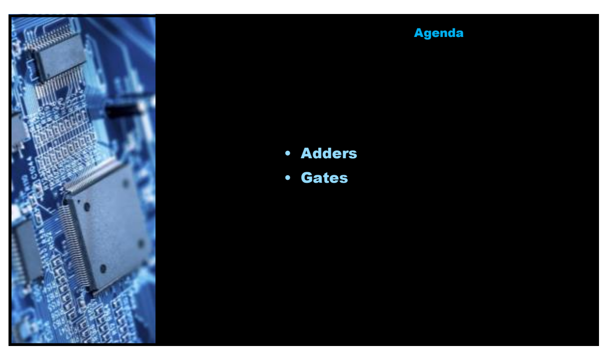


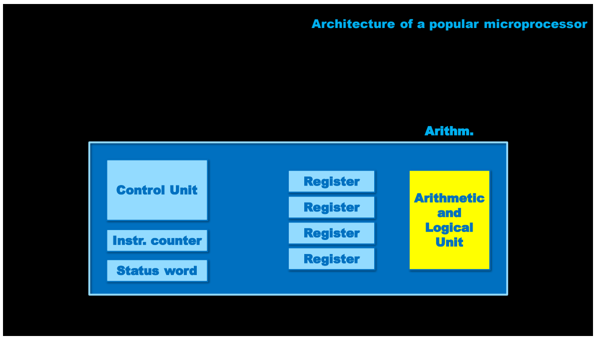

Aim of this lecture
What are the main components
of contemporary computers?
https://www.vecteezy.com


Architecture of IBM PC
System bus


Architecture of IBM PC
Memory
Microprocessor
Arithmetic oper.
System bus
Input-output device
Input-output device

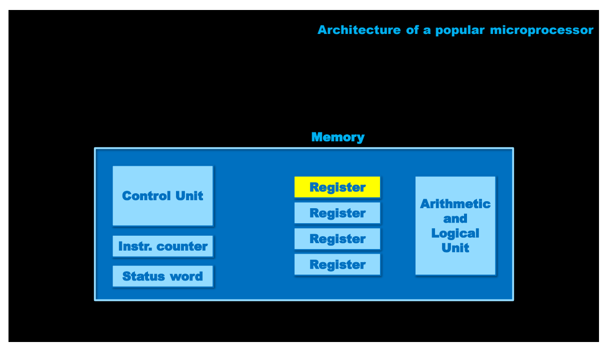

Architecture of a popular microprocessor
Memory
Arithm.
Control Unit
Instr. counter
Status word
Register
Register
Register
Register
Arithmetic and Logical Unit

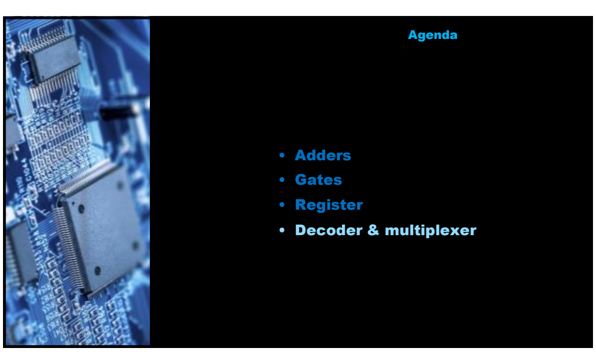

Agenda
• Adders
• Gates

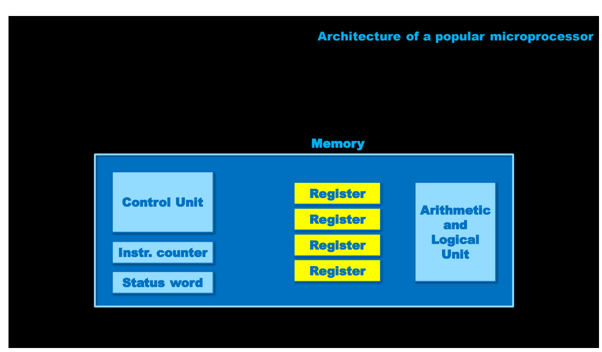

Architecture of a popular microprocessor
Arithm.
Control Unit
Instr. counter
Status word
Register
Register
Register
Register
Arithmetic and Logical Unit


Pascaline (1642)
https://kids.britannica.com/students/article/Pascaline/332603/media
Blaise Pascal
1623 - 1662
https://en.wikipedia.org/wiki/Blaise_Pascal


Agenda
• Adders
• Gates
• Register


Architecture of a popular microprocessor
Memory
Control Unit
Instr. counter
Status word
Register
Register
Register
Register
Arithmetic and Logical Unit


Agenda
• Adders
• Gates
• Register
• Decoder & multiplexer


Architecture of a popular microprocessor
Memory
Control Unit
Instr. counter
Status word
Register
Register
Register
Register
Arithmetic and Logical Unit

Agenda

- Adders
- Gates
- Register
- Decoder & multiplexer

Architecture of a popular microprocessor

Decimal arithmetics

$$101_{10}$$

$$1*10^2 + 0*10^1 + 1*10^0$$

$$= 100 + 0 + 1$$

Decimal arithmetics

$$101_2$$

$$1*2^2 + 0*2^1 + 1*2^0$$

$$= 4 + 0 + 1$$

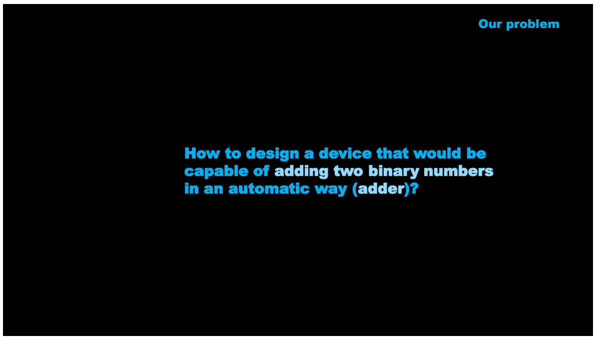
Binary arithmetics

Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000

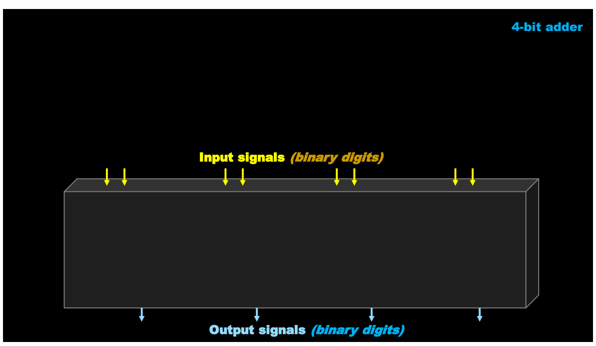
Quiz

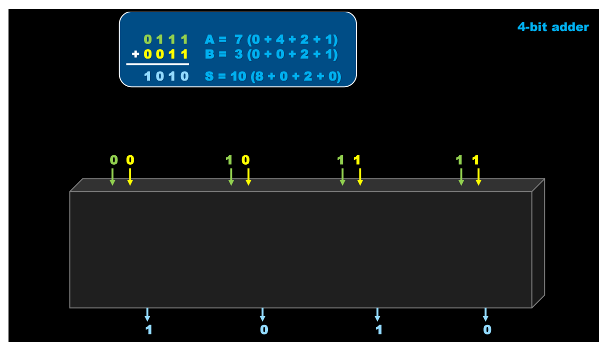
Which of the following is the binary representation of decimal number 15?

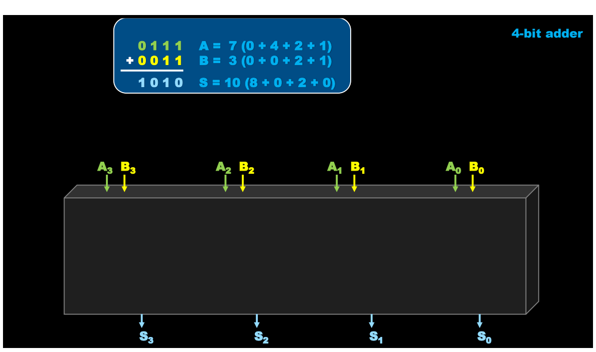
a) 11
b) 111
c) 1111
d) 11111

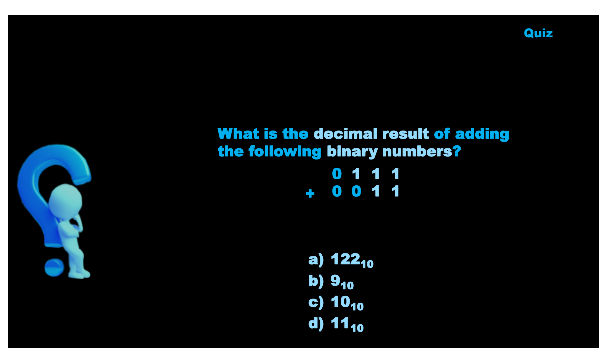

Our problem
How to design a device that would be capable of adding two binary numbers in an automatic way (adder)?

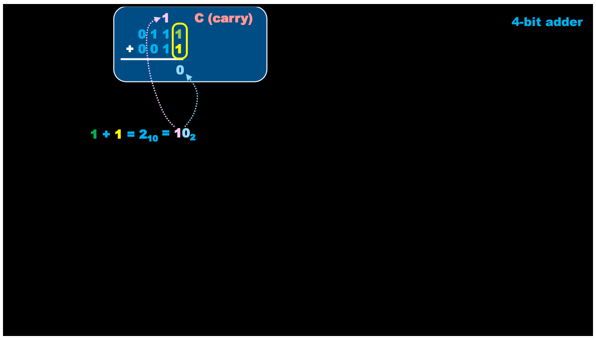

Pascaline (1642)
https://kids.britannica.com/students/article/Pascaline/332603/media
Blaise Pascal
1623 - 1662
https://en.wikipedia.org/wiki/Blaise_Pascal

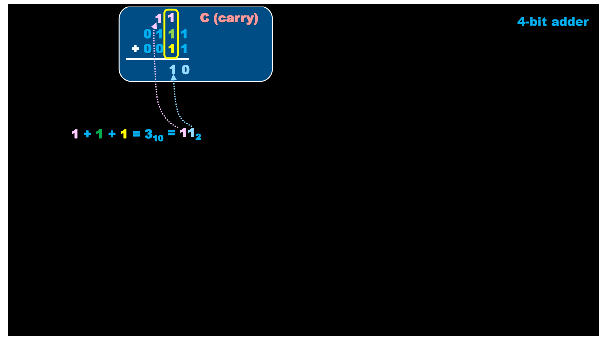

4-bit adder
Input signals (binary digits)
Output signals (binary digits)


0 1 1 1 A = 7 (0 + 4 + 2 + 1)
+ 0 0 1 1 B = 3 (0 + 0 + 2 + 1)
1 0 1 0 S = 10 (8 + 0 + 2 + 0)
4-bit adder
0 0
1 0
1 1
1 1
1
0
1
0


0 1 1 1 A = 7 (0 + 4 + 2 + 1)
+ 0 0 1 1 B = 3 (0 + 0 + 2 + 1)
1 0 1 0 S = 10 (8 + 0 + 2 + 0)
4-bit adder
A_3 B_3
A_2 B_2
A_1 B_1
A_0 B_0
S_3
S_2
S_1
S_0

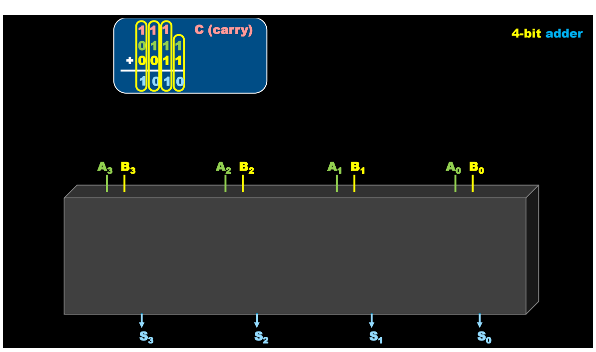

Quiz
What is the decimal result of adding the following binary numbers?
0 1 1 1
+ 0 0 1 1
a) 122_{10}
b) 9_{10}
c) 10_{10}
d) 11_{10}

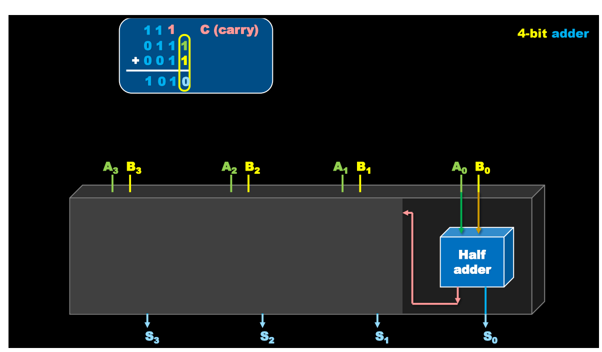

C (carry)
4-bit adder
$1 + 1 = 2_{10} = 10_2$

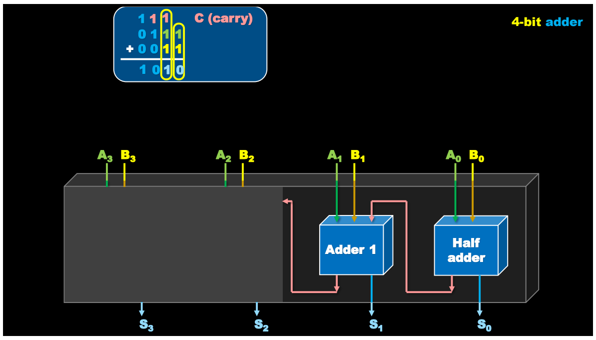

C (carry)
4-bit adder
$1 + 1 + 1 = 3_{10} = 11_2$

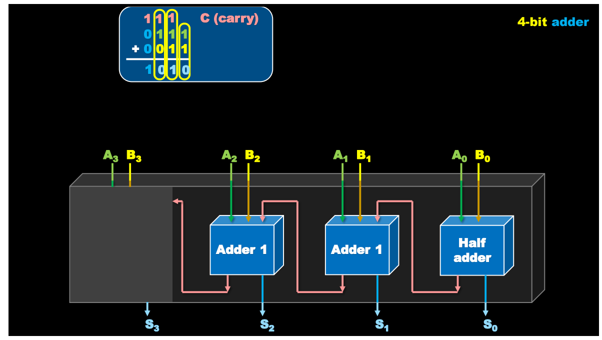

C (carry)
4-bit adder
$1 + 1 + 0 = 2_{10} = 10_2$

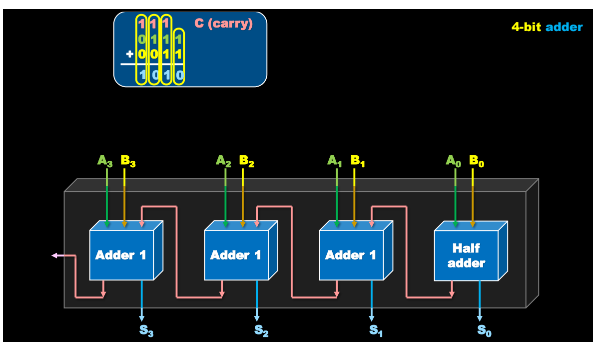

Question
What have we learnt from this exercise?

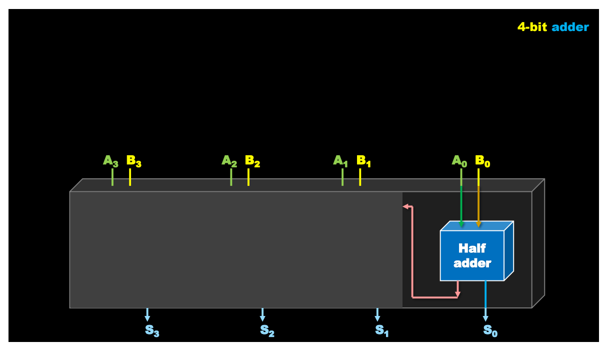

C (carry)
4-bit adder
A_3 B_3
A_2 B_2
A_1 B_1
A_0 B_0
S_3
S_2
S_1
S_0


C (carry)
4-bit adder
A_3 B_3
A_2 B_2
A_1 B_1
A_0 B_0
Half adder
S_3
S_2
S_1
S_0

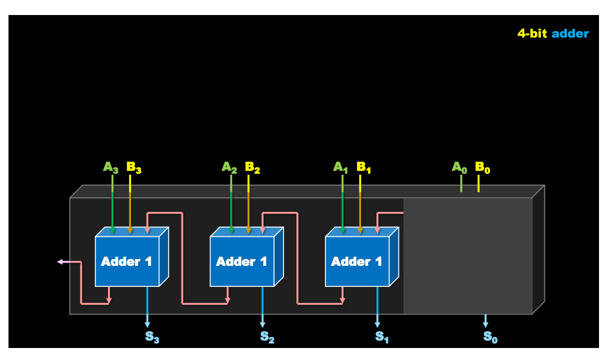




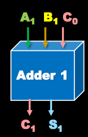




A_0	B_0	C_0	S_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



Adder



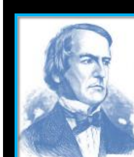
A	B	C_0	C_1	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Agenda

- Adders
- **Gates**
- Register
- Decoder & multiplexer

Architecture of a popular microprocessor

Boolean algebra

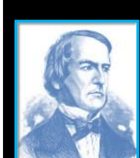

George Boole
1815 – 1864

< {F, T}, T, and, or, not >

A	B	A and B A · B
0	0	0
0	1	0
1	0	0
1	1	1

C: `A && B`
Python: `A and B`

Boolean algebra

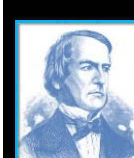

George Boole
1815 – 1864

< {F, T}, T, and, or, not >

A	B	A and B A · B	A or B
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

C: `A || B`
Python: `A or B`

Boolean algebra

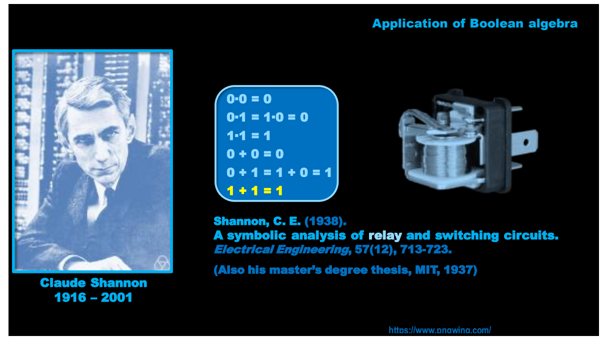

George Boole
1815 – 1864

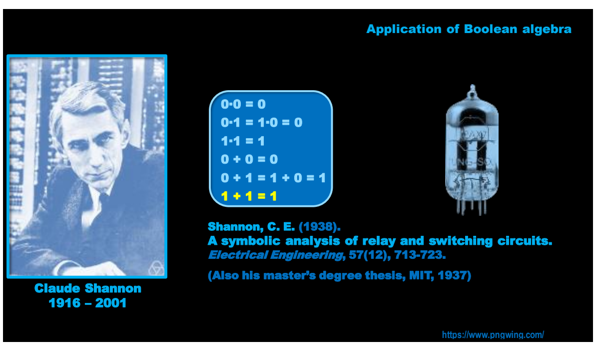
< {F, T}, T, and, or, not >

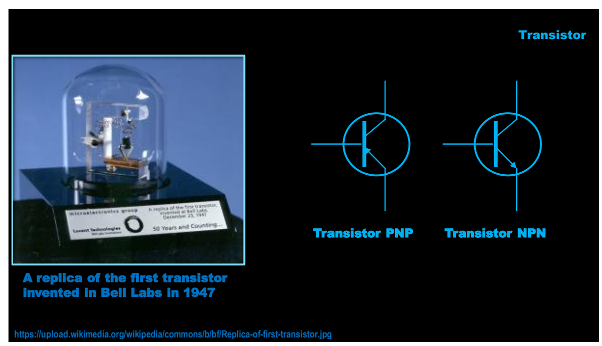
A	B	A and B A · B	A or B	not A
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

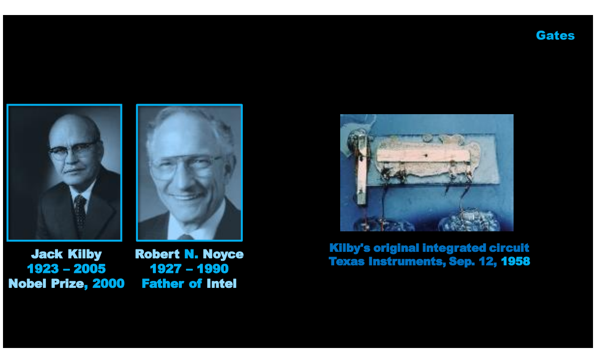
C: `! A`
Python: `not A`

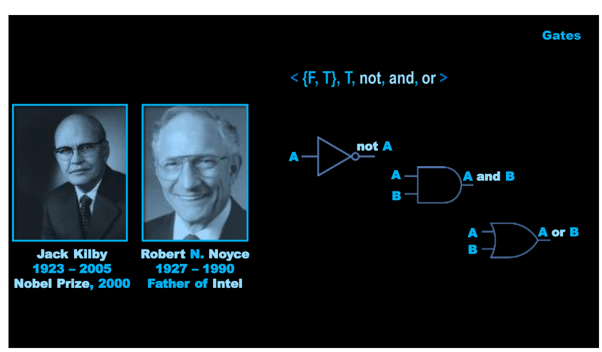

Boolean algebra
George Boole
1815 – 1864
A = 0, 1
B = 0, 1
A and B = A * B
not A = 1 – A
A or B = ~~A + B~~
A or B = A + B – A*B

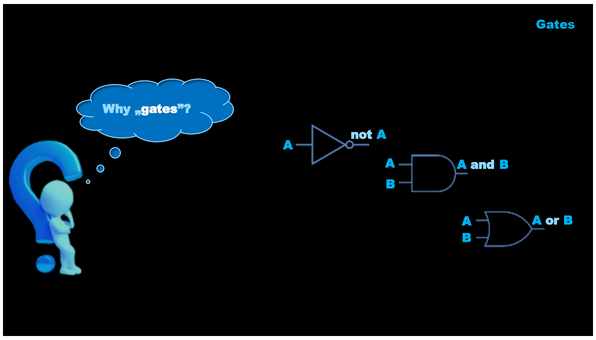

Application of Boolean algebra
Claude Shannon
1916 – 2001
0·0 = 0
0·1 = 1·0 = 0
1·1 = 1
0 + 0 = 0
0 + 1 = 1 + 0 = 1
1 + 1 = 1
Shannon, C. E. (1938).
A symbolic analysis of relay and switching circuits.
Electrical Engineering, 57(12), 713-723.
(Also his master's degree thesis, MIT, 1937)
https://www.pngwing.com/

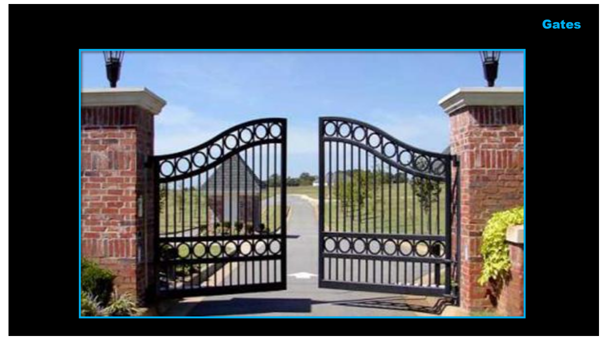

Application of Boolean algebra
Claude Shannon
1916 – 2001
0·0 = 0
0·1 = 1·0 = 0
1·1 = 1
0 + 0 = 0
0 + 1 = 1 + 0 = 1
1 + 1 = 1
Shannon, C. E. (1938).
A symbolic analysis of relay and switching circuits.
Electrical Engineering, 57(12), 713-723.
(Also his master's degree thesis, MIT, 1937)
https://www.pngwing.com/

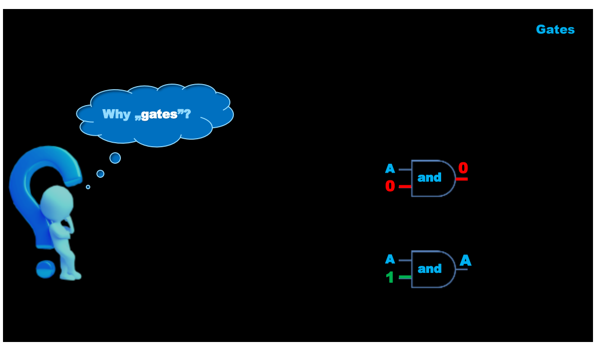

Transistor
A replica of the first transistor
invented in Bell Labs in 1947
Transistor PNP
Transistor NPN
https://upload.wikimedia.org/wikipedia/commons/b/bf/Replica-of-first-transistor.jpg

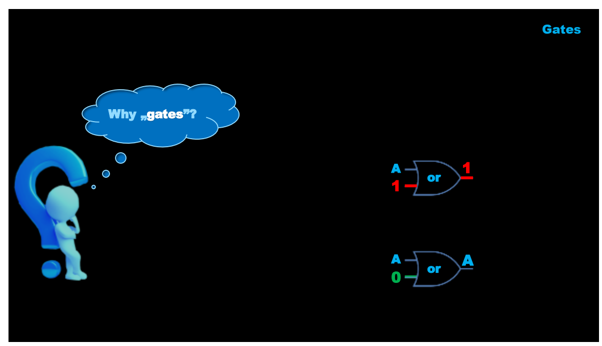

Gates
Jack Kilby
1923 – 2005
Nobel Prize, 2000
Robert N. Noyce
1927 – 1990
Father of Intel
Kilby's original integrated circuit
Texas Instruments, Sep. 12, 1958

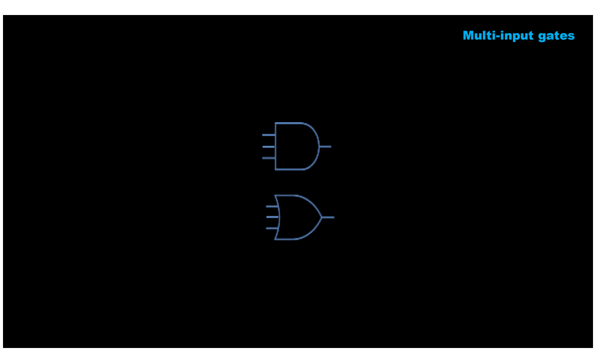

Gates
< {F, T}, T, not, and, or >
Jack Kilby
1923 – 2005
Nobel Prize, 2000
Robert N. Noyce
1927 – 1990
Father of Intel
A
not A
A
B
A and B
A
B
A or B

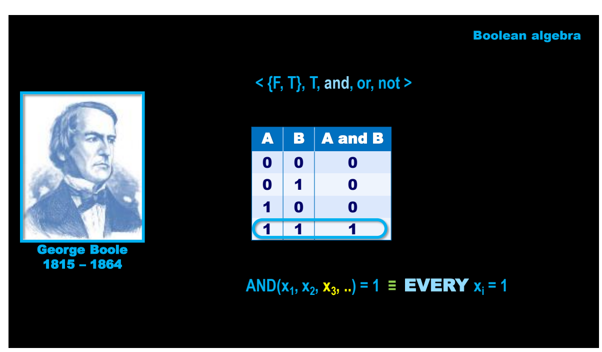

Gates
Why „gates"?
A
not A
A
B
A and B
A
B
A or B

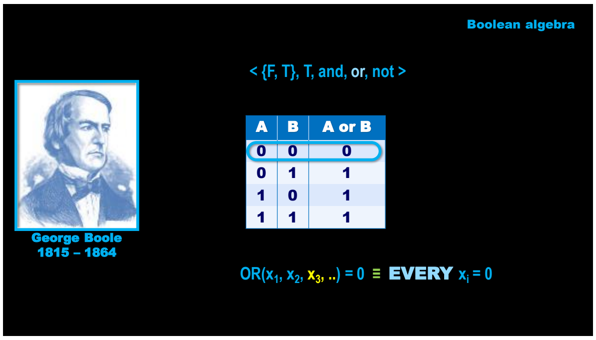

Gates

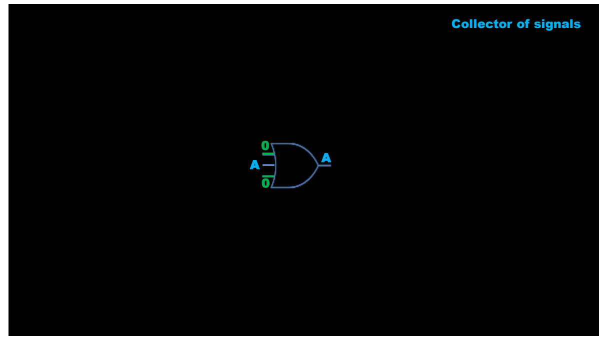

Gates
Why „gates"?
A
0
and
0
A
1
and
A

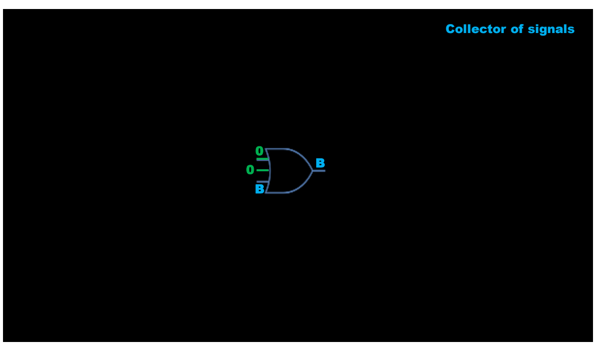

Gates
Why „gates"?
A
1
or
1
A
0
or
A

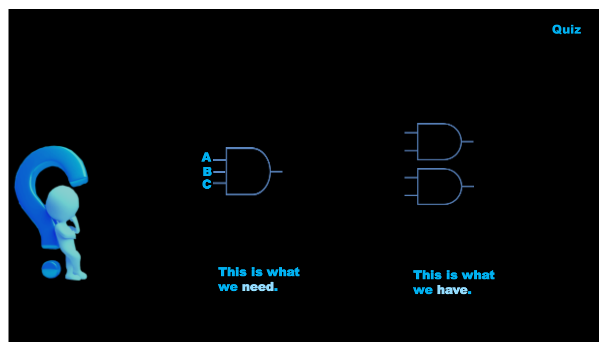

Multi-input gates

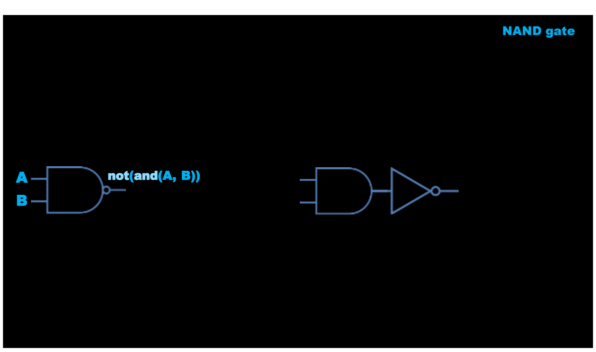

Boolean algebra
< {F, T}, T, and, or, not >
A B A and B
0 0 0
0 1 0
1 0 0
1 1 1
George Boole
1815 – 1864
AND(x_1, x_2, x_3, ..) = 1 ≡ EVERY x_i = 1

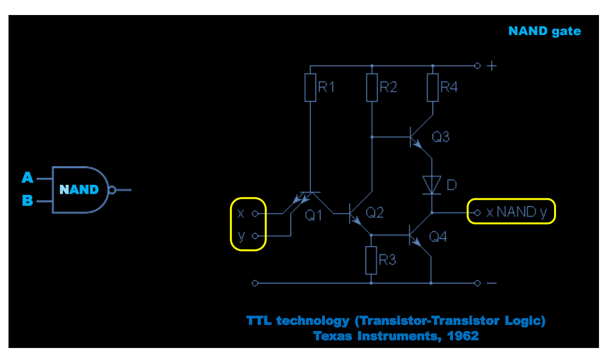

Boolean algebra
< {F, T}, T, and, or, not >
A | B | A or B
0 | 0 | 0
0 | 1 | 1
1 | 0 | 1
1 | 1 | 1
George Boole
1815 – 1864
$OR(x_1, x_2, x_3, ..) = 0 \equiv$ EVERY $x_i = 0$

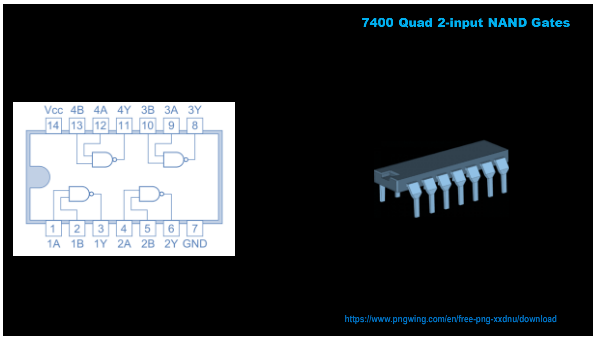

Collector of signals
0
A
0
A

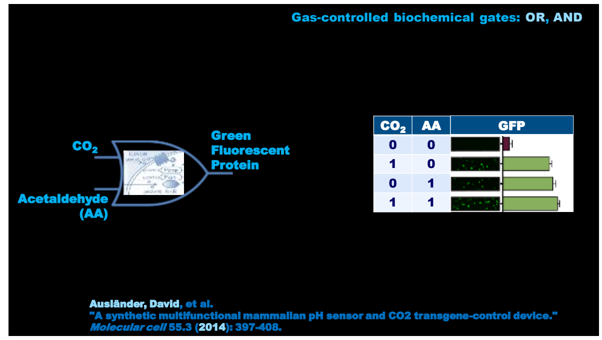

Collector of signals
0
0
B
B

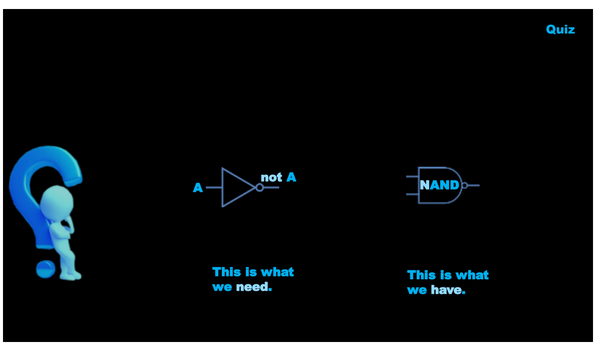

Quiz
A
B
C
This is what
we need.
This is what
we have.

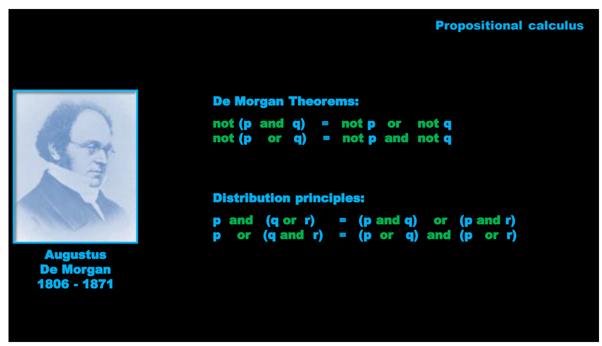

NAND gate
A
B
not(and(A, B))

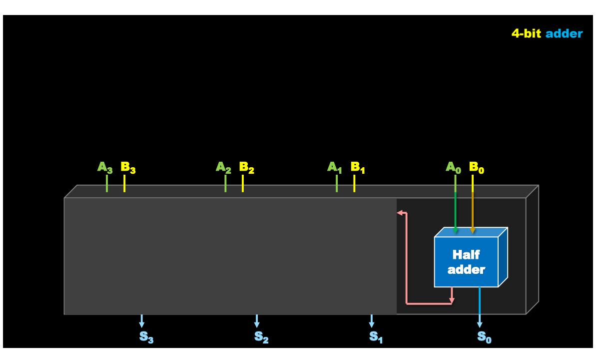

NAND gate
A
B
NAND
R1
R2
R4
+
Q3
D
x
y
Q1
Q2
Q4
x NAND y
R3
−
TTL technology (Transistor-Transistor Logic)
Texas Instruments, 1962

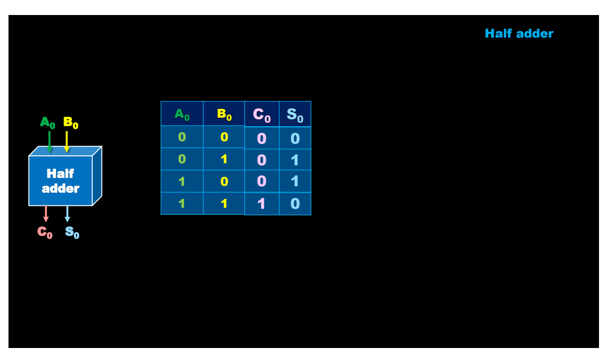

7400 Quad 2-input NAND Gates
Vcc 4B 4A 4Y 3B 3A 3Y
14 13 12 11 10 9 8
1 2 3 4 5 6 7
1A 1B 1Y 2A 2B 2Y GND
https://www.pngwing.com/en/free-png-xxdnu/download

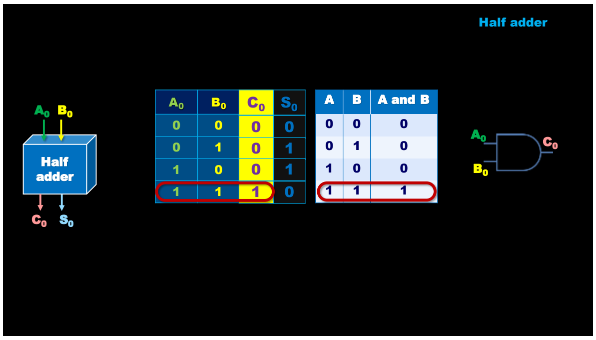

Gas-controlled biochemical gates: OR, AND
CO_2
Acetaldehyde (AA)
Green Fluorescent Protein
CO_2 AA GFP
0 0
1 0
0 1
1 1
Ausländer, David, et al.
"A synthetic multifunctional mammalian pH sensor and CO2 transgene-control device."
Molecular cell 55.3 (2014): 397-408.

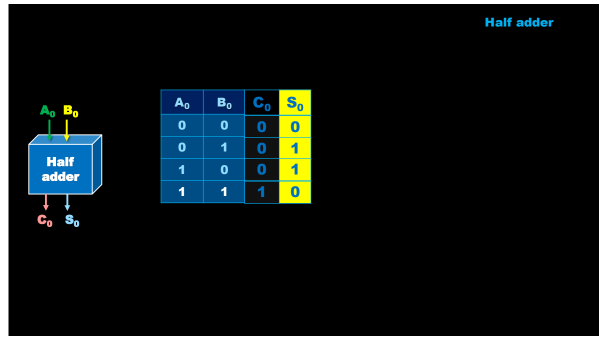

Quiz
A
not A
NAND
This is what we need.
This is what we have.

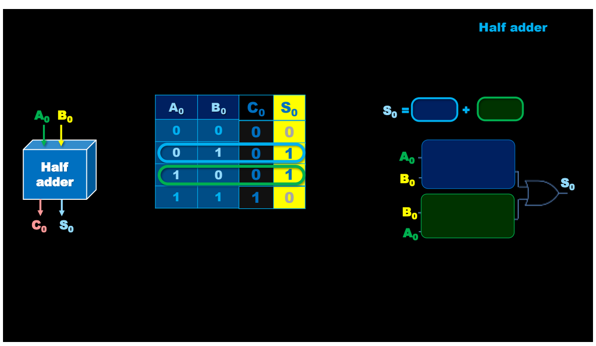

Propositional calculus
Augustus De Morgan 1806 - 1871
De Morgan Theorems:
not (p and q) = not p or not q
not (p or q) = not p and not q
Distribution principles:
p and (q or r) = (p and q) or (p and r)
p or (q and r) = (p or q) and (p or r)

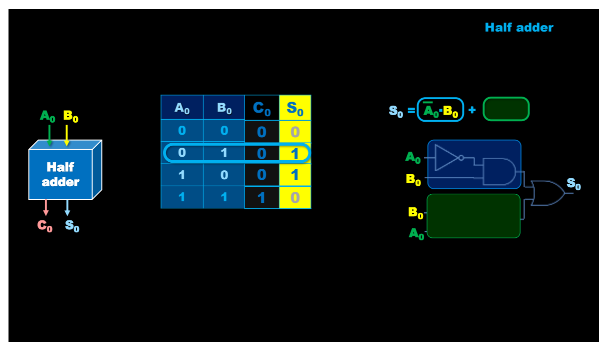

4-bit adder
A_3 B_3
A_2 B_2
A_1 B_1
A_0 B_0
Half adder
S_3
S_2
S_1
S_0

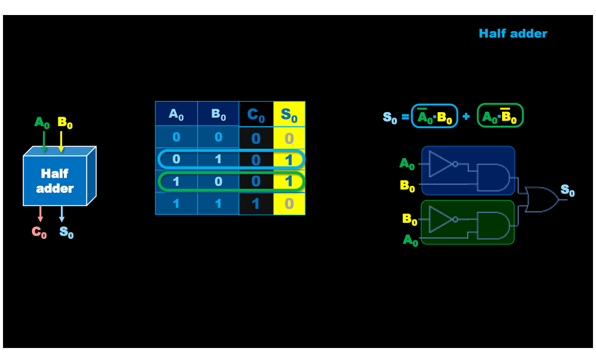

Half adder
A_0 B_0
Half adder
C_0 S_0
A_0 B_0 C_0 S_0
0 0 0 0
0 1 0 1
1 0 0 1
1 1 1 0

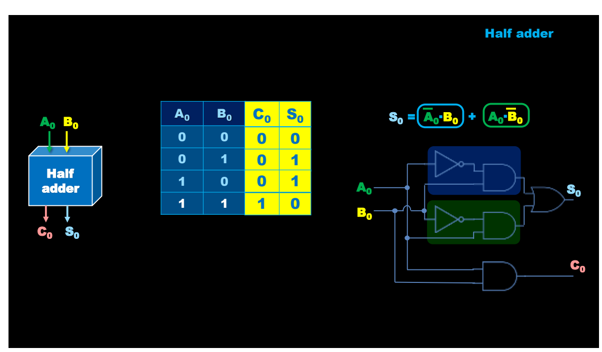

Half adder
A0 B0
Half adder
C0 S0
A0 B0 C0 S0
0 0 0 0
0 1 0 1
1 0 0 1
1 1 1 0
A B A and B
0 0 0
0 1 0
1 0 0
1 1 1
A0
B0
C0

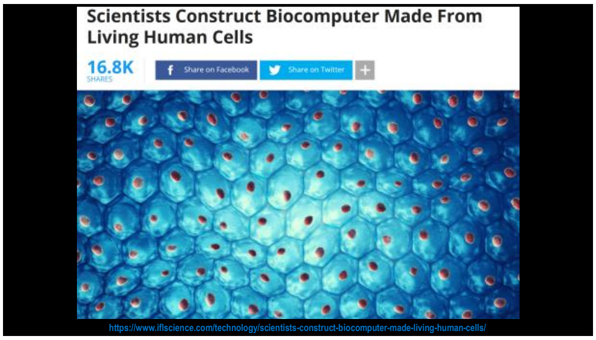

Half adder
A0 B0
Half adder
C0 S0
A0 B0 C0 S0
0 0 0 0
0 1 0 1
1 0 0 1
1 1 1 0

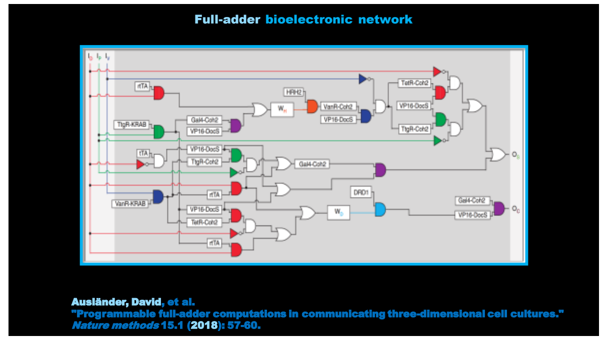

Half adder
A0 B0
Half adder
C0 S0
A0 B0 C0 S0
0 0 0 0
0 1 0 1
1 0 0 1
1 1 1 0
S0 = +
A0
B0
B0
A0
S0


Half adder
A0 B0
Half adder
C0 S0
A0 B0 C0 S0
0 0 0 0
0 1 0 1
1 0 0 1
1 1 1 0
S0 = $\overline{A_0}\cdot B_0$ +
A0
B0
B0
A0
S0

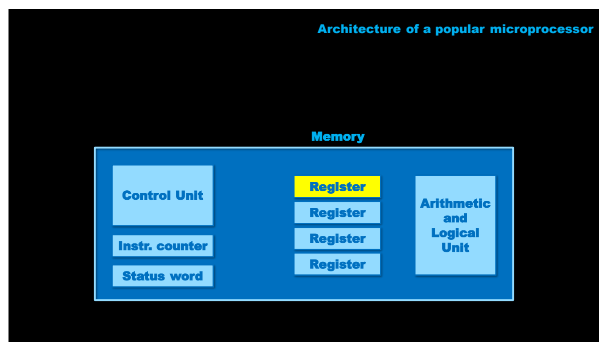

Half adder
A0 B0
Half adder
C0 S0
A0 B0 C0 S0
0 0 0 0
0 1 0 1
1 0 0 1
1 1 1 0
S0 = $\overline{A_0}\cdot B_0$ + $A_0\cdot\overline{B_0}$
A0
B0
B0
A0
S0

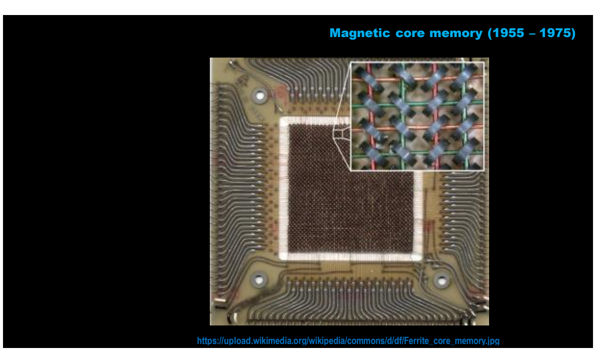

Half adder
A0 B0
Half adder
C0 S0
A0 B0 C0 S0
0 0 0 0
0 1 0 1
1 0 0 1
1 1 1 0
S0 = $\overline{A_0}\cdot B_0$ + $A_0\cdot\overline{B_0}$
A0
B0
S0
C0

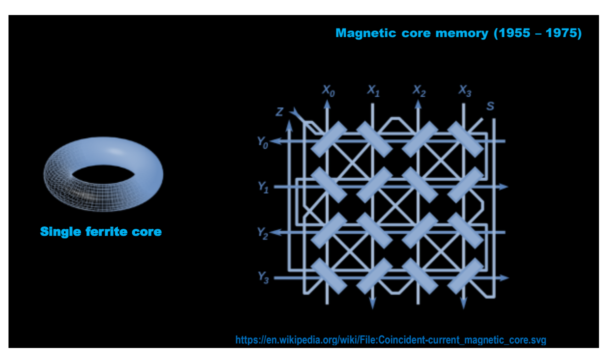

Scientists Construct Biocomputer Made From Living Human Cells
16.8K
SHARES
Share on Facebook
Share on Twitter
https://www.iflscience.com/technology/scientists-construct-biocomputer-made-living-human-cells/

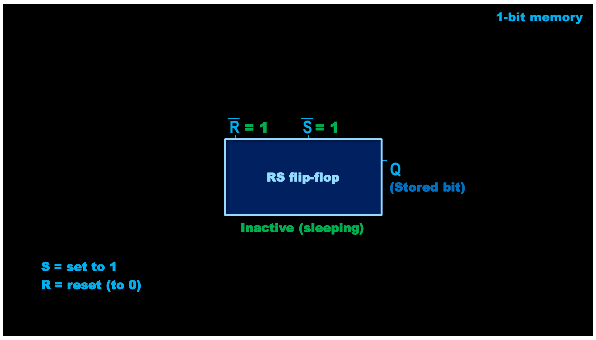

Full-adder bioelectronic network
Ausländer, David, et al.
"Programmable full-adder computations in communicating three-dimensional cell cultures."
Nature methods 15.1 (2018): 57-60.

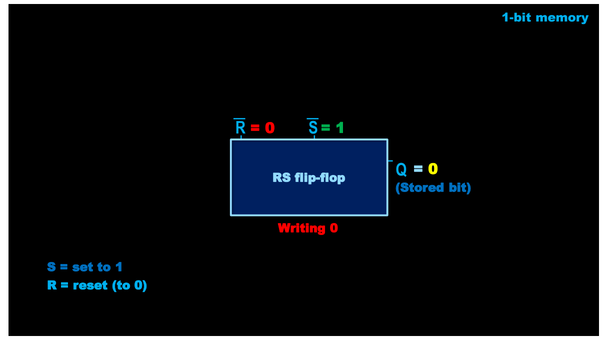

Agenda
• Adders
• Gates
• Register
• Decoder & multiplexer

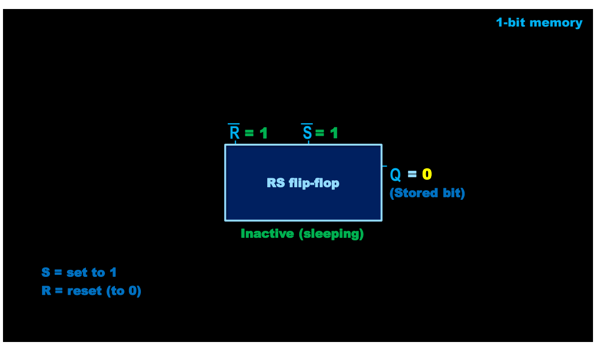

Architecture of a popular microprocessor
Memory
Control Unit
Instr. counter
Status word
Register
Register
Register
Register
Arithmetic and Logical Unit

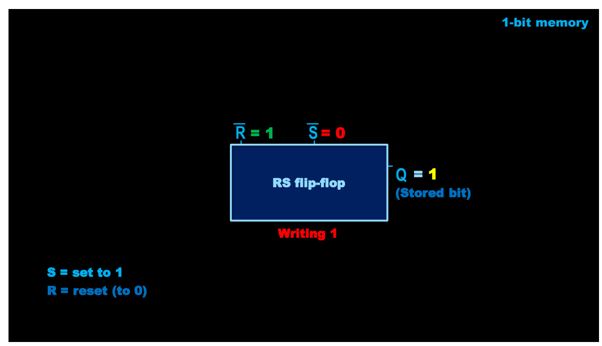

Magnetic core memory (1955 – 1975)
https://upload.wikimedia.org/wikipedia/commons/d/df/Ferrite_core_memory.jpg

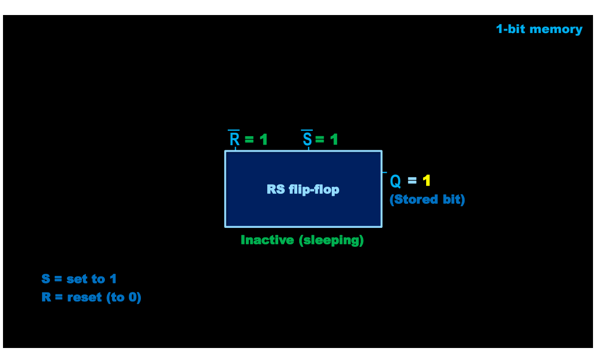

Magnetic core memory (1955 – 1975)
Single ferrite core
X_0 X_1 X_2 X_3
S
Z
Y_0
Y_1
Y_2
Y_3
https://en.wikipedia.org/wiki/File:Coincident-current_magnetic_core.svg

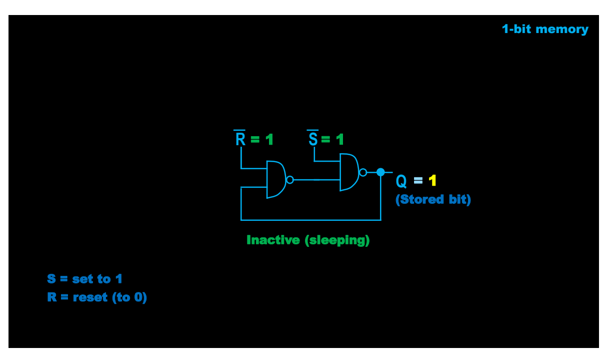

1-bit memory
$\overline{R}$ = 1 $\overline{S}$ = 1
RS flip-flop
Q
(Stored bit)
Inactive (sleeping)
S = set to 1
R = reset (to 0)

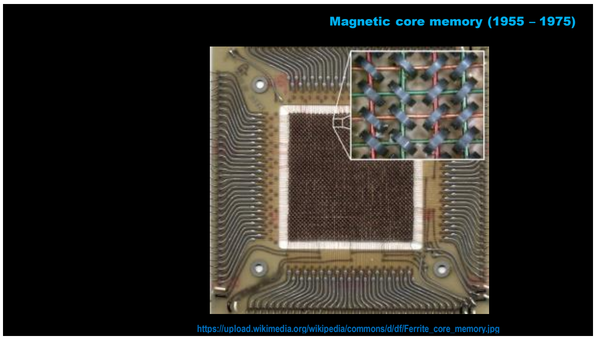

1-bit memory
$\overline{R}$ = 0 $\overline{S}$ = 1
RS flip-flop
Q = 0
(Stored bit)
Writing 0
S = set to 1
R = reset (to 0)

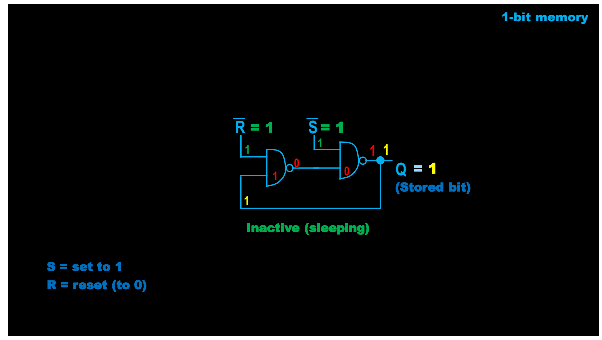

1-bit memory
$\overline{R}$ = 1 $\overline{S}$ = 1
RS flip-flop
Q = 0
(Stored bit)
Inactive (sleeping)
S = set to 1
R = reset (to 0)

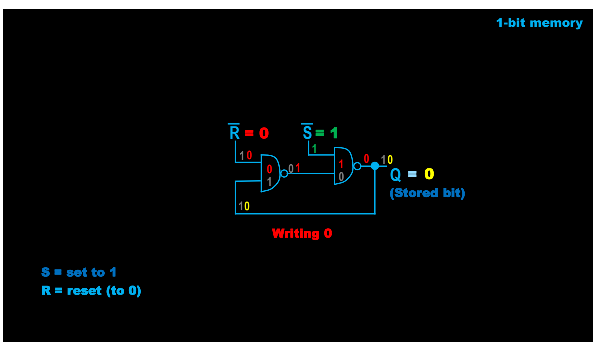

1-bit memory
$\overline{R}$ = 1 $\overline{S}$ = 0
RS flip-flop
Q = 1
(Stored bit)
Writing 1
S = set to 1
R = reset (to 0)

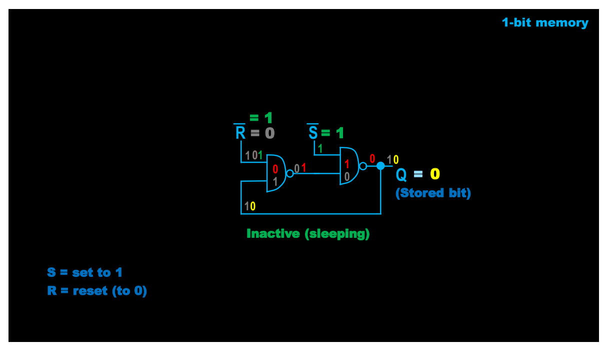

1-bit memory
$\overline{R}$ = 1 $\overline{S}$ = 1
RS flip-flop
Q = 1
(Stored bit)
Inactive (sleeping)
S = set to 1
R = reset (to 0)

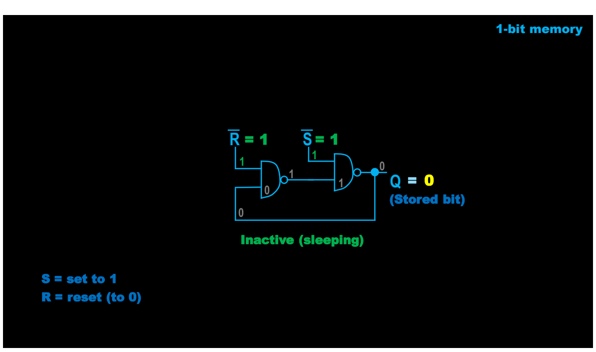

1-bit memory
$\overline{R}$ = 1 $\overline{S}$ = 1
Q = 1
(Stored bit)
Inactive (sleeping)
S = set to 1
R = reset (to 0)

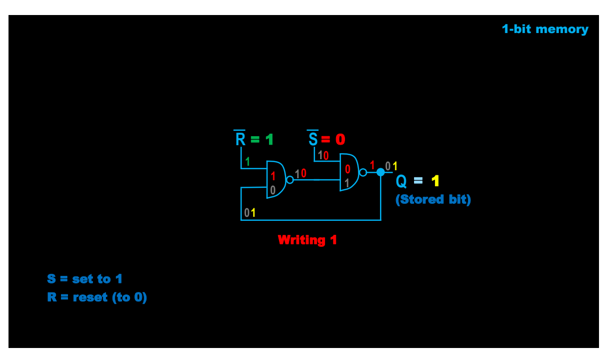

Magnetic core memory (1955 – 1975)
https://upload.wikimedia.org/wikipedia/commons/d/df/Ferrite_core_memory.jpg

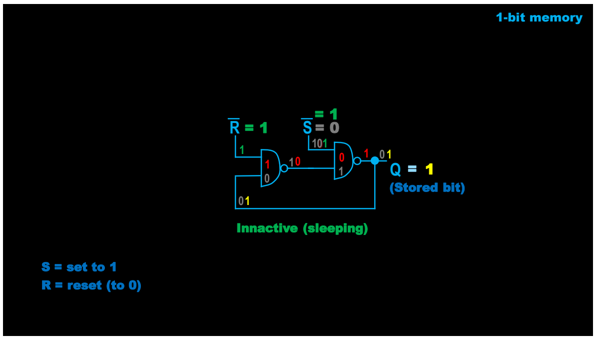

1-bit memory
R = 1
S = 1
Q = 1
(Stored bit)
Inactive (sleeping)
S = set to 1
R = reset (to 0)

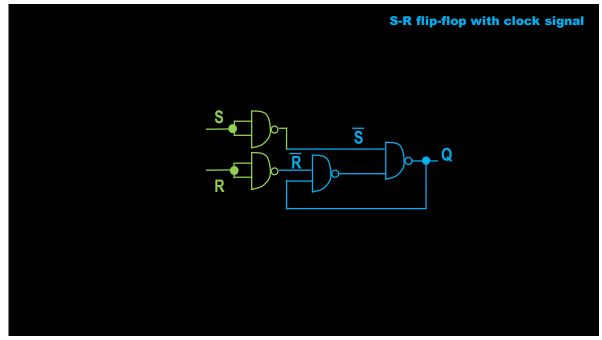

1-bit memory
R = 0
S = 1
Q = 0
(Stored bit)
Writing 0
S = set to 1
R = reset (to 0)

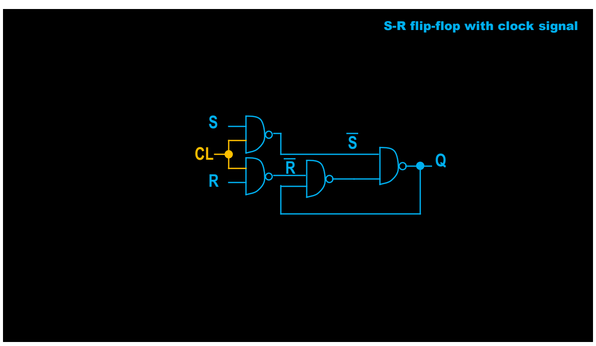

1-bit memory
= 1
R = 0
S = 1
Q = 0
(Stored bit)
Inactive (sleeping)
S = set to 1
R = reset (to 0)

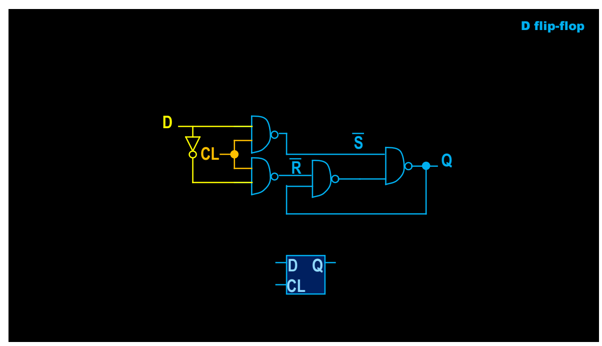

1-bit memory
R = 1
S = 1
Q = 0
(Stored bit)
Inactive (sleeping)
S = set to 1
R = reset (to 0)

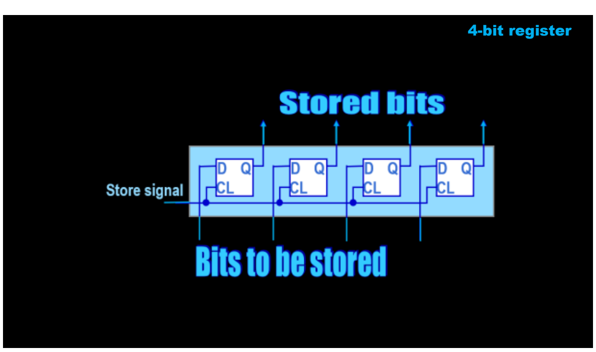

1-bit memory
R = 1
S = 0
Q = 1
(Stored bit)
Writing 1
S = set to 1
R = reset (to 0)

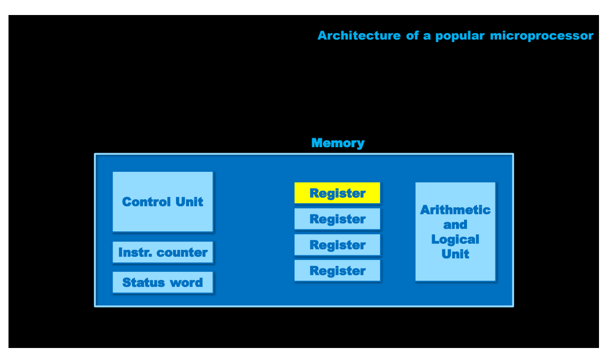

1-bit memory
R = 1
S = 0
Q = 1
(Stored bit)
Innactive (sleeping)
S = set to 1
R = reset (to 0)


S-R flip-flop with clock signal
S
R
S
R
Q

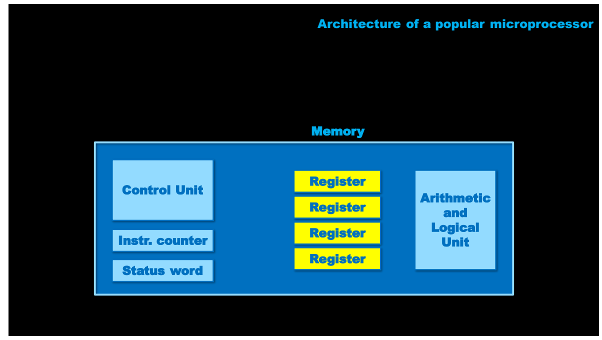

S-R flip-flop with clock signal
S
CL
R
S
R
Q

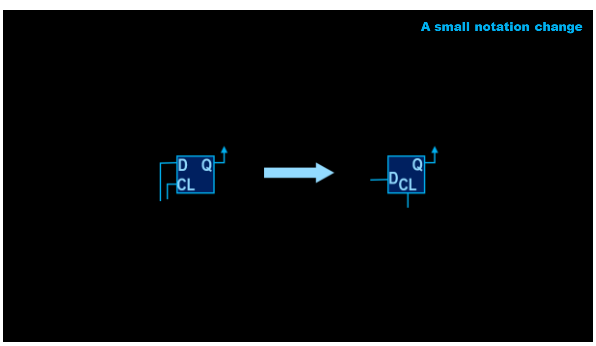

D flip-flop
D
CL
S
R
Q
D Q
CL

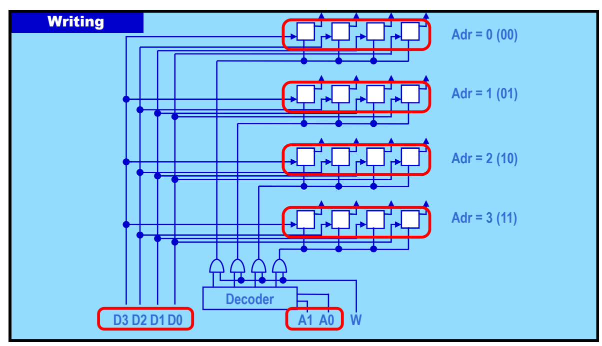

4-bit register
Stored bits
Store signal
D Q
CL
Bits to be stored

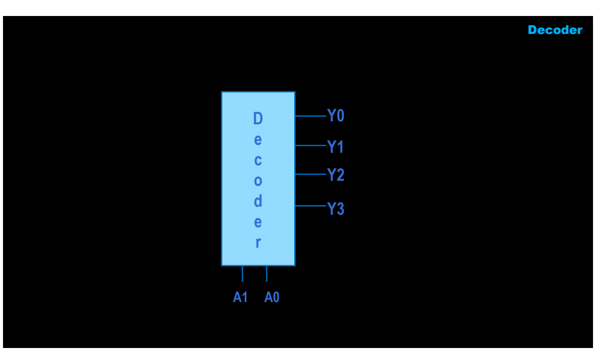

Architecture of a popular microprocessor
Memory
Control Unit
Instr. counter
Status word
Register
Register
Register
Register
Arithmetic and Logical Unit

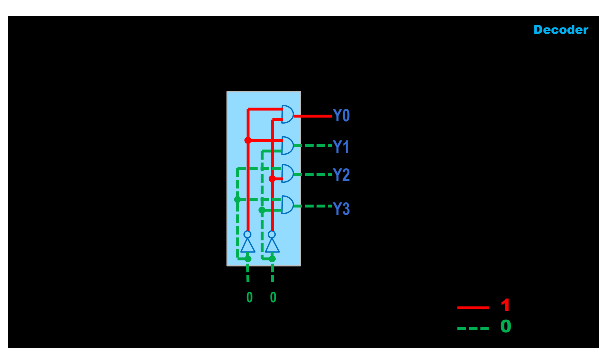

Agenda
• Adders
• Gates
• Register
• Decoder & multiplexer

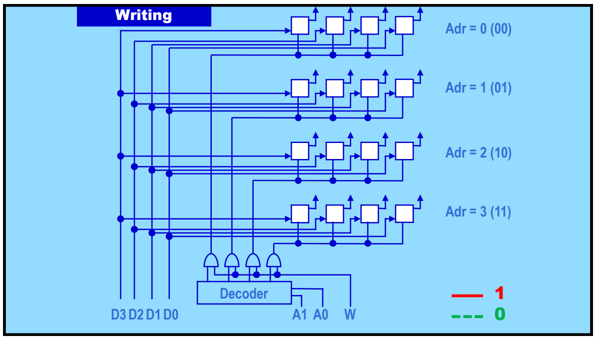

Architecture of a popular microprocessor
Memory
Control Unit
Instr. counter
Status word
Register
Register
Register
Register
Arithmetic and Logical Unit

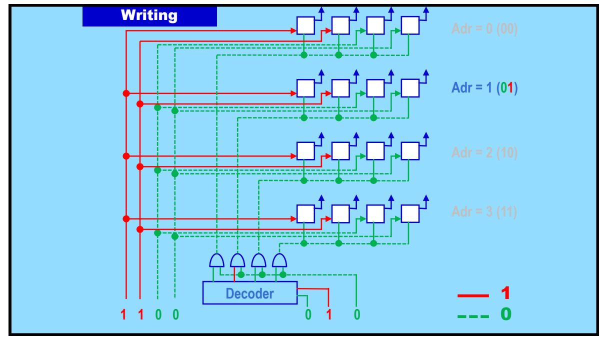

A small notation change
D Q
CL
D Q
CL

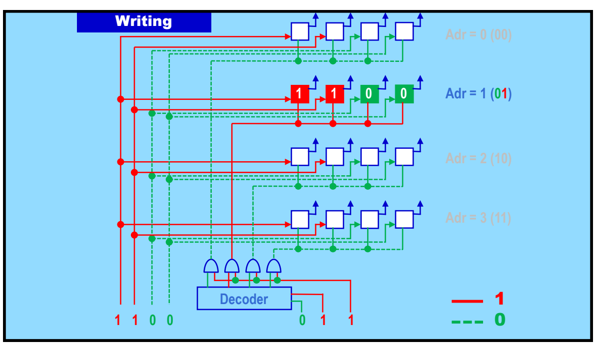

Writing
Adr = 0 (00)
Adr = 1 (01)
Adr = 2 (10)
Adr = 3 (11)
Decoder
D3 D2 D1 D0
A1 A0
W

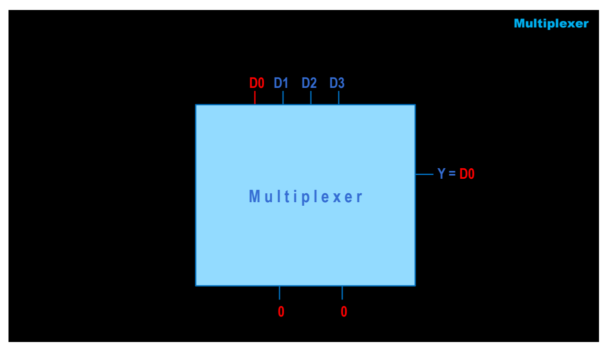

Decoder
D e c o d e r
Y0
Y1
Y2
Y3
A1 A0

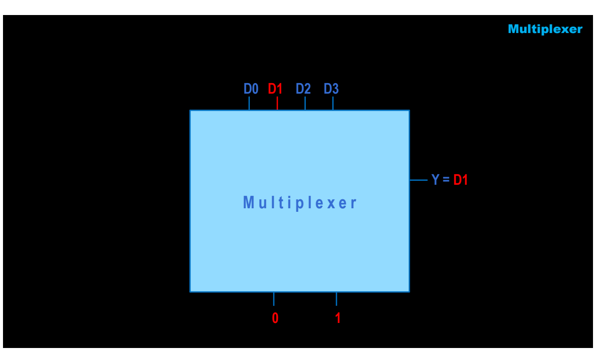

Decoder
Y0
Y1
Y2
Y3
0 0
1
0

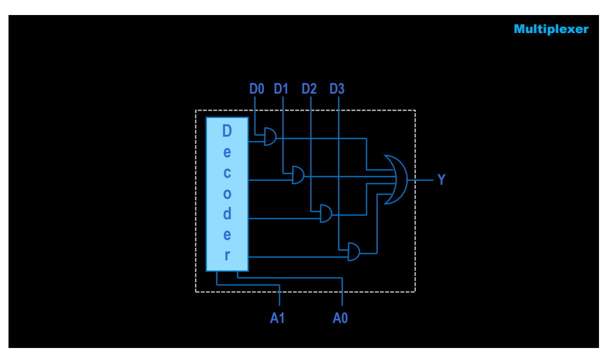

Writing
Adr = 0 (00)
Adr = 1 (01)
Adr = 2 (10)
Adr = 3 (11)
Decoder
D3 D2 D1 D0
A1 A0 W
1
0

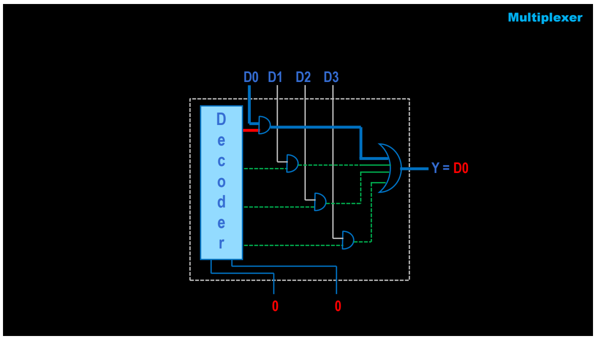

Writing
Adr = 0 (00)
Adr = 1 (01)
Adr = 2 (10)
Adr = 3 (11)
Decoder
1 1 0 0
0 1 0
1
0

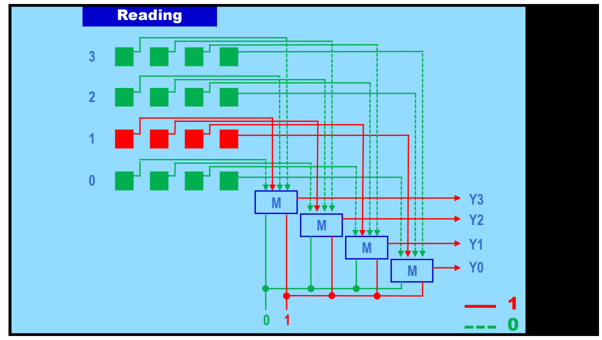

Writing
Adr = 0 (00)
1 1 0 0
Adr = 1 (01)
Adr = 2 (10)
Adr = 3 (11)
Decoder
1 1 0 0
0 1 1
1
0


Multiplexer
D0 D1 D2 D3
Multiplexer
Y = D0
0 0

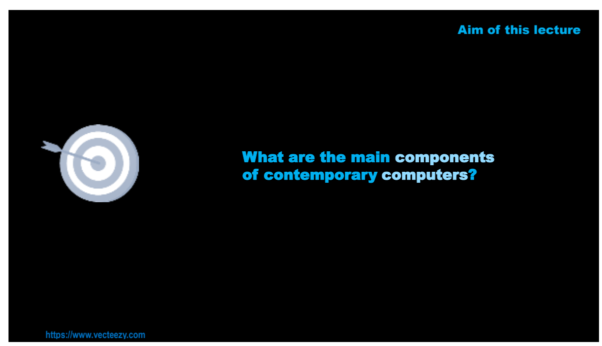

Multiplexer
D0 D1 D2 D3
Multiplexer
Y = D1
0 1

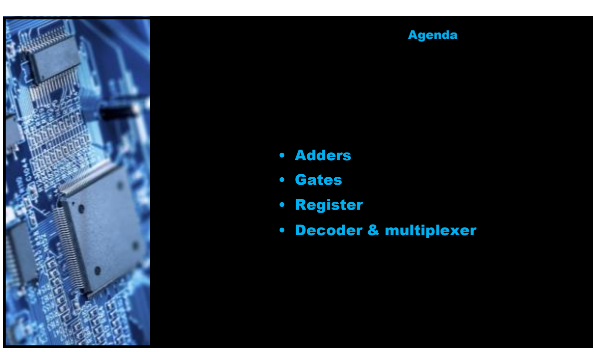

Multiplexer
D0 D1 D2 D3
Decoder
Y
A1 A0

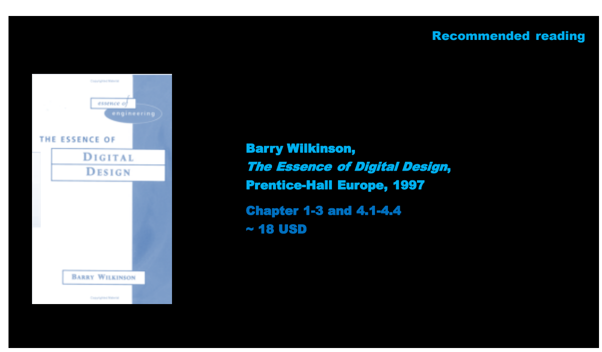

Multiplexer
D0 D1 D2 D3
Decoder
Y = D0
0
0

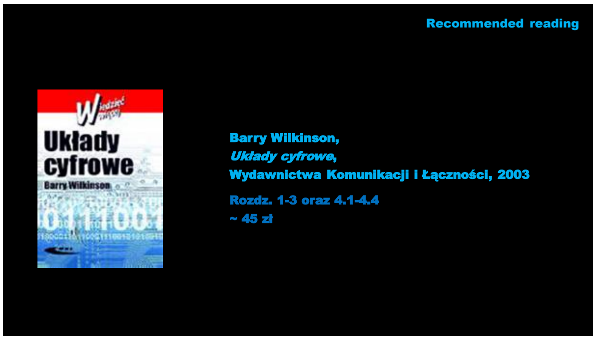

Reading
3
2
1
0
M
M
M
M
Y3
Y2
Y1
Y0
0 1
1
0


Summary

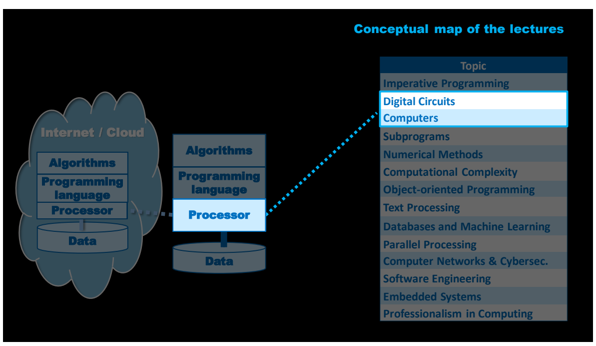

Aim of this lecture
What are the main components of contemporary computers?
https://www.vecteezy.com


Agenda
• Adders
• Gates
• Register
• Decoder & multiplexer


Recommended reading
THE ESSENCE OF DIGITAL DESIGN
BARRY WILKINSON
Barry Wilkinson,
The Essence of Digital Design,
Prentice-Hall Europe, 1997
Chapter 1-3 and 4.1-4.4
~ 18 USD


Recommended reading
Układy cyfrowe
Barry Wilkinson
Barry Wilkinson,
Układy cyfrowe,
Wydawnictwa Komunikacji i Łączności, 2003
Rozdz. 1-3 oraz 4.1-4.4
~ 45 zł


Thank you for your attention!
Questions?
https://pngtree.com/


Conceptual map of the lectures
Internet / Cloud
Algorithms
Programming language
Processor
Data
Algorithms
Programming language
Processor
Data
Topic
Imperative Programming
Digital Circuits
Computers
Subprograms
Numerical Methods
Computational Complexity
Object-oriented Programming
Text Processing
Databases and Machine Learning
Parallel Processing
Computer Networks & Cybersec.
Software Engineering
Embedded Systems
Professionalism in Computing