QC10

Monday, March 8, 2021 3:41 PM

From Logical Algebra to Classical Computers

⇒ Logical / Boolean Variables are Binary T and F

- We can replace them with F → 0, T → 1

- Then for example

P	Q	P∨Q
T	T	T
T	F	T
F	T	T
F	F	F

⟹

P	Q	P∨Q
1	1	1
1	0	1
0	1	1
0	0	0

George Boole
1847 Mathematical Analysis of Logics

⇒ Gates

Logics → Logical Algebra (13th) → Circuits can be designed to perform logical Algebra

Claude Shannon (MIT) (20th) 1937 Symbolic Analysis of Relay and switching Circuits

Ancient Greece
Aristotle Organon 384-322BC

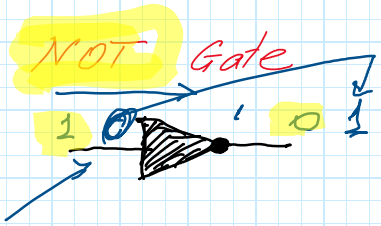
Start with:

- At discrete time intervals either a pulse of electricity is transmitted or it is not transmitted

Receiving
- transmitted → T → 1
- not transmitted → F → 0

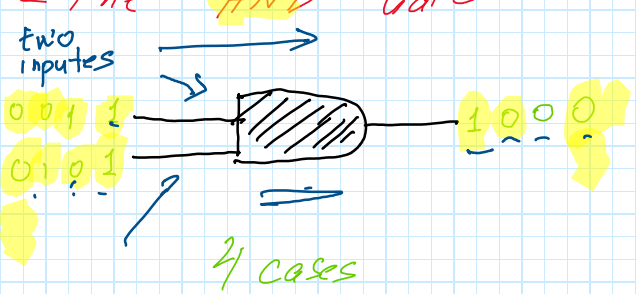
- Combination of switches that corresponds to discussed logical Operators - called Gates

- The NOT Gate



time interval.

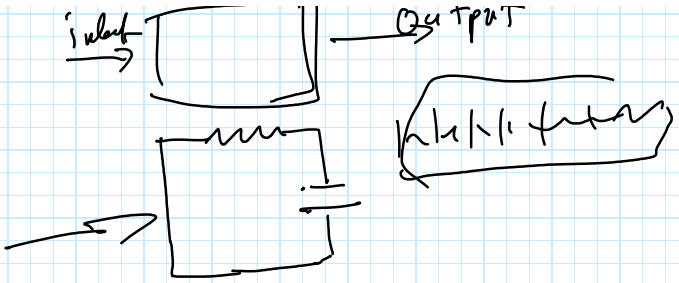
- The AND Gate



- The OR Gate

- The NAND Gate

$\neg(x \wedge y)$

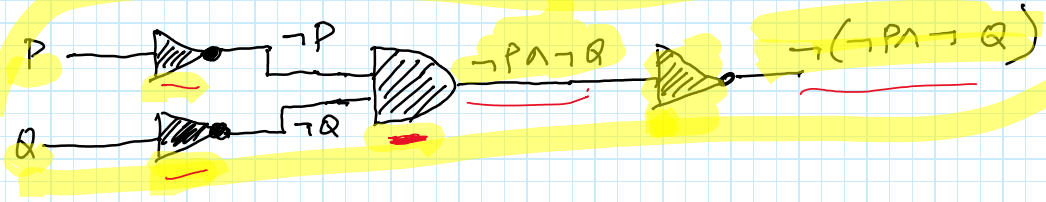


$\Rightarrow$ Circuits

- Connecting Gates forms a circuit
- Circuits are linear and beeng read from Left to right
- Electrical Pulse enters from the Left and being read at the Right

Examples

$\neg(\neg P \wedge \neg Q)$



(or)

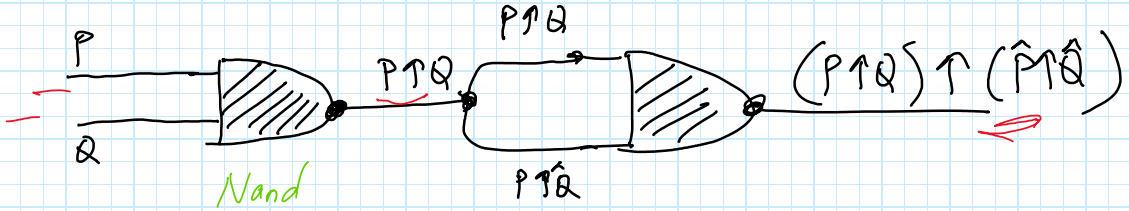
recall that $\neg(\neg P \wedge \neg Q) \equiv P \vee Q$

- Example $P \uparrow P$ NAND

Splitting the input — fan-out

- $P \uparrow P \equiv \neg P$

- Example $(P \uparrow Q) \uparrow (P \uparrow Q) \equiv P \wedge Q$



$\Rightarrow$ NAND is a Universal Gate

- for NAND any function can be expressed using NAND — Universal Gate

⟹ Gates and Computation

- Gates — fundamental building blocks of modern computers

- Gates - can be used to compute

- Some example

Recall exclusive or ⊕

F → 0
T → 1

P	Q	P⊕Q
T	T	F
T	F	T
F	T	T
F	F	F

⟹

P	Q	P⊕Q
F	F	F
F	T	T
T	F	T
T	T	F

⟹

P	Q	P⊕Q
0	0	0
0	1	1
1	0	1
1	1	0

- This can be compared to adding odd and even numbers

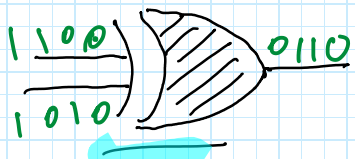
I even + even = even
II even + odd = odd

} addition modulo 2

iii) odd + even = odd

iv) odd + odd = even

- ⊕ → gate is called XOR Gate



- Express through NAND

- Using this we can construct a half-adder

- half-adder - circuit adds two binary digits

- Remember Decimal half adder

if the sum < 10 2+4=6, 3+5=8 < 10

if the sum ≥ 10 7+5=12 ⇒ 7+5=2, carry of 1

12 → 1

$$\begin{array}{r} 117 \\ +\ 235 \\ \hline 352 \end{array}$$

- Binary half-adder

if the sum < 2 1+0=1, 0+0=0, 0+1=1

if the sum ≥ 2 1+1=2 ⇒ 1+1=0, carry of 1

10

$$\begin{array}{r} 101 \\ 010 \\ \hline 111 \end{array}$$

$$\begin{array}{r} 101 \\ 101 \\ \hline 1010 \end{array}$$

101 ≡ 5

- Schematically

Schematically

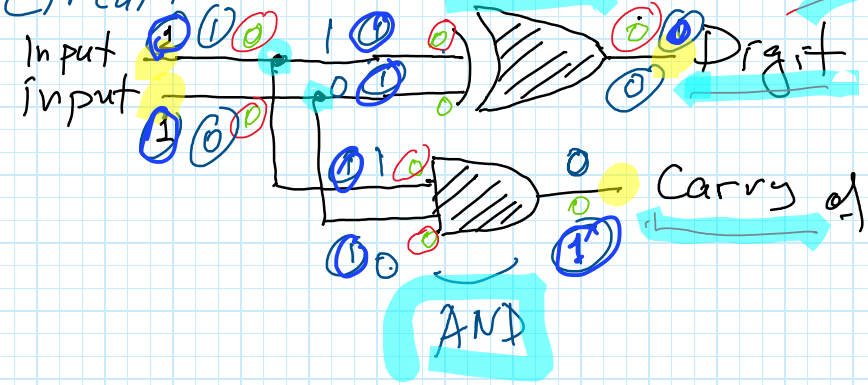
$0 \leftarrow 0 + 0$, carry of 0

$1 \leftarrow 0 + 1$, carry of 0

$1 \leftarrow 1 + 0$, carry of 0

$0 \leftarrow 1 + 1$, carry of 1

- Circuit



reading

64	32	16	8	4	2	1
1	0	0	1	0	1	0

75 | 64+8+2+1

64

11

8

3

2

1

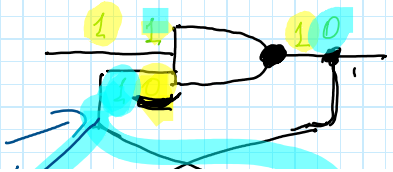
1010 = [illegible]

⇒ Memory

- How to store data → flip-flop

L gate with the feedback

NAND



Puls 1

X AND