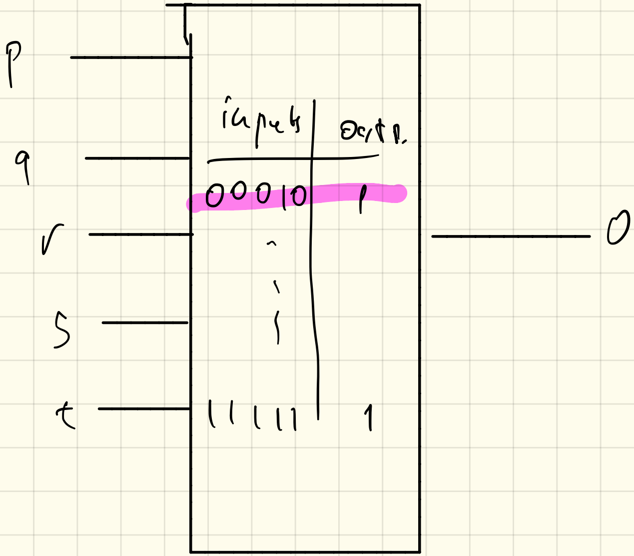
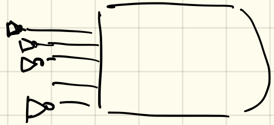
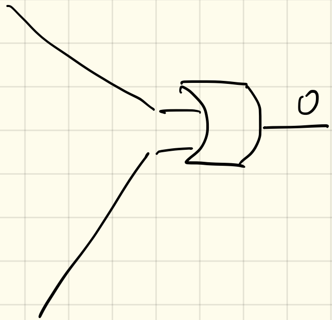
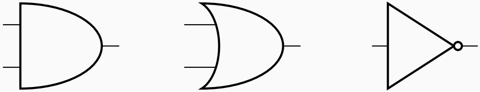

p
q
r
s
t
inputs
outp.
00010 1
11111 1
O




O

Boolean functions

- A function $F : \{0,1\}^k \to \{0,1\}$, where $k \in \mathbb{Z}^+$ is arity of F, is called a Boolean function
- Any Boolean function can be expressed as a combination of $\wedge$, $\vee$, $\neg$



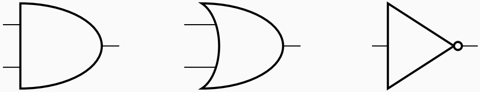
- Do we need all three types of gates?

Boolean functions of arity 2

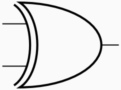
P	Q	$\bot$	$\wedge$	$\neg\Rightarrow$ $\neg(P\Rightarrow Q)$	P	$\neg\Leftarrow$	Q	$\neg\Leftrightarrow$ XOR	$\vee$
1	1	0	1	0	1	0	1	0	1
1	0	0	0	1	1	0	0	1	1
0	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0

P	Q	$\neg\vee$	$\Leftrightarrow$	$\neg Q$	$\Leftarrow$	$\neg P$	$\Rightarrow$	$\neg\wedge$	$\top$
1	1	0	1	0	1	0	1	0	1
1	0	0	0	1	1	0	0	1	1
0	1	0	0	0	0	1	1	1	1
0	0	1	1	1	1	1	1	1	1

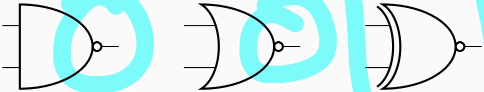
- AND, OR, NOT



- XOR



- NAND, NOR, XNOR



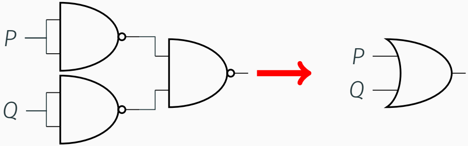
Universality of NAND and NOR

- NAND (AKA Sheffer stroke) $P \mid Q = \neg(P \land Q)$
- and NOR (AKA Pierce arrow) $P \downarrow Q = \neg(P \lor Q)$

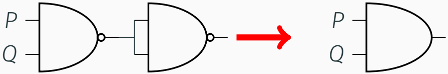
are universal:

$$\neg P \quad \equiv \quad P \mid P$$

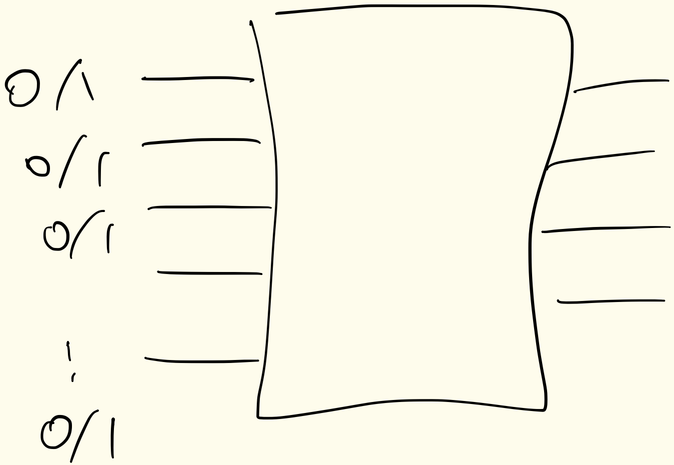
$$P \lor Q \quad \equiv \quad (P \mid P) \mid (Q \mid Q)$$



$$P \land Q \quad \equiv \quad (P \mid Q) \mid (P \mid Q)$$



Application: Number systems and circuits for addition


0/1
0/1
0/1
⋮
0/1

Binary number system

Positional system: multiply each digit by its place value

- Decimal notation:

$$4268_{10} = 4 \cdot 10^3 + 2 \cdot 10^2 + 6 \cdot 10^1 + 8 \cdot 10^0$$

- Binary notation

$$\begin{aligned} 1100\,0111_2 &= 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ &= 128 + 64 + 0 + 0 + 0 + 4 + 2 + 1 = 199_{10} \end{aligned}$$

Here indices 10 *and* 2 *are used to highlight the base of the number system*

$$01110_2 = \underset{0\cdot 16}{0\cdot 2^4} + \underbrace{1\cdot 2^3}_{8} + \underbrace{1\cdot 2^2}_{4} + \underbrace{1\cdot 2^1}_{2} + \underbrace{0\cdot 2^0}_{0} = 8+4+2 = 14_{10}$$

Convert decimal numbers to binaries: divide by 2

Rule: divide repeatedly by 2, writing down the reminder from each stage from right to left.

Example:

$533/2 = 266$	remainder = 1
$266/2 = 133$	remainder = 0
$133/2 = 66$	remainder = 1
$66/2 = 33$	remainder = 0
$33/2 = 16$	remainder = 1
$16/2 = 8$	remainder = 0
$8/2 = 4$	remainder = 0
$4/2 = 2$	remainder = 0
$2/2 = 1$	remainder = 0
$1/2 = 0$	remainder = 1

$$533_{10} = 1000010101_2$$

Alternative method

- If you know powers of 2, continually subtract largest power value from the number

$$
\begin{aligned}
123_{10} &= 64 + (123 - 64) = 64 + 59 \\
&= 64 + 32 + (59 - 32) = 64 + 32 + 27 \\
&= 64 + 32 + 16 + (27 - 16) = 64 + 32 + 16 + 11 \\
&= 64 + 32 + 16 + 8 + (11 - 8) = 64 + 32 + 16 + 8 + 3 = \\
&= 64 + 32 + 16 + 8 + 2 + (3 - 2) \\
&= 64 + 32 + 16 + 8 + 2 + 1 = \\
&= 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = \\
&= 1111011_2
\end{aligned}
$$

Binary addition

$$0_2 + 0_2 = 0_2 \qquad 0_2 + 1_2 = 1_2$$

$$1_2 + 0_2 = 1_2 \qquad 1_2 + 1_2 = 10_2$$

$$\begin{array}{rccccc}
 & & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & \\
 & & 1 & 1 & 1 & 1 \\
+ & & 1 & 0 & 1 & 1 \\
\hline
 & 1 & 1 & 0 & 1 & 0
\end{array}$$

$$\begin{array}{r} {}^{1}\quad\ \ {}^{1}\quad\ \ \\ 11001 \\ +\ 10101 \\ \hline 101110_2 \end{array}$$

$$\begin{array}{rl} 11001 & A \\ -\ 10111 & B \\ \hline 00010 & C \end{array}$$

$A - B = C$

$A = B + C$

$$\begin{array}{r} {}^{1}\quad\ \ \\ 10111 \\ +\ 00010 \\ \hline 11001 \end{array}$$

$a_7\, a_6\, a_5\, a_4\, a_3\, a_2\, a_1\, a_0$ $\qquad$ A

$b_7\, b_6\, b_5\, b_4\, b_3\, b_2\, b_1\, b_0$ $\qquad$ B

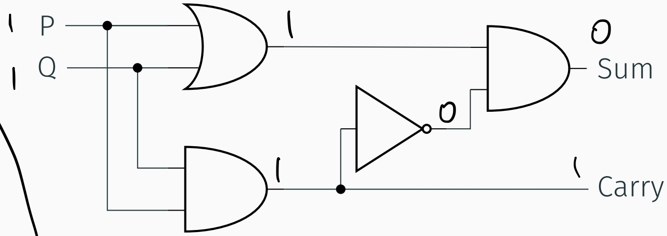
2^{16}

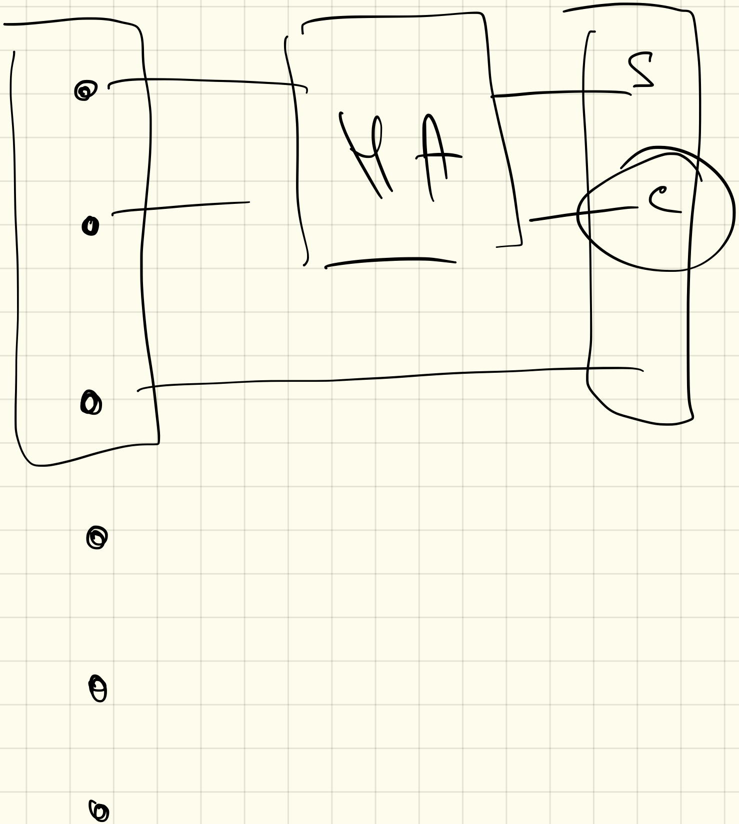
a_7	$- - -$ a_0	b_7 $- - -$ b_0	$c_8\, c_7$ $- - -$ c_0

Half-adder

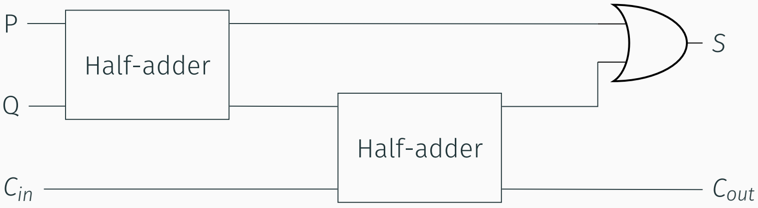
$$
\begin{array}{cccccc}
 & & 1 & 1 & 1 & \\
 & & 1 & 1 & 1 & 1 \\
+ & & 1 & 0 & 1 & 1 \\
\hline
 & 1 & 1 & 0 & 1 & 0
\end{array}
$$

P	Q	Carry	Sum
1	1	1	0
1	0	0	1
0	1	0	1
0	0	0	0



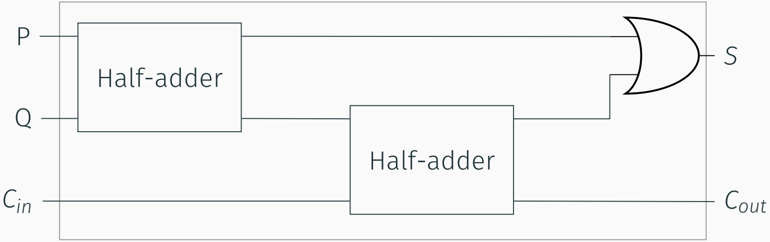

HA
S
C

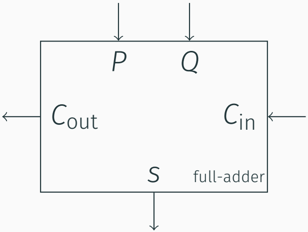
Full-adder



P	Q	C_{in}	C_{out}	S
1	1	1	1	1
1	1	0	1	0
1	0	1	1	0
1	0	0	0	1
0	1	1	1	0
0	1	0	0	1
0	0	1	0	1
0	0	0	0	0

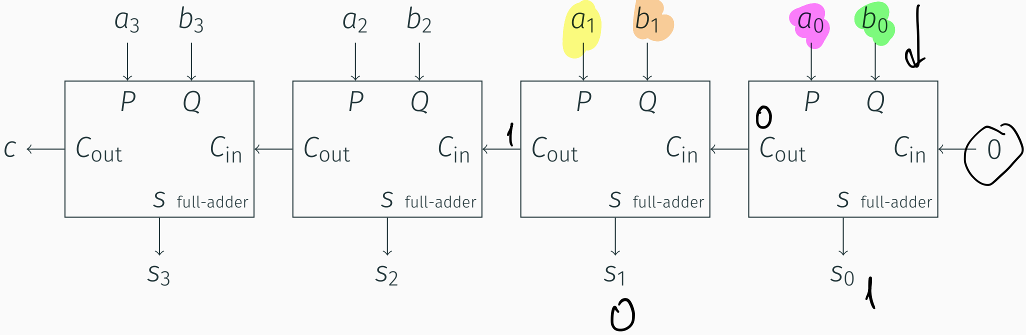
‘Black box’ notation

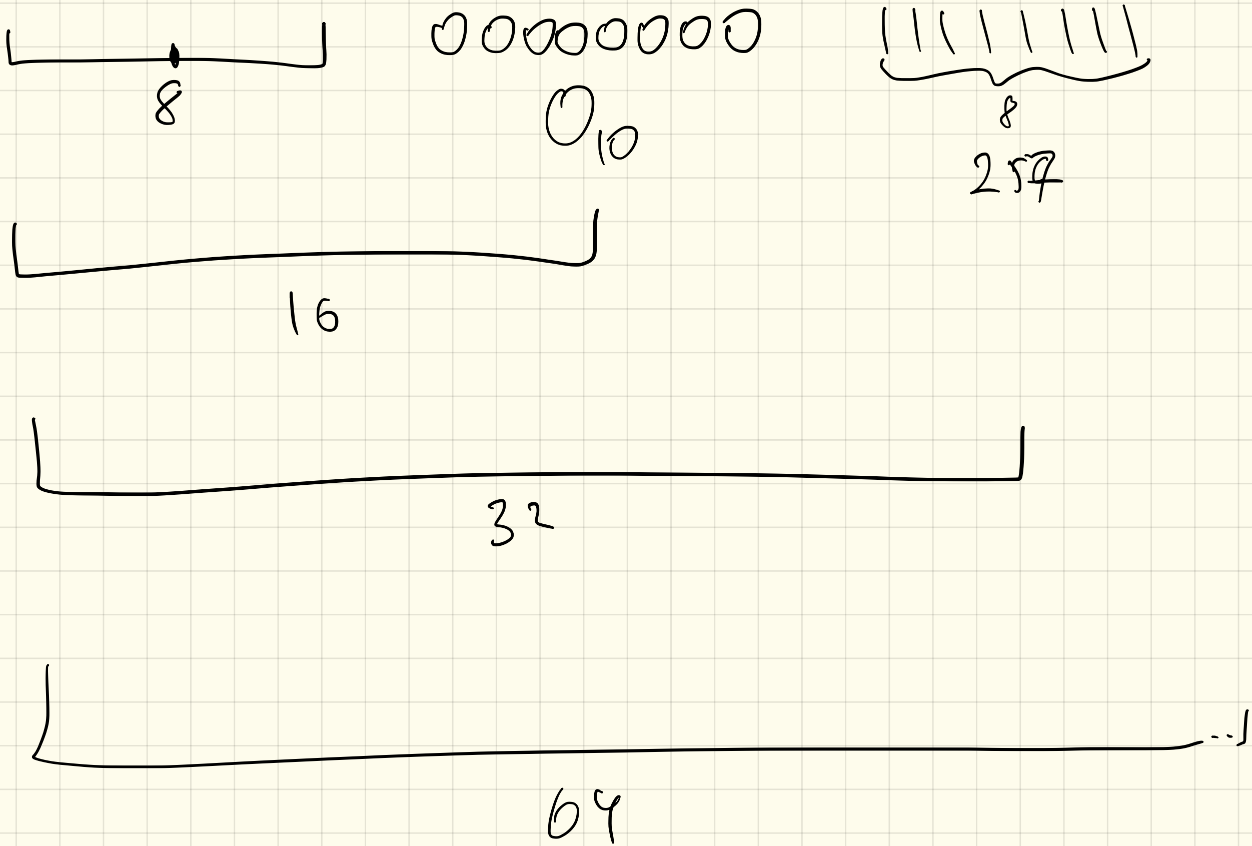




$$
\begin{array}{cccccc}
 & & a_3 & a_2 & a_1 & a_0 \\
+ & & b_3 & b_2 & b_1 & b_0 \\
\hline
 & c & s_3 & s_2 & s_1 & s_0
\end{array}
$$

$$
\begin{array}{cccccc}
 & & 1 & 1 & & \\
 & & 0 & 1 & 1 & 0 \\
+ & & 1 & 1 & 1 & 1 \\
\hline
 & 1 & 0 & 1 & 0 & 1
\end{array}
$$




8
00000000
0_{10}
8
257
16
32
64

$$\underbrace{0\;\overbrace{\quad - \quad - \quad - \quad}^{7}}_{8} \qquad +127$$

$$1\;\underbrace{\boxed{\qquad\qquad\qquad}}_{7} \qquad -127$$

$$0\;0000000 \qquad +0$$

$$1\;0000000 \qquad -0$$

Computer representation of negative integers

- Typically a fixed number of bits is used to represent integers: 8, 16, 32 or 64 bits
  - Unsigned integer can take all space available
- Signed integers
  - Leading sign

$$\begin{aligned} \mathbf{0}\,000\,0001_2 &= 1_{10} \\ \mathbf{1}\,000\,0001_2 &= -1_{10} \end{aligned}$$

    but then

$$\mathbf{1}\,000\,0000_2 = -0_{10} \text{ (?!)}$$

  - Two's complement:
    given a positive integer a, the **two's complement of a relative to a fixed bit length n** is the binary representation of

$$2^n - a.$$