Application: Number systems and circuits for addition

Binary number system

Positional system: multiply each digit by its place value

- Decimal notation:

3210

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

- Binary notation

7   10

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

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

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 $= 1$	lsb
$266/2 = 133$	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 $= 1$	msb

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

$57 = 2 \cdot 28 + 1$

$$\begin{array}{r|l} 28 & 1 \\ 14 & 0 \\ 7 & 0 \\ 3 & 1 \\ 1 & 1 \\ 0 & 1 \end{array}$$

$57_{10} = 111001_2$

Alternative method

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

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

Binary addition

$$
\boxed{\begin{array}{ll} 0_2 + 0_2 = 0_2 & 0_2 + 1_2 = 1_2 \\ 1_2 + 0_2 = 1_2 & 1_2 + 1_2 = 10_2 \end{array}}
$$

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

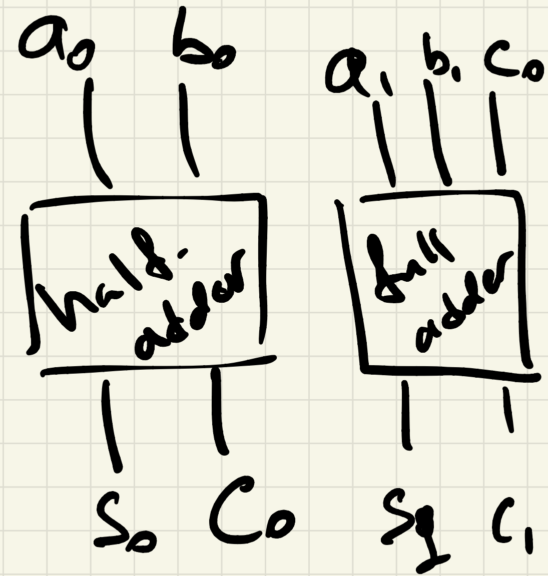
$$
\begin{array}{rrr}
 & {\scriptstyle 1} & \\
 & 2 & 7 \\
+ & 5 & 9 \\
\hline
 & 8 & 6
\end{array}
$$

```
 1
  1010
  1001
 -----
 10011
```

```
1 1 1 1
  1 1 1 1
  1 1 1 1
 ---------
 1 1 1 1 0
```

```
                  1
                 +1
            c1    0
       1  1  1  1
a         1  1  1  1
b         0  0  0  1
         -----------
s      1  0  0  0  0
                 s1
```

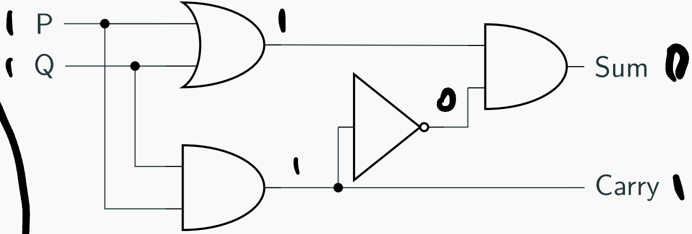
```
a    1 1 1 1

b    0 0 0 1
```



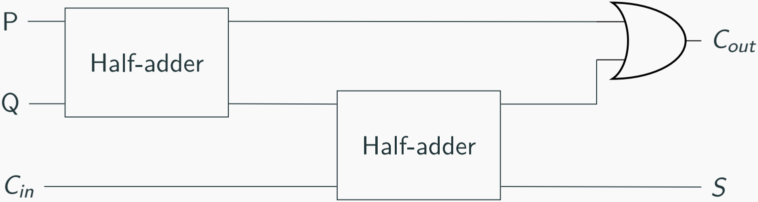
Half-adder

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

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

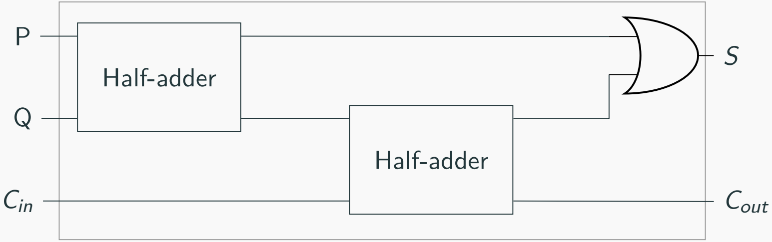


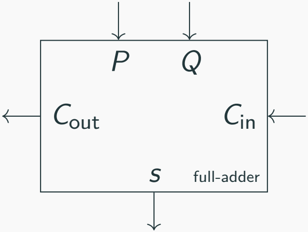
Full-adder



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

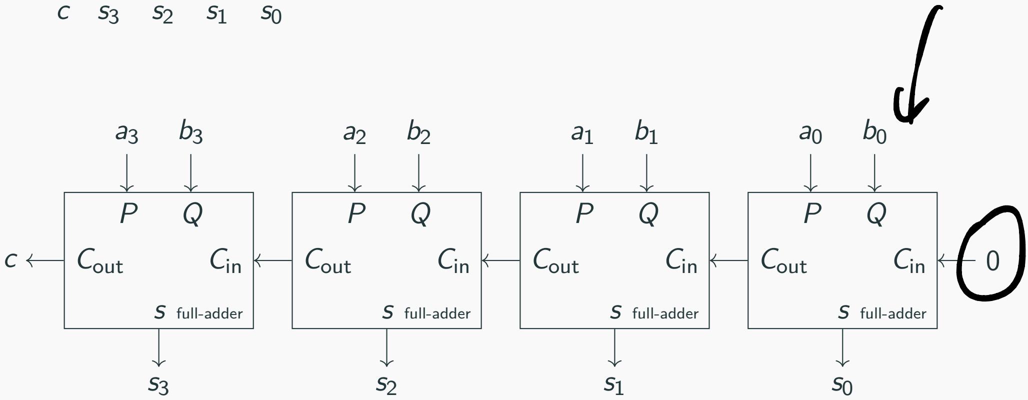
‘Black box’ notation

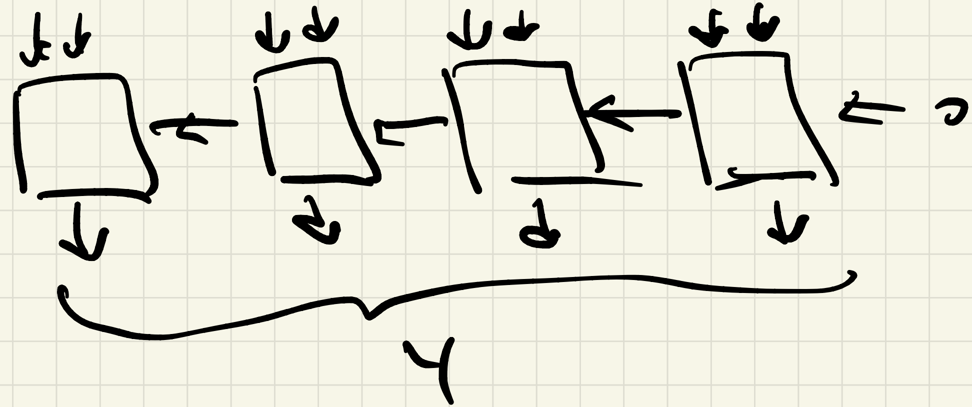




4-bit adder

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





$$\left.\boxed{\begin{array}{c}0000\\ \vdots\\ 1111\end{array}}\right\}\ \text{mod } 16$$

$$2^3 + 2^2 + 2^1 + 2^0 = 8 + 4 + 2 + 1 = 15$$

-17

$0111 = 7_{10}$

$1111 = -7_{10}$

$0000 = 0$

$\boxed{1000 = -0}$

mod 16

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

$$2^n - a.$$

n=4

Example: 4-bit two's complement (n=4)

- $a = 1$, two's complement: $2^4 - 1 = 15 = 1111_2$ $= -1$
- $a = 2$, two's complement: $2^4 - 2 = 14 = 1110_2$ $= -2$
- $a = 3$, two's complement: $2^4 - 3 = 13 = 1101_2$ $= -3$
- ...
- $a = 8$, two's complement: $2^4 - 8 = 8 = 1000_2$ $= -8$

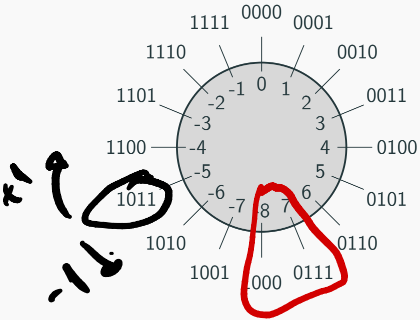
"normal" binary

interpretation

Properties

- Positive numbers start with 0, negative numbers start with 1
- 0 is always represented as a string of zeros
- -1 is always represented as a string of ones

Example: 4-bits



- The number range is split unevenly between +ve and -ve numbers
- The range of numbers we can represent in n bits is -2^{n-1} to $2^{n-1} - 1$