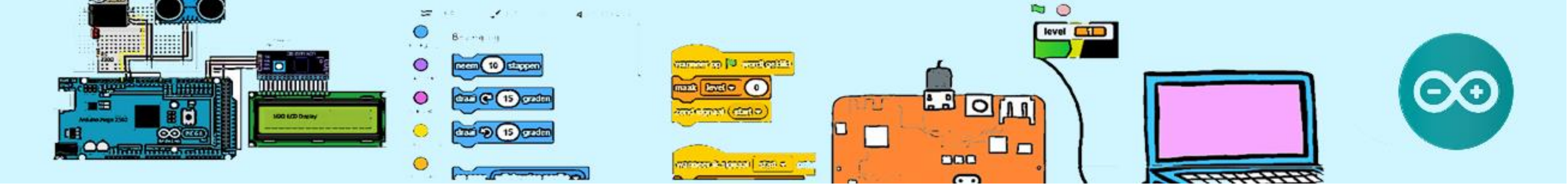


Voorkennis		Niveau	
Benodigdheden		Leerdoelen	

Computer binair rekenen

Onderstaande is van het Sondervick College te Veldhoven, zie: Informatica.haperen.com/klas4/h1/h1.htm .

In ons decimale of tientallige getallenstelsel gebruiken we maar 10 symbolen (cijfers) namelijk 0, 1, 2, 3, 4, 5, 6, 7, 8 en 9.

De waarde die de cijfers hebben hangt af van hun plaats, bijvoorbeeld:

$3847 = 7 + 40 + 800 + 3000 = 7 + 4 \times 10^1 + 8 \times 10^2 + 3 \times 10^3$

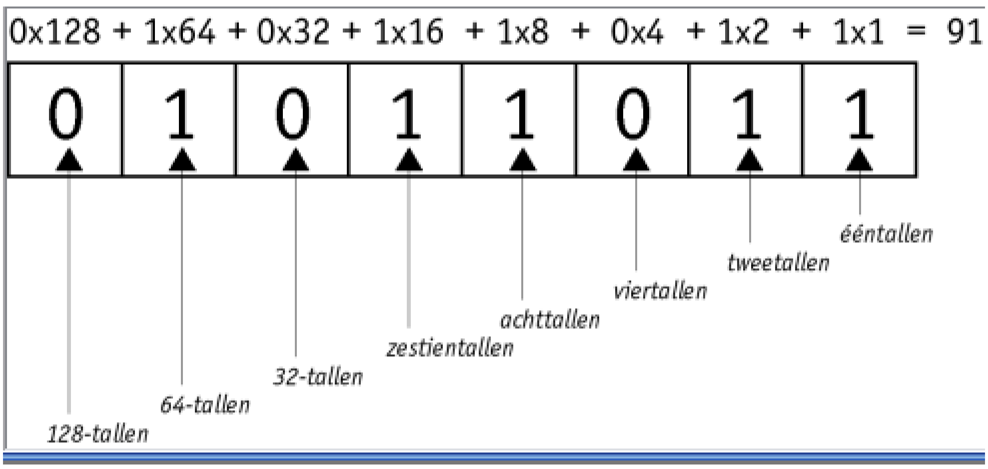
In de computerwereld wordt meestal gewerkt met het binaire of tweetallig getallenstelsel. Daarin worden maar twee tekens gebruikt, namelijk 0 en 1.

Omdat een bit maar twee waarden kan aannemen, namelijk 0 en 1, is het tweetallig getallenstelsel het voor de hand liggende systeem voor de computer.

In het tientallig stelsel moet je een getal van achter naar voren lezen. Het laatste cijfer blijft gelijk, het cijfer daarvoor moet keer 10, het cijfer daarvoor moet keer 100 (=10^2), het cijfer daarvoor moet keer 1000 (=10^3), enz.

In het binaire getallenstelsel moet je een getal ook van achter naar voren lezen. Het laatste cijfer blijft weer gelijk, het cijfer daarvoor moet keer 2, het cijfer daarvoor moet keer 4 (=2^2), het cijfer daarvoor moet keer 8 (=2^3), enz.

De decimale waarde van een binair getal kun je dus op de volgende manier bepalen:



decimaal	hexadecimaal	binair
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	1 0000
17	11	1 0001
18	12	1 0010
19	13	1 0011
20	14	1 0100

```
 1110
 0111 +
 ----
  ...1
```
0 + 1 = 01 (één)
1 opschrijven, 0 onthouden

```
    0
 1110
 0111 +
 ----
 ...01
```
0 + 1 + 1 = 10 (twee)
0 opschrijven, 1 onthouden

```
   1
 1110
 0111 +
 ----
 ..101
```
1 + 1 + 1 = 11 (drie)
1 opschrijven, 1 onthouden

```
  1
 1110
 0111 +
 ----
..0101
```
1 + 1 + 0 = 10 (twee)
0 opschrijven, 1 onthouden

```
 1
 1110
 0111 +
 ----
10101
```
1 opschrijven

```
       1  1  1  1   <---- carry
       1  1  1  0  1
(+)    1  1  0  1  1
    ------------------
    1  1  1  0  0  0
```

Optellen

```
          0  10   <---- borrow
       1  1  0  0
(-)    1  0  1  0
    --------------
       0  0  1  0
```

Aftrekken

decimale waarde --

```
                      2  9  10
0 3 0 0               0  3  0  0         0 3 0 0
0 2 0 9 -    →        0  2  0  9 -       0 3 0 9 -
                      0  0  9  1            -9
```

Maar een computer kent geen min teken, dus is er een “trucje” verzonnen. Bij een verschil berekening doen we alsof voor het Most Significant Bit (MSB) nog een extra 1 staat.

4 bit waarde ---

0 0 1 0 decimaal 2 we moeten bij het 1de digit gaan lenen

0 0 1 1 - decimaal 3

0 10 lenen | 0 1 1 10 lenen

1 0 0 ~~1~~ ~~0~~ | ~~0~~ ~~0~~ ~~0~~ 0

0 0 1 1 - | 0 0 1 1 min

1 | 1 1 1 1 = decimaal -1 2-complement methode

4 bit waarde ---

0 1 0 0 decimaal 4 we moeten bij het 1de digit gaan lenen

0 1 0 1 - decimaal 5

0 1 10 lenen | 1 10 lenen

1 0 ~~1~~ ~~0~~ ~~0~~ | 0 ~~0~~ ~~0~~ 1 0

0 1 0 1 - | 0 1 0 1 -

0 1 | 1 1 1 1 decimaal -1 2-complement method

In binair rekenen is het MSB van negatief getal 1, en omgekeerd als het MSB 1 is, dan is het een negatief getal!

Integer van twee byte ---

0 0 0 1 0 1 0 0 1 0 0 1 1 1 0 0 dit is decimaal 5.276

0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 - dit is decimaal 10.280

1 1 1 0 1 1 0 0 0 1 1 1 0 1 0 0 dit is decimaal -5.004

0 0 0 1 0 0 1 1 1 0 0 0 1 1 0 0 + dit is decimaal +5.004

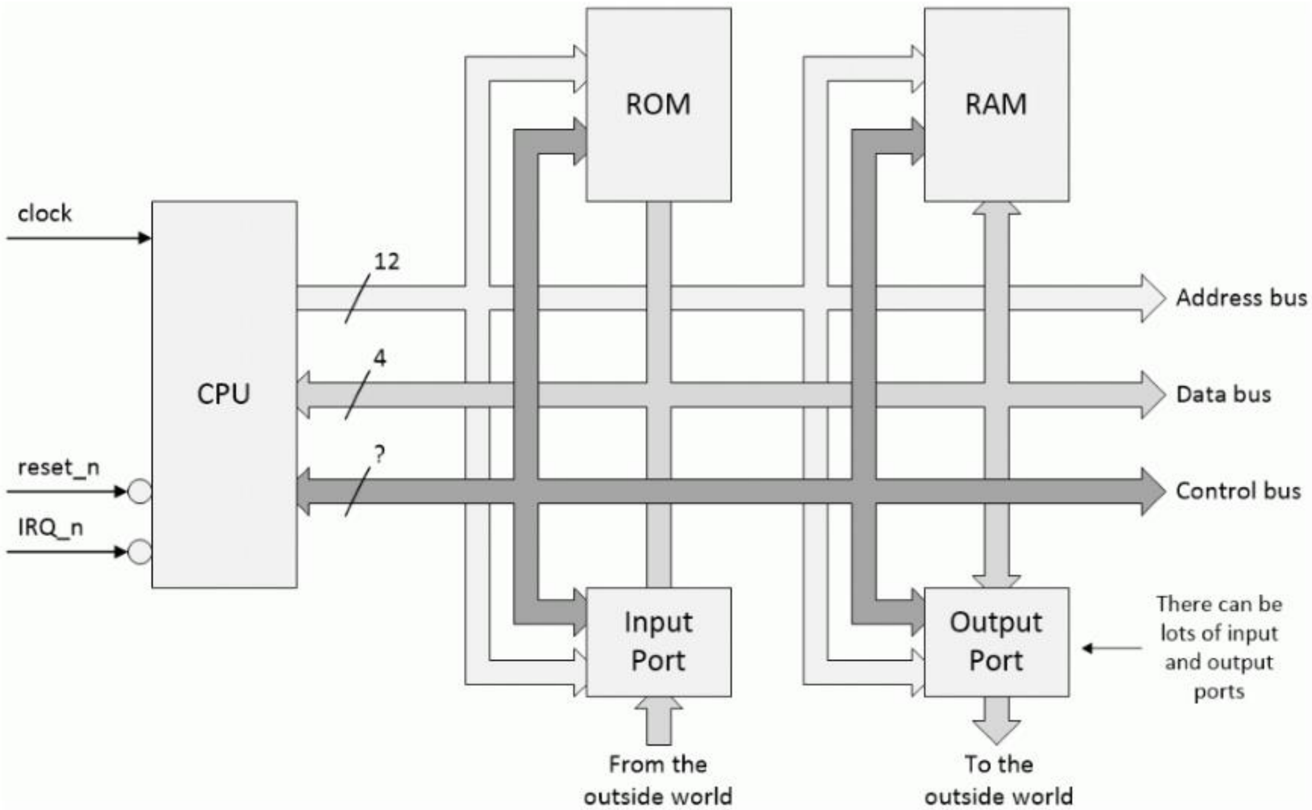
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

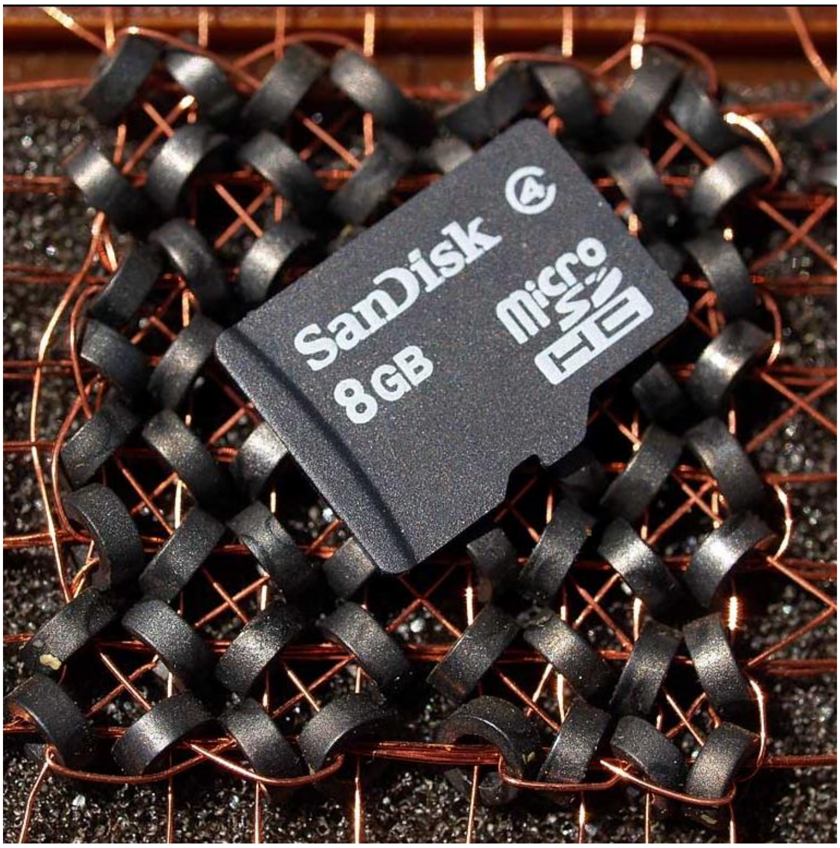
de 1 zou het 17de bit zijn, maar een 2 byte integer heeft maar 16 bit, dus die valt weg, dus het antwoord is 0.

0 0 0 1 1 1 0 0 1 0 0 0 dec 456

1 1 1 0 0 0 1 1 1 0 0 0 + dec -456

1 0 0 0 0 0 0 0 0 0 0 0 0


ROM
RAM
clock
CPU
12
4
?
Address bus
Data bus
Control bus
reset_n
IRQ_n
Input
Port
Output
Port
There can be
lots of input
and output
ports
From the
outside world
To the
outside world


SanDisk
8GB
micro
SD
HC
4