

- Talstelsels

- Decimale stelsel
- Binaire stelsel
- Hexadecimale stelsel
- Octale stelsel

- **Negatieve binaire getallen**

- Teken/grootte notatie
- Plus n-notatie
- Een- en tweecomplementnotatie
- Overflow

- Niet-gehele binaire getallen

- 'Floating point'-getallen
- IEEE-notatie

- BCD-getallen



Teken-en-grootte (sign-and-magnitude)

→ 7^o bit = tekenbit

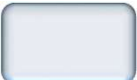
Niet getekend = 


Getekend = 

TEKENBIT

0 = Positief (+)

1 = Negatief (-)

 = Getalwaarde

 = Teken



• Voorbeeld

$+ 7_{(10)}$	=	0	0	0	0	0	1	1	1	$_{(2)}$
$- 7_{(10)}$	=	1	0	0	0	0	1	1	1	$_{(2)}$
$+ 21_{(10)}$	=	0	0	0	1	0	1	0	1	$_{(2)}$
$- 21_{(10)}$	=	1	0	0	1	0	1	0	1	$_{(2)}$

 = Getalwaarde
 = Teken

2 voorstellingen voor getal 0 overlappen

$$\Rightarrow -0 = 1000\ 0000_{(2)}$$

$$\Rightarrow +0 = 0000\ 0000_{(2)}$$

Binaire optelling dient anders te gebeuren



- Vaste lengte van het bitpatroon
- Waarde 0 = waarde in het midden van het spectrum van mogelijke waarden
- Waarde één hoger = 1 binair bij optellen
- Waarde één lager = 1 binair van aftrekken



→ 3 bits als vaste lengte

0	(referentiewaarde)	→	100 ₍₂₎
+1	(één optellen)	→	101 ₍₂₎
-1	(één aftellen)	→	011 ₍₂₎

1 1 1	→	+ 3
1 1 0	→	+ 2
1 0 1	→	+ 1
1 0 0	→	0
0 1 1	→	- 1
0 1 0	→	- 2
0 0 1	→	- 3
0 0 0	→	- 4

Plus 4 notatie

$$4_{(10)} \Leftrightarrow 100_{(2)}$$

$$100_{(2)} = \text{referentiewaarde} = 0_{(10)}$$

4 bits lengte => plus 8 OF plus 7-notatie

Plus 7

- 0111 = referentiewaarde 0

Plus 8

- 1000 = referentiewaarde 0

Plus 7	Binair	Plus 8
+8	1111	+7
+7	1110	+6
+6	1101	+5
+5	1100	+4
+4	1011	+3
+3	1010	+2
+2	1001	+1
+1	1000	0
0	0111	-1
-1	0110	-2
-2	0101	-3
-3	0100	-4
-4	0011	-5
-5	0010	-6
-6	0001	-7
-7	0000	-8



Decimaal → binair

1. Getal **optellen** met referentiewaarde (n)
2. Omzetten naar binaire waarde.

Voorbeeld (+ 4-notatie)

+2	→ $2 + 4 = 6$	→	binair : 1 1 0
-2	→ $-2 + 4 = 2$	→	binair : 0 1 0
-4	→ $-4 + 4 = 0$	→	binair : 0 0 0

Binair → decimaal

1. Omzetten naar decimaal
2. **Verminderen** met referentie waarde

Voorbeeld (+4-notatie)

$$1\ 1\ 0 = 6 \rightarrow 6 - 4 = +2$$

$$0\ 1\ 0 = 2 \rightarrow 2 - 4 = -2$$

$$0\ 0\ 0 = 0 \rightarrow 0 - 4 = -4$$

Voordeel:

1 aanduiding van nul ('n')

Nadeel:

Niet bruikbaar voor rekenkundige bewerkingen.

Voorbeeld:

+4 + (-3) [+8 Notatie]:

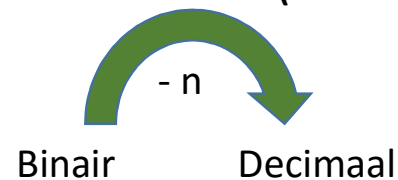
+4 : 1 1 0 0

-3 : 0 1 0 1

0 0 0 1 → (-7)

0001 → +8 notatie → 1 -8 = -7 !!

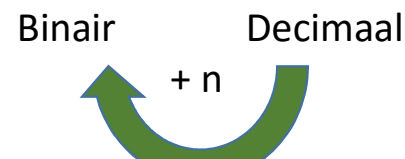
“n”_(binair) = Nieuwe referentie waarde voor 0_(decimaal)



Plus 4: 100 → 0

plus 8: 1000 → 0

plus 127: 0111 1111 → 0



“Het 1-complement bekomt men door het n-bits getal af te trekken van een n-bits getal met alle bits op 1.”

Voorbeeld:

$$\begin{array}{r}
 11111111 \\
 +14 = 00001110 \Rightarrow -14 = \frac{-00001110}{11110001}
 \end{array}$$

\Rightarrow in praktijk inversie van alle bits

Probleem

2 voorstellingen voor het getal 0

$\Rightarrow +0 = 0\ 000\ 0000$

$\Rightarrow -0 = 1\ 111\ 1111$



“ Negatief decimaal getal omzetten naar 2sC

Het twee-complement bekomt men door het n-bits getal af te trekken van een n+1-bits getal met de MSB op 1 en alle volgende bits op 0.”

- Voorbeeld:

$$\begin{array}{r}
 100000000 \\
 +14 = 00001110 \Rightarrow -14 = \frac{-00001110}{11110010}
 \end{array}$$

⇒ in praktijk inversie van alle bits + 1 optellen



- **Indien een positief getal (MSB = 0 !!)**
 - ⇒ Binaire waarde = absolute waarde
- **Indien een negatief getal (MSB = 1 !!)**
 - ⇒ Alle bits inverteren en plus 1. (=2'sc)
 - ⇒ Bekomen waarde is de absolute waarde!

!! Een negatief getal (*binair 2'complement*) begint met een 1



Voorbeeld: 1111 1011: → negatief want MSB = 1 !!!

1 1 1 1 1 0 1 1	→ (= 2-complement voorstelling)
0 0 0 0 0 1 0 0	→ (= complementaire waarde)
+ 1	
0 0 0 0 0 1 0 1	→ abs. waarde = 0101 = 5

De waarde van 1111 1011 is dus -5 !!

!!! Een negatief getal (binair 2'complent) begint met een 1 !!!

Positief getal

- Dec. waarde omzetten.
(Binair start met een nul!)

Negatief getal

- Absolute waarde omzetten (DEC → BIN).
- Alle bits inverteren en tel binair 1 op. (=2'sc)
Resultaat is altijd MSB = 1!!

!!! Een negatief getal (*binair 2'complement*) begint met een 1 !!!



Voorbeeld: -12

0 0 0 0 1 1 0 0	→ (=Binaire voorstellen 12)
1 1 1 1 0 0 1 1	→ (= complementaire waarde)
+ 1	
1 1 1 1 0 1 0 0	

-12 = 1111 0100

Plus 8-notatie	2s-C	
1 1 1 1	→ +7	0 1 1 1
1 1 1 0	→ +6	0 1 1 0
1 1 0 1	→ +5	0 1 0 1
1 1 0 0	→ +4	0 1 0 0
1 0 1 1	→ +3	0 0 1 1
1 0 1 0	→ +2	0 0 1 0
1 0 0 1	→ +1	0 0 0 1
1 0 0 0	→ 0	0 0 0 0
0 1 1 1	→ -1	1 1 1 1
0 1 1 0	→ -2	1 1 1 0
0 1 0 1	→ -3	1 1 0 1
0 1 0 0	→ -4	1 1 0 0
0 0 1 1	→ -5	1 0 1 1
0 0 1 0	→ -6	1 0 1 0
0 0 0 1	→ -7	1 0 0 1
0 0 0 0	→ -8	1 0 0 0

Bij de 2-complement methode geeft een bewerking de correcte uitkomst:

Notatie van -4: 0000 0100 (+4)

1111 1011 (1'sC)

1111 1100 (2'sC van −4)

+4: 0 0 0 0 0 1 0 0

-4: 1 1 1 1 1 1 0 0

≠ 0 0 0 0 0 0 0 0

De overdrachtsbit of carrybit speelt op dit moment geen rol en die laten we wegvallen

Beide getallen positief

$$\begin{array}{rcl}
 (+4) & \xrightarrow{\text{binair}} & 0000\ 0100 \\
 (+9) & \xrightarrow{\text{binair}} & 0000\ 1001 \\
 \hline
 (+13) & \xrightarrow{\text{binair}} & 0000\ 1101
 \end{array}$$

Een getal negatief, carry uit tekenbits

$$\begin{array}{rcl}
 (-4) & \xrightarrow{\text{binair}} & 1111\ 1100 \\
 (+9) & \xrightarrow{\text{binair}} & 0000\ 1001 \\
 \hline
 (+5) & \xrightarrow{\text{binair}} & 1\ 0000\ 0101
 \end{array}$$

Beide getallen negatief

$$\begin{array}{rcl}
 (-5) & \xrightarrow{\text{binair}} & 1111\ 1011 \\
 (-9) & \xrightarrow{\text{binair}} & 1111\ 0111 \\
 \hline
 (-14) & \xrightarrow{\text{binair}} & 1\ 1111\ 0010
 \end{array}$$

“end-around-carry” wordt verworpen
aangezien de computer werkt met een
vaste bitlengte (in dit geval 8)



1 voorstelling voor het getal 0

$\Rightarrow +0 = 0\ 000\ 0000$

$\Rightarrow -0 = 0\ 000\ 0000$

$$\begin{array}{r} \rightarrow \quad 1111\ 1111 \\ + \quad \quad \quad 1 \\ \hline 1\ 0000\ 0000 \end{array}$$

1 voorstelling voor het getal 0

- $+0 = 0\ 000\ 0000$
- $-0 = 0\ 000\ 0000$

Waarde-range van één byte

- Positief van $+0\ (00000000_{(2)})$ tot $+127\ (01111111_{(2)})$
- Negatief van $-1\ (11111111_{(2)})$ tot $-128\ (10000000_{(2)})$

Binaire optelregels blijven geldig

$$\begin{array}{lcl} (+4) & \xrightarrow{\text{binair}} & 0\ 000\ 0100 \\ (-4) & \xrightarrow{\text{binair}} & 1111\ 1100 \\ \hline (-0) & \xrightarrow{\text{binair}} & 1\ 0000\ 0000 \end{array}$$

Meest gebruikte voorstelling



+ → MSB = 0 ⇒ positief getal

⇒ waarde omzetten

- → MSB = 1 ⇒ negatief getal

⇒ 2's complement ⇒ = |abs waarde|

• Teken-grootte notatie:

+60 = ... (2)
 -33 = ... (2)
 -129 = ... (2)
 -598 = ... (2)

0011 1101 = ... (10)
 1010 0011 = ... (10)
 1001 1111 = ... (10)
 0110 1101 = ... (10)

• Plus n notatie:

-2 (plus4) = ... (2)
 +2 (plus4) = ... (2)
 -7 (plus8) = ... (2)
 +1 (plus8) = ... (2)
 +10 (plus127) = ... (2)

001 (plus 4) = ... (10)
 110 (plus 4) = ... (10)
 1010 (plus 8) = ... (10)
 0011 (plus 8) = ... (10)
 0110 1010 (+127) = ... (10)

• 2'sC notatie:

+60 = ... (2)
 - 33 = ... (2)
 - 98 = ... (2)
 + 192 = ... (2)
 - 129 = ... (2)

0011 1101 = ... (10)
 1010 0011 = ... (10)
 1001 1111 = ... (10)
 0110 1101 = ... (10)
 1010 0000 = ... (10)



Teken-grootte notatie:

+60 =	0011 1100 ₍₂₎	0011 1101 = + 61 ₍₁₀₎
- 33 =	1010 0001 ₍₂₎	1010 0011 = - 35 ₍₁₀₎
-129 =	1000 0000 1000 0001 ₍₂₎	1001 1111 = - 31 ₍₁₀₎
-598 =	1000 0010 0101 0110 ₍₂₎	0110 1101 = + 109 ₍₁₀₎

ALGEMEEN (teken en grootte):

MSB = TEKEN

0 = +

1 = -

Plus n notatie:

- 2 (plus4)	=	010 ₍₂₎	001 (plus 4)=	-3 ₍₁₀₎
+ 2 (plus4)	=	110 ₍₂₎	110 (plus 4)=	+2 ₍₁₀₎
- 7 (plus8)	=	0001 ₍₂₎	1010 (plus 8)=	+2 ₍₁₀₎
+ 1 (plus8)	=	1001 ₍₂₎	0011 (plus 8)=	-5 ₍₁₀₎
+ 10 (plus127)	=	1000 1001 ₍₂₎	0110 1010 (+127)=	-21 ₍₁₀₎

Opgelet: vaste bitlengte !!

ALGEMEEN (plus n notatie):

"n" (Binair) \Leftrightarrow 0 (Decimaal)

vb: plus 8: 1000 \Leftrightarrow 0

vb: plus 127: 0111 1111 \Leftrightarrow 0

➔ van BIN naar DEC ➔ - 'n'

➔ van DEC naar BIN ➔ + 'n'





Teken-grootte notatie:

$+60 = 0011\ 1100_{(2)}$
 $-33 = 1010\ 0001_{(2)}$
 $-129 = 1000\ 0000\ 1000\ 0001_{(2)}$
 $-598 = 1000\ 0010\ 0101\ 0110_{(2)}$

$0011\ 1101 = +61_{(10)}$
 $1010\ 0011 = -35_{(10)}$
 $1001\ 1111 = -31_{(10)}$
 $0110\ 1101 = +109_{(10)}$

Plus-n notatie:

$-2\ (\text{plus}4) = 010_{(2)}$
 $+2\ (\text{plus}4) = 110_{(2)}$
 $-7\ (\text{plus}8) = 0001_{(2)}$
 $+1\ (\text{plus}8) = 1001_{(2)}$
 $+10\ (\text{plus}127) = 1000\ 1001_{(2)}$

$001\ (\text{plus } 4) = -3_{(10)}$
 $110\ (\text{plus } 4) = +2_{(10)}$
 $1010\ (\text{plus } 8) = +2_{(10)}$
 $0011\ (\text{plus } 8) = -5_{(10)}$
 $0110\ 1010\ (+127) = -21_{(10)}$

2'sC notatie:

$+60 = 0011\ 1100_{(2)}$
 $-33 = 1101\ 1111_{(2)}$
 $-98 = 1001\ 1110_{(2)}$
 $+192 = 0000\ 0000\ 1100\ 0000_{(2)}$
 $-129 = 1111\ 1111\ 0111\ 1111_{(2)}$

$0011\ 1101 = +61_{(10)}$
 $1010\ 0011 = -93_{(10)}$
 $1001\ 1111 = -97_{(10)}$
 $0110\ 1101 = +109_{(10)}$
 $1010\ 0000 = -96_{(10)}$

Er is onvoldoende geheugenruimte om resultaat voor te stellen =
OVERFLOW

2 manieren om overflow te detecteren:

+ & + = -
- & - = +

Cary over NAAR de tekenbit
=/
Cary over VAN de tekenbit

$$\begin{array}{rcl}
 (-3) & \xrightarrow{\text{binair}} & 1101 \\
 (+2) & \xrightarrow{\text{binair}} & 0010 \\
 \hline
 (-1) & \xrightarrow{\text{binair}} & 1111
 \end{array}$$

GEEN OVERFLOW:

- & + → onmogelijk overflow

Carry van tekenbit = carry naar tekenbit

↓

1111 = Negatief getal (MSB =1)

2cs = 0001 (invert +1)

➔ |1| (absolute waarde)

$$\begin{array}{lcl} (+3) & \xrightarrow{\text{binair}} & 0011 \\ (-5) & \xrightarrow{\text{binair}} & 1011 \\ \hline (-2) & \xrightarrow{\text{binair}} & 1110 \end{array}$$

GEEN OVERFLOW:

+ & - \rightarrow onmogelijk overflow

Carry van tekenbit = carry naar tekenbit



1110 = Negatief getal ($MSB = 1$)
2cs = 0010 ($invert + 1$)
 $\rightarrow |2|$ ($absolute\ waarde$)

$$\begin{array}{rcl}
 (+3) & \xrightarrow{\text{binair}} & 0011 \\
 (+5) & \xrightarrow{\text{binair}} & 0101 \\
 \hline
 (-8) & \xrightarrow{\text{binair}} & 1000
 \end{array}$$

OVERFLOW !!:

+ & + → Uitkomst is – !!!

Carry van tekenbit ≠ carry naar tekenbit !!!

↓
 1000 = Negatief getal
 2cs = 0111 + 1 = 1000
 → |8|

(MSB =1)
 (invert +1)
 (absolute waarde)

Overflow !! (4bit) → oplossing: uitbreiden naar 8 bit !!



$$\begin{array}{lcl} (+3) & \xrightarrow{\text{binair}} & 0000\ 0011 \\ (+5) & \xrightarrow{\text{binair}} & 0000\ 0101 \\ \hline (+8) & \xrightarrow{\text{binair}} & 0000\ 1000 \end{array}$$

GEEN OVERFLOW !!:

+ & + → Uitkomst is +

Carry van tekenbit = carry naar tekenbit !!!

(Overflow 4bit) → oplossing: uitbreiden naar 8 bit !!



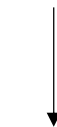
$$\begin{array}{rcl}
 (-4) & \xrightarrow{\text{binair}} & 1100 \\
 (-7) & \xrightarrow{\text{binair}} & 1001 \\
 \hline
 (???) & \xrightarrow{\text{binair}} & \cancel{1} 0101
 \end{array}$$

OVERFLOW !!:

- & - → Uitkomst is + !!!

Carry van tekenbit ≠ carry naar tekenbit !!!

Overflow !! (4bit)



0101 = Positief getal
→ |5|

(MSB = 0)
(absolute waarde)

Overflow !! (4bit) → oplossing: uitbreiden naar 8 bit !!



Uitbreiding naar 8 bit !!

$$\begin{array}{rcl} (-4) & \xrightarrow{\text{binair}} & 1111\ 1100 \\ (-7) & \xrightarrow{\text{binair}} & 1111\ 1001 \\ \hline (-11) & \xrightarrow{\text{binair}} & 1111\ 0101 \end{array}$$

↓
1111 0101 = Negatief getal (MSB = 1)
→ |5| (absolute waarde)

1: optelling van 2 positieve getallen

$$\begin{array}{rcl}
 (+65) & \xrightarrow{\text{binair}} & 0100\ 0001 \\
 (+66) & \xrightarrow{\text{binair}} & 0100\ 0010 \\
 \hline
 (????) & \xrightarrow{\text{binair}} & 1000\ 0011
 \end{array}$$

2-complement = -61

2: optelling van 2 negatieve getallen

$$\begin{array}{rcl}
 (-29) & \xrightarrow{\text{binair}} & 1110\ 0011 \\
 (-100) & \xrightarrow{\text{binair}} & 1001\ 1011 \\
 \hline
 (????) & \xrightarrow{\text{binair}} & 1\ 0111\ 1110
 \end{array}$$

2-complement = +127

“ De fout wordt in beide gevallen opgemerkt door het plots
verschijnen van een ander teken.”

“Sommatie van twee positieve getallen resulteert in een negatief getal”

“Sommatie van twee negatieve getallen resulteert in een positief getal”

*Vermits we hier werken met vaste bitpatronen van 8 bits zitten we zoals in de
voorbeelden werd aangegeven, al vlug buiten het bereik.*



- **Data types**

bits	naam	Java	.Net	bereik
1 bit	bit	boolean	boolean	true false
8	byte, octet	byte	byte	-128 tot 127
16	word	char (unicode)	char	0 tot 65535
16	word	short	short	-32768 tot 32767
32	doubleword	int	integer	-2.147.483.648 tot 2.147.483.647
64	quadword	long	long	-9.223.372.036.854.775.808 tot +9.223.372.036.854.775.807
128	octaword			
32		float	single	$1,401298.e^{-45}$ tot $3,40282.e^{+38}$
64		double	double	$4,940656.e^{-324}$ tot $1,797693.e^{308}$

Maak de optelling en controleer op overflow.

$$+14 + (-30)$$

$$+58 + (-72)$$

$$-68 + (-50)$$

$$+87 + 82$$

$$-87 + (-82)$$

$$+76 + 118$$

$$70_{(8)} + 32_{(8)}$$

$$73_{(h)} + 7A_{(h)}$$