GETEKENDE GETALLEN

Voorstelling van positieve en negatieve getallen

1. Vul de onderstaande tabel verder aan.

teken-grootte	decimaal	binair (unsigned)	1-complement	2-complement
0 001 0011	+19	0 001 0011	0 001 0011	0 001 0011
1 001 1011	-27	0 001 1011	1 110 0100	1 110 0101
0 000 0000	+0	0 000 0000	0 000 0000	0 000 0000
1 000 0000	-0	0 000 0000	1 111 1111	0 000 0000
1 111 0111	-119	0 111 0111	1 000 1000	1 000 1001
0 101 0011	+83	0 101 0011	0 101 0011	0 101 0011
1 110 1100	-108	0 110 1100	1 001 0011	1 001 0100
1 101 1011	-91	0 101 1011	1 010 0100	1 010 0101
1 111 1111	-127	0 111 1111	1 000 0000	1 000 0001
1 000 1000 0001	-129	0 000 1000 0001	1 111 0111 1110	1 111 0111 1111
1 001 0000 1101	-269	0 001 0000 1101	1 110 1111 0010	1 110 1111 0011
1 010 0000 0010	-514	0 010 0000 0010	1 101 1111 1101	1 101 1111 1110
1 011 1110 1001	-1001	0 011 1110 1001	1 100 0001 0110	1 100 0001 0111

- Begin met de ongetekende binaire weergave (absolute waarde).
- Zet voor de teken-en-grootte notatie het juiste teken in de MSB (most significant bit)
- Voor de 1-complement notatie: converteer elke bit vertrekkende van de ongetekende binaire weergave.
- Voor de 2-complement notatie: tel binair 1 bij de 1-complement notatie.

2. Geef de decimale waarde van de volgende binaire getallen (in plus 8-notaties)
   a) 1 0 0 1 = +1
   b) 0 1 0 0 = -4
   c) 0 0 0 0 = -8

3. Controleer de bekomen resultaten van vraag 2 door de omgekeerde berekening uit te voeren (van decimaal naar plus 8-notatie).
   a) +1 +8 = 9 → 1 0 0 1
   b) -4 +8 = 4 → 0 1 0 0
   c) -8 +8 = 0 → 0 0 0 0

4. Geef aan hoe groot het grootste positieve getal is in 2-complement notatie, als je werkt met patronen van 32 bits.

   Grootste positief 32-bit getal = 0 111 1111 1111 1111 1111 1111 1111 1111 = 2147483647

5. Bereken de decimale waarde van de volgende getallen in 2-complement notatie.
   a) 0 1 0 1 1 0 1 0 = 90
   b) 1 0 0 1 1 0 0 1 → is een negatief getal (1sC nemen en 1 bijtellen) = 0110 0111 = - 103
   c) 1 0 0 0 0 0 0 1→ is een negatief getal (1sC nemen en 1 bijtellen) = 01111111 = - 127

$$\begin{array}{r} (+29) \\ +(+13) \\ \hline +42 \end{array} \xrightarrow{binair} \begin{array}{r} 0\ 001\ 1101 \\ 0\ 000\ 1101 \\ \hline 0\ 010\ 1010 \end{array}$$

$$\begin{array}{r} (+40) \\ +(-13) \\ \hline +27 \end{array} \xrightarrow{binair} \begin{array}{r} 0\ 010\ 1000 \\ 1\ 111\ 0011 \\ \hline \cancel{1}0\ 001\ 1011 \end{array}$$

$$\begin{array}{r} (-50) \\ +(+4) \\ \hline -46 \end{array} \xrightarrow{binair} \begin{array}{r} 1\ 100\ 1110 \\ 0\ 000\ 0100 \\ \hline 1\ 101\ 0010 \end{array} \xrightarrow{negatief} -0010\ 1110$$

$$\begin{array}{r} (-57) \\ +(-17) \\ \hline -74 \end{array} \xrightarrow{binair} \begin{array}{r} 1\ 100\ 0111 \\ 1\ 110\ 1111 \\ \hline \cancel{1}1\ 011\ 0110 \end{array} \xrightarrow{negatief} -0100\ 1010$$

$$\begin{array}{r} (-56) \\ -(+27) \\ \hline -83 \end{array} \xrightarrow{binair} \begin{array}{r} 1\ 100\ 1000 \\ 1\ 110\ 0101 \\ \hline \cancel{1}\ 1010\ 1101 \end{array} \xrightarrow{negatief} -0101\ 0011$$

$$\begin{array}{r} (-65) \\ +(-64) \\ \hline -129 \end{array} \xrightarrow{binair} \begin{array}{r} 1\ 011\ 1111 \\ 1\ 100\ 0000 \\ \hline \cancel{1}\ 0\ 111\ 1111 \end{array} \xrightarrow{positief} \text{OVERFLOW !!!!!}$$

$$\begin{array}{r} (-65) \\ +(-64) \\ \hline -129 \end{array} \xrightarrow{binair} \begin{array}{r} 1\ 111\ 1111\ 1011\ 1111 \\ 1\ 111\ 1111\ 1100\ 0000 \\ \hline 1\ 111\ 1111\ 0111\ 1111 \end{array} \xrightarrow{negatief} -0000\ 1000\ 00$$

$$\begin{array}{r} (-165) \\ +(+64) \\ \hline -101 \end{array} \xrightarrow{binair} \begin{array}{r} 1\ 111\ 1111\ 0101\ 1011 \\ 0\ 000\ 0000\ 0100\ 0000 \\ \hline 1\ 111\ 1111\ 1001\ 1011 \end{array}$$

Optelling met decimale, octale en hexadecimale getallen

$$\begin{array}{r} (-40) \\ +(-28) \\ \hline -68 \end{array} \xrightarrow{binair} \begin{array}{r} 1\ 101\ 1000 \\ 1\ 110\ 0100 \\ \hline 1\ 011\ 1100 \end{array}$$

$$\begin{array}{r} (-268) \\ +(+200) \\ \hline -68 \end{array} \xrightarrow{binair} \begin{array}{r} 1\ 111\ 1110\ 1111\ 0100 \\ 0\ 000\ 0000\ 1100\ 1000 \\ \hline 1\ 111\ 1111\ 1011\ 1100 \end{array}$$

Beide bovenstaande opgaven hebben -68 als resultaat, maar aangezien de operandi in de tweede opgave een 16-bit notatie vereisen, staat de uitkomst hier dus ook in 16 bit. Beide binaire uitkomsten stellen dus -68 voor, resp. in 8-bit en 16-bit notatie.

$$\begin{array}{r} 13_{(8)} \\ +27_{(8)} \\ \hline 42_{(8)} \end{array} \xrightarrow{binair} \begin{array}{r} 00\ 001\ 011 \\ 00\ 010\ 111 \\ \hline 00\ 100\ 010 \end{array}$$

$$\begin{array}{r} 33_{(8)} \\ +20_{(8)} \\ \hline 53_{(8)} \end{array} \xrightarrow{binair} \begin{array}{r} 00\ 011\ 011 \\ 00\ 010\ 000 \\ \hline 00\ 101\ 011 \end{array}$$

$$\begin{array}{r} (+7D_h) \\ +(+29_h) \\ \hline A6_h \end{array} \xrightarrow{binair} \begin{array}{r} 0\ 111\ 1101 \\ 0\ 010\ 1001 \\ \hline 1\ 010\ 0110 \end{array} \xrightarrow{Overflow} \begin{array}{r} 0\ 000\ 0000\ 0111\ 1101 \\ 0\ 000\ 0000\ 0010\ 1001 \\ \hline 0\ 000\ 0000\ 1010\ 0110 \end{array}$$

$$\begin{array}{r} (+AC_h) \\ +(-D4_h) \\ \hline D8 \end{array} \xrightarrow{binair} \begin{array}{r} 0\,000\ 0000\ 1010\ 1100 \\ -0\,000\ 0000\ 1101\ 0100 \\ \hline \end{array} \xrightarrow{\text{Omvorming naar optelling}} \begin{array}{r} 0\,000\ 0000\ 1010\ 1100 \\ 1\ 111\ 1111\ 0010\ 1100 \\ \hline 1\ 111\ 1111\ 1101\ 1000 \end{array}$$

Omdat $D4_h$ reeds binair met een “1” begint, wordt dit beschouwd als een negatief getal. Er moet dus gestart worden met $00D4_h$ en hiervan dient het 2-complement genomen te worden.

Overflow

Technisch gezien treedt overflow op als de carry naar de tekenbit verschilt van de carry vanuit de tekenbit (**end-around-carry**).

Carry naar tekenbit	Carry vanuit tekenbit	Overflow
Neen	Neen	Neen
Ja	Neen	Ja
Neen	Ja	Ja
Ja	Ja	Neen

Geef voor de volgende bewerkingen aan of er een overdracht is naar de tekenbit, vanuit de tekenbit en bepaal hiermee of er overflow is of niet.

1. 002B + 04C1

```
carry
002B  =      0 000 0000 0010 1011
04C1  =      0 000 0100 1100 0001
04EC         0 000 0100 1110 1100
```

Geen Carry_{in} en geen Carry_{out} dus geen overflow

2. 1C39 + 49C3

```
carry
1C39  =      0 001 1100 0011 1001
49C3  =      0 100 1001 1100 0011
65FC         0 110 0101 1111 1100
```

Geen Carry_{in} en geen Carry_{out} dus geen overflow

3. 7A12 + 4A59

```
Carry        1
7A12  =      0 111 1010 0001 0010
4A59  =      0 100 1010 0101 1001
C46B =       1 100 0100 0110 1011
```

Carry_{in} , maar geen Carry_{out} dus overflow

```
Carry
00007A12  =    0 000 0000 0000 0000 0111 1010 0001 0010
00004A59  =    0 000 0000 0000 0000 0100 1010 0101 1001
0000C46B   =   0 000 0000 0000 0000 1100 0100 0110 1011
```

Geen Carry_{in} en geen Carry_{out} dus niet langer een overflow

4. 8FFE + 0002

```
Carry
8FFE  =      1 000 1111 1111 1110
0002  =      0 000 0000 0000 0010
9000  =      1 001 0000 0000 0000
```

Geen Carry_{in} en geen Carry_{out} dus geen overflow

5. FF1A + 084D

```
Carry    1 1
 FF1A =    1 111 1111 0001 1010
 084D =    0 000 1000 0100 1101
10767 =  1 0 000 0111 0110 0111
```

Carry_{in} en Carry_{out} dus geen overflow

6. 89CC + D944

```
Carry    1
 89CC =    1 000 1001 1100 1100
 D944 =    1 101 1001 0100 0100
16310 =  1 0 110 0011 0001 0000
```

Geen Carry_{in} , maar wel Carry_{out} dus overflow

```
Carry        1 1
 FFFF89CC =    1 111 1111 1111 1111 1000 1001 1100 1100
 FFFFD944 =    1 111 1111 1111 1111 1101 1001 0100 0100
 FFFF6310 =  1 1 111 1111 1111 1111 0110 0011 0001 0000
```

Carry_{in} en Carry_{out} dus niet langer een overflow.