Internal representations

• Addition and subtraction in complementary code

$x = 53$

$y = 80$

	S							
$[53]_{compl}$	0	0	1	1	0	1	0	1
$[-53]_{compl}$	1	1	0	0	1	0	1	1
$[80]_{compl}$	0	1	0	1	0	0	0	0
$[-80]_{compl}$	1	0	1	1	0	0	0	0

$$[x+y]_{compl} = [x]_{compl} \oplus [y]_{compl}$$

$$[x-y]_{compl} = [x]_{compl} \oplus [-y]_{compl}$$

$$[53+53]_{compl} = [53]_{compl} \oplus [53]_{compl}$$

$[53]_{compl}$	0	0	1	1	0	1	0	1	$\oplus$
$[53]_{compl}$	0	0	1	1	0	1	0	1	
$[53+53]_{compl}$	0	1	1	0	1	0	1	0	

no overflow ✓ ⇒ correct result

[106]

$[53]_{compl}$ $0|0110101$ $\oplus$

$[80]_{compl}$ $0|1010000$

$[53+80]_{compl}$ $1|0000101$ overflow X ⇒ incorrect answer

It is an overflow because the operands are positive numbers and the result is negative.

$[53]_{compl}$ $0|0110101$ $\oplus$

$[-80]_{compl}$ $1|0110000$

$[50-80]_{compl}$ $1|1100101$ correct

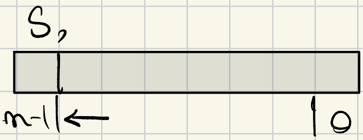
-27

complement ⇒ absolute value

$[27]_{compl}$: $0|0011011$

$11011 = 2^4+2^3+2^1+2^0 = 16+8+2+1 = 27$

Codes for subunitary numbers $|x| < 1$



n bits ; $n = 8$ bits

$x = 11/16 = 11 \cdot 16^{-1} = 0{,}B_{(16)} = 0{,}1011_{(2)}$

$y = 0{,}45 = 0{,}\underbrace{7}_{4\text{ bits}}\underbrace{3}_{3\text{ bits}}{}_{(16)} = 0{,}\underbrace{0111001}_{\text{we take only 7 bits}}{}_{(2)}$

$0{,}45 \cdot 16 = 7{,}20$

$0{,}20 \cdot 16 = 3{,}20$

$$\begin{array}{r} 45 \cdot \\ 16 \\ \hline 270 \\ 45 \\ \hline 720 \end{array} \rightarrow 7 \qquad \begin{array}{r} 20 \cdot \\ 16 \\ \hline 120 \\ 20 \\ \hline 320 \end{array}$$

$[11/16]_{dir} = [11/16]_{inv} = [11/16]_{compl} =$ 0 | 1 0 1 1 0 0 0 (S,)

$[-11/16]_{dir} =$ 1 | 1 0 1 1 0 0 0 (S,)

$[-11/16]_{inv} =$ 1 | 0 1 0 0 1 1 1 (S,)

$[-11/16]_{compl} =$ 1 | 0 1 0 1 0 0 0 (S,)

$$[0,45 + 0,45]_{compl} = [0,45]_{compl} \oplus [0,45]_{compl}$$

		S,								
$[0,45]_{compl}$	:	0	0	1	1	1	0	0	1	$\oplus$
$[0,45]_{compl}$	:	0	0	1	1	1	0	0	1	
$[0,89]_{compl} \simeq$		0	1	1	1	0	0	1	0	

no overflow, correct result

$$0,\overset{-1}{1}\overset{-2}{1}\overset{-3}{1}00\overset{-6}{1}0_{(2)} = 2^{-1} + 2^{-2} + 2^{-3} + 2^{-6}$$

$$= {}^{32/}\frac{1}{2} + {}^{16/}\frac{1}{4} + {}^{8/}\frac{1}{8} + \frac{1}{64} = \frac{32+16+8+1}{64} = \frac{57}{64} = 0,89$$

we do not get the exact result
bcs. we did not represent 0,45 on all it's bits!

$$[11/16 + 0,45]_{compl} = [11/16]_{compl} + [0,45]_{compl}$$

		S,								
$[11/16]_{compl}$	:	0	1	0	1	1	0	0	0	$\oplus$
$[0,45]_{compl}$	:	0	0	1	1	1	0	0	1	
		1	0	0	1	0	0	0	1	

$$[11/16 - 0,45]_{compl} = [11/16]_{compl} + [-0,45]_{compl}$$

$[11/16]_{compl}$: S_7 | 0 | 1 0 1 1 0 0 0 | $\oplus$

$[-0{,}45]_{compl}$: S_7 | 1 | 1 1 0 0 0 1 1 1 |

~~1~~ S_7 | 0 | 0 0 1 1 1 1 1 |

outside ⇒ DO NOT CARE ⇒ correct result!

$$0{,}0011111 = 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7}$$

Fixed point representation of real numbers

S | I bits | F bits

$\overset{\text{sign bit}}{1} + I + F = n$ bits

$X = 1324{,}37$, $n = 32$ bits

$I = 13$ bits , $F = 18$ bits

$1324 = 2454_{(8)} = \underbrace{10}_{2}\underbrace{100}_{4}\underbrace{101}_{5}\underbrace{100}_{4}{}_{(2)}$

1324 : 8 = 165, 8, 52, 48, 44, 40, = 4

165 : 8 = 20, 16, = 5, 0, 5

20 : 8 = 2, 16, = 4

2 : 8 = 0, 0, 2

0 : 8 = 0

$0,37 = 0,275341_{(8)} = 0,\underbrace{010}_{2}\underbrace{111}_{7}\underbrace{101}_{5}\underbrace{011}_{3}\underbrace{100}_{4}\underbrace{001}_{1}$

$0,37 * 8 = 2,96$
$0,96 * 8 = 7,68$
$0,68 * 8 = 5,44$
$0,44 * 8 = 3,52$
$0,52 * 8 = 4,16$
$0,16 * 8 = 1,28$

S

$\underbrace{0001}_{1}\ \underbrace{0100}_{4}\ \underbrace{1011}_{B}\ \underbrace{00|01}_{1}\ \underbrace{0111}_{7}\ \underbrace{1010}_{A}\ \underbrace{1110}_{E}\ \underbrace{0001}_{1}$

Floating point representation:

Single precision, 32 bits

S	$C=127+e$ (8 bits)	M (23 bits) — mantissa

$M > 1$

$-1324,37 = 10100101100,010111101011100001_{(2)}$

$1324,37 = 0,\underbrace{132437}_{\text{Mantisa}} \cdot 10^{4}$ $\quad 4 = e$

$$= \underbrace{1}_{\text{hidden bit}}, \underbrace{0100101100\,0101111010111\,000001}_{m} \cdot 2^{10 = e}$$

$$c = 127 + e = 127 + 10 = 137 = 2^7 + 2^3 + 2^0 =$$

$$= 10001001_{(2)}$$

$$\overset{s}{1} \,|\, \underbrace{10001001}_{c} \,|\, 0100101100\,0101111010111$$

in mantissa we put first the initial integer part and complete with the bits remaining from the fractional parts