potęga	8	7	6	5	4	3	2	1	0
	256	128	64	32	16	8	4	2	1
potęga	**-8**	**-7**	**-6**	**-5**	**-4**	**-3**	**-2**	**-1**	
	0.00390625	0.0078125	0.015625	0.03125	0.0625	0.125	0.25	1	

Zamiana 2 na 16- podziel po 4 liczby i odczytaj hex

Algorytm Hornera, gdy podstawa mniejsza niż 10: >>>

- **4 bitowe liczby ZM mają zakres:**

od -7 = $1111_{(ZM)}$

do 7 = $0111_{(ZM)}$

- **8 bitowe liczby ZM mają zakres:**

od -127 = $11111111_{(ZM)}$

do 127 = $01111111_{(ZM)}$

Liczba 23201_4 -> $?_{10}$

L0 = 2

L1 = 2*4+3=11

L2 = 11*4+2=46

L3 = 46*4+0=184

L4 = 184*4+1=737

Stąd liczba $23201_4 = 737_{10}$

4-bitowy system U1		
Kod U1	**Przeliczenie**	**Wartość**
0000	0	0
0001	2^0	1
0010	2^1	2
0011	$2^1 + 2^0$	3
0100	2^2	4
0101	$2^2 + 2^0$	5
0110	$2^2 + 2^1$	6
0111	$2^2 + 2^1 + 2^0$	7
1000	$(-2^3 + 1)$	(-7)
1001	$(-2^3 + 1) + 2^0$	(-6)
1010	$(-2^3 + 1) + 2^1$	(-5)
1011	$(-2^3 + 1) + 2^1 + 2^0$	(-4)
1100	$(-2^3 + 1) + 2^2$	(-3)
1101	$(-2^3 + 1) + 2^2 + 2^0$	(-2)
1110	$(-2^3 + 1) + 2^2 + 2^1$	(-1)
1111	$(-2^3 + 1) + 2^2 + 2^1 + 2^0$	0

4 bitowy system U2		
Kod U2	**Przeliczenie**	**Wartość**
0000	0	0
0001	2^0	1
0010	2^1	2
0011	$2^1 + 2^0$	3
0100	2^2	4
0101	$2^2 + 2^0$	5
0110	$2^2 + 2^1$	6
0111	$2^2 + 2^1 + 2^0$	7
1000	(-2^3)	(-8)
1001	$(-2^3) + 2^0$	(-7)
1010	$(-2^3) + 2^1$	(-6)
1011	$(-2^3) + 2^1 + 2^0$	(-5)
1100	$(-2^3) + 2^2$	(-4)
1101	$(-2^3) + 2^2 + 2^0$	(-3)
1110	$(-2^3) + 2^2 + 2^1$	(-2)
1111	$(-2^3) + 2^2 + 2^1 + 2^0$	(-1)

4 bitowy system ZM		
Kod ZM	**Przeliczenie**	**Wartość**
0000	$(-1)^0 \cdot 0$	0
0001	$(-1)^0 \cdot (2^0)$	1
0010	$(-1)^0 \cdot (2^1)$	2
0011	$(-1)^0 \cdot (2^1 + 2^0)$	3
0100	$(-1)^0 \cdot (2^2)$	4
0101	$(-1)^0 \cdot (2^2 + 2^0)$	5
0110	$(-1)^0 \cdot (2^2 + 2^1)$	6
0111	$(-1)^0 \cdot (2^2 + 2^1 + 2^0)$	7
1000	$(-1)^1 \cdot 0$	0
1001	$(-1)^1 \cdot (2^0)$	(-1)
1010	$(-1)^1 \cdot (2^1)$	(-2)
1011	$(-1)^1 \cdot (2^1 + 2^0)$	(-3)
1100	$(-1)^1 \cdot (2^2)$	(-4)
1101	$(-1)^1 \cdot (2^2 + 2^0)$	(-5)
1110	$(-1)^1 \cdot (2^2 + 2^1)$	(-6)
1111	$(-1)^1 \cdot (2^2 + 2^1 + 2^0)$	(-7)

Wyznaczmy wartość liczby x = –56 w kodzie U2:

1. **Moduł liczby x w kodzie binarnym to: 00111000**
2. **00111000 po zamianie bitów to 11000111**
3. **11000111 + 1 = 11001000**

c1(bias) + c2(bias) = (liczba1 + liczba2) - bias

c1(bias) - c2(bias) = (liczba1 – libcza2) +bias

Wykonać operację $0011_{(BIAS=7)} + 1010_{(BIAS=7)}$.

```
  0011
+ 1010
  1101
- 0111
  0110
```

$0011_{(BIAS=7)} + 1010_{(BIAS=7)} = 0110_{(BIAS=7)}$.

$0,1875_{10}$ -> $?_2$

$0,1875 \cdot 2 = 0,375$

$0,375 \cdot 2 = 0,75$

$0,75 \cdot 2 = 1,5$

$0,5 \cdot 2 = 1,0$

liczba = $m \cdot 2^c$

gdzie:

liczba — liczba, którą chcemy zapisać w reprezentacji zmiennopozycyjnej;

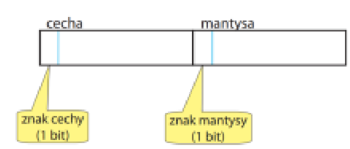
m — mantysa (ułamek właściwy);

c — cecha (liczba całkowita)

Mantysa powinna spełniać warunek |m|∈[0.1, 1). Wówczas kodowana liczba jest w postaci znormalizowanej.

Liczba 0000111011100001 zajmuje 2 bajty z czego 7 bitów to cecha, a pozostałe 9 bitów to mantysa. Liczba składa się z następujących elementów:

- **0 — bit znaku cechy,**
- **000111 — cecha,**
- **0 — bit znaku mantysy,**
- **11100001 — mantysa**



Cecha i mantysa to liczby nieujemne.

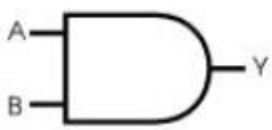
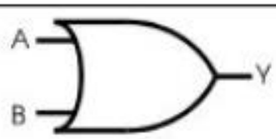
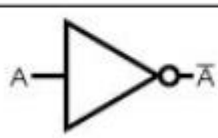
$\Delta x = |x - x_0|$ x – wynik oczekiwany

x_0 – wynik uzyskany

$$\delta = \frac{\Delta x}{x} * 100\%$$ **BŁĄD WZGLĘDNY**

element neutralny	$x + 0 = x$	$x \cdot 1 = x$
uzupełnienie	$x + \bar{x} = 1$	$x \cdot \bar{x} = 0$
przemienność	$x + y = y + x$	$x \cdot y = y \cdot x$
łączność	$(x + y) + z = x + (y + z)$	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$
rozdzielczość	$x + (y \cdot z) = (x + y) \cdot (x + z)$	$x \cdot (y + z) = x \cdot y + x \cdot z$

idempotentność	$\boldsymbol{x + x = x}$	$\boldsymbol{x \cdot x = x}$
dominacja	$x + 1 = 1$	$x \cdot 0 = 0$
inwolucja	$x = \bar{\bar{x}}$	
pochłanianie	$x + (x \cdot y) = x$	$x \cdot (x + y) = x$
uproszczenie	$x + (\bar{x} \cdot y) = x + y$	$x \cdot (\bar{x} + y) = x \cdot y$
minimalizacja	$(x \cdot y) + (x \cdot \bar{y}) = x$	$(x + y) \cdot (x + \bar{y}) = x$
prawa De Morgana	$\bar{x}+\bar{y} = \bar{x} \cdot \bar{y}$	$\overline{x \cdot y} = \bar{x} + \bar{y}$

FUNKCJA LOGICZNA	SYMBOL LOGICZNY	WYRAŻENIE ALGEBRAICZNE	TRUTH TABLE Inputs A	Inputs B	Output Y
AND		$A \cdot B = Y$	0	0	0
			0	1	0
			1	0	0
			1	1	1
OR		$A + B = Y$	0	0	0
			0	1	1
			1	0	1
			1	1	1
NOT *(Inverter)*		$A = \overline{A}$	0		1
			1		0