

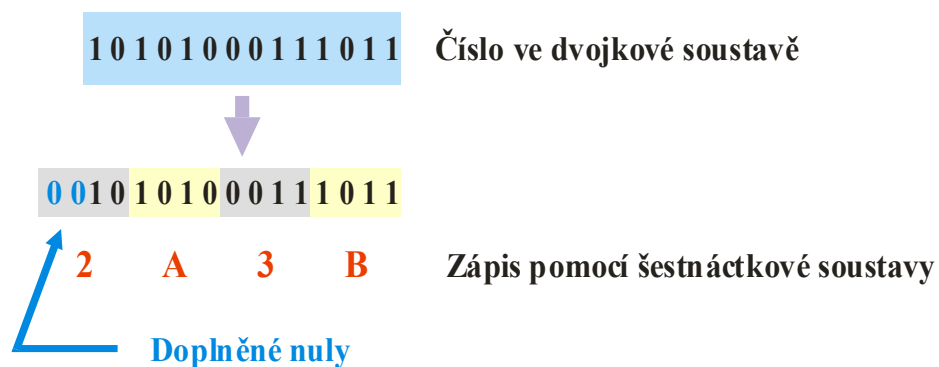
Práce s čísly

v aritmetické jednotce počítače

(Opakování)

Dvojková, osmičková a šestnáctková soustava (1)

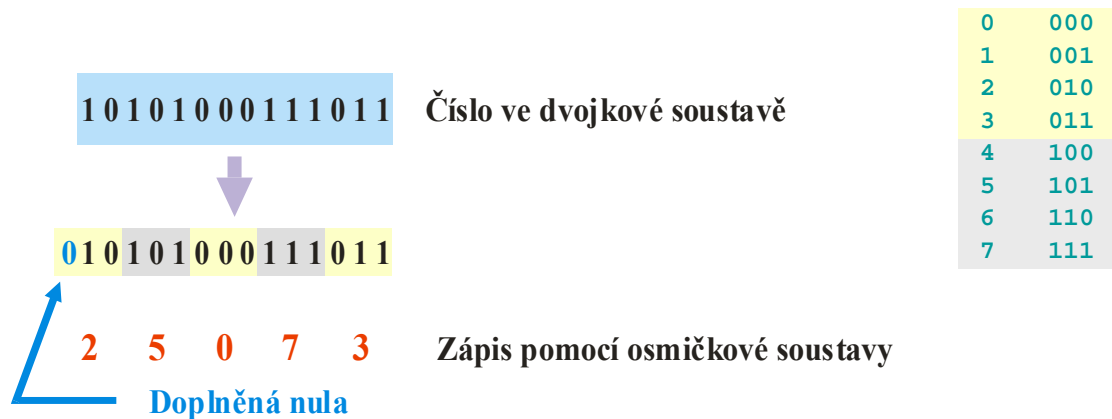
- Osmičková nebo šestnáctková soustava se používá ke snadnému zápisu binárních čísel.



0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Dvojková, osmičková a šestnáctková soustava (2)

- Osmičková nebo šestnáctková soustava se používá ke snadnému zápisu binárních čísel.

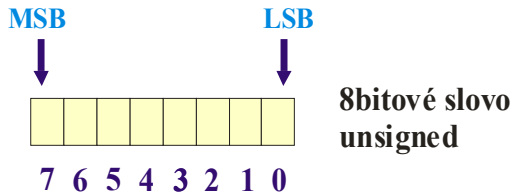


Způsoby zobrazení čísel

- Podle rozsahu a způsobu použití mohou být čísla uložena v některém z následujících formátů:
 - Celá čísla bez znaménka (unsigned integer).
 - Celá čísla se znaménkem (signed integer).
 - Celá čísla v BCD kódu.
 - Čísla v pohyblivé řádové čárce (float).
- Každý formát má při provádění aritmetických operací různá omezení, která musí programátor respektovat.

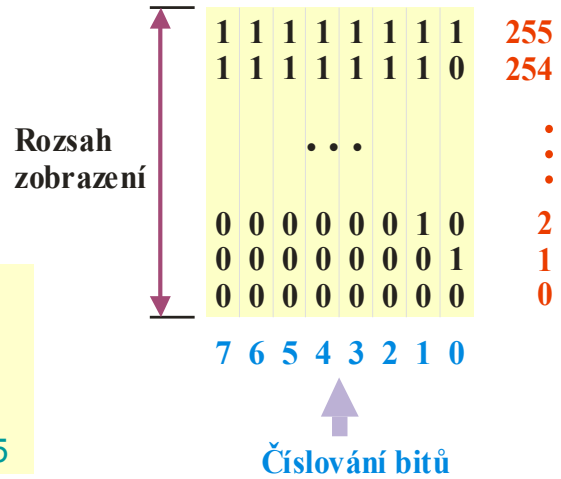
Čísla „unsigned integer“

- Na **n** bitech lze zobrazit číslo v rozsahu $0 \dots 2^n - 1$.



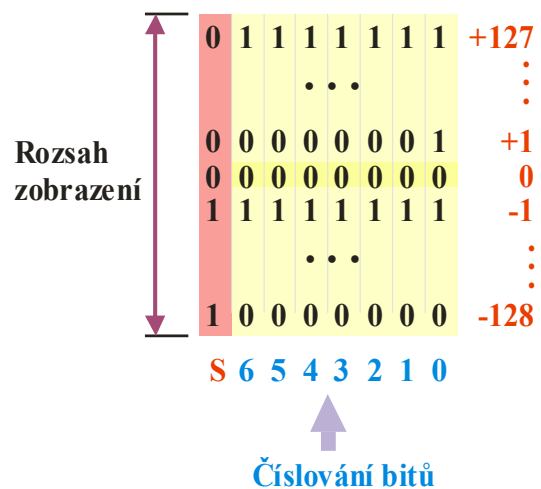
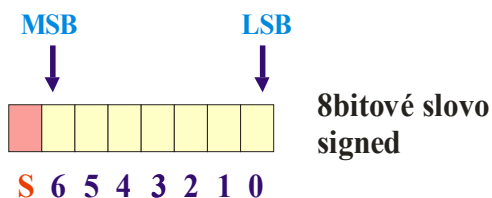
Rozsah zobrazených čísel

- 8 bitů: 0 – 255
- 16 bitů: 0 – 65535
- 32 bitů: 0 – 4 294 967 295
- 64 bitů: 0 – 18 446 744 073 709 551 615



Čísla „signed integer“

- Pro záporná čísla se používá doplňkový kód.
- Na **n** bitech lze zobrazit číslo v rozsahu $-2^{n-1} \dots 2^{n-1} - 1$.

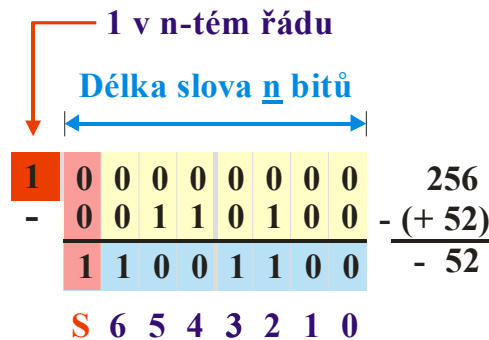


Odvození čísla „signed integer“

- Pro zobrazení na **n** bitech:

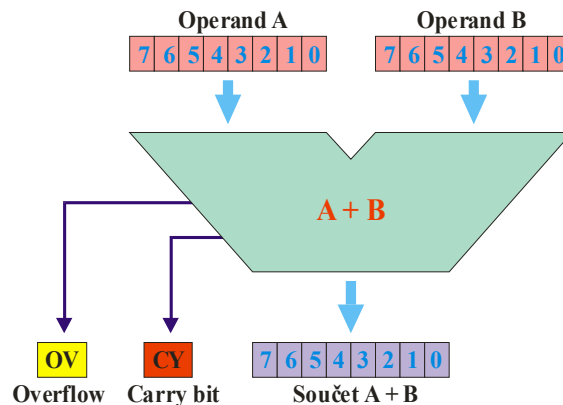
Kladné číslo se odečte od čísla 2^n (tj. **1** v řádu, který je mimo rozsah zobrazení).

Příklad: číslo -52:



Jednoduchá sčítačka

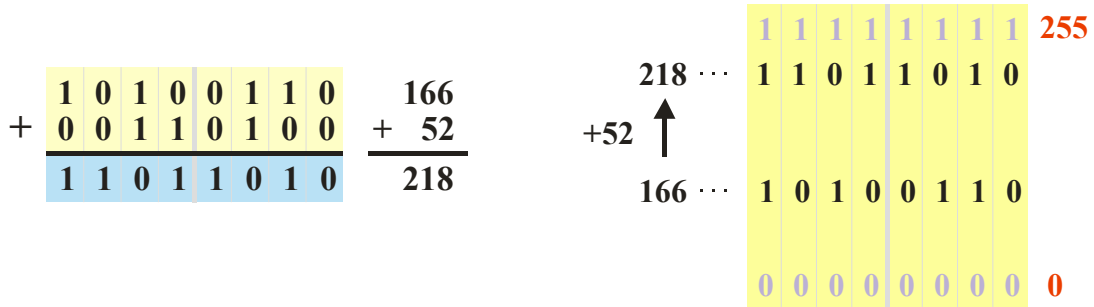
- Sčítačka (a odčítačka) pro 8bitová čísla integer.
- Kromě 8bitového výsledku generuje dva příznakové bity:
 - **Carry bit** – přenos do vyššího řádu.
 - **Overflow** – příznak přetečení čísel signed integer.



Sčítačka pracuje stejně s čísly unsigned integer i signed integer (!).

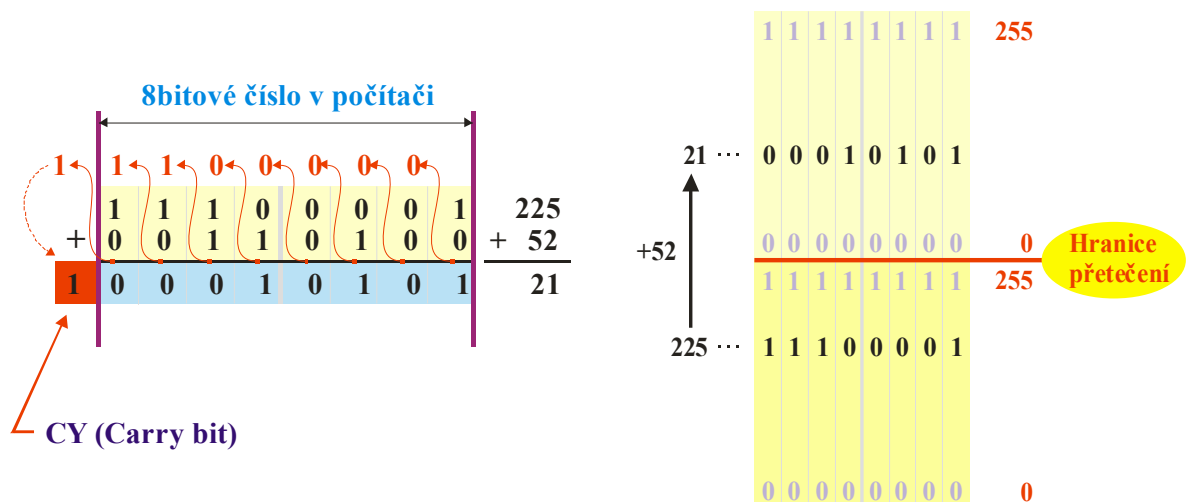
Sečítání čísel „unsigned integer“

- Dále budeme předpokládat, že délka slova operandů i výsledku je omezena na 8 bitů.
 - V praxi je délka slova omezena na 8, 16, 32 nebo 64 bitů.



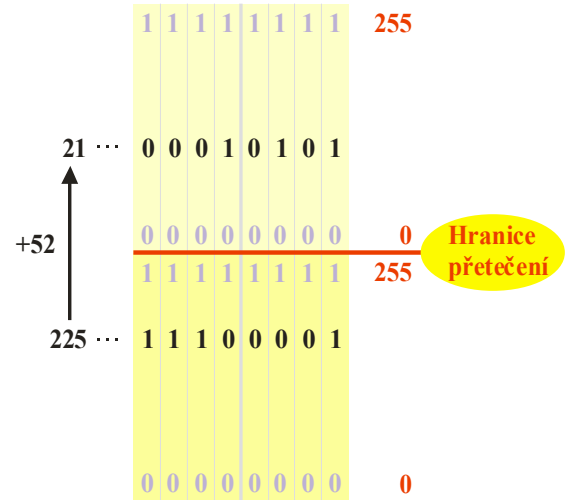
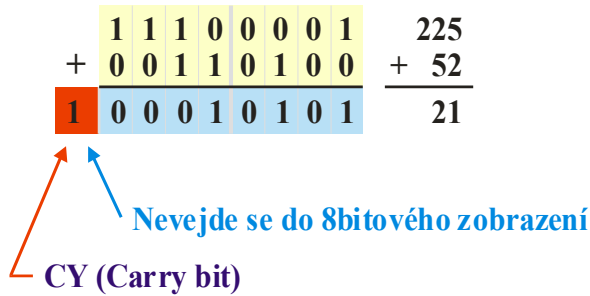
Sečítání čísel „unsigned integer“

- Při překročení rozsahu zobrazení dojde k přetečení \Rightarrow dostaneme nesprávný výsledek.
- Přetečení je indikováno příznakovým bitem **CY**.



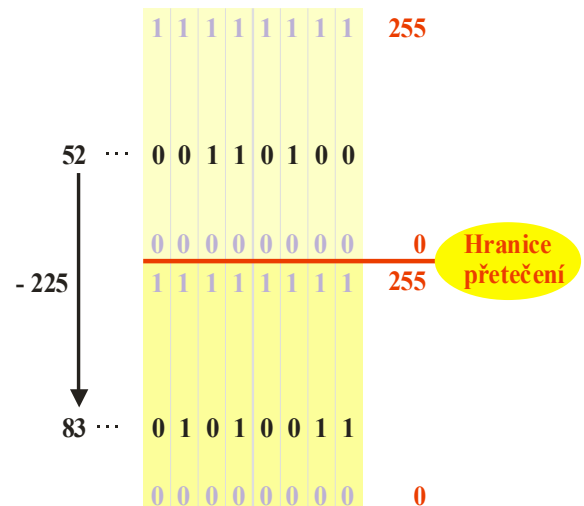
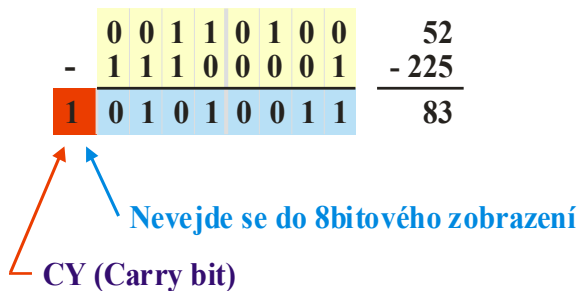
Sečítání čísel „unsigned integer“

- Při překročení rozsahu zobrazení dojde k přetečení ⇒ dostaneme nesprávný výsledek.
- Přetečení je indikováno příznakovým bitem **CY**.



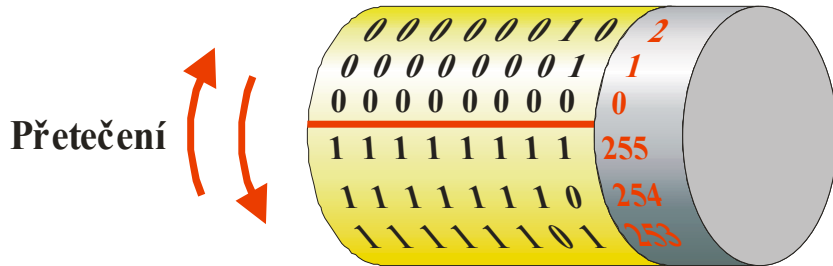
Odečítání čísel „unsigned integer“

- Při překročení rozsahu zobrazení dojde k přetečení ⇒ dostaneme nesprávný výsledek.



Rozsah zobrazení pro „unsigned integer“

- Hranice přetečení (pro 8bitové zobrazení) je mezi čísly 0 a 255.



Sečítání čísel „signed integer“

- Přetečení je indikováno příznakovým bitem Overflow (OV).

+	0	1	1	0	1	0	0	1	105
+	0	0	1	0	0	1	0	1	+ 37
	0	1	0	0	1	1	1	0	- 114

└───┬───> S = 1

└───┬───> CY = 0

1 ───> OV (Overflow) = 1

	0	1	1	1	1	1	1	1	+127
	0	0	0	0	0	0	0	0	0
- 114 ...	1	0	0	0	1	1	1	0	
+ 37	1	0	0	0	0	0	0	0	-128
0	0	1	1	1	1	1	1	1	+127
105 ...	0	1	1	0	1	0	0	1	
	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	-128

Hranice přetečení

Podmínky pro nastavení bitu OV

Příznakový bit **OV** se nastavuje podle typu operace a znamének operandů a výsledku.

Znaménko 1. operandu	Znaménko 2. operandu	Znaménko výsledku	Operace	OV
+	+	+	ADD	0
+	+	-	ADD	1
-	-	-	ADD	0
-	-	+	ADD	1
+	-	+	SUB	0
+	-	-	SUB	1
-	+	-	SUB	0
-	+	+	SUB	1
+	-	×	ADD	0
-	-	×	SUB	0
+	+	×	SUB	0

Shrnutí (1)

- Sčítačka pracuje stejně s čísly signed i unsigned.
- Výsledek a význam bitů **CY** a **OV** se liší podle toho, jsou-li operandy interpretovány jako číslo signed nebo unsigned.

$$\begin{array}{r}
 1\ 1\ 1\ 0\ 0\ 0\ 0\ 1 \\
 +\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0 \\
 \hline
 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1
 \end{array}$$

$$\begin{array}{l}
 \text{CY} = 1 \\
 \text{OV} = 0
 \end{array}$$

$$\begin{array}{r}
 225 \\
 +\ 52 \\
 \hline
 21
 \end{array}$$

Unsigned

$$\begin{array}{r}
 -\ 31 \\
 +\ 52 \\
 \hline
 21
 \end{array}$$

Signed

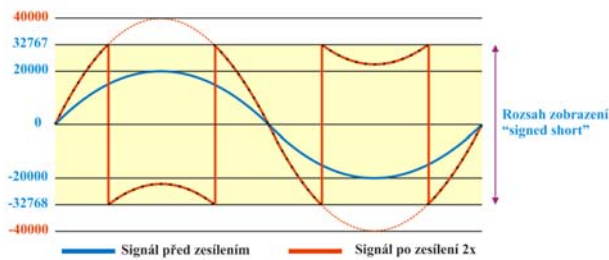
Shrnutí (2)

- Sčítačka pracuje stejně s čísly signed i unsigned.
- Výsledek a význam bitů **CY** a **OV** se liší podle toho, jsou-li operandy interpretovány jako číslo signed nebo unsigned.

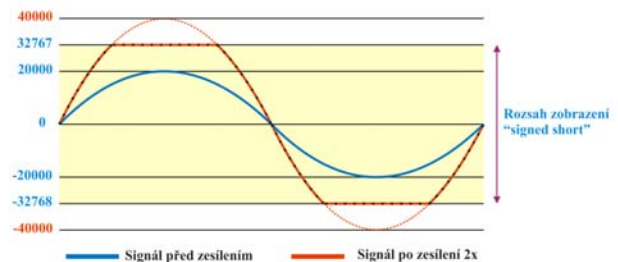
	0 1 1 0 1 0 0 1	105	105
+	0 0 1 0 0 1 0 1	+ 37	+ 37
=	0 1 0 0 0 1 1 1 0	142	- 114
	CY = 0 OV = 1	Unsigned	Signed

Poznámka: Aritmetika se saturací

- Při přetečení se výsledek nahradí největším (resp. nejmenším) zobrazitelným číslem.
- Použití např. při zpracování signálů (signálové procesory).



Normální ALU



ALU se saturací

Porovnávání čísel integer

- Při porovnávání se musí rozlišovat mezi čísly unsigned a signed.

1 1 1 0 0 0 0 1 A = 225

0 0 1 1 0 1 0 0 B = 52

Unsigned: A > B

Pro unsigned se používají operace „nad“ a „pod“ (above / below).

1 1 1 0 0 0 0 1 A = -31

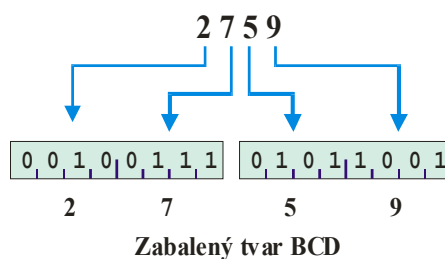
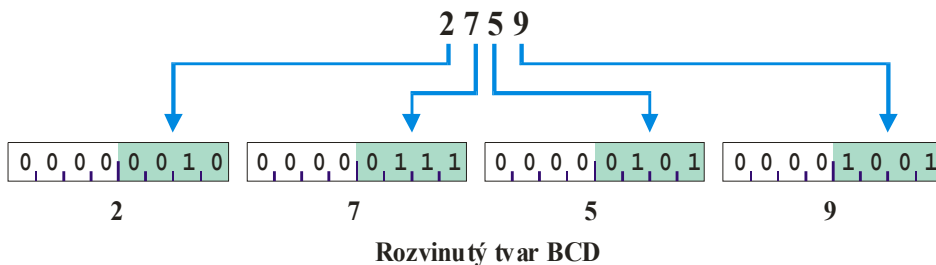
0 0 1 1 0 1 0 0 B = 52

Signed: A < B

Pro signed se používají operace „větší“ a „menší“ (greater / less).

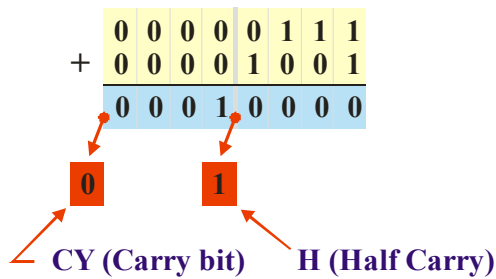
BCD kód

- Jednotlivé dekadické číslice se ukládají v binárním tvaru.
- Např. 2759₁₀:



Sečítání BCD čísel (1)

- BCD čísla se sečítají běžnou sčítačkou instrukcí **ADD**.
- Sčítačka generuje bit **H** – přenos mezi 3. a 4. bitem (**H**alf **C**arry).
- Po sečtení BCD čísel se musí provádět dekadická korekce speciální instrukcí **DAA** (**D**ecimal **A**djust **A**dd).



Instrukce DAA:

```

if( ((operand > 9) && (H == 0)) ||
    ((operand < 4) && (H == 1)) ) {
    operand = operand + 6;
}
    
```

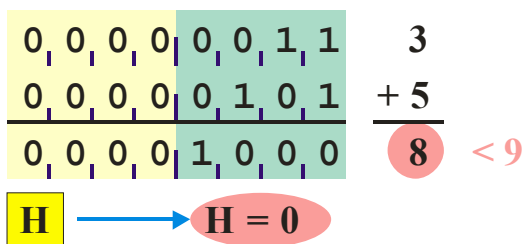
Poznámka: Operand jsou jen bity 3...0.

Sečítání BCD čísel (2)

- Sčítačka generuje bit **H** – přenos mezi 3. a 4. bitem (**H**alf **C**arry).
- Po sečtení čísel BCD se musí provádět dekadická korekce:

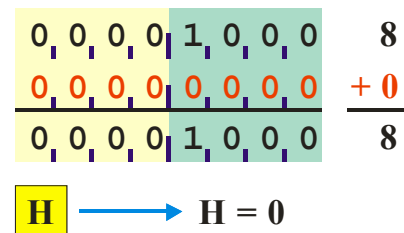
Příklad: $3 + 5 = 8$

Sečtení (ADD)



H = Half Carry

Dekadická korekce (DAA)



H = 0 a současně výsledek < 9
 \Rightarrow DAA neprovede úpravu výsledku

Sečítání BCD čísel (3)

- Po sečtení čísel BCD se musí provádět dekadická korekce:

Příklad: 7 + 5

Sečtení (ADD)

0	0	0	0	0	1	1	1	7
0	0	0	0	0	1	0	1	+ 5
0	0	0	0	1	1	0	0	C > 9

H → **H = 0**

Dekadická korekce (DAA)

0	0	0	0	1	1	0	0	C
0	0	0	0	0	1	1	0	+ 6
0	0	0	1	0	0	1	0	1 2

H → **H = 1**

H = 0, **ale výsledek > 9**
⇒ DAA upraví výsledek

H = Half Carry

Sečítání BCD čísel (4)

- Po sečtení čísel BCD se musí provádět dekadická korekce:

Příklad: 9 + 8

Sečtení (ADD)

0	0	0	0	1	0	0	1	9
0	0	0	0	1	0	0	0	+ 8
0	0	0	1	0	0	0	1	1 1 < 4

H → **H = 1**

Dekadická korekce (DAA)

0	0	0	1	0	0	0	1	1 1
0	0	0	0	0	1	1	0	+ 6
0	0	0	1	0	1	1	1	1 7

H → **H = 1**

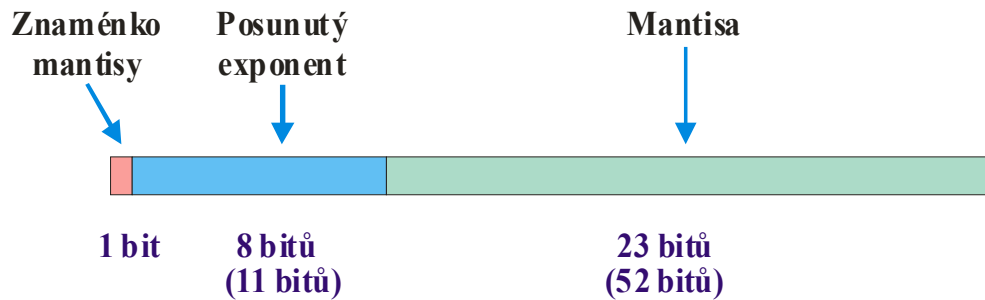
H = 1, **výsledek < 4**
⇒ DAA upraví výsledek

Číslo v pohyblivé řádové čárce

- Před uložením jsou převedena do normalizovaného tvaru:

$$1.\text{dddd}\dots\text{d} \times 2^{\text{exp}}$$

- Formát uloženého čísla (podle IEEE 754):
 - Podle přesnosti je číslo uloženo na 32 nebo 64 bitech.



Převod čísla do normalizovaného tvaru

- Příklad: normalizace čísla $+6.625_{10}$.

$$+6.625_{10} = 110.101_2$$

- Normalizovaný tvar:

$$110.101 = 1.10101 \times 2^2$$

Dvojkový exponent

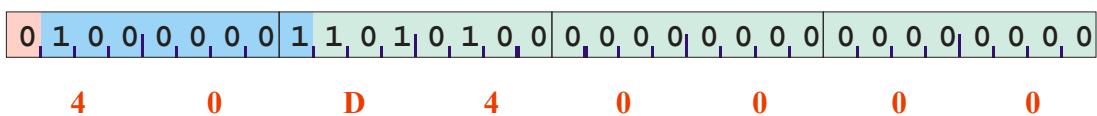
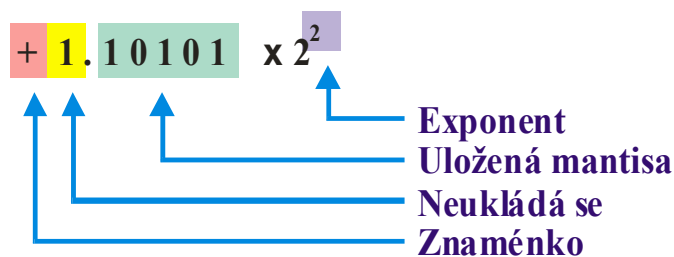
- Exponent se ukládá na 8 (resp. 11) bitech v posunutém tvaru.

$$\text{Posunutý exponent} = \text{Exponent} + 127$$

Exponent	Posunutý exponent
Zvláštní význam	1111 1111
+ 127	1111 1110
+ 1	1000 0000
0	0111 1111
- 126	0000 0001
Zvláštní význam	0000 0000

Mantisa

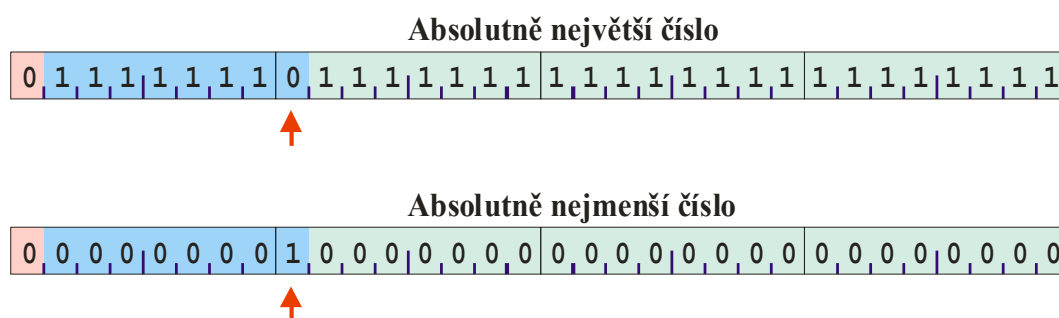
- Zobrazuje se v přímém kódu.
- Číslice **1** před řádovou čárkou se neukládá.



Úplné zobrazení čísla 6.625 ⇒ **0x40D40000**

Příklady zobrazení (3)

- Absolutně největší číslo:
 $2^{128} - \text{LSB}$
- Absolutně nejmenší číslo:
 2^{-126}



Rozsah zobrazení

