Úvod, Číselné soustavy a kódy

Úvod do digitální techniky, číselné soustavy, převody, aritmetické operace, číselné kódy

Ing. Pavel Lafata, Ph.D.
lafatpav@fel.cvut.cz

- **B2B32DITA – Digitální technika**
  - **rozsah výuky – 2p + 2l**, **zakončení předmětu – klasifikovaný zápočet**, **4 kredity**
  - **vyučující** – Katedra telekomunikační techniky, FEL, ČVUT v Praze:
    přednášející, garant – **Ing. Pavel Lafata, Ph.D.**, místnost 810, e-mail: **lafatpav@fel.cvut.cz**
    vedoucí cvičení – **Ing. Tomáš Zeman, Ph.D.**, místnost 711, e-mail: **zeman@fel.cvut.cz**
    asistent – **Ing. Josef Šebánek**, e-mail: **sebanjos@fel.cvut.cz**
  - <u>stručná osnova předmětu</u>:
    - **Digitální systémy, číselné soustavy a kódy**
    - **Logické funkce** – elementární log. funkce a hradla, vyjadřování log. funkcí, Booleova algebra, minimalizace log. funkcí pomocí Karnaughových map, realizace log. funkcí pomocí hradel
    - **Kombinační a sekvenční log. obvody** – rozdíly, příklady komb. obvodů, logické hazardy, klopné obvody, čítače, konečné stavové automaty (Mealy vs. Moore)
    - **Technologie pro realizaci logických obvodů** – TTL, CMOS, FPGA
    - **Moderní způsoby realizace log. obvodů** – programovatelná pole FPGA a jazyk VHDL
  - **klasifikovaný zápočet** – maximální počet 100 bodů
    - **průběžné hodnocení na cvičení + zápočtový test**
    - **ze cvičení lze získat maximálně 49 bodů** (vstupní testy + laboratorní měření)
    - **zápočtový test** – příklady + písemné otázky, **z testu lze získat až 51 bodů (podmínkou je získat více než 50%, tj. 26 bodů z testu)**
    - hodnocení ECTS, podrobněji viz úvodní cvičení a v průběhu semestru

- **B2B32DITA – Digitální technika**

1. **Přednášky předmětu – MS Teams, materiály na seminární cvičení, návody na lab. úlohy** – vše dostupné na Moodlu v kurzu B2B32DITA: **https://moodle.fel.cvut.cz/**
2. **Skripta:**

[1] Lafata, P. - Hampl, P. - Pravda, M.: Digitální technika. 1. vyd. Praha: Česká technika - nakladatelství ČVUT, 2011. 164 s. ISBN 978-80-01-04914-3.

3. **Další užitečná literatura:**

[1] Pinker, J. - Poupa, M.: Číslicové systémy a jazyk VHDL. Praha : BEN - technická literatura, 2006. 349 s. ISBN 80-7300-198-5.

[2] Šťastný, J.: FPGA prakticky: realizace číslicových systémů pro programovatelná hradlová pole. Praha : BEN - technická literatura, 2010. 199 s. ISBN 978-80-7300-261-9.

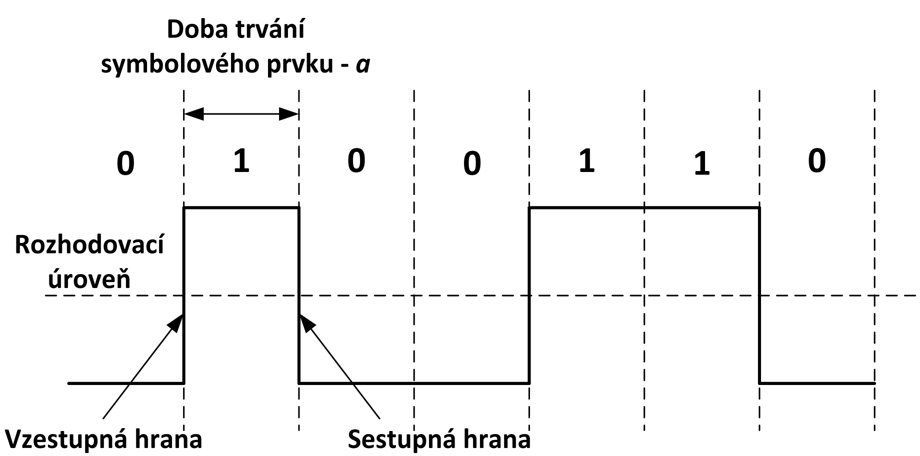
[3] Antošová, M. - Davídek, V.: Číslicová technika. České Budějovice : KOPP, 2003. 286 s. ISBN 80-7232-206-0.

[4] Strnad, L.: Základy číslicové techniky: cvičení. Praha : ČVUT, 1996. 124 s. ISBN 80-01-01433-9.

[5] Ashender, P., J.: The VHDL Cookbook. online: https://tams-www.informatik.uni-hamburg.de/vhdl/doc/cookbook/VHDL-Cookbook.pdf.

[6] SYNARIO: VHDL Reference Manual. online: http://www.ics.uci.edu/~jmoorkan/vhdlref/Synario%20VHDL%20Manual.pdf.

- # Úvod
  - **digitální systém, digitální obvod** – obecná definice?
    - digitální systém (obvod) = obecně jakýkoliv obvod umožňující zpracovávat digitální signály
  - co je tedy **digitální signál**?
    - sekvence diskrétních hodnot – signál nespojitý v čase i amplitudě
    - **3 operace** při převodu analogového signálu na digitální:
      vzorkování – odebírání vzorků z analogového signálu v periodických časových okamžicích (vzorkovací teorém) – diskretizace v časové oblasti
      kvantování – signál spojitý v amplitudě se převádí na soubor diskrétních hodnot pomocí kvantizačních hladin (kvantizační šum) – diskretizace v amplitudové oblasti
      kódování – přiřazení (vyjádření) kódové hodnoty kvantizačním hladinám
  - **logický signál (dvouhodnotová logika)** – digitální signál nabývající jen 2 hodnot
    - logická 1 vs. logická 0, high (H) vs. low (L), pravda (true) vs. nepravda (false)…



modulační rychlost

$$v_m = \frac{1}{a} \quad [\mathrm{Bd}]$$

- **Úvodní pojmy**
  - **logický signál, logická funkce**
    - **logická funkce** – soubor pravidel pro jednoznačné přiřazení výstupních logických hodnot kombinacím vstupních logických proměnných
    - **dvouhodnotová logika** (existují i jiné) – operace s dvěma hodnotami – logická 1 a logická 0 – s výhodou lze využít **binární (dvojkovou) polyadickou číselnou soustavu**
    - **logický obvod** – digitální obvod pracující s logickými signály

- **Číselné soustavy**
  - **číselná soustava** = soubor pravidel pro jednoznačný zápis libovolného čísla pomocí jednotlivých číslic
  - **číslice** = symbol reprezentující v dané číselné soustavě určitou hodnotu:
    např. v desítkové soustavě – 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

1. **polyadické (poziční) číselné soustavy**
- stejný základ na všech řádových místech, vyjádření daného čísla pomocí mnohočlenu

2. **nepolyadické (nepoziční) číselné soustavy**
- různý základ na jednotlivých řádových místech, nelze rozvinout pomocí mnohočlenu

- **Polyadické (poziční) číselné soustavy**
  - **základ** číselné soustavy – přirozené číslo ***Z***, větší než 1: ***Z > 1***
  - pojmenování číselných soustav pomocí jejich základů, nejpoužívanější:
    dvojková (binární) soustava – základ číslo **2** (Z = 2)
    osmičková (oktalová) soustava – základ číslo **8** (Z = 8)
    desítková (dekadická) soustava – základ číslo **10** (Z = 10)
    šestnáctková (hexadecimální) soustava – základ číslo **16** (Z = 16)
  - libovolné kladné číslo *N* lze v polyadické číselné soustavě se základem *Z* zapsat pomocí polynomického rozvoje:

$$N_{(Z)} = \sum_{i=-n}^{m-1} a_i \cdot Z^i = a_{m-1} \cdot Z^{m-1} + a_{m-2} \cdot Z^{m-2} + \ldots + a_1 \cdot Z^1 + a_0 \cdot Z^0 + a_{-1} \cdot Z^{-1} + \ldots + a_{-n+1} \cdot Z^{-n+1} + a_{-n} \cdot Z^{-n}$$

  - $\boldsymbol{N_{(Z)}}$ – je číslo vyjádřené v číselné soustavě se základem ***Z***
    Z – je základ číselné soustavy
    $\boldsymbol{a_i}$ – jsou číselné koeficienty (číslice), pro které platí $a_i \in \{0, 1, 2 \ldots, Z-1\}$
    m – je počet míst na kterých má základ *Z* kladný exponent, celá část čísla $N_{(Z)}$
    n – je počet míst na kterých má základ *Z* záporný exponent, desetinná část čísla $N_{(Z)}$
    , – je desetinná čárka, odděluje celou a desetinnou část čísla $N_{(Z)}$

  - příklad rozvoje libovolného čísla v desítkové soustavě:
    $\mathbf{1307{,}12_{(10)} = 1x10^3 + 3x10^2 + 0x10^1 + 7x10^0 + 1x10^{-1} + 2x10^{-2}}$

- **kapacita soustavy, maximální číslo v dané soustavě**
  - ***K*** – kapacita číselné soustavy se základem ***Z*** a s ***m*** kladnými řádovými místy:
    $K = Z^m$
  - N_{max} – nejvyšší číslo číselné soustavy se základem ***Z*** a s ***m*** kladnými řádovými místy:
    $N_{max} = Z^m - 1$

1. **převod z desítkové soustavy do soustavy se základem Z**
   - pokud obsahuje číslo desetinnou část, převedeme nejprve celou část a pak desetinnou
   - dva možné postupy:
   1. postupné dělení základem soustavy Z
   2. postupné dělení mocninami základu soustavy Z
   - **příklad** – převeďte číslo z desítkové soustavy $41_{(10)}$ do dvojkové soustavy (**Z = 2**)
   - metoda 1 – postupné dělení základem soustavy Z:

     41 : 2 = 20 + **1**/2 ↑ LSB
     20 : 2 = 10 + **0**/2
     10 : 2 = 5 + **0**/2
     5 : 2 = 2 + **1**/2
     2 : 2 = 1 + **0**/2
     1 : 2 = 0 + **1**/2 MSB
   - výsledek obdržíme sepsáním jednotlivých zbytků dělení směrem odspodu nahoru:
     $\underline{\underline{\mathbf{41_{(10)} = 101001_{(2)}}}}$

- metoda 2 – dělením mocninami základu Z:
  mocniny základu 2:
  $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, $2^7 = 128$…
  nejprve vydělíme převáděné číslo $N_{(10)}$ nejbližší nižší mocninou základu *Z* – dále snižujeme o 1 mocninu základu a postupně dělíme zbytky po předchozím dělení, až se dostaneme do posledního dělení 1:
  41 : 32 = **1** + 9/2 MSB
  9 : 16 = **0** + 9/2
  9 : 8 = **1** + 1/2
  1 : 4 = **0** + 1/2
  1 : 2 = **0** + 1/2
  1 : 1 = **1** + 0/2 LSB
- výsledek obdržíme sepsáním jednotlivých výsledků dělení směrem seshora dolu:
  $\underline{\underline{41_{(10)} = 101001_{(2)}}}$

- převod desetinného (neceléno) čísla z desítkové soustavy do soustavy se základem Z:

1. nejprve převedeme zvlášť celou část čísla (metodou 1 či 2)
2. pak převedeme zvlášť desetinnou část, předchozí metodou 2, nebo pomocí metody postupného násobení základem Z a odečítáním celé části výsledku

Úvod, Číselné soustavy a kódy – Polyadické číselné soustavy, převody

- příklad – převeďte číslo $54{,}6875_{(10)}$ do dvojkové soustavy
  nejprve převedeme celou část čísla (například metodou 1):
  54 : 2 = 27 + **0**/2 LSB
  27 : 2 = 13 + **1**/2
  13 : 2 = 6 + **1**/2
  6 : 2 = 3 + **0**/2
  3 : 2 = 1 + **1**/2
  1 : 2 = 0 + **1**/2 MSB
  dále převedeme desetinnou část pomocí obou způsobů:
  - **způsob 1** – dělení mocninami čísla 2 (metoda 2):
  mocniny 2: $\mathbf{2^{-1} = 0{,}5,\ 2^{-2} = 0{,}25,\ 2^{-3} = 0{,}125,\ 2^{-4} = 0{,}0625,\ 2^{-5} = 0{,}03125}$…
  0,6875 : 0,5 = **1** + 0,1875/2 MSB
  0,1875 : 0,25 = **0** + 0,1875/2
  0,1875 : 0,125 = **1** + 0,0625/2
  0,0625 : 0,0625 = **1** + 0/2 LSB
  - **způsob 2** – násobení základem a odečítání celé části výsledku:
  0,6875 x 2 = 1,3750 – odečteme celou část **1**, desetinná část 0,3750 MSB
  0,3750 x 2 = 0,7500 – odečteme celou část **0**, desetinná část 0,7500
  0,7500 x 2 = 1,5000 – odečteme celou část **1**, desetinná část 0,5000
  0,5000 x 2 = 1,0000 – odečteme celou část **1**, desetinná část 0,0000 LSB
  výsledek: $\underline{\underline{\mathbf{54{,}6875_{(10)} = 110110{,}1011_{(2)}}}}$
- převod z desítkové do libovolné jiné **číselné soustavy se základem Z** – zcela stejný postup

2. převod čísel ze soustavy se základem Z do desítkové soustavy

- jednoduše dosadíme do polynomiálního rozvoje čísla
- příklad – převeďte číslo z dvojkové soustavy $10011101{,}10001_{(2)}$ do desítkové:
  $\mathbf{10011101{,}10001_{(2)}} = 1x2^7 + 0x2^6 + 0x2^5 + 1x2^4 + 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0 + 1x2^{-1} + 0x2^{-2} + 0x2^{-3} + 0x2^{-4} + 1x2^{-5} = 128 + 0 + 0 + 16 + 8 + 4 + 0 + 1 + 0{,}5 + 0 + 0 + 0 + 0{,}03125 =$
  $\underline{\underline{\mathbf{157{,}53125_{(10)}}}}$
- převod z libovolné jiné soustavy se základem Z do desítkové je zcela stejný

- **příklady s převody z/do osmičkové a šestnáctkové soustavy:**
  - převeďte číslo z desítkové soustavy $109{,}125_{(10)}$ do osmičkové:

    $109 : 8 = 13 + \mathbf{5}/8$
    $13 : 8 = 1 + \mathbf{5}/8$
    $1 : 8 = 0 + \mathbf{1}/8$

    $0{,}125 \text{ x } 8 = \mathbf{1} + 0{,}000$

    výsledek: $\underline{\underline{\mathbf{109{,}125_{(10)} = 155{,}1_{(8)}}}}$

  - převeďte číslo z osmičkové soustavy $307{,}21_{(8)}$ do desítkové:
    $307{,}21_{(8)} = 3x8^2 + 0x8^1 + 7x8^0 + 2x8^{-1} + 1x8^{-2} = 192 + 0 + 7 + 0{,}25 + 0{,}015625 =$
    $\underline{\underline{\mathbf{199{,}265625_{(10)}}}}$

Úvod, Číselné soustavy a kódy – Polyadické číselné soustavy, převody

- šestnáctková (hexadecimální) soustava – číslice 0, 1, 2, …, 9, A, B, C, D, E, F
  A = 10 v desítkové soustavě
  B = 11 v desítkové soustavě
  C = 12 v desítkové soustavě
  D = 13 v desítkové soustavě
  E = 14 v desítkové soustavě
  F = 15 v desítkové soustavě
- příklad – převeďte číslo z desítkové soustavy $698{,}1875_{(10)}$ do šestnáctkové:
  $698 : 16 = 43 + \mathbf{10}/16$ $\quad 10 = \mathbf{A}$
  $43 : 16 = 2 + \mathbf{11}/16$ $\quad 11 = \mathbf{B}$
  $2 : 16 = 0 + \mathbf{2}/16$

  $0{,}1875 \times 16 = \mathbf{3} + 0{,}0000$

  výsledek: $\underline{\underline{\mathbf{2BA{,}3_{(16)}}}}$

- příklad – převeďte číslo z šestnáctkové soustavy $E0F{,}A8_{(16)}$ do desítkové:
  $E0F{,}A8_{(16)} = 14 \times 16^2 + 0 \times 16^1 + 15 \times 16^0 + 10 \times 16^{-1} + 8 \times 16^{-2} = 3584 + 0 + 15 + 0{,}625 + 0{,}03125 = \underline{\underline{\mathbf{3599{,}65625_{(10)}}}}$

3. převod ze soustavy se základem Z1 do soustavy se základem Z2

- **obecně** – převedeme číslo ze soustavy se základem Z1 do desítkové a následně provedeme převod čísla z desítkové soustavy do soustavy se základem Z2
- pokud je však základ **Z1 mocninou základu Z2** (nebo naopak), lze převod provést jednodušeji – použitelné např. pro převody dvojková – osmičková, dvojková – šestnáctková, neboť $\mathbf{2^3 = 8}$ a $\mathbf{2^4 = 16}$
- např. převod dvojková ↔ osmičková soustava, neboť $\mathbf{2^3 = 8}$ – tedy **3 řádová místa (číslice) ve dvojkové soustavě představují 1 řádové místo v osmičkové soustavě**
- a také dvojková ↔ šestnáctková soustava, neboť $\mathbf{2^4 = 16}$ – tedy **4 řádová místa (číslice) ve dvojkové soustavě představují 1 řádové místo v šestnáctkové soustavě**
- při převodu **z dvojkové do osmičkové** (nebo **šestnáctkové**) soustavy, **rozdělujeme převáděné číslo vždy od konce!** (od LSB směrem k MSB) **po trojicích** (nebo **čtveřicích**)
- při opačném převodu **z osmičkové** (nebo **šestnáctkové**) soustavy **do dvojkové**, **rozepisujeme každou číslici na trojici** (nebo **čtveřici**) ve dvojkové soustavě

- **příklady:**
- převeďte číslo z dvojkové soustavy $1011100111_{(2)}$ do osmičkové:
  $001|011|100|111_{(2)} = 1347_{(8)}$ – rozdělujeme převáděné číslo na trojice vždy odzadu! na začátek čísla můžeme doplnit nuly pro získání poslední trojice (nebo čtveřice), nula před číslem nezmění jeho hodnotu!

3. **převod ze soustavy se základem Z1 do soustavy se základem Z2**
   - **další příklady:**
   - převeďte číslo z dvojkové soustavy $110111001101011_{(2)}$ do šestnáctkové:
     $0110|1110|0110|1011_{(2)} = 6E6B_{(16)}$
   - převeďte číslo z osmičkové soustavy $7013_{(8)}$ do dvojkové soustavy:
     $7013_{(8)} = 111|000|001|011_{(2)}$ – každou 1 číslici v osmičkové soustavě musíme převést jako 3 číslice ve dvojkové soustavě, **0** tedy není 0 ale **000**!, **1** není 1 ale **001**!
   - převeďte číslo ze šestnáctkové soustavy $3AC1_{(16)}$ do dvojkové:
     $3AC1_{(16)} = \cancel{00}11|1010|1100|0001_{(2)}$ – každou 1 číslici v šestnáctkové soustavě musíme převést jako 4 číslice ve dvojkové soustavě, 3 není 11 ale 0011, 1 není 1 ale 0001 dvojici nul na počátku výsledného čísla v dvojkové soustavě můžeme vypustit

- **Čísla signed (se znaménkem) vs. unsigned (bez) – vyjádření záporných čísel**
  1. **znaménkový bit** (tzv. signum bit)
     - vyhrazení jednoho bitu (řádového místa) pro znaménko – tzv. signum bit
     - pokud první bit (signum bit) = 0 – kladné číslo, pokud bit = 1 – záporné číslo
     - příklad: 7 bitové binární číslo (*m* = 7) jeho rozsah může být $0000000\text{-}1111111_{(2)}$ (0 až 127), přidáme 8. (signum) bit, získáme $\mathbf{01111111_{(2)} = +127}$, $\mathbf{11111111_{(2)} = -127}$
  2. **logický doplněk** (tzv. one's complement)
  3. **dvojkový doplněk** (tzv. two's complement)

Úvod, Číselné soustavy a kódy – Logický doplněk, dvojkový doplněk, záporná čísla

- **logický doplněk**
  - záporné číslo je vytvořeno pomocí inverze na všech řádových místech binárního čísla (zaměníme 0 za 1 a naopak)
  - symetrický interval kolem nuly +n-1… -n+1, dvě vyjádření pro nulu
- **dvojkový doplněk**
  - nejprve získáme logický doplněk (inverze 0 a 1) a pak přičteme + 1 = dvojkový doplněk
  - nesymetrický interval kolem nuly +n-1… -n

číslo bez znaménka (nezáporné) – dvojková/desítková soust.	**logický doplněk – dvojková/desítková soust.**	**dvojkový doplněk – dvojková/desítková soust.**
$000_{(2)} = 0_{(10)}$	$111_{(2)} = -0_{(10)}$	$000_{(2)} = 0_{(10)}$
$001_{(2)} = 1_{(10)}$	$110_{(2)} = -1_{(10)}$	$111_{(2)} = -1_{(10)}$
$010_{(2)} = 2_{(10)}$	$101_{(2)} = -2_{(10)}$	$110_{(2)} = -2_{(10)}$
$011_{(2)} = 3_{(10)}$	$100_{(2)} = -3_{(10)}$	$101_{(2)} = -3_{(10)}$
$100_{(2)} = 4_{(10)}$	$011_{(2)} = 3_{(10)}$	$100_{(2)} = -4_{(10)}$
$101_{(2)} = 5_{(10)}$	$010_{(2)} = 2_{10)}$	$011_{(2)} = 3_{(10)}$
$110_{(2)} = 6_{(10)}$	$001_{(2)} = 1_{(10)}$	$010_{(2)} = 2_{(10)}$
$111_{(2)} = 7_{(10)}$	$000_{(2)} = 0_{(10)}$	$001_{(2)} = 1_{(10)}$

- **Číselné kódy**
- **číselný kód** = soubor pravidel pro jednoznačné přiřazení znaku (číslice, symbolu) jeho hodnotě na dané pozici v čísle
- různé kódy přinášejí různé výhody pro specifické aplikace
- **Hammingova vzdálenost** = počet pozic (řádových míst), na kterých se dvě stejně dlouhá kódová slova liší (příklad, Hammingova vzdálenost mezi **0000** a **1111** je **4**)
- **BCD kód** = **Binary-coded decimal**
  - každá číslice na jednotlivé řádové pozici v desítkové soustavě je vyjádřena v binární soustavě pomocí čtveřice číslic (bitů)
  - výhodou je rychlost a snadnost převodu (čitelnost), nevýhodou je redundance
  - příklad: číslo v desítkové soustavě $23_{(10)}$ -> 0010 0011 v BCD kódu
- **Grayův kód**
  - binární kód v němž se dvě po sobě následující čísla liší vždy právě jen na 1 řádovém místě (Hammingova vzdálenost dvou po sobě jdoucích čísel se vždy = **1**)
  - široké uplatnění – detekční a korekční kódy, kódování pozic, Karnaughovy mapy, aj.
- **Kód 1 z n**
  - jednotlivá čísla jsou vyjádřena jako pozice číslice 1 v kódovém slově
  - vysoká redundance, ale snadná čitelnost a možnost rychlé detekce chyby v kódu
- **Váhové kódy**
  - každému řádovému místu odpovídá jiná váha, některé kombinace váh mohou vytvořit úplné vyjádření čísel 0-9 v desítkové soustavě, např.: 5421, 84-2-1, 2421, aj.

- ## Kód F+3 (XS-3)
  - kód F+3 (také **XS-3**) je kód, kde k číslu v binárním zápise je přičteno $+3_{(10)} = +0011_{(2)}$
- ## Kód pro zobrazení znaků na 7segmentovém displeji
  - 7 samostatných světelných segmentů rozmístěných tak, aby pomocí kombinací jeho rozsvícených a zhasnutých segmentů bylo možno zobrazovat dekadické číslice a vybrané znaky abecedy.

$N_{(10)}$	BCD kód	Váhový 5421	Váhový 84-2-1	Grayův kód	F+3 kód	Kód 1 z n	Kód 7segm. displeje
0	0000	0000	0000	0000	0011	0000000001	1000000
1	0001	0001	0111	0001	0100	0000000010	1111001
2	0010	0010	0110	0011	0101	0000000100	0100100
3	0011	0011	0101	0010	0110	0000001000	0110000
4	0100	0100	0100	0110	0111	0000010000	0011001
5	0101	1000	1011	0111	1000	0000100000	0010010
6	0110	1001	1010	0101	1001	0001000000	0000010
7	0111	1010	1001	0100	1010	0010000000	1111000
8	1000	1011	1000	1100	1011	0100000000	0000000
9	1001	1100	1111	1101	1100	1000000000	0010000

- **aritmetické operace v číselných soustavách**
  - zaměříme se na dvojkovou, osmičkovou a šestnáctkovou, postup stejný pro libovolnou
  - lze samozřejmě nejprve převést obě čísla do desítkové, provést příslušnou operaci a výsledek převést zpět – obvykle ale rychlejší provést operaci přímo v dané soustavě

1. **Sčítání**

- provádí se stejně jako v desítkové soustavě – pouze kontrolujeme přenos do vyšších řádů
- příklad – sečtěte $101101_{(2)} + 11011_{(2)}$:

1. krok

$$
\begin{array}{rcccccc}
 & 1 & 0 & 1 & 1 & 0 & 1 \\
+ & 0 & 1 & 1 & 0 & 1 & 1 \\
\hline
 & & & & & 1+1 = 2_{(10)} = 10_{(2)} & 0
\end{array}
$$

přeneseme zapíšeme

2. krok

$$
\begin{array}{rcccccc}
 & & & & \text{přenos} & 1 & \\
 & 1 & 0 & 1 & 1 & 0 & 1 \\
+ & 0 & 1 & 1 & 0 & 1 & 1 \\
\hline
 & & & & 1+0+1 = 2_{(10)} = 10_{(2)} & 0 & 0
\end{array}
$$

3. krok

$$
\begin{array}{rcccccc}
 & & & \text{přenos} & 1 & & \\
 & 1 & 0 & 1 & 1 & 0 & 1 \\
+ & 0 & 1 & 1 & 0 & 1 & 1 \\
\hline
 & & & 1+1+0 = 2_{(10)} = 10_{(2)} & 0 & 0 & 0
\end{array}
$$

4. krok

$$
\begin{array}{rcccccc}
 & & \text{přenos} & 1 & & & \\
 & 1 & 0 & 1 & 1 & 0 & 1 \\
+ & 0 & 1 & 1 & 0 & 1 & 1 \\
\hline
 & & 1+1+1 = 3_{(10)} = 11_{(2)} & 1 & 0 & 0 & 0
\end{array}
$$

- dokončení – sečtěte $101101_{(2)} + 11011_{(2)}$:

5. krok

přenos		1				
	1	0	1	1	0	1
+	0	1	1	0	1	1
$1+0+1 = 2_{(10)} = 10_{(2)}$		0	1	0	0	0

6. krok

přenos	1					
	1	0	1	1	0	1
+	0	1	1	0	1	1
$1+1+0 = 2_{(10)} = 10_{(2)}$	0	0	1	0	0	0

7. krok

přenos	1						
	0	1	0	1	1	0	1
+	0	0	1	1	0	1	1
	1	**0**	**0**	**1**	**0**	**0**	**0**

1. Sčítání

- takto lze sčítat čísla v libovolné polyadické číselné soustavě – je jen nutné kontrolovat přenos do vyšších řádů při překročení *Z-1*
- jiné příklady a soustavy:

dvojková soustava

	+1	+1	+1	+1	+1	+1	+1	
	0	1	1	0	1	0	1	1
+	0	0	1	1	0	1	1	1
	1	0	1	0	0	0	1	0

osmičková soustava

	+1	+1	+1	
	0	3	2	6
+	0	7	6	7
	1	3	1	5

$6+7 = 13_{(10)} = 15_{(8)}$

$1+2+6 = 9_{(10)} = 11_{(8)}$

šestnáctková soustava

	+1	+1	+1	
	0	C	A	D
+	0	3	F	6
	1	0	A	3

$13+6 = 19_{(10)} = 13_{(16)}$

$1+10+15 = 26_{(10)} = 1A_{(16)}$

2. Odečítání – 2 metody

a) odečítání stejným způsobem jako v desítkové soustavě, „vypůjčení“ z vyššího řádu

- při odečítání si můžeme vypůjčit z vyššího řádu jedničku a místo 0 si představit 2 neboť $2_{(10)} = 10_{(2)}$, nebo místo 1 si představit 3 neboť $3_{(10)} = 11_{(2)}$

příklad – odečtěte ve dvojkové soustavě $101100_{(2)} - 11111_{(2)}$:

1. krok

vypůjčení -1

	1	0	1	1	0	0
-	0	1	1	1	1	1
						1

0 < 1, místo 0 si představíme $2_{(10)} \dots 2_{(10)} - 1_{(10)} = 1$

2. krok

vypůjčení -1

	1	0	1	1	0	0
-	0	1	1	1	1	1
					0	**1**

0 < 1+1, místo 0 si představíme $2_{(10)} \dots 2_{(10)} - 2_{(10)} = 0$

3. krok

vypůjčení -1

	1	0	1	1	0	0
-	0	1	1	1	1	1
				1	**0**	**1**

1 < 1+1, místo 1 si představíme $3_{(10)} \dots 3_{(10)} - 2_{(10)} = 1$

4. krok

vypůjčení -1

	1	0	1	1	0	0
-	0	1	1	1	1	1
			1	**1**	**0**	**1**

1 < 1+1, místo 1 si představíme $3_{(10)} \dots 3_{(10)} - 2_{(10)} = 1$

2. Odečítání – 2 metody

a) dokončení příkladu – odečtěte ve dvojkové soustavě $101100_{(2)}$ - $11111_{(2)}$:

5. krok

vypůjčení	-1					
	1	0	1	1	0	0
-	0	1	1	1	1	1
0 < 1+1, místo 0 si představíme $2_{(10)}$ … $2_{(10)}$-$2_{(10)}$ = 0		**0**	**1**	**1**	**0**	**1**

6. krok

	-1					
	1	0	1	1	0	0
-	0	1	1	1	1	1
1-1 =	0	0	**1**	**1**	**0**	**1**

2. Odečítání – 2 metody

b) použití dvojkového doplňku – 4 kroky:

1. vyrovnání počtu řádových míst menšence a menšitele – doplníme před menšitele zleva tolik 0, aby byl počet řádových míst obou čísel shodný
2. vytvoříme dvojkový doplněk menšitele
3. sečteme (**nikoliv odečteme!**) menšence a dvojkový doplněk menšitele
4. pokud byl při sčítání překročen počet řádových míst (výsledek má větší počet řádových míst než původní menšenec), tato přetékající řádová místa vyškrtneme

- stejný příklad jako předchozí – odečtěte ve dvojkové soustavě $101100_{(2)}$ - $11111_{(2)}$:
1. krok – vyrovnání počtu řádových míst – doplníme jednu 0 před menšitele

$$
\begin{array}{rcccccc}
 & 1 & 0 & 1 & 1 & 0 & 0 \\
- & \color{red}{0} & 1 & 1 & 1 & 1 & 1 \\
\hline
\end{array}
$$

2. krok – vytvoříme dvojkový doplněk upraveného menšitele
$011111_{(2)}$ -> $100000_{(2)}+1_{(2)}$ = $\mathbf{100001_{(2)}}$

2. Odečítání – 2 metody

b) dokončení příkladu – odečtěte ve dvojkové soustavě $101100_{(2)}$ - $11111_{(2)}$:

3. krok – nyní sečteme menšence a dvojkový doplněk menšitele

$$
\begin{array}{ccccccc}
 & 1 & 0 & 1 & 1 & 0 & 0 \\
+ & 1 & 0 & 0 & 0 & 0 & 1 \\
\hline
1 & 0 & 0 & 1 & 1 & 0 & 1
\end{array}
$$

4. krok – zkontrolujeme počet řádových míst výsledku a škrtneme přetékající

$$
\begin{array}{ccccccc}
 & 1 & 0 & 1 & 1 & 0 & 0 \\
+ & 1 & 0 & 0 & 0 & 0 & 1 \\
\hline
\cancel{1} & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1}
\end{array}
$$

porovnejte s výsledkem první metody = $1101_{(2)}$

2. Odečítání

- další příklady a jiné soustavy:
- stejný postup i pro jiné soustavy (metoda 1), metoda 2 lze použít jen pro dvojkovou s.

dvojková soustava

	-1	-1		-1	-1	-1	
	1	1	0	1	0	0	0
-	0	1	1	0	1	1	1
	0	1	1	0	0	0	1

osmičková soustava

	-1	-1	
	5	0	1
-	3	7	7
	1	0	2

1-7 = 9-7 = vypůjčíme 1, $2_{(8)}$

0-7-1 = 8-8 = vypůjčíme 1, $0_{(8)}$

šestnáctková soustava

	-1	-1	
	1	A	5
-	0	F	E
	0	A	7

5-14 = 21-14 = vypůjčíme 1, $7_{(16)}$

10-15-1 = 26-16 = vypůjčíme 1, $A_{(16)}$

3. Násobení

- postup násobení ve dvojkové soustavě je stejný jako v desítkové – při násobení 1 jen opisujeme, při násobení 0 napíšeme řádek nul nebo jen jednu nulu a posuneme
- příklad – vynásobte ve dvojkové soustavě $101100_{(2)}$ x $1011_{(2)}$:

$$
\begin{array}{ccccccccc}
 & & & 1 & 0 & 1 & 1 & 0 & 0 \\
 & & \text{x} & & & 1 & 0 & 1 & 1 \\
\hline
 & & & 1 & 0 & 1 & 1 & 0 & 0 \\
 & & 1 & 0 & 1 & 1 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & & \\
1 & 0 & 1 & 1 & 0 & 0 & & & \\
\hline
\mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0}
\end{array}
$$

- složitější situace může nastat, pokud při konečném sčítání narazíme na přenos o více řádů

3. Násobení

- příklad – vynásobte ve dvojkové soustavě $101111_{(2)}$ x $1111_{(2)}$:

				1	0	1	1	1	1
			x			1	1	1	1
					+1	**+0**			
				1	0	1	1	1	1
			1	0	1	1	1	1	
		1	0	1	1	1	1		
	1	0	1	1	1	1			
1	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**1**

1+1 = 0 +1 přenos

1+1+1+1 = 4 = $\boxed{1}\,\boxed{0}\,\boxed{0}_{(2)}$

- protože $4_{(10)}$ je ve dvojkové soustavě $100_{(2)}$, nepřičteme přenos 1 o jednu ale o dvě pozice, přičteme tedy přenos 0 o jednu pozici vlevo a přenos 1 o dvě pozice vlevo

3. Násobení

- další příklady a jiné soustavy:

dvojková soustava

			1	0	1	0
		x		1	1	1
			1	0	1	0
		1	0	1	0	
	1	0	1	0		
1	0	0	0	1	1	0

osmičková soustava

		5	3	1
	x		4	7
	4	5	5	7
2	5	4	4	
3	2	2	1	7

šestnáctková soustava

		2	A	C
	x		B	B
	1	D	6	4
1	D	6	4	
1	F	3	A	4

4. Dělení

- obecně lze použít stejný postup jako při dělení v desítkové soustavě – to je ale například ve dvojkové soustavě těžké si představit
- proto obvykle převádíme operaci dělení na odečítání
- 2 metody dělení založené na odečítání:

a) **přímé odečítání** – kontinuálně odečítáme od dělence dělitele dokud je výsledek odečítání nezáporný:

- příklad – vydělte ve dvojkové soustavě $10010_{(2)} : 110_{(2)}$

1. krok

$$\begin{array}{rccccc} & 1 & 0 & 0 & 1 & 0 \\ - & 0 & 0 & 1 & 1 & 0 \\ \hline & & 1 & 1 & 0 & 0 \end{array}$$

2. krok

$$\begin{array}{rcccc} & 1 & 1 & 0 & 0 \\ - & 0 & 1 & 1 & 0 \\ \hline & & 1 & 1 & 0 \end{array}$$

3. krok

$$\begin{array}{rccc} & 1 & 1 & 0 \\ - & 1 & 1 & 0 \\ \hline & & & 0 \end{array}$$

3 odečty = $3_{(10)}$ = $\underline{\underline{\mathbf{11_{(2)}}}}$

4. Dělení

b) **odečítání s posunem o řády** – postupně odečítáme od dělence dělitele posunutého o ***n***-řádů:

- nejprve posuneme dělitele o ***n***-řádů směrem vlevo (doplníme nuly zprava) tak, abychom vyrovnali počet řádových míst dělence a dělitele
- pokud je výsledek odečítání kladný, provedeme odečtení a **zapíšeme 1**, pokud je výsledek záporný, neprovedeme odečtení a **napíšeme 0**
- potom provedeme posun dělitele o ***n-1***-řádů a opět zkusíme odečtení od dělence či výsledku předchozího odečítání
- tímto způsobem pokračujeme, dokud se nedostaneme na nulový posun
- výsledek sepíšeme pomocí poznamenaných nul a jedniček – **nezáporný výsledek = 1**, **záporný výsledek = 0**

4. Dělení

b) odečítání s posunem o řády

- příklad – vydělte ve dvojkové soustavě $100011_{(2)} : 111_{(2)}$

1. krok – posuneme dělitele o 3 řády (doplníme 3 nuly)

$$\begin{array}{r} 1\;0\;0\;0\;1\;1 \\ -\;1\;1\;1\;\color{red}{0}\;\color{red}{0}\;\color{red}{0} \\ \hline x \end{array} \longrightarrow \text{dělitel} > \text{dělenec} = \color{red}{\mathbf{0}}$$

2. krok – posuneme dělitele o 2 řády (doplníme 2 nuly)

$$\begin{array}{r} 1\;0\;0\;0\;1\;1 \\ -\;\;\;1\;1\;1\;\color{red}{0}\;\color{red}{0} \\ \hline 1\;1\;1 \end{array} \longrightarrow \text{dělitel} < \text{dělenec} = \color{red}{\mathbf{1}}$$

3. krok – posuneme dělitele o 1 řád (doplníme 1 nulu)

$$\begin{array}{r} 1\;1\;1 \\ -\;1\;1\;1\;\color{red}{0} \\ \hline x \end{array} \longrightarrow \text{dělitel} > \text{dělenec} = \color{red}{\mathbf{0}}$$

4. krok – bez posunu dělitele (nedoplňujeme nuly)

$$\begin{array}{r} 1\;1\;1 \\ -\;1\;1\;1 \\ \hline 0 \end{array} \longrightarrow \text{dělitel} = \text{dělenec} = \color{red}{\mathbf{1}}$$

výsledek zapíšeme odshora dolů = $\underline{\underline{\mathbf{101}_{(2)}}}$