

# Kompiuterių architektūra

Irus Grinis  
Rudens semestras  
2022/23 m.m.

# Laboratorinių darbų ir kurso vertinimas (iš sando)

- **Vertinimo būdas:**

- Egzamino rezultatai (60%) ir laboratorinių darbų užduotys (**40%**);

- **Atsiskaitymo reikalavimai:**

- Egzaminas raštu. Egzaminą gali laikyti tik tie studentai, kurių laboratorinių darbų užduočių atlikimas sumoje įvertintas ne mažiau kaip **25 %** nuo pilno užduočių atlikimo įvertinimo.

# Laboratoriniai darbai (mūsų grupei)

- **Laboratorinių darbų (jų bus **3**) struktūra:**
  - 5 – 7 užsiėmimai sudaro vieną laboratorinį darbą;
  - **Kokie būna užsiėmimai:**
    - **Minikontrolinis/testas**, kurio tikslas patikrinti per praėjusius užsiėmimus įgytas žinias: trukmė yra 25-45 minutės. Testai rašomi pagal žemiau pateiktą grafiką;
    - **Naujos medžiagos ir individualiųjų užduočių pristatymas:** 20-25 min.;
    - **Dėstytojo prižiūrimos pratybos**, skirtos naujai medžiagai įsisavinti: 20-25 min.;
    - **Individualiųjų užduočių atsiskaitymas**, kontrolinio darbo rezultatų apžvalga, konsultavimas užduočių atlikimo klausimais: likęs laikas
- **Lankomumo reikalavimai: privaloma lankyti užsiėmimus, per kuriuos rašomas kontrolinis**

# Laboratoriniai darbai (mūsų grupei)

- Laboratorinių darbų aukščiausi vertinimai:
  - 1 laboratorinis darbas: **1 balas;**
  - 2 laboratorinis darbas: **1,5 balo;**
  - 3 laboratorinis darbas: **1,5 balo;**
- Laboratorinių darbų galutinių vertinimų grafikas:
  - 1 laboratorinis darbas: **~2022 10 10 – 16, priklausomai nuo pogrupio;**
  - 2 laboratorinis darbas: **2022 11 14 – 20, priklausomai nuo pogrupio;**
  - 3 laboratorinis darbas: **iki paskutiniojo Jūsų pogrupio užsiėmimo.**

# **Testų grafikas ir aukščiausi vertinimai (mūsų grupei)**

## **Laboratorinių darbų testų rašymo grafikas (datos gali keistis +/-1 savaitė):**

### **–1-o laboratorinio darbo:**

- **2022 09 15 – 16 (0,25 balo) – priklausomai nuo pogrupio;**
- **2022 09 29 – 30 (0,25 balo) – priklausomai nuo pogrupio;**

### **–2-o laboratorinio darbo:**

- **2022 10 27 – 28 (0,5 balo) – priklausomai nuo pogrupio;**

### **–3-o laboratorinio darbo:**

- **2022 12 01 – 02 (0,5 balo) – priklausomai nuo pogrupio**

# 1-s laboratorinis darbas, 1 užsiėmimas

- Tematika:
  - Pozicinės skaičiavimo sistemos (sveikųjų skaičių atvejis), vertimas iš vienos sistemos į kitą, aritmetiniai veiksmai:
    - Dvejetainė skaičiavimo sistema, aritmetiniai veiksmai joje;
    - Aštuntainė skaičiavimo sistema, aritmetiniai veiksmai joje;
    - 16-tainė skaičiavimo sistema, aritmetiniai veiksmai joje;
    - Skaičių su ženklu ir be ženklo vaizdavimas.
  - Bitai, baitai, žodžiai, įvairios operacijos

# Nuoroda

*[www.johnloomis.org/ece314/notes/carch/node3.html](http://www.johnloomis.org/ece314/notes/carch/node3.html)*

# Dvejetainė sistema

- 0, 1, 10, 11, 100, 101, 110, 111, 1000,...
- Vertimas iš 10-tainės:
  - Pvz.:
    - $259_{10} \rightarrow \text{??????}_2$



# Dvejetainės išraiškos gavimas

Algoritmas natūraliajam skaičiui  $n$ :

- $j \leftarrow 1;$
- ***kol*** ( $n > 0$ )
  - $\langle j - \text{tasis skaitmuo nuo galo} \rangle \leftarrow n \bmod 2;$
  - $n \leftarrow n \text{ div } 2;$
  - $j \leftarrow j + 1;$

# Veikimo pavyzdys

- 1 žingsnis (duoda pirmą nuo galo skaitmenį)
  - $259 \bmod 2 = 1$ ;
  - $259 \div 2 = 129$ ;
- 2 žingsnis (duoda 2-ą nuo galo skaitmenį)
  - $129 \bmod 2 = 1$ ;
  - $129 \div 2 = 64$ ;
- 3 žingsnis
  - $64 \bmod 2 = 0$ ;
  - $64 \div 2 = 32$ ;
- 4 žingsnis
  - $32 \bmod 2 = 0$ ;
  - $32 \div 2 = 16$ ;
  - 
  -

# Veikimo pavyzdys

- 5 žingsnis
  - $16 \bmod 2 = 0$ ;
  - $16 \operatorname{div} 2 = 8$ ;
- 6 žingsnis
  - $8 \bmod 2 = 0$ ;
  - $8 \operatorname{div} 2 = 4$ ;
- 7 žingsnis
  - $4 \bmod 2 = 0$ ;
  - $4 \operatorname{div} 2 = 2$ ;
- 8 žingsnis
  - $2 \bmod 2 = 0$ ;
  - $2 \operatorname{div} 2 = 1$ ;
- 9 žingsnis (**paskutinis**)
  - $1 \bmod 2 = 1$ ;
  - $1 \operatorname{div} 2 = 0$ ;
  - 
  -

# Išvada

$$259_{10} \rightarrow 100000011_2$$

# Iš 2-tainės į 10-tainę

- Idėja: sumuojame **atitinkamus** dvejetainio laipsnius:

**7**6**5**432**1**0

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

$10111011_2 \rightarrow$

$2^7 + 2^5 + 2^4 + 2^3 + 2^1 + 2^0$

$\rightarrow 128 + 32 + 16 + 8 + 2 + 1 \rightarrow \mathbf{187}$

# Aritmetiniai veiksmai su dvejetaisiais skaičiais: sudėtis

- Pvz.:

$$\begin{array}{r} 10001101 \\ + \quad 111011 \\ \hline 11001000 \end{array}$$

- Taisyklės:

$$1 + 1 = 10$$

$$10 + 1 = 11$$

# 8-tainė

- 0,1,2,3,4,5,6,7,10,11,12,13,14,15,16,17,20, ...

Vertimo iš dešimtainės algoritmas – tas pats, tik dalinti ir ieškoti liekanos reikia iš 8;

- Analogiškai – vertimas į dešimtainę;
- Įdomesnis vertimas į/iš dvejetainę(-ės). Tam labai praverčia tokia lentelė:

– 0 ↔ 000

– 1 ↔ 001

– 2 ↔ 010

– 3 ↔ 011

– ...

# 8-tainė (tęsiny)

Vertimas iš dvejetainės į aštuntainę:

Pvz.:

1110111010110 → <suskaidome po tris skaitmenis nuo galo> →

**1 110 111 010 110 →**

**|   |   |   |   |**

**1   6   7   2   6 → 16726<sub>8</sub>**

- **Analogiškai verčiame iš aštuntainės į dvejetainę: kiekvieną skaitmenį keičiame dvejetainiu „kodu“.**



# 16-tainė sistema

- 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,  
10,11,12,13,14,15,16,17,18,19,1A,1B,...

- **Dėmesio:**

Šiame *institute* labai patartina išmokti **ATMINTINAI** 16-tainių skaitmenų dvejetainius „kodus“, nes tai yra „klasikinių programerių kultūrinių įgūdžių dalis“

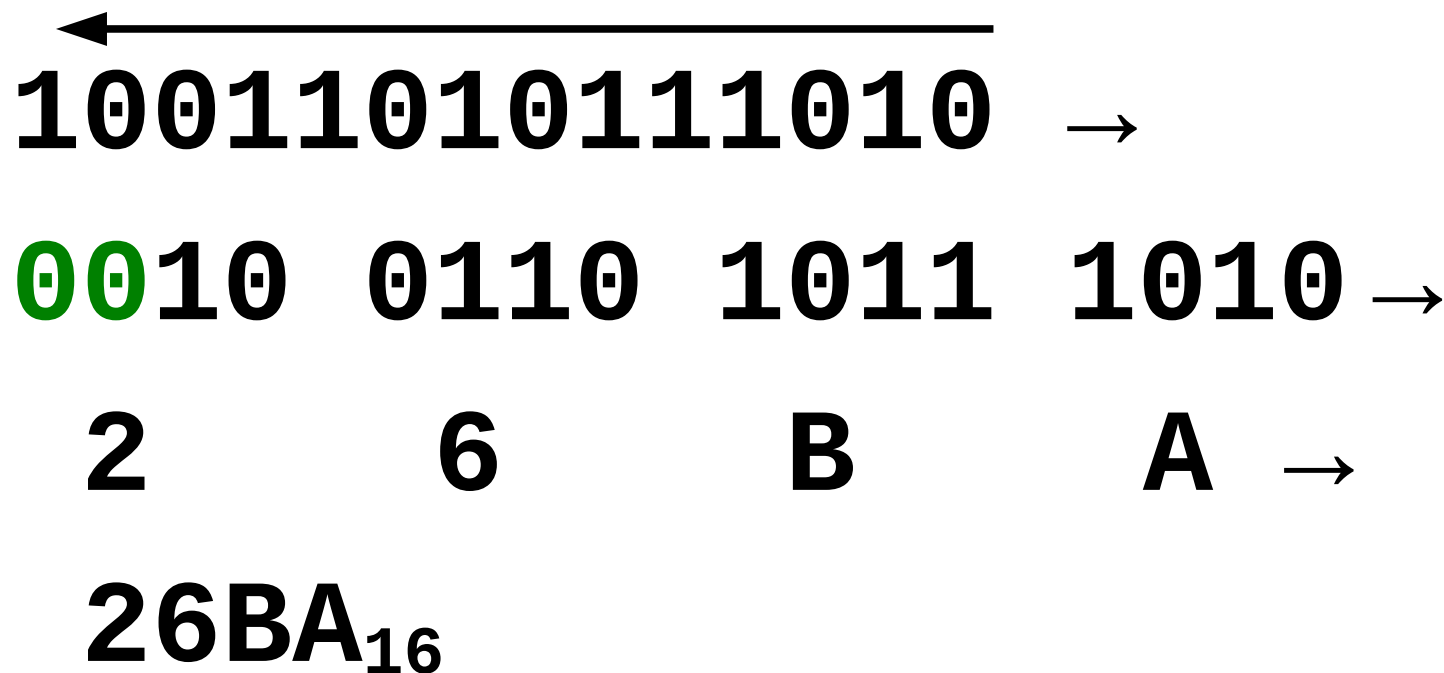
- Toliau paprastame tekste visur 16-tainius skaičius rašysime su priesaga „h“, pvz., 123Fh

# 16-tainės sistemos skaitmenų dvejetainė išraiška

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

# Vertimas iš dvejetainės į 16-tainę

- Pvz.:

  
**10011010111010** →  
**0010 0110 1011 1010** →  
**2 6 B A** →  
**26BA<sub>16</sub>**

# Sudētis 16-tainēje sistemoje

- Pvz.:

$$\begin{array}{r} 3ABC09 \\ + \quad F08D1 \\ \hline 49C4DA \end{array}$$

- Taisyklė :)  $F + 1 = 10$

# Skaičiai su ženklu

- Yra trys pagrindiniai būdai, kurie naudojami skaičiams su ženklu pavaizduoti:
  - poslinkio būdas;
  - [ženklas] [modulis];
  - Dvejeto papildinys;
  - ...

Pastaba. Turime omenyje, kad dirbame su fiksuotu dvejetainių skaitmenų skaičiumi.

# Poslinkio būdas

- Per vidurį intervalo nustatome nulį, pvz., jeigu reikėtų vaizduoti neigiamus skaičius „pusbaityje“ (4 bitai), tai galima būtų imti nulį „1000“:

0000 → -7

1001 → +1

0001 → -6

1000 → 0

1010 → +2

...

0111 → -1

1111 → +7

# [ženklas][modulis]

- Vyresnį bitą skelbiame „ženklu“ („1“ → „-“, o „0“ → „+“), o likusieji bitai rodo modulį. Pvz., baitas

10001010 → -10,

o baitas

00001010 → +10

Pastaba: yra du nuliai :)

# Dvejeto papildinys

- Jis patogus, nes paprastai realizuojamas pačiame procesoriuje.
- Kaip jis sudaromas?
  - 1) Kiekvieną bitą keičiame komplementariu ( $0 \leftrightarrow 1$ );
  - 2) Pridedame 1.



# Dvejeto papildinys

Tarkime, dirbame su 8 bitais. Raskime skaičiui  $123_{10} \rightarrow 01111011$  2' papildymą:

$01111011 \rightarrow$  (komplimentuojame)  $10000100 \rightarrow$   
(pridedame 1)  $10000101 \rightarrow 133_{10}$

Gautas skaičius „skelbiamas“ -123.

# Dvejeto papildinys

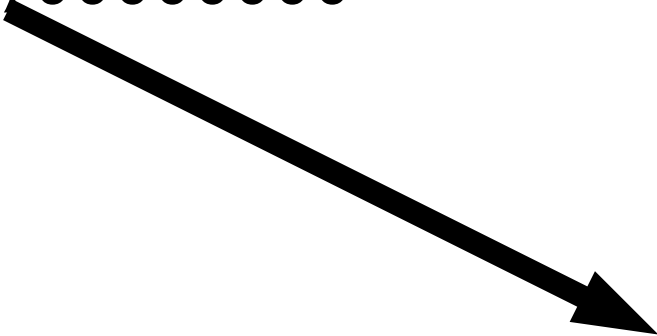
Sudėkime dabar abu skaičius:

01111011

10000101

-----

(1)00000000



Jau išėjome iš 8 bitų.

# Pratimas

Tarkime, kad dirbame su pusbaičiais. Raskite kiekvieno pusbaičio papildomą kodą:

0000, 0001, ..., 1111

Bitai, baitai, žodžiai

# Bitai ir baitai

- Kompiuterio operatyviają atmintį galima įsivaizduoti kaip bloką, kurie sudaryti iš vienodo bitų skaičiaus, masyvą
- Mažiausiai toks blokas – baitas (8 bitai)
  - Beženkliai skaičiai sudaro diapazoną  $[0,255]$  →  $[00,FF]$  (16-tainėje)
- Mes nagrinėsime šiandieninių personalinių kompiuterių senelį, IBM PC (toliau – tiesiog PK), su jame tada plačiai naudota OS MS DOS :)

# ASCII kodai

- Simboliai klasikiname PC talpinami į baitus. Labai patartina žinoti kai kuriuos ASCII kodus (16-tainėje):
- 0D 0A → '\n'
- 20 → tarpas
- '0',..., '9' → 30,...,39
- 'A',..., 'Z' → 41,..., 5A
- 'a',..., 'z' → 61, ..., 7A

# Žodžiai (word)

- Tai tiesiog du baitai.
- Jeigu nagrinėsime skaičius be ženklo, tai turėsime diapazoną  $[0, 65535] \rightarrow [0000, \text{FFFF}]$  (16-tainėje).
- Intel architektūroje atmintyje baitai eina nuo jaunesnio prie vyresnio (little-endian). Ką tai reiškia? Jeigu turime, pvz., skaičių  $321_{10} \rightarrow 141_{16}$ , tai pradžioje eina baitas  $41_{16}$ , po to  $01_{16}$ .

# Perėjimas nuo baido iki žodžio ir ženklas

- CPU turi specialią operaciją, kurios leidžia pereiti nuo baido iki žodžio neprarandant ženklo. Toks perėjimas vadinamas ženklo išplėtimu (**sign extension**):

Pvz.:

1) 00101100 → 00000000000101100;

2) 10101100 → 1111111110101100



# Loginės operacijos su baitais ir žodžiais

- Mūsų CPU (8086) gali atlikti įvairias logines operacijas su baitais ir žodžiais:
  - AND
  - OR
  - NOT
  - XOR

Visos jos atliekamos su kiekvienu bitu.

# Loginės operacijos su baitais ir žodžiais

- Pavyzdys (baitams):

10001100 AND 11101010 → 10001000

10001100 OR 11101010 → 11101110

10001100 XOR 11101010 → 01100110

NOT 10001100 → 01110011

# Postūmiai ir cikliniai postūmiai

- Pavyzdys baitams:

01011010 **SHL** 1 (postūmis kairėn per vieną bitą) → 10110100 (tas pats kaip padauginti iš 2);

01011010 **SHR** 1 (postūmis dešinėn per vieną bitą) → 00101101 (tas pats kaip padalinti iš 2);

11011010 **ROL** 1 (ciklinis postūmis kairėn per vieną bitą) → 10110101

01011011 **ROR** 1 (ciklinis postūmis dešinėn per vieną bitą) → 10101101

# Per kontrolinį Nr. 1

- Gebėti versti skaičius iš vienos sk. sistemos į kitą (ir puikiai žinoti vertimo algoritmą);
- Atlikti aritmetines operacijas 2-tainėje ir 16-tainėje sistemoje;
- Mokėti paskaičiuoti dvejetainį papildomą kodą baitams ir žodžiams, konvertuoti baitą į žodį išplečiant ženklą;
- Atlikti logines ir postūmių operacijas su baitais ir žodžiais

Baitai ir žodžiai užduotyse gali būti užrašyti 10-tainėje, 2-tainėje ir 16-tainėje sistemose