

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЖИТОМИРСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

АРХІТЕКТУРА КОМП'ЮТЕРА

Частина 1

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
ДЛЯ ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ**

Житомир
2018

УДК 004.22
А-63

*Рекомендовано до друку науково-методичною радою
Житомирського державного технологічного університету
(протокол № 5 від 30 січня 2018 року)*

Рецензенти:

Н. М. Лобанчикова – кандидат технічних наук, доцент

І. І. Сугоняк – кандидат технічних наук, доцент

А-63 Архітектура комп'ютера : методичні рекомендації для виконання лабораторних робіт. Ч. 1 / підг. А. А. Єфіменко Є. М. Байлюк, О. А. Покотило. – Житомир : ЖДТУ, 2018. – 58 с.

Методичні рекомендації містять теоретичний матеріал, приклади та вказівки для виконання лабораторних робіт, пов'язаних із вивченням питань представлення даних в пам'яті комп'ютера.

Методичні рекомендації призначені для студентів, які навчаються за спеціальностями 121 „Інженерія програмного забезпечення”, 123 „Комп'ютерна інженерія”, 125 „Кібербезпека” галузі знань 12 „Інформаційні технології”.

Підготували: кандидат технічних наук **А. А. Єфіменко, Є. М. Байлюк, О. А. Покотило.**

УДК 004.22

ЗМІСТ

ВСТУП	4
Лабораторна робота № 1. СИСТЕМИ ЧИСЛЕННЯ.....	6
Лабораторна робота № 2. ВНУТРІШНЄ ПРЕДСТАВЛЕННЯ ЦІЛОЧИСЕЛЬНИХ ДАНИХ	17
Лабораторна робота № 3. ВНУТРІШНЄ ПРЕДСТАВЛЕННЯ ДІЙСНИХ ДАНИХ	35
СПИСОК ЛІТЕРАТУРИ.....	51

ВСТУП

Дані методичні рекомендації розроблені для студентів, які навчаються за спеціальностями 121 „Інженерія програмного забезпечення”, 123 „Комп’ютерна інженерія”, 125 „Кібербезпека” галузі знань 12 „Інформаційні технології”, для вивчення змістовного модуля „Системи числення та внутрішнє представлення чисел” навчальної дисципліни „Архітектура комп’ютера”. Зазначена дисципліна згідно з освітньою програмою та навчальними планами підготовки бакалаврів належить до нормативної частини циклу дисциплін професійної і практичної підготовки.

Основним призначенням даного видання є: поглиблення теоретичних знань, отриманих студентами під час вивчення змістового модуля; набуття практичних навичок з переведення чисел з однієї системи числення в іншу; набуття навичок з внутрішньомашинного представлення чисел.

Слід зазначити, що роботи виконуються індивідуально за варіантами. Вибір номера варіанта проводиться за списком групи.

Повне виконання лабораторної роботи передбачає виконання таких етапів: ознайомлення з теоретичними відомостями; аналіз прикладу розрахунків; аналіз індивідуального варіанта завдання; формування висновків по роботі; оформлення та захист звіту; підготовка до подальших контрольних заходів.

Звіт із лабораторної роботи оформлюється згідно з вимогами Державного стандарту України ДСТУ 3008-95 „Документація. Звіти у сфері науки і техніки. Структура і правила оформлення”, міжнародних стандартів ISO 5966:1982, ГОСТ 19.404 ЕСПД „Пояснительная записка. Требования к содержанию и оформлению” і повинен містити такі складові: номер роботи; тема роботи; мета роботи; розділ, у якому описано хід виконання роботи відповідно до пунктів завдання; висновки; додаток із лістингами налагоджень усіх комунікаційних пристроїв (за необхідності), рукописні відповіді на контрольні питання. Звіт виконується в електронному вигляді, роздруковується, доповнюється відповідями на контрольні питання і здається на кафедру для перевірки та захисту.

Під час оформлення звіту необхідно дотримуватися таких вимог: звіт оформлюється на аркушах формату А4 з рамками (перша сторінка – з кутовим штампом форми 2, решта сторінок – форми 2а

за ГОСТ 2.104-68 ЕСКД. „Основные надписи”). Штampi заповнюютьс я згідно з вимогами. Нумерація сторінок наскрізна в межах роботи. Поля для тексту: ліве – 25 мм, праве – 10 мм, верхнє – 20 мм від краю аркуша, нижнє – 10 мм від верхнього краю штампa. Текст роботи оформлюється шрифтом Times New Roman 14-го розміру, міжрядковий інтервал 1 – 1,5. Абзацний відступ 5 символів. Ілюстрації та таблиці повинні розміщуватися безпосередньо після тексту, де вони зустрічаються.

Ілюстрації (креслення, рисунки, схеми тощо) позначаються таким чином: „Рисунок № – Назва рисунка” (вирівнювання по центру, без абзацного відступу). Позначення ілюстрації виконується під ілюстрацією. Якщо кількість ілюстрацій значна і вони однотипні та невеликі за розміром, то їх рекомендується групувати в одну ілюстрацію, позначати літерами *a*), *б*) ... і підписувати їх як одну ілюстрацію.

Таблиці позначаються таким чином: „Таблиця № – Назва таблиці” (вирівнювання по лівому краю, без абзацного відступу). Їх нумерація здійснюється окремо і наскрізно у межах роботи. Позначення таблиці виконується над таблицею. Текст таблиць рекомендується оформлювати шрифтом New Roman 12-го розміру, міжрядковий інтервал 1. Якщо таблиця займає понад одну сторінку, то на другій і наступних сторінках ставиться позначення „Продовження табл. №”. Головка таблиці повторюється в усіх її частинах.

У тексті звіту можна використовувати переліки (списки в термінології текстових редакторів). Перед перерахуванням ставиться двокрапка. Перед кожною позицією переліку можна ставити тире, малу літеру українського алфавіту з дужкою, арабські цифри з дужкою. Елементи переліку пишуться з маленької літери, наприкінці ставиться крапка з комою, для останнього елемента – крапка.

Важливо, щоб контрольні питання, які наводяться після кожної лабораторної роботи, були ретельно відпрацьованими студентом самостійно з метою підготовки до контрольних заходів, таких як тестування та вирішення практичних завдань.

Лабораторна робота № 1

СИСТЕМИ ЧИСЛЕННЯ

Мета заняття: ознайомитися з системами числення, їх видами та особливостями; розглянути методи та способи переведення чисел із системи в систему; отримати практичні навички переведення чисел із однієї системи числення в іншу.

Теоретичні відомості

Поняття систем числення. Позиційні і непозиційні системи числення

Система числення – сукупність прийомів і правил найменування і позначення чисел. Найбільш звичною для нас є позиційна десятикова система числення. Як умовні знаки для запису чисел вживаються цифри.

Для того, щоб визначити число, недостатньо просто знати тип і алфавіт системи числення. Потрібно ще додати правила, які дають змогу за значеннями цифр встановити значення числа.

Розрізняють такі типи систем числення:

- непозиційні;
- позиційні;
- змішані.

У непозиційній системі числення значення цифри не залежить від позиції, в якій вона розміщена в даному числі. Прикладом непозиційної системи числення є римська. В ній роль цифр відіграють букви алфавіту: I – один, V – п'ять, X – десять, C – сто, L – п'ятдесят, D – п'ятсот, M – тисяча.

Недоліками непозиційних систем є громіздкість зображення числа, а також складність виконання арифметичних операцій.

Важливими питаннями функціонування сучасних комп'ютерних та телекомунікаційних мереж є питання, пов'язані з адресацією кінцевих вузлів та комунікаційних пристроїв, зокрема питання:

- забезпечення унікальності адрес у межах мережі;
- узгодження застосування адрес різних типів;
- конфігурування адрес мережних адаптерів/інтерфейсів та адрес мережних додатків.

Позиційна система – система числення, в якій значення кожної цифри залежить від місця в послідовності цифр у записі числа. Місця розташування символів у числі називаються розрядами і нумеруються справа наліво. Характерними рисами цих систем є наочність зображення чисел і відносна простота виконання операцій.

Основа позиційної системи – кількість одиниць молодшого розряду, що утворюють одну одиницю старшого розряду. Відповідно для запису числа в системі числення з основою N потрібно N символів. По суті основа позиційної системи – це кількість цифр, які застосовуються для запису чисел системи.

На сьогоднішній день загальноприйнятою є десяткова позиційна система числення. Її основою є число десять. Для запису числа використовують десять цифр від 0 до 9. Число у цій системі числення можна представити у вигляді степенів десяти:

$$\begin{matrix} & 2 & 1 & 0 \\ (237)_{10} = & 7 * 10^0 & + 3 * 10^1 & + 2 * 10^2. \end{matrix}$$

Система числення з основою 2 є позиційною системою і нічим не відрізняється від позиційної системи числення з будь-якою основою.

Для комп'ютера ця система числення має перевагу, адже її алфавіт має всього два символи – 0 і 1. Тобто для фіксації її символів досить мати пристрій, який може мати два суттєво різних стійких стани.

Крім двійкової системи у програмуванні вагове місце займають вісімкова і шістнадцяткова системи числення, які використовуються для скороченого запису двійкових чисел.

У вісімковій системі числення використовуються такі цифри: 0, 1, 2, 3, 4, 5, 6, 7. В шістнадцятковій системі потрібно 16 символів, серед яких – арабські цифри і п'ять букв латинського алфавіту: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Десяткові еквіваленти символів: A = 10, B = 11, C = 12, D = 13, E = 14, F = 15.

Алгоритми переведення чисел з однієї позиційної системи в іншу

Переведення чисел із десятикової системи числення в будь-яку іншу здійснюється шляхом послідовного ділення числа на основу нової системи числення. Ділення виконуємо до тих пір, поки остання частка не стане менше дільника. Отримані остачі від ділення, записані у зворотному порядку, будуть значеннями розрядів числа в новій системі числення. Остання частка дає старшу цифру числа.

Приклад 1. Перевести десятикове число 2410 у двійкову систему числення.

$$\begin{array}{r} 24 \overline{) 2} \\ 24 \overline{) 12} \overline{) 2} \\ 0 \overline{) 12} \overline{) 6} \overline{) 2} \\ \quad 0 \overline{) 6} \overline{) 3} \overline{) 2} \\ \qquad 0 \overline{) 2} \overline{) 1} \\ \qquad \qquad 1 \end{array} \quad (24)_{10} = (11000)_2$$

Приклад 2. Перевести десятикове число 68710 у шістнадцяткову систему числення.

$$\begin{array}{r} 687 \overline{) 16} \\ 672 \overline{) 42} \overline{) 16} \\ 15 \overline{) 32} \overline{) 2} \\ \quad 10 \end{array} \quad \begin{array}{r} 687 \overline{) 16} \\ 672 \overline{) 42} \overline{) 16} \\ 15(F) \overline{) 32} \overline{) 2} \\ \quad 10(A) \end{array}$$

$$(687)_{10} = (2AF)_{16}$$

Щоб перевести будь-яке число з системи числення p у десятикову систему числення необхідно розкласти це число в системі числення у многочлен і виконати обчислення у десятиковій системі числення.

Приклад 3. Перевести двійкове число 100112 у десятикову систему числення.

$$\begin{array}{ccccccc} & 4 & 3 & 2 & 1 & 0 & \\ 1 & 0 & 0 & 1 & 1 & 2 & \\ 10011_2 & = & 1 * 2^0 & + & 1 * 2^1 & + & 0 * 2^2 & + & 0 * 2^3 & + & 1 * 2^4 & = & 19_{10} \end{array}$$

Приклад 4. Перевести шістнадцяткове число 1BC16 у десятикову систему числення.

$$1BC_{16} = 12 * 16^0 + 11 * 16^1 + 1 * 16^2 = 444_{10}$$

Для переведення двійкового числа у шістнадцяткову систему числення потрібно розбити двійкове число на групи по чотири розряди (тетради), доповнюючи крайні неповні до повних нулями. Далі потрібно перевести кожну тетраду із двійкової системи числення у шістнадцяткову. Під час переведення шістнадцяткового числа у двійкову систему кожна шістнадцяткова цифра записується двійковою тетрадою.

Приклад 5. Перевести двійкове число 0101101000112 у шістнадцяткову систему числення.

$$010110100011_2 = 0101\ 1010\ 0011_2 = 5A3_{16}$$

Приклад 6. Перевести шістнадцяткове число 8AF16 у двійкову систему числення.

$$803_{16} = 1000\ 0000\ 0011_2 = 100000000011_2$$

Завдання на лабораторну роботу

1. Заповнити таблицю степенів двійки (табл. 1.1).

Таблиця 1.1

Степені двійки				
2^x	2^0	2^1	...	2^{10}
Значення			...	
2^x	2^{11}	2^{12}	...	2^{20}
Значення			...	

2. Представити задані значення відповідними степенями чисел 2, 4, 8 та 16. Оформити результат у вигляді таблиці (табл. 1.2).

Таблиця 1.2

Степені цілих чисел				
Значення	2^x	4^x	8^x	16^x
1	2^0	4^0	8^0	16^0
2	2^1	-	-	-
4	2^2	4^1	-	-
8	...	...	...	...
16				
32				
64				
128				
256				
512				
1024				
2048				
4096				
8192				
16384				
32768				
65536				
131072				
262144				
524288				
1048576				

3. Перевести два заданих числа з десяткової форми у двійкову систему числення за даними табл. 1.3 (згідно варіанту). Виконати перевірку переведення.

Таблиця 1.3

Дані для переведення у двійкову систему

№ варіанта	Число 1	Число 2
1	3+G+N	22000+G+N
2	5+G+N	11000+G+N
3	7+G+N	10000+G+N
4	9+G+N	30000+G+N
5	12+G+N	1000+G+N
6	15+G+N	17000+G+N
7	13+G+N	23000+G+N
8	20+G+N	21000+G+N
9	8+G+N	14000+G+N
10	26+G+N	16000+G+N
11	19+G+N	18000+G+N
12	2+G+N	3000+G+N
13	28+G+N	4000+G+N
14	4+G+N	6000+G+N
15	29+G+N	27000+G+N
16	27+G+N	29000+G+N
17	6+G+N	4000+G+N
18	4+G+N	28000+G+N
19	3+G+N	2000+G+N
20	18+G+N	19000+G+N
21	16+G+N	26000+G+N
22	14+G+N	8000+G+N
23	21+G+N	20000+G+N
24	23+G+N	13000+G+N
25	17+G+N	15000+G+N
26	1+G+N	12000+G+N
27	0+G+N	9000+G+N
28	10+G+N	7000+G+N
29	11+G+N	5000+G+N
30	22+G+N	3000+G+N

G – номер групи, N – номер варіанту

4. Перевести два заданих числа з десяткової форми у шістнадцяткову систему числення за даними табл. 1.4 (згідно варіанту). Виконати перевірку переведення.

Таблиця 1.4

Дані для переведення у 16-кову систему

№ варіанта	Число 1	Число 2
1	3+G+N	22000+G+N
2	5+G+N	11000+G+N
3	7+G+N	10000+G+N
4	9+G+N	30000+G+N
5	12+G+N	1000+G+N
6	15+G+N	17000+G+N
7	13+G+N	23000+G+N
8	20+G+N	21000+G+N
9	8+G+N	14000+G+N
10	26+G+N	16000+G+N
11	19+G+N	18000+G+N
12	2+G+N	3000+G+N
13	28+G+N	4000+G+N
14	4+G+N	6000+G+N
15	29+G+N	27000+G+N
16	27+G+N	29000+G+N
17	6+G+N	4000+G+N
18	4+G+N	28000+G+N
19	3+G+N	2000+G+N
20	18+G+N	19000+G+N
21	16+G+N	26000+G+N
22	14+G+N	8000+G+N
23	21+G+N	20000+G+N
24	23+G+N	13000+G+N
25	17+G+N	15000+G+N
26	1+G+N	12000+G+N
27	0+G+N	9000+G+N
28	10+G+N	7000+G+N
29	11+G+N	5000+G+N
30	22+G+N	3000+G+N

G – номер групи, N – номер варіанту

5. Перевести два заданих числа з двійкової форми у шістнадцяткову систему числення за даними табл. 1.5 (згідно варіанту). Виконати перевірку переведення.

Таблиця 1.5

Дані для переведення у 16-кову систему

№ варіанта	Число 1	Число 2
1	10010	1000110001
2	1000011	011111001111
3	011000	10000110100
4	10100	01010111101
5	010101	100101000011
6	11101	01100010100
7	111100	1000001111100
8	1000001	1111001000001
9	101011	1010100111
10	00111	10101111101
11	001111	001111001111
12	1111001	0011111111001
13	111010	1111001111010
14	01100	11101001100
15	011001	01100011001
16	01111	01100101111
17	11111	0111111111
18	111111	10101011111
19	1111111	11111101010
20	101010	111111110101
21	10101	1010110101
22	1010101	1000111111
23	01010	011111010001
24	0101010	100010111110
25	0111001	01111101110
26	01110	0111000111
27	011111	0011110001
28	0111111	01100101110
29	0111110	111100101010
30	10001	00111110001

6. Перевести два заданих числа з шістнадцяткової форми у двійкову систему числення за даними табл. 1.6 (згідно варіанту).

Таблиця 1.6

Дані для переведення у двійкову систему

№ варіанта	Число 1	Число 2
1	1483	E0330BD25
2	785A7	8213E147EF
3	89523	3962F1483
4	EF489	785A789523
5	C321D	EF489785A7
6	47DD	14763C321D
7	5A6A	47DD86452
8	3962F	34FE147EF
9	86452	5A6A47DD
10	147EF	3962FC321D
11	BAC43	8645289523
12	14763	951CCBD25
13	951CC	3641B5A6A
14	34FE	147EF3641B
15	3641B	34FEBD25
16	BD25	F4053641B
17	D645A	987C6A5698
18	F698	CE063641B
19	4E62	8213E987D3
20	987C6	525A5CC68A
21	CC68A	4E62F698
22	525A5	F698D645A
23	987D3	3641BB0631
24	8213E	987D3A5698
25	A5698	D645AE0330
26	B0631	E0330987D3
27	CE06	A5698D096
28	F405	6D096CE06
29	6D096	CC68AB0631
30	E0330	F698F405

7. Заповнити таблицю відповідності кількості біт двійкового числа діапазону можливих значень, утворених з них (табл. 1.7).

Таблиця 1.7

Таблиця діапазонів значень	
Кількість біт	Діапазон значень (max і min)
1(x)	0...1
2(xx)	0...3
...	...
...	...
9(xxxxxxxx)	0...511
10(xxxxxxxxxx)	0...1023

8. Перевести два заданих числа з двійкової форми у десяткову систему числення за допомогою віднімання за даними табл. 1.8 (згідно варіанту). Для здійснення переведення від максимально можливо-го значення, що відповідає заданій кількості біт, віднімаємо 2 в тому степені, розряд якого дорівнює нулю.

Таблиця 1.8

Дані для переведення у десяткову систему		
№ варіанта	Число 1	Число 2
1	10011	1111110001
2	1110011	01111110011
3	011010	1001111011
4	11100	0111011110
5	011101	1111010011
6	11101	0111111010
7	111101	1001011111
8	1110111	1111111000
9	101011	1011100111
10	00111	1010111110
11	001111	0011101111
12	1111001	0011111111
13	111010	1111001111
14	01110	1111100110
15	011001	0110001111
16	01111	0110010111
17	11111	0111111111
18	111111	1010101111
19	1111111	1111110101
20	101011	1111111101

Дані для переведення у десяткову систему

№ варіанта	Число 1	Число 2
21	10101	1010111101
22	1011101	1000111111
23	01110	111110100
24	0111011	100011111
25	1111001	111110111
26	01110	111000111
27	0111011	111100001
28	01101111	11011011
29	01111010	111100101
30	11001	111110001

Контрольні питання

1. Що таке система числення?
2. Які типи систем числення ви знаєте?
3. Що таке основа позиційної системи числення?
4. Які недоліки має непозиційна система?
5. Наведіть приклад непозиційної системи.
6. Які позиційні системи вам відомі? Які з них найчастіше використовуються?
7. Скільки символів і які використовуються в шістнадцятковій системі?
8. Алгоритм переведення з десяткової системи числення в іншу та навпаки.
9. Алгоритм переведення з двійкової системи числення в іншу та навпаки.
10. Алгоритм переведення з шістнадцяткової системи числення в іншу та навпаки.

Лабораторна робота № 2

ВНУТРІШНЄ ПРЕДСТАВЛЕННЯ ЦІЛОЧИ- СЕЛЬНИХ ДАНИХ

Мета заняття: ознайомитися з числовими типами даних, їх видами та особливостями; розглянути алгоритм внутрішнього (машинного) представлення чисел у відповідності з діапазоном в знакових і беззнакових форматах типів ShortInt (signed char), Byte (unsigned char), Integer (int), Word (unsigned int); написати програму на мові Асемблер для опису чисел, перевірити правильність виконаних розрахунків.

Теоретичні відомості

Поняття типу даних. Числові типи даних. Цілочисельні дані

Тип даних – характеристика, яку явно чи неявно надано об'єкту (змінній, функції, полю запису, константі, масиву тощо). Тип даних визначає множину припустимих значень, формат їхнього збереження, розмір виділеної пам'яті та набір операцій, які можна робити над даними.

Машинні типи даних. У всіх комп'ютерах, заснованих на цифровій електроніці, інформація на найнижчому рівні представляється у вигляді бітів (із значенням 0 або 1). Найменша адресована одиниця інформації називається байт (зазвичай як октет, який містить 8 бітів). Одиниця інформації, яка оброблюється інструкціями машинного коду, називається словом (станом на 2006 рік, зазвичай по 32 або 64 біти). Через наявність доповнюючого коду, машині не потрібно розрізняти знакове та беззнакове числа для більшості випадків.

Існує спеціальний набір арифметичних інструкцій, які використовують різні представлення бітів у слова, для операцій з рухомою крапкою.

Прості типи даних. Мови програмування представляють деякі прості типи даних (або примітивні), як базові блоки для програм та спеціалізованіших складних типів даних. Зазвичай прості типи даних включають числові (кілька цілих та дійсних типів), логічний (булевий), символьний та байтовий.

Існують такі числові типи даних:

- цілі та дійсні;
- беззнакові та знакові.

Цілі числа (англ. integer) не можуть містити у собі дріб. Для від'ємного числа треба ставити знак мінус (-) перед значенням (числом). Не можна використовувати кому у введенні такого числа, бо інакше буде викликана синтаксична помилка. Приклади цілих чисел:

42
10000
-233000
-100

Дійсні числа – це числа, які можуть містити у собі як цілі, так і дробові значення з точкою відокремлення від цілої частини. Для від'ємного числа треба ставити знак мінус (-) перед значенням (числом). Приклади дійсних чисел:

20.0005
99.9
-5000.12
-9999.9991

Кожний числовий тип даних має мінімальне та максимальне значення, яке називають діапазон значень. Важливо знати діапазон значень, особливо, коли працюєш з «маленькими» типами даних, оскільки у них можна зберігати лише значення у вузькому діапазоні. Спроба внести число, більше за доступний діапазон може призвести до помилок періоду компіляції/виконання, або до неправильних підрахунків (через відкидання) залежно від мови програмування, яка використовується.

Базові цілочислені типи даних для мови C та їх діапазони допустимих значень наведені в таблиці 2.1.

Таблиця 2.1

Цілочисельні типи даних		
Тип	Діапазон значень	Потрібна пам'ять
Byte	0 ...255	1 байт
Shortint	-128 ... 127	1 байт
Word	0 ... 65535	2 байти
Integer	-32768 ... 32767	2 байти
Longint	-2147483648 ... 2147483647	4 байти

Логічний тип даних – це тип даних, об'єкти якого можуть приймати одне з двох значень: істина (англ. true) та хибність (англ. false).

Перелічуваний тип, перелік (англ. enumeration type) – тип даних, що описується шляхом перелічення всіх можливих значень (кожне з яких позначається власним ідентифікатором), які можуть приймати об'єкти даного типу. Приклад (Pascal):

type Cardsuit = (clubs, diamonds, hearts, spades);

Символьний тип даних – це тип даних, що описує літери та інші знаки, використовувані на письмі. В залежності від мови програмування та конкретної реалізації, може займати 1 чи 2 байти, рідше 4. Однобайтний символьний тип може використовуватись для представлення символів з набору ASCII та інших восьмирозрядних кодувань, тоді як для представлення символів з набору Unicode потрібно щонайменше 2 байти.

До **байтових типів даних** відносяться:

- байт або byte(8 біт);
- слово або word(16 біт);
- подвійне слово або dword(32 біти);
- зчетверене слово або qword(64 біти);
- 10 байт ten byte(80 біт) і т.д.

Бітом називають двійкову цифру, одиничне значення 0 або 1. Групу з 8-ми бітів називають байтом. Байт заслужив своє власне ім'я з декількох причин. Елементарний елемент пам'яті (комірка) має довжину 8 бітів.

Упродовж кожного звертання до пам'яті процесор опрацьовує рівно 8 бітів інформації. Байт – це найменша одиниця інформації, з якою можна маніпулювати безпосередньо. Крім того, байт використовують для зображення одного символу. За допомогою одного байта можна визначити 256 (2⁸) окремих символів (межі для цілих чисел: -128 ... 127).

Для розміщення певного значення в одному байті пам'яті асемблер має в своєму розпорядженні спеціальний механізм визначення байта (define byte) – псевдокоманду (директиву) DB формату.

Отже, псевдокоманда DB:

- формує однобайтову константу – число;
- формує у пам'яті заданий рядок символів (літерал);
- просто резервує 1 байт пам'яті (за умови наявності).

Розмір **слова** становить 16 бітів. Цей розмір визначено каналами передачі даних у процесорі. Над числами до 16-ти бітів операції можуть здійснюватися однією командою.

Псевдокоманда (директива) DW визначає слова (define word).

Залишився ще один тип даних, який постійно використовують у програмах мовою асемблера. Це – **подвійне слово**, значення у 32 біти довжиною. У програмах використовують подвійні слова для зберігання адрес і дуже великих чисел. Щоб визначити область, яка містить значення подвійного слова, використовують псевдокоманду (директиву) DD (define doubleword). Ця псевдокоманда генерує поле розміром у 4 байти. Так само, як у випадку з DW, асемблер розміщує в пам'яті молодший байт подвійного слова нижче, а старший – вище. У такому ж порядку зберігаються середні два байти.

Подібно визначається пам'ять для **зчетвереного слова** (DQ) та поля у 10 байтів (DT).

Складні типи даних – це типи, які складаються з елементів, що відносяться до простих типів. До складних типів даних відносяться: масиви; множини; рядки; записи; файли; динамічні змінні; вказівники; лінійні списки (стеки, черги); нелінійні списки (двійкові дерева, несиметричні дерева, тексти, графи); процедурний тип; об'єкти.

Основні типи даних в Асемблері наведено в таблиці 2.2.

Таблиця 2.2

Основні типи даних в Асемблері		
Тип	Директива	Кількість байт
байт (byte)	DB	1
слово (word)	DW	2
подвійне слово (dword)	DD	4
зчетверене слово (qword)	DQ	8
10 байт (tbyte)	DT	10

Алгоритм внутрішнього представлення цілочисельних даних

Внутрішнє представлення цілочисельних даних відбувається в два кроки. Спочатку задане число переводиться з десяткової системи числення в двійкову.

Для того, щоб змінити знак числа, потрібно інвертувати всі його біти і додати до нього одиницю – отримаємо представлення ВІД’ЄМНОГО числа в додатковому коді. Для цілих типів із знаком, під знак відводиться старший біт, причому для додатніх чисел він дорівнює 0, а для від’ємних – 1.

Потім отриману двійкову послідовність необхідно перевести в шістнадцяткову систему числення.

Приклад 1. Для заданих чисел 59 та 2254 виконати перетворення у внутрішній(машинний) формат подання чисел в пам’яті комп’ютера за умови, що числа є беззнаковими. Для збереження чисел використовується 8 та 16 бітів.

Розв’язання. Використовуючи відомі правила переводу, спочатку переведемо задані числа в двійкову систему числення, а потім дамо їх внутрішнє представлення. Результати зведемо в таблицю 2.3.

59	2				
58	29	2			
1	28	14	2		
	1	14	7	2	
		0	6	3	2
			1	2	1
				1	

59d → 0011 1011b (BYTE)

2254	2										
2254	1127	2									

0	1126	563	2								
	1	562	281	2							
		1	280	140	2						
			1	140	70	2					
				0	70	35	2				
					0	34	17	2			
						1	16	8	2		
							1	8	4	2	
								0	4	2	2
									0	2	1
										0	

2254d → 0000 1000 1100 1110b (WORD)

Таблиця 2.3

Машинне представлення заданих чисел				
Dec	Byte(8 біт)		Word(16 біт)	
	Bin	Hex	Bin	Hex
59	0011.1011	3B	0000.0000.0011.1011	003B
2254			0000.1000.1100.1110	08CE

Приклад 2. Для заданих чисел -21 та -2216 виконати перетворення у внутрішній (машинний) формат подання чисел в пам'яті комп'ютера за умови, що числа є знаковими. Для збереження чисел використовується 8 та 16 бітів.

Розв'язання. Використовуючи відомі правила перетворення, спочатку переведемо задані числа в двійкову систему числення за допомогою додаткового коду, а потім дамо їх внутрішнє представлення. Результати зведемо в таблицю 2.4.

21	2			
20	10	2		
1	10	5	2	
	0	4	2	2
		1	2	1
			0	

21d → 0001 0101b (BYTE)

-21d → 1) 0001 0101b - двійковий код числа |-21|

2) 1110 1010b - інверсія

3) + 1

11101011b - додатковий код

2216	2										
2216	1108	2									
0	1108	554	2								
	0	554	277	2							
		0	276	138	2						
			1	138	69	2					
				0	68	34	2				
					1	34	17	2			
						0	16	8	2		
							1	8	4	2	
								0	4	2	2
									0	2	1
										0	

2216d → 0000 1000 1010 1000b (WORD)

-2216d → 1) 0000 1000 1010 1000b – двійковий код числа |-2216|

2) 1111 0111 0101 0111b – інверсія

3) + 1

1111 0111 0101 1000b – додатковий код

Таблиця 2.4

Машинне представлення заданих чисел

Dec	Byte		Word	
	Bin	Hex	Bin	Hex
-21	1110.1011	EB	1111.1111.1110.1011	FFEB
-2216			1111.0111.0101.1000	F758

Приклад 3. Для зданого набору байтів D2C8 визначити які числові значення може містити додатний набір байтів згідно стандарту 1251 (ANSI, WIN).

Розв’язання. Спочатку ділимо заданий набір байтів по два шістнадцяткових числа:

D2 C8

Потім обираємо відповідні символи з таблиці 2.12:

D2 → T

C8 → И

Заданий набір байтів містить символи ТИ.

Реалізація в turbo assembler 3.2

Файл lr_02.asm

```
.MODEL tiny
.DATA
    ; lr_01

    ; ==== Byte ====
    lb  db 59

    ; ==== Word ====
    lw  dw 2254

    ; ==== Shortint ====
    lsh db -21

    ; ==== Integer ====
    li  dw -21
        dw -2216

    ; ==== Longint ====
    ll  dd 2254
        dd -2216
END
```


Файл лістингу lr_02.lst

```
Turbo Assembler          Version 3.2          10/07/04
09:12:58    Page 1
lr_02.asm

1
2
3 0000                      .MODEL tiny
4 0000                      .DATA
5                          ; lr_01
6
7                          ; ==== Byte ====
8 0000  3B                  lb  db 59
9
10                         ; ==== Word ====
11 0002  003B               lw  dw 59
12 0008  08CE               dw 2254
13
14                         ; ==== Shortint
====
15 000B  EB                  lsh  db -21
16
17                         ; ==== Integer
====
18 000E  FFEB               li  dw -21
19 0010  F758               dw -2216
20
21                         ; ==== Longint
====
22 0018  000008CE           ll  dd 2254
23 0020  FFFFFFF758         dd -2216
24                          END

...
```

Завдання на лабораторну роботу

1. Для заданих чисел (табл. 2.5) виконати перетворення у внутрішній (машинний) формат подання чисел в пам'яті комп'ютера за умови, що числа є беззнаковими. Для збереження чисел використовуються 8, 16 та 32 біти.

Таблиця 2.5

Дані для перетворення беззнакових чисел у внутрішній машинний формат

№ варіанта	Число 1	Число 2
1	13+G+N	126-G-N
2	19+G+N	122-G-N
3	23+G+N	118-G-N
4	14+G+N	121-G-N
5	6+G+N	120-G-N
6	16+G+N	122-G-N
7	28+G+N	117-G-N
8	8+G+N	120-G-N
9	21+G+N	123-G-N
10	10+G+N	114-G-N
11	24+G+N	112-G-N
12	4+G+N	116-G-N
13	27+G+N	116-G-N
14	9+G+N	124-G-N
15	20+G+N	112-G-N
16	1+G+N	119-G-N
17	29+G+N	111-G-N
18	12+G+N	121-G-N
19	26+G+N	115-G-N
20	3+G+N	113-G-N
21	17+G+N	110-G-N
22	25+G+N	125-G-N
23	11+G+N	119-G-N
24	15+G+N	111-G-N
25	2+G+N	114-G-N
26	22+G+N	118-G-N
27	7+G+N	115-G-N
28	18+G+N	117-G-N
29	5+G+N	113-G-N
30	0+G+N	110-G-N

G – номер групи, N – номер варіанта

2. Для заданих чисел (табл. 2.6) виконати перетворення у внутрішній (машинний) формат подання чисел в пам'яті комп'ютера за умови, що числа є беззнаковими. Для збереження чисел використовується 16 та 32 біти.

Таблиця 2.6

Дані для перетворення беззнакових чисел у внутрішній машинний формат

№ варіанта	Число 1	Число 2
1	10214+G+N	10214-G-N
2	11106+G+N	11106-G-N
3	10054+G+N	10054-G-N
4	11001+G+N	11001-G-N
5	10549+G+N	10549-G-N
6	10653+G+N	10653-G-N
7	10517+G+N	10517-G-N
8	11048+G+N	11048-G-N
9	10652+G+N	10652-G-N
10	10557+G+N	10557-G-N
11	11005+G+N	11005-G-N
12	10812+G+N	10812-G-N
13	11032+G+N	11032-G-N
14	10146+G+N	10146-G-N
15	11010+G+N	11010-G-N
16	10930+G+N	10930-G-N
17	10522+G+N	10522-G-N
18	10023+G+N	10023-G-N
19	10475+G+N	10475-G-N
20	10933+G+N	10933-G-N
21	10736+G+N	10736-G-N
22	11357+G+N	11357-G-N
23	10732+G+N	10732-G-N
24	10925+G+N	10925-G-N
25	11320+G+N	11320-G-N
26	10369+G+N	10369-G-N
27	10698+G+N	10698-G-N
28	10547+G+N	10547-G-N
29	10053+G+N	10053-G-N
30	10150+G+N	10150-G-N

G – номер групи, N – номер варіанта

3. Для заданих чисел (табл. 2.7) виконати перетворення у внутрішній (машинний) формат подання чисел в пам'яті комп'ютера за умови, що числа є беззнаковими. Для збереження чисел використується 32 біти.

Таблиця 2.7

Дані для перетворення беззнакових чисел у внутрішній машинний формат

№ варіанта	Число 1	Число 2
1	32547+G+N	32547-G-N
2	32526+G+N	32526-G-N
3	32491+G+N	32491-G-N
4	32412+G+N	32412-G-N
5	32385+G+N	32385-G-N
6	32344+G+N	32344-G-N
7	32293+G+N	32293-G-N
8	32208+G+N	32208-G-N
9	32199+G+N	32199-G-N
10	32100+G+N	32100-G-N
11	31953+G+N	31953-G-N
12	31946+G+N	31946-G-N
13	31817+G+N	31817-G-N
14	31822+G+N	31822-G-N
15	31735+G+N	31735-G-N
16	31708+G+N	31708-G-N
17	31699+G+N	31699-G-N
18	31624+G+N	31624-G-N
19	31573+G+N	31573-G-N
20	31510+G+N	31510-G-N
21	30975+G+N	30975-G-N
22	30912+G+N	30912-G-N
23	30867+G+N	30867-G-N
24	30808+G+N	30808-G-N
25	30799+G+N	30799-G-N
26	30734+G+N	30734-G-N
27	30683+G+N	30683-G-N
28	30616+G+N	30616-G-N
29	30571+G+N	30571-G-N
30	30522+G+N	30522-G-N

G – номер групи, N – номер варіанта

4. Для заданих чисел (табл. 2.8) виконати перетворення у внутрішній (машинний) формат подання чисел в пам'яті комп'ютера за

умови, що числа є знаковими. Для збереження чисел використовуються 8, 16 та 32 біти.

Таблиця 2.8

Дані для перетворення знакових чисел у внутрішній машинний формат

№ варіанта	Число 1	Число 2
1	$\pm 13 + G + N$	$\pm 126 - G - N$
2	$\pm 19 + G + N$	$\pm 122 - G - N$
3	$\pm 23 + G + N$	$\pm 118 - G - N$
4	$\pm 14 + G + N$	$\pm 121 - G - N$
5	$\pm 6 + G + N$	$\pm 120 - G - N$
6	$\pm 16 + G + N$	$\pm 122 - G - N$
7	$\pm 28 + G + N$	$\pm 117 - G - N$
8	$\pm 8 + G + N$	$\pm 120 - G - N$
9	$\pm 21 + G + N$	$\pm 123 - G - N$
10	$\pm 10 + G + N$	$\pm 114 - G - N$
11	$\pm 24 + G + N$	$\pm 112 - G - N$
12	$\pm 4 + G + N$	$\pm 116 - G - N$
13	$\pm 27 + G + N$	$\pm 116 - G - N$
14	$\pm 9 + G + N$	$\pm 124 - G - N$
15	$\pm 20 + G + N$	$\pm 112 - G - N$
16	$\pm 1 + G + N$	$\pm 119 - G - N$
17	$\pm 29 + G + N$	$\pm 111 - G - N$
18	$\pm 12 + G + N$	$\pm 121 - G - N$
19	$\pm 26 + G + N$	$\pm 115 - G - N$
20	$\pm 3 + G + N$	$\pm 113 - G - N$
21	$\pm 17 + G + N$	$\pm 110 - G - N$
22	$\pm 25 + G + N$	$\pm 125 - G - N$
23	$\pm 11 + G + N$	$\pm 119 - G - N$
24	$\pm 15 + G + N$	$\pm 111 - G - N$
25	$\pm 2 + G + N$	$\pm 114 - G - N$
26	$\pm 22 + G + N$	$\pm 118 - G - N$
27	$\pm 7 + G + N$	$\pm 115 - G - N$
28	$\pm 18 + G + N$	$\pm 117 - G - N$
29	$\pm 5 + G + N$	$\pm 113 - G - N$
30	$\pm 0 + G + N$	$\pm 110 - G - N$

G – номер групи, N – номер варіанта

5. Для заданих чисел (табл. 2.9) виконати перетворення у внутрішній (машинний) формат подання чисел в пам'яті комп'ютера за

умови, що числа є знаковими. Для збереження чисел використовуються 16 та 32 біти.

Таблиця 2.9

Дані для перетворення знакових чисел у внутрішній машинний формат

№ варіанта	Число 1	Число 2
1	$\pm 10214 + G + N$	$\pm 10214 - G - N$
2	$\pm 11106 + G + N$	$\pm 11106 - G - N$
3	$\pm 10054 + G + N$	$\pm 10054 - G - N$
4	$\pm 11001 + G + N$	$\pm 11001 - G - N$
5	$\pm 10549 + G + N$	$\pm 10549 - G - N$
6	$\pm 10653 + G + N$	$\pm 10653 - G - N$
7	$\pm 10517 + G + N$	$\pm 10517 - G - N$
8	$\pm 11048 + G + N$	$\pm 11048 - G - N$
9	$\pm 10652 + G + N$	$\pm 10652 - G - N$
10	$\pm 10557 + G + N$	$\pm 10557 - G - N$
11	$\pm 11005 + G + N$	$\pm 11005 - G - N$
12	$\pm 10812 + G + N$	$\pm 10812 - G - N$
13	$\pm 11032 + G + N$	$\pm 11032 - G - N$
14	$\pm 10146 + G + N$	$\pm 10146 - G - N$
15	$\pm 11010 + G + N$	$\pm 11010 - G - N$
16	$\pm 10930 + G + N$	$\pm 10930 - G - N$
17	$\pm 10522 + G + N$	$\pm 10522 - G - N$
18	$\pm 10023 + G + N$	$\pm 10023 - G - N$
19	$\pm 10475 + G + N$	$\pm 10475 - G - N$
20	$\pm 10933 + G + N$	$\pm 10933 - G - N$
21	$\pm 10736 + G + N$	$\pm 10736 - G - N$
22	$\pm 11357 + G + N$	$\pm 11357 - G - N$
23	$\pm 10732 + G + N$	$\pm 10732 - G - N$
24	$\pm 10925 + G + N$	$\pm 10925 - G - N$
25	$\pm 11320 + G + N$	$\pm 11320 - G - N$
26	$\pm 10369 + G + N$	$\pm 10369 - G - N$
27	$\pm 10698 + G + N$	$\pm 10698 - G - N$
28	$\pm 10547 + G + N$	$\pm 10547 - G - N$
29	$\pm 10053 + G + N$	$\pm 10053 - G - N$
30	$\pm 10150 + G + N$	$\pm 10150 - G - N$

G – номер групи, N – номер варіанта

6. Для заданих чисел (табл. 2.10) виконати перетворення у внутрішній (машинний) формат подання чисел в пам'яті комп'ютера за

умови, що числа є знаковими. Для збереження чисел використовуються 32 біти.

Таблиця 2.10

Дані для перетворення знакових чисел у внутрішній машинний формат

№ варіанта	Число 1	Число 2
1	$\pm 32547 + G + N$	$\pm 32547 - G - N$
2	$\pm 32526 + G + N$	$\pm 32526 - G - N$
3	$\pm 32491 + G + N$	$\pm 32491 - G - N$
4	$\pm 32412 + G + N$	$\pm 32412 - G - N$
5	$\pm 32385 + G + N$	$\pm 32385 - G - N$
6	$\pm 32344 + G + N$	$\pm 32344 - G - N$
7	$\pm 32293 + G + N$	$\pm 32293 - G - N$
8	$\pm 32208 + G + N$	$\pm 32208 - G - N$
9	$\pm 32199 + G + N$	$\pm 32199 - G - N$
10	$\pm 32100 + G + N$	$\pm 32100 - G - N$
11	$\pm 31953 + G + N$	$\pm 31953 - G - N$
12	$\pm 31946 + G + N$	$\pm 31946 - G - N$
13	$\pm 31817 + G + N$	$\pm 31817 - G - N$
14	$\pm 31822 + G + N$	$\pm 31822 - G - N$
15	$\pm 31735 + G + N$	$\pm 31735 - G - N$
16	$\pm 31708 + G + N$	$\pm 31708 - G - N$
17	$\pm 31699 + G + N$	$\pm 31699 - G - N$
18	$\pm 31624 + G + N$	$\pm 31624 - G - N$
19	$\pm 31573 + G + N$	$\pm 31573 - G - N$
20	$\pm 31510 + G + N$	$\pm 31510 - G - N$
21	$\pm 30975 + G + N$	$\pm 30975 - G - N$
22	$\pm 30912 + G + N$	$\pm 30912 - G - N$
23	$\pm 30867 + G + N$	$\pm 30867 - G - N$
24	$\pm 30808 + G + N$	$\pm 30808 - G - N$
25	$\pm 30799 + G + N$	$\pm 30799 - G - N$
26	$\pm 30734 + G + N$	$\pm 30734 - G - N$
27	$\pm 30683 + G + N$	$\pm 30683 - G - N$
28	$\pm 30616 + G + N$	$\pm 30616 - G - N$
29	$\pm 30571 + G + N$	$\pm 30571 - G - N$
30	$\pm 30522 + G + N$	$\pm 30522 - G - N$

G – номер групи, N – номер варіанта

7. За даними табл. 2.11 визначити які числові значення може містити знаковий набір байтів, взятий у пам'яті комп'ютера.

Таблиця 2.11

Дані для визначення цілих числових значень, що представлені в пам'яті комп'ютера

№ варіанта	Число 1	Число 2
1	3F4A	EF489785
2	AB02	641B5A6A
3	130E	3962F148
4	2C36	62FC321D
5	4B78	987C6A56
6	E107	F698D645
7	7D95	E0330BD2
8	309B	34FE147E
9	85A0	5A5CC68A
10	C803	47DD8645
11	4A66	987D3A56
12	9B0E	951CCBD2
13	F032	6D096CE0
14	19C7	763C321D
15	7F41	6986D096
16	22AB	13E987D3
17	339E	CC68AB06
18	E12D	13E147EF
19	640F	F698F405
20	83EE	4053641B
21	D73C	68AB0631
22	A0CB	4E62F698
23	5AD1	E0330987
24	12AA	785A7895
25	C15F	45AE0330
26	EA07	34FEBD25
27	84DE	147EF364
28	A254	41BB0631
29	36BF	CE063641
30	1DC7	5A6A47DD

9. За даними табл. 2.11 визначити які числові значення може містити додатний набір байтів згідно стандарту 1251 (ANSI, WIN) (табл.2.12).

Таблиця 2.12

Таблиця символів 1251 (ANSI, WIN)

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
--	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

0.	0 ¹²	☉	☽	♥	♦	♣	♠	•	♠	○	♠	♂	♀	♪	♫	✱
1.	▶	◀	↑	!!	¶	§	—	↓	↑	↓	→	←	⌞	↔	▲	▼
2.		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3.	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4.	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5.	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6.	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7.	p	q	r	s	t	u	v	w	x	y	z	{		}	~	△
8.	ѣ	ѣ	,	ѣ	„	...	†	‡	€	%	љ	<	њ	ќ	ћ	џ
9.	ђ	‘	’	“	”	•	—	—	™	љ	>	њ	ќ	ћ	џ	џ
A.	160 ⁵	Ў	Ў	Ј	ѡ	Г	І	§	Ё	©	Є	«	¬	173 ⁶	®	Ї
B.	°	±	І	і	Г	μ	¶	·	ё	№	є	»	ј	Ѕ	ѕ	ї
C.	A	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D.	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E.	a	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F.	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Контрольні питання

1. Що таке тип даних?

2. Які існують типи даних?
3. Поняття цілого і дійсного числа. Приклади.
4. Поняття біт, байт, слово, подвійне слово, зчетверене слово.
5. Який розмір в байтах мають слово, подвійне слово, зчетверене слово?
6. Які директиви використовуються у мові програмування Асемблер?
7. Базові цілочислені типи даних для мови С та їх діапазони допустимих значень.
8. Правило формування знакових цілих чисел за допомогою додаткового коду.
9. Алгоритм внутрішнього(машинного) представлення цілочисельних беззнакових даних.
10. Алгоритм внутрішнього(машинного) представлення цілочисельних знакових даних.

Лабораторна робота № 3

ВНУТРІШНЄ ПРЕДСТАВЛЕННЯ ДІЙСНИХ ДАНИХ

Мета заняття: ознайомитися з дійсними числами; розглянути алгоритм переведення дійсних чисел з десяткової в двійкову систему числення і їх представлення в нормалізованому вигляді.

Теоретичні відомості

Поняття дійсних чисел. Їх формат і представлення

Дійсні числа – це числа, які можуть містити у собі як цілі, так і дробові значення з точкою відокремлення від цілої частини. Для від'ємного числа треба ставити знак мінус (-) перед значенням (числом). Приклади дійсних чисел:

20.0005

99.9

-5000.12

-9999.9991

Дійсні базові величини можуть бути формату dword, qword або tbyte. Внутрішнє (машинне) перетворення цих чисел досить важке.

Формат дійсного числа

S	Характеристика	Нормалізована мантиса
---	----------------	-----------------------

Біт S – знак числа.

Характеристика = Зміщення $\pm$ Порядок.

Зміщення – число, яке дорівнює половині максимально можливого, яке може поміститися в поле Характеристика. Таким чином, економляться місце і час, так як не потрібно виділяти розряд для знака порядку і робити додатковий код для від'ємних порядків.

Розмір простору, який ПК використовує для збереження Характеристики і Мантиси, встановлений стандартом IEEE – 1985 і підтримується всіма сучасними комп'ютерними архітектурами.

Для отримання внутрішнього представлення дійсного числа потрібно в будь-якому випадку перевести його в двійкову форму.

Переведення дійсних чисел з десяткової в двійкову систему числення

Для переведення дійсного десяткового числа в двійкову систему числення необхідно дане число (тільки мантису) послідовно множити на число 2 до тих пір, поки в мантисі не вийде або чистий нуль, або необхідна кількість розрядів.

При множенні отримані цілі значення не враховуються. Зате вони послідовно зараховуються в результат.

Отримуємо впорядковану послідовність цілих двійкових чисел.

Приклад 1. перевести дійсні числа в двійкову систему числення.

Розв'язання. Спочатку переведемо ціле число 105 в двійкову систему числення.

105	2					
104	52	2				
1	52	26	2			
	0	26	13	2		
		0	12	6	2	
			1	6	3	2
				0	2	1
					1	

105d → 1101001b

Тепер переведемо в двійкову систему числення правильний десятковий дріб. Для перевірки представлення в 32-х бітовому форматі нам досить 24-х біт в мантисі (зробимо із запасом – 26 біт)

0.4612. → ??? b

№ біта	Біт	Мантиса	Множник	Пояснення
--------	-----	---------	---------	-----------

		(Dec)		
	0	4612	2	Початкове число
1	0	9224		Результати множення мантиси на 2
2	1	8448		
3	1	6896		
4	1	3792		
5	0	7584		
6	1	5168		
7	1	0336		
8	0	0672		
9	0	1344		
10	0	2688		
11	0	5376		
12	1	0752		
13	0	1504		
14	0	3008		
15	0	6016		
16	1	2032		
17	0	4064		
18	0	8128		
19	1	6256		
20	1	2512		
21	0	5024		
22	1	0048		
23	0	0096		
24	0	0192		
25	0	0384		
26	0	0768		

0.4612d → 0.01110110000100010011010000b

Тепер є вся інформація для отримання двійкового представлення змішаного дробу:

105.4612d → 1101001.01110110000100010011010000b

Представлення дійсних чисел в двійковому нормалізованому вигляді

Отримавши двійкове представлення, необхідно його нормалізувати, тобто двійкове число має завжди починатися з одиниці і мати наступний вигляд:

$$\pm 1. m_2 * 2^p$$

де m_2 – двійкова мантиса числа,
 p – порядок двійкового числа.

Приклад 2. Представити двійкове число в нормалізованому вигляді:

$$\pm 1.0d \rightarrow \pm 1.0b * 2^0$$

Приклад 3. Представити десяткове число в нормалізованому вигляді:

$$\pm 0.5d \rightarrow \pm 0.1b,$$

в нормалізованому вигляді: $\pm 0.1b \rightarrow \pm 1.0b * 2^{-1}$

Приклад 4. Представити десяткове число в нормалізованому вигляді:

$$\pm 0.703125d \rightarrow \pm 0.101101b,$$

в нормалізованому вигляді: $\pm 0.101101b \rightarrow \pm 1.01101b * 2^{-1}$

Приклад 5. Представити десяткове число в нормалізованому вигляді:

$$\pm 0.05d \rightarrow \pm 0.000011(0011)b,$$

в нормалізованому вигляді: $\pm 0.000011(0011)b \rightarrow \pm 1.1(0011)b * 2^{-5}$

Приклад 6. Представити десяткове число в нормалізованому вигляді:

$$\pm 117.25d \rightarrow \pm 1110101.01b,$$

в нормалізованому вигляді: $\pm 1110101.01b \rightarrow \pm 1.11010101b * 2^6$

Машинні формати дійсних чисел

Дійсні базові величини можуть бути формату dword, qword або tbyte. Всі вони мають для збереження комірки різної довжини. Звідси і різний діапазон представлення для різних типів дійсних чисел, хоча ідея збереження майже одна і та ж.

Формат Dword

Це число зберігається в комірці довжиною 32 біти, які розподіляються наступним чином:

1 – прихований розряд



S	Характеристика			Нормалізована мантиса				
31	30	...	23	22	21	...	1	0
Біти								

На Характеристику виділяється 8 бітів (розряди 23-30). Максимально можливе шістнадцяткове число, яке можна розмістити в цих бітах, дорівнює FFh, а його половина – це Зміщення = 7Fh. Оскільки на представлення мантиси залишається всього 23 біти (розряди 0-22), а мантиса завжди є нормалізованою, то перша одиниця в інформаційні розряди мантиси не потрапляє (прихований розряд), а на мантису в результаті відводиться не 23 біти, а 24. Таке представлення дозволяє правильно відображати 7-8 десяткових цифр, а його діапазон представлення складає: $1.5 * 10^{-45} .. 3.4 * 10^{38}$.

Приклад 7. Представити десяткове число в нормалізованому вигляді і подати його в 32-бітному форматі.

Розв'язання. Представимо задане дійсне число в нормалізованому вигляді:

$$\pm 1.0d \rightarrow \pm 1.0 * 2^0b.$$

Тоді характеристика буде дорівнювати: $7F + 0 = 7F$.

Розписуємо детально склад 32-х бітів, а потім переводимо двійкове представлення в шістнадцяткове.

1 – прихований розряд

V

0	0	1	1	1	1	1	1	1	0	0	0	0	...	0	0
S	Характеристика								Нормалізована мантиса						
31	30	29	28	27	26	25	24	23	22	21	20	19	...	1	0
Біти															
3				F				8				0		0 0 0	
HEX-код															

Від'ємне число -1.0 буде таке ж саме, тільки в знаковому біті S=1:

1 – прихований розряд

V

1	0	1	1	1	1	1	1	1	0	0	0	0	...	0	0
S	Характеристика								Нормалізована мантиса						
31	30	29	28	27	26	25	24	23	22	21	20	19	...	1	0
Біти															
B				F				8				0		0 0 0	
HEX-код															

Приклад 8. Представити десяткове число в нормалізованому вигляді і подати його в 32-бітному форматі.

Розв'язання. Представимо задане дійсне число в нормалізованому вигляді:

$$\pm 0.5d \rightarrow \pm 1.0 * 2^{-1}b.$$

Тоді характеристика буде дорівнювати: $7F - 1 = 7E$.

Знову розписуємо склад 32-х бітів, а потім перекладаємо двійкове представлення в шістнадцяткове. В цьому прикладі змінився тільки порядок. В мантисі знову тільки один розряд, та й той прихований.

Додатне число 0.5:

1 – прихований розряд

V

0	0	1	1	1	1	1	1	0	0	0	0	0	...	0	0
S	Характеристика								Нормалізована мантиса						
31	30	29	28	27	26	25	24	23	22	21	20	19	...	1	0
Біти															
3				F				0				0		0 0 0	
HEX-код															

Від'ємне число -1.0 буде таке ж саме, тільки в знаковому біті S=1:

1 – прихований розряд

V

1	0	1	1	1	1	1	1	0	0	0	0	0	...	0	0
S	Характеристика								Нормалізована мантиса						
31	30	29	28	27	26	25	24	23	22	21	20	19	...	1	0
Біти															
B				F				0				0		0 0 0	
HEX-код															

Формат Qword

В цьому форматі ідея та ж сама, що і в попередньому. На все число відводиться комірка довжиною 64 біти. На характеристику виділено 11 бітів (розряди 52-62).

1 – прихований розряд

V

S	Характеристика				Нормалізована мантиса				
63	62	...	52	51	50	...	1	0	
Біти									

Максимально можливе число, яке можна розмістити в 11-ти бітах: **111 1111 1111b**. Це означає, що:

Зміщення = 011 1111 1111b → 3FFh.

Мантиса також має прихований розряд. Таке представлення дозволяє отримати наступний допустимий діапазон для даних типу double:

$$5.0 * 10^{-324} .. 1.7 * 10^{308},$$

і, відповідно, представити 15-16 десяткових цифр.

Приклад 9. Представити десяткове число в нормалізованому вигляді і подати його в 64-бітному форматі:

Розв’язання. Представимо задане дійсне число в нормалізованому вигляді:

$$\pm 0.05d \rightarrow \pm 1.1(0011) * 2^{-5}b.$$

Тоді характеристика буде дорівнювати: 7FF - 5 = 3FA.

Розпишемо наше число спочатку в двійковому вигляді:

0 011 1111 1010 1001 1001 1001 1001 1001 ... 1001

^

1

Тепер можна представити його і в шістнадцятковому вигляді: **3FA99999999999h.**

Формат Tbyte

S	Характеристика			Нормалізована мантиса				
79	78	...	64	63	62	...	1	0
Біти								

На все число відводиться 80 бітів. Цей формат є основним робочим форматом для даних сопроцесора. Тому, щоб НЕ створювати додаткових перетворень при обчисленнях, у **мантисі нема прихованого розряду**. На характеристику витрачається 15 бітів (розряди 64-78). Максимально можливе число, яке можна розмістити в 15-ти бітах: **111 1111 1111 1111b**. Це означає, що:

Зміщення = 011 1111 1111 1111b → 3FFFh.

Таке представлення дозволяє отримати наступний допустимий діапазон для даних цього типу:

$$5.0 * 10^{-324} .. 1.7 * 10^{308},$$

і, відповідно, представити 19-20 десяткових цифр.

Приклад 10. Представити десяткове число в нормалізованому вигляді і подати його в 80-бітному форматі.

Розв'язання. Представимо задане дійсне число в нормалізованому вигляді:

$$\pm 117.25d \rightarrow \pm 1.11010101 * 2^6b.$$

Тоді характеристика буде дорівнювати: $3FFF + 6 = 4005$ (при додаванні повна аналогія з десятковою системою числення).

Розпишемо наше число спочатку в двійковому вигляді:

0 100 0000 0000 0101 1110 1010 1000 0000 ... 0000b

$$117.25d \rightarrow 4005EA80000000000000h$$

1 100 0000 0000 0101 1110 1010 1000 0000 ... 0000b

$$-117.25d \rightarrow C005EA80000000000000h$$

Реалізація в turbo assembler 3.2

Файл lr_03.asm

```

title lr_03
.MODEL tiny
.DATA
;----- float (DWord)
f    dd  -105.4612
      dd   105.4612
      dd  -15.4522
      dd   15.4522
      dd   105.
      dd  -105.
      dd   15.
      dd  -15.

```

```

dd    0.4612
dd   -0.4612
dd    0.4522
dd   -0.4522
;----- double (QWord)
d    dQ  -105.4612
      dQ   105.4612
      dQ  -15.4522
      dQ   15.4522
      dQ   105.
      dQ  -105.
      dQ   15.
      dQ  -15.
      dQ   0.4612
      dQ  -0.4612
      dQ   0.4522
      dQ  -0.4522
;----- long double
t    dT  -105.4612
      dT   105.4612
      dT  -15.4522
      dT   15.4522
      dT   105.
      dT  -105.
      dT   15.
      dT  -15.
      dT   0.4612
      dT  -0.4612

```

dT 0.4522

dT -0.4522

END

; кінець asm-модуля

Файл лістингу lr_03.lst

```
Turbo Assembler      Version 4.1      28/05/01
02:00:05      Page 1
lr_03.asm

lr_03

      1      0000      .MODEL
tiny
      2      0000      .DATA
      3
сопроцессор (FPU) i8087      ;
      4      ;-----
-- float (DWord)
      5      0000      C2D2EC22      f
dd -105.4612
      6      0004      42D2EC22
dd 105.4612
      7      0008      C1773C36
dd -15.4522
      8      000C      41773C36
dd 15.4522
      9      0010      42D20000
dd 105.
     10      0014      C2D20000
dd -105.
     11      0018      41700000
dd 15.
     12      001C      C1700000
dd -15.
     13      0020      3EEC2268
dd 0.4612
     14      0024      BEEC2268
dd -0.4612
```

	15	0028	3EE786C2	
dd	0.4522			
	16	002C	BEE786C2	
dd	-0.4522			
	17			;-----
-- double	(QWord)			
	18	0030	C05A5D844D013A92	d
dQ	-105.4612			
	19	0038	405A5D844D013A92	
dQ	105.4612			
	20	0040	C02EE786C226809D	
dQ	-15.4522			
	21	0048	402EE786C226809D	
dQ	15.4522			
	22	0050	405A400000000000	
dQ	105.			
	23	0058	C05A400000000000	
dQ	-105.			
	24	0060	402E000000000000	
dQ	15.			
	25	0068	C02E000000000000	
dQ	-15.			
	26	0070	3FDD844D013A92A3	
dQ	0.4612			
	27	0078	BFDD844D013A92A3	
dQ	-0.4612			
	28	0080	3FDCF0D844D013A9	
dQ	0.4522			
	29	0088	BFDCF0D844D013A9	
dQ	-0.4522			
	30			;-----
-- long double				
	31	0090	C005D2EC226809D49518	t
dT	-105.4612			
	32	009A	4005D2EC226809D49518	
dT	105.4612			
	33	00A4	C002F73C36113404EA4A	
dT	-15.4522			
	34	00AE	4002F73C36113404EA4A	
dT	15.4522			
	35	00B8	4005D2000000000000	
dT	105.			

```

          36      00C2  C005D20000000000000000
dT  -105.      37      00CC  4002F00000000000000000
          38      00D6  C002F00000000000000000
dT   -15.      39      00E0  3FFDEC226809D495182A
dT   0.4612     40      00EA  BFFDEC226809D495182A
          41      00F4  3FFDE786C226809D4951
dT  -0.4612     42      00FE  BFFDE786C226809D4951
          43
; кінець asm-модуля
END

```

```

Turbo Assembler      Version 4.1      28/05/01
02:00:05      Page 2
Symbol Table

```

Symbol Name	Type	Value
??DATE	Text	"28/05/01"
??FILENAME	Text	"Asm_digt"
??TIME	Text	"02:00:05"
??VERSION	Number	040A
@32BIT	Text	0
@CODE	Text	DGROUP
@CODESIZE	Text	0
@CPU	Text	0101H
@CURSEG	Text	_DATA
@DATA	Text	DGROUP
@DATASIZE	Text	0
@FILENAME	Text	ASM_DIGT
@INTERFACE	Text	000H
@MODEL	Text	1
@STACK	Text	DGROUP
@WORDSIZE	Text	2
D	Qword	DGROUP:0030
F	Dword	DGROUP:0000

...

Завдання на лабораторну роботу

1. Для заданих чисел (табл.3.1) виконати перетворення у внутрішній(машинний) формат подання чисел в пам'яті комп'ютера за умови, що числа є знаковими. Для збереження чисел використовується 32 біти.

Таблиця 3.1

Дані для перетворення знакових дійсних чисел у внутрішній машинний формат

№ варіанта	Число 1	Число 2	Число 3
1	±3444,00	±0,00015	±3444,00015
2	±4123,00	±0,00298	±4123,00298
3	±3631,00	±0,00022	±3631,00022
4	±4486,00	±0,00233	±4486,00233
5	±4662,00	±0,00315	±4662,00315
6	±3925,00	±0,00198	±3925,00198
7	±4243,00	±0,00342	±4243,00342
8	±4018,00	±0,00179	±4018,00179
9	±4955,00	±0,00236	±4955,00236
10	±4166,00	±0,00323	±4166,00323
11	±3573,00	±0,00092	±3573,00092
12	±3811,00	±0,00124	±3811,00124
13	±4707,00	±0,00292	±4707,00292
14	±3726,00	±0,00077	±3726,00077
15	±3492,00	±0,00073	±3492,00073
16	±4448,00	±0,00301	±4448,00301
17	±3157,00	±0,00024	±3157,00024
18	±3098,00	±0,00061	±3098,00061
19	±4535,00	±0,00252	±4535,00252
20	±3309,00	±0,00012	±3309,00012
21	±3968,00	±0,00095	±3968,00095
22	±4322,00	±0,00183	±4322,00183
23	±4853,00	±0,00364	±4853,00364
24	±3288,00	±0,00013	±3288,00013
25	±3675,00	±0,00152	±3675,00152
26	±4206,00	±0,00271	±4206,00271

27	$\pm 3799,00$	$\pm 0,00084$	$\pm 3799,00084$
28	$\pm 4337,00$	$\pm 0,00312$	$\pm 4337,00312$
29	$\pm 3862,00$	$\pm 0,00099$	$\pm 3862,00099$
30	$\pm 4571,00$	$\pm 0,00268$	$\pm 4571,00268$

2. Для заданих чисел (табл.3.1) виконати перетворення у внутрішній(машинний) формат подання чисел в пам'яті комп'ютера за умови, що числа є знаковими. Для збереження чисел використовується 64 біти.

3. Для заданих чисел (табл.3.1) виконати перетворення у внутрішній(машинний) формат подання чисел в пам'яті комп'ютера за умови, що числа є знаковими. Для збереження чисел використовується 80 біт.

4. Для отриманих результатів в завданні 1-3 здійснити перевірку за допомогою turbo assembler.

Контрольні питання

1. Що таке дійсні числа?
2. Наведіть приклади дійсних чисел.
3. Формат представлення дійсних чисел.
4. Що таке зміщення?
5. Алгоритм переведення дійсних чисел з десяткової в двійкову систему числення.
6. Алгоритм представлення дійсних чисел в двійковому нормалізованому вигляді.
7. Які існують машинні формати дійсних чисел?
8. Наведіть значення зміщень для кожного машинного формату дійсних чисел.
9. Формати машинного представлення дійсних чисел Dword і Qword.
10. Формат машинного представлення дійсних чисел Tbyte.

СПИСОК ЛІТЕРАТУРИ

1. Голубь Н.Г. Искусство программирования на Ассемблере. Лекции и упражнения. 2-е изд. испр. и доп. / Н.Г. Голубь. – СПб.: ООО "ДиаСофтЮП", 2002. – 656 с.URL (веб-посилання)
2. Юров В.И. Assembler. Учебник для вузов. 2-е изд. / В.И. Юров. – СПб.: Питер, 2010. – 637 с.URL (веб-посилання)
3. Юров В.И. Assembler. Учебник для вузов. / В.И. Юров. – СПб.: Питер, 2003. – 637 с.URL (веб-посилання)
4. Юров В.И. Assembler: Практикум. / В.И. Юров. – СПб.: Питер, 2003. – 400 с.URL (веб-посилання)
5. Пирогов В.Ю. Assembler. Учебный курс. / В.Ю. Пирогов. – М.: Издатель Молгачева С.В., Издательство Нолидж, 2001. – 848 с.URL (веб-посилання)
6. Магда Ю.С. Ассемблер для процессоров Intel Pentium. / Ю.С. Магда. – СПб.: Питер, 2006. – 400 с.URL (веб-посилання)
7. Рудольф Марек. Ассемблер на примерах. Базовый курс. / Марек Рудольф. – СПб: Наука и Техника, 2005. – 240 с.URL (веб-посилання)
8. Ирвин, Кип. Язык ассемблера для процессоров Intel, 4-е издание.: Пер. с англ. / Кип Ирвин. – М.: Издательский дом «Вильямс», 2005. – 912 с.URL (веб-посилання)
9. Абель П. Язык Ассемблера для IBM PC и программирования / Пер. с англ. Ю.В. Сальникова. - М.: Высш. шк., 1992. 447 с.

На в ч а л ь н о - м е т о д и ч н е в и д а н н я

АРХІТЕКТУРА КОМП'ЮТЕРА

Частина 1

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДЛЯ ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

Підготували

**Єфіменко Андрій Анатолійович, Байлюк Єлизавета Максимів-
на, Покотило Олександра Андріївна**

Редактори

Комп'ютерне верстання –

Свідоцтво про реєстрацію № __ від _____ 201_ року

Підписано до друку __.__.16. Формат 60×84/16.

Ум. друк. арк. _____. Зам. __ офс.

Безкоштовно

Друкарня ЖДТУ