

Лабораторна робота № 2

ВНУТРІШНЄ ПРЕДСТАВЛЕННЯ

ЦІЛОЧИСЕЛЬНИХ 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), символний 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

Кожне 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 ± 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.

Використовуючи відомі правила переводу, спочатку переведемо задані числа в двійкову систему числення за допомогою додаткового коду, а потім дамо їх внутрішнє представлення. Результати зведемо в таблицю 2.3.

$$\begin{array}{r}
 21 \mid 2 \\
 \hline
 \underline{20} \quad \underline{10} \mid 2 \\
 \textcolor{red}{1} \quad \underline{10} \quad 5 \mid 2 \\
 \hline
 \quad \textcolor{red}{0} \quad \underline{4} \quad 2 \mid 2 \\
 \hline
 \quad \quad \textcolor{red}{1} \quad \underline{2} \quad \textcolor{red}{1} \\
 \hline
 \quad \quad \quad \textcolor{red}{0}
 \end{array}$$

$21d \rightarrow 0001\ 0101b$ (BYTE)

- **$21d \rightarrow 1)$** $0001\ 0101b$ - двійковий код числа $|-21|$

2) $1110\ 1010b$ - інверсія

3) $\begin{array}{r} + \quad \quad 1 \\ \hline \end{array}$

$1110\ 1011b$ - додатковий код

Перевірка на assembler 5.0

Файл lr_02.asm

```
.MODEL tiny
.DATA
; lr_02
; ==== Byte ====
f0 db 21
; ==== Shortint ====
f1 db -21
; ==== Word ====
f2 dw 2216
; ==== Integer ====
f3 dw -2216
END
```

Файл лістингу lr_02.lst

```
Turbo Assembler Version 5.0                      01-21-22
22:17:56                      Page 1
lr_02.asm

1  0000                      .MODEL tiny
2  0000                      .DATA
3
4                          ; lr_02
5
6                          ; ==== Byte ====
7  0000  15                  f0 db 21
8
9                          ; ==== Shortint ====
10 0001  EB                 f1 db -21
11
12                          ; ==== Word ====
13 0002  08A8              f2 dw 2216
14
15                          ; ==== Integer ====
16 0004  F758              f3 dw -2216
17
18                          END
```

Приклад 2. Для заданого набору байтів 5A08 визначити які числові значення може містити цей набір, взятий у пам'яті комп'ютера.

Припустим що заданий набір байтів 5A08 – беззнаковий, то нам спочатку необхідно перевести його з шістнадцяткової системи числення у двійкову:

$$5A08_{16} = 5_{\sim 0101} A_{\sim 1010} 0_{\sim 0000} 8_{\sim 1000} = \\ 0101\ 1010\ 0000\ 1000_2,$$

а потім з шістнадцяткової у десяткову:

$$5A08_{16} = 5 \cdot 16^3 + 10 \cdot 16^2 + 8 \cdot 16^0 = 23048_{10},$$

або з двійкової у десяткову:

$$0101\ 1010\ 0000\ 1000_2 = 2^3 + 2^9 + 2^{11} + 2^{12} + 2^{14} \\ = = 23048_{10}$$

Але, якщо припустити що заданий набір байтів 5A08 – знаковий, то нам необхідно отримане двійкове значення інвертувати і додати одиницю до молодшого розряду для отримання додаткового коду:

1) 0101 1010 0000 1000 - двійковий код набору байтів 5A08

2) 1010 0101 1111 0111 - інверсія

3) $\begin{array}{r} + \\ \hline \end{array}$ 1

1010 0101 1111 0000 - **додатковий код**

Отримане двійкове значення переводимо у десяткову систему числення:

$$1010\ 0101\ 1111\ 1000_2 = -42488_{10}$$

Виходячи з вище приведених розрахунків заданий набір байтів 5A08 може відповідати двом десятковим числам: 23048₁₀ та -42488₁₀.

Перевірка на assembler 5.0

```
Turbo Assembler Version 5.0          01-21-22
22:27:30          Page 1
lr_02.asm
1  0000          .MODEL tiny
2  0000          .DATA
3
4              ; lr_02
5
6              ; ==== Integer ====
7  0000  5A08     f0      dw  23048
8  0002  5A08     dw      -42488
```

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;
- знаковими.

Для збереження чисел використовується 16 та 32 біти. Представити перевірки на мові assembler 5.0.

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;
- знаковими.

Для збереження чисел використовується 32 біти. Представити перевірки на мові assembler 5.0.

3. За даними табл. 2.7 визначити які числові значення може містити знаковий набір байтів, взятий у пам'яті комп'ютера. Представити перевірки на мові assembler 5.0.

Вимоги до оформлення звіту

Звіт оформлюється у текстовому документі і має містити:

1. Правильно оформлену рамку.
2. Тему, мету, номер варіанту.
3. Виконані **всі** завдання з ходом обрахунків і перевітками на мові Assmbler.
4. Висновки.

Звіт оформлений неналежним чином і без ходу обрахунків не оцінюється і має бути перероблений!

Таблиця 2.5

Дані для виконання завдання 1

№ варіанта	Число 1	Число 2
1	$\pm(6788+G)$	$\pm(13237+G)$
2	$\pm(4309+G)$	$\pm(10580+G)$
3	$\pm(8248+G)$	$\pm(14408+G)$
4	$\pm(5177+G)$	$\pm(11891+G)$
5	$\pm(9042+G)$	$\pm(10149+G)$
6	$\pm(6054+G)$	$\pm(13673+G)$
7	$\pm(4750+G)$	$\pm(12791+G)$
8	$\pm(7974+G)$	$\pm(11441+G)$
9	$\pm(5462+G)$	$\pm(14850+G)$
10	$\pm(8698+G)$	$\pm(11004+G)$
11	$\pm(4842+G)$	$\pm(12945+G)$
12	$\pm(7229+G)$	$\pm(11738+G)$
13	$\pm(5898+G)$	$\pm(14124+G)$
14	$\pm(8401+G)$	$\pm(10289+G)$
15	$\pm(4605+G)$	$\pm(13531+G)$
16	$\pm(6939+G)$	$\pm(12191+G)$
17	$\pm(8847+G)$	$\pm(10723+G)$
18	$\pm(5037+G)$	$\pm(14698+G)$
19	$\pm(7380+G)$	$\pm(11295+G)$
20	$\pm(4159+G)$	$\pm(13975+G)$
21	$\pm(8114+G)$	$\pm(12479+G)$
22	$\pm(5605+G)$	$\pm(15035+G)$
23	$\pm(6496+G)$	$\pm(10860+G)$
24	$\pm(7670+G)$	$\pm(14264+G)$
25	$\pm(4892+G)$	$\pm(11593+G)$
26	$\pm(8549+G)$	$\pm(12644+G)$
27	$\pm(5313+G)$	$\pm(13467+G)$
28	$\pm(7815+G)$	$\pm(13832+G)$
29	$\pm(6346+G)$	$\pm(12042+G)$
30	$\pm(7518+G)$	$\pm(13390+G)$
31	$\pm(4461+G)$	$\pm(14546+G)$
32	$\pm(8695+G)$	$\pm(13090+G)$
33	$\pm(5730+G)$	$\pm(11145+G)$
34	$\pm(5892+G)$	$\pm(12646+G)$
35	$\pm(7506+G)$	$\pm(11955+G)$

Таблиця 2.6

Дані для виконання завдання 2

№ варіанта	Число 1	Число 2
1	$\pm(22036+G)$	$\pm(60825+G)$
2	$\pm(24744+G)$	$\pm(69990+G)$
3	$\pm(28894+G)$	$\pm(69572+G)$
4	$\pm(23560+G)$	$\pm(63828+G)$
5	$\pm(28598+G)$	$\pm(64162+G)$
6	$\pm(20440+G)$	$\pm(62997+G)$
7	$\pm(25062+G)$	$\pm(66178+G)$
8	$\pm(24149+G)$	$\pm(60582+G)$
9	$\pm(24389+G)$	$\pm(61610+G)$
10	$\pm(23898+G)$	$\pm(62299+G)$
11	$\pm(28369+G)$	$\pm(63256+G)$
12	$\pm(27124+G)$	$\pm(61367+G)$
13	$\pm(20662+G)$	$\pm(64552+G)$
14	$\pm(25940+G)$	$\pm(65109+G)$
15	$\pm(26296+G)$	$\pm(68164+G)$
16	$\pm(29665+G)$	$\pm(67459+G)$
17	$\pm(29431+G)$	$\pm(67951+G)$
18	$\pm(25584+G)$	$\pm(61966+G)$
19	$\pm(28165+G)$	$\pm(68573+G)$
20	$\pm(22629+G)$	$\pm(63601+G)$
21	$\pm(21086+G)$	$\pm(69320+G)$
22	$\pm(29145+G)$	$\pm(67715+G)$
23	$\pm(27418+G)$	$\pm(62649+G)$
24	$\pm(29950+G)$	$\pm(65658+G)$
25	$\pm(20010+G)$	$\pm(69105+G)$
26	$\pm(22932+G)$	$\pm(60203+G)$
27	$\pm(27744+G)$	$\pm(61143+G)$
28	$\pm(20216+G)$	$\pm(67102+G)$
29	$\pm(23203+G)$	$\pm(68365+G)$
30	$\pm(25358+G)$	$\pm(66797+G)$
31	$\pm(21720+G)$	$\pm(65890+G)$
32	$\pm(21484+G)$	$\pm(66535+G)$
33	$\pm(22377+G)$	$\pm(65396+G)$
34	$\pm(26531+G)$	$\pm(68857+G)$
35	$\pm(26800+G)$	$\pm(64787+G)$

Таблиця 2.7

Дані для виконання завдання 4

№ варіанта	Число 1	Число 2
1	8ADA	61DE74CB
2	24D1	57FDAF20
3	2577	6129B2A9
4	3D64	E5E6AAB7
5	BB68	B794FFE2
6	CE5E	A52E350F
7	313D	2961319B
8	5E44	C73331BE
9	C894	8E523095
10	F210	81C6C852
11	D204	93B77A38
12	ADF9	91039248
13	A85C	79F1FEAB
14	B5A4	FDEA17D6
15	4321	31011210
16	9D7C	70F6CAAEE
17	725D	C9C88F30
18	2734	AEB275E9
19	51CF	8EF1F2BF
20	783B	92A36E24
21	1304	AB9BF9F3
22	BC0D	88FB0DB6
23	2022	286F9385
24	A2D7	5F8C6A22
25	8091	23820AF6
26	C4F0	ECC42697
27	55C4	F23C4DE3
28	1578	7586D306
29	77B9	3010ADA3
30	A039	C8699D73
31	3F8C	D2EF2925
32	8E01	D7944590
33	4E68	F7A9F85A
34	A94F	9DE95FEC
35	6BC9	CB565885

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.