

переносу *cf* в регістрі ознак), а у наймолодший розряд записується 0, наприклад:

$$\begin{array}{r} \text{SHL} \\ \hline \begin{array}{cccccccc} X_7 & X_6 & X_5 & X_4 & X_3 & X_2 & X_1 & X_0 \end{array} \\ = \\ \begin{array}{cccccccc} X_6 & X_5 & X_4 & X_3 & X_2 & X_1 & X_0 & 0 \end{array} \end{array}$$

Лінійний зсув вправо (SHift Right - SHR). В цій операції значення розряду даних заміщується значенням попереднього старшого розряду. При цьому значення самого молодшого розряду даних втрачається (у відповідних командах EOM значення самого молодшого розряду записується в ознаку переносу в регістрі ознак), а у найстарший розряд записується 0, наприклад:

$$\begin{array}{r} \text{SHR} \\ \hline \begin{array}{cccccccc} X_7 & X_6 & X_5 & X_4 & X_3 & X_2 & X_1 & X_0 \end{array} \\ = \\ \begin{array}{cccccccc} 0 & X_7 & X_6 & X_5 & X_4 & X_3 & X_2 & X_1 \end{array} \end{array}$$

Зсув на декілька розрядів можна розглядати як послідовність зсувів на один розряд.

1.3. Завдання на виконання роботи

Перше заняття

1) Переглянути текст програми lab1.asm, зміст якої представлений нижче.

При відсутності відповідного файлу на диску, підготувати його, наприклад, за допомогою додатку Notepad.

Програма lab1.asm ілюструє основні елементи, що властиві програмам мовою Асемблера. Вона складається з двох логічних сегментів: *data* і *code*. Логічний сегмент *data* розміщується у фізичному сегменті даних, який адресується регістром **DS**. Логічний сегмент кодів *code* розташовується у фізичному сегменті кодів, який адресується сегментним регістром **CS**.

; програма lab1.asm

```
data SEGMENT BYTE
    val1      db    3
    val2      db    2
    result    db    ?
```

```
data ENDS
```

```
code SEGMENT
```

```
    ASSUME cs:code, ds:data
```

; директива assume інформує Асемблер, що у сегментному регістрі cs міститься адреса

; сегмента code (її в регістр cs помістив завантажувач), а у сегментному регістрі ds –

; адреса сегмента data

begin:

```
    MOV      AX, DATA ; запис адреси сегмента data в регістр ax
    MOV      DS, AX    ; запис адреси сегмента data в регістр ds
                        ; (це було “обіцяно” Асемблеру в директиві assume)
    mov      al, val1   ; запис в регістр al значення змінної за адресою val1
    add      al, val2   ; додавання значення змінної за адресою val2
                        ; до вмісту регістра al і запис результату в регістр al
    mov      result, al ; пересилання вмісту регістра al у сегмент даних за
                        ; адресою result
```

; фактично реалізований оператор result := val1+val2;

; **ОСОБЛИВІСТЬ АСЕМБЛЕРА:**

; якщо мовою Паскаль під ідентифікаторами result, val1 та val2 звичайно розуміють

; значення змінних, то мовою Асемблера – **адреси змінних** у сегменті даних

```
    nop      ; холоста команда процесора, у даному випадку
                ; використовується для зручності при роботі з налагоджувачем
```

```
    MOV      AX, 4C00H ; повернення в операційну систему
```

```
    INT      21H
```

```
code ENDS
```

```
end      begin
```

- 2) Відтранслювати програму з файлу lab1.asm за допомогою транслятора MASM. Створити об'єктний файл і файл лістингу.
- 3) Створити exe-файл за допомогою компоувальника LINK.

- 4) Запустити налагоджувач AFD. За допомогою команди L завантажити програму в середовище налагоджувача. Виконати програму покомандно (клавіша F1) до команди **пор**. Звернути увагу на зміну вмісту регістрів процесора та пам'яті після виконання кожної з команд.
- 5) Повернутися на початок програми. Виконати перші дві команди, далі здійснити заміну вмісту регістра **AL** мікропроцесора та заміну вмісту областей оперативної пам'яті **val1**, **val2** перед командами, що їх використовують. Пересвідчитись в отриманні нових результатів.

Друге заняття

- 1) Розробити програму, яка виконує перетворення вмісту довільного однобайтового числа шляхом переміщення (перестановок) його окремих бітів відповідно до варіанта (табл. 1.4). Наприклад, для варіанта 1: значення 7-го розряду вхідного операнда потрібно записати в 0-й та 2-й розряд результату, 2-го – у 1-й, 4-й та 7-й і т.д., 0-го – у 6-й. Вхідний операнд помістити в регістр DL, а результат записати в регістр DH.

Для виконання завдання необхідно ознайомитися з командами пересилання даних (**MOV**), порозрядної логічної обробки (**OR**, **AND**) та командами лінійного зсуву (**SHL**, **SHR**) процесорів Intel 80x86 і Pentium. Допускається використання будь-яких інших команд за вибором студента.

- 2) Перевірити коректність переміщень у налагоджувачі на різних даних.

Таблиця 1.4

Варіанти завдання

	№ варіанта	Номер розряду результату 7 6 5 4 3 2 1 0	№ варіанта	Номер розряду результату 7 6 5 4 3 2 1 0
	1	2 0 1 2 3 7 2 7	16	3 4 5 6 1 3 2 5
Номери розрядів початкових даних	2	7 7 2 3 2 6 7 5	17	5 2 7 3 1 3 0 5
	3	1 6 4 3 6 2 0 6	18	0 5 4 0 6 7 4 1
	4	7 2 5 4 7 7 0 4	19	3 0 5 5 2 6 2 7
	5	6 4 7 6 5 1 5 1	20	5 0 6 6 1 2 4 4
	6	4 7 7 5 7 0 1 3	21	5 4 1 6 5 1 1 1
	7	1 0 6 7 2 2 2 6	22	0 6 7 6 4 3 0 5
	8	7 3 2 0 6 3 5 3	23	6 3 3 3 7 0 6 1
	9	6 3 0 5 3 5 1 1	24	7 7 7 5 5 2 0 5
	10	0 0 5 2 2 5 6 1	25	0 3 4 5 4 1 0 4
	11	6 1 2 5 6 0 5 7	26	4 3 6 1 1 3 7 6
	12	0 0 7 5 3 4 0 4	27	7 0 5 1 7 6 7 1
	13	4 5 2 4 4 1 1 4	28	1 7 2 3 3 6 4 5
	14	4 7 7 5 5 5 5 7	29	0 7 4 1 5 0 3 2
	15	4 3 7 6 4 4 3 6	30	6 1 5 5 1 6 2 2

1.4. Контрольні запитання

1. Яку структуру має програма мовою Асемблера?
2. Яке призначення мають програми MASM, LINK і AFD?
3. Назвіть основні опції програм MASM, LINK і AFD.
4. Як визначити основну програму мовою Асемблера?
5. Чим відрізняються директиви мови Асемблера від її машинних інструкцій?