

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине “Архитектура вычислительных систем”

Студент

Станиславчук С. М.

Группа АС-21-1

Руководитель

Болдырихин О. В.

Ст. преподаватель

Липецк 2023

Цель работы:

Изучение основ устройства и принципов работы компьютера фон-неймановской архитектуры.

Задание кафедры: Вариант 27

Написать на языке ассемблера программу, выполняющую преобразование числа в упакованный двоично-десятичный код.

При помощи отладчика прогнать программу покомандно и после выполнения каждой команды фиксировать состояние аккумулятора, указателя команд, других регистров, задействованных в программе, ячеек памяти данных.

Результаты анализа работы программы оформить в виде таблицы. Последовательность строк в таблице должна соответствовать последовательности выполнения команд в период прогона программы, а не их последовательности в тексте программы. В строке, соответствующей данной команде, содержимое регистров и памяти должно быть таким, каким оно является после ее выполнения.

Проанализировать таблицу, выполнить необходимые сравнения, сделать выводы.

27	Преобразование числа в упакованный двоично-десятичный код	Сегмент данных (по DS) и сегмент команд	Дополнительный сегмент данных (по ES)
----	---	---	---------------------------------------

Ход работы:

1. Блок-схема алгоритма программы

Составим блок-схему алгоритма преобразования в код с дублированием битов – результат указан на рисунке 1.

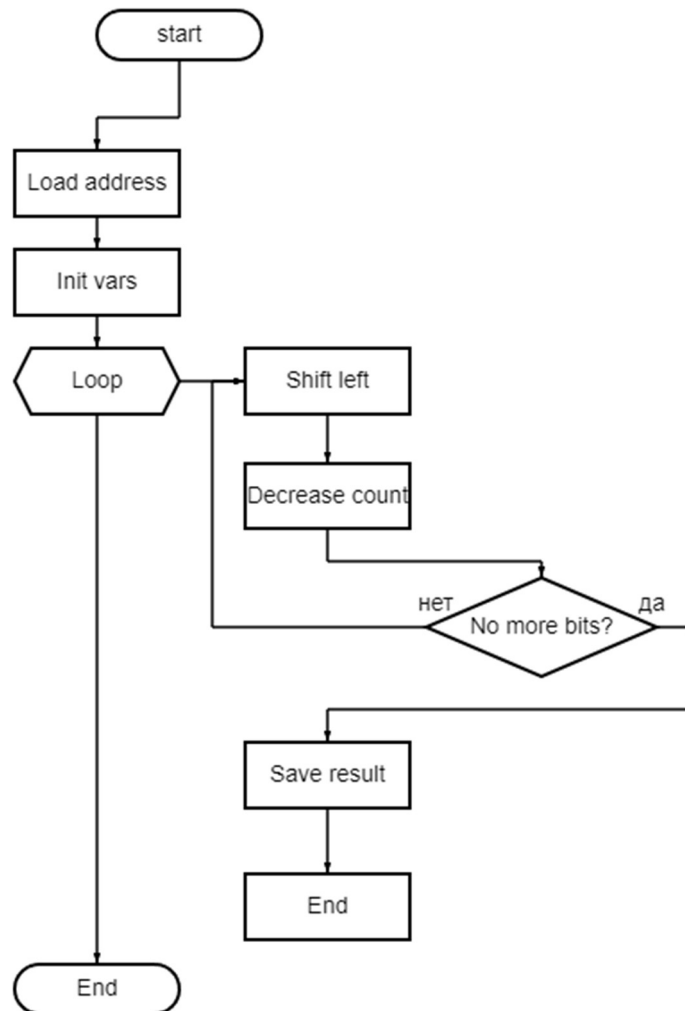


Рисунок 1 – Блок-схема программы.

2. Ручной расчет по алгоритму:

Число 83 в упакованном двоично десятичном коде (packed BCD):

8 -> 1000

3 -> 0011

83 -> 1000 0011

3. Код программы

```
.model small

data segment
    input db 83
    res1 db 0
data ends

code segment
    res2 db 0
    assume DS:data, ES:code, CS:code

start:
    mov ax, data
    mov ds, ax

    mov ax, code
    mov es, ax

    mov al, ds:[input]
    xor ah, ah
    mov bl, 10
    div bl

    mov dl, al
    mov al, ah

    shl dl, 4
    or al, dl

    mov ds:[res1], al
    mov es:[res2], al

    mov ax, 0
    mov al, ds:[res1]
    mov ah, 4Ch
    int 21h
end start
```

4. Листинг программы

```
1. 0000: mov ax, data          ; 0000: B8 678
2. 0004: mov ds, ax            ; 0004: 8E D8
3. 0006: mov ax, code          ; 0006: B8 B348
4. 0009: mov es, ax            ; 0009: 8E C0
5. 000B: mov al, ds:[input]    ; 000B: A0 0000
6. 000E: xor ah, ah            ; 000E: 32 E4
7. 0010: mov bl, 10            ; 0010: B3 0A
8. 0012: div bl                ; 0012: F6 F3
9. 0014: mov dl, al            ; 0014: 8A D0
10. 0016: mov al, ah           ; 0016: 8A C4
11. 0018: shl dl, 1            ; 0018: D0 E2
12. 001A: shl dl, 1            ; 001A: D0 E2
13. 001C: shl dl, 1            ; 001C: D0 E2
14. 001E: shl dl, 1            ; 001E: D0 E2
15. 0020: or al, dl            ; 0020: 0A C2
16. 0022: mov ds:[res1], al    ; 0022: A2 0100
17. 0025: mov es:[res2], al    ; 0025: 26 A20000
18. 0029: mov al, ds:[res1]    ; 0029: B8 0000
19. 002C: mov al, ds:[res1]    ; 002C: A0 0100
20. 002E: mov ah, 4Ch          ; 002E: B4 4C
21. 0031: int 21h              ; 0031: CD 21
```

5. Таблица состояния системы

Составим таблицу состояний системы после выполнения каждой команды (таблица 1)

Таблица 1 – Состояния системы после выполнения команд программы

Номер команд ы	Адрес команд ы	Команда на машинном языке	Регистр команд	Команда на языке ассемблера	Указатель команд	Содержание изменившихся регистров и ячеек памяти
1	0000	B8 678	B8	mov ax, data	0001	ax 48B7
2	0004	8E D8	8E	mov ds, ax	0004	ax 48B7
3	0006	B8 B348	B8	mov ax, code	0006	ds 48B7
4	0009	8E C0	8E	mov es, ax	0009	es 48B3

5	000B	A0 0000	A0	mov al, ds:[input]	000B	es 48B3
6	000E	32 E4	32	xor ah, ah	000E	es 4853
7	0010	B3 0A	B3	mov bl, 10	0010	ax 0053
8	0012	F6 F3	F6	div bl	0012	bx 000A
9	0014	8ADO	8A	mov dl, al	0014	ax 0308
10	0016	8A C4	8A	mov al, ah	0016	dx 0008
11	0018	D0 E2	D0	shl dl, 1	0018	dx 0303
12	001A	D0 E2	D0	shl dl, 1	001A	dx 0010
13	0016	D0 E2	D0	shl dl, 1	0016	dx 0020
14	001E	D0 E2	D0	shl dl, 1	001E	dx 0040
15	0020	0AC2	0A	or al, dl	0020	dx 0080
16	0022	A20100	A2	mov ds:[res1], al	0022	ax 0383, ds:0001 = 00
17	0025	26A20000	A2	mov es:[res2], al	0025	es:0000 = 00
18	0029	B80000	B8	mov al, ds:[res1]	002C	ax 0000
19	002C	A00100	A0	mov al, ds:[res1]	002E	ax 0083
20	002E	B44C	B4	mov ah, 4Ch	002F	ax 0083
21	0031	CD21	CD	int 21h	0031	ax 4C83

6. Проверка работы алгоритма на правильных числах

Упакованный двоично-десятичный код (Packed Binary Coded Decimal, PBCD) - это способ представления десятичных чисел в формате, где каждая десятичная цифра представлена в виде 4-битного двоичного числа. В упакованном PBCD каждая десятичная цифра (0-9) кодируется с использованием 4 битов, и эти коды объединяются вместе, чтобы представить десятичное число.

На вход программе подается число 83. Программа разбивает это число на составные цифры (8 и 3) с помощью битовых масок. После разбиения

происходит перевод и склеивание битов этих чисел с последующим занесением результата в переменную result, которая находится в сегменте DS. На рисунке 2 видно, что в сегменте DS по смещению 0000 (переменная result) лежит число 83h. А это значит, что программа отработала верно. Результат программы и состояние регистров CPU можно увидеть на рисунках 2 и 3 соответственно.

The screenshot shows the DOSBox 0.74-3 interface with the CPU 80486 window open. The CPU window displays the state of the CPU registers and the DS segment. The DS segment is shown at the bottom of the CPU window, with the address 0000 containing the value 83h. The CPU registers are shown on the right side of the CPU window, with the DS register containing the value 48B3. The status bar at the bottom shows the current instruction and the state of the CPU.

Address	Value
cs:001E	D0E2
cs:0020	0AC2
cs:0022	A20000
cs:0025	B80000
cs:0028	A00000
cs:002B	B44C
cs:002D	CD21
cs:002F	005300
cs:0032	0000
cs:0034	0000
cs:0036	0000
cs:0038	0000
cs:003A	0000
ds:0000	83 00 00 00 00 00 00 00
ds:0008	00 00 00 00 00 00 00 00
ds:0010	FB 52 09 02 22 00 00 00
ds:0018	05 00 1E 00 00 00 03 00

Register	Value
ax	4C83
bx	1A0A
cx	1A01
dx	F680
si	26F1
di	01A5
bp	0100
sp	0000
ds	48B7
es	48B6
ss	48B2
cs	48B3
ip	002D

Рисунок 2 – Состояние сегмента DS (result) на момент завершения программы.

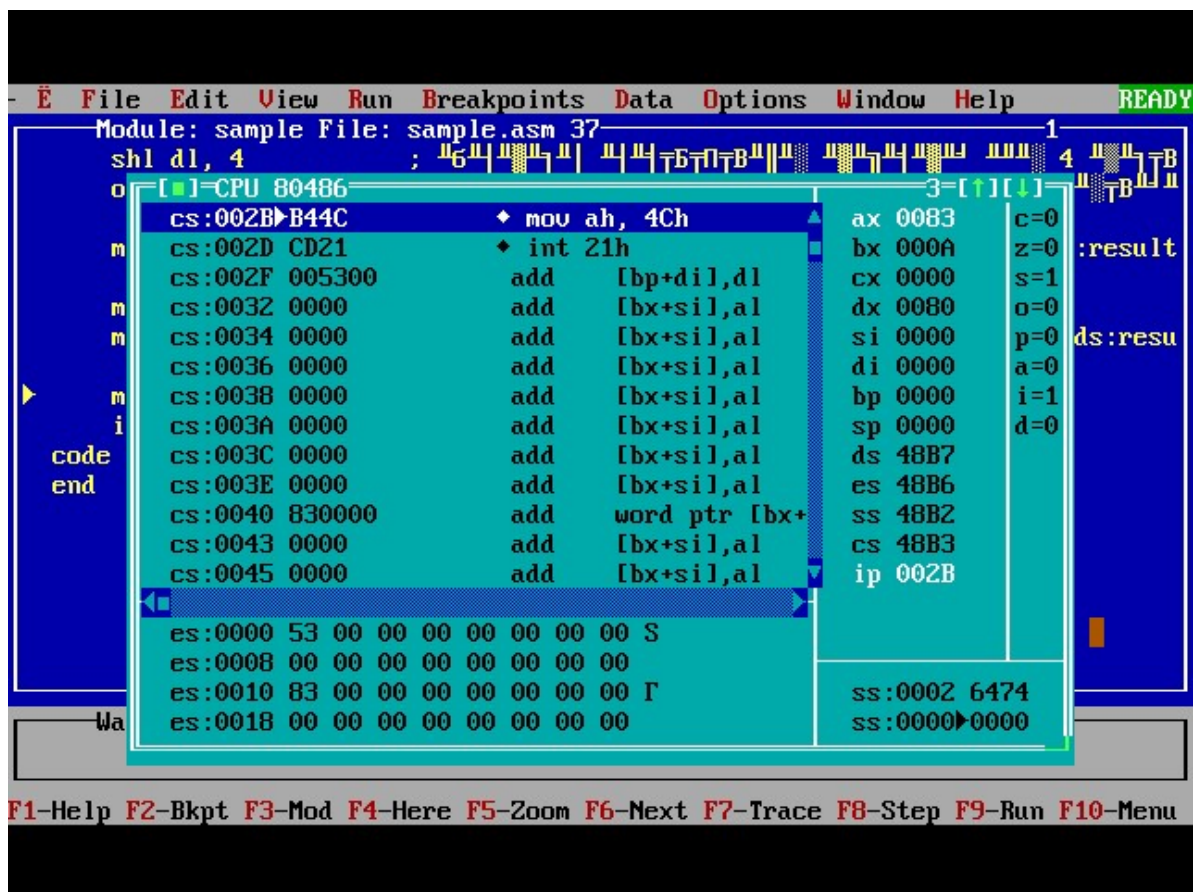


Рисунок 3 – Состояние сегмента ES (data) на момент завершения программы.

7. Вывод

В ходе выполненной работы рассмотрел и проанализировал программу на ассемблере, которая выполняет преобразование двоичного числа в упакованный двоично-десятичный код.

Заметил, что при использовании одинаковых команд на переменные с одинаковыми значениями, но находящихся в разных сегментах, команды на машинном языке отличаются (таблица 1, номера команд 16 и 17. Команда номер 17 имеет префикс “26”, что данные будут читаться или записываться в сегмент ES).