**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине Архитектура вычислительных систем

«Взаимодействие процессора с памятью и устройствами ввода-вывода. Функционирование ЭВМ»

|  |  |  |
| --- | --- | --- |
| Студент  Группа АИ-20 |  | Глубоков Г.В. |
| Руководитель |  | Болдырихин О. В. |

Липецк 2022 г.

**Цель работы**

Изучение порядка функционирования ЭВМ, организации взаимодействия процессора с памятью и устройствами ввода-  
вывода.

**Задание кафедры**

Написать на языке ассемблера программу, выполняющую определенную задачу в соответствии с вариантом.

Исходные данные вводятся в память с клавиатуры, результаты выводятся на дисплей. Сообщение об ошибке выводится в виде звукового сигнала через динамик.

Работа с устройствами ввода-вывода должна полностью осуществляться средствами программы (использование функций BIOS или операционной системы не допускается), то есть в программе должны быть команды ввода-вывода из соответствующих портов (регистров) клавиатуры, видеоадаптера, таймера.

Исследовать работу программы, отслеживая состояние системы после выполнения команд: значения регистров процессора, ячеек памяти, портов ввода-вывода. 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, ячеек памяти и портов ввода-вывода должно быть таким, каким оно является после ее выполнения.

Проанализировать таблицу, выполнить необходимые сравнения, сделать выводы.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Задача, выполняемая программой | Формат ввода-вывода | Расположение данных на экране |
| 2 | Преобразование числа из кода Грея | Двоичный | Исходные данные в  первой строке, результаты в третьей |

**Ход работы**

Листинг программы

Turbo Assembler Version 4.1 11/16/22 00:37:14 Page 1

main.asm

1 0000 data segment

2 0000 00 in\_num db (?)

3 0001 00 out\_num db (?)

4 0002 data ends

5

6 0000 code segment

7 assume cs:code, ds:data

8 org 100h

9 0100 main:

10 0100 BA 0000s mov dx, data

11 0103 8E DA mov ds, dx

12 0105 BA B800 mov dx, 0B800h

13 0108 8E C2 mov es, dx

14 010A 33 FF xor di, di

15 010C 33 D2 xor dx, dx

16 010E B8 0720 mov ax, 0720h

17 0111 B9 07D0 mov cx, 80\*25

18 0114 F3> AB rep stosw

19 0116 33 FF xor di, di

20 0118 B9 0008 mov cx, 8h

21 011B inloop:

22 011B E4 60 in al, 60h

23 011D 3A C3 cmp al, bl

24 011F 74 FA je inloop

25 0121 8A D8 mov bl, al

26 0123 3C 01 cmp al, 01h

27 0125 75 03 jne NotEsc

28 0127 EB 7C 90 jmp quit

29 012A NotEsc:

30 012A 3C 02 cmp al, 02h

31 012C 74 0E je num1

32 012E 3C 0B cmp al, 0Bh

33 0130 75 E9 jne inloop

34 0132 D1 E2 shl dx, 1

35 0134 26: C7 05 0730 mov es:[di], 0730h

36 0139 EB 0B 90 jmp NxtNum

37 013C num1:

38 013C D1 E2 shl dx, 1

39 013E 83 CA 01 or dx, 1

40 0141 26: C7 05 0731 mov es:[di], 0731h

41 0146 NxtNum:

42 0146 47 inc di

43 0147 47 inc di

44 0148 49 dec cx

45 0149 75 D0 jnz inloop

46

47 014B 88 16 0000r mov in\_num, dl

48 014F A0 0000r mov al, in\_num

49 0152 B9 0007 mov cx, 7

50 0155 Gray:

51 0155 D0 E8 shr al, 1

52 0157 32 06 0000r xor al, in\_num

53 015B E2 F8 loop Gray

54

55 015D A2 0001r mov out\_num, al

56

57 0160 B0 24 mov al, 24h

58 0162 E6 42 out 42h, al

59 0164 B0 0E mov al, 0Eh

60 0166 E6 42 out 42h, al

61 0168 E4 61 in al, 61h

62 016A 0C 03 or al, 00000011b

63 016C E6 61 out 61h, al

64 016E B9 FFFF mov cx, 0FFFFh

65 0171 l1:

66 0171 51 push cx

67 0172 B9 000F mov cx, 00Fh

68 0175 l2:

69 0175 E2 FE loop l2

70 0177 59 pop cx

71 0178 E2 F7 loop l1

72 017A E4 61 in al, 61h

73 017C 24 FC and al, 11111100b

74 017E E6 61 out 61h, al

75 0180 B3 80 mov bl, 80h

76 0182 BF 0140 mov di, 80\*4

77 0185 outloop:

78 0185 A0 0001r mov al, out\_num

79 0188 84 C3 test al, bl

80 018A 75 08 jnz bit1

81 018C 26: C7 05 0730 mov es:[di], 0730h

82 0191 EB 06 90 jmp NxtBit

83 0194 bit1:

84 0194 26: C7 05 0731 mov es:[di], 0731h

85 0199 NxtBit:

86 0199 47 inc di

87 019A 47 inc di

88 019B D0 EB shr bl, 1

89 019D 73 E6 jnc outloop

90 019F pause:

91 019F E4 60 in al, 60h

92 01A1 3C 01 cmp al, 01h

93 01A3 75 FA jne pause

94 01A5 quit:

95 01A5 B8 4C00 mov ax, 4C00h

96 01A8 CD 21 int 21h

97 01AA code ends

98 end main

Таблица 1 - Состояния системы после выполнения команд

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Адрес команды | Команда на машинном языке | Команда на языке ассемблера | IP | IR | Содержимое изменившихся регистров и ячеек памяти |
| 1 | 0100 | BA051A | mov dx, 1A05 | 0103 | BA | DX=1A05 |
| 2 | 0103 | 8EDA | mov ds, dx | 0105 | 8EDA | DS=1A05 |
| 3 | 0105 | BA00B8 | mov dx, B800 | 0108 | BA | DX=B800 |
| 4 | 0108 | 8EC2 | mov es, dx | 010A | 8EC2 | ES=B800 |
| 5 | 010A | 33FF | xor di, di | 010C | 33FF | Z=1 P=1 |
| 6 | 010C | 33D2 | xor dx, dx | 010E | 33D2 | DX=0000 |
| 7 | 010E | B82007 | mov ax, 0720 | 0111 | B8 | AX=0720 |
| 8 | 0111 | B9D007 | mov cx, 07D0 | 0114 | B9 | CX=07D0 |
| 9 | 0114 | F3AB | rep stosw | 0116 | F3AB | CX = 0000  DI = 0FA0  ES:[0000]-ES:[0F9E]= 0720 |
| 10 | 0116 | 33FF | xor di, di | 0118 | 33FF | DI=0000 |
| 11 | 0118 | B90800 | mov cx, 08 | 011B | B9 | CX=0008 |
|  |  |  | Начало цикла ввода |  |  |  |
| 12 | 011B | E460 | in al, 60 | 011D | E4 | AX=070B |
| 13 | 011D | 3AC3 | cmp al, bl | 011F | 3AC3 | Z=0 P=0 |
| 14 | 011F | 74FA | je 011B | 0121 | 74 |  |
| 15 | 0121 | 8AD8 | mov bl, al | 0123 | 8AD8 | BX=000B |
| 16 | 0123 | 3C01 | cmp al, 01 | 0125 | 3C | P=1 |
| 17 | 0125 | 7503 | jne 012A | 012A | 75 |  |
| 18 | 012A | 3C02 | cmp al, 02 | 012C | 3C | P=1 |
| 19 | 012C | 740E | je 013C | 012E | 74 |  |
| 20 | 012E | 3C0B | cmp al, 0B | 0130 | 3C | Z=1 |
| 21 | 0130 | 75E9 | jne 011B | 0132 | 75 |  |
| 22 | 0132 | D1E2 | shl dx, 1 | 0134 | D1E2 | DX=0000 |
| 23 | 0134 | 26C7053007 | mov es:[di], 0730 | 0139 | 26C705 | ES:[DI] = 0730 |
| 24 | 0139 | EB0B | jmp 0146 | 0146 | EB |  |
| 25 | 0146 | 47 | inc di | 0147 | 47 | DI = 0001 Z=0 P=0 |
| 26 | 0147 | 47 | inc di | 0148 | 47 | DI=0002 |
| 27 | 0148 | 49 | dec cx | 0149 | 49 | CX=000F P=1 A=1 |
| 28 | 0149 | 75D0 | jne 011B | 011B | 75 |  |
|  |  |  | Конец цикла ввода | 014B |  | AX=0702  BX=0002  DX=0709  DI=0020 |
| 29 | 014B | 88160000 | mov [0000], dl | 014F | 8816 | DS[0000]=3 |
| 30 | 014F | A00000 | mov al, [0000] | 0152 | A1 | AL=3 |
| 31 | 0152 | B90007 | Mov cx, 7 | 0155 | 2306 | cX=0007 |
| 32 | 0156 | D0E8 | Shr al, 1 | 0157 | D0 | AL = 1 |
| 33 | 0157 | 32060000 | Xor al, [0000] | 015B | A1 | AL = 2 |
| 34 | 015B | E2F8 | Loop 0155 | 0155 | E2F8 |  |
| 35 | 0175 | B024 | mov al, 24 | 0177 | B0 | AX=0024 |
| 36 | 0177 | E642 | out 42, al | 0179 | E6 | Порт 42h=24 |
| 37 | 0179 | B00E | mov al, 0E | 017B | B0 | AX=000E |
| 38 | 017B | E642 | out 42, al | 017D | E6 | Порт 42h=0E |
| 39 | 017D | E461 | in al, 61 | 017F | E4 | AX=0031 |
| 40 | 017F | 0C03 | or al, 03 | 0181 | 0C | AX=0033 |
| 41 | 0181 | E661 | out 61, al | 0183 | E6 | Порт 61h=33 |
| 42 | 0183 | B9FFFF | mov cx, FFFF | 0186 | B9 | CX=FFFF |
| 43 | 0186 | 51 | push cx | 0187 | 51 | SP=0000 |
| 44 | 0187 | B9FF0E | mov cx, 03FF | 018A | B9 | CX=03FF |
| 45 | 018A | E2FE | loop 018A | 018C | E2 | CX=0000 |
| 46 | 018C | 59 | pop cx | 018D | 59 | CX=FFFF SP=0002 |
| 47 | 018D | E2F7 | loop 0186 | 018F | E2 | CX=FFFE |
| 48 | 018F | E461 | in al, 61 | 0191 | E4 | AX=0021 |
| 49 | 0191 | 24FC | and al, FC | 0193 | 24 | AX=0020 P=0 |
|  | 0193 | E661 | out 61, al | 0195 | E6 | Порт 61h=20 |
| 50 |  |  | Начало цикла вывода |  |  |  |
| 51 | 0195 | BB0080 | mov bx, 8000 | 0198 | BB | BX=8000 |
| 52 | 0198 | BF4001 | mov di, 0140 | 019B | BF | DI=0140 |
| 53 | 019B | A10600 | mov ax, [0006] | 019E | A1 | AX=0079 |
| 54 | 019E | 85C3 | test bx, ax | 01A0 | 85C3 | Z=1 P=1 |
| 55 | 01A0 | 7508 | jne 01AA | 01A2 | 75 |  |
| 56 | 01A2 | 26C7053007 | mov es:[di], 0730 | 01A7 | 26C705 | ES:[di] = 0730 |
| 57 | 01A7 | EB06 | jmp 01AF | 01AF | EB |  |
| 58 | 01AF | 47 | inc di | 01B0 | 47 | DI=0141 Z=0 |
| 59 | 01B0 | 47 | inc di | 01B1 | 47 | DI=0142 |
| 60 | 01B1 | D1EB | shr bx, 1 | 01B3 | D1EB | BX=4000 O=1 |
|  | 01B3 | 73E6 | jnb 019B | 019B | 73 |  |
|  |  |  | Конец цикла вывода | 01B5 |  | BX=0000  DI=0160  C=1  Z=1  O=0 |
| 61 |  |  | Начало цикла ожидания нажатия клавиши выхода |  |  |  |
| 62 | 01B5 | E460 | in al, 60h | 01B7 | E4 | AX=0041 |
| 63 | 01B7 | 3C01 | cmp al, 01 | 01B9 | 3C | Z=0 C=0 P=0 |
| 64 | 01B9 | 75FA | jne 01B5 | 01B5 | 75 |  |
| 65 | 01B5 | E460 | in al, 60h | 01B7 | E4 | AX=0001 |
| 66 | 01B7 | 3C01 | cmp al, 01 | 01B9 | 3C | Z=1 P=1 |
|  | 01B9 | 75FA | jne 01B5 | 01BB | 75 |  |
| 67 |  |  | Конец цикла ожидания нажатия клавиши выхода |  |  |  |
| 68 | 01BB | B8004C | mov ax, 4C00 | 01BE | B8 | AX=4С00 |
|  | 01BE | CD21 | int 21h |  | CD |  |

Пример работы программы приведен на рисунке 1.

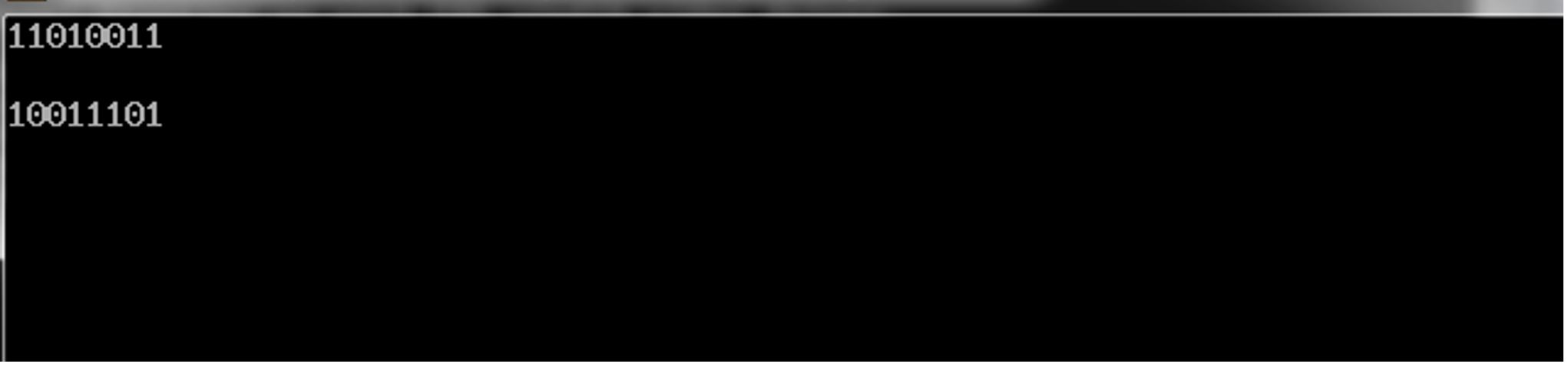


Рисунок 1 – Пример работы программы

**Вывод**

В данной лабораторной работе я написал программу для преобразования в число из кода Грея, при этом в ней исходные данные вводились в программу с клавиатуры, а полученный код выводился в консоль. Для вывода данных на дисплей использовался режим 80\*25 символов, 16 цветов.

При выполнении лабораторной убедился, что:

1. Скорость передачи данных по убыванию:

1. Внутри процессора;
2. Процессор — память;
3. Процессор — устройство ввода-вывода

2. Количество используемых команд при передаче операндов по возрастанию:

1. Внутри процессора;
2. Процессор — память;
3. Процессор — устройство ввода-вывода

3. В течение одного цикла шины процессор выполняет одну законченную операцию обмена информацией с памятью или устройством ввода-вывода. Были задействованы следующие типы циклов шины: ввод; вывод; извлечение кода операции команды; чтение; запись, а следующие – не задействованы: подтверждение прерывания; останов. При взаимодействии процессора с памятью и УВВ происходила передача данных и от процессора, и к процессору.

По итогу выполнения лабораторной работы ознакомился с чтением и записью из порта ввода-вывода: для ввода из порта I/O в регистр используется команда in, для записи данных из регистра в порт I/O – команда out.

Ввод с клавиатуры и вывод на дисплей:

1) Перед вводом происходит инициализация указателей видеопамяти и приемника вводимого числа, а именно присвоение адреса базы видеопамяти (адрес 0B800h) регистру dx, смещения регистру di (верхняя левая позиция на дисплее) и инициализация дополнительного сегмента данных.

2) Для ввода с клавиатуры (получения скан-кода последней нажатой клавиши) используется команда in с обращением к порту 60h клавиатуры; для вывода на дисплей – команда mov с указанием базы и смещения адреса записи определенного символа (es:[di]).