# Лабораторная работа №1

## Структура программы на языке Assembler. Работа с программой DEBUG и пакетом TASM

### Вариант 1

Цель работы: изучить организацию ЭВМ и процессора, структуру программы на ассемблере, приобрести навыки работы с программой-отладчиком DEBUG и программами пакета TASM.

### Задание к работе

1. Ознакомиться с теоретическим материалом.
2. В соответствии со своим вариантом описать действия команд ассемблера и выполнить их в программе DEBUG (в этой работе все численные значения представлены в шестнадцатеричной системе счисления, а в остальных — в десятичной). В отчёт включить результаты ассемблирования команд, содержимое дампа памяти и трассировку программы.
3. Эти же команды представить в виде программы на ассемблере для TASM, произвести трансляцию и компоновку программы. В отчёт включить текст программы, листинг и карту распределения памяти.
4. Запустить скомпилированную программу в отладчике TD в пошаговом режиме. Сравнить содержимое регистров и флагов процессора после выполнения каждой инструкции с трассировкой этой программы в DEBUG. Сделать выводы.

### Индивидуальное задание

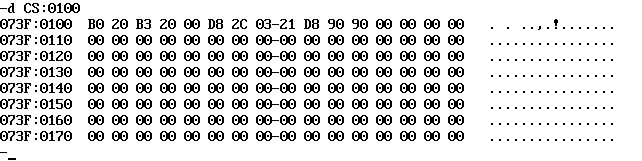
MOV AL, 20  
MOV BL, 10  
ADD AL, BL  
SUB BL, 3   
AND AX, BX

Отчёт

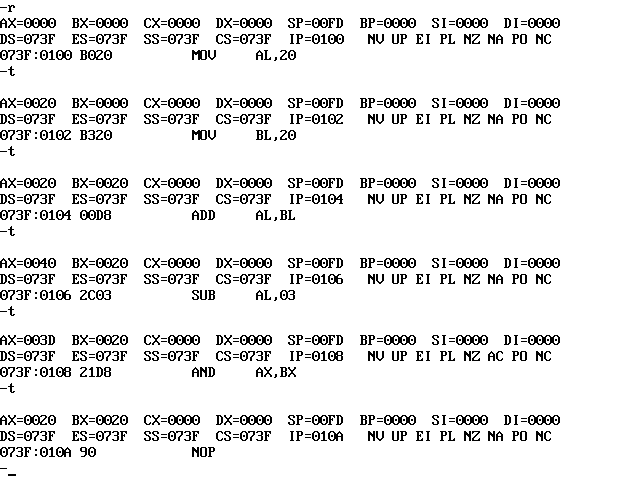
Результаты ассемблирования команд



Содержимое дампа памяти



Трассировка программы



Текст программы

|  |  |
| --- | --- |
| 01 | MASM |
| 02 | MODEL SMALL |
| 03 | .STACK 100h |
| 04 | .CODE |
| 05 | main PROC |
| 06 | MOV AL, 20h |
| 07 | MOV BL, 10h |
| 08 | ADD AL, BL |
| 09 | SUB BL, 3h |
| 10 | AND AX, BX |
| 11 |  |
| 12 | MOV AX, 4c00h |
| 13 | INT 21h |
| 14 | main ENDP |
| 15 | END main |
| 16 |  |

Листинг

|  |  |
| --- | --- |
| 01 | Turbo Assembler Version 4.1 09/13/14 02:19:45 Page 1 |
| 02 | prog.ASM |
| 03 |  |
| 04 |  |
| 05 |  |
| 06 | 1 MASM |
| 07 | 2 0000 MODEL SMALL |
| 08 | 3 0000 .STACK 100h |
| 09 | 4 0000 .CODE |
| 10 | 5 0000 main PROC |
| 11 | 6 0000 B0 20 MOV AL, 20h |
| 12 | 7 0002 B3 10 MOV BL, 10h |
| 13 | 8 0004 02 C3 ADD AL, BL |
| 14 | 9 0006 80 EB 03 SUB BL, 3h |
| 15 | 10 0009 23 C3 AND AX, BX |
| 16 | 11 |
| 17 | 12 000B B8 4C00 MOV AX, 4c00h |
| 18 | 13 000E CD 21 INT 21h |
| 19 | 14 0010 main ENDP |
| 20 | 15 END main |
| 21 | Turbo Assembler Version 4.1 09/13/14 02:19:45 Page 2 |
| 22 | Symbol Table |
| 23 |  |
| 24 |  |
| 25 |  |
| 26 |  |
| 27 | Symbol Name Type Value Cref (defined at #) |
| 28 |  |
| 29 | ??DATE Text "09/13/14" |
| 30 | ??FILENAME Text "prog " |
| 31 | ??TIME Text "02:19:45" |
| 32 | ??VERSION Number 040A |
| 33 | @32BIT Text 0 #2 |
| 34 | @CODE Text \_TEXT #2 #2 #4 |
| 35 | @CODESIZE Text 0 #2 |
| 36 | @CPU Text 0101H |
| 37 | @CURSEG Text \_TEXT #4 |
| 38 | @DATA Text DGROUP #2 |
| 39 | @DATASIZE Text 0 #2 |
| 40 | @FILENAME Text PROG |
| 41 | @INTERFACE Text 000H #2 |
| 42 | @MODEL Text 2 #2 |
| 43 | @STACK Text DGROUP #2 |
| 44 | @WORDSIZE Text 2 #4 |
| 45 | MAIN Near \_TEXT:0000 #5 15 |
| 46 |  |
| 47 | Groups & Segments Bit Size Align Combine Class Cref (defined at #) |
| 48 |  |
| 49 | DGROUP Group #2 2 |
| 50 | STACK 16 0100 Para Stack STACK #3 |
| 51 | \_DATA 16 0000 Word Public DATA #2 |
| 52 | \_TEXT 16 0010 Word Public CODE #2 2 #4 4 |
| 52 |  |

Карта распределения памяти

|  |  |
| --- | --- |
| 01 |  |
| 02 | Start Stop Length Name Class |
| 03 |  |
| 04 | 00000H 0000FH 00010H \_TEXT CODE |
| 05 | 00010H 00010H 00000H \_DATA DATA |
| 06 | 00010H 0010FH 00100H STACK STACK |
| 07 |  |
| 08 | Program entry point at 0000:0000 |
| 09 |  |

Скриншот программы

