



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Московский государственный технический университет имени  
Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ \_\_\_\_\_ «Информатика и системы управления»

КАФЕДРА \_\_\_\_\_ «Программное обеспечение ЭВМ и информационные технологии»

НАПРАВЛЕНИЕ ПОДГОТОВКИ \_\_\_\_\_ «09.03.04 Программная инженерия»

## ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

НАЗВАНИЕ \_\_\_\_\_ Прерывание INT 8h

ДИСЦИПЛИНА \_\_\_\_\_ Операционные системы

Студент \_\_\_\_\_ ИУ7-54Б \_\_\_\_\_ Цховребова Я.Р.

Группа

Подпись, дата

Фамилия И. О.

Преподаватель \_\_\_\_\_ Рязанова Н.Ю.

Подпись, дата

Фамилия И. О.

Москва, 2024 г.

# 1. Дизассемблированный код

## 1.1. Обработчик прерывания INT 8h

```
1      020C:0746 E8 0070      call     sub_2          ; (07B9)
2      020C:0746 E8 70 00      db      0E8h, 70h, 00h
3      020C:0749 06           push     es
4      020C:074A 1E           push     ds
5      020C:074B 50           push     ax
6      020C:074C 52           push     dx
7      020C:074D B8 0040      mov     ax,40h
8      020C:0750 8E D8      mov     ds,ax
9      020C:0752 33 C0      xor     ax,ax          ; Zero register
10     020C:0754 8E C0      mov     es,ax
11     020C:0756 FF 06 006C      inc     word ptr ds:[6Ch] ; (0040:006C=0F4D7h)
12     020C:075A 75 04      jnz     loc_1          ; Jump if not zero
13     020C:075C FF 06 006E      inc     word ptr ds:[6Eh] ; (0040:006E=2)
14     020C:0760      loc_1:
15     020C:0760 83 3E 006E 18      cmp     word ptr ds:[6Eh],18h ; (0040:006E=2)
16     020C:0765 75 15      jne     loc_2          ; Jump if not equal
17     020C:0767 81 3E 006C 00B0      cmp     word ptr ds:[6Ch],0B0h ; (0040:006C=0F4D7h)
18     020C:076D 75 0D      jne     loc_2          ; Jump if not equal
19     020C:076F A3 006E      mov     word ptr ds:[6Eh],ax ; (0040:006E=2)
20     020C:0772 A3 006C      mov     word ptr ds:[6Ch],ax ; (0040:006C=0F4D7h)
21     020C:0775 C6 06 0070 01      mov     byte ptr ds:[70h],1 ; (0040:0070=0)
22     020C:077A 0C 08      or      al,8
23     020C:077C      loc_2:
24     020C:077C 50           push     ax
25     020C:077D FE 0E 0040      dec     byte ptr ds:[40h] ; (0040:0040=3Dh)
26     020C:0781 75 0B      jnz     loc_3          ; Jump if not zero
27     020C:0783 80 26 003F F0      and     byte ptr ds:[3Fh],0F0h ; (0040:003F=0)
28     020C:0788 B0 0C      mov     al,0Ch
29     020C:078A BA 03F2      mov     dx,3F2h
30     020C:078D EE           out     dx,al          ; port 3F2h, dsk0 contrl output
31     020C:078E      loc_3:
32     020C:078E 58           pop      ax
33     020C:078F F7 06 0314 0004      test     word ptr ds:[314h],4 ; (0040:0314=3200h)
34     020C:0795 75 0C      jnz     loc_4          ; Jump if not zero
35     020C:0797 9F           lahf          ; Load ah from flags
36     020C:0798 86 E0      xchg     ah,al
37     020C:079A 50           push     ax
38     020C:079B 26: FF 1E 0070      call     dword ptr es:[70h] ; (0000:0070=6ADh)
39     020C:07A0 EB 03      jmp     short loc_5     ; (07A5)
40     020C:07A2 90           nop
41     020C:07A3      loc_4:
42     020C:07A3 CD 1C      int     1Ch          ; Timer break (call each 18.2ms)
43     020C:07A5      loc_5:
44     020C:07A5 E8 0011      call     sub_2          ; (07B9)
45     020C:07A8 B0 20      mov     al,20h
46     020C:07AA E6 20      out     20h,al        ; port 20h, 8259-1 int command
47     ; al = 20h, end of interrupt
48     020C:07AC 5A           pop      dx
49     020C:07AD 58           pop      ax
50     020C:07AE 1F           pop      ds
51     020C:07AF 07           pop      es
52     020C:07B0 E9 FE99      jmp     $-164h
53     ;; <...>
54     020C:064C      loc_5:
55     020C:064C 1E           push     ds
56     020C:064D 50           push     ax
57     ;; <...>
58     020C:06AA 58           pop      ax
59     020C:06AB 1F           pop      ds
60     ;; <...>
61     020C:06AC CF           iret          ; Interrupt return
```

Листинг 1. Листинг прерывания INT 8h

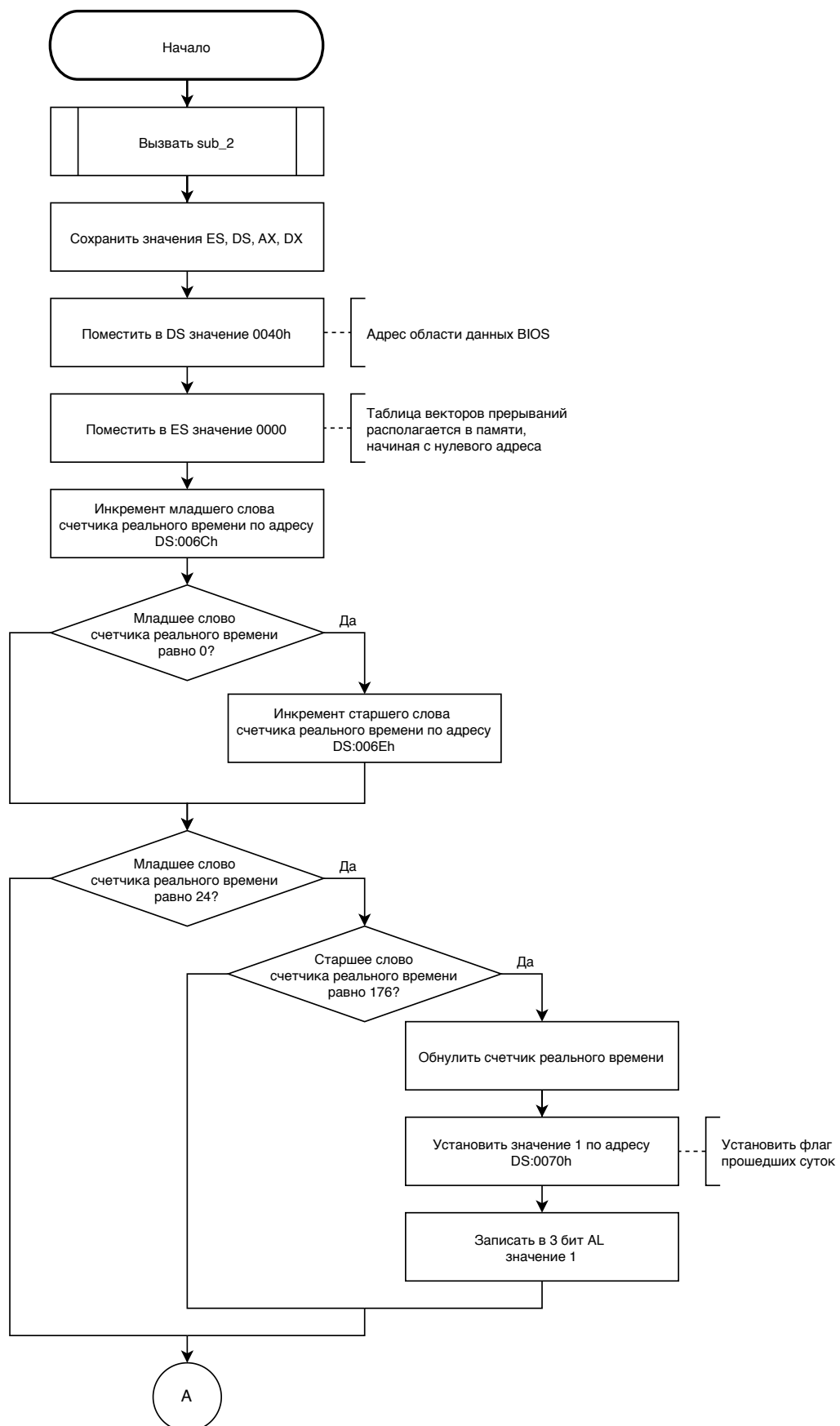
## 1.2. Процедура sub\_2

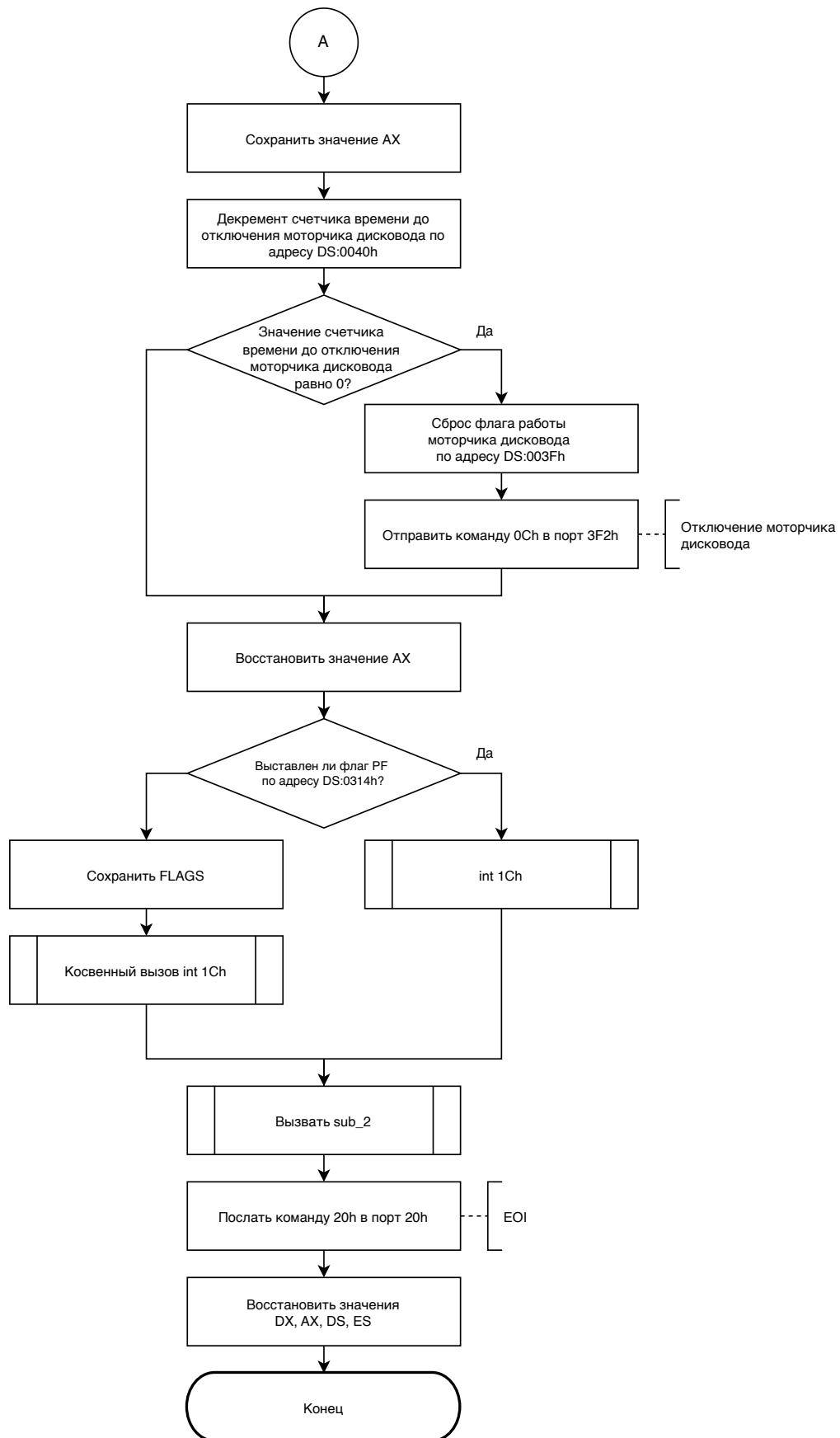
```
1      020C:07B9      sub_2      proc      near
2      020C:07B9  1E      push      ds
3      020C:07BA  50      push      ax
4      020C:07BB  B8 0040  mov ax,40h
5      020C:07BE  8E D8    mov ds,ax
6      020C:07C0  9F      lahf                      ; Load ah from flags
7      020C:07C1  F7 06 0314 2400    test     word ptr ds:[314h],2400h ; (0040:0314=3200h)
8      020C:07C7  75 0C      jnz loc_7      ; Jump if not zero
9      020C:07C9  F0> 81 26 0314 FDFF    lock and word ptr ds:[314h],0
10     020C:07D0      loc_6:
11     020C:07D0  9E      sahf                      ; Store ah into flags
12     020C:07D1  58      pop ax
13     020C:07D2  1F      pop ds
14     020C:07D3  EB 03    jmp short loc_8      ; (07D8)
15     020C:07D5      loc_7:
16     020C:07D5  FA      cli                      ; Disable interrupts
17     020C:07D6  EB F8    jmp short loc_6      ; (07D0)
18     020C:07D8      loc_8:
19     020C:07D8  C3      retn
20     sub_2      endp
```

Листинг 2. Листинг процедуры sub\_2

## 2. Схемы алгоритмов

### 2.1. Схема алгоритма обработчика INT 8h





## 2.2. Схема алгоритма процедуры sub\_2

