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

ФАКУЛЬТЕТ ИУ «Информатика и системы управления»

КАФЕДРА ИУ-7 «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа №1 по дисциплине «Операционные системы»

Тема: Дизассемблирование INT 8h

Студент: Куликов Н. В.

Группа: ИУ7-53Б

Преподаватель: Рязанова Н. Ю.

Москва, 2025 г.

1. Дизассемблированный код

Листинг 1. Обработчик прерывания INT 8h.

```
          Temp.lst                        Sourcer     v5.10    8-Sep-25  12:23 am
Page 1

020C:0746  E8 0070            ;*          call  sub_1             ; (07B9)
020C:0746  E8 70 00                       db    0E8h, 70h, 00h
020C:0749  06                             push  es
020C:074A  1E                             push  ds
020C:074B  50                             push  ax
020C:074C  52                             push  dx
020C:074D  B8 0040                        mov   ax,40h
020C:0750  8E D8                    mov   ds,ax
020C:0752  33 C0                    xor   ax,ax             ; Zero register
020C:0754  8E C0                    mov   es,ax
020C:0756  FF 06 006C                     inc   word ptr ds:[6Ch] ; (0040:006C=62F0h)
020C:075A  75 04                    jnz   loc_1             ; Jump if not zero
020C:075C  FF 06 006E                     inc   word ptr ds:[6Eh] ; (0040:006E=0)
020C:0760                loc_1:
020C:0760  83 3E 006E 18                  cmp   word ptr ds:[6Eh],18h   ; (0040:006E=0)
020C:0765  75 15                    jne   loc_2             ; Jump if not equal
020C:0767  81 3E 006C 00B0                cmp   word ptr ds:[6Ch],0B0h  ;
(0040:006C=62F0h)
020C:076D  75 0D                    jne   loc_2             ; Jump if not equal
020C:076F  A3 006E                        mov   word ptr ds:[6Eh],ax    ; (0040:006E=0)
020C:0772  A3 006C                        mov   word ptr ds:[6Ch],ax    ;
(0040:006C=62F0h)
020C:0775  C6 06 0070 01                  mov   byte ptr ds:[70h],1     ; (0040:0070=0)
020C:077A  0C 08                    or    al,8
020C:077C                loc_2:
020C:077C  50                             push  ax
020C:077D  FE 0E 0040                     dec   byte ptr ds:[40h] ; (0040:0040=0DFh)
020C:0781  75 0B                    jnz   loc_3             ; Jump if not zero
020C:0783  80 26 003F F0                  and   byte ptr ds:[3Fh],0F0h  ; (0040:003F=0)
020C:0788  B0 0C                    mov   al,0Ch
020C:078A  BA 03F2                        mov   dx,3F2h
020C:078D  EE                             out   dx,al             ; port 3F2h, dsk0
contrl output
020C:078E                loc_3:
020C:078E  58                             pop   ax
020C:078F  F7 06 0314 0004                test  word ptr ds:[314h],4    ;
(0040:0314=3200h)
020C:0795  75 0C                    jnz   loc_4             ; Jump if not zero
020C:0797  9F                             lahf                    ; Load ah from flags
020C:0798  86 E0                    xchg  ah,al
020C:079A  50                             push  ax
020C:079B  26: FF 1E 0070                 call  dword ptr es:[70h]      ;
(0000:0070=6ADh)
020C:07A0  EB 03                    jmp   short loc_5       ; (07A5)
020C:07A2  90                             nop
020C:07A3                loc_4:
020C:07A3  CD 1C                    int   1Ch               ; Timer break (call each
18.2ms)
020C:07A5                loc_5:
020C:07A5  E8 0011                        call  sub_1             ; (07B9)
020C:07A8  B0 20                    mov   al,20h                ; ' '
020C:07AA  E6 20                    out   20h,al                ; port 20h, 8259-1
int command
```

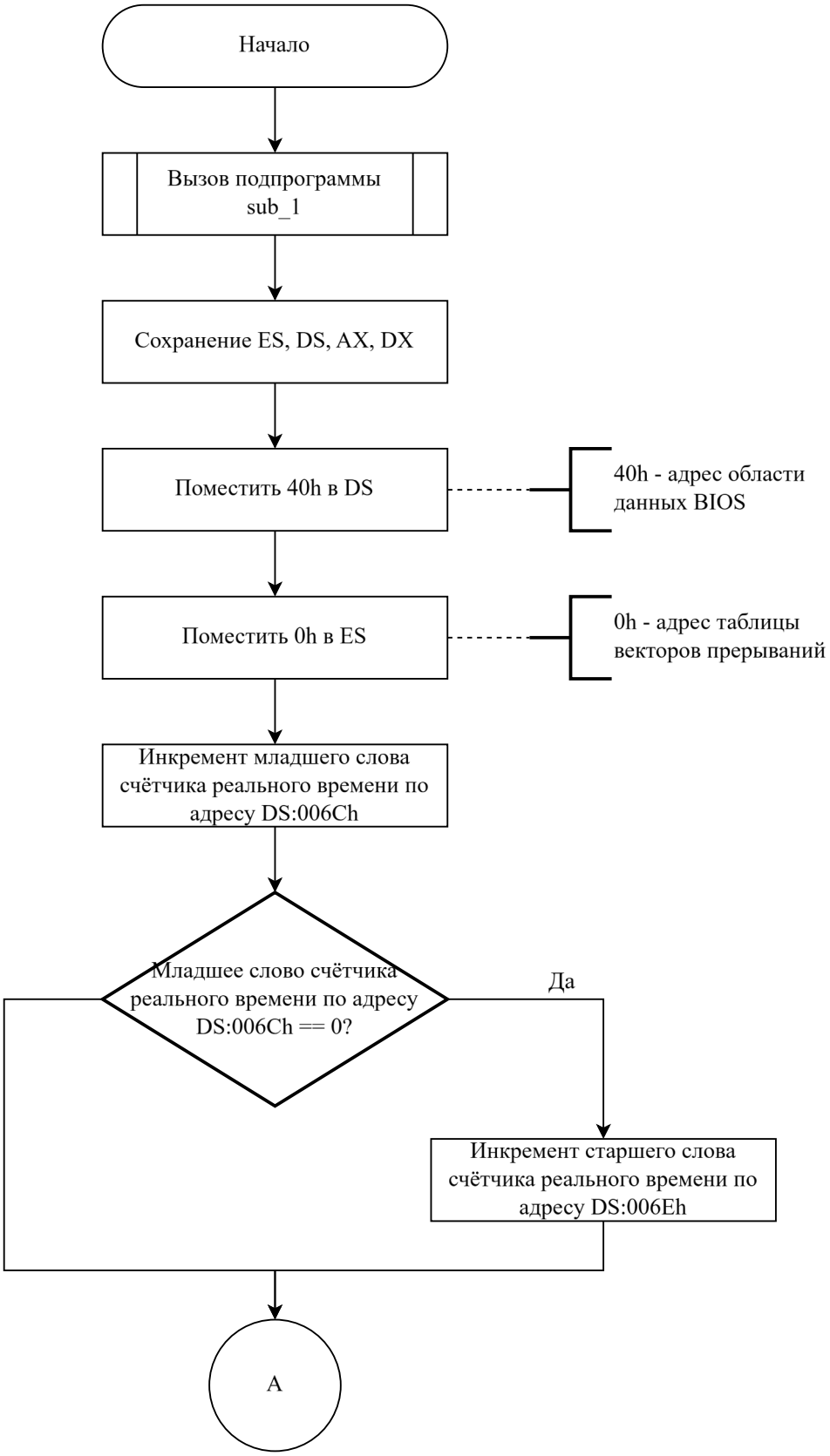
```
;  al = 20h, end of interrupt
020C:07AC  5A                              pop   dx
020C:07AD  58                              pop   ax
020C:07AE  1F                              pop   ds
020C:07AF  07                              pop   es
020C:07B0  E9 FE99                         jmp   $-164h
020C:07B3  C4                              db    0C4h
                                                        ;* No entry point to code
020C:07B4  C4 0E 93E9                      les   cx,dword ptr ds:[93E9h] ;
(0000:93E9=8B52h) Load 32 bit ptr
020C:07B8  FE                              db    0FEh

020C:064C  1E                  ;*          push  ds
020C:064C  1E                              db    1Eh
020C:064D  50                              push  ax

020C:0689  58                  ;*          pop   ax
020C:0689  58                              db    58h
020C:068A  1F                              pop     ds

020C:06AC  CF                  ;*          iret
```

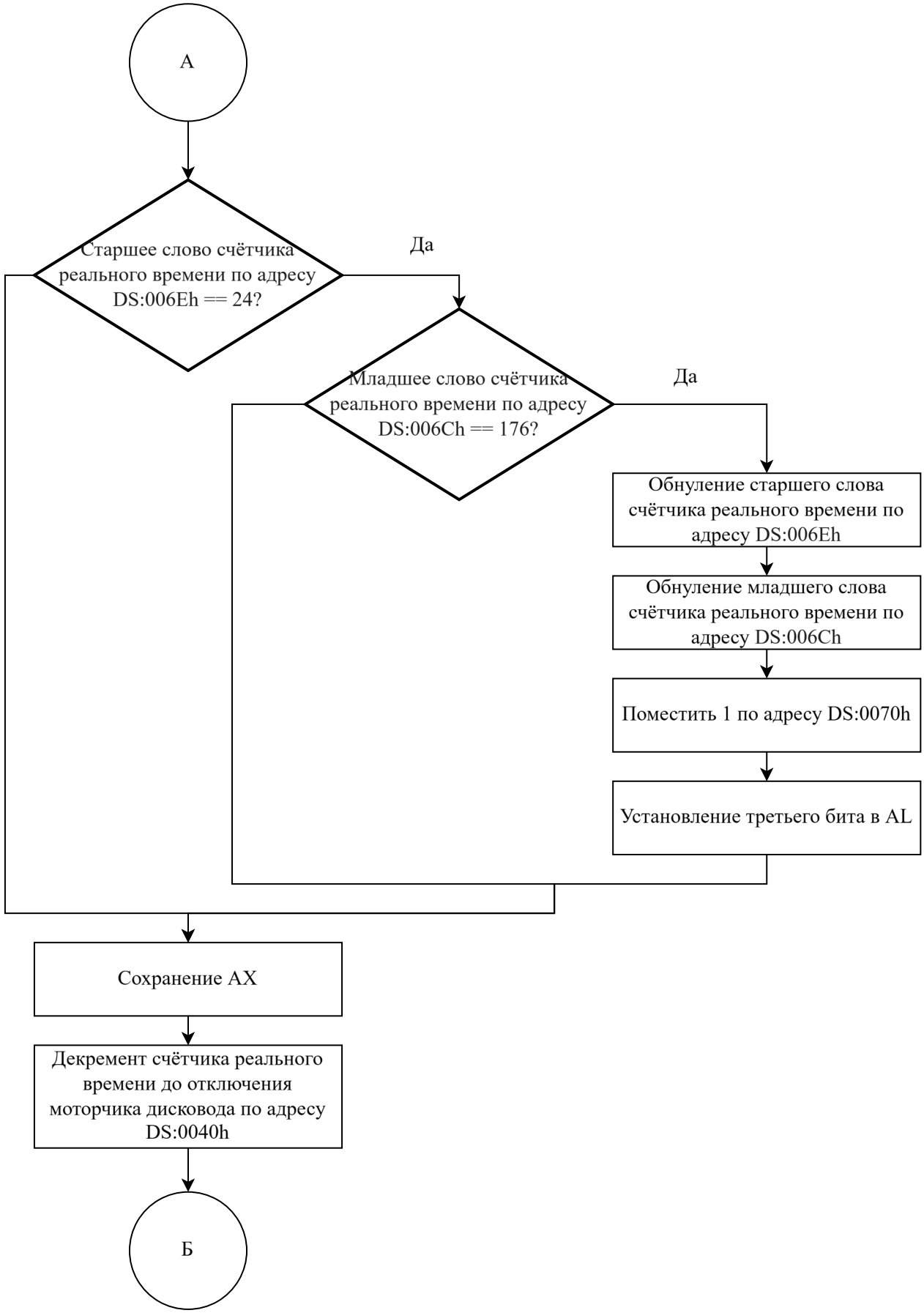
Листинг 2. Процедура sub_1.

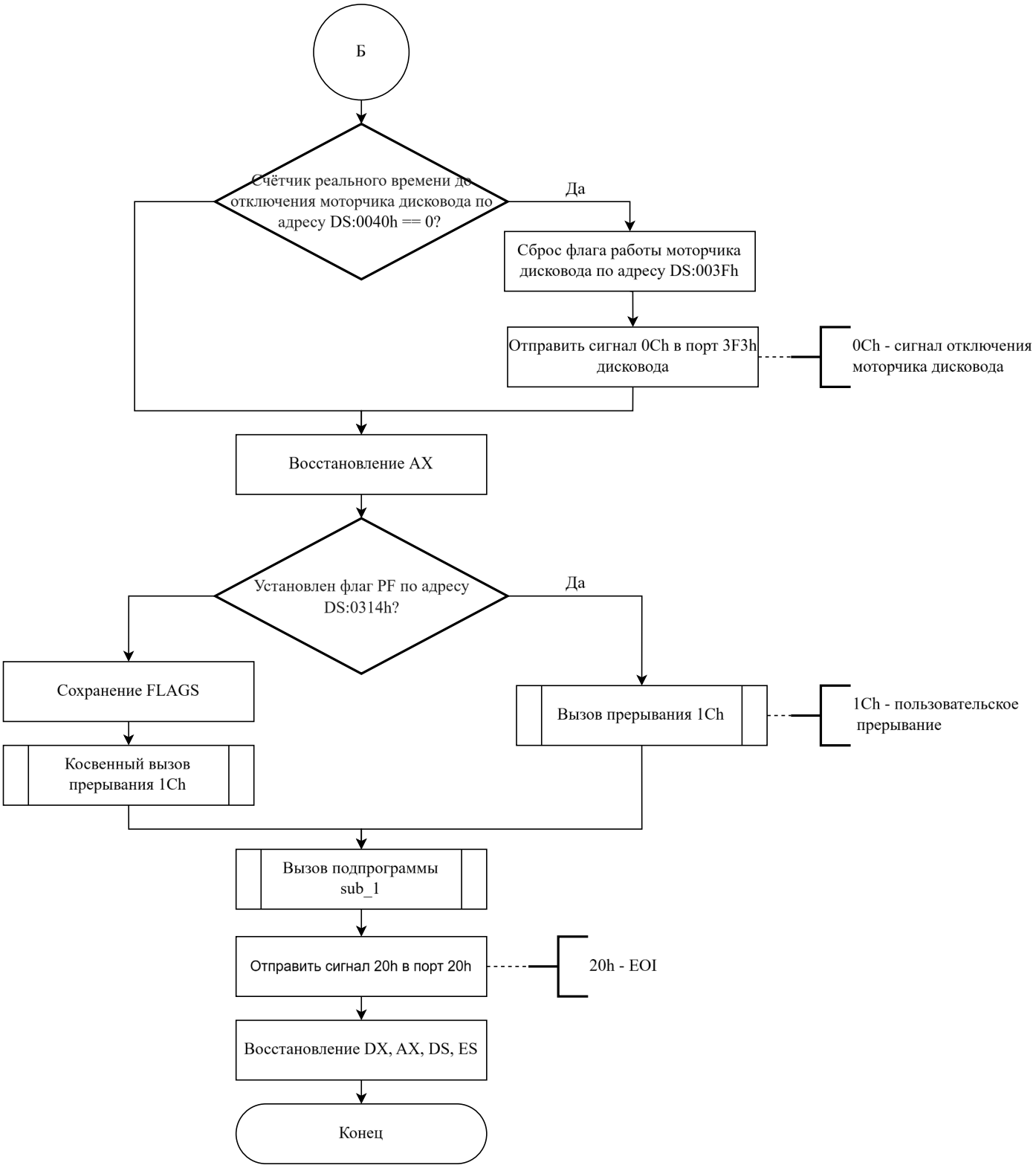
```
                sub_1       proc  near
020C:07B9  1E                              push  ds
020C:07BA  50                              push  ax
020C:07BB  B8 0040                         mov   ax,40h
020C:07BE  8E D8                   mov     ds,ax
020C:07C0  9F                              lahf                    ; Load ah from flags
020C:07C1  F7 06 0314 2400                 test  word ptr ds:[314h],2400h      ;
(0040:0314=3200h)
020C:07C7  75 0C                   jnz     loc_7               ; Jump if not zero
020C:07C9  F0> 81 26 0314 FDFF                                    lock     and   word ptr
ds:[314h],0FDFFh  ; (0040:0314=3200h)
020C:07D0                loc_6:
020C:07D0  9E                              sahf                    ; Store ah into flags
020C:07D1  58                              pop   ax
020C:07D2  1F                              pop   ds
020C:07D3  EB 03                   jmp     short loc_8         ; (07D8)
020C:07D5                loc_7:
020C:07D5  FA                              cli                     ; Disable interrupts
020C:07D6  EB F8                   jmp     short loc_6         ; (07D0)
020C:07D8                loc_8:
020C:07D8  C3                              retn
                sub_1       endp
```

2. Схема алгоритмов

2.1. Схема алгоритма работы INT 8h




А
Старшее слово счётчика реального времени по адресу DS:006Eh == 24?
Да
Младшее слово счётчика реального времени по адресу DS:006Ch == 176?
Да
Обнуление старшего слова счётчика реального времени по адресу DS:006Eh
Обнуление младшего слова счётчика реального времени по адресу DS:006Ch
Поместить 1 по адресу DS:0070h
Установление третьего бита в AL
Сохранение AX
Декремент счётчика реального времени до отключения моторчика дисковода по адресу DS:0040h
Б


Б
Счётчик реального времени до отключения моторчика дисковода по адресу DS:0040h == 0?
Да
Сброс флага работы моторчика дисковода по адресу DS:003Fh
Отправить сигнал 0Ch в порт 3F3h дисковода
0Ch - сигнал отключения моторчика дисковода
Восстановление AX
Установлен флаг PF по адресу DS:0314h?
Да
Сохранение FLAGS
Косвенный вызов прерывания 1Ch
Вызов прерывания 1Ch
1Ch - пользовательское прерывание
Вызов подпрограммы sub_1
Отправить сигнал 20h в порт 20h
20h - EOI
Восстановление DX, AX, DS, ES
Конец

2.2. Схема алгоритма работы процедуры sub_1.