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»

Лабораторная работа №1
по дисциплине "Операционные системы"

Тема Дизассемблирование INT 8h

Студент Романов А.В.

Группа ИУ7-53Б

Преподаватель Рязанова Н.Ю.

Москва — 2020 г.

1. Полученный дизассемблированный код

1.1. Листинг INT8h

```
1     ; -- Вызов sub_1
2     020A:0746  E8 0070              call    sub_1; (07B9)
3     ; -- Сохранение регистров es, ds, ax, dx
4     020A:0749  06                   push    es
5     020A:074A  1E                   push    ds
6     020A:074B  50                   push    ax
7     020A:074C  52                   push    dx
8     ; -- Загрузка в DS 0040H
9     020A:074D  B8 0040              mov ax,40h
10    020A:0750  8E D8                mov ds,ax
11    ; -- AX = ES = 0
12    020A:0752  33 C0                xor ax,ax; Zero register
13    020A:0754  8E C0                mov es,ax
14    ; -- Инкрементирование счётчика таймера, по адресу 0040:006C
15    020A:0756  FF 06 006C           inc word ptr ds:[6Ch; (0040:006C=5AEBh)
16    020A:075A  75 04                jnz loc_1; Jump if not zero
17    ; -- Инкрементирование старших 2 байта счётчика таймера
18    020A:075C  FF 06 006E           inc word ptr ds:[6Eh]; (0040:006E=2)
19    ; -- Проверка, что прошло 24 часа:
20    ;    0040H:006EH == 18H и 0040H:006CH == 00B0H
21    ;    24 * 60 * 60 * t == 18H << 16 + B0H, t - количество вызовов таймера в с
         екунду
22    020A:0760               loc_1:
23    020A:0760  83 3E 006E 18        cmp word ptr ds:[6Eh],18h; (0040:006E=2)
24    020A:0765  75 15                jne loc_2; Jump if not equal
25    020A:0767  81 3E 006C 00B0      cmp word ptr ds:[6Ch],0B0h; (0040:006C=5AEBh
         )
26    020A:076D  75 0D                jne loc_2; Jump if not equal
27    ; -- Зануление счётчика таймера, и занесение 1 в 0040H:0070H по прошествии
         24 часов.
28    020A:076F  A3 006E              mov word ptr ds:[6Eh],ax    ; (0040:006E=2)
29    020A:0772  A3 006C              mov word ptr ds:[6Ch],ax    ; (0040:006C=5
         AEBh)
30    020A:0775  C6 06 0070 01        mov byte ptr ds:[70h],1 ; (0040:0070=0)
31    ; -- AL = 8, т.к. до этой строчки AL = 0.
32    020A:077A  0C 08                or  al,8
33    020A:077C               loc_2:
34    ; -- Сохранение регистра AX
35    020A:077C  50                   push    ax
36    ; -- Декрементирование счетчика отключения моторчика
37    020A:077D  FE 0E 0040           dec byte ptr ds:[40h]; (0040:0040=0F7h)
38    020A:0781  75 0B                jnz loc_3; Jump if not zero
39    ; -- Установка флагов, отвечающих за отключение моторчика дисковода
40    020A:0783  80 26 003F F0        and byte ptr ds:[3Fh],0F0h; (0040:003F=0)
41    ; -- Отправить команду отключения моторчика дисковода
42    020A:0788  B0 0C                mov al,0Ch
43    020A:078A  BA 03F2              mov dx,3F2h
44    020A:078D  EE                   out dx,al; port 3F2h, dsk0 contrl output
45    020A:078E               loc_3:
46    ; -- Восстановление регистра AX
47    020A:078E  58                   pop ax
48    ; -- Проверка 2 бита (Parity Flag)
49    020A:078F  F7 06 0314 0004      test    word ptr ds:[314h],4;
         (0040:0314=3200h)
```

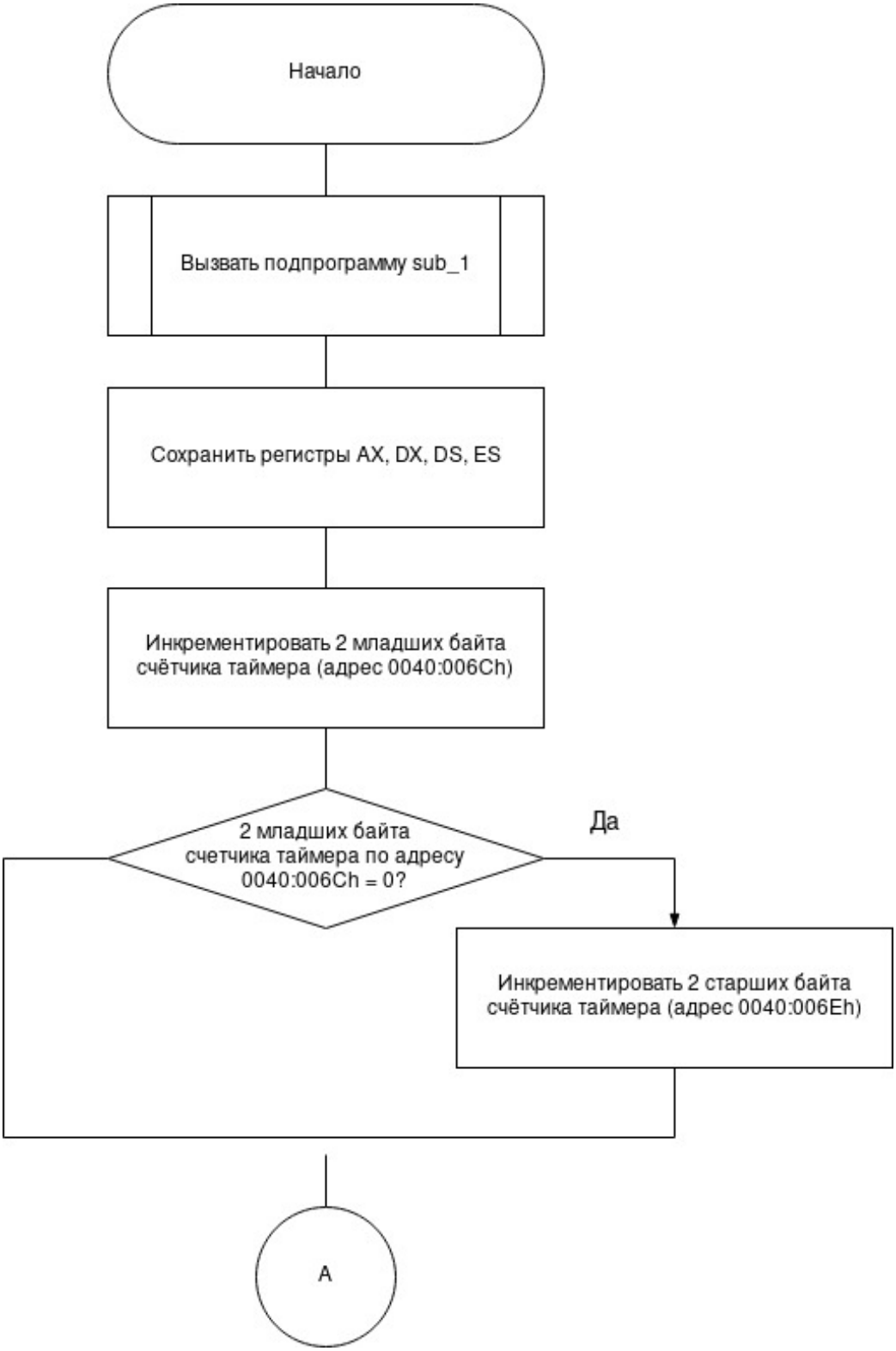
```
020A:0795  75 0C                jnz loc_4; Jump if not zero
; -- Загрузка младшего байта FLAGS в регистр AH
020A:0797  9F                   lahf; Load ah from flags
; -- Обмен ah и al. Таким образом: AX = 08[AH], [AH] - младший байт регистра FALGS
020A:0798  86 E0                xchg    ah,al
; -- Сохранение регистра AX
020A:079A  50                   push    ax
; -- Вызов 1CH с помощью адреса в таблице векторов.
;    При вызове через int произошел бы пуш регистра FLAGS в стек,
;    а в случае через call на его месте будет лежать положенный до AX,
;    который по выходу из 1CH будет установлен в FLAGS с помощью iret.
020A:079B  26: FF 1E 0070       call dword ptr es:[70h]
020A:07A0  EB 03                jmp short loc_5; (07A5)
020A:07A2  90                   nop
020A:07A3                  loc_4:
020A:07A3  CD 1C                int 1Ch; Timer break (call each 18.2ms)
020A:07A5                  loc_5:
020A:07A5  E8 0011              call    sub_1; (07B9)
; -- Сброс контроллера прерываний.
020A:07A8  B0 20                mov al,20h
020A:07AA  E6 20                out 20h,al; port 20h, 8259-1 int command
; -- Востановление регистров dx, ax, ds, es
020A:07AC  5A                   pop dx
020A:07AD  58                   pop ax
020A:07AE  1F                   pop ds
020A:07AF  07                   pop es
020A:07B0  E9 FE99              jmp $-164h ; (020A:07B0h - 164h = 020A:064Ch)
; ..........
020A:064C  1E                   push    ds
020A:064D  50                   push    ax
; ...........
020A:06AA  58                   pop ax
020A:06AB  1F                   pop ds
; -- Возврат из прерывания
020A:06AC  CF                   iret; Interrupt return
```

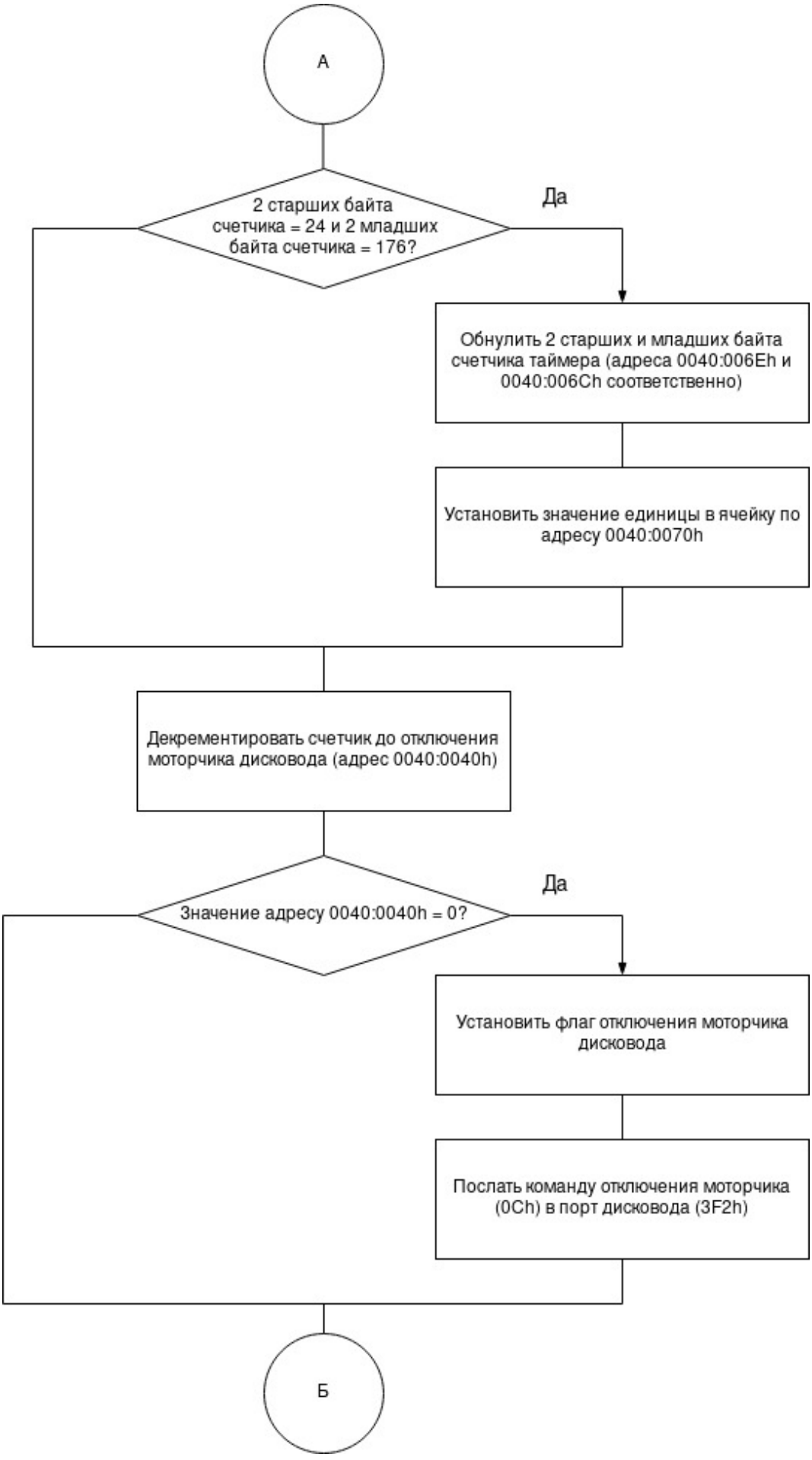
1.2. Листинг процедуры sub_1

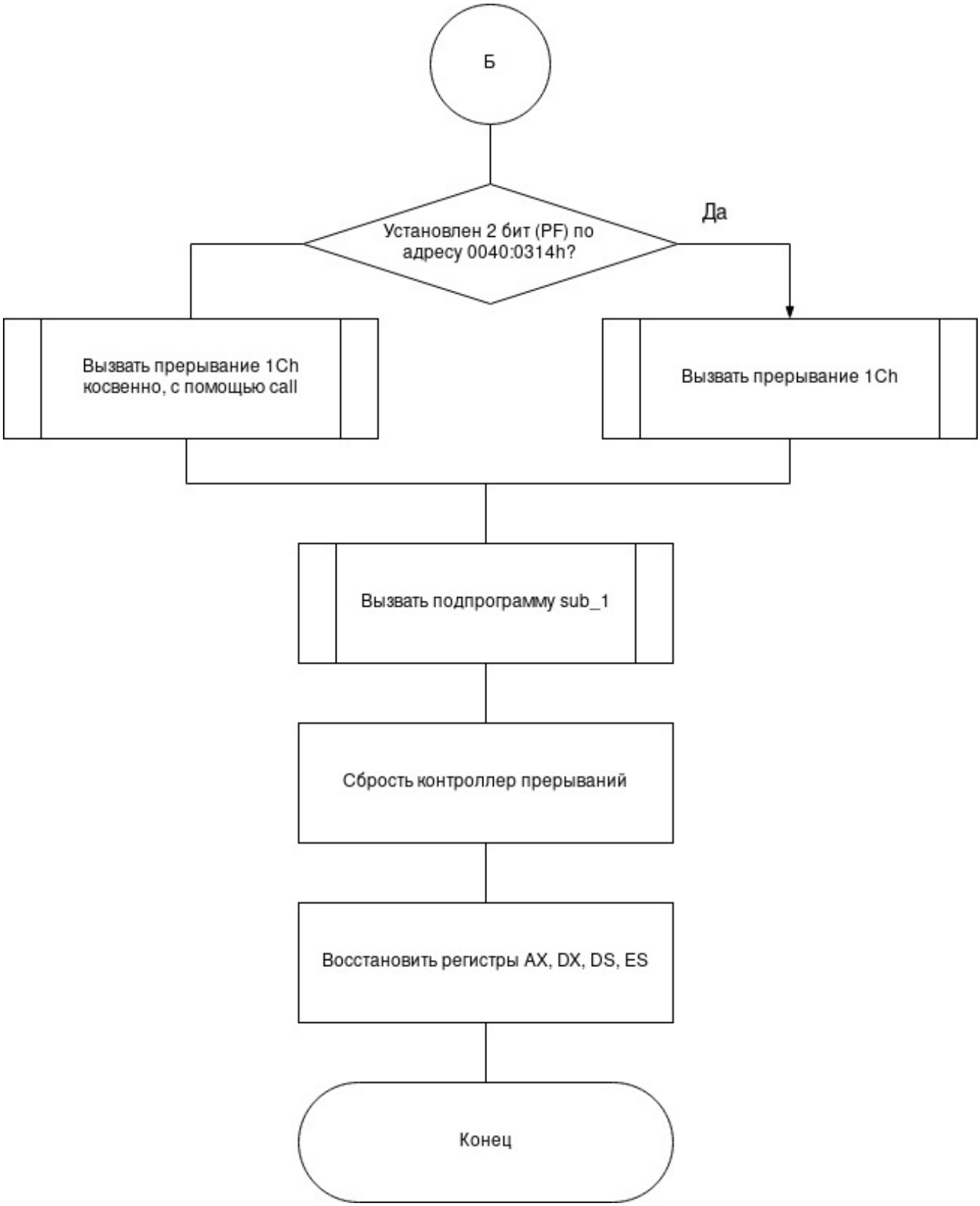
```
1   sub_1       proc    near
2   ; Сохранение регистров dx, ax
3   020A:07B9  1E                   push    ds
4   020A:07BA  50                   push    ax
5   ; -- AX = DS = 0040H
6   020A:07BB  B8 0040              mov ax,40h
7   020A:07BE  8E D8                mov ds,ax
8   ; -- Сохранение младшего байта FLAGS в AH
9   020A:07C0  9F                   lahf; Load ah from flags
10  ; -- Проверка флага DF или старшего бита IOPL.
11  ;    Если кто-то установлен хотя бы один, то IF сбрасывается через cli
12  020A:07C1  F7 06 0314 2400      test    word ptr ds:[314h],2400h    ;
        (0040:0314=3200h)
13  020A:07C7  75 0C                jnz loc_7; Jump if not zero
14  ; -- Сброс Interrput Enable Flag (9 бит - занулить).
15  ;    lock для того чтобы команда была неделимой.
16  020A:07C9  F0> 81 26 0314 FDFF  lock and word ptr ds:[314h],0FDFFh  ;
        (0040:0314=3200h)
17  020A:07D0               loc_6:
18  ; -- Загрузка AH в  младший байт FLAGS.
19  020A:07D0  9E                   sahf; Store ah into flags
20  020A:07D1  58                   pop ax
21  020A:07D2  1F                   pop ds
22  020A:07D3  EB 03                jmp short loc_8; (07D8)
23  020A:07D5               loc_7:
24  ; -- Сброс Interrput Enable Flag с помощью cli.
25  020A:07D5  FA                   cli; Disable interrupts
26  020A:07D6  EB F8                jmp short loc_6; (07D0)
27  020A:07D8               loc_8:
28  ; -- Возврат из подпрограммы.
29  020A:07D8  C3                   retn
30  sub_1       endp
```

2. Схема алгоритмов

2.1. Схема алгоритма обработчика INT8h




А
2 старших байта счетчика = 24 и 2 младших байта счетчика = 176?
Да
Обнулить 2 старших и младших байта счетчика таймера (адреса 0040:006Eh и 0040:006Ch соответственно)
Установить значение единицы в ячейку по адресу 0040:0070h
Декрементировать счетчик до отключения моторчика дисковода (адрес 0040:0040h)
Значение адресу 0040:0040h = 0?
Да
Установить флаг отключения моторчика дисковода
Послать команду отключения моторчика (0Ch) в порт дисковода (3F2h)
Б


Б
Установлен 2 бит (PF) по адресу 0040:0314h?
Да
Вызвать прерывание 1Ch косвенно, с помощью call
Вызвать прерывание 1Ch
Вызвать подпрограмму sub_1
Сбрость контроллер прерываний
Восстановить регистры AX, DX, DS, ES
Конец

2.2. Схема алгоритма процедуры sub_1