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 (часть 1) по дисциплине "Операционные системы"

Тема Дизассемблирование INT 8h

Студент Сапожков А. М.

Группа ИУ7-53Б

Преподаватель Рязанова Н. Ю.

Москва — 2022 г.

1. Полученный дизассемблированный код

1.1. Листинг обработчика прерывания INT 8h

```
; Вызов подпрограммы sub_3:
020A:0746  E8 0070              call    sub_3                   ; (07B9)
; Сохранение значений регистров es, ds, ax, dx:
020A:0749  06                   push    es
020A:074A  1E                   push    ds
020A:074B  50                   push    ax
020A:074C  52                   push    dx
; Загрузка сегментных регистров ds, es:
; (40h - сегментная часть адреса области данных BIOS)
020A:074D  B8 0040              mov ax,40h
020A:0750  8E D8                mov ds,ax
020A:0752  33 C0                xor ax,ax
020A:0754  8E C0                mov es,ax
; Инкремент значений счётчиков таймера:
; 0040:006C, 0040:006E - адреса младшего и старшего слова
; счётчика прерываний таймера BIOS
020A:0756  FF 06 006C           inc word ptr ds:[6Ch]           ; (0040:006C=9A82h)
020A:075A  75 04                jnz loc_2                       ; Jump if not zero
020A:075C  FF 06 006E           inc word ptr ds:[6Eh]           ; (0040:006E=0)
; Сброс счётчиков времени при наступлении нового дня:
; 0040:006E == 18h (24), 0040:006C == B0h (176)
; 18h << 16 + B0h == 86400 * c;
; c = 1573040 / 86400 = 18.2... - количество срабатываний таймера в секунду
020A:0760           loc_2:
020A:0760  83 3E 006E 18        cmp word ptr ds:[6Eh],18h       ; (0040:006E=0)
020A:0765  75 15                jne loc_3                       ; Jump if not equal
020A:0767  81 3E 006C 00B0      cmp word ptr ds:[6Ch],0B0h      ; (0040:006C=9A82h)
020A:076D  75 0D                jne loc_3                       ; Jump if not equal
020A:076F  A3 006E              mov word ptr ds:[6Eh],ax        ; (0040:006E=0)
020A:0772  A3 006C              mov word ptr ds:[6Ch],ax        ; (0040:006C=9A82h)
; Установка флага наращивания даты (начальное значение - 0)
020A:0775  C6 06 0070 01        mov byte ptr ds:[70h],1         ; (0040:0070=0)
; Установка al = 8:
020A:077A  0C 08                or  al,8
020A:077C           loc_3:
; Сохранение значения регистра ax:
020A:077C  50                   push    ax
; Декремент значения счётчика времени до отключения моторчика дисковода:
; (0040:0040 - адрес счётчика времени в области данных накопителя FDD)
020A:077D  FE 0E 0040           dec byte ptr ds:[40h]           ; (0040:0040=39h)
020A:0781  75 0B                jnz loc_4                       ; Jump if not zero
; Установка флагов, отвечающих за отключение моторчикка дисковода:
020A:0783  80 26 003F F0        and byte ptr ds:[3Fh],0F0h      ; (0040:003F=0)
; Отправка команды отключения моторчика дисковода:
020A:0788  B0 0C                mov al,0Ch
020A:078A  BA 03F2              mov dx,3F2h
020A:078D  EE                   out dx,al                       ; port 3F2h, dsk0
    contrl output
020A:078E           loc_4:
; Восстановление значения регистра ax:
020A:078E  58                   pop ax
; Проверка второго бита (Parity Flag - флаг чётности):
; 0040:0314h - адрес области данных BIOS, содержащей копию флагов
```

```
020A:078F  F7 06 0314 0004      test    word ptr ds:[314h],4    ;
    (0040:0314=3200h)
020A:0795  75 0C                jnz loc_5                       ; Jump if not zero
; Сохранение младшего байта регистра FLAGS в AH:
020A:0797  9F                   lahf                            ; Load ah from flags
; Обмен значений регистров ah и al:
; Теперь младший байт регистра FLAGS находится в младшем байте регистра ax
020A:0798  86 E0                xchg    ah,al
; Сохранение регистра ax:
020A:079A  50                   push    ax
; Косвенный вызов пользовательского прерывания по адресу в таблице векторов прер
    ываний:
; В этом случае не произойдёт push регистра FLAGS, на его месте будет AX,
; который восстановится в регистр FLAGS после выхода из обработчика прерывания
020A:079B  26: FF 1E 0070       call    dword ptr es:[70h]  ; (0000:0070=6ADh)
020A:07A0  EB 03                jmp short loc_6             ; (07A5)
020A:07A2  90                   nop
; Вызов пользовательского прерывания через int 1Ch:
020A:07A3           loc_5:
020A:07A3  CD 1C                int 1Ch                     ; Timer break (call
    each 18.2ms)
020A:07A5           loc_6:
; Вызов подпрограммы sub_3, чтобы запретить вызов int 8h:
020A:07A5  E8 0011              call    sub_3               ; (07B9)
; Сброс контроллера прерываний (отправка команды End Of Interrupt):
; Разрешение обработки всех прерываний
020A:07A8  B0 20                mov al,20h      ; ' '
020A:07AA  E6 20                out 20h,al      ; port 20h, 8259-1 int command
                                                ; al = 20h, end of interrupt
; Восстановление значений регистров es, ds, ax, dx:
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
; ...
; Возврат из прерывания
020A:06AC  CF                   iret                        ; Interrupt return
```

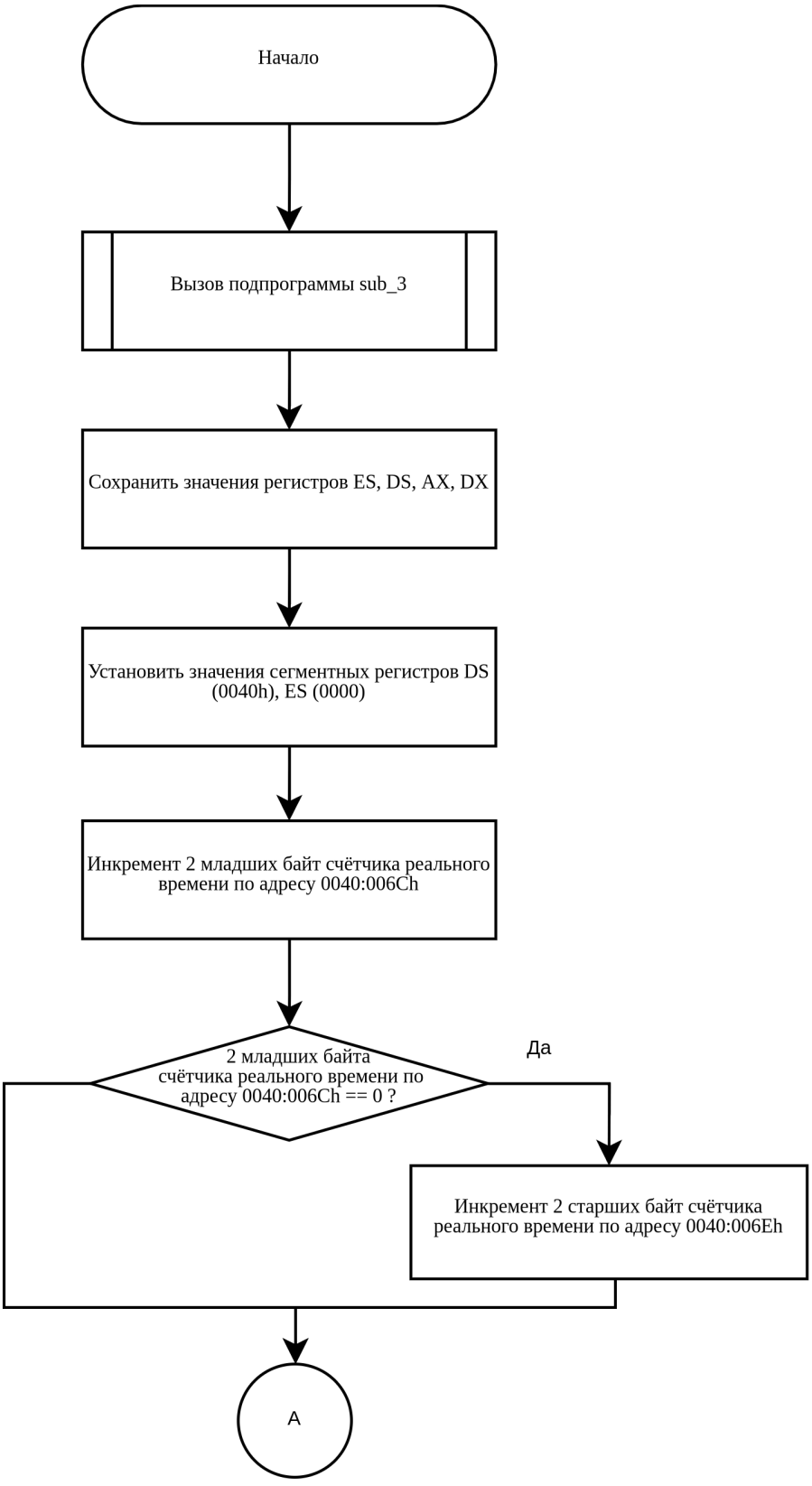
1.2. Листинг процедуры sub_3

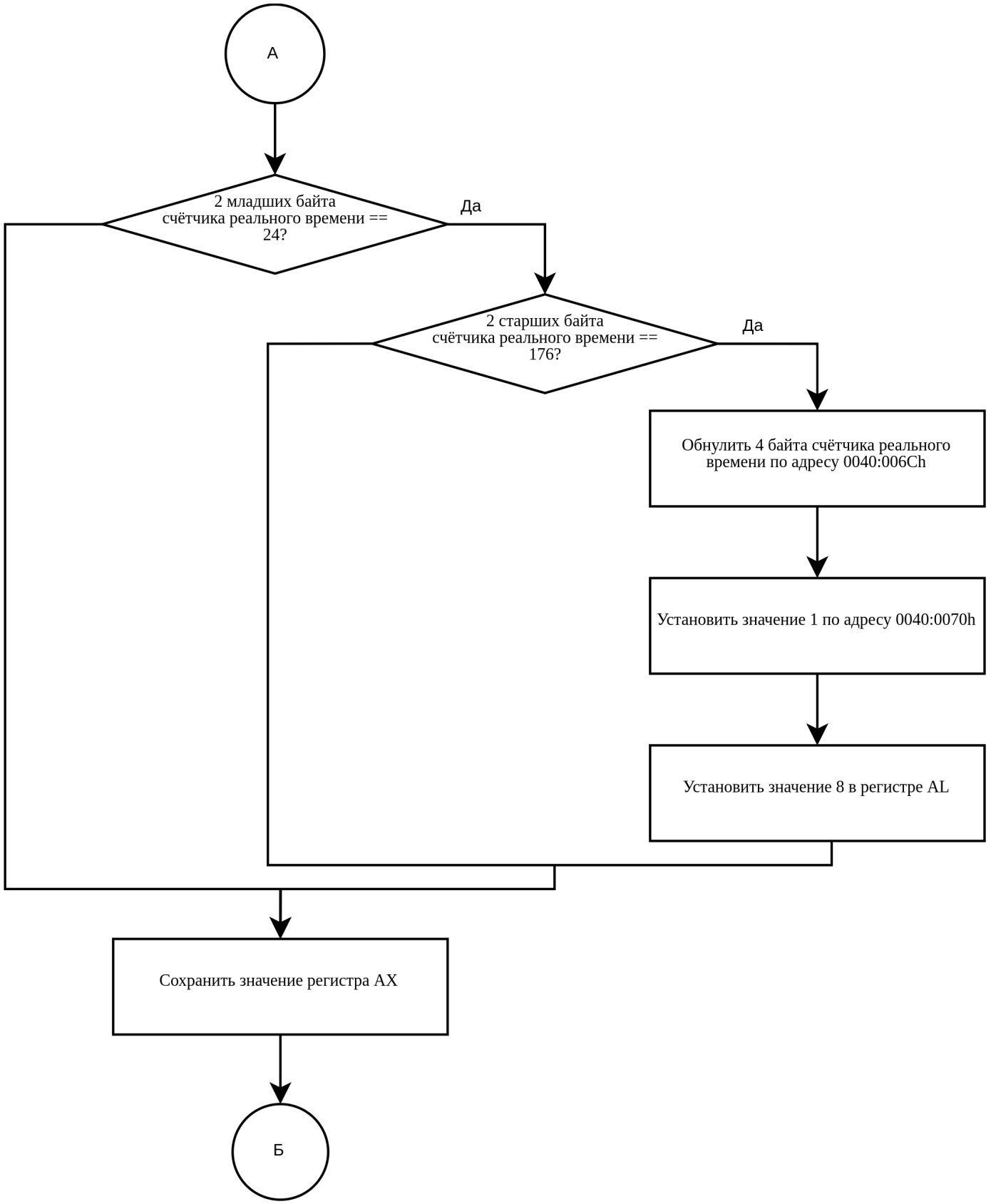
```
020A:07B9        sub_3      proc    near
; Сохранение значений регистров ds, ax:
020A:07B9  1E               push    ds
020A:07BA  50               push    ax
; Загрузка сегментного регистра ds:
020A:07BB  B8 0040          mov ax,40h
020A:07BE  8E D8            mov ds,ax
; Запись младшего байта регистра FLAGS в AH:
020A:07C0  9F               lahf                                ; Load ah
    from flags
; Проверка DF и старшего бита IOPL по адресу 0040:0314h:
020A:07C1  F7 06 0314 2400  test    word ptr ds:[314h],2400h    ;
    (0040:0314=3200h)
020A:07C7  75 0C            jnz loc_8                           ; Jump if
    not zero
```

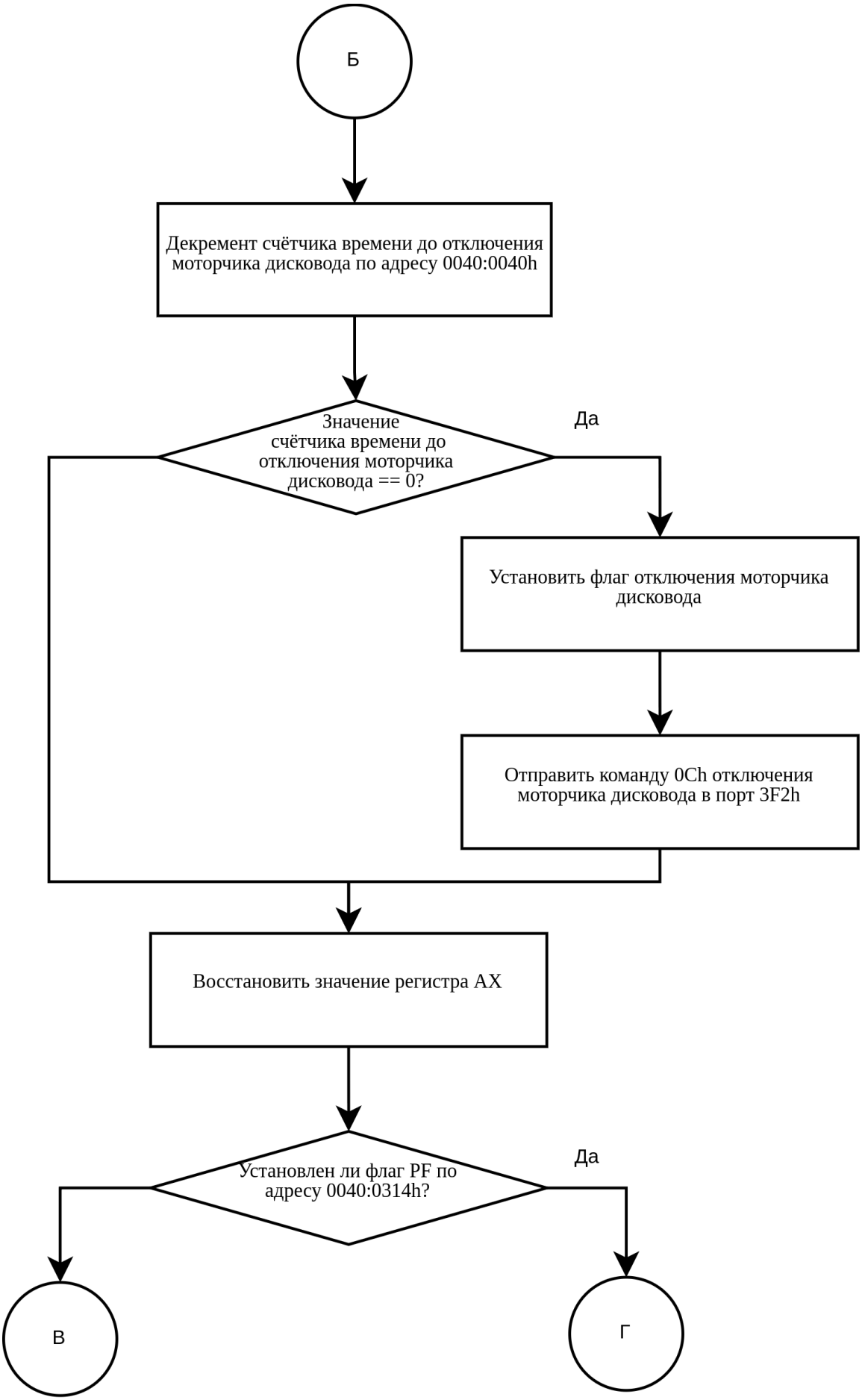
```
13 ; Установка 9 бита в 0 - сброс IF (запрет прерываний):
14 020A:07C9  F0> 81 26 0314 FDFF  lock and word ptr ds:[314h],0FDFFh  ;
       (0040:0314=3200h)
15 020A:07D0         loc_7:
16 ; Запись регистра AH в младший байт FLAGS:
17 020A:07D0  9E                   sahf                                 ; Store ah
       into flags
18 ; Восстановление значений регистров ds, ax:
19 020A:07D1  58                   pop ax
20 020A:07D2  1F                   pop ds
21 020A:07D3  EB 03                jmp short loc_9                      ; (07D8)
22 020A:07D5         loc_8:
23 ; Сброс флага IF:
24 020A:07D5  FA                   cli                                  ; Disable
       interrupts
25 020A:07D6  EB F8                jmp short loc_7                      ; (07D0)
26 020A:07D8         loc_9:
27 ; Возврат из подпрограммы:
28 020A:07D8  C3                   retn
29                   sub_3         endp
```

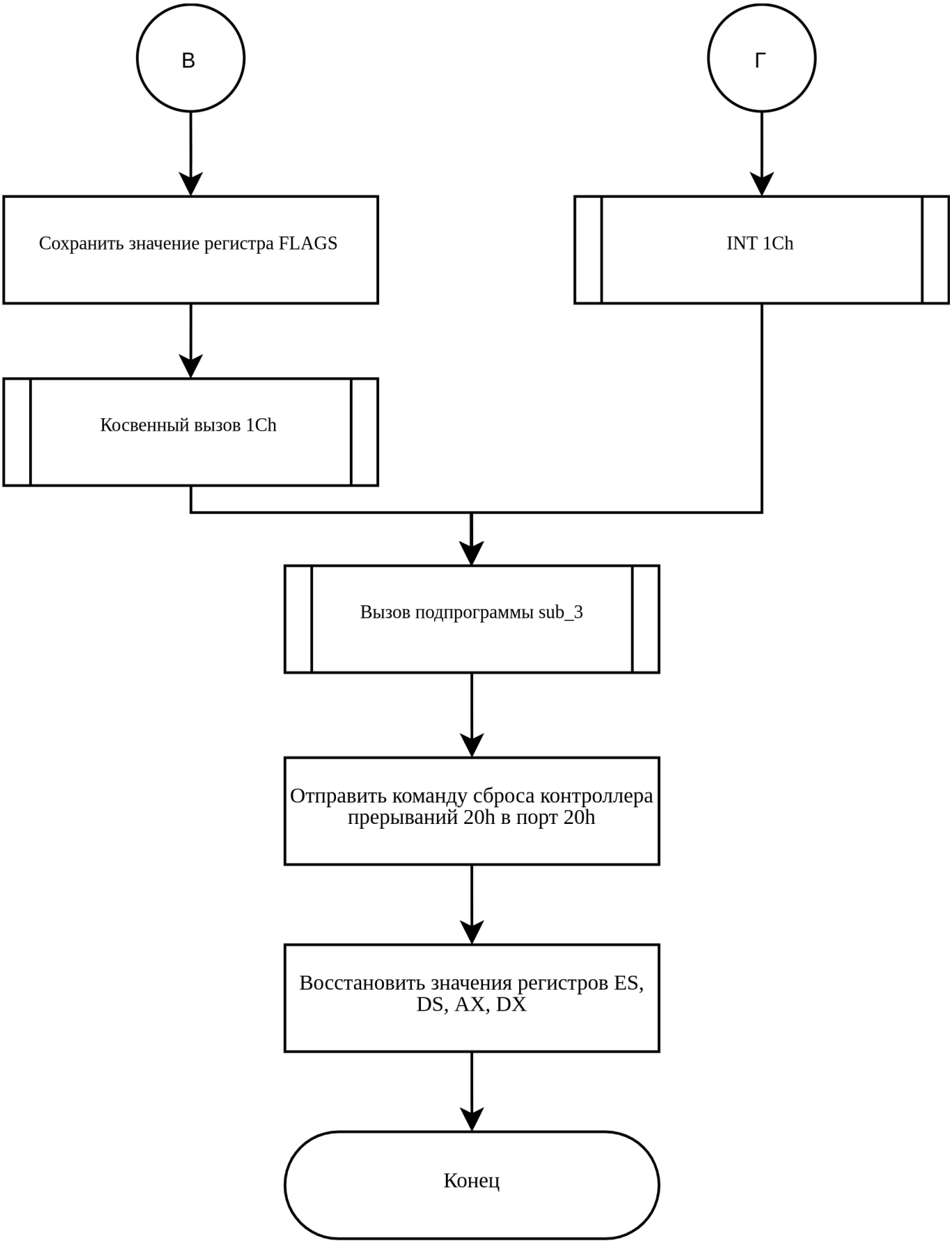
2. Схема алгоритмов

2.1. Схема алгоритма обработчика INT8h




А
2 младших байта счётчика реального времени == 24?
Да
2 старших байта счётчика реального времени == 176?
Да
Обнулить 4 байта счётчика реального времени по адресу 0040:006Ch
Установить значение 1 по адресу 0040:0070h
Установить значение 8 в регистре AL
Сохранить значение регистра AX
Б


Б
Декремент счётчика времени до отключения моторчика дисковода по адресу 0040:0040h
Значение счётчика времени до отключения моторчика дисковода == 0?
Да
Установить флаг отключения моторчика дисковода
Отправить команду 0Ch отключения моторчика дисковода в порт 3F2h
Восстановить значение регистра AX
Установлен ли флаг PF по адресу 0040:0314h?
Да
В
Г


В
Г
Сохранить значение регистра FLAGS
INT 1Ch
Косвенный вызов 1Ch
Вызов подпрограммы sub_3
Отправить команду сброса контроллера прерываний 20h в порт 20h
Восстановить значения регистров ES, DS, AX, DX
Конец

2.2. Схема алгоритма процедуры sub_3