



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 Баринов Н. Ю.

Группа ИУ7-51Б

Преподаватель Рязанова Н. Ю.

Москва — 2022 г.

1. Полученный дизассемблированный код

1.1. Листинг обработчика прерывания INT 8h

```
1 ; Вызов подпрограммы sub_3:
2 020A:0746 E8 0070          call    sub_3                ; (07B9)
3 ; Сохранение значений 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 сегментных регистров ds, es:
9 ; (40h - сегментная часть адреса области данных BIOS)
10 020A:074D B8 0040         mov     ax,40h
11 020A:0750 8E D8          mov     ds,ax
12 020A:0752 33 C0          xor     ax,ax
13 020A:0754 8E C0          mov     es,ax
14 ; Инкрементирование счётчиков таймера:
15 ; 0040:006C, 0040:006E - адреса младшего и старшего слова
16 ; счётчика прерываний таймера BIOSa
17 020A:0756 FF 06 006C      inc     word ptr ds:[6Ch]        ; (0040:006C=9A82h)
18 020A:075A 75 04          jnz     loc_2                ; Jump if not zero
19 020A:075C FF 06 006E      inc     word ptr ds:[6Eh]        ; (0040:006E=0)
20 ; Сброс счётчиков времени при наступлении нового дня:
21 ; 0040:006E == 18h (24), 0040:006C == B0h (176)
22 ; 18h << 16 + B0h == 86400 * с;
23 ; с = 1573040 / 86400 = 18.2... - количество срабатываний таймера в секунду
24 020A:0760          loc_2:
25 020A:0760 83 3E 006E 18    cmp     word ptr ds:[6Eh],18h    ; (0040:006E=0)
26 020A:0765 75 15          jne     loc_3                ; Jump if not equal
27 020A:0767 81 3E 006C 00B0  cmp     word ptr ds:[6Ch],0B0h   ; (0040:006C=9A82h)
28 020A:076D 75 0D          jne     loc_3                ; Jump if not equal
29 020A:076F A3 006E          mov     word ptr ds:[6Eh],ax    ; (0040:006E=0)
30 020A:0772 A3 006C          mov     word ptr ds:[6Ch],ax    ; (0040:006C=9A82h)
31 ; Установка в 1 флага наращивания даты (начальное значение - 0)
32 020A:0775 C6 06 0070 01      mov     byte ptr ds:[70h],1     ; (0040:0070=0)
33 ; Установка al = 8:
34 020A:077A 0C 08          or      al,8
35 020A:077C          loc_3:
36 ; Сохранение регистра ax:
37 020A:077C 50              push    ax
38 ; Декремент счётчика времени до отключения моторчика дисковод:
39 ; (0040:0040 - адрес счётчика времени в области данных накопителя FDD)
40 020A:077D FE 0E 0040      dec     byte ptr ds:[40h]        ; (0040:0040=39h)
41 020A:0781 75 0B          jnz     loc_4                ; Jump if not zero
42 ; Установка флагов, отвечающих за отключение моторчика дисковод:
43 020A:0783 80 26 003F F0    and     byte ptr ds:[3Fh],0F0h   ; (0040:003F=0)
44 ; Отправка команды отключения моторчика дисковод:
45 020A:0788 B0 0C          mov     al,0Ch
46 020A:078A BA 03F2        mov     dx,3F2h
47 020A:078D EE              out     dx,al                ; port 3F2h, disk0
48          ctrl output
49 020A:078E          loc_4:
50 ; Восстановление регистра ax:
51 020A:078E 58              pop     ax
52 ; Проверка второго бита (Parity Flag - флаг чётности):
53 ; 0040:0314h - адрес области данных BIOS, содержащей копию флагов
```

```

53 020A:078F F7 06 0314 0004      test    word ptr ds:[314h],4      ;
    (0040:0314=3200h)
54 020A:0795 75 0C                  jnz     loc_5                      ; Jump if not zero
55 ; Сохранение младшего байта регистра FLAGS в AH:
56 020A:0797 9F                      lahf                                ; Load ah from flags
57 ; Обмен значений регистров ah и al:
58 ; Теперь младший байт регистра FLAGS находится в младшем байте регистра ah
59 020A:0798 86 E0                  xchg    ah,al
60 ; Сохранение регистра ah:
61 020A:079A 50                      push     ax
62 ; Косвенный вызов пользовательского прерывания по адресу в таблице векторов прерываний:
63 ; В этом случае не произойдёт push регистра FLAGS, на его месте будет AX,
64 ; который восстановится в регистр FLAGS после выхода из обработчика прерывания
65 020A:079B 26 FF 1E 0070          call     dword ptr es:[70h]      ; (0000:0070=6ADh)
66 020A:07A0 EB 03                  jmp     short loc_6              ; (07A5)
67 020A:07A2 90                      nop
68 ; Вызов пользовательского прерывания через int 1Ch:
69 020A:07A3                loc_5:
70 020A:07A3 CD 1C                  int      1Ch                    ; Timer break (call
    each 18.2ms)
71 020A:07A5                loc_6:
72 ; Вызов подпрограммы sub_3:
73 020A:07A5 E8 0011              call     sub_3                  ; (07B9)
74 ; Сброс контроллера прерываний (отправка команды End Of Interrupt):
75 ; Разрешение обработки прерываний с текущим или более низким приоритетом
76 020A:07A8 B0 20                  mov     al,20h                 ; ' '
77 020A:07AA E6 20                  out     20h,al                 ; port 20h, 8259-1 int command
78 ; al = 20h, end of interrupt
79 ; Восстановление значений регистров es, ds, ax, dx:
80 020A:07AC 5A                      pop     dx
81 020A:07AD 58                      pop     ax
82 020A:07AE 1F                      pop     ds
83 020A:07AF 07                      pop     es
84 020A:07B0 E9 FE99              jmp     $-164h                 ; 020A:07B0h - 164h
    = 020A:064Ch
85 ; ...
86 ; Возврат из прерывания
87 020A:06AC CF                      iret                            ; Interrupt return

```

1.2. Листинг процедуры sub_3

```

1 020A:07B9      sub_3      proc    near
2 ; Сохранение значений регистров ds, ax:
3 020A:07B9 1E              push    ds
4 020A:07BA 50              push    ax
5 ; Загрузка сегментного регистра ds:
6 020A:07BB B8 0040          mov     ax,40h
7 020A:07BE 8E D8          mov     ds,ax
8 ; Сохранение младшего байта регистра FLAGS в AH:
9 020A:07C0 9F                      lahf                                ; Load ah
    from flags
10 ; Проверка DF и старшего бита IOPL по адресу 0040:0314h:
11 020A:07C1 F7 06 0314 2400      test     word ptr ds:[314h],2400h      ;
    (0040:0314=3200h)
12 020A:07C7 75 0C                  jnz     loc_8                      ; Jump if
    not zero

```

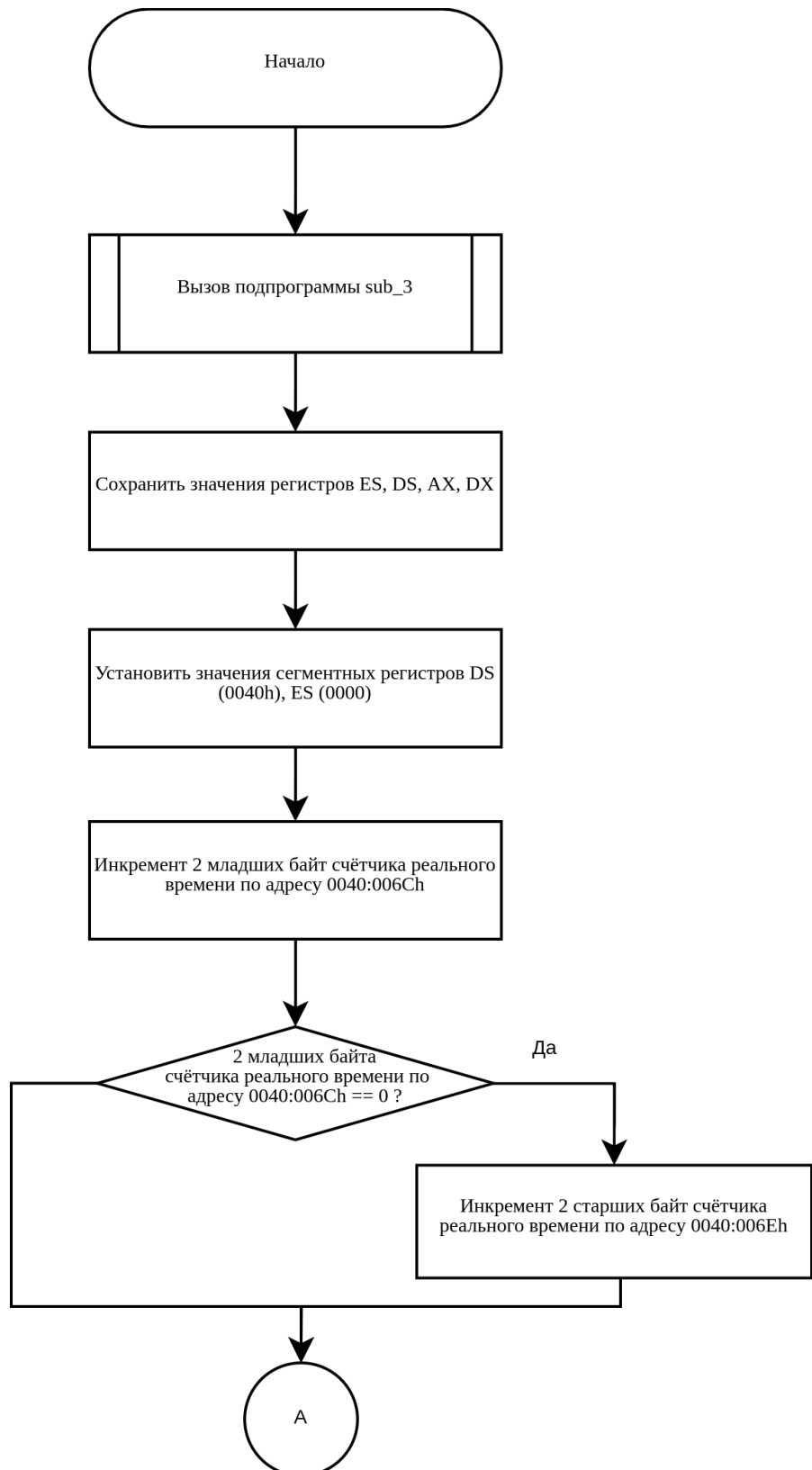
```

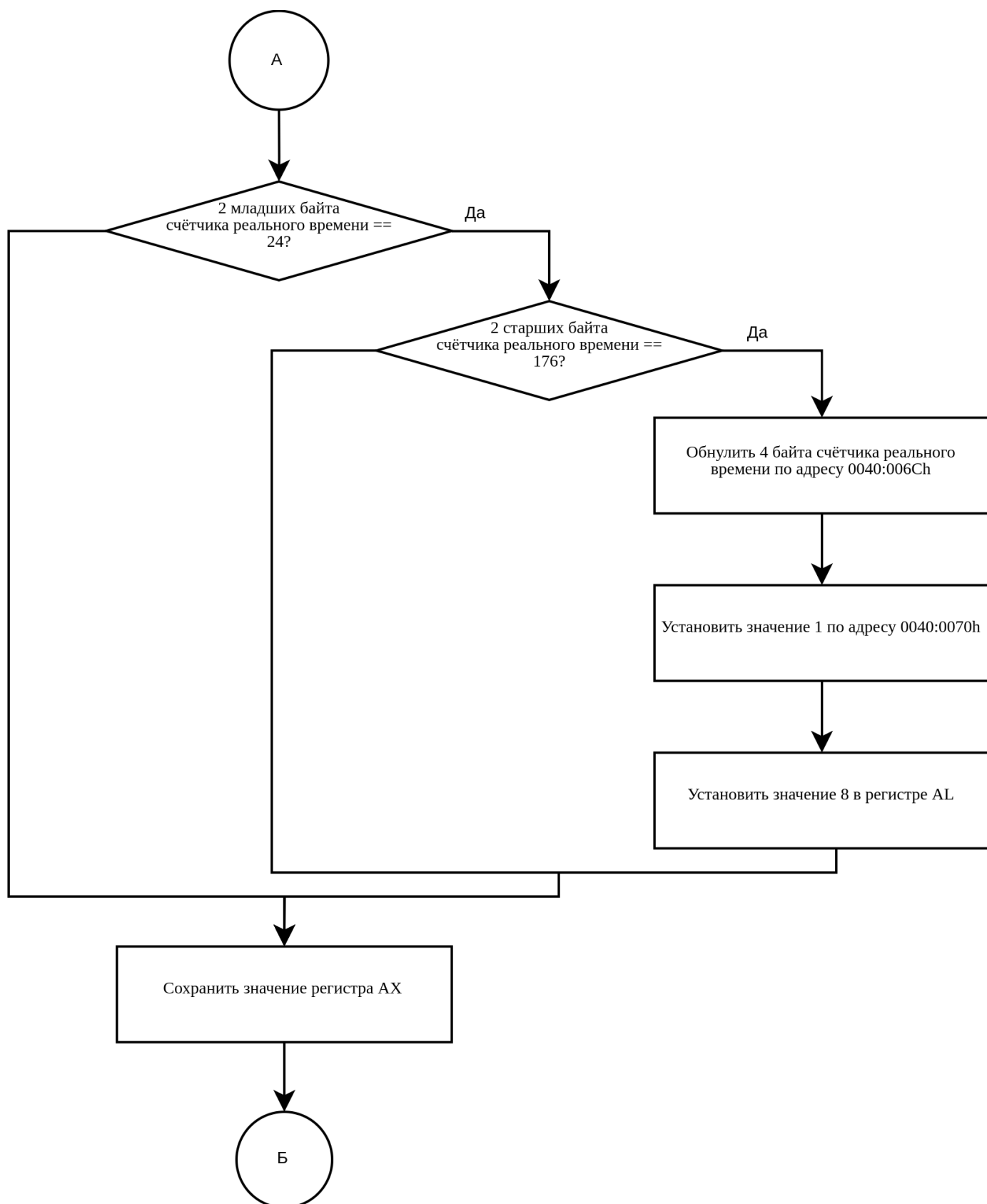
13 ; Установка 9 бита в 0 - сброс IF (запрет прерываний):
14 020A:07C9 F0> 81 26 0314 FDFF lock and word ptr ds:[314h],0FDFFh ;
    (0040:0314=3200h)
15 020A:07D0      loc_7:
16 ; Сохранение регистра AH в младший байт FLAGS:
17 020A:07D0 9E      sahf      ; Store ah
    into flags
18 ; Восстановление 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

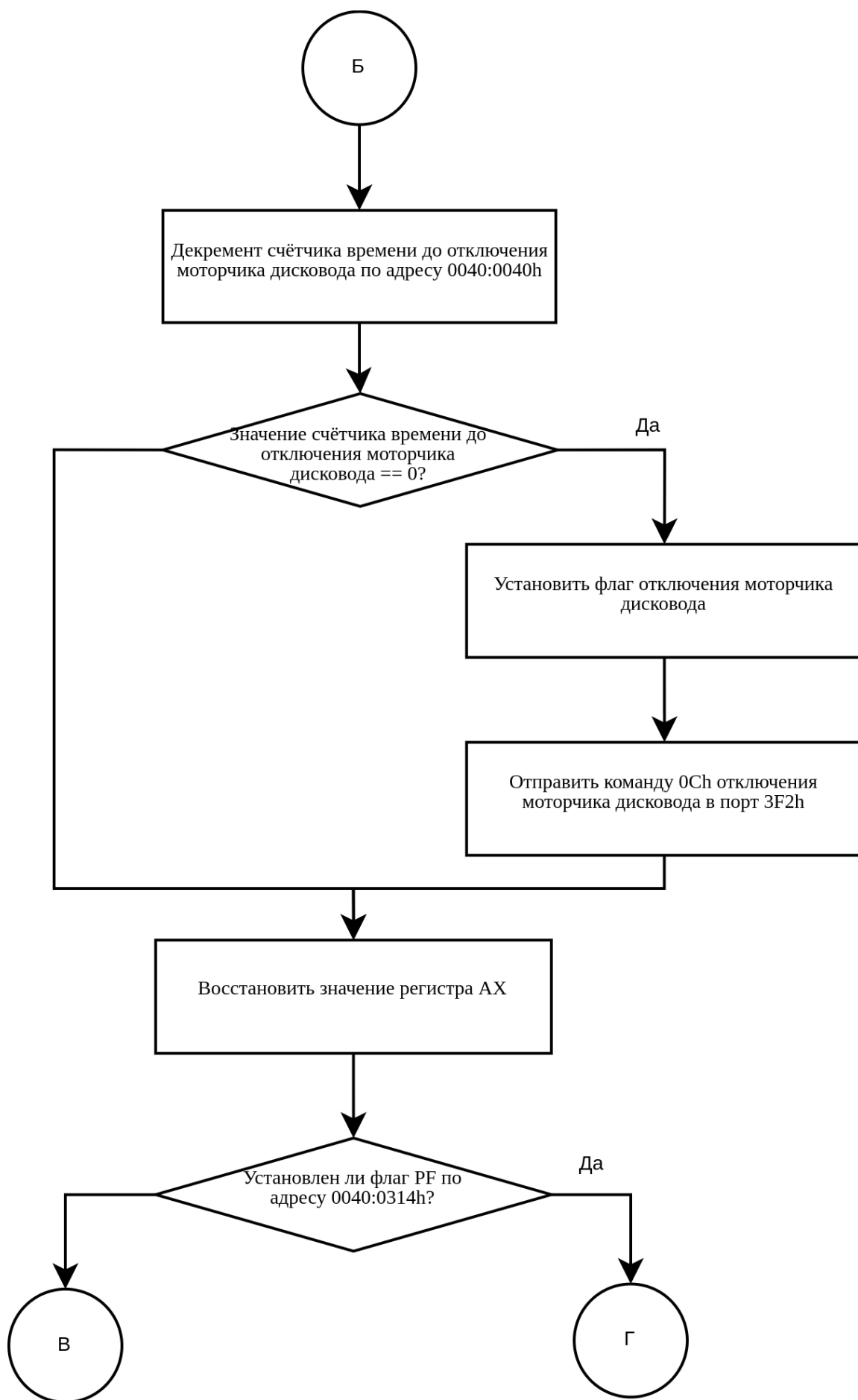
```

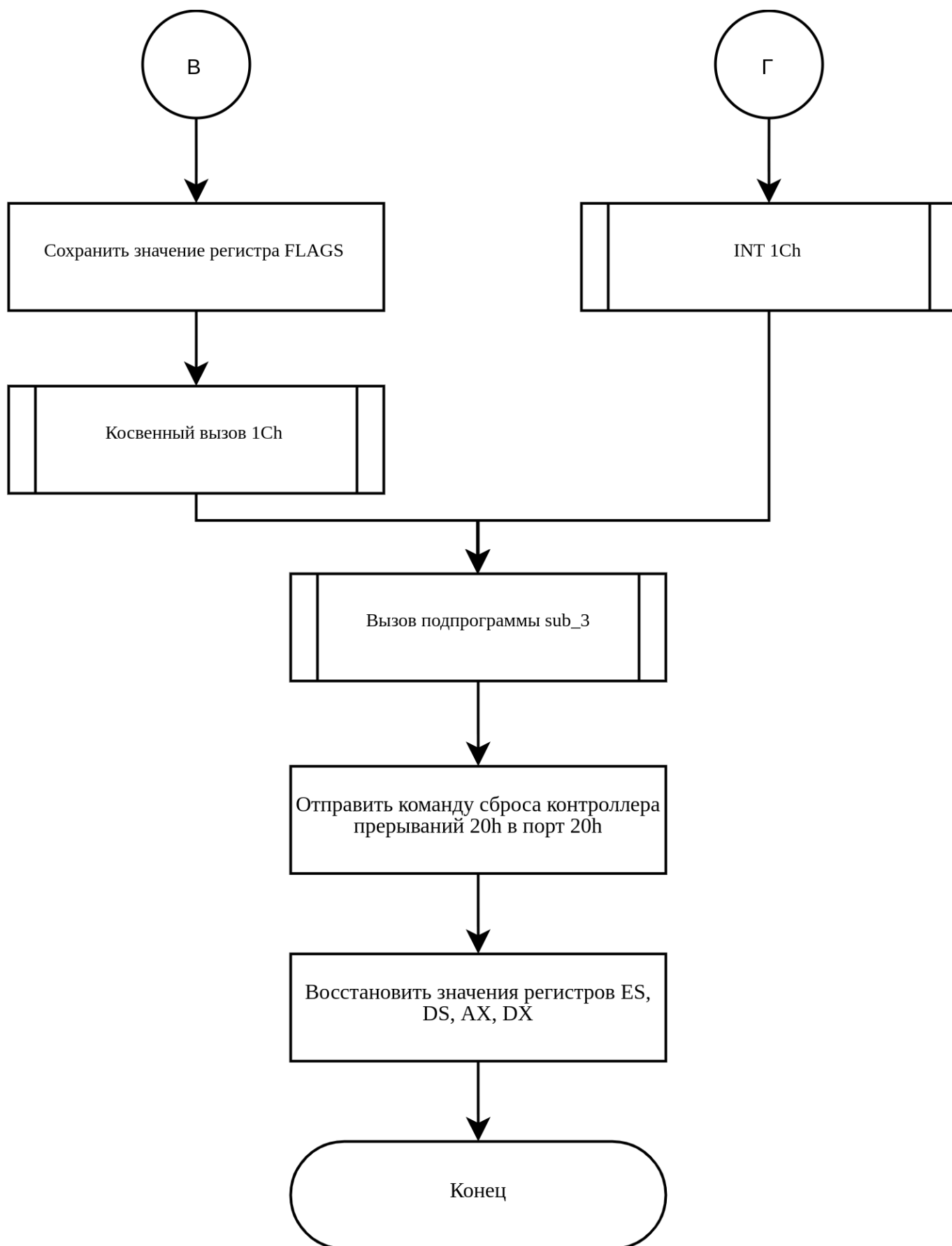
2. Схема алгоритмов

2.1. Схема алгоритма обработчика INT8h









2.2. Схема алгоритма процедуры sub_3

