



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

О Т Ч Е Т
по лабораторной работе № 1

Дисциплина: Операционные системы

Название: Дизассемблирование INT 8h

Студент

ИУ7 - 53Б

(Группа)

(Подпись, дата)

(И. О. Фамилия)

Преподаватель

Н. Ю. Рязанова

(Подпись, дата)

(И. О. Фамилия)

Москва, 2022

1. Цель и задачи лабораторной работы

Знакомство со средством дизассемблирования – **sourcer** и с получением дизассемблерного кода ядра операционной системы Windows на примере обработчика прерывания **INT 8h** в **virtual mode** – специальном режиме защищенного режима, который эмулирует реальный режим работы вычислительной системы на базе процессоров Intel.

Используя **sourcer (sr.exe)** получить дизассемблированный код обработчика аппаратного прерывания от системного таймера **INT 8h**. На основе полученного кода составить алгоритм работы обработчика **INT 8h**. Составить отчет в письменном виде.

2. Полученный дизассемблированный код

2.1. Листинг обработчика прерывания INT 8h

```
020A:0746 E8 0070          call    sub_1              ; (07B9)
020A:0749 06              push    es
020A:074A 1E              push    ds
020A:074B 50              push    ax
020A:074C 52              push    dx
020A:074D B8 0040          mov     ax,40h
020A:0750 8E D8           mov     ds,ax
020A:0752 33 C0           xor     ax,ax              ; Zero register
020A:0754 8E C0           mov     es,ax
020A:0756 FF 06 006C       inc     word ptr ds:[6Ch]   ; (0040:006C=12A2h)
020A:075A 75 04           jnz     loc_1              ; Jump if not zero
020A:075C FF 06 006E       inc     word ptr ds:[6Eh]   ; (0040:006E=10h)
020A:0760                                loc_1:
020A:0760 83 3E 006E 18    cmp     word ptr ds:[6Eh],18h ; (0040:006E=10h)
020A:0765 75 15           jne     loc_2              ; Jump if not equal
020A:0767 81 3E 006C 00B0  cmp     word ptr ds:[6Ch],0B0h ; (0040:006C=12A2h)
020A:076D 75 0D           jne     loc_2              ; Jump if not equal
020A:076F A3 006E         mov     word ptr ds:[6Eh],ax ; (0040:006E=10h)
020A:0772 A3 006C         mov     word ptr ds:[6Ch],ax ; (0040:006C=12A2h)
020A:0775 C6 06 0070 01    mov     byte ptr ds:[70h],1 ; (0040:0070=0)
020A:077A 0C 08          or      al,8
```

```

020A:077C                                loc_2:
020A:077C  50                                push    ax
020A:077D  FE 0E 0040                       dec     byte ptr ds:[40h]      ; (0040:0040=34h)
020A:0781  75 0B                             jnz     loc_3                  ; Jump if not zero
020A:0783  80 26 003F F0                     and     byte ptr ds:[3Fh],0F0h ; (0040:003F=0)
020A:0788  B0 0C                             mov     al,0Ch
020A:078A  BA 03F2                           mov     dx,3F2h
020A:078D  EE                             out     dx,al
; port 3F2h, disk0 control output
020A:078E                                loc_3:
020A:078E  58                                pop     ax
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
020A:0797  9F                                lahf                                ; Load ah from flags
020A:0798  86 E0                             xchg    ah,al
020A:079A  50                                push    ax
020A:079B  26: FF 1E 0070                   call    dword ptr es:[70h]      ; (0000:0070=6ADh)
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)
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
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                                loc_9:
020A:06AA  58                                pop     ax
020A:06AB  1F                                pop     ds

```

```
020A:06AC CF                                iret                                ; Interrupt return
```

2.2. Листинг подпрограммы sub_1

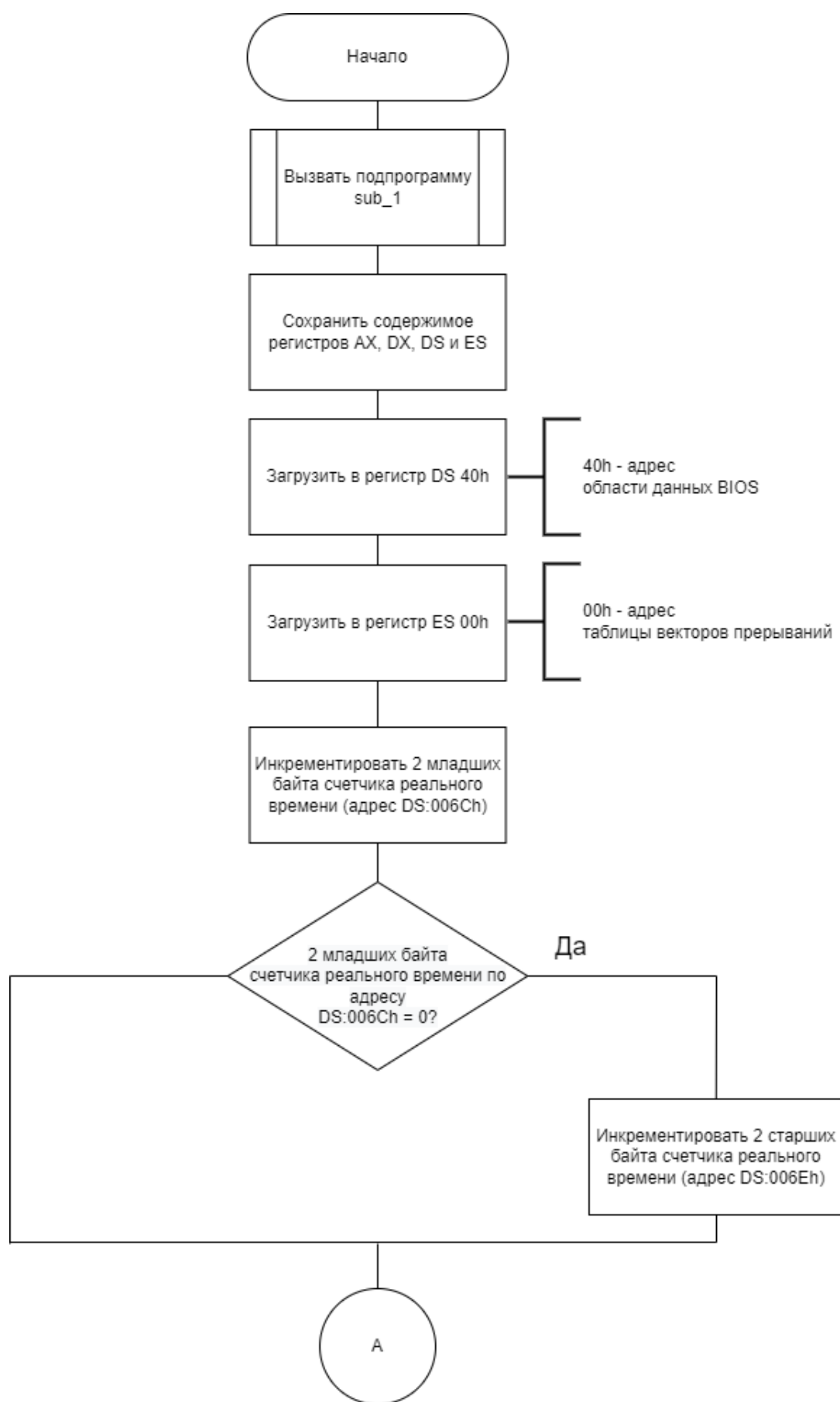
```

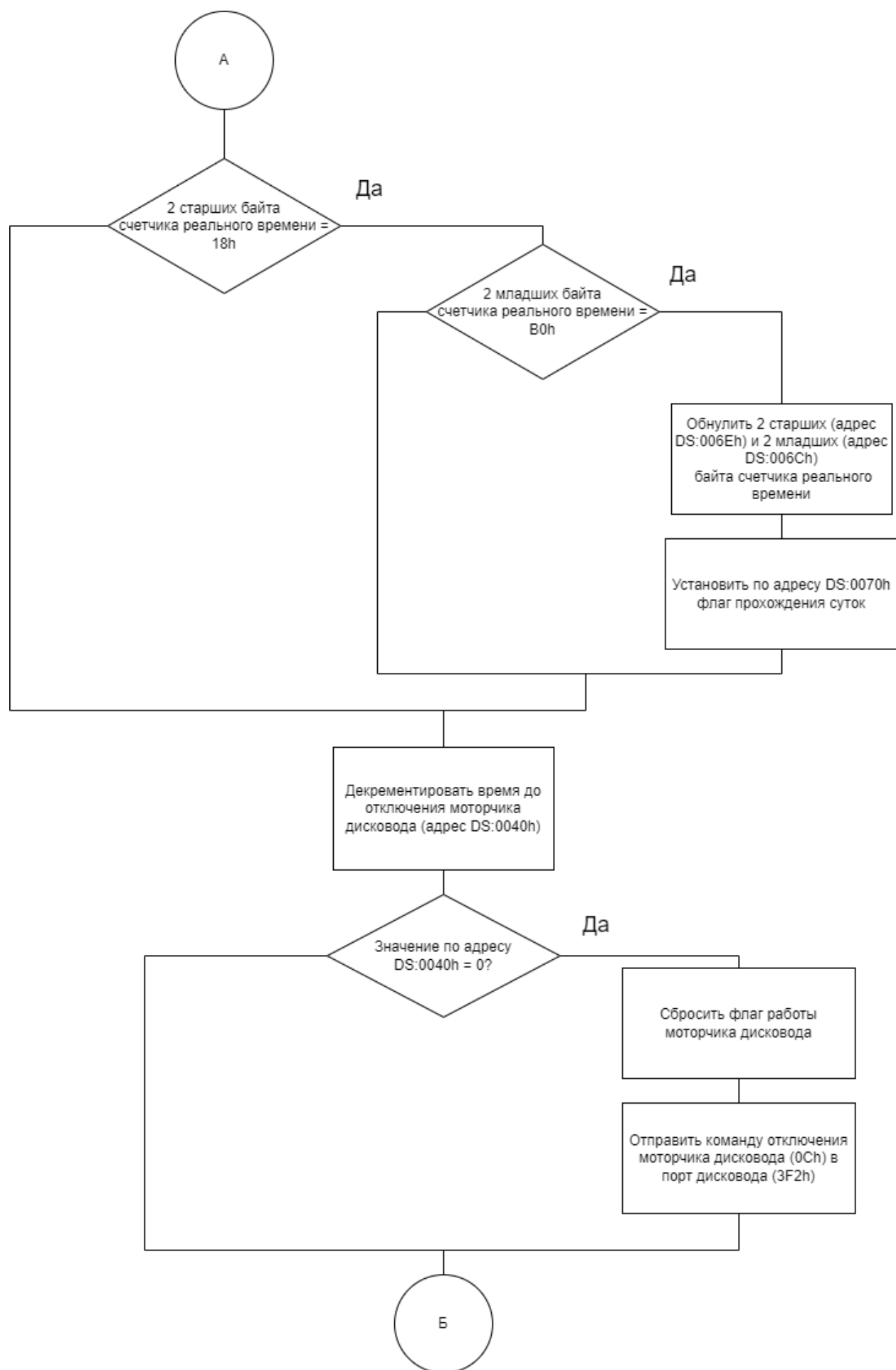
                                sub_1      proc    near
020A:07B9 1E                                push    ds
020A:07BA 50                                push    ax
020A:07BB B8 0040                        mov     ax,40h
020A:07BE 8E D8                        mov     ds,ax
020A:07C0 9F                                lahf         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

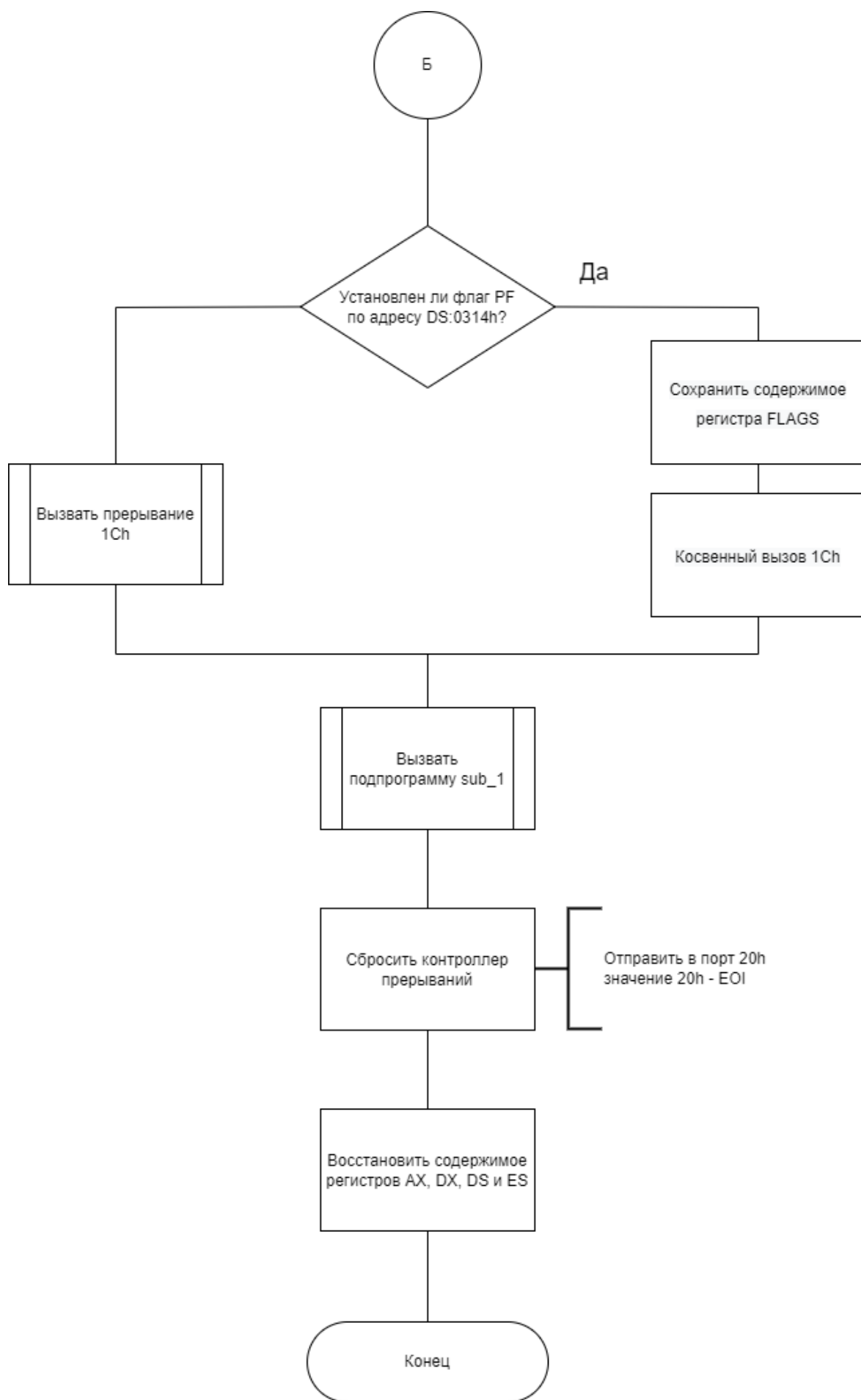
```

3. Схемы алгоритмов

3.1. Схема алгоритма обработчика прерывания INT 8h







3.2. Схема алгоритма подпрограммы sub_1

