



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, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

## ОТЧЕТ

по лабораторной работе № 1

по курсу «Операционные системы»

на тему: «Обработчик прерывания от системного таймера»

Студент ИУ7-55Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Романов С. К.  
(И. О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Рязанова Н. Ю.  
(И. О. Фамилия)

2022 г.

## Оглавление

<b>1</b>	<b>Исходный дизассемблированный код</b>	<b>3</b>
1.1	Прерывание 8h . . . . .	3
1.2	SUB_6 . . . . .	6
<b>2</b>	<b>Схема алгоритмов</b>	<b>7</b>
2.1	Схема прерывания int 8h . . . . .	7
2.2	Схема подпрограммы sub_6 . . . . .	10

# 1 Исходный дизассемблированный код

## 1.1 Прерывание 8h

```
1      Temp.lst                      Sourcer Listing v3.07          6-Sep-22
2
3      7:15 pm    Page 1
4
5      ;; вызов sub_1
6      020A:0746  E8 0070              call    sub_6                ; (07B9)
7      ;; Сохранение регистров ES, DS, AX, DX
8      020A:0749  06                  push    es
9      020A:074A  1E                  push    ds
10     020A:074B  50                  push    ax
11     020A:074C  52                  push    dx
12
13     ;; DS = 0040
14     020A:074D  B8 0040              mov     ax,40h
15     020A:0750  8E D8              mov     ds,ax
16     ;; AX = 0
17     020A:0752  33 C0              xor     ax,ax
18     020A:0754  8E C0              mov     es,ax
19
20     ;; 0040:006Ch - адрес счетчика таймера
21     020A:0756  FF 06 006C          inc     word ptr ds:[6Ch]    ;
22     (0040:006C=41B9h)
23     020A:075A  75 04              jnz     loc_2                ; Jump if not zero
24
25     ;; 0040:006Eh - старшие 2 байта счетчика таймера
26     020A:075C  FF 06 006E          inc     word ptr ds:[6Eh]    ;
27     (0040:006E=13h)
28     020A:0760              loc_2:
29
30     ;; Проверка: 0040:006Eh == 18h (24) и 0040:006Ch == B0h (176)
31     ;; Можно убедиться в том, что: 18h << 16 + B0h = 24 * 60 * 60 * freq,
32     ;; где freq - кол-во раз, которое вызывается таймер в секунду.
33     ;; Таким образом из того, что условие выполняется, следует, что прошли
34     ;; сутки.
35
36     020A:0760  83 3E 006E 18          cmp     word ptr ds:[6Eh],18h    ;
37     (0040:006E=13h)
38     020A:0765  75 15              jne     loc_3                ; Jump if not
39     equal
40     020A:0767  81 3E 006C 00B0        cmp     word ptr ds:[6Ch],0B0h    ;
41     (0040:006C=41B9h)
42     020A:076D  75 0D              jne     loc_3                ; Jump if not
43     equal
44
45     ;; Обнуление счетчика (старшего слова и младшего слова)
```

```

37
38 020A:076F  A3 006E          mov word ptr ds:[6Eh],ax      ;
    (0040:006E=13h)
39 020A:0772  A3 006C          mov word ptr ds:[6Ch],ax      ;
    (0040:006C=41B9h)
40
41 ;; Прошло более 24 часов, занесение значения 1 в 0040:0070
42
43 020A:0775  C6 06 0070 01      mov byte ptr ds:[70h],1    ;
    (0040:0070=0)
44
45 ;; AL = 8
46 020A:077A  0C 08          or al,8
47
48 020A:077C          loc_3:
49 ;; сохранение AX
50 020A:077C  50          push ax
51
52 ;; Декремент счетчика отключения моторчика
53 020A:077D  FE 0E 0040      dec byte ptr ds:[40h]    ;
    (0040:0040=6Ah)
54 020A:0781  75 0B          jnz loc_4                ; Jump if not zero
55
56 ;; Установка флага отключения моторчика дисковод (1-3 бита == 0)
57 020A:0783  80 26 003F F0      and byte ptr ds:[3Fh],0F0h ;
    (0040:003F=0)
58
59 ;; 3 строчки - посылка команды отключения дисководу
60 020A:0788  B0 0C          mov al,0Ch
61 020A:078A  BA 03F2        mov dx,3F2h
62 020A:078D  EE          out dx,al                ; port 3F2h, dsk0
    contrl output
63
64 020A:078E          loc_4:
65 ;; Восстановление AX
66 020A:078E  58          pop ax
67
68 ;; Проверка третьего бита (Parity Flag)
69 020A:078F  F7 06 0314 0004      test word ptr ds:[314h],4    ;
    (0040:0314=3200h)
70 020A:0795  75 0C          jnz loc_5                ; Jump if not zero
71
72 ;; Копирование младшего байта FLAGS в ah
73 020A:0797  9F          lahf                ; Load ah from
    flags
74 ;; Смена мест:
75 ;; теперь в ax: 08XXh - где XX - младший байт FLAGS
76 020A:0798  86 E0          xchg ah,al

```

```

77      ;; Кладем это на стек и вызываем прерывание
78 020A:079A  50                      push    ax
79
80      ;; Вызываем 1Ch через адрес в таблице векторов. До этого мы добавили в
      стек AX, в то время как
81      ;; вызов int делает push флагов (то есть наш ax, описанный 6 строками
      выше будет как FLAGS в 1Ch)
82 020A:079B  26: FF 1E 0070          call    dword ptr es:[70h]    ;
      (0000:0070=6ADh)
83 020A:07A0  EB 03                  jmp short loc_6          ; (07A5)
84 020A:07A2  90                      nop
85 020A:07A3                      loc_5:
86 020A:07A3  CD 1C                  int 1Ch                  ; Timer break (call
      each 18.2ms)
87 020A:07A5                      loc_6:
88 020A:07A5  E8 0011                call    sub_6            ; (07B9)
89
90      ;; Сброс контроллера прерываний
91 020A:07A8  B0 20                  mov al,20h              ; ' '
92 020A:07AA  E6 20                  out 20h,al              ; port 20h,
      8259-1 int command
93
94                                     ; al = 20h, end of interrupt
95                                     ; al = 20h, end of interrupt
96
97      ;; Восстановление регистров
98 020A:07AC  5A                      pop dx
99 020A:07AD  58                      pop ax
100 020A:07AE  1F                      pop ds
101 020A:07AF  07                      pop es
102 020A:07B0  E9 FE99                jmp $-164h
103 ...
104      ;; 07B0h - 0164h = 064Ch
105      ;; Листинг ниже
      ...

```

## 1.2 SUB\_6

```
1          sub_6      proc      near
2      ;; Сохранение регистров
3      020A:07B9  1E          push    ds
4      020A:07BA  50          push    ax
5      ;; DS = 40h
6      020A:07BB  B8 0040      mov     ax,40h
7      020A:07BE  8E D8      mov     ds,ax
8
9      ;; Младший байт FLAGS в AH
10     020A:07C0  9F          lahf             ; Load ah from
        flags
11
12     ;; Установлены ли старший бит IOPL или DF?
13     020A:07C1  F7 06 0314 2400      test     word ptr
        ds:[314h],2400h ; (0040:0314=3200h)
14     020A:07C7  75 0C          jnz     loc_8             ; Jump if not zero
15
16
17     ;; сброс IF (Interrupt flag) в 0040:0314h (обнуление 9 бита)
18     020A:07C9  F0> 81 26 0314 FDFF      lock and word ptr
        ds:[314h],0FDFFh ; (0040:0314=3200h)
19
20     020A:07D0          loc_7:
21     ;; AH копируется в младший байт FLAGS
22     020A:07D0  9E          sahf             ; Store ah into
        flags
23     020A:07D1  58          pop     ax
24     020A:07D2  1F          pop     ds
25     020A:07D3  EB 03      jmp     short loc_9 ; (07D8)
26
27     020A:07D5          loc_8:
28     ;; Сброс IF (Interrupt flag)
29     020A:07D5  FA          cli             ; Disable interrupts
30     020A:07D6  EB F8      jmp     short loc_7 ; (07D0)
31     020A:07D8          loc_9:
32     020A:07D8  C3          retn
33     sub_6      endp
```

## 2 Схема алгоритмов

### 2.1 Схема прерывания int 8h

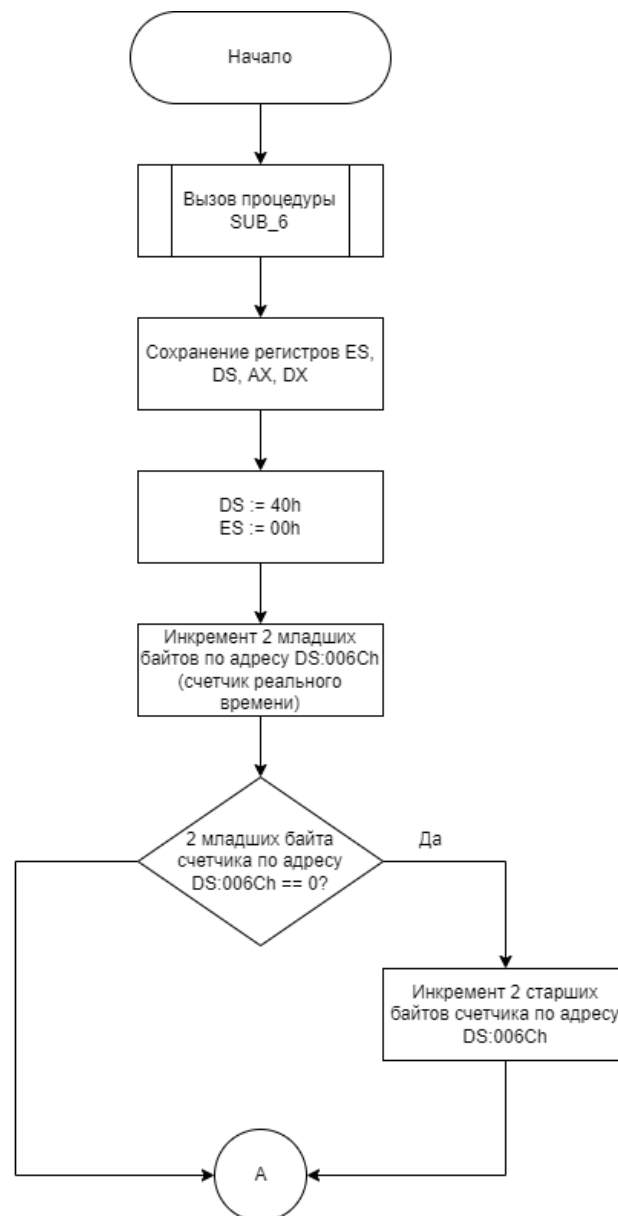


Рисунок 2.1 – Схема А

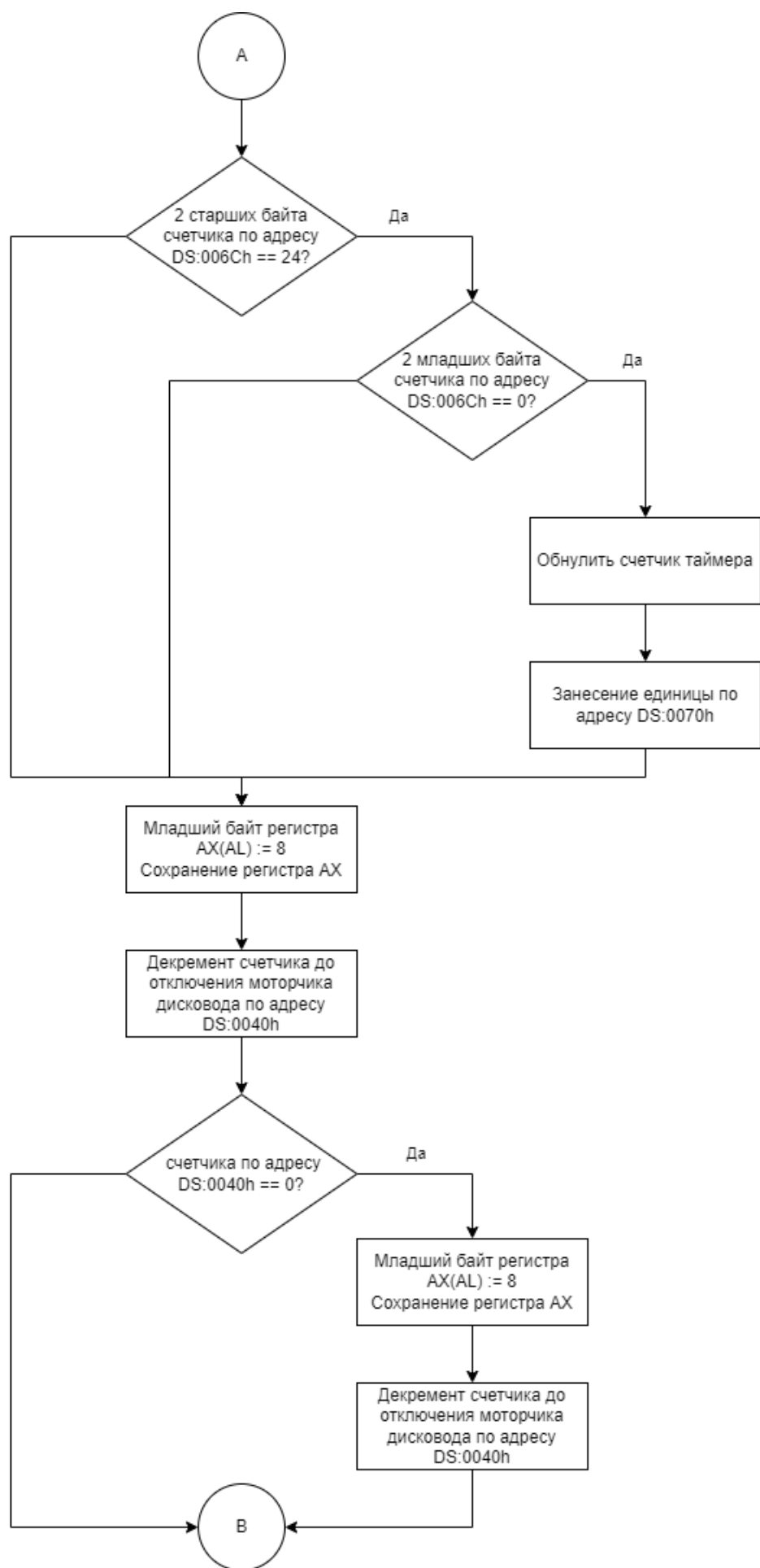


Рисунок 2.2 – Схема Б



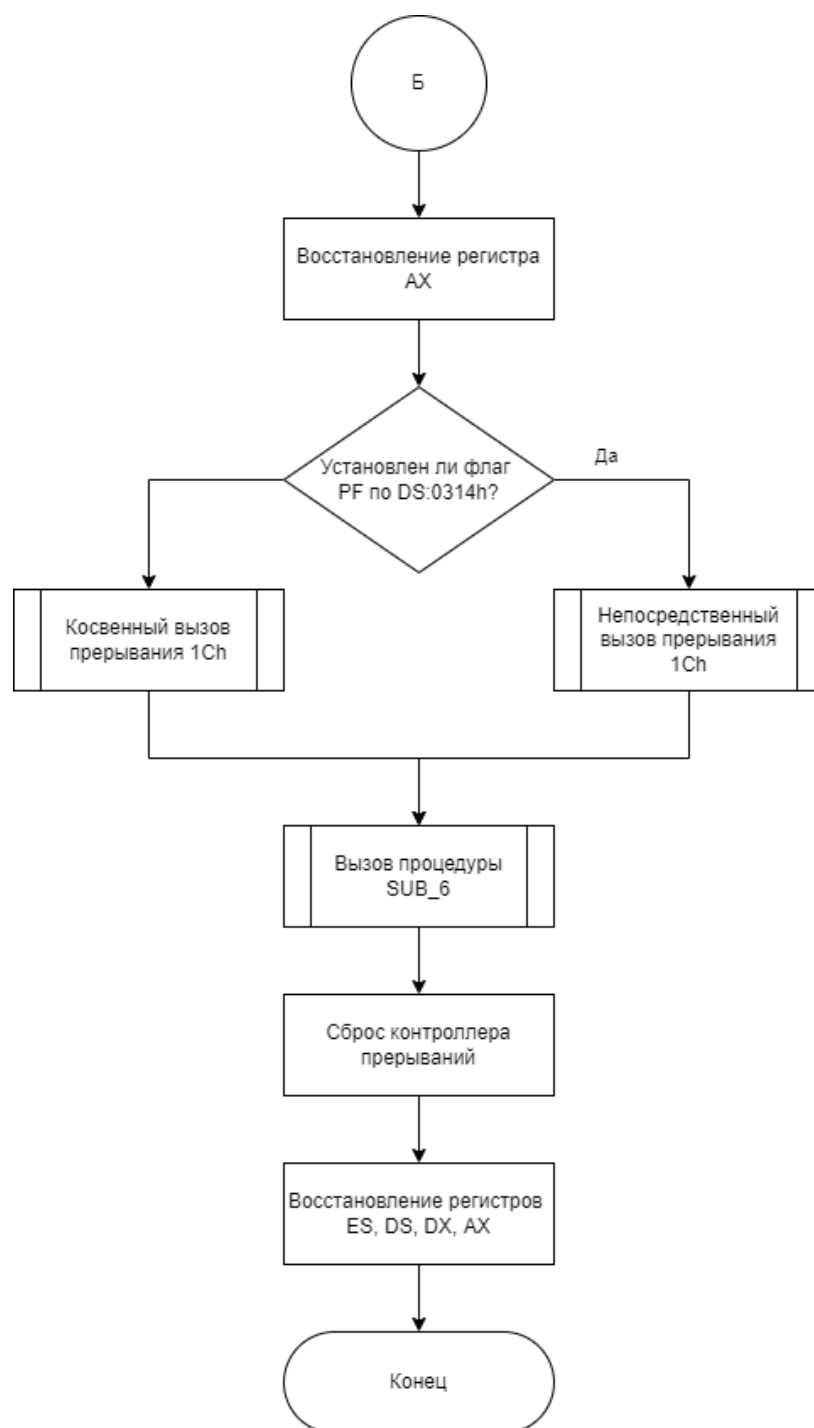


Рисунок 2.3 – Схема С

## 2.2 Схема подпрограммы sub\_6

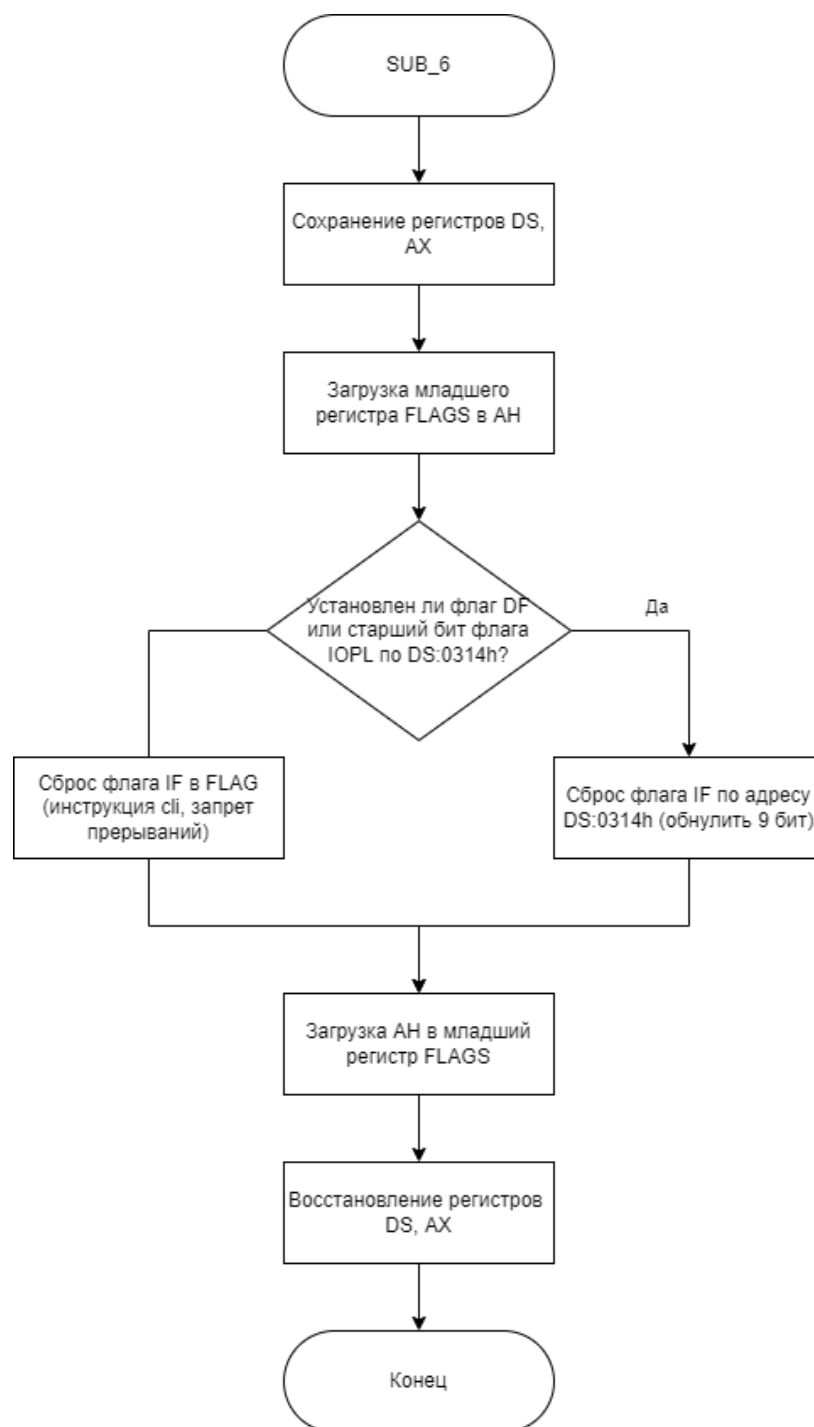


Рисунок 2.4 – sub\_6