

# 1 Листинги дизассемблированного кода

## 1.1 Листинг обработчика прерывания INT 8

```
1      Temp.lst      Sourcer v5.10      10-Sep-25
2      6:58 pm      Page 1
3
4      ;* No entry
5      ;* point to
6      ;* code
7
8      ;; -- Вызов подпрограммы sub_1
9      020A:0746 E8 0070      ;*      call      sub_1      ;*
10     020A:0746 E8 70 00      db      0E8h, 70h, 00h
11     ;; -- Сохранение регистров es,dx,ax,dx
12     020A:0749 06      push      es
13     020A:074A 1E      push      ds
14     020A:074B 50      push      ax
15     020A:074C 52      push      dx
16     ;; -- Загрузка в ds 0040h
17     020A:074D B8 0040      mov      ax,40h
18     020A:0750 8E D8      mov      ds,ax
19     020A:0752 33 C0      xor      ax,ax      ; Zero register
20     020A:0754 8E C0      mov      es,ax
21     ;; -- Инкремент счетчика реального времени по адресу DS:06Ch
22     020A:0756 FF 06 006C      inc word ptr ds:[6Ch] ;
23     (0040:006C=0F9B1h)
24     020A:075A 75 04      jnz loc_1      ; Jump if not
25     zero
26     ;; -- Инкремент 2 старших байтов счетчика реального времени по адре
27     су DS:06Ch
28     020A:075C FF 06 006E      inc word ptr ds:[6Eh] ;
29     (0040:006E=12h)
30     ;; -- Сброс счетчика времени при наступлении нового дня:
31     ;; 0040:006E = 18h (24), 0040:006C = 0B0h (176)
32     ;; 18h << 16 + B0h = 24 * 60 * 60 * с
33     ;; с = 1573040 / 86400 = 18.2... - количество срабатываний таймера
34     в секунду
35     ;; Таким образом из выполнения условия следует, что прошли сутки
36     020A:0760      loc_1:
37     020A:0760 83 3E 006E 18      cmp word ptr ds:[6Eh],18h ;
38     (0040:006E=12h)
39     020A:0765 75 15      jne loc_2      ; Jump if not
40     equal
41     020A:0767 81 3E 006C 00B0      cmp word ptr ds:[6Ch],0B0h ;
42     (0040:006C=0F9B1h)
43     020A:076D 75 0D      jne loc_2      ; Jump if not
44     equal
45     ;; -- Обнуление счетчика реального времени
46     020A:076F A3 006E      mov word ptr ds:[6Eh],ax ;
47     (0040:006E=12h)
48     020A:0772 A3 006C      mov word ptr ds:[6Ch],ax ;
49     (0040:006C=0F9B1h)
50     ;; -- Установка в DS:0070 так как прошло более 24 часов
51     020A:0775 C6 06 0070 01      mov byte ptr ds:[70h],1 ;
52     (0040:0070=0)
53     ;; -- Установка 3-й бит в al
54     020A:077A 0C 08      or      al,8
```

```

39 020A:077C          loc_2:
40 ;; -- Сохранение значения регистра ax
41 020A:077C  50          push    ax
42 ;; -- Декремент счетчика времени до отключения моторчика дисковод
43 020A:077D  FE 0E 0040    dec byte ptr ds:[40h] ;
      (0040:0040=0D3h)
44 020A:0781  75 0B          jnz loc_3          ; Jump if not
      zero
45 ;; -- Сброс флагов, отвечающих за работу моторчика дисковод
46 020A:0783  80 26 003F F0    and byte ptr ds:[3Fh],0F0h ;
      (0040:003F=0)
47 ;; -- Отправка команды на отключение моторчика дисковод
48 020A:0788  B0 0C          mov     al,0Ch
49 020A:078A  BA 03F2        mov     dx,3F2h
50 020A:078D  EE          out     dx,al          ; port 3F2h,
      dsk0 contrl output
51 020A:078E          loc_3:
52 ;; -- Восстановление регистра ax
53 020-- A:078E  58          pop     ax
54 ;; Проверка второго бита флага PF(Parity Flag - флаг четности)
55 ;; DS:0314 - адрес области данных BIOS, содержащей копию флагов
56 020A:078F  F7 06 0314 0004    test    word ptr ds:[314h],4
      ; (0040:0314=3200h)
57 020A:0795  75 0C          jnz loc_4          ; Jump if not
      zero
58 ;; -- Сохранение младшего байта регистра FLAGS в ah
59 020A:0797  9F          lahf          ; Load ah from
      flags
60 ;; -- Обмен значений регистров AH и AL
61 ;; Таким образом младший байт регистра FLAGS находится в младшем ба
      йте регистра ax
62 020A:0798  86 E0          xchg     ah,al
63 ;; -- Сохранение регистра ax
64 020A:079A  50          push    ax
65 ;; -- Косвенный вызов прерывания 1Ch
66 ;; В этом случае не произойдет push регистра FLAGS в стек, на его
      месте будет ax,
67 ;; который по выходу из 1ch будет установлен в FLAGS через iret
68 020A:079B  26: FF 1E 0070    call    dword ptr es:[70h] ;
      (0000:0070=6ADh)
69 020A:07A0  EB 03          jmp     short loc_5
70                                     ; now data
                                     because:
                                     after jmp/
                                     ret
71 020A:07A2  90          db      90h
72 ;; -- Вызов пользовательского прерывания через 1Ch
73 020A:07A3          loc_4:
74 020A:07A3  CD 1C          int     1Ch          ; Timer break (call
      each 18.2ms)
75 020A:07A5          loc_5:
76 020A:07A5  E8 0011          ;*      call    sub_1          ;*
77 020A:07A5  E8 11 00          db      0E8h, 11h, 00h
78 ;; -- Сброс контроллера прерываний
79 020A:07A8  B0 20          mov     al,20h          ; ' '
80 020A:07AA  E6 20          out     20h,al          ; port 20h,
      8259-1 int command

```

```

81 |                                     ; al = 20h, end of
                                     interrupt
82 | ;; -- Возврат значений регистров es,dx,ax,dx
83 | 020A:07AC  5A                      pop dx
84 | 020A:07AD  58                      pop ax
85 | 020A:07AE  1F                      pop ds
86 | 020A:07AF  07                      pop es
87 | 020A:07B0  E9 FE99                 jmp $-164h
88 | ;; ...
89 | 020A:064C  1E                      push  ds
90 | 020A:064D  50                      push  ax
91 | ;; ...
92 | 020A:06AA  58                      pop ax
93 | 020A:06AB  1F                      pop ds
94 | ;; -- Возврат из прерывания
95 | 020A:06AC  CF                      iret                ; Interrupt
    | return

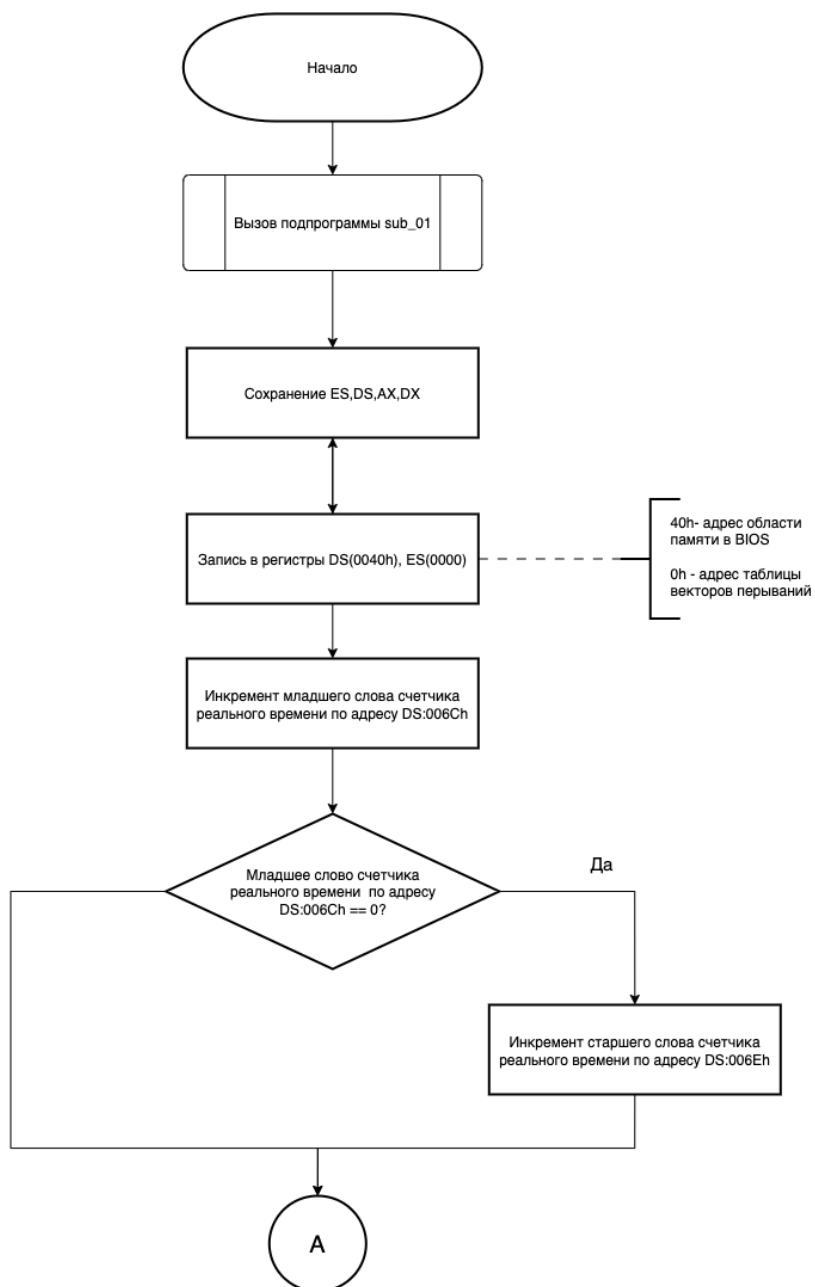
```

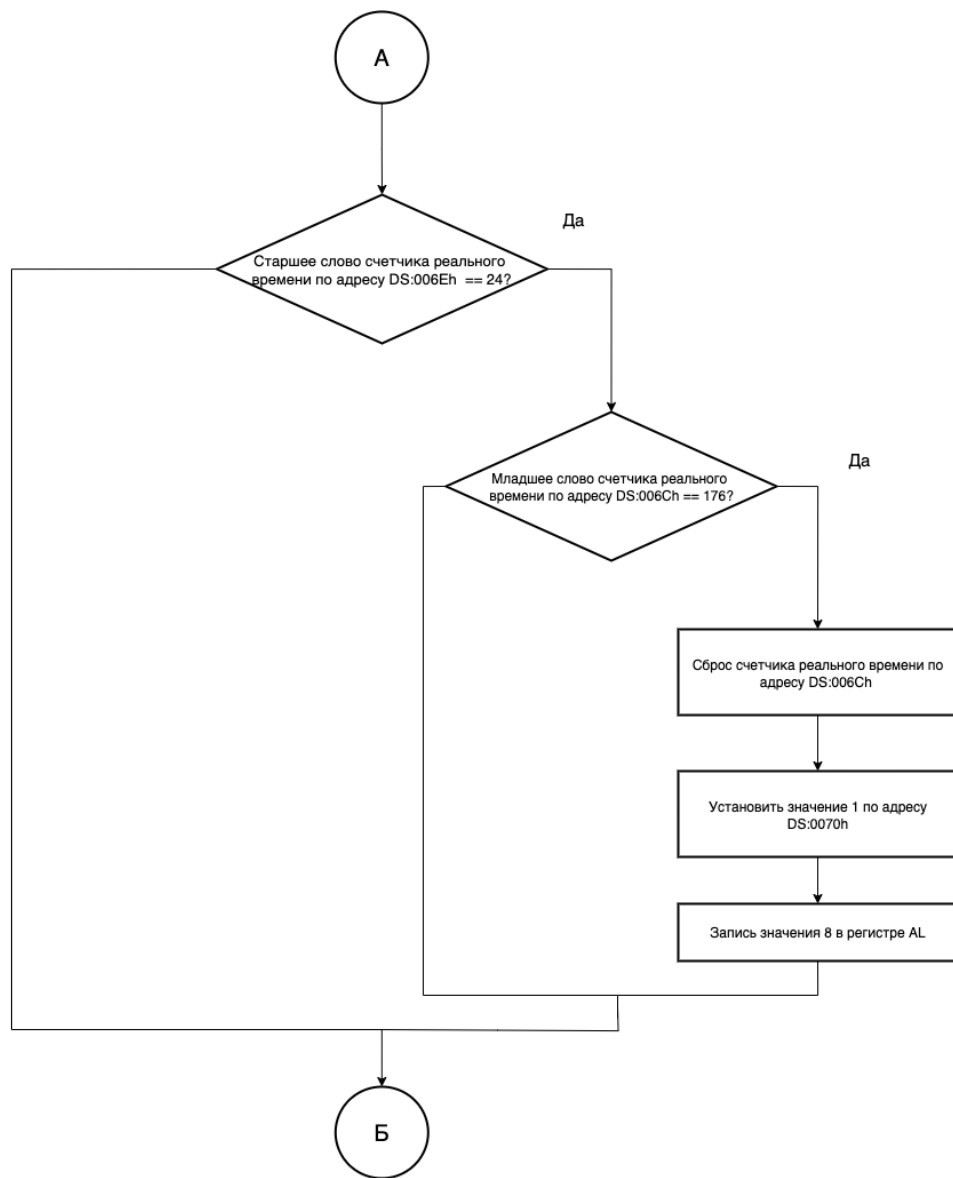
## 1.2 Листинг процедуры sub\_1

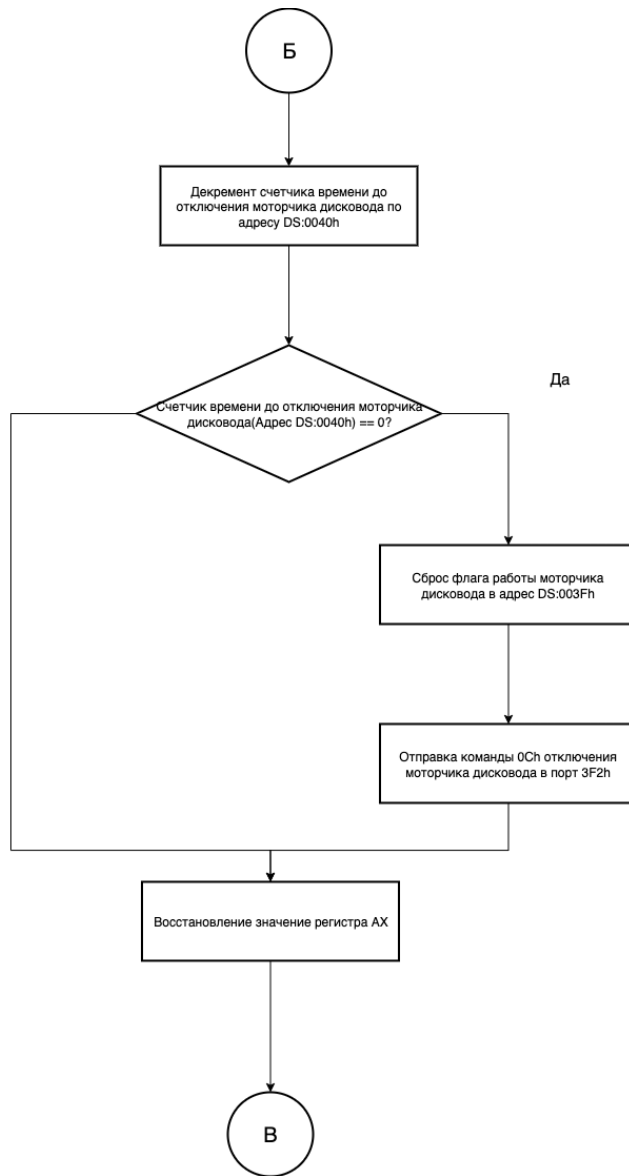
```
1          Temp.lst          Sourcer v5.10    10-Sep-25
2          7:22 pm    Page 1
3  ;; -- Сохранение регистров ds, ax
4  020A:07B9  1E                push    ds
5  020A:07BA  50                push    ax
6  ;; -- Загрузка сегментного регистра dx
7  020A:07BB  B8 0040          mov     ax,40h
8  020A:07BE  8E D8          mov     ds,ax
9  ;; -- Сохранение младшего байта регистра FLAGS в ah
10 020A:07C0  9F                a      lahf                ; Load ah from
    flags
11 ;; -- Проверка флага DF и старшего бита IORL по адресу DS:0314h
12 020A:07C1  F7 06 0314 2400    test    word ptr ds:[314h],2400
    h      ; (0040:0314=3200h)
13 020A:07C7  75 0C                jnz    loc_2                ; Jump if not
    zero
14 ;; -- Сброс флага прерываний IF в BIOS
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_1:
18 ;; -- Загрузка ah в младший байт FLAGS
19 020A:07D0  9E                sahf                ; Store ah into
    flags
20 ;; -- Восстановление регистров
21 020A:07D1  58                pop     ax
22 020A:07D2  1F                pop     ds
23 020A:07D3  EB 03                jmp     short loc_3        ; (07D8)
24 020A:07D5                loc_2:
25 ;; -- Сброс флага прерываний IF
26 020A:07D5  FA                cli                ; Disable
    interrupts
27 020A:07D6  EB F8                jmp     short loc_1        ; (07D0)
28 020A:07D8                loc_3:
29 ;; -- Возврат из подпрограммы
30 020A:07D8  C3                retn
```

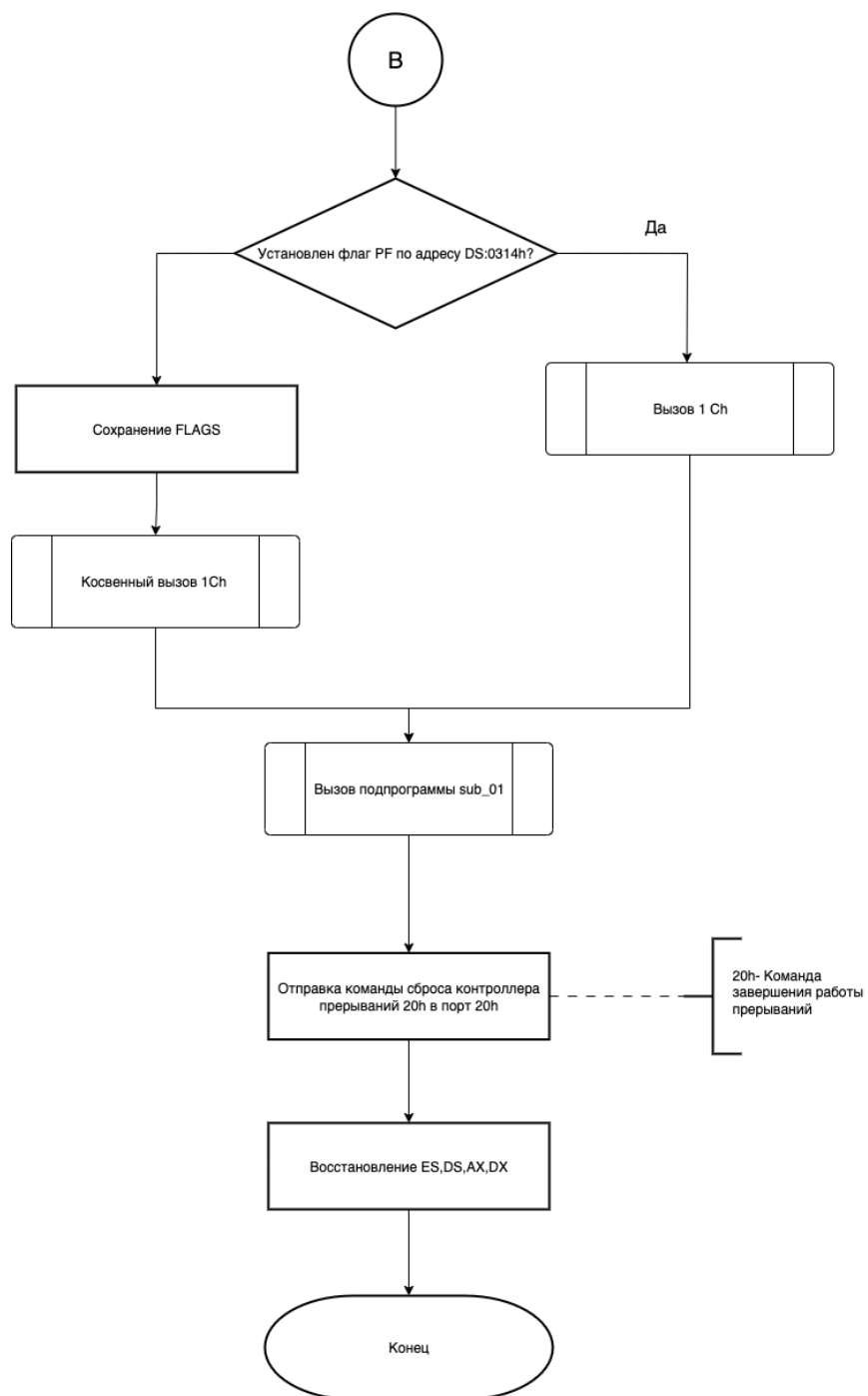
## 2 Схема алгоритмов

### 2.1 Схема алгоритма обработчика INT8h











## 2.2 Схема алгоритма процедуры sub\_1

