



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

Отчет по лабораторной работе №1 по теме "Дисассемблирование INT 8h"

Студент Недолужко Д.В.

Группа ИУ7-53Б

Преподаватель Рязанова Н.Ю.

Москва

2021 г.

1 Ассемблерный код

Листинг 1.1 — Обработчик INT 8h

```

1      = 0070      data_1e      equ      70h          ; (0000:0070=0ADh)
2      = 003F      dsk_motor_stat equ      3Fh          ; (0040:003F=0)
3      = 0040      dsk_motor_tmr equ      40h          ; (0040:0040=0D6h)
4      = 006C      timer_low     equ      6Ch          ; (0040:006C=908Ah)
5      = 006E      timer_hi      equ      6Eh          ; (0040:006E=15h)
6      = 0070      timer_rolled  equ      70h          ; (0040:0070=0)
7      = 0314      data_2e      equ      314h          ; *(0040:0314=3200h)
8
9      ; Вызов сопрограммы sub_1
10     020A:0746  E8 0070          call     sub_1          ; *(07B9)
11     020A:0746  E8 70 00          db       0E8h, 70h, 00h
12     ; Записывание в стек значений регистров es, ds, ax, dx
13     020A:0749  06                push     es
14     020A:074A  1E                push     ds
15     020A:074B  50                push     ax
16     020A:074C  52                push     dx
17     ; ds = 0040h
18     020A:074D  B8 0040          mov      ax,40h
19     020A:0750  8E D8            mov      ds,ax
20     ; ax = es = 0
21     020A:0752  33 C0            xor      ax,ax          ; Zero register
22     020A:0754  8E C0            mov      es,ax
23     ; Инкремент младшей части таймера.
24     ; Данный обработчик вызывается примерно 18.2 раз в секунду, поэтому
25     ; переполнение младшей части таймера происходит каждый час
26     ; (2^16 / 18.2 = 3600 сек. = 1 час)
27     ; Если 1 час прошел, то происходит инкремент старшей части таймера
28     020A:0756  FF 06 006C        inc      word ptr ds:timer_low ; (0040:006C=908Ah)
29     020A:075A  75 04            jnz      loc_1          ; Jump if not zero
30     020A:075C  FF 06 006E        inc      word ptr ds:timer_hi ; (0040:006E=15h)
31     020A:0760                loc_1:                ; xref 020A:075A
32     ; Если на старшей части таймера установлена значение 24, а на младшей — 176,
33     ; значит прошли сутки. В таком случае младшей и старшей частям таймера присвоим
34     ; значение 0, установим флаг timer_rolled и в регистре al установим 3й бит.
35     020A:0760  83 3E 006E 18     cmp      word ptr ds:timer_hi,18h ; (0040:006E=15h)
36     020A:0765  75 15            jne      loc_2          ; Jump if not equal
37     020A:0767  81 3E 006C 00B0   cmp      word ptr ds:timer_low,0B0h ; (0040:006C=908Ah)
38     020A:076D  75 0D            jne      loc_2          ; Jump if not equal
39     020A:076F  A3 006E          mov      ds:timer_hi,ax ; (0040:006E=15h)
40     020A:0772  A3 006C          mov      ds:timer_low,ax ; (0040:006C=908Ah)
41     020A:0775  C6 06 0070 01     mov      byte ptr ds:timer_rolled,1 ; (0040:0070=0)
42     020A:077A  0C 08            or       al,8
43     020A:077C                loc_2:                ; xref 020A:0765, 076D
44     ; Декремент счетчика дисководов. Если счетчику дисководов установлено значение 0,
45     ; то отправляется команда отключения дисководов.
46     020A:077C  50                push     ax
47     020A:077D  FE 0E 0040        dec      byte ptr ds:dsk_motor_tmr ; (0040:0040=0D6h)
48     020A:0781  75 0B            jnz      loc_3          ; Jump if not zero
49     020A:0783  80 26 003F F0     and      byte ptr ds:dsk_motor_stat,0F0h ; (0040:003F=0)
50     020A:0788  B0 0C            mov      al,0Ch
51     020A:078A  BA 03F2          mov      dx,3F2h
52     020A:078D  EE                out      dx,al          ; port 3F2h, dsk0 contrl output
53     020A:078E                loc_3:                ; xref 020A:0781
54     020A:078E  58                pop      ax

```

```

55
56 ; Проверка флага четности.
57 020A:078F F7 06 0314 0004 test word ptr ds:data_2e,4 ; (0040:0314=3200h)
58 020A:0795 75 0C jnz loc_4 ; Jump if not zero
59 020A:0797 9F lahf ; Load ah from flags
60 020A:0798 86 E0 xchg ah,al
61 020A:079A 50 push ax
62 ; Вызов 1Ch по адресу в таблице векторов
63 020A:079B 26: FF 1E 0070 call dword ptr es:data_1e ; (0000:0070=6ADh)
64 020A:07A0 EB 03 jmp short loc_5 ; (07A5)
65 020A:07A2 90 db 90h
66 020A:07A3 loc_4: ; xref 020A:0795
67 ; Вызов 1Ch
68 020A:07A3 CD 1C int 1Ch ; Timer break (call each 18.2ms)
69 020A:07A5 loc_5: ; xref 020A:07A0
70 020A:07A5 E8 0011 ;* call sub_1 ;*(07B9)
71 020A:07A5 E8 11 00 db 0E8h, 11h, 00h
72 020A:07A8 B0 20 mov al,20h ; ' '
73 020A:07AA E6 20 out 20h,al ; port 20h, 8259-1 int command
74 ; al = 20h, end of interrupt
75 020A:07AC 5A pop dx
76 020A:07AD 58 pop ax
77 020A:07AE 1F pop ds
78 020A:07AF 07 pop es
79 020A:07B0 E9 FE99 jmp $-164h
80
81 020A:064C 1E push ds
82 020A:064D 50 push ax
83 020A:064E B8 0040 mov ax,40h
84 020A:0651 8E D8 mov ds,ax
85 020A:0653 F7 06 0314 2400 test word ptr ds:data_1e,2400h ; (0040:0314=3200h)
86 020A:0659 75 4F jnz loc_8 ; Jump if not zero
87
88 020A:06AA loc_8: ; xref 020A:0659, 0665, 0679
89 020A:06AA 58 pop ax
90 020A:06AB 1F pop ds
91 020A:06AC CF iret ; Interrupt return

```

Листинг 1.2 — Сопрограмма sub_1

```

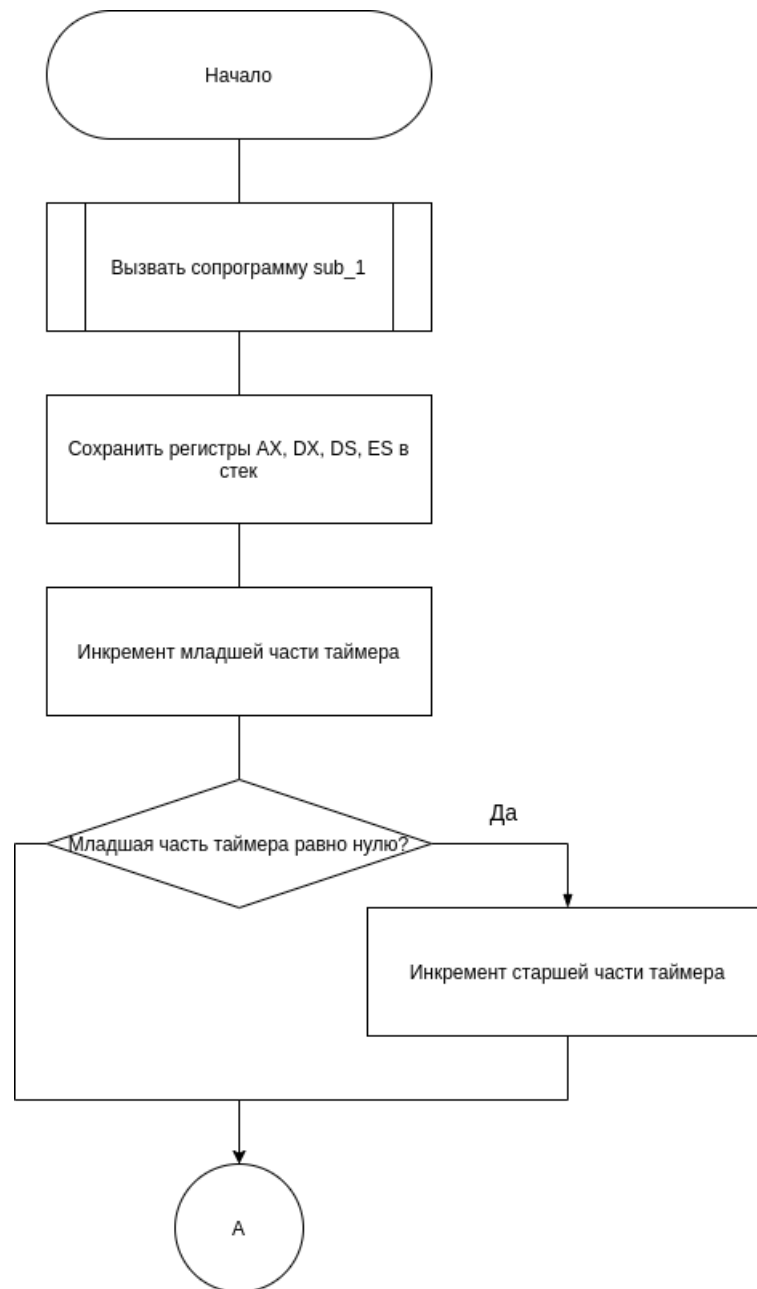
1 020A:07B9 sub_1 proc near
2 020A:07B9 1E push ds
3 020A:07BA 50 push ax
4 ; ds = ax = 0040h
5 020A:07BB B8 0040 mov ax,40h
6 020A:07BE 8E D8 mov ds,ax
7 ; Сохранение младнего байта регистра флагов в ah
8 020A:07C0 9F lahf ; Load ah from flags
9 020A:07C1 F7 06 0314 2400 test word ptr ds:data_2e,2400h ; (0040:0314=3200h)
10 020A:07C7 75 0C jnz loc_7 ; Jump if not zero
11 ; Сброс флага IF через зануления 9го бита
12
13 ; (0040:0314=3200h)
14 020A:07C9 F0> 81 26 0314 FDFF lock and word ptr ds:data_2e,0FDFFh
15 ; Установка младшему байту регистра флагов значения ah
16 020A:07D0 loc_6: ; xref 020A:07D6
17 020A:07D0 9E sahf ; Store ah into flags
18 020A:07D1 58 pop ax

```

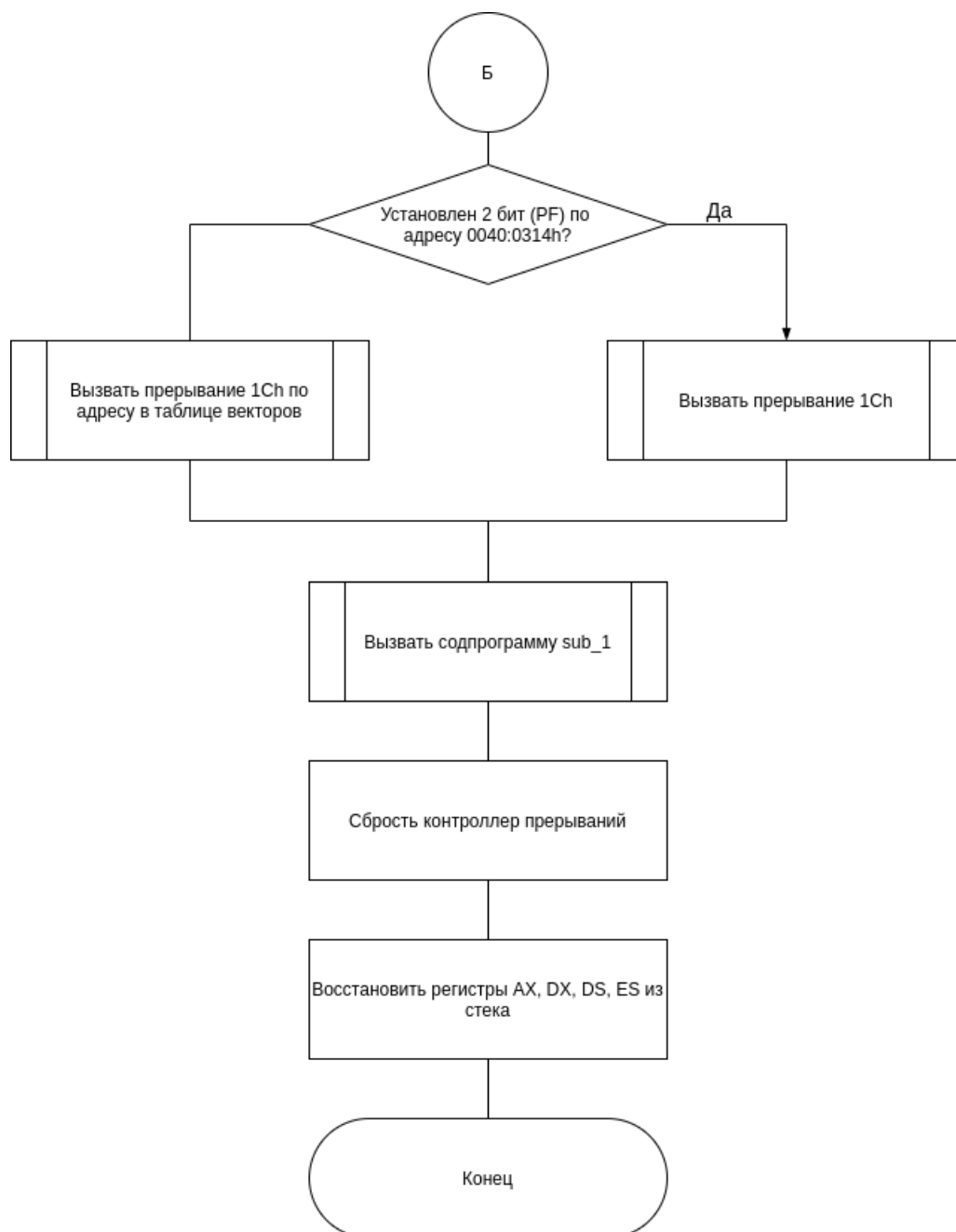
19	020A:07D2	1F	pop	ds	
20	020A:07D3	EB 03	jmp	short	loc_ret_8 ; (07D8)
21	020A:07D5			loc_7:	; xref 020A:07C7
22					; <i>Сброс флага IF</i>
23	020A:07D5	FA	cli		; <i>Disable interrupts</i>
24	020A:07D6	EB F8	jmp	short	loc_6 ; (07D0)
25					
26	020A:07D8			loc_ret_8:	; xref 020A:07D3
27	020A:07D8	C3		retn	
28		sub_1	endp		

2 Схемы алгоритмов

2.1 Схема алгоритма обработчика INT 8h







2.2 Схема алгоритма сопрограммы sub_01

