



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»

ОТЧЁТ

по лабораторной работе №1

Название: Исследование прерывания int8h

Дисциплина: Операционные системы

Группа: ИУ7-52Б

Студент: Е.О. Карпова

подпись _____

дата _____

Преподаватель: Н. Ю. Рязанова

подпись _____

дата _____

Москва — 2022 г.

Листинг

1. int 8h

Temp.lst Sourcer v5.10 10-Sep-22 13:44 pm Page

1

```
; Вызов sub_2: запрет маскируемых прерываний от внешних устройств.
020A:0746 E8 0070          call    sub_2          ; (07B9)

; Сохранение в стек регистров ES, DS, AX, DX.
020A:0749 06              push    es
020A:074A 1E              push    ds
020A:074B 50              push    ax
020A:074C 52              push    dx

; Перемещение в ds адреса сегмента данных BIOS.
020A:074D B8 0040          mov     ax,40h
020A:0750 8E D8           mov     ds,ax

; Перемещение в ES адреса начала таблицы векторов прерываний.
020A:0752 33 C0           xor     ax,ax          ; Zero register
020A:0754 8E C0           mov     es,ax

; Инкремент счётчика реального времени.
020A:0756 FF 06 006C       inc     word ptr ds:[6Ch] ; (0040:006C=0D02Dh)
020A:075A 75 04           jnz     loc_1          ; Jump if not zero
020A:075C FF 06 006E       inc     word ptr ds:[6Eh] ; (0040:006E=0Ch)

020A:0760      loc_1:

; Проверка, прошло ли 24 часа (сутки) с учётом погрешности.
020A:0760 83 3E 006E 18    cmp     word ptr ds:[6Eh],18h ; (0040:006E=0Ch)
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=0D02Dh)
020A:076D 75 0D jne     loc_2          ; Jump if not equal
```

; Сброс таймера.

020A:076F A3 006E mov word ptr ds:[6Eh],ax ; (0040:006E=0Ch)

020A:0772 A3 006C mov word ptr ds:[6Ch],ax ; (0040:006C=0D02Dh)

; По смещению 70h помещается флаг 1 - флаг о том, что прошли сутки.

020A:0775 C6 06 0070 01 mov byte ptr ds:[70h],1 ; (0040:0070=0)

; Записываем в al 8 (al = 1000).

020A:077A 0C 08 or al,8

020A:077C loc_2:

; Сохраняем ax (al = 8 если прошли сутки, иначе 0).

020A:077C 50 push ax

; Декремент счётчика реального времени, оставшегося до отключения моторчика дисководов.

020A:077D FE 0E 0040 dec byte ptr ds:[40h] ; (0040:0040=0D3h)

020A:0781 75 0B jnz loc_3 ; Jump if not zero

; Если счётчик == 0, то устанавливаем для всех четырёх приводов моторчика дисководов нулевой бит - состояние "выключен".

020A:0783 80 26 003F F0 and byte ptr ds:[3Fh],0F0h ; (0040:003F=0)

; Посылаем в порт управления дисководом команду отключения дисководов.

020A:0788 B0 0C mov al,0Ch

020A:078A BA 03F2 mov dx,3F2h

020A:078D EE out dx,al ; port 3F2h, dsk0 control output

020A:078E loc_3:

; Возвращаем регистр AX (значение регистра == 8 если новые сутки, иначе значение == 0).

020A:078E 58 pop ax

```

; Проверяем флаг чётности - разрешён ли ответ на маскируемые прерывания (1 -
разрешён, 0 - нет).
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

; Сохраняем младший байт флагов в AH.
020A:0797  9F      lahf    ; Load ah from flags

; Если прошли сутки - устанавливается флаг переполнения в 1.
020A:0798  86 E0  xchg    ah,al

; Сохранение флагов в стек.
020A:079A  50      push    ax

; Вызов пользовательского прерывания 1Ch.
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

; Вызов пользовательского прерывания 1Ch, если был разрешён ответ на маскируемые
прерывания.
020A:07A3  loc_4:
020A:07A3  CD 1C  int     1Ch ; Timer break (call each 18.2ms)

020A:07A5  loc_5:
; Вызов sub_2: запрет маскируемых прерываний от внешних устройств, так как в 1Ch
; мог измениться флаг прерываний.
020A:07A5  E8 0011 call    sub_2    ; (07B9)

; Посылаем в порт ведущего контроллера прерываний команду завершения работы
прерывания
; и готовности ответа на иные прерывания.
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

```

```
; Восстановление регистров ES, DS, AX, DX.
```

```
020A:07AC  5A      pop  dx
```

```
020A:07AD  58      pop  ax
```

```
020A:07AE  1F      pop  ds
```

```
020A:07AF  07      pop  es
```

```
; Переход на возврат из прерывания.
```

```
020A:07B0  E9 FE99  jmp  $-164h
```

2. subroutine sub_2

Temp.lst Sourcer v5.10 10-Sep-22 13:44 pm Page 2

```
;BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
```

```
; SUBROUTINE
```

```
;UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
```

```
; sub_2.
```

```
; Запрещает маскируемые прерывания от внешних устройств.
```

```
sub_2 proc      near
```

```
; Сохраняем значения регистров ds и ax в стек.
```

```
020A:07B9  1E      push  ds
```

```
020A:07BA  50      push  ax
```

```
; В ds помещаем адрес сегмента данных BIOS.
```

```
020A:07BB  B8 0040 mov  ax,40h
```

```
020A:07BE  8E D8   mov  ds,ax
```

```
; Сохраняем младший байт регистра флагов в старший байт регистра AX.
```

```
020A:07C0  9F      lahf   ; Load ah from flags
```

```
; Проверяем установку флагов порядка обработки строк и уровня привилегий  
ввода/вывода.
```

```
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
```

```
; Принудительно устанавливаем в 0 флаг разрешения прерываний, если уровень  
привилегий не позволяет сделать это командой cli.
```

```
; Сброс выполняется с блокировкой шины данных.
```

```
020A:07C9  F0> 81 26 0314 FDFF lock and word ptr ds:[314h],0FDFFh ;  
(0040:0314=3200h)
```

```
020A:07D0      loc_6:
```

```
; Восстанавливается младший байт флагов из AH.
```

```
020A:07D0  9E      sahf ; Store ah into flags
```

```
; Восстанавливаем регистры AX и DS.
```

```
020A:07D1  58      pop ax
```

```
020A:07D2  1F      pop ds
```

```
020A:07D3  EB 03  jmp short loc_8      ; (07D8)
```

```
; Устанавливаем флаг разрешения прерываний в 0.
```

```
020A:07D5      loc_7:
```

```
020A:07D5  FA      cli ; Disable interrupts
```

```
; Восстановление регистров.
```

```
020A:07D6  EB F8  jmp short loc_6      ; (07D0)
```

```
; Возврат из sub_2.
```

```
020A:07D8      loc_8:
```

```
020A:07D8  C3      retn
```

```
sub_2 endp
```

Схема алгоритма

1. Схема алгоритма обработчика прерывания int 8h

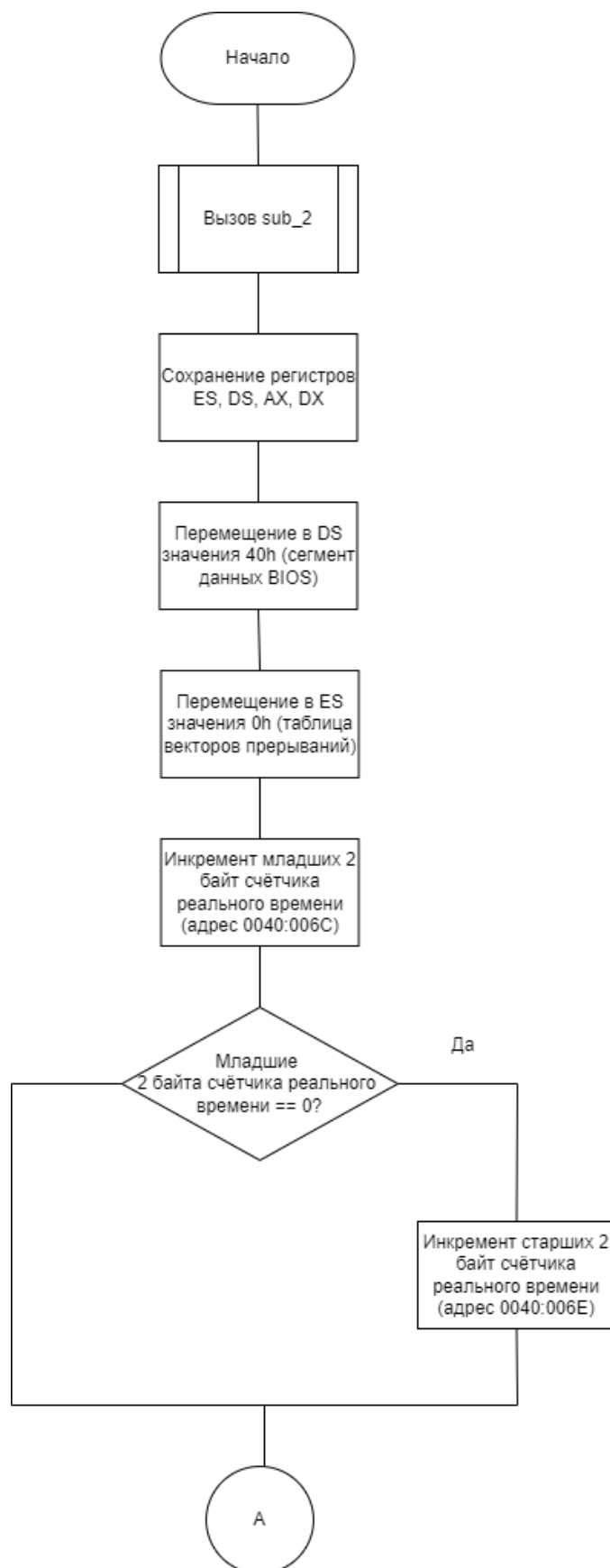


Рисунок 1 – схема алгоритма int 8h (1)

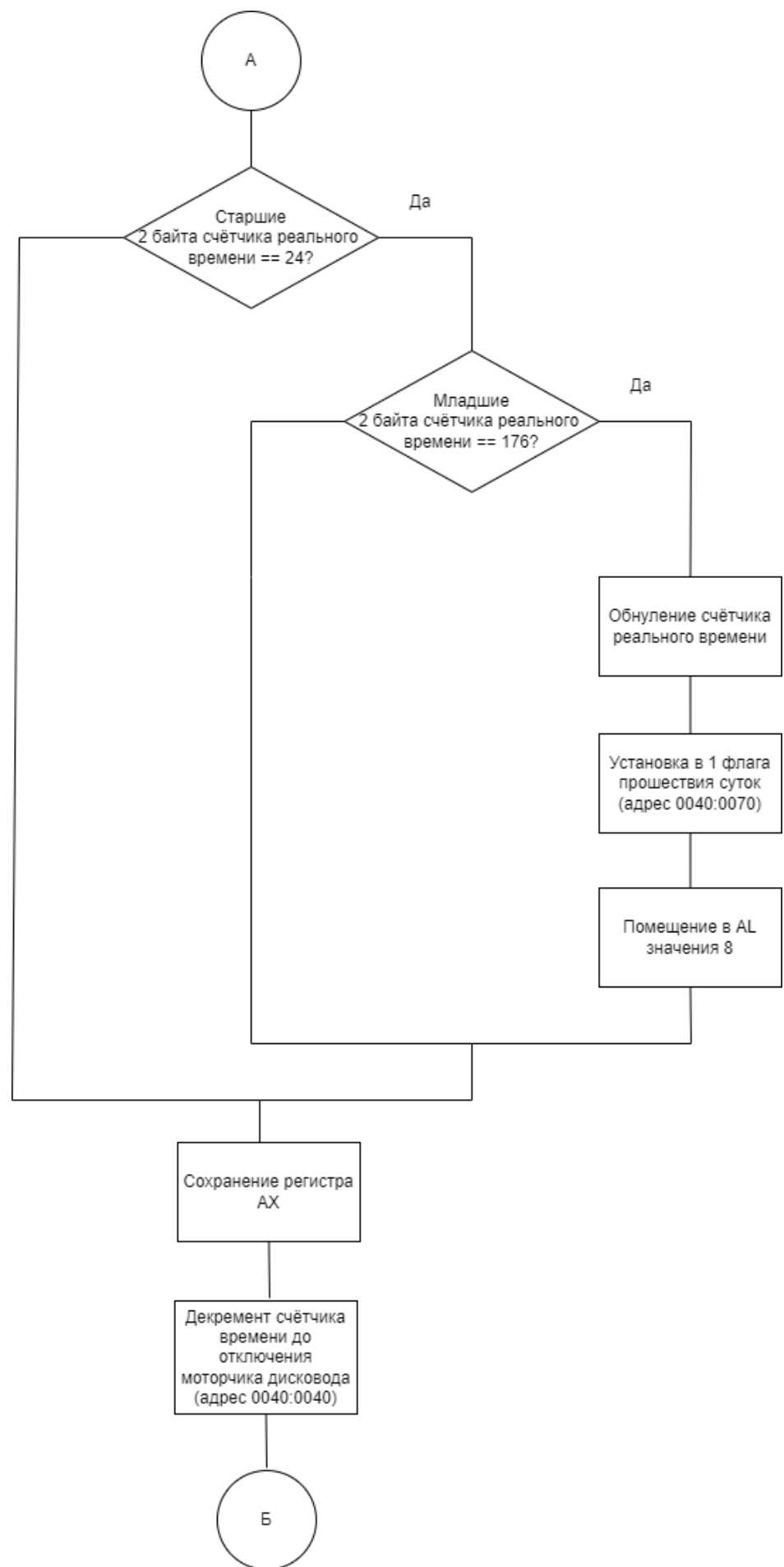


Рисунок 2 – схема алгоритма int 8h (2)

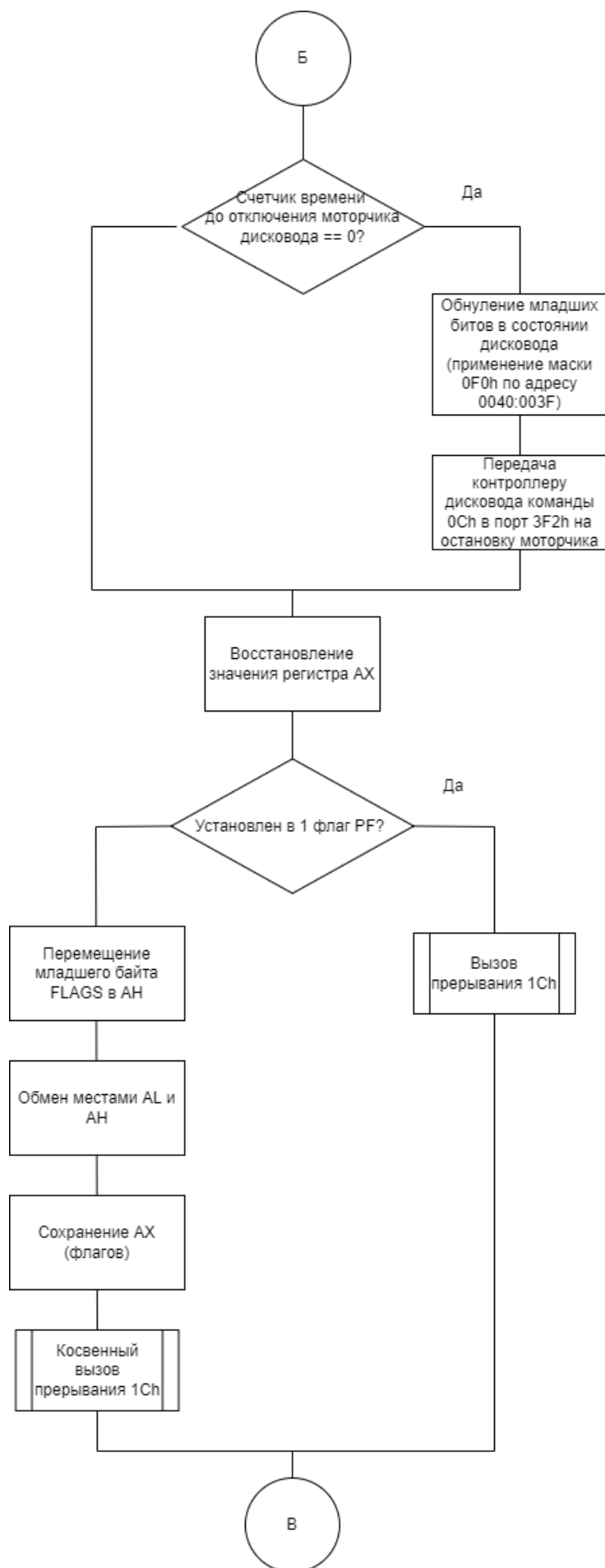


Рисунок 3 – схема алгоритма int 8h (3)

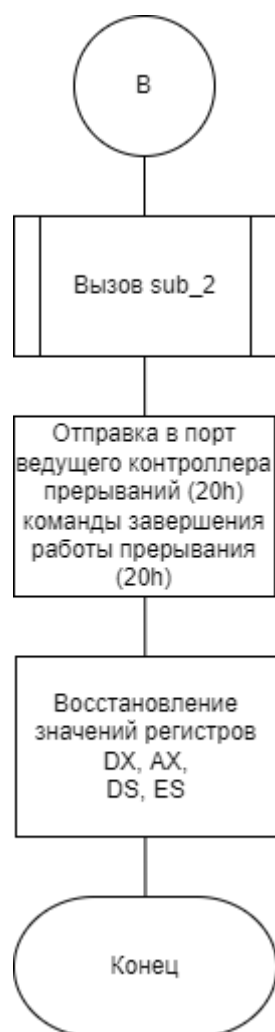


Рисунок 4 – схема алгоритма `int 8h (4)`

2. Схема алгоритма sub_2

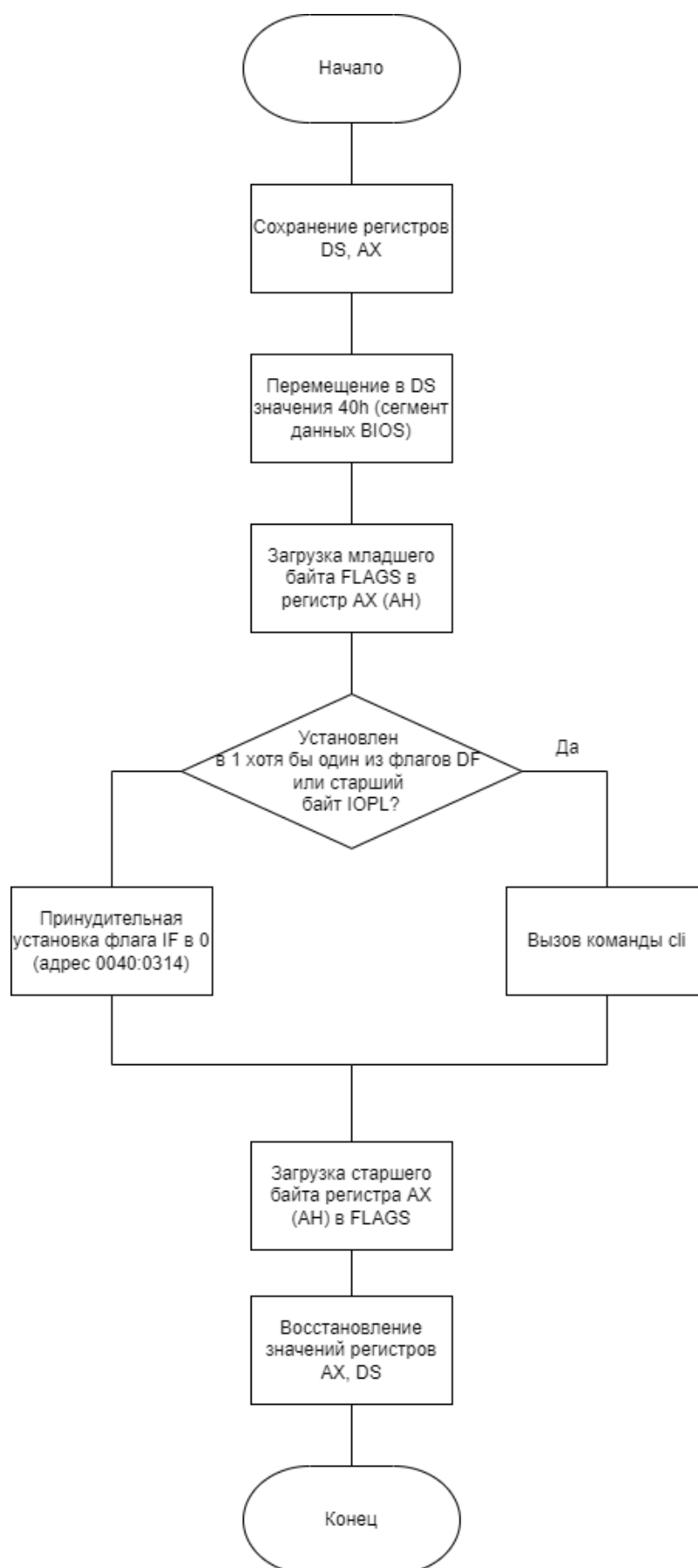


Рисунок 5 – схема алгоритма sub_2