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»

Лабораторная работа № 1 (часть 1)
По курсу «Операционные Системы»

Тема Дизассемблирование INT 8h

Студент Зайцева А. А.

Группа ИУ7-52Б

Оценка (баллы) _____

Преподаватель Рязанова Н. Ю.

Москва
2021 г.

Листинг прерывания INT 8h

```
; Вызов подпрограммы sub_2 (запрет прерываний)
020A:0746  E8 0070            call  sub_2             ; (07B9)
; Сохранение значений 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
; Инициализация DS значением 0040h (адресом начала области данных BIOS)
020A:074D  B8 0040            mov   ax,40h
020A:0750  8E D8              mov   ds,ax
; Инициализация ES значением 0 (адресом начала таблица векторов прерываний)
020A:0752  33 C0              xor   ax,ax             ; Zero register
020A:0754  8E C0              mov   es,ax
;
; ПЕРВОЕ ДЕЙСТВИЕ, выполняемое стандартным обработчиком прерывания таймера.
; Увеличение на единицу текущего значения 4-байтовой переменной, располагающейся
; по адресу 0000:046Ch – счётчика таймера. Если счетчик переполнился из-за того,
; что прошло более 24 часов с момента запуска таймера, в ячейку 0000:0470h
; заносится значение 1.
;
; Инкремент младших 2 байтов счетчика суточного времени
020A:0756  FF 06 006C         inc   word ptr ds:[6Ch] ; (0040:006C=8E8Bh)
; Если ZF==0 (счетчик не переполнился), то переход на loc_1
020A:075A  75 04              jnz   loc_1             ; Jump if not zero
; Если же ZF==1 (счетчик переполнился), значит прошел очередной час с момента
; запуска счетчика суточного времени.
; (Максимальное значение, которое могут закодировать 2 байта: 2^16-1=65535.
; Тики происходят 1193180/65536(~18.2) раза в секунду. Если счетчик переполнился,
; значит прошло 65536 тиков = 65536*65536/1193180 секунд = 3600 секунд = 1 час)
; Инкремент часов 0040:006Eh (старших 2 байтов счетчика суточного времени)
020A:075C  FF 06 006E         inc   word ptr ds:[6Eh] ; (0040:006E=14h)
020A:0760         loc_1:
; Проверка, прошли ли сутки с момента запуска счетчика суточного времени:
; В сутках 86400 секунд. Тики происходят 1193180/65536(~18.2) раза в секунду.
; В сутках 86400*1193180/65536=1573040=1800B0h тиков. То есть в старших 2 байтах
; счетчика суточного времени (по адресу 0040:006Eh, часы) должно находиться
; значение 18h(=24 часа), а в младших 2 байтах (0040:006Ch) – значение B0h.
; Если хотя бы одно из условий не выполняется, переход на loc_2
020A:0760  83 3E 006E 18      cmp   word ptr ds:[6Eh],18h    ; (0040:006E=14h)
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=8E8Bh)
020A:076D  75 0D              jne   loc_2             ; Jump if not equal
; Прошли очередные сутки с момента запуска таймера:
; 1) обнуление счетчика суточного времени
020A:076F  A3 006E            mov   word ptr ds:[6Eh],ax     ; (0040:006E=14h)
020A:0772  A3 006C            mov   word ptr ds:[6Ch],ax     ; (0040:006C=8E8Bh)
; 2) занесение единицы в ячейку 0040:0070h
020A:0775  C6 06 0070 01      mov   byte ptr ds:[70h],1      ; (0040:0070=0)
; 3) загрузка 8 в AX (AX до этого был равен нулю)
020A:077A  0C 08              or    al,8
;
; ВТОРОЕ ДЕЙСТВИЕ, выполняемое стандартным обработчиком прерывания таймера.
; Контроль за работой двигателей НГМД (накопителей на гибких магнитных дисках):
; если после последнего обращения к НГМД прошло > 2 секунд, выключение двигателя.
;
```

```
020A:077C          loc_2:
; Сохранение значения регистра AX
020A:077C  50                  push   ax
; Декремент времени, оставшегося до выключения моторчика дисковода
; (расположено в ячейке с адресом 0040:0040h)
020A:077D  FE 0E 0040          dec    byte ptr ds:[40h] ; (0040:0040=78h)
; Если ZF==1 (результат декрементирования не равен нулю), значит еще не прошло 2
; секунды после последнего обращения к НГМД. Переход на loc_3
020A:0781  75 0B               jnz    loc_3             ; Jump if not zero
; Если же ZF==0, значит после последнего обращения к НГМД прошло 2 секунды.
; Посылка в порт дисковода команды отключения моторчика дисковода
; 1) сброс соответствующих флагов моторчика дисковода (младшие 4 бита)
020A:0783  80 26 003F F0       and    byte ptr ds:[3Fh],0F0h   ; (0040:003F=0)
; 2) занесение в AL (данные для вывода в последующей команде out) значения
; 0Ch=00001100b: 2 бит=1 (разрешение работы контроллера), 3 бит=1 (разрешение
; прерываний и прямого доступа к памяти), 4-7 биты сброшены (значения 1 в каждом
; разряде вызывали бы включение соответствующего двигателя НГМД)
020A:0788  B0 0C               mov    al,0Ch
; 3) занесение в DX (приемник в последующей команде out) номер порта 3F2
; (порт цифрового управления)
020A:078A  BA 03F2             mov    dx,3F2h
; 4) вывод данных в порт
020A:078D  EE                  out    dx,al             ; port 3F2h, dsk0 contrl
output
;
; ТРЕТЬЕ ДЕЙСТВИЕ, выполняемое стандартным обработчиком прерывания таймера.
; Вызов прерывания INT 1Ch (вызывается до сброса контроллера прерывания, поэтому
; во время его выполнения все аппаратные прерывания запрещены). После
; инициализации системы вектор INT 1Ch указывает на команду IRET, то есть
; обработчик прерывания INT 1Ch ничего не делает.
;
020A:078E          loc_3:
; Восстановление значения регистра AX
020A:078E  58                  pop    ax
; Проверка флага четноси PF(0100 – 2 бит в области BIOS по адресу
; 0040:0314h, где находится копия флагов, отвечает за флаг PF)
020A:078F  F7 06 0314 0004     test   word ptr ds:[314h],4     ; (0040:0314=3200h)
; Если он установлен, переход на loc_4
020A:0795  75 0C               jnz    loc_4             ; Jump if not zero
; Иначе будет осуществлен косвенный вызов прерывания 1CH с другими флагами
; Загрузка младшего байта FLAGS в регистр AH
020A:0797  9F                  lahf                     ; Load ah from flags
; Обмен AH (младший байт FLAGS) и AL (8) -> AX=[08][младший байт FLAGS]
020A:0798  86 E0               xchg   ah,al
; Сохранение значения регистра AX
020A:079A  50                  push   ax
; Косвенный вызов прерывания 1Ch с помощью адреса в таблице векторов прерываний
; (1Ch*4=28*4=112=70H)
; При вызове командой int регистр FLAGS был бы загружен в стек, а в данном случае
; на его месте лежит AX. Тогда при выходе из прерывания 1CH именно AX будет
; установлен в FLAGS командой iret. Отличие AX и FLAGS – в старшем байте
020A:079B  26: FF 1E 0070      call   dword ptr es:[70h]; (0000:0070=6ADh)
; Переход на loc_5
020A:07A0  EB 03               jmp    short loc_5       ; (07A5)
020A:07A2  90                  nop
; Вызов прерывания 1CH
020A:07A3          loc_4:
020A:07A3  CD 1C               int    1Ch               ; Timer break (call each
18.2ms)
```

```
; Вызов подпрограммы sub_2 (запрет прерываний)
020A:07A5          loc_5:
020A:07A5  E8 0011             call  sub_2             ; (07B9)
; Сброс контроллера прерываний
; (Чтобы позволить прерываниям меньшего приоритета обрабатываться)
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 значений регистров DX, AX, DS, ES
020A:07AC  5A                  pop   dx
020A:07AD  58                  pop   ax
020A:07AE  1F                  pop   ds
020A:07AF  07                  pop   es
; Переход в сторону выхода (020A:07B0  - 164h = 020A:064C)
020A:07B0  E9 FE99             jmp   $-164h
;…
; Сохраненение значений регистров DS, AX
020A:064C  1E                  push  ds
020A:064D  50                  push  ax
;…
; Восстановление значений регистров DS, AX
020A:06AA  58                  pop   ax
020A:06AB  1F                  pop   ds
; Выход из прерывания
020A:06AC  CF                  iret                    ; Interrupt return
```

Листинг подпрограммы sub_2

```
                    sub_2       proc  near
; Сохранение значений регистров DS, AX
020A:07B9  1E                   push  ds
020A:07BA  50                   push  ax
; Инициализация DS значением 0040h (адресом начала области данных BIOS)
020A:07BB  B8 0040              mov   ax,40h
020A:07BE  8E D8                mov   ds,ax
; Загрузка младшего байта FLAGS в регистр AH
020A:07C0  9F                   lahf                        ; Load ah from flags
; Проверка: поднят ли хотя бы один из флагов 10 или 13
; (2400h = 0010 0100 0000 0000b) в области BIOS по адресу 0040:0314h, где
; находится копия флагов?
; 10 – DF – Флаг направления, контролирует поведение команд обработки строк: 1 -
; в сторону уменьшения адресов, 0 – наоборот
; 12 и 13 – IOPL – Уровень приоритета ввода/вывода.
020A:07C1  F7 06 0314 2400      test  word ptr ds:[314h],2400h ; (0040:0314=3200h)
; Если поднят хотя бы один, то переход на loc_22, чтобы командой cli сбросить
; флаг разрешения прерываний IF.
; Процессор перестанет обрабатывать прерывания от внешних устройств
; (только маскируемые, так как они вызываются по маске. Немаскируемые запретить
; нельзя, например, различные ошибки)
020A:07C7  75 0C                jnz   loc_22                ; Jump if not zero
; Если оба сброшены, то сброс IF (9 бит) командой and.
; Операция and объёмная, 2 раза обращается к памяти: считывает значение по адресу
; 0040:0314, затем изменяет его и еще раз обращается к памяти на запись.
; Необходимо, чтобы в промежуток, когда выполняется сама логическая операция,
; никто не обращался к этому участку памяти, для чего используется префиксная
; команда lock. Будет заблокирована шина данных, и, если в системе присутствует
; другой процессор, он не сможет обращаться к памяти, пока не закончится
; выполнение and
020A:07C9  F0> 81 26 0314 FDFF  lock and  word ptr ds:[314h],0FDFFh     ;
(0040:0314=3200h)
020A:07D0          loc_21:
; Восстановление значений флагов SF, ZF, AF, PF и CF регистра FLAGS из AH
020A:07D0  9E                   sahf                        ; Store ah into flags
; Восстановление значений регистров AX, DS
020A:07D1  58                   pop   ax
020A:07D2  1F                   pop   ds
; Переход на loc_23
020A:07D3  EB 03                jmp   short loc_23          ; (07D8)
020A:07D5          loc_22:
; Сброс IF
020A:07D5  FA                   cli                         ; Disable interrupts
020A:07D6  EB F8                jmp   short loc_21          ; (07D0)
; Выход из подпрограммы
020A:07D8          loc_23:
020A:07D8  C3                   retn
                    sub_2       endp
```

Схема прерывания INT 8h

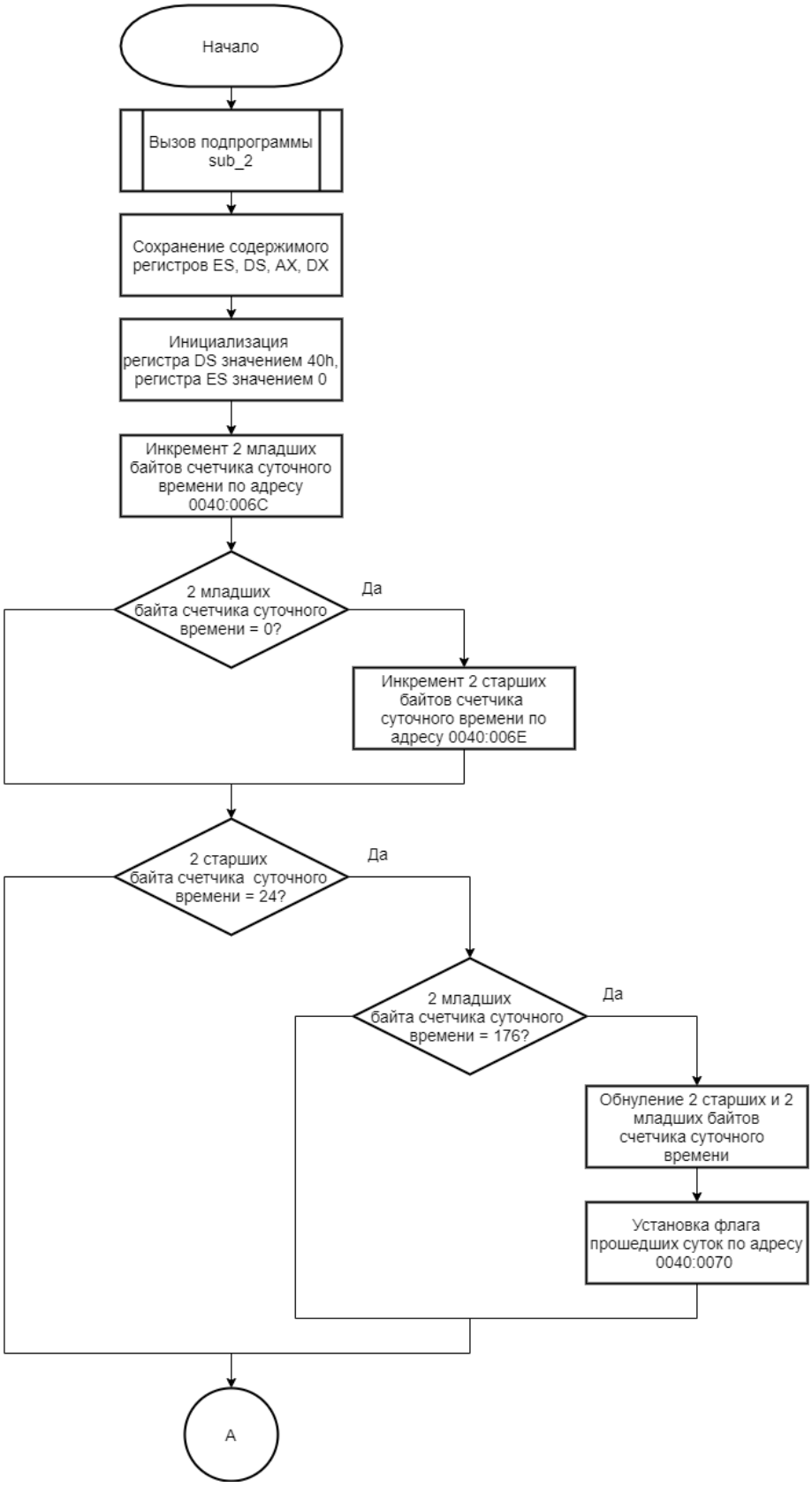

Рисунок 1 – Схема прерывания INT 8h (начало)

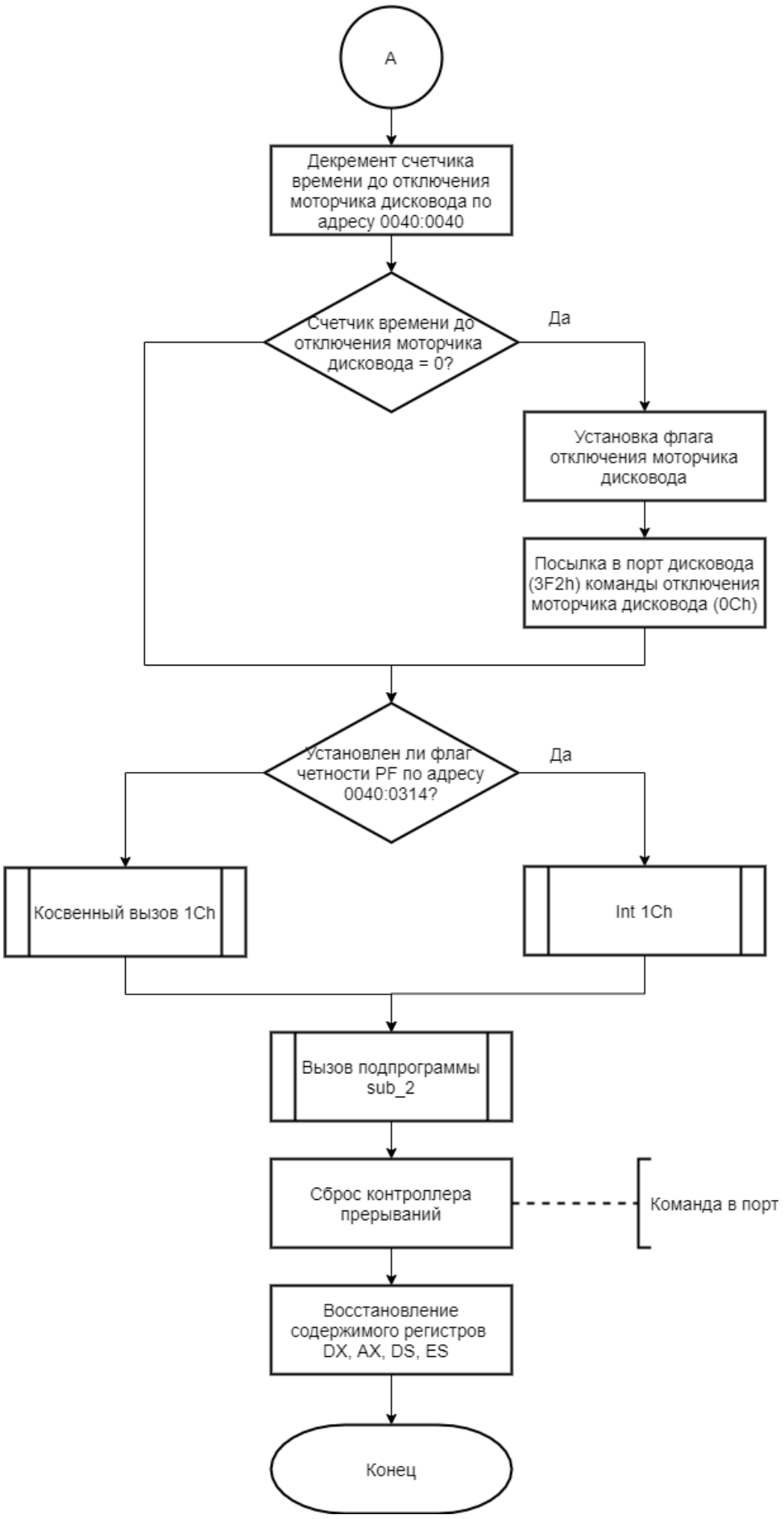

Рисунок 2 – Схема прерывания INT 8h (конец)

Схема подпрограммы sub_2

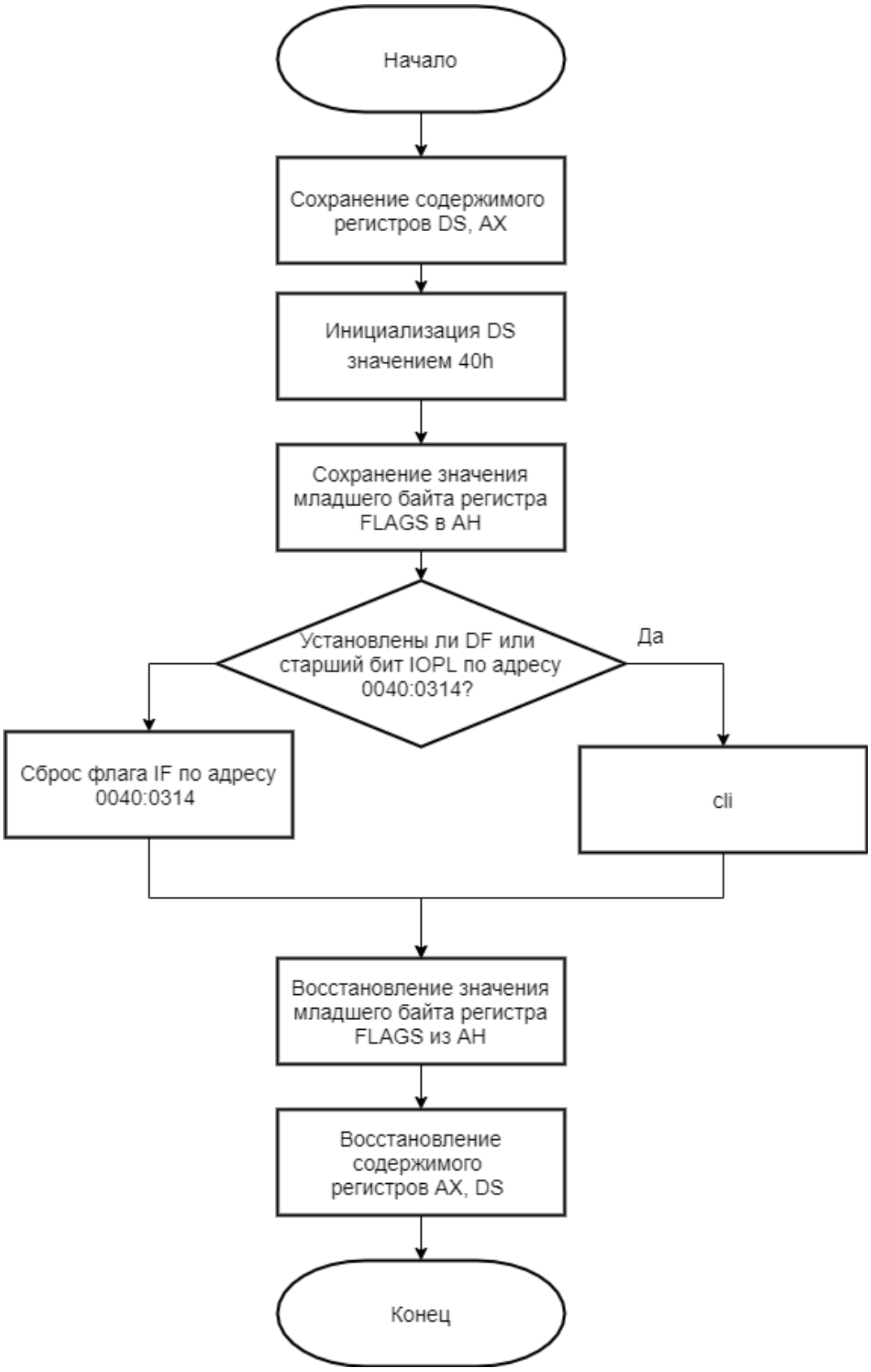

Рисунок 3 – Схема подпрограммы sub_2