Лабораторная работа №4

Ввод–вывод с квитированием по прерыванию

Файл task4in или task4out.asm

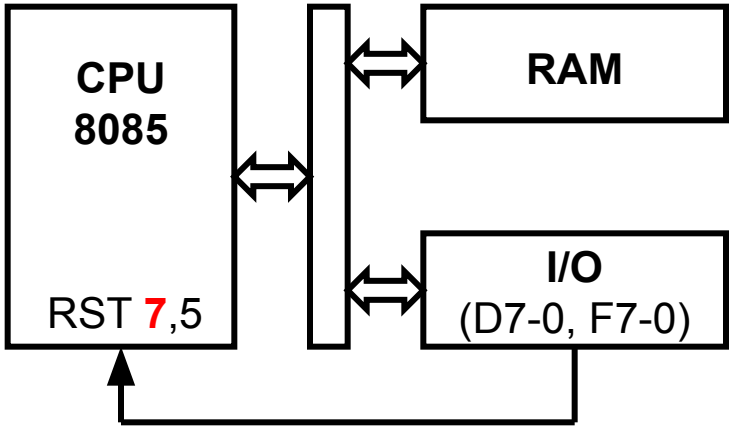
Автор: асс. каф. РС
Гайворонский Дмитрий

Постановка задачи

На языке Ассемблера написать программу **ввода** с квитированием по запросу прерывания от внешнего устройства **18** байт данных и размещения их в памяти, начиная с адреса **1234**h

Задать: **2000**h - вершина стека, **32** – размер стека

Структурная схема МП-системы

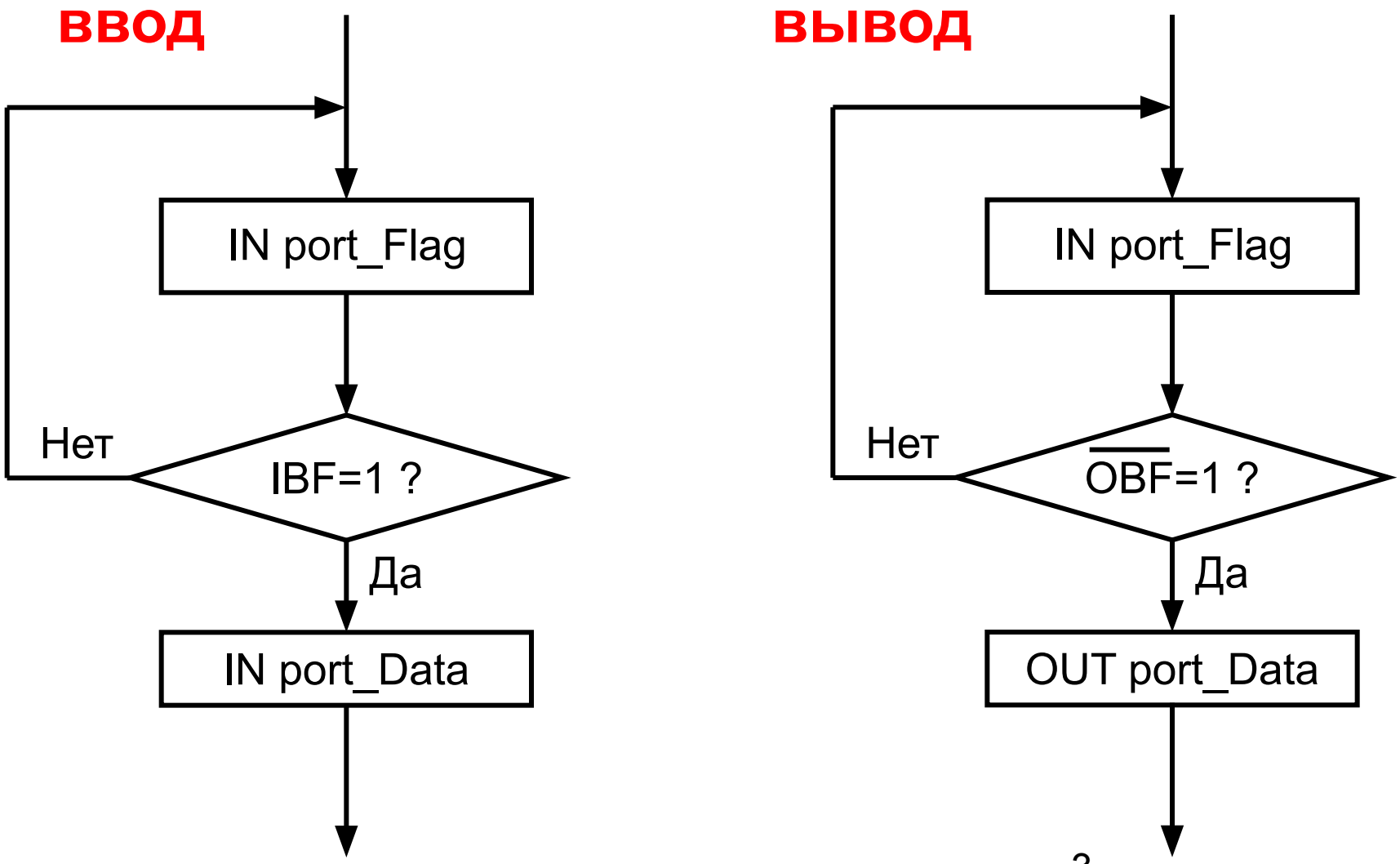


Формат байта флагов F_{7-0}

7	6	5	4	3	2	1	0
X	X	**IBF**	X	X	X	X	X

Порты I/O: **38**h для F_{7-0} и **39**h для данных D_{7-0}

Программный ввод-вывод с квитированием



Назначение разрядов аккумулятора для команды SIM

SOD	SOE	X	R7.5	MSE	M7.5	M6.5	M5.5
0	0	X	X	1	X	X	X

Сброс триггера RST7.5
0 – нет сброса
1 – сброс

Разрешение установки маски

Маска

разрешение прерывания по входу
0 – разрешение
1 - запрет

!

Адреса прерываний
5,5x8 = 44d =2Ch
6.5x8 = 52d = 34h
7.5x8 = 60d = 3Ch

Data segment

```
;DATA SEGMENT - определение сегмента памяти данных
      defseg D_seg, start = 1234h, class = Data
      seg    D_seg
bet       ds       18       ; Зарезервировать ячейки памяти M(bet)

;I/O SEGMENT - определение сегмента внешних устройств
     defseg IO_seg, start = 38h, class = IOspace
     seg    IO_seg
F_38    ds       1       ;
I_39    ds       1       ;

; Задание сегмента стека
     defseg stack_seg, start=2000h-32, class= data
     seg    stack_seg
      ds       32
```

Code segment

```
defseg rst_75, start=3Ch, class= code
    seg    rst_75
    jmp    L2

; CODE SEGMENT - сегмент кода, содержащий программу

    defseg Main_seg, start = 100h, class = Code
    seg    Main_seg

; Служебные команды для инициализации устройств
    LXI    SP,2000h
    MVI    A, 00100000b  ; A <- 20h
    OUT     38h      ; I/O(38h) <- A = 20h
    MVI    A, 46h  ; A <- 46h
    OUT     39h      ; I/O(39h) <- A = 46h
    MVI    A,00001011b
    SIM
    EI
L1:  JMP    L1
```

Code segment

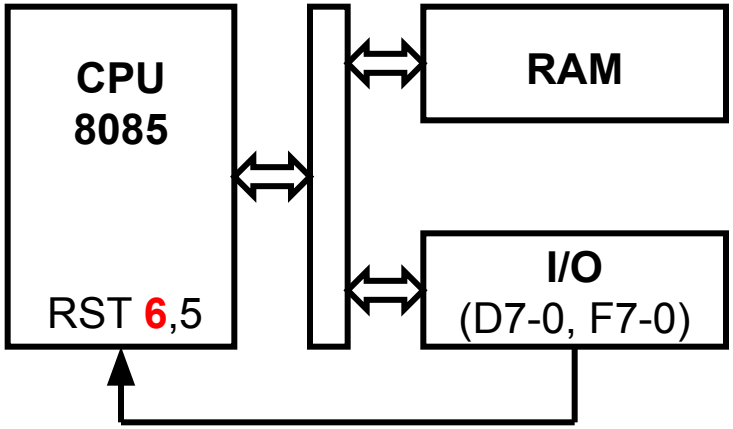
```
; ПРОГРАММА ВВОДА ДАННЫХ
L2:   PUSH   PSW
      PUSH   H
      PUSH   B
      LXI    H,bet
      MVI    C,18
L3:   IN     F_38
      XRI    00100000b
      OUT    F_38
      IN     F_38
      ANI    00100000b
      JZ     L3
      IN     I_39    ; A <- I/O(39) - ввод из внешнего устройства
      MOV    M,A     ; M(HL) <- A - запись в память по адресу
      INX    H       ; Содержимое rp H увеличить на 1
      DCR    C
      JNZ    L3
      MVI    A,00010000b
      SIM
      POP    B
      POP    H
      POP    PSW
      EI
      RET
      end           ; Конец программы
```

Постановка задачи

На языке Ассемблера написать программу **вывода** с квитированием по запросу прерывания от внешнего устройства **18** байт данных и размещения их в памяти, начиная с адреса **1234**h

Задать: **2000**h - вершина стека, **32** – размер стека

Структурная схема МП-системы



Формат байта флагов F_{7-0}

7	6	5	4	3	2	1	0
X	**OBF**	X	X	X	X	X	X

Порты I/O: **38**h для F_{7-0} и **39**h для данных D_{7-0}

Data segment

```
;DATA SEGMENT - определение сегмента памяти данных
      defseg D_seg, start = 1234h, class = Data
      seg    D_seg
bet       db       1,2,3,4,5…; Задание чисел для вывода M(bet)

;I/O SEGMENT - определение сегмента внешних устройств
     defseg IO_seg, start = 38h, class = IOspace
     seg    IO_seg
F_38   ds       1       ;
O_39   ds       1       ;

; Задание сегмента стека
     defseg stack_seg, start=2000h-32, class= data
     seg    stack_seg
      ds       32
```

Code segment

```
 defseg rst_65, start=34h, class= code
    seg    rst_65
    jmp    L2

; CODE SEGMENT - сегмент кода, содержащий программу

    defseg Main_seg, start = 100h, class = Code
    seg    Main_seg

; Служебные команды для инициализации устройств
    LXI    SP,2000h
    MVI    A, 01000000b  ; A <- 40h
    OUT    38h     ; I/O(38) <- A = 40h
    MVI    A,00001101b
    SIM
    EI
L1:  JMP    L1
```

Code segment

```
; ПРОГРАММА ВЫВОДА ДАННЫХ
L2:   PUSH   PSW
      PUSH   H
      PUSH   B
      LXI    H,bet
      MVI    C,18
L3:   IN     F_38
      XRI    01000000b
      OUT    F_38
      IN     F_38
      ANI    01000000b
      JNZ    L3
      MOV    A,M
            OUT    O_39    ; I/O(39) <- A - вывод во внешнее устройство
      INX    H       ; Содержимое rp H увеличить на 1
      DCR    C
      JNZ    L3
      POP    B
      POP    H
      POP    PSW
      EI
      RET
      end            ; Конец программы
```

Командный файл task4in.cmd

LAtask4in	Load Avocet task4in (загрузка файла task4in)
D1A**1234**h	Dump 1 Address 1234h (настройка окна памяти отладчика)
D2AI:**38**h	Dump 2 Address I/O 38h (настройка окна I/O отладчика)
← 0100	Адрес входа в программу 100h
←SY	Включение счетчика циклов (счет тактов выполнения программы)