

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ
«МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ»**



**Институт Интеллектуальных Кибернетических Систем
Кафедра «Компьютерные системы и технологии»**

**Отчёт о лабораторной работе №3
«Обработчик внешнего прерывания»**

Студент группы Б20-503 Коломенский В. Г. / _____ /
Руководитель _____ / _____ /

Москва 2023

Оглавление

1. Введение.....	1
2. Реализация программы	3
3. Тестирование программы.....	6
4. Список литературы и ссылки.....	10

1. Введение

По заданию лабораторной работы необходимо создать программу, в которой циклически опрашивается рабочий регистр и выводится его содержимое на P4.

При выполнении работы необходимо разработать обработчик внешнего прерывания INT0, выполняющий логическую или арифметическую операцию над операндами из памяти, адрес нахождения которых задан в управляющем слове. Считывание управляющего слова происходит из внешней памяти.

Полное условие представлено на рисунках 1.1 и 1.2.

Работа №3 Обработчик внешнего прерывания

Задание:

Основная программа циклически опрашивает:

1. Буферный регистр РГ1_{буф} и выводит его содержимое в порт P4 для индикации.
2. Ячейку внешней памяти данных с адресом 7FFAh, в 5-ти младших битах содержится управляющее слово следующего формата:

D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	C	B1	B0	A1	A0

где C – бит выбора операции,
B1B0 – относительный адрес операнда В, извлекаемого из массива В;
A1A0 – относительный адрес операнда А, извлекаемого из массива А;

Управляющее слово записывается в буферный регистр РГ2_{буф}

Массивы А и В, в каждом из которых по четыре элемента, хранятся во внешней памяти данных. Базовые адреса массивов А и В выбираются в адресном пространстве внешней памяти данных начиная с адреса 8000h.

Требуется разработать обработчик внешнего прерывания INT0, который в зависимости от значения бита С в управляющем слове выполняет одну из 2-х операций над операндами массивов А и В, адреса которых указаны в управляющем слове, которое обработчиком считывается из буферного регистра РГ2_{буф}

Результаты выполнения операций обработчиком прерывания для вывода на индикацию записываются в рабочий регистр РГ1_{буф}.

Рисунок 1.1 – Общие положения условия

ВАРИАНТ 28

Коломенский

$C = 0$	Определить значение наибольшего целого, которое меньше, чем корень из операнда «А» и меньше, чем операнд «В»
$C = 1$	Построить зеркальное отображение операнда «А» и выполнить исключающее ИЛИ с операндом «В».

Рисунок 1.2 – Условие варианта 28

2. Реализация программы

Реализация программы представлена в приложении 1 ниже.

Приложение 1.

```
1      P4      EQU      0C0h      ; Define P4
2      ORG      0000h      ; \ Link to the beginning
3      JMP      PREP      ; / of the program
4      ORG      0003h      ; \ Reference to the interrupt
5      JMP      INTIN      ; / handler function int0
6
7
8  PREP:
9      MOV      DPTR,      #8000h      ; \
10     MOV      A,          #04h      ; |
11     MOVX     @DPTR,      A          ; | Loading array addresses
12     INC      DPTR        ; | into external memory
13     MOV      A,          #12h      ; |
14     MOVX     @DPTR,      A          ; /
15
16     MOV      DPTR,      #8000h      ; \
17     MOVX     A,          @DPTR      ; |
18     MOV      DPL,        A          ; |
19     MOV      A,          #0FFh      ; |
20     MOVX     @DPTR,      A          ; |
21     INC      DPTR        ; |
22     MOV      A,          #0ADh      ; | Loading array "A"
23     MOVX     @DPTR,      A          ; | into external memory
24     INC      DPTR        ; |
25     MOV      A,          #64h      ; |
26     MOVX     @DPTR,      A          ; |
27     INC      DPTR        ; |
28     MOV      A,          #79h      ; |
29     MOVX     @DPTR,      A          ; /
30
31     MOV      DPTR,      #8001h      ; \
32     MOVX     A,          @DPTR      ; |
33     MOV      DPL,        A          ; |
34     MOV      A,          #17h      ; |
35     MOVX     @DPTR,      A          ; |
36     INC      DPTR        ; |
37     MOV      A,          #07h      ; | Loading array "B"
38     MOVX     @DPTR,      A          ; | into external memory
39     INC      DPTR        ; |
40     MOV      A,          #0Ah      ; |
41     MOVX     @DPTR,      A          ; |
42     INC      DPTR        ; |
43     MOV      A,          #0Dh      ; |
44     MOVX     @DPTR,      A          ; /
45
46     SETB     EA          ; \
47     SETB     EX0        ; | Permission to catch interrupts
```

```

48     SETB     IT0                ; /
49
50
51 MAIN:
52     MOV      P4,      R1        ; \
53     MOV      DPTR,    #7FFAh   ; | Updating the result in port
54     MOVX     A,       @DPTR     ; | P4 and updating the control
55     MOV      R2,      A         ; | word to register R2 in the loop
56     JMP      MAIN          ; /
57
58
59 INTIN:
60     MOV      A,       R2        ; \
61     ANL      A,       #03h      ; |
62     MOV      R0,      A         ; |
63     MOV      DPTR,    #8000h    ; | Reading a number from array
64     MOVX     A,       @DPTR     ; | "A" according to the control word
65     ADD      A,       R0        ; | and write it in R3
66     MOV      DPL,     A         ; |
67     MOVX     A,       @DPTR     ; |
68     MOV      R3,      A         ; /
69
70     MOV      A,       R2        ; \
71     ANL      A,       #0Ch      ; |
72     RR       A          ; |
73     RR       A          ; |
74     MOV      R0,      A         ; | Reading a number from array
75     MOV      DPTR,    #8001h    ; | "B" according to the control word
76     MOVX     A,       @DPTR     ; | and write it in R4
77     ADD      A,       R0        ; |
78     MOV      DPL,     A         ; |
79     MOVX     A,       @DPTR     ; |
80     MOV      R4,      A         ; /
81
82     MOV      A,       R2        ; \ Checking the
83     JNB      ACC.4,    F1        ; | operation bit
84     JMP      F2              ; / in the control word
85
86
87 F1:
88     MOV      R0,      #0        ; Performing the operation at C=0
89 F1_1:
90     MOV      A,       R4        ; |
91     CLR      C          ; |
92     SUBB     A,       R0        ; |
93     JC       F1_2        ; |
94     MOV      A,       R0        ; |
95     MOV      B,       A         ; |
96     MUL      AB         ; |
97     MOV      R5,      A         ; |
98     MOV      R6,      B         ; |
99     MOV      A,       R3        ; |
100    CLR      C          ; |
101    SUBB     A,       R5        ; |
102    JC       F1_2        ; |
103    MOV      A,       R6        ; |
104    JNZ      F1_2        ; |

```

```

105     MOV     A,      R0      ;
106     INC     A              ;
107     MOV     R0,     A       ;
108     JMP     F1_1           ;
109 F1_2:                          ;
110     MOV     A,      R0      ;
111     DEC     A              ;
112     MOV     R1,     A       ; ^
113     JMP     INTOUT        ; |
114
115
116 F2:
117     MOV     A,      R3      ; Performing the operation at C=1
118     MOV     C,      ACC.0   ; |
119     MOV     07h,    C       ; V
120     MOV     C,      ACC.1   ;
121     MOV     06h,    C       ;
122     MOV     C,      ACC.2   ;
123     MOV     05h,    C       ;
124     MOV     C,      ACC.3   ;
125     MOV     04h,    C       ;
126     MOV     C,      ACC.4   ;
127     MOV     03h,    C       ;
128     MOV     C,      ACC.5   ;
129     MOV     02h,    C       ;
130     MOV     C,      ACC.6   ;
131     MOV     01h,    C       ;
132     MOV     C,      ACC.7   ;
133     MOV     00h,    C       ;
134     MOV     A,      20h     ;
135     XRL     A,      R4      ;
136     MOV     R1,     A       ; ^
137     JMP     INTOUT        ; |
138
139
140 INTOUT:
141     RETI                          ; Return from Interrupt Handling
142
143 END

```

3. Тестирование программы

Разработанная программа тестировалась на примерах, представленных на рисунке 2.

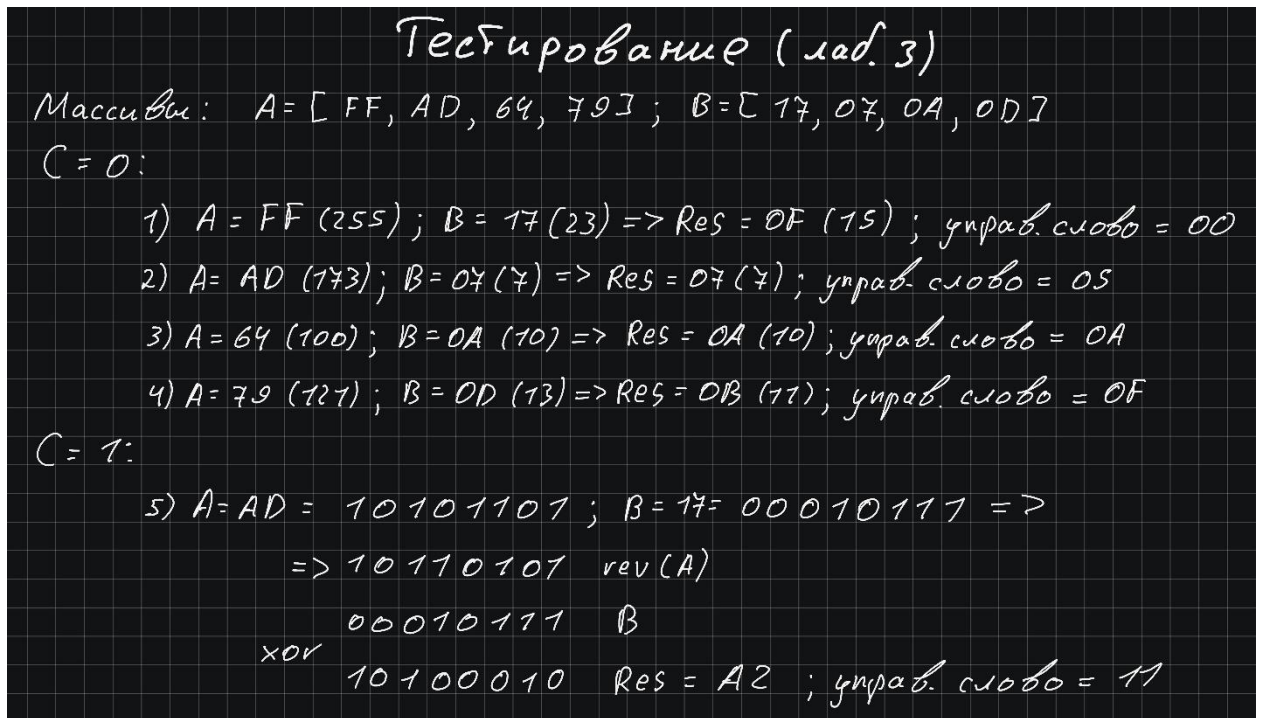


Рисунок 2. –Тестовые примеры

Результаты тестирования разработанной программы можно увидеть на рисунках 3.1 – 3.5 ниже.

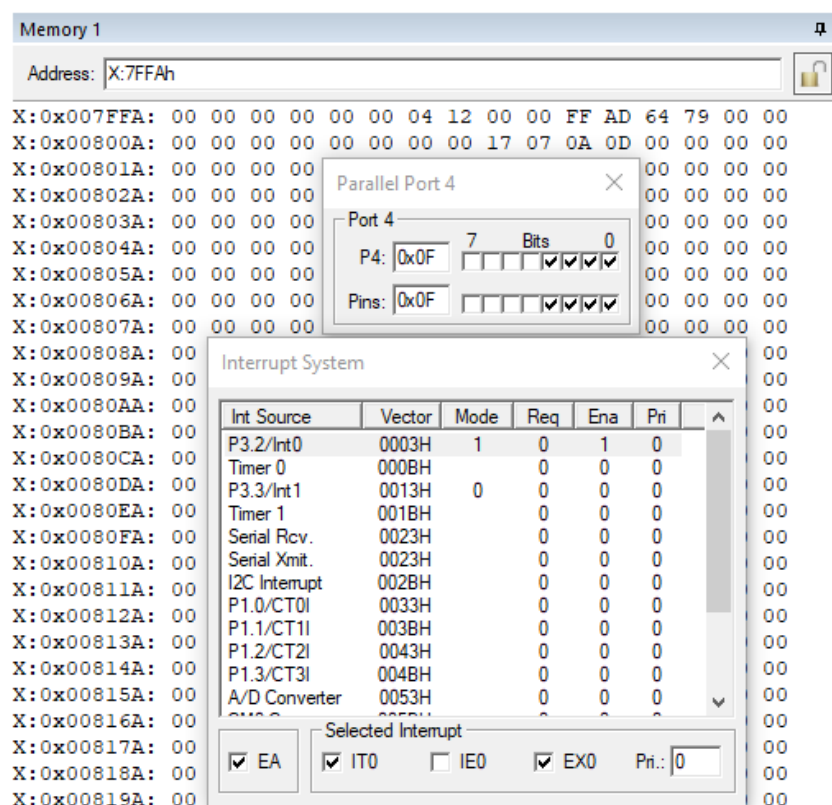


Рисунок 3.1 – Тестовый пример № 1

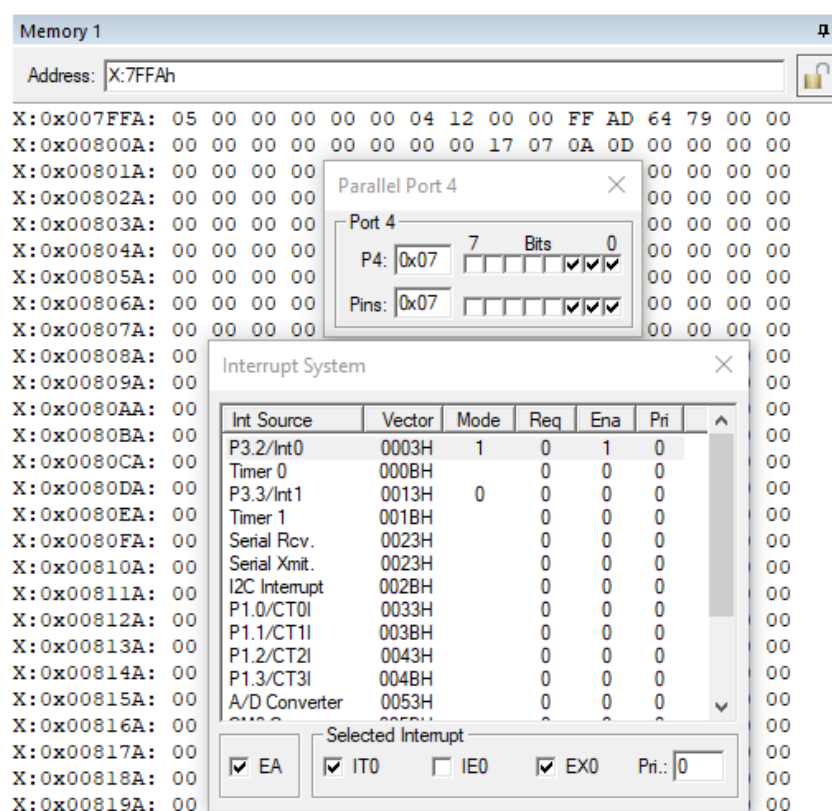


Рисунок 3.2 – Тестовый пример № 2

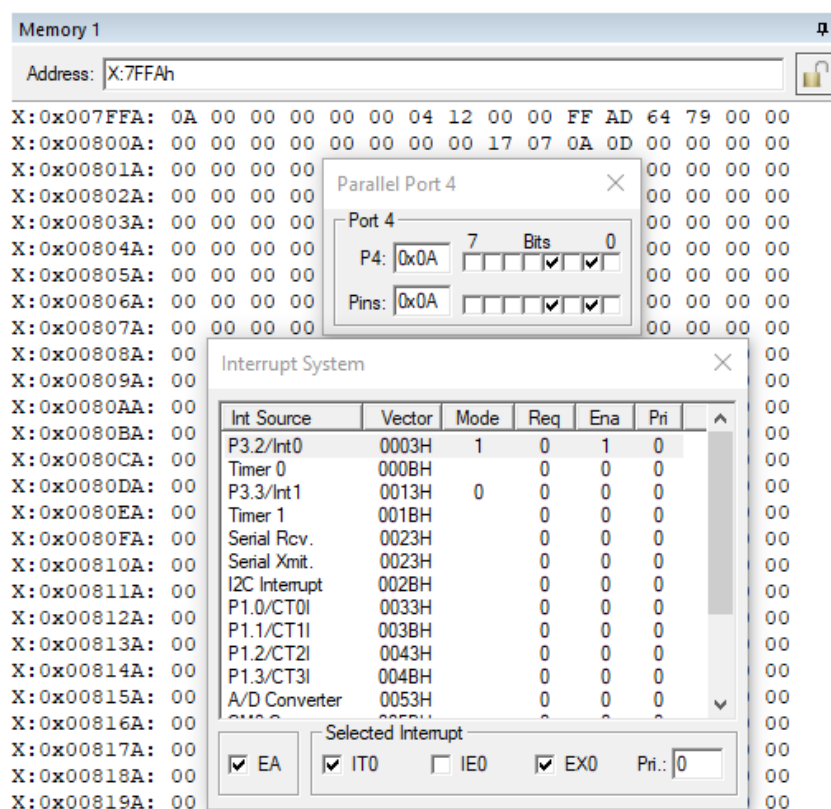


Рисунок 3.3 – Тестовый пример № 3

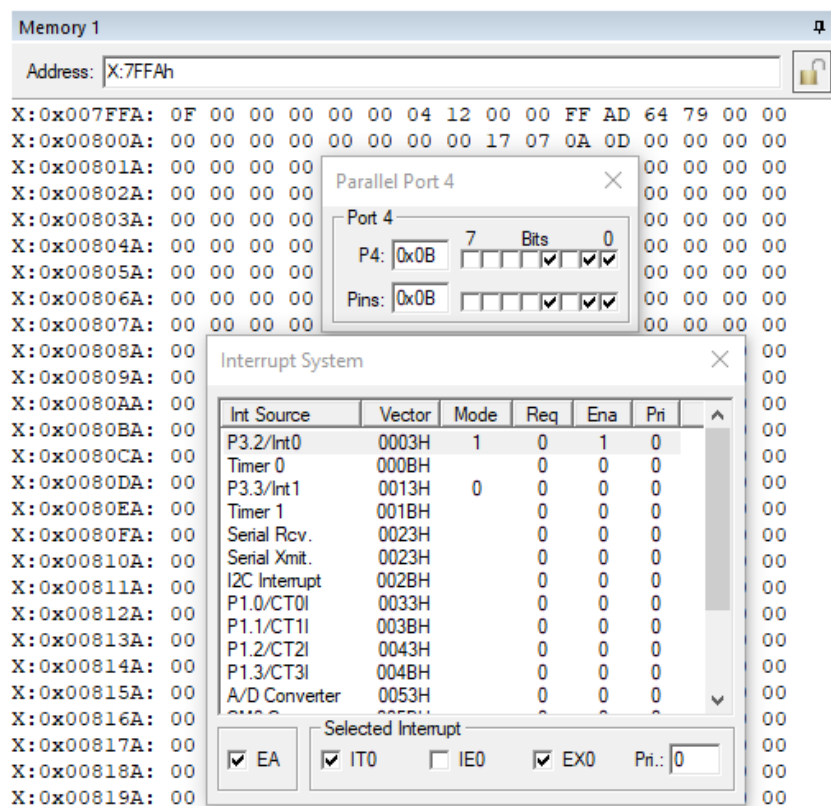


Рисунок 3.4 – Тестовый пример № 4

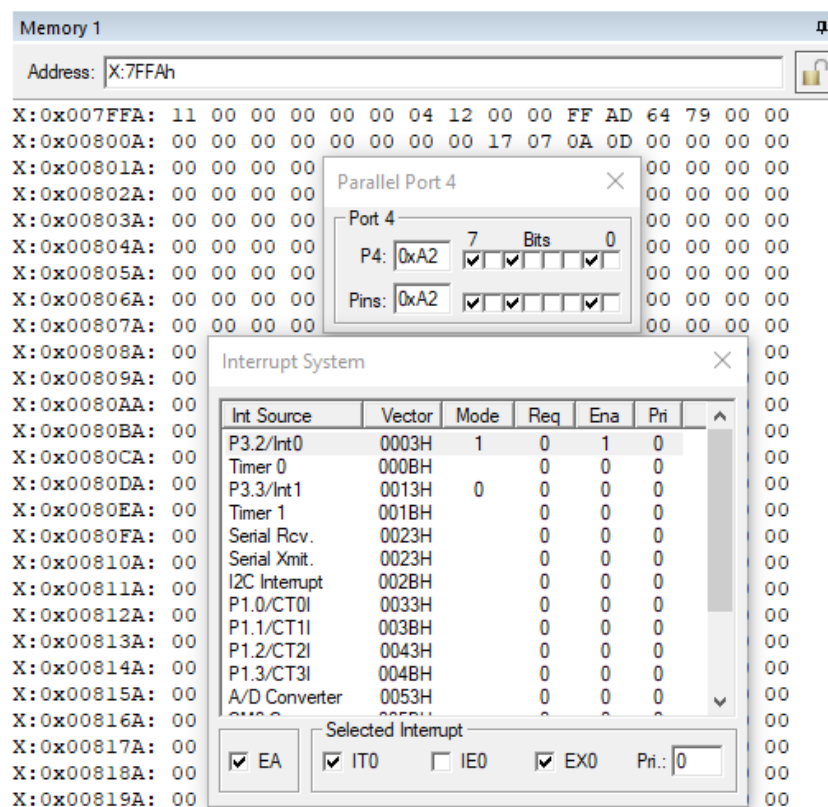


Рисунок 3.5 – Тестовый пример № 5

4. Список литературы и ссылки

1. Стрелец А. И., Иванников В. С., Ёхин М.Н. Методические указания для выполнения лабораторной работы “Битовый процессор” по курсу “Микропроцессорные устройства и системы” с использованием виртуального стенда. Москва 2018.
2. Е. В. Моисейкин. Микроконтроллеры семейства MCS-51 Теория и практика. Учебно-методическое пособие. Екатеринбург Издательство Уральского университета 2017.
3. Исходный код программы [Электронный ресурс] // <https://github.com/Hypex146/MDnS-Lab-3>.