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. Введение1

2. Реализация программы3

3. Тестирование программы6

4. Список литературы и ссылки10

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.

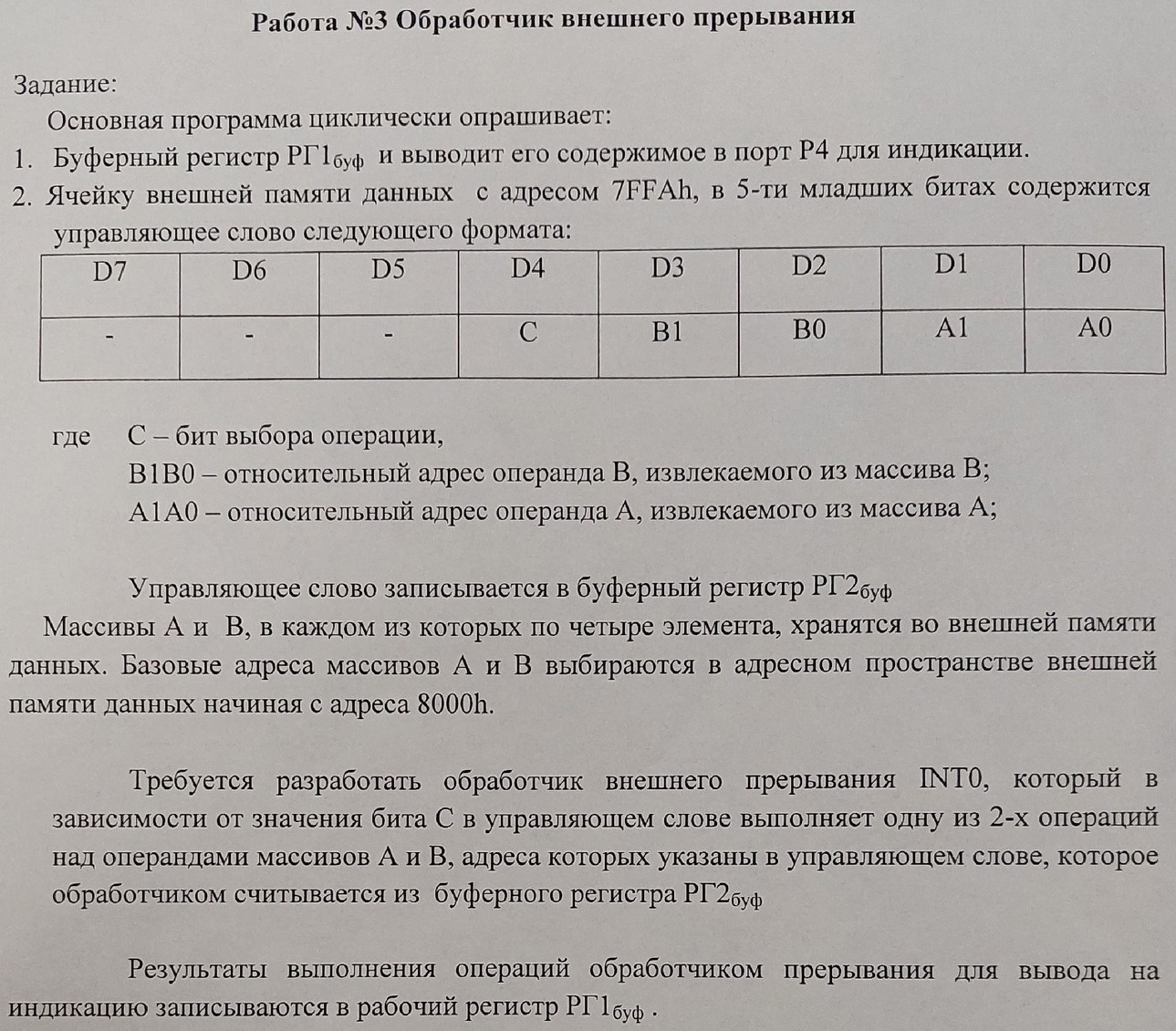


Рисунок 1.1 – Общие положения условия

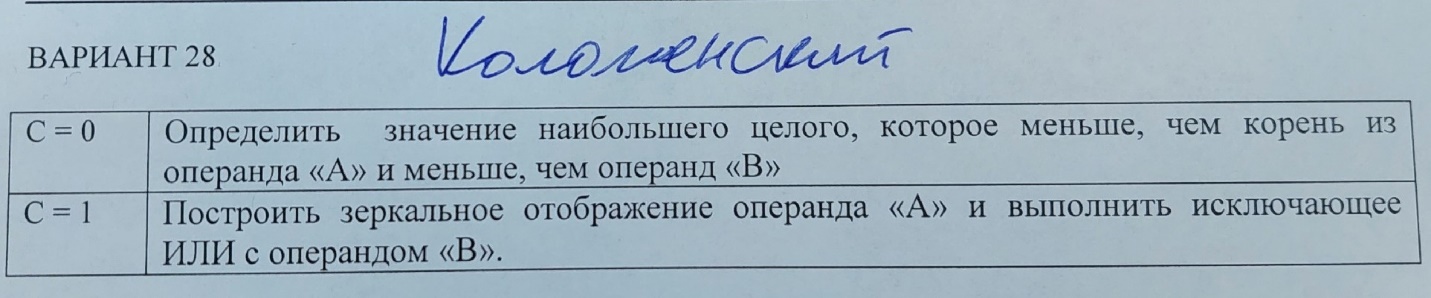


Рисунок 1.2 – Условие варианта 28

2. Реализация программы

Реализация программы представлена в приложении 1 ниже.

Приложение 1.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143 | P4 EQU 0C0h ; Define P4  ORG 0000h ; \ Link to the beginning  JMP PREP ; / of the program  ORG 0003h ; \ Reference to the interrupt  JMP INTIN ; / handler function int0  PREP:  MOV DPTR, #8000h ; \  MOV A, #04h ; |  MOVX @DPTR, A ; | Loading array addresses  INC DPTR ; | into external memory  MOV A, #12h ; |  MOVX @DPTR, A ; /  MOV DPTR, #8000h ; \  MOVX A, @DPTR ; |  MOV DPL, A ; |  MOV A, #0FFh ; |  MOVX @DPTR, A ; |  INC DPTR ; |  MOV A, #0ADh ; | Loading array "A"  MOVX @DPTR, A ; | into external memory  INC DPTR ; |  MOV A, #64h ; |  MOVX @DPTR, A ; |  INC DPTR ; |  MOV A, #79h ; |  MOVX @DPTR, A ; /  MOV DPTR, #8001h ; \  MOVX A, @DPTR ; |  MOV DPL, A ; |  MOV A, #17h ; |  MOVX @DPTR, A ; |  INC DPTR ; |  MOV A, #07h ; | Loading array "B"  MOVX @DPTR, A ; | into external memory  INC DPTR ; |  MOV A, #0Ah ; |  MOVX @DPTR, A ; |  INC DPTR ; |  MOV A, #0Dh ; |  MOVX @DPTR, A ; /  SETB EA ; \  SETB EX0 ; | Permission to catch interrupts  SETB IT0 ; /  MAIN:  MOV P4, R1 ; \  MOV DPTR, #7FFAh ; | Updating the result in port  MOVX A, @DPTR ; | P4 and updating the control  MOV R2, A ; | word to register R2 in the loop  JMP MAIN ; /  INTIN:  MOV A, R2 ; \  ANL A, #03h ; |  MOV R0, A ; |  MOV DPTR, #8000h ; | Reading a number from array  MOVX A, @DPTR ; | "A" according to the control word  ADD A, R0 ; | and write it in R3  MOV DPL, A ; |  MOVX A, @DPTR ; |  MOV R3, A ; /  MOV A, R2 ; \  ANL A, #0Ch ; |  RR A ; |  RR A ; |  MOV R0, A ; | Reading a number from array  MOV DPTR, #8001h ; | "B" according to the control word  MOVX A, @DPTR ; | and write it in R4  ADD A, R0 ; |  MOV DPL, A ; |  MOVX A, @DPTR ; |  MOV R4, A ; /  MOV A, R2 ; \ Checking the  JNB ACC.4, F1 ; | operation bit  JMP F2 ; / in the control word  F1:  MOV R0, #0 ; Performing the operation at C=0  F1\_1: ; |  MOV A, R4 ; V  CLR C ;  SUBB A, R0 ;  JC F1\_2 ;  MOV A, R0 ;  MOV B, A ;  MUL AB ;  MOV R5, A ;  MOV R6, B ;  MOV A, R3 ;  CLR C ;  SUBB A, R5 ;  JC F1\_2 ;  MOV A, R6 ;  JNZ F1\_2 ;  MOV A, R0 ;  INC A ;  MOV R0, A ;  JMP F1\_1 ;  F1\_2: ;  MOV A, R0 ;  DEC A ;  MOV R1, A ; ^  JMP INTOUT ; |  F2:  MOV A, R3 ; Performing the operation at C=1  MOV C, ACC.0 ; |  MOV 07h, C ; V  MOV C, ACC.1 ;  MOV 06h, C ;  MOV C, ACC.2 ;  MOV 05h, C ;  MOV C, ACC.3 ;  MOV 04h, C ;  MOV C, ACC.4 ;  MOV 03h, C ;  MOV C, ACC.5 ;  MOV 02h, C ;  MOV C, ACC.6 ;  MOV 01h, C ;  MOV C, ACC.7 ;  MOV 00h, C ;  MOV A, 20h ;  XRL A, R4 ;  MOV R1, A ; ^  JMP INTOUT ; |  INTOUT:  RETI ; Return from Interrupt Handling  END |

3. Тестирование программы

Разработанная программа тестировалась на примерах, представленных на рисунке 2.

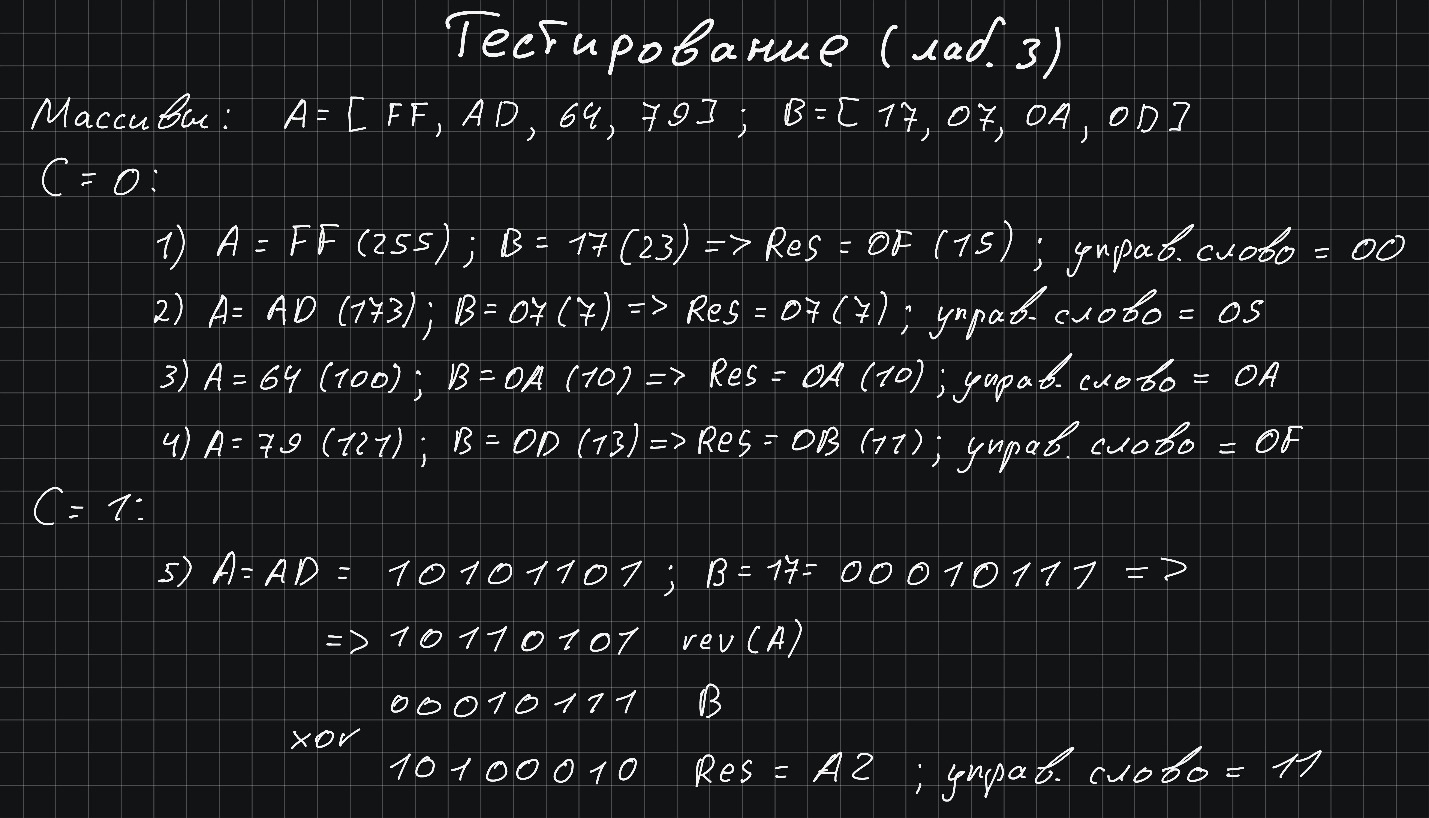


Рисунок 2. –Тестовые примеры

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

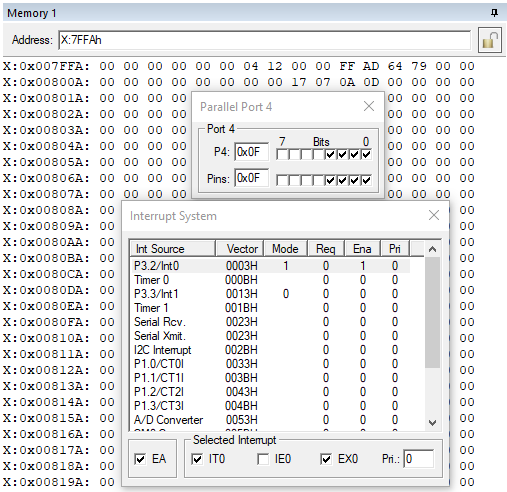


Рисунок 3.1 – Тестовый пример № 1

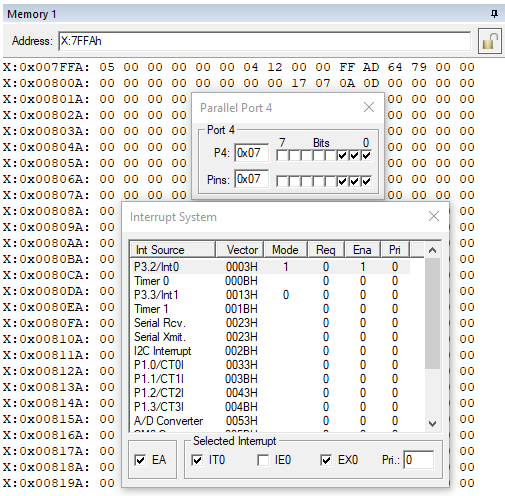


Рисунок 3.2 – Тестовый пример № 2

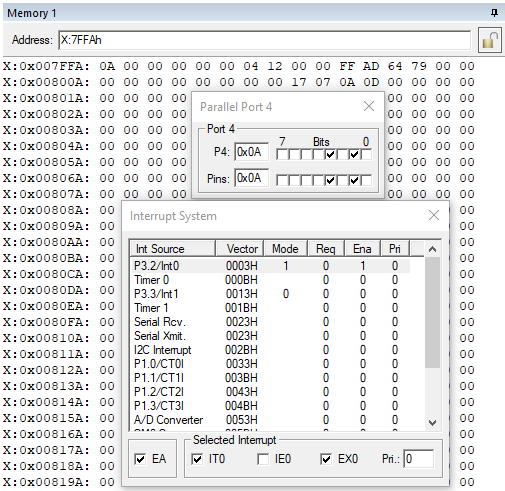


Рисунок 3.3 – Тестовый пример № 3

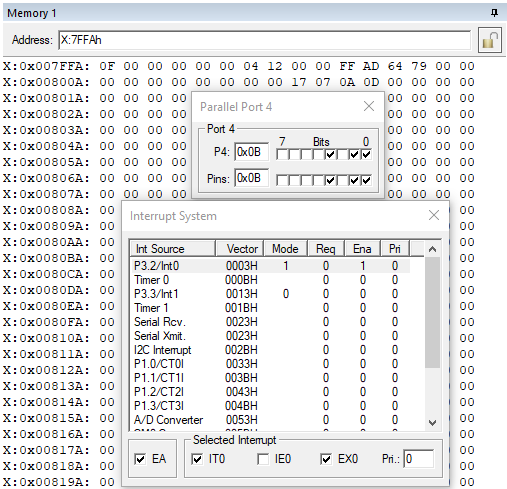


Рисунок 3.4 – Тестовый пример № 4

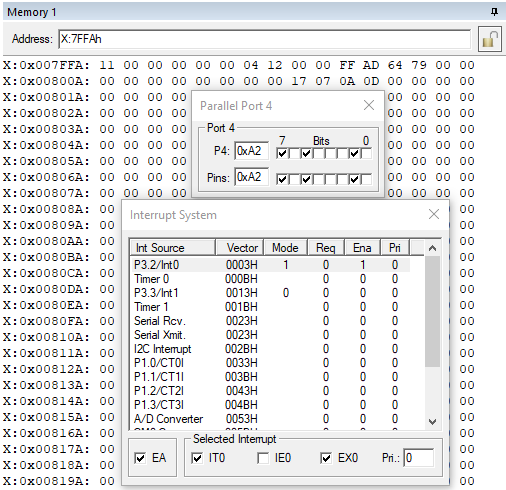


Рисунок 3.5 – Тестовый пример № 5

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>.