**Национальный исследовательский ядерный университет «МИФИ»**

|  |  |
| --- | --- |
|  | **Факультет Кибернетики и информационной безопасности**  **Кафедра «Компьютерные системы и технологии»** |

**ОТЧЕТ**

**по выполнению лабораторного практикума**

**по куру «Микропроцессорные устройства и системы»**

Студент гр. Б18-503 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Григорьев М.П. /

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Ёхин М.Н. /

**Лабораторная работа №4**

Вариант 22

Условие:

1. Сумма количества “1” в разрядах 0, 1, 5 байта “A”, количества “0” в разрядах 2, 4, 7 байта “B” и модуля разности адреса байтов “A” и “B”.

Прерывание по нажатию на “5”

1. Сдвиг влево на N разрядов, где N=(addr A) \* ( |addr B – 2|) с вводом “0”.

Прерывание по нажатию на Прерывание по “7”

Вывод результата на ССИ и ЖКИ:

* ССИ в HEX формате – старшая тетрада на 2 знакоместо
* ЖКИ вторая строка младшая тетрада в HEX формате 4 знакоместо.

**Исходный код программы:**

ORG 8000H

LJMP PROGRAM

ORG 8013H

LJMP INTER

PROGRAM:

P4 EQU 0C0h

SETB EA; allow interrupts

SETB EX1; allow INT1

MOV DPTR, #7FFFh

MOV A, #01h

MOVX @DPTR, A

MOV 40H, #0F3h; 0

MOV 41H, #60h; 1

MOV 42H, #0B5h; 2

MOV 43H, #0F4h; 3

MOV 44H, #66h; 4

MOV 45H, #0D6h; 5

MOV 46H, #0D7h; 6

MOV 47H, #70h; 7

MOV 48H, #0F7h; 8

MOV 49H, #0F6h; 9

MOV 4AH, #77h; 10 A

MOV 4BH, #0C7h; 11 B

MOV 4CH, #93h; 12 C

MOV 4DH, #0E5h; 13 D

MOV 4EH, #97h; 14 E

MOV 4FH, #17h; 15 F

LJMP OUTPUT

OUTPUT:

MOV P4, A

SJMP OUTPUT

INTER:

MOV DPTR, #7FFFh ; control command

MOV A, #40h

MOVX @DPTR, A; allow read FIFO keyboard

MOV DPTR, #7FFEh

MOVX A, @DPTR

CJNE A, #11001001B, K2 ; check 5 key if Arifmetic

MOV A, #00h

MOV R4, A ; save kop to R4

SJMP READ\_INPUT

K2:

CJNE A, #11010000B, EXIT ; check 7 key if Logic

MOV A, #01h

MOV R4, A ; save kop to R4

READ\_INPUT:

MOV DPTR, #7FFAh

MOVX A, @DPTR

MOV 20H, A; Save input

MOV DPTR, #8000h

MOV R5, #0 ; Address A

JNB ACC.3, A1

INC DPTR

INC R5

A1:

JNB ACC.4, B0

INC DPTR

INC DPTR

INC R5

INC R5

B0:

MOVX A, @DPTR

MOV R0, A; R0 - A

MOV A, 20H

MOV DPTR, #8004h

MOV R6, #0 ; Address B

JNB ACC.1, B1

INC DPTR

INC R6

B1:

JNB ACC.2, MOV\_B

INC DPTR

INC DPTR

INC R6

INC R6

MOV\_B:

MOVX A, @DPTR

MOV R1, A; R1 - B

MOV A, R4

JB ACC.0, LOG1; Analyze operation A0 == Logic

JNB ACC.0, ARIFM

LOG1:

LJMP LOGIC

EXIT:

RETI

ARIFM:

MOV A, #00h

MOV R2, A; 0 TO COUNT 1 IN A (R0)

MOV R3, A; 0 TO COUNT 0 IN B (R1)

MOV A, R0; COUNT 1 IN A

JNB ACC.0, COUNT\_1\_1

INC R2 ; COUNT A0

COUNT\_1\_1:

JNB ACC.1, COUNT\_1\_5

INC R2 ; COUNT A1

COUNT\_1\_5:

JNB ACC.5, COUNT\_0

INC R2 ; COUNT A5

COUNT\_0:

MOV A, R1; COUNT ZEROS IN B

JB ACC.2, COUNT\_0\_2

INC R3

COUNT\_0\_2:

JB ACC.4, COUNT\_0\_7

INC R3

COUNT\_0\_7:

JB ACC.7, ADDRESSES

INC R3

ADDRESSES:

CLR C

MOV A, R6 ; B Adress

SUBB A, R5; Sub A Adress | B-A

JNC ARIF\_RES

CLR C

MOV A, R5 ; A Adress

SUBB A, R6; Sub B Adress | A-B

ARIF\_RES:

ADD A, R2

ADD A, R3

LJMP OP\_END

LOGIC:

CLR C

MOV A, R6 ; addr B -> reg A

SUBB A, #2h ; Sub 2 -> addr B - 2

JNC LOGIC\_CYCLE

CLR C

MOV A, #2h ; #2 -> reg A

SUBB A, R6 ; sub B -> 2 - addr B

LOGIC\_CYCLE:

MOV B, R5 ; A -> reg B

MUL AB ; addr A \* (addr B-2)

MOV R5, A

MOV A, R0 ; A -> Acc

CYCLE:

CJNE R5, #00h, LEFT\_SHIFT

AJMP OP\_END

LEFT\_SHIFT:

CLR C

RLC A

DEC R5

AJMP CYCLE

OP\_END:

MOV R4, A

LCALL JK\_VIVOD

MOV A, R4

LCALL SSI\_VIVOD

MOV A, R4

SWAP A

LJMP EXIT

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; подпрограмма вывода на ССИ дисплей

SSI\_VIVOD:

MOV R3, A

ANL A, #0F0h

SWAP A

MOV R0, A; HIGH

MOV DPTR, #7FFFh

MOV A, #91h ; Write to video memory addr - 1

MOVX @DPTR, A

MOV A, R0

ADD A, #40H; Offset to right letter

MOV R1, A

MOV A, @R1

MOV DPTR, #7FFEh; Write

MOVX @DPTR, A

;;;;; CLEAR

MOV DPTR, #7FFFh

MOV A, #90h ; Write to video memory addr - 0

MOVX @DPTR, A

MOV A, #00h; Clear screen

MOV DPTR, #7FFEh

MOVX @DPTR, A

MOV DPTR, #7FFFh

MOV A, #92h ; Write to video memory addr – 2

MOVX @DPTR, A

MOV A, #00h; Clear screen

MOV DPTR, #7FFEh

MOVX @DPTR, A

MOV DPTR, #7FFFh

MOV A, #93h ; Write to video memory addr - 3

MOVX @DPTR, A

MOV A, #00h; Clear screen

MOV DPTR, #7FFEh

MOVX @DPTR, A

RET

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; подпрограмма вывода на ЖКИ

JK\_VIVOD:

MOV A, #04H ; shift cursor

LCALL DINIT

MOV A, #38H ; 2 strings

LCALL DINIT

MOV A, #0CH ; turn on display

LCALL DINIT

MOV A, #01H ; clear display

LCALL DINIT

MOV A, #10101011B ; write in 2B

LCALL DINIT

MOV A, R4

LCALL DISP\_JK

RET

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; подпрограмма вывода символов на дисплей

DISP\_JK:

ANL A, #0Fh

MOV R2, A; LOW

CLR C

SUBB A, #0Ah ; A-10

JNC CALC\_SYM ; if 0-9

MOV A, R2

ADD A, #30H

SJMP CALC\_RET

CALC\_SYM:

ADD A, #41H

CALC\_RET:

LCALL DISP\_SYM

RET

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; подпрограмма записи команды в управляющий регистр дисплея

DISP\_SYM:

MOV R0, A

MOV DPTR, #7FF6H; Ending of writing waiting

BF1:

MOVX A, @DPTR

ANL A, #80H

JNZ BF1

MOV DPTR, #7FF5H; Write to data register of display

MOV A, R0

MOVX @DPTR, A

RET

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; подпрограмма записи кода символа в регистр данных дисплея

DINIT:

MOV R0,A

MOV DPTR,#7FF6H ; Ending of writing waiting

BF:

MOVX A,@DPTR

ANL A,#80H

JNZ BF

MOV DPTR,#7FF4H; Write command to data register of display

MOV A,R0

MOVX @DPTR,A

RET

END

**Пример заполнения памяти:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7FFAh | … | 8000 | 8001 | 8002 | 8003 | 8004 | 8005 | 8006 | 8007 |
| 00 |  | 23 | 23 | 00 | FF | 23 | 23 | FF | 23 |

**Тестовые случаи:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Адрес A | Адрес B | A | B | Операция | Результат |
| 10 | 10 | 00000000 | 11111111 | Арифметическая | 00000000 |
| 10 | 01 | 00000000 | 00100011 | Арифметическая | 00000100 |
| 11 | 11 | 11111111 | 00100011 | Арифметическая | 00000110 |
| 00 | 11 | 00100011 | 00100011 | Арифметическая | 00001001 |
| 11 | 00 | 11111111 | 00100011 | Арифметическая | 00001001 |
| 11 | 10 | 11111111 | - | Логическая | 11111111 |
| 11 | 00 | 11111111 | - | Логическая | 11000000 |
| 11 | 11 | 11111111 | - | Логическая | 11111000 |
| 11 | 01 | 11111111 | - | Логическая | 11111000 |

Итог: тестирование программы на стенде выполнено успешно, все требования выполнены, так как результаты выполнения операций корректны.