МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ»

Кафедра ИиСП

Отчет
по лабораторной работе № 3

по дисциплине «Машинно-зависимые языки программирования»
Вариант 8

Выполнил: ст. гр. ПС-14
Смирнов И.В
Проверил: доцент
кафедры ИиСП Баев А.А.

г. Йошкар-Ола
2023

Цель работы:

протестировать команды и посмотреть как
изменяются флаги

Задания на лабораторную работу:
написать программу на ассемблере и
протестировать на изменение флагов

1. Теоретические сведения

Мнемоника OR
Операнды Rd,Rr
Описание Логическое ИЛИ
Операция Rd = Rd V Rr
Флаги Z,N,V,S
Циклы 1

Получившийся код программы:

```
reset:
        rjmp main

main:
        ;       загрузка значений в регистры
        ldi r18, 0xFF
        ldi r19, 0xFF
        ;       вывод данных из РОН в IO регистр для отображения
        out     OCR0A, r18
        out OCR0B, r19
        nop
loop:
        ;       ввод данных из IO регистров в РОН для обработки
        ;       так как арифм и логические инструкции работают с РОН
        in r18, OCR0A
        in r19, OCR0B
        ;       выполнение операции
        or r18, r19
        ;       вывод данных из РОН в IO регистр для отображения
        out     OCR0A, r18
        out OCR0B, r19

        rjmp loop
```

Тестовый файл:
$log OCR0A
$log OCR0B
$log SREG
$startlog
Asm_Lab_Math_log_output.stim
#6 OCR0A = 0
OCR0B = 0
#7
OCR0A = 28
OCR0B = 128
#7
OCR0A = 255
OCR0B = 255
#7
$stoplog
$break

Выходной файл:
#3
OCR0A = 0xff
#1
OCR0B = 0xff
#2
OCR0B = 0x00 OCR0A = 0x00
#2
SREG = 0x02
#5
OCR0B = 0x80 OCR0A = 0x1c
#2
SREG = 0x14
#1
OCR0A = 0x9c
#4
OCR0B = 0xff OCR0A = 0xff

Разбор файла:
Через три такта после запуска программы OCR0A = 0xff, затем через 1 такт OCR0B = 0xff.
Еще через 2 такта значения регистров изменены из тестового файла, далее 3 такта на запись значений в r18 и r19 и один такт на выполнение логического или с последующим переносом значения в OCR0A. 4 Такта на запись в IO регистры и переход на новую итерацию.

Флаг Z ставится, когда операнды равны 0.
Флаг N ставится, когда один из операндов равен от 128 до 255 включительно

Флаг V всегда 0

Флаг S ставится, когда установлен флаг N,

т.к N XOR V = N XOR 0 = N

Мнемоника AND
Операнды Rd,Rr
Описание Логическое И
Операция Rd = Rd & Rr
Флаги Z,N,V,S
Циклы 1

Получившийся код программы:

```
reset:  rjmp main

main:
        ;       загрузка значений в регистры
        ldi r18, 0xFF
        ldi r19, 0xFF
        ;       вывод данных из РОН в IO регистр для отображения
        out     OCR0A, r18
        out OCR0B, r19
        nop
loop:
        ;       ввод данных из IO регистров в РОН для обработки
        ;       так как арифм и логические инструкции работают с РОН
        in r18, OCR0A
        in r19, OCR0B
        ;       выполнение операции
        and r18, r19
        ;       вывод данных из РОН в IO регистр для отображения
        out     OCR0A, r18
        out OCR0B, r19

        rjmp loop
```

Тестовый файл:

```
$log OCR0A
$log OCR0B
$log SREG
$startlog Asm_Lab_Math_log_output.stim
#6
OCR0A = 0
OCR0B = 0
#7
OCR0A = 20
OCR0B = 128
#7
$stoplog
$break
```

Результат тестового файла:

```
#3
OCR0A = 0xff
#1
OCR0B = 0xff
#2
OCR0B = 0x00
OCR0A = 0x00
#2
SREG = 0x02
#5
OCR0B = 0x81
OCR0A = 0x14
#3
OCR0A = 0x00
#4
OCR0B = 0xfc
OCR0A = 0x03
#3
OCR0A = 0x00
#4
OCR0B = 0x80
#7
OCR0A = 0x80
#2
SREG = 0x14
```

Разбор файла:

Через три такта после запуска программы OCR0A = 0xff, затем через 1 такт OCR0B = 0xff.

Еще через 2 такта значения регистров изменены из тестового файла, далее 3 такта на запись значений в r18 и r19 и один такт на выполнение логического и с последующим переносом значения в OCR0A. 4 Такта на запись в IO регистры и переход на новую итерацию.

Флаг Z ставится, когда либо один из операндов равен 0, либо сумма операндов равна 255
Флаг N, когда у обоих операндов старший бит равен 1
Флаг V = 0
Флаг S, когда N = 1, т.к V = 0, S = N XOR V, N XOR 0 = N

SBIW

Мнемоника SBIW
Операнды Rdl,K6
Описание Вычесть константу из слова
Операция Rdh:Rdl = Rdh:Rdl -K 6
Флаги Z,C,N,V,S
Циклы 2

Получившийся код:

```
reset:
	rjmp main

main:
	;	загрузка значений в регистры
	ldi r25, 0xFF
	ldi r24, 0xFF
	;	вывод данных из РОН в IO регистр для отображения
	out	OCR0A, r25
	out OCR0B, r24

	nop
loop:
	;	ввод данных из IO регистров в РОН для обработки
	;	так как арифм и логические инструкции работают с РОН
	in r25, OCR0A
	in r24, OCR0B
	;	выполнение операции
	sbiw r24, 5
	;	вывод данных из РОН в IO регистр для отображения
	out	OCR0A, r25
	out OCR0B, r24

	rjmp loop
```

Тестовый файл:

```
$log OCR0A
$log OCR0B
$log SREG
$startlog Asm_Lab_Math_log_output.stim
#6
OCR0A = 0
OCR0B = 63
#8
OCR0A = 0
OCR0B = 62
#8
OCR0A = 128
OCR0B = 122
#8
OCR0A = 128
OCR0B = 0
#8
$stoplog
$break
```

Выходной файл:

```
#3
OCR0A = 0xff
#1
OCR0B = 0xff
#2
OCR0B = 0x3f
OCR0A = 0x00
#5
OCR0B = 0x3a
#3
OCR0B = 0x3e
#5
OCR0B = 0x39
#3
OCR0B = 0x7a
OCR0A = 0x80
#3
SREG = 0x14
#2
OCR0B = 0x75
#3
OCR0B = 0x00
#2
SREG = 0x15
#1
SREG = 0x18
#1
OCR0A = 0x7f
#1
OCR0B = 0xfb
```

Флаг Z ставится, когда старший байт равен 0, а младший константе
Флаг C, когда старший байт(в моем случае r25) = 0, а младший байт(r24) меньше константы
Флаг N, когда младший байт больше константы, а старший байт от 128 до 255 включительно;
или когда младший байт меньше константы, а старший байт равен 0 или от 129 до 255 включительно
Флаг V, когда старший байт равен 128, а младший меньше константы
Флаг S, когда либо установлен флаг N, либо V

FMULS

Мнемоника FMULS
Операнды Rd,Rr
Описание Умножение дробных чисел со знаком
Операция R1:R0 = (Rd *Rr) <<1
Флаги Z,C
Циклы 2

Получившийся код:

```
reset:
        rjmp main


main:
 ; загрузка значений в регистры
        ldi r21, 0xFF
        ldi r20, 0xFF
 ; вывод данных из РОН в IO регистр для отображения
        out    OCR0A, r21
        out OCR0B, r20


        nop
loop:
 ; ввод данных из IO регистров в РОН для обработки
 ; так как арифм и логические инструкции работают с РОН
        in r21, OCR0A
        in r20, OCR0B
 ; выполнение операции
        fmuls r21, r20
 ; вывод данных из РОН в IO регистр для отображения
        out    OCR0A, r21
        out OCR0B, r20

        rjmp loop
```

Тестовый файл:

```
$log OCR0A
$log OCR0B
$log SREG
$startlog Asm_Lab_Math_log_output.stim
#6
OCR0A = 116
OCR0B = 0
#8
OCR0A = 0
OCR0B = 0
#8
OCR0A = 5
OCR0B = 128
#8
OCR0A = 255
OCR0B = 127
#8
$stoplog
$break
```

Выходной файл:

```
#3
OCR0A = 0xff
#1
OCR0B = 0xff
#2
OCR0B = 0x00
OCR0A = 0x74
#3
SREG = 0x02
#5
OCR0A = 0x00
#8
OCR0B = 0x80
OCR0A = 0x05
#3
SREG = 0x01
#5
OCR0B = 0x7f
OCR0A = 0xff
```

Флаг Z, когда хотя бы один из операндов равен 0
Флаг C, когда только один из операндов равен 128 и больше