

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ

Кафедра Радиотехнических устройств

Жаворонков С.С.

**Программирование на языке Ассемблер
микроконтроллера AVR ATmega128**

Материалы к практическим занятиям по дисциплине
«Микропроцессорные устройства в РЭО»

Москва 2008

1) Сложить три числа 60, 7A, 14 без учета единицы переноса. Результат сравнить с содержимым ячейки 0083, если сумма не равна содержимому, то переслать в регистр R25 результат.

```
LDI R1, $60
LDI R2, $7A
LDI R3, $14
ADD R1, R2
ADD R1, R3
LDS R4, $0083
CPSE R1, R4
MOV R25, R1
LOOP:
RJMP LOOP.
```

2) В ячейках ОЗУ 0100 и 0101 записаны два числа. Записать число второй ячейки на место первой, а первой на место второй используя

- а) прямую адресацию
- б) косвенную адресацию

а)

```
LDS R1, $0100
LDS R2, $0101
STS $0100, R2
STS $0101, R1
LOOP:
RJMP LOOP.
```

б)

```
загрузка адреса в X
LDI R26, 00
LDI R27, 01
```

```
загрузка адреса в Y
LDI R28, 01
LDI R29, 01
```

```
LD R1, X
LD R2, Y
```

```
ST X, R2
ST Y, R1
LOOP:
RJMP LOOP.
```

3) Прибавить к числу \$5A содержимое ячейки памяти 0051. Результат записать в ячейку 0049, уменьшив его на единицу. Использовать косвенную адресацию при работе с ячейками ОЗУ.

```
LDI R16, $5A
Загрузка адреса 0051 в X
LDI R26, 51
LDI R27, 00
LD R17, X
AND R16, R17
```

```
Загрузка адреса 0049 в X
LDI R26, 49
LDI R27, 00
DEC R16
ST X, R16
LOOP:
RJMP LOOP.
```

4) Сложить два числа 0054 и 0127 без учета единицы переноса. Результат записать в стек с началом в ячейке 2030. После выполнения подпрограммы извлечь содержимое из стека в регистр R1.

```
ldi R20, $30      ; загрузка промежуточного регистра R20 младшим
                  ; байтом адреса начала стека
out SPL, R20      ; загрузка младшего байта указателя стека
ldi R20, $20      ; загрузка промежуточного регистра R20 старшим
                  ; байтом адреса начала стека
out SPH, R20      ; загрузка старшего байта указатель стека
```

```
LDI R16 $0054
LDI R17 $0127
AND R16, R17
call Rout
ROUT:
PUSH R16
.....
```

```
POP R1
RET
LOOP:
RJMP LOOP.
```

5) Написать подпрограмму вычитания (с заемом) двух чисел из ячеек 0145 и 0099. Результат сравнить с содержимым ячейки 0070. В случае равенства инкрементировать результат вычитания на единицу. Результат сохранить в ячейке, адрес которой содержится в регистрах R19, R20 использовать косвенную адресацию. Обеспечить сохранение содержания указательных регистров данными, записанными до вызова подпрограммы.

```
ROUT:  
PUSH X  
PUSH Y  
PUSH Z
```

```
LDI R26, 45  
LDI R27, 01  
LDI R28, 99  
LDI R29, 00  
LD R1, X  
LD R2, Y  
SBN R1, R2
```

```
LDI R30, 70  
LDI R31, 00  
LD R2, Z  
CPSE R1, R2  
RJMP CAT  
INC R1
```

```
CAT:  
MOV R31, R20  
MOV R30, R19  
ST Z, R1  
POP Z  
POP Y  
POP X  
RET
```

б) Сумму трех чисел из ячеек ОЗУ 0106, 0110 и 0115 поместить в стек с начальным адресом 2001 В подпрограмме прибавить к содержимому регистра суммы число 80, результат поместить в ячейку 0070. При выходе из подпрограммы не сохранять результат прибавления числа 80. При пересылках из ОЗУ использовать косвенную адресацию.

```
;***** Инициализация указателя стека *****
    ldi R20, $01      ; загрузка промежуточного регистра R20 младшим
                      ; байтом адреса начала стека
    out SPL, R20     ; загрузка младшего байта указателя стека
    ldi R20, $20     ; загрузка промежуточного регистра R20 старшим
                      ; байтом адреса начала стека
    out SPH, R20     ; загрузка старшего байта указатель стека
```

загрузка адреса в X

```
LDI R26, 06
LDI R27, 01
```

загрузка адреса в Y

```
LDI R28, 10
LDI R29, 01
```

загрузка адреса в Z

```
LDI R28, 15
LDI R29, 01
```

```
LD R1, X
LD R2, Y
LD R3, Z
```

```
AND R1, R2
AND R1, R3
```

call Rout

```
ROUT:
PUSH R1
LDI R4, 80
AND R1, R4
STS $0070, R1
POP R1
RET
LOOP:
RJMP LOOP.
```

7) Сложить три числа 16, 72, 35 (без учета единица переноса). Результат записать в ячейку ОЗУ 0075, используя косвенную адресацию. Записать в регистр R16 число 15.

Осуществить вызов подпрограммы, в которой требуется:

– сохранить указатели X, Z и содержимое регистра R16 а стеке. Начальный адрес стека 10FF;

– записать в регистр R16 содержимое ячейки ОЗУ 0090 и сравнить с содержимым ячейки ОЗУ 0078, в случае неравенства увеличить содержимое регистра R16 на единицу;

– извлечь сохраненные данные из стека, поменяв содержимое указательных регистров X и Y местами.

; Инициализация указателя стека

LDI R20, \$FF ; загрузка регистра R20 младшим байтом адреса начала стека

OUT SPL, R20 ; загрузка младшего байта указателя стека

LDI R20, \$10 ; загрузка регистра R20 старшим байтом адреса начала стека

OUT SPH, R20 ; загрузка младшего байта указателя стека

LDI R17, \$16 ; загрузка регистра R17 числом 16

LDI R18, \$72 ; загрузка регистра R18 числом 72

LDI R19, \$35 ; загрузка регистра R19 числом 35

ADD R17, R18 ; суммирование содержимого регистров R17 и R18

ADD R17, R19 ; суммирование содержимого регистров R17 и R19

; загрузка в регистровую пару Z адреса ячейки ОЗУ 0075

LDI R30, \$75

LDI R31, \$00

ST Z, R17; загрузка ячейки ОЗУ с адресом 0075 значением из регистра R17

LDI R16, \$15 ; загрузка регистра R16 числом 15

CALL ROUT ; вызов подпрограммы ROUT

LOOP:

RJMP LOOP

ROUT:

; сохранение указателя X в стеке

PUSH R26

PUSH R27

; сохранение указателя Y в стеке

PUSH R30

PUSH R31

LDS R16, \$0090 ; загрузка регистра R16 содержимым ячейки ОЗУ с адресом 0090

LDS R17, \$0078 ; загрузка регистра R17 содержимым ячейки ОЗУ с адресом 0078

CPSE R16, R17 ; пропустить следующую команду если значения регистров R16 и R17 равны

INC R16 ; увеличить содержимое регистра R16 на единицу

POP R27

POP R26

; извлечение сохраненного содержимого X в Y

POP R31

POP R30

RET ; возврат из подпрограммы.