*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана***  ***(национальный исследовательский университет)»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

**Отчет**

**по лабораторной работе №3**

**Вариант 5**

**Дисциплина:** Микропроцессорные системы

**Название лабораторной работы:**

Арифметическая обработка данных

Студент гр. ИУ6-62Б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.С. Бурлаков**

(Подпись, дата) (И.О. Фамилия)

Студент гр. ИУ6-62Б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.А. Мотичев**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.К. Халайджи**

(Подпись, дата) (И.О. Фамилия)

Москва, 2019

# ВВедение

## Цель работы:

- изучение способов представления числовых данных в микроконтроллерах,

- изучение двоичных арифметических операций,

- программирование арифметических процедур.

# основная часть

**Задание 1.** Изучить программу для исследования арифметических операций в стартовом наборе

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

;Программа тестирования в STK500 двоичных арифметических операций

; сложения, вычитания, умножения, деления

;Порт PD - порт управления для выбора операндов и операций

;Порт PB - порт индикации исходных операндов и результатов операции

;Соединения шлейфами: порт PB-LED, порт PD-SW

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

.include "m8515def.inc" ;файл определений для ATmega8515

;назначение входов порта PD

.equ SW\_op\_AL = 0 ;кнопка выбора операнда op\_AL

.equ SW\_op\_AH = 1 ;кнопка выбора операнда op\_AH

.equ SW\_op\_BL = 2 ;кнопка выбора операнда op\_ВL

.equ SW\_ADD = 3 ;кнопка операции сложения res=op\_AL+op\_ВL

.equ SW\_SUB = 4 ;кнопка операции вычитания res=op\_AL-op\_ВL

.equ SW\_MUL = 5 ;кнопка операции умножения shov.res=op\_AL x op\_ВL

.equ SW\_DIV = 6 ;кнопка операции деления res=op\_AH.op\_AL/op\_ВL

.equ SW\_SHOW = 7 ;кнопка для просмотра признаков сложения-вычитания,

;старшего байта произведения или остатка при делении

.def op\_AL = r16 ;1-й операнд АL

.def op\_AH = r17 ;старший байт делимого AH

.def op\_BL = r18 ;2-й операнд ВL

.def res = r1 ;результат операции (сумма, разность,

; младший байт произведения или частное)

.def show = r31 ;регистр признаков сложения-вычитания,

; старшего байта произведения или остатка при делении

.def mul\_l = r21 ;младший байт произведения

.def mul\_h = r22 ;старший байт произведения

.def copy\_AH = r23 ;копия старшего байта делимого

.def copy\_AL = r24 ;копия младшего байта делимого

.def copy\_BL = r25 ;копия множителя

.def temp = r26 ;временный регистр

.def sw\_reg = r27 ;регистр состояния кнопок

.def count = r28 ;число операндов в таблице операндов

.def c\_bit = r29 ;счетчик циклов умножения (деления)

.def temp = r30

.macro vvod ;ввод операнда

lpm ;считывание байта из flash-памяти в r0

mov @0,r0 ; и пересылка в регистр операнда

mov res, r0

adiw zl, 1 ;увеличение указателя адреса на 1

dec count

brne exit

ldi ZL,low(tabl\_op\*2) ;перезагрузка начала таблицы операндов

ldi ZH,high(tabl\_op\*2) ; в регистр Z

ldi count, 10 ;число заданных операндов в таблице 10

exit: nop

.endmacro

.org $000 ;Инициализация стека, портов, адреcного регистра Z

ldi temp,low(RAMEND) ;установка

out SPL,temp ; указателя стека

ldi temp,high(RAMEND) ; на последнюю

out SPH,temp ; ячейку ОЗУ

ser temp ;настройка

out DDRB,temp ; порта PB

out PORTB,temp ; на вывод

clr temp ;настройка

out DDRD,temp ; порта PD

ser temp ; на

out PORTD,temp ; ввод

ldi ZL,low(tabl\_op\*2) ;загрузка адреса таблицы операндов

ldi ZH,high(tabl\_op\*2) ; в регистр Z

ldi count,10 ;число операндов 10

;Опрос кнопок и идентификация нажатой

LOOP: in sw\_reg,PIND

sbrs sw\_reg,0

rjmp f\_op\_AL

sbrs sw\_reg,1

rjmp f\_op\_AH

sbrs sw\_reg,2

rjmp f\_op\_BL

sbrs sw\_reg,3

rjmp add\_bin

sbrs sw\_reg,4

rjmp sub\_bin

sbrs sw\_reg,5

rjmp mul\_bin

sbrs sw\_reg,6

rjmp div\_bin

sbrc sw\_reg,7

rjmp loop

mov res,show

rjmp outled

;Выборка 1-го операнда из таблицы операндов

f\_op\_AL: vvod op\_AL

rjmp outled ;Выборка старшего байта 1-го операнда (при делении)

f\_op\_AH: vvod op\_AH

rjmp outled ;Выборка 2-го операнда

f\_op\_BL: vvod op\_BL

rjmp outled ;Сложение 8-разрядных операндов

add\_bin: mov res,op\_AL

add res,op\_BL

in show,SREG ;выборка из регистра SREG

rjmp outled ;Вычитание 8-разрядных операндов

sub\_bin: mov res,op\_AL

sub res,op\_BL

in show,SREG ;выборка из регистра SREG

rjmp outled ;Умножение 8-разрядных операндов

mul\_bin: clr mul\_l ;очистка младшего

clr mul\_h ; и старшего байта произведения

ldi c\_bit,8 ;счетчик циклов

mov copy\_BL,op\_BL

L1: clc ;очистка флага C

sbrc copy\_BL,0 ;проверка младшего бита множителя

add mul\_h,op\_AL ;прибавление множимого AL

L2: ror mul\_h ;сдвиг вправо

ror mul\_l ; 2-х байтов произведения

lsr copy\_BL ;сдвиг множителя вправо

L3: dec c\_bit ;уменьшение счётчика циклов

brne L1 ;если не 0, продолжаем умножение

mov res,mul\_l ;выводимые значения - младший

mov show,mul\_h ; и старший байты произведения

rjmp outled ;Деление 16-разрядного числа на 8-разрядное

div\_bin: sbrc op\_AH,7 ;ошибки исходных данных

rjmp error

sbrc op\_BL,7

rjmp error

tst op\_BL ;ошибка при делении на 0

breq error

cp op\_AH,op\_BL ;ошибка при переполнении

brge error clr res ;обнуляем частное

ldi c\_bit,8 ; число итераций

mov copy\_AH,op\_AH

mov copy\_AL,op\_AL

L4: clc

rol copy\_AL ;сдвиг

rol copy\_AH ; делимого

lsl res ;сдвиг частного влево

sub copy\_AH,op\_BL ;вычитание делителя

brcs recov ;если остаток < 0,переход

inc res ; иначе добавить 1 в частное

rjmp L5

recov: add copy\_AH,op\_BL ;восстановление остатка

L5: dec c\_bit

brne L4

mov show,copy\_AH ;пересылка остатка

rjmp outled

error: clr show ;сигнал об ошибке деления

out PORTB,show

rcall delay

ser show

out PORTB,show

rjmp wait

outled: com res

out portb,res

rcall delay

wait: in sw\_reg,PIND ;ждать, пока кнопка не отпущена

com sw\_reg

brne wait

rjmp loop

; Задержка

DELAY: ldi r19,10

m1: ldi r20,250

m3: ldi r21,250

m2: dec r21

brne m2

dec r20

brne m3

dec r19

brne m1

ret

; Таблица операндов в шестнадцатеричном представлении

tabl\_op: .db 0xD8,0xA5,0xB3,0x6E,0x4C,0x3C,0x9F,0x7D,0xB0,0xB8

**Задание 2.** Сложение/вычитание двоичных чисел

tabl\_op: .db 0xD8,0xA5,0xB3,0x6E,0x4C,0x3C,0x9F,0x7D,0xB0,0xB8

|  |  |  |  |
| --- | --- | --- | --- |
| **Число А2/A10** | **Число B2/B10** | **A+B**  **A-B** | **Признаки:**  **HSVNZC** |
| 11011000 (216) | 10100101(165) | 01111101 ( 125)  00110011 ( 51) | 0---01  0---00 |
| 10110011 (-77) | 01101110 (110) | 00100001 ( 33)  01000101 ( 69) | 100001  111000 |
| 01001100 ( 76) | 00111100 ( 60) | 10001000 (-120)  00010000 ( 16) | 101100  000000 |
| 10011111 (-97) | 01111101(125) | 00011100 ( 28)  00100010 ( 34) | 100001  011000 |
| 10110000 (-80) | 10111000 (-72) | 01101000 ( 104)  11111000 ( -8) | 011001  110101 |

**Задание 3**. Умножение целых чисел

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C** | **B** | **B=1?** | **Next step** | **n** |
| 0000.0000  1101.1000  0110.1100  0011.0110  1.0000.1110  1000.0111  0100.0011  0010.0001  1111.1001  0111.1100  0011.1110  1.0001.0110  1000.1011 | 1010.0101  1010.0101  0101.0010  0010.1001  0010.1001  0001.0100  1000.1010  1100.0101  1100.0101  1110.0010  0111.0001  0111.0001  0011.1000 | true  false  true  false  false  true  false  true | C+=A  C, B >>  C, B >>  C+=A  C, B >>  C, B >>  C, B >>  C+=A  C, B >>  C, B >>  C+=A  C, B >> | 8  7  6  5  4  3  2  1 |

A = 1101.10002 = 216

216 \* 165 = 35640 = 1000.1011.0011.10002

**Задание 4**. Деление целых чисел

A = 13221 = 33A5h

B = 72 = 48h = 0100.1000 [B] = 1011.1000

0011.0011.1010.0101 Делимое A(AH.AL)

1011.1000 пробное вычитание B

1110.1011 Переполнения нет

0110.0111.0100.101х Сдвиг восстановленного AH

1011.1000 Вычитание B

0001.1111 Остаток > 0 с7 = 1

0011.1110.1001.01xх Сдвиг остатка

1011.1000 Вычитание B

1111.0110 Остаток < 0 с6 = 0

0111.1101.0010.1xxх Сдвиг восстановленного AH

1011.1000 Вычитание B

0011.0101 Остаток > 0 с5 = 1

0110.1010.0101.xxxх Сдвиг остатка

1011.1000 Вычитание B

0010.0010 Остаток > 0 с4 = 1

0100.0100.101x.xxxх Сдвиг остатка

1011.1000 Вычитание B

1111.1100 Остаток < 0 с3 = 0

1000.1001.01xx.xxxх Сдвиг восстановленного AH

1011.1000 Вычитание B

0100.0001 Остаток > 0 с2 = 1

1000.0010.1xxx.xxxх Сдвиг остатка

1011.1000 Вычитание B

0011.1011 Остаток > 0 с1 = 1

0111.0101.xxxx.xxxх Сдвиг остатка

1011.1000 Вычитание B

0010.1101 Остаток > 0 с0 = 1

Частное: 10110111 (183); Остаток: 0010.1101 (45)

# Заключение

1. Изучены способы представления числовых данных в микроконтроллерах,
2. Изучены двоичные арифметических операции,
3. Описаны программы арифметических процедур.