



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«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»

---

НАПРАВЛЕНИЕ ПОДГОТОВКИ «09.03.01 Информатика и вычислительная техника»

## ОТЧЕТ по лабораторной работе № 3

**Название:** «Арифметическая обработка данных.»

**Дисциплина:** «Микропроцессорные системы.»

**Вариант № 5**

Студент ИУ6-62Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

А.Е.Медведев  
(И. О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Б.И.Бычков  
(И. О. Фамилия)

2022 г.

## 1 Цель работы:

- изучение способов представления числовых данных в микроконтроллерах,
- изучение двоичных арифметических операций,
- программирование арифметических процедур.

## 2 Задание 1.

Изучить программу для исследования арифметических операций в стартовом наборе STK500, приведенную ниже. Программой предусмотрен ввод кода операции, операндов, выполнение заданной операции и показ результатов. В стартовом наборе STK500 всего восемь кнопок общего назначения (SW7...SW0). При тестировании арифметических операций эти кнопки используются следующим образом: кнопки SW0...SW2 – для ввода младшего (AL) и старшего байта (AH) первого операнда и одного байта второго операнда (BL), SW3...SW6 – для выполнения операций сложения, вычитания, умножения и деления, SW7 – для просмотра. Ввести в таблицу операндов в конце программы вместо приведенных в тексте программы 10 операндов из таблицы вариантов (Таблица – 2.1).

Таблица 2.1 – Таблица вариантов программируемых операндов в шестнадцатеричном коде

№ вар.	Беззнаковые		Числовые операнды со знаком в дополнительном коде							
	$AL_0$	$BL_0$	$AL_1$	$BL_1$	$AL_2$	$BL_2$	$AL_3$	$BL_3$	$AL_4$	$BL_4$
5	0xD8	0xA5	0xB3	0x6E	0x4C	0x3C	0x9F	0x7D	0xB0	0xB8

После загрузки программы в микроконтроллер проверить работу программы на плате, перебирая операнды таблицы с помощью кнопок SW0, SW2 и наблюдая выбираемые операнды в двоичном коде на светодиодной линейке.

Схема алгоритма работы программы представлена на рисунках 3.1 — 3.3.



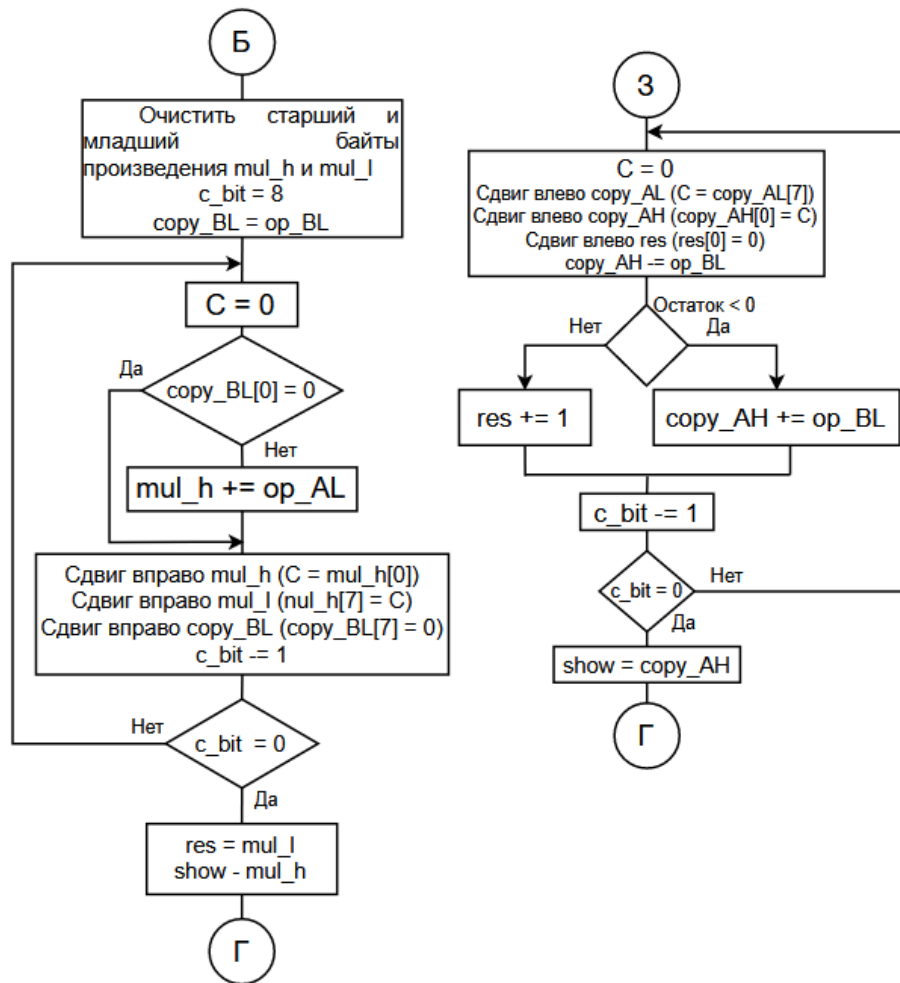


Рисунок 3.3 – Первая схема алгоритма

## 4 Код программы:

Листинг 4.1 – Код программы

```

;*****
;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_BL
.equ SW_ADD = 3            ;кнопка операции сложения res=op_AL+op_BL

```

```

.equ SW_SUB = 4      ;кнопка операции вычитания res=op_AL-op_BL
.equ SW_MUL = 5      ;кнопка операции умножения shov.res=op_AL x
    op_BL
.equ SW_DIV = 6      ;кнопка операции деления res=op_AH.op_AL/op_BL
.equ SW_SHOW = 7     ;кнопка для просмотра признаков
    сложения-вычитания,
                        ;старшего байта произведения или остатка при
                        делении
.def op_AL = r16      ;1-й операнд AL
.def op_AH = r17      ;старший байт делимого AH
.def op_BL = r18      ;2-й операнд BL
.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)

.macro vvod           ;ввод операнда
    lpm              ;считывание байта из flash-памяти в r0
    mov @0,r0        ; и пересылка в регистр операнда
    mov res, r0
    adiw z1, 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, адресного 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 выполнение заданных действий
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:    mul op_AL,op_BL
mov show,r1      ;копируем старший и
            mov res,r0        ; младший байт 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
            cpr 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 resov         ;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 temp            ;сигнал об ошибке деления
    out PORTB, temp
    rcall delay
    ser temp
    out PORTB, temp
    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

```



## 5 Задание 2.

Выполнить ряд примеров на сложение и вычитание, выбирая операнды слагаемых AL и BL нажатием кнопок SW0 и SW2. Объяснить результаты операций при нажатиях кнопки SW3 (сложение) и SW4 (вычитание), рассматривая операнды как беззнаковые числа, затем как числа со знаком. В последнем случае загружаемые из таблицы операндов программы отрицательные числа, содержащие единицу в старшем разряде, следует рассматривать в дополнительном коде. Нажатие кнопки SW7 показывает признаки результата операции, формируемые в регистре SREG (табл.2): C – перенос при сложении (заем при вычитании), Z – признак нулевого результата, N – знак результата при операциях с числами со знаком, V – переполнение разрядной сетки,  $S = N \oplus V$  – знак результата вне зависимости от переполнения, H – межтетрадный перенос (заем).

Таблица 5.1 – Байт признаков результата

№ разряда	7	6	5	4	3	2	1	0
Флаг	–	–	H	S	V	N	Z	C

Результаты наблюдений (исходные операнды, результаты операций и признаки) привести в табл. 3 в двоичном ( $A_2, B_2$ ) и десятичном ( $_{10}, _{10}$ ) виде. При оценке результатов соблюдать типы обрабатываемых переменных (беззнаковые целые или целые со знаком).

## 6 Выполнение:

Для вычислений используем беззнаковые числа и первые четыре знаковых числа от AL1 до BL2. Результаты выполнения операций сложения и вычитания, а также значения признаков в регистре SREG приведены в таблице 6.1.

Таблица 6.1 – Результаты операций сложения и вычитания

Число $A_2/A_{10}$	Число $B_2/B_{10}$	$A + B / A - B$	Признаки: H S V N Z C
1101.1000/+216 Беззнаковое	1010.0101/+165 Беззнаковое	0111.1101 (+125) / 0011.0011 (+51)	0 1 1 0 0 1 0 - - - 0 0
1011.0011/-77 Со знаком	0110.1110/+110 Со знаком	0010.0001 (+33) / 0100.0101 (+37)	1 0 0 0 0 1 1 1 1 0 0 0
0100.1100/+76 Со знаком	0011.1100/+60 Со знаком	1000.1000 (-120) / 0100.0101 (+16)	1 0 1 1 0 0 0 0 0 0 0 0
1001.1111/-98 Со знаком	0111.1101/+125 Со знаком	0001.1100 (+28) / 0010.0010 (+34)	1 0 0 0 0 1 0 1 1 0 0 0
1011.0000/-88 Со знаком	1011.1000/-72 Со знаком	0110.1000 (+104) / 1111.1000 (-8)	0 1 1 0 0 1 1 1 0 1 0 1

## 7 Умножение и деление целых чисел

### Задание 3.

Выполнить ряд примеров умножения 8-разрядных двоичных чисел с разными операциями умножения, выбирая соответствующие им операнды из таблицы операндов и проверяя результаты с помощью калькулятора. Нажатие кнопки SW5 показывает младший байт произведения, SW7 – старший байт.

**Выполнение:** На рисунке 7.1 изображена схема алгоритма операции умножения.



Рисунок 7.1 – Схема умножение двоичных чисел

Таблица 7.1 – Входные данные для умножения

Таблица 7.2 – Произведение двух чисел равно:

Таблица 7.3 – Результаты операции умножения

1. При перемножении двух знаковых чисел операция mul была заменена на muls
2. При перемножении знакового и беззнакового числа операция mul была заменена на mulsu.

## 8 Задание 4.

Выполнить деление беззнаковых чисел, 16-разрядного делимого на 8-разрядный делитель, с восстановлением остатка при условиях, что делитель не равен 0 и его значение не вызовет переполнения, а также делимое и делитель заданы с нулевыми значениями старших разрядов. Если деление невозможно, выводится предупреждение путем зажигания и гашения всех светодиодов. Нажатие кнопки SW6 показывает частное, SW7 – остаток.

Выполнить 2-3 примера на деление двоичных чисел, самостоятельно подобрав делимое и делитель. Подобрать пример с максимальными значениями делимого АН.AL и делителя В, при которых частное С будет равно 0b11111111, изменив в случае необходимости программную таблицу операндов.

Запротоколировать деление двух операндов по шагам по образцу примера из описания алгоритма, указывая промежуточные значения в регистрах делимого (остатка) АН.AL и частного С.

### Выполнение:

На рисунке 8.1 изображена схема алгоритма операции деления. В таблице 8.1 приведены результаты выполнения операций деления.

Таблица 8.1 – Результаты операции умножения

$AH_2AL_2/A_{10}$	$BL_2/B_{10}$	Частное	Остаток
0000.0100 0110.1110/1134	0100.1100/76	00001100/14	01000110/70
0111.1010 0111.1010/7A7A	1010.1010/AA	10111000/184	01001010/74
0111.1110 1111.1111/7EFF	0111.1111/7F	11111111/255	01111111/126

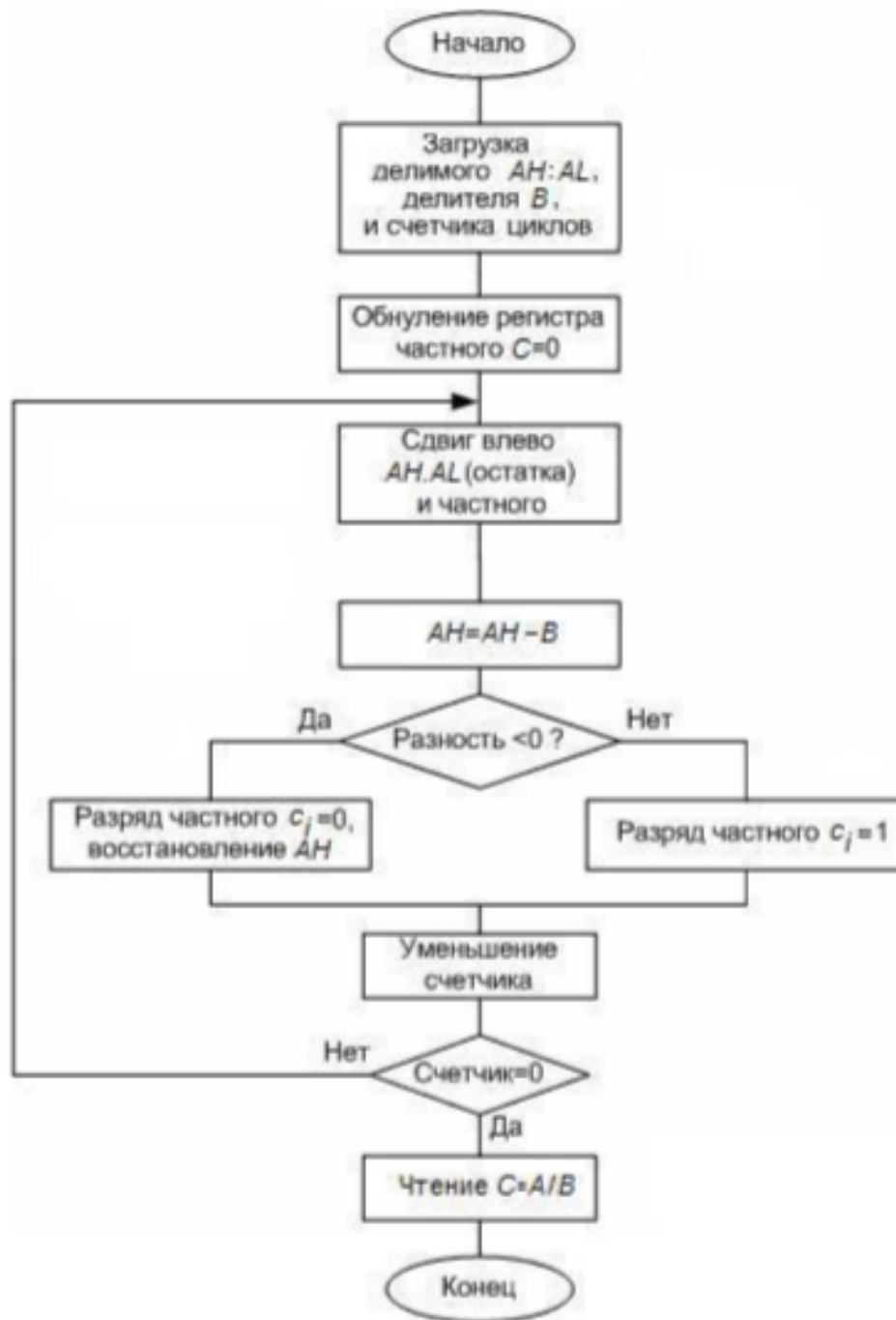


Рисунок 8.1 – Схема алгоритма деления

Далее пошагово расписано деление чисел из первой строчки таблицы 8.1:

Таблица 8.2 – Результаты операции умножения

0 0 0 0 0 1 0 0	0 1 1 0 1 1 1 0	>0 Переполнения нет!
1 0 1 1 0 1 0 0		
1 0 1 1 1 0 0 0		
0 0 0 0 1 0 0 0	1 1 0 1 1 1 0 x	<0 => $c_7 = 0$ Восстановление!
1 0 1 1 0 1 0 0		
1 0 1 1 1 1 0 0		
0 0 0 1 0 0 0 1	1 0 1 1 1 0 x x	<0 => $c_6 = 0$ Восстановление!
1 0 1 1 0 1 0 0		
1 1 0 0 0 1 0 1		
0 0 1 0 0 0 1 1	0 1 1 1 0 x x x	<0 => $c_5 = 0$ Восстановление!
1 0 1 1 0 1 0 0		
1 1 0 1 0 1 1 1		
0 1 0 0 0 1 1 0	1 1 1 0 x x x x	<0 => $c_4 = 0$ Восстановление!
1 0 1 1 0 1 0 0		
1 1 1 1 1 0 1 0		
1 0 0 0 1 1 0 1	1 1 0 x x x x x	>0 => $c_3 = 1$ Сдвиг!
1 0 1 1 0 1 0 0		
0 1 0 0 0 0 0 1		
1 0 0 0 0 0 1 1	1 0 x x x x x x	>0 => $c_2 = 1$ Сдвиг!
1 0 1 1 0 1 0 0		
0 0 1 1 0 1 1 1		
0 1 1 0 1 1 1 1	0 x x x x x x x	>0 => $c_1 = 1$ Сдвиг!
1 0 1 1 0 1 0 0		
0 0 1 0 0 0 1 1		
0 1 0 0 0 1 1 0	x x x x x x x x	<0 => $c_0 = 0$
1 0 1 1 0 1 0 0		
1 1 1 1 1 0 1 0		
1 1 1 1 1 0 1 0		Остаток
0 1 0 0 1 1 0 0		
0 1 0 0 0 1 1 0		

После деления  $A$  на  $B$  получаем число  $C = 00001110$  в двоичной системе. Переведя его в десятичную получим, что при делении 1134 на 76 получаем 14 целых и 70 в остатке.

В третьей строке таблицы 8.1 подобран пример с максимальными значениями делимого  $AN.AL$  и делителя  $B$ , при которых частное  $C$  будет равно  $11111111 = 255$ . В соответствии с условием задачи, делитель не может начинаться с 0, тогда наибольшим делителем будет число  $01111111 = 127$ , а его остатком –  $01111110 = 126$ . Тогда наибольшим значением делимого будет число  $127 * 255 + 126 = 32511 = 0111111011111111$ , что удовлетворяет условию задания.

## 9 Вывод

В ходе данной лабораторной работы были изучены способы представления целых чисел, дробных чисел, чисел со знаком и без знака в микроконтроллерах: целые числа имеют формат с фиксированной точкой справа, дробные числа меньше единицы имеют формат с точкой слева. При целочисленном представлении старший разряд используется для представления знака: 0 – положительный, а 1 – отрицательный. Изучены предназначения битов регистра SREG:  $C$  – перенос при сложении,  $Z$  – признак нулевого результата,  $N$  – знак результата при операциях с числами со знаком,  $V$  – переполнение разрядной сетки,  $S = N \oplus V$  – знак результата вне зависимости от переполнения,  $H$  – межтетрадный перенос. Также были изучены двоичные арифметические операции и описаны программы арифметических процедур.