



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 (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 Прикладная информатика

**О Т Ч Е Т**

по лабораторной работе № 3

Вариант 6

Название: Арифметическая обработка данных

Дисциплина: Микропроцессорные системы

Студент

ИУ6-64Б

ИУ6-64Б

(Группа)

Я.А.Гришина

С.П. Пантелеев

(И.О. Фамилия)

Преподаватель

Б.И. Бычков

(Подпись, дата)

(И.О. Фамилия)

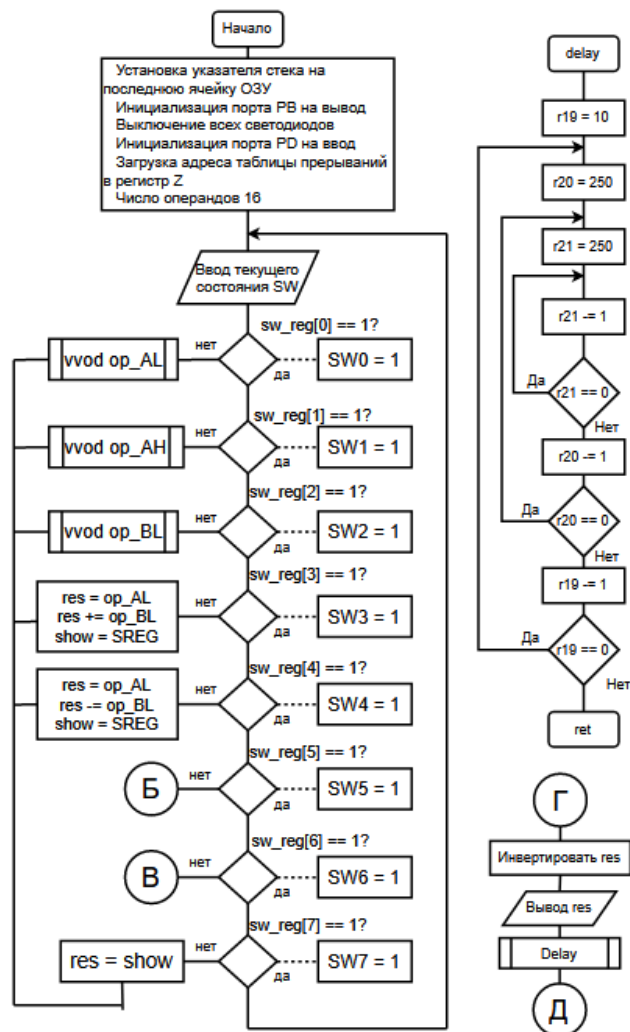
Москва, 2021

## Цель работы

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

**Задание 1.** Изучить программу для исследования арифметических операций в стартовом наборе STK500.

Схема алгоритма программы показана ниже, на рисунке 1.



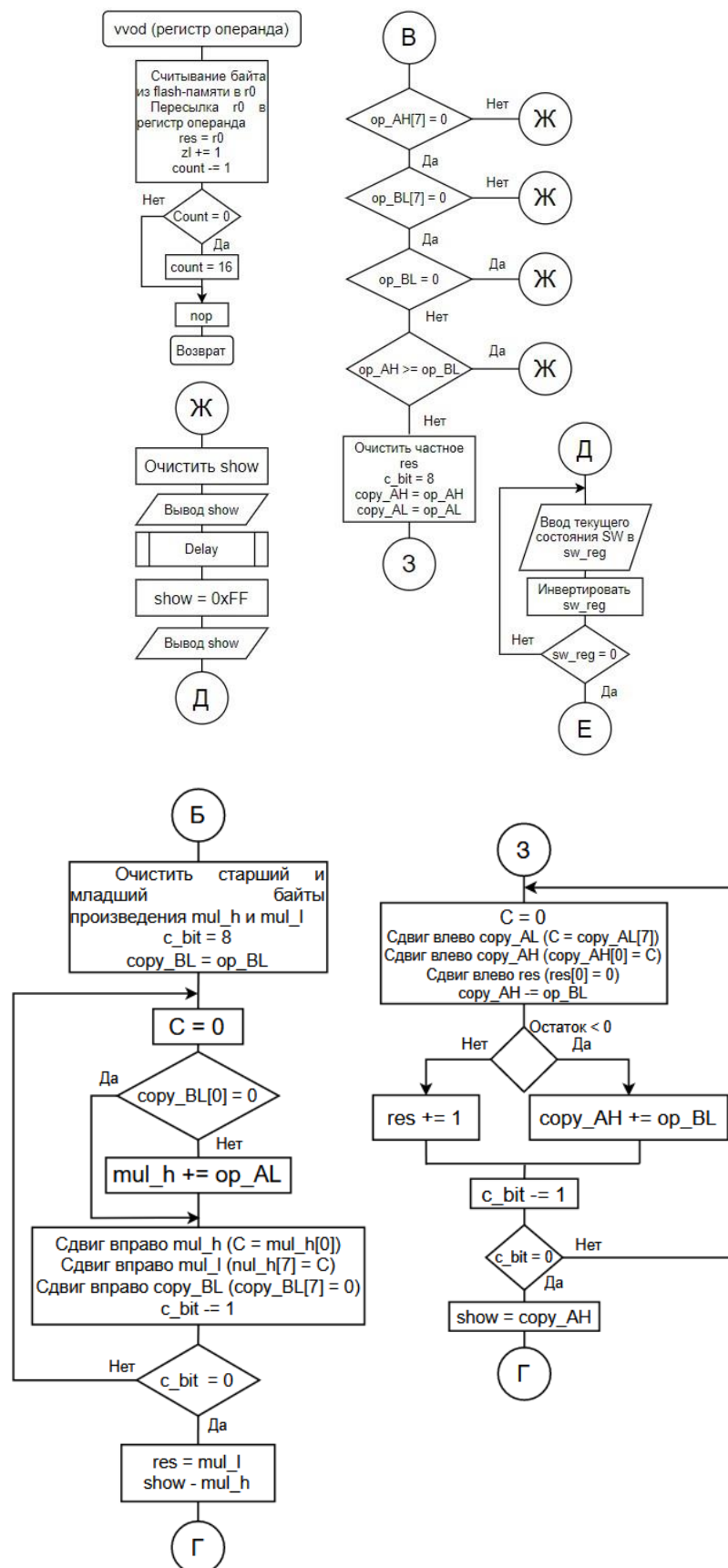


Рисунок 1 – Схема алгоритма программы

Изначальная программа:

```

;*****
;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 OЗУ
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 ;счётчик циклов
mov copy_BL,op_BL
L1: clc ;очистка флага C
sbrs copy_BL,0 ;проверка младшего бита множителя
add mul_h,op_AL ;прибавление множимого AL
ror mul_h ;сдвиг вправо
ror mul_l ; 2-х байтов произведения
lsr copy_BL ;сдвиг множителя вправо
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: sbrs op_AH,7 ;ошибки исходных данных
rjmp error
sbrs 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 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
d1: ldi r20,250
d2: ldi r21,250
d3: dec r21
brne d3
dec r20
brne d2
dec r19
brne d1
ret
; Таблица операндов в шестнадцатеричном представлении
tabl_op: .db 0xE5,0x10,0x1E,0xAA,0x6C,0xC7,0x1D,0xE2,0xD7,0x9B

```

Ввести в таблицу операндов в конце программы вместо приведенных в тексте 10 операндов из таблицы вариантов.

; Таблица операндов в шестнадцатеричном представлении согласно варианту  
tabl\_op: .db 0xE4,0x96,0xA6,0x7F,0x5A,0x9E,0x2E,0x9C,0xDC,0xE9

**Задание 2.** Выполнить ряд примеров на сложение и вычитание, выбирая операнды слагаемых AL и BL нажатием кнопок SW0 и SW2.

Таблица 1. Результаты наблюдений операции сложения и вычитания

Число $A_2/A_{10}$	Число $B_2/B_{10}$	$A + B /$ $A - B$	Признаки: H-S-V-N-Z-C
11100100/228 Беззнаковое	10010110/150 Беззнаковое	01111010 (122) 01001110(78)	0 - - - 0 1 1 - - - 0 0
10100110/-90 Со знаком	01111111/127 Со знаком	00100101 (37) 00100111 (39)	1 0 0 0 0 1 1 1 1 0 0 0
01011010/90 Со знаком	10011110/-98 Со знаком	11111000 (-8) 11110101 (-68)	1 1 0 1 0 0 1 0 1 1 0 1
00101110/46 Со знаком	10011100/-100 Со знаком	11001010 (-54) 10010010 (146*)	1 1 0 1 0 0 0 0 1 1 0 1

\*с учетом коррекции, т.к. флаг S (истинный знак числа с учетом переполнения) равен 0, то число положительное.

**Задание 3.** Выполнить ряд примеров умножения 8-разрядных двоичных чисел.

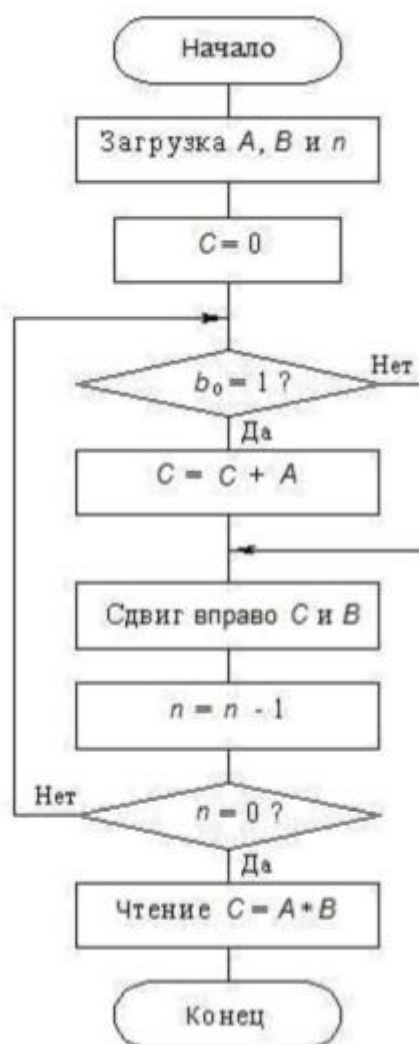


Рисунок 2 – Схема алгоритма умножения, начиная с младших разрядов множителя

Таблица 2. Результаты наблюдений операции умножения натуральных чисел

Число $A_2/A_{10}$	Число $B_2/B_{10}$	MUL Младший Байт произведения	RES Старший Байт произведения	Полный результат
0101 1010 / 90	1001 1110 / 158	1000 1100	0011 0111	0011 0111 1000 1100 (14220)
0010 1110 / 46	1001 1100 / 156	0000 1000	0001 1100	0001 1100 0000 1000 (7176)
1110 0100 / 228	1001 0110 / 150	1001 1000	1000 0101	1000 0101 1001 1000 (34200)

**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.

Выполнить 2-3 примера на деление двоичных чисел, самостоятельно подобрав делимое и делитель.

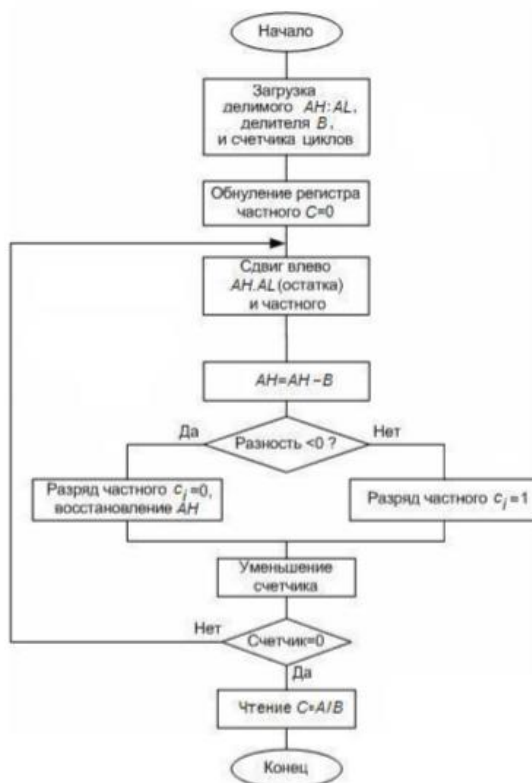


Рисунок 3 – Схема алгоритма деления с восстановлением остатка

Таблица 3. Результаты наблюдений операции деления

Число $AH_2/AH_h$ Старший Байт делимого	Число $AL_2/AL_h$ Младший Байт делимого	Число $BL_2/BL_h$ Делитель	DIV Частное	RES Остаток	Проверка
<u>0</u> 010 1110 / 2E	10011 1100 / 9C	<u>0</u> 101 1010 / 5A	1000 0100 / 132	0011 0100 / 52	$11932 = 132 \cdot 90 + 52$ (верно)
<u>0</u> 101 1010 / 5A	1001 1110 / 9E	<u>0</u> 111 1111 / 7F	1011 0110 / 182	0101 0100 / 84	$23198 = 127 \cdot 182 + 84$ (верно)
<u>0</u> 111 1110 / 7E	1111 1111 / FF	<u>0</u> 111 1111 / 7F	1111 1111 / 255	0111 1110 / 126	$32511 = 127 \cdot 255 + 126$

**Объяснение:**

$11932 = 0010\ 1110.1001\ 1100$  – делимое A(AH.AL)

$90 = 0101\ 1010$  – BL

$-90 = 1010\ 0110$  – BL в дополнительном коде

C – (c7) (c6) (c5) (c4) (c3) (c2) (c1) (c0) – частное  
 x – бит, свободно определяемый при сдвиге

0010 1110.1001 1100  
 +    1010 0110  
 1101 0100 – пробное вычитание B, так как разность меньше 0, то переполнения нет

0101 1101.0011 100x – сдвиг A влево  
 +    1010 0110  
 0000 0011 – вычитание B, остаток больше 0, значит разряд частного (c7) = 1

0000 0110.0111 00xx – сдвиг влево остатка  
 +    1010 0110  
 1010 1100 – вычитание B, остаток меньше 0, значит разряд частного (c6) = 0

0000 1100.1110 0xxx – сдвиг влево восстановленного АН  
 +    1010 0110  
 1011 0010 – вычитание B, остаток меньше 0, значит разряд частного (c5) = 0

0001 1001.1100 xxxx – сдвиг влево восстановленного АН  
 +    1010 0110  
 1011 1111 – вычитание B, остаток меньше 0, значит разряд частного (c4) = 0

0011 0011.100x xxxx – сдвиг влево восстановленного АН  
 +    1010 0110  
 1101 1001 – вычитание B, остаток меньше 0, значит разряд частного (c3) = 0

0110 0111.00xx xxxx – сдвиг влево восстановленного АН  
 +    1010 0110  
 0000 1101 – вычитание B, остаток больше 0, значит разряд частного (c2) = 1

0001 1010.0xxx xxxx – сдвиг влево остатка  
 +    1010 0110  
 1100 0000 – вычитание B, остаток меньше 0, значит разряд частного (c1) = 0

0011 0100.xxxx xxxx – сдвиг влево восстановленного АН  
 +    1010 0110  
 1101 1010 – вычитание B, остаток меньше 0, значит разряд частного (c0) = 0

$$C = 1000\ 0100_2 = 132_{10}$$

1101 1010  
 +    0101 1010

0011 0100 – прибавление В, восстановлен остаток  $АН = 52_{10}$

### **Вывод**

Исследована и изучена программа арифметических операций в наборе STK500. Выполнены операции сложения и вычитания двоичных чисел, проанализирован байт признаков результата (С – 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) формируемый в регистре SREG, а также умножения и деления целых чисел.