



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 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ
ТЕХНИКА

О Т Ч Е Т

по лабораторной работе № 3

Название: Арифметическая обработка данных

Дисциплина: Микропроцессорные системы

Вариант: 2

Студент

ИУ6-62Б

(Группа)

Ашуров Д. Н.
Марчук И. С.

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

Бычков Б. А.

(Подпись, дата)

(И.О. Фамилия)

Москва, 2022

Цели работы:

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

Ход работы.

Задание 1.

Изучить программу для исследования арифметических операций в стартовом наборе STK500, приведенную ниже.

Алгоритм программы приведен на рисунках 1 и 2.

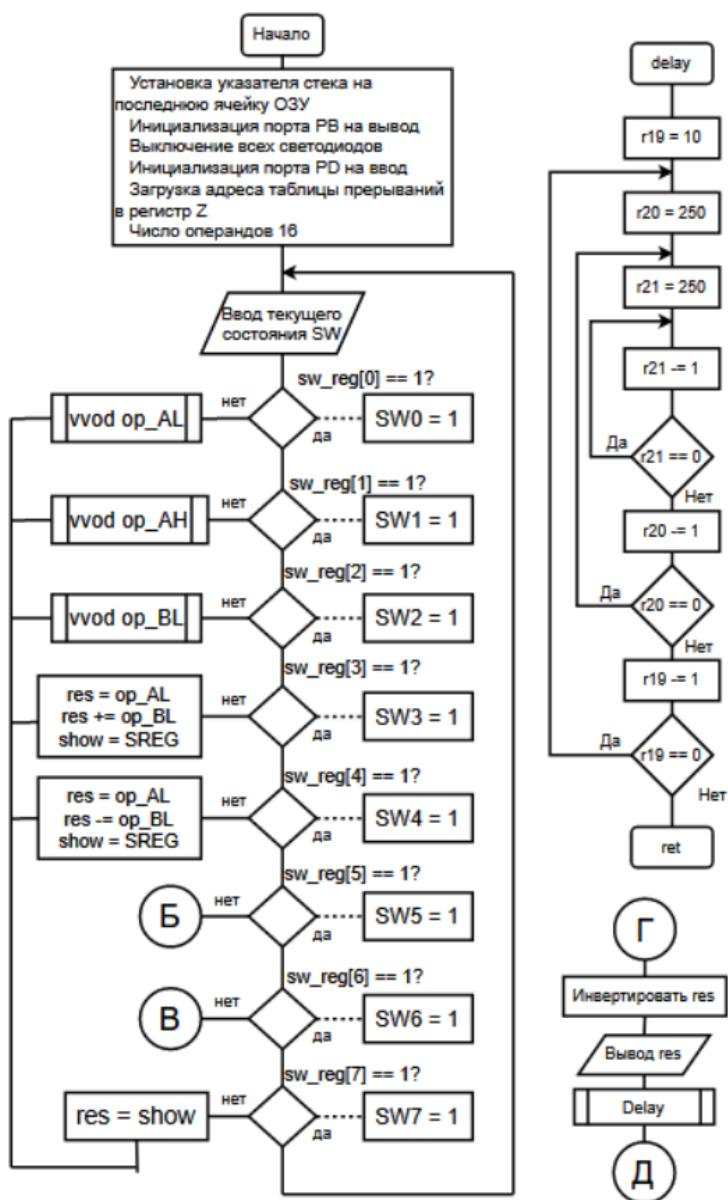


Рисунок 1 – схема алгоритма

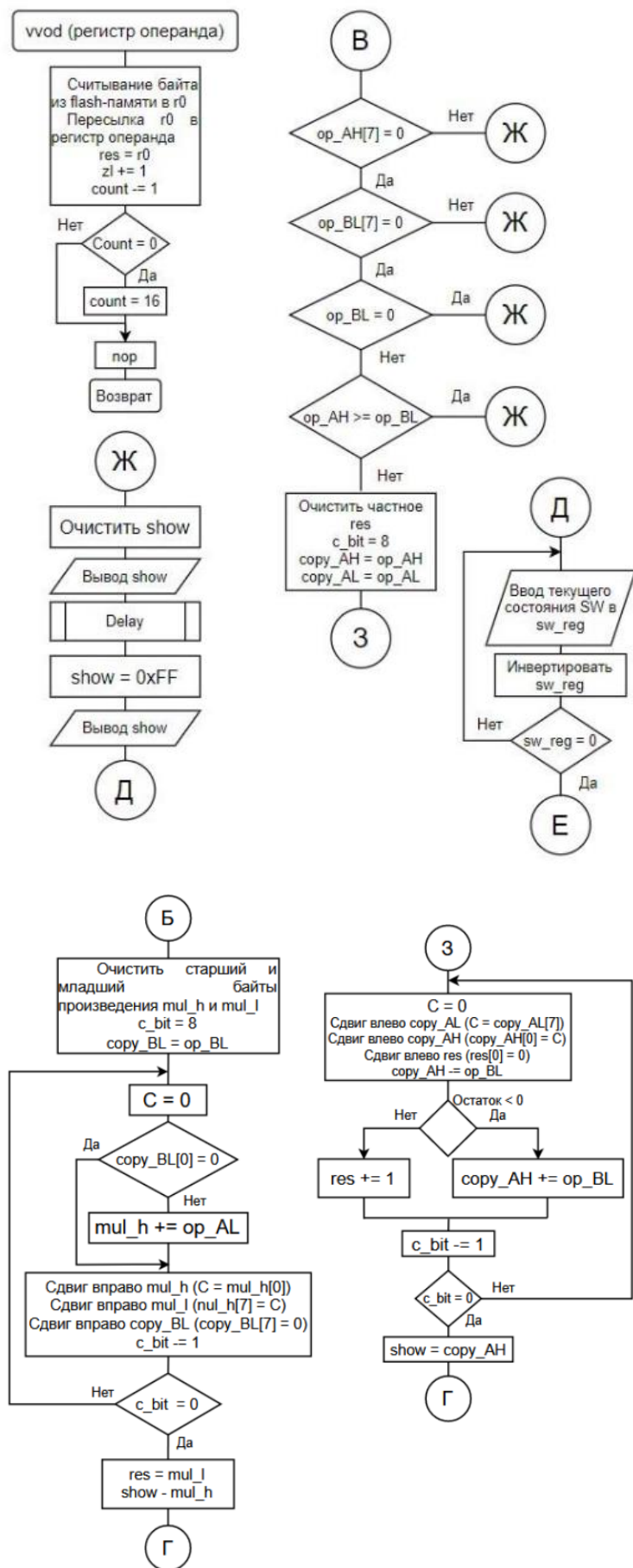


Рисунок 2 – схема алгоритма

Исходный код:

```
,*****
;
;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 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, 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
sbrs 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
brq 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
m1: ldi r20,1;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 0xA6, 0xE4, 0xE8, 0x3B, 0x7E, 0x6D, 0x4B, 0x6B, 0x56, 0xA6

```

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.

Результаты наблюдений показаны в таблице 1.

Таблица 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
1010 0110 / 166 Беззнаковое	1110 0100 / 228 Беззнаковое	1000 1010 (138) 1100 0010 (194)	0 - - - 0 1 0 - - - 0 1
1110 1000 / 232 Беззнаковое	0011 1011 / 59 Беззнаковое	0010 0011 (35) 1010 1101 (173)	1 - - - 0 1 1 - - - 0 0
1010 0110 / -90 Со знаком	0110 1011 / 107 Со знаком	0001 0001 (17) 0011 1011 (59)	100001 111000
1010 0110 / -90 Со знаком	1110 1000 / -24 Со знаком	1000 1110 (-114) 1011 1110 (-66)	010101 110101

Задание 3.

Выполнить ряд примеров умножения 8-разрядных двоичных чисел.

Схема алгоритма умножения показана на рисунке 2.

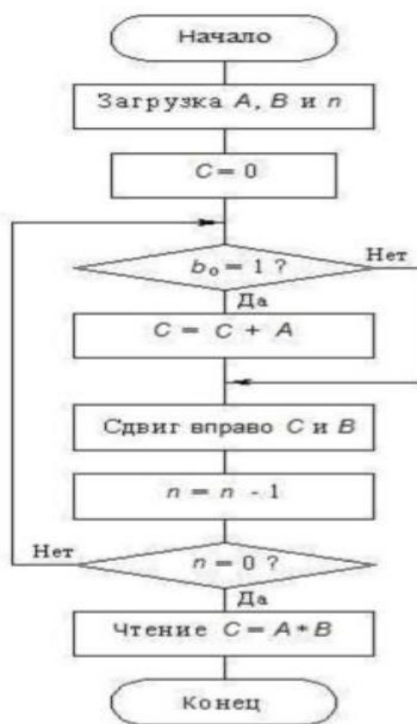


Рисунок 2 – схема алгоритма умножения

Результаты работы программы показаны в таблице 2.

Таблица 2 - результаты умножения

Число A_2/A_{10}	Число B_2/B_{10}	Результат
0110 1101 / 109	0100 1011 / 75	0001 1111 1110 1111 (8175)
0111 1110 / 126	0101 0110 / 86	0010 1010 0101 0100 (10836)
1110 1000 / 232	1110 0100 / 228	1100 1110 1010 0000 (52896)

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.

Подобрать пример с максимальными значениями делимого АН:АL и делителя В, при которых частное С будет равно 0b 1111 1111, изменив в случае необходимости программную таблицу операндов.

Схема алгоритма деления представлена на рисунке 3.

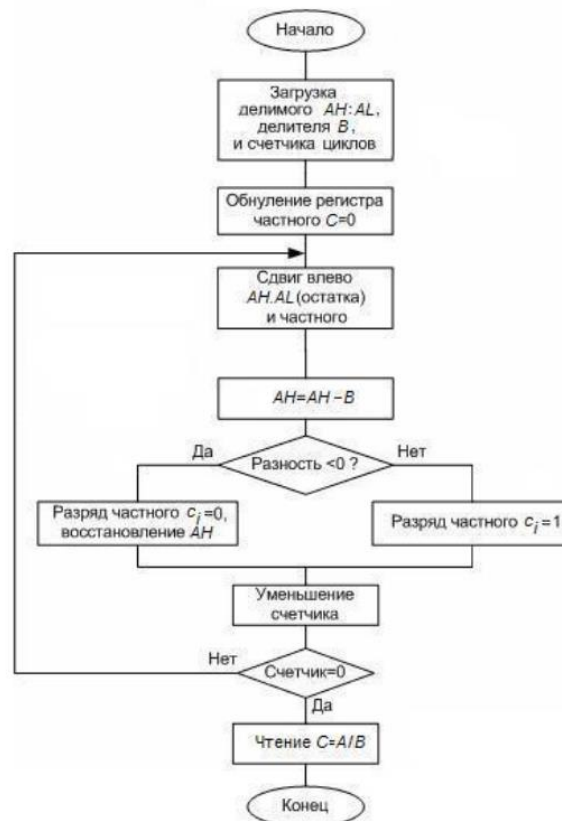


Рисунок 3 – схема алгоритма деления

Подберем максимальное делимое и делитель при частном $1111\ 1111_2 = 255_{10}$. Попробуем в качестве делителя взять максимально допустимое число $0111\ 1111_2 = 127_{10}$. Делимое $АН:АL = 127*255 + (127-1) = 32511_{10} = 0111\ 1110\ 1111\ 1111_2$. Добавив 126, мы добились получения максимального значения делимого,

не изменив результат, так как 126 будет остатком от деления. Данные представлены в последней строке таблицы 3.

Результаты деления представлены в таблице 3.

Таблица 3 – результаты деления

$(AH:AL)_2/(AH:AL)_{10}$	BL_2/BL_{10}	Частное	Остаток	Проверка
0110 1101 0110 0000 / 28000	1100 1000 / 200	1000 1100 / 140	0000 0000 / 0	$28000 = 200 \cdot 140$ + 0
0000 0101 0011 1001 / 1337	1110 0100 / 228	0000 0101 / 5	1100 0101 / 197	$1337 = 228 \cdot 5 +$ 197
0111 1110 1111 1111 / 32511	0111 1111 / 127	1111 1111 / 255	0111 1110 / 0 126	$32511 = 255 \cdot 127$ + 126
0110 1110 0011 1011 / 28219	0111 1101 / 125	1110 0001 / 225	0101 1110 / 94	$28219 = 125 \cdot$ 225 + 94

Работа алгоритма деления для делимого 28219_{10} и делителя 125_{10} :

$$28219_{10} = 0110\ 1110\ 0011\ 1011_2$$

$$125_{10} = 0111\ 1101_2$$

$$-125_{10} = 10000011_2 \text{ (дополнительный код)}$$

01101110.00111011

+

10000011

11110001

пробный

вычитание В

разность < 0, переполнения нет

11011100.01110110

+

10000011

01011111

сдвиг влево

вычитание В

остаток больше 0, $C7 = 1$

10111110.11101100

+

10000011

01000001

сдвиг влево остатка

вычитание В

остаток больше 0, $C6 = 1$

10000011.11011000

+

10000011

сдвиг влево остатка

вычитание В

остаток больше 0, $C5 = 1$

00000110

00001101.10110000

+

10000011

10010000

сдвиг влево остатка
вычитание В
остаток < 0 , $C4 = 0$

00011011.01100000

+

10000011

10011110

сдвиг влево
вычитание В
остаток < 0 , $C3 = 0$

00110110.11000000

+

10000011

10111001

сдвиг влево
вычитание В
остаток < 0 , $C2 = 0$

01101101.10000000

+

10000011

11110000

сдвиг влево
вычитание В
остаток < 0 , $C1 = 0$

11011011.00000000

+

10000011

01011110

сдвиг влево
вычитание В
остаток > 0 , $C0 = 1$

$C = 11100001 = E1 = 225$

Остаток = $01011110 = 5E = 94$

Вывод: в ходе лабораторной работы были рассмотрены выполнение арифметических операций над двоичными числами в микроконтроллере, таких как сложение, вычитание, умножение и деление; изменение флагов регистра SREG при выполнении операций; алгоритмы умножения и деления, при делении надо учитывать следующие ограничения: делитель не равен 0, значение делителя больше значения старшего байта делимого во избежание переполнения.