Национальный технический университет Украины

«Киевский политехнический институт»

Факультет информатики и вычислительной техники

Кафедра вычислительной техники

**Модульная работа №1**

*по курсу «Архитектура компьютеров»*

Выполнил

студент группы ИВ-73

Захожий Игорь

Номер зачетной книжки: 7308

Киев-2010

**Задание**

**Тема:** Выполнение операций сложения и вычитания с плавающей запятой в МК 51.

**Цель работы:** Изучение структуры памяти МК51, системы команд, форматов подачи данных и способов адресации операндов, получение навыков разработки программ выполнения простых арифметических операций над числами с плавающей запятой для МК 51.

**Теоретические сведения**

Сумма двух чисел X=2Px\*Mx и Y =2Py\*My, поданные в формате с плавающей запятой можно записать в виде:

2Px\*Mx+2Py\*My=2Pz\*Mz

Для сложения чисел с плавающей запятой необходимо привести их к общему порядку П, в качестве которого лучше выбрать больший порядок из двух Pz = max(Px, Py).

Во время этого уменьшения за счёт сдвига вправо мантиссы числа с меньшим порядком. В противном случае возникнет переполнение разрядной сетки мантиссы числа, которое преобразуется. После этого сумму чисел можно подать в виде

2Pz\*Mx+2Pz\*My`=2Pz\*( Mx+ My`)

Выполнение операции сложения или вычитания чисел с плавающей запятой в общем виде можно состоит из следующих этапов:

1) Выравнивание порядков.

2) Сумма мантисс.

3) Определение порядка результата.

4) Нормализация результата.

5) Округление результата.

6) Конечная нормализация результата.

*Формат числа с плавающей запятой*

Для реализации арифметических операций с плавающей запятой в МК51 числа подаются в виде 32 разрядного двоичного кода, где один байт отвечает за порядок числа и 3 за мантиссу числа.

Симметричный порядок подаётся в положительном коде и изменяется (-128) –(127), где старший разряд знаковый. Смещённый порядок использует положительное число без знака от 0 до 255(нулевой порядок – сдвиг +126).

**Выполнение**

Номер зачётки: 7308.

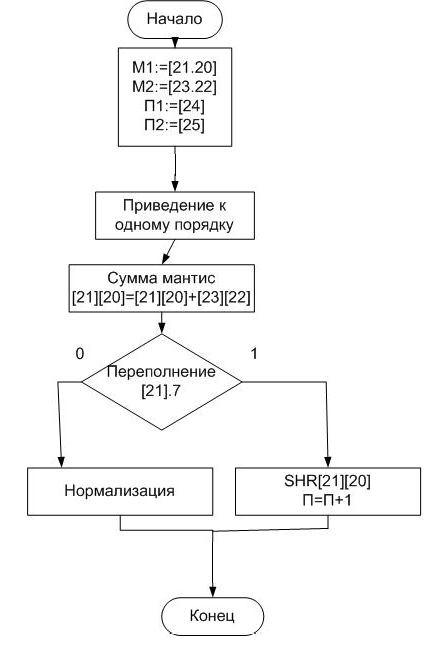
h1 = 0 – Операция сложение, длина мантиссы – 2 байта.

h4 = 1 – Формат подачи мантиссы – ДК.

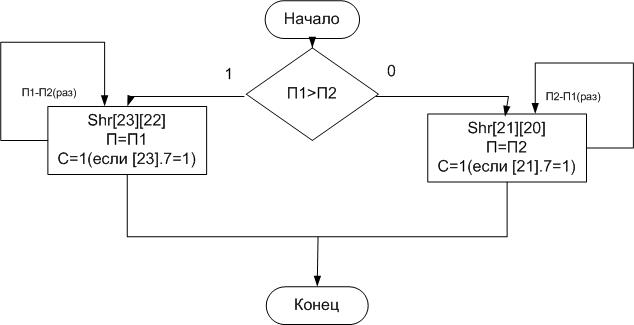
h2 = 0 - Формат подачи порядка – Симметричный.

h5 = 0- Первый операнд, результат - РПД; Второй операнд – ЗПД.

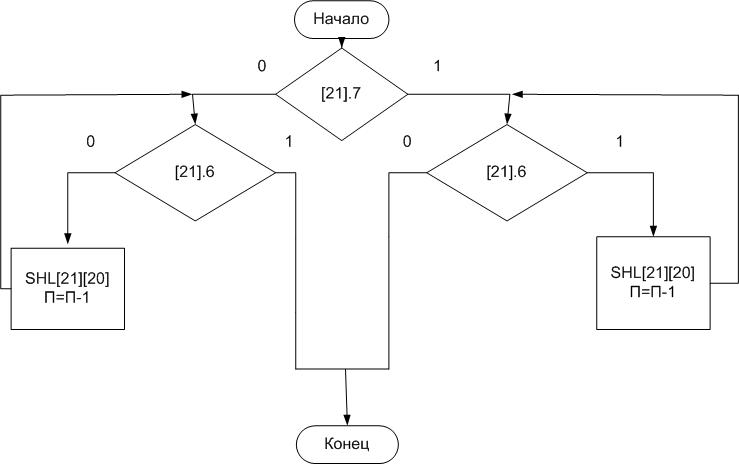
*Алгоритм*



*Приведение к одному порядку*



*Нормализация*

****

*Листинг*

ljmp begin

mant: ljmp mant1

sravnen: ljmp sravnen1

sdvig: ljmp sdvig1

normal: ljmp normal1

summ: ljmp summ1

norm: ljmp norm1

poryad: ljmp poryad1

begin:

; VVedenie chisel M1=[21.20] M2=[23.22] P1=[24] p2=[25]

mov r1, #20h;

mov @r1, #03h;

mov r1, #21h;

mov @r1, #80h;

mov r1, #22h;

mov @r1, #07h;

mov r1, #23h;

mov @r1, #00h;

mov r1, #24h;

mov @r1, #7;

mov r1, #25h;

mov @r1, #8;

;Privedenie k odnomu poryadku

acall poryad

;----------------------------------------------

; Summa mantis

acall sum

; Proverka na nul

mov a, #0

orl a, r6;

jnz noZ;

orl a, r5

jnz noZ

ljmp endd

;Perepolnenie

noZ: mov a, r5

; Proverka perepolneniya

jb acc.7, per

; Normalizaciya

acall normal

ljmp dalshe

; Normalizaciya pri perepolnenii

per: rrc a

mov r5, a

mov a, r6

rrc a

mov r6, a

inc r3

dalshe:

ljmp endd

;-----------------------Obyavlenie procedur--------------------------------------

;---------------------Procedure normalizacii-------------------------------------

normal1:

jb acc.7, otric

;Polozhitelnoe

mov r1,#20h

mov a, @r1

mov r6, a

mov r1,#21h

mov a, @r1

mov r5, a

; sdvigat` poka 0 v 6 razryade

jb6: jb acc.6, lll;

acall norm

ljmp jb6

lll: ljmp dalshe1;

; otricatelnoe

; sdvigat` poka 1 v 6 razryade

otric: jb acc.6, llll

ljmp dalshe1

llll: acall norm

ljmp otric

dalshe1: ret

;------------------------------Procedure---------------------------------------------

; sravnenie poryadkov mantis r4-na skolko sdvigat; r3-rezultat poryadka; r2- kto bolse

poryad1:

;------------Sravnenie poryadkov

sravnen1:

mov r1, #24h

mov a, @r1

mov r1, #25h

mov b, @r1

clr c

subb a, b

jnc snext

; poryadok 25

mov r3,b

add a, b

xch a, b

clr c

subb a, b

mov r4, a

mov r2,#2

ljmp send

snext:

; poryadok 24

mov r4, a

mov r2, #1

add a, b

mov r3, a

send:

;-----------------Preobrazovanie chisla s menshim poryadkom----------------------------

mov a,#2

anl a, r2

jz zff;

; Sdvig pervogo chisla

mov r1,#21h

mov a, @r1

mov r5, a

mov r1,#20h

mov a, @r1

mov r6, a

acall sdvig

;------------

; Sohranenie v pamyat

mov r1,#21h

mov a, r5

mov @r1, a

mov r1,#20h

mov a, r6

mov @r1, a

ljmp afts

zff:

; Sdvig vtorogo chisla

mov r1,#23h

mov a, @r1

mov r5, a

mov r1,#22h

mov a, @r1

mov r6, a

; viravninanie poryadkov-------------------------------

acall sdvig

; Sohranenie v pamyat

mov r1,#23h

mov a, r5

mov @r1, a

mov r1,#22h

mov a, r6

mov @r1, a

afts: ret

;------------------------Procedure viravninanie poryadkov-------------------------------

sdvig1:

mov a, #0

add a, r4

jz s33

mov a, r5

jb acc.7,s11;

s22: clr c

mov a, r5

rrc a

mov r5, a

mov a, r6

rrc a

mov r6,a

djnz r4, s22

ljmp s33

s11: clr c

cpl c

mov a, r5

rrc a

mov r5, a

mov a, r6

rrc a

mov r6,a

djnz r4, s22

s33: ret

;----------------------Procedure mantis preobrazovanie r5=M1; r6=M0-------------------

mant1:

mov a, r5

jb acc.7, mll

ljmp mnext

mll: anl a, #7Fh

mov r5, a

mov a, r6

cpl a

add a, #1

mov r6, a

mov a, r5

cpl a

addc a, #0

mov r5, a

mnext: ret;

;------------------------Procedure net perepolneniya normalizaciya---------------------

norm1:

clr c

mov a, r6

rlc a;

mov r6,a

mov a, r5

rlc a

mov r5, a

dec r3

ret

;------------------------Procedure summirovanie [21][20]= [21][20]+[23][22]-----------

summ1:

mov r1,#22h

mov a, @r1

mov r1,#20h

mov b, @r1

add a,b

mov @r1, a

mov r6, a

mov r1,#23h

mov a, @r1

mov r1,#21h

mov b, @r1

addc a,b

mov @r1, a

mov r5, a

ret

endd: end

**Тема:** Выполнение сложных арифметических операций с плавающей запятой в МК 51.

**Цель работы:** Изучение структуры памяти МК51, системы команд, форматов подачи данных и способов адресации операндов, получение навыков разработки программ выполнения сложных арифметических операций над числами с плавающей запятой для МК 51.

**Теоретические сведения**

*Умножение чисел с плавающей запятой*

Умножение двух чисел X=2Px\*Mx и Y =2Py\*My, поданные в формате с плавающей запятой можно записать в виде:

2Px\*Mx\*2Py\*My=2(Px+Py)\*( Mx\* My) .

Можна выделить следующие этапы умножения чисел с плавающей запятой:

1) Определение порядка результата.

2) Нахождение мантиссы результата.

3) Нормализация результата.(Приведение мантиссы к виду 2-1<Mz<1)

*Деление чисел с плавающей запятой*

Деление двух чисел X=2Px\*Mx и Y =2Py\*My, поданные в формате с плавающей запятой можно записать в виде:

2PzMz=(2Px\*Mx)/(2Py\*My)=2(Px-Py)\*Mx/My

Деление мантисс должно выполняться при выполнении условия Mx<My, которая не всегда выполняется при подаче мантисс в нормализованной форме. По этому перед началом деления мантиссу делимого всегда сдвигают вправо, чем обеспечивают её уменьшение в 2 раза.

2Px\*Mx= 2Px+1\* Mx\*2-1

Этапы деления чисел с плавающей запятой следующие:

1) Определение порядка результата.

2) Нахождение мантиссы результата.

3) Нормализация результата.(Приведение мантиссы к виду 2-1<Mz<1)

**Выполнение**

Номер зачётки : 7308

h0h1 = 00 – Операция деление, длина мантиссы – 3 байта.

h4 = 0 – Формат подачи мантиссы – ДК.

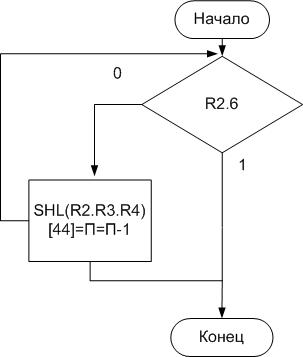
h2 = 1 - Формат подачи порядка – Симметричный.

h5 = 0 - Первый операнд, результат - 3ПД; Второй операнд – РПД.

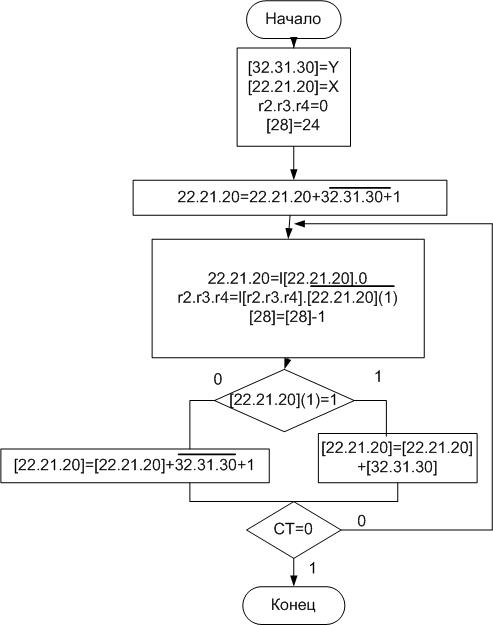
*Алгоритм*

****

*Нормализация*

****

*Алгоритм деления*

****

*Листинг*

begin:

; VVedenie chisel M1=[22.21.20] M2=[32.31.30] P1=[24] p2=[34]

mov r1, #22h;

mov @r1, #20h;

mov r1, #21h;

mov @r1, #00h;

mov r1, #20h;

mov @r1, #00h;

mov r1, #32h;

mov @r1, #40h;

mov r1, #31h;

mov @r1, #00h;

mov r1, #30h;

mov @r1, #00h;

mov r1, #24h;

mov @r1, #8;

mov r1, #34h;

mov @r1, #7;

;Opredelenie poryadka

mov r1, #34h

mov a, @r1

mov r7, a

mov r1, #24h

mov a, @r1

clr c;

cpl c;

subb a, r7

mov r1, #44h

mov @r1, a

;Sohranenie znaka rezultata

mov r1, #22h

mov a, @r1

mov r7, a

mov r1, #32h

mov a, @r1

xrl a, r7

mov r7, a

;---Perevod v PK

mov r1, #22h

mov a, @r1

jnb acc.7, m1ok

mov r1, #22h

mov a, @r1

mov r4, a

mov r1, #21h

mov a, @r1

mov r5, a

mov r1, #20h

mov a, @r1

mov r6, a

;-------OBSHAYA CHAST---------

mov a, r6

cpl a

add a, #1

mov r6, a

mov a, r5

cpl a

addc a, #0

mov r5, a

mov a, r4

cpl a

addc a, #0

mov r4, a

;-----------------------------

;--------SDVIG M1

mov r1, #22h

mov a, r4

mov @r1, a

mov r1, #21h

mov a, r5

mov @r1, a

mov r1, #20h

mov a, r6

mov @r1, a

m1Ok:

mov r1, #32h

mov a, @r1

jnb acc.7, m2ok

mov r1, #32h

mov a, @r1

mov r4, a

mov r1, #31h

mov a, @r1

mov r5, a

mov r1, #30h

mov a, @r1

mov r6, a

;-------OBSHAYA CHAST---------

mov a, r6

cpl a

add a, #1

mov r6, a

mov a, r5

cpl a

addc a, #0

mov r5, a

mov a, r4

cpl a

addc a, #0

mov r4, a

;-----------------------------

mov r1, #32h

mov a, r4

mov @r1, a

mov r1, #31h

mov a, r5

mov @r1, a

mov r1, #30h

mov a, r6

mov @r1, a

m2ok:

;------------------SDVIG---------

;--------SDVIG M1

mov r1, #22h

mov a, @r1

clr c

rrc a

mov @r1, a

mov r1, #21h

mov a, @r1

rrc a

mov @r1, a

mov r1, #20h

mov a, @r1

rrc a

mov @r1, a

;--------------------Delenie----------------------Rezult R2.R3.R4

mov r2, #0

mov r3, #0

mov r4, #0

mov r1, #28h

mov a, #24

mov @r1, a

;----pervaya iteraciya

mov r1, #20h

mov a, @r1

mov r5, a

mov r1, #30h

mov a, @r1

cpl a

clr c

cpl c

addc a, r5

mov r1, #20h

mov @r1, a

;----vtoraya iteraciya

mov r1, #21h

mov a, @r1

mov r5, a

mov r1, #31h

mov a, @r1

cpl a

addc a, r5

mov r1, #21h

mov @r1, a

;-----tretiya iteraciya

mov r1, #22h

mov a, @r1

mov r5, a

mov r1, #32h

mov a, @r1

cpl a

addc a, r5

mov r1, #22h

mov @r1, a

nachalo:

;------------ leviy sdvig

mov r1, #20h

mov a, @r1

clr c

rlc a

mov @r1, a

mov r1, #21h

mov a, @r1

rlc a

mov @r1, a

mov r1, #22h

mov a, @r1

rlc a

mov @r1, a

;-----------------Sdvig r2.r3.r4

cpl c

mov a, r4

rlc a

mov r4, a

mov a, r3

rlc a

mov r3, a

mov a, r2

rlc a

mov r2, a

;--------------[28]=[28]-1

mov r1, #28h

mov a, @r1

dec a

mov @r1, a

;--------------

mov r1, #22h

mov a, @r1

jb acc.7, metka1

;----pervaya iteraciya

mov r1, #20h

mov a, @r1

mov r5, a

mov r1, #30h

mov a, @r1

cpl a

clr c

cpl c

addc a, r5

mov r1, #20h

mov @r1, a

;----vtoraya iteraciya

mov r1, #21h

mov a, @r1

mov r5, a

mov r1, #31h

mov a, @r1

cpl a

addc a, r5

mov r1, #21h

mov @r1, a

;-----tretiya iteraciya

mov r1, #22h

mov a, @r1

mov r5, a

mov r1, #32h

mov a, @r1

cpl a

addc a, r5

mov r1, #22h

mov @r1, a

ljmp forU

metka1:

;----pervaya iteraciya

mov r1, #20h

mov a, @r1

mov r5, a

mov r1, #30h

mov a, @r1

add a, r5

mov r1, #20h

mov @r1, a

;----vtoraya iteraciya

mov r1, #21h

mov a, @r1

mov r5, a

mov r1, #31h

mov a, @r1

addc a, r5

mov r1, #21h

mov @r1, a

;-----tretiya iteraciya

mov r1, #22h

mov a, @r1

mov r5, a

mov r1, #32h

mov a, @r1

addc a, r5

mov r1, #22h

mov @r1, a

forU:

mov r1, #28h

mov a, @r1

add a, #0

jnb acc.7, nachalo

;----Normalizaciya-------------------------------

mov r1, #44h

mov a, @r1

mov r5, a

mov a, r2

jb6: jb acc.6, lll;

clr c

mov a, r5

subb a, #1

mov r5, a

mov a, r4

clr c

rlc a

mov r4, a

mov a, r3

rlc a

mov r3, a

mov a, r2

rlc a

mov r2, a

ljmp jb6

;--------------Vozvrashenie znaka r2, r3, r4

lll: mov r1, #44h

mov a, r5

mov @r1, a

mov a, r7

jnb acc.7, endd

mov a, r4

cpl a

add a, #1

mov r4, a

mov a, r3

cpl a

addc a, #0

mov r3, a

mov a, r2

cpl a

addc a, #0

mov r2, a

;----Zapis v pamyat [40][41][42]

endd: mov r1, #40h

mov a, r2

mov @r1, a

mov r1, #41h

mov a, r3

mov @r1, a

mov r1, #42h

mov a, r4

mov @r1, a

end