МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ УКРАИНЫ

«КИЕВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ»

**Протокол по лабораторной работе №4**

Компьютерная арифметика

Выполнил:

Студент I курса ФИВТ ИВ - 91

Гурбан Мирослав.

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КИЕВ 2010

**Тема**: Дослідження методів виконання арифметичних операцій над операндами з фіксованою комою та побудування мікропрограмного керування операційними пристроями.

**Ціль роботи:** Вивчити методи реалізації різноманітних арифметичних операцій, будування мікропрограмного забезпечення виконання операцій, одержати навички в підготовці до проектування арифметичних пристроїв, мікропрограмного управління та операційних пристроїв.

**Завдання:** Вивчити методи реалізації операцій множення і ділення операндів з фіксованою комою, скласти блок-схему та розробити програму виконання методу обчислення, який зазначений у варіанті, кінцевий та проміжний результати подати у таблиці. Скласти функціональний мікроалгоритм та таблицю логічного моделювання роботи операційного пристрою.

**Виконання завдання:**

*Вариант 24*

Виконати операцію множення у оберненому коді молодшими розрядами вперед із зсувом праворуч.

Десятичные числа:

A10 = 442

B10 = 612

Переведем в двоичную систему:

A2 = 1101110102

B2 = 10011001002

Запишем числа с плавающей запятой:

A2 ПЗ = 00.110111010.1001

B2 ПЗ = 00.1001100100.1010

Определение знака результата:

|  |  |
| --- | --- |
| ⊕ | 00 |
| 00 |
|  | 00 |

Определение порядка результата:

|  |  |
| --- | --- |
| + | 1001 |
| 1010 |
|  | 10011 |

**Схема сумматора:**

SM

0 n 1 n

n

n

RG2

1 n

RG1

0 n

RG2(n)

n

RG3

0 n

**ГСА**

1

CT = 0

RG2(0)=1

1

0

0

RG1 := 0.r[RG1]

RG2 := RG1(n).r[RG2]

CT := CT - 1

RG1 := RG1+RG2

0

RG1 := 0

RG2 := A

RG3 := B

CT := CT +1

**Выполним операцию умножения:**

|  |  |  |  |
| --- | --- | --- | --- |
| № такту | RG1 | RG2 | RG3 |
| 1 | 0000000000  →  0000000000 | 1101110100  →  0110111010 | 1001100100 |
| 2 | →  0000000000 | →  0011011101 |  |
| 3 | +RG3  0000000000  1001100100 | →  0001101110 |  |
| 1001100100  →  0100110010 |
| 4 | →  0010011001 | 0000110111 |  |
| 5 | +RG3  0010011001  1001100100 | 1000011011 |  |
| 1011111101  →  0101111110 |
| 6 | +RG3  0101111110  1001100100 | →  0100001101 |  |
| 1111100010  →  0111110001 |
| 7 | +RG3  0111110001  1001100100 | 1010000110 |  |
| 1 | 0001010101  →  1000101010 |
| 8 | →  0100010101 | 0101000011 |  |
| 9 | +RG3  0100010101  1001100100 | →  1010100001 |  |
| 1101111001  →  0110111100 |
| 10 | +RG3  0110111100  1001100100 | →  0101010000 |  |
| 1 | 0000100000  →  1000010000 |

**Код программы:**

Результат:

00.10000100000101010000.10011

program add;

type

TRegister=array [1..20] of byte;

procedure dec\_bin(dec:integer; var bin:TRegister);

var

i, j:byte;

begin

if dec<0 then

begin

bin[20]:=1; bin[19]:=1; dec:=-dec;

end

else

begin

bin[20]:=0; bin[19]:=0;

end;

i:=1;

while dec<>0 do

begin

bin[i]:=dec mod 2; dec:=dec div 2; i:=i+1;

end;

for j:=i to 18 do

bin[j]:=0;

if bin[20]=1 then

begin

for i:=1 to 18 do

if bin[i]=0 then bin[i]:=1

else bin[i]:=0;

bin[1]:=bin[1]+1;

for i:=1 to 18 do

begin

if bin[i]>1 then

begin

bin[i+1]:=bin[i+1]+1; bin[i]:=bin[i] mod 2;

end;

end;

end;

end;

procedure sub(var rx, ry, rr:TRegister);{rr:=rx-ry}

var

i:byte;

begin

for i:=1 to 20 do rr[i]:=0;

for i:=1 to 20 do

if ry[i]=0 then ry[i]:=1

else ry[i]:=0;

ry[1]:=ry[1]+1;

for i:=1 to 20 do

if ry[i]>1 then

begin

if i<20 then ry[i+1]:=ry[i+1]+1;

ry[i]:=ry[i] mod 2;

end;

for i:=1 to 20 do

begin

rr[i]:=rr[i]+rx[i]+ry[i];

if rr[i]>1 then

begin

if i<20 then rr[i+1]:=rr[i+1]+1;

rr[i]:=rr[i] mod 2;

end;

end;

end;

var

rg1, rg2, rg3:TRegister;

i, j:byte;

nom:integer;

begin

write('Enter A(decimal): '); read(nom); dec\_bin(nom, rg1);

write('Enter B(decimal): '); read(nom); dec\_bin(nom, rg2);

sub(rg1, rg2, rg3);

writeln('Result');

if rg3[19]<>rg3[20] then writeln('Over flow')

else

begin

write(rg3[20], rg3[19], '.');

for i:=18 downto 1 do write(rg3[i]);

end;

writeln;

readln; readln;

end.