БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра программного обеспечения информационных технологий

Факультет ФНиДО

Специальность ПОИТ

Контрольная работа № 1

по дисциплине «Основы компьютерной техники»

Вариант № 18

Выполнила: Карпеко Н. Г.

Договор № 696 от 16.05.2017г.

Минск 2017

Номер договора (вместо номера зачетной книжки) – 696 (от 16.05.2017г).

Номер варианта = 18 (96/26= 3 ост 18).

Задание 1.1

Найти значения С1, С2, С3, С4, определяемые выражениями:

1). С1 = А + В, 2). С2 = А - В, 3). С3 = В - А, 4). С4 = - А -В,

А = - 339010; В = + 87510.

Использовать обратный код.

При выполнении задания операнды А и В представить в двоично-десятичной системе, сформировать для них прямые коды и, используя заданный код по правилам двоично-десятичной арифметики, найти значения С1, С2, С3, С4.

Результат представить в прямом коде.

Решение.

При выполнении операций с алгебраическими двоично-десятичными числами на базе средств, используется обработка чисел по правилам двоичной арифметики с ограниченным количеством корректирующих операций.

Прямой двоично-десятичный код операндов А и В:

А10 = - 339010;

[А]пк = 1. 0011 0011 1001 00002-10,

В = + 87510;

[B] пк = 0. 0000 1000 0111 01012-10.

Обратный и инверсный модифицированный код А и В:

[А] мик = 11.1100 1100 0110 1111

[B] мок = 00. 0000 1000 0111 0101.

Расчеты выражений для C1, C2, C3, C4 выполнены таким образом:

С1 = А + В

[С1] пк : \* \*

11.1100 1100 0110 1111 [А]мик =[А]мок + 6

+ 00. 0000 1000 0111 0101 [B]мок

11. 1101 0100 1110 0100 - сумма ([А]мик и [B]мок по правилам

суммирования двоичного кода

+ 0110 0110 - коррекция в тедрадах, где был . 11. 1101 1010 1110 1010 перенос на 6.

11. 0010 0101 0001 0101 - [С1]пк

- 2 5 1 5 - С110 (десятичный эквивалент С1)

При выполнении первого суммирования по правилам двоичной арифметики возникающий перенос (в тетрадах, отмеченных знаком \*) унес лишнюю 6-ку (двоичный перенос унес из тетрады 16, а по правилам десятичного сложения он должен унести 10), т.е. избыточная «6» (использование инверсного кода числа вместо обратного) исчезает в тетрадах, где был перенос, и сохраняется в тетрадах, где переноса не было. Поэтому где был перенос, выполнена коррекция на +6, после чего во всех тетрадах будет избыточная 6, что позволяет перейти от такой записи к прямому коду результата за счет инвертирования тетрад модульной части.

2), 3). Формирование С2 и С3: С2=А-В, С3= В-А.

В этих выражениях операции сложения - за счет сложения абсолютных значений операндов и присвоением результату знака одного из операндов.

Поэтому, С2 и С3 вычисляем так:

[А]пк = 1. 0011 0011 1001 0000,

[ А]пк =0. 0011 0011 1001 0000

[B] пк  = 0. 0000 1000 0111 0101

[ B] пк = 0. 0000 1000 0111 0101

|С2| = |С3|:

\* \*

0. 0011 0011 1001 0000 -[| А|]пк

+ 0. 0000 1000 0111 0101 -[| B|] пк

0. 0011 1100 0000 0101

+ 0110 0110 - коррекция на 6

0.0100 0010 0110 0101 - [| С2|] пк = [|С3|] ПК

4 2 6 5 - |С2|10  = |С3|10  (десятичный эквивалент)

Таким образом, [С2] пк = 1.0100 0010 0110 0101,

[С3] пк = 0.0100 0010 0110 0101.

4). С4 = - А – В = -(- А) – В

[А]пк = 1. 0011 0011 1001 0000,

[A]ок =00. 0011 0011 1001 0000

[B] пк = 0. 0000 1000 0111 0101

[B] мик = 11. 1111 0111 1000 1010

[С4] пк :

\* \*

00. 0011 0011 1001 0000 [-А] мок

+ 11. 1111 0111 1000 1010 [-B] мик = [В] мок + 6

100. 0010 1011 0001 1010 - учет переноса при суммировании

+ 1 обратных кодов

00. 0010 1011 0001 1011

+ 1010 1010 - коррекция на 10, где не было переноса

00. 0010 0101 0010 0101 - [С4] мок=[С4]пк

+ 2 5 1 5 - С410 (десятичный эквивалент).

Коррекция на 10 выполняется в тетрадах, где не было переноса (прибавлением 10, которая является дополнительным кодом -6, с блокировкой переноса из тетрады заменяется операция вычитания 6 из кода тетрады). Тогда, после коррекции во всех тетрадах будет точное значение и эта запись соответствует искомому прямому коду С1.

Ответ: [С1] ПК = 11. 0010 0101 0001 0101 (-251510);

[С2] ПК = 1.0100 0010 0110 0101 (-426510);

[С3] ПК = 0.0100 0010 0110 0101 (426510);

[С4] ПК = 00. 0010 0101 0010 0101 (251510);

Задание 1.2

Выполнить операции над числами А и В с плавающей точкой.

1). знак и абсолютное значение порядка операнда А = - 2;

2). знак и абс. значение мантиссы операнда А = + 0.56;

3). знак и абс. значение порядка операнда В = - 0;

4). знак и абс. значение мантиссы операнда В = - 0.51;

5). «д» - **дополнительны**й код.

6). операция над А и В - дсво - **деление с восстановлением** остатка;

7). порядки и мантиссы операндов А и В представить в двоичной системе счисления;

8). сформировать для А и В прямые коды. Разрядность модуля порядка д. б = 3, разрядность модуля мантиссы = 6 .

9).Результат (порядок и мантисса) представить в прямом коде в нормализованной форме.

Решение

Переведём числа в двоичную систему счисления:

0,56 o,12 0,24 0,48 0,96 0,92 0,84

\* 2 \* 2 \* 2 \* 2 \* 2 \* 2 \* 2

1,12 0,24 0,48 0,96 1,92 1,84 1,68

0,51 0,02 0,04 0,08 0,16 0,32 0,64

\* 2 \* 2 \* 2 \* 2 \* 2 \* 2 \* 2

1,02 0,04 0,08 0,16 0,32 0,64 1,28

Таким образом:

0.5610= 0.10001112 ~ 00.100100(мод. доп. к.)

0.5110 = 0.1000012 ~ 00.100001 (мод. доп. к.)

Знак частного будет отрицательный, так как знаки мантисс делимого и делителя разные.

Предварительное значение порядка частного:

Сп = Ап + (– Вп):

11.010 [Ап]мдк

00.000 [Вп]мдк

11.010

1.010 - значение порядка в прямом коде.

Абсолютные значения операндов А и В имеют вид:

[|Ам|]мдк = 00.100100, [|Вм|]мдк =00.1000001, [-|Вм|]мдк = 11.0111110.

Найдем предварительное абсолютное значение мантиссы частного за счет выполнения восьми тактов деления.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 00.100100  11.011111 |  |  |
| Единица игнорируется, так как код дополнительный. | ~~1~~00.000011  00.000110  11.011111 | 1 | остаток >0  арифметический сдвиг влево |
|  | 11.100101  00.100001 | 0 | остаток <0  восстановление |
| Единица игнорируется, так как код дополнительный. | ~~1~~00.000110  00.001100  11.011111 |  | арифметический сдвиг влево |
|  | 11.101011  00.100001 | 0 | так как остаток<0  восстановление |
| Единица игнорируется, так как код дополнительный. | ~~1~~00.001100  00.011000  11.011111 |  | арифметический сдвиг влево |
|  | 11.110111  00.100001 | 0 | так как остаток<0  восстановление |
| Единица игнорируется, так как код дополнительный. | ~~1~~00.011000  00.110000  11.011111 |  | арифметический сдвиг влево |
| Единица игнорируется, так как код дополнительный. | ~~1~~00.001111  00.011110  11.011111 | 1 | так как остаток>0  арифметический сдвиг влево |
|  | 11.111101  00.100001 | 0 | так как остаток<0  восстановление |
| Единица игнорируется, так как код дополнительный. | ~~1~~00.011110  00.111100  11.011111 |  | арифметический сдвиг влево |
| Единица игнорируется, так как код дополнительный | ~~1~~00.011011  00.110110  11.011110 | 1 | так как остаток>0  арифметический сдвиг влево |
| Единица игнорируется, так как код дополнительный | ~~1~~00.010100  00.101000 | 1 | так как остаток>0  арифметический сдвиг влево |

Учитывая знаки остатков, полученные на восьми тактах, абсолютное предварительное значение мантиссы искомого частного равно:

[|См|]пк = 1.0001011.

С учетом округления:

[|См|]пк = 1.000110.

Нормализуем мантиссу частного, сдвинув ее на один разряд вправо, а предварительное значение порядка частного увеличим на единицу.

[|См|]пк = 0.100011;

00.001 [с1 п] мпк `

+ 00.001 +1

00.010 [с1 п] мпк

Ответ:

[|См|]пк = 0.100011;

[Сп]пк = 0.010.