Министр науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет информационных технологий и программирования

Домашнее задание № 1

Выполнение арифметических операций с двоичными числами

Выполнил студент группы № M3101
Михеев Артем Романович
Подпись:

Проверил:
Бабич Мария Сергеевна

Санкт-Петербург
2020

<u>Цель задания</u> - овладеть простейшими навыками перевода чисел в различные системы счисления и выявить ошибки, возникающие из-за их ограниченной разрядности.

Задание, вариант 17

A = 2678
C = 16160

1. Получить набор десятичных чисел:
   - X1 = A
   - X2 = C
   - X3 = A+C
   - X4 = A+C+C
   - X5 = C-A
   - X6 = 65536-X4
   - X7 = -X1
   - X8 = -X2
   - X9 = -X3
   - X10 = -X4
   - X11 = -X5
   - X12 = -X6

   Выполнить перевод чисел X1..X12 в двоичную С.С., получив их двоичных эквиваленты B1..B12 соответственно. Для представления двоичных чисел B1..B12 использовать 16-разрядный двоичный формат со знаком. Для контроля правильности перевода выполнить обратный перевод двоичных чисел в десятичные и подробно проиллюстрировать последовательность прямого и обратного перевода для чисел X1, B1, X7 и B7.
2. Выполнить следующие сложения двоичных чисел: B1+B2, B2+B3, B7+B8, B8+B9, B2+B7, B1+B8. Для представления слагаемых и результатов сложения использовать 16-разрядный двоичный формат со знаком. Результаты сложения перевести в десятичную С.С., сравнить с соответствующими десятичными числами. Дать подробные комментарии полученным результатам.

Вычисления

Задание 1

Из условий получаем числа:
X1 = 2678
X2 = 16160
X3 = 2678 + 16160 = 18838
X4 = 2678 + 16160 + 16160 = 34998
X5 = 16160 - 2678 = 13482
X6 = 65536 – 34998 = 30538
X7 = -2678
X8 = -16160
X9 = -18838
X10 = -34998
X11 = -13482
X12 = -30538

Переведём числа в двоичную систему, сначала первые 6, которые положительны, и, соответственно, с ними не придется проводить никаких манипуляций после обычного перевода в двоичную:

- X1 = 2048 + 512 + 64 + 32 + 16 + 4 + 2 = 2^11 + 2^9 + 2^6 + 2^5 + 2^4 + 2^2 + 2^1, получаем B1 = 0000101001110110
- X2 = 8192 + 4096 + 2048 + 1024 + 512 + 256 + 32 = 2^13 + 2^12 + 2^11 + 2^10 + 2^9 + 2^8 + 2^5, получаем B2 = 0011111100100000
- X3 = 16384 + 2048 + 256 + 128 + 16 + 4 + 2 = 2^14 + 2^11 + 2^8 + 2^7 + 2^4 + 2^2 + 2^1, получаем B3 = 0100100110010110
- X4 = 32768 + 2048 + 128 + 32 + 16 + 4 + 2 = 2^15 + 2^11 + 2^7 + 2^5 + 2^4 + 2^2 + 2^1, получаем B4 = 1000100010110110
- X5 = 8192 + 4096 + 1024 + 128 + 32 + 8 + 2 = 2^13 + 2^12 + 2^10 + 2^7 + 2^5 + 2^3 + 2^1, получаем B5 = 0011010010101010
- X6 = 16384 + 8192 + 4096 + 1024 + 512 + 256 + 64 + 8 + 2 = 2^14 + 2^13 + 2^12 + 2^10 + 2^9 + 2^8 + 2^6 + 2^3 + 2^1, получаем B6 = 0111011101001010

Теперь переведём оставшиеся 6 числа в двоичную С.С. Так как эти числа отрицательны, то после перевода нам нужно будет инвертировать все биты и прибавить 1.
- X7 = -(2678) = -(2048 + 512 + 64 + 32 + 16 + 4 + 2) = -(2^11 + 2^9 + 2^6 + 2^5 + 2^4 + 2^2 + 2^1), получаем B7 = -0000101001110110 = 1111010110001001 + 1 = 1111010110001010
- X8 = -(16160) = - (8192 + 4096 + 2048 + 1024 + 512 + 256 + 32) = -(2^13 + 2^12 + 2^11 + 2^10 + 2^9 + 2^8 + 2^5), получаем B8 = -0011111100100000 = 1100000011011111 + 1 = 1100000011100000
- X9 = -(18838) = -(16384 + 2048 + 256 + 128 + 16 + 4 + 2) = -(2^14 + 2^11 + 2^8 + 2^7 + 2^4 + 2^2 + 2^1), получаем B9 = -0100100110010110 = 1011011001101001 + 1 = 1011011001101010
- X10 = -(34998) = -(32768 + 2048 + 128 + 32 + 16 + 4 + 2) = -(2^15 + 2^11 + 2^7 + 2^5 + 2^4 + 2^2 + 2^1), получаем B10 = -1000100010110110 = 0111011101001001 + 1 = 0111011101001010
- X11 = -(13482) = -(8192 + 4096 + 1024 + 128 + 32 + 8 + 2) = -(2^13 + 2^12 + 2^10 + 2^7 + 2^5 + 2^3 + 2^1), получаем B11 = -0011010010101010 = 1100101101010101 + 1 = 1100101101010110
- X12 = -(30538) = -(16384 + 8192 + 4096 + 1024 + 512 + 256 + 64 + 8 + 2) = -(2^14 + 2^13 + 2^12 + 2^10 + 2^9 + 2^8 + 2^6 + 2^3 + 2^1), получаем B12 = -0111011101001010 = 1000100010110101 + 1 = 1000100010110110

Для перевода из чисел в двоичном формате со знаком, будем использовать такую последовательность действий:
1. Если самый левый бит = 1, то вычитаем из числа 1, инвертируем (~) его биты.
2. Переводим двоичное число в десятичное.

B1 = 0000101001110110 => X1 = 2^11 + 2^9 + 2^6 + 2^5 + 2^4 + 2^2 + 2^1 = 2678
B2 = 0011111100100000 => X2 = 2^13 + 2^12 + 2^11 + 2^10 + 2^9 + 2^8 + 2^5 = 16160
B3 = 0100100110010110 => X3 = 2^14 + 2^11 + 2^8 + 2^7 + 2^4 + 2^2 + 2^1 = 18838
B4 = 1000100010110110 = - ~(1000100010110110 - 1) = - ~(1000100010110101) = -0111011101001010 => X4 = -(2^14 + 2^13 + 2^12 + 2^10 + 2^9 + 2^8 + 2^6 + 2^3 + 2^1) = -30538
B5 = 0011010010101010 => X5 = 2^13 + 2^12 + 2^10 + 2^7 + 2^5 + 2^3 + 2^1 = 13482
B6 = 0111011101001010 => X6 = 2^14 + 2^13 + 2^12 + 2^10 + 2^9 + 2^8 + 2^6 + 2^3 + 2^1 = 30538
B7 = 1111010110001010 = - ~(1111010110001010 - 1) = - ~(1111010110001001) = -0000101001110110 => X7 = -(2^11 + 2^9 + 2^6 + 2^5 + 2^4 + 2^2 + 2^1) = -2678
B8 = 1100000011100000 = - ~(1100000011100000 - 1) = - ~(1100000011011111) = -0011111100100000 => X8 = -(2^13 + 2^12 + 2^11 + 2^10 + 2^9 + 2^8 + 2^5) = -16160

B9 = 1011011001101010 = - ~(1011011001101010 - 1) = - ~(1011011001101001) = -
0100100110010101 => X9 = -(2^14 + 2^11 + 2^8 + 2^7 + 2^4 + 2^2 + 2^1) = -18838
B10 = 0111011101001010 = 2^14 + 2^13 + 2^12 + 2^10 + 2^9 + 2^8 + 2^6 + 2^3 + 2^1 => X10
= 30538
B11 = 1100101101010110 = - ~(1100101101010110 - 1) = - ~(1100101101010101) = -
0011010010101010 => X11 = -(2^13 + 2^12 + 2^10 + 2^7 + 2^5 + 2^3 + 2^1) = -13482
B12 = 1000100010110110 = - ~(1000100010110110 - 1) = - ~(1000100010110101) = -
0111011101001010 => X12 = -(2^14 + 2^13 + 2^12 + 2^10 + 2^9 + 2^8 + 2^6 + 2^3 + 2^1) = -
30538

Как видно из обратного перевода, который и служил в качестве проверки, числа X4 и X10 при переводе в двоичное представление со знаком, перевелись так, что при обратном переводе мы получили совершенно другие числа. На самом деле, конечно, это не случайные числа, и получились они такими из-за того, что число X4 больше, чем 2^15-1, что означает, что 15-ый, то есть последний, бит был выставлен при переводе в двоичное представление, из-за чего при обратном переводе он интерпретировался как бит знака, и знак, как и значение по модулю, у числа поменялось.

Более подробно для чисел X1, B1, X7, B7:

X1:

Для того, чтобы найти, какие степени двойки входят в число X1, будем брать по модулю (%) 2 и делить нацело на 2, и соответственно получать i-тый бит числа X1 в бинарном виде. Получаем такие действия:

2678 % 2 = 0 => B1 = 0
2678 // 2 = 1339
1339 % 2 = 1 => B1 = 10
1339 // 2 = 669
669 % 2 = 1 => B1 = 110
669 // 2 = 334
334 % 2 = 0 => B1 = 0110
334 // 2 = 167
167 % 2 = 1 => B1 = 10110
167 // 2 = 83
83 % 2 = 1 => B1 = 110110
83 // 2 = 41
41 % 2 = 1 => B1 = 1110110
41 // 2 = 20
20 % 2 = 0 => B1 = 01110110
20 // 2 = 10
10 % 2 = 0 => B1 = 001110110
10 // 2 = 5
5 % 2 = 1 => B1 = 1001110110
5 // 2 = 2
2 % 2 = 0 => B1 = 01001110110
2 // 2 = 1
1 % 2 = 1 => B1 = 101001110110
1 // 2 = 0
Дополняя B1 нулями слева, чтобы было 16 бит, получаем B1 = 0000101001110110

B1:

Для перевода B1 в десятичную систему, будем на i-том шаге делать ничего, если i-тый бит равен нулю, иначе будем прибавлять 2^i.

B1 = 0000101001110110 => на позициях 1, 2, 4, 5, 6, 9, 11 стоят единицы, значит наша сумма будет 2^1 + 2^2 + 2^4 + 2^5 + 2^6 + 2^9 + 2^11 = 2678 => X1 = 2678

X7:

Так как у нас число отрицательно, то сначала забудем про знак, и, аналогично X1, будем брать по модулю 2 и делить нацело на 2, формируя B7, пока не получим 0. Потом же инвертируем полученные биты и прибавим 1, чтобы получить представление числа X7 в дополнительном коде.

Т.к. X7 = -X1, то просто возьмем уже готовое число в бинарном виде – 0000101001110110. Инвертируем у него биты, и получим 1111010110001001. Здесь сразу уже можно увидеть, что бит 15 = 1, что и соответствует отрицательности числа.

Теперь прибавим 1 и получим B7: 1111010110001001 + 1 = 1111010110001010 => B7 = 1111010110001010. Для проверки можно сложить B1 и B7, получим число 10000000000000000, первые 16 бит которого равны нулю, что соответствует числу 0, если у нас доступно всего 16 разрядов.

B7:

Так как бит 15 равен единице, то сначала вычтем из B7 единицу - 1111010110001010 – 1 = 1111010110001001, а потом инвертиурем биты у полученного числа 1111010110001001, получив 0000101001110110. Так как мы такое число уже переводили (B1), то знаем, что оно равно 2678 в десятичной С.С., но так как у нас был выставлен бит 15, нужно сделать его отрицательным и получить X7 = -2678.

Задание 2

Числа для этого задания:

B1 = 0000101001110110
B2 = 0011111100100000
B3 = 0100100110010110
B7 = 1111010110001010
B8 = 1100000011100000
B9 = 1011011001101010

B1 + B2 =

0000101001110110
\+ 0011111100100000 =

0000101001010110
\+ 0000000001000000
\+ 0011111100000000 =

0000100110010110
\+ 0000010000000000
\+ 0011110000000000 =

0000100110010110
\+ 0000100000000000
\+ 0011100000000000 =

0000100110010110
\+ 0001000000000000
\+ 0011000000000000 =

0100100110010110 = 18838 == X2

B2 + B3 =

0011111100100000
\+ 0100100110010110 =

0011111010110110
\+ 0000001000000000
\+ 0100100000000000 =

0011100010110110
\+ 0000100000000000
\+ 0100100000000000 =

0011100010110110
\+ 0001000000000000
\+ 0100000000000000 =

0100100010110110
\+ 0100000000000000 =

1000100010110110 =

- (65536 - X4) = -30538

B7 + B8 =

1111010110001010
\+ 1100000011100000 =

1111010101101010
\+ 0000000100000000
\+ 1100000000000000 =

1011011001101010
\+ 1000000000000000
\+ 1000000000000000 =

1011011001101010 = -X3 = -18838

B8 + B9 =

1100000011100000
\+ 1011011001101010 =

1100000011001010
\+ 0000000001000000
\+ 1011011001000000 =

1100000011001010
\+ 0000000010000000
\+ 1011011000000000 =

1111011101001010
\+ 1000000000000000 =

0111011101001010 =
65536 - X4 = 30538

B2 + B7 =

0011111100100000
\+ 1111010110001010 =

0011111010101010
\+ 0000001000000000
\+ 1111010000000000 =

0011110010101010
\+ 0000010000000000
\+ 1111010000000000 =

0011110010101010
\+ 0000100000000000
\+ 1111000000000000 =

0100010010101010
\+ 1111000000000000 =

0011010010101010
\+ 1000000000000000
\+ 1000000000000000 =

0011010010101010 =
X2 – X1 = 13482

B1 + B8 =

0000101001110110
\+ 1100000011100000 =

0000101001010110
\+ 0000000001000000
\+ 1100000011000000 =

0000101001010110
\+ 0000000010000000
\+ 1100000010000000 =

1100101101010110 =
X1 – X2 = -13482

Из выражений 2, 4 можно еще раз заметить, что происходит при “переполнении” – в примере 2, казалось бы, складываем 2 положительных числа, которые в десятичном виде были бы равны 34998, но из-за того что у нас выставляется единица в бите 15, мы получаем не 34998, а -30538. В 4-ом примере наоборот, но тоже проявляется этот эффект – складываем 2 отрицательных числа и должны получить -34998, а на самом деле получаем 30538, так как при сложении 15-ых битов они переносятся в 16-ый, и мы получаем не отрицательное а положительное число.

Также, из этих выражений можно заметить (если изначально это неизвестно) тот факт, что сложение и вычитание чисел в двоичном представлении в дополнительном коде это одна и та же операция. То есть, для вычитания числа b из a, мы должны записать число -b в дополнительном коде, а потом просто сложить их.

Выводы
Из проделанной работы самыми главными результатами являются знания о том, что необходимо осторожно работать с числами со знаком, когда разрядность у них ограничена, а так как изначально абсолютно все числа у любой ЭВМ имеют некоторую ограниченность (будь то 16, 32 или 64 бита/разряда), всегда нужно принимать это в учёт.