Информатика

Аудиторный практикум 2

ЧАСТЬ 1. АРИФМЕТИКА ДВОИЧНЫХ ЧИСЕЛ

Правила сложения двоичных чисел похожи на привычные нам: сложение происходит поразрядно справа налево, при этом важно помнить о переносе чисел в новый разряд.

В десятичной системе счисления, которую мы используем каждый день, всего 10 цифр (от 0 до 9). Когда мы складываем 1 и 9, получается 10. Но так как в одном разряде мы можем записать только цифру от 0 до 9, мы переносим единицу в следующий разряд.

Двоичная система счисления работает по такому же принципу, только с двумя цифрами: 0 и 1. Если сложить 1 и 1, мы получим переполнениеа значит, единица пойдёт в следующий разряд, результатом станет 10 (только не «десять», а «один-ноль»).

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 10$$

Вычислите:

а) 1011_2+111_2

б) 1111_2+11_2

Умножение в двоичной системе, как в десятичной, основано на сложении — и умении считать в столбик.
Сведем в таблицу правила умножения двоичных чисел:

$$0 \times 0 = 0$$
$$0 \times 1 = 0$$
$$1 \times 0 = 0$$
$$1 \times 1 = 1$$

Давайте подробнее рассмотрим пример – умножим 110 на 10.

$$\begin{array}{r} \times \begin{array}{r} 110 \\ 10 \end{array} \\ \hline 000 \\ + \; 110 \\ \hline 1100 \end{array}$$

Здесь мы воспользуемся привычным школьным «столбиком»: сначала умножаем верхнее число, 110, на 0, затем на 1, а потом складываем полученные два и получаем результат.
По сути, если мы умножаем число на ноль, то оно превращается в ноль, а если на единицу — остаётся неизменным, но сдвигается на число разрядов, равное номеру разряда этой единицы, как в обычном умножении:

110 × 0 = 000;
110 × 1 = 110.

Сдвигаем 110 на один разряд влево и складываем результаты промежуточных умножений:

000 + 1100 = 1100.

Мы получили 1100, потому что сместили результат умножения 110 × 1 на один разряд влево, а затем добавили один 0 справа — как в обычном умножении.

Вычислите:
в) $101_2 * 11_2$
г) $1011_2 * 101_2$

Правила двоичного вычитания тоже ничем не отличаются от десятичного. Мы также вычитаем поразрядно и, если нужно, занимаем единицу из старшего разряда.
Таблица вычитания выглядит так:

$$0 - 0 = 0$$
$$0 - 1 = 1$$
$$1 - 0 = 1$$
$$1 - 1 = 0$$

Заметьте, что 0 − 1 = 1. Это всё потому, что мы занимаем единицу из старшего разряда и получаем 10, или 2 в десятичной системе, а если вычесть из 10 число 1, получим 1 (ведь 2 − 1 = 1).
Перейдем к примерам, чтобы понять, как вычесть одно число из другого.

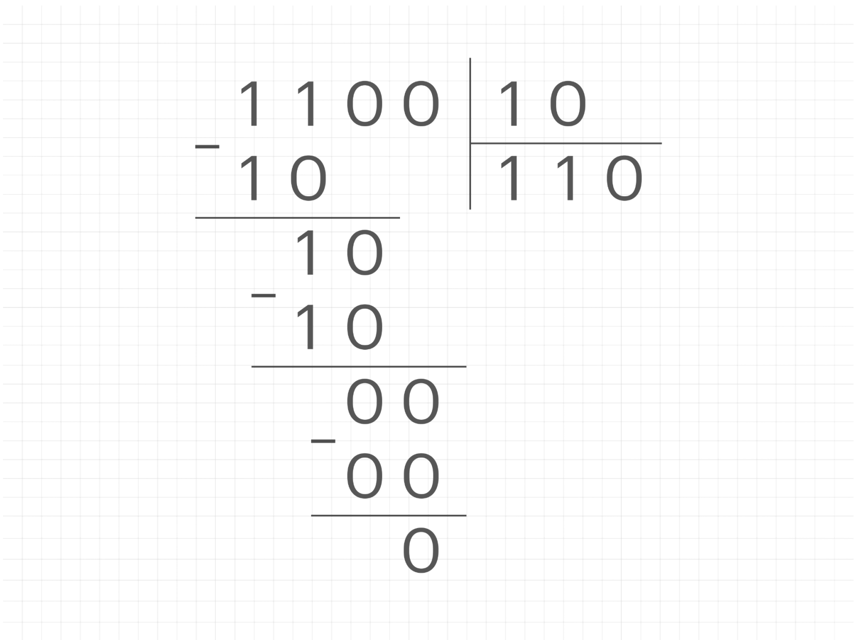
$$\begin{array}{r} 1011 \\ -\quad 101 \\ \hline 110 \end{array}$$

Алгоритм по разрядам:

Первый: 1 − 1 = 0.
Второй: 1 − 0 = 1.
Третий: 0 − 1 = 1 — заняли единицу из следующего разряда.
Четвертый: 0 − 0 = 0 — отдали единицу в предыдущий разряд.

Вычислите:
д) 10111_2-1000_2
е) 10101_2-10_2

Вы удивитесь, но **правила деления двоичных чисел** похожи на деление десятичных. Рисуем привычный «столбик», умножаем, вычитаем, получаем результат.



У нас есть только два варианта: умножить делитель на 1 или на 0. Поэтому алгоритм будет таким:

> Смотрим на делимое, видим, что первые две его цифры — 11.
> Умножаем делитель на 1 и вычитаем из 11 число 10.
> Получили 1, дописываем справа следующую по порядку цифру — 0.
> Теперь 10 равно делителю, значит, тоже умножаем его на 1 и вычитаем.
> Получаем 0. Но у нас ещё остался один 0 у делимого — дописываем его справа от полученного 0.
> Число 0 меньше, чем 10, поэтому умножаем делитель на 0. Получаем конечный ответ — 110.

Вычислите:
ж) $1111_2 \div 11_2$
з) $11001_2 \div 101_2$

ЧАСТЬ 2. ПРЕДСТАВЛЕНИЕ ЦЕЛЫХ ЧИСЕЛ В ЭВМ

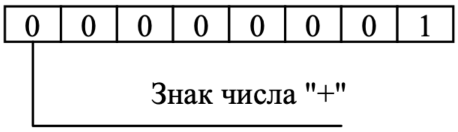
ЭВМ используют двоичный код для представления всех типов данных, как числовых, так и текстовых, графических и других.
Минимальная единица хранения информации – это байт, который состоит из 8 бит. Наибольшую последовательность бит, которую ЭВМ может обрабатывать как единое целое, называют машинным словом. Длина машинного слова зависит от разрядности процессора и может быть равной 16, 32, 64 битам и т.д.
В двоичной системе счисления для записи чисел используются только 0 и 1. Однако, чтобы представить дополнительные сведения, такие как знак числа или положение десятичной точки, применяются специальные соглашения:

- Знак числа: Для обозначения положительного (+) или отрицательного (-) знака числа используется один из битов в двоичном коде (обычно старший бит). Значение 0 может соответствовать плюсу, а 1 - минусу.
- Двоичная точка: В двоичном коде десятичная точка не ставится явно, ее позиция определяется по контексту. Например, для представления дробных чисел, может быть принято, что точка находится между определенными битами.
- Незначащие разряды: В двоичном коде, чтобы заполнить все биты, пустые позиции заполняются нулями. В некоторых случаях для представления знака числа могут использоваться повторяющиеся символы (например, единицы для отрицательных чисел).

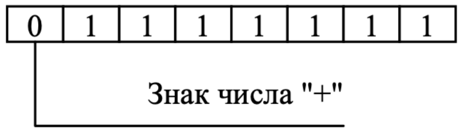
Для целых чисел существуют два представления: беззнаковое (только для неотрицательных целых чисел) и со знаком. Очевидно, что отрицательные числа можно представлять только в знаковом виде. В беззнаковом формате все разряды предназначены для записи значения числа, при представлении чисел со знаком старший (самый левый) бит предназначен для знака. Если старший бит числа равен нулю, то это — положительное число. Если старший бит числа равен единице, то это — отрицательное число. Положительное число представляется его прямым кодом, отрицательное — дополнительным кодом.

Прямой код — старший бит кода равен нулю, остальные биты представляют двоичное представление числа.

Число $1_{10} = 1_2$

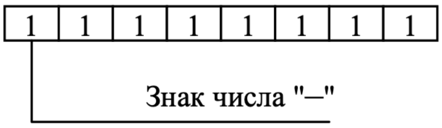


Число $127_{10} = 1111111_2$

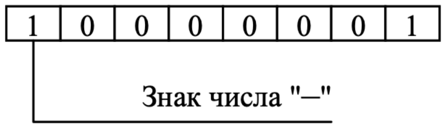


Дополнительный код получается из прямого путем инверсии (замена нулей единицами, а единиц нулями) с последующим добавлением единицы.

Дополнительный код числа –1:



Дополнительный код числа –127:



Вычислите 8-ми битное представление десятичных чисел в ЭМВ. Положительное число представляется его прямым кодом, отрицательное — дополнительным кодом.

пример: 87 = 01010111
пример: -87 = 10101001

а) 45
б) -45
в) 122
г) -122

ЧАСТЬ 3. МАШИННАЯ АРИФМЕТИКА

1. Сложение двух положительных 8-ми битных числа.

Вычислим $11_{10} + 101_{10}$

Переведем в двоичную сс $11 = 1011_2$
$101 = 1100101_2$

Запишем их в прямом коде:
$11 = 00001011_2$
$101 = 01100101_2$

Произведем сложение:

	0	0	0	0	1	0	1	1
+	0	1	1	0	0	1	0	1
	0	1	1	1	0	0	0	0

2. A положительное, B отрицательное и по абсолютной величине больше, чем A.

Вычислим $3_{10} + -10_{10}$

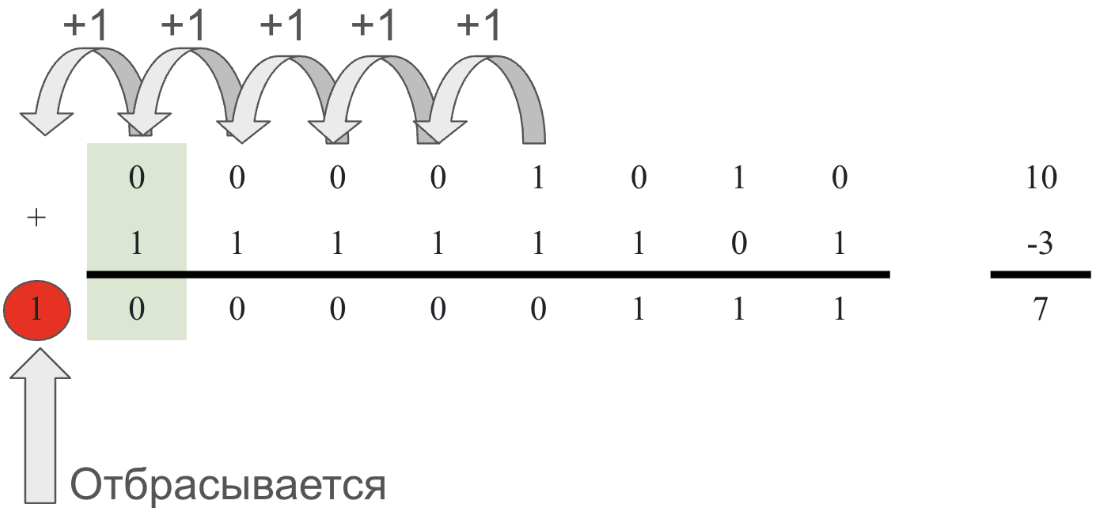
Произведем сложение:

	0	0	0	0	0	0	1	1	3
+	1	1	1	1	0	1	1	0	-10
	1	1	1	1	1	0	0	1	-7

Получен правильный результат в дополнительном коде. При переводе в прямой код биты цифровой части результата инвертируются и к младшему разряду прибавляется единица.

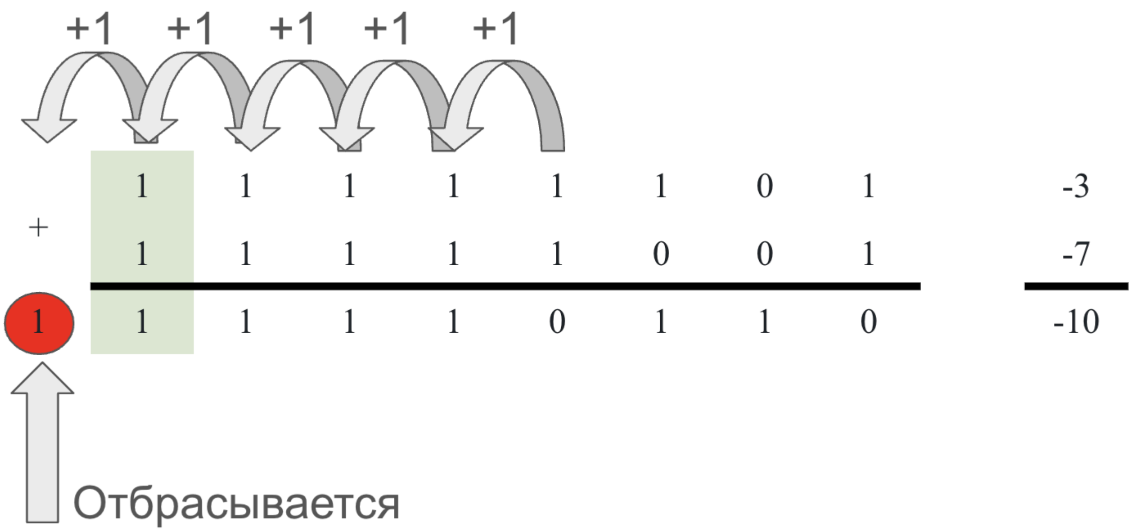
3. A положительное, B отрицательное и по абсолютной величине меньше, чем A

Вычислим $10_{10} + -3_{10}$
Произведем сложение:



Получен правильный результат. Единицу переноса из знакового разряда компьютер отбрасывает.

4. A и B отрицательные.
Вычислим $-3_{10} + -7_{10}$
Произведем сложение:



Получен правильный результат в дополнительном коде. Единицу переноса из знакового разряда компьютер отбрасывает.

Переполнение разрядной сетки

В приведенных примерах соблюдалось условие попадания суммы чисел А и В в интервал допустимых значений. Это условие, как было отмечено выше, всегда соблюдается при сложении двух чисел различного знака. При сложении двух чисел одинакового знака возможно нарушение указанного условия.
Рассмотрим следующий пример. Пусть заданы целые числа 88_{10} и 50_{10}. Суммируя коды заданных чисел, имеем:

	0	1	0	1	1	0	0	0		88
+	0	0	1	1	0	0	1	0		50
	1	0	0	0	1	0	1	0	!=	138

Полученный результат не совпадает с кодом суммы чисел А и В, так как имеет знак (отрицательный в данном случае) отличный от знака просуммированных чисел. В таких ситуациях говорят что произошло переполнение разрядной сетки. В приведенном примере А+В=138, что превышает максимальное значение равное 127.
Такая же ситуация может случиться и с отрицательными числами.

Ответы для данного практикума

ЧАСТЬ 1							
а	б	в	г	д	е	ж	з
10010	10010	1111	110111	1111	10011	101	101

ЧАСТЬ 2			
а	б	в	г
00101101	11010011	01111010	10000110