ОГЛАВЛЕНИЕ

Введение

За прошедшие десятилетия электронно-вычислительные машины, или по-другому, что наиболее привычно для уха современного человека – компьютеры, претерпели серьезные изменения в сторону уменьшения своих размеров, увеличения производительности и функциональности в работе, охвате все более широкого круга, решаемых задач. Однако основы построения компьютеров – математические алгоритмы, способы обработки сигналов, элементная база и многое другое не сильно изменились за прошедшее время. Данное учебное пособие посвящено изучению, как основ построения вычислительной техники, так и освещению ближайших перспектив связанных с дальнейшим прогрессом в области компьютеростроения.

Данное учебное пособие предназначено для студентов очной и заочной форм обучения по направлению Прикладная информатика, изучающих предмет Основы компьютерной электроники.

Цель данного учебного пособия - сформировать у студентов представление не только об отдельных элементах, технологиях и математических методах используемых при создании компьютеров, но и дать четкое представление об их тесной и неразрывной взаимосвязи. Поэтому в учебном пособии рассматривается большой спектр материала, связанного с компьютером, предложенный в сжатом виде, но вполне достаточный для понимания роли каждого из рассматриваемых объектов, в системе построения вычислительной цифровой техники.

Особое внимание уделяется вопросам, связанным с будущим развитием компьютерной электроники, проблемам, которые стоят перед ней в настоящий момент, и перспективам по их разрешению.

Данное учебное пособие можно рассматривать и как план для более углубленного изучения, каждой из освещаемых тем, касающихся основ компьютерной электроники.

В конце каждой главы предлагаются вопросы для самопроверки, призванные закрепить знания обучаемого и еще раз подчеркнуть наиболее важные моменты пройденного материала.

Учебное пособие может быть полезно не только студентам, но и школьникам, проходящим обучение в профильных классах информационной направленности.

Глава 1. Понятие о сигнале

Составной частью всех *электронно-вычислительных машин* (ЭВМ), различных модулей автоматизированных систем управления являются *цифровые устройства,* которые выполняют обработку, хранение и передачу *информации*.

Информация - это отражение окружающего нас реального мира. Более специфичное определение информации дает наука *информатика* - совокупность фактов, явлений и событий, представляющих интерес и подлежащих регистрации и обработке.

Восприятие информации происходит посредством ее носителей - речи, текста, цифр и т.п.; которые сами по себе не являются информацией, а служат лишь элементами для ее переноса.

В электронике информацию, воплощенную и зафиксированную в некоторой материальной форме, называют *сообщением* и передают с помощью *сигналов*.

При передаче *сообщений* используют различные физические процессы (электрический ток, световой поток и др.), которые могут существовать сами по себе или использоваться для других целей, как, например, для передачи энергии. И лишь в случае, когда какая-либо физическая величина этих процессов несет в себе информацию, говорят, что такой процесс является *сигналом*. Именно в этом смысле используются понятия: *электрический сигнал*, *световой сигнал* и т.д. Итак, электрический сигнал - не просто электрический ток, а ток, *величина* которого несет в себе определенную информацию.

1.1. Виды сигналов

Среди множества сигналов можно выделить два основных типа, используемых для передачи, обработки и хранения информации - это *аналоговый* и *дискретный сигналы* (рисунок 1.1).

Аналоговый или непрерывный сигнал представляет собой определенный для любого момента времени и по амплитуде процесс, а поскольку порождающие такие сигналы физические процессы, обычно сами являются непрерывными, этим объясняется, почему сигналы такого типа и называются еще *аналоговыми,* т.е. аналогичные порождающим их процессам.

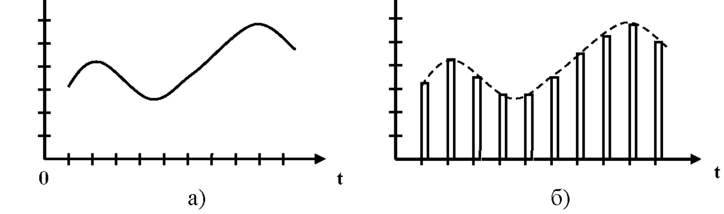


Рисунок 1.1 - Виды сигналов: аналоговый (а) и дискретный (б) сигналы.

Аналоговый сигнал всегда является функцией времени $x(t)$. Причем эта функция может принимать любые вещественные значения в диапазоне изменения аргумента t.

Дискретным называют сигнал, определенный только в отдельные (дискретные) моменты времени, например, через одну миллисекунду. Каждое значение дискретного сигнала может быть представлено числом любой приемлемой системы счисления. В цифровых системах представление дискретных значений сигнала числом, называют *кодированием*.

1.2. Преобразование аналоговых сигналов в цифровые сигналы

Любой аналоговый сигнал можно привести к дискретной, а затем, после кодирования к цифровой форме. Это широко используется в компьютерной электронике, которая построена на использовании в основном цифровых устройств и поэтому оперирует дискретными значениями. Помимо этого информацию в цифровом виде легче хранить, а также передавать практически без потерь по современным линиям связи.

Преобразование аналогового сигнала в дискретный состоит из двух этапов: *дискретизации по времени* и *квантовании по амплитуде* (рисунок 1.2).

Дискретизация по времени означает, что сигнал представляется рядом своих отсчётов (дискретов) непрерывных по амплитуде и взятых через равные промежутки времени $\Delta t = t_n - t_{n-1}$ (хотя в некоторых специальных случаях может применяться и неравномерная по времени дискретизация, например при оцифровке узкополосных сигналов).

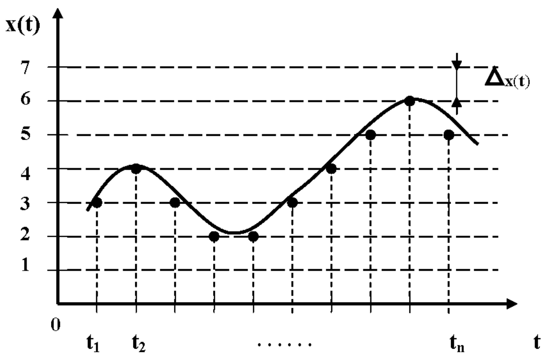


Рисунок 1.2. - Процесс преобразования аналогового сигнала в цифровой.

При *квантовании по амплитуде* происходит замена возможных значений сигнала $x_1, x_2 \ldots x_n$, когда каждому x(t) сопоставляется ближайшее число из набора фиксированных величин, называемых *уровнями квантования*.

По сути, процесс квантования это та же дискретизация, поскольку шкала квантования состоит из дискретных отсчетов, и значения присваиваются не непрерывно, а с интервалом, т.е. дискретно. Тем не менее, в практику вошло называть этот процесс - *квантованием*. Шаг квантования определяют как:

$\Delta_{x(t)} = \mathbf{x(n\,\Delta\,t)/\,K_n}$, где

n – *количество отсчетов за единицу времени*;

Δ **t** – *период времени между двумя отсчетами* ($\Delta t = t_n - t_{n-1}$);

$\mathbf{K_n}$ - *десятичный эквивалент количества шагов квантования.*

На рисунке 1.3 представлены два варианта преобразование одного и того же аналогового сигнала в дискретный. Не трудно заметить, что вариант на рисунке 1.3, *б* предпочтительней, так как цифровой сигнал более точно описывает изначальный аналоговый. Это произошло благодаря тому, что период времени между двумя отсчетами на рисунке 1.3, *б* меньше, чем на рисунке 1.3, *а*: $\Delta t_б < \Delta t_а$, другими словами частота дискретизации по времени (обратная величина периоду времени Δt) во втором случае была задана выше, чем в первом.

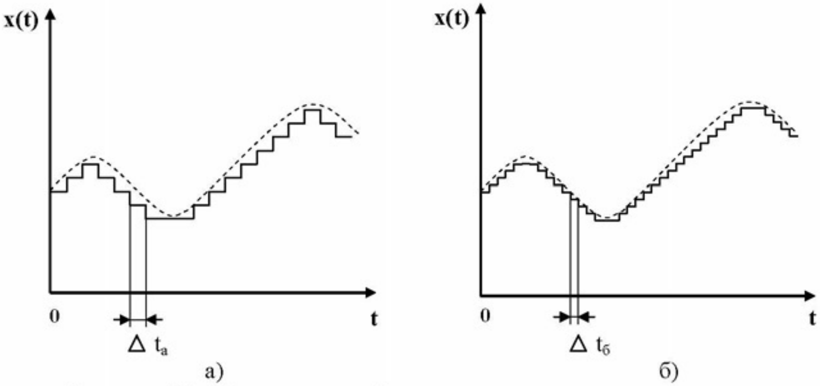


Рисунок 1.3. -Варианты преобразования одного и того же аналогового сигнала в цифровой. Вариант *б* точнее, т.к. $\Delta t_б < \Delta t_а$.

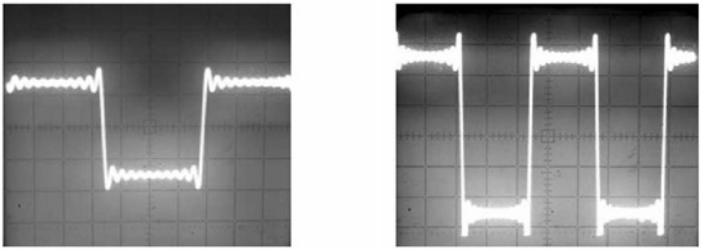
Возникает закономерный вопрос: какой должна быть оптимальная частота дискретизации по времени? Ответ на него дал Гарри Найквист (1889-1976), американский физик-электрик и изобретатель, в статье "Certain Topics in Telegraph Transmission Theory" ("Некоторые вопросы теории телеграфной передачи") в 1928 году, в которой он изложил принципы осуществления выборки непрерывных сигналов для преобразования их в цифровой вид.

Спустя 5 лет тот же самый результат независимо от американского коллеги был получен в СССР В. А. Котельниковым, который изложил результаты своих изысканий в работе “О пропускной способности “эфира” и проволоки в электросвязи” в 1933 году. Поэтому в России соответствующие положения чаще называют теоремой Котельникова.

Согласно предложенной теореме, чтобы аналоговый (непрерывный) сигнал можно было абсолютно точно восстановить по его отсчётам, частота дискретизации должна быть в два раза выше *максимальной* частоты сигнала: $f_{д} = 2f_{c\,max}$ (Гц), или, отсчёты сигнала должны браться не реже чем через : $\Delta t = 1/(2f_{c\,max})$ секунды.

1.3. Элементы цифрового сигнала

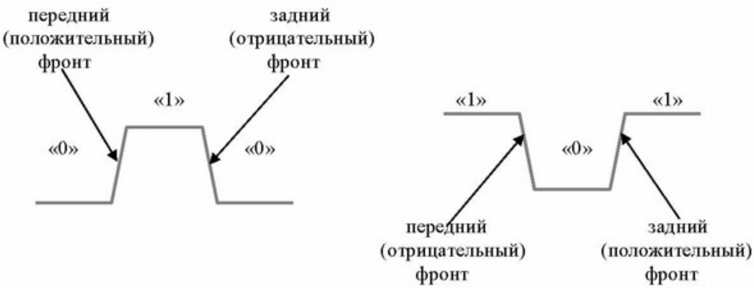
Как выглядит цифровой сигнал на экране осциллографа (прибор для изучения параметров электрических сигналов непосредственно на экране) можно увидеть на рисунке 1.4.



а) б)

Рисунок 1.4. – Цифровой сигнал на экране осциллографа: а – одиночный, б – множественный.

Необходимо уяснить, что при всей внешней простоте формы цифрового сигнала - его отдельные элементы играют важное значение при проектировании электронной техники. На рисунке 1.5 показаны положительный и отрицательный сигналы, имеющие прямопротивоположные пассивный и активный уровни (например, у положительного сигнала пассивным является уровень соответствующий логическому «0», а активный – «1»), а также обладающие передним и задним фронтами.



а) б)

Рисунок 1.5. - Элементы положительного (а) и отрицательного (б) цифровых сигналов.

Цифровые устройства могут работать с дискретными сигналами как в статическом режиме (когда на входе устройства уже присутствует сигнал определенного уровня), так и в динамическом режиме, когда уровень сигнала меняется, как это показано на рисунке 1.5. В последнем случае возможны три варианта срабатывания цифрового устройства (рассмотрим их на примере положительного цифрового сигнала, представленного на рисунке 1.5, а):

- при переходе из нулевого уровня сигнала в единичный (или говорят: по переднему фронту);
- при переходе из единичного уровня сигнала в нулевой (по заднему фронту);
- поочередно – вначале по переднему, а затем по заднему фронту.

Так, современная оперативная память компьютера за один тактовый импульс, срабатывает по двум фронтам и успевает выполнить сразу две операции, например, чтения информации или ее записи.

Подводя итог первой главе, нужно еще раз уделить внимание преимуществам цифрового сигнала перед аналоговым, поскольку именно благодаря этими преимуществами существует современная тенденция перехода различных информационных каналов на цифровой формат:

- информацию в цифровом виде можно длительно хранить, причем без потерь;
- информацию в цифровом виде можно многократно копировать (перезаписывать) без искажений;
- цифровые сигналы – это качественная и скоростная передача информации на большие расстояния;
- устройства, работающие с цифровыми сигналами проще проектировать (всегда можно точно рассчитать и предсказать их «поведение»);
- устройства, работающие с цифровыми сигналами легче тестировать и ремонтировать, нежели чем аналоговые устройства.

Вопросы для самопроверки:

1. Что такое *информация*?
2. Дайте определение *сообщению*.
3. Что подразумевают под понятием *сигнала*?
4. Какие виды сигналов существуют и чем отличаются друг от друга?
5. Как в цифровых системах называют представление дискретных значений сигнала числом?
6. Из каких этапов состоит преобразование аналогового сигнала в цифровой?
7. Как определяют шаг квантования?
8. О чем гласит теорема Найквиста-Котельникова?
9. Какие основные элементы цифрового сигнала Вы знаете?
10. Какое значение имеет срабатывание цифрового устройства одновременно по двум фронтам управляющего сигала?
11. Какими преимуществами обладает цифровой сигнал перед аналоговым сигналом?

Глава 2. Системы счисления

2.1. Кодирование

Поскольку дискретность - это случай, когда объект или явление имеет конечное (счетное) число вариаций, то, для того чтобы выделить конкретное из всего возможного, нужно каждому конкретному дать оригинальное имя (другими словами перечислить). Эти имена и будут нести в себе информацию об объектах, явлениях и т. п. В качестве имен часто используют целые числа 0, 1, 2…, т.е. применяют *кодирование*. Именно такая *цифровая* форма представления информации используется в ЭВМ.

Код, в котором использованы специальные символы для обозначения количества каких либо объектов называют *системой счисления*.

Количество символов в системе счисления носит название его *основания*.

Например, самая известная десятичная система имеет символы 0,1,2,3,4,5,6,7,8 и 9 всего их 10, поэтому её называют *системой счисления с основанием 10*. Двоичная система счисления имеет только 2 символа 0 и 1, поэтому её называют *системой счисления с основанием 2*. В шестнадцатеричной системе используется 16 символов: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F и т.д.

Чем меньше основание системы счисления, тем больше разрядов требуется для представления одного и того же количества объектов, как это видно из таблицы 2.1.

Таблица 2.1. - Сопоставление чисел разных систем счисления

Десятичное число	0	1	2	3	4	5	6	7	8	9
Двоичное число	0	1	10	11	100	101	110	111	1000	1001

Выше приведенный пример также служит иллюстрацией к тому, что символы одной системы счисления могут быть представлены символами другой системы счисления.

2.2. Перевод из одной системы счисления в другую

Рассмотрим число 758 в *десятичной системе* - его записывают еще так: 758_{10}. В этом числе:

Цифра 7 обозначает 700, так как она занимают 3 разряд слева от десятичной точки,

Цифра 5 обозначает 50, так как она занимает второй разряд от десятичной точки,

Цифра 8 представляет число 8, поскольку она находится в первом разряде слева от десятичной точки, таким образом, всё число есть сумма:

$758=700+50+8=7\cdot 10^2+5\cdot 10^1+8\cdot 10^0$.

Рассмотренный пример иллюстрирует понятие *веса разряда*.

Аналогичное понятие вес разряда используется и в других системах счисления, например в двоичной, так для числа $1011_2=2^3\cdot 1+2^2\cdot 0+2^1\cdot 1+2^0\cdot 1$

Вес следующего разряда всегда равен весу предыдущего разряда умноженному на основание системы счисления.

Учитывая это правило, запишем веса десяти первых разрядов двоичной системы счисления:

Таблица 2.2. - Веса десяти первых разрядов двоичной системы счисления.

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
512	256	128	64	32	16	8	4	2	1

Используя понятие *весов* легко преобразовывать числа одной системы счисления в числа другой системы счисления, например двоичное число **110001** необходимо преобразовать в десятичное:

Искомое десятичное число будет равно сумме произведений соответствующих разрядов двоичного числа и их *весов* из таблицы 2.3: $110001=32\cdot 1+16\cdot 1+8\cdot 0+4\cdot 0+2\cdot 0+1\cdot 1=49_{10}$.

Таблица 2.3. - Веса десяти первых разрядов двоичной системы счисления и заданное двоичное число.

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
512	256	128	64	32	16	8	4	2	1
				1	**1**	**0**	**0**	**0**	**1**

Рассмотрим обратный случай – преобразование числа десятичной системы счисления в число двоичной системы счисления.

Например, нужно перевести в двоичную систему счисления число 11_{10}. Это можно осуществить делением числа 11_{10}, на основание системы счисления в которую переходим, в целых числах с выписыванием остатков деления, по следующей схеме:

11:2 = 5 остаток **1** это разряд весом 1
5 : 2 = 2 остаток **1** это разряд весом 2
2 : 2 = 1 остаток **0** это разряд весом 4
1 : 2 = 0 остаток **1** это разряд весом 8

Процесс перехода заканчивается в тот момент, когда очередной результат деления даст ноль (0) целых.

Помня о том, что самый младший разряд всегда занимает крайнее правое место в записанном числе в любой системе счисления, записываем результат:

$11_{10} = 1011_2$.

Таким образом, *остатки, от деления, выписанные в соответствии с весами разрядов, дают искомое число.*

Легко проверить полученный результат по таблице 2.4:

Таблица 2.4. - Веса десяти первых разрядов двоичной системы счисления и искомое двоичное число.

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
512	256	128	64	32	16	8	4	2	1
						1	**0**	**1**	**1**

Проверяем, подстановкой из таблицы 2.4: $2^3+2^1+2^0+ = 8+2+1 = 11_{10}$.

Если говорить о *шестнадцатеричной* системе счисления, то согласно определению, в ее составе должно быть 16 различных символов, это - 0,1,2,3,4,5,6,7,8,9,A, B, C, D, E, F. Здесь буква А обозначает число 10, В обозначает число 11 и т.д.

Таблица 2.5 - Сопоставление чисел разных систем счисления.

Десятичное число	Двоичное число	Шестнадцатеричное число
0	0000	**0**
1	0001	**1**
2	0010	**2**
3	0011	**3**
4	0100	**4**
5	0101	**5**
6	0110	**6**
7	0111	7
8	1000	**8**
9	1001	**9**
10	1010	**A**
11	1011	**B**
12	1100	**C**
13	1101	**D**
14	1110	**E**
15	1111	**F**

Преимущество шестнадцатеричной системы в отличие от десятичной состоит в том, что она позволяет реализовать переход от шестнадцатеричной к двоичной системе счисления достаточно просто, используя *тетрады* («tetra» в переводе с греческого означает четыре) двоичных символов (таблица 2.5.).

2.3. Арифметические операции над двоичными числами

При выполнении различных операций в современных ЭВМ, базирующихся на цифровых (дискретных) устройствах, информация обычно представляется (кодируется) числами двоичной системы счисления. Это связано с тем, что можно использовать электронные устройства всего с двумя электрическими состояниями (“1” и “0”), что значительно упрощает как изготовление самих устройств, так и представление информации в цифровом виде.

Общепринято, что символ “1” представляется некоторым стандартным уровнем напряжения или тока, а “0” - нулевым или близким к нулю уровнем напряжения или тока (так, например, для ТТЛ-логики *логической единице* соответствует напряжение в 4,5 вольта, а *логическому нулю* соответствует напряжение в 0,8 вольт).

Сложение двоичных чисел выглядит следующим образом:

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 10_2$ - т.к. единица получившаяся в результате сложения двух единиц младших разрядов переносится в старший разряд, в ответе показаны уже два разряда двоичного числа, а не цифра 10_{10} десятичной системы счисления. Еще один пример с переносом единицы в старший разряд:

$1 + 1 + 1 = 11_2$

Теперь рассмотрим более сложные примеры:

$$\begin{array}{r} +\ 1000011 = 67_{10} \\ 0101010 = 42_{10} \\ \hline 1101101 = 109_{10} \end{array} \qquad \text{или} \qquad \begin{array}{r} +\ 0111101 = 61_{10} \\ 0110101 = 53_{10} \\ \hline 1110010 = 114_{10} \end{array}$$

В двоичной системе счисления для представления знака числа используется дополнительный знаковый разряд (иногда несколько разрядов), который располагается перед старшим числовым разрядом. Для положительных чисел значение знакового разряда равно 0 (*нулю)*, для отрицательного - 1 (*единице)*.

Двоичный код со знаком называют также *прямым кодом*. В качестве примера рассмотрим положительное и отрицательное числа, десятичный эквивалент которых равен 153.

$\boxed{0}10011001$ – код положительного числа 153_{10};
$\boxed{1}10011001$ – код отрицательного числа - 153_{10}.

Обратный код получается путем замены (*инвертирования*) всех “0” на “1” и всех “1” на “0” прямого кода. Причем, знаковый разряд при этом остается неизменным.

$\boxed{0}10011001$ – код положительного числа 153_{10};
$\boxed{0}01100110$ – обратный код положительного числа 153_{10}

Обратный код, *дополненный единицей в младшем разряде*, называется *дополнительным кодом*. Так для рассмотренного числа $\boxed{0}10011001$, дополнительный код будет выглядеть как: $\boxed{0}01100111$.

Вычитание двоичных чисел происходит так:

$10_2 - 1 = 1$
$1 - 0 = 1$
$1 - 1 = 0$
$0 - 0 = 0$

Проверим первую строку выше записанного:

$10_2 - 1 = 1$

Для этого переведём левую и правую части этого равенства в десятичную систему:

$1 * 2^1 + 0 * 2^0 - 1 * 2^0 = 1 * 2^0$

получаем в десятичной системе:

$2 + 0 - 1 = 1$

Более сложный пример: вычтем из числа 100_2 число 10_2. Вычитать будем «столбиком», как и в десятичной системе счисления:

$$\begin{array}{r} 1\,0\,0 \\ \underline{-\ 1\,0} \\ 1\,0 \end{array}$$

Вначале из правого нуля в 100_2 вычли ноль, а затем, чтоб из среднего нуля в 100_2 вычесть 1, занимают 1 из третьей позиции. Всё как в десятичной системе, но только с 0 и 1.

Еще один пример: вычтем из числа 1001_2 число 10_2:

$$\begin{array}{r} 1\,0\,0\,1 \\ \underline{-\quad 1\,0} \\ 1\,1\,1 \end{array}$$

Следующий пример: вычтем из числа 1000_2 число 1_2:

$$\begin{array}{r} 1\,0\,0\,0 \\ -\qquad 1 \\ \hline 1\,1\,1 \end{array}$$

Операция вычитания в цифровых системах нередко реализуется с помощью операции сложения. Вычитаемое при этом представляется в дополнительном коде (если расчет не требует высокой точности - в обратном коде).

Например, из числа 67_{10} необходимо вычесть число $42_{10.}$ Вначале вычитающее представляется в дополнительном коде со знаком «минус»:

$- 42_{10} = \boxed{1}0101010$ --> $\boxed{1}1010101$ --> $\boxed{1}1010110$

Затем выполняется операция сложения:

$$\begin{array}{ll} + \boxed{0}1000011 & = 67_{10} \\ \boxed{1}1010110 & = -42_{10} \text{ (в дополнительном коде)} \\ \hline \boxed{0}0011001 & = 25_{10} \end{array}$$

Теперь от числа 42_{10} отнимем число $67_{10.}$ Представим вычитающее в дополнительном коде со знаком «минус»:

$- 67_{10} = \boxed{1}1000011$ --> $\boxed{1}0111100$ --> $\boxed{1}0111101$

Затем выполняется операция сложения:

$$\begin{array}{ll} + \boxed{0}0101010 & = 42_{10} \\ \boxed{1}0111101 & = -67_{10} \text{ (в дополнительном коде)} \\ \hline \end{array}$$

$\boxed{1}1100111$ **-** поскольку результат отрицательное число его необходимо перевести в прямой код:

$\boxed{1}0011000$ - инвертируем, и после добавления единицы получаем:

$\boxed{1}0011001 = - 25_{10}$

Перейдем к умножению и делению двоичных чисел. Для умножения двоичных чисел действуют правила, которые полностью совпадают с аналогичными, применяемыми к десятичным числам: при умножении на нуль получается нуль; при умножении единицы на единицу получается единица.

Допустим, необходимо перемножить двоичные числа 1001 и 101. Решение будет выглядеть следующим образом:

× 1 0 0 1	(первый множитель)
1 0 1	(второй множитель)
1 0 0 1	(результат умножения единицы числа 101 на 1011)
1 0 0 1	(результат умножения единицы числа 101 на 1011)
1 0 1 1 0 1	(произведение)

Проверим вычисления:

$1001_2 = 9_{10}$,
$101_2 = 5_{10}$,
$9_{10} \cdot 5_{10} = 45_{10} = 101101_2$

При записи чисел "в столбик" необходимо, как и с десятичными, выровнять их так, чтобы крайние правые ненулевые цифры оказались друг под другом. Например, необходимо найти произведение чисел 110_2 и 101000_2.

× 1 1 0	(первый множитель)
1 0 1 0 0	(второй множитель)
+ 1 1	(результат умножения единицы числа 101000 на 11)
1 1	(результат умножения единицы числа 101000 на 11)
1 1 1 1 0 0 0	(произведение)

Проверим вычисления:

$110_2 = 6_{10}$,
$10100_2 = 20_{10}$,
$6_{10} \cdot 20_{10} = 120_{10} = 1111000_2$

При умножении дробных чисел необходимо вычислить результат, не обращая внимания на двоичные запятые, и затем в произведении определить столько знаков после запятой, сколько их имеется в обоих числах вместе. После этого можно удалить крайние правые нули дробной части, если они есть.

Прежде чем показать, как перемножаются дробные числа, небольшое отступление, связанное с тем как осуществляется перевод простой десятичной дроби в двоичную систему счисления.

При переводе правильной десятичной дроби в систему счисления с основанием 2 необходимо сначала саму дробь, а затем дробные части всех последующих произведений последовательно умножать на 2, отделяя после каждого умножения целую часть произведения. Число в новой системе счисления записывается как последовательность полученных целых частей произведения. Умножение производится до тех пор, пока дробная часть произведения не станет равной нулю. Это значит, что сделан точный перевод. В противном случае перевод осуществляется до заданной точности.

Допустим необходимо перевести число 0,35 из десятичной системы в двоичную.

$$
\begin{array}{r|l}
0, & 35 \\
\times & \;\;2 \\
\hline
0 & 70 \\
\times & \;\;2 \\
\hline
1 & 40 \\
\times & \;\;2 \\
\hline
0 & 80 \\
\times & \;\;2 \\
\hline
1 & 60 \\
\times & \;\;2 \\
\hline
1 & 20
\end{array}
$$

Итак, получаем: $0{,}35_{10} = 0{,}01011_2$

Обратный процесс преобразования представлен ниже:

$0,\,01011_2 = 2^{-1}\cdot 0 + 2^{-2}\cdot 1 + 2^{-3}\cdot 0 + 2^{-4}\cdot 1 + 2^{-5}\cdot 1 = 0 + 0{,}25 + 0 + 0{,}0625 + 0{,}03125 = 0{,}34375$

Правило округления двоичных чисел до целого значения состоит в следующем: если в старшем разряде дробной части числа имеется нуль, то дробная часть этого числа просто отсекается, а если единица, то к целой части числа следует прибавить единицу. Аналогично производится округление с другой точностью.

Чтобы выполнить умножение дробных двоичных чисел, например 11,1 и 10,101, запишем числа друг под другом, выровненные по правому краю и без двоичных запятых:

$$
\begin{array}{rl}
\times\quad 111 & \text{(первый множитель)} \\
10101 & \text{(второй множитель)} \\
+\qquad\quad & \\
111 & \text{(результат умножения единицы числа 10011 на 111)} \\
111\;\;\;\; & \text{(результат умножения единицы числа 10011 на 111)} \\
111\;\;\;\;\;\;\;\; & \text{(результат умножения единицы числа 10011 на 111)} \\
\hline
10010011 & \text{(произведение)}
\end{array}
$$

Поскольку в двух исходных числах справа от запятых находятся в совокупности 4 знака, то и в произведении нужно отделить 4 знака после двоичной запятой. Таким образом, реальным результатом умножения двух исходных двоичных чисел является

$11{,}1 \cdot 10{,}101 = 1001{,}0011$

Проверим вычисления:

$11{,}1_2 = 3{,}5_{10}$,

$10{,}101_2 = 2{,}625_{10}$,

$3{,}5_{10} \cdot 2{,}625_{10} = 9{,}1875_{10} = 1001{,}0011_2$

Деление двоичных чисел, как и десятичных, выполняется при записи чисел "углом". Для выполнения действия необходимо в делимом выбрать первую часть числа, которая совпадает с делителем по количеству знаков, — если число, образованное этими знаками, не меньше делителя. В противном случае выбирается такая первая часть числа, в которой знаков на один больше, чем в делителе.

В обоих случаях первая цифра частного равна единице, но не тогда, когда делимое меньше делителя: лишь в этом случае первой цифрой частного будет 0, и это означает, что частное содержит нуль целых. Далее деление производится так же, как и в десятичной системе. Не стоит забывать, что цифры частного — это лишь 1 или 0.

Ниже представлен пример обычного деления двоичных чисел без остатка. Допустим, необходимо разделить число 1110_2 на 111_2.

$$
\begin{array}{r|l}
-\,1110 & 111 \\ \cline{2-2}
111 & 10 \\ \cline{1-1}
0 &
\end{array}
$$

Проверим вычисления:

$1110_2 = 14_{10}$,

$111_2 = 7_{10}$,

$10_2 = 2_{10}$,

$14_{10} : 7_{10} = 2_{10} = 10_2$

Вот другой пример: найти частное от деления двоичных чисел 1001011_2 на 101_2.

```
– 1 0 0 1 0 1 1  | 1 0 1
  1 0 1          |-------
 – 1 0 0 0       | 1 1 1 1
     1 0 1
    – 1 1 1
      1 0 1
     – 1 0 1
       1 0 1
           0
```

Проверим вычисления:

$1001011_2 = 75_{10}$,

$101_2 = 5_{10}$,

$1111_2 = 15_{10}$,

$75_{10} : 5_{10} = 15_{10} = 1111_2$

Таким образом, на основании рассмотренных примеров было показано, как выполняются арифметические операции над двоичными числами.

2.4. Единицы измерения компьютерной информации

Как уже упоминалось, в современных ЭВМ используется двоичная система счисления, поэтому всего двумя цифрами (“1” или “0”) может быть представлена минимальная единица компьютерной информации, название которой и происходит от английского:

binary digi**t** (*двоичное число*) – сокращенно **bit** (*бит*).

Однако наиболее часто употребляется другая единица измерения информации – **byte** (*байт*):

1 *байт* = 8 *бит*

Сколько различных значений может принимать один *байт* информации? Учитывая, что речь идет о системе счисления с основанием два, и, принимая во внимание, что байт состоит из восьми бит, получаем - 2^8 или 256 целых чисел. Этого вполне достаточно чтобы дать уникальное 8-ми битовое кодовое обозначение каждой заглавной и строчной букве любого алфавита, всем цифрам, знакам препинания и другим необходимым символам. Такой способ кодирования (с помощью 256 знаков) применен в ASCII (American Standard Coding for Information Interchange).

К битам и байтам применяются приставки кило-, мега-, гига- и т.д.. В соответствии с международными соглашениями по системам измерения (СИ) они означают соответственно приставки 10^3, 10^6, 10^9 и т.д. Однако, учитывая, что речь идет о двоичной системе счисления, то, например 1 килобайт, не будет равен 1000 байт, как это было бы в десятичной системе счисления.

Вот как выглядят более крупные единицы измерения компьютерной информации:

1 килобайт (Кб) = 2^{10} байт (б) = 1024 байт (б);
1 мегабайт (Мб) = 1024 килобайт (Кб) = 2^{20} байт (б) = 1 048 576 байт (б);
1 гигабайт (Гб) = 1024 мегабайт (Мб) = 2^{30} байт (б) =
= 1 073 741 824 байт (б);
1 терабайт (Тб) = 1024 гигабайт (Гб) = 2^{40} байт (б) =
= 1 099 511 627 776 байт (б).

Считается, что 1 байт, как правило, соответствует одному печатному знаку (букве, цифре, символу). Стандартная страница машинописного текста с двумя интервалами между строк содержит приблизительно 2000 байт, или два килобайта – 2 Кб.

Обычный лазерный DVD-диск вместимостью 4.7 Гб позволит записать порядка 2.5 миллионов страниц или 5 тысяч томов книг объемом по 500 страниц каждый.

В о п р о с ы   д л я   с а м о п р о в е р к и:

1. Что называют *системой счисления,* ее *основанием*?
2. Какая связь между дискретными (цифровыми) *сигналами*, используемым в компьютерной электронике, и *системами счисления*?
3. Как вы понимаете выражение – *вес разряда*?
4. Для чего используют *прямой код*?
5. Для чего используют *дополнительный код*?
6. Какие системы счисления используют в *ЭВМ*?
7. Что такое *bit*?
8. Сколько бит в одном *байте*?
9. Сколько значений может принимать один *byte*?
10. Чему равен один *килобайт* и почему?

Глава 3. Булева алгебра

3.1. Понятия алгебры логики

Основоположником алгебры логики является великий немецкий математик Готфрид Вильгельм Лейбниц (1646 - 1716 гг.). Он сделал попытку построить универсальный язык, с помощью которого споры между людьми можно было бы разрешать посредством вычислений. На заложенном Лейбницем фундаменте ирландский математик Джордж Буль (1815-1864 гг.) в середине XIX века построил здание новой науки - алгебры логики, - которая в отличие от обычной алгебры оперирует не числами, а *высказываниями.*

Высказывание - *это любое утверждение, относительно которого можно сказать истинно оно или ложно*, т.е. соответствует оно действительности или нет.

Буль произвел революцию в науке, о которой сам не подозревал. Его разработки много позже, чем они появились, были взяты за основу построения электронно-вычислительных устройств. Поэтому алгебру логики также называют *булевой алгеброй.* В самом деле, поскольку в булевой алгебре используются всего два значения – *истина* и *ложь*, то, присвоив им код “1” и “0” соответственно, получили отличный математический инструмент синтеза и анализа различных цифровых устройств.

3.2. Основные функции

Алгебра логики базируется на *трёх* функциях, определяющих *три* основные логические операции. Реализуют функции алгебры логики с помощью *логических элементов* (ЛЭ), которые используются для построения преобразователей цифровых сигналов *комбинационного* типа. В комбинационных устройствах отсутствует внутренняя память. Сигналы на их выходах в любой момент однозначно определяются сочетаниями сигналов на входах и не зависят от предыдущих состояний схемы. Характерной особенностью комбинационных устройств является отсутствие петель обратной связи.

1. *Функция логического умножения* (**конъюнкции**). Функция логического умножения записывается в виде: $f = X_1 \cdot X_2$ (символы логического умножения: «&», «Λ», «·», «×»), и читается как: f есть (эквивалентна) X_1 *и* X_2, поскольку функция истинна тогда, когда истинны 1-й *и* 2-й аргументы (переменные). *Конъюнкцию* называют функцией **И**, а *логический элемент*, реализующий эту функцию, *элементом* **И** (рисунок 3.1).

Необходимо отметить, что входов у логического элемента может быть и один, а может быть и несколько, поэтому при обозначении ЛЭ всегда указывается их количество, т.е. в представленном примере (показано на рисунке 3.1) элемент, строго говоря, должен называться **2И**.

Количество переменных (аргументов), участвующих в одной *конъюнкции*, соответствует количеству входов элемента **И**.

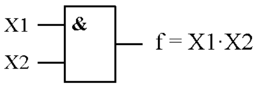


Таблица истинности элемента **И**

X1	X2	f
0	0	0
1	0	0
0	1	0
1	1	1

Рисунок 3.1 – Так изображается на функциональных и принципиальных схемах элемент **И.**

2. *Функция логического сложения* (**дизъюнкция**). Функция логического сложения записывается в виде: f = X1 + X2 (символы логического сложения «+», «V»), и читается как: f есть X1 *или* X2, поскольку функция истинна, когда истинна одна *или* другая переменная (хотя бы одна), поэтому функцию дизъюнкции часто называют функцией **ИЛИ** (рисунок 3.2).

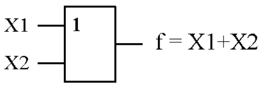


Таблица истинности элемента **ИЛИ**

X1	X2	f
0	0	0
1	0	1
0	1	1
1	1	1

Рисунок 3.2 – Так изображается на функциональных и принципиальных схемах элемент **ИЛИ**

3. *Функция отрицания* (**инверсии**). Записывается в виде: $f = \overline{X}$, и читается, как f есть (эквивалентна) *не* X. Элемент, реализующий функцию **НЕ,** называется *инвертором* (рисунок 3.3).

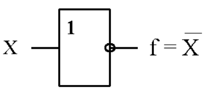


Таблица истинности элемента **НЕ**

X	f
0	1
1	0

Рисунок 3.3 – Так изображается на функциональных и принципиальных схемах элемент **НЕ**

Существуют разные *способы задания* функций алгебры логики. Два из них были продемонстрированы выше – это *табличный*, когда функция задается в виде *таблицы истинности*, представляющей собой совокупность всех наборов переменных и