2. ЭЛЕМЕНТЫ ДВОИЧНОЙ АЛГЕБРЫ

2.1. Понятие системы счисления. Основные системы счисления

Под *системой счисления* как правило понимают совокупность приемов записи и наименования чисел [5].

Системы счисления разделяют на *непозиционные*, в которых значение цифры не зависит от ее положения в записи числа, и *позиционные*, где значение каждой цифры изменяется с изменением ее позиции в числе [5].

На сегодняшний день в основном используются позиционные системы счисления. Главной характеристикой позиционной системы можно считать *основание системы счисления* p, определяющее количество цифр/знаков (от 0 до $p-1$), использующихся для записи чисел. Сами цифры $0\dots p-1$ называются *базисными числами* [5, 6].

Любое число N в системе счисления представляется в виде комбинации степеней основания p с коэффициентами (цифрами) a_i, относящимися к множеству базисных чисел $0\dots p-1$, как показано в формуле

$$a_k p^k + a_{k-1} p^{k-1} + \ldots + a_1 p + a_0,$$

и сокращенно записывается в виде $(a_k a_{k-1} \ldots a_1 a_0)_p$ [6].

В настоящее время широко используются четыре системы счисления.

1. *Десятичная система счисления* — позиционная система счисления с основанием 10. Для записи чисел в десятичной системе используются цифры: $0, 1, \ldots, 9$. Это основная система счисления, использующаяся повсеместно [5].

2. *Двоичная система счисления* — позиционная система счисления с основанием 2. Для записи чисел в двоичной системе используются две цифры: 0 и 1 [5]. Каждая цифра двоичного числа соответствует в десятичной системе счисления двойке в степени, равной номеру позиции цифры слева. В табл. 2.1 приведены значения первых двенадцати степеней.

Таблица 2.1

Степенной ряд двойки (до 12-й степени включительно)

Показатель степени	1	2	3	4	5	6	7	8	9	10	11	12
Значение	2	4	8	16	32	64	128	256	512	1024	2048	4096

3. *Восьмеричная система счисления* — позиционная система счисления с основанием 8. Для записи чисел используются цифры: $0, 1, \ldots, 7$ [5]. Используется, например, в некоторых справочниках для представления полиномов.

4. *Шестнадцатеричная система счисления* — позиционная система счисления с основанием 16. Для записи чисел используются цифры:

$0, 1, \ldots, 9, A, B, C, D, E, F$ [5]. Широко используется в программировании и телекоммуникациях, так как позволяет компактно и удобно представлять длинные двоичные последовательности.

Также стоит отметить так называемую *двоично-десятичную* систему счисления, в которой каждая цифра десятичного числа записывается соответствущим ей двоичным четырехразрядным числом. Эта система счисления используется в ЭВМ как промежуточная при преобразовании десятичных чисел в двоичные [5].

Для того, чтобы отличать числа в разных системах счисления (если надо записывать их вперемешку), используется несколько способов.

1. В конце числа указывается нижний индекс со значением основания системы счисления.
2. В конце числа указывается нижний индекс с названием системы счисления. Обычно используются первые три буквы латинского названия.
3. После числа указывается буквенный постфикс. Обычно используют первую букву латинского названия.
4. Перед числом указывается определенный префикс.

Последние два способа записи широко применяются в информатике, в частности в программировании, где невозможно использовать верхние и нижние индексы.

В табл. 2.2 показаны примеры различной записи чисел в разных системах счисления. Приведены только некоторые формы записи. В различных языках программирования и языках разметки могут использоваться другие префиксы и постфиксы. Также необходимо отметить, что при записи десятичных чисел постфиксы и префиксы не используются. Число без них по умолчанию считается десятичным, если иное не следует из контекста.

Таблица 2.2

Формы записи чисел в различных системах счисления

Основание системы	**Ниж. индекс со значением**	**Ниж. индекс с названием**	**Постфикс**	**Префикс**
2	1010_2	1010_{bin}	1010*b*	0*b*1010
8	1724_8	1724_{oct}	1724*o*	0*o*1724
10	1942_{10}	1942_{dec}	—	—
16	$3AF5_{16}$	$3AF5_{hex}$	3AF5*h*	0*x*3AF5

2.2. Перевод чисел между системами счисления

2.2.1. Перевод двоичного числа в десятичное

Перевод двоичного числа в десятичное производится по классической схеме сложения степеней двойки с коэффициентами, которыми являются соответствующие цифры этого двоичного числа.

Процесс перевода двоичного числа в десятичное рассмотрим на примере двоичного числа

$$100111001011_2.$$

Для простоты распишем показатели степени

$$\overset{11}{1}\ \overset{10}{0}\ \overset{9}{0}\ \overset{8}{1}\ \overset{7}{1}\ \overset{6}{1}\ \overset{5}{0}\ \overset{4}{0}\ \overset{3}{1}\ \overset{2}{0}\ \overset{1}{1}\ \overset{0}{1}{}_2.$$

Таким образом, для получения десятичного числа можно написать

$$1\cdot2^{11}+0\cdot2^{10}+0\cdot2^{9}+1\cdot2^{8}+1\cdot2^{7}+1\cdot2^{6}+0\cdot2^{5}+0\cdot2^{4}+1\cdot2^{3}+0\cdot2^{2}+1\cdot2^{1}+1\cdot2^{0}.$$

Убрав степени с коэффициентом 0, получим

$$1\cdot2^{11}+1\cdot2^{8}+1\cdot2^{7}+1\cdot2^{6}+1\cdot2^{3}+1\cdot2^{1}+1\cdot2^{0}.$$

Уберем коэффициенты и раскроем степени

$$2^{11}+2^{8}+2^{7}+2^{6}+2^{3}+2^{1}+2^{0}=2048+256+128+64+8+2+1=2507.$$

Таким образом

$$100111001011_2=2507_{10}.$$

2.2.2. Перевод десятичного числа в двоичное

Для перевода десятичного числа в двоичное используется процедура последовательного деления десятичного числа на два с накоплением остатка.

Для примера рассмотрим обратный перевод числа 2507_{10}, полученного в примере, приведенном в предыдущем подпункте.

$$
\begin{array}{r|r|r|r|r|r|r|r|r|r|r|r}
2507 & 2 &&&&&&&&&& \\
1 & 1253 & 2 &&&&&&&&& \\
 & 1 & 626 & 2 &&&&&&&& \\
 && 0 & 313 & 2 &&&&&&& \\
 &&& 1 & 156 & 2 &&&&&& \\
 &&&& 0 & 78 & 2 &&&&& \\
 &&&&& 0 & 39 & 2 &&&& \\
 &&&&&& 1 & 19 & 2 &&& \\
 &&&&&&& 1 & 9 & 2 && \\
 &&&&&&&& 1 & 4 & 2 & \\
 &&&&&&&&& 0 & 2 & 2 \\
 &&&&&&&&&& 0 & 1
\end{array}
$$

Результат записывается, начиная с конца. Таким образом

$$2507_{10} = 100111001011_2.$$

При переводе небольших десятичных чисел иногда удобнее использовать разложение в ряд степеней двойки (табл. 2.1), обратно рассмотренной в пп. 2.2.1 процедуре.

Например, число 292_{10} можно представить как

$$292_{10} = 256 + 32 + 4 = 2^8 + 2^5 + 2^2 = 100100100_2.$$

Вначале берем ближайшую к заданному десятичному числу степень двойки — в примере это число $256 = 2^8$. Теперь считаем разность $292 - 256 = 36$. Далее снова берем ближайшую степень двойки $32 = 2^5$. Остается $4 = 2^2$. В итоге получаем искомое число в двоичном виде.

2.2.3. Перевод между двоичной и восьмеричной системами счисления

Перевод целых чисел из двоичной системы в восьмеричную и обратно основан на том, что каждому восьмеричному числу ставится в соответствие его значение в двоичной системе счисления, выраженное в виде блока из трех двоичных цифр — *двоичная триада* [5]. Соответствия между восьмеричными цифрами и двоичными числами показаны в табл. 2.3

Таблица 2.3

Соответствия между цифрами восьмеричной системы счисления и двоичными числами (в виде триад)

Восьмеричные цифры	**Двоичные триады**	**Восьмеричные цифры**	**Двоичные триады**
0	000	4	100

Окончание табл. 2.3

Соответствия между цифрами восьмеричной системы счисления и двоичными числами (в виде триад)

Восьмеричные цифры	**Двоичные триады**	**Восьмеричные цифры**	**Двоичные триады**
1	001	5	101
2	010	6	110
3	011	7	111

При переводе двоичного целого числа в восьмеричную систему счисления его необходимо разбить на триады, начиная с младшего разряда, а затем вместо каждой триады записать соответствующую ей восьмеричную цифру [5]. Для примера рассмотрим перевод двоичного числа 1001001000011_2:

$$1001001000011_2 \Rightarrow 001.001.001.000.011 \Rightarrow 1.1.1.0.3 \Rightarrow 11103_8.$$

Заметим, что при разбиении двоичного числа на триады пришлось дополнить его нулями слева.

При переводе целого восьмеричного числа в двоичную систему счисления достаточно вместо каждой восьмеричной цифры записать соответствующую ей двоичную триаду [5]. Для примера рассмотрим обратный перевод восьмеричного числа 11103_8:

$$11103_8 \Rightarrow 001.001.001.000.011 \Rightarrow 1001001000011_2.$$

Незначащие нули в левой части полученного двоичного числа были удалены.

2.2.4. Перевод между двоичной и шестнадцатеричной системами счисления

Перевод целых чисел из двоичной системы в шестнадцатеричную и обратно аналогичен переводу между двоичной и восьмеричной системами. Каждой шестнадцатеричной цифре ставится в соответствие ее значение в двоичной системе счисления, выраженное в виде блока из четырех двоичных цифр — *двоичная тетрада* [5]. Соответствия между восьмеричными цифрами и двоичными числами показаны в табл. 2.4

Таблица 2.4

Соответствия между цифрами шестнадцатеричной системы счисления и двоичными числами (в виде тетрад)

Шестнадцатеричные цифры	**Двоичные тетрады**	**Шестнадцатеричные цифры**	**Двоичные тетрады**
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010

Окончание табл. 2.4

Соответствия между цифрами шестнадцатеричной системы счисления и двоичными числами (в виде тетрад)

Шестнадцатеричные цифры	**Двоичные тетрады**	**Шестнадцатеричные цифры**	**Двоичные тетрады**
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

При переводе двоичного целого числа в шестнадцатеричную систему счисления его необходимо разбить на тетрады, начиная с младшего разряда, а затем вместо каждой тетрады записать соответствующую ей шестнадцатеричную цифру [5]. Для примера рассмотрим перевод двоичного числа 1001001001011_2:

$$1001001001011_2 \Rightarrow 0001.0010.0100.1011 \Rightarrow 1.2.4.B \Rightarrow 124B_{16}.$$

Заметим, что при разбиении двоичного числа на тетрады пришлось дополнить его нулями слева.

При переводе целого шестнадцатеричного числа в двоичную систему счисления достаточно вместо каждой шестнадцатеричной цифры записать соответствующую ей двоичную тетраду [5]. Для примера рассмотрим обратный перевод шестнадцатеричного числа $124B_{16}$:

$$124B_{16} \Rightarrow 0001.0010.0100.1011 \Rightarrow 1001001001011_2.$$

Незначащие нули в левой части полученного двоичного числа были удалены.

2.2.5. Перевод между десятичной и двоично-десятичной системами счисления

Перевод целых чисел между десятичной и двоично-десятичной системами счисления аналогичен переводу между двоичной и шестнадцатеричной системами. Каждой десятичной цифре ставится в соответствие ее значение в двоичной системе счисления, выраженное в виде двоичной тетрады [5]. Соответствия между десятичными цифрами и двоичными тетрадами показаны в табл. 2.5

Таблица 2.5

Соответствия между цифрами десятичной системы счисления и двоичными тетрадами

Десятичные цифры	**Двоичные тетрады**	**Десятичные цифры**	**Двоичные тетрады**
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

При переводе целого десятичного числа в двоично-десятичную систему счисления достаточно вместо каждой десятичной цифры записать соответствующую ей двоичную тетраду [5]. Для примера рассмотрим перевод десятичного числа 124_{10}:

$$124_{10} \Rightarrow 0001.0010.0100 \Rightarrow 000100100100_{2-10}.$$

Обратный перевод из двоично-десятичной системы в десятичную полностью аналогичен переводу из двоичной в шестнадцатеричную систему.

2.3. Операции над двоичными числами

К основным операциям над двоичными числами относятся логические операции *инверсия*, *конъюнкция* и *дизъюнкция* и арифметические операции *сложение* и *умножение*, а также *сложение по модулю 2*, соответствующее логической операции «Исключающее ИЛИ».

Инверсия является *унарной* операцией, т. е. осуществляется над одним операндом. Прочие операции являются *бинарными*, т. е. осуществляются над двумя операндами.

Инверсия, конъюнкция и дизъюнкция являются основными логическими операциями. Этим операциям соответствуют простейшие логические элементы «НЕ» (*инвертор*), «И» и «ИЛИ», на основе которых строятся любые цифровые микросхемы. Для описания логических элементов используются *таблицы истинности*, содержащие совокупность всех возможных комбинаций логических сигналов на входе цифрового устройства и значений выходных сигналов для каждой комбинации. К основным логическим элементам также относят элемент «Исключающее ИЛИ», буфер памяти, а также обратные элементы «И–НЕ», «ИЛИ–НЕ», «Исключающее ИЛИ–НЕ».

Основные логические элементы имеют стандартные условно-графические обозначения. Существует несколько стандартов обозначения.

1. Стандарт Международной электротехнической комиссии (МЭК, International Electrotechnical Commission, IEC), аналогичный британскому стандарту BS3939.

2. Американский стандарт MIL/ANSI.

3. Обозначения, согласно книге A. Croft, R. Davidson and M. Hargreaves *Engineering Mathematics*, 1992. Эти обозначения часто называют CDH. Эти обозначения аналогичны стандарту MIL/ANSI за исключением обозначений для элементов «И»/«И–НЕ».

4. ГОСТ 2.743-91 «Единая система конструкторской документации. Обозначения условные графические в схемах. Элементы цифровой техники» (ЕСКД). Этот стандарт во многом аналогичен стандарту IEC.

Для основных логических элементов в пособии будут приведены все варианты обозначений. Основным вариантом, который будет использоваться для более сложных схем, как наиболее употребимый на практике, выбран стандарт MIL/ANSI.

2.3.1. Инверсия

Инверсия (*отрицание*, *дополнение*, *операция «НЕ»*, NOT) осуществляет смену значения операнда на противоположное. В том случае, если речь идет об инверсии двоичного числа из нескольких разрядов, эта операция осуществляется *поразрядно* (побитово), т. е. инвертируется каждый разряд (бит) числа.

В тексте операция инверсии может обозначаться различными способами, как показано в формуле:

$$\bar{a}; \qquad !a; \qquad \neg a. \tag{2.1}$$

В настоящем пособии будем использовать первое обозначение. Например

$$\overline{01110010} = 10001101.$$

Таблица истинности инвертора показана в табл. 2.6.

Таблица 2.6

Таблица истинности инвертора

a	$\bar{a}$
0	1
1	0

На рис. 2.1 представлены основные условно-графические обозначения инвертора.

(а) (б)

Рис. 2.1. Условно-графические обозначения инвертора:
(а) по стандартам IEC и ЕСКД; (б) по стандартам MIL/ANSI и CDH

2.3.2. Конъюнкция

Конъюнкция (*логическое умножение*, *операция «И»*, AND) является бинарной операцией, возвращающей 1, только когда оба операнда равны 1. При работе с двоичными числами из нескольких разрядов, эта операция осуществляется *поразрядно*. Поразрядные операции как правило осуществляются с операндами равной длины. В случае разной длины операндов, тот, который имеет меньшую длину, дополняется нулями в старших степенях. Например, при необходимости осуществить поразрядное логическое умножение чисел $c = 110010_2$ и $d = 1010_2$, число d следует представить как $d = 001010_2$.

В записи операция логического умножения может обозначаться различными способами, как показано в формуле:

$$a \wedge b; \qquad a \cap b; \qquad a \& b. \tag{2.2}$$

В настоящем пособии будем использовать первое обозначение.

Далее приведем таблицу логического умножения (табл. 2.7) и таблицу истинности соответствующего ему логического элемента «И» (табл. 2.8).

Таблица 2.7

Таблица логического умножения

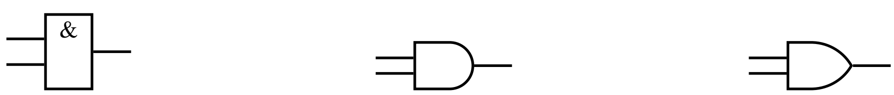
$\wedge$	0	1
0	0	0
1	0	1

Таблица 2.8

Таблица истинности элемента «И»

a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

На рис. 2.2 представлены основные условно-графические обозначения логического элемента «И».



(а) (б) (в)

Рис. 2.2. Условно-графические обозначения логического элемента «И»:
(а) по стандартам IEC и ЕСКД; (б) по стандарту MIL/ANSI; (в) по стандарту CDH

Рассмотрим операцию поразрядного логического умножения на примере двух двоичных чисел a и b:

$$a = 101101_2; \qquad b = 100110_2.$$

$$\begin{array}{r} \wedge \begin{array}{cccccc} 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 & 0 \end{array} \end{array}$$

Таким образом, $a \wedge b = 101101_2 \wedge 100110_2 = 100100_2$.

2.3.3. Дизъюнкция

Дизъюнкция (*логическое сложение*, *операция «ИЛИ»*, OR) является бинарной операцией, возвращающей 1, когда хотя бы один операнд равен 1. При работе с двоичными числами из нескольких разрядов, эта операция осуществляется *поразрядно*.

В записи операция логического сложения может обозначаться различными способами:

$$a \vee b; \qquad a \cup b; \qquad a | b. \tag{2.3}$$

В настоящем пособии будем использовать первое обозначение.

Далее приведем таблицу логического сложения (табл. 2.9) и таблицу истинности соответствующего ему логического элемента «ИЛИ» (табл. 2.10).

Таблица 2.9

Таблица логического сложения

$\vee$	0	1
0	0	1
1	1	1

Таблица 2.10

Таблица истинности элемента «ИЛИ»

a	b	$a \vee b$
0	0	0
0	1	1
1	0	1
1	1	1

На рис. 2.3 представлены основные условно-графические обозначения логического элемента «ИЛИ».

(а) (б)

Рис. 2.3. Условно-графические обозначения логического элемента «ИЛИ»: (а) по стандартам IEC и ЕСКД; (б) по стандартам MIL/ANSI и CDH

Рассмотрим операцию поразрядного логического сложения на примере двух двоичных чисел a и b:

$$a \vee b = 101101_2 \vee 100110_2 = 101111_2.$$

$$
\begin{array}{r}
\vee \begin{array}{cccccc} 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{array} \\
\hline
\begin{array}{cccccc} 1 & 0 & 1 & 1 & 1 & 1 \end{array}
\end{array}
$$

2.3.4. Сложение по модулю 2

Сложение по модулю 2 (*операция «Исключающее ИЛИ»*, XOR) является бинарной операцией, возвращающей 1, когда один операнд равен 0, а второй 1, и возвращающей 0 при равенстве операндов. Фактически, этот сумматор выполняет суммирование без учета переноса. При работе с двоичными числами из нескольких разрядов, эта операция осуществляется *поразрядно*.

В записи операция сложения по модулю 2 обычно обозначается способом, показанным в формуле:

$$a \oplus b. \tag{2.4}$$

Далее приведем таблицу сложения по модулю 2 (табл. 2.11) и таблицу истинности сумматора по модулю 2 (табл. 2.12).

Таблица 2.11

Таблица сложения по модулю 2

$\oplus$	0	1
0	0	1
1	1	0

Таблица 2.12

Таблица истинности сумматора по модулю 2

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Принципиальная схема сумматора по модулю 2 показана на рис. 2.4.

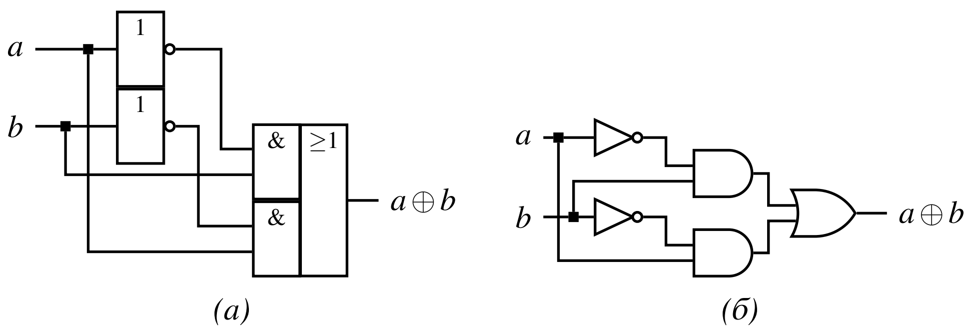


Рис. 2.4. Принципиальная схема сумматора по модулю 2: (а) по стандартам IEC и ЕСКД; (б) по стандарту MIL/ANSI

На рис. 2.5 представлены основные условно-графические обозначения сумматора по модулю 2.

(а) (б)

Рис. 2.5. Условно-графические обозначения сумматора по модулю 2: (а) по стандартам IEC и ЕСКД; (б) по стандартам MIL/ANSI и CDH

Рассмотрим операцию поразрядного сложения по модулю 2 на примере двух двоичных чисел a и b:

$$a \oplus b = 101101_2 \oplus 100110_2 = 001011_2.$$

$$\begin{array}{c|cccccc} & 1 & 0 & 1 & 1 & 0 & 1 \\ \oplus & 1 & 0 & 0 & 1 & 1 & 0 \\ \hline & 0 & 0 & 1 & 0 & 1 & 1 \end{array}$$

2.3.5. Сложение

В отличие от сложения по модулю 2 операция двоичного арифметического сложения производится с переносом в следующий двоичный разряд. То есть, $1_2 + 1_2 = 10_2$. В записи для обозначения двоичного арифметического сложения используется обычный символ сложения «+».

Поскольку необходимо учитывать перенос, схема построения двоичного сумматора усложняется по сравнению с сумматором по модулю 2. Выделяют два типа схем:

1) *полусумматор*, который формирует перенос в следующий разряд, но не может учитывать перенос из предыдущего разряда;

2) *полный сумматор*, который формирует перенос в следующий разряд, и учитывает перенос из предыдущего разряда.

Далее приведем таблицу двоичного сложения (табл. 2.13) и таблицу истинности одноразрядного полусумматора (табл. 2.14), в которой S обозначает сумму, а PO — перенос.

Таблица 2.13

Таблица двоичного сложения

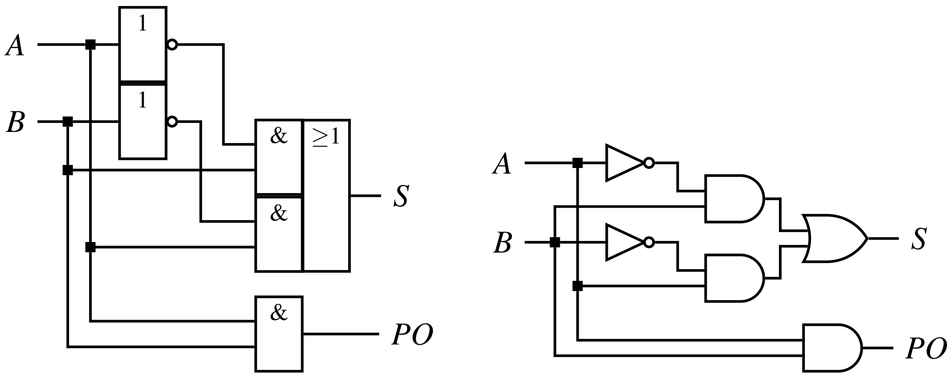
+	0	1
0	0	1
1	1	10

Таблица 2.14

Таблица истинности одноразрядного полусумматора

A	B	S	PO
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

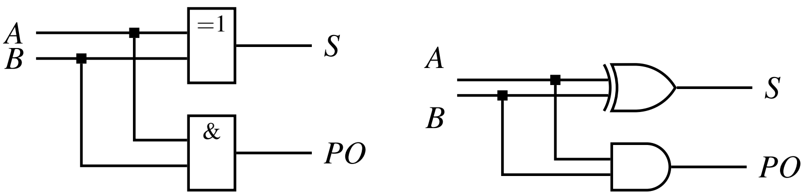
Принципиальная схема, реализующая таблицу истинности одноразрядного полусумматора, показана на рис. 2.6.



(а) (б)

Рис. 2.6. Принципиальная схема одноразрядного полусумматора:
(а) по стандартам IEC и ЕСКД; (б) по стандарту MIL/ANSI

Анализируя схему полусумматора на рис. 2.6, можно увидеть, что часть схемы, отвечающая за получение суммы, соответствует схеме сумматора по модулю 2, следовательно, схему полусумматора можно преобразовать так, как показано на рис. 2.7.



(а) (б)

Рис. 2.7. Принципиальная схема полусумматора на основе сумматора по модулю 2:
(а) по стандартам IEC и ЕСКД; (б) по стандарту MIL/ANSI

На рис. 2.8 приведено условно-графическое обозначение одноразрядного полусумматора.

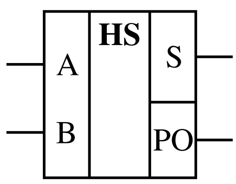


Рис. 2.8. Условно-графическое обозначение одноразрядного полусумматора

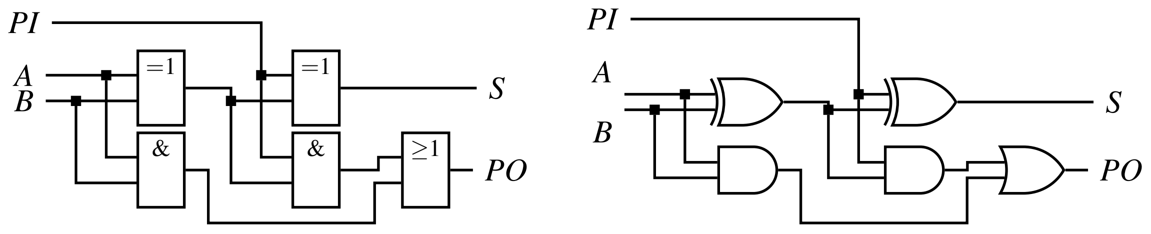
Таблица истинности полного двоичного одноразрядного сумматора показана в табл. 2.15, где S обозначает сумму, PI — перенос на входе, а PO — перенос на выходе.

Таблица 2.15

Таблица истинности полного двоичного одноразрядного сумматора

PI	*A*	*B*	*S*	*PO*
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Принципиальная схема полного двоичного одноразрядного сумматора на основе сумматоров по модулю 2 показана на рис. 2.9. Можно видеть, что эта схема состоит из двух полусумматоров и схемы ИЛИ, объединяющей сигналы переносов от полусумматоров.



(а) (б)

Рис. 2.9. Принципиальная схема полного двоичного одноразрядного сумматора на основе сумматоров по модулю 2:
(а) по стандартам IEC и ЕСКД; (б) по стандарту MIL/ANSI

На рис. 2.10 приведено условно-графическое обозначение полного двоичного одноразрядного сумматора.

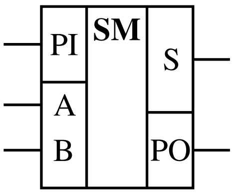
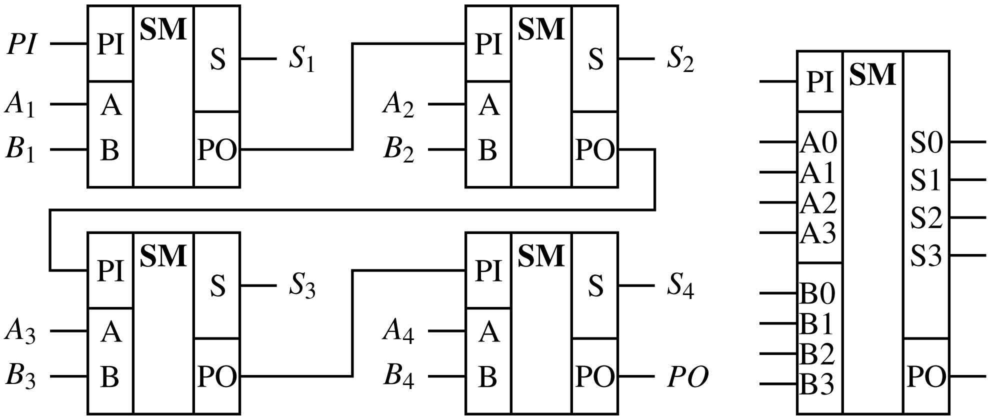


Рис. 2.10. Условно-графическое обозначение полного двоичного одноразрядного сумматора

Для того чтобы получить многоразрядный сумматор, достаточно соединить входы и выходы переносов соответствующих двоичных разрядов. На практике одноразрядные сумматоры практически никогда не использовались, так как почти сразу же были выпущены микросхемы многоразрядных сумматоров. На рис. 2.11 приведены принципиальная схема и условно-графическое обозначение полного двоичного четырехразрядного сумматора.



(а) (б)

Рис. 2.11. Полный двоичный четырехразрядный сумматор:
(а) принципиальная схема; (б) условно-графическое обозначение

При ручных расчетах арифметическое двоичное сложение a и b легко осуществляется методом сложения «в столбик»:

$$
\begin{array}{ccccccccl}
 & & & ^{1} & ^{1} & & & & \\
 & & 1 & 0 & 1 & 1 & 0 & 1 & = 45_{10} = a \\
+ & & 1 & 0 & 0 & 1 & 1 & 0 & = 38_{10} = b \\
\hline
 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & = 83_{10}
\end{array}
$$

2.3.6. *Умножение*

Операция арифметического умножения обычно рассматривается как многоразрядная, поскольку одноразрядное умножение полностью соответствует логическому умножению (конъюнкции), а одноразрядный умножитель, соответственно, схеме «И».

При ручных расчетах умножение, как и сложение, удобно вычисляется «в столбик». Рассмотрим на примере $b \cdot a$:

$$
\begin{array}{ccccccccccccl}
 & & & & & \times & 1 & 0 & 0 & 1 & 1 & 0 & = 38_{10} = b \\
 & & & & & & 1 & 0 & 1 & 1 & 0 & 1 & = 45_{10} = a \\
\hline
 & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & \\
 & & & & & 1 & 0 & 1 & 1 & 0 & 1 & & \\
 & & & & 1 & 0 & 1 & 1 & 0 & 1 & & & \\
 & 1 & 0 & 1 & 1 & 0 & 1 & & & & & & \\
\hline
 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & = 1710_{10}
\end{array}
$$

Из этого расчета видно, что умножение двух двоичных чисел представляет собой последовательное суммирование первого множителя с ним же, но сдвинутым согласно позициям единиц во втором множителе.

Контрольные вопросы

1. Что такое система счисления? Какие системы счисления сейчас применяются?
2. Как осуществляется перевод из десятичной системы счисления в двоичную и обратно?
3. Как осуществляется перевод между двоичной и шестнадцатеричной системами счисления?
4. Что такое двоично-десятичная система счисления?
5. Что такое конъюнкция?
6. Что такое дизъюнкция?
7. Нарисуйте схему сумматора по модулю 2.
8. Как работает полный двоичный сумматор? Нарисуйте схему.
9. Как строится полный двоичный многоразрядный сумматор?
10. Как производится перемножение многоразрядных двоичных чисел?