типа «Internet on a chip» («Интернет на чипе»). Подобная продукция может быть представлена десятками фирм.

Для подключения СК в сеть нужны каналы связи. Принципиально возможно использование любых каналов. Наиболее дешевыми, но мало скоростными являются телефонные линии связи. Их пропускная способность составляет до 30 Кбайт в секунду через аналоговые модемы и в несколько раз больше при переходе к цифровым методам связи. Каждый СК при работе с сетью должен пользоваться сетевыми ресурсами, что может вызывать перегруженность линий, обслуживающих большое число СК. Поэтому требуется повышать скорости передачи данных в сетях и качество используемых каналов.

Для новых аппаратных средств сети необходимы компактные управляющие программы и приложения для серверов. Индустрия соответствующего программного обеспечения постепенно набирает силу.

Контрольные вопросы

1. Каково понятие архитектуры ЭВМ?
2. По каким техническим характеристикам осуществляется оценка и выбор ЭВМ?
3. Какова связь областей применения ЭВМ и их структур?
4. Каковы основные тенденции развития ЭВМ?
5. Охарактеризуйте понятие машинного парка.
6. Каковы основные принципы построения ЭВМ?
7. Поясните место и роль программного обеспечения ЭВМ.
8. Что представляет собой класс персональных ЭВМ?
9. Основы классификации сетевых компьютеров.
10. Назначение и отличительные особенности построения сетевых компьютеров.

Глава 2
ИНФОРМАЦИОННО-ЛОГИЧЕСКИЕ ОСНОВЫ ЭВМ

2.1. Системы счисления

Системой счисления называется способ изображения чисел с помощью ограниченного набора символов, имеющих определенные количественные значения. Систему счисления образует совокупность правил и приемов представления чисел с помощью набора знаков (цифр).

Различают ***позиционные*** и ***непозиционные*** системы счисления. В позиционных системах каждая цифра числа имеет определенный вес, зависящий от позиции цифры в последовательности, изображающей число. Позиция цифры называется ***разрядом.*** В позиционной системе счисления любое число можно представить в виде

$$A_n = a_{m-1}a_{m-2}\ldots a_i \ldots a_0 \cdot a_{-1}a_{-2}\ldots a_{-k} = a_{m-1}\cdot N^{m-1} + a_{m-2}\cdot N^{m-2}\cdots + a_{-k}\cdot N^{-k}$$

$$A_n = \sum_{i=-k}^{m-1} a_i \cdot N^i, \tag{2.1}$$

где: a_i — i-я цифра числа;
k — количество цифр в дробной части числа;
m — количество цифр в целой части числа;
N — основание системы счисления.

Основание системы счисления N показывает, во сколько раз «вес» i-го разряда больше $(i-1)$ разряда. Целая часть числа отделяется от дробной части точкой (запятой).

Пример 2.1. A_{10}=37.25.

В соответствии с формулой (2.1) это число формируется из цифр с весами разрядов:

$$A_{10} = 3\cdot 10^1 + 7\cdot 10^0 + 2\cdot 10^{-1} + 5\cdot 10^{-2}.$$

Теоретически наиболее экономичной системой счисления является система с основанием e =2,71828..., находящимся между числами 2 и 3.

Во всех современных ЭВМ для представления числовой информации используется двоичная система счисления. Это обусловлено:

- более простой реализацией алгоритмов выполнения арифметических и логических операций;
- более надежной физической реализацией основных функций, так как они имеют всего два состояния (0 и 1);
- экономичностью аппаратной реализации всех схем ЭВМ.

При N=2 число различных цифр, используемых для записи чисел, ограничено множеством из двух цифр (нуль и единица). Кроме двоичной системы счисления широкое распространение получили и производные системы:

- двоичная — {0,1};
- десятичная, точнее, двоично-десятичное представление десятичных чисел — {0, 1, ..., 9};
- шестнадцатеричная — {0, 1, 2, ..., 9, A, B, C, D, E, F}. Здесь шестнадцатеричная цифра A обозначает число 10, B — число 11, ..., F — число 15;
- восьмеричная (от слова *восьмерик*) — {0, 1, 2, 3, 4, 5, 6, 7}. Она широко используется во многих специализированных ЭВМ.

Восьмеричная и шестнадцатеричная системы счисления являются производными от двоичной, так как $16 = 2^4$ и $8 = 2^3$. Они используются в основном для более компактного изображения двоичной информации, так как запись значения чисел производится существенно меньшим числом знаков.

Пример 2.2. Число $A_{10} = 100.625$ в двоичной, восьмеричной и шестнадцатеричной системах счисления имеет следующее представление:

$A_2 = 1100100.101;$
$A_8 = 144.5;$
$A_{16} = 64.A;$
$A_2 = 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3};$
$A_8 = 1 \cdot 8^2 + 4 \cdot 8^1 + 4 \cdot 8^0 + 5 \cdot 8^{-1};$
$A_{16} = 6 \cdot 16^1 + 4 \cdot 16^0 + 10 \cdot 16^{-1}.$

Представление чисел в различных системах счисления допускает однозначное преобразование их из одной системы в другую. В ЭВМ перевод из одной системы в другую осуществляется автоматически, по специальным программам. Правила перевода целых и дробных чисел отличаются.

2.1.1. Перевод целых чисел

Целое число с основанием N_1 переводится в систему счисления с основанием N_2 путем последовательного деления числа A_{N_1} на основание N_2, записанного в виде числа с основанием N_1, до получения остатка. Полученное частное следует вновь делить на основание N_2, и этот процесс надо повторять до тех пор, пока частное не станет меньше делителя. Полученные остатки от деления и последнее частное записываются в порядке, обратном полученному при делении. Сформированное число и будет являться числом с основанием N_2.

Пример 2.3. $A_{10} = 37$; $A_2 = ?$; $A_{16} = ?$

1) 37 | 2
1 18 | 2
0 9 | 2
1 4 | 2
0 2 | 2
0 1

2) 37 | 16
5 2

$A_{10} = 37$
$A_2 = 100101$

$A_{10} = 37$
$A_{16} = 25$

2.1.2. Перевод дробных чисел

Дробное число с основанием N_1 переводится в систему счисления с основанием N_2 путем последовательного умножения A_{N_1} на основание N_2, записанное в виде числа с основанием N_1. При каждом умножении целая часть произведения берется в виде очередной цифры соответствующего разряда, а оставшаяся дробная часть принимается за новое множимое. Число умножений определяет разрядность полученного результата, представляющего число A_{N_1} в системе счисления N_2.

Пример 2.4. $A_{10} = 0{,}625$; $A_2 = ?$; $A_{16} = ?$

a)	b)	c)
0.625	0.625	0.625
×2	×8	×16
1 250	5 000	10.000
×2		
0 500		
×2		
1 000		
$A_2 = 0.101$	$A_8 = 0.5$	$A_{16} = 0.A$

Так как двоичная, восьмеричная и шестнадцатеричная системы связаны через степени числа 2, то преобразования между ними можно выполнять другим, более простым, способом. Для перевода из шестнадцатеричной (восьмеричной) системы счисления в двоичную достаточно двоичным кодом записать шестнадцатеричные коды цифр тетрадами (по 4 двоичных разряда) и триадами (по 3 двоичных разряда) — для восьмеричных цифр. Обратный перевод из двоичного кода производится в обратном порядке: двоичное число разбивается влево и вправо от границы целой и дробной частей: на тетрады — для последующей записи цифр в шестнадцатеричном представлении; на триады — для записи их значений восьмеричными цифрами.

2.2. Представление информации в ЭВМ

Информация — это сведения об окружающем мире и протекающих в нем процессах, воспринимаемые человеком или специализированным устройством, например ЭВМ, для обеспечения целенаправленной деятельности.

Информация может быть по своей физической природе: числовой, текстовой, графической, звуковой, видео и др. Она также может быть постоянной (не меняющейся), переменной, случайной, вероятностной. Наибольший интерес представляет переменная информация, так как она позволяет выявлять причинно-следственные связи в процессах и явлениях. Существуют различные способы оценки количества информации. Классическим является подход, использующий формулу К.Шеннона. Применительно к двоичной системе она имеет вид

$$H = \log_2 N,$$

где: H — количество информации, несущей представление о состоянии, в котором находится объект;

N — количество равновероятных альтернативных состояний объекта.

Любая информация, обрабатываемая в ЭВМ, должна быть представлена двоичными цифрами {0,1}, т.е. должна быть закодирована комбинацией этих цифр. Различные виды информации (числа, тексты, графика, звук) имеют свои правила кодирования. Коды отдельных значений, относящиеся к различным видам информации, могут совпадать. Поэтому расшифровка кодированных данных осуществляется по контексту при выполнении команд программы.

2.2.1. Представление числовой информации

В ЭВМ используются три вида чисел: с фиксированной точкой (запятой), с плавающей точкой (запятой) и двоично-десятичное представление. Точка (запятая) — это подразумеваемая граница целой и дробной частей числа.

У чисел с фиксированной точкой в двоичном формате предполагается строго определенное место точки (запятой). Обычно это место определяется или перед первой значащей цифрой числа, или после последней значащей цифры числа. Если точка фиксируется перед первой значащей цифрой, то это означает, что число по модулю меньше единицы. Диапазон изменения значений чисел определяется неравенством

$$2^{-n} \le |A_2| \le 1 - 2^{-n}.$$

Если точка фиксируется после последней значащей цифры, то это означает, что n-разрядные двоичные числа являются целыми. Диапазон изменения их значений составляет:

$$0 \le |A_2| \le 2^n - 1.$$

Перед самым старшим из возможных разрядов двоичного числа фиксируется его знак. Положительные числа имеют нулевое значение знакового разряда, отрицательные — единичное.

Другой формой представления чисел является представление их в виде чисел с плавающей точкой (запятой). Числа с плавающей точкой представляются в виде мантиссы m_a и порядка p_a, иногда это представление называют полулогарифмической формой числа. Например, число $A_{10} = 373$ можно представить в виде $0.373 \cdot 10^3$, при этом $m_a = 0.373$, $p_a = 3$, основание системы счисления подразумевается фиксированным и равным десяти. Для двоичных чисел A_2 в этом представлении также формируется мантисса m_a и порядок p_a при основании системы счисления, равном двум:

$$A_2 = \pm p_a; \pm m_a,$$

что соответствует записи

$$A_2 = 2^{\pm p_a} \cdot (\pm m_a).$$

Порядок числа p_a определяет положение точки (запятой) в двоичном числе. Значение порядка лежит в диапазоне $-p_a^{\max} \le p_a \le p_a^{\max}$, где величина $p_a^{\max}$ определяется числом разрядов r, отведенных для представления порядка

$$p_a^{\max} = 2^r - 1.$$

Положительные и отрицательные значения порядка значительно усложняют обработку вещественных чисел. Поэтому во многих современных ЭВМ используют не прямое значение p_a, а модифицированное p_a', приведенное к интервалу

$$0 \le p_a' \le 2\, p_a^{\max}.$$

Значение p_a' носит название «характеристики числа».

Обычно под порядок (модифицированный порядок — характеристику) выделяют один байт. Старший разряд характеристики отводится под знак числа, а семь оставшихся разрядов обеспечивают изменение порядка в диапазоне

$$-64 \le p_a \le 63.$$

Модифицированный порядок p_a' вычисляется по зависимости

$$p_a' = p_a + 64.$$

Этим самым значения p_a' формируются в диапазоне положительных чисел

$$0 \le p_a' \le 127.$$

Мантисса числа m_a представляется двоичным числом, у которого точка фиксируется перед старшим разрядом, т. е.

$$0 \le |m_a| \le 1 - 2^{-k},$$

где k — число разрядов, отведенных для представления мантиссы.

Если

$$1/N \le |m_a| \le 1 - 2^{-k},$$

то старший значащий разряд мантиссы в системе счисления с основанием N отличен от нуля. Такое число называется нормализованным. Например, $A_2 = (100;0.101101)_2$ — нормализованное число $A_2 = 1011.01$ или $A_{10} = 11.25$, а то же самое число $A_2 = (101;0.0101101)$ — число ненормализованное, так как старший разряд мантиссы равен нулю.

Диапазон представления нормализованных чисел с плавающей точкой определяется так:

$$2^{-1} \cdot 2^{(2^r - 1)} \le |A_2| \le (1 - 2^{-k}) \cdot 2^{(2^r - 1)},$$

где r и k — соответственно количество разрядов, используемых для представления порядка и мантиссы.

Третья форма представления двоичных чисел — двоично-десятичная. Ее появление объясняется следующим. При обработке больших массивов десятичных чисел (например, больших экономических документов) приходится тратить много времени на перевод этих чисел из десятичной системы счисления в двоичную для последующей обработки и обратно — для вывода результатов. Каждый такой перевод требует выполнения двух — четырех десятков машинных команд. С включением в состав отдельных ЭВМ специальных функциональных блоков или спецпроцессоров десятичной арифметики появляется возможность обрабатывать десятичные числа напрямую, без их преобразования, что сокращает время вычислений. При этом каждая цифра десятичного числа представляется двоичной тетрадой. Например, $A_{10} = 3759$, $A_{2-10} = 0011\ 0111\ 0101\ 1001$. Положение десятичной точки (запятой), отделяющей целую часть от дробной, обычно заранее фиксируется. Значение знака числа отмечается кодом, отличным от кодов цифр. Например, знак «+» имеет значение тетрады «1100», а знак «—» — «1101».

2.2.2. Представление других видов информации

До последнего времени практически все системы связи России, системы передачи аудио- и видеоинформации, включая центральное радио и телевидение, строились на принципах передачи аналоговой информации. Это подразумевало выполнение процедур модуляции (преобразование данных в высокочастотные сигналы при передаче) и демодуляции для обратного преобразования и воспроизведения принятых данных.

С развитием микроэлектроники и компьютерных технологий все большее распространение получают цифровые системы передачи данных. В их основу положены процедуры квантования аналоговой информации по времени и величине. Значения функции $y=f(t)$ измеряются с большой точностью в моменты времени 0, Δt, $2\Delta t$, ..., $n\Delta t$ (Δt=const). Эта последовательность дискретных измерений пересылается абоненту, у которого по ним воссоздается значение функции. Качество воспроизведения функции $y=f(t)$ при $\Delta t \rightarrow 0$ может быть очень высоким. Более подробно эти вопросы освещаются в п. 13.6.

По скорости изменения обрабатываемых цифровых данных информация может быть условно разделена на два вида: ***статический*** и ***динамический.*** Например, числовая, логическая и символическая информация является статической, так как ее значение не связано со временем. В отличие от перечисленных типов вся аудиоинформация имеет динамический характер. Она существует только в режиме реального времени, ее нельзя остановить для более подробного изучения. Если изменить масштаб времени (увеличить или уменьшить), аудиоинформация искажается. Это свойство иногда используется для получения звуковых эффектов.

Видеоинформация может быть как статической, так и динамической. Статическая видеоинформация включает текст, рисунки, графики, чертежи, таблицы и др. Рисунки делятся также на плоские — двумерные и объемные — трехмерные.

Динамическая видеоинформация — это видео-, мульт- и слайд-фильмы. В их основе лежит последовательное экспонирование на экране в реальном масштабе времени отдельных кадров в соответствии со сценарием.

Динамическая видеоинформация используется либо для передачи движущихся изображений (анимация), либо для последовательной демонстрации отдельных кадров вывода (слайд-фильмы).

Для демонстрации анимационных и слайд-фильмов используются различные принципы. Анимационные фильмы демонстрируются так, чтобы зрительный аппарат человека не мог зафиксировать отдельные кадры. В современных высококачественных мониторах и в телевизорах с цифровым управлением электронно-лучевой трубкой кадры сменяются до 70 раз в секунду, что позволяет высококачественно передавать движение объектов.

При демонстрации слайд-фильмов каждый кадр экспонируется на экране столько времени, сколько необходимо для восприятия его человеком (обычно от 30 с до 1 мин.). Слайд-фильмы можно отнести к статической видеоинформации.

По способу формирования видеоизображения бывают ***растровые, матричные*** и ***векторные.***

Растровые видеоизображения используются в телевидении, а в ЭВМ практически не применяются.

Матричные изображения получили в ЭВМ наиболее широкое распространение. Изображение на экране рисуется электронным лучом точками.

Информация представляется в виде характеристик значений каждой точки — пиксела (picture element), рассматриваемой в качестве наименьшей структурной единицы изображения. Количество высвечиваемых одновременно пикселов на экране дисплея определяется его разрешающей способностью. В качестве характеристик графической информации выступают: координаты точки (пиксела) на экране, цвет пиксела, цвет фона (градация яркости). Вся эта информация хранится в видеопамяти дисплея. При выводе графической информации на печать изображение также воспроизводится по точкам.

Изображение может быть представлено и в векторной форме. Тогда оно составляется из отрезков линий (в простейшем случае — прямых), для которых задаются: начальные координаты, угол наклона и длина отрезка (может указываться и код используемой линии). Векторный способ имеет ряд преимуществ перед матричным: изображение легко масштабируется с сохранением формы, является «прозрачным» и может быть наложено на любой фон и т.д.

Способы представления информации в ЭВМ, ее кодирование и преобразование имеют очень большое значение в информационных системах. Они сильно зависят от стандартов, используемых в отдельных странах и фирмах, от типа приобретенного и действующего оборудования и других условий. С появлением вычислительных сетей, в которых информация циркулирует между странами и континентами, претерпевая многократные перекодировки, возникла проблема адекватного ее воспроизведения. Существует множество стандартов (и они продолжают множиться), используемых в сетях связи и представлении данных в ПК (МТК-5, КОИ-7, ДКОИ-8, EBDIC, кодировки DOS, 866, Windows-1251, Западно-европейская и др.). Рассмотрим особенности такого кодирования.

Для кодирования ***символьной*** и ***текстовой*** информации последовательно используется несколько систем кодировок. При вводе информации с клавиатуры нажатие определенной клавиши вырабатывает так называемый scan-код, представляющий собой двоичное число, равное порядковому номеру клавиши.

Номер нажатой клавиши никак не связан с формой символа, нанесенного на клавише. Опознание символа и присвоение ему внутреннего кода ЭВМ производится специальной программой по специальным таблицам: ДКОИ, КОИ-7, ASCII (Американский стандартный код передачи информации).

Всего с помощью таблицы кодирования ASCII (табл. 2.1) можно закодировать 256 различных символов. Эта таблица разделена на две части: основную (с кодами от 00h до 7Fh) и дополнительную (от 80h до FFh, где буква h обозначает принадлежность кода к шестнадцатеричной системе счисления).

Первая половина таблицы стандартизована. Она содержит управляющие коды (от 00h до 20h и 77). Эти коды в таблице занимают две первые строки. Они не относятся к текстовым элементам, поэтому часть из них опущена. Здесь же размещаются знаки пунктуации и математические знаки: 21h — !, 26h — &, 28h — (, 2Bh — +, ..., большие и малые латинские буквы: 41h — A, 61h — a, ...

Таблица 2.1

Таблица кодирования текстовой информации

ASCII

Radix: Hex

		♥	♦	♣	♠										
s	s		█	¶	§			↑	↓	→	←			s	t
	!	"	#	$	%	&	'	(	)	*	+	,	–	.	/
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	█
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
□ р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
Ё	ё	Є	є	Ї	ї	Ў	ў	°	•	·	√	№	¤	■	

Вторая половина таблицы содержит национальные шрифты, символы псевдографики, из которых могут быть построены таблицы, специальные математические знаки. Нижнюю часть таблицы кодировок можно заменять, используя соответствующие драйверы — управляющие вспомогательные программы. Этот прием позволяет применять несколько шрифтов и их гарнитур.

Дисплей по этому коду должен вывести на экран изображение символа — не просто цифровой код, а соответствующую ему картинку, так как каждый символ имеет свою форму.

Описание формы каждого символа хранится в специальной памяти дисплея — знакогенераторе.

Высвечивание символа на экране дисплея IBM PC осуществляется с помощью точек, образующих символьную матрицу.

Каждый пиксел в такой матрице является элементом изображения и может быть ярким или темным. Темная точка кодируется цифрой «0», светлая (яркая) — цифрой «1».

Если изображать в матричном поле знака темные пикселы точкой, а светлые — звездочкой, то можно графически изобразить форму символа.

Программы, работающие в операционной среде Windows, применяют совершенно другую кодовую таблицу, поддерживающую векторные шрифты TrueType. В ней отсутствуют все символы псевдографики, так как используется настоящая графика.

Кодирование аудиоинформации — процесс более сложный. Аудиоинформация является аналоговой. Для преобразования ее в цифровую форму используют аппаратные средства: аналого-цифровые преобразователи (АЦП), в результате работы которых аналоговый сигнал оцифровывается — представляется в виде числовой последовательности. Для вывода оцифрованного звука на аудиоустройства необходимо проводить обратное преобразование, которое осуществляется с помощью цифро-аналоговых преобразователей (ЦАП) .

2.3. Арифметические основы ЭВМ

Все современные ЭВМ имеют достаточно развитую систему команд, включающую десятки и сотни машинных операций. Однако выполнение любой операции основано на использовании простейших микроопераций типа сложения и сдвиг. Это позволяет иметь единое арифметико-логическое устройство для выполнения любых операций, связанных с обработкой информации. Правила сложения двоичных цифр двух чисел А и В представлены в табл. 2.2.

Таблица 2.2

Правила сложения двоичных цифр

Значения двоичных чисел А и В			Разряд суммы Si	Перенос в следующий разряд Pi
ai	bi	pi-1		
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

Здесь показаны правила сложения двоичных цифр a_i, b_i одноименных разрядов с учетом возможных переносов из предыдущего разряда p_{i-1}.

Подобные таблицы можно было бы построить для любой другой арифметической или логической операции (вычитание, умножение и т.д.), но именно данные этой таблицы положены в основу выполнения любой операции ЭВМ. Под знак чисел отводится специальный знаковый разряд. Знак «+» кодируется двоичным нулем, а знак «–» — единицей. Действия над прямыми кодами двоичных чисел при выполнении операций создают большие трудности, связанные с необходимостью учета значений знаковых разрядов:

- во-первых, следует отдельно обрабатывать значащие разряды чисел и разряды знака;
- во-вторых, значение разряда знака влияет на алгоритм выполнения операции (сложение может заменяться вычитанием и наоборот).

Во всех без исключения ЭВМ все операции выполняются над числами, представленными специальными машинными кодами. Их использование позволяет обрабатывать знаковые разряды чисел так же, как и значащие разряды, а также заменять операцию вычитания операцией сложения.

Различают прямой код (П), обратный код (ОК) и дополнительный код (ДК) двоичных чисел.

2.3.1. Машинные коды

Прямой код двоичного числа образуется из абсолютного значения этого числа и кода знака (нуль или единица) перед его старшим числовым разрядом.

Пример 2.5.

$A_{10} = +10;\quad A_2 = +1010;\quad [A_2]п = 0 \vdots 1010;$

$B_{10} = -15;\quad B_2 = -1111;\quad [B_2]п = 1 \vdots 1111.$

Точечной вертикальной линией здесь отмечена условная граница, отделяющая знаковый разряд от значащих.

Обратный код двоичного числа образуется по следующему правилу. Обратный код положительных чисел совпадает с их прямым кодом. Обратный код отрицательного числа содержит единицу в знаковом разряде числа, а значащие разряды числа заменяются на инверсные, т.е. нули заменяются единицами, а единицы — нулями.

Пример 2.6.

$A_{10} = +5;\quad A_2 = +101;\quad [A_2]п = [A_2]ок = 0 \vdots 101;$

$B_{10} = -13;\quad B_2 = -1101;\quad [B_2]ок = 1 \vdots 0010.$

Свое название обратный код чисел получил потому, что коды цифр отрицательного числа заменены на инверсные. Укажем наиболее важные свойства обратного кода чисел:

- сложение положительного числа С с его отрицательным значением в обратном коде дает так называемую машинную единицу МЕок=1⁝ 111...11, состоящую из единиц в знаковом и в значащих разрядах числа;
- нуль в обратном коде имеет двоякое значение. Он может быть как положительным числом — 0⁝ 00...0, так и отрицательным — 1⁝ 11...11. Значение отрицательного нуля совпадает с МЕок. Двойственное представление нуля явилось причиной того, что в современных ЭВМ все числа представляются не обратным, а дополнительным кодом.

Дополнительный код положительных чисел совпадает с их прямым кодом. Дополнительный код отрицательного числа представляет собой результат суммирования обратного кода числа с единицей младшего разряда (2^0 — для целых чисел, 2^{-k} — для дробных).

Пример 2.7.

$A_{10} = +19$; $A_2 = +10011$; $[A_2]п = [A_2]ок = [A_2]дк = 0⁝ 10011$;

$B_{10} = -13$; $B_2 = -1101$; $[B_2]дк = [B_2]ок + 2^0 = 1⁝ 0010 + 1 = 1⁝ 0011$.

Укажем основные свойства дополнительного кода:

- сложение дополнительных кодов положительного числа С с его отрицательным значением дает так называемую машинную единицу дополнительного кода:

$$МЕдк=МЕок+2^0=10⁝ 00...00,$$

т.е. число 10 (два) в знаковых разрядах числа;

- дополнительный код получил такое название потому, что представление отрицательных чисел является дополнением прямого кода чисел до машинной единицы МЕдк.

Модифицированные обратные и ***дополнительные коды*** двоичных чисел отличаются соответственно от обратных и дополнительных кодов удвоением значений знаковых разрядов. Знак «+» в этих кодах кодируется двумя нулевыми знаковыми разрядами, а знак «–» — двумя единичными разрядами.

Пример 2.8.

$A_{10} = 9$; $A_2 = +1001$; $[A_2]п = [A_2]ок = [A_2]дк = 0⁝ 1001$;
$[A_2]мок = [A_2]мдк = 00⁝ 1001$;

$B_{10} = -9$; $B_2 = -1001$; $[B_2]ок = 1⁝ 0110$; $[B_2]дк = 1⁝ 0111$;
$[B_2]мок = 11⁝ 0110$ $[B_2]мдк = 11⁝ 0111$.

Целью введения модифицированных кодов являются фиксация и обнаружение случаев получения неправильного результата, когда значение результата превышает максимально возможный результат в отведенной разрядной сетке машины. В этом случае перенос из значащего разряда может исказить значение младшего знакового разряда. Значение знаковых разрядов «01» свидетельствует о положительном переполнении разрядной сетки, а «10» — об отрицательном переполнении. В настоящее время практически во всех моделях ЭВМ роль удвоенных разрядов для фиксации переполнения разрядной сетки играют переносы, идущие в знаковый и из знакового разряда.

2.3.2. Арифметические операции над числами с фиксированной точкой

Сложение (вычитание). Операция вычитания приводится к операции сложения путем преобразования чисел в обратный или дополнительный код. Пусть числа A≥0 и B≥0, тогда операция алгебраического сложения выполняется в соответствии с табл. 2.3.

Таблица 2.3

Таблица преобразования кодов при алгебраическом сложении

Требуемая операция	Необходимое преобразование
A+B	A+B
A–B	A+(–B)
–A+B	(–A)+B
–A–B	(–A)+(–B)

Скобки в представленных выражениях указывают на замену операции вычитания операцией сложения с обратным или дополнительным кодом соответствующего числа. Сложение двоичных чисел осуществляется последовательно, поразрядно в соответствии с табл. 2.2. При выполнении сложения цифр необходимо соблюдать следующие правила.

1. Слагаемые должны иметь одинаковое число разрядов. Для выравнивания разрядной сетки слагаемых можно дописывать незначащие нули слева к целой части числа и незначащие нули справа к дробной части числа.

2. Знаковые разряды чисел участвуют в сложении так же, как и значащие.

3. Необходимые преобразования кодов (п.2.3.1) производятся с изменением знаков чисел. Приписанные незначащие нули изменяют свое значение при преобразованиях по общему правилу.

4. При образовании единицы переноса из старшего знакового разряда, в случае использования ОК, эта единица складывается с младшим числовым разрядом. При использовании ДК единица переноса теряется. Знак результата формируется автоматически, результат представляется в том коде, в котором представлены исходные слагаемые.

Пример 2.9. Сложить два числа: $A_{10}=7$; $B_{10}=16$.

$$A_2 = +111 = +0111;$$
$$B_2 = +1000 = +10000.$$

Исходные числа имеют различную разрядность, необходимо провести выравнивание разрядной сетки:

$$[A_2]п = [A_2]ок = [A_2]дк = 0 \vdots 00111;$$
$$[B_2]п = [B_2]ок = [B_2]дк = 0 \vdots 10000.$$

Сложение в обратном или дополнительном коде дает один и тот же результат:

$$\begin{array}{r} 0 \vdots 00111 \\ +0 \vdots 10000 \\ \hline C_2 = 0 \vdots 10111_2 \\ C_{10} = +23. \end{array}$$

Обратим внимание, что при сложении цифр отсутствуют переносы в знаковый разряд и из знакового разряда, что свидетельствует о получении правильного результата.

Пример 2.10. Сложить два числа: $A_{10} = +16$; $B_{10} = -7$ в ОК и ДК.

В соответствии с табл. 2.3 должна быть реализована зависимость A+(–B), в которой второй член преобразуется с учетом знака

$$[A_2]п = 0 \vdots 10000 = 0 \vdots 10000; \quad [A_2]ок = 0 \vdots 10000; \quad [A_2]дк = 0 \vdots 10000;$$
$$[B_2]п = 1 \vdots 111 = 1 \vdots 00111; \quad [B_2]ок = 1 \vdots 11000; \quad [B_2]дк = 1 \vdots 11001.$$

Сложение в ОК	Сложение в ДК
$\begin{array}{r} [A_2]ок = 0 \vdots 10000 \\ +[B_2]ок = 1 \vdots 11000 \\ \hline 0 \vdots 01000 \\ + \qquad 1 \\ \hline 0 \vdots 01001 \end{array}$	$\begin{array}{r} [A_2]дк = 0 \vdots 10000 \\ +[B_2]дк = 1 \vdots 11001 \\ \hline 0 \vdots 01001 \end{array}$
$C_2 = 0 \vdots 01001$	$C_2 = 0 \vdots 01001$
$C_{10} = +9$	$C_{10} = +9$

При сложении чисел в ОК и ДК были получены переносы в знаковый разряд и из знакового разряда. В случае ОК перенос из знакового разряда требует дополнительного прибавления единицы младшего разряда (см. п.4 правил). В случае ДК этот перенос игнорируется.

Умножение. Умножение двоичных чисел наиболее просто реализуется в прямом коде. Рассмотрим, каким образом оно приводится к операциям сложения и сдвигам.

Пример 2.11. Умножить два числа $A_{10}=7$; $B_{10}=5$.

Перемножим эти числа, представленные прямыми двоичными кодами, так же, как это делается в десятичной системе.

$[A_2]п = 111$ — множимое
$\times \quad \times$
$[B_2]п = 101$ — множитель

111 — множимое (сдвиг на 0 разрядов)
$+\ \cancel{000}$ — умножение на 0 (сдвиг на 1 разряд)
111 — множимое (сдвиг на 2 разряда)

$[C_2]п = 100011$ — произведение
$C_{10} = 35$

Нетрудно видеть, что произведение получается путем сложения частных произведений, представляющих собой разряды множимого, сдвинутые влево в соответствии с позициями разрядов множителя. Частные произведения, полученные умножением на нуль, игнорируются. Важной особенностью операции умножения n-разрядных сомножителей является увеличение разрядности произведения до $n+n=2n$. Знак произведения формируется путем сложения знаковых разрядов сомножителей. Возможные переносы из знакового разряда игнорируются.

Деление. Операция деления, как и в десятичной арифметике, является обратной операции умножения. Покажем, что и эта операция приводится к последовательности операций сложения и сдвига.

Пример 2.12. Разделить два числа $A_{10} = 45$; $B_{10} = 5$.

$[A_2]п = 101101$
$[B_2]п = 101$

Делимое Делитель

$101101 \mid 101$
$-101 \quad 1001$ — частное
0101
$-\ 101$
0

$[C_2]п = 1001$
$C_{10} = 9$

Деление произведено так же, как это делается обычно в десятичной системе. Сначала проверяется, можно ли вычесть значение делителя из старших разрядов делимого. Если возможно, то в разряде частного записывается единица и определяется частная разница. В противном случае в частное записывается нуль и разряды делителя сдвигаются вправо на один разряд по отношению к разрядам делимого. К полученной предыдущей разнице сносится очередная цифра делимого, и данный процесс повторяется до тех пор, пока не будет получена необходимая точность. Если учесть, что все вычитания в ЭВМ заменяются сложением в ОК или ДК (см. табл. 2.3), то действительно операция деления приводится к операциям сложения и сдвигам вправо разрядов делителя относительно разрядов делимого. Отметим, что делимое перед операцией деления должно быть приведено к $2n$-разрядной сетке. Только в этом случае при делении на n-разрядный делитель получается n-разрядное частное.

Знак частного формируется также путем сложения знаковых разрядов делимого и делителя, как это делалось при умножении.

2.3.3. Арифметические операции над двоичными числами с плавающей точкой

В современных ЭВМ числа с плавающей точкой хранятся в памяти машин, имея мантиссу и порядок (характеристику) в прямом коде и нормализованном виде. Все арифметические действия над этими числами выполняются так же, как это делается с ними, если они представлены в полулогарифмической форме (мантисса и десятичный порядок) в десятичной системе счисления. Порядки и мантиссы обрабатываются раздельно.

Сложение (вычитание). Операция сложения (вычитания) производится в следующей последовательности.

1. Сравниваются порядки (характеристики) исходных чисел путем их вычитания $\Delta p = p_1 - p_2$. При выполнении этой операции определяется, одинаковый ли порядок имеют исходные слагаемые.

2. Если разность порядков равна нулю, то это значит, что одноименные разряды мантисс имеют одинаковые веса (двоичный порядок). В противном случае должно проводиться выравнивание порядков.

3. Для выравнивания порядков число с меньшим порядком сдвигается вправо на разницу порядков Δp. Младшие выталкиваемые разряды при этом теряются.

4. После выравнивания порядков мантиссы чисел можно складывать (вычитать) в зависимости от требуемой операции. Операция вычитания заменяется операцией сложения в соответствии с данными табл. 2.3. Действия над слагаемыми производятся в ОК или ДК по общим правилам.

5. Порядок результата берется равным большему порядку.

6. Если мантисса результата не нормализована, то осуществляются нормализация и коррекция значений порядка.

Пример 2.13. Сложить два числа: $A_{10} = +1.375$; $B_{10} = -0.625$.

$A_2 = +1.011 = 0 \vdots 1011 \cdot 10^1$; $B_2 = -0.101 = -0 \vdots 101 \cdot 10^0$.

В нормализованном виде эти числа будут иметь вид:

Порядок	Мантисса
$[A_2]п = 0 \vdots 1$	$0 \vdots 1011$
	знак числа
$[B_2]п = 0 \vdots 0$	$1 \vdots 101$

1. Вычитаем порядки $\Delta p = p_1 - p_2 = 1 - 0 = 1$. В машине эта операция требует операции сложения с преобразованием порядка чисел в дополнительный код:

$$p_1 = 0 \vdots 1 \qquad [p_1]дк = 0 \vdots 1$$
$$p_2 = 0 \vdots 0 \quad + \quad [p_2]дк = 0 \vdots 0$$
$$\Delta p = 0 \vdots 1$$

Определяем, что $\Delta p \neq 0$.

2. Порядок первого числа больше порядка второго числа на единицу. Требуется выравнивание порядков.

3. Для выравнивания порядков необходимо второе число сдвинуть вправо на один разряд:

$[B_2]исх = 0 \vdots 0 \quad 1 \vdots 101$

после сдвига

$[B_2']п = 0 \vdots 1 \quad 1 \vdots 0101$

$[m_B']дк = 1 \vdots 1011$

4. Складываем мантиссы:

$$[m_A]дк = 0 \vdots 1011$$
$$+ \; [m_B']дк = 1 \vdots 1011$$
$$[m_C]дк = 0 \vdots 0110$$

Мантисса числа C — положительная.

5. Порядок числа C равен порядку числа с бóльшим порядком, т.е. $p_c = +1$.

$$[C_2]п = 0 \vdots 1 \quad 0 \vdots 0110.$$

Видно, что мантисса результата не нормализована, так как старшая цифра мантиссы равна нулю.

6. Нормализуем результат путем сдвига мантиссы на один разряд влево и соответственно вычитаем из значения порядка единицу:

$$[C_2]п = 0 \vdots 0 \quad 0 \vdots 110;$$
$$C_{10} = +0.75.$$

Умножение (деление). Операция умножения (деления) чисел с плавающей точкой также требует разных действий над порядками и мантиссами. Алгоритмы этих операций выполняются в следующей последовательности.

1. При умножении (делении) порядки складываются (вычитаются) так, как это делается над числами с фиксированной точкой.

2. При умножении (делении) мантиссы перемножаются (делятся).

3. Знаки произведения (частного) формируются путем сложения знаковых разрядов сомножителей (делимого и делителя). Возможные переносы из знакового разряда игнорируются.

2.3.4. Арифметические операции над двоично-десятичными кодами чисел

При обработке больших массивов экономической информации переводы чисел из десятичной системы в двоичную и обратно могут требовать значительного машинного времени. Некоторые образцы ЭВМ поэтому имеют или встроенные, или подключаемые блоки, которые обрабатывают десятичные целые числа в их двоично-десятичном представлении. Действия над ними также приводятся к операции алгебраического сложения отдельных цифр чисел, представленных дополнительными кодами в соответствии с табл. 2.3.

Существует несколько алгоритмов сложения двоично-десятичных кодов десятичных чисел. Приведем один из алгоритмов сложения, который получил довольно широкое распространение.

1. Сложение чисел начинается с младших цифр (тетрад) и производится с учетом возникающих переносов из младших разрядов в старшие.

2. Знак суммы формируется специальной логической схемой по знаку большего слагаемого.

3. Для того чтобы при сложении двоично-десятичных цифр возникали переносы, аналогичные при сложении чисел в десятичном представлении, необходимо проводить так называемую десятичную коррекцию. Для этого к каждой тетраде первого числа прибавляется дополнительно по цифре 6_{10}=0110_2, что позволяет исключить шесть неиспользуемых комбинаций $(1010—1111)_2$, так как они кодируют шестнадцатеричные цифры A—F (числа $10—15_{10}$).

4. После операции суммирования осуществляется корректировка суммы. Из тех тетрад суммы, из которых не было переносов, изымаются ранее внесенные избытки 6_{10}=0110_2. Для этого проводится вторая коррекция. Операция вычитания заменяется, как и обычно, операцией сложения с числом –6, представленным дополнительным кодом 1010_2, но только в тех разрядах, в которых отсутствовали переносы. При этой второй коррекции переносы из тетрад блокируются.

5. Операция вычитания реализуется достаточно своеобразно. По общему правилу сложения (см. п. 1—4) к тетрадам числа с бо́льшим модулем прибавляются дополнительные коды тетрад другого числа. При этом первая коррекция не проводится, так как в дополнениях тетрад она учитывается автоматически. Знак результата определяется по знаку числа с бо́льшим модулем.

Пример 2.14. Сложить два числа $A_{10}=177$; $B_{10}=418$.

A_{2-10}	00001	0111	0111	
+				1-я коррекция
	0110	0110	0110	
A'	0111	1101	1101	
+				Сложение $A'+B$
B_{2-10}	0100	0001	1000	
	1011	1111	0101	результат C'
+				
	1010	1010		2-я коррекция
C_{2-10}	0101	1001	0101	результат

$C_{10}=595$

2.4. Логические основы ЭВМ

2.4.1. Основные сведения из алгебры логики

Теоретической основой построения ЭВМ являются специальные математические дисциплины. Одной из них является алгебра логики, или булева алгебра (Дж. Буль — английский математик прошлого столетия, основоположник этой дисциплины). Ее аппарат широко используют для описания схем ЭВМ, их оптимизации и проектирования.

Вся информация в ЭВМ представляется в двоичной системе счисления. Поставим в соответствие входным сигналам отдельных устройств ЭВМ значения переменных $x_i (i=\overline{1,n})$, а выходным сигналам — значения функций $y_j (j=\overline{1,m})$ (рис. 2.1).

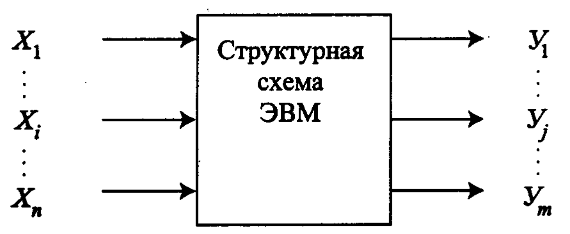

Рис. 2.1. Представление схемы ЭВМ