4.4. ПРЕОБРАЗОВАНИЕ ИЗ ОДНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ

Если бы наши предки, изобретая арифметику, вели счет при помощи двух рук или восьми пальцев, а не десяти "цифр", у нас никогда бы не было хлопот с разработкой программ двоично-десятичного преобразования чисел. (И, возможно, мы бы никогда столько не узнали о системах счисления.) В этом разделе будут рассмотрены вопросы, связанные с преобразованием чисел из позиционной системы счисления по одному основанию в позиционную систему счисления по другому основанию. Конечно, этот процесс наиболее важен для двоичных компьютеров при преобразовании входных данных, представленных в десятичном формате, в двоичный формат и преобразовании результатов из двоичного формата обратно в десятичный.

А. Четыре основных метода. Преобразование чисел из двоичной системы счисления в десятичную и обратно — одна из наиболее машинно-зависимых операций, поскольку разработчикам компьютеров приходится постоянно изобретать различные способы аппаратной реализации этой операции. В связи с этим далее будут рассматриваться только основные принципы решения данной задачи, на основании которых программисты могут выбирать процедуры, наиболее подходящие для реализации на компьютере конкретной конфигурации.

Будем предполагать, что операции преобразований выполняются только с неотрицательными числами, так как манипулирование знаками легко учесть.

Предположим, что выполняется преобразование из системы счисления по основанию b в систему счисления по основанию B. (Обобщение для систем счисления со смешанным основанием рассматривается в упр. 1 и 2.) Большинство процедур преобразования из одного основания в другое основано на операциях умножения и деления, использующих один из четырех методов, которые описываются ниже. Два первых метода применяются для преобразования целых чисел (разделяющая точка расположена справа), а два других — для дробных частей чисел (разделяющая точка расположена слева). Чаще всего нельзя *точно* выразить конечную дробную часть по основанию b $(0.u_{-1}u_{-2}\ldots u_{-m})_b$ в виде конечной дробной части по основанию B $(0.U_{-1}U_{-2}\ldots U_{-M})_B$. Например, десятичная дробь $\frac{1}{10}$ в двоичном формате.представляется бесконечной дробью $(0.0001100110011\ldots)_2$. В связи с этим приходится иногда округлять результат до M разрядов.

Метод 1, a (Деление на B с использованием представления чисел в формате по основанию b). Для заданного целого числа u можно получить представление в формате по основанию B вида $(\ldots U_2U_1U_0)_B$, выполняя

$$U_0 = u \bmod B, \qquad U_1 = \lfloor u/B \rfloor \bmod B, \qquad U_2 = \lfloor \lfloor u/B \rfloor /B \rfloor \bmod B, \qquad \ldots$$

до тех пор, пока не окажется $\lfloor \ldots \lfloor \lfloor u/B \rfloor /B \rfloor \ldots /B \rfloor = 0$.

Метод 1, b (Умножение на B с использованием представления чисел в формате по основанию b). Если представление числа u по основанию b имеет вид $(u_m\ldots u_1u_0)_b$, то можно, воспользовавшись арифметическими операциями с числами, которые представлены в формате по основанию B, получить полином $u_mb^m+\cdots+u_1b+u_0 = u$ в виде

$$\bigl((\ldots(u_m b + u_{m-1})\, b + \cdots)\, b + u_1\bigr)\, b + u_0.$$

Метод 2, а (Умножение на b с использованием представления чисел в формате по основанию B). Для данного дробного числа u можно вычислить значения разрядов $(.U_{-1}U_{-2}\ldots)_B$ его представления по основанию B следующим образом:

$$U_{-1} = \lfloor uB \rfloor, \qquad U_{-2} = \lfloor \{uB\}B \rfloor, \qquad U_{-3} = \lfloor \{\{uB\}B\}B \rfloor, \qquad \ldots,$$

где $\{x\}$ означает $x \bmod 1 = x - \lfloor x \rfloor$. Чтобы округлить результат до M разрядов, вычисления можно прервать после получения U_{-M}, причем если $\{\ldots\{\{uB\}B\}\ldots B\}$ больше $\frac{1}{2}$, то значение U_{-M} следует увеличить на единицу. (Заметим, однако, что эта операция может привести к необходимости выполнения переносов, которые должны быть при помощи арифметических операций по основанию B включены в результат. Было бы проще перед началом вычислений прибавить к исходному числу u константу $\frac{1}{2}B^{-M}$, но это может привести к неправильному результату, если в компьютере число $\frac{1}{2}B^{-M}$ не может быть точно представлено в формате по основанию b. Заметим также, что возможно округление результата до $(1.00\ldots0)_B$, если $b^m \geq 2B^M$.)

В упр. 3 рассматривается обобщение этого метода на случай *переменного* M, достаточно большого для представления исходного числа с заданной точностью. В такой ситуации проблема переносов не возникает.

Метод 2, b (Деление на b с использованием представления чисел в формате по основанию B). Если число u представлено по основанию b в виде $(0.u_{-1}u_{-2}\ldots u_{-m})_b$, то можно, используя арифметические операции по основанию B, вычислить $u_{-1}b^{-1} + u_{-2}b^{-2} + \cdots + u_{-m}b^{-m}$ в виде

$$\bigl((\ldots(u_{-m}/b + u_{1-m})/b + \cdots + u_{-2})/b + u_{-1}\bigr)/b.$$

Необходимо внимательно следить за погрешностями, возникающими при усечениях или округлениях во время выполнения операции деления на b; они, как правило, незначительны, но это бывает не всегда.

Подведем итоги. Методы 1, a; 1, b; 2, a и 2, b предоставляют по два возможных способа преобразования целых и дробных чисел. И, конечно, существует возможность выполнять преобразования целых чисел в дробные и наоборот путем умножения или деления на соответствующую степень b или B. Поэтому для выполнения преобразования имеется на выбор по крайней мере четыре метода.

B. Преобразование с однократной точностью. Чтобы проиллюстрировать эти методы, предположим, что MIX — двоичный компьютер и нужно преобразовать неотрицательное целое число u, представленное в двоичном формате, в десятичный формат, т. е. получить $b = 2$ и $B = 10$. Метод 1, a можно запрограммировать следующим образом.

```
    ENT1 0           Присвоить j ← 0.
    LDX  U
    ENTA 0           Присвоить rAX ← u.
 1H DIV  =10=        (rA, rX) ← (⌊rAX/10⌋, rAX mod 10).
    STX  ANSWER,1    U_j ← rX.                                  (1)
    INC1 1           j ← j + 1.
    SRAX 5           rAX ← rA.
    JXP  1B          Повторять до тех пор,
                        пока в результате не получим нуль. ▮
```

Для вычисления M разрядов необходимо затратить $18M + 4$ циклов.

В методе 1, а используется операция деления на 10. В методе 2, а используется *умножение* на 10, так что программа, реализующая этот метод, может выполняться немного быстрее. Но, применяя метод 2, а, нужно иметь дело с дробями, а это приводит к интересной ситуации. Пусть w — размер машинного слова, и положим, что $u < 10^n < w$. При помощи одного деления можно найти q и r, где

$$wu = 10^n q + r, \qquad 0 \le r < 10^n. \tag{2}$$

Если же применить метод 2, а к дроби $(q+1)/w$, то за n шагов получим разряды числа u слева направо, так как

$$\left\lfloor 10^n \,\frac{q+1}{w} \right\rfloor = \left\lfloor u + \frac{10^n - r}{w} \right\rfloor = u. \tag{3}$$

(Эта идея предложена П. А. Саметом (P. A. Samet) и опубликована в журнале *Software Practice & Experience* **1** (1971), 93–96.)

Приведем соответствующую MIX-программу.

	JOV	OFLO	Удостовериться в отсутствии переполнения.
	LDA	U	
	LDX	$=10^n=$	$\mathrm{rAX} \leftarrow wu + 10^n$.
	DIV	$=10^n=$	$\mathrm{rA} \leftarrow q+1$, $\mathrm{rX} \leftarrow r$.
	JOV	ERROR	Переход при $u \ge 10^n$.
	ENT1	n-1	Присвоить $j \leftarrow n-1$.
2H	MUL	=10=	Представим, что разделяющая точка находится слева, $\mathrm{rA} = x$.
	STA	ANSWER,1	Присвоить $U_j \leftarrow \lfloor 10x \rfloor$.
	SLAX	5	$x \leftarrow \{10x\}$.
	DEC1	1	$j \leftarrow j-1$.
	J1NN	2B	Повторить для $n > j \ge 0$. ▌

(4)

Данная программа немного длиннее предыдущей, и на ее выполнение требуется $16n + 19$ циклов, так что при $n = M \ge 8$ она выполняется быстрее программы (1). Однако при наличии ведущих нулей программа (1) будет выполняться быстрее.

Программа (4) в представленном виде при $10^m < w < 10^{m+1}$ не может использоваться для преобразования целых чисел $u \ge 10^m$, так как нужно принять $n = m+1$. В этом случае ведущий разряд числа u можно получить, вычисляя $\lfloor u/10^m \rfloor$; после этого можно преобразовать число $u \bmod 10^m$ по приведенному выше методу при $n = m$.

То обстоятельство, что разряды результата могут быть получены слева направо, может оказаться полезным в некоторых приложениях (например, при последовательном выводе разрядов результата на печать). Итак, метод, применимый для дробей, можно использовать и для преобразований целых чисел, хотя, применив неточное деление при преобразованиях, придется выполнять численный анализ метода.

В методе 1, а деление на 10 можно заменить двумя умножениями. Это может оказаться существенным, поскольку преобразование оснований часто выполняется в компьютерах-"сателлитах", в которых отсутствует встроенная процедура деления. Если предположить, что x — приближение к $\frac{1}{10}$, так что

$$\frac{1}{10} < x < \frac{1}{10} + \frac{1}{w},$$

то легко доказать (см. упр. 7), что $\lfloor ux \rfloor = \lfloor u/10 \rfloor$ или $\lfloor u/10 \rfloor + 1$, пока $0 \le u < w$. Поэтому при вычислении $u - 10\lfloor ux \rfloor$ можно определить значение $\lfloor u/10 \rfloor$:

$$\lfloor u/10 \rfloor = \lfloor ux \rfloor - \bigl[u < 10\lfloor ux \rfloor\bigr]. \tag{5}$$

Одновременно можно определить $u \bmod 10$. MIX-программа, выполняющая преобразование с использованием (5), приведена в упр. 8. На вычисление каждого разряда она затрачивает около 33 циклов.

Метод 1, a может быть с успехом применен и в компьютерах, в которых отсутствуют встроенные команды как деления, так и умножения, путем выполнения операций сдвига и сложения, как продемонстрировано в упр. 9.

Другой способ преобразования из двоичного представления в десятичное заключается в использовании метода 1, b, но при этом необходимо имитировать удвоение в *десятичной* системе счисления. Этот способ в общем случае наиболее подходит для аппаратной реализации в компьютере; тем не менее процесс удвоения в десятичной системе счисления можно и запрограммировать при помощи операций двоичного сложения, двоичного сдвига и двоичного извлечения или маскирования (побитовое AND), как показано в табл. 1, составленной Петером Л. Монтгомери (Peter L. Montgomery).

Таблица 1

УДВОЕНИЕ ДЕСЯТИЧНОГО ЧИСЛА,
ЗАКОДИРОВАННОГО В ДВОИЧНОЙ СИСТЕМЕ

Операция	*Общий вид*	*Пример*
1. Задать число	$u_{11}\,u_{10}\,u_9\,u_8\;\;u_7\,u_6\,u_5\,u_4\;\;u_3\,u_2\,u_1\,u_0$	0011 0110 1001 = 3 6 9
2. Прибавить 3 к каждому разряду	$v_{11}\,v_{10}\,v_9\,v_8\;\;v_7\,v_6\,v_5\,v_4\;\;v_3\,v_2\,v_1\,v_0$	0110 1001 1100
3. Извлечь каждый старший бит	$v_{11}\;0\;0\;0\;\;v_7\;0\;0\;0\;\;v_3\;0\;0\;0$	0000 1000 1000
4. Сдвинуть вправо на 2 разряда и вычесть	$0\;v_{11}\;v_{11}\;0\;\;0\;v_7\;v_7\;0\;\;0\;v_3\;v_3\;0$	0000 0110 0110
5. Прибавить исходное число	$w_{11}w_{10}w_9w_8\;\;w_7w_6w_5w_4\;\;w_3w_2w_1w_0$	0011 1100 1111
6. Прибавить исходное число	$x_{12}\,x_{11}\,x_{10}\,x_9\,x_8\;\;x_7\,x_6\,x_5\,x_4\;\;x_3\,x_2\,x_1\,x_0$	0 0111 0011 1000 = 7 3 8

При помощи этого метода значение каждого разряда d заменяется на $2d$ при $0 \le d \le 4$ и на $6 + 2d = (2d - 10) + 2^4$ при $5 \le d \le 9$, что и требуется для удвоения десятичных чисел, каждый разряд которых закодирован четырьмя битами.

Другая идея состоит в том, чтобы хранить таблицу степеней двоек в десятичном формате и складывать соответствующие степени, имитируя операции десятичного сложения. В разделе 7.1 описываются приемы оперирования битами.

И наконец, для преобразования целых чисел, представленных в двоичном формате, в целые числа, представляемые в десятичном формате, можно использовать даже метод 2, b. Можно найти q, как в (2), а затем имитировать операцию десятичного деления $q+1$ на w, используя процесс "уполовинивания" (упр. 10), аналогичный описанному выше процессу удваивания, сохраняя при этом в результате только первые n разрядов справа от разделяющей точки. Похоже, что в этом случае метод 2, b не имеет каких-либо преимуществ по сравнению с остальными тремя методами, проанализированными выше, тем не менее подтверждается высказанный выше тезис о том, что для преобразования целых чисел из одной системы счисления в другую существует четыре различных метода.

Теперь рассмотрим преобразование из десятичной системы счисления в двоичную (т. е. $b = 10$, $B = 2$). Метод 1, a позволяет имитировать операцию десятичного деления на 2, что допустимо, но предпочтительнее реализовать ее аппаратно, а не программно (см. упр. 10).

В большинстве случаев наиболее удобным методом преобразования из десятичной системы счисления в двоичную является метод 1, b. В приводимой ниже MIX-программе принято, что число $(u_m \ldots u_1 u_0)_{10}$ содержит по крайней мере два разряда, подлежащих преобразованию, и неравенство $10^{m+1} < w$ выполняется так, чтобы не возникало переполнение.

```
   ENT1 M-1       Присвоить j ← m − 1.
   LDA  INPUT+M   Присвоить U ← u_m.
1H MUL  =10=
   SLAX 5                                          (6)
   ADD  INPUT,1   U ← 10U + u_j.
   DEC1 1
   J1NN 1B        Повторить при m > j ≥ 0. ▮
```

Операция умножения на 10 может быть заменена операциями сдвига и сложения.

В упр. 19 рассмотрен более быстрый, возможно, способ преобразования, в котором вместо $m - 1$ операций умножения используется примерно $\lg m$ операций умножения, маскирования и сложения.

Для преобразования в двоичный формат десятичных дробей $(0.u_{-1}u_{-2}\ldots u_{-m})_{10}$ можно воспользоваться методом 2, b или, в более общем случае, сначала преобразовать целое число $(u_{-1}u_{-2}\ldots u_{-m})_{10}$ при помощи метода 1, a, а затем разделить результат на 10^m.

C. Вычисления вручную. Иногда в процессе программирования возникает необходимость выполнить преобразование чисел вручную, а поскольку в обычных школах этому пока что не учат, имеет смысл кратко обсудить здесь этот вопрос. Известны простые методы преобразования чисел из десятичного формата в восьмеричный и обратно, выполняемые вручную; этим методам легко научиться, так что они должны стать известными более широко.

Преобразование целых чисел из восьмеричной системы счисления в десятичную. Простейшим является преобразование из восьмеричного формата в десятичный. Этот способ, по-видимому, впервые опубликовал Уолтер Соден (Walter Soden) в *Math. Comp.* **7** (1953), 273–274. Чтобы выполнить преобразование, нужно записать данное восьмеричное число, удвоить на k-м шаге k ведущих разрядов, используя десятичную арифметику, и вычесть полученный результат из $(k+1)$ ведущих разрядов при помощи опять же десятичной арифметики. Если заданное число содержит $(m+1)$ разрядов, то процесс прекращается через m шагов. Удачной оказалась идея ввода разделяющей точки для того, чтобы выделить удваиваемые разряды, как показано в следующем примере. Это помогает исключить возможные ошибки.

Пример 1. Преобразовать число $(\mathit{5325121})_8$ в десятичный формат.

$$
\begin{array}{c|ccccccc}
 & 5. & \mathit{3} & \mathit{2} & \mathit{5} & \mathit{1} & \mathit{2} & \mathit{1}\\
- & 1 & 0 & & & & & \\
\cline{2-3}
 & 4 & 3. & \mathit{2} & \mathit{5} & \mathit{1} & \mathit{2} & \mathit{1}\\
- & & 8 & 6 & & & & \\
\cline{2-4}
 & 3 & 4 & 6. & \mathit{5} & \mathit{1} & \mathit{2} & \mathit{1}\\
- & & 6 & 9 & 2 & & & \\
\cline{2-5}
 & 2 & 7 & 7 & 3. & \mathit{1} & \mathit{2} & \mathit{1}\\
- & & 5 & 5 & 4 & 6 & & \\
\cline{2-6}
 & 2 & 2 & 1 & 8 & 5. & \mathit{2} & \mathit{1}\\
- & & 4 & 4 & 3 & 7 & 0 & \\
\cline{2-7}
 & 1 & 7 & 7 & 4 & 8 & 2. & \mathit{1}\\
- & & 3 & 5 & 4 & 9 & 6 & 4\\
\cline{2-8}
 & 1 & 4 & 1 & 9 & 8 & 5 & 7
\end{array}
$$

Результат: $(1419857)_{10}$.

Достаточно надежный способ проверки вычислений состоит в "выбрасывании девяток": сумма разрядов десятичного числа должна быть конгруэнтной по модулю 9 попеременно сумме и разности разрядов числа, представленного в восьмеричном формате, причем крайний справа разряд последнего числа берется со знаком "плюс". В вышеприведенном примере имеем $1+4+1+9+8+5+7=35$ и $1-2+1-5+2-3+5=-1$; разность равна 36 (кратна 9). Если проверка дала отрицательный результат, то она повторяется с $(k+1)$ ведущими разрядами после k-го шага и местоположение ошибки определяется при помощи процедуры двоичного поиска. Другими словами, можно определить, где произошла ошибка, начав с проверки среднего результата, и затем в зависимости от того, верен ли результат, применить ту же процедуру к первой или второй части вычислений.

Ввиду того, что существует один шанс из девяти, что два *случайных* целых числа будут отличаться по модулю девять, надежность процедуры выбрасывания девяток равна только 89%. Более надежный способ проверки — преобразовать результат обратно в восьмеричный формат с использованием обратного метода, который и будет сейчас рассмотрен.

Преобразование целых чисел, представленных в десятичном формате, в восьмеричный формат. Для выполнения обратного преобразования можно использовать аналогичную процедуру. Запишем заданное в десятичном формате число, удвоим на k-м шаге k ведущих разрядов, используя представление в *восьмеричном* формате, и *сложим* полученные $(k+1)$ ведущих разрядов, опять же используя представление в *восьмеричном* формате. Для заданного числа, содержащего $(m+1)$ разрядов, процесс заканчивается через m шагов.

Пример 2. Преобразовать число $(1419857)_{10}$ в восьмеричный формат.

$$
\begin{array}{cccccccc}
 & \mathit{1}. & 4 & 1 & 9 & 8 & 5 & 7 \\
+ & & \mathit{2} & & & & & \\ \cline{2-3}
 & \mathit{1} & \mathit{6}. & 1 & 9 & 8 & 5 & 7 \\
+ & & \mathit{3} & \mathit{4} & & & & \\ \cline{2-4}
 & \mathit{2} & \mathit{1} & \mathit{5}. & 9 & 8 & 5 & 7 \\
+ & & \mathit{4} & \mathit{3} & \mathit{2} & & & \\ \cline{2-5}
 & \mathit{2} & \mathit{6} & \mathit{1} & \mathit{3}. & 8 & 5 & 7 \\
+ & & \mathit{5} & \mathit{4} & \mathit{2} & \mathit{6} & & \\ \cline{2-6}
 & \mathit{3} & \mathit{3} & \mathit{5} & \mathit{6} & \mathit{6}. & 5 & 7 \\
+ & & \mathit{6} & \mathit{7} & \mathit{3} & \mathit{5} & \mathit{4} & \\ \cline{2-7}
 & \mathit{4} & \mathit{2} & \mathit{5} & \mathit{2} & \mathit{4} & \mathit{1}. & 7 \\
+ & \mathit{1} & \mathit{0} & \mathit{5} & \mathit{2} & \mathit{5} & \mathit{0} & \mathit{2} \\ \cline{2-8}
 & \mathit{5} & \mathit{3} & \mathit{2} & \mathit{5} & \mathit{1} & \mathit{2} & \mathit{1}
\end{array}
\qquad \textit{Результат: } (\mathit{5325121})_8.
$$

(Заметим, что при вычислении восьмеричного представления присутствуют невосьмеричные цифры 8 и 9.) Проверка результата может быть выполнена описанным выше способом. Этот метод был опубликован Шарлем П. Розье (Charles P. Rozier), *IEEE Trans.* **CE-11** (1962), 708–709.

Обе описанные процедуры представляют собой, по существу, вариации метода 1, b обобщенного преобразования из одной системы счисления в другую. Операции удваивания и вычитания в десятичной системе счисления подобны умножению на $10 - 2 = 8$. Операции удваивания и вычитания в восьмеричной системе счисления подобны умножению на $8 + 2 = 10$. Аналогичный метод существует и для преобразования из шестнадцатеричной системы счисления в десятичную, но это преобразование выполняется немного сложнее, так как вместо операции умножения на 2 оно включает операцию умножения на 6.

Чтобы запомнить оба этих метода, нужно уяснить, что при переводе числа из восьмеричной системы счисления в десятичную выполняется *вычитание*, так как представление чисел в десятичном формате короче, чем в восьмеричном. Аналогично при переводе числа из представления в десятичном формате в восьмеричный необходимо выполнять *сложение*. Вычисления выполняются в формате представления *результата*, а не в исходном формате представления числа; в противном случае требуемый результат получен не будет.

Преобразование дробей. Аналогичные методы, пригодные для столь же быстрого преобразования вручную дробей, неизвестны. Похоже, что наилучшим является метод 2, а, в котором с целью упрощения операций умножения на 10 или 8 выполняются операции удвоения и сложения. В этом случае критерий выбора сложения и/или вычитания меняется на обратный — при преобразовании чисел в десятичный формат выполняется сложение, а при преобразовании чисел в восьмеричный формат выполняется вычитание. Кроме того, при вычислениях используется формат, в котором представлено исходное число, а *не* формат, в котором подставляется результат (см. приведенные ниже примеры 3 и 4). Реализация этого процесса требует примерно в два раза больше вычислений, чем рассмотренный выше метод преобразования целых чисел.

Пример 3. Преобразование числа $(.14159)_{10}$ в восьмеричный формат.

```
 1 4 1 5 9
   2 8 3 1 8−
 ───────────
 1.1 3 2 7 2
     2 6 5 4 4−
   ───────────
   1.0 6 1 7 6
       1 2 3 5 2−
     ───────────
     0.4 9 4 0 8
         9 8 8 1 6−
       ───────────
       3.9 5 2 4 6
         1 9 0 5 2 8−
         ───────────
         7.6 2 1 1 2
           1 2 4 2 2 4−
           ───────────
           4.9 6 8 9 6
```

Результат: $(.110374\ldots)_8$.

Пример 4. Преобразование числа $(.110374)_8$ в десятичный формат.

```
 .1 1 0 3 7 4
   2 2 0 7 7 0+
 ─────────────
 1.3 2 4 7 3 0
     6 5 1 6 6 0+
   ─────────────
   4.1 2 1 1 6 0
       2 4 2 3 4 0+
     ─────────────
     1.4 5 4 1 4 0
       1 1 3 0 3 0 0+
       ─────────────
       5.6 7 1 7 0 0
         1 5 6 3 6 0 0+
         ─────────────
         8.5 0 2 6 0 0
           1 2 0 5 4 0 0+
           ─────────────
           6.2 3 3 4 0 0
```

Результат: $(.141586\ldots)_{10}$.

D. Преобразование чисел с плавающей точкой. При выполнении преобразований чисел с плавающей точкой необходимо одновременно выполнять операции как с целой частью числа, так и с дробной, поскольку преобразование целой части числа оказывает влияние на дробную часть. Для преобразования числа $f \cdot 2^e$ в десятичный формат можно сначала представить 2^e в виде $F \cdot 10^E$ (обычно при помощи вспомогательных таблиц) и затем уже преобразовать Ff в десятичный формат. Аналогично можно умножить e на $\log_{10} 2$ и затем округлить результат до ближайшего целого числа E; после этого разделить $f \cdot 2^e$ на 10^E и преобразовать результат. Обратно, для преобразования числа $F \cdot 10^E$ в двоичный формат можно преобразовать F, а затем умножить его на число 10^E, представленное в формате с плавающей точкой (снова используя вспомогательные таблицы). Для уменьшения максимальных размеров вспомогательных таблиц используется обычная методика, основанная на применении нескольких операций умножения и/или деления, хотя это может привести к распространению ошибки вследствие округления промежуточных результатов. Вопросы минимизации таких ошибок рассмотрены в упр. 17.

E. Преобразование с многократной точностью. Начинать преобразование очень длинных чисел удобнее всего с преобразования блоков разрядов, операции с которыми можно выполнять с однократной точностью. Затем следует объединить эти блоки, пользуясь простыми способами, которые специфичны для многократной точности. Например, пусть 10^n — наивысшая степень 10, меньшая, чем размер машинного слова. Тогда:

a) чтобы преобразовать *целое число* с многократной точностью из двоичного формата в десятичный, необходимо многократно разделить его на 10^n (выполняя таким образом преобразование из двоичной системы счисления в десятичную с основанием 10^n по методу 1, a); при помощи операций с однократной точностью получим n десятичных разрядов для каждой единицы представления в системе счисления с основанием 10^n;

b) чтобы преобразовать *дробную часть* числа с многократной точностью из двоичного формата в десятичный, поступим подобным образом, умножив его на 10^n (т. е. использовав метод 2, a, где $B = 10^n$);

c) чтобы преобразовать целое число с многократной точностью из десятичной системы счисления в двоичную, преобразуем сначала блоки по n разрядов; затем для перехода из системы счисления с основанием 10^n в двоичный формат используем метод 1, b;

d) для преобразования дробной части с многократной точностью из представления в десятичном формате в двоичный сначала выполним преобразование в систему с основанием 10^n, как и в процедуре (c), а затем используем метод 2, b.

F. История и библиография. Приемы преобразования чисел из одной системы счисления в другую использовались еще в древности в задачах, связанных с мерами, весами и деньгами, когда обычно приходилось иметь дело с системами счисления со смешанными основаниями. Эти преобразования обычно выполнялись с помощью вспомогательных таблиц. В 17 веке, когда шестидесятеричные дроби были вытеснены десятичными, возникла необходимость выполнять преобразование из одной системы счисления в другую с тем, чтобы можно было пользоваться имеющимися книгами астрономических таблиц. В 1667 году в книге под редакцией Вильяма Отреда (William Oughtred) *Clavis Mathematicæ* (см. гл. 6, раздел 18) был дан систематический метод перевода дробей из системы счисления с основанием 60 в систему счисления с основанием 10. (В первом издании книги Отреда 1631 года этот материал отсутствовал.) Правила преобразований из одной системы счисления в другую были сформулированы еще аль-Каши из Самарканда в его работе *Ключ к арифметике* (1427), в которой ясно изложены методы 1, a; 1, b и 2, a [*Историко-математические исследования* **7** (1954), 126–135], но в Европе его работа была неизвестна. В 18 веке американский математик Хью Джонс (Hugh Jones) для описания правил преобразования из восьмеричной системы счисления в десятичную ввел термины “octavation” (октавирование) и “decimation” (децимирование), но его методы оказались столь же неясными, как и его терминология. А. М. Лежандр заметил, что положительные целые числа могут быть легко преобразованы в двоичный формат с помощью повторного деления на 64 [*Théorie des Nombres* (Paris, 1798), 229].

В 1946 году Г. Г. Гольдстайн (H. H. Goldstine) и Дж. фон Нейман (J. von Neumann) в своем классическом сочинении *Planning and Coding Problems for an Electronic Computing Instrument* дали глубокий анализ проблемы преобразований из одной системы счисления в другую в связи с необходимостью обоснования применимости двоичной арифметики. (См. John von Neumann, *Collected Works* **5** (New York: Macmillan, 1963), 127–142.) Другая ранняя работа, посвященная преобразованию чисел из одной системы счисления в другую в двоичных компьютерах, была опубликована Ф. Кунс (F. Koons) и С. Лубкиным (S. Lubkin) в журнале *Math. Comp.* **3** (1949), 427–431, где они предложили не совсем обычный метод преобразований. Несколько позже Ф. Л. Бауэр (F. L. Bauer) и К. Замельсон (K. Samelson) опубликовали первое обсуждение проблемы преобразования чисел с плавающей точкой [*Zeit. für angewandte Math. und Physik* **4** (1953), 312–316].

Исторический интерес представляют также заметки Г. Т. Лэйка (G. T. Lake) [*CACM* **5** (1962), 468–469], в которых описываются методы аппаратной реализации преобразований и приводятся понятные примеры, и статья А. Х. Строуда (A. H. Stroud) и Д. Секреста (D. Secrest) [*Comp. J.* **6** (1963), 62–66], в которой рассмотрено преобразование чисел, заданных с многократной точностью. Преобразования *ненормализованных* чисел с плавающей точкой, сохраняющие соответствующую представлению "значимость", были рассмотрены Г. Кэннером (H. Kanner) в *JACM* **12** (1965), 242–246, и Н. Метрополисом (N. Metropolis) и Р. Л. Эшенхерстом (R. L. Ashenhurst) в *Math. Comp.* **19** (1965), 435–441. (См. также статью K. Sikdar, *Sankhyā* **B30** (1968), 315–334, и приведенный в ней список литературы.) Ф. Дж. Плогер (P. J. Plauger) в книге *The Standard C Library* (Prentice-Hall, 1992), 301–331, приводит подробное описание подпрограмм форматированного ввода-вывода целых чисел и чисел с плавающей точкой, написанных на языке программирования C.

УПРАЖНЕНИЯ

▸ **1.** [*25*] Обобщите метод 1, b таким образом, чтобы он был применим к позиционным системам счисления со смешанным основанием; преобразуйте выражение

$$a_m b_{m-1} \ldots b_1 b_0 + \cdots + a_1 b_0 + a_0 \quad \text{в} \quad A_M B_{M-1} \ldots B_1 B_0 + \cdots + A_1 B_0 + A_0,$$

где $0 \le a_j < b_j$ и $0 \le A_J < B_J$ при $0 \le j < m$ и $0 \le J < M$.

Проиллюстрируйте работу обобщенного таким образом метода на примере, выполнив перевод вручную величины "3 дня, 9 часов, 12 минут и 37 секунд" в длинные тонны, хандредвейты, стоуны, фунты и унции. (Пусть одна секунда равна одной унции. В британской системе весов 1 стоун равен 14 фунтам, 1 хандредвейт равен 8 стоунам, 1 длинная тонна равна 20 хандредвейтам.) Другими словами, пусть $b_0 = 60$, $b_1 = 60$, $b_2 = 24$, $m = 3$, $B_0 = 16$, $B_1 = 14$, $B_2 = 8$, $B_3 = 20$, $M = 4$. Задача заключается в поиске при помощи систематического метода, обобщающего метод 1, b, таких чисел $A_4, \ldots, A_0$, расположенных в надлежащих интервалах, чтобы $3b_2b_1b_0 + 9b_1b_0 + 12b_0 + 37 = A_4B_3B_2B_1B_0 + A_3B_2B_1B_0 + A_2B_1B_0 + A_1B_0 + A_0$. (Все арифметические операции должны выполняться в системе счисления со смешанным основанием.)

2. [*25*] Обобщите метод 1, a так, чтобы он был применим к позиционной системе счисления со смешанным основанием, как в упр. 1, и приведите пример работы полученного обобщения, решив вручную ту же задачу преобразования, что и в упр. 1.

▸ **3.** [*25*] (Д. Таранто (D. Taranto).) При преобразовании дробей остается открытым вопрос о количестве разрядов представления результата. Разработайте простое обобщение метода 2, а, такое, что для заданных двух положительных дробей u и ϵ, принимающих значения между 0 и 1 и представленных в формате по основанию b, дробь u преобразуется в свой округленный эквивалент по основанию B, который имеет достаточный размер M справа от разделяющей точки, чтобы обеспечить выполнение неравенства $|U - u| < \epsilon$. (В частности, если u кратно b^{-m} и $\epsilon = b^{-m}/2$, значение U будет представлено достаточным количеством разрядов, так что по заданным U и m дробь u может быть восстановлена точно. Заметим, что M может равняться нулю. Например, если $\epsilon \leq \frac{1}{2}$ и $u > 1 - \epsilon$, то правильный ответ — $U = 1$.)

4. [*M21*] (a) Докажите, что любое вещественное число с конечным *двоичным* представлением имеет также конечное *десятичное* представление. (b) Найдите простое соотношение между положительными числами b и B, которое дает необходимые и достаточные условия для того, чтобы любое вещественное число, имеющее конечное представление в формате по основанию b, имело также конечное представление в формате по основанию B.

5. [*M20*] Покажите, что программа (4) будет выполняться, если команду `LDX =10^n=` заменить командой `LDX =c=` при определенных значениях константы c.

6. [*30*] Исследуйте методы 1, a; 1, b; 2, a и 2, b для случая, когда b или B равно -2.

7. [*M18*] Известно, что $0 < \alpha \leq x \leq \alpha + 1/w$ и $0 \leq u \leq w$, где u — целое число. Докажите, что $\lfloor ux \rfloor$ равно либо $\lfloor \alpha u \rfloor$, либо $\lfloor \alpha u \rfloor + 1$. Более того, $\lfloor ux \rfloor = \lfloor \alpha u \rfloor$ точно, если $u < \alpha w$ и α^{-1} — целое число.

8. [*24*] Напишите MIX-программу, аналогичную (1), которая использует соотношение (5) и не содержит команд деления.

▸ **9.** [*M29*] Назначение данного упражнения — вычисление $\lfloor u/10 \rfloor$ и $u \bmod 10$ только при помощи операций двоичного сдвига, маскирования и сложения, если u — неотрицательное целое число. Положите k фиксированным целым числом, которое ≥ 2, и рассмотрите процесс вычисления

$$v \leftarrow u + 1, v \leftarrow v + \left\lfloor \frac{v}{2} \right\rfloor, v \leftarrow v + \left\lfloor \frac{v}{16} \right\rfloor, v \leftarrow v + \left\lfloor \frac{v}{256} \right\rfloor, v \leftarrow v + \left\lfloor \frac{v}{2^{2^k}} \right\rfloor;$$

$$q \leftarrow \left\lfloor \frac{v}{16} \right\rfloor, r \leftarrow v \bmod 16, r \leftarrow r + \left\lfloor \frac{r}{4} \right\rfloor, r \leftarrow \left\lfloor \frac{r}{2} \right\rfloor.$$

Чему равно наименьшее положительное целое число u, такое, что $q \neq \lfloor u/10 \rfloor$ или $r \neq u \bmod 10$?

10. [*22*] В табл. 1 показано, как на двоичном компьютере с использованием различных операций сдвига, маскирования и сложения может быть удвоено десятичное число, закодированное в двоичной системе. Предложите аналогичный метод, который позволял бы вычислять *половину* двоично-кодированного десятичного числа (с отбрасыванием остатка в случае, когда число нечетное).

11. [*16*] Преобразуйте число $(57721)_8$ в десятичное представление.

▸ **12.** [*22*] Придумайте быстрый метод преобразования вручную целых чисел из троичной системы счисления в десятичную и проиллюстрируйте его, преобразовав в десятичный вид число $(1212011210210)_3$. Как перевести число из десятичной системы счисления в троичную?

▸ **13.** [*25*] Предположим, что в ячейках памяти $\mathtt{U}+1$, $\mathtt{U}+2$, $\ldots$, $\mathtt{U}+m$ содержится заданная с многократной точностью дробь $(.u_{-1}u_{-2}\ldots u_{-m})_b$, где b — размер слова компьютера MIX. Напишите MIX-программу, выполняющую преобразование этой дроби в десятичный формат и усекающую ее до 180 десятичных разрядов. Ответ должен быть напечатан в двух строчках, разряды должны быть сгруппированы в 20 блоков по 9 разрядов в каждом, разделенных пробелами. (Воспользуйтесь командой `CHAR`.)

▶ **14.** [*M27*] (А. Шёнхаге (A. Schönhage).) При большом n время, необходимое для выполнения преобразования n-разрядного целого числа рассмотренным в разделе методом преобразования целых чисел многократной точности, имеет порядок n^2. Покажите, что n-разрядное целое число можно перевести в двоичный формат за $O(M(n)\log n)$ шагов, где $M(n)$ — количество циклов, необходимых для выполнения операции умножения n-битовых чисел, которые удовлетворяют "условиям гладкости" $M(2n) \geq 2M(n)$.

15. [*M47*] Можно ли существенным образом понизить верхнюю грань времени выполнения преобразования больших целых чисел, данную в упр. 14? (См. упр. 4.3.3–12.)

16. [*41*] Постройте быструю линейную итерационную конфигурацию для преобразования чисел из десятичной системы счисления в двоичную (см. раздел 4.3.3E).

17. [*M40*] Разработайте "идеальные" подпрограммы, выполняющие преобразования чисел с плавающей точкой, которые переводили бы p-разрядные десятичные числа в P-разрядные двоичные числа и наоборот, выдавая в обоих случаях правильный округленный результат в терминах раздела 4.2.2.

18. [*HM34*] (Дэвид В. Матула (David W. Matula).) Пусть $\mathrm{round}_b(u,p)$ — функция b, u и p, которые являются наилучшим приближением p-битового числа u с плавающей точкой, представленного в системе счисления по основанию b в смысле раздела 4.2.2. Предполагая, что $\log_B b$ иррационально и что целая часть принадлежит бесконечному интервалу, докажите, что

$$u = \mathrm{round}_b(\mathrm{round}_B(u,P),p)$$

для всех p-битовых чисел u с плавающей точкой, представленных по основанию b тогда и только тогда, когда $B^{P-1} \geq b^p$. (Другими словами, "идеальное" входное преобразование произвольного числа u в представление по независимому основанию B и выполняемое после него "идеальное" выходное преобразование этого результата всегда снова даст число u тогда и только тогда, когда промежуточная точность P будет достаточно большой, как определено вышеприведенной формулой.)

19. [*M23*] Предположим, что десятичное число $u = (u_7 \ldots u_1 u_0)_{10}$ представлено как двоично-закодированное десятичное число $U = (u_7 \ldots u_1 u_0)_{16}$. Найдите соответствующие константы c_i и маски m_i, такие, что операция $U \leftarrow U - c_i(U \wedge m_i)$, повторенная для $i = 1, 2, 3$, переводит число U в двоичное представление числа u, где "$\wedge$" означает извлечение (побитового AND).