

ОСОБЕННОСТИ ПРЕДСТАВЛЕНИЯ ВЕЩЕСТВЕННЫХ ЧИСЕЛ В ПОСТБИНАРНЫХ ФОРМАТАХ

Анотація. Розглянуто актуальність переходу до постбінарного комп'ютеризації як важливого кроку до досягнення «гнучкої розрядності» при кодуванні кількісних значень у сучасних комп'ютерних системах. Запропоновано ряд модифікацій традиційних форматів чисел з рухомою комою. Показані принципи формування дрібних, інтервальних і постбінарних форматів, заснованих на модифікації стандартних уявлень чисел у форматі з рухомою комою.

Ключові слова: постбінарні формати, тетралогіка, тетракоди, інтервальні обчислення, постбінарні обчислення, рухома кома, мантиса.

Аннотация. Рассмотрена актуальность перехода к постбинарному компьютеризации как важному шагу к достижению «гибкой разрядности» при кодировании количественных значений в современных компьютерных системах. Предложен ряд модификаций традиционных форматов чисел с плавающей запятой. Показаны принципы формирования дробных, интервальных и постбинарных форматов, основанных на модификации стандартных представлений чисел в формате с плавающей запятой.

Ключевые слова: постбинарные форматы, тетралогика, тетракоды, интервальные вычисления, постбинарные вычисления, плавающая запятая, мантисса.

Abstract. The actual transition to postbinary computing as an important step towards “flexible capacity” encoding quantitative values in the modern computer systems was regarded. A variety of modifications of traditional floating decimal point formats was proposed. Some examples of development of fractional, interval and postbinary formats based on the modification of standard representations of floating decimal point numbers were described.

Keywords: postbinary formats, tetralogic, tetracodes, interval computations, postbinary computations, floating decimal point, mantissa.

1. Введение

На сегодняшний день аппаратное обеспечение компьютеров ориентировано на операции с числами, представленными всего в двух форматах: целые числа и числа с плавающей запятой. Целочисленная арифметика оперирует конечным подмножеством множества целых чисел. Если аппаратная часть исправна, а программы не содержат ошибок, то целочисленная арифметика и программные надстройки над ней работают без погрешностей. Например, на основе целочисленной арифметики можно построить арифметику систематических дробей произвольной разрядности и обыкновенных дробей. Для точного представления результатов целочисленных арифметических операций современный компьютер имеет возможность сохранять необходимое количество разрядов для точного представления достаточно больших целых чисел.

Но при решении большинства практических задач операции производятся над вещественными числами. Произвольное вещественное число представляется бесконечной десятичной или двоичной дробью. На практике в научных и инженерных вычислениях вещественные числа приходится представлять в компьютере конечными дробями, чаще всего числами с плавающей запятой. Арифметика чисел с плавающей запятой поддерживается аппаратным обеспечением современных компьютеров и поэтому выполняется очень быстро. Но каждая операция с плавающей запятой может вносить погрешность, поскольку числа, представленные в формате с плавающей запятой (стандарт IEEE 754), представляют конечное множество, на которое отображается бесконечное множество вещественных чи-

сел. В итоге результат нескольких последовательных операций может оказаться совершенно неверным.

Приведем простой пример (аналогичный представленному в работе [1]). Пусть даны два целочисленных вектора x и y :

$$x = (10^{15}, 1500, -10^{18}, 10^{20}, 2, -10^{15}),$$

$$y = (10^{15}, 3, 10^{12}, 10^{13}, 222, 10^{18}).$$

Обозначим скалярное произведение x и y через $x \times y$, при выполнении которого соответственные элементы векторов перемножаются и эти произведения складываются. В точной целочисленной арифметике имеем

$$x \times y = 10^{30} + 4500 - 10^{30} + 10^{33} + 444 - 10^{33} = 4944.$$

Ниже приведен пример реализации данного целочисленного выражения в системе компьютерной алгебры Mathematica:

$$\text{In}[5]:=10^{30}+4500-10^{30}+10^{33}+444-10^{33}.$$

$$\text{Out}[5]=4944.$$

Однако арифметика чисел с плавающей запятой на любом современном компьютере (включая те компьютеры, на которых арифметика реализована в соответствии с обновленным в 2008 году стандартом формата представления чисел с плавающей запятой IEEE 754-2008) выдаст для такого скалярного произведения нулевое значение. Причина этого в столь большой разнице порядков слагаемых, что обычное представление чисел с плавающей запятой не позволяет корректно выполнить вычисление. Ниже приведен пример реализации данного выражения в системе компьютерной алгебры Mathematica при представлении элементов векторов x и y вещественными числами:

$$\text{In}[6]:=10.^{30}+4500.-10.^{30}+10.^{33}+444.-10.^{33}$$

$$\text{Out}[6]=0.$$

Эта катастрофическая погрешность возникает несмотря на то, что данные (в данном случае элементы векторов) используют менее 5% диапазона значений порядка, доступного на большинстве компьютеров.

Ниже показаны результаты символьного (признак знака « \rightarrow ») и численного (признак знака « $=$ ») процессоров математического пакета Mathcad Professional при вычислении данного примера:

$$10^{30}+4500-10^{30}+10^{33}+444-10^{33} \rightarrow 4944=4.944 \times 10^3.$$

$$10^{30}+4500-10^{30}+10^{33}+444-10^{33}=0.$$

$$10.^{30}+4500.-10.^{30}+10.^{33}+444.-10.^{33} \rightarrow 0=0.$$

В последнем варианте вычислений (элементы векторов – вещественные числа) символьный и численный процессоры Mathcad оказались бессильны.

Стоит также отметить, что при вычислении относительно простого полинома Румпа [2] с определенными входными значениями большинство существующих на сегодняшний день математических пакетов не в состоянии получить верный результат. Способы получения верного результата при вычислении полинома Румпа, в принципе, существуют [2, 3], однако принятие решения о выборе тех или иных способов вычислений полностью возлагается на пользователя. Причем такой пользователь должен хорошо понимать суть вычислительных процессов и иметь достаточную математическую подготовку. Естественно,

что при нарастании объемов вычислений в процессе исследования, моделирования и проектирования сложных систем и динамических процессов становится невозможным буквально «вручную» отслеживать проблемные участки в вычислениях и выявлять подобные примеру Румпа вычислительные аномалии. При этом в большинстве случаев ошибки в вычислениях остаются просто незамеченными, существенно искажая полученные результаты. Это позволяет предположить, например, что многие техногенные катастрофы последних десятилетий были в первую очередь обусловлены не «человеческим фактором», а разного рода вычислительными ошибками [4].

В данной статье рассматриваются, во-первых, актуальность перехода к постбинарному компьютерингу и «гибкой разрядности» при кодировании количественных значений в современных компьютерных системах, и, во-вторых, предлагаются постбинарные форматы кодирования вещественных чисел.

2. От бинарного к постбинарному компьютерингу

Первая волна интенсивного развития цифровых технологий, отсчет которой можно вести примерно с 1945 года, прошла почти исключительно на базе бинарной логики и арифметики. Однако, уже в 90-е годы, когда начался массовый переход к технологиям параллельных вычислений, ограниченность бинарного кодо-логического базиса стала проявляться особенно остро, в связи с чем актуализировались поиски дальнейших, постбинарных, путей развития компьютерных технологий.

Само понятие «компьютеринг» начало использоваться еще в эпоху механических вычислительных устройств, означая первоначально просто процесс реализации каких-либо вычислений. Однако к настоящему времени смысл этого термина существенно расширился, что нашло отражение в следующем современном определении: «Под компьютерингом обычно понимается деятельность, направленная на использование и разработку компьютерных технологий, компьютерной техники и программного обеспечения. Это связанная с компьютерами часть информационных технологий. С этим понятием тесно связаны компьютерные науки, направленные на изучение теоретических и практических основ вычислений и информатики, а также – на их реализацию в компьютерных системах» [5].

Такое расширенное понимание компьютеринга сложилось уже в бинарную эпоху, в связи с чем под этим термином традиционно предполагается бинарный компьютеринг, в основе которого лежит использование двоичной логики и двоичной системы счисления. Целесообразным и оправданным в настоящее время следует также признать и использование термина постбинарный компьютеринг, включающего в себя все, что выходит за рамки двоичной логики и систем счисления, однозначно сводимых к двоичной (по сути точечной и однозначной) системе представления количественной информации.

В целом можно констатировать, что возможности продуктивного системного подхода к исследованию вопросов постбинарного компьютеринга могут быть обеспечены различными способами, следствием чего является естественное многообразие соответствующих исследований. Целесообразно, в частности, назвать такие перспективные направления, как интервальные вычисления [6, 7], гипервычисления, квантовый и молекулярный компьютеринг [8].

В рамках кодо-логического подхода, развиваемого авторами данной статьи, важнейшими предпосылками для системного исследования проблематики постбинарного компьютеринга явились следующие моменты.

Во-первых, введение в научный оборот понятия «кодо-логический базис» (с доопределением «расширенный» или «обобщенный»), предполагающего рассмотрение эволюции логических и арифметических основ компьютерных технологий в неразрывном единстве и в достаточно широкой исторической перспективе и ретроспективе [9].

Во-вторых, выявление и анализ как целостного явления добинарного (прабинарного) кодо-логического базиса, основными составляющими которого являются монологаика и различные эволюционирующие формы монокодов [10]. В целом появились основания говорить о целой эпохе прабинарного компьютеринга, предшествовавшей бинарному этапу в развитии вычислительных средств. Исследование закономерностей добинарной эволюции и перехода к бинарному компьютерингу позволило более системно и продуктивно подойти к анализу вопросов перехода к постбинарным вычислениям.

В-третьих, введение в научный оборот таких взаимосвязанных понятий, как тетралогаика и тетракоды [11], позволило впервые одновременно выйти за пределы как одномерного пространства двоичной логики, так и точечного бинарного представления количественной информации, что явилось решающим шагом к последующим исследованиям постбинарного компьютеринга в контексте кодо-логической эволюции.

Логические, алгоритмические и прочие предпосылки перехода к постбинарному компьютерингу начали складываться уже в первой половине XX века в ходе интенсивного развития средств и методов бинарного компьютеринга во второй половине прошлого века, но только на рубеже тысячелетий (в первую очередь, благодаря тотальному переходу к параллельным вычислениям и формированию тесно взаимосвязанной глобальной компьютерной инфраструктуры) начала проявляться настоятельная необходимость преодоления многочисленных ограничений традиционного бинарного кодо-логического базиса, ориентированного преимущественно на последовательную (так называемую фон-неймановскую) вычислительную архитектуру.

В связи с этим назрела необходимость комплексного исследования всей совокупности вопросов, связанных с переходом к постбинарному компьютерингу, основанному на использовании различных форм гиперлогики и гиперкодов, которые могут рассматриваться как обобщение тетралогики и тетракодов применительно к многомерным логическим пространствам.

3. Способы представления вещественных чисел в постбинарных форматах

В работе [3] был предложен ряд способов для преодоления проблем, связанных с ограничением разрядности чисел, поскольку использование при вычислениях разрядности, существенно превышающей стандартную, является одним из способов получения правильных результатов. Все эти способы в совокупности позволяют в процессе вычислений выполнять следующие операции:

- увеличение (или выравнивание) разрядности во избежание переполнения разрядов результата и выполнение корректных вычислений;
- выполнение так называемого отложенного деления, когда отдельно вычисляются числитель и знаменатель, а деление производится на последнем шаге вычисления;
- использование интервальных вычислений.

В рамках реализации вышеперечисленных операций возможно достижение достаточно надежного и эффективного контроля за корректностью вычислений. Сами же опера-

ции можно эффективно использовать при разработке и реализации постбинарных методов вычислений, основанных на постбинарном представлении количественных значений [12, 13]. Однако введение дан-

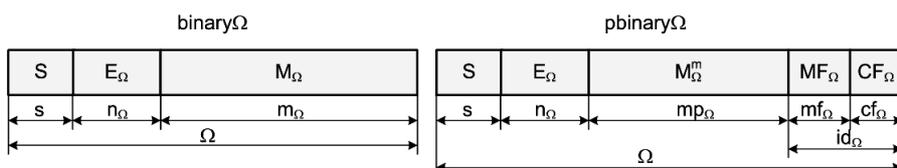


Рис. 1. Структура числа в формате $\text{binary}\Omega$ и в эквивалентном ему формате $\text{pbinary}\Omega$ (Ω – количество разрядов формата числа, S – знак, E_Ω

– смещенная экспонента, M_Ω и M_Ω^m – остаток мантииссы соответствующих форматов, MF_Ω – модификатор формата, CF_Ω – код формата)

ных операций в постбинарный вычислительный процесс невозможно без незначительных модификаций самих форматов чисел с плавающей запятой. Поэтому на основании форматов чисел (binary32, binary64, binary128) стандарта IEEE754-2008 было предложено 5 модифицированных (постбинарных) форматов чисел различной точности (рис. 1): от одинарной (pbinary32) до увеличенной в 8 раз по сравнению со стандартной (pbinary256).

В предложенных постбинарных форматах поле мантиссы числа претерпело модификацию путем выделения необходимого количества битов для идентификатора формата (поле id_{Ω}), состоящего из модификатора MF_{Ω} (внутренняя модификация формата $pbinary_{\Omega}$) и кода формата CF_{Ω} (признак принадлежности к формату $pbinary_{\Omega}$).

Формирование кода идентификатора формата CF_{Ω} определено таким образом (табл. 2), что по положению младшего нуля относительно указателя p на начальный (первый младший) бит числа можно точно определить как сам формат числа $pbinary_{\Omega}$, так и его границы в диапазоне $[p + \Omega - 1; p]$.

В то же время, в зависимости от модификатора MF_{Ω} , формат $pbinary_{\Omega}$ может содержать в себе другие подформаты представления данных. Для таких форматов приняты обозначения в виде $pbinary_{\Omega}/\Psi\phi$, где Ψ – точность формата, в 2 (для $\phi \in \{f, i, p\}$) или 4 (для $\phi \in \{fp, ip\}$) раза меньшая Ω и показывающая формат составной части чисел, заключенных в определяемую разрядность Ω ; ϕ – указатель типа числа с плавающей запятой (представлены следующие типы: f – дробное число, i – интервальное, p – постбинарное) или типа постбинарного числа (представлены fp – постбинарное дробное и ip – постбинарное интервальное). В табл. 1 приведены все разработанные модификации форматов чисел различной точности в зависимости от значения модификатора MF_{Ω} .

Таблица 1. Предлагаемые модификации формата $pbinary_{\Omega}$

| Модификатор $MF[11:0]$ | pbinary32 $MF[0]$ | pbinary64 $MF[1:0]$ | pbinary128 $MF[4:0]$ | pbinary256 $MF[11:0]$ |
|---------------------------|-----------------------------|-------------------------------|--------------------------------|---------------------------------|
| 0000 ... 0000 | pbinary32 | pbinary64 | pbinary128 | pbinary256 |
| 0000 ... 0001 | pbinary32/16p | pbinary64/32f | pbinary128/64f | pbinary256/128f |
| 0000 ... 0010 | | pbinary64/32i | pbinary128/64i | pbinary256/128i |
| 0000 ... 0011 | | pbinary64/32p | pbinary128/64p | pbinary256/128p |
| 0000 ... 0100 | | | pbinary128/32fp | pbinary256/64fp |
| 0000 ... 0101 | | | pbinary128/32ip | pbinary256/64ip |
| ... | | | резерв | резерв |

Рассмотрим структуру, особенности формирования и назначение каждой из представленных модификаций постбинарного формата $pbinary_{\Omega}$ (в качестве примера на рис. 2 представлены все модификации формата $pbinary_{128}$):

- $pbinary_{\Omega}$ – фактически число формата $binary_{\Omega}$, имеющее модифицированную (уменьшенную на id_{Ω} разрядов) мантиссу. Для постбинарного формата $pbinary_{\Omega}$ справедливы следующие соотношения, связанные с определением количества разрядов полей после модификации:

$$id_{\Omega} = \frac{\Omega}{16} \text{ – разрядность идентификатора формата;}$$

$cf_{\Omega} = \log_2 \frac{\Omega}{16}$ – разрядность кода формата;

$mf_{\Omega} = id_{\Omega} - cf_{\Omega} = \frac{\Omega}{16} - \log_2 \frac{\Omega}{16}$ – разрядность модификатора формата;

$mp_{\Omega} = m_{\Omega} - id_{\Omega}$ – разрядность модифицированной мантиссы.

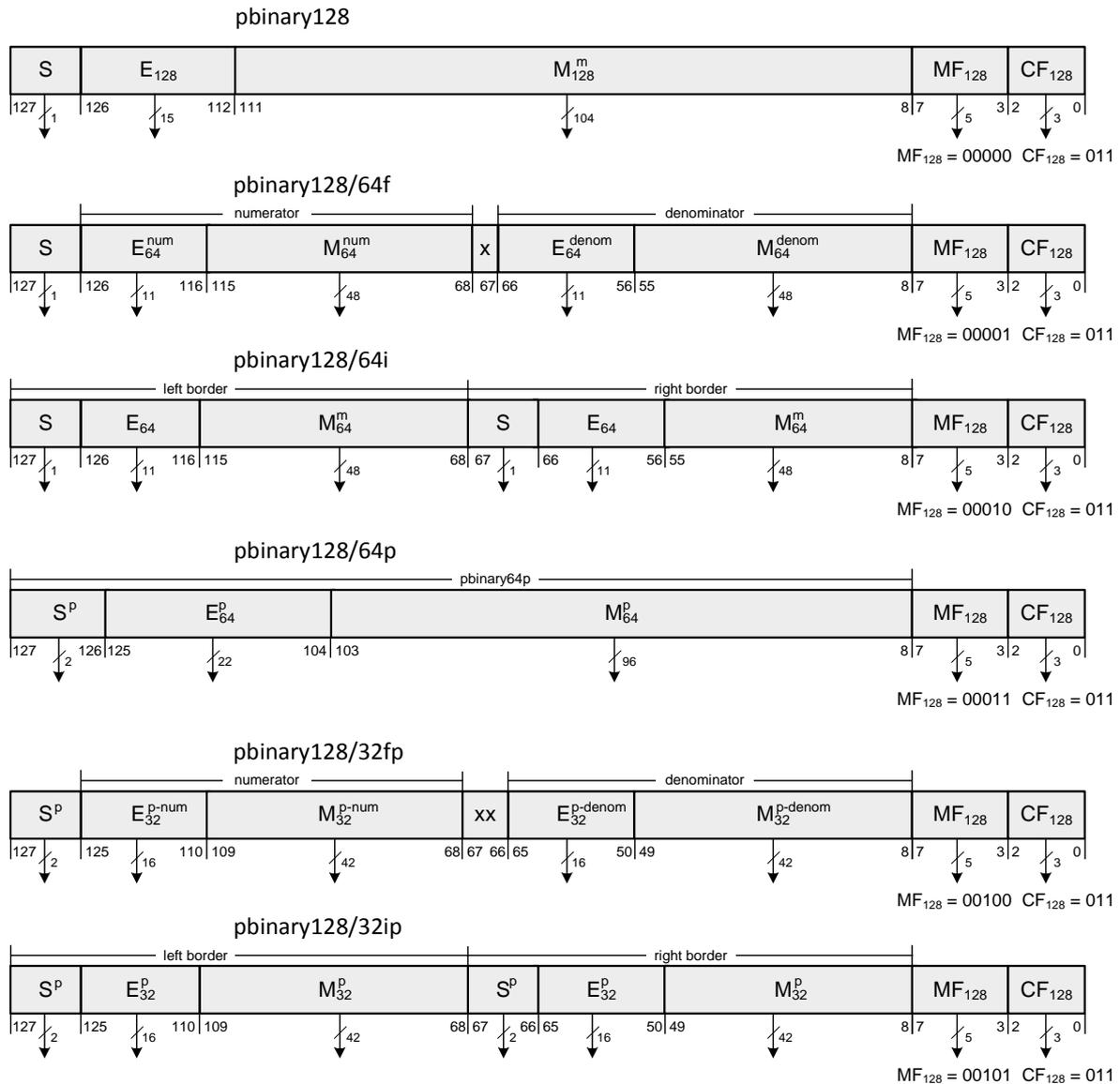


Рис. 2. Модификации формата pbinary128

Поля знака (S) и смещенной экспоненты (E_{Ω}) переносятся в модифицированный формат без изменения. В табл. 2 показаны разрядности форматов pbinary Ω по отношению к binary Ω .

Каждое число, представленное в формате pbinary Ω , имеет знаковый бит S (рис. 1): $S = 0$ для положительного и $S = 1$ для отрицательного числа. Значение смещенной экспоненты E_{Ω} формируется следующим образом:

$$E_{\Omega} = \exp_2 + \text{offset}_2, \quad (1)$$

где exp_2 – значение экспоненты двоичной дроби в нормализованном экспоненциальном виде, offset – заданное смещение экспоненты в Ω -битном формате: $\text{offset}_{10} = 2^{(n_\Omega - 1)} - 1 \Rightarrow \text{offset}_2 = (\text{offset}_{10})_2$.

Таблица 2. Разрядность $\text{rbinary}\Omega$ с указанием значения кода формата

| Ω | s | n_Ω | mp_Ω | id_Ω | mf_Ω | cf_Ω | CF_Ω |
|----------|---|------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| 32 | 1 | 8 | 21 | 2 | 1 | 1 | 0 |
| 64 | 1 | 11 | 48 | 4 | 2 | 2 | 01 |
| 128 | 1 | 15 | 104 | 8 | 5 | 3 | 011 |
| 256 | 1 | 20 | 219 | 16 | 12 | 4 | 0111 |

В поле M_Ω записывается остаток мантиссы двоичного нормализованного числа с плавающей точкой (отброшена старшая 1 от нормализованного двоичного числа вида $\pm 1, \text{xx} \dots \text{x} \cdot \text{exp}_2$). В формате полей $\text{rbinary}\Omega$ область нормализованных чисел лежит в пределах: $S = x$ (любое значение); $E_\Omega = 000 \dots 01 \div 111 \dots 10$; $M_\Omega^m = 000 \dots 00 \div 111 \dots 11$.

Таким образом, для формата $\text{rbinary}\Omega$ справедливы следующие соотношения:

– двоичное нормализованное число dn :

$$dn = (-1)^S \cdot 1, M_\Omega^m \cdot \exp_2^{(E_\Omega - \text{offset}_2)}; \quad (2)$$

– десятичное нормализованное число xn :

$$xn = (-1)^S \cdot 2^{((E_\Omega)_{10} - \text{offset}_{10})} \cdot \left(1 + \frac{(M_\Omega^m)_{10}}{2^{\text{mp}_\Omega}} \right); \quad (3)$$

– абсолютная максимально возможная погрешность Δ_{\max} представления числа равна половине шага числа, т.е. половине величины наименьшего разряда:

$$\Delta_{\max} = 2^{((E_\Omega)_{10} - \text{offset}_{10} - \text{mp}_\Omega + 1)}; \quad (4)$$

– относительная максимально возможная погрешность $\delta_{N_{\max}}$ представления нормализованного числа:

$$\delta_{N_{\max}} = \frac{1}{2^{\text{mp}_\Omega} + (M_\Omega^m)_{10}} \cdot 100\%. \quad (5)$$

Аналогично $\text{binary}\Omega$, формат $\text{rbinary}\Omega$ имеет ряд исключительных чисел, к которым нельзя применять формулы (2–5):

1) положительный и отрицательный ноль: $+0$ ($S = x$; $E_\Omega = 0$; $M_\Omega^m = 0$) и -0 ($S = 1$; $E_\Omega = 0$; $M_\Omega^m = 0$). Большинство программных средств эти нули не различает (поскольку в этом нет необходимости), считая их просто нулевыми значениями;

2) положительная и отрицательная бесконечности ($+\text{Infinity}$, $-\text{Infinity}$): $+\infty$ ($S = 0$; $E_\Omega = 111 \dots 11$; $M_\Omega^m = 0$) и $-\infty$ ($S = 1$; $E_\Omega = 111 \dots 11$; $M_\Omega^m = 0$). Это числа, которые больше границ диапазона представления чисел;

3) не числа – NaN (No a Numbers), к которым относятся символы, или результаты недопустимых операций. При $E_{\Omega} = 111\dots 11$ и M_{Ω}^m – любое ненулевое значение, различают +NaN ($S = 1$) и –NaN ($S = 0$);

4) ненормализованные числа – числа, мантиссы которых лежат в диапазоне $0,1 \leq M_{\Omega}^m < 1$. Ненормализованные числа находятся ближе к нулю, чем нормализованные, и разбивают минимальный разряд нормализованного числа на некоторое подмножество. Для ненормализованных чисел справедливы следующие равенства:

– двоичное ненормализованное число dd :

$$dd = (-1)^S \cdot 0, M_{\Omega}^m \cdot \exp^{-\text{offset}_2}; \quad (6)$$

– десятичное ненормализованное число xd :

$$xd = (-1)^S \cdot 2^{(1-\text{offset}_{10})} \cdot \frac{(M_{\Omega}^m)_{10}}{2^{\text{mp}_{\Omega}}}; \quad (7)$$

– относительная максимально возможная погрешность $\delta_{D_{\max}}$ представления ненормализованного числа:

$$\delta_{D_{\max}} = \frac{1}{2 \cdot (M_{\Omega}^m)_{10}} \cdot 100\%; \quad (8)$$

Формула (4) справедлива также для ненормализованных чисел.

На рис. 3 представлено отображение чисел форматов binary_{Ω} и rbinary_{Ω} на числовой оси.

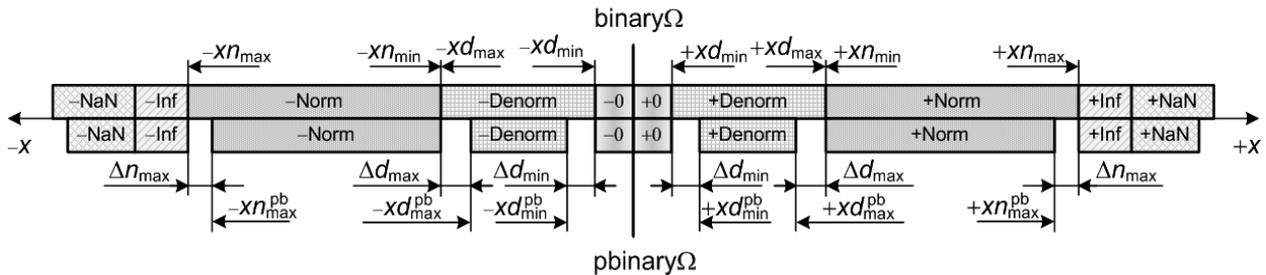


Рис. 3. Отображение чисел форматов binary_{Ω} и rbinary_{Ω} на числовой оси (Norm, Denorm – области нормализованных и ненормализованных чисел; pb – признак постбинарного формата; xd и xn – ненормализованные и нормализованные числа)

Совокупности полей $\pm\text{Norm}$ и $\pm\text{Denorm}$ ($[-xn_{\max}^{\text{pb}}; -xd_{\min}^{\text{pb}}] \cup [+xd_{\min}^{\text{pb}}; +xn_{\max}^{\text{pb}}]$) являются диапазоном представления положительных или отрицательных чисел формата rbinary_{Ω} на всей числовой оси. Используя соотношения (3) и (7), можно получить десятичные значения модулей минимальной xd_{\min}^{pb} (значение по модулю наименьшего ненормализованного числа) и максимальной xn_{\max}^{pb} (значение по модулю наибольшего нормализованного числа) границ диапазона представления чисел в формате rbinary_{Ω} (табл. 3). Модификация (т.е. уменьшение разрядности) мантиссы постбинарного формата приводит, прежде всего, к сужению диапазона представления близких к нулю чисел (область ненормализованных чисел) по отношению к эквивалентному двоичному формату binary_{Ω} . Используя формулы (3, 7), получим абсолютные погрешности границ диапазона представления чисел по отношению к формату binary_{Ω} (xn_{\max} , xd_{\max} и xd_{\min} – значения модулей соответствующих границ диапазона чисел формата binary_{Ω}):

– абсолютная погрешность представления минимального (минимального ненормализованного) Ω -разрядного числа:

$$\Delta d_{\min} = \left| xd_{\min} - xd_{\min}^{\text{pb}} \right| = 2^{(1-\text{offset}_{10}-\text{mp}_{\Omega})} \cdot (1 - 2^{-\text{id}_{\Omega}}); \quad (9)$$

– абсолютная погрешность представления максимального (максимального нормализованного) Ω -разрядного числа:

$$\Delta n_{\max} = \left| xn_{\max} - xn_{\max}^{\text{pb}} \right| = 2^{(\text{offset}_{10}-\text{mp}_{\Omega})} \cdot (1 - 2^{-\text{id}_{\Omega}}). \quad (10)$$

Абсолютная погрешность представления максимального ненормализованного Ω -разрядного числа Δd_{\max} соответствует формуле (9), следовательно, $\Delta d_{\max} = \Delta d_{\min}$. Последнее равенство можно объяснить тем, что разница в представлении ненормализованных чисел rbinary_{Ω} и binary_{Ω} обусловлена одинаковой погрешностью представления значений мантисс постбинарного и бинарного форматов (все ненормализованные числа имеют нулевое значение порядка) на протяжении всего диапазона $\pm \text{Denorm}$. Таким образом,

$$\begin{aligned} \left[xd_{\min}^{\text{pb}}; xd_{\max}^{\text{pb}} \right] &= \left[xd_{\min} + \Delta d_{\min}; xd_{\max} - \Delta d_{\max} \right] = \\ &= \left[xd_{\min} + \Delta d_{\min}; xd_{\max} - \Delta d_{\min} \right] = \left[2^{\text{id}_{\Omega}} \cdot xd_{\min}; 2^{-\text{id}_{\Omega}} \cdot xd_{\max} \right]. \end{aligned} \quad (11)$$

Равенство (11) показывает, что модуль минимального ненормализованного числа формата rbinary_{Ω} больше модуля аналогичного числа формата binary_{Ω} в $2^{\text{id}_{\Omega}}$ раза, и наоборот, модуль максимального ненормализованного числа формата rbinary_{Ω} меньше модуля аналогичного числа формата binary_{Ω} в $2^{\text{id}_{\Omega}}$ раза. Таким образом, диапазон $\pm \text{Denorm}$ формата rbinary_{Ω} в $2^{\text{id}_{\Omega}+1}$ раза меньше соответствующего диапазона формата binary_{Ω} .

Отношение максимальных границ диапазона нормализованных чисел для форматов rbinary_{Ω} и binary_{Ω} близко к единице, однако следует учитывать величину абсолютной погрешности представления числа в постбинарном и бинарном форматах. Десятичное значение минимального нормализованного числа rbinary_{Ω} соответствует аналогичному числу binary_{Ω} ($\pm xn_{\min}^{\text{pb}} = \pm xn_{\min}$, $\Delta n_{\min} = 0$), поскольку в формировании этих чисел участвуют только значения полей знака и порядка, которые идентичны в соответствующих постбинарных и бинарных форматах (рис. 1). Таким образом,

$$\left[xn_{\min}^{\text{pb}}; xn_{\max}^{\text{pb}} \right] = \left[xn_{\min}; xn_{\max} - \Delta n_{\max} \right] = \left[xn_{\min}; \frac{2^{-(\text{mp}_{\Omega}+1)-\text{id}_{\Omega}} - 1}{2^{-(\text{mp}_{\Omega}+1)} - 1} \cdot xn_{\max} \right]. \quad (12)$$

Рассмотрим функцию $f(a,b) = \frac{2^{-(a+1)-b} - 1}{2^{-(a+1)} - 1}$. Имеем $\lim_{a \rightarrow \infty} f(a,b) = 1$, откуда следует, что $f(a,b) \approx 1$ при возрастающем аргументе a . Соответственно, для значений mp_{Ω} и id_{Ω} представленных форматов справедливо соотношение

$$\frac{2^{-(\text{mp}_{\Omega}+1)-\text{id}_{\Omega}} - 1}{2^{-(\text{mp}_{\Omega}+1)} - 1} \approx 1, \quad (13)$$

причем данное приближенное равенство стремится к строгому равенству при возрастании значения mp_{Ω} .

Исходя из (13), выражение (12) можно записать в виде

$$\left[xn_{\min}^{\text{pb}}; xn_{\max}^{\text{pb}} \right] = \left[xn_{\min}; \approx xn_{\max} \right] \Rightarrow \left[xn_{\min}^{\text{pb}}; xn_{\max}^{\text{pb}} \right] \approx \left[xn_{\min}; xn_{\max} \right]. \quad (14)$$

Десятичные значения максимальных и минимальных чисел формата $\text{rbinary}\Omega$, их абсолютной погрешности по отношению к формату $\text{binary}\Omega$ представлены в табл. 3.

Таблица 3. Значения максимальных и минимальных границ диапазонов Norm и Denorm положительных и отрицательных чисел формата $\text{rbinary}\Omega$ (в скобках указаны значения абсолютной погрешности представления чисел в сопоставлении с форматом $\text{binary}\Omega$)

| Ω | $xd_{\min}^{\text{pb}} (\Delta d_{\min})$ | xd_{\max}^{pb} | xn_{\min}^{pb} | $xn_{\max}^{\text{pb}} (\Delta n_{\max})$ |
|----------|--|---|--|---|
| 32 | $\pm 2^{-147} \approx$ $\approx \pm 5,60519386 \cdot 10^{-45}$ ($\approx 4,204 \cdot e^{-45}$) | $\pm 2^{-126} \cdot (1 - 2^{-21}) \approx$ $\approx \pm 1,17549379 \cdot 10^{-38}$ | $\pm 2^{-126} \approx$ $\approx \pm 1,17549435 \cdot 10^{-38}$ | $\pm 2^{127} \cdot (2 - 2^{-21}) \approx$ $\approx \pm 3,40282286 \cdot 10^{38}$ ($\approx 6,085 \cdot e^{31}$) |
| 64 | $\pm 2^{-1070} \approx$ $\approx \pm 7,90505033 \cdot 10^{-323}$ ($\approx 7,411 \cdot e^{-323}$) | $\pm 2^{-1022} \cdot (1 - 2^{-48}) \approx$ $\approx \pm 2,22507386 \cdot 10^{-308}$ | $\pm 2^{-1022} \approx$ $\approx \pm 2,22507386 \cdot 10^{-308}$ | $\pm 2^{1023} \cdot (2 - 2^{-48}) \approx$ $\approx \pm 1,79769313 \cdot 10^{308}$ ($\approx 2,994 \cdot e^{293}$) |
| 128 | $\pm 2^{-16486} \approx$ $\approx \pm 1,65764483 \cdot 10^{-4963}$ ($\approx 1,651 \cdot e^{-4963}$) | $\pm 2^{-16382} \cdot (1 - 2^{-104}) \approx$ $\approx \pm 3,36210314 \cdot 10^{-4932}$ | $\pm 2^{-16382} \approx$ $\approx \pm 3,36210314 \cdot 10^{-4932}$ | $\pm 2^{16383} \cdot (2 - 2^{-104}) \approx$ $\approx \pm 1,18973150 \cdot 10^{4932}$ ($\approx 2,921 \cdot e^{4900}$) |
| 256 | $\pm 2^{-524505} \approx$ $\approx \pm 1,8286336 \cdot 10^{-157892}$ ($\approx 1,829 \cdot e^{-157892}$) | $\pm 2^{-524286} \cdot (1 - 2^{-219}) \approx$ $\approx \pm 1,54061213 \cdot 10^{-157826}$ | $\pm 2^{-524286} \approx$ $\approx \pm 1,54061213 \cdot 10^{-157826}$ | $\pm 2^{524287} \cdot (2 - 2^{-219}) \approx$ $\approx \pm 2,59637057 \cdot 10^{157826}$ ($\approx 1,541 \cdot e^{157760}$) |

• $\text{rbinary}\Omega/\Psi\text{f}$ – дробное Ω -разрядное число, состоящее из числителя (numerator), знаменателя (denominator) Ψ -разрядной точности (т.е. двух чисел формата $\text{rbinary}\Psi$) и общего знака для дроби. Значения Ω и Ψ связаны следующим соотношением: $\Psi = \Omega / 2$ при $\Omega = \{64, 128, 256, \dots\}$.

Пусть a и b – числа формата $\text{binary}\Psi$, т.е. фактически числа стандарта IEEE 754, над которыми нужно выполнить операцию деления a/b . В ряде случаев, обусловленных, прежде всего, сложностью общего цикла вычислений, данную операцию можно отложить, т.е. сохранить результат в виде дроби $\frac{a}{b}$ до заключительного этапа вычислений. В таком случае справедливы следующие равенства:

$S = S_{\Psi a} \oplus S_{\Psi b}$ – знак дроби;

$P_{\Psi}^{\text{num}} = P_{\Psi a}$, $M_{\Psi}^{\text{num}} = M_{\Psi a}^{\text{m}}$ – порядок и мантисса числителя дроби;

$P_{\Psi}^{\text{denom}} = P_{\Psi b}$, $M_{\Psi}^{\text{denom}} = M_{\Psi b}^{\text{m}}$ – порядок и мантисса знаменателя дроби.

• $\text{rbinary}\Omega/\Psi\text{i}$ – интервальный Ω -разрядный формат числа, содержащего два полноценных числа формата $\text{rbinary}\Psi$ ($\Psi = \Omega / 2$, $\Omega = \{64, 128, 256, \dots\}$), которые являются левой (left border) и правой (right border) границами интервала [14]. Заполнение полей данного формата аналогично дробному, за исключением наличия знаковых полей для каждого значения границ интервала. Числа в формате $\text{rbinary}\Omega/\Psi\text{i}$ могут быть использованы для решения задач в рамках интервального анализа или интервальной математики, где они будут представлять собой интервальный тип данных, вычислительные операции с которыми исключают возможные ошибки округления [15].

• $\text{rbinary}\Omega/\Psi\text{p}$ – постбинарное Ψ -разрядное число ($\Psi = \Omega / 2$, $\Omega = \{32, 64, 128, 256, \dots\}$). В таком числе в качестве основного постбинарного формата рассматривается тетракод, каждый разряд которого представлен тетритом [16], кодирующим одно из четырех состояний: 0, 1, а также состояния «неопределенности» (A) и «множественности» (M).

Поскольку тетрит должен кодировать четыре возможных значения разряда, для его хранения и использования в современных компьютерных системах потребуются два бита. Поэтому расширение числа формата $\text{rbinary}\Psi$ до постбинарного Ω -разрядного формата $\text{rbinary}\Omega/\Psi_p$ происходит за счет удвоения разрядности знака, порядка и модифицированной мантиссы. При этом каждая пара битов постбинарных значений знака, мантиссы и порядка имеет одно из значений множества $\{0, 1, A, M\}$ [17], кодируемое следующим образом: 00 – состояние неопределенности (A); 01 – ноль (0); 10 – единица (1); 11 – состояние множественности (M).

При формировании разрядности знака, порядка и мантиссы постбинарного числа формата $\text{rbinary}\Omega/\Psi_p$ справедливы следующие соотношения: $s_\Omega = s_\Psi^p = 2 \cdot s_\Psi$, $n_\Omega = n_\Psi^p = 2 \cdot n_\Psi$, $mp_\Omega = mp_\Psi^p = 2 \cdot mp_\Psi$.

- $\text{rbinary}\Omega/\Psi_{fp}$ – представляет формат постбинарного дробного Ψ -разрядного числа ($\Psi = \Omega/4, \Omega = \{128, 256, \dots\}$). Формируется объединением описанных выше преобразований $\text{rbinary}\Omega/\Psi_f \cup \text{rbinary}\Omega/\Psi_p$.

- $\text{rbinary}\Omega/\Psi_{ip}$ – представляет формат постбинарного интервального Ψ -разрядного числа ($\Psi = \Omega/4, \Omega = \{128, 256, \dots\}$). Формирование полей данного числа происходит по схеме: $\text{rbinary}\Omega/\Psi_i \cup \text{rbinary}\Omega/\Psi_p$.

4. Заключение

В контексте перехода к постбинарному компьютерному многообразию представления логических и численных значений будет неизбежно возрасть в связи с необходимостью развития как логических и алгоритмических основ компьютерных технологий, так и самого понятия числа, в том числе на уровне базовых форматов представления информации. Предложенная в работе система компьютерных форматов данных и соответствующая система обозначений могут рассматриваться как прототипы базовых элементов для нового поколения компьютерных архитектур, соответствующих начальному этапу постбинарного компьютерного.

Реализация вычислительных алгоритмов на базе предложенных форматов позволит обеспечить существенное расширение функциональных возможностей перспективных процессоров, в том числе за счет реализации текущего контроля требуемой разрядности и обеспечения на этой базе «гибкого форматирования» и «гибкой разрядности» при работе с компьютерным представлением численной информации. Важнейшим результатом этого должно стать кардинальное повышение надежности вычислений и оптимизация их разрядности.

На первом этапе речь, естественно, может идти преимущественно о программном моделировании новых форматов, алгоритмов и архитектур на базе существующих компьютерных систем. На следующем этапе предполагается реализация соответствующих схемных решений на базе FPGA, вплоть до реализации экспериментальных прототипов постбинарных процессоров.

В перспективе переход от бинарного компьютерного к постбинарному представляется практически неизбежным [18]. Во времени этот процесс может растянуться на десятилетия, но начинать его необходимо уже сейчас.

СПИСОК ЛИТЕРАТУРЫ

1. Numerical Toolbox for Verified Computing (Pascal-XSC Programs) / R. Hammer, M. Hocks, U. Kulisch, D. Ratz. – Berlin: Springer-Verlag Heidelberg, 1993. – С. 8.
2. Loh E. Rump's Example Revisited / E. Loh, G. Walster // Reliable Computing. – 2002. – N 8. – P. 245 – 248.

3. Аноприенко А.Я. Пример Румпа в контексте традиционных, интервальных и постбинарных вычислений / А.Я. Аноприенко, В.А. Гранковский, С.В. Иваница // Научные труды Донецкого национального технического университета. – (Серия «Проблемы моделирования и автоматизации проектирования динамических систем» (МАП-2011)). – Донецк: ДонНТУ, 2011. – Вып. 9 (179). – С. 324 – 343.
4. Петров Ю.П. Обеспечение надежности и достоверности компьютерных расчетов / Петров Ю.П. – СПб: БХВ-Петербург, 2008. – 160 с.
5. Computing – Wikipedia, the free encyclopedia (Wikipedia.org). – Режим доступа: <http://en.wikipedia.org/wiki/Computing>.
6. Алефельд Г. Введение в интервальные вычисления / Г. Алефельд, Ю. Херцбергер. – М.: Мир, 1987. – 360 с.
7. Шарый С.П. Конечномерный интервальный анализ [Электронный ресурс] / С.П. Шарый // Институт вычислительных технологий СО РАН. – Изд-во «XYZ»-2012. – С. 604. – Режим доступа: <http://www.nsc.ru> (interval).
8. Прохоров А. Прогнозы развития информационных технологий (по материалам компании Gartner) [Электронный ресурс] / А. Прохоров // Компьютер пресс. – 2006. – № 1. – Режим доступа: <http://www.cpress.ru>.
9. Аноприенко А.Я. Расширенный кодо-логический базис компьютерного моделирования / А.Я. Аноприенко // Информатика, кибернетика и вычислительная техника (ИКВТ-97): сб. научн. тр. ДонГТУ. – Донецк: ДонГТУ, 1997. – Вып. 1. – С. 59 – 64.
10. Аноприенко А.Я. Восхождение интеллекта: эволюция монокодовых вычислительных моделей / А.Я. Аноприенко // Научные труды Донецкого государственного технического университета. – (Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ-2000)). – Донецк: ДонГТУ, 2000. – Вып. 15. – С. 87 – 107.
11. Аноприенко А.Я. Тетралогики и тетракоды / А.Я. Аноприенко // Сб. тр. факультета вычислительной техники и информатики. – Донецк: ДонГТУ, 1996. – Вып. 1. – С. 32 – 43.
12. Аноприенко А.Я. Обобщенный кодо-логический базис в вычислительном моделировании и представлении знаний: эволюция идеи и перспективы развития / А.Я. Аноприенко // Научные труды Донецкого национального технического университета. – (Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ-2005)). – Донецк: ДонНТУ, 2005. – Вып. 93. – С. 289 – 316.
13. Аноприенко А.Я. Особенности постбинарного кодирования на примере интервального представления результатов вычислений по формуле Бэйли-Боруэйна-Плаффа / А.Я. Аноприенко, С.В. Иваница // Научные труды Донецкого национального технического университета. – (Серия: «Информатика, кибернетика и вычислительная техника» (ИКВТ-2010)). – Донецк: ДонНТУ, 2010. – Вып. 11 (164). – С. 19 – 23.
14. Аноприенко А.Я. Интервальные вычисления и перспективы их развития в контексте кодо-логической эволюции / А.Я. Аноприенко, С.В. Иваница // Научные труды Донецкого национального технического университета. – (Серия «Проблемы моделирования и автоматизации проектирования динамических систем» (МАП-2010)). – Донецк: ДонНТУ, 2010. – Вып. 8 (168). – С. 150 – 160.
15. Moore R.E. Interval analysis. Englewood Cliffs / Moore R.E. – N.J.: Prentic-e-Hall, 1966. – P. 144.
16. Иваница С.В. Особенности реализации операций тетралогики / С.В. Иваница, А.Я. Аноприенко // Научные труды Донецкого национального технического университета. Серия: «Информатика, кибернетика и вычислительная техника» (ИКВТ-2011). – Донецк: ДонНТУ, 2011. – Вып. 13 (185). – С. 134 – 140.
17. Аноприенко А.Я. Археомоделирование: модели и инструменты докомпьютерной эпохи / Аноприенко А.Я. – Донецк: УНИТЕХ, 2007. – 318 с.
18. Аноприенко А.Я. Постбинарный компьютеринг и моделирование сложных систем в контексте кодо-логической эволюции / А.Я. Аноприенко // Доклад на международной научной конференции «Моделирование-2010» (Киев, 13–14 мая 2010 г.). – Киев: Институт проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины, 2010. – С. 150 – 161.

Стаття надійшла до редакції 26.12.2011