УДК 004.942

А.Я. Аноприенко, к.т.н.,
С.В. Иваница, асп.

ГИБКАЯ РАЗРЯДНОСТЬ И ПОСТБИНАРНЫЕ ФОРМАТЫ ПРЕДСТАВЛЕНИЯ ВЕЩЕСТВЕННЫХ ЧИСЕЛ

Донецкий национальный технический университет,
anoprien@cs.dgtu.donetsk.ua, isv@cs.dgtu.donetsk.ua

Рассмотрена актуальность перехода к постбинарному компьютингу 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 с плавающей запятой.

Введение

Современные компьютеры ориентированы на операции с числами, представленными всего в двух форматах: целые числа и числа с плавающей запятой. 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 чисел. В итоге результат нескольких последовательных операций может оказаться совершенно неверным.

Подтверждением этого является относительно 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:

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

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:

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

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! Символьный и численный процессоры математического пакета Mathcad также оказались бессильны.

Другим примером является вычисление 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 актуальность перехода к постбинарному компьютингу и «гибкой разрядности» при кодировании количественных значений в современных компьютерных системах и предлагаются постбинарные форматы кодирования вещественных чисел.

От бинарного к постбинарному компьютингу

Термин «компьютинг» сложился в бинарную эпоху, в связи с чем под этим термином традиционно предполагается **бинарный компьютинг**, в основе которого лежит использование двоичной логики и двоичной системы счисления. Целесообразным и оправданным в настоящее время следует также признать и использование термина **постбинарный компьютинг**, 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 рамках кодо-логического подхода, развиваемого авторами данной статьи, важнейшими предпосылками для системного исследования проблематики постбинарного компьютинга 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 [5].

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

В-третьих, введение в научный оборот таких взаимосвязанных понятий как тетралогика и тетракоды, что 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 компьютинга в контексте кодо-логической эволюции.

Логические, алгоритмические и прочие предпосылки перехода к постбинарному компьютингу начали складываться уже в первой половине XX века в ходе интенсивного развития средств и методов бинарного компьютинга 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 компьютингу, основанному на использовании различных форм гиперлогики и гиперкодов, которые могут рассматриваться как обобщение тетралогики и тетракодов применительно к многомерным логическим пространствам.

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 выполнять следующие операции:

– **увеличение (или выравнивание) разрядности** во избежание переполнения разрядов результата и выполнения 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 операции можно эффективно использовать при разработке и реализации **постбинарных методов вычислений**, основанных на постбинарном представлении количественных значений [6, 7]. Однако введение данных 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).

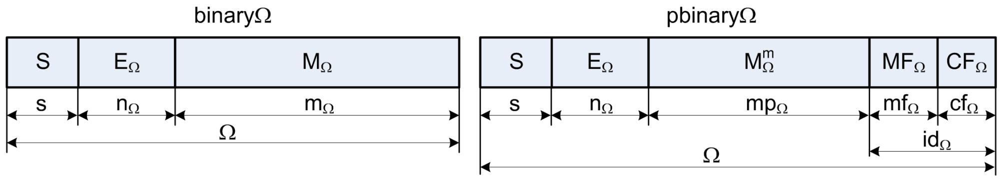

Рис. 1. Структура числа в формате binary**Ω** и в эквивалентном ему формате pbinary**Ω**: **Ω** — количество разрядов формата числа; S — знак; E_Ω — смещенная экспонента; M_Ω и $M^m{}_\Omega$ — остаток мантиссы соответствующих форматов; MF_Ω, CF_Ω — модификатор и код формата

В предложенных постбинарных форматах поле мантиссы числа претерпело модификацию путем выделения необходимого количества бит для **идентификатора** формата (поле id_Ω), состоящего из **модификатора** MF_Ω (внутренняя модификация формата pbinary**Ω**) и **кода формата** CF_Ω (признак принадлежности к формату pbinary**Ω**).

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

В то же время, в зависимости от модификатора MF_Ω, формат pbinary**Ω** может содержать в себе другие подформаты представления данных. Для таких форматов приняты обозначения в виде pbinary**Ω**/Ψϕ, 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).

Рассмотрим структуру, особенности формирования и назначение каждой из представленных модификаций постбинарного формата pbinary**Ω** (в качестве примера на рис. 2 представлены все модификации формата pbinary128):

• **pbinaryΩ** — фактически число формата binary**Ω** имеющее модифицированную (уменьшенную на id_Ω разрядов) мантиссу. Для постбинарного формата pbinary**Ω** справедливы следующие соотношения, связанные с определением количества разрядов полей после модификации: $id_\Omega = \Omega/16$ — разрядность идентификатора формата; $cf_\Omega = \log_2 id_\Omega = \log_2 \frac{\Omega}{16}$ — 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.

Поля знака (S) и смещенной экспоненты (E_Ω) переносятся в модифицированный формат без изменения. В таблице 2 показаны разрядности форматов pbinary**Ω** по отношению к binary**Ω**.

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

$$E_{\Omega} = \exp_2 + \text{offset}_2\,, \tag{1}$$

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

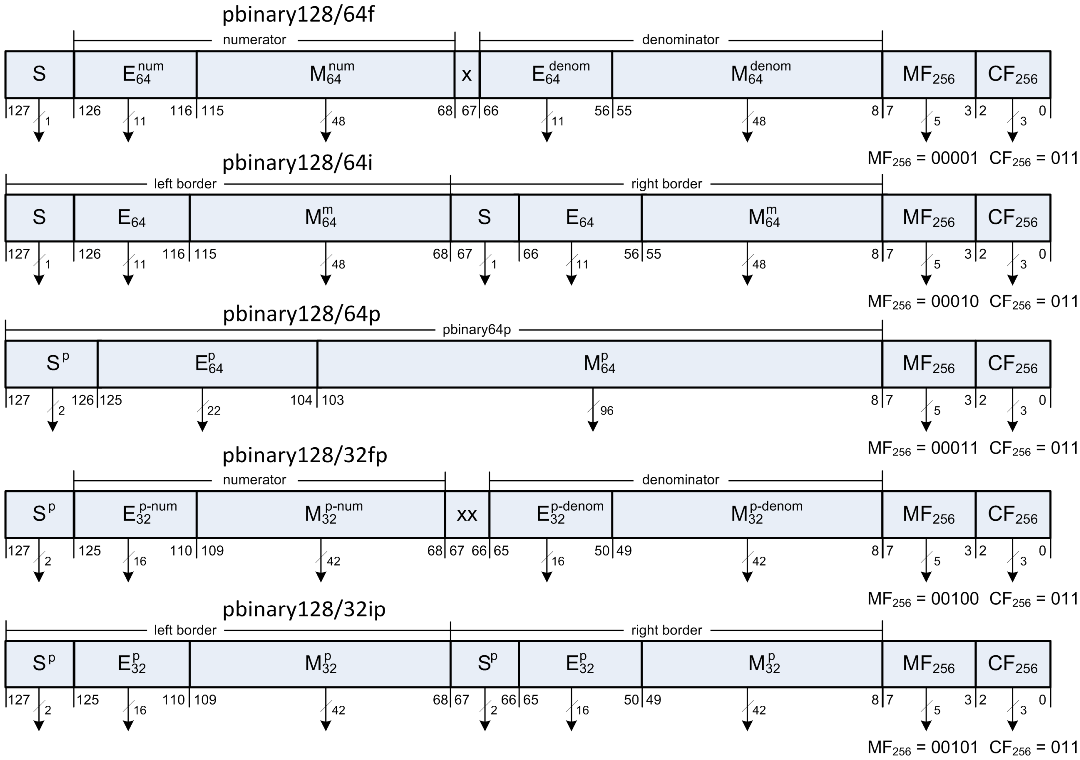

Рис. 2. Модификации формата pbinary64

Таблица 2

Разрядность pbinaryΩ 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 $\pm 1,\text{xx...x} \cdot \exp_2$). В формате полей pbinaryΩ область нормализованных чисел лежит в пределах: S = x (любое значение); E_{Ω} = 000…01 ÷ 111…10; M_{Ω}^{m} = 000…00 ÷ 111…11.

Таким образом, для формата pbinaryΩ справедливы следующие соотношения:

– двоичное нормализованное число *dn*:

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

– десятичное нормализованное число *xn*:

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

Аналогично binaryΩ, формат pbinaryΩ имеет ряд исключительных чисел, к которым нельзя применять формулы (2, 3):

1) Положительный и отрицательный ноль: +0 (S = 0; E_Ω = 0; M^m_Ω = 0) и –0 (S = 1; E_Ω = 0; M^m_Ω = 0). Большинство программных средств эти нули не различает (поскольку в этом нет необходимости), считая их просто нулевыми значениями.

2) Положительная и отрицательная бесконечности (+Infinity, –Infinity): +∞ (S = 0; E_Ω = 111…11; M^m_Ω = 0) и –∞ (S = 1; E_Ω = 111…11; M^m_Ω = 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_Ω = 111…11 и M^m_Ω — любое ненулевое значение, различают +NaN (S = 1) и –NaN (S = 0).

4) Ненормализованные (денормализованные) числа — числа, мантиссы которых лежат в диапазоне $0{,}1 \le M^m_\Omega < 1$. 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*:

$$dd = (-1)^S \cdot 0{,}M^m_\Omega \cdot \exp_2^{-\mathrm{offset}_2} \tag{4}$$

– десятичное ненормализованное число *xd*:

$$xd = (-1)^S \cdot 2^{1-\mathrm{offset}_{10}} \cdot \frac{(M^m_\Omega)_{10}}{2^{\mathrm{mp}_\Omega}} \tag{5}$$

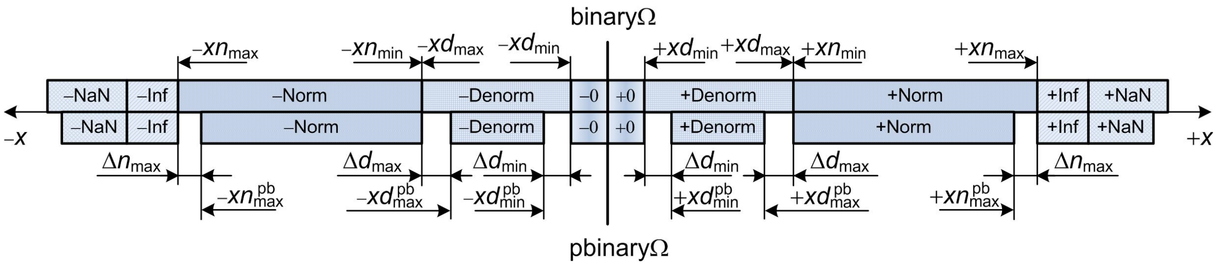

Рис. 3. Отображение чисел форматов binaryΩ и pbinaryΩ на числовой оси: Norm, Denorm — области нормализованных и ненормализованных чисел; pb — признак постбинарного формата; *xd* и *xn* — ненормализованные и нормализованные числа

Совокупности полей ±Norm и ±Denorm ($[-xn^{pb}_{max};\ -xd^{pb}_{min}] \cup [+xd^{pb}_{min}; +xn^{pb}_{max}]$) являются диапазоном представления положительных или отрицательных чисел формата pbinaryΩ на всей числовой оси. Используя соотношения (3) и (5), можно получить десятичные значения модулей минимальной xd^{pb}_{min} (значение по модулю наименьшего ненормализованного числа) и максимальной xn^{pb}_{max} (значение по модулю наибольшего нормализованного числа) границ диапазона представления чисел в формате pbinaryΩ. 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 binaryΩ. Используя формулы (3, 5) получим абсолютные погрешности границ диапазона представленя чисел по отношению к формату binaryΩ (xn_{max}, xd_{max} и xd_{min} — значения модулей соответствующих границ диапазона чисел формата binaryΩ):

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

$$\Delta d_{min} = \left| xd_{min} - xd^{pb}_{min} \right| = 2^{1-\mathrm{offset}_{10}-\mathrm{mp}_\Omega} \cdot (1 - 2^{-\mathrm{id}_\Omega})\,; \tag{6}$$

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

$$\Delta n_{max} = \left| xn_{max} - xn^{pb}_{max} \right| = 2^{\mathrm{offset}_{10}-\mathrm{mp}_\Omega} \cdot (1 - 2^{-\mathrm{id}_\Omega})\,. \tag{7}$$

Абсолютная погрешность представления максимального ненормализованного Ω-разрядного числа Δd_{max} соответствует формуле (6), следовательно $\Delta d_{max} = \Delta d_{min}$. Последнее равенство можно объяснить тем, что разница в представлении ненормализованных чисел pbinaryΩ и binaryΩ обусловлена одинаковой погрешностью представления значений мантисс постбинарного и бинарного форматов (все ненормализованные числа имеют нулевое значение порядка) на протяжении всего диапазона ±Denorm. Таким образом,

$$\left[xd_{\min}^{\mathrm{pb}};\, xd_{\max}^{\mathrm{pb}}\right]=\left[xd_{\min}+\Delta d_{\min};\, xd_{\max}-\Delta d_{\min}\right]=\left[2^{\mathrm{id}_\Omega}\cdot xd_{\min};\, 2^{-\mathrm{id}_\Omega}\cdot xd_{\max}\right]. \quad (8)$$

Равенство (8) показывает, что модуль минимального ненормализованного числа формата pbinaryΩ больше модуля аналогичного числа формата binaryΩ в 2^{id_Ω} раза, и наоборот, модуль максимального ненормализованного числа формата pbinaryΩ меньше модуля аналогичного числа формата binaryΩ в 2^{id_Ω} раза. Таким образом, диапазон ±Denorm формата pbinaryΩ уже соответствующего диапазона формата binaryΩ в $2^{\mathrm{id}_\Omega+1}$ раза.

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

$$\left[xn_{\min}^{\mathrm{pb}};\, xn_{\max}^{\mathrm{pb}}\right]=\left[xn_{\min};\, xn_{\max}-\Delta n_{\max}\right]=\left[xn_{\min};\, \frac{2^{-(\mathrm{mp}_\Omega+1)-\mathrm{id}_\Omega}-1}{2^{-(\mathrm{mp}_\Omega+1)}-1}\cdot xn_{\max}\right]. \quad (9)$$

Рассмотрим функцию $f(a,b)=\dfrac{2^{-(\mathrm{mp}_\Omega+1)-\mathrm{id}_\Omega}-1}{2^{-(\mathrm{mp}_\Omega+1)}-1}$. Имеем $\lim\limits_{a\to\infty}f(a,b)=1$, откуда следует, что $f(a,b)\approx 1$ при возрастающем аргументе a. Соответственно, для значений mp_Ω и id_Ω представленных форматов справедливо соотношение

$$\frac{2^{-(\mathrm{mp}_\Omega+1)-\mathrm{id}_\Omega}-1}{2^{-(\mathrm{mp}_\Omega+1)}-1}\approx 1, \quad (10)$$

причем данное приближенное равенство стремится к строгому равенству при возрастании значения mp_Ω. Исходя из (10), выражение (9) можно записать в виде

$$\left[xn_{\min}^{\mathrm{pb}};\, xn_{\max}^{\mathrm{pb}}\right]=\left[xn_{\min};\, \approx xn_{\max}\right]\ \Rightarrow\ \left[xn_{\min}^{\mathrm{pb}};\, xn_{\max}^{\mathrm{pb}}\right]\approx\left[xn_{\min};\, xn_{\max}\right] \quad (11)$$

Десятичные значения максимальных и минимальных чисел формата pbinaryΩ, их абсолютной погрешности по отношению к формату binaryΩ представлены в таблице 3.

- **pbinaryΩ/Ψf** — дробное Ω-разрядное число, состоящее из числителя (numerator), знаменателя (denominator) Ψ-разрядной точности (т. е. двух чисел формата pbinaryΨ) и общего знака для дроби. Значения Ω и Ψ связаны следующим соотношением: Ψ = Ω/2 при Ω = {64, 128, 256, …}.

- **pbinaryΩ/Ψi** — интервальный Ω-разрядный формат числа, содержащее два полноценных числа формата pbinaryΨ (Ψ = Ω/2, Ω = {64, 128, 256, …}), которые являются левой (left border) и правой (right border) границами интервала [8]. Заполнение полей данного формата аналогично дробному, за исключением наличия знаковых полей для каждого значения границ интервала. Числа в формате pbinaryΩ/Ψi 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 исключат возможные ошибки округления [9].
- **pbinaryΩ/Ψp** — постбинарное Ψ-разрядное число (Ψ = Ω/2, Ω = {32, 64, 128, 256, …}). В таком числе в качестве основного постбинарного формата рассматривается тетракод, каждый разряд которого представлен тетритом [10], 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 потребуется два бита, поэтому расширение числа формата pbinaryΨ до постбинарного Ω-разрядного формата pbinaryΩ/Ψp происходит за счет удвоения разрядности знака, порядка и модифицированной мантиссы.

• **pbinaryΩ/Ψfp** — представляет формат постбинарного дробного Ψ-разрядного числа (Ψ = Ω/4, Ω = {128, 256, …}). Формируется объединением описанных выше преобразований pbinaryΩ/Ψf ∪ pbinaryΩ/Ψp.

• **pbinaryΩ/Ψip** — представляет формат постбинарного интервального Ψ-разрядного числа (Ψ = Ω/4, Ω = {128, 256, …}). Формирование полей данного числа происходит по схеме: pbinaryΩ/Ψi ∪ pbinaryΩ/Ψp.

Заключение

В контексте перехода к постбинарному компьютингу многообразие представления логических и численных значений будет неизбежно возрастать 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 компьютинга.

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. Важнейшими результатом этого должно стать кардинальное повышение надежности вычислений и оптимизация их разрядности.

Список литературных источников:

1. Hammer R., Hocks M., Kulisch U., Ratz D. — Numerical Toolbox for Verified Computing (Pascal-XSC Programs) Springer-Verlag Berlin Heidelberg, 1993, c.8.
2. Loh E., Walster G. Rump’s Example Revisited // Reliable Computing 8: 2002, p. 245–248.
3. Аноприенко А.Я., Гранковский В.А., Иваница С.В. Пример Румпа в контексте традиционных, интервальных и постбинарных вычислений // Научные труды Донецкого национального технического университета. Серия «Проблемы моделирования и автоматизации проектирования динамических систем» (МАП-2011). Выпуск 9 (179): Донецк: ДонНТУ, 2011. С. 324–343.
4. Петров Ю.П. Обеспечение надежности и достоверности компьютерных расчетов. — СПб: БХВ-Петербург, 2008. — 160 с.
5. Аноприенко А.Я. Расширенный кодо-логический базис компьютерного моделирования / В кн. “Информатика, кибернетика и вычислительная техника (ИКВТ-97). Сборник научных трудов ДонГТУ.” Выпуск 1. Донецк, ДонГТУ. — 1997. — С. 59–64.
6. Аноприенко А.Я. Обобщенный кодо-логический базис в вычислительном моделировании и представлении знаний: эволюция идеи и перспективы развития // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ-2005) выпуск 93: — Донецк: ДонНТУ, 2005. С. 289–316.
7. Аноприенко А.Я., Иваница С.В. Особенности постбинарного кодирования на примере интервального представления результатов вычислений по формуле Бэйли-Боруэйна-Плаффа // Научные труды Донецкого национального технического университета. Серия: «Информатика, кибернетика и вычислительная техника» (ИКВТ-2010). Выпуск 11 (164). — Донецк: ДонНТУ, 2010. С. 19-23.
8. Аноприенко А.Я., Иваница С.В. Интервальные вычисления и перспективы их развития в контексте кодо-логической эволюции // Научные труды Донецкого национального технического университета. Серия «Проблемы моделирования и автоматизации проектирования динамических систем» (МАП-2010). Выпуск 8 (168): Донецк: ДонНТУ, 2010. С. 150–160.
9. Moore R.E. Interval analysis. Eiiglewood Cliffs / R.E. Moore — N.J.:Prentic-e-llall, 1966.
10. Иваница С.В., Аноприенко А.Я. Особенности реализации операций тетралогики // Научные труды Донецкого национального технического университета. Серия: «Информатика, кибернетика и вычислительная техника» (ИКВТ-2011). Выпуск 13 (185). – Донецк: ДонНТУ, 2011. С. 134-140.

Как правильно ссылаться на данную статью:

Аноприенко А.Я., Иваница С.В. Гибкая разрядность и постбинарные форматы представления вещественных чисел // Вестник Инженерной Академии Украины. Теоретический и научно-практический журнал Инженерной Академии Украины. Выпуск 1, 2012. С. 92-98.