Функціональний ЦАП дає можливість вимірювати залежність $\sin\varphi$ в межах одного квадранту. Такі ж елементи використовуються для регулювання залежності $\sin\varphi$ у всіх чотирьох квадрантах, але у третьому та четвертому для отримання від'ємних величин використовується інвертор, що встановлюється у вхідному або вихідному колі. Також всі елементи приведеної схеми використовуються для побудови дискретного аналога косинусного потенціометра, оскільки $\cos\varphi=\sin(\frac{\pi}{2}+\varphi)$. Таким чином, розглянута схема може повністю забезпечити одночасне моделювання як синусного, так і косинусного перетворювачів.

Висновок

В даній роботі ми розглянули основні переваги використання функціональних ЦАП в сучасних автоматизованих системах керування. В роботі було досліджено використання кусково-лінійної апроксимації для відображення нелінійних функціональних залежностей на прикладі схеми, що є дискретним аналогом синусного потенціометра.

ЛІТЕРАТУРА

1. Довгалюк Б. П. Електронні промислові пристрої: Навчальний посібник для вузів; частина 1. — Дніпродзержинськ : ДДТУ. 1998. — 145 с., іл.
2. Функциональные цифроаналоговые преобразователи: принципы построения / В. М. Сапельников, Р. А. Хакимов, А. А. Газизов, М. А. Шабанов // Датчики и системы. 2007. — №7. — С.46—58.
3. Электроника: справочная книга / Ю. А. Быстров. — СПб. : Энергоатомиздат. 1996. — 544 с.
4. Темников Ф. Е., Афонин В. А., Дмитриев В. И. Теоретические основы информационной техники. — М. : Энергия. 1979. — 512 с.

Арутюнов В. О. Электрические измерительные приборы и измерения / В. О. Арутюнов. — М. : Госэнергоиздат. 1958. — 632 с.

.

пост.21.04.15

Вычисление граничных значений субнормальных чисел в IEEE-стандарте

И. И. ЖУЛЬКОВСКАЯ, О. А. ЖУЛЬКОВСКИЙ, Р. Г. ШАГАНЕНКО

Днепродзержинский государственный технический университет

В работе описано машинное представление и особенности использования субнормальных чисел в стандарте *IEEE* 754. Получены формулы для вычисления и рассчитаны граничные значения (максимальные и минимальные) субнормальных чисел в десятичной системе счисления для различных форматов. Показано, что использование субнормальных чисел, существенно замедляет работу всех современных процессоров. Очевидный выход из сложившейся ситуации, когда предлагаемая субнормальными числами точность не является необходимой, – их программное обнуление.

У роботі описано машинне представлення та особливості використання субнормальних чисел у стандарті IEEE 754. Отримано формули для обчислення і розраховані граничні значення (максимальні і мінімальні) субнормальних чисел в десятковій системі числення для різних форматів. Показано, що використання субнормальних чисел суттєво уповільнює роботу всіх сучасних процесорів. Очевидний вихід із ситуації, коли запропонована точність не є необхідною, – їх програмне обнуління.

In this paper we describe the computer representation and features of the use of subnormal numbers in standard IEEE 754. The formulas for calculating and calculated boundary values (maximum and minimum) of subnormal numbers in the decimal system for different formats. It is shown that the use of subnormal numbers, significantly slows down all modern processors. The obvious solution to the situation when the proposed subnormal numbers precision is not necessary, – their software ignored.

Общая характеристика проблемы. В современном мире информационный сектор экономики развивается значительно быстрее других отраслей. В частности, необычайно высокими темпами развивается область суперкомпьютеров и высокопроизводительных вычислений. Динамику роста производительности современных суперкомпьютеров можно проследить по данным регулярного списка TOP500 [1] – рейтинга с описанием 500 наимощнейших общественно известных компьютерных систем мира. Основой для рейтинга являются результаты исполнения теста LINPACK (HPL), решающего большие системы линейных алгебраических уравнений.

Революционный технологический прорыв сразу на нескольких направлениях (передовая архитектура с параллелизацией вычислений, быстрые процессоры и межсоединения, новые подходы к программированию) позволил создать первую в истории компьютерную систему *ASCI Red* (США, 1999 г.) производительностью в один терафлопс (10^{12} операций с плавающей запятой в секунду). Уже к 2008 г. в США появился первый петафлопсный (10^{15} флопс) компьютер – система *Roadrunner* Лос-Аламосской национальной лаборатории. При сохранении нынешней скорости прогресса суперкомпьютеров достижение следующего рубежа производитель-

ности в один экзафлопс (10^{18} флопс) ожидается к 2019, а в один зеттафлопс (10^{21} флокс) – примерно к 2030 г.

С ростом производительности современных ЭВМ расширяется спектр и сложность решаемых с их помощью задач и, особенно, прикладных, повышаются требования к точности компьютерных вычислений. В связи с этим проблему повышения производительности ЭВМ необходимо решать в тесной взаимосвязи с задачами повышения точности вычислений.

Большинство компьютерных вычислений проводится в арифметике с плавающей запятой, где ошибки неизбежны. Известны случаи, когда вычислительные ошибки приводили к техногенным катастрофам и авариям с колоссальными убытками и человеческими жертвами.

Краткий аналитический обзор. Численное решение большинства практических задач сопряжено, как правило, с выполнением операций над действительными (вещественными) числами. На практике в научных и инженерных расчетах действительные числа представляются в компьютере конечными дробями в виде чисел с плавающей запятой.

Наиболее часто используется представление действительных чисел в виде двоичных чисел с плавающей запятой, описанное в стандарте *IEEE* 754 (*IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Std* 754-1985). Этот, наиболее распространенный стандарт для вычислений с плавающей запятой, используется многими современными аппаратными и программными средствами ЭВМ.

В связи с реорганизацией ассоциации *IEEE* из международной общественной инженерной организации в коммерческую структуру авторские стандарты *IEEE* стали коммерческим продуктом и не находятся в свободном распространении и обращении.

Не все действительные числа представимы, поэтому большинство вычислений с числами с плавающей запятой являются приближенными. Расчеты с погрешностью приводят к накоплению промежуточных ошибок вычислений, которые в определенных задачах могут полностью исказить достоверность окончательных результатов.

Из всех математических функций стандартом *IEEE* 754 описывается только извлечение квадратного корня. Это приводит к непереносимости и ненадежной работе приложений, использующих другие функции, не затронутые в стандарте *IEEE* 754. Отсюда и столь широкий интерес [2] к стандартизации реализаций большого количества математических функций, работающих с числами с плавающей запятой в форматах *IEEE* 754. Множество работ [3] посвящено также тестированию на соответствие стандарту *IEEE* 754, вопросам погрешностей результатов при вычислении функций [4], путям минимизации ошибок округления и потери точности [5]. Ранее [6] авторами данной работы исследован и описан алгоритм современного подхода к формированию машинного представления и хранения числовой информации в формате с плавающей запятой.

Перечисленные работы содержат общий обзор и анализ стандарта *IEEE* 754. Однако в них нет описания актуального алгоритма вычисления граничных значений числовой информации в формате с плавающей запятой.

Цель работы (постановка задачи). Перед данным исследованием ставится задача рассмотрения особенностей представления, а также вычисления граничных (максимальных и минимальных) значений субнормальных чисел в стандарте *IEEE* 754.

Результаты работы. Согласно стандарту *IEEE* 754 двоичное число с плавающей запятой (*binary floating point number*) – битовая строка, характеризующаяся тремя составляющими: знаком (*sign*), знаковым порядком или экспонентой (*signed exponent*) и мантиссой (*significand*). Знаковый бит равен 0 для положительных чисел, и 1 – для отрицательных.

Так, число с плавающей запятой можно представить в следующем виде:

$$A = \pm a_0.a_1a_2a_3..a_{n-1} \times S^e \qquad (1)$$

где S – основа системы счисления;

n – число значащих разрядов мантиссы;

a_i – цифры ($0 \le a_i < S$);

e – порядок или экспонента (не путать с числом e).

Однако в такой форме произвольное действительное число записывается неоднозначно, оно зависит от «положения запятой». Поэтому для увеличения количества значащих цифр при представлении действительного числа и предотвращения переполнения при выполнении арифметических операций мантиссу нормализуют. В нормализованном двоичном числе старший разряд всегда равен единице, поэтому в памяти его можно не хранить. Это позволяет при сохранении действительных чисел не записывать в память заведомо известный старший разряд, а за счет освободившегося бита сохранить еще один дополнительный разряд мантиссы (перед вычислениями в процессоре «отброшенная» единица восстанавливается). Такой метод хранения носит название «скрытая единица» (*hidden bit*).

Стандарт *IEEE* 754 определяют несколько возможных типов чисел с плавающей запятой, из которых чаще всего используются числа одинарной точности (*single precision*), числа двойной точности (*double precision*) и числа двойной расширенной точности (*double-extended precision)*. Они отличаются диапазоном представимых в них значений. Стандарт однозначно определяет основные параметры форматов представления чисел с плавающей запятой.

Параметры базовых форматов приведены в *табл. 1.*

Особенность представления действительных чисел со скрытой единицей в том, что имеется довольно большой разрыв между нулем и ближайшим к нему представимым числом – потеря значимости (*underflow*) около нуля. Это обстоятельство может приводить к ошибкам при работе с малыми величинами.

Таблица 1. Параметры базовых форматов чисел с плавающей запятой

Формат	**Всего бит**	**Бит в порядке**	**Бит в мантиссе**
single	32	8	23
double	64	11	52
extended	80	15	64

В стандарте *IEEE* 754 для того, чтобы не хранить в ячейке памяти знак порядка, он представляется не как целое число со знаком в явном виде, а в виде беззнакового числа, называемого смещенным порядком или характеристикой. При этом характеристика (X) отличается от порядка на некоторую фиксированную для данного формата величину, называемую смещением (b):

$$X=e+b. \qquad (2)$$

В общем виде, если экспонента действительного числа занимает k бит и находится в промежутке от 0 до 2^{k-1}, то величина смещения порядка или смещения экспоненты (*exponent bias*) определяется по формуле:

$$b = 2^{k-1} - 1. \qquad (3)$$

Для повышения точности вычислений при работе с «маленькими» числами в стандарте предусмотрена возможность использования так называемого мягкого исчезновения порядка – мягкого антипереполнения. Суть этого подхода состоит в трактовке кодов с нулевым порядком $000\ldots00_2$ и ненулевой мантиссой как специальных значений, которые принято называть денормализованными числами. При этом, используется следующее правило: если код содержит все нули в отведенных под порядок разрядах $000\ldots00_2$ и ненулевую мантиссу, то считается, что порядок числа равен $000\ldots001_2$, а явно не указанная в поле целая часть мантиссы принимается равной нулю. Следовательно, мантисса ненормализованная, и ее код фактически совпадает с числом. В версии стандарта *IEEE* 754-2008 денормализованные числа (*denormal* или *denormalized numbers*) были переименованы в *subnormal numbers*, т.е. в числа, меньшие «нормальных». Поэтому их иногда еще называют «субнормальными».

Исходя из этого правила, минимальное субнормальное число в формате *single* можно представить в показанном на *рис. 1* виде.

Рис. 1. Представление минимального субнормального числа в формате *single*

Очевидно, что мантисса минимального субнормального числа содержит единицу в самом младшем бите, а все остальные биты мантиссы содержат нули. Следовательно, значение мантиссы в десятичной системе счисления равняется 2^{-23}. Смещенный порядок минимального субнормального числа равняется единице в двоичном коде и отличается от порядка на величину смещения.

Воспользовавшись формулами (2) и (3), можно рассчитать значение порядка в десятичной системе счисления:

$$e = 2^0 - \left(2^{8-1} - 1\right) = -126.$$

Соответственно, используя формулу (1), можно рассчитать значение минимального субнормального числа в формате *single* в десятичной системе счисления:

$$A_{min} = \pm 2^{-23} \cdot 2^{-126} = \pm 2^{-149} \approx \pm 1.401298E-45 .$$

Таким образом, можно получить общую формулу для расчета значений минимальных субнормальных чисел в различных форматах:

$$A_{min} = \pm 2^{-(n-k-1)} \cdot 2^{1-b} = \pm 2^{-b-n+k+2} . \qquad (5)$$

Используя формулу (5) получим значения минимальных субнормальных чисел для различных форматов:

– для чисел двойной точности

$$A_{min} = \pm 2^{-1023-64+11+2} = \pm 2^{-1074} \approx \pm 4.940656E-324;$$

– для чисел двойной расширенной точности

$$A_{min} = \pm 2^{-16383-80+15+2} = \pm 2^{-16446} \approx \pm 3.6451995E-4951.$$

Мантисса максимального субнормального числа содержит единицы во всех битах (см. *рис. 2*).

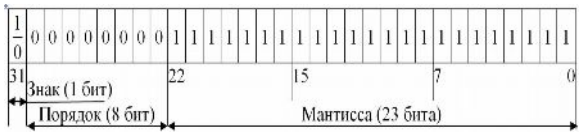

Рис. 2. Представление максимального субнормального числа в формате *single*

Смещенный порядок для всех форматов равняется единице в двоичном коде. Поэтому, воспользовавшись формулой (5), получим общую формулу для расчета значений максимальных субнормальных чисел в различных форматах:

$$A_{max} = \pm\left(1 - 2^{-(n-k-1)}\right) \cdot 2^{1-b} . \qquad (6)$$

Используя формулу (6) получим значения максимальных субнормальных чисел в различных форматах:

– для чисел одинарной точности

$$A_{max} = \pm\left(1 - 2^{-(32-8-1)}\right) \cdot 2^{1-127} = \pm\left(1 - 2^{-23}\right) \cdot 2^{-126} \approx \pm 1.175494E38$$

– для чисел двойной точности

$$A_{max} = \pm\left(1 - 2^{-(64-11-1)}\right) \cdot 2^{1-1023} = \pm\left(1 - 2^{-52}\right) \cdot 2^{-1022} \approx$$

$$\approx \pm 2.225073E-308 ;$$

– для чисел двойной расширенной точности

$$A_{max} = \pm\left(1 - 2^{-(80-15-1)}\right) \cdot 2^{1-16383} = \pm\left(1 - 2^{-64}\right) \cdot 2^{-16382} \approx$$

$$\approx \pm 3.362103E-4932 .$$

Полученные граничные значения сведены в *табл.2.*

Таблица 2. Граничные значения субнормальных чисел с плавающей запятой

Формат	Максимальное значение	Минимальное значение
single	±1.175494E-38	±1.401298E-45
double	±2.225073E-308	±4.940656E-324
extended	±3.362103E-4932	±3.6451995E-4951

Значения максимальных субнормальных чисел точно совпадают с минимальными значениями нормализованных чисел в различных форматах. Таким образом, использование субнормальных чисел – это способ увеличения количества представимых числом с плавающей запятой значений около нуля для повышения точности вычислений.

Работа с субнормальными числами производится большинством процессоров посредством специального микрокода – более медленного, чем стандартные

инструкции математического сопроцессора. Это накладывает дополнительные сложности при реализации АЛУ в процессоре. Поэтому эта функциональность стала камнем преткновения при разработке стандарта и встретила самое сильное сопротивление со стороны разработчиков *hard ware*.

В современных процессорах обработка субнормальных чисел происходит в десятки раз медленнее, чем обработка нормализованных чисел. Проводимые исследования [7] по оценке влияния использования субнормальных чисел при вычислениях с плавающей запятой на работу процессора показывают значительное ухудшение производительности в любой современной микро-архитектуре.

Наихудшие результаты тестирования некоторых процессоров по данным исследованиям приведены в *табл. 3*.

Поскольку в стандартных форматах субнормальные числа оказываются действительно очень маленькими и практически никак не влияют на результат некоторых вычислений (при этом заметно замедляя их скорость), то иногда они просто игнорируются. При этом используются два простых механизма, получивших называние *Flush-to-zero* (*FTZ*) и *Denormals-are-zero* (*DAZ*). Первый механизм заставляет операции возвращать ноль, как только становится ясно, что результат будет субнормальным. Второй механизм заставляет операции рассматривать поступающие на вход субнормальные числа как нули.

Таблица 3. Ухудшение производительности работы процессоров при обработке субнормальных чисел

Производитель	**Процессор**	**Замедление (разы)**
AMD	K6	1,4
IBM	PowerPC 970	2,4
AMD	Athlon	6,0
Intel	Pentium 3	15,8
AMD	Athlon 64	21,4
AMD	Opteron64	23,8
Intel	Core Duo	44,2
Intel	P4 Xeon	97,9
Intel	Pentium 4	131,0

Ярким примером подобного «отсечения» субнормальных чисел могут послужить видеокарты, в которых резкое падение скорости вычислений в сотню раз недопустимо. Аналогично поступают в областях, связанных с обработкой звука, отбрасывая фоновый шум низкой амплитуды, который, будучи неразличимым для слуха, просто загружает процессор.

Выводы

В работе описано машинное представление и особенности использования субнормальных чисел в стандарте *IEEE* 754. Получены формулы для вычисления и рассчитаны граничные значения (максимальные и минимальные) субнормальных чисел в десятичной системе счисления для различных форматов. Показано, что использование субнормальных чисел, существенно замедляет работу всех современных процессоров. Очевидный выход из сложившейся ситуации, когда предлагаемая субнормальными числами точность не является необходимой, – их обнуление путем программного *FTZ*.

Множество целых чисел – бесконечно, но всегда при решении конкретной задачи можно подобрать необходимое число бит для представления произвольного целого числа. Множество же действительных чисел не только бесконечно, но еще и непрерывно. Поэтому, не зависимо от числа задействованных двоичных разрядов, возникают ситуации неточного их представления. Числа с плавающей запятой — один из возможных способов представления действительных чисел, который является компромиссом между точностью и диапазоном принимаемых значений.

Поэтому от разработчиков программного обеспечения требуется предметное понимание современного подхода к формированию машинного представления и хранения числовой информации в формате с плавающей запятой. Очень часто именно эти вопросы имеют решающее значение в достижении максимально эффективных характеристик высокопроизводительных приложений.

ЛИТЕРАТУРА

1. TOP500 Supercomputing Sites. Режим доступа: http://www.top500.org.
2. Proposal for a standardization of mathematical function implementation in floating-point arithmetic / D. Defour, G. Hanrot, V. Lefevre, J.-M. Muller and other // Numerical Algorithms. — 2004. — №37(1–4). — P.367—375.
3. Searching Worst Cases of a One-Variable Function Using Lattice Reduction / D. Stehle, V. Lefevre, P. Zimmermann // IEEE Transactions on Computers. — 2005. — №54 (3). — P.340—346.
4. Оцінка точності обчислень спеціальних функцій при розробці комп'ютерних програм математичного моделювання / О. Я. Ніконов, О. В. Мнушка, В. М. Савченко // Вісник НТУ «ХПІ». Тематичний випуск : Інформатика і моделювання. — Харків : НТУ. — 2011. — № 17. — С.115—121.
5. 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. — Киев : 2012. — С.92—98.
6. Жульковская И. И., Жульковский О. А. Алгоритм формирования машинного представления числовых данных в формате с плавающей запятой // Математичне моделювання. — Дніпродзержинськ : ДДТУ. 2013. — №2 (29). — С.69—72.
7. I. Dooley, L. Kale. Quantifying the Interference Caused by Subnormal Floating-Point // Values In OSIHPA Workshop. — 2006.

.

пост.28.04.2015