

АНАЛИЗ КОМПЬЮТЕРНЫХ ЧИСЛОВЫХ ТИПОВ ДАННЫХ В МУЛЬТИРАЗРЯДНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ, ПРИМЕНИТЕЛЬНО К ОПЕРАЦИИ Понижения РАЗРЯДНОСТИ ВЫЧИСЛЕНИЙ

ANALYSIS OF COMPUTER NUMERIC DATA TYPES IN MULTI-BIT COMPUTER SYSTEMS IN RELATION TO OPERATION OF COMPUTATION REDUCTION

D. Prikhodko

Summary. The article deals with mathematical features, which are possessed by computer numerical sets in relation to operation of computation reduction, for this purpose universal methods of compressing numerical data types from one bit to another are considered, at the same time operations of addition and multiplication on the subject of operation speed and speed are considered. Reductive formulas for the operation of reducing the bitness of calculations are described. This is a characteristic calculation for the AMD64 processor using matlab.

Keywords: reduction of bit length, mathematical features of the reduction of bit length operation.

Приходько Дмитрий Игоревич

*Магистр, инженер вычислительных и электронно-вычислительных машин, Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук, г. Москва
mitry1205@mail.ru*

Аннотация. В статье рассмотрены математические особенности, которыми обладают компьютерные числовые множества применительно к операции понижения разрядности вычислений, для этого рассмотрены универсальные способы сжатия числовых типов данных с одной разрядности на другую, при этом рассмотрены операции сложения и умножения на предмет быстродействия, скорости операции. Описаны редуцирующие формулы для операции снижения разрядности вычислений. Приведен характеристический расчет для процессора AMD64 с помощью программы matlab.

Ключевые слова: понижение разрядности, математические особенности операции понижения разрядности.

Введение

В рамках существующей модели при реализации механизмов смены разрядности вычислений самым сложным этапом является этап смены разрядности вычислений с точки зрения целостности числовых типов данных, по основной причине — все типы данных в современных вычислительных системах всегда преобразуются в числовые операции. Это связано с тем, что микропроцессор не знает, что такое класс, строка, функция, объект и т.д. Более того, т.к. при смене режима работы микропроцессора идет не только изменение размера адресов, которыми оперируют операционная система и программы, но и размерам самих чисел.

По этой причине смена режима работы вычислительной системы в первую очередь влечет за собой потерю данных из-за смены режима микропроцессора или разрядности вычислений, что в лучшем случае приводит к ошибкам в расчетах, а в худшем случае приводит к критическим ошибкам вычислительной системы.

Цель статьи

Рассмотреть базовые способы преобразования основных числовых типов данных в компьютере с точки зрения уменьшения размеров использования.

Методология

При выполнении смены разрядности вычислений, основной вопрос заключается в несовпадении доступных диапазонов и возможностей микропроцессора при работе с числовыми данными [1]. В свою очередь точность вычислительной системы ограничена 2-мя значениями:

1. Максимальным размером ячейки памяти.
2. Технические особенности представления типа данных.

С точки зрения операции снижения разрядности вычислений, все *типы данных* вписываются в имеющиеся ячейки памяти. Поэтому при смене разрядности вычислений происходит сужение размеров доступной

ячейки памяти [2], что неизбежно приводит к тому, что числовые типы данных, позволяющие работать с большими числами — начинают страдать от снижения точности.

Для преобразования типов данных существуют несколько методов. Из них наиболее очевидны следующие:

1. *Простое аппаратное отсечение разрядов от числа.* Самый простой способ для преобразования данных, и потому самый плохой, поскольку не гарантирует целостность данных вообще.
2. *Использование программных средств преобразований числовых множеств.*
3. *Комбинированный подход.* Способ объединяет оба описанных выше. Является сравнительно медленным по сравнению с программным методом.

Простое аппаратное отсечение разрядов от числа. Стандартный способ, который заключается в простом логическом отключении поврежденных триггеров без учета резервных копий и внутренней структуры типов данных, последствия которых могут быть фатальными как для программы, так и для операционной системы в целом.

Использование программных средств преобразований числовых множеств. Этот способ основан на том, что мы преобразуем, типы данных, которые недоступны для младшей разрядности вычислений (не хватает размера или самих регистров). Важно отметить, что преобразование типов данных без физической изоляции триггеров на практике невозможно, из-за наличия риска повторного использования поврежденного триггера. В чистом виде — не рекомендуется.

Комбинированный подход. Данный подход объединяет способ «Простое аппаратное отсечение разрядов от числа» и способ «Использование программных средств преобразований числовых множеств». Оба способа описаны в предыдущих абзацах, и их объединение выполнено с целью избавиться от недостатков каждого из способов. При этом алгоритм использования этого способа состоит из следующих этапов:

1. Выполнить преобразования всех больших типов данных в меньшие типы данных без потерь данных.
2. После того как было выполнено преобразование типов данных, выявить несжимаемые вычисления. Для них выполняются следующая последовательность действий:
 - а. Числа с фиксированной точкой и целые числа необходимо конвертировать в числа с плавающей точкой.

б. Числа с плавающей точкой необходимо конвертировать в доступные для выбранного режима микропроцессора типы данных. В исключительных случаях можно вместо регистров использовать специальные библиотеки для эмуляции поведения.

3. После выполнения пункта 2 выполняем окончательное преобразование числовых типов данных.
4. По первому способу приводим размерность регистров с одной режима работы микропроцессора на другой.

Самый главный недостаток заключается в том, что механизм преобразования достаточно трудоемкий с точки зрения производительности, что приводит к падению как программ, так и в самом худшем случае приводит к фатальной ошибке в операционной системе, что приводит либо к перезагрузки всей системы или её компонентов.

В рамках настоящей статьи будет разобран комбинированный способ переключения и его погрешности. *Целью статьи является анализ применения комбинированного подхода к основным числовым типам в компьютере и базовая оценка погрешности.*

Результаты

В частности, для данных чисел сначала рассмотрим символьные особенности формул, с помощью которых можно получить итоговые числовые данные. При этом несмотря на то, что числовые типы данных в компьютере имеют различные как математические аналоги [3], так и особенности их построения, можно отметить, что в рамках символьных деталей.

При этом для анализа будет рассмотрен базовый случай, когда новой разрядности (в меньшую сторону), когда разрядов достаточно.

Взамен падения точности идет увеличение скорости операций чтения/записи в триггеры регистра, поэтому скорость работы с числовыми операциями увеличивается. Для начала определим её увеличение в символьном виде применительно к операциям [4]:

1. Анализ операции умножения.
2. Анализ операции сложения

Для выполнения анализа операций можно использовать следующие формат записи:

1. $M1_1 = M'1_1 * 2^{k1_1} + a1_1;$
2. $M2_1 = M'2_1 * 2^{k2_1} + a2_1;$

Где:

1. $M1_1$ и $M2_1$ — исходные числа.
2. $M'1_1$ и $M'2_1$ — числа, преобразованные в другую разрядность вычислений (в первую очередь на пониженную разрядность). Компактный формат числа, который характеризуется тем, что
3. $k1_1$ ($k2_1$) — показатели конверсии (т.е. на что надо умножить, чтобы получить исходное число)
4. $a1_1$ и $a2_1$ — остатки чисел (сгорающая часть при понижении разрядности). Т.е. эти разряды будут обнулены в простейшем случае, при сохранении точности вычислений.

Поскольку для каждой модели и режима микропроцессора размерности ячеек памяти и регистров определены документацией, то, следовательно, можно говорить о том, что величины $a1_1$ и $a2_1$ являются исчисляемыми.

А раз числа $a1_1$ и $a2_1$ являются исчисляемыми, то значит что в рамках архитектуры микропроцессоров, которая характеризует модельный ряд, можно определить $a1_1$ и $a2_1$ для каждого режима микропроцессора, т.е. сделать её малой в рамках дозволенного архитектурой.

По этой причине можно отметить, что *операция понижения разрядности вычислений* представляет собой с точки зрения числовых типов данных переход от точных вычислений к приближенным вычислениям [5].

Тогда операции сложения и умножения согласно законам приближенных вычислений, приобретает значение, равное сумме разрядов $a1_1$ и $a2_1$.

Тем не менее, так как число разрядов, с помощью которого представляются числовые множества в компьютере в упрощенном анализе у нас уменьшено, то, следовательно, можно оценить приращение скорости с точки зрения выполнения операции, так как микропроцессору нужно задействовать меньшую часть АЛУ.

Разумеется, при оценке приращения скорости, вызванной потерей точности, необходимости проводить оценку отдельно для целых чисел, и чисел с плавающей точкой, так как каждая часть числа с плавающей точкой — мантисса, порядок, и для последнего стандарта на момент публикации IEEE-754–2019[6] записывается как целое число, а поэтому можно оценивать только целые числа.

Поэтому выполнить анализ на оптимизацию можно при помощи одного набора формул для выбранных типов чисел.

Анализ операции умножения. Операция умножение имеет несколько алгоритмов, начиная от самого простого варианта — $n^m = n^m$ (прямое произведение чисел) и заканчивая некоторыми продвинутыми алгоритмами, которые сокращают число операций внутри микропроцессора в несколько раз. Оценку операции умножения авторы выполняют на основе самого простого алгоритма. Причем для анализа достаточно оценить следующую дробь:

$$(M'1_1 * M'2_1) / ((M'1_1 * 2^{k1_1} + a1_1) * (M'2_1 * 2^{k2_1} + a2_1)).$$

Если устремить $M'1_1$ и $M'2_1$ к бесконечности, тогда результат анализа может быть выполнен в виде дроби:

$$1 / (2^{k1_1} * 2^{k2_1} + O(2^{k1_1} * 2^{k2_1})), \text{ т.е. дроби с } k1_1 \text{ и } k2_1 \text{ с некоторой надбавкой по скорости, которая устремлена к } 0. \text{ Следовательно, для оценки достаточно дроби } 1 / (2^{k1_1} * 2^{k2_1}).$$

Для чисел с плавающей точкой, особенности арифметики которых, были рассмотрены авторами в работах [7][8], можно рассмотреть в упрощенном виде как сумму дробей из мантиссы, порядка и возможно, экспоненты, если используется десятичное число из стандарта.

Поэтому, скорость при сдвинутых числах получается быстрее при умножении на помноженное количество откинутых разрядов у сомножителей как для типов float, так и для типов int на $1 / (2^{k1_1} * 2^{k2_1})$.

Анализ операции сложения. Для выполнения анализа будет использовано тоже представление чисел, которое было использовано для операции умножения. Для выполнения оценки операции умножения достаточно оценить дробь:

$$(M'1_1 + M'2_1) / ((M'1_1 * 2^{k1_1} + a1_1) + (M'2_1 * 2^{k2_1} + a2_1)).$$

Поделим дроби на числитель, получим:

$$1 / ((M'1_1 * 2^{k1_1} + M'2_1 * 2^{k2_1}) / (M'1_1 + M'2_1) + (a1_1 + a2_1) / (M'1_1 + M'2_1)).$$

Последнее выражение — величина младшего порядка, следовательно, можно пренебречь:

$$1 / (M'1_1 * 2^{k1_1} + M'2_1 * 2^{k2_1}) / (M'1_1 + M'2_1).$$

Для получения оценки теперь нужен анализ знаменателя:

$$(M'1_1 * 2^{k1_1} + M'2_1 * 2^{k2_1}) / (M'1_1 + M'2_1) = 2^{k1_1} / (1 + M'2_1 / M'1_1) + 2^{k2_1} / (M'1_1 / M'2_1)$$

Очевидно, что числители каждой из дробей больше порядка, а, следовательно, для оценки можно использовать следующую дробь:

$$1 / (2^{k1_1} + 2^{k2_1})$$

Поэтому скорость при сдвинутых числах увеличивается при сложении на значение, равное $(2^{k1_1} + 2^{k2_1})$, где $k1_1$ и $k2_1$ — порядок оставшихся разрядов, как для float, так и для int.

Т.е. в целом скорость возрастает на значение потери разрядов.

Теперь необходимо рассмотреть оптимизацию с точки зрения выполнения нескольких операций. Для этих целей воспользуемся редуктивными формулами.

Для описываемых формул введем следующие общие обозначения:

1. S — итоговое число.
2. N — исходное число.
3. i — индекс разряда числа.
4. k — число разрядов итогового числа.
5. ArI — исходная разрядность вычислений.
6. ArR — итоговая разрядность вычислений.
7. L — длина исходного числа или части числа.

Для целых чисел запишем общую символьную формулу, состоящую из суммы разрядов (см. формула 1).

$$S = \sum_{i=0}^n N(L-i) * p^i$$

Формула 1. Общая формула целых чисел

Для того, чтобы получить число на другом режиме микропроцессора или разрядности, обязательно учесть, что смена разрядности вычислений уменьшает число доступных разрядов в регистрах, и, следовательно, для записи чисел после смены разрядности вычислений необходимо использовать формулу 2.

$$S = \sum_{i=0}^k N(L-i) * p^k$$

$$k = \frac{ArI}{ArR} * L$$

Формула 2. Редуктивные формулы работы с целыми числами.

Как уже было описано выше, для смены разрядности вычислений с точки зрения анализа разрядности числа в упрощенном виде изображены на скриншоте формулы 3.

$$S_{\text{Мантиссы}} = \sum_{i=0}^k N(L_{\text{Мантиссы}} - i) * p^{k_{\text{Мантиссы}}}$$

$$k_{\text{Мантиссы}} = \frac{ArI}{ArR} * N(L_{\text{Мантиссы}})$$

$$S_{\text{Порядка}} = \sum_{i=0}^k N(L_{\text{Порядка}} - i) * p^{k_{\text{Порядка}}}$$

$$k_{\text{Порядка}} = \frac{ArI}{ArR} * N(L_{\text{Порядка}})$$

Формула 3. Редуктивные формулы для чисел с плавающей точкой

Основываясь на приведенных формулах, авторы перейдут к примерному математическому расчету, на котором рассмотрят математический расчет, основанный на сравнении данных моделей микропроцессора AMD64[9]. Расчет не является точным, но он дает первичное представление о возможных положительных эффектах для оптимизационных числовых алгоритмов операции понижения разрядности вычислений.

Для оценки выполним сравнение по нескольким показателям:

1. *Время выполнения одной операции.* Его можно получить из известной скорости выполнения нескольких операций в секунду.
2. *Оценочное значение операции чтения/записи данных в одну ячейку памяти регистра.* Получается, через деление первого показателя на размер регистра.

Время выполнения одной операции рассчитывается таким образом:

$T_{one} = T_N / N$, где:

1. T(one) — это время выполнения одной операции.
2. T(N) — время операций, которое документировано.
3. N — Число операций за документированную единицу времени.

Оценочное значение операции чтения/записи данных в одну ячейку памяти регистра рассчитывается таким образом:

$V_{итог} = V_{ed} * N_{рег}$, где:

1. V(ed) — время операции записи/чтения в ячейку памяти.
2. V(итог) — итоговое время операции записи/чтения в регистр.
3. N(рег) число ячеек памяти в регистре.

Таблица 1. Таблица характеристик процессоров

Показатель	Athlon 64	Pentium 4 Prescott	Athlon K7	Intel Pentium 4 Northwood
Частота, ГГц	2,2	2	2,2	1,6–3,4
Разрядность	64	64	32	32
Число транзисторов, млн	105,9	125	37,5	55

Name	Value
Diff_16	2.3333
Diff_32	1.4000

Рис 1. Результат вычисления по формуле результата оптимизации.

Теперь определим, как именно повысится скорость и точность вычислений. Под удовлетворительной точностью вычислений, будем понимать ситуацию, когда результаты компьютерной операции соответствуют требуемым результатам полностью или в пределах допустимой погрешности.

Для анализа воспользуемся данными статьи из источника [10], в которой были рассмотрены первые поколения процессоров AMD64. При выполнении расчета будет использоваться только тактовая частота процессора, так как модель Athlon 64 является одноядерной.

Для расчета был выбран Athlon 64, так как является самым первым из всех современных микропроцессоров с набором команд AMD64. Расчет был выполнен на примере анализа программы по вычислениям, состоящей из следующих блоков:

1. 300 команд 64 бита.
2. 300 команд 32бита.
3. 300 команд 16 бит.

Рассмотрим теперь 2 пониженных в разрядности аналога программ:

1. 600 команд 32бита.
2. 300 команд 16 бит.

Формула оценки времени выполнения программы по отношению к тактам:

$$T_{итог} = \sum Takt_{64} + \sum Takt_{32} + \sum Takt_{16},$$

где:

1. T(itog) — итоговое время выполнения тактов по программе, или итоговое выполнение программы.

2. Takt_32, Takt_16, Takt_64 — тактовое время операций с числами 32, 16 и 64 бит соответственно.

Для автоматизации выполнения оценки использовалась следующая программа на языке matlab:

%Тактовая частота:

Takt_CPU=2.2*10^9;%Взятая тактовая частота процессора:

```
T_one_Tact=1/Takt_CPU;
T_tact_reg_64=T_one_Tact*64;
T_tact_reg_32=T_one_Tact*32;
T_tact_reg_16=T_one_Tact*16;
```

%Определяем время реальной операции на старшей и младшей разрядности:

```
Program_64=300*T_tact_reg_64+T_tact_reg_32*300+T_tact_reg_16*300;
```

%Пониженная до 32 бит версия выглядит таким образом:

```
Program_32=T_tact_reg_32*600+T_tact_reg_16*300;
```

%Для 16 битного понижения итоговая программа будет выглядеть таким образом:

```
Program_16=T_tact_reg_16*900;
```

%Теперь определим эффект от понижения разрядности:

```
Diff_32=Program_64/Program_32;
Diff_16=Program_64/Program_16;
```

В Переменной Takt_CPU использовалось тактовая частота процессора Athlon 64. В результате вычислений получились результаты, изображенные на Рис 1.

В данном случае небольшой положительный эффект отличается следующими особенностями:

1. Равное разделение вычислений между типами данных. Однако в программах вычисления представлены в первую очередь максимальной разрядностью вычислений.
2. В программе считается, что размеры операндов команд не превышают размеры регистров.
3. Число тактов для всех операций одинаково, что не совсем корректно для современных программ.
4. Для операций с числами полагается одинаковое число тактов на каждой разрядности вычислений. Учитывая особенности работы процессора в эмуляции старых моделей — это не совсем корректно.
5. Не учтены коэффициенты операций, т.е. предполагается, что каждая операция выполняется за один временной такт не зависимо от разрядности.

Обсуждение

Описанные особенности позволяют совершенно точно и правильно доработать модели снижения разрядности вычислений при проектировании информационной системы. В частности, авторами были предложены несколько моделей технического решения [11] [12]. Более того, при помощи указанной арифметики можно добиться вспомогательного эффекта при внедрении в интеллектуальные системы [13], которые предполагается разместить на борту космических аппаратов. Первоочередное отличие заключается в том, что в предыдущих работах рассматривался вопрос об аппаратном понижении разрядности вычислений,

но наиболее серьезные последствия заключаются в том, что при повреждении кристалла идет сокращение программных возможностей микропроцессоров. При этом даже если микропроцессор сможет выполнять свои функции, но, тем не менее, результатом будет в первую очередь программные повреждения, что приводит к серьезнейшим ошибкам, в некоторых случаях падение системы задолго до реального уничтожения кристалла.

Более того, при использовании специальных эмулирующих библиотек для реализации функций можно добиться возникновения ещё одного вида погрешности. Например, во времена разработки 16 битных программ [14], в частности во времена i8086 и до i487, когда процессор и сопроцессор были спаяны в одно устройство, были доступны библиотеки, которые эмулируют поведение сопроцессора i8087, и позволяющие таким образом обходиться без сопроцессора вообще. В таком случае необходимо учитывать ещё и возможные ошибки работы используемой библиотеки.

Заключение

В рамках данной статьи были описаны математические особенности компьютерной математики, которые напрямую влияют на операцию понижения разрядности вычислений. Были рассмотрены виды сжатия числовых данных, которые можно применить для микропроцессора. Был предложен способ сжатия данных через метод преобразования целочисленных вычислений в вычисления с числами плавающей точкой.

При этом была проведена оценка потери разрядов при смене разрядности вычислений. Были рассмотрены типы погрешностей, которыми характеризуются числовые множества, попадающие в операции смены разрядности вычислений.

ЛИТЕРАТУРА

1. IEEE Standard for Binary Floating-Point Arithmetic. Copyright 2019 by The Institute of Electrical and Electronics Engineers, Inc 345 East 47th Street, New York, NY10017, USA.
2. Горшков В.В., Мокряков А.В., Приходько Д.И. Алгебраическая структура чисел с плавающей точкой. Перспективы науки. 2020. № 12 (135). С. 78–83.
3. Горшков В.В., Мокряков А.В., Приходько Д.И. Свойства компьютерных операций над числами с плавающей точкой / международная научная конференция: «стандартизация и техническое регулирование: современное состояние и перспективы развития. Информационно-экономические аспекты стандартизации и технического регулирования. 2020. № 6 (58). С. 349–355.
4. AMD64 Architecture Programmer's Manual, Volume 3. URL: <https://www.amd.com/system/files/TechDocs/24594.pdf> (дата обращения 25.08.2022)
5. Компьютеры. Athlon 64, Athlon 64 FX и Pentium 4. Часть первая. URL: <https://www.ferra.ru/review/computers/s25340.htm> (дата обращения 25.02.2021).
6. Горшков, В.В., Приходько, Д.И. & Мокряков, А.В. Устройство для управления конфигурацией вычислительной системы. (2021). Патент на полезную модель 207176 U1, 15.10.2021. Заявка № 2021118609 от 25.06.2021.
7. Горшков, В.В., Приходько, Д.И. & Мокряков, А.В. Устройство для управления разрядностью вычислений. (2022). Патент на полезную модель 209758 U1, 22.03.2022. Заявка № 2021118607 от 25.06.2021.

8. М.В. Дебиев, Р. А-М. Магомадов, Т.Ш. Амхаев, Ш.З. Зиниев. Эффекты внедрения автоматизированных электроэнергетических систем на основе интеллектуальных сетей. (2021). Вестник ГНТУ. Технические науки, том XVII, № 1 (23). С. 15–20.
9. Приходько, Д.И. Методы оптимизации размера программы на языке ++ / Д.И. Приходько, А.В. Мокряков, В.В. Горшков // Наука и образование: сохраняя прошлое, создаём будущее: сборник статей XXXI Международной научно-практической конференции: в 2 ч., Пенза, 10 декабря 2020 года. — Пенза: «Наука и Просвещение» (ИП Гуляев Г.Ю.), 2020. — С. 63–67. — EDN EKGUEO.

© Приходько Дмитрий Игоревич (mitry1205@mail.ru).

Журнал «Современная наука: актуальные проблемы теории и практики»



Российская академия наук