

### Введение

#### 1 Системы счисления и их применение

1. Деньги в конвертах и зерна на шахматной доске  
Пример работы алгоритма с числом  $2^n - 1$   
Двоичная запись числа  
Задача о шахматной доске
2. Взвешивание с помощью гири возведение в степень  
Возведение числа в произвольную степень на калькуляторе
3. Аддитивные цепочки и фляги с молоком  
Двоичный (бинарный) метод построения аддитивных цепочек  
Понятия ширины и глубины аддитивной цепочки
4. Краткая история двоичной системы  
Следы двоичной системы в истории  
Томас Харриот и открытие двоичной системы  
Чем удобна двоичная система?  
Сложение двух битов  
Операции дизъюнкции  $x \vee y$ , отрицания  $\neg x$ , конъюнкции  $x \bullet y$
5. Двоичная система и логические операции  
Элементы, реализующие логические операции  
Что такое булевы схемы? Как реализуется полусумматор этими схемами?  
Сложность, глубина и задержка логических схем
6. Схема для сложения двух чисел в двоичной системе
7. Ханойская башня, код Грея двоичный  $n$ -мерный куб  
Код Грея  
Код Грея и многомерный куб
8. "Книга Перемен"
9. Шрифты Брайля  
Брайль и "Книга Перемен"
10. Азбука Морзе и алфавитные коды  
Алфавитное кодирование
11. Алфавитное кодирование и шифрование
12. Фотопленка и штрих-код
13. Задачи о переливаниях  
Применение двоичной системы
14. Игра "Ним"  
Алгоритм распознавания выигрышной позиции
15. Д.И.Менделеев и троичная система  
Уравновешенная троичная система
16. Троичная система и фокус Жергонна
17. Немного об истории позиционных систем счисления  
Почему используется десятичная система?  
Экзотические системы счисления
18. Схема Горнера и перевод из одной позиционной системы в другую  
Алгоритмы перевода из одной системы в другую

19. Признаки делимости
20. Арифметические коды и коды Хемминга  
Арифметические коды позволяют найти ошибку  
Коды, исправляющие ошибки

## 2 Вычисления точные и приближенные

1. Устный счет  
Предварительные оценки  
Сложение и умножение  
Рецепты вычислений для чисел определенного вида  
Использование таблицы квадратов  
Деление  
Деление на числа, близкие к степеням десяти  
Контроль за правильностью вычислений  
Извлечение корней  
Квадратные корни  
Кубические корни  
Корни высоких степеней  
Квадратные корни -- метод "цифра за цифрой"
2. Применение логарифмов  
Изобретение логарифмов  
Как вычисляли логарифмы в Англии XVII в.  
Методы Бриггса  
Нужны ли в наше время логарифмы?  
Использование логарифмов в компьютерных вычислениях
3. Обычные и позиционные дроби  
Сведение к операциям над целыми числами  
Десятичные дроби
4. Алгоритм Евклида  
Цепные дроби  
Числа Фибоначчи  
Золотое сечение  
Формула Бине для чисел Фибоначчи  
Расширенный алгоритм Евклида
5. Вычисления при минимуме используемых средств  
Без деления  
Без умножения  
Без квадратных и кубических корней  
Без операции возведения в степень  
Без тригонометрии  
Без экспонент и логарифмов  
Без схемы Горнера
6. Еще о вычислениях на микрокалькуляторах  
Недокументированные возможности простейшего микрокалькулятора и вычисление чисел Фибоначчи  
Вычисления с длинными числами  
Система счисления по основанию  $10^9$

Как решать квадратные уравнения?

Квадратное уравнение общего вида

Как вычислять элементарные и неэлементарные функции на простейшем калькуляторе?

Арифметико-геометрическое среднее Гаусса

Как решать кубические уравнения?

Как выполнять вычисления с комплексными числами на калькуляторе?

7. Машинная арифметика и ошибки округления

Причины ошибок

Арифметика с фиксированной запятой

Арифметика с плавающей запятой

Нарушения законов арифметики и правил обращения с неравенствами

Ошибки округления

8. Вычисления двоичные и десятичные

Вычисления с очень большим числом значащих цифр

Вычисление  $2$  в корне

Вычисление  $e$

Вычисление  $\pi$

## Литература

### Введение

*Когда мне было 14 лет, мой отец был так глуп, что я едва выносил его. Когда же мне стукнуло 21, я поразился, увидев, как поумнел старик за эти 7 лет.*  
Марк Твен  
(цитируется по книге Ф.Харари "Теория графов")

Здесь уместно разъяснить, что означает термин "компьютерная арифметика" или, по крайней мере, как он понимается в книге. Что означает прилагательное "занимательная", разъясняться не будет.

Как и любая область науки, компьютерная арифметика состоит из нескольких связанных между собой направлений и по-разному выглядит с разных точек зрения. В ней есть и многочисленные технические детали, и фундаментальные понятия. Естественно, вторым будет уделяться больше внимания, чем первым, поэтому изложение не всегда будет исчерпывающим.

С одной стороны, компьютерная арифметика, как и следует ожидать, должна объяснять, как выполняются на компьютере различные операции с числами и/или машинными словами. Компьютеры имеют многочисленные команды для выполнения таких операций, среди которых есть и арифметические, и логические. Разные компьютеры (а к их числу без большой ошибки можно причислить и калькуляторы, да и почти любую цифровую технику) делают это по-разному и имеют различные наборы команд. Знание особенностей системы команд компьютера, конечно, нужно каждому программисту, хотя оно выходит за пределы собственно программирования -- чтобы писать программы, в общем-то не обязательно знать, каким образом компьютер выполняет команду сравнения чисел или их деления с остатком. Кроме того, полное понимание процессов, реально происходящих в компьютере во время выполнения программы, требует знания массы технических

деталей, связанных с устройством процессора и других микросхем, составляющих компьютер, и выходит за рамки собственно компьютерной арифметики, относясь к области, которую можно назвать архитектурой вычислительных систем.

Тем не менее компьютерные процессоры и другие большие цифровые микросхемы ("чипы") содержат в себе сравнительно небольшие микросхемы для выполнения отдельных арифметических или логических операций. Разобраться в их устройстве, по крайней мере в общих принципах функционирования, не так уж и сложно, потому что электронные микросхемы, как старые, так и самые современные, имеют очень простую математическую модель -- схему из функциональных логических элементов, часто называемую в инженерной практике комбинационной схемой, а в теории -- булевой логической схемой, или просто схемой (реализующей булевы функции, называемые также функциями алгебры логики). Построение булевых схем, выполняющих арифметические и логические операции, относится к сфере первоочередных интересов компьютерной арифметики и в то же время составляет существенный раздел теории сложности булевых функций. Впрочем, вопросы, связанные с поиском эффективных алгоритмов построения булевых схем и с автоматизацией проектирования реальных электронных микросхем, содержат множество технических деталей и выходят за рамки как компьютерной арифметики, так и теории сложности булевых функций, относясь к специальной области компьютерных наук -- автоматизации проектирования микросхем. Разумеется, касаться этой области в популярной книге неуместно.

В программировании иногда возникает необходимость выполнять арифметические действия с "длинными" числами, не уместными в разрядную сетку компьютера, например, в научных расчетах для достижения требуемой точности, в криптографии для выполнения действий с числами, состоящими из сотен цифр, в рекордных вычислениях, например, с целью найти миллиард знаков числа  $\pi$  и т.д. Для выполнения этих действий нельзя непосредственно использовать команды машинной арифметики, а необходимо писать программы для так называемой длинной арифметики. Такие программы давно написаны и входят в пакеты многих систем компьютерной алгебры. Чтобы самому написать подобные программы и разобраться в работе известных программ, нужна компьютерная арифметика. На самом деле используемые в этих программах арифметические алгоритмы весьма близки к алгоритмам, "зашитым в железе" электронных микросхем для выполнения арифметических операций. Но вместо двоичной системы счисления, которая используется в микросхемах (хотя используются и микросхемы, реализующие с помощью двоичных элементов десятичную арифметику), в компьютерных программах "длинной арифметики" обычно применяется система счисления с основанием  $2^w$ , где  $w$  -- длина машинного слова применяемого компьютера, а для записи самих программ вместо двоичной удобно использовать шестнадцатичную систему (так как это в четыре раза сокращает длину записи чисел).

Использование подходящей (как правило, того или иного варианта позиционной) системы счисления является простейшей, старейшей, и в то же время важнейшей и фундаментальной идеей, используемой в компьютерной арифметике. Поэтому знакомство с ней естественно начинать с рассмотрения различных систем счисления. Книга с этого и начинается, причем содержит почти целиком (и в несколько дополненном виде) материал из брошюры автора "Системы счисления и их применение". На самом деле упомянутая брошюра и задумывалась как первая часть популярной книги по компьютерной арифметике, но ее написание несколько затянулось, и первую часть пришлось опубликовать отдельно. Арифметические алгоритмы в нашей книге обычно излагаются для произвольной позиционной системы, но иногда они даются только для десятичной системы (чтобы подчеркнуть удобство их применения в ручных вычислениях

и даже в устном счете), а иногда -- только для двоичной (потому что в двоичной системе они имеют наиболее простой вид и наиболее широкую область применения).

Кроме чисто арифметических алгоритмов уместно показалось рассмотреть в книге некоторые вопросы вычислений с обыкновенными дробями, многочленами и дробно-рациональными функциями. Эти вопросы являются пограничными между компьютерной алгеброй и компьютерной арифметикой и весьма близки к последней. Их включение в книгу оправдано также тем, что, например, алгоритмы арифметических действий с многочленами на самом деле проще аналогичных алгоритмов для чисел.

Особенностью книги является также то, что помимо описания различных алгоритмов часто дается обоснованная оценка их сложности (т.е. число выполняемых ими операций). В случае логических схем оценка их сложности позволяет объективно судить о размерах, которые может иметь реальная электронная микросхема, моделируемая рассматриваемой логической схемой. В случае же программных алгоритмов оценка их сложности позволяет судить о времени их работы. Умение оценивать сложность алгоритма, таким образом, полезно и в программировании, и в проектировании (и производстве) электронной техники. Разделы, связанные с оценками сложности, находятся в другой книге и составляют, по-видимому, наиболее сложную ее часть (извините за каламбур). Это не удивительно, поскольку они находятся на границе между собственно компьютерной арифметикой и теорией сложности вычислений.

Чтобы компенсировать неизбежную сложность некоторых разделов книги, в нее показалось разумным вставить различные развлекательные темы, вроде приемов устных вычислений, интересных даже младшеклассникам, алгоритмов вычислений на калькуляторах, применений систем счисления за пределами компьютерной арифметики, исторических сведений о первооткрывателях тех или иных идей, затрагиваемых в книге и т.д. (однако точные библиографические ссылки на используемые результаты, как правило, не даются). Рассмотрены также вопросы эффективного выполнения некоторых манипуляций с числами и машинными словами на компьютере в условиях, когда отсутствуют команды, непосредственно их выполняющие. Подобные вопросы несомненно также относятся к компьютерной арифметике.

Книга, претендующая на популярность, не может освещать рассматриваемые вопросы достаточно полно. Поэтому некоторые вопросы, связанные с компьютерной арифметикой, затронуты лишь мимоходом или не затронуты вовсе. Это и вопросы оценки точности машинных вычислений, арифметика с плавающей запятой, логические схемы с памятью (автоматные схемы), алгоритмы вычисления элементарных функций, принципы работы компьютерного процессора, физические принципы работы электронных микросхем и их математического моделирования, вопросы проектирования, тестирования и производства чипов и т.д. Многим из подобных вопросов уже посвящаются целые книги (разумеется, не научно-популярные). Из великого множества алгоритмов компьютерной арифметики в книге упоминается лишь небольшая часть.

Предполагается, что читатель знает школьную математику (сейчас изучает ее или еще не успел забыть), поэтому иногда в книге используются без пояснений некоторые понятия и обозначения. Но во многих случаях необходимые пояснения даются. В отличие от учебников изложение не всегда систематичное, одним вопросам уделяется больше внимания, другим -- меньше, некоторые темы даже не упоминаются, в книге встречаются ссылки как назад, так и вперед и т.д. Все это неизбежные особенности вольного стиля изложения (вообще свойственного, как нам кажется, научно-популярной литературе). Но из-за почти полного отсутствия на русском языке литературы по компьютерной

арифметике автор надеется, что эту книгу можно будет использовать и с учебной целью как в школах, так и в вузах.

*В заключение автор благодарит своих бывших учеников (и нынешних коллег) А.А.Бурцева и И.С.Сергеева за помощь в работе.*

## Об авторе



**ГАШКОВ Сергей Борисович**

Доктор физико-математических наук, профессор кафедры дискретной математики МГУ имени М. В. Ломоносова. Автор и соавтор книг «Примени математику», «Арифметика. Алгоритмы. Сложность вычислений», «Системы счисления и их применения», «Современная элементарная алгебра», «Элементарное введение в эллиптическую криптографию» (М.: URSS), «Криптографические методы защиты информации», «Занимательная компьютерная арифметика: Математика и искусство счета на компьютерах и без них» (М.: URSS) и «Занимательная компьютерная арифметика: Быстрые алгоритмы операций с числами и многочленами» (М.: URSS).