КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

УДК 681.3

Р.В. ВАЛОВОЙ

КОМПЛЕКСНАЯ ИЕРАРХИЧЕСКАЯ МОДЕЛЬ РЕШЕНИЯ ЗАДАЧИ МОДУЛЯРНОГО ВОЗВЕДЕНИЯ В СТЕПЕНЬ

Рассматривается построение аппаратно-программных комплексов выполнения арифметических операций над многоразрядными числами. Предлагается иерархическая модель решения задачи модулярного возведения в степень, которая, анализируя входные данные, предлагает наиболее оптимальный по быстродействию алгоритм для данного набора входных данных.

Для решения ряда классов задач вычислительной, прикладной и дискретной математики необходима техника вычислений, которая использует алгоритмы выполнения различных арифметических операций над многоразрядными числами. Среди таких задач следует отметить задачи инерционной навигации, проектирования современных самолётов, задачи аппроксимации функций, некорректные задачи гидравлики, задачи моделирования физических, химических процессов, криптографии и т.д. При решении вышеприведенных задач широко используются вычисления с многократной точностью над целыми числами большой размерности. Такие вычисления проводят с помощью универсальных ЭВМ, специализированных аппаратных и аппаратно-программных комплексов, для которых создаются эффективные алгоритмы арифметики сверхбольших чисел соответственно для универсальных ЭВМ, специализированных аппаратных и аппаратно-программных комплексов. Подробно алгоритмы арифметики сверхбольших чисел для программной и аппаратной реализации проанализированы в [1]. Среди этих алгоритмов оптимизации подлежат алгоритмы умножения и возведения в степень с учетом их программной реализации на универсальных ЭВМ и аппаратной (аппаратно-прграммной) реализации на специализированных вычислительных устройствах. При аппаратной реализации, как было отмечено в [2], появляется известная свобода в выборе модели вычислений.

На протяжении нескольких десятилетий и до настоящего времени продолжаются исследования в области оптимизации вычислений модулярного возведения в степень для чисел большой разрядности. Следует отметить, что в данной области исследований были достигнуты большие успехи [3, 4]. Уменьшить время вычислений можно за счёт использования аппаратной реализации данной операции [1]. Поэтому исследованиям в области аппаратной реализации операции многоразрядного модулярного возведения в степень уделяется большое внимание.

В работе [5] предложено три основных направления решения проблемы уменьшения количества вычислений при реализации операции модулярного возведения в степень:

минимизация количества умножений;

минимизация числа шагов при выполнении каждого отдельного умножения;

минимизация количества вычислений при выполнении модулярной редукции.

Эти направления обычно рассматриваются как самостоятельные подходы для решения частных задач.

В работе [2] предложено при возведении в степень применять алгоритмы, адаптивные к входным данным. Для реализации этой идеи предлагается комплексная трёхуровневая модель решения задачи модулярного возведения в степень. Верхний уровень включает методы, позволяющие уменьшить количество операций модулярного умножения. Средний уровень содержит методы, позволяющие уменьшить временные и аппаратные затраты при реализации операций модулярного умножения. Нижний (вентильный) уровень объединяет в себе методы, реализующие на уровне элементарных операций алгоритмы, построенные на основе методов, входящих в верхние уровни. Для каждого из уровней на сегодняшний день предложено большое количество методов, позволяющих уменьшить вычислительную сложность операции модулярного возведения в степень. Однако все эти методы для оценки своего быстродействия и количества затрачиваемого оборудования используют максимальные и усреднённые значения. Это означает, что если выбранный метод является эффективным для одного показателя, то для другого – он может оказаться неэффективным. При этом диапазон оценок весьма широк.

В то же время, как показывают исследования, базой для оптимизации методов на двух верхних уровнях является анализ показателя степени или его основания. Если такой анализ выполнять оперативно, появляется возможность для регулирования временных и аппаратурных затрат, а также потребляемой мощности в процессе работы устройства. В данной работе рассматривается возможность оперативной адаптации вычислительного алгоритма к входным данным на основе комплексной трёхуровневой модели решения задачи модулярного возведения в степень.

Как было отмечено выше, верхний уровень занимают методы, позволяющие уменьшить количество операций модулярного умножения. Эти методы можно разделить на методы перекодировки и методы аддитивных цепочек. Установле-

но, что верхней границей количества возведений в квадрат, требуемых для возведения произвольного числа в степень e, является величина k-1, где k – число битов для представления показателя e. Применение данных методов позволяет уменьшить количество умножений, следующих за возведениями в квадрат. Методы перекодировки позволяют сделать это за счёт перекодировки показателя степени, в результате чего разрушаются блоки из единиц и получается разреженное представление показателя. Методы аддитивных цепочек генерируют короткие аддитивные цепочки для получения показателя степени. К методам аддитивных цепочек можно отнести: бинарный метод (показатель сканируется по одному биту), m-арный метод (сканирование показателя осуществляется по m бит), метод с плавающей длиной окон (показатель разбивается на «нулевые» и «ненулевые» блоки, с последующим поблочным сканированием показателя), фактор-метод (основан на факторизации показателя), метод дерева степеней (строится дерево степеней согласно некоторой эвристике) [1].

Известно, что методы перекодировки позволяют разрушить блоки из единиц и получить разреженное представление показателя, т.е. эти методы ориентированы на показатели с большими блоками единиц. При этом данные методы абсолютно неэффективны при показателях, состоящих из блоков повторяющихся комбинаций «01». В свою очередь метод с плавающей длиной окон ориентирован на показатели с большими блоками нулей. Эффективность таких методов как фактор-метод, метод дерева степеней и другие целиком зависят от конкретных значений показателей степени. Поэтому логичным способом повышения эффективности вычислительного устройства, построенного на основе трёхуровневой модели, является использование для каждого значения входных данных (например, для каждого операнда) соответствующего метода, оптимального (или близкого к нему) именно для этих значений входных данных. Выявив параметры входных данных, влияющие на эффективность, следует в дальнейшем оперативно оценивать входные данные, выбирать метод или комбинацию нескольких методов, доступных на данном уровне для конкретного значения показателя степени.

Проанализировав показатель степени, выбрав наиболее подходящий для данного показателя метод, можно опуститься на уровень ниже, где появляется возможность уменьшения времени и аппаратных затрат при реализации операций модулярного умножения. Хотя методы, доступные на среднем уровне, не оказывают на общее время, затрачиваемое вычислительным устройством, такого влияния как методы верхнего уровня, тем не менее они позволяют существенно уменьшить время выполнения модулярного умножения, что в общем итоге также позволит уменьшить общее время, затрачиваемое на выполнение операции модулярного возведения в степень.

Методы, представленные на этом уровне, можно условно разделить на методы, включающие в себя алгоритмы модулярного умножения (алгоритмы Blakey, Montgomery), и методы, оптимизирующие эти алгоритмы. Для увеличения скорости операции умножения были разработаны различные методы умно-

жения [6], которые работают в фиксированных конечных полях. Однако метод, разработанный для конечного поля n-разрядного числа, не может быть использован для поля k-разрядного ($n < k$). Поэтому получили распространение методы, реализующие масштабируемую (scalable) архитектуру. Арифметический модуль называется масштабируемым, если этот модуль может быть использован для генерации результата высокой точности, независимо от того на данные какой точности он был разработан. Вполне очевидно, что эти методы для реализации масштабируемости вносят дополнительные временные и аппаратные затраты. Более эффективно было бы использовать ту аппаратную реализацию алгоритма модулярного умножения, которая спроектирована для данной характеристики поля. Для этого необходимо проанализировать разрядность основания поля и выбрать соответствующую аппаратную реализацию. На данном уровне существует большое количество методов, позволяющих оптимизировать операцию модулярного умножения за счёт детального анализа основания степени и основания поля. Детальный анализ основания поля при предложенном подходе позволит выбрать наиболее оптимальный алгоритм модулярного умножения для конкретного набора входных данных.

Рассмотрим вентильный уровень. Этот уровень объединяет в себе методы, позволяющие оптимизировать аппаратную реализацию алгоритмов, выбранных на двух первых уровнях. От эффективности аппаратной реализации этих алгоритмов зависит эффективность вычислительного устройства в целом. Если неэффективно, с точки зрения аппаратной реализации, реализованы алгоритмы, выбранные на двух верхних уровнях, то теряется весь выигрыш оптимизации, достигнутый на предыдущих уровнях. Однако оптимизация аппаратной реализации алгоритмов – это инженерная задача. Поэтому основное внимание в статье уделяется верхним двум уровням.

Рассмотрев все уровни, можно представить работу вычислительного устройства, построенного на трёхуровневой модели, адаптированной ко входным данным, следующим образом:

1) на верхнем уровне проводится анализ показателя степени и на основе зависимостей эффективности методов оптимизации от входных данных выбирается метод возведения в степень;

2) на среднем уровне проводится анализ характеристики поля и на основе зависимостей эффективности методов оптимизации от входных данных выбирается метод модулярного умножения;

3) для выбранных методов возведения в степень и модулярного умножения осуществляется выбор метода аппаратной реализации алгоритмов;

4) устройство динамически настраивается на работу в соответствии с выбранными алгоритмами.

Таким образом, для реализации операции модулярного возведения в степень вышеописанного вычислительного устройства затрачивается общее время

$$t_{общ} = t_{ан} + t_{кфг.} + t_{выч}, \quad (1)$$

где $t_{ан}$ – время, затрачиваемое на анализ входных данных; $t_{кфг}$ – время на реконфигурирование вычислительного устройства; $t_{выч}$ – время вычисления выбранного набора алгоритмов для модулярного возведения в степень.

Время выполнения той же операции вычислительной системой, построенной на универсальном алгоритме, будет состоять только из времени реализации универсального алгоритма возведения в степень.

Очевидно, что время работы алгоритма, оптимального для конкретных входных значений, будет меньше или равно времени реализации универсального алгоритма. Однако время анализа и особенно время реконфигурирования вычислительной системы будут вносить дополнительные временные задержки в общее время выполнения операции модулярного возведения в степень. Наиболее эффективным такой подход будет при построении вычислительного устройства, в котором анализ входных данных проводится очень редко по сравнению с общим временем работы данного вычислительного устройства. Например, для вычислительного устройства, являющегося частью криптографической системы, которая в свою очередь использует ассиметричную криптографию. Это обусловлено следующей особенностью криптографической системы, использующей ассиметричную криптографию: после установления «защищенного» канала связи между двумя абонентами (отправитель и получатель определяют ключи для зашифрования-расшифрования), и далее при шифровании сообщений показатель степени, влияющий на выбор метода модулярного умножения, и модуль, определяющий характеристику конечного поля множительного устройства, в дальнейшем не меняются. Это означает, что общее время работы вычислительной системы $t_{общ}$ (1) будет включать в себя три составляющие только в момент установления «защищённого» канала связи. В дальнейшем это время будет состоять только из одной составляющей – времени реализации выбранного алгоритма модулярного возведения в степень $t_{выч}$.

Построение вычислительного устройства на основе трёхуровневой модели для решения задачи модулярного возведения в степень, адаптированной к конкретным значениям входных данных, позволит уменьшить вычислительные затраты по сравнению с вычислительной системой, построенной на универсальных алгоритмах. Время вычисления операции модулярного возведения в степень будет зависеть в каждом конкретном случае от входных данных. За счёт выбора методов оптимизации на всех уровнях на основе анализа входных данных это время может быть уменьшено по сравнению со временем, затрачиваемым на ту же операцию с теми же входными данными вычислительной системой, построенной на универсальных методах. Например, m-арный метод по сравнению с бинарным методом возведения в степень, позволяет уменьшить количество умножений (в зависимости от разрядности показателя степени) на 21%. В свою очередь, методы ненулевых окон позволяют уменьшить количество умножений, по сравнению с m-арным методом, приблизительно на 7% [6]. Даже, если в ка-

честве универсального метода использован метод ненулевых окон, то для некоторых показателей степени количество умножений может быть уменьшено ещё более существенно [2].

При программной реализации какого-либо алгоритма разработчик имеет в своём распоряжении набор арифметических функций, реализованных на аппаратном уровне выбранным им процессором. При этом разработчик может только менять порядок вызова этих функций и не имеет возможности менять алгоритм реализации арифметических функций процессора. Такой возможностью обладает разработчик, разрабатывающий программно-аппаратное устройство на базе микросхем программируемой логики. На сегодняшний день технология изготовления программируемых логических интегральных схем (ПЛИС) предоставляет разработчику уникальную возможность динамически менять алгоритм реализации функции, исполняемой аппаратно-программным устройством. Такие микросхемы получили широкое распространение при аппаратной реализации различных сложных вычислительных устройств за счёт невысокой стоимости при большом количестве программируемых вентилей. Микросхемы программируемой логики позволят решить задачу динамического реконфигурирования вычислительной системы, построенной на основе трёхуровневой модели для решения задачи модулярного возведения в степень.

Дальнейшие исследования в данной области предполагается направить на выявление зависимостей эффективности методов от входных данных на каждом из уровней трёхуровневой модели решения задачи модулярного возведения в степень.

1. *Задірака В.К., Олексюк О.С.* Комп'ютерна арифметика багаторозрядних чисел: Наукове видання. – К.: 2003.– 264 с.
2. *Задірака В.К., Кудін А.М.* Построение программно-аппаратных комплексов арифметики сверхбольших чисел // Комп'ютерна математика. Оптимізація обчислень. – К.: Ін-т кібернетики ім. В.М. Глушкова НАН України , 2001. – **1**. – С. 452–460.
3. *Alan Daly, William Marnane*. Efficient Architectures for implementing Montgomery Modular Multiplication and RSA Modular Exponention on Reconfigurable Logic // Dept. of Electrical and Electronic Engineering, Technical report, 2002.
4. *Pascal Paillier*. Low-cost double-size modular exponention // ENST. Computer science departament, Lecture Notes in computer science. – 1999. – P. 223–234.
5. *Богданов А.М., Зинченко Я.В*. Умножение сверхбольших чисел и быстрое преобразование Хаара // Наук.-техн. журнал „Захист інформації". – 2002. – № 4. – С. 58–67.
6. *Кос С.К*. High-speed RSA implementation. – RSA Laboratories, Technical report TR 201, 1994.

Получено 24.03.2005