ИСПОЛЬЗОВАНИЕ МОДУЛЯРНОЙ АРИФМЕТИКИ ДЛЯ СИНТЕЗА СИСТЕМ ОБРАБОТКИ МНОГОРАЗРЯДНЫХ ЧИСЕЛ НА FPGA

М.А. Дерябин, А.А. Зайцев

Северо-Кавказский федеральный университет, Ставрополь

Рассмотрены возможности приложения модулярной арифметики при синтезе интегральных схем на ПЛИС архитектуры FPGA для работы с числами большой разрядности. Проведен анализ архитектурных особенностей FPGA, и выработаны требования к алгоритмам, реализующим операции с целыми многоразрядными числами. Приведен пример синтеза устройства на основе системы остаточных классов.

Введение

Программируемые пользователем вентильные матрицы (FPGA) являются перепрограммируемыми кремниевыми чипами. Использование FPGA-чипов во всех отраслях промышленности обусловлено тем, что данные ПЛИС сочетают в себе лучшие части специализированных интегральных схем (ASIC) и систем, основанных на базе традиционных процессоров [1]. За счет архитектурных особенностей и внутреннего параллелизма FPGA способны обеспечивать аппаратно-тактируемую скорость и надежность, не требуя при этом таких же объемов, как стандартный микропроцессор, и дополнительных накладных расходов на индивидуальный дизайн ASIC.

Однако при реализации операций с длинными числами, превышающими разрядную сетку конкретной схемы, возникают определенные сложности, связанные с ограниченностью возможностей функциональных блоков FPGA. Данная особенность элементарных вычислительных узлов ПЛИС дает возможность выполнять простые операции с высокой скоростью в реальных ситуациях, однако понижает возможности при реализации сложных алгоритмов. При этом отдельный отпечаток накладывает абсолютный параллелизм FPGA. В соответствии с этим реализация многоразрядной арифметики на ПЛИС данной архитектуры требует индивидуального подхода и анализа. Один из эффективных методов решения данной проблемы открывает модулярная арифметика и ее приложения.

1. Использование модулярной арифметики при обработке многоразрядных чисел

Система остаточных классов (Residue number system, RNS, СОК) является непозиционной системой представления чисел [2]. Числа в ней определяются кортежем разрядов, являющихся остатками от деления данного числа на модули системы – набор взаимно простых чисел, определяющих СОК. Согласно китайской теореме об остатках (КТО), такое представление является единственным, если исходное число не превышает заранее известного диапазона СОК – произведения оснований. При этом значение любого разряда не влияет на значения остальных, что дает данной системе определенные преимущества перед более традиционными позиционными системами счисления.

Учитывая отсутствие межразрядных связей, модулярная арифметика позволяет проводить декомпозицию системы большого динамического диапазона на ряд параллельных независимых каналов меньшей разрядности. Использование такого подхода

увеличивает эффективность вычислений, в частности при реализации операций сложения, умножения, возведения в степень.

Особенностью адаптации алгоритмов для СОК является необходимость перевода данных из позиционной системы в СОК и обратно. В [3] предлагается эффективный метод преобразования двоичного числа в СОК на основе разбиения исходного двоичного числа на отдельные форматы, для которых отводится заранее известное количество двоичных разрядов.

Обратное преобразование числа из модулярного представления в двоичную форму базируется на классической теореме из теории чисел, которая называется китайской теоремой об остатках. Предлагается метод восстановления чисел на основе совместного использования КТО и обобщенной позиционной системы счисления (ОПСС) [3], в которой числа заменяются кортежем цифр небольшой разрядности.

Использование данных методов перевода позволяет обеспечить возможность обработки данных без использования на вычислительном устройстве операций с многоразрядными числами. Благодаря чему в алгоритмах достигается параллелизм и высокая скорость работы основных арифметических операций, что продемонстрировано в работе [4]. Согласно данным исследованиям, ускорение, получаемое при использовании СОК для решения задачи модульного возведения в степень на персональном компьютере, относительно последовательного аналога данного алгоритма, превосходит количество параллельных потоков. При соблюдении определенных условий СОК позволяет достичь высокой скорости работы алгоритма при использовании даже небольшого числа параллельных потоков выполнения. Данный эффект достигается за счет снижения разрядности обрабатываемых операндов и перехода от многоразрядной арифметики к стандартной, поддерживаемой вычислительной системой на архитектурном уровне.

2. Синтез схем многоразрядной арифметики на FPGA

Основываясь на данных исследованиях, можно перенести имеющиеся алгоритмы на FPGA. Однако при этом на алгоритмы накладываются определенные ограничения, связанные с архитектурными особенностями вычислительной системы. Во-первых, FPGA реализует реальный параллелизм (true parallelism), что следует учитывать при реализации алгоритмов. При этом система остаточных классов за счет параллелизма на уровне операций позволяет максимально эффективно использовать ресурсы ПЛИС. Во-вторых, накладывается существенное ограничение на размерность элементарных операндов математических преобразований. Чаще всего разрядность FPGA ограничена 16, 32 или 64 битами на число. В связи с этим необходимо особым образом подходить к выбору оснований СОК. Важно учитывать, что элементарные операции не должны приводить к переполнению простого типа, иначе неизбежны потери данных. В-третьих, использование совместно с ПЛИС стандартного микропроцессора реального времени позволяет вынести ряд незначительных операций и тем самым снизить загруженность конечной схемы. В-четвертых, для реализации действий с константными данными особо эффективным методом при использовании ПЛИС являются LUT (look-up tables), предоставляющие способ для быстрого однотактового доступа к данным, хранящимся в табличном виде.

На основе описанного подхода к представлению чисел была синтезирована система, реализующая базовые арифметические операции. Данная система базируется на FPGA фирмы XILINX версии Spartan 6, являющейся 32-битной вычислительной системой. В связи с чем выбор модулей был обусловлен ограничениями системы – были подобраны девять 16-битных чисел. Общая разрядность обрабатываемых чисел, таким образом, не превышает 144 бит.

На рис. 1 представлена функциональная схема блока перевода из позиционного представления числа в СОК. Перевод осуществляется путем дробления на 16-битные форматы, каждый из которых масштабируется и накапливается в общую сумму. Преобразование производится параллельно одновременно по всем модулям.

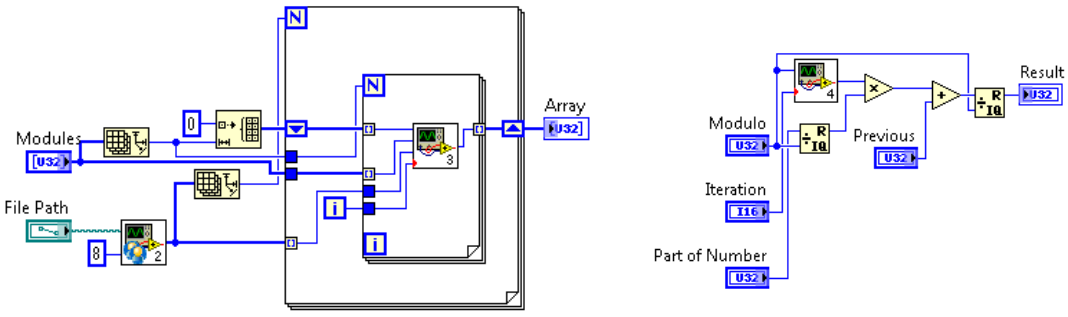

Рис. 1. Чтение и перевод шестнадцатеричного числа в СОК
в среде программирования LabVIEW

На рис. 2 представлен блок, в рамках которого выполняется умножение вектора числа в СОК на матрицу базисов, производится учет межразрядных переносов, в результате на выходе получается представление числа в ОПСС.

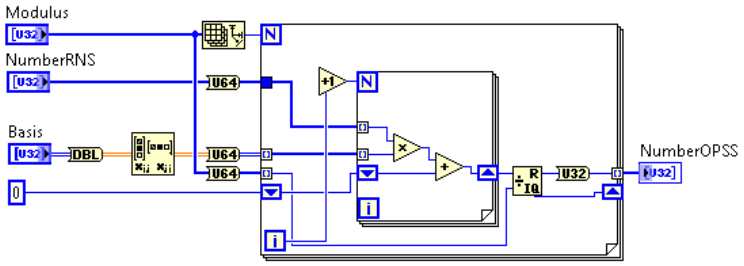

Рис. 2. Функциональная схема перевода числа из СОК в ОПСС

На рис. 3 представлен блок модульного возведения в степень, реализующий алгоритм быстрого модульного возведения, представленный в работе [5].

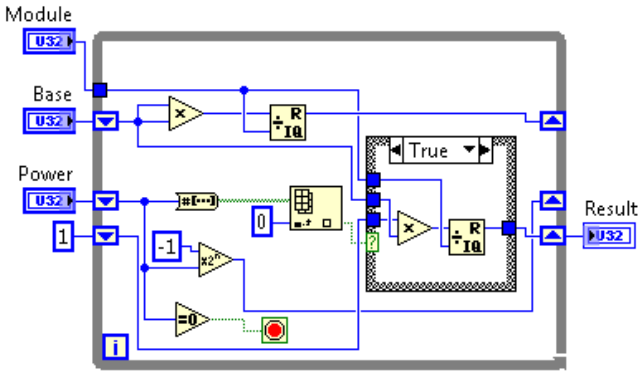

Рис. 3. Модульное возведение в степень

Разработанный модуль способен производить операции над многоразрядными числами в реальном времени, используя основные преимущества FPGA. Основной результат – снижение загруженности самой схемы при реализации арифметических операций над длинными числами.

Вывод

Работа с многоразрядными числами в FPGA затруднена аппаратными и архитектурными ограничениями самой платформы. В этом случае наиболее эффективное использование всех возможностей ПЛИС данного вида достигается с использованием параллельных методов обработки данных на уровне арифметических операций. Показано, что использование СОК является параллельной альтернативой классическим методам работы с большими числами. Система остаточных классов делает возможным реализация многоразрядной арифметики на ПЛИС с жестко ограниченной разрядной сеткой.

Литература

1. Баран Е.Д. LabVIEW FPGA. Реконфигурируемые измерительные и управляющие системы. – М.: ДМК Пресс, 2009. – 448 с.
2. Omondi A., Premkumar B. Residue number systems. Theory and Implementation. – London: Imperial College Press, 2007. – 258 p.
3. Червяков Н.И. Реализация высокоэффективной модулярной цифровой обработки сигналов на основе программируемых логических интегральных схем. // Нейрокомпьютеры: разработка и применение. №10. 2006. С. 24–36.
4. Дерябин М.А., Зайцев А.А. Использование модулярной арифметики для ускоре-ния выполнения операций с числами большой разрядности // Материалы XII Всероссийской конференции «Высокопроизводительные вычисления на кластерных системах». Нижний Новгород: Издательство Нижегородского университета, 2012. С. 129-133.
5. Schneier Bruce. Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition. – New York: John Wiley & Sons, 1995. – 662 p.