РЕАЛИЗАЦИЯ ОПЕРАЦИЙ УМНОЖЕНИЯ В ПОЛЯХ ГАЛУА НА БАЗЕ ПЛИС/FPGA

Листопад Е. В.
Кафедра электронных вычислительных средств, Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: listopadev@bsuir.by

В работе рассмотрены различные варианты аппаратных реализаций операций умножения в полях Галуа для эффективного решения задач построения цифровых устройств. За основу берется программируемая логическая интегральная схема (ПЛИС) с архитектурой FPGA типа XC6SLX75 (Spartan 6) фирмы Xilinx. Предложены подходы к аппаратной реализации операций умножения в полях Галуа, как наиболее требовательных к аппаратным ресурсам FPGA и ограничивающих быстродействие цифровых устройств. Продемонстрированы различные варианты аппаратных реализаций, выполненных с применением предложенных подходов.

Введение

Для решения задачи прототипирования цифровых устройств, решающих задачи теории кодирования, компьютерной алгебры, криптографии, цифровой обработки сигналов зачастую возникает необходимость эффективной реализации арифметических операций над элементами поля Галуа. Особый интерес представляет операция умножения в поле, как наиболее требовательная к аппаратным ресурсам FPGA, и лежащая в основе более сложных операций в поле. Достоинством любых вычислений в поле Галуа является то, что они допускают параллельную реализацию [1]. Это позволяет рассматривать их как адекватные архитектуре ПЛИС типа FPGA.

I. Определение элементов поля Галуа

Поля Галуа описываются двумя основными параметрами: m и p [2]. Параметр m указывает число двоичных разрядов, использующихся для двоичного представления символа множества, а также определяет количество элементов множества как 2^m. Таким образом, в поле $GF(2^4)$, где $m = 4$, содержится всего 16 элементов, и для двоичного представления каждого из них необходимо четыре двоичных разряда. Параметр p (генерирующий полином) указывает порядок, в котором элементы поля Галуа следуют друг за другом. Например, генерирующий полином $p(x)$ для поля $GF(2^4)$ может быть следующим: $p(x) = 1 + x^3 + x^4$. Часто используется представление полинома в виде двоичного числа с разрядностью $m + 1$. В данном случае $p = 25$ в десятичной системе, или 11001 в двоичной, или $1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0$. Если корень полинома обозначить через a, то $a^4 = a^3 + 1$. Элементы поля $GF(2^4)$ можно привести в трех представлениях:

1. степенное представление, в котором нулевой элемент равен 0, первый равен 1, второй равен a и т.д.;
2. полиномиальное представление (в виде многочлена): $x = k_0 * 1 + k_1 * a + k_2 * a^2 + k_3 * a^3$, где $k_0, k_1, k_2, k_3 = (0, 1)$;
3. бинарное представление или двоичная форма.

При реализации аппаратных умножителей часто применяется полиномиальное представление элементов поля [3,4].

II. Реализация аппаратного умножения за 16 шагов

Рассмотрим поле с параметрами $m = 16$ и $p = 126977$, в котором опишем особенности аппаратной реализации операций умножения. Как видно из параметров поля, операнды для произведения являются 16-битными. Первый подход предусматривает умножение за 16 шагов. При этом на каждом шаге выполняется умножение на 1 бит операнда и осуществляется приведение по модулю полинома. На рис. 1 приведена универсальная структура IP-ядра.

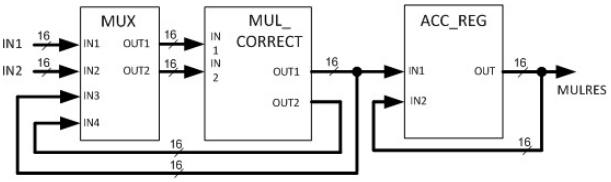

Рис. 1 – Универсальная структура IP-ядра, выполняющего умножение за 16 шагов

Были разработаны 3 экспериментальные реализации IP-ядер для данного подхода. Реализация 1 выполняет умножение за 16 тактов, при этом за 1 такт выполняется 1 шаг умножения с приведением. Реализация 2 выполняет умножение за 8 тактов, при этом за 1 такт выполняется 2 шага умножения с приведением и анализируется 2 бита операнда. Реализация 3 выполняет умножение за 4 такта, при этом за 1 такт выполняется 4 шага умножения с приведением и анализируется 4 бита операнда. Выполнена оценка произ-

водительности разработанных IP-ядер и количества требуемых ресурсов кристалла FPGA.

III. Реализация аппаратного умножения за 2 шага

Если главной особенностью первого подхода являлась реализация операции приведения по модулю полинома после каждой операции умножения на очередной бит, то во втором подходе операция приведения по модулю полинома выполняется после всех операций умножения на бит (то есть после 16-и операций). Таким образом, второй подход предусматривает умножение за 2 шага (см. рис. 2): непосредственно умножение 16-битных операндов с получением 32-битного промежуточного результата и приведение его по модулю полинома к 16-битному результату.

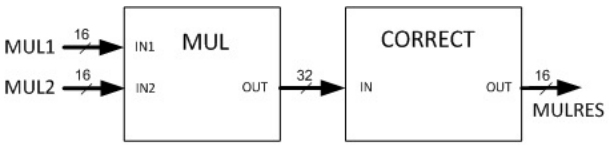

Рис. 2 – Универсальная структура IP-ядра, выполняющего умножение за 2 шага

Реализация 4 выполняет полное умножение за 1 такт, при этом под полным умножением понимается умножение арифметическое с приведением по модулю полинома. Реализация 5 выполняет полное умножение за 2 такта. При этом на первом такте выполняется арифметическое 32-разрядное умножение, на втором – приведение по модулю полинома. В реализации 6 была произведена попытка разделить асинхронную вычислительную часть, состоящую из блоков умножения и приведения по модулю полинома, дополнительными регистрами. Усматривается возможность оптимизировать структуру блоков умножения таким образом, чтобы разрядность логических функций не превышала 6 и соответствовала структуре слайсов (см. рис. 3), имеющихся на базовом кристалле FPGA. Такая возможность заключается в установке дополнительных регистров внутри блоков умножения после первого уровня логики в каждом из блоков логических функций.

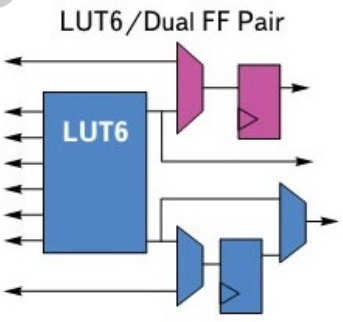

Рис. 3 – Структура слайса Spartan6

IV. Результаты исследований

В ходе исследований были выполнены аппаратные реализации операций умножения в поле Галуа с параметрами $m = 16$ и $p = 126977$ с применением одного из двух подходов. Первый подход предусматривает реализацию операции приведения по модулю полинома после каждой операции умножения на очередной бит. С применением данного подхода были выполнены реализации 1-3, характеристики которых приведены в таблице 1.

Таблица 1 – Характеристики реализаций с умножением за 16 шагов

№	Кол-во тактов	Ресурсы FPGA, Слайсы	Производительность, Мбит/с	Частота, МГц
1.	16	17	371.9	390
2.	8	24	555.0	291
3.	4	55	778.2	204

Второй подход предусматривает реализацию операции приведения по модулю полинома единожды после 32-разрядного арифметического умножения. С применением данного подхода были также выполнены реализации 4-6, характеристики которых приведены в таблице 2.

Таблица 2 – Характеристики реализаций с умножением за 2 шага

№	Кол-во тактов	Ресурсы FPGA, Слайсы	Производительность, Мбит/с	Частота, МГц
4.	1	50	2807.6	184
5.	2	46	1686.1	221
6.	2	48	2128.6	279

Выбор оптимальной реализации в качестве IP-ядра в полной мере зависит от аппаратных требований и ограничений той системы, в которую такое IP-ядро необходимо встраивать.

1. Шалагин, С.В. Реализация цифровых устройств в архитектуре ПЛИС/FPGA при использовании распределенных вычислений в полях Галуа: монография / С. В. Шалагин – Казань: Изд-во КНИТУ-КАИ, 2016. – 228 с.
2. Поляков, А. Библиотека VERILOG описаний арифметических операций в поле Галуа / Поляков Аркадий, Мехди Тайлеб, Незхат Тайлеб //Современная электроника. 2007. № 5.
3. Reyhani, M. A. Low Complexity Bit Parallel Architectures for Polynomial Basis Multiplication over GF(2m) / Reyhani Massolem A., Hasan M.A. // IEEE Transaction on Computers. 2004. V. 63. № 8.
4. José, L. I. Low Latency GF(2m) Polynomial Basis Multiplier / José Luis Imaña // IEEE Transaction on Circuits and Systems. 2011. V. 58. № 5.