

ВЫЧИСЛЕНИЕ ПОКАЗАТЕЛЬНОЙ ФУНКЦИИ НА ПЛИС

Попов С. Д., Опадчий Ю. Ф.

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «МАТИ – Российский государственный технологический университет имени К. Э. Циолковского», Россия (г. Москва, ул. Оршанская, д. 3), e-mail: electron_inform@mail.ru.

В данной работе рассматривается оптимизация табличного метода для вычисления значения показательной функции. Главный недостаток табличного метода заключается в том, что на практике объем информации, который необходимо хранить в таблице, оказывается чрезмерно большим, что влечет высокие аппаратные затраты. Предложен способ разбиения двоичного представления аргумента, позволяющий свести задачу к чтением из нескольких маленьких таблиц, содержащих по 4 строки. Предложенный метод подробно рассмотрен на примере вычисления значения экспоненциальной функции. На примере ПЛИС серии Stratix III фирмы Альтера проведено сравнение предложенного метода и алгоритма, встроенного в САПР Quartus II фирмы Альтера. При ограниченном диапазоне изменения аргумента рассмотренный алгоритм обладает явным преимуществом как по затратам аппаратных ресурсов ПЛИС, так и по быстрдействию.

Ключевые слова: ПЛИС, алгоритм вычисления, показательная функция, табличный метод.

COMPUTATION OF THE EXPONENTIAL FUNCTION ON FPGA

Popov S. D., Opadchiy Y. F.

«МАТИ» – Russian State University of Aviation Technology n.a. K.E. Tsiolkovsky, Moscow, Russia (Orshanskaya ulitsa, 3), e-mail: electron_inform@mail.ru.

In this article, we have considered the optimization of the tabular method for the computation of the values of the exponential function. The main drawback of tabular method lies in the fact that, in practice, the amount of information that must be stored in the table is too large, which implies high hardware costs. The proposed method consists in splitting of the binary representation of the argument, which allows to reduce the problem to the readings of several small tables, that contain 4 lines. The proposed method considered in detail by the example of calculation of the value of the exponential function. We have carried out a comparison of the proposed method and algorithm, built-in in the CAD Altera Quartus II, on the FPGA Altera Stratix III. With the limited range of changes of the argument, the proposed algorithm has a clear advantage, as the cost of hardware resources of the FPGA, and so on speed.

Keywords: FPGA, algorithm of calculation, exponential function, tabular method.

При разработке различных специализированных вычислителей на ПЛИС часто встает вопрос нахождения значения некоторой показательной функции вида $a^{q(x)}$. К таким функциям относятся $\exp(x)$, 2^x , 10^x и им подобные. Обычно для получения численного значения этих функций используют известные разложения в степенные ряды. Однако такой подход не является универсальным, так как требует создания для вычисления каждой из них собственного алгоритма, адаптированного для ПЛИС. Здесь, в зависимости от конкретных технических требований к разработке, могут применяться, либо широко известная схема Горнера [2], либо более универсальные схемы с предварительной обработкой коэффициентов [4], либо алгоритмы, построенные на основе графа, представленного в параллельной канонической форме, либо табличный метод.

Последний из перечисленных табличный метод является наиболее универсальным, однако при повышении точности вычисления он требует значительных аппаратных ресурсов,

используемых для хранения исходных таблиц. Так при абсолютной погрешности вычисления функции $\exp(x)$ в диапазоне $0 \leq x < \ln(2)$ порядка $2.3E-10$, что соответствует 32 точным разрядам результата, требуется хранить не менее 2^{32} табличных значений функции. Этот недостаток может быть частично компенсирован за счет использования дополнительной обработки табличных данных, например, методом линейной интерполяции [3]. Однако и в этом случае сохраняется необходимость в хранении значительных объемов информации.

Используя одно из основных свойств показательной функции, а именно – соотношение:

$$a^{\left(\sum_i x_i\right)} = \prod_i a^{x_i}, \quad (1)$$

можно предложить универсальный алгоритм вычисления, обеспечивающий высокое быстродействие при малых аппаратных затратах на реализацию.

Рассмотрим данный подход на примере вычисления экспоненциальной функции. Предположим, что диапазон изменения аргумента равен $0 \leq x < \ln(2)$ и его разрядность составляет 16 бит. Требуемое число верных разрядов результата – 32.

Представим аргумент в виде:

$$x_a = a_1 a_2 \cdot 2^{-2} + a_3 a_4 \cdot 2^{-4} + a_5 a_6 \cdot 2^{-6} + \dots = \sum_i a_{2i-1} a_{2i} \cdot 2^{-2i},$$

где $a_i \in \{0,1\}$; a_1, a_2 и т.д. – соответствующие разряды x_a ; $a_{2i-1} a_{2i}$ – целые числа, составленные из соответствующих разрядов x_a .

Тогда, используя (1), для рассматриваемой функции можно записать:

$$\exp(x) = \prod_i a_{2i-1} \cdot a_{2i} \cdot 2^{-2i}.$$

Таким образом, процесс вычисления функции разбивается на несколько параллельных процессов. В нашем случае процессов 8, причем в каждом из них, так как используется два разряда аргумента, вычисляются только 4 значения функции, одно из которых нулевое. Применим к каждому процессу табличный метод. В результате для вычисления функции необходимо хранить только 32 табличных значения.

На рис. 1 приведен соответствующий данному случаю граф алгоритма вычисления.

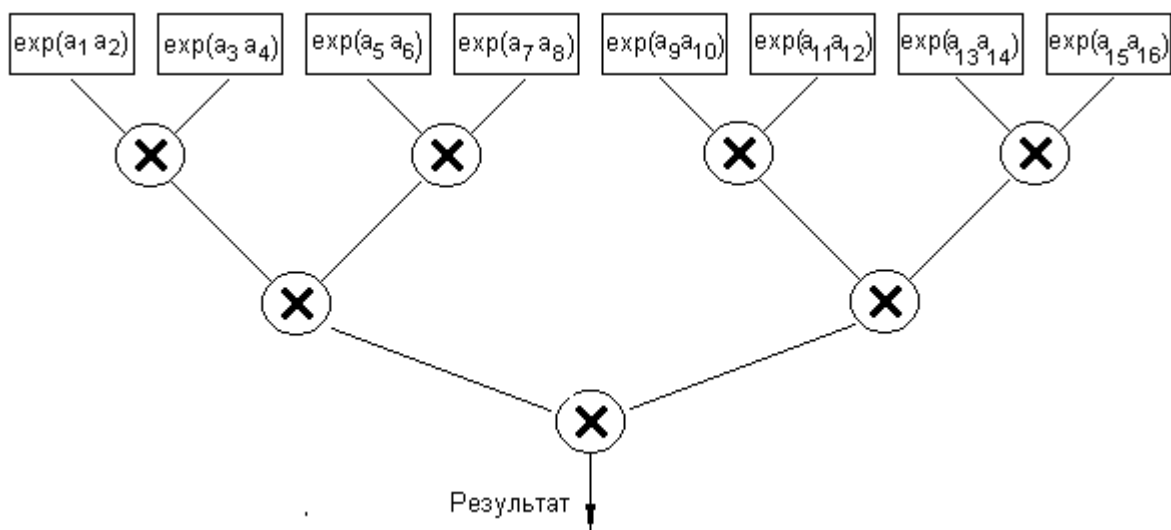


Рис. 1. Алгоритм вычисления $\exp(x)$ при разрядности аргумента – 16 бит

Реализация данного алгоритма требует 3-х ярусов умножения, то есть суммарное время вычисления будет равно:

$$t_{\Sigma} = t_{RD} + t_{MULT} \cdot n_{MULT}, \quad (2)$$

где t_{RD} – время чтения табличного значения функции, t_{MULT} – время выполнения операции умножения, n_{MULT} – количество последовательно выполняемых операций умножения.

Анализируя граф на рис. 1, нетрудно получить общую формулу для вычисления значения n_{MULT} :

$$n_{MULT} = Ceil(\log_2(n_a) - 1), \quad (3)$$

где n_a – число разрядов аргумента x_a .

Для определения разрядности табличных значений функции воспользуемся известным правилом, согласно которому при перемножении двух приближенных чисел их относительные погрешности складываются. Тогда, используя полученную, согласно [1], связь между относительной ошибкой и числом верных разрядов приближенного числа ($\delta_i \geq 2^{-m}$, где m – число разрядов дробной части числа), для числа верных разрядов произведения двух чисел можно записать:

$$m_{\Sigma} = m_1 - \log_2(1 + 2^{-\Delta m}),$$

где m_1 и Δm – минимальная разрядность дробной части и разность разрядностей сомножителей соответственно.

Если разрядности сомножителей одинаковы, то, согласно приведенному соотношению, число верных разрядов результата будет на единицу меньше, чем у сомножителей.

Возвращаясь к графу на рис. 1, можно сказать, что разрядности табличных значений должны быть, как минимум, на 3 разряда больше разрядности результата. В нашем случае табличные значения должны содержать не менее 35 верных разрядов. В тестируемом ниже варианте программы число верных разрядов табличных данных было принято равным 36.

На рис. 2 приведены временные диаграммы моделирования рассмотренного алгоритма в среде САПР Quartus II фирмы Альтера [5], а в таблице – расшифровка полученных результатов.

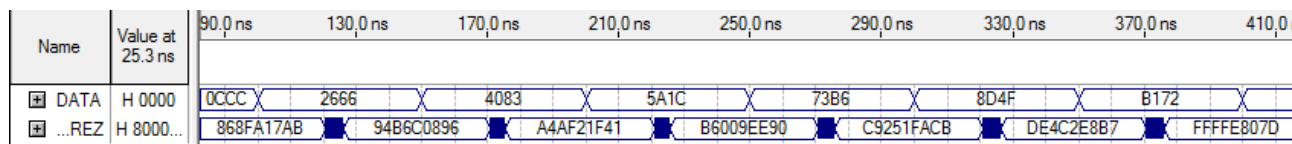


Рис. 2. Временные диаграммы моделирования рассмотренного алгоритма

Таблица. Результаты тестирования проекта

Аргумент	Результат вычисления	Абсолютная ошибка	Число верных разрядов
H"0CCC"	H"868FA17AB"	5.69E-11	35
H"2666"	H"94B6C0896"	1.04E-10	34
H"4083"	H"A4AF21F41"	6.01E-11	34
H"5A1C"	H" B6009EE90"	1.1E-10	34
H"73B6"	H" C9251FACB"	1.59E-10	33
H"8D4F"	H" DE4C2E8B7"	1.27E-10	33
H"B172"	H" FFFFE807D"	1.73E-10	33

При реализации проекта на ПЛИС серии Stratix III потребовалось 27 комбинационных таблиц (ALUTs) и 28 18-битных DSP блоков. Максимальное время вычисления не превосходит 27 nS. Анализ результатов тестирования показал, что во всем диапазоне изменения аргументов получено не менее 33 верных разрядов результата, что подтверждает правильность выбора точности табличных значений функции.

Следует отметить, что увеличение разрядности аргумента, согласно выражениям (2) и (3), не сильно скажется на времени вычисления. Так, при разрядности аргумента не более 32 в графе на рис. 1 появится еще только один дополнительный ярус умножения, то есть время вычисления увеличится на время t_{MULT} . Значительно изменится ширина графа, то есть количество одновременно выполняемых операций.

Согласно данным из User Guide для мегафункции altpf_exp фирмы Альтера, вычисление функции $\exp(x)$ при числе верных разрядов результата 23 требует: комбинационных блоков

(ALUTs) – 631 шт, специализированных регистров (DLRs) – 521 шт, модулей ALMs – 445 шт, 18-ти разрядных DSP блоков – 19 шт. Общее время вычисления для ПЛИС той же серии составляет 62 nS. Следует отметить, что мегафункция `altpf_exp` работает с данными, представленными, согласно стандартам IEEE 754, что делает ее более универсальной. Однако при ограниченном диапазоне изменения аргумента рассмотренный алгоритм обладает явным преимуществом как по затратам аппаратных ресурсов ПЛИС, так и по быстродействию.

Список литературы

1. Демидович Б. П., Марон И. А. Основы вычислительной математики. – Изд-во «Наука», Москва, 1966. – 664 с.
2. Люстерик Л. А., Червоненкис О. А., Янпольский А. Р. Математический анализ. Вычисление элементарных функций. М.: Физматгиз, 1963. – 248 с.
3. Опадчий Ю. Ф., Чумакова Е. В. Реализация на ПЛИС вычисления элементарных математических функций // Проектирование и технология электронных средств. – Владимир. – 2005. – № 4. – С. 7–12.
4. Пан В. Я. О способах вычисления значений многочленов // Успехи математических наук. – 1966. – Т. 21. – № 1. – С. 103–134.
5. Стешенко В. Б. ПЛИС фирмы Altera. Проектирование устройств обработки сигналов. – М.: Додека-XXI, 2000. – 128 с.

Рецензенты:

Волков Н.Н., д.т.н., профессор Московского государственного технического университета им. Н. Э. Баумана (МГТУ им. Н. Э. Баумана), кафедра «Информационные системы и телекоммуникации», г.Москва.

Шевцов Д.А., д.т.н., профессор Московского авиационного института (национального исследовательского университета), кафедра № 306 «Микроэлектронные электросистемы», г.Москва.