

ПРИМЕНЕНИЕ МЕТОДА БВЕ ДЛЯ ВЫЧИСЛЕНИЯ ФУНКЦИИ EXP(X) НА ПЛИС

Попов С.Д., Опадчий Ю.Ф.

ФГБОУ ВПО «МАТИ – Российский государственный технологический университет
имени К.Э. Циолковского», Москва, e-mail: electron_inform@mail.ru

В данной статье рассмотрена модификация метода БВЕ для вычисления экспоненциальной функции применительно к ПЛИС. В целях повышения быстродействия и уменьшения себестоимости метод БВЕ комбинируется с таблицами заранее вычисленных значений. Также рассмотрен способ приведения произвольного аргумента, так как область определения метода БВЕ ограничена промежутком $[0, 0,25]$. Оценки быстродействия и аппаратных затрат проведены с помощью САПР Quartus II фирмы Altera применительно к ПЛИС серии Stratix III. Проведено сравнение предложенного способа вычисления экспоненциальной функции и встроенной в САПР Quartus II мегафункцией `altpf_exp`, в результате чего показано, что предложенная комбинация табличного метода и метода БВЕ позволяет уменьшить как время вычисления, так и затраты аппаратных ресурсов ПЛИС на реализацию.

Ключевые слова: ПЛИС, алгоритм вычисления, экспоненциальная функция

APPLICATION OF THE FEE METHOD TO CALCULATE EXP(X) FUNCTION ON FPGA

Popov S.D., Opadchiy Y.F.

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

In this article the modification of the FEE method is used to calculate the exponential function on FPGA. FEE method is combined with tables of pre-computed values in order to improve performance and reduce the cost of hardware resources. Also a method of bringing arbitrary argument is considered, as the domain of the FEE method is limited by the interval $[0, 0.25]$. Evaluation of performance and hardware costs carried by Altera Quartus II software, Altera Stratix III is used as a target FPGA. The proposed method for calculating the exponential function is compared with built into Quartus II software megafunction `altpf_exp`, whereby is shown that the proposed combination of table method and the FEE method reduces computation time as well as the cost of hardware resources of FPGA.

Keywords: FPGA, algorithm of calculation, exp(x) function

Разработка специализированных вычислителей на ПЛИС часто требует нахождения численного значения элементарных математических функций. Обычно для этого используются имеющиеся в стандартном пакете САПР мегафункции. Так, например, в САПР Quartus II фирмы «Альтера» [5], имеются мегафункции для вычисления основных математических функций, такие как `altpf_exp`, `altpf_log`, `altpf_sincos`. Эти мегафункции работают с данными, представленными согласно стандарту IEEE 754

в формате с плавающей точкой. Допускается использование вычислений с одинарной (1 бит знака, 8 бит – порядок, 23 бита – мантисса), двойной (1 бит знака, 11 бит – порядок, 52 бита – мантисса) и одинарной расширенной (1 бит знака, 31 – 52 мантисса) точностями.

Ниже приведена таблица, заимствованная из документации к САПР Quartus II, характеризующая аппаратные затраты и производительность мегафункции `altpf_exp` применительно к ПЛИС серии Stratix III.

Таблица 1

Параметры мегафункции `altpf_exp`

Точность	Задержка (такты)	Логические элементы				Макс. частота работы, МГц
		Таблицы перекодировок (ALUTs)	Спец. регистры (DLRs)	Модули ALMs	18-битные DSP блоки	
Одинарная	17	631	521	445	19	275
Двойная	25	4138	2022	2959	46	257

Согласно приведенным данным, время получения результата при одинарной точности составляет порядка 62 нс, а при двойной точности – порядка 98 нс.

На практике диапазон изменения аргумента заранее известен, и он, как правило,

значительно уже, чем допускает мегафункция. Поэтому практический интерес представляет поиск более гибких решений, обеспечивающих получение требуемой точности вычисления при меньших аппаратных затратах и большем быстродействии.

Нахождение численного значения элементарных функций всегда сводится к вычислению некоторого степенного многочлена. Широко известные методы вычисления многочлена, такие как метод Горнера [2], методы с предварительной обработкой коэффициентов [4], хотя и позволяют ускорить вычисление многочлена, но по своей структуре фактически нацелены на последовательное, итерационное вычисление, свойственное классическим вычислительным машинам.

Применение ПЛИС позволяет произвольно синтезировать структуру вычислителя, широко используя распараллеливание процесса, что предполагает значительное сокращение времени получения результата. Одним из методов, позволяющих реализовать такую возможность, является метод

БВЕ (Быстрое Вычисление Е-функций). Рассмотрим особенности применения этого метода на примере вычисления функции $\exp(x)$.

Алгоритм, описанный в [1], предполагает, что диапазон изменения аргумента удовлетворяет условию $0 < x_0 < 0,25$ и число верных разрядов результата $n > 8$. Расчетная разрядность аргумента для требуемого числа верных разрядов результата определяется выражением $N = 2^{k+1}$, где $k = \text{CEIL}(\ln n / \ln 2)$.

Сгруппируем весовые коэффициенты расчетного аргумента

$$x_N = \alpha_3 \cdot 2^{-3} + \alpha_4 \cdot 2^{-4} + \alpha_5 \cdot 2^{-5} + \dots + \alpha_N \cdot 2^{-N}$$

по степеням 2^{-2^m} :

$$x_N = \beta_2 \cdot 2^{-4} + \beta_3 \cdot 2^{-8} + \beta_4 \cdot 2^{-16} + \dots + \beta_{k+1} \cdot 2^{-N_0}, \quad (1)$$

где

$$\begin{aligned} \beta_2 &= \alpha_3 \alpha_4, \quad \beta_3 = \alpha_5 \alpha_6 \alpha_7 \alpha_8, \\ \beta_4 &= \alpha_9 \alpha_{10} \alpha_{11} \alpha_{12} \alpha_{13} \alpha_{14} \alpha_{15} \alpha_{16}, \\ \beta_5 &= \alpha_{17} \alpha_{18} \alpha_{19} \alpha_{20} \dots \alpha_{32} \end{aligned}$$

и так далее. В общем случае β_v есть некоторое целое двоичное число, полная запись которого определяется выражением:

$$\beta_v = \sum_{i=0}^{2^{v-1}-1} a_{2^{v-1}+i} \cdot 2^{-2^{v-1}-1-i}.$$

Используя выражение (1), функцию $\exp(x_N)$ можно представить в следующем виде

$$\exp(x_N) = \prod_{v=2}^{k+1} \exp(\beta_v \cdot 2^{-2^v}). \quad (2)$$

Выражение (2) позволяет разбить вычислительный процесс определения численного значения функции $\exp(x_N)$ на несколько параллельно выполняемых ветвей.

Вычисление каждой ветви выполняется с использованием ее разложения в ряд Маклорена:

$$\exp(\beta_v \cdot 2^{-2^v}) = 1 + \frac{\beta_v \cdot 2^{-2^v}}{1!} + \frac{(\beta_v \cdot 2^{-2^v})^2}{2!} + \frac{(\beta_v \cdot 2^{-2^v})^3}{3!} + \dots + \frac{(\beta_v \cdot 2^{-2^v})^r}{r!} + R_v(r), \quad (3)$$

причем число членов каждого разложения определяется выражением:

$$r = N \cdot 2^{1-v}. \quad (4)$$

Так, например, для вычисления функции с 12 верными разрядами имеем $k = 4$, $N = 32$,

$$x_N = \beta_2 \cdot 2^{-4} + \beta_3 \cdot 2^{-8} + \beta_4 \cdot 2^{-16} + \beta_5 \cdot 2^{-32};$$

$$\exp(x_N) = \exp(\beta_2 \cdot 2^{-4}) \cdot \exp(\beta_3 \cdot 2^{-8}) \cdot \exp(\beta_4 \cdot 2^{-16}) \cdot \exp(\beta_5 \cdot 2^{-32}),$$

причем число членов в соответствующих разложениях ряда Маклорена определяется из условия (4) следующим образом: $r = 16$ для $\exp(\beta_2 \cdot 2^{-4})$, $r = 8$ для $\exp(\beta_3 \cdot 2^{-8})$, $r = 4$ для $\exp(\beta_4 \cdot 2^{-16})$, $r = 2$ для $\exp(\beta_5 \cdot 2^{-32})$.

Из приведенного примера видно, что чем больше в (3) значение v , тем меньше членов в соответствующем разложении, то есть тем проще реализация вычислителя.

На рис. 1 приведен граф вычислительного процесса, соответствующий выражению (2) для 12 верных разрядов результата.

$$\exp(\beta_5 \cdot 2^{-32}) \approx 1.0000000000000000 \alpha_{17} \alpha_{18} \alpha_{19} \alpha_{20} \alpha_{21} \alpha_{22} \alpha_{23} \alpha_{24} \alpha_{25} \alpha_{26} \alpha_{27} \alpha_{28} \alpha_{29} \alpha_{30} \alpha_{31} \alpha_{32} \cdot$$

Модуль BL_B2 реализует вычисление $\exp(\beta_2 \cdot 2^{-4})$, модуль BL_B3 – вычисление $\exp(\beta_3 \cdot 2^{-8})$, модуль BL_B4 – вычисление $\exp(\beta_4 \cdot 2^{-16})$, а модуль BL_B5 – вычисление $\exp(\beta_5 \cdot 2^{-32})$. Очевидно, что длина графа для каждого модуля будет определяться значением r . Так, $\exp(\beta_5 \cdot 2^{-32}) \approx 1 + \beta_5 \cdot 2^{-32}$, что не требует выполнения каких-либо вычислений и сводится к перезаписи соответствующих разрядов аргумента в результат:

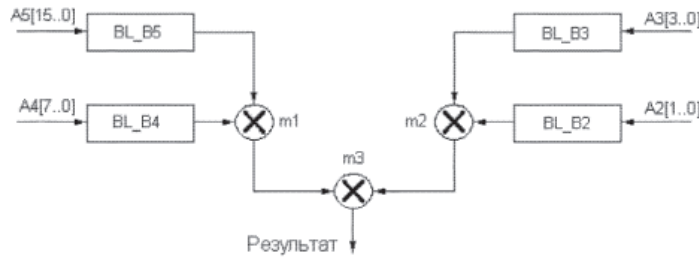


Рис. 1. Граф вычислительного процесса, соответствующий выражению (2) для 12 верных разрядов результата

Модуль BL_B3 реализует вычисление согласно выражению:

$$\exp(\beta_4 \cdot 2^{-16}) \approx \tilde{\beta}_4 \cdot 2^{-16} \cdot \left(\frac{1}{2} + \frac{\tilde{\beta}_4}{A_3} \cdot 2^{-10} \right) \cdot +1 + \tilde{\beta}_4 \cdot 2^{-8},$$

где $\tilde{\beta}_4 = 0.\alpha_9\alpha_{10}\alpha_{11}\alpha_{12}\alpha_{13}\alpha_{14}\alpha_{15}\alpha_{16}$ – число < 1 , и $A_3 = 1,5$.

Граф процесса, реализованный в данном модуле, показан на рис. 2. Его реализация требует только двух умножителей. Очевидно, что графы, реализованные в модулях BL_B2 и BL_B3, имеют значительно более сложную структуру.

Результаты моделирования проекта в среде Quartus II приведены на рис. 3. Минимальное число верных разрядов результата равно 33, что значительно превосходит исходные требования (12 разрядов). Следовательно, для заданной точности такое решение является избыточным. Аппаратные затраты на реализацию вычислителя сведены в табл. 2.

Анализ полученных результатов показал, что основные затраты аппаратных ресурсов расходуются для реализации модулей BL_B3 и BL_B2. При этом на входах данных модулей соответственно действуют 2 и 4-разрядные аргументы, то есть возможное число выходных значений соответственно равны 4 и 16. В этом случае целесообразно отказаться

от применения в этих модулях разложения в ряд Маклорена в пользу табличного выбора заранее рассчитанных результатов соответствующих аргументов.

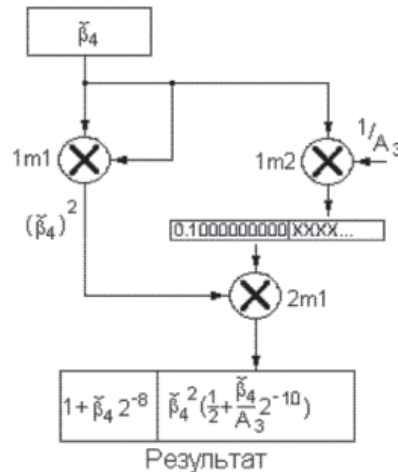


Рис. 2. Граф модуля BL_B3

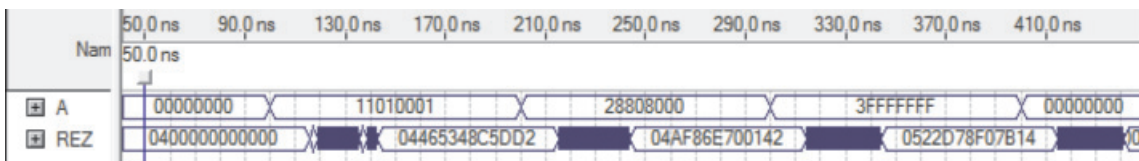


Рис. 3. Результаты моделирования проекта в среде Quartus II

Таблица 2

Аппаратные затраты на реализацию метода БВЕ

Модуль	ALUTs блоки	18 бит DSP блоки
BL_B5	0	0
BL_B4	26	11
BL_B3	72	26
BL_B2	271	56
Общие для рис. 1	622	139

Реализация такого подхода в среде Quartus II потребовала 49 ALUTs и 17 DSP блоков. При этом в модулях BL_B2 и BL_B3 использованы константы с 32 верными разрядами. Суммарное время вычисления не превосходит 30 нс. Минимальное число верных разрядов результата равно 31.

Следует отметить, что, так как константы в модулях BL_B2 и BL_B3 могут быть рассчитаны с произвольным числом верных разрядов, общая точность вычисления

определяется процессами в модулях BL_B4 и BL_B5.

Расчеты показывают, что предельная точность вычисления в модуле BL_B5 по предложенному алгоритму соответствует максимальному значению аргумента и составляет 34 верных разряда результата. Предельная точность вычисления по алгоритму на рис. 2 для модуля BL_B4 составляет

$$\exp(x_{\text{РАСЧЕТ}} + \text{const}) = \exp(x_{\text{РАСЧЕТ}}) \cdot \exp(\text{const}), \quad (5)$$

где $x_{\text{РАСЧЕТ}}$ – расчетное значение аргумента из диапазона $0 < x_0 < 0,25$; $\exp(\text{const})$ – заранее вычисленная константа.

Это потребует использования дополнительного умножителя и блока памяти с вычисленными значениями $\exp(\text{const})$.

Наиболее просто решить задачу расширения диапазона изменения аргумента при реализации на ПЛИС можно в случае, если изменение аргумента лежит в диапазоне $0 \leq x < \ln(2)$ [3]. В этом случае результат при более широком диапазоне изменения аргумента сдвигается влево на величину s , равную неполному частному от деления действительного аргумента на $\ln(2)$:

$$s = \text{FLOOR} \left[\frac{x}{\ln(2)} \right].$$

При таком подходе реализация выражения (5) потребует введения в устройство блока вычисления s , а сама функция ищется от нового аргумента $x_{\text{РАС}}$, равного:

$$x_{\text{РАС}} = x \cdot \frac{1}{\ln(2)} - \text{FLOOR} \left[\frac{x}{\ln(2)} \right].$$

Полученный результат при помощи сдвигового регистра может быть сдвинут на s разрядов влево, либо представлен в показательной форме в виде порядка и мантииссы. При этом порядок – целое число, а мантиисса всегда содержит один разряд целой части. Очевидно, что такой подход вносит в расчет дополнительную погрешность, обусловленную приближенностью значения $1/\ln(2)$. Для ее уменьшения желательно число верных разрядов констант выбирать большим числа верных разрядов аргумента, как минимум на один.

На рис. 4 приведена заимствованная из [3] структура блока подготовки данных, выделяющая из исходного аргумента порядок и новое расчетное значение аргумента, удовлетворяющее условию $0 \leq x < \ln(2)$. Ее реализация требует двух умножителей.

Для использования такого подхода модуль BL_B2 должен содержать все констан-

ты, удовлетворяющие условию $0 \leq x < \ln(2)$, что легко обеспечить на практике.

37 верных разрядов. Эти цифры являются определяющими при выборе разрядностей констант в модулях BL_B2 и BL_B3. Как было отмечено ранее, рассмотренный алгоритм позволяет вычислить значение функции при условии, что аргумент лежит в диапазоне $0 < x_0 < 0,25$. При больших значениях аргумента результат может быть получен с использованием выражения:

ты, удовлетворяющие условию $0 \leq x < \ln(2)$, что легко обеспечить на практике.

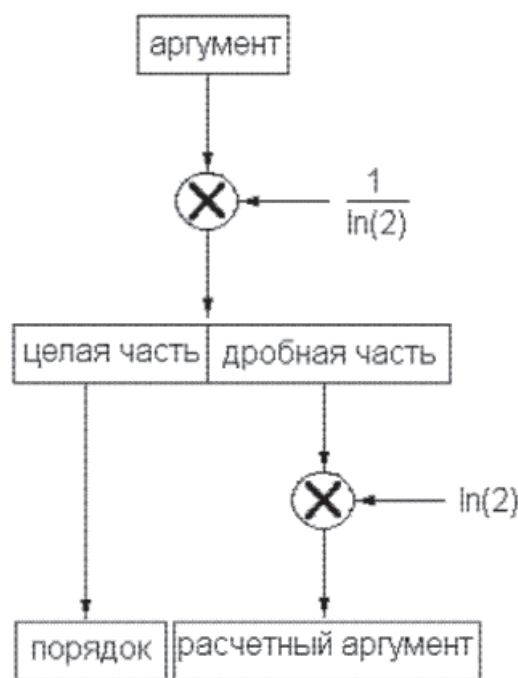


Рис. 4. Структура блока подготовки данных.

С использованием описанного подхода был разработан вычислитель для функции $\exp(x)$, формирующий на выходе порядок и мантииссу. Его основные параметры:

- исходный аргумент 4 разряда целой части + 33 разряда дробной части;
- разрядность порядка – 5 разрядов;
- разрядность мантииссы – 1 разряд целой части + 34 разряда дробной части;
- аппаратные затраты – 117 блоков ALUTs и 29 DSP блоков;
- максимальное время вычисления на ПЛИС Stratix III не превосходит 45 nS;
- минимальное число верных разрядов на выходе – 30 разрядов.

На рис. 5 приведены временные диаграммы, полученные в результате моделирования вычислителя в среде Quartus II фирмы Altera (A – аргумент, man – мантиисса, rog – порядок).

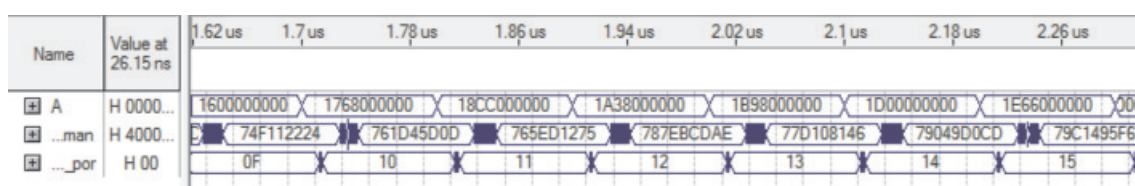


Рис. 5. Временные диаграммы, полученные в результате моделирования вычислителя

Из проведенных исследований следует вывод, что комбинация табличного и БВЕ методов делает возможным разработать вычислитель для функции $\exp(x)$, который при ограниченном диапазоне изменения аргумента позволяет реализовать устройство с меньшими аппаратными затратами и большим быстродействием. При увеличении точности вычисления достоинства предлагаемого метода становятся более ощутимыми.

Список литературы

1. Карацуба Е.А. Быстрые алгоритмы и метод БВЕ. URL: <http://www.ccas.ru/personal/karatsuba/alg.htm> (дата обращения: 19.04.2012).
2. Люстерик Л.А., Червоненкис О.А., Янпольский А.Р. Математический анализ. Вычисление элементарных функций. – М.: Физматгиз, 1963. – 248 с.
3. Мо Чжо Чо, Опадчий Ю.Ф. Алгоритм вычисления показательной функции на ПЛИС // Новые материалы и технологии: сборник трудов Всероссийской научно-технической конференции (Москва, 11–13 ноября 2008 г.). – М., 2008. – Т. 3. – С. 113–115.
4. Пан В.Я. О способах вычисления значений многочленов // Успехи математических наук. – 1966. – Т. 21. – № 1. – С. 103–134.
5. Стешенко В.В. ПЛИС фирмы Altera. Проектирование устройств обработки сигналов. – М.: Додека-XXI, 2000. – 128 с.

References

1. Karacuba E.A. Bystrye algoritmy i metod BVE. Available at: <http://www.ccas.ru/personal/karatsuba/alg.htm> (accessed 19 April 2012).
2. Ljusterik L.A., Chervonenkis O.A., Janpolskij A.R. Matematicheskij Analiz. Vychislenie Jelementarnyh Funkcij. Fizmatgiz, Moscow, 1963, 248 p.
3. Mo Chzho Cho, Opadchij Yu.F. Algoritm Vychislenija Pokazatelnoj Funkcii na PLIS // *Sbornik Trudov Vserossijskoj Nauchno-Tehnicheskoy Konferencii «Novye Materialy i Tehnologii»* (Moscow, 11–13 November 2008). Moscow, 2008, vol. 3, pp. 113–115.
4. Pan V.Ja. O Sposobah Vychislenija Znachenij Mnogochlenov. // *Uspеhi Matematicheskikh Nauk*, 1966, vol. 21, no. 1, pp. 103–134.
5. Steshenko V.B. PLIS firmy Altera. Proektirovanie ustroystv obrabotki signalov. M.: Dodeka-XXI, 2000, 128 p.

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), кафедры № 306 «Микроэлектронные электросистемы», г. Москва.

Работа поступила в редакцию 18.09.2013.