

РЕАЛИЗАЦИЯ ЦИФРОВЫХ ИНТЕГРАТОРОВ НА ПЛИС

Национальный технический университет Украины "КПИ"

Показана возможность улучшения технико-экономических характеристик цифровых интеграторов, реализованных на базе микросхем FPGA, за счет применения неавтономных методов выполнения арифметических операций при вычислении экстраполяционных формул квадратичных и кубических парабол.

Введение

Расширение области применения систем управления и моделирования в реальном времени предъявляет повышенные требования к быстродействию и надежности вычислительных средств для решения широкого круга задач. К таким задачам можно отнести решение систем дифференциальных уравнений, линейных и нелинейных алгебраических уравнений, трансцендентных уравнений, вычисление сложных интегралов и функциональных зависимостей.

Один из подходов к решению указанных задач состоит в применении цифровых интегрирующих машин (ЦИМ), реализующих эквивалентные системы уравнений Шеннона. Характеристики точности, быстродействия и аппаратных затрат ЦИМ зависят от используемых формул численного интегрирования, алгоритмов выполнения арифметических операций и элементной базы.

С точки зрения быстродействия, точности и возможности использования однотипных вычислительных блоков предпочтительными являются экстраполяционные параллельные ЦИМ, реализующие формулы точного интегрирования по Стильтесу, рассчитанные на работу с фиксированной точкой [1]. Основными операциями в таких ЦИМ являются сложение и умножение, которые обычно выполняются по правилам параллельной арифметики.

Для реализации ЦИМ могут быть эффективно использованы программируемые логические интегральные схемы (ПЛИС), например, типа *FPGA* (*Field*

Programmable Gate Array) [2]. Это обусловлено их архитектурными особенностями, связанными с большой степенью интеграции функциональных элементов, высоким быстродействием и надежностью. Кроме того, наличие развитых средств автоматизированного проектирования систем на базе *FPGA* упрощает и ускоряет процесс получения готового продукта.

Основу ЦИМ составляют цифровые интеграторы (ЦИ), количество которых определяет функциональные возможности ЦИМ. Число вариантов построения ЦИМ на основе N ЦИ, определяется показательной зависимостью [1, с. 421]

$$Q = 2^{2N(N-1)}. \quad (1)$$

Из (1) видно, что для повышения функциональных возможностей ЦИМ необходимо увеличивать количество ЦИ в ее составе, что в свою очередь увеличивает необходимую ресурсоемкость *FPGA* (число логических блоков и соединительных элементов). При использовании параллельных операционных блоков (ОБ) это объясняется не только их сложностью, но и необходимостью пересылки внутри и между ЦИ многоразрядных слов, на что дополнительно затрачиваются коммутационные ресурсы *FPGA*.

Рассмотрим метод построения ЦИМ, позволяющий снизить необходимую ресурсоемкость *FPGA* по сравнению с известными методами при реализации одинаковых формул численного интегрирования.

Методи реалізації ЦИ

Будем рассматривать следующие экстраполяционные формулы численного интегрирования по Стилтьесу на основе первых разностей.

Формула квадратичных парабол [1]:

$$\Delta z_{(i+1)} = \Delta y_{q(i+1)}^2 (y_{pi} + \frac{1}{12} \Delta y_{pi}) + \frac{1}{2} \Delta y_{p(i+1)}^2 (\Delta y_{q(i+1)}^2 - \frac{1}{6} \Delta y_{qi}). \tag{2}$$

Здесь $\Delta z_{(i+1)}$ – значение приращения интеграла Стилтьеса на $(i+1)$ -м шаге; $\Delta y_{q(i+1)}^2$ – экстраполированное значение

приращения переменной интегрирования для $(i+1)$ -го шага; y_{pi} – значение подынтегральной функции на i -м шаге; $\Delta y_{p(i+1)}^2$ – экстраполированное значение приращения подынтегральной функции для $(i+1)$ -го шага; Δy_{qi} – значение приращения переменной интегрирования на i -м шаге; Δy_{pi} – значение приращения подынтегральной функции на i -м шаге.

Ярусно параллельная форма (ЯПФ) графа вычислений по формуле (2) представлена на рис. 1.

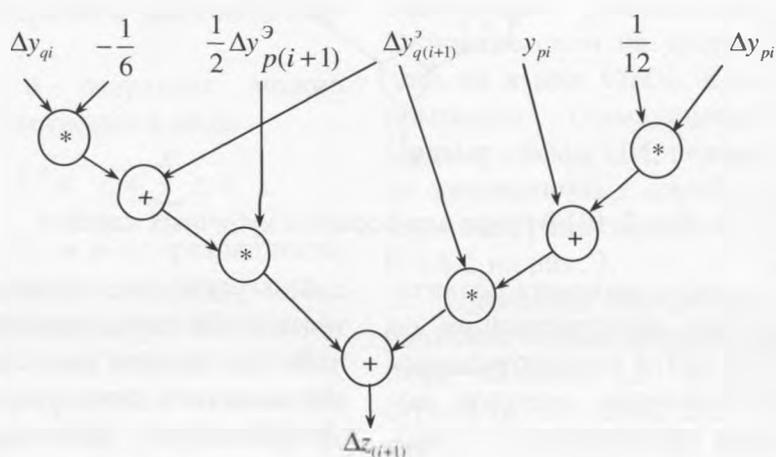


Рис. 1. ЯПФ графа для формулы квадратичных парабол

Формула кубичных парабол [1]:

$$\Delta z_{(i+1)} = \Delta y_{q(i+1)}^2 (y_{pi} + \frac{1}{2} \Delta y_{p(i+1)}^2 + \frac{1}{6} (\Delta y_{pi} - \frac{1}{4} \Delta y_{p(i-1)})) - \frac{1}{6} \Delta y_{p(i+1)}^2 (\Delta y_{qi} - \frac{1}{4} \Delta y_{q(i-1)}), \tag{3}$$

где $\Delta y_{q(i-1)}$ – значение приращения переменной интегрирования на $(i-1)$ -м шаге, $\Delta y_{p(i-1)}$ – значение приращения подынтегральной функции на $(i-1)$ -м шаге.

ЯПФ графа вычислений по формуле (3) представлена на рис. 2.

При анализе возможных методов реализации ЦИ на базе традиционных ОБ параллельного типа [3] с учетом (2), (3) и архитектурных особенностей *FPGA* необходимо отметить следующее. Каждая итерация численного интегрирования предполагает вычисление очередных прираще-

ний интегралов Стилтьеса. Количество таких итераций в процессе решения определенной задачи с помощью ЦИМ может быть значительным (сотни и тысячи). Поэтому для систем реального времени нужно минимизировать количество тактов на выполнение одной итерации. При рассмотрении критических путей ЯПФ видно, что основную составляющую в общее время вычислений вносят операции умножения. В связи с этим целесообразно использовать быстродействующие ОБ умножения (например, матричного типа). Однако эквивалентная система уравнений Шеннона может предполагать использование большого количества ЦИ, что при многозарядных вычислениях приводит к большим аппаратным затратам, то есть предъявляет высокие требования к ресурсоемкости *FPGA*. В свою очередь, это

может потребовать использование дорогостоящих микросхем старшего ряда семейств *FPGA*. Как правило, построение ЦИ путем коммутации параллельных ОБ в соответствии со структурой ЯПФ превышает допустимую ресурсоемкость *FPGA*.

Поэтому в качестве первого варианта рассматривалась архитектура ЦИ на базе операционного автомата, включающего параллельный умножитель и сумматор, а также управляющего автомата.

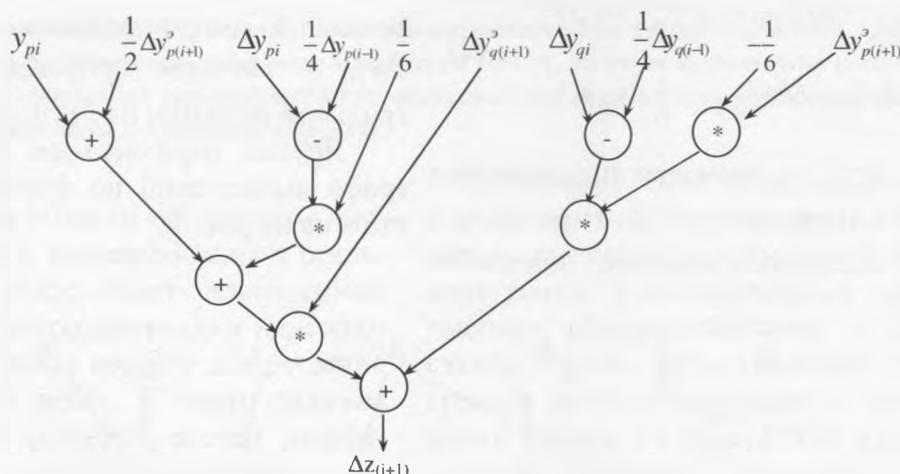


Рис. 2. ЯПФ графа для формулы кубических парабол

В качестве второго варианта реализации ЦИ численного интегрирования по Стилтесу на базе *FPGA* рассматривается применение неавтономных методов вычислений, которые реализуются с помощью вычислительных устройств квазипараллельного типа. Известно, что такие методы существенно уменьшают количество используемых выводов микросхем, позволяют совмещать процессы поразрядного ввода и переработки информации, а также улучшают основные технико-экономические параметры вычислительных устройств в интегральном исполнении [4]. Кроме того, поразрядная пересылка информации между ОБ квазипараллельного типа снижает требуемую коммутационную ресурсоемкость *FPGA*. Архитектуры ОБ сложения и умножения квазипараллельного типа основываются на алгоритмах, предложенных в [5-7]. При этом архитектуры квазипараллельных ЦИ основываются на структурном методе [5], предполагающем коммутацию квазипараллельных ОБ в соответствии с ЯПФ графа алгоритма вычислений. Применение таких ОБ является эффективным спо-

собом ускорения выполнения последовательностей нераспараллеливаемых операций. На каждом шаге вычислений в ОБ вводятся по одному разряду операндов и с задержкой на небольшое число шагов формируется очередной разряд результата, начиная со старшего. При выполнении последовательности зависимых операций разряд промежуточного результата, полученный на i -м шаге в одном ОБ может быть использован на $(i+1)$ -м шаге в другом ОБ в качестве очередного разряда операнда. В этом случае выполнение каждой последующей операции начинается сразу же после получения первого (старшего) разряда результата этой операции. В связи с этим последовательности зависимых операции выполняются в режиме частичного совмещения, что создает предпосылки для уменьшения общего времени вычислений по сравнению с ОБ параллельного типа. Для представления операндов и результата в квазипараллельных ОБ используется избыточная система счисления (например, симметричная двоичная система с цифрами $\{-1, 0, 1\}$ [5,6]).

Алгоритм умножения

При численном интегрировании используется операция умножения переменной на константу (см. рис. 1 и 2). На основе метода неавтономного умножения [5] можно получить алгоритм умножения на константу, который требует меньших затрат оборудования по сравнению с обычным алгоритмом неавтономного умножения двух переменных.

Алгоритм обеспечивает вычисление функции

$$Z = 2^{-1} X C.$$

Здесь C – константа, представленная параллельным кодом, X – переменная в избыточной симметричной двоичной системе счисления.

Переменную и результат можно представить соответственно в виде

$$X = \sum_{i=1}^n x_i 2^{-i} \text{ и } Z = \sum_{i=1}^m z_i 2^{-i},$$

где $x_i, z_i \in \{-1, 0, 1\}$, n и m – разрядности соответственно переменной и результата.

Алгоритм включает следующие действия (R_i и H_i – промежуточные переменные с нулевыми начальными значениями).

1. Для $i=1$ выполнить пункты 3 и 4, принять $z_1=0$.

$$2. H_i = 2R_i + \frac{1}{2} x_i C.$$

$$3. R_i = H_i.$$

4. Для $i=2$ до m выполнять пункты 2, 5 и 6.

$$5. z_i = \begin{cases} -1, & \text{если } H_i < -\frac{1}{2}; \\ 0, & \text{если } -\frac{1}{2} \leq H_i < \frac{1}{2}; \\ 1, & \text{если } H_i \geq \frac{1}{2}. \end{cases}$$

$$6. R_i = H_i - z_i.$$

Приведенный алгоритм обеспечивает совмещение процессов поразрядного ввода операнда X со старших разрядов и поразрядного формирования результата Z с запаздыванием на один шаг.

Результаты исследования вариантов реализации ЦИ

На основе описанных выше вариантов архитектур были разработаны ЦИ различной разрядности на базе семейства *FPGA Altera Cyclone*. Для всех вариантов ЦИ использовалась микросхема *EP1C12Q240I7*. Моделирование производилось с применением САПР *Mentor Graphics ModelSim SE v. 6.1g*, а синтез схем ЦИ выполнялся с помощью САПР *Altera Quartus II v.6.1*. При этом схемный синтезатор и загрузчик САПР настраивались на получение схемы с минимальной ресурсоемкостью и максимальной компактностью размещения в кристалле. Описание схем на уровне ОБ производилось на языке *VHDL*, а на уровне ЦИ – с помощью схематического редактора. Пример схемы ЦИ, реализующего формулу квадратичных парабол, представлен в терминах редактора САПР *Altera Quartus II v.6.1* на рис. 3.

Сравнение вариантов ЦИ по характеристикам использования логических ячеек (*logic elements, LE*), коммутационных ресурсов (*direct links, DL*) и максимальной частоты тактирования схем (*fmax*) проведено в зависимости от разрядности вычислений.

Полученные результаты отражены на рис. 4-6, где кривые 1 и 2 получены для ЦИ параллельного типа по формулам соответственно квадратичных и кубических парабол, а кривые 3 и 4 – аналогично для ЦИ квазипараллельного типа.

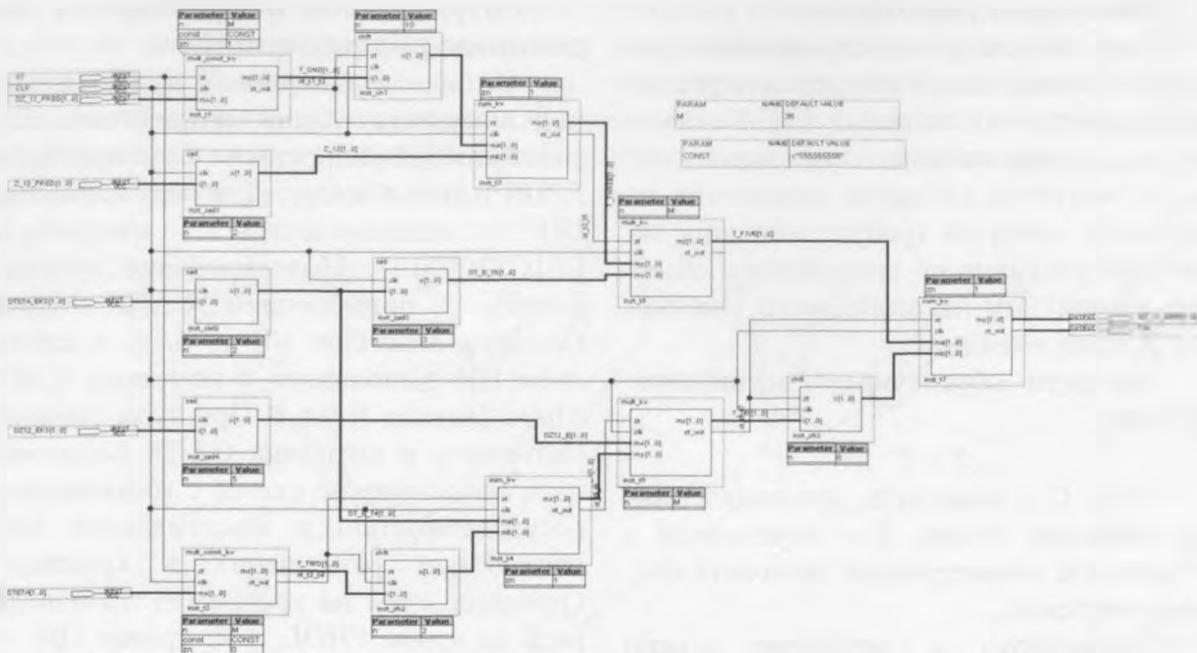


Рис. 3. Схема ЦИ, реализующего формулу квадратичных парабол

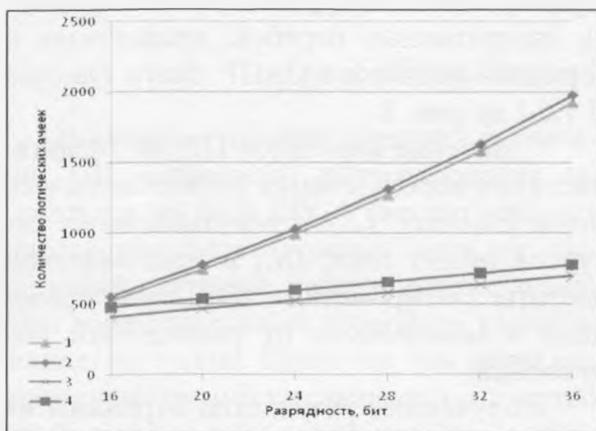


Рис. 4. Использование логических ячеек LE

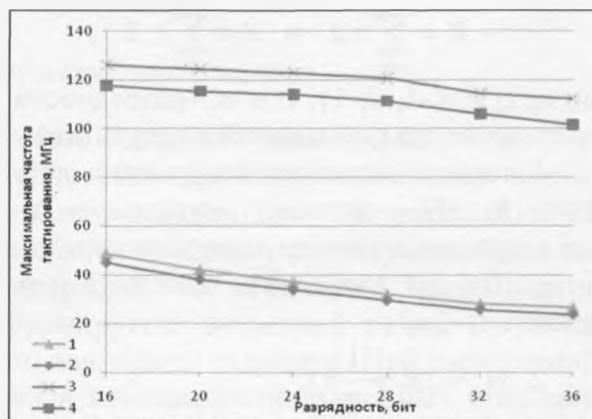


Рис. 6. Максимальная частота тактирования ЦИ

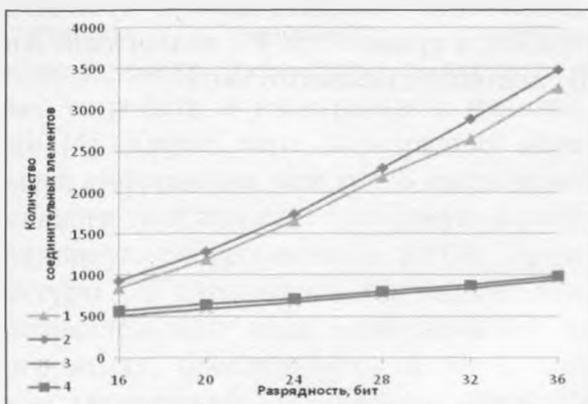


Рис. 5. Использование коммутационных ресурсов DL

Из приведенных зависимостей видно, что применение квазипараллельных ОБ снижает требования к ресурсоемкости *FPGA*. С увеличением разрядности вычислений это преимущество возрастает в 2 и более раз. Это позволяет намного компактнее размещать схемы ЦИ квазипараллельного типа внутри кристалла *FPGA*. Благодаря архитектурным особенностям квазипараллельных ОБ появляется возможность увеличения максимальной частоты тактирования схем, причем, по сравнению с ЦИ параллельного типа частота в меньшей степени зависит от разрядности вычислений. При большой разрядности вычислений преимущество квазипараллельных ЦИ по этому показателю

увеличивается в 3-4 раза. Использование методов неавтономной арифметики при реализации ЦИМ может привести к большому количеству тактов на одну итерацию численного интегрирования по сравнению с вариантом использования ЦИ параллельного типа. Однако за счет большей частоты тактирования квазипараллельные ЦИМ могут значительно компенсировать этот недостаток.

Выводы

Применение неавтономных методов вычислений позволяет реализовать большее количество интеграторов на одной микросхеме и, следовательно, увеличить функциональные возможности ЦИМ.

Реализация вычислительных систем на одной микросхеме, как правило, повышает их надежность, быстродействие и снижает стоимость.

Минимизация количества необходимых выводов *FPGA* в случае применения неавтономных методов вычислений позволяет увеличить число свободных выводов для других устройств, реализованных в этой же микросхеме.

Все это позволяет повысить эффективность реализации ЦИМ на базе программируемых микросхем.

Список литературы

1. *Каляев А.В.* Теория цифровых интегрирующих машин и структур. – М.: Советское радио, 1970. – 472 с.
2. *Стешенко В.Б.* *FPGA* фирмы *ALTERA*: проектирование устройств обработки сигналов. – М.: Додэка, 2000. – 128 с.
3. *Самофалов К.Г., Корнейчук В.И., Тарасенко В.П.* Цифровые ЭВМ. – К.: Выща школа, 1989. – 424 с.
4. *Жабин В.И., Ковалев Н.А.* Исследование методов построения вычислительных устройств на основе *FPGA* // Технология и конструирование в электронной аппаратуре. – 2002. – № 2. – С. 35-39.
5. *Жабин В.И., Корнейчук В.И., Тарасенко В.П.* Некоторые машинные методы вычисления рациональных функций многих переменных // Автоматика и телемеханика. – 1976. – № 12. – С. 145-154.
6. *Kishor S. Trivedi, Milos D. Ercegovac.* On-Line Algorithms for Division and Multiplication // *IEEE Transactions on Computers.* – 1977. – vol. C-26. – № 7 – P. 681-687.
7. *Жабин В.И., Корнейчук В.И., Макаров В.В., Тарасенко В.П.* Влияние точности вычисления на сложность квазипараллельных операционных устройств в мультипроцессорных системах // Автоматика и вычислительная техника. – 1982. – № 3. – С. 29-32.