

УДК 004.312.12

А. Е. ПЕРЕПЕЛИЦЫН*Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Украина***МЕТОД РАЗРАБОТКИ МУЛЬТИПАРАМЕТРИЗИРУЕМЫХ ПРОЕКТОВ ПРОГРАММИРУЕМОЙ ЛОГИКИ**

Предлагается классификация средств обеспечения гибкости проектов в рамках языка описания аппаратуры VHDL. Приводятся результаты экспериментального исследования зависимости количества ресурсов программируемых логических интегральных схем (ПЛИС), требуемых для реализации арифметических узлов, от ширины их входных данных. Анализируются особенности реализации на ПЛИС арифметических операций с фиксированной точкой. Демонстрируется, что при наличии аппаратно реализованных умножителей в составе ПЛИС рост количества требуемых ресурсов для реализации целочисленного умножителя при увеличении его разрядности происходит скачкообразно. Даются определения статической параметризации, сквозной параметризации и мультипараметризации FPGA проектов. Обсуждаются пути построения широко параметризуемых проектов на ПЛИС. Рекомендуются применение сквозной параметризации для достижения максимальной эффективности использования ресурсов FPGA. Предлагается последовательность разработки мультипараметризуемых проектов на ПЛИС и последовательность оценки количества требуемых ресурсов ПЛИС для реализации мультипараметризуемого проекта. Приводится практический пример применения описываемого метода разработки мультипараметризуемых проектов на ПЛИС, включающий реализацию операций умножения с накоплением промежуточных сумм.

Ключевые слова: FPGA, ПЛИС, параметризация, мультипараметризация, ресурсы ПЛИС.

Введение

Технология FPGA (Field Programmable Gate Array) находит широкое применение в решении задач построения нестандартных аппаратных решений.

Использование нестандартных библиотечных модулей (IP cores) позволяет снизить трудозатраты в процессе проектирования и повысить надежность проектов за счет протестированных компонентов [1].

Параметризация IP-ядер позволяет создавать универсальные проекты, которые могут быть адаптированы к требуемым условиям путем задания значений настроечных параметров. Язык VHDL позволяет также управлять структурой описываемого устройства в зависимости от значений параметров [2].

Аппаратная реализация операций умножения с накоплением промежуточных сумм, используемых в цифровой обработке сигналов, включая обработку изображений и томографической информации, предоставляет возможность выбора между объемом ресурсов (числом параллельных умножителей) и временем вычисления одного результата (количеством итераций). Этот компромисс между быстротой действия, надежностью и количеством использованных ресурсов предоставляет возможность построения универсальных параметризуемых FPGA проектов, таких как модификация устройства управления противоблестенной системой самолета Ан-140 [2, 3].

Этот факт, а также разнообразие серий, семейств в рамках серии и широкий ассортимент устройств в рамках одного семейства открывают для FPGA проектов новую грань адаптивности, учитывающую как особенности проекта, так и потенциал конкретного кристалла ПЛИС, в котором он будет реализован.

Разработка проектов, предусматривающих параметрическую настройку с учетом указанных особенностей, позволяет, с одной стороны, сделать их масштабируемыми для применения в широком диапазоне доступных аппаратных ресурсов разной номенклатуры элементной базы FPGA, а с другой – добиться наивысшей производительности конкретного проекта при реализации в заданном кристалле ПЛИС.

Целью данной статьи является анализ сложности FPGA-реализации в зависимости от разрядности, а также изложение метода разработки мультипараметризуемых проектов программируемой логики.

Для достижения поставленной цели решаются задачи анализа возможностей, предоставляемых языком VHDL для построения параметризуемых проектов (раздел 2), исследования зависимости потребления ресурсов ПЛИС, требуемых для реализации параметризуемых арифметических узлов, в том числе с фиксированной точкой (раздел 3), разработки последовательности проектирования и оценки количества ресурсов мультипараметризуемых проектов на ПЛИС (разделы 4 и 5 соответственно).

1. Аналитический обзор новых семейств кристаллов фирмы Altera

Выход новых семейств FPGA с меньшим технологическим процессом предоставляет прирост производительности и доступных аппаратных ресурсов в сравнении с предыдущими семействами в рамках серии.

Так, например, семейство Stratix 10, выполненное по 14-нм технологическому процессу (14nm FinFET process), предоставляет в 1,4-1,6 раз большую производительность, чем предыдущее семейство Stratix V при той же потребляемой мощности, и удвоенную производительность при повышении мощности в 1,3 раза [4]. При этом эквивалентная производительность предыдущего семейства может быть достигнута Stratix 10 всего при 0,3 от его потребляемой мощности.

Семейство Arria 10, производимое по более дешевому 20-нм технологическому процессу (planar TSMC process), более энергоэффективное и на 15% более производительное, нежели Stratix V (предыдущее семейство другой серии).

Новое семейство чипов Cyclone 10, представляющее самую бюджетную серию из рассматриваемых, также было обновлено и теперь обладает повышенной эффективностью, производительностью и новыми ресурсами FPGA.

Переход к новой элементной базе может потребовать переработки существующих проектов, что вызвано необходимостью задействовать дополнительное количество аппаратных ресурсов кристалла ПЛИС. При этом проект, предусматривающий широкую параметризацию, может быть перенесён на новую элементную базу с использованием всего её потенциала без перепроектирования.

2. Классификация средств обеспечения гибкости в языках описания аппаратуры

Исследование особенностей языка VHDL показывает, что гибкость проектов на этом языке может быть обеспечена тремя различными способами.

Первый и самый важный – это использование констант в секции generate внутри декларативной части, которые могут быть заданы отдельно для каждого конкретного случая использования экземпляра этого проекта. Именно этот процесс в большинстве случаев подразумевается под параметризацией.

Вторым способом является использование констант, объявленных в пакетах, что позволяет задавать константы для проекта в целом без возможности указания частных значений.

Использование констант предоставляет возможность не только обеспечить гибкость разрядности шин устройства, но и управлять реализуемой архитектурой посредством оператора generate, позволяющего автоматически формировать структуру компонента на этапе компиляции [5].

Также управление реализуемой архитектурой может быть осуществлено посредством третьего способа обеспечения гибкости проектов – механизма конфигурации, предусматривающего выбор разработчиком требуемой реализации архитектурного тела из заранее разработанного набора архитектурных тел [6].

3. Исследование влияния параметризации проектов на потребление ресурсов

3.1. Анализ влияния параметризации

Изменение параметров, отвечающих за разрядность шин арифметических блоков, приводит к изменению количества требуемых для их реализации ресурсов ПЛИС. Также это приводит к изменению точности вычислений, требования к которой могут быть различными для отдельных случаев использования параметризуемого проекта. В других же случаях акцент делается на производительность или необходимость размещения проекта в ограниченных аппаратных ресурсах заданной номенклатуры корпусов ПЛИС.

Прирост как точности, так и производительности, приводит к увеличению требуемого количества ресурсов ПЛИС, т.е. увеличение любой из этих полезных характеристик ведёт к нежелательному росту числа логических элементов и других встроенных ресурсов, занимаемых проектом.

Закон изменения количества ресурсов в зависимости от разрядности нелинеен и требует учёта специфики реализуемых арифметических узлов и возможностей заданной элементной базы, таких как число аппаратно реализованных умножителей.

Построение аналитической модели, описывающей количество требуемых ресурсов нелинейной функцией от разрядности и числа параллельных элементов системы, на этапе разработки параметризуемого проекта позволяет не только подобрать параметры для выбранного корпуса ПЛИС, но и отыскать их наиболее оптимальное сочетание в заданном допороговом диапазоне с учётом скачкообразного изменения такой функции.

Таким образом, количество ресурсов является функцией от точности и производительности.

3.2. Исследование параметризации целочисленного сумматора и умножителя

Исследование особенностей реализации целочисленного сумматора в FPGA фирмы Altera показывает, что один разряд сумматора требует для реализации один логический элемент (LE). Эта линейная зависимость экспериментально проверена для всех значений разрядности вплоть до 150. Исследование проводилось для различных кристаллов, включая Cyclone, Cyclone II, Cyclone III, Cyclone IV.

Исследование параметризации целочисленного умножителя выявляет значительно более сложную зависимость количества требуемых ресурсов от разрядности входных данных. Большой вклад в это вносит наличие в составе ПЛИС определенного числа аппаратно реализованных 18-битных умножителей, каждый из которых может быть разделен на два 9-битных умножителя. Эти умножители доступны в чипах FPGA указанных семейств. Конкретное количество встроенных умножителей зависит от семейства чипа и выбранного корпуса. Экспериментально установлено, что один 18-битный умножитель эквивалентен 400 логическим элементам.

При разрядности вплоть до 4 умножитель реализуется на логических элементах, которых требуется до 29. При дальнейшем увеличении разрядности вплоть до 9 умножитель будет имплементирован посредством одного 9-битного аппаратно реализованного умножителя, если таковой доступен. В диапазоне разрядностей от 10 до 18 уже требуется один 18-битный умножитель (табл. 1, рис. 1). Дальнейший прирост разрядности демонстрирует суще-

ственный скачок сложности реализации. Так в диапазоне от 19 до 27 уже требуется три с половиной 18-битных умножителя и от 41 до 73 логических элементов. Разрядность от 28 до 36 уже требует четыре полных 18-битных умножителя и до 109 LE (рис. 2).

В дальнейшем каждое увеличение разрядности на 18 бит сопровождается скачкообразным приростом сложности реализации умножителя, заключающемся в резком увеличении числа стандартных 18-битных умножителей, необходимых для его реализации. В диапазоне разрядностей от 37 до 54 их нужно 9, от 55 до 72 их нужно 16, от 73 до 90 – нужно 25. Для реализации умножителя, разрядность входов которого равна 91, потребуется 35 18-битных умножителей и 1066 логических элементов (табл. 2).

3.3. Операции с фиксированной точкой

Учет описанных нелинейных свойств позволяет выбрать подходящую разрядность, определяющую точность вычислений с фиксированной точкой.

Анализ особенностей вычислений с фиксированной точкой показывает, что для них могут использоваться параметризуемые целочисленные арифметические блоки. При этом длина целой и дробной частей может задаваться отдельными параметрами, позволяя тонко управлять точностью вычислений. В таком случае, для дальнейшего использования данных с выхода умножителя необходимо от его удвоенной разрядности отбросить слева длину левой части, а справа – длину правой.

Таблица 1
Результаты экспериментальной оценки ресурсов ПЛИС параметризованного умножителя

Разрядность входов	Реализация с использованием обоих ресурсов		Реализация средствами LE
	9-бит. умножители	LE	LE
1	0	1	1
2	0	4	4
3	0	17	17
4	0	29	29
5	1	0	41
6	1	0	58
7	1	0	74
8	1	0	95
9	1	0	119
10	2	0	141
11	2	0	167
12	2	0	195
13	2	0	221
14	2	0	256
15	2	0	287
16	2	0	337
17	2	0	363
18	2	0	400

Таблица 2

Результаты экспериментальной оценки зависимости ресурсов ПЛИС от разрядности умножителя

Разрядность входов	Реализация с использованием обоих ресурсов		Реализация средствами LE
	9-бит. умножители	LE	LE
19	7	41	478
36	8	109	1709
37	17	189	1843
54	18	291	3891
55	31	409	4080
72	32	545	6945
73	49	701	7293
81	49	781	8967
82	50	791	9189
90	50	871	10871
91	70	1066	11349

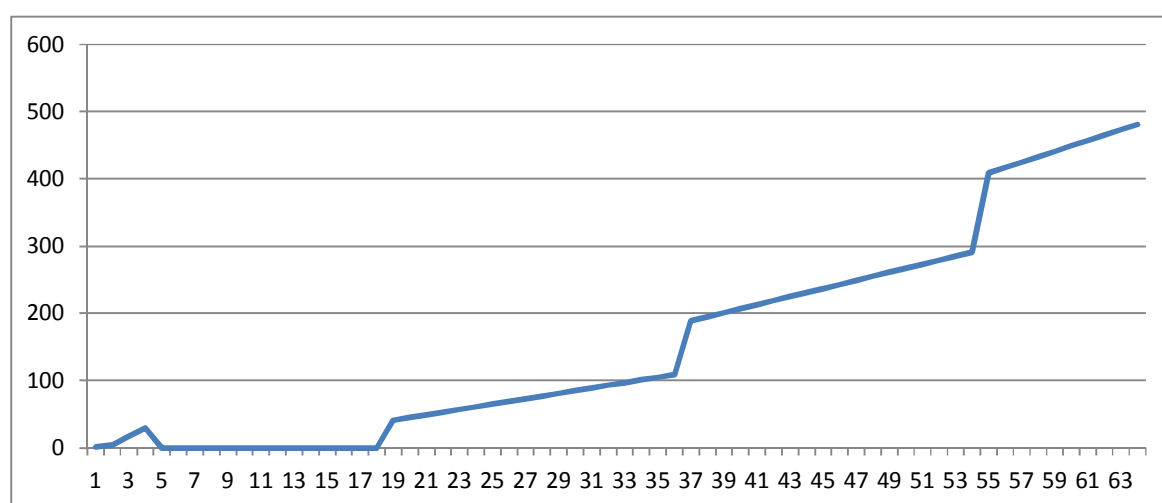


Рис. 1. Зависимость количества необходимых логических элементов (по вертикали) от разрядности входных данных синтезируемого умножителя (по горизонтали)

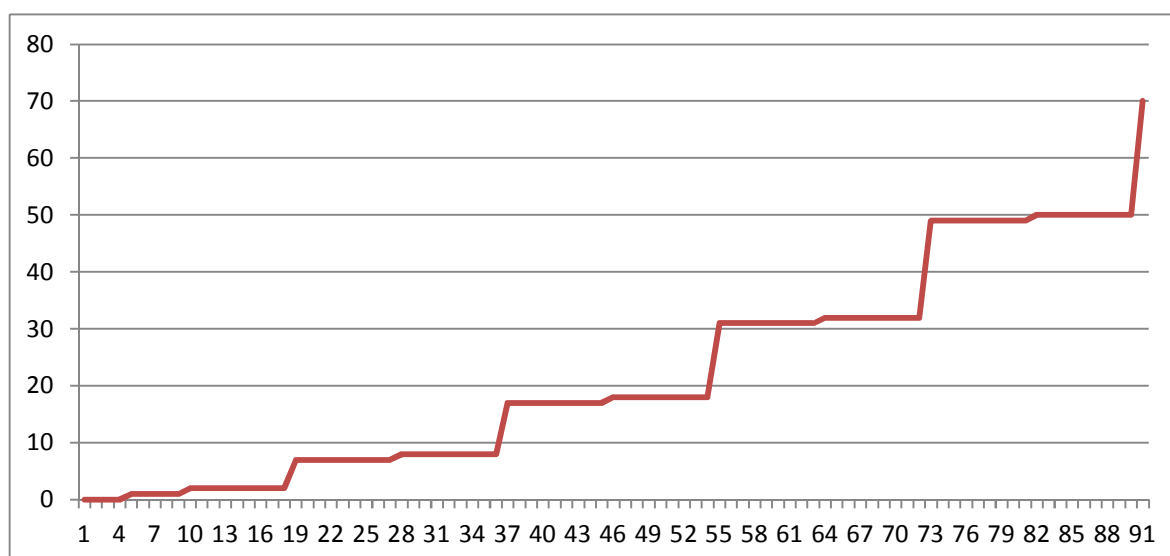


Рис. 2. Зависимость количества необходимых встроенных умножителей (по вертикали) от разрядности входных данных синтезируемого умножителя (по горизонтали)

4. Метод разработки мультипараметризуемых проектов на ПЛИС

4.1. Понятие мультипараметризации

Описанные средства языка VHDL предоставляют возможность реализации как статической, так и динамической параметризации проектов ПЛИС.

Под статической параметризацией понимается обеспечение возможности модификации проекта на ПЛИС без перепроектирования на этапе компиляции, а под динамической – во время работы устройства.

Сквозная параметризация предполагает обеспечение возможности параметризации системы на всех уровнях декомпозиции. Т.к. большинство арифметических операций, а также более сложных проектов, могут быть описаны как параллельным, так и последовательным образом, существует возможность предоставления параметрического выбора параллельного или последовательного стиля описания.

Мультипараметризация подразумевает обеспечение возможности выбора архитектуры, параллельной или последовательной реализации компонентов проекта посредством сквозной параметризации (с предоставлением необходимых параметров на все уровни декомпозиции проекта).

Использование параметрического выбора между двух описаний в одном проекте может быть полезным при использовании одного и того же IP-ядра для реализации основных вычислений, требующих высокой производительности, так и вспомогательного однократного расчёта констант, который может быть выполнен однократно за время работы устройства. Во втором случае может использоваться последовательный способ описания, позволяющий в разы понизить количество требуемых аппаратных ресурсов за счёт увеличения времени вычислений.

4.2. Последовательность разработки мультипараметризуемых проектов ПЛИС

Возможности параметризации, предоставляемые языком VHDL, позволяют создавать универсальные проекты, параметрическая настройка которых может обеспечить соответствие проекта требованиям производительности, заданного уровня точности вычислений или количества используемых аппаратных ресурсов ПЛИС.

Предлагаемый метод разработки широко параметризуемых проектов на языке VHDL включает в себя следующие этапы:

- отыскание граничных значений разрядности обрабатываемых данных и задание этого значения (разрядности) в качестве параметра;

- отыскание частей алгоритма, которые могут быть распараллелены. Даже если алгоритм имеет

зависимость по данным между реализуемыми итерациями, то распараллеливание может быть реализовано на уровне арифметических модулей, предусматривающих как параллельный, так и последовательный стиль описания;

- задание количества параллельных элементов (каналов) в качестве независимых параметров, определяющих формирование внутренней структуры реализуемых модулей и указание способа работы арифметических операций (параллельного или последовательного с тактированием) путём использования операторов generate;

- предоставление всего набора параметров от используемых компонентов данного уровня компонентам более высокого уровня иерархии для организации сквозной параметризации.

Следование приведенным пунктам последовательности позволяет создать проект с параметризацией как разрядностей шин, так и управлением распараллеливания внутренней структуры, а также передачей всех параметров в компонент проекта верхнего уровня иерархии.

4.3. Последовательность оценки количества требуемых ресурсов ПЛИС для реализации мультипараметризуемого проекта

Для многих реализаций, разработанных в соответствии с приведённой последовательностью создания мультипараметризуемых проектов, может быть применена следующая последовательность оценки использования количества необходимых ресурсов ПЛИС для требуемого значения параметров широко параметризуемых решений:

- построение модели разрабатываемого проекта, отражающей внутреннюю структуру проекта с учётом участков, допускающих распараллеливание;

- аналитическая и экспериментальная оценка зависимости количества аппаратных ресурсов FPGA, необходимых для реализации отдельных элементов проекта, от значений конкретных параметров, влияющих именно на эти части;

- построение формулы зависимости количества ресурсов от значения параметров в допустимом диапазоне на основании построенной модели параметризуемой структуры проекта и найденной зависимости количества требуемых ресурсов при параметризации его составных частей.

Применение описанных последовательностей позволяет разрабатывать широко параметризуемые проекты, которые могут быть применены в широком диапазоне номенклатуры элементной базы ПЛИС. В этом случае производительность системы определяется заданной точностью вычислений (разрядностью) и доступным количеством аппаратных ресурсов конкретного корпуса ПЛИС.

Выводы

В рамках исследовательской работы были достигнуты следующие результаты:

- предложена классификация средств обеспечения гибкости проектов, предоставляемых языком описания аппаратуры VHDL;

- проведен анализ зависимости ресурсов ПЛИС, требуемых для реализации арифметических узлов, от разрядности и изучены особенности реализации на ПЛИС арифметических блоков, осуществляющих вычисления с фиксированной точкой;

- получены аналитические соотношения количества логических элементов ПЛИС фирмы Altera от разрядности реализуемых целочисленных сумматоров и умножителей;

- сформирована аналитическая запись соотношения, позволяющего определить теоретически количество ресурсов ПЛИС, необходимых для реализации целочисленного сумматора и умножителя заданной разрядности;

- показано, что при наличии аппаратно реализованных умножителей в составе ПЛИС рост количества требуемых ресурсов для реализации целочисленного умножителя при увеличении его разрядности происходит скачкообразно;

- на основании анализа экспериментальных данных сделан вывод, что наивысшая эффективность реализации целочисленного умножителя в ПЛИС фирмы Altera достигается при выборе значений разрядности, кратных 18, а самая низкая эффективность – при увеличении указанных значений разрядности на единицу;

- предложена последовательность построения мультипараметризуемого проекта на ПЛИС;

- предложена последовательность оценки количества требуемых ресурсов ПЛИС для реализации мультипараметризуемого проекта.

Учет полученных результатов дает возможность повысить эффективность использования аппаратных ресурсов FPGA при реализации компонентов, содержащих целочисленные умножители и умножители с фиксированной точкой, за счет выбора допороговой разрядности.

Также на основе проведенных исследований:

- разработан параметризуемый библиотечный модуль, реализующий целочисленный сумматор с произвольным числом входных значений и наименьшей возможной внутренней задержкой;

- реализован практический пример применения предложенного метода разработки мультипараметризуемых проектов на ПЛИС, включающий реализацию арифметического блока для операций умножения с накоплением промежуточных сумм чисел с фиксированной точкой.

Литература

1. Verilog & System Verilog. Проектирование и верификация цифровых систем на кристаллах [Текст] / В. И. Хаханов, И. В. Хаханова, Е. И. Литвинова, О. А. Гузь. – Харьков : ХНУРЭ. – 2010. – 526 с.

2. Перепелицын, А. Е. Использование параметризуемых IP инфраструктур для разработки встроенных отказоустойчивых систем на ПЛИС [Текст] / А. Е. Перепелицын // *Радиоелектронні і комп'ютерні системи*. – 2016. – № 5 (79). – С. 104–112.

3. Dependable SoPC-based On-board Ice Protection System: from Research Project to Implementation [Text] / Y. Prokhorova, V. Kharchenko, S. Ostroumov, S. Yatsenko, M. Sidorenko, B. Ostroumov. // *Proceedings of International Conference on Dependability of Computer Systems (DepCoS-RELCOMEX 2008)*, Szklarska Poreba, Poland, June 26-28, 2008. – P. 135-142.

4. Intel FPGA Product Catalog v16.1 - Mouser Electronic [Electronic resource]. – Access mode: <https://www.mouser.com> – 5.02.2018.

5. Kulanov, V. Parameterized IP Infrastructures for Fault-Tolerant FPGA-Based Systems: Development, Assessment, Case-Study [Text] / V. Kulanov, V. Kharchenko, A. Perepelitsyn // *Proceedings of IEEE East-West Design & Test Symposium*. – 2009. – P. 322–325.

6. Грушвицкий, Р. И. Проектирование систем на микросхемах программируемой логики [Текст] / Р. И. Грушвицкий, А. Х. Мурцаев, Е. П. Угрюмов. – СПб. : БХВ-Петербург, 2002. – 608 с.

References

1. Hahanov, V. I., Hahanova, I. V., Litvinova, Y. I., Guz, O. A. *Verilog & System Verilog. Proektirovanie i verifikacija cifrovyyh sistem na kristal-lah* [Verilog & System Verilog. Design and verification for SoC]. Kharkov, KhNURE Publ., 2010. 526 p.

2. Perepelitsyn, A. E. Ispol'zovaniye parametriziruyemykh IP infrastruktur dlya razrabotki vstroyennykh otkazoustoychivykh sistem na PLIS [Usage of parametrizable IP infrastructures for FPGA-based fault-tolerant onboard systems development]. *Radioelektronni i komp'uterni sistemi - Radioelectronic and computer systems*, 2016, vol. 5, pp. 104–112.

3. Prokhorova, Y., Kharchenko, V. Ostroumov, S., Yatsenko, S., Sidorenko, M., Ostroumov, B. Dependable SoPC-based On-board Ice Protection System: from Research Project to Implementation. *Proceedings of International Conference on Dependability of Computer Systems (DepCoS-RELCOMEX 2008)*, Szklarska Poreba, Poland, June 26-28, 2008, pp. 135-142.

4. Intel FPGA Product Catalog v16.1 - Mouser Electronics. Available at: <https://www.mouser.com> (accessed 5.02.2018).

5. Kulanov, V., Kharchenko, V., Perepelitsyn, A. Parameterized IP Infrastructures for Fault-Tolerant FPGA-Based Systems: Development, Assessment, Case-Study. *Proceedings of IEEE East-West Design & Test Symposium*, 2009, pp. 322–325.

6. Grushvitsky, R. I., Mursaev, A. H. Ugrumov, E. P. *Proektirovanie sistem na mikroshemah programiruemoj logiki* [System designing on programmable logic chips]. Saint-Petersburg, BKHV-Peterburg Publ., 2002. 608 p.

Поступила в редакцию 15.02.2018, рассмотрена на редколлегии 4.04.2018

МЕТОД РОЗРОБКИ МУЛЬТИПАРАМЕТРИЗОВНИХ ПРОЕКТІВ ПРОГРАМОВАНОЇ ЛОГІКИ

А. Є. Перепелицын

Пропонується класифікація засобів забезпечення гнучкості проектів в межах мови опису VHDL. Наведено результати аналізу залежності ресурсів ПЛІС, необхідних для реалізації арифметичних вузлів, від їх розрядності. Аналізуються особливості реалізації на ПЛІС арифметичних операцій з фіксованою точкою. Даються аналітичні співвідношення кількості логічних елементів ПЛІС фірми Altera від розрядності цільових суматорів. Наведено результати експериментального дослідження залежності кількості необхідних апаратних ресурсів ПЛІС від довжини вхідних даних арифметичних блоків. Демонструється, що при наявності вбудованих помножувачів у складі ПЛІС збільшення кількості необхідних ресурсів для реалізації помножувача при збільшенні його розрядності призводить до стрибкоподібного збільшення використання ресурсів. Даються визначення параметризації та мультипараметризації FPGA проектів. Обговорюється шлях побудови широко параметризованих проектів на ПЛІС. Рекомендується застосування параметризації для досягнення максимальної ефективності використання ресурсів FPGA. Пропонується послідовність розробки мультипараметризованих проектів на ПЛІС і послідовність кількісного оцінювання необхідних ресурсів ПЛІС для реалізації мультипараметризованого проекту. Наводиться практичний приклад застосування описаного методу розробки мультипараметризованих проектів на ПЛІС, що включає реалізацію операцій множення з накопиченням.

Ключові слова: FPGA, ПЛІС, параметризація, мультипараметризація, ресурси ПЛІС.

METHOD OF MULTIPARAMETRIZED FPGA-BASED PROJECTS DEVELOPMENT

A. E. Perepelitsyn

A classification of project flexibility ways provided in VHDL language is proposed. The results of the analysis of the dependence of the FPGA resources required for the implementation of arithmetic blocks are presented. The peculiarities of implementation of FPGA arithmetic operations with a fixed point are analyzed. The analytical ratios of the of logical elements number for the Altera FPGA from the input data width of the arithmetic blocks are given. The results of an experimental study of the dependence of required FPGA resources amount for parametrizable arithmetic blocks implementation are given. It is demonstrated that in case if build-in hardware-implemented multipliers are available, the required FPGA resources amount for integer multiplier implementation changes with abrupt shape with adding one additional bit to input data width. Definitions of static parametrization, through parameterization and multiparametrization of FPGA projects are given. The ways of wide-parameterizable FPGA projects development are discussed. It is recommended to use parameterization to maximize the efficiency of FPGA resources usage. The technique of multiparametrized FPGA-based projects development is proposed. The technique of evaluating of the required FPGA resources amount for implementation of a multiparametrized project is proposed. A practical example of the use of the described method of multiparametrized FPGA-based projects development is proposed. The example includes the FPGA implementation of multiply-accumulate operations.

Keywords: FPGA, PLD, parameterization, multiparametrization, FPGA resources.

Перепелицын Артём Евгеньевич – старший преподаватель кафедры компьютерных систем, сетей и кибербезопасности, Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт», Харьков, Украина, e-mail: a.perepelitsyn@khai.edu.

Perepelitsyn Artem Evgenievich – senior lecturer, Computer Systems, Networks and Cybersecurity Department, National Aerospace University «Kharkov Aviation Institute», Kharkov, Ukraine, e-mail: a.perepelitsyn@khai.edu.