

Исследование и разработка алгоритма формирования тестовой последовательности для оценки энергопотребления RTL-модели процессора

А. В. Гаращенко, Л. Г. Гагарина

Национальный исследовательский университет «Московский институт электронной техники» (МИЭТ)

Аннотация. В статье рассмотрена задача формирования тестовой последовательности, обеспечивающей максимальную переключающую активность для всех блоков процессора. Предложена математическая модель максимизации целевой функции переключающей активности с использованием генетического алгоритма, для параллельного запуска которого рассмотрена модифицированная архитектура островной модели на основе клеточного автомата. Теоретически обоснована реализация операторов скрещивания, мутации и миграции. При помощи алгоритма сформирована тестовая последовательность для разрабатываемого VLIW DSP процессора с RISC архитектурой.

Ключевые слова: верификация вычислительных ядер, процессоры, энергопотребление, генетический алгоритм, клеточные автоматы, параллельные алгоритмы.

DOI 10.14357/20718632200309

Введение

Одной из важнейших тенденций развития современной микроэлектроники, ввиду уменьшения технологического процесса полупроводникового производства и повышения степени интеграции микросхем, является увеличение производительности вычислительных систем за счет наращивания гетерогенности. Многоядерные гетерогенные вычислительные системы на кристалле (СнК) состоят из ядер общего назначения и специализированных вычислительных ядер с разной архитектурой. Это связано, прежде всего, с известными преимуществами применения разных типов процессоров для выполнения отдельных классов задач. Также одной из основных характеристик вычислительных си-

стем наравне с производительностью является энергопотребление в разных режимах работы.

На текущем этапе развития микроэлектроники проектирование изделий происходит на уровне RTL-описания (Register-transfer level) СФ-блоков. Проектирование является многоэтапным итерационным процессом. В связи с архитектурной сложностью современных многоядерных процессоров, применяемых при разработке СнК, крайне сложно на ранних этапах проектирования теоретически оценить его энергопотребление в разных режимах работы.

Основная проблематика в области исследования энергопотребления процессора на этапе RTL-описания заключена в разработке методов и алгоритмов создания тестовых последовательностей из ISA (Instruction set architecture), обеспечи-

вающих максимальную переключающую активность для всех блоков процессора.

Вопросы исследования энергопотребления и пиковой производительности, включая разработку эффективных алгоритмов и методик генерации тестов, рассмотрены во многих научных работах [1–6]. Данные работы посвящены глубокому анализу отдельных способов генерации стрессовых последовательностей, что свидетельствует о необходимости исследований в этом направлении. От качества и эффективности работы алгоритмов генерации тестов зависят технические и экономические характеристики верификационного процесса.

Задача формирования тестовой последовательности, обеспечивающей максимальную переключающую активность для всех блоков процессора, относится к классу тех задач, сложность которых растет экспоненциально с ростом числа входных данных (количество инструкций в ISA и зависимостей между ними). Данную задачу можно свести к задаче дискретной оптимизации. Для решения таких задач целесообразно использовать эвристические и стохастические подходы. Одними из самых перспективных на сегодняшний день методов поиска экстремумов являются эволюционные, в том числе, генетические алгоритмы.

1. Математическая модель

Рассмотрим задачу формирования тестовой последовательности, обеспечивающей максимальную переключающую активность для всех блоков VLIW (very large instruction word) DSP (digital signal processor) процессора с RISC (reduced instruction set computer) архитектурой, набор инструкций которого содержит свыше одной тысячи ассемблерных мнемоник.

В качестве исходных данных для задачи выступают следующие множества:

- $P = \{p_1, p_2, \dots, p_n\}$ – множество всех возможных VLIW пакетов, содержащих векторные и скалярные инструкции, где n – количество пакетов;

- $S = \{s_1, s_2, \dots, s_m\}$ – множество тестовых последовательностей, где $s_i = [p_j, \dots, p_k]$, m – количество последовательностей.

Функция $f(p_i)$ ставит в соответствие каждому элементу из P его переключающую ак-

тивность (количество сигналов, которые переключились с 0 на 1 или с 1 на 0).

Целевая функция:

$$F(s_i) = \sum_{j=0}^n f(p_j), \text{ где } p_j \in s_i$$

Теперь сформулируем задачу: необходимо сформировать такую тестовую последовательность, которая обеспечит максимальную переключающую активность для всех блоков процессора, т.е. найти вектор s_i , при котором будет максимальным значение целевой функции F .

$$F(s_i) \rightarrow \max_{s_i \in S}$$

Рассмотрим математическую модель оптимизации (максимизации) целевой функции F с использованием генетического алгоритма.

Генетический алгоритм – это простая модель эволюции, общая схема которой приведена на Рис. 1. Каждая особь в популяции – это некоторое решение задачи, т.е. вектор s_i . Особь считается тем более приспособленной, чем лучше соответствующее решение, т.е. чем больше значение целевой функции. Тогда задача максимизации сводится к поиску наиболее приспособленной особи. Следуя этой модели, особь содержит одну хромосому (Рис. 2), состоящую из генов. Гены – это VLIW пакеты из множества P . За формирование генов, с учетом ограничений попадания скалярных и векторных инструкций в один пакет, которые можно

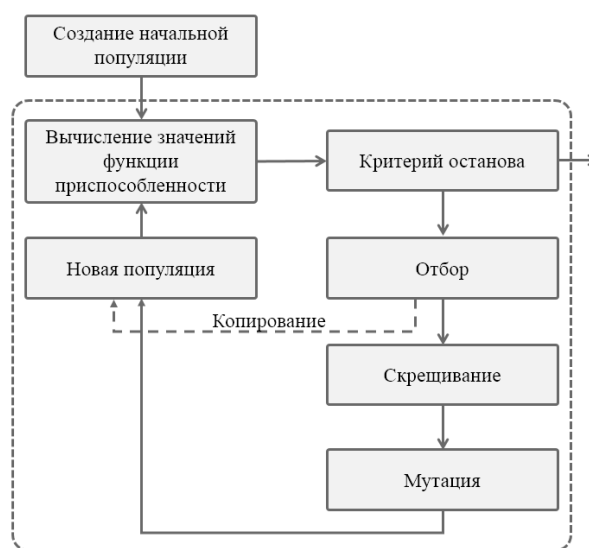


Рис. 1. Общая схема генетического алгоритма



Рис. 2. Структура особи

исполнить в рамках одного VLIW пакета, отвечает отдельный программный модуль, описанный в работах [7–9].

Первым шагом алгоритма является создание начальной популяции особей по следующему правилу: каждому гену хромосомы сопоставляется случайный VLIW пакет из множества P , таким образом формируется N особей.

Следующим шагом является вычисление значений целевой функции F для каждой особи, т.е. $F(s_i)$ для каждого вектора s_i , где $i \in [0, N-1]$.

На третьем шаге осуществляется формирование следующего поколения (Рис. 3). Из предыдущей популяции производится отбор K ($K < 0.07 * N$) более приспособленных особей — тестовых последовательностей, которые переходят без изменений. Оставшиеся места ($N - K$) в новой популяции заполняются вариантами, полученными в результате последовательного применения ниже описанных операторов скрещивания и мутации.

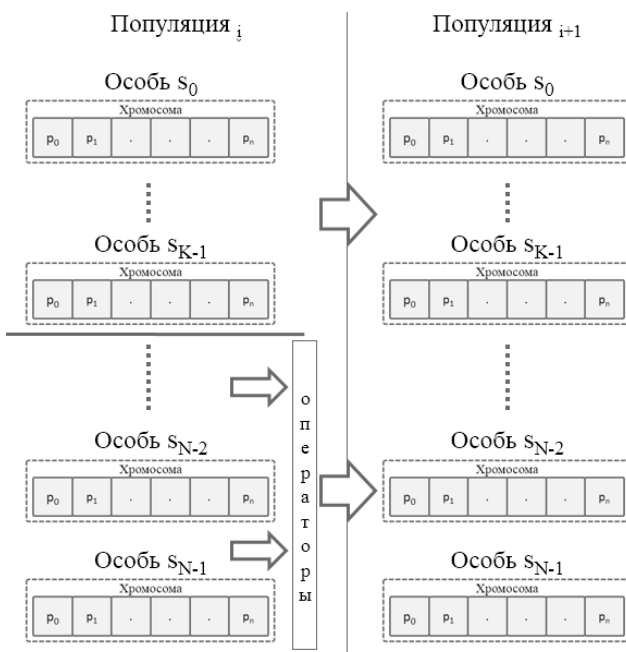


Рис. 3. Схема формирования популяции

1.1. Оператор скрещивания

С учетом распределения вероятностей из текущего поколения формируется K пар особей. Вероятность выбора вектора s_i тем выше, чем больше значение целевой функции $F(s_i)$, и вычисляется по следующей формуле (метод рулетки):

$$p_i = \frac{F(s_i)}{\sum_{j=0}^N F(s_j)}$$

Дальнейшая реализация оператора скрещивания в предлагаемом генетическом алгоритме для каждой пары выбирается случайным образом из следующих вариантов:

- двухточечный – т.к. в конвейере процессора существует зависимость по данным, как между регистрами, так и инструкциями, порядок генов (VLIW пакетов) в последовательности хромосомы является существенным. Для того, чтобы снизить вероятность разбиения таких последовательностей оператором скрещивания, в каждой паре отобранных вариантов хромосомы делятся не на две части, а на три (т.е. имеется две точки деления):

$$s_i = [p_0, \dots, p_{e-1}] + [p_e, \dots, p_k] + [p_{k+1}, \dots, p_n],$$

где $p \in s_i$,

($\forall e, k \in [1, \dots, n-1]$) ($0 < e < k < n$), n – количество пакетов.

Тогда новая пара образуется обменом средними частями, следовательно для пары (s_i, s_j) :

$$s_{i\text{new}} = s_{i,0} + s_{j,1} + s_{i,2}$$

$$s_{j\text{new}} = s_{j,0} + s_{i,1} + s_{j,2}$$

- перестановкой – т.к. множеством допустимых решений задачи формирования тестовой последовательности является множество всех перестановок из n элементов, где n – количество всех возможных VLIW пакетов. Это означает, что скрещивание, выполненное по двухточечному способу, не всегда может привести к корректным решениям. Следовательно, необходим такой вариант реализации оператора, который для любых двух перестановок создает две новые перестановки, наследующие свойства своих родителей [14]. Тогда для пары (s_i, s_j) новая пара образуется следующим образом: после выполнения по формулам выше двухточечного скрещивания для расчета $s_{i\text{new}}$ перебираем все p_m , ($\forall m \in [0, n]$) $m \notin [e, k]$, n – количество пакетов.

Производим замену:

$$o_m = u_t \forall p_m (\exists t \in [e, k]) (o_t = u_m), \\ o \in S_{inew}, u \in S_j$$

Аналогичным образом для того же сегмента $[e, k]$ рассчитывается S_{jnew} .

Для всех K пар в обоих случаях оператора скрещивания исходная пара (s_i, s_j) заменяется на вычисленные S_{inew} и S_{jnew} .

1.2. Оператор мутации

Далее у каждой особи, полученной из K пар, случайным образом выбирается g генов, где g – эвристически подобранное значение, которые подвергаются мутации одним из следующих способов:

- заменяются на новые VLIW пакеты, полученные при помощи специального программного модуля [7, 9];
- изменяется порядок следования пакетов на обратный;
- порядок следования пакетов перемещается.

1.3. Критерий останова и выбор наилучшей особи

Полученная в результате последовательного применения операторов скрещивания и мутации новая популяция замещает старую и подаётся на вход второго шага.

После чего выполняется проверка условия останова [10–13]. Если выполнено максимальное количество итераций алгоритма – эпох эволюции, алгоритм завершает свою работу. В качестве итоговой тестовой последовательности – решения задачи, выбирается особь из последнего поколения с максимальным значением целевой функции переключательной активности F .

2. Клеточный автомат

Эволюционные и, в частности, генетические алгоритмы обладают хорошим потенциалом для распараллеливания [14]. Существует множество предложенных архитектур и стратегий для их запуска на параллельных процессорах. Обобщенно стратегии можно разделить на два подхода.

Первый – распараллеливание вычисления отдельных шагов алгоритма: оценивание, отбор, скрещивание и мутация. При таком подхо-

де операторы скрещивания и мутации, кроме формирования пар, в общем случае не требуют взаимодействие между различными особями популяции и имеют высокую степень параллелизма. Единственным препятствием к эффективному распараллеливанию является реализация оператора отбора.

Второй – островные модели, суть которых заключается в том, что вся популяция разбивается на несколько независимых генофондов – островов, каждый из которых итерируется в отдельном потоке или процессоре. Для реализации такого подхода необходимо ввести оператор миграции – обмен ДНК между всеми генофондами. Существует несколько базовых стратегий миграции от одной лучшей до нескольких случайных особей.

Очевидно, что при программной реализации второй подход имеет неоспоримые преимущества, т.к. естественно отображается на архитектуру межпроцессорных связей и не требует затрат на синхронизацию потоков или ядер. Кроме того, на каждом острове возможно использовать различные вариации реализаций операторов скрещивания, мутации и отбора, что позволяет шире охватить область допустимых значений целевой функции F и снижает вероятность заикливания алгоритма в локальных экстремумах.

Для решения поставленной задачи предлагается модифицированный подход островной модели на основе клеточного автомата.

Суть подхода заключается в том, что все особи генофонда – это отдельные клетки в клеточной структуре, другими словами – узлы в прямоугольной двухмерной сетке. Особенностью является то, что все бинарные операторы, включая миграцию, применяются только к особям, расположенным в соседних клетках. В силу пространственного распределения индивидов по клеточной структуре в такой модели возможно образование устойчивых генофондов, между которыми из-за отсутствия явных границ ведется конкурентная борьба за выживание. Для увеличения эффективности модели и снижения объёма пересылаемых данных между процессорами, каждый поток или ядро – это часть сетки, т.е. кластер, состоящий из клеток.

Заключение

Описанный подход формирования тестовой последовательности, обеспечивающей максимальную переключающую активность всех блоков, был реализован для разрабатываемого VLIW DSP процессора с RISC архитектурой, набор инструкций которого содержит свыше одной тысячи ассемблерных мнемоник. В качестве критерия для расчета значений целевой функции F выбрано количество сигналов, которые переключились с 0 на 1 или с 1 на 0 в течение моделирования RTL описания процессора. После исполнения алгоритма со следующими параметрами:

- количество итераций (эпох) равно 100;
- количество особей (решений) в популяции равно 10;
- количество хромосом в особи равно 1;
- количество генов (длина тестовой последовательности) лежит в отрезке от 20 до 50;
- вероятность мутации генов равна 40%;

значение функции F для самой приспособленной особи из последнего поколения составило около 97% от общего количества сигналов. На Рис. 4 продемонстрирован график зависимости максимальной переключающей активности (значение F) от количества запусков для предложенного генетического алгоритма (GA), случайных (Random) и ручных (Manual) тестов. Переключающая активность тестовых последовательностей, полученных при помощи GA, быстрее сходится в направлении максимума при меньшем количестве итераций их генерации.

Дальнейшая модификация расчета значений целевой функции F позволила применить данный подход для максимизации переключающей активности на каждом такте.

Литература

1. Ajay M. Joshi, Lieven Eeckhout, Lizy K. John, Ciji IsenCiji Isen, Automated Microprocessor Stressmark Generation, Conference: High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium.
2. Formal Verification of Floating-Point RTL at AMD Using the ACL2 Theorem Prover David M. Russinoff.
3. Y. Kim and L. K. John, Automated di/dt stressmark generation for microprocessor power delivery networks, IEEE/ACM International Symposium on Low Power Electronics and Design, Fukuoka, 2011, pp. 253-258, doi: 10.1109/ISLPED.2011.5993645.

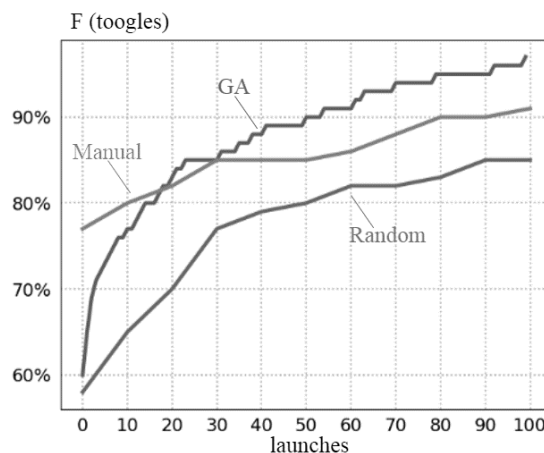


Рис. 4. Значение функции F

4. Alexandre Otto Strube, Dolores Rexachs, Emilio Luque, Software Probes: A Method for Quickly Characterizing Applications Performance on Heterogeneous Environments, Parallel Processing Workshops 2009. ICPPW '09. International Conference on, pp. 262-269, 2009.
5. Principles of Verifiable RTL Design: A functional coding style supporting verification processes in Verilog Hardcover, Springer – May 31, 2001.
6. Камкин А., Коцыняк А., Смолков С., Татарников А., Чупилко М., Сортов А., Средства функциональной верификации микропроцессоров // Труды Института системного программирования РАН, 2014, Т. 26, № 1, С. 149–206.
7. Garashchenko A., Nikolaev A., Putrya F., Sardaryan S., System of combined specialized test generators for a new generation of VLIW DSP processors with Elcore50 architecture // Problems of developing promising micro- and nanoelectronic systems. 2018. №2. С. 9–15.
8. Putrya F., The use of random program generators and random background effects in the functional verification of multicore systems on a chip // Computer-aided design of discrete systems. 2010, С. 234–241.
9. Gagarina L., Garashchenko A., Shiryayev A., Fedorov A., Dorogova E., An approach to automatic test generation for verification of microprocessor cores // Young Researchers in Electrical and Electronic Engineering (EIConRus), 2018, С. 1490–1491.
10. Shamsul Alam S., Performance Analysis of LT Codec Architecture Using Different Processor Templates // International Journal of Information Technology and Computer Science(IJITCS), 2019, №.8, С.41–48.
11. Kamath A., Automatic Verification of Microprocessor designs using Random Simulation // Computer Science, 2012, С. 1–5.
12. Lagoon V, Constraint-Based Test Generation // Cadence, 2012, С. 1.
13. Litterick M., Harnisch M., Advanced UVM Register Modeling // Cadence, 2012, С. 1.
14. Ершов Н.М., Н.Н. Попова, Естественные модели параллельных вычислений // Компьютерные исследования и моделирование. 2015, Т. 7, №3, С. 81–785.

Гарашенко Антон Витальевич. Национальный исследовательский университет «Московский институт электронной техники» (МИЭТ) г. Москва, г. Зеленоград, Россия. Аспирант. Количество печатных работ: 8. Область научный интересов: архитектура вычислительных систем и алгоритмы оптимизации. E-mail: ant.gar1@mail.ru

Гагарина Лариса Геннадьевна. Национальный исследовательский университет «Московский институт электронной техники» (МИЭТ) г. Москва, г. Зеленоград, Россия. Директор Института системной и программной инженерии и информационных технологий, д.т.н., профессор. Количество печатных работ: 125 (из них 9 монографий). Область научный интересов: информационные технологии. Email: ant.gar1@mail.ru

Research and Development of the Algorithm for generating a Test Sequence for Evaluating the Power Consumption of an RTL-Model of Processor

A. V. Garashchenko, L. G. Gagarina

National Research University of Electronic Technology (MIET)

Abstract. One of the most important trends in the development of modern microelectronics, due to a decrease in the technological process of semiconductor production and an increase in the degree of integration of microcircuits, is an increase in the performance of computer systems by increasing heterogeneity, but the energy consumption in different operating modes is a limiting factor. The article discusses the task of forming a test sequence that provides maximum switching activity for all processor units. This task belongs to the class of those problems whose complexity grows exponentially with an increasing number of input data (the number of instructions in the ISA and the dependencies between them). However, its solution can be reduced to solving the discrete optimization problem. A mathematical model is proposed for maximizing the objective function of switching activity using a genetic algorithm, for the parallel launch of which a modified architecture of the island model based on a cellular automaton is considered. The implementation of crossbreeding, mutation, and migration operators is theoretically justified. Using the algorithm, a test sequence for the developed VLIW DSP processor with RISC architecture is formed.

Keywords: verification, processors, power consumption, genetic algorithm, cellular automata, parallel algorithms.

DOI 10.14357/20718632200309

References

1. Ajay M. Joshi, Lieven Eeckhout, Lizy K. John, Ciji IsenCiji Isen, Automated Microprocessor Stressmark Generation, Conference: High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium.
2. Formal Verification of Floating-Point RTL at AMD Using the ACL2 Theorem Prover David M. Russinoff.
3. Y. Kim and L. K. John, Automated di/dt stressmark generation for microprocessor power delivery networks, IEEE/ACM International Symposium on Low Power Electronics and Design, Fukuoka, 2011, pp. 253-258, doi: 10.1109/ISLPED.2011.5993645.
4. Alexandre Otto Strube, Dolores Rexachs, Emilio Luque, Software Probes: A Method for Quickly Characterizing Applications Performance on Heterogeneous Environments, Parallel Processing Workshops 2009. ICPPW '09. International Conference on, pp. 262-269, 2009.
5. Principles of Verifiable RTL Design: A functional coding style supporting verification processes in Verilog Hardcover, Springer – May 31, 2001.
6. Kamkin A., Kotsynyak A., Smolov S., Tatarnikov A., Chupilko M., Sortov A., Means of functional verification of microprocessors // Proceedings of the Institute for System Programming of the Russian Academy of Sciences, 2014, Vol. 26, No. 1, pp. 149–206.
7. Garashchenko A., Nikolaev A., Putrya F., Sardaryan S., System of combined specialized test generators for a new generation of VLIW DSP processors with Elcore50 architecture // Problems of developing promising micro- and nanoelectronic systems. 2018. №2. С. 9–15.
8. Putrya F., The use of random program generators and random background effects in the functional verification

- of multicore systems on a chip // Computer-aided design of discrete systems. 2010, С. 234–241.
9. Gagarina L., Garashchenko A., Shiryaev A., Fedorov A. Dorogova E., An approach to automatic test generation for verification of microprocessor cores // Young Researchers in Electrical and Electronic Engineering (EIconRus), 2018, С. 1490–1491.
 10. Shamsul Alam S., Performance Analysis of LT Codec Architecture Using Different Processor Templates // International Journal of Information Technology and Computer Science(IJITCS), 2019, №.8, С.41–48.
 11. Kamath A., Automatic Verification of Microprocessor designs using Random Simulation // Computer Science, 2012, С 1–5.
 12. Lagoon V, Constraint-Based Test Generation // Cadence, 2012, С. 1.
 13. Litterick M., Harnisch M., Advanced UVM Register Modeling // Cadence, 2012, С. 1.
 14. Ershov N.M., N.N. Popova, Natural models of parallel computing // Computer Research and Modeling. 2015, Vol. 7, No. 3, pp. 781–785.

Garashchenko A. V. PhD student, National Research University of Electronic Technology (MIET), Id.1, Shokin Square, Zelenograd, Moscow, 124498, Russia. E-mail: ant.gar1@mail.ru

Gagarina L. G. National Research University "Moscow Institute of Electronic Technology" (MIET), Moscow. Moscow, Moscow Zelenograd, Russia. Director of the Institute of Systems and Software Engineering and Information Technology, PhD, Professor. E-mail: ant.gar1@mail.ru