

АНАЛИЗ ПОДХОДОВ К СИНТЕЗУ СЕТЕЙ НА КРИСТАЛЛЕ С ИСПОЛЬЗОВАНИЕМ РЕГУЛЯРНЫХ ТОПОЛОГИЙ

А.Ю. Романов

Выполнен обзор существующих способов проектирования сетей на кристалле, основанных на подходе, при котором осуществляется проекция характеристического графа задачи на заданную регулярную топологию. Охарактеризована общая задача синтеза сетей на кристалле. Топология сети может быть заранее известной (обычно это регулярная топология) или выбираться в зависимости от задачи, которая будет выполняться сетью на кристалле. Первый способ синтеза сетей на кристалле благодаря своей относительной простоте и очевидности получил большое распространение среди разработчиков и представлен во множестве реализаций, обзор которых проведен в данной статье. Показаны преимущества и недостатки данного подхода, достигнутый эффект от его применения для различных реализаций сетей на кристалле, а также предложен путь его усовершенствования, который заключается в расширении области применения решений для регулярных топологий сетей на заранее заданные нерегулярные топологии с лучшими характеристиками.

Ключевые слова: сеть на кристалле; система на кристалле; регулярная топология сети на кристалле; нерегулярная топология сети на кристалле; проектирование сетей на кристалле; синтез сетей на кристалле; характеристический граф задачи.

Введение

На фоне значительного прогресса в области производства полупроводниковых технологий в современных встроенных системах существуют тенденции к все большему применению архитектурных решений на базе систем на кристалле (СнК). Такие системы включают в себя большое количество процессоров и периферийных узлов. При этом число процессорных ядер постоянно увеличивается, поскольку однопроцессорные системы уже не справляются с задачами, требующими высокой вычислительной производительности и обмена большими потоками данных, как, например, при обработке видео- и фотоизображений, в области IP-телефонии и т. д. [1]. На передний план выходят многопроцессорные СнК [2], где стандартные способы объединения микропроцессоров с помощью высокопроизводительных шин малоэффективны [3] и вытесняются сетевыми архитектурами в виде сетей на кристалле (СтнК) [4–6].

В общем случае СтнК – это объединение IP-ядер, каждое из которых, как правило, является процессорным ядром с локальной памятью; IP-ядра связаны между собой дополнительными устройствами в виде специализированных маршрутизаторов [4]. Способ соединения маршрутизаторов определяется топологией сети.

1. Задача синтеза СтнК

Задача синтеза СтнК является многокритериальной, сложной для формулирования (из-за большого количества параметров и требований) и имеет много способов решения. Рассмотрим ее подробнее.

Исходными данными для задачи синтеза СтнК является характеристический граф задачи, вершины которого соответствуют вычислительным узлам и характеризуются законом введения пакетов в СтнК, а направленные ребра – коммуникационному процессу между двумя узлами и характеризуются минимальной и максимальной задержкой передачи, шириной канала и интенсивностью передачи [7].

Иногда задачу формулируют в виде графа отдельных подзадач, ребра которого описывают потоки данных между ними. Тогда характеристический граф задачи получают путем проекции подзадач на вычислительные узлы (на одном вычислительном узле может выполняться несколько задач) и проекцией потоков данных между подзадачами на ребра графа.

Задача синтеза заключается в выборе топологии СтнК и проекции характеристического графа задачи в соответствии с требованиями по затратам ресурсов кристалла, энергозатратам, пропускной способности, минимальной и максимальной задержке прохождения пакетов. Топологию представляют в виде нумерованного графа, вершины которого соответствуют маршрутизаторам, а ребра – соединительным линиям. Кроме топологии выбирают метод синхронизации коммуникационной подсистемы, технологию коммуникации потоков данных, метод обеспечения качества обслуживания, метод и способ маршрутизации, объем буферных элементов, структуру маршрутизаторов и т. п.

2. Обзор способов синтеза СтнК путем проекции характеристического графа задачи на регулярную топологию

Широко распространенным является подход, когда в начале синтеза выбирается одна из регулярных топологий, которая остается неизменной в ходе дальнейшей проекции характеристического графа задачи на нее. Преимуществом данного подхода является то, что такие известные регулярные топологии как torus, mesh, hypercube, star и т. д. являются достаточно исследованными, имеют предсказуемые характеристики и детерминированные алгоритмы маршрутизации [5]. Кроме того, существует большой опыт их использования в глобальных сетях, который можно применять и при проектировании СтнК, конечно, с учетом специфики аппаратной платформы ASIC и FPGA.

Примером такого подхода является алгоритм энергоэффективного построения СтнК на базе mesh топологии, предложенный в работе [8]. Характеристический граф задачи отображается на ячеистую mesh-образную топологию, состоящую из гомогенных ячеек (вычислительное ядро и маршрутизатор), с помощью алгоритма поиска по дереву альтернатив отображения узлов: на каждом шаге спуска по ветвям дерева альтернатив рассчитывается потребляемая мощность отображенных узлов, и менее оптимальные альтернативы отбрасываются. По утверждению авторов, это позволяет уменьшить энергопотребление СтнК примерно на 60 % по сравнению с произвольным отображением задачи и в 27 раз сократить время поиска по сравнению с алгоритмом полного перебора.

В работе [9] используется линейное программирование при оптимизации задержек передачи пакетов путем устранения коллизий данных на пути передачи и уменьшения взвешенного коммуникационного расстояния при отображении характеристического графа задачи на ячеистую mesh топологию. Данный подход дает увеличение пропускной способности на 17 % в сравнении с предыдущим решением, но приводит к увеличению затрат энергии на 9 %.

Характерно, что уже в следующей работе [10] авторами предпринята попытка компенсировать недостатки ячеистой топологии путем реализации возможности добавления регионов, занимающих размеры нескольких ячеек, а также предварительного фиксирования размещения некоторых узлов (например, тех, что связаны с вводами / выводами микросхемы).

В работе [11] предложен алгоритм NMAP для оптимального отображения характеристического графа задачи на mesh топологию с учетом требований к пропускной способности. Согласно алгоритму, вначале наиболее связный узел отображается в центр сети на маршрутизатор, имеющий наибольшее количество соседей, и далее, в зависимости от степени связности с уже размещенными на свободные маршрутизаторы узлами, последовательно отображаются другие узлы с целью минимизации стоимости коммуникации между ними. Полученный результат улучшают путем попарных перестановок узлов с целью достижения минимального среднего расстояния между узлами. Потоки данных, которым пропускной способности линий связи недостаточно, разбивают на несколько потоков и в таблице маршрутизации задают для них различные альтернативные кратчайшие пути. Данный алгоритм был применен при синтезе СтнК для видеоприложений с применением библиотеки XPipes [12] и, по утверждению авторов, это позволило уменьшить расходы коммуникационных ресурсов на 32 % и увеличить пропускную способность на 53 %. Этот подход был развит в работе [13] за счет планирования оптимального размещения компонентов на кристалле путем линейного программирования и введения возможности выбора отображения графа задачи на различные регулярные топологии из библиотеки XPipes. А в работе [14] предложено специализированное программное обеспечение SUNMAP для отображения графа задачи на заданную топологию, входящую в среду генерации СтнК NetChip [15].

Алгоритм МОСА [16], по утверждению авторов, дает близкие результаты по сравнению с предыдущим подходом, но за меньшее время. Данный алгоритм используется для разделения графа задачи на две равные части в виде искусственного дерева, которые затем легко отображаются на ячеистую топологию. Дополнительно выполняется оптимизация пропускной способности соединительных линий и задержек передачи данных. По словам авторов алгоритма, это привело к уменьшению потребления энергии на 14 % по сравнению с отображением, полученным путем линейного программирования для задач обработки видео- и аудиоданных.

В работе [17] используется метод линейного программирования для поиска оптимальной проекции графа задачи на ячеистую топологию с сокращением среднего расстояния между узлами и уменьшением максимальной загрузки линий связи. Проекция выполняется в два этапа – сначала отображаются узлы, а затем соединения между ними. Это позволило в $10\sim 10^3$ раз ускорить решение задачи линейного программирования. Результатом моделирования различных видеоприложений стало уменьшение энергопотребления и требований к пропускной способности СтнК соответственно на 55,5 и 49,2 % по сравнению с эвристическими алгоритмами.

В работе [18] на базе Aethereal СтнК предложено решение, которое может быть применено для произвольной топологии СтнК, и учитывает распределение потоков данных с разным качеством обслуживания. При этом используется алгоритм UMARS+, согласно которому при выборе топологии СтнК на нее отображаются потоки данных с гарантированным сервисом минимальной задержки, обнаруживаются запрещенные повороты [19] для избегания блокировок и, с учетом этого, отображаются потоки с сервисом гарантированной пропускной способности при формировании таблиц маршрутизации и назначения тайм-слотов портам маршрутизаторов. На примере MPEG-декодера показано, что использование информации о потоках данных с различными сервисами обслуживания и их пространственно-временное распределение при отображении графа задачи на топологию СтнК дает выигрыш в ресурсах на 33 % и в энергопотреблении – на 35 %, а также позволяет уменьшить в четыре раза максимальную задержку прохождения пакетов по сравнению с подходом без учета информации о качестве обслуживания.

В работе [20] предложено проводить генерацию вариантов отображения графа задачи на mesh-образную топологию с помощью генетических алгоритмов. Поведенческая модель полученных отображений затем подвергается дискретно-событийному моделированию для оценки выигрыша в производительности и энергопотреблении СтнК и их сравнения с результатами других отображений и выбора наилучшего.

Двухэтапный генетический алгоритм для отображения графа задач на СтнК с топологией mesh используется в работе [21]. Алгоритм учитывает, что вычислительные узлы являются неоднородными и разбиты на группы с разной производительностью. Каждое возможное отображение графа задач выражается в виде хромосомы, где гены соответствуют вершинам графа, а их значения – узлам СтнК. На первом шаге алгоритма для расчета функции пригодности хромосом используется упрощенная формула, по которой задержка передачи между вершинами графа задач напрямую зависит от среднего минимального расстояния между узлами для mesh топологии. Лучшие особи популяции подвергаются мутациям и перестановкам. Наиболее оптимальные экземпляры становятся исходными данными для второго шага генетического алгоритма, который отличается от первого только более точной формулой расчета задержки передачи данных и зависит непосредственно от минимального расстояния между узлами и уточняет размещение вершин графа задач на узлы СтнК соответствующего типа.

Аналогичное представление хромосом используется в работе [22], где параметром оптимальности экземпляра является рабочая температура кристалла, оценка которой выполняется с помощью утилиты HotSpot [23]. Значения энергопотребления каждого вычислительного узла двумерной mesh сети, необходимые для утилиты HotSpot, оцениваются с помощью синтеза в среде Synopsys Power Compiler. Для генерации тестовых последовательностей и моделирования данного подхода на примере создания декодера кода с низкой плотностью проверок на четность используется симулятор NoCSim [24]. Развитием данного подхода является работа [25], где проекция выполняется на 3D mesh, а также работа [26], где, кроме параметра оптимальности по энергопотреблению, добавлен параметр минимизации занимаемой площади кристалла. Внимания также заслуживает работа [27], где рассмотрены вопросы энергоэффективного отображения задач на уже заданную архитектуру размещения узлов СтнК.

В работе [28] процесс синтеза энергоэффективной СтнК состоит из таких этапов, как отображение графа подзадач на характеристический граф задачи, отображение процессорных элементов на ячейки 2D mesh-подобной топологии, отображение маршрутной информации на соединительные ресурсы, распределение приоритетов между задачами и установка частоты функционирования линий связи в зависимости от их загрузки по алгоритму DVS [29]. Первые три этапа выполняются с помощью генетических алгоритмов. Данный подход дает возможность на 39 % уменьшить энергопотребление по сравнению с алгоритмом, который основан на использовании случайного отображения задачи на СтнК и XY-маршрутизацию.

Выводы

Таким образом, анализ различных способов синтеза СтнК путем проекции графа задачи на предварительно заданную регулярную топологию показал, что все способы оптимизации заключаются в поиске такого размещения вычислительных задач на узлах топологии, чтобы минимизировать потоки данных и расстояния между узлами с наиболее интенсивным обменом данными. Достигается это путем применения различных подходов, основанных на решении задачи линейного программирования, генетических алгоритмов и различных инструментов по оценке распределения энергопотребления и других характеристик эффективности СтнК.

В большинстве случаев в качестве predetermined топологии СтнК применяется ячеистая топология mesh, которой свойственны простота организации и использование детерминированных алгоритмов маршрутизации, что является ее преимуществом, что в то же время определяет и недостатки – наличие ограничений, связанных с неоптимальной формой топологии, особенно в случае использования mesh топологии прямоугольной формы. По этой причине подход к синтезу СтнК путем проекции графа задачи на предварительно заданную регулярную топологию в основном используется, когда СтнК имеет гомогенную структуру с однородными узлами и равномерным распределением потоков данных, и даже в этом случае актуальным является поиск нерегулярных топологических решений СтнК с лучшими характеристиками. К тому же, большинство рассмотренных подходов к синтезу СтнК достаточно универсальны и применимы для синтеза СтнК на основе нерегулярных топологий, что позволит использовать преимущества детерминированного подхода, но с применением более эффективных топологий СтнК, оптимизированных под конкретную задачу.

Литература/References

1. Meenderinck C., Azevedo A., Juurlink B., Messa M.A., Ramires A. Parallel Scalability of Video Decoders. *Journal of Signal Processing Systems*, 2009, vol. 57, no. 2, pp. 173–194.
2. Ishebabi H., Mahr P., Bobda C., Gebser M., Shaub T. Answer Set Versus Integer Linear Programming for Automatic Synthesis of Multiprocessor Systems from Real-Time Parallel Programs. *International Journal of Reconfigurable Computing*, 2009. Available at: <http://dx.doi.org/10.1155/2009/863630/>.
3. Angiolini F., Meloni P., Carta S.M., Raffo L., Benini L. A Layout-Aware Analysis of Networks-on-Chip and Traditional Interconnects for MPSoCs. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 2007, vol. 26, no. 3, pp. 421–434.
4. Axel J., Hannu T. *Networks on Chip*. Dordrecht: Kluwer Academic Publishers, 2003. 303 p.
5. Benini L., Bertozzi D. Network-on-Chip Architectures and Design Methods. *IEE Proceedings Computers & Digital Techniques*, 2005, vol. 152, no. 2, pp. 261-272.
6. Fan D., Yuan N., Zhang J., Zhou Y., Lin W., Song F.L., Ye X., Huang H., Yu L., Long G., Zhang H., Liu L. Godson-T: An Efficient Many-Core Architecture for Parallel Program Executions. *Journal of Computer Science and Technology*, 2009, vol. 24, no. 6, pp. 1061–1073.
7. Marculescu R., Ogras U. Outstanding Research Problems in NoC Design: System, Microarchitecture, and Circuit Perspectives. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 2009, vol. 28, no. 1, pp. 3–21.
8. Jingcao H., Marculescu R. Energy-aware Mapping for Tile-based NoC Architectures under Performance Constraints. *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC 2003)*, 2003, pp. 233–239.
9. Chou C., Marculescu R. Contention-Aware Application Mapping for Network-on-Chip Commu-

nication Architectures. *IEEE International Conference on Computer Design (ICCD 2008)*, 2008, pp. 164–169.

10. Hu J., Marculescu R. Energy- and Performance-Aware Mapping for Regular NoC Architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2005, vol. 24, no. 4, pp. 551–562.

11. Murali S., De Micheli G. Bandwidth-constrained Mapping of Cores onto NoC Architectures. *Proceedings of the Conference on Design, Automation and Test in Europe (DATE'04)*, Paris, 2004, vol. 2, pp. 16–20.

12. Bertozzi D., Benini L. Xpipes: A Network-on-Chip Architecture for Gigascale Systems-on-Chip. *IEEE Circuits and Systems Magazine*, 2004, vol. 4, no. 2, pp. 18–31.

13. Murali S., Benini L., De Micheli G. Mapping and Physical Planning of Networks-on-Chip Architectures with Quality-of-Service Guarantees. *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC 2005)*, 2005, vol. 1, pp. 27–32.

14. Murali S., De Micheli G. SUNMAP: A Tool for Automatic Topology Selection and Generation for NoCs. 41st Conference on Design Automation (DAC'04), 2004, pp. 914–919.

15. Murali S. *Methodologies for Reliable and Efficient Design of Networks on chips: Ph.D. dissertation*. Stanford University, 2007. 272 p.

16. Srinivasan K., Chatha K.S. A technique for Low Energy Mapping and Routing in Network-on-Chip Architectures. *Proceedings of the 2005 International Symposium on Low Power Electronics and Design (ISLPED'05)*, 2005, pp. 387–392.

17. Rhee C., Jeong H.Y., Ha S. Many-to-Many Core-Switch Mapping in 2-D Mesh NoC Architectures. *IEEE International Conference on Computer Design: VLSI in Computers and Processors, 2004 (ICCD 2004). Proceedings*, 2004, pp. 438–443.

18. Hansson A., Goossens K., Radulescu A. A Unified Approach to Mapping and Routing on a Network-on-Chip for Both Best-Effort and Guaranteed Service Traffic. *VLSI Design*, 2007, pp. 1–16.

19. Starobinski D., Karpovsky M., Zakrevski L.A. Application of Network Calculus to General Topologies Using Turn-Prohibition. *IEEE/ACM Transactions on Networking (TON)*, 2003, vol. 11, no. 3, pp. 411–421.

20. Ascia G., Catania V., Palesi M. Multi-Objective Mapping for Mesh-Based NoC Architectures. *Proceedings of the 2nd IEEE/ACM/IFIP International Conference on Hardware/software Codesign and System Synthesis (CODES+ISSS'04)*, 2004, pp. 182–187.

21. Lei T., Kumar S. A Two-step Genetic Algorithm for Mapping Task Graphs to a Network on Chip Architecture. *Euromicro Symposium on Digital System Design. Proceedings*, 2003, pp. 180–187.

22. Hung W., Addo-Quaye C., Theocharides T., Xie Y., Vijakrishan N., Irwin M.J. Thermal-Aware IP Virtualization and Placement for Networks-on-Chip Architecture. *IEEE International Conference on Computer Design: VLSI in Computers and Processors, 2004 (ICCD 2004). Proceedings*. 2004. pp. 430–437.

23. Skadron K., Stan M.R., Huang W. Temperature-aware Microarchitecture. *Proceedings of the 30th Annual International Symposium on Computer Architecture (ISCA'03)*, 2003, vol. 31, no. 2, pp. 2–13.

24. Whelihan D. The NOCsim Simulator Users Guide: Version 2.0. Pittsburgh, CMU, 2003. 51 p.

25. Addo-Quaye C. Thermal-aware Mapping and Placement for 3-D NoC Designs. *IEEE International SOC Conference. Proceedings*. 2005. pp. 25–28.

26. Hung W., Xie Y., Vijaykrishnan N. Thermal-aware Floorplanning Using Genetic Algorithms. *Sixth International Symposium on Quality of Electronic Design (ISQED 2005)*, 2005, pp. 634–639.

27. Xie Y., Hung W. Temperature-Aware Task Allocation and Scheduling for Embedded Multiprocessor Systems-on-Chip (MPSoC) Design. *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, 2006, vol. 45, no. 3. pp. 177–189.

28. Shin D., Kim J. Power-Aware Communication Optimization for Networks-on-Chips with Voltage Scalable Links. *International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS 2004)*, 2004, pp. 170–175.

29. Schmitz M.T., Al-Hashimi B.M. Considering Power Variations of DVS Processing Elements for Energy Minimization in Distributed Systems. *Proceedings of the 14th International Symposium on Systems Synthesis (ISSS'01)*, 2001, pp. 250–255.

Романов Александр Юрьевич, ассистент кафедры информационных технологий и автоматизированных систем, Московский институт электроники и математики, Национальный исследовательский университет «Высшая школа экономики» (г. Москва); a.romanov@hse.ru.

Поступила в редакцию 1 декабря 2014 г.

Bulletin of the South Ural State University
Series “Computer Technologies, Automatic Control, Radio Electronics”
2015, vol. 15, no. 1, pp. 133–138

THE ANALYSIS OF APPROACHES FOR THE SYNTHESIS OF NETWORKS-ON-CHIP BY USING REGULAR TOPOLOGIES

A.Yu. Romanov, Moscow Institute of Electronics and Mathematics, National Research University “Higher School of Economics”, Moscow, Russian Federation, a.romanov@hse.ru

The article gives a review of existing methods of networks-on-chip design, based on the approach, in which the projection of the characteristic tasks graph is performed on a given regular topology. The general problem of the synthesis of networks-on-chip is characterized. The network topology can be foreknown (usually a regular topology) or selected in accordance with the tasks that will be performed by the network-on-chip. The first method of synthesis of networks-on-chip is widespread among the developers due to its relative simplicity and obviousness and presented in a variety of implementations, which are reviewed in this article. The advantages and disadvantages of this approach, the effect achieved by its application to various implementations of networks-on-chip and the way of its improvement, which consists in extension of the scope of solutions for regular network topologies on the predetermined irregular topologies with better characteristics are offered.

Keywords: network-on-chip, system-on-chip, network-on-chip regular topology, network-on-chip irregular topology, networks-on-chip design, networks-on-chip synthesis, problems characteristic graph.

Received 1 December 2014