

3. **Холопов Ю. А., Чан Ван Хань, Преображенский Н. Б.** Аппаратная оптимизация бортовой распределенной системы управления подвеской автомобиля на основе ее свободных степеней информационных связей // Глобальный научный потенциал. 2014. № 11 (44). С. 113—116.

4. **Ле Ба Чунг.** Контроллер периферии с CAN-интерфейсом // Труды 56-й всероссийской научной конференции МФТИ "Актуальные проблемы фундаментальных и прикладных наук в совре-

менном информационном обществе. Москва, 25—30 ноября 2013 г.". М.: МФТИ, 2013. С. 59—60.

5. **Чан Ван Хань.** Исследование и разработка метода построения высокоавтономного регистратора параметров в цифровой системе управления // XXXIX ГАГАРИНСКИЕ ЧТЕНИЯ: научные труды Международной молодежной научной конференции в 9 т. Москва, 9—13 апреля 2013 г.". М.: МАТИ, 2013. Т. 4. С. 183—185.

Ju. A. Holopov¹, Leader Engineer, **Le Ba Chung^{1, 2}**, Graduate Student, **Nguyen Thanh Trung^{1, 2}**, Graduate student, **Tran Van Khanh^{1, 2}**, PhD,
¹Lebedev Institute of Precision Mechanics and Computer Engineering
²Moscow Institute of Physics and Technology (State University),
E-mail: chungbaumanvietnam@gmail.com

Features of Implementation of Distributed Control Systems Based on Hardware Network Solutions

The article discusses aspects of the implementation of network technology in distributed control systems. The described various approaches of hardware support for network solutions provide higher throughput capacity in the information environment by: exclusion of arbitration procedures, elimination of control information from the data packages, the organization of exchange between the CPU and peripherals based on a unified packages of state and control. The proposed technology, implemented on FPGA, allows to achieve minimal and stable delay of transfer data in on-board networks.

Keywords: distributed control system, network interfaces, switch, converter, recorder, CAN, Ethernet

References

1. **Преображенский Н. В., Ле Ба Чунг, Тран Ван Хань, Дам Trong Нам.** Некоторые особенности реализации сетевых решений в системах управления, *Международная конференция Инжиниринг & Телекоммуникации — En & T 2014, Tezisy dokladov*, Moscow/Dolgoprudnyj, 26—28 November 2014, Moscow: MFTI, 2014, pp. 217—218.
2. **Дам Trong Нам.** Комплексование MODBUS и PROFBUS в промышленных сетях, *XL GAGARINSKIE ChTENIJA. Nauchnye trudy Mezhdunarodnoj molodezhnoj nauchnoj konferencii v 9 vol.* Moscow, 7—11 April 2014, Moscow: MATI, 2014, vol. 5, pp. 218—219.
3. **Холопов Ю. А., Тран Ван Хань, Преображенский Н. В.** Аппаратная оптимизация бортовой распределенной системы управления

подвеской автомобиля на основе ее свободных степеней информационных связей, *Global'nyj nauchnyj potencial*, 2014, no. 11 (44), pp. 113—116.

4. **Ле Ба Чунг.** Контроллер периферии с CAN-интерфейсом, *Труды 56-ой всероссийской научной конференции МФТИ "Актуальные проблемы фундаментальных и прикладных наук в современном информационном обществе"*, Moscow, 25—30 November 2013. М.: МФТИ, 2013, pp. 59—60.

5. **Тран Ван Хань.** Исследование и разработка метода построения высоко-автономного регистратора параметров в цифровой системе управления, *XXXIX GAGARINSKIE ChTENIJA. Nauchnye trudy Mezhdunarodnoj molodezhnoj nauchnoj konferencii v 9 vol.* Moscow, 9—13 April 2013. Moscow, MATI, 2013, vol. 4, pp. 183—185.

УДК 004.942

А. Ю. Романов, канд. техн. наук, ст. преп., e-mail: a.romanov@hse.ru,
Национальный исследовательский университет "Высшая школа экономики"

Исследование сетей на кристалле с топологией *mesh* с помощью модели *NoCTweak*

Проанализировано влияние геометрической формы топологии и размещения "горячих точек" на эффективность сетей на кристалле с помощью модели NoCTweak на базе языка SystemC. Результаты моделирования показали, что использование топологий, наиболее близких по форме к квадратной, позволяет до 24,7 % увеличить пропускную способность сети на кристалле, а центральное размещение "горячих точек" дает возможность до 9 % уменьшить задержки прохождения пакетов и до 15,2 % — энергозатраты на передачу каждого из них при увеличении до 19,5 % пропускной способности сети.

Ключевые слова: сеть на кристалле, система на кристалле, регулярная топология сети на кристалле, проектирование сетей на кристалле, язык программирования SystemC, модель сети на кристалле, топология mesh, "горячие точки"

Одной из основных тенденций развития современных систем на кристалле (СнК) считается появление многоядерных систем, где в рамках одного кристалла интегрируются десятки вычислительных узлов, периферийных модулей и модулей памяти. Насущная необходимость в объединении различных компонентов таких СнК обуславливает широкое использование сетевых решений и применение сетей на кристалле (СтнК). СтнК представляет собой множество вычислительных узлов и вспомогательных модулей, объединенных общей подсистемой связи, состоящей из маршрутизаторов и соединений между ними. Подсистема связи СтнК, занимая значительные ресурсы кристалла, является энергозатратной [1, 2]. Всевозрастающие требования к ресурсоемкости и скорости передачи данных по сети диктуют необходимость анализа факторов, влияющих на эффективность СтнК (топология сети, структура маршрутизаторов и алгоритмы маршрутизации данных, распределение вычислительной нагрузки по сети и др.).

Так, последний фактор связан с распределением наиболее емких задач по вычислительной нагрузке и обмену данными на узлах СтнК. Это так называемые "горячие точки" (*hot spots*) — узлы, с которыми сетевой обмен наиболее интенсивен. "Горячая точка" — это управляющий или интерфейсный узел, через который поступают и извлекаются данные в сети. Поскольку в отличие от компьютерных сетей в СтнК вычислительные ядра располагаются компактно и обмен данными происходит на высоких частотах, распределение "горячих точек" имеет определяющее значение.

Как указывает В. Делли [2, 3], наиболее ценными из ресурсов при реализации СтнК являются физические каналы передачи, буферная память, а также коммутационная часть маршрутизаторов (роутеров). Поэтому важной характеристикой СтнК является ее топология, которая, будучи одним из основных факторов эффективности СтнК, в значительной степени определяет структуру роутеров, алгоритм маршрутизации и расходы соединительных ресурсов. В общем случае топология СтнК представляет собой неориентированный связный граф, состоящий из вершин (роутеров) и ребер (физических линий связи между ними), и фактически описывает, каким образом роутеры соединены между собой и сколько линий связи задействовано для этого [1]. В настоящее время наибольшее распространение получили регулярные топологии, такие как *mesh*, *torus*, *hypercube*, *spidergon*, *chordal ring*, *butterfly fat tree (BFT)* и т.п. [3—6]. Это обусловлено тем, что регулярные топологии имеют простую структуру роутеров, соединений между ними и заранее определенные алгоритмы маршрутизации.

Цель статьи путем использования высокоуровневого моделирования с применением модели *NoCTweak* на основе языка *SystemC* провести анализ влияния топологии и расположения "горячих точек" на эффективность СтнК.

Очевидным решением для анализа эффективности СтнК является реализация их в рамках кристалла по технологии *ASIC* или на *FPGA*-чипе. Но ввиду сложности реализаций СтнК и больших затрат на изготовление многоядерных чипов поиск оптимальных решений для СтнК с помощью прототипирования не целесообразен [1]. Для выполнения таких задач используют модели-симуляторы СтнК, которые позволяют осуществлять тестирование лучших архитектурных решений при построении СнК, и в настоящее время именно они являются неотъемлемой составляющей конструирования и оптимизации СтнК. Среди моделей СтнК выделяют аналитические и высокоуровневые/низкоуровневые имитационные модели. Каждый из приведенных подходов к моделированию СтнК характеризуется различным уровнем абстракции и, соответственно, имеет различную точность и затраты ресурсов на проведение моделирования [5].

Низкоуровневое моделирование — эмуляция работы сети на уровне логических вентилей. Компоненты модели разрабатывают с применением языков описания аппаратуры (например, *Verilog* или *VHDL*) или собирают из стандартных блоков в специализированных средах проектирования, например, *Altera Quartus II* или *Xilinx ISE*. При этом функционирование разрабатываемых компонентов модели устройства анализируется с помощью специализированных программ моделирования аппаратных средств (например, пакет *ModelSim*). Такая модель может быть синтезирована с помощью специализированных САПР. Недостатком данного подхода являются значительные затраты времени на построение модели и само моделирование. Его целесообразно использовать для оценки эффективности отдельных решений при построении СтнК.

Высокоуровневое моделирование — это симуляция работы сети на уровне распространения потоков данных. Этот подход характеризуется скоростью разработки, гибкостью настроек и значительно меньшим временем моделирования. Данные модели делят на синтезируемые и несинтезируемые.

Синтезируемые модели обычно разрабатывают с использованием языка *SystemC* [7, 8]. Этот язык проектирования и верификации моделей системного уровня реализован в виде *C++* библиотеки с открытым кодом. Библиотека содержит ядро событийного моделирования и позволяет получить исполняемую модель устройства. Язык *SystemC* используют для построения транзакционных и поведенческих моделей, а также для высокоуровневого синтеза устройств. В *SystemC* используются понятия, подобные тем, которые присущи языкам описания аппаратуры — *VHDL* и *Verilog* — интерфейсы, процессы, сигналы, событийность и иерархия модулей. Благодаря этому *SystemC* пригоден как для поведенческого моделирования, так и для *RTL*-синтеза (на уровне регистров).

Применение языка *SystemC* позволяет уменьшить влияние недостатков и объединить преимущества низкоуровневого и высокоуровневого подходов, чем и обеспечивается его широкая популярность при создании симуляторов работы СтнК. На основе *SystemC* создана библиотека *Xpipes* [9], что дает возможность осуществлять полный цикл моделирования и синтеза СтнК [10, 11]. В работе [12] предложена высокоуровневая *SystemC*-модель *ARTS* для сравнительного моделирования сетевого и шинного методов построения систем на кристалле. С использованием *SystemC* разработаны достаточно известные симуляторы *Noxim* [13], *NIRGAM* [14] и др.

Популярность *SystemC* обусловлена также и тем, что он базируется на языке *C/C++*, однако по своей природе *C/C++* — последовательный язык, в то время как процессы в аппаратуре происходят одновременно и параллельно. Это требует освоения новой парадигмы программирования, а также специфических инструментов, таких как процессы, события, сигналы и др. [7].

Описание модели СтнК *NoCTweak*

Одним из симуляторов, основанных на *SystemC*, является *NoCTweak* [15]. Эта модель СтнК характеризуется открытым исходным кодом и предназначена для исследования производительности и энергетической эффективности сетей на кристалле, а использование связки языков *SystemC* и *C++* в *NoCTweak* дает возможность проводить моделирование с высокой скоростью на уровне циклов. Симулятор ориентирован на моделирование процесса передачи данных в подсистеме связи СтнК с топологией *mesh* и имеет большое число настраиваемых параметров.

Открытый исходный код *NoCTweak* позволяет выполнять его модификацию, оптимизацию и настройку под конкретную прикладную задачу. Так, поскольку симулятор был разработан для функционирования в *Unix*-подобных операционных средах, он был адаптирован под *MS Windows*. Были исправлены некоторые ошибки и оптимизирован исходный код *NoCTweak*, а также разработаны *bash*-скрипты для автоматического запуска нескольких прогонов симулятора и программный код в среде *MATLAB* для сбора и анализа статистики и по-

строения сводных графиков изменения характеристик СтнК.

Посредством *NoCTweak* выполняется моделирование работы сети на кристалле, включающей в себя множество ядер, связанных между собой с помощью сети маршрутизаторов с топологией *mesh* (рис. 1). Каждый узел сети состоит из вычислительного узла (ядра и сетевых интерфейсов) и связанного с ним маршрутизатора. Каждый вычислительный узел генерирует пакеты данных и посылает их в сеть через маршрутизаторы. Пакеты перемещаются по сети согласно алгоритму маршрутизации и поглощаются при достижении пункта назначения (другого вычислительного узла).

Основным преимуществом симулятора *NoCTweak* по сравнению с аналогами является его достаточно простая структура и большое число конфигурируемых параметров, которые можно как задавать в конфигурационном файле перед началом симуляции, так и изменять непосредственно в исходном коде. К основным параметрам относят геометрические пропорции сети и ее размер. От того, какую форму имеет СтнК, зависят значения диаметра и среднего расстояния между узлами, ширина бисекции и т. д. Анализ влияния геометрической формы топологии на производительность СтнК является важной задачей проектирования.

NoCTweak поддерживает два типа режимов симуляции: встроенные тесты и синтетические тесты [15]. Встроенные тесты — это заранее подготовленные файлы конфигураций, отвечающих реальным объектам (например, *MPEG4*-декодер с 12 задачами или 802.11a *WiFi*-передатчик с 25 задачами). Для их моделирования достаточно передать файл конфигурации в программу и указать лишь несколько основных опций или оставить настройки по умолчанию. Другой вид тестов (синтетические тесты) предполагает ручную настройку всех параметров моделирования, среди которых основными являются тип трафика, размер сети, частота генерации флитов (*flit* — составная часть пакета данных), а также расположение "горячих точек". Как уже было сказано, "горячие точки" — это узлы с наиболее интенсивным сетевым трафиком; обычно на них приходится до 90 % сгенерированных и принятых пакетов в сети. Правильный выбор "горячих точек" во многом определяет эффективность проекции характеристического графа задачи на сеть и ее конечную производительность.

Влияние геометрической формы топологии на характеристики СтнК

Влияние геометрической формы топологии на характеристики СтнК было исследовано на примере сетей с 99 и 100 узлами. Обычно используют топологию *mesh* квадратной формы, но это возможно только для сетей с числом узлов, кратным степени натуральных чисел (например, 4, 9, 16, ...). В остальных случаях используют топологии прямоугольной

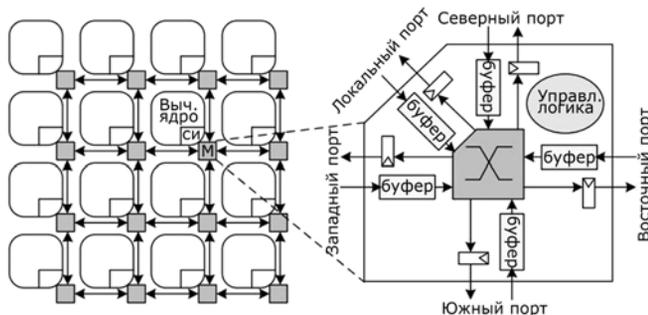


Рис. 1. Упрощенная структура подсистемы связи СтнК в *NoCTweak* [15]

формы, что приводит к увеличению диаметра и уменьшению ширины бисекции, а если число узлов представлено простым числом (например, 5, 7, 11, ...), задача усложняется еще больше, поскольку приходится отказываться от детерминистических алгоритмов маршрутизации (главного преимущества *mesh*-топологии) или брать топологию для большего числа узлов, оставляя один из маршрутизаторов как вспомогательный и не присоединенный к вычислительному узлу [16].

Для проведения моделирования были взяты две сети на кристалле с числом узлов, равным 100: одна сеть — квадратной формы (10 × 10); другая — прямоугольной (5 × 20). Кроме того, была взята СтНК с меньшим числом узлов, равным 99, и формой 9 × 11, приближенной к квадратной. Длительность прогонов модели *NoCTweak* выбрана в 30 000 циклов моделирования, время разогрева сети (*warm-up time*) — 5000 циклов. Прогоны проводили для различной частоты генерации флитов (*fir*, флит/цикл/узел) — от 0,05 до 0,3, а режим генерации пакетов был установлен как "случайный". Полученные зависимости средней задержки прохождения пакета (L_{av} , цикл) (рис. 2, а, б) и средней пропускной способности (T_{av} , флит/цикл/узел) (рис. 2, в) от частоты генерации флитов приведены ниже.

Анализ полученных результатов моделирования (рис. 2) свидетельствует о том, что СтНК с топологией прямоугольной формы имеет значительно худшие показатели задержки прохождения пакетов и пропускной способности. Так, насыщение сети 5 × 20 происходит при $fir = 0,15$, $L_{av} = 567,3$ и достигает $T_{av} = 0,121$. При этом на графиках рис. 3, а, б видно, что и при меньшем *fir* показатель задержки прохождения пакетов сети 5 × 20 в разы больше, чем у сети 10 × 10 с топологией квадратной формы. Это происходит вследствие значительно большей удаленности узлов друг от друга — диаметр сети возрос с 18 до 23 хопов (*hop* — переход между маршрутизаторами). Для сети 10 × 10 насыщение происходит при $fir = 0,17$, когда L_{av} достигает 475,6 и в дальнейшем резко возрастает. При этом максимальное значение $T_{av} = 0,16$, что на 24,4 % больше, чем у топологии с неоптимальной прямоугольной формой.

Анализ формы графиков пропускной способности при дальнейшем увеличении частоты генерации флитов показывает, что если пропускная способность топологии 5 × 20 начиная с $fir = 0,17$ стабилизируется на уровне 0,07 и остается на нем вне зависимости от *fir*, то для топологии 10 × 10 снижение является более плавным (так, для $fir = 0,3$ $T_{av} = 0,097$, что на 24,7 % больше, чем значение $T_{av} = 0,073$ для топологии 5 × 20) и только при $fir = 0,5$ достигает уровня топологии 5 × 20. Это сви-

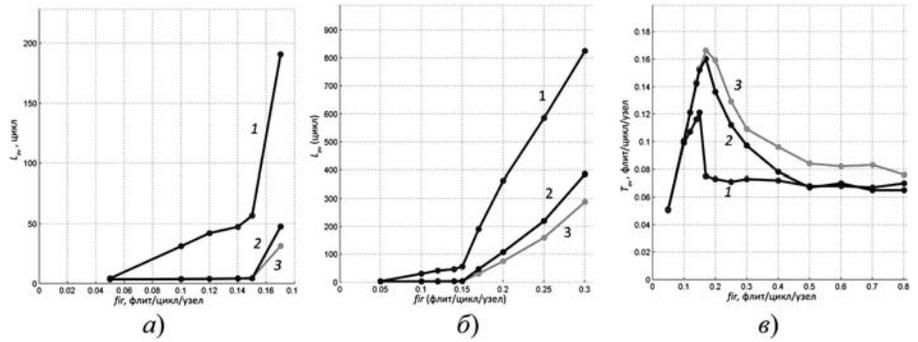


Рис. 2. Результаты моделирования СтНК топологии *mesh* в *NoCTweak*: 1 — при размерности сети 5 × 20; 2 — при размерности сети 10 × 10; 3 — при размерности сети 9 × 11

детельствует о значительно большей устойчивости подсистемы связи СтНК к повышенной нагрузке за счет применения топологии квадратной формы.

Отдельно на рис. 2 показаны графики, относящиеся к топологии 9 × 11. Их форма практически повторяет форму графиков топологии 10 × 10: пропускная способность является ненамного большей, а задержка прохождения пакетов — ненамного меньшей (при $fir = 0,17$ $L_{av} = 313,9$, а максимальная $T_{av} = 0,166$, что на 3,6 % больше, чем у топологии 10 × 10). Такие результаты объясняются тем, что СтНК с топологией 9 × 11 содержит на один узел меньше, чем СтНК с топологией 10 × 10. При этом форма топологии 9 × 11 близка к квадратной, за счет чего негативное влияние на характеристики сети не оказывается. Данные результаты согласуются с подобными, полученными при моделировании СтНК с использованием высокоуровневой модели *OCNS* в работе [17].

Влияние расположения "горячих точек" на характеристики СтНК

Как уже было отмечено, топология *mesh* является одной из наиболее распространенных в СтНК, что и стало определяющим для ее использования в модели *NoCTweak*. Собственно, это единственная топология, поддерживаемая симулятором в его основной конфигурации, хотя данное ограничение может быть снято путем расширения набора поддерживаемых топологий благодаря открытости кода симулятора. При необходимости в модель достаточно легко могут быть добавлены топологии *torus* и *hypercube*, близкие по своей структуре к топологии *mesh*.

В данном исследовании интерес представляла именно топология *mesh*. При такой топологии "горячие точки" могут располагаться или в центре сети (*central*), или по краям (*corner*). Для оценки влияния "горячих точек" на характеристики сети была проведена серия прогонов модели *NoCTweak* для сети с топологией *mesh* размером 5 × 5 (25 узлов); длительность каждого прогона моделирования составляла 100 000 циклов; время разогрева сети было установлено на 10 000 циклов. Прогоны проводились для различной частоты генерации флитов —

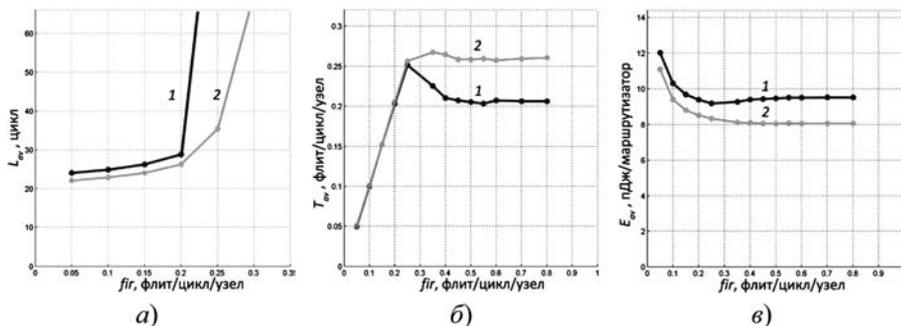


Рис. 3. Результаты моделирования СтНК для 25 узлов в *NoCTweak*. Расположение "горячих точек":
1 — по краям сети; 2 — в центре

от 0,05 до 0,8. Полученные зависимости средней задержки прохождения пакета (см. рис. 3, а), средней пропускной способности (рис. 3, б) и среднего распределения энергии на пакет (E_{av} , пДж/маршрутизатор) (рис. 3, в) от частоты генерации флитов приведены ниже.

Анализ результатов моделирования (см. рис. 3) свидетельствует о том, что при размещении "горячих точек" на периферии насыщение сети происходит при $fir = 0,2$ и $L_{av} = 28,8$ и уже при $fir = 0,25$ L_{av} возрастает в разы, достигая значения 514,0. График же средней задержки пакетов для центрального размещения "горячих точек" проходит значительно ниже. Так, при $fir = 0,2$ средняя задержка прохождения пакета $L_{av} = 26,2$ (что на 9,0 % меньше), а насыщение наступает при $fir = 0,25$ и $L_{av} = 35,3$. При дальнейшем увеличении fir происходит резкое возрастание задержки пакетов, которая при $fir = 0,35$ достигает порядка 10 000 циклов для обоих способов размещения "горячих точек". Данные наблюдения подтверждаются и графиками на рис. 3, б: сначала пропускная способность T_{av} возрастает линейно с увеличением fir , но при достижении значения 0,25 происходит насыщение, и T_{av} стабилизируется на уровне 0,257 и 0,207 (данные для $fir = 0,6$) для центрального (что на 19,5 % больше) и периферийного размещения "горячих точек" соответственно. Общую картину подтверждает и график на рис. 3, в: по достижении $fir = 0,25$ средние затраты энергии на передачу одного пакета стабилизируются и составляют 8,04 и 9,48 для центрального (что на 15,2 % меньше) и периферийного размещения "горячих точек" соответственно. Данные результаты согласуются с результатами, полученными другими авторами для трехмерных регулярных топологий [18].

Заключение

Таким образом, в результате моделирования СтНК с числом узлов, равным 100, в симуляторе *NoCTweak* была выявлена высокая чувствительность сетей с топологией *mesh* к тому, какие геометрические пропорции имеет топология сети.

Чем форма топологии ближе к квадратной, тем выше основные характеристики СтНК, что продемонстрировано на примере сетей размерности 10×10 , 9×11 и 5×20 : у первых двух топологий характеристики примерно одинаковы ввиду близости их формы к квадратной; различия в 3,6 % по пропускной способности проявляются вследствие разницы в числе их узлов на единицу. При сравнении топологий 10×10 и 5×20 различие в пропускной способности достигает 24,4...24,7 % в пользу топологии

10×10 как в момент насыщения СтНК, так в установившемся режиме.

Также с помощью моделирования СтНК с 25 узлами в симуляторе *NoCTweak* получены результаты, характеризующие влияние размещения "горячих точек" в сетях с топологией *mesh* на их производительность. Размещение задач, требующих большого объема сетевого обмена на центральных узлах, позволяет до 9 % уменьшить задержки прохождения пакетов и до 15,2 % — энергозатраты на передачу каждого пакета данных, при этом до 19,5 % увеличивая пропускную способность сети.

Перспективой дальнейших исследований является модификация симулятора *NoCTweak* для моделирования СтНК с топологиями *torus*, *hypercube* и нерегулярными топологиями для оценки влияния на них геометрической формы топологии и расположения "горячих точек".

Список литературы

1. Axel J., Hannu T. Networks on Chip. Dordrecht: Kluwer Academic Publishers, 2003. 303 p.
2. Dally W., Towles B. Principles and practices of interconnection networks. Elsevier, 2004. 550 p.
3. Dally W., Aoki H. Deadlock-free adaptive routing in Multi-computer Networks using virtual channels // IEEE Transactions on Parallel Distributed Systems. 1993. Vol. 4. N. 4. P. 466—475.
4. Balfour J., Dally W. Design Tradeoffs for Tiled CMP On-Chip Networks // ICS'06: Proceedings of the 20th Annual International Conference on Supercomputing. ACM Press, 2006. P. 187—198.
5. Saldana M., Shannon M., Chow P. The Routability of Multi-processor Network Topologies in FPGAs // Proc. of the 2006 international workshop on System-level interconnect prediction (SLIP'06). NY: ACM Press, 2006. P. 49—56.
6. Benini L., Bertozzi D. Network-on-chip architectures and design methods // Computers and Digital Techniques. IEEE. 2005. Vol. 152, N. 2. P. 261—272.
7. Романов О. Ю., Лисенко О. М. Обґрунтування вибору методу та засобів моделювання мереж на кристалі // Наукоємні технології. Збірник наукових праць. Київ: НАУ. 2014. № 1 (21). С. 49—54.
8. Genko N., Atienza D., De Micheli G., Benini L. Feature-NoC emulation: a tool and design flow for MPSoC // IEEE Circuits and Systems Magazine. 2007. Vol. 7, N. 4. P. 42—51.
9. Bertozzi D., Benini L. Xpipes: A network-on-chip architecture for gigascale systems-on-chip // IEEE Circuits and Systems Magazine. 2004. Vol. 4, N. 2. P. 18—31.
10. Murali S., De Micheli G. Bandwidth-constrained mapping of cores onto NoC architectures // Proc. of the Conference on Design,

Automation and Test in Europe, 2004 (DATE'04). Paris, 2004. Vol. 2. P. 16–20.

11. **Bertozzi D., Murali S., Jalabert A.** NoC synthesis flow for customized domain specific multiprocessor systems-on-chip // *IEEE Transactions on Parallel and Distributed Systems*. 2005. Vol. 16, N. 2. P. 113–129.

12. **Mahadevan S., Virk K., Madsen J.** ARTS: A SystemC-based framework for multi-processor Systems-on-Chip modeling // *Design Automation for Embedded Systems*. Springer, 2007. Vol. 11, N. 4. P. 285–311.

13. **Fazzino F., Palesi M., Patti D.** Noxim: Network-on-chip simulator. Режим доступа: <http://noxim.sourceforge.net/> (Доступ свободный). Загл. с экрана.

14. **Jain L.** NIRGAM: A Simulator for NoC Interconnect Routing and Application Modeling. Version 1.1. 2007. 27 p. URL: <http://nirgam.ecs.soton.ac.uk/Documentation.php>.

15. **Tran A. N.** On-Chip Network Designs for Many-Core Computational Platforms: Ph.D. thesis. USA, Davis: University of California, 2012. 156 p.

16. **Romanov O., Lysenko O.** The Comparative Analysis of the Efficiency of Regular and Pseudo-optimal Topologies of Networks-on-Chip Based on Netmaker // 1st Mediterranean Conference on Embedded Computing, MECO 2012. Advances and Challenges in Embedded Computing: Proceedings. Montenegro, Bar: IEEE, 2012. P. 13–16.

17. **Феськов Д. О., Романов О. Ю., Короткий С. В.** Программная модель сетей на кристалле с нерегулярными топологиями // Проблемы информатизации та управління. Збірник наукових праць. Київ: НАУ. 2013. № 2 (42). С. 118–123.

18. **Gebali F., Elmiligi H., Watheq M.** Networks-on-Chips: Theory and Practice. FL: CRC Press, 2009. 389 p.

A. Yu. Romanov, Senior Lecturer, e-mail: a.romanov@hse.ru,
National Research University Higher School of Economics, Moscow

The Research of Network-on-Chip with Mesh Topology by Using NoCTweak Model

This article gives an analysis of the effect of geometric shape of the topology and "hot spots" placement on the effectiveness of networks-on-chip. For this reason, a review of the main approaches to the modeling of networks-on-chip was made and the approach for networks-on-chip modeling by using SystemC NoCTweak networks-on-chip simulator was chosen. The analysis of the simulation results makes it possible to formulate the following conclusion: the high sensitivity of mesh network topology to its geometric proportions is found; the use of topologies which are close to the square form allows to increase the network capacity up to 24,7 %, and the central location of "hot spots" allows up to 9 % reduction in the delay of packets, up to 15,2 % — in energy consumption for each packet transmission and up to 19,5 % increase of the network capacity.

Keywords: network-on-chip, system-on-chip, network-on-chip regular topology, networks-on-chip design, SystemC programming language, network-on-chip model, mesh topology, "hot spots"

References

1. **Axel J., Hannu T.** *Networks on Chip*, Dordrecht, Kluwer Academic Publishers, 2003, 303 p.

2. **Dally W., Towles B.** Principles and practices of interconnection networks. Elsevier, 2004. 550 p.

3. **Dally W., Aoki H.** Deadlock-free adaptive routing in Multi-computer Networks using virtual channels, *IEEE Transactions on Parallel Distributed Systems*, 1993, vol. 4, no. 4, pp. 466–475.

4. **Balfour J., Dally W.** Design Tradeoffs for Tiled CMP On-Chip Networks, *ICS'06: Proceedings of the 20th Annual International Conference on Supercomputing*, ACM Press, 2006, pp. 187–198.

5. **Saldana M., Shannon M., Chow P.** The Routability of Multi-processor Network Topologies in FPGAs, Proceedings of the 2006 international workshop on System-level interconnect prediction (SLIP'06), NY, ACM Press, 2006, pp. 49–56.

6. **Benini L., Bertozzi D.** Network-on-chip architectures and design methods, *Computers and Digital Techniques*, IEEE, 2005, vol. 152, no. 2, pp. 261–272.

7. **Romanov O. Yu., Lysenko O. M.** Obg'runtuvannja vyboru metodu ta zasobiv modeljuvannja mrezh na krystali [Justification of the choice of methods and means of networks-on-chip modeling], *Naukojenni tehnologii, Zbirnyk naukovykh prac'*, Kyiv, NAU, 2014, no. 1 (21), pp. 49–54.

8. **Genko N., Bertozzi D., De Micheli G., Benini L.** Feature-NoC emulation: a tool and design flow for MPSoC, *IEEE Circuits and Systems Magazine*, 2007, vol. 7, no. 4, pp. 42–51.

9. **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.

10. **Murali S., De Micheli G.** Bandwidth-constrained mapping of cores onto NoC architectures, *Proceedings of the Conference on De-*

sign, Automation and Test in Europe, 2004 (DATE'04), Paris, 2004, vol. 2, pp. 16–20.

11. **Bertozzi D., Murali S., Jalabert A.** NoC synthesis flow for customized domain specific multiprocessor systems-on-chip, *IEEE Transactions on Parallel and Distributed Systems*, 2005, vol. 16, no. 2, pp. 113–129.

12. **Mahadevan S., Virk K., Madsen J.** ARTS: A SystemC-based framework for multi-processor Systems-on-Chip modeling, *Design Automation for Embedded Systems*. Springer, 2007, vol. 11, no. 4, pp. 285–311.

13. **Fazzino F., Palesi M., Patti D.** Noxim: Network-on-chip simulator. URL: <http://noxim.sourceforge.net/>

14. **Jain L.** NIRGAM: A Simulator for NoC Interconnect Routing and Application Modeling, Version 1.1, 2007, 27 p., URL: <http://nirgam.ecs.soton.ac.uk/Documentation.php>.

15. **Tran A. N.** On-Chip Network Designs for Many-Core Computational Platforms: Ph. D. thesis, USA, Davis: University of California, 2012, 156 p.

16. **Romanov O., Lysenko O.** The Comparative Analysis of the Efficiency of Regular and Pseudo-optimal Topologies of Networks-on-Chip Based on Netmaker, *1st Mediterranean Conference on Embedded Computing, MECO 2012. Advances and Challenges in Embedded Computing: Proceedings. Montenegro, Bar: IEEE, 2012, pp. 13–16.*

17. **Fes'kov D. O., Romanov O. Yu., Korotkyj Je. V.** Programna model' mrezh na krystali z nereguljarnymy topologijamy [Networks-on-chip program model with irregular topologies], *Problemy informatyzacii ta upravlinnja. Zbirnyk naukovykh prac'*, Kyiv, NAU, 2013, no. 2 (42), pp. 118–123.

18. **Gebali F., Elmiligi H., Watheq M.** Networks-on-Chips: Theory and Practice, FL: CRC Press, 2009, 389 p.