

Феськов Д.О.,
Романов О.Ю.,
Короткий Є.В.

ПРОГРАМНА МОДЕЛЬ МЕРЕЖ НА КРИСТАЛІ З НЕРЕГУЛЯРНИМИ ТОПОЛОГІЯМИ

Національний технічний університет України «КПІ»

Проведено огляд різних підходів до моделювання мереж на кристалі (МнК). Розроблено симулятор МнК, у якому топологія задається матрицею зв'язків між роутерами, що керують трафіком за допомогою таблиць маршрутизації. Розглянуто можливості симулятора МнК та представлені результати його апробації на прикладі регулярних і квазіоптимальних мереж

Постановка задачі

З метою дослідження можливостей мереж на кристалі (МнК) існує необхідність у їх моделюванні. Залежно від рівня деталізації розрізняють моделі низького та високого рівнів.

Низькорівневе моделювання – емуляція роботи мережі на рівні логічних вентилів. Компоненти моделі формуються із застосуванням мов опису апаратури (наприклад, Verilog або VHDL). При цьому їх функціонування аналізується за допомогою спеціалізованих програм моделювання апаратних засобів (наприклад, пакет ModelSim). Така модель може бути синтезованою за допомогою спеціалізованих САПР. Недоліком даного підходу є значні витрати часу на побудову моделі та моделювання [1]. Його доцільно використовувати для оцінювання ефективності окремих рішень при побудові МнК.

Високорівневе моделювання – симуляція роботи мережі на рівні розповсюдження потоків даних. Цей підхід характеризується швидкістю розробки, гнучкістю налаштувань та значно меншим часом моделювання. Дані моделі поділяються на ті, що синтезуються, та такі, що не синтезуються.

Моделі, що синтезуються, формуються із використанням мови SystemC. Це мова проектування та верифікації моделей системного рівня, реалізована у вигляді C++ бібліотеки з відкритим кодом. Бібліотека містить ядро подієвого моделювання, що дозволяє отримати викону-

вану модель пристрою. Мова застосовується для побудови транзакційних і поведінкових моделей, а також для високорівневого синтезу пристроїв. SystemC використовує ряд понять, схожих до тих, які мають мови опису апаратури VHDL і Verilog – інтерфейси, процеси, сигнали, подієвість, ієрархія модулів. Завдяки цьому SystemC придатна для поведінкового моделювання та RTL-синтезу (Register transfer level – на рівні регістрів).

SystemC широко використовується серед розробників МнК. На основі SystemC створена бібліотека Xpipes [2], що дає можливість здійснювати повний цикл моделювання та синтезу МнК [3, 4]. У роботі [5] запропонована високорівнева SystemC-модель ARTS для порівняльного моделювання мережевого та шинного методів побудови систем на кристалі. На основі SystemC достатньо відомими є симулятори Noxim [6], NIRGAM [7] та інші.

Популярність SystemC обумовлена тим, що вона базується на мові C/C++. Однак за своєю природою C/C++ є послідовною мовою, у той час як процеси в апаратурі відбуваються одночасно і паралельно. Це вимагає опанування нової парадигми програмування, а також специфічних інструментів, таких як процеси, події, сигнали та ін.

Згідно з роботою [8] максимальна швидкість моделювання за допомогою ModelSim становить приблизно $3,2 \cdot 10^3$ циклів/с, SystemC – $20 \cdot 10^3$ циклів/с, при цьому кристал FPGA функціонує із частотою від 50 МГц ($50 \cdot 10^3$ цик-

лів/с). Найшвидшим є прототипування МнК в FPGA, але воно вимагає великих витрат у часі на розробку проекту та наявності спеціального обладнання.

Альтернативою може слугувати симуляція, що представляє собою тестування моделі розповсюдження даних у МнК, описаної мовою високого рівня. Так, у дослідженні [9] у вигляді формули представлено залежність швидкості паралельної обробки даних від параметрів МнК та аналіз впливу на неї затримок при передачі даних зі збільшенням розмірності

мережі. У праці [10] використовується моделювання у мережах Petri у середовищі симулятора Visual Object Net для аналізу конкуренції, взаємодії та конфліктів даних у комунікаційному середовищі МнК. Окремо слід відзначити роботи [11, 12], де запропоновано універсальний програмний симулятор мовою Java та наведено результати моделювання різних регулярних топологій. Проте, введення нової топології вимагає переробки архітектури симулятора.

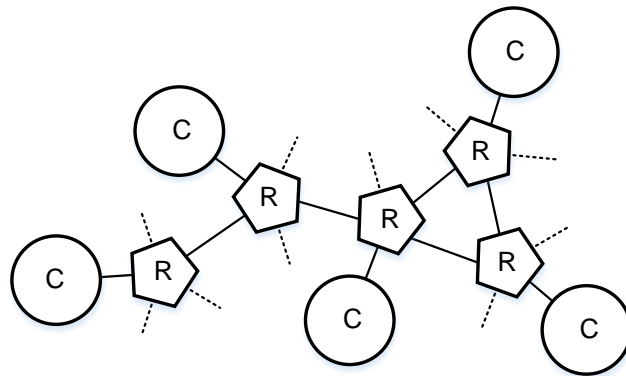


Рис. 1. Верхній рівень моделі МнК із нерегулярною топологією

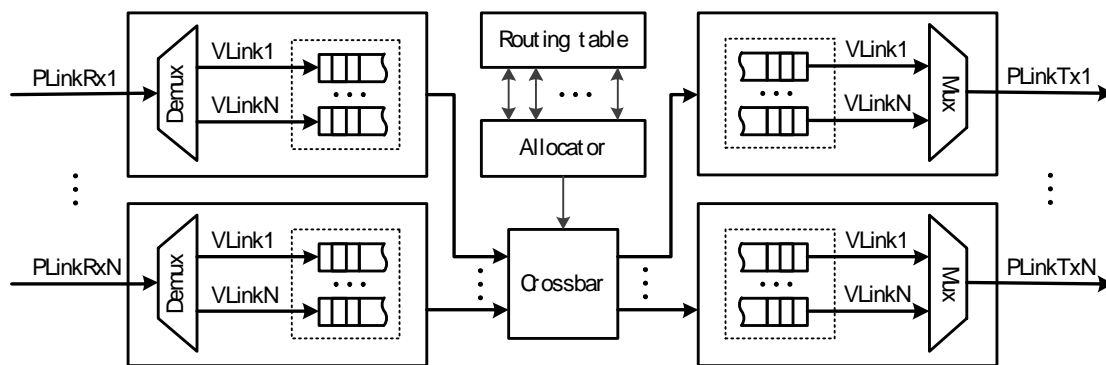


Рис.

2. Модель МнК на рівні роутера

У працях [13, 14] запропоновано підхід до побудови квазіоптимальних нерегулярних топологій МнК, що мають кращі характеристики, порівняно з регулярними. Існує необхідність у високорівневому порівняльному моделюванні нових квазіоптимальних топологій із поширеними регулярними.

Симулятор нерегулярних МнК

У працях [15, 16] запропоновано швидко високорівневу модель МнК на основі мережевої моделі OSI з застосуванням мови Java та фреймворку

Qt Jambi. Компоненти даної моделі було модифіковано з метою моделювання нерегулярних топологій: у результаті кожний роутер містить таблицю маршрутизації, а топологія МнК задається за допомогою матриці зв'язків між роутерами.

Структура моделі ілюструється рис. 1, 2. За кожен компонент моделі OCNS (On-Chip Network Simulator) відповідає власний модуль. Модуль верхнього рівня ієрархії Runner керує процесом моделювання. Модуль RunnerData управляє доступом до ресурсів мережі та містить модулі Config, Network і Statistic. Модуль

Config відображає налаштування моделі відповідно до параметрів вхідного файлу. Модуль Network відповідає за побудову мережі, переміщення трафіку та збір статистики і містить множину модулів Core та Router. Модуль Core симулює роботу

IP-ядра: генерацію пакетів із заданою частотою, дроблення їх на фліти та відправку роутеру, прийняття флітів, відтворення вихідного пакету, збір статистики. Модуль Router моделює роботу роутера і містить таблицю маршрутизації Routing

Табл. 1. Параметри конфігураційного файлу

Параметр	Зміст
Netlist	Матриця з'єднань
Routing	Таблиця маршрутизації
VLinkSize	Розмір буфера віртуального каналу, флітів
VLinkCount	Кількість віртуальних каналів
FlitSize	Розмір фліта, бітів
PacketSizeAvg	Середнє значення довжини пакету
PacketFixed	Опція. Фіксована довжина пакету
RunCount	Кількість прогонів симулятора
RunTime	Час моделювання, прийнятих пакетів
WarmUpTime	Час насичення моделі, прийнятих пакетів

Table, схему комутації Allocator та множину контролерів вхідних та вихідних з'єднань – PLinkRx та PLinkTx відповідно. Для наочності, на рис. 2 не вказаний модуль управління Contol, що координує роботу всіх підсистем роутера. Контролери PLinkRx та PLinkTx містять буфери для флітів вхідних та вихідних пакетів – BufferRx та BufferTx відповідно. Кожен із буферів організований у вигляді набору черг віртуальних каналів. BufferRx додатково містить інформацію для комутації Data-флітів, що передаються вслід за Head-флітом вхідного пакету.

Параметри моделі задаються за допомогою конфігураційного xml-файлу (табл. 1). Результати моделювання виводяться у діалогове вікно, а вибрані параметри зберігаються у зведену таблицю. Симулятор дозволяє виконувати декілька ітерацій моделювання підряд із різною конфігурацією.

За результатами моделювання симулятор надає такі параметри продуктивності МнК, як: кількість відправлених і прийнятих пакетів; середній час доставки пакета; середня кількість проміжних сегментів, які проходить пакет (кількість хопів); пропускна здатність роутера; пропускна здатність мережі; середнє завантаження фізичного каналу зв'язку; заванта-

женість вхідних і вихідних буферів роутера та IP-блоку тощо.

Симулятор реалізовано мовою програмування Java з використанням фреймворку Qt Jambi [17], що надає всі переваги об'єктно-орієнтованого програмування, кросплатформності програмних рішень і швидкості їх розробки. Повна незалежність компонентів OCNS дає можливість виконувати розробку, модифікацію та апробацію різних моделей МнК.

Апробація моделі OCNS

Проведено моделювання mesh, torus, деревоподібної та квазіоптимальних топологій із кількістю зв'язків від 8 до 18 для 9 вузлів, середньої довжини пакету 10 флітів по 32 біти і з різною інтенсивністю введення пакетів у мережу. Параметри моделювання: застосовуються маршрутизатори із п'ятьма портами, кожен порт яких має по 4 віртуальні канали розміром 4 фліта; процес моделювання відбувається, доки кожен вузол не відправить по 1100 пакетів і вони не будуть прийняті; для досягнення сталого процесу, підрахунок статистики кожним вузлом починається після надходження перших 100 warm-up пакетів.

Час моделювання одного варіанту топології не перевищував 1 хвилини (комп'ютер ASUS K40AB), що значно

менше, ніж із використанням низькорівневої Verilog-моделі Netmaker [18]. Результати моделювання (рис. 3, 4) повною мірою корелюють із даними, отриманими за допомогою Netmaker – у випадку, коли кількість зв'язань квазіоптимальної топології співпадає із регулярною при однако-

вій кількості вузлів і квадратній формі регулярної топології, її пропускна здатність вища на 2-4%. Тобто використання високорівневої моделі OCNS дає результати моделювання, близькі до точної низькорівневої HDL-моделі Netmaker, але за значно менший час (прибл. у 120 разів).

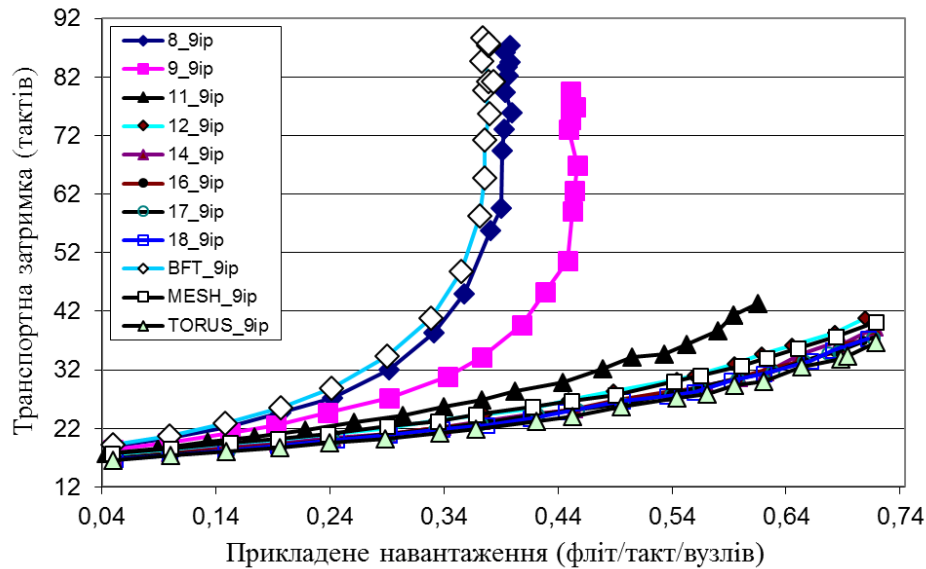


Рис. 3. Результати моделювання регулярних та квазіоптимальних топологій із кількістю зв'язків від 8 до 18 для 9 вузлів

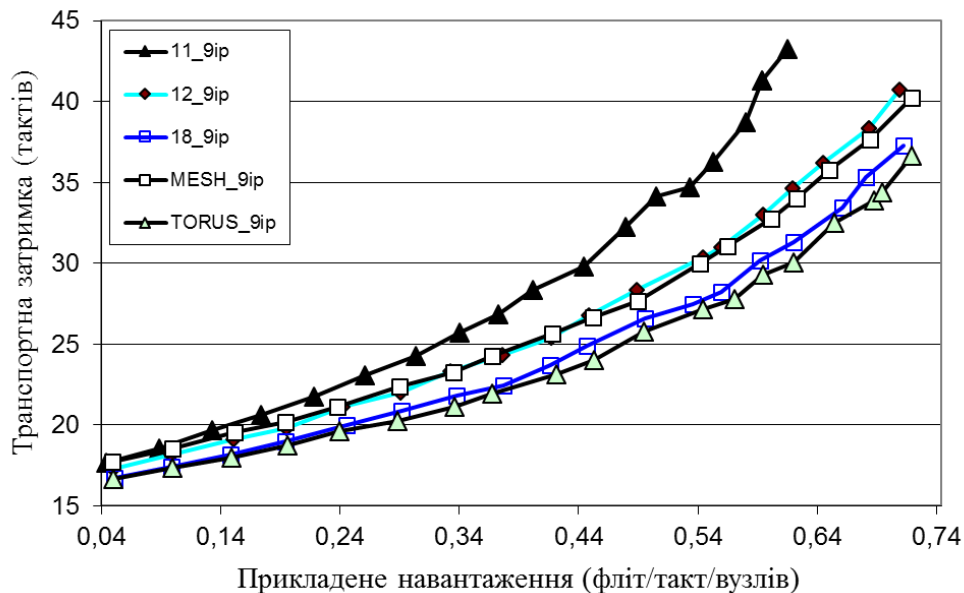


Рис. 4. Результати моделювання регулярних і квазіоптимальних топологій із кількістю зв'язків від 11 до 18 для 9 вузлів

Через значні витрати часу моделювання топологій, кількість вузлів яких сягає десятків, за допомогою HDL-підходу є проблематичним. Саме для моделювання таких МнК і розроблено OCNS. На рис. 5 наведено графіки залежностей затримки пакетів від інтенсивності введення їх до

МнК, отримані в результаті моделювання mesh, torus та квазіоптимальної топологій для кількості вузлів 99 і 100. Параметри моделювання було залишено такими, як і в попередньому випадку. Тривалість повного циклу моделювання при різних інтенсивностях введення пакетів виявилася

максимальною у топології mesh із 100 вузлами і не перевищила 5 хвилин

(комп'ютер ASUS K40AB).

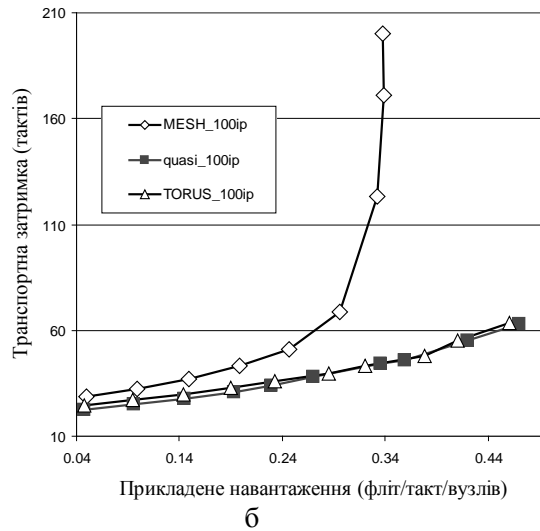
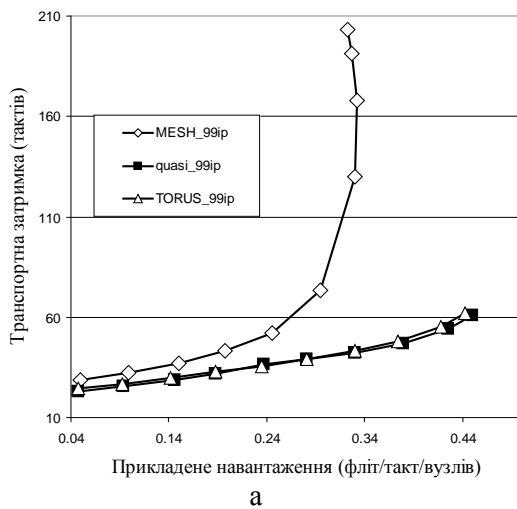


Рис. 5. Результати моделювання топологій для 99 (а) та 100 (б) вузлів

Для зручності порівняння пропускної спроможності топологій обрано поріг насичення на рівні середньої затримки проходження пакетів, рівної 60 тактам (у три рази більше за мінімальну затримку для топологій – 20 тактів). Проведено сплайн-інтерполяцію графіків функцій і знайдено значення пропускної спроможності при пороговому значенні затримки пакетів у 60 тактів.

Як можна побачити з рис. 5, графік, що відповідає квазіоптимальній топології, практично співпадає із torus, у той час як mesh має значно гіршу пропускну спроможність. При $N = 100$ (рис. 5а), пропускну спроможність квазіоптимальної топології – 0,440, тоді як у torus – 0,429 (лише на 2,5% менше), а у mesh – 0,279 (на 36,6% менше). При цьому квазіоптимальна топологія має на 8 з'єднань менше, ніж torus, і лише на 12 – більше за mesh. Для топологій із 99 вузлами (рис. 5б) ситуація практично не змінилась (пропускну спроможність збільшилась на 2–3% за рахунок меншої кількості вузлів).

Висновки

Проведено огляд різних підходів до моделювання МНК. Запропоновано швидку високорівневу модель МНК OCNS на основі мережевої моделі OSI з застосуванням мови Java і фреймворку Qt, що надало можливість досягти точності ре-

зультатів моделювання, близьких до Netmaker [18], але зменшило час моделювання більше, ніж у 120 разів.

Представлено результати апробації на прикладі регулярних та квазіоптимальних МНК. Моделювання топологій для кількості вузлів 99, 100 не перевищило 5 хвилин (комп'ютер ASUS K40AB).

Результати моделювання МНК за допомогою OCNS виявили при $N = 100$ пропускну спроможність квазіоптимальної топології – 0,440, тоді як у torus – 0,429, а у mesh – 0,279. При $N = 99$ пропускну спроможність відповідних мереж збільшилась на 2–3% за рахунок меншої кількості вузлів. В обох випадках квазіоптимальна топологія має менше з'єднань у порівнянні з torus, та значно більшу пропускну спроможність, у порівнянні із mesh.

Перспективним напрямком подальших досліджень є розробка моделі МНК, що надасть змогу об'єднати у мережу роутери з різною кількістю портів.

Список літератури

1. Genko N. Feature-NoC emulation: a tool and design flow for MPSoC / N. Genko, D. Atienza, G. De Micheli, L. Benini // IEEE Circuits and Systems Magazine. – 2007. – Vol. 7. – No. 4. – pp. 42–51.
2. Bertozzi D. Xpipes: A network-on-chip architecture for gigascale systems-on-

- chip / D. Bertozzi, L. Benini // IEEE Circuits and Systems Magazine. – 2004. – Vol. 4. – No. 2. – pp. 18–31.
3. Murali S. Bandwidth-constrained mapping of cores onto NoC architectures / Murali S., De Micheli G. // Proceedings of the Conference on Design, Automation and Test in Europe, 2004 (DATE'04). – Paris, 2004. – Vol. 2. – pp. 16–20.
4. Bertozzi D. NoC synthesis flow for customized domain specific multiprocessor systems-on-chip / D. Bertozzi, S. Murali, A. Jalabert // IEEE Transactions on Parallel and Distributed Systems. – 2005. – Vol. 16. – No. 2. – pp. 113–129.
5. Mahadevan S. ARTS: A SystemC-based framework for multiprocessor Systems-on-Chip modelling / S. Mahadevan, K. Virk, J. Madsen // Design Automation for Embedded Systems. – Springer, 2007. – Vol. 11. – No. 4. – pp. 285–311.
6. Fazzino F. Noxim: Network-on-chip simulator [Електронний ресурс] / F. Fazzino, M. Palesi, D. Patti. – Режим доступу: <http://noxim.sourceforge.net/>.
7. Jain L. NIRGAM: A Simulator for NoC Interconnect Routing and Application Modeling [Електронний ресурс] / L. Jain. – [Version 1.1]. – 2007. – 27 p. – Режим доступу: <http://nirgam.ecs.soton.ac.uk/Documentation.php>.
8. Genko N. A Complete Network-On-Chip Emulation Framework / N. Genko, D. Atienza, G. De Micheli, et al. // Design, Automation and Test in Europe, 2005. Proceedings. – 2005. – Vol. 1. – pp. 246–251.
9. Xiaowen C. Speedup Analysis of Data-parallel Applications on Multi-core NoCs / C. Xiaowen, L. Zhonghai; A. Jantsch, C. Shuming // IEEE 8th International Conference on ASIC, 2009 (ASICON'09). – 2009. – pp. 105–108.
10. Freitas H.C. Evaluating On-Chip Interconnection Architectures for Parallel Processing / H.C. Freitas, P.O.A. Navaux // 11th IEEE International Conference on Computational Science and Engineering Workshops, 2008 (CSEWORKSHOPS'08). – 2008. – pp. 188–193.
11. Hossain H. GpNoCsim – A General Purpose Simulator for Network-on-Chip / H. Hossain, M. Ahmed, A. Al-Nayeem // International Conference on Information and Communication Technology, 2007 (ICICT'07). – 2007. – pp. 254–257.
12. Al-Nayeem A. GpNoCsim 1.0 User's Guide / A. Al-Nayeem, T. Z. Islam. – 2006. – 13 p.
13. Романов О.Ю. Оптимальні топології мереж на кристалі / О.Ю. Романов // Комп'ютерні науки та інженерія: Матеріали V Міжнародної конференції молодих вчених CSE-2011. – Львів: Львівська політехніка, 2011. – С. 132–135.
14. Романов А.Ю. Оптимизация топологий сетей на кристалле / А.Ю. Романов // Вісник НТУ "ХПІ". Збірник наукових праць. Тематичний випуск: Інформатика і моделювання. – Харків: НТУ "ХПІ", 2011. – № 36. – С. 149–155.
15. Романов А.Ю. Разработка программного симулятора сетей на кристалле / А.Ю. Романов, Д.А. Феськов // Электроника и связь: Электроника и нанотехнологии. – Киев: НТУУ «КПИ», 2011. – Т. 4(63). – С. 48–52.
16. Романов О.Ю. Програмна модель багатопроекторної мережі на кристалі / О.Ю. Романов, Д.О. Феськов // IV міжнародна науково-технічна конференція молодих вчених «Електроніка-2011». Збірник статей. – Київ: «АВЕРС», 2011. – Ч. 2. – С. 118–123.
17. Офіційний веб-сайт “Qt Jambi” [Електронний ресурс]. – Режим доступу: <http://qt.nokia.com/>
18. Romanov O. The Comparative Analysis of the Efficiency of Regular and Pseudo-optimal Topologies of Networks-on-Chip Based on Netmaker / O. Romanov, O. Lysenko // Advances and Challenges in Embedded Computing. Proceedings. – Montenegro, Bar: 2012. – pp. 13–16.