

**Висновки.** Сукупність показників шкідливих речовин і випромінювань та методика їх використання можуть використовуватися як засоби оцінювання екологічності комп'ютера.

Запропонована авторами методика ґрунтується на використанні узагальненого коефіцієнта екологічності, який враховує екологічність на життєвих етапах виробництва, експлуатації та утилізації комп'ютера.

Список показників, що використовуються на кожному життєвому етапі, може уточнюватись і доповнюватись, що дасть змогу отримувати достовірніші оцінки екологічності комп'ютера.

1. Arik Hesseldahl, "Computers' Elusive Eco Factor", [електрон.ресурс] – <http://businessweek.com>. 2. Greenpeace International "Problem of ecological of computers" [електрон.ресурс] – <http://www.greenpeace.org/electronics> 3. HP Eco Solutions "How Green Is Your PC?" [електрон.ресурс] – <http://hp.com> 4. EPEAT's environmental criteria [електрон.ресурс] – <http://www.epeat.net> 5. Green Electronics: the Search Continues, pp.16-18, Dec. 2009 6. Голубович О. Технології енергосбереження, [електрон.ресурс] – <http://ITC.UA>

УДК 004.031.43

Н.Б. Козак

Національний університет "Львівська політехніка",  
кафедра електронних обчислювальних машин

## ОЦІНКА АПАРАТНИХ ВИТРАТ НА РЕАЛІЗАЦІЮ МАТРИЧНОЇ ОДНОЯРУСНОЇ КОМУТУЮЧОЇ МЕРЕЖІ НА ПЛІС СІМЕЙСТВА VIRTEX

© Козак Н.Б., 2009

Для деяких багатопроцесорних систем на кристалі потрібно використовувати неблокуючі комутаційні мережі. Разом з тим функціонування такі системи вимагають високої швидкодії. Матрична одноярусна комутуюча мережа володіє властивістю неблокуючої і є найшвидшою серед всіх комутуючих мереж. Недоліком цієї комутуючої мережі є великі апаратні витрати на її реалізацію. Але зі збільшенням інтеграції ПЛІС реалізація одноярусної комутуючої мережі стала доцільною. Отже, необхідно розглянути її реалізацію на ПЛІС.

**It is necessary to utilize nonblocking comutating networks for some multiprocessor systems on chip. At the same time a functioning of such systems imposes the requirement of high fast-acting. The matrix single-stage comutating network owns a property of nonblocking and is most rapid among of all comutating networks. The shortcoming of this comutating network are large hardware charges on its implementation. But with the increase of integration of FPGA a implementation of single-stage comutating network became expedient. So,it is necessary to consider its implementation on FPGA.**

**Вступ.** З розвитком комп'ютерної техніки на одному кристалі почали створювати цілі мережі на основі багатопроцесорної технології з паралельним обробленням інформації. До цього для передавання даних між компонентами комп'ютерної системи спочатку використовували шини. Використання такого підходу насамперед було пов'язано з економією апаратних ресурсів, які тоді були значно обмежені у використанні, а використання шини вимагало лише апаратних витрат на реалізацію арбітра і шинного формувача[1]. Та оскільки одна шина може одночасно забезпечувати доступ тільки одного процесорного елемента до іншого, то продуктивність такого підходу стала

недостатньою. Тому для збільшення продуктивності таких систем почали одночасно використовувати декілька шин. Поширеною методологією стало використання окремих шин для швидких елементів системи і для повільних.

Цей підхід виявився недостатнім для створення високопродуктивних систем, тому наступним кроком мало би стати повернення до ідеї мультиплексування. Але розвиток мікроелектроніки і тоді не давав змоги використовувати велику кількість мультиплексорів. Тим не менше він все-таки дав змогу прийти до деякого компромісного варіанта. Річ у тім, що коли ми маємо ряд паралельно включених мультиплексорів, то в їх структурі можна певні частини об'єднувати, значно економлячи апаратні ресурси. Натомість тоді така топологія матиме ознаки блокуючої, оскільки спрацювання спільних блоків топології може виконувати корисну функцію лише для однієї з передбачених ситуацій. Попри блокуючу властивість такого підходу, він все одно принципово значно продуктивніший за спосіб використання кількох шин. Такий підхід в останні роки майже повністю замінив класичний підхід використання шин, і хоча системні шини CoreConect та AMBA [3, 4] і називаються системними шинами, але в їх основу покладено саме такий підхід.

Згодом з'явилися задачі, які вимагали неблокуючих топологій. Такі топології вже на той час були відомі. В чому ж полягав програш у використанні неблокуючих топологій, оскільки вони давали виграв у апаратурі і були неблокуючими, а завжди за підвищення одного з параметрів доводиться платити пониженням іншого? Річ у тім, що для усунення властивості блокування вводиться деяка надлишковість у можливих ситуаціях проходження сигналу. Але для реалізації цієї надлишковості загальна довжина шляху проходження сигналу збільшується. От і програємо в швидкодії.

І лише в останні роки з'явилася тенденція до використання мультиплексорів. Такий спосіб [2] отримав назву одноярусної комутуючої мережі. Тому потрібно розглянути аспекти її реалізації на структурі ПЛІС. І оскільки лідером ринку в цій галузі є XILINX зі своєю передовою архітектурою Virtex, то доцільно орієнтуватись саме на неї під час оцінювання ефективності такого рішення.

**Реалізація мультиплексорів на ПЛІС сімейства Virtex.** У кристалі програмованої логічної матриці будь-яка логіка синтезується з примітивів, які елементарно конфігуруються в одній секції логіки. Для реалізації якогось мультиплексора в структурі ПЛІС доцільно використовувати мультиплексорні примітиви. В архітектурі Virtex-5 та Virtex-6 кожна секція логіки має по чотири генератори логіки[5], що формують логіку відповідно до 64 бітів пам'яті, яка може відобразитися як таблиця 5 в 2 або 6 в 1. Тобто один генератор логіки відображує мультиплексор 4 в 1. Як вже було сказано, в одній секції логіки таких генераторів чотири, тому і мультиплексорів можна створити чотири. Крім того, в складі кожної секції логіки є спеціально три двовходові мультиплексори для об'єднання цих чотиривходових мультиплексорів в один 16-входовий. Відповідно цей примітив може бути використаний для побудови багатовходових мультиплексорів. Такий підхід швидше за все залишиться актуальним і в наступних архітектурах ПЛІС Virtex. Тому потрібно розглянути математичні вирази для оцінювання апаратних витрат на реалізацію мультиплексора заданої розрядності на базі мультиплексорних примітивів визначеної розрядності. Також буде доцільним розглянути вираз для оцінювання швидкодії створеного мультиплексора, що фактично виражатиме кількість ярусів мультиплексорних примітивів для реалізації заданого мультиплексора.

Нехай ми маємо створити мультиплексор розрядності  $M$ . Тоді на першому ярусі маємо  $n_0 = M$  входів, на другому –  $n_1$  і на  $i$ -му –  $n_i$ . Коли ми використовуємо мультиплексорні примітиви вхідної розрядності  $m$ , то буде справедливе таке співвідношення, яке показує відношення кількості мультиплексорів між сусідніми ярусами(*формула дісна для цілочислової арифметики*):

$$n_{i+1} = \frac{n_i - 1 + (m - (n_i - 1) \bmod m) \bmod m}{m} \quad (1)$$

У формулі зроблено заміну  $(n_i - 1)$  замість  $n_i$  для того, щоб знехтувати тим випадком, коли залишатиметься лише один зайвий вхід, який просто увійде у наступний ярус.

Для обчислення загальної кількості ярусів, що буде показником швидкодії мультиплексора, потрібно розбити кількість входів на дві складові  $M_1$  і  $M_2$ , де  $M_1$  – кількість входів, з яких можна взяти цілочисловий логарифм за основою  $m$ . Обчислити кількість ярусів  $J_1$  мультиплексора, який би мав  $M_1$  входів, можна за формулою:

$$J_1 = \log_m M, \quad (2)$$

тоді

$$M_1 = m^{J_1}. \quad (3)$$

Зрозуміло: якщо  $M=M_1$ , то  $J=J_1$ . Якщо ж  $M \neq M_1$ , то  $J=J_1+1$ . Для вираження цієї умови можна скористатися формулою, яка буде справедлива для цілочислової арифметики:

$$J = \log_m (M + (mM_1 - M) \bmod (mM_1 - M_1)). \quad (4)$$

Тоді загальну кількість мультиплексорних примітивів можна визначити за формулою

$$N = \sum_{i=0}^J \frac{n_i - 1 + (m - (n_i - 1) \bmod m) \bmod m}{m}. \quad (5)$$

**Реалізація одноярусної комутуючої мережі.** Із створених так мультиплексорів вхідної розрядності  $M$  можна елементарно створити одноярусну комутуючу мережу  $M$  в  $K$ . Та для дуже великих проектів може скластися ситуація, коли недоцільно використовувати повну версію одноярусної комутуючої мережі, оскільки процесорні елементи, що виконують складні функції, можуть працювати на частотах, значно менших, ніж які дозволить така швидка комутуюча мережа. Тому можна розглянути деяку **одноярусну „урізану“ комутуючу мережу  $M$  в  $L[S]$** . Така комутуюча мережа працюватиме ітераційно. Суть її „урізаного“ функціонування полягатиме не в демультиплексуванні проміжних етапів комутації (це б сповільнювало процес комутації) на вихідні точки призначення сигналу, а в фіксації їх на вхідних тригерах у процесорні елементи за допомогою стробового сигналу, який формується з коду номеру ітерації.

Тоді для такого каскаду в режимі ітерацій існуватимуть певні залежності між апаратними затратами, швидкістю і невикористаним залишком виходів мультиплексорів в останньому циклі.

Спочатку потрібно розрахувати кількість невикористаних залишкових мультиплексорів в останньому циклі  $V$ , виходи яких будуть непотрібні в кінці процесу комутації. Закономірно, що їхня кількість повинна дорівнювати оберненому залишку (доповненню) від ділення на кількість мультиплексорів  $L$ , що їх буде використано. Формула матиме вигляд:

$$V = (L - K \bmod L) \bmod L. \quad (6)$$

Наступним кроком буде обчислення сповільнення  $S$  порівняно з повністю паралельною реалізацією:

$$S = \frac{K + V}{L}. \quad (7)$$

Також потрібно визначити кількість можливих варіантів побудови  $k$  цього ітерційного каскаду мультиплексорів, що ґрунтуватиметься на кількості мультиплексорів :

$$k = \frac{M + 1}{2} + 1. \quad (8)$$

**Висновок.** Одноярусна комутуюча мережа має добрі перспективи для реалізації у системах на кристалі, процесорні елементи яких вимагають практично миттєвого обміну одинарними даними. З подальшим ростом об'єму конфігурованої логіки на кристалі ПЛІС доцільність її реалізації ще збільшуватиметься.

1. Леонтьев В.П. Новейшая энциклопедия персонального компьютера. – М: ОЛМА-ПРЕСС, 2002. 2. Мельник А.О. Архитектура комп'ютера: Підручник. – Луцьк: Видавництво обласної друкарні, 2008. 3. [http://www.arm.com/sitearchitek/armtech.ns4/html/amba?Open\\_Document&style=SoC\\_Customization](http://www.arm.com/sitearchitek/armtech.ns4/html/amba?Open_Document&style=SoC_Customization) 4. <http://www.chips.ibm.com/products/coreconnect/> 5. [www.xilinx.com](http://www.xilinx.com).