

УДК 004.722

*Краснобаєв В.А., доктор технічних наук, професор,
Гриценко Д.М., студент групи 601-ТСм
Полтавський національний технічний університет
імени Юрія Кондратюка*

ПЕРЕДОБЧИСЛЕННЯ МІЖЗ'ЄДНАНЬ ДЛЯ ПРОГРАМОВАНИХ ЛОГІЧНИХ ІНТЕГРАЛЬНИХ СХЕМ

В даній статті проведений огляд деяких проблем, пов'язаних з оцінкою лінійного обчислення для проектування ПЛІС. Вказані характерні відмінності між тим, які міжз'єднання використовуються в ПЛІС і вентильних матрицях.

***Ключові слова:** програмована логічна інтегральна схема, архітектура, передобчислення міжз'єднань.*

Вступ

Класичне передобчислення міжз'єднань, здавалося б, ідеально підходить для проектування програмованих логічних інтегральних схем (ПЛІС). Ще теоретичні моделі, ті, які засновані на правилі Рента зазвичай використовуються тільки для грубого обчислення на ранніх стадіях розробки архітектури. На практиці основними методами обчислень є емпіричні методи (обчислення за допомогою багатьох тестів). Основними причинами такої різниці між теорією і практикою є те, що моделі важко перенести на реальні архітектури з ієрархічними і гетерогенними ресурсами.

Програмована логічна інтегральна схема розміру n є апаратним пристроєм, який містить n логічних елементів (4-LUT і DFF) і програмовану мережу міжз'єднань. Розробка користувача потім реалізується на загальній мікросхемі шляхом програмування кожної LUT (Lookup Table, пошукова таблиця). Мережева маршрутизація в PLD повинна бути в змозі реалізувати широкий діапазон розробок користувача.

Передумова цієї роботи ґрунтується головним чином на спостереженнях Рента, що лог-лінійна залежність між логікою і ОС відбуватиметься, в середньому, для «добре» розміщеної логіки:

$$\log(\text{зовнішні з'єднання}) = c + r \log(\text{розмір блоку}).$$

Це рівняння більш типово сформульоване у вигляді:

$$P = kB^r$$

Для блоків розміру B і зовнішніх з'єднань P . Показник ' r ' є "Рент Параметром", який зазвичай вважається характеристикою розробки або списком з'єднань користувача, в "типовому" діапазоні від 0,5 до 0,8, k вважається константою.

Використовуючи правило Рента, можна вивести цілий ряд різних стохастичних моделей для розподілу довжини дротів.

Фундаментальними допущеннями моделей є те, що горизонтальні і вертикальні блоки дроту є однорідними і рівномірними (тобто довжина всіх одиниць не відрізняються одна від одної).

Друге фундаментальне припущення полягає в тому, що у нас є «хороший» САПР. Ненадійність цього останнього припущення мотивує поняття Рент параметра як ланцюга Рент параметра, що виникає з найкращого з доступних САПР.

Передобчислення міжз'єднань для програмованих логічних інтегральних схем

Архітектура ПЛІС має фіксовану структуру маршрутизації, яка потім буде використовуватися кожним вхідним ланцюгом. У розробці ПЛІС, передобчислення міжз'єднань по суті означає, пошук відповідей на наступні питання :

- Скільки потрібно горизонтальних і вертикальних каналів?
- Як повинні бути сегментовані дроти?
- Якщо довжина дротів різна, то в якому співвідношенні?
- Чим можна замінити дроти для комутаторів?

- Наскільки добре підтримує апарат конкретні загальні функції такі як суматори, мультиплексори і алгоритми множення?
- Як буде організовано комутування?

На практиці ми також хочемо, щоб мати можливість відповісти на ці питання з деякими обмеженнями для параметрів. Наприклад, при впровадженні нового члена сім'ї архітектури може бути дозволено додавання одного типу дроту, але не іншого, або дозволено зростання тільки в одному напрямку, через обмеження конструкції.

Сегментована архітектура ПЛІС показана на рисунку 1, яка являє собою грубу абстракцію Xilinx 4000. Цей пристрій виглядає дуже схожим на стандартну вентиляну матрицю. Блоки L представляють собою 4-LUT логічні елементи які проходять від дротів в каналі між ними за допомогою комутаторів блока C (з'єднання). Дроти потім також з'єднуються між собою в блоці S (комутатор).

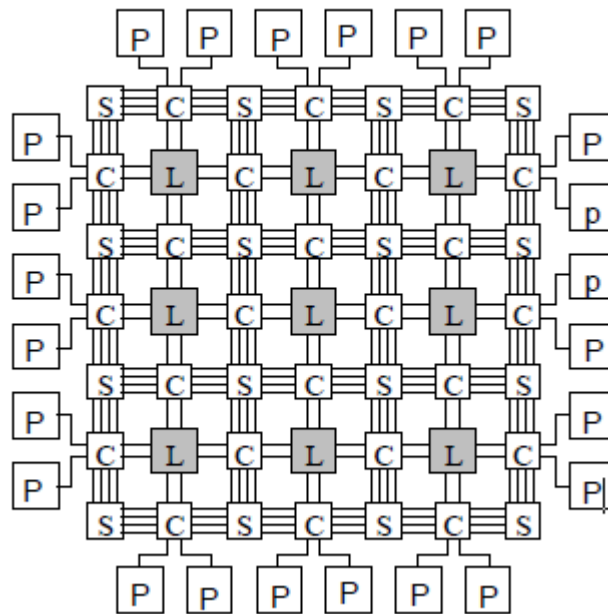


Рис 1. Проста сегментована архітектура ПЛІС

Використовуючи різні архітектурні моделі високого рівня, мікросхема Altera FLEX 10K є фундаментально ієрархічною (рисунок 2). Логічні елементи згруповані в кластерах 8-мого розміру. Кластери на одному і тому ж рядку

можуть управляти або управлятися за допомогою горизонтального дроту, який охоплює або всю ширину мікросхеми(GH), або півширину мікросхеми(HH). Сітка між різними рядками вертикального дроту(V) переходить на горизонтальний (GH або HH) дріт перед тим як ввійти в кластер.

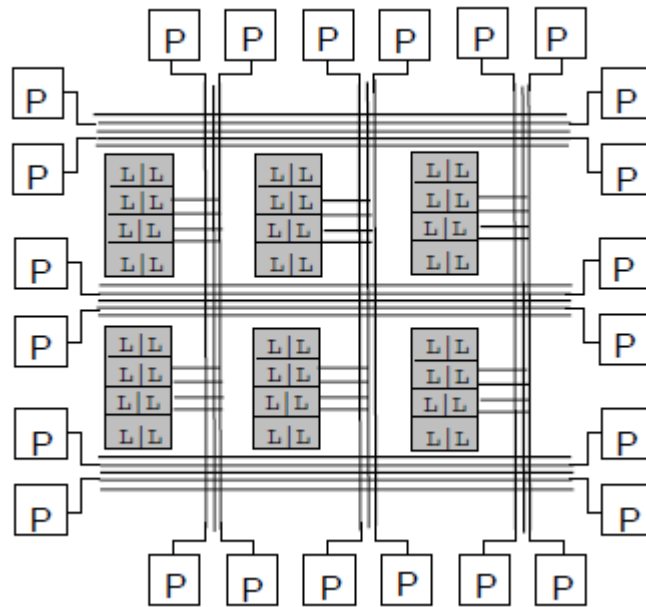


Рис 2. Проста ієрархічна архітектура ПЛІС

Для того, щоб зрозуміти різницю між цими двома стилями архітектури, необхідно представити дані архітектури як значно більші (в даний час ПЛІС можуть мати близько 50000 логічних елементів). На 1000 логічних елементів, сегментована архітектура (рис. 1) була б масивом приблизно 32x32 логічних елементів, і ієрархічна архітектура (рис. 2), 6 рядків по 16 стовпців кластерів, кожен кластер містив би 8 ЛЕ.

Проблеми передобчислень між'єднань

При застосуванні ASIC(інтегральна мікросхема спеціального призначення), оцінка між'єднань має припущення компактності і ефективності. Наприклад, якщо існує з'єднання точка-точка з відстанню 3, можна припустити, що будуть використовуватися 3 одиниці дроту.

Проте, програмована логіка по суті своїй є неефективною. Конфігурування 4-LUT робить дуже дорогою решітку «I», але це ціна яку ми платимо за

можливість для іншої схеми використовувати ту ж фізичну логіку для реалізації логічного елемента АБО. Точно так же, недовикористання інших ресурсів, таких як дроти, пам'ять і т.д. дозволяє іншим проектам бути реалізованими на цьому ж пристрої.

У програмованій логіці, площа мікросхеми переважається комутуванням. Більш конкретно площу можна розглядати як пропорційну кількість конфігурації SRAM бітів, необхідних для контролювання програмованих комутаторів. Таким чином, приймаючи довжину лінії за 10 проходиться відстань тільки 5.

Другою проблемою з передобчисленням між'єднань в ПЛІС, є те що ми повинні обробляти найгіршу ділянку мікросхеми всюди, де вона з'являється. У той час як вентиляльні матриці можуть розрізняти ділянки мікросхеми присвячені каналу даних і присвячені керуючій логіці, ПЛІС же повинні підтримувати різні ступені каналів даних. Проблема каналу даних в тому, що «не середня» - 32-бітна шина буде споживати набагато більше ресурсів як для «середньої» шини в конкретному каналі. Також достатня наявність ресурсів в кожному каналі для декількох шин буде непомірно дорогою.

Найбільшою проблемою яка впливає з усього цього є те, що лінія підрахунку (ширина каналу) для однієї мікросхеми повинна підтримувати найгіршу лінію підрахунку для найгіршого випадку списку з'єднань користувача. У разі алгоритмічних підрахунків, ми можемо легко переносити невелику кількість помилок, і можемо перенастроювати алгоритми під час виконання. Проте, як тільки архітектура фіксована, 5% помилка при оцінці каналу з найгіршим між'єднанням призведе до значного числа конструкцій які не помістяться на ділянці.

Третьою фундаментальною відмінністю між ПЛІС архітектурою і вентиляльною матрицею є можливість заміни гнучкої маршрутизації на самомаршрутизацію. Наприклад, за рахунок зменшення числа витків від V до H в ієрархічній частині, або впровадження комутуючих блоків в сегментній

частині можна зменшити загальну площу мікросхеми, навіть якщо кількість дротів на кожному каналі збільшиться.

В якості ще одного прикладу зовнішнього впливу на комутування розглянемо як контакти з'єднанні в загальній структурі маршрутизації. На обох моделях Рис 1 і Рис 2, існує природна відповідність між рядками/стовпцями і фізичною мікросхемою введення-виведення.

Попереднє призначення контактів (Наприклад, користувач який вносить зміни в ПЛІС після того, як розташування контактів на друкованій платі було завершено) може привести до різкого збільшення затримки в дротах і мережі. Ще однією проблемою пов'язаною з комутуванням є вторинні сигнали. В ASICS, невелика кількість вторинних сигналів (годинник, скидання, вмикання) можуть бути направлені незалежно один від одного, і не впливають на інші мережі. Однак в предструктурованій ПЛІС, вторинний сигнал мережі вже існує, і "доступність" конкретних сигналів часто обмежена в різних ділянках впливаючи на розміщення, а отже і на лінію підрахунків.

Наступною проблемою є затримка і площа, які є основними компромісами при розробці ПЛІС архітектур. Чим менше використовуються комутатори, тим більш продуктивним стає ПЛІС. Ахмед і Роуз обґрунтували, що архітектура на основі 2-х або 3-х вхідних LUTs краще для площі, що контрастує з більш ранніми результатами Брауна, LUTs розміром 3 або 4 були найбільш ефективними. Це вказує на те, що, геометричні форми змінюють взаємозв'язок між маршрутизацією і комутацією, і на те, що ПЛІС архітектури дуже чутливі до цього.

Друге питання, що виникає в контексті затримки є робота з некритичними з'єднаннями. По причині розміру ASIC не буде враховувати обхідний маршрут. Але в ПЛІС, ми вже "заплатили" за ці преструктуровані дроти. Це призводить до нераціонального використання вільних дротів.

Проста модель для з'єднань ПЛІС «APEX»

Застосуємо класичну теорію міжз'єднань для передобчислення кількості H і V дротів, необхідних для ПЛІС «APEX». хоча, в якійсь мірі це теорія, наша основна мета полягає в ілюстрації проблеми передобчислення міжз'єднань.

Розглянемо ієрархічну ПЛІС «APEX», показану на рисунку 3.

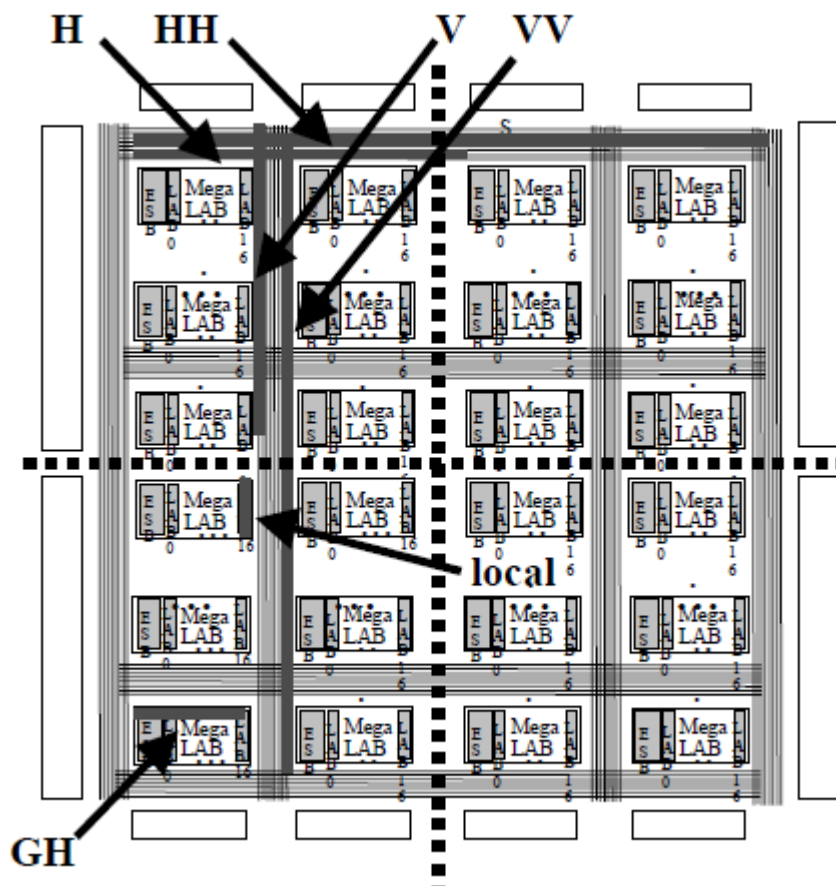


Рис 3. Ієрархічна ПЛІС "APEX"

Базовий логічний елемент (LE) являє собою LUT з 4 входами і DFF. Скупчення з 10 логічних елементів групуються в логічній матриці блока або LAB. Міжз'єднання в LAB вважається повним. Група із 16 LAB, утворює MegaLab. Сигнали, що використовують горизонтальні дроти в межах MegaLab для управління локальним міжз'єднанням з іншим LAB позначені як GH.

Прилеглі LAB, мають локальну маршрутизацію, і можуть обмінюватися даними без використання дроту GH.

Маршрутизація верхнього рівня складається з лівої і правої лінії Н, пов'язаної за допомогою двобічного, програмованого буфера сегментації. Аналогічним чином верх і низ V. Є деяка кількість Н рядків в рядку ($2 * 13$ рядків в цілому) і V рядків в стовпці (всього $16 * 2 * 2$ стовпця). Оскільки пристрій є ієрархічним, дроти Н або V повинні керуватися GH дротом в даному MegaLab.

Розглянемо розрахунки в таблиці 1. Ми знаходилися в межах 7% на дротах GH. Трохи «перемаршрутизували» в V, і злегка «недомаршрутизували» в Н.

Таблиця 1

Розрахунки для ПЛІС «APEX»

r	0		Розраху нки	Практи чні дані	Різниця %
k	2	$lab_ext = k * labsize^r$	12		
labsi ze	1 0	$GH_int = labs * lab_ext$	193		
labs	1 6	$GH_ext = k * (mlsize)^r$	105		
mlsi ze	1 60	$GH = GH_int + GH_ext$	298	279	6.7
gow s	1 3	$row_v = k * (2 * mlsize)^r$	180		
gols	2	$quad_v = rows * row_v$	2339		
qua dsize	2 080	$double_v = k * (quadsize)^r$	775		
		$total\ V = double_v + quad_v$	3113		
		$V\ per\ col = total_v / cols$	97	80	21.6
		$quad_h = k * (quadsize/2)^r$	451		
		$double_h = k * (quadsize/2)^r$	775		

		$quad_size^r$			
		$total_H = quad_h + double_h$	1226		
		$H\ per\ row = total_H / rows$	94	100	-5.7

Давайте проігноруємо, що наші початкові k і r мають апаратно задані значення і маршрути (також проігноруємо всі підключення до вбудованої пам'яті кожного MegaLab) і подивимося на аналіз чутливості нашого поточного результату. Ми, ймовірно, задали б $k = 4$, а не 2. На жаль, це призводить до 595 GH дротів, 195 дротів V і 189 H дротів - в 2 рази більше в кожному з випадків. Навіть збільшення k до 2,2 від початкового 2 збільшує нашу помилку до 20% в GH, 30% в V (H помилка знизилася до 4%). Змінювати значення r однаково проблематично. Розумним припущенням про (k, r) є $(4, 0.73)$. Це призводить до помилок в діапазоні від 30% до 80% для різних дротів.

Висновки

В теорії відсутнє розуміння того, як розробити міжз'єднання для програмованої логіки. Хоча ми можемо припускати це з використанням класичної теорії реальність така, що ми не можемо отримати навіть близькі до якісних результати, яких ми можемо досягти емпірично.

Ефективність, яка передбачається в більшості теоретичних моделей відсутня в ПЛІС. Передумова «середнього» або очікуваного прогнозу підходить для алгоритмів САПР, але не для оцінки ширини каналу.

Література:

1. E. Ahmed and J. Rose, *The Effect of LUT and Cluster Size on Deep Sub-Micron FPGA Performance and Density*, in *Proc. ACM/IEEE On FPGAs (FPGA00)*, 2000.- 57 p.
2. S.D. Brown, *Routing Algorithms and Architectures for Field-Programmable Gate Arrays*, Ph.D. Thesis, University of Toronto, 1992 -.344 p.
3. Клайв Максфілд, *Проектирование на ПЛИС. Курс молодого бойца*. М.: Додэка-21, 2007г. – 408 стр.
4. Бродин В.В., Калинин А. В. *Системы на микроконтроллерах и БИС программируемой логики*. М., 2002 год - 398 стр.

5. Вальна О.Д. *Полезные схемы с применением микроконтроллеров и ПЛИС*. М., 2006 г. - 420 стр.

6. Стешенко В.Б. *ПЛИС фирмы ALTERA: элементная база, система проектирования и языки описания аппаратуры*. М., 2007 г. - 576 стр.

7. F. Heile and A. Leaver, "Hybrid Product Term and LUTBased Architectures using Embedded Memory Blocks", in *Proc. ACM/IEEE Symp. On FPGAs (FPGA99)*, 1999 – 64 p.

8. S. Kaptanoglu, G. Bakker, A. Kundu, I. Corneillet and B. Ting. "A New High-Density and Very Low Cost Reprogrammable FPGA Architecture", in *Proc. ACM/IEEE Symp. On FPGAs (FPGA99)*, 1999 – 37 p.