УДК 621.382

Базилевич Р.П., Рибак О.Г.
ДУ "Львівська політехніка", кафедра ПЗ

АЛГОРИТМ ПОЧАТКОВОГО ПАКУВАННЯ ПРОГРАМОВАНИХ ЛОГІЧНИХ МАТРИЦЬ НА ОСНОВІ ОПТИМАЛЬНОГО ЗГОРТАННЯ СХЕМИ

Розроблено нові конструктивні алгоритми пакування програмованих логічних матриць на основі методу оптимального згортаня схеми. Експерименти підтвердили ефективність запропонованого підходу, як з точки якості отриманих розв'язків так і обчислювальних затрат.

ВСТУП

Пакування складних схем мінімальною кількістю ПЛМ (програмованих логічних матриць) є однією з важливих задач, які виникають при проектуванні сучасних засобів комп'ютерної техніки. Кількість елементів та зовнішніх зв'язків є найбільш важливими обмеженнями, які необхідно задовольняти при розбитті схем на декілька частин. Задача з математичної точки зору належить до важко розв'язуваних неполіноміальних комбінаторних оптимізаційних задач. Проблема розглядалася багатьма авторами [2-4], проте отримувана за допомогою існуючих алгоритмів якість є недостатньою. Класичний підхід до розв'язання задачі пакування у більшості випадків поділяється на дві підзадачі: одержання початкового розв'язку та подальша його оптимізація. Пропонується новий алгоритм одержання початкового розв'язку задачі пакування, який використовує метод оптимального згортання схем [1].

ФОРМУЛЮВАННЯ ЗАДАЧІ

Задано схему $\mathbf{N}=\{\mathbf{P},\mathbf{E}\}$,

де

$\mathbf{P}=\{\mathbf{p}_1, \dots, \mathbf{p}_n\}$ – множина елементів, $\mathbf{E}=\{\mathbf{e}_1, \dots, \mathbf{e}_m\}$ – множина зв'язків, (1)

На основі цієї вхідної інформації формуємо: систему множин зв'язків, інцидентних до кожного елемента:

$$\mathbf{E}(\mathbf{P})=\{\mathbf{E}(\mathbf{p}_1),\dots,\mathbf{E}(\mathbf{p}_n)\},\ (\forall \mathbf{p}\in\mathbf{P})\ [\mathbf{p}\rightarrow\mathbf{E}(\mathbf{p})=\{\mathbf{e} \mid \mathbf{e} \text{ є інцидентним до } \mathbf{p}\}]; \quad (2)$$

та систему множин елементів, інцидентних до кожного зв'язку:

$$\mathbf{P}(\mathbf{E})=\{\mathbf{P}(\mathbf{e1}),\dots,\mathbf{P}(\mathbf{e}_m)\},\ (\forall \mathbf{e}\in\mathbf{E})\ [\mathbf{e}\rightarrow\mathbf{P}(\mathbf{e})=\{\mathbf{p} \mid \mathbf{p} \text{ є інцидентним до } \mathbf{e}\}]; \quad (3)$$

Системи (3) та (4) є взаємно відповідними: $\mathbf{P}\leftrightarrow\mathbf{E}$.

Початкові параметри зазвичай містять певні характеристики вхідних елементів та зв'язків, які повинні бути враховані при прийнятті рішення (такі як площа елементів або затримка в розповсюдженні сигналів для зв'язків). Звичайно найбільш важливими обмеженнями є кількість елементів та зовнішніх зв'язків, які не можуть перевищити заданого значення:

$$\mathbf{n}_i \le \mathbf{n}_{i\,max},\quad \mathbf{m}_i^{ex} \le \mathbf{m}_i^{ex}{}_{max}\ ; \quad (4)$$

Необхідно отримати таке розбиття $\mathbf{P}^*$ множини елементів $\mathbf{P}$, щоб мінімізувати загальне число частин:

$$\mathbf{P} \to \mathbf{P}^* = \{\mathbf{P_1}, \ldots, \mathbf{P_k}\}, \quad \mathbf{k} \to \mathbf{min}, \tag{5}$$

при забезпеченні виконання заданих обмежень:

$$(\forall \mathbf{P}^i \in \mathbf{P}^*)\ [(\mathbf{n_i} \le \mathbf{n_{i\ max}}) \& (\mathbf{m_i}^{\mathbf{ex}} \le \mathbf{m_i}^{\mathbf{ex}}{}_{\mathbf{max}})].$$

АЛГОРИТМ ОДЕРЖАННЯ ПОЧАТКОВОГО РОЗВ'ЯЗКУ

Подамо довільну групу елементів схеми як кластер

$$\mathbf{C_i} = \{\mathbf{P_i}, \mathbf{E_i}^{\mathbf{in}}, \mathbf{E_i}^{\mathbf{ex}}\},$$

де $\mathbf{P_i}$ – множина елементів кластера; $\mathbf{E_i}^{\mathbf{in}}$ – множина цілком внутрішніх зв'язків кластера, тобто таких зв'язків, що охоплюють лише елементи схеми: $(\forall \mathbf{e} \in \mathbf{E_i}^{\mathbf{in}})\ [\mathbf{P(e)} \subseteq \mathbf{P_i}]$; $\mathbf{E_i}^{\mathbf{ex}}$ – множина зовнішніх зв'язків кластера, тобто таких, що охоплюють як елементи кластера $\mathbf{C_i}$, так і елементи інших кластерів одночасно: $(\forall \mathbf{e} \in \mathbf{E_i}^{\mathbf{ex}})\ (\exists \mathbf{p_a}, \mathbf{p_b} \in \mathbf{P(e)})\ [\ \mathbf{p_a} \in \mathbf{P_i}, \mathbf{p_b} \notin \mathbf{P_i}]$.

Для одержання початкового розв'язку реалізована стратегія ітераційного виділення підсхем з перебудовою дерева:

Крок 1. Множина частин є порожньою $\mathbf{P}^* = \varnothing$.

Крок 2. Для елементів схеми $\mathbf{N}$, що не увійшли в розв'язок будуємо дерево згортання. На початку розглядаємо елементи схеми як вершини дерева першого рівня, тобто кожен елемент схеми утворює одну групу (кластер). Для всіх пар вершин дерева $(\mathbf{t_i}, \mathbf{t_j})$, яким відповідають кластери $\mathbf{C_i}$ та $\mathbf{C_j}$, існує хоча б один безпосередній зв'язок між двома елементами кластерів $\mathbf{C_i}$ та $\mathbf{C_j}$ (тобто $(\exists \mathbf{e} \in \mathbf{E} : \exists \mathbf{p_a}, \mathbf{p_b} \in \mathbf{P(e)})\ [\ \mathbf{p_a} \in \mathbf{P_i}, \mathbf{p_b} \in \mathbf{P_j},]$, причому цей зв'язок охоплює не більше ніж $\mathbf{l}$ елементів схеми (використання параметра $\mathbf{l}$ дозволяє керувати процесом згортки для зменьшення обчислювальних затрат), обчислимо значення критерію згортання:

$$\Delta = \mathbf{m'_{ij}} - \mathbf{m''_{ij}},$$

де $\mathbf{m'_{ij}}$ – кількість додаткових внутрішніх зв'язків кластера $\mathbf{C_{ij}}$, що одержуються шляхом об'єднання кластерів

$$\mathbf{C_i} \text{ та } \mathbf{C_j}, \quad \mathbf{m'_{ij}} = |\ \delta \mathbf{E_{ij}}^{\mathbf{in}}\ |,$$

де $(\forall \mathbf{e} \in \Delta \mathbf{E_{ij}}^{\mathbf{in}})\ [\delta \mathbf{E_{ij}}^{\mathbf{in}} = (\mathbf{E_i}^{\mathbf{ex}} \cap \mathbf{E_j}^{\mathbf{ex}}) \& (\mathbf{P(e)} \subset \mathbf{P_{ij}})]$; $\mathbf{m''_{ij}}$ – кількість зовнішніх зв'язків кластера $\mathbf{C_{ij}}$, $\mathbf{m''_i} == |\ \mathbf{E_{ij}}^{\mathbf{ex}}\ |$, де $(\forall \mathbf{e} \in \mathbf{E_{ij}}^{\mathbf{ex}})\ [\mathbf{E_{ij}}^{\mathbf{ex}} = (\mathbf{E_i}^{\mathbf{ex}} \cup \mathbf{E_j}^{\mathbf{ex}}) \& (\mathbf{P(e)} \backslash \mathbf{P_{ij}} \ne \varnothing)]$.

Впорядковуємо пари вершин за спаданням значення критерію згортання. Починаючи з пари, що має найбільше значення критерію, об'єднуємо λ% пар (за значенням критерію згортання), утворюючи новий рівень дерева. Процес побудови продовжуємо доки всі елементи не увійдуть в один кластер. Використання параметра λ дозволяє керувати висотою дерева згортання, а отже і швидкістю побудови дерева.

Крок 3. Серед вершин дерева згортання вибираємо найбільшу за кількістю елементів вершину $\mathbf{v}$, для якої обмеження на кількість елементів не порушується. Цю вершину будемо використовувати як базову для одержання чергової частини розв'язка. На основі (4) визначаємо верхні обмеження ітераційного процесу одержання розв'язка як :

$$\mathbf{n_t} = \max(\mathbf{n_{i\ max}}, |\mathbf{P_t}|), \quad \mathbf{m_t}^{\mathbf{ex}} = \max(\mathbf{m_i}^{\mathbf{ex}}{}_{\mathbf{max}}, |\mathbf{E_t}^{\mathbf{ex}}|); \tag{6}$$

Крок 3.1. Якщо для вершини $\mathbf{v}$ порушується хоча б одне з обмежень, то для вилучення елементів переходимо на крок 3.2, в інакшому випадку переходимо до добирання елементів на крок 3.3.

Крок 3.2. Будуємо дерево згортання, як описано на кроці 2 лише для елементів піддерева вершини **v**. Серед кластерів нового дерева вилучаємо такий, у якому відношення кількості зовнішніх зв'язків до кількості елементів було максимальним. Переходимо на крок 3.1.

Крок 3.3. Будуємо дерево згортання, як описано на кроці 2 без врахування елементів піддерева **v**. Серед вузлів нового дерева виберемо такий вузол, для якого після приєднання до піддерева **v** відношення приросту кількості зовнішніх зв'язків (відносно характеристик піддерева **v**) до приросту кількості елементів буде мінімальним і обмеження (6) не порушуються. Знайдене піддерево долучаємо до піддерева **v** і переходимо на крок 3.1. Якщо не знайдено вершини, яка б задовольняла вищенаведені умови і поточне піддерево задовольняє обмеження задачі, то вважаємо, що чергова частина розв'язку знайдена: елементи піддерева **t** долучаємо до розв'язку $\mathbf{P}^*$ і переходимо до виділення наступної частини (крок 2.). Якщо у розв'язок $\mathbf{P}^*$ увійшли всі елементи початкової схеми **N**, то рішення знайдено.

Одержана таким чином множина частин $\mathbf{P}^*$ і буде початковим розв'язком задачі пакування.

ЕКСПЕРИМЕНТАЛЬНІ РЕЗУЛЬТАТИ

Для експериментального дослідження взято 8 описаних в [4] тестів. В таблицях 1 та 2 наведені характеристики тестів та результати, отримані методами RFM, SC та ОЗС (описаним в цій статті). При розв'язуванні задач в таблиці 1 використовуються обмеження з 64 елементів та 58 зовнішні зв'язки для кожної з підсхем, для нашого методу параметри **l** та λ відповідно дорівнюють ∞ та 0. Для задач з таблиці 2 використано обмеження з 320 елементів та 144 зовнішні зв'язки ($\mathbf{l} = 5$ та $\lambda = 20\%$). Як видно з таблиць, одержані результати є відповідними до результатів RFM, SC алгоритмів. Зазначимо, що RFM та SC алгоритми є оптимізуючими, що і обумовлює одержання в деяких випадках кращих результатів.

Таблиця 1

Тест	Елементів	Зв'язків	RFM	SC	ОЗС
C3540	373	569	6	6	8
C5315	531	936	11	12	12
C7552	611	1057	11	11	12
C6288	833	1472	14	14	14

Таблиця 2

Тест	Елементів	Зв'язків	RFM	SC	ОЗС
S15580	842	1265	4	3	4
S13207	915	1377	7	6	6
S38417	2221	3216	12	10	10
S38584	2904	3884	17	14	14

ВИСНОВКИ

Розроблений алгоритм забезпечує одержання початкового розв'язку задач пакування з достатньою точністю. Результати експериментів підтвердили доцільність використання для розв'язування задач даного класу методу оптимального згортання схеми. Для поліпшення

розв'язків, що одержуються ORTM алгоритмом, можуть бути використані довільні ітераційні підходи. Ефективний новий алгоритм на основі обмінів кластерів запропонований в [5].

1. Базилевич Р.П. Декомпозионные и топологические методы автоматизованого проектирования электронних устройств. Львів, 1981. 2. Ober U., Glesner M. Multiway netlist partitioning onto FPGA-based board architectures. Proc. European Design Automation Conference with EURO-VHDL (EURO-DAC'95), 1995. Pp.150-155. 3. Kuznar R., Brglez F., Kozminski K. Cost minimization of partitioning into multiple devices. IEEE/ACM 30th Design Automation Conference, 1993.5. Nan-Chi Chou, Lung-Rtien Liu, Chung-Kuan Cheng, Wei-Jin Dai, Rodney Lindelof. Local Ration Cut and Set Covering Partitioning for Huge Logic Emulation System. IEEE Trans. On CAD of Integrated Circuit and System, vol.14, No.9, Sept.1995, pp:1085-1092. 6. Базилевич Р.П., Рибак О.Г. Оптимізація пакування програмованих логічних матриць на основі оптимального згортання схем. В цьому віснику.

УДК 621.326

Березко Л.О., Ірисов О.Є.
ДУ "Львівська політехніка", кафедра ЕОМ

ЗАСОБИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ОБРОБКИ НЕЧІТКИХ ДАНИХ

Засоби обробки нечіткої, неповної та невизначеної інформації дають відчутні переваги в ситуаціях, коли вимагається отримати достатньо прийнятні результати, грунтуючись на нечітких вихідних даних. Впровадження цих засобів вимагає наявності в комп'ютерних системах відповідного апаратного та програмного забезпечення. Пропонується один з можливих варіантів реалізації підтримки нечіткої арифметики на RISC-процесорі.

Вступ. Бінарні операції з нечіткими даними можна реалізувати, застосовуючи інтервальну арифметику на рівневих множинах (α-зрізах) або min/max згортку [1,2,3]. Перший метод порівняно з другим дозволяє сильно скоротити кількість необхідних операцій та зменшити об'єм пам'яті для збереження операндів оскільки, при виконанні обчислень кількість відліків функції належності залишається постійною. Кількість α-зрізів, що описують нечітку величину, вибирається, виходячи з необхідної точності подання та допустимого часу обробки даних.

Хоча перший метод набагато швидший, ніж другий, він все одно забезпечує значно повільнішу обробку даних, ніж традиційна машинна арифметика. Для підвищення швидкості виконання нечітких операцій існують такі можливості: розробити сопроцесор (додатковий модуль інтервальної арифметики); додати до існуючого процесора невеликий набір інструкцій, що забезпечує інтервальну арифметику (в цьому випадку ставиться мета - кожна інтервальна операція повинна виконуватись з використанням не більше двох звичайних операцій) [4].