УДК 681.32

Н.Т. Процай

Национальный технический университет «ХПИ», Харьков

МИНИМИЗАЦИЯ ДЛИНЫ ТОПОЛОГИЧЕСКИХ ПУТЕЙ ПРОХОЖДЕНИЯ СИГНАЛОВ ПО МАТРИЦЕ ПРОЦЕССОРОВ ПРИ РАЗМЕЩЕНИИ СИСТЕМЫ БУЛЕВЫХ УРАВНЕНИЙ В PROGRAMMABLE UNLIMITED SYSTEMS

В статье предложен метод решения задачи минимизация общей длины топологических путей прохождения сигналов по матрице процессоров при размещении системы булевых уравнений в Programmable Unlimited Systems, в целях максимального использования параллельного и конвейерного режимов работы мультипроцессора для получения решения за минимально возможное время. Приведены топологическая и математическая модели структуры процессора. Проанализированы достоинства и недостатки метода. Очерчены перспективы дальнейших исследований в данном направлении.

***Ключевые слова**: мультипроцессор, секвенсер, Programmable Unlimited Systems, планарная структура, волновой алгоритм, булевы уравнения, граф, минимизация пути на графе.*

Введение

Programmable Unlimited Systems – система аппаратного ускорения верификации цифровых устройств. Система позволяет сократить время моделирования и верификации цифровых устройств прироста интеграции систем на кристаллах; снизить объем использования оперативной памяти при моделировании проекта; уменьшить интервал времени выхода готового продукта на рынок электронных технологий. Система существенно повышает быстродействие решения систем булевых уравнений большой размерности, путём разработки однобитового мультипроцессора, имплементируемого в ASIC с матрично-сферической организацией взаимных связей и минимальным набором команд. Он позволяет осуществлять параллельную, последовательную и конвейерную обработку булевых уравнений, записанных в базисе операций AND, OR, NOT, XOR.

Мультипроцессор экономичен в аппаратурном исполнении: для обработки системы уравнений, насчитывающей 20 миллионов вентилей, необходимо иметь всего 256 Мбайт оперативной памяти. [1 – 3].

Область применения. Электронные схемы, компоненты и оборудование. Электронная техника, конструирование и моделирование. Данная система может представлять интерес для проектирования систем управления в таких областях человеческой деятельности, как индустрия, медицина, защита информации, геология, прогнозирование погоды, искусственный интеллект, космонавтика. Это представляет особый интерес для цифровой обработки данных, распознавания образов, криптоанализа. Одним из основных приложений в EDA (Electronic Design Automation) технологиях является эмуляция больших проектов, имплементируемых в ASICs и FPGA [1, 4, 5].

Цикл проектирования цифрового устройства по данной технологии состоит из 3-х основных частей:

1) преобразование проекта в систему булевых уравнений;

2) распределение булевых уравнений среди процессоров;

3) эмуляция проекта в мультипроцессоре [1].

При поиске оптимального размещения уравнений по процессорам с учетом топологии мультипроцессора в [1] сформулированы основные критерии размещения.

Одним из критериев является минимизация общей длины топологических путей прохождения сигналов по матрице процессоров при размещении системы булевых уравнений.

Цель работы - минимизация общей длины топологических путей прохождения сигналов по матрице процессоров при размещении системы булевых уравнений в Programmable Unlimited Systems.

Для достижения поставленной цели необходимо решить следующие ***задачи*:**

1. Рассмотреть структуру процессора Programmable Unlimited Systems

2. Построить математическую модель мультипликативного процессора и сформулировать задача минимизации общей длины топологических путей прохождения сигналов по матрице процессоров при размещении системы булевых уравнений

3. Рассмотреть волновой алгоритм нахождения минимального пути на графе

4. Модифицировать волновой алгоритм согласно особенностей строения Programmable Unlimited Systems.

Структура процессора Programmable Unlimited Systems

Для оптимизации распределения булевых уравнений по частям мультипроцессора необходимо иметь представление о его структуре.

Архитектура Programmable Unlimited Systems представляет собой матрицу параллельных процессоров, каждый из которых связан с восемью другими линиями передачи данных. Структура и модель обработки уравнений разработаны доктором Stanley Hyduke и носят название “Compiler Synchronized Parallel-processor Network-based Logic Device” – сеть параллельных компилятивно синхронизированных процессоров.

Планарная структура Programmable Unlimited Systems представлена на рис. 1 [2, 3, 5].

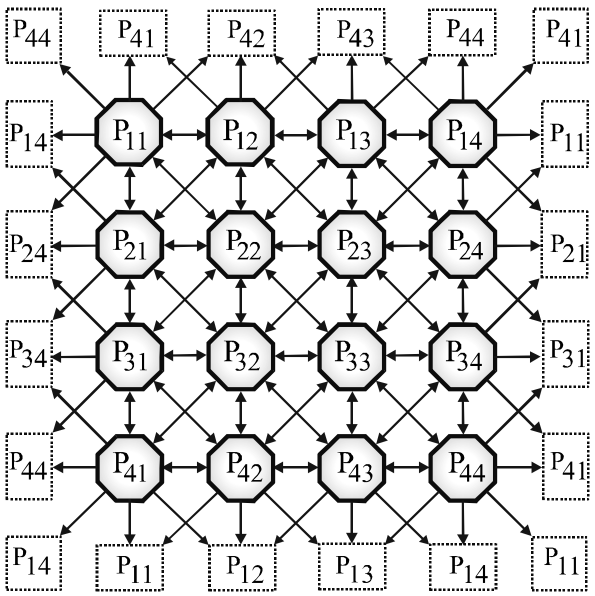

Рис. 1. Планарная структура процессора

Такое представление удобно для решения оптимизационных задач на плоскости, к которым можно отнести задачи размещения булевых уравнений по процессорам, минимизацию функциональных связей между секвенсерами при эмуляции уравнений.

Построение математической модели мультипликативного процессора

Для построения формальной математической модели мультипликативного процессора удобно использовать теорию графов.

При этом планарную структуру процессора, изображенного на рис. 1 интерпретируют неориентированным графом, представленным на рис. 2, в котором каждому процессору ставят в соответствие вершину графа, а межсоединениям – его ребра.

Граф G задается множеством точек или вершин $X = \{x_1, x_2, ..., x_m\}$ и множеством линий или ребер $A = \{a_1, a_2, ..., a_n\}$, соединяющих между собой все или часть этих точек. Ребра будем обозначать парой, состоящей из начальной x_i и конечной x_j вершин (т.е. двумя концевыми вершинами ребра) $a = (x_i, x_j), x_i \neq x_j$. Т.о., граф G полностью задается и обозначается парой (X,A).

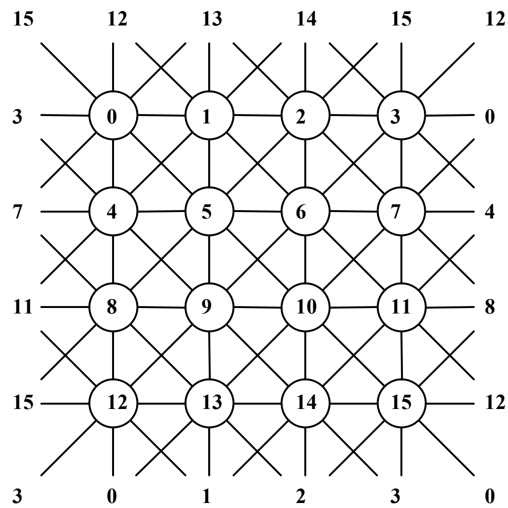

Рис. 2. Графовое представление планарной структуры процессора

Граф $G = (X,A)$, изображенный на рис.2, задается множествами:

$$X = \{x_0, x_1, ..., x_{15}\};$$

$$A = \{(x_0,x_1),(x_0,x_5),(x_0,x_4),...,(x_1,x_2),(x_1,x_6),...,(x_{14},x_{15})\}.$$

Другое описание графа G состоит в задании множества вершин X и соответствия Γ, которое показывает, как между собой связаны вершины. Граф в этом случае обозначается парой $G = (X,\Gamma)$.

Так, например, для графа на рис. 2 имеем

$$\Gamma(x_1) = \{x_2, x_6, x_5, x_4, x_0, x_{12}, x_{13}, x_{14}\},$$

т.е. вершины $x_2, x_6, x_5, ..., x_{14}$ являются конечными вершинами неориентированных ребер, у которых начальной вершиной является x_1. Такие соответствия можно задать для каждой из шестнадцати вершин графа, изображенного на рис.2.

Поскольку $\Gamma(x_i)$ представляет собой множество таких вершин $x_j \in X$ для которых в графе G существует ребро (x_i, x_j), то через $\Gamma^{-1}(x_i)$ естественно обозначить множество вершин x_k, для которых в G существует ребро (x_k, x_i). Отношение $\Gamma^{-1}(x_i)$ называется обратным соответствием. Очевидно, что для неориентированного графа, каким является граф на рис. 2, $\Gamma^{-1}(x_i) = \Gamma(x_i)$ для всех $x_i \in X$.

Число ребер, которые имеют вершину x_i своей начальной вершиной, называется полустепенью исхода вершины x_i, и, аналогично, число ребер, которые имеют x_i своей конечной вершиной, называется полустепенью захода вершины x_i. Очевидно, что сумма полустепеней исхода всех вершин ориентированного графа, а также сумма полустепеней захода всех вершин равны общему числу дуг графа G. Для неориентированного графа $G = (X, \Gamma)$, степень вершины x_i определяется с помощью соотношения

$$d(x_i) \equiv |\Gamma(x_i)|.$$

Маршрутом в неориентированном графе является последовательность ребер, в которой конечная вершина всякого ребра, отличная от последней, является начальной вершиной следующего. Так на рис. 2 последовательности ребер:

$$(x_8, x_5), (x_5, x_6), (x_6, x_7), (x_7, x_3); \tag{1}$$

$$\begin{gathered}(x_{14}, x_9), (x_9, x_4), (x_4, x_5), (x_5, x_6), (x_6, x_9),\\ (x_9, x_4), (x_4, x_1)\end{gathered} \tag{2}$$

являются маршрутами.

Маршрут можно изображать как с помощью последовательности ребер, как это сделано выше, так и с помощью последовательности вершин, так для маршрутов (1) и (2) соответственно имеем:

$$(x_8, x_5, x_6, x_7, x_3);$$

$$(x_{14}, x_9, x_4, x_5, x_6, x_9, x_4, x_1).$$

Такое представление часто оказывается более полезным в тех случаях, когда осуществляется поиск простых цепей.

Для алгебраического задания невзвешенных графов (графов, ребра которых не имеют весов) или в том случае, когда веса всех ребер равны, а именно таким и является граф, изображенный на рис. 2, удобно использовать матрицу смежности (связности) размера $n \times n$

$$C = [c_{ij}],$$

где n количество вершин графа, т.е. мощность множества X:

$c_{ij} = 1$, если в графе существует ребро (x_i, x_j),

$c_{ij} = 0$, если в графе нет ребра (x_i, x_j).

Очевидно, что матрица $C = [c_{ij}]$ симметрична, если граф неориентированный, и может быть не симметричной в противном случае. При этом полагаем, что $c_{ii} = 0$, т.е. в графе нет петель. Матрица смежности полностью определяет структуру графа.

Например, сумма всех элементов строки x_i матрицы дает полустепень исхода вершины x_i, а сумма элементов столбца x_i - полустепень захода вершины x_i. Множество столбцов, имеющих 1 в строке x_i, есть множество $\Gamma(x_i)$, а множество строк, которые имеют 1 в столбце x_i, совпадает с множеством $\Gamma^{-1}(x_i)$.

Принимая во внимание все сказанное выше, задача минимизации общей длины топологических путей прохождения сигналов по матрице процессоров при размещении системы булевых уравнений формулируется следующим образом:

пусть задан граф $G = (X, \Gamma)$*; необходимо найти маршрут от заданной начальной вершины* $s \in X$ *до заданной конечной вершины* $t \in X$*, количество промежуточных вершин (и, соответственно, ребер) в котором минимально, при условии, что такой маршрут существует.*

Длиной (или мощностью) маршрута μ называется число ребер, входящих в него. Каждое ребро рассматривается столько раз, сколько оно встречается в данном маршруте.

Так длины маршрутов (1) и (2) соответственно равны $\mu_1 = 4, \mu_2 = 7$ [7].

Для решения задачи минимизация общей длины топологических путей прохождения сигналов по матрице процессоров при размещении системы булевых уравнений существует эффективный алгоритм, имеющий условно линейную сложность как по числу вершин, так и по числу ребер - волновой алгоритм (другие названия – алгоритм Ли, поиск в ширину, алгоритм степного пожара) [7].

Волновой алгоритм минимизации пути на графе

Основные принципы построения кратчайших маршрутов с помощью волнового алгоритма сводятся к следующему.

Все процессоры (вершины графа) подразделяют на занятые и свободные. Занятыми считаются процессоры, в которых уже выполняются соответствующие булевы операции. Каждый раз при нахождении кратчайшего маршрута между двумя заданными вершинами можно использовать лишь свободные процессоры.

Алгоритм находит один из минимальных маршрутов (маршрутов, проходящих через минимальное количество вершин) в графе $G = (X,A)$, заданном матрицей связности C. Маршрут ищется из вершины номер s к вершине номер t.

Волновой алгоритм:

1) каждой вершине x_i приписывается целое число $T(x_i)$ – волновая (временная) метка и массив вершин $P(x_i)$, (начальное значение $T(x_i) = -1$ и $P(x_i) = [\ ]$);

2) заводятся два списка OldFront (OF) и NewFront (NF) (старый и новый "фронт волны"), а также переменная T (текущее время);

3) $OF := \{s\}$; $NF := \{\ \}$; $T(s) := 0$; $T := 0$;

4) для каждой из вершин x_i, входящих в OF, просматриваются инцидентные (смежные) ей вершины x_j, и если $T(x_j) = -1$, то

$$T(x_j) := T + 1,$$

$$NF := NF + \{x_j\},$$

$$P(x_j) = \left[P(x_i) + \{x_j\}\right];$$

5) если $NF := \{\ \}$, то маршрут не существует, переход к *шагу 8*;

6) если $t \in NF$ (т.е. одна из вершин x_j совпадает с t), то найден кратчайший маршрут между s и t с $T(t) := T + 1$ промежуточными ребрами, т.е. длины $T + 1$; переход к *шагу 8*;

7) $OF := NF$; $NF := \{\ \}$; $T := T + 1$; возврат к *шагу 4*.

8) восстанавливаем маршрут с помощью массива P.

Достоинства метода - простота, надёжность, 100% нахождение самого короткого маршрута. Существенными недостатками волнового алгоритма являются малое быстродействие и большой объем оперативной памяти ЭВМ, необходимый для хранения информации о текущем состоянии всех вершин графа. Попытки устранить указанные недостатки привели к модификации волнового алгоритма.

Модификации волнового алгоритма

В случае острой нехватки памяти, может применяться усечённое кодирование волны. Т.е. первая волна имеет номер 1, вторая - 2, третья - снова 2, четвёртая - 1 и т.д.

На кодировку одного элемента потребуется два бита. При поиске маршрута ищем соседние ячейки памяти в том же порядке (... 1 1 2 2 1 1 2 2 1 1 2 2...). Однако в данном методе ни о каких диагональных перемещениях не может быть и речи. А в нашем графе диагональные перемещения используются.

В лучевом алгоритме, предложенным Л. Б. Абрайтисом, выбор вершин для определения маршрута между начальной и конечной вершинами, производят по заранее заданным направлениям, подобным лучам. Это позволяет сократить число просматриваемых алгоритмом вершин, а следовательно, и время на анализ и кодировку их состояния, однако приводит к снижению вероятности нахождения пути сложной конфигурации.

Работа алгоритма заключается в следующем. Задается число лучей, распространяемых из вершин графа s и t, а также порядок присвоения путевых координат (обычно число лучей для каждой вершины принимается одинаковым). Лучи $s(1)$, $s(2),\ldots,$ $s(n)$ и $t(1)$, $t(2),\ldots,$ $t(n)$ считают одноименными, если они распространяются из одноименных источников s или t. Лучи $s(i)$ и $t(i)$ являются разноименными по отношению друг к другу. Распространение лучей производят одновременно из двух вершин до встречи двух разноименных лучей в некоторой вершине p. Путь проводится из ячейки p и проходит через ячейки, по которым распространялись лучи.

Обычно с помощью лучевого алгоритма удается построить до 70-80% маршрутов, остальные строят, используя волновой алгоритм или вручную. В нашем же случае необходимо получить 100% результат.

Т.о., рассмотрев вышеизложенные алгоритмы и проанализировав имеющиеся ресурсы, для нахождения кратчайшего маршрута между двумя заданными точками, рассмотрим метод, лишенный вышеперечисленных недостатков и применим в нашем случае.

В данном методе источниками волн являются начальная и конечная вершины маршрута. При этом на каждом k-м шаге поочередно строят соответствующие фронты первой и второй волн, распространяющихся из этих вершин. Процесс продолжается до тех пор, пока какая-либо вершина из фронта первой волны не попадет на фронт второй волны или наоборот. Маршрут состоит из двух участков - от точки встречи волн до старта и до финиша.

Оценим число вершин, просматриваемых на этапе распространения волны, при использовании в качестве источников двух вершин. Пусть расстояние между этими вершинами R. Тогда для первого случая в момент достижения волной конечной вершины, площадь просмотренной окрестности имеет величину $Q^{(2)} = \pi R^2$. Для второго случая в момент встречи фронтов двух волн площадь просмотренной окрестности

$$Q^{(2)} = 2\pi(0,5R)^2 = 0,5\pi R^2.$$

Т. о., при использовании метода встречной волны просматриваемая площадь, а следовательно, и время, затрачиваемое на этапе распространения волны, уменьшаются примерно вдвое.

Выводы

В статье предложен метод решения задачи минимизация общей длины топологических путей прохождения сигналов по матрице процессоров при размещении системы булевых уравнений в Programmable Unlimited Systems. Метод позволяет в два раза сократить время, затрачиваемое на этапе распространения волны.

Недостатком метода является необходимость выделения дополнительного разряда памяти на каждую рабочую ячейку поля для хранения информации о принадлежности её к первой или второй волне. Однако выигрыш в повышении быстродействия восполняет указанный недостаток, поэтому данный метод используют во всех случаях, когда это позволяет объем оперативной памяти ЭВМ.

В перспективах дальнейшего развития следовало бы модифицировать существующие методы или разработать новые, который бы решил не только вопрос быстродействия метода, но и вопрос экономии объема оперативной памяти.

Список литературы

1. Сферический мультипроцессор PRUS для решения булевых уравнений / С.М. Гайдук, В.И. Хаханов, В.И. Обризан, Е.А. Каменюка // Радиоэлектроника и информатика. – 2004. – № 4(29). – С. 107−116.

2. Хаханов В.И. Мультипроцессор для анализа информационного пространства. 1. Архитектура логического ассоциативного мультипроцессора / В.И. Хаханов, Е.И. Литвинова // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2011. – № 1 (14). – С. 95-108.

3. Хаханов В.И. Мультипроцессор для анализа информационного пространства. 2. инфраструктура векторно-логического анализа / В.И. Хаханов, С.В. Чумаченко // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2011. – № 2 (15). – С. 108-114.

4. Бондаренко М.Ф. Структура логического ассоциативного мультипроцессора / М.Ф. Бондаренко, В.И. Хаханов, Е.И. Литвинова // Автомат. и телемех. – 2012. – № 10. – С. 71-92.

5. Hahanova I.V., "Quantum Processor for the Optimal Coverage", 2013 12th International Conference on the Experience of Designing and Application of Cad Systems in Microelectronics (Cadsm 2013), IEEE, 2013. – P. 132–137, ISBN: 978-1-4673-6461-4.

6. Cohen A.A. Addressing architecture for Brain-like Massively Parallel Computers // Euromicro Symposium on Digital System Design (DSD'04). 2004. – P. 594−597.

7. Кристофидес Н. Теория графов. Алгоритмический подход / Н. Кристофидес. – М. Мир, 1978. – 357 с.

Поступила в редколлегию 23.01.2015

Рецензент: д-р техн. наук проф. Л. М. Любчик, Национальный технический университет «Харьковский политехнический институт», Харьков.

МІНІМІЗАЦІЯ ДОВЖИНИ ТОПОЛОГІЧНИХ ШЛЯХІВ ПРОХОДЖЕННЯ СИГНАЛІВ ПО МАТРИЦІ ПРОЦЕСОРІВ ПРИ РОЗМІЩЕННІ СИСТЕМИ БУЛЕВИХ РІВНЯНЬ У PROGRAMMABLE UNLIMITED SYSTEMS

Н.Т. Процай

В статті запропоновано метод розв'язання задачі мінімізації загальної довжини топологічних шляхів проходження сигналів по матриці процесорів при розміщенні системи мулевих рівнянь у Programmable Unlimited Systems з метою максимального використання паралельного та конвеєрного режимів роботи мультипроцесора для отримання розв'язку за мінімально можливий час. Наведено топологічна та математична моделі структури процесора. Проаналізовано достоїнства та недоліки метода. Окреслені перспективи подальших досліджень у зазначеному напрямку

Ключові слова: *мультипроцесор, секвенсер, Programmable Unlimited Systems, планарна структура, хвильовий алгоритм, булеві рівняння, граф, мінімізація шляху на графі.*

MINIMIZATION OF TOPOLOGICAL PATH LENGTH OF SIGNALS PASSING TROUGH THE PROCESSORS MATRIX IN THE PROCESS OF BOOLEAN EQUATIONS SYSTEM LOCATION AT PROGRAMMABLE UNLIMITED SYSTEMS

N.T. Protsai

In the article is proposed a method solution of the problem minimization length topological tract of signals passing from the processors matrix location boolean equations system at programmable unlimited systems. This method is useful, when it is need to use maximally parallel and conveyor operating modes of multiprocessor to get solution of the problem in the shortest possible time. There are topology and mathematical models of processor structure. Dignities and shortcomings of method are analyzed. There are outlined perspectives of discovery at this direction.

Keywords: *multiprocessor, sequencer, Programmable Unlimited Systems, planar structure, wave algorithm, Boolean equations system, graph, minimization of path in the graph.*