ПРОЕКТИРОВАНИЕ СИСТОЛИЧЕСКИХ МАТРИЧНЫХ ПРОЦЕССОРОВ С ИСПОЛЬЗОВАНИЕМ РАСЩЕПЛЕНИЯ И РАЗДЕЛЬНОГО ТАЙМИРОВАНИЯ МАКРООПЕРАЦИЙ

Е. В. Адуцкевич

Введение

Систолический вычислитель (СВ) – это спецпроцессор, ориентированный на СБИС-реализацию, предназначенный для решения одного класса однотипных задач, для которых он спроектирован. СВ состоит из однотипных процессорных элементов (ПЭ), связанных между собой локальными связями (не зависящими от размера задачи). СВ широко применяются для решения различных векторно-матричных задач.

В имеющихся на данный момент работах [1–4] приводятся методы проектирования СВ на основе единого таймирования макроопераций алгоритма. В данной работе описан метод проектирования СВ на основе расщепления и раздельного таймирования макроопераций алгоритма. Теория расщепления и раздельного таймирования макроопераций алгоритмов появилась недавно [5; 6], и до настоящего времени не было разработано технологии проектирования спецпроцессоров с применением этой теории.

Применение вышеуказанной теории при проектировании СВ позволяет расширить класс алгоритмов, для которых возможно спроектировать СВ и уменьшить время выполнения задачи.

1. Математическая модель алгоритма

Будем рассматривать алгоритмы, заданные в виде систем однородных рекуррентных уравнений (ОРУ), которые определяются следующим образом:

$$x^{(k)}(v) = F_\lambda^{(k)}(x^{(1)}(v-\varphi^{(1,k)}), x^{(2)}(v-\varphi^{(2,k)}), \dots, x^{(K)}(v-\varphi^{(K,k)})),$$

$$1 \le k \le K, \quad 1 \le \lambda \le \Lambda \;\; K, \Lambda \in N, \quad v \in V_\lambda \subset Z^d,$$

$$V_{\lambda_1} \cap V_{\lambda_2} = \varnothing,\ 1 \le \lambda_1, \lambda_2 \le \Lambda,\ \lambda_1 \ne \lambda_2,\ \bigcup_{\lambda=1}^{\Lambda} V_\lambda = V, \qquad (1)$$

где $x^{(1)}, x^{(2)}, \dots, x^{(K)}$ – переменные алгоритма; $F_\lambda^{(1)}, F_\lambda^{(2)}, \dots, F_\lambda^{(K)}$ – функции K переменных (не обязательно строго зависящие от всех K

переменных), описывающие базовые операции алгоритма; $\varphi^{(i,j)}$, $1 \le i$, $j \le K$, – d-мерные векторы зависимостей с целочисленными координатами. Множество V называется областью вычислений алгоритма (1).

Базовой операцией алгоритма (1), приписанной точке $v \in V_\lambda$, назовем правило вычисления переменной $x^{(k)}$ в этой точке. Базовой макрооперацией алгоритма (1), приписанной точке $v \in V_\lambda$, назовем правило вычисления всех переменных $x^{(k)}$, $1 \le k \le K$, в этой точке.

Для выявления связей между операциями алгоритма используют математическую модель алгоритма, основанную на построении графа зависимостей алгоритма, который строится следующим образом. Множеству операций алгоритма ставят во взаимно однозначное соответствие некоторое множество точек V (вершины графа) d-мерной целочисленной решетки Z^d. Если аргумент операции есть результат выполнения другой операции, то соответствующие вершины соединяют дугой, исходящей из вершины, откуда берется результат. Такие вершины графа называют информационно зависимыми. Множество дуг, определяемое множеством пар информационно связанных вершин, обозначают E.

Идея расщепления макроопераций заключается в том, что каждому из K наборов операций $F_\lambda^{(k)}$, $1 \le \lambda \le \Lambda$, приписанных точкам из d-мерного пространства, ставится во взаимно однозначное соответствие точки из $(d+p)$-мерного пространства путем добавления к каждой точке p координат, индивидуальных для каждого k. Таким образом, получаем систему ОРУ, в которой векторы $\varphi^{(i,j)}$, $1 \le i, j \le K$, – $(d+p)$-мерные векторы зависимостей с целочисленными координатами, и каждая переменная $x^{(k)}$ алгоритма вычисляется в своей индивидуальной подобласти $V^{(k)} \subset V$. В полученном алгоритме в каждой точке области вычислений алгоритма V выполняется одна базовая операция $F_\lambda^{(k)}$, т. е., вычисляется только одна переменная.

2. Распараллеливание алгоритма

Разбиение множества вершин графа алгоритма V на N непересекающихся подмножеств V_s таких, что дуги графа алгоритма соединяют вершины только из подмножеств с меньшим номером с вершинами только из подмножеств с большим номером, называется параллельной формой алгоритма (ПФ). Подмножества V_s называются ярусами ПФ. Операции, принадлежащие одному и тому же ярусу ПФ, могут выполняться одновременно. Число N называется высотой ПФ. Чем меньше высота ПФ, тем быстрее может решать задачу полученный спецпроцессор.

Для синхронизации ПЭ вводят понятие таймирующей функции. Ее аргументами являются вершины графа алгоритма. Таймирующая функция разбивает вершины графа алгоритма по ярусам ПФ. Значение таймирующей функции в какой-то вершине означает временной момент выполнения операций, приписанных данной вершине. При едином таймировании вводится единая для всех вершин графа алгоритма функция $t(v) = n \cdot v + t_0$, $v \in V$, $n \in Z^d$, $t_0 \in Z$. Раздельное таймирование удобнее рассматривать после расщепления. При раздельном таймировании вводится K таймирующих функций:

$$t^{(k)}(v) = n^{(k)} \cdot v + t_0{}^{(k)}, \quad v \in V^{(k)}, \quad n^{(k)} \in Z^{d+p}, \; t_0{}^{(k)} \in Z, \; 1 \le k \le K.$$

Ограничением на единую таймирующую функцию алгоритма является условие принадлежности вектора n конусу допустимых направлений графа, который определяется следующим образом: $K(G) =$ $= \{n \in Z^d \mid n \cdot \varphi^{(i,j)} > 0, \; \forall \varphi^{(i,j)}, \; 1 \le i, j \le K\}$. Единая таймирующая функция для алгоритма существует тогда и только тогда, когда конус допустимых направлений не пуст. Ограничениями на таймирующие функции при раздельном таймировании являются условия:

$$t^{(j)}(v) > t^{(i)}(v - \varphi^{(i,j)}), \quad v \in V^{(j)}, \; v - \varphi^{(i,j)} \in V^{(i)}, \; (v - \varphi^{(i,j)}, v) \in E, \; 1 \le i, j \le K.$$

Существуют случаи, когда конус допустимых направлений алгоритма пуст, т. е. единого таймирования для такого алгоритма не существует, но раздельное таймирование может быть найдено.

Высота ПФ при едином таймировании определяется следующей формулой: $H = \max\limits_{v \in V} t(v) - \min\limits_{v \in V} t(v) + 1$; при раздельном таймировании следующей формулой: $H = \max\limits_{1 \le k \le K} (\max\limits_{v \in V^{(k)}} t^{(k)}(v)) - \min\limits_{1 \le k \le K} (\min\limits_{v \in V^{(k)}} t^{(k)}(v)) + 1$.

3. Проектирование спецпроцессора

Можно представить граф алгоритма как виртуальный процессор, поместив в вершинах графа алгоритма ПЭ. Каждый ПЭ имеет входы, выходы и блоки ячеек локальной памяти для реализации задержек, с которыми из ПЭ должны выводиться результаты вычислений. Для проектирования СВ необходимо в лаконичной форме указать, что должно выполняться в каждом ПЭ в терминах входов, выходов и ячеек локальной памяти. Для этого вводится описание функционирования вершин графа алгоритма. В данной работе выведен общий вид описания функционирования вершин графа алгоритма при использовании теории расщепления и раздельного таймирования макроопераций алгоритма.

Основной особенностью проектирования СВ при использовании теории расщепления и раздельного таймирования макроопераций алгоритма является введение переменных задержек (число тактов, на которых результат вычисления должен храниться в ПЭ, прежде чем попасть на его выход). При использовании теории единого таймирования задержки постоянные для каждой вершины.

В некоторых случаях, для того чтобы описать функционирование ПЭ процессора, необходимо вводить в систему управляющие сигналы. Были получены условия введения управления и выведены новые условия на таймирование, связанные с введением управления.

На основе описанного в работе метода был спроектирован спецпроцессор для перемножения матриц. Высота ПФ алгоритма при этом определяется формулой $H = 3N$. Высота ПФ алгоритма при использовании теории единого таймирования макрооперации алгоритма $H = 4N - 3$. Таким образом, при размере задачи $N > 3$ вычислитель, спроектированный с использованием теории расщепления и раздельного таймирования макрооперации алгоритма, выполняет умножение матриц на $N - 3$ тактов быстрее.

Литература

1. *Кун С.* Матричные процессоры на СБИС / Пер. с англ. М.: Мир, 1991. 672 с.
2. *Воеводин В. В.* Математические модели и методы в параллельных процессах / М.: Наука, 1986. 296 с.
3. *Косьянчук В. В.* О синтезе систолических вычислителей // Весці АН БССР. Сер. фіз-мат. навук. 1991. № 2. С. 91–98.
4. *Седухин С. Г., Карапетян Г. З.* Проектирование оптимальных систолических систем для произведения матриц различной структуры / Новосибирск, 1990. 44 с. (Препринт/ АН СССР Сибирского отделения № 885).
5. *Лиходед Н. А., Соболевский П. И., Тиунчик А. А.* Распараллеливание алгоритмов на основе расщепления макроопераций // Докл. НАН Беларуси. 2001. Т. 45. № 4. С. 42–45.
6. *Баханович С. В., Лиходед Н. А., Соболевский П. И.* Пространственно-временное отображение алгоритмов с однородными зависимостями на параллельные архитектуры: Раздельное таймирование // Тр. Ин-та математики НАН Беларуси. Мн., 1999. Т. 3. С. 134–140.