КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКИ ПЕРЕСТРАИВАЕМЫХ МНОГОСТУПЕНЧАТЫХ МИКРОКОНВЕЙЕРОВ

С.В. Пискунов, Е.В. Умрихина

ВВЕДЕНИЕ

В настоящее время увеличения производительности вычислительного устройства, как правило, добиваются путем организации параллельной обработки данных на всех уровнях исполнения алгоритма, а уменьшения стоимости – переходом к построению устройств, в том числе конвейерных, собранных из некоторых стандартных, однотипным модулей как на микроуровне, так и на макроуровне. Этим обусловлен повышенный интерес к клеточным моделям вычислений, обладающих естественной локальностью, регулярностью и параллелизмом.

Клеточная технология организации вычислений, базирующаяся на модели распределенных вычислений, называемой Алгоритмом Параллельных Подстановок [1], представлена в [2, гл. 4]. Там же представлены основы проектирования 3D (многослойных) клеточных структур, в том числе универсальных. В таких структурах могут быть реализованы динамически перестраиваемые в процессе вычислений микроконвейеры [3].

Цель доклада – представить комплексный подход к построению динамически перестраиваемых многоступенчатых микроконвейеров, включающий как построение компьютерной модели конвейера, так и построение компьютерной модели загрузки конвейера данными и операциями. Выполняемый на конвейере алгоритм является некоторой асинхронной композицией составляющих его операций [4]. Модели строятся с использованием системы имитационного моделирования мелкозернистых алгоритмов и структур WinALT [2, 5].

ОБЩАЯ ХАРАКТЕРИСТИКА СРЕДСТВ, ИСПОЛЬЗУЕМЫХ ДЛЯ КОНСТРУИРОВАНИЯ И ИССЛЕДОВАНИЯ МИКРОКОНВЕЙЕРОВ

Опишем основные части клеточной технологии.

Алгоритм параллельных подстановок (АПП). Концептуально АПП объединяет в себе подстановочный характер алгоритма Маркова с пространственной параллельностью клеточного автомата, основываясь на общем для них ассоциативном механизме применения элементарных операций. Основная идея АПП заключена в следующих положениях.

Обрабатываемая информация задается в виде клеточного массива – множества клеток. Каждая клетка – это данное (бит, символ, число и т. д.) с приписанным «местом» нахождения в массиве.

Алгоритм задается множеством параллельных подстановок, имеющих левые и правые части. Выражение, стоящее в левой части подстановки, для каждого имени клетки из перерабатываемого клеточного массива порождает ассоциированный с этим именем клеточный массив и, если он содержится в перерабатываемом клеточном массиве, то подстановка выполняется. Ее выполнение означает, что некоторая, «базовая», часть найденного массива заменяется массивом, сгенерированным правой частью параллельной подстановки для того же самого имени клетки.

Процесс вычисления имеет итерационный характер: на каждом шаге выполняются все применимые в клеточном массиве параллельные подстановки. Вычисления заканчиваются тогда, когда к полученному на предыдущем шаге клеточному массиву не применима ни одна параллельная подстановка. Этот массив и есть результат работы АПП.

Система имитационного моделирования WinALT. Система построена на основе АПП и используется для конструирования, отладки и получения характеристик моделей мелкозернистых алгоритмов и структур. Имитационная модель оформляется в системе WinALT как проект, содержащий множество окон, в которых представлены графические объекты модели и тексты моделирующих программ.

В систему WinALT широко внедрены средства визуализации как для поддержки конструирования описаний параллельных алгоритмов (графическое представление объектов, изображающих клеточные массивы и шаблоны, задающие левые и правые части параллельных подстановок), так и при их моделировании (существует возможность наблюдения на экране динамики применения отдельных подстановок). Языковая часть системы WinALT состоит из трех частей. Первая часть предназначена для описания параллельных вычислений в виде параллельных подстановок, представляемых специальным составным оператором, включающим операторы *in – at – do*. В простейшем случае параметром оператора *in* является имя перерабатываемого клеточного массива, параметром оператора *at* – имя шаблона левой части команды параллельной подстановки, параметром оператора *do* – имя шаблона правой части команды параллельной подстановки. В более общем случае в составной оператор включаются конструкции, описывающие композицию клеточных массивов, использование переменных и функций. Такая конструкция языка как синхроблок для включенных в него составных операторов, описывающих параллельные

подстановки, реализует итеративную процедуру применения АПП. Вторая часть включает в себя набор конструкций последовательного языка подобных конструкциям языка Pascal. Объединение параллельной и последовательной частей языка достигается тем, что в синхроблоках возможно использование операторов и той, и другой части. Третья часть отвечает за расширяемость системы путем импорта в нее внешних для системы библиотек, написанных как на языке системы, так и на языках C и C++. Вариант параллельной версии системы был разработан в рамках проекта РФФИ № 99-07-90442 «Программное обеспечение суперкомпьютерного центра коллективного пользования Новосибирского научного центра СО РАН».

МНОГОСТУПЕНЧАТЫЙ МИКРОКОНВЕЙЕР

Главное назначение раздела – показать основные элементы микроконвейера, модель которого будет построена далее.

С логической точки зрения многоступенчатый конвейер является двухслойным клеточным автоматом (матрицей) с окрестностью Марголуса [6], в котором путем настройки имитируется одновременная работа множества копий одной и той же цифровой схемы, реализующей некоторую операцию, например, сложение или умножение, причем каждая копия (назовем ее виртуальной цифровой схемой) выполняет преобразование своего комплекта входных данных. На рисунках матрица изображается в виде развертки на плоскости разделенных на клетки слоев (рис. 1в, г). Слой, в котором хранится клеточный образ цифровой схемы и выполняется преобразование данных, называется информационным. Клетки информационного слоя могут находиться в одном из трех состояний: белом («пустом»), сером («логический 0») или черном («логическая 1»). Цифровая схема в информационном слое изображается серыми клетками. На рис. 1в представлен образ комбинационной схемы, изображенной на рис. 1б. Каждый блок информационного слоя (размером 2x2 клетки) настраивается на выполнение одной или двух команд параллельных подстановок из списка, приведенного на рис. 1а. Номера команд, на которые настроен блок, хранятся во втором, управляющем слое. Ступенью в микроконвейере называется вертикальный столбец блоков. Техника компактного отображения цифровой схемы в информационный слой клеточной матрицы, система сдвигов клеток информационного слоя относительно клеток управляющего слоя, механизм чередования четной и нечетной решеток Марголуса на каждом такте работы микроконвейера, вместе обеспечивающие преобразование и передачу данных со ступени на ступень в соответствии с номерами команд подстановок, детально описаны в [2, 3]. Каждая ступень конвейера – это «срез» образа виртуальной схемы в некоторой фазе преобразования предназначенной ей информации. Эта особенность матрицы позволяет подавать на вход конвейера на каждом такте новый комплект данных и после заполнения конвейера данными получать на ее выходе новые результаты на каждом такте. В матрице, показанной на рис. 1, одновременно может присутствовать шесть комплектов данных в разных стадиях преобразования.

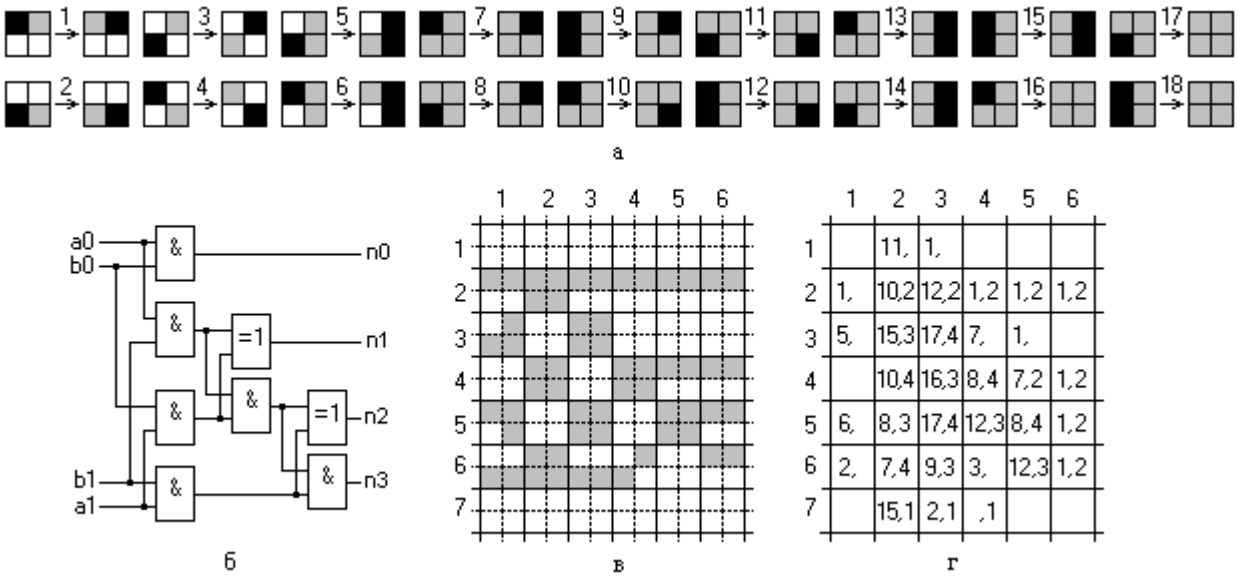

Рис. 1. Универсальная клеточная структура: а – команды параллельных подстановок, б – комбинационная схема, в – информационный слой, г – управляющий слой

Вместе с тем следует отметить, что микроконвейер, имитирующий работу больших цифровых схем, содержит большое число ступеней. Это означает, что возникает проблема его разгона: первого результата на выходе приходится ждать в несколько раз дольше, чем на выходе той цифровой схемы, поведение которой в нем имитируется. Как правило, необходимость повторного разгона микроконвейера возникает в том случае, когда нет исходных данных для той операции, на которую он настроен. Но это совсем не означает, что нет данных для какой-то другой операции, которую, однако, микроконвейер не может выполнить. Введение динамической перестройки микроконвейера позволяет избегать его повторного разгона, поскольку становится возможной его загрузка на каждом такте, не только новыми данными, но и новыми операциями, выполняемыми над этими данными, т.е. одновременно реализуется множество виртуальных цифровых схем разного назначения.

Для осуществления динамической перестройки конвейера каждый комплект входных данных снабжается ярлыком – кодом операции. В ступень конвейера, помеченную кодом операции, в блоки ее управляющего слоя вписываются номера подстановок на которые должен быть настроен информационный слой ступени конвейера. Модель такого конвейера представлена далее, в разделе 5.

С возможной физической реализацией микроконвейера можно ознакомиться в [2]. Для настройки конвейера используется буферная память, хранящая номера команд, выборкой которых из памяти управляют коды операций.

СХЕМА ЗАГРУЗКИ МИКРОКОНВЕЙЕРА

Задача устройства загрузки – обеспечить непрерывный поток данных и операций независимо от их типа: лишь бы они входили в ассортимент операций, на которые микроконвейер может быть настроен. Вариантов загрузки может быть много. В докладе рассматривается одна из возможных моделей загрузки, когда выполняемый на конвейере алгоритм является некоторой асинхронной композицией составляющих его операций.

МОДЕЛЬ МИКРОКОНВЕЙЕРА В СИСТЕМЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ WINALT

Модель загрузки микроконвейера. Управляющий алгоритм представляется в виде сети Петри [7]. Эта сеть обеспечивает выработку последовательности кодов операций, которые будут подаваться на вход динамически перестраиваемого конвейера. На рис. 2 показан скриншот модели загрузки конвейера для простейшей сети Петри. Не вдаваясь в детали, отметим характерные черты модели. В окне справа представлены графические объекты модели. В первом слое (показан слева) двухслойного клеточного массива byte::PNetFunc серыми клетками изображен образ сети Петри. Переход изображается серым прямоугольником из 2x7 клеток, позиция – квадратом 3x3 клетки, все клетки которого, кроме центральной серые. Черная центральная клетка изображает наличие маркера, белая – его отсутствие. Движением маркеров в соответствии с правилами функционирования сети Петри, представленными в виде АПП, управляет моделирующая программа, фрагмент которой представлен в окне справа. Информация для управления движением маркеров хранится в подмассивах, оконтуренных серыми клетками в управляющем слое (показан справа) массива byte::PNetFunc. Они поставлены в соответствие переходам сети. Языковые средства системы WinALT, использующие функции с переменными, определенными на значениях клеток выделенных подмас-

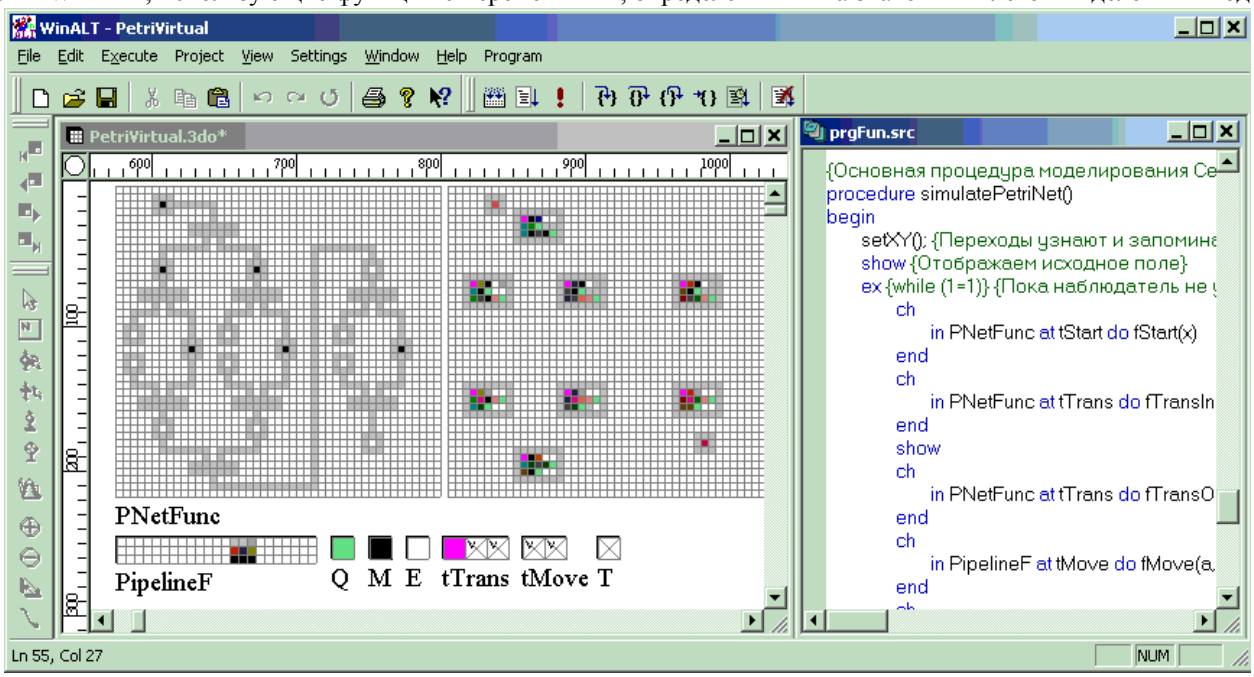

Рис. 2. WinALT-модель управляющей сети Петри

сивов, обеспечивают при срабатывании перехода непосредственную передачу маркеров из входных позиций в выходные. Это означает, что линии из серых клеток, соединяющие позиции и переходы играют чисто декоративную роль, обеспечивая наглядность представления сети Петри на экране компьютера, и могут быть проведены так, как это удобно пользователю. Каждый вертикальный столбец клеток массива byte::PipelineF хранит код операции, признак завершения операции и адрес помещения результата.

Имитационная модель микроконвейера. На рис. 3 показан скриншот, содержащий пример модели динамически перестраиваемого микроконвейера, выполняющего операции сложения и умножение четырехразрядных двоичных чисел. Массив byte::operation хранит коды операций. Информационный слой массива byte::grid хранит образы виртуальных цифровых схем. В текущий момент времени, зафиксированный на экране, выполняется группа операций умножения, потом – сложения, потом – снова умножения. Черной клеткой представлен код «1», перерабатываемой в ступени микроконвейера информации, серой – «0». В управляющем слое номера команд подстановки представлены цветом. В верхнем правом окне видна часть шаблонов, задающих левые и правые части команд подстановок. В нижнем правом окне представлен фрагмент текста моделирующей программы. Виден некоторый вариант оператора синхроблока *ch* и видны составные операторы, имитирующие команды подстановки.

Сделаем одно общее замечание. Для объединения моделей достаточно использовать средства языка системы WinALT, обеспечивающие композицию клеточных массивов или их подмассивов. Например, вместо массива byte::operation, должен использоваться подмассив массива byte::PipelineF, содержащий коды операций.

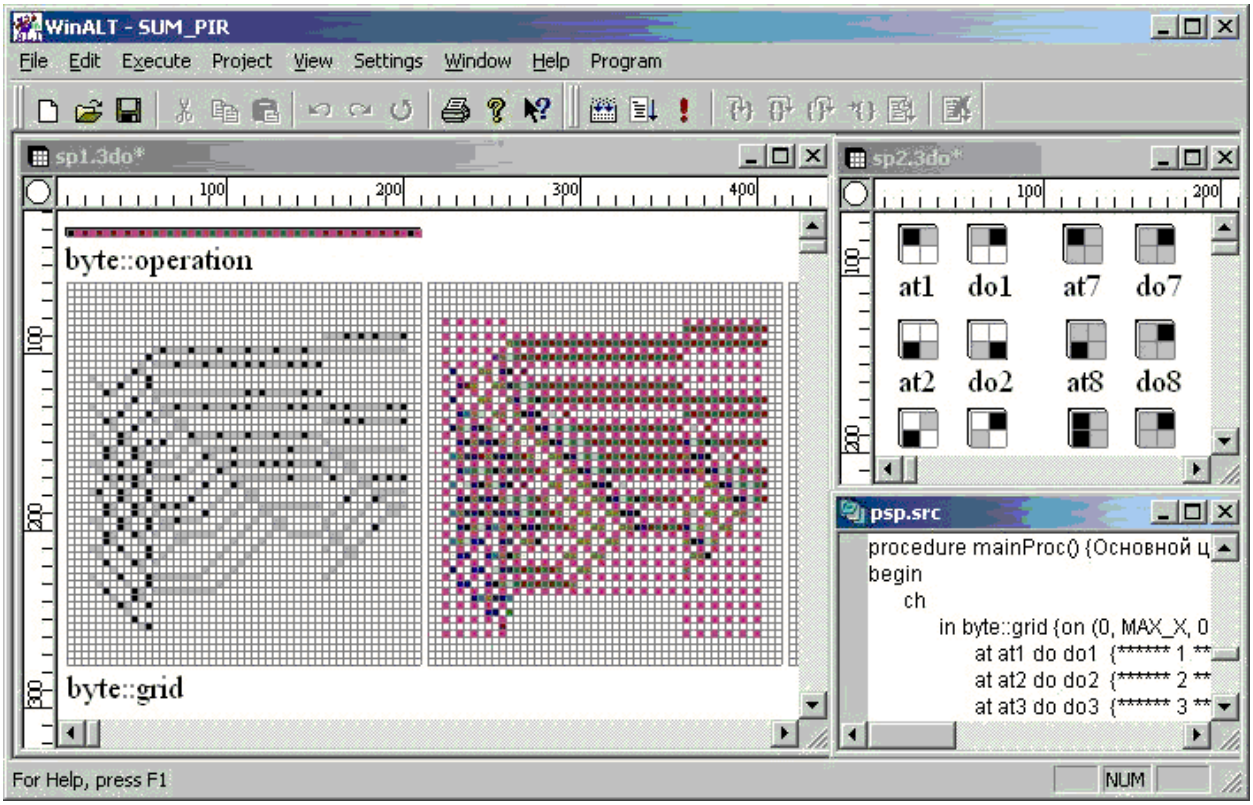

Рис. 3. WinALT-модель динамически перестраиваемого конвейера

ЗАКЛЮЧЕНИЕ

Предложена методика, позволяющая в едином клеточном пространстве представлять модели и сетевого управления, и массовых конвейерных вычислений. Можно ожидать, что ее использование при разработке программируемых СБИС с однородной структурой, позволит создать микросхемы конкурентоспособные по производительности с заказными СБИС и при этом существенно более технологичные, в первую очередь за счет замены проектирования заказной СБИС программированием матрицы на выполнение заданных операций непосредственно в процессе выполнения вычислений. Такие СБИС могут быть использованы при построении АЛУ современных вычислительных устройств.

ЛИТЕРАТУРА:

1. S.M. Achasova, O.L. Bandman., V.P. Markova, S.V. Piskunov. Parallel substitution algorithm. Theory and Application // World Scientific, Singapore, 1994, 220 p.
2. 3D лазерные технологии / Отв. редактор П.Е. Твердохлеб, 2003, Новосибирск, 550 с.
3. Э.Г. Косцов, С.В. Пискунов, Е.В. Умрихина. Перестраиваемые многослойные конвейерные оптико-электронные структуры // Автометрия. 2004. № 6. С. 75-86.
4. С.В. Пискунов, Е.В. Умрихина. Компьютерное моделирование асинхронной композиции алгоритмов с мелкозернистым параллелизмом // Научный вестник НГТУ. 2007. №3(28). С. 51-64.
5. D.T. Beletkov, M.B. Ostapkevich, S.V. Piskunov, I.V. Zhileev. WinALT, a software tool for fine-grain algorithms and structures synthesis and simulation // Lecture Notes in Computer Science. Berlin: Springer-Verlag, 1999, 1662. P. 491-496
6. Т. Тоффоли, Н Марголус. Машины клеточных автоматов. – М.: Мир, 1991, 278 с.
7. Дж. Петерсон. Теория сетей Петри и моделирование систем. – М.: Мир, 1984, 263 с.