

УДК 681.32

ЯЗЫК МАКРОСОВ ДЛЯ ПРОГРАММИРОВАНИЯ ОДНОРОДНОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЫ MiniTera II

В.Г. Титов, Н.А. Лукин

Институт машиноведения УрО РАН, г. Екатеринбург
E-mail: tit@imach.uran.ru, lookin@imach.uran.ru, nicklookin@mail.ru

Рассматривается решение одной из задач разработки среды программирования для однородной вычислительной среды, состоящей из локально связанных процессорных элементов. Предлагается язык описания макросов, позволяющий однозначно сопоставить произвольную алгоритмическую конструкцию топологии массива процессорных элементов.

Ключевые слова:

Однородная вычислительная среда, язык программирования, макрос.

Введение

Разработка средств программирования для параллельных вычислительных систем реального времени является бурно развивающейся областью информационных технологий. Объемы создаваемых программных проектов в ряде случаев исчисляются величинами в сотни тысяч исполняемых операторов. Это делает создание средств программирования параллельных вычислительных систем актуальным. Кроме того, для эффективного применения указанных систем постоянно требуется, чтобы время работы скомпилированной программы было минимальным. Это обуславливает необходимость оптимизации на каждом этапе разработки программного обеспечения, поэтому создание программных средств для параллельных вычислительных систем является весьма сложным. Большое значение для разработки программного обеспечения таких систем имеет учет в синтаксических и семантических конструкциях языков программирования существенных свойств архитектур параллельных вычислительных систем. Это представляет основную трудность для разработчиков таких языков, так как наработанный багаж теории и практики современного программирования основан на последовательных вычислениях на архитектуре фон-Неймана. Таким образом создание средств программирования для параллельных вычислительных систем реального времени является актуальным и требующим проведения исследований взаимного влияния алгоритмов и архитектур.

Предметной областью настоящей работы является программное обеспечение для определенного класса многопроцессорных потоковых архитектур – однородных вычислительных сред (ОВС), реализующих систолический принцип обработки данных. Для этой архитектуры декларативная парадигма программирования является более предпочтительной, чем императивная [1]. Более конкретно, в работе предлагается один из инструментов языка программирования ОВС – язык описания макросов для разработки программных проектов ОВС.

Однородные вычислительные среды

ОВС являются естественным этапом развития концепции параллельной обработки данных. В общем случае архитектура ОВС представляет собой N -мерную целочисленную решетку, в узлах которой размещены ПЭ, которые связаны между собой локальными информационными каналами. Топология ОВС в общем случае однозначно описывается конечным однородным неориентированным графом G [2], вершины которого соответствуют ПЭ, а ребра – информационным каналам. Степень i -ой вершины соответствует числу ПЭ, связанных локальными связями с ПЭ. Особенностью графов для топологии ОВС является отсутствие пересечений между любыми ребрами, это приводит к понятию размерности графов. Поэтому существуют 1-, 2-х, 3-х мерные графы, и каждая модификация графа соответствует топологии соединений ПЭ в архитектуре ОВС. Известные решетчатые графы [3] являются частным случаем рассматриваемых нами графов ОВС.

История развития ОВС насчитывает более 40 лет. Были реализованы многие интересные проекты (например проект 1-мерной систолической ОВС iWARP [4]). Для многих задач вычислительной математики и математической физики было показано, что архитектуры ОВС являются масштабируемыми, что обеспечивает максимальную производительность среди других типов параллельных архитектур. Тем не менее, широкого применения эти архитектуры пока не получили. Тому имеется две основных причины. Первая (и главная) – отсутствие до конца 90-х гг. XX в. развитой микроэлектроники, которая позволила бы в габаритах одного кристалла СБИС реализовать фрагмент ОВС либо всю среду целиком. Вторая – отсутствие развитых средств программирования, которые позволяют максимально эффективно использовать все множество ПЭ для параллельной обработки данных. И если первая причина в настоящее время практически устранена, то устранение второй представляется довольно сложной научно-техни-

ческой проблемой. В настоящее время известно примерно 20 реализованных проектов ОВС [5], но ни для одного из них не создано языка программирования, который обеспечивал бы эффективную загрузку массива ПЭ. Практически во всех случаях основой для языка программирования ОВС служат императивные языки типа С с дополнениями, обеспечивающими возможность параллельной обработки. Это приводит к значительным накладным расходам во времени реализации алгоритмов на ОВС и уменьшает их реальную производительность. Поэтому создание средств эффективного программирования ОВС является актуальной задачей настоящего времени.

В Лаборатории функционально-ориентированных процессоров ИМаш УрО РАН проводятся исследования и разработки в области ОВС, при этом основное внимание уделяется двум типам архитектур двумерных ОВС SIMD и MIMD.

В настоящей работе внимание уделено ОВС второго типа.

В начале 2000-х гг. коллектив лаборатории принимал участие в разработке ОВС MiniTera II [5], которая проводилась в рамках проекта «СКИФ». Архитектура этой ОВС представляет собой двумерный массив ПЭ; каждый ПЭ соединен с каждым из своих четырех непосредственных «соседей» четырьмя группами однонаправленных связей — двумя входными и двумя выходными. Таким образом, каждый ПЭ имеет восемь входных соединений и восемь выходных. Каждый ПЭ представляет собой набор независимых устройств предназначенных для обработки, хранения и передачи информации. В процессе работы программы ПЭ производит вычисления над операндами, поступающими с входов, и устанавливает значения своих выходов на каждом такте работы программы. Операнды и результаты поступают в последовательном коде, образуя поток битов. Операция, выполняемая ПЭ в течение работы одной программы, не изменяется. Все процессоры работают синхронно и независимо друг от друга. Таким образом, ОВС данного типа представляет собой плоскую двумерную систолическую машину обработки множества потоков данных. После предварительной настройки на выполняемую программу ОВС переходит в режим приема и обработки потоков данных, что позволяет рассматривать ее как мультимониторный потоковый функционально-ориентированный процессор (ФОП).

Архитектура ОВС MiniTera II приведена на рис. 1.

Программирование ОВС представляет собой процесс настройки каждого ПЭ на выполнение конкретной операции, что позволяет интерпретировать программирование как процесс укладки алгоритма на клеточном пространстве ПЭ. Компилирование алгоритма, написанного на языке высокого уровня, в совокупность исполняемых инструкций на множестве ПЭ представляет собой принципиально другой процесс, чем для архитектуры фон-

Неймана. Кроме того, сам язык программирования, вероятно, будет значительно отличаться от языков типа С, прежде всего, в части средств описания множественных потоков и их взаимодействия. Поэтому перечень проблем, которые нужно решить для создания средств программирования ОВС MiniTera II, весьма велик, а сами проблемы во многом не разрешены даже на теоретическом уровне. Все это заставляет уделять внимание многим конкретным и даже возможно частным задачам создания средств программирования ОВС. Одной из таких задач является разработка языка описания макросов.

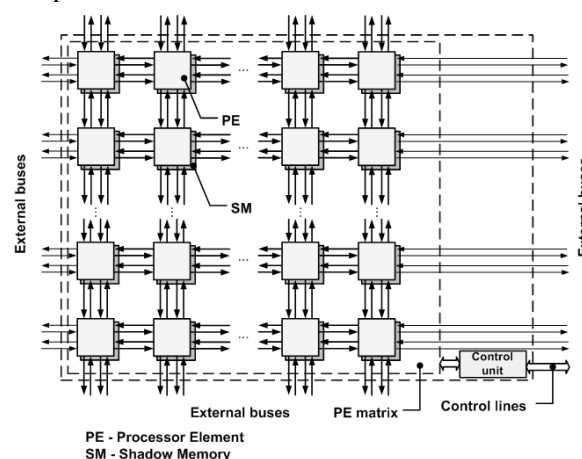


Рис. 1. Архитектура ОВС MiniTera II

Макросы в системе программирования ОВС

При создании реальных проектов часто требуется значительное число ПЭ, что может представлять трудность в программировании ОВС. Так, например, реализация алгоритма БПФ для N-отсчетного цифрового сигнала при N порядка тысяч может потребовать нескольких десятков тысяч ПЭ. Запрограммировать такой массив чрезвычайно сложно, поэтому весьма актуальным является использование иерархического программирования посредством макросов. Под макросом здесь подразумевается группа ПЭ или макросов для выполнения функции или операции.

Задача стандартизации базовых вычислительных процедур, считающаяся сложной для параллельных архитектур общего типа, становится еще более сложной для ОВС, поскольку в число информативных параметров макроса в этом случае необходимо включать геометрические (или топологические) параметры.

Таким образом, создание макросов и их поддержка системой программирования ОВС является актуальной научной задачей. Без ее решения практическое программирование ОВС является не реальным.

Настоящая работа посвящена решению одной из задач создания инструментальной среды программирования ОВС — языку описания макросов

ОВС, позволяющему, с одной стороны, однозначно описывать топологию соединений массива ПЭ любой конфигурации, а, с другой – реализовать однозначное соответствие между множествами входных/выходных переменных алгоритмической процедуры и соответствующими множествами внешних полюсов произвольного двумерного макроса.

В существующей системе программирования ОВС MiniTera II отсутствуют работоспособные механизмы текстового описания макросов.

Язык описания макросов

Этот язык состоит из одной списковой структуры, полное описание которой дано в Приложении. Эта списковая структура описывает макрос, как состоящий из ПЭ или из других макросов.

Макрос описывается списком вида:

`<имя исходного макроса>/XM,YM/((SI), (SO), (SPE&M));`
 где `<имя исходного макроса>` – имя длиной не более 8 символов; XM, YM – ширина и высота исходного макроса в количестве ПЭ; SI, SO – списки входов и выходов исходного макроса; SPE&M – список ПЭ и макросов исходного макроса.

Разработан также компилятор, который переводит эту списковую структуру в описание, которое воспринимается существующей системой программирования MiniTera II.

Пример

Пусть нужно вычислить:

$$c1 = (a1 + a2) * (a3 + a4)$$

$$c2 = (b1 - b2) * (b3 - b4)$$

$$c3 = d1 + d2 + d3 + d4 + d5 + d6$$

Реализуем вычисление $c1$ посредством макроса a (рис. 2):

```
a/3,2/( ( a1/0,0,UL/, a2/0,0,UH/, a3/2,0,UL/,
a4/2,0,UH/ ), ( c1/1,1,DL/ ),
( /0,0/( ( a1, a2 ), ( aa1/RL/ ), ( ALU( I1 = UL, I2 = UH, O = Add, O1 = RL ) ) ),
/2,0/( ( a3, a4 ), ( aa2/LL/ ), ( ALU( I1 = UL, I2 = UH, O = Add, O1 = LL ) ) ),
/1,0/( ( aa1, aa2 ), ( 1/DL/ ), ( ALU( I1 = LL, I2 = RL, O = Mul, O1 = DL ) ) ),
/1,1/( ( 1 ), ( c1 ), ( Tranzit( I1 = UL, O1 = DL ) ) ) );
```

Реализуем вычисление $c2$ посредством макроса b (рис. 2):

```
b/3,2/( ( b1/0,0,UL/, b2/0,0,UH/, b3/2,0,UL/,
b4/2,0,UH/ ), ( c2/1,1,DL/ ),
( /0,0/( ( b1, b2 ), ( bb1/RL/ ), ( ALU( I1 = UL, I2 = UH, O = Sub, O1 = RL ) ) ),
/2,0/( ( b3, b4 ), ( bb2/LL/ ), ( ALU( I1 = UL, I2 = UH, O = Sub, O1 = LL ) ) ),
/1,0/( ( bb1, bb2 ), ( 1/DL/ ), ( ALU( I1 = LL, I2 = RL, O = Mul, O1 = DL ) ) ),
/1,1/( ( 1 ), ( c2 ), ( Tranzit( I1 = UL, O1 = DL ) ) ) );
```

Реализуем вычисление $c3$ посредством макроса d (рис. 2):

```
d/3,2/( ( d1/0,0,UL/, d2/0,0,UH/, d3/1,0,UL/,
d4/1,0,UH/, d5/2,0,UL/, d6/2,0,UH/ ),
( c3/2,1,DL/ ),
( /0,0/( ( d1, d2 ), ( dd1/DL/ ), ( ALU( I1 = UL, I2 = UH, O = Add, O1 = DL ) ) ),
/1,0/( ( d3, d4 ), ( dd2/DL/ ), ( ALU( I1 = UL, I2 = UH, O = Add, O1 = DL ) ) ),
/2,0/( ( d5, d6 ), ( dd3/DL/ ), ( ALU( I1 = UL, I2 = UH, O = Add, O1 = DL ) ) ),
/0,1/( ( dd1, ddd2/RL/ ), ( dd4/RL/ ), ( ALU( I1 = UL, I2 = RL, O = Add, O1 = RL ) ) ),
/1,1/( ( dd2, dd4 ), ( ddd2, ddd4/RL/ ), ( Tranzit( I1 = UL, O1 = LL, I2 = LL, O2 = RL ) ) ),
/2,1/( ( dd3, ddd4 ), ( c1 ), ( ALU( I1 = UL, O1D = 1, I2 = LL, O = Add, O1 = DL ) ) ) );
```

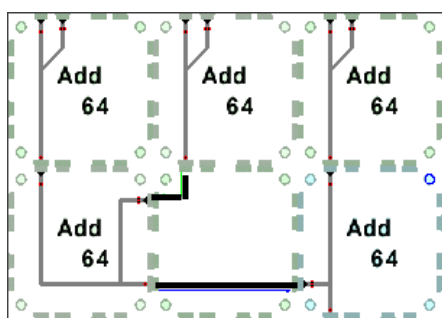
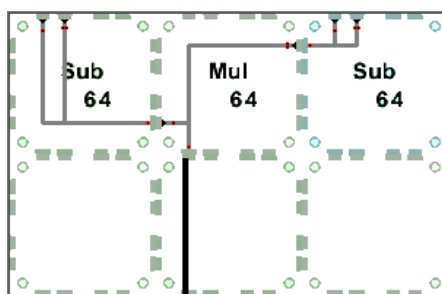
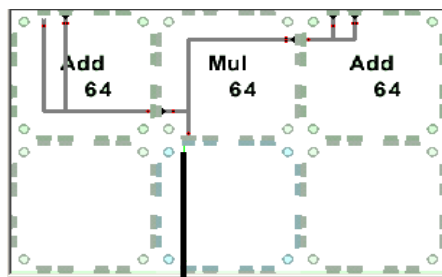


Рис. 2. Макросы a, b, d

Теперь можно с помощью макроса ccc вычислить $c1, c2, c3$ (рис. 3):

```
ccc/9,2/( ( a1, a2, a3, a4, b1, b2, b3, b4, d1, d2, d3,
d4, d5, d6 ), ( c1, c2, c3 ),
( a/0,0/( ( a1, a2, a3, a4 ), ( c1 ) ), b/3,0/( ( b1, b2,
b3, b4 ), ( c2 ), d/6,0/( ( d1, d2, d3, d4, d5, d6 ), ( c3 ) ) ) );
```

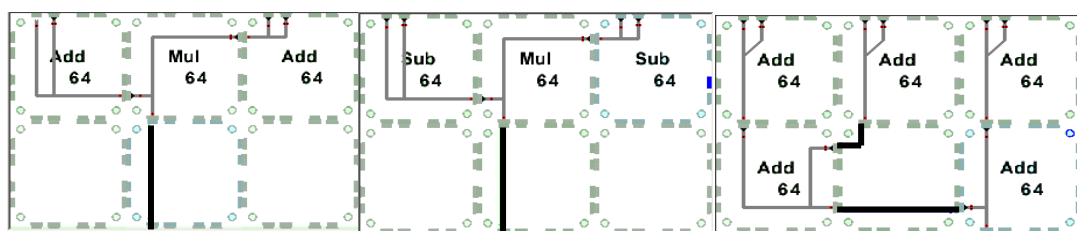


Рис. 3. Макрос ссс

Заключение

Язык описания макросов для ОВС MiniTera II предназначен для текстового описания и последующего порождения макросов. Этот язык дает возможность строить макросы любого типа, наличие компилятора позволяет визуализировать гра-

фическое обозначение макроса в существующей системе программирования ОВС.

В дальнейшем язык описания макросов будет дополнен средствами синхронизации потоков данных, что позволит автоматизировать компоновку макросов в составе целого проекта, а также отлаживать программу в клеточном пространстве ОВС.

СПИСОК ЛИТЕРАТУРЫ

1. Колдовский В. Язык Erlang и программирование для мультимедийных процессоров [Электронный ресурс]. – режим доступа: <http://itc.ua/node/26721/>. – 23.09.2008.
2. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера. – М.: Энергия, 1980. – 409 с.
3. Воеводин В.В. Математические модели и методы в параллельных процессах. – М.: Наука, 1986. – 296 с.
4. Gross T., O'Hallaron D.R. iWARP. Anatomy of a Parallel Computing System. – MIT Press, March 1998. – 530 p.
5. Лукин Н.А. Реконфигурируемые процессорные массивы для систем реального времени: архитектуры, эффективность, области применения. // Известия ТРТУ. – 2004. – № 9. – С. 36–45.

Поступила 14.07.2008 г.

УДК 004.89

НЕЙРОННАЯ СЕТЬ, ОСНОВАННАЯ НА ТОЧЕЧНЫХ ОТОБРАЖЕНИЯХ

А.К. Стоянов

Томский политехнический университет
E-mail: stoj-ak@ad.ctpu.edu.ru

Показана возможность реализации точечных отображений в нейронной сети прямого распространения. Модель нейронной сети, основанной на одномерных точечных отображениях, позволяет рассчитывать параметры нейронов и их количество. Приводятся результаты экспериментальной проверки модели на примере создания нейронной сети-классификатора.

Ключевые слова:

Нейрон, нейронная сеть, точечные отображения, особые точки, классификация, фильтрация.

Одной из особенностей применения нейронных сетей является «непрозрачность» механизма получения конечного результата. Нейронная сеть представляется «чёрным ящиком», который путем обучения настраивается на достижения результата. Это затрудняет как поиск новых путей использования, так и подбор параметров сетей, не говоря уже о возможности их прямого расчёта. «Непрозрачность» не позволяет сформулировать рекомендации по выбору числа нейронов и слоёв в сети, не позволяет прогнозировать поведение сети. В настоящей работе рассматривается модель нейронной сети с ясным механизмом получения конечного результата и возможностью её расчёта.

Моделирование нейронных сетей одномерными отображениями

Пусть имеется многослойная нейронная сеть прямого распространения [1], схематично представленная на рис. 1. Слой входных нейронов – 0 слой, слой выходных нейронов – 3 слой.

Каждый из нейронов одного слоя соединен своим выходом с входами всех нейронов соседнего слоя: нейроны входного слоя с нейронами 1 (скрытого) слоя; эти, в свою очередь, с нейронами 2 слоя и т. д. На рис. 1 изображена часть таких соединений. На нейроны входного 0 слоя подается дискретный сигнал $S=\{x_i\}$, $i=0,1,2,\dots$. Здесь i – номер