

ПРОЕКТИРОВАНИЕ МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ НА БАЗЕ ПРОГРАММИРУЕМЫХ ИНТЕГРАЛЬНЫХ МИКРОСХЕМ И ВОЗМОЖНОСТИ ИХ АВТОМАТИЗАЦИИ

к.т.н. О.И. Богатов, к.т.н. Ю.В. Кулявец, к.т.н. Д.П. Лабенко
(представил д.т.н., проф. Е.И. Бобыр)

Рассматривается один из путей автоматизации разработки много-процессорных вычислительных систем на базе программируемых интегральных микросхем.

Постановка проблемы. Успехи микроэлектроники за последние десятилетия резко расширили границы алгоритмических возможностей в вычислительной технике. Это явилось следствием значительного увеличения быстродействия и степени интеграции элементов вычислительных устройств, объясняющимися рядом технологических достижений, а также распараллеливания и конвейеризации вычислительного процесса в мультипроцессорных вычислительных системах (МВС), которые стали возможными благодаря улучшению надежностных и экономических показателей процессорных элементов (ПЭ).

В числе требований к основным техническим характеристикам специализированных МВС, входящих в состав тактико-технического задания (ТТЗ), обычно доминирующую роль играют требования к производительности (в сущности, само возникновение МВС явилось результатом непрерывно растущих требований к производительности вычислительных средств). Неотъемлемой частью ТТЗ являются также сроки разработки, которые, как правило, конфликтуют с требованиями к техническим характеристикам. Это имеет и свои объективные причины, так как сроки разработки не должны быть соизмеримы со сроками цикла морального старения МВС, которые в настоящее время продолжают существенно сокращаться. Отсюда следует **актуальность** следующих задач:

- автоматизации всех этапов проектирования МВС, и в первую очередь, наиболее сложных и трудоемких;
- разработки перспективных направлений увеличения производительности МВС.

Анализ литературы. В течение последнего десятилетия ведется интенсивный поиск новых решений, которые могут быть положены в

основу перспективных МВС, что приводит обычно к сотовым автоматам и искусственным нейронным сетям [1, 2], которые могут быть объединены под общим названием систем искусственного интеллекта. Однако такое направление предполагает полный отход от общепринятой в настоящее время теории проектирования, как аппаратных, так и программных вычислительных систем, которые за полвека существования позволили накопить достаточно большой и ценный теоретический и экспериментальный материал. Поэтому, несмотря на большой интерес со стороны специалистов во многих областях и некоторые практические результаты (например, экспериментальная вычислительная система фирмы Thinking Machines CM-1 (Connection Machine), содержащая 65536 элементарных ПЭ), подобные системы еще очень далеки от возможностей широкого внедрения.

Одна из ведущих в мире фирм в области программируемых логических интегральных схем (ПЛИС) Altera [3] в настоящее время специализируется в области разработки процессоров обработки сигналов на основе ПЛИС, что подтверждает правильность выбранного направления в части применения ПЛИС. Однако, если для схемной реализации алгоритма конкретной задачи можно воспользоваться диалоговым процессом проектирования в рамках пакета MAX + PLUS II, то для эффективной организации вычислительного процесса в МВС на базе ПЛИС по алгоритму заранее оговоренной задачи необходима определенная методология и гораздо большая степень автоматизации.

1. Формализация фрагментного представления. Для разработки многопроцессорных вычислительных систем (МВС) заданной производительности необходимо выполнить соответствующее преобразование алгоритма решаемой задачи с целью представления его в терминах некоторых фрагментов, которые могут быть эффективно реализованы структурами типа ПЛИС.

Фрагменты должны удовлетворять следующим требованиям:

1) каждый фрагмент должен реализовать в составе алгоритма определенную функцию или несколько функций, которые могут быть описаны в терминах общепринятых математических операций из современных языков программирования, обеспечивающих возможности эквивалентных преобразований с целью достижения определенного оптимального решения;

2) каждый фрагмент должен иметь реальную возможность быть реализованным на уровне комбинационных схем;

3) разбиение на фрагменты должно преследовать цели обеспечения регулярности структуры МВС, реализующей заданный алгоритм, то есть

каждый фрагмент по возможности должен удовлетворять условиям максимальной повторяемости в алгоритме;

4) фрагменты могут иметь иерархическую структуру.

В определенном смысле здесь термин «фрагмент» является обобщением понятия операция из общего списка операций ПЭ. Однако в отличие от операций в ПЭ, которые никак не отражают алгоритмических особенностей решаемой задачи, фрагмент в указанном выше смысле слова строго привязан к соответствующему алгоритму.

Для возможностей схемной реализации фрагмента согласно требованиям 1 и 2 следует необходимость разработки автоматизированного перевода из описания фрагмента на алгоритмическом уровне в систему булевых уравнений (необходимым условием этого является четкая конкретизация типов используемых в алгоритме переменных). Первый вариант перевода был реализован в архитектуре IBM 370 в виде пакета под именем METACAD.

Исходя из некоторого опыта эксплуатации данного пакета совместно с системой PLDshell программной поддержки микросхем ПЛИС типа Flexlogic, более конкретно вырисовались требования к объему фрагмента. Так, суммарный объем исходных данных фрагмента при нетривиальных преобразованиях в нем не должен превышать, как правило, 10 – 13 бит (максимально возможное число бит для пакета METACAD составляет 16 бит). Такое ограничение диктуется, с одной стороны, конструктивными особенностями ПЛИС, и с другой, временем вычисления булевых выражений в пакете, которое носит явно экспоненциальный характер в зависимости от суммарного числа бит. Конструктивные особенности микросхем определяют максимально допустимое число термов в дизъюнктивной форме булевых выражений, которые могут быть реализованы в одной макроячейке ПЛИС (для семейства Flexlogic эта величина составляет 8 и в определенных случаях 16 бит). Если число термов превышает допустимое, необходимо искусственное разделение одного булевого выражения на несколько, что влечет за собой увеличение числа задействованных макроячеек.

Общее число бит в переменных вычислительных алгоритмов, как правило, превышает указанные выше величины. Поэтому их целесообразно представлять в виде нескольких слов, состоящих из небольшого числа бит. Например, пусть некоторая переменная X требует для представления 36 бит. В таком случае ее можно разделить, например, на несколько слов в соответствии с одним из представлений в правых частях:

$$X = \sum_{i=1}^{18} x_i^{(2)} \cdot 2^{2(i-1)} = \sum_{i=1}^{12} x_i^{(3)} \cdot 2^{3(i-1)} = \sum_{i=1}^9 x_i^{(4)} \cdot 2^{4(i-1)} = \sum_{i=1}^6 x_i^{(6)} \cdot 2^{6(i-1)}, \quad (1)$$

где $x_i^{(j)}$ – соответствующее i -е слово длиной j бит.

Выбор конкретного представления из указанных в (1) производится исходя из числа переменных, одновременно принимающих участие в данном фрагменте (предполагается, что в каждом фрагменте участвует какая-либо часть $x_i^{(j)}$ переменных X). Подобное разбиение хорошо согласуется с одним из принципов, положенных в основу суперкомпьютера MPP (Massively Parallel Processor) [4], разработанного по заказу Национального управления по авиации и исследованию космического пространства США, по которому операнды могут иметь произвольную разрядность.

Циклический характер подавляющего числа вычислительных алгоритмов хорошо согласуется с требованием 3. Разбиение переменных на отдельные слова, подобное (1), с последующей обработкой укороченных переменных внесет дополнительные циклы в алгоритм, согласуясь с требованием 4.

2. Этапы логического проектирования МВС. Исходя из вышеизложенного, процесс логического проектирования МВС для конкретной задачи (или конфигурирования существующей МВС с ПЭ на основе ПЛИС) будет состоять из трех последовательных этапов.

Этап 1. Представление алгоритма задачи в терминах соответствующих фрагментов и операторов управления.

Этап 2. Перевод алгоритма в терминах фрагментов в абстрактную пространственно-временную модель, однозначно определяющую структуру МВС, построенную из схем реализации данных фрагментов со всеми информационными связями между ними и входными (выходными) потоками данных, а также временную последовательность их работы в процессе реализации алгоритма задачи.

Этап 3. Перевод абстрактной пространственно временной модели в схему на конкретных микросхемах ПЛИС, включающий в себя назначение выводов микросхем в соответствии с их информационными связями, реализацию обратных связей на регистрах или элементах памяти и схемную организацию временной последовательности управляющих сигналов.

В такой постановке процесс проектирования представляется в виде ряда причинно связанных друг с другом этапов, имеющих четкую математическую подоплеку. Как показывает анализ, первый этап является наиболее ответственным, требующим творческого подхода, в конечном итоге определяющим качество работы системы в процессе решения задачи. На данном этапе требуется привлечение современных методов вычислительной математики, а также разработка новых с учетом указан-

ных выше требований. Из средств автоматизации можно назвать пока лишь пакет METACAD с последующими доработками и переводом в другие архитектуры, а также математические пакеты, работающие с символьными выражениями типа Maple-6.

Решение задачи этапа 2 стало возможным благодаря введению математической модели элемента вычислительного процесса и основных принципов организации математического аппарата формальных преобразований, базирующихся на этой модели, которые позволили выбором отдельных числовых параметров с единых позиций находить множество пространственно-временных описаний всех схемных решений, начиная от полностью последовательного и кончая решением с наивысшей степенью параллелизма. На этом этапе в процессе формальных преобразований однозначно определяются требования к памяти и УУ. Второй этап, несмотря на большую трудоемкость и объем перерабатываемой информации, согласно предварительной проработке хорошо формализуется и не содержит принципиальных препятствий, не допускающих его полной автоматизации.

Третий этап также может быть полностью автоматизирован. Однако он должен содержать большое количество различных вариантов, зависящих, с одной стороны, от типа ПЛИС, а с другой, от различной организации одних и тех же функций, например, адресная память или сдвиговые регистры, D-триггеры или T-триггеры и т.д. К тому же, при большом количестве ПЭ в МВС практически невозможно организовать непосредственные связи между каждой парой ПЭ. Поэтому становится актуальной задача разработки принципов организации регулярной сети обмена между ПЭ, оптимальной по времени его реализации.

3. Практическая реализация. Принципы организации указанных выше этапов проектирования были частично верифицированы на примерах построения проектов устройств для сортировки в рамках пакета PLDshell, основанных на параллельном алгоритме [5]. В примерах подвергались сортировке массивы натуральных чисел Z в диапазоне $0 \leq Z \leq 2^{16} - 1$, целых чисел Z , представленных в дополнительном коде, в диапазоне $-2^{15} + 1 \leq Z \leq 2^{15} - 1$, и натуральных чисел, представленных кодами Грея. Мощность массивов сортируемых чисел была представлена величинами 6, 7, 8, 9 и 10. Подобные ограничения были связаны с возможностями размещения соответствующих проектов в одной микросхеме ПЛИС EPX780 или EPX8160 семейства Flexlogic, так как в пакете PLDshell не предусмотрены возможности моделирования более одной микросхемы.

В соответствии с этапом 1 алгоритм из [3] был представлен фрагментами F1 и F2, реализующими некоторые функции:

$$F1: \{1, 2, \dots, p\}^2 \times \{0, 1, \dots, n-1\}^2 \rightarrow \{0, 1\};$$

$$F2: \{0, 1, \dots, n-1\} \times \{0, 1\}^q \rightarrow \{0, 1, \dots, n-1\}$$

при $p = 2, 3$ и $q = 2, 3$, и операторами цикла со степенью вложенности: до 4 включительно (параметр p обозначает число бит в каждом слове, образованном из сортируемых чисел, подлежащих обработке во время каждого такта подобно (1), и характеризует функцию F1; параметр q характеризует функцию F2 и ее связи с функциями F1). Функции F1 и F2 генерировались в пакете METACAD из соответствующего математического описания в алгоритме [5]. На этапе 2 выбором параметров соответствующих циклов и их вложенностью было определено множество решений проекта с различной степенью параллелизма. На этапе 3 с использованием пакета METACAD, на основании полученных на этапе 2 временных последовательностей работы системы, определялись схемные решения обратных связей на сдвиговых регистрах и схемные решения реализации временных последовательностей в двух вариантах: на D-триггерах и на T-триггерах, а также, где это возможно, полное схемное решение на микросхемах EPX780 и EPX8160.

Было получено более 40 различных проектов для решения задачи сортировки в указанной выше постановке, начиная от последовательного варианта, использующего только по одной схеме для реализации фрагментов F1 и F2 соответственно, требующего для полного завершения 378 тактов работы устройства, до варианта с наибольшей степенью параллелизма, размещение которого возможно в одной ПЛИС EPX8160, использующей 42 схемы, реализующих фрагмент F1 и 14 схем, реализующих фрагмент F2, с общим временем работы 9 тактов. Следует также отметить, что используемая на этапе 3 методика [6] нахождения схемной реализации временных последовательностей управляющих воздействий на основе пакета METACAD оказалась более эффективной, чем применяемая во всех САПР методика на основе теории конечных автоматов (или последовательных машин). Эффективность эта заключается в наиболее простой и естественной с точки зрения обычной математики форме задания условий работы устройства и не требующей громоздкого перечисления полного списка последовательности переходов из одного состояния в другое.

Для организации оптимальной сети обмена между ПЭ в МВС был проведен анализ возможностей использования подстановок в симметрических группах [7, 8] в качестве основного аппарата для нахождения

последовательностей параллельных обменов. Результаты анализа дали положительные результаты для сетей с архитектурой типа «гиперкуб», которые, по-видимому, могут быть распространены и на другие сетевые архитектуры.

Важную роль в процессе проектирования МВС на основе ПЛИС играет выбор соответствующего языка для описания и моделирования проектируемых структур. В этом отношении практически идеально подходящим является язык VHDL (Very High Speed Integrated Circuit Hardware Description Language), разработанный в США по заказу Министерства обороны и с 1987 года ставший стандартом IEEE Std 1076 [9]. VHDL является языком, основанным на синтаксисе языка Ада, и позволяет описывать цифровые схемы с любой степенью детализации и практически любой сложности, начиная с простейших вентилях и кончая самыми мощными современными вычислительными комплексами.

Выводы и перспективы дальнейших исследований. Таким образом, эффективность реализации алгоритма задач в МВС зависит от этапа 1, который является прямым аналогом этапа программирования (в узком смысле) для МВС традиционной структуры. Однако он в определенном смысле является более сложным, чем программирование по заданному алгоритму, хотя бы потому, что отсутствует достаточный опыт, а также требуется критически оценить и, возможно, видоизменить ряд методов из численного анализа. Естественным способом, как было отмечено в разделе 3, является использование операций, работающих одновременно со многими переменными, имеющими малую разрядность, т.е. являющимися частями многоразрядных чисел, представляющих входные данные и промежуточные результаты в алгоритме.

Для случая с МВС на основе ПЛИС такое применение легко распространяется на случай реализации полиномов, или, более того, на любые линейные участки алгоритмов, содержащие только операции умножения и сложения (вычитания). В участках алгоритма такого рода обработка аргументов начинается со слов, представляющих младшие разряды, и заканчивается старшими, что объясняется спецификой распространения переносов. Однако в других задачах, в частности в алгоритме сортировки [5], обработка начинается со слов, представляющих старшие разряды. К такому типу, как показал предварительный анализ, могут быть приведены и алгоритмы медианной фильтрации. Таким образом, проектирование МВС на основе ПЛИС даст дополнительный толчок развитию алгоритмов для работы со многими переменными

Естественно, вторым главным достоинством САПР на основе МВС, является возможность автоматизации максимального числа стадий логи-

ческого проектирования. К тому же, если результат будет представлен на языке VHDL, то становятся доступными все наиболее известные системы технического проектирования, в том числе и кремниевые компиляторы.

И, наконец, развитие МВС на основе ПЛИС позволит устранить парадокс, когда самая «интеллектуальная» научная дисциплина в своем перспективном развитии методично следует по пятам технологии в микроэлектронной промышленности.

ЛИТЕРАТУРА

1. Ферри Д., Эйкерс Л., Гринич Э. Электроника ультрабольших интегральных схем. – М.: Мир, 1991. – 328 с.
2. Кун С. Матричные процессоры на СБИС. – М.: Мир, 1991. – 672 с.
3. Beacher D. Next-Generation Tool Requirements // News&Views, Newleter for Altera Customers. – August, 1998. – Third Quarter. – P. 18 – 19.
4. Кучук Г.А. Минимизация загрузки каналов связи вычислительной сети // Системы обработки інформації. – Х. : НАНУ, ПАНМ, ХВУ, 1998. – Вип. 1(5). – С. 149-154.
5. Кондратьев В.Н., Богатов О.И., Горовой С.И. Параллельный алгоритм сортировки // Збірник наук. пр.. ХВУ. – Х.: ХВУ. – 1998. – № 14. – С. 49 – 55.
6. Кондратьев В.Н., Богатов О.И., Тесля В.Н. О возможностях учета алгоритмических задач // Системы обработки інформації. – Х.: НАНУ, ПАНМ, ХВУ. – 1999. – Вип. 1 (5). – С. 124 – 127.
7. Кондратьев В.Н., Богатов О.И., Тесля В.Н. Оптимальный обмен данными в мультипроцессорной системе // Системы обработки інформації. – Х.: НАНУ, ПАНМ, ХВУ. – 2000. – Вип. 3 (9). – С. 3 – 5.
8. Кондратьев В.Н., Богатов О.И., Тесля В.Н. Моделирование процесса оптимального обмена в мультипроцессорной системе // Системы обработки інформації. – Х.: ХФВ “Транспорт України”. – 2001. – Вип. 1 (11). – С. 41 – 44.
9. Армстронг Дж. Моделирование цифровых систем на языке VHDL. – М.: Мир, 1992. – 171 с.

Поступила 26.11.2004

БОГАТОВ Олег Игоревич, канд. техн. наук, ст. научн. сотр., начальник научно-исследовательского отдела Объединенного НИИ Вооруженных Сил. В 1990 году окончил ВИРТА ПВО. Область научных исследований – автоматизированные системы управления и обработки информации.

КУЛЯВЕЦ Юрий Владленович, канд. техн. наук, заместитель начальника научно-исследовательского отдела Объединенного НИИ Вооруженных Сил. В 1987 году окончил ЖВУРЭ ПВО. Область научных исследований – обработка радиолокационной информации.

ЛАБЕНКО Дмитрий Петрович, канд. техн. наук, доцент, начальник научно-исследовательского отдела Объединенного НИИ Вооруженных Сил. В 1991 году окончил ВИРТА ПВО. Область научных исследований – автоматизированные системы управления и обработки информации.