

УДК 681.3

## Комплекс для проектирования аппаратных решений, эффективно реализующих сложные алгоритмы обработки данных

А.И.Толкачѐв

В процессе разработки встроенных систем обычно перед разработчиками стоит проблема обеспечения требуемой производительности с использованием ограниченного количества аппаратных ресурсов. В большинстве случаев используется традиционная архитектура – микропроцессор с постоянной кодовой памятью, адресуемой оперативной памятью и набором регистров. Для некоторых задач используются специализированные процессоры, например, процессоры для цифровой обработки сигналов (DSP, digital signal processing), VLIW процессоры (very long instruction word, процессоры с длинным словом инструкций) и другие. Применение специализированных процессоров позволяет увеличить производительность системы за счет распараллеливания обработки и наличия специфических инструкций, быстро реализующих определенные алгоритмы.

С появлением конфигурируемых чипов открылись широкие возможности для проектирования архитектуры вычислительной системы, наиболее эффективно использующей имеющиеся ресурсы. Однако встает проблема отсутствия эффективных средств для проектирования таких систем. Имеющееся на рынке программное обеспечение обычно предназначено для проектирования и синтеза схемы, ориентировано на использование определенного типа устройств и не предоставляет широких возможностей отладки. При проектировании устройств, в которых требуется реализация относительно сложных алгоритмов обработки, использование таких средств проектирования связано с большими затратами на разработку и сопровождение.

### Используемый подход и область возможного применения

Разрабатываемая нами система представляет собой комплекс для проектирования аппаратных решений, эффективно реализующих сложные алгоритмы обработки данных. В основе системы лежит оптимизирующий компилятор языка высокого уровня в описание аппаратуры. Интеграция с системами высокоуровневого проектирования цифровых устройств WInter [1] и IEESD [1] дает широкие возможности для моделирования, отладки и тестирования разрабатываемой системы.

Наиболее перспективная область применения системы – разработка специализированных процессоров для встроенных систем. Использование специально разработанных для конкретной задачи схемных решений может иметь преимущество перед использованием серийных моделей микроконтроллеров.

Эффективность специализированных схемных решений объясняется более полным использованием аппаратных ресурсов, в частности, возможностью параллельного выполнения инструкций и обращений к памяти. Многие задачи, требующие интенсивных вычислений, могут эффективно использовать возможность параллельного исполнения. К ним относятся, например, задачи обработки аудио и видеoinформации. Это делает возможным использование специализированных схемных решений в различных портативных устройствах, для которых важна высокая производительность и жесткие требования к используемому объему ресурсов.

### Альтернативные решения

Альтернативными решениями являются использование VLIW и DSP процессоров. Использование нашей технологии является более эффективным в случае, когда алгоритм об-

работки может быть значительно ускорен на счет распараллеливания. Возможности параллельной обработки процессоров ограничены относительно небольшим количеством инструкций или единиц данных. Кроме этого, VLIW технология требует жесткого определения групп одновременно исполняемых инструкций в исполняемом коде. Это накладывает дополнительные ограничения на возможность распараллеливания.

Используемая в нашей системе технология построения схемы на основе микропрограммных автоматов позволяет распараллеливать произвольное количество инструкций и лишена указанного выше недостатка VLIW процессоров. Используя нашу систему, разработчик описывает алгоритм работы устройства на традиционном языке программирования высокого уровня без явного описания параллелизма. Компилятор анализирует программу, выделяет независимые блоки, которые могут исполняться одновременно, и генерирует соответствующую схему.

Использование традиционного последовательного языка программирования значительно облегчает разработку и отладку программы. Анализатор может обнаружить возможность распараллеливания как отдельных инструкций, так и целых фрагментов программы.

Возможно, более эффективным решением с точки зрения производительности разрабатываемой системы является проектирование разработчиками аппаратной схемы, а не ее автоматизированная генерация с использованием нашей системы. Однако в этом случае значительно увеличиваются затраты на разработку, поскольку создание и отладка аппаратной схемы – значительно более трудоемкий процесс, чем написание программы на языке высокого уровня.

### Компоненты системы

Система включает в себя следующие компоненты:

- Настраиваемый компилятор языков высокого уровня в промежуточное представление программы. Эта часть реализована на основе универсального синтаксического анализатора Unisan [2]. В настоящее время планируется реализация компилятора для языка Си.
- Оптимизатор промежуточного представления. Эта часть содержит алгоритмы анализа программы и выявления параллелизма на уровне отдельных инструкций, а также другие алгоритмы оптимизации.
- Генератор псевдокода, используемого позже для моделирования и генерации схемы.
- Генератор схемы. Эта компонента интегрирована в систему высокоуровневого проектирования цифровых устройств IEESD и позволяет сгенерировать оптимизированную по выбранным критериям схему для реализации заданного алгоритма. Для генерации схемы используется технология микропрограммных автоматов. Средствами системы IEESD может быть получено синтезируемое описание аппаратуры на языке Vhdl.
- Модель процессора микропрограммных автоматов. Для эффективного моделирования исполнения исходной программы, а также отладки, вместо сгенерированной схемы может использоваться программная модель микропрограммных автоматов. Эта возможность полезна при отладке больших схем, моделирование которых на уровне логических элементов требует значительных ресурсов. Модель микропрограммных автоматов в качестве исходных данных получает псевдокод, сгенерированный в результате работы компилятора. В системе IEESD модель выглядит как процессор, имеющий выходы, описанные в исходной программе.
- Отладчик исходного кода. Отладчик интегрирован в системы WInter и IEESD и позволяет исполнять по шагам и отлаживать программу по исходному тексту. Имеются все обычные возможности отладчика языка высокого уровня - просмотр значений переменных, установка точек останова и др. Кроме этого, может быть использована встроенная в системы IEESD и WInter система автоматического тестирования на заданном наборе тестов.

Качество получаемой схемы зависит главным образом от двух компонент системы – оптимизатора промежуточного представления и генератора схемы.

После того, как применен описанный алгоритм, можно применить его для каждого блока операторов, являющегося телом оператора for, while, if и др.

Применив этот алгоритм для всех блоков операторов на всех уровнях вложенности, получим достаточно эффективную схему передачи управления.

Для иллюстрации алгоритма рассмотрим фрагмент программы на языке Си:

```
void func(int count, double a[], double b[], double ab[][]))
{
    int i, j;
    double scalar, norm_a2;
    scalar = 0.0;
    norm_a2 = 0.0;

    // вычисление скалярного произведения векторов a и b
    // и квадрата нормы вектора a
    for( i=0; i<count; i++)
    {
        scalar += a[i] * b[i];
        norm_a2 += a[i] * a[i];
    }

    // вычисление произведения векторов a и b'
    // (a - столбец, b' - строка)
    for( i=0; i<count; i++)
        for( j=0; j<count; j++)
            ab[i][j] = 0.0;

    for( i=0; i < count*count; i++)
        ab[ i/count ][ i%count ] += a[i] * b[j];

    ...
}
```

На рис. 1 изображен граф передачи управления после применения алгоритма для блоков всех уровней. Блок с оператором Wait используется для синхронизации параллельно работающих инструкций. Оператор Wait передает управление следующим операторам только после того, как получит управление от всех операторов, указанных входящими в него ребрами.

### Abstract

The paper describes the system for developing hardware schemes of the complicated data processing algorithms. It is integrated with hardware debugger and simulator. The system can be used to design FPGA based devices. Our approach is alternative to use of DSP and VLIW processors. It can be useful to design parallel processing hardware schemes that are used in the wide range of devices.

### Литература

1. Долинский М.С., Литвинов В.А., Ермолаев И.Ю., Федорцов А.О. Пакет инструментальных комплексов сквозного совместного проектирования программного и аппаратного обеспечения встроенных мультимикросистем. М.: Chip News. Инженерная микроэлектроника, 2002. – № 5. – С. 52-57.

2. Толкачёв А.И. Универсальный синтаксический анализатор // Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях, Материалы IV Республиканской научной конференции студентов и аспирантов 19-22 марта 2001. – С. 194-196.

3. Толкачёв А.И. Поиск алгоритмических конструкций, допускающих параллельное выполнение // Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях, Материалы V Республиканской научной конференции студентов и аспирантов 18-20 марта 2002. – С. 220-221.

4. Толкачёв А.И. Оптимизация кода для микроконтроллеров с параллельной архитектурой // Творчество молодых 2002. Сборник научных работ студентов и аспирантов Учреждения образования "Гомельский государственный университет имени Франциска Скорины", Гомель, 2002. – С.89-90.

Гомельский государственный  
университет им. Ф.Скорины

Поступило 9.04.2003

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ