

Архитектура, программное обеспечение и применения компьютеров серии «Эльбрус»

*А.К. Ким, Генеральный директор ОАО «ИНЭУМ им. И.С.Брука»,
В.Ю. Волконский, нач. отделения ОАО «ИНЭУМ им. И.С.Брука»,
Ф.А. Груздов, нач. отдела ЗАО «МЦСТ»,
М.С. Михайлов, нач. отдела ЗАО «МЦСТ»,
Ю.Н. Парахин, нач. отдела ЗАО «МЦСТ»,
Ю.Х. Сахин, нач. отделения ОАО «ИНЭУМ им. И.С.Брука»,
С.В. Семенухин, нач. отделения ОАО «ИНЭУМ им. И.С.Брука»,
М.В. Слесарев, нач. сектора ЗАО «МЦСТ»,
В.М. Фельдман, нач. отделения ЗАО «МЦСТ»*

Разработанные в России универсальные микропроцессоры с архитектурой «Эльбрус» обладают высокой логической скоростью при минимальной потребляемой мощности за счет явного параллелизма операций, многоядерной гомогенной и гетерогенной структуры, и оптимизирующего компилятора. Кроме этого микропроцессоры обеспечивают полную двоичную совместимость с архитектурой Intel x86 на базе технологии скрытой динамической двоичной трансляции, а также эффективные средства защищенного исполнения программ на базе аппаратных тегов и контекстной межмодульной защиты данных. Все эти свойства базируются на всестороннем использовании параллелизма и глубокой аппаратно-программной интеграции, позволяя создавать на базе микропроцессоров «Эльбрус» вычислительные комплексы широкого диапазона применений – настольные, встраиваемые, серверы, суперкомпьютеры.

1. Введение

Российские микропроцессоры серии «Эльбрус» базируются на новых архитектурных принципах, в основу которых положено всестороннее использование параллелизма, опирающееся на глубокую аппаратно-программную интеграцию. Первые микропроцессоры этого семейства успешно прошли Государственные испытания в 2007 г. Но их появлению предшествовал длительный этап разработки архитектуры, занявший более 20 лет.

Работы над новой параллельной архитектурой начались в 1986 г. в коллективе Института точной механики и вычислительной техники (ИТМ и ВТ) им. С.А. Лебедева, в котором до этого были созданы советские высокопроизводительные комплексы «Эльбрус-1» и «Эльбрус-2» [1]. Разработка вычислительного комплекса (ВК) «Эльбрус-3», которая велась под руководством Бабаяна Б.А., была завершена в 1991 г. В этом вычислительном комплексе впервые были воплощены в жизнь идеи

явного управления параллелизмом операций с помощью компилятора. «Эльбрус-3» разрабатывался как большая машина. Его центральный процессор занимал целый шкаф, т.к. при разработке использовалась советская микроэлектроника, которая значительно уступала зарубежной. На базе последней к тому времени уже было создано несколько поколений микропроцессоров. Начавшиеся с 1992 г. экономические изменения в России не позволили разработчикам «Эльбруса-3» завершить наладку комплекса. Стало ясно, что будущие поколения вычислительной техники должны базироваться на микропроцессорах. В том же 1992 г. коллектив разработчиков машин семейства «Эльбрус» выделился в компанию ЗАО «МЦСТ» и начал вести работы над микропроцессорной реализацией разработанной архитектуры.

В настоящее время компания ЗАО «МЦСТ», которая с 2007 г. интегрируется с ОАО «ИНЭУМ им. И.С. Брука» в отраслевой институт, разрабатывает универсальные микропроцессоры двух архитектурных линий: «Эльбрус» – для высокопроизводительных вычислений и «МЦСТ-R» – для встраиваемых применений. На базе этих микропроцессоров серийно выпускаются также разработанные в ЗАО «МЦСТ» ВК «Эльбрус-3М» и семейство ВК «Эльбрус-90микро», соответственно. Поскольку в микропроцессорах семейства «Эльбрус» используется принципиально новая архитектура, в которой значительную роль играет системное программное обеспечение, компания ЗАО «МЦСТ» разрабатывает операционные системы, оптимизирующие компиляторы, технологию оптимизирующей двоичной трансляции для эффективной совместимости с Intel x86 и средства обеспечения защищенного исполнения программ [2,3].

Микропроцессоры семейства «МЦСТ-R» производятся с 1998 г. За это время было создано 4 поколения микропроцессоров. Первые микропроцессоры выпускались на технологии 0,5 мкм, последние как система на кристалле – на 0,13 мкм, а их производительность за это время выросла с 22 до 400 Mflops при росте тактовых частот с 80 до 500 МГц.

Первый микропроцессор линии «Эльбрус» был выпущен на технологии 0,13 мкм с тактовой частотой 300 МГц, но при этом его производительность за счет параллелизма операций составила 2,4/4,8 Gflops для 64-/32-разрядных вещественных операций. Характерной особенностью всех этих микропроцессоров является постоянное улучшение показателя производительности на 1 Вт затрачиваемой мощности. В микропроцессорах линии «Эльбрус» этот показатель получается за счет аппаратно-программной интеграции, заложенной в его архитектуре, и составляет (400 Mflops/Wm). Эти микропроцессоры используются в ВК «Эльбрус-3М», который реализован как двухпроцессорный сервер и как двухпроцессорный встраиваемый модуль. Линия микропроцессоров

«Эльбрус» активно развивается и рассчитана на широкий диапазон применения, а именно: настольные компьютер и суперкомпьютер, встраиваемые системы, высокопроизводительные серверы, суперкомпьютеры.

2. Аппаратно-программная интеграция

Аппаратно-программная интеграция в микропроцессорах семейства «Эльбрус» базируется на строгом разделении действий между аппаратурой и системным программным обеспечением. В аппаратуре реализованы только те функции, которые необходимы для эффективного обеспечения с помощью оптимизирующих компиляторов высокой производительности, эффективной совместимости и защищенного исполнения программ.

Архитектура предоставляет *параллельные вычислительные ресурсы*, доступ к которым осуществляется с помощью широкой команды, позволяющей выполнить в одном такте до 23 обычных операций – арифметико-логических, обращений в память, передач управления. Параллелизм возрастает при использовании упакованных операций, которые позволяют выполнить одновременно несколько операций над упакованными данными меньшего размера в 64-разрядных регистрах. Так, например, в одном такте можно выполнить одновременно до 33 операций над 32-разрядными данными и существенно больше – над данными меньших форматов.

Архитектура обеспечивает *средства для оптимизации* обычных программ под столь высокий уровень параллельности операций. Эти средства включают:

- предикатные и спекулятивные операции, используемые для повышения параллелизма и устранения ложных зависимостей между операциями;
- базированные регистры и асинхронную предварительную подкачку данных, используемые для предельно эффективной конвейеризации циклов;
- операции подготовки переходов, позволяющие распараллеливать и оптимизировать передачи управления в программах.

Наряду со средствами распараллеливания на уровне операций архитектура предоставляет средства параллельного исполнения потоков управления на общей памяти с когерентным параллельным доступом, обеспечивая темп до 2 байтов на одну 64-разрядную вещественную операцию.

Архитектура предоставляет *средства, поддерживающие эффективную совместимость* с архитектурой Intel x86. Эти средства рассчитаны на технологию скрытой динамической двоичной трансляции кодов Intel

x86 (в которых представлена программа) в параллельные коды архитектуры «Эльбрус». Они включают в себя как поддержку самой технологии, так и поддержку наиболее эффективного отображения операций Intel x86 в операции архитектуры «Эльбрус».

Наконец, архитектура предоставляет *средства эффективного защищенного исполнения программ*. Они включают в себя аппаратную защиту указателей с помощью тегов, а также контекстную межмодульную защиту, ограничивающую каждую функцию набором доступных ей областей данных.

В свою очередь системное программное обеспечение и, в первую очередь, оптимизирующие компиляторы обеспечивают:

- автоматическое распараллеливание программ на уровне операций, в том числе и упакованных (векторизация), автоматическое распараллеливание на уровне потоков управления на общей памяти и автоматическое распараллеливание под распределенную память;
- технологию скрытой динамической трансляции любых программ, представленных в кодах Intel x86, включая любые операционные системы и приложения;
- реализацию языков и систем программирования для защищенного исполнения программ при поддержке операционной системы

Остановимся подробнее на особенностях архитектуры «Эльбрус».

3. Аппаратно-программные средства распараллеливания

Большинство современных микропроцессорных архитектур используют параллелизм операций для повышения производительности, т.к. этот вид параллелизма присутствует во всех без исключения классах программ. Самым распространенным является на сегодняшний день суперскалярный принцип распараллеливания операций. В архитектурах, использующих этот принцип, *аппаратно* распараллеливаются *последовательные* коды, которые получаются после компиляции программ. В результате аппаратуре приходится перекодировать сложно закодированные операции переменной длины в более простые операции, анализировать зависимости между этими операциями, чтобы обнаружить параллелизм, переименовывать регистры, чтобы избавиться от ложных зависимостей, и, наконец, планировать последовательные операции на параллельные устройства исполнения. Эта работа требует значительных аппаратных и энергетических затрат и не позволяет суперскалярным архитектурам исполнять более 4-х операций за такт из-за сложностей аппаратного распараллеливания.

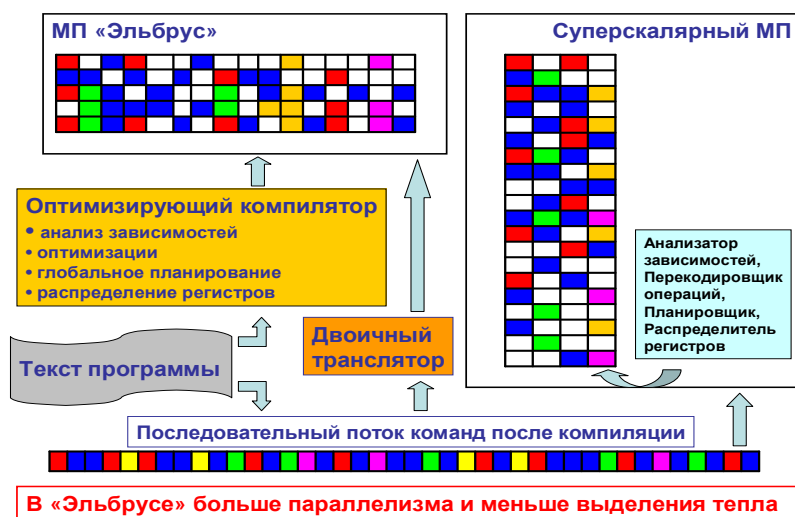


Рис. 1. Различия в подходе к распараллеливанию операций в микропроцессоре (МП) «Эльбрус» и в суперскалярном МП.

В отличие от суперскалярных архитектур управление параллельным исполнением операций в архитектуре «Эльбрус» задается явно с помощью широкой команды. При этом распараллеливание операций, анализ зависимостей, распределение регистров и планирование выполняются программно оптимизирующим компилятором (рис. 1). Это позволяет получить более параллельный код, который выполняется быстрее суперскалярного, и тратить на исполнение меньше энергии, переложив значительную часть работы с аппаратуры на компилятор.

Оптимизирующий компилятор для архитектуры «Эльбрус» (рис. 2) поддерживает входные языки C и C++, включая GNU-расширения этих языков для более полной совместимости со свободным программным обеспечением, а также Fortran-77. Эти языки реализуются для 32- и 64-разрядных адресов, а также для защищенного исполнения, о котором пойдет речь дальше. Важным входным языком является, так называемый, переносимый код. Он задействован в кодах стандартных библиотек для повышения производительности программ, позволяя оптимизировать их в контексте конкретных приложений. Еще одной важной функцией переносимого кода является обеспечение совместимости при развитии архитектуры семейства «Эльбрус».

Ключевым элементом оптимизирующего компилятора является блок оптимизации и генерации параллельного кода. Он обеспечивает глубо-

кий анализ зависимостей как внутри, так и между процедурами, результаты которого используются в оптимизациях и распараллеливании вычислений. В компиляторе реализованы принципиально новые и самые современные алгоритмы анализа и оптимизаций, многие из которых значительно усовершенствованы. Алгоритмы глобального планирования операций и распределения регистров объединяются со специфическими для архитектуры оптимизациями, а эвристики обеспечивают оптимальное планирование операций для различных регионов. Тем самым удается сокращать время выполнения до длин критических путей в регионах с ограниченным параллелизмом и достигать предельной загрузки аппаратных ресурсов, т.е. максимально использовать аппаратный параллелизм в регионах с большим параллелизмом, таких как конвейеризованные циклы.

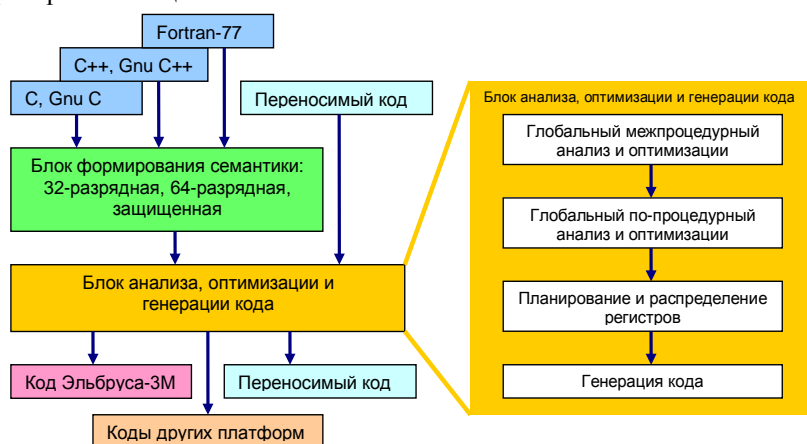


Рис. 2. Структура и основные функции оптимизирующего компилятора для МП «Эльбрус».

Архитектурные показатели микропроцессора «Эльбрус» – производительность 16/8 Гфлоп/ГГц для 32-/64-разрядных вещественных операций и отношение производительности к мощности 400 Мфлоп/Вт превосходят аналогичные показатели самых современных зарубежных микропроцессоров, являясь прямым следствием использования явного параллелизма операций с передачей функций их распараллеливания оптимизирующему компилятору. Проведенные в ходе Государственных испытаний замеры производительности на программных системах заказчика, представленных различными организациями, показали, что оптимизированные компилятором коды исполняются на ВК «Эльбрус-

3М» 300 МГц в среднем в 1,44 раза быстрее, чем на Pentium 4 1,4 ГГц. Эти результаты были в дальнейшем подтверждены на более широком классе задач пользователей ВК «Эльбрус-3М».

4. Аппаратно-программные средства совместимости с Intel x86

При создании новой микропроцессорной архитектуры нельзя абстрагироваться от проблемы совместимости с существующим программным обеспечением. На сегодняшний день наибольшее распространение в мире получила архитектурная платформа Intel x86 и ее 64-разрядные расширения, выпускаемые американскими компаниями Intel и AMD. В этой архитектуре используется сложно кодируемая последовательная система команд, причем владелец этой системы команд – Intel – не лицензирует ее никому в мире.

При разработке архитектуры «Эльбрус» была поставлена и успешно решена задача обеспечения полной совместимости с архитектурой Intel x86 без повторения ее системы команд. Для достижения этой цели была разработана технология скрытой двоичной трансляции программ из кодов x86 в коды архитектуры «Эльбрус». Эта программная технология существенно поддержана архитектурой микропроцессора «Эльбрус».

Упрощенная схема работы динамической системы двоичной трансляции ВК «Эльбрус-3М» представлена на рис. 3. Основными ее компонентами являются:

- интерпретатор кодов IA-32, который работает при первом исполнении, добавляя средства мониторинга;
- двоичный транслятор, который перекодирует коды IA-32 в коды микропроцессора «Эльбрус», используя для этого несколько уровней оптимизации;
- база кодов, обеспечивающая хранение и накопление оптимизированных кодов;
- система динамической поддержки, которая обслуживает весь процесс.

Для ускорения исполнения оптимизирующая трансляция запускается на свободном процессоре, а хорошо оптимизированные регионы сохраняются в базе кодов и при повторном исполнении той же программы используются без потери времени на их перекомпиляцию и оптимизацию.



Рис. 3. Схема обеспечения совместимости с помощью технологии двоичной трансляции.

Аппаратная поддержка технологии двоичной трансляции включает в себя общие средства распараллеливания программ на уровне операций, о которых было сказано ранее, средства поддержки самой технологии двоичной трансляции и средства поддержки совместимости с конкретной архитектурой x86. Ключевыми аппаратными средствами, обеспечивающими в ВК «Эльбрус-3М» эффективную и надежно работающую технологию двоичной трансляции, являются:

- полная изоляция всех компонентов двоично-транслирующей системы от исполняемой программы и ее данных;
- эффективное исполнение важнейших операций исходной платформы с соблюдением точной их семантики, в частности, операций обращения в память, для которых необходимо сохранение принятого порядка исполнения;
- воспроизведение точного состояния контекста исходной платформы (регистров и памяти) в момент прерывания исполнения (как синхронного, так и асинхронного), гарантирующее отсутствие потерь прерываний;
- полный контроль над возможными перемещениями и модификацией кодов исходной платформы при исполнении программы;
- поддержка специфических для двоичной трансляции методов оптимизации, в частности, скрытого канала для базы кодов.

Система двоичной трансляции обладает высокой надежностью. Она обеспечила успешное исполнение на ВК «Эльбрус-3М» более 20 операционных систем в кодах x86, в том числе MS DOS, несколько версий Windows (95, NT, 2000, XP и др.), Linux, FreeBSD, QNX. Под управлением этих операционных систем успешно и эффективно работают свыше 1000 популярных приложений, в том числе интерактивные компьютерные игры, программы из состава пакета MS Office (MS Word, MS Excel, MS PowerPoint и др.), видео ролики, программы компрессии/декомпрессии данных, драйверы всех внешних устройств. Наряду с возможностью запуска любых программ под управлением любых операционных систем в кодах x86 предусмотрен запуск приложений в кодах x86 под управлением ОС Эльбрус-Linux. Это позволяет одновременно использовать ВК «Эльбрус-3М» для запуска программ в кодах x86 и в кодах архитектуры «Эльбрус». Таким образом, в микропроцессоре «Эльбрус» успешно решена задача полной и эффективной совместимости с архитектурой Intel x86 при лицензионной независимости от Intel.

5. Аппаратно-программные средства для защищенного исполнения программ

На всех современных вычислительных системах адреса и числа неразличимы. Это дает большую свободу программистам при использовании адресов в ассемблерах и указателей в языках типа C и C++, но делает данные и сами программы незащищенными от программных ошибок и злонамеренных проникновений. В архитектуре «Эльбрус» в качестве указателей используются дескрипторы, которые защищены *аппаратными тегами* (рис. 4). Дескрипторы не могут быть сконструированы из отдельных частей и содержат информацию о размере объекта, на который указывают, и правах доступа к нему и к отдельным его компонентам. Благодаря этому в архитектуре «Эльбрус» контролируется любое обращение за границу объекта, описываемого дескриптором. Аппаратные теги используются также для обозначения неинициализированных данных, обращение к которым является очень распространенным источником ошибок в программах. Наконец, в аппаратуре реализован механизм *контекстной защиты*, который ограничивает каждую функцию набором доступных ей областей данных. Эффективность защищенного исполнения программ определяется тем, что все необходимые динамические проверки выполняются аппаратурой параллельно с выполнением операций и никак их не замедляют.

Все эти аппаратные средства позволяют реализовать наиболее распространенные языки C и C++ как абсолютно надежные. Для этого используются специальные методы компиляции и расширения операционной системы Эльбрус-Linux, опирающиеся на аппаратные средства поддержки защищенного исполнения программ. К сожалению, приме-

нение этих средств ко всем программам при их переносе на архитектуру «Эльбрус» ограничено, т.к. многие даже хорошо отлаженные программы содержат скрытые ошибки или используют опасные отклонения от стандарта языка, диагностируемые в режиме защищенного исполнения как ошибки.

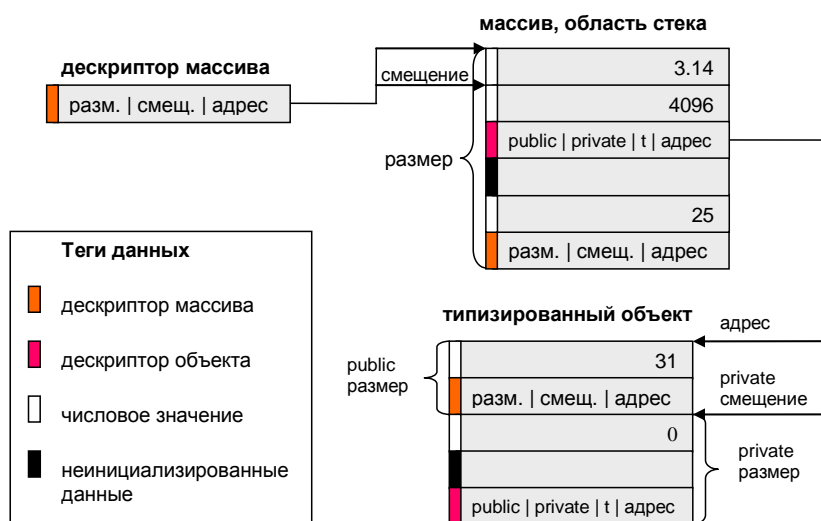


Рис. 4. Использование тегированной памяти для защиты данных.

Так около 50% программ пользователей и более 80% программ из пакета SPECint95, которые были оттранслированы и исполнены в защищенном режиме, выявили ошибки в этих программах – использование неинициализированных данных, выходы за границу объектов, опасные отклонения от стандартов языков, использование зависших ссылок. Вместе с тем, после отладки исправления ошибок в программах они могут столь же успешно исполняться на обычных машинах, где отсутствует само понятие защищенного исполнения, но при этом программы будут значительно более надежными.

6. Программное обеспечение ВК серии «Эльбрус»

Программное обеспечение, в первую очередь оптимизирующие компиляторы и операционная система играют важнейшую роль в производительности ВК серии «Эльбрус» «Эльбрус-3М», поскольку они обеспечивают распараллеливание программ на всех уровнях параллелизма. Поскольку работа технологии двоичной трансляции была

вкратце описана в 4, здесь будут рассмотрены оптимизирующий компилятор и операционная система, в основном, с точки зрения использования особенностей архитектуры «Эльбрус».

6.1. Оптимизирующий компилятор

Важнейшим компонентом в компиляторе является блок анализа, оптимизации и генерации кода, основные функции которого представлены на рис. 2. В нем выполняется анализ промежуточного представления программы, ее оптимизация, а также распараллеливание операций компилируемой программы (более 150 различных алгоритмов).

Методы анализа программ. Анализ программы необходим для вычисления при компиляции некоторых ее свойств, которые затем используются для оптимизации и распараллеливания операций.

Основная задача, которую решают методы статического анализа программ в оптимизирующих компиляторах – это определение отношения зависимости по данным и управлению между различными группами операций и отдельными операциями программы. Для обнаружения параллелизма в программах в компиляторе используются самые современные методы анализа зависимостей, многие из которых были значительно усовершенствованы, а некоторые – специально разработаны в компиляторе для архитектуры «Эльбрус».

Анализ управления выполняется на межпроцедурном и процедурном уровнях. На межпроцедурном уровне анализируется граф вызовов процедур, в нем находятся рекурсивные циклы, которые затем могут преобразовываться в обычные циклы. Основной задачей анализа управления на процедурном уровне является обнаружение циклов и ациклических регионов, в которых, в основном, работает программа. Именно эти регионы затем оптимизируются наиболее интенсивно.

Анализ потоков данных также выполняется на межпроцедурном и процедурном уровне. На межпроцедурном уровне вычисляются свойства параметров и результатов функций, такие, как константное значение или ссылка на конкретную переменную или функцию для указателей на переменные и функции. Эта информация затем используется в по-процедурном анализе для уточнения зависимостей между операциями или в оптимизациях, например, для клонирования процедур с последующей их непосредственной подстановкой. Анализ потока данных внутри процедуры в основном также анализирует зависимости между операциями, а кроме этого вычисляет свойства переменных, такие как константное значение, общие подвыражения, а для циклов – индуктивные переменные и возможность совпадения индексов при обращении к элементам массивов на разных итерациях. Все эти свойства также затем используются в оптимизациях и для распараллеливания операций.

В тех случаях, когда невозможно точно определить отношение зависимости между операциями при компиляции программы используются достаточно мощные средства динамического анализа, как программные, так и аппаратные, которые считаются оптимизациями.

Важнейшие оптимизации. Важнейшими оптимизациями наряду с распараллеливанием операций являются универсальные преобразования, устраняющие ненужные вычисления, а также оптимизации, сокращающие число блокировок при обращении за данными и кодами из-за наличия иерархии памяти.

Методы *устранения избыточных вычислений* включают свертку константных выражений, сбор общих подвыражений, удаление ненужных копирований данных, удаление лишних обращений в память за считыванием и записью данных, вынос инвариантов из циклов, упрощение индексных выражений, вычисление и сокращение индуктивных переменных, удаление «мертвого» (ненужного) кода и многие другие. Эти оптимизации, как правило, применяются во всех оптимизирующих компиляторах, но в компиляторе для ВК «Эльбрус-3М» они существенно развиты в части использования предикатов для управления операциями.

Оптимизации управления, увеличивающие параллелизм, – это непосредственная подстановка функции в точку вызова, которой может предшествовать клонирование (создание копии) функции, если она вызывалась через указатель, раскрутка цикла, конвейеризация цикла, формирование гиперблоков с преобразованием операций передачи управления в предикаты и многое другое.

К важнейшим *аппаратным средствам борьбы с зависимостями*, реализованным в микропроцессоре «Эльбрус», относятся предикатное и спекулятивное исполнение операций, аппаратные средства конвейеризации циклов, аппаратные средства сравнения адресов при изменении порядка обращений в память.

Предикатное исполнение операций позволяет компилятору планировать одновременно несколько путей управления в программе. Такие пути могут исполняться одновременно и бесконфликтно, поскольку каждая операция стоит под предикатом, в точности повторяющим условие, при котором операция была бы выполнена в исходной программе. Нулевое значение предиката отменяет выполнение операции, обеспечивая правильную семантику выполнения программы. Применение предикатных операций позволяет избавиться от многих локальных переходов в пределах региона планирования операций. *Спекулятивное исполнение операций* избавляет операции от предикатных зависимостей и позволяет выполнять их раньше, чем становится известно значение предиката. *Аппаратные средства конвейеризации циклов* позволяют исполнять

одновременно операции с разных итераций цикла, а средства базирования регистров позволяют полностью избавиться от зависимостей, вызванных переиспользованием регистров. *Аппаратные средства сравнения адресов при изменении порядка обращений в память* позволяют компилятору выполнять операции чтения раньше операций записи, сокращая тем самым критические пути и ускоряя исполнение. В случае обнаружения реальной зависимости между переставленными операциями аппаратно вырабатывает признак, позволяющий передать управление на компенсирующий код, заранее подготовленный компилятором для восстановления правильного исходного порядка исполнения операций. После выполнения компенсирующего кода корректное исполнение программы восстанавливается.

Можно выделить три основных стратегии *сокращения потерь производительности при обращении в память*, которые составляют 20-50%, а на отдельных задачах – до 80%: сокращение числа обращений, заблаговременная подкачка данных и кода, оптимальное размещение данных.

Наряду с оптимизацией *удаления повторных обращений в память* в пределах региона оптимизации, число обращений в память сокращается за счет *объединения нескольких обращений в одно* при обработке смежно-расположенных в памяти данных, совокупный размер которых не превышает размер аппаратного регистра (64 разряда). Такая оптимизация, дополненная операциями над упакованными значениями (операции типа MMX/SSE) позволяет существенно поднять производительность широкого класса задач, включая универсальные приложения.

Предварительная подкачка данных в буферные памяти или в КЭШ является эффективным средством оптимизации обращений в память. Наряду с предварительной подкачкой данных в КЭШ компилятором используются аппаратные средства предварительной *асинхронной* подкачки элементов массивов в специальный буфер. Эти средства полностью снимают проблему приостановки вычислений ввиду неизвестности времени доступа в память, поскольку асинхронная подкачка обеспечивает оптимальную адаптацию к произвольному времени доступа. Такой подход дает существенный прирост производительности, особенно на задачах с вещественной арифметикой, в которых доминируют циклы.

Еще одной стратегией уменьшения потерь от обращений в память является *оптимальное размещение данных*. Линейная память, используемая для хранения данных, а также ограниченные размеры, строковая организация и ограниченная ассоциативность КЭШ создают проблемы, известные как фрагментация, и приводят к возникновению

дополнительных конфликтов. Практические методы оптимизации размещения данных в памяти, реализованные в компиляторе для ВК «Эльбрус-3М», позволили сократить потери как на обращение за данными, так и за кодами.

Планирование и распределение регистров. Важнейшую роль в получении оптимальных параллельных кодов играет планирование операций и распределение аппаратных регистров. Роль планирования сводится к оптимальному отображению абстрактного параллельного представления программы в виде графа зависимостей на параллельные ресурсы архитектуры микропроцессора.

С точки зрения стратегий планирования все регионы можно разделить на три категории: *ациклические*, *конвейеризируемые циклы* и *смешанные* (регионы со сложным управлением).

Ациклический регион, как правило, представляет собой совокупность базовых блоков с одним входом и многими выходами. Все внутренние переходы внутри него удалены и заменены предикатным управлением (это гиперблок) с использованием предикатных и спекулятивных операций. Планирование выполняется с помощью усовершенствованного алгоритма планирования списков готовых операций. Благодаря наличию большого набора параллельных устройств исполнения в широкой команде архитектуры «Эльбрус», с одной стороны, и, как правило, ограниченного параллелизма операций в гиперблоке – с другой, алгоритм планирования списков позволяет отобразить абстрактное параллельное представление программы на конкретные аппаратные ресурсы, не увеличивая критические пути.

Конвейеризируемый цикл – это регион, позволяющий совмещать выполнение нескольких последовательных итераций цикла. Такой способ планирования применяется к самым внутренним циклам, которые повторяются много раз (от нескольких десятков до сотен и более). В архитектуре «Эльбрус» он имеет мощную аппаратную поддержку, включающую асинхронную предварительную подкачку данных, базируемые регистры, управление прологом и эпилогом. Конвейеризация циклов с дает значительный прирост производительности (в 2,5-3 раза на задачах с интенсивной обработкой массивов в циклах) по сравнению с чисто программными методами конвейеризации.

Смешанный регион или регион со сложным управлением представляет собой совокупность ациклических и цикловых регионов, в которых работа внутри циклов не столь продолжительна, как для конвейеризируемых циклов. К этим регионам применяются методы глобального планирования, соединяющие воедино оптимизацию, распараллеливание и планирование операций. В основе глобального планирования лежит перенос операций через точки схождения управления. При этом учиты-

ваются влияние переноса операций на изменение критических путей выполнения различных частей региона планирования, включая циклы и ациклические участки. В конечном итоге циклы оказываются частично конвейеризованными, и при этом в них самих и в ациклических участках активно используются предикатные и спекулятивные операции. Методы глобального планирования повышают эффективность исполнения регионов со смешанным управлением на 20-50% по сравнению с независимым планированием циклов и ациклических регионов.

Оптимальное *распределение регистров* является важнейшей составляющей процесса оптимального планирования и распараллеливания операций, поскольку все операции берут свои операнды из регистров и помещают результат в регистр. На уровне абстрактного представления программы используются виртуальные регистры, число которых никак не ограничивается. В реальном микропроцессоре число регистров ограничено. Если в процессе планирования не удастся все результаты операций поместить на аппаратные регистры, то появляются дополнительные операции перемещения содержимого регистров в память, что замедляет выполнение программы. В оптимизирующем компиляторе для ВК «Эльбрус-3М» разработан и реализован уникальный алгоритм *одновременных планирования и распределения регистров*. Он повышает эффективность распределения регистров на 10-20% по сравнению с другими самыми современными методами распределения регистров.

6.2 Операционная система

Основная работа по созданию операционной системы (ОС) для ВК "Эльбрус-3М1" заключалась в создании архитектурно зависимой части ядра ОС, которая является главным и наиболее сложным элементом базовой операционной системы. Архитектурно зависимая часть ядра реализована на базе ОС Linux версии 2.6.14. Далее рассматривается информация по компонентам, которые потребовали наибольшей работы и усилий при разработке архитектурно зависимой части ОС.

Управление обработкой прерываний. В архитектуре "Эльбрус" определены два вида прерываний: программные и аппаратные. Типичным представителем программного прерывания является вход в интерфейсную процедуру ОС. Типичными представителями аппаратных прерываний являются внутренние прерывания, возникающие при выполнении команд в процессоре, и внешние – прерывания от внешних устройств.

Архитектура широкой команды предполагает возможность появления нескольких внутренних прерываний при исполнении широкой команды. В ряде случаев нет возможности повторить всю широкую команду после обработки прерывания, как это происходит в машинах с

традиционной архитектурой. Это привело к необходимости создания механизмов интерпретации команд с использованием аппаратных средств, в которых сохраняется информация, необходимая для интерпретации.

Управление распределением памяти. В основе работы виртуальной памяти лежит возможность аппаратуры работать с четырехуровневой таблицей виртуальной памяти. Для работы аппаратуры ОС должна построить все уровни таблицы страниц и затем правильным образом формировать элементы этих таблиц, например, при обработке прерывания по отсутствию страницы в таблице страниц.

Управление процессами. Одним из основных действий в управлении процессами является создание процесса. Создание процесса возможно из ядра и из программы пользователя. В архитектуре "Эльбрус" работа по исполнению программ происходит с использованием 3-х стеков: стека компилятора, стека процедур (регистровых окон) и стека связующей информации. ОС поддерживает работу этих стеков. Стеки компилятора для всех процессов ядра размещаются в памяти ядра. Стек процедур и стек связующей информации недоступны программам пользователей, и на этом строятся очень полезные оптимизации управления процессами. В данном случае нет необходимости создавать эти стеки в памяти ядра ОС и можно работать на стеках, расположенных в пространстве пользователя.

Загрузка и запуск программ с поддержкой защищенного исполнения. В архитектуре "Эльбрус" понятие программного модуля поддержано на уровне аппаратуры. Естественно, что в такой ситуации работа по загрузке модуля и его активизация – это новое свойство операционной системы. Дополнительных усилий потребовала также поддержка защищенного исполнения программ, о которых говорилось в 5.

Средства синхронизации параллельных вычислений. Средства синхронизации всегда базируются на некоторых основных примитивах, реализация которых зависит от архитектуры. Архитектура «Эльбрус» здесь не стала исключением. Все примитивы – это новые примитивы синхронизации. Необходимо отметить, что аппаратная поддержка позволила реализовать эти примитивы максимально эффективно – только с запретом обращения по определенному адресу.

Управление работой внешних устройств. Управление работой внешних устройств определяется работой контроллера прерываний и контроллеров самих устройств. В ВК "Эльбрус-3М" набор системных контроллеров близок к общепринятым, но собственная реализация неизбежно привела к некоторым отличиям, которые необходимо было учитывать при адаптации соответствующих драйверов. Следует отме-

титель, что она потребовала больших усилий, чем разработка новых драйверов.

Поддержка режима реального времени. Операционная система Linux – типичный представитель серверных операционных систем, где все ориентировано на повышение средней производительности. Такая ориентация приводит к недетерминированному времени исполнения различных действий. Это положение недопустимо для систем реального времени, что потребовало разработать и реализовать новые принципы управления при включении режима реального времени в целях обеспечения необходимых временных показателей. Основным принципом является необходимость исполнять любые действия в соответствии со стратегией приоритетного планирования. В стандартном Linux это далеко не так. Например, обработка прерываний и работа основной процедуры управления (handler) происходит с приоритетом процесса, на который пришло прерывание. Для реального времени это неизбежно приводит к большим временным разбросам исполнения каких-то действий, что недопустимо. В результате планомерной работы удалось практически полностью избавиться от временных разбросов, улучшив в этой части показатели OS Linux со стандартными, доступными на www.kernel.org изменениями (rt_patch) почти в 30 раз [4].

Поддержка режима двоичной трансляции. В разработанном ядре ОС реализовано уникальное свойство поддержки исполнения кодов IA-32 в режиме двоичной компиляции. В ВК "Эльбрус-3М1" существует аппаратная поддержка специальной памяти (вторичного виртуального пространства), характерной для архитектуры IA-32. В ядре ОС реализовано отображение вторичного виртуального пространства в основное, что позволяет двоичному транслятору обращаться во вторичное пространство обычными командами (load/store) с обработкой возникающих прерываний обычными средствами ОС. Это свойство используется двоичным транслятором приложений в кодах IA-32 под управлением стандартной ОС.

Управление сетевым взаимодействием. Сетевые протоколы и их реализация остались без изменений. Были разработаны специальные средства для работы в режиме реального времени с возможностью привязки обработки прерываний (от конкретного сетевого устройства) к одному процессу. В стандартном Linux эта работа масштабируется по процессорам, что совершенно не эффективно для ОС реального времени.

Средства инициализации и перезапуска. Средства инициализации и перезапуска можно разделить на две части: загрузка ОС (boot) и инициализация ОС. Оба этих компонента реализованы не только для последующей работы, но и с возможностью быстрого восстановления при

перезапуске системы. В ВК "Эльбрус-3М" имеется возможность не прерывать режим регенерации оперативной памяти при перезапуске системы. Это дало возможность реализовать контрольную точку перезапуска не только на диске, но и в памяти. Как следствие, в самом быстром варианте перезапуск выполняется за несколько секунд.

7. Применение ВК семейства «Эльбрус»

Вычислительные комплексы на базе микропроцессоров «Эльбрус» предназначены для использования в стационарных и мобильных (перебазированных) системах управления обработки информации.

В настоящее время ВК «Эльбрус-3М» выполняется в двух вариантах конструктивного исполнения: серверном варианте, пригодном также для использования в качестве настольного, и в исполнении compact PCI.

Двухпроцессорные ВК семейства «Эльбрус» поддерживают работу с современными средствами ввода-вывода, работу в сетях и обеспечивают производительность свыше 4,5 Гфлпс для 64-разрядных данных и свыше 8,5 Гфлпс для 32-разрядных данных. Они могут использоваться как настольные машины. Они имеют средства быстрого межмашинного обмена, что позволяет строить разнообразные конфигурации для применения в качестве серверов и мощных мобильных систем.

Таблица 1. Производительность ВК «Эльбрус-3М» на задачах из области суперкомпьютеров (данные запуска задачи на одном процессоре)

Реальные замеры / пиковые возможности	DGEMM Гвешоп./сек.	STREAM Гбайт/сек.	FFT Гвешоп./сек.
64-разрядные данные	2,3 / 2,4	4,6 / 4,8	1,5 / 1,5
32-разрядные данные	4,5 / 4,8	4,6 / 4,8	2,8 / 3,0

ВК семейства «Эльбрус» позволяют на задачах из области суперкомпьютеров достигать производительности, близкой к пиковым характеристикам процессоров (табл.1). Это достигается за счет хорошей сбалансированности вычислительных возможностей микропроцессоров и пропускной способности памяти.

8. Перспективы развития линии «Эльбрус»

Работа над микропроцессорами семейства «Эльбрус» и компьютерами на базе этих микропроцессоров является важнейшей для компании ЗАО «МЦСТ». В настоящее время завершаются работы над созданием одноядерной системы на кристалле «Эльбрус-S» на базе технологии 90 нм. В активной фазе находится разработка гетерогенного 6-ядерного микропроцессора с двумя универсальными ядрами «Эльбрус» и четырьмя специализированными ядрами для обработки сигналов. Этот микропроцессор также разрабатывается на технологии 90 нм и предна-

значен для использования в различных встраиваемых системах. Оба микропроцессора будут работать на тактовой частоте 500 МГц. Начаты работы над гомогенным 4-ядерным микропроцессором «Эльбрус-2S» на технологии 65 нм, который также будет реализован как система на кристалле, работающая на частоте 1 ГГц.

В части развития работ над микропроцессорами предполагается их реализация на технологиях 45 нм, 32 нм, 22 нм и более высоких по мере их появления. При этом ставится задача предельно оптимизировать логическое и физическое проектирование, обеспечив достижение *тактовых частот выше 4 ГГц*. Предполагается также дальнейшее развитие архитектуры широкого командного слова, используемой в ядре микропроцессора, чтобы обеспечить в будущем производительность *выше 16 Gflops/ГГц* на ядро. При этом число процессорных ядер в гомогенных системах должно возрасти до 16 на один микропроцессор, а в гетерогенных – возможно и больше. За счет активного применения энергосберегающих технологий при проектировании микропроцессоров предполагается на порядок увеличить производительность к выделяемой мощности и довести его *до 4 Gflops/Wm*. В микропроцессоры будут активно встраиваться быстрые каналы и контроллеры памяти, поддерживающие создание многопроцессорных и много машинных систем, а также средства, обеспечивающие достоверность и надежность вычислений. Все это позволит выйти на уровень *микропроцессоров с терафлопной* производительностью.

В части развития вычислительных комплексов предполагается создание много процессорных систем на общей памяти с целью создания мощного серверного узла, содержащего до 64 процессоров и до 1024 процессорных ядер. При этом большое внимание предполагается уделять средствам параллельного доступа в память, обеспечивая темп 2 байта на 64-разрядную вещественную операцию. Это позволит получить *серверы терафлопного диапазона*, а объединение столь мощных серверных узлов в многомашинный комплекс позволит создать *суперкомпьютеры петафлопного диапазона*. Эти вычислительные комплексы будут поддерживаться эффективными автоматическими средствами распараллеливания задач на многопроцессорные и многомашинные конфигурации. Предполагается также создать специальные системы контроля состояния подобных вычислительных комплексов, облегчающие их обслуживание, и снабдить их средствами автоматической реконфигурации для поддержания высокой надежности и готовности. Все это позволит сократить или даже ликвидировать отставание в стратегически важной области и повысит технологическую независимость и информационную безопасность.

9. Заключение

Подводя итоги, хотелось бы еще раз перечислить основные свойства, которые обеспечивают и предоставляют пользователям процессоры семейства «Эльбрус» и их программное обеспечение:

- высокую производительность вычислений;
- возможность использования всех видов параллелизма;
 - параллелизм операций,
 - дополнительный параллелизм упакованных операций (короткие вектора),
 - параллелизм нескольких процессорных ядер и процессоров, работающих на общей памяти с когерентным доступом,
 - параллелизм систем с распределенной памятью;
- экономное энергопотребление;
- современное программное обеспечение, глубоко интегрированное с архитектурой микропроцессора;
- полную и эффективную двоичную совместимость с архитектурой Intel x86 при лицензионной независимости от компании Интел;
- средства разработки и эффективной отладки надежных программных систем большими коллективами в сжатые сроки;
- конкурентоспособные аппаратные и программные средства при массовом их производстве;
- технологическую независимость и информационную безопасность.

Литература

1. Бабаян Б.А., Сахин Ю.Х. Система Эльбрус. // Программирование. 1980. № 6.
2. Ким А.К., Волконский В.Ю., Груздов Ф.А., Михайлов М.С., Парухин Ю.Н., Сахин Ю.Х., Семенихин С.В., Слесарев М.В., Фельдман В.М. Микропроцессорные вычислительные комплексы с архитектурой «Эльбрус» и их программное обеспечение. // Вопросы радиоэлектроники, сер. ЭВТ, 2009, вып. 3. С.5-37.
3. Волконский В., Груздов Ф., Ким А., Сахин Ю. «Эльбрус» сегодня // Открытые системы. 2009. № 2. <http://www.osp.ru/os/2009/02/7314081/>
4. Семенихин С.В., Ревякин В.А., Ананьев Л.И., Костин Д.А., Гилязов С.С., Харитонов М.И., Ситников А.В., Чернис Е.Н., Лебедев В.П. ОС Linux и режим реального времени. // Вопросы радиоэлектроники, сер. ЭВТ, 2009, вып. 3. С. 37-67.