

Основные направления развития архитектуры универсальных микропроцессоров

Развитие микропроцессорной техники в области универсальных микропроцессоров идет по пути постоянного повышения их производительности. Традиционными направлениями такого развития являются повышение тактовой частоты работы МП и увеличение количества одновременно выполняемых команд за счет увеличения числа конвейеров (исполнительных устройств) в МП.

Однако оба эти направления следует признать экстенсивными, имеющими естественные ограничения.

Повышение тактовой частоты, которое в основном обеспечивается путем увеличения количества ступеней в конвейере, приводит к большим потерям времени при необходимости перезагрузки конвейера вследствие конфликтов по управлению или при переключении на новую задачу. Такое увеличение имеет также и физические ограничения, связанные со схемотехникой кристалла БИС. Ограничения определяются также влиянием накладных расходов при передаче частично обработанной команды на следующую ступень конвейера (значение t при определении длительности такта). На больших частотах эти расходы становятся соизмеримыми с длительностью обработки на очередном этапе. Во многом это направление исчерпало себя в микропроцессоре Pentium 4, работающем на частотах, близких к 4 ГГц.

Повышение производительности за счет увеличения числа конвейеров в микропроцессоре можно оценить увеличением числа команд, выполняемых программами за такт (IPC - INSTRUCTIONS Per Cycle). Так, для МП Alpha 21264 этот показатель равен 6, столько же микроопераций за такт может выдать Pentium 4. Но это предельные значения, а реальные программные коды, в частности, из-за различных взаимозависимостей, дают гораздо более низкое значение IPC. Дальнейшее увеличение числа исполнительных устройств ведет к усложнению расположенного в БИС устройства управления, распределяющего команды по конвейерам, а также к сложным взаимозависимостям между данными. К тому же реальные коды программ не позволяют обеспечить эффективную загрузку всех имеющихся в МП исполнительных устройств, что приводит к их простоям. Следует отметить также, что рост производительности микропроцессора не является прямо пропорциональным росту количества конвейеров, а обычно существенно ниже. ___

В настоящее время для повышения производительности микропроцессоров используется ряд новых подходов, основными из которых являются:

- **CMP (Chip Multi ProcessING)** - создание на одном кристалле системы из нескольких микропроцессоров (многоядерность);
- **SMT (Simultaneous MultiThreadING)** - многонитевая архитектура;
- **EPIC (Explicitly Parallel INSTRUCTION ComputING)** - вычисления с явным параллелизмом в командах.

Рассмотрим эти направления подробнее.

1. **Направление CMP** обеспечивается возросшими технологическими возможностями, которые позволяют создать на одном кристалле несколько микропроцессоров и организовать их работу по принципу мультипроцессорных систем.

Производители чипов уже не гонятся за частотой, сместив акцент на многоядерную архитектуру, которая позволяет наращивать производительность, сохраняя в приемлемых границах энергопотребление и тепловыделение.

Многоядерные процессоры хорошо приспособлены для требовательных мультимедийных задач, таких как обработка видеозаписей, работы с большими базами данных, одновременное выполнение нескольких ресурсоемких заданий, например, компьютерной игры, записи DVD и загрузки файлов из Интернета.

При таком подходе задача повышения производительности работы отдельных приложений требует распараллеливания последних, то есть проблема перемещается с аппаратного на программный уровень. На данный момент сложности заключаются в том, что большая часть существующего программного обеспечения создавалась без расчета на использование в многоядерных и

многопроцессорных конфигурациях. Другими словами, прогресс в области аппаратных средств на какое-то время опередил прогресс в области программного обеспечения.

Развитие микропроцессорной техники в этом направлении идет очень быстрыми темпами. Так, компания Tiler в 2007 году начала поставки специализированных процессоров Tile64, насчитывающих 64 ядра. А в 2009 году, как ожидается, свет увидит 120-ядерная модификация этого процессора.

80-ядерный процессор от фирмы INTel

Еще в 2001 году руководство фирмы Intel уверенно обещало дотянуть одноядерную архитектуру процессоров Pentium до 2010 года с поднятием планки их рабочей частоты до 10 ГГц. Однако уже в 2008 году Intel объявила о прекращении выпуска одноядерных микропроцессоров. Новый 80-ядерный процессор от Intel, который в некоторых документах имеет обозначение Polaris, обеспечивает производительность до 1,28 триллиона операций с плавающей точкой в секунду (терафлопс) при частоте 4 ГГц.

Микропроцессор построен по технологии 65 нм из 100 млн транзисторов на кристалле площадью 275 мм². Для сравнения: двухъядерный Intel Core 2 Ex TR eme, также с технологическими нормами 65 нм, содержит 291 млн транзисторов на площади 143 мм². Разница обусловлена малым количеством памяти на кристалле нового МП и доминированием логических схем и схем ввода-вывода, размер которых трудно минимизировать. Большое внимание в нем уделено новому показателю, который занимает все более главенствующее положение, - производительность/ватт. По этому показателю Polaris обеспечивает чрезвычайную энергоэффективность: 16 гигафлопс/Вт. Процессор работает при напряжении питания 0,95 В.

Согласно заявлению представителей Intel, достигнутые показатели отнюдь не являются предельными. Инженеры компании могут значительно увеличить производительность процессора за счет увеличения его тактовой частоты. Так, например, на частоте 5,1 ГГц процессор демонстрирует 1,63 терафлопс, а на частоте 5,7 ГГц - 1,81 терафлопс. Однако с ростом тактовой частоты растет и потребляемая мощность, составляя при вышеуказанных режимах работы 175 и 265 ватт соответственно.

Polaris представляет собой 80 одинаковых процессорных элементов, каждый из которых состоит из вычислителя и маршрутизатора на 6 портов. Вычислитель содержит 2 устройства для операций умножения с накоплением (FP MultiPle-Accumulator, FPMAC), 3 Кбайт памяти команд и 2

Кбайт памяти данных, набор 32-разрядных регистров и оперирует 96-разрядными сверхдлинными словами команд (Very Long INsTRuction Word -VLIW). Планирование и параллельное выполнение команд возлагается на компилятор (архитектура EPIC, аналогичная реализованной разработчиками Intel и в процессоре Itanium). Она позволяет процессору исполнять до восьми команд за один такт. По своему набору инструкций новый микропроцессор несовместим с x86.

Все арифметические операции выполняются на FPMAC-модулях.

Эффективность многоядерных процессоров во многом определяется пропускной способностью памяти системы. Применение многоканальных шин памяти и повышение их частоты сопряжено с чрезмерным усложнением контроллеров. Другой путь уменьшения задержки - ввести еще один уровень иерархии кэш-памяти. Однако это приведет к неоправданному увеличению памяти на кристалле, площадь которого эффективнее использовать для вычислительной логики. Несколько мегабайт дополнительной кэш-памяти занимают площадь, достаточную для 10 процессорных ядер. Такой подход обеспечит широкий доступ ядер процессора к памяти. Каждый слой будет содержать до 256 Мбайт.

Другой фактор, ограничивающий производительность современных компьютеров, - пропускная способность процессорной шины. Разделяемые шины уже уступают свое место соединениям типа "точка-точка". В Polaris используется сеть процессорных элементов, основанная на передаче данных с коммутацией каналов. Каждый маршрутизатор содержит шесть 39-битных портов и обеспечивает

суммарную пропускную способность до 80 Гбайт/с при частоте микросхемы 4 ГГц. Четыре порта ведут к смежным процессорным элементам, один к вычислителю, а шестой порт в перспективе будет взаимодействовать с многослойной памятью. Система коммутации предусматривает взаимодействие каждого маршрутизатора с каждым. В микропроцессоре ядра независимы в плане операций ввода-вывода. Поэтому, добавив необходимое ПО, можно реализовать различные интерфейсы и добиться любой функциональности устройства.

В настоящее время главное достижение Polaris - это создание фундамента (как аппаратного, так и идейного) для отработки новых технологий многоядерных архитектур. Он представляет собой скорее исследовательский проект, чем инженерное решение.

2. **Направление SMT** в развитии архитектуры микропроцессоров базируется на том, что одна задача не в состоянии полностью загрузить все возрастающие ресурсы микропроцессора. Поэтому на одном процессоре осуществляется запуск нескольких задач одновременно, при этом распараллеливание программ осуществляется аппаратными средствами МП.

Это позволяет более равномерно загрузить ресурсы процессора. Параллельно в разных устройствах МП могут выполняться команды из разных задач. Так, микропроцессор Alpha 21264 поддерживает выполнение до 4 задач одновременно. При поддержке SMT на 4 нити каждый процессор с точки зрения операционной системы выглядит как 4 логических процессора. Исследования показали, что SMT позволяет увеличить производительность данного процессора до двух раз, а дополнительные схемы управления занимают всего около 10 % площади кристалла.

Некоторые микропроцессоры для максимального повышения своей производительности используют оба вышеназванных подхода. Так, компания Sun Microsystems представила новый процессор ULTRaSPARC T2.

Новый чип снабжен восемью ядрами, каждое из которых может обрабатывать восемь потоков инструкций. Таким образом, он одновременно способен оперировать с 64 потоками. Тактовая частота ULTRaSPARC T2 составляет от 900 МГц до 1,4 ГГц.

3. **Направление EPIC** фактически использует известную технологию **VLIW (Very Large INstruction Word)** - очень длинного командного слова.

Распараллеливание алгоритма между исполнительными модулями производится компилятором на этапе создания машинного кода, когда команды объединяются в связки и не конкурируют между собой за ресурсы микропроцессора. При этом упрощается блок управления на кристалле.

Особенности архитектуры EPIC:

1. Явный параллелизм в машинном коде. Поиск зависимостей между командами проводит не процессор, а компилятор.
2. Большое количество регистров.
3. Масштабируемость архитектуры до большого количества функциональных устройств (АЛУ, FPU, MMX, SSE и т. п.).
4. Применение предикатов. Предикатный подход исходит из предпосылки, что возросшие мощности микропроцессоров позволяют запускать параллельно команды из разных ветвей условного ветвления вместо того, чтобы ожидать формирования истинных признаков для выбора правильного направления или полагаться на блок предсказания переходов, рискуя прийти к необходимости перезагрузки достаточно длинных конвейеров в случае неудачного предсказания. При этом каждая команда снабжается специальным полем условия (предикатом) (рис. 11.1). По мере определения истинных признаков ветвления те команды, предикаты которых указывали, что они выбраны из другой ветви, снимаются с обработки в конвейере. Результаты команд не записываются в приемник до определения правильности направления перехода.

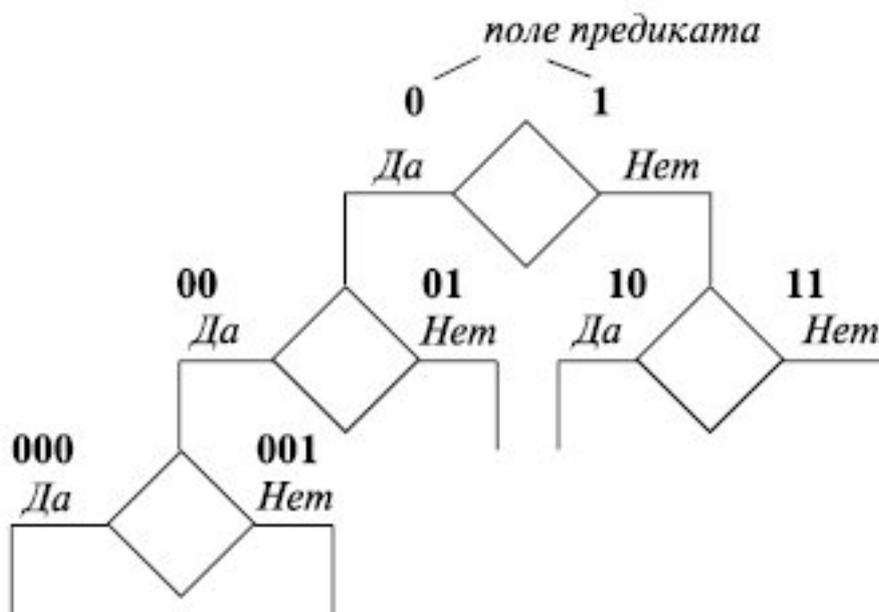


Рис. 11.1. Предикатное исполнение команд

Отметим **основные достоинства** этого подхода:

1. Упрощается архитектура процессора. Вместо логики распараллеливания на EPC-процессоре можно разместить больше регистров, функциональных устройств и т. п.
2. Процессор не тратит время на анализ потока команд.
3. Возможности процессора по анализу программы во время выполнения ограничены сравнительно небольшим участком программы, тогда как компилятор способен произвести анализ всей программы.
4. Если некоторая программа должна запускаться многократно (а именно так и бывает в подавляющем большинстве случаев), выгоднее распараллелить ее один раз при компиляции, а не тратить на это время каждый раз, когда она исполняется на процессоре.

Однако архитектуре EPC присущ и ряд **недостатков**:

1. Компилятор производит статический анализ программы, раз и навсегда планируя вычисления. Однако даже при небольших изменениях исходных данных путь выполнения программы существенно изменяется.
2. Значительно усложняются компиляторы, следовательно, увеличиваются время компиляции программы и число ошибок в самих компиляторах. Если первый фактор, учитывая высокое быстродействие современных компьютеров, не очень существенен, то на второй следует обратить определенное внимание. Исследования показывают, что к моменту поставки даже ответственного программного обеспечения в нем содержится примерно 1 ошибка на 10 000 строк исходного кода. Следовательно, программа из 500 тыс. строк будет содержать около 50 ошибок, как бы хорошо ни работали тестировщики. И эти ошибки могут проявиться самым неожиданным образом.
3. Производительность микропроцессора во многом определяется качеством компилятора. Правда, здесь необходимо четко определить, что понимается под производительностью, ведь количество операций, выполняемых микропроцессором в единицу времени, от компилятора не зависит. Но это тема будет особо рассмотрена далее.
4. Увеличивается сложность отладки, так как отлаживается не исходная программа, а оптимизированный параллельный код. Программисту тяжело определить место и причину появления ошибки, так как в процессе трансляции исходной программы ее отдельные команды будут переставлены компилятором для обеспечения оптимальности работы микропроцессора.

Типичным представителем архитектуры EPC является микропроцессор Itanium фирмы Intel.

Архитектура микропроцессора Itanium

Данный микропроцессор относится к новой, 64-разрядной архитектуре IA-64. Структура микропроцессора Itanium представлена на рис. 11.2.

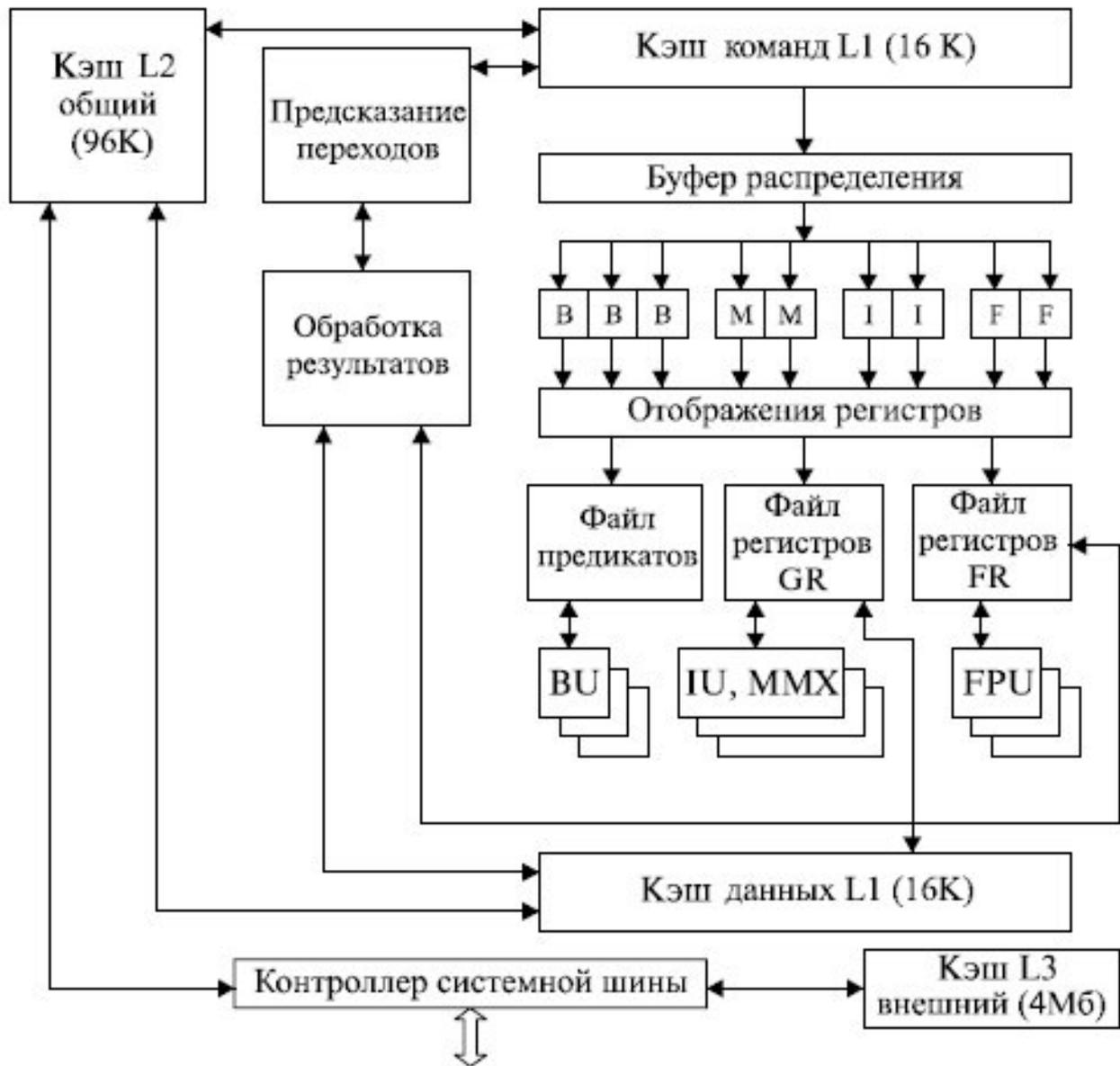


Рис. 11.2. Структура микропроцессора Itanium

Системная шина Itanium позволяет соединить до 4 микропроцессоров между собой без дополнительных мостов. Ее шина данных имеет ширину 64 разряда плюс 8 разрядов для коррекции ошибок. Тактовая частота шины равна 133 МГц, но возможны 2 передачи за такт. Это обеспечивает пропускную способность 2,1 Гбайт/с. Адресная шина содержит 44 разряда.

Основные исполнительные ресурсы Itanium включают 4 целочисленных АЛУ (IU), 4 мультимедийных исполнительных устройства (MMX), 3 устройства обработки команд перехода (BU), 2 устройства с плавающей точкой, работающих с данными расширенной точности, и 2 устройства с плавающей запятой одинарной точности (FPU). Кроме того, имеются 2 устройства загрузки регистров / записи в память.

После трансляции на исполнение команды поступают в виде 128-разрядных связок, каждая из которых содержит 3 команды и маску (рис. 11.3).

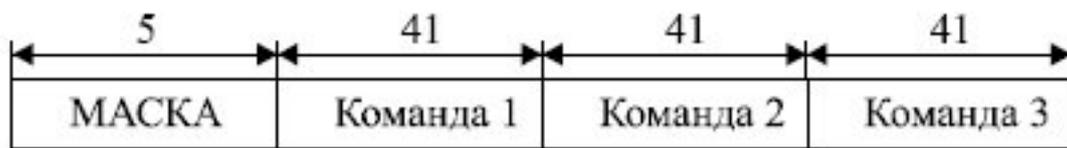


Рис. 11.3. Структура связки команд микропроцессора Itanium

В маске указывается, какой тип исполнительных устройств требуется для той или иной команды, то есть команды компонуются таким образом, чтобы они не конфликтовали между собой из-за ресурсов микропроцессора.

Каждая команда имеет следующий формат (рис. 11.4):



Рис. 11.4. Формат команды микропроцессора Itanium

В микропроцессоре имеется 9 портов четырех типов: 2 порта для команд группы М, 2 - для команд I, 2 - для команд F и 3 - для команд группы В. Они играют роль шлюзов к устройствам, выполняющим соответственно операции с памятью, целочисленные операции, операции перехода и операции с плавающей запятой. Поле маски связки команд позволяет не выдавать на одновременное выполнение больше команд, чем имеется исполнительных ресурсов микропроцессора. Команды разного типа для выполнения направляются в разные порты, и это позволяет эффективно организовать параллельное выполнение команд.

Для обеспечения более полной загрузки ресурсов Itanium в состоянии выдавать на выполнение до 6 команд за такт. При прохождении через порты осуществляется отображение виртуальных номеров регистров из программного кода в физические регистры.

Архитектура IA-64 характеризуется очень большими емкостями файлов регистров. Файл регистров общего назначения GR в IA-64 имеет емкость 128 строк. В Itanium он имеет 8 портов чтения и 6 портов записи. Эти порты позволяют поддерживать одновременно 2 М-операции и 2 I-операции за такт. Файл регистров с плавающей запятой FR имеет ту же емкость и обеспечивает одновременную работу двух М-операций и двух команд FMAC ("умножить-и-сложить"). Наконец, PRF-файл однобитных регистров-предикатов (их всего 64) имеет 15 портов чтения и 11 портов записи.

Эффективность работы функциональных устройств увеличена за счет возможности прямой подачи результата выполнения на вход другой команды, минуя запись в регистры (**продвижение данных**).

Целочисленное исполнительное ядро микропроцессора имеет 4 порта: 2 порта памяти и 2 собственно целочисленных порта. Все 4 порта могут выдавать на выполнение арифметические команды, команды сдвига, логические команды, команды сравнения и большинство целочисленных мультимедийных портов SIMD. М-порты могут также выполнять команды загрузки регистров / записи в память, а I-порты - более редкие целочисленные команды: проверку бит, поиск нулевого байта и некоторые виды сдвигов.

Устройства обработки чисел с плавающей точкой предполагают возможность работы с 82-разрядным представлением чисел. Это обеспечивает, в частности, форматы чисел с плавающей точкой одинарной, двойной и расширенной точности.

Глубина конвейера в МП Itanium составляет 10 стадий, что во многом определяет частоту работы микропроцессора. Она составляет от 800 МГц до 1 ГГц.

Двухпортовый кэш данных первого уровня является 4-канальным и имеет емкость 16 Кбайт с длиной строки 32 байта. В нем используется алгоритм сквозной записи. Благодаря наличию двух портов он может

выполнять одновременно 2 операции загрузки регистров / записи в память в любом сочетании. Загрузка операндов с плавающей точкой всегда начинается с кэша второго уровня. Это связано с малой емкостью кэша данных первого уровня. Кэш команд первого уровня емкостью 16 Кбайт также является 4-канальным.

Двухпортовый кэш второго уровня емкостью 96 Кбайт содержит и команды, и данные. Он является 6-канальным с длиной строки 64 байта.

Как и кэш данных первого уровня, он может выполнять 2 операции с памятью за такт, но в нем используется уже алгоритм обратной записи.

Внешний кэш третьего уровня имеет емкость 4 Мбайт и длину строки 64 байта. Он является 4-канальным и может работать на частоте ядра микропроцессора.

В микроархитектуре Itanium поддерживается широкий спектр возможных размеров страниц памяти: 4/8/16/64/256 Кбайт, 1/4/16/64/256 Мбайт.

Itanium имеет очень мощную систему предсказания переходов, которая использует практически все имеющиеся в этом направлении разработки.

Во-первых, это традиционные таблицы предсказания переходов, которых в Itanium имеется два типа - ВРТ (Branch Prediction Table) и МВРТ (Multiway Branch Prediction Table). Обе таблицы устроены, по сути, одинаково, являясь 4-канальной множественно-ассоциативной кэш-памятью на основе адресов связей. В них применяется двухуровневый предиктор с 4-разрядным полем истории (счетчики последних переходов). ВРТ имеет емкость 512 строк для обычных переходов, а МВРТ - всего 64 строки, но для переходов с ветвлением на несколько направлений. Строки МВРТ включают соответственно информацию о трех слотах переходов, а не об одном, как в ВРТ. Применение МВРТ целесообразно именно для ЕРISкодов с применением спекулятивной выборки и предикатов, поэтому наиболее часто употребляемыми становятся переходы со многими ветвями.

Кроме МВРТ/ВРТ для предсказания переходов используются кэш адресов перехода ТАС (Target Address Cache), содержащий по одной строке на связку, емкостью 64 строки, регистр адресов перехода ТАР (Target Address Register), содержащий по одной строке на пару связок, емкостью 4 строки, буфер адресов возврата RSB (RETurn Stack Buffer), используемый для команд перехода с возвратом, емкостью 8 строк. Для переходов относительно счетчика команд IP имеется два блока корректоров адреса перехода.

Содержимое ТАР устанавливается командой предсказания перехода bpr с суффиксом imp. Если в ТАР указан соответствующий признак попадания, переход будет выполнен безотносительно к тому, что имеется в ВРТ/МВРТ. ТАР содержит не только признак, но и адрес перехода. Предсказание ТАР может игнорироваться только предиктором циклов (то есть когда счетчик цикла указывает, что нужно выйти из цикла).

Кэш ТАС заполняется командами перехода или аппаратурой предсказания переходов Itanium. Если в МВРТ/ВРТ имеется признак попадания, ТАС предоставляет соответствующий адрес перехода.

Таким образом, от МВРТ/ВРТ зависит, должен ли происходить переход, а адрес берется из ТАС. Если же переход отсутствует в МВРТ/ВРТ, но в ТАС есть признак попадания, переход будет осуществлен.

Если данные о переходе в ВРТ/МВРТ отсутствуют, может использоваться статическое предсказание, кодируемое в команде перехода.