**Сравнение архитектуры POWER с другими RISC архитектурами.**

**Аркадьев Александр, Буторин Махмуд, гр. 339.**

**Архитектура POWER**   
**1. Эволюция архитектуры POWER в направлении архитектуры PowerPC**   
**2. PowerPC 601**   
**3. Процессор PowerPC 603**   
**Описание архитектуры и принципов работы микропроцессоров  семейства PowerPC**   
**1. Общие сведения**   
**2.  Архитектура и работа процессора.**   
**2.1    Поток команд.**   
**2.2  Очередь команд и устройство распределения.**   
**2.3  Устройство обработки переходов.**   
**2.4  Устройство завершения команд.**   
**2.5 Устройства выполнения.**   
**2.5.1 Устройства выполнения целочисленных команд (IU).**   
**2.5.2 Устройство выполнения команд с плавающей точкой (FPU)**   
**2.5.3  Устройство загрузки/записи (LSU).**   
**2.5.4 Устройство системных регистров (SRU).**   
**2.6 Устройство управления памятью (MMU)**   
**2.7   Встроенные кэши команд и данных.**   
**3.  Системный интерфейс. Схема выводов процессора.**   
**3.1 Шины адреса и данных функционируют раздельно. Используются два вида доступов к памяти и**   
**пересылки данных.**   
**3.2 Группы выводов процессора 750.**   
**4.  Регистры и программная модель PowerPC.**   
**4.1 Регистры PowerPC.**   
**4.2  Система команд PowerPC.**

**Архитектура POWER**

Архитектура POWER во многих отношениях представляет собой традиционную RISC-архитектуру. Она придерживается наиболее важных отличительных особенностей RISC: фиксированной длины команд, архитектуры регистр-регистр, простых способов адресации, простых (не требующих интерпретации) команд, большого регистрового файла и трехоперандного (неразрушительного) формата команд. Однако архитектура POWER имеет также несколько дополнительных свойств, которые отличают ее от других RISC-архитектур.

Во-первых, набор команд был основан на идее суперскалярной обработки. В базовой архитектуре команды распределяются по трем независимым исполнительным устройствам: устройству переходов, устройству с фиксированной точкой и устройству с плавающей точкой. Команды могут направляться в каждое из этих устройств одновременно, где они могут выполняться одновременно и заканчиваться не в порядке поступления. Для увеличения уровня параллелизма, который может быть достигнут на практике, архитектура набора команд определяет для каждого из устройств независимый набор регистров. Это минимизирует связи и синхронизацию, требуемые между устройствами, позволяя тем самым исполнительным устройствам настраиваться на динамическую смесь команд. Любая связь по данным, требующаяся между устройствами, должна анализироваться компилятором, который может ее эффективно спланировать. Следует отметить, что это только концептуальная модель. Любой конкретный процессор с архитектурой POWER может рассматривать любое из концептуальных устройств как множество исполнительных устройств для поддержки дополнительного параллелизма команд. Но существование модели приводит к согласованной разработке набора команд, который естественно поддерживает степень параллелизма по крайней мере равную трем.

Во-вторых, архитектура POWER расширена несколькими "смешанными" командами для сокращения времен выполнения. Возможно единственным недостатком технологии RISC по сравнению с CISC, является то, что иногда она использует большее количество команд для выполнения одного и того же задания. Было обнаружено, что во многих случаях увеличения размера кода можно избежать путем небольшого расширения набора команд, которое вовсе не означает возврат к сложным командам, подобным командам CISC. Например, значительная часть увеличения программного кода была обнаружена в кодах пролога и эпилога, связанных с сохранением и восстановлением регистров во время вызова процедуры. Чтобы устранить этот фактор IBM ввела команды "групповой загрузки и записи", которые обеспечивают пересылку нескольких регистров в/из памяти с помощью единственной команды. Соглашения о связях, используемые компиляторами POWER, рассматривают задачи планирования, разделяемые библиотеки и динамическое связывание как простой, единый механизм. Это было сделано с помощью косвенной адресации посредством таблицы содержания (TOC - Table Of Contents), которая модифицируется во время загрузки. Команды групповой загрузки и записи были важным элементом этих соглашений о связях.

Другим примером смешанных команд является возможность модификации базового регистра вновь вычисленным эффективным адресом при выполнении операций загрузки или записи (аналог автоинкрементной адресации). Эти команды устраняют необходимость выполнения дополнительных команд сложения, которые в противном случае потребовались бы для инкрементирования индекса при обращениях к массивам. Хотя это смешанная операция, она не мешает работе традиционного RISC-конвейера, поскольку модифицированный адрес уже вычислен и порт записи регистрового файла во время ожидания операции с памятью свободен.

Архитектура POWER обеспечивает также несколько других способов сокращения времени выполнения команд такие как: обширный набор команд для манипуляции битовыми полями, смешанные команды умножения-сложения с плавающей точкой, установку регистра условий в качестве побочного эффекта нормального выполнения команды и команды загрузки и записи строк (которые работают с произвольно выровненными строками байтов).

Третьим фактором, который отличает архитектуру POWER от многих других RISC-архитектур, является отсутствие механизма "задержанных переходов". Обычно этот механизм обеспечивает выполнение команды, следующей за командой условного перехода, перед выполнением самого перехода. Этот механизм эффективно работал в ранних RISC-машинах для заполнения "пузыря", появляющегося при оценке условий для выбора направления перехода и выборки нового потока команд. Однако в более продвинутых, суперскалярных машинах, этот механизм может оказаться неэффективным, поскольку один такт задержки команды перехода может привести к появлению нескольких "пузырей", которые не могут быть покрыты с помощью одного архитектурного слота задержки. Почти все такие машины, чтобы устранить влияние этих "пузырей", вынуждены вводить дополнительное оборудование (например, кэш-память адресов переходов). В таких машинах механизм задержанных переходов становится не только мало эффективным, но и привносит значительную сложность в логику обработки последовательности команд. Вместо этого архитектура переходов POWER была организована для поддержки методики "предварительного просмотра условных переходов" (branch-lockahead) и методики "свертывания переходов" (branch-folding).

Методика реализации условных переходов, используемая в архитектуре POWER, является четвертым уникальным свойством по сравнению с другими RISC-процессорами. Архитектура POWER определяет расширенные свойства регистра условий. Проблема архитектур с традиционным регистром условий заключается в том, что установка битов условий как побочного эффекта выполнения команды, ставит серьезные ограничения на возможность компилятора изменить порядок следования команд. Кроме того, регистр условий представляет собой единственный архитектурный ресурс, создающий серьезное узкое горло в машине, которая параллельно выполняет несколько команд или выполняет команды не в порядке их появления в программе. Некоторые RISC-архитектуры обходят эту проблему путем полного исключения из своего состава регистра условий и требуют установки кода условий с помощью команд сравнения в универсальный регистр, либо путем включения операции сравнения в саму команду перехода. Последний подход потенциально перегружает конвейер команд при выполнении перехода. Поэтому архитектура POWER вместо того, чтобы исправлять проблемы, связанные с традиционным подходом к регистру условий, предлагает: a) наличие специального бита в коде операции каждой команды, что делает модификацию регистра условий дополнительной возможностью, и тем самым восстанавливает способность компилятора реорганизовать код, и b) несколько (восемь) регистров условий для того, чтобы обойти проблему единственного ресурса и обеспечить большее число имен регистра условий так, что компилятор может разместить и распределить ресурсы регистра условий, как он это делает для универсальных регистров.

Другой причиной выбора модели расширенного регистра условий является то, что она согласуется с организацией машины в виде независимых исполнительных устройств. Концептуально регистр условий является локальным по отношению к устройству переходов. Следовательно, для оценки направления выполнения условного перехода не обязательно обращаться к универсальному регистровому файлу (который является локальным для устройства с фиксированной точкой). Для той степени, с которой компилятор может заранее спланировать модификацию кода условия (и/или загрузить заранее регистры адреса перехода), аппаратура может заранее просмотреть и свернуть условные переходы, выделяя их из потока команд. Это позволяет освободить в конвейере временной слот (такт) выдачи команды, обычно занятый командой перехода, и дает возможность диспетчеру команд создавать непрерывный линейный поток команд для вычислительных исполнительных устройств.

Первая реализация архитектуры POWER появилась на рынке в 1990 году. С тех пор компания IBM представила на рынок еще две версии процессоров POWER2 и POWER2+, обеспечивающих поддержку кэш-памяти второго уровня и имеющих расширенный набор команд.

По данным IBM процессор POWER требует менее одного такта для выполнении одной команды по сравнению с примерно 1.25 такта у процессора Motorola 68040, 1.45 такта у процессора SPARC, 1.8 такта у Intel i486DX и 1.8 такта Hewlett-Packard PA-RISC. Тактовая частота архитектурного ряда в зависимости от модели меняется от 25 МГц до 62 МГц.

Процессоры POWER работают на частоте 33, 41.6, 45, 50 и 62.5 МГЦ. Архитектура POWER включает раздельную кэш-память команд и данных (за исключением рабочих станций и серверов рабочих групп начального уровня, которые имеют однокристальную реализацию процессора POWER и общую кэш-память команд и данных), 64- или 128-битовую шину памяти и 52-битовый виртуальный адрес. Она также имеет интегрированный процессор плавающей точки и таким образом хорошо подходит для приложений с интенсивными вычислениями, типичными для технической среды, хотя текущая стратегия RS/6000 нацелена как на коммерческие, так и на технические приложения. RS/6000 показывает хорошую производительность на плавающей точке: 134.6 SPECp92 для POWERstation/Powerserver 580. Это меньше, чем уровень моделей Hewlett-Packard 9000 Series 800 G/H/I-50, которые достигают уровня 150 SPECfp92.

Для реализации быстрой обработки ввода/вывода в архитектуре POWER используется шина Micro Channel, имеющая пропускную способность 40 или 80 Мбайт/сек. Шина Micro Channel включает 64-битовую шину данных и обеспечивает поддержку работы нескольких главных адаптеров шины. Такая поддержка позволяет сетевым контроллерам, видеоадаптерам и другим интеллектуальным устройствам передавать информацию по шине независимо от основного процессора, что снижает нагрузку на процессор и соответственно увеличивает системную производительность.

Многокристальный набор POWER2 состоит из восьми полузаказных микросхем (устройств):

* Блок кэш-памяти команд (ICU) - 32 Кбайт, имеет два порта с 128-битовыми шинами;
* Блок устройств целочисленной арифметики (FXU) - содержит два целочисленных конвейера и два блока регистров общего назначения (по 32 32-битовых регистра). Выполняет все целочисленные и логические операции, а также все операции обращения к памяти;
* Блок устройств плавающей точки (FPU) - содержит два конвейера для выполнения операций с плавающей точкой двойной точности, а также 54 64-битовых регистра плавающей точки;
* Четыре блока кэш-памяти данных - максимальный объем кэш-памяти первого уровня составляет 256 Кбайт. Каждый блок имеет два порта. Устройство реализует также ряд функций обнаружения и коррекции ошибок при взаимодействии с системой памяти;
* Блок управления памятью (MMU).

Набор кристаллов POWER2 содержит порядка 23 миллионов транзисторов на площади 1217 квадратных мм и изготовлен по технологии КМОП с проектными нормами 0.45 микрон. Рассеиваемая мощность на частоте 66.5 МГц составляет 65 Вт.

Производительность процессора POWER2 по сравнению с POWER значительно повышена: при тактовой частоте 71.5 МГц она достигает 131 SPECint92 и 274 SPECfp92.

**1. Эволюция архитектуры POWER в направлении архитектуры PowerPC**

Компания IBM распространяет влияние архитектуры POWER в направлении малых систем с помощью платформы PowerPC. Архитектура POWER в этой форме может обеспечивать уровень производительности и масштабируемость, превышающие возможности современных персональных компьютеров. PowerPC базируется на платформе RS/6000 в дешевой конфигурации. В архитектурном плане основные отличия этих двух разработок заключаются лишь в том, что системы PowerPC используют однокристальную реализацию архитектуры POWER, изготавливаемую компанией Motorola, в то время как большинство систем RS/6000 используют многокристальную реализацию. Имеется несколько вариаций процессора PowerPC, обеспечивающих потребности портативных изделий и настольных рабочих станций, но это не исключает возможность применения этих процессоров в больших системах. Первым на рынке был объявлен процессор 601, предназначенный для использования в настольных рабочих станциях компаний IBM и Apple. За ним последовали кристаллы 603 для портативных и настольных систем начального уровня и 604 для высокопроизводительных настольных систем. Наконец, процессор 620 разработан специально для серверных конфигураций и ожидается, что со своей 64-битовой организацией он обеспечит исключительно высокий уровень производительности.

При разработке архитектуры PowerPC для удовлетворения потребностей трех различных компаний (Apple, IBM и Motorola) при сохранении совместимости с RS/6000, в архитектуре POWER было сделано несколько изменений в следующих направлениях:

* упрощение архитектуры с целью ее приспособления ее для реализации дешевых однокристальных процессоров;
* устранение команд, которые могут стать препятствием повышения тактовой частоты;
* устранение архитектурных препятствий суперскалярной обработке и внеочередному выполнению команд;
* добавление свойств, необходимых для поддержки симметричной многопроцессорной обработки;
* добавление новых свойств, считающихся необходимыми для будущих прикладных программ;
* ясное определение линии раздела между "архитектурой" и "реализацией";
* обеспечение длительного времени жизни архитектуры путем ее расширения до 64-битовой.

Архитектура PowerPC поддерживает ту же самую базовую модель программирования и назначение кодов операций команд, что и архитектура POWER. В тех местах, где были сделаны изменения, которые могли потенциально препятствовать процессорам PowerPC выполнять существующие двоичные коды RS/6000, были расставлены "ловушки", обеспечивающие прерывание и эмуляцию с помощью программного обеспечения. Такие изменения вводились, естественно, только в тех случаях, если соответствующая возможность либо использовалась не очень часто в кодах прикладных программ, либо была изолирована в библиотечных программах, которые можно просто заменить.

**2. PowerPC 601**

Первый микропроцессор PowerPC, PowerPC 601, в настоящее время выпускается как компанией IBM, так и компанией Motorola. Он представляет собой процессор среднего класса и предназначен для использования в настольных вычислительных системах малой и средней стоимости. Он был разработан в качестве переходной модели от архитектуры POWER к архитектуре PowerPC и реализует возможности обеих архитектур. При этом двоичные коды RS/6000 выполняются на нем без изменений, что дало дополнительное время разработчикам компиляторов для освоения архитектуры PowerPC, а также разработчикам прикладных систем, которые должны перекомпилировать свои программы, чтобы полностью использовать возможности архитектуры PowerPC.

Процессор 601 базировался на однокристальном процессоре IBM, который был разработан к моменту создания альянса трех ведущих фирм. Но по сравнению со своим предшественником, PowerPC 601 претерпел серьезные изменения в сторону повышения производительности и снижения стоимости. Например, в его состав было включено более сложное устройство переходов, расширенные возможностями мультипроцессорной работы, включая интерфейс шины высокопроизводительного процессора 88110 компании Motorola. В Power 601 реализована суперскалярная обработка, позволяющая выдавать на выполнение в каждом такте 3 команды, возможно не в порядке их расположения в программном коде.

**3. Процессор PowerPC 603**

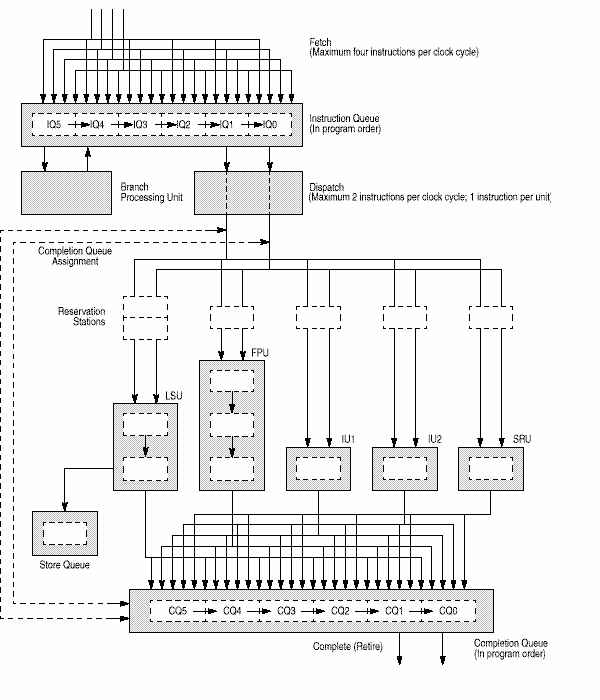
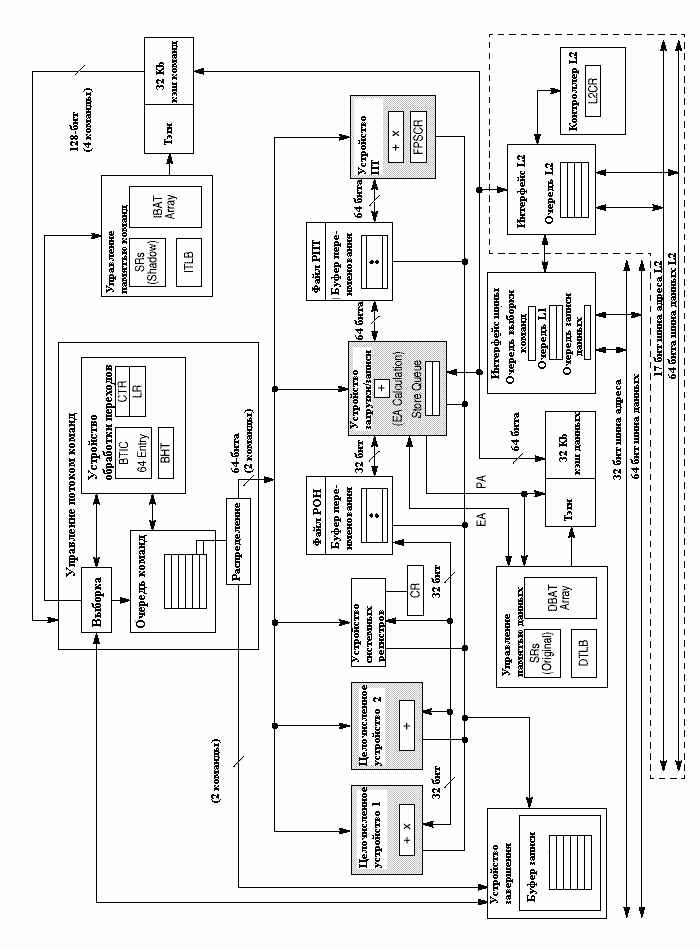
PowerPC 603 является первым микропроцессором в семействе PowerPC, который полностью поддерживает архитектуру PowerPC (рисунок 5.20). Он включает пять функциональных устройств: устройство переходов, целочисленное устройство, устройство плавающей точки, устройство загрузки/записи и устройство системных регистров, а также две, расположенных на кристалле кэш-памяти для команд и данных, емкостью по 8 Кбайт. Поскольку PowerPC 603 - суперскалярный микропроцессор, он может выдавать в эти исполнительные устройства и завершать выполнение до трех команд в каждом такте. Для увеличения производительности PowerPC 603 допускает внеочередное выполнение команд. Кроме того он обеспечивает программируемые режимы снижения потребляемой мощности, которые дают разработчикам систем гибкость реализации различных технологий управления питанием.

При обработке в процессоре команды распределяются по пяти исполнительным устройствам в заданном программой порядке. Если отсутствуют зависимости по операндам, выполнение происходит немедленно. Целочисленное устройство выполняет большинство команд за один такт. Устройство плавающей точки имеет конвейерную организацию и выполняет операции с плавающей точкой как с одинарной, так и с двойной точностью. Команды условных переходов обрабатывается в устройстве переходов. Если условия перехода доступны, то решение о направлении перехода принимается немедленно, в противном случае выполнение последующих команд продолжается по предположению (спекулятивно). Команды, модифицирующие состояние регистров управления процессором, выполняются устройством системных регистров. Наконец, пересылки данных между кэш-памятью данных, с одной стороны, и регистрами общего назначения и регистрами плавающей точки, с другой стороны, обрабатываются устройством загрузки/записи.   
В случае промаха при обращении к кэш-памяти, обращение к основной памяти осуществляется с помощью 64-битовой высокопроизводительной шины, подобной шине микропроцессора MC88110. Для максимизации пропускной способности и, как следствие, увеличения общей производительности кэш-память взаимодействует с основной памятью главным образом посредством групповых операций, которые позволяют заполнить строку кэш-памяти за одну транзакцию.

**Описание архитектуры и принципов работы микропроцессоров  семейства PowerPC**

**1. Общие сведения**

  Семейство RISC-процессоров PowerPC в настоящее время состоит из следующих моделей: EC603e, 603e, 604e, 740, 750 (производятся фирмами Motorola и IBM).   
  В данной работе архитектура и работа микропроцессоров PowerPC рассматривается на базе процессора PowerPC 750. Полную документация по архитектуре и программированию всех процессоров доступна на сайтах Motorola и IBM.   
750 реализует 32-разрядную архитектуру PowerPC,  которая предоставляет 32-разрядную адресацию, обработку целочисленных данных (8, 16, 32 разряда), данных с плавающей точкой (32 и 64 разряда).   
Процессор 750 состоит из следующих устройств выполнения :   
· Устройство с плавающей точкой (FPU)   
· Устройство обработки переходов (BPU)   
· Устройство системных регистров (SRU)   
· Устройство загрузки/записи (LSU)   
· Два целочисленных устройства (IUs): IU1 - выполняет все команды IU2 - выполняет все команды, кроме умножения и деления   
750 является суперскалярным процессором:  возможна выборка четырех команд из кэша и выполнение шести команд за один такт. Большинство целочисленных команд выполняется за один такт. Выполнение команд с плавающей точкой разбиваются на три ступени. Одна команда с ПТ занимает одну ступень, таким образом одновременно FPU может выполнять три команды с плавающей точкой (32-разрядные операнды). Сложение 64-разрядных операндов выполняется за три такта, умножение и умножение-сложение за четыре.   
750 имеет независимый встроенный восьмиканальный, 32 Кб, физически адресуемый кэш команд и данных, а также независимые устройства управления памятью команд и данных (memory management unit, MMU). Каждое MMU имеет ассоциативный буфер TLB (DTLB и ITLB) для сохранения адресов недавно использованных страниц. Архитектурой PowerPC также определяется наличие таблиц трансляции адресов блоков памяти (block address translation array, IBAT и DBAT), Подробнее работа памяти описана в соответствующем разделе.   
Кэш L2 реализован в виде встроенных памяти тэгов и внешней памяти SRAM. Доступ к внешней SRAM происходит через порт кэша L2, который поддерживает один банк памяти до 1 Мб SRAM.   
750 имеет 32-разрядную адресную шину и 64-разрядную шину данных. Внешние устройства получают системные ресурсы через устройство внешнего центрального арбитра. В 750 используется MEI (modified/exclusive/invalid) протокол для синхронизации кэша и памяти и предотвращения ошибок при обращении к кэшу.   
    
**2.  Архитектура и работа процессора.**   
  
**2.1    Поток команд.**   
Как видно из рисунка, устройство  управления потоком команд состоит из устройства последовательной выборки (fetcher), очереди из шести команд (instruction queue, IQ), устройства распределения команд и устройства обработки переходов (BPU).   
Оно определяет адрес следующей команды для выборки по информации из устройства выборки и BPU.   
Команда загружается из кэша команд в очередь команд. BPU извлекает команды перехода из последовательного загрузчика. Команды перехода, которые не могут быть обработаны немедленно, предсказываются с помощью специальных алгоритмов динамического или статического (определен архитектурой) предсказания переходов.   
Команды перехода, не влияющие на LR или CTR (регистры, содержащие адреса переходов), удаляются из потока команд.   
Команды из предсказанной ветви не завершаются, пока переход не обработан наверняка, сохраняя программную модель последовательного выполнения. Если переход был неправильным, устройство выполнения уничтожает все  предсказанные пути команд и выбирает команды из правильной ветви.



**2.2  Очередь команд и устройство распределения.**   
Очередь команд (IQ) содержит шесть команд и может быть загружена четырьмя командами за такт. Устройство выборки пытается загрузить команды на все свободные места в очереди. Все команды распределяются к соответствующим устройствам выполнения (IU1, IU2, FPU, LSU, SRU) из двух верхних позиций в очереди с максимальной скоростью две за такт. Устройство распределения проверяет зависимости регистров источника и приемника, определяет свободна ли место в очереди завершения команд, и распределяет последовательные команды по назначению.

**2.3  Устройство обработки переходов.**   
BPU получает команды перехода из устройства выборки и делает упреждающий поиск условных ветвей для их раннего предсказания, достигая попадания в большинстве случаев.   
Команды безусловного перехода или с известным условием могут быть предсказаны сразу. Для переходов с неопределенными условиями используется динамическое или статическое предсказание. Команды из предсказанной ветви выполняются, но не завершаются и не записывают результаты до подтверждения корректности перехода.   
Динамическое предсказание использует таблицу истории переходов (BHT) из 512 записей, кэш который содержит по 2 бита, определяющие 4 уровня вероятности перехода. Когда динамическое предсказание запрещено переход выбирается исходя из бита в коде команды для предсказания условных переходов.   
Когда переход сделан ( или предсказан), команды из остальных ветвей удаляются и загружаются команды из нужной ветви. BTIC - кэш на 64 элемента, содержащий команды из последних переходов. Когда команды находятся в BTIC, они считываются  на следующем такте, иначе через один такт.   
BPU содержит сумматор для вычисления адресов переходов и использует три регистра - регистр связи (LR), регистр-счетчик (CTR) и CR. BPU вычисляет точку возврата из процедуры и сохраняет результат в LR определенных команд перехода. Также в регистрах LR и CTR содержат адреса для некоторых команд обработки переходов.   Из-за использования специальных регистров обработка команд переходов независима от выполнения целочисленных команд и команд с ПТ.

**2.4  Устройство завершения команд.**   
В точке распределения команд, порядок выполнения команд поддерживается назначение команде места в очереди завершения на 6 мест. Устройство завершения отслеживает команды от распределения через устройства выполнения и возвращает результаты в порядке выполнения команд в программе из 2 нижних мест в очереди выполнения.   
Команда не может быть отправлена на выполнение, если нет места в очереди завершения. Команды перехода, не модифицирующие CTR и LR удаляются из потока команд и не занимают места в очереди завершения. Команды, модифицирующие CTR и LR занимают место в очереди, но не посылаются на выполнение.   
Завершение команды состоит в записи результатов в регистры (GPR, FPR, LR и CTR).   
Завершенные команды удаляются из очереди завершения.

**2.5 Устройства выполнения.**

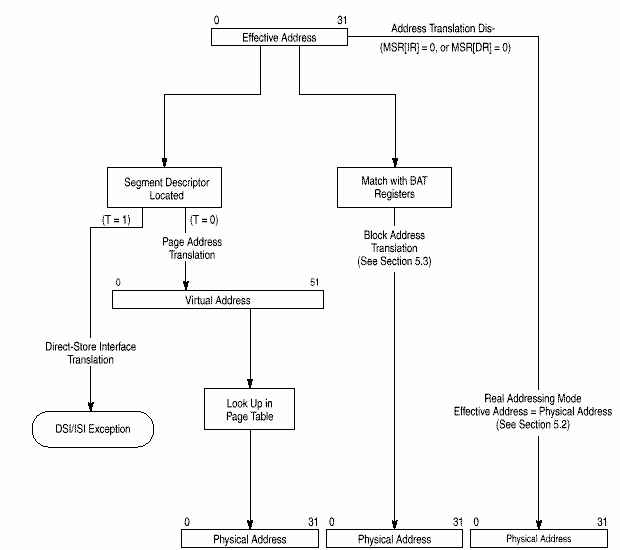
**2.5.1 Устройства выполнения целочисленных команд (IU).**   
Каждое IU состоит из трех однотактовых подустройств - быстрый сумматор/компаратор, обработки логических операций и выполнения сдвигов и циклических сдвигов. Только одно подустройство может выполнять команду в каждый момент времени.

**2.5.2 Устройство выполнения команд с плавающей точкой (FPU)**   
FPU выполняет операции одинарной точности (32 разряда) за один проход, состоящий из трех тактов. Операнды берутся из регистров FPR или буфера переименования FPR. Результаты записываются в буфер переименования регистров и доступны для последующих команд. Команды поступают в FPU в порядке распределения устройством управления командами.   
FPU содержит массив для умножения-сложения одинарной точности и контрольный регистр (FPSCR). Массив умножения-сложения позволяет 750 эффективно выполнять команды умножения и умножения-сложения. FPU является конвейерным, так что за один такт выдается одна обработанная команда. Для поддержки команд с ПТ предоставляются 32 64-разрядных регистра. Остановки, вызванные конфликтами при записи в FPR минимизируются 6 регистрами переименования с ПТ. 750 записывает содержимое регистров переименования при выходе команды из устройства завершения.   
750 поддерживает все форматы с ПТ стандарта IEEE 754 (нормализованные, ненормализованные, NaN, ноль, бесконечность).

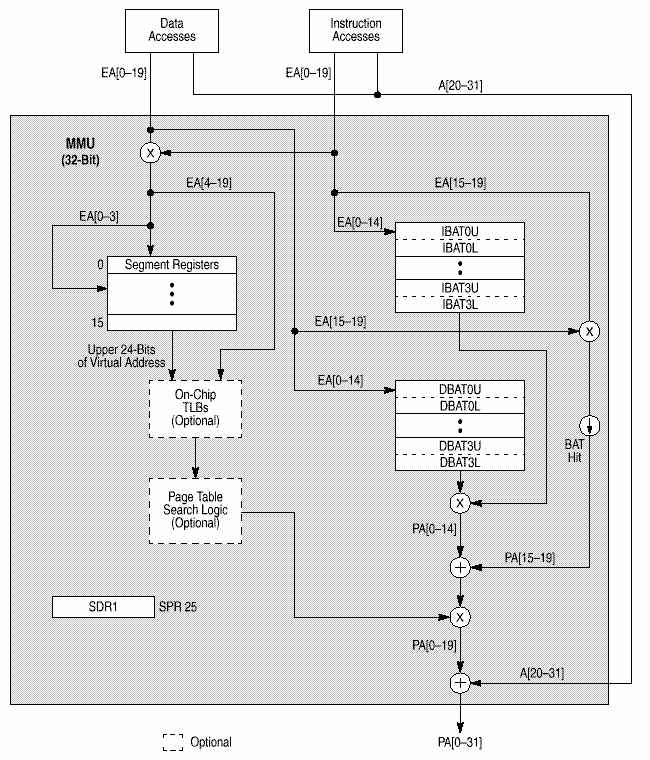
**2.5.3  Устройство загрузки/записи (LSU).**   
LSU выполняет все команды загрузки и сохранения и предоставляет интерфейс пересылки данных между GPR, FPR, и подсистем кэш/память.   
Команды загрузки и записи выполняются в порядке программы; однако некоторые обращения в память могут происходить вне очереди команд. Команды синхронизации могут быть использованы для изменения порядка команд. Максимум одна операция загрузки из кэша вне очереди может быть выполнена за такт, с двухтактовой задержкой загрузки из кэша. Данные из кэша хранятся в регистрах переименования до их записи в GPR или FPR. Сохранение не может выполняться вне очереди и операции сохранения находятся в очереди сохранения до разрешения на запись. 750 выполняет команды сохранения максимум одну за такт с общей трехтактовой задержкой записи в кэш.

**2.5.4 Устройство системных регистров (SRU).**   
SRU выполняет различные команды системного уровня, такие как логические операции с регистром условия и команды работы с регистрами специального назначения. Команды, выполняемые SRU, сохраняются в нем и обрабатываются  после выполнения  всех предыдущих команд.

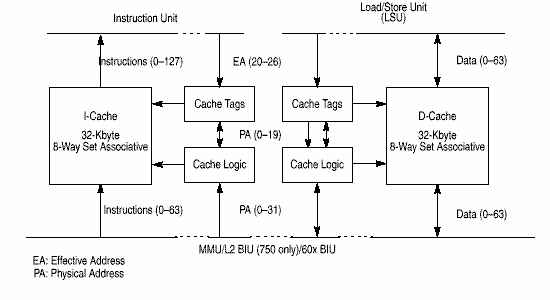
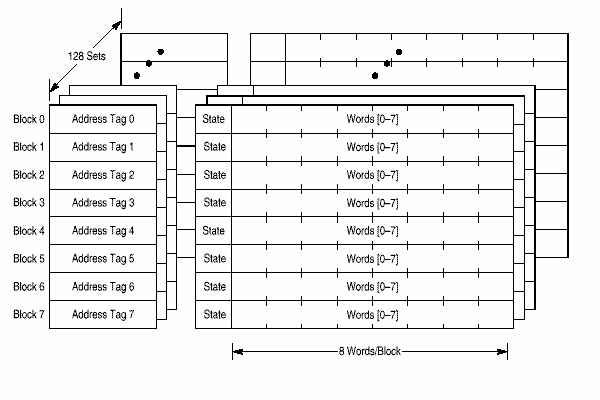
**2.6 Устройство управления памятью (MMU)**.   
MMU поддерживает до 4 Петабайт (252) виртуальной памяти и до 4 Гигабайт (232)  физической памяти для команд и данных со страницами по 4 Кб и сегментами по 256 Мб. MMU контролирует привилегии доступа , разбивая память на блоки и страницы. Вообще, механизм преобразования адресов состоит в преобразовании эффективного адреса в промежуточный виртуальный исходя из сегментной информации и затем в физический по таблицам страниц. Дескрипторы сегментов, используемые для генерации промежуточного внутреннего адреса, хранятся как встроенные 32-разрядные сегментные регистры.   
В 750 реализовано 2 буфера TLB, так что доступ к TLB для команд и данных может производится независимо.



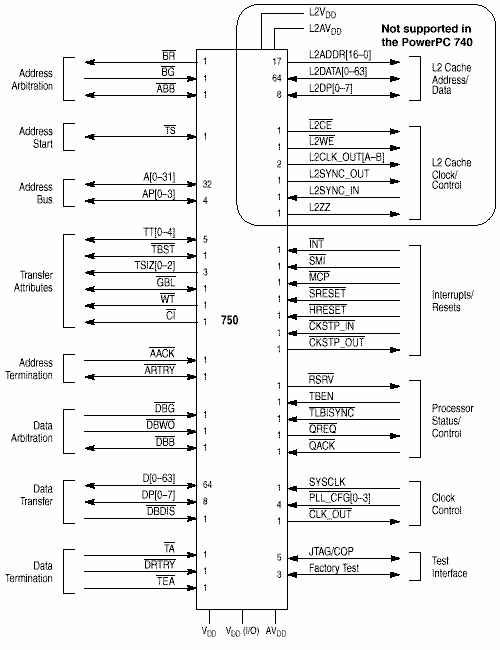
Механизм преобразования адресов блоков (block address translation, BAT) - программно-контролируемый массив доступных преобразований адресов блоков. Механизм BAT управляет преобразованием блоков до 256 Мб из 32-разрядного эффективного адресного пространства в физическое. Используются для преобразования адресов, не часто меняющих свое отображение. Элементами массива BAT являются пары BAT-регистров, доступных в режиме супервизора. В 750 есть отдельные механизмы BAT для команд и данных (4 IBAT и 4 DBAT).   
  LSU и устройство управления потоком команд вычисляют эффективные адреса данных и команд. MMU преобразует эффективные адреса в физические для доступа к памяти.   
  750 поддерживает следующие режимы преобразования адресов:   
· Реальный (real addressing) - физический адрес совпадает с эффективным.   
· Страничный (page address translation) - преобразует адреса страниц (4 Кб)   
· Блочный (block address translation) - преобразует базовые адреса блоков (от 128 Кб до 256 Мб)   
Если работает преобразование адресов, MMU преобразует старшие биты эффективного адреса в физический. Если адрес найден в массиве BAT, то физический адрес выдается сразу, иначе 32-битный эффективный адрес расширяется в 52-битный виртуальный замещением 24 старших битов на сегментный регистр, адресуемый 4 старшими битами ЕА. 52-разрядные виртуальные адреса разделены на 4 Кб страницы, отображаемые в физические.   
Младшие биты адреса одинаковы и используются для вычисления индекса в массиве тэгов кэша. После преобразования адресов MMU посылает физический адрес в кэш и данные считываются. Если кэш не используется или данных в нем нет, то не преобразованные младшие биты соединяются с преобразованными старшими в 32-разрядный физический адрес, который используется для доступа к внешней памяти.   
  
TLB используется для сохранения последних преобразований адресов при обращении к памяти. При каждом обращении, преобразование эффективного адреса по страницам или блокам производится одновременно. Если преобразование найдено и в TLB и в BAT, используется преобразование адреса блока в BAT. Обычно преобразование адреса находится в TLB и физический адрес доступен для считывания из кэша сразу. Иначе адрес ищется в таблице страниц следуя модели, определенной архитектурой PowerPC.   
TLB команд и данных преобразует адрес одновременно с доступом к внутреннему кэшу, исключая затраты времени в случае попадания в TLB.



**2.7   Встроенные кэши команд и данных.**   
    
В 750 реализованы отдельные 32 Кб кэши для команд и данных (Гарвардская архитектура). Оба кэша восьмиканальные частично ассоциативные. По архитектуре PowerPC кэши адресуются физически, реальный адрес хранится в директории кэша. Оба кэша организованы в блоки по 32 байта. Блок кэша - блок памяти, для которого описано состояние когерентности. Для каждого банка данных кэша используется алгоритм PLRU (pseudo least-recently-used) замены данных (алгоритм изложен в документации). Когерентность данных глобальной памяти и данных в кэше производится по протоколу MEI процедурой слежения  (snooping) по шине когерентности. Состояния когерентности для кэша данных: Modified (Exclusive) (M), Exclusive (Unmodified) (E), Invalid (I). Для кэша команд: Invalid (INV), Valid (VAL). Каждый кэш можно сделать недействительным (invalidate) установкой соответствующих битов в специальных регистрах (HID0).   
При непопадании в кэш, блоки заполняются за 4 прохода по 64 бита. Двойные слова одновременно записываются в кэш и в запрашивающее устройство, минимизируя задержки.   
 Оба кэша тесно связаны с устройством шины интерфейса (bus interface unit, BUI). BUI получает запросы от кэшей и выполняет эти операции в соответствии с протоколом 60х. BUI предоставляет очереди адресов, логику приоритетов и логику контроля шины.   
Кэш данных предоставляет буферы для сохранения и загрузки операций шины. Все данные из соответствующих адресных очередей помещены в кэш данных. В кэше данных также производится сохранение тэгов, требуемых для когерентности с памятью и замещение блоков кэша функциями PLRU.   
Кэш данных организован в 128 банков по 8 блоков. Каждый блок содержит 32 байта, 2 бита состояния и адресный тэг. Каждый блок кэша содержит 8 последовательных слов из памяти которые загружаются с границы в 8 слов (т.е. биты ЕА[27-31] равны 0): таким образом блок кэша никогда не пересекает границу страницы. Не выровненные обращения через границу страницы могут повлечь уменьшение производительности. Во время загрузки данных кэш не блокируется для внешних доступов до полной загрузки.   
    
В течение такта кэш данных предоставляет для считывания в LSU двойное слово. Как и кэш команд, кэш данных может быть сделан недействительным (invalidated) весь или поблочно. Данные кэша делаются недоступными и недействительными сбросом HID0[DCE] и установкой HID0[DCFI], кэш данных может быть заблокирован установкой HID0[DLOCK].  Тэги кэша имеют один порт, поэтому одновременная загрузка/сохранение и обращения для когерентности кэша вызывают конфликт, при этом LSU внутренне блокируется на один такт для записи блока данных в 8 слов в буфер обратной записи.   
Кэш команд также состоит из 128 банков по 8 блоков. Каждый блок состоит из 32 байтов, 1 бита состояния  и адресного тэга. За один цикл кэш команд предоставляет до 4 команд в очередь команд. В кэше команд поддерживаются только состояния верно/неверно (valid/invalid) для данных. Кэш команд не отслеживается, поэтому если изменяется память, данные которой содержатся в кэше, программа должна сообщать об этом устройству выборки команд. Кэш команд может быть сделан недействительным весь или поблочно. Доступ к кэшу запрещается и кэш делается недействительным сбросом HID0[ICE]  и установкой HID0[ICFI], кэш блокируется установкой HID0[ILOCK].   
В 750 реализован также кэш команд переходов (branch target instruction cache, BTIC). В BTIC хранятся встретившиеся в программе команды переходов/циклов. Если команда находится в BTIC она поступает в очередь команд на такт быстрее, чем из кэша команд.



**3.  Системный интерфейс. Схема выводов процессора.**   
    
**3.1 Шины адреса и данных функционируют раздельно. Используются два вида доступов к памяти и пересылки данных:**

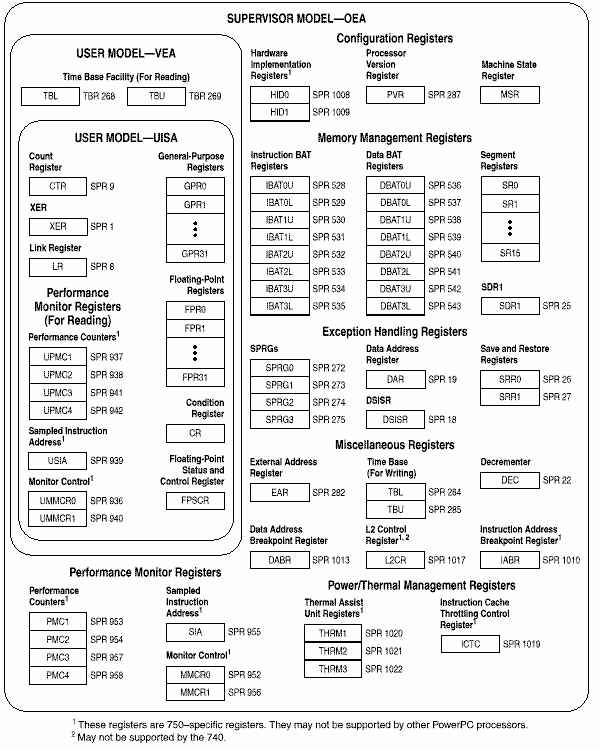


· Однотактовая (single-beat) пересылка - пересылка 8, 16, 24, 32 или 64 битов за такт шины. Используется при некэшируемых доступах в память.   
· Четырехтактовая пакетная (four-beat burst) пересылка данных - используется для чтения блоков кэша. Так как кэши первого уровня с обратной записью, то  пакетные доступы к памяти используются наиболее часто.   
Доступ к системной шине дается через механизм внутреннего арбитража.   
Обычно доступы к памяти слабо упорядочены - последовательности команд загрузки/записи не обязательно выполнять в порядке их следования - то есть можно максимизировать эффективность использования шины без потери когерентности. 750 позволяет выполнять операции загрузки/записи не в порядке расположения в очереди команд (если нет зависимостей между ними и нет случаев некэшируемых доступов).   
 

**3.2  Процессор 750 имеют следующие группы выводов:**   
· Арбитраж шины адресов   
1.  BR (Bus Request) Output - запрос операций на шине адреса   
2.  BG (Bus Grant) Input - разрешение операций на шине адреса   
3.  ABB (Address Bus Busy) Output - указывает, что 750 является мастером шины. Input - указывает на занятость шины.   
· Старт передачи адреса - показывает, что мастер шины (bus master) начал транзакцию на шине адреса.   
1. TS (Transfer Start) Output - указывает, что 750 начал транзакцию и шина адреса и атрибуты верны. Input - другой мастер шины начал транзакцию и шина адреса и ат-рибуты доступны для слежения.   
· Передачи адреса - включает в себя шину адреса и сигналы четности адреса для про-верки.   
1.  A[0-31] (Address Bus) Output - физический адрес данных для пересылки. Input -  фи-зический адрес для операции слежения (snooping) .   
2.  AP[0-3] (Address Bus Parity) Output - четность для каждых 4 байт адреса. Input - чет-ность для каждых 4 байт адреса для операции слежения .   
· Атрибуты пересылки - показывают размер пересылки, является ли пакетной, прямой записью или бескэшевой.   
1. TT[0-4] (Transfer Type) Output - текущий тип пересылки   
2. TSIZ[0-2] (Transfer Size) Output - размер текущей пересылки данных.   
3. TBST (Transfer Burst) Output - указывает на текущую пакетную пересылку.   
4. CI (Cache Inhibit) Output - указывает, что однотактовые пересылки не кэшируются.   
5. WT (Write Through) Output - указывает, что однотактовая транзакция записывается write-through. Сигнал чтении показывает выборку команды или загрузку данных.   
6. GBL (Global) Output - указывает, что транзакция глобальная. Input - указывает, что транзакция должна быть отслежена.   
· Завершения передачи адреса - подтверждают конец адресной фазы транзакции.   
1.  AACK (Address Acknowledge) Input - указывает на завершение адресной фазы тран-закции.   
2.  ARTRY (Address Retry) Output - указывает, что 750 обнаружил условия при которых фаза передачи адреса должна быть повторена.   
· Арбитраж шины данных.   
1. DBG (Data Bus Grant) Input - доступ к операциям на шине данных   
2. DBWO (Data Bus Write Only) Input - указывает на возможность записи по внеоче-редному адресу, даже если адрес чтения стоит перед адресом записи.   
3. DBB (Data Bus Busy) Output - указывает, что 750 является мастером шины. Input - другое устройство мастер шины.   
· Пересылка данных - содержит сигналы шины данных и четности.   
1.  DH[0-31], DL[0-31] (Data Bus) - шина данных раздела на две половины - верхняя шина (DH) и нижнюю шину (DL). Output - состояние данных при записи. Input - данные при чтении.   
2.  DP[0-7] (Data Bus Parity) Output - четность для каждых 8 бит при транзакции запи-си/чтения.   
3.  DBDIS (Data Bus Disable) - указывает, что 750 должен освободить шину данных и четности в течение текущего цикла.   
· Завершения пересылки данных - сигналы завершения требуются после каждого цикла пересылки данных. В одноцикловых (single-beat) транзакциях сигналы завершения указывают на окончания владения шиной, при пакетных доступах сигналы завершения  поступают после каждого цикла и отдельно сигнал окончания владения шиной.   
1. TA (Transfer Acknowledge) - указывает на удачное завершение цикла пересылки данных.   
2. DRTRY (Data Retry) - указывает на недействительность данных предыдущей пере-сылки   
3. TEA (Transfer Error Acknowledge) - ошибка на шине.   
· Адрес/данные кэша L2 - 750 имеет отдельные шины адреса и данных для доступа к кэшу L2.   
· Тактовые/контрольные сигналы кэша L2.   
· Сигналы прерываний/сброса - сигналы внешних прерываний, ошибочных остановок, программных и аппаратных сбросов.   
1.  INT (Interrupt) - 750 вызывает прерывание если MSR[EE] установлен, иначе преры-вание игнорируется.   
2.  SMI (System Management Interrupt) - прерывание системного управления, если   
 MSR [EE] установлен.   
3.  MCP (Machine Check Interrupt)   
4.  CKSTP\_IN (Checkstop Interrupt) - 750 должен прервать операцию и сбросить все выходы   
5.  CKSTP\_OUT - указывает на наступление условия остановки.   
6.  HRESET (Hard Reset)  - полное аппаратное прерывание.   
7.  SRESET (Soft Reset) - начинает процесс исключительной ситуации сброса.   
· Статус процессора и контроль - устанавливают бит когерентности станций резерва-ции, временную базу и другие функции.   
· Контроля тактовой частоты.   
· Тестовый интерфейс.

**4.  Регистры и программная модель PowerPC.**

Архитектура PowerPC включает в себя следующие уровни:   
· Уровень команд пользователя (user instruction set architecture, UISA) - определяет базовые пользовательские команды, регистры, типы данных, модели память и про-граммирования для однопроцессорной среды.   
· Архитектура виртуальной среды (virtual environment architecture,VEA) - определяет модель памяти для мультипроцессорной среды, команды контроля кэша и другие па-раметры виртуальной среды.   
· Архитектура операционной среды (OEA) - определяет модель управления памя-тью, регистры супервизора, требования синхронизации и модель исключений.   
  
**4.1 Регистры PowerPC.**   
Регистры определены на всех трех уровнях архитектуры PowerPC. Архитектура PowerPC определяет операции регистр-регистр для всех команд обработки. Источником данных является встроенные регистры или непосредственные операнды. Трехрегистровый формат команд позволяет отличать регистр результата от 2 регистров-источников, по-зволяя использовать их в других командах. Данные пересылаются между памятью и ре-гистрами только специальными командами загрузки/сохранения.   
Регистры процессора 750 показаны на рисунке.   
· Регистры пользовательского уровня (UISA) - доступны из всех программ с уровнями привилегий    пользователя или супервизора.   
  Регистры общего назначения (GPR). 32 регистра GPR (GPR0-GPR31) используются как источники или   приемники для целочисленных команд и предоставляют данные для формирования адреса.   
  Регистры с ПТ (FPR). 32 регистра FPR (FPR0-FPR31) используются как источники или приемники данных в командах с ПТ. Все регистры FPR поддерживают формат ПТ с двойной точностью.   
  Регистр условия (CR). 32-битный регистр, состоит из восьми 4-битных полей, CR0-CR7, отражает результаты  определенных арифметических операций и обес-печивают механизм тестирования и переходов.   
  Регистр контроля и статуса операций с ПТ (FPSCR). FPSCR содержит тип резуль-тата операции с ПТ, все сигнальные биты исключительных ситуаций с ПТ, биты общих исключений, биты разрешения исключений и биты контроля для совмес-тимости со стандартом IEEE 754.   
Остальные регистры уровня пользователя являются регистрами специального назна-чения (SPR). В PowerPC используется специальный механизм для явного доступа к SPR (команды mtspr и mfspr).   
  Регистр целочисленных исключений (XER). XER показывает переполнение или перенос для целочисленных операций.   
  Регистр связи (LR)  LR предоставляет адрес перехода для команды Branch condition to link register (bclr) и может быть использован для хранения логического адреса команды следующей за переходом и связи команд, обычно для связи с про-цедурами.   
  Регистр-счетчик (CTR). Содержит счетчик цикла, который может уменьшатся на 1 при выполнении определенных команд перехода.   
    
· Регистры пользовательского уровня (VEA) — PowerPC VEA определяет механизм базы времени (time base facility, TB), который состоит из двух 32-битных регистров - time base upper (TBU) и time base lower (TBL). Изменяются командами супервизора и доступны для чтения на пользовательском уровне.



· Регистры уровня супервизора (OEA) — OEA определяет регистры, используемые операционной системой для управления памятью, конфигурированием, обработкой исключений и других функций. Определены следующие 32-разрядные  регистры:

  Регистры конфигурации   
 -Регистр состояния машины (MSR). MSR определяет режимы работы   
  процессора.   
 Изменяется командами Move to machine state register (mtmsr), System Call (sc) и   
 Return from Exception (rfi).   
     -Регистр модели процессора (PVR). Регистр только для чтения, идентифицирует   
     модель и уровень архитектуры PowerPC процессора (0х0008 для 750).   
  Регистры управления памятью   
      -Регистры преобразования адресов блоков (BAT). PowerPC OEA включает в се-   
      бя массив регистров для преобразований адресов блоков, которые могут зада -   
      вать 4 блока для команд и 4 блока для данных. Регистры BAT реализованы по-   
      парно - 4 пары BAT команд (IBAT0U-IBAT3U и IBAT0L-IBAT3L) и 4 пары BAT   
      данных (DBAT0U-DBAT3U и DBAT0L-DBAT3L). Из-за раздельной загрузки   
      старших и нижних слов BAT, программа должна проверять преобразование   
      BAT во время загрузки регистров.   
     -Сегментный регистр (SR). Определены 16 32-битных сегментных регистра   
      (SR0-SR15). Поля сегментных регистров интерпретируется по разному в   
      зависимости от значения бита 0. В 750 реализованы отдельные MMU для  ко-   
      манд и данных. Регистры SR относятся к MMU данных. Для MMU команд   
      используются SR в других, так называемых «теневых» сегментных регистрах.   
  Регистры обработки исключений   
 -Регистр адреса данных (DAR). После ошибки преобразования адреса DAR   
  содержит эффективный адрес, сгенерированный ошибочной командой.   
      -SPRG0-SPRG3. Используются операционной системой.   
      -DSISR. Определяет причину ошибки преобразования адреса.   
      -Регистр сохранения/восстановления статуса машины 0 (SRR0). Содержит адрес   
      команды выполняющейся после обработки исключения.   
      -Регистр сохранения/восстановления статуса машины 1 (SRR1). Сохраняет статус   
      машины при исключении и восстанавливает после обработки исключения.   
  Разные регистры.   
           -База времени (TB). TB является 64-битной структурой для поддержки времени и   
           таймера.   
               -Регистр декремента (DEC). 32-битный счетчик с декрементом,   
               предоставляющий механизм вызова исключительной ситуации декремента после   
               программной задержки.

Остальные регистры уровня супервизора являются регистрами специального назначения, специфическими для процессора 750. С их назначением можно ознакомится в соответст-вующей документации.   
    
**4.2  Система команд PowerPC.**   
Все команды имеют длину 32 бита. Форматы всех команд можно найти в документации В основном команды имеют следующие поля:   
  0-5      биты       -     код команды   
  6-10    биты       - rD(destination) / rS(source) / crfD(результат в поле регистра CR или FPSCR) / crbD(результат как бит регистра CR или FPSCR)   
  11-15  биты    -   rA(регистр  источник или приемник) / crbA(бит регистра CR как источник) / crfS(поле регистра CR или FPSCR как источник)   
  16-20   биты      -  rB(регистр как источник) / crbB(бит регистра CR как источник)   
  16-31   биты      - SIMM(операнд со знаком) / UIMM (операнд без знака)   
Целочисленные команды оперируют с операндами размером в слово (16 бит). Команды с ПТ оперируют с операндами одинарной или двойной точности. По архитектуре PowerPC ко-манды имеют длину 4 байта. Пересылка между памятью и регистрами общего назначения производится байтами, полусловами и словами. Пересылка между памятью и регистрами с ПТ производится словами и двойными словами.   
Арифметические и логические команды не читают и не изменяют память. Содержимое па-мяти должно быть загружено в регистр и затем использовано в вычислениях. Загрузку и за-пись выполняют специальные команды.   
Программы обращаются к памяти используя 32-битный эффективный (логический) адрес, вычисляемый процессором при обращении к памяти, выполнении перехода или выборке команды. Если при вычислении эффективный адрес превышает максимальный, адрес опе-ранда рассматривается как расположенный циклически через нулевой эффективный адрес.   
Байты в памяти нумеруются последовательно, начиная с 0 и каждый номер является адресом байта. Адресом операнда в памяти является адрес его первого байта.   
Для команд перехода как эффективный адрес используется непосредственный операнд или  косвенно содержимое регистра связи или счетчика.   
 Целочисленные команды   
· Арифметические.   
· Сравнения. Выполняют сравнение rA с операндом UIMM, SIMM или регистром rB. Ре-зультат помещается в поле регистра CR, по умолчанию в CR0.   
· Логические. Выполняют побитовые логические операции.   
· Сдвига и циклического сдвига.   
 Команды обработки чисел с ПТ.   
· Арифметические. При этом операнды с одинарной точностью преобразовываются в опе-ранды с двойной точностью.   
· Сложения с суммированием.   
· Округления и преобразования 64-битных чисел с ПТ двойной точности в 32-битные оди-нарной точности.   
· Сравнения. Сравнение содержимого двух регистров FPR.   
· Команды работы с регистром FPSCR.   
· Копирования. Выполняют копирование данных из одного регистра FPR в другой.   
Команды сохранения/загрузки.   
Доступы к памяти в PowerPC делятся на выровненные и невыровненные. Адрес операнда считается не выровненным, если он не кратен его длине. Например, операнд длиной 12 байт выровнен по слову если его адрес кратен 4. Некоторые команды требуют выравнивания опе-рандов в памяти. Вообще, наилучшая производительность достигается, когда операнды в памяти выровнены.   
· Целочисленной загрузки/сохранения. Сохранение/загрузка регистра rS/rD как байта, по-луслова, слова или двойного слова по эффективному адресу.   
· Целочисленной загрузки/сохранения с реверсом байтов   
· Множественной целочисленной загрузки/сохранения. Используются для пересылки блоков данных в и из регистров общего назначения, данные могут быть выровнены (множественная загрузка) или нет (загрузка строк, выполняется гораздо дольше).   
· Загрузки/сохранения чисел с ПТ. Эффективный адрес генерируется используя косвенно регистр с индексной константой. Прямое сохранение в память не поддерживается. При загрузке числа одинарной точности преобразуются к двойной при загрузке в регистры FPR. При сохранении наоборот.   
· Синхронизации памяти   
Команды перехода и контроля потока команд   
Некоторые команды перехода выполняются условно исходя из значения битов регистра CR. Если ни одна выполняемая в данный момент команда не влияет на CR, то переход решается немедленно.   
· Относительный переход   
· Условный переход на относительный адрес   
· Переход по абсолютному адресу   
· Условный переход на абсолютный адрес   
· Переход по регистру связи   
· Переход по регистру-счетчику   
Другие команды   
· Команды системных вызовов   
· Команды контроля процессора. Чтение и запись CR, MSR и SPR.   
· Команды синхронизации завершения операций с памятью с асинхронными событиями.   
· Команды контроля памяти. Управления кэшем (пользователь и супервизор), управления сегментными регистрами (супервизор) и TLB (супервизор)