

Структура и особенности работы процессора Itanium

1. Структура процессора Itanium
2. Сериализация
 - 2.1 Сериализация команд
 - 2.2 Сериализация данных
 - 2.3 Неопределенное состояние ресурса
3. Прерывания
 - 3.1 Классификация прерываний
 - 3.2 Обработка запросов на прерывание
4. Отладка и анализ производительности
 - 4.1 Отладка
 - 4.2 Анализ производительности

- **Знать:**

- краткую характеристику структуры процессора,
- понятие сериализации команд и данных,
- типы и особенности обработки прерываний,
- возможности отладки и анализа производительности.

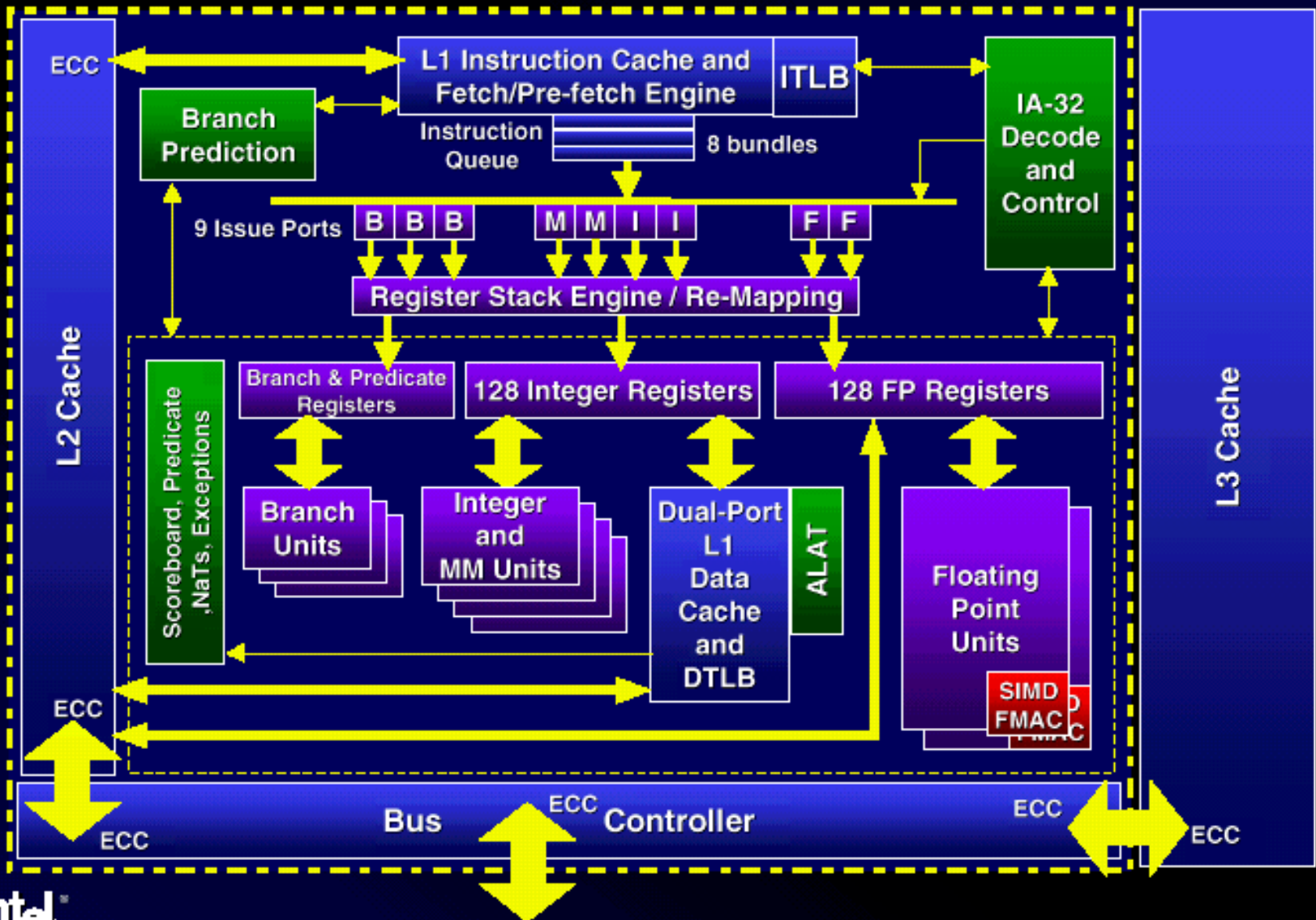
Литература:

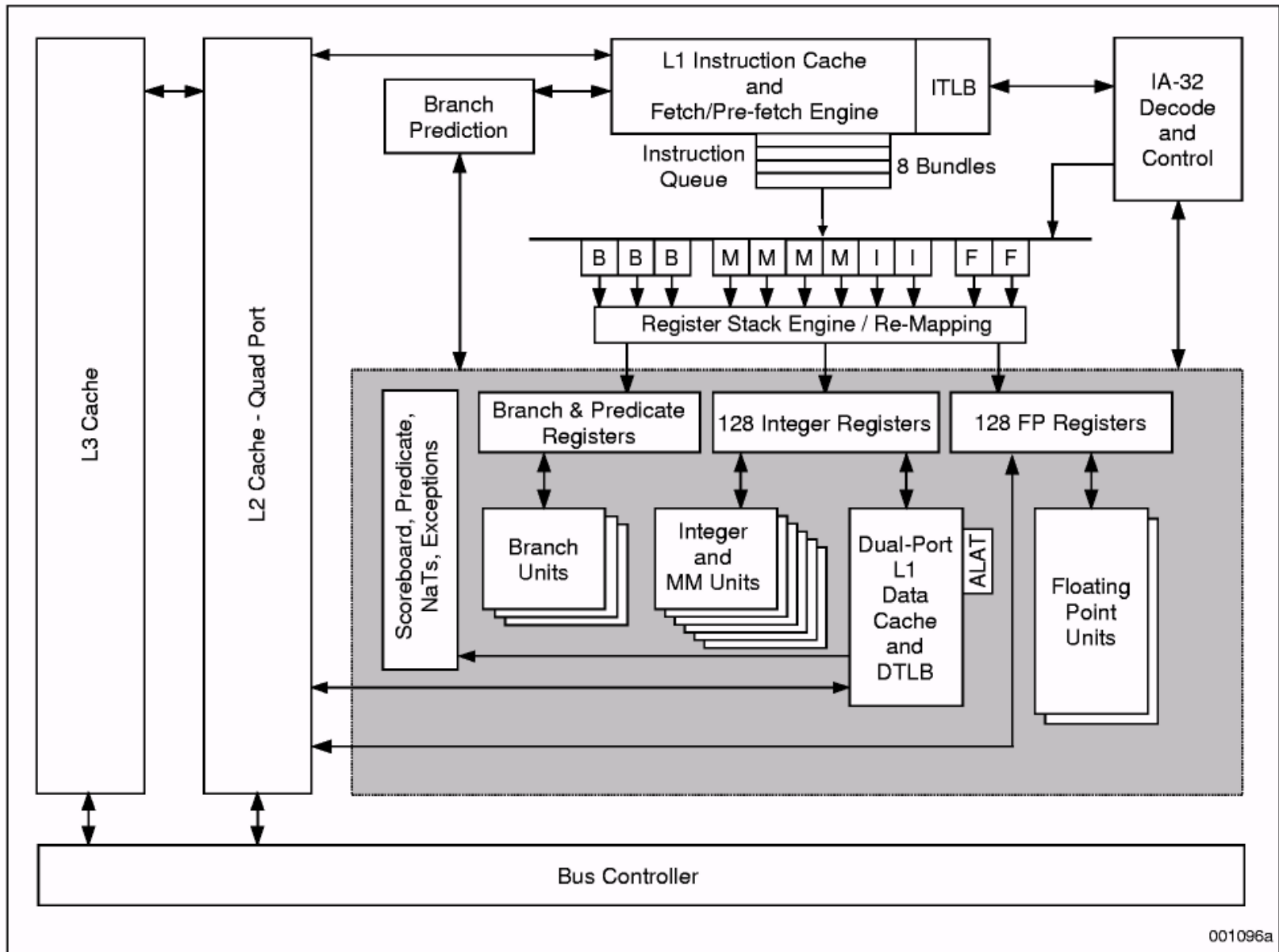
- Цилькер Б.Я., Орлов С.А. Организация ЭВМ и систем. Учебник для вузов. – СПб.: Питер, 2004. – 668 с. (с. 413-446)

1. Структура процессора Itanium

- Структура процессора Itanium приведена рисунке, где использованы следующие обозначения.
- Fetch/Pre-fetch Engine – машина выборки/предвыборки (команд).
- Issue Port – выходные порты.
- Register Stack Engin – машина регистрового стека.
- Branch Prediction – предсказание ветвлений.
- Scroboard. Predicate NaTs. Exception – Табло. Предикаты «Не факт». Исключения.

Intel® Itanium™ Processor Block Diagram





2 Сериализация

- Существуют команды, выполнение которых требует предварительного выполнения, так называемой, операции сериализации.
- Сериализация заключается во внесении всех изменений для флагов, регистров и памяти перед выполнением последующих команд программы.
- Эти побочные эффекты появляются в связи с использованием различных методов увеличения производительности, таких как изменение порядка выполнения команд, параллельное выполнение нескольких команд и т.д.
- Выделяют два вида сериализации: сериализация команд и сериализация данных.

2.1 Сериализация команд

- Сериализация команд обеспечивает необходимые изменения ресурсов процессора до того, как последующая группа команд будет выбрана.
- Программа должна использовать сериализацию команд перед любой группой команд, выполнение которых зависит от изменённых системных ресурсов. Побочные эффекты могут возникнуть в любом месте до выполнения явной операции сериализации.
- Сериализации команд требуют изменения системных ресурсов (RR, PKR, PSR (часть битов) и других), а также ряда управляющих регистров.
- Команда возврата из прерывания (rfi) и команда сериализации команд (srlz.i) выполняют явную сериализацию команд.

2.2 Сериализация данных

- Сериализация данных обеспечивает учёт всех изменений ресурсов процессора, которые влияют на выполнения команд, включая доступ к данным в памяти.
- Программа должна выполнить операцию сериализации данных до команды, выполнение которой зависит от измененных ресурсов. Побочные эффекты могут возникнуть в любом месте до выполнения явной операции сериализации.
- Сериализации данных требуют изменения системных ресурсов (RR, PKR, PSR (часть битов) и других), а также ряда управляющих регистров.
- Работа с управляющими регистрами отличаются от работы с регистрами общего назначения и других регистров. Большинство управляющих регистров требует явной сериализации данных между записью в управляющий регистр и чтением его значения.
- Для явной сериализации данных используются специальные команды. Прерывания также выполняют сериализацию данных.

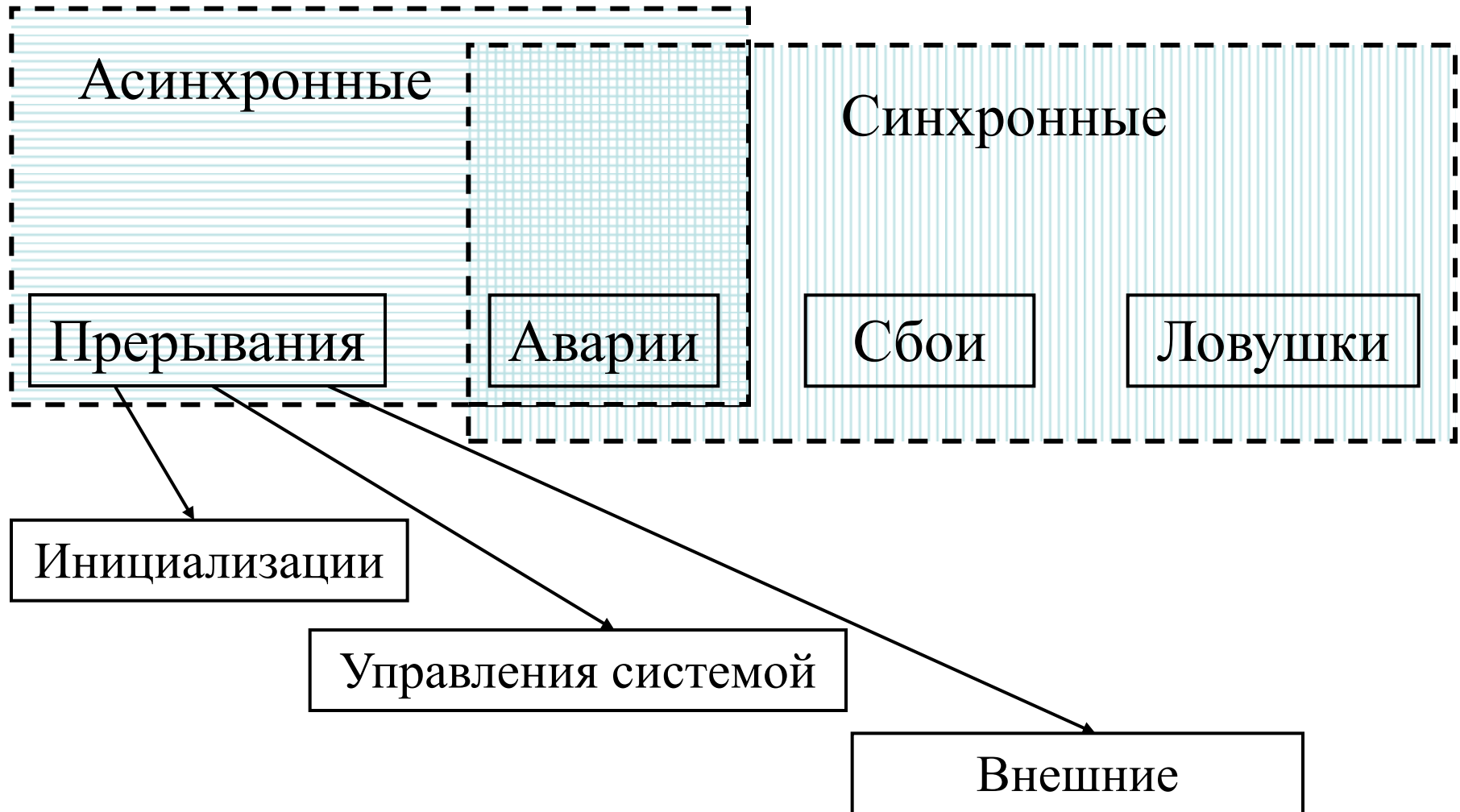
2.3 Неопределенное состояние ресурса

- Говорят, что ресурс, для которого необходима сериализация одного или нескольких новых значений, находится в неопределённом состоянии (“in-flight”) до тех пор, пока не будет выполнена операция сериализации.
- Команда, которая производит чтение ресурса находящегося в неопределённом состоянии, получит одно из промежуточных значений, либо исходное значение с момента последней сериализации.

3 Прерывания

- В зависимости от того, как обслуживаются прерывания, их можно разделить на программные прерывания, которые описываются таблицей векторов прерываний, и на аппаратные (прерывания уровня абстракции процессора (PAL)).
- Программные прерывания обслуживаются операционной системой, а PAL-прерывания обслуживаются встроенным программным обеспечением и иногда ОС (эти прерывания описываются через набор входных точек во встроенное ПО).
- Адрес таблицы прерываний (IVA – CR2) определяет расположение таблицы векторов прерываний в виртуальном или физическом (если бит PSR.it равен 0) адресном пространстве. Размер таблицы – 32 Кб и она выровнена по адресу кратному 32 Кб.

3.1 Классификация прерываний



Типы прерываний

- **Аварии** возникают, когда процессор обнаруживает внутренний сбой или при сбросе процессора (RESET). Аварии могут возникать синхронно или асинхронно относительно выполнения команд и приводят к тому, что процессор приостанавливает поток команд в непредсказуемом месте с частично обновлённым состоянием регистров и памяти. Аварии – это PAL-прерывания.
- **Прерывания** – внешнее или независимое событие (событие таймера, сообщение от устройства ввода-вывода или другого процессора). Являются асинхронными относительно потока команд. Все команды, принятые к исполнению, выполняются до конца. Подразделяются на прерывания инициализации, прерывания управления системой и внешние прерывания.

Типы прерываний (продолжение)

- **Сбои** возникают, когда текущая команда запрашивает действие, которое не может быть выполнено, либо, когда необходимо вмешательство системы до выполнения команды. Являются синхронными прерываниями. Команда, вызвавшая сбой и последующие за ней, не изменяют состояния машины.
- **Ловушки** — команды, после выполнения которых, требуется вмешательство системы. Являются синхронными прерываниями.

3.2 Обработка запросов на прерывания

- При возникновении прерывания, аппаратура компьютера как минимум должна сохранить состояние процессора для обработки события и продолжения работы.
- Состояние процессора и вектор прерывания позволяют программному обеспечению либо разрешить причину прерывания, либо передать событие для обработки на более высокий уровень ОС.
- ПО осуществляет управление структурой информационных сообщений и преобразованиями между низкоуровневыми обработчиками и высокоуровневым кодом. Такая схема позволяет ПО, вместо аппаратного обеспечения, диктовать, как наиболее лучше оптимизировать производительность для каждого прерывания в своём окружении.

Особенности обработки прерываний

- При прерывании состояние процессора сохраняется, что позволяет обработчику прерывания выполнить необходимые действия с минимальными затратами. Блоки регистров общего назначения используются как рабочие регистры для обработки прерывания.
- Для низкоуровневых обработчиков нет необходимости открывать регистровое окно, путём освобождения регистров, сохраняя их значения в памяти или в управляющих регистрах.
- Асинхронные события, такие как внешние прерывания, автоматически блокируются аппаратурой прерываний для того, чтобы ПО могло либо немедленно, либо безопасно выгрузить прерываемые ресурсы и сохранить их в памяти.

4 Отладка и анализ производительности

4.1 Отладка

- Процессоры основанные на архитектуре Itanium должны обеспечивать разнообразные возможности для отладки и анализа производительности для Itanium и IA-32 команд.
- Возможности отладки включают в себя:
 - несколько отладочных регистров точек останова для данных и команд;
 - пошаговую отладку;
 - остановку при ошибке выполнения команды;
 - ловушку при переходе;
 - ловушку перехода на более низкий уровень привилегий;
 - отладку при сбоях для команд и данных.

4.2 Анализ производительности

- Возможности анализа производительности включают в себя:
 - два набора регистров для настройки и сбора различной статистической информации о производительности;
 - отладочные регистры точек останова для команд и данных.
- Адрес команды или данных в памяти, который совпадает со значением одного из отладочных регистров точек останова для команд или данных вызывает сбой или ловушку (IA-32_Exception).
- Регистры с чётными номерами содержат адрес точки останова, а с нечётными – маску условий останова.
- Все модели процессоров поддерживают, по крайней мере, четыре регистра для команд и данных.

Анализ производительности (продолжение)

- Анализ производительности позволяет следить за событиями в процессоре при помощи программных счётчиков.
- Анализатор позволяет настраивать приложения, ОС и производительность системы. Для этого определены два набора регистров для наблюдения за производительностью.
- Регистры настройки анализатора производительности используются для управления анализаторами.
- Регистры данных анализатора производительности служат для хранения показателей производительности для Itanium и IA-32 команд.
- Любое исполнение процессора поддерживает как минимум четыре счётчика производительности и четыре регистра переполнения счётчиков производительности.
- Анализаторы производительности также контролируются регистром состояние процессора PSR, основным регистром управления (DCR) и регистром вектора анализатора производительности (PMV).

Анализ производительности (окончание)

- Запись и чтение в регистры анализатора являются привилегированными операциями, которые могут быть выполнены на нулевом уровне привилегий.
- Чтение регистров на ненулевом уровне привилегий контролируется битом PSR.sp. Когда бит PSR.sp равен единице, чтение регистров на ненулевом уровне привилегий возвращает ноль.
- Набор анализируемых событий зависит от исполнения процессора. Однако все модели процессоров должны обеспечивать как минимум два события:
 - Число выполненных команд, что включает в себя все команды, которые были выполнены без сбоев, включая «пустые» команды (nop) и команды под предикатом, которые не были выполнены.
 - Число циклов процессора как в обычном режиме (NORMAL), так и в режиме экономии электроэнергии (LOW-POWER).