

Архитектура, которая разрабатывалась совместно компаниями Intel и Hewlett Packard получила название **IA-64**. Это полностью 64-битная машина. Самым первым процессором этого типа был процессор **Itanium**.

Основной принцип организации архитектуры IA-64 сводится к тому, чтобы перенести нагрузку периода выполнения на период компиляции. Процессор в ходе выполнения переупорядочивает команды, подменяет регистры и т.д., что ведет к большой нагрузке на аппаратное обеспечение. А архитектуре IA-64 эту нагрузку выполняет компилятор. Согласно архитектуре IA-64 компилятор получает достоверную информацию о количестве регистров процессора и генерирует программу, в которой нет конфликтов между регистрами. Кроме того, компилятор следит за загрузкой функциональных блоков. Модель, в которой аппаратный параллелизм виден для компилятора, называется EPIC (Explicitly Parallel Instruction Computing – вычисления с явным параллелизмом).

В модели памяти предусмотрено 2^{64} адресов. Имеющиеся команды могут обращаться к блокам памяти по 1, 2, 4, 8, 16 и 10. Лучший способ ускорить обращение в память – выполнять эту операцию в фоновом режиме.

В архитектуре IA-64 функции планирования возложены на компилятор.

Отличительной особенностью IA-64 является идея о пучке связанных команд. Команды поступают группами по три штуки. Такая группа называется пучком. Каждый 128-битный пучок содержит три 40-битных команды фиксированного формата и 8-битный шаблон. Пучки могут быть связаны между собой, поэтому в пучке может быть более трех команд. Формат содержит информацию о том, какие команды могут выполняться параллельно. При такой системе компилятор может выделять блоки команд и сообщать процессору, что эти команды можно выполнять параллельно. Таким образом компилятор должен переупорядочивать команды, проверять, нет ли взаимосвязей и т.д. вместо аппаратного обеспечения. Основная идея – работа упорядочивания и распределения RISC-команд передается от аппаратного обеспечения к компилятору.

Причины, по которым упорядочивание команд возложены на компилятор:

- Теперь всю работу компилятор, а аппаратное обеспечение можно упростить и увеличить объем кэш-памяти.
- Для любой программы распределение происходит только один раз на этапе компиляции.

Предикация – технология, которая сокращает число условных переходов. А архитектуре IA-64 предикация происходит следующим образом. Каждая команда действительно выполняется, и в самом конце конвейера, когда уже нужно сохранять результат в выходной регистр, производится проверка, истинно ли предсказание. Если истинно – результат записывается в выходной регистр, а если ложно – запись не происходит.

Идея **спекулятивной загрузки** заключается в том, чтобы компилятор помещал команды считывания из памяти в более ранние позиции относительно других команд. Поскольку эти команды начинаются раньше, чем нужно, то они и могут завершиться до того, как потребуются результаты. Компилятор вставляет команду СHECK в том месте, где ему нужно получить значение определенного регистра. Если значение есть, то выполнение программы продолжается; в противном случае команда простаивает.