**ТЫК.ПЕРВЫЙ.СЛАЙД**

Что такое конвейерная обработка

Разработчики архитектуры компьютеров издавна прибегали к методам проектирования, известным под общим названием "совмещение операций", при котором аппаратура компьютера в любой момент времени выполняет одновременно более одной базовой операции. **ТЫК.СЛАЙД.** Этот общий метод включает два понятия: параллелизм и конвейеризацию. Хотя у них много общего и их зачастую трудно различать на практике, эти термины отражают два совершенно различных подхода. При параллелизме совмещение операций достигается путем воспроизведения в нескольких копиях аппаратной структуры. Высокая производительность достигается за счет одновременной работы всех элементов структур, осуществляющих решение различных частей задачи.

Конвейеризация (или конвейерная обработка) в общем случае основана на разделении подлежащей исполнению функции на более мелкие части, называемые ступенями, и выделении для каждой из них отдельного блока аппаратуры. Так обработку любой машинной команды можно разделить на несколько этапов (несколько ступеней), организовав передачу данных от одного этапа к следующему. При этом конвейерную обработку можно использовать для совмещения этапов выполнения разных команд. Производительность при этом возрастает благодаря тому, что одновременно на различных ступенях конвейера выполняются несколько команд. Конвейерная обработка такого рода широко применяется во всех современных быстродействующих процессорах. Поговорим больше про конвейеризацию.

**ТЫК.СЛАЙД.** Уровни конвейеризации

• Макроконвейер – отвечает за конвейеризацию на уровне процессоров

• Конвейер команд – отвечает за конвейеризацию команд процессора

• Конвейер арифметический – обеспечивает конвейеризацию на уровне выполнения команд процессора

## **ТЫК.СЛАЙД.** Простейшая организация конвейера и оценка его производительности

Для иллюстрации основных принципов построения процессоров мы будем использовать простейшую архитектуру, содержащую 32 целочисленных регистра общего назначения (R0,...,R31), 32 регистра плавающей точки (F0,...,F31) и счетчик команд PC. Будем считать, что набор команд нашего процессора включает типичные арифметические и логические операции, операции с плавающей точкой, операции пересылки данных, операции управления потоком команд и системные операции. В арифметических командах используется трехадресный формат, типичный для RISC-процессоров, а для обращения к памяти используются операции загрузки и записи содержимого регистров в память.

**ТЫК.СЛАЙД.** Выполнение типичной команды можно разделить на следующие этапы:

• выборка команды - IF (по адресу, заданному счетчиком команд, из памяти извлекается команда);

• декодирование команды / выборка операндов из регистров - ID;

• выполнение операции / вычисление эффективного адреса памяти - EX;

• обращение к памяти - MEM;

• запоминание результата - WB.

Чтобы конвейеризовать выполнение команд можно просто разбить выполнение каждой команды на указанные выше этапы, отведя для выполнения каждого этапа один такт синхронизации, и начинать в каждом такте выполнение новой команды. Для хранения промежуточных результатов каждого этапа необходимо использовать регистровую станцию (память). Промежуточные регистровые станции обеспечивают передачу данных и управляющих сигналов с одной ступени конвейера на следующую. Хотя общее время выполнения одной команды в таком конвейере будет составлять пять тактов, в каждом такте аппаратура будет выполнять в совмещенном режиме пять различных команд.

**ТЫК.СЛАЙД.** Работу конвейера можно условно представить в виде сдвинутых во времени схем процессора (см. рис.). Что-то подобное у нас было на лекциях. Этот рисунок хорошо отражает совмещение во времени выполнения различных этапов команд. Однако чаще для представления работы конвейера используются временные диаграммы **ТЫК.СЛАЙД.** (см. рис.), на которых обычно изображаются выполняемые команды, номера тактов и этапы выполнения команд.

Конвейеризация увеличивает пропускную способность процессора

(количество команд, завершающихся в единицу времени), но она не сокращает время выполнения отдельной команды. В действительности, она даже несколько увеличивает время выполнения каждой команды из-за накладных расходов, связанных с управлением регистровыми станциями. Однако увеличение пропускной способности означает, что программа будет выполняться быстрее по сравнению с простой не конвейерной схемой.

**ТЫК.СЛАЙД.** Тот факт, что время выполнения каждой команды в конвейере не уменьшается, накладывает некоторые ограничения на практическую длину конвейера. Кроме ограничений, связанных с задержкой конвейера, имеются также ограничения, возникающие в результате несбалансированности задержки на каждой его ступени и из-за накладных расходов на конвейеризацию. Частота синхронизации не может быть выше, а, следовательно, такт синхронизации не может быть меньше, чем время, необходимое для работы наиболее медленной ступени конвейера.

Накладные расходы на организацию конвейера возникают из-за:

• задержки сигналов в конвейерных регистрах (защелках) и

• из-за перекосов сигналов синхронизации.

Конвейерные регистры к длительности такта добавляют время установки и задержку распространения сигналов. В предельном случае длительность такта можно уменьшить до суммы накладных расходов и перекоса сигналов синхронизации, однако при этом в такте не останется времени для выполнения полезной работы по преобразованию информации.

**ТЫК.СЛАЙД.** В качестве примера рассмотрим не конвейерную машину с пятью этапами выполнения операций, которые имеют длительность 50, 50, 60, 50 и 50 нс соответственно (рис. 3). Пусть накладные расходы на организацию конвейерной обработки составляют 5 нс. Тогда среднее время выполнения команды в не конвейерной машине будет равно 260 нс. Если же используется конвейерная организация, длительность такта будет равна длительности самого медленного этапа обработки плюс накладные расходы, т.е. 65 нс. Это время соответствует среднему времени выполнения команды в конвейере. Таким образом, ускорение, полученное в результате конвейеризации, будет равно отношению 260/65=4:

Среднее время выполнения команды в неконвейерном режиме = 260

Среднее время выполнения команды в конвейерном режиме = 65

Ускорение от конвейеризации команды = Ср. время выполнения команды в неконвейерном режиме / Ср. время выполнения команды в конвейерном режиме

Ускорение от конвейеризации = 260 / 65 = 4

Общая формула для арифметического конвейера

Ускорения от конвейеризации операции имеет вид:

ξ = n\*k/(n+k),

где k – число ступеней конвейера, а n – длина одной последовательности исходных данных (команд) (вывод см. на примере сложении двух векторов длины n с плавающей запятой) *Конвейеризация эффективна только тогда, когда загрузка конвейера близка к полной, а скорость подачи новых команд и операндов соответствует максимальной производительности конвейера. Если произойдет задержка, то параллельно будет выполняться меньше операций и суммарная производительность снизится. Такие задержки могут возникать в результате возникновения конфликтных ситуаций.*

**ТЫК.СЛАЙД.** При реализации конвейерной обработки возникают ситуации, которые препятствуют выполнению очередной команды из потока команд в предназначенном для нее такте. Такие ситуации называются конфликтами.

Конфликты снижают реальную производительность конвейера, которая могла бы быть достигнута в идеальном случае.

Рассмотрим различные типы конфликтов, возникающие при выполнении команд в конвейере, и способы их разрешения.

**ТЫК.СЛАЙД.** Существуют три класса конфликтов:

1. Структурные конфликты, которые возникают из-за конфликтов по ресурсам, когда аппаратные средства не могут поддерживать все возможные комбинации команд в режиме одновременного выполнения с совмещением.

2. Конфликты по данным, возникающие в случае, когда выполнение одной команды зависит от результата выполнения предыдущей команды.

3. Конфликты по управлению, которые возникают при конвейеризации команд переходов и других команд, которые изменяют значение счетчика команд.

Конфликты в конвейере приводят к необходимости приостановки выполнения команд (pipeline stall). Обычно в простейших конвейерах, если приостанавливается какая-либо команда, то все следующие за ней команды также приостанавливаются. Команды, предшествующие приостановленной, могут продолжать выполняться, но во время приостановки не выбирается ни одна новая команда.

# ТЫК.СЛАЙД. Структурные конфликты и способы их минимизации

Совмещенный режим выполнения команд в общем случае требует конвейеризации функциональных устройств и дублирования ресурсов для разрешения всех возможных комбинаций команд в конвейере. Если какая-нибудь

комбинация команд не может быть принята из-за конфликта по ресурсам, то говорят, что в машине имеется структурный конфликт.

**ТЫК.СЛАЙД.** Примеры архитектур, в которых возможно появление структурных конфликтов:

*• Машины с не полностью конвейерными функциональными устройствами.*

Время работы такого устройства может составлять несколько тактов синхронизации конвейера. В этом случае последовательные команды, которые используют данное функциональное устройство, не могут поступать в него в каждом такте.

*• Машины с недостаточным дублированием некоторых ресурсов, что препятствует выполнению произвольной последовательности команд в конвейере без его приостановки. Например, машина может иметь только один порт записи в регистровый файл, но при определенных обстоятельствах конвейеру может потребоваться выполнить две записи в регистровый файл в одном такте. Это также приведет к структурному конфликту. Когда последовательность команд наталкивается на такой конфликт, конвейер приостанавливает выполнение одной из команд до тех пор, пока не станет доступным требуемое устройство.*

*• Машины, в которых имеется единственный конвейер памяти для команд и данных (рис. 5). В этом случае, когда одна команда содержит обращение к памяти за данными, оно будет конфликтовать с выборкой более поздней команды из памяти. Чтобы разрешить эту ситуацию, можно просто приостановить конвейер на один такт, когда происходит обращение к памяти за данными. Подобная приостановка часто называются "конвейерным пузырем" (pipeline bubble) или просто пузырем, поскольку пузырь проходит по конвейеру, занимая место, но не выполняя никакой полезной работы.*

При всех прочих обстоятельствах, машина без структурных конфликтов будет всегда иметь более низкий CPI (среднее число тактов на выдачу

команды).

Возникает вопрос: почему разработчики допускают наличие структурных конфликтов? Для этого имеются две причины: снижение стоимости и уменьшение задержки устройства. Конвейеризация всех функциональных устройств может оказаться слишком дорогой. Машины, допускающие два обращения к памяти в одном такте, должны иметь удвоенную пропускную способность памяти, например, путем организации раздельных кэшей для команд и данных. Аналогично, полностью конвейерное устройство деления с плавающей точкой требует огромного количества вентилей. Если структурные конфликты не будут возникать слишком часто, то может быть и не стоит платить за то, чтобы их обойти. Как правило, можно разработать не конвейерное, или не полностью конвейерное устройство, имеющее меньшую общую задержку, чем полностью конвейерное. Например, разработчики устройств с плавающей точкой компьютеров CDC7600 и MIPS R2010 предпочли иметь меньшую задержку выполнения операций вместо полной их конвейеризации. **ТЫК.ПОСЛЕДНИЙ.СЛАЙД.** На этом все.