

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ ЛЕКЦИЯ №5 «Архитектура микропроцессора» (продолжение)

Лектор:
доцент каф. ЭАФУ ФТИ
Горюнов А.Г.

Томск 2014 г.

План лекции

- 5.1 Система команд микропроцессора;
- 5.2 Общие сведения о МП Intel и AMD (обзор микроархитектур современных десктопных процессоров).

5.1 Система команд микропроцессора

Решение задач на ЭВМ реализуется программным способом, т.е. путем выполнения последовательно во времени отдельных операций над информацией, предусмотренных алгоритмом решения задачи.

Алгоритм – это точно определенная последовательность действий, которые необходимо выполнить над исходной информацией, чтобы получить решение задачи.

Алгоритм решения задачи, заданный в виде последовательности команд на языке вычислительной машины (в кодах машины), называется **машинной программой**.

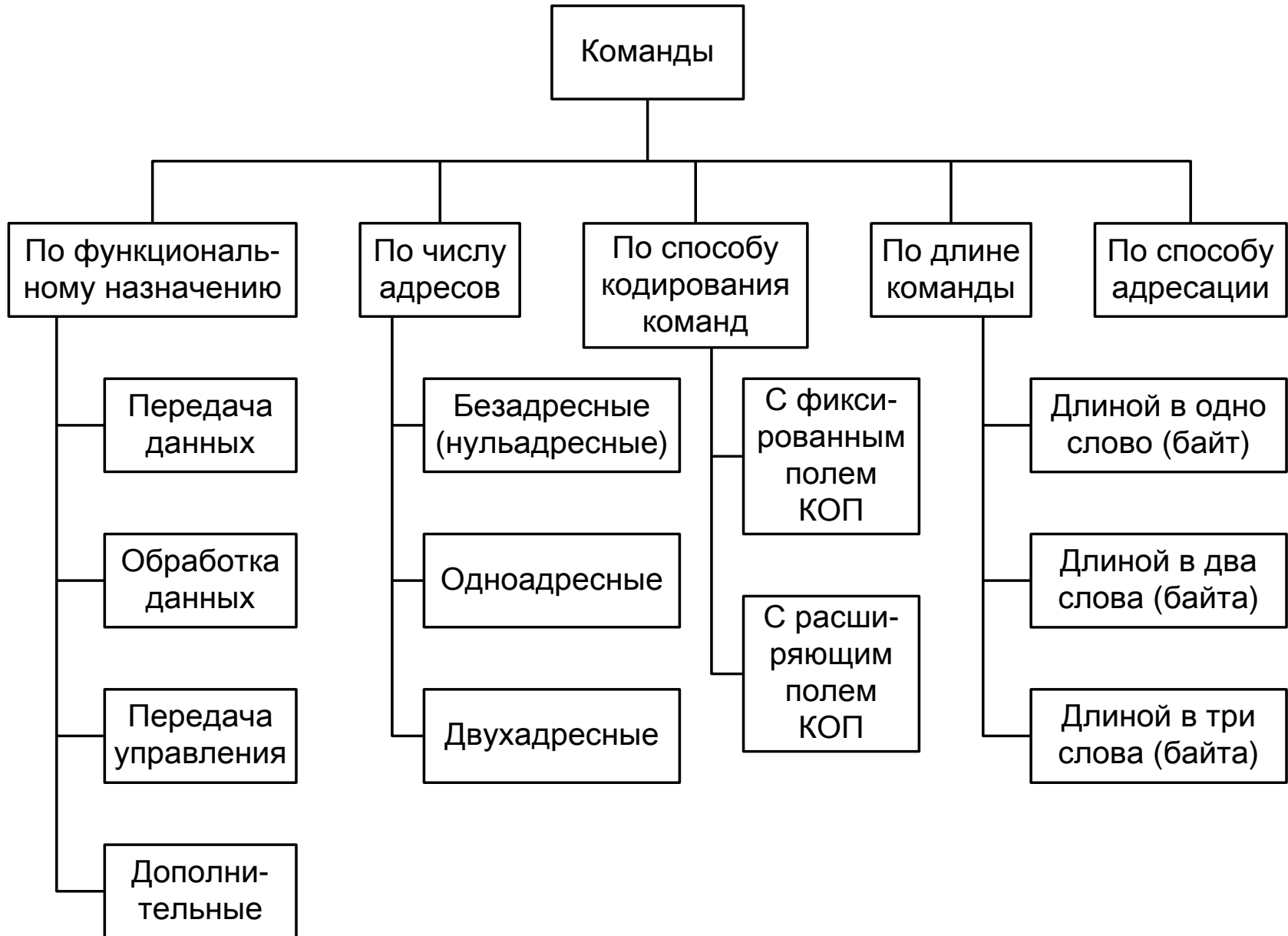
Оптимальную систему команд определяют как совокупность команд, которая удовлетворяет требованиям проблемно-ориентированных применений таким образом, что избыточность аппаратуры и аппаратно-программных средств на реализацию редко используемых команд оказывается минимальной.

Важной характеристикой команды является ее формат, определяющий структурные элементы команды, каждый из которых интерпретируется определенным образом при ее выполнении.

Среди таких элементов (полей) команды выделяют следующие:

- **код операции**, определяющий выполняемое действие;
- **адрес** ячейки памяти, регистра процессора, внешнего устройства;
- режим адресации;
- операнд при использовании непосредственной адресации;
- код анализируемых признаков для команд условного перехода.

Классификация команд:



Важнейшим структурным элементом формата любой команды является **код операции** (КОП), определяющей действие, которое должно быть выполнено.

Большое число КОП в процессоре очень важно, так как аппаратная реализация команд экономит память и время.

Но при выборе МП необходимо концентрировать внимание на полноте операций с конкретными типами данных, а не только на числе команд, на доступных режимах адресации.

Число бит, отводимое под КОП, является функцией полного набора реализуемых команд.

Например:

Таблица 5.1

КОП							Операнд	
С	С	С	С	С	Р	Р	Р	...

Команды бывают однооперандными и двухоперандными. **Однооперандные** команды – команды обрабатывающие только один операнд, а **двухоперандные** – два операнда.

При наличии двух операндов командой обычно изменяется только один из них.

Так как информация берется только из одной ячейки, эту ячейку называются **источником**; ячейка, содержимое которой изменяется, называется **приемником**.

Команда машинной программы (иначе, машинная команда) – это элементарная инструкция машине, выполняемая ею автоматически без каких-либо дополнительных указаний и пояснений.

Машинная команда состоит из двух частей: операционной и адресной.

Операционная часть команды – это группа разрядов в команде, предназначенная для представления кода операции машины.

Адресная часть команды – это группа разрядов в команде, в которой записываются коды адреса (адресов) ячеек памяти машины, предназначенных для оперативного хранения информации или иных объектов, задействованных при выполнении команды. Часто эти адреса называются адресами операндов, т. е. чисел, участвующих в операции.

По количеству адресов, записываемых в команде, команды делятся на безадресные, одно-, двух- и трехадресные.

Типовая структура трехадресной команды:

КОП a1, a2, a3,

где a1 и a2 – адреса ячеек (регистров), где расположены, соответственно, первое и второе числа, участвующие в операции;

a3 – адрес ячейки (регистра), куда следует поместить число, полученное в результате выполнения операции.

Типовая структура двухадресной команды:

КОП a1, a2,

где a1 – это обычно адрес ячейки (регистра), где хранится первое из чисел, участвующих в операции, и куда после завершения операции должен быть записан результат операции;

a2 – обычно адрес ячейки (регистра), где хранится второе участвующее в операции число.

КОП а1,

где а1 в зависимости от модификации команды может обозначать либо адрес ячейки (регистра), где хранится одно из чисел, участвующих в операции, либо адрес ячейки (регистра), куда следует поместить число результат операции.

Безадресная команда содержит только код операции, а информация для нее должна быть заранее помещена в определенные регистры машины (безадресные команды могут использоваться только совместно с командами другой адресности).

Пример. Поступила представленная на языке символического кодирования команда:

СЛ 0103, 5102.

Такую команду следует расшифровать так: «сложить число, записанное в ячейке 0103 памяти, с числом, записанным в ячейке 5102, а затем результат (т. е. сумму) поместить в ячейку 0103».

5.2 Общие сведения о МП Intel и AMD (Обзор микроархитектур современных десктопных процессоров)

За последние годы сформировалось несколько направлений, или семейств массовых высокопроизводительных микропроцессоров, конкурирующих друг с другом на рынке применений для десктопных компьютеров.

Каждое семейство имеет свою микроархитектуру.

Микроархитектура – это внутренняя организация процессорного ядра, разработанная на основе представлений её создателей о потребностях пользователей, критериях оценки производительности и прочих потребительских качествах, перспективах развития и рыночных тенденциях.

Эти представления влияют на принятие разработчиками принципиальных решений о тех или иных ключевых архитектурных особенностях процессора и о различных компромиссах, заложенных в архитектуру.

Рассмотрим следующие основные семейства процессоров: Intel Pentium III, Intel Pentium 4, Intel Core, AMD Athlon 64 / Opteron и IBM PowerPC 970. Дадим краткие характеристики каждому из перечисленных семейств, а также конкретным процессорам, его представляющим:

- **Intel Pentium III (P-III)**

Наиболее типичный представитель 6-го поколения процессорной архитектуры (P6) компании Intel. Процессоры, являющиеся развитием P-III – Pentium M (P-M) и Core Duo (P-M2). В связи с наличием серьёзных архитектурных усовершенствований процессоры P-M и P-M2 выделяют в отдельную подгруппу (P6+).

- **Intel Pentium 4 (P-4)**

Представитель 7-го поколения процессоров (P7), эту архитектуру также называют NetBurst. Развитием семейства является процессор Prescott (P-4E), имеющий определенные микроархитектурные отличия и поддерживающий 64-битный режим целочисленной и адресной арифметики EM64T (x86-64).

- **Intel Core (P8)**

Процессор новой микроархитектуры 8-го поколения P8, 2006 г. В продажу поступил под названиями Core 2 Duo и Core 2 Extreme. Известен также под кодовым именем Conroe. Несмотря на то, что этот процессор базируется в основном на архитектурных принципах семейства P6, он имеет много принципиальных отличий количественного и качественного плана.

- **AMD Athlon 64 / Opteron (K8)**

Представитель высокопроизводительной микроархитектуры компании AMD. Базируется на архитектурных принципах предыдущего семейства K7, отличается от него определёнными усовершенствованиями, поддержкой 64-битного режима AMD64 (x86-64) и наличием встроенного контроллера памяти.

- **IBM PowerPC 970 (PPC970)**

Процессор RISC-архитектуры IBM Power, известен также под названием G5. До недавнего времени был основным процессором персональных компьютеров компании Apple (в настоящее время Apple постепенно переводит свои ПК на процессоры компании Intel). Основан на микроархитектуре серверного процессора Power4 (самостоятельно).

Общая организация современного микропроцессора

Большинство современных микропроцессоров относятся к классу

- конвейерных;
- суперскалярных процессоров;
- с внеочередным исполнением операций.

Рассмотрим кратко каждую из этих трех ключевых характеристик.

Конвейерная организация процессора.

Выполнение каждой команды складывается из ряда последовательных этапов (шагов, стадий), суть которых не меняется от команды к команде. С целью увеличения быстродействия процессора и максимального использования всех его возможностей в современных микропроцессорах используется **конвейерный принцип обработки информации.**

Конвейерный принцип подразумевает, что в каждый момент времени процессор работает над различными стадиями выполнения нескольких команд, причем на выполнение каждой стадии выделяются отдельные аппаратные ресурсы.

По очередному тактовому импульсу каждая команда в конвейере продвигается на следующую стадию обработки, выполненная команда покидает конвейер, а новая поступает в него.

В различных процессорах количество и суть этапов различаются.

Рассмотрим принципы конвейерной обработки информации на примере пятиступенчатого конвейера, в котором выполнение команды складывается из следующих этапов:

- IF (Instruction Fetch) – считывание команды в процессор;
- ID (Instruction Decoding) – декодирование команды;
- OR (Operand Reading) – считывание операндов;
- EX (Executing) – выполнение команды;
- WB (Write Back) – запись результата.

Таблица 5.2

Команда	Такт								
	1	2	3	4	5	6	7	8	9
i	IF	ID	OR	EX	WB				
i+1		IF	ID	OR	EX	WB			
i+2			IF	ID	OR	EX	WB		
i+3				IF	ID	OR	EX	WB	
i+4					IF	ID	OR	EX	WB

Анализ таблицы показывает, что при конвейерной обработке после того, как получен результат выполнения первой команды, результат очередной команды появляется в следующем такте работы процессора.

Очевидно, что при достаточно длительной работе конвейера его быстродействие будет существенно превышать быстродействие, достигаемое при последовательной обработке команд.

Это увеличение будет тем больше, чем меньше длительность такта конвейера и чем больше количество выполненных команд.

Сокращение длительности такта достигается, в частности, разбиением выполнения команды на большое число этапов, каждый из которых включает в себя относительно простые операции и поэтому может выполняться за короткий промежуток времени.

Конфликты в конвейере и способы минимизации их влияния на производительность процессора.

Значительное преимущество конвейерной обработки перед последовательной имеет место в идеальном конвейере, в котором отсутствуют конфликты и все команды выполняются друг за другом без перезагрузки конвейера.

Наличие конфликтов снижает реальную производительность конвейера по сравнению с идеальным случаем.

Конфликты – это такие ситуации в конвейерной обработке, которые препятствуют выполнению очередной команды в предназначенном для нее такте.

Конфликты делятся на три группы:

- структурные;
- по управлению;
- по данным.

Структурные конфликты. Структурные конфликты возникают в том случае, когда аппаратные средства процессора не могут поддерживать все возможные комбинации команд в режиме одновременного выполнения с совмещением.

Конфликты по управлению. Конфликты по управлению возникают при конвейеризации команд переходов и других команд, изменяющих значение счетчика команд.

Конфликты по данным. Конфликты по данным возникают в случаях, когда выполнение одной команды зависит от результата выполнения предыдущей команды.

Как отмечалось выше, наличие конфликтов приводит к значительному снижению производительности микропроцессора. Определенные типы конфликтов требуют приостановки конвейера. При этом останавливается выполнение всех команд, находящихся на различных стадиях обработки (до 20-ти команд в Pentium-4).

Другие конфликты, например, при неверном предсказанном направлении перехода, ведут к необходимости полной перезагрузки конвейера. Потери будут тем больше, чем более длинный конвейер используется в микропроцессоре. Такая ситуация явилась одной из причин сокращения числа ступеней в микропроцессорах последних моделей.

Суперскалярность – архитектура вычислительного ядра, использующая несколько декодеров команд, которые могут нагружать работой множество исполнительных блоков. Планирование исполнения потока команд является динамическим и осуществляется самим вычислительным ядром.

Суперскалярная организация означает, что на каждом этапе обрабатываются сразу несколько потоков инструкций (операций) в параллель – от выборки из кэша инструкций до полного завершения (отставки).

Суперскалярность наряду с тактовой частотой является важнейшим показателем пропускной способности процессора.

Уровень суперскалярности («ширина обработки», гарантированно обеспеченная на всех этапах) в современных производительных процессорах варьируется от 3 (P-III, P-4, K8) до 4-5 (P8, PPC970).

Внеочередное исполнение операций.

Внеочередное исполнение операций означает, что операции не обязаны выполняться в функциональных устройствах строго в том порядке, который определён в программном коде.

Более поздние (по коду) операции могут исполняться перед более ранними, если не зависят от порождаемых ими результатов.

Процессор должен лишь гарантировать, чтобы результаты «внеочередного» выполнения программы совпадали с результатами «правильного» последовательного выполнения.

Механизм внеочередного исполнения позволяет в значительной степени сгладить эффект от ожидания считывания данных из кэшей верхних уровней и из оперативной памяти, что может занимать десятки и сотни тактов.

Также он позволяет оптимизировать выполнение смежных операций, особенно при наличии сложных зависимостей между ними в условиях высокой задержки исполнения в устройствах и недостаточного количества регистров.