1. АК как набор взаимодействующих компонент. Функции компонент

Архитектура компьютера, характеризующая его логическую организацию, может быть представлена как множество взаимосвязанных компонент, включающих, на первый взгляд, элементы различной природы –ПО, аппаратное обеспечение, алгоритмическое обеспечение, специальное фирменное обеспечение – и поддерживающих его слаженное функционирование в форме единого архитектурного ансамбля, позволяющего вести эффективную обработку различных объектов. Архитектура вычислительной системы – это совокупность основных функциональных возможностей системы, сфер применения (научно-техническая, экономическая, управление и т.д.), режимов работы (пакетный, мультипрограммный, разделения времени, диалоговый и т.д.), характеристик ВС (быстродействие, набор и объем памяти, набор периферийных устройств и т.д.), особенностей структуры (одно-, многопроцессорная) и реализующей базы, т.е. совокупность software, hardware, brainware и firmware. Архитектура ВС – вычислительная и логическая возможности, аппаратные средства, ПО.

1. Конвейеризация: Проблемы и решения

В большинстве ранних компьютеров команды выполнялись последовательно, то есть сначала происходило чтение команды, затем дешифрация, выбор операндов и собственно исполнение операции. Последовательный способ работы процессора определяет простоту его логической организации. При этом для всей памяти системы достаточно иметь один узел управления циклом памяти. Так как процессор во время поиска команды свободен, он может быть использован для индексных операций над адресами. Такое совмещение позволяет эффективнее использовать процессор. Стремление повысить быстродействие ВС привело к появлению процессоров с совмещением операций, то есть к распараллеливанию процессов, несмотря на удорожание оборудования. Идея об использовании принципа совмещения операций в целях повышения скорости работы компьютеров была высказана академиком C.А. Лебедевым на сессии АН СССР в 1957 году. Хотя эти идеи были высказаны около 50 лет назад, они актуальны и сегодня. Одна из возможностей совмещения операций состоит в модульной организации памяти, при которой каждый модуль имеет собственный адресный и числовой регистры, то есть модули способны работать одновременно. Количество модулей обычно равно небольшой степени числа 2. В начальных моделях IBM 7030 имелись две группы модулей памяти с чередующимися в каждой группе адресами. Одна группа главным образом используется для хранения команд, а другая – для хранения операндов. Применяя указанные свойства в сочетании с соответствующим программированием задачи, можно достичь более высокой скорости поступления данных из памяти, чем скорость, которая обеспечивается физическим циклом памяти. Более сложные способы реализации совмещения обращений к памяти состоят в построении многовходовой памяти, что допускает одновременное обращение к ней со стороны многих устройств, разрешая при этом конфликт одновременного обращения. Дальнейшим развитием идеи совмещения операций и совершенствования методов ускорения обработки данных в компьютерах стала идея конвейерной (pipeline) обработки, позволяющей совмещать во времени выполнение различных шагов операций для некоторого множества команд. Принцип конвейерной обработки основан на разбиении вычислительного процесса на несколько подпроцессов, каждый из которых выполняется в отдельном устройстве, как это имеет место при выполнении сложных операций последовательно по подоперациям на промышленном конвейере, в связи с чем этот метод и называют конвейерным.

3. Иерархия памяти: Все типы и их функции

Память компьютера может быть организована как многоуровневая с различным объемом и временем доступа к ней сверхоперативная (СОЗУ), оперативная (ОП), внешняя (ВнП), так и одноуровневая, виртуальная. Почти всегда виртуальная память есть переупорядоченное подмножество реальной памяти. Уровни иерархии памяти взаимосвязаны между собой -- все данные одного уровня могут присутствовать на более низком уровне. Основная память предназначена для хранения команд и данных и обеспечивает адресный доступ к ним от процессора. Современные устройства памяти работают со скоростью, близкой к скорости работы процессора. При обращении процессора к памяти сначала производится поиск необходимых данных в СОЗУ. Так как время доступа к этому типу памяти меньше, чем к ОП, и, как правило, большинство необходимых для работы данных уже содержится там, то среднее время доступа к памяти уменьшается. Постоянная память (ПЗУ, ROM - read only memory) предназначена только для чтения ее содержимого. В ней хранятся программы проверки оборудования компьютера, инициирования загрузки операционной системы и выполнения базовых функций по обслуживанию устройств, а также настройки конфигурации компьютера (SETUP). Так как основное содержимое ПЗУ связано с обслуживанием ввода-вывода, часто программы ROM-памяти называют BIOS (Basic Input-Output System). Небольшая по объему полупостоянная память используется для хранения параметров конфигурации компьютера. По технологии изготовления ее часто называют CMOS-памятью (complementary metal-oxide semiconductor), она имеет низкое энергопотребление.

Регистровая память – простейшая регистровая структура используется для организации стековой памяти.

Кэши - следующий уровень иерархии памяти. Кэши сравнительно небольшие по объему, но имеют высокую скорость доступа. Использование кэш-памяти основывается на двух принципах: локализация в пространстве и локализация во времени. В кэш-памяти хранятся команды и данные, которые часто используются и требуют малых временных затрат для доступа к ним.

Кэш-память позволяет организовать работу медленной оперативной памяти как быстродействующей, оптимизируя следующие аспекты: максимизация коэффициента попадания; уменьшение времени доступа; уменьшение штрафа промаха; уменьшение непроизводительных затрат времени, требуемых для поддержания консистентности кэша.

Внешняя память – это память, реализованная в виде внешних запоминающих устройств (ВЗУ) с разными принципами хранения информации. Внешняя память относится к внешним устройствам компьютера и используется для долговременного хранения любой информации, которая может потребоваться для решения задач.

1. Система в коде вычетов. Операции и примеры

Поиск новых систем кодирования данных преследует цель постоянно оптимизировать ресурсы, необходимые для обработки данных: память, процессорное время, средства надежности, точность вычислений и т.д. Не явилась исключением и система кодирования данных с применением вычетов, призванная ускорить арифметические операции, повысить точность и надежность вычислений. Пусть P1, P2, ..., Pn – целые числа; Pi > 1, (Pi, Pj ) = 1, i ≠ j; M = (I = 1 to n)∏ Pi ; xi – наименьшие неотрицательные решения системы сравнений A ≡ xi (mod Pi ), i = 1, 2, ..., n; A ⊂ [0, M). Тогда кортеж (x1, x2, ..., xn) будем называть кодом числа A в системе в коде вычетов (СКВ) при заданных основаниях P1, P2, ..., Pn. Записывают это обычно так: A ∼ (x1, x2, ..., xn). Компоненты xi называют разрядами числа A в СКВ. Идейными корнями данная система восходит к так называемой китайской теореме об остатках. Существуют алгоритмы арифметических операций на основе СКВ, алгоритмы перевода из позиционной системы счисления в СКВ и т.д.: каждый разряд числа несет информацию обо всем числе, откуда следует независимость разрядов друг от друга; отсутствие переносов между разрядами при выполнении арифметических операций упрощает их выполнение; малоразрядность компонент числа превращает арифметические операции в однотактные, иногда просто в выборку результата из таблицы. Система в коде вычетов в основном применяется в специализированных вычислительных устройствах из-за высокой сложности операций определения знака числа и сравнения чисел.