

---

М. К. БУЗА

---

# Архитектура КОМПЬЮТЕРОВ

---



М. К. БУЗА

---

# Архитектура КОМПЬЮТЕРОВ

---

*Утверждено  
Министерством образования  
Республики Беларусь  
в качестве учебника для студентов  
учреждений высшего образования  
по специальностям «Информатика»,  
«Прикладная информатика»,  
«Прикладная математика»*



Минск  
«Вышэйшая школа»  
2015

УДК 004.2(075.8)  
ББК 32.973-02я73  
Б90

Рецензенты: кафедра интеллектуальных информационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (заведующий кафедрой доктор технических наук, профессор *В.В. Голенков*); главный научный сотрудник Объединенного института проблем информатики Национальной академии наук Беларуси доктор технических наук, профессор *С.Ф. Липницкий*

*Все права на данное издание защищены. Воспроизведение всей книги или любой ее части не может быть осуществлено без разрешения издательства.*

**Буза, М. К.**

Б90 Архитектура компьютеров : учебник / М. К. Буза. — Минск : Вышэйшая школа, 2015. — 414 с. : ил.  
ISBN 978-985-06-2652-3.

Учебник подготовлен в полном соответствии с учебной программой дисциплины «Архитектура компьютеров» для учреждений высшего образования. Содержит сведения о последних достижениях в области архитектур вычислительных систем (компьютеров), в том числе многоядерных, графических и квантовых процессоров.

Актуальность книги обусловлена отсутствием соответствующей отечественной учебной литературы в данной предметной области.

Для студентов учреждений высшего образования, специализирующихся в области системного программирования, разработки эффективных программных проектов, особенно операционных систем. Может быть использован разработчиками новых архитектурных решений. Отдельные главы будут полезны специалистам, работающим в смежных областях информатики, главная цель которых — проектирование эффективных приложений в собственной предметной области.

**УДК 004.2(075.8)**  
**ББК 32.973-02я73**

ISBN 978-985-06-2652-3

© Буза М.К., 2015  
© Оформление. УП «Издательство  
“Вышэйшая школа”», 2015

## ПРЕДИСЛОВИЕ

Компьютерная наука находится сегодня в стадии интенсивного и многопланового развития, в связи с чем нет строгого и обоснованного разделения этой области знания на самостоятельные учебные дисциплины. Она прошла путь от создания некоторого языка программирования до цельной системы знаний, воплощенных в серию учебных дисциплин, преподаваемых в университетах. Среди них проектирование и анализ алгоритмов, базы данных, архитектура компьютеров, операционные системы, компьютерные сети, компиляторы и т.д. В последние годы реализованы новые архитектурные решения компьютеров, существенным изменениям подверглась логическая организация действующих вычислительных систем, расширилась сфера применения нейровычислителей. Значимые успехи достигнуты в области проектирования многоядерных, графических и квантовых процессоров.

Дисциплина «Архитектура компьютеров» изучает внутреннюю организацию вычислительной системы, знание которой позволяет программистам любого уровня рационально использовать все ресурсы системы и проектировать эффективные программы.

Значительные достижения в развитии технологий проектирования средств вычислительной техники, программного обеспечения и его надежности, инструментариев и методов инжиниринга, аттестации и верификации программных проектов, а также новый спектр приложений вычислительной техники и программного обеспечения привели к необходимости пересмотреть существующие архитектурные решения компьютеров. Вместо монопольной концепции последовательного исполнения операций появились идеи совместной, параллельной и распределенной обработки данных. На смену однопроцессорным компьютерам, базирующимся на принципах Дж. фон Неймана, пришли многопроцессорные, конвейерные и параллельные архитектуры.

Наряду с традиционными компьютерами на CISC-процессорах многочисленными фирмами реализованы концепции RISC-процессоров, воплотивших сокращенный набор регистровых команд.

В области суперкомпьютеров в настоящее время достигнута пиковая производительность в 100 Пбайт/с. Среди последних разработок такие вычислительные гиганты, как суперкомпьютер «Titan-Cray XK7» (США) с производительностью 17,59 Пфлопс и пиковой производительностью 20,13 Пфлопс, а также лидер среди компьютеров — суперкомпьютер «Tianhe-2» (Китай) с производительностью 33,86 Пфлопс на тесте Linpack и пиковой производительностью 54,90 Пфлопс.

Существенное развитие получили компьютеры с VLIW-архитектурой, позволившей за счет упаковки в одну связку нескольких команд, масштабируемости, предикации, загрузки по предположению, тегов и дескрипторов ускорить процесс обработки.

Широкое распространение нашли векторно-конвейерные компьютеры, массово-параллельные компьютеры с распределенной памятью, компьютеры с кластерной архитектурой, позволяющей достигать практически неограниченной производительности.

Все эти решения требуют осмысления, чему в немалой степени способствует предлагаемый учебник. Одна из его целей — пройти путь от компьютеров фон Неймана до создания новых архитектурных ансамблей.

В книге уделяется внимание как теоретическим аспектам, так и реальным архитектурам, воплощенным в действующих вычислительных системах и способствующим развитию существующих и генерации новых идей в этой области.

Базируясь на понятии процесса, рассмотрены ключевые теоретические решения, многие из которых можно обнаружить в большинстве современных вычислительных систем. Для освоения учебника необходимы знания в области структур компьютеров, проектирования программ и владение начальными сведениями по операционным системам.

Каждая из 12 глав учебника представляет отдельный интерес, что дает возможность некоторые главы включать в смежные курсы («Высокопроизводительные вычисления», «Операционные системы», «Компьютерные сети», «Системы параллельного действия», «Проектирование процессорной обработки» и т.д.). Это позволяет работать с учебником, не обязательно придерживаясь порядка, предложенного автором.

Учебник подготовлен в полном соответствии с учебной программой дисциплины «Архитектура компьютеров» для учреждений высшего образования Республики Беларусь.

Автор выражает глубокую благодарность доктору технических наук, профессору В.В. Голенкову (Белорусский государственный университет информатики и радиоэлектроники) и доктору технических наук, профессору С.Ф. Липницкому (Объединенный институт проблем информатики Национальной академии наук Беларуси) за полезные замечания и рекомендации, способствующие улучшению содержания книги.

*Автор*



## ВВЕДЕНИЕ

Сегодня прогресс невозможен без компьютеризации всех сфер деятельности. От скорости и полноты обработки и передачи информации зависят не только достижения в научной, образовательной, экономической деятельности, но и успехи в сфере политики и защиты государственных интересов, в совокупности способствующие устойчивому развитию всех стран. Важная роль в этом процессе принадлежит суперкомпьютерам, производительность которых возростала в последнее время на порядок за каждое пятилетие. Приведем основные параметры некоторых машин этого класса.

Векторно-конвейерный компьютер «Cray T932» фирмы «Cray Research Inc.» (в настоящее время это подразделение «Silicon Graphics») выпускается с 1996 г. Максимальная производительность одного процессора — около 2 млрд операций в секунду, оперативная память наращивается до 8 Гбайт, дисковое пространство — до 256 000 Гбайт (т.е. 256 Тбайт). Компьютер в максимальной конфигурации содержит 32 идентичных процессора, работающих над общей памятью, поэтому максимальная производительность всей вычислительной системы составляет более 60 млрд операций в секунду.

Массово-параллельный компьютер IBM SP2 фирмы IBM (иногда такие компьютеры называют компьютерами с массовым параллелизмом) строится на основе стандартных микропроцессоров «PowerPC 604e» или «Power2 SC», соединенных между собой через высокоскоростной коммутатор, причем каждый имеет свою локальную оперативную память и дисковую подсистему. В частности, максимальная система, установленная в Тихоокеанской северо-западной лаборатории (Pacific Northwest National Laboratory) в США, содержит 512 процессоров. Исходя из числа процессоров, можно представить суммарную мощность всей вычислительной системы.

Среди суперкомпьютеров с массовым параллелизмом можно выделить «Intel Red» (пиковая производительность 4 Тфлопс) и «IBM White» (12 Тфлопс).

Компьютер с кластерной архитектурой «HP Exemplar» от «Hewlett-Packard», например, модели V2250 (класс V) построен на основе микропроцессора PA-8200, работающего с тактовой частотой 240 МГц. До 16 процессоров можно объединить в рамках одного узла с общей оперативной памятью до 16 Гбайт. В свою очередь узлы в рамках одной вычислительной системы соединяются между собой через высокоскоростные каналы передачи данных.

Компания «Linux Networks» создала высокопроизводительный кластерный Linux-суперкомпьютер с 1323 процессорами для Министерства обороны США.

Суперкомпьютер «Titan-Cray XK7», разработанный в США, имеет производительность 17,59 Пфлопс и пиковую предельную производительность 20,13 Пфлопс. Компьютер содержит более 18 тыс. процессоров «Opteron 6200», а общее количество ядер составляет 560 640.

Лидер среди компьютеров — суперкомпьютер «Tianhe-2», созданный в Китае, с производительностью 33,86 Пфлопс на тесте Linpack и пиковой (теоретической) производительностью 54,90 Пфлопс. Он построен на основе 12-ядерных процессоров «Intel Xeon Phi ES-2692», работающих с частотой 2,2 ГГц. Общее количество ядер 3 120 000. Объем памяти 1024 Тбайт. Работает под операционной системой Kylin Linux.

Возникает естественный вопрос: при решении каких задач необходим суперкомпьютер?

Чтобы оценить сложность решаемых на практике задач, обратимся к такой области, как, например, оптимизация процесса добычи нефти. Представим подземный нефтяной резервуар с неким количеством пробуренных скважин: по одним на по-

верхность откачивается нефть, по другим обратно закачивается вода. Нужно смоделировать ситуацию в данном резервуаре, чтобы оценить запасы нефти или необходимость в дополнительных скважинах.

Для определения необходимых арифметических операций достаточно принять упрощенную схему, при которой моделируемая область отображается в куб. Разумные размеры куба для получения правдоподобных результатов –  $100 \cdot 100 \cdot 100$  точек. В каждой точке куба надо вычислить от 5 до 20 функций: три компонента скорости, давление, температуру, концентрацию компонент (вода, газ и нефть – это минимальный набор компонент, в более реалистичных моделях рассматривают, например, различные фракции нефти). Далее значения функций находят как решение нелинейных уравнений, что требует от 200 до 1000 арифметических операций. И наконец, если исследуется нестационарный процесс, т.е. нужно понять, как эта система ведет себя во времени, то производится от 100 до 1000 шагов по времени. В итоге получаем:

$$10^6 \text{ (точек сетки)} \cdot 10 \text{ (функций)} \cdot 500 \text{ (операций)} \cdot 500 \text{ (шагов по времени)} = 2,5 \cdot 10^{12},$$

т.е. 2500 млрд арифметических операций для выполнения одного лишь расчета. А изменение параметров модели? А отслеживание текущей ситуации при изменении входных данных? Подобные расчеты необходимо делать много раз, что предъявляет очень жесткие требования к производительности используемых вычислительных систем.

Примеры использования суперкомпьютеров можно найти в автомобилестроении, фармакологии, задачах прогноза погоды и моделирования изменения климата, сейсморазведки и т.д.

Первый компьютер EDSAC (1949 г.) с тактом 2 мкс мог выполнить  $2n$  арифметических операций за  $18n$  мс, т.е. в среднем 100 арифметических операций в секунду, а суперкомпьютер «Cray C90» имеет время такта около 4 нс с пиковой производительностью около 1 млрд арифметических операций в секунду.

Производительность компьютеров за этот период выросла приблизительно в 10 млн раз. Уменьшение времени такта является прямым способом увеличения производительности, однако эта составляющая (с 2 мкс до 4 нс) в общем объеме обеспечивает ускорение лишь в 500 раз. Остальной рост производительности дает использование новых архитектурных решений, среди которых значительное место занимает воплощение принципа параллельной, распределенной и конвейерной обработки данных.

Идея конвейерной обработки состоит в выделении отдельных этапов выполнения общей операции, причем результат работы каждого этапа передается следующему и одновременно принимается новая порция входных данных. Таким образом, за счет совмещения операций, раньше разнесенных во времени, увеличивается скорость обработки данных.

Обратимся к истории основных нововведений в архитектуре процессоров.

- IBM 701 (1953 г.), IBM 704 (1955 г.): *разрядно-параллельная память, разрядно-параллельная арифметика*. Первые компьютеры (EDSAC, EDVAC, UNIVAC) имели разрядно-последовательную память, из которой слова считывались последовательно бит за битом. Первым компьютером, использующим разрядно-параллельную память и разрядно-параллельную арифметику, стал IBM 701, а наибольшую популярность получила модель IBM 704, в которой впервые были применены память на ферритовых сердечниках и аппаратное арифметическое устройство (АУ) с плавающей точкой.

- IBM 709 (1958 г.): *независимые процессоры ввода-вывода*. Процессоры первых компьютеров сами управляли вводом-выводом. Однако скорость работы самого бы-

строго внешнего устройства — магнитной ленты была в 1000 раз меньше быстродействия процессора, поэтому во время операций ввода-вывода процессор фактически простаивал. В 1958 г. к компьютеру IBM 704 присоединили 6 независимых процессоров ввода-вывода, которые могли работать параллельно с основным процессором, а сам компьютер переименовали в IBM 709.

- «IBM Stretch» (1961 г.): *опережающий просмотр, расслоение памяти*. Компьютер имеет две принципиально важные особенности: опережающий просмотр вперед для выборки команд и расслоение памяти на два банка для согласования низкой скорости выборки из памяти и скорости выполнения операций.

- «Atlas» (1963 г.): *конвейер команд*. Впервые конвейерный принцип выполнения команд был использован в машине «Atlas», разработанной в Манчестерском университете. Выполнение команд разбито на 4 стадии: выборка команды, вычисление адреса операнда, выборка операнда и выполнение операции. Конвейеризация позволила уменьшить время выполнения команд с 6 до 1,6 мкс. Данный компьютер оказал огромное влияние как на архитектуру ЭВМ, так и на программное обеспечение. В нем впервые применена мультипрограммная операционная система (ОС), основанная на использовании виртуальной памяти и системы прерываний.

- CDC 6600 (1964 г.): *независимые функциональные устройства (ФУ)*. Фирма «Control Data Corporation» (CDC) при непосредственном участии одного из ее основателей, Сеймура Р. Крэя (Seymour R. Cray), выпустила первый компьютер, в котором использовалось несколько независимых функциональных устройств. Приведем некоторые параметры компьютера: время такта 100 нс, производительность 2–3 млн операций в секунду, оперативная память разбита на 32 банка 60-разрядных слов по 4096 слов в каждом, цикл памяти 1 мкс, 10 независимых функциональных устройств.

- CDC 7600 (1969 г.): *конвейерные независимые функциональные устройства*. CDC выпускает компьютер CDC 7600 с восемью независимыми конвейерными функциональными устройствами — сочетание параллельной и конвейерной обработки. Основные параметры: время такта 27,5 нс, производительность 10–15 млн операций в секунду, 8 конвейерных функциональных устройств, двухуровневая память.

- ILLIAC IV (1974 г.): *матричные процессоры*. Проект: 256 процессорных элементов (ПЭ), включающих 4 квадранта по 64 ПЭ, возможность реконфигурации (2 квадранта по 128 ПЭ или 1 квадрант из 256 ПЭ), время такта 40 нс, производительность 1 Гфлопс. Реализована матрица из 64 ПЭ, все элементы которой работали в синхронном режиме, выполняя в каждый момент времени одну и ту же команду, поступившую от устройства управления (УУ), но над своими данными; ПЭ имел собственное арифметико-логическое устройство (АЛУ) с полным набором команд. Оперативная память (ОП), или оперативное запоминающее устройство (ОЗУ), — 2 К слов по 64 разряда, цикл памяти 350 нс, каждый ПЭ имел непосредственный доступ только к своей ОП. Сеть пересылки данных — двумерный тор со сдвигом на единицу на границе по горизонтали.

Стоимость проекта в 4 раза выше планируемой. Сделан лишь 1 квадрант, такт 80 нс, реальная производительность до 50 Мфлопс. Данный проект оказал огромное влияние на архитектуру последующих машин, построенных по схожему принципу, в частности, на архитектуру PEPE, BSP, ICL, DAP.

- «Cray-1» (1976 г.): *векторно-конвейерные процессоры*. Компания «Cray Research» в 1976 г. выпускает первый векторно-конвейерный компьютер «Cray-1»: время такта 12,5 нс, 12 конвейерных функциональных устройств, пиковая производительность 160 млн операций в секунду, оперативная память до 1 М слов по 64 разряда, цикл памяти 50 нс.



Главным новшеством проекта является введение векторных команд, работающих с целыми массивами независимых данных и позволяющих эффективно использовать конвейерные функциональные устройства.

Иерархия памяти прямого отношения к параллелизму не имеет, однако, безусловно, относится к тем особенностям архитектуры компьютеров, которые имеют значение для повышения их производительности (сглаживание разницы между скоростью работы процессора и временем выборки из памяти). Основные уровни: регистры, кэш-память, оперативная память, дисковая память. Время выборки по уровням от дисковой памяти к регистрам уменьшается, стоимость в пересчете на 1 слово (байт) растет. В настоящее время подобная иерархия поддерживается даже на персональных компьютерах.

На современном этапе высокопроизводительная техника развивается по следующим основным направлениям.

1. *Векторно-конвейерные компьютеры.* Особенностью таких машин являются конвейерные функциональные устройства и набор векторных инструкций в системе команд. Векторные команды оперируют целыми массивами независимых данных, что позволяет эффективно загружать доступные конвейеры.

2. *Массово-параллельные компьютеры с распределенной памятью.* Идея построения компьютеров этого класса проста: серийные микропроцессоры снабжаются локальной памятью и соединяются посредством некоторой коммуникационной среды. Они обладают свойством масштабируемости. Недостатком компьютеров этого класса является то, что межпроцессорное взаимодействие идет намного медленнее, чем локальная обработка данных самими процессорами. К этому же классу можно отнести и сети компьютеров, которые все чаще рассматривают как дешевую альтернативу очень дорогим суперкомпьютерам.

3. *Параллельные компьютеры с общей памятью.* Вся оперативная память таких компьютеров разделяется несколькими одинаковыми процессорами. Это снимает проблемы предыдущего класса, но добавляет новые: число процессоров, имеющих доступ к общей памяти, по чисто техническим причинам нельзя сделать большим.

4. *Компьютеры с кластерной архитектурой.* Это направление, строго говоря, представляет собой комбинации трех предыдущих. Из нескольких процессоров (традиционных или векторно-конвейерных) и общей для них памяти формируется вычислительный узел. Если полученной вычислительной мощности недостаточно, то несколько узлов объединяются высокоскоростными каналами. В настоящее время такие компьютеры признаны наиболее перспективными. В странах СНГ в этом направлении работают Объединенный институт проблем информатики Национальной академии наук Беларуси, Институт проблем информатики Российской академии наук, Московский государственный университет им. М.В. Ломоносова, Белорусский государственный университет, Научно-исследовательский институт многопроцессорных вычислительных систем (Таганрог), Сибирский государственный университет телекоммуникаций и информатики (Новосибирск) и др.

5. *Грид-системы (Grid system).* Через средства коммуникации эти системы должны объединить все компьютеры, их программное обеспечение и профессионалов в области информатики. С помощью системы осуществляется переход на оказание информационных услуг с соответствующей оплатой.

Влияние многопроцессорных компьютеров на скорость обработки программ представлено законом Амдала.

Предположим, что в программе доля операций, которые нужно выполнять последовательно, равна  $f$ , где  $0 \leq f \leq 1$  (при этом доля понимается по числу операций в процессе выполнения).

Предельные случаи в значениях  $f$  соответствуют полностью параллельным ( $f = 0$ ) и полностью последовательным ( $f = 1$ ) программам. Для того чтобы оценить, какое ускорение  $S$  может быть получено на компьютере из  $p$  процессоров при данном значении  $f$ , можно воспользоваться законом Амдала:

$$S \leq \frac{1}{f + (1 - f) / p}.$$

Если 9/10 программы исполняются параллельно, а 1/10 по-прежнему последовательно, то ускорения более чем в 10 раз получить в принципе невозможно вне зависимости от качества реализации параллельной части кода и числа используемых процессоров.

*Прямая задача.* Какое количество процессоров необходимо взять, чтобы ускорить вычисления в  $S$  раз?

*Обратная задача.* Какую часть кода надо выполнить эффективно, чтобы получить заданное ускорение?

Ответ можно найти в следствии из закона Амдала.

*Следствие.* Для того чтобы уменьшить время выполнения программы в  $q$  раз, необходимо ускорить не менее чем в  $q$  раз не менее чем  $(1 - 1/q)$ -ю часть программы. Значит, для ускорения программы в 100 раз по сравнению с ее последовательным вариантом необходимо получить не меньшее ускорение не менее чем на 99,99% программного кода.

Чтобы вычислительная система работала с максимальной эффективностью на конкретной программе, требуется тщательное согласование структуры программы и ее алгоритма с особенностями архитектуры вычислительной системы.

# 1. ПОНЯТИЕ АРХИТЕКТУРЫ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

---

Информатика – развивающаяся наука. Слово «informatik» в данном контексте было впервые употреблено в 1968 г. федеральным министром Г. Штольтенбергом по случаю открытия научной конференции в Западном Берлине, а незадолго до этого слово «informatique» возникло во французском языке. Со временем в голландском языке вошло в употребление слово «informatika», в итальянском – «informatica», в польском – «informatyka», в русском – «информатика», в испанском – «informatica». В английском языке, по-видимому, остается «computer science» (компьютерная наука), причем этот термин в большей степени отражает теоретические аспекты данного направления.

Если считать, что термин «информация» происходит от латинского «informare», то в переводе на содержательный язык это означает «придать форму». Значит, с этимологической точки зрения информация – это акция придания структуры некоторой неопределенной массе, т.е. перед обработкой информацию следует трансформировать в форму, пригодную для различных манипуляций с ней, в том числе для анализа.

Современная информатика является результатом бурного развития исследований в различных областях знаний за последние годы. Но многие ее корни уходят далеко в историю. Собственно говоря, информатика началась тогда, когда впервые попытались механизировать так называемую «умственную деятельность». Это требовало усилий многих ученых, однако во многих отношениях основателем информатики можно считать Г.-В. Лейбница.

Лейбниц занимается (1678) универсальным символическим языком как средством для построения универсальной науки, которая в рамках одного исчисления позволяла бы давать ответы на все вопросы простейшим образом и со свойственной математике достоверностью. Сюда же относятся философские сочинения («Discours de métaphysique» («Рассуждение о метафизике»), 1685–1686) и работа над рядом технических проектов, в том числе потерпевшим неудачу проектом откачивания воды из рудников в Гарце с помощью ветряных мельниц. Построена наконец и вычислительная машина (1694), и все же это не было полным успехом, ибо механика в то время еще не достигла достаточного уровня развития. Лейбниц занимается также двоичной системой счисления. В рукописи на латинском языке, подписанной 15 марта 1679 г., ученый описывает, как выполнять вычисления в двоичной системе, а позже разрабатывает в общих чертах проект вычислительной машины, работающей в двоичной системе счисления. В многочисленных письмах и в трактате «Explication de L'Arithmétique Binaire» (1703) Лейбниц снова и снова возвращается к двоичной арифметике.

Исторически автоматизация умственной деятельности началась с цифровых вычислений, производимых над арабскими десятичными цифрами, получившими широкое распространение в Европе с начала XVI в., и так называемых «вычислений алгоритмических процессов», объектами которых могут служить, например, понятия, формулируемые с помощью символов, или логические высказывания и связи. Подробная разработка этой программы, набросок которой дал Лейбниц, позволила выяснить сущность информатики. Сюда относятся вопросы кодирования, в особенности двоичного кодирования, а также смежной области – криптографии. Разработка включает в себя и полную автоматизацию процесса вычислений (чего у Лейбница еще нет), находящую выражение в алгоритмическом мышлении и развитую в вопросах синтаксиса и семантики алгоритмических языков.

Информатика, как всякая прикладная наука, зависит от инженерно-технических возможностей соответствующего времени — от развития техники связи, техники автоматического регулирования и управления (механического, электрического или электронного), а также техники реализации запоминающих устройств, включая устройства считывания и записи.

Корень встречающегося во многих языках слова «калькуляция» (счет, вычисление, исчисление) ведет происхождение от лат. *calculi* — счетные камешки. Римский абак, китайский суан-пан, русские счеты (последние два приспособления употребляются до сих пор) служили для вычислений, которые осуществлялись перемещением счетных марок (камешков, косточек, монет) по направляющим проволочкам. Однако ни этот «счет по линиям», ни вычисления посредством счетных палочек, на которых нарезались насечки-числа (позднелат. *samputare*), не знаменуют собой начала механизации умственной деятельности. Только возникшие в Индии и пришедшие в Европу от арабов цифры, которые, будучи дополнены нулем, позволили записывать числа в позиционной системе счисления, привели к техническому решению проблемы счета, где место счетных косточек заняли зубчики шестеренок. Речь идет об изобретенной В. Шиккардом в 1623 г. суммирующей машине с переносом цифр, причем для умножения можно было пользоваться неперовыми счетными палочками, считывая с их помощью кратные данного числа.

Проект суммирующей машины Б. Паскаля был реализован к 1645 г. Наконец в 1671–1674 гг. Лейбницу удается построить машину, выполняющую все четыре арифметических действия, введя передвижной челнок-счетчик и применив фиксацию сомножителей посредством уступчатого валика. Другое техническое решение, зубчатую шестерню, нашел в 1709 г. Дж. Полени из Падуи. Около 1726 г. А. Браун в Вене, а начиная с 1770 г. Ф.-М. Хан в Эхтердингене создают машины со счетными шестеренками, концентрически расположенными вокруг уступчатого валика, — первые устройства, которыми действительно можно было пользоваться. (Эта конструкция возродилась в 1948 г. в виде похожей на четки машины «Cirta» швейцарца К. Хецштарка.) В 1666 г. С. Морлэнд в Англии, в 1678 г. Р. Грийе де Ровен во Франции, в 1722 г. К. Герстен в Гиссене изобретают сумматоры на зубчатых рейках, применяемые до наших дней. Начиная с 1818 г. серийное производство арифмометров наладил в Париже Ш.-К. Тома де Кольмар, и к 1878 г. их было продано уже около 1500 единиц.

Выполнение умножения автоматизируется в 1896 г. П.Л. Чебышевым в Париже, а выполнение деления автоматизируется в машине «Madas» по принципу, предложенному А. Рехницером в 1902 г.

Печатающее устройство впервые подключается к механической настольной счетной машине фирмой «Берроуз» в 1889 г. Однако автоматическое составление печатных матриц предусматривалось еще в машине «Difference engine» Ч. Бэббиджа, проектирование которой он начал в 1823 г. Она предназначалась для вычисления таблиц с помощью интерполяции, была доведена в Швеции П.-Г. Шойцем до возможности практического применения и работала в обсерватории Дадли в Олбэни (США).

## 1.1. Вычисления в компьютерах

Системы счисления, отличные от десятичной, были довольно привычны. Так, машина Паскаля в последних двух позициях имела шестеренки с 20 и 12 зубцами в соответствии с делением тогдашнего ливра на 20 су по 12 денье. В 1679 г. Лейбниц описывает в общих чертах машину, работающую в двоичной системе счисления.

В 1931 г. цифровые шестеренки с 8 позициями патентует Р. Вальта во Франции. В 1933 г. К. Цузе решил использовать двоичную систему счисления как естественное следствие применения электромагнитных реле, которые могут находиться в двух состояниях (якорь опущен, поднят).

В 1936 г. Л. Куффиньяль и Р. Вальта во Франции, Э. Филлипс в Англии указывают на преимущества двоичных вычислений для построения механических вычислительных устройств. Филлипс демонстрировал механическую модель для умножения в двоичной системе счисления и рекомендовал использовать для числовых таблиц восьмеричную систему счисления.

Электронные счетчики, работающие в двоичной системе, в 1931 г. использовал К. Уинн-Уильямс. Дж. фон Нейман вместе с Х. Голдстейном применяли двоичную систему в проекте «Принстонская машина», отчет о котором получил широкое распространение в 1946–1948 гг. Разработка, начатая в 1945 г. Вумерсли и Хартри в Англии, была основана на применении двоичной системы «внутри», а восьмеричной — «снаружи». Другие английские разработки (М. Уилкса, Ф. Уильямса, Т. Килберна) ориентированы на двоичную систему счисления, а американские проекты Х. Айкена, Дж. Штибица, Дж. Эккерта, Дж. Моучли остались ориентированными на десятичную систему.

К. Цузе ввел «полулогарифмическую форму», называемую сегодня представлением чисел с плавающей точкой, осознавая, что порядок числа удобнее представлять его логарифмом. В первом устройстве Цузе ZI было семь двоичных разрядов для порядка и шестнадцать двоичных разрядов для мантиссы. Эта идея не скоро проникла в США и Англию, если не считать модели Штибица (1947) на электромагнитных реле. Операции над числами с плавающей точкой впервые реализованы в шведском компьютере BARK, в советской Большой электронно-счетной машине (БЭСМ), ЭВМ «Стрела» и др., а также в мюнхенской PERM и цюрихской ERMETH.

Введение букв для обозначения произвольных математических объектов — это значительное научное достижение индийской математики раннего Средневековья. Лейбниц несколько десятилетий подряд развивал идею символического представления смысла понятий, которое позволило бы проводить вычисления над ними. В трактате «Об универсальном знании, или философском исчислении», написанном около 1680 г., он выражает надежду, что когда-нибудь научные споры можно будет разрешать посредством вычислений. Лейбниц пытался понятиям сопоставлять числа, при этом отношение подчинения понятий должно было выражаться отношением делимости чисел. По современной терминологии, он ввел структуру понятий.

Ч. Бэббидж в проекте своей аналитической машины, содержащей арифметическое устройство, память, устройство ввода и устройство печати, предусмотрел устройства, реализующие управление в зависимости от текущего результата вычислений. Сын Бэббиджа Х. Бэббидж усовершенствовал отдельные части машины так, что смог вычислить таблицу чисел, кратных 14.

Леди Лавлейс, ученица Бэббиджа, убедительно показала, что машину Бэббиджа нельзя ставить в один ряд с простыми счетными машинами: «...она занимает совершенно особое место... среди механизмов, предоставляющих возможность комбинировать произвольные символы».

П. Ладгейт (1909) понял значение условных переходов и ввел трехадресные команды. В табулирующих машинах (1936, тип германского Холлерита) и фактурных машинах, а также в связанной со счетной машиной пишущей машинке Л. Торреса де Кеведо (1910) можно найти первые попытки программирования.



К 1941 г. Цузе удалось построить первую действующую вычислительную машину с программным управлением (модель Z3). В 1942 г. в лаборатории «Белл Телефон» была сдана в эксплуатацию вычислительная машина на электромагнитных реле, которую разработал Штибиц. В 1944 г. правительство США приняло в эксплуатацию вычислительную машину «Mark I», созданную Айкеном. Первая выполненная на электронных лампах вычислительная машина ЭНИАК (ENIAC) Эккерта и Моучли, к проектированию которой приступили в 1943 г., начала функционировать в 1946 г. Это устройство не достигло той универсальности, которой обладал проект Цузе.

Основополагающая идея вычислительной машины, управляемой размещенной в ее памяти программой, впервые описана фон Нейманом и его сотрудниками Моучли и Эккертом 30 июня 1945 г. Она была развита Уилксом, который в мае 1949 г. сдал в эксплуатацию первую вычислительную машину ЭДСАК (англ. EDSAC).

В 1944 г. Моучли, Эккерт и Голдстейн ввели принцип последовательных вычислений, применив одноразрядный сумматор для последовательного суммирования  $n$ -разрядных чисел.

В 1948 г. С.А. Лебедев (СССР) независимо от фон Неймана обосновал принцип построения ЭВМ с хранимой в памяти программой.

Первая в СССР цифровая Малая электронная счетная машина (МЭСМ) создана в 1951 г. под руководством Лебедева. В 1952 г. появилась ЭВМ М-1 (руководители И.С. Брук, Н.Я. Матюхин), а в 1953 г. — ЭВМ «Стрела» (создатели Ю.Я. Базилевский, Б.И. Рамеев). В 1955 г. в СССР сформирован первый Вычислительный центр Академии наук СССР (директор А.А. Дородницын). В 1958 г. в СССР построена первая и единственная в мире ЭВМ «Сетунь» (Н.П. Брусенцов), работающая в троичной системе счисления с симметричным представлением цифр, и первая ЭВМ с системой кода вычетов (И.Я. Акушский).

В последнее время появились сведения о более ранних проектах вычислителей. В 1900 г. на дне Эгейского моря на затонувшем корабле обнаружен аналоговый механический компьютер для моделирования движения планет. Это было вычислительное устройство, способное с высокой точностью определять координаты небесных тел. Градуированные циферблаты могли вращаться независимо друг от друга с помощью дисков и приводных механизмов из зубчатых колес (все зубья выточены под углом в  $60^\circ$ ). Дифференциальная передача состояла из 30 бронзовых шестеренок, четырех циферблатов и рукоятки. Устройство создано в 87 г. н.э. Есть гипотеза, что ее изобрел Архимед. Это так называемый антикиферийский (по месту находки) вычислительный механизм. Подобную зубчатую передачу заново изобрели в XIV в.

Чтобы понять принцип действия цифровой вычислительной машины (ЦВМ), рассмотрим процесс вычислений с использованием обычного арифмометра. Вначале на лист бумаги выписываются исходные данные, формулы расчета и готовится таблица для занесения промежуточных и конечных результатов. В процессе вычислений с листа бумаги на регистры арифмометра переносятся числа, участвующие в очередной операции. На арифмометре выполняется нужная операция в соответствии с расчетной формулой, и полученный результат переписывается с регистра машины в таблицу на листе бумаги. В этом процессе счетная машина выполняет арифметические операции над числами, которые вводятся человеком. Лист бумаги служит запоминающим устройством, хранящим программу вычислений (расчетные формулы), исходные данные, промежуточные и конечные результаты. Человек осуществляет управление процессом вычислений, в том числе переносом информации с листа бумаги в счетную машину и обратно, заставляет машину выполнять необходимую операцию и выбирает нужный вариант продолжения процесса вычислений в зависимости от результата, полученного на очередном этапе счета.

Электронное арифметическое устройство существенно ускоряет процесс выполнения арифметических операций. Однако принципиальный эффект будет достигнут, если к электронному быстродействующему арифметическому устройству добавить, во-первых, быстродействующую память, которая, как лист бумаги при расчете, хранит программу вычислений, исходные данные, промежуточные и конечные результаты, и, во-вторых, быстродействующее управляющее устройство, производящее необходимый для реализации программы вычислений обмен числами между памятью и арифметическим устройством и инициирующее последнее на выполнение необходимой операции.

Если комплекс нашей аппаратуры дополнить средствами связи с внешним миром, т.е. устройствами для ввода в память данных и программы вычислений, а также устройствами вывода результатов вычислений, то придем к классической **блок-схеме ЦВМ** (рис. 1.1).

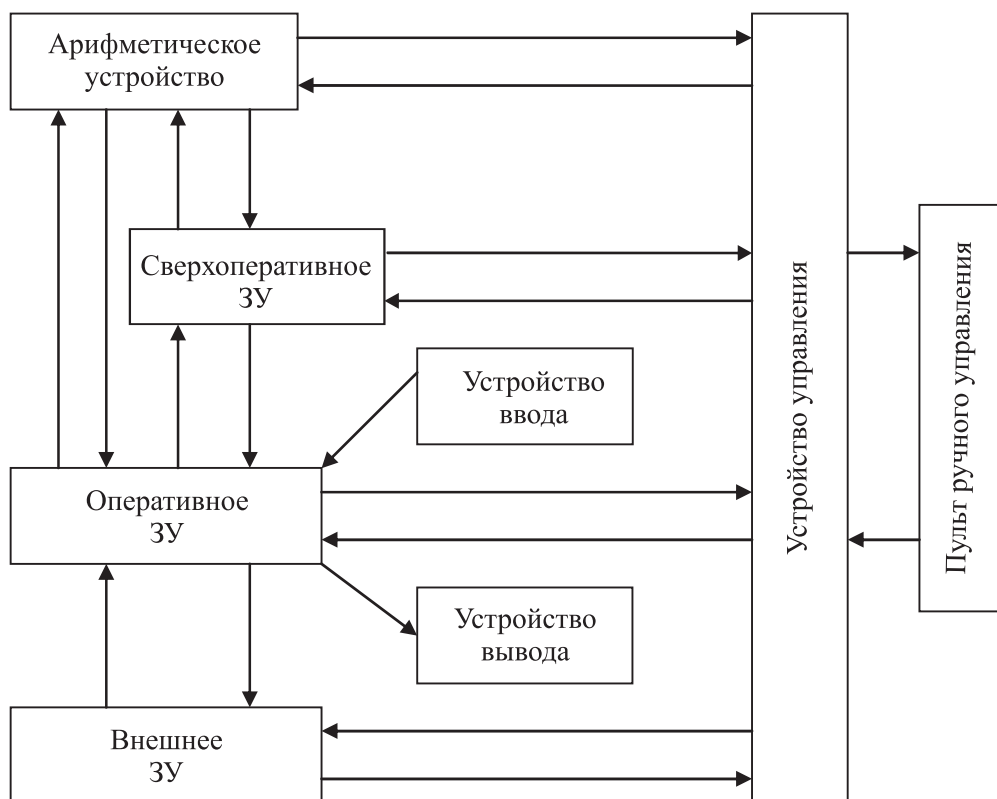


Рис. 1.1. Блок-схема цифровой вычислительной машины

ЦВМ содержит следующие основные устройства: арифметическое устройство (АУ); запоминающее устройство (ЗУ) – память; устройство управления (УУ); устройство ввода данных в машину (Вв) и вывода из нее результатов расчета (Выв); пульт ручного управления (клавиатуру).

**Арифметическое устройство** производит арифметические и логические операции над поступающими в него словами.

**Запоминающее устройство (память)** хранит информацию, передаваемую в него из других устройств, в том числе поступающую в машину извне через устройство ввода, и выдает во все другие устройства информацию, необходимую для продолжения вычислительного процесса. Память машины, как правило, состоит из двух частей: быстродействующего оперативного запоминающего устройства (ОЗУ) и сравнительно медленнодействующего внешнего запоминающего устройства (ВЗУ), способного хранить значительный объем информации.

Каждая ячейка ОЗУ служит для хранения машинного слова, т.е. кода определенной длины, представляющего число или другой тип информации. Номера ячеек называются их адресами.

Запоминающие устройства используются для считывания хранимой в них информации с целью передачи в другие устройства и записи информации, поступающей из других устройств. При считывании слова из ячейки содержимое последней не меняется и при необходимости слово может быть снова выбрано из той же ячейки. При записи хранившееся в ячейке слово стирается и его место занимает новое слово.

Объем запоминающего устройства определяется количеством хранимых слов, а его быстродействие — временем обращения, т.е. продолжительностью операции записи или считывания нужного слова. Чтобы сократить потери времени при вычислениях, быстродействие ОЗУ должно соответствовать скорости работы электронного АУ.

Многие задачи требуют памяти большого объема для хранения необходимой информации, однако технически трудно и дорого строить быстродействующее ЗУ большой емкости. В связи с этим ЦВМ кроме оперативного содержит внешнее запоминающее устройство, способное хранить большой объем информации.

ЦВМ выполняет вычисления только над данными, хранящимися в оперативном запоминающем устройстве, и лишь после окончания отдельных этапов вычислений из ВЗУ в ОЗУ передается информация, необходимая для последующих этапов решения задачи.

Если быстродействие ОЗУ оказывается недостаточным для получения нужной скорости работы машины, между АУ и ОЗУ размещают сверхоперативное запоминающее устройство (СОЗУ) на несколько десятков или сотен слов (рис. 1.1). Ячейки СОЗУ хранят главным образом промежуточные результаты и другую информацию, многократно используемую на текущем этапе вычислений.

**Устройство управления** автоматически управляет вычислительным процессом, посылая всем другим устройствам сигналы, предписывающие им те или иные действия. В частности, оно указывает ОЗУ, какие слова должны быть переданы в АУ и в другие устройства, инициирует АУ на выполнение нужной операции и помещает полученный результат в ОЗУ, базируясь на принципе программного управления.

Идея программного управления ЦВМ была существенно развита американским математиком фон Нейманом, который в 1945 г. сформулировал принцип хранимой в памяти программы. Правда, ряд исследователей считают, что принцип хранимой программы в памяти машины принадлежит создателям компьютера ENIAC — инженерам Моучли и Эккерт. Согласно этому принципу программа, закодированная в цифровом виде, хранится в памяти ЦВМ наравне с числами. В командах указываются не сами участвующие в операциях числа, а адреса ячеек ОЗУ, в которых они находятся, и адреса ячеек, куда помещаются результаты операции. Так как программа хранится в памяти, одни и те же команды при необходимости могут требовать количество раз извлекаться из памяти и выполняться. Поскольку команды представляются в машине в форме чисел, то над командами, как над числами, машина может производить операции.

С помощью *пульты ручного управления (клавиатуры)* оператор запускает и останавливает машину, а при необходимости может вмешиваться в процесс решения задачи.

## 1.2. Архитектура как набор взаимодействующих компонент

Серийно выпускаемые сверхбольшие надежные и дешевые интегральные схемы, массовое производство микропроцессоров, возобновившийся интерес к разработке языков программирования и программного обеспечения порождают необходимость в качественном улучшении программно-аппаратного интерфейса при проектировании компьютеров, т.е. в установлении семантической связи между возможностями аппаратных средств современных компьютеров и их программным обеспечением. Организация вычислительной системы на этом уровне лежит в основе понятия «архитектура». Для неспециалистов в области программного обеспечения термин «архитектура» ассоциируется, как правило, со строительными объектами. И такую аналогию действительно можно провести.

Архитектура компьютера, характеризующая его логическую организацию, может быть представлена как множество взаимосвязанных компонент, включающих, на первый взгляд, элементы различной природы — программное обеспечение (software), аппаратное обеспечение (hardware), алгоритмическое обеспечение (brainware), специальное фирменное обеспечение (firmware) — и поддерживающих его слаженное функционирование в форме единого архитектурного ансамбля, позволяющего вести эффективную обработку различных объектов.

Кроме того, архитектура может быть задана как абстрактное многоуровневое представление физической системы с точки зрения программиста, с закреплением функций за каждым уровнем и установлением интерфейса между различными уровнями.

В зависимости от типов данных и операций, поддерживаемых hardware, можно рассматривать архитектуру системы на различных уровнях. Первые компьютеры поддерживали только один тип данных — целые числа со знаком, что требовало от программиста колоссальных усилий по организации вычислений, в частности решения вопроса масштабирования чисел. С появлением языка Fortran, определившего два типа данных — Real и Integer, возникла необходимость аппаратной реализации этих типов данных. Сегодняшние компьютеры аппаратно поддерживают разнообразные типы данных — от битовых последовательностей до атомарных объектов машинной графики (пикселов), контролируя при этом допустимость выполнения тех или иных операций над данными определенного типа.

Знание особенностей разнообразных архитектурных решений дает возможность пользователям компьютеров эффективно распоряжаться всеми предоставляемыми ресурсами, осуществляя их направленный выбор и тем самым повышая эффективность обработки данных.

Ранее область применения вычислительных систем определялась ее быстродействием. Однако существует достаточно большое количество вычислительных систем (ВС), обладающих равным быстродействием, но имеющих совершенно разные способы представления данных, методы организации памяти, режимы работы, системы команд, набор внешних устройств и т.д. Таким образом, ВС имеет кроме быстродействия ряд других характеристик, необычайно важных в той или иной области применения. Совокупность таких характеристик и легла в основу понятия архитектуры ВС.

**Архитектура вычислительной системы** — это совокупность основных функциональных возможностей системы, сфер применения (научно-техническая, экономи-

ческая, 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.

Стандарт IEEE Std 1472000 определяет термин «архитектура» следующим образом: **архитектура** – это базовая организация системы, воплощенная в ее компонентах, их отношениях между собой и с окружением, а также принципы, определяющие проектирование и развитие системы.

Составные части понятия «архитектура» можно представить в виде схемы (рис. 1.2).

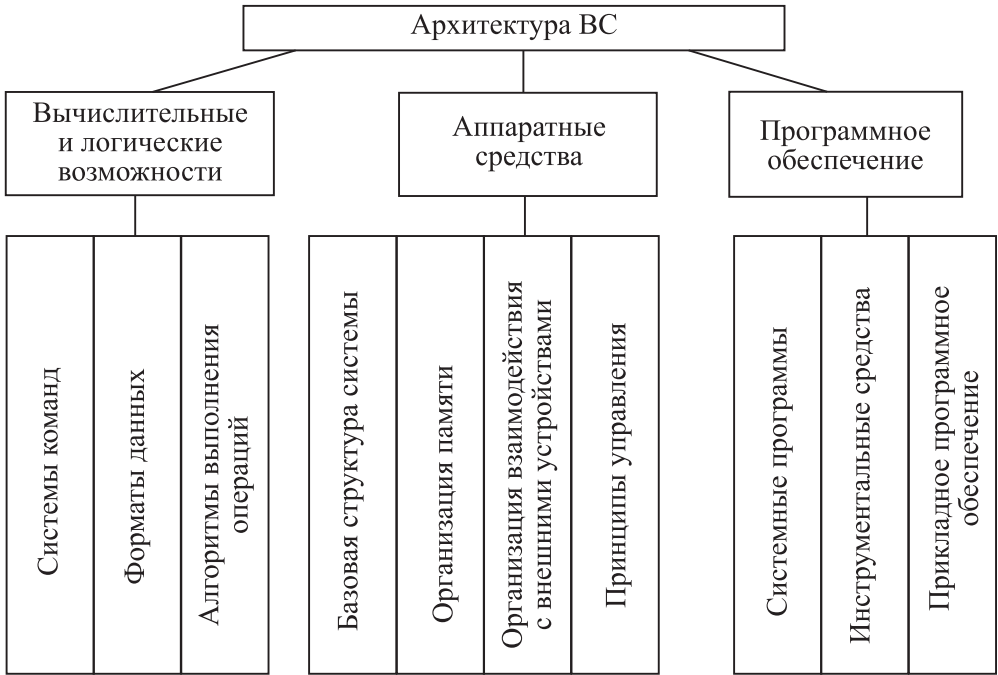


Рис. 1.2. Функциональные возможности ВС

**Вычислительные и логические возможности ВС.** Возможности вычислительной системы обуславливаются системой команд (СК), характеризующей гибкость программирования, форматами данных и скоростью выполнения операций, определяющих класс задач, наиболее эффективно решаемых на ВС. Система команд ВС, базирующихся на архитектуре фон Неймана, сегодня мало чем отличается от СК ЭВМ 1950-х гг. Большинство достижений в этой области остались незамеченными проектировщиками и соответственно не нашли адекватного воплощения в архитектуре современных компьютеров.

Анализ показывает, что в различных программах чаще всего встречаются достаточно простые команды: команды пересылки и команды процессора с использованием регистров и простых режимов адресации. Не нашли широкого применения и нетрадиционные способы кодирования данных, несмотря на значительные воз-



возможности их в плане разработки быстродействующих алгоритмов арифметических операций и повышения надежности вычислений. Среди них знакоразрядные системы, системы в коде вычетов и др.

Рассмотрим структуру системы команд в зависимости от класса решаемых задач (рис. 1.3).



Рис. 1.3. Классификация СК по назначению

К командам управления относятся команды ввода-вывода данных и команды управления состоянием процессора, памяти и каналов.

Как видно из рис. 1.3, для решения задач любого класса необходимы команды, востребованные задачами управления. Следовательно, эти типы команд должны присутствовать в любом компьютере.

Большое влияние на точность выполнения операций оказывают форматы данных. Современные компьютеры имеют развитую систему форматов. Например, компьютеры фирм ЕС ЭВМ и IBM имеют форматы в 2, 4, 8 и 16 байт.

Алгоритмы выполнения операций достаточно полно отражают производительность только однопроцессорных ВС.

**Аппаратные средства.** Простейшая ВС включает модули пяти типов: центральный процессор, основную память, каналы, контроллеры и внешние устройства.

**Процессор** (УУ + АЛУ + ЗУ) управляет работой системы и обеспечивает вычисления непосредственно по программе. Выполнение машинных команд, команд ввода-вывода (I/O), обращение к памяти, управление состоянием устройств инициализируются или выполняются с помощью процессора.

**Основная память** предназначена для хранения команд и данных и обеспечивает адресный доступ к ним от процессора. Современная память работает со скоростью, близкой к скорости работы процессора.

**Каналы** – спецустройства, управляющие обменом данных с внешними устройствами. Каналы инициализируют свою работу с помощью процессора и затем переходят в автономный режим работы. Это, по сути, спецпроцессор ввода-вывода, обеспечивающий работу внешних устройств, контроль информации и т.д.

**Контроллеры ввода-вывода** служат для подсоединения внешних устройств (ВнУ) к каналам и обеспечивают обмен управляющей информацией с ВнУ, присвоение приоритетов и выдачу информации о состоянии ВнУ для канала, т.е. это устройства управления ВнУ.

**Внешние устройства** служат для ввода-вывода информации с различных носителей.

Память может быть организована:

- как многоуровневая с различным объемом, стоимостью хранения данных в расчете на один бит и временем доступа к ней: внутренние регистры процессора (ВНР), сверхоперативная, оперативная, внешняя память;
- как одноуровневая, виртуальная.

Почти всегда виртуальная память есть переупорядоченное подмножество реальной памяти.

Более высокий уровень памяти быстрее, меньше по объему и дороже в пересчете на бит, чем более низкий ее уровень. Кэш-память является компромиссом между стоимостью памяти и ее объемом за счет динамического копирования в быстродействующее ЗУ наиболее часто используемой информации из медленного ЗУ.

Уровни иерархии памяти взаимосвязаны между собой: все данные одного уровня могут быть найдены на более низком уровне.

**Оперативное запоминающее устройство (ОЗУ)**, или **оперативная память (ОП)**, **память с произвольным доступом (RAM – Random Access Memory)**, отвечает на все запросы процессора, которые не может удовлетворить кэш-память.

**Постоянное запоминающее устройство (ПЗУ)**, или **постоянная память (ROM – Read Only Memory)**, предназначено только для чтения, является энергонезависимым, обладает высоким быстродействием и малой стоимостью. Содержимое постоянной памяти определяется заводом-изготовителем и в дальнейшем не изменяется.

Устройство ввода-вывода, как правило, состоит из двух компонент: непосредственно устройства и его контроллера. Последний состоит из набора микросхем, которые управляют устройством на физическом уровне. Программа, посылающая команды контроллеру и реагирующая на отклики, называется драйвером.

Для подключения к компьютеру низкоскоростных устройств ввода-вывода (например, клавиатуры и мыши) существует универсальная последовательная шина USB (Universal Serial Bus).

Шина IEEE 1394 обеспечивает пакетную передачу данных с большой скоростью, что позволяет подключать к компьютеру различные мультимедийные устройства, в частности цифровую видеокамеру.

С каждым процессом связано собственное адресное пространство и набор ресурсов, среди которых регистры, список открытых файлов, необработанные прерывания, список связанных процессов. Существенное влияние на производительность ВС оказывают каналы ввода-вывода. Мультиплексный канал обеспечивает работу группы медленных устройств, блок-мультиплексный – работу группы быстрых устройств, селекторный канал монополизует информационную магистраль только одним быстродействующим устройством.

Для повышения пропускной способности каналов используют некоторые дополнительные меры, например буферизацию ВнУ путем введения памяти в состав самого устройства или контроллера.

Аппаратные средства защиты памяти служат для управления доступом к различным областям памяти в соответствии с имеющимися у пользователя полномочиями.

**Программное обеспечение.** Программное обеспечение является составной частью архитектуры компьютера и существенно влияет на весь вычислительный процесс, в частности позволяет эффективно эксплуатировать аппаратные средства системы.

**Операционная система (ОС)** управляет ресурсами, разрешает конфликтные ситуации, оптимизирует функционирование системы в целом.

# СОДЕРЖАНИЕ

Список основных сокращений .....	3
Предисловие .....	5
Введение .....	7
<b>1. ПОНЯТИЕ АРХИТЕКТУРЫ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ .....</b>	<b>12</b>
1.1. Вычисления в компьютерах .....	13
1.2. Архитектура как набор взаимодействующих компонент .....	18
1.3. Архитектура как интерфейс между уровнями физической системы .....	23
1.4. Особенности проектирования программной архитектуры .....	26
1.5. Семантический разрыв .....	30
1.6. Анализ архитектурных принципов фон Неймана .....	34
1.7. Способы совершенствования архитектуры .....	36
1.7.1. Хранение информации в виде самоопределяемых данных .....	36
1.7.2. Области санкционированного доступа .....	37
1.7.3. Одноуровневая память .....	38
1.8. RISC- и CISC-архитектуры .....	39
1.8.1. Основные принципы RISC-архитектуры .....	40
1.8.2. Отличительные черты RISC- и CISC-архитектур .....	40
1.8.3. Проблемы реализации RISC-процессоров .....	42
1.9. Функционирование управляющего компьютера .....	45
<i>Вопросы и задания .....</i>	<i>47</i>
<b>2. КОНВЕЙЕРИЗАЦИЯ .....</b>	<b>48</b>
2.1. Конвейерная обработка .....	48
2.2. Классификация конфликтов по данным .....	50
2.3. Предикация .....	53
2.4. Конвейерные системы .....	58
<i>Вопросы и задания .....</i>	<i>60</i>
<b>3. ОРГАНИЗАЦИЯ ПАМЯТИ .....</b>	<b>61</b>
3.1. Иерархия памяти .....	61
3.2. Регистровая память .....	62
3.3. Организация кэш-памяти .....	63
3.3.1. Принципы создания кэш-памяти .....	66
3.3.2. Простейшая кэш-память .....	66
3.4. Концепция виртуальной памяти .....	70
3.4.1. Задачи, решаемые виртуальной памятью .....	70
3.4.2. Страничная организация памяти .....	71
3.5. Оперативные и постоянные запоминающие устройства .....	74
3.6. Консистентность данных в вычислительных системах .....	79
3.7. Дополнительная память .....	85
3.8. Управление памятью .....	87
<i>Вопросы и задания .....</i>	<i>91</i>

<b>4. КОМПЬЮТЕРНЫЕ СЕТИ</b> .....	92
4.1. Эталонная модель сети .....	94
4.2. Топология локальных сетей .....	97
4.3. Различные типы шин. Микросхемы процессоров и шины .....	102
4.4. Простое средство связи в сетях (шина) .....	108
4.4.1. Связь компьютера с периферийными устройствами .....	108
4.4.2. Взаимодействие двух компьютеров .....	110
4.4.3. Объединение нескольких компьютеров. Характеристики линии передачи данных .....	112
4.5. Сетевые технологии (Lan/Wan) .....	116
4.5.1. Подключение сетевых компонентов .....	116
4.5.2. Сетевой адаптер .....	120
4.5.3. Модель IEEE «Project 802» .....	121
4.5.4. Глобальные сети .....	122
<i>Вопросы и задания</i> .....	123
<b>5. КОДИРОВАНИЕ ДАННЫХ В КОМПЬЮТЕРАХ</b> .....	124
5.1. Система двоичного кодирования .....	124
5.2. Кодирование текстовых данных .....	124
5.3. Кодирование графической информации .....	126
5.4. Кодирование звуковой информации .....	127
5.5. Формы чисел с фиксированной и плавающей точкой .....	128
5.6. Помехозащищенные коды .....	130
5.6.1. Код «2 из 5» .....	132
5.6.2. Коды Хемминга .....	132
5.6.3. Коды Рида – Соломона .....	134
5.6.4. Контроль по модулю .....	134
5.6.5. Сравнительные характеристики корректирующих кодов .....	136
5.7. Кодирование данных с симметричным представлением цифр .....	140
5.8. Кодирование данных в системах с отрицательным основанием .....	142
5.9. Кодирование данных с помощью вычетов .....	147
5.10. Стандарт IEEE 754 .....	154
5.11. Особенности вычислений с числами конечной точности .....	156
<i>Вопросы и задания</i> .....	157
<b>6. МИКРОПРОЦЕССОРЫ</b> .....	158
6.1. Методы адресации и типы команд .....	159
6.2. Компьютеры со стековой архитектурой .....	160
6.3. Процессоры с микропрограммным управлением .....	164
6.3.1. Горизонтальное микропрограммирование .....	165
6.3.2. Вертикальное микропрограммирование .....	167
6.4. Процессоры с архитектурой 80x86 и «Pentium» .....	168
6.5. Особенности процессоров с архитектурой SPARC компании «Sun Microsystems» .....	172
6.6. Процессоры PA-RISC компании «Hewlett Packard» .....	178
6.7. Процессор MC 88110 компании «Motorola» .....	182
6.8. Архитектура MIPS компании «MIPS Technology» .....	184
6.9. Особенности архитектуры «Alpha» компании DEC .....	187

6.10. Особенности архитектуры «Power» . . . . .	191
6.11. Многоядерные процессоры . . . . .	198
6.11.1. Варианты создания многоядерных систем . . . . .	199
6.11.2. Особенности архитектурных решений «Hyper Transport» . . . . .	201
6.11.3. Асимметричная многоядерная архитектура . . . . .	202
6.12. Графические процессоры . . . . .	204
6.12.1. Принцип работы графического процессора . . . . .	205
6.12.2. Недостатки графического конвейера . . . . .	206
6.12.3. Унифицированный потоковый процессор . . . . .	209
6.12.4. Использование графического процессора для общих вычислений . . . . .	215
6.12.5. Параллельные вычисления на CPU и GPU . . . . .	217
6.12.6. Программирование CUDA . . . . .	218
6.13. Квантовые процессоры . . . . .	219
6.13.1. Квантовая схема . . . . .	221
6.13.2. Применение квантовых компьютеров . . . . .	223
<i>Вопросы и задания</i> . . . . .	225
<b>7. ВЗАИМОДЕЙСТВИЕ ПРОЦЕССОВ И УПРАВЛЕНИЕ ИМИ</b> . . . . .	226
7.1. Понятие процесса и состояния . . . . .	226
7.2. Графическое представление процессов . . . . .	228
7.3. Управление процессами в многопроцессорном компьютере . . . . .	230
7.4. Управление процессами в однопроцессорном компьютере . . . . .	231
7.5. Форматы таблиц процессов . . . . .	232
7.6. Синхронизация процессов . . . . .	233
7.7. Операции <i>P</i> и <i>V</i> над семафорами . . . . .	234
7.8. Почтовые ящики . . . . .	235
7.9. Монитор Хоара . . . . .	236
7.10. Проблема тупиков . . . . .	236
7.11. Организация системы прерывания . . . . .	239
7.11.1. Основные параметры системы прерывания . . . . .	239
7.11.2. Вход в прерывающую программу . . . . .	242
7.11.3. Приоритетное обслуживание прерываний . . . . .	243
7.11.4. Функционирование типовой системы прерывания . . . . .	246
<i>Вопросы и задания</i> . . . . .	248
<b>8. ПРОЦЕССЫ ПОСЛЕДОВАТЕЛЬНЫЕ И ПАРАЛЛЕЛЬНЫЕ</b> . . . . .	249
8.1. Предпосылки создания систем параллельного действия . . . . .	251
8.2. Отношение предшествования процессов . . . . .	253
8.3. Типы параллелизма . . . . .	255
8.4. Информационные модели . . . . .	257
8.4.1. Мультипроцессоры . . . . .	257
8.4.2. Мультикомпьютеры . . . . .	258
8.4.3. Сети межсоединений . . . . .	262
8.5. Программное обеспечение для мультикомпьютеров . . . . .	262
8.5.1. Система PVM . . . . .	263
8.5.2. Система MPI . . . . .	264
8.6. Повышение эффективности функционирования компьютеров . . . . .	265
8.6.1. Эффективность вычислений . . . . .	266



8.6.2. Основные подходы к проектированию программного обеспечения для параллельных компьютеров	267
8.6.3. Модели управления	268
8.6.4. Степень распараллеливания процессов	270
8.6.5. Вычислительные парадигмы	271
8.6.6. Методы коммутации	272
8.7. Алгоритмы выбора маршрутов для доставки сообщений	273
8.8. Метрика аппаратного и программного обеспечения	273
8.9. Классификация компьютеров	275
8.10. Некоторые модели параллельных программ	277
8.11. Формальная модель программ на сетях Петри	281
<i>Вопросы и задания</i>	286
<b>9. СИСТЕМЫ ПАРАЛЛЕЛЬНОГО ДЕЙСТВИЯ</b>	287
9.1. Вычислительные системы и многомашинные комплексы на базе однопроцессорных компьютеров	287
9.2. Многопроцессорный вычислительный комплекс «Эльбрус»	291
9.2.1. Структура вычислительного комплекса	291
9.2.2. Системы программирования комплекса «Эльбрус»	293
9.3. Матричные компьютеры	295
9.4. Концепции вычислительных систем с комбинированной структурой	296
9.5. Архитектура типа «гиперкуб»	299
9.6. Нейрокомпьютеры	300
9.6.1. Предпосылки создания нейрокомпьютеров	301
9.6.2. Функционирование нейрокомпьютеров	304
9.6.3. Формальная модель нейрона Маккалока – Питтса	306
9.6.4. Пример решения задачи на модели одного нейрона	307
9.6.5. Однослойная нейронная сеть	308
9.6.6. Нейроматематика	311
9.6.7. Структура нейрокомпьютера	313
9.7. Процессоры с архитектурой VLIW	314
9.7.1. Структура процессора «Itanium»	315
9.7.2. Параллелизм	315
9.7.3. Предикация и загрузка по предположению	316
9.8. Поточковые компьютеры	317
9.8.1. Концепция управления потоком данных	318
9.8.2. Граф потока операндов	319
9.8.3. Языки потока операндов	320
9.8.4. Принцип однократного присваивания	321
9.8.5. Система LAU: многопроцессорная система с управлением потоком операндов	322
9.9. Суперкомпьютеры	322
9.10. Производительность вычислительных систем	328
<i>Вопросы и задания</i>	330
<b>10. КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ</b>	331
10.1. Коммутаторы вычислительных систем	331
10.2. Коммуникационная среда SCI	337

10.2.1. Структура коммуникационных сред на базе SCI	337
10.2.2. Логическая структура SCI	341
10.2.3. Архитектура SCI	343
10.2.4. Когерентность кэш-памяти	344
10.2.5. Функциональная организация узла SCI	346
10.3. Коммуникационная среда Murginet	347
10.3.1. Структура среды Murginet	348
10.3.2. Коммутаторы и маршрутизация	349
10.3.3. Логический уровень	350
10.4. Коммуникационная среда InfiniBand	351
<i>Вопросы и задания</i>	353

## **11. ЯЗЫКИ ОПИСАНИЯ ПАРАЛЛЕЛЬНЫХ ПРОЦЕССОВ** . . . . . 354

11.1. Основные подходы к проектированию языков параллельного программирования	354
11.2. Примеры языков параллельного программирования	357
11.2.1. P-язык	357
11.2.2. ЯПФ-язык	357
11.2.3. K-язык	358
11.2.4. Язык диспозиций	358
11.2.5. Язык Occam	362
11.2.6. Язык Erlang	363
11.3. Преобразование последовательных программ в последовательно-параллельные	365
11.4. Распределение задач по процессорам	368
11.5. Планирование в мультипрограммных системах	370
11.5.1. Планирование по наивысшему приоритету	371
11.5.2. Метод круговорота (карусель)	371
11.5.3. Очереди с обратной связью	372
11.5.4. Многоуровневое планирование	373
11.5.5. Генетический алгоритм	373
<i>Вопросы и задания</i>	374

## **12. ЦЕЛОСТНОСТЬ, СЖАТИЕ И ЗАЩИТА ДАННЫХ** . . . . . 376

12.1. Сжатие данных	376
12.1.1. Простые алгоритмы	377
12.1.2. Сжатие документов	378
12.1.3. Программы для обработки документов	380
12.1.4. Кодирование цветных изображений	381
12.1.5. Сжатие цветных изображений	382
12.1.6. Алгоритмы сжатия видеоинформации	383
12.1.7. Инструменты разработчиков	385
12.2. Методы защиты информации	386
12.2.1. Классификация и особенности программных методов защиты от копирования	386
12.2.2. Способы увеличения эффективности и надежности защиты от копирования	388
12.2.3. Особенности защиты информации в компьютерных сетях	390

12.3. Контроль данных	393
12.3.1. Специфика передачи информации в вычислительных системах	393
12.3.2. Классификация ошибок и их характеристики	394
12.3.3. Методы обнаружения и исправления ошибок в ЭВМ	395
12.3.4. Программные методы контроля	396
12.3.5. Безопасность облачных вычислений	398
<i>Вопросы и задания</i>	400
Приложения	401
<i>Приложение 1</i>	401
<i>Приложение 2</i>	403
<i>Приложение 3</i>	405
Литература	407

Учебное издание

**Буза Михаил Константинович**

**АРХИТЕКТУРА КОМПЬЮТЕРОВ**

Учебник

Редактор *Е.В. Савицкая*

Художественный редактор *Т.В. Шабунько*

Технический редактор *А.Н. Бабенкова*

Корректоры *Л.Н. Макейчик, Т.В. Кульнис, Е.А. Бабич*

Компьютерная верстка *А.Н. Бабенковой*

Подписано в печать 25.11.2015. Формат 70×100/16. Бумага офсетная. Гарнитура «NewtonС».  
Офсетная печать. Усл. печ. л. 33,8. Уч.-изд. л. 34,7. Тираж 500 экз. Заказ 505.

Республиканское унитарное предприятие «Издательство “Вышэйшая школа”».

Свидетельство о государственной регистрации издателя, изготовителя,  
распространителя печатных изданий № 1/3 от 08.07.2013.

Пр. Победителей, 11, 220048, Минск.

e-mail: market@vshph.com <http://vshph.com>

Открытое акционерное общество «Полиграфкомбинат им. Я.Коласа».

Свидетельство о государственной регистрации издателя, изготовителя,  
распространителя печатных изданий № 2/3 от 04.10.2013.

Ул. Корженевского, 20, 220024, Минск.