

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.

ISBN 978-985-06-2652-3

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

СПИСОК ОСНОВНЫХ СОКРАЩЕНИЙ

АЛУ	– арифметико-логическое устройство
АО	– арифметическая операция
АПА	– аппарат преобразования адресов
ВнУ	– внешнее устройство
ВС	– вычислительная система
ЗУ	– запоминающее устройство
МС	– многопроцессорная система
МП	– микропроцессор
НМД	– накопитель на магнитных дисках
ОЗУ	– оперативное запоминающее устройство
ОП	– оперативная память
ОС	– операционная система
ПВВ	– процессор ввода-вывода
ПЗУ	– постоянное запоминающее устройство
ПК	– персональный компьютер
ПО	– программное обеспечение
ПУ	– периферийное устройство
ПЭ	– процессорный элемент
РОН	– регистр общего назначения
СК	– система команд
СКВ	– система в коде вычетов
СОЗУ	– сверхоперативное запоминающее устройство
ССП	– слово состояния программы
УВМ	– управляющая вычислительная машина
УУ	– устройство управления
ФИУ	– функциональное исполнительное устройство
ФУ	– функциональное устройство
ЦП	– центральный процессор
ЯВУ	– язык высокого уровня
ЯП	– язык программирования
ЯПП	– язык параллельного программирования
ЯПФ	– ярусно-параллельные формы
ADSL	– Asymmetric Digital Subscriber Line
ASIC	– Application Specific Integrated Circuits
BIOS	– Basic Input-Output System
BREQ	– Bus Request Signal
CAL	– Compute Abstraction Layer
CD	– Compact Disk
CF	– Compact Flash
CISC	– Complete Instruction Set Computer
COMA	– Cache Only Memory Access
CPU	– Central Processing Unit
CRC	– Cyclic Redundancy Code (Check)
DSM	– Distributed Shared Memory
EDO	– Extended Date Output
EMS	– Extended Memory System
FB	– Feedback
FIFO	– First In, First Out
FILO	– First In, Last Out
FPGA	– Field Programmable Gate Arrays
FSM	– Finite State Machine
GPU	– Graphics Processing Unit

GPGPU – General Purpose Graphics Processing Units
 HD – High Density
 HDLC – High Level Data Link Control
 HPF – Highest Priority First
 HT – Hyper Transport
 HTT – Hyper-Threading Technology
 IP – Internet Protocol
 ISA – Industry Standard Architecture
 ISO – International Organization for Standardization
 LAN – Local Area Network
 LD – Low Density
 LDT – Lightning Data Transport
 LLC – Logical Link Control
 LRU – Least Recently Used
 MAC – Media Access Control
 MAP – Multy Associative Processor
 MIMD – Multiple Instruction Stream / Multiple Data Stream
 MIPS – Millions of Instructions Per Second
 MISD – Multiple Instruction Stream / Single Data Stream
 MISP – Multiple Instruction Stream Processing
 MPI – Message Passing Interface
 MPP – Massively Parallel Processing
 NUMA – Non Uniform Memory Access
 OpenCL – Open Computing Language
 OSI – Open System Interconnection
 PCI – Peripheral Component Interconnect
 PMA – Physical Medium Attachment
 PS – Physical Signaling
 PVM – Parallel Virtual Machine
 RISC – Reduced Instruction Set Computer
 ROM – Read Only Memory
 ROP – Raster Operations
 RPC – Remote Procedure Call
 RR – Round Robin
 SCI – Scalable Coherent Interface
 SIMD – Single Instruction Stream / Multiple Data Stream
 SISD – Single Instruction Stream / Single Data Stream
 SJF – Shortest Job First
 SMP – Symmetrical Multiprocessing
 SP – Unified Streaming Processors
 SPARC – Scalable Processor Architecture
 SPMD – Single Program Multiple Data
 SR – Second hand Resource
 SUMA – Slightly Uniform Memory Architecture
 TA – Texture Address Unit
 TCP – Transmission Control Protocol
 TF – Texture Filtering Unit
 TLP – Thread Level Parallelism
 UDP – User Datagram Protocol
 UMA – Uniform Memory Access
 USB – Universal Serial Bus
 VDSL – Very high data rate DSL
 VLIW – Very Long Instruction Word
 WAN – Wide Area Network

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, massively-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 «Orteron 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-

строгое 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.

СОДЕРЖАНИЕ

Список основных сокращений	3
Предисловие	5
Введение	7
1. ПОНЯТИЕ АРХИТЕКТУРЫ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ	12
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>	47
2. КОНВЕЙЕРИЗАЦИЯ	48
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>	60
3. ОРГАНИЗАЦИЯ ПАМЯТИ	61
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>	91

4. КОМПЬЮТЕРНЫЕ СЕТИ	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
5. КОДИРОВАНИЕ ДАННЫХ В КОМПЬЮТЕРАХ	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
6. МИКРОПРОЦЕССОРЫ	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
7. ВЗАИМОДЕЙСТВИЕ ПРОЦЕССОВ И УПРАВЛЕНИЕ ИМИ	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
8. ПРОЦЕССЫ ПОСЛЕДОВАТЕЛЬНЫЕ И ПАРАЛЛЕЛЬНЫЕ	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
9. СИСТЕМЫ ПАРАЛЛЕЛЬНОГО ДЕЙСТВИЯ	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
10. КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ	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 Myrinet	347
10.3.1. Структура среды Myrinet	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. К-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.

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