МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Кузбасский государственный технический университет имени Т. Ф. Горбачева»

Кафедра информационной безопасности

Составители
Е. В. Прокопенко
И. В. Чичерин

ОРГАНИЗАЦИЯ ЭВМ И ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Методические материалы

Рекомендованы учебно-методической комиссией специальности 10.05.03 Информационная безопасность автоматизированных систем в качестве электронного издания для использования в образовательном процессе

Кемерово 2018

Рецензенты

Стенин Д. В.– кандидат технических наук, доцент директор ИИТМА

Сыркин И. С. – кандидат технических наук, доцент кафедры информационных и автоматизированных производственных систем

Прокопенко Евгения Викторовна

Чичерин Иван Владимирович

Организация ЭВМ и вычислительных систем: методические материалы [Электронный ресурс] для обучающихся специальности 10.05.03 Информационная безопасность автоматизированных систем очной формы обучения / сост. Е. В. Прокопенко, И. В. Чичерин; КузГТУ. – Электрон. издан. – Кемерово, 2018.

© КузГТУ, 2018

© Е. В. Прокопенко,
И. В. Чичерин,
составление, 2018

Введение

Без вычислительных машин, или компьютеров в настоящее время невозможна ни одна сфера человеческой деятельности. Компьютеры проникли не только в сферу производства, но и в домашний быт; многие из нас проводят целые часы за экраном своего компьютера, получая последние новости, биржевые сводки, цены, технические сведения, прогнозы погоды и многое другое из сети Интернет, а также используют компьютер для игр и развлечений.

Термин «электронная вычислительная машина», или ЭВМ совершенно не означает, что с ее помощью мы выполняем лишь какие-либо вычисления. Более того, мы уже давно имеем дело не просто с вычислительными машинами, а с системами обработки данных, т. е. системами, способными хранить информацию, редактировать, обновлять, выполнять сортировку и поиск нужных данных, формировать таблицы, диаграммы и отчеты, осуществлять логические преобразования, выдачу результатов и т. п.

Небольшие по размерам, но обладающие достаточной «вычислительной мощностью» компьютеры стали широко доступными, а объединение нескольких машин между собой, т. е. создание компьютерных сетей, привело к возможности получения самой разнообразной информации. В настоящее время машинами пользуется все цивилизованное человечество; это стало возможным благодаря «дружественным» программам и интерфейсам, ставшими непременной частью машины, и огромным объемам информации, которую теперь стало легко получать с помощью компьютера.

1. Архитектуры, характеристики, классификация ЭВМ

Электронно-вычислительная машина (сокращённо ЭВМ) – комплекс технических, аппаратных и программных средств, предназначенных для автоматической обработки информации, вычислений, автоматического управления. При этом основные функциональные элементы (логические, запоминающие, индикационные и др.) выполнены на электронных элементах.

Под архитектурой ЭВМ понимается общая функциональная и структурная организация машины, определяющая методы кодирования данных, состав, назначение, принципы взаимодействия технических средств и программного обеспечения.

Можно выделить следующие важные для пользователя группы характеристик ЭВМ, определяющих ее архитектуру:

- характеристики машинного языка и системы команд (количество и номенклатура команд, их форматы, системы адресации, наличие программно-доступных регистров в процессоре и т. п.), которые определяют алгоритмические возможности процессора ЭВМ;
- технические и эксплуатационные характеристики ЭВМ;
- характеристики и состав модулей базовой конфигурации ЭВМ;
- состав программного обеспечения ЭВМ и принципы его взаимодействия с техническими средствами ЭВМ.

Исторически первыми появились однопроцессорные архитектуры. Классическим примером однопроцессорной архитектуры является архитектура фон Неймана со строго последовательным выполнением команд: процессор по очереди выбирает команды программы и также по очереди обрабатывает данные. По мере развития вычислительной техники архитектура фон Неймана обогатилась сначала конвейером команд, а затем многофункциональной обработкой и по классификации М. Флина получила обобщенное название SISD (Single Instruction Single Data – один поток команд, один поток данных). Основная масса современных ЭВМ функционирует в соответствии с принципом фон Неймана и имеет архитектуру класса SISD.

Данная архитектура породила CISC, RISC и архитектуру с суперскалярной обработкой.

Архитектуры класса SISD охватывают те уровни программного параллелизма, которые связаны с одинарным потоком данных.

Параллелизм циклов и итераций тесно связан с понятием множественности потоков данных и реализуется векторной обработкой. В классификации компьютерных архитектур М. Флина выделена специальная группа однопроцессорных систем с параллельной обработкой потоков данных – SIMD (Single Instruction Multiple Data, один поток команд – множество потоков данных).

Ведущие поставщики микропроцессоров ищут пути повышения их производительности и снижения энергопотребления за счет использования многоядерных структур процессоров и многопотоковой обработки команд.

Процедура выполнения команд процессором включает несколько характерных этапов. В простейшем случае можно выделить, как минимум, четыре этапа обработки команд: выборка (ВК), декодирование (ДК), выполнение операции (ОП) и запись результата (ЗР).

Каждый этап в процессоре выполняется за один такт. При последовательной обработке команд, выполнение следующей (n + 1)-й команды начинается только после завершения предыдущей (n)-й команды. Это приводит к низкой производительности и простоям аппаратуры процессора.

Совмещенные принципы обработки (конвейер команд) существенно увеличивают пропускную способность процессора.

Смысл термина «суперскалярная обработка» заключается в том, что в аппаратуру процессора закладываются средства, позволяющие одновременно выполнять две или более скалярные операции, т. е. команды обработки пары чисел. Суперскалярная архитектура базируется на многофункциональном параллелизме и позволяет увеличить производительность компьютера пропорционально числу одновременно выполняемых операций.

Реализация суперскалярной обработки заключается в чисто аппаратном механизме выборки из буфера инструкций (или кэша инструкций) несвязанных команд и параллельном запуске их на исполнение.

Суперскалярная аппаратура динамически строит план вычислений на основе последовательного кода программы. Хотя такой подход и увеличивает сложность физической реализации, скалярный процессор создает план, используя преимущества тех факторов, которые могут быть определены только во время выполнения.

Этот метод хорош тем, что он «прозрачен» для программиста, составление программ для подобных процессоров не требует никаких специальных усилий, ответственность за параллельное выполнение операций возлагается в основном на аппаратные средства.

Суперскалярная обработка широко используется в современных процессорах корпораций Intel, Advanced Micro Devices (AMD), International Business Machines (IBM), Sun Microsystems и др.

Архитектура SISD породила целый ряд архитектур: CISC, RISC, VLIW и EPIC-концепцию.

Компьютеры с CISC (Complex Instruction Set Computer)-архитектурой имеют комплексную (полную) систему команд, под управлением которой выполняются всевозможные операции типа «память – память», «память – регистр», «регистр – память», «регистр – регистр».

CISC-архитектура появилась еще на заре вычислительной техники. Лидером в разработке микропроцессоров с полным набором команд считается компания Intel со своей серией процессоров x86, Pentium, Intel Core и др. Эта архитектура, получившая название x86, является практически стандартом на рынке микропроцессоров.

Данная архитектура характеризуется:

- большим числом команд (более 200);
- переменной длиной команд (от 1 до 13 байт);
- значительным числом способов адресации и форматов команд;
- наличием сложных команд и многотактностью их выполнения;
- наличием микропрограммного управления для сложных команд.

На мировых рынках полная система команд x86 представлена в процессорах фирм Intel, AMD, VIA Technologies и др.

Компьютеры с RISC (Reduced Instruction Set Computer)-архи-тектурой содержат набор простых, часто употребляемых в программах команд. Основными являются операции типа «регистр – регистр».

Понятие RISC в современном его понимании оформилось на базе трех исследовательских проектов компьютеров: процессора 801 компании IBM, процессора RISC университета Беркли и процессора MIPS Стенфордского университета. Простота архитектуры и ее эффективность, подтвержденная этими проектами, вызвали большой интерес в компьютерной индустрии, и с 1986 г. началась активная промышленная реализация архитектуры RISC.

Отличительные черты данной архитектуры:

- сокращенное число команд;
- большинство команд выполняется за один машинный такт;
- постоянная длина команд;
- небольшое количество способов адресации и форматов команд;
- для простых команд нет необходимости в использовании микропрограммного управления;
- большое число регистров внутренней памяти процессора.

Исходя из перечисленных характеристик, компьютеры с RISC-архитектурой «обязаны» иметь преимущество в производительности по сравнению с CISC-компьютерами.

В настоящее время основными разрабочиками RISC-процессоров являются корпорации Sun/Oracle (Ultra Sparc T1, T2), IBM (POWER 6, 6+, 7, Cell). Эти процессоры используются в высокопроизводительных компьютерах (рабочих станциях, серверах, суперкомпьютерах).

Для мобильных устройств (карманных ПК, смартфонов, коммуникаторов) наибольшее распространение получили RISC-процессоры семейства ARM (корпорация ARM Ltd, Великобритания).

Уступая во многом RISC, процессоры с системой команд x86 сохранили лидерство на рынке персональных систем за счет постоянной модернизации системы команд, нацеленной на увеличение производительности процессоров, а также за счет того, что программное обеспечение, разработанное для x86-компьютеров, начиная с 1980 г., способно функционировать и на современных компьютерах с этой архитектурой.

VLIW-архитектура связана с кардинальной перестройкой всего процесса трансляции и исполнения программ. Уже на этапе подготовки программы компилятор группирует несвязанные операции в пакеты, содержимое которых

строго соответствует структуре процессора. Сформированные пакеты операций преобразуются компилятором в командные слова, которые по сравнению с обычными инструкциями выглядят очень большими. Отсюда и название этих суперкоманд, и соответствующей им архитектуры – VLIW (Very Long Instruction Word – очень длинное командное слово). По идее, затраты на формирование суперкоманд должны окупаться скоростью их выполнения и простотой аппаратуры процессора, с которого снята вся «интеллектуальная» работа по поиску параллелизма несвязанных операций.

Компилятор VLIW, в отличие от суперскалярной обработки, производит статический анализ программы и создает точный план того, как процессор будет выполнять программу: указывается, когда будет выполнена каждая операция, какие функциональные устройства будут работать и какие регистры будут содержать операнды.

Компилятор VLIW передает план вычисления аппаратному обеспечению, которое, в свою очередь, выполняет указанный план. Этот план позволяет VLIW использовать относительно простое аппаратное обеспечение, способное добиться высокого уровня параллелизма на уровне команд.

Однако даже при небольшом изменении начальных данных путь выполнения программы сколь угодно сильно изменяется.

VLIW-архитектура в свое время использовалась в RISC-процессорах семейств PA-8000, 9000 корпорации HP (Hewlett Packard).

Тенденции, заложенные в P6, получили развитие в концепции EPIC. Концепция EPIC (Explicity Parallel Instruction Computing – вычисления с явным параллелизмом команд, где «явным» означает явно указанным при трансляции) разработана совместно фирмами Intel и Hewlett Packard и имеет ту же значимость, что и CISC- и RISC-архитектуры.

Концепция реализации параллелизма на уровне команд (EPIC) определяет новый тип архитектуры, способной конкурировать по масштабам влияния с RISC. Эта идеология направлена на то, чтобы упростить аппаратное обеспечение и, в то же время, извлечь как можно больше «скрытого параллелизма» на уровне команд, чем это можно сделать при реализации VLIW и суперскалярных стратегий, используя большую ширину «выдачи» команд и длинные (глубокие) конвейеры.

Особенности EPIC:

− большое количество регистров (128 64-разрядных регистров общего назначения);

− использование простых инструкций, сгруппированных по три, одинаковой длины, образующих длинные командные слова LIW (long instruction words);

− переупорядочиванием и оптимизацией команд, так же как и во VLIW, занимается компилятор, а не процессор;

− команды из разных ветвей узлового ветвления снабжаются предикатными полями (полями условий) и запускаются параллельно;

− выборка данных по предположению (выборка данных до того, как они потребуются, т. е. заранее);

− масштабируемость архитектуры до большого количества функциональных устройств.

EPIC-технология с явным заимствованием лучших идей из CISC- и RISC-архитектур использована в 64-разрядной интеловской архитектуре (IA-64) процессоров Itanium, Itanium2. В IA-64 используется новый набор команд, разработанный Intel и HP.

Как было сказано выше, архитектура SIMD связана с параллельной обработкой потоков данных. Существуют несколько способов реализации этой архитектуры: матричная структура процессора, векторно-конвейерная, технология MMX и потоковые SIMD-расширения.

Суть матричной структуры заключается в том, что имеется множество процессорных элементов, исполняющих одну и ту же команду над различными элементами матрицы, объединенных коммутатором. Основная проблема заключается в программировании обмена данными между процессорными элементами через коммутатор.

В отличие от матричной векторно-конвейерная структура процессора содержит конвейер операций, на котором обрабатываются параллельно элементы векторов и полученные результаты последовательно записываются в единую память. При этом отпадает необходимость в коммутаторе процессорных элементов, служащем камнем преткновения в матричных компьютерах.

Общим для всех векторных компьютеров является наличие в системе команд векторных операций, допускающих работу с векторами определенной длины. В таких компьютерах операции с векторами обычно выполняются над содержимым векторных регистров.

По назначению ЭВМ можно разделить на три группы: универсальные (общего назначения), проблемно-ориентированные и специализированные.

Универсальные ЭВМ предназначены для решения самых различных видов задач: научных, инженерно-технических, экономических, информационных, управленческих и др. В качестве универсальных ЭВМ используются различные типы компьютеров, начиная от суперЭВМ и кончая персональными ЭВМ. Причем одни универсальные ЭВМ могут работать в многопользовательском режиме (в вычислительных центрах коллективного пользования, в локальных компьютерных сетях и т. д.), другие – в однопользовательском режиме.

Проблемно-ориентированные ЭВМ служат для решения более узкого круга задач, связанных, как правило, с управлением технологическими объектами, автоматизированным проектированием, разведкой и добычей нефти, банковским делом, издательской деятельностью и т. д.

Специализированные ЭВМ используются для решения еще более узкого круга задач или реализации строго определенной группы функций. Такая узкая ориентация ЭВМ позволяет четко специализировать их структуру, во мно-

гих случаях существенно снизить их сложность и стоимость при сохранении высокой производительности и надежности их работы.

По функциональным возможностям и размерам ЭВМ можно разделить на суперЭВМ, большие и микроЭВМ.

Функциональные возможности ЭВМ обусловливаются основными технико-эксплуатационными характеристиками.

Исторически первыми появились большие ЭВМ, элементная база которых прошла путь от электронных ламп до интегральных схем со сверхвысокой степенью интеграции.

Большие ЭВМ за рубежом часто называют мэйнфреймами (Mainframe). Мэйнфрейм – это высокопроизводительная вычислительная система с большим объемом оперативной и внешней памяти, поддерживающая многопользовательский и многозадачный режимы работы.

Особенности и характеристики современных мэйнфреймов:

1. Высокая надежность (среднее время наработки на отказ оценивается в 12–15 лет) – результат почти 50-летнего совершенствования мэйнфреймов.

2. Повышенная устойчивость систем. Мэйнфреймы могут обнаруживать, исправлять и изолировать большинство аппаратных и программных ошибок.

3. Целостность данных. В мэйнфреймах используется память с исправлением ошибок.

4. Рабочая нагрузка мэйнфреймов может составлять 80–95 % от их пиковой производительности.

5. Высокая пропускная способность подсистемы ввода/вывода (канальная архитектура).

6. Масштабирование может быть как вертикальным, так и горизонтальным. Вертикальное масштабирование обеспечивается наращиванием до 64 центральных процессоров в одном компьютере. Горизонтальное – реализуется объединением компьютеров в многомашинный (до 32 машин) кластер, выглядящий, с точки зрения пользователя, единым компьютером.

7. Доступ к данным. При централизованной обработке информации данные хранятся на одном компьютере, прикладные программы не нуждаются в сборе исходной информации из множества источников (как при распределенной обработке), не требуется дополнительное дисковое пространство для их временного хранения, не возникают сомнения в их актуальности.

8. Защита. Встроенные аппаратные и программные средства защиты, такие как криптографические устройства, программные продукты защиты операционных систем, обеспечивают совершенную защиту информации.

9. Непрекращающаяся совместимость – до сих пор в мэйнфреймах используются приложения, написанные в 70-е гг.

Централизованная архитектура остается востребованной, несмотря на преобладание в современном крупном бизнесе распределенных вычислительных систем. Сторонники мэйнфреймов утверждают, что такая архитектура обеспечивает нормальное функционирование системы при 100 %-й нагрузке

процессоров, тогда как производительность стандартных серверов ощутимо снижается уже при 65%-й нагрузке.

За долгие годы существования мэйнфреймов для них было разработано великое множество прикладного программного обеспечения, однако лучшим доказательством заинтересованности рынка является разработка и выпуск новых моделей этого класса.

До сегодняшнего дня бесспорным лидером в производстве мэйнфреймов является корпорация IBM, начиная от серии System/360, затем 370, 390 и до серии z Series. Мэйнфрейм System z10 поддерживает операционные системы z/OS, z/OSe, z/VM, z/VSE, Linux и может обслуживать от сотен до миллионов пользователей в зависимости от приложений.

Основными направлениями эффективного применения мэйнфреймов являются пакетная обработка заданий (когда компьютер выполняет работу без участия человека) и обработка заданий в реальном времени (On-line), например транзакционные системы, такие как система приобретения железнодорожных билетов, система оплаты по кредитной карте и т.п.

СуперЭВМ – мощные, высокоскоростные вычислительные машины (системы) с производительностью от десятков триллионов (GFLOPS) до нескольких квадриллионов (PFLOPS) операций с плавающей запятой в секунду. СуперЭВМ выгодно отличаются от больших универсальных ЭВМ по быстродействию числовой обработки, а от специализированных машин, обладающих высоким быстродействием в сугубо ограниченных областях, – возможностью решения широкого класса задач с числовыми расчетами.

В настоящее время развитие суперкомпьютеров идет по следующим направлениям: векторно-конвейерные компьютеры, параллельные компьютеры с общей памятью, массивно-параллельные системы с распределенной памятью, кластерные системы.

В каких рыночных нишах востребованы суперкомпьютеры? Прежде всего это проектирование самолетов и ракет, создание лекарств, предсказание погоды и природных катаклизмов, повышение эффективности электростанций и надежности автомобилей (преимущественно путем моделирования их столкновений) и фундаментальные научные исследования.

МикроЭВМ по назначению можно разделить на серверы, рабочие станции, персональные компьютеры, встраиваемые и промышленные микроЭВМ.

В настоящее время уже редко встретишь офис или предприятие, в котором бы не использовалась компьютерная сеть, время разрозненных персональных компьютеров давно ушло.

Примером таких функций может быть хранение файлов и обеспечение доступа к ним пользователей (клиентов), маршрутизация потоков данных, управление печатью сетевого принтера, обработка писем электронной почты, рассылка факсов и т. д. Серверами также называются программы, выполняющие эти функции. Ниже под термином «сервер» будет пониматься в первую очередь аппаратное решение.

По функциональному назначению серверы можно подразделить на файл-серверы, серверы приложений (чаще всего используются для баз данных и поддержки документооборота), FTP-серверы (для удаленного доступа к данным через Internet), серверы внешних устройств (печати, сканирования, факсимильной связи) и Web-серверы.

В последние годы большой популярностью пользуется операционная система Linux с открытым исходным кодом, которая была создана в 1991 г. в качестве версии Unix-подобной системы для ПК. Настоящий успех пришел к системе при ее использовании не на настольных ПК, а на серверах. Этому способствует мощная поддержка «китов индустрии» (IBM, HP, Dell, Sun), вкладывающих в развитие бесплатной ОС огромные средства.

Для оценки производительности серверов используется тест TPC-C, разработанный фирмой Transaction Processing Perfomance Council (TPC), с помощью которого подсчитывается количество транзакций (взаимодействий между узлами компьютерной сети по принципу запрос – ответ) в минуту (tpm C) и удельная стоимость ($/tpm C) транзакций.

Новый класс устройств – миниатюрные домашние серверы – становится обыденным явлением не только дома, но и в офисах малых и мобильных компаний. Домашний сервер берет на себя организацию передачи различных типов информации внутри небольшой локальной компьютерной сети.

По конструктивному исполнению серверы могут быть башенными, стоечными и блейд-серверами («лезвиями»).

Персональные компьютеры (ПК) – это однопользовательские микро-ЭВМ, удовлетворяющие требованиям общедоступности и универсальности применения.

На сегодняшний день большинство настольных ПК базируется на x86 процессорах с основной операционной системой из семейства Windows. Другая платформа представлена довольно популярными на Западе компьютерами Macintosh фирмы Apple. На сегодняшний день формальным отличием является операционная система (MacOSX, имеющая Unix-подобное ядро).

По назначению ПК можно классифицировать: на бытовые, общего назначения, профессиональные и игровые.

Переносные ПК по своим конструктивным особенностям можно разделить: на ПК-блокноты (ноутбуки), планшетные ПК, ультрамобильные ПК и карманные устройства.

2. Функциональная и структурная организация ЭВМ

Существуют два взгляда на построение и функционирование ЭВМ. Первый – взгляд пользователя, не интересующегося технической реализацией ЭВМ и озабоченного только получением некоторого набора функций и услуг, обеспечивающих эффективное решение его задач; второй – разработчика ЭВМ, усилия которого направлены на рациональную техническую реализацию необходимых пользователю функций. С учетом этого обстоятельства и вводятся понятия «функциональная» и «структурная» организация компьютера.

Действительно, с точки зрения пользователя решение любой задачи на ЭВМ требует поэтапного выполнения некоторой последовательности действий: программирования, кодирования, ввода, обработки, документирования. На каждом из этих этапов учет запросов пользователя может потребовать расширения реализуемых ЭВМ функций и услуг, что решается при проектировании ЭВМ и входит в понятие функциональной организации ЭВМ.

В результате создается абстрактная модель ЭВМ, описывающая функциональные возможности машины и предоставляемые ею услуги. Функциональная организация ЭВМ в значительной степени определяется предъявляемыми к ней требованиями, уровнем подготовки потенциальных пользователей, типом решаемых ими задач, потребностями в развитии компьютера (по емкости памяти, разрядности, составу периферийных устройств и др.).

Предусматриваемые абстрактной моделью функции ЭВМ реализуются на основе реальных физических средств (устройств, блоков, узлов, элементов) в рамках определенной структуры. В общем случае под структурной организацией ЭВМ понимается некоторая физическая модель, устанавливающая состав, порядок и принципы взаимодействия основных функциональных частей машины (без излишних деталей их технической реализации).

Функциональная организация ЭВМ играет ведущую роль и в значительной степени определяет структурную организацию машины, хотя и не дает жестких ограничений на конечную техническую реализацию структурных элементов. Вместе с тем функции и структура любого элемента находятся в диалектической взаимосвязи и взаимозависимости. С одной стороны, функциональным назначением устройства (блока, узла) ЭВМ определяется необходимый состав материальных объектов (реальных аппаратных и программных средств) и характер их взаимодействия. С другой стороны, одна и та же функция может быть реализована на совершенно разных технических средствах, а изменение состава или связей между элементами, изменение пропорций между аппаратными и программными средствами может сохранить неизменной функцию системы, сообщив ей новые свойства.

Развитие архитектуры неизбежно ведет к развитию структуры ЭВМ. Реализация принципов интеллектуализации, которые все больше определяют развитие архитектуры ЭВМ, возможна при совершенствовании структурной организации, обеспечивающей повышение эффективности вычислительного

процесса и, как следствие этого, рост производительности ЭВМ. В конечном счёте условием и критерием развития структуры является рост производительности ЭВМ.

Обобщенная структура ЭВМ состоит из следующих составных частей:

- обрабатывающей подсистемы;
- подсистемы памяти;
- подсистемы ввода/вывода;
- подсистемы управления и обслуживания.

Развитие обрабатывающей подсистемы в большей степени, чем всех остальных подсистем, идет по пути разделения функций и повышения специализации составляющих ее устройств. Создаются специальные средства, которые осуществляют функции управления системой, освобождая от этих функций средства обработки. Такое распределение функций сокращает эффективное время обработки информации и повышает производительность ЭВМ. В то же время, средства управления, как и средства обработки, становятся более специализированными. Устройство управления памятью реализует эффективные методы передачи данных между средствами обработки и подсистемой памяти. Меняются функции центрального устройства управления.

Подсистема памяти современных компьютеров имеет иерархическую структуру, состоящую из нескольких уровней:

- сверхоперативный уровень (память процессора, кэш-память);
- оперативный уровень (оперативная память);
- внешний уровень (внешние ЗУ на дисках, лентах и т. д.).

В состав подсистемы ввода/вывода входит набор специализированных устройств, между которыми распределены функции ввода/вывода, что позволяет свести к минимуму потери производительности системы при операциях ввода/вывода. Эти устройства определяют пропускную способность подсистемы ввода/вывода.

Подсистема управления и обслуживания – это совокупность аппаратно-программных средств, предназначенных для обеспечения максимальной производительности, заданной надежности, ремонтопригодности, удобства настройки и эксплуатации. Она обеспечивает проблемную ориентацию и заданное время наработки на отказ, подготовку и накопление статистических сведений о загрузке и прохождении вычислительного процесса, выполняет функции «интеллектуального» интерфейса с различными категориями обслуживающего персонала, осуществляет инициализацию, тестирование и отладку. Подсистема управления и обслуживания позволяет поднять на качественно новый уровень эксплуатацию современных ЭВМ.

При разработке структуры ЭВМ все подсистемы должны быть сбалансированы между собой. Только оптимальное согласование быстродействия обрабатывающей подсистемы с объемами и скоростью передачи информации подсистемой памяти, с пропускной способностью подсистемы ввода/вывода позволяет добиться максимальной эффективности использования ЭВМ.

Важнейшими факторами, определяющими функциональную и структурную организацию ЭВМ, являются выбор системы и форматов команд, типов данных и способов адресации.

Основными типами данных в компьютерах интеловской архитектуры являются: байт, слово, двойное слово, квадрослово и 128-разрядное слово.

Каждый из представленных типов данных может начинаться с любого адреса: это означает, что слово не обязано начинаться с чётного адреса; двойное слово – с адреса, кратного 4 и т.д. Таким образом, достигается максимальная гибкость структур данных и эффективность использования памяти.

3. Функциональная и структурная организация центрального процессора ЭВМ

В области вычислительной техники различают процессоры центральные, графические, ввода/вывода, передачи данных, коммуникационные и специализированные.

Центральный процессор – основное устройство ЭВМ, которое наряду с обработкой данных выполняет функции управления системой: инициирование ввода/вывода, обработку системных событий, управление доступом к основной памяти и т. п.

Структурная организация центрального процессора (ЦП) определяется функционально-логической организацией, микроархитектурой и требованиями к технико-экономическим показателям.

Логическую структуру ЦП представляет ряд функциональных средств: средства обработки, средства управления системой и программой (центральное устройство управления), локальная память, буферная память (кэш-память L1, L2, …), средства инициализации ввода/вывода, средства управления памятью, системные средства.

Средства обработки обеспечивают выполнение операций над данными с фиксированной (целочисленные данные) и плавающей точкой, векторными данными, полями переменной длины и т. д. Локальная память состоит из регистров общего назначения, регистров данных с плавающей точкой, управляющих регистров и др. К средствам управления памятью относятся средства управления доступом к ОП и предвыборкой команд и данных. Буферная память включает в себя кэш-память команд и данных первого (L1), второго (L2), а в некоторых процессорах и третьего (L3) уровней. Средства инициализации ввода/вывода обеспечивают активизацию контроллеров (каналов) периферийных устройств. К системным средствам относятся средства службы времени: часы астрономического времени, таймер, коммутатор и т. д.

Существует обязательный (стандартный) минимальный набор функциональных средств для каждого типа центрального процессора. Он включает в себя регистры общего назначения, средства выполнения стандартного набора операций и средства управления вычислительным процессом. Конкретная

реализация ЦП может различаться составом средств, способом их реализации, техническими параметрами.

Центральное устройство управления (ЦУУ) – это комплекс средств автоматического управления процессом передачи и обработки информации. ЦУУ вырабатывает управляющие сигналы (УС), необходимые для выполнения всех операций, предусмотренных системой команд, а также координирует работу всех узлов и блоков ЭВМ. В связи с этим можно считать ЦУУ преобразователем первичной командной информации, представленной программой решения задачи, во вторичную командную информацию, представляемую управляющими сигналами.

В общем случае ЦУУ формирует управляющие сигналы для реализации следующих функций:

- выборки из памяти кода очередной команды;
- расшифровки кода операции и признаков выбранной команды;
- выборки операндов и выполнения машинной операции;
- обеспечения прерываний при выполнении команд;
- формирования адреса следующей команды;
- учета состояний других устройств машины;
- инициализации работы контроллеров (каналов) ввода/вывода;
- организации контроля работоспособности ЭВМ.

По общей организации управление может быть центральным, распределенным и смешанным. В первом случае в блоке управления ЦУУ вырабатываются все УС для всех команд, выполняемых процессором ЭВМ. Во втором случае операционные и другие устройства процессора имеют собственные блоки местного управления. В последнем случае ЦУУ вырабатывает сигналы для запуска в работу блоков местного управления.

По способу синхронизации работы различают ЦУУ:

- синхронного типа, в которых время цикла может быть постоянным или переменным;
- асинхронного типа, в которых продолжительность цикла определяется фактическими затратами времени на выполнение каждой операции, в этом случае необходимо вырабатывать сигналы об окончании операции;
- смешанного типа, где частично реализуются оба предыдущих принципа организации циклов.

По принципу формирования и развертывания временной последовательности УС различают ЦУУ:

- аппаратного (схемного) типа;
- микропрограммного типа.

Микропрограммный принцип управления обеспечивает реализацию одной сложной машинной команды путем выполнения определенной микропрограммы, интерпретирующей алгоритм выполнения данной операции. Совокупность микропрограмм, необходимая для реализации сложных команд ЭВМ, хранится в специальной памяти микропрограмм. Каждая микропро-

грамма состоит из определенной последовательности микрокоманд, которые после выборки из памяти преобразуются в набор управляющих.

Анализ аппаратурной (схемной) и микропрограммной реализации устройства управления указывает на зависимость стоимости управления от сложности выполняемых команд. Для простых команд выгодно использовать схемное управление, а для сложных команд – микропрограммное. Однако последнее приводит к увеличению затрат времени на выработку управляющих воздействий. Основным же преимуществом микропрограммного управления является его гибкость, которая позволяет повышать эффективность серийно выпускаемых и эксплуатируемых машин за счет введения новых средств математического обеспечения, использующих дополнительный набор команд и новые функции процессора.

4. Принципы организации подсистемы памяти ЭВМ и ВС

Памятью ЭВМ называется совокупность устройств, предназначенных для запоминания, хранения и выдачи информации.

Основными характеристиками отдельных устройств памяти (запоминающих устройств) являются емкость памяти, быстродействие и стоимость хранения единицы информации (бита).

Емкость памяти определяется максимальным количеством данных, которые могут в ней храниться. Емкость измеряется в двоичных единицах (битах), машинных словах, но большей частью в байтах (1 байт = 8 бит). Часто емкость памяти выражают через число К = 210 = 1024, например, 1024 бит = Кбит (килобит), 1024 байт = Кбайт (килобайт), 1024 Кбайт = 1 Мбайт (мегабайт), 1024 Мбайт = 1 Гбайт (гигабайт), 1024 Гбайт = 1 Тбайт (терабайт).

Быстродействие (задержка) памяти определяется временем доступа и длительностью цикла памяти. Время доступа представляет собой промежуток времени между выдачей запроса на чтение и моментом поступления запрошенного слова из памяти. Длительность цикла памяти определяется минимальным временем между двумя последовательными обращениями к памяти.

Требования к увеличению емкости и быстродействия памяти, а также к снижению ее стоимости являются противоречивыми. Чем больше быстродействие, тем технически труднее достигается и дороже обходится увеличение емкости памяти. Исходя из этого, память ЭВМ организуется в виде иерархической структуры запоминающих устройств, обладающих различным быстродействием, емкостью и стоимостью. Причем более высокий уровень меньше по объему, быстрее и имеет большую стоимость в пересчёте на байт, чем более низкий уровень. Уровни иерархии взаимосвязаны: все данные на одном уровне могут быть также найдены на низком уровне, и все данные на этом (более низком) уровне могут быть найдены на следующем, нижележащем уровне, и так далее, пока мы не достигнем основания иерархии.

В структуре памяти к верхнему (сверхоперативному) уровню относятся: управляющая память, регистры различного назначения, стек регистров, буферная память. На втором уровне находится основная (или оперативная) память. На последующих уровнях размещается внешняя и архивная память. Система управления памятью обеспечивает обмен информационными блоками между уровнями, причем обычно первое обращение к блоку информации вызывает его перемещение с низкого (медленного) уровня на более высокий. Это позволяет при последующих обращениях к данному блоку осуществлять его выборку с более быстродействующего уровня памяти.

Успешное или неуспешное обращение к более высокому уровню называется соответственно «попаданием» (hit) или «промахом» (miss). Попадание есть обращение к объекту в памяти, который найден на более высоком уровне, в то время как промах означает, что он не найден на этом уровне. Доля попаданий, или коэффициент попаданий, есть доля обращений, найденных на более высоком уровне. Иногда она представляется в процентах. Аналогично для промахов.

Сравнительно небольшая емкость оперативной памяти компенсируется практически неограниченной емкостью внешних запоминающих устройств. Однако эти устройства работают намного медленнее, чем оперативная память. Время обращения за данными для магнитных дисков составляет десятки микросекунд. Для сравнения: цикл обращения к оперативной памяти (ОП) составляет несколько десятков наносекунд. Исходя из этого, вычислительный процесс должен протекать с возможно меньшим числом обращений к внешней памяти. Память современных компьютеров реализуется на микросхемах статических и динамических запоминающих устройств с произвольной выборкой.

Непрерывный рост производительности ЭВМ проявляется в первую очередь в повышении скорости работы процессора. Быстродействие ОП также растет, но все время отстает от быстродействия аппаратных средств процессора в значительной степени потому, что одновременно происходит опережающий рост её емкости, что делает более трудным уменьшение времени цикла работы памяти. Вследствие этого быстродействие ОП часто оказывается недостаточным для обеспечения требуемой производительности ЭВМ. Это проявляется в несоответствии пропускных способностей процессора и ОП. Возникающая проблема выравнивания их пропускных способностей решается путем использования сверхоперативной буферной памяти небольшой емкости и повышенного быстродействия, хранящей команды и данные, относящиеся к обрабатываемому участку программы.

Буферная память не является программно доступной. Это значит, что она влияет только на производительность ЭВМ, но не должна оказывать влияния на программирование прикладных задач. Поэтому она получила название кэш-памяти (в переводе с английского – тайник). В современных компьютерах применяют многоуровневую кэш-память (до трех уровней), которая еще больше способствует производительности ЭВМ. Как правило, на первом

уровне используются раздельные кэш-памяти для команд и данных, а на других уровнях данные и команды хранятся в одних и тех же кэш-памятях.

Стек регистров, реализующий безадресное задание операндов, является эффективным элементом архитектуры ЭВМ. Стек представляет собой группу последовательно пронумерованных регистров, снабжённых указателем стека, в котором автоматически при записи устанавливается номер первого свободного регистра стека (вершина стека). Существует два основных способа организации стека регистров:

- LIFO (Last-in First-Out) – последний пришёл – первый ушёл;
- FIFO (First-in First-Out) – первый пришёл – первый ушёл.

В современных архитектурах процессоров стек и стековая адресация широко используются при организации переходов к подпрограммам и возврата из них, а также в системах прерывания.

В функциональном отношении кэш-память рассматривается как буферное ЗУ, размещённое между основной (оперативной) памятью и процессором. Основное назначение кэш-памяти – кратковременное хранение и выдача активной информации процессору, что сокращает число обращений к основной памяти, скорость работы которой меньше, чем кэш-памяти.

За единицу информации при обмене между основной памятью и кэш-памятью принята строка (линейка), причём под строкой понимается набор слов, выбираемый из оперативной памяти при одном к ней обращении. Хранимая в оперативной памяти информация представляется, таким образом, совокупностью строк с последовательными адресами. В любой момент времени строки в кэш-памяти представляют собой копии строк из некоторого их набора в ОП, однако расположены они необязательно в такой же последовательности, как в ОП.

Построение кэш-памяти может осуществляться по различным принципам, которые будут рассмотрены ниже. Но общим для всех способов построения кэш-памяти является использование так называемых адресных тегов. Адресный тег – это расширенный адрес, который объединяет адреса всех слов, принадлежащих данной строке. Он указывает, какую строку в ОП представляет данная строка в кэш-памяти.

В режиме записи информации в память адрес обрабатывается также, как и при чтении. Само же слово информации из ЦП проходит через демультиплексор и заносится в регистр строк. Далее, в зависимости от выбранного способа записи оно может загрузиться в накопитель строк кэш-памяти и в ОП или только в кэш-память.

Для высвобождения места в кэш-памяти с целью записи выбираемой из ОП строки одна из строк удаляется. Определение удаляемой строки производится посредством блока замены строк, в котором хранится информация, необходимая для реализации принятой стратегии обновления находящихся в накопителе строк.

Существует три основных способа размещения данных в кэш-памяти: прямое распределение (отображение), полностью ассоциативное распределение и частично ассоциативное распределение. Ниже подробно описан каждый способ размещения и его механизм преобразования адресов. Для того чтобы конкретизировать описание, положим, что кэш-память может содержать 128 строк, размер строки – 16 слов, а основная память может содержать 16384 строк. Для адресации основной памяти используется 18 бит.

Оперативная (основная) память представляет собой следующий уровень иерархии памяти. Оперативная память удовлетворяет запросы кэш-памяти и устройств ввода/вывода. Она является местом назначения для ввода и источником для вывода. Для оценки производительности (быстродействия) основной памяти используются следующие параметры: время доступа, длительность цикла памяти, латентность и пропускная способность.

Как было сказано выше, время доступа – это время, проходящее с момента обращения к памяти до момента считывания данных. Данная величина приблизительно одинакова для всех типов динамической памяти и составляет примерно 50 нс. Время доступа актуально при случайном доступе к памяти, т. е. когда последовательно считываемые ячейки памяти принадлежат различным строкам матрицы памяти.

Самым простым способом управления оперативной памятью является разделение её на несколько разделов (сегментов) фиксированной величины (статическое распределение). Это может быть выполнено вручную оператором во время старта системы или во время её генерации. Очередная задача, поступающая на выполнение, помещается либо в общую очередь, либо в очередь к некоторому разделу. Подсистема управления памятью в этом случае выполняет следующие задачи: сравнивает размер программы, поступившей на выполнение, и свободных разделов памяти; выбирает подходящий раздел; осуществляет загрузку программы и настройку адресов.

Общепринятая в настоящее время концепция виртуальной памяти появилась достаточно давно. Она позволила решить целый ряд актуальных вопросов организации вычислений. Прежде всего к числу таких вопросов относится обеспечение надежного функционирования мультипрограммных систем. В любой момент времени компьютер выполняет множество процессов (или задач), каждый из которых располагает своим адресным пространством. Было бы слишком накладно отдавать всю физическую оперативную память какой-то одной задаче, тем более, что многие задачи реально используют только небольшую часть своего адресного пространства. Поэтому необходим механизм разделения небольшой физической памяти между различными задачами. Виртуальная память является одним из способов реализации такой возможности. Она делит физическую память на блоки и распределяет их между различными задачами, при этом она предусматривает также некоторую схему защиты, которая ограничивает задачу теми блоками, которые ей принадлежат. Большинство типов виртуальной памяти сокращают также время начального запуска

программы на процессоре, поскольку не весь программный код и данные требуются ей в физической памяти, чтобы начать выполнение.

Другой вопрос, тесно связанный с реализацией концепции виртуальной памяти, касается организации вычислений на компьютере задач очень большого объёма. Раньше, если программа становилась слишком большой для физической оперативной памяти, часть её необходимо было хранить во внешней памяти (на диске) и задача приспособить её для решения на компьютере ложилась на программиста. Программисты делили программы на части и затем определяли те из них, которые можно было бы выполнять независимо, организуя оверлейные структуры, которые загружались в основную память и выгружались из неё под управлением программы пользователя. Программист должен был следить за тем, чтобы программа не обращалась вне отведённого ей пространства физической памяти. Виртуальная память освободила программистов от этого бремени.

Виртуальным называется такой ресурс, который для пользователя (пользовательской программы) представляется обладающим свойствами, которыми он в действительности не обладает. Так, например, пользователю может быть предоставлена виртуальная оперативная память, размер которой превосходит всю имеющуюся в системе реальную ОП. Пользователь пишет программы так, как будто в его распоряжении имеется однородная (одноуровневая) оперативная память большого объёма, но в действительности все данные, используемые программой, хранятся на нескольких разнородных запоминающих устройствах, обычно в ОП и на дисках, и при необходимости частями перемещаются между ними. Все эти действия выполняются автоматически, без участия программиста, т. е. механизм виртуальной памяти является прозрачным по отношению к пользователю.

Наиболее распространенными реализациями виртуальной памяти являются страничное, сегментное и странично-сегментное распределение памяти.

5. Организация системного интерфейса и ввода-вывода информации

Связь устройств ЭВМ друг с другом осуществляется с помощью интерфейсов.

Интерфейс представляет собой совокупность линий и шин, сигналов, электронных схем и алгоритмов (протоколов), предназначенных для осуществления обмена информацией между устройствами.

Производительность и эффективность использования компьютера определяются не только возможностями его процессора и пропускной способностью основной памяти, но в очень большой степени характеристиками интерфейсов, составом периферийных устройств (ПУ), их техническими данными.

Объединение отдельных подсистем (устройств, модулей) ЭВМ в единую систему основывается на многоуровневом принципе с унифицированным сопряжением между всеми уровнями – стандартными интерфейсами. Под стан-

дартными интерфейсами понимают такие интерфейсы, которые приняты и рекомендованы в качестве обязательных отраслевыми или государственными стандартами, различными международными комиссиями, а также крупными зарубежными фирмами.

Интерфейсы характеризуются следующими параметрами:

− пропускной способностью интерфейса – количеством информации, которое может быть передано через интерфейс в единицу времени;

− максимальной частотой передачи информационных сигналов через интерфейс;

− информационной шириной интерфейса – числом бит или байт данных, передаваемых параллельно через интерфейс;

− максимально допустимым расстоянием между соединяемыми устройствами;

− динамическими параметрами интерфейса – временем передачи отдельного слова или блока данных с учётом продолжительности процедур подготовки и завершения передачи;

− общим числом проводов (линий) в интерфейсе.

Можно выделить следующие четыре классификационных признака интерфейсов:

− способ соединения компонентов системы (радиальный, магистральный, смешанный);

− способ передачи информации (параллельный, последовательный, параллельно-последовательный);

− принцип обмена информацией (асинхронный, синхронный);

− режим передачи информации (двусторонняя поочередная передача, односторонняя передача).

Классификация интерфейсов по назначению содержит следующие уровни сопряжений:

− системные интерфейсы;

− локальные интерфейсы;

− интерфейсы периферийных устройств (малые интерфейсы);

− межмашинные интерфейсы.

В подсистеме ввода/вывода ЭВМ используются три основных способа организации передачи данных между памятью и ПУ: программно-управляемая передача, передача по запросу прерывания от ПУ и прямой доступ к памяти (ПДП).

Программно-управляемая передача данных осуществляется при непосредственном участии и под управлением процессора, который при этом выполняет специальную подпрограмму ввода/вывода. Операция ввода/вывода инициируется центральным процессором, т. е. текущей командой программы. Данный способ является простым в реализации, но при обработке команды ввода/вывода ЦП бесполезно тратит время, ожидая готовности ПУ. Это значительно снижает производительность ЭВМ.

Второй способ передачи данных по запросу прерывания от ПУ реализуется под управлением контроллера прерываний (КПР) и позволяет организовывать более гибкое взаимодействие между ЦП и ПУ. Предположим, что в качестве ПУ используется клавиатура, предназначенная для ввода в ЭВМ команд, инструкций и данных. Каждый раз, когда пользователь (оператор) нажимает клавишу, ПУ выдает в КПР запрос на прерывание, который, в свою очередь, вырабатывает для ЦП сигнал прерывания. ЦП по этому сигналу приостанавливает работу текущей программы и передает управление подпрограмме ввода/вывода. Подпрограмма обрабатывает запрос и по её завершении ЦП возвращается к работе по текущей программе. Выполнение текущей программы продолжается до следующего нажатия клавиши, и далее процесс повторяется. В этом случае преимущество от использования прерывания очевидно (принципы работы системы прерывания программ описаны в разд. 2.6).

При программно-управляемой передаче данных ЦП на всё время этой передачи отвлекается от выполнения основной программы. Операция пересылки данных логически слишком проста, чтобы эффективно загружать логически сложную быстродействующую аппаратуру процессора. Вместе с тем при пересылке блока данных ЦП приходится для каждой единицы передаваемых данных (байт, слово) выполнять довольно много инструкций, чтобы обеспечить буферизацию данных, преобразование форматов, подсчёт количества переданных данных, формирование адресов в памяти и т. п. В результате скорость передачи данных при пересылке блока данных под управлением процессора оказывается недостаточной. Поэтому для быстрого ввода/вывода блоков данных и разгрузки ЦП от управления операциями ввода/вывода используют прямой доступ к памяти.

Прямой доступ к памяти (DMA – Direct Memory Access) – это такой способ обмена данными, который обеспечивает автономно от ЦП установление связи и передачу данных между ОП и ПУ. Прямой доступ к памяти освобождает процессор от управления операциями ввода/вывода, позволяет осуществлять параллельно во времени выполнение процессором программы с обменом данными между ОП и ПУ, производить этот обмен со скоростью, ограничиваемой только пропускной способностью ОП или ПУ.

Заключение

В течение всего периода эволюции компьютерных систем прослеживается тенденция к повышению скорости обработки информации процессором, уменьшение физических размеров компонентов, росту объема памяти и повышению пропускной способности каналов ввода-вывода.

Не отрицая того факта, что одной из причин повышения производительности процессоров явился прогресс в области микроэлектроники, в частности миниатюризация электронных компонентов, все же отметим, что не меньшее, если не большее, влияние на этот процесс, особенно в последние годы, оказа-

ли новые идеи в отношении структурной организации процессора, в частности широкое использование принципов конвейерной и параллельной обработки и внедрение технологии предпочтительного выбора направления ветвления программы, т. е. выполнение условных переходов на основании прогнозных оценок еще до формирования условий перехода. Все эти идеи преследуют одну цель – максимально сократить время простоя процессора.