**Архитектура ЭВМ БЭСМ-6**

Из книги Л.Н.Королева "Структуры ЭВМ и их математическое обеспечение"

Мы выделили описание машины БЭСМ-6 в отдельный раздел по той причине, что эта машина занимает особое место в развитии отечественной вычислительной техники. Принципы, заложенные в основу ее структурной организации, не потеряли своего значения до сих пор. Машины БЭСМ-6 и вычислительные комплексы, созданные на их основе, до сих пор интенсивно используются в научно-исследовательских институтах, центрах обработки информации, решающих наиболее важные для народного хозяйства, науки и обороны задачи. За время более 10-летнего использования этой машины накоплено огромное по составу, чрезвычайно важное по значимости математическое обеспечение, сконцентрировавшее в себя лучшие достижения советских ученых в области вычислительной математики, программирования, решения задач проектирования сложнейших объектов, в области решения задач, связанных с освоением космического пространства.

Разработка машины БЭСМ-6, главный конструктор которой академик С.А.Лебедев и зам. главного конструктора В. А. Мельников, была закончена в конце 1966 г. Машина вступила в строй в 1967 г.

БЭСМ-6 обладает рядом интересных особенностей по организации виртуальной (математической) памяти, по принятому в ее структуре принципу "водопровода", по организации защиты памяти, прерываний, по организации связи с внешними устройствами и каналами.

Машина БЭСМ-6 - быстродействующая машина, выполняющая около 1 млн. одноадресных операций в секунду. Она выполнена на полупроводниках, на элементной базе, допускающей высокую частоту переключений (основная тактовая частота - 10 Мгц). По своим структурным характеристикам и архитектуре машина БЭСМ-6 вполне может быть отнесена к машинам 3-го поколения, хотя она и выполнена не на интегральных схемах, а на "навесных" деталях, т. е. на технологической основе машин второго поколения.

Основная цель, которую преследовали авторы проекта машины БЭСМ-6, была такова: создать быстродействующую серийную машину, сравнительно дешевую, удовлетворяющую наиболее важным современным требованиям с точки зрения автоматизации программирования и развития операционных систем, оснащенную имевшимися в то время в отечественном серийном производстве внешними запоминающими устройствами и устройствами ввода-вывода. Машина предназначалась для использования в крупных вычислительных центрах для решения научных и экономических задач, требующих большого объема вычислений.

Как показало время, эта цель была достигнута. Машина БЭСМ-6 производится серийно, и круг заказчиков этой машины продолжает расти. Сферами ее применения оказались научно-исследовательские институты, университеты, крупные конструкторские бюро.

Основные технические характеристики машины БЭСМ-6 таковы:

Быстродействие - около 1 млн. операций/сек.;

объем ОЗУ - от 32 до 128 тысяч машинных слов;

время выполнения сложения с плавающей запятой - 1,1 мксек;

время умножения - 1,9 мксек;

время деления - 4,9 мксек;

время выполнения логических поразрядных операций - 0,5 мксек.

Работа арифметического устройства совмещена с выборкой операндов из памяти.

Разрядность машинного слова - 48 двоичных разрядов.

Объем промежуточной памяти на магнитных барабанах - 512 тысяч слов.

К центральному процессору могут быть подключены 32 лентопротяжных механизма, каждый емкостью до 1 млн. слов. В состав вводных-выводных устройств входят два алфавитно-цифровых печатающих устройства (400 строк в минуту), два устройства вывода на перфокарты (ПИ-80), четыре устройства вывода на перфоленту, четыре устройства ввода с перфоленты, два устройства ввода с перфокарт (ВУ-700 или УВвК-601), 24 телетайпа.

К БЭСМ-6 возможно подключение дисков и графопостроителей, однако в комплектацию машин, поступивших с завода-изготовителя до 1970 г., это важное оборудование не входило. В комплектацию серийных машин БЭСМ-6 диски были включены лишь в 1972 г.

**Структурные особенности машины БЭСМ-6**

Для того чтобы достичь необходимого баланса между высокой скоростью выполнения арифметических и логических действий в центральном процессоре и ограниченным быстродействием блоков оперативного ферритового запоминающего устройства (время цикла работы каждого блока - 2 мксек), были предприняты следующие меры в структурной организации. Оперативное запоминающее устройство состоит из восьми блоков, допускающих одновременную выборку информации (командных слов и операндов), что резко повышает эффективное быстродействие системы памяти. Подряд идущие физические адреса памяти относятся к разным блокам, и если оказалось, например, так, что последовательно выбираемые операнды имеют последовательно возрастающие (убывающие) адреса, то они могут выбираться со средней скоростью, равной 2 мксек/8=0,25 мксек.

В реальных программах, конечно, далеко не всегда производится последовательная выборка операндов, и на самом деле эффективное быстродействие памяти не достигает указанных границ. В предположении, что запросы к восьми блокам памяти идут в хаотическом порядке (равновероятное распределение номеров запросов), эффективное быстродействие оценивается формулой

,

где Т - время цикла одного блока памяти и n - число параллельно работающих блоков.

Приведенные выше формулы показывают значение метода "запараллеливания" блоков памяти для получения выигрыша в эффективном быстродействии.

В реальных программах законы выборки адресов весьма сложны, и для них трудно подобрать соответствующий закон распределения. Поэтому при конструировании БЭСМ-6 была проделана большая работа по моделированию структуры машины и анализу временных параметров работы отдельных ее устройств на потоках информации, генерируемых реальными программами, и в частности, на модели считались реальные задержки при обращении к различному количеству параллельно работающих блоков памяти.

Второй структурной особенностью организации обращений к оперативному запоминающему устройству является метод буфферизации, или метод накопления очереди заказов к системе памяти. В машине БЭСМ-6 существуют группы регистров, на которых хранятся запросы (адреса), называемые буферами адресов слов и команд. Разумеется, что эти буфера могут работать эффективно только в том случае, если структура машины позволяет просматривать команды "вперед", т. е. загодя готовить запросы. Устройство управления БЭСМ-6 позволяет это делать. Буфера адресов позволяют в конечном итоге сгладить неравномерность поступления запросов к памяти и тем самым повысить эффективность ее использования.

Третьей структурной особенностью БЭСМ-6 является метод использования сверхоперативной, неадресуемой из программы памяти небольшого объема, цель которого≈автоматическая экономия обращений к основному оперативному запоминающему устройству. Эта сверхоперативная память управляется таким образом, что часто используемые операнды и небольшие внутренние командные циклы оказываются на быстрых регистрах и готовы к немедленному использованию в арифметическом устройстве или в системе управления машиной. Быстрые регистры в ряде случаев позволяют экономить до 60% всех обращений к памяти и уменьшают тем самым временные затраты на ожидание чисел и команд из основной памяти.

Следует еще раз подчеркнуть, что об использовании быстрых регистров заботится аппаратура самой машины и при составлении программ об экономии обращений к памяти думать нет необходимости.

Эти структурные особенности БЭСМ-6 получили название водопроводного принципа построения структуры машины. В самом деле, если подсчитать время от начала выполнения команды до его окончания, то для каждой команды оно будет очень велико, однако глубокий параллелизм выполнения, просмотр вперед, наличие буфера адресов, быстрых регистров приводят к тому, что "поток" команд и темп обработки информации очень высок. Аналогия с водопроводом состоит в том, что если проследить время, за которое частица воды проходит по некоторому участку водопровода, то оно будет большим, хотя скорость на выходе потока может быть очень велика. Четвертой структурной особенностью БЭСМ-6, имеющей очень важное значение для построения операционных систем и работы машины в мультипрограммном режиме, является принятый аппаратный способ преобразования математических, или виртуальных адресов в физические адреса машины. В машине БЭСМ-6 четко выдержано деление на физическую и математическую память, принята постраничная организация, однако способ отображения, заложенный в аппаратуру, значительно отличается от того, который был применен в машине Атлас. Блок-схема процессора БЭСМ-6 представлена на рис. 1.

рис. 1

Система команд БЭСМ-6 допускает индексную и косвенную адресацию. Как правило, действительный (исполнительный) адрес получается путем прибавления содержимого одного из индексных регистров к коду адреса, записанному в команде. Полученный после таких манипуляций адрес называется исполнительным математическим адресом и не является еще адресом конкретной физической ячейки памяти, к которой должно быть произведено обращение по этому адресу.

**Как же получается действительный физический адрес?**

Математический 15-разрядный адрес БЭСМ-6 делится на две части Старшие 5 разрядов определяют номер математической страницы, а младшие 10 разрядов - номер слова в данной странице. В структуре машины предусмотрена небольшая по объему, очень быстрая память, хранящая таблицу соответствия математических страниц (или листов) физическим страницам, и пятиразрядный номер математической страницы аппаратно подменяется соответствующим номером физической страницы. Иными словами, номер математической страницы можно рассматривать как адрес строки в таблице соответствий, по которому выбирается необходимая физическая страница реально существующей памяти машины. Таким образом, полный физический адрес слова в памяти формируется из раз рядов номера физической страницы, выбранных из таблицы, и разрядов номера слова в странице, взятых из математического адреса.

Отметим еще раз, что подмена номеров страниц производится очень быстро и делается полностью автоматически. Этот аппарат подмены номеров страниц позволяет в конечном итоге любой математической странице приписывать физический номер, а следовательно, дает возможность произвольным образом распределять реальную физическую память между математическими страницами путем изменения содержания регистровой памяти таблицы соответствия, или "приписки" таблицы.

Это открывает широкие возможности для динамического распределения и перераспределения ресурсов физической памяти между задачами, идущими в мультипрограммном режиме.

Аппарат защиты памяти, имеющий также очень важное значение для машин, работающих в мультипрограммном режиме, в БЭСМ-6 также тесно связан с постраничной организацией памяти. Каждая математическая страница может находиться в состоянии или закрытом, или открытом для чтения и записи чисел и команд. Аппаратно каждой странице ставится в соответствие триггер, указывающий ее состояние (открыта или закрыта по записи и чтению чисел); признаком защиты страницы по обращению за командами является код из всех нулей в строке таблицы соответствий. Тем самым в БЭСМ-6 защита по числам и по командам разделена. Это полезно для организации программ, обслуживающих несколько независимых задач.

Кроме аппарата защиты памяти в БЭСМ-6 существует аппарат защиты привилегированных команд. Ряд команд, с помощью которых можно изменять содержимое управляющих регистров, непосредственно воздействовать на внешние устройства и внешние каналы, связанные с машиной, могут выполняться только в "супервизорном" режиме, т. е. в режиме, в котором работает операционная система. Попытка выполнить привилегированную команду в обычной программе приводит к прерыванию и прекращению выполнения этой программы. Супервизорный и обычный режимы работы машины тем самым аппаратурно различаются, в супервизорном режиме допустимо выполнение всего репертуара команд машины, в обычном математическом режиме часть команд не может быть выполнена.

Защита памяти и привилегированных команд позволяет так организовать одновременное решение нескольких задач на машине, чтобы полностью исключить всякое влияние одних задач на другие. Как правило, почти во всех программах, первый раз пропускаемых на машине, содержатся ошибки, порой самые непредвиденные, и очень важно, чтобы эти ошибки не повлекли за собой отказ всей системы в целом или порчи соседних программ и задач. Система прерываний БЭСМ-6 построена следующим образом. В аппаратуре машины предусмотрен так называемый главный регистр прерываний (ГРП) и регистр маски (РМ).

За каждым разрядом ГРП закреплена функция сигнализации появления сигнала прерываний от конкретных внешних устройств или от вполне определенных ситуаций, сложившихся в центральном устройстве машины. Соответственно главный 48-разрядный регистр прерываний разделен на две половины. В первой его половине расположены сигналы прерываний, поступающие от внешних устройств, сигнализирующие об окончании обменов, о появлении информации на входных регистрах, пришедшей с внешних устройств и т.д. Во второй половине расположены разряды, сигнализирующие об особых ситуациях, возникших внутри машины, например о переполнениях на арифметическом устройстве, о попытке деления на нуль, о попытках обращения к закрытой странице за числами или за командами, о возникновении контроля при передачах слов и командной информации между памятью и арифметическим устройством. Внутренние прерывания имеют более старший приоритет по сравнению с прерываниями от внешних устройств.

Регистр маски, содержание которого может меняться определенными привилегированными командами, служит для того, чтобы позволить операционной системе на определенное время блокировать срабатывание схемы прерывания от внешних устройств, не требующих моментальной реакции на свои сигналы. Маска тем самым позволяет "закрывать глаза" на некоторые внешние прерывания. Это важно для организации программно управляемого приоритета обслуживания внешних запросов, и операционная система широко этим пользуется. Один из разрядов главного регистра прерываний связан с периферийным регистром прерывания и сигнализирует лишь о том, что на периферийном регистре появился где-то сигнал прерывания. Периферийный регистр связан с медленнодействующими внешними устройствами, сигналы от которых поступают достаточно редко. Как и главный регистр прерываний, периферийный регистр имеет свою маску, позволяющую блокировать появление прерываний, реагировать на которые в данный момент по какой-либо причине нежелательно. Таким образом, система прерываний БЭСМ-6 построена по иерархическому принципу. Кроме того, в машине существует несколько регистров, в разрядах которых появляется информация о текущем состоянии внешних устройств. Эту информацию в любой момент можно извлечь и уяснить, например, такие вещи: находится ли данное устройство или канал в исправном состоянии, включено ли данное устройство, подматывается ли магнитная лента и т. д. Главный регистр прерываний, периферийный регистр прерываний и регистры состояний позволяют вкупе машине БЭСМ-6 реагировать на 240 различных признаков, характеризующих состояние внешней среды и внутреннее состояние машины.

Как же происходит прерывание? Если в каком-либо разряде ГРП появился сигнал прерывания и в соответствующем разряде маски этот сигнал не заблокирован, текущая программа прекращает свое выполнение, и управление принудительно передается на некоторую фиксированную физическую ячейку памяти. Одновременно машина переводится в состояние супервизорного режима, отменяются все другие прерывания и в некоторых управляющих регистрах аппаратно запоминается состояние прерванной программы. Это состояние характеризуется номером команды, на которой произошло прерывание, и статусом прерванной программы, т. е. запоминается тот режим, в котором она выполнялась. Прерванная программа, например, могла выполняться, в свою очередь, в режиме супервизора или в обычном режиме, могла выполняться в режиме блокировки схемы приписки адресов или в режиме преобразования математических адресов в физические, наконец, могла выполняться в режиме экстракодов, о котором будет сказано несколько ниже, в режиме блокировки прерываний по переполнению арифметического устройства. Иными словами, при прерывании схемно запоминается тот минимум, который необходим для последующего возврата в прерванную программу. Блок управляющей программы, анализирующий причину прерывания и организующий обращение к соответствующему блоку реакции на данное прерывание, располагается, начиная с фиксированной ячейки, на которую передается управление. Можно сказать, что операционная система начинается с этой фиксированной ячейки.

Для того чтобы пояснить значение структурных особенностей БЭСМ-6, мы расскажем о том, как работают некоторые блоки диспетчера операционной системы, реагирующие на прерывания от внешних устройств, на прерывания по защите памяти, организующие таблицу соответствий, обеспечивающие переключение задач в мультипрограммном режиме. Эти блоки называют внутренними блоками диспетчера, они обычно принадлежат к резидентской части операционной системы и составляют основу ее ядра. Мы начнем с блока реакций на прерывания от внешних устройств. Как уже говорилось, при прерывании производится принудительная передача управления на фиксированную ячейку, как раз с этой ячейки и начинается программа реакции на прерывания, и начинается она блоком упрятывания.

Прежде всего в этом блоке производится пересылка содержимого сумматора (его старших и младших разрядов) в некоторые ячейки закрытой области памяти. Это делается для того, чтобы программы операционной системы могли свободно пользоваться сумматором, не заботясь о его сохранности. Затем производится пересылка значений двух индексных регистров в память с тем, чтобы сохранить то их содержание, которое было им придано прерванной задачей, и тем самым дать возможность программам операционной системы пользоваться этими индексными регистрами для своих целей. Далее на сумматор считывается содержимое главного регистра прерываний, отыскивается номер самого старшего разряда, давшего прерывание, и по этому номеру производится передача управления по переключателю на блок, реагирующий именно на прерывание с данным номером. Например, если оказалось, что номер самого старшего разряда равен 3, то это означает, что закончился обмен по магнитному барабану и управление передается на специальный блок, занимающийся обработкой прерываний по концу обмена. Если номер оказался равным 1, то это означает наступление момента, когда можно выдавать некоторый символ на строку печати, и передача управления происходит на соответствующий блок, организующий выдачу строки символов на печатающее устройство. Процедура упрятывания, сохранения необходимой информации и передачи управления на соответствующий блок реакции отнимает у центрального процессора около 23 мксек.

Несмотря на то, что это время кажется не очень большим, оно ставит некоторые ограничения на максимально допустимую частоту поступающих прерываний, с которыми могут справиться процессор и операционная система. Поэтому очень важно оптимизировать по времени работу блоков операционной системы, идущих с закрытым прерыванием. Как мы знаем, при прерывании машина переводится в супервизорный режим, в состояние с запрещением дальнейшего прерывания и с блокировкой схемы "приписывания" адресов. Для того чтобы не потерять, точнее, не упустить время реакции на другие прерывания, которые могут возникнуть в момент работы центрального процессора в этом режиме с "закрытыми глазами", нужно заботиться о том, чтобы как можно скорее "открыть" прерывания.

В диспетчере принята следующая логика, уменьшающая вероятность потери прерываний. Любая программа, реагирующая на конкретную причину прерываний, состоит из двух блоков: блока минимально необходимой реакции, идущего с закрытым прерыванием, и блока, который может производить работу с открытым (незапрещенным) прерыванием. После выполнения блоков минимально необходимой реакции управление передается на блок поиска старшего разряда, давшего прерывание. В том случае, если не появилось новых прерываний, управление передается блоку реакции, который идет с открытым прерыванием. Блок-схема программы реакции на прерывания представлена на рис. 2. Из приведенной блок-схемы видно, что блоки минимальной реакции на соответствующее прерывание завершают свою работу постановкой в очередь заказа на дальнейшую обработку данного прерывания, которая может идти уже с открытым прерыванием. Чтобы была более понятна разница между блоком минимальной реакции на i-е прерывание и блоком дальнейшей обработки, приведем следующий пример.

рис. 2

В процессе считывания информации с перфокарт прерыванием фиксируется момент появления считанной строки (или столбца) на регистре мультиплексного канала. Эта информация должна как можно скорее быть снята с этого регистра, чтобы освободить его для дальнейшей работы.

Блок минимальной реакции выполняет работу по освобождению входного регистра и записи информации в ячейки ОЗУ. Кроме этой работы блок реакции на прерывания должен выполнить большой объем работы по семантическому контролю содержания строки, по компоновке символов для засылки их в память машины, т. е. выполнить работы по редактированию введенной информации. Эта работа должна быть выполнена до прихода со считывающего устройства следующей строки информации. Между моментами появления строк проходит сравнительно большой промежуток времени, и эти работы по редактированию на проходе можно выполнять "не спеша".

Блок минимальной реакции заканчивается выдачей заказа на работу по редактированию, причем этой работе назначается сравнительно невысокий приоритет. Блок выбора работы для продолжения обслуживаний прерываний выбирает на исполнение работу с наивысшим приоритетом, вообще говоря, вне зависимости от того, в какой последовательности реально возникли прерывания, поставившие в очередь работы на исполнение.

Для того чтобы лучше пояснить значение аппарата защиты памяти в БЭСМ- 6, расскажем о том, как этим аппаратом пользуется операционная система (ОС). Задача, поступающая в систему, снабжается некоторым описанием тех ресурсов, которые она потребует от системы в процессе своего решения. Эта информация называется паспортом задачи. В частности, в паспорте указывается число листов ОЗУ, которое ей потребуется для решения.

На основе описания ресурсов требуемой математической памяти ОС анализирует наличие свободных ресурсов в машине к моменту поступления задачи, и если свободных ресурсов достаточно, то задача вводится в решение. При этом ОС назначает каждому требуемому математическому листу памяти задачи физический лист оперативного запоминающего устройства либо страницу на магнитном барабане. В результате такого назначения в операционной системе формируется таблица соответствия математических листов их физическому месторасположению.

В тот момент, когда данная задача начинает решаться, точнее, в тот момент, когда в ее распоряжение отдается центральный процессор, на основе этой таблицы формируется содержание аппаратных регистров приписки для тех математических листов, которым выделена в данный момент оперативная физическая память. Тем математическим листам, которым выделена память на барабанах или которые вообще не затребовала задача, приписывается признак защиты. Если в ходе своего решения задача не обращается к листам, отображенным на магнитные барабаны, или к листам, для нее не существующим, то процесс счета происходит нормально. В том случае, если программа задачи обратилась к математическому листу, который находится на барабане, срабатывает прерывание по защите, и операционная система, анализируя причину прерывания, стремится "перекачать" этот лист с барабана. на первый свободный лист оперативной памяти. Иногда для этого ей приходится "сбросить" какой-либо лист на барабан, освободив тем самым лист оперативной памяти. Пока производится "подкачка" листов для того, чтобы можно было продолжить решение задачи, прерванное по защите программы, центральный процессор отдается в распоряжение другой задаче. Очевидно, стратегия, т. е. методы распределения листов задач между оперативной памятью и памятью на магнитном барабане должны быть выбраны так, чтобы минимизировать число "сбросов" и "подкачек", так как это прямо влияет на эффективность мультипрограммной работы. Таких методов существует много, и о них должен быть особый разговор, выходящий за рамки данной книги. Итак, система защиты и приписки используется операционной системой для организации динамического распределения ресурсов двухуровневой памяти. Но этой же системой ОС пользуется для того, чтобы совместить вычисления с работой внешних запоминающих устройств и устройств ввода-вывода.

Как только программа задачи обращается к процедуре вывода (ввода) информации с какого-либо математического листа, этот лист "закрывается" по защите вплоть до окончания процесса вывода (ввода). Теперь задача, задав ОС директиву выполнить ввод-вывод, может свободно продолжать вычисления, которые не пользуются информацией с листа, находящегося в обмене, и в этом случае работа по вводу-выводу будет совмещена со счетом. Стоит лишь задаче обратиться к листу, находящемуся в процессе обмена, как произойдет прерывание и операционная система заставит эту задачу подождать окончания обмена. Тем самым достигается синхронизация параллельных процессов счета и ввода-вывода. И, наконец, если программа задачи обратилась к незаказанному листу ОЗУ, т.е. совершила попытку обратиться к "чужому" листу, то опять срабатывает прерывание по защите и операционная система фиксирует ошибку в программе и выбрасывает задачу из решения, предоставляя занятые ею ресурсы в распоряжение других задач.