

О СТРУКТУРЕ СВЕРХОПЕРАТИВНЫХ ЗАПОМИНАЮЩИХ УСТРОЙСТВ В ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

А.А. Папернов, О.И. Гаранина
(Москва)

Рассматриваются методы использования сверхоперативных запоминающих устройств (ЗУ) в вычислительных системах, состоящих из нескольких вычислителей, одновременно выполняющих несколько программ, заранее не распределенных между вычислителями. В связи с этим исключается возможность хранения программы и исходной информации к ним в ЗУ, связанных только с отдельными вычислителями. Результаты вычислений также должны направляться в некоторые зоны памяти, независимые от того, каким вычислителем они выработаны. В то же время обращение всех вычислителей за каждой командой и операндом в общее поле памяти (ОПП), доступное всем вычислителям, связано со снижением быстродействия всей вычислительной системы из-за весьма вероятного столкновения запросов отдельных вычислителей в ОПП и образования очередей на исполнение запросов. Основным назначением сверхоперативных ЗУ (СЗУ) в вычислительных системах является вынесение из ОПП тех величин, которые легче всего поддаются такому вынесению, а именно, промежуточных результатов расчетов, отдельных исходных данных, многократно используемых в процессе расчета и циклических участков программ. Объем памяти, необходимой для хранения этих данных, оказывается отно-

сительно небольшим, а количество обращений в процессе исполнения программ, приходящееся на одну ячейку, значительно больше, чем количество обращений, приходящееся на одну ячейку ОПП.

Введение в состав вычислителей СЗУ небольшого объема позволяет:

а) уменьшить поток запросов к ОПП от каждого вычислителя, возможность становления запросов в ОПП и, тем самым, время обслуживания запросов;

б) увеличить собственное быстродействие вычислителя за счет существенной разницы во времени обращения к ОПП и СЗУ при обращении за промежуточными результатами и командами циклической части программы;

в) увеличить быстродействие вычислителя за счет того, что введение СЗУ позволяет организовать более глубокое совмещение операций в вычислителе.

Основными вариантами адресации ячеек СЗУ для хранения данных являются: СЗУ с непосредственной адресацией, СЗУ с магазинной адресацией, СЗУ с ассоциативной адресацией. Выбор способа адресации ячеек СЗУ существенно влияет на структуру вычислителя.

Непосредственная адресация сверхоперативной памяти

В командах в явном виде записываются адреса оперативного ЗУ (ОЗУ) и СЗУ. Ячейки СЗУ удобно использовать не только для хранения промежуточных результатов, но и для хранения индексных величин, счетчиков циклов и адресов возврата из подпрограмм. В этом случае операции с индексными величинами ничем не отличаются от арифметических. Это позволяет сократить количество кодов операции, а также для операций над индексными величинами использовать основной сумматор вычислителя. Структура СЗУ с непосредственной адресацией проста. СЗУ должно иметь числовую часть, где записываются промежуточные результаты и операнды; числовая часть может быть выполнена на регистрах или на обычном ЗУ на ферритовых сердечниках с малым временем обращения. Примерная логическая схема СЗУ с непосредственной адресацией ячеек приведена на рис. 1.

Магазинная адресация сверхоперативной памяти

При этом способе адресации команды являются безадресными,

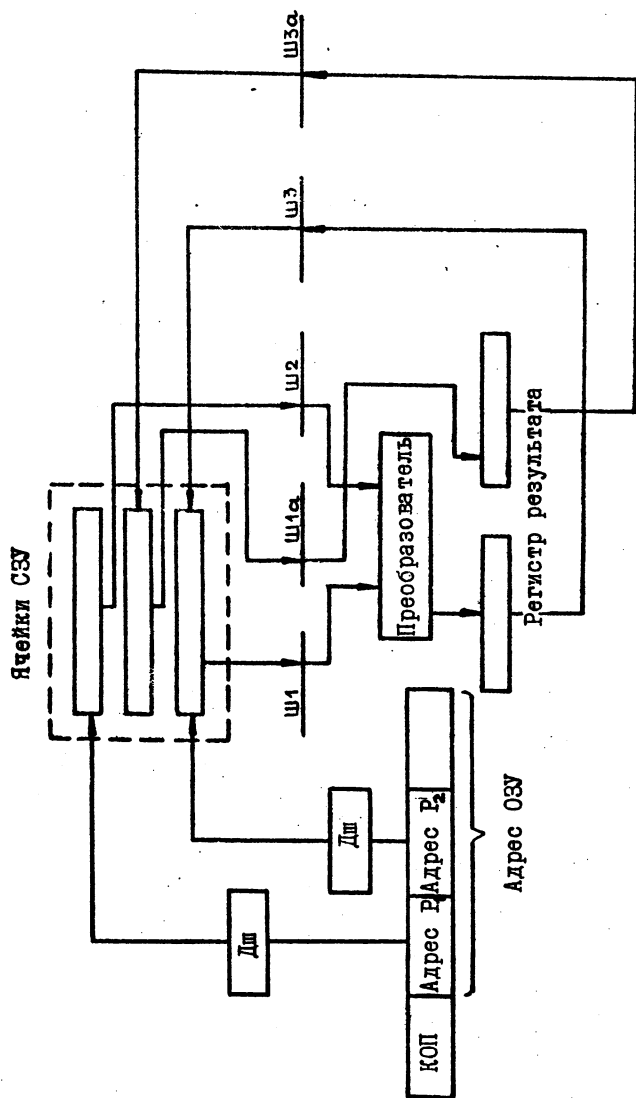


Рис. 1. СЗУ с непосредственной адресацией ячеек.

а адрес следующего операнта формируется из адреса предыдущего операнта путем прибавления ± 1 . Поэтому адресация такого СЗУ осуществляется специальным счетчиком, в любой момент времени указывающим номер ячейки, из которой надлежит выполнить очередное считывание из СЗУ либо в ОЗУ, либо в арифметическое устройство (АУ). После того, как считывание из СЗУ окончено, из кода, записанного на счетчике, всегда вычитается единица. Это значит, что серия последовательных считываний из СЗУ может производиться только из группы ячеек с последовательно убывающими номерами, начиная с номера, указываемого счетчиком перед началом первого считывания. Перед записью в СЗУ кода из ОЗУ или результата операции из АУ к коду, записанному на счетчике, всегда прибавляется единица. Это значит, что серия последовательных записей в СЗУ может производиться только в группу ячеек с последовательно возрастающими номерами, начиная с номера, превышающего на единицу число, указываемое счетчиком перед началом первой записи. При незаполненном СЗУ показание счетчика устанавливается в нуль. Это значит, что СЗУ всегда начинает заполняться со своей первой ячейки.

Если в момент начала двухместной операции показание счетчика было равно m , то в качестве её операндов используются коды, хранившиеся в ячейках m и $m - 1$, а результат записывается в ячейку $m - 1$. В качестве операнта одностепенной операции используется код из ячейки, номер которой указывается счетчиком. После окончания операции в эту же ячейку записывается результат операции. Операнды, использованные в любой из операций, тем самым автоматически выводятся из СЗУ.

Ввиду того, что в любом случае операнды занимают в памяти вполне определенное положение, нет необходимости указывать их адреса в команде. Поэтому команды, содержащие код арифметической или логической операции, являются безадресными. Обмен информацией между СЗУ и ОЗУ выполняется под управлением специальных команд, содержащих адрес ячейки ОЗУ и признак, указывающий направление передачи.

Описанная выше структура СЗУ позволяет выполнять вычисления по любой формуле, содержащей основные арифметические операции без записи промежуточных результатов в ОЗУ. После использования всех операндов и окончания вычислений по формуле СЗУ освобождается. Однако, если один и тот же промежуточный результат используется в формуле многократно, то его нужно ли-

бо каждый раз вычислять заново, либо запоминать в оперативном ЗУ. С целью уменьшения объема вычислений и количества обращений в ОПЗ в состав основных операций вычислителя вводятся операции: повторить и обменять. Если показание счетчика равно m , то по команде "повторить" к счетчику прибавляется единица и в $(m + 1)$ -ой команде повторяется содержимое m -ой ячейки. По команде "обменять" код счетчика не изменяется, а содержимое m -ой и $(m - 1)$ -ой ячеек меняется местами. Следует заметить, что операции "повторить" и "обменять" позволяют только сократить, но не исключить полностью необходимость запоминания отдельных промежуточных результатов. При наличии в составе вычислителя кроме СЗУ для записи промежуточных результатов и операндов еще одного СЗУ магазинного типа для записи адресов возврата из подпрограмм, возможно неограниченное обращение из одних подпрограмм в другие, в том числе и рекурсивные.

Ассоциативная адресация сверхоперативной памяти

Ассоциативная адресация означает, что поиск в СЗУ осуществляется не по адресу, а по заданной информации о его значении. В ячейках СЗУ операнды и промежуточные результаты записываются вместе со своими адресами, относящимися к ОЗУ. Следовательно, СЗУ содержит адресное поле и информационное поле, в котором записываются коды, подлежащие хранению в СЗУ. Вся программа составляется в адресах ОЗУ, как будто СЗУ не существует. Выполнение любой команды, связанной с обращением в ОЗУ для записи или считывания, начинается с проверки наличия в СЗУ данной ячейки, т.е. с проверки совпадения адреса, записанного в команде, с адресами, записанными в адресном поле всех ячеек СЗУ. Запоминающее устройство с ассоциативной адресацией только тогда может выполнять функции сверхбыстродействующей памяти, если сравнение заданного адреса ОЗУ с адресами, записанными в СЗУ, будет производиться одновременно во всех ячейках СЗУ. Сигнал S_1 совпадения заданного m -разрядного адреса с хранящимся в i -ячейке СЗУ адресом может быть непосредственно использован для считывания кода, записанного в информационном поле той же ячейки, в которой зафиксировано совпадение. Вытеснение ячеек из СЗУ для записи в СЗУ новой информации и запись в СЗУ происходит автоматически. Вытесняемая из СЗУ информация передается в ОЗУ по адресу, записанному

в адресном поле ячейки СЗУ. Для того, чтобы иметь возможность выбрать в СЗУ ячейку для записи новой информации, организуется очередь ячеек СЗУ. После обращения к любой ячейке СЗУ очередь перестраивается: ячейке, номер которой был равен q , после обращения к ней присваивается номер 1, соответственно всем ячейкам, номера которых были меньше, чем q , присваиваются новые номера, на единицу больше прежних. Если одна из двух любых ячеек имеет номер больший, чем вторая, то это значит, что последнее обращение к первой ячейке имело место раньше, чем ко второй. Вытесняемая из СЗУ ячейка имеет наибольший номер в очереди. Фиксация и перестройка очереди осуществляется специальным устройством очереди, входящим в состав устройства управления СЗУ. Сигнал обращения к любой ячейке СЗУ одновременно является сигналом для перестройки очереди. Логическая структура СЗУ с ассоциативным способом адресации представлена на рис. 2.

Сравнивая все три способа адресации СЗУ следует отметить, что основным достоинством способа ассоциативной адресации СЗУ является простота программирования, связанная с тем, что программист может вообще не учитывать наличия СЗУ. При прерывании также нет необходимости запоминать содержимое ячеек СЗУ, т.к. они будут автоматически вытесняться из СЗУ в ОЗУ в процессе выполнения прерывающей программы. Недостатком способа ассоциативной адресации является больший по сравнению с другими способами объем программ, связанный с тем, что к промежуточным результатам приходится обращаться по полным адресам, и сравнительная сложность исполнения, связанная с наличием в составе СЗУ дополнительного адресного поля, схемы ассоциативного поиска, а также устройства очереди. Достоинством магазинного принципа организации СЗУ является более экономная запись программы, связанная с исключением из неё адресов промежуточных величин. Недостатком магазинного принципа организации СЗУ является меньшая её гибкость по сравнению с непосредственно адресуемым СЗУ, затрудняющая использование ячеек СЗУ в качестве индексных регистров, счётчиков циклов; кроме того, чтобы сделать возможной правильную работу вычислителя при переполнении магазинного СЗУ, необходимо усложнить структуру вычислителя.

Достоинством способа непосредственной адресации является уменьшение объема программ, связанное с применением коротких

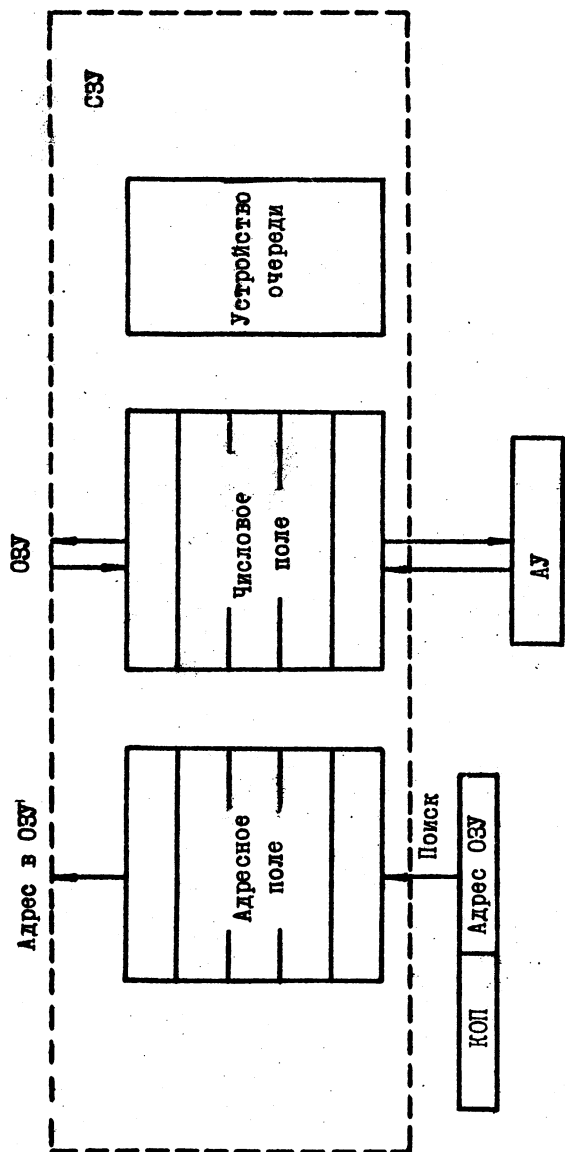


Рис. 2. СЗУ с ассоциативным способом обращения.

адресов, универсальность использования ячеек СЗУ, значительная логическая простота. Недостатком как непосредственной адресации, так и магазинной организации является усложнение программирования и необходимость при прерываниях запоминать полный объем СЗУ.

Сверхоперативные ЗУ для хранения программ

Основными вариантами СЗУ для команд являются: СЗУк типа упреждающей памяти, СЗУк для хранения циклической части программ.

В связи с тем, что в программах команды имеют последовательные адреса, целесообразно использовать непосредственную адресацию ячеек СЗУк. Для СЗУк типа упреждающей памяти может быть использовано несколько регистров или обычное ЗУ на ферритовых сердечниках малого объема с малым временем обращения. При использовании СЗУк с гибким уровнем опережения для целей адресации СЗУк при записи и считывании используются два отдельных счётчика. Эффективность применения СЗУк типа упреждающей памяти получается за счёт совмещения во времени обращения за командами с выполнением операций в вычислителе. Условные переходы должны останавливать дальнейшую подготовку команд, либо должна готовиться одна ветвь. Если при выполнении команды условного перехода оказывается, что подготовлена ненужная ветвь, оба счётчика сбрасываются в "0" и команды выбираются из ОЗУ. Уровень опережения в подготовке команд может быть различным, максимально возможный уровень опережения равен размеру СЗУк. На циклических участках программы СЗУк такого типа используется так же, как и на линейных участках, поэтому сократить количество обращений в ОПП за командами не удастся, в связи с этим в вычислительных системах с ОПП СЗУк только как упреждающую память для команд использовать нецелесообразно. Для того, чтобы снизить поток запросов в ОЗУ за командами путем многократного считывания из СЗУк команд, входящих в циклические участки программы, нужно ввести специальный режим работы внутри цикла. В этом режиме заполнение СЗУк прекращается и команды внутри цикла считываются из числа уже хранящихся в СЗУк. Сигналы для входа и выхода из режима цикла могут вырабатываться следующим образом: перед записью любой команды в СЗУк код её операции анализируется. Если операция не связана с передачей управления, или представляет собой безус-

ловную передачу управления, то заполнение СЗУк продолжается. Если операция представляет собой условную передачу управления, то заполнение СЗУк приостанавливается до тех пор, пока не будет выработано условие, определяющее направление перехода. Если оказывается, что переход не состоится или состоится переход вперед (к команде с большим номером) или, наконец, переход назад (т.е. к команде с меньшим номером), выводящий программу за пределы СЗУк, то режим заполнения СЗУк восстанавливается. В случае перехода назад к команде, не выводящей за пределы СЗУк, режим цикла сохраняется. Выход из цикла производится следующей командой условного перехода, в результате выполнения которой переход не состоится.

Недостатком способа автоматического входа и выхода из режима цикла является неэффективное использование СЗУк в случае, когда размер цикла превышает размер СЗУк или в случае сложной внутренней структуры цикла. Если вход и выход из режима цикла осуществляется специальными командами, которые включаются программистом, то эффективность использования СЗУк может быть повышена.