

4.4. Запоминающие устройства с произвольной выборкой

На рис. 23 показано обозначение запоминающее устройство с произвольной выборкой (ЗУПВ) и его внутренняя структура [2, 9].

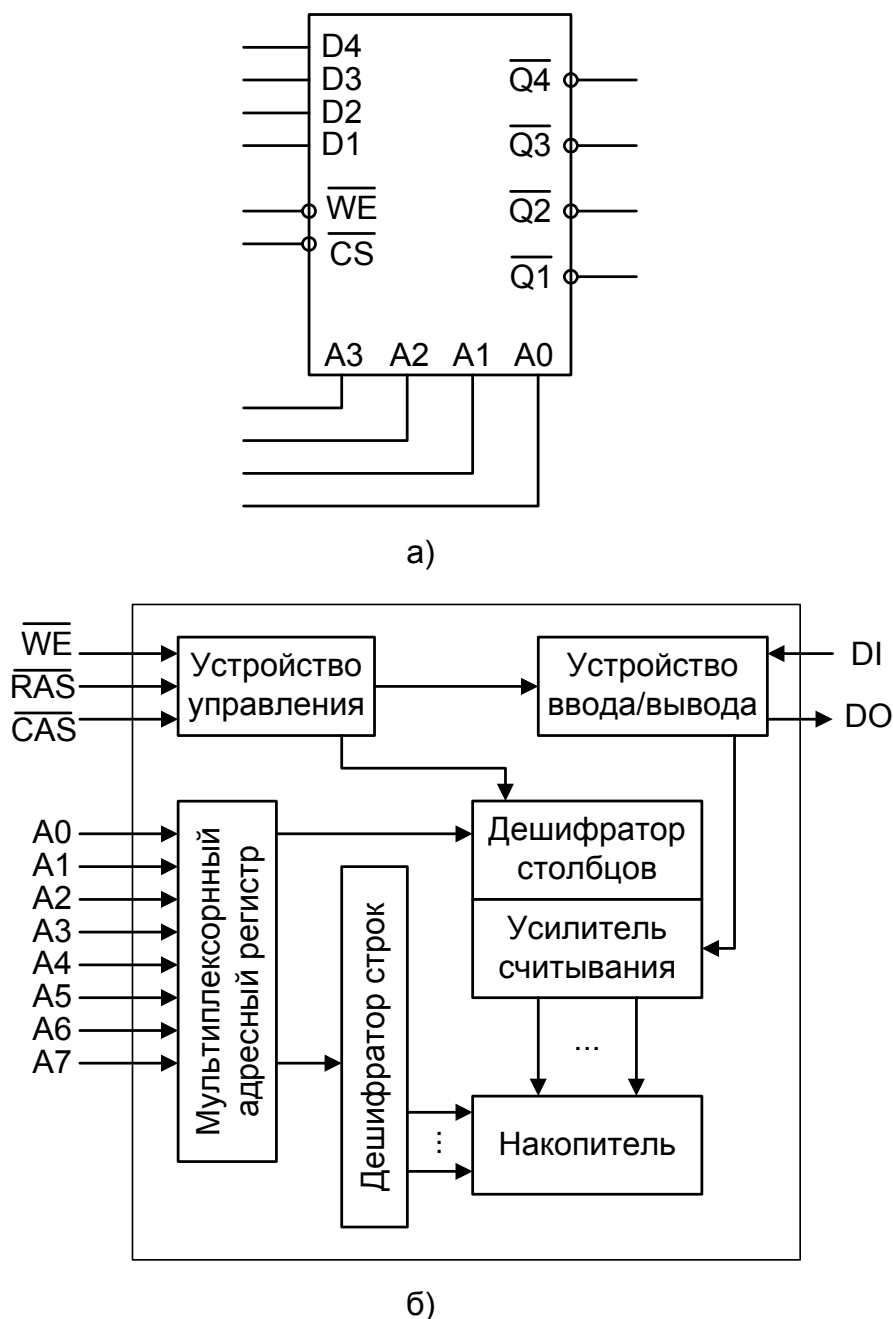


Рис. 23. ОЗУ с произвольной выборкой: а) ОЗУ 16x4 бит; б) ОЗУ 64 кбит x1 (KP565PY5)

Здесь: D1–Dn (справа DI) – информационные входы; Q1–Qn (справа DO) – инверсные выходы; A1–An – адресные входы; WE# – запись/чтение; CS# (Chip Select) – выбор кристалла; CAS# (Column

Address Strobe) и RAS# (Row Address Strobe) – сигнал выборки столбца и строки соответственно.

Представленное здесь ЗУПВ – это ДОЗУ с организацией хранения информации 65536 бит на 1 разряд. Накопительная матрица с однотранзисторными запоминающими элементами имеет размер 512x128. Для уменьшения количества задействованных ножек у ИМС (16-входовый DIP-корпус) применена мультипликация адреса, что видно на рисунке по наличию отдельных дешифраторов строк и столбцов. Устройство управления включает два генератора тактовых сигналов и генератор сигналов записи и обеспечивает 4 режима работы: записи, считывания, регенерации и мультипликации адреса. Время регенерации – 2 мс.

Для сравнения на рис. 24 показана конструкция ППЗУ с ультрафиолетовым (УФ) стиранием на МНОП-структурах с организацией представления информации 2 кб x 8 (16384 бита).

4.5. Микросхемы памяти в составе микропроцессорной системы

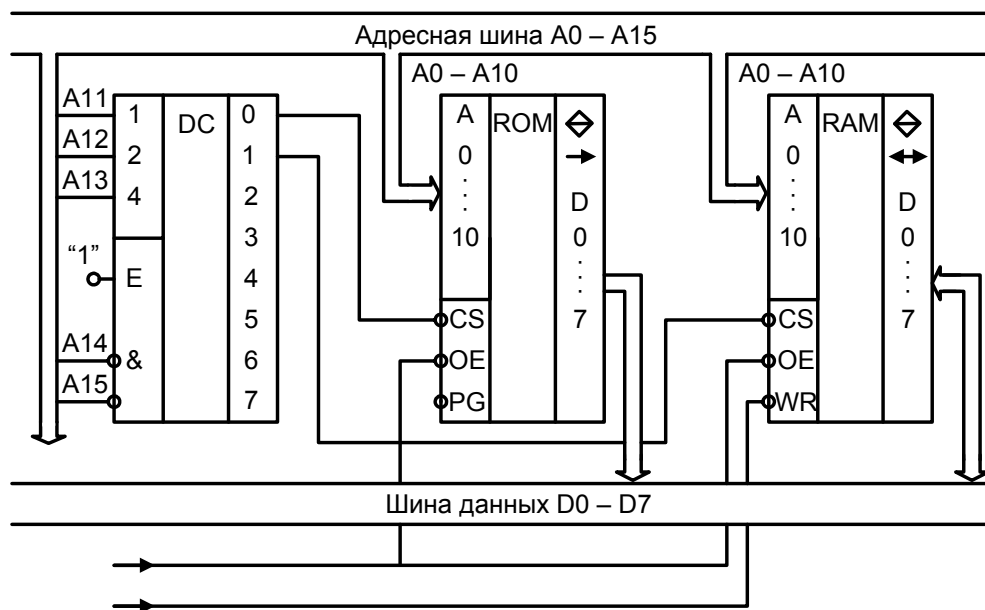


Рис. 24. Микросхемы ОЗУ (K573PY9) и ППЗУ (K573PФ5) в составе микропроцессорной системы

На рис. 24 представлено взаимодействие K573PФ2(5) и K573PY9, имеющих одинаковую организацию 2 кбит x 8, с системной магистра-

лью [9]. Байт данных с шины данных (линии D0–D7) считывается (или записывается) по адресу, выставленному на шине адреса (линии A0–A10). Естественно, число адресуемых ячеек составляет $2^{11} = 800h = 2048$. Микросхема-дешифратор K555ИД7 посредством сигнала CS# (выбор кристалла) позволяет выбрать положение ИМС ЗУ в адресном пространстве. Для данного случая это адреса 0000h–07FFh для ПЗУ(ROM) и 0800h-0FFFh для ОЗУ(RAM). Низкий уровень сигналов управления MEMW# и MEMR# активизирует процесс записи и чтения, соответственно. Напомним, что запись информации в данную ИМС ПЗУ возможен только вне микропроцессорной системы в специальном программаторе после УФ-стирания путем подачи достаточно высокого напряжения на вход PG.

4.6. Буферная память

В вычислительных системах используются подсистемы с различным быстродействием, и, в частности, с различной скоростью передачи данных (рис. 25). Обычно обмен данными между такими подсистемами реализуется с использованием прерываний или канала прямого доступа к памяти. В первую очередь, подсистема 1 формирует запрос на обслуживание по мере готовности данных к обмену. Однако обслуживание прерываний связано с непроизводительными потерями времени и при пакетном обмене производительность подсистемы 2 заметно уменьшается. При обмене данными с использованием канала прямого доступа к памяти подсистема 1 передает данные в память подсистемы 2. Данный способ обмена достаточно эффективен с точки зрения быстродействия, но для его реализации необходим довольно сложный контроллер прямого доступа к памяти.



Рис. 25. Применение буферной памяти

Наиболее эффективно обмен данными между подсистемами с различным быстродействием реализуется при наличии между ними спе-

циальной буферной памяти [2, 9]. Данные от подсистемы 1 временно запоминаются в буферной памяти до готовности подсистемы 2 принять их. Емкость буферной памяти должна быть достаточной для хранения тех блоков данных, которые подсистема 1 формирует между считываниями их подсистемой 2. Отличительной особенностью буферной памяти является запись данных с быстроедействием и под управлением подсистемы 1, а считывание – с быстроедействием и под управлением подсистемы 2 ("эластичная память"). В общем случае память должна выполнять операции записи и считывания совершенно независимо и даже одновременно, что устраняет необходимость синхронизации подсистем. Буферная память должна сохранять порядок поступления данных от подсистемы 1, т. е. работать по принципу "первое записанное слово считывается первым" (First Input First Output – FIFO). Таким образом, под буферной памятью типа FIFO понимается ЗУПВ, которое автоматически следит за порядком поступления данных и выдает их в том же порядке, допуская выполнение независимых и одновременных операций записи и считывания. На рис. 26 приведена структурная схема буферной памяти типа FIFO емкостью 64x4.

На кристалле размещены 64 4-битных регистра с независимыми цепями сдвига, организованных в 4-х последовательных 64-битных регистрах данных, 64-битный управляющий регистр, а также схема управления. Входные данные поступают на линии DI0–DI3, а вывод данных осуществляется через контакты DO0–DO3. Ввод (запись) данных производится управляющим сигналом SI (shift in), а вывод (считывание) – сигналом вывода SO (shift out). Ввод данных осуществляется только при наличии сигнала готовности ввода IR (input ready), а вывод – при наличии сигнала готовности вывода OR (output ready). Управляющий сигнал R (reset) производит сброс содержимого буфера.

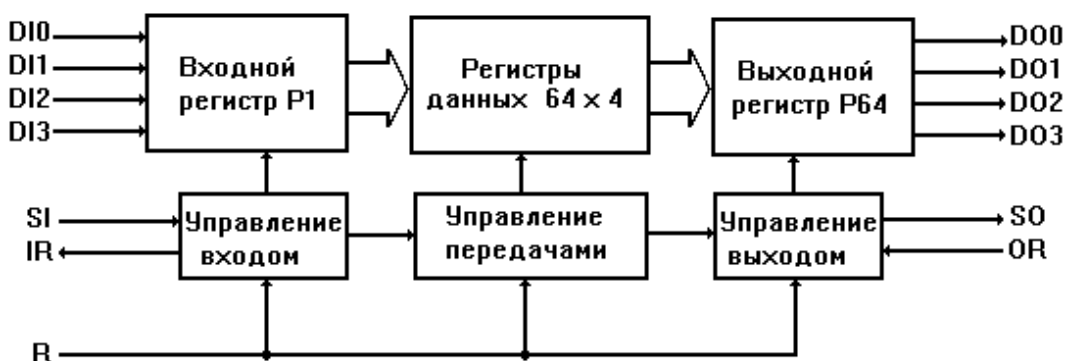


Рис. 26. Структурная схема буфера 64x4

При вводе 4-битного слова под действием сигнала SI оно автоматически передвигается в ближайший к выходу свободный регистр. Состояние регистра данных отображается в соответствующем ему управляющем триггере, совокупность триггеров образует 64-битный управляющий регистр. Если регистр содержит данные, то управляющий триггер находится в состоянии 1, а если регистр не содержит данных, то триггер находится в состоянии 0. Как только управляющий бит соседнего справа регистра изменяется на 0, слово данных автоматически сдвигается к выходу. Перед началом работы в буфер подается сигнал сброса R и все управляющие триггеры переводятся в состояние 0 (все регистры буфера свободны). На выводе IR формируется логическая 1, т. е. буфер готов воспринимать входные данные. При действии сигнала ввода SI входное слово загружается в регистр P1, а управляющий триггер этого регистра устанавливается в состояние 1: на входе IR формируется логический 0. Связи между регистрами организованы таким образом, что поступившее в P1 слово "спонтанно" копируется во всех регистрах данных FIFO и появляется на выходных линиях DO0–DO3. Теперь все 64 регистра буфера содержат одинаковые слова, управляющий триггер последнего регистра P64 находится в состоянии 1, а остальные управляющие триггеры сброшены при передаче данных в соседние справа регистры. Состояние управляющего триггера P64 выведено на линию готовности выхода OR; OR принимает значение 1, когда в триггер записывается 1. Процесс ввода может продолжаться до полного заполнения буфера; в этом случае все управляющие триггеры находятся в состоянии 1 и на линии IR сохраняется логический 0.

При подаче сигнала SO производится восприятие слова с линий DO0–DO3, управляющий триггер P64 переводится в состояние 1, на линии OR появляется логическая 1, а управляющий триггер P64 сбрасывается в 0. Затем этот процесс повторяется для остальных регистров и нуль в управляющем регистре перемещается ко входу по мере сдвига данных вправо.

В некоторых кристаллах буфера FIFO имеется дополнительная выходная линия флажка заполнения наполовину. На ней формируется сигнал 1, если число слов составляет более половины емкости буфера.

Рассмотренный принцип организации FIFO допускает выполнение записи и считывания данных независимо и одновременно. Скорость ввода определяется временным интервалом, необходимым для передачи данных из P1, а выводить данные можно с такой же скоростью. Единственным ограничением является время распространения данных через FIFO, равное времени передачи входного слова на выход незаполненного буфера FIFO. Оно равняется произведению времени внут-

ренного сдвига и числа регистра данных. В буферах FIFO, выполненных по МОП-технологии и имеющих емкость 64 слова, время распространения составляет примерно 30 мкс, а в биполярных FIFO такой же емкости – примерно 2 мкс.

Буферы можно наращивать как по числу слов, так и по их длине.

4.7. Стековая память

Стековой называют память, доступ к которой организован по принципу: последним записан – первым считан (Last Input First Output – LIFO). Использование принципа доступа к памяти на основе механизма LIFO началось с больших ЭВМ [2, 9]. Применение стековой памяти оказалось очень эффективным при построении компилирующих и интерпретирующих программ, при вычислении арифметических выражений с использованием польской инверсной записи. В малых ЭВМ она стала широко использоваться в связи с удобствами реализации процедур вызова подпрограмм и при обработке прерываний.

Принцип работы стековой памяти состоит в следующем (см. рис. 27). Когда слово А помещается в стек, оно располагается в первой свободной ячейке памяти. Следующее записываемое слово перемещает предыдущее на одну ячейку вверх и занимает его место и т. д. Запись 8-го слова (после Н) приводит к переполнению стека и потере слова А. Считывание слов из стека осуществляется в обратном порядке, начиная с слова Н, который был записан последним. Заметим, что выборка, например, слова Е невозможна до выборки слова F, что определяется механизмом обращения при записи и чтении типа LIFO. Для фиксации переполнения стека желательно формировать признак переполнения.

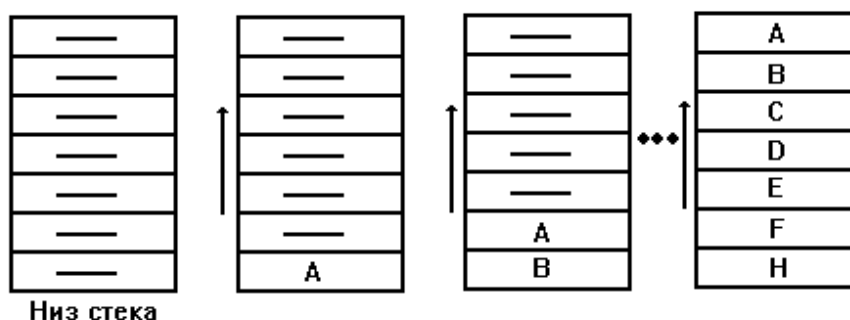


Рис. 27. Принцип работы стековой памяти

Перемещение данных при записи и считывании информации в стековой памяти подобно тому, как это имеет место в сдвигающих регистрах. С точки зрения реализации механизма доступа к стековой памяти выделяют аппаратный и аппаратно-программный (внешний) стеки.

Аппаратный стек представляет собой совокупность регистров, связи между которыми организованы таким образом, что при записи и считывании данных содержимое стека автоматически сдвигается. Обычно емкость аппаратного стека ограничена диапазоном от нескольких регистров до нескольких десятков регистров, поэтому в большинстве МП такой стек используется для хранения содержимого программного счетчика и его называют стеком команд. Основное достоинство аппаратного стека – высокое быстродействие, а недостаток – ограниченная емкость.

Наиболее распространенным в настоящее время и, возможно, лучшим вариантом организации стека в ЭВМ является использование области памяти. Для адресации стека используется указатель стека, который предварительно загружается в регистр и определяет адрес последней занятой ячейки. Помимо команд CALL и RET, по которым записывается в стек и восстанавливается содержимое программного счетчика, имеются команды PUSH и POP, которые используются для временного запоминания в стеке содержимого регистров и их восстановления, соответственно. В некоторых МП содержимое основных регистров запоминается в стеке автоматически при прерывании программ. Содержимое регистра указателя стека при записи уменьшается, а при считывании увеличивается на 1 при выполнении команд PUSH и POP, соответственно.