

5.3. Параллельная передача данных

Параллельная передача данных между контроллером и ВУ является по своей организации наиболее простым способом обмена. Для организации параллельной передачи данных помимо шины данных, количество линий в которой равно числу одновременно передаваемых битов данных, используется минимальное количество управляющих сигналов.

В простом контроллере ВУ, обеспечивающем побайтную передачу данных на внешнее устройство (рис. 32), в шине связи с ВУ используются всего два управляющих сигнала: "Выходные данные готовы" и "Данные приняты" [2, 5, 6].

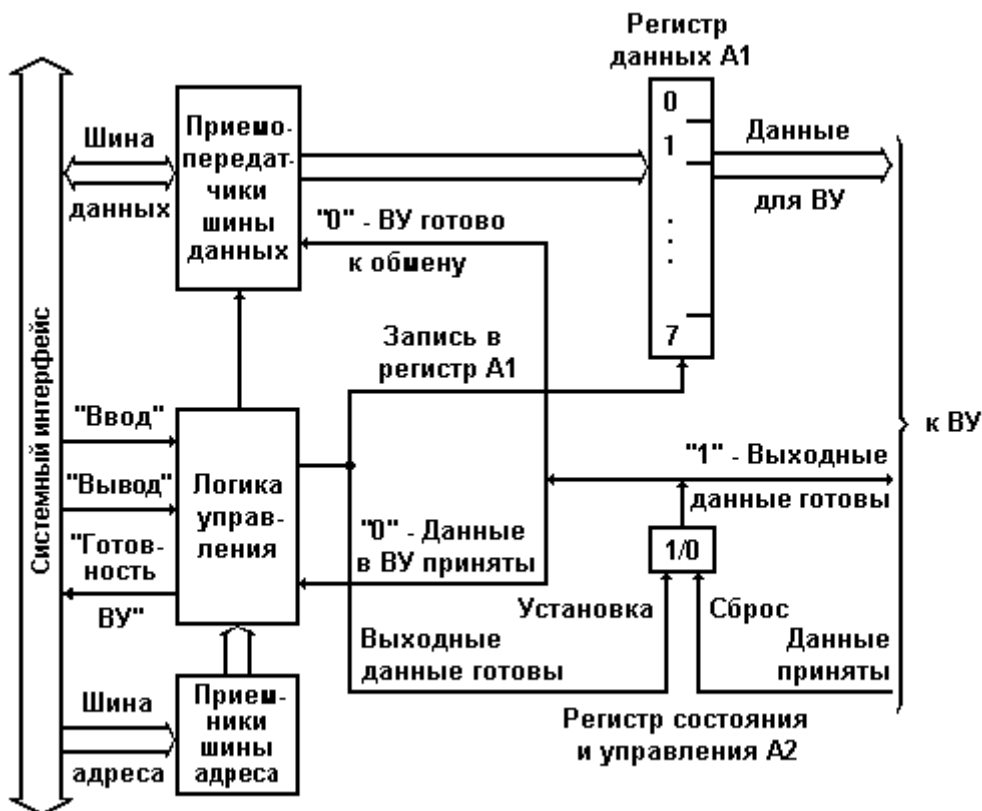


Рис. 32. Простой параллельный контроллер вывода

Для формирования управляющего сигнала "Выходные данные готовы" и приема из ВУ управляющего сигнала "Данные приняты" в контроллере используется одноразрядный адресуемый регистр состояния и управления A2 (обычно используются отдельные регистры состояния и регистра управления). Одновременно с записью очередного байта данных с шины данных системного интерфейса в адресуемый регистр

данных контроллера (порт вывода A1) в регистр состояния и управления записывается логическая единица. Тем самым формируется управляющий сигнал "Выходные данные готовы" в шине связи с ВУ.

Внешнее устройство, приняв байт данных, управляющим сигналом "Данные приняты" обнуляет регистр состояния контроллера. При этом формируются управляющий сигнал системного интерфейса "Готовность ВУ" и признак готовности ВУ к обмену, передаваемый в процессор по одной из линий шины данных системного интерфейса посредством стандартной операции ввода при реализации программы асинхронного обмена.

Логика управления контроллера обеспечивает селекцию адресов регистров контроллера, прием управляющих сигналов системного интерфейса и формирование на их основе внутренних управляющих сигналов контроллера, формирование управляющего сигнала системного интерфейса "Готовность ВУ". Для сопряжения регистров контроллера с шинами адреса и данных системного интерфейса в контроллере используются соответственно приемники шины адреса и приемопередатчики шины данных.

Рассмотрим на примере, каким образом контроллер ВУ обеспечивает параллельную передачу данных в ВУ под управлением программы асинхронного обмена. Алгоритм асинхронного обмена в данном случае передачи прост.

1. Процессор микроЭВМ проверяет готовность ВУ к приему данных.

2. Если ВУ готово к приему данных (в данном случае это логический 0 в нулевом разряде регистра A2), то данные передаются с шины данных системного интерфейса в регистр данных A1 контроллера и далее в ВУ. Иначе повторяется п. 1.

Пример 1. Фрагмент программы передачи байта данных в асинхронном режиме с использованием параллельного контроллера ВУ (рис. 32). Для написания программы асинхронной передачи воспользуемся командами процессора 8086.

	MOV DX, A2;	номер порта A2 помещаем в DX
m1:	IN AL, DX;	чтение байта из порта A2
	TEST AL, 1;	проверка нулевого состояния регистра A2
	JNS ml;	переход на метку ml если разряд не нулевой
	MOV AL, 64;	выводимый байт данных помещается в AL
	MOV DX, A1;	номер порта A1 записываем в DX
	OUT DX, AL;	содержимое регистра AX передаем в порт A1

Команда во второй строке приводит к следующим действиям. При ее выполнении процессор по шине адреса передает в контроллер адрес A2, сопровождая его сигналом "Ввод" (IORD#; здесь и далее в скобках указаны сигналы на шине ISA). Логика управления контроллера, реагируя на эти сигналы, обеспечивает передачу в процессор содержимого регистра состояния A2 по шине данных системного интерфейса.

Команда в третьей строке приводит к следующим действиям. Процессор проверяет значение соответствующего разряда принятых данных. Нуль в этом разряде указывает на неготовность ВУ к приему данных и, следовательно, на необходимость возврата к проверке содержимого A2, т. е. процессор, выполняя три первые команды, ожидает готовности ВУ к приему данных. Единица в этом разряде подтверждает готовность ВУ и, следовательно, возможность передачи байта данных.

В седьмой строке осуществляется пересылка данных из регистра AX процессора в регистр данных контроллера A1. Процессор по шине адреса передает в контроллер адрес A1, а по шине данных – байт данных, сопровождая их сигналом "Вывод" (IOWR#). Логика управления контроллера обеспечивает запись данных с шины данных в регистр данных A1 и устанавливает в нуль бит готовности регистра состояния A2, формируя тем самым управляющий сигнал для ВУ "Выходные данные готовы". Внешнее устройство принимает байт данных и управляющим сигналом "Данные приняты" устанавливает в единицу регистр состояния A2. Далее контроллер ВУ по этому сигналу может сформировать и передать в процессор сигнал "Готовность ВУ", который в данном случае извещает процессор о приеме данных внешним устройством и разрешает процессору снять сигнал "Вывод" и тем самым завершить цикл вывода данных в команде пересылки, однако в IBM-совместимых персональных компьютерах с шиной ISA сигнал "Готовность ВУ" не формируется, а имеется сигнал IO CH RDY#, позволяющий продлить цикл обмена, если устройство недостаточно быстрое. В данном случае нет необходимости в сигнале "Готовность ВУ", т.к. шина ISA является синхронной и, следовательно, все операции выполняются по тактовым импульсам.

Блок-схема простого контроллера ВУ, обеспечивающего побайтный прием данных из ВУ, приведена на рис. 33 [5, 6]. В этом контроллере при взаимодействии с внешним устройством также используются два управляющих сигнала: "Данные от ВУ готовы" и "Данные приняты".

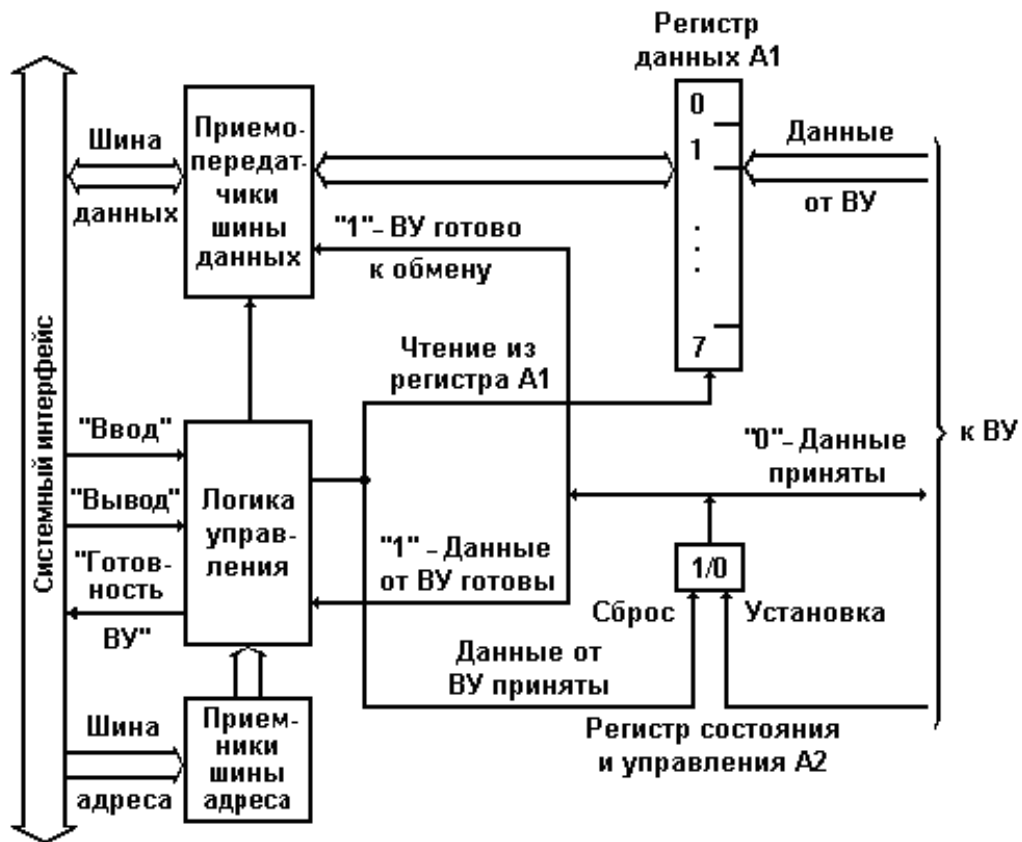


Рис. 33. Простой параллельный контроллер ввода

Для формирования управляющего сигнала "Данные приняты" и приема из ВУ управляющего сигнала "Данные от ВУ готовы" используется одноразрядный адресуемый регистр состояния и управления А2.

Внешнее устройство записывает в регистр данных контроллера А1 очередной байт данных и управляющим сигналом "Данные от ВУ готовы" устанавливает в единицу регистр состояния и управления А2.

При этом формируются: управляющий сигнал системного интерфейса "Готовность ВУ"; признак готовности ВУ к обмену, передаваемый в процессор по одной из линий шины данных системного интерфейса посредством операции ввода при реализации программы асинхронного обмена.

Тем самым контроллер извещает процессор о готовности данных в регистре А1. Процессор, выполняя программу асинхронного обмена, читает байт данных из регистра данных контроллера и обнуляет регистр состояния и управления А2. При этом формируется управляющий сигнал "Данные приняты" в шине связи с внешним устройством.

Логика управления контроллера и приемопередатчики шин системного интерфейса выполняют те же функции, что и в контроллере вывода (см. рис. 32).

Рассмотрим работу параллельного интерфейса ввода при реализации программы асинхронного обмена. Алгоритм асинхронного ввода так же прост, как и асинхронного вывода.

1. Процессор проверяет наличие данных в регистре данных контроллера A1.

2. Если данные готовы (логическая 1 в регистре A2), то они передаются из регистра данных A1 на шину данных системного интерфейса и далее в регистр процессора или ячейку памяти микрокомпьютера. Иначе повторяется п. 1.

Пример 2. Фрагмент программы приема байта данных в асинхронном режиме с использованием параллельного интерфейса (контроллер ВУ, рис. 33):

	MOV DX, A2;	номер порта A2 помещаем в DX
m1:	IN AL, DX;	чтение байта из порта A2
	TEST AL, 1;	проверка нулевого разряда состояния регистра A2
	JZ m1;	переход на метку m1 если разряд не нулевой
	MOV DX, A1;	номер порта A1 записываем в DX
	IN AL, DX;	содержимое регистра A1 передаем в регистр AL

В третьей строке выполняется проверка содержимого регистра A2, т. е. признака наличия данных в регистре данных A1. Команда выполняется точно так же, как и в примере 1. Единица в нулевом разряде (содержимое регистра A2) подтверждает, что данные от ВУ записаны в регистр данных контроллера и необходимо переслать их на шину данных. Нуль в знаковом разряде указывает на неготовность данных от ВУ и, следовательно, на необходимость вернуться к проверке готовности.

IN AL, DX – пересылка данных из регистра данных контроллера A1 в регистр процессора AL. Процессор передает в контроллер по шине адреса системного интерфейса адрес A1, сопровождая его сигналом "Ввод". Логика управления контроллера в ответ на сигнал "Ввод" (IORD#) обеспечивает передачу байта данных из регистра данных A1 на шину данных и, в общем случае, но не в IBM-совместимом персональном компьютере с шиной ISA, сопровождает его сигналом "Готовность ВУ", который подтверждает наличие данных от ВУ на шине данных и по которому процессор считывает байт с шины данных и помещает его в указанный регистр. В IBM-совместимом персональном компьютере с шиной ISA процессор считывает байт с шины данных по истечении определенного времени после установки сигнала IORD#. За-

тем логика управления обнуляет регистр состояния и управления A2, формируя тем самым управляющий сигнал для внешнего устройства "Данные приняты". Таким образом завершается цикл ввода данных.

Как видно из рассмотренных примеров, для приема или передачи одного байта данных процессору необходимо выполнить всего несколько команд, время выполнения которых и определяет максимально достижимую скорость обмена данными при параллельной передаче. Таким образом, при параллельной передаче обеспечивается довольно высокая скорость обмена данными, которая ограничивается только быстрым воздействием ВУ.

5.4. Последовательная передача данных

Использование последовательных линий связи для обмена данными с внешними устройствами возлагает на контроллеры ВУ дополнительные, по сравнению с контроллерами для параллельного обмена, функции. Во-первых, возникает необходимость преобразования формата данных: из параллельного формата, в котором они поступают в контроллер ВУ из системного интерфейса микроЭВМ, в последовательный при передаче в ВУ и из последовательного в параллельный при приеме данных из ВУ. Во-вторых, требуется реализовать соответствующий режиму работы внешнего устройства способ обмена данными: синхронный или асинхронный.

5.4.1. Синхронный последовательный интерфейс

Простой контроллер для синхронной передачи данных в ВУ по последовательной линии связи (последовательный интерфейс) представлен на рис. 34 [2, 5, 6].

Восьмиразрядный адресуемый буферный регистр контроллера A1 служит для временного хранения байта данных до его загрузки в сдвиговый регистр. Запись байта данных в буферный регистр с шины данных системного интерфейса производится так же, как и в параллельном интерфейсе (см. разд. 5.3 и рис. 32), только при наличии единицы в одноразрядном адресуемом регистре состояния контроллера A2. Единица в регистре состояния указывает на готовность контроллера принять очередной байт в буферный регистр. Содержимое регистра A2 передается в процессор по одной из линий шины данных системного интерфейса и используется для формирования управляющего сигнала системного интерфейса "Готовность ВУ". При записи очередного байта в буферный регистр A1 обнуляется регистр состояния A2.