

## **5. Организация ввода/вывода в микропроцессорной системе**

Вводом/выводом (ВВ) называется передача данных между ядром ЭВМ, включающим в себя микропроцессор и основную память, и внешними устройствами (ВУ) [5, 6]. Это единственное средство взаимодействия ЭВМ с "внешним миром", и архитектура ВВ (режимы работы, форматы команд, особенности прерываний, скорость обмена и др.) непосредственно влияет на эффективность всей системы. За время эволюции ЭВМ подсистема ВВ претерпела наибольшие изменения благодаря расширению сферы применения ЭВМ и появлению новых внешних устройств. Особенно важную роль средства ВВ играют в управляющих ЭВМ. Разработка аппаратных средств и программного обеспечения ВВ является наиболее сложным этапом проектирования новых систем на базе ЭВМ, а возможности ВВ серийных машин представляют собой один из важных параметров, определяющих выбор машины для конкретного применения.

### **5.1. Программная модель внешнего устройства**

Подключение внешних устройств к системной шине осуществляется посредством электронных схем, называемых контроллерами ВВ (интерфейсами ВВ). Они согласуют уровни электрических сигналов, а также преобразуют машинные данные в формат, необходимый устройству, и наоборот. Обычно контроллеры ВВ конструктивно оформляются вместе с процессором в виде интерфейсных плат.

В процессе ввода/вывода передается информация двух видов: управляющие данные (слова) и собственно данные, или данные-сообщения. Управляющие данные от процессора, называемые также командными словами или приказами, инициируют действия, не связанные непосредственно с передачей данных, например запуск устройства, запрещение прерываний и т. п. Управляющие данные от внешних устройств называются словами состояния; они содержат информацию об определенных признаках, например о готовности устройства к передаче данных, о наличии ошибок при обмене и т. п. Состояние обычно представляется в декодированной форме – один бит для каждого признака.

Регистр, содержащий группу бит, к которой процессор обращается в операциях ВВ, образует порт ВВ. Таким образом, наиболее общая программная модель внешнего устройства, которое может выполнять ввод и вывод, содержит четыре регистра ВВ: регистр выходных дан-

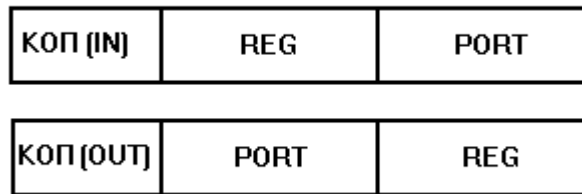
ных (выходной порт), регистр входных данных (входной порт), регистр управления и регистр состояния (рис. 28). Каждый из этих регистров должен иметь однозначный адрес, который идентифицируется дешифратором адреса. В зависимости от особенностей устройства общая модель конкретизируется. Например, отдельные регистры состояния и управления объединяются в один регистр, в устройстве ввода (вывода) имеется только регистр входных (выходных) данных, для ввода и вывода используется двунаправленный порт.



Рис. 28. Программная модель внешнего устройства

Непосредственные действия, связанные с вводом/выводом, реализуются одним из двух способов, различающихся адресацией регистров ВВ.

Интерфейс с изолированными шинами характеризуется отдельной адресацией памяти и внешних устройств при обмене информацией. Изолированный ВВ предполагает наличие специальных команд ввода/вывода, общий формат которых показан на рис. 29. При выполнении команды ввода IN содержимое адресуемого входного регистра PORT передается во внутренний регистр процессора – REG, а при выполнении команды OUT содержимое регистра REG передается в выходной порт PORT. В процессоре могут быть и другие команды, относящиеся к ВВ и связанные с проверкой и модификацией содержимого регистра управления и состояния.



*Рис. 29. Команды ввода/вывода (общий формат)*

Нетрудно заметить, что в этом способе адресное пространство портов ввода и вывода изолировано от адресного пространства памяти, т. е. в ЭВМ один и тот же адрес могут иметь порт ВВ и ячейка памяти. Разделение адресных пространств осуществляется с помощью управляющих сигналов, относящихся к системам ВВ и памяти (MEMRD# – считывание данных из памяти, MEMWR# – запись данных в память, IORD# – чтение порта ВВ, IOWR# – запись в порт ВВ) (# – активный низкий уровень сигналов).

В ЭВМ, рассчитанной на изолированный ВВ, нетрудно перейти к ВВ, отображенному на память. Если, например, адресное пространство памяти составляет 64 кб (ША состоит из 16 бит), а для программного обеспечения достаточно 32 кб, то область адресов от 0 до 32 кб-1 (младшие 8 бит ША) используется для памяти, а от 32 кб до 64 кб-1 (старшие 8 бит ША) – для ввода/вывода. При этом признаком, дифференцирующим обращения к памяти и портам ВВ, может быть старший бит адреса.

Таким образом, интерфейс с общими шинами (ввод/вывод с отображением на память) имеет организацию, при которой часть общего адресного пространства отводится для внешних устройств, регистры которых адресуются так же, как и ячейки памяти. В этом случае для адресации портов ВВ используются полные адресные сигналы: READ – чтение, WRITE – запись.

В операционных системах ЭВМ имеется набор подпрограмм (драйверов ВВ), управляющих операциями ВВ стандартных внешних устройств. Благодаря им, пользователь может не знать многих особенностей ВУ и интерфейсов ВВ, а применять четкие программные протоколы.

## 5.2. Форматы передачи данных

Рассмотрим некоторые общие вопросы, связанные с обменом данными между ВУ и микроЭВМ. Существуют два способа передачи слов информации по линиям данных: параллельный, когда одновременно пересылаются все биты слова, и последовательный, когда биты слова пересылаются поочередно, начиная, например, с его младшего разряда.

Так как между отдельными проводниками шины для параллельной передачи данных существует электрическая емкость, то при изменении сигнала, передаваемого по одному из проводников, возникает помеха (короткий выброс напряжения) на других проводниках. С увеличением длины шины (увеличением емкости проводников) помехи возрастают и могут восприниматься приемником как сигналы. Поэтому рабочее расстояние для шины параллельной передачи данных ограничивается длиной 1–2 м, и только за счет существенного удорожания шины или снижения скорости передачи длину шины можно увеличить до 10–20 м.

Указанное обстоятельство и желание использовать для дистанционной передачи информации телеграфные и телефонные линии обусловили широкое распространение способа последовательного обмена данными между ВУ и микроЭВМ и между несколькими микроЭВМ. Возможны два режима последовательной передачи данных: синхронный и асинхронный.

При синхронной последовательной передаче каждый передаваемый бит данных сопровождается импульсом синхронизации, информирующим приемник о наличии на линии информационного бита. Следовательно, между передатчиком и приемником должны быть протянуты минимум три провода: два для передачи импульсов синхронизации и бит данных, а также общий заземленный проводник. Если же передатчик (например, микроЭВМ) и приемник (например, дисплей) разнесены на несколько метров, то каждый из сигналов (информационный и синхронизирующий) придется посылать либо по экранированному (телевизионному) кабелю, либо с помощью витой пары проводов, один из которых заземлен или передает сигнал, инверсный основному.

Синхронная последовательная передача начинается с пересылки в приемник одного или двух символов синхронизации (не путать с импульсами синхронизации). Получив такой символ (символы), приемник начинает прием данных и их преобразование в параллельный формат. Естественно, что при такой организации синхронной последовательной передачи она целесообразна лишь для пересылки массивов слов, а не

отдельных символов. Это обстоятельство, а также необходимость использования для обмена сравнительно дорогих (четырёх проводных или кабельных) линий связи помешало широкому распространению синхронной последовательности передачи данных.

Асинхронная последовательная передача данных означает, что у передатчика и приемника нет общего генератора синхроимпульсов и что синхронизирующий сигнал не посылается вместе с данными. Как же в таком случае приемник будет узнавать о моментах начала и завершения передачи бит данных? Опишем простую процедуру, которую можно использовать, если передатчик и приемник асинхронной последовательной передачи данных согласованы по формату и скорости передачи.

Стандартный формат асинхронной последовательной передачи данных, используемый в ЭВМ и ВУ, содержит  $n$  пересылаемых бит информации (при пересылке символов  $n$  равно 7 или 8 битам) и 3–4 дополнительных бита: стартовый бит, бит контроля четности (или нечетности) и 1 или 2 стоповых бита (рис. 30, а). Бит четности (или нечетности) может отсутствовать. Когда передатчик бездействует (данные не посылаются на линию), на линии сохраняется уровень сигнала, соответствующий логической 1 [5, 6].



Рис. 30. Формат асинхронной последовательной передачи данных

Передатчик может начать пересылку символа в любой момент времени посредством генерирования стартового бита, т. е. перевода линии в состояние логического 0 на время, точно равное времени передачи бита. Затем происходит передача битов символа, начиная с младшего значащего бита, за которым следует дополнительный бит контро-

ля по четности или нечетности. Далее с помощью стопового бита линия переводится в состояние логической единицы (рис. 30, б). При единичном бите контроля стоповый бит не изменяет состояния сигнала на линии. Состояние логической единицы должно поддерживаться в течение промежутка времени, равного 1 или 2 временам передачи бита.

Промежуток времени от начала стартового бита до конца стопового бита (стоповых бит) называется кадром. Сразу после стоповых бит передатчик может посылать новый стартовый бит, если имеется другой символ для передачи; в противном случае уровень логической единицы может сохраняться на протяжении всего времени, пока бездействует передатчик. Новый стартовый бит может быть послан в любой момент времени после окончания стопового бита, например, через промежуток времени, равный 0.43 или 1.5 времени передачи бита.

В линиях последовательной передачи данных передатчик и приемник должны быть согласованы по всем параметрам формата, изображенного на рис. 8, включая номинальное время передачи бита. Для этого в приемнике устанавливается генератор синхроимпульсов, частота которого должна совпадать с частотой аналогичного генератора передатчика. Кроме того, для обеспечения оптимальной защищенности сигнала от искажения, шумов и разброса частоты синхроимпульсов приемник должен считывать принимаемый бит в середине его длительности. Рассмотрим работу приемника с того момента, когда он закончил прием символа данных и перешел в режим обнаружения стартового бита следующего слова.

Если линия перешла в состояние логического нуля и находится в этом состоянии в течение времени не меньшего, чем половина временного интервала передачи бита, то приемник переводится в режим считывания бит информации. В противном случае приемник остается в режиме обнаружения, так как вероятнее всего это был не стартовый бит, а шумовая помеха. В новом режиме приемник вырабатывает сигналы считывания через интервалы, равные времени передачи бита, т. е. выполняет считывание и сохранение принимаемых бит примерно на середине их передачи. Аналогичным образом будут считаны бит контроля четности и сигнал логической единицы (стоповый бит). Если оказалось, что на месте стопового бита обнаружен сигнал логического нуля, то произошла "Ошибка кадра" и символ принят неправильно. Иначе проверяется, четно ли общее число единиц в информационных битах и бите контроля, и если оно четно, производится запись принятого символа в буфер приемника.

Передний фронт стартового бита сигнализирует о начале поступления передаваемой информации, а момент его появления служит точкой отсчета времени для считывания бит данных. Стоповый бит предоставляет время для записи принятого символа в буфер приемника и обеспечивает возможность выявления ошибки кадра. Наиболее часто ошибки кадра появляются тогда, когда приемник ошибочно синхронизирован с битом 0, который в действительности не является стартовым битом. Если передатчик бездействует (посылает сигнал логической единицы) в течение одного кадра или более, то всегда можно восстановить правильную синхронизацию. Хуже обстоит дело при рассинхронизации генераторов передатчика и приемника, когда временной интервал между сигналами считывания принимаемых битов будет меньше или больше времени передачи бита.

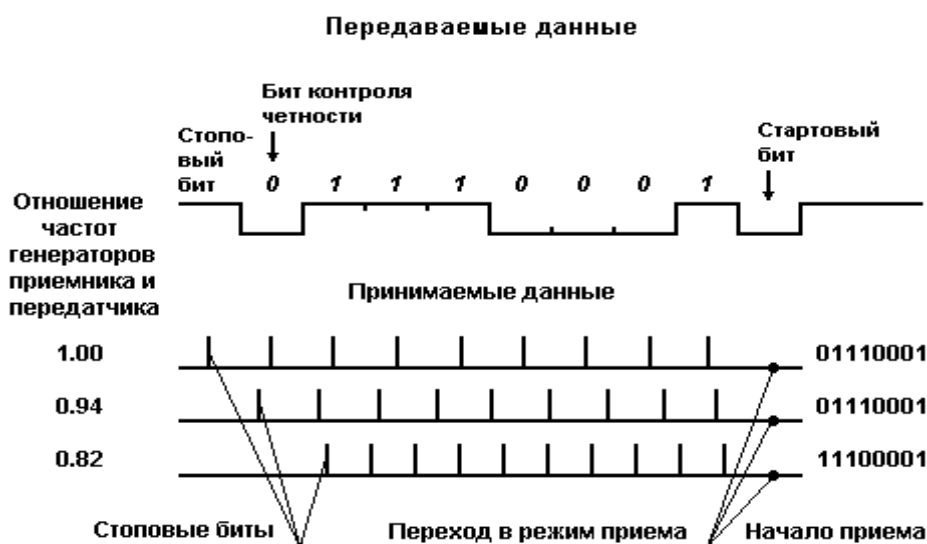


Рис. 31. Ошибка из-за рассинхронизации генераторов передатчика и приемника

Например, если при считывании битов посылки, показанной на рис. 3.3 б, временной интервал между сигналами считывания станет на 6 % меньше, чем время передачи бита, то восьмой и девятый сигналы считывания будут выработаны тогда, когда на линии находится бит контроля четности (рис. 31). Следовательно, не будет обнаружен стоповый бит и будет зафиксирована ошибка кадра, несмотря на правильность принятой информации. Однако при 18%-й рассинхронизации генераторов, когда вместо кода (01110001) приемник зафиксирует код (11100001), никаких ошибок не будет обнаружено – четность соблюдена и стоповый (девятый по порядку) бит равен 1 (см. рис. 31).