

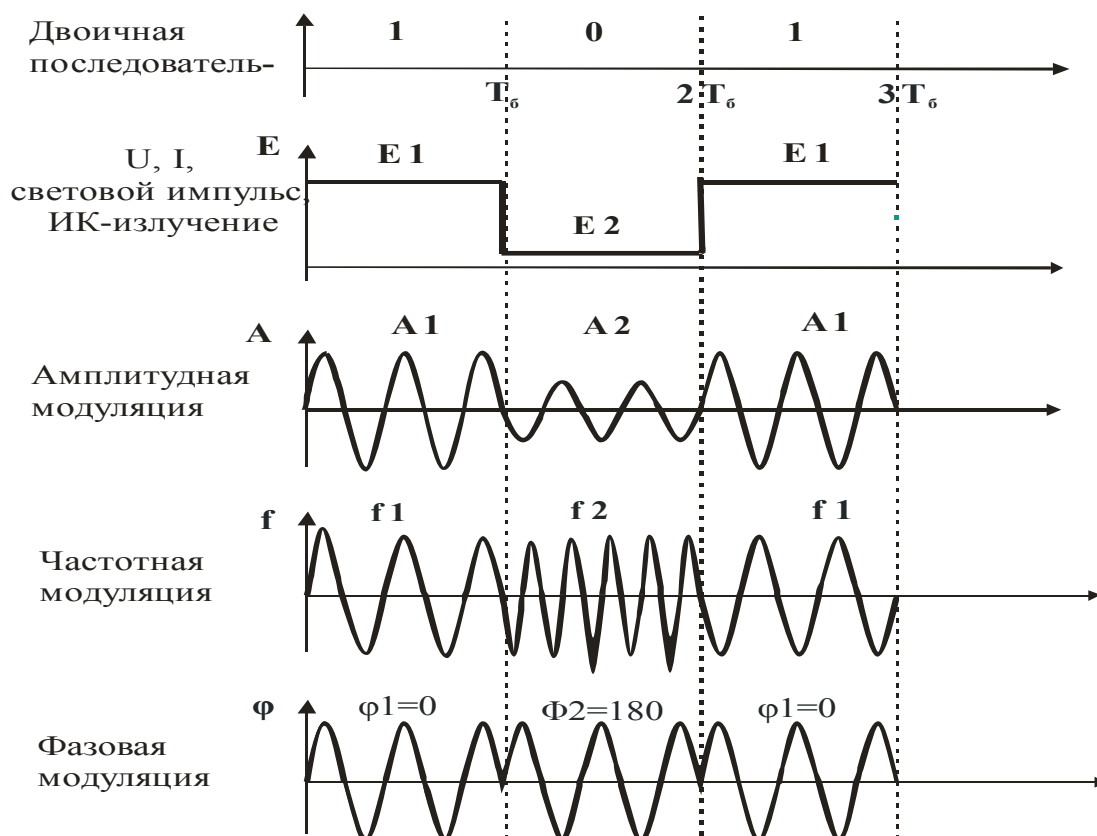
### ***Представление информации в линиях связи интерфейсов на физическом уровне.***

На физическом уровне в линиях связи интерфейсов с параллельной передачей информация представляется, как правило, **дискретными сигналами**.

На физическом уровне в линиях связи интерфейсов с последовательной передачей информация может представляться **дискретными и аналоговыми сигналами**.

В качестве физического представления логического «0» и «1» можно использовать:

- уровни напряжения
- наличие или отсутствие тока в линии
- инфракрасное (ИК) излучение
- световое излучение



Варианты представления информации на физическом уровне

### ***Коды, применяемые в интерфейсах с последовательной передачей данных.***

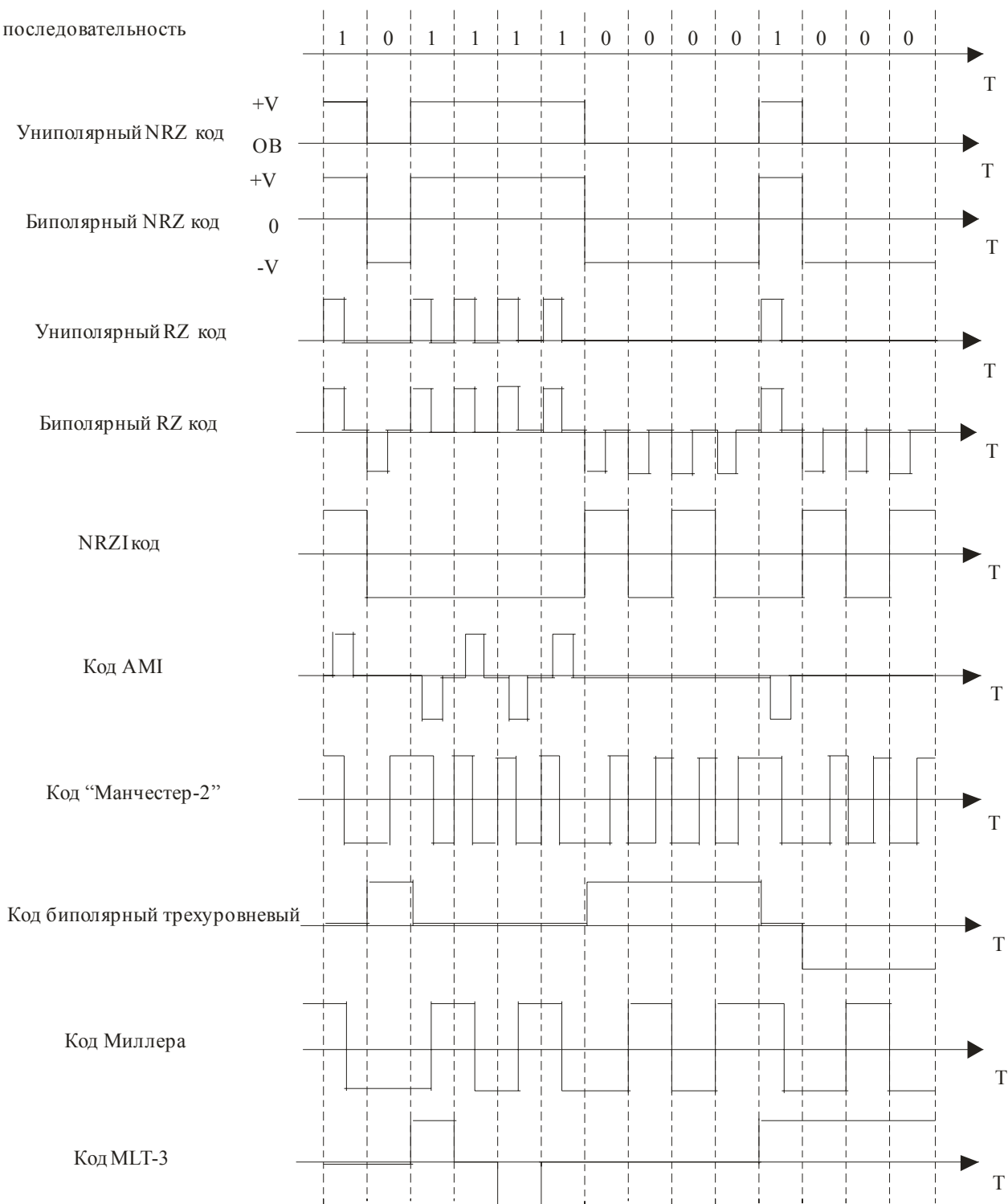
**Кодирование информации** в интерфейсах с последовательной передачей служит удовлетворению следующих требований:

- **повышение скорости** передачи информации (алфавитные коды);
- **обеспечение** максимального числа изменений уровня сигнала в единицу времени для поддержания **битовой синхронизации** ;
- **сужения частотного спектра** сигнала для передачи сигналов по физической среде с ограниченной полосой пропускания;
- удаление постоянной составляющей из спектра частот сигнала, что обеспечивает

гальваническую развязку между устройствами, связанными некоторым интерфейсом;

- простота обнаружения ошибок при передаче информации;
- простота аппаратного решения процедуры самого кодирования;
- обеспечения равномерности спектральной характеристики сигнала передаваемой последовательности для уменьшения создаваемого уровня электромагнитных помех;
- снижение требований по мощности к передатчику.

Двоичная последовательность



Коды, применяемые в интерфейсах с последовательной передачей данных

**Униполярный код без возвращения к нулю** или униполярный NRZ (non return to zero) код образуется из двоичной последовательности, если логический ноль передается уровнем 0 вольт в течение

битового интервала, а логическая единица передается уровнем  $+U$  вольт в течение битового интервала.

**Биполярный код без возвращения к нулю** или биполярный NRZ (non return to zero) код образуется из двоичной последовательности, если логический ноль передается уровнем  $-U$  вольт в течение битового интервала, а логическая единица передается уровнем  $+U$  вольт в течение битового интервала.

Основные плюсы такого кодирования: минимальная полоса частотного спектра при конкретной скорости передачи данных и простота кодирования (декодирования). Недостатки NRZ кода: не обеспечивает битовой синхронизации, гальванической развязки и высокой помехозащищенности

**Униполярный код с возвращением к нулю** или униполярный RZ (return to zero) код образуется из двоичной последовательности, если логический ноль передается уровнем  $0$  вольт в течение битового интервала, а логическая единица передается уровнем  $+U$  вольт в течение первой половины битового интервала и уровнем  $0$  вольт в течение второй половины битового интервала.

**Биполярный код с возвращением к нулю** или биполярный RZ код образуется из двоичной последовательности, если логический ноль передается уровнем  $-U$  вольт в течение первой половины битового интервала и уровнем  $0$  вольт в течение второй половины битового интервала, а логическая единица передается уровнем  $+U$  вольт в течение первой половины битового интервала и уровнем  $0$  вольт в течение второй половины битового интервала.

Основные плюсы биполярного RZ кодирования: обеспечение битовой синхронизации и простота кодирования (декодирования). Недостаток RZ кода: не обеспечивает гальванической развязки и высокой помехозащищенности, требует в два раза шире полосу частот, чем код NRZ, при одинаковой битовой скорости.

**Код без возвращения к нулю с инверсией** (БВНИ), иначе NRZI-код (non return to zero with inversion).

При формировании **кода без возвращения к нулю с инверсией по нулям** логическая «1» передается отсутствием изменения уровня напряжения предшествующего бита; логический «0» передается изменением уровня напряжения предшествующего бита на противоположный. При формировании **кода без возвращения к нулю с инверсией по единицам** логический «0» передается отсутствием изменения уровня напряжения предшествующего бита; логическая «1» передается изменением уровня напряжения предшествующего бита на противоположный.

Достоинства кода: достаточно легко реализуется аппаратно и требует минимальную полосу пропускания, как и код NRZ, частично обеспечивает битовую синхронизацию. Недостатки кода: не обеспечивает полную битовую синхронизацию и гальваническую развязку устройств.

**Код AMI (alternative mark inversion).** Код, предложенный Баркером для цифровой

передачи речевых сигналов (1962 г.). При формировании кода АМІ логический «0» передается нулевым уровнем напряжения в течение битового интервала, логическая «1» передается импульсом напряжения либо  $+V$  вольт, либо  $-V$  вольт, причем все время меняется полярность этого напряжения, импульс напряжения длительностью  $T_B/2$  и расположен в середине битового интервала.

Основное достоинство кода заключается в том, что он может передаваться по линии связи с трансформаторной развязкой (требование цифровой телефонии), т.к. в спектре частот нет нулевой частоты, но в “чистом” виде на практике код Баркера не используется, т.к. не поддерживает битовой синхронизации при передаче длинных двоичных последовательностей из нулей. Практически в линиях цифровой телефонии используется модифицированный код АМІ: **код биполярный с замещением четырех нулей (B4ZS-bipolar with four-zero substitution) или иначе код высокой плотности биполярный порядка 3 (HDB3 – High Density Bipolar 3)**. Модификация кода АМІ заключается в том, что, если непрерывная последовательность логических «0» равна и больше четырех, то первая тетрада нулей (0000) заменяется двоичной последовательностью 000V, где V - логическая «1» с нарушением правил кодирования АМІ, т.е. импульс с сохранением полярности предшествующей последней логической «1». Вторая и последующие тетрады из нулей заменяются двоичной последовательностью B00V, где B - логическая «1» с соблюдением правил кодирования АМІ, т.е. импульс с изменением полярности предшествующей последней логической «1», а V - логическая «1» с нарушением правил кодирования АМІ, т.е. импульс с сохранением полярности предшествующей последней логической «1».

Самосинхронизирующийся **Код Манчестер-2** представляет каждый бит двоичной последовательности сменой уровня напряжения в середине битового интервала.

Например, в интерфейсе МИК (мультиплексный информационный канал), если низкий уровень напряжения сменяется высоким, то передается логический «0», если высокий уровень напряжения сменяется низким, то передается логическая «1».

Возможно и инверсное представление такого кодирования, например, как в ЛВС ETHERNET (технологии 10BASE-2, 10BASE-5, 10BASE-T )

Достоинства кода: битовая синхронизация по единицам и нулям, отсутствие постоянной составляющей в сигнале (возможна и на практике обеспечивается гальваническая развязка), высокая помехозащищенность. Недостаток заключается в том, что ему нужна достаточно широкая полоса частот.

**Код биполярный трехуровневый.** В нем логическая «1» представляется нулевым уровнем напряжения; логический «0» - уровнем напряжения  $+V$  вольт или  $-V$  вольт, причем полярность меняется всякий раз, когда перед логическим «0» следует нечетное количество логических «1».

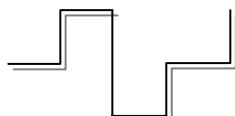
**Код Миллера.** Логическая «1» отображается сменой полярности импульса в середине битового интервала, логический «0» - не вызывает смены полярности, кроме

как на границе соседних (смежных) битовых интервалов, соответствующих логическим «0».

### **Алфавитный код или код « $m$ бит $n$ периодов» ( $mBnS$ ).**

В ЛВС ETHERNET по технологии 100Base-T4 используется алфавитный код 4B3S (4 бита 3 периода). Логическая двоичная последовательность разбивается на группы по 4 бита и каждая тетрада кодируется троичным кодом (+V, 0, -V) длиной три периода.

Двоичная тетрада имеет 16 различных кодовых наборов, в то время как троичная триада может закодировать 27 различных кодовых набора. Переход к троичной системе счисления уменьшает длину последовательности и, следовательно, повышает скорость передачи.



### **Избыточные коды**

При передаче информации по оптоволокну в ЛВС FDDI или по витой паре в сети 100Base-TX используются *избыточные коды* 4B5S. Исходная логическая последовательность разбивается на тетрады (16 различных кодовых комбинаций). При кодировании используют 5 бит для передачи одной тетрады, т.е. вместо кода из 4-х бит передается избыточный код из 5 бит. В этом случае можно закодировать 32 различные кодовые комбинации, но из них выбирают лишь такие, которые содержат не менее 2-х единиц, все остальные кодовые комбинации считаются запрещенными.

#### **Избыточный код 4B5S**

Исходный	Результу		Исходный	Результир
0000	11110		1000	10010
0001	01001		1001	10011
0010	10100		1010	10110
0011	10101		1011	10111
0100	01010		1100	11010
0101	01011		1101	11011
0110	01110		1110	11100
0111	01111		1111	11101

Достоинства: - решается задача битовой синхронизации,  
 - избыточные запрещенные кодовые комбинации используются для целей управления (например, чтобы сообщить, что передача отсутствует).

Недостатки: избыточность увеличивает рабочую частоту для сохранения скорости передачи.

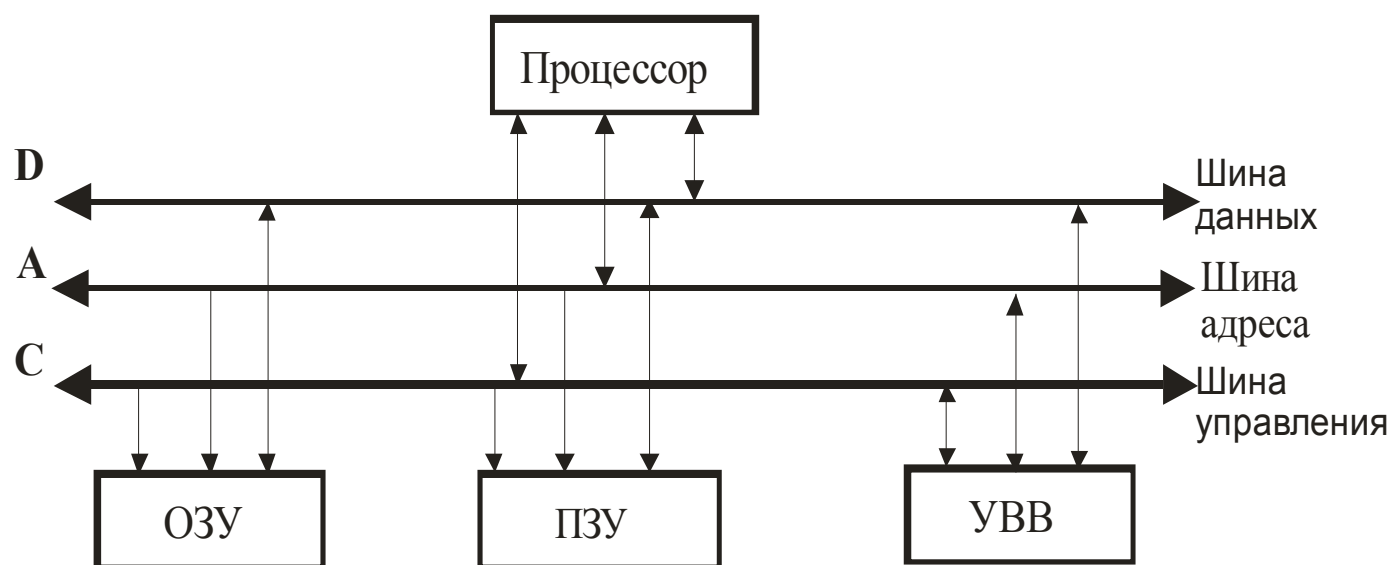
### Код MLT-3.

В коде MLT-3 используются 3 уровня напряжения  $+V$ ,  $0$ ,  $-V$ . Логический «0» передается отсутствием изменения уровня напряжения в линии, логическая «1» передается циклическим изменением уровня напряжения:  $+V, 0, -V, 0, +V$ .

Применяется в ЛВС ETHERNET в технологии 100BASE-TX.

### 4 лекция

Проблема синхронизации интерфейсов при использовании параллельных принципов работы.



Шины данных, адреса и управления многоразрядные

Все сигналы, которыми обмениваются устройства по линиям интерфейса, во избежание недоразумений должны быть некоторым образом синхронизированы. Рассмотрим, к примеру, взаимодействие между процессором и ОЗУ, смотри рис 1.1.6. Так как, обмен информацией производится по инициативе процессора, то он в этом случае является ведущим (активным) устройством (master-устройством), а ОЗУ – ведомым (пассивным) устройством (slave-устройством). Временные диаграммы, поясняющие взаимодействие этих двух устройств в случае записи информации из процессора в ОЗУ в асинхронном режиме, приведены на рис. 1.3.1.

На адресной магистрали (шине) ведущее устройство (в данном случае, процессор) выставляет адрес, принадлежащий адресному пространству ОЗУ и строб адреса СА, по которым однозначно выбирается в качестве ведомого устройства устройство ОЗУ из

числа возможных для этого примера ведомых устройств (ОЗУ, ПЗУ, УВВ). Этот адрес ведущее устройство либо удерживает на магистрали адреса всё время взаимодействия, либо он запоминается ведомым устройством по стробу адреса СА. Далее ведущее устройство (в нашем случае, процессор) выставляет с некоторой задержкой данные на магистраль данных и затем сигнал синхронизации ведущего (активного) СИА на магистрали (шине) управления. Установление (падающий фронт) управляющего сигнала СИА означает для ведомого устройства, что данные на магистрали данных установлены ведущим устройством и истинны.

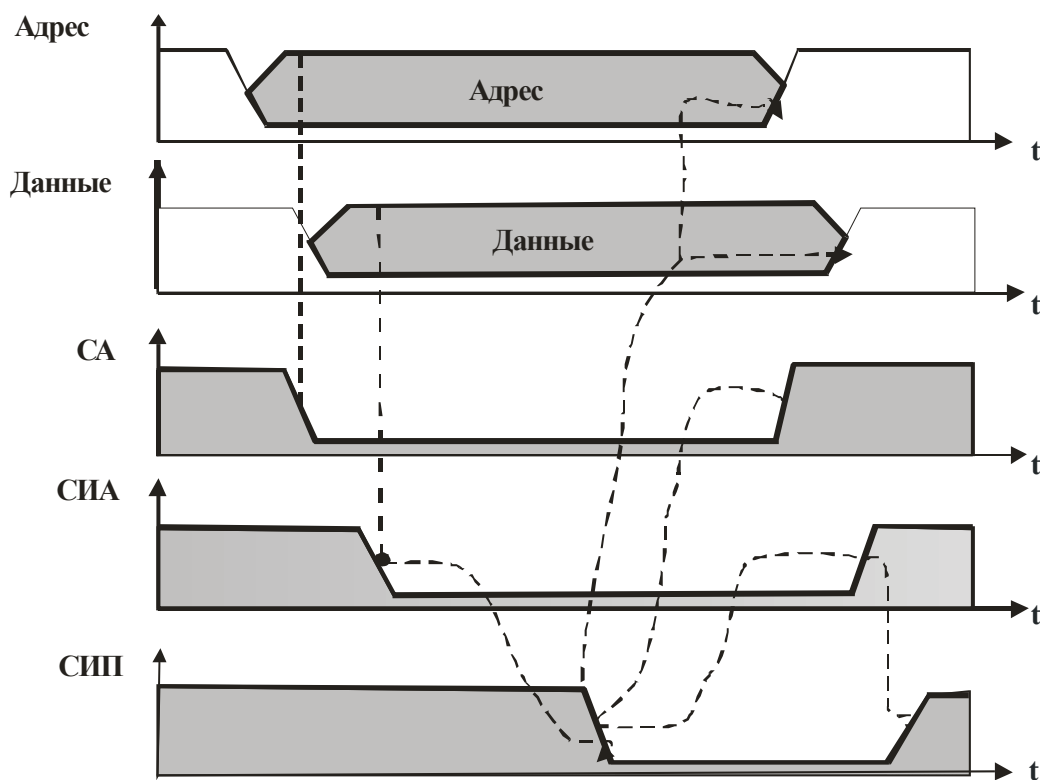


Рис 1.3.1. Временные диаграммы записи данных процессором в ОЗУ

По истечении некоторого произвольного интервала времени, которое определяется конкретным быстродействием ведомого (пассивного) устройства, в нашем случае ОЗУ, последнее должно считать данные с магистрали данных во внутреннюю память ОЗУ и затем установить (падающий фронт) сигнал синхронизации ведомого (пассивного) устройства, СИП, что означает для ведущего устройства, что запись в ОЗУ по существу закончена. После этого ведущее устройство, процессор, может сбросить управляющий сигнал СА (нарастающий фронт сигналов), сбросить адрес и данные с соответствующих магистралей, а затем сбросить сигнал СИА. В ответ ведомое устройство после сброса СИА с некоторой задержкой сбрасывает управляющий сигнал СИП, сообщая ведущему устройству, что цикл записи окончен, интерфейс приведен в исходное состояние (уровни напряжения во всех линиях не активны), и ведущее

устройство может начать новый цикл.

Таким образом, проблема синхронизации в данном случае решается выполнением обмена информацией между ведущим и ведомым устройствами с использованием сигналов квитирования (“рукопожатия”), роль которых в этом примере выполняют управляющие сигналы СИА и СИП.

Классическими примерами интерфейсов с асинхронным параллельным обменом могут служить интерфейсы VME, VXI, SCSI, IEEE-488. В современных высокоскоростных системных интерфейсах, таких как PCI, PCI-X, PXI, Compact PCI, применяется синхронный параллельный обмен, особенности которого рассматриваются ниже в разделах 2.1-2.4. Все сигналы, которыми обмениваются устройства по линиям интерфейса, во избежание коллизий должны быть синхронизированы. Рассмотрим, к примеру, взаимодействие между процессором и ОЗУ. Так как, обмен информацией производится по инициативе процессора, то он в этом случае является ведущим (активным) устройством, а ОЗУ – ведомым (пассивным) устройством. Временные диаграммы, поясняющие взаимодействие этих двух устройств в случае записи информации из процессора в ОЗУ в асинхронном режиме, приведены на рис.

На адресной магистрали (шине) ведущее устройство (в данном случае, процессор) выставляет адрес, принадлежащий адресному пространству ОЗУ и строб адреса СА, по которым однозначно выбирается в качестве ведомого устройства устройство ОЗУ из числа возможных для этого примера ведомых устройств (ОЗУ, ПЗУ, УВВ). Этот адрес ведущее устройство либо удерживает на магистрали адреса всё время взаимодействия, либо он запоминается ведомым устройством по стробу адреса СА. Далее ведущее устройство (в нашем случае, процессор) выставляет с некоторой задержкой данные на магистраль данных и затем сигнал синхронизации ведущего (активного) СИА на шине управления. Установление (падающий фронт) управляющего сигнала СИА означает для ведомого устройства, что данные на магистрали данных установлены ведущим устройством и истинны.

По истечении некоторого произвольного интервала времени, которое определяется конкретным быстродействием ведомого (пассивного) устройства, в нашем случае ОЗУ, последнее должно считать данные с магистрали данных во внутреннюю память ОЗУ и затем установить (падающий фронт) сигнал синхронизации ведомого (пассивного) устройства, СИП, что означает для ведущего устройства, что запись в ОЗУ по существу закончена. После этого ведущее устройство, процессор, может сбросить управляющий сигнал СА (нарастающий фронт сигналов), сбросить адрес и данные с соответствующих магистралей, а затем сбросить сигнал СИА. В ответ ведомое устройство после сброса СИА с некоторой задержкой сбрасывает управляющий сигнал СИП, сообщая ведущему устройству, что цикл записи окончен, интерфейс приведен в исходное состояние (уровни напряжения во всех линиях не активны), и ведущее



устройство может начать новый цикл.

Таким образом, проблема синхронизации в данном случае решается выполнением обмена информацией между ведущим и ведомым устройствами с использованием сигналов квитирования, роль которых в этом примере выполняют управляющие сигналы СИА и СИП.

Классическими примерами интерфейсов с асинхронным параллельным обменом могут служить интерфейсы VME, VXI, SCSI, IEEE-488. В современных высокоскоростных системных интерфейсах, таких как PCI, PCI-X, PXI, Compact PCI, применяется синхронный параллельный обмен, особенности которого рассмотрим позже.