

Микропроцессорные системы ЭФУ

Конспект лекций

11.04.2021

Кафедра электрофизических установок МИФИ

Александр Коршунов

Оглавление

ВВЕДЕНИЕ	1
Система управления Большого адронного коллайдера.....	1
1. ПРИНЦИПЫ ПОСТРОЕНИЯ МИКРОПРОЦЕССОРНЫХ СИСТЕМ.....	6
1.1. Агрегатно-модульные системы	6
1.2. Стандартные интерфейсы	9
1.3. Развитие магистрально-модульных систем.....	12
1.4. Интерфейсная система SCI	14
2. ИЗМЕРИТЕЛЬНЫЙ ИНТЕРФЕЙС МЭК 625.1.....	17
2.1. Структура интерфейса.....	17
2.2. Состав и назначение шин интерфейса	19
2.3. Интерфейсные функции	21
2.4. Пример реализации интерфейсной функции	23
2.5. Сопряжение с микропроцессорной магистралью	28
3. СИСТЕМА VME	29
3.1. Структура системной магистрали	29
3.2. Шина передачи данных	31
3.3. Производительность магистрали.....	37
3.4. Арбитраж магистрали.....	39
3.5. Шина прерываний.....	42
3.6. Служебная шина.....	44
4. СИСТЕМА WorldFIP.....	47
4.1. Общая конфигурация WorldFIP	48
4.2. Слой физического уровня.....	48
4.3. Слой связи данных.....	51
4.4. Интерфейсы прикладного - физического уровней	53

4.5. Передача данных	55
4.6. Механизм доступа к шине	56
4.7. Таблицы арбитража шины	58
4.8. Режимы работы WorldFIP	62
4.9. Прикладной слой	69
5. СИСТЕМА PROFIBUS	75
5.1. Структура PROFIBUS	75
5.2. Технология передачи данных	77
5.3. Передача данных	81
5.4. Протокол передачи данных	84
5.5. Топология сети	88
6. СИСТЕМА SIMATIC ET 200	90
6.1. Система распределенного ввода-вывода SIMATIC ET 200SP	90
6.2. Отказоустойчивые системы автоматизации и отказоустойчивые модули	94
6.3. SIMATIC F-система ET 200SP	95
6.4. Системные компоненты	98
7. СИСТЕМА NI CompactRIO	101
7.1. Архитектура CompactRIO	101
7.2. Конфигурации системы	107
7.3. Базовая архитектура CompactRIO	109
7.4. Режимы программирования CompactRIO	111
7.5. Технология FPGA	115

ВВЕДЕНИЕ

Система управления Большого адронного коллайдера

Большой адронный коллайдер (БАК) CERN длиной 26 659 м получает пучки заряженных частиц от комплекса машин, являющихся предварительными ускорителями БАК. Эти машины - Linac2 (H⁺), Linac3 (Pb⁵⁴⁺), PS-бустер, PS и SPS - существуют уже много лет. На PSB, PS и SPS проводятся собственные физические эксперименты. Все эти машины были доработаны так, чтобы они были в состоянии не только обеспечить необходимые для БАК параметры пучка, но и продолжить, когда они не используются для БАК, проведение на них собственных экспериментов. На рис. 1 показаны эти машины и основные площадки для проведения экспериментов.

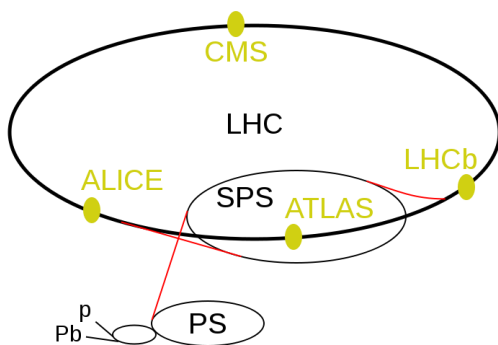


Рис.1. Ускорители CERN

ALICE (A Large Ion Collider Experiment) - для изучения кварк-глюонной плазмы в столкновениях тяжёлых ионов свинца

ATLAS (A Toroidal LHC Apparatus) и CMS (Compact Muon Solenoid) - детекторы общего назначения, предназначены для поиска бозона Хиггса и «нестандартной физики», в частности, тёмной материи.

LHCb (The Large Hadron Collider beauty experiment) - для исследования физики *b*-кварков, что позволит лучше понять различия между материей и антиматерией

Энергетическая компоновка БАК чрезвычайно сложна, она включает 1232 главных дипольных магнита, приблизительно 450 квадрупольных магнитов и несколько тысяч магнитов коррекции, приводящиеся в действие 1612 электрическими схемами. Питающий ток колеблется от 60 А (для малых магнитных корректоров) до 12 кА для главных дипольных и квадрупольных магнитов. Энергия, сохраненная в сверхпроводящих магнитах, составляет приблизительно 10 ГДж. Для установки и ввода в действие оборудования БАК в туннеле были выполнены и отлажены около 80000 коммутаций тока высокого напряжения.

Архитектура системы управления БАК

Управление БАК и его предварительными ускорителями осуществляется из единой диспетчерской с помощью объединенной системы управления.

В CERN существуют две сети Ethernet: сеть общего назначения General Purpose Network (GPN) и техническая сеть Technical Network (TN). Первая развернута во всех зданиях CERN с доступом в интернет, вторая предназначена для управления оборудованием и не имеет доступа к интернету. Эта сеть, разбитая на подсети и основанная на гигабитном Ethernet, использует волокно-оптические каналы связи.

Как показано на рис. 2, у системы управления БАК имеется 3 иерархического слоя управления, объединенного технической сетью CERN с использованием TCP-IP протокола.

Доступ к оборудованию БАК осуществляется через следующие интерфейсные системы:

- Фронтальные (front-end) компьютеры VME. 350 VME-компьютеров добавлено к уже существующим для комплекса PS и SPS 300 крейтам, работающих в режиме реального времени с высокоэффективными средствами сбора данных, которые представляют собой разнообразные коммерческие или сделанные в CERN модули ввода/вывода. Эти компьютеры работают под управлением операционных систем реального времени LynxOS, так же как и на синхротронах PS и SPS, и Red Hat Linux. Главным образом, для повышения надежности они бездисковые и загружаются по сети. Как правило, фронтальные VME используются для инструментовки пучка БАК и системы блокировки (interlock) пучка (системы, принимающей в случае нештатной ситуации решение о сбросе пучка).
- Фронтальные промышленные PC с ОС Linux, которые для уменьшения затрат, когда нужна простая функциональность, заменяют крейты VME. Приблизительно 250 модульных PC установлены в зданиях БАК. Они, как правило, содержат одну или две интерфейсные платы промышленной шины (fieldbus) и плату синхронизации PCI и связаны с оборудованием в туннеле и подземных областях оптическими кабелями. Через PC-шлюзы они соединены с преобразователями мощности и системами защиты сверхпроводимости БАК.

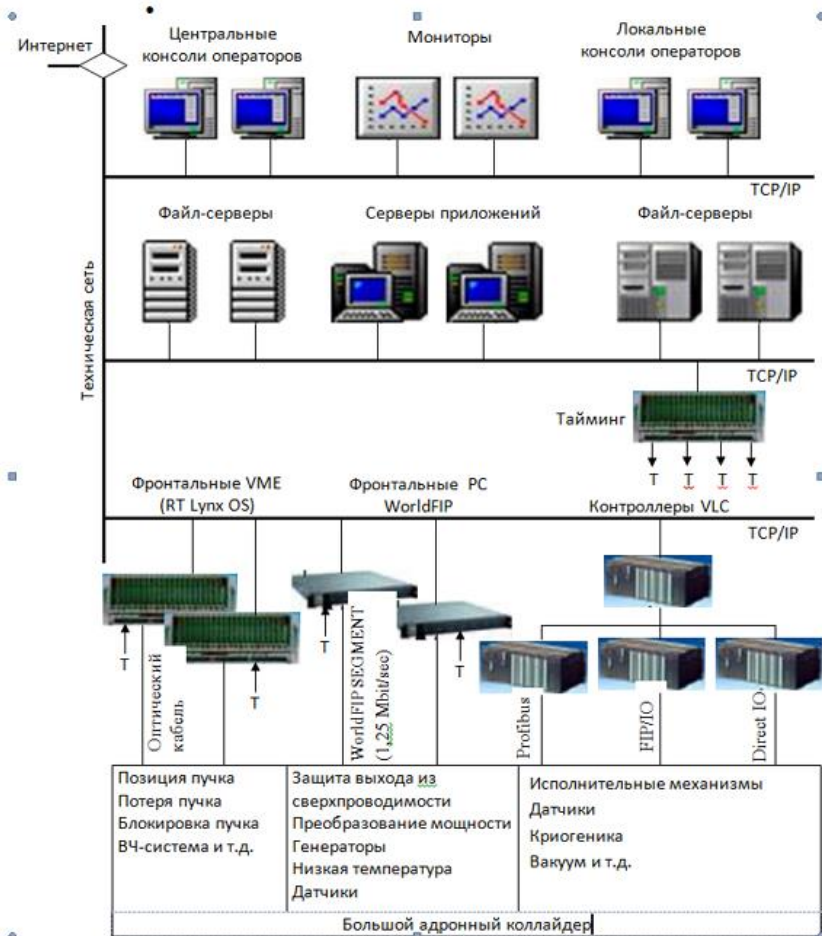


Рис.2. Архитектура системы управления

- Фронтальные компьютеры PLC фирмы Siemens, традиционно используемые для управления промышленным оборудованием. В некоторых случаях доступ к PLC осуществляется через VME. В БАК они используются для машинной криогеники, блокировок, вакуума, модуляции и вывода пучка.
- В CERN для БАК используются две промышленные шины: WorldFIP и Profibus:

- WorldFIP выбрана, когда требуется временное разрешение меньше 10 μ s, надежность при работе в сложной окружающей среде и высокие скорости передачи данных (1 Мбит/с и 2.5 Мбит/с). Она используется для преобразователей мощности БАК, системы защиты выхода из сверхпроводимости Quench Protection System (QPS), контроля температуры криогеники. Инфраструктура WorldFIP состоит из 350 км кабеля, приблизительно 40000 пассивных и 1100 активных модулей, среди которых имеются специально предназначенные для работы в жесткой радиационной окружающей среде.
- Шина Profibus выбрана из-за ее простой настройки, большого разнообразия систем удаленного ввода/вывода и интеграции с PLC. Ее производительность позволяет использование в качестве инструментальной шины (Profibus-PA) с широким диапазоном продуктов инструментровки Profibus таких, как стандартные средства связи. Она используется для быстрых поворотных магнитов, охлаждения и вентиляции, вакуума, криогеники, управления магнитами и блокировок.

Серверы

- Серверы UNIX-приложений среднего уровня системы управления БАК используются для запуска программного обеспечения, хостинга эксплуатационных программ и файлов с данными, а также для выполнения определенных сервисов (веб-службы для операций, дисплеи, серверы систем сбора данных SCADA, серверы базы данных и т.д.). Они работают под управлением ОС Linux. Выбор обоснован надежностью аппаратных средств и стоимостью их реализации. Для гарантирования целостности данных и устранения ошибок используется RAID массивы. Кроме нескольких больших SUN-компьютеров, используемых для баз данных, обычными серверами являются HP машины типа ProLiant. Приблизительно 50 из них располагаются в управляющем центре CERN Control Center (CCC).
- Система центрального тайминга, базирующаяся на VME, работает как сервер данных для отсчета времени и синхронизации всех машин, а также временной отметки данных БАК.

Процессы, запущенные в этих серверах, общаются с фронтальными машинами доступа к оборудованию, используя различные механизмы и протоколы, такие как Common Object Request Broker Architecture (CORBA) и TCP-Modbus.

Пульты операторов и мониторы

Как показано на рис. 2, верхний слой системы управления БАК состоит из операционных пультов и мониторов, которые располагаются в CCC или в технических зданиях. Эти пульты на базе PC с одним или 3 дисплеями работают под управлением Linux. Некоторые из них используют Windows, главным образом, для управления сервисами CERN общего назначения. Пульты представляют собой интерактивные графические приложения, которые запускаются на серверах. Сорок из них расположены в CCC, и столько же расположено в технических зданиях. Они все связаны технической сетью CERN.

Мониторы показывают статусную информацию о процессах в машине и пучке. Она формируется в управляющем центре и распределяется по сети Ethernet, а также через сети абонентского телевидения.

Две промышленные шины, используемые в ускорителе БАК, а именно, WorldFIP и Profibus, являются частью трех интерфейсных систем, включая VME, рекомендованных для применения в CERN. Они обе поддерживаются внутренним обслуживанием CERN. Эти шины, как правило, позволяют более длинные расстояния и более надежные протоколы, чем Ethernet. Кроме того, они - единственные средства соединения оборудования, расположенного в туннеле БАК или других радиоактивных зонах.

Шина WorldFIP

Табл. 1 суммирует инсталляционные потребности WorldFIP в БАК. Она выбирается, когда требуются обеспечить:

- Временное разрешение (до 10 мкс) для:
 - управления в реальном времени и синхронизации оборудования БАК,
 - распределения времени высокой точности,
 - управления периодическими данными.
- Надежность работы в опасной окружающей среде и в особенности, когда необходима:
 - устойчивость к электромагнитному шуму (уровень 3),
 - хорошая устойчивость к высоким радиационным уровням (основывается на радиационно-стойких полупроводниках и индуктивной связи, гарантирующей гальваническую изоляцию).
- Скорость передачи данных:

- WorldFIP используется для управления, в основном, распределенными системами БАК на высоких скоростях передачи данных (1 Мбит/с и 2,5 Мбит/с),
- возможный высокий коэффициент нагрузки сети (70 - 80% сетевой полосы пропускания).

Табл.1. Использование шины WorldFIP для БАК

Система ЛНС	Точность времени (мс)	Длина кабеля (км)	Скорость передачи данных
Защита магнитов	1	61	1 Мбит/с
Преобразование мощности	0,01	45	2.5 Мбит/с
Управление пучком	500	50	31.25 Кбит/с
Радиочастота	1	5	1 Мбит/с
Охлаждение	500	90	1 Мбит/с
Радиационный контроль	1000	44	31.25 Кбит/с

Шина Profibus

Profibus была выбрана для использования с оборудованием в БАК, таким как быстрые кикер-магниты, охлаждение и вентиляция, вакуум, криогеника, блокировка мощности.

Главные причины выбора Profibus для этих применений:

- Надежность протокола и простота конфигурации.
- Большое разнообразие удаленных систем ввода/вывода, доступных с Profibus.
- Простая интеграция с PLC фирмы Siemens.
- Возможность стойкого к радиации удаленного ввода/вывода на Profibus.
- Использование в качестве инструментальной шины (Profibus-PA) с широким диапазоном предложений продуктов инструментки Profibus как стандартных средства связи.

1. ПРИНЦИПЫ ПОСТРОЕНИЯ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

1.1. Агрегатно-модульные системы

Наиболее рациональный принцип построения любых технических систем - принцип агрегатирования. Его сущность заключается в том, что система выполняется как агрегат, состоящий из независимых функциональных блоков - модулей.

Модульный принцип разработки систем изобретен в глубокой древности, когда люди научились складывать дома и стены из блоков. В технике модульные структуры и стандарты на них впервые появились в XIX веке на железной дороге (ширина рельсовой колеи, габариты вагонов, стрелки, размеры сцепок и в дальнейшем - воздушные и электрические соединители). В начале XX века модульность оказалась удобной для электротехнических устройств. В 1922 г. фирма Bell Systems для размещения реле и других деталей телефонных станций сконструировала блоки с передней панелью шириной 19 дюймов (482,6 мм), высота которых могла наращиваться с шагом 13/4 дюйма (82,05 мм). Из блоков могли компоноваться стойки, которые оказались удобными и для размещения там электронной аппаратуры: вначале ламповой, затем транзисторной и, наконец, выполненной на основе современной элементной базы.

Стоит отметить и неприятную сторону стандартизации модулей, прежде всего живучесть стандарта, усугубляющую последствия неадекватности, заложенную при его разработке. Когда в 1825 г. английский парламент не позволил Джорджу Стефенсону сделать железнодорожную колею шире колеи конных дилижансов, никто не предвидел, что людям, перемещающимся на дальние расстояния, будет тесно в купе комфортабельных вагонов. Или введение в отечественные стандарты шага модульного элемента блока кратного 20 мм а не 17,2 мм, следующего из дюймовой единицы измерения, привело к несоответствию нашей аппаратуры с международным стандартом и, как следствие, к отставанию отечественного приборостроения.

В каждой отрасли складывалась своя методология применения и развития агрегатно-модульного принципа. В приборостроении России, например, еще в начале 60-х годов прошлого века была создана Государственная система приборов (ГСП). Были разработаны ее общая структура и принципы энергетической, информационной, метрологической, конструктивной и эксплуатационной совместимости технических средств ГСП.

В ГСП для целей АСУТП, АСУП, АСНИ и САПР можно выделить три группы: общепромышленные технические средства АСУ, технические средства для локальных систем автоматического регулирования и средства вычислительной техники. По виду используемой энергии носителя информации средства ГСП подразделяются на электрические, пневматические и гидравлические. По виду выполняемой функции средства ГСП объединяются в следующие группы: измерение физических величин, преобразование сигналов и передача информации по каналам связи, сбор, хранение и обработка информации, воздействие на управляемый объект.

Важное место в структуре ГСП занимают средства измерительной и вычислительной техники. На первом этапе были созданы средства агрегатных систем электроизмерительной техники АСЭТ. Следующим этапом стала разработка агрегатной системы средств вычислительной техники АСВТ. К четвертому поколению этого направления относится, например, комплекс технических средств для локальных информационно-управляющих систем КТС ЛИУС-2 с интерфейсной системой ИК1, который был разработан в СССР в конце 80-х НПО САУ, г. Харьков, на базе интегральных схем с повышенной интеграцией и микропроцессоров.

В качестве модулей средств измерительной и вычислительной техники используются конструктивно законченные элементы, например, память, устройства ввода-вывода, АЦП и ЦАП, генераторы функций, мультиплексоры, регистры и т. п. Многообразие систем, решающих определенные задачи, достигается путем использования различных сочетаний, комбинаций модулей. Открытые системы предполагают наличие возможности наращивания структуры и в процессе их эксплуатации. Модули объединяются в группы, называемые, например, крейтами, которые, в свою очередь, размещаются в стойках, упорядочивая таким образом размещение системы в целом. Управление работой системы осуществляют специальные модули, называемые контроллерами, которые координируют и контролируют действия отдельных устройств.

Главной идеей при разработке модулей для агрегатной технической системы является понятие обобщенной совместимости, которое можно разделить в порядке важности на следующие основные категории совместимости: функциональную, конструктивную и программную.

Функциональная совместимость модулей предусматривает принципиальную возможность их совместной работы для выполнения заданной функциональной цели системы. При этом может оказаться, что используются модули, имеющие несогласованные входные и выходные параметры. Для преодоления таких трудностей применяются различного рода согласующие устройства часто разового применения, что особенно затрудняет их изготовление. При этом неоправданно возрастает стоимость проектирования, изготовления и наладки, а также снижается надежность всей системы. Это приводит к необходимости более жесткой регламентации входных и выходных параметров модулей, сигналов и команд управления.

В наиболее законченном виде функциональная совместимость разработана в цифровых устройствах контроля, управления и средствах вычислительной техники, где ее называют информационной совместимостью

(информационным интерфейсом). К ней относится согласованность входных и выходных сигналов модулей по видам и номенклатуре, информативным параметрам и уровням. Для информационных связей модулей применяют сигналы нескольких разновидностей: информационные, адресные, управляющие, специальные.

Параметры, определяющие информационную совместимость, обычно оговариваются в технических условиях на систему в целом и на каждый модуль этой системы в отдельности.

Конструктивная совместимость модулей охватывает, прежде всего, требования, оговоренные в рабочих чертежах на модуль: габаритные и присоединительные размеры, оформление внешнего вида, особенности входов и выходов модуля, подключение источников питания и ряд других требований. Наиболее важными являются те конструктивные признаки модуля, которые позволяют включить его в систему без дополнительных согласующих элементов, что достигается обычно разработкой единой конструктивной базы (конструктивного интерфейса).

Программная совместимость модулей в наибольшей степени проявляется в цифровых устройствах контроля, управления и средствах вычислительной техники. Для технических систем, использующих эти средства, стоимость разработки программного обеспечения может превышать стоимость аппаратной части системы. Следует отметить, что наиболее перспективным вопросом агрегатирования цифровых систем является совмещение аппаратных и программных методов. В конечном итоге каждый функциональный модуль может иметь встроенную программу функционирования, что превращает его в аналог автономного устройства, а программирование системы сводится лишь к координации работы этих модулей, что легко выполняется на языке высокого уровня.

1.2. Стандартные интерфейсы

Унификация интерфейсов является основой развития цифровых средств управления, контроля и вычислительной техники. Под стандартным интерфейсом понимается совокупность унифицированных аппаратных, программных и конструктивных средств, необходимых для взаимодействия различных функциональных элементов в системах при условиях, предписанных стандартом и направленных на обеспечение информационной, электрической и конструктивной совместимости вплоть до нагрузочной способности на линии и нумерации контактов разъемов.

Под интерфейсной системой понимается совокупность логических устройств, объединенных унифицированным набором связей на основе

стандартного интерфейса и предназначенных для реализации алгоритмов взаимодействия функциональных модулей системы в соответствии с установленными нормами и правилами.

На рис. 1.1 приведена классификация наиболее известных интерфейсных систем. Все множество интерфейсов в зависимости от назначения можно разделить на две группы: машинно-независимые и машинные. В случае использования машинно-независимого интерфейса система проектируется независимо от применяемой ЭВМ, так как характеристики конкретной машины не влияют на функционирование системы. Замена машины ведет лишь к изменению интерфейсного модуля сопряжения (адаптера/моста). Под машинными интерфейсами подразумевают такие, которые решают задачу соединения центрального процессора конкретного типа с другими его функциональными модулями, а также подключения внешних устройств. Особенности микропроцессора полностью определяют как электрические и конструктивные характеристики интерфейса, так и принципы его функционирования.

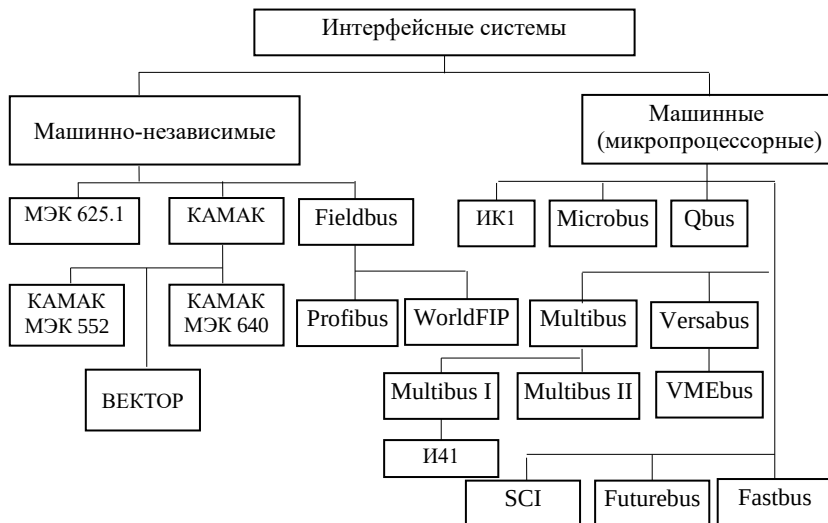


Рис. 1.1. Классификация машинно-независимых и машинных интерфейсных систем

В зависимости от структуры системы и, в особенности, от взаимодействия компонентов можно условно выделить четыре уровня интерфейсов:

- внутрисплатные и внутримодульные;
- межплатные и внутриблочные;
- межблочные;
- интерфейсы распределенных систем.

К интерфейсам первой группы относятся Qbus, Microbus, обеспечивающие связь БИС и СБИС на печатных платах или в пределах небольшой группы близко расположенных модулей. Интерфейсы второй группы ориентированы на многопроцессорный режим, к их числу можно отнести ИК-1, Multibus, VME. Интерфейсы третьего уровня в основном используются в сложных высокопроизводительных системах для объединения крупных устройств, крейтов, периферийных подсистем. К их числу относятся, например, КАМАК МЭК 552. Интерфейсы 4-го уровня предназначены для распределенных систем (КАМАК МЭК 640, SCI, Futurebus, Fastbus).

Стандартные интерфейсы можно классифицировать и в зависимости от способа соединения модулей между собой и центральным модулем (устройством обработки) системы. Различают три основные схемы соединений: каскадную, радиальную и магистральную, изображенные на рис. 1.2.

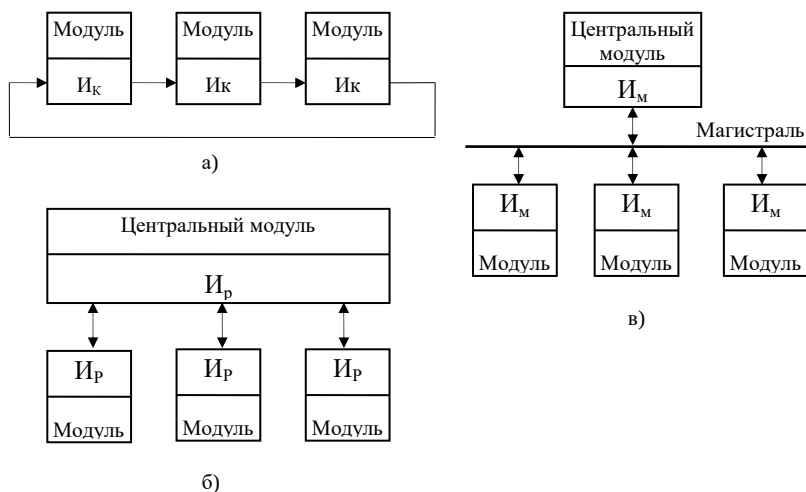


Рис. 1.2. Способы соединения модулей: а) каскадный;
б) радиальный; в) магистральный

Каскадная схема (рис.1.2,а) применима, когда общий поток информационных сигналов таков, что в каждый момент времени связываются между собой только два модуля (I_k - интерфейс каскадный).

Радиальная схема (рис. 1.2.,б) характерна для случая, когда к центральному модулю подключены несколько модулей. Причем, количество каналов связи у центрального модуля достаточно, чтобы не использовать коммутатор.

Магистральная схема (рис.1.2,в) предполагает, что все модули в системе, включая центральный, связаны друг с другом через общую магистраль с последовательным во времени адресным обращением к любому из них.

Возможны и комбинированные схемы: каскадно-радиальная, каскадно-магистральная и др.

Для соединения функциональных модулей и блоков между собой служат электрические цепи, называемые линиями интерфейса, которые могут объединяться по функциональному признаку в шины. Назначение отдельных шин и линий, их перечень и топология играют основополагающую роль при рассмотрении интерфейса. В качестве среды распространения сигналов используются проводные линии связи (витые пары), коаксиальные кабели или волоконно-оптические линии связи.

1.3. Развитие магистрально-модульных систем

Первая магистрально-модульная система была стандартизована в 1969 г, когда был опубликован европейский стандарт EUR-6100 на систему САМАС, разработанную электронщиками ЦЕРН для оснащения сложных экспериментов на ускорителях заряженных частиц. В электронной системе модулем является печатная плата с узкой передней панелью и многоконтактным разъемом на противоположной стороне платы. Модули вставляются в каркас по направляющим, в которых скользит плата. Задняя стенка каркаса выполнена также в виде платы (объединительная панель) с ответными частями разъемов, которые соединены печатными или навесными проводниками, образуя таким образом линии магистрали для передачи кодированной информации. По специально назначенным проводникам магистрали в модули подается электрическое питание.

Все присоединительные размеры модулей и каркасов строго стандартизованы: габариты и толщина печатных плат модулей, ширина каналов и направляющих и расстояние между ними, расположение контактов и другие размеры. Определены длительности и амплитуды электрических

сигналов, а также напряжения питания модулей. Впервые были стандартизованы не только размеры, но и логический протокол - правила передачи информации по линиям магистралей.

Система САМАС существует уже много лет и еще продолжает успешно использоваться с персональными ЭВМ и микропроцессорами в электрофизических установках. Дело в том, что если модули достаточно широко распространились, и их количество превзошло некоторый критический уровень, то эксплуатировать даже морально устаревшую аппаратуру оказывается выгодным.

Однако система создавалась во времена, когда никто не подозревал о появлении микропроцессоров, и поэтому магистраль не была рассчитана на их применение в модулях. Теперь же введение микропроцессоров в модули превращал их в микрокомпьютеры, а крейт в многопроцессорную систему. Поэтому к середине 70-х годов ведущие фирмы: Motorola, Intel и Digital Equipment создали магистрально-модульные системы следующего поколения Versabus, Multibus и Qbus, соответственно, с магистралями, обеспечивающими емкость системы до 1М адресов и многопроцессорный режим работы.

Добавление новых интерфейсных функций потребовало изменение логического протокола. Некоторые из микропроцессорных модулей выполняли важные задачи в системе, другие включались в работу реже, поэтому пришлось устанавливать приоритеты модулей на право владения магистралью, а также разрешать конфликтные ситуации, когда два или больше модулей одновременно пытаются захватить магистраль. Для этого потребовались дополнительные линии и, чтобы ограничить их количество, стали использовать для передачи адресов и данных одни и те же линии в мультиплексном режиме работы.

Эти новшества заложили основы магистрально-модульных многопроцессорных информационно-измерительно-управляющих систем.

В дальнейшем, Motorola перевела систему на европейские конструктивы, назвав ее Versa Module for Europe bus (VMEbus), а Intel также выпустил в европейских конструктивах новую систему Multibus-II, используя новые идеи, реализованные в системе Fastbus, созданной электронщиками США в 1982 году.

Все системы имеют 32-разрядные шины данных и адреса, что обеспечило емкость системы до 4Г адресов. Система VME первая вышла на международный рынок, а более совершенная Multibus-II опоздала и не получила должного распространения, хотя эту аппаратуру начали выпус-

кать более 100 фирм. Аппаратуру VME и ее улучшенные модификации выпускают более 300 фирм в разных странах, включая и Россию, которая собирает модули VME в небольших количествах, используя иностранные комплектующие.

В стандарте Fastbus была определена локальная информационная сеть произвольной конфигурации, работающая с логическим протоколом, впервые едином для модулей, крейтов и сети. Скорость передачи информации через магистраль была доведена до рекордной, по тем временам, значении 80 Мбайт/с, для этого длительность фронтов пришлось уменьшить до 10 нс, что на порядок меньше, чем в системе САМАС. Однако в системе не была разработана более дешевая подсистема - более медленная и с меньшим числом разрядов, поэтому Fastbus применяется главным образом в самых сложных физических экспериментах.

Развитие магистрально-модульных систем завершает разработка в 1991 году в США стандарта Futurebus. В этой системе сделано принципиальное добавление, учитывающую особенность многопроцессорных компьютеров. В таких системах каждый процессор имеет в своем распоряжении вспомогательную кэш-память. Во время выполнения параллельных ветвей программы у одного из процессора появляется промежуточный результат, необходимый другим процессорам для дальнейшей работы. Этот результат можно быстро передать в кэш-памяти нуждающихся процессоров. Процедура таких передач как раз и предусмотрена в стандарте Futurebus.

С появлением быстродействующих процессоров с тактовой частотой свыше 1000 МГц возникли проблемы передачи информации по магистрали, если к ней подключены несколько процессоров: процессор, быстро подготовивший данные, занимает магистраль для их передачи, а остальные простаивают в течение относительно медленной передачи. Магистраль, бывшая верхом достижений в 70-80 годы, к концу 80-х стала узким местом, и потребовались новые решения. В 1988 г. была начата разработка нового стандарта, известного сейчас как ANSI/IEEE Std 1596 - 1992 Scalable Coherent Interface (SCI), в русском обозначении - Расширяемый Связный Интерфейс (РСИ).

1.4. Интерфейсная система SCI

В новой системе процессорные узлы соединяются каскадно индивидуальными каналами связи, причем информация передается только в одном направлении. Узел, изображенный на рис. 1.4, получает информацию из входного канала в дешифратор адреса. Если сообщение адресовано

данному узлу, оно поступает на входную промежуточную память с очередью типа FIFO и далее проходит на прикладные схемы узла для обработки, например, микропроцессорами.

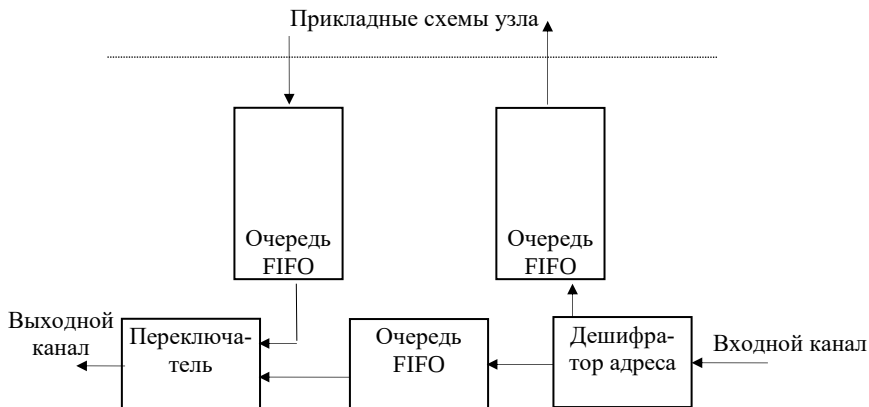


Рис. 1.4. Модель узла SCI

Если сообщение адресовано другому узлу, оно через проходную память FIFO и переключатель передается к следующему узлу. Если ранее уже началась выдача обработанной информации из выходной FIFO, передача проходящей информации задерживается до окончания выдачи. Последний из цепочки узлов SCI соединяется с первым узлом, образуя кольцо из нескольких узлов и связей. Кольцеобразная структура позволяет любому узлу получать подтверждение о приеме своего сообщения. Для этого адресованный узел сразу же после приема сообщения вырабатывает эхо-сообщение и передает его в выходной канал для передачи в узел-источник.

В системе предусмотрены узлы-агенты, имеющие выходы на боковые каналы, для соединения с другими кольцами или устройствами, выполненных и в других стандартах, как это изображено на рис. 1.5. При помощи интерфейсных агентов система может быть расширена добавлением новых колец с образованием сети произвольной конфигурации. Агент-мост на VME в настоящее время уже создан.

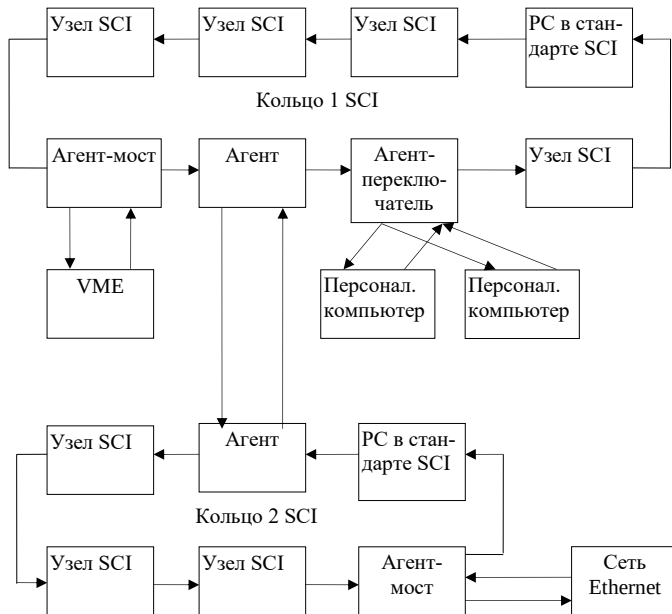


Рис. 1.5. Структура системы SCI

Слово coherent (согласованный, связный) в названии системы означает, что в стандарте предусмотрены логические средства для образования группы кэш-памятей, получающих идентичную обновленную информацию. Связность устанавливается программно при помощи указателей адресов тех узлов, кэши которых должны войти в группу. Затем процессор, источник информации, быстро выполняет ее запись в основную память и в группу кэшей.

Для компоновки аппаратных средств в стандарте определены каналы связи двух типов. Для передачи сообщений между модулями в стандартизованном каркасе используется параллельная связь. Передача между удаленными узлами осуществляется последовательно по коаксиальным или оптоволоконным линиям связи. Скорость передачи на частоте 250 МГц достигает при параллельной передаче 1 Гбайт/с, при последовательной - 1 Гбит/с. Адрес имеет 64 разряда, причем 16 - задают узел, а остальные определяют адресное пространство в каждом узле. Стандартом предусмотрены и малые дешевые системы с 32 разрядным адресом.

2. ИЗМЕРИТЕЛЬНЫЙ ИНТЕРФЕЙС МЭК 625.1

Этот интерфейс называют также GPIB (General Purpose Interface Bus), IEEE-488 (Institute of Electrical and Electronics Engineers) и HP-IB (Hewlett-Packard Interface Bus) а также КОП (Канал Общего Пользования) ГОСТ 26.003-80 в России. Международная электротехническая комиссия (МЭК) рекомендовала этот стандарт в качестве международного под номером МЭК 625.1.

Интерфейс с параллельной передачей данных предназначен для обеспечения взаимодействия программируемых и непрограммируемых средств измерений с другой аппаратурой в рамках автоматизированной измерительной системы. Он ориентирован на сопряжение не более 15 устройств, располагаемых относительно друг от друга на расстоянии, не более 4 м, при полной длине кабеля до 20 м. Скорость передачи данных по шинам не превышает 1 Мбит/с.

Интерфейс позволяет иметь в системе приборы различной сложности, допускает прямой обмен информацией между ними, дистанционное и местное управление приборами. Данный интерфейс относится к магистральным.

2.1. Структура интерфейса

На рис. 2.1 показана структурная схема интерфейса. Она представляет собой 16-линейную двунаправленную пассивную систему связи, называемую каналом общего пользования, к которой подключаются параллельно функциональные блоки. Информационные линии, образующие магистраль, сгруппированы в три шины: данных, согласования передачи и общего управления.

Конструктивно интерфейс состоит из кабеля, разъемов и печатных плат. С помощью кабеля и разъемов все устройства параллельно соединяются между собой таким образом, что данные от конкретного устройства могут передаваться либо одному, либо нескольким другим устройствам, входящим в систему. Печатные платы, с помощью которых производится обмен информацией по установленным правилам, называются интерфейсными картами.

Обмен информацией может происходить между передающими приборами, принимающими приборами или между контроллером и подчиненными приборами. По характеру взаимодействия модулей с шиной стандарт разделяет их на четыре группы: устройство-контроллер (прибор-контроллер ПК), устройство, передающее и принимающее (прибор-

источник-приемник ПИП), устройство только передающее (прибор-источник ПИ), устройство только принимающее (прибор-приемник ПП).

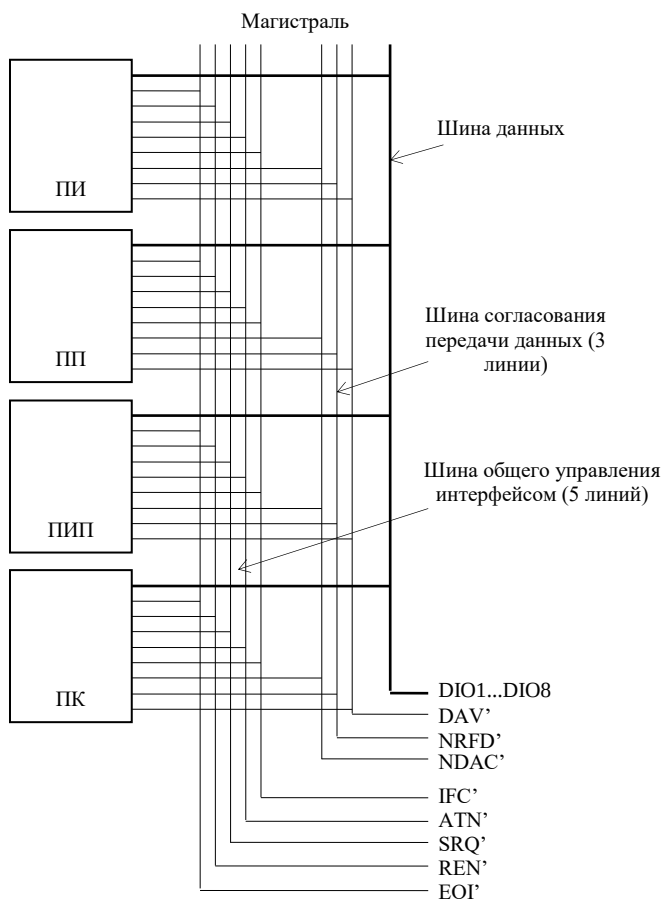


Рис. 2.1. Структурная схема измерительной системы с магистралью МЭК 625.1

Каждое устройство, имеющееся в составе системы (измерительный прибор, контроллер, вспомогательный модуль) должно выполнять, по крайней мере, одну из функций: быть передающим, принимающим или управляющим. Данные от передающего устройства передаются по шине всем устройствам, но получает их только адресованный принимающий прибор. Некоторые устройства могут выполнять обе функции передачи и приема, как например, программируемый мультиметр. Он принимает

управляющие команды как принимающий прибор и передает результаты измерений (данные) как передающий прибор. Особенность интерфейсной шины такова, что одновременно может работать несколько приемников, но только один прибор-источник.

Назначение управляющего прибора-контроллера состоит в том, чтобы организовать взаимодействие всех модулей системы. Команды контроллера указывают адрес модуля, какой модуль должен принимать данные, а какой их передавать, а также характер и последовательность выполнения других операций. Вычислительный контроллер, который строится на основе микропроцессора, выполняет более сложные функции. Он определяет программу измерений, задает согласно ей виды измерений определенным приборам, управляет процедурой согласованных измерений, обрабатывает полученные результаты. Контроллер обладает всеми интерфейсными функциями.

2.2. Состав и назначение шин интерфейса

Шина данных (Data bus) состоит из 8 линий, обозначенных DIO (Data Input/Output) или ЛД (линии данных, далее кириллицей обозначаются сигналы российского стандарта) с соответствующим номером DIO1...DIO8. По этим линиям осуществляется обмен информацией бит-параллельным, байт-последовательным способом. Шина данных служит для передачи/приема результатов измерений, адресных, программных, управляющих данных и данных состояния устройств.

Шина согласования передачи (Data byte transfer control bus), которую иначе называют шиной синхронизации объединяет три линии, обозначенные DAV' (Data valid) или СД (сопровождение данных), NRFD' (Not ready for data) с сигналом низкого уровня в активном состоянии или ГП (готов к приему) высокого уровня в активном состоянии, NDAC' (Not data accepted) или ДП (данные приняты) с теми же замечаниями относительно активных уровней напряжения передаваемого сигнала.

По этим линиям передаются сигналы согласования, подтверждающие соответствие состояний приборов, что необходимо для обмена информацией, то есть управления передачей каждого байта информации по шине данных от контроллера или прибора-источника к одному или нескольким приборам-приемникам. Эти сигналы называются также квитирующими.

Названия линий шины синхронизации определяется передаваемыми по ним сигналами. Линия DAV' служит для передачи сигнала, низкий уровень напряжения которого указывает на наличие информации на шине

данных или, иначе, для установки флага только передающего прибора. По линии NRFD' устанавливается флаг готовности только принимающим прибором. Эта линия является общей для всех принимающих приборов. Линия NDAC' предназначена для передачи квитирующего сигнала приборами-приемниками: наличие низкого уровня напряжения на ней свидетельствует, что самый медленный прибор еще не принял информации.

Шина общего управления (General interface management bus) состоит из пяти линий. По ним передаются сигналы между контроллером и другими приборами.

Линия ATN' (Attention) или УП (управление) предназначена для команды, посылаемой контроллером. Наличие сигнала низкого уровня определяет, что все остальные устройства переходят в режим ожидания и только контроллер является передающим устройством. При этом по шине данных передаются адреса или команды. После сброса сигнала в передаче информации участвуют те устройства, адреса которых были переданы контроллером при активном состоянии этой линии.

По линии IFC' (Interface clear) или ОИ (интерфейс очищен) передается сигнал контроллера, приводящий схемы интерфейса и все приборы в исходное состояние. Эта команда, используемая при запуске интерфейса и устанавливающая низкий уровень напряжения в линии, прекращает передачу информации по шине данных.

Линия SRQ' (Service request) или ЗО (запрос на обслуживание) является общей для всех приборов и устанавливается низким уровнем напряжения, когда какой либо из приборов, подключенный к интерфейсу, посылает в контроллер сигнал запроса на обслуживание, то есть требует прерывания текущего обмена в магистрали и приоритетного обслуживания данного прибора контроллером.

Линия REN' (Remount enable) или ДУ (дистанционное управление) предназначена для передачи контроллером сигналов программного управления приборами. По сигналу низкого уровня напряжения приборы переключаются с местного управления (с передней панели) на дистанционное.

Линия EOI' (End of identify) или КП (конец передачи) служит для посылки источником команды, указывающей окончание передачи сообщений по шине данных. Сигнал низкого уровня напряжения, устанавливающийся синхронно с передачей последнего байта данных, указывает на то, что данных больше нет. Если сигнал устанавливается контроллером при

параллельном опросе, то конец передачи интерпретируется как идентификация при анализе требований на обслуживание.

2.3. Интерфейсные функции

Каждый прибор, входящий в состав системы, объединенной интерфейсом, выполняет три вида функций: приборные, кодирования информации и интерфейсные. Структурная схема интерфейсного соединения прибора с магистралью изображена на рис. 2.2.

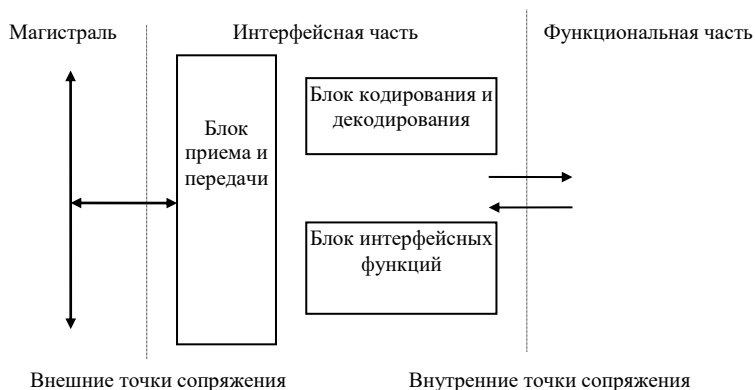


Рис. 2.2. Структурная схема интерфейсного соединения прибора к магистрали МЭК 625.1

Приборные функции задаются разработчиком прибора и могут включать в себя, например, вид измеряемой данным прибором величины, диапазон измерений, режим работы и т.п. Они выполняются в приборе его функциональной частью с учетом сигналов из интерфейсного набора. Кодирование информации выполняется в соответствии с кодами и форматами, имеющимися в стандарте.

Интерфейсные функции - это совокупность типовых операций, выполняемых при обмене данными в системе. Каждая интерфейсная функция, осуществленная в приборе, позволяет ему принимать, передавать или выполнять определенную их обработку. Взаимодействие соединенных магистралью приборов достигается в результате выполнения десяти интерфейсных функций, пять из которых относят к основным, а остальные к дополнительным.

Основные функции:

1. Прием данных L (П) или расширенный (от многих источников) прием данных LE (ПР). Эта функция позволяет прибору получить данные, переданные через интерфейс другим прибором. Такая возможность осуществляется тогда, когда контроллер или местный входной сигнал <только прием>, полученный с передней панели, задает интерфейсную функцию приема.

2. Согласование приемника АН (СП) - функция квитированного обмена с принимающим прибором, гарантирующая правильный прием информации.

3. Передача информации источником Т (И) или расширенная передача ТЕ (ИР) - функция, позволяющая передающему прибору передавать через интерфейс формируемые им данные другим приборам, подключенным к интерфейсу. Подобная возможность осуществима тогда, когда контроллер задают интерфейсную функцию передачи.

4. Согласование источника SH (СИ) - функция квитированного обмена с передающим прибором, гарантирующая правильную передачу информации.

5. Контроллер С (К) - функция, дающая возможность прибору выдавать другим приборам, подключенных к интерфейсу, адреса и универсальные команды. Если в системе несколько устройств имеют функции контроллера, то они все за исключением одного, называемого действующим контроллером, должны находиться в состоянии холостого хода.

Дополнительные функции:

1. Запрос на обслуживание SR (З) - функция, которая позволяет прибору запрашивать у контроллера операции по обслуживанию, например, последовательный или параллельный опрос.

2. Дистанционное и местное управление RL (ДМ) - функция, создающая для данного прибора возможность получать программирующую информацию либо с его передней панели (местное управление), либо от интерфейса (дистанционное управление).

3. Параллельный опрос PP (ОП) - функция, позволяющая прибору выдавать в контроллер ответную информацию при параллельном опросе без предварительной команды передачи информации. Для передачи контроллеру требований на обслуживание используется шина данных, что

обеспечивает возможность одновременной сигнализации о необходимости обслужить максимально 8 приборов или групп приборов.

4. Очистка прибора DC (СБ) - функция установки прибора в исходное состояние.

5. Запуск прибора DT (Device Trigger) (ЗП) - функция, создающая возможность контроллеру выдать прибору команду начала работы.

Приведенные интерфейсные функции гарантируют полное и гибкое управление системой. Они позволяют приборам, входящим в систему, принимать, обрабатывать и передавать информацию. С помощью этих функций протоколом работы интерфейса достигается согласование работы прибора по его прямому назначению.

2.4. Пример реализации интерфейсной функции

Алгоритм квитированного обмена данными между прибором-источником и прибором-приемником приведен на рис. 2.3.

В исходном состоянии ПИ устанавливает на линии DAV' сигнал высокого уровня, свидетельствующий, что на шине данных отсутствует информация, а ПП на линиях NRFD' и NDAC' сигналы низкого уровня, указывающие на то, что он не готов принимать данные и данные еще не приняты (состояние неготовности ANRS), тем самым подтверждая о своем участии в сеансе обмена данными.

Если до отправки данных ПИ обнаруживает, что линии NRFD' и NDAC' одновременно находятся под высоким уровнем напряжения, что означает отсутствие приборов с функцией приема, он должен прекратить обмен до следующего сеанса. В противном случае ПИ выставляет данные на шину данных и ожидает сигнала готовности к приему от ПП.

Приемник, если его функциональная часть готова к приему данных (сигнал rdy') выдает на линию сигнал высокого уровня NRFD'=0 (false), сообщая этим передающему прибору о своей готовности принять данные. Линия готовности переходит в состояние высокого уровня только тогда, когда все приемники готовы к приему. Затем приемник ожидает сигнала на линии DAV', подтверждающего о достоверности данных на шине.

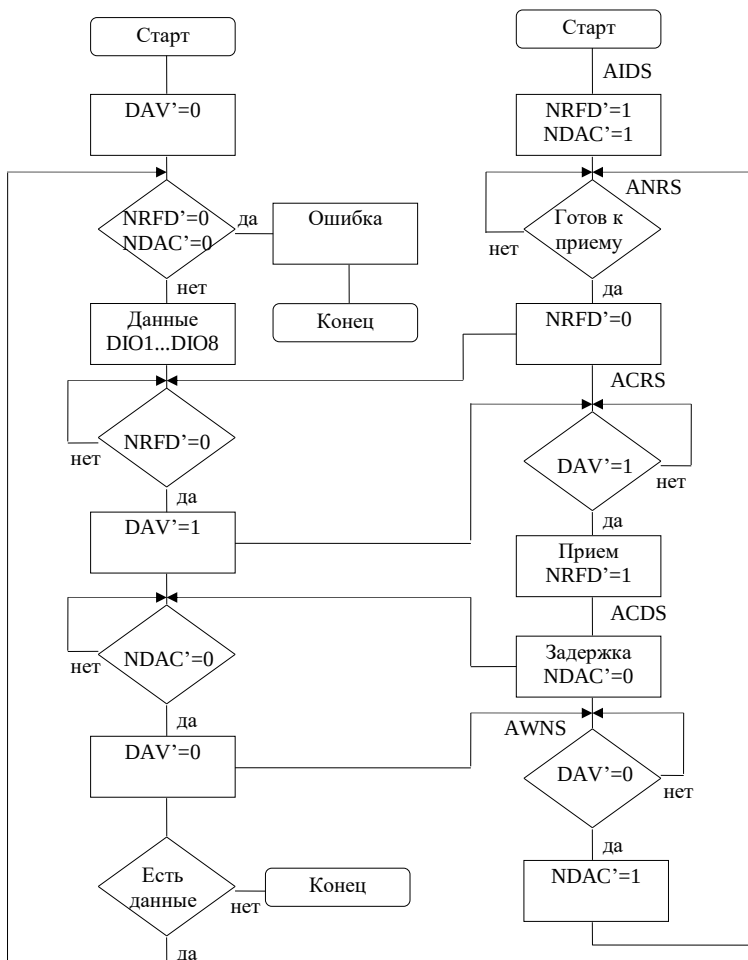


Рис. 2.3. Структурная схема алгоритма обмена данными двух приборов: источника и приемника

После того, как источник устанавливает сигнал DAV' в состояние низкого уровня (DAV'=1(true)), указывая, что данные можно принимать, приемник принимает байт данных и устанавливает сигнал неготовности NRFD'=1, сообщая источнику, что он не готов к приему и что информация принята NDAC'=0(false). Так как скорость обмена данными может не совпадать у источника и приемника, то более медленный приемник может задерживать сигнал, подтверждающий прием данных.

В ответ на последний сигнал источник устанавливает сигнал DAV' в состояние высокого уровня и с этого момента данные на шине считаются недействительными. После этого приемник устанавливает сигнал $NDAC'=1$ и переходит в исходное состояние для приема следующего сообщения. Источник, если имеются следующие данные для передачи, начинает следующий цикл обмена данными, иначе - заканчивает передачу.

В стандарте МЭК 625.1 интерфейсные функции описываются графами их состояний. Вершины графа отображают состояния, каждое из которых обозначается четырьмя заглавными латинскими буквами. Вершины графа связаны между собой дугами, отображающими допустимые переходы соответствующих состояний. Условие выполнения перехода задано в виде выражения, относящегося к данной дуге, значение которого может быть ложным или истинным. Переход интерфейсной функции в другое состояние происходит при истинном значении выражения соответствующей дуги.

Граф состояний функции АН приведен на рис. 2.4, а их соответствие передаваемым сообщениям по интерфейсным линиям управления в табл. 2.1.

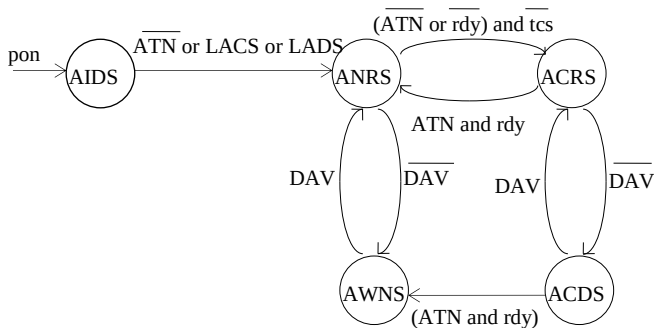


Рис. 2.4. Граф состояний функции АН

Функция обеспечивает организацию приема по шинам данных, как интерфейсных сообщений (адресов, команд), так и приборных сообщений (данных измерений и управления).

После включения питающего напряжения (pon) функция принимает исходное состояние AIDS, это же состояние она принимает при ложном значении сообщения ATN' , а также если реализуемая совместно с функцией АН функция приема L не находится ни в адресуемом состоянии LADS, ни в активном состоянии LACS.

Состояния LADS или LACS функции L или истинное значение сигнала ATN' переводит функцию АН в состояние ANRS, в котором прибор истинными сигналами NRFD' и NDAC' сигнализирует о неподготовленности к приему байта информации.

При приеме по шине данных интерфейсного сообщения с истинным значением ATN' выполняется безусловный переход функции в состояние готовности ACRS, который не зависит от внутренней готовности прибора к приему данных (rdy'). При приеме приборного сообщения сохраняется состояние неготовности неизменным до прихода сигнала готовности прибора rdy'. Причем оба указанных перехода зависимы от сигнала tcs' (период синхронизации), которое является ложным лишь в случае реализации функции АН контроллером, а в остальных случаях это сообщение истинно. Это позволяет в состоянии ACRS обеспечить организацию контроллером синхронной передачи приборных сообщений. В состоянии ACRS передачей ложного сообщения NRFD' прибор информирует о своей готовности к приему байта информации

Табл. 2.1. Сообщения, передаваемые при реализации функции АН

Состояние	Сообщения		Примечание
	NRFD'	NDAC'	
AIDS	F	F	Исходное состояние
ANRS	T	T	Состояние неготовности
ACRS	F	T	Состояние готовности
ACDS	T	T	Состояние приема
AWNS	T	F	Состояние ожидания

В ответ на истинное значение сообщения DAV', переданного прибором, реализующего функцию передачи SH, функция АН переходит в состояние приема ACDS, при котором могут выполняться прием по шине данных интерфейсных и приборных сообщений. Переходы в состояние ACDS целесообразно использовать для тактирования приема приборных сообщений. В состоянии ACDS функция передачей истинных значений NRFD' и NDAC' сообщает о неготовности прибора к приему нового байта и о продолжающемся приеме ранее установленного байта.

В процессе приема интерфейсного сообщения через время T_3 функция АН переходит в состояние ожидания AWNS. Время задержки T_3 задается разработчиком аппаратуры в зависимости от быстродействия схемы

сопряжения с таким расчетом, чтобы обеспечить надежный прием и обработку интерфейсного сообщения.

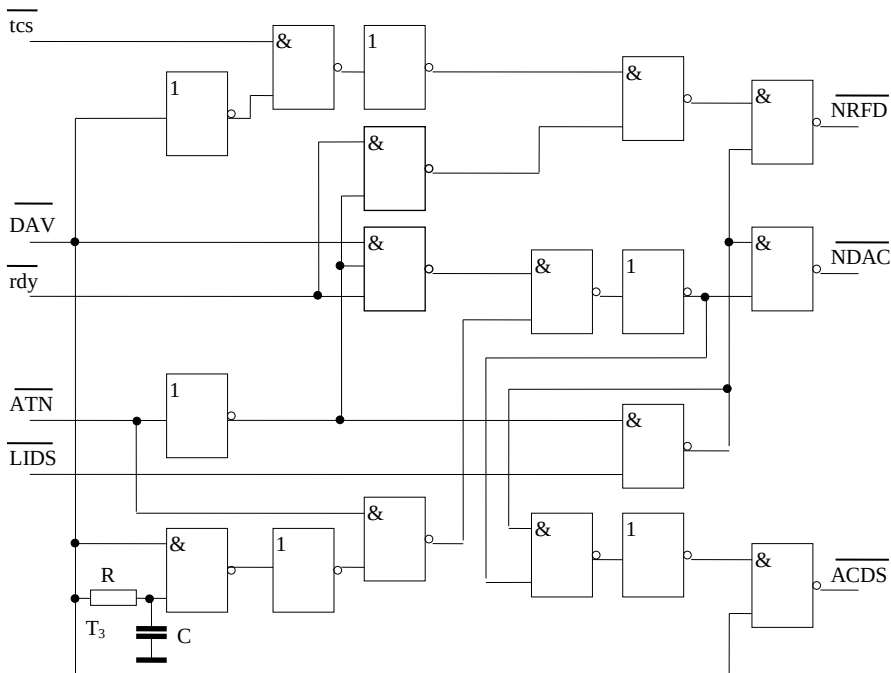


Рис.2.5. Принципиальная схема реализации функции АН

При приеме приборных сообщений следующий цикл обмена задерживается до получения истинного значения сигнала готовности $\overline{rdy}'$ либо интерфейсного сообщения $\overline{ATN}'$. В состоянии ожидания $\overline{AWNS}$ функция АН передачей ложного значения сообщения $\overline{NDAC}'$ извещается о завершении приема данных. Сообщения $\overline{NDAC}$, переданные всеми приборами в этом состоянии функции АН, будут восприняты прибором, реализующим функцию передачи SH, который заблокирует передачу сообщения $\overline{DAV}'$. После этого функция АН вновь переходит в состояние $\overline{ANRS}$ для следующего цикла согласованного приема.

Принципиальная электрическая схема реализации функции АН приведена на рис. 2.5. Схема выполнена на логических элементах и не содержит элементов памяти. Так как выходное состояние зависит только от значений входных величин, отпадает необходимость в формировании внутреннего сообщения $\overline{rop}$ (источник питания включен). С включением напряжения питания схема всегда принимает одно и то же требуемое со-

стояние. Быстродействие этой схемы обеспечивает надежный прием и обработку интерфейсных сообщений. Если реализация функции АН не связана с процессом управления приборами, то нет необходимости и во внутреннем приборном сообщении tcs, что приводит к упрощению схемы.

2.5. Сопряжение с микропроцессорной магистралью

Если при разработке микропроцессорного устройства требуется предусмотреть его работу с каналом общего пользования, то должна быть проработана схема сопряжения системной магистрали с магистралью интерфейса МЭК 625.1. Ее можно сделать, используя значительное количество микросхем малой и средней степени интеграции (40 - 50 ИС), что усложняет разработку устройства и увеличивает ее стоимость. Более эффективно и экономично использование для сопряжения специализированные БИС.

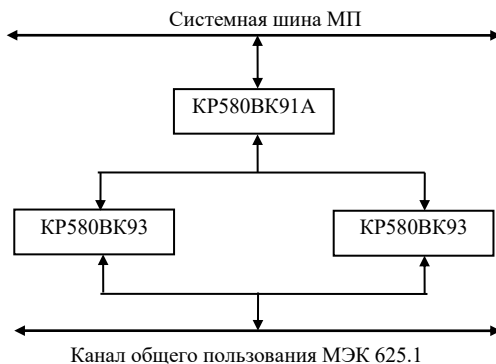


Рис. 2.6. Подключение КОП к системной шине МП КР580ВМ80А

Структурная схема, иллюстрирующая включение специализированной БИС для этих целей, приведена на рис. 2.6. В комплекте МП580 имеется БИС КР580ВК91А, которая осуществляет связь между КОП и устройствами, управляемыми микропроцессором. В ее функции входит передача данных, протокол синхронизации обмена, процедуры адресации приемников и передатчиков, очистка и запуск устройств, запрос обслуживания, последовательный и параллельный опрос, а также все остальные функции интерфейса.

Непосредственное сопряжение БИС с КОП осуществляется двумя ИС КР580ВА93, являющимися магистральными усилителями и предназначенные для буферизации шин интерфейса. Дополнительные активные и пассивные компоненты не требуются.

3. СИСТЕМА VME

Магистрально-модульная интерфейсная система VME (Versa Module for Europe или Versa Module Eurocard bus) - МЭК 821 предназначена для построения микропроцессорных систем на основе 32-разрядных микропроцессоров. Архитектура системы VME включает в себя три магистрали: системную параллельную VME, последовательную, ориентированную на связь модулей внутри одного крейта, VMS и дополнительную локальную параллельную VMX.

Необходимость дополнительной магистрали возникает при организации многопроцессорных систем, процессоры которых имеют локальные ресурсы, например, дополнительную память, устройства ввода-вывода и др. Локальная магистраль имеет несколько модификаций и для унификации модулей различных изготовителей МЭК, учтя опыт разработок, предложила новый стандарт на локальную шину - VSB (VME Subsystem bus).

Системная магистраль VME поддерживает адресные пространства, подразделяющие на области пользователя, супервизора, ввода-вывода и др., каждое из которых может достигать 4Г. Она обеспечивает передачу данных с длиной слова 8, 16, 24, 32 бит, допуская использование микропроцессоров различной разрядности. Для обеспечения эффективной работы многопроцессорных систем имеются многоуровневые системы арбитража и прерываний.

Структура магистрали построена по принципу "ведущий - ведомый модули", когда ведущее устройство в системе управляет шиной, а ведомое, после декодирования своего адреса, выполняет команду, посланную ведущим устройством с последующим подтверждением выполненной команды. Протокол обмена данными обеспечивает скорость передачи до 40 Мбайт/с.

3.1. Структура системной магистрали

Система VME, структура которой показана на рис. 3.1, состоит из интерфейсных логических схем, функциональных модулей и четырех групп линий (шин): для передачи данных, арбитража, прерываний и служебного назначения. В составе функциональных модулей наряду с ведущими и ведомыми модулями имеется системный контроллер, размещающийся в позиции первой станции крейта.

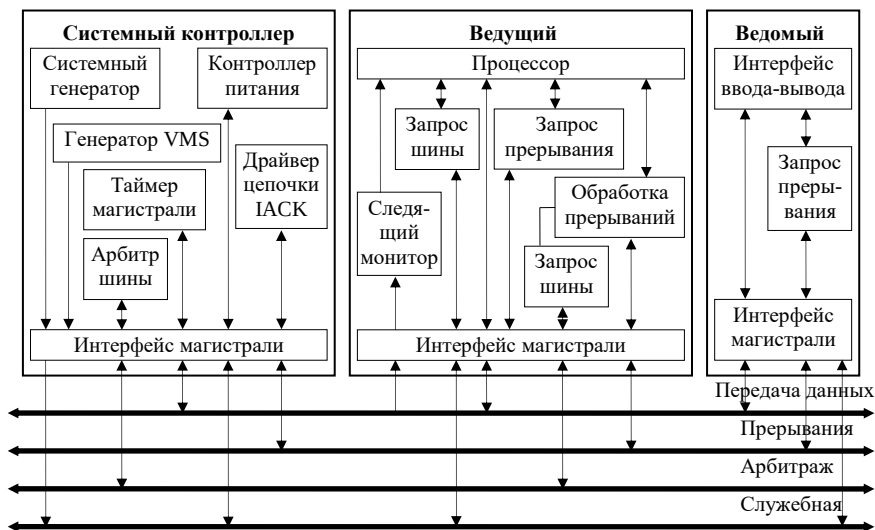


Рис. 3.1. Структурная схема организации VME магистрали

Системный контроллер включает в себя следующие функциональные устройства: системный генератор, генератор тактовых импульсов для последовательной магистрали VMS, контроллер питания, таймер магистрали, арбитр шины и драйвер последовательной цепочки подтверждения прерывания.

Системный генератор формирует импульсы с частотой 16 МГц, задает стандартную частоту для определения временных интервалов.

Генератор тактовых импульсов для VMS вырабатывает сигналы специальной формы, используемые интерфейсами шины VMS, расположенными на платах VME.

Контроллер питания обеспечивает необходимую информацию о состоянии источников питания и в случае сбоя первичного источника формирует сигнал ACFAIL', вызывающий аварийное завершение работы системы.

Таймер магистрали контролирует время ответа и заканчивает слишком длинные циклы обмена сигналом ошибки BERR'. Так как VME использует асинхронный протокол передачи данных, ведущий ожидает ответ ведомого для завершения цикла. Для предотвращения зависания

таймер автоматически формирует сигнал BERR' по истечении определенного времени, определяемого характеристиками конкретной системы.

Арбитр шины принимает запросы от источников запроса магистрали и предоставляет управление только одному из них при одновременном поступлении нескольких запросов.

Драйвер последовательной цепочки инициирует цепочку подтверждения прерывания, гарантирующую участие только одного источника в цикле подтверждения прерывания.

Ведущий модуль в своем составе может иметь следующие устройства: процессор для обработки данных, устройство обработки прерывания, источники запроса прерывания и магистрали, а также следящий монитор.

Источник прерываний генерирует запрос прерывания по одной из линий IRQ1'...IRQ7'. При получении подтверждения сообщает устройству обработки прерываний свой статус, позволяющий процессору запустить программу обработки.

Устройство обработки прерываний выделяет запросы прерываний, получает шину данных в цикле арбитража, считывает в цикле подтверждения прерывания статусную информацию источника прерывания, после чего процессор запускает программу обслуживания.

Источник требования магистрали размещается на одной плате с ведущим или устройством обработки прерывания. Он выставляет запрос на захват магистрали BR'_n и ожидает разрешение по цепочки BGIN_n - BGOUT_n. После получения магистрали выставляет сигнал занятой магистрали BBSY' и сохраняет его до конца обмена.

Следящий монитор следит за передачей данных и при обращении других ведущих по известным ему адресам вырабатывает сигнал, например, запрос прерывания своему процессору. Так, если процессор А записывает данные в ячейку глобальной памяти, адрес которой находится под наблюдением монитора на плате процессора В, то эта запись вызовет прерывание процессора В.

Ведомый модуль, например устройство ввода-вывода, может иметь источник прерывания.

3.2. Шина передачи данных

В шине передачи данных можно выделить три функциональные группы линий:

адресные A01...A31, AM0...AM5;

данных D00...D31;

управляющие AS', DS0', DS1', BERR', DTACK', WRITE', LWORD'.

Адресные линии.

Минимальная единица адресации - байт. Разряды A02...A31 определяют одну из четырехбайтных групп, а выбор одного или нескольких байт внутри группы осуществляется линиями A01, LWORD', DS0' DS1'.

Шесть линий модификации адреса AM0...AM5 дают дополнительную информацию о цикле передачи данных. Коды на этих линиях выделяют три группы адресов: специфицированные (блочная передача, область программ, область данных, внешние устройства), определяемые пользователем и резервные. Типы адресов, определяемые пользователем, могут применяться для разделения памяти между отдельными процессорами в многопроцессорных системах. В этом случае каждый процессор имеет собственное 32-разрядное пространство адресов, защищенное аппаратно.

Управляющие линии.

AS' - синхронизация адреса. Спад сигнала на линии сигнализирует ведомому, что информация на адресных линиях верна и может быть использована.

DS0', DS1' - стробы данных; DTACK' - подтверждение данных. В цикле записи спад сигнала на любой из линий DS0', DS1' указывает на истинность установленной ведущим устройством информации, а спадом на DTACK' ведомый сообщает о ее принятии. В цикле чтения ведомый сопровождает выдачу данных спадом на DTACK', а ведущий фронтом на DS0', DS1' сигнализирует об их приеме.

BERR' - низкий уровень на этой линии, устанавливаемый ведомым или таймером магистрали, сообщает ведущему об ошибке при передаче данных. Например, при попытке записать информацию в память, предназначенную только для чтения, ведомый установкой BERR' сигнализирует ведущему об ошибке. При использовании сигнала BERR' низкий уровень на линии подтверждения данных DTACK' не устанавливается и наоборот.

WRITE' - запись. Уровень сигнала на этой линии устанавливается до прихода первого из строба данных и определяет направление передачи данных. Низкий уровень означает передачу данных от ведущего к ведомому.

Основные циклы шины передачи данных

В VME определены пять основных циклов передачи данных: чтение-запись, чтение-модификация-запись, блочное чтение-запись, только адресация, подтверждение прерывания.

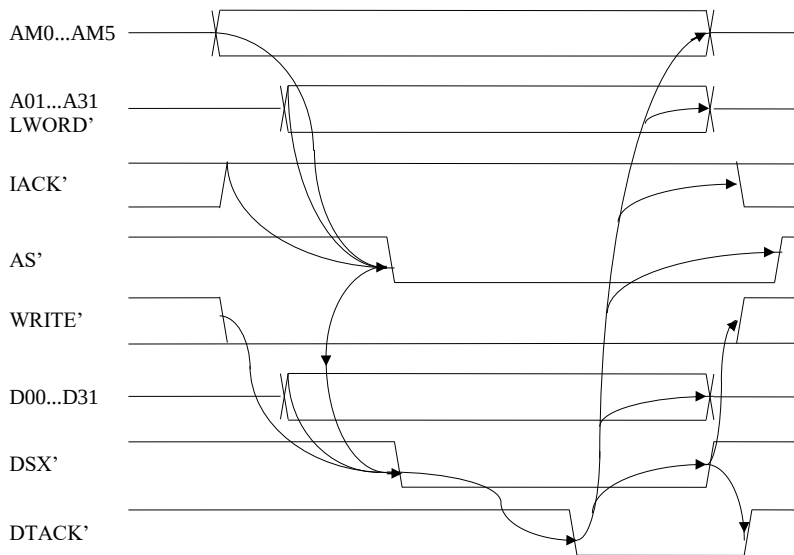


Рис. 3.2. Временные диаграммы цикла записи

Цикл чтение/запись, временные диаграммы которого приведены на рис. 3.2, использует для передачи 1, 2, 3 и 4 байт данных.

Ведомый читает информацию на линиях AM0...AM5, LWORD', IACK', A01...A31, истинную в момент прихода строба адреса AS', а на линиях WRITE', D00...D31 - при получении любого из стробов данных DS0', DS1'. Ведущий снимает информацию на всех линиях сразу после получения DTACK' и обязан выключить драйверы до снятия последнего из сигналов BBSY' и AS' только при освобождении магистрали.

В цикле **чтение-модификация-запись**, изображенном на рис. 3.3, происходит чтение информации и ее перезапись по тому же адресу. Этот цикл не может быть прерван другим ведущим, так как строб AS' не снимается.

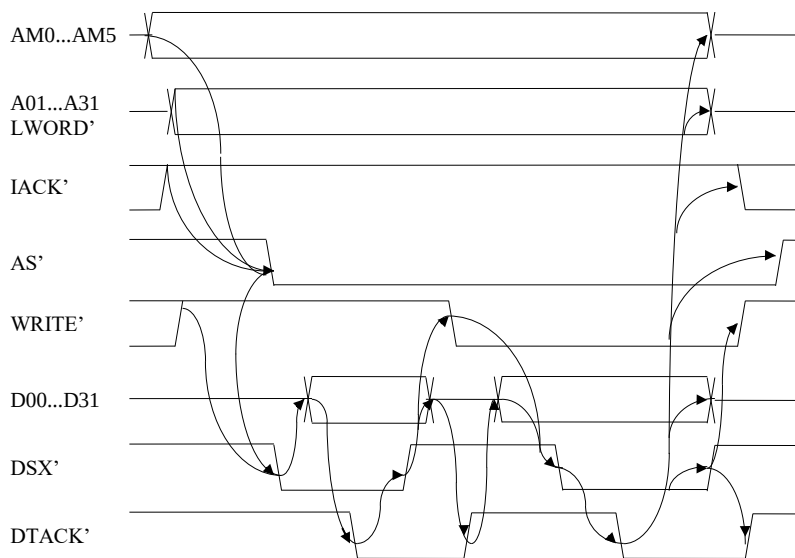


Рис. 3.3. Временные диаграммы цикла чтения-модификации-записи

Данный цикл позволяет организовать семафоры в многопроцессорных системах. Например, два ведущих совместно используют некий объект. Имеется ячейка памяти, один из разрядов которой указывает на занятость объекта. Ведущий А перед началом работы с объектом должен проверить его готовность, считывая эту ячейку, и установить соответствующий бит в цикле записи, указывая, что он занял объект. В промежутке между этими циклами ведущий В, выиграв арбитраж, может в свою очередь проверить готовность этого объекта и установить бит его занятости. В результате оба ведущих станут одновременно работать с этим объектом. Для предотвращения подобных ситуаций и введен цикл чтение-модификация-запись.

Блочные передачи, временные диаграммы которых приведены на рис. 3.4, применяются для передачи до 256 байт данных.

Этот цикл используется для обращения к последовательным ячейкам памяти. Ведущий модуль адресует первую ячейку, а затем передает данные, не заботясь об адресе, который автоматически наращивается на нужную величину ведомым в соответствии с типом передаваемых данных. В начале передачи ведомый запоминает адрес в своем адресном счетчике, а ведущий последовательно передает данные, сопровождая их стробами

данных. Принимая данные, ведомый использует для внутрислотовой адресации свой счетчик, наращивая его после каждой передачи и информируя ведущего сигналом на линии DTACK'.

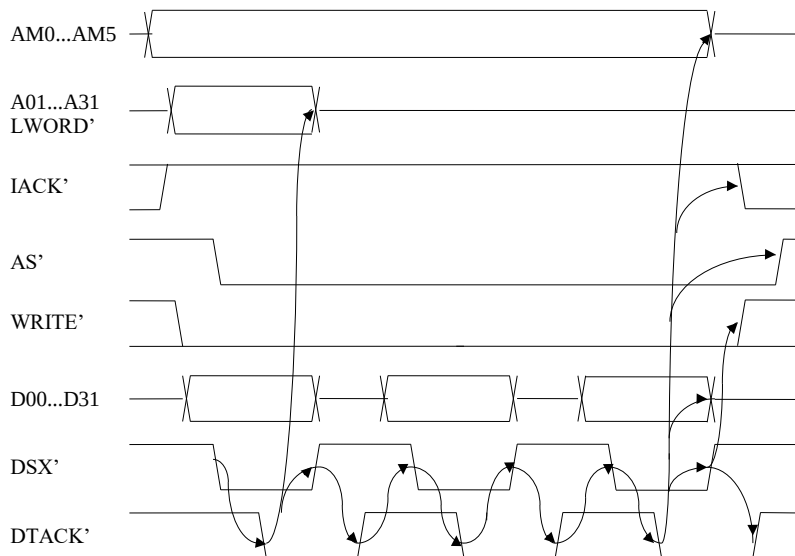


Рис. 3.4. Временные диаграммы цикла блочной передачи данных

Адрес в цикле блочной передачи не должен пересекать 256 байтных границ, изменяться могут только 7 младших разрядов адреса A01...A07. Поэтому старший адрес проверяется лишь однажды в начале передачи блока, что существенно упрощает схему ведомого. Ведущий сохраняет непрерывный контроль над магистралью данных в течение всего времени передачи блока, удерживая низкий уровень на линии AS'. Если ведущему требуется передать больше 256 байт, он должен инициировать несколько циклов блочной передачи.

В **цикле только адресации**, изображенном на рис. 3.5, передачи данных не происходит, ведущий выставляет адрес, определяет его как свой, и спустя некоторое время сам завершает обмен без сигнала подтверждения.

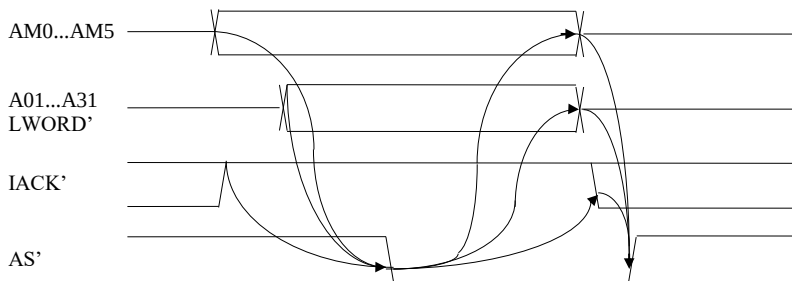


Рис. 3.5. Временные диаграммы цикла только адресации

Этот цикл используется для повышения производительности процессорных плат, позволяя процессору начать цикл магистрали прежде, чем выяснится, где находится адресат: на самой процессорной плате (локальный адрес) или вне ее. Ведущий завершает цикл магистрали без передачи по ней данных, если расположенная на плате логика после выставления адреса обнаруживает его локальность.

В **цикле подтверждения прерывания**, временные диаграммы которого приведены на рис. 3.6, происходит считывание статусной информации источника прерывания (см. раздел Шина прерываний). Слово статуса может быть 8-, 16- и 32-разрядным. Устройство обработки прерывания выполняет этот цикл при поступлении запроса от источника. Признаком цикла подтверждения прерывания служит низкий уровень на линии **IACK'**.

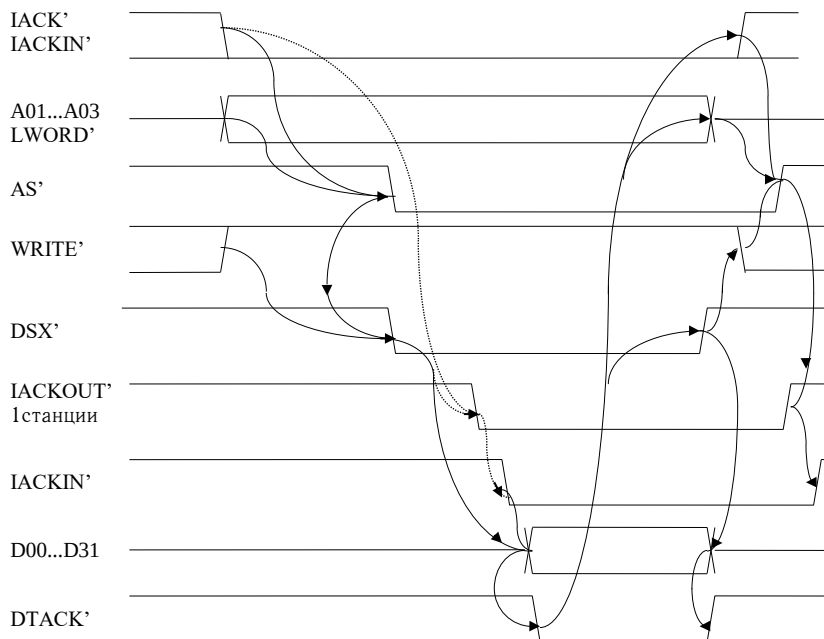


Рис. 3.6. Временные диаграммы цикла подтверждения прерывания

3.3. Производительность магистрали

Магистраль имеет отдельные стробы адреса и данных, что позволяет ведущему выставить адрес для следующего цикла в момент передачи данных предыдущего. Этот способ выставления адреса называется адресным конвейером. В этом случае из памяти можно начинать выборку следующих данных, не ожидая окончания передачи предыдущих, что обеспечивает сокращение среднего времени выборки данных.

Пропускная способность магистрали определяется электрическими характеристиками объединительной панели, быстродействием логики используемых плат и обязательными задержками, вносимыми протоколом.

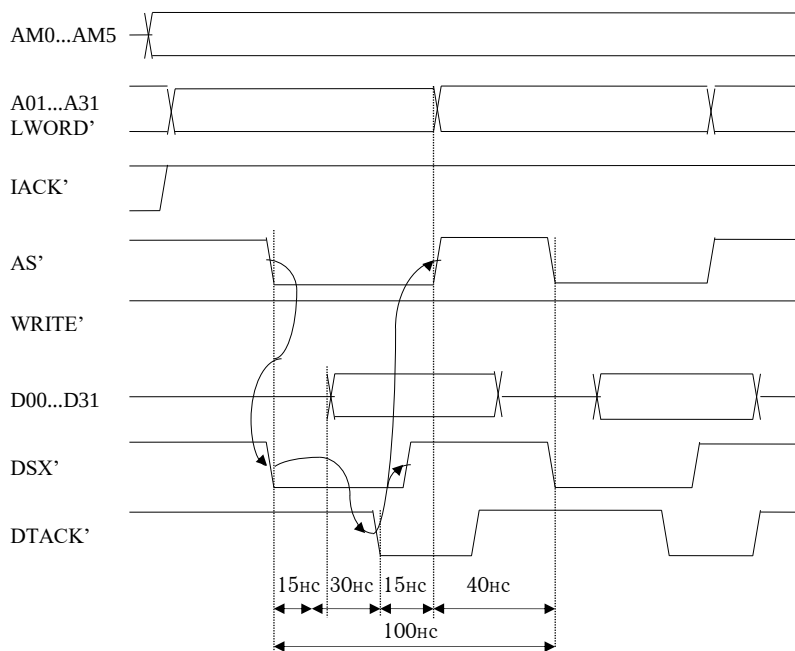


Рис. 3.7. Максимальная пропускная способность магистрали

При определении минимального времени цикла VME (рис. 3.7) задержки логических схем не учитываются. Характеристики нагруженной шины обеспечивают время распространения сигнала от первой до последней станции 15 нс. Стробы AS' , $DS0'$, $DS1'$ выставляются ведущим через 35 нс после одновременной выдачи адреса и данных. Протокол разрешает ведомому выдачу сигналов $DTACK'$, $BERR'$ не ранее чем через 30 нс после прихода первого из стробов данных $DS0'$, $DS1'$. Получая ответ ведомого, ведущий освобождает магистраль, а новый цикл может быть начат только через 40 нс. Таким образом, минимальное время цикла магистрали равно 100 нс, что соответствует скорости передачи 40 Мбайт/с.

Рис. 3.8 иллюстрирует практически достижимую скорость передачи данных по VME магистрали при современном уровне технологии. Процессор выполняет операцию чтения данных из памяти не в блочном режиме. На каждое слово данных передается свой адрес. Получаемая скорость 28 Мбайт/с достаточна для большинства применений магистрали.

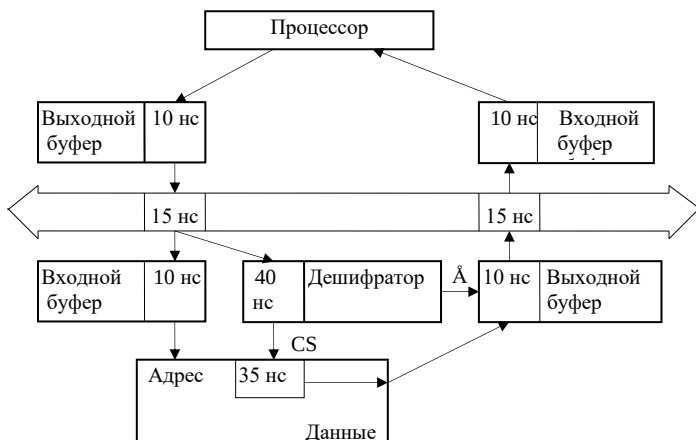


Рис. 3.8. Реальная скорость передачи данных по магистрали

3.4. Арбитраж магистрали

В многопроцессорных системах часто возникает ситуация, когда магистраль требуется одновременно нескольким ведущим или устройствам обработки прерывания. Перед каждым циклом передачи данных по магистрали ведущий или устройство обработки прерываний обязаны получить разрешение на ее использование. Процесс, определяющий, какой из них сможет использовать магистраль при одновременном поступлении требований, называется арбитражем.

Шина, обеспечивающая арбитраж, предназначена для решения двух основных задач: предотвращения одновременного использования магистрали ведущими и оптимальное распределение магистрали по требованиям от различных ведущих.

Она использует: 4 линии требования BR0'...BR3', 4 последовательные цепочки предоставления магистрали BG0'...BG3' (Bus Grant), линию занятости магистрали BBSY' и линию освобождения магистрали BCLR'.

Шина арбитража использует два типа функциональных модулей: арбитр и источник требований магистрали (см. рис. 3.1).

Существуют три основных типа арбитров магистрали: приоритетный, циклический и одноуровневый.

Приоритетный арбитр присваивает приоритеты линиям требований от низшего BR0' до высшего BR3'. При поступлении требований от ведущего с более высоким приоритетом, чем текущий, он вырабатывает сигнал BCLR'.

Циклический арбитр осуществляет циклическую смену приоритетов после каждого акта арбитража. Если в предыдущем цикле магистраль была предоставлена по требованию линии BR'_n, наивысшим приоритетом будет обладать линия BR'_{n-1}. Арбитр с циклической сменой приоритетов может использовать линию BCLR' для указания ведущему, что поступил запрос более высокого уровня на использование магистрали.

Одноуровневый арбитр воспринимает только запросы по линии BR3' и выдает разрешение по последовательной цепочке BG3IN/BG3OUT.

Кроме перечисленных типов арбитров допускаются и любые другие. Например, арбитр может рассматривать линию BR3' как обладающую наивысшим приоритетом, а на линиях требований BR0'...BR2' менять приоритет циклически.

Источник требования магистрали сигналом BBSY' удерживает магистраль до окончания обмена. Существует два способа освобождения магистрали: по выполнению и по требованию.

По выполнению - магистраль освобождается после завершения передачи всех данных.

По требованию - при появлении сигнала на одной из линий требования BR0'...BR3' или BCLR'. В этом случае сокращается среднее время доступа по магистрали, если имеется ведущий, использующий магистраль значительно чаще остальных.

Следует отметить, что определены только способы освобождения магистрали, но никаких временных ограничений на работу ведущего не накладывается. Например, контроллеру диска, получившему сигнал BCLR' в самом начале передачи блока из 256 слов в режиме прямого доступа к памяти, может потребоваться значительное время для нормального завершения операции на магистрали и предоставления ее другому ведущему.

Особое место занимают операции при появлении сигнала аварии сетевого питания ACFAIL', когда необходимо быстрое освобождение магистрали. Рекомендуется, чтобы ведущий, получив этот сигнал, освободил магистраль не позже, чем через 200 мкс.

Процессы арбитража на шине при одновременном поступлении двух запросов различного уровня отображены на рис. 7.9.

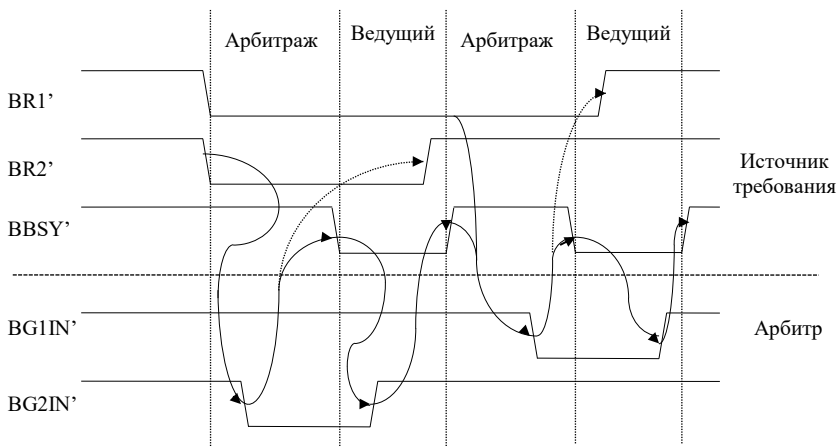


Рис. 3.9. Последовательность сигналов при одновременном поступлении двух запросов разных уровней

Два источника требований магистрали А и В одновременно выставляют сигналы на линиях BR1' и BR2' соответственно. Приоритетный арбитр отвечает сигналом BG2IN' в первой станции крейта. Получая сигнал BG2IN' по последовательной цепочки BG2IN/BG2OUT', источник В запрещает его дальнейшее прохождение, выставляет сигнал занятости магистрали BBSY' и снимает требование с линии BR2', сообщая одновременно своему ведущему о доступности магистрали данных. Следующий цикл арбитража начнется с момента снятия сигнала BBSY' ведущим В. Сигнал может быть снят по окончании работы с магистралью или во время последнего цикла передачи данных. В этом случае арбитраж будет происходить одновременно с передачей данных, а новый ведущий, выигравший его, сможет начать свои операции сразу же по окончании цикла передачи данных (по снятию сигнала AS'). При блочных передачах сигнал BBSY' рекомендуется снимать в конце блока, так как арбитраж проводится после снятия сигнала BBSY', а за время передачи блока возможен приход запросов более высокого уровня, которые смогут принять участие только в следующем цикле арбитража.

3.5. Шина прерываний

Протокол VME описывает шину прерываний, включающую в себя линии, необходимые для выработки, приема и обслуживания прерываний. Любая система, использующая эту шину, содержит некоторый набор программ обслуживания прерываний. Каждая из этих программ может рассматриваться как отдельная задача, запускаемая соответствующим прерыванием. Спецификация VME не накладывает никаких ограничений на эти подпрограммы.

Процесс прерывания можно разделить на три фазы:

- выработка запроса прерывания - с момента установления низкого уровня источником прерывания на линии запроса IRQ_n до получения шины данных устройством обработки прерываний в цикле арбитража;
- подтверждение прерывания - устройство обработки прерываний считывает по шине данных статус источника прерывания и передает управление соответствующей подпрограмме;
- обслуживание прерывания - работает подпрограмма обслуживания прерывания (не определяется протоколом VME).

Магистраль прерываний использует следующие линии:

$IRQ0' \dots IRQ7'$ - семь линий запроса прерывания. В приоритетных системах $IRQ7'$ обладает наивысшим приоритетом.

$IACK'$ - подтверждение прерывания соединяет все станции крейта с контактом $IACKIN'$ первой станции. Низкий уровень на этой линии инициирует последовательную цепочку подтверждения прерывания через ее драйвер.

$IACKIN'/IACKOUT'$ - последовательная цепочка подтверждения прерывания.

Работу системы прерываний обеспечивают три типа функциональных модулей (см. рис. 3.1): источник прерываний, устройство обработки прерывания и драйвер последовательной цепочки.

Процесс прерывания начинается в момент отправки запроса по одной из линий IRQ_n источником прерываний. Устройство обработки прерываний, обнаруживая требование от источника, запрашивает магистраль в стандартном цикле арбитража и ожидает разрешение на ее использование. Получив шину, устройство обработки начинает цикл подтверждения прерывания (рис. 3.6), выставляя на адресных линиях $A01 \dots A03$ уровень об-

рабатываемого прерывания, и сопровождая его сигналом AS' и $IACK'$. Низкий уровень на линии $IACK'$ указывает ведомому устройству, что выполняется цикл подтверждения прерывания.

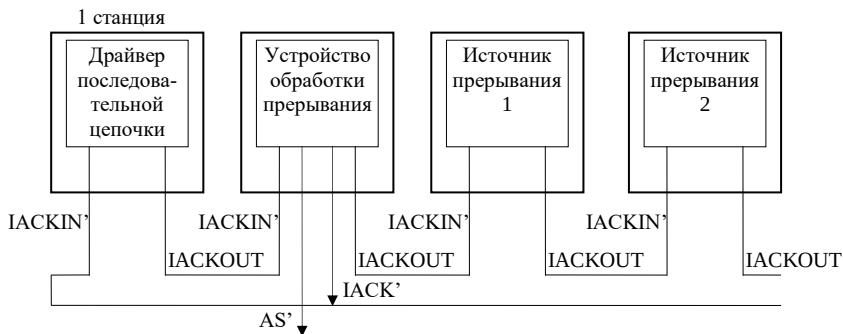


Рис. 3.10. Процесс подтверждения прерывания

Драйвер последовательной цепочки передает сигнал $IACK'$, поступивший на его вход $IACKIN'$ 1-ой станции, всем источникам прерываний, как изображено на рис. 3.10.

Получив подтверждение, источник проверяет выполнение следующих условий: запрашивал ли он прерывание и соответствует ли его приоритет уровням на линиях $A01...A03$. При невыполнении какого-либо условия источник прерывания пропускает сигнал дальше по последовательной цепочке $IACKIN'/IACKOUT'$. Если же все условия выполнены, источник выставляет на шину данных свой статус, подтверждая его сигналом $DTACK'$. Принимая статусную информацию источника, устройство обработки прерываний освобождает магистраль, заканчивая цикл подтверждения прерывания.

Системы, использующие прерывания, могут быть разделены на две группы: с централизованной обработкой запросов, имеющие только одно устройство обработки прерываний, и с распределенной обработкой запросов, включающие несколько устройств обработки, каждое из которых обрабатывает свою группу запросов прерываний.

В системах с централизованной обработкой все прерывания принимаются одним устройством, все программы обслуживания выполняются одним процессором. Такая структура применяется в системах управления, где главный процессор координирует работу подчиненных устройств, которые, как правило, представляют собой специализированные процес-

сы. В этих системах каждая из семи линий запроса имеет собственный приоритет, причем наивысший - линия IRQ_7 '.

Системы с распределенной обработкой запросов включают от 2 до 7 устройств обработки прерываний, каждый из которых обрабатывает свою группу запросов от 1 до 6. Внутри этих групп запросы могут иметь свои приоритеты. Такая структура удобна для многопроцессорных систем с распределенной обработкой данных, имеющих несколько равноправных процессоров. Процессору А, чтобы связаться с процессором В, достаточно запросить прерывание по одной из соответствующих процессору В линий IRQ_n . При одновременном возникновении запросов прерываний в разных группах очередность обслуживания будет определяться приоритетом соответствующего устройства обработки прерываний.

3.6. Служебная шина

Служебная шина определяет сигналы, предназначенные для организации работы модулей VME, содержит функциональные модули системного генератора, генератора тактовых импульсов для последовательной магистрали, контроллера питания (см. рис. 3.1).

В состав служебной магистрали входят следующие линии:

$SYSCLC'$ - системный генератор;

$SYSRESET'$ - сброс системы; может иметь ручное управление;

$SYSFAIL'$ - неработоспособность системы, возникновение аварийных ситуаций в модулях системы;

$ACFAIL'$ - авария сетевого питания, управляется контроллером питания;

+5 В - основной источник питания;

+12 В - питание RS-232;

-12 В - питание RS-232 и получение -5,2 В для ЭСЛ;

+5 В STDBY - батарейное питание;

RESERVED - резервные, не могут быть использованы разработчиком.

После подачи питающего напряжения контроллер питания устанавливает высокий уровень на линии $ACFAIL'$ и в течение 200 мс поддерживает низкий на линии $SYSRESET'$, как показано на рис. 3.11. Для управления линией $SYSRESET'$ используются драйверы с открытым коллектором, что дает возможность модулям задержать снятие сигнала

SYSRESET' на время, необходимое для выполнения процедуры начального пуска.

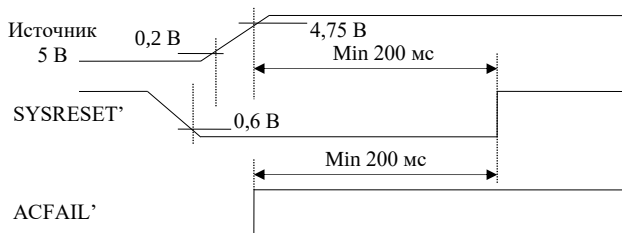


Рис. 3.11. Временные соотношения сигналов при запуске системы

Многие модули выполняют программы самотестирования при включении питания. Во время их выполнения на линии SYSFAIL', управляемой драйверами с открытым коллектором, должен поддерживаться низкий уровень. Так как сигнал SYSFAIL' в отличие от SYSRESET' не запрещает обмен по шине VME, именно он использован для обеспечения необходимого времени самотестирования всем модулям системы. Для этого вводится специальный регистр на платах, один из разрядов которого определяет уровень напряжения на линии SYSFAIL'. При включении питания этот разряд устанавливается низким уровнем напряжения и может быть снят записью в регистр после окончания программ самотестирования. Таким образом обеспечивается время, достаточное для выполнения всех программ самотестирования.

Требования к электрическим параметрам объединительной панели

К объединительной панели, представляющей собой одну или две многослойные печатные платы длиной не свыше 500 мм, может быть подключено через специальные разъемы до 21 модуля. При использовании разъема P1 магистраль имеет 24 разряда адреса и 16 разрядов данных (см. табл. 3.1), а при добавлении разъема P2 - 32 разряда адреса и 32 разряда данных.

На обоих концах линий предусмотрены согласующие резисторы, которые обеспечивают высокий уровень на линиях, управляемый передатчиками с трехстабильными выходами и схемами с открытым коллектором, уменьшают отражение от концов линий, улучшают фронты сигналов при отключении трехстабильных передатчиков.

Табл. 3.1. Назначение контактов разъемов магистрали VME

Контакт	Разъем P1			Разъем P2
	A	B	C	B
1	D00	BBSY'	D08	+ 5 B
2	D01	BCLR'	D09	GROUND
3	D02	ACFAIL'	D10	RESERVED
4	D03	BG0IN'	D11	A24
5	D04	BG0OUT'	D12	A25
6	D05	BG1IN'	D13	A26
7	D06	BG1OUT'	D14	A27
8	D07	BG2IN'	D15	A28
9	GROUND	BG2OUT'	GROUND	A29
10	SYSCLK	BG3IN'	SYSFAIL'	A30
11	GROUND	BG3OUT'	BERR'	A31
12	DS1'	BR0'	SYSRESET'	GROUND
13	DS0'	BR1'	LWORD'	+5 B
14	WRITE'	BR2'	AM5	D16
15	GROUND	BR3'	A23	D17
16	DTACK'	AM0	A22	D18
17	GROUND	AM1	A21	D19
18	AS'	AM2	A20	D20
19	GROUND	AM3	A19	D21
20	IACK'	GROUND	A18	D22
21	IACKIN'	SERCLK	A17	D23
22	IACKOUT'	SERDAT	A16	GROUND
23	AM4	GROUND	A15	D24
24	A07	IRQ7'	A14	D25
25	A06	IRQ6'	A13	D26
26	A05	IRQ5'	A12	D27
27	A04	IRQ4'	A11	D28
28	A03	IRQ3'	A10	D29
29	A02	IRQ2'	A09	D30
30	A01	IRQ1'	A08	D31
31	- 12 B	+5 B STDBY	+ 12 B	GROUND
32	+ 5 B	+ 5 B	+ 5 B	+ 5 B

Проводники объединительной панели обеспечивают характеристическое волновое сопротивление не менее 100 Ом. Однако отверстия на платах и контакты разъемов, внося дополнительную емкость, снижают его. Волновое сопротивление линий без вставленных плат 50 - 60 Ом считается достаточным для нормальной работы системы.

Платы VME могут быть одно- и двукратной высоты и имеют один или два 96-контактных разъема. На плате двойной высоты при использовании многослойного монтажа размещается до 100 микросхем в 16-выводных корпусах DIP.

4. СИСТЕМА WorldFIP

WorldFIP (FIP – Factory Interoperable Protocol) - это сетевой протокол полевой шины (fieldbus), предназначенный для обеспечения связи в системах автоматизации между нижним уровнем (датчики, исполнительные механизмы) и верхним уровнем (ПЛК, контроллеры и т. д.).

WorldFIP - это промышленная шина, требования к которой определены пользователями. С самого начала разработка протокола основывалась на:

- экономических соображениях:
 - снижение затрат на прокладку кабеля, экономия на проектировании, установке и вводе в эксплуатацию;
- технических соображениях:
 - простота обслуживания и модификации, упрощение традиционной двухточечной проводки между датчиками и процессорами;
 - гарантированное время отклика, безопасность, доступность данных.

WorldFIP может использоваться со всеми типами архитектур приложений: включая централизованную, децентрализованную, распределенную, ведущий-ведомый, которые могут быть синхронными или асинхронными.

Протокол WorldFIP полностью определен и является частью европейского стандарта полевой шины EN50170. Он относительно прост в изучении по сравнению с другими протоколами. Он максимально использует международные стандарты, такие как MMS (Manufacturing Messaging Specification). Здесь стоит добавить, что происхождение протокола WorldFIP связано с предложениями разработчиков автоматизированных систем во Франции.

В данном разделе будут рассмотрены основные положения протокола WorldFIP, описывающие вопросы управления сетью.

4.1. Общая конфигурация WorldFIP

Сеть WorldFIP основана на трехслойной модели (см. рис. 4.1.) по сравнению с известной семислойной сетевой моделью OSI (Open Systems Interconnection). Здесь используются 1, 2 и 7 слои.

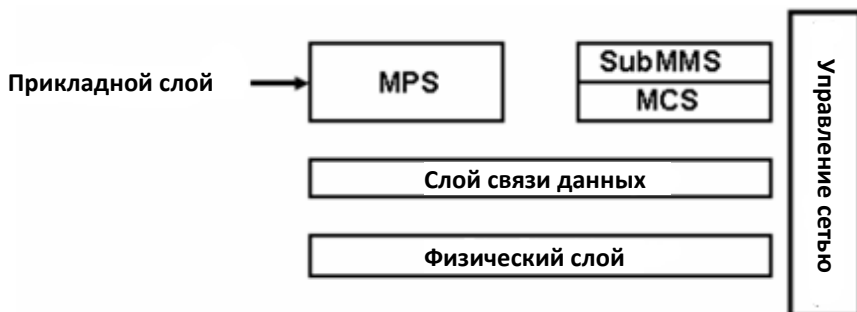


Рис.4.1. Многоуровневая архитектура WorldFIP

На рис. 4.1. MPS (Manufacturing Periodic/Aperiodic Services) означает производство периодических/апериодических служб, SubMMS означает подмножество служб обмена сообщениями, а MCS - это элемент служб приложений, обеспечивающий высокое качество обмена сообщениями. Все три уровня WorldFIP:

- Физический уровень.
- Канальный уровень передачи данных.
- Прикладной уровень.

Фактически, большинство существующих полевых шин, в том числе PROFIBUS, состоят из тех же трех уровней.

4.2. Слой физического уровня

Кабельная система WorldFIP состоит из магистрального кабеля и кабелей ответвления для подключения узлов.

Узел может подключаться к кабелю ответвления двумя способами:

- Миниатюрным 9-пиновым разъемом D-типа.
- Круглым разъемом при жестких условиях эксплуатации.

Магистральному кабелю необходим согласующий резистор на каждом его конце; он физически мал, и для удобства обычно включается в ответвитель узлов, расположенных на концах магистрального кабеля.

Общая конфигурация физических соединений одной системы WorldFIP показана на рис.4.2.

На основных кабелях расположены следующие устройства:

JB (Junction Box)	Распределительный узел - пассивный отвод, который имеет как минимум два ответвления.
TAP	Ответвитель - обеспечивает точку подключения на основном кабеле.
REP (Repeater)	Повторитель - это особый вид активной звезды. Он объединяет два основных кабеля, чтобы сформировать полевую шину.
DS (Disconnectable Subscriber)	Локально отключаемый абонент
NDS	Локально не отключаемый абонент

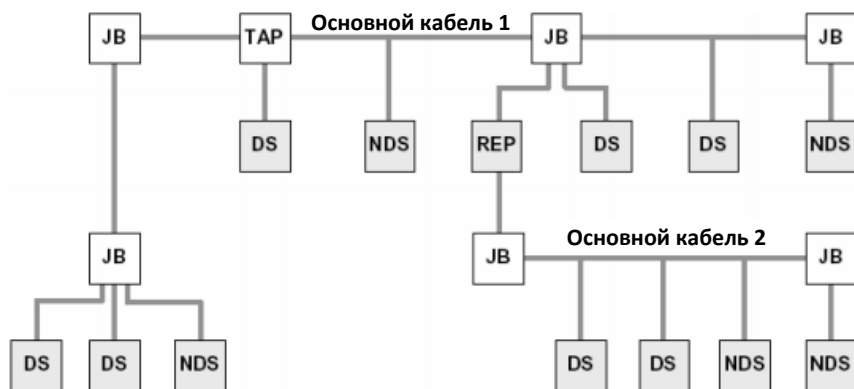


Рис.4.2. Пример физического подключения в системе WorldFIP

Физический уровень WorldFIP обеспечивает передачу битов информации от одного устройства ко всем другим устройствам, подключенным к шине. Средой передачи может быть экранированная витая пара или оптоволокно.

Для физического уровня с медным проводом определены три скорости передачи:

- S1: 31,25 кбит/с (низкая скорость)
- S2: 1 Мбит/с (высокая скорость)
- S3: 2,5 Мбит/с (высокая скорость)

S2 - стандартная скорость.

S1 и S3 используются только для специальных приложений.

Для оптоволокну определена дополнительная скорость 5 Мбит/с

Основные правила подключения

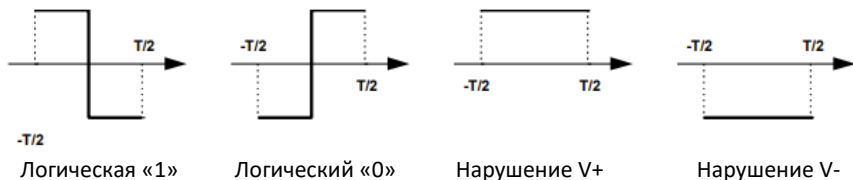
До 64 устройств могут быть связаны в одном сегменте шины WorldFIP со скоростью передачи данных 1Мбит/с, которая имеет структуру "магистраль + ответвление", с ответвлениями до 0,5 м и максимальной рекомендуемой общей длиной сегмента 750 м. Для расширения магистрали может быть использовано до 4 повторителей, делая максимальную длину для всей сети из конца в конец 3.75 км. Повторитель учитывается как одно устройство на каждом из сегментов, из которых состоит сеть.

Минимальное требование для кабеля - одна витая пара плюс экран с характеристическим импедансом 150 Ом. Кабель с 2-мя отдельно экранированными парами также может быть полезен (вторая пара может быть использована либо в качестве дубликата шины, либо для питания). Основную магистраль необходимо согласовать, чтобы предотвратить нежелательные отражения сигнала. Простейшим терминатором является 150 Ом резистор, подключенный между проводниками.

На физическом уровне кодируются биты, передаваемые уровнем канала данных, с использованием манчестерского кода.

Этот код позволяет одновременно передавать временную синхронизацию сигналов и данные. Каждый временной интервал, используемый для кодирования бита, делится на две части равной продолжительности.

Биты представлены следующим образом:



Обмен данными по шине осуществляется посредством передачи кадров, структура которых изображена на рис. 4.3.

Все кадры WorldFIP (frame_question, response, message и т. д.) состоят из трех частей:

- FSS (Frame Start Sequence) - последовательности начала кадра,
- CAD - поля управления, данных и проверки,
- FES (Frame End Sequence) - последовательности конца кадра.

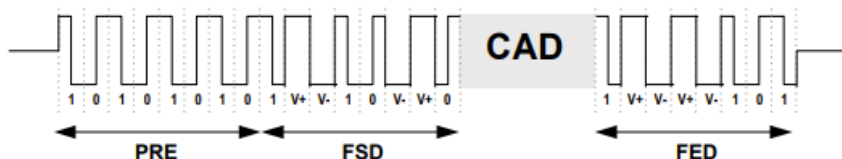


Рис.4.3. Структура кадра с обрамлением

Последовательность начала кадра (FSS) содержит следующие поля:

- Преамбула (PRE - Preamble). Эта серия из 8 битов «1» и «0» используется приемниками для синхронизации с передатчиком.
- Разделитель начала кадра (FSD - Frame Start Delimiter). Эта последовательность битов указывает на уровне канала передачи данных начало полезной информации CAD.
- Поле CAD – (Control and Data) содержит только логическую информацию («0» и «1») на канальном уровне данных.

Последовательность конца кадра содержит поле:

- Ограничитель конца кадра (FED - Frame End Delimiter). Эта серия битов используется уровнем канала данных для определения конца поля CAD.

Физический уровень добавляет 24 бита к каждому передаваемому кадру.

4.3. Слой связи данных

Уровень канала передачи данных WorldFIP использует известную модель производителя/потребителя в сети с централизованным планировщиком шины. Этот централизованный планировщик называется арбитром шины (Bus Arbitrator) или BA.

Службы уровня канала передачи данных

Протокол WorldFIP в основном поддерживает два типа служб передачи:

- Обмен идентифицированными переменными.
- Обмен сообщениями.

WorldFIP использует два типа переменных: данные в реальном времени, как критичные по времени, так и некритичные по времени.

Переменные реального времени могут быть периодическими или аperiодическими, то есть переменными на основе событий. Кроме того, протокол поддерживает асинхронную передачу сообщений между узлами.

Для обмена сообщениями имеется два типа: широковеcательная рассылка неподтвержденных аperiодических сообщений и широковеcательная рассылка подтвержденных аperiодических сообщений.

Адресация переменных и сообщений

Метод передачи в WorldFIP, в основном, основан на широковеcательной передаче по сетевой среде (то есть по шине). Поэтому необходимо знать, как WorldFIP может различать переменные. То же самое относится к сообщениям. WorldFIP использует два различных адресных пространства: адресное пространство переменных и адресное пространство сообщений.

Адресация переменных

В протоколе WorldFIP каждая переменная в системе связана с уникальным идентификатором, который используется для характеристики этой переменной. Для кодирования этого идентификатора используется 16 бит. Другими словами, существует $2^{16} = 65536$ переменных, которые могут иметь идентификаторы. Объекты, участвующие в обмене переменной, физически не обращаются друг к другу. Скорее, они ссылаются на идентификаторы, которые они распознают при производстве или потреблении.

Для данного идентификатора может быть один и только один производитель, но несколько потребителей. Обмен переменными осуществляется посредством широковеcательной передачи.

Адресация сообщений

В WorldFIP обмен сообщениями происходит двумя способами либо точка-точка (point-to-point), либо в многоточечном (широковеcательном) режиме на одном сегменте. Каждое переданное сообщение содержит адреса передающего и целевого объектов. Для кодирования адресов сообщений используются 24 бита. Это означает, что для сообщений в WorldFIP существует $2^{24} = 16777216$ адресов. Схема адресации основана

на адресации объектов. Адрес объекта содержит адрес сегмента сети, а также адрес станции (узла) в этом сегменте.

4.4. Интерфейсы прикладного - физического уровней

Уровень канала данных предоставляет службы прикладному уровню и использует службы физического уровня.

Уровень канала передачи данных состоит из набора создаваемых и используемых буферов. Эти буферы содержат последние значения, обновленные пользователем или сетью. Когда новое значение вводится в созданный или используемый буфер, предыдущее значение перезаписывается.

Ресурсы, создающие эти буферы, выделяются при первоначальной настройке станции. Доступ к буферу осуществляется через создаваемый или используемый идентификатор.

На рис. 4.4. показан интерфейс прикладного уровня - уровня канала данных.

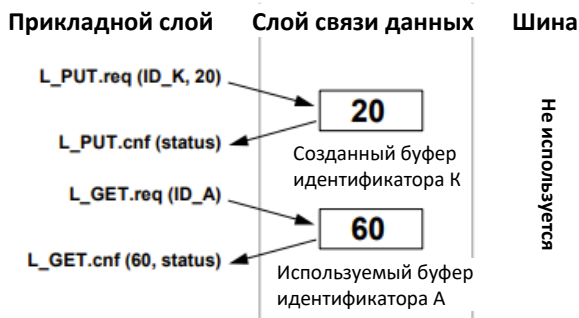


Рис. 4.4. Интерфейс прикладного уровня - уровня канала данных

В этом примере уровень канала передачи данных содержит два буфера:

- созданный буфер, соответствующий идентификатору K
- используемый буфер, соответствующий идентификатору A.

Прикладной уровень использует службу записи (L_PUT.req) для помещения нового значения (20) в созданный буфер. L_PUT.cnf - подтверждение записи.

Затем прикладной уровень использует службу чтения (L_GET.req) для получения значения (60) используемой переменной. L_GET.cnf - подтверждение чтения.

Эти службы чтения и записи являются локальными для станции и не вызывают активности на шине.

Создаваемые или используемые буферы также доступны через шину, через которую можно получать или сохранять значения переменных. Этот механизм известен как буферный перенос.

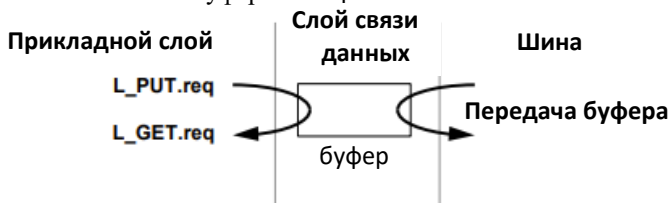


Рис.4.5. Буферный перенос

Поскольку буферы имеют два доступа, уровень канала передачи данных должен решать проблемы, вызванные конфликтом доступа. Когда переменная создается и используется в одном коммуникационном объекте, уровень канала данных содержит как созданный, так и используемый буфер.

Для передачи буфера арбитр шины передает кадр вопроса ID_DAT и указывает идентификатор.

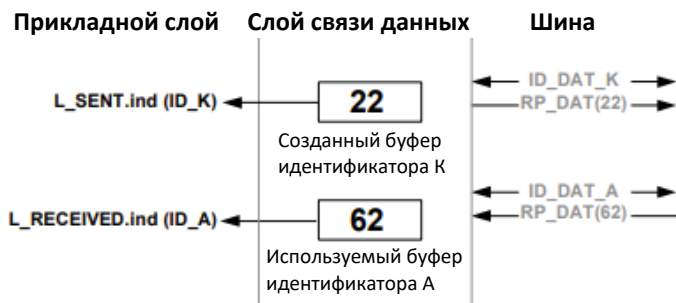


Рис.4.6. Передача буфера

Если станция объявлена производителем идентификатора, уровень канала передачи данных отвечает значением переменной, используя кадр ответа RP_DAT. Затем уровень канала передачи данных отправляет указание передачи значения на прикладной уровень (L_SENT.ind).

Если станция объявлена потребителем идентификатора, уровень канала данных принимает значение последующего кадра ответа RP_DAT.

Затем он отправляет индикацию получения идентификатора на прикладной уровень (L_RECEIVED.ind).

Максимальный размер буфера - 128 байт. Буферы содержат только значения идентифицированных переменных; они не содержат сообщений.

4.5. Передача данных

Механизм передачи данных, как уже было сказано, использует два типа кадров: ID_DAT и RP_DAT. Каждый передаваемый кадр WorldFIP обрамлен информацией физического уровня. Кадр помещается между полем FSS (Frame start sequence или начало кадра) и полем FES (Frame end sequence или конец кадра). Эти поля объясняются в разделе о физическом уровне.

Все кадры WorldFIP начинаются с управляющего байта Control, который используется сетевыми абонентами для распознавания типа кадра, который они получают. Это поле управления используется для кодирования запросов на передачу переменных, кадров подтверждения и т. д.

Все кадры WorldFIP заканчиваются двумя байтами FCS (Frame Check Sequence), используемыми получателем кадра для проверки целостности полученного кадра. FCS является результатом полиномиальной операции (сложение без переноса), выполненной на предыдущих байтах.

Все кадры запросов, будь то кадры ID_RQ или ID_MSG, относятся к типу ID_DAT. Они отличаются только несколькими битами в поле управления.

ID_DAT	FSS	Control	Identifier	FCS	FES
	2 байта	1 байт	2 байта	2 байта	1 байт

Формат кадра, переданного в ответ на кадр ID_DAT, показан ниже:

RP_DAT	FSS	Control	Data	FCS	FES
	2 байта	1 байт	n байтов (n<=128)	2 байта	1 байт

Для уровня приложения поле DATA может содержать до 128 байтов. Поле управления указывает также, есть ли какие-либо аperiodические запросы на передачу переменных или сообщений.

Передача ответа на запрос переменной (RP_RQ)

Когда станция, которая сделала запрос на передачу переменной, принимает кадр ID_RQ, она отвечает кадром RP_RQ, закодированным следующим образом:

RP_RQ	FSS	Control	List of identifier	FCS	FES
	2 байта	1 байт	n x16 bit (n<=64)	2 байта	1 байт

Передача ответа на запрос сообщения (RP_MSG_xxx)

Когда станция, сделавшая запрос на передачу сообщения, получает кадр ID_MSG, она отвечает кадром RP_MSG_NOACK или RP_MSG_ACK, закодированными следующим образом:

RP_MSG_xx	FSS	Control	dest.adr	src.adr	Message	FCS	FES
	2 байта	1 байт	3 байта	3 байта	макс.256 байт	2 байта	1 байт

Бит в поле управления указывает, подтверждается ли передача сообщения или нет. Поля назначения и источника показывают адреса взаимодействующих объектов.

Передача ответа подтверждения (RP_ACK)

Когда станция назначения получает сообщение с запросом подтверждения, она передает кадр подтверждения:

RP_ACK	FSS	Control	FCS	FES
	2 байта	1 байт	2 байта	1 байт

Этот кадр очень короткий, так как информация о подтверждении содержится в поле управления.

Передача ответа конца транзакции сообщения (RP_FIN)

Когда сообщение было передано, отправитель, дождавшись подтверждения, если необходимо, передает кадр завершения транзакции сообщения:

RP_FIN	FSS	Control	FCS	FES
	2 байта	1 байт	2 байта	1 байт

Этот кадр так же короткий, так как информация о завершении транзакции содержится в поле управления.

4.6. Механизм доступа к шине

Сеть WorldFIP состоит из станций (S) с двумя типами функций:

- арбитраж шины: управление доступом к среде передачи.
- функция производства/потребления данных.

Любая станция WorldFIP может одновременно выполнять эти две функции, но в любой момент только одна станция может выполнять функцию активного арбитража шины.

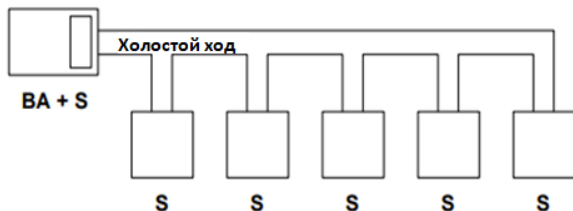


Рис.4.7. Конфигурация сети WorldFIP

Арбитр шины (BA) имеет ресурсы, необходимые для сканирования переменных в моменты времени, определенные при настройке системы. Арбитр шины имеет таблицу сканирования со списком идентификаторов для передачи по шине.

Работа шинного арбитра относительно проста. Он использует вопросительный кадр ID_DAT для широковещательной передачи по шине идентификатора переменной (рис.4.8). Этот вопрос одновременно записывается по всем каналам передачи данных всем станциям, подключенным к шине. Одна и только одна из этих станций (P) признает себя производителем переменной. Одна или несколько других станций (C) распознают, что они являются потребителями этой переменной.

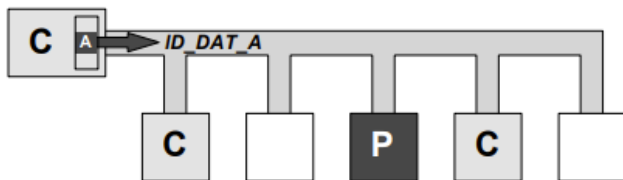


Рис.4.8. Передача идентификатора A

Затем производитель переменной передает значение переменной в кадре ответа RP_DAT (рис.4.9). Это значение одновременно фиксируется всеми потребляющими станциями. Затем арбитр шины переходит к следующему идентификатору в таблице сканирования, и тот же цикл вопрос-ответ повторяется.

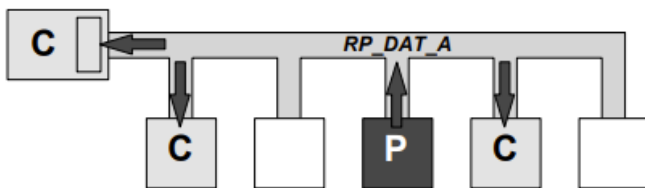


Рис.4.9. Передача значения переменной

Передача по шине произведенного значения или прием потребляемого значения включает механизм передачи буфера. Передача буфера происходит по инициативе арбитра шины, она полностью не зависит от активности пользователя.

Когда новые станции, которые потребляют одну или несколько переменных, подключаются к шине, то не требуется дополнительного времени для передачи этих переменных новым станциям. Механизм остается прежним.

Таймеры механизма состояния уровней канала данных постоянно отслеживают активность на шине. Таким образом, если кадр ответа (RP_DAT) потерян или поступает медленно, потребители возвращаются в состояние ожидания кадра вопроса и игнорируют все другие типы кадров.

4.7. Таблицы арбитража шины

Арбитр шины должен иметь таблицу сканирования или BAT (Bus Arbitrator Table), в которую входит список идентификаторов периодических переменных для их распространения или сканирования по шине через равные промежутки времени. Когда система настроена, арбитру шины предоставляется список идентификаторов периодических переменных, которые будут сканироваться, и периодичность, связанная с каждой из этих переменных. Также арбитр шины должен учитывать время передачи или длину каждого пакета данных, чтобы оценить требуемое время передачи.

BAT состоит из малых циклов сканирования, которые называются микроциклами, и больших циклов сканирования, которые называются макроциклами. Микроцикл устанавливает максимальную скорость, с которой БА выполняет набор сканирований (периодических и аperiodических).

Макроцикл включает в себя заданное количество микроциклов.

ВА проверяет конфигурацию системы, чтобы увидеть, будет ли система соответствовать ограничениям по времени или нет. После проверки конфигурации ВА будет бесконечно повторять механизм сканирования до тех пор, пока система не будет выключена или не произойдет сбой.

Алгоритм сканирования WorldFIP основан на циклическом исполнительном планировании. Циклическое исполнительное планирование основано на фиксированном графике, называемом основным циклом (макроциклом), который состоит из нескольких элементарных циклов фиксированной продолжительности (микроциклов). Для определения микроцикла и макроцикла используется метод наибольшего общего множителя/наименьшего общего кратного (HCF/LCM).

Использование метода HCF/LCM для создания BAT

Прежде необходимо определить сетевую модель, которая будет использоваться для создания таблицы арбитра шины.

Данная сетевая модель выглядит следующим образом:

n = количество станций, одна из них - ВА.

n_p = количество периодических переменных.

Sp_i = периодический поток $i = \{Cp_i, Tp_i, Dp_i\}$, $i = 1, 2, \dots, n_p$.

Cp_i = максимальное время периодической передачи буфера.

Tp_i = Периодичность Sp_i .

Dp_i = предельный срок периодического потока Sp_i .

N = количество микроциклов в макроциклах.

μ_{cy} = продолжительность микроцикла.

M_{cy} = продолжительность макроцикла.

Оценка микроцикла и макроцикла выполняется следующим образом:

Микроцикл или элементарный цикл равен наибольшему общему множителю всех периодичностей периодических переменных.

$$\mu_{cy} = HCF(Tp_i), i = 1 \dots n_p, \quad (2.1)$$

где HCF - наибольший общий множитель, а i изменяется от 1 до n_p или количества потоков периодических переменных.

Чтобы микроцикл был целым числом, кратным любой периодичности периодических переменных, уравнение (2.2,а) показывает, как этого добиться.

$$\mu_{cy} = \max \{ \Omega \}, \Omega \in \aleph, \quad (2.2,а)$$

где « $\aleph$ » - положительное целое число; а " Ω " рассчитывается как HCF с использованием уравнения (2.2,б) следующим образом:

$$Tri / \Omega = [Tri / \Omega], i = 1...np \quad (2.2,б)$$

Теперь, после расчета интервала микроцикла, найдем макроцикл БАТ. Из сетевой модели, приведенной выше, макроцикл равен N целому числу, кратному микроциклу (уравнение (2.3)).

$$Mcy = N * \mu_{cy} \quad (2.3)$$

Проблема состоит в том, чтобы найти целое число N. Уравнение (2.4,а) показывает, как можно найти это целое число.

$$N = \min \{ \Phi \}, \Phi \in \aleph, \quad (2.4,а)$$

где « $\aleph$ » - положительное целое число; а " Φ " рассчитывается как LCM с использованием уравнения (2.4,б) следующим образом:

$$\Phi / (Tri / \mu_{cy}) = [\Phi / (Tri / \mu_{cy})], i = 1...np \quad (2.4,б)$$

Таким образом, вычислили целое число "N", используя LCM или наименьшее общее кратное периодичностей периодических переменных.

Теперь, найдя два ключевых параметра БАТ, можно построить таблицу. Здесь используется алгоритм монотонного планирования, в котором ВА назначает фиксированные приоритеты периодическим переменным на основе периодичности этих переменных. Затем ВА заполняет БАТ периодическими переменными от наивысшего приоритета до самого низкого приоритета.

Числовой пример БАТ

Предположим, что имеется список из шести периодических переменных, как показано в табл. 4.1, и требуется создать БАТ для отображения этих переменных, которые будут использоваться в системе WorldFIP. Здесь S_{pi} представляет время, необходимое для завершения отправки переменной в место назначения.

Табл. 4.1. Пример ВАТ

Переменная	Периодичность (мс)	Cpi (мкс)
A	5	170
B	10	178
C	15	418
D	20	194
E	20	194
F	30	290

Применив упомянутый ранее алгоритм для вычисления длительности микроцикла и макроцикла (уравнения 2.2 и 2.3), определим, что микроцикл равен 5 мс, а макроцикл равен 12 микроциклам или 60 мс.

Затем применим алгоритм монотонного планирования скорости для создания таблицы. На рис. 4.10 показан пример, как нагрузка трафика WorldFIP будет отображаться в первых 15 элементарных циклах.

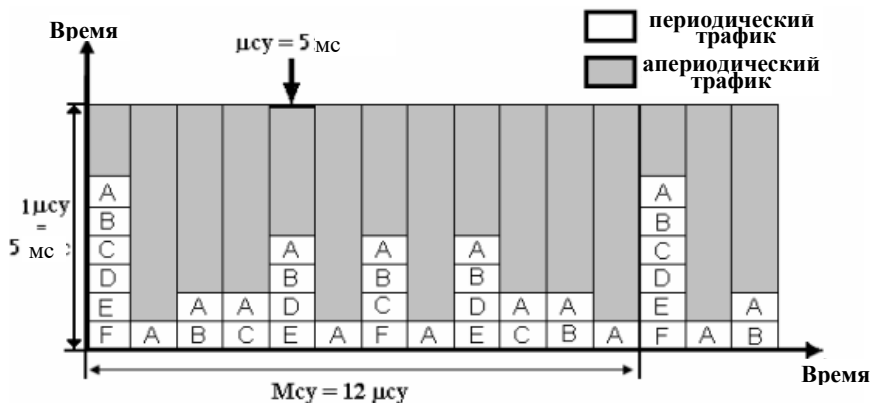


Рис.4.10. Загрузка сети WorldFIP

Горизонтальная ось соответствует течению элементарных циклов во времени. Эта ось может продолжаться бесконечно. Отметим, что нет элементарного цикла, который может выйти за пределы 5 миллисекунд. Другими словами, когда ВА завершает весь периодический трафик в одном микроцикле и в этом микроцикле еще остается некоторое время, ВА может переключиться и использовать оставшееся время для выполнения запросов на аперiodические передачи или сообщения. Если для аперiodического трафика не поступают запросы, ВА передает кадры заполнения до конца этого элементарного цикла.

Рис. 4.11 показывает, что каждый микроцикл состоит из четырех неравных окон. Первое окно предназначено для периодического трафика.

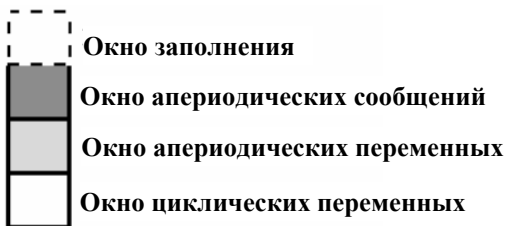


Рис.4. 11. Окна элементарного цикла WorldFIP

Второе окно предназначено для транзакции с аperiodической переменной для обоих типов: срочных и нормальных аperiodических переменных. Эти аperiodические переменные в основном используются для подачи сигналов тревоги в критических ситуациях или сбоях системы.

Третье окно предназначено для аperiodической передачи сообщений двух типов: подтвержденных и неподтвержденных.

Наконец, четвертое окно - это окно заполнения, которое используется для поддержания синхронизации микроциклов.

Таким образом, единственное ограничение на микроцикл состоит в том, что всегда должно быть время для завершения периодического трафика, а затем, если есть дополнительное время, его можно использовать для аperiodического трафика.

4.8. Режимы работы WorldFIP

Одним из наиболее распространенных режимов работы является синхронный режим. Основная особенность этого режима состоит в том, что микроциклы всегда имеют постоянную продолжительность, которая должна быть определена в начале настройки ВАТ. Другой важной особенностью этого синхронного режима является то, что следующее сканирование ВАТ начинается сразу же после завершения текущего сканирования. В этом режиме всегда можно гарантировать периодические переменные с возможным небольшим временным дрожанием (jitter).

Когда обмен одной периодической переменной будет занимать меньше времени, чем продолжительность микроцикла, тогда ВА будет передавать кадры заполнения до конца этого микроцикла (в случае отсутствия ожидающего аperiodического трафика). Когда используется этот синхронный режим, продолжительность микроцикла обычно устанавли-

вается равной наибольшему общему множителю (HCF) периодов сканирования переменных.

Есть и другие режимы, которые может использовать WorldFIP. Например, режим без заполнения, когда ВА разрешает запуск следующего микроцикла, как только транзакции текущего микроцикла завершаются (то есть, нет вставки кадров заполнения). В этом случае продолжительность микроцикла непостоянна.

Еще один пример асинхронного режима, когда следующее сканирование может быть инициировано внешним событием, независимо от завершения текущего сканирования (например, системный сбой или особое состояние системы для контроля или обслуживания).

Далее рассматривается только один режим - синхронный.

Механизм использования среды для аperiodических переменных

Не все переменные в сети WorldFIP включены в ВАТ, фактически, только периодические переменные. Механизм, который ВА использует для выполнения транзакций с аperiodическими переменными, разделен на три этапа:

Первый этап

Предположим, что ВА передает вопросительный кадр, касающийся периодической переменной А, как показано на рис. 4.12,а. Это делается обычным способом в циклическом окне любого микроцикла. Производитель переменной А отвечает соответствующим значением переменной А и устанавливает бит аperiodического запроса в поле управления кадра ответа RQ, как показано на рис. 4.12,б.

Когда производитель переменной А устанавливает аperiodический бит, это означает, что он запрашивает аperiodическую передачу переменной. У такого аperiodического запроса на передачу есть два уровня приоритета, и станция должна определить, какой уровень ей нужен. Эти уровни называются срочными и нормальными аperiodическими запросами. Понятно, что уровень срочного приоритета выше, чем уровень нормального приоритета. ВА размещает переменную А в одной из двух очередей запросов на передачу аperiodической переменной, которые являются очередями «Срочные» и «Нормальные».

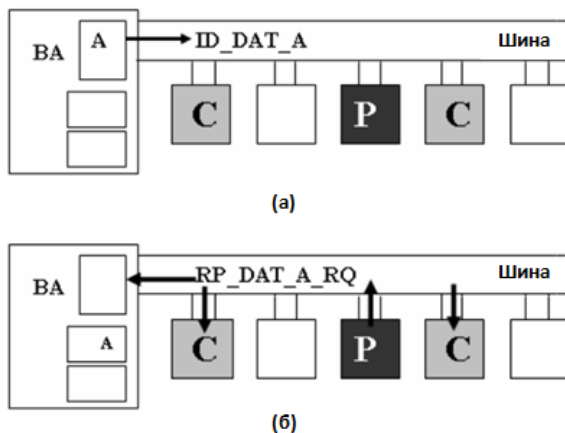


Рис.4.12. Первый этап операции с аperiodической переменной

Второй этап

Во время окна аperiodического трафика ВА отправляет кадр запроса идентификации ID_RQ производителю переменной А для передачи своего запроса, как показано на рис. 4.13,а.

Производитель переменной А ответит кадром RP_RQ, который содержит список идентификаторов аperiodических переменных, которые ему нужны. На рис. 4.13,б этот список затем сохраняется в другой очереди ВА, здесь также есть две очереди, по одной для каждого уровня приоритета. Максимальный размер списка - 64 идентификатора.

Арбитр шины может передать кадр запроса идентификации ID_RQ вскоре после того, как запрос отмечен, или позже в любом пустом микроцикле. Задержка зависит от периодической нагрузки трафика и других запросов с более высоким уровнем приоритета, которые обслуживаются ВА в это время.

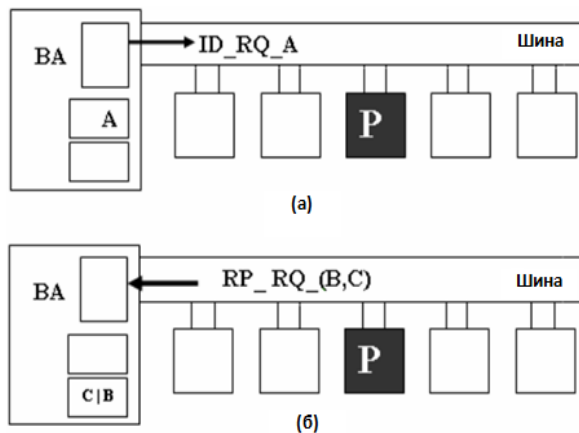


Рис.4.13. Второй этап операции с аperiodической переменной

Этап третий

Наконец, БА начинает выполнять запросы на аperiodические передачи переменных, которые хранятся в его очередях, начиная с очереди «Срочные», а затем очереди «Нормальные». БА использует тот же механизм для выполнения этих запросов аperiodических переменных, что и для периодических переменных (то есть он отправляет кадр вопроса с аperiodической переменной, затем производитель этой переменной ответит требуемым значением). БА выполняет один или несколько запросов в зависимости от времени, доступного в окне микроцикла, назначенного для этого трафика аperiodических переменных. Третий этап операции с аperiodической переменной изображен на рис. 4.14.

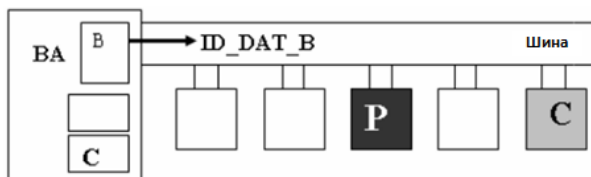


Рис.4.14. Третий этап операции с аperiodической переменной

Есть несколько замечаний по поводу этой схемы использования аperiodических переменных. Во-первых, любой узел может запрашивать передачу аperiodических переменных только в ответ на периодическую переменную, которую он производит, то есть, если узел, который не является производителем каких-либо периодических переменных, он не может

запрашивать передачу аperiodической переменной. В дополнение к этому станция, которая запрашивает аperiodическую передачу переменной, может быть производителем этой переменной, потребителем или производителем и потребителем.

Один и тот же сценарий может применяться для обоих типов аperiodических переменных (т.е. срочных или нормальных аperiodических переменных), поэтому нет необходимости демонстрировать механизм распределения нормальных аperiodических переменных.

Механизм использования среды для передачи аperiodических сообщений

Переменные периодические или аperiodические - не единственный трафик, поддерживаемый сетью WorldFIP. Кроме них есть сообщения, которые, в свою очередь, имеют два типа: подтвержденные и неподтвержденные.

Сообщения - это единственный тип трафика, который может передаваться точка-точка в дополнение к обычному способу ширококестельной передачи.

Схема, которая используется для аperiodических запросов на передачу сообщений, аналогична схеме аperiodических запросов на передачу переменных. Аperiodические сообщения имеют самый низкий приоритет в транзакциях WorldFIP.

Запрос на передачу неподтвержденного сообщения

Первый этап

Предположим, что ВА вызывает периодическую переменную В. Узел-производитель переменной В отвечает самым новым значением В, используя кадр RP_DAT, и устанавливает в этом кадре бит MSG в поле управления. Это означает, что у станции есть запрос на передачу сообщения, как показано на рисунке 4.15,а. Затем ВА сохранит идентификатор В в специальной очереди (очереди запросов сообщений), как показано на рис. 4.15,б.

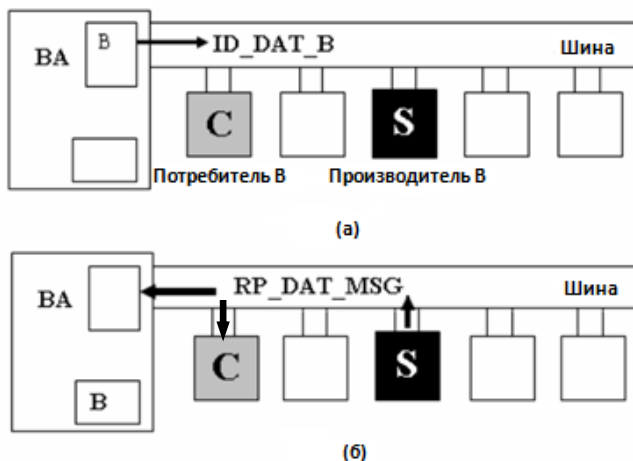


Рис.4.15. Первый этап транзакции с неподтвержденным сообщением

Второй этап

В окне аperiodического трафика и после того, как ВА завершает работу с аperiodическими переменными, ВА проверяет, нет ли больше запросов на аperiodические переменные, и начинает обрабатывать запросы сообщений.

ВА дает производителю переменной В право отправить сообщение. Другими словами, шинный арбитр предоставляет шину станции, отправив кадр ID_MSG_B на узел производителя переменной В.

Кадр сообщения должен включать в себя адреса отправителя и получателя, и он имеет тип RP_MSG_NOACK. Этот сценарий показан на рис. 4.16,а и рис. 4.16,б.

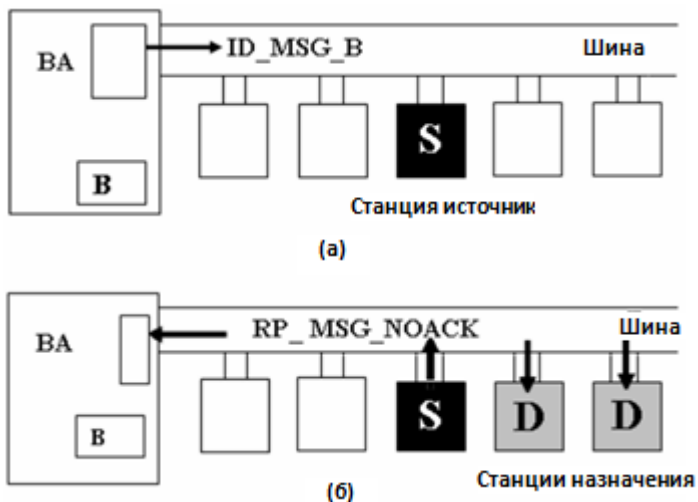


Рис.4.16. Второй этап транзакции с неподтвержденным сообщением

Третий этап

ВА ожидает получения кадра, который указывает, что транзакция передачи сообщения успешно завершена. Этот кадр, как видно на рис. 4.17, называется RP_FIN. После этого ВА снова возьмет на себя управление шиной и начнет отправлять и получать кадры от производителей переменных по шине.

Когда ВА дает станции право отправлять свое сообщение, он не знает заранее, подтверждается это сообщение или нет. Таким образом, ВА должен убедиться, что времени, которое остается в аperiodическом окне микроцикла, достаточно для обоих типов сообщений. ВА устанавливает таймер, чтобы избежать бесконечного ожидания кадра RP_FIN. Если этот таймер истечет, ВА захватит шину и начнет передавать периодические переменные.

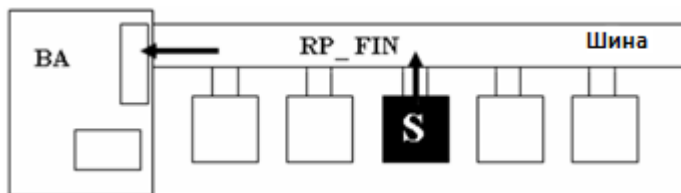


Рис.4.17. Третий этап транзакции неподтвержденного сообщения

Подтвержденный запрос передачи сообщения

Уровень канала передачи данных WorldFIP также предоставляет службы передачи подтвержденных сообщений, что делает обмен данными точка-точка более надежным. Тот же механизм, что и при передаче сообщения без подтверждения, используется для передачи сообщения с подтверждением, но с некоторыми изменениями. На втором этапе вместо ответа станции кадром `RP_MSG_NOACK` она отвечает кадром `RP_MSG_ACK`, показанным на рис. 4.18.

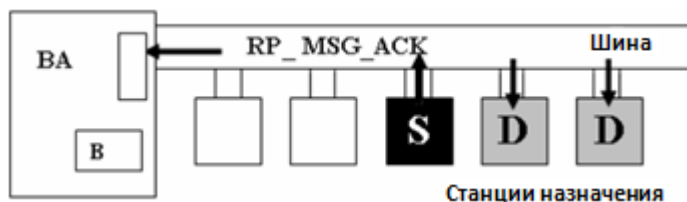


Рис.4.18. Подтверждение сообщения от отправителя

Принимающая станция должна ответить кадром подтверждения типа `RP_ACK`, который принимается станцией-отправителем сообщения, указывающего на завершение транзакции сообщения, как мы видим на рис. 4.19.

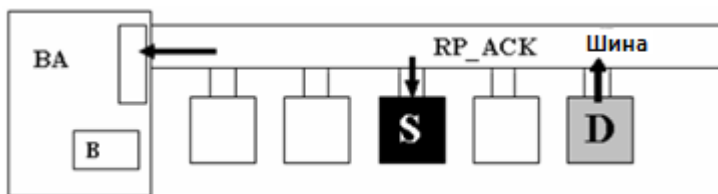


Рис..4.19. Подтверждение полученного сообщения принимающей станцией

Наконец, отправитель сообщения отправляет в BA кадр `RP_FIN` для завершения транзакции.

4.9. Прикладной слой

Прикладной уровень WorldFIP предоставляет пользователям службы чтения/записи локальных/удаленных переменных для помещения значений переменных в буферы или удаления значений переменных из буферов. Из рис. 4.1. видно, что переменные доступны через интерфейс MPS. Что касается сообщений, то они доступны через интерфейс SubMMS. MPS и SubMMS используют несколько служб. Возьмем, к примеру, MPS: он должен предоставлять пользователю службы локального/удаленного

чтения/записи, индикаторы успешной передачи/приема переменных, показанные в разделе 4.4, а также информацию об оперативности, пространственной и временной согласованности данных.

Обновление

Читая значение переменной на прикладном уровне, можно узнать некоторую качественную информацию о том, насколько свежа эта переменная. Протокол WorldFIP называет эту информацию статусом оперативности и статусом обновления. Эти статусы являются логическими (True или False) и обычно используются для синхронизации процессов на прикладном уровне между узлом-производителем и узлом-потребителем. Эти два состояния могут быть асинхронными или синхронными.

Асинхронное обновление.

Статусы обновления оцениваются прикладным уровнем производителей переменных. Для каждой создаваемой периодической переменной должно быть указано, предоставлен ли ей статус обновления. Если переменная имеет статус обновления, то также должен быть указан период производства (т.е. задержка производства), связанный с этой переменной. Рис.4.20. показывает пример, указывающий на механизм асинхронного обновления.



Рис.4.20. Асинхронный механизм обновления

Узел-производитель (слева) создает (т. е. записывает) новые значения переменных в свой буфер прикладного уровня. При каждой новой операции записи прикладной уровень производителя использует производственную задержку для расчета статуса обновления. Статус «Истина» указывает на то, что прикладной процесс производителя работает правильно. Потребители этой переменной (например, узел справа) потребляют объект, состоящий из значения V и статуса обновления S. Эти потре-

бители используют статус обновления, чтобы убедиться, что производитель учитывает задержку производства этой переменной. Рис.4.21. показывает временную диаграмму этого примера.

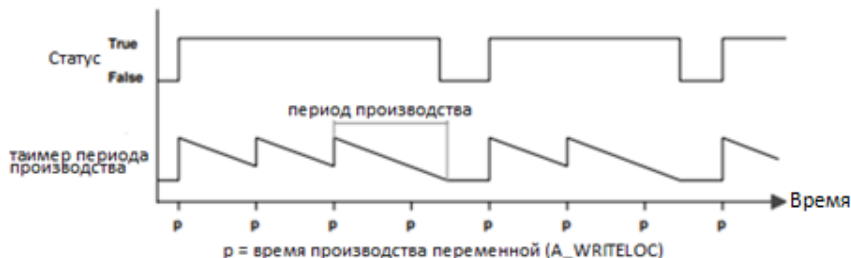


Рис.4.21. Временная диаграмма примера асинхронного обновления

Всякий раз, когда записывается новое значение переменной; прикладной уровень производителя устанавливает таймер обновления, который связан с этой переменной со значением периода производства. Статус будет "истинным" до тех пор, пока не истечет время таймера.

Синхронное обновление.

Если механизм состояния синхронного обновления должен быть выполнен для какой-либо периодической переменной, тогда должны быть указаны период производства и переменная синхронизации для оценки состояния обновления. Каждый раз, когда принимается переменная синхронизации, таймер обновления сбрасывается со значением производственного периода, и статус обновления становится «ложным». Рис.4.22 показывает пример механизма синхронного обновления.

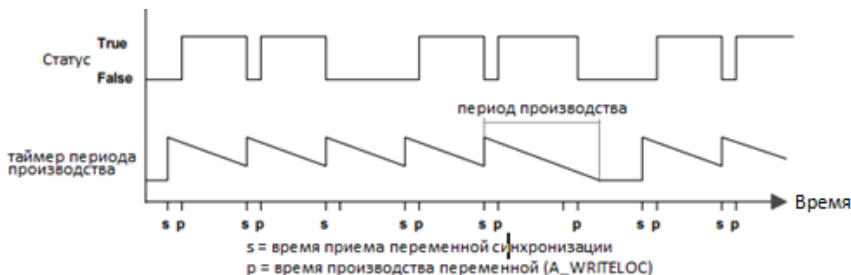


Рис.4.22. Временная диаграмма примера синхронного обновления.

Если время таймера обновления не истекло, когда прикладной уровень записывает новое значение переменной, статус обновления становится «истина». Оно остается "истинным" до тех пор, пока не истечет

время таймера обновления или пока не будет получена новая переменная синхронизации.

Оперативность

Как уже излагалось, статус оперативности - это один из двух статусов, которые указывают на актуальность значений переменных на уровне приложения. Статус оперативности также является индикатором того, что ВА учитывает периоды периодических переменных.

Асинхронная оперативность

В отличие от статусов обновления, статусы асинхронного запроса оперативности вычисляется прикладными уровнями узлов-потребителей переменных, а не узлами-производителями. Для каждой потребляемой переменной необходимо решить, будет ли она иметь статус оперативности или нет. Если требуется статус оперативности, необходимо также указать период потребления переменной. Когда статус асинхронной оперативности истинен, это означает, что механизм передачи буфера работает правильно. Рис.4.23 и 4.24 показывают пример механизма асинхронной оперативности.

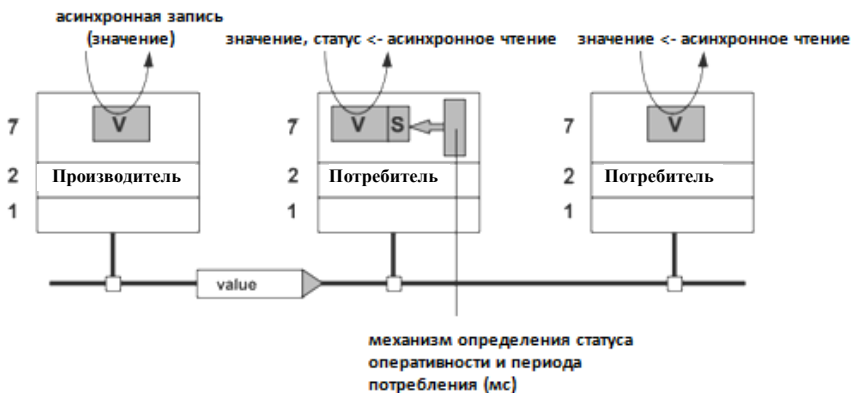


Рис.4.23. Пример механизма асинхронной оперативности

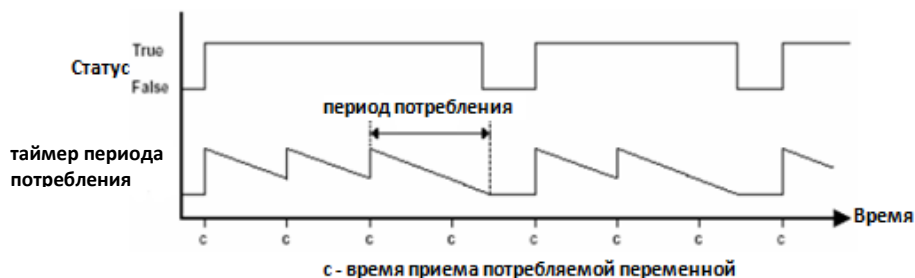


Рис.4.24. Временная диаграмма примера асинхронной оперативности.

Когда прикладной уровень абонентской станции получает новое значение переменной, он устанавливает таймер асинхронной оперативности, связанный с переменной, потребляемой со значением периода потребления. Статус оперативности становится «истинным» и остается таковым до истечения времени таймера.

В результате видно, что, когда станция-производитель хочет убедиться, соблюдает ли ВА периоды периодических переменных, она использует статус обновления.

Когда абонентская станция хочет убедиться, что ВА учитывает периоды периодических переменных, она использует статус оперативности.

Пространственная и временная согласованность

Уровень приложений WorldFIP предоставляет службу чтения списков. Список состоит из набора потребляемых переменных. Эта служба считывает на уровне приложения все значения, составляющие список. Затем она предоставляет пользователю «супер-статус», информируя его о свежести информации, которую он потребляет. Она также может предоставить статус пространственной согласованности, указывающий, что все копии этого списка идентичны во всех объектах связи, которые используют этот список.

Согласованность во времени.

Согласованность во времени - это логические состояния, которые могут дополнительно уточняться потребителями списка идентифицированных переменных. Временная согласованность может относиться к производству или потреблению. Они могут быть асинхронными или пунктуальными.

В приведенном ниже примере три датчика (поток, температура, давление) выдают значения, с которыми связано состояние обновления. Станция использует список, составленный из этих трех переменных.

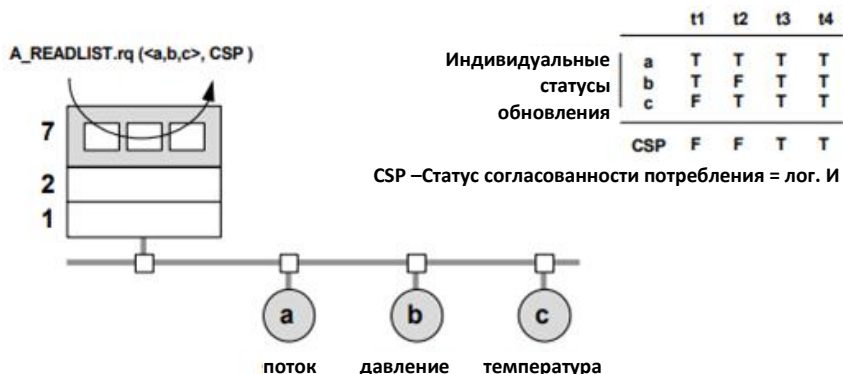


Рис.4.25. Механизм определения статуса согласованности потребления

Прикладной уровень потребителя списка использует три пары <значение, статус> и применяет логическое И к отдельным статусам обновления. Если все эти статусы истинны, статус временной согласованности производства истинен; в противном случае это ложь.

Пользователь с помощью одной операции чтения узнает, соблюдали ли все производители периоды производства, связанные с производимыми переменными.

Последовательность потребления во времени достигается применением логического И ко всем индивидуальным статусам оперативности.

Пространственная согласованность

Пространственная согласованность - это логическое состояние, которое может быть определено прикладными уровнями, которые используют список переменных. Принцип относительно прост: если все копии списка идентичны, статус истинен, иначе - ложь.

Реализация этого принципа намного сложнее и использует конструкции специальных элементарных циклов, обмена отдельными переменными пространственной согласованности и механизмы перезапуска в случае ошибочного приема.

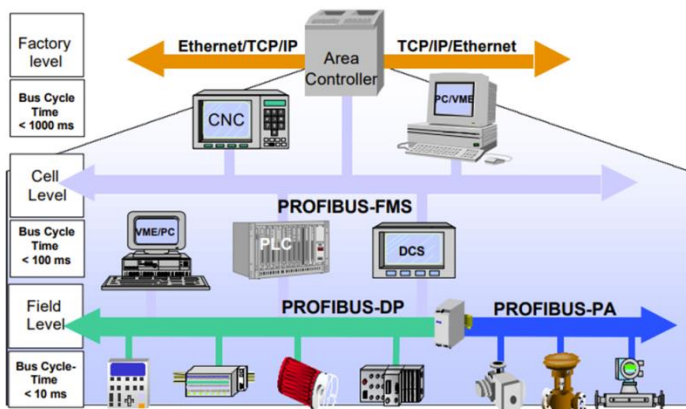
5. СИСТЕМА PROFIBUS

PROFIBUS (Process Field Bus - шина полевого процесса) - открытая промышленная сеть, прототип которой был разработан компанией Siemens для своих контроллеров Simatic. Система PROFIBUS широко распространена в Европе, особенно в машиностроении, для управления промышленным оборудованием. Сеть PROFIBUS - это комплексное понятие, она основывается на нескольких стандартах и протоколах. Сеть отвечает требованиям международных стандартов IEC 61158 и EN 50170.

5.1. Структура PROFIBUS

PROFIBUS - магистральная система с последовательной передачей данных по двухпроводной линии, использует протоколы RS-485 со скоростью до 12 Мбит/с и IEC 1158-2 с 31, 25 кбит/с во взрывоопасной среде. Кроме того, возможно использование оптической и инфракрасной связи.

Доступ к магистрали (шине) PROFIBUS реализован тремя протоколами. Он основан на протоколе PROFIBUS FMS (Field bus Message Specification) - спецификации сообщений полевой шины, который, в дальнейшем, был переработан в более простой и быстрый PROFIBUS DP (Decentralized Peripherals), а также дополнен для работы во взрывоопасных средах протоколом PROFIBUS PA (Process Automation). На рис. 5.1 показана система PROFIBUS и типичное ее быстродействие на разных уровнях применения:



CNC - Computer Numerical Control (ЧПУ)

DCS - Distributed Control System

Рис. 5.1 Структура системы PROFIBUS

В настоящее время имеется три версии PROFIBUS DP:

- PROFIBUS DP-V0 - циклическая передача данных для быстрого обмена данными;
- PROFIBUS DP-V1 - добавление специальных служб для нециклической передачи данных (программирование и диагностика устройства через PROFIBUS);
- PROFIBUS DP-V2 - общий термин для дальнейших добавлений функций PROFIBUS, таких как изохронный режим, широковещательный обмен данными, синхронизация, концепция дублирования, а также загрузка и выгрузка информации.

Принцип связи устройств, подключенных к шине, основывается на концепции ведущий – ведомый (master-slave), или ведущий – ведущий с несколькими ведущими устройствами в системе. Только один ведущий может управлять обменом данными с другими устройствами. Передача управления между ведущими осуществляется посредством передачи ему маркера (token).

Ведущий DP, класс 1:

Эти ведущие устройства являются контроллерами систем DP и назначенных им ведомых устройств DP. Обычно это контроллеры, ПЛК или системы на базе ПК.

Ведущий DP, класс 2:

Эти ведущие являются инструментами для ввода в эксплуатацию, проектирования и обслуживания. Они позволяют загружать программы в контроллеры и ведомые DP для диагностики и задания их параметров. Обычно это системы на базе ПК.

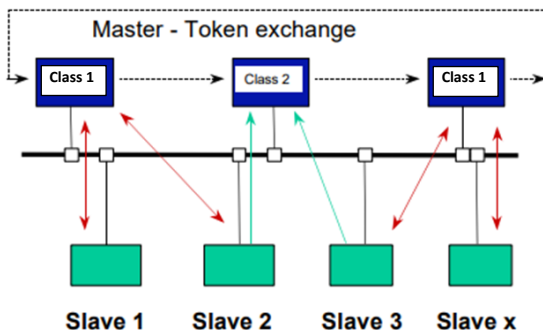


Рис. 5.2. Связь устройств в системе PROFIBUS

5.2. Технология передачи данных

PROFIBUS используется для различных областей применения и поэтому использует разные технологии передачи. PROFIBUS DP и PROFIBUS FMS используют для передачи протокол RS-485 (Recommended Standard 485). Кроме того, может использоваться оптическая передача через оптоволоконный кабель. PROFIBUS PA использует также специальную технологию передачи, известную как MBP (Manchester coded Bus Powered), которая также позволяет питать полевые устройства через шину.

Протокол RS-485

RS-485 - это симметричная (сбалансированная) двухпроводная система передачи: информация передается за счет разности напряжений между двумя проводниками, а не за счет напряжения одного проводника относительно корпуса или земли. Симметричные двухпроводные линии гораздо менее чувствительны к помехам, чем однопроводные или асимметричные линии. RS-485 - это многоточечное соединение, то есть множество станций могут отправлять и получать по одному и тому же кабелю (но только одна станция может отправлять одновременно!).

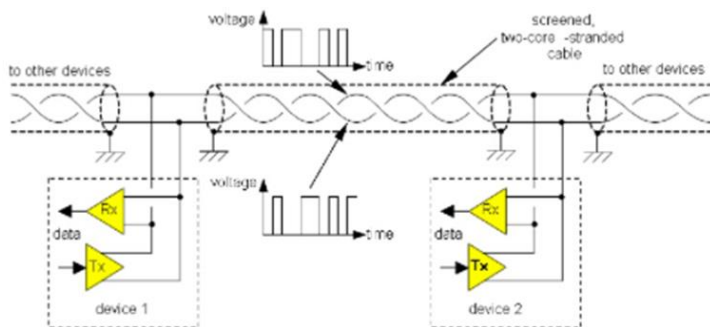


Рис.5.3. Технология передачи RS-485

Все станции PROFIBUS подключены к шине с высоким сопротивлением в состоянии ожидания. Приемные станции создают небольшую нагрузку на шину, а передающие станции, когда они не активны, переходят в состояние с высоким импедансом (третье состояние).

Поэтому на шинный кабель может наводиться любое напряжение. Это может привести к неопределенным состояниям сигнала или даже к разрушению отдельных устройств. Чтобы этого не произошло, использу-

ются активные терминаторы шины. Терминатор шины представляет собой делитель напряжения и обеспечивает уровень холостого хода приблизительно 1 В как напряжение смещения на кабеле. Для надежной работы PROFIBUS хотя бы одна оконечная нагрузка должна иметь питание. На рис.5.4 для облегчения планирования проекта оба терминатора реализованы как активные.

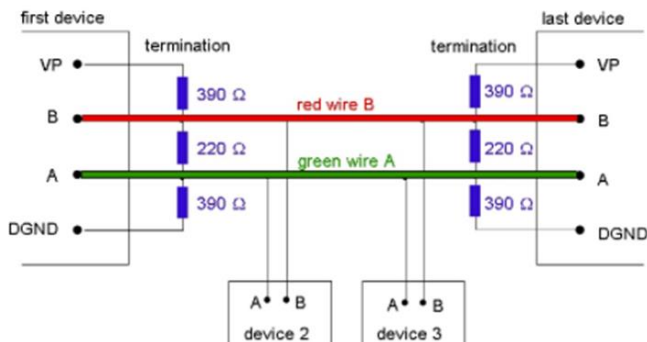


Рис. 5.4. Схема RS-485

Физический уровень

Последовательность напряжений на проводе A:

- Состояние ожидания: приблизительно +2 В
- Состояние передачи:
 - Минимальное напряжение около +1 В
 - Максимальное напряжение около +4 В



Рис. 5.5. Сигнал на проводе A

Последовательность напряжений на проводнике B сопоставима:

- Состояние ожидания: приблизительно +3 В
- Состояние передачи:

- Минимальное напряжение около + 1В
- Максимальное напряжение около + 4В



Рис.5.6. Сигнал на проводе В

Разница напряжений между высоким и низким уровнями, измеряемая между В и А, должна составлять от 4 до 7 В. Значения положительного и отрицательного напряжения должны быть примерно равными. На практике они могут отличаться на 0,5 В. Уровень холостого хода должен составлять приблизительно 1 В.

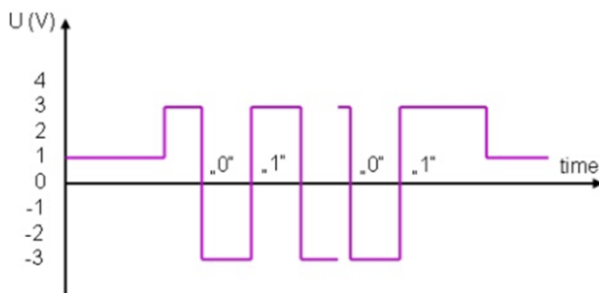


Рис.5.7. Дифференциальный сигнал В-А

Состояние ожидания определяется активным терминатором шины.

На шине PROFIBUS PA для передачи бита используется манчестерский код (рис. 5.8.).

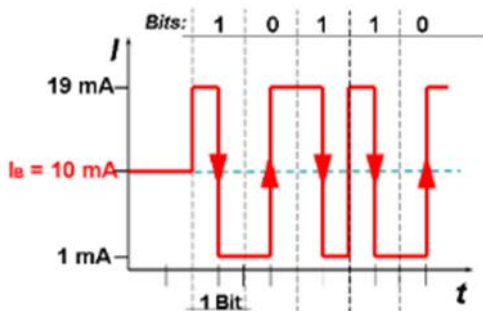


Рис. 5.8. Манчестерский код

При манчестерском кодировании каждый бит кодируется как изменение уровня. В этом случае положительное изменение - значение «0», а отрицательное изменение - значение «1». Если необходимо, между битами произойдет дополнительное изменение уровня. Это показано на рис. 5.8.

Манчестерский код имеет то преимущество, что тактовую частоту битов можно получить непосредственно из сигнала, так как каждый бит повторно синхронизируется. Это позволяет передавать данные без дополнительных сигналов синхронизации. Этот тип передачи называется синхронным.

Поскольку каждый бит всегда состоит из равных пропорций положительного и отрицательного сигнала, среднее значение этого сигнала всегда постоянно. Это среднее значение используется для питания устройств. Отсюда и название MBP (Manchester coded Bus Powered).

В MBP терминатор шины имеет совершенно другую структуру по сравнению с оконечной нагрузкой шины RS-485. Оконечная нагрузка шины MBP состоит из сопротивления и емкости, последовательно включенной между двумя проводниками (см. рис.5.9):

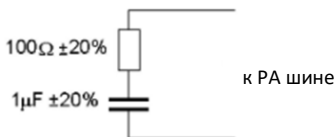


Рис. 5.9. Терминатор шины MBP

Этот терминатор пассивный и не требует специального источника питания. Оконечная нагрузка шины должна применяться только к основной линии, а не к отдельным отрезкам. Большинство соединителей сегментов имеют встроенные терминаторы шины, которые, следовательно, должны быть на последнем или самом дальнем удаленном полевом устройстве, либо в последней распределительной коробке или ответвительной розетке.

5.3. Передача данных

В PROFIBUS за единообразную передачу данных отвечают функции уровня канала передачи данных (Fieldbus Data Link - FDL). Эти функции идентичны для всех профилей PROFIBUS.

Табл. 5.1. Функции PROFIBUS

Сервис	Функция	DP-V0	DP-V1	DP-V2	FMS
SDN	Отправка данных без подтверждения	X	X	X	X
SDA	Отправка данных с подтверждением	X			X
SRD	Отправка и запрос данных	X	X	X	X
CSRD	Циклическая отправка и запрос данных				X
MSRD	Отправка и запрос данных с многоадресным ответом			X	
CS	Синхронизация		X	X	

Управление доступом к среде передачи

Шина - это обычное средство передачи данных. Управление доступом к среде передачи (MAC - Media Access Control) служит для обеспечения того, чтобы не более одной ведущей станции использовали шину одновременно. В PROFIBUS есть два типа пользователей шины: активные ведущие и пассивные ведомые. Ведущие получают разрешение на доступ к шине на определенное время. Эта авторизация доступа явно передается от одного ведущего к другому с помощью токен-телеграммы. Этот процесс называется передачей токена. Ведомые устройства могут отправить телеграмму по шине только один раз - в ответ на запрос от ведущего. Это определяет собой связь ведущий-ведомый. Таким образом, процедура доступа PROFIBUS FDL представляет собой гибрид ведущего-ведомого и передачи токена (рис.5.10).

Принцип ведущий-ведомый

Минимальная конфигурация сети PROFIBUS состоит как минимум из одного ведущего и нескольких ведомых устройств. Ведущее устройство может отправлять телеграммы в произвольной последовательности всем или отдельным ведомым устройствам. При использовании функции SRD ведущее устройство при передаче одновременно запрашивает ответ.

Принцип передачи токенов

Если в сети PROFIBUS присутствует более одного ведущего, они передают разрешение на передачу друг другу с помощью специальной телеграммы «токен». Маркер всегда передается в порядке возрастания адреса станции.

Независимо от того, какой ведущий имеет токен, он может использовать связь для адресации любых других станций - ведущих и ведомых.

На рис. 5.10 показана типичная структура PROFIBUS, состоящая из 2 активных ведущих и 5 пассивных ведомых станций. Последовательность передачи токенов формирует логическое кольцо, то есть токен передается в возрастающем адресном порядке от одной активной станции к другой. Наконец, последняя станция передает токен обратно первой.

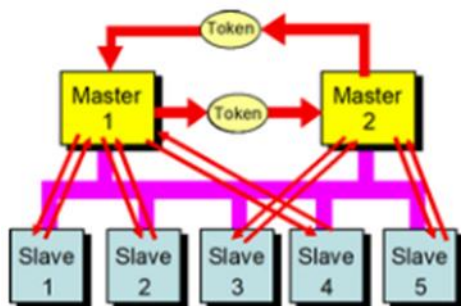


Рис.5.10. Принцип гибридного доступа (ведущий-ведущий, ведущий-ведомый)

Конечный автомат

Все необходимые состояния контроллера MAC для хорошо регулируемой обработки связи можно отследить на рис. 5.11. Соответствующие состояния подробно описаны в стандарте (IEC 61158-4-3, раздел 8.2), здесь дается только обзор.

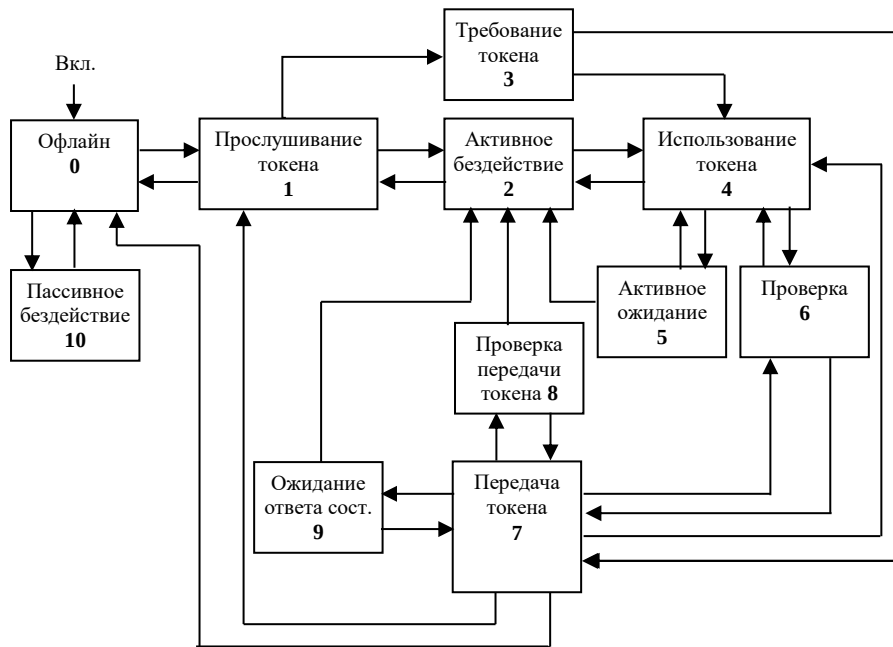


Рис. 5.11. Алгоритм доступа к шине

После включения рабочего питания как активные (ведущие), так и пассивные (ведомые) пользователи шины оказываются в автономном состоянии, выполняют самотестирование и загружают рабочие параметры, необходимые для связи. Затем пассивные станции переходят в состояние «Пассивное бездействие» (10), в котором их приемник включен, и они пассивно слушают линию. Если телеграммы распознаются с их собственным адресом станции, они должны быть должным образом подтверждены станцией.

Активные пользователи шины переходят в состояние «Прослушивание токена» после состояния «Офлайн», если они готовы к включению в логическое кольцо. В состоянии «Прослушивание токена» станция прослушивает телеграммы на линии шины и из полученных токен-телеграмм формирует список уже активных станций (LAS: список активных станций). После того как список создан, станция должна дожидаться, пока она не будет адресована предшествующей ей станцией с «запросом статуса FDL» и «приглашена» в логическое кольцо. «Новичок» подтверждает сообщение «готов для кольца» и переходит в состояние активного ожидания ($0 > 1 > 2$). Это делает его официальным абонентом кольца. В этом состо-

янии он прослушивает линию, подтверждает и отвечает так же, как пассивная станция в состоянии пассивного ожидания. Активная станция после получения направленной ей токен-телеграммы переходит в состояние «Использование токена». Если активная станция не получает токен-телеграмму в течение длительного периода, она переходит в состояние «Требование токена» ($0 > 1 > 3$), после чего логическое кольцо инициализируется снова или повторно ($3 > 7 > 0$), в последнем случае LAS останется действительным.

Обычно, активная станция получает телеграмму с токеном в течение определенного времени, так что, проверив оставшееся время хранения токена, доступное в состоянии проверки времени доступа, циклы сообщений могут обрабатываться в состоянии использования токена (4). Если функция, используемая для связи, требует ответа, станция перейдет в состояние ожидания ответа данных и будет ожидать ответа в течение периода времени слота (в зависимости от скорости передачи данных). Если ожидаемый ответ не приходит, после нескольких повторных попыток пользователю будет отправлено сообщение об ошибке.

Циклы сообщений могут обрабатываться до истечения времени хранения токена. Затем станция перейдет в состояние «Передача токена» (7), в котором токен будет передан следующей активной станцией. Эта передача отслеживается в состоянии «Проверка передачи токена» (8). Если что-то пойдет не так с прохождением токена, например, из-за того, что следующая станция не представляет себя, станция переключится в состояние «Ожидания ответа состояния» (9), где она ожидает сигнала подтверждения в течение определенного времени. Если этот сигнал не получен, вызов повторяется в состоянии «Передача токена» (7). Если станция получает другую телеграмму (ситуация с несколькими токенами), она переходит в состояние «Активное ожидание» (5). После передачи токена, обычно без проблем, станция окажется в состоянии активного ожидания до следующего приема токена.

5.4. Протокол передачи данных

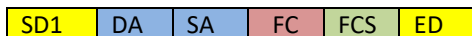
Телеграммы и последовательности телеграмм в PROFIBUS едины. Для практической реализации существуют две формы для UART (Universal Asynchronous Receiver-Transmitter) и MBP (Manchester coded Bus Powered). Разница между двумя версиями заключается в кодировании отдельных бит и в структуре телеграммы.

Форматы телеграмм

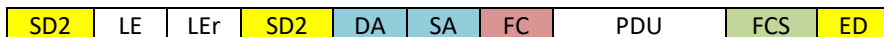
Имеются следующие форматы телеграмм:

- Телеграмма без данных
- Телеграмма с переменной длиной данных
- Телеграмма с фиксированной длиной данных
- Токен-телеграмма
- Для ответов определена короткая телеграмма.

Телеграмма без поля данных:

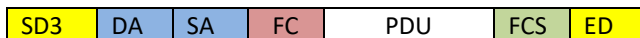


Телеграмма переменной длины:



PDU имеет переменную длину от 1 до 246 байтов.

Телеграмма с фиксированной длиной данных:



PDU имеет фиксированную длину 8 байтов.

Токен-телеграмма



Краткое подтверждение:



В телеграммах используются следующие обозначения:

PDU - данные

SD - начальный разделитель, который может принимать четыре разных значения в зависимости от типа телеграммы.

ED - конечный разделитель отмечает конец телеграммы.

DA - адрес назначения

SA - адрес источника

LE - длина

LEr -длина повторяется для безопасности

FC - код функции

FCS - контрольная сумма телеграммы

Каждое поле телеграммы имеет размер 1 байт.

Адресные байты

Два байта адреса в заголовке телеграммы (телеграммы запроса, подтверждения и ответа) содержат адрес назначения станции (DA) и адрес источника (SA).

Адрес 127 зарезервирован как глобальный адрес для широковещательных или многоадресных сообщений (телеграммы всем пользователям шины или группам пользователей, разрешено только при отправке данных без подтверждения, SDN).

Байты адреса телеграммы запроса должны быть отражены в телеграмме ответа в обратном порядке, то есть байт SA телеграммы ответа содержит адрес станции назначения, а байт DA - адрес исходной станции запроса.

Длина

Два байта эквивалентной длины в заголовке телеграммы переменного формата содержат количество байтов информации в телеграмме. К ним относятся: DA, SA, FC и PDU. Диапазон значений от 4 до 249, так что в PDU телеграммы можно передать не более 246 байт. Значение меньше 4 недопустимо, поскольку телеграмма должна содержать, по крайней мере, один DA, SA, FC и байт данных. Таким образом, самая длинная телеграмма составляет 255 байтов.

Контрольная сумма

Для телеграммы без данных (SD1) контрольная сумма должна быть образована арифметической суммой без учета переноса DA, SA и FC без ограничителя начала (SD) и ограничителя конца (ED). Для телеграммы фиксированной длины с данными (SD3) и формата переменной длины (SD2) контрольная сумма также должна включать байты данных (PDU).

Последовательности телеграмм и временные соотношения

Между последовательностью телеграмм существует пауза синхронизации SYN, состоящая из 33 бит.

В PROFIBUS необходимо соблюдать определенные временные соотношения, чтобы гарантировать правильную работу последовательности телеграмм (см. рис.5.12).

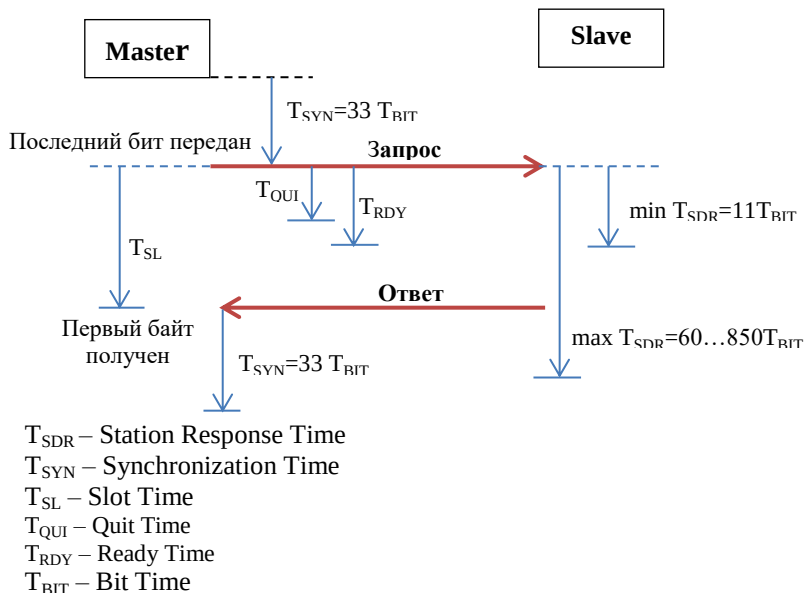


Рис. 5.12. Временные соотношения

После паузы синхронизации, длящейся T_{SYN} , чтобы отметить начало последовательности, телеграмма запроса отправляется ведущим ведомому. Ведомое устройство должно начать свой ответ не раньше минимального T_{SDR} (SDR- Station Delay of Responder), но не позже максимального T_{SDR} . Минимальное время T_{SDR} включает в себя как минимум время задержки линейного усилителя и повторителя T_{QUI} и время готовности ведомого устройства T_{RDY} . В современных системах PROFIBUS-DP $\min T_{SDR} = 11T_{BIT}$. Более медленные ведомые устройства PROFIBUS-FMS имеют минимальные значения T_{SDR} до $255 T_{BIT}$. Ведущее устройство ожидает ответа от ведомого в течение максимального периода T_{SL} (SL - слот). Это время определяется максимальным временем ответа T_{SDR} .

Табл. 5.2 Значения по умолчанию для времен в тактах T_{BIT}

Параметр	Битрейт, Кбит/с					
	$\leq 187,5$	500	1500	3000	6000	12 000
T_{QUI}	0	0	0	3	6	9
T_{RDY}	≤ 11	≤ 11	≤ 11	≤ 11	≤ 11	≤ 11
$\min T_{SDR}$	11	11	11	11	11	11
$\max T_{SDR}$	60	100	150	250	450	800
T_{SL}	100	200	300	400	600	1000

5.5. Топология сети

Все устройства соединяются в шинную структуру (магистраль). Она может состоять из нескольких сегментов. В одном сегменте шины можно подключить до 32 станций.

В число этих станций входят не только ведущие и ведомые устройства DP, но также отдельные повторители или оптические преобразователи. Через PROFIBUS можно адресовать до 126 станций. Максимальную сеть PROFIBUS можно построить только с помощью повторителей:

- В любом первом или последнем сегменте можно подключить максимум 31 станцию и один повторитель.
- В любом среднем сегменте можно подключить до 30 станций и 2 повторителя.
- Таким образом, для максимальной сети потребуются, как минимум, 4 повторителя.

На рис. 5.12 показана структура с повторителями и терминаторами T, подключенными к питанию:

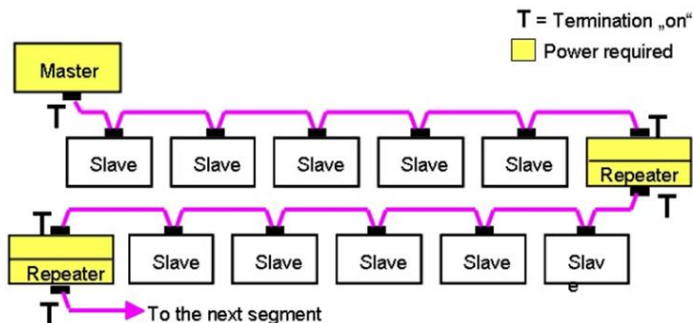


Рис.5.13. Структура сети с одним ведущим устройством

Адресация станций

Каждая станция PROFIBUS должна иметь уникальный адрес для связи. Эти адреса PROFIBUS кодируются в один байт и составляют диапазон от 0 до 127. Однако отдельные значения адресов зарезервированы и не могут быть присвоены устройствам.

Адрес	Использование
0	Зарезервировано, как правило, для диагностических инструментов или устройств программирования.
1 ... n	Адреса ведущих станций должны начинаться с наименьшего адреса. Таким образом, один ведущий будет иметь адрес 1. Дополнительные ведущие станции будут иметь адреса 2, 3 ... и т.д.
n ... 125	Это означает, что в сети PROFIBUS с одним ведущим устройством не более 124 адресов будет оставлено свободными для ведомых станций.
126	Зарезервирован как адрес доставки (по умолчанию) для станций, адрес которых может быть изменен через шину.
127	Зарезервирован для адресации всех или групп (широковещательная рассылка) и поэтому не может быть установлен для одной станции.

Шинный кабель

В соответствии с технологией передачи PROFIBUS DP/FMS с RS-485, применяется двухжильный, скрученный и экранированный кабель. В настоящее время используется кабель типа А со следующими электрическими параметрами:

Табл. 5.3. Электрические параметры кабеля типа А

Волновое сопротивление R_w	135 ... 165 Ом
Емкость на единицу длины C'	<30 пФ/м
Сопротивление на единицу длины R'	110 Ом/км
Диаметр d	0,64 мм
Сечение жилы q	> 0,34 мм ²

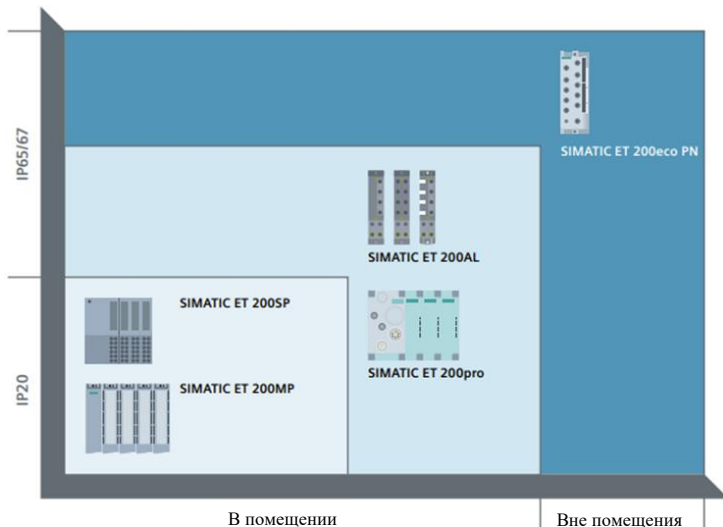
Максимальная длина линии сегмента зависит от скорости передачи и типа кабеля:

Табл. 5.4. Максимальная длина кабеля на один сегмент

Скорость передачи в Кбит/с	Макс. длина, тип кабеля А
45,45	1200 м
93,75	1200 м
187,5	1000 м
500	400 м
1500	200 м
3000	100 м
6000	100 м
12000	100 м

6. СИСТЕМА SIMATIC ET 200

Модульные системы распределенного ввода-вывода, разработанные фирмой Siemens под торговой маркой SIMATIC, получившие название из слов **Siemens and Automatic**, широкого спектра применения в зависимости от места использования и степени защиты¹ показаны на рис. 6.1:



¹ IP20 – Защита от частиц > 12,5 мм; IP65/67 – от пыли полностью и струй воды под небольшим давлением/временного погружения в воду до 1 м.

Рис 6.1. Системы ET 200

6.1. Система распределенного ввода-вывода SIMATIC ET 200SP

SIMATIC ET 200SP - это масштабируемая и гибкая система распределенного ввода-вывода для подключения через полевую шину сигналов автоматизированного процесса к контроллеру более высокого уровня.

Преимущества системы для пользователей

Маркетинговая служба производителя отмечает следующие особенности системы:

Простота в использовании

- Компактные модули, электромонтаж с одножильным или многожильным подключением.

- Экономия времени за счет системы подключения без инструментов с помощью вставных клемм.
- Адаптация конфигурации для будущих расширений посредством интегрированного управления системы.

Компактность конструкции

- Небольшой размер и высокая вариативность благодаря масштабируемости.
- Максимальная прозрачность в ограниченном пространстве благодаря инновационной системе маркировки.
- Интегрированный в систему источник питания.

Интегрированная безопасность

- Простая интеграция отказоустойчивых процессоров и модулей.
- Настройка всех F-параметров через программное обеспечение.

Высокая производительность

- Исохронная система PROFINET IO с профилями PROFIsafe и PROFInergy.

Стандартная связь

- PROFINET IO (промышленный Ethernet)
- PROFIBUS DP
- ET-соединение (с шиной объединительной панели)
- AS-интерфейс (Actuator Sensor Interface, ASi, предназначен для подключения датчиков)
- IO-Link (промышленный коммуникационный интерфейс)
- Point-to-point (RS232, RS485)

Энергоэффективность

- PROFInergy в качестве интегрированной функции позволяет автоматизировать энергопотребление оборудования на производстве.

Процессор

- Интерфейс PROFINET с 3-мя портами
- Контроллер ввода-вывода
- Интеллектуальное устройство (I-device)
- Коммуникационный модуль CM DP для подключения к PROFIBUS DP

Стартер двигателя

- Простая интеграция пускателей двигателей с защитой от перегрузки и короткого замыкания.
- Компактный дизайн с максимальной переключаемой мощностью двигателя до 5,5 кВт.

Широкий ассортимент модулей

- Модули для функций: счета, позиционирования, взвешивания, измерения электрических сигналов.

Область применения

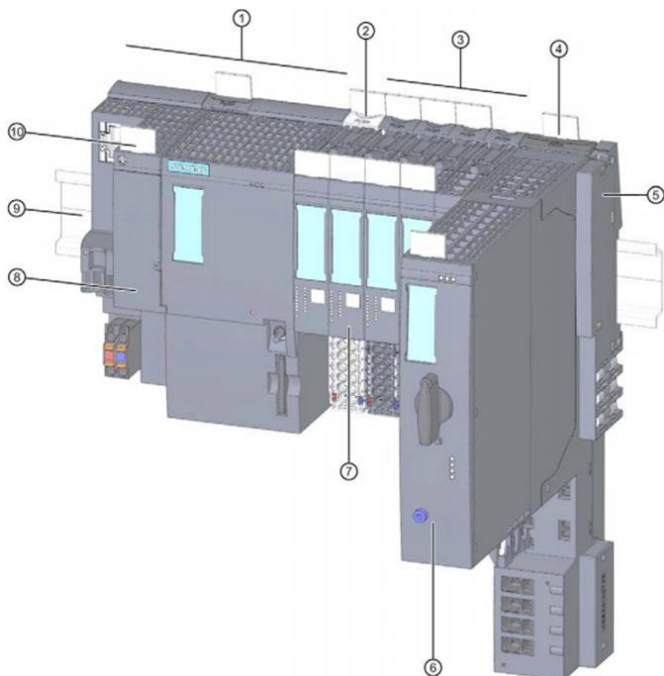
Благодаря своей многофункциональности система распределенного ввода-вывода SIMATIC ET 200SP подходит для широкого спектра приложений. Его масштабируемая конструкция позволяет адаптировать конфигурацию к местным требованиям. Доступны различные процессоры/интерфейсные модули для подключения к PROFINET IO или PROFIBUS DP.

SIMATIC ET 200SP с центральным процессором (ЦП) обеспечивает интеллектуальную предварительную обработку для облегчения работы контроллера верхнего уровня. ЦП также можно использовать как автономное устройство.

Используя отказоустойчивые ЦП, можно реализовать приложения для разработки безопасных систем. Конфигурирование и программирование для обеспечения безопасности происходит так же, как и для стандартных процессоров.

SIMATIC ET 200SP имеет степень защиты IP20 и предназначен для установки в стойке управления.

Конфигурация



- ① ЦП/интерфейсный модуль
- ② Светлый базовый блок с подачей напряжения питания
- ③ Базовые блоки темного цвета для расширения потенциальной группы
- ④ Базовый блок для пускателя двигателя
- ⑤ Сервисный модуль (входит в состав модуля ЦП/интерфейса)
- ⑥ Пускатель двигателя
- ⑦ Модуль ввода/вывода
- ⑧ Шинный адаптер
- ⑨ Монтажная рейка
- ⑩ Идентификационная этикетка

Рис. 6.2. Пример конфигурации ET 200SP

Система распределенного ввода-вывода SIMATIC ET 200SP устанавливается на монтажной рейке. Она состоит из:

- ЦП/интерфейсного модуля,
- до 64 модулей ввода/вывода, которые могут быть подключены к базовым блокам в любой комбинации,
- до 31 пускателя двигателя,

- сервисного модуля, завершающего настройку конфигурации ET 200SP.

6.2. Отказоустойчивые системы автоматизации и отказоустойчивые модули

Системы безотказной автоматики (F-системы) используются в системах с повышенными требованиями к безопасности. F-системы управляют процессами, обеспечивая их безопасное состояние сразу после отключения. Другими словами, F-системы управляют процессами, в которых неожиданное отключение не представит опасности для людей или окружающей среды.

Интегрированная безопасность

Интегрированная безопасность - это концепция безопасности для автоматизации процессов и приводной техники от Siemens.

В системах безопасности используются проверенные технологии и системы автоматизации. Интегрированная безопасность включает в себя полную последовательность действий по обеспечению безопасности, начиная от датчиков, исполнительных механизмов и отказоустойчивых модулей до контроллера, и включая безопасную связь через стандартные полевые шины. Приводы и контроллеры выполняют задачи обеспечения безопасности в дополнение к своим фактическим функциям.

Отказоустойчивые модули

Основное различие между отказоустойчивыми модулями (F-модулями) и стандартными модулями заключается в том, что они имеют внутреннюю двухканальную конструкцию. Это означает, что два интегрированных процессора контролируют друг друга, автоматически проверяя входные и выходные цепи, и в случае неисправности переводят отказоустойчивый модуль в безопасное состояние.

F-CPU связывается с отказоустойчивым модулем через безопасный профиль шины PROFIsafe.

Отказоустойчивые пускатели двигателей.

Отказоустойчивые пускатели двигателей обеспечивают безопасное отключение двигателя. Отказоустойчивые пускатели двигателей не являются узлами PROFIsafe. Пускатели двигателей работают вместе с модулями повышенной безопасности системы ET 200SP.

Область применения ET 200SP с отказоустойчивыми модулями ввода/вывода

При использовании системы распределенного ввода-вывода ET 200SP с отказоустойчивыми модулями меняют традиционные средства техники безопасности. Сюда входит применение коммутационных аппаратов для аварийного останова, защитных дверных датчиков, двуручного управления и т. д.

6.3. SIMATIC F-система ET 200SP

На рис. 6.3 показан пример конфигурации F-системы SIMATIC с системой распределенного ввода-вывода ET 200SP и PROFINET IO. Для линий PROFINET IO можно использовать медный кабель, оптоволокно или WLAN.

Особенности PROFINET:

- PROFINET является открытым стандартом Industrial Ethernet для систем автоматизации.
- PROFINET базируется на Industrial Ethernet.
- PROFINET использует TCP/IP и IT стандарты.
- PROFINET поддерживает обмен данными через Ethernet в реальном масштабе времени.
- PROFINET поддерживает простую интеграцию сетей полевого уровня.
- PROFINET поддерживает профиль PROFIsafe, в том числе, и в IWLAN, предназначенный для минимизации вероятности неправильного функционирования системы управления.

В конфигурации ET 200SP можно комбинировать отказоустойчивые и не отказоустойчивые модули ввода-вывода.

Отказоустойчивый контроллер ввода-вывода (F-CPU) обменивается данными, связанными как с безопасностью, так и не связанными с безопасностью, с отказоустойчивыми и не отказоустойчивыми модулями ET 200SP.

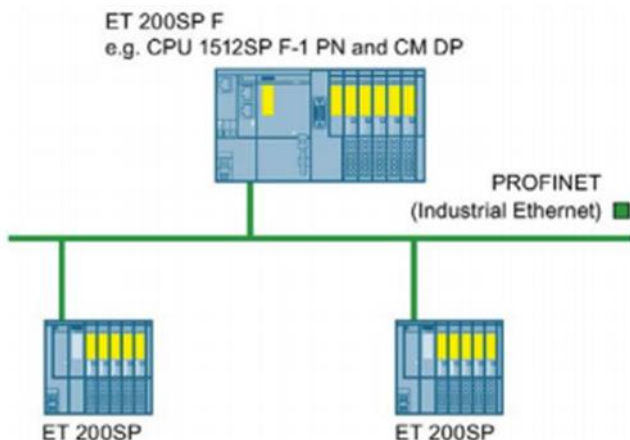


Рис. 6.3. Пример конфигурации отказоустойчивой системы автоматизации SIMATIC Safety

Отказоустойчивые модули ввода-вывода ET 200SP

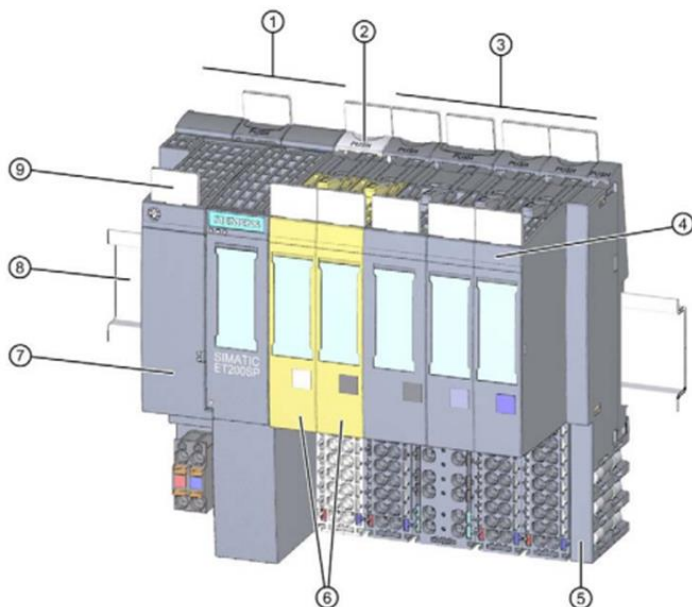
Для системы распределенного ввода-вывода ET 200SP доступны следующие отказоустойчивые модули ввода-вывода:

- Отказоустойчивые силовые модули используются для подачи напряжения потенциальной групповой нагрузки и безопасного отключения напряжения нагрузки для отказоустойчивых модулей вывода.
- Отказоустойчивые цифровые модули ввода определяют состояния сигналов датчиков, связанных с безопасностью, и отправляют соответствующие кадры безопасности в F-CPU.
- Отказоустойчивые цифровые модули вывода используются для безопасных процедур отключения с защитой от короткого замыкания и перекрестного замыкания вплоть до привода.

Пускатели двигателей повышенной безопасности ET 200SP

Отказоустойчивые пускатели двигателей используются для безопасного отключения нагрузки двигателя.

Пример конфигурации с отказоустойчивыми модулями ввода/вывода показан на рис.6.4.



- ① Интерфейсный модуль
- ② Светлый базовый блок с подачей напряжения питания
- ③ Базовые блоки темного цвета для дальнейшего продолжения потенциальной группы
- ④ Модуль ввода/вывода
- ⑤ Сервисный модуль (поставляется с интерфейсным модулем)
- ⑥ Отказоустойчивые модули ввода/вывода
- ⑦ Шинный адаптер
- ⑧ Монтажная рейка
- ⑨ Идентификационная этикетка

Рис. 6.4. Пример конфигурации ET 200SP с отказоустойчивыми модулями ввода/вывода

6.4. Системные компоненты

Основные компоненты системы ET 200SP:

CPU/F-CPU

Центральный процессор запускает программу пользователя. F-CPU, кроме того, выполняет программу безопасности. Он может использоваться как IO-контроллер или I-Device на шине PROFINET IO или как автономный CPU, обеспечивающий обмен данными с модулями ввода/вывода посредством шины объединительной панели.

Дополнительно в системе PROFIBUS DP может использоваться как ведущее устройство или ведомое в сочетании с коммуникационным модулем CM DP.

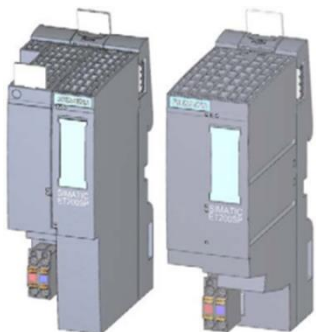
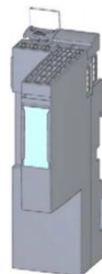
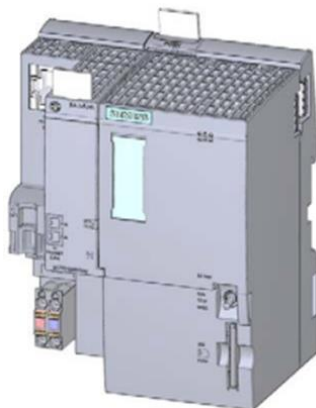
Модуль CM DP

Коммуникационный модуль CM DP связывает CPU с шиной PROFIBUS DP, которая использует интерфейс RS-485.

Интерфейсный модуль PROFINET IO

Может использоваться как устройство ввода-вывода на шине PROFINET IO

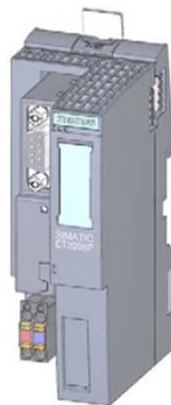
Связывает ET 200SP с контроллером ввода-вывода для обмена данными с модулями ввода/вывода через шину объединительной панели.



Интерфейсный модуль PROFIBUS DP

Может использоваться как ведомое устройство на шине PROFIBUS DP

Связывает ET 200SP с ведущим DP-устройством для обмена данными с модулями ввода/вывода через шину объединительной панели.



Шинные адаптеры BA

Шинные адаптеры позволяют осуществить свободный выбор технологии подключения для PROFINET IO. Для модулей ЦП/интерфейсных модулей PROFINET доступны следующие варианты:

-для стандартного разъема RJ45 (BA 2×RJ45) ①

-для прямого подключения шинного кабеля (BA 2×FC) ②

-для оптоволоконного кабеля POF/PCF (BA 2×SCRJ) ③

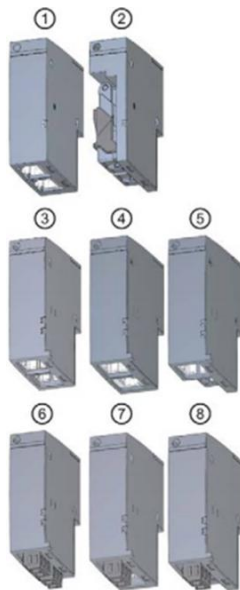
-в качестве конвертера для оптоволоконного кабеля POF/PCF ⇔ стандартный штекер RJ45 (BA SCRJ/RJ45) ④

-в качестве конвертера для оптоволоконного кабеля POF/PCF ⇔ прямое подключение шинного кабеля (BA SCRJ/FC) ⑤

-для оптоволоконного кабеля (BA 2×LC) ⑥

-в качестве конвертера для оптоволоконного кабеля ⇔ стандартный штекер RJ45 (BA LC/RJ45) ⑦

-в качестве конвертера для оптоволоконного кабеля ⇔ прямое подключение шинного кабеля (BA LC/FC) ⑧



RJ-Registered Jack
FC-Ferrule (штекер) Connector
SC-Square/subscriber Connector
LC-Link Connector
POF/PCF - Polymer Optical Fiber/
Polymer Clad Fiber

Базовые блоки BaseUnit

Базовые блоки обеспечивают механическое и электрическое соединение модулей ET 200SP.

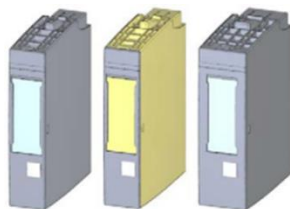
Для подключения модулей ввода/вывода или пускателя двигателя к базовым блокам доступны подходящие устройства разного назначения.



Модуль ввода/вывода/отказоустойчивый модуль ввода/вывода

Модуль ввода/вывода определяет функцию на его клеммах. Контроллер определяет текущее состояние процесса через подключенные датчики и исполнительные механизмы и соответственно реагирует на них. Модули ввода/вывода делятся на следующие типы:

- цифровой вход (DI, F-DI)
- цифровой выход (DQ, F-DQ PM, F-DQ PP, F-RQ)
- аналоговый вход (AI)
- аналоговый выход (AQ)
- технологический модуль (TM)
- коммуникационный модуль (CM)



7. СИСТЕМА NI CompactRIO

Разработчик системы National Instruments (NI) является одним из мировых лидеров в технологии виртуальных приборов и в разработке и изготовлении аппаратного и программного обеспечения для систем автоматизированного управления. Традиционно популярны у потребителей такие семейства продуктов NI, как устройства сбора данных (data acquisition), приборные интерфейсы (instrument control), системы управления электроприводами и машинного зрения (machine vision).

Система управления и мониторинга NI CompactRIO с использованием программного обеспечения для проектирования систем NI LabVIEW вместе с модулями LabVIEW RealTime и LabVIEW FPGA используются в самых разных приложениях. Диапазон использования: от управления ускорителями заряженных частиц, таких как Большой адронный коллайдер ЦЕРН, до тестирования электронных блоков управления двигателем (ECUs), от адаптивного управления бурением нефтяных скважин до высокоскоростного мониторинга вибраций для профилактического обслуживания.

7.1. Архитектура CompactRIO

CompactRIO - это жесткая настраиваемая встроенная система, содержащая три компонента: процессор, работающий под управлением операционной системы реального времени (RTOS), реконфигурируемая FPGA (Field-Programmable Gate Array - программируемая вентильная матрица) и сменные промышленные модули ввода-вывода. На рис.7.1 показана связь процессора через FPGA с модулями АЦП, ЦАП, цифрового ввода и вывода с преобразованием сигналов на входе/выхода сенсоров и актуаторов и на входе/выходе модулей. Процессор реального времени обеспечивает надежное, предсказуемое поведение и отлично выполняет анализ и вычисления с плавающей точкой, в то время как FPGA превосходно справляется с небольшими задачами, требующими высокоскоростной логики и точного тайминга. Часто приложения CompactRIO используют человеко-машинный интерфейс (HMI), который предоставляет оператору графический интерфейс пользователя (GUI) для мониторинга состояния системы и настройки рабочих параметров.

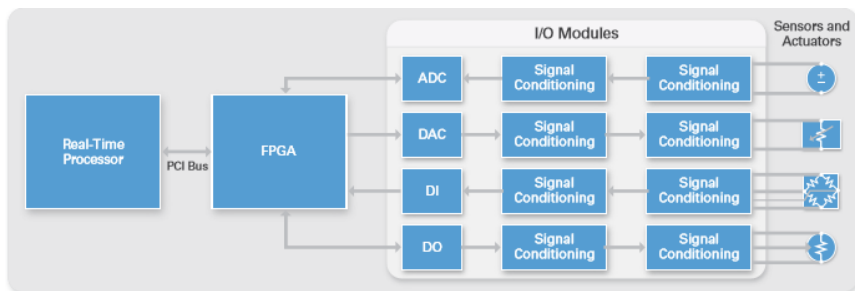


Рис. 7.1. Реконфигурируемая архитектура встроенной системы

LabVIEW

Среда LabVIEW (Laboratory Virtual Instrument Engineering Workbench: среда виртуальных инструментальных средств) - это графическая среда программирования, используемая для разработки сложных систем управления с использованием графических значков и проводов, напоминающих блок-схемы. Она обеспечивает интеграцию с тысячами аппаратными устройствами и предоставляет сотни встроенных библиотек для программ управления, анализа и визуализации данных - все для быстрого создания пользовательских систем. Платформа среды LabVIEW предназначена для нескольких целевых объектов и операционных систем, а в случае CompactRIO, LabVIEW можно использовать для доступа и интеграции всех компонентов архитектуры реконфигурируемого ввода-вывода (RIO).

Некоторые замечания для понимания технологии программирования в среде LabVIEW.

Графический язык программирования «G», используемый в LabVIEW, основан на архитектуре потоков данных. Последовательность выполнения операторов в таких языках определяется не порядком их следования (как в процедурных языках программирования), а наличием данных на входах этих операторов. Операторы, не связанные данными, выполняются параллельно в произвольном порядке.

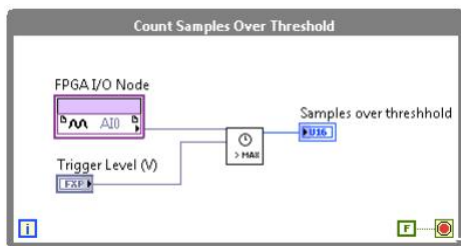
Программа LabVIEW (*.vi) называется и является виртуальным прибором (Virtual Instrument), и состоит из двух частей:

- блочной диаграммы, описывающей логику работы виртуального прибора;
- лицевой панели, описывающей внешний интерфейс виртуального прибора.

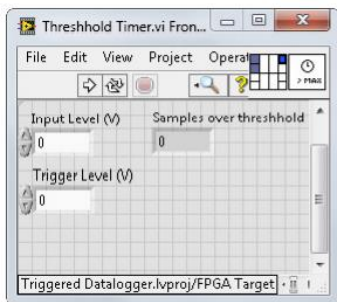
Виртуальные приборы могут использоваться в качестве составных частей для построения других виртуальных приборов.

Лицевая панель виртуального прибора содержит средства ввода-вывода: кнопки, переключатели, светодиоды, верньеры, шкалы, информационные табло и т. п. Они используются человеком для управления виртуальным прибором, а также другими виртуальными приборами для обмена данными.

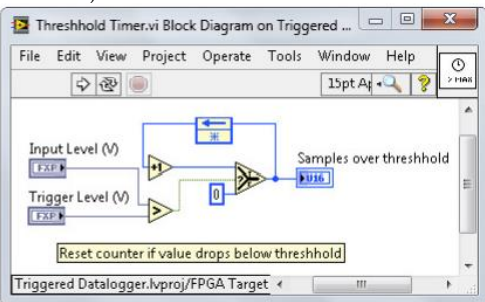
Блочная диаграмма содержит функциональные узлы, являющиеся источниками, приёмниками и средствами обработки данных. Также компонентами блочной диаграммы являются терминалы («контакты» объектов лицевой панели) и управляющие структуры, являющиеся аналогами таких элементов текстовых языков программирования, как условный оператор «IF», операторы цикла «FOR» и «WHILE» и т. п. Функциональные узлы и терминалы объединены в единую схему линиями связей, цвет которых определяется типом данных.



а)



б)



в)

Рис. 7.2. Пример программы LabVIEW

На рис. 7.2,а показан пример программы для определения количества выборок со значением больше заданного, поступающих из узла вывода

аналоговых данных FPGA I/O. Структура программы предполагает наличие подпрограммы (subVI) с именем «>MAX». Сама подпрограмма представлена лицевой панелью (рис.7.2,б) и блочной диаграммой (рис. 7.2,в), где используются функции сравнения, инкрементирования и выбора, а также узел обратной связи.

Контроллер реального времени

Контроллер реального времени, внешний вид которого изображен на рис. 7.3, содержит процессор, который надежно и обусловлено выполняет приложения LabVIEW Real-Time, и обеспечивает высокоскоростное управление, отслеживание выполнения, встроенную регистрацию данных и связь с периферийными устройствами. Дополнительные опции включают в себя резервные входы питания от 9 до 30 В постоянного тока, часы реального времени, аппаратные сторожевые таймеры, два порта Ethernet, до 2 ГБ памяти для хранения данных и встроенную поддержку USB, RS232 и RS485.

Операционная система реального времени (ОСРВ)

ОСРВ может выполнять программы с определенными требованиями ко времени, что важно для многих научных и инженерных проектов.



Рис. 7.3. Контроллер реального времени NI cRIO-9024

Во многих случаях неприемлемо выполнение программ измерения или управления на стандартном ПК с установленной ОС общего назначения, например, Windows. В любой момент ОС может задержать выполнение пользовательской программы по многим причинам: запуска проверки на вирусы, обновления, выполнения системных задач в фоновом режиме и т. д. Для программ, которые должны работать с определенной скоростью

без прерывания (например, система круиз контроля), эта задержка может вызвать сбой системы.

Обратите внимание, что такое поведение является конструктивным, так как ОС общего назначения оптимизированы для одновременного запуска множества процессов и приложений и предоставляют другие функции, такие как, например, сложная графика пользовательского интерфейса. Напротив, ОС реального времени предназначены для запуска одной программы с точным таймингом. В частности, ОС реального времени помогут реализовать следующее:

- Выполнение задачи в гарантированно лучшие сроки.
- Тщательное задание приоритетов для различных разделов программы.
- Выполнение циклов программы с почти одинаковым временем на каждой итерации (обычно в пределах микросекунд).
- Обнаружение, если цикл не достигает цели по времени.

Помимо точного тайминга, ОСПВ могут быть настроены для надежной работы в течение нескольких дней, месяцев или лет без остановки. Это важно не только для инженеров, создающих системы, требующие круглосуточной работы, но и для любого приложения, в котором простой обходится дорого. Функция «сторожевого пса» также обычно включается в систему реального времени для автоматического ее перезапуска, если программа пользователя перестает работать. Кроме того, оборудование, используемое в системе реального времени, часто спроектировано таким образом, чтобы выдерживать тяжелые условия в течение длительного времени.

Контроллеры реального времени NI работают на одной из трех различных ОСПВ: Phar Lap/ETS, VxWorks или NI Linux Real-Time. LabVIEW Real-Time обеспечивает достаточный уровень абстракции, который помогает программистам в значительной степени игнорировать детали используемой ОСПВ.

FPGA с реконфигурируемым вводом/выводом

Шасси реконфигурируемого ввода/вывода с FPGA, изображенного на рис. 7.4, является основой встраиваемой системной архитектуры CompactRIO. Шасси позволяет напрямую подключаться к вводу-выводу, обеспечивая высокопроизводительный доступ к схемам ввода-вывода каждого модуля, а также к их таймингу, запуску и синхронизации.



Рис. 7.4. Шасси с реконфигурируемой FPGA

Поскольку каждый модуль подключается непосредственно к FPGA, а не через шину, то по сравнению с другими архитектурами контроллеров почти не происходит задержка управления вводом-выводом для реакции системы. По умолчанию, FPGA автоматически взаимодействует с модулями ввода-вывода и обеспечивает процессу реального времени детерминированный ввод-вывод. Изначально FPGA позволяет программам на контроллере получать доступ к вводу-выводу с джиттером между циклами менее 500 нс.

Также можно напрямую запрограммировать FPGA для дальнейшей настройки системы. Из-за скорости FPGA эти шасси часто используются для создания контроллеров, которые включают высокоскоростной буферизованный ввод-вывод, быстрые циклы управления или настраиваемую фильтрацию сигналов. Например, используя FPGA, одно шасси может одновременно выполнять более 20 циклов управления аналогового пропорционально-интегрально-дифференцирующего (PID) регулятора с частотой 100 кГц.

Кроме того, поскольку весь код FPGA выполняется аппаратно, он обеспечивает высочайшую надежность и детерминизм, что идеально подходит для аппаратных блокировок, настраиваемой синхронизации и запуска, устранения нестандартных схем, которые обычно требуются для нестандартных датчиков и шин.

Модули ввода/вывода

Модули ввода/вывода обеспечивают изоляцию, схемы преобразования сигналов и встроенные средства для прямого подключения к промышленным датчикам или исполнительным механизмам. Предлагая различные варианты подключения и интегрируя модули через соединительные узлы шасси, система CompactRIO значительно снижает требования к пространству и затраты на электромонтаж. Для CompactRIO можно вы-

брать из более 70 модулей ввода-вывода серии С нужный для подключения практически к любому датчику или актуатору (рис.7.5).



Рис.7.5. Модули серии С (из более 70) для подключения к CompactRIO

Различные типы модулей обеспечивают: входы для термопар; выборку цифровых сигналов ± 10 В, 24-битные аналоговые входы/выходы; промышленные цифровые входы/выходы 24 В с током до 1 А; дифференциальные или цифровые входы с уровнем TTL; 24-битные входы пьезоэлектрического акселерометра IEPE; аналоговые выходы; измерения деформации; измерения терморезисторов RTD; измерения мощности; возможность подключения к сети контроллеров (CAN); использование для регистрации защищенных карт памяти (SD).

Кроме того, можно создавать свои собственные модули или приобретать модули у сторонних поставщиков. С помощью комплекта разработчика модулей cRIO-9951 NI CompactRIO можно разрабатывать собственные модули для удовлетворения конкретных потребностей приложений. Комплект обеспечивает доступ к низкоуровневой архитектуре встроенной системы CompactRIO с электрическими схемами для проектирования специализированных модулей ввода-вывода, связи и управления. Он включает библиотеки LabVIEW FPGA для взаимодействия со схемой разработанного модуля.

7.2. Конфигурации системы

Простейшая встроенная система состоит из одного контроллера, работающего в «безхостной» конфигурации. Эта конфигурация используется в приложениях, которым не требуется человеко-машинный интерфейс (HMI), за исключением целей обслуживания или диагностики. Однако для большинства приложений управления и мониторинга требуется, чтобы HMI отображал данные оператору или позволял оператору отправлять команды во встроенную систему. Обычная конфигурация - 1:1 или 1 хост к 1 CompactRIO, как показано на рис. 7.6. HMI связывается с оборудованием CompactRIO через прямое соединение Ethernet, концентратор или беспроводный маршрутизатор.

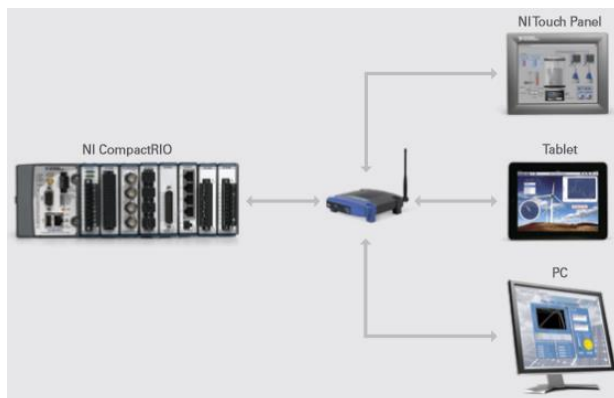


Рис. 7.6. Конфигурация 1:1 (от 1 хоста к 1 CompactRIO)

Следующий уровень возможностей и сложности системы - это конфигурация либо 1:N (от 1 хоста к N CompactRIO), либо N:1 (от N хостов к 1 CompactRIO). Хостом обычно является настольный ПК или промышленное устройство с сенсорной панелью. Конфигурация 1:N, показанная на рис. 7.7, типична для систем, управляемых одним оператором. Конфигурация N:1 является обычной, когда несколько операторов используют HMI для проверки состояния системы из разных мест.

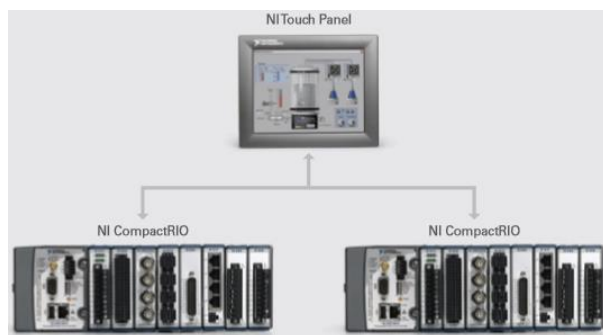


Рис. 7.7. Конфигурация 1:N (от 1 хоста к N CompactRIO), обычная для систем, управляемых одним оператором

Сложные приложения для управления машиной могут иметь конфигурацию N:N с множеством контроллеров и HMI (рис 7.8). Они часто включают в себя высокопроизводительный сервер, который действует как средство регистрации и пересылки данных. Эта конфигурация системы подходит для физически больших или сложных электрофизических уста-

новок. Используя ее, можно взаимодействовать с установкой из разных мест или распределять определенные обязанности по мониторингу и управлению между группой операторов.

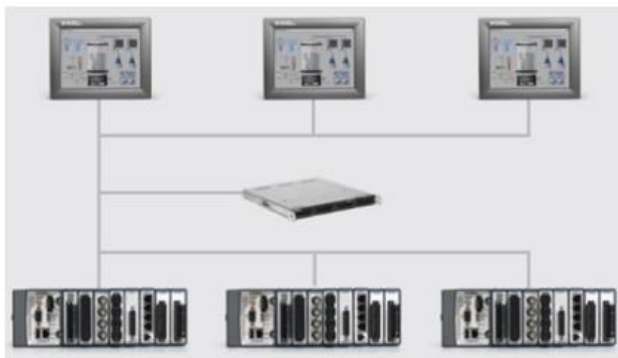


Рис. 7.8. Распределенная система управления

7.3. Базовая архитектура CompactRIO

Архитектура CompactRIO должна обеспечивать общий обзор системы. Базовая схема включает процессы и пути передачи данных. Процесс представляет собой независимый, асинхронно выполняющийся сегмент кода - в основном, цикл.

Процессы определяются задачами, которые выполняются приложением. Общие задачи включают ПИД-регулирование, регистрацию данных, связь с человеко-машинным интерфейсом, связь с вводом-выводом, а также программы безопасности или обработки ошибок.

Обмен данными - один из наиболее важных факторов, которые следует учитывать при проектировании встроенной системы. LabVIEW, LabVIEW Real-Time и LabVIEW FPGA включают множество различных механизмов для передачи данных между процессами на одном объекте, а также процессы, которые обмениваются данными между объектами.

Выбор механизма передачи данных зависит от нескольких факторов, главным из которых является модель передачи данных. Большинство приложений CompactRIO могут содержать как минимум две или более моделей передачи данных, перечисленных ниже:

- Сообщение/команда
- Текущее значение (tag)
- Поток

Сообщение/команда

Связь на основе команд или сообщений происходит относительно нечасто и запускается каким-то определенным событием. Например, пользователь нажимает кнопку на панели, обозначенной в HMI, чтобы запустить вакуумный насос. При обмене на основе сообщений необходимо гарантировать своевременную передачу сообщения с малой задержкой.

Текущее значение (тег)

Теги используются для периодической передачи текущих значений между процессами или от контроллера к HMI. Теги обычно используются для представления переменных процесса, таких как заданное значение для системы управления или текущие данные температуры. Часто теги привязаны к каналам ввода-вывода в системе с большим количеством каналов. Используя теги текущего значения, каждая передача данных не требует гарантии, потому что контроллер или HMI всегда интересуются последним значением, а не предыдущими значениями.

Поток

Потоковая передача данных буферизируется и включает высокопроизводительную передачу данных, где пропускная способность важнее задержки. Обычно один процесс записывает данные в другой процесс, который читает, отображает или обрабатывает эти данные. Примером может служить бортовой регистратор данных, который отправляет данные от FPGA в RTOS для регистрации данных. Потоковая передача часто требует многоуровневой буферизации, во время которой каждый конец (отправитель и получатель) имеет свой собственный буфер уникального размера. Это идеально, когда у вас есть цикл процессора, который может читать большие порции данных за раз, и цикл сбора данных, который записывает небольшие порции данных с более короткими интервалами.

Сводка различных методов передачи данных и их функций приведена в табл. 7.1.

Табл. 7.1. Сводка методов передачи данных

Тип связи	Основные функции	Дополнительные функции	Производительность
Сообщение/команда	Буферизация, блокировка (тайм-аут)	Подтверждение передачи данных	Малая задержка
Текущее значение (тег)	Только текущее значение, периодическое чтение	Динамический поиск, управление группами	Малая задержка, большое количество каналов
Поток	Буферизация, блокировка (тайм-аут)	Многослойная буферизация	Высокая пропускная способность

7.4. Режимы программирования CompactRIO

Архитектуры CompactRIO предоставляют возможность реализовать ввод-вывод путем настройки FPGA с помощью LabVIEW FPGA или с использованием режима сканирования NI CompactRIO. Если есть LabVIEW Real-Time и LabVIEW FPGA на компьютере разработчика, необходимо выбрать, какой режим программирования будет использован, когда добавляется целевой объект CompactRIO в разрабатываемый проект LabVIEW. Выбрав режим, можно перейти в гибридный режим, если предполагается использовать для разрабатываемого приложения гибридный LabVIEW FPGA и CompactRIO Scan Mode (режим сканирования).

Режим интерфейса LabVIEW FPGA

В режиме интерфейса LabVIEW FPGA раскрывается реальная мощь CompactRIO, настроив индивидуальные особенности FPGA в дополнение к программированию процессора реального времени. Это поможет достичь производительности, для которой обычно требуется специальное оборудование. Используя LabVIEW FPGA, можно реализовать настраиваемую синхронизацию и запуск, разгрузить процессор от обработки и анализа сигналов, создать собственные протоколы обмена данными и получить доступ к вводу/выводу с максимальной скоростью.

При обмене данными между VI FPGA и VI реального времени имеется возможность передавать данные с очень высокой скоростью с помощью DMA FIFO или передавать одноканальные данные с помощью элементов управления и индикаторов.

Режим сканирования CompactRIO

Используя режим сканирования CompactRIO, программируется процессор реального времени системы CompactRIO, но не FPGA. В этом режиме NI предоставляет сконфигурированный FPGA, который периодически сканирует ввод/вывод и помещает данные в карту памяти, делая ее доступной для LabVIEW Real-Time. Режим сканирования CompactRIO достаточен для приложений, которым требуется одноточечный доступ к вводу-выводу с частотой в несколько сотен герц. Он не поддерживает потоковую передачу данных с высокой скоростью.

Гибридный режим

Одновременное использование режима сканирования CompactRIO и LabVIEW FPGA на целевом объекте известно как гибридный режим. При таком подходе модули, которые выбираются для программирования непосредственно с помощью LabVIEW FPGA, удаляются из сканирования ввода-вывода, а оставшиеся модули связываются с интерфейсом сканирования RIO. При этом интерфейс сканирования RIO использует два из трех каналов прямого доступа к памяти (DMA), которые обычно доступны для использования в LabVIEW FPGA.

Использование режима LabVIEW FPGA

Как и системы управления на базе процессоров, FPGA использовались для реализации разных типов промышленных систем управления, включая аналоговое управление процессами, дискретную логику, а также системы управления на основе пакетной обработки или конечного автомата. Однако системы управления на основе FPGA существенно отличаются от систем на базе процессоров. Если разрабатываемое приложение имеет какие-либо требования, перечисленные ниже, то следует запрограммировать ввод-вывод и другие низкоуровневые задачи с помощью LabVIEW FPGA (рис.7.9).

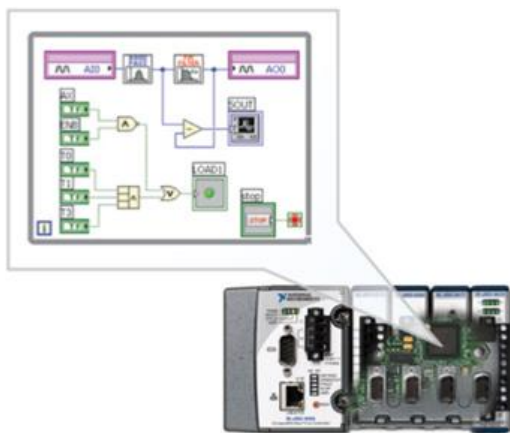


Рис.7.9. С помощью LabVIEW FPGA создаются пользовательские VIs FPGA для высокоскоростного сбора данных, циклов управления или настраиваемого тайминга и запуска.

Максимальная производительность и надежность

Когда компилируется управляющее приложение для устройства с FPGA, результатом является высоко оптимизированная твердотельная реализация, которая обеспечивает параллельную обработку с преимуществами производительности и надежности выделенных аппаратных схем. Поскольку на микросхеме FPGA нет ОС, код реализован таким образом, чтобы обеспечить максимальную производительность и надежность.

Высокоскоростной сбор или генерация сигналов (> 500 Гц)

Интерфейс сканирования RIO оптимизирован для циклов управления, работающих с частотой менее 500 Гц, но многие модули ввода-вывода серии С способны получать и генерировать данные с гораздо более высокой скоростью. Если необходимо в полной мере использовать преимущества этих функций модуля и получать или генерировать данные на скоростях выше 500 Гц, можно использовать LabVIEW FPGA для сбора данных с заданной пользователем скоростью, адаптированной к разрабатываемому приложению.

Пользовательский запуск, тайминг или синхронизация

С помощью реконфигурируемой FPGA можно создавать простые, расширенные или иным образом настраиваемые реализации запусков, схем тайминга и синхронизации ввода-вывода. Они могут быть сложными, как запуск пользовательского сообщения с использованием интер-

фейсной системы Controller Area Network (CAN) для сбора аналоговых данных, превышающих пороговое значение, или простыми, как получение входных значений по нарастающему фронту внешнего источника синхронизации.

Аппаратный анализ/генерация и совместная обработка

Многие датчики выдают больше данных, чем может быть обработано одним процессором реального времени. Тогда можно использовать FPGA как сопроцессор для анализа или генерации сложных сигналов, освобождая процессор для других критических потоков. Этот тип сопроцессора на основе FPGA обычно реализуется в приложениях, которые используют:

- Датчики кодирования/декодирования
 - тахометры
 - стандартные и/или настраиваемые цифровые протоколы
- Обработку и анализ сигналов
 - спектральный анализ (быстрое преобразование Фурье и обработка окон)
 - фильтрация, усреднение и т. д.
 - сокращение данных
- Имитацию датчика
 - кулачок и кривошип
 - линейно-регулируемые дифференциальные трансформаторы (LVDT)

Управление с наивысшей производительностью

Можно использовать FPGA не только для высокоскоростного сбора и генерации данных, но также можно реализовать множество алгоритмов управления на FPGA. Можно использовать одноточечный ввод-вывод с многоканальным настраиваемым ПИД-регулятором или другими алгоритмами управления для реализации детерминированного управления со скоростью цикла выше 1 МГц. Например, алгоритм ПИД-регулирования, входящий в состав модуля LabVIEW FPGA, выполняется всего за 300 нс.

Использование режима сканирования CompactRIO

Когда используется контроллер CompactRIO в проекте LabVIEW, можно выбрать интерфейс сканирования в качестве режима программирования. Интерфейс сканирования RIO автоматически обнаруживает установленные в шасси модули ввода/вывода и добавляет их в проект LabVIEW. Затем можно разметить переменные ввода-вывода на разрабатываемые блок-схемы LabVIEW Real-Time и хост VI и сразу же читать и

записывать масштабированные, откалиброванные данные ввода-вывода без какого-либо программирования или конфигурирования FPGA.

7.5. Технология FPGA

СБИС FPGA предлагает параллельную и настраиваемую платформу, которую можно использовать для выполнения сложных задач обработки и управления на аппаратной скорости. FPGA - это программируемый чип, состоящий из трех основных компонентов: логических блоков, программируемых межсоединений и блоков ввода/вывода. На рис. 7.10 показана простая структура FPGA фирмы Xilinx.

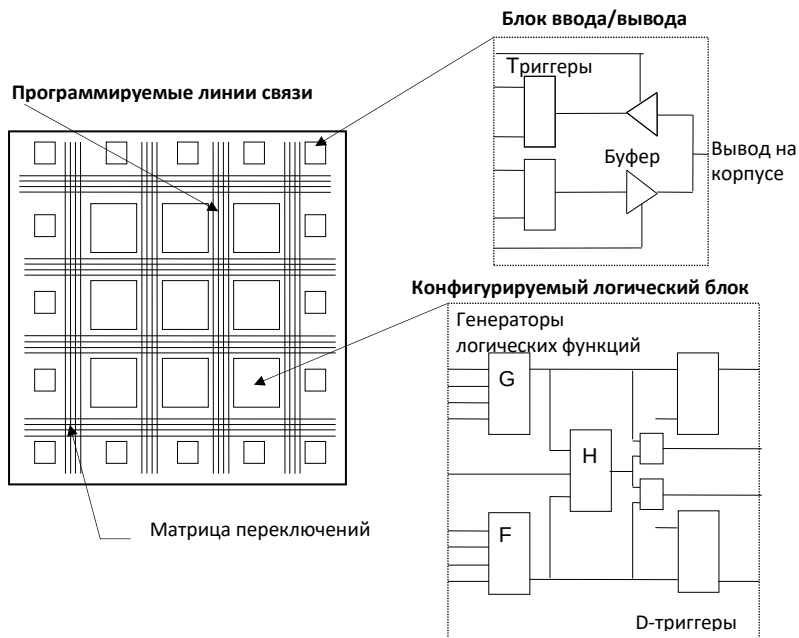


Рис.7.10. Структура FPGA XC4000

Логические блоки более сложных FPGA представляют собой набор цифровых компонентов, таких как таблицы кодировки данных, умножители и мультиплексоры, где цифровые значения и сигналы обрабатываются для генерации нужного логического выхода. Эти логические блоки соединены программируемыми связями, которые направляют сигналы от одного логического блока к другому. Также программируемые связи мо-

гут направлять сигналы к блокам ввода/вывода для двусторонней связи с окружающими схемами.

Контроллер cRIO-9045 имеет на борту FPGA типа Xilinx Kintex-7 7K70T с характеристиками, указанными в табл. 7.2.

Табл. 7.2. Основные характеристики FPGA типа Xilinx Kintex-7 7K70T

Количество триггеров	82000
Количество LUT (Look-Up Table) ¹ с 6 входами	41000
Количество слоев DSP ² (Множители 18 × 25)	240
Доступная блочная RAM	4860 кбит
Количество каналов DMA ³	16
Количество логических прерываний	32

¹ – таблица перекодировки данных

² – процессор цифровой обработки сигналов

³ – прямой доступ к памяти

FPGA работают на относительно более низкой частоте, чем CPU, но они компенсируют разницу в тактовой частоте за счет создания специализированных схем, которые могут выполнять несколько операций в течение одного тактового периода. Объединив это с их тесной интеграцией ввода-вывода на устройствах NI с реконфигурируемым вводом-выводом (RIO), можно достичь гораздо более высокой пропускной способности и детерминированности, а также более быстрого времени отклика, чем с использованием только процессора. Это поможет справиться с высокоскоростной потоковой передачей, цифровой обработкой сигналов (DSP), управлением и приложениями цифрового протокола.

Поскольку программы VIs LabVIEW FPGA синтезируются до аппаратной реализации, процесс компиляции FPGA отличается от процесса компиляции для традиционного приложения LabVIEW для Windows или LabVIEW Real-Time (см. рис.7.11). При написании кода для FPGA, создается тот же код LabVIEW, что и для любой другой цели, но после нажатия кнопки «Выполнить», LabVIEW внутренне выполняет другой процесс. Сначала LabVIEW FPGA генерирует код на языке описания аппаратуры интегральных схем VHDL (Very high speed integrated circuits **H**ardware **D**escription Language) и передает его компилятору Xilinx. Затем, компилятор Xilinx синтезирует объектный файл VHDL, помещая туда необходимые компоненты, и направляет все в битовый файл. Наконец, битовый файл загружается в FPGA, и FPGA получает запрограммированную функциональность. Этот процесс, который является более сложным, чем другие компиляции LabVIEW, может, в зависимости от сложности проекта, занять несколько часов.

