СИНХРОННО-ВРЕМЕННОЙ ПРОТОКОЛ ИНФОРМАЦИОННОГО ОБМЕНА

Захаров Н.А., Клепиков В.И., Подхватилин Д.С.
НПП «Дозор» ОАО «Концерн КЭМЗ»
nazakharov@npp-dozor.ru, dspodkhvatilin@npp-dozor.ru, viklepikov@mail.ru

Аннотация. Предложен синхронно-временной протокол для построения распределенных систем управления. Показана необходимость использования в контурах регулирования передачи данных по расписанию. Рассмотрены преимущества предлагаемого протокола перед аналогами.

Ключевые слова: распределенная система управления, протокол обмена, синхронно-временной протокол.

Введение

Развитие распределенных систем управления (РСУ) на современном этапе характеризуется повышением функциональной насыщенности сетевых связей. Обмен данными в реальном масштабе времени ведет к многократному увеличению трафика в РСУ, причем возрастают потоки информации между интеллектуальными датчиками, управляющими контроллерами и исполнительными элементами.

Как отмечено в монографии [1], необходимость перехода к распределенным системам управления обусловлена требованиями повышения функциональности и качества управления при одновременном повышении надежности и улучшении массогабаритных характеристик. Рост количества измеряемых параметров ведет к увеличению количества кабелей и, соответственно, количества и размеров разъемов на корпусе управляющего устройства. При этом миниатюризация электронных устройств, в том числе за счет прогресса в области микроэлектроники, приводит к тому, что габаритные размеры центрального блока управления определяются не входящими в его состав электронными блоками, а его разъемами для внешних подключений. Неисправности разъемов и соединительных кабелей, в свою очередь, являются одним из значимых источников отказов системы управления в целом.

Современный подход к изготовлению узлов и агрегатов систем управления и технологических объектов управления – датчиков, ИМ, законченных функциональных узлов, заключается в том, что они выпускаются различными компаниями-изготовителями, имеют встроенные микропроцессорные блоки с стандартизованным сетевым интерфейсом. Это обеспечивает высокую степень унификации, снижает стоимость разработки и сокращает время вывода на рынок новых продуктов.

Вследствие интеграции в компоненты РСУ микропроцессорных устройств со встроенным сетевым интерфейсом в составе одного контура регулирования оказывается несколько микропроцессорных устройств, связанных между собой цифровыми линиями связи. Это порождает особую специфику требований, предъявляемых к сетевому обмену и особенностям его реализации.

Рассмотрим получивший широкое распространение в РСУ протокол CAN, первоначально разработанный фирмой Robert Bosch GmbH для использования в автомобильной электронике [2-4]. Данный протокол обладает высокой помехоустойчивостью и надежностью. При длине линии до 60 м скорость обмена составляет 1 Мбит/с. Топология сети – шина. На физическом уровне шина CAN представляет собой витую пару волновым сопротивлением 120 Ом±10 %. Возможна реализация протокола CAN и в иных физических средах, в частности, в оптоволокне. Стандартом [3] предписаны следующие свойства шины CAN:

- доступ к шине по принципу мультимастера на основе приоритета;
- недеструктивный арбитраж по содержимому;
- групповая передача кадров с полосовой фильтрацией;
- удаленный запрос данных;
- универсальность конфигурации;
- целостность данных в пределах всей системы;
- обнаружение ошибок и сигнализация об ошибках;
- автоматическая повторная передача кадров, пропущенных при арбитраже или нарушенных ошибками при передаче;
- различение кратковременных ошибок и постоянных отказов узлов, самостоятельное отключение дефективных узлов.

Этим же стандартом предписаны следующие меры для обнаружения ошибок:

– мониторинг (передатчики сравнивают уровни битов, которые должны передаваться, с уровнями битов, присутствующими на шине);
– контроль при помощи 5-разрядного циклического избыточного кода;
– переменное заполнение битами с шириной заполнения, равной 5;
– проверка кадра;
– проверка подтверждения приема.

Обмен по шине CAN организован следующим образом. Любой узел сети может в любой момент начать передачу, если линия свободна. Передача может быть инициирована каким-либо событием, наступившем в узле, например, превышением измеряемой величиной порогового значения, поступлением дискретного сигнала, срабатыванием локального таймера и т.п. Поэтому протокол CAN относят к событийным протоколам. В случае попытки одновременной передачи двумя и более узлами на шине возникает коллизия. Для разрешения коллизий на шине предусмотрен механизм арбитража.

Механизм арбитража, следующий: В описании протокола CAN применительно к значениям битов используются термины «доминантный» и «рецессивный». При одновременной передаче доминантного и рецессивного бита на шине установится и будет прочитано всеми абонентами значение доминантного бита. Применительно к витой паре в качестве физической среды доминантным является логический ноль. В составе каждого фрейма за областью начала фрейма следует 11-битовая область арбитража. Во время передачи области арбитража передатчик сравнивает текущий уровень на шине с уровнем (0 или 1), который он должен передать. Если узел, передавший рецессивный уровень (1), обнаружит на шине доминантный уровень (0), он должен прекратить передачу и освободить шину для другого выполняющего передачу данных узла. Таким образом, арбитраж выигрывает узел, передающий сообщение с меньшим значением в поле арбитража.

В ходе планирования сетевого обмена по шине CAN приоритет задается не абоненту сети, а каждому сообщению отдельно. Один и тот же абонент может передавать сообщения с разными приоритетами. Это позволяет эффективно организовать передачу критически важной информации, например, сообщений об аварийной ситуации. Недостатком такой организации обмена является то, что в условиях сильной загрузки шины сообщение с низким приоритетом может быть не передано никогда или передано с неприемлемой задержкой. Еще одним недостатком протокола CAN является значительный объем передаваемой служебной информации, что снижает скорость обмена собственно данными.

Существенным недостатком протокола CAN, проявляющимся в случае, когда несколько абонентов сети участвуют в работе одного контура регулирования, является то, что асинхронная природа протокола и применение рассмотренного выше механизма арбитража делают нестабильным время передачи результата измерения от датчика и рассчитанного управляющего воздействия к исполнительному элементу. Это приводит к ухудшению качества регулирования, в том числе к снижению запаса устойчивости системы. Нестабильность времени передачи данных от цикла к циклу наглядно показана на рис. 1.

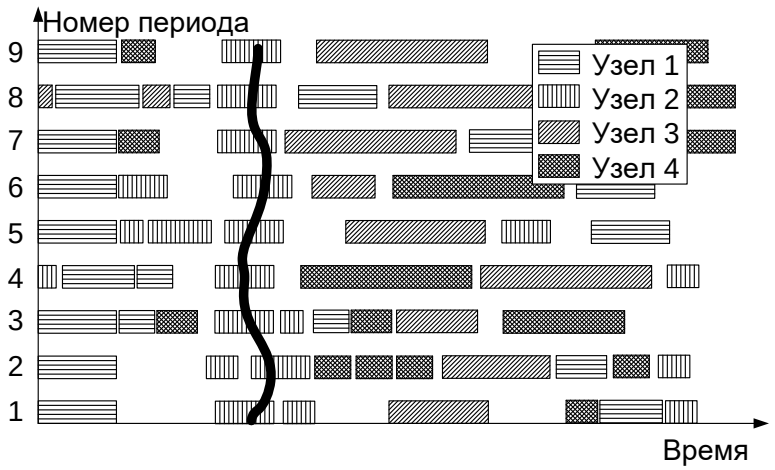

Рис. 1. – Пример передачи данных в сети с событийным протоколом обмена

Стандартом [3] в разделе 9.2 предусмотрена опция разделения времени передачи на шине. Тем не менее, она не всегда эффективна. В статье [5] описано построение РСУ с обменом по CAN, использующей данную опцию. Для компенсации указанной нестабильности используются громоздкие программно-аппаратные решения, основанные на формировании и передаче в каждом пакете данных двух дополнительных временных меток.

Для РСУ, в которых датчики, исполнительные элементы и контроллеры являются узлами единой сетевой управляющей структуры, принципиальным является то обстоятельство, что коммуникационный канал между датчиками, контроллерами и исполнительными элементами (в отличие от децентрализованных систем) оказывается включенным в замкнутые контуры управления и регулирования; отказы, сбои и непредсказуемые задержки доставки информации (джиттер) являются недопустимыми. Это обстоятельство в 2000-х годах привело к началу активных разработок специализированных коммуникационных протоколов, пригодных для построения систем управления жесткого реального времени, например, CANAerospace и Time Triggered Protocol (TTP) для авиационных СУ, протокол FlexRay для реализации концепции "управления по проводам" (x-by-wire) в автомобильной технике и др. Особенности построения РСУ, работающих в жестком реальном времени, рассмотрены в статье [6].

Для построения РСУ предлагается синхронно-временной протокол (СВП), свободный от рассмотренных выше на примере CAN недостатков событийного протокола [7]. В ходе обмена по указанному протоколу каждый узел РСУ осуществляет передачу данных строго в определенное для него время. Время передачи задается единым для всей сети расписанием. Топология сети – дублированная шина. На физическом уровне используется витая пара номинальным волновым сопротивлением 120 Ом±10 %. Драйверы шины соответствуют спецификациям интерфейса RS-485. Контроль целостности и достоверности передачи данных обеспечивается посредством 32-битовой контрольной суммы.

Основной структурной единицей сети является кластер. Под кластером понимается комплекс объединенных общей шиной узлов сети, обмен данными, между которыми осуществляется по единому общему для всех узлов расписанию. Каждый узел кластера (абонент сети) хранит у себя копию общего расписания. Отрезки времени, составляющие расписание, называются слотами. В каждом слоте передавать сообщение разрешается только одному узлу, прочие узлы в это время находятся в режиме приема. Выход при передаче за предустановленные временные границы запрещен. Предусмотрена возможность наличия нескольких расписаний в одном кластере. Протокол поддерживает переключение между расписаниями.

Временная диаграмма обмена по СВП протоколу показана на рис. 2. Обмен данными в кластере организован в виде циклов фиксированной длительности. В течение цикла происходит повторяющийся обмен полным набором сообщений. Цикл кластера разделен на слоты. Каждому узлу кластера выделен один или несколько слотов, в которых он должен в каждом цикле выполнять передачу пакетов. При этом время доставки каждого сообщения строго соблюдается.

В настоящее время на рынке представлено несколько типов коммуникационных протоколов как с событийной синхронизацией (Event Triggered, ET, например, рассмотренный выше CAN), так и с временной синхронизацией (Time Triggered, TT). Событийные протоколы обеспечивают бóльшую гибкость системы при разработке и модернизации (механизмы Plug&Play). При этом их использование в замкнутых контурах регулирования в силу отмеченных выше причин практически невозможно. Протоколы с временной синхронизацией обеспечивают бóльшую жесткость и предсказуемость поведения каждого узла сети, особенно при возникновении сбоев, отказов и нештатных ситуаций, обеспечивают фиксированный временной джиттер, что позволяет строить замкнутые распределенные контуры регулирования. Применение таких протоколов осложняет разработку, модернизацию и расширение РСУ, поскольку каждый раз требуется проектирование нового расписания работы узлов. Несмотря на очевидные преимущества временных протоколов, событийные протоколы, такие как CAN и Ethernet имеют широкое распространение благодаря легкости интеграции и модернизации СУ.

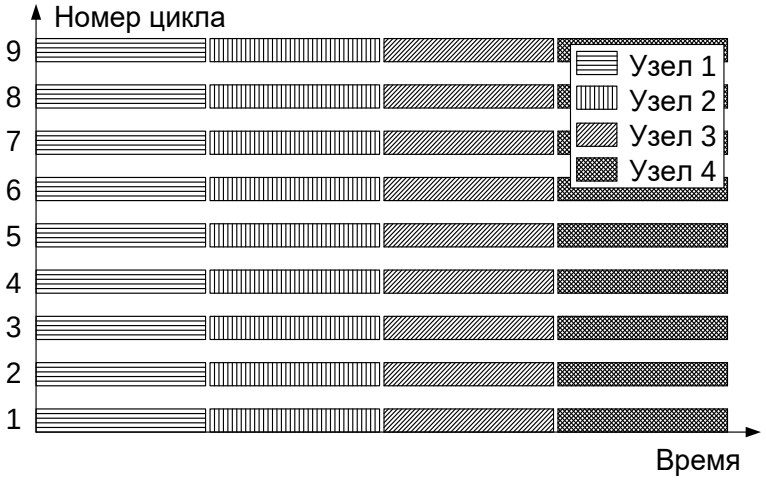

Рис. 2 - Пример передачи данных в кластере сети СВП

С существующими протоколами с временной синхронизацией связан ряд проблем:

- отсутствие механизма P&P – добавление нового узла сети требует перепланировки расписания работы всех узлов;
- жесткий детерминизм в построении расписания приводит к недетерминизму в работе замкнутых контуров регулирования т.к. при реконфигурациях сети вследствие отказов отдельных узлов изменяются времена доставки информации;
- используемые механизмы синхронизации узлов сети и механизмы включения узла в работающую сеть приводят к проблеме распада сети на независимо друг от друга работающие сегменты – клики (clique);
- в известных реализациях отсутствует механизм гарантированной синхронной выдачи выходных сигналов несколькими узлами;
- имеющийся европейский стандарт на TTP жестко определяет применение дублированной (двухканальной) шины и не предусматривает механизмов повышений степени резервирования шины;
- доступные IP-ядра TTP контроллеров дороги, доступные микросхемы TTP контроллеров являются двухканальными, причем оба канала реализованы в одном кристалле, что снижает надежность;
- в России отсутствуют стандарты и специализированные лаборатории по тестированию и сертификации систем на основе временных протоколов.

Для решения обозначенных проблем в предлагаемом СВП протоколе разработан ряд новых механизмов обеспечения отказоустойчивости, целостности и реконфигурации сети:

- механизм принудительной синхронизации сети;
- разрешительный механизм вхождения в сеть нового или повторно входящего узла;
- механизм онлайн планирования и рассылки расписаний;
- механизм предопределенных резервных расписаний;
- механизм распределенной базы данных контрольных точек восстановления узлов сети;
- механизм гарантированной синхронной выдачи сигналов несколькими узлами;
- структура контроллера с заданным количеством резервных шин.

В отличие от известного механизма автономной синхронизации, когда каждый узел самостоятельно входит в сеть в соответствии с заранее определенным расписанием, в предлагаемом СВП механизм принудительной синхронизации опрашивает сеть с заданным периодом времени для определения фактического состава присутствующих узлов. Собранная информация рассылается всем работающим узлам, соответственно у всех работающих узлов имеется единая информация о составе сети, что полностью исключает проблему сегментации сети.

Если в сети появляется новый, неизвестный ранее узел (чего в принципе недопустимо в известных реализациях), то в СВП в процессе опроса сети узлом-синхронизатором данный узел должен сообщить свой формуляр – приоритет, требуемый набор входных данных и требуемый период их получения, состав своих выходных данных и другую служебную информацию. Узел-синхронизатор планирует новое расписание работы сети, в котором определяет место нового узла.

Новое расписание рассылается всем узлам сети, после этого новый узел получает разрешение войти в сеть. Данная процедура обеспечивает реализацию механизма P&P.

Если в процессе работы сети происходит отказ одного из узлов, то его выходные данные должны замещаться данными резервного узла. При жестком офлайн планировании расписания (как это сделано в известных реализациях) данные резервного узла передаются в отведенном ему интервале времени (слоте), который не совпадает со слотом отказавшего узла. Это нарушает настройки контуров регулирования и может приводить к снижению качества динамических процессов. В СВП механизмы планирования и рассылки расписаний обеспечивают замену расписания работы сети на нужное предопределенное резервное расписание, в котором резервный узел передает свои данные в том же слоте, в котором работал отказавший узел. Отказавший узел исключается из состава активных узлов, но если происходит его восстановление, то он включается в сеть в соответствии с механизмом повторного вхождения. Распределенная база данных контрольных точек обеспечивает автоматическое вхождение узла в режим работы, соответствующий текущему состоянию объекта управления.

В СВП введен механизм гарантированной синхронной выдачи сигналов несколькими узлами, который необходим в тех случаях, когда выходные сигналы нескольких узлов управляют кинематически или гидравлически связанными исполнительными элементами (например, подвижной платформой тренажера). Механизм гарантирует либо скоординированную выдачу сигналов всеми узлами сети, либо блокировку выдачи сигналов всеми узлами при отказе одного из них.

В СВП в отличие от известных реализаций предусмотрена возможность реализации различных степеней резервирования шины.

На базе разработанных спецификаций совместно с ФГУП ГосНИИАС начата работа по выпуску стандарта СВП. Создание сертифицирующей лаборатории планируется на базе Инновационного центра Сколково.

Согласно предлагаемой технологии, обобщенная модель РСУ содержит несколько сетевых узлов, связанных с объектом управления датчиками и исполнительными устройствами и связанных между собой последовательным коммуникационным каналом на основе СВП протокола.

Каждый узел состоит из:

- контроллера ввода-вывода, управляющего датчиками и исполнительными устройствами;
- управляющего контроллера, выполняющего прикладное программное обеспечение;
- коммуникационного контроллера, обеспечивающего логическое взаимодействие с коммуникационным каналом;
- блока защиты шины (БЗШ), связывающего коммуникационный контроллер с коммуникационным каналом.

Блок защиты шины (БЗШ) является автономной подсистемой и предотвращает отказы узла типа «забивание» шины (babbling idiot faults). В терминах дискретной логики БЗШ можно рассматривать как логическое «И», т.е. сообщение может быть отправлено, только если коммуникационный контроллер и БЗШ согласованно обеспечивают доступ к каналу.

В СВП обмен данными организован в виде повторяющихся раундов. Раунд разделен на слоты. Каждый узел в коммуникационной системе имеет свой слот и должен в каждом раунде выполнять в данном слоте передачу своих пакетов. Длина пакета задается для каждого узла и может варьироваться в пределах от 2 до 240 байтов, в пакете может содержаться несколько сообщений. Последовательность раундов циклически повторяется, образуя цикл кластера.

В пределах цикла кластера каждый узел в своем раунде может передавать различные прикладные сообщения.

Данные защищаются 32 битовым CRC (Cyclic Redundancy Check) кодом, который для пакетов размером до 34 байт обеспечивает кодовое расстояние Хемминга HD = 8, для пакетов до 4092 байт – HD = 6.

Все узлы сети работают в единой временной базе, для чего каждый коммуникационный контроллер и каждый БЗШ содержат отказоустойчивый усредняющий алгоритм коррекции локальных часов с тем, чтобы они находились в синхронизации с часами всех остальных узлов кластера.

Наборы расписаний работы сети рассчитывается и хранится в двух или более коммуникационных контроллерах узлов-синхронизаторов вместе с копиями базы данных контрольных точек.

Контрольные точки (т. е. наборы переменных процесса) позволяют новому узлу вступить в работу с того места, где процесс был прерван сбоем или отказом.

СВП обеспечивает подключение стартующего узла к работающему кластеру без нарушения функционирования остальных узлов. После старта узел слушает шину и ожидает получения стартового пакета от узла-конфигуратора. В стартовом пакете узел получает информацию о расписании сети и о своем месте в расписании для сообщения узлам-синхронизаторам своих требований по периодам обмена с другими узлами. После этого узлы-синхронизаторы корректируют расписание сети с учетом требований нового узла и сообщают это расписание всем узлам. Данный алгоритм обеспечивает реализацию механизма Plug&Play.

Механизм функциональной синхронизации обеспечивает синхронное выполнение требуемых функций разными узлами сети. Суть работы механизма заключается в том, что на уровне коммуникационного контроллера отслеживается прохождение по шине заданного набора сообщений, и если нужные сообщения были отправлены и приняты всеми узлами, то напрямую, минуя управляющий контроллер узла, коммуникационный контроллер выдает команды на устройства ввода-вывода.

Как отмечено выше, все узлы работают по единому расписанию обмена. Для обеспечения всех узлов единой временной базой требуется синхронизация часов. Каждый узел на основе априорно известного ожидаемого времени прихода корректного сообщения и фактического времени его прихода вычисляет разницу хода часов передатчика и приемника. Отказоустойчивый усредняющий алгоритм вычисляет коррекцию локальных часов с тем, чтобы они находились в синхронизации со всеми остальными часами кластера. Распределенный алгоритм контроля целостности кластера в случае возникновения отказа выявляет место его возникновения – выходная цепь передатчика или входная цепь приемника.

Базовые алгоритмы СВП были сформировано верифицированы и успешно протестированы в условиях имитации миллионов отказов, в том числе при воздействии радиационного и электромагнитного излучений.

В СВП реализована концепция парирования одиночных сбоев и отказов, заключающаяся в том, что системы на его основе содержат достаточную избыточность, чтобы одиночный сбой или отказ никаким образом не отразились на поведении системы: ни с точки зрения функциональности, ни во временных соотношениях. Данная концепция основана на том, что вероятность одновременного проявления отказов в двух различных компонентах пренебрежимо мала. При появлении множественных отказов, которые не могут быть парированы самим протоколом, системное программное обеспечение СВП информирует об этом прикладную программу, которая, в свою очередь, может принять решение о прекращении своей работы или о переходе в безопасный режим.

Реализация перечисленных механизмов отказоустойчивости обеспечивается реализованной в СВП поддержке согласованности (consistency) данных. В однопроцессорной системе согласованность гарантируется благодаря возможности всем компонентам ПО пользоваться одной копией данных, хранящихся в ОЗУ.

Такой вид согласованности данных не работает в РСУ по следующим причинам. Во-первыз, из-за задержек при передаче нет гарантии, что переданное сообщение будет принято всеми узлами-приемниками в одно и тоже время. Во-вторых, некоторые узлы могут находиться в нерабочем состоянии, или сообщение из-за сбоя в коммуникационной системе может быть потеряно. Поддержка согласованности данных в СВП обеспечивается на уровне коммуникационного контроля целостности кластера (Membership) и подтверждений (Acknowledgment). Контроль целостности кластера заключается в следующем. Благодаря циклической (round-robin) схеме раундов обмена по расписанию каждый узел ожидает и проверяет список членов кластера для всех узлов данного раунда. Каждый передатчик, не соответствующий списку членов, определяется как неисправный. Это обеспечивает согласованное взаимодействие узлов, каждый из которых видит других в своих списках членов кластера.

Для построения узлов СВП сети в настоящее время разработана и выпущена микросхема контроллера СВП. Разработчик – научно-производственное подразделение «Дозор» ОАО «Концерн КЭМЗ» (г. Москва), изготовитель - АО "ПКК Миландр" (г. Зеленоград). В перспективе планируется интеграция данного контроллера в систему на кристалле. Упрощенная структурная схема контроллера СВП приведена на рис. 3. Контроллер поддерживает связь с процессором по 16-разрядной параллельной шине данных или по интерфейсу SPI.

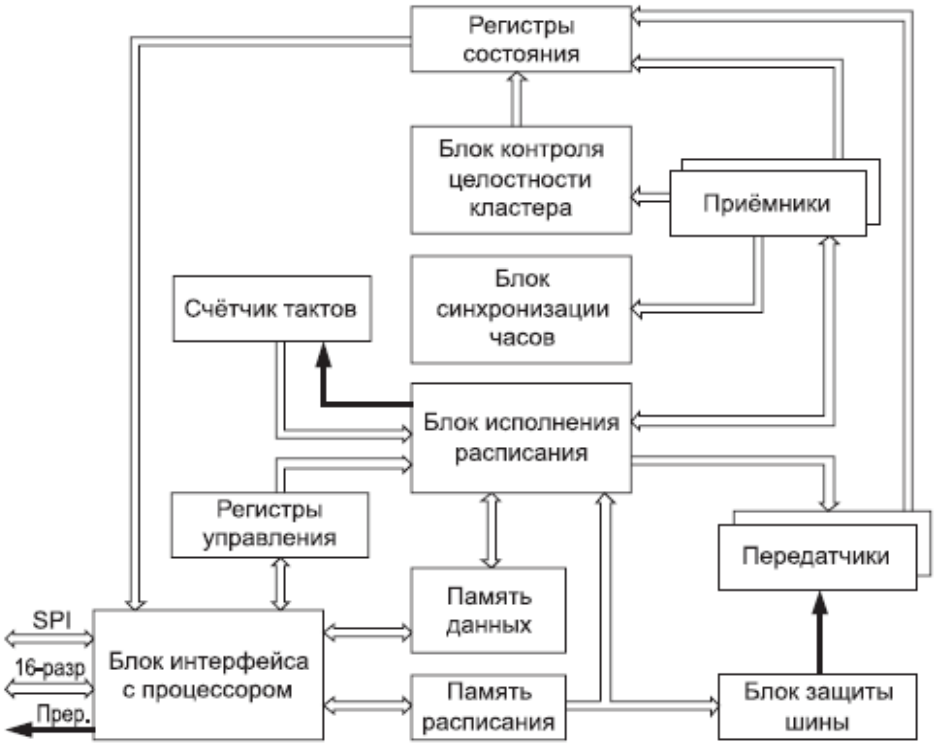

Рис. 3 – Структурная схема контроллера СВП

Расписание загружается в контроллер СВП программным обеспечением процессора и сохраняется в памяти. После старта контроллера СВП эта память становится доступной только для чтения.

После старта контроллер СВП работает асинхронно по отношению к процессору, получая синхроимпульсы от отдельного генератора (хотя и может генерировать прерывания для процессора в соответствии с установками в регистрах управления). Процессор обменивается принимаемыми и передаваемыми данными через память данных. Если процессор не изменит выдаваемые данные в некотором цикле, то контроллером СВП будет повторно передана на шину старая копия данных.

Данные забираются от приемников и выдаются передатчикам блоком исполнения расписания, который, обращаясь в память расписания, получает описание каждого нового слота и инструкции, какие операции выполнить в этом слоте (выдать данные, принять данные, синхронизировать время и др.). В выдаваемом сообщении каждый узел также дополнительно перечисляет узлы, от которых он успешно принял сообщения в прошлом цикле. На основе этой информации каждый узел может заключить, слышал ли его предыдущее сообщение некоторый другой абонент шины (и слышал ли кто-либо вообще). Данная информация помещается в регистры статуса и, наряду с признаками состояния приемников, передатчиков и узла в целом, позволяет программному обеспечению определить ситуации потери связи, отказа передатчиков/приемников и пр.

Описанный выше контроллер СВП реализован НПП «Дозор» совместно с ЗАО «ПКК Миландр» в виде микросхемы К5600ВГ2У.

Литература

1. *Клепиков В. И.* Отказоустойчивость распределенных систем управления. М. «Золотое сечение», 2014. - 392 с.
2. *Карпенко Е*. Возможности CAN-протокола// Современные технологии автоматизации, 1998, № 4. - С.16-20.
3. ГОСТ Р ИСО 11898-1-2015. Транспорт дорожный. Местная контроллерная сеть (CAN). Часть 1. Канальный уровень и передача сигналов.
4. ГОСТ Р ИСО 11898-2-2015. Транспорт дорожный. Местная контроллерная сеть (CAN). Часть 2. Устройство доступа к высокоскоростной среде.
5. *Синявский С., Шергин В., Власюк В.* Использование координатных датчиков в распределенной АСУ большого азимутального телескопа // Современные технологии автоматизации, 2012. - № 4. – С.68-73.

6. *Захаров Н. А., Калин С.В., Клепиков В. И., Подхватилин Д.С.* Архитектура распределенных систем управления жесткого реального времени // Радиоэлектронные и компьютерные системы – 2008. - № 5. С.57-61.
7. *Захаров Н. А., Клепиков В. И., Подхватилин Д.С.* Синхронно-временной протокол для распределенных систем управления // Автоматизация в промышленности – 2013. - № 2. С.37-39.