

Разработка архитектуры универсального модульного контроллера авионики

Зайцев Д. Ю.*, Неретин Е. С.**, Рамзаев А. М.***

*Объединенная авиастроительная корпорация - Центр комплексирования,
Авиационный переулок, 5, Москва, 125139, Россия*

*e-mail: dmitry.zaytsev@uac-ic.ru

**e-mail: evgeniy.neretin@uac-ic.ru

***e-mail: anton.ramzaev@uac-ic.ru

Аннотация

Предложена архитектура высокопроизводительного аппаратного контроллера, реализующего функции самолётных систем с уровнем гарантии проектирования FDAL «А», позволяющего минимизировать вероятность внесения ошибки при проектировании систем и значительно снизить расходы на разработку и сертификацию.

Ключевые слова: архитектура контроллера, центральный вычислитель авионики, интегральная модульная авионика.

Введение

Проектирование перспективного бортового радиоэлектронного оборудования (БРЭО) относится сегодня к приоритетным направлениям развития авиационной промышленности в Российской Федерации [25]. Основными направлениями

научных исследований в этой области является поиск новых концепций проектирования аппаратного и программного обеспечения и внедрение новых технологий, материалов и элементной базы в перспективные образцы авиационной техники. [23, 25, 27, 31, 33]

Разработчики авионики достигли больших успехов в решении частных задач проектирования отдельных компонентов, однако проблема построения интегрированных вычислительных комплексов БРЭО и их вычислителей до настоящего времени остаётся нерешённой в полной мере [24, 25, 27-29, 31, 32, 33, 39].

Внедрение концепции интегральной модульной авионики (ИМА) в перспективных объектах авиационной техники предполагает разделение функциональных компонентов авионики на три иерархических уровня [15, 20, 22, 24, 32, 34]:

- нижний уровень: унифицированные конструктивно-функциональные модули различного назначения, имеющие собственные вычислительные средства в компактном стандартизованном исполнении;

- средний уровень: мультипроцессорные вычислительные системы, создаваемые из модулей нижнего уровня и конструктивно выполненные в стандартизованном корпусе;

- верхний уровень: бортовая локальная вычислительная сеть, интегрирующая вычислительные средства систем среднего уровня на основе центрального сетевого интерфейса высокой пропускной способности.

Приведём ряд результатов, полученных отечественными и зарубежными предприятиями в этом направлении: ОАО «Раменское приборостроительное конструкторское бюро (бортовая центральная вычислительная система БЦВС-1 [19]), ФГУП «СПб ОКБ «Электроавтоматика» им. П.А. Ефимова» (изделие «Крейт» [18, 28, 30]), ОАО Холдинговая компания «АВИАПРОБИОР-ХОЛДИНГ», ОАО «Научно-конструкторское бюро вычислительных систем», НПО «Полет», ЗАО НПП «Авиационная и Морская Электроника» (многопроцессорный вычислительный комплекс «МПВК» [34, 36]), ФГУП «Государственный научно-исследовательский институт авиационных систем» [26], ОАО «Научно-конструкторское бюро вычислительных систем» (проект «Базис 5.0» [29]), ОАО «НИЦЭВТ» (работы в рамках НИР «Беркут», проект «Ястреб», проект «Ангара» [37-38]), ЗАО НТЦ «Модуль» (проекты «Борт-М», «БПТС-2» [17]), ОАО «НИИ ВК им. М.А. Карцева», ЗАО «МЦСТ» (проект «Эльбрус-90микро» [21]), КБ «Корунд-М» (проект «Багет» [16]), Thales Avionics (центральный вычислитель авионики СРЮМ [11], комплекс интегрированной модульной авионики Topdeck), Elbit Systems (центральный вычислитель авионики СРЮМ [6]), Rockwell Collins (интегрированный комплекс модульной авионики ProLine 21 [13]), Honeywell (бортовой комплекс Primus Epic [12]).

Все перечисленные изделия являются программно-аппаратными, что обязывает разработчиков проходить полный процесс сертификации в соответствии с требованиями DO-178C [7], DO-254 [8] и другими нормативными документами. Отметим, что любое изменение как аппаратной, так и программной

части изделия, требует значительных затрат на повторное прохождение процесса сертификации.

Высокий уровень развития микроэлектронных технологий, широкое распространение программируемых логических интегральных схем (ПЛИС) и специализированных интегральных схем (Application Specific Integrated Circuit – ASIC) позволяют проектировать перспективные устройства на базе полностью аппаратных средств без применения программного обеспечения (ПО). Применение данного подхода позволит значительно сократить сроки и затраты на разработку и сертификацию оборудования, а также снизить вероятность внесения ошибки при разработке за счёт исключения процессов разработки и верификации ПО.

Работа посвящена разработке высокопроизводительного аппаратного контроллера, реализующего функции самолётных систем с уровнем гарантии проектирования FDAL «А» в соответствии с ARP4754a [4, 35], позволяющего снизить расходы на разработку и сертификацию бортового оборудования за счёт отсутствия необходимости применения встроенного программного обеспечения.

Назначение уровня гарантии проектирования FDAL «А» требует применения в разрабатываемом устройстве механизмов, используемых для разделения устройства на отдельные составные части, обладающие такой независимостью, что для каждой из этих частей может быть установлен и подтверждён свой уровень гарантии проектирования.

Архитектура разрабатываемого контроллера

Разработанная архитектура контроллера включает в себя универсальную платформу, к которой подключаются необходимые аппаратные модули

приложений, обеспечивающие выполнение конкретных алгоритмов в зависимости от требуемых для решения задач.

Архитектура платформы базируется на применении концепции отражаемой памяти (reflective memory) [5, 14], то есть общей памяти для нескольких независимых устройств (приложений), предназначенных для строго детерминированного обмена данными между этими устройствами (приложениями). Проведённый анализ существующих устройств с данным устройством общей памяти, применяемых в авиационном приборостроении, выявил отсутствие устройств данного типа.

Прямая видимость памяти каждого приложения каждым приложением реализовано в архитектуре за счёт применения синхронизации (записи в общую память платформы и чтения из общей памяти платформы). Для исключения возможных коллизий при обращении приложений к общей памяти вводится контроллер платформы, также являющийся контроллером сети приложений, связанных между собой через отражаемую память. Контроллер платформы детерминирует данные в сети, генерируя периодические запросы на синхронизацию для каждого приложения.

Каждый модуль приложения выполняет исключительно назначенные ему функции. На одну платформу может быть установлено несколько одинаковых модулей приложений.

Предлагаемая архитектура универсального контроллера (рис. 1) является **модульной**, что обеспечивает свободную масштабируемость и гибкость, как при добавлении новых функций, так и модернизации уже существующих функций,

не затрагивая существующие модули, а высокопроизводительная платформа обеспечивает работу нескольких приложений в соответствии с концепцией ИМА [26].

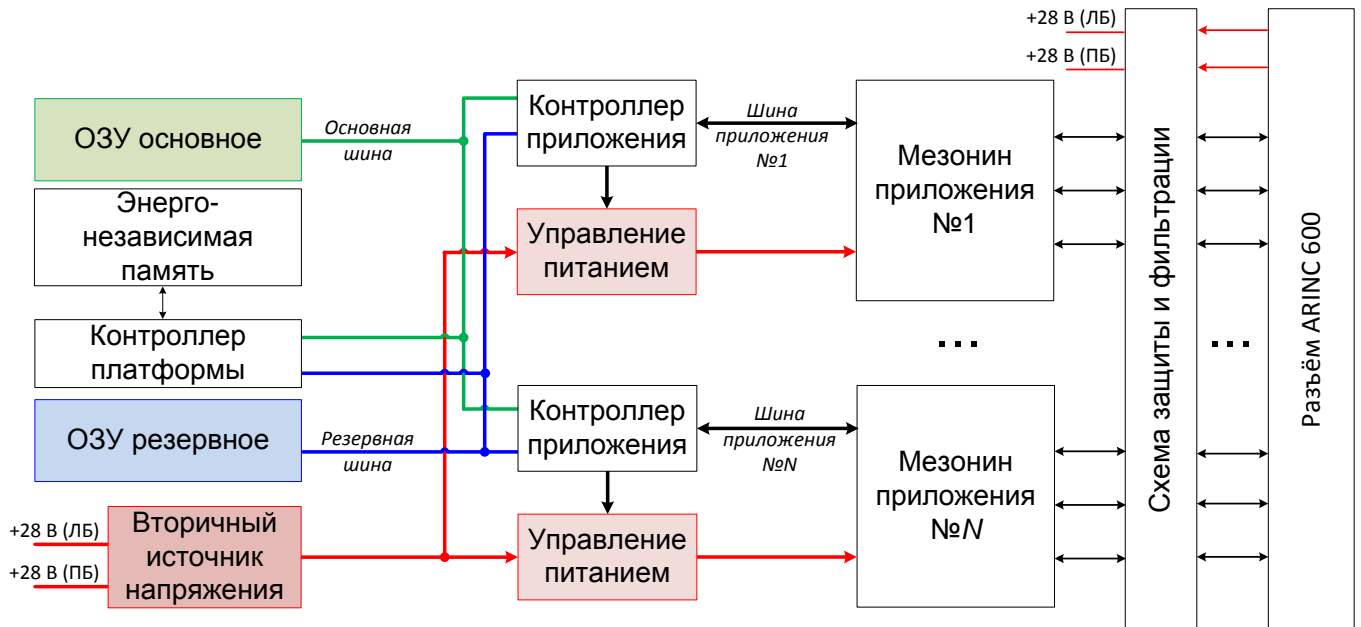


Рис. 1. Архитектура платформы универсального модульного контроллера

Разработанная архитектура платформы контроллера состоит из следующих основных узлов: контроллера платформы, контроллеров приложений, памяти и узла управления питанием.

Оперативное запоминающее устройство

В отличие от существующих реализаций концепции ИМА, предлагаемая платформа не содержит вычислительного устройства. Все вычисления производятся в приложениях, а основной задачей платформы является синхронизация памяти приложений, используя принцип разделения ресурсов.

Каждое приложение содержит разделённое на страницы оперативное запоминающее устройство (ОЗУ). Количество страниц памяти зависит от максимального количества приложений на платформе и в данном примере

реализации ограничивается пятью. Контроллер каждого приложения обеспечивает синхронизацию памяти своего приложения с памятью платформы.

Для обеспечения требований безопасности на платформе должны размещаться два независимых архитектурно различных модуля памяти: основная и резервная.

Основная память представляет собой синхронную динамическую память с произвольным доступом и удвоенной скоростью передачи данных второго поколения (DDR2).

Резервная память – это статическая память с произвольным доступом (SRAM).

Энергонезависимое запоминающее устройство

Энергонезависимая память выполняет функцию памяти встроенных средств контроля платформы. В ней сохраняется информация об отказах аппаратуры, либо о внештатных условиях работы платформы.

Энергонезависимая память должна представлять собой энергонезависимое статическое запоминающее устройство с произвольной выборкой и разбиваться на секторы и сегменты [38].

Вся энергонезависимая память разбивается на $(N+1)$ секторов: один служебный сектор и N информационных секторов. Информационные секторы предназначены для хранения информации за последние N полётов. Каждый информационный сектор разбивается на два сегмента: сегмент описания полета и сегмент данных.

Служебный сектор предназначен для хранения данных о порядковом номере последнего полёта (полетом называется период времени между перезагрузками или включениями платформы).

В сегменте описания полёта должны сохраняться: порядковый номер полёта, время начала полёта, дата начала полёта, бортовой номер летательного аппарата (ЛА), номер рейса.

В сегменте данных хранится информация об отказах аппаратуры и о нештатных условиях работы платформы.

Контроллер приложения

Контроллер приложения предназначен для обеспечения взаимодействия соответствующего приложения с платформой, а также для контроля его состояния и управления электропитанием. Каждое приложение платформы управляется контроллером.

Контроллер выполняет следующие основные функции – синхронизацию памяти приложения и платформы, контроль состояния приложения, передачу информации об отказах приложения в контроллер платформы, информационной развязки приложения и шин данных платформы (обеспечение гарантии информационного обмена платформы при отказе приложения), а также отключение питания приложения в случае выявления его нештатной работы (формирование соответствующей команды на узел управления питанием). Схема синхронизации памяти между платформой и приложениями представлена на рис. 2.

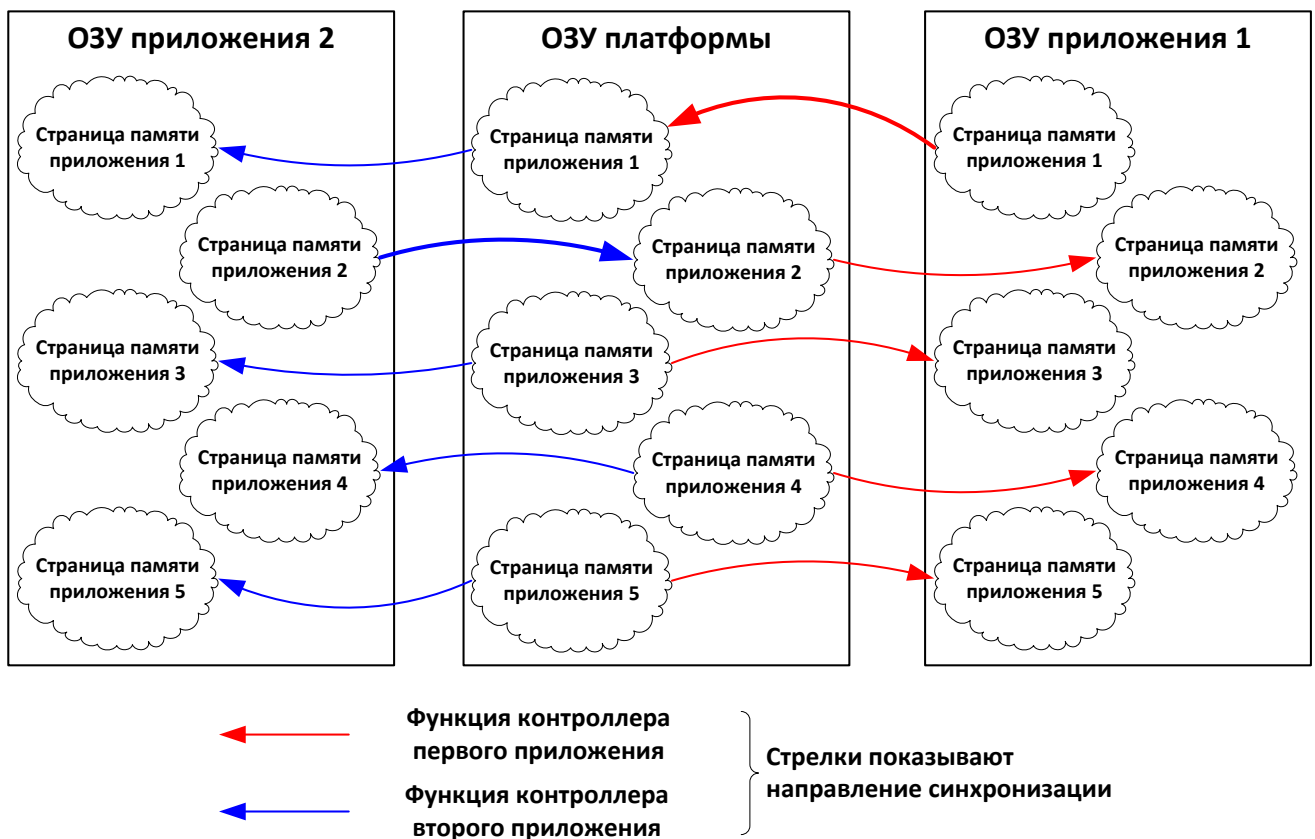


Рис. 2. Схема синхронизации памяти между платформой и приложениями

Контроллер каждого приложения выполняет синхронизацию памяти приложения с памятью платформы следующим образом:

- страница памяти подключённого к платформе приложения с номером, соответствующим номеру контроллера приложения, отображается в аналогичную страницу памяти платформы;
- все страницы памяти контроллера, кроме страницы с номером, соответствующим номеру контроллера приложения, отображаются в аналогичные страницы памяти приложения.

Контроллер представляет собой конечный автомат, реализованный на базе ПЛИС или в виде специализированной интегральной схемы (ASIC). Все пять контроллеров, размещённых на платформе, являются идентичными. Каждый

контроллер определяет своё положение по уникальному набору замкнутых на землю конфигурационных контактов. Разработанная структура контроллера представлена на рис. 3.

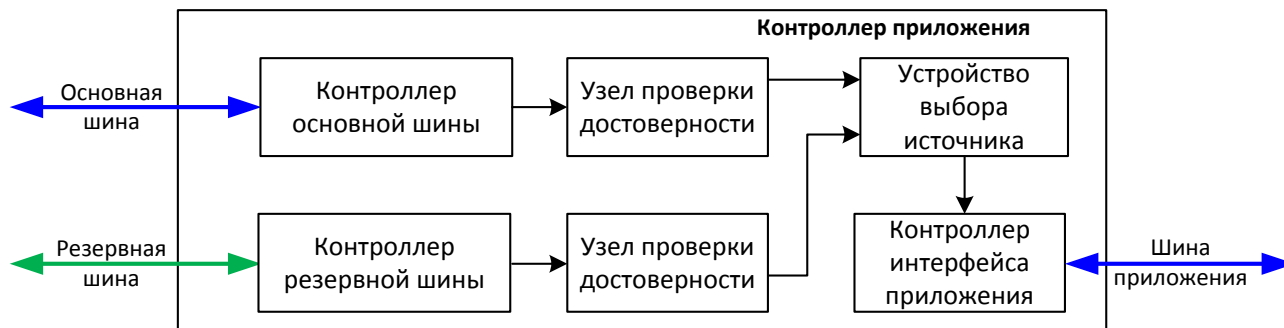


Рис. 3. Структурная схема контроллера приложения

Контроллеры шин обеспечивают двухсторонний обмен информацией по интерфейсам платформы. Узел проверки достоверности подсчитывает контрольную сумму страницы памяти и сравнивает её с подсчитанной в контроллере платформы. Если контрольная сумма страницы памяти, полученной по основной шине, совпадает с расчётной, то данная страница передаётся в приложение через контроллер интерфейса приложения. Если контрольная сумма страницы памяти, полученной по основной шине, не совпадает с расчётной, а контрольная сумма страницы памяти, полученной по резервной шине, совпадает с расчётной, то в приложение через контроллер интерфейса приложения передаётся страница памяти из резервного ОЗУ. Данные операции реализуются в устройстве выбора источника.

Контроллер платформы

Контроллер платформы, выполняет следующие основные функции: разделение времени доступа приложений к основной и резервной шинам данных,

проведение постоянного встроенного контроля памяти и вычисление контрольной суммы страниц памяти.

Контроллер платформы должен тактироваться с частотой не менее 40 МГц и работать по циклограмме, приведённой на рис. 4.

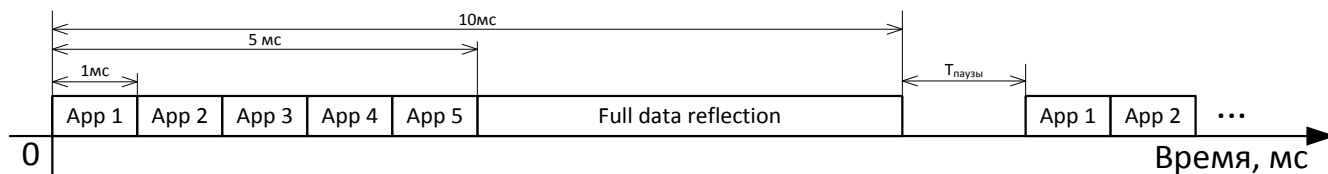


Рис. 4. Циклограмма работы контроллера платформы

Контроллеры приложений начинают синхронизацию областей памяти исключительно по запросу контроллера платформы. App X (X принимает значения от одного до пяти) – время, за которое контроллер приложения X должен скопировать собственную страницу памяти в ОЗУ платформы. Время App X ограничено 1 мс. После передачи областей памяти всех приложений в ОЗУ платформы будет находиться актуальное отражение памяти всех приложений, которое должно быть распространено всем приложениям. На это в циклограмме отводится 5 мс.

На синхронизацию каждого приложения отводится 20 мс. После того, как контроллер приложения выполнит синхронизацию, то есть обновит страницу памяти соответствующего приложения в памяти платформы, контроллер платформы выполняет вычисление контрольной суммы страницы памяти приложения, синхронизация памяти которого выполнялась на предыдущем шаге.

Аналогично контроллеру приложения, контроллер платформы представляет собой конечный автомат, реализованный на базе ПЛИС или в виде ASIC.

Узел управления питанием

Наряду с контроллером, каждое приложение сопровождается узлом управления питанием. Основными функциями этого узла являются измерение потребляемого приложением тока, передача полученного значения в контроллер приложения и отключение электроснабжения приложения при получении соответствующей команды от контроллера приложения.

Каждое приложение имеет ограничение на потребляемый ток, при превышении которого приложение должно быть принудительно остановлено («обесточено»). Значение этого порога хранится в памяти самого приложения в области конфигурации и передаётся в контроллер приложения при инициализации приложения платформой.

Для управления питанием применены твердотельные коммутационные элементы, обеспечивающие измерение тока, проходящего через коммутационный элемент.

Все узлы управления питанием приложений коммутируют напряжение, выдаваемое вторичным источником питания. Основными функциями вторичного источника питания являются:

- выбор источника питания (на борту ЛА электропитание устройства должно осуществляться от двух независимых каналов системы электроснабжения постоянного тока),

- преобразование напряжения бортовой сети в требуемые напряжения постоянного тока для обеспечения питания приложений.

Шины платформы

В состав разрабатываемой платформы входят три основные шины: основная, резервная и шина приложения.

К *основной шине* подключаются контроллер платформы, основное и резервное ОЗУ платформы и контроллеры всех приложений.

Резервная шина полностью дублирует основную шину (как по подключаемым абонентам, так и по циклограмме работы). Информация в резервной шине передаётся постоянно в соответствии с циклограммой передачи данных, но используется только в случае отказа основной шины или контроллера одного из подключенных к ней устройств.

Шина приложения работает в режиме точка-точка и объединяет каждое приложение с контроллером приложения.

Для выбора интерфейса при реализации основной и резервной шин произведена оценка требуемой пропускной способности. В соответствии с разработанной циклограммой память всех приложений должна обновляться в течение 10 мс. Так как синхронизируемая страница памяти каждого приложения составляет 1 МБ, общее количество приложений – пять, синхронизация память – в двух направлениях (от приложений в платформу и от платформы в приложения), то общая пропускная способность не должна быть менее 10 МБ / 10 с, что соответствует 1000 МБ/с.

В процессе выбора интерфейса проведено ранжирование интересов в соответствии с пропускной способностью, результаты которого представлены в табл. 1.

Табл. 1. Анализ характеристик интерфейсов

Интерфейс	Разрядность / частота	Полная пропускная способность	Полезная пропускная способность, Мбайт/с
PCI 2.0	32 / 33 МГц	1 Гбит/с	106,4
PCI Express 1.0 (x1)	32 / 33 МГц	2 000 Мбит/с	200
PCI 2.1-3.0	64 / 33 МГц	2 Гбит/с	212,8
AGP 1.0 (1x)	32 / 66 МГц	2 Гбит/с	212,8
Intel Hub Link	32 / 66 МГц	2 Гбит/с	212,8
VME320[10]	64 / 40 МГц	2560 Мбит/с	256
PCI 2.1-3.0	64 / 66 МГц	4 Гбит/с	426,4
AGP 1.0 (2x)	32 / 66 МГц	4,2 Гбит/с	426,4
RapidIO LP-Serial 4x/1,25	4x / 1,25 ГГц	4000 Мбит/с	400
RapidIO LP-LVDS	8 / 250 МГц	8528 Мбит/с	852,8
PCI-X	64 / 133 МГц	8528 Мбит/с	852,8
AGP 2.0 (4x)	32 / 66 МГц	8528 Мбит/с	852,8
PCI-X DDR (266)	64 / 266 МГц	17 066 Мбит/с	1706,4
AGP 3.0 (8x)	32 / 66 МГц	17 066 Мбит/с	1706,4
PCI Express 1.0 (x16)	32 / 66 МГц	32 000 Мбит/с	3200
AGP 3.0 (8x)	64 / 66 МГц	34 133 Мбит/с	1012,8
PCI-X QDR (533)	64 / 266 МГц	34 133 Мбит/с	3412,8
PCI Express 2.0 (x32)	64 / 266 МГц	128,00 Гбит/с	13107,2
QPI	64x / 2,40 ГГц	153,60 Гбит/с	15728,64
HyperTransport 2.0	32 / 1,4 ГГц	179,20 Гбит/с	18350,08
QPI	64x / 2,93 ГГц	187,52 Гбит/с	19202,048
PCI Express 3.0 (x32)	64x / 2,93 ГГц	204,80 Гбит/с	20971,52
QPI	64x / 3,20 ГГц	204,80 Гбит/с	20971,52
HyperTransport 3.1	32 / 3,20 ГГц	409,60 Гбит/с	41943,04

Существует достаточное количество интерфейсов, которые могут быть применены в составе разрабатываемой платформы. Учитывая такие факторы, как: опыт применения в авиации, популярность (анализировалось количество поставщиков контроллеров на рынке), сложность реализации контроллеров, наличие готовых сертифицируемых *IP*-ядер для реализации контроллера на базе ПЛИС из интерфейсов, представленных в табл. 1, выбраны следующие интерфейсы для применения в составе разрабатываемой платформы:

- PCI Express 1.0 (x16) – для основной шины;
- HyperTransport 2.0 – для резервной шины;

– PCI Express 1.0 (x1) – для шины приложения.

Приложения

В терминах концепции ИМА приложением [9, 18] называется аппаратный и/или программный модуль, который, будучи интегрированным в платформу, выполняет определённый набор функций. Разработанная платформа может предоставлять ресурсы приложениям, представленным в виде аппаратной и программно-аппаратной реализации. Преимущественной является разработка полностью аппаратных приложений за счёт исключения из процесса сертификации программного обеспечения. Кроме того, применение ПЛИС или ASIC является экономически целесообразным и энергетически эффективным решением для высокопроизводительных вычислений, которые могут потребоваться при реализации требуемой функции.

Приложение в рамках разрабатываемого устройства представляет собой мезонинный модуль, включающий в себя приёмопередатчики и средства обработки сигналов (параметров, потоков данных). Данный мезонинный модуль устанавливается на платформу в любой из универсальных разъёмов и получает доступ к ресурсам платформы: чтению страниц памяти всех приложений платформы и запись в собственную страницу памяти. Структура разрабатываемого приложения представлена на рис. 5.

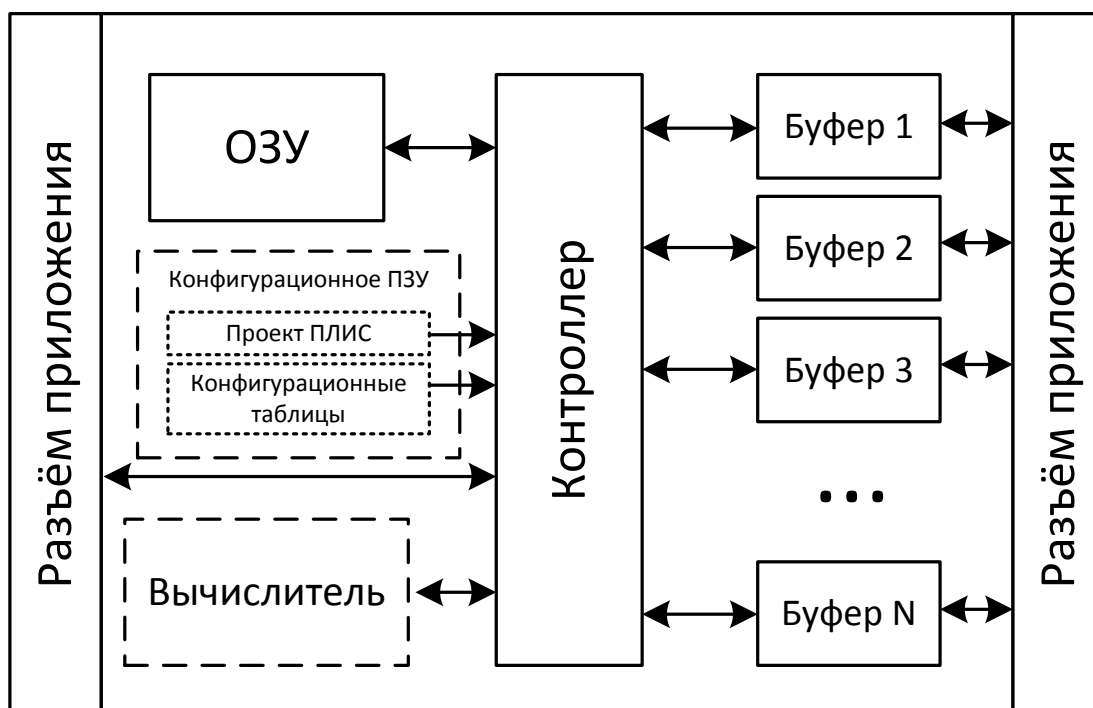


Рис. 5. Структурная схема приложения

Буферы служат для преобразования сигналов, формируемых сопряжёнными системами и передаваемыми по какому-либо интерфейсу авиационного применения в разрабатываемое устройство, в сигналы ТТЛ-логики для контроллера приложения и наоборот.

Под интерфейсами авиационного применения понимаются не только цифровые линии передачи данных (например, ARINC 429 [1], ARINC 825 [3], ARINC 664 [2]), но и аналоговые сигналы, передаваемые в виде напряжения постоянного тока, активного сопротивления, сигнала от синусно-косинусного вращающегося трансформатора или датчика приближения. В случае применения нецифровых сигналов под буферами подразумевается аналого-цифровой (АЦП) и/или цифро-аналоговый преобразователь (ЦАП) с измерительной схемой (например, мостовой).

Количество буферов в приложении зависит от функциональных требований к приложению, физически доступного на плате приложения места и доступных контактов внешнего соединителя, к которым подключено приложение. В случае необходимости коммутации силовых линий приложение может включать в себя набор твердотельных или электромеханических реле, однако их количество будет ограничено массо-габаритными требованиями к приложению.

ОЗУ, устанавливаемое в каждом приложении, необходимо для хранения зеркала памяти платформы. Данное ОЗУ должна иметь страничную организацию и его объём должен быть не менее объёма ОЗУ платформы. Страницы памяти, которые не участвуют в синхронизации с платформой, могут быть задействованы под внутренние задачи приложения.

Контроллер приложения – это устройство, реализованное на ПЛИС или в виде ASIC, которое обеспечивает функциональную обработку данных, получаемых от подключенных к внешнему соединителю буферов или от других приложений через общую память платформы.

Приложение может быть как универсальным, так и функционально специфическим. Предлагается использование ряда универсальных приложений, которые могут сопрягаться в готовое устройство без дополнительной аттестации.

В результате анализа существующих устройств сформирован следующий базовый список аппаратных приложений:

- приёмник/передатчик дискретных сигналов;
- приёмник/передатчик аналоговых сигналов (набор ЦАП и АЦП);
- контроллер бортовых интерфейсов (ARINC 429 и ARINC 825);

- контроллер сети AFDX (ARINC 664);
- контроллер логической обработки (универсальный контроллер, способный рассчитывать выходные параметры по логике, приведённой в конфигурационной таблице).

Конфигурационное постоянное запоминающее устройство (ПЗУ) устанавливается в приложении для выполнения двух функций:

- загрузки проекта ПЛИС при подаче питания в приложение (если разработчик приложения принял решение о применении ASIC в качестве контроллера приложения, то данный элемент может быть исключён);
- хранения конфигурационных таблиц данных, требуемых приложению для корректного взаимодействия с другими приложениями платформы.

Необходимость установки, а также тип и объём памяти конфигурационного ПЗУ определяется разработчиком. Каскадное подключение нескольких ПЗУ возможно в случае необходимости работы с большим объёмом конфигурационных данных.

Структурная схема, представленная на рис. 5, демонстрирует минимально необходимый набор узлов, достаточный для функционирования приложения (за исключением буферов – их наличие необязательно). Приложение может быть дополнено определённым набором узлов для выполнения требований по безопасности (например, установки дополнительного зеркала памяти, обратных связей формируемых линий, контроллеров встроенных средств контроля) или же функциональных требований (например, энергонезависимых запоминающих

устройств, или измерителей интервалов времени). Однако приложение может и не содержать буферов.

Конструктивное исполнение

Пример реализации конструкции платформы, позволяющей размещать на ней до пяти мезонинных модулей (приложений) и расположенной в корпусе типоразмера 2 *MCU* представлен на рис. 6. Предлагаемый размер платформы: длина – 270 мм, ширина – 170 мм.

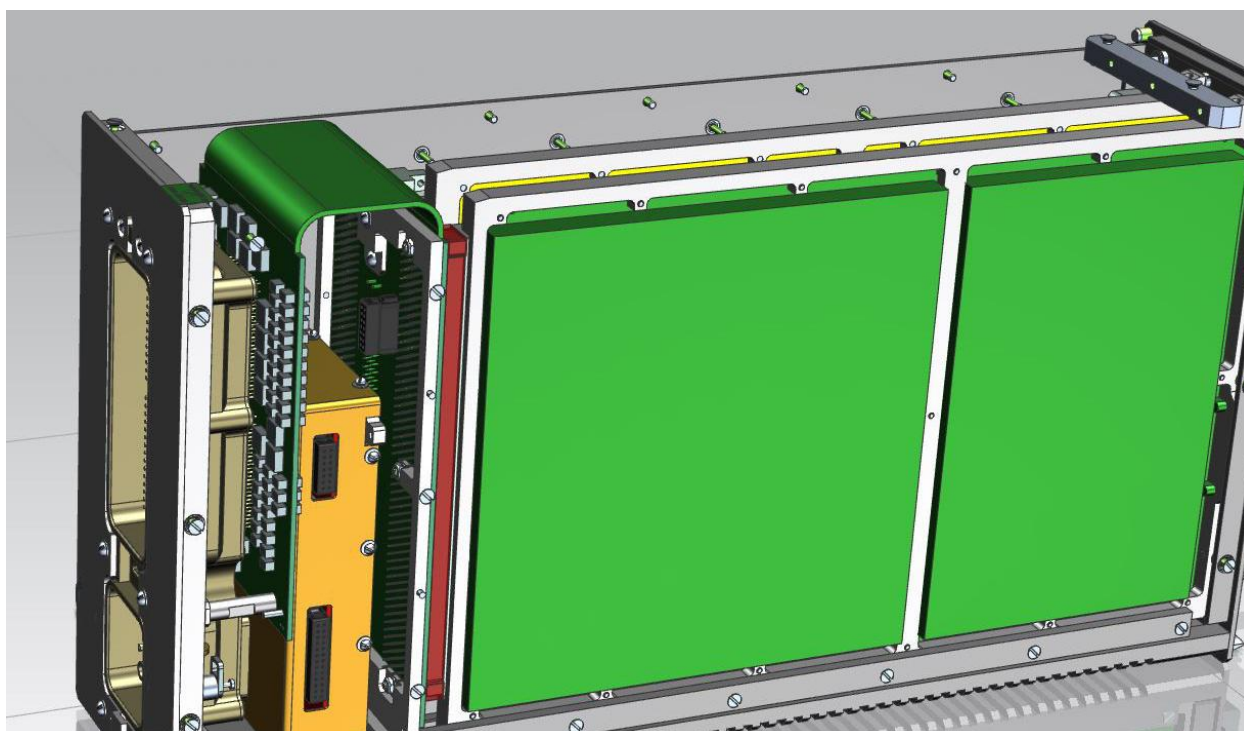


Рис. 6. Пример расположения платформы в корпусе типоразмера 2 MCU

На рис. 7 представлен пример платформы с установленными на ней пятью мезонинными модулями стандартного типоразмера 90x45 мм.

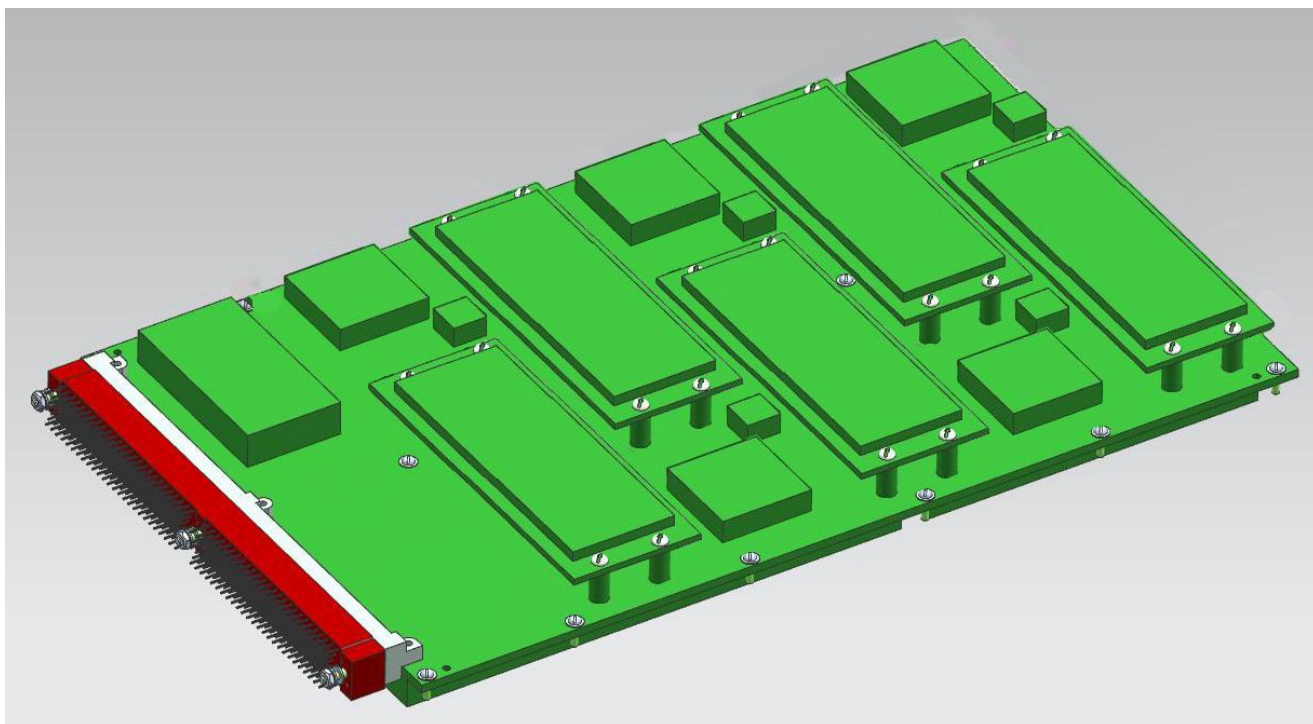


Рис. 7. Общий вид платформы с пятью мезонинными модулями.

Сертификационные аспекты разработки

Разработка изделия ведётся двумя параллельными процессами – разработкой платформы и разработкой приложений с последующей интеграцией системы ИМА. В качестве сертификационного подхода предлагается также использовать поэтапную аттестацию (модель ИМА), описанную в DO-297 [9].

Поэтапная аттестация позволит получить одобрение на устройство путём получения подтверждения соответствия как платформы, так и приложений заданным требованиям поэтапно. Весь процесс сертификации разделён на три этапа (рассматривается только уровень поставщика изделия):

- аттестация платформы (Task 1 в соответствии с DO-297);
- аттестация приложений (Task 2 в соответствии с DO-297);
- аттестация системы ИМА (Task 3 в соответствии с DO-297).

Зачётная оценка, полученная при выполнении каждого этапа, используется при получении итогового сертификационного зачёта. Поэтапная аттестация обеспечивает возможность интеграции и аттестации новых приложений в устройстве с одновременной эксплуатацией ранее разработанных приложений без необходимости их повторной аттестации.

Задача аттестации платформы сводится к разработке изделия в соответствии с требованиями DO-254 [8] по уровню гарантии разработки IDAL A. В соответствии с меморандумом EASA SWCEH-001 [10] разработка печатных плат, сборок и устройства в целом должна производиться по IDAL D. Процесс разработки изделия подчиняется системному подходу и включает планы, требования, а также подтверждение соответствия этим планам и требованиям.

По результатам разработки и аттестации платформы выдаётся одобрение авиационных властей.

Аттестация приложений аналогична процессу аттестации платформы. Единственным отличием от описанного процесса является необходимость дополнительного доказательства возможности корректного функционирования приложения на конкретной платформе, для которой оно разрабатывается. Таким образом, приложение разрабатывается не отдельно от платформы, а совместно с ней. Все приложения разрабатываются в соответствии с требованиями ARP4754a [4] и DO-254 [8].

В случае, когда разработчиком принято решение применять контроллеры, включающие в свой состав программное обеспечение, необходимо выполнить полный спектр работ в соответствии с требованиями DO-178C [7], включая полную

верификацию. Ответственность за аттестацию приложения лежит исключительно на разработчике приложения. Ответственность же разработчика платформы в данном случае состоит в предоставлении разработчику приложения всего необходимого для сопряжения этого приложения с платформой.

Одним из самых сложных этапов процесса разработки и сертификации является выполнение задачи 3 (Task 3) в соответствии с DO-297 [9]. Это аттестация системы ИМА, которая включает в себя интеграцию всех приложений и платформы в единую систему. Целью данной задачи является подтверждение того, что каждое интегрированное в платформу приложение выполняет назначенные ему функции и не оказывает воздействия на другие приложения платформы.

Применяя предлагаемый подход невозможно полностью избежать процесса разработки и сертификации. Даже если функция, возложенная на предлагаемое устройство, может быть решена средствами платформы с базовым набором приложений, необходимо проводить интеграцию платформы – то есть выполнять задачу 3 в соответствии с DO-297 [9].

Платформа отвечает требованиям по обособлению: обеспечению возможности выполнения нескольких приложений на одной платформе, возможно, с различными уровнями критичности, с полной изоляцией по времени и памяти между ними. В частности, обособление гарантирует, что приложение, независимо от сбоя, произошедшего в нём, не сможет повлиять на работу других приложений. Из этого определения следует, что раздел – это единица разделения по времени выполнения и памяти. Предлагаемая концепция платформы обеспечивает данное обособление за счёт следующего.

Во-первых, каждому приложению выделена определённая страница общей памяти. Каждый из контроллеров приложения при инициализации определяет свое положение и соответствующий ему номер страницы, а приложение обращается к общей памяти через контроллер приложения.

Во вторых, время доступа каждого приложения к общим шинам данных для синхронизации памяти детерминировано. Доступ к шине происходит исключительно по запросу контроллера платформы. Попытка обращения к шине без получения соответствующей команды приводит к выключению приложения (снятием питания с мезонинного модуля приложения).

В третьих, каждое приложение гальванически развязано от общей электросети платформы через твердотельный коммутационный элемент. Коммутационным элементом управляет контроллер приложения. В случае необоснованного повышения токопотребления приложение будет отключено от общей электросети.

Выводы

В работе предложена архитектура универсального модульного аппаратного контроллера с уровнем гарантии проектирования FDAL «А», реализующего основные аспекты концепции ИМА и удовлетворяющего требованиям действующих руководящих документов авиационных властей России, Европы и США.

Предложенная архитектура универсального контроллера является модульной, что обеспечивает свободную масштабируемость и гибкость в зависимости от реализуемого набора функций самолётных систем.

Применённый подход позволяет снизить расходы на разработку и сертификацию контроллера за счёт реализации функций самолётных систем аппаратными методами без использования встроенного программного обеспечения.

Предложенная архитектура может применяться при разработке бортовых вычислителей перспективных объектов авиационной техники.

Библиографический список

1. ARINC Specification 429P1-17. MARK 33 Digital Information Transfer System (DITS). Part 1 "Functional Description, Electrical Interface, Label Assignments and Word Formats". – The USA: Aeronautical Radio, Inc. 2004. 309 p.

2. ARINC Specification 664P1-1. Aircraft Data Network. Part 1 "Systems Concepts and Overview". – The USA: Aeronautical Radio, Inc. 2006. 51 p.

3. ARINC Specification 825-2. General Standardization of CAN (Controller Area Network) Bus Protocol for Airborne Use. – The USA: Aeronautical Radio, Inc. 2011. 170 p.

4. ARP4754. "Aerospace Recommended Practice. Guidelines for Development of Civil Aircraft and Systems, Revision A". – The USA: SAE International. – December 2010. – 115 p.

5. Catani, L., Gabrielli, E., Gatta, M., Sabene, M., Salamon, A., Salina, G. A general purpose reflective memory board for accelerator data acquisition and control system applications // IEEE Nuclear Science Symposium Conference Record. – vol. 14, no. 141. 2005. pp. 692-695.

6. Core Processing & Input/Output Module (CPIOM) for the MS-21 Program. Technical Proposal Specification. – Israel, Haifa: Elbit Systems - Aerospace. 2012. 98 p.

7. DO-178C. Software Considerations in Airborne Systems and Equipment Certification. – The USA, Washington, DC: RTCA, Inc. 2011. 144 p.
8. DO-254. Design Assurance Guidance for Airborne Electronic Hardware. – The USA, Washington, DC: RTCA, Inc. 2000. 137 p.
9. DO-297. Integrated Modular Avionics (IMA) Development. Guidance and Certification Considerations. – Washington, DC: RTCA, Inc. 2005. 137 p.
10. EASA CM-SWCEH-001. Certification Memorandum. Development Assurance of Airborne Electronic Hardware. – EU: EASA. 2011. 66 p.
11. MS-21 Program Integrated Modular Avionics System. CPIOM Specification. – EU: THALES Avionics. 2012. 97 p.
12. Primus® Epic [Электронный ресурс]. URL: <https://aerospace.honeywell.com/en/products/integrated-avionics/primus-epic> (дата обращения 01.12.2015).
13. Pro Line 21™ Integrated Avionics System. [Электронный ресурс]. URL: https://www.rockwellcollins.com/Data/Products/Integrated_Systems/Flight_Deck/Pro_Line_21_Integrated_Avionics_System.aspx (дата обращения 01.12.2015).
14. Real Time Networking with Reflective Memory™. – The USA: GE Fanuc. 2007. 10 p.
15. Агеев В.М., Павлова Н.В. Приборные комплексы летательных аппаратов и их проектирование. – М.: Машиностроение, 1990. - 432 с.
16. Багет. Семейство ЭВМ для специальных применений. – М.: Корунд-М, 2004. - 5 с.

17. Блок преобразования телевизионных сигналов (БПТС-2). [Электронный ресурс]. URL:

http://www.module.ru/catalog/avia/blok_preobrazovaniya_televizionnih_signalov_bpts2/

(дата обращения 01.12.2015).

18. Платформа интегрированной модульной авионики. Патент на полезную модель RU №108868 U1 / Богданов А.В., Васильев Г.А., Виноградов П.С., Егоров К.А., Зайченко А.Н., Ковернинский И.В., Петухов В.И., Романов А.Н., Смирнов Е.В., Уткин Б.В., Федосов Е.А., Шукало А.В. Бюл. №27, 27.09.2011.

19. Бортовая центральная вычислительная система [Электронный ресурс]. URL:

http://www.rpkb.ru/lines-of-business/electronic_direction/on-board-computers/onboard-central-computer-system/index.php?sphrase_id=3710 (дата обращения 01.12.2015).

20. Буравлев А., Чельдиев М., Барыбин А., Костенко В., Тумакин Д., Петров Г. Масштабируемые мультипроцессорные вычислительные системы высокой производительности // Современные технологии автоматизации. 2009. № 3. С. 72–82.

21. Вычислительный комплекс Эльбрус-90 микро исп.52. [Электронный ресурс]. URL: <http://www.mcst.ru/14-15.htm> (дата обращения 01.12.2015).

22. Гатчин Ю.А., Жаринов И.О. Основы проектирования вычислительных систем интегрированной модульной авионики: монография. – М.: Машиностроение, 2010. - 224 с.

23. Джанджгава Г.И. Авионика пятого поколения: новые задачи – новая структура // Вестник авиации и космонавтики. 2001. № 5. С. 8–10.

24. Джонсон К., Леру П. Использование технологии объединения ресурсов для создания безопасных отказоустойчивых военных систем // Современные технологии автоматизации. 2007. № 4. С. 72–76.

25. Евгенов А.В. Направления развития интегрированных комплексов бортового оборудования самолетов гражданской авиации // Авиакосмическое приборостроение. 2003. № 3. С. 48–53.

26. Платформа интегрированной модульной авионики. Патент RU №2413280 С1 / Егоров К.А., Итенберг И.И., Ковернинский И.В., Тимченко А.П., Федосов Е.А., Чуюнов Г.А. Бюл. № 6, 27.02.2011.

27. Ефанов В.Н., Бодрунов С.Д. Открытые архитектуры в концепции авионики пятого поколения // Мир авионики. 2004. № 5. С. 20–28.

28. Жаринов О.О., Видин Б.В., Шек-Иовсепянц Р.А. Принципы построения крейта бортовой многопроцессорной вычислительной системы для авионики пятого поколения // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2010. № 4 (68). С. 21–27.

29. Итенберг И. Интегрированная модульная электроника – новая стратегия на рынке приборостроения // Новый оборонный заказ. Стратегии. 2010. № 5. с. 64–65.

30. Кузнецова О.А. Оценка надежности структурно избыточных комплексов авионики с учетом среднего времени между восстановлениями при отказах // Изв. вузов. Приборостроение. 2012. Т. 55. № 3. С. 65–69.

31. Михайлуца К.Т., Чернышев Е.Э., Шейнин Ю.Е. Основные архитектурные концепции информационно-вычислительной среды бортового оборудования

перспективных летательных аппаратов. Современные технологии извлечения и обработки информации. – СПб: Радиоавионика, 2001. - 175 с.

32. Павлов А.М. Принцип организации бортовых вычислительных систем перспективных летательных аппаратов // Мир компьютерной автоматизации. 2001. № 4 [Электронный ресурс]. URL: www.mka.ru/?p=41177 (дата обращения 01.12.2015).

33. Зайцев Д. Ю., Неретин Е. С. Дистанционный концентратор данных // Справочник инженера. 2014. №5. С. 41–46.

34. Пятницких А. Бортовые компьютеры: варианты построения готовых систем // Современные технологии автоматизации. 2008. № 2. С. 20–24.

35. Р4754. Руководство по процессам сертификации высокоинтегрированных сложных бортовых систем воздушных судов гражданской авиации. – М.: Межгосударственный авиационный комитет, 2007. - 103 с.

36. Севбо В., Орлов А., Лошаков А. Многопроцессорный вычислительный комплекс для задач «жесткого» реального времени // Современные технологии автоматизации. 2007. № 3. С. 32–38.

37. Слуцкий А.И., Симонов А.С., Казаков Д.В. Универсальная высокопроизводительная вычислительная платформа «Ангара» // Научные технологии. 2014. № 1. Т. 15. С. 17–20.

38. Слуцкий А., Эйсымонт Л. Российский суперкомпьютер с глобально адресуемой памятью // Открытые системы. 2007. № 9 [Электронный ресурс]. URL: <http://www.osp.ru/os/2007/09/4569294> (дата обращения 01.12.2015).

39. Хорошевский В.Г. Архитектура вычислительных систем: Учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2008. - 520 с.