УДК 004.054 DOI: 10.31114/2078-7707-2020-4-134-139

Построение высокопроизводительной СнК на основе 16-разрядного процессорного ядра

В.Я. Архипкин[1], М.И. Дябин[2], В.В. Ерохин[3], Ю.Л. Леохин[4]

[1]ООО «Аккорд», генеральный директор, г. Москва, info@akkordsystems.ru

[2]ООО «Каскад», ведущий инженер, г. Москва, dyabin@mail.ru

[3]АО «НИИМА «Прогресс», начальник отдела разработки аппаратуры связи, г. Москва, vladimir.v.erokhin@gmail.com

[4] Московский технический университет связи и информатики, проректор по научной работе, г. Москва, y.l.leokhin@mtuci.ru

***Аннотация* — В статье приводится описание одного из способов построения высокопроизводительной системы на кристалле для встроенных применений на основе 16-разрядного процессорного ядра и широкой 64-разрядной шины. Даётся обоснование правильности выбранного подхода в плане использования 16-разрядного i80186-совместимого процессорного ядра в качестве основы для построения системы. Отмечается вклад 64-разрядной шины и адаптированных к этой шине каналов прямого доступа к памяти в повышение производительности системы при работе с большими потоками данных. Приводится пример организации работы высокопроизводительного устройства (криптомодуля) в составе системы с демонстрацией преимущества использования широкой шины при обработке больших массивов данных.**

***Ключевые слова* — шина, конвейер, процессорное ядро, контроллер ПДП.**

I. Введение

При построении систем на кристалле (СнК) для встроенных применений приходится принимать во внимание множество факторов, важнейшими из которых являются компактность проекта и его производительность, причем эти факторы, как правило, находятся в некотором противоречии друг с другом. Например, использование 8-разрядного процессорного ядра обеспечивает компактность системы, однако с производительностью в этом случае могут быть серьёзные проблемы. И тем не менее приемлемое решение проблемы высокой производительности при использовании процессорного ядра малой разрядности вполне можно найти, особенно для задач управления и обмена данными, не требующих значительного объёма вычислений.

В настоящее время в качестве основы при разработке СнК разного типа и специализации максимально широкое распространение получили 32-разрядные процессорные ядра ARM. Для этой архитектуры наработано ПО практически любого уровня – от компиляторов и отладчиков до операционных систем, включая Linux [1].

В данной статье рассматривается СнК, предназначенная для управления потоками данных, построенная на основе 16-разрядного процессорного ядра. Структурная схема СнК представлена на рис. 1.

Основной особенностью задачи управления потоками данных является значительное количество разнообразных пересылок, при этом большого количества вычислений не требуется. Например, надо принять данные из Ethernet, пропустить их через блок криптообработки, после чего переслать зашифрованные данные в модем. Одновременно в обратном порядке требуется обработать встречный поток данных.

Очевидно, что если этими процессами будет заниматься непосредственно процессор, то хорошей производительности добиться не удастся. Гораздо большего эффекта можно достичь, если пересылки данных возложить на контроллер прямого доступа к памяти (ПДП). Но в этом случае большая разрядность процессора и его производительность в арифметико-логических операциях совершенно не требуется. Вместе с этим, для увеличения объёма передаваемых данных в единицу времени, имеет смысл использовать широкую шину (например, шину ёмкостью 64 разряда), а не использовать расширение шины процессора с невысокой разрядностью. Если идти таким путём, то вполне допустимо строить СнК передачи данных на 16-разрядном процессорном ядре с использованием широкой 64-разрядной шины с адаптированным для этой шины контроллером ПДП [2] .

В качестве основы для построения рассматриваемой СнК было использовано процессорное ядро, полностью совместимое по архитектуре и системе команд с 16-разрядным микроконтроллером i80186 [3, 4].

В своё время микроконтроллеры i80186 получили достаточно широкое распространение благодаря ряду достоинств. Прежде всего это хорошо известная и активно используемая архитектура, для которой было наработано значительное количество программ и приложений. Разработаны известные операционные

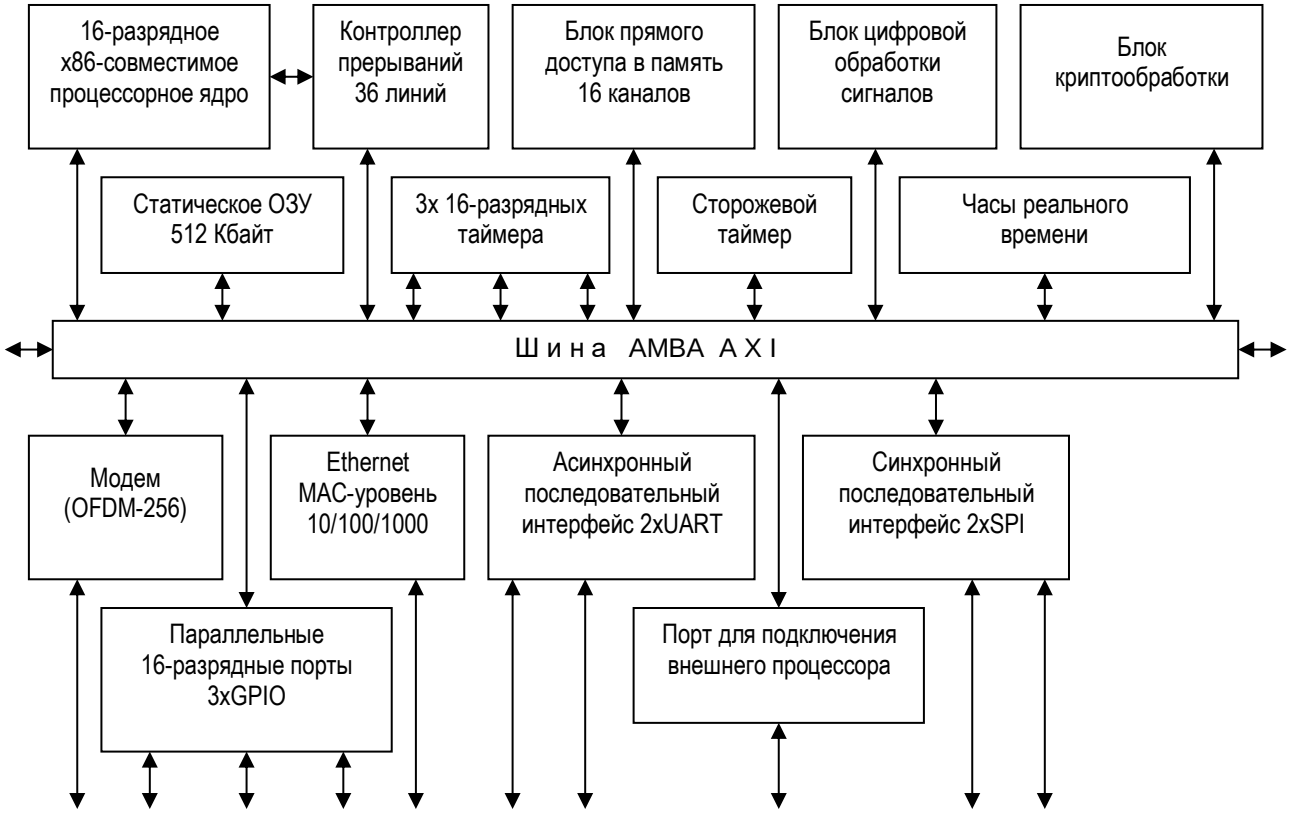

Рис. 1. Структурная схема СНК

системы, разработан и адаптирован весь необходимый программный инструментарий (компиляторы, отладчики, симуляторы) [5].

Вторым важным достоинством данной архитектуры является развитая система команд и, как следствие, компактный исполняемый код. Указанное свойство позволяет существенно снизить требования к объёму памяти, необходимой для исполнения программы, что, в свою очередь, при разработке системы позволяет обойтись без внешней динамической памяти, а всю память разместить внутри кристалла. Эта особенность имеет важное значение при построении оптимальных по аппаратуре и энергопотреблению систем, предназначенных для встроенных применений [4].

Далее следует более подробное рассмотрение ряда мер, предпринятых для повышения общей производительности СнК.

II. Оптимизация процессорного ядра

Основными факторами, влияющими на производительность процессорного ядра, являются хорошо организованная структура, применение конвейерной обработки потока команд, а также глубина конвейера. Применение глубокой конвейерной обработки при разработке процессорного ядра (в нашем случае до 7-ми уровней, включая предвыборку команд) позволило уменьшить число тактов, затрачиваемых на обработку одной команды. Для большинства регистровых операций это число составило один такт системной частоты, для операций, в которых один из операндов располагается в памяти, на выполнение операции затрачивается два или три такта системной частоты.

Отдельно следует отметить результат оптимизации команд умножения и сдвигов. Если в оригинальном микроконтроллере i80186 операции умножения выполнялись последовательно за много тактов, а время выполнения операций сдвигов зависело от количества разрядов, на которое нужно было сдвинуть операнд, то в оптимизированном процессорном ядре операции байтового умножения выполняются за 2 такта, операции пословного умножения выполняются за 4 такта, а все операции сдвигов выполняются за 1 такт системной частоты, независимо от числа разрядов, на которое необходимо сдвинуть операнд.

Ещё одним направлением в плане оптимизации процессорного ядра стала оптимизация механизма взаимодействия процессора с памятью команд и данных. Микроконтроллеры i80186 по своей архитектуре относятся к классу архитектур фон Неймана, для которых характерно использование единого интерфейса между процессором и памятью для загрузки команд и данных. В оригинальном микроконтроллере такой подход к организации обменов между процессором и памятью не являлся проблемой, поскольку интерфейс успешно справлялся с потоком команд и данных, передаваемых по единой шине. Однако, в процессе оптимизации процессорного ядра, приведшей к уменьшению количества тактов, затрачиваемых на выполнение большинства

команд, стало ясно, что необходим другой подход в организации обменов между процессором и памятью команд и данных.

В свете сказанного, в новой разработке было принято решение использовать подход, характерный для Гарвардской архитектуры, в которой потоки команд и данных, как правило, разнесены по разным шинам и процессы загрузки команд и обмена данными могут быть совмещены по времени.

Далее логика разработки привела к необходимости включения в состав системы кэш команд и кэш данных, и замены стандартной шины микроконтроллера i80186 на современную процессорно-независимую шину AMBA AXI. Особенности этой шины и её вклад в повышение общей производительности системы будут рассмотрены в следующей главе.

Образец микросхемы с оптимизированным процессорным ядром, изготовленной по малопотребляющей технологии с проектными нормами 65 нм, показал устойчивый результат по максимальной тактовой частоте, которая составила 650 МГц.

III. Использование широкой шины

Поскольку рассматриваемая архитектура является классической шинной архитектурой, то очевидно, что именно шина является тем узким местом, через которое потоки команд и данных направляются к своим потребителям. Чтобы снизить нагрузку на шину и увеличить её пропускную способность, было принято решение вместо штатной 16-разрядной шины микроконтроллера i80186 применить в СнК 64-разрядную шину AMBA AXI. В этой шине каналы чтения и записи данных имеют разрядность 64 бита.

Для ускорения процесса выборки команд и данных в ходе выполнения программы в состав СнК были включены кэш команд и кэш данных. Выборка команд и данных из кэш-памяти осуществляется 16-разрядными словами. Однако, загрузка команд и данных в кэш-память из основной памяти производится 64-разрядными словами. Поскольку в процессе выполнения программы основной объём обменов между процессором и памятью составляют обращения процессора в кэш-память, то нагрузка на шину AMBA AXI в значительной степени определяется активностью каналов ПДП, обслуживающих конкретные устройства, задействованные при выполнении данной программы.

В процессе оптимизации системы необходимо было определить оптимальный объём кэш-памяти, необходимый для функционирования системы. Ряд экспериментов показал, что для обеспечения приемлемой производительности интерфейсов команд и данных вполне достаточно блоков по 8 Кбайт кэш-памяти команд и кэш-памяти данных. В то же время такой объем кэш-памяти является не слишком обременительным с точки зрения аппаратных затрат с учетом того, что объем основной памяти может варьироваться в проектах от 128 до 512 Кбайт.

IV. Использование каналов прямого доступа к памяти

В рассматриваемую СнК встроено 16 каналов ПДП, при этом каждый канал ПДП закреплён за своим устройством и может быть синхронизирован на обмен данными от этого устройства. Для того, чтобы в полной мере использовать возможности 64-разрядной шины, каналы ПДП также спроектированы 64-разрядными. При этом у каналов ПДП остаётся возможность вести обмен словами меньшей разрядности (по 8, 16, 32 бита).

При обслуживании медленных устройств с малой разрядностью (в качестве примера можно привести последовательный интерфейс UART), эффект от расширения разрядности шины и разрядности каналов ПДП остаётся минимальным или нулевым. Однако, при обслуживании устройств с высокой интенсивностью обмена данными (например модуля криптообработки или OFDM-модема), при использовании каналов ПДП в 64-разрядном режиме интенсивность потока данных увеличивается в 4 раза по сравнению с 16-разрядным режимом. С другой стороны, количество обменов по шине при пересылке равнозначного по размеру блока данных уменьшается тоже в 4 раза. При этом очевидно, что само устройство также должно быть настроено на обмен 64-разрядными словами.

К таким устройствам в составе СнК можно отнести модуль криптообработки данных, блок цифровой обработки сигналов, интерфейс Ethernet, OFDM-модем.

В общем плане каналы прямого доступа к памяти равноправны и могут быть настроены на любые типы пересылок из числа заложенных в эти устройства. Однако каналы ПДП, обслуживающие интерфейс Ethernet на передачу и приём данных, имеют ряд особенностей, о которых следует упомянуть отдельно. Так канал ПДП, обслуживающий интерфейс Ethernet на передачу данных, обеспечивает повторную передачу фрейма в автоматическом режиме в случае неудачной попытки при передаче этого фрейма контроллером Ethernet. Канал ПДП, обслуживающий интерфейс Ethernet на приём данных, автоматически переключается на приём очередного фрейма, переходя к новому буферу данных для размещения этого фрейма в памяти. Максимальное число таких буферов может достигать шестидесяти четырёх, при этом размер каждого буфера равен максимальному размеру пакета Ethernet. В случае заполнения всех буферов из заданного в данном сеансе количества, канал ПДП автоматически прекращает приём данных из Ethernet и даёт сигнал обслуживающей программе о переполнении зарезервированной памяти.

Очевидно, что при таком подходе в плане использования каналов ПДП для организации работы с интерфейсом Ethernet, на процессор возлагается минимум нагрузки, что несомненно сказывается на повышении общей производительности системы.

Помимо прямого эффекта от увеличения разрядности шины и каналов ПДП, в рассматриваемой СнК с шиной AMBA AXI получаем ещё один важный эффект, который определённым образом влияет на общую

производительность системы. Эксперименты, проведённые на аналогичной СнК, построенной на основе стандартной шины микроконтроллера i80186, показали, что при одновременном запуске в работу 3-х каналов ПДП с максимальной интенсивностью обменов (режим асинхронного обмена данными), они полностью загружают шину, не оставляя возможности для осуществления обменов четвёртому каналу ПДП. В случае использования в СнК шины AMBA AXI, эффект блокирования каналов ПДП отсутствует. Можно запустить в работу с максимальной интенсивностью все 16 каналов ПДП, при этом ресурс шины будет равномерно (при равноприоритетном режиме работы) распределён между всеми каналами ПДП.

Следует отметить, что контроллер ПДП имеет два режима организации приоритетов: режим с фиксированными приоритетами и режим с вращающимися приоритетами. При работе с фиксированными приоритетами наивысший приоритет имеет назначенное устройство, далее приоритеты уменьшаются в порядке увеличения номера устройства. Вращающиеся приоритеты подразумевают, что в каждый отдельный момент времени наивысшим приоритетом обладает устройство, которое обслуживается; как только оно будет обслужено, оно получает самый низкий приоритет, а наиболее высокий приоритет получает устройство, следующее по номеру за обслуженным, которое в этот момент выставило свой запрос на обмен.

V. Обеспечение целостности данных

В общем случае в системе имеется как минимум два устройства, которые могут иметь доступ к одним и тем же данным, расположенным в основной памяти – это контроллер ПДП и процессор. В реальности их больше, поскольку каждый канал ПДП обслуживает отдельное устройство, и, следовательно, каждое из этих устройств может иметь доступ к данным, разделяемым с процессором или друг с другом.

Задача обеспечения целостности данных возникает вследствие того, что между процессором и основной памятью есть хранилище-посредник, в качестве которого выступает кэш-память. Каналы ПДП обмениваются данными непосредственно с основной памятью. Процессор же обращается к данным в кэш-память, при этом в кэш-память данные поступают из основной памяти (подгрузка данных из основной памяти производится в тот момент, когда требуемых данных в кэш памяти не оказывается).

Для простоты можно рассмотреть две ситуации, когда требуется вмешательство контроллера когерентности в решение задачи обеспечения целостности данных. При каждом обращении со стороны процессора за данными в кэш-память необходимо следить за тем, чтобы данные в кэш-памяти соответствовали тем же данным, расположенным в основной памяти. Если в данных обнаруживается несоответствие, то требуется обновление запрошенных данных в кэш-памяти. Такая ситуация может возникнуть в результате модификации данных в основной памяти при работе с этими данными каналов ПДП. Вторая ситуация может возникнуть при обращении каналов ПДП к данным для чтения из основной памяти. В этом случае контроллер когерентности должен отследить соответствие запрошенных из основной памяти данных аналогичным данным, расположенным в кэш-памяти. Если обнаруживается несоответствие данных, размещённых в кэш-памяти и в основной памяти, то необходимо эти данные из кэш-памяти сбросить в основную память.

В реальности ситуация выглядит несколько сложнее, поскольку одни и те же данные могут одновременно располагаться в основной памяти, в кэш команд и в кэш данных. В результате алгоритм динамической синхронизации данных в трёх разных хранилищах оказывается достаточно сложным и требует существенных аппаратных затрат для своей реализации.

VI. Организация работы по широкой шине

В качестве примера организации работы высокопроизводительного устройства в составе СнК с использованием широкой шины можно привести работу с модулем криптообработки данных.

Предположим, нам необходимо зашифровать блок данных с использованием некоторого стандартного криптоалгоритма. Перед запуском криптомодуля в работу предварительно требуется настроить криптомодуль на обмен данными посредством ПДП, загрузить в регистры криптомодуля ключи и другие настроечные данные, задать режим работы криптомодуля, после чего запустить криптомодуль на выполнение операции.

Поскольку в процессе работы криптомодуль должен одновременно подгружать исходные данные и выгружать зашифрованные данные, то для обеспечения этих процессов необходимо использовать два канала ПДП. При этом один канал ПДП должен быть настроен на область памяти (или на устройство) с исходными данными, а второй канал – на область памяти (или на устройство), куда следует разместить зашифрованные данные.

Очевидно, что для того, чтобы процессы загрузки и выгрузки данных шли с максимальной интенсивностью, необходимо оба канала ПДП и сам криптомодуль настроить на обмен данными максимально широкими словами (в нашем случае максимальная разрядность слова составляет 64 разряда). Нет необходимости при этом объяснять, что частота обменов по шине (а следовательно, и загрузка шины) будет пропорционально меньше, чем при использовании для обменов данными слов с меньшей разрядностью.

Эксперименты показали, что производительность криптомодуля не зависит от того, словами какой разрядности будет производиться обмен данными между памятью и криптомодулем – 16-разрядными или 64-разрядными. Загрузка и выгрузка данных в любом случае осуществляется быстрее, чем криптомодуль успевает эти данные обработать. Однако, в случае использования 64-разрядных слов для обмена данными по шине, количество обменов данными при обработке блока одного и

того-же размера будет в 4 раза меньше, а значит и нагрузка на шину со стороны криптомодуля будет в 4 раза меньше, чем в случае использования для обменов данными 16-разрядных слов. Следовательно, шина в нашем случае в большей степени остаётся свободной для обслуживания других параллельно выполняемых процессов, что прямым образом сказывается на общей производительности системы.

Операции настройки криптомодуля и каналов ПДП осуществляются при этом в обычном процессорном режиме, но их вклад в оценку общей производительности системы не является существенным.

Следует отметить, что наличие блока криптообработки данных является практически непременным атрибутом современных систем передачи данных. В некоторых случаях допускается программная реализация алгоритмов криптообработки, однако такой подход влечёт за собой существенное замедление процесса обработки данных и дополнительные энергетические затраты. По существу, в современных микропроцессорных системах аппаратный блок криптообработки данных является узкоспециализированным сопроцессором, позволяющим резко повысить скорость обработки данных и существенным образом повлиять на общую производительность системы.

VII. Основные выводы

В предложенном для рассмотрения варианте реализации СнК на базе 16-разрядного процессорного ядра с использованием 64-разрядной шины, предпринята попытка объединить достоинства 16-разрядных процессорных систем и многоразрядных шинных структур.

Как уже было отмечено ранее, основным достоинством выбранного 16-разрядного процессорного ядра является компактность исполняемого кода, и как следствие, невысокие требования к объёму памяти, необходимой для исполнения программ. Это важное свойство позволяет создавать компактные однокристальные системы со встроенной статической памятью и с небольшим количеством внешних выводов, необходимых только для подключения внешней периферии.

Для достижения нужных показателей производительности, помимо оптимизации структуры процессорного ядра, в состав СнК включена 64-разрядная шина, с более высокой (в несколько раз) пропускной способностью. В сочетании с оптимизированным под эту шину контроллером ПДП, обеспечивающим обмены данными 64-разрядными словами, мы получаем существенный прирост в общей производительности системы.

Дополнительным важным способом повышения эффективности вычислительной системы является специализация путём включения в её состав аппаратных вычислительных или управляющих блоков. Такого рода блоки можно рассматривать как специализированные сопроцессоры, которые могут решать свою задачу параллельно основному процессору под его контролем и управлением. Такой подход даёт выигрыш как в плане быстродействия, так и в плане энергопотребления, поскольку существенно снижает нагрузку на основной процессор и разделяет общую задачу на несколько параллельных вычислительных процессов.

Таким образом, в результате всех предпринятых мер (оптимизация процессорного ядра путём изменения структуры и введения глубокой конвейерной обработки команд, а также путём разнесения потоков команд и данных на два разных интерфейса, использование широкой полностью когерентной 64-разрядной шины и адаптированного к ней контроллера прямого доступа в память, применение специализированных аппаратных блоков для решения конкретных вычислительных задач) удалось получить компактную систему на кристалле с высокой производительностью, ориентированную для применения в составе систем передачи данных и управления. Приблизительные подсчёты показывают, что по сравнению с оригинальным микроконтроллером i80186, с учётом более совершенной технологии производства, общая производительность представленной СнК возросла не менее чем на два порядка.

Литература

[1] Leokhin Y., Panfilov P. A Study of Cloud/IX Operating System for the ARM-based Data Center Server Platform // Procedia Engineering. 2015. Vol. 100. P. 1696-1705.

[2] В.В. Ерохин Элементная база систем на кристалле: процессоры, Нано- и микросистемная техника. 2005. № 1. С. 35-44.

[3] Intel Corporation. 80C186EC/80C188EC microprocessor user's manual. Intel Corp., 1995. 515 p.

[5] N.B. Bahadure, Microprocessors: The 8086/8088, 80186/80286, 80386/80486 and The Pentium Family. PHI Learning Pvt. Ltd., New Delhi, 2010.

[5] А.В. Вишнеков, В.В. Ерохин, Е.М. Иванова Выбор IP-блока при разработке системы на кристалле // Нано- и микросистемная техника. 2016. Т. 18. № 12. С. 773-782.

Designing a High-Performance SoC Based on a 16-bit Processor Core

V.Ya. Arkhipkin[1], M.I. Dyabin[2], V.V. Erokhin[3], Yu.L. Leokhin[4]

[1]"Accord" Ltd., director general, Moscow, info@akkordsystems.ru

[2]"Kaskad" Ltd., leading engineer, Moscow, dyabin@mail.ru

[3]JSC Progress MRI, chief of the communication equipment design division, Moscow, vladimir.v.erokhin@gmail.com

[4]Moscow Technical University of Communications and Informatics, prorector of science, Moscow, y.l.leokhin@mtuci.ru

***Abstract* — The article describes one of the ways to build a high-performance system on a chip for embedded applications based on a 16-bit processor core and a wide 64-bit bus. The rationale for the correctness of the chosen approach in terms of using a 16-bit i80186-compatible processor core as the basis for building the system is given. The main merits of the chosen 16-bit architecture are listed.**

The article describes one of the ways to build a high-performance system on a chip for embedded applications based on a 16-bit processor core and a wide 64-bit bus. The rationale for the correctness of the chosen approach in terms of using a 16-bit i80186-compatible processor core as the basis for building the system is given. The contribution of the 64-bit bus and direct memory access channels adapted to this bus to improving system performance when working with large data streams is noted. An example of organizing the operation of a high-performance device (cryptomodule) as part of a system with a demonstration of the advantages of using a wide bus when processing large data arrays is given.

A usage of modern processor independent AMBA AXI bus in the SoC as well as using a command cache and a data cache in the system are considered in this paper. It is noted that the 64-bit bus and a DMA controller adapted to this bus, contribute to increasing the system's performance while processing huge data streams. The authors consider two variants of organizing priorities between DMA channels when more than one channel is started to work simultaneously.

This paper presents using a coherence controller to provide data integrity while accessing the same data simultaneously (the procedures of writing and reading) when this data is placed in different information storages (the main RAM, the command cache, the data cache), of different active devices (the processor, the DMA channels).

An example of organizing the operation of a high-performance device (namely, a cryptographic device) as part of a system with a demonstration of the advantages of using a wide bus when processing large data arrays is given. The authors describe the process by which the crypto device processes data using DMA channels for loading source data into the device and saving result to memory.

In conclusion, the paper justifies the correctness of the chosen approach for designing high-performance systems for embedded usage. The authors give a comparative evaluation of increasing the performance of the presented SoC relatively to the original microcontroller.

***Keywords* — bus, pipeline, processor core, DMA controller.**

REFERENCES

[1] Leokhin Y., Panfilov P. A Study of Cloud/IX Operating System for the ARM-based Data Center Server Platform // Procedia Engineering. 2015. Vol. 100. P. 1696-1705.

[2] V.V. Erokhin Elementnaya baza sistem na kristalle: processory // Nano- i mikrosistemnaya tekhnika. 2005. № 1. S. 35-44 (in Russian).

[3] Intel Corporation. 80C186EC/80C188EC microprocessor user's manual. Intel Corp., 1995. 515 p.

[4] Bahadure N.B. Microprocessors: The 8086/8088, 80186/80286, 80386/80486 and The Pentium Family. PHI Learning Pvt. Ltd., New Delhi, 2010.

[5] A.V. Vishenkov, V.V. Erokhin, E.M. Ivanova Vybor IP-bloka pri razrabotke sistemy na kristalle // Nano- i mikrosistemnaya tekhnika. 2016. T. 18. № 12. S. 773-782 (in Russian).