

# НОВЕЙШИЕ ПРОЦЕССОРНЫЕ ЯДРА SYNOPSYS ДЛЯ ВСТРАИВАЕМЫХ ПРИЛОЖЕНИЙ – БОЛЕЕ ВЫСОКАЯ ПРОИЗВОДИТЕЛЬНОСТЬ И ГИБКОСТЬ

ЧАСТЬ 2

Т.Халфхилл

В первой части статьи, опубликованной в журнале "ЭЛЕКТРОНИКА: НТБ" № 7 за 2015 год, мы начали знакомить читателя с новыми лицензируемыми процессорными ядрами семейства ARC HS компании Synopsys, которые позволяют создавать высокоэффективные решения для встраиваемых приложений. В частности, рассмотрели возможности конфигурирования и расширения системы команд, особенности работы усовершенствованного 10-ступенчатого конвейера и других функций, повышающих производительность процессорного ядра. Кроме того, мы обсудили новые методы и инструменты разработки, верификации и отладки проектов на базе ARC HS. Во второй части статьи речь пойдет о технологии симметричной мультипроцессорной обработки, опциональной L2-кэш-памяти, блоке управления памятью, режимах энергосбережения и системе управления питанием. В заключение сравним ARC HS38 с основными конкурентами и остановимся на ключевых особенностях этих процессорных ядер, которые следует учитывать разработчику встраиваемой системы.

## ТЕХНОЛОГИЯ СИММЕТРИЧНОЙ МУЛЬТИПРОЦЕССОРНОЙ ОБРАБОТКИ

Традиционным способом увеличения производительности процессорной системы является повышение тактовой частоты ядра, однако это ведет к росту потребляемой кристаллом мощности. Как было показано в первой части статьи, в процессорных ядрах ARC HS38 для улучшения производительности применены альтернативные подходы – возможность расширения системы команд ARCv2 и конфигурирование ядер ARC, что позволяет обойтись без увеличения тактовой частоты.

В некоторых приложениях дополнительные преимущества по производительности дает использование многоядерной архитектуры, особенно, если программное обеспечение способно эффективно распределять рабочие задания между ядрами. На основе ядер ARC HS38 и ARC HS36 можно реализовать двух- и четырехъядерные конфигурации с использованием эффективной технологии симметричной мультипроцессорной обработки с когерентной кэш-памятью при меньших по сравнению с конкурирующими решениями затратами. Заметим, что хотя серия ARC HS34 также

включает в себя двух- и четырехъядерные конфигурации, она не поддерживает технологию симметричной мультипроцессорной обработки.

Поддержка когерентности L1-кэш-памяти является важнейшим условием симметричной мультипроцессорной обработки. Когда два и более центральных процессора обращаются к одному и тому же участку памяти, необходим механизм, который предохраняет выбираемые данные от их независимой модификации. Данные в кэш-памяти должны оставаться когерентными, то есть взаимосогласованными с данными в других кэшах. Поддержка такой когерентности программным способом трудоемка и требует траты множества тактовых циклов, поэтому процессоры с когерентной кэш-памятью реализуют этот механизм аппаратным путем. В сериях ARC HS38 и ARC HS36 используется широко известный метод, который называется слежением (snooping), который позволяет контролировать операции чтения и записи L1-кэш-памяти и поддерживает когерентность данных этой кэш-памяти с данными в других кэшах.

Слежение в процессорах ARC HS осуществляет усовершенствованный блок поддержки когерентности кэш-памяти. Каждое ядро процессора имеет трехканальный интерфейс слежения: первый канал передает команды слежения от блока когерентности к ядру, второй – возвращает отклик ядра в блок когерентности,

а третий – передает данные из L1-кэш-памяти ядра в блок когерентности, который распределяет их между L1-кэш-памятью других ядер (рис.1).

Для управления передачей данных между кэш-памятью блок когерентности использует специальный протокол MOESI (Modified, Owned, Exclusive, Shared, Invalid), который отражает пять возможных состояний кэшируемых данных. Например, когда ядро записывает результат выполнения команды в L1-кэш-память данных, оно обозначает эту строку кэш-памяти как модифицированную, и эта строка становится единственной достоверной копией данных. Блок поддержки когерентности должен передать эти данные другим L1-кэшам данных до того, как другие ядра смогут начать их использовать.

Аналогичным образом опциональный блок когерентности ввода/вывода обеспечивает согласованный с L1-кэш-памятью трафик данных. Когда устройство ввода/вывода модифицирует данные в L1-кэш-памяти одного ядра, этот блок обновляет также и другие L1-кэши. Важным моментом является то, что разработчикам прикладного ПО не нужно беспокоиться об этих деталях, поскольку аппаратные средства поддержки когерентности автоматически отслеживают использование системных ресурсов.

Еще одной важной функцией технологии симметричной мультипроцессорной обработки является

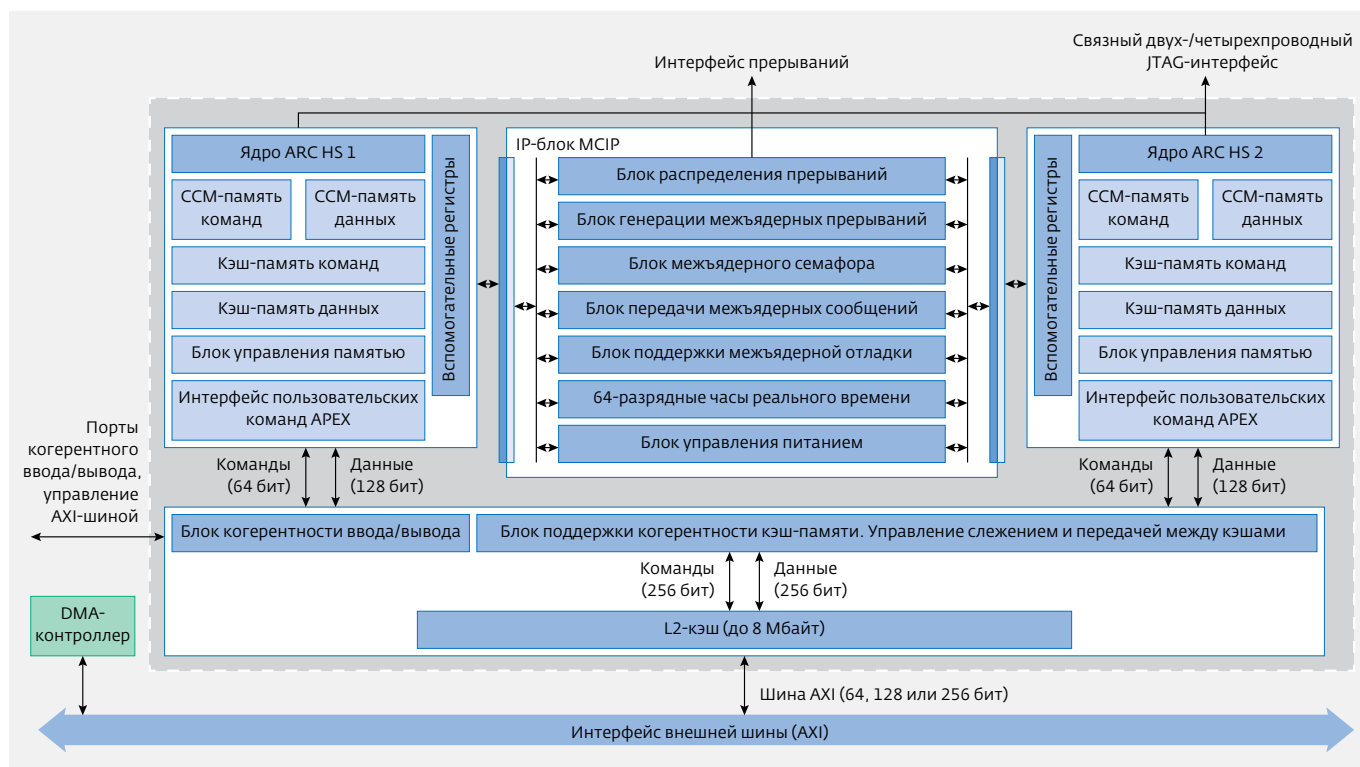


Рис.1. Двухъядерный процессорный кластер на базе ARC HS38

межъядерная коммуникация. Для того чтобы обеспечить ядрам обмен сообщениями в многоядерной системе, процессорный кластер разделяет центральное СОЗУ между всеми ядрами. Если два и более ядра одновременно предпринимают попытку получить доступ к этой памяти, специальная схема разрешения конфликтов реализует циклическое обслуживание каждого ядра.

Подобным же образом межъядерный обработчик прерываний позволяет любому ядру передать прерывание другому ядру, например, когда процесс, запущенный на одном ядре, генерирует ошибку, которую должен обработать процесс на другом ядре.

Для синхронизации множества процессов имеется еще один централизованный ресурс – 64-разрядные часы реального времени. Эта функция позволяет устранить необходимость в избыточной реализации тактовых сигналов для каждого процессорного ядра. Поддержку этой синхронизации и управление доступом каждого из ядер к совместно используемым ресурсам процессорного кластера осуществляют аппаратные семафоры. Разработчики в полной мере оценят централизованный механизм, позволяющий внешнему отладчику или системе разработки останавливать, запускать или переустанавливать любое ядро или комбинацию ядер в кластере.

Кроме того, совместную работу нескольких ядер помогает осуществлять ряд дополнительных функций. Например, имеется блок распределения прерываний, который автоматически направляет внешние прерывания соответствующему ядру кластера. В наборе команд ARCv2 предусмотрен ряд новых команд, в том числе DSYNC (для синхронизации памяти данных) и DMB (для определения границы памяти данных). Команды LLOCK и SCOND позволяют процессорам выполнять атомарные операции над памятью, одновременно обеспечивая когерентность между всеми ядрами. Средства разработки ARC MetaWare поддерживают эти новые команды в компиляторе C/C++, ассемблере и отладчике.

## L2-КЭШ-ПАМЯТЬ

Когерентные L1-кэши могут работать во взаимодействии с опциональной L2-кэш-памятью. Эта кэш-память является новой для семейства ARC HS и доступна в качестве отдельно лицензируемой опции для ядер ARC HS38 и ARC HS36. L2-кэш-память играет значительно большую роль, чем просто блок СОЗУ, соединенный с центральным процессором. Ряд ее функций обеспечивает высокую производительность при минимальном энергопотреблении.

Во-первых, следует отметить, что все центральные процессоры в многоядерном кластере совместно

используют одну L2-кэш-память, объем которой конфигурируется пользователем и может достигать 8 Мбайт. Эта кэш-память работает на тактовой частоте процессорного ядра (до 1,6 ГГц при 28-нм процессе НРМ). Кроме того, чтобы не замедлять периферийную шину AXI, L2-кэш-память подключена к тыловой шине (backside bus) каждого ядра. Фактически, эта кэш-память имеет отдельную 64-разрядную шину команд и 128-разрядную шину данных (так называемая гарвардская архитектура). Со стороны входа, подключенного к периферийной шине AXI, L2-кэш-память имеет конфигурируемый интерфейс, который может иметь разрядность 64, 128 или 256 бит. Это обеспечивает одинаковую скорость работы L2-кэш-памяти и ядер, не создавая при этом помех трафику данных по шине AXI между процессорными ядрами и L2-кэшем.

В дополнение к конфигурированию тактовой частоты, объема памяти и AXI-интерфейсов разработчики кристаллов могут изменять и другие свойства L2-кэш-памяти. Строки кэш-памяти могут иметь длину 64 или 128 байт, а ассоциативность кэша может достигать 16-ти каналов. Для ответственных приложений L2-кэш-память может поддерживать коды с коррекцией ошибок (ECC). Эта кэш-память также поддерживает AXI-протоколы, в том числе протоколы read-through, write-through, read-no-allocate и write-no-allocate. Протоколы без распределения (no-allocate) полезны в том случае, когда когерентный контроллер прямого доступа к памяти (DMA-контроллер) должен хранить данные в памяти без их кэширования для последующего использования.

При нормальных условиях работы L2-кэш-память замещает данные с наиболее давним использованием (алгоритм LRU). В каждой 128-байтной строке кэш-памяти имеются четыре бита изменений, идентифицирующие данные, то есть имеется один модифицируемый бит (dirty bit) на каждый 32-байтный фрагмент данных. Когда возникает необходимость заменить строку кэш-памяти, требуется перезаписать только изменяемые фрагменты данных, что позволяет не снижать полосу пропускания канала ввода / вывода. Чтобы предотвратить замещение важных данных из кэш-памяти, которые заведомо будут использоваться позднее, программисты могут заблокировать определенные строки кэш-памяти. Например, программа может зафиксировать в кэш-памяти необходимый обработчик прерываний, обеспечивая быстрый отклик на предполагаемое событие.

Программисты могут также обеспечивать сброс или аннулирование определенных строк или более крупных фрагментов кэш-памяти, а также изменять число строк кэш-памяти, которые выделены для повторного заполнения после непопаданий в кэш-память. Если при

запросе памяти пропускается L1-кэш-память, то центральный процессор может начать запрос данных из L2-кэш-памяти, одновременно отслеживая L1-кэши других процессорных ядер. Если при запросе пропускаются также и другие L1-кэши, процессор вызывает данные из L2-кэш-памяти. Когда исполняется запрос данных из другой L1-кэш-памяти, запрос из L2-кэш-памяти прерывается без последствий. Функции контроля за производительностью позволяют программам изменять режимы работы L2-кэш-памяти во время выполнения задания.

С целью энергосбережения L2-кэш-память может быстро переходить в режим ожидания (sleep mode), в котором сохраняется содержимое СОЗУ, а энергопотребление сокращается примерно на 90%. Еще один режим энергосбережения – режим холостого хода (idle) – более быстрый с точки зрения входа/выхода из него, но он не является столь же эффективным с точки зрения экономии энергии. Как правило, наибольшую мощность можно сэкономить в режиме полного отключения (shut-down mode), но при этом данные в кэш-памяти будут потеряны.

Когда разработчики задействуют в своих проектах L2-кэш-память, они могут выбрать СОЗУ большей емкости, чтобы снизить энергопотребление и сократить площадь кристалла, хотя производительность

при этом немного падает. В системе управления кэш-памятью реализовано четыре уровня стробирования тактового сигнала – на один уровень больше, чем в процессорном ядре.

Компания Synopsys разработала L2-кэш-память с расчетом на появление в будущем новых продуктов. Эта кэш-память может совместно использоваться восемью процессорными ядрами, то есть в два раза большим числом ядер, чем доступно в настоящее время. Кроме того, L2-кэш-память можно продублировать в качестве L3-кэш-памяти.

Заметим, что проекты на базе процессора ARC HS не требуют задействовать L2-кэш-память. Пользователи могут не включать их в проект или использовать вместо них сверхоперативную память (scratchpad memory), которая имеет до 8 Мбайт СОЗУ на процессорный кластер. Эта память тесно связана с процессорными ядрами и может реагировать на запросы чтения/записи примерно за шесть тактовых циклов. (Не следует путать сверхоперативную память с тесно связанной памятью меньшего объема, которую разработчики могут использовать вместо L1-кэш-памяти внутри процессорного ядра). Сверхоперативная память может также служить в качестве программно-управляемой L2-кэш-памяти. На самом деле в приложениях, требующих большого

объема встроенной памяти, один процессорный кластер может содержать как программно-управляемую L2-кэш-память, так и программно-управляемую сверхоперативную память.

## БЛОК УПРАВЛЕНИЯ ПАМЯТЬЮ

Еще одной ключевой особенностью процессоров ARC HS38 является усовершенствованный блок управления памятью (Memory-Management Unit – MMU), который дает возможность этим ядрам работать под управлением встраиваемых ОС с поддержкой симметричной мультипроцессорной обработки и виртуальной памяти.

В этом блоке конфигурируемое пространство физических адресов расширено до 40 бит, что достаточно для адресации 1Тбайт основной памяти. Для сравнения, в MMU процессоров ARC 700 пространство физических адресов ограничено 32-мя битами, что соответствует 4 Гбайтам памяти. Поскольку эта функция является конфигурируемой, пользователи могут по своему усмотрению выбирать разрядность адресации памяти. Следует отметить, что для большинства встраиваемых приложений разрядность в 40 бит будет вполне достаточной еще в течение многих лет.

Кроме того, новый MMU поддерживает страницы памяти изменяемого размера. Фактически этот блок может одновременно работать с двумя разными размерами страниц, в то время как MMU в процессоре ARC 700 работает лишь с одним фиксированным размером – 8 Кбайт (величина по умолчанию для Linux). Процессоры ARC HS38 одновременно поддерживают страницы памяти стандартного размера (4 Кбайт, 8 Кбайт или 16 Кбайт), а также очень крупные страницы памяти (4 Мбайт, 8 Мбайт или 16 Мбайт). Более крупные размеры страниц снижают число пропущенных ссылок в буфере ассоциативной трансляции (TLB).

Первичный буфер ассоциативной трансляции (TLB) нового MMU имеет 1024 элемента записи и является четырехканальным модульно-ассоциативным буфером. В отличие от этого, первичный TLB процессора ARC 700 имеет 256 элементов записи и два канала ассоциативности – хотя он уже является достаточно крупным для ядра встраиваемого процессора. Благодаря еще большему размеру TLB, у ядра ARC HS38 имеется возможность трансляции виртуальных адресов в физические адреса памяти, не тратя значительное число тактовых циклов на просмотр таблицы переадресации страниц в основной памяти. Кроме того, новый TLB имеет такие же полностью ассоциативные микро-TLB, как и в процессоре ARC 700: четырехэлементный iTLB для команд и восьмиэлементный dTLB для данных.

Новый TLB имеет прогнозирующее устройство (предиктор), которое способно дополнительно ускорить

трансляцию адресов. Оно нужно только тогда, когда объем данных канала кэш-памяти превышает размер страницы MMU – например, когда 32-Кбайтная кэш-память данных является двухканальной модульно-ассоциативной кэш-памятью, а размер страницы составляет 8 Кбайт. В таких случаях только один бит виртуального адреса, который индексирует кэш-память, может измениться в процессе трансляции из виртуального в физический адрес. Вместо того, чтобы ожидать завершения трансляции перед выборкой кэш-памяти, TLB может контролировать последние трансляции и прогнозировать с 90%-ной точностью, будет ли этот бит "нулем" или "единицей", и тем самым предполагать значение физического адреса. (Кэш-память данных и индексируется, и помечается физически, в то время как кэш-память команд индексируется виртуально, а помечается физически). Используя такое прогнозирование, центральный процессор может выбирать данные, в то время как TLB одновременно выполняет трансляцию адреса. Если прогноз верен, то процессорное ядро переходит к вызову данных. Если прогноз не верен, то процессорное ядро останавливает вызов данных и переходит к корректному адресу памяти в TLB.

Множественные процессы, запущенные на ядрах ARC HS38 могут совместно использовать единое распределение данных в TLB. На ОС Linux, например, идентификатор пространства адресов (address-space identifier – ASID) блока управления памятью допускает совместное использование TLB максимально 64-мя библиотеками без сброса и повторного заполнения буфера. Это в два раза больше числа разделяемых библиотек, допускаемых в ядре ARC 700.

Чтобы обеспечить более эффективное программное управление TLB, новые команды набора ARCV2 выполняют обычные операции за меньшее число тактовых циклов по сравнению с ядром ARC 700. Такими новыми командами являются: TLBInsertEntry, TLBDeleteEntry и TLBProbeAndGetId. Каждая из них выполняет полную операцию над TLB, тогда как раньше для этого требовалось несколько команд.

## УСОВЕРШЕНСТВОВАННОЕ КОНТЕКСТНОЕ ПЕРЕКЛЮЧЕНИЕ

Процессорные ядра ARC поддерживают несколько банков регистров для быстрого переключения контекста. За один тактовый цикл процессор может изменить указатель на дублирующий блок регистров, который сохраняет полное состояние другого процесса. Такое переключение банков происходит намного быстрее, чем копирование регистров в память и последующая их перезагрузка. В то время как процессор ARC 700 может иметь лишь два таких банка, пользователи могут

конфигурировать в ядре ARC HS38 до восьми банков. Ядра ARC HS34 и HS36 в настоящее время также получили эту возможность.

## НЕСКОЛЬКО ДОМЕНОВ ПИТАНИЯ

В многоядерном проекте на базе ARC HS (два или четыре ядра) каждое процессорное ядро находится в своем домене питания. Для снижения энергопотребления, когда в системе не требуется максимальная производительность, отдельные ядра могут находиться в режиме ожидания (режиме сна) или режиме отключения. Этими режимами управляет централизованный блок управления питанием, который обслуживает все ядра (рис.2).

Блок управления питанием может также регулировать (повышать и понижать) напряжение питания ядра и тактовую частоту ядра, чтобы изменять производительность процессора при изменении рабочей нагрузки. Такое динамическое масштабирование напряжения/частоты может существенно снизить потребляемую мощность, которая находится в линейной зависимости от тактовой частоты и в квадратичной

зависимости от напряжения. Как правило, эти параметры регулируют ступенчато. Блок управления питанием изменяет тактовую частоту на небольшую величину до тех пор, пока не будет достигнута определенная пороговая величина. Затем он начинает изменять напряжение питания, от которого в наибольшей степени зависит энергопотребление.

Программисты могут контролировать эти параметры с помощью блока управления питанием, настраивая систему под определенную нагрузку. Однако, как и в большинстве процессорных кластеров, поддерживающих динамическое масштабирование напряжения/частоты, все процессорные ядра должны работать синхронно. Отдельные ядра могут переходить в дежурный режим или режим отключения, но не могут работать при напряжениях питания или на тактовых частотах, которые отличаются от других ядер в одном и том же кластере.

Кроме того, отдельные процессорные ядра и другие элементы многоядерного кластера имеют несколько собственных доменов питания. Каждое

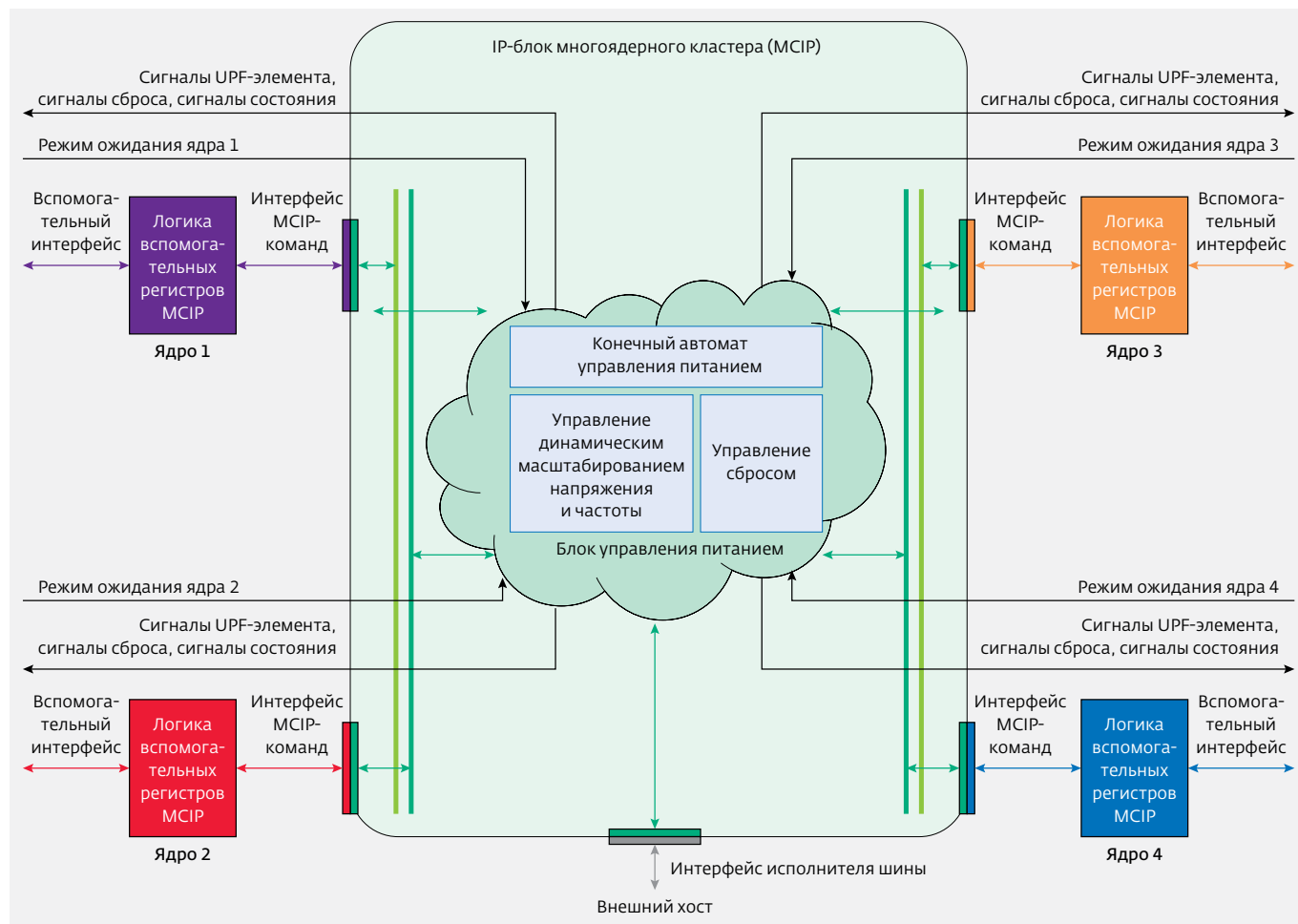
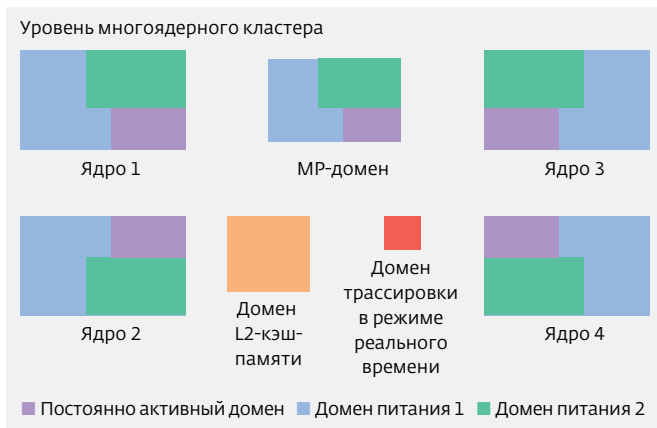


Рис.2. Блок управления питанием процессоров семейства ARC HS



**Рис.3.** Домены питания в четырехъядерном кластере на базе ARC HS38

ядро разделено на три домена (рис.3): первый домен всегда поддерживает в активном состоянии небольшую часть критически важной логики; второй домен обеспечивает питание логики ядра и скрытых для программиста регистров; третий домен обеспечивает питание видимых для программиста регистров и СОЗУ L2-кэш-памяти. Постоянно активный домен содержит логику, которая должна оставаться доступной для откликов на команды активизации. Второй домен может переходить в режим ожидания (режим сна), сохраняя при этом состояние или прекращая работу после сохранения состояния. Третий домен может хранить состояние, находясь в режиме ожидания при низком напряжении, что существенно снижает ток утечки.

Другие элементы многоядерного кластера также расположены в своих доменах питания и могут переходить в состояние пониженного энергопотребления. Эти элементы включают в себя блок поддержки когерентности кэш-памяти и блок когерентности ввода/вывода. Например, если все ядра в кластере, кроме одного, отключены, некоторые из этих мультипроцессорных функций становятся не нужными и могут потреблять меньший ток.

## СРАВНЕНИЕ С КОНКУРЕНТАМИ

Несмотря на то что процессорные ядра ARC HS38 демонстрируют производительность на уровне прикладных процессоров среднего класса, компания Synopsys не стремится на рынок устройств, программно совместимых с процессорами ARM или MIPS. Семейство ARC HS конкурирует в сегменте высокопроизводительных встраиваемых систем, где нужно обеспечить эффективную обработку специфических, четко определенных задач. В частности, процессор ARC HS38 предназначен для приложений, работающих

под управлением Linux или требующих симметричной мультипроцессорной обработки.

Сравним процессорное ядро ARC HS38 с некоторыми похожими процессорными ядрами от ARM, Imagination Technologies и Cadence (см. табл.). Из пяти сравниваемых ядер только в Diamond Standard 233L отсутствует аппаратная поддержка когерентности кэш-памяти. Многоядерные проекты на базе ядер Diamond Standard 233L осуществимы практически, но когерентностью кэш-памяти необходимо управлять программным способом (некоторые сборки Linux поддерживают программно-управляемую когерентность). Ядро Diamond Standard 233L содержит блок управления памятью, поэтому на нем можно запускать полнофункциональную версию Linux с поддержкой виртуальной памяти. Поскольку 233L – это достаточно простой процессор со сравнительно коротким пятиступенчатым конвейером, он занимает меньшую по сравнению с другими ядрами площадь кремния. Но среди сравниваемых процессоров его рабочая тактовая частота минимальна при любом используемом технологическом процессе.

Два ядра от ARM в данном сравнении входят в семейство высокопроизводительных процессоров Cortex-A, однако не являются самыми быстродействующими его членами. В многоядерных системах ядро Cortex-A7 может работать совместно с более производительным ядром Cortex-A15. При малой нагрузке более крупное ядро Cortex-A15 может отключаться, а менее производительное ядро Cortex-A7 поддерживает работу системы. Однако Cortex-A7 может работать как автономное ядро в одноядерных проектах или работать совместно с другими процессорами Cortex-A7 в системе с когерентной кэш-памятью. Хотя в своей основе конфигурация ядра Cortex-A7 подобна Synopsys ARC HS38, его производительность по CoreMark хуже, а объем L2-кэш-памяти ограничен одним мегабайтом (по сравнению с 8 Мбайт в ARC HS38). Меньший размер L2-кэш-памяти достаточен для обычных двухъядерных конфигураций, но ограничивает производительность в четырехъядерном кластере. Однако, как и ARC HS38, Cortex-A7 поддерживает 40-разрядную адресацию физической памяти.

Cortex-A9 – более старый процессор от ARM, который стал первым ядром на базе ARMv7A с поддержкой когерентности кэш-памяти. Он выделяется среди других сравниваемых ядер тем, что это единственный процессор со суперскалярной архитектурой и двухпоточковым конвейером. Он также поддерживает переупорядочивание команд (moderate instruction reordering), спекулятивное выполнение операций и развитое предсказание переходов. В результате, он может выполнять больше инструкций за тактовый

Сравнение ядра ARC HS38 с четырьмя конкурирующими процессорными ядрами

| Процессорное ядро                         | ARC HS38 (Synopsys)                                    | Cortex-A7 (ARM)                                 | Cortex-A9 (ARM)                               | MIPS interAptiv (Imagination)                        | Diamond 233L (Cadence)                                   |
|---|--|---|---|--|--|
| Система команд                            | ARCV2  | ARMv7A  | ARMv7A  | MIPS32-R5  | Tensilica Xtensa   |
| Симметричная мультипроцессорная обработка | 2-4 ядра, аппаратная поддержка когерентности           | 2-4 ядра, аппаратная поддержка когерентности    | 2-4 ядра, аппаратная поддержка когерентности  | 2-4 ядра, аппаратная поддержка когерентности         | Только программная поддержка когерентности               |
| Конвейер                                  | 10 ступеней  | 8 ступеней                                      | 9-10 ступеней                                 | 9 ступеней   | 5 ступеней   |
| Скорость выдачи команд                    | Один за цикл   | Один за цикл                                    | Два за цикл                                   | Один за цикл   | Один за цикл   |
| Производительность, CoreMarks/МГц         | 3,5  | 2,6   | 3,9   | 3,2 (2 потока)                                       | Нет данных   |
| Производительность, DMIPS/МГц             | 1,94   | 1,9   | 2,5   | 1,57   | 1,38   |
| Максимальная частота процессора           | 2,2 ГГц (тип.)   | 1,6 ГГц (тип.)                                  | 2,0 ГГц (тип.)                                | 2,0 ГГц (тип.)                                       | 1,0 ГГц (тип.)   |
| Тесно связанная память                    | До 16 Мбайт (память команд) + 16 Мбайт (память данных) | Нет   | Нет   | До 1 Мбайт (память команд) + 1 Мбайт (память данных) | До 128 Кбайт (память команд) + 128 Кбайт (память данных) |
| L1-кэш-память команд и данных             | До 64 Кбайт каждый                                     | До 64 Кбайт каждый                              | До 64 Кбайт каждый                            | До 64 Кбайт каждый                                   | 8 Кбайт каждый   |
| L2-кэш-память                             | До 8 Мбайт   | До 1 Мбайт                                      | До 8 Мбайт                                    | До 8 Мбайт   | Нет  |
| Когерентность ввода/вывода (Accel)        | Есть   | Есть  | Есть  | Есть   | Нет  |
| Модуль операций с плавающей запятой       | Одинарная/двойная точность                             | Одинарная/двойная точность                      | Одинарная/двойная точность                    | Одинарная/двойная точность                           | Нет  |
| MAC-операции за цикл                      | 1×32×32 бит или 2×32×16 бит или 4×16×16 бит            | 1×32×32 бит или 2×16×16 бит                     | 1×32×32 бит или 2×16×16 бит                   | 2×16×16 бит  | 1×16×16 бит  |
| Дополнительные функции                    | DSP, ISA-расширения, 64-разр. загрузка/хранение, ECC   | Neon SIMD-расширения, 40-разр. адреса памяти    | Neon SIMD-расширения, ускорение Java-программ | Двухканальная поточность, виртуализация              | Предварительно конфигурируемое ядро Xtensa               |
| Интерфейс СнК                             | 2×32- или 64-разр. шины AXI, ANV-Lite, BVCI            | 128-разр. шина AMBA4 (ACE), APB (ведомое устр.) | 2×64-разр. шина AMBA3 AXI (ведущее устр.)     | 64-разр. OCP   | Конфигурируемые шины ANV-Lite, AXI, XLMI                 |



цикл, чем другие процессоры этой группы, что отражается в более высоких показателях производительности по CoreMark и Dhrystone. Поскольку неизбежным следствием таких характеристик является более высокая сложность, ядро Cortex-A9 занимает большую площадь кремния на кристалле и потребляет больше мощности, чем другие ядра, изготовленные с использованием такой же технологии. Заметим, что компания ARM выпустила за эти годы несколько модификаций ядра Cortex-A9, а также предлагает ряд аппаратных ядер, оптимизированных под определенные технологические процессы, поэтому их производительность может меняться в широких пределах. Более современным ядром от ARM является 32-разрядный процессор Cortex-A12 на базе ARMv7A. Он использует такую же технологию симметричной мультипроцессорной обработки, что и Cortex-A9, а также обеспечивает аппаратную поддержку виртуализации и 40-разрядную адресацию физической памяти.

Ядро MIPS interAptiv от Imagination Technologies конкурирует с ARC HS38 в некоторых приложениях. Среди приведенных в таблице процессорных ядер interAptiv выделяется как единственное ядро с опциональной многопоточностью. Двухъядерный двухпоточковый кластер может обрабатывать суммарно восемь потоков с поддержкой когерентности кэш-памяти. Кроме того, восьмиступенчатый конвейер команд ядра interAptiv может обеспечить высокую тактовую частоту и демонстрирует хорошую однопоточковую производительность. InterAptiv построен на основе новейшего набора команд MIPS32-R5, поэтому в этом процессоре реализована аппаратная поддержка виртуализации и некоторые новые расширения SIMD-команд. Однако все эти функции повышают сложность кристалла. Например, когда задействуется двухпоточковый режим, площадь базового ядра увеличивается примерно на 30%. Оценки показывают, что ядро interAptiv занимает большую площадь кремния и потребляет большую мощность, чем ARC HS38 аналогичной конфигурации. Для однопоточковых приложений и, в особенности, для задач с интенсивной DSP-обработкой большую эффективность покажет ядро ARC HS38.

Благодаря более глубокому 10-ступенчатому конвейеру команд ядро ARC HS38 обладает достаточным запасом по тактовой частоте даже при низких напряжениях. Более сложные высокопроизводительные процессоры расходуют больше мощности и насчитывают большее число транзисторов. Ядра ARC HS, которые используют меньшее число транзисторов и обладают меньшим энергопотреблением, тем не менее демонстрируют высокую производительность и гибкость, благодаря чему разработчики СМК

могут в широких пределах подстраивать эти процессоры под свои требования.

Так или иначе, опытные разработчики кристаллов должны быть внимательны к выбору тактовой частоты работы процессора. Достижение максимальной номинальной рабочей частоты синтезируемого ядра, даже при использовании проверенного RTL-описания проекта, обычно требует дополнительных усилий по оптимизации топологии и решения некоторых схемотехнических проблем, таких как минимизация динамического тока потребления и уровня шума в сигналах. По этой причине лишь в редких случаях можно достичь номинального быстродействия программного ядра. Кроме того, в большинстве случаев занимаемую площадь и потребляемую мощность оценивают по наиболее оптимистичному варианту. Даже когда вендоры предоставляют информацию о характеристиках кристаллов, изготовленных по одним и тем же технологическим нормам, сравнение не является точным, так как данные не отражают истинного положения дел (особенно в отношении энергопотребления и площади кристалла) из-за конкурентной борьбы на рынке. Также надо иметь в виду, что некоторые поставщики раскрывают только часть конфигураций процессора. Следовательно, читатель не должен придавать слишком большого значения несущественным различиям в публикуемых поставщиками данных. Разработчикам СМК всегда следует удостовериться, что при измерении мощности и площади ядер были включены все необходимые RTL-блоки.

Кроме того, результаты тестирования процессоров также зависят от многих факторов. Тест CoreMark консорциума EEMBC лучше устаревшего теста Dhrystone, но самые последние данные EEMBC отражают результаты оптимизации с помощью компилятора, которые способны существенно, до 50%, увеличить размер кода. Хотя некоторые поставщики процессоров применяют активную развертку циклов, удаление обращений к подпрограммам и другие методы оптимизации, повышающие производительность за счет длины программы, EEMBC не требует, чтобы оценки по CoreMark декларировали размер кода. Поскольку микроконтроллеры часто хранят программы в памяти на кристалле, увеличение объема кода может оказаться дорогой альтернативой, если компиляторные оптимизации всего лишь улучшают оценку по какому-либо тесту. Пользователям встраиваемых приложений нужно увидеть улучшения в реальных программах прежде, чем устанавливать соответствующие флажки компилятора. Следует заметить, что в компиляторах также имеются флажки минимизации размера кода, но при их установке может ухудшиться оценка по CoreMark.

Потребителям следует проводить собственное тестирование реальных программ при различных установках компилятора.

## ЗАКЛЮЧЕНИЕ

При создании семейства ARC HS компания Synopsys преследовала доминирующую цель – спроектировать по возможности максимально быстрые встраиваемые процессорные ядра, сочетающие как простоту и энергоэффективность скалярного конвейера, так и малую площадь, занимаемую на кремниевом кристалле. Суперскалярное исполнение команд обеспечило бы более высокую производительность, однако дополнительная логика, которая нужна для нескольких конвейеров, требует увеличения энергопотребления и размеров кристалла. Микроархитектура с полностью внеочередным исполнением команд и использованием более глубокой конвейерной обработки так же могла бы повысить пропускную способность, но опять же – за счет более высокого энергопотребления и дополнительной логики, необходимой для сложной диспетчеризации команд.

Разработать высокопроизводительный процессор не так сложно, если нет ограничений по энергопотреблению и количеству транзисторов. Гораздо труднее создать компактный эффективный процессор, который обеспечивал бы достаточно высокую производительность сегодня с возможностью дальнейшего роста. Наиболее важные свойства ядер ARC HS – это те, которые действительно имеют значение для разработчиков встраиваемых систем. Перечислим еще раз ключевые особенности этих процессорных ядер:

- высокие тактовые частоты (типичное значение до 2,2 ГГц, в наихудшем случае – 1,6 ГГц) при использовании широко распространенного технологического процесса (28-нм КМОП-процесс HPM); возможность повышения тактовой частоты при переходе на более совершенный технологический процесс;
- хорошая энергоэффективность: всего 0,036 мВт/МГц для ядра ARC HS38, изготовленного по 28-нм технологии HPM; управление питанием с динамическим масштабированием напряжения / частоты;
- низкая стоимость за счет малой площади кристалла: всего 0,2 мм<sup>2</sup> при использовании 28-нм процесса HPM;
- ядро ARC HS38: усовершенствованный блок управления памятью с конфигурируемой 32–40-разрядной адресацией физической памяти и расширяемым буфером ассоциативной трансляции с упреждением;
- ядра ARC HS38 и ARC HS36: опциональная L2-кэш-память (конфигурируемая до 8 Мбайт) и опциональная коррекция ошибок (ECC);
- приблизительно на 18% более высокая плотность кода по сравнению с процессорными ядрами на базе ARCv1;
- широкие возможности расширения и конфигурирования, в том числе определяемых пользователем команд и регистров;
- быстрое переключение контекста с использованием до восьми массивов регистров;
- новые векторные команды для приложений с большим объемом вычислений;
- опциональный модуль операций с плавающей запятой с одинарной и двойной точностью (100%-ная совместимость с IEEE-754.2008);
- шины и интерфейсы AXI и AHB-Lite с малой задержкой;
- полная поддержка трассировки кристалла извне в режиме реального времени;
- обширная экосистема аппаратных и программных средств проектирования и отладки, макетных плат, а также поддержка сторонних компаний;
- возможность использования DesignWare IP и инструментов верификации от Synopsys.