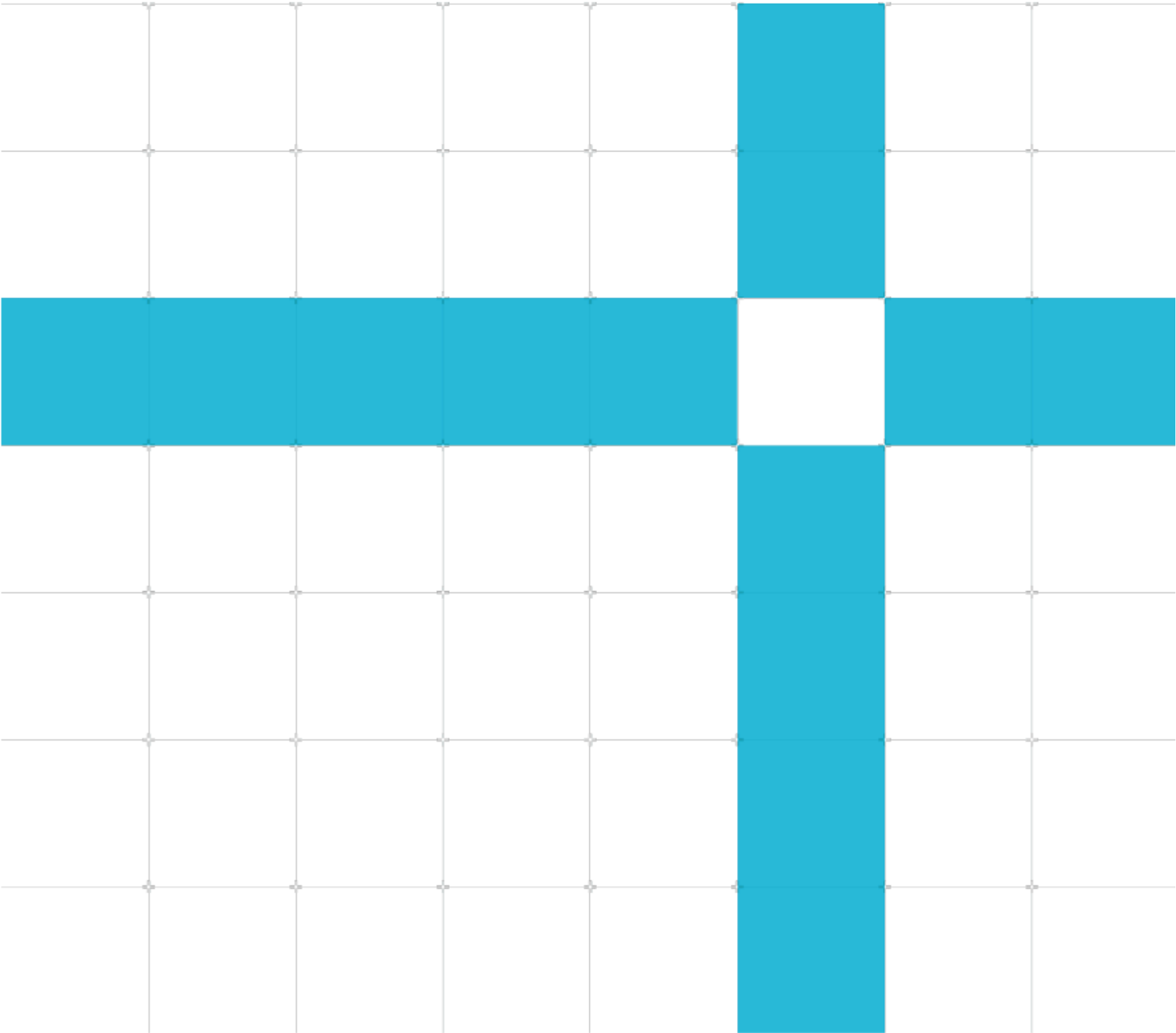




Введение в AMBA AXI4

Неконфиденциальный
Авторские права © 2020 Arm Limited (или ее филиалы). Все права защищены.

Выпуск 0101
102202



Введение в AMBA AXI4

Copyright 2020 Arm Limited (или ее аффилированные лица). Все права защищены.

Информация о выпуске

История документов

Проблема	Дата	Конфиденциальность	Изменение
01	25 августа 2020 г.	Неконфиденциальный характер	Первая версия

Уведомление о правах собственности неконфиденциального характера

Этот документ защищен авторским правом и другими смежными правами, и практика или реализация информации, содержащейся в этом документе, может быть защищена одним или несколькими патентами или находящимися на рассмотрении патентными заявками. Никакая часть этого документа не может быть воспроизведена в любой форме любыми средствами без явно выраженного предварительного письменного разрешения Arm. Настоящий документ не предоставляет никаких лицензий, явных или подразумеваемых, путем эстоппеля или иным образом на какие-либо права интеллектуальной собственности, за исключением специально оговоренных случаев.

Ваш доступ к информации, содержащейся в этом документе, зависит от вашего согласия с тем, что вы не будете использовать или разрешать другим лицам использовать информацию для целей определения того, нарушают ли реализации какие-либо патенты третьих сторон.

ЭТОТ ДОКУМЕНТ ПРЕДОСТАВЛЯЕТСЯ "КАК ЕСТЬ". ARM НЕ ПРЕДОСТАВЛЯЕТ НИКАКИХ ЗАЯВЛЕНИЙ И ГАРАНТИЙ, ЯВНЫХ, ПОДРАЗУМЕВАЕМЫХ ИЛИ УСТАНОВЛЕННЫХ ЗАКОНОМ, ВКЛЮЧАЯ, ПОМИМО ПРОЧЕГО, ПОДРАЗУМЕВАЕМЫЕ ГАРАНТИИ ТОВАРНОЙ ПРИГОДНОСТИ, УДОВЛЕТВОРИТЕЛЬНОГО КАЧЕСТВА, НЕНАРУШЕНИЯ ПРАВ ИЛИ ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ В ОТНОШЕНИИ ДОКУМЕНТА. Во избежание сомнений, Arm не делает никаких заявлений в отношении патентов, авторских прав, коммерческой тайны или других прав и не проводила никакого анализа для выявления или понимания объема и содержания этих патентов.

Этот документ может содержать технические неточности или типографские ошибки.

В ПРЕДЕЛАХ, НЕ ЗАПРЕЩЕННЫХ ЗАКОНОМ, ARM НИ ПРИ КАКИХ ОБСТОЯТЕЛЬСТВАХ НЕ НЕСЕТ ОТВЕТСТВЕННОСТИ ЗА КАКИЕ-ЛИБО УБЫТКИ, ВКЛЮЧАЯ, ПОМИМО ПРОЧЕГО, ЛЮБЫЕ ПРЯМЫЕ, КОСВЕННЫЕ, ОСОБЫЕ, СЛУЧАЙНЫЕ, ШТРАФНЫЕ Или КОСВЕННЫЕ УБЫТКИ, НЕЗАВИСИМО ОТ ПРИЧИН И ТЕОРИИ ОТВЕТСТВЕННОСТИ, ВОЗНИКАЮЩИЕ В РЕЗУЛЬТАТЕ ЛЮБОГО ИСПОЛЬЗОВАНИЯ ЭТОГО ДОКУМЕНТА, ДАЖЕ ЕСЛИ ARM БЫЛА ПРЕДУПРЕЖДЕНА О ВОЗМОЖНОСТИ ТАКИХ УБЫТКОВ. Этот документ состоит исключительно из коммерческих материалов. Вы несете ответственность за обеспечение того, чтобы любое использование, дублирование или раскрытие этого документа полностью соответствовало любым соответствующим экспортным законам и нормативным актам, гарантирующим, что этот документ или любая его часть не экспортируется, прямо или косвенно, в нарушение таких экспортных законов.

Использование слова "партнер" по отношению к клиентам Arm не предназначено для создания или обозначения каких-либо партнерских отношений с какой-либо другой компанией. Arm может вносить изменения в этот документ в любое время и без предварительного уведомления.

Если какое-либо из положений, содержащихся в настоящих условиях, вступает в противоречие с каким-либо из положений любого пользовательского или подписанного письменного соглашения, охватывающего этот документ, с Arm, то пользовательское или подписанное письменное соглашение имеет преимущественную силу и заменяет противоречащие положения настоящих условий. Для удобства этот документ может быть переведен на другие языки, и вы соглашаетесь с тем, что в случае возникновения каких-либо противоречий между английской версией этого документа и любым переводом преимущественную силу имеют условия английской версии Соглашения.

На Руке фирменный логотип и слова, помеченные ® or ™ являются зарегистрированными товарными знаками или торговыми знаками Arm Limited (или ее аффилированных лиц) в США и / или где-либо еще. Все права защищены. Другие торговые марки и названия, упомянутые в этом документе, могут быть товарными знаками соответствующих владельцев. Пожалуйста, следуйте рекомендациям Arm по использованию товарных знаков по адресу <http://www.arm.com/company/policies/trademarks>.

Авторские права © 2020 Arm Limited (или ее филиалы). Все права защищены.

Arm Limited. Компания 02557590 зарегистрирована в Англии.

110 Fulbourn Road, Кембридж, Англия CB1 9NJ.

(ФАЙЛ, действовавший ДО 20349 года)

Статус конфиденциальности

Этот документ не является конфиденциальным. Право на использование, копирование и разглашение этого документа может быть ограничено лицензией в соответствии с условиями соглашения, заключенного Arm и стороной, которой Arm передала этот документ.

Неограниченный доступ является внутренней классификацией Arm.

Веб-адрес

www.arm.com

Содержание

1 . Overview.....6

2 Что такое AMBA и зачем использовать it?.....7

2.1 Где находится AMBA used?.....7

2.2 Зачем использовать AMBA?.....8

2.3 Как работает AMBA evolved?.....8

2.3.1 AMBA 19

2.3.2 AMBA 29

2.3.3 AMBA 39

2.3.4 AMBA 410

2.3.5 AMBA 510

3 Обзор протокола AXI11

3.1 AXI в мульти-мастере system.....12

3.2 Каналы AXI14

3.3 Основные AXI features.....15

Передача данных по 4 каналам и transactions..... 17

4.1 Канал handshake.....17

4.2 Различия между передачей и transactions.....17

4.3 Канал передачи examples.....19

4.4 Транзакция записи: отдельные данные item.....21

4.5 Транзакция записи: несколько элементов данных23

4.6 Транзакция чтения: один элемент данных24

4.7 Транзакция чтения: несколько данных items.....26

4.8 Активные транзакции27

5 Канал signals.....29

5.1 Канал записи signals.....29

5.2 Сигналы канала чтения30

5.3 Размер, длина и тип пакета данных32

5.4 Поддержка уровня защиты.....32

Кэш 5.5 support.....33

5.6 Сигнализация ответа35

5.7 Стробоскопы записи данных35

5.8 Атомарные обращения с блокировкой signal.....36

5.9 Качество обслуживания37

5.10 Сигнализация региона38

5.11 Пользователь signals.....38

5.12 Канал AXI dependencies.....39

6 Атомных accesses..... 40

6.1 Заблокирован accesses.....40

6.2 Эксклюзивные доступа42

6.3 Аппаратный монитор эксклюзивного доступа operation.....42

6.4 Эксклюзивные пары транзакций: обе pass.....43

6.5 Эксклюзивные пары транзакций: один проход, один сбой45

7 Поведение при передаче и транзакция ordering..... 47

7.1 Примеры простых транзакций47

7.2 Идентификаторы переводов49

7.3 Правила записи порядка транзакций50

7.4 Правила чтения порядка транзакций51

7.5 Порядок чтения и записи каналов53

7.6 Невыровненный начальный адрес передачи54

7.7 Поддержка порядкового номера55

7.8 Атрибуты интерфейса чтения и записи56

8 Проверьте свои знания 58

9 Сопутствующая информация 59

10 Следующий steps..... 60

1 Обзор

Это руководство знакомит с основными функциями Advanced Microcontroller Bus

Architecture (AMBA) AXI4, подчеркивая отличия от предыдущей версии AXI3. В руководстве

объясняются ключевые концепции и детали, которые помогут вам реализовать протокол

AXI4. В этом руководстве мы описываем:

- Что такое AMBA.
- Почему AMBA так популярна в современном дизайне SoC.
- Концепции переводов и транзакций, лежащие в основе работы AMBA.
- Различные сигналы каналов и функциональность, которую они предоставляют.

• Передача эксклюзивного доступа, которая позволяет нескольким ведущим устройствам

одновременно получать доступ к одному ведомому устройству в одно и то же время.

• Правила и условия, которые диктует протокол AMBA. Ключевые

• атрибуты и поддержка распространенных элементов, таких как смешанные конечные

структуры. В конце этого руководства вы можете [Проверить свои знания](#).

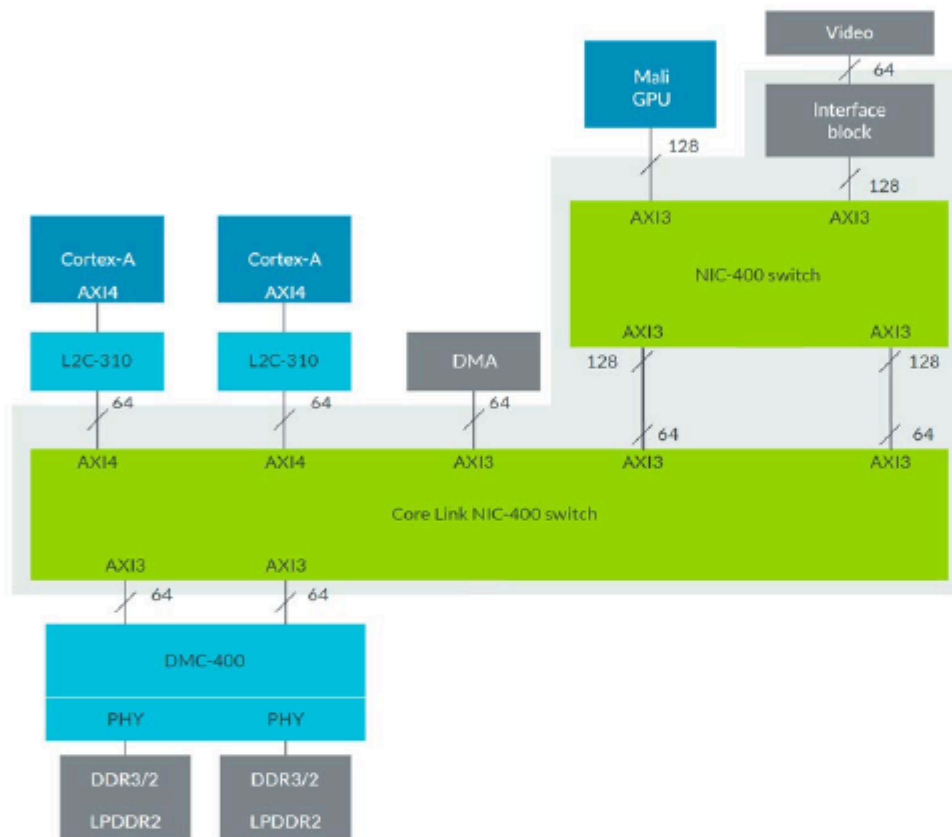
2 Что такое AMBA и зачем его использовать?

Архитектура шины микроконтроллера Arm Advanced, или AMBA, представляет собой открытый стандарт на кристалле.

спецификация межсоединений для подключения функциональных блоков и управления ими в конструкциях "система на кристалле" (SoC).

По сути, протоколы AMBA определяют, как функциональные блоки взаимодействуют друг с другом.

На следующей диаграмме показан пример конструкции SoC. Этот SoC имеет несколько функциональных блоков, которые используют протоколы AMBA, такие как AXI4 и AXI3, для связи друг с другом:



2.1 Где используется AMBA?

AMBA упрощает разработку конструкций с несколькими процессорами и большим количеством контроллеров и периферийных устройств. Однако со временем сфера применения AMBA расширилась, выйдя далеко за рамки только микроконтроллерных устройств. Сегодня AMBA широко используется в ряде компонентов ASIC и SoC. Эти компоненты включают приложения процессоры, которые используются в таких устройствах, как подсистемы Интернета вещей, смартфоны и сетевые SoC.

2.2 Зачем использовать AMBA?

AMBA предоставляет ряд преимуществ:

- Эффективное повторное использование IP

Повторное использование IP является важным компонентом сокращения затрат на разработку SoC и сроков.

Спецификации AMBA обеспечивают стандарт интерфейса, позволяющий повторно использовать IP. Таким образом, тысячи SOC и IP-продуктов используют интерфейсы AMBA.

- Гибкость •

AMBA обеспечивает гибкость при работе с целым рядом SOC. Для повторного использования IP требуется единый стандарт, при этом поддерживается широкий спектр SOC с различными требованиями к мощности, производительности и площади. Arm предлагает ряд спецификаций интерфейса, оптимизированных для этих различных требований.

- Совместимость •

Стандартная спецификация интерфейса, такая как AMBA, обеспечивает совместимость между IP-компонентами от разных групп разработчиков или поставщиков.

- Поддержка •

AMBA хорошо поддерживается. Он широко внедрен и поддерживается во всей полупроводниковой отрасли, включая поддержку IP-продуктов и инструментов сторонних производителей.

Стандарты интерфейса шины, такие как AMBA, различаются по производительности, которую они обеспечивают.

Двумя основными характеристиками производительности интерфейса шины являются:

- Пропускная способность

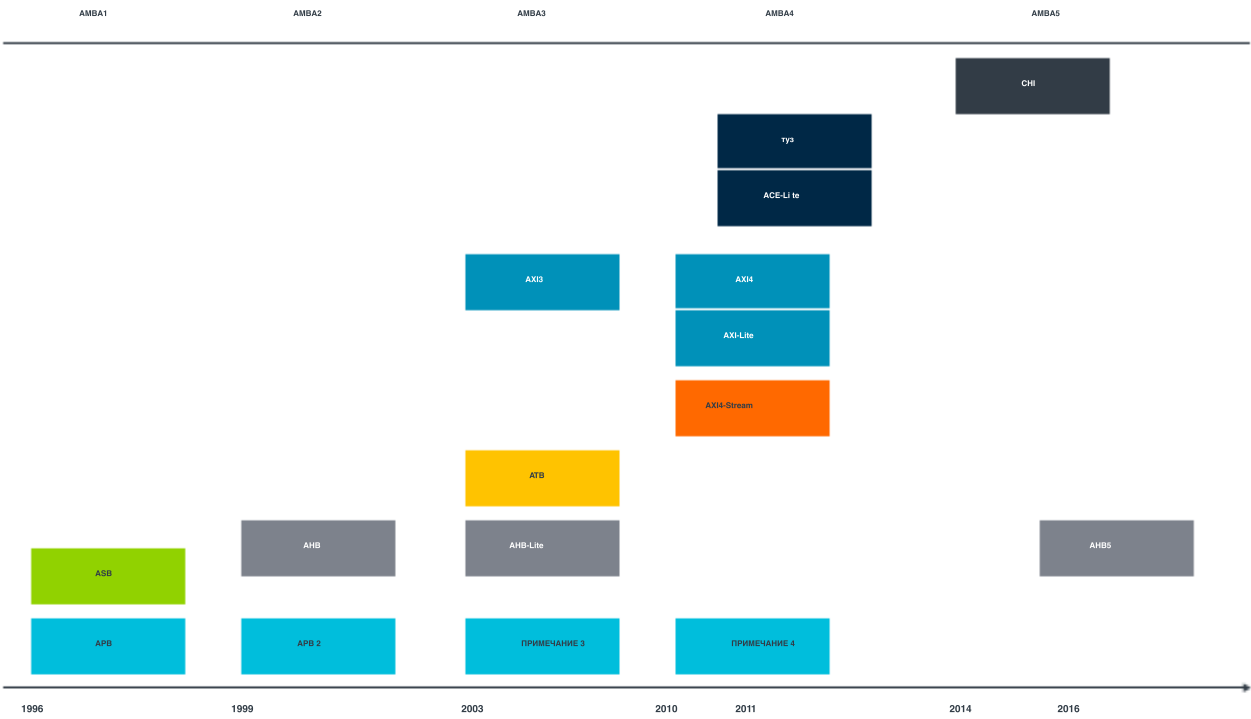
Скорость, с которой данные могут передаваться через интерфейс. В синхронной системе максимальная пропускная способность ограничена производением тактовой частоты на ширину шины данных.

- Задержка •

Задержка между инициированием и завершением транзакции. В системе, основанной на пакетной передаче, показатель задержки часто относится к завершению первой передачи, а не ко всему пакету. Эффективность вашего интерфейса зависит от степени, в которой он обеспечивает максимальную пропускную способность при нулевой задержке.

2.3 Как развивался AMBA?

С годами AMBA развивалась в соответствии с требованиями процессоров и новых технологий, как показано на следующей диаграмме:



2.3.1 AMBA 1

Arm представила AMBA в конце 1990-х годов. Первыми шинами AMBA были Advanced System Bus (ASB) и Advanced Peripheral Bus (APB). ASB был заменен более свежими протоколами, в то время как APB все еще широко используется сегодня. APB разработан для доступа к управлению с низкой пропускной способностью, например, к интерфейсам регистрации на системной периферии. Эта шина имеет простой адрес и фазу передачи данных, а также список сигналов низкой сложности.

2.3.2 AMBA 2

В 1999 году AMBA 2 добавила высокопроизводительную шину AMBA (AHB), которая представляет собой единый тактовый интерфейс протокола. Простая транзакция в AHB состоит из фазы обращения и последующей фазы передачи данных. Доступ к целевому устройству контролируется через мультиплексор, разрешающий доступ к одному ведущему устройству одновременно. AHB конвейеризован для повышения производительности, в то время как APB не конвейеризован для простоты проектирования.

2.3.3 AMBA 3

В 2003 году Arm представила третье поколение AMBA 3, которое включает ATB и AHB-Lite.

Расширенная шина трассировки (ATB) является частью решения CoreSight для отладки и трассировки на кристалле.

AHB-Lite является подмножеством AHB. Это подмножество упрощает конструкцию шины с одним ведущим устройством.

Расширенный расширяемый интерфейс (AXI), третье поколение интерфейса AMBA, определенное в спецификации AMBA 3, предназначено для высокопроизводительных систем с высокой тактовой частотой. AXI включает в себя функции, которые делают его подходящим для высокоскоростного субмикрометрического соединения.

2.3.4 AMBA 4

В 2010 году были представлены спецификации AMBA 4, начиная с AMBA 4 AXI4, а затем AMBA 4 AXI Coherency Extensions (ACE) в 2011 году. ACE расширяет AXI дополнительной сигнализацией, обеспечивая общесистемную согласованность. Эта общесистемная согласованность позволяет нескольким процессорам совместно использовать память и обеспечивает такие технологии, как big.MАЛЕНЬКАЯ обработка. В то же время протокол ACE-Lite обеспечивает одностороннюю согласованность. Односторонняя согласованность позволяет сетевому интерфейсу считывать данные из кэшей полностью согласованного процессора ACE. Протокол AXI4-Stream предназначен для односторонней передачи данных от ведущего устройства к ведомому с уменьшенной маршрутизацией сигналов, что идеально подходит для реализации в ПЛИС.

2.3.5 AMBA 5

В 2014 году была представлена спецификация интерфейса AMBA 5 Coherent Hub Interface (CHI) с переработанным высокоскоростным транспортным уровнем и функциями, предназначенными для уменьшения перегрузки. Было выпущено несколько версий протокола CHI, и каждая новая версия добавляет новые функции. В 2016 году протокол AHB-Lite был обновлен до AHB5, чтобы дополнить архитектуру Armv8-M и расширить TrustZone security foundation от процессора до системы. В 2019 году была представлена программа AMBA Adaptive Traffic Profiles (ATP). ATP дополняет существующие протоколы AMBA и используется для моделирования поведения при доступе к памяти высокого уровня в сжатой, простой и переносимой форме.

Обзор протокола AXI 3

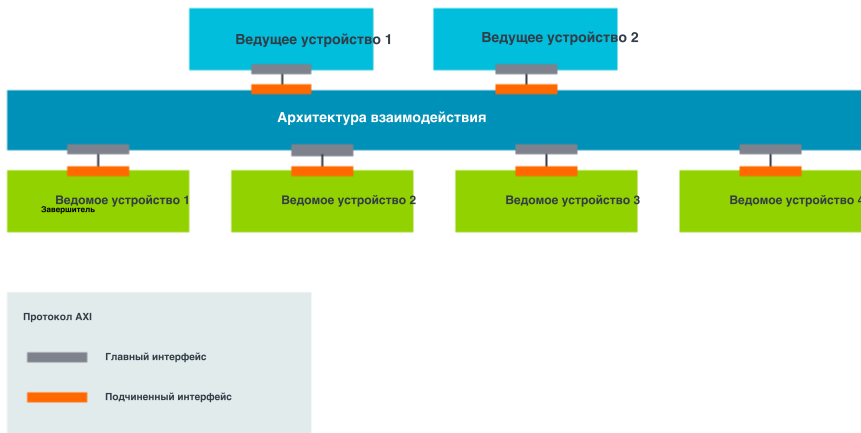
AXI - это спецификация интерфейса, которая определяет интерфейс IP-блоков, а не само межсоединение . На следующей схеме показано, как AXI используется для взаимодействия с компонентом межсоединения:



В AX3 и AXI4 есть только два типа интерфейса AXI, ведущий и ведомый. Эти типы интерфейса симметричны. Все соединения AXI осуществляются между ведущими и ведомыми интерфейсами. Интерфейсы AXI interconnect содержат одинаковые сигналы, что делает интеграцию различных IP относительно простой. На предыдущей схеме показано, как соединения AXI соединяют интерфейсы master и slave. Прямое соединение обеспечивает максимальную пропускную способность между компонентами master и slave без дополнительной логики. А в AXI существует только один протокол для проверки.

3.1 AXI в системе с несколькими мастерами

На следующей диаграмме показан упрощенный пример SoC-системы, которая состоит из ведущих устройств, ведомых устройств и межсоединения, которое связывает их все:



Процессор Arm является примером ведущего устройства, а простым примером ведомого устройства является контроллер памяти.

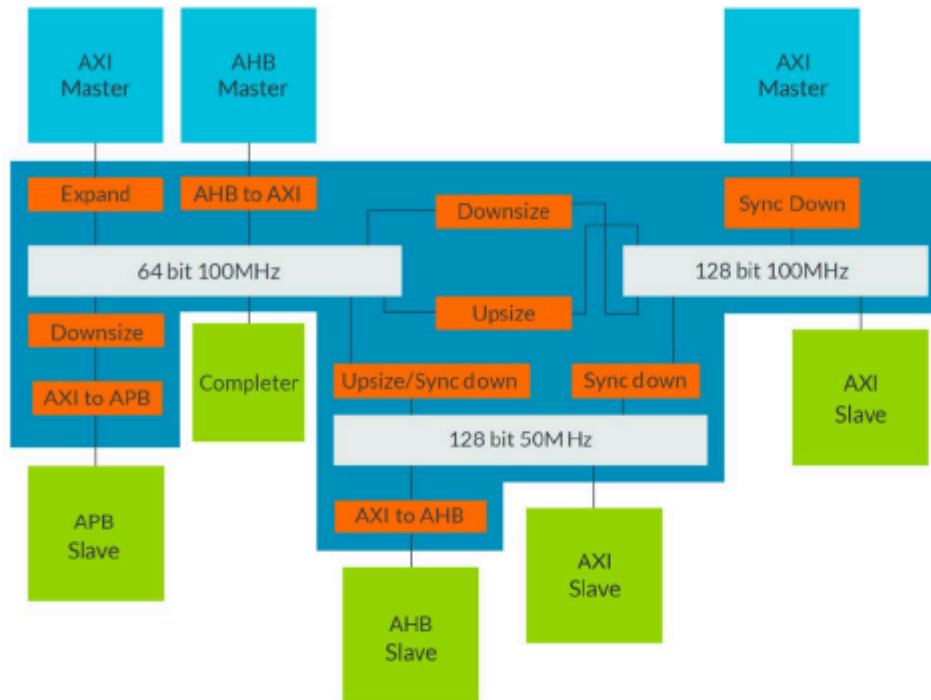
Протокол AXI определяет сигналы и синхронизацию соединений "точка-точка" между ведущими устройствами и ведомыми устройствами.



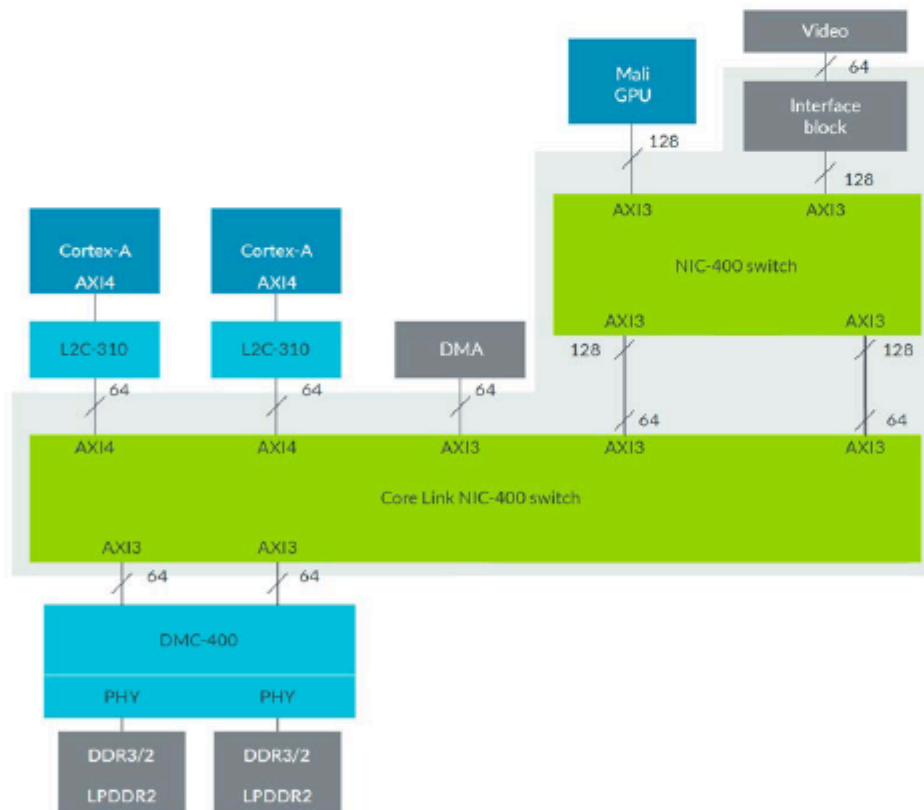
Протокол AXI - это спецификация "точка-точка", а не спецификация шины. Следовательно, он описывает только сигналы и синхронизацию между интерфейсами.

На предыдущей схеме показано, что каждый главный интерфейс AXI подключен к одному подчиненному интерфейсу AXI . Там, где задействовано несколько ведущих и ведомых устройств, требуется структура межсоединений. Эта структура межсоединений также реализует интерфейсы slave и master, в которых реализован протокол AXI.

На следующей диаграмме показано, что межсоединение является сложным элементом, для которого требуется собственный AXI интерфейсы master и slave для взаимодействия с внешними функциональными блоками:



На следующей диаграмме показан пример SoC с различными процессорами и функциональными блоками:

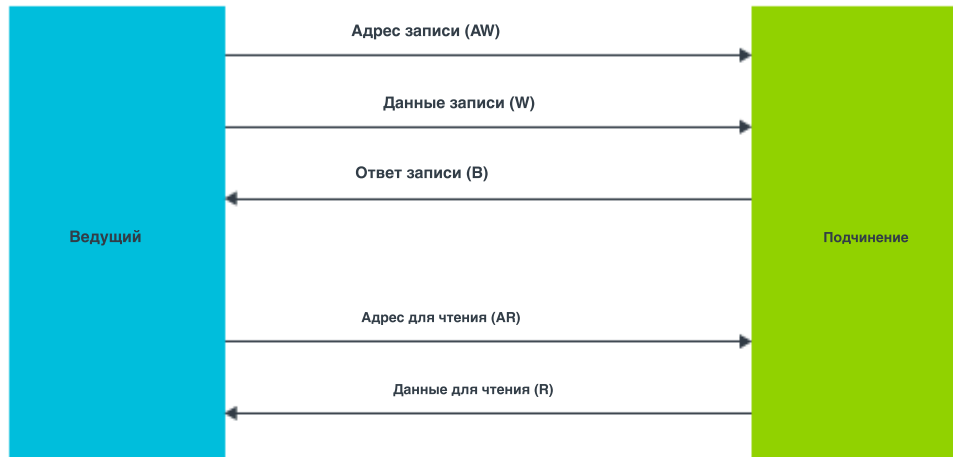


На предыдущей диаграмме показаны все соединения, в которых используется AXI. Вы можете видеть, что AXI3 и AXI4 используются в рамках одного SoC, что является обычной практикой. В таких случаях интерконнект выполняет преобразование протокола между различными интерфейсами AXI.

3.2 Каналы AXI

Спецификация AXI описывает протокол "точка-точка" между двумя интерфейсами: ведущим и подчиненным. На следующей диаграмме

показаны пять основных каналов, которые каждый интерфейс AXI использует для связи.:



Операции записи используют

следующие каналы: • Ведущий отправляет адрес по каналу Адреса записи (AW) и передает данные по каналу записи Data (W) на ведомое устройство.

- Ведомое устройство записывает полученные данные по указанному адресу. Как только ведомое устройство завершит операцию записи, оно отвечает сообщением ведущему устройству по каналу ответа на запись (B).

Для операций чтения используются

следующие каналы: • Ведущее устройство отправляет адрес, который оно хочет прочитать, по каналу адреса чтения (AR).

- Ведомое устройство отправляет данные с запрошенного адреса ведущему устройству по каналу данных чтения (R).

Ведомое устройство также может возвращать сообщение об ошибке по каналу считанных данных (R). Ошибка возникает, если, например, адрес недействителен, или данные повреждены, или доступ не имеет надлежащего разрешения безопасности.



Каждый канал является однонаправленным, поэтому для передачи ответов обратно ведущему устройству необходим отдельный канал ответа на запись. Однако нет необходимости в канале ответа на чтение, поскольку ответ на чтение передается как часть канала считанных данных.

Использование отдельных каналов адреса и данных для передачи данных чтения и записи помогает максимально увеличить пропускную способность интерфейса. Между группами каналов чтения и записи нет временной зависимости. Это означает, что последовательность чтения может выполняться одновременно с последовательностью записи. Каждый из этих пяти каналов содержит несколько сигналов, и все эти сигналы в каждом канале имеют следующий префикс:

AW для сигналов на канале адреса

AR для сигналов на канале считывания
записи **
адреса *

W для сигналов на канале
R для сигналов на канале считывания данных
данных записи *

* B для сигналов в канале ответа на запись



B означает буферизованный, потому что ответ от ведомого устройства приходит после завершения всех операций записи.

3.3 Основные функции AXI

Протокол AXI обладает несколькими ключевыми функциями, которые предназначены для улучшения пропускной способности и задержки при передаче данных и транзакциях, как вы можете видеть здесь: Независимые каналы чтения и записи

AXI поддерживает два разных набора каналов, один для операций записи, а другой для операций чтения. Наличие двух независимых наборов каналов помогает улучшить пропускную способность интерфейсов. Это связано с тем, что операции чтения и записи могут выполняться одновременно.

Несколько незавершенных адресов.

AXI допускает наличие нескольких незавершенных адресов. Это означает, что мастер может выполнять транзакции, не дожидаясь завершения предыдущих транзакций. Это может повысить производительность системы поскольку позволяет выполнять параллельную обработку транзакций.

Нет строгой временной зависимости между операциями с адресом и данными

В AXI нет строгой временной зависимости между операциями с адресом и данными. Это означает, что, например, ведущее устройство могло бы выдать адрес записи в канале адреса записи, но нет требования по времени, когда ведущее устройство должно предоставить соответствующие данные для записи в канале данных записи.

Поддержка несогласованных передач данных

Для любого пакета, состоящего из передач данных размером более одного байта, первые доступные байты могут не совпадать с границей естественного адреса. Например, 32-разрядный пакет данных, который начинается с байта по адресу, который не является границей 32-разрядного адреса.

Завершение транзакции не по порядку

С помощью AXI возможно завершение транзакции не по порядку. Протокол AXI включает идентификаторы транзакции, и нет ограничений на завершение транзакций с разными значениями идентификатора.

Это означает, что один физический порт может поддерживать неупорядоченные транзакции, действуя как несколько логических портов, каждый из которых обрабатывает свои транзакции по порядку.

Пакетные транзакции на основе начального адреса.

Ведущие устройства AXI выдают начальный адрес только для первой передачи. Для любых последующих передач подчиненное устройство вычислит адрес следующей передачи на основе типа пакета.

Передачи и транзакции по 4

каналам

В этом разделе объясняется принцип подтверждения связи для каналов AXI и показано, как подтверждение связи является базовым механизмом для всех транзакций чтения и записи.

Подтверждение связи по каналу 4.1

Протокол AXI4 определяет пять различных каналов, как описано в разделе Каналы AXI. Все эти каналы используют один и тот же механизм установления связи, основанный на действительных сигналах и сигналах ГОТОВНОСТИ, как показано на следующей диаграмме:



ДЕЙСТВИТЕЛЬНЫЙ сигнал поступает от источника к получателю, а ГОТОВЫЙ - от получателя к источнику.

Является ли источник или получатель ведущим или ведомым, зависит от того, какой канал используется. Для например, мастер является источником для канала чтения адресов, но местом назначения для канала чтения данных. Источник использует ДЕЙСТВИТЕЛЬНЫЙ сигнал, чтобы указать, когда доступна действительная информация. ДЕЙСТВИТЕЛЬНЫЙ сигнал должен оставаться подтвержденным, то есть установленным на высокий уровень, до тех пор, пока получатель не примет информацию. Сигналы, которые остаются подтвержденными таким образом, называются фиксированными сигналами.

Адресат указывает, когда он может принять информацию, используя сигнал ГОТОВНОСТИ. Сигнал ГОТОВНОСТИ направляется от канала назначения к каналу источника.

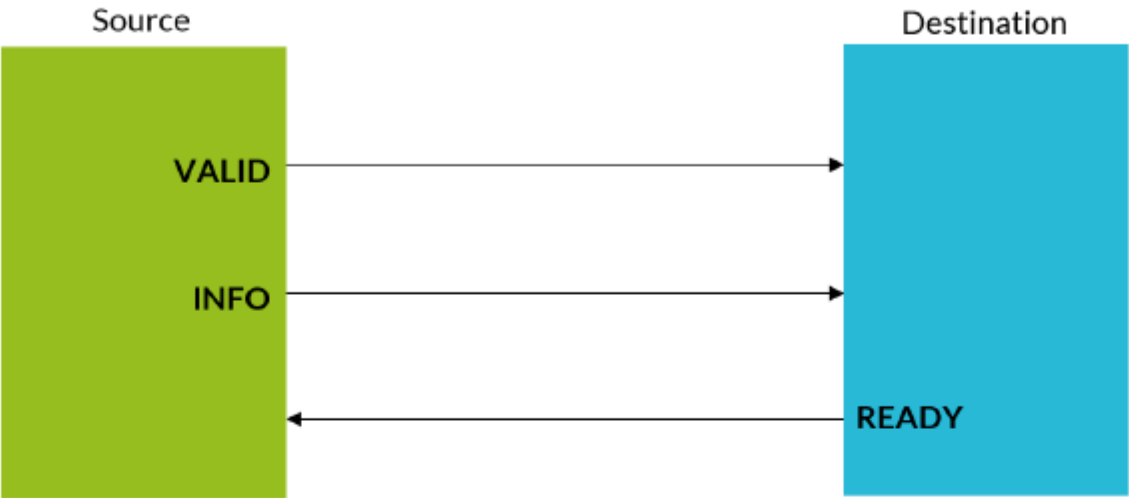
Этот механизм не является асинхронным обменом данными, и для завершения обмена данными требуется время нарастания тактовой частоты.

4.2 Различия между переводами и транзакциями

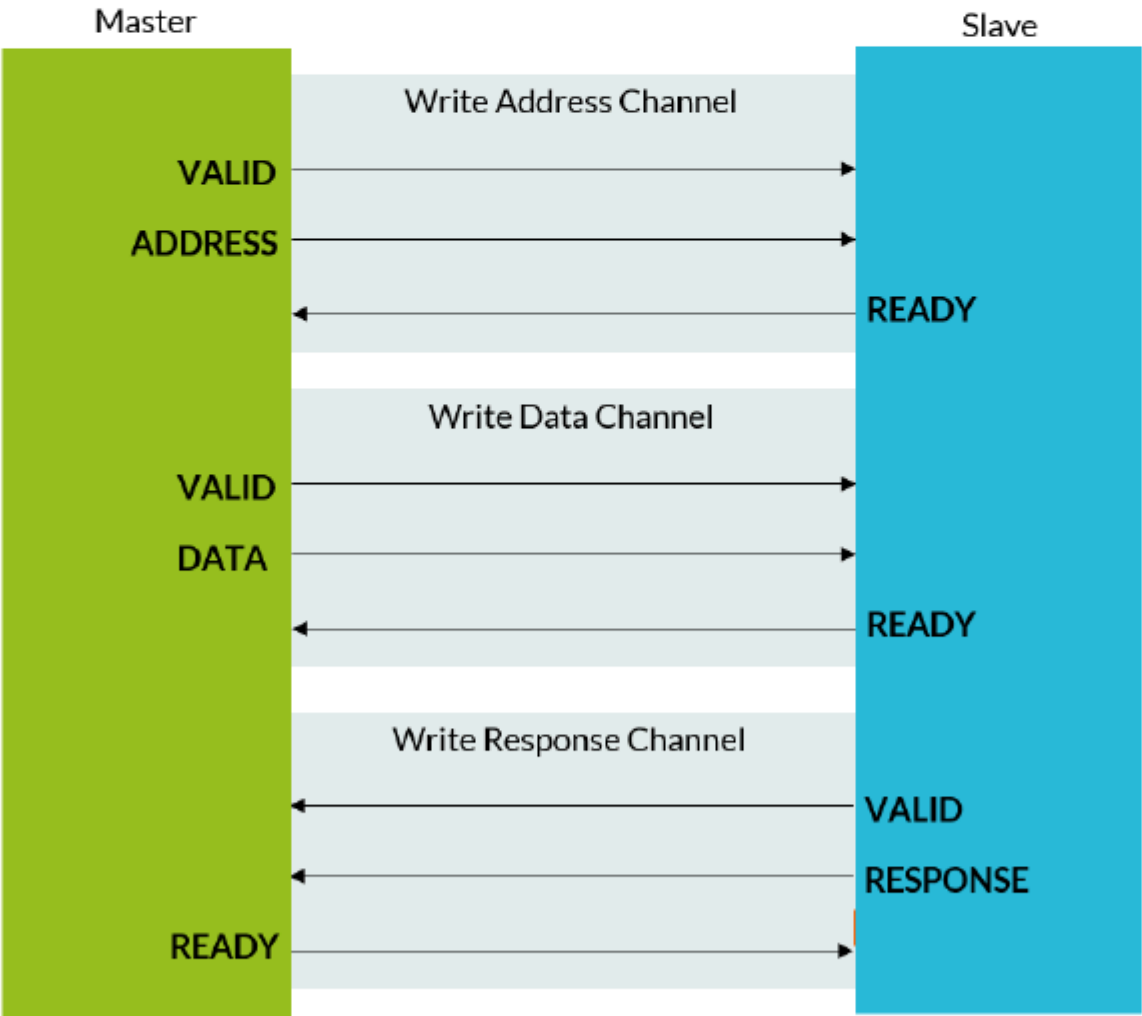
При проектировании структуры межсоединений вы должны знать возможности основных и ведомых устройств, которые подключаются. Знание этой информации позволяет включить достаточную логику буферизации, отслеживания и декодирования для поддержки различных вариантов упорядочения передачи данных, которые позволяют повысить производительность на более быстрых устройствах.

Использование стандартной терминологии облегчает понимание взаимодействия между подключенными компонентами. AXI проводит различие между переводами и транзакциями:

- Передача - это единый обмен информацией с одним **ДЕЙСТВИТЕЛЬНЫМ** и **ГОТОВЫМ** рукопожатием. На следующей диаграмме показана передача:

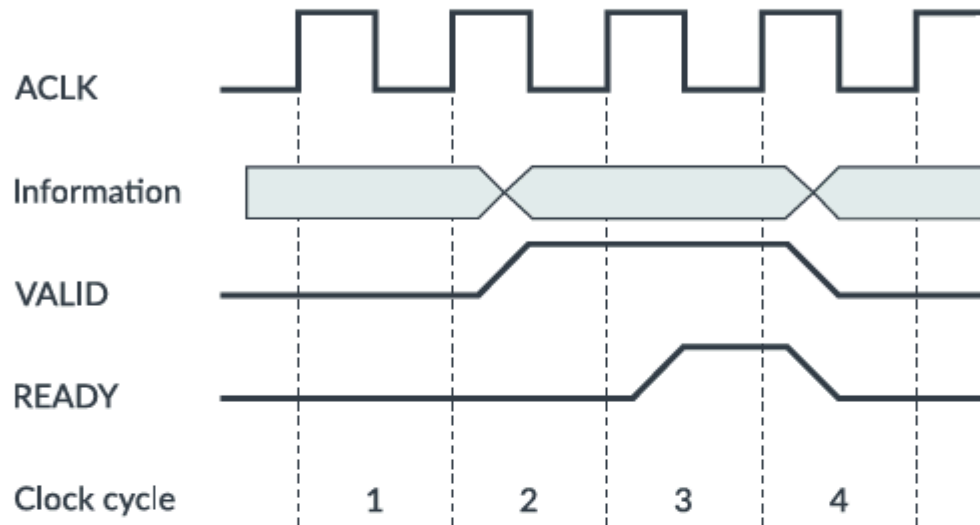


- Транзакция представляет собой целую серию передач, содержащую передачу адреса, одну или несколько передач данных и, для последовательностей записи, передачу ответа. На следующей диаграмме показана транзакция:



Примеры передачи данных по каналам 4.3

В этом разделе рассматриваются некоторые примеры возможных рукопожатий между источником и получателем. В нем показано несколько возможных комбинаций допустимых и ГОТОВЫХ последовательностей, соответствующих спецификациям протокола AXI . В первом примере, показанном на следующей диаграмме, у нас есть тактовый сигнал, за которым следует информационная шина, а затем **ДЕЙСТВИТЕЛЬНЫЕ** сигналы и сигналы **ГОТОВНОСТИ**:



Этот пример имеет следующую

последовательность событий: 1.

В такте синхронизации 2 утверждается **ДЕЙСТВИТЕЛЬНЫЙ** сигнал, указывающий на то, что

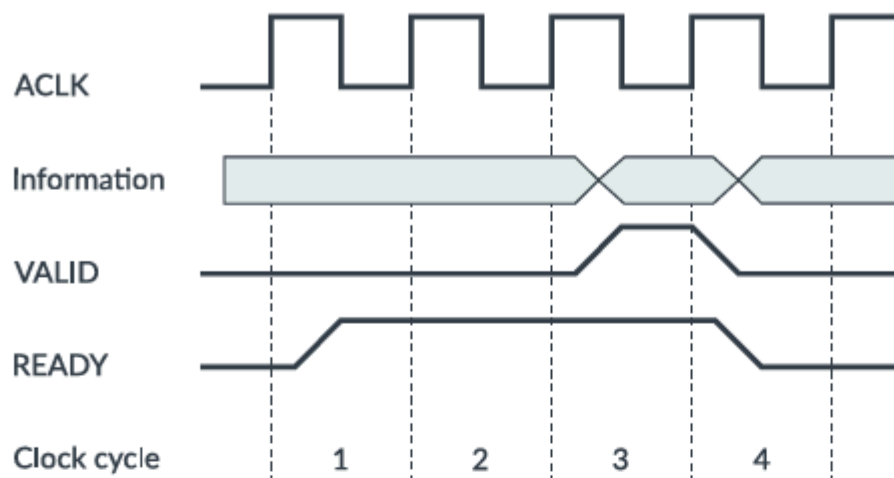
данные в информационном канале действительны. 2. В тактовом

цикле 3, следующем тактовом цикле, утверждается сигнал **ГОТОВНОСТИ**. 3. Рукопожатие

завершается на переднем фронте тактового цикла 4, поскольку оба **ГОТОВЫ** и **ДЕЙСТВИТЕЛЬНЫ**.

сигналы подтверждены.

На следующей диаграмме показан другой пример:



В этом примере представлена следующая

последовательность событий: 1. В такте синхронизации 1

утверждается сигнал ГОТОВНОСТИ. 2. ДЕЙСТВИТЕЛЬНЫЙ сигнал не

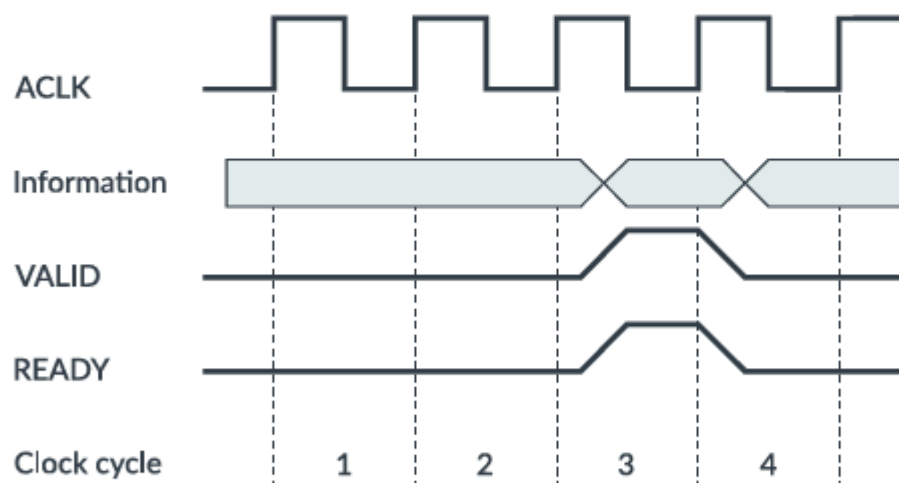
утверждается до 3-го такта синхронизации. 3.

Рукопожатие завершается на переднем фронте тактового цикла 4, когда

подтверждены как ДЕЙСТВИТЕЛЬНОСТЬ, так и ГОТОВНОСТЬ. Последний

пример показывает, что как ДЕЙСТВИТЕЛЬНЫЕ, так и ГОТОВЫЕ сигналы подтверждаются в

течение тактового цикла 3, как показано на следующей диаграмме:



И снова рукопожатие завершается на переднем фронте тактового цикла 4, когда

подтверждены как ДЕЙСТВИТЕЛЬНОСТЬ, так и ГОТОВНОСТЬ.

Во всех трех примерах информация передается по каналу, когда подтверждается ГОТОВНОСТЬ и

ДЕЙСТВИТЕЛЬНОСТЬ на переднем фронте синхросигнала.

При рукопожатии при чтении и записи должны

соблюдаться следующие правила: •
Источник не может ждать утверждения ГОТОВНОСТИ, прежде чем утверждать VALID.

- Получатель может ждать утверждения VALID, прежде чем утверждать READY.

Эти правила означают, что READY может быть подтвержден до или после VALID или даже одновременно.

4.4 Транзакция записи: единый элемент данных.

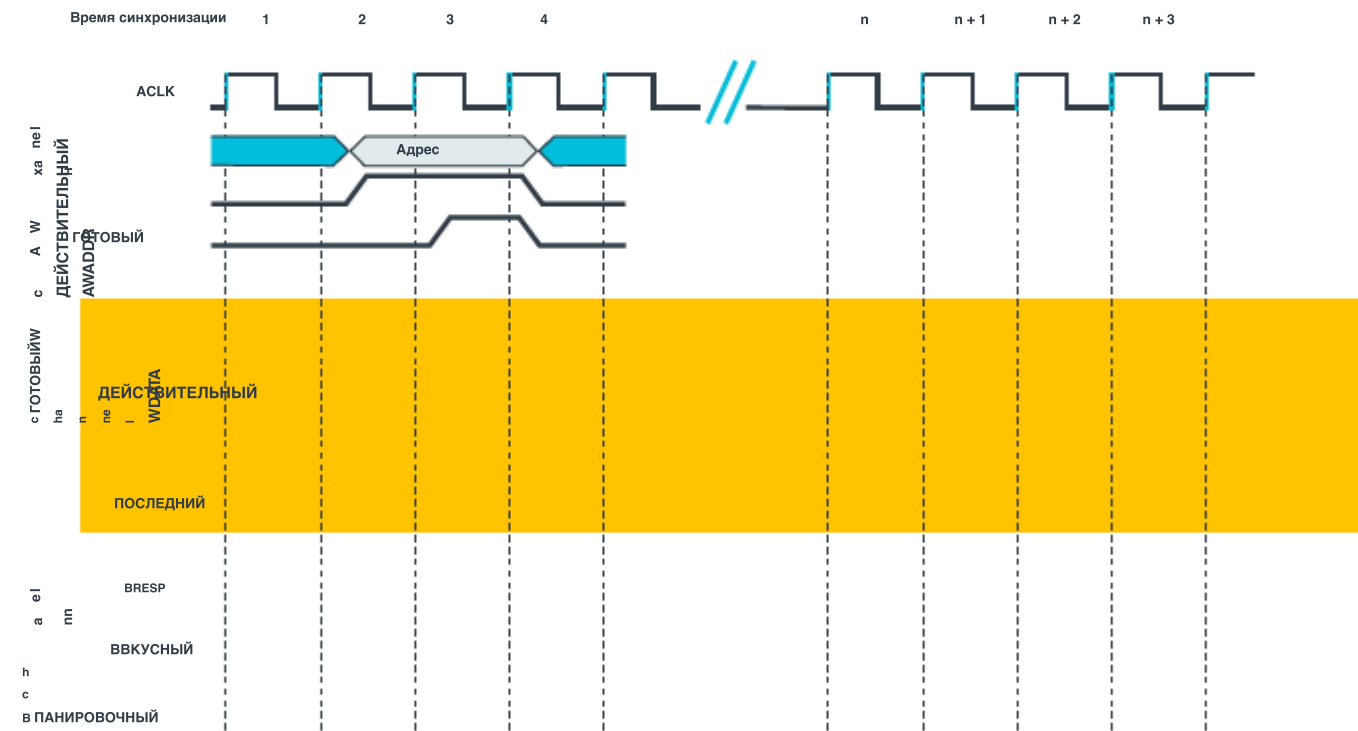
В этом разделе описывается процесс транзакции записи для отдельного элемента данных и различные

каналы, которые используются для завершения транзакции.

В этой транзакции записи задействованы следующие каналы:

- Адрес записи
(AW) •Написать (W)
- Ответ на запись (B)

Первый, происходит подтверждение связи по каналу адреса записи (AW), как показано на следующей диаграмме:

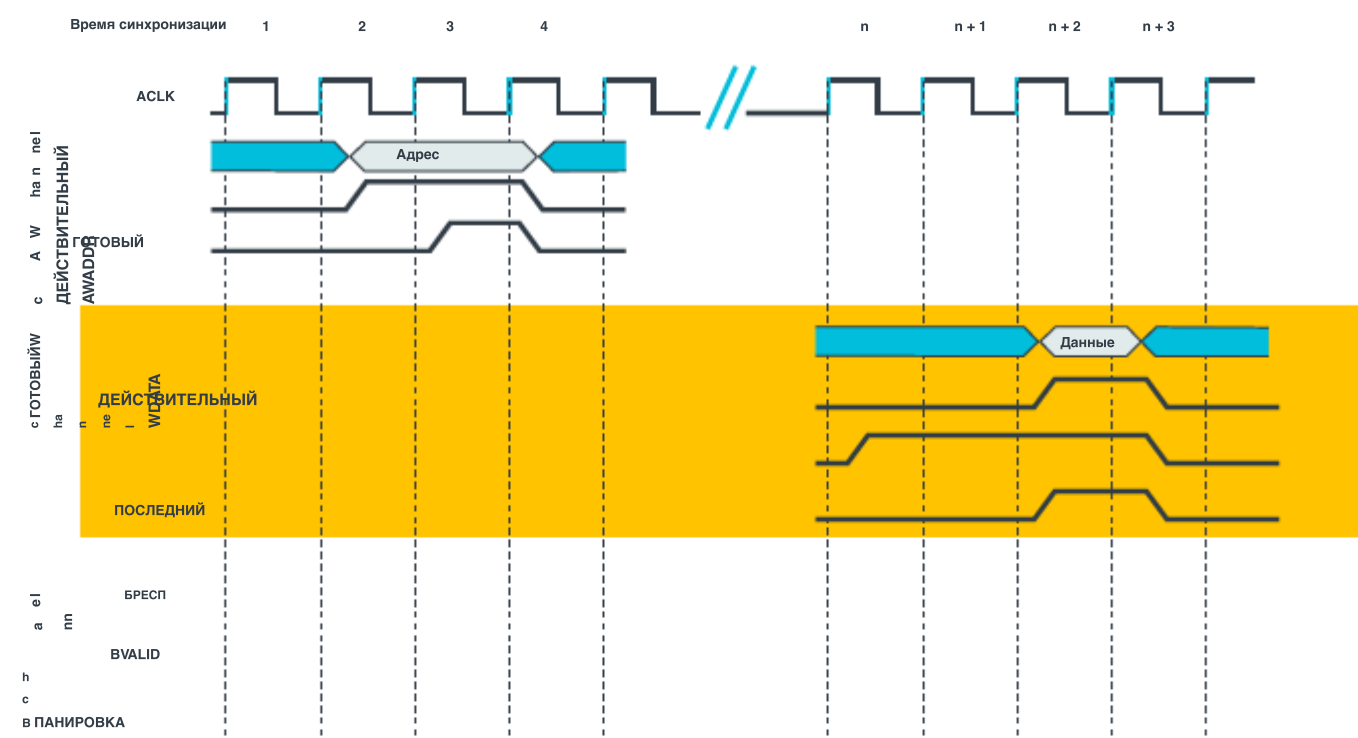


В этом квитировании ведущий передает адрес записи ведомому устройству. В

квитировании происходит следующая последовательность событий:

1. Ведущий вводит адрес в AWADDR и утверждает AWVALID в тактовом цикле 2.
2. Подчиненный утверждает AWREADY в тактовом цикле 3, чтобы указать свою способность получать значение адреса. 3. Рукопожатие завершается на восходящем фронте тактового цикла 4.

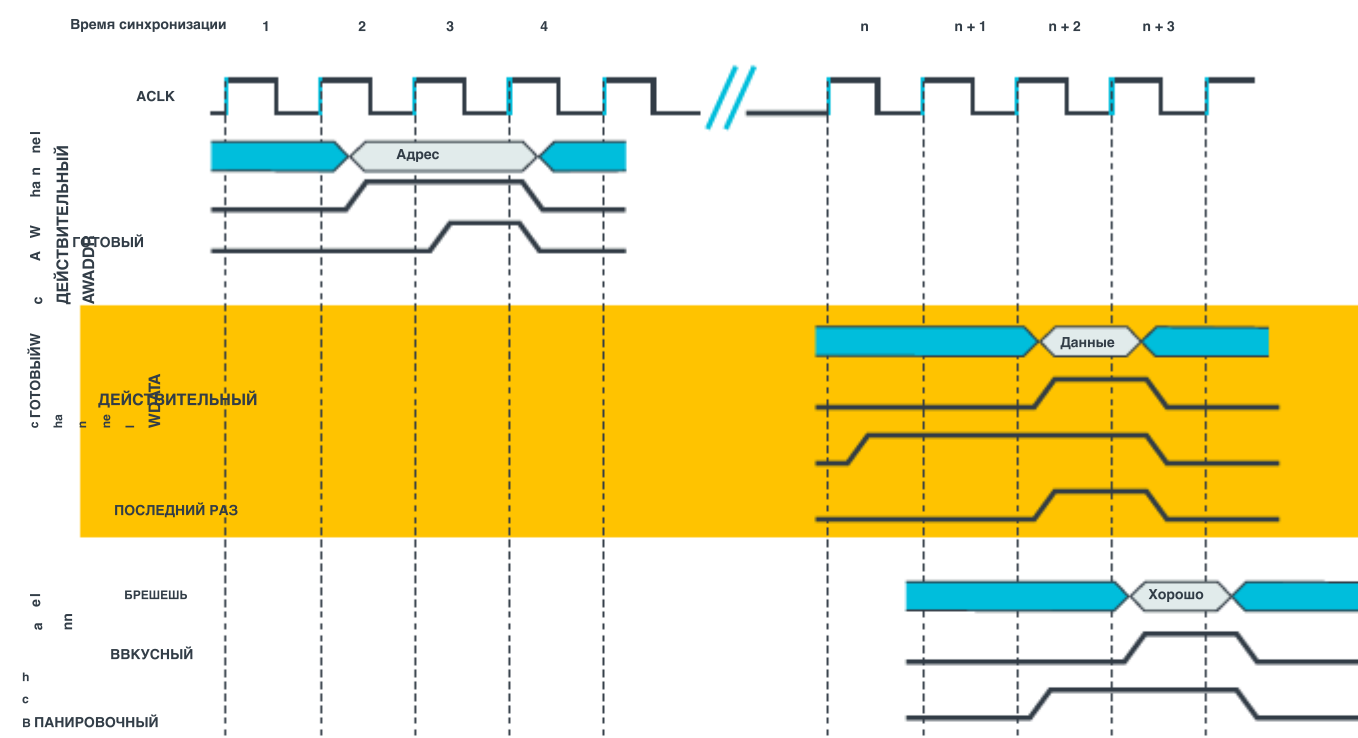
После этого первого подтверждения связи ведущее устройство передает данные ведомому устройству по каналу записи (W), как показано на следующей схеме:



Передача данных происходит в следующей последовательности событий:

1. Ведомое устройство ожидает получения данных с WREADY, установленным на high в такте n.
2. Мастер передает данные на шину WDATA и утверждает значение WVALID в такте n + 2.
3. Подтверждение связи завершается на переднем фронте такта n +3

Наконец, подчиненное устройство использует канал ответа на запись (B), чтобы подтвердить, что транзакция записи завершена после получения всех WDATA. Этот ответ показан на следующей диаграмме:



Ответ на запись имеет следующую последовательность событий:

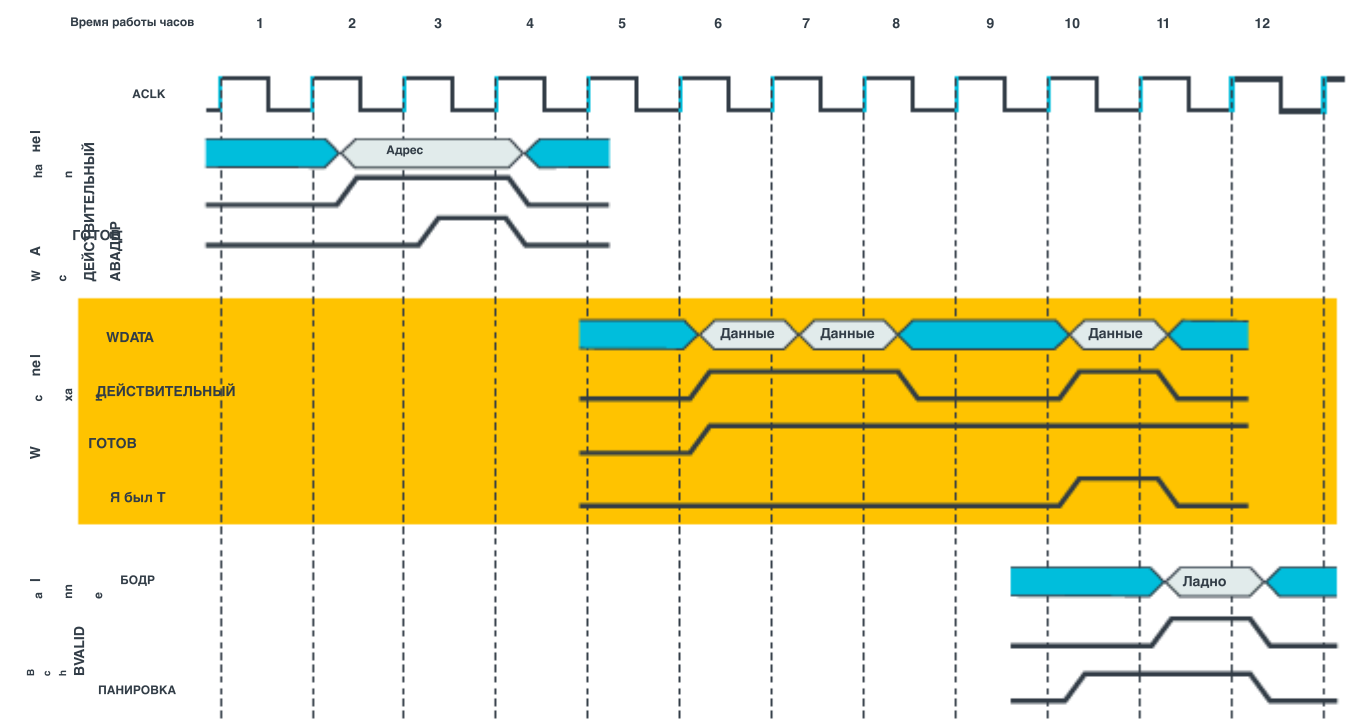
1. Ведущий утверждает BREADY.
2. Подчиненный запускает BRESP, чтобы указать успех или неудачу транзакции записи, и утверждает BVALID.

Подтверждение связи завершается на переднем крае такта n + 3.

4.5 Транзакция записи: несколько элементов данных.

AXI - это пакетный протокол, который означает, что за одну транзакцию можно передавать несколько данных. Мы можем передавать один адрес по каналу AW для передачи нескольких данных с соответствующей информацией о ширине и длине пакета.

На следующей диаграмме показан пример многократной передачи данных:



В этом случае канал AW указывает последовательность из трех передач, а на канале W мы видим три передачи данных. Ведущее устройство увеличивает значение WLAST, чтобы указать конечные данные WDATA. Это означает, что ведомое устройство может либо подсчитывать передачи данных, либо просто отслеживать WLAST. Как только все передачи WDATA получены, ведомое устройство выдает единственное значение BRESP на канале B. Один одиночный BRESP покрывает весь пакет. Если подчиненное устройство решает, что какая-либо из передач содержит ошибку, оно должно дождаться завершения всего пакета, прежде чем сообщить ведущему устройству о произошедшей ошибке.

4.6 Транзакция чтения: отдельный элемент данных.

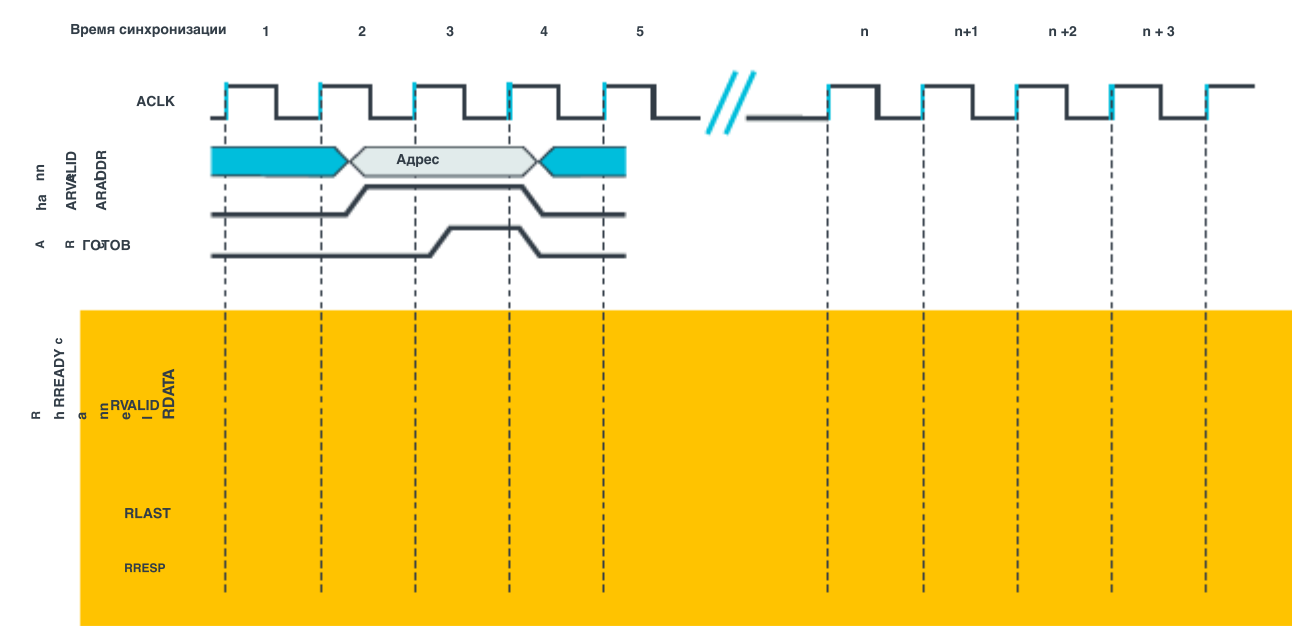
В этом разделе подробно рассматривается процесс транзакции чтения для одного элемента данных и различные каналы, используемые для завершения транзакции.

Эта сделка написать включает в себя следующие каналы:

- Чтение адреса

(Арканзас) Чтение (R)

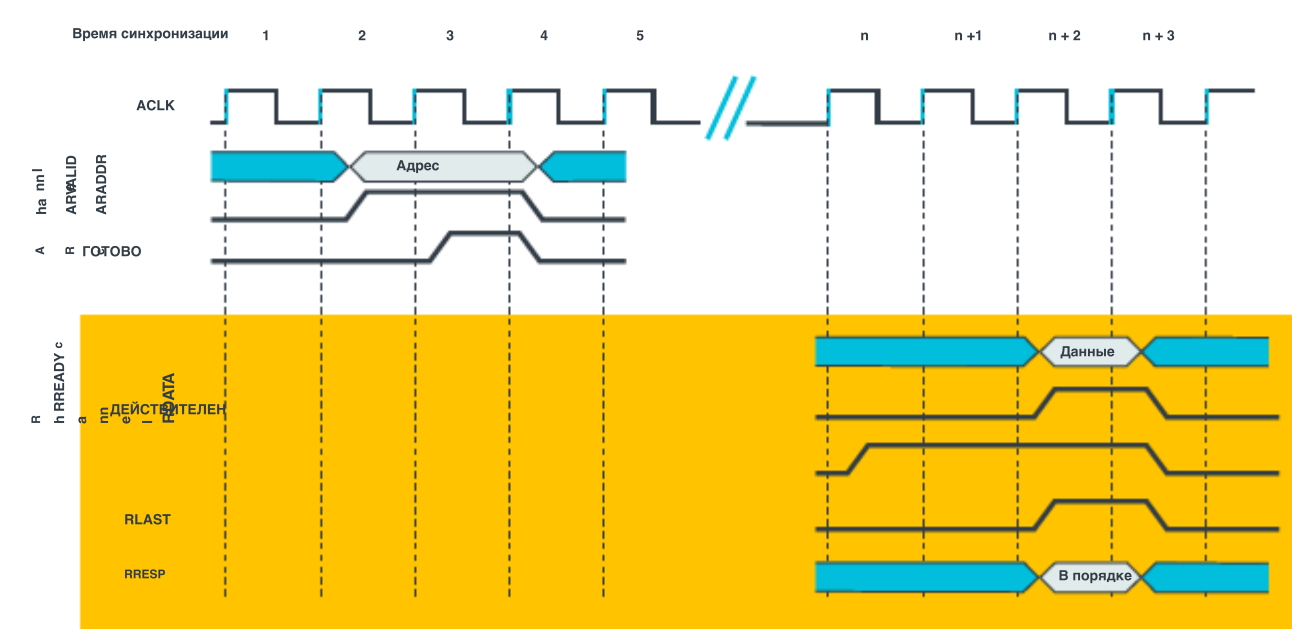
Сначала происходит подтверждение связи по каналу адреса чтения (AR), как показано на следующей диаграмме:



Рукопожатие происходит в следующей последовательности событий: 1. В такте 2 ведущий передает адрес считанного файла ведомому устройству по ARADDR и утверждает ARVALID. 2. В такте 3 подчиненное устройство утверждает ARREADY, чтобы указать, что оно готово принять адрес значение.

Подтверждение связи завершается на переднем фронте такта 4.

Затем по каналу считывания (R) ведомое устройство передает данные ведущему устройству. На следующей диаграмме показан процесс передачи данных.:

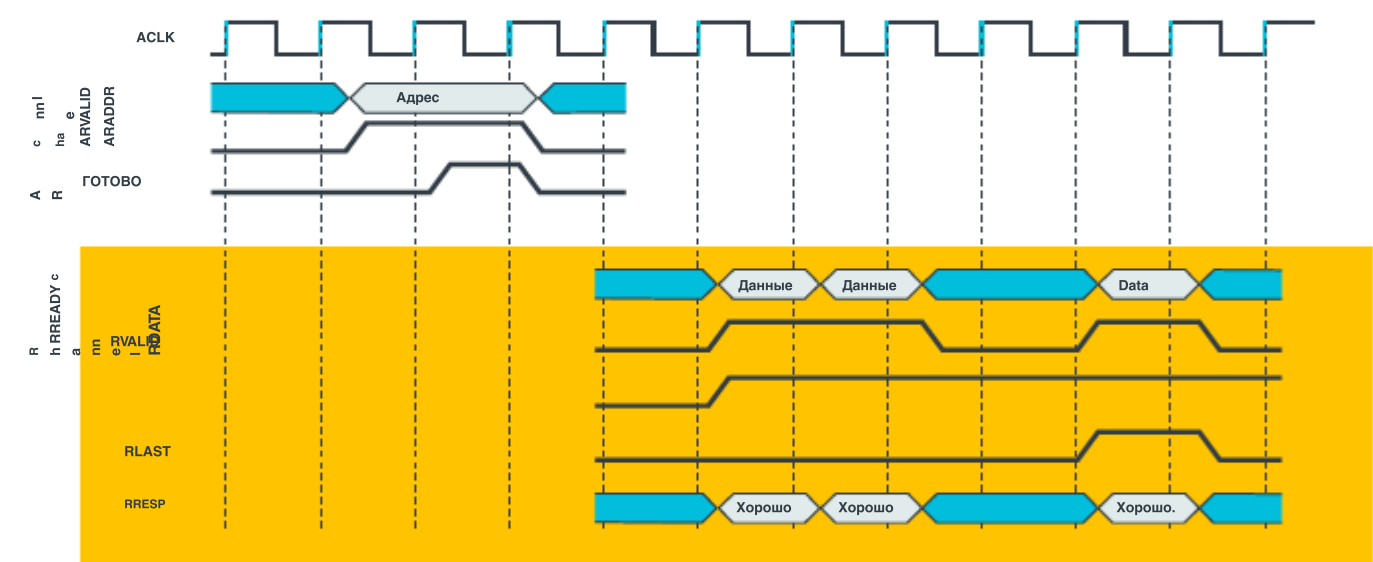


Подтверждение связи при передаче данных происходит в следующей последовательности событий: 1. В такте n ведущее устройство указывает, что оно ожидает получения данных, утверждая RREADY. В ведомое устройство извлекает данные и помещает их в RDATA в такте n + 2. В этом случае, поскольку это единственная транзакция данных, подчиненное устройство также устанавливает сигнал RLAST на высокий уровень. В то же время подчиненное устройство использует RRESP, чтобы сообщить ведущему об успехе или неудаче транзакции чтения и утверждает RVALID. 3. Поскольку RREADY уже подтвержден ведущим, рукопожатие завершается на восходящем фронте такта n + 3.

4.7 Транзакция чтения: несколько элементов данных.

Протокол AXI также допускает пакет чтения при многократной передаче данных в рамках одной транзакции. Это похоже на пакет записи, который описан в разделе Транзакция записи: множественные элементы данных .

На следующей диаграмме показан пример пакетной передачи данных для чтения:



В этом примере мы передаем один адрес по AR-каналу для передачи нескольких элементов данных с соответствующей информацией о ширине и длине пакета. Здесь канал AR указывает последовательность из трех передач, следовательно, на канале R мы видим три передачи данных от ведомого устройства к ведущему. По каналу R подчиненное устройство передает данные ведущему устройству. В этом примере ведущее устройство ожидает данных, как показано параметром RREADY, установленным на high. Ведомое устройство передает действительные RDATA и утверждает RVALID для каждой передачи. Одно из различий между транзакцией чтения и транзакцией записи заключается в том, что для транзакции чтения существует ответ RRESP для каждой передачи в транзакции. Это происходит потому, что в транзакции записи подчиненное устройство должно отправить ответ в виде отдельной передачи по каналу B. В транзакции чтения подчиненное устройство использует тот же канал для отправки данных обратно ведущему устройству и для указания состояния операции чтения .

Если для любого из переводов в транзакции указана ошибка, все равно должна быть завершена полная указанная продолжительность транзакции. Такого понятия, как досрочное пакетное завершение, не существует.

4.8 Активные транзакции.

Активные транзакции также известны как незавершенные транзакции.

Активная транзакция чтения - это транзакция, для которой адрес чтения был передан, но данные последнего чтения еще не были переданы на текущий момент времени.

При чтении данные должны поступать после адреса, поэтому существует простая точка отсчета для того, когда начнется транзакция. Это показано на следующей диаграмме:

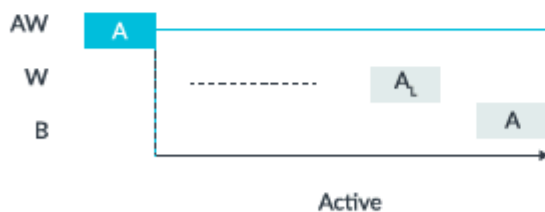


Для транзакций записи данные могут идти после адреса, но также допускаются данные записи в начале. Следовательно, начало транзакции записи может быть любым из следующих: *

- Передача адреса записи Передача основной информации о записи

Следовательно, активная транзакция записи - это транзакция, для которой адрес записи или начальные данные записи были переданы, но ответ на запись еще не был передан.

Следующая диаграмма показывает активную транзакцию записи, где пишут адрес, было передано, но написать ответ до сих пор не переданы:



Следующая диаграмма показывает активную транзакцию записи, где записи, как данные были переданы, но написать ответ до сих пор не переданы:



5-канальные сигналы

В этом разделе представлены основные сигналы и атрибуты AXI и объясняется, как они используются для повышения производительности системы. Протокол AXI определяет пять каналов: три для сигналов записи и два для сигналов чтения.

5.1 Сигналы канала записи

Для транзакции записи используются

следующие каналы:

- Адрес записи
- Ответ на запись
- Запись данных

В следующей таблице показаны сигналы канала адреса записи:

Сигналы канала адреса записи (AW)	Версия AXI
AWVALID	АКСИ3 и АКСИ4
AWREADY	АКСИ3 и АКСИ4
AWADDR[31:0]	АКСИ3 и АКСИ4
AWSIZE[2:0]	АКСИ3 и АКСИ4
AWBURST[1:0]	АКСИ3 и АКСИ4
AWCACHE[3:0]	АКСИ3 и АКСИ4
AWPROT[2:0]	АКСИ3 и АКСИ4
AWID[x:0]	АКСИ3 и АКСИ4
AWLEN[3:0]	только АКСИ3
AWLEN[7:0]	только АКСИ4
AWLOCK[1:0]	только АКСИ3
AWLOCK	только АКСИ4
AWQOS[3:0]	только АКСИ4
AWREGION[3:0]	только АКСИ4
AWUSER[x:0]	только АКСИ4

В следующей таблице показаны сигналы канала записи данных:

Сигналы канала записи данных (W)	Версия AXI
WVALID	AXI3 и AXI4
WREADY	AXI3 и AXI4
последняя	AXI3 и AXI4
WDATA[x:0]	AXI3 и AXI4
WSTRB[x:0]	AXI3 и AXI4
WID[x:0]	Только AXI3

Запись сигналов канала	Версия AXI
WUSER [x: 0]	Только для AXI4

В следующей таблице показаны сигналы канала ответа на запись:

Сигналы канала ответа на запись (B)	Версия AXI
BVALID	AXI3 и AXI4
BREADY	AXI3 и AXI4
BRESP[1:0]	AXI3 и AXI4
СТАВКА [x: 0]	AXI3 и AXI4
ЗАГРУЗЧИК [x: 0]	Только AXI4

Все сигналы в каждом канале имеют один и тот

же префикс, для канала адреса записи

- W для канала записи данных
- B для канала ответа на запись

Существуют некоторые различия между протоколами AXI3 и AXI4 для каналов записи:

- Для канала адреса записи сигнал AWLEN шире для протокола AXI4. Следовательно, AXI4 способен генерировать более длинные пакеты, чем AXI3.
- AXI4 уменьшает сигнал AWLOCK до одного бита, чтобы обеспечить только эксклюзивные передачи, поскольку заблокированные передачи не поддерживаются.
- AXI4 добавляет сигнал AWQOS в канал AW. Этот сигнал поддерживает концепцию качества обслуживания (QoS) в протоколе AXI4.
- AXI4 добавляет сигнал AWREGION к каналу AW. Этот сигнал поддерживает подчиненные области, которые допускают несколько логических интерфейсов из одного физического подчиненного интерфейса.
- AXI4 удаляет сигнал WID из канала W. Это связано с тем, что изменение порядка записи данных больше не допускается.
- AXI4 добавляет определяемые пользователем сигналы к каждому каналу.

5.2 Сигналы канала чтения

Для транзакции чтения используются

- следующие каналы:
- Адрес для чтения
 - Данные для чтения

В следующей таблице показаны сигналы канала с адресом считывания:

Сигналы канала с адресом	Версия AXI
ARVALID	AXI3 и AXI4
AREADY	AXI3 и AXI4
ARADDR[31:0]	AXI3 и AXI4

Считывание сигналов канала адреса	Версия AXI
ARSIZE[2:0]	АКСИ3 и АКСИ4
ARBURST[1:0]	АКСИ3 и АКСИ4
АРКАШ[3:0]	АКСИ3 и АКСИ4
ARPROT[2:0]	АКСИ3 и АКСИ4
ARID[x:0]	АКСИ3 и АКСИ4
АРЛЕН[3:0]	только АКСИ3
АРЛЕН[7:0]	только АКСИ4
АРЛОК[1:0]	только АКСИ3
АРЛОК	только АКСИ4
ARQOS[3:0]	только АКСИ4
РЕГИОН[3:0]	только АКСИ4
ARUSER[x:0]	только АКСИ4

В следующей таблице показаны сигналы канала считывания данных:

Сигналы канала считывания данных (R)	Версия AXI
RVALID	AXI3 и AXI4
ГОТОВЫ	AXI3 и AXI4
RLAST	AXI3 и AXI4
RDATA[x:0]	AXI3 и AXI4
RRESP[1:0]	AXI3 и AXI4
RID[x:0]	AXI3 и AXI4
RUSER[x:0]	Только AXI4

Все сигналы в каждом канале имеют одинаковый

префикс:

AR для канала считывания

адреса * R для канала считывания данных

Существуют некоторые различия между протоколами AXI3 и AXI4 для каналов считывания:

- Для протокола AXI4 сигнал ARLEN длины адреса чтения шире. Следовательно, AXI4 способен генерировать более длинные пакеты чтения, чем AXI3.
- AXI4 снижает ARLOCK сигнал на один бит, чтобы разместиться только эксклюзивные переводы, потому что заблокированы переводы не поддерживаются. Как с
- написать сигналы канала, концепции качества обслуживания и подчиненных регионов применимы к транзакциям чтения. В них используются сигналы ARQOS и ARREGION в канале
- AR. AXI4 добавляет определяемые пользователем сигналы к двум каналам считывания.

5.3 Размер, длина и тип пакета данных

Каждая транзакция чтения и записи имеет атрибуты, которые определяют длину данных, размер и пакетный сигнал атрибуты для этой транзакции.

В следующем списке атрибутов x обозначает запись и чтение, поэтому они применяются как к каналу адреса записи , так и к каналу адреса чтения:

- Значение AxLEN описывает длину транзакции в количестве передач.

- Для AXI3 значение AxLEN[3: 0] имеет 4 бита, что определяет диапазон от 1 до 16 передач в транзакции. Для AXI4 AxLEN[7:0] имеет 8 бит, что определяет диапазон от 1 до 256 передач данных в транзакции. AxSize[2:0] описывает максимальное количество байт для передачи при каждой передаче данных. Три бита кодировки обозначают 1, 2, 4, 8, 16, 32, 64, или 128 байт на передачу. AxBURST[1:0] описывает пакетный тип транзакции: фиксированный, увеличивающий или завершающий.

В следующей таблице показаны различные свойства этих типов пакетов:

Значение	Тип пакета	Примечания по использованию	Длина (количество передач)	Выравнивание
0x00	ИСПРАВЛЕНО	Повторное считывание одного и того же адреса. Полезно для FIFOs.	1-16	Только фиксированные байтовые полосы, определяемые начальным адресом и размером.
0x01	INCR	Увеличивающийся пакет. Подчиненное устройство увеличивает адрес для каждой передачи в пакете по сравнению с адресом для предыдущей передачи. Добавочное значение зависит от размера передачи, определенного атрибутом AxSIZE. Полезно для блочных передач.	АКСИ3: 1-16 АКСИ4: 1-256	Поддерживаются невыровненные передачи.
0x10	ПЕРЕНОС	Пакетное перенос. Аналогично увеличивающемуся пакету, за исключением того, что при достижении верхнего предела адреса адрес переносится на нижний адрес. Обычно используется для обращений к строке кэша.	2, 4, 8 или 16	Начальный адрес должен быть выровнен по размеру передачи.
0x11	ЗАРЕЗЕРВИРОВАНО	Не для использования.	-	-

Поддержка уровня защиты 5.4

AXI предоставляет сигналы разрешений доступа AWPROT и ARPROT, которые могут защитить от незаконных транзакций ниже по потоку в системе. Например, если транзакция не имеет надлежащего уровня защиты, контроллер памяти может отказать в доступе на чтение или запись, используя эти сигналы.

Это полезно для решений безопасности, таких как Arm TrustZone, где процессор имеет два отдельных состояния: Безопасное и небезопасное.

AxPROT определяет три уровня защиты доступа, как показано на следующей схеме:



Распределение битов AxPROT определяет следующие

атрибуты:

- AxPROT[0] (P) определяет доступ как непривилегированный или привилегированный:

- 1 указывает на привилегированный доступ.
 - 0 указывает на непривилегированный доступ.

Хотя некоторые процессоры поддерживают несколько уровней привилегий, единственное различие, которое может обеспечить AXI, - это привилегированный и непривилегированный

- доступ. AxPROT[1] (NS) определяет доступ как безопасный или небезопасный:

- 1 указывает на небезопасную транзакцию.
 - 0 указывает на безопасную транзакцию.

- AxPROT[2] (I) указывает, является ли транзакция доступом по инструкции или доступом к данным:

- 1 указывает доступ по инструкции.
 - 0 указывает доступ к данным.

Протокол AXI определяет это указание как подсказку.

Это неточно во всех случаях, например, когда транзакция содержит комбинацию инструкций и элементов данных. Спецификация Arm AXI как для AXI 3,

так и для AXI 4 рекомендует, чтобы мастер устанавливал бит 2 равным нулю для указания доступа к данным, если только конкретно не известно, что доступ является доступом по инструкции.

5.5 Поддержка кэша

Современные SoC-системы часто содержат кэши, размещенные в нескольких точках системы. Например, кэш уровня 2 может быть внешним по отношению к процессору, или кэш уровня 3 может находиться перед контроллером памяти. Для поддержки систем, использующих разные политики кэширования, сигналы AWCACHE и ARCACHE указывают, как транзакции должны выполняться в системе.

На следующей диаграмме показано распределение битов AxCACHE:



Распределение битов AxCACHE определяет следующие атрибуты:

- АхCACHЕ [0] (В) - это буферный бит.

Когда этот бит установлен в 1, межсоединение или любой компонент может задержать транзакцию, достигающую конечного пункта назначения, на любое количество циклов.

Буферный бит указывает, может ли ответ прийти из промежуточной точки, или должен ли ответ прийти от подчиненного устройства назначения.

- АхCACHЕ [1] - это кэшируемый бит в AXI3 или изменяемый бит в AXI4.

Этот бит указывает, что атрибуты транзакции в конечном пункте назначения не обязательно должны совпадать с атрибутами исходной транзакции.

Для операций записи установка изменяемого бита означает, что несколько разных операций записи могут быть объединены, или одна запись может быть разбита на несколько транзакций.

Для операций чтения установка изменяемого бита означает, что содержимое местоположения может быть предварительно выбрано, или значения из одной выборки могут использоваться для нескольких

- транзакций чтения. АхCACHЕ [2] - это бит RA.

Бит RA указывает, что при чтении распределение транзакции рекомендуется, но не обязательно.

Если утверждается либо АхCACHЕ [2], либо АхCACHЕ [3], то транзакцию необходимо искать в кэше, поскольку она могла быть выделена в этом кэше другим ведущим

- устройством. АхCACHЕ [3] - это бит WA.

Бит WA указывает, что при записи распределение транзакции рекомендуется, но не обязательно. Если утверждается

либо АхCACHЕ [2], либо АхCACHЕ [3], то транзакция должна быть найдена в кэше, поскольку она могла быть выделена в этом кэше другим мастером.



Если АхCACHЕ [1], кэшируемый бит, не указан, то АхCACHЕ [2] и АхCACHЕ [3] не могут быть указаны.

Причина включения распределения функций чтения и записи как на шинах адресов чтения, так и записи заключается в том, что это позволяет системному кэшу оптимизировать свою производительность.

Например, рассмотрим кэш, в котором доступ на чтение определяется как "выделение для записи, но не для чтения

". В этом случае кэш знает, что адрес может храниться в кэше, потому что он мог

быть выделен при предыдущей записи, и поэтому он должен выполнить поиск в кэше.

Однако теперь учтите, что кэш видит доступ на чтение, который определен как "no write-allocate и no read-allocate". В этом случае кэш знает, что адрес не был выделен в кэше.

Кэш может избежать поиска и немедленно передать транзакцию другой стороне. Кэш

может делать это только в том случае, если он знает как о выделении времени для чтения, так и для записи для

каждой транзакции. Не обязательно, чтобы кэши работали таким образом, но протокол AXI определен с помощью

RA и WA как для чтения, так и для записи, чтобы разрешить этот режим работы, если вы или ваш разработчик кэша хотите реализовать его.

5.6 Сигнализация ответа

AXI обеспечивает сигнализацию ответа как для транзакций чтения, так и для транзакций записи.

Для транзакций чтения информация об ответе от ведомого устройства передается по каналу считывания данных с использованием RRESP.

Для транзакций записи информация об ответе передается по каналу ответа на запись с использованием BRESP. RRESP

и BRESP оба состоят из двух битов, и кодирование этих сигналов может передавать четыре ответа, как показано в следующей таблице:

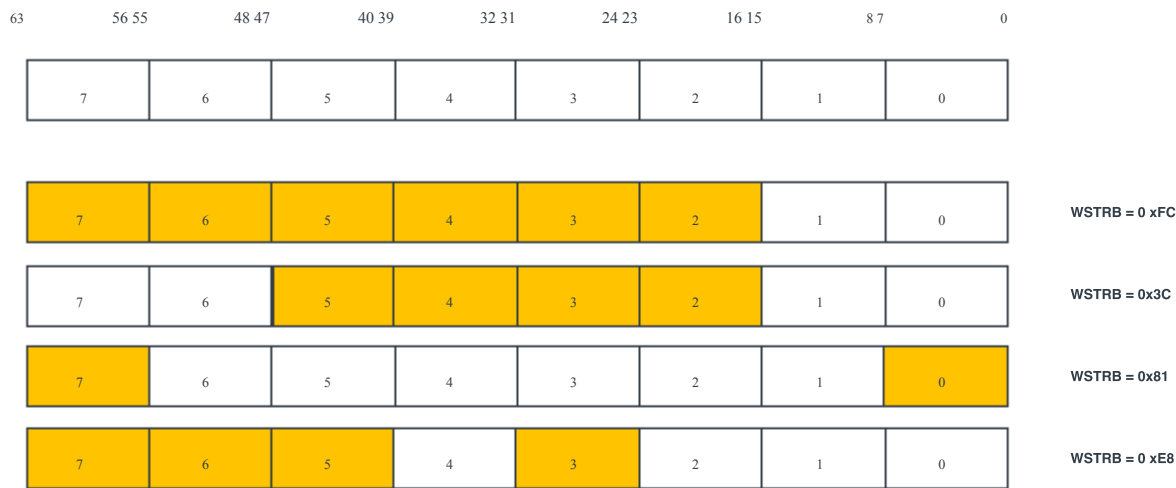
Код ответа	Описание
00 - OKAY	Обычный успешный доступ или сбой эксклюзивного доступа. OKAY - это ответ, который используется для большинства транзакций. OKAY указывает, что обычный доступ был успешным. Этот ответ также может указывать на то, что произошел сбой эксклюзивного доступа. Эксклюзивный доступ - это когда более одного ведущего устройства могут получить доступ к ведомому устройству одновременно, но эти ведущие устройства не могут получить доступ к одному и тому же диапазону памяти.
01 - EXOKAY	Эксклюзивный доступ в порядке. EXOKAY указывает, что либо чтение, либо запись части эксклюзивного доступа были успешными.
10 - SLVERR	Ошибка подчиненного устройства. SLVERR используется, когда доступ к ведомому устройству получен успешно, но ведомое устройство хочет вернуть исходящему ведущему устройству сообщение об ошибке. Это указывает на неудачную транзакцию. Например, при попытке передачи неподдерживаемого размера или попытке доступа на запись в папку, доступную только для чтения.
11 - DECERR	Ошибка декодирования. DECERR часто генерируется компонентом межсоединения, чтобы указать, что по адресу транзакции нет подчиненного устройства.

5.7 Стробоскопы записи данных

Стробоскопический сигнал записи данных используется ведущим устройством для указания ведомому устройству, какие байты шины данных требуются. Стробоскопы записи данных полезны при обращении к кэшу для эффективного перемещения разреженных массивов данных . В дополнение к использованию стробоскопов записи данных, вы можете оптимизировать передачу данных, используя невыровненные начальные адреса.

Канал записи имеет один стробоскопический бит на байт шины данных. Эти биты формируют сигнал WSTRB. Мастер должен убедиться, что стробоскопы записи установлены на 1 только для байтовых дорожек, содержащих действительные данные. Например, рассмотрим 64-разрядную шину записи данных. Сигнал WSTRB состоит из 8 битов, по одному на каждый байт. На следующей диаграмме показано, как примерные значения WSTRB определяют, какие байтовые полосы допустимы.:

Для 64-битной шины WDATA



Рассматривая первый пример, мы предполагаем, что допустимые данные находятся только в шести верхних значащих байтах шины данных, от байта 7 до байта 2. Это означает, что ведущее устройство должно управлять сигналом WSTRB с помощью шестнадцатеричного значения 0xFC.

Аналогично, в остальных примерах указаны допустимые байтовые полосы шины данных следующим образом:

- Допустимые данные только в байтах 2, 3, 4 и 5 шины данных требуют значения сигнала WSTRB, равного 0x3C.
- Допустимые данные только в байтах 0 и 7 шины данных требуют значения сигнала WSTRB, равного 0x81.
- Допустимые данные только в байтах 3, 5, 6 и 7 шины данных требуют значения сигнала WSTRB, равного 0xE8.

Стробоскопы байтовых полос обеспечивают эффективное перемещение разреженных массивов данных. Используя этот метод, транзакции записи могут быть досрочно завершены путем установки стробоскопов оставшихся байтовых полос передачи в 0, хотя оставшиеся передачи все равно должны быть завершены. Сигнал WSTRB также может меняться между переводами в транзакции.

Эквивалентного сигнала для канала считывания нет. Это связано с тем, что ведущий указывает на требуемую передачу и может замаскировать любые нежелательные байты, полученные от ведомого устройства.

5.8 Атомарный доступ осуществляется с помощью сигнала блокировки.

Сигнал блокировки оси используется для указания того, когда выполняются атомарные обращения. Смотрите раздел Атомарные обращения для получения дополнительной информации и объяснения концепции и функционирования эксклюзивного доступа передачи.

Протокол AXI предоставляет два механизма для поддержки атомарности:

- Заблокированный доступ

Заблокированная передача блокирует канал, который остается заблокированным до тех пор, пока не будет сгенерирована разблокированная передача. Заблокированный доступ аналогичен механизму, поддерживаемому протоколом AHB.

Когда ведущее устройство использует сигналы AxLOCK для транзакции, чтобы показать, что это заблокированная транзакция, тогда межсоединение должно гарантировать, что только это ведущее устройство может получить доступ к целевой подчиненной области,

до завершения разблокированной транзакции от того же ведущего сервера. Арбитр в интерконнекте должен обеспечить соблюдение этого ограничения. Поскольку заблокированный доступ требует, чтобы межсоединение предотвращало любые другие транзакции, происходящие во время выполнения заблокированной последовательности, они могут оказывать важное влияние на производительность межсоединения.

Заблокированные транзакции следует использовать только для устаревших устройств. Только AXI3 поддерживает заблокированный доступ. AXI4 не поддерживает заблокированный доступ.

- **Эксклюзивные обращения** Эксклюзивные обращения более эффективны, чем заблокированные транзакции, и они позволяют нескольким ведущим устройствам обращаться к ведомому устройству одновременно.

Механизм эксклюзивного доступа позволяет выполнять операции семафорного типа, не требуя, чтобы шина оставалась привязанной к определенному ведущему устройству во время операции. Поскольку заблокированный доступ не так эффективен, как эксклюзивный, и большинство компонентов не требуют заблокированных транзакций, они были удалены из протокола AXI4.

В AXI3 сигнал AxLOCK состоит из двух битов со следующими

значениями: • - Обычный

0b00

- - Эксклюзивный

0b01

- - Заблокированный 0b10

- - Зарезервировано 0b11

В AXI4 сигнал блокировки оси состоит из одного бита со следующими

значениями: • - Нормальный

0b0

5.9 Качество обслуживания

- - Эксклюзивно 0b1

Протокол AXI4 вводит дополнительные сигналы для поддержки качества обслуживания (QoS).

Качество обслуживания позволяет вам расставлять приоритеты в транзакциях, что позволяет повысить производительность системы, обеспечивая более приоритетное выполнение более важных транзакций.

Существуют два сигнала качества обслуживания:

- AWQOS отправляется по каналу адреса записи для каждой транзакции записи.
- ARQOS отправляется по каналу адреса чтения для каждой транзакции чтения.

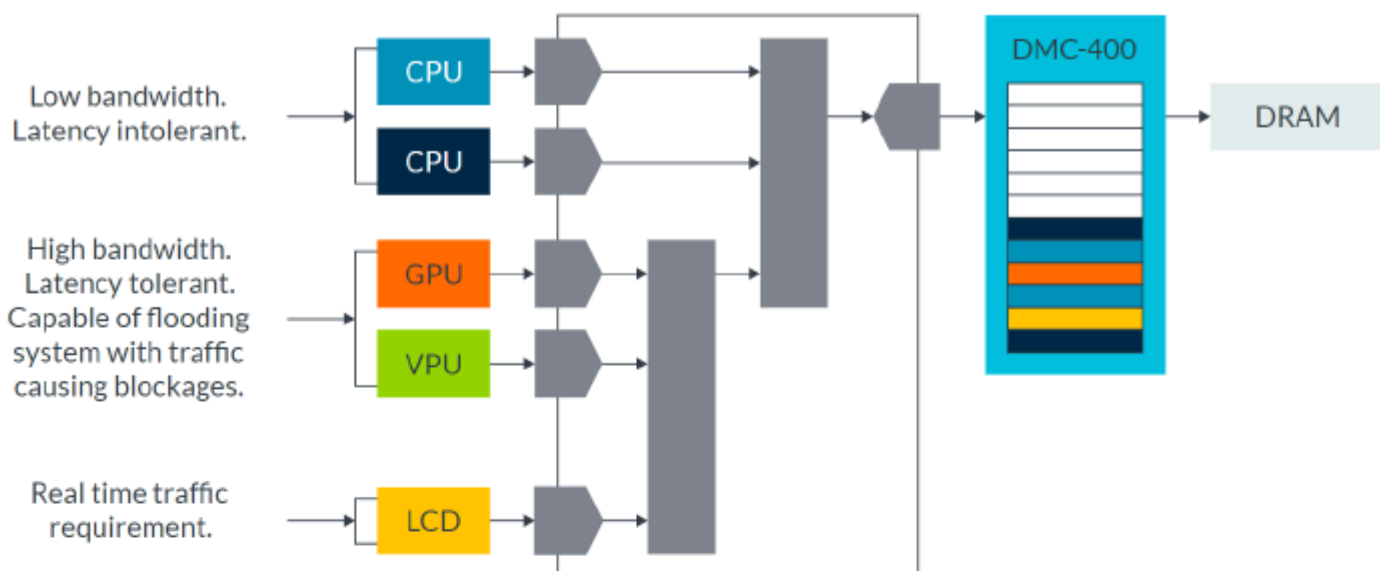
Оба сигнала имеют ширину 4 бита, где значение 0x0 указывает на самый низкий приоритет, а значение 0xF указывает на самый высокий приоритет. Реализация качества

обслуживания на системном уровне по умолчанию заключается в том, что любой компонент с

возможностью выбора из более чем одной транзакции сначала обрабатывает транзакцию с более высоким

значением QoS. На следующей схеме показан пример системы с контроллером прямой памяти (DMC), в

частности DMC-400. Этот контроллер управляет транзакциями в DRAM:



На практике некоторые элементы, такие как центральный процессор, требуют доступа к памяти, который гораздо важнее, чем доступ к другим компонентам, таким как графический процессор или VPU.

Когда транзакциям присваиваются соответствующие значения QoS, интерконнект может рассматривать транзакцию с более высоким приоритетом перед транзакциями с более низким приоритетом, и DMC переупорядочивает транзакции, чтобы гарантировать, что присвоен правильный приоритет.

5.10 Региональная сигнализация

Региональная сигнализация - это новая дополнительная функция в AXI4.

Когда вы используете идентификаторы регионов, это означает, что один физический интерфейс на ведомом устройстве может предоставлять несколько логических интерфейсов. Каждый логический интерфейс может иметь разное местоположение в системной адресной карте.

Когда используется идентификатор региона, ведомому устройству не нужно поддерживать декодирование адреса между различными логическими интерфейсами.

Региональная сигнализация использует два 4-разрядных идентификатора региона, AWREGION и ARREGION. Эти региональные идентификаторы могут однозначно идентифицировать до 16 различных регионов.

5.11 Сигналы пользователя.

Набор сигналов интерфейса AXI4 имеет возможность включать набор определяемых пользователем сигналов, называемых пользовательскими сигналами. Пользовательские сигналы могут использоваться на каждом канале для передачи дополнительной пользовательской управляющей информации между ведущим и ведомым компонентами. Эти сигналы являются необязательными и не обязательно должны поддерживаться на всех каналах. Если они используются, то ширина сигналы пользователя, определяется реализацией и может быть различным на каждом канале.



Поскольку протокол AXI не определяет функции этих пользовательских сигналов, проблемы взаимодействия могут возникнуть, если два компонента используют одни и те же пользовательские сигналы несовместимым образом.

5.12 Зависимости каналов AXI.

Протокол AXI определяет зависимости между различными каналами.

Ниже приведены три основных зависимости.:

- Передача WLAST должна завершиться до утверждения BVALID.
 - Ведущий сервер должен отправить все данные для записи, прежде чем ведущий сервер сможет увидеть ответ на запись.

Эта зависимость не существует в AXI3, но введена для AXI4:

В AXI3 адрес не обязательно должен быть виден перед отправкой ответа на запись.

В AXI4 все данные и адрес должны быть переданы до того, как мастер сможет увидеть ответ на запись.

- RVALID не может быть подтвержден до тех пор, пока не будет передан ARADDR.

Ведомое устройство не может передать какие-либо прочитанные данные без предварительного просмотра адреса. Это связано с тем, что ведомое устройство не может отправить данные обратно ведущему устройству, если оно не знает адрес, с которого будут считаны данные.

- WVALID может быть установлен перед AWVALID.
 - Ведущий может использовать канал записи данных для отправки данных ведомому устройству, прежде чем передавать адрес, куда ведомое устройство должно записать эти данные.

6 атомарных доступов

Атомарный доступ - это термин, обозначающий серию обращений к области памяти. Атомарные обращения используются главными устройствами, когда они хотели бы выполнить последовательность обращений к определенной области памяти, при этом будучи уверенными, что исходные данные в этой области не повреждены записями из других основных устройств. Эта последовательность обычно представляет собой последовательность чтения, изменения и записи. Существует два типа атомарных доступов:

- **Заблокированный**

Пока ведущее устройство выполняет последовательность транзакций с заблокированными обращениями, обращения от любых других ведущих устройств к тому же ведомому устройству отклоняются.

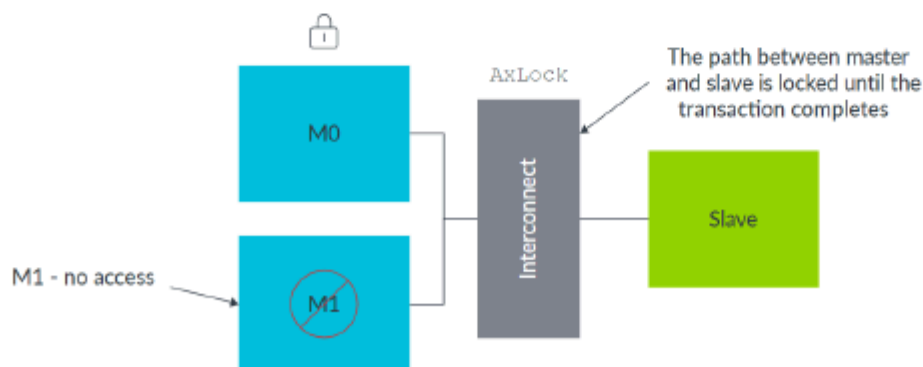
- **Эксклюзивный** Когда ведущий сервер успешно выполняет последовательность транзакций с исключительными правами доступа, другие ведущие устройства могут получить доступ к ведомому устройству, но не к области памяти, к которой осуществляется доступ.

6.1 Заблокированные права доступа

Заблокированные транзакции следует использовать только для устаревших устройств. AXI4 не поддерживает заблокированные транзакции, но реализации AXI3 должны поддерживать заблокированные транзакции.

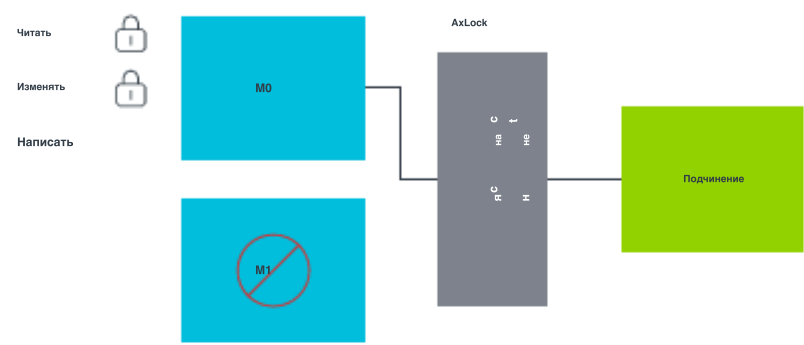
Прежде чем ведущий сервер сможет запустить заблокированную последовательность транзакций, он должен убедиться, что у него нет других транзакций, ожидающих завершения. Транзакция с установленным сигналом **AxLOCK** указывает на заблокированную транзакцию. Заблокированная последовательность транзакций вынуждает интерконнект отклонять доступ к подчиненному устройству от любых других ведущих устройств.

Заблокированная последовательность всегда должна завершаться окончательной транзакцией, для которой не установлен сигнал **AxLOCK**. Эта заключительная транзакция по-прежнему включена в заблокированную последовательность, но эффективно снимает блокировку, позволяя другим ведущим устройствам получить доступ к ведомому устройству. Поскольку заблокированный доступ требует, чтобы межсоединение предотвращало выполнение любых других транзакций во время выполнения заблокированной последовательности, они оказывают важное влияние на производительность межсоединения. Следующая диаграмма показывает акси заблокирован доступ к работе с пример используя двум господам, M0 и M1:



Прежде чем ведущее устройство сможет запустить заблокированную последовательность транзакций, ведущее устройство должно убедиться, что у него нет других транзакций, ожидающих завершения. Когда M0 использует сигнал блокировки для транзакции, чтобы указать, что это заблокированная транзакция, то межсоединение использует арбитра, чтобы гарантировать, что только M0 может получить доступ к целевому подчиненному устройству. Межсоединение блокирует любые обращения с M1 до завершения разблокированной транзакции с M0.

На следующей диаграмме показано, как заблокированный доступ работает с последовательностью транзакций:



Шаги в этом примере следующие:

1. Мастер M0 инициирует последовательность ЧТЕНИЯ, ИЗМЕНЕНИЯ и ЗАПИСИ.

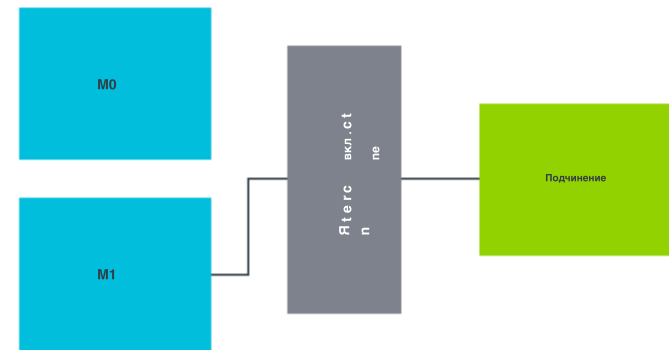
Для первой транзакции READ утверждается сигнал БЛОКИРОВКИ, указывающий на то, что она запускает заблокированную транзакцию.

2. Интерконнект блокирует любые другие транзакции.

С этого момента ведущий M1 не имеет доступа к ведомому устройству.

3. Последняя транзакция в последовательности, WRITE, не имеет подтвержденного сигнала БЛОКИРОВКИ. Эта транзакция указывает на конец заблокированной последовательности.

Межсоединение снимает блокировку, и другие ведущие устройства теперь могут получить доступ к ведомому устройству.



6.2 Эксклюзивные возможности доступа

В AXI 4 эксклюзивные возможности доступа выполняют атомарные операции более эффективно, чем заблокированные возможности доступа. Это связано с тем, что эксклюзивные возможности доступа более эффективно используют пропускную способность межсоединения. При исключительной последовательности доступа другие ведущие устройства могут одновременно обращаться к ведомому устройству, но только одному ведущему устройству будет предоставлен доступ к тому же диапазону памяти.

Механизм, который используется для эксклюзивного доступа, может обеспечивать операции семафорного типа, не требуя, чтобы шина оставалась выделенной определенному ведущему устройству во время операции. Это означает, что на задержку доступа к шине и максимально достижимую пропускную способность это не влияет.

Эксклюзивный доступ может состоять из более чем одной передачи данных, но все транзакции должны иметь идентичные атрибуты адресного канала.

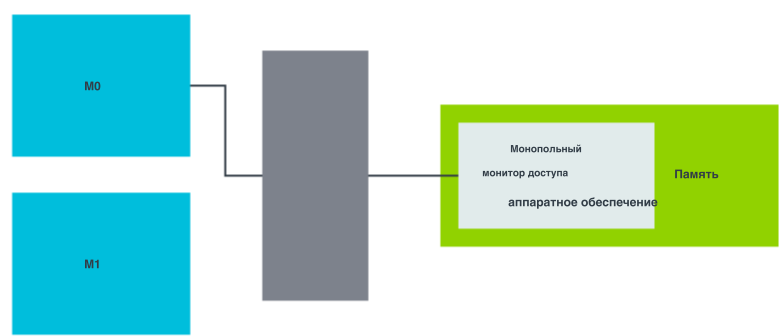
Аппаратный монитор исключительного доступа требуется подчиненному устройству для записи информации о транзакции для исключительной последовательности, чтобы оно знало диапазон памяти, к которому осуществляется доступ, и личность ведущего устройства, выполняющего доступ.

Если ни один другой ведущий сервер не обращается к контролируемому диапазону до завершения эксклюзивной последовательности, доступ является атомарным. Ведомое устройство открыто для доступа от других ведущих устройств, что приводит к общему повышению эффективности использования полосы пропускания для системы.

6.3 Работа аппаратного монитора с эксклюзивным доступом.

Базовый механизм эксклюзивного доступа регулируется монитором эксклюзивного доступа, который вы должны реализовать.

На следующей диаграмме показан пример, в котором ведущее устройство M0 выполняет эксклюзивное чтение с адреса:



Аппаратное обеспечение монитора эксклюзивного доступа выдает один из следующих ответов: • EXOKAY

- Считывается значение, и идентификатор транзакции сохраняется в аппаратном обеспечении монитора эксклюзивного доступа. ХОРОШО

Значение считывается, но исключительный доступ не поддерживается, и мастер должен обработать этот ответ как ошибку для исключающей операции.

Некоторое время спустя, если EXOKAY был получен во время эксклюзивного чтения, M0 пытается завершить эксклюзивную последовательность, выполняя эксклюзивную запись на тот же адрес.

При эксклюзивной записи используется тот же идентификатор транзакции, что и при эксклюзивном чтении. Ответ от аппаратного обеспечения монитора эксклюзивного доступа один из следующих:

- EXOKAY

Ни один другой ведущий сервер не выполнял запись в это хранилище с момента предоставления исключительного доступа на чтение, поэтому запись прошла успешно. В этом случае исключительная запись

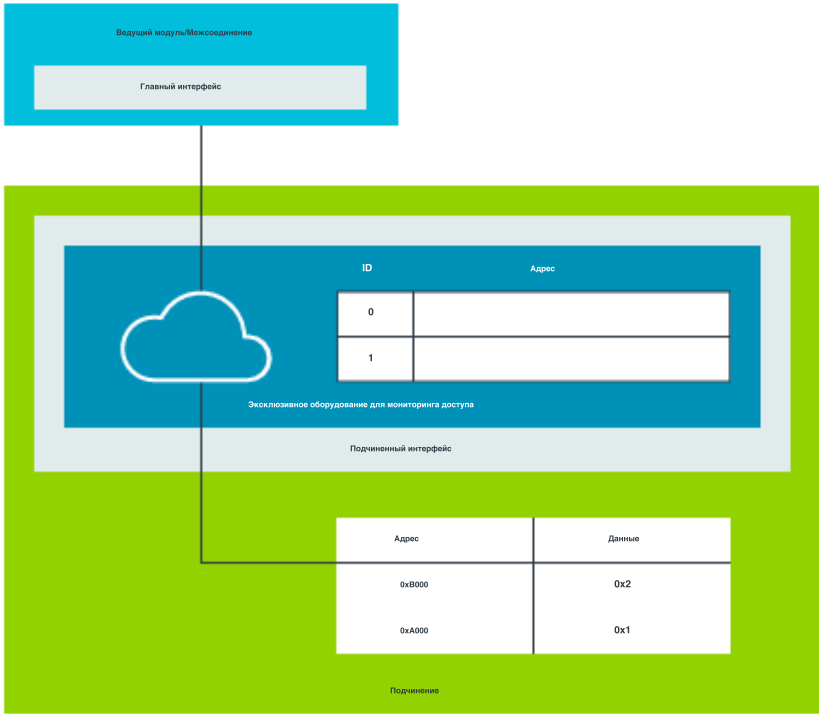
- обновляет память. Хорошо, другой мастер, например M1, произвел запись в папку с момента получения исключительного доступа на чтение, поэтому запись завершается ошибкой. В этом случае ячейка памяти не обновляется.

Некоторым подчиненным устройствам требуется дополнительная логика для поддержки эксклюзивного доступа. Мониторинг эксклюзивного доступа аппаратное обеспечение отслеживает только один адрес для каждого идентификатора транзакции. Это должно быть реализовано так, чтобы оно могло отслеживать каждый возможный эксклюзивный идентификатор, который можно увидеть.

6.4 Пары эксклюзивных транзакций: проходят обе.

В этом разделе описывается пример двух успешных последовательностей эксклюзивного доступа, которые проходят обе.

На следующей схеме показана система, содержащая ведущий модуль с его ведущим интерфейсом AXI и подчиненный модуль:



Ведомый интерфейс включает в себя эксклюзивное оборудование для мониторинга доступа, которое может сохранять идентификатор и адрес, к которому осуществляется доступ для каждой транзакции. В следующей таблице описаны различные транзакции в приведенной в примере последовательности. Все транзакции в таблице имеют исключительный доступ.:

Номер транзакции	Чтение или	Идентификатор	Адрес	Данные	xRESP
1	R	0	0xA000	0x1	EXOKAY
2	R	1	0xB000	0x2	EXOKAY
3	W	0	0xA000	0x3	EXOKAY
4	W	1	0xB000	0x4	EXOKAY

Последовательность транзакций, показанная в предыдущей таблице, выполняется следующим образом:

1. Первая транзакция - ведущая, которая выполняет транзакцию с исключительным правом чтения с идентификатором 0 с адреса 0xA000.

Оборудование для мониторинга эксклюзивного доступа сохраняет идентификатор и адрес этой транзакции в своей таблице, и подчиненное устройство отвечает прочитанными данными, 0x1, поскольку для этого подчиненного устройства корректно поддерживаются эксклюзивные обращения, оборудование для мониторинга эксклюзивного доступа отвечает ответом EXOKAY. 2. Затем мастер выполняет новую транзакцию с эксклюзивным чтением с идентификатором 1 из адреса 0xB000.

Опять же, оборудование для мониторинга эксклюзивного доступа сохраняет детали этой новой транзакции в таблице, и подчиненное устройство отвечает считанными данными, 0x2.

Поскольку эксклюзивные права доступа корректно поддерживаются для этого подчиненного устройства, оборудование для мониторинга эксклюзивного доступа снова отвечает ответом EXOKAY. На данный момент в нашем примере выполняются две отдельные эксклюзивные последовательности. 3. После того, как ведущий сервер завершил свою работу, он выполняет транзакцию с исключением записи с идентификатором 0 для адреса 0xA000.

Аппаратное обеспечение для мониторинга эксклюзивного доступа проверяет детали этой транзакции в таблице и, из-за существующей записи с идентификатором 0 и адресом, отвечает ведущему с помощью ответа EXOKAY. Это означает, что ни одно другое ведущее устройство не обращалось к этой ячейке памяти, и ведомое устройство обновляет ее новым значением, которое оно получает, которое в этом примере равно 0x3.

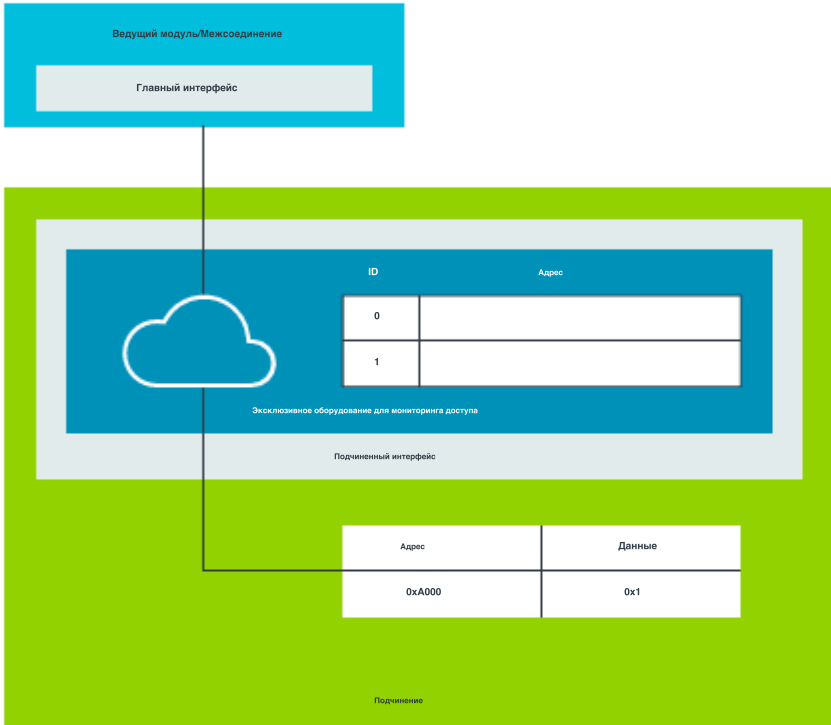
Оборудование для мониторинга эксклюзивного доступа удаляет идентификатор и адрес для этой транзакции из своей таблицы, поскольку последовательность эксклюзивного доступа для этого местоположения адреса теперь завершена. 4. Наконец, мастер выполняет новую транзакцию с правом монопольной записи с идентификатором 1 по адресу 0xB000.

Оборудование для мониторинга монопольного доступа проверяет детали этой транзакции в своей таблице. Увидев существующую запись с идентификатором 1 и адресом 0xB000, она снова отправляет мастеру ответ EXOKAY. Это означает, что ни одно другое ведущее устройство не обращалось к этой ячейке памяти, и ведомое устройство обновляет ее полученным новым значением, которое в нашем примере равно 0x4.

Опять же, оборудование для мониторинга эксклюзивного доступа удаляет идентификатор и адрес для этой транзакции из своей таблицы, поскольку последовательность эксклюзивного доступа для этого местоположения адреса теперь завершена.

6.5 Пары эксклюзивных транзакций: один проход, один сбой.

В этом разделе описывается пример двух последовательностей эксклюзивного доступа, когда первая завершается успешно, а вторая завершается неудачей. На следующей схеме показана система, содержащая ведущий модуль с его ведущим интерфейсом AXI и подчиненный модуль:



Подчиненный интерфейс включает в себя эксклюзивное оборудование для мониторинга доступа, которое может сохранять идентификатор и адрес, к которому осуществляется доступ для каждой транзакции.

В следующей таблице описаны различные транзакции в последовательности примеров. Все транзакции в таблице имеют исключительный доступ:

Номер транзакции	Чтение или запись	Идентификатор транзакции	Адрес	Данные	xRESP
1	R	0	0xA000	0x1	ЭКЗОКАЙ
2	R	1	0xA000	0x1	ЭКЗОКАЙ
3	W	0	0xA000	0x3	ЭКЗОКАЙ
4	W	1	0xA000	0x4	ЛАДНО

Последовательность транзакций, показанная в предыдущей таблице,

выполняется следующим образом: 1.
Первая транзакция - это ведущая транзакция, выполняющая транзакцию с
исключительным чтением с идентификатором 0 из адреса 0xA000*

Эксклюзивное оборудование для мониторинга доступа сохраняет идентификатор и адрес этой транзакции в своей таблице, и подчиненное устройство отвечает прочитанными данными, 0x1*

Поскольку эксклюзивные права доступа корректно поддерживаются для этого подчиненного устройства, оборудование для мониторинга эксклюзивного доступа отвечает ответом EXOKAY.

Позже ведущее устройство выполняет новую транзакцию с исключительным доступом к чтению с идентификатором 1 с того же адреса, `0xA000`, что и 2. при первой транзакции

аппаратное обеспечение мониторинга исключительного доступа сохраняет подробную информацию об этой новой транзакции в таблице, а ведомое устройство отвечает прочитанными данными, `0x1`.

Опять же, поскольку эксклюзивные права доступа корректно поддерживаются для этого подчиненного устройства, аппаратное обеспечение мониторинга с эксклюзивным доступом выдает ответ EXOKAY.

На данный момент в нашем примере у нас есть две разные текущие эксклюзивные последовательности для одной и той же ячейки памяти.

После того, как ведущее устройство завершило свою работу, оно выполняет эксклюзивную транзакцию записи с идентификатором 0 3. для адреса `0xA000`

Аппаратное обеспечение мониторинга эксклюзивного доступа проверяет детали этой транзакции в своей таблице и,

видя запись с идентификатором 0 и адресом, отвечает ведущему с помощью EXOKAY `0xA000`

ответ. Это означает, что ни один другой ведущий сервер не обновлял эту ячейку памяти, и подчиненное устройство может обновить ее полученным новым значением, которое в нашем примере является `0x3`.

Поскольку содержимое адресной ячейки `0xA000` если был изменен монопольный доступ оборудование мониторинга удаляет из своей таблицы все записи, соответствующие этому адресу местоположения.

4. Наконец, мастер снова выполняет новую транзакцию с правами записи с идентификатором 1 для адреса

`0xA000`.

Оборудование мониторинга монопольного доступа проверяет детали этой транзакции в своей таблице. Не найдя никаких записей с адресом, `0xA000`, он выдает ответ ОК.

ответ "ОК" означает, что в этом месте в памяти была выполнена

предыдущая операция записи, которая обновила данные. В этом случае подчиненное устройство не может

обновить ячейку памяти новым значением, эта ситуация является ошибкой `0x4`

исключительного доступа. В этом случае ведущее устройство должно перезапустить полный эксклюзивный последовательность доступа, начинающуюся с эксклюзивного чтения, а затем снова с эксклюзивной записи.

Этот пример демонстрирует, как эксклюзивные режимы доступа реализуют неблокирующее поведение. Именно это поведение обеспечивает большую пропускную способность системы по сравнению с блокировочными режимами доступа.

7 Порядок передачи данных и порядок оформления транзакций

В этом разделе руководства анализируются некоторые примеры последовательностей транзакций чтения и записи, чтобы помочь вам понять взаимосвязи между различными каналами AXI. В этом разделе также объясняются некоторые правила, регулирующие транзакции, и то, как идентификаторы переводов могут поддерживать транзакции, выполненные не по порядку. Мы также рассмотрим:

- •

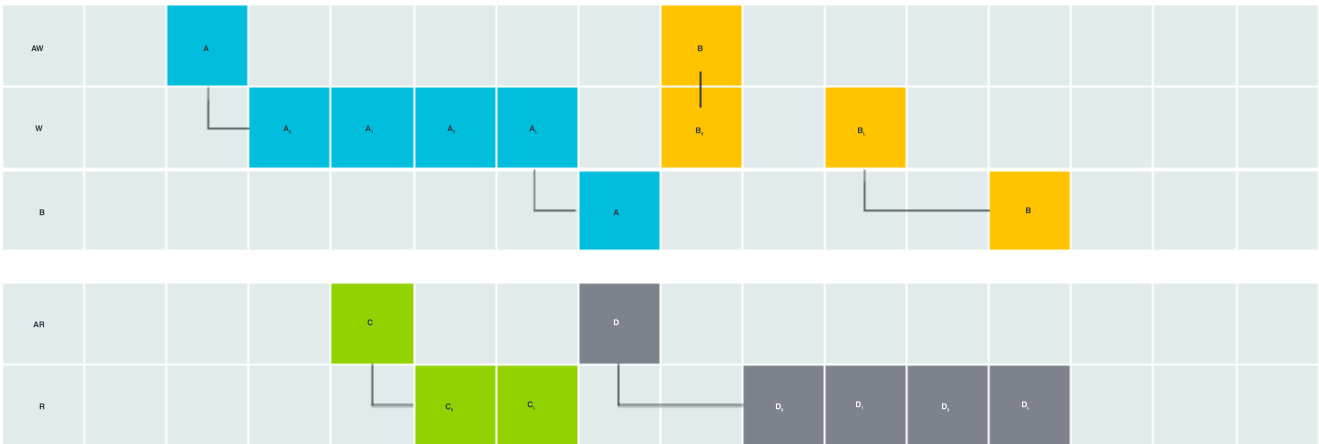
Невыровненные передачи и то, как они помогают оптимизировать использование полосы пропускания

Различия между кодировками основных параметров, относящиеся к интерфейсам AXI. Эти параметры полезны при реализации межсоединения с большим количеством каналов, с несколькими транзакциями

7.1 Примеры простых транзакций

Примеры простых транзакций помогают объяснить взаимосвязи между различными каналами AXI.

На следующей диаграмме показано временное представление нескольких допустимых транзакций по пяти каналам интерфейса AXI3 или AXI4:



- Различные транзакции в этом примере следующие:
1. Транзакция A, которая представляет собой транзакцию записи, содержащую четыре передачи.

Ведущий сначала помещает адрес A в канал AW, затем вскоре помещает последовательность из четырех передач данных по каналу W, заканчивающуюся на AL, где L означает последнюю. Как только все четыре передачи данных завершаются, ведомое устройство отвечает по каналу B. Пока выполнялась транзакция A, мастер также использовал каналы чтения для выполнения транзакции чтения C, которая содержит две передачи.

Поскольку это транзакция чтения, ответ от подчиненного устройства по другому каналу отсутствует, когда транзакция завершается. Вместо этого ответ от ведомого устройства включается в канал R одновременно с данными. Как

3. только транзакция C завершается, ведущее устройство использует канал адреса чтения AR для отправки нового адреса чтения, D, ведомому устройству. В этом случае ответ от ведомого устройства приходит не сразу. На это указывает пустой временной интервал между D и D0. Подобные задержки могут происходить. Подчиненное устройство не обязано отвечать немедленно. Например, подчиненное устройство может быть занято выполнением другой операции или может потребоваться время для извлечения данных. В

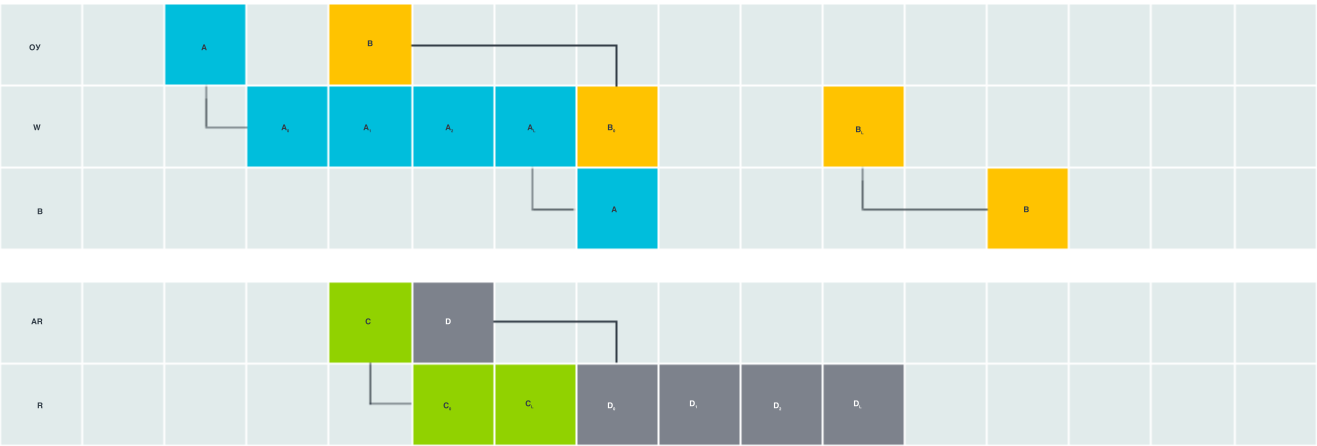
конце концов, ведомое устройство отвечает четырьмя последовательными передачами, от D0 до D3, по каналу R.

Наконец, пока выполняется транзакция чтения D, ведущее устройство использует канал адреса записи, AW, чтобы отправить новый адрес, B, ведомому устройству для операции записи.

Ведущий вводит данные B0 в канал W одновременно с вводом соответствующего адреса B в канал AW. В этом примере существует задержка между передачами данных B0 и B3, и другая задержка перед ответом B. Транзакция завершается только тогда, когда подчиненное устройство отправляет ответ ведущему устройству.

Все эти примеры являются допустимыми транзакциями.

На следующей диаграмме показана та же последовательность транзакций чтения и записи на другой, но все еще действительной временной шкале:



В этом примере ведущий запускает транзакцию B до того, как завершит транзакцию A.

Ведущий использует канал адреса записи, AW, чтобы начать новую транзакцию, передавая новый адрес B ведомому устройству до того, как оно завершит передачу данных для транзакции A по каналу W. Данные для транзакции B передаются подчиненному устройству, когда все данные для транзакции A завершены. Ведущий не ожидает ответа по каналу B для транзакции A, прежде чем он начнет передавать данные для транзакции B. В то же время, ведущий использует канал адреса чтения для последовательной передачи прочитанных адресов C и D для ведомого устройства. Ведомое устройство последовательно отвечает на два запроса чтения.

В этом примере показана другая допустимая комбинация транзакций чтения и записи, происходящая по разным каналам. Это демонстрирует гибкость протокола AXI и возможность оптимизировать производительность межсоединения.

7.2 Идентификаторы передачи

Протокол AXI определяет шину сигналов идентификатора для каждого канала. Маркировка каждой транзакции идентификатором дает возможность завершать транзакции не по порядку. Это означает, что транзакции в более быстрые области памяти могут завершаться без ожидания более ранних транзакций в более медленные области памяти. Использование идентификаторов передачи данных позволяет реализовать высокопроизводительное межсоединение, максимизируя пропускную способность данных и эффективность системы. Эта функция также может повысить производительность системы, поскольку она уменьшает влияние задержки транзакций.

Шины сигналов ID следующие:

- AWID
 -
 - WID
- СТАВКА
ЗАСУШЛИВЫЙ
- •
- ИЗБАВИТЬСЯ

Протокол AXI поддерживает неупорядоченные транзакции, позволяя каждому интерфейсу действовать как нескольким упорядоченным интерфейсам. Согласно спецификациям протокола AXI, все транзакции с заданным идентификатором должны быть упорядочены. Однако нет ограничений на упорядочивание транзакций с разными идентификаторами. При работе с идентификаторами переводов следуйте этим правилам:

- Все переводы должны иметь идентификатор.
- Все переводы в транзакции должны иметь один и тот же идентификатор.
- Ведущие устройства могут поддерживать несколько идентификаторов для нескольких потоков.
- Подчиненным устройствам обычно требуется настраиваемая ширина идентификатора.

Вы также должны помнить эти два важных параметра AXI для сигналов идентификатора:

- Ширина идентификатора записи, представляющая собой количество битов, используемых для шин AWID, WID и BID
- Ширина идентификатора чтения, представляющая собой количество битов, используемых для шин ARID и RID

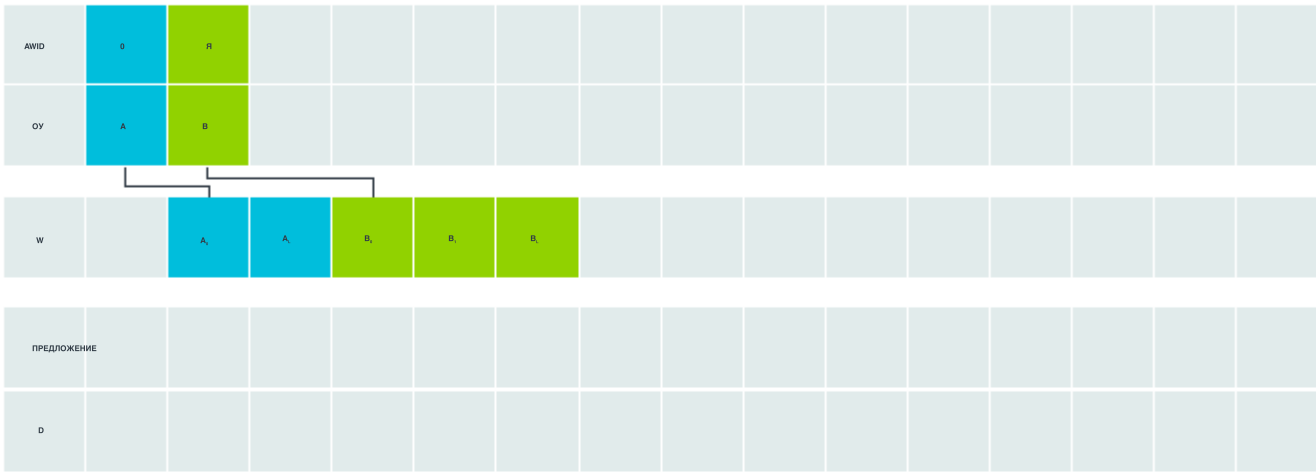
7.3 Правила упорядочения транзакций записи

Существуют три правила упорядочения AXI для транзакций записи.


Правила следующие:

- Запись данных по каналу W должна выполняться в том же порядке, что и передача адресов по каналу AW .

Следующая диаграмма иллюстрирует это правило.:



В этом примере основные проблемы адресуются A, затем B, поэтому данные должны начинаться с A0 перед B0.



Note

Чередование данных записи с разными идентификаторами в канале W было разрешено в AXI3, но устарело в AXI4 и более поздних версиях.

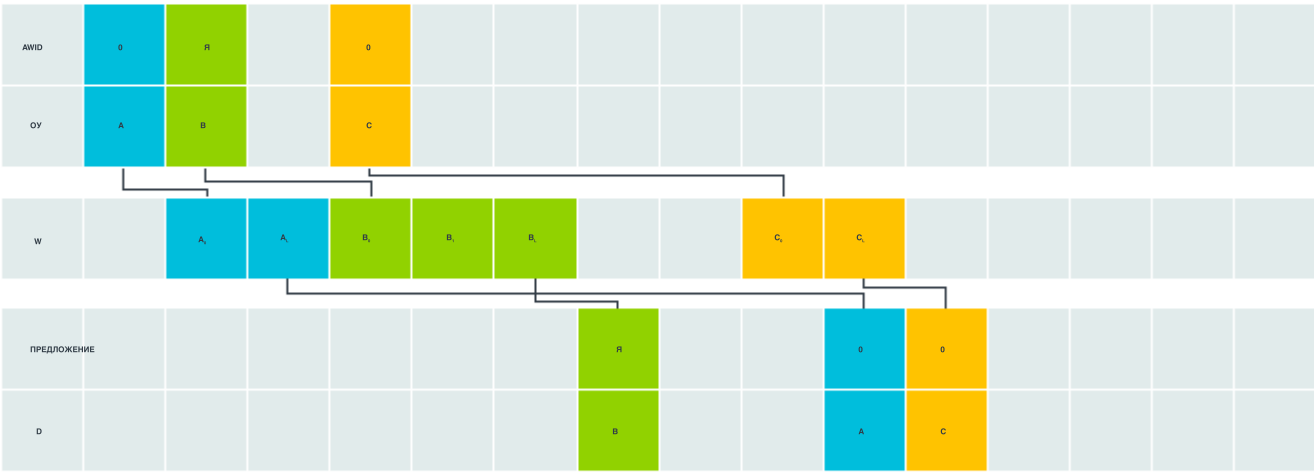
- Транзакции с разными идентификаторами могут завершаться в любом порядке. Следующая диаграмма иллюстрирует это правило.:



В этом примере транзакция В завершается раньше транзакции А, даже если транзакция А началась первой. У основного сервера может быть несколько

- незавершенных транзакций с одним и тем же идентификатором, но они должны выполняться по порядку и завершаться по порядку.

Следующая диаграмма иллюстрирует это правило:



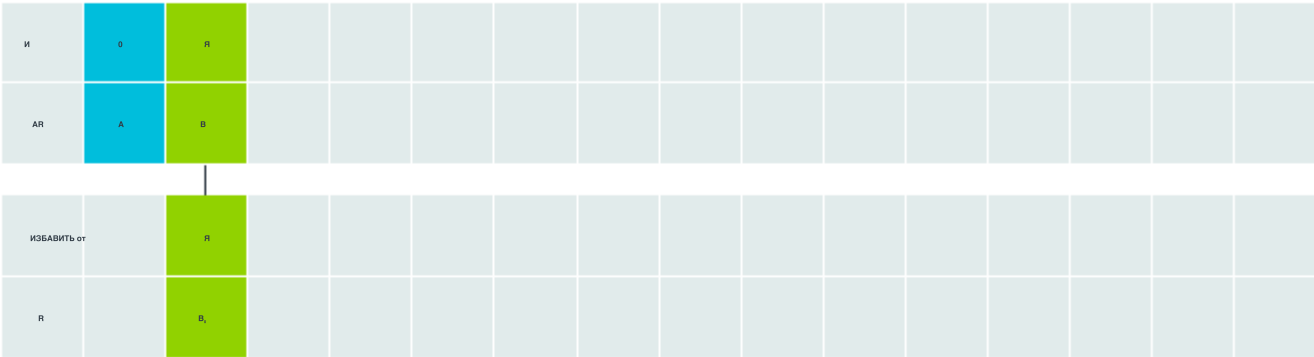
В этом примере транзакция В имеет идентификатор, отличный от других транзакций, поэтому она может завершиться в любой момент. Однако транзакции А и С имеют одинаковый идентификатор, поэтому они должны завершаться в том же порядке, в котором они были выпущены: сначала А, затем С.

7.4 Ознакомьтесь с правилами упорядочения транзакций

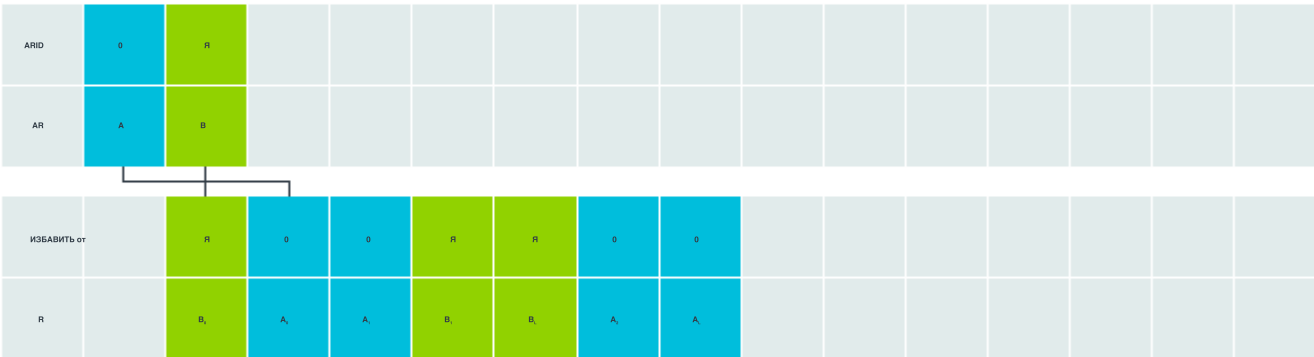
Существуют три правила оформления транзакций для чтения.

Правила следующие:

- Данные для чтения с разными идентификаторами в канале R не имеют ограничений по порядку. Это означает, что подчиненное устройство может отправлять его в любом порядке.
На следующей диаграмме показан пример, в котором транзакция В обслуживается раньше, чем А, даже несмотря на то, что адрес для транзакции А получен первым:



- Считанные данные для различных идентификаторов в канале R могут чередоваться, при этом значение RID определяет, к какой транзакции относятся данные. На следующей диаграмме показан пример, в котором R-данные для транзакций А и В чередуются.:



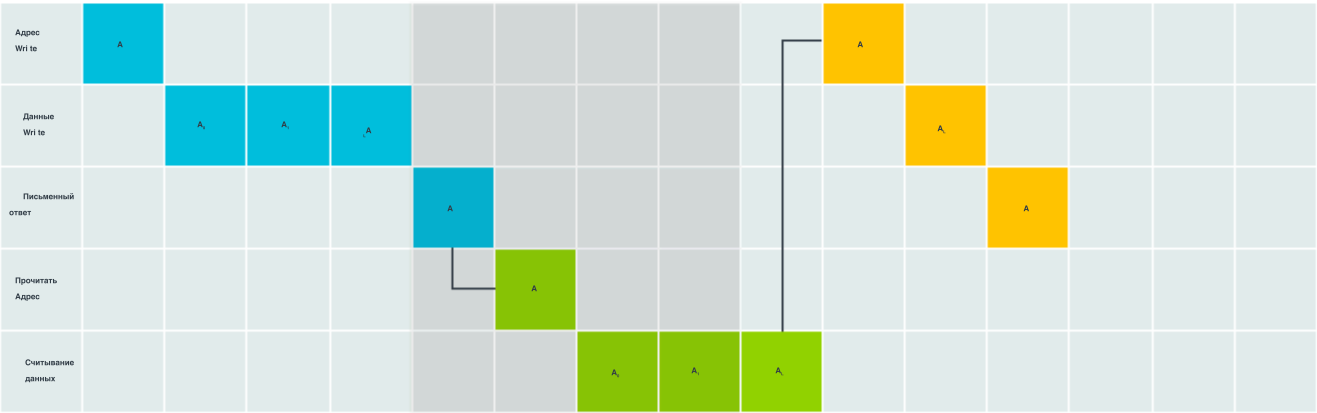
- Для транзакций с одинаковым идентификатором данные для чтения по каналу R должны быть возвращены в том порядке, в котором они были запрошены.
На следующей диаграмме показан пример, когда транзакции А и С имеют одинаковое значение RID, равное 0:



Поскольку транзакция А была запрошена до транзакции С, подчиненное устройство должно вернуть все четыре значения R data для А перед значениями данных для С.

7.5 Порядок каналов чтения и записи

Каналы чтения и записи не имеют правил упорядочивания по отношению друг к другу. Это означает, что они могут завершаться в любом порядке. Итак, если ведущему устройству требуется упорядочение для определенной последовательности операций чтения и записи, ведущее устройство должно гарантировать, что порядок транзакций соблюдается, явно ожидая завершения транзакций перед выдачей новых. На следующей диаграмме показан пример, в котором ведущему устройству требуется определенный порядок для последовательности транзакций записи- чтения-записи с адреса:




- Последовательность операций следующая:
1. Мастер запускает первую транзакцию записи.
 2. Ведущее устройство гарантирует, что ведомое устройство завершило транзакцию записи, ожидая сигнала на канале ответа на запись.
 3. Мастер запускает транзакцию чтения.
 4. Ведущий ожидает окончательного ответа по каналу считанных данных.

5. Мастер запускает вторую транзакцию.

7.6 Невыровненный начальный адрес перевода

Протокол AXI поддерживает транзакции с невыровненным начальным адресом, который влияет только на первую передачу в транзакции. После первой передачи в транзакции все остальные передачи выравниваются.



Note

Протокол AXI также поддерживает невыровненную передачу с использованием стробоскопических сигналов. Смотрите раздел [Стробоскопическая запись данных](#) для получения дополнительной информации.

При несогласованной передаче значения AхADDR не обязательно выравнивать по ширине транзакции. Например, не выровнена

32-разрядный пакет данных, который начинается с байтового адреса 0х1002к естественной границе 32-разрядного адреса, поскольку в точности не делится на 0х1002 0х20.

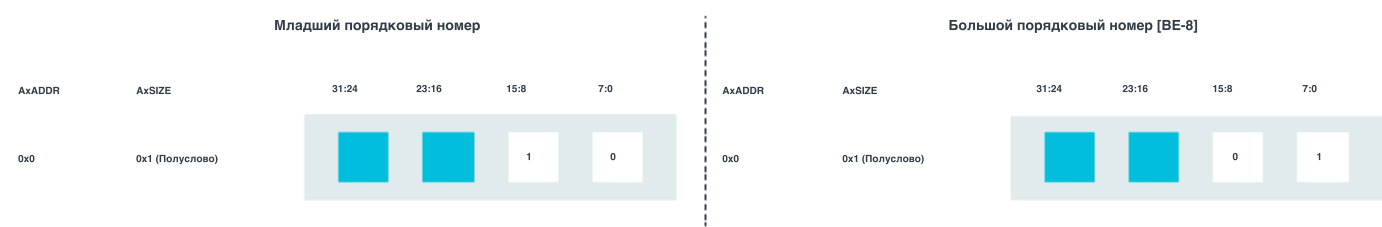
В следующем примере показана 5-битная 32-разрядная передача, начинающаяся с невыровненного адреса 0х01:



В следующем примере показана пятибитная транзакция размером 16 бит, начинающаяся с адреса 0x03:

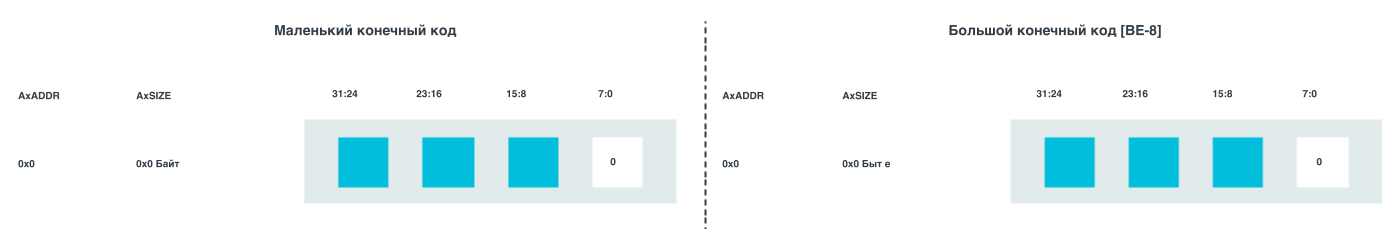


В следующем примере показаны представления одного и того же двухбайтового слова как в младшем, так и в старшем разрядах.:



Для полуслова из двух байтов в режиме строчной последовательности самый значащий байт использует байтовую полосу 1, а наименее значащий байт использует байтовую полосу 0. Опять же, в режиме big-endian BE-8 переключаются полосы, которые используются двумя байтами. Самый значимый байт использует байтовую полосу 0, а наименее значимый байт использует байтовую полосу 1.

Наконец, для одного байта нет разницы между режимом с младшим и старшим порядком, как показано в следующем примере:



В обоих случаях байт использует байтовую полосу 0.

В настраиваемом компоненте endianness, таком как ядро Arm, которое поддерживает BE-8, переупорядочение байтов должно выполняться внутренне, так что ничего не нужно делать на уровне межсоединения. С другой стороны, пользовательское устройство, подключенное к межсоединению AXI, которое по своей природе является BE-8, уже имело бы правильный порядок байтов. Наличие BE-8 в протоколе AXI упрощает поддержку динамического переключения по порядковому номеру.

7.8 Атрибуты интерфейса чтения и записи.

В этом разделе руководства освещаются некоторые из наиболее важных атрибутов для настройки каналов записи AXI и чтения.

К атрибутам интерфейса записи относятся следующие:

- Возможность выдачи разрешения на запись

Представляет максимальное количество активных транзакций записи, которые может сгенерировать главный интерфейс.

Возможность чередования записи (только AXI3)

Количество активных транзакций записи, для которых главный интерфейс способен передавать данные.

- Возможность принятия записи (только AXI3)

Представляет максимальное количество активных транзакций записи, которые может
принять подчиненный интерфейс *

Атрибут глубины передования записи

Представляет количество активных транзакций записи, от которых подчиненный интерфейс может получать данные

Атрибуты интерфейса чтения включают следующее:

- Атрибут возможности выдачи чтения

Представляет максимальное количество активных транзакций чтения, которые
может сгенерировать
главный интерфейс с возможностью принятия чтения

Максимальное количество активных транзакций чтения, которые может
принять подчиненный интерфейс *

Переупорядочивание данных, прочитайте глубина

Число активных читать транзакции, по которым подчиненный интерфейс
может передавать данные, отсчитывается с первой сделки

8 Проверьте свои знания

Вопрос: Какой тип пакета должен выдать master, если он хочет выполнить запись в

FIFO: исправленный, переносащий или увеличивающий?

А: Исправлено. FIFO работает путем записи на фиксированный адрес и чтения с него.

Вопрос: Все каналы AXI4 используют один и тот же механизм подтверждения связи.

ДОСТОВЕРНЫЙ сигнал поступает от источника к получателю, указывая, когда доступна достоверная информация.

Какой сигнал поступает от получателя к источнику, указывая, когда он может принять

информацию? А: Сигнал ГОТОВНОСТИ.

Вопрос: Какие сигналы могут использоваться для защиты от незаконных транзакций ниже по потоку в системе?

А: Сигналы AWPROT и ARPROT

Вопрос: Какова цель идентификаторов передачи?

О: Маркировка транзакций разными идентификаторами позволяет транзакциям с разными идентификаторами завершаться не в порядке. Это означает, что транзакции в более быстрые области памяти могут завершаться без ожидания более ранних транзакций в более медленные области памяти.

9 Сопутствующая информация

Вот некоторые ресурсы, связанные с материалами

этого руководства

AMBA для разработчика Arm

- Видеоуроки по Arm:
 - Каналы AXI
 - Основные функции AXI Протокол AXI
 - Протокол AXI в проектировании системы с несколькими мастерами Введение в протокол AMBA AXI
 - Что такое AMBA и зачем его использовать?

10 следующих шагов

В этом руководстве представлен обзор основных тем, относящихся к AMBA AXI, включая использование и работу различных каналов и сигналов. Эти знания будут полезны, когда вы узнаете больше об AMBA AXI, прочитав [Спецификации протоколов AMBA AXI и ACE](#). Вы можете применить свои знания на практике для разработки интерфейсов, реализующих протокол AMBA AXI.