

## **Арбитраж систем мультимастер**

Система структуры внутренних соединений поддерживает системы с несколькими мастер компонентами. В системе с несколькими мастер компонентами, например система на рис. 2-1 на странице 2-2, система структуры внутренних соединений предоставляет расширенный доступ к слейвам, используя технологию, называемую арбитраж на стороне слейва. Арбитраж на стороне слейва перемещает арбитражную логику ближе к слейву, например, это алгоритм, определяющий то, как мастер получает доступ к определённому слейву в момент, когда несколько мастеров запрашивают доступ к этому же слейву в это время.

Архитектура мультимастера используется системой структуры внутренних соединений для получения следующих преимуществ:

- Исключает необходимость создавать аппаратную часть арбитража вручную.
- Позволяет нескольким мастерам последовательно передавать данные. В отличие от традиционной архитектуры арбитража на стороне хоста, когда каждый мастер должен ждать, пока ему выделят доступ к общей шине, несколько мастеров Avalon-MM могут последовательно выполнять обмен с независимыми слейвами. Арбитражная логика приостанавливает мастер только тогда, когда несколько мастеров одновременно запрашивают доступ к одному слейву.
- Исключает ненужные соединения мастер-слейв. Соединения между мастером и слейвом существуют только, если они заданы в SOPC Builder. Если мастер никогда не инициализирует трансферт к определённому слейву, не требуется их соединение, и поэтому SOPC Builder не расходует ресурсы логики для подключения двух портов.

- Предоставляет конфигурируемые настройки арбитража, а арбитраж для каждого слейва может быть задан независимо. Например, вы можете выделить одному мастеру больше арбитража, чем другим, позволяя ему больше циклов доступа к слейву. Общие настройки арбитража определяются для каждого слейва независимо.
- Упрощает разработку мастер компонента. Детальность арбитража скрыты внутри системы структуры внутренних соединений. Каждый Avalon-MM мастер подключен к системе структуры внутренних соединений так, как будто он единственный мастер в системе. В результате, вы можете использовать компонент в системе с одним мастером и в системе мультимастер, без изменений в проекте компонента.

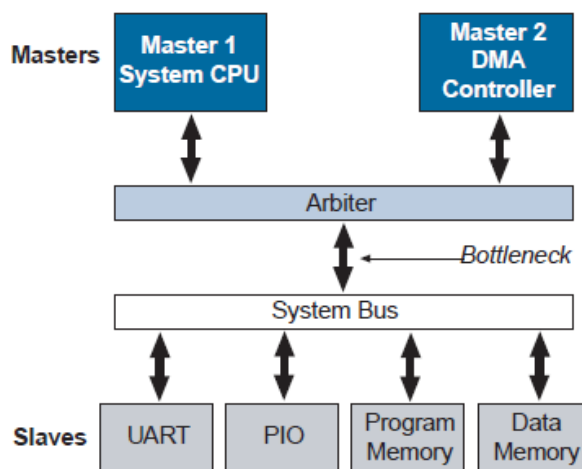
### Традиционная архитектура общей шины

В этой секции описывается архитектура системы структуры внутренних соединений, генерируемая SOPC Builder для систем мультимастер. В связи с обсуждением мультимастера и арбитража, в этой секции описывается традиционная архитектура шины.

В традиционной архитектуре шины, один или более мастеров шины и слейвов шины подключены к общей шине, состоящей из проводников на печатной плате или внутри чипа. Один арбитр контролирует шину (т.е., путь между мастерами и слейвами на шине), так что несколько мастеров шины не могут последовательно управлять шиной. Каждый мастер шины запрашивает контроль над шиной от арбитра, и арбитр предоставляет временный доступ одному мастеру. После того, как мастер начинает контролировать шину, он инициализирует трансферт с одним из слейвов шины. Когда несколько мастеров одновременно запрашивают доступ к шине, арбитр выделяет ресурсы шины одному мастеру, принуждая остальных мастеров ждать.

На рис. 2-6 показана архитектура шины для традиционной процессорной системы. Доступ к общей шине системы – узкое место для параметра пропускной способности: только один мастер имеет доступ к шине, заставляя остальных мастеров ждать, и только один слейв может передавать в это время данные.

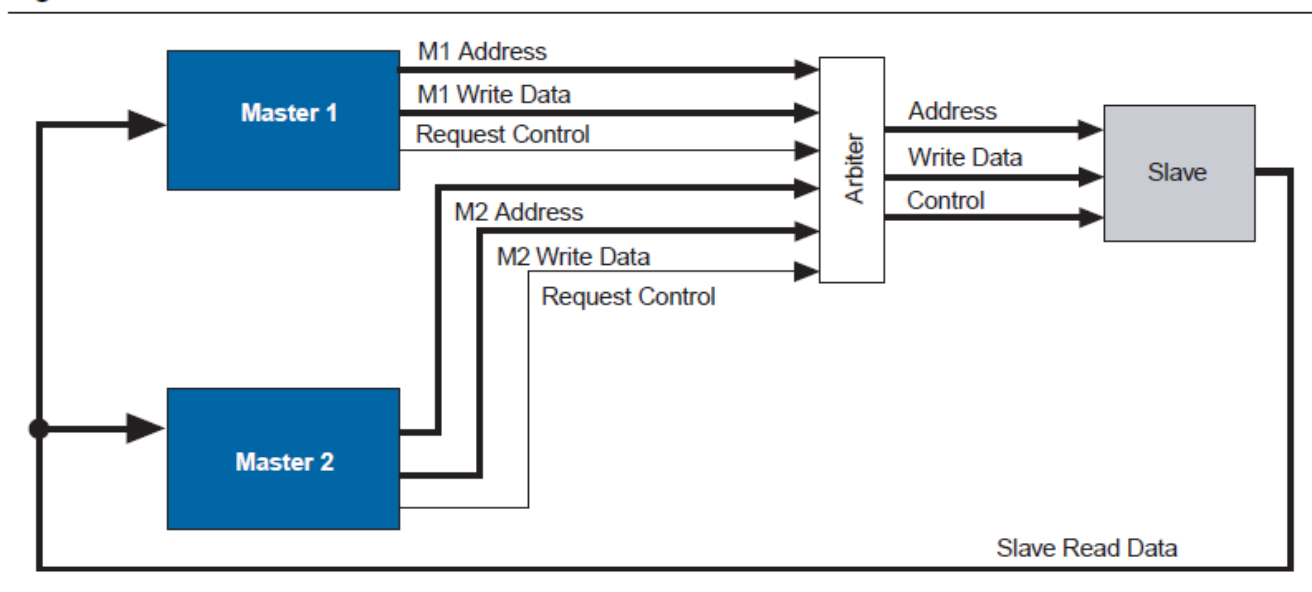
**Figure 2–6.** Bus Architecture in a Traditional Microprocessor System



**Арбитраж на стороне слейва**

Система структуры внутренних соединений использует архитектуру мультимастера для исключения узкого места при доступе по общей шине. Несколько мастеров могут быть активными одновременно, последовательно обмениваясь данными с независимыми слейвами. Например, на рис. 2-1 на стр. 2-2 показана система с двумя мастерами (CPU и DMA контроллер) и общим слейвом (SDRAM контроллер). Арбитраж выполняется для SDRAM слейва; арбитр выбирает одному мастеру право обращаться к слейву, если они оба одновременно инициировали обмен.

На рис. 2-7 изображены два мастера и общий слейв, а также показаны дополнительно пути данных, адреса и контроля. Арбитражная логика мультиплексирует все сигналы данных, адреса и контроля от мастера к общему слейву.

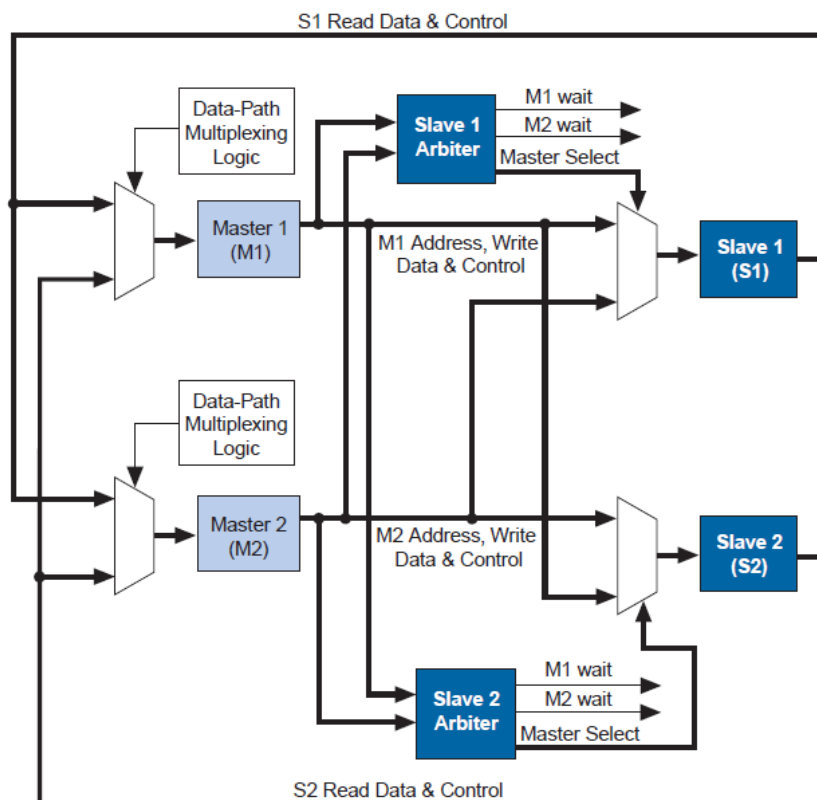
**Figure 2-7. Detailed View of Multimaster Connections****Подробнее об арбитраже**

SOPC Builder генерирует арбитр для каждого слейва, основываясь на параметрах арбитража, заданных в SOPC Builder. Арбитражная логика выполняет следующие функции для своего слейва:

- Вычисляет сигналы адреса и контроля от каждого мастера и определяет, какой мастер получит следующим доступ к слейву.
- Предоставляет доступ выбранному мастеру и приостанавливает других запрашивающих мастеров.
- Использует мультиплексор для подключения путей адреса, контроля и данных между несколькими мастерами и слейвом.

На рис. 2-8 показана арбитражная логика на примере системы мультимастер с двумя мастерами, каждый из которых подключен к двум слейвам.

**Figure 2–8.** Block Diagram of Arbiter Logic



### Правила арбитража

В этой секции описываются правила, по которым арбитр предоставляет доступ мастерам, когда они конфликтуют.

### Установка параметров арбитража в SOPC Builder

Вы можете задать общий арбитраж для каждого мастера, используя панель соединений на вкладке **System Contents** SOPC Builder, как показано на рис. 2-9.

**Figure 2–9.** Arbitration Settings on the System Contents Tab

Module Name	Description	Clock
<input type="checkbox"/> <b>cpu</b>	Nios II Processor - Alte...	clk
└─ <input type="checkbox"/> instruction_master	Master port	
└─ <input type="checkbox"/> data_master	Master port	
1 └─ <input type="checkbox"/> jtag_debug_module	Slave port	
<input checked="" type="checkbox"/> <b>sys_clk_timer</b>	Interval timer	clk
1 <input checked="" type="checkbox"/> <b>ext_ram_bus</b>	Avalon Tri-State Bridge	clk
└─ <input type="checkbox"/> ext_flash	Flash Memory (Commo...	
└─ <input type="checkbox"/> ext_ram	IDT71V416 SRAM	
1 <input checked="" type="checkbox"/> <b>epcs_controller</b>	EPCS Serial Flash Cont...	clk
└─ <input type="checkbox"/> lan91c111	LAN91c111 Interface (...)	
1 <input checked="" type="checkbox"/> <b>jtag_uart</b>	JTAG UART	clk

Настройки арбитража по умолчанию скрыты. Чтобы увидеть их, в меню View нажмите **Show Arbitration** (Показать арбитраж).

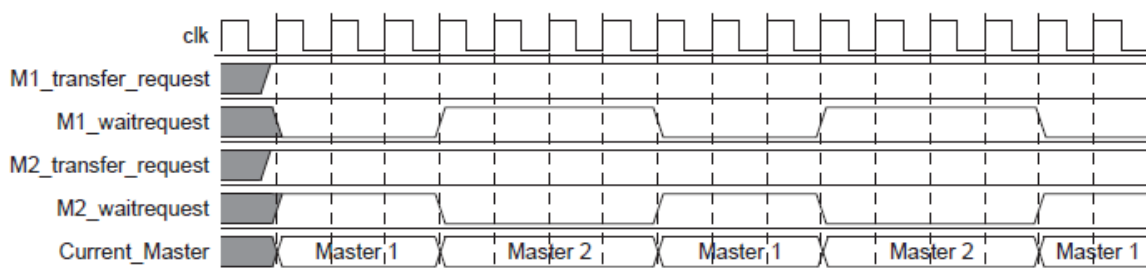
### Равноправные части

Арбитражная логика использует равноправную (fairness-based) арбитражную схему. В этой схеме, каждая пара мастера имеет целое значение трансфертов *частей*, по отношению к слейву.

Одна часть является разрешением для выполнения одного трансферта.

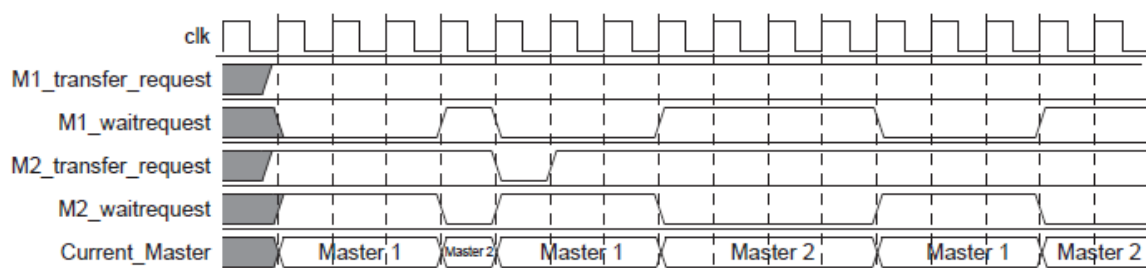
Например, представим, что два мастера непрерывно обращаются к слейву для выполнения непрерывных трансфертов. Мастеру 1 назначено три части, а мастеру 2 – четыре части. В этом случае, арбитр предоставляет мастеру 1 доступ к трём трансфертам, а мастеру 2 – четыре трансферта. Этот цикл повторяется бесконечно. На рис. 2-10 показан этот случай, показаны выход запроса каждого трансферта мастера, ожидание входа запроса (который управляется арбитражной логикой), и текущий мастер, контролирующий слейв.

**Figure 2-10.** Arbitration of Continuous Transfer Requests from Two Masters



Если мастер останавливает трансферты запроса, он пропускает свои части, он теряет все свои оставшиеся части, а арбитр предоставляет доступ другому запрашивающему мастеру. Посмотрите на рис. 2-11. После выполнения одного трансферта, мастер 2 останавливает запрос на один тактовый цикл. В результате, арбитр предоставляет доступ снова мастеру 1, который полностью получает свои части.

**Figure 2-11.** Arbitration of Two Masters with a Gap in Transfer Requests



### Циклическое планирование

Когда несколько мастеров имеют доступ к слейву, арбитр выделяет части в циклическом порядке. Циклическое планирование управляет интерфейсом запросов в зависимости с доступным пространством и доступными данными, разрешёнными для передачи по интерфейсу. Для каждого слейв трансферта, только запрашивающие мастера включены в арбитраж.

***Трансферты пакетов***

Трансферты пакетов Avalon-MM предоставляют непрерывный мастер доступ к слейву для заданного количества трансфертов. Мастер определяет количество трансфертов, когда инициализирует пакет. Когда пакет передаётся по каналу мастер-слейв, арбитражная логика не позволяет другим мастерам доступ к слейву, пока пакет не закончится. Для мастер пакетов, размер пакета определяется количеством циклов, когда мастер имеет доступ к слейву, а выбор частей арбитража (см. "Равноправные части") не учитывается.