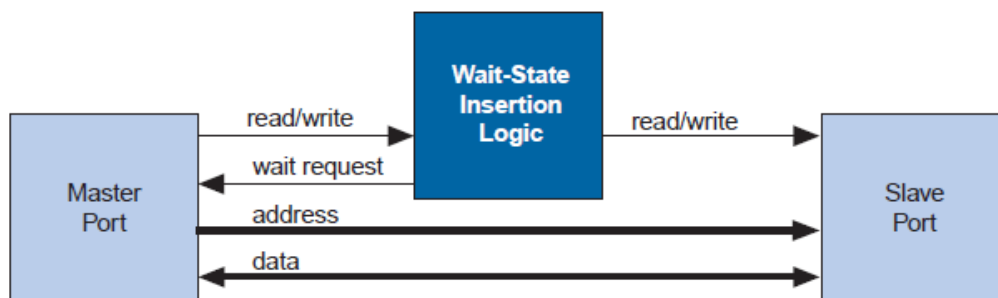


Вставка состояния ожидания

Состояния ожидания расширяют время перехода на один или более циклов. Вставка логики состояния ожидания обеспечивает временные потребности для каждого слеива, и вводит мастера в ожидание, пока работает слеив. Система структуры внутренних соединений вставляет состояния ожидания в трансферт, когда выбранный слеив не может ответить за один тактовый цикл. Система структуры внутренних соединений также вставляет состояния ожидания в тех случаях, когда сигналы `read_enable` и `write_enable` (разрешение чтения/записи) имеют требования ко времени установки и удержания.

Логика вставки состояния ожидания – это небольшой конечный автомат, который транслирует последовательность контрольных сигналов между стороной слеив и стороной мастера. На рис. 2-5 показана блок-схема логики вставки состояния ожидания между одним мастером и одним слеивом.

Figure 2–5. Block Diagram of Wait State Insertion Logic

Система структуры внутренних соединений может форсировать перевод мастера в состояние ожидания, в дополнение к потребностям в состоянии ожидания от слейва. Например, арбитражная логика в системе мультимастер может форсировать перевод мастера в состояние ожидания, пока система предоставляет доступ для слейва.

SOPC Builder генерирует логику вставки состояния ожидания, основываясь на свойствах всех слейвов системы.

Конвейерное чтение трансфертов

Интерфейс Avalon-MM поддерживает конвейерное чтение трансфертов, позволяя конвейеру мастера начинать множество чтений трансфертов в последовательности, без ожидания завершения приоритетных трансфертов. Конвейерные трансферты позволяют паре мастер-слейв достигать высокой пропускной способности, даже если слейву требуется один или несколько циклов задержки перед чтением данных для каждого трансферта.

SOPC Builder генерирует систему структуры внутренних соединений с логикой управления конвейером, чтобы извлечь возможную выгоду из конвейерных компонентов, основываясь на свойствах каждой пары мастер-слейв в системе. Независимо от задержек конвейера в определенном слейве, SOPC Builder гарантирует, что для каждого мастера данные чтения будут поступать в порядке запроса. Поскольку мастер и слейв зачастую имеют неправильный подбор конвейерных задержек, система структуры внутренних соединений часто имеет логику для согласования различий. В большинстве случаев, конвейерная задержка возможна, см. табл. 2-1.

Таблица 2-1. Различные случаи возможной конвейерной задержки в паре мастер-слейв

Мастер	Слейв	Структура логики управления конвейером
Нет конвейера	Нет конвейера	Система структуры внутренних соединений не устанавливает логику обработки конвейерной задержки.
Нет конвейера	Конвейер с фиксированной или переменной задержкой	Система структуры внутренних соединений форсирует перевод мастера в режим ожидания на время циклов задержки со стороны слейва. Такая пара мастер-слейв не использует преимущества конвейерной обработки, поскольку мастер ожидает окончания трансферта, прежде чем начать новый трансферт. Однако пока мастер ожидает, слейв может получить трансферт от другого мастера.
Конвейер	Нет конвейера	Система структуры внутренних соединений выполняет трансферт как будто ни мастер, ни слейв не имеют

Volume 4: SOPC Builder

2. Система структуры внутренних соединений для интерфейсов с распределением в памяти

Перевод: Егоров А.В., 2010 г.

		конвейера, мастер ожидает, пока слейв возвратит данные.
Конвейер	Конвейер с фиксированной задержкой	Система структуры внутренних соединений позволяет мастеру получать данные по точному тактовому циклу, когда поступают правильные данные от слейва. Этот процесс позволяет получить максимальную пропускную способность пары мастер-слейв.
Конвейер	Конвейер с переменной задержкой	Это простейший случай конвейера, при котором слейв назначает сигнал, при правильном состоянии сигнала readdata, и мастер получает данные. Этот случай позволяет достигнуть максимальной пропускной способности пары мастер-слейв.

SOPC Builder генерирует логику обработки конвейерной задержки, основываясь на свойствах всех мастеров и слейвов системы. При конфигурировании системы SOPC Builder, нет настроек, которые бы напрямую контролировали логику управления конвейером в системе структуры внутренних соединений.

Динамический размер шины и собственное выравнивание адресов

SOPC Builder генерирует систему структуры внутренних соединений для согласования мастера и слейвов с несогласованной шириной данных. Выравнивание адресов влияет на то, какие данные слейва будут выровнены в адресном пространстве мастера, в том случае, когда ширина данных мастера и слейва отличаются. Выравнивание адреса – это свойство каждого слейва, оно может быть различаться между слейвами в системе. Слейв может декларировать себе использовать одно из следующих средств:

- Динамический размер шины
- Собственное выравнивание адресов

В следующих секциях описана реализация средства выравнивания адресов слейв устройств.

Динамический размер шины

Динамический размер шины скрывает подробности интерфейса узких компонентов устройства с широким мастером и наоборот. Когда N-битный мастер обращается к слейву с динамическим размером шины, мастер работает независимо с полными N-битными словами данных, не задумываясь о ширине данных слейва.

Когда используется динамический размер шины, ширина данных слейва в байтах должна быть в степени два.

Динамический размер шины даёт следующие преимущества:

- Исключается потребность создавать вручную аппаратную подстройку адреса.
- Уменьшается сложность разработки мастер компонента.
- Разрешается любому мастер компоненту иметь доступ к любому устройству памяти, в независимости от ширины данных.

В случае с динамическим размером шины, система структуры внутренних соединений включает небольшой конечный автомат, который согласовывает разницу между шириной данных мастера и слейва. Его поведение зависит от того, будет ли ширина данных мастера шире или уже, чем у слейва.

Широкий мастер

В случае если мастер шире, логика динамического размера шины берёт один, широкий трансферт на стороне мастера, и затем выполняет несколько узких трансфертов на стороне слейва. Для ширины данных $<N>:1$, динамический размер шины генерирует до $<N>$ слейв трансфертов на каждый мастер трансферт. Мастер ждёт, пока не закончатся трансферты на стороне слейва; трансферт мастера заканчивается, когда заканчиваются все трансферты на стороне слейва.

Логика динамического размера шины использует сигналы `byte-enable` (разрешение байта) на стороне мастера для генерирования соответствующих слейв трансфертов. Логика динамического размера шины выполняет столько трансфертов на стороне слейва, сколько требуется для записи или чтения определённого байтового тракта.

Узкий мастер

В случае, когда мастер уже, один трансферт на стороне мастера генерирует один трансферт на стороне слейва. В этом случае, несколько слов мастера адресуются к одному начальному номеру адресного пространства слейва. Логика динамического размера шины размещает каждый мастер адрес в поднабор байтового тракта в соответствующем начальном номере адреса слейва. Все байты слейв памяти доступны в адресном пространстве мастера.

В табл. 2-2 показан доступ 32-битного мастера к 64-битному слейву с динамическим размером шины. В таблице начальный номер (OFFSET) ссылается на начальный номер в адресном пространстве слейва.

Table 2-2. 32-Bit Master View of 64-Bit Slave with Dynamic Bus Sizing

32-bit Address	Data
0x00000000 (word 0)	OFFSET[0] _{31:0}
0x00000004 (word 1)	OFFSET[0] _{63:32}
0x00000008 (word 2)	OFFSET[1] _{31:0}
0x0000000C (word 3)	OFFSET[1] _{63:32}

В случае трансферта чтения, логика динамического размера шины мультиплексирует соответствующие байтовые тракты слейв данных для узкого мастера. В случае трансферта записи, логика динамического размера шины использует сигналы `byte-enable` на стороне слейва для записи только в соответствующий байтовый тракт.

Altera рекомендует вам выбирать динамический размер шины в любом удобном случае. Динамический размер шины предоставляет большую гибкость, когда компоненты мастер и слейв в вашей системе имеют различную ширину.

Собственное выравнивание адресов

В табл. 2-3 показано собственное выравнивание адресов и динамический размер шины для 32-битного мастера, подключенного к 16-битному слейву (соотношение 2:1). В этом примере, слейву предназначен базовый адрес $<BASE>$ в адресном пространстве мастера. В табл. 2-3 OFFSET ссылается на начальный адрес в 16-битном адресном пространстве слейва.

Когда подключается широкий мастер к узкому порту слейва, который использует собственное выравнивание адресов, следующая формула адреса используется для

Volume 4: SOPC Builder

2. Система структуры внутренних соединений для интерфейсов с распределением в памяти

Перевод: Егоров А.В., 2010 г.

вычисления, какой адрес будет представлен системе структуры внутренних соединений:

$$\langle \text{master address} \rangle = \langle \text{slave base address} \rangle + (\langle \text{slave word offset} \rangle * \langle \text{master data width in bytes} \rangle)$$

Например, 64-битному мастеру требуется записать второе слово 32-битного слейва, использующего собственное выравнивание адресов, формула преобразуется:

$$\langle \text{master address} \rangle = \langle \text{slave base address} \rangle + (1 * 8)$$

Table 2-3. 32-Bit Master View of 16-Bit Slave Data

32-bit Master Address	Data with Native Alignment	Data with Dynamic Bus Sizing
<i>BASE</i> + 0x0 (word 0)	0x0000:OFFSET[0]	OFFSET[1]:OFFSET[0]
<i>BASE</i> + 0x4 (word 1)	0x0000:OFFSET[1]	OFFSET[3]:OFFSET[2]
<i>BASE</i> + 0x8 (word 2)	0x0000:OFFSET[2]	OFFSET[5]:OFFSET[4]
<i>BASE</i> + 0xC (word 3)	0x0000:OFFSET[3]	OFFSET[7]:OFFSET[6]
...
<i>BASE</i> + 4 <i>N</i> (word <i>N</i>)	0x0000:OFFSET[<i>N</i>]	OFFSET[2 <i>N</i> +1]:OFFSET[2 <i>N</i>]

SOPC Builder генерирует соответствующую логику выравнивания адресов, основываясь на свойствах мастера и слейвов в системе. Когда конфигурируется система SOPC Builder, нет настроек, которые бы напрямую контролировали выравнивание адреса в системе структуры внутренних соединений.