

Avalon® Streaming (Avalon-ST) interconnect components facilitate the design of high-speed, low-latency datapaths for the system-on-a-programmable-chip (SOPC) environment. Interconnect components in SOPC Builder act as a part of the system interconnect fabric. They are not end points, but adapters that allow you to connect different, but compatible, streaming interfaces. You use Avalon-ST interconnect components to connect cores that send and receive high-bandwidth data, including multiplexed streams, packets, cells, time-division multiplexed (TDM) frames, and digital signal processor (DSP) data.

The interconnect components that you add to an SOPC Builder system insert logic between a source and sink interface, enabling that interface to operate correctly. This chapter describes four Avalon-ST interconnect components, also called adapters:

- **“Timing Adapter” on page 12–3**—adapts between sinks and sources that have different characteristics, such as ready latencies.
- **“Data Format Adapter” on page 12–6**—adapts source and sink interfaces that have different data widths.
- **“Channel Adapter” on page 12–8**—adapts source and sink interfaces that have different settings for the channel signal.
- **“Error Adapter” on page 12–9**—ensures that per-bit error information recorded at the source is correctly transferred to the sink

All of these interconnect components adapt initially incompatible Avalon-ST source and sink interfaces so that they function correctly, facilitating the development of high-speed, low-latency datapaths.

Interconnect Component Usage

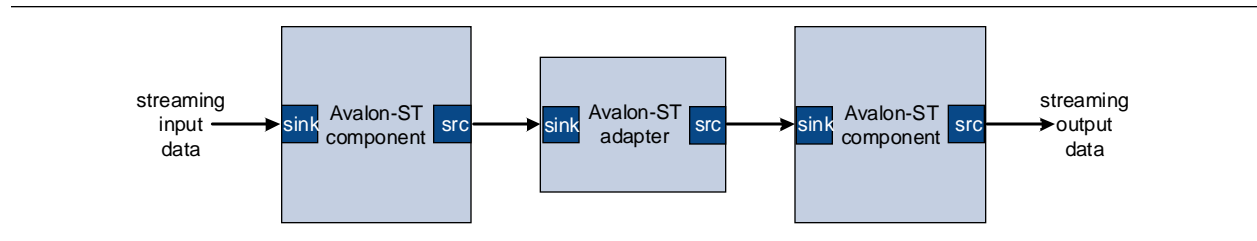
Interconnect components can adapt the data or control signals of the Avalon-ST interface. Typical adaptations to control signals include:

- Adding pipeline stages to adjust the timing of the ready signal
- Tying signals that are not used by either the source or sink to 0 or 1

Typical adaptations to data signals include:

- Changing the number of symbols (words) that are driven per cycle
- Changing the number of channels driven

When the interconnect component adapts the data interface, it has one Avalon-ST sink interface and one Avalon-ST source interface, as shown in [Figure 12–1](#). You configure the adapter components manually, using SOPC Builder. In contrast to the Avalon-MM interface, which allows you to create various topologies with a number of different master and slave components, you always use the Avalon-ST interconnect components to adapt point-to-point connections between streaming cores.

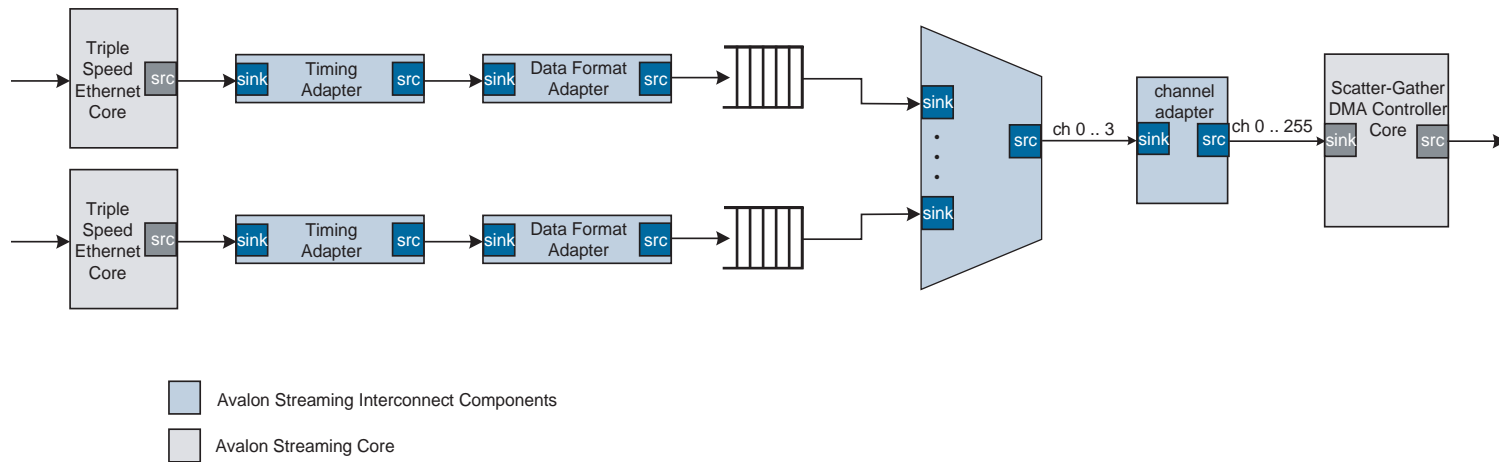
Figure 12-1. Example of an Avalon-ST Interconnect Component in an SOPC Builder System

For details about the system interconnect fabric, refer to the *System Interconnect Fabric for Streaming Interfaces* chapter in volume 4 of the *Quartus II Handbook*. For details about the Avalon-ST interface protocol, refer to the *Avalon Interface Specifications*.

Figure 12-2 illustrates a datapath that connects a Triple Speed Ethernet MegaCore function to a Scatter-Gather DMA controller core using a timing adapter, data format adapter, and channel adapter so that the cores can interoperate.

Address Mapping

Figure 12–2. Avalon-ST Datapath Constructed Using Avalon Streaming Interconnect Components



The control and status signals for the components containing source or sink interfaces can be mapped to a slave interface which is then accessible in the global Avalon address space.

Timing Adapter

The timing adapter has two functions:

- It adapts source and sink interfaces that support the `ready` signal and those that do not.
- It adapts source and sink interfaces that support the `valid` signal and those that do not.
- It adapts source and sink interfaces that have different ready latencies.

The timing adapter treats all signals other than the `ready` and `valid` signals as payload, and simply drives them from the source to the sink. [Table 12–1](#) outlines the adaptations that the timing adapter provides.

Table 12-1. Timing Adapter

Condition	Adaptation
The source has <code>ready</code> , but the sink does not.	In this case, the source can respond to backpressure, but the sink never needs to apply it. The <code>ready</code> input to the source interface is connected directly to logical 1.
The source does not have <code>ready</code> , but the sink does.	The sink may apply backpressure, but the source is unable to respond to it. There is no logic that the adapter can insert that prevents data loss when the source asserts <code>valid</code> but the sink is not ready. The adapter provides simulation time error messages and an error indication if data is ever lost. The user is presented with a warning, and the connection is allowed.
The source and sink both support backpressure, but the sink's ready latency is greater than the source's.	The source responds to <code>ready</code> assertion or deassertion faster than the sink requires it. A number of pipeline stages equal to the difference in ready latency are inserted in the <code>ready</code> path from the sink back to the source, causing the source and the sink to see the same cycles as <code>ready</code> cycles.
The source and sink both support backpressure, but the sink's ready latency is less than the source's.	The source cannot respond to <code>ready</code> assertion or deassertion in time to satisfy the sink. A buffer whose depth is equal to the difference in ready latency is inserted to compensate for the source's inability to respond in time.

Resource Usage and Performance

Resource utilization for the timing adapter depends upon the function that it performs. [Table 12-2](#) provides estimated resource utilization for seven different configurations of the timing adapter

Table 12-2. Timing Adapter Estimated Resource Usage and Performance

Input Ready Latency	Output Ready Latency	Stratix® II and Stratix II GX (Approximate LEs)			Cyclone® II		Stratix (Approximate LEs)		
		f _{MAX} (MHz)	ALM Count	Mem Bits	f _{MAX} (MHz)	Logic Cells	f _{MAX} (MHz)	Logic Cells	Mem Bits
1	2	500	2	0	420	2	422	1	0
1	3	500	2	0	420	3	422	2	0
1	4	500	4	0	420	4	422	3	0
1	0	500	21	80	420	183	422	20	80
2	1	456	21	80	401	188	317	21	80
3	1	456	21	80	401	188	317	21	80
4	1	456	21	80	401	188	317	21	80

Instantiating the Timing Adapter in SOPC Builder

You can use the Avalon-ST configuration wizard in SOPC Builder to specify the hardware features. [Table 12-3](#) describes the options available on the **Parameter Settings** page of the configuration wizard

Table 12-3. Avalon-ST Timing Adapter Parameters

Input Interface Parameters	
Parameter	Description
Support Backpressure with the ready signal	Turn on this option to add the backpressure functionality to the interface.
Ready Latency	When the <code>ready</code> signal is used, the value for <code>ready_latency</code> indicates the number of cycles between when the <code>ready</code> signal is asserted and when valid data is driven.
Include valid signal	Turn this option on if the interface includes the <code>valid</code> signal. Turning this option off means that data being received is always valid.
Output Interface Parameters	
Support Backpressure with the ready signal	Turn on this option to add the backpressure functionality to the interface.
Ready Latency	When the <code>ready</code> signal is used, the value for <code>ready_latency</code> indicates the number of cycles between when the <code>ready</code> signal is asserted and when valid data is driven.
Include valid signal	Turn this option on if the interface includes the <code>valid</code> signal. Turning this option off means that data driven is always valid.
Common to Input and Output Interfaces	
Channel Signal Width (bits)	Type the width of the <code>channel</code> signal. A channel width of 4 allows up to 16 channels. The maximum width of the <code>channel</code> signal is eight bits. Set to 0 if channels are not used.
Max Channel	Type the maximum number of channels that the interface supports. Valid values are 0–255.
Data Bits Per Symbol	Type the number of bits per symbol.
Data Symbols Per Beat	Type the number of symbols per active transfer.
Include Packet Support	Turn this option on if the interfaces supports a packet protocol, including the <code>startofpacket</code> , <code>endofpacket</code> and <code>empty</code> signals.
Include Empty Signal	You can use this signal to specify the number of empty symbols in the cycle that includes the <code>endofpacket</code> signal. This signal is not necessary if the number of symbols per beat is 1.
Error Signal Width (Bits)	Type the width of the error signal. Valid values are 0–31 bits. Type 0 if the error signal is not used.
Error Signal Description	Type the description for each of the error bits. Separate the description fields by semicolons. For a connection to be made, the description of the error bits in the source and sink must match. Refer to “Error Adapter” on page 12-9 for the adaptations that can be made when the bits do not match.

Data Format Adapter

The data format adapter handles interfaces that have different definitions for the data signal. One of the more common adaptations that this component performs is data width adaptation, such as converting a data interface that drives two, 8-bit symbols per beat to an interface that drives four, 8-bit symbols per beat. The available data format adaptations are listed in [Table 12-4](#).

Table 12-4. Data Format Adapter

Condition	Description of Adapter Logic
The source and sink's bits per symbol are different.	The connection cannot be made.
The source and sink have a different number of symbols per beat.	<p>The adapter converts from the source's width to the sink's width.</p> <p>If the adaptation is from a wider to a narrower interface, a beat of data at the input corresponds to multiple beats of data at the output. If the input <code>error</code> signal is asserted for a single beat, it is asserted on output for multiple beats.</p> <p>If the adaptation is from a narrow to a wider interface, multiple input beats are required to fill a single output beat, and the output <code>error</code> is the logical OR of the input <code>error</code> signal.</p>

Resource Usage and Performance

Resource utilization for the data format adapter depends upon the function that it performs. [Table 12-5](#) provides estimated resource utilization for numerous configurations of the data format adapter.

Table 12-5. Data Format Adapter Estimated Resource Usage and Performance, 8 Bits per Symbol (Sheet 1 of 2)

Input Symbols per Beat	Output Symbols per Beat	Number of Channels	Packet Support	Stratix II and Stratix II GX (Approximate LEs)			Cyclone II			Stratix (Approximate LEs)		
				f _{MAX} (MHz)	ALM Count	Memory Bits	f _{MAX} (MHz)	Logic Cells	Memory Bits	f _{MAX} (MHz)	Logic Cells	Memory Bits
1	2	1	y	500	96	0	391	93	0	375	105	0
4	1	1	y	459	106	0	311	97	0	306	76	0
4	2	1	y	500	118	0	343	107	0	326	85	0
4	8	1	y	437	326	0	346	370	0	303	330	0
4	16	1	y	357	930	0	264	1005	0	231	806	0
1	2	188	y	321	110	15	187	137	15	209	153	15
4	1	105	y	244	125	2	148	183	2	150	137	2
4	2	105	y	277	101	2	172	134	2	173	108	2
4	8	130	y	322	255	41	175	279	41	187	262	41
4	16	30	y	268	341	106	166	563	106	153	471	106
4	1	105	n	269	107	2	177	185	2	167	99	2
4	2	54	n	290	109	1	193	203	1	176	91	1
4	3	10	n	249	149	18	189	251	16	159	217	18
4	5	222	n	281	300	40	199	381	40	182	316	40
4	6	30	n	312	184	40	201	385	40	198	241	40
4	7	139	n	253	285	56	159	416	56	161	427	56

Table 12-5. Data Format Adapter Estimated Resource Usage and Performance, 8 Bits per Symbol (Sheet 2 of 2)

Input Symbols per Beat	Output Symbols per Beat	Number of Channels	Packet Support	Stratix II and Stratix II GX (Approximate LEs)			Cyclone II			Stratix (Approximate LEs)		
				f _{MAX} (MHz)	ALM Count	Memory Bits	f _{MAX} (MHz)	Logic Cells	Memory Bits	f _{MAX} (MHz)	Logic Cells	Memory Bits
4	8	198	n	311	281	40	190	247	40	198	257	40
4	15	160	n	259	370	121	165	733	121	149	697	121
4	16	36	n	227	255	105	391	93	0	146	491	105

Instantiating the Data Format Adapter in SOPC Builder

You can use the Avalon-ST configuration wizard in SOPC Builder to specify the hardware features. [Table 12-6](#) describes the options available on the **Parameter Settings** page of the configuration wizard.

Table 12-6. Data Format Adapter Parameters

Input Interface Parameters	
Parameter	Description
Data Symbols Per Beat	Type the number of symbols transferred per active cycle.
Include the empty signal	Turn this option on if the cycle that includes the <code>endofpacket</code> signal can include empty symbols. This signal is not necessary if the number of symbols per beat is 1.
Output Interface Parameters	
Data Symbols Per Beat	Type the number of symbols transferred per active cycle.
Include the empty signal	Turn this option on if the cycle that includes the <code>endofpacket</code> signal can include empty symbols. This signal is not necessary if the number of symbols per beat is 1.
Common to Input & Output	
Channel Signal Width (bits)	Type the width of the <code>channel</code> signal. A channel width of 4 allows up to 16 channels. The maximum width of the channel signal is 8 bits. Type 0 if you do not need to send channel numbers.
Max Channel	Type the maximum number of channels that the interface supports. Valid values are 0-255.
Include Packet Support	Turn this option on if the interface supports a packet protocol, including the <code>startofpacket</code> , <code>endofpacket</code> , and <code>empty</code> signals.
Error Signal Width (Bits)	Type the width of the error signal. Valid values are 0-31 bits. Type 0 if the error signal is not used.
Error Signal Description	Type the description for each of the error bits. Separate the description fields by semicolons. For a connection to be made, the description of the error bits in the source and sink must match. Refer to “Error Adapter” on page 12-9 for the adaptations that can be made when the bits do not match.
Data Bits Per Symbol	Type the number of bits per symbol.

Channel Adapter

The channel adapter provides adaptations between interfaces that have different support for the `channel` signal or for the maximum number of channels supported. The adaptations are described in [Table 12-7](#).

Table 12-7. Channel Adapter

Condition	Description of Adapter Logic
The source uses channels, but the sink does not.	You are given a warning at generation time. The adapter provides a simulation error and signals an error for data for any channel from the source other than 0.
The sink has channel, but the source does not.	You are given a warning, and the channel inputs to the sink are all tied to a logical 0.
The source and sink both support channels, and the source's maximum number of channels is less than the sink's.	The source's channel is connected to the sink's channel unchanged. If the sink's channel signal has more bits, the higher bits are tied to a logical 0.
The source and sink both support channels, but the source's maximum number of channels is greater than the sink's.	The source's channel is connected to the sink's channel unchanged. If the source's channel signal has more bits, the higher bits are left unconnected. You are given a warning that channel information may be lost. An adapter provides a simulation error message and an error indication if the value of channel from the source is greater than the sink's maximum number of channels. In addition, the <code>valid</code> signal to the sink is deasserted so that the sink never sees data for channels that are out of range.

Resource Usage and Performance

The channel adapter typically uses fewer than 30 LEs. Its frequency is limited by the maximum frequency of the device you choose.

Instantiating the Channel Adapter in SOPC Builder

You can use the Avalon-ST configuration wizard in SOPC Builder to specify the hardware features. [Table 12-8](#) describes the options available on the **Parameter Settings** page of the configuration wizard.

Table 12-8. Avalon-ST Channel Adapter Parameters (Sheet 1 of 2)

Parameter	Description
Input Interface Parameters	
Channel Signal Width (bits)	Type the width of the <code>channel</code> signal. A channel width of 4 allows up to 16 channels. The maximum width of the <code>channel</code> signal is eight bits. Set to 0 if channels are not used.
Max Channel	Type the maximum number of channels that the interface supports. Valid values are 0-255.
Output Interface Parameters	
Channel Signal Width (bits)	Type the width of the <code>channel</code> signal. A channel width of 4 allows up to 16 channels. The maximum width of the <code>channel</code> signal is eight bits. Set to 0 if channels are not used.
Max Channel	Type the maximum number of channels that the interface supports. Valid values are 0-255.

Table 12-8. Avalon-ST Channel Adapter Parameters (Sheet 2 of 2)

Parameter	Description
Common to Input and Output Interfaces	
Support Backpressure with the ready signal	Turn on this option to add the backpressure functionality to the interface.
Ready Latency	When the <code>ready</code> signal is used, the value for <code>ready_latency</code> indicates the number of cycles between when the <code>ready</code> signal is asserted and when valid data is driven.
Data Bits Per Symbol	Type the number of bits per symbol.
Data Symbols Per Beat	Type the number of symbols per active transfer.
Include Packet Support	Turn this option on if the interfaces supports a packet protocol, including the <code>startofpacket</code> , <code>endofpacket</code> and <code>empty</code> signals.
Include Empty Signal	You can use this signal to specify the number of empty symbols in the cycle that includes the <code>endofpacket</code> signal. This signal is not necessary if the number of symbols per beat is 1.
Error Signal Width (bits)	Type the width of the error signal. Valid values are 0–31 bits. Type 0 if you do not need to send error values.
Error Signal Description	Type the description for each of the error bits. Separate the description fields by semicolons. For a connection to be made, the description of the error bits in the source and sink must match. Refer to “ Error Adapter ” on page 12-9 for the adaptations that can be made when the bits do not match.

Error Adapter

The error adapter ensures that per-bit error information provided by source interfaces is correctly connected to the sink interface's input error signal. The adaptations are described in [Table 12-9](#):

Instantiating the Error Adapter in SOPC Builder

You can use the Avalon-ST configuration wizard in SOPC Builder to specify the hardware features. [Table 12-9](#) describes the options available on the **Parameter Settings** page of the configuration wizard.

Table 12-9. Avalon-ST Error Adapter Parameters

Parameter	Description
Input Interface Parameters	
Error Signal Width (bits)	Type the width of the error signal. Valid values are 0–31 bits. Type 0 if the error signal is not used.
Error Signal Description	Type the description for each of the error bits. Separate the description fields by semicolons. For a connection to be made, the description of the error bits in the source and sink must match. Refer to “ Error Adapter ” on page 12-9 for the adaptations that can be made when the bits do not match.
Output Interface Parameters	
Error Signal Width (bits)	Type the width of the error signal. Valid values are 0–31 bits. Type 0 if you do not need to send error values.

Table 12-9. Avalon-ST Error Adapter Parameters

Parameter	Description
Error Signal Description	Type the description for each of the error bits. Separate the description fields by semicolons. For a connection to be made, the description of the error bits in the source and sink must match. Refer to “Error Adapter” on page 12-9 for the adaptations that can be made when the bits do not match.
Common to Input and Output Interfaces	
Support Backpressure with the ready signal	Turn on this option to add the backpressure functionality to the interface.
Ready Latency	When the <code>ready</code> signal is used, the value for <code>ready_latency</code> indicates the number of cycles between when the ready signal is asserted and when valid data is driven.
Channel Signal Width (bits)	Type the width of the <code>channel</code> signal. A channel width of 4 allows up to 16 channels. The maximum width of the <code>channel</code> signal is eight bits. Set to 0 if channels are not used.
Max Channel	Type the maximum number of channels that the interface supports. Valid values are 0-255.
Data Bits Per Symbol	Type the number of bits per symbol.
Data Symbols Per Beat	Type the number of symbols per active transfer.
Include Packet Support	Turn this option on if the interfaces supports a packet protocol, including the <code>startofpacket</code> , <code>endofpacket</code> and <code>empty</code> signals.
Include Empty Signal	Turn this option on if the cycle that includes the <code>endofpacket</code> signal can include empty symbols. This signal is not necessary if the number of symbols per beat is 1.

Installation and Licensing

The Avalon-ST interconnect components are included in the Altera MegaCore® IP Library, which is part of the Quartus II software installation. After you install the MegaCore IP Library, SOPC Builder recognizes these components and can instantiate them into a system.

You can use the Avalon-ST components without a license in any design that targets an Altera device.

Hardware Simulation Considerations

The Avalon-ST interconnect components do not provide a simulation testbench for simulating a stand-alone instance of the component. However, you can use the standard SOPC Builder simulation flow to simulate the component design files inside an SOPC Builder system.

Software Programming Model

The Avalon-ST interconnect components do not have any control or status registers that you can see. Therefore, software cannot control or configure any aspect of the interconnect components at run-time. These components cannot generate interrupts.

Document Revision History

Table 12-10 shows the revision history for this chapter.

Table 12-10. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2009, v9.1.0	<ul style="list-style-type: none"> ■ No changes from previous release. 	—
March 2009, v9.0.0	<ul style="list-style-type: none"> ■ No changes from previous release. 	—
November 2008, v8.1.1	<ul style="list-style-type: none"> ■ Removed private comments 	—
November 2008, v8.1.0	<ul style="list-style-type: none"> ■ Added documentation for Avalon-ST error adapter. ■ Reformatted parameter settings in tables. ■ Changed page size to 8.5 x 11 inches. 	Minor changes for 8.1.
May 2008, v8.0.0	<ul style="list-style-type: none"> ■ Chapter renumbered from 11 to 12. ■ Deleted references to Avalon Memory-Mapped and Streaming Interface Specifications and changed to Avalon Interface Specifications. 	—
October 2007, v7.2.0	<ul style="list-style-type: none"> ■ No changes to this release. 	—
May 2007, v7.1.0	<ul style="list-style-type: none"> ■ Initial release. 	—

