- The possible widths of the signal
- The direction of the signal from the perspective of the peripheral
- Whether or not the signal type is required on an Avalon-MM port
- A brief description of the purpose and function of the signal type, and any special usage requirements

Table 1 and Table 2 categorize each signal by the transfer property that uses the signal. For details, refer to Transfer Properties on page 26.

Table 1. Avalon-MM Slave Port Signals

Signal Type	Width	Direction	Required	Description
Fundamental Signals				
clk	1	In	No	Synchronization clock for the Avalon-MM slave interface. All signals are synchronous to `clk`. Asynchronous slave ports can omit `clk`.
chipselect	1	In	No	Chip-select signal to the slave port. The slave port ignores all other Avalon-MM signal inputs unless `chipselect` is asserted.
address	1-32	In	No	Address lines from the system interconnect fabric to the slave port. Specifies a word offset into the slave address space.
read	1	In	No	Read-request signal to the slave port. Not required if the slave port never outputs data. If used, `readdata` or `data` must also be used.
readdata	1-1024 (1)(2)	Out	No	Data lines to the system interconnect fabric for read transfers. Not required if the slave port never outputs data. If used, `data` cannot be used.

Table 1. Avalon-MM Slave Port Signals

Signal Type	Width	Direction	Required	Description
Fundamental Signals				
write	1	In	No	Write-request signal to the slave port. Not required if the slave port never receives data from a master. If used, `writedata` or `data` must also be used, and `writebyteenable` cannot be used.
writedata	1-1024 (1)(2)	In	No	Data lines from the system interconnect fabric for write transfers. Not required if the slave port never receives data. If used, `write` or `writebyteenable` must also be used, and `data` cannot be used.
byteenable	2,4,8, 16, 32, 64, 128	In	No	Byte-enable signals to enable specific byte lane(s) during transfers on ports of width greater than 8 bits. If used, `writedata` must also be used, and `writebyteenable` cannot be used.
writebyteenable	2,4,8,16, 32, 64, 128	In	No	Equivalent to the logical AND of the `byteenable` and `write` signals. If used, `writedata` must also be used. `write` and `byteenable` cannot be used.
begintransfer	1	In	No	Asserted during the first cycle of every transfer. Usage is peripheral-specific.
Wait-State Signals				

Table 1. Avalon-MM Slave Port Signals

Signal Type	Width	Direction	Required	Description
Fundamental Signals				
waitrequest	1	Out	No	Used to stall the system interconnect fabric when the slave port is not able to respond immediately.
Pipeline Signals				
readdatavalid	1	Out	No	Used for pipelined read transfers with variable latency. Marks the rising clock edge when the slave asserts valid `readdata`.
Burst Signals				
burstcount	2-32	In	No	Used for burst transfers. Indicates the number of transfers in a burst. When used, `waitrequest` must also be used.
beginbursttransfer	1	In	No	Asserted for the first cycle of a burst to indicate when a burst transfer is starting. Usage is peripheral-specific.
Flow Control Signals				
readyfordata	1	Out	No	Used for transfers with flow control. Indicates that the peripheral is ready for a write transfer.
dataavailable	1	Out	No	Used for transfers with flow control. Indicates that the peripheral is ready for a read transfer.
endofpacket	1	Out	No	Used for transfers with flow control. Indicates an end-of-packet condition to the system interconnect fabric. Implementation is peripheral specific.

Table 1. Avalon-MM Slave Port Signals

Signal Type	Width	Direction	Required	Description
Fundamental Signals				
Tristate Signals				
data	1-1024 (1)	Bi-directional	No	Bidirectional read and write data for tristate slave ports. If used, `readdata` and `writedata` cannot be used.
outputenable	1	In	No	Output-enable signal for the `data` lines. When deasserted, tristate slave port must not drive its `data` lines. If used, `data` must also be used.
Other Signals				
irq	1	Out	No	Interrupt request. A slave port asserts `irq` when it needs to be serviced by a master.
reset	1	In	No	Peripheral reset signal. When asserted, slave peripheral must enter a deterministic reset state.
resetrequest	1	Out	No	Allows the peripheral to reset the entire Avalon-MM system. The result is immediate.

Notes to Table 1:
(A) If the slave port uses dynamic bus sizing, this signal's width must be a power of two.
(B) If a slave port uses both `readdata` and `writedata`, the width of both signals must be equal.

The Avalon-MM interface specification does not mandate the presence of any particular signal in an Avalon-MM slave port.

Table 2. Avalon-MM Master Port Signals

Signal Type	Width	Direction	Required	Description
Fundamental Signals				
clk	1	In	Yes	Synchronization clock for the Avalon-MM slave interface. All signals are synchronous to `clk`.
waitrequest	1	In	Yes	Forces the master port to wait until the system interconnect fabric is ready to proceed with the transfer.
address	1-32	Out	Yes	Address lines from the master port to the system interconnect fabric. The `address` signal represents a byte address. However, the master port must assert `address` on word boundaries only.
read	1	Out	No	Read request signal from master port. Not required if master port never performs read transfers. If used, `readdata` or `data` must also be used.
readdata	8,16,32,64, 128, 256, 512, 1024 (1)	In	No	Data lines from the system interconnect fabric for read transfers. Not required if the master port never performs read transfers. If used, `read` must also be used, and `data` cannot be used.
write	1	Out	No	Write request signal from master port. Not required if the master port never performs write transfers. If used, `writedata` or `data` must also be used.

Table 2. Avalon-MM Master Port Signals

Signal Type	Width	Direction	Required	Description
Fundamental Signals				
writedata	8,16,32,64, 128, 256, 512, 1024 (1)	Out	No	Data lines to the system interconnect fabric for write transfers. Not required if the master port never performs write transfers. If used, `write must` also be used, and data cannot be used.
byteenable	2,4,8, 16, 32, 64, 128	Out	No	Byte-enable signals to enable specific byte lane(s) during transfers on ports of width greater than 8 bits. The master port must assert all `byteenable` lines during read transfers.
Pipeline Signals				
readdatavalid	1	In	No	Used for pipelined read transfers with latency. Indicates that valid data from the system interconnect fabric is present on the `readdata` lines. Required if the master is pipelined.
flush	1	Out	No	Used for pipelined read transfers. The master port asserts `flush` to clear any pending transfers in the pipeline.
Burst Signals				
burstcount	2-32	Out	No	Used for burst transfers. Indicates the number of transfers in a burst.
Flow Control Signals				

Table 2. Avalon-MM Master Port Signals

Signal Type	Width	Direction	Required	Description
Fundamental Signals				
endofpacket	1	In	No	Used for transfers with flow control. Indicates an end-of-packet condition from the system interconnect fabric. Implementation is peripheral specific.
Tristate Signals				
data	8,16,32,64, 128, 256, 512, 1024			Bidirectional read and write data for tristate master ports. If used, `readdata` and `writedata` cannot be used.
Other Signals				
irq	1, 32	In	No	Indicates when one or more slave ports have requested an interrupt. If `irq` is a 32-bit vector, each line corresponds directly to the `irq` signal on a slave port, with no inherent assumption of priority. If `irq` is one bit wide, it is the logical OR of all slave `irq` signals, and the interrupt priority is encoded on `irqnumber`.
irqnumber	6	In	No	Indicates the interrupt priority of a slave port asserting its interrupt request. Lower value means higher priority. Used only when the `irq` signal is one bit wide.
reset	1	In	No	Global reset signal. Implementation is peripheral specific.

Table 2. Avalon-MM Master Port Signals

Signal Type	Width	Direction	Required	Description
Fundamental Signals				
resetrequest	1	Out	No	Allows the peripheral to reset the entire Avalon-MM system. The result is immediate.

Note to Table 2:
(A) If a master port uses both `readdata` and `writedata`, the width of both signals must be equal.

The Avalon-MM interface specification only mandates the existence of three signals on an Avalon-MM master port: `clk`, `address`, and `waitrequest`.

2.2. Signal Polarity

The signal types listed in Table 1 and Table 2 are active high. The Avalon-MM interface also offers the negated version of each signal type, indicated by appending `_n` to the signal type name (e.g., `irq_n`, `read_n`). This is useful for interfacing to off-chip peripherals that use active-low logic.

2.3. Signal Naming Conventions

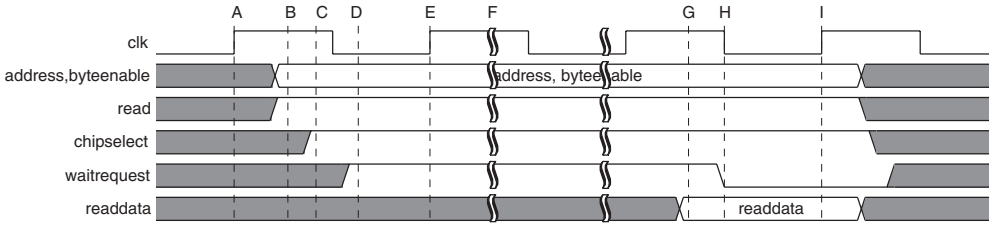
The Avalon-MM interface specification does not dictate a naming convention for the signals that appear on Avalon-MM peripherals. A signal in an Avalon-MM port can be named the same as its signal type, or it can be named differently to comply with a system-wide naming convention. For example, an Avalon-MM peripheral may have an Avalon-MM slave port with an input signal named `clock_100mhz` of type `clk`.

In the discussion of Avalon-MM transfers in this document, the signal names are the same as the signal type, but this naming convention is not part of the Avalon-MM interface specification.

2.4. Signal Sequencing and Timing

This section describes issues related to timing and sequencing of Avalon-MM signals.

Figure 7. Slave Read Transfer with Variable Wait-States



Notes to Figure 7:

(A) First cycle starts on the rising edge of `clk`.
(B) System interconnect fabric asserts `address` and `read` signals.
(C) System interconnect fabric decodes `address` then asserts `chipselect`.
(D) Slave port asserts `waitrequest` before the next rising edge of `clk`.
(E) System interconnect fabric samples `waitrequest` at the rising edge of `clk`; `waitrequest` is asserted, and therefore `readdata` is not captured on this clock edge.
(F) With `waitrequest` asserted throughout, an undefined number of cycles elapse.
(G) Slave port presents valid `readdata`.
(H) Slave port deasserts `waitrequest`.
(I) System interconnect fabric captures `readdata` on the next rising edge of `clk`, and the read transfer ends here. The next cycle begins here and could be the start of another transfer.

3.2.2.3. Restrictions

The following restrictions apply to ports that use wait-states:

- If a port that uses variable wait-states is capable of both read and write transfers, the port must use variable wait-states for both read and write transfers.
- If variable wait-states are specified, the slave port cannot also use setup and hold properties. In almost all cases, a peripheral that can generate the `waitrequest` signal will be on-chip and synchronous, making setup- and hold-time considerations unnecessary.

3.2.3. Setup Time

Some peripherals, most commonly asynchronous, off-chip devices, require `address` and `chipselect` signals to be stable for a period of time before the `read` signal is asserted. Avalon-MM transfers with setup time accommodate for such setup time requirements. The signals used for a read transfer with setup time are identical to those used for the fundamental read transfer. The difference is in the timing of signals only.

A nonzero setup time of means that, after the system interconnect fabric asserts `address` and `chipselect` to the slave port, there is a delay of *N* cycles before it asserts `read`. The total number of cycles to complete the