The following sections detail the standard handshaking that takes place when a 32-bit processor communicates to a slave peripheral or memory device connected to the relevant Wishbone interface port. Both of the processor's Wishbone ports can be configured for 8-, 16- or 32-bit data transfer, depending on the width of the data bus supported by the connected slave device. Configuration is achieved using the relevant IO\_SEL\_O or ME\_SEL\_O output, which defines where on the corresponding DAT\_O and DAT\_I lines the data appears when writing and reading respectively.

## Writing to a Slave Wishbone Peripheral Device

Data is written from the host processor (Wishbone Master) to a Wishbone-compliant peripheral device (Wishbone Slave) in accordance with the standard Wishbone data transfer handshaking protocol. This data transfer cycle can be summarized as follows:

* The host presents an address on its IO\_ADR\_O output for the register it wants to write to and valid data on its IO\_DAT\_O output. It then asserts its IO\_WE\_O output to specify a Write cycle
* The host defines where the data will be sent on the IO\_DAT\_O line using its IO\_SEL\_O signal
* The slave device receives the address at its ADR\_I input and prepares to receive the data
* The host asserts its IO\_STB\_O and IO\_CYC\_O outputs, indicating that the transfer is to begin. The slave device, monitoring its STB\_I and CYC\_I inputs, reacts to this assertion by latching the data appearing at its DAT\_I input into the requested register and asserting its ACK\_O signal – to indicate to the host that the data has been received
* The host, monitoring its IO\_ACK\_I input, responds by negating the IO\_STB\_O and IO\_CYC\_O signals. At the same time, the slave device negates the ACK\_O signal and the data transfer cycle is naturally terminated.

## Reading from a Slave Wishbone Peripheral Device

Data is read by the host processor (Wishbone Master) from a Wishbone-compliant peripheral device (Wishbone Slave) in accordance with the standard Wishbone data transfer handshaking protocol. This data transfer cycle can be summarized as follows:

* The host presents an address on its IO\_ADR\_O output for the register it wishes to read. It then negates its IO\_WE\_O output to specify a Read cycle
* The host defines where it expects the data to appear on its IO\_DAT\_I line using its IO\_SEL\_O signal
* The slave device receives the address at its ADR\_I input and prepares to transmit the data from the selected register
* The host asserts its IO\_STB\_O and IO\_CYC\_O outputs, indicating that the transfer is to begin. The slave device, monitoring its STB\_I and CYC\_I inputs, reacts to this assertion by presenting the valid data from the requested register at its DAT\_O output and asserting its ACK\_O signal – to indicate to the host that valid data is present
* The host, monitoring its IO\_ACK\_I input, responds by latching the data appearing at its IO\_DAT\_I input and negating the IO\_STB\_O and IO\_CYC\_O signals. At the same time, the slave device negates the ACK\_O signal and the data transfer cycle is naturally terminated.

## Writing to a Slave Wishbone Memory Device

Data is written from the host processor (Wishbone Master) to a Wishbone-compliant memory device or memory controller (Wishbone Slave) in accordance with the standard Wishbone data transfer handshaking protocol. This data transfer cycle can be summarized as follows:

* The host presents an address on its ME\_ADR\_O output for the address in memory that it wants to write to and valid data on its ME\_DAT\_O output. It then asserts its ME\_WE\_O output to specify a Write cycle
* The host defines where the data will be sent on the ME\_DAT\_O line using its ME\_SEL\_O signal
* The slave device receives the address at its ADR\_I input and prepares to receive the data
* The host asserts its ME\_STB\_O and ME\_CYC\_O outputs, indicating that the transfer is to begin. The slave device, monitoring its STB\_I and CYC\_I inputs, reacts to this assertion by storing the data appearing at its DAT\_I input at the requested address and asserting its ACK\_O signal – to indicate to the host that the data has been received
* The host, monitoring its ME\_ACK\_I input, responds by negating the ME\_STB\_O and ME\_CYC\_O signals. At the same time, the slave device negates the ACK\_O signal and the data transfer cycle is naturally terminated.

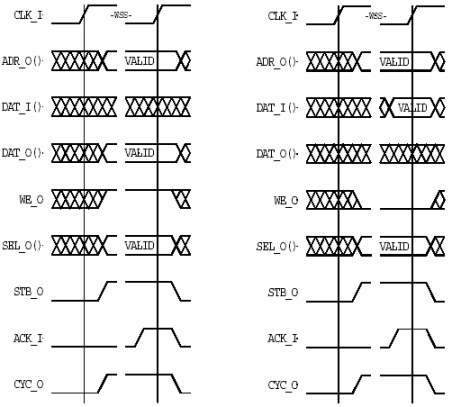
## Reading from a Slave Wishbone Memory Device

Data is read by the host processor (Wishbone Master) from a Wishbone-compliant memory device or memory controller (Wishbone Slave) in accordance with the standard Wishbone data transfer handshaking protocol. This data transfer cycle can be summarized as follows:

* The host presents an address on its ME\_ADR\_O output for the address in memory that it wishes to read. It then negates its ME\_WE\_O output to specify a Read cycle
* The host defines where it expects the data to appear on its ME\_DAT\_I line using its ME\_SEL\_O signal
* The slave device receives the address at its ADR\_I input and prepares to transmit the data from the selected memory location
* The host asserts its ME\_STB\_O and ME\_CYC\_O outputs, indicating that the transfer is to begin. The slave device, monitoring its STB\_I and CYC\_I inputs, reacts to this assertion by presenting the valid data from the requested memory location at its DAT\_O output and asserting its ACK\_O signal – to indicate to the host that valid data is present
* The host, monitoring its ME\_ACK\_I input, responds by latching the data appearing at its ME\_DAT\_I input and negating the ME\_STB\_O and ME\_CYC\_O signals. At the same time, the slave device negates the ACK\_O signal and the data transfer cycle is naturally terminated.

## Wishbone Timing

Figure 1 shows the signal timing for a standard single Wishbone Write Cycle (left) and Read Cycle (right), respectively. The timing diagrams are presented assuming point-to-point connection of the Master and Slave interfaces, with only signals on the Master side of the interface shown. Note that cycle speed can be throttled by the Slave device inserting wait states (represented as WSS on the diagrams) before asserting its acknowledgement line (ACK\_I input at the Master side).

   
Figure 1. Timing diagrams for single Wishbone Write (left) and Read (right) cycles.

* [Printer-friendly version](http://techdocs.altium.com/print/212714)
* [PDF version](http://techdocs.altium.com/printpdf/212714)