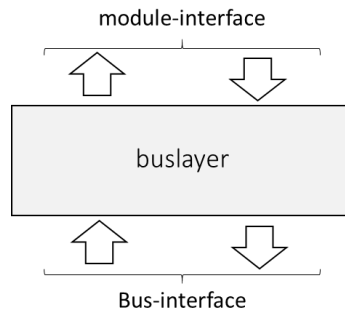


## BUSLAYER SPECIFICATION



The buslayer\_ (master and slave) module defines the glue logic between a module and a shared bus. It provides a module-interface (master and slave) and a bus-interface (master and slave). The module-interface provides a generic interface to perform read and write transactions in a bus regardless the bus protocol. The bus-interface is a wishbone compatible bus interface. It acts following the wishbone protocol to turn any read/write transaction coming from the module-interface into a read/write bus transaction following the wishbone protocol directives.

### MODULE-INTERFACE

#### *Signals common to Master and Slave module-interfaces (portName\_(direction))*

**DATA\_TO\_BUS\_O:** The data out array [DATA\_TO\_BUS\_O()] is used to pass binary data. The array boundaries are determined by the port size, with a maximum port size of 32-bits (e.g. [DATA\_TO\_BUS\_O(31..0)]).

**DATA\_FROM\_BUS\_I:** The data input array [DATA\_FROM\_BUS\_I()] is used to pass binary data. The array boundaries are determined by the port size, with a maximum port size of 32-bits (e.g. [DATA\_FROM\_BUS\_I(31..0)]).

**RESET\_O:** The reset output [RESET\_O] indicates that the WISHBONE bus interface is restarting. It is not required to reset other parts of an IP core (although it may be used that way).

#### *Master module signals (portName\_(direction))*

**REQUEST\_I:** The request input [REQUEST\_I], when high, indicates the beginning of a bus new request, and it must remain high until either the signal READY\_FROM\_BUS\_O or ERROR\_FROM\_BUS\_O is high.

**WRITE\_I:** The write enable input [WRITE\_I] indicates whether the current request is a READ or WRITE request. The signal is low during READ cycles and is high during WRITE cycles.

**ADDRESS\_I:** The address input array [ADDRESS\_I()] is used to pass a binary address. The higher array boundary is specific to the address width of the core, and the lower array boundary is determined by the data port size and granularity. For example the array size on a 32-bit data port with BYTE granularity is [ADDRESS\_I(n..2)].

**BYTE\_SEL\_I:** The select input array [BYTE\_SEL\_I()] indicates where valid data is placed on the [DATA\_TO\_BUS()] signal array during WRITE cycles, and where it should be present on the [DATA\_FROM\_BUS\_I()] signal array during READ cycles. The array boundaries are determined by the granularity of a port. For example, if 8-bit granularity is used on a 32-bit port, then there would be an array of eight select signals with boundaries of [BYTE\_SEL\_I(3..0)]. Each individual select signal correlates to one of four active bytes on the 32-bit data port.

**READY\_FROM\_BUS\_O:** The ready output [READY\_FROM\_BUS\_O], when high, indicates the normal termination of a request.

**ERROR\_FROM\_BUS\_O:** The error output [ERROR\_FROM\_BUS\_O] indicates an abnormal request termination. The source of the error, and the response generated is defined by the IP core supplier.

**BUSY\_O:** The busy output [BUAY\_O], when high, indicates that no new request can be begun.

### ***Slave signals (portName\_(direction))***

**REQUEST\_O:** The request input [REQUEST\_O], when high, indicates the beginning of a new request.

**WRITE\_O:** The write enable output [WRITE\_O] indicates whether the current local bus cycle is a READ or WRITE cycle. The signal is low during READ cycles and is high during WRITE cycles.

**ADDRESS\_O:** The address output array [ADDRESS\_O()] is used to pass a binary address. The higher array boundary is specific to the address width of the core, and the lower array boundary is determined by the data port size. For example, the array size on a 32-bit data port with BYTE granularity is [ADR\_O(n..2)].

**BYTE\_SEL\_O:** The select output array [BYTE\_SEL\_O()] indicates where valid data is placed on the [DATA\_FROM\_BUS()] signal array during WRITE cycles, and where it should be present on the [DATA\_TO\_BUS\_I()] signal array during READ cycles. The array boundaries are determined by the granularity of a port. For example, if 8-bit granularity is used on a 32-bit port, then there would be an array of eight select signals with boundaries of [BYTE\_SEL\_I(3..0)]. Each individual select signal correlates to one of four active bytes on the 32-bit data port.

**DONE\_I:** The acknowledge input [ACK\_I], when high, indicates the termination of a normal request.

**ERR\_I:** The error input [ERR\_I] indicates an abnormal request termination. The source of the error, and the response generated is defined by the IP core supplier.

## Read transaction (master)

The module-interface for a read transaction works as follows:

- CLOCK EDGE 0
  - MODULE presents a valid address on [ADDRESS\_I]
  - MODULE negates [WRITE\_I]
  - MODULE presents bank select [BYTE\_SEL\_I]
  - MODULE rises [REQUEST\_I]
  - BUSY\_O is low
- CLOCK EDGE 1
  - BUSLAYER rises BUSY\_O in response to high [REQUEST\_I], and starts a new bus READ CYCLE (see next section)

Normal termination of a request

- CLOCK EDGE N ( $N > 1$ )
  - BUSLAYER rises READY\_FROM\_BUS\_O
  - BUSLAYER presents valid data in DATA\_FROM\_BUS\_O
  - MODULE negates [REQUEST\_I] in response to high [READY\_FROM\_BUS\_O]
- CLOCK EDGE N + 1
  - BUSLAYER negates READY\_FROM\_BUS\_O
  - BUSLAYER negates BUSY\_O

Abnormal termination of a request

- CLOCK EDGE N ( $N > 1$ )
  - BUSLAYER rises ERROR\_FROM\_BUS\_O
  - MODULE negates [REQUEST\_I] in response to high [READY\_FROM\_BUS\_O]
- CLOCK EDGE N + 1
  - BUSLAYER negates ERROR\_FROM\_BUS\_O
  - BUSLAYER negates BUSY\_O

## Write transaction (master)

The module-interface for a write transaction works as follows:

- CLOCK EDGE 0
  - MODULE presents a valid address on [ADDRESS\_I]
  - MODULE presents a valid data on [DATA\_TO\_BUS\_I]
  - MODULE rises [WRITE\_I]
  - MODULE presents bank select [BYTE\_SEL\_I]
  - MODULE rises [REQUEST\_I]

- BUSY\_O is low
- CLOCK EDGE 1
  - BUSLAYER rises BUSY\_O in response to high [REQUEST\_I], and starts a new bus WRITE CYCLE (see next section)

Normal termination of a request

- CLOCK EDGE N ( $N > 1$ )
  - BUSLAYER rises READY\_FROM\_BUS\_O
  - MODULE negates [REQUEST\_I] in response to high [READY\_FROM\_BUS\_O]
- CLOCK EDGE N + 1
  - BUSLAYER negates READY\_FROM\_BUS\_O
  - BUSLAYER negates BUSY\_O

Abnormal termination of a request

- CLOCK EDGE N ( $N > 1$ )
  - BUSLAYER rises ERROR\_FROM\_BUS\_O
  - MODULE negates [REQUEST\_I] in response to high [READY\_FROM\_BUS\_O]
- CLOCK EDGE N + 1
  - BUSLAYER negates ERROR\_FROM\_BUS\_O
  - BUSLAYER negates BUSY\_O

### **Read transaction (slave)**

The module-interface for a read transaction works as follows:

- CLOCK EDGE 0
  - BUSLAYER presents a valid address on [ADDRESS\_O]
  - BUSLAYER negates [WRITE\_O]
  - BUSLAYER presents bank select [BYTE\_SEL\_O]
  - BUSLAYER rises [REQUEST\_O]

Normal termination of a request

- CLOCK EDGE N ( $N \geq 0$ )
  - MODULE rises [DONE\_I]
  - MODULE presents valid data on [DATA\_TO\_BUS\_I]
- CLOCK EDGE N + 1
  - BUSLAYER negates [REQUEST\_O]
  - MODULE negates [DONE\_I]

Abnormal termination of a request

- CLOCK EDGE N ( $N \geq 0$ )
  - MODULE rises [ERR\_I]
- CLOCK EDGE N + 1
  - BUSLAYER negates [REQUEST\_O]
  - MODULE negates [ERR\_I]

### **Read transaction (write)**

The module-interface for a read transaction works as follows:

- CLOCK EDGE 0
  - BUSLAYER presents a valid address on [ADDRESS\_O]
  - BUSLAYER presents a valid data on [DATA\_FROM\_BUS\_O]
  - BUSLAYER rises [WRITE\_O]
  - BUSLAYER presents bank select [BYTE\_SEL\_O]
  - BUSLAYER rises [REQUEST\_O]

Normal termination of a request

- CLOCK EDGE N ( $N \geq 0$ )
  - MODULE rises [DONE\_I]
- CLOCK EDGE N + 1
  - BUSLAYER negates [REQUEST\_O]
  - MODULE negates [DONE\_I]

Abnormal termination of a request

- CLOCK EDGE N ( $N \geq 0$ )
  - MODULE rises [ERR\_I]
- CLOCK EDGE N + 1
  - BUSLAYER negates [REQUEST\_O]
  - MODULE negates [ERR\_I]

## BUS-INTERFACE

### ***SYSCON Module Signal (portName\_(direction))***

**CLK\_O:** The system clock output [CLK\_O] is generated by the SYSCON module. It coordinates all activities for the internal logic within the WISHBONE interconnect. The INTERCON module connects the [CLK\_O] output to the [CLK\_I] input on MASTER and SLAVE interfaces.

**RST\_O:** The reset output [RST\_O] is generated by the SYSCON module. It forces all WISHBONE interfaces to restart. All internal self-starting state machines are forced into an initial state. The INTERCON connects the [RST\_O] output to the [RST\_I] input on MASTER and SLAVE interfaces.

### ***Signals common to Master and Slave interfaces (portName\_(direction))***

**CLK\_I:** The clock input [CLK\_I] coordinates all activities for the internal logic within the WISHBONE interconnect. All WISHBONE output signals are registered at the rising edge of [CLK\_I]. All WISHBONE input signals are stable before the rising edge of [CLK\_I].

**DAT\_I:** The data input array [DAT\_I()] is used to pass binary data. The array boundaries are determined by the port size, with a maximum port size of 64-bits (e.g. [DAT\_I(63..0)]).

**DAT\_O:** The data output array [DAT\_O()] is used to pass binary data. The array boundaries are determined by the port size, with a maximum port size of 64-bits (e.g. [DAT\_I(63..0)]).

**RST\_I:** The reset input [RST\_I] forces the WISHBONE interface to restart. Furthermore, all internal self-starting state machines will be forced into an initial state. This signal only resets the WISHBONE interface. It is not required to reset other parts of an IP core (although it may be used that way).

### ***Master signals (portName\_(direction))***

**ACK\_I:** The acknowledge input [ACK\_I], when high, indicates the normal termination of a bus cycle.

**ADR\_O:** The address output array [ADR\_O()] is used to pass a binary address. The higher array boundary is specific to the address width of the core, and the lower array boundary is determined by the data port size and granularity. For example the array size on a 32-bit data port with BYTE granularity is [ADR\_O(n..2)].

**CYC\_O:** The cycle output [CYC\_O], when high, indicates that a valid bus cycle is in progress. The signal is high for the duration of all bus cycles. The [CYC\_O] signal is high during the first data transfer and remains high until the last data transfer.

**STALL\_I:** The pipeline stall input [STALL\_I] indicates that current slave is not able to accept the transfer in the transaction queue. This signal is used in pipelined mode.

**ERR\_I:** The error input [ERR\_I] indicates an abnormal cycle termination. The source of the error, and the response generated by the MASTER is defined by the IP core supplier.

**SEL\_O:** The select output array [SEL\_O()] indicates where valid data is expected on the [DAT\_I()] signal array during READ cycles, and where it is placed on the [DAT\_O()] signal array during WRITE cycles. The array boundaries are determined by the granularity of a port. For example, if 8-bit granularity is used on a 64-bit port, then there would be an array of eight select signals with boundaries of [SEL\_O(7..0)]. Each individual select signal correlates to one of eight active bytes on the 64-bit data port.

**STB\_O:** The strobe output [STB\_O] indicates a valid data transfer cycle. It is used to qualify various other signals on the interface such as [SEL\_O()]. The SLAVE rises either the [ACK\_I], or [ERR\_I] signals in response to every assertion of the [STB\_O] signal.

**WE\_O:** The write enable output [WE\_O] indicates whether the current local bus cycle is a READ or WRITE cycle. The signal is low during READ cycles and is high during WRITE cycles.

### ***Slave signals (portName\_(direction))***

**ACK\_O:** The acknowledge output [ACK\_O], when high, indicates the termination of a normal bus cycle.

**ADR\_I:** The address input array [ADR\_I()] is used to pass a binary address. The higher array boundary is specific to the address width of the core, and the lower array boundary is determined by the data port size. For example, the array size on a 32-bit data port with BYTE granularity is [ADR\_O(n..2)].

**CYC\_I:** The cycle input [CYC\_I], when high, indicates that a valid bus cycle is in progress. The signal is high for the duration of all bus cycles. The [CYC\_I] signal is high during the first data transfer and remains high until the last data transfer.

**STALL\_O:** The pipeline stall signal [STALL\_O] indicates that the slave cannot accept additional transactions in its queue. This signal is used in pipelined mode.

**ERR\_O** The error output [ERR\_O] indicates an abnormal cycle termination. The source of the error, and the response generated by the MASTER is defined by the IP core supplier.

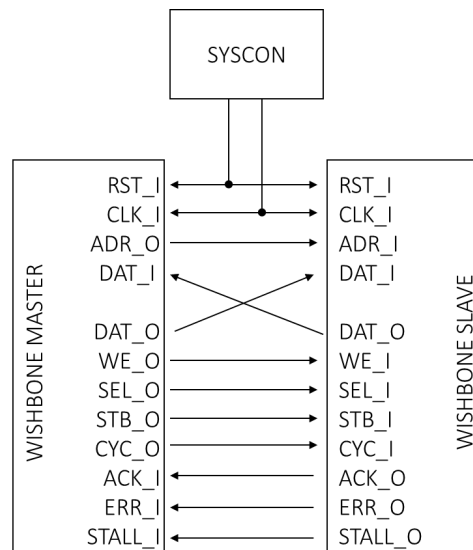
**SEL\_I:** The select input array [SEL\_I()] indicates where valid data is placed on the [DAT\_I()] signal array during WRITE cycles, and where it should be present on the [DAT\_O()] signal array during READ cycles. The array boundaries are determined by the granularity of a port. For example, if 8-bit granularity is used on a 64-bit port, then there would be an array of eight select signals with boundaries of [SEL\_I(7..0)]. Each individual select signal correlates to one of eight active bytes on the 64-bit data port.

**STB\_I:** The strobe input [STB\_I], when high, indicates that the SLAVE is selected. A SLAVE shall respond to other WISHBONE signals only when this [STB\_I] is high (except for the [RST\_I] signal which should always be responded to). The SLAVE rises either the [ACK\_O], or [ERR\_O] signals in response to every assertion of the [STB\_I] signal.

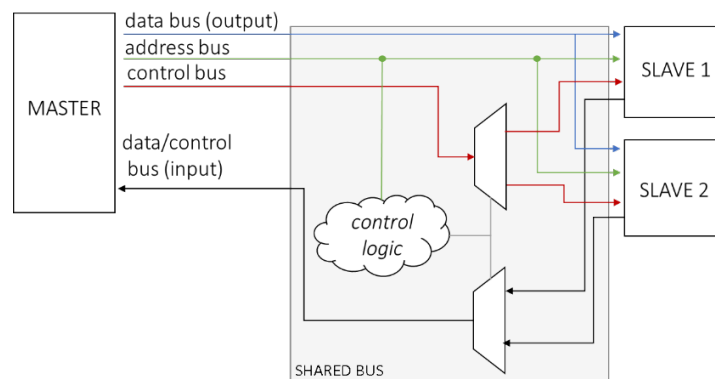
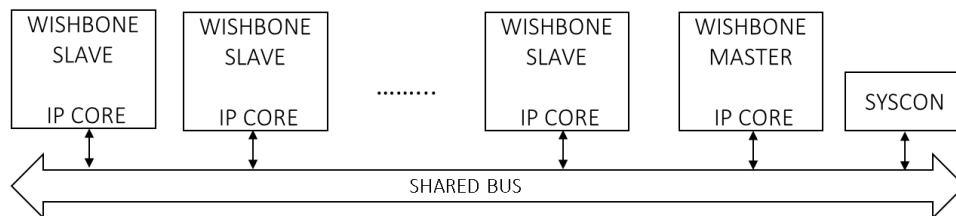
**WE\_I:** The write enable input [WE\_I] indicates whether the current local bus cycle is a READ or WRITE cycle. The signal is low during READ cycles and is high during WRITE cycles.

## Wishbone bus interconnections

### 1) point-to-point connection between 1 master and 1 slave



### 2) shared bus interconnection





## Wishbone bus read cycle

The bus protocol works as follows:

- CLOCK EDGE 0
  - MASTER presents a valid address on [ADR\_O]
  - MASTER negates [WE\_O]
  - MASTER presents bank select [SEL\_O]
  - MASTER rises [CYC\_O] and [STB\_O]

SLAVE presents valid data immediately

- CLOCK EDGE 1
  - responding SLAVE presents valid data on [DAT\_I]
  - responding SLAVE rises [ACK\_I] in response to [STB\_O] to indicate valid data
- CLOCK EDGE 2
  - MASTER latches data on [DAT\_I]
  - MASTER negates [STB\_O] and [CYC\_O] to indicate the end of the cycle
  - Responding SLAVE negates [ACK\_I] in response to low [STB\_O]

SLAVE does not present valid data immediately

- CLOCK EDGE 1
  - responding SLAVE rises [STALL\_I]
  - MASTER negates [STB\_O] in response to high [STALL\_I]
- CLOCK EDGE N
  - responding SLAVE presents valid data on [DAT\_I]
  - responding SLAVE negates [STALL\_I] and rises [ACK\_I] to indicate valid data
- CLOCK EDGE N+1
  - MASTER latches data on [DAT\_I]
  - MASTER negates [CYC\_O] to indicate the end of the cycle
  - responding SLAVE negates [ACK\_I] in response to low [CYC\_O]

SLAVE cannot present valid data

- CLOCK EDGE 1
  - responding SLAVE rises [ERR\_I] in response to [STB\_O] to indicate not available data
- CLOCK EDGE 2
  - MASTER negates [STB\_O] and [CYC\_O] to indicate the end of the cycle
  - Responding SLAVE negates [ERR\_I] in response to low [STB\_O]

## Wishbone bus write cycle

The bus protocol works as follows:

- CLOCK EDGE 0
  - MASTER presents a valid address on [ADR\_O]
  - MASTER presents a valid data on [DAT\_O]
  - MASTER rises [WE\_O]
  - MASTER presents bank select [SEL\_O]
  - MASTER rises [CYC\_O] and [STB\_O]

SLAVE consumes data immediately

- CLOCK EDGE 1
  - responding SLAVE rises [ACK\_I]
- CLOCK EDGE 2
  - MASTER negates [STB\_O] and [CYC\_O] to indicate the end of the cycle
  - Responding SLAVE negates [ACK\_I] in response to low [STB\_O]

SLAVE does not consume data immediately

- CLOCK EDGE 1
  - responding SLAVE rises [STALL\_I]
  - MASTER negates [STB\_O] in response to high [STALL\_I]
- CLOCK EDGE N
  - responding SLAVE negates [STALL\_I] and rises [ACK\_I] to indicate data was accepted
- CLOCK EDGE N+1
  - MASTER negates [CYC\_O] to indicate the end of the cycle
  - responding SLAVE negates [ACK\_I] in response to low [CYC\_O]

SLAVE cannot consume data

- CLOCK EDGE 1
  - responding SLAVE rises [ERR\_I] in response to [STB\_O]
- CLOCK EDGE 2
  - MASTER negates [STB\_O] and [CYC\_O] to indicate the end of the cycle
  - Responding SLAVE negates [ERR\_I] in response to low [STB\_O]