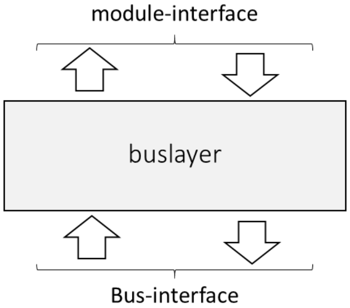
BUSLAYER SPECIFICATION



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_I(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. [DAT_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 asserted, indicates the beginning of a bus new request, and it must remain asserted until either the signal READY_FROM_BUS_O or ERROR_FROM_BUS_O is asserted.

WRITE_I: The write enable input [WRITE_I] indicates whether the current request is a READ or WRITE request. The signal is negated during READ cycles and is asserted during WRITE cycles.

ADDRESS_I: The address input array [ADDRESS()] 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)].

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 asserted, 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 asserted, indicates that no new request can be begun.

Slave signals (portName_(direction))

REQUEST_O: The request input [REQUEST_O], when asserted, 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 negated during READ cycles and is asserted 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 asserted, 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 asserts [REQUEST_I]

- CLOCK EDGE 1
  - BUSLAYER asserts BUSY_O in response to asserted [REQUEST_I], and starts a new bus READ CYCLE (see next section)

Normal termination of a request

- CLOCK EDGE N (N > 1)
  - BUSLAYER asserts READY_FROM_BUS_O
  - BUSLAYER presents valid data in DATA_FROM_BUS_0
  - MODULE negates [REQUEST_I] in response to asserted [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 asserts ERROR_FROM_BUS_O
  - MODULE negates [REQUEST_I] in response to asserted [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 asserts [WRITE_I]
  - MODULE presents bank select [BYTE_SEL_I]
  - MODULE asserts [REQUEST_I]

- CLOCK EDGE 1
  - BUSLAYER asserts BUSY_O in response to asserted [REQUEST_I], and starts a new bus WRITE CYCLE (see next section)

Normal termination of a request

- CLOCK EDGE N (N > 1)
  - BUSLAYER asserts READY_FROM_BUS_O
  - MODULE negates [REQUEST_I] in response to asserted [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 asserts ERROR_FROM_BUS_O
  - MODULE negates [REQUEST_I] in response to asserted [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 asserts [REQUEST_O]

Normal termination of a request

- CLOCK EDGE N (N >= 0)
  - MODULE asserts [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 >= 0)
  - MODULE asserts [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 asserts [WRITE_O]
  - BUSLAYER presents bank select [BYTE_SEL_O]
  - BUSLAYER asserts [REQUEST_O]

Normal termination of a request

- CLOCK EDGE N (N >= 0)
  - MODULE asserts [DONE_I]

- CLOCK EDGE N + 1
  - BUSLAYER negates [REQUEST_O]
  - MODULE negates [DONE_I]

Abnormal termination of a request

- CLOCK EDGE N (N >= 0)
  - MODULE asserts [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 asserted, 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 asserted, indicates that a valid bus cycle is in progress. The signal is asserted for the duration of all bus cycles. The [CYC_O] signal is asserted during the first data transfer and remains asserted 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 asserts 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 negated during READ cycles and is asserted during WRITE cycles.

Slave signals (portName_(direction))

ACK_O: The acknowledge output [ACK_O], when asserted, 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 asserted, indicates that a valid bus cycle is in progress. The signal is asserted for the duration of all bus cycles. The [CYC_I] signal is asserted during the first data transfer and remains asserted 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.

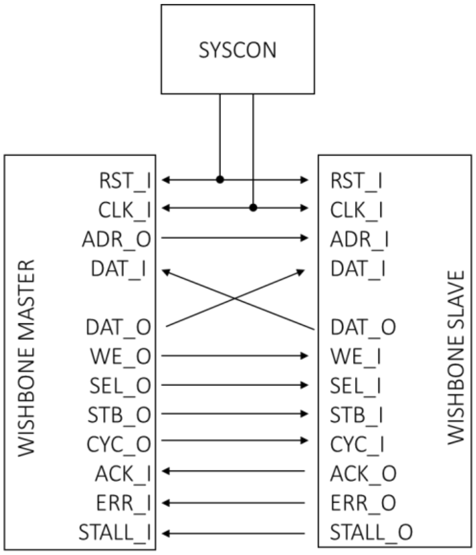
SEL_I: The select input array [SEL_I()] indicates where valid data is placed on the [DAT_I()] signal array during WRITE cycles, and where it should be present on the [DAT_O()] signal array during READ cycles. The array boundaries are determined by the granularity of a port. For example, if 8-bit granularity 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 asserted, indicates that the SLAVE is selected. A SLAVE shall respond to other WISHBONE signals only when this [STB_I] is asserted (except for the [RST_I] signal which should always be responded to). The SLAVE asserts either the [ACK_O], or [ERR_O] signals in response to every assertion of the [STB_I] signal.

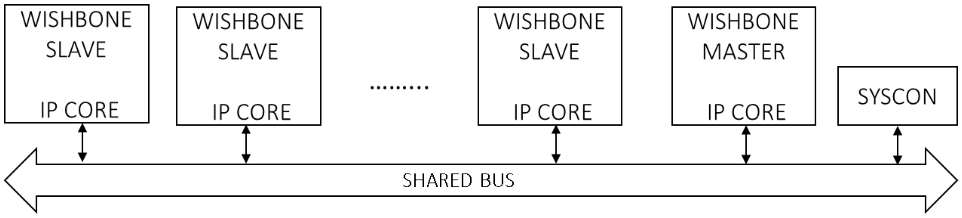
WE_I: The write enable input [WE_I] indicates whether the current local bus cycle is a READ or WRITE cycle. The signal is negated during READ cycles and is asserted during WRITE cycles.

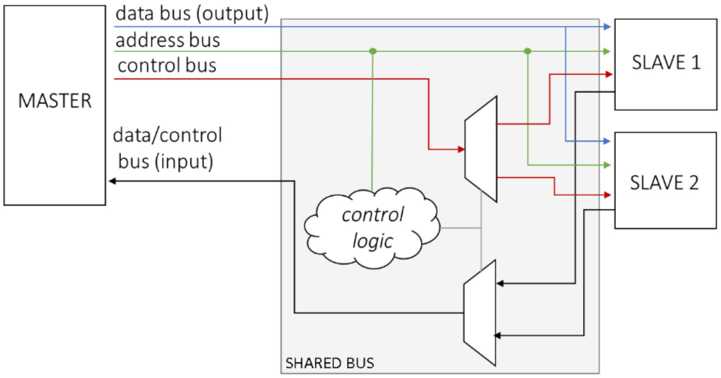
Wishbone bus interconnections

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 asserts [CYC_O] and [STB_O]

SLAVE presents valid data immediately

- CLOCK EDGE 1
  - responding SLAVE presents valid data on [DAT_I]
  - responding SLAVE asserts [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 negated [STB _O]

SLAVE does not present valid data immediately

- CLOCK EDGE 1
  - responding SLAVE asserts [STALL_I]
  - MASTER negates [STB_O] in response to asserted [STALL_I]
- CLOCK EDGE N
  - responding SLAVE presents valid data on [DAT_I]
  - responding SLAVE negates [STALL_I] and asserts [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 negated [CYC_O]

SLAVE cannot present valid data

- CLOCK EDGE 1
  - responding SLAVE asserts [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 negated [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 asserts [WE_O]
    - MASTER presents bank select [SEL_O]
    - MASTER asserts [CYC_O] and [STB_O]

SLAVE consumes data immediately

- CLOCK EDGE 1
    - responding SLAVE asserts [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 negated [STB _O]

SLAVE does not consume data immediately

- CLOCK EDGE 1
    - responding SLAVE asserts [STALL_I]
    - MASTER negates [STB_O] in response to asserted [STALL_I]
- CLOCK EDGE N
    - responding SLAVE negates [STALL_I] and asserts [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 negated [CYC_O]

SLAVE cannot consume data

- CLOCK EDGE 1
    - responding SLAVE asserts [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 negated [STB _O]