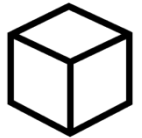


# AXI4-Lite BFM – Quick Reference

**BFM**



*axilite\_bfm\_pkg.vhd*

**axilite\_write** (addr\_value, data\_value, [byte\_enable], msg, clk, axilite\_if, [scope, [msg\_id\_panel, [config]]])

**Example:** axilite\_write(x"00456000", x"AA11", "Writing data to Peripheral 1", clk, axilite\_if); -- Without byte\_enable

**Example:** axilite\_write(C\_ADDR\_PERIPHERAL\_1, x"AA11", "01", "Writing data to Peripheral 1", clk, axilite\_if); --With byte\_enable

**Suggested usage:** axilite\_write(C\_ADDR\_DMA, x"AA11", "Writing data to DMA"); -- Suggested usage requires local overload (see section 5)

**axilite\_read** (addr\_value, data\_value, msg, clk, axilite\_if, [scope, [msg\_id\_panel, [config, [proc\_name]]]])

**Example:** axilite\_read(x"11355000", v\_data\_out, "Read from Peripheral 1", clk, axilite\_if);

**Suggested usage:** axilite\_read(C\_ADDR\_IO, v\_data\_out, "Read from IO"); -- Suggested usage requires local overload (see section 5)

**axilite\_check** (addr\_value, data\_exp, msg, clk, axilite\_if, [alert\_level, [scope, [msg\_id\_panel, [config]]]])

**Example:** axilite\_check(x"6840A000", x"3B16", "Check data from Peripheral 1", clk, axilite\_if);

**Suggested usage:** axilite\_check(C\_ADDR\_IO, x"3B16", "Check data from IO"); -- Suggested usage requires local overload (see section 5)

**init\_axilite\_if\_signals** (addr\_width, data\_width)

**Example:** axilite\_if <= init\_axilite\_if\_signals(addr\_width, data\_width);

## BFM Configuration record 't\_axilite\_bfm\_config'

| Record element             | Type                      | C_AXILITE_BFM_CONFIG_DEFAULT |
|----------------------------|---------------------------|------------------------------|
| max_wait_cycles            | natural                   | 10                           |
| max_wait_cycles_severity   | t_alert_level             | TB_FAILURE                   |
| clock_period               | time                      | 10 ns                        |
| clock_period_margin        | time                      | 0 ns                         |
| clock_margin_severity      | t_alert_level             | TB_ERROR                     |
| setup_time                 | time                      | 2.5 ns                       |
| hold_time                  | time                      | 2.5 ns                       |
| expected_response          | t_axilite_response_status | OKAY                         |
| expected_response_severity | t_alert_level             | TB_FAILURE                   |
| protection_setting         | t_axilite_protection      | UNPRIVILEGED_UNSECURE_DATA   |
| num_aw_pipe_stages         | natural                   | 1                            |
| num_w_pipe_stages          | natural                   | 1                            |
| num_ar_pipe_stages         | natural                   | 1                            |
| num_r_pipe_stages          | natural                   | 1                            |
| num_b_pipe_stages          | natural                   | 1                            |
| id_for_bfm                 | t_msg_id                  | ID_BFM                       |
| id_for_bfm_wait            | t_msg_id                  | ID_BFM_WAIT                  |
| id_for_bfm_poll            | t_msg_id                  | ID_BFM_POLL                  |

## Signal record 't\_axilite\_if'

| Record element         | Type                             |
|------------------------|----------------------------------|
| write_address_channel  | t_axilite_write_address_channel  |
| write_data_channel     | t_axilite_write_data_channel     |
| write_response_channel | t_axilite_write_response_channel |
| read_address_channel   | t_axilite_read_address_channel   |
| read_data_channel      | t_axilite_read_data_channel      |



## BFM non-signal parameters

| Name         | Type                 | Example(s)                         | Description  |
|--------------|----------------------|------------------------------------|--|
| addr_value   | unsigned             | x"125A"                            | The address of an AXI4-Lite accessible register.   |
| data_value   | std_logic_vector     | x"20D3"                            | The data value to be written to the addressed register   |
| data_exp     | std_logic_vector     | x"0D"                              | The data value to expect when reading the addressed register. A mismatch results in an alert 'alert_level'   |
| byte_enable  | std_logic_vector     | x"11"                              | This argument selects which bytes to use (all '1' means all bytes are updated)   |
| alert_level  | t_alert_level        | ERROR or TB_WARNING                | Set the severity for the alert that may be asserted by the procedure.  |
| msg          | string               | "Set state active on peripheral 1" | A custom message to be appended in the log/alert.  |
| scope        | string               | "AXILITE BFM"                      | A string describing the scope from which the log/alert originates.<br>In a simple single sequencer typically "AXILITE BFM". In a verification component typically "AXILITE_VVC". |
| msg_id_panel | t_msg_id_panel       | shared_msg_id_panel                | Optional msg_id_panel, controlling verbosity within a specified scope. Defaults to a common message ID panel defined in the UVVM-Util adaptations package.                       |
| config       | t_axilite_bfm_config | C_AXILITE_BFM_CONFIG_DEFAULT       | Configuration of BFM behaviour and restrictions. See section 2 for details.  |

## BFM signal parameters

| Name       | Type         | Description   |
|------------|--------------|---|
| clk        | std_logic    | The clock signal used to read and write data in/out of the AXI4-Lite BFM. |
| axilite_if | t_axilite_if | See table "Signal record 'axilite_if'"                                    |

Note: All signals are active high. See AXI4-Lite documentation for protocol description.

For more information on the AXI4-Lite signals, please see the AXI4-Lite specification.

## Signal record 'axilite\_if'

| Record element                | Type                                    |
|-------------------------------|---|
| <b>write_address_channel</b>  | <b>t_axilite_write_address_channel</b>  |
| - awaddr                      | - std_logic_vector                      |
| - awvalid                     | - std_logic                             |
| - awprot                      | - std_logic_vector(2 downto 0)          |
| - awready                     | - std_logic                             |
| <b>write_data_channel</b>     | <b>t_axilite_write_data_channel</b>     |
| - wdata                       | - std_logic_vector                      |
| - wstrb                       | - std_logic_vector                      |
| - wvalid                      | - std_logic                             |
| - wready                      | - std_logic                             |
| <b>write_response_channel</b> | <b>t_axilite_write_response_channel</b> |
| - bready                      | - std_logic                             |
| - bresp                       | - std_logic_vector(1 downto 0)          |
| - bvalid                      | - std_logic                             |
| <b>read_address_channel</b>   | <b>t_axilite_read_address_channel</b>   |
| - araddr                      | - std_logic_vector                      |
| - arvalid                     | - std_logic                             |
| - arprot                      | - std_logic_vector(2 downto 0)          |
| - arready                     | - std_logic                             |
| <b>read_data_channel</b>      | <b>t_axilite_read_data_channel</b>      |
| - rready                      | - std_logic                             |
| - rdata                       | - std_logic_vector                      |
| - rresp                       | - std_logic_vector(1 downto 0)          |
| - rvalid                      | - std_logic                             |

# BFM details

## 1 BFM procedure details and examples

| Procedure              | Description   |
|------------------------|---|
| <b>axilite_write()</b> | <p><b>axilite_write(addr_value, data_value, [byte_enable,] msg, clk, axilite_if, [scope, [msg_id_panel, [config]]])</b></p> <p>The axilite_write() procedure writes the given data to the given address of the DUT, using the AXI4-Lite protocol. For protocol details, see the AXI4-Lite specification.</p> <ul style="list-style-type: none"> <li>- If the byte_enable argument is not used, it will be set to all '1', i.e. all bytes are used.</li> <li>- The default value of scope is C_SCOPE ("AXILITE BFM")</li> <li>- The default value of msg_id_panel is shared_msg_id_panel, defined in UVVM-Util.</li> <li>- The default value of config is C_AXILITE_BFM_CONFIG_DEFAULT, see table on the first page.</li> <li>- A log message is written if ID_BFM ID is enabled for the specified message ID panel.</li> </ul> <p>The procedure reports an alert if:</p> <ul style="list-style-type: none"> <li>- Data length is neither 32 bit nor 64 bit (alert level: TB_ERROR)</li> <li>- wready does not occur within max_wait_cycles clock cycles (alert level: max_wait_cycles_severity, set in the config)</li> <li>- awready does not occur within max_wait_cycles clock cycles (alert level: max_wait_cycles_severity, set in the config)</li> <li>- bresp is not set to expected_response (set in the config) when bvalid is set to '1' (alert level: expected_response_severity, set in the config)</li> <li>- bvalid is not set within max_wait_cycles clock cycles (alert level: max_wait_cycles_severity, set in the config)</li> </ul> <p>Examples:</p> <pre>axilite_write(x"00101155", x"AAAA", "Writing data to Peripheral 1", clk, axilite_if, C_SCOPE, shared_msg_id_panel,              C_AXILITE_BFM_CONFIG_DEFAULT); axilite_write(C_ADDR_PERIPHERAL_1, x"00F1", "01", "Writing first byte to Peripheral 1", clk, axilite_if, C_SCOPE,              shared_msg_id_panel, C_AXILITE_BFM_CONFIG_DEFAULT);</pre> <p>Suggested usage (requires local overload, see section 5):</p> <pre>axilite_write(C_ADDR_DMA, x"AAAA", "Writing data to DMA");</pre> |
| <b>axilite_read()</b>  | <p><b>axilite_read(addr_value, data_value, msg, clk, axilite_if, [scope, [msg_id_panel, [config, [proc_name]]]])</b></p> <p>The axilite_read() procedure reads data from the DUT at the given address, using the AXI4-Lite protocol. For protocol details, see the AXI4-Lite specification. The read data is placed on the output 'data_value' when the read has completed.</p> <ul style="list-style-type: none"> <li>- The default value of scope is C_SCOPE ("AXILITE BFM")</li> <li>- The default value of msg_id_panel is shared_msg_id_panel, defined in UVVM-Util.</li> <li>- The default value of config is C_AXILITE_BFM_CONFIG_DEFAULT, see table on the first page.</li> <li>- The default value of proc_name is "axilite_read". This argument is intended to be used internally, when the procedure is called by axilite_check().</li> <li>- A log message is written if ID_BFM ID is enabled for the specified message ID panel. This will only occur if the argument proc_name is left unchanged.</li> </ul> <p>The procedure reports an alert if:</p> <ul style="list-style-type: none"> <li>- The read data length (rdata) is neither 32 bit nor 64 bit (alert level: TB_ERROR)</li> <li>- arready does not occur within max_wait_cycles clock cycles (alert level: max_wait_cycles_severity, set in the config)</li> <li>- rresp is not set to expected_response (set in the config) when rvalid is set to '1' (alert level: expected_response_severity, set in the config)</li> </ul>   |

- rvalid is not set within max\_wait\_cycles clock cycles (alert level: max\_wait\_cycles\_severity, set in the config)

Example:

```
axilite_read(C_ADDR_PERIPHERAL_1, v_data_out, "Read from Peripheral 1", clk, axilite_if, C_SCOPE, shared_msg_id_panel,
            C_AXILITE_BFM_CONFIG_DEFAULT);
```

Suggested usage (requires local overload, see section 5):

```
axilite_read(C_ADDR_IO, v_data_out, "Reading from IO device");
```

---

## axilite\_check()

**axilite\_check(addr\_value, data\_exp, msg, clk, axilite\_if, [alert\_level, [scope, [msg\_id\_panel, [config]]]])**

The axilite\_check() procedure reads data from the DUT at the given address, using the AXI4-Lite protocol. For protocol details, see the AXI4-Lite specification. After reading data from the AXI4-Lite bus, the read data is compared with the expected data, 'data\_exp'.

- The default value of alert\_level is ERROR
- The default value of scope is C\_SCOPE ("AXILITE BFM")
- The default value of msg\_id\_panel is shared\_msg\_id\_panel, defined in UVVM-Util.
- The default value of config is C\_AXILITE\_BFM\_CONFIG\_DEFAULT, see table on the first page.
- If the check was successful, and the read data matches the expected data, a log message is written with ID\_BFM ID (if this ID has been enabled).
- If the read data did not match the expected data, an alert with severity 'alert\_level' will be reported.

The procedure also report alerts for the same conditions as the axilite\_read() procedure.

Example:

```
axilite_check(C_ADDR_PERIPHERAL_1, x"3B", "Check data from Peripheral 1", clk, axilite_if, C_SCOPE, shared_msg_id_panel, ERROR,
            C_AXILITE_BFM_CONFIG_DEFAULT); -- All parameters included
```

Suggested usage (requires local overload, see section 5):

```
axilite_check(C_ADDR_UART_RX, x"3B", "Checking data in UART RX register");
```

---

## init\_axilite\_if\_signals()

**init\_axilite\_if\_signals(addr\_width, data\_width)**

This function initializes the AXI4-Lite interface. All the BFM outputs are set to zeros ('0') and BFM inputs are set to 'Z'. awprot and arprot are set to UNPRIVILEGED\_UNSECURE\_DATA("010").

Example:

```
axilite_if <= init_axilite_if_signals(addr_width, data_width)
```

## 2 BFM Configuration record

Type name: t\_axilite\_bfm\_config

| Record element             | Type                      | C_AXILITE_BFM_CONFIG_DEFAULT | Description  |
|----------------------------|---------------------------|------------------------------|--|
| max_wait_cycles            | natural                   | 10                           | Used for setting the maximum cycles to wait before an alert is issued when waiting for ready and valid signals from the DUT.   |
| max_wait_cycles_severity   | t_alert_level             | TB_FAILURE                   | The above timeout will have this severity  |
| clock_period               | time                      | 10 ns                        | Period of the clock signal.  |
| clock_period_margin        | time                      | 0 ns                         | Input clock period margin to specified clock_period  |
| clock_margin_severity      | t_alert_level             | TB_ERROR                     | The above margin will have the severity  |
| setup_time                 | time                      | 2.5 ns                       | Setup time for generated signals. Suggested value is clock_period/4. An alert is reported if setup_time exceed clock_period/2. |
| hold_time                  | time                      | 2.5 ns                       | Hold time for generated signals. Suggested value is clock_period/4. An alert is reported if hold_time exceed clock_period/2.   |
| expected_response          | t_axilite_response_status | OKAY                         | Sets the expected response for both read and write transactions.   |
| expected_response_severity | t_alert_level             | TB_FAILURE                   | A response mismatch will have this severity.   |
| protection_setting         | t_axilite_protection      | UNPRIVILEGED_UNSECURE_DATA   | Sets the AXI access permissions (e.g. write to data/instruction, privileged and secure access).                                |
| num_aw_pipe_stages         | natural                   | 1                            | Write Address Channel pipeline steps   |
| num_w_pipe_stages          | natural                   | 1                            | Write Data Channel pipeline steps  |
| num_ar_pipe_stages         | natural                   | 1                            | Read Address Channel pipeline steps  |
| num_r_pipe_stages          | natural                   | 1                            | Read Data Channel pipeline steps   |
| num_b_pipe_stages          | natural                   | 1                            | Response Channel pipeline steps  |
| id_for_bfm                 | t_msg_id                  | ID_BFM                       | The message ID used as a general message ID in the AXI-Lite BFM  |
| id_for_bfm_wait            | t_msg_id                  | ID_BFM_WAIT                  | The message ID used for logging waits in the AXI-Lite BFM  |
| id_for_bfm_poll            | t_msg_id                  | ID_BFM_POLL                  | The message ID used for logging polling in the AXI-Lite BFM  |

## 3 Additional Documentation

For additional documentation on the AXI4-Lite standard, please see the AXI4-Lite specification “AMBA® AXI™ and ACE™ Protocol Specification - AXI3™, AXI4™, and AXI4-Lite™ ACE and ACE-Lite™”, available from ARM.

## 4 Compilation

The AXI4-Lite BFM may only be compiled with VHDL 2008. It is dependent on the UVVM Utility Library (UVVM-Util), which is only compatible with VHDL 2008. See the separate UVVM-Util documentation for more info. After UVVM-Util has been compiled, the axilite\_bfm\_pkg.vhd BFM can be compiled into any desired library. See the UVVM Essential Mechanisms located in uvvm\_vvc\_framework/doc for information about compile scripts.

### 4.1 Simulator compatibility and setup

See README.md for a list of supported simulators. For required simulator setup see UVVM-Util Quick reference.

## 5 Local BFM overloads

A good approach for better readability and maintainability is to make simple, local overloads for the BFM procedures in the TB process.

This allows calling the BFM procedures with the key parameters only

e.g.

```
axilite_write(C_ADDR_PERIPHERAL_1, C_TEST_DATA, "Sending data to Peripheral 1");
```

rather than

```
axilite_write(C_ADDR_PERIPHERAL_1, C_TEST_DATA, "Sending data to Peripheral 1", clk, axilite_if, C_SCOPE,
             shared_msg_id_panel, C_AXILITE_BFM_CONFIG_DEFAULT);
```

By defining the local overload as e.g.:

```
procedure axilite_write(
  constant addr_value  : in unsigned;
  constant data_value  : in std_logic_vector;
  constant msg         : in string) is
begin
  axilite_write(addr_value,
                data_value,
                msg,
                clk,
                axilite_if,
                C_SCOPE,
                shared_msg_id_panel,
                C_AXILITE_BFM_CONFIG_LOCAL);
end;
```

-- keep as is  
 -- keep as is  
 -- keep as is  
 -- Clock signal  
 -- Signal must be visible in local process scope  
 -- Just use the default  
 -- Use global, shared msg\_id\_panel  
 -- Use locally defined configuration or C\_AXILITE\_BFM\_CONFIG\_DEFAULT

Using a local overload like this also allows the following – if wanted:

- Have address value as natural – and convert in the overload
- Set up defaults for constants. May be different for two overloads of the same BFM
- Apply dedicated message\_id\_panel to allow dedicated verbosity control

### IMPORTANT

This is a simplified Bus Functional Model (BFM) for AXI4-Lite. The given BFM complies with the basic AXI4-Lite protocol and thus allows a normal access towards an AXI4-Lite interface. This BFM is not AXI4-Lite protocol checker. For a more advanced BFM please contact Bitvis AS at [support@bitvis.no](mailto:support@bitvis.no)

### INTELLECTUAL PROPERTY

Disclaimer: This IP and any part thereof are provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with this IP.