GMII BFM – Quick Reference

This is a stripped-down version of GMII with only data lines.

For general information see UVVM Essential Mechanisms located in uvvm_vvc_framework/doc.

BFM

gmii_bfm_pkg.vhd

init_gmii_to_dut_if (VOID)

Example: gmii_to_dut_if <= init_gmii_to_dut_if(VOID);

gmii_write (data, msg, gmii_to_dut_if, [scope, [msg_id_panel, [config]]])

Example: gmii_write(v_write_bytes, “Send 10 bytes of data”, gmii_to_dut_if);

gmii_read (data, msg, gmii_from_dut_if, [scope, [msg_id_panel, [config]]])

Example: gmii_read(v_receive_bytes, “Read 10 bytes of data”, gmii_from_dut_if);

BFM Configuration record ´**t_gmii_bfm_config**´

Record element	Type	C_GMII_BFM_CONFIG_DEFAULT
clock_period	time	-1 ns
setup_time	time	-1 ns
hold_time	time	-1 ns
timeout	time	1 us
timeout_severity	t_alert_level	TB_ERROR
id_for_bfm	t_msg_id	ID_BFM
id_for_bfm_wait	t_msg_id	ID_BFM_WAIT

Signal record ´**t_gmii_to_dut_if**´

Record element	Type
gtxclk	std_logic
txd	std_logic_vector(7 downto 0)
txen	std_logic

Signal record ´**t_dut_to_gmii_if**´

Record element	Type
rxclk	std_logic
rxd	std_logic_vector(7 downto 0)
rxdv	std_logic

UVVM™

VHDL 2008 only

BFM non-signal parameters

Name	Type	Example(s)	Description
data	t_byte_array	(x"1A", x"1B", x"2A")	The data value written to or read from the DUT.
msg	string	"Write to Peripheral 1"	A custom message to be appended in the log/alert.
scope	string	"GMII BFM"	A string describing the scope from which the log/alert originates. In a simple single sequencer typically "GMII BFM". In a verification component typically "GMII_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 ID panel defined in the adaptations package.
config	t_gmii_bfm_config	C_GMII_BFM_CONFIG_DEFAULT	Configuration of BFM behaviour and restrictions. See section 2 for details.

BFM signal parameters

Name	Type	Description
gtxclk	std_logic	Clock signal for TX signals.
txd	std_logic_vector	Data from DUT.
txen	std_logic	Transmit enable from DUT.
rxclk	std_logic	Clock signal for RX signals.
rxd	std_logic_vector	Data to DUT.
rxdv	std_logic	Data valid signal to DUT.

BFM details

1 BFM procedure details and examples

Procedure	Description
init_gmii_to_dut_if()	**init_gmii_to_dut_if (VOID)** This function initializes the gmii_to_dut_if interface. All output-signals are set to '0', input signals are set to 'Z'. Example: `gmii_to_dut_if <= init_gmii_to_dut_if(VOID);`
gmii_write()	**gmii_write(data, msg, gmii_to_dut_if, [scope, [msg_id_panel, [config]]])** The gmii_write() procedure writes the given data to the DUT, using the GMII protocol. Example: `gmii_write(v_write_data_bytes, "Write data to DUT", gmii_to_dut_if);`
gmii_read()	**gmii_read(data, msg, gmii_from_dut_if, [scope, [msg_id_panel, [config]]])** The gmii_read() procedure reads data from the DUT, using the GMII protocol. Example: `gmii_write(v_receive_data_bytes, "Read data from DUT", gmii_from_dut_if);`

2 BFM Configuration record

Type name: t_gmii_bfm_config

Record element	Type	C_GMII_BFM_CONFIG_DEFAULT	Description
clock_period	time	-1 ns	Period of the clock signal.
setup_time	time	-1 ns	Generated signals setup time. Suggested value is clock_period/4. An alert is reported if setup_time exceed clock_period/2.
hold_time	time	-1 ns	Generated signals hold time. Suggested value is clock_period/4. An alert is reported if hold_time exceed clock_period/2.
timeout	time	1 us	The maximum time allowed to wait for DUT.
timeout_severity	t_alert_level	TB_ERROR	Severity of alert when timeout.
id_for_bfm	t_msg_id	ID_BFM	The message ID used as a general message ID in the GMII BFM
id_for_bfm_wait	t_msg_id	ID_BFM_WAIT	The message ID used for logging waits in the GMII BFM

3 Compilation

The GMII 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 gmii_bfm_pkg.vhd can be compiled into any desired library.
See UVVM Essential Mechanisms located in uvvm_vvc_framework/doc for information about compile scripts.

3.1 Simulator compatibility and setup

This BFM has been compiled and tested with Modelsim version 10.5b and Riviera-PRO version 2018.02.111.6909

For required simulator setup see UVVM-Util Quick reference.

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.

```
gmii_write(v_write_data_bytes, "Write data to DUT");
```

rather than

```
gmii_write(v_write_data_bytes, "Write data to DUT", gmii_to_dut_if, C_SCOPE,
      shared_msg_id_panel, C_GMII_BFM_CONFIG_DEFAULT);
```

By defining the local overload as e.g.:

```
procedure gmii_write(
  constant data : in t_byte_array;
  constant msg  : in string) is
begin
  gmii_write(data,                          -- keep as is
             msg,                           -- keep as is
             gmii_to_dut_if,                -- Signal must be visible in local process scope
             C_SCOPE,                       -- Just use the default
             shared_msg_id_panel,           -- Use global, shared msg_id_panel
             C_GMII_BFM_CONFIG_LOCAL);      -- Use locally defined configuration or C_GMII_BFM_CONFIG_DEFAULT
end;
```

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

- 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 GMII.
The given BFM complies with the basic GMII protocol and thus allows a normal access towards a GMII interface. This BFM is not a GMII protocol checker.
For a more advanced BFM please contact Bitvis AS at support@bitvis.no