UART MONITOR – Quick Reference

MONITOR

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

UART Monitor Configuration record ´**t_uart_monitor_config**´ is accessible via shared variable array **shared_uart_monitor_config(channel, instance)**. The UART transaction information is located in **‘t_transaction_group’** accessible via global signal array **global_uart_monitor_transaction(channel, instance)**.

uart_monitor.vhd

t_uart_monitor_config

Record element	Type	C_UART_MONITOR_CONFIG_DEFAULT
scope_name	string	
msg_id_panel	t_msg_id_panel	C_UART_MONITOR_MSG_ID_PANEL_DEFAULT
interface_config	t_uart_interface_config	C_UART_MONITOR_INTERFACE_CONFIG_DEFAULT
transaction_display_time	time	0 ns

t_uart_interface_config

Record element	Type	Description
bit_time	time	The time used to transfer one bit.
num_data_bits	positive range 7 to 8	Number of data bits.
parity	t_parity	The parity used, PARITY_ODD or PARITY_EVEN.
num_stop_bits	t_stop_bits	Number of stop bits, STOP_BITS_ONE, STOP_BIT_ONE_AND_HALF or STOP_BITS_TWO.

t_transaction

Record element	Type	Description
operation	t_operation	Operation on UART line, TRANSMIT, RECEIVE or NO_OPERATION.
data	std_logic_vector	UART data.
vvc_meta[1]	t_vvc_meta	Only used by VVC.
transaction_status	t_transaction_status	Status of transaction, SUCCEEDED, FAILED, INACTIVE or IN_PROGRESS.
error_info	t_error_info	Error information when failed transaction.

t_error_info

Record element	Type	Description
parity_bit_error	boolean	True if parity error detected.
stop_bit_error	boolean	True if stop bit error detected.

Record hierarchy of **‘t_transaction_group’ –** accessible via global_uart_monitor_transaction

Record element	Type
→ bt	t_transaction
→ operation	t_operation
→ data	std_logic_vector
→ vvc_meta[1]	t_vvc_meta
→ transaction_status	t_transaction_status
→ error_info	t_error_info
→ parity_bit_error	boolean
→ stop_bit_error	boolean
→ ct	t_transaction

Message IDs for UART Monitor

Message ID	Description
ID_FRAME_INITIATE	Logs start of UART frame.
ID_MONITOR	Logs information about monitored transaction

[1] vvc_meta only applies for the VVC

UVVM™

VHDL 2008 only

Monitor entity signals

Name	Type	Direction	Description
uart_dut_rx	std_logic	Input	Input of DUTs UART RX signal.
uart_dut_tx	std_logic	Input	Input of DUTs UART TX signal.

Monitor entity generic constants

Name	Type	Default	Description
GC_INSTANCE_IDX	natural	1	Instance number to assign the monitor.
GC_MONITOR_CONFIG	t_uart_monitor_config	C_UART_MONITOR_CONFIG_DEFAULT	Configuration of the UART monitor, both channels get initiated with this configuration.

1 Use of the monitor Direct Transaction Transfer (DTT) signal

All transaction information from the UART Monitor is located in the DTT global signal **global_uart_monitor_transaction(channel, instance).**

Name	Type	Example(s)	Description
global_uart_monitor_transaction(channel, instance_idx)	t_uart_transaction_array	global_uart_monitor_transaction(RX, 1)	Global signal containing all UART Monitor transaction information.
channel	t_channel	TX, RX	The interface channel of the monitor instance
instance_idx	natural	1,2, etc.	Instance number of the monitor

An example of use of the global_uart_monitor_transaction signal is seen below. A process extracts the DTT data from the global signal.

```
p_monitor_tx : process
  variable v_transaction : t_uart_transaction;
begin
  wait until (global_uart_monitor_transaction(TX, 1).bt.transaction_status = SUCCEEDED or
              global_uart_monitor_transaction(TX, 1).bt.transaction_status = FAILED);
  v_transaction := global_uart_monitor_transaction(TX, 1).bt;

  -- Processing received transaction
  ...

end process p_monitor_tx;
```

2 Monitor Configuration

Record element	Type	C_UART_MONITOR_CONFIG_DEFAULT	Description
scope_name	string		A string describing the scope from which the log/alert originates.
msg_id_panel	t_msg_id_panel	C_UART_MONITOR_MSG_ID_PANEL_DEFAULT	The message id panel used by the monitor instance.
interface_config	t_uart_interface_config	C_UART_MONITOR_INTERFACE_CONFIG_DEFAULT	The configuration for the interface.
transaction_display_time	time	0 ns	After this amount of time operation is set to NO_OPERATION and transaction_status is set to INACTIVE if not a new transaction is received. If set to 0 ns operation and transaction_status will be unchanged until the next transfer is started.

The configuration record can be accessed from the Central Testbench Sequencer through the shared variable array, e.g.:

```
  shared_uart_monitor_config(TX, 1).msg_id_panel                     := new_msg_id_panel;
  shared_uart_monitor_config(TX, 1).interface_config.num_data_bits := 8;
```

3 Additional Documentation

Additional documentation about UVVM and its features can be found under “/uvvm_vvc_framework/doc/”.
For additional documentation on the UART protocol, please see the UART specification.

4 Compilation

The UART Monitor must be compiled with VHDL 2008.
It is dependent on the following libraries

- ***UVVM Utility Library (UVVM-Util), version 2.10.0 and up***
- ***UVVM VVC Framework, version 2.7.0 and up***
- ***UART BFM***

Before compiling the UART Monitor, make sure that uvvm_vvc_framework and uvvm_util have been compiled.

See UVVM Essential Mechanisms located in uvvm_vvc_framework/doc for information about compile scripts.

Compile order for the UART Monitor:

Compile to library	File	Comment
bitvis_vip_uart	transaction_pkg.vhd	UART transaction types
bitvis_vip_uart	uart_bfm_pkg.vhd	UART BFM
bitvis_vip_uart	vvc_cmd_pkg.vhd	UART VVC command types and operations
bitvis_vip_uart	monitor_cmd_pkg.vhd	UART Monitor command types and operations
bitvis_vip_uart	../uvvm_vvc_framework/src_target_dependent/td_target_support_pkg.vhd	UVVM VVC target support package, compiled into the UART VIP library.
bitvis_vip_uart	vvc_methods_pkg.vhd	UART VVC methods
bitvis_vip_uart	uart_monitor.vhd	UART Monitor

5 Simulator compatibility and setup

This Monitor has been compiled and tested with Modelsim version 10.5b and Riviera-PRO version 2015.10.85.
For required simulator setup see ***UVVM-Util*** Quick reference.

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.