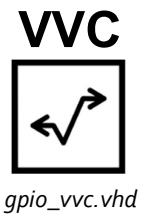
GPIO VVC – Quick Reference



gpio_vvc.vhd

gpio_set (VVCT, vvc_instance_idx, data, msg)

Example: gpio_set(GPIO_VVCT, 1, C_BAUDRATE_9600, “Set baudrate to 9600”);

gpio_get (VVCT, vvc_instance_idx, msg)

Example: gpio_get(GPIO_VVCT, 1, “Read baudrate”);

gpio_check (VVCT, vvc_instance_idx, data_exp, msg, [alert_level])

Example: gpio_check(GPIO_VVCT, 1, x”3B”, “Check data from UART RX”, ERROR);

gpi_expect (VVCT, vvc_instance_idx, data_exp, timeout, msg, [alert_level])

Example: gpio_expect(GPIO_VVCT, 1, x"0D", 2 ms, “Read UART RX until CR is found or timeout”, ERROR);

Common VVC procedures applicable for this VVC

- See UVVM Methods QuickRef for details.

Name
await_completion()
await_any_completion()
enable_log_msg()
disable_log_msg()
flush_command_queue()
terminate_current_command()
fetch_result()
insert_delay()
get_last_received_cmd_idx()
terminate_current_command()

GPIO VVC Configuration record ‘t_vvc_config’

- Accessible via **shared_gpio_vvc_config** – se section 2.

Record element
inter_bfm_delay
[cmd/result]_queue_count_max
[cmd/result]_queue_count_threshold
[cmd/result]_queue_count_threshold_severity
result_queue_count_max
result_queue_count_threshold_severity
result_queue_count_threshold
bfm_config
msg_id_panel

GPIO VVC Status record signal ‘t_vvc_status’

- Accessible via **shared_gpio_vvc_status** – se section 3.

Record element
current_cmd_idx
previous_cmd_idx
pending_cmd_idx

VHDL 2008 only

VVC target parameters

Name	Type	Example(s)	Description
VVCT	t_vvc_target_record	GPIO_VVCT	VVC target type compiled into each VVC in order to differentiate between VVCs.
vvc_instance_idx	integer	1	Instance number of the VVC

VVC functional parameters

Name	Type	Example(s)	Description
data	std_logic_vector	x"FF"	The data to be written (in gpio_set).
data_exp	std_logic_vector	x"FF"	The expected data to be read (in gpio_check or gpio_expect).
msg	string	"Set baudrate"	A custom message to be appended in the log/alert
alert_level	t_alert_level	ERROR or TB_WARNING	Set the severity for the alert that may be asserted by the method.

VVC entity signals

Name	Type	Direction	Description
gpio_vvc_if	t_gpio_if	Inout	See GPIO BFM documentation

VVC entity generic constants

Name	Type	Default	Description
GC_DATA_WIDTH	natural	32	Bits in the GPIO data word
GC_INSTANCE_IDX	natural	1	Instance number to assign the VVC
GC_DEFAULT_LINE_VALUE	boolean	TRUE	Default value of input or output GPIO.
GC_GPIO_BFM_CONFIG	t_gpio_bfm_config	C_GPIO_BFM_CONFIG_DEFAULT	Configuration for the GPIO BFM, see GPIO BFM documentation.
GC_CMD_QUEUE_COUNT_MAX	natural	1000	Absolute maximum number of commands in the VVC command queue
GC_CMD_QUEUE_COUNT_THRESHOLD	natural	950	An alert will be generated when reaching this threshold to indicate that the command queue is almost full. The queue will still accept new commands until it reaches C_CMD_QUEUE_COUNT_MAX.
GC_CMD_QUEUE_COUNT_THRESHOLD_SEVERITY	t_alert_level	WARNING	Alert severity which will be used when command queue reaches GC_CMD_QUEUE_COUNT_THRESHOLD.
GC_RESULT_QUEUE_COUNT_MAX	natural	1000	Maximum number of unfetched results before result_queue is full.
GC_RESULT_QUEUE_COUNT_THRESHOLD	natural	950	An alert with severity 'result_queue_count_threshold_severity' will be issued if command queue exceeds this count. Used for early warning if result queue is almost full. Will be ignored if set to 0.
GC_RESULT_QUEUE_COUNT_THRESHOLD_SEVERITY	t_alert_level	WARNING	Severity of alert to be initiated if exceeding result_queue_count_threshold

VVC details

All VVC procedures are defined in vvc_methods_pkg (dedicated this VVC), and uvvm_vvc_framework.uvvm_methods_pkg and uvvm_vvc_framework.uvvm_support_pkg (common VVC procedures). It is also possible to send a multicast to all instances of a VVC with ALL_INSTANCES as parameter for vvc_instance_idx.

1 VVC procedure details and examples

Procedure	Description
gpio_set()	**gpio_set (VVCT, vvc_instance_idx, data, msg)** The gpio_set() VVC procedure adds a SET command to the GPIO VVC executor queue, that will run as soon as all preceding commands have completed. When the SET command is scheduled to run, the executor calls the GPIO BFM gpio_set() procedure, described in the GPIO BFM QuickRef. Example: `gpio_set (GPIO_VVCT, 1, C_BAUDRATE_9600, "Set baudrate to 9600");`

gpio_get()	**gpio_get (VVCT, vvc_instance_idx, msg)**
	
	The gpio_get() VVC procedure adds a GET command to the GPIO VVC executor queue, that will run as soon as all preceding commands have completed. When the transmit command is scheduled to run, the executor calls the GPIO BFM gpio_get() procedure, described in the GPIO BFM QuickRef. The received data from DUT will not be returned in this procedure call since it is non-blocking for the sequencer/caller, but the received data will be stored in the VVC for a potential future fetch (see example with fetch_result below).
	
	Example:
	`gpio_get (GPI_VVCT, 1, "Read baudrate");`
	
	Example with fetch_result() call: Result is placed in **`v_data`**
	`variable v_cmd_idx : natural; -- Command index for the last read`
	`variable v_data : std_logic_vector(31 downto 0); -- Result from read`
	`(…)`
	`gpio_get(GPIO_VVCT, 1, "Read baudrate");`
	`v_cmd_idx := get_last_received_cmd_idx(GPIO_VVCT, 1);`
	`await_completion(GPIO_VVCT,1, v_cmd_idx, 1 us, "Wait for receive to finish");`
	`fetch_result(GPIO_VVCT,1, v_cmd_idx, v_data, "Fetching result from receive operation");`

gpio_check()	**gpio_check (VVCT, vvc_instance_idx, data_exp, msg, alert_level)**
	
	The gpio_check() VVC procedure adds a CHECK command to the GPIO VVC executor queue, that will run as soon as all preceding commands have completed. When the CHECK command is scheduled to run, the executor calls the GPIO BFM gpio_check() procedure, described in the GPIO BFM QuickRef.
	
	Example:
	`gpio_check (GPIO_VVCT, 1, x"F5", "Check data from UART RX");`
	The procedure can also be called with the optional parameters, e.g.:
	`gpio_check (GPIO_VVCT, 1, x"5F", "Check data from UART RX", ERROR);`

gpio_expect()

gpio_expect (VVCT, vvc_instance_idx, data_exp, timeout, msg, alert_level)

The gpio_expect() VVC procedure adds a EXPECT command to the GPIO VVC executor queue, that will run as soon as all preceding commands have completed. When the EXPECT command is scheduled to run, the executor calls the GPIO BFM gpio_expect() procedure, described in the GPIO BFM QuickRef.

Example:

```
gpio_expect(GPIO_VVCT, 1, x"0D", 2 ms, "Read UART RX until CR is found or timeout");
```

The procedure can also be called with the optional parameters, e.g.:

```
gpio_expect(GPIO_VVCT, 1, x"0D", 2 ms, "Read UART RX until CR is found or timeout", ERROR);
```

2 VVC Configuration

Record element	Type	C_SPI_VVC_CONFIG_DEFAULT	Description
inter_bfm_delay	t_inter_bfm_delay	C_GPIO_INTER_BFM_DELAY_DEFAULT	Delay between any requested BFM accesses towards the DUT. - TIME_START2START: Time from a BFM start to the next BFM start (A TB_WARNING will be issued if access takes longer than TIME_START2START). - TIME_FINISH2START: Time from a BFM end to the next BFM start. Any insert_delay() command will add to the above minimum delays, giving for instance the ability to skew the BFM starting time.
cmd_queue_count_max	natural	C_CMD_QUEUE_COUNT_MAX	Maximum pending number in command queue before queue is full. Adding additional commands will result in an ERROR.
cmd_queue_count_threshold	natural	C_CMD_QUEUE_COUNT_THRESHOLD	An alert with severity "cmd_queue_count_threshold_severity" will be issued if command queue exceeds this count. Used for early warning if command queue is almost full. Will be ignored if set to 0.
cmd_queue_count_threshold_severity	t_alert_level	C_CMD_QUEUE_COUNT_THRESHOLD_SEVERITY	Severity of alert to be triggered if command count exceeding cmd_queue_count_threshold
result_queue_count_max	natural	C_RESULT_QUEUE_COUNT_MAX	Maximum number of unfetched results before result_queue is full.
result_queue_count_threshold	natural	C_RESULT_QUEUE_COUNT_THRESHOLD	An alert with severity 'result_queue_count_threshold_severity' will be issued if command queue exceeds this count. Used for early warning if result queue is almost full. Will be ignored if set to 0.
result_queue_count_threshold_severity	t_alert_level	C_ RESULT_QUEUE_COUNT_THRESHOLD_SEVERITY	Severity of alert to be initiated if exceeding result_queue_count_threshold
bfm_config	t_gpio_bfm_config	C_GPIO_BFM_CONFIG_DEFAULT	Configuration for GPIO BFM. See QuickRef for GPIO BFM
msg_id_panel	t_msg_id_panel	C_VVC_MSG_ID_PANEL_DEFAULT	VVC dedicated message ID panel

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

```
shared_gpio_vvc_config(C_VVC_IDX).inter_bfm_delay.delay_in_time := 10 ms;
```

3 VVC Status

The current status of the VVC can be retrieved during simulation. This is done by reading from the shared variable shared_spi_vvc_status record from the test sequencer. The record contains status for both channels, specified with the channel axis of the shared_spi_vvc_status array.The record contents can be seen below:

Record element	Type	Description
current_cmd_idx	natural	Command index currently running
previous_cmd_idx	natural	Previous command index to run
pending_cmd_cnt	natural	Pending number of commands in the command queue

4 Additional Documentation

Additional documentation about UVVM and its features can be found under “/uvvm_vvc_framework/doc/”.

5 Compilation

The GPOI VVC must be compiled with VHDL 2008.
It is dependent on the following libraries

- ***UVVM Utility Library (UVVM-Util), version 2.2.0 and up***
- ***UVVM VVC Framework, version 2.1.0 and up***
- ***GPIO BFM***

Before compiling the GPIO VVC, make sure that uvvm_vvc_framework and uvvm_util have been compiled.

Compile order for the GPIO VVC:

Compile to library	File	Comment
bitvis_vip_gpio	gpio_bfm_pkg.vhd	GPIO BFM
bitvis_vip_gpio	vvc_cmd_pkg.vhd	GPIO VVC command types and operations
bitvis_vip_gpio	../uvvm_vvc_framework/src_target_dependent/td_target_support_pkg.vhd	UVVM VVC target support package, compiled into the GPIO VVC library.
bitvis_vip_gpio	../uvvm_vvc_framework/src_target_dependent/td_vvc_framework_common_methods_pkg.vhd	UVVM framework common methods compiled into the GPIO VVC library
bitvis_vip_gpio	vvc_methods_pkg.vhd	GPIO VVC methods
bitvis_vip_gpio	../uvvm_vvc_framework/src_target_dependent/td_queue_pkg.vhd	UVVM queue package for the VVC
bitvis_vip_gpio	../uvvm_vvc_framework/src_target_dependent/td_vvc_entity_support_pkg.vhd	UVVM VVC entity methods compiled into the GPIO VVC library
bitvis_vip_gpio	gpio_vvc.vhd	GPIO VVC

6 Simulator compatibility and setup

This VVC has been compiled and tested with Modelsim version 10.3d and Riviera-PRO version 2015.10.85.
For required simulator setup see ***UVVM-Util*** Quick reference.