

9.13.4.2 Supported unidirectional TLM interfaces

NOTE—Non-blocking TLM interface calls are zero-delay calls. The blocking interface calls, which correspond to *e* TCM methods, consume an additional tick and one cycle of simulation time.

Table 25—Supported TLM interfaces and related methods

TLM interface	Interface methods
Blocking unidirectional interfaces	
tlm_blocking_put of type	put(value:type)@sys.any
tlm_blocking_get of type	get(value:*type)@sys.any
tlm_blocking_peek of type	peek(value:*type)@sys.any
tlm_blocking_get_peek of type	get(value:*type)@sys.any peek(value:*type)@sys.any
Non-blocking unidirectional interfaces	
tlm_nonblocking_put of type	try_put(value:type) : bool can_put() : bool ok_to_put() : tlm_event
tlm_nonblocking_get of type	try_get(value:*type) : bool can_get() : bool ok_to_get() : tlm_event
tlm_nonblocking_peek of type	try_peek(value:*type) : bool can_peek() : bool ok_to_peek() : tlm_event
tlm_nonblocking_get_peek of type	try_get(value:*type) : bool can_get() : bool ok_to_get() : tlm_event try_peek(value:*type) : bool can_peek() : bool ok_to_peek() : tlm_event
Combined unidirectional interfaces (blocking and non-blocking)	
tlm_put of type	put(value:type)@sys.any try_put(value:type) : bool can_put() : bool ok_to_put() : tlm_event
tlm_get of type	get(value:*type)@sys.any try_get(value:*type) : bool can_get() : bool ok_to_get() : tlm_event
tlm_peek of type	peek(value:*type)@sys.any try_peek(value:*type) : bool can_peek() : bool ok_to_peek() : tlm_event

9.13.4.3 Supported bidirectional TLM interfaces

NOTE—Non-blocking TLM interface calls are zero-delay calls. The blocking interface calls, which correspond to *e* TCM methods, consume an additional tick and one cycle of simulation time.

Table 26—Supported bidirectional TLM interfaces and related methods

TLM interface	Interface methods
Blocking bidirectional interfaces	
tlm_blocking_master of (req-type, rsp-type)	put(value: req-type)@sys.any get(value: *rsp-type)@sys.any peek(value: *rsp-type)@sys.any
tlm_blocking_slave of (req-type, rsp-type)	put(value: rsp-type)@sys.any get(value: *req-type)@sys.any peek(value: *req-type)@sys.any
tlm_blocking_transport of (req-type, rsp-type)	transport(request: req-type, response: *rsp-type)@sys.any
Non-blocking bidirectional interfaces	
tlm_nonblocking_master of (req-type, rsp-type)	try_put(value: req-type) : bool can_put() : bool ok_to_put() : tlm_event try_get(value: *rsp-type): bool can_get(): bool ok_to_get(): tlm_event try_peek(value: *rsp-type): bool can_peek(): bool ok_to_peek(): tlm_event
tlm_nonblocking_slave of (req-type, rsp-type)	try_put(value: rsp-type) : bool can_put() : bool ok_to_put() : tlm_event try_get(value: *req-type): bool can_get(): bool ok_to_get(): tlm_event try_peek(value: *req-type): bool can_peek(): bool ok_to_peek(): tlm_event
tlm_nonblocking_transport of (req-type, rsp-type)	nb_transport(request: req-type, response: *rsp-type): bool
Combined bidirectional interfaces (blocking and non-blocking)	
tlm_master of (req-type, rsp-type)	put(value: req-type)@sys.any get(value: *rsp-type)@sys.any peek(value: *rsp-type)@sys.any try_put(value: req-type): bool can_put(): bool ok_to_put(): tlm_event try_get(value: *rsp-type): bool can_get(): bool ok_to_get(): tlm_event try_peek(value: *rsp-type): bool can_peek(): bool ok_to_peek(): tlm_event