



IEEE

IEC 61691-7

Edition 2.0 2025-06

INTERNATIONAL STANDARD

IEEE Std 1666™

Behavioural languages –
Part 7: SystemC® Language Reference Manual

iTeh Standards
(<https://standards.iteh.ai>)
Document Preview

[IEC 61691-7:2025](https://standards.iteh.ai/catalog/standards/iec/a948d02d-c6ee-4faf-be5b-f00ac4d88327/iec-61691-7-2025)

<https://standards.iteh.ai/catalog/standards/iec/a948d02d-c6ee-4faf-be5b-f00ac4d88327/iec-61691-7-2025>



THIS PUBLICATION IS COPYRIGHT PROTECTED
Copyright © 2023 IEEE

All rights reserved. IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Inc. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the IEC Central Office. Any questions about IEEE copyright should be addressed to the IEEE. Enquiries about obtaining additional rights to this publication and other information requests should be addressed to the IEC or your local IEC member National Committee.

IEC Secretariat
3, rue de Varembe
CH-1211 Geneva 20
Switzerland
Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue
New York, NY 10016-5997
United States of America
stds.ipr@ieee.org
www.ieee.org

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC/IEEE publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee, ...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch

Discover our powerful search engine and read freely all the publications previews, graphical symbols and the glossary. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 500 terminological entries in English and French, with equivalent terms in 25 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC 61691-7:2025

<https://standards.iteh.ai/catalog/standards/iec/a948d02d-c6ee-4faf-be5b-f00ac4d88327/iec-61691-7-2025>

Warning! Make sure that you obtained this publication from an authorized distributor.

Contents

1.	Overview.....	23
1.1	Scope.....	23
1.2	Purpose.....	23
1.3	Word usage	23
1.4	Subsets	24
1.5	Relationship with C++ standard	24
1.6	Guidance for readers	24
2.	Normative references	26
3.	Terminology and conventions used in this standard.....	27
3.1	Terminology.....	27
3.1.1	Shall, should, may, can	27
3.1.2	Implementation, application	27
3.1.3	Call, called from, derived from.....	27
3.1.4	Specific technical terms	27
3.2	Syntactical conventions	29
3.2.1	Implementation-defined.....	29
3.2.2	Disabled	29
3.2.3	Ellipsis (...).....	29
3.2.4	Class names.....	29
3.2.5	Embolded text.....	30
3.3	Semantic conventions	30
3.3.1	Class definitions and the inheritance hierarchy	30
3.3.2	Function definitions and side-effects	30
3.3.3	Functions that use const char* as parameter.....	30
3.3.4	Functions whose return type is a reference or a pointer	31
3.3.5	Namespaces and internal naming	33
3.3.6	Non-compliant applications and errors.....	33
3.4	Notes and examples	33
4.	Elaboration and simulation semantics	34
4.1	Overview.....	34
4.2	Elaboration.....	34
4.2.1	Overview.....	34
4.2.2	Instantiation	35
4.2.3	Process macros.....	36
4.2.4	Port binding and export binding	36
4.2.5	Setting the time resolution	37
4.3	Simulation.....	38
4.3.1	Overview.....	38
4.3.2	The scheduling algorithm	38
4.3.3	Initialization, cycles, pauses, and suspension in the scheduling algorithm	41
4.4	Running elaboration and simulation	42
4.4.1	Overview.....	42
4.4.2	Function declarations	42
4.4.3	sc_elab_and_sim.....	43
4.4.4	sc_argc and sc_argv	43

4.4.5	Running under application control using functions <code>sc_main</code> and <code>sc_start</code>	44
4.4.6	Running under control of the kernel	46
4.5	Elaboration and simulation callbacks	46
4.5.1	Overview	46
4.5.2	<code>before_end_of_elaboration</code>	47
4.5.3	<code>end_of_elaboration</code>	48
4.5.4	<code>start_of_simulation</code>	49
4.5.5	<code>end_of_simulation</code>	50
4.6	Other functions related to the scheduler	50
4.6.1	Function declarations	50
4.6.2	<code>sc_pause</code>	52
4.6.3	<code>sc_suspend_all</code> , <code>sc_unsuspend_all</code> , <code>sc_unsuspendable</code> , and <code>sc_suspendable</code>	53
4.6.4	<code>sc_stop</code> , <code>sc_set_stop_mode</code> , and <code>sc_get_stop_mode</code>	53
4.6.5	<code>sc_time_stamp</code>	54
4.6.6	<code>sc_delta_count</code> and <code>sc_delta_count_at_current_time</code>	55
4.6.7	<code>sc_is_running</code>	55
4.6.8	Functions to detect pending activity	56
4.6.9	<code>sc_get_status</code>	57
4.6.10	<code>sc_stage_callback_if</code>	58
4.6.11	<code>sc_register_stage_callback</code>	60
4.6.12	<code>sc_unregister_stage_callback</code>	60
4.6.13	Callbacks execution	60
5.	Core language class definitions	61
5.1	Class header files	61
5.1.1	Overview	61
5.1.2	<code>#include "systemc"</code>	61
5.1.3	<code>#include "systemc.h"</code>	61
5.2	<code>sc_module</code>	62
5.2.1	Description	62
5.2.2	Class definition	62
5.2.3	Constraints on usage	65
5.2.4	<code>kind</code>	65
5.2.5	<code>SC_MODULE</code>	65
5.2.6	Constructors	66
5.2.7	<code>SC_CTOR</code>	66
5.2.8	<code>SC_METHOD</code> , <code>SC_THREAD</code> , <code>SC_CTHREAD</code>	67
5.2.9	<code>SC_NAMED</code>	68
5.2.10	Method process	68
5.2.11	Thread and clocked thread processes	69
5.2.12	Clocked thread processes	70
5.2.13	<code>reset_signal_is</code> and <code>async_reset_signal_is</code>	72
5.2.14	<code>sensitive</code>	73
5.2.15	<code>dont_initialize</code>	74
5.2.16	<code>set_stack_size</code>	75
5.2.17	<code>next_trigger</code>	75
5.2.18	<code>wait</code>	77
5.2.19	Positional port binding	78
5.2.20	<code>before_end_of_elaboration</code> , <code>end_of_elaboration</code> , <code>start_of_simulation</code> , <code>end_of_simulation</code>	80
5.2.21	<code>get_child_objects</code> and <code>get_child_events</code>	80
5.2.22	<code>sc_gen_unique_name</code>	81
5.2.23	<code>sc_behavior</code> and <code>sc_channel</code>	81

5.3	<i>sc_module_name</i>	82
5.3.1	Description	82
5.3.2	Class definition	82
5.3.3	Constraints on usage	83
5.3.4	Module hierarchy	83
5.3.5	Member functions	84
5.4	<i>sc_sensitive†</i>	85
5.4.1	Description	85
5.4.2	Class definition	85
5.4.3	Constraints on usage	85
5.4.4	operator<<	85
5.5	<i>sc_spawn_options</i> and <i>sc_spawn</i>	86
5.5.1	Description	86
5.5.2	Class definition	86
5.5.3	Constraints on usage	88
5.5.4	Constructors	88
5.5.5	Member functions	88
5.5.6	<i>sc_spawn</i>	89
5.5.7	SC_FORK and SC_JOIN	92
5.6	<i>sc_process_handle</i>	93
5.6.1	Description	93
5.6.2	Class definition	93
5.6.3	Constraints on usage	95
5.6.4	Constructors	95
5.6.5	Member functions	95
5.6.6	Member functions for process control	98
5.6.7	<i>sc_get_current_process_handle</i>	114
5.6.8	<i>sc_is_unwinding</i>	115
5.7	<i>sc_event_finder</i> and <i>sc_event_finder_t</i>	116
5.7.1	Description	116
5.7.2	Class definition	116
5.7.3	Constraints on usage	116
5.8	<i>sc_event_and_list</i> and <i>sc_event_or_list</i>	118
5.8.1	Description	118
5.8.2	Class definition	118
5.8.3	Constraints and usage	119
5.8.4	Constructors, destructor, assignment	119
5.8.5	Member functions and operators	119
5.9	<i>sc_event_and_expr†</i> and <i>sc_event_or_expr†</i>	121
5.9.1	Description	121
5.9.2	Class definition	121
5.9.3	Constraints on usage	122
5.9.4	Operators	122
5.10	<i>sc_event</i>	123
5.10.1	Description	123
5.10.2	Class definition	123
5.10.3	Constraints on usage	124
5.10.4	Constructors, destructor, and event naming	124
5.10.5	Functions for naming and hierarchy traversal	125
5.10.6	notify and cancel	126
5.10.7	Event lists	127
5.10.8	None event	127
5.10.9	Multiple event notifications	127
5.11	<i>sc_time</i>	128

5.11.1	Description.....	128
5.11.2	Class definition	128
5.11.3	Constructors	129
5.11.4	Functions and operators	130
5.11.5	Time resolution	130
5.11.6	sc_max_time	131
5.11.7	SC_ZERO_TIME	131
5.12	sc_port.....	131
5.12.1	Description.....	131
5.12.2	Class definition	131
5.12.3	Template parameters.....	133
5.12.4	Constraints on usage	134
5.12.5	Constructors	135
5.12.6	kind	135
5.12.7	Named port binding	135
5.12.8	Member functions for bound ports and port-to-port binding.....	136
5.12.9	before_end_of_elaboration, end_of_elaboration, start_of_simulation, end_of_simulation	140
5.13	sc_export.....	140
5.13.1	Description.....	140
5.13.2	Class definition	140
5.13.3	Template parameters.....	141
5.13.4	Constraints on usage	141
5.13.5	Constructors	142
5.13.6	kind	142
5.13.7	Export binding	142
5.13.8	Member functions for bound exports and export-to-export binding	143
5.13.9	before_end_of_elaboration, end_of_elaboration, start_of_simulation, end_of_simulation	144
5.14	sc_interface	144
5.14.1	Description.....	144
5.14.2	Class definition	145
5.14.3	Constraints on usage	145
5.14.4	register_port	145
5.14.5	default_event.....	146
5.15	sc_prim_channel	147
5.15.1	Description.....	147
5.15.2	Class definition	147
5.15.3	Constraints on usage	148
5.15.4	Constructors, destructor, and hierarchical names	148
5.15.5	kind	149
5.15.6	request_update and update.....	149
5.15.7	async_attach_suspending and async_detach_suspending	150
5.15.8	next_trigger and wait	150
5.15.9	before_end_of_elaboration, end_of_elaboration, start_of_simulation, end_of_simulation	151
5.16	sc_object	152
5.16.1	Description.....	152
5.16.2	Class definition	152
5.16.3	Constraints on usage	153
5.16.4	Constructors and destructor	153
5.16.5	name, basename, and kind	154
5.16.6	print and dump	155
5.16.7	Functions for object hierarchy traversal	155

5.16.8	Member functions for attributes	157
5.16.9	get_hierarchy_scope	158
5.17	Hierarchical naming of objects and events	158
5.17.1	Overview	158
5.17.2	sc_hierarchical_name_exists	159
5.17.3	sc_register_hierarchical_name, sc_unregister_hierarchical_name	160
5.18	sc_attr_base	160
5.18.1	Description	160
5.18.2	Class definition	160
5.18.3	Member functions	161
5.19	sc_attribute	161
5.19.1	Description	161
5.19.2	Class definition	161
5.19.3	Template parameters	161
5.19.4	Member functions and data members	161
5.20	sc_attr_cltn	162
5.20.1	Description	162
5.20.2	Class definition	162
5.20.3	Constraints on usage	162
5.20.4	Iterators	162
5.21	sc_hierarchy_scope	163
5.21.1	Description	163
5.21.2	Class definition	163
5.21.3	Constraints on usage	163
5.21.4	Constructor	164
5.21.5	get_root	164
6.	Predefined channel class definitions	165
6.1	sc_signal_in_if	165
6.1.1	Description	165
6.1.2	Class definition	165
6.1.3	Member functions	165
6.2	sc_signal_in_if<bool> and sc_signal_in_if<sc_dt::sc_logic>	166
6.2.1	Description	166
6.2.2	Class definition	166
6.2.3	Member functions	167
6.3	sc_signal_inout_if	167
6.3.1	Description	167
6.3.2	Class definition	167
6.3.3	Member functions	168
6.4	sc_signal	168
6.4.1	Description	168
6.4.2	Class definition	169
6.4.3	Template parameter T	169
6.4.4	Reading and writing signals	170
6.4.5	Constructors	171
6.4.6	register_port	171
6.4.7	Member functions for reading	172
6.4.8	Member functions for writing	172
6.4.9	Member functions for events	172
6.4.10	Diagnostic member functions	173
6.4.11	operator<<	173
6.5	sc_signal<bool,WRITER_POLICY> and sc_signal<sc_dt::sc_logic,WRITER_POLICY> ..	174