

**Specifikacijski jeziki elektronskega sistema – Ekspresni informacijski model
VHDL 87**

Electronic system specification language - Express information model VHDL 87

**iTeh STANDARD PREVIEW
(standards.iteh.ai)**

SIST-TP CLC/R217-011:2004
[https://standards.iteh.ai/catalog/standards/sist/5640b91f-feef-4fa7-9c73-
eb346a48f3a3/sist-tp-clc-r217-011-2004](https://standards.iteh.ai/catalog/standards/sist/5640b91f-feef-4fa7-9c73-eb346a48f3a3/sist-tp-clc-r217-011-2004)

iTeh STANDARD PREVIEW **(standards.iteh.ai)**

[SIST-TP CLC/R217-011:2004](https://standards.iteh.ai/catalog/standards/sist/5640b91f-feef-4fa7-9c73-eb346a48f3a3/sist-tp-clc-r217-011-2004)

[https://standards.iteh.ai/catalog/standards/sist/5640b91f-feef-4fa7-9c73-
eb346a48f3a3/sist-tp-clc-r217-011-2004](https://standards.iteh.ai/catalog/standards/sist/5640b91f-feef-4fa7-9c73-eb346a48f3a3/sist-tp-clc-r217-011-2004)

English version

**Electronic system specification languages
Express information model VHDL 87**

This CENELEC Report has been prepared by the Technical Committee CENELEC TC 217, Electronic Design Automation (EDA). It was approved by TC 217 on 1996-01-30 and endorsed by the CENELEC Technical Board on 1996-07-02.

iTeh STANDARD PREVIEW

CENELEC members are the national electrotechnical committees of Austria, Belgium, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.

[SIST-TP CLC/R217-011:2004](https://standards.iteh.ai/catalog/standards/sist/5640b91f-feef-4fa7-9c73-eb346a483a3/sist-tp-clc-r217-011-2004)

<https://standards.iteh.ai/catalog/standards/sist/5640b91f-feef-4fa7-9c73-eb346a483a3/sist-tp-clc-r217-011-2004>

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Central Secretariat: rue de Stassart 35, B - 1050 Brussels

Foreword

This report, which contains an information model written in EXPRESS language¹⁾, was developed by Christian A. Giumale²⁾ with support from both the ESIP project and the UK Defence Research Agency.

The model has been reviewed by a wide body of people from the VHDL community and presentations on this work have been given at VIUF in Washington and at an ESIP/CENELEC meeting hosted by Thomson in Paris.

The text of the report was approved by TC 217 on 1996-01-30 and approved for publication by the CENELEC Technical Board on 1996-07-02.

Background

The problems discussed in the present document resulted from work supported by the UK Defence Research Agency to produce a comprehensive information model of VHDL'87. As the research progressed it became clear that a single model of VHDL is not a satisfactory solution. Instead, a hierarchy of models, each of which describing the relevant aspects relating to a specific application purpose and from a specific perspective of VHDL, is a more satisfactory alternative. However, the model at the apex of this hierarchy describes the essential objects and the semantics of VHDL at the description, analysis, elaboration and simulation levels of design. It can be seen as an abstraction of all the other models in the hierarchy. Such a model, called here core model, and which is part of the present document, is a result of the above mentioned project.

Acknowledgements

Support for the work described has been received from the UK Defence Research Agency and the Commission of the European Communities through the ESIP (ESPRIT 8370) project. We wish to thank in particular Andy Carpenter and Alan R. Williams (Manchester University UK), Jacques Rouillard (ESIM France), Serafin Olcoz and Juana Lopez (TGI Spain) for their useful comments and suggestions concerning modelling, VHDL modelling, and VHDL itself.

1) EXPRESS Language reference manual ISO (DIS) 10303-11:1992, BSI.

2) University of Manchester, Oxford Road, Manchester M13 9PL UK

Contents

1	Information modelling of VHDL	13
1.1	Models of VHDL	13
1.2	Information modelling	14
1.3	Individual object modelling	16
1.4	Cross-level modelling	17
1.5	Behavioural modelling	19
1.5.1	General function modelling schemata	19
1.5.2	Modelling VHDL behavioural semantics	20
1.6	VHDL modelling space	23
1.7	VHDL core models	24
1.8	Model hierarchies	26
1.9	Proving properties of VHDL	27
1.10	Model based applications	30
1.11	The purpose and benefits of a VHDL information model	31
1.12	Conclusions	32
2	General modelling problems	33
2.1	Isomorphic hierarchies of objects	33
2.2	Object definition versus object reference	36
2.3	The cardinality of relationships of isomorphic objects	37
3	Model Overview	39
3.1	Model organisation	39
3.2	Abstract_block_schema	41
3.3	Time_schema	42
3.4	Basic_object_schema	43
3.4.1	Contents Overview	43
3.4.2	Modelling Problems	47
3.5	Association_schema	50
3.5.1	Contents Overview	50
3.5.2	Modelling Problems	51
3.6	Design_description_schema	55
3.6.1	Contents Overview	55
3.6.2	Modelling Problems	56
3.7	Design_configuration_schema	60
3.7.1	Contents Overview	60
3.7.2	Modelling Problems	60
3.8	Design_library_schema	63
3.8.1	Contents Overview	63
3.9	Design_elaboration_schema	65
3.9.1	Contents Overview	65
3.9.2	Modelling Problems	66
3.10	Design_simulation_schema	69

3.10.1 Contents Overview	69
3.10.2 Modelling Problems	73
3.11 Type_schema	79
4 The VHDL model	83
SCHEMA express_function_schema	85
FUNCTION is_type	85
FUNCTION select_objects	85
FUNCTION unique_instance	85
FUNCTION is_used_as	85
FUNCTION context_of	85
FUNCTION one_to_one	86
FUNCTION partial_one_to_one	87
FUNCTION not_defined	87
SCHEMA abstract_block_schema	89
ENTITY abstract_object	89
ENTITY abstract_block	89
ENTITY abstract_root_block	89
ENTITY abstract_nested_block	89
FUNCTION no_cycles	90
FUNCTION is_visible	90
SCHEMA type_schema	91
CONSTANT	91
ENTITY VHDL_identifier	91
ENTITY extended_VHDL_type	92
ENTITY VHDL_type	92
ENTITY VHDL_subtype	92
ENTITY extended_VHDL_value	92
ENTITY VHDL_value	93
ENTITY user_type	93
ENTITY null_value_type	93
RULE unique_null_value_type	94
ENTITY null_value	94
RULE unique_null_value	94
FUNCTION is_null_value	94
ENTITY scalar_type	94
ENTITY scalar_value	94
ENTITY numeric_type	94
ENTITY discrete_type	95
ENTITY numeric_value	95
ENTITY integer_type	95
RULE unique_integer_type	95
ENTITY integer_value	95
ENTITY floating_point_type	96
ENTITY floating_point_value	96
RULE unique_floating_point_type	96
ENTITY physical_type	97
ENTITY physical_unit	97
ENTITY physical_value	98
ENTITY enumeration_type	98
TYPE enumeration_literal	99
ENTITY graphic_character	99
ENTITY enumeration_value	99
FUNCTION index_of_value	100

iteh STANDARD PREVIEW
(standards.iteh.ai)

SIST-TP CLC/R217-011:2004

standards.iteh.ai/catalog/standards/sist/5640b91f-feef-4fa7-9c73-48fba3/sist-tp-clc-r217-011-2004

ENTITY range_type	100
ENTITY discrete_range	100
ENTITY integer_range	100
ENTITY enumeration_range	101
ENTITY floating_point_range	101
ENTITY physical_range	101
ENTITY composite_type	101
ENTITY composite_value	102
ENTITY record_type	102
ENTITY field_descriptor	102
ENTITY record_value	102
ENTITY field_value	103
FUNCTION record_field_values	103
ENTITY array_type	103
ENTITY constrained_array_type	104
ENTITY unconstrained_array_type	104
ENTITY index_specification	104
ENTITY unconstrained_index_specification	104
ENTITY constrained_index_specification	104
ENTITY array_value	105
FUNCTION array_item_values	105
ENTITY array_item_value	105
FUNCTION indexes_in_range	106
ENTITY access_type	107
ENTITY access_value	107
ENTITY file_type	107
FUNCTION is_one_dimensional_array	108
FUNCTION access_or_file_type_in_VHDL_type	108
FUNCTION access_or_file_type_in_array_type	108
FUNCTION access_or_file_type_in_record_type	108
ENTITY file_value	109
ENTITY VHDL_time_type	109
ENTITY VHDL_time_value	109
ENTITY natural_type	109
ENTITY positive_type	110
ENTITY boolean_type	110
RULE unique_boolean_type	110
ENTITY bit_type	110
RULE unique_bit_type	110
ENTITY character_type	110
RULE unique_character_type	110
ENTITY string_type	110
RULE unique_string_type	110
FUNCTION is_basic_type	111
FUNCTION is_user_type	111
FUNCTION is_same_type	111
RULE complete_identity_functions	111
SCHEMA basic_object_schema	113
ENTITY extended_typed_object	113
ENTITY typed_object	113
ENTITY basic_object	113
ENTITY initialised_object	113
FUNCTION valid_default_value	114
ENTITY VHDL_expression	114

STANDARD PREVIEW
(standards.iteh.ai)

SIST-TP CLC/R217-011:2004

<https://standards.iteh.ai/catalog/standards/sist/5640b91f-fee7-4fa7-9c73-b146e43f5196/sist-r217-011-2004>

ENTITY time_expression	114
ENTITY VHDL_function	115
ENTITY function_point	115
ENTITY conversion_function	116
ENTITY defined_conversion_function	116
ENTITY identity_conversion_function	116
FUNCTION make_identity_points	117
FUNCTION is_identity_conversion_function	117
FUNCTION get_identity_conversion_function	117
ENTITY composite_conversion_function	117
FUNCTION composable_functions	118
FUNCTION composite_function_points	118
RULE complete_composite_conversion_functions	119
FUNCTION get_composite_conversion_function	119
ENTITY resolution_function	119
ENTITY component_declaration	120
FUNCTION check_component_port	121
ENTITY generic_constant	121
TYPE bus_or_register	121
ENTITY signal	121
ENTITY guard_signal	122
ENTITY explicit_signal	122
FUNCTION check_resolution_function_type	124
TYPE port_mode	124
ENTITY port	124
FUNCTION visible_port_resolution_function	125
ENTITY internal_signal	125
FUNCTION visible_signal_resolution_function	126
ENTITY scalar_signal	126
ENTITY composite_signal	126
FUNCTION no_recursive_signal	127
FUNCTION valid_signal_elements	127
FUNCTION no_of_array_elements	128
ENTITY complete_signal	128
ENTITY signal_slice	128
FUNCTION check_slice_validity	129
FUNCTION slice_of_resolved_signal	129
FUNCTION is_signal	130
FUNCTION is_port	130
FUNCTION is_port_mode	130
FUNCTION is_readable	130
FUNCTION is_writable	130
ENTITY signal_source	131
ENTITY composite_signal_source	131
FUNCTION compatible_driver_sources	132
FUNCTION compatible_connection_sources	132
ENTITY scalar_signal_source	133
ENTITY signal_driver_source	133
ENTITY port_connection_source	134
ENTITY port_assoc_formal_to_actual	134
ENTITY port_assoc_actual_to_formal	135
RULE complete_port_connection_sources	135
FUNCTION composed_port_connections	135
FUNCTION get_port_connection_source	136

STANDARD PREVIEW
(standards.iteh.ai)

SCHEMA association_schema	137
TYPE assoc_value	137
ENTITY assoc_element	137
ENTITY port_assoc_element	137
ENTITY derived_port_assoc_element	139
ENTITY direct_port_assoc_element	139
ENTITY composite_port_assoc_element	140
FUNCTION compose_port_connections	142
ENTITY defined_port_assoc_element	143
ENTITY one_way_port_assoc	144
ENTITY two_way_port_assoc	144
FUNCTION check_conversion_type	144
FUNCTION port_connections_from_port_assocs	145
ENTITY generic_assoc_element	145
ENTITY derived_generic_assoc_element	146
ENTITY direct_generic_assoc_element	146
ENTITY composite_generic_assoc_element	147
ENTITY defined_generic_assoc_element	147
ENTITY map_aspect	148
ENTITY port_map_aspect	148
ENTITY defined_port_map_aspect	148
FUNCTION check_port_assoc_element	149
ENTITY derived_port_map_aspect	150
ENTITY generic_map_aspect	151
ENTITY defined_generic_map_aspect	151
ENTITY derived_generic_map_aspect	152
FUNCTION actual_for_formal	153
FUNCTION is_associated_formal	153
FUNCTION derive_port_assocs	153
FUNCTION derive_generic_assocs	154
FUNCTION check_unconnected_port	154
SCHEMA design_description_schema	155
ENTITY package_declaration	155
ENTITY package_body	157
ENTITY function_body	158
FUNCTION no_recursive_package_use	158
FUNCTION test_for_defined_types	159
FUNCTION test_for_function_defined_types	159
FUNCTION types_from_packages	159
FUNCTION functions_from_packages	160
FUNCTION signals_from_packages	160
FUNCTION components_from_packages	160
ENTITY entity_declaration	160
ENTITY program_block	162
ENTITY architecture_body	164
ENTITY block	165
FUNCTION check_guard_expression	167
FUNCTION test_for_visible_types	168
FUNCTION test_for_function_visible_types	168
FUNCTION convert_set	168
ENTITY concurrent_statement	169
ENTITY non_passive_concurrent_statement	169
ENTITY passive_concurrent_statement	169
ENTITY component_instance	169

ENTITY process	171
ENTITY passive_process	171
ENTITY non_passive_process	172
ENTITY sequential_statement	173
ENTITY computation_statement	173
TYPE assignment_delay	173
ENTITY signal_assignment	173
ENTITY waveform_element	174
ENTITY waveform_expression	174
ENTITY null_waveform_element	174
ENTITY non_null_waveform_element	174
ENTITY unguarded_signal_assignment	175
ENTITY guarded_signal_assignment	175
FUNCTION valid_guarded_assignment	175
ENTITY wait_statement	176
ENTITY delay_wait_clause	176
ENTITY signal_wait_clause	176
ENTITY condition_wait_clause	176
FUNCTION is_visible_signal	177
SCHEMA design_configuration_schema	179
ENTITY aspect	179
ENTITY design_entity	179
FUNCTION fully_bound_component_instance	180
FUNCTION check_design_entity	180
FUNCTION get_entity	180
FUNCTION get_architecture	181
ENTITY component_instance_binding	181
ENTITY configuration_specification	182
FUNCTION entity_visibility_from_config_spec	183
ENTITY component_configuration	183
FUNCTION entity_visibility_from_component_config	184
FUNCTION check_component_block_configuration	185
ENTITY configuration_hierarchy_block	185
ENTITY configuration_declaration	186
ENTITY block_configuration	187
ENTITY architecture_body_configuration	189
ENTITY internal_block_configuration	189
FUNCTION derive_visible_used_libraries	189
FUNCTION derive_visible_used_packages	189
FUNCTION blocks_from_component_configs	190
SCHEMA design_library_schema	191
ENTITY design_unit	191
ENTITY context_clause	191
ENTITY library_unit	191
FUNCTION derive_context_resource_libraries	192
FUNCTION derive_context_used_packages	192
ENTITY library_primary_unit	192
ENTITY library_secondary_unit	193
ENTITY design_library	193
SCHEMA design_elaboration_schema	195
ENTITY elaborated_design	195
ENTITY elaborated_design_entity	195
ENTITY elaborated_configuration_declaration	196
ENTITY elaborated_block_configuration	196

ENTITY elaborated_architecture_body_configuration	198
ENTITY elaborated_internal_block_configuration	198
ENTITY elaborated_VHDL_expression	198
ENTITY elaborated_time_expression	198
ENTITY evaluated_VHDL_expression	199
ENTITY evaluated_time_expression	199
FUNCTION valid_evaluated_expression	199
ENTITY elaborated_VHDL_function	200
ENTITY elaborated_resolution_function	200
ENTITY elaborated_conversion_function	200
ENTITY elaborated_package_declaration	200
ENTITY elaborated_program_block	201
ENTITY elaborated_architecture_body	203
ENTITY elaborated_block	204
FUNCTION valid_elaborated_guard	204
FUNCTION check_elaborated_generic	205
FUNCTION packages_from_elaborated_processes	205
FUNCTION signals_from_elaborated_packages	205
FUNCTION visible_signal	206
FUNCTION visible_signals	206
FUNCTION look_up_block_signal	206
FUNCTION select_signal	207
ENTITY elaborated_component_instance	207
ENTITY elaborated_component_configuration	209
ENTITY elaborated_configuration_specification	211
ENTITY elaborated_generic_constant	212
FUNCTION derive_generics_value	213
ENTITY elaborated_signal	213
ENTITY elaborated_guard_signal	213
ENTITY elaborated_explicit_signal	214
FUNCTION signal_default_value	215
FUNCTION check_elaborated_signal_resolution_function	215
FUNCTION check_disconnection_expression	215
ENTITY elaborated_scalar_signal	216
ENTITY elaborated_composite_signal	216
FUNCTION check_composite_default_value	216
ENTITY elaborated_complete_signal	217
ENTITY elaborated_signal_slice	217
ENTITY elaborated_signal_source	217
ENTITY elaborated_composite_signal_source	217
ENTITY elaborated_scalar_signal_source	218
ENTITY elaborated_signal_driver_source	218
ENTITY elaborated_port_connection_source	219
FUNCTION check_the_elaborated_actual	220
FUNCTION select_elaborated_source	220
FUNCTION check_elaborated_port_connection_conversion_function	220
FUNCTION check_port_to_port_connection	221
ENTITY block_port_connection	222
ENTITY component_instance_port_connection	222
ENTITY configuration_specification_port_connection	223
ENTITY component_configuration_port_connection	223
FUNCTION look_up_package_signal	224
ENTITY elaborated_process	224
ENTITY elaborated_sequential_statement	225

ENTITY elaborated_computation_statement	226
ENTITY elaborated_signal_assignment	226
ENTITY elaborated_waveform_element	226
ENTITY elaborated_waveform_expression	227
ENTITY elaborated_null_waveform_element	227
ENTITY elaborated_non_null_waveform_element	227
FUNCTION check_elaborated_waveform_elements	227
ENTITY elaborated_unguarded_signal_assignment	228
ENTITY elaborated_guarded_signal_assignment	228
ENTITY elaborated_wait_statement	228
FUNCTION get_sensitivity_signals	229
ENTITY elaborated_signal_wait	229
ENTITY elaborated_condition_wait	230
ENTITY elaborated_delay_wait	231
SCHEMA design_simulation_schema	233
ENTITY design_simulation	233
ENTITY simulation_cycle	234
ENTITY simulation_start_cycle	235
ENTITY simulation_subsequent_cycle	235
FUNCTION time_ordered_simulation_cycles	235
FUNCTION design_elaborated_signals	236
FUNCTION design_architecture_signals	236
FUNCTION design_configuration_signals	237
FUNCTION design_elaborated_processes	237
FUNCTION design_elaborated_signal_sources	238
FUNCTION previous_simulation_cycle	238
FUNCTION is_start_cycle	238
ENTITY simulated_process_state	238
FUNCTION process_state	239
ENTITY blocked_process_state	239
ENTITY active_process_state	240
ENTITY start_active_process_state	241
ENTITY subsequent_active_process_state	241
ENTITY executed_wait	242
FUNCTION get_wait_condition_result	242
ENTITY executed_signal_wait	243
ENTITY executed_condition_wait	243
ENTITY executed_delay_wait	243
ENTITY blocking_executed_wait	244
ENTITY blocked_executed_wait	244
ENTITY active_executed_wait	245
ENTITY active_signal_wait	245
ENTITY blocked_signal_wait	246
ENTITY active_condition_wait	246
ENTITY blocked_condition_wait	246
ENTITY active_delay_wait	246
ENTITY blocked_delay_wait	247
ENTITY executed_signal_assignment	247
ENTITY executed_unguarded_signal_assignment	248
ENTITY executed_guarded_signal_assignment	248
ENTITY active_guarded_assignment	249
ENTITY passive_guarded_assignment_of_guarded_signal	249
FUNCTION get_disconnection_delay	250
FUNCTION is_active_guard	250

ENTITY waveform_transaction	251
ENTITY null_waveform_transaction	251
ENTITY non_null_waveform_transaction	252
FUNCTION check_transaction_ordering	252
FUNCTION check_waveform_transactions	252
ENTITY simulated_signal_state	253
FUNCTION signal_state	253
FUNCTION is_active_signal	253
ENTITY simulated_guard_signal_state	254
FUNCTION compute_guard_value	254
ENTITY simulated_explicit_signal_state	255
ENTITY simulated_scalar_signal_state	256
ENTITY simulated_composite_signal_state	257
ENTITY resolved_composite_signal_state	257
ENTITY unresolved_composite_signal_state	258
ENTITY simulated_resolved_slice_state	258
ENTITY simulated_signal_source	259
FUNCTION source_state	259
ENTITY simulated_composite_signal_source	260
ENTITY simulated_scalar_signal_source	260
ENTITY port_connection	261
ENTITY signal_driver	262
ENTITY first_cycle_signal_driver	262
ENTITY subsequent_cycle_signal_driver	263
FUNCTION current_cycle_transaction	263
FUNCTION waveform_transform	263
FUNCTION update_waveform	264
FUNCTION transaction_predecessor	266
FUNCTION signal_driver_disconnection	266
FUNCTION signal_driver_value	266
ENTITY executed_VHDL_expression	267
ENTITY executed_time_expression	267
ENTITY executed_VHDL_function	268
ENTITY executed_resolution_function	268
ENTITY executed_conversion_function	269
ENTITY executed_driving_value_conversion	269
ENTITY executed_effective_value_conversion	269
FUNCTION get_source_values	269
FUNCTION get_connection_source	270
FUNCTION event_on_signal	270
FUNCTION scalar_signal_driving_value	270
FUNCTION scalar_signal_effective_value	271
FUNCTION scalar_signal_current_value	272
FUNCTION is_connected	272
FUNCTION port_connection_driving_value	272
FUNCTION port_connection_effective_value	272
FUNCTION resolved_signal_slice_value	273
FUNCTION build_aggregate	273
FUNCTION composite_source_value	273
FUNCTION composite_signal_value	274
FUNCTION composite_signal_driving_value	274
FUNCTION composite_signal_effective_value	275
FUNCTION composite_signal_current_value	275
SCHEMA time_schema	277

STANDARD PREVIEW
(standards.iteh.ai)

SIST-TP-CLC/R217-011:2004
<https://standards.iteh.ai/catalog/standards/sist/5640b91f-fee7-4fa7-9c73-c6346a483a3/sist-tp-clc-r217-011-2004>

CONSTANT start_time	277
TYPE Time_value	277
ENTITY Event	277
ENTITY Time_point	277
RULE linearise_time	278
FUNCTION is_next	278
FUNCTION is_concurrent	278
FUNCTION is_previous	279
FUNCTION is_before	279
FUNCTION is_after	279
Index	281
Glossary	287
Bibliography	293

iTeh STANDARD PREVIEW **(standards.iteh.ai)**

SIST-TP CLC/R217-011:2004
[https://standards.iteh.ai/catalog/standards/sist/5640b91f-feef-4fa7-9c73-
eb346a48f3a3/sist-tp-clc-r217-011-2004](https://standards.iteh.ai/catalog/standards/sist/5640b91f-feef-4fa7-9c73-eb346a48f3a3/sist-tp-clc-r217-011-2004)

Chapter 1

Information modelling of VHDL

The general aim of this chapter is to outline the power and the benefits which information modelling can provide to both the VHDL expert and the VHDL user according to the purpose of the model. More specifically, it is shown that (a) an information model of VHDL is worth considering as a viable alternative to other kinds models; (b) an information model can integrate different modelling goals and perspectives of VHDL more directly and uniformly; (c) an information model can be used to prove properties of the language, according to the model purpose; (d) there can be several VHDL information models depending on the goal of the model and the aspect of VHDL being considered and that these models can be organised into hierarchies for better support of complex applications, and (e) there are specific benefits of information modelling which can turn this paradigm into a valuable practical tool rather than a purely academic exercise. The discussion follows the plan shown below.

In section 1.2 a short introduction to information modelling, using the EXPRESS language [ISO91] is presented. It is followed in section 1.3 by an example of individual VHDL object modelling which is the simplest stratum of a VHDL model. Section 1.4 shows that the core model has to consider transformation processes which interrelate different levels of VHDL design: description, analysis, elaboration and simulation and that the relevant features of these transformations can be conveniently modelled. Section 1.5 suggests the way the behavioural semantics of VHDL can be modelled. The three modelling goals discussed, individual object, cross-level modelling and behavioural modelling form a modelling space which is presented in section 1.6. In sections 1.7 and 1.9 the core model concept is presented and it is shown how a core model can be used to prove properties of the language. Section 1.8 presents the problem of hierarchies of models according to the purpose and the perspective of VHDL modelling. Finally the modelling purpose and benefits are considered.

The discussion which follows applies to VHDL'87, called VHDL hereafter. Note that for brevity the examples in this chapter are not meant to be complete.

1.1 Models of VHDL

VHDL [IE88] [IE93] is intended for use as a standard hardware design language in the microelectronic design community. It describes general digital hardware devices at an abstract level. The description is performed at the structural, behavioural and timing levels. An abstract event-driven simulation model is used to define the semantics of the language constructs by specifying the way they are evaluated while the simulation process unfolds.

There are many attempts to describe the VHDL semantics, particularly its behavioural semantics. The main purpose of these efforts is to capture in a clear and unambiguous notation what is described informally in the VHDL reference manual and then to use the resulting descriptions for different purposes such as the informal comparison of VHDL with other HDLs, the formal proof of properties of VHDL programs, the implementation of related