Web: www.model.com

ModelSim 6.0

PSL Quick Guide

Operators

In order of precedence (LRM 1.1)

union	union operator
@	clock operator
[*]	consecutive repetition
[+]	consecutive repetition
[=]	non-consecutive repetition
[->]	goto repetition
within	sequence within operator - a sequence occurs during the course of another or within a time bounded interval
&	non-length-matching sequence conjunction
&&	length-matching sequence conjunction
|	sequence disjunction
:	sequence fusion
;	sequence concatenation
abort	termination operator - immediate termination of current and futrue obligations
eventually!	must hold at some time in the indefinite future
next*	must hold at some specified future time or range of future times
until*	must hold up to a given event
before*	must hold at some time before a given event
|->	overlapping suffix implication
|=>	non-overlapping suffix implication
->	logical IF implication
<->	logical IFF implication
always	must hold, globally
never	must NOT hold, globally

Email: support@model.com

ModelSim 6.0

PSL Quick Guide

Note: The asterisk * represents the entire family of related operators, i.e next, next_e, etc.

ModelSim commands

vcom/vlog -nopsl	Ignore embedded assertions
vcom/vlog -pslfile <name>	Specify external assertion file
vsim -nopsl	Ignore compiled assertions
assertion fail	Configure failure behavior
assertion pass	Configure pass behavior
assertion report	Produce report on assertions
fcover configure	Configure functional cover point
fcover report	Produce functional coverage report
fcover save	Save functional coverage database
fcover reload	Reload previously saved coverage database
vcover merge	Merge functional coverage databases off line
vcover stats	Compute statistical summary on saved functional coverage database
vcover report	Report coverage statistics on saved functional coverage database

PSL Quick Guide

ModelSim 6.0

ModelSim PSL assertion support

PSL is an Accellera standard that was born out of the Sugar language created at IBM. The syntax and semantics of PSL are described in the Property Specification Language Reference Manual, Version 1.1, published June 9, 2004. We strongly encourage you to get a copy of this specification.

In the current implementation, ModelSim supports only the simple subset of PSL (refer to Section 4.4.4, pg 25 of PSL LRM 1.1 for a description of this subset).

Verilog vs. VHDL flavors of PSL

Syntax	*Verilog*	*VHDL*
Declaration	**=**	**is**
Range	**[n:m]**	**[n to m]**
Path separator	**.**	**:**
Comment	**//**	**--**

ModelSim®

Web: www.model.com

Model*Sim* 6.0 PSL Quick Guide

Properties

Syntax

property <name> = Boolean | Sequence;
assert <name>;

Examples (Verilog syntax)

```
        property p0 = always a->b;
        assert p0;
        property check_write = always
        {(addr_out = addr_in[7:4]);
          ack};
        assert check_write;
```

Embedded assertions

prefix with "// psl"; "psl" is only necessary on the first line of a multi-line assertion

Example (Verilog syntax)

```
        // psl property s0 = always
        //  {b0; b1; b2};
        // psl assert s0;
```

Sequential expressions

group with curly braces ({ });
can be referenced in other properties

Example (VHDL syntax)

```
        sequence refresh_seq is
         {(cas and ras and we)[*2]; (not cas and not ras)};

        property check_refresh_rate is always {
         (not reset_n)[+]; rose(reset_n);
        (rose(refresh))[->1 to inf]}
         |->
        {[*18 to 32]; refresh_sequence};
        assert check_refresh_rate;
```

Email: support@model.com

Model*Sim* 6.0 PSL Quick Guide

Cover statement

Enable Functional Coverage metrics

Syntax <cover_name> cover : Sequence ;

Example (cover of Verilog sequence)

```
        sequence seq_n64 = {rose(MRxDV); MRxDV[*38]; (MRxD == 4'h0)[*2];
                (MRxD == 4'h2); (MRxD == 4'h4)};
        cover_seq_n64 : cover seq_n64;
```

Endpoint statement

PSL statement with boolean signal semantics

Syntax endpoint <name> = Sequence ;
endpoint end_seq_n64 = seq_n64;

Example (endpoint used in Verilog always block. See Cover statment for seq_64 deffiniton)

```
 always @(negedge mrx_clk)
 begin
      if (end_seq_n64  ) begin
             top.endpoint_n64       += 1;
             top.endpoint_small_pkt += 1;
      end
  ...
 end
```

Phone: 503-685-0820

Model*Sim* 6.0 PSL Quick Guide

External assertion files

Group assertions in verification units; compile with HDL source

Syntax

```
vunit name (hierarchical_HDL_design_unit)
{
        default clock is <clock_decl>;
        <PSL_stmts_and/or_HDL_decls_and_stmts>;
        ...
}
```

Example (VHDL syntax)

```
vunit check_dram_controller(dram_control(RTL))
{
        default clock is rising_edge(clk);
        sequence refresh_seq is {(cas and ras and we)[*2];
                (not cas and not ras)};
        ...
}
```

Mentor Graphics®

8005 SW Boeckman Road, Wilsonville, OR 97070
Phone: 503.685.0820 Toll free: 877.744.6699
Fax: 503.685.0910