Chapter 9
TIMING AND POWER ANALYSIS

The HSIM simulator provides a set of timing and power checking commands for timing and power analysis. Each command must be specified within the netlist file, or in a separate file, which is included into the netlist file by using the **. include** option. For information about **.include**, see .INCLUDE Statement on page 66.

The timing check command starts with the keyword **.tcheck**. The power check command starts with the keyword **.pcheck**. If there is a timing/power check error, it will be reported in the file hsim.chk, or out_file.chk if "-**o** out_file" is specified when invoking the HSIM simulator.

TIMING CHECKS

This section describes the HSIM timing checks.

SETUP TIME CHECK

SYNTAX:

```
.tcheck  title_name setup  sig_name  edge_type  ref_name
ref_edge_type  setup_time
<subckt=sub_name> <window=window_limit>  <vlth=logic_low_voltage>
<vhth=logic_high_voltage> <refvlth=ref_logic_low_voltage>
<refvhth=ref_logic_high_voltage>
```

PARAMETERS:

title_name	Defines the title name of this setup time check. Setup time errors during this timing check are listed in the timing/power check error file. The name of the setup time error starts with this title name.
sig_name	Defines signal node name(s) that can be the node name of a single node containing a wildcard character '*' that represents a group of node names. The node name with a wildcard character must be specified as v(node_name) or '*node_name*' because in the HSIM netlist parser as in SPICE syntax, all characters after wildcards are treated as comments and are ignored. When using the **subckt** parameter, *sig_name* is the node name inside the subcircuit—it should not be the full path name. If the **subckt** parameter is not used, then *sig_name* should be the full path name.
ref_name	Defines the reference node name. *ref_name* is the node name of a single node; it must not contain a wildcard character '*'. When

	using the **subckt** parameter, the `ref_name` is the node name inside the subcircuit, which must not be the full path name. If the **subckt** parameter is not used, then the *ref_name* must represent the full path name.
`edge_type`	Defines the permissible state transition type for signal node with **R**, **F**, and **X**. **R** indicates timing check is only performed when the state transition is from 0 (zero) to 1 (one). **F** indicates the 1 to 0 transition. **X** indicates any state transition—from 0 to 1 or from 1 to 0.
`ref_edge_type`	Defines the permissible state transition type for reference node with **R**, **F**, and **X**. **R** indicates timing check is only performed when the state transition is from 0 (zero) to 1 (one). **F** indicates the 1 to 0 transition. **X** indicates any state transition—from 0 to 1 or from 1 to 0.
`setup_time`	Defines the setup time. A setup timing error occurs when the time difference between the permissible state transition time of the reference node and the permissible state transition time of the signal node is less than *`setup_time`*. If *`setup_time`* is negative, *`window_ limit`* must be specified.
subckt	When this parameter is specified, the operation is performed to all instances of the particular subcircuit. When using the **subckt** parameter, *`sig_name`* and *`ref_name`* are the node names inside the subckt, and both *`sig_name`* and *`ref_name`* must not contain a wildcard character. In the HSIM netlist parser as in SPICE syntax, all characters after wildcards are treated as comments and are ignored.
`window_ limit`	When *`window_ limit`* is specified, the setup timing check error occurs when the signal node has a permissible transition at the time interval (t_ref - setup_time, t_ref + window_limit). If *`setup_time`* is negative, *`window_ limit`* must be specified. If *`setup_time`* is positive and *`window_ limit`* is not specified, the setup timing check error occurs when the signal node has a permissible transition at the time interval (t_ref - setup_time, t_ref).
`logic_ low_voltage`	Represents the threshold voltage of logic state 0 (zero). The signal node is in logic 0 state if its node voltage is lower than *`logic_ low_voltage`*.
`logic_high_voltage`	Represents the threshold voltage of logic state 1 (one). The signal

node is in logic 1 state if the signal node voltage is higher than logic_high_voltage.

ref_logic_low_voltage
: Represents the low logic threshold voltages for the reference node. When *logic_low_voltage* is specified, *ref_logic_low_voltage* is defined to the same value if it is not separately defined by **refvlth**.

ref_logic_high_voltage
: Represents the high logic threshold voltages for the reference node. When *logic_high_voltage* is specified, *ref_logic_high_voltage* follows the same value if it is not separately defined by **refvhth**.

EXAMPLE:

```
.tcheck  check1  setup  data1  x  clk  f  2.1n
.tcheck  check2  setup  v(x1.*.data1)  x  clk  f  2.1n
.tcheck  check33 setup  d  x  clk  r  1.5n  subckt=dff
```

A setup time error occurs when a state transition at node *data1* occurs less than 2.1 ns before the fall transition of the node clk. The error is reported following the title name of *check1*. Similar checks are performed for all nodes that match the pattern x1.*.*data1*. The setup timing error, if any, is reported in the timing/power check file following the title name of *check2*.

EXAMPLE:

.tcheck check33 **setup** d x clk r 1.5n **subckt**=dff

subckt is specified in which d and clk are the node names inside the **subckt**. The setup time check reports error for every dff subcircuit instance if the state transition at data pin d occurs less than 1.5 ns before the rise transition at the clk pin within the same dff subcircuit instance.

EXAMPLE:

```
.tcheck  check3  setup  qn  f  clk  r  -0.5n  window=10n
```

A setup time error, which follows the title name of check3, is reported when all of the following conditions occur: a fall state transition at node qn occurs 0.5 ns after the node ck has a rise transition; this fall state transition at qn occurs no later than 10 ns after the rise transition at node ck.

HOLD TIME CHECK

SYNTAX:

.tcheck *title_name* **hold** *sig_name* *edge_type* *ref_name* *ref_edge_type* *hold_time*

<**subckt**=*sub_name*> <**window**=*window_limit*> <**vlth**=*logic_low_voltage*>

<**vhth**=*logic_high_voltage*> <**refvlth**=*ref_logic_low_voltage*>

<**refvhth**=*ref_logic_high_voltage*>

PARAMETERS:

title_name	Defines the title name of this hold time check. Hold time errors during this timing check are listed in the timing/power check error file and starts with this title name.
sig_name	Defines signal node name(s) that can be the node name of a single node, or a node name containing wildcard character '*' that represents a group of node names. The node name with wildcard character must be specified as v(node_name). In the HSIM netlist parser as in SPICE syntax, all characters after the wildcard character "*" are treated as comments and are ignored. When using the **subckt** parameter, `sig_name` is the node name inside the subcircuit—it must not be the full path name. If the **subckt** parameter is not used, then `sig_name` should be the full path name.
ref_name	Defines the reference node name. *ref_name* is the node name of a single node, and it cannot contain wildcard character "*". When using the **subckt** parameter, the *ref_name* is the node name inside the subcircuit—it must not be the full path name. If the **subckt** parameter is not used, then the ref_name must be the full path name.
edge_type	Defines the permissible state transition type for the signal node with **R**, **F**, and **X**. **R** indicates timing check is only performed when the state transition is from 0 (zero) to 1 (one). **F** indicates the 1 to 0 transition. **X** indicates any state transition—from 0 to 1 or from 1 to 0.
ref_edge_type	Defines the permissible state transition type for the reference node with **R**, **F**, and **X**. **R** indicates timing check is only performed when the state transition is from 0 (zero) to 1 (one). **F** indicates the 1 to 0 transition. **X** indicates any state transition—from 0 to 1 or from 1 to 0.
hold_time	Defines the hold time. A hold timing error occurs when the time difference between the permissible state transition time of signal node and the permissible state transition time of reference node is less than `hold_time`. If `hold_time` is negative, `window_limit` must be specified.
subckt	When this parameter is specified, the operation is performed to all instances of the particular subcircuit. When using the **subckt** parameter, `sig_name` and `ref_name` are the node names inside the subckt, and both `sig_name` and `ref_name` must not contain the wildcard character "*". In the HSIM netlist parser as in SPICE syntax, all characters after wildcards are treated as comments and are ignored.

window_limit	When *window_limit* is specified, the setup timing check error occurs when the signal node has a permissible transition at the time interval (*t_ref – window_limit, t_ref + hold_time*). If *hold_time* is negative, *window_limit* must be specified. If *hold_time* is positive and *window_limit* is not specified, the hold timing check error occurs when the signal node has a permissible transition at the time interval (*t_ref, t_ref+hold_time*).
logic_low_voltage	Defines the threshold voltage of the logic 0 (zero) state of the signal node. The signal node is in the logic 0 state if its node voltage is less than *logic_low_voltage*.
logic_high_voltage	Defines the threshold voltage of logic 1 state of the signal node. The signal node is in logic 1 state if the signal node voltage is greater than *logic_high_voltage*.
ref_logic_low_voltage	Defines the low logic threshold voltage for the reference node. When *logic_low_voltage* is specified, *ref_logic_low_voltage* will be defined to the same value if it is not separately defined by **refvlth**.
ref_logic_high_voltage	Defines the high logic threshold voltage for the reference node. When *logic_high_voltage* is specified, *ref_logic_high_voltage* will be defined to the same value if it is not separately defined by **refvhth**.

EXAMPLE:

```
.tcheck  check3  hold  data  x  clk  f  2.1n
```

A hold time error occurs when any state transition at node data occurs at the time interval (t, t+2.1ns), where t is the time when node clk has a fall state transition. The error is reported following the title name of check3.

EXAMPLE

```
.tcheck  check4  hold  qn  f  ck  r  -0.5n  window=10n
```

A hold time error, following the title name of check4, is reported when a fall state transition at node qn occurs at the time interval (t – 10 ns, t – 0.5 ns) in which where **t** is the time when the node ck has a rise transition.

PULSE WIDTH CHECK

This command checks whether the pulse width (time difference between rise and fall state transitions) of the specified node meets the required range.

SYNTAX:

.tcheck *title_name* **pulsew** *node_name* *low_min_time* *low_max_time* *high_min_time*

high_max_time <**subckt**=*sub_name*> <**vlth**=*logic_low_voltage*> <**vhth**=*logic_high_voltage*>

PARAMETERS:

title_name	Defines the title name of this pulse width check. Any violation against the required pulse width range resulted from this check will be listed in the timing/power check error file, and starts with this title name.
node_name	Defines signal node name(s) which can be the node name of a single node or a node name containing wildcard character "*" representing a group of node names. The node name with a wildcard character must be specified as v(node_name) or '*node_name*'. In the HSIM netlist parser as in SPICE syntax, all characters after a wildcard are treated as comments and are ignored. When using the subckt parameter, '*node_name*' is the node name inside the subcircuit, and should not be the full path name. If the subckt parameter is not used, then the *node_name* must be the full path name.
logic_low_voltage	Defines the threshold voltage of logic 0 stat. The node has a logic 0 state if its node voltage is lower than *logic_low_voltage*.
logic_high_voltage	Defines the threshold voltage of logic 1 state of the node. The node is in the logic 1 state if the node voltage is greater than *logic_high_voltage.*
low_min_time	Defines the minimum value of the low-state pulse. Each low-state pulse (the time period the node stays at the logic 0 state) is required to be greater than *low_min_time* and less than *low_max_time*.
low_max_time	Defines the maximum value of the low-state pulse. Each low-state pulse (the time period the node stays at the logic 0 state) is required to be greater than *low_min_time* and less than *low_max_time*.
high_min_time	Defines the minimum value of the high-state pulse. Each high-state pulse (the time period the node stays at the logic 1 state) is required to be greater than *high_min_time* and less than *high_max_time*.
high_max_time	Defines the maximum value of the high-state pulse. Each high-state pulse (the time period the node stays at the logic 1 state) is required to be greater than *high_min_time* and less than *high_max_time*.
subckt	When this parameter is specified, the operation is performed to all instances of the particular subcircuit. When using the **subckt**

	parameter, `sig_name` and `ref_name` are the node names inside the subckt in which both `sig_name` and `ref_name` must not contain the wildcard character "*". In the HSIM netlist parser as in SPICE syntax, all characters after wildcard characters are treated as comments and are ignored.

EXAMPLE:

```
.tcheck  check5  pulsew  data[3]  8n 11n  7n  9n
```

A pulse width violation error, following the title name of check5, is reported when the low pulse width at node data[3] is less than 8 ns or greater than 11 ns; or when the high pulse width is less than 7 ns or greater than 9 ns.

```
In this example because vlth is not defined the default value is
used: 3V*0.3 = 0.9V. The default value for vhth is 3V*0.7 = 2.1V
```

TIMING EDGE CHECK

This command checks the time delay between two specified nodes and reports error when the delay doesn't meet the specified range.

SYNTAX:

```
.tcheck  title_name  edge  sig_name  edge_type  ref_name
ref_edge_type  min_time  max_time <subckt=sub_name>
<window=window_limit>  <vlth=logic_low_voltage>
<vhth=logic_high_voltage> <refvlth=ref_logic_low_voltage>
<refvhth=ref_logic_high_voltage> <trigger=0|1|2>
```

PARAMETERS:

title_name	Defines the title name of this timing edge check. Every violation resulting from this timing check is listed in the timing/power check error file and starts with this title name.
sig_name	Defines signal node name(s) which can be the node name of a single node containing wildcard character "*" representing a group of node names. The node name with wildcard character must be specified as v(node_name). In the HSIM netlist parser as in SPICE syntax, all characters after a wildcard are treated as comments and are ignored. When using the subckt parameter, *sig_name* is the node name inside the subcircuit, and must not be the full path name. If the subckt parameter is not used, then the *sig_name* should be the full path name
ref_name	Defines the reference node name. *ref_name* is the node name of a single node, and it cannot contain wildcard character.

	When using the **subckt** parameter, the *ref_name* is the node name inside the subcircuit—it must not be the full path name. If the **subckt** parameter is not used, then the ref_name should be the full path name.
edge_type	Defines the permissible state transition type for the signal node with **R**, **F**, or **X**. **R** indicates timing check is only performed when the state transition is from 0 (zero) to 1 (one). **F** indicates the 1 to 0 transition. **X** indicates any state transition—from 0 to 1 or from 1 to 0.
ref_edge_type	Defines the permissible state transition type for the reference node with **R**, **F**, or **X**. **R** indicates timing check is only performed when the state transition is from 0 (zero) to 1 (one). **F** indicates the 1 to 0 transition. **X** indicates any state transition—from 0 to 1 or from 1 to 0.
min_time	Defines the lower boundary of the timing edge difference between the signal and reference nodes. A timing edge error is reported when the timing edge difference is less than *min_time*. The timing edge difference is calculated only for the pair of permissible state transitions at the signal node and the reference node.
max_time	Define the upper boundary of the timing edge difference between the signal and reference nodes. A timing edge error is reported when the timing edge difference is greater than *max_time*. The timing edge difference is calculated only for the pair of permissible state transitions at the signal node and the reference node.
window_limit	Eliminates a timing edge error report if the timing edge difference exceeds window_limit. *window_limit* is optional.
l*ogic_low_voltage*	Defines the threshold voltage of the logic 0 state of the signal node. The stat of the signal node is logic 0 if its node voltage is lower than *logic_low_voltage*.
logic_high_voltage	Defines the threshold voltage of the logic 1 state of the signal node. The state of the signal node is logic 1 if the signal node voltage is higher than *logic_high_voltage*.
ref_logic_low_voltage	Defines the low logic threshold voltage for the reference node. When *logic_low_voltage* is specified, *ref_logic_low_voltage* will be defined to the same value if not separately defined by **refvlth**.
ref_logic_high_voltage	Defines the high logic threshold voltage for the reference node. When *logic_high_voltage* is specified, *ref_logic_high_voltage* will be defined to the same value if not separately defined by **refvhth**.

subckt	When this parameter is specified, the operation is performed to all instances of the particular subcircuit. When using the **subckt** parameter, `sig_name` and `ref_name` are the node names inside the subckt in which both `sig_name` and `ref_name` must not contain a wildcard character. In the HSIM netlist parser as in SPICE syntax, all characters after wildcards are treated as comments and are ignored.
trigger value	Defines the condition to trigger the timing edge check. The default value is 0. The trigger values are as follows: at trigger value 0, any permissible state transition at either signal node or reference node triggers the timing edge check; at trigger value 1, only the permissible state transition at the signal node triggers the check; at trigger value 2, only the permissible state transition at the reference node triggers the check. Trigger value is optional.

EXAMPLE:

```
.tcheck  check6  edge  data  x  ctrl  r  2n  5n
```

When node data has a state transition such as *time t2*, the time edge difference of t2-t1 must be within the range of 2 ns and 5 ns. Otherwise, a timing edge error is reported following the title name of check6. Time t1 is the most recent rise state transition time at node ctrl that occurs before time t2. When node ctrl has a rise state transition at time t4, the time edge difference t4-t3 must be within the range of 2 ns and 5 ns. Otherwise a timing edge error is also reported. The time t3 is the most recent state transition time at node data before time t4.

EXAMPLE:

```
.tcheck  check7  edge  data  x  ctrl  r  2n  5n  trigger=2
```

Similar to the above example, except the edge error check is triggered only by the rise state transition at node ctrl. Any edge error is reported following the title name of check7.

EXAMPLE:

```
.tcheck  check8  edge  data  x  ctrl  r  2n  5n  trigger=2
window=10n
```

When node ctrl has a rise state transition at time t1, an edge error is reported (following the title name of check8) only when t1-t2 is less than 2ns or is less than 10ns but greater than 5ns. The time t2 is the most recent state transition time at node data before time t1.

TIMING CHECK WINDOWS

This command defines the timing windows for the timing check commands.

SYNTAX:

```
.tcheck window start_time1 <stop_time1 <start_time2 <stop_time2
….>>>
```

All timing check commands use the same set of windows specified by this command. If this command is not specified, the default window for all timing check commands will be from the beginning of simulation to the end of simulation.

Multiple timing check windows can be specified by providing multiple pairs of start and stop time values. If the stop time is not provided in the last window, the timing check will be extended to the end of simulation.

EXAMPLE:

```
.tcheck  window  10n  20n  110n  120n  210n
```

In this example, the command specifies three timing check windows. The first window is from 10 ns to 20 ns, the second window is from 110 ns to 120 ns, and the third window is from 210 ns until the end of simulation.

BISECTION OPTIMIZATION

The HSIM simulator supports bisection optimization. The bisection results are printed in the .optz file. The results of the **.measure** and **.print** statements of the last bisection iteration are stored in the .mt and .fsdb/.out files, respectively. Several statements are required to perform bisection optimization.

```
.model  statement
.param  statement
.tran  statement
```

SYNTAX FOR THE **.MODEL** STATEMENT

```
.model  opt_model_name  opt  method=bisection| passfail
```

The optimization model reference name is *opt_model_name*. The keyword **opt** indicates that this particular **.model** statement is for bisection optimization usage. The same name is used in the corresponding **.tran** statement. Parameter **method** specifies the method to use in the bisection optimization—a valid value is either *bisection* or *passfail*.

SYNTAX FOR THE **.PARAM** STATEMENT:

```
.param param_name= optxxx(init_value, low_value, upper_value)
```

The name of the parameter used in bisection optimization is *param_name*. **opt**xxx is the selected optimization parameter reference name—xxx can be replaced with a suitable choice. The same **opt**xxx name is referenced in the corresponding **.tran** statement. The initial value, the lower boundary, and the upper boundary of the parameter are specified as *init_value*, *low_value*, and *upper_value*, respectively.

SYNTAX FOR THE **.TRAN** STATEMENT:

```
.tran  steptime  stoptime  sweep  optimize=optxxx
result=measure_var model= opt_model_name
```

The step time and stop time are specified as steptime and stoptime, respectively. **opt**xxx is the same optimization parameter reference name in the corresponding .**param** statement. Parameter **result** specifies the measure variable defined in .**measure** statement. Parameter **model** is specified with the same model optimization reference name in the corresponding **.model** statement.

EXAMPLE:

```
.model  optmod  opt  method=bisection
.param  delaytime= opt1(1n,  0n,  20n)
.tran  0.1n  40n  sweep  optimze=opt1  result=maxvout  model=optmod
```

POWER CHECK

This section describes power check commands.

DC PATH CHECK

This command checks the dc current path(s) among voltage sources in the circuit.

SYNTAX:

```
.pcheck  title_name  dcpath  <ith=threshold_current> <node1 <node2 .
. .>>
<period= period_time | delay=delay_time> <start=start_time1
<stop=stop_tim1 <start=start_time2 <stop=stop_time2 . . .>>>>
```

or

```
.pcheck  title_name  dcpath  <ith=threshold_current> <node1
<node2. . .>> at=t1 <at=t2 …>
```

All reported dc current paths are written into a separate file. This prevents mixing data with other violation reported by other timing and power checks.

PARAMETERS:

title_name	Each dc path is reported in the file *title_name*.
threshold_current	Defines the threshold current. Each reported dc path current exceeds *threshold_current.* The default value is 50 uA.
node1, node2 .. nodeN	The specified names define the nodes that dc current path checking will be performed between these nodes.
dcpath	The dc path search starts from any node specified in this node list and ends when it reaches another node in the list. If no node is specified, then the HSIM simulator reports dc current path(s) between any pair of voltage source nodes.
period	When specified, the dc path is checked for every *period_time* starting from the start time of each specified window defined by *start_time* and *stop_time*.
delay	When specified, the dc current path is checked at each time defined as *t+delay_time*. *t* is the time an input voltage source changes to a new voltage level. The parameters **period** and **delay** cannot be used simultaneously.

at	When specified, the dc current path is checked at the time defined by **at**=t1.
start	Specifies the *start_time* of the window.
stop	Specifies the *stop_time* of the window.

EXAMPLE:

```
.pcheck  dc1 dcpath  ith=1e-6  vdd gnd  delay=5n  start=10n
stop=210n
```

This command checks each dc current path between vdd and gnd every 5 ns after any input voltage source changes its voltage level. The dc current path is checked only during the time window between 10 ns and 210 ns. The dc current path is reported in the file *dc1* if the dc path current exceeds 1 uA when checked.

EXAMPLE:

```
.pcheck  dc2 dcpath ith=1e-6 vcc vss  period=10n  start=10n
stop=210n
```

This command checks each dc current path between vcc and vss at every 10 ns starting from simulation time at 10 ns and simulation time at 210 ns (included). The dc current path is reported in the file dc2 if the dc path current exceeds 1 uA when checked.

EXAMPLE:

```
.pcheck  dc3 dcpath  vcc vss  at=130n  at=150n
```

This command checks each dc current path between vcc and vss at 130 ns time and 150 ns time. The dc current path is reported in the file dc3 if the dc path current exceeds the default value of 50 uA at 130 ns or at 150 ns time.

EXCESSIVE CURRENT CHECK

The excessive element-current check **exi** checks when the current of specified element(s) exceeds a threshold value.

SYNTAX:

```
.pcheck  title_name exi  elem1 <elem2…> <ith=threshold_current>
<tth=exi_time>  <start=start_time1 <stop=stop_time1
<start=start_time2 <stop=stop_time2 …>>>>
```

PARAMETERS:

title_name	Defines the title name of this excessive current check. Each excessive current report resulted from this check is listed in the timing/power check error file, and starts with this title name. The excessive current check is applied to each specified element *elem1*, *elem2* . . . *elemN*. The element name can include the wildcard character '*' —the excessive current check applies to each element with matching pattern.
	The element name with a wildcard character must be specified as

	i(*elem_name)* or *'elem_name'* because in the HSIM netlist parser as in SPICE syntax, all characters after '*' are treated as comments and are ignored.
ith	Defines *threshold_current*, the value of the threshold current. An element is reported to have excessive current if its element current exceeds *threshold_current* for time duration greater than *exi_time.*
tth	Defines *exi_time*, the time duration. An element is reported to have excessive current if its element current exceeds *threshold_current* for time duration greater than *exi_time*

start_time1, stop_time1 . . . start_timeN, stop_timeN..

Specify the time window(s). The excessive element current is checked at the time within the specified time window(s). If no time window is specified, the check is performed from the time 0 ns to the end of simulation.

EXAMPLE:

```
.pcheck  largei exi  m1 m2 i(x1.*) ith=1e-3 tth=3n start=100n
```

This command checks if the elements m1, m2, and all elements within instance x1 have current greater than 1mA for longer than 3ns. Any detected excessive current element after 100 ns is reported in the timing/power check f*ile out_file.chk* following the title name of largei. The name *out_file* is the output file prefix with the default name hsim.

EXCESSIVE RISE/FALL TIME CHECK

The excessive rise/fall time check **exrf** checks if the specified node(s) have excessive rise and/or fall times.

SYNTAX:

```
.pcheck  title_name  exrf  node1 <node2 …> <fanout=val2>
<rise=rtime> <fall=ftime> <utime= utime1> <vlth=logic_low_v>
<vhth=logic_high_v> <start=start_time1 <stop=stop_time1
<start=start_time2 <stop=stop_time2 …>>>>
```

PARAMETERS:

title_name	Any excessive rise/fall time violation is reported in the timing/power check file xxxx.chk following the title name of *title_name*.
node1, node2 …nodeN	Defines signal node name which can be the node name of a single node or a node name with the wildcard character '*' that represents a group of node names. The node name with wildcard character must be specified as v(node_name) or *'node_name'* because in the HSIM netlist parser as in SPICE syntax, all characters after '*' are treated as comments and are ignored.
fanout	Defines the fanout of driver nodes. Set to 1 (one), **fanout** limits the rise/fall time to checking the driver nodes with fanouts, which

	avoids unnecessary check on internal nodes within logic gates. Set to 0 (zero), **fanout** checks either the internal node or the driver node. The default value is 0. This parameter is optional.
`rise`	Defines the rise time of the signal as time duration *`t2-t1`*. *`t1`* is the time the rising signal voltage crosses the voltage level *`logic_low_v`*. *`t2`* is the time when the same continuously rising signal voltage crosses the voltage level *`logic_high_v`*. The default value is of *`rtime`* is 5 ns.
`fall`	The fall time (*`ftime`*) of the signal is defined as *`t4-t3`*. *`t3`* is the time when the falling signal voltage crosses the voltage *`logic_high_v`*. *`t4`* is the time when the same continuously falling signal voltage crosses the voltage *`logic_low_v`*. The default value is 5 ns.
`utime`	The *`U-state`* time (*`utime1`*)of the signal is defined as *`t6-t5`*. *`t5`* is the time signal voltage enters the U state ,which occurs when the signal voltage is between *`logic_low_v`* and *`logic_high_v. t6`* is the time the signal voltage leaves the *`U-state`* and the signal voltage is the same as the earlier voltage at t5— *`U-state`* represents an incomplete rise or fall transition. The default value is 5 ns.
`vlth`	Defines the logic low state (*`logic_low_v).`*
`vhth`	Defines the logic high state (*`logic_high_v`*).
`exrf`	The **exrf** check is performed within the time window defined by start_time1, stop_time1 . . . start_time*N*, stop_time*N*. An excessive rise/fall time violation is reported in the following conditions. • The signal rise time exceeds rtime if it is specified following *`rtime,`* or *`utime1`*. • The signal fall time exceeds ftime if it is specified following *`ftime,`* or *`utime1`*. • The *`U-state`* duration exceeds *`utime1`*.

EXAMPLE:

```
.pcheck  longrf  exrf  a1 a2 v(x1.x2.*)  fanout=1
+ rise=3n  fall=4n  vlth=0.3 vhth= 2.7  start=100n  stop=1000n
```

This command checks if the signal voltages at nodes a1, a2, and the driver nodes that match the pattern of x1.x2.* have excessive rise and fall times. An **exrf** is checked between 100 ns and 1,000 ns. A violation is reported in the timing/power check file following the title name of longrf under the following conditions: the signal rise time exceeds 3 ns, or the signal fall time exceeds 4 ns, or the *`U-state`* time exceeds the defaulted 5 ns. The *`U-state`* is defined as the voltage between 0.3 V and 2.7 V.

HIGH IMPEDANCE NODE CHECK

This command checks for the high impedance state in the circuit.

SYNTAX:

```
.pcheck  title_name zstate node1 <node2 …> <fanout=val2>
<ztime=ztime1>
<start=start_time1 <stop=stop_time1 <start=start_time2
<stop=stop_time2 …>>>>
```

title_name	Every high-impedance is reported in the high-impedance check file *xxxx.chk* following the title name of *title_name*.
node1 .. nodeN	Each name, specified by *node1 . . .nodeN* defines signal node name. Each name can be the node name of a single node or a node name containing wildcard character **'*'** which represents a group of node names. The node name with wildcard character must be specified as v(*node_name*) or '*node_name'*, because in the HSIM netlist parser--similar to the SPICE simulator, all characters after **'*'** are treated as comments and are ignored.
fanout	The optional setting **fanout**=1 limits the high impedance state checking to those driver nodes with fanouts. This setting will avoid unnecessary check on internal nodes within logic gates. The default **fanout**=0 will check either internal node or driver node.
zstate	This check is performed within the time window defined by *start_time1, stop_time1 .. start_timeN, stop_timeN*.
ztime	A specified node that stays in the high-impedance state longer than *ztime1* will be reported in the timing/power check file following the title name of name1. The default value of **ztime** is 5 ns.

EXAMPLE:

```
.pcheck  highz zstate  a1 a2 v(x1.x2.*) ztime=50n  start=100n
stop=1000n
```

This command checks the high-impedance state of the nodes a1, a2 and the nodes that match the pattern of x1.x2.*. The high-impedance state is checked between 100 ns and 1,000 ns. If a specified node remains in the high impedance state longer than 50 ns, it will be reported in the timing/power check file following the title name of *highz*.

POWER CHECK WINDOWS

This command defines the time window(s) for each power check that doesn't have its own time window specification. Each power check with its own time window specification will not be affected by this **window** setting..

SYNTAX

```
.pcheck   window start_time1 <stop_time1 <start_time2
<stop_time2 …>>>
```

EXAMPLE:

```
.pcheck  window  10n  20n 110n 120n 210n
```

This sets three time windows for those power checks without time window specification. The time windows are as follows: from 10 ns to 20 ns, from 110 ns to 120 ns, and from 210 ns to the end of simulation.