APPENDIX

Part Five

Trial Exhibit 1441, pages 191-259

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793821

si_filter_per_aggr_noise_peak_ratio

Specifies the threshold for the voltage bump introduced by an aggressor at a victim node, divided by Vcc, below which the aggressor net can be filtered out during electrical filtering.

TYPE

float

DEFAULT

0.01

DESCRIPTION

Specifies the threshold for the voltage bump introduced by an aggressor at a victim node; the default is 0.01. This variable, along with **si_filter_accum_aggr_noise_peak_ratio**, makes up a pair of variables used by PrimeTime-SI during the electrical filtering phase, to determine whether an aggressor net can be filtered.

An aggressor net, along with its coupling capacitors, is filtered when either of the following are true:

1. The peak voltage of the voltage bump induced on the victim net divided by Vcc is less than the value of **si_filter_per_aggr_noise_peak_ratio**.

2. The accumulated peak voltage of voltage bumps induced on the victim by aggressors to the victim net divided by Vcc is less than the value of **si_filter_accum_aggr_noise_peak_ratio**.

Parasitic filtering criteria previously checked are controlled by the variables **si_filter_per_aggr_xcap**, **si_filter_per_aggr_xcap_to_gcap_ratio**, and **fi_filter_single_to_average_all_xcap_ratio**.

To determine the current value of this variable, type **printvar si_filter_per_aggr_noise_peak_ratio**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018567

filter_per_aggr_noise_peak_ratio

Persistent parameter that specifies the noise peak voltage threshold below which an aggressor is ignored. The threshold is specified as a fraction of the supply voltage.

Syntax:

`filter_per_aggr_noise_peak_ratio` *`ratio`*

where *`ratio`* is the minimum noise peak value as a fraction of the supply voltage below which the noise peak is ignored.

The default value is *1.000e-02*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793822-23

si_filter_per_aggr_to_average_aggr_xcap_ratio

Specifies the minimum value of the ratio of the total cross-coupled capacitance between the aggressor net and the victim net to the average cross-coupled capacitance between the victim net and all of its aggressor nets, below which an aggressor net can be filtered out during parasitic filtering.

TYPE

float

DEFAULT

0.0

DESCRIPTION

Specifies the threshold of the ratio of the total cross-coupled capacitance between the aggressor net and the victim net to the average cross-coupled capacitance between the victim net and all of its aggressor nets. This variable, along with **si_filter_per_aggr_xcap** and **si_filter_per_aggr_xcap_to_gcap_ratio**, makes up a set of three variables used by PrimeTime-SI during the second stage of the parasitic filtering phase, to determine whether an aggressor net can be filtered for a particular victim net. If it meets all of the filtering criteria described in the section labeled "Filtering Criteria", a net is filtered as an aggressor net for that particular victim net. However, the aggressor net can still be considered as an aggressor net to another victim net. Note that a coupling capacitor can be filtered at one end and not at the other.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018570

si_filter_per_aggr_to_average_aggr_xcap_ratio

Persistent parameter that specifes whether to ignore the aggressor net if the ratio of the total cross-coupled capacitance between the aggressor net and the victim net to the average cross-coupled capacitance between the victim net and all of its aggressor nets is less than this value.

Syntax:

`si_filter_per_aggr_to_average_aggr_xcap_ratio value`

The default values are *0*.

Pl's Ex.336 (PrimeTime 2006.12) at ATopTech 1793824

si_filter_per_aggr_xcap

Specifies the minimum value of the total cross-coupled capacitance between the aggressor net and the victim net, below which an aggressor net can be filtered out during parasitic filtering.

TYPE

float

DEFAULT

0.0

DESCRIPTION

Specifies the threshold, in library units, of the total cross-coupled capacitance between the aggressor net and the victim net. This variable, along with **si_filter_per_aggr_xcap_to_gcap_ratio** and **si_filter_per_aggr_to_average_aggr_xcap_ratio**, makes up a set of three variables used by PrimeTime-SI during the second stage of the parasitic filtering phase, to determine whether an aggressor net can be filtered for a particular victim net. If it meets all of the filtering criteria described in the section labeled "Filtering Criteria", a net is filtered as an aggressor net for that particular victim net. However, the aggressor net can still be considered as an aggressor net to another victim net. Note that a coupling capacitor can be filtered at one end and not at the other.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 018570

si_filter_per_aggr_xcap

Persistent parameter that specifes whether to ignore the aggressor net if the total cross-coupled capacitance between the aggressor net and the victim net is less than the specified value.

Syntax:

`si_filter_per_aggr_xcap` *value*

The default values are *0*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793826-27

si_filter_per_aggr_xcap_to_gcap_ratio

Specifies the minimum value of the ratio of the total cross-coupled capacitance between the aggressor net and the victim net to the total ground capacitance of the victim net, below which an aggressor net can be filtered out during parasitic filtering.

TYPE

float

DEFAULT

0.0

DESCRIPTION

Specifies the threshold, in library units, of the ratio of the total cross-coupled capacitance between the aggressor net and the victim net to the total ground capacitance of the victim net. This variable, along with **si_filter_per_aggr_xcap** and **si_filter_per_aggr_to_average_aggr_xcap_ratio**, makes up a set of three variables used by PrimeTime-SI during the second stage of the parasitic filtering phase, to determine whether an aggressor net can be filtered for a particular victim net. If it meets all of the filtering criteria described in the section labeled "Filtering Criteria", a net is filtered as an aggressor net for that particular victim net. However, the aggressor net can still be considered as an aggressor net to another victim net. Note that a coupling capacitor can be filtered at one end and not at the other.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018571

si_filter_per_aggr_xcap_to_gcap_ratio

Persistent parameter that specifes whether to ignore the aggressor net if the ratio of the total cross-coupled capacitance between the aggressor net and the victim net to the total ground capacitance of the victim net is less than the specified value.

Syntax:

`si_filter_per_aggr_xcap_to_gcap_ratio` *value*

The default values are *0*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793828-29

si_filter_total_aggr_xcap

Specifies the minimum value of the total cross-coupled capacitance between the victim net and all aggressor nets, below which a victim net can be filtered out during parasitic filtering.

TYPE

float

DEFAULT

0.0

DESCRIPTION

Specifies the threshold, in library units, of the total cross- coupled capacitance between the victim net and all aggressor nets. This variable, along with **si_filter_total_aggr_xcap_to_gcap_ratio**, makes up a pair of variables used by PrimeTime-SI during the first stage of the parasitic filtering phase, to determine whether a victim net can be filtered. If it meets all of the filtering criteria described in the section labeled "Filtering Criteria", a victim net is filtered. However, the victim net can still be considered as an aggressor net. Note that a coupling capacitor can be filtered at one end and not at the other.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 001857-72

si_filter_total_aggr_xcap

Persistent parameter that specifes whether to ignore the victim net if the total cross-coupled capacitance between the victim net and all aggressor nets is less than the specified value.

Syntax:

si_filter_total_aggr_xcap *value*

The default values are *0*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793830-31

si_filter_total_aggr_xcap_to_gcap_ratio

Specifies the minimum value of the ratio of total cross-coupled capacitance to the total ground and cross-coupled capacitance, below which a victim net can be filtered out during parasitic filtering.

TYPE

float

DEFAULT

0.0

DESCRIPTION

Specifies the threshold of the ratio of total cross-coupled capacitance to the total ground and cross-coupled capacitance of a victim net. This variable, along with **si_filter_total_aggr_xcap**, makes up a pair of variables used by PrimeTime-SI during the first stage of the parasitic filtering phase, to determine whether a victim net can be filtered. If it meets all of the filtering criterial described in the section labeled "Filtering Criteria", a victim net is filtered. However, the victim net can still be considered as an aggressor net. Note that a coupling capacitor can be filtered at one end and not at the other.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018572

si_filter_total_aggr_xcap_to_gcap_ratio

Persistent parameter that specifes whether to ignore the victim net if the ratio of total cross-coupled capacitance to the total ground and cross-coupled capacitance is less than the specified value.

Syntax:

`si_filter_total_aggr_xcap_to_gcap_ratio` *value*

The default values are ***0***.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793833

si_noise_composite_aggr_mode

Specifies the composite aggressor mode for noise analysis.

TYPE

String

DEFAULT

disabled

DESCRIPTION

This variable specifies which composite aggressor mode is used in PrimeTime SI noise analysis. Allowed values are *disabled* (the default), which turns off the composite aggressor feature. *normal*, causes PrimeTime SI to calculate noise by utilizing the non-statistical composite aggressor feature. Selecting *statistical* causes PrimeTime SI to calculate noise by using the statistical composite aggressor flow.

In *disabled* composite aggressor mode, PrimeTime SI uses its original flow with composite aggressor completely off to analyze the noise.

In *normal* composite aggressor mode, PrimeTime SI aggregates the effect of multiple small aggressors into a single composite aggressor, thereby reducing the computational complexity and improving the performance.

statistical composite aggressor mode reduces the pessimism for noise analysis by reducing the effect of composite aggressor.

For the current value of this variable, type **printvar si_noise_composite_aggr_mode**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018568-69

noise_composite_aggr_mode

Persistent parameter that controls whether a composite aggressor is used to model the combined noise of all small aggressors, including the filtered ones, when performing noise analysis using the PTSI-like model.

Syntax:

```
noise_composite_aggr_mode disabled | normal | statistical
```

where the values have the following meaning:

disabled	Do not use a composite aggressor to model the combined effect of all weak aggressors.
normal	Use a composite aggressor to model the combined effect of all weak aggressors and assume the worst possible aggressor alignment.
statistical	Use a composite aggressor to model the combined effect of all weak aggressors, and use a statistical model. Compared to the *normal* mode, this reduces the pessimism.

The default value is *disabled*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793844

si_use_driving_cell_derate_for_delta_delay

Allows crosstalk delta delay for one net to be derated using the relevant derate factor for the cell driving that net.

TYPE

boolean

DEFAULT

FALSE

GROUP

si_variables

DESCRIPTION

When this variable is set to *true* the crosstalk delta delays for each net will be derated using the derate factors from the cell driving that net.

The relevant derate factor to be applied will adhere to the same precedence rules as the driving cell itself. For example, if no instance-specific derate factor was set on the driving cell then the hierarchical cell, the library cell and finally the global derate factors will be checked for a relevant derate factor.

To see what derate factors are to be applied to the net in question, first obtain the driving cell ($driving_cell) and use: pt_shell **report_timing_derate [get_cells $driving_cell]**

If the command **report_timing** is invoked with the **-derate** option then the un-derated crosstalk delta delay will be reported as before. In addition the derate column will report the net derate factor used to derate the delta-free net delay.

To determine the current value of this variable, enter the following command: pt_shell **printvar si_use_driving_cell_derate_for_delta_delay** or pt_shell **echo $si_use_driving_cell_derate_for_delta_delay**

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018520-21

si_use_driving_cell_derate_for_delta_delay

Persistent parameter that controls whether crosstalk delta delay on a victim net will be derated using the same derate factor for the cell driving that net.

Syntax:

```
si_use_driving_cell_derate_for_delta_delay true | false
```

where the values have the following meaning:

`true`	Use the derating factor for the delay on a victim net also as a derating factor when calculating the delta delay on a victim net caused by any aggressor net.
`false`	Do not use the derating factor on a victim net when calculating the delta delay that an aggressor causes on a victim net.

The default value is *false*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793847

si_xtalk_composite_aggr_mode

Specifies the composite aggressor mode for crosstalk delay.

TYPE

String

DEFAULT

disabled

DESCRIPTION

This variable specifies which composite aggressor mode is used in PrimeTime SI crosstalk delay analysis. Allowed values are *disabled* (the default), which turns off the composite aggressor feature. *normal*, causes PrimeTime SI to calculate crosstalk delay by utilizing the non-statistical composite aggressor feature. Selecting *statistical* causes PrimeTime SI to calculate crosstalk by using the statistical composite aggressor flow.

In *disabled* composite aggressor mode, PrimeTime SI uses its original flow with composite aggressor completely off to calculate the xtalk delay.

In *normal* composite aggressor mode, PrimeTime SI aggregates the effect of some small aggressors (including filtered ones) into a single composite aggressor, thereby reducing the computational complexity and improving the performance.

statistical composite aggressor mode reduces the pessimism for xtalk delay analysis by reducing the effect of composite aggressor.

For the current value of this variable, type **printvar si_xtalk_composite_aggr_mode**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018573

xtalk_composite_aggr_mode

Persistent parameter that controls whether a composite aggressor is used to model the combined impact of all small aggressors, including the filtered ones, when performing crosstalk delay analysis using the PTSI-like model.

Syntax:

```
xtalk_composite_aggr_mode disabled | normal | statistical
```

where the values have the following meaning:

disabled	Do not use a composite aggressor to model the combined effect of all weak aggressors.
normal	Use a composite aggressor to model the combined effect of all weak aggressors and assume the worst possible aggressor alignment.
statistical	Use a composite aggressor to model the combined effect of all weak aggressors, and use a statistical model. Compared to the *normal* mode, this reduces the pessimism.

The default value is *disabled*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793848

si_xtalk_composite_aggr_noise_peak_ratio

Used to control the composite aggressor selection for xtalk analysis.

TYPE

float

DEFAULT

0.01

DESCRIPTION

Specifies the threshold value in crosstalk bump to Vdd ratio, below which aggressors are selected into composite aggressor group. The default value is 0.01, which means all the aggressor nets with crosstalk bump to Vdd ratio less than 0.01 will be selected into composite aggressor group. It works together with other filtering thresholds **si_filter_per_aggr_noise_peak_ratio** and **si_filter_accum_aggr_noise_peak_ratio** to determine which aggressors can be selected into composite aggressor group.

To determine the current value of this variable, type **printvar si_xtalk_composite_aggr_noise_peak_ratio** at the PT shell prompt.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018573-74

xtalk_composite_aggr_noise_peak_ratio

Persistent parameter that specifies the noise peak threshold of an individual aggressor. If the noise peak is above the threshold, the noise impact of the aggressor is calculated individually. If the noise peak is below the threshold, the aggressor is considered weak and its noise impact is modeled by the composite aggressor.

Syntax:

`xtalk_composite_aggr_noise_peak_ratio` *`ratio`*

where *`ratio`* is the threshold noise voltage as a fraction of the supply below which the impact of the aggressor is modeled as part of the composite aggressor.

The default value is *1.000e-02*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793849

si_xtalk_composite_aggr_quantile_high_pct

Used to control the composite aggressor creation for statistical analysis.

TYPE

float

DEFAULT

99.73

DESCRIPTION

This variable is set to the desired probability in percentage format that any given real combined bump height will be less than or equal to the computed composite aggressor bump height. Given the desired probability, the resulting quantile value for the composite aggressor bump height will be calculated.

The default value of this variable is 99.73, which corresponds to a 3-sigma probability that the real bump height from any randomly-chosen combination of aggressors will be covered by the composite aggressor bump height.

To determine the current value of this variable, type **printvar si_xtalk_composite_aggr_quantile_high_pct** at the PT shell prompt.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018574

xtalk_composite_aggr_quantile_high_pct

Persistent parameter that specifies the required confidence level to use for the statistical model of the composite aggressor. The value is the probability in percentage that a real noise bump will be smaller than the calculated bump.

This parameter is also used in the calculation of the virtual aggressor capacitance, if the Celtic-like noise model is chosen, the capacitance mode is chosen for the modelling of weak aggressors, and the binomial method is chosen to calculate that capacitance.

Syntax:

`xtalk_composite_aggr_quantile_high_pct` *`percentage`*

where *`percentage`* is the likelihood that a real bump is smaller than or equal to the calculated composite bump.

The default value is *99.73%*, which corresponds to a 3-sigma probability.

si_xtalk_delay_analysis_mode

Specifies the arrival window alignment mode for crosstalk delay.

TYPE

String

DEFAULT

all_paths

DESCRIPTION

This variable specifies how the alignment between victim & aggressors is performed in crosstalk delay analysis PrimeTime SI. Allowed values are *all_paths* (the default), which causes PrimeTime SI to calculate crosstalk for all paths through the victim net. *worst_path*, causes PrimeTime SI to calculate crosstalk for all the worst paths(the earliest/latest path) through the victim net. Selecting *all_violating_paths* causes PrimeTime SI to calculate crosstalk for all worst paths and paths with the negative slack.

In *all_paths* alignment mode, PrimeTime SI considers the largest possible crosstalk delta delay for the given victim & aggressor arrival windows. This gurantees that all paths going through the victim net with different arrival times are conservative. This is default and tradational way PTSI calculated delta delay. The limitation of this approach is worst crosstalk delta delay applied to all paths including the worst path which causes slack of the design to be pessimistic. When the path based analysis is done on a path using *get_recalculated_timing_paths* the above pessimism is removed for the specific path. However it is too expensive do path based analysis analysis on all the paths of the design.

In *worst_path* alignment mode, PrimeTime SI aligns aggressors for the the earlist/ latest paths on the victim, so only crosstalk affecting to these worst path is considered. So only the crosstalk affect that makes the slowest (earliest) path any slower(faster) is calculated. If the slowest/earliest path is a *set_false_path*, the true path is considerd. Considering the worst path instead of all paths, typically generates smaller delta delays and the worst paths and the design slack becomes less pessimistic. This approach makes sure that design slack & worst path are conservative.

There is a caveat to *worst_path* alignment mode, the crosstalk delay is applicable to worst path only, so the sub-critical path delay may be inaccurate. The side affect of this limitation *report_timing -nworst N*, N1 could report paths with optimistic slacks. Also *report_constraint -all_violators -max_delay -min_delay* will report less number of violations than really exist on the design. Also as violating critical paths is fixed, the optimistic sub-critical paths will be critical and start violating. Also bottleneck commands like *report_timing_bottleneck* and *report_si_bottleneck* will be less effective.

For some design flows the sub-critical path optimism is less of an issue if the design meets the timing constraints, i.e. all endpoints in the design show positive slacks. However, when the design has not met the timing yet, getting conservative crosstalk deltas for all the violating paths (whose slack is negative) is essential for the fixing flow. The alignment mode *all_violating_paths* addresses this by

xtalk_delay_analysis_mode

Persistent parameter that specifies the alignment that is assumed between victim and aggressors during crosstalk delay analysis.

Syntax:

```
xtalk_delay_analysis_mode all_paths | all_violating_paths | worst_path
```

where the values have the following meaning:

`all_paths`	Calculate crosstalk for all paths through the victim net. This is the most conservative approach.
`all_violating_paths`	Calculate crosstalk only for paths with a negative slack.
`worst_path`	Calculate crosstalk only for the worst paths through the net, that is, the earliest and latest path.

The default value is *all_paths*.

aligning the aggressors for all the violating and the worst path through any pin in the design. This means, all paths with negative slacks and also all the critical paths through any pin in the design (even if the slack for that worst path is positive) have a conservative slack. This mode may show more pessimism on worst paths than the worst_path mode and also the runtime might get slightly higher than *worst_path*. To reduce pessimism furthur, in the new modes *worst_path* and *all_violating_path*, another feature is enabled, where in the clock n/w seperate delay calculation is	

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793858

si_xtalk_exit_on_max_iteration_count

Specifies a maximum number of incremental timing iterations, after which PrimeTime-SI exits the analysis loop.

TYPE

integer

DEFAULT

2

DESCRIPTION

Specifies a maximum number of incremental timing iterations. PrimeTime-SI exits the analysis loop after performing this number of iterations.

The default value of this variable is 2, meaning that PrimeTime-SI exits the analysis loop after performing two iterations. You can override this default by setting the variable to another integer; the minimum allowed value is 1.

This variable is one of a set of six variables that determine exit criteria; PrimeTime-SI exits the analysis loop after completing the current iteration if one or more of the following is true:

1. The number of iterations performed equals the value of the **xtalk_max_iteration_count** variable.

2. All delta delays fall between the values of the **si_xtalk_exit_on_min_delta_delay** and **si_xtalk_exit_on_max_delta_delay** variables.

3. The number of nets selected for reevaluation in the next iteration is less than the value of the **si_xtalk_exit_on_number_of_reevaluated_nets** variable.

4. The percentage of nets (relative to the total number of nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_reevaluated_nets_pct** variable.

5. The percentage of nets (relative to the number of cross-coupled nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_coupled_reevaluated_nets_pct** variable.

6. You manually exit the analysis loop by pressing Control-C to send an interrupt signal to the PrimeTime process. The interrupt is handled as any other exit criteria, at the end of the current iteration of the crosstalk analysis. You cannot interrupt iteration immediately without exiting PrimeTime.

To determine the current value of this variable, type **printvar si_xtalk_exit_on_max_iteration_count**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018575-76

xtalk_exit_on_max_iteration_count

Persistent parameter that specifies the maximum number of iterations allowed between total delay calculation and crosstalk-induced delay calculation. If there is no convergence after the specified maximum number of iterations, Aprisa fails and issues an error.

Syntax:

`xtalk_exit_on_max_iteration_count` *`maxcount`*

where *`maxcount`* is the maximum number of iterations allowed.

The default value is *2*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793870

si_xtalk_reselect_delta_delay

Specifies the threshold of net delay change caused by crosstalk analysis, above which PrimeTime-SI reselects the net for subsequent delay calculations.

TYPE

float

DEFAULT

5

DESCRIPTION

This variable specifies a reselection threshold in terms of absolute delta delay. Nets that have at least one net arc with a crosstalk-annotated delta delay above this threshold are selected for the next iteration of PrimeTime-SI delay calculations. Note that delta delays are annotated on net arcs, but they capture the change of stage delay (cell plus net) because of crosstalk.

This variable is one of a set of four variables that determine net reselection criteria. The other three variables are as follows:

si_xtalk_reselect_delta_delay_ratio
si_xtalk_reselect_max_mode_slack
si_xtalk_reselect_min_mode_slack

All four variables are ignored if the variable **si_xtalk_reselect_critical_path** is true.

To determine the current value of this variable, type **printvar si_xtalk_reselect_delta_delay**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018577

xtalk_reselect_delta_delay

Persistent parameter that specifies the threshold delta delay in ns above which the net is reselected for window filtering, provided reselection is enabled with the xtalk_reselect_delta_and_slack parameter.

Syntax:

`xtalk_reselect_delta_delay` *deltaT*

where *deltaT* is the threshold delta delay above which the net is reselected.

The default value is *5.000*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793871

si_xtalk_reselect_delta_delay_ratio

Specifies the threshold of the ratio of net delay change caused by crosstalk analysis to the total stage delay, above which PrimeTime-SI reselects a net for subsequent delay calculations.

TYPE

float

DEFAULT

0.95

DESCRIPTION

This variable specifies a reselection threshold in terms of the delta delay ratio. Nets that have at least one net arc with a crosstalk-annotated delta delay, where the ratio of the annotated delta to the stage delay is above this threshold, are selected for the next iteration of PrimeTime-SI delay calculations.

If a net has multiple stage delays (because of a net fanout greater than one or multiple cell arcs), PrimeTime-SI considers the stage delta delay and stage delay that result in higher delta to stage delay ratio, thus making reselection conservative.

This variable is one of a set of four variables that determine net reselection criteria. The other three variables are as follows:

si_xtalk_reselect_delta_delay
si_xtalk_reselect_max_mode_slack
si_xtalk_reselect_min_mode_slack

All four variables are ignored if the variable **si_xtalk_reselect_critical_path** is true.

To determine the current value of this variable, type **printvar si_xtalk_reselect_delta_delay_ratio**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018577-78

xtalk_reselect_delta_delay_ratio

Persistent parameter that specifies the threshold delta delay as a fraction of the stage delay above which the net is reselected for window filtering, provided reselection is enabled with the xtalk_reselect_delta_and_slack parameter.

Syntax:

`xtalk_reselect_delta_delay_ratio` *deltaT*

where *deltaT* is the fraction of stage delay above which the net is reselected.

The default value is *0.950*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793872

si_xtalk_reselect_max_mode_slack

Specifies the max mode pin slack threshold, below which PrimeTime-SI reselects a net for subsequent delay calculations.

TYPE

float

DEFAULT

0

DESCRIPTION

This variable specifies the pin slack threshold in the max mode. Nets that have at least one pin with a max mode slack below this threshold are selected for the next iteration of PrimeTime-SI delay calculations. Max-mode pin slack is the slack of the worst max-mode (setup) path through the pin.

This variable is one of a set of four variables that determine net reselection criteria. The other three variables are as follows:

si_xtalk_reselect_delta_delay
si_xtalk_reselect_delta_delay_ratio
si_xtalk_reselect_min_mode_slack

All four variables are ignored if the variable **si_xtalk_reselect_critical_path** is true.

To determine the current value of this variable, type **printvar si_xtalk_reselect_max_mode_slack**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018578

xtalk_reselect_max_mode_slack

Persistent parameter that specifies the threshold delta delay as a fraction of the stage delay above which the net is reselected for window filtering, provided reselection is enabled with the xtalk_reselect_delta_and_slack parameter.

Syntax:

`xtalk_reselect_max_mode_slack` *`setupSlack`*

where *`setupSlack`* is the threshold slack below which a net is reselected.

The default value is *0.000*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793873

si_xtalk_reselect_min_mode_slack

Specifies the min mode pin slack threshold, below which PrimeTime-SI reselects a net for subsequent delay calculations.

TYPE

float

DEFAULT

0

DESCRIPTION

This variable specifies the pin slack threshold in the min mode. Nets that have at least one pin with a min mode slack below this threshold are selected for the next iteration of PrimeTime-SI delay calculations. Min-mode pin slack is the slack of the worst min-mode (hold) path through the pin.

This variable is one of a set of four variables that determine net reselection criteria. The other three variables are as follows:

si_xtalk_reselect_delta_delay
si_xtalk_reselect_delta_delay_ratio
si_xtalk_reselect_max_mode_slack

All four variables are ignored if the variable **si_xtalk_reselect_critical_path** is true.

To determine the current value of this variable, type **printvar si_xtalk_reselect_min_mode_slack**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018579

xtalk_reselect_min_mode_slack

Persistent parameter that specifies the threshold delta delay as a fraction of the stage delay above which the net is reselected for window filtering, provided reselection is enabled with the xtalk_reselect_delta_and_slack parameter.

Syntax:

`xtalk_reselect_min_mode_slack` *holdSlack*

where *holdSlack* is the threshold slack below which a net is reselected.

The default value is *0.000*.

Pl's Ex. 36 (PrimeTime D-2010.06) at SYNPS-00010150-51

timing_aocvm_analysis_mode

Configure an AOCVM analysis.

TYPE

string

DEFAULT

""

GROUP

timing_variables

DESCRIPTION

When this variable is set to "", the default AOCVM analysis is performed. The default analysis is defined as follows.

Depth is used to index the random component of variation in an AOCVM derate table. Depth is defined as the number of cell (or net) delay timing arcs in a path from the path common point. Separate depth values are calculated for cells and nets. Separate depth values are calculated for launch and capture paths. Both clock and data networks objects are included in the depth computation. Random coefficients affect the depth computation; for more information see the **set_aocvm_coefficient** manpage.

Distance is used to index the systematic component of variation in an AOCVM derate table. Distance is defined as the length of the diagonal of the bounding box enclosing the cell (or net) delay timing arcs in a path from the path common point. Separate bounding boxes and distance values are calculated for cells and nets. Both clock and data networks objects are included in the bounding box. The cell at the path endpoint is included in the cell bounding box. Only nodes and terminals of the network along the net arc are included in the net bounding box.

The **timing_aocvm_analysis_mode** variable is used to configure an AOCVM analysis. Choose from the following analysis modes, which can all be combined except for the modes combined_launch_capture_depth and separate_data_and_clock_metrics (only one of the 2 modes can be specified due to their inherently contradictory flows):

- *clock_network_only*
- *combined_launch_capture_depth*
- *separate_data_and_clock_metrics*
- *single_path_metrics*

To configure an AOCVM analysis, specify the analysis mode(s) required in the

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018524-25

timing_aocvm_analysis_mode

Persistent parameter that controls how the guardband derating factors specified with the *-late* or *-early* arguments should be combined with the distance- and logic-depth-dependent derating factor you defined with the *read_aocvm* Tcl command.

These methods will be illustrated with the following example, based in the following assumptions:

- The guardband derating was specified as follows:
  `set_timing_derate -aocvm_guardband -late 1.1 -early 0.9`
- A proper on-chip-variation file was read in with the *read_aocvm* Tcl command.
- For a particular delay, using this on-chip-variation file, the aocv late delay derating factor is *1.2*, and the early delay derating factor is *0.8*.

The following three methods are provided to combine the guardband factors with the aocv factors:

- *corrolated_components*—In this method, the marginal derating factor is the sum of the marginal guardband derating factor and the marginal aocv derating factor.
  Late derating = 1 + (1.1-1) + (1.2-1) = 1.3
  Early derating = 1 + (0.9-1) + (0.8-1) = 0.7
- *convoluted_components*—In this method, the total derating factor is the product of the guardband derating factor and the aocv derating factor.
  Late derating = 1.1 * 1.2 = 1.32
  Early derating = 0.8 * 0.9 = 0.72
- normal—In this method, the marginal derating factor is the root-mean-square of the marginal guardband derating factor and the marginal aocv derating factor.
  Late derating = 1 + sqrt((1.1-1)**2 + (1.2-1)**2) = 1.22
  Early derating = 1 + sqrt((0.9-1)**2 + (0.8-1)**2) = 0.78

Syntax:

```
timing_aocvm_analysis_mode \
   correlated_components | convoluted_components | normal
```

where the values have the following meaning:

`correlated_components`	Add the marginal guardband derating factor to the marginal aocv derating factor to determine the total marginal derating factor.
`convoluted_components`	Multiply the guardband with the aocv derating factor.
`normal`	Use the root-mean-square of the marginal derating factors to determine the total marginal derating factor.

The default value is *normal*.

timing_aocvm_analysis_mode variable. For example:

```
pt_shell> set timing_aocvm_analysis_mode "clock_network_only"
```

To determine the current value of this variable, enter the following command:

```
pt_shell> printvar timing_aocvm_analysis_mode
```

The effect that of each of the AOCVM analysis modes has on the default analysis is described below in detail:

clock_network_only When this option is not specified (default), AOCVM derating is applied throughout the design and constant (OCV) derating is ignored.

When this option is specified, AOCVM derating is applied to arc delays in the clock network only. Clock network AOCVM depth and distance metrics are calculated based on clock network topology only. The data network receives constant (OCV) derating, if constant derates have been annotated for data network objects; otherwise they are not derated.

In the *clock_network_only* delay timing arcs in the data network are excluded from depth and distance calculations. The cell at the path endpoint is still included in the cell bounding box.

combined_launch_capture_depth In this mode, launch and capture depths are not calculated separately. The launch and capture paths are considered together and a combined depth is calculated for the entire path. This option cannot be specified together with the separate_data_and_clock_metrics mode.

separate_data_and_clock_metrics In this mode, separate depths are computed for the clock and data paths and the appropriate AOCV derates based on the separate depths are used for derating delays. This option cannot be specified together with the combined_launch_capture_depth mode.

single_path_metrics In this mode, a single set of path metrics is calculated for both cell and net objects. Separate path metrics are *not* calculated for cells and nets in a path. This behaviour is backwardly compatible with the legacy Tcl-based "LOCV" solution.

Distance is measured by computing the diagonal of a bounding box around all of the cells in a path. Ports in the path and the common point are also included. This distance is used to lookup the systematic component of variation for both cells and nets.

Depth is measured considering only the cells in a path. This depth is used to lookup the random component of variation for both cells and nets.

Pl's Ex. 36 (PrimeTime D-2010.06) at SYNPS-00010152

timing_aocvm_enable_analysis

Enable PrimeTime's graph-based AOCVM analysis.

TYPE

Boolean

DEFAULT

false

GROUP

timing_variables

DESCRIPTION

When *false* (default) the graph-based AOCVM timing update is not performed. A path-based aocvm analysis can be performed in this mode using the **-pba_mode** option on the **report_timing** and **get_timing_paths** commands. In this mode constant timing derates specified using the **set_timing_derate command are required to pessimistically bound the analysis. You should specify constant derates that do not clip the range of the path-based AOCVM derates to avoid optimism.**

When *true* the graph-based aocvm timing update is performed as part of the **update_timing** command. A path-based aocvm analysis can also be performed in this mode. In this mode constant timing derates are not required and, in fact, constant derates for *static delays* are ignored. Graph-based AOCVM derates computed during **update_timing** tightly bound the path-based AOCVM derates without clipping their range. Note that setting this variable to *true* will automatically switch the design into *on_chip_variation* analysis mode using the **set_operating_conditions** command.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018527

timing_aocvm_enable_analysis

Persistent parameter that controls whether to enable advanced on-chip variation analysis. This advanced analysis uses knowledge about geographical distance and logic depth to better calculate the impact of systematic and random variations within one chip.

Use the *read_aocvm* Tcl command to specify the two-dimensional distance- and logic-depth-dependent derating models.

Use the *set_timing_derate -aocvm_guardband* Tcl command to specify an additional guardband derating factor to use for either early or late paths.

Use the *timing_aocvm_analysis_mode* Tcl variable to control how to combine these early and late guardbands with the advanced OCV derating factor.

- If this variable is not set, then these derating factors are considered independent, and the total effect of these derating factors is the square root of the sum of the squares of the effects of derating.
- If the variable is set with the value *correlated_components*, then the derating factors are considered fully correlated and the total effect of deration is the sum of the effects of each derating factor.

Use the *report_timing* Tcl command with the *-aocvm* argument to see the resulting derating factors that were applied to net and cell delay models in the reported timing paths.

Syntax:

```
timing_aocvm_enable_analysis true | false
```

where the values have the following meaning:

`true`	Enable advanced on-chip variation analysis.
`false`	Disable advanced on-chip variation analysis.

The default value is *false*.

timing_clock_reconvergence_pessimism

Select signal transition sense matching for computing clock reconvergence pessimism removal.

TYPE

string

DEFAULT

normal

DESCRIPTION

Determines how the value of the clock reconvergence pessimism removal (crpr) is computed with respect to transition sense. Allowed values are *normal* (the default) and *same_transition*.

When set to *normal*, the crpr value is computed even if the clock transitions to the source and destination latches are in different directions on the common clock path. It is computed separately for rise and fall transitions and the value with smaller absolute value is used.

When set to *same_transition*, the crpr value is computed only when the clock transition to the source and destination latches have a common path and the transition is in the same direction on each pin of the common path. Thus if the source and destination latches are triggered by different edge types, crpr has a value of zero.

To determine the current value of this variable, type **printvar timing_clock_reconvergence_pessimism** or **echo $timing_clock_reconvergence_pessimism**.

timing_clock_reconvergence_pessimism

Persistent parameter that controls when clock reconvergence pessimism removal (CRPR) is applied.

There are two cases:

- Both launch and capture clocks are active on the same transition of the clock at the end of the common path.
- The launch and capture clocks are active on different transitions of the clock at the end of the common path.

Use this parameter to control whether pessimism is removed for both cases, or if it is only removed in case both clocks are triggered from the same clock transition in the common path

This parameter only applies when CRPR is enabled, that is, when the *ta* parameter *timing_remove_clock_reconvergence_pessimism* is set.

The calculated pessimism is the difference between the maximum and minimum delay on the clock path that is common between the launching and the capturing clock.

You can further scale down the applied pessimism for these two cases as follows:

- Set the *ta* parameter, timing_crpr_same_trans_scaling_factor, to a number between *0* and *1* to reduce the CRPR, when the launch and capture clocks are triggered from the same common clock transition.
- Set the *ta* parameter, *timing_crpr_scaling_factor*, to a number between 0 and 1 to reduce the CRPR when the launch and capture clocks are triggered from different common clock transitions.

Syntax:

```
timing_clock_reconvergence_pessimism normal | same_transition
```

where the values have the following meaning:

`normal`

Apply common-reconvergence pessimism (CRPR). Pessimism will be removed for both of the following cases:

1. Both launch and capture clocks are active on the same transition of the clock at the end of the common path.
2. The launch and capture clocks are active on different transitions of the clock at the end of the common path.

You can scale down the applied pessimism as follows:

- Set the *ta* parameter, timing_crpr_same_trans_scaling_factor, to a number between *0* and *1* to reduce the CRPR, for the case the launch and capture clocks are triggered from the clock transition.
- Set the *ta* parameter, *timing_crpr_scaling_factor*, to a number between 0 and 1 to reduce the CRPR, in case the launch and capture clock edges are in a different phase.

`same_transition`

Only apply common-reconvergence pessimism (CRPR) when both launch and capture clocks are active on the same transition of the clock at the end of the common path. This pessimism is the difference between the maximum and minimum delay on the clock path that is common between the launching and the capturing clock.

Set the *ta* parameter, timing_crpr_same_trans_scaling_factor, to a number between *0* and *1* to scale the CRPR, for the case the launch and capture clocks are triggered from the clock transition.

The default value is *normal*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793888

timing_crpr_remove_clock_to_data_crp

Allows the removal of Clock Reconvergence Pessimism (CRP) from paths that fan out directly from clock source to the data pins of sequential devices.

TYPE

boolean

DEFAULT

FALSE

DESCRIPTION

When this variable is set to *true* then CRP will be removed for all paths that fan out directly from clock source pins to the data pins of sequential devices.

It should be noted that when this variable is set to *true* all sequential devices that reside in the fanout of clock source pins must be handled seperately in the subsequent timing update. This may cause a severe performance degradation to the timing update.

pt_shell **echo $timing_crpr_remove_clock_to_data_crp**
or
pt_shell **printvar timing_crpr_remove_clock_to_data_crp**

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018532

timing_crpr_remove_clock_to_data_crp

Persistent parameter that controls whether to remove Clock-to-data Reconvergence Pessimism (CRP) during timing analysis.

By default, arrival time uncertainty of the clock signal and data signal are handled independent of each other. This approach may be overly pessimistic for cases where the clock path and data path share some logic. To reduce that pessimism, and to obtain a more accurate analysis, you may want to enable this parameter.

Syntax:

```
timing_crpr_remove_clock_to_data_crp true | false
```

where the values have the following meaning:

`true`	Remove clock-to-data CRP. This setting causes Aprisa to consume a lot more memory, and to run for a much longer time.
`false`	Do not remove clock-to-data CRP. This is much more efficient, but may be overly pessimistic.

The default value is *false*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793889

timing_crpr_threshold_ps

Specifies amount of pessimism that clock reconvergence pessimism removal (CRPR) is allowed to leave in the report.

TYPE

float

DEFAULT

20

DESCRIPTION

Specifies amount of pessimism that clock reconvergence pessimism removal (CRPR) is allowed to leave in the report. The unit is in pico seconds (ps), regardless of the units of the main library.

The threshold is per reported slack: setting the this variable to the *TH1* value means that reported slack is no worse than *S - TH1*, where *S* is the reported slack when **timing_crpr_threshold_ps** is set close to zero (the minimum allowed value is **1** picosecond).

The variable has no effect if CRPR is not active (**timing_remove_clock_reconvergence_pessimism** is false). The larger the value of **timing_crpr_threshold_ps**, the faster the runtime when CRPR is active. The recommended setting is about one half of the stage (gate plus net) delay of a typical stage in the clock network. It provides a reasonable trade-off between accuracy and runtime in most cases. You may want to use different settings throughout the design cycle: larger during the design phase, smaller for sign-off. You might have to experiment and set a different value when moving to a different technology.

To determine the current value of this variable, type **printvar timing_crpr_threshold_ps**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018534-35

timing_crpr_threshold_ps

Persistent parameter that specifies the amount of pessimism that Clock Reconvergence Pessimism Removal (CRPR) is allowed to leave in the report. So, this parameter effectively controls how much effort CRPR should spend on a reconvergent path.

Syntax:

`timing_crpr_threshold_ps` *threshold*

where *threshold* is a real number greater than *1.0*, defining the maximum amount of pessimism that can be left in a reconvergent path after CRPR.

The default value is *20.000*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793890

timing_disable_bus_contention_check

Disable checking for timing violations resulting from transient contention on design busses.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

Applies only to bus designs that have multiple three-state drivers.

When *true*, PrimeTime ignores timing setup and hold (max and min) violations that occur as a result of transient bus contention. When *false* (the default), PrimeTime reports these timing violations.

Bus contention occurs when more than one driver is enabled at the same time. By default, PrimeTime treats the bus as if it is in an unknown state during this region of contention, and reports a timing violation if the setup and hold regions extend into the contention region. Note that checking is done only for timing violations, and not for logical and excessive power dissipation violations, which are outside the scope of static timing analysis tools.

Set this variable to *true* only if you are certain that transient bus contention regions will never occur. By setting the value to *true*, you guarantee that on a multi-driven three-state bus, the drivers in the previous clock cycle are disabled before the drivers in the current clock cycle are enabled. If you set this variable to *true*, you must ensure that the variable **timing_disable_bus_contention_check** is *false*. The variables **timing_disable_bus_contention_check** and **timing_disable_floating_bus_check** cannot both be *true* at the same time.

During the switching between the high-impedance (Z) state and the high/low state, the timing behavior (for example, intrinsic delay) of three-state buffers is captured in the Synopsys library using the timing arc types three_state_disable and three_state_enable. These timing arcs connect the enable pin to the output pin of the three-state buffers. For details, see the *Library Compiler Reference Manual*.

To determine the current value of this variable, type **printvar timing_disable_bus_contention_checks** or **echo $timing_disable_bus_contention_checks**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018535

timing_disable_bus_contention_check

Persistent parameter that controls whether the timing analyzer should perform bus contention checks.

Syntax:

```
timing_disable_bus_contention_check true | false
```

where the values have the following meaning:

`true`	Do not perform bus contention checks.
`false`	Perform bus contention checks.

The default value is *false*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793891

timing_disable_clock_gating_checks

Disable checking for setup and hold clock gating violations.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When *true*, disables clock-gating setup and hold checks. When *false* (the default), PrimeTime automatically determines clock-gating and performs clock-gating setup and hold checks.

To determine the current value of this variable, type **printvar timing_disable_clock_gating_checks** or **echo $timing_disable_clock_gating_checks**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018536

timing_disable_clock_gating_checks

Persistent parameter that controls whether to perform clock-gating setup and hold checks.

Syntax:

```
timing_disable_clock_gating_checks true | false
```

where the values have the following meaning:

true	Do not perform clock gating setup and hold checks.
false	Perform clock gating setup and hold checks.

The default value is *false*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793892

timing_disable_cond_default_arcs

Disable the default, non-conditional timing arc between pins that do have conditional arcs.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When *true*, disables nonconditional timing arcs between any pair of pins that have at least one conditional arc. When *false* (the default), these nonconditional timing arcs are not disabled. This variable is primarily intended to deal with the situation between two pins that have conditional arcs, where there is always a default timing arc with no condition.

Set this variable to *true* when the specified conditions cover all possible state-dependent delays, so that the default arc is useless. For example, consider a 2-input XOR gate with inputs as A and B and with output as Z. If the delays between A and Z are specified with 2 arcs with respective conditions 'B' and 'B~", the default arc between A and Z is useless and should be disabled.

To determine the current value of this variable, type **printvar timing_disable_cond_default_arcs** or **echo $timing_disable_cond_default_arcs**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018537

timing_disable_cond_default_arcs

Persistent parameter that controls whether the default condition of any conditional timing arc should be ignored.

Syntax:

```
timing_disable_cond_default_arcs true | false
```

where the values have the following meaning:

`true`	Do not examine the default condition of conditional timing arcs.
`false`	Do consider the default condition as any other condition of conditional timing arcs.

The default value is *false*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793893

timing_disable_floating_bus_check

Disable checking for timing violations resulting from transient floating design buses.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

Applies only to bus designs that have multiple three-state drivers.

When *true*, PrimeTime ignores timing setup and hold (max and min) violations that occur as a result of transient floating buses. When *false* (the default), PrimeTime reports these timing violations.

Floating bus condition occurs when no driver controls the bus at a given time. By default, PrimeTime treats the bus as if it is in an unknown state during this region of contention, and reports a timing violation if the setup and hold regions extend into the floating region. Note that checking is done only for timing violations, and not for logical violations, which are outside the scope of static timing analysis tools.

Set this value to *true* only if you are certain that transient floating bus regions will never occur. By setting the value to *true*, you guarantee that on a multi-driven three-state bus, the drivers in the previous clock cycle are disabled before the new drivers in the current clock cycle are enabled. If you set this variable to *true*, you must ensure that the variable **timing_disable_bus_contention_check** is *false*. The variables **timing_disable_floating_bus_check** and **timing_disable_bus_contention_check** cannot both be *true* at the same time.

During the switching between the high-impedance (Z) state and the high/low state, the timing behavior (for example, intrinsic delay) of three-state buffers is captured in the Synopsys library using the timing arc types three_state_disable and three_state_enable. These timing arcs connect the enable pin to the output pin of the three-state buffers. For details, see the *Library Compiler Reference Manual*.

To determine the current value of this variable, type **printvar timing_disable_floating_bus_check** or **echo $timing_disable_floating_bus_check**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018538

timing_disable_floating_bus_check

Persistent parameter that controls whether to check for floating busses.

Syntax:

```
timing_disable_floating_bus_check true | false
```

where the values have the following meaning:

`true`	Do not check for floating busses.
`false`	Check for floating busses.

The default value is *false*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1973894

timing_disable_internal_inout_cell_paths

Enable bidirectional feedback paths within a cell.

TYPE

Boolean

DEFAULT

true

DESCRIPTION

When *true* (the default), PrimeTime automatically disables bidirectional feedback paths in a cell. When *false*, bidirectional feedback paths in cells are enabled.

This variable has no effect on timing of bidirectional feedback paths that involve more than one cell (that is, if nets are involved); these feedback paths are controlled by the variable **timing_disable_internal_inout_net_arcs**.

To determine the current value of this variable, type **printvar timing_disable_internal_inout_cell_paths** or **echo $timing_disable_internal_inout_cell_paths**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018540

timing_disable_internal_inout_cell_paths

Persistent parameter that controls whether to analyze feedback paths inside a cell through a bidirectional pin of that cell.

Syntax:

```
timing_disable_internal_inout_cell_paths true | false
```

where the values have the following meaning:

`true`	Do not analyze timing paths inside a cell through a bidirectional pin of that cell.
`false`	Also analyze timing paths inside a cell through a bidirectional pin of that cell.

The default value is *true*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793895

timing_disable_internal_inout_net_arcs

Controls whether bidirectional feedback paths across nets are disabled or not.

TYPE

Boolean

DEFAULT

true

DESCRIPTION

When *true* (the default), PrimeTime automatically disables bidirectional feedback paths that involve more than one cell; no path segmentation is required. When *false*, these bidirectional feedback paths are enabled.

This variable has no effect on timing of bidirectional feedback paths that are completely contained in one cell (that is, if nets are not involved); these feedback paths are controlled by the variable **timing_disable_internal_inout_cell_paths**.

To determine the current value of this variable, type **printvar timing_disable_internal_inout_net_arcs** or **echo $timing_disable_internal_inout_net_arcs**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018540-41

timing_disable_internal_inout_net_arcs

Persistent parameter that controls whether to analyze feedback paths through a bidirectional pin.

Syntax:

```
timing_disable_internal_inout_net_arcs true | false
```

where the values have the following meaning:

`true`	Ignore feedback paths through bidirectional pins.
`false`	Analyze feedback paths through bidirectional pins.

The default value is *false*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793896

timing_disable_recovery_removal_checks

Disable or enable the timing analysis of recovery and removal checks in the design.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When *true*, disables recovery and removal timing analysis. When *false* (the default), PrimeTime performs recovery and removal checks; for descriptions of these checks, see the man page for the **report_constraint** command.

To determine the current value of this variable, type **printvar timing_disable_recovery_removal_checks** or **echo $timing_disable_recovery_removal_checks**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018541

timing_disable_recovery_removal_checks

Persistent parameter that controls whether to perform recovery and removal timing checks. Recovery and removal checks are similar to setup and hold checks, but are intended for asynchronous signals, such as set and reset signals.

Syntax:

```
timing_disable_recovery_removal_checks true | false
```

where the values have the following meaning:

true	Do not perform recovery and removal checks.
false	Perform recovery and removal checks.

The default value is *false*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793898

timing_early_launch_at_borrowing_latches

Removes clock latency pessimism from the launch times for paths which begin at the data pins of transparent latches.

TYPE

Boolean

DEFAULT

true

DESCRIPTION

In the following description we assume that the data paths of interest are setup paths since we refer specifically to time borrowing scenarios. However, if **timing_allow_short_path_borrowing** is enabled then the same discussion applies to borrowing hold paths too.

When a latch is in its transparent phase, data arriving at the D-pin passes through the element as though it were combinational. To model this scenario, whenever PrimeTime determines that time borrowing occurs at such a D-pin, paths which originate at the D-pin are created.

Sometimes there is a difference between the launching and capturing latch latencies, due either to reconvergent paths in the clock network or different min and max delays of cells in the clock network. For setup paths, PrimeTime uses the late value to launch and the early value to capture. This achieves the tightest constraint and avoids optimism. However, for paths starting from latch D-pins this is pessimistic since data simply passes through and thus does not even "see" the clock edge at the latch.

When this timing variable is set to *true* (the default), such pessimism is eliminated by using the early latch latency to launch such paths. Note that only paths which originate from a latch D-pin are affected. When the variable is set to *false*, late clock latency is used to launch all setup paths in the design.

It is recommended that the user avail of this form of pessimism removal since it does not cause the run-time of the analysis to increase. However, it is also advised that the user disable it when clock reconvergence pessimism removal (CRPR) is enabled (i.e. when **timing_remove_clock_reconvergence_pessimism** is *true*). CRPR may not be applied to paths which have been launched using an early latency or the results may be optimistic. Since CRPR is a more sophisticated and accurate means of pessimism removal, the user should disable **timing_early_launch_at_borrowing_latches** when CRPR is enabled so that CRPR applies to all paths in the design. In this mode, note that the D-pin launch time is not modified by the open edge CRP - since late launch latency is used at the path startpoint, to additionally add CRP would be pessimistic, representing a "double-counting" of early-late differences.

To determine the current value of this variable, type **printvar timing_early_launch_at_borrowing_latches** or **echo $timing_early_launch_at_borrowing_latches**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018542

timing_early_launch_at_borrowing_latches

Persistent parameter that controls the clock latency to use for latches in paths requiring time borrowing.

Syntax:

```
timing_early_launch_at_borrowing_latches true | false
```

where the values have the following meaning:

`true`	Use the early clock latency for the latch cell.
`false`	Use the late clock latency for the latch cell.

The default value is *true*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793906

timing_enable_preset_clear_arcs

Controls whether PrimeTime enables or disables preset and clear arcs.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When *true*, permanently enables asynchronous preset and clear timing arcs, so that you use them to analyze timing paths. When *false* (the default), PrimeTime disables all preset and clear timing arcs.

Note that if there are any minimum pulse width checks defined on asynchronous preset and clear pins they are performed regardless of the value of this variable. Also note the the -true and the -justify options of report_timing cannot be used unless this variable is at its default value.

To determine the current value of this variable, type **printvar timing_enable_preset_clear_arcs**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018019-20

timing_enable_preset_clear_arcs

Persistent parameter that controls whether to analyze preset and clear timing constraints.

Syntax:

```
timing_enable_preset_clear_arcs true | false
```

where the values have the following meaning:

`true`	Take into account preset and clear timing constraints.
`false`	Ignore preset and clear timing constraints.

The default value is *false*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793911

timing_input_port_default_clock

Determines whether a default clock is assumed at input ports for which the user has not defined a clock with set_input_delay.

TYPE

Boolean

DEFAULT

true

DESCRIPTION

This Boolean variable affects the behavior of PrimeTime when the user sets an input delay without a clock on an input port. When *true* (the default value), the input delay on the port is set with respect to one imaginary clock so that the inputs are constrained. This also causes the clocks along the paths driven by these input ports to become related. When *false*, no such imaginary clock is assumed.

To determine the current value of this variable, type **printvar timing_input_port_default_clock**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018544

timing_input_port_default_clock

Persistent parameter that controls whether to assign a default clock to an input port if no input delay is specified on that port.

Syntax:

```
timing_input_port_default_clock true | false
```

where the values have the following meaning:

`true`	Assign a default clock if no input delay is specified.
`false`	Do not assign a default clock if no input delay is specified.

The default value is *false*.

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793915

timing_propagate_interclock_uncertainty

Enables or disables the propagation of interclock uncertainty through transparent latches in PrimeTime.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When false (the default), the interclock uncertainty is calculated for each latch-to-latch path independently, from the clock at the launch latch to the clock at the capture latch, even when latches operate in transparent mode.

When true, clock uncertainty information is propagated through each latch operating in transparent mode, as though it were a combinational element. This allows an entire sequence of latch-to-latch stages to be considered a single path for interclock uncertainty calculation, provided that time borrowing occurs at the endpoint of each intermediate stage.

Operating with this variable set to true can lead to more accurate results for designs containing transparent latches, at the cost of some CPU time and memory resources. To illustrate, consider a pipeline containing latches A, B, and C, clocked by clocks 1, 2, and 3, respectively. PrimeTime treats the paths between A and B and between B and C as distinct. In reality, however, if latch B is in transparent mode, data passes through it as though it were a combinational element. Regardless of whether interclock uncertainty has been applied between clocks 1 and 3, the default behavior is to apply the uncertainty between clocks 2 and 3 when calculating slack at latch C. It is more accurate, however, to apply the uncertainty between the clock at the path startpoint (clock 1, latch A) and the clock at the path endpoint (clock 3, latch C), if defined.

With **timing_propagate_interclock_uncertainty** set to true, the correct interclock uncertainty is applied, as though the path from latch A to latch C through the transparent latch B were a single, extended path. That is, the uncertainty is propagated through the transparent latch. To find out the startpoint of this extended path, use **report_timing -trace_latch_borrow**.

To determine the current value of this variable, type **printvar timing_propagate_interclock_uncertainty** or **echo $timing_propagate_interclock_uncertainty**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018546

timing_propagate_interclock_uncertainty

Persistent parameter that controls how Aprisa's timing analyzer handles clock uncertainty when the circuit contains transparent latches.

By default, when this variable is *false*, inter-clock uncertainty is calculated for each latch-to-latch path independently. For example, assume a pipeline of three latches, *A*, *B*, and *C*, clocked with clocks *1*, 2, and 3, respectively. Even if latch *B* is in transparent mode, the timing analyzer treats the path from *A* to *B* and the path from *B* to *C* as independent paths, each with its own clock uncertainty.

When the variable is set *true*, the timing analyzer treats the extended path from *A* through *B* to *C* as a single path, and calculates clock uncertainty based on clocks *1* and 3. This approach is more accurate, but requires more run time and increases the memory foot print.

Syntax:

```
timing_propagate_interclock_uncertainty true | false
```

where the values have the following meaning:

`true`	Handle transparent latches when calculating clock uncertainty.
`false`	Do simple and fast clock uncertainty propagation that ignores latches in transparent mode.

The default value is *false*.

timing_remove_clock_reconvergence_pessimism

Enables or disables clock reconvergence pessimism removal

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When this variable is set to *true*, PrimeTime removes clock reconvergence pessimism from slack calculation and minimum pulse width checks. This variable replaces the following discontinued options:

-report_clock_reconvergence_pessimism
-remove_clock_reconvergence_pessimism

of the **report_timing**, **report_constraint**, and **get_timing_paths** commands.

Clock reconvergence pessimism (CRP) is a difference in delay along the common part of the launching and capturing clock paths. The most common causes of CRP are reconvergent paths in the clock network, and different min and max delay of cells in the clock network.

CRP is independently calculated for rise and fall clock paths. You can use the variable **timing_clock_reconvergence_pessimism** to control CRP calculation with respect to transition sense. In the case of the capturing device being a level-sensitive latch two CRP values will be caculated:
• crp_open, which is the CRP corresponding to the opening edge of the latch
• crp_close, which is the CRP corresponding to the closing edge of the latch The required time at the latch will be increased by the value of crp_open and hence reduce the amount of borrowing (if any) at the latch. Meanwhile, the maximum time borrow allowed at the latch is affected by shifting the closing edge by crp_close. For more details, see the *PrimeTime User Guide: Fundamentals*.

For a more detailed description of a CRP calulation, use the **report_crpr** command.

CRP is calculated differently for minimum pulse-width checks. It is given as the minimum of (maximum rise arrival time - minimum rise arrival time) and (maximum fall arrival time - minimum fall arrival time) at the pin where the check is being made.

If the variable **si_enable analysis** is set to *true* delays in the clock network may also include delta delays resulting from crosstalk interaction. Such delays are dynamic in nature, that is, they may vary from one clock cycle to the next, causing different delay variations (either speed-up or slow-down) on the same network, but during different clock cycles.

Starting with U-2003.03 release PrimeTime only considers SI delta delays as part of the CRP calculation if the type of timing check deployed derives its data from the same clock cycle.

In transparent-latch based designs, it is recommended that the variable **timing_early_launch_at_borrowing_latches** should be set to *false* when CRP removal (CRPR) is enabled. In this case, CRPR will apply even to paths whose startpoints are borrowing, leading to better pessimism reduction overall.

Any effective change in the value of the **timing_remove_clock_reconvergence_pessimism** variable causes full update_timing. You cannot perform one report_timing operation that considers CRP and one that does not without full update_timing in between.

timing_remove_clock_reconvergence_pessimism

Persistent parameter that controls whether the timing analyzer will remove clock reconvergence pessimism. This pessimism arises in the default timing analysis when, in presence of on-chip-variation (OCV) and reconvergent clocks, overly conservative clock skew is calculated because one clock path uses minimum delay and another path uses maximum delay for the same cell.

This pessimism can be removed but at a significant runtime penalty.

Use the *ta* parameter timing_clock_reconvergence_pessimism to select whether to apply the removal when both launching clock and capturing clock are in the same clock phase, that is, triggered by the same clock transition at the end of the common clock path, or when to apply the removal even if the launching and capturing clocks are driven from different clock transitions.

The calculated pessimism is the difference between the maximum and minimum delay of the clock path that is common between the launching clock and the capturing clock.. Use the the *ta* parameter, timing_crpr_same_trans_scaling_factor, to scale that pessimism when the launch and capture clocks are triggered from the same clock transition. Use the the *ta* parameter, timing_crpr_scaling_factor, to scale that pessimism when the launch and capture clocks are triggered from different clock transitions.

Syntax:

```
timing_remove_clock_reconvergence_pessimism true | false
```

where the values have the following meaning:

`true`	Remove clock reconvergence pessimism.
`false`	Do not remove clock reconvergence pessimism.

The default value is *false*.

For backward compatibility, the discontinued options will appear for the first few releases after they are obsoleted. However, if the design is not up to date at the time they are executed, they will only set timing_remove_clock_reconvergence_pessimism to *true*

If the design is up to date, then the command with the discontinued option fails. Since the discontinued command options only set timing_remove_clock_reconvergence_pessimism to *true*, the **-report_clock_reconvergence_pessimism** option behavior is not backward compatible. It causes slack to be removed prior to selecting the worst path. In other words, it behaves the same as the discontinued **-remove_clock_reconvergence_pessimism** option of the **report_timing**, **reyport_constraint**, and **get_timing_paths** commands. As soon as possible, update your scripts to set the **timing_remove_clock_reconvergence_pessimism** variable to *true* instead of using the discontinued options.

Limitations: CRPR does not support paths that fan out directly from clock source pins to the data pins of sequential devices. To enable support for such paths the variable **timing_crpr_remove_clock_to_data_crp** must be set to TRUE. CRPR does not support ideal clock latency set on pins or ports. CRPR does not support propagated clocks set on pins or ports as opposed to clock objects.

To turn CRP removal on:

```
   pt_shell set timing_remove_clock_reconvergence_pessimism TRUE
TRUE
pt_shell report_timing
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793817

si_ccs_aggressor_alignment_mode

Specifies aggressor alignment mode used in the CCS-based gate-level simulation engine.

TYPE

String

DEFAULT

lookahead

DESCRIPTION

Specifies aggressor alignment mode used in the CCS-based gate-level simulation engine. Valid values include **stage** and **lookahead**. The default value is **lookahead** which enables the lookahead alignment feature for the CCS-based gate-level simulation engine so that PrimeTime-SI will find the alignment which results to worst-case path delay. It is noted that such alignment may not correspond to worst-case stage delay.

For complete information about the difference between worst-case stage alignment and worst-case path alignment, see the PrimeTime-SI User Guide.

To determine the current value of this variable, type **printvar si_ccs_aggressor_alignment_mode**.

Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938

```
set_param si1 ccs_aggressor_alignment_mode lookahead
#       Values : lookahead stage
#       Usage  : alignment mode for aggressors in computing delay
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793838

si_noise_nmos_threshold_ratio

Specifies the technology threshold voltage for NMOS devices divided by the Vcc.

TYPE

float

DEFAULT

0.2

DESCRIPTION

Specifies the technology threshold voltage for NMOS devices divided by Vcc; the default is 0.2. This variable, along with **si_noise_pmos_threshold_ratio**, makes up a pair of variables used by PrimeTime-SI during the noise analysis phase, to determine the steady state resistance value of the drivers in absence of a noise library.

When noise library is not present for a cell or for a design, this activates the PrimeTime-SI steady state resistance estimation mode. In this mode, steady state resistance gets estimated based on the value of the PMOS and NMOS threshold voltage values as well as other information extracted from the delay and slew tables.

Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938

```
set_param si1 noise_nmos_threshold_ratio 0.200
#       Type    : float
#       Usage   : NMOS threshold for SI noise analysis
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793839

si_noise_pmos_threshold_ratio

Specifies the technology threshold voltage for PMOS devices divided by the Vcc.

TYPE

float

DEFAULT

0.2

DESCRIPTION

Specifies the technology threshold voltage for PMOS devices divided by Vcc; the default is 0.2. This variable, along with **si_noise_nmos_threshold_ratio**, makes up a pair of variables used by PrimeTime-SI during the noise analysis phase, to determine the steady state resistance value of the drivers in absence of a noise library.

When noise library is not present for a cell or for a design, this activates the PrimeTime-SI steady state resistance estimation mode. In this mode, steady state resistance gets estimated based on the value of the PMOS and NMOS threshold voltage values as well as other information extracted from the delay and slew tables.

Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938

```
set_param si1 noise_pmos_threshold_ratio 0.200
#       Type    : float
#       Usage   : PMOS threshold for SI noise analysis
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793854-55

si_xtalk_exit_on_coupled_reevaluated_nets_pct

Specifies a maximum percentage of nets selected for reevaluation relative to the total number of coupled nets, below which PrimeTime-SI exits the analysis loop.

TYPE

float

DEFAULT

0

DESCRIPTION

Specifies a maximum percentage of nets selected for reevaluation relative to the total number of coupled nets. PrimeTime-SI exits the analysis loop after completing the current iteration, when the percentage of nets selected for reevaluation in the next iteration is less than this number. The number of coupled nets is based on detailed parasitics as read in by **read_parasitics**. That is, crosstalk filtering does not impact the count of coupled nets for the purpose of this variable. The number of coupled nets counts all individual net segments in the same way that [get_nets -hierarchical *] counts all nets in the design.

This variable is one of a set of six variables that determine exit criteria; PrimeTime-SI exits the analysis loop after completing the current iteration if one or more of the following is true:

1. The number of iterations performed equals the value of the **xtalk_max_iteration_count** variable.

2. All delta delays fall between the values of the **si_xtalk_exit_on_min_delta_delay** and **si_xtalk_exit_on_max_delta_delay** variables.

3. The number of nets selected for reevaluation in the next iteration is less than the value of the **si_xtalk_exit_on_number_of_reevaluated_nets** variable. the analysis loop.

4. The percentage of nets (relative to the total number of nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_reevaluated_nets_pct** variable.

5. The percentage of nets (relative to the number of cross-coupled nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_coupled_reevaluated_nets_pct** variable.

6. You manually exit the analysis loop by pressing Control-C to send an interrupt signal to the PrimeTime process. The interrupt is handled as any other exit criteria, at the end of the current iteration of the crosstalk analysis. You cannot interrupt iteration immediately without exiting PrimeTime.

To determine the current value of this variable, type **printvar si_xtalk_exit_on_coupled_reevaluated_nets_pct**.

Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938

```
set_param si1 xtalk_exit_on_coupled_reevaluated_nets_pct 0.000
#       Type    : float
#       Usage   :
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793863

si_xtalk_exit_on_number_of_reevaluated_nets

Specifies a maximum number of nets selected for reevaluation, below which PrimeTime-SI exits the analysis loop.

TYPE

integer

DEFAULT

0

DESCRIPTION

Specifies a maximum number of nets selected for reevaluation. PrimeTime-SI exits the analysis loop after completing the current iteration, when the number of nets selected for reevaluation in the the next iteration is less than this number.

This variable is one of a set of six variables that determine exit criteria; PrimeTime-SI exits the analysis loop after completing the current iteration if one or more of the following is true:

1. The number of iterations performed equals the value of the **xtalk_max_iteration_count** variable.

2. All delta delays fall between the values of the **si_xtalk_exit_on_min_delta_delay** and **si_xtalk_exit_on_max_delta_delay** variables.

3. The number of nets selected for reevaluation in the next iteration is less than the value of the **si_xtalk_exit_on_number_of_reevaluated_nets** variable.

4. The percentage of nets (relative to the total number of nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_reevaluated_nets_pct** variable.

5. The percentage of nets (relative to the number of cross-coupled nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_coupled_reevaluated_nets_pct** variable.

6. You manually exit the analysis loop by pressing Control-C to send an interrupt signal to the PrimeTime process. The interrupt is handled as any other exit criteria, at the end of the current iteration of the crosstalk analysis. You cannot interrupt iteration immediately without exiting PrimeTime.

To determine the current value of this variable, type **printvar si_xtalk_exit_on_number_of_reevaluated_nets**.

Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938

```
set_param si1 xtalk_exit_on_number_of_reevaluated_nets 0
#       Type    : uint
#       Usage   :
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793865

si_xtalk_exit_on_reevaluated_nets_pct

Specifies a maximum percentage of nets selected for reevaluation relative to the total number of nets, below which PrimeTime-SI exits the analysis loop.

TYPE

float

DEFAULT

0

DESCRIPTION

Specifies a maximum percentage of nets reselected for evaluation, relative to the total number of nets. PrimeTime-SI exits the analysis loop after completing the current iteration, when the percentage of nets selected for reevaluation in the next iteration is less than this number.

This variable is one of a set of six variables that determine exit criteria; PrimeTime-SI exits the analysis loop after completing the current iteration if one or more of the following is true:

1. The number of iterations performed equals the value of the **xtalk_max_iteration_count** variable.

2. All delta delays fall between the values of the **si_xtalk_exit_on_min_delta_delay** and **si_xtalk_exit_on_max_delta_delay** variables.

3. The number of nets selected for reevaluation in the next iteration is less than the value of the **si_xtalk_exit_on_number_of_reevaluated_nets** variable.

4. The percentage of nets (relative to the total number of nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_reevaluated_nets_pct** variable.

5. The percentage of nets (relative to the number of cross-coupled nets) selected for reevaluation is less than the value of the **si_xtalk_exit_on_coupled_reevaluated_nets_pct** variable.

6. You manually exit the analysis loop by pressing Control-C to send an interrupt signal to the PrimeTime process. The interrupt is handled as any other exit criteria, at the end of the current iteration of the crosstalk analysis. You cannot interrupt iteration immediately without exiting PrimeTime.

To determine the current value of this variable, type **printvar si_xtalk_exit_on_reevaluated_nets_pct**.

Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938

```
set_param si1 xtalk_exit_on_reevaluated_nets_pct 0.000
#       Type    : float
#       Usage   :
```

Pl's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793869

si_xtalk_reselect_delta_and_slack

Reselect nets that satisfy both delta delay and slack reselection criteria.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When true, the intersection of sets of nets reselected by delta delay and slack based criteria is used. For a net to be reselected the following must be true: - The net is reselected by absolute delta delay AND - The net is reselected by relative delta delay AND - The net is reselected by setup OR hold slack OR borrowing AND - Critical path reselection is not enabled.

When true, the nets that satisfy only one of the above criteria (e.g., absolute delta) but not others (e.g., slack) are not counted in the report associated with **XTALK-004**.

When false, the union of sets of nets reselected by delta delay and slack based criteria is used. For a net to be reselected the following must be true: - The net is reselected by absolute delta delay OR - The net is reselected by relative delta delay OR - The net is reselected by setup OR hold slack OR borrowing OR - Critical path reselection is enabled AND the net is on the critical path.

To determine the current value of this variable, type **printvar si_xtalk_reselect_delta_and_slack**.

Pl's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018577

xtalk_reselect_delta_and_slack

Persistent parameter that controls whether window filtering is performed on nets based on the reselection criteria xtalk_reselect_delta_delay, xtalk_reselect_delta_delay_ratio, xtalk_reselect_max_mode_slack, and xtalk_reselect_min_mode_slack.

Syntax:

```
xtalk_reselect_delta_and_slack true | false
```

where the values have the following meaning:

`true`	Perform window filtering on nets based on the delta delay, max_mode_slack and min_mode_slack criteria.
`false`	Do not perform window filtering.

The default value is *false*.

PI's Ex. 336 (PrimeTime 2006.12) at ATopTech 1793869

si_xtalk_reselect_delta_and_slack

Reselect nets that satisfy both delta delay and slack reselection criteria.

TYPE

Boolean

DEFAULT

false

DESCRIPTION

When true, the intersection of sets of nets reselected by delta delay and slack based criteria is used. For a net to be reselected the following must be true: - The net is reselected by absolute delta delay AND - The net is reselected by relative delta delay AND - The net is reselected by setup OR hold slack OR borrowing AND - Critical path reselection is not enabled.

When true, the nets that satisfy only one of the above criteria (e.g., absolute delta) but not others (e.g., slack) are not counted in the report associated with **XTALK-004**.

When false, the union of sets of nets reselected by delta delay and slack based criteria is used. For a net to be reselected the following must be true: - The net is reselected by absolute delta delay OR - The net is reselected by relative delta delay OR - The net is reselected by setup OR hold slack OR borrowing OR - Critical path reselection is enabled AND the net is on the critical path.

To determine the current value of this variable, type **printvar si_xtalk_reselect_delta_and_slack**.

PI's Ex. 419 (AP 13.11.rel.4) at ATopTech 0018577

xtalk_reselect_delta_and_slack

Persistent parameter that controls whether window filtering is performed on nets based on the reselection criteria xtalk_reselect_delta_delay, xtalk_reselect_delta_delay_ratio, xtalk_reselect_max_mode_slack, and xtalk_reselect_min_mode_slack.

Syntax:

```
xtalk_reselect_delta_and_slack true | false
```

where the values have the following meaning:

`true`	Perform window filtering on nets based on the delta delay, max_mode_slack and min_mode_slack criteria.
`false`	Do not perform window filtering.

The default value is *false*.

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002587-89			Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938	
Attributes of the cell Object Class				
area	float	The area of a cell. If the cell is hierarchical, this includes net area.	area	: double (read-only) : area
base_name	string	The leaf name of a cell. For example, the base_name of cell U1/U2/U3 is U3.	base_name	: string (read-only) : base name
dont_touch	boolean	Identifies cells to be excluded from optimization in Design Compiler. Cells with the dont_touch attribute set to true are not modified or replaced during compilation in Design Compiler. Setting dont_touch on a hierarchical cell sets the attribute on all cells below it. Set with set_dont_touch, and used by characterize_context and create_timing_context. You can set and unset the dont_touch attribute.	dont_touch	: bool : dont touch in optimization
full_name	string	The complete name of a cell. For example, the full name cell U3 within cell U2 within cell U1 is U1/U2/U3. The full_name attribute is not affected by current_instance.	full_name	: string (read-only) : cell instance fullname
is_sequential	boolean	A cell is sequential if it is not combinational.	is_sequential	: bool (read-only) : is sequential cell
number_of_pins	integer	Number of pins on the cell. The number of pins can be different before and after linking. For example, if some pins were unconnected in a Verilog instance, after linking to the lower-level design, additional pins can be created on the cell.	number_of_pins	: uint (read-only) : number of pins

PI's Ex.18 (PrimeTime 2006.12 Executable) at SNPS_EXE_011				PI's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
Attributes of the cell Object Class				
is_clock_gating_check	cell	boolean	A	is_clock_gating_check : bool (read-only) : is an ICG or combinational cell with clock gating cell
ref_name	cell	string	A	ref_name : string (read-only) : referred lib_cell name
is_memory_cell	cell	boolean	A	is_memory_cell : bool (read-only) : is a memory cell

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002592-95			Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938	
Attributes of the clock Object Class				
`full_name`	`string`	The name of the clock. This is set with `create_clock`. It is either the name given with the `-name` option, or the name of the first object to which the clock is attached. Once set, this attribute is read only.	`full_name`	`: string (read-only)` `: full name`
`period`	`float`	The clock period (or cycle time) is the shortest time during which the clock waveform repeats. For a simple waveform with one rising and one falling edge, the period is the difference between successive rising edges. Set with `create_clock -period`.	`period`	`: float` `: clock period`
`propagated_clock`	`boolean`	Specifies that clock latency (insertion delay) be determined by propagating delays from the clock source to destination register clock pins. If this attribute is not present, ideal clocking is assumed. Set with `set_propagated_clock`.	`propagated_clock`	`: bool (read-only)` `: compute propagate delay on clock network for latency`
`sources`	`string`	This is a collection of the source pins or ports of the clock. The sources are defined with the `create_clock` command.	`sources`	`: collection (read-only)` `: clock source`

Pl's Ex.18 (PrimeTime 2006.12 Executable) at SNPS_EXE_011	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
Attributes of the clock Object Class	
`is_generated clock boolean A`	`is_generated : bool` `: is generated clock`

Pl's Ex.27 (PrimeTime 2010.06) at SYNPS-00006955			Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
Attributes of the clock Object Class			
`master_clock`	`collection`	This attribute is defined for generated clock objects and returns the master clock object for that generated clock.	`master_clock : collection (read-only)` `: master clock of generated clock`

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002604			Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
Attributes of the lib_cell Object Class			
`full_name`	`string`	The name of a library. For example, the `full_name` of library tech1 read in from /u/user/lib1.db is tech1. This name can be ambiguous because several libraries of the same name can be read in from different files.	`full_name : string (read-only)` `: full name`
`source_file_name`	`string`	The name of the file from which the library was read. For example, the source_file_name of library tech1 read in from /u/user/lib1.db is /u/user/lib1.db.	`source_file_name : string (read-only)` `: source file name`

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002605-06			Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938		
Attributes of the lib_cell Object Class					
area	float	A floating-point value representing the area of a library cell.	area	: double : cell area	(read-only)
base_name	string	The name of a library cell. For example, the base_name of library cell tech1/AN2 is AN2.	base_name	: string : base name	(read-only)
dont_touch	boolean	Identifies library cells to be excluded from optimization. Values are true (the default) or false. Library cells with the dont_touch attribute set to true are not modified or replaced during compile. Set in Design Compiler with set_dont_touch.	dont_use	: bool : do not use in optimization, equal to (user_dont_use OR auto_dont_use)	
full_name	string	The fully qualified name of a library cell. This is the name of the library followed by the library cell name. For example, the full_name of library cell AN2 in library tech1 is tech1/AN2.	full_name	: string : full name	(read-only)
is_sequential	boolean	This attribute is true if the library cell is sequential.	is_sequential	: bool : is a sequential library cell	(read-only)
number_of_pins	integer	Number of pins on the library cell.	number_of_pins	: uint (read-only) : number of pins	

PI's Ex.18 (PrimeTime 2006.12 Executable) at SNPS_EXE_011	PI's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
Attributes of the lib_cell Object Class	
dont_touch lib_cell boolean A	dont_touch : bool : do not touch in optimization, equal to (user_dont_touch OR auto_dont_touch)
is_memory_cell lib_cell boolean A	is_memory_cell : bool (read-only) : is a memory library cell

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002607-11			Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938	
Attributes of the lib_pin Object Type				
base_name	string	The leaf name of the library cell pin. For example, the base_name of tech1/AN2/Z is Z.	base_name	: string (read-only) : base name
fanout_load	float	A floating-point value representing the fanout load value of a library pin. This value is used in computing max_fanout design rule cost.	fanout_load	: float (read-only) : fanout load
full_name	string	The fully qualified name of a library cell pin. This is the name of the library followed by the library cell name followed by a pin name. For example, the full_name of pin Z on library cell AN2 in library tech1 is tech1/AN2/Z.	full_name	: string (read-only) : full name
max_capacitance	float	A floating-point value representing the maximum capacitance design rule limit for a library pin.	max_capacitance	: float : max capacitance load
max_fanout	float	A floating-point value representing the maximum fanout design rule limit for a library pin.	max_fanout	: float (read-only) : max fanout
max_transition	float	A floating-point value representing the maximum transition time design rule limit for a library pin.	max_transition	: float : max transition
pin_capacitance	float	A floating-point value representing the capacitance of a library pin.	pin_capacitance	: float : pin cap

Pl's Ex.18 (PrimeTime 2006.12 Executable) at SNPS_EXE_011	Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
Attributes of the lib_pin Object Type	
`pin_direction lib_pin string A`	`pin_direction : in out inout internal unknown (read-only) : signal direction`

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002611-14			Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
Attributes of the net Object Type			
base_name	string	The leaf name of a net. For example, the base name of net i1/i1z1 is i1z1. You cannot set this attribute.	base_name : string (read-only) : base name
dont_touch	boolean	Identifies nets to be excluded from optimization in Design Compiler. Values are true (the default) or false. Nets with the dont_touch attribute set to true are not modified or replaced during compile with Design Compiler. Set with set_dont_touch.	dont_touch : bool : dont touch in optimization
full_name	string	The complete name of a net. For example, the full_name of net i1z1 within cell i1 is i1/i1z1. The full_name attribute is not affected by current instance. The full_name attribute is read-only.	full_name : string (read-only) : full name
total_capacitance_max	float	A floating-point value representing the sum of all pin capacitances and the wire capacitance of a net for maximum conditions. You cannot set this attribute.	total_capacitance_max : float [range: 0.0000 .. inf] (read-only,application) : max total cap
total_capacitance_min	float	A floating-point value representing the sum of all pin capacitances and the wire capacitance of a net for minimum conditions. You cannot set this attribute.	total_capacitance_min : float [range: 0.0000 .. inf] (read-only,application) : min total cap

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002616-25			Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
Attributes of the pin Object Class			

`actual_fall_transition_max`	`float`	A floating-point value representing the largest falling transition time for a pin.	`actual_fall_transition_max : float [range: -inf .. inf]`
			`(read-only,application)`
			`: actual max fall transition`

`actual_fall_transition_min`	`float`	A floating-point value representing the smallest falling transition time for a pin.	`actual_fall_transition_min : float [range: -inf .. inf]`
			`(read-only,application)`
			`: actual min fall transition`

`actual_rise_transition_max`	`float`	A floating-point value representing the largest rising transition time for a pin.	`actual_rise_transition_max : float [range: -inf .. inf]`
			`(read-only,application)`
			`: actual max rise transition`

`actual_rise_transition_min`	`float`	A floating-point value representing the smallest rising transition time for a pin.	`actual_rise_transition_min : float [range: -inf .. inf]`
			`(read-only,application)`
			`: actual min rise transition`

`arrival_window`	`string`	The minimum and maximum arrivals for rise and fall transitions. In order to get the `arrival_window` attribute on pins that are not endpoints, set the variable `timing_save_pin_arrival_and_slack` to true.	`arrival_window : string (read-only,application)` `: arrival window`

clocks	string	The collection of clock objects which propagate through this pin. It is undefined if no clocks are present.	clocks : collection (read-only,application) : clocks
direction	string	The direction of a pin. Value can be in, out, inout, or internal. The pin_direction attribute is a synonym for direction. Directions can change as a result of linking a design, as references are resolved.	direction : in out inout internal unknown (read-only) : signal direction
full_name	string	The complete name of a pin to the top of the hierarchy. For example, the full name of pin Z on cell U2 within cell U1 is U1/U2/Z. The setting of the current instance has no effect on the full name of a pin. See also the lib_pin_name attribute.	full_name : string (read-only) : pin full name
is_three_state	boolean	This attribute is true if a pin is a three-state driver.	is_three_state : bool (read-only) : is tri-state
max_transition	float	A floating-point value representing the maximum transition time design rule limit for a pin.	max_transition : float [range: -inf .. inf] (read-only,application) : max transition constraint

PI's Ex.18 (PrimeTime 2006.12 Executable) at SNPS_EXE_011	PI's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
Attributes of the pin Object Class	
`is_clock_gating_clock  pin  boolean  A`	`is_clock_gating_clock : bool (read-only)` `: is the clock pin of an clock gating cell`
`is_clock_gating_enable  pin  boolean  A`	`is_clock_gating_enable : bool (read-only)` `: is the enable pin of an clock gating cell`

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002630-42			Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
Attributes of the port Object Class			
actual_fall_transition_max	float	A floating-point value representing the largest falling transition time for a port. You cannot set this attribute.	actual_fall_transition_max : float [range: -inf .. inf] (read-only,application) : actual max fall transition
actual_fall_transition_min	float	A floating-point value representing the smallest falling transition time for a port. You cannot set this attribute.	actual_fall_transition_min : float [range: -inf .. inf] (read-only,application) : actual min fall transition
actual_rise_transition_max	float	A floating-point value representing the largest rising transition time for a port. You cannot set this attribute.	actual_rise_transition_max : float [range: -inf .. inf] (read-only,application) : actual max rise transition
actual_rise_transition_min	float	A floating-point value representing the smallest rising transition time for a port. You cannot set this attribute.	actual_rise_transition_min : float [range: -inf .. inf] (read-only,application) : actual min rise transition
clocks	string	The collection of clock objects which propagate through this port. It is undefined if no clocks are present.	clocks : collection (read-only,application) : clocks
direction	string	The direction of a port. Value can be in, out, inout, or internal. The port_direction attribute is a synonym for direction. You cannot set this attribute.	direction : in out inout internal unknown : signal direction

full_name	string	The name of a port. You cannot set this attribute.	full_name	: string (read-only) : full name
max_transition	float	Specifies the maximum transition time for the net connected to this port. The compile command ensures that value. Set with set_max_transition.	max_transition	: float [range: -inf .. inf] (read-only,application) : max transition constraint

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002645-651			Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
Attributes of the timing_path Object Class			
clock_uncertainty	float	The clock uncertainty of the timing_path. The uncertainty can be defined with the set_clock_uncertainty command.	clock_uncertainty : float [range: -inf .. inf] (read-only) : clock uncertainty
endpoint	string	The timing endpoint name of the timing_path, for example, U1/U5/par_reg/D.	endpoint : collection (read-only) : end point
endpoint_clock	string	The clock name of the clock at the path endpoint.	endpoint_clock : collection (read-only) : end-point clock
endpoint_clock_close_edge_type	string	The type of clock edge (rise or fall) that closes (latches) the data.	endpoint_clock_close_edge_type : rise fall (read-only) : end-point clock close edge type
endpoint_clock_close_edge_value	float	The value of the closing edge of the endpoint clock.	endpoint_clock_close_edge_value : float [range: -inf .. inf] (read-only) : end-point clock close edge value
endpoint_clock_is_inverted	boolean	Returns true if the endpoint clock has been inverted.	endpoint_clock_is_inverted : bool (read-only) : end-point clock is inverted
endpoint_clock_is_propagated	boolean	Returns true if the endpoint clock is a propagated clock, false if it is an ideal clock. You can set a clock as propagated using the set_propagated_clock command.	endpoint_clock_is_propagated : bool (read-only) : end-point clock is propagated

endpoint_clock_latency	float	The latency of the endpoint clock. If the clock is propagated, it is the computed latency (or delay) from the clock source to the endpoint. You can set clock latency using the set_clock_latency command.	endpoint_clock_latency : float [range: -inf .. inf] (read-only) : end-point clock latency
endpoint_clock_open_edge_type	string	The type of clock edge (rise or fall) that opens the endpoint latch. If the endpoint is edge-triggered, the open and close edges are the same.	endpoint_clock_open_edge_type : rise fall (read-only) : end-point clock open edge type
endpoint_clock_open_edge_value	float	The value of the opening edge of the endpoint clock.	endpoint_clock_open_edge_value : float [range: -inf .. inf] (read-only) : end-point clock open edge value
endpoint_clock_pin	string	The complete path name of the endpoint clock pin, for example, U23/U_reg/out_reg[2]/CP.	endpoint_clock_pin : collection (read-only) : end-point clock pin
endpoint_hold_time_value	float	The value of the register hold time at the timing endpoint. For example, for a flip-flop this would be the library hold time for the flip-flop cell.	endpoint_hold_time_value : float [range: -inf .. inf] (read-only) : end-point hold time
endpoint_is_level_sensitive	boolean	Returns true if the endpoint is a level-sensitive device, for example, a latch. Returns false if the endpoint is edge-triggered.	endpoint_is_level_sensitive : bool (read-only) : end-point is level sensitive
endpoint_output_delay_value	float	The value of the output delay of the timing endpoint. You can set the output delay value with set_output_delay.	endpoint_output_delay_value : float [range: -inf .. inf] (read-only) : end-point output delay
endpoint_recovery_time_value	float	The value of the recovery time at the timing endpoint. Recovery and removal times are often defined for the asynchronous set and clear pins of registers.	endpoint_recovery_time_value : float [range: -inf .. inf] (read-only) : end-point recovery time

endpoint_removal_time_value	float	The value of the removal time at the timing endpoint. Recovery and removal times are often defined for the asynchronous set/clear pins of registers.	endpoint_removal_time_value : float [range: -inf .. inf] (read-only) : end-point removal time
endpoint_setup_time_value	float	The value of the setup time at the timing endpoint. Recovery and removal times are often defined for the asynchronous set and clear pins of registers.	endpoint_setup_time_value : float [range: -inf .. inf] (read-only) : end-point setup time
path_group	string	The path group of the timing path.	path_group : string (read-only) : path group name
path_type	string	The type of timing path (maximum or minimum). A path for a setup check is path_type of max.	path_type : max min min_max max_rise max_fall min_rise min_fall (read-only) : path type
points	string	Returns a collection of the timing points that comprise a timing path. For example, the timing points listed in the left-hand column of a report_timing command correspond to this collection. A single timing path can consist of many timing points. You can iterate through each timing point using foreach_in_collection.	points : collection (read-only) : timing points
slack	string	The slack of the timing path. Negative values represent violated paths. Corresponds to the slack of a timing report.	slack : float [range: -inf .. inf] (read-only) : slack
startpoint	string	The startpoint of the timing path. Corresponds to the startpoint in the header of a timing report.	startpoint : collection (read-only) : start point

startpoint_clock	string	The startpoint clock name of the timing path.	startpoint_clock : collection (read-only) : start-point clock
startpoint_clock_is_inverted	boolean	Returns true if the startpoint clock is inverted.	startpoint_clock_is_inverted : bool (read-only) : start-point clock is inverted
startpoint_clock_is_propagated	boolean	Returns true if the startpoint clock is a propagated clock, false if it is an ideal clock. You can set a clock as propagated using the set_propagated_clock command.	startpoint_clock_is_propagated : bool (read-only) : start-point clock is propagated
startpoint_clock_latency	float	The latency of the startpoint clock. If the clock is propagated, it is the computed latency (or delay) from the clock source to the endpoint. You can set clock latency using the set_clock_latency command.	startpoint_clock_latency : float [range: -inf .. inf] (read-only) : start-point clock latency
startpoint_clock_open_edge_type	string	The type of clock edge (rise or fall) that launches the data.	startpoint_clock_open_edge_type : rise fall (read-only) : start-point clock open edge type
startpoint_clock_open_edge_value	float	The value of the opening edge of the startpoint clock.	startpoint_clock_open_edge_value : float [range: -inf .. inf] (read-only) : start-point clock open edge value
startpoint_input_delay_value	float	The value of the startpoint input delay.	startpoint_input_delay_value : float [range: -inf .. inf] (read-only) : start-point input delay

startpoint_is_level_sensitive	boolean	Returns true if the startpoint is a level-sensitive device, such as a latch. Returns false if the startpoint is edge-triggered.	startpoint_is_level_sensitive : bool (read-only) : start-point is level sensitive
time_borrowed_from_endpoint	float	Returns the amount of time borrowed from the timing endpoint. Time borrowing occurs in paths involving level-sensitive devices.	time_borrowed_from_endpoint : float [range: -inf .. inf] (read-only) : time borrowed from end-point
time_lent_to_startpoint	float	Returns the amount of time lent to the timing startpoint. Time borrowing occurs in paths involving level-sensitive devices.	time_lent_to_startpoint : float [range: -inf .. inf] (read-only) : time lent to start-point

Pl's Ex. 6 (PrimeTime 2006.12) at SYNPS-00002652			Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938	
Attributes of the timing_point Object Class				
arrival	string	The arrival time at the timing point, but not accounting for the following. - Clock latency to the startpoint clock - Time lent to the startpoint (due to latch borrowing) - Input delay These must be added to the arrival attribute value to arrive at the total arrival time based on the desired startpoint.	arrival	: float [range: -inf .. inf] (read-only) : arrival time
object	string	The object at this point in the timing path.	object	: collection (read-only) : pin or port
rise_fall	string	Returns rise if the timing point is a rising-edge delay. Returns fall if the timing point is a falling-edge delay.	rise_fall	: rise fall (read-only) : rise or fall type
slack	string	The slack value at the timing point.	slack	: float [range: -inf .. inf] (read-only) : slack
transition	string	The transition value at the timing point.	transition	: float [range: -inf .. inf] (read-only) : transition time

Pl's Ex.27 (PrimeTime 2010.06) at SYNPS-00007041			Pl's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
Attributes of the timing_point Object Class			
voltage	float	The voltage level of the pin.	voltage : float [range: -inf .. inf] (read-only) : rail voltage

PI's Ex.27 (PrimeTime 2010.06) at SYNPS-00007041			PI's Ex. 602 (AP 13.11.rel.4 Executable) at ATopTech 0057938
Attributes of the timing_point Object Class			
`voltage`	`float`	The voltage level of the pin.	`voltage : float [range: -inf .. inf] (read-only) : rail voltage`