

V2003.12 Liberty Update

Somil Ingle

CAE

Synopsys, Inc.



October 16, 2003

 *Your Interoperability Partner*

Outline

- Liberty Noise Modeling Enhancements
- Clock Insertion Delay Model
- Generic Parameters in SPM
- Half Unate Sequential Timing Arcs
- Plib enhancements

Noise Modeling Enhancements

■ Motivation

- Extend Liberty Noise Modeling syntax to support tied-off cells and Noise Width range limits

■ Features

- Modeling steady state resistance and I/V curves of tied-off cells
- Noise Width range limits

Noise Modeling in Tied-off cells

- In tied-off cells the output pins are tied to either high or low
- No need to define timing information with related pins
- Model noise information using new boolean attribute “tied_off” and set it to “true”

Noise Modeling in Tied-off cells


```
pin(namestring) {  
    ...  
    function : "0 | 1" ; /* tied to low or  
        high*/  
    timing() {  
        tied_off : boolean;  
        /* timing type is not defined */  
        ---Noise Information---  
        /* I/V curves or steady state resistance */
```

Noise Width Range Limits

- Two new attributes to specify noise width range limit
- The attributes are optional and specify min/max values of the `input_noise_width`
- Always used as a pair and can be present only on input or inout pins

Noise Width Range Limits

```
pin(namestring) {  
    ...  
    /* used for noise immunity or propagation */  
    min_input_noise_width : float;  
    max_input_noise_width: float;  
    ...  
}
```



Modeling Clock Insertion Delay

- **Motivation**


- Extend Liberty syntax to capture arrival timing paths from an input clock pin to clock pins internal to the cell

- **Features**

- Timing_types *max_clock_tree_path* and *min_clock_tree_path* are used in timing groups under a clock pin
- No related_pin specified and table is indexed only by input_transition time of the clock

Modeling Clock Insertion Delay

```
pin(clock_pin) {  
  ....  
  timing (arc_name) {  
    timing_type : max_clock_tree_path | min_clock_tree_path;  
    timing_sense : positive_unate | negative_unate | non_unate  
    ;  
    cell_rise(template1) { /* rise delay of clock tree path */  
      ...  
    }  
    cell_fall(template1) { /* fall delay of clock tree path */  
      ...  
    }  
  }  
}
```



Generic Parameters in SPM

■ Motivation

- Extend Liberty SPM syntax to support generic variables

■ Features

- Support up to five generic variables in SPM
- New group “*user_parameters*” to define the default values for these generic variables
- Enhance “*operating_conditions*” group to support these five generic variables
- “*poly_template*” group enhanced to support these variables

Generic Parameters in SPM

```
library(new_lib) {  
  user_parameters() {  
    parameter1: 0.18;  
  }  
  operating_conditions ("nominal") {  
    process : 1.000000 ;  
    temperature : 25.00 ;  
    voltage : 1.80 ;  
    parameter1 : 0.20;  
  }  
  poly_template (T1) {  
    variables ("input_transition_time", "parameter1");  
  }  
}
```

Half Unate Sequential Timing Arcs

- Motivation
 - Support half unate sequential arcs
- Features
 - support the half unate sequential timing arcs: the rising/falling_edge arcs that only have half of the legal transitions
 - You can specify either `cell_rise(rise_propagation)/rise_transition` or `cell_fall(fall_propagation)/fall_transition`

Half Unate Sequential Arcs

```
timing() {  
  timing_type : rising_edge;  
  timing_sense : positive_unate;  
  cell_rise() {...}  
  rise_transition() {...}  
}
```



Plib Enhancements

- Antenna rules for 90nm
- New spacing, via-array and contact rules for 90nm
- New metal density rules
- Attributes for supporting LEF 5.5 syntax
- Plib documentation for all the latest enhancements available in Jan

Summary

- New features added to Liberty for the V-2003.12 release
- Liberty V-2003.12 docs will be available at the end of Jan 2004
- You can get everything related to Liberty at the following location

www.synopsys.com/tapin