Application Note for DC Ultra Library Guide

Introduction

Synopsys has added a new delay optimization algorithm to Design Compiler Ultra™ (DC Ultra) that provides delay quality of result (QoR) improvement in many cases, when using a suitable technology library. This algorithm uses a new logic synthesis load-independent delay model that is a hybrid between the constant delay model and the traditional non-linear delay model (NLDM).

The initial library analysis and evaluation steps can automatically determine whether a technology library is suitable for this new optimization algorithm. This application note is intended for library providers who want to tune their libraries to take advantage of this optimization capabitlity.   To invoke these capablilities, apply the variable set_ultra_optimization and set compile_new_optimization to true.

Pre-Synthesis Library Analysis

DC Ultra's library analysis creates a scalable technology library (internal to the Design Compiler) by analysing discrete cells from the target library. This step performs cell clustering, non-linear delay model generation and input capacitance model generation.

Cells having the same logic function and similar characteristerics are grouped into a cluster. Many clusters of the same functions may exist but only one with the best fit will be used. From this cluster, the algorithm derives a continuous (scalable) model.   The scalable cells are then characterized with a new delay model, in which delay is modeled as a non-linear function of gain and slew. This model is computed based on the delay of all the discrete cells within a given cluster.

In addition, library analysis also creates models for area and input capacitance of the scalable cells. The new optimization algorithms then work with this scalable model. The resulting netlist is descretized, i.e. the scalable cells are converted to discrete cells in the target library. Following this, the traditional technology-dependent optizmiztion techniques in Design Compiler are invoked for further optimization.

When the variables are set, library analysis is performed automatically. This analysis considers only a subset of the combinational cells with single output.   Following library analysis is the library evaluation step.

Library Evaluation

Definitions:

discrete cell
a library cell. In this application note, discrete cells are referred to as the set of library cells being included in the library analysis and evaluation steps.

cluster
a group of discrete cells having the same logic function and similar delay and capacitance characteristics, typically a cluster contains cells with different sizes or drive strength.

gain
gain of a discrte cell is the ratio of output load capacitance to the average input capacitance of the discrete cell. For example, an inverter with an input capacitance of XX , a gain value of 3 means the load at its output is 3XX. For the same input capacitanc but the gain value of 4 means the load at its output is 4XX. For cells with more than one input, the scaling size is used as the input capacitance.

scaling size
scaling size of a library cell is defined as the average input capacitance over all the inputs of the cell.

basic set of cells
a set of cells includes inverters, nand2, nand3, nor2 and nor3. This set of cells will be considered in the library analysis/evaluation steps.

extended set of cells
a set of cells includes and2, and3, or2, or3, xor2, xor3, xnor2, xnor3, mux2, mux3, aoi, and oai. This set of cells will be considered in the library analysis/evaluation steps.

NLDM
non-linear delay model

INN / inn
inverter, nand2, and nor2.

Library Development guideline

The previous sections describe briefly the elements that library analysis and evaluation steps focus on. The accurarcy of the algorithms depends upon the accuracy of the scalable model derived during library analysis. This section provides some guideline to build appropriete cell library that will take advantage of this delay optimization algorithm.

1. Functional classes

Library evaluation considers only a subset of the combinational cells that were analyzed during library analysis. These are categorized into two sets: basic and extended. All cells that do not belong to these two categories are ignored during library evaluation. The two sets are:

- Basic set of cells: Inverters, NAND (2 and 3 inputs), NOR (2 and 3 inputs)
- Extended set of cells: AND (2 and 3 inputs), OR (2 and 3 inputs), XOR (2 and 3 inputs), XNOR (2 and 3 inputs), MUX (2 and 3 inputs), IMUX (2 and 3 inputs), AOI, and OAI

Buffers and other common library cells have not been included here, but they are used by the technology-dependent optimization engine and should be included in the library.

2. Number of discrete cells in a cluster

For library evaluation, the number of sizes of a logic function is the number of discrete cells in the cluster associated with the scalable cell implementing that function. The recommended number of sizes for the logic functions in the basic and extended sets is as follows:

- Basic set of cells:

  There should be a minimun of six discrete cells corresponding to all scalable cells in the basic set. Preferably, the number of sizes of each logic function in the basic set should be greater than six.
- Extended set of cells:

  There should be a minimum of five discrete cells corresponding to all scalable cells in the extended set. Preferably, the size of each logic function in the extended set should be greater than five.

Note : Simply having the recommended number of discrete cells for a particular logic function is not sufficient. It is important for the discrete cells to have similar characteristics for them to be included in the same cluster (see other recommendations below).

3. Size distribution of discrete cells within a cluster

The sizes of discrete cells within each cluster should be geometrically distributed. The

4. Input capacitance

The variance of the capacitances among different input pins of each discrete cell should be within 10 percent of its average input pin capacitance. This will ensure that the scaling size approximation that is used in library analysis will be less erroneous.

5. Timing characteristics of discrete cells within a cluster

For a given reasonable gain value, the corresponding timing arcs of all descrete cells within a cluster should be within 10 percent of each other. In the following example, the output load capacitance is set to be 3.5 times the input capacitance for a set of three different discrete nand2, i.e. a gain value of 3.5. Cells Nand22 and Nand23 have consistance timing arcs and would be clustered together. Cells Nand1 will not be part of the cluster because its timing arc at pinA is significantly different from the other two discrete cells in the cluster.

Cell	Avg input Cap	Output Load	pinA(rise)	pinA(fall)	pinB(rise)	pinB(fall)
Nand21	0.014	0.049	0.2545	0.3325	0.0968	0.0722
Nand22	0.026	0.091	0.0877	0.0768	0.0994	0.0705
Nand23	0.050	0.175	0.0901	0.0797	0.1015	0.0738

6. Maximum output capacitance constraint

The maximum output capacitance constsraint of the each discrete cell within a cluster should be linearly proportional to the average input capacitance of the discrete cell.

7. Other guidelines

The above guidelines are targeted to build library cells that will take advantage of the new delay algorithms of DC Ultra. A library may contain many cells that do not meet the above criteria. Clustering will just ignore them as long as there are sufficient cells that do meet the above requirements. Buffers and other nonconfromant cells should also be present in technology libraries for optimization reasons other than delay cost. All library cells are considered during technology-dependent optimization such as area recovery to further improve the overall QOR.

Library evaluation report

A library evaluation report will be generated when the library_analysis command is applied. This command works only with a library source file (.lib). This report shows the functional classes (clusters), number of discrete cells in each cluster, the size consistency max error and the delay max errors. The size and delay consistency errors are derived from some complex algorithms, based on the size and delay models of the cells in each cluster. To determine if a library is suitable for the new optimization engine, these errors should be as small as possible and not exceed the maximum permissible range. Adhering to the guidelines in the previous section will minimize these errors.

Within a cluster, discrete cells are sorted based on their scaling size. The size consistency max error is computed for each cell, by comparing its NLDM to the NLDM of the cell succeeding the current cell after normalizing the NLDM tables based on the scaling size of the two cells. The maximum error for all the cells in a cluster is reported. The overall average should not exceed 10% and the average of the inn cells (inverter, nand2, nor2) must not exceed 15%.

The delay max error is the maximum delay error beween the NLDM table of the scalable cell derived by library analysis and the NLDM table of all the corresponding discrete cells. It is based on the maximum overall delay arcs and the rise/fall NLDM tables. The average delay max error for all the scalable cells in all clusters should not exceed 20%.

Below is an example of a report of a qualified library:

Cell	num of sizes	size consistence max err	delay max error
and2	7	8.1%	7.8%
aoi4	10	5.9%	7.4%
oai4	9	5.0%	4.9%
inv	21	15.1%	15.5%
imux2	8	10.3%	14.5%
nand2	24	11.0%	14.0%
nand3	19	11.7%	13.0%
nor2	28	17.6%	18.4%
nor3	26	9.0%	13.4%
or2	7	4.7%	5.5%
xnor2	9	6.5%	6.3%
xor2	10	5.9%	10.4%
Avg:	14.8	9.2%	10.2%

To invoke the new delay optimization alogarithms

To invoke the library analysis, evaulation and optimization algorithms, set the two switches:

- `set_ultra_optimization`
- `compile_new_optimization = true`

To obtain a library evaluation report, follow these steps:

- `read_lib xxx.lib`
- `target_library = xxx.db`
- `library_analysis`

If library evaluation determines that a library is not suitable for optimization, it will switch back to the base optimization engine.