設 計 自 動 化 58－9
（1991. 7. 12）

ＶＬＳＩ遅延ライブラリ作成支援システム

豊田 徹　鈴木正宏　西口信行　岸本有豊

日本電気株式会社

ＶＬＳＩ設計用遅延ライブラリ作成を支援するシステムについて述べる．遅延ライブラリ作成において，回路シミュレーションファイルの自動作成，回路シミュレータの分散実行，回路シミュレーション結果の解析からなるキャラクタライゼーションの自動化を行った．遅延ライブラリ作成に関わる全てのデータ要素をオブジェクト指向データベースに格納し，版数管理を可能にした．データを中心に見せるユーザインターフェースを採用した．これらによって，一定の品質の遅延ライブラリを短期間に作成可能になった．

A VLSI DELAY LIBRARY GENERATION AID SYSTEM

Toru Toyoda　Masahiro Suzuki　Nobuyuki Nishiguchi　Aritoyo Kishimoto

NEC Corporation

1753,Shimonumabe, Nakahara-ku, KAWASAKI, KANAGAWA, 211 JAPAN

A VLSI delay library generation aid system was described. The system has an automatic delay characterization from layout data with an object oriented data base. The characterization consists of a circuit simulation input file generator, a distributed execution of circuit simulation and a circuit simulation output file analizer. The system also supports data management and friendly look and feel. It is able to generate delay libraries with high quality at the short time.

1．はじめに

近年，超ＬＳＩ設計へのＣＡＤ導入が盛んにおこなわれている．ゲートアレイをはじめとする最近のＡＳＩＣ品種の設計においては，ＡＳＩＣカスタマが各種ＥＷＳ上の設計環境を使用して回路設計及び論理シミュレーションを行う．このため，ＡＳＩＣベンダーは各種設計環境に対応したライブラリを遅滞なく提供しなければならない．このライブラリにはカスタマが回路設計に使用するブロックのシンボルのライブラリ，プロセス毎のブロックのレイウトのライブラリ，遅延論理シミュレーション用のライブラリなどがある．

これらの過程で作成されるライブラリの数は多数にわたり，ブロックの追加やレイアウトの変更に伴う改版分も含めると，膨大な数のデータを作成，管理を行わなければならない．

また，１つのプロセスに含まれるブロック数は増加する一方であり，プロセスの更新周期は短くなってきている．さらに，多数のライブラリ設計を扱うため，版数管理も行う必要がある．

そこで本文では，従来では自動化の難しかった遅延論理シミュレーションライブラリの作成の全自動化手法（ブロックキャラクタライゼーション）を示すとともに，システム全体にオブジェクト指向に基づく統一データベースを採用することによって，多数のライブラリのデータ管理，版数管理が可能になったので，報告する．

第２章ではＡＳＩＣにおけるライブラリの作成の手順の概略を述べる．第３章では遅延論理シミュレーションの作成手法であるブロックのキャラクタライゼーション手法について説明する．第４章では本システムの構成要素を述べ，オブジェクト指向データベースを中心とした各ツールの配置と，版数管理の仕組み，ユーザインタフェースについて説明する．第５章では本システムの試行結果と今後の課題について述べる．

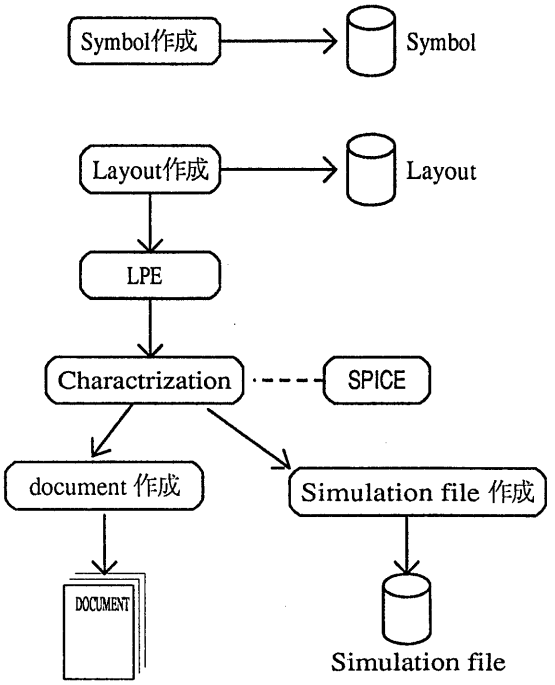

図１　　ライブラリ設計の流れ

2．ライブラリの作成手順概要

図１にライブラリ設計の流れを示す．

１つは回路図の論理シンボルライブラリを作成することである．シンボルは１ブロックに１つあり，多くの場合プロセスを越えて使用される．その為１ブロックについて１度作成すれば，それを改版することはほとんど無い．但し，異なる種類の設計環境ではシンボルファイルの形式やファイルに含まれる情報が異なるので，設計環境毎にシンボルのライブラリを用意しなければならない．

次にレイアウトライブラリを作成する．レイアウトはブロック毎に作成する．回路図エディ

タや論理シミュレータとは無関係である為，設計環境とは関係なく１種類だけ作成する．当然のことながら，プロセスが異なればレイアウトも異なる．

レイアウトライブラリが作成されると，各ブロックの遅延時間，出力端子の駆動力，端子容量，端子しきい値，消費電力などを求める．これらを求める為には，まず，レイアウトから抵抗や容量等を含んだネットリストを抽出する．一般的にこの工程はLPE(Layout Parameter Extraction) ツールで自動で行われる．

次に，回路シミュレータに入力する波形を作成し，プロセス毎のモデルパラメータを付加し，温度，電源電圧等の使用条件を設定して回路シミュレーションを行う．回路シミュレーションは，負荷容量を変更しながら何度か繰り返す．回路シミュレーションが終了したら，結果のファイルから各条件でのピン間遅延値を求め，それらの値から必要な値を計算する．この工程をキャラクタライゼーションと呼ぶ[1] [2]．

遅延時間などが計算されたら，それらを設計環境毎の論理シミュレーションライブラリに変換する．論理シミュレーションライブラリは，含まれる情報の種類によって３つに分類できる．第１は，主にピン間遅延値のみを持つものであり，これは既にもとめてあるピン間遅延値等をフォーマット変換することで作成できる．第２は，ピン間遅延値の他に機能に関する記述を含むものであり，これはピン間遅延値をフォーマット変換した上に，機能に関する記述をマージすることで作成できる．第３は，ブロックの動作を，論理シミュレータが準備しているプリミティブなゲートを使って記述し，ピン間遅延値を満たす様に，プリミティブなゲートに対して遅延値を割り付けるものである．この場合，ピン間遅延値を各プリミティブゲートに割り振る作業が必要となる．

最後に，各ブロックの遅延時間を知る為のドキュメントを作成して，全ての作業が完了する．

３．キャラクタライゼーション

３．１　ＳＰＩＣＥ入力ファイル作成部

ブロックの遅延時間などを知るためにＳＰＩＣＥを実行するが，そのためには各入力端子に加える波形を定義する必要がある．ブロックの機能によって波形を定義する．インバータであれば入力は１端子で出力も１端子で，入力端子にはRiseの波形とFallの波形の各々が入った時の遅延時間を調べたい．その時は①入力端子ＡにRiseを加えた時，出力端子ＹはFallになる．②入力端子ＡにFallを加えた時，出力端子ＹはRiseになる．という２つの論理的な波形関係を波形ライブラリ に定義すれば良い．　ＳＰＩＣＥ入力ファイル作成部は，波形ライブラリから論理的な波形関係を読みだし，プロセスライブラリから入力波形の傾きや０レベル，１レベルの電位を読みだし，それらからＳＰＩＣＥの入力波形を作成する．

出力端子に接続する負荷容量が増加すると遅延時間も増加するが，その増加率を調べるためには，接続する負荷容量を変えてＳＰＩＣＥを実行しなければならない．プロセスライブラリにはＳＰＩＣＥ実行で使用すべき負荷容量を記述しておき，その全てに対してＳＰＩＣＥ記述を作成する．

同様に，条件を変化させることにより遅延時間が変化するものとして，本手法ではいくつかの項目を用意している．負荷容量の他には，入力波形の傾き，電源電圧，プロセスによるＭＯＳのゲート長の変動値，プロセスによる抵抗の増減率，温度，モデルパラメータそのものがある．これらに関して，条件を変化させたＳＰＩＣＥ記述が作成される．

３．２　ＳＰＩＣＥ実行

ＳＰＩＣＥ実行は，一連のライブラリ作成作業のうちで最もＣＰＵコストのかかる部分である．特に，３．１で述べた様に条件振りを行った場合，ＳＰＩＣＥの実行回数はとても多くな

る．そのため，ＥＷＳで処理を行う場合は何らかの方法で複数台のＥＷＳに分散させなければならない．

本手法では，ＳＰＩＣＥを実行するホストを管理するテーブルを持っていて，各ホストで起動されたＳＰＩＣＥが常に１本以下になる様にＳＰＩＣＥを複数台のＥＷＳに分けて起動する．

３．３　遅延時間算出

ＳＰＩＣＥの実行が終了したら各入力端子から出力端子までの遅延時間を計算する．ＳＰＩＣＥの結果ファイルは，時刻と指定した各ノードの電位羅列になっているので，過渡解析を行い，ノードの電位が論理しきい値を横切った時刻を調べ，出力端子での時刻と入力端子での時刻の差をとることで遅延時間が計算できる．

遅延時間計算に使用する論理しきい値は，プロセスライブラリでも指定できるが，入力端子のしきい値に関しては，ＤＣ解析で入力端子のしきい値を調べ，その値を使用した方が正確である．プロセスライブラリに一般的なブロックの入力端子のしきい値を記述しておく．ＤＣ解析の結果から，出力端子の電位が一般的なブロックの入力端子のしきい値と一致した時，入力端子に加えている電位をそのブロックのしきい値としている．

何種類かの負荷容量における遅延時間が求まると，負荷容量の増加に対する遅延時間の増加の割合が計算できる．これを出力端子の駆動力としている．

条件振りをして異なる条件での遅延時間が計算された場合は，各々の条件に対する遅延時間の増加割合を計算する．例えば，負荷容量と温度をプロセスライブラリに複数記述したとすると，温度を一定に保ち負荷容量を変化させた時の遅延時間の変化率が，各温度について求まる（つまり，各温度条件における，負荷容量に対する遅延時間の変化率が求まる）．負荷容量に対する遅延時間の変化率（複数）とプロセスライブラリに記述された「ばらつきの最小値／最大値」から，負荷容量に対する遅延時間の増加割合の最大値と最小値と標準値を次式で計算してデータベースに保存する．

TNOM1……ばらつき最小値
TNOM2……ばらつき最大値
MIN＝TNOM1*変化率の最小値
MAX＝TNOM2*変化率の最大値
TYP=MIN+TNOM*(MAX-MIN)

また，温度に対する遅延時間の変化率から，同様の計算式で温度に対する遅延時間の増加割合を計算してデータベースに保存する．

尚，遅延時間，端子のしきい値などでも上記の計算式を使用している．

３．４　タイミング情報算出

ラッチやフリップフロップ（ＦＦ）に関してはタイミング情報（セットアップ／ホールド／リリース／リムーバブル）を計算する．本システムは，処理速度を重視した方法と精度を重視した方法の，２つのモードを備えている．

処理速度を重視するモードでは，図２に示す様に，データ端子からある内部ノードまでの遅延時間と，クロック端子からある内部ノードまでの遅延時間の差から，タイミング情報を計算する．例えばセットアップの計算式の概略は次の通りである．

DATA1……データ端子側の最小値
CLOCK1……クロック端子側の最小値
CLOCK2……クロック端子側の最大値
SETUP(MIN)=DATA2-CLOCK2
SETUP(MAX)=DATA2-CLOCK1

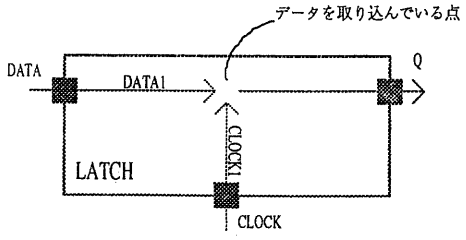

図２　タイミング計算方法

精度を重視したモードでは，クロックが遷位する時刻を基準にしてデータ遷位時刻を変化させ，データが正常に取り込めるかどうかを出力端子側で観測する．

3．5　端子情報算出

端子情報は，3．3で計算した，入力端子のしきい値と出力端子の駆動力の他に，端子容量を計算する．端子容量はＬＰＥ後のＳＰＩＣＥネットリストを追いかけて行くことで求める．入力端子はＭＯＳのゲート電極まで，出力端子はＭＯＳの拡散までの容量を考慮している．

3．6　シミュレーションファイル出力

今まで述べてきた方法で計算したデータ要素は，各論理シミュレータ専用のフォーマットで記述することで論理シミュレータに使用できる．本システムではデータベースのデータを各論理シミュレータのフォーマットで出力するプログラムを用意した．遅延を各プリミィブゲートに割り振る必要がある論理シミュレータも存在するので，遅延を各プリミティブに割り振る機能も含ませた．

4．システムの構成

4．1　システム概要

前章で述べたキャラクタライゼーションによって求められた遅延論理シミュレータ用のライブラリを含めた形で，各種ライブラリのデータ管理，版数管理をオブジェクト指向に基づく統一データベースを用いて実現した（図3）．

本システムは，あらゆるデータ要素を格納するデータベースを中心に，全てのプログラムを配置する構成をとっている．

4．2　データベース

一連のライブラリ作成作業では，多くのファイルが作成されるが，これらの多くは共通のデータ要素を持っていたり，親子関係のあるデータを持っていたりする．本システムではデータベースを使用してこれらの管理を行っている．

データの構造を図4に示す．各セル，デザインはオブジェクトとして表現されており，管理されている．すなわち，プロセス単位にデータベースを作成し，データベースはセルというデータオブジェクトを持つ．セルには，デバイスライブラリ，プロセスライブラリ，ブロック，といったプロセス固有のデータオブジェクトを持つ．各々のセルは更に下位にデザインと呼ぶデータオブジェクトを持っている．

デバイスライブラリは，ＳＰＩＣＥ実行時にＳＰＩＣＥが持っている各モデル（PMOS,NMOS等）の特性を与えるデータであり，データ毎に値を入れる場所がある．

プロセスライブラリは，チップの動作条件，計算方法の指定などをまとめたデータオブジェクトである．動作条件としては，図5に概要を示す様に電源電圧，入力波形の傾き，論理しきい値，負荷容量，温度，単位面積あたり／単位周囲長あたりの拡散層容量，ゲート容量，プロセスによるＭＯＳゲート長の変動量，プロセスによる抵抗の変動率などがある．計算方法の指定内容としては，図6に概要を示す様に，各種マージン値，端子容量の計算方法指定などがある．

ブロックは，名前，機能，セル寸法，I/Oレベル，動作記述（複数），ネットリスト（複数），レイアウト，シンボル，端子情報，遅延情報（遅延値，負荷容量による遅延値の増加律，消費電力，タイミング，端子しきい値，端子容量，駆動力など），SPICE記述(サーキットライブラリ)，構成要素数（MOSの数など），波形情報(波形ライブラリ)，などを持っている．これらのオブジェクトはさらに下位のオブジェクトを持つ構造になっている．

データベースは，これらのデータ要素に対して，セルおよびデザインの単位でデータの作成

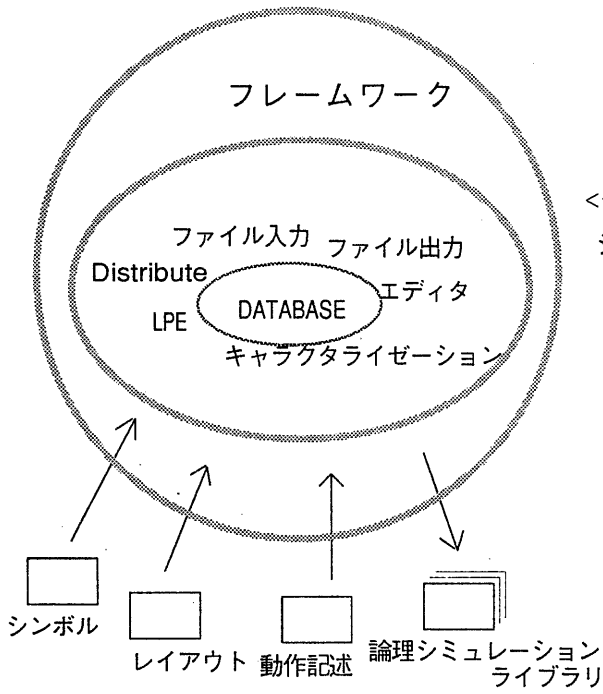

図 3　システム概要

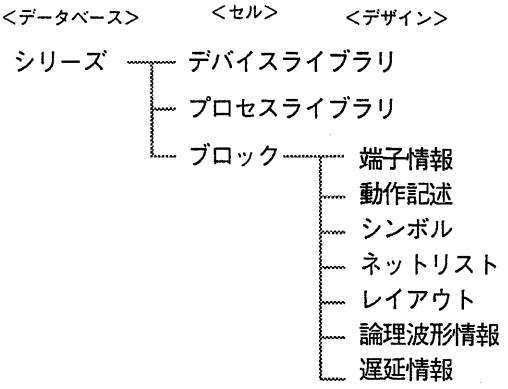

図 4　データ構造

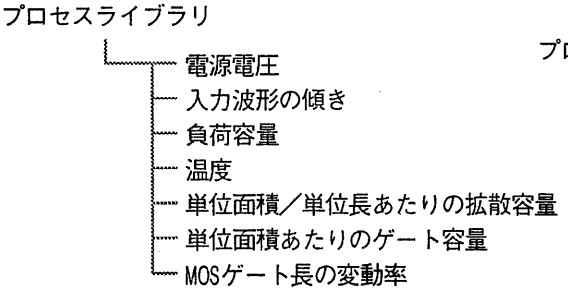

図 5　動作条件の概要

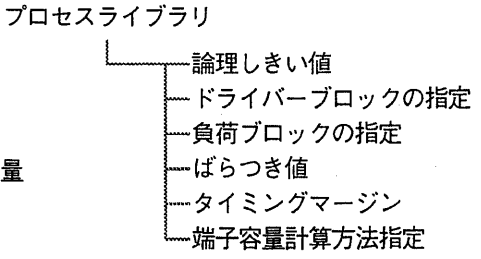

図 6　計算方法の概要

／更新を管理し，セルの単位で版数を管理している．また階層関係を成すセルーデザインに関しては，下位のデザインが更新されたら，それは上位のセルに反映される仕組みになっている．

セルは，各々任意のレベルの版数を保存できる．例えばデフォルトで以前の３版までを保存し，ブロックに関しては５版まで保存する，ということを指定できる．また，版数を指定した削除，版数を指定したロック（書き込みや削除を不可能にする）などの操作も可能である．

４．３　フレームワーク

ＬＳＩ設計者のスキルによらず一定の工数で一定の品質をもったライブラリ作成を目指した．そのためにデータを中心として描いたグラフィカルなユーザインタフェース(MP;Management Program)を用意した．データをクリックすると，そのデータに対して起動できるコマンドがメニューとなって現われる．その中から所望の項目をクリックすればよい．

図７に履歴の表示例を示す．NAND-2 の履歴を調べたい人は，まずデータベースをクリックし，メニューの中から "db_open"（データベースを開く）を選択する．すると新たにウィンドウが開きデザイン（デバイスライブラリ，プロセスライブラリ，ブロック）が表示される．NAND-2 のブロックをクリックすると更にメニューが出るので，その中から "version_list" をクリックすると，更に新しいウィンドウが開き，ブロックNAND-2とNAND-2の下にあるデザインの履歴が表示される．

プロセス "test-1" の論理シミュータ"sim-1" 用の論理シミュレーションライブラリを出力する時は，"test-1" をクリックしてメニューを出し，"OutputFile"の中の"sim-1"をクリックすればsim-1用の論理シミュレーションライブラリが作成される．また，必要に応じてウィンドウが開き必要なファイル名などを問い合わせてくる．

５．適用結果

13MIPSと20MIPSのＥＷＳ各１台を組み合わせた環境で，本システムを遅延ライブラリ作成に適用した．その結果，キャラクタライゼーション部分では，インバータで約２０秒（回路シミュレーション６回），4-BIT SYNC BINARY COUNTER WITH LOADB & RB で約２時間（回路シミュレーション３９６回）を必要とした．シミュレーションライブラリの出力は，遅延をプリミティブゲートへ分配しないもので，１０ブロックあたり約１５秒だった．

６．むすび

本論文では，遅延ライブラリ作成支援システムの概略とキャラクタライゼーションの処理を述べた．まず，データベースに全てのデータを格納し自動で版数管理を行っているため，版数間違いの発生を防ぐことができた．次に，データを中心としたグラフィカルなユーザインターフェイスを採用したため，設計者のスキルに依存せずライブラリ作成を行うことができた．そして，条件振りを含んだキャラクタライゼーションの自動化により，ライブラリ作成を大幅に短縮できた．

今後の課題として，機能記述からの波形ライブラリ自動発生，及びデータベースからのドキュメント出力を検討する予定である．

【参考文献】

[1] Dam Vo:"AUTOMATED SPICE CHARACTERIZATION OF GATE ARRAY AND STANDARD CELL LIBRARIES",Proc.IEEE CustomIntegrated Circuit Conf., 1997 pp.363-366

[2] Teri Hike McFal and Rari Perrey:"CHARACTERIZING A CELL LIBRARY USING ICCS",ASIC'90 ASIC Conferenct and Exhibit,1990 p12-5.1 p12-5.4

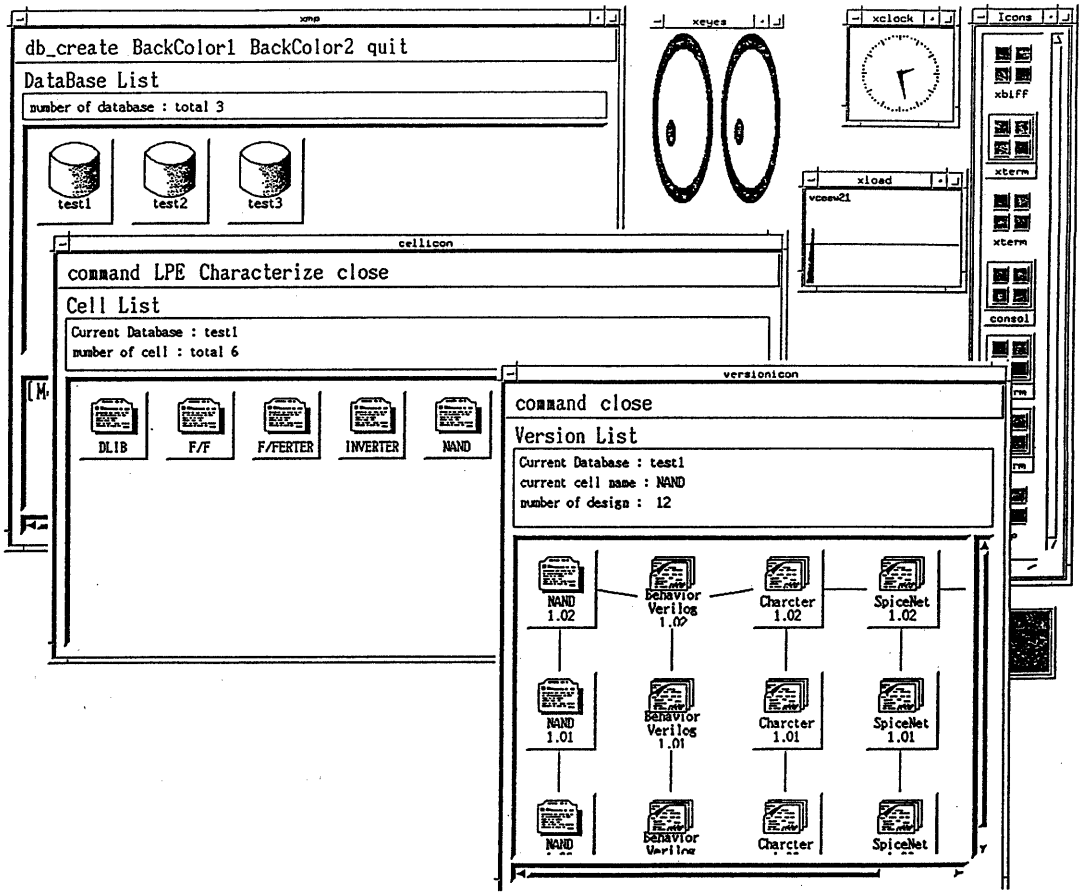
xmp
db_create BackColor1 BackColor2 quit
DataBase List
number of database : total 3
test1
test2
test3
xeyes
xclock
Icons
xbiff
xterm
xterm
consol
xload
vcaev21
cellicon
command LPE Characterize close
Cell List
Current Database : test1
number of cell : total 6
DLIB
F/F
F/FERTER
INVERTER
NAND
versionicon
command close
Version List
Current Database : test1
current cell name : NAND
number of design : 12
NAND 1.02
Behavior Verilog 1.02
Charcter 1.02
SpiceNet 1.02
NAND 1.01
Behavior Verilog 1.01
Charcter 1.01
SpiceNet 1.01
NAND
Behavior Verilog
Charcter
SpiceNet

図7　表示例