SYNOPSYS®

DesignWare High Speed Standard Cell Tapless Logic Library Full for Common Platform 65nm LPe LowK Standard Vt Process

Databook

DesignWare® Logic Libraries
cp65npkslogcassst000f

LPED-0191
Version 3.01B
September 2012

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Registered Trademarks (®)

Synopsys, AEON, AMPS, ARC, Astro, Behavior Extracting Synthesis Technology, Cadabra, CATS, Certify, CHIPit, CoMET, Confirma, CODE V, Design Compiler, DesignSphere, DesignWare, Eclypse, Formality, Galaxy Custom Designer, Global Synthesis, HAPS, HapsTrak, HDL Analyst, HSIM, HSPICE, Identify, Leda, LightTools, MAST, MaVeric, METeor, ModelTools, NanoSim, NOVeA, OpenVera, ORA, PathMill, Physical Compiler, PrimeTime, SCOPE, SiVL, SNUG, SolvNet, Sonic Focus, STAR Memory System, SVP Café, Syndicated, Synplicity, the Synplicity logo, Synplify, Synplify Pro, Synthesis Constraints Optimization Environment, TetraMAX, UMRBus, VCS, Vera, and YieldExplorer are registered trademarks of Synopsys, Inc.

Trademarks (™)

AFGen, Apollo, ASAP, Astro-Rail, Astro-Xtalk, Aurora, AvanWaves, BEST, Columbia, Columbia-CE, Cosmos, CosmosLE, CosmosScope, CRITIC, CustomExplorer, CustomSim, Custom WaveView, DC Expert, DC Professional, DC Ultra, Design Analyzer, Design Vision, DesignerHDL, DesignPower, DFTMAX, Direct Silicon Access, Discovery, Encore, EPIC, Galaxy, HANEX, HDL Compiler, Hercules, Hierarchical Optimization Technology, High-performance ASIC Prototyping System, HSIMplus, i-Virtual Stepper, IC Compiler, IICE, in-Sync, iN-Tandem, Intelli, Jupiter, Jupiter-DP, JupiterXT, JupiterXT-ASIC, Liberty, Libra-Passport, Library Compiler, Macro-PLUS, Magellan, Mars, Mars-Rail, Mars-Xtalk, Milkyway, ModelSource, Module Compiler, MultiPoint, ORAengineering, Physical Analyst, Planet, Planet-PL, Platform Architect, Polaris, Power Compiler, Processor Designer, Raphael, RippledMixer, Saturn, Scirocco, Scirocco-i, SiWare, SPW, Star-RCXT, Star-SimXT, StarRC, Synphony Model System Compiler, System Designer, System Studio, Taurus, TotalRecall, TSUPREM-4, VCSi, VHDL Compiler, VMC, and Worksheet Buffer are trademarks of Synopsys, Inc.

Service Marks (SM)

MAP-in and TAP-in are service marks of Synopsys, Inc.

Third Party Trademark Acknowledgements

SystemC is a trademark of the Open SystemC Initiative and is used under license.

ARM and AMBA are registered trademarks of ARM Limited.

Saber is a registered trademark of SabreMark Limited Partnership and is used under license.

Entrust is a registered trademark of Entrust Inc. in the United States and in certain other countries. In Canada, Entrust is a trademark or registered trademark of Entrust Technologies Limited. Used by Entrust.net Inc. under license.

All other product or company names may be trademarks of their respective owners.

Synopsys, Inc.
700 E. Middlefield Road
Mountain View, CA 94043
www.synopsys.com

Revision History

Note	Links and references to section, table, figure, and page numbers in this table are only assured to be valid for the version in which the change is made.

Date	Document Version	Description
August 2011	B01	Updated for B01 Full GDS library release. Migrated the document to Synopsys standards.
September 2011	B01P1	B01P1 was a patch release with NLDM models only.
December 2011	B01P2	Updated document for B01P2 Full GDS library release.
February 2012	B01P3	Updated document for B01P3 Full GDS library release.
September 2012	3.01B	Changed revision code format as per new release flow. Removed reference to DCAP cells that are not supported in the library.

Contents

1 Introduction

1.1 DesignWare Logic

The DesignWare Logic libraries are based on Synopsys' proprietary and patented NXT routing and cell architecture. This architecture has been proven across a wide range of designs, including high volume production, with over 2500 products.

The NXT architecture's innovative approach to routing and cell design consistently yields smaller designs without sacrificing performance. Synopsys' NXT architecture offers up to a 20% improvement in utilization. Depending on the design needs, this increase in utilization budget can be allocated to attain various goals:

- Higher density without a loss of performance
- Lower power
- Higher performance without an increase in area
- Mask cost reduction

1.2 Logic Libraries

The DesignWare Logic family is comprised of **Metal-Programmable** and **Standard Cell** libraries optimized for different tasks. Within each product family are several architectures that are application-optimized libraries targeted to a variety of market requirements: High-Speed, High-Density, Ultra-High-Density, and Ultra-Low-Power.

Each DesignWare Logic architecture is developed to fit a particular target process and application segment. All cells come with hand optimized circuit design and layout for optimized performance, power consumption, and minimized raw cell area. DesignWare Logic cell libraries are correlated to silicon for maximum accuracy and predictability of results.

The DesignWare Logic Standard Cell libraries may be available with or without taps depending on the specific architecture.

In addition, the cells in these libraries are:

- Optimized for Synthesis through the careful selection of the cell set
- Optimized for Place and Route by improving pin access

- Optimized for Performance and Power by providing a choice of drive strengths.

Note DesignWare Logic Ultra-Low-Power (ULP) Standard Cell libraries provide up to 30% smaller area as compared to conventional standard cell libraries while improving stand-by power dissipation up to 30X by using optional off-set biasing techniques of the unique multi-power rail ULP architecture.

1.3 This Manual

This manual provides information about this DesignWare Logic library product. It is intended for the design engineer using DesignWare Logic libraries to design and develop a system-on-a-chip. It assumes some familiarity with semi-custom and custom chip design.

For any problems with this documentation, please contact the Synopsys' Support Team at https://solvnet.synopsys.com. For technical assistance with the libraries themselves, please see Technical Assistance below.

The rest of this manual consists of the following chapters:

- Chapter 2, "Introduction to NXT Architecture"
- Chapter 3, "Design Guidelines"
- Chapter 4, "Library Characteristics"
- Chapter 5, "Datasheets"
- Chapter 6, "Characterization Methodology"
- Chapter 7, "Modeling Methodology and Validation"

1.4 Technical Assistance

Synopsys' Application Engineers can answer questions about this and other Synopsys products as well as assist with problems. Please contact the Synopsys Support Team at https://solvnet.synopsys.com.

2 Introduction to NXT Architecture

2.1 NXT Architecture

In designs using NXT architecture, the preferred direction for routing Metal2 is parallel to the cell power/ground (P/G) rails; this routing direction is conventionally referred to as **"horizontal"**. The preferred routing directions in the cell the cell layouts naturally extend to routing at the block and chip level as well. As a result, the preferred routing directions in NXT routing are **vertical** for the first layer, **horizontal** for the second layer, **vertical** for the third layer, and so on. To use the conventional notation, this is a **VHVH** routing scheme. This preferred routing scheme for NXT routing is therefore orthogonal to those used with conventional cell libraries, which are routed with an **HVHV** routing scheme. These conventional designs have the first metal layer routed horizontally, or parallel to the P/G rails, the second vertically, and so on.

One of the advantages of NXT routing is that the number of Metal2 tracks that can access the pins on a given cell is increased. This enhances the ability of a routing tool to connect to these pins. Improving pin access in this way minimizes congestion in the vicinity of a cell, since the router is no longer restricted to a single track when connecting to a given pin.

Another feature of this methodology is that the cell (P/G) rails for DesignWare Logic libraries are on Metal2. Routing P/G rails on this layer frees up a significant amount of space on Metal1 for routing, making it possible to further improve pin access, minimize Metal2 blockages, and increase Metal1 porosity in a design. Also, moving P/G rails to Metal2 enables short vertical connections between cells to be routed under the cell P/G rails, on the Metal1 layer rather than on Metal2, further improving routing by reducing the local congestion around cells.

2.2 Optimizing Floorplanning and Placement

When migrating a design that was previously implemented with conventional **HVHV** routing to the NXT architecture's **VHVH** routing scheme, rotating the floorplan can be of benefit. When rotating the floorplan, take care to rotate hard macros in the design by 90 degrees, in addition to rotating the IO constraints. If a legacy hard macro implemented with **HVHV** routing is used in an NXT-routed chip, consider rotating the hard macro so that its routing directions conform to those at the chip level. Likewise, when placing an NXT-routed block on a chip implemented using **HVHV** routing, the NXT block might need to be rotated so that its routing directions coincide with those used at the chip level.

2.3 Optimizing Power Routing

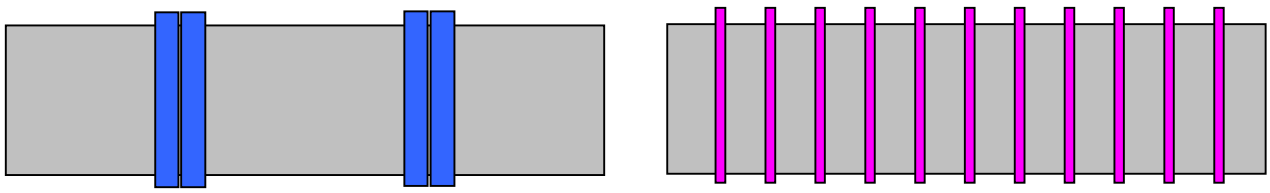
With conventional **HVHV** routing, vertical power straps are placed on Metal2, Metal4 or on both Metal2 and Metal4. Straps on Metal2 prevent cells with Metal2 blockages from being placed under them. They also block access to Metal1 pins of cells which can otherwise be placed under them. As a consequence, cell placement under Metal2 power straps is usually not permitted at all. Vertical Metal2 power and ground straps are therefore usually routed in pairs, with wide spacing between successive P/G strap pairs. On the other hand, in the absence of Metal2 straps, vertical Metal4 straps require stacks of via arrays to be dropped from Metal4 all the way down to Metal1. This causes routing congestion on the intervening layers.

The vertical Metal3 power straps in NXT routing are therefore:

- Relatively narrow (4-12 tracks wide)
- More frequent, to compensate for being narrower
- Evenly spaced, with power and ground straps spaced apart.

This is in contrast to the wide, infrequent, and clustered power straps that are the result of conventional **HVHV** routing. The power routing methodology in NXT routing ensures good pin access for cells placed under the vertical straps and minimizes the worst-case IR drop. Figure 2-1 shows the contrast between evenly spaced power strapping with thin straps on the one hand, and conventional clustered power strapping with wide straps on the other.

Figure 2-1



Conventional (metal2) power straps

NXT-style (metal3) power straps

2.4 Optimizing Routing

DesignWare Logic libraries are designed with **VHVH** routing style in mind. However, the routing direction for Metal1 is assumed to be somewhat more flexible than other layers. As a result, if router permits explicit control of wrong-way routing costs on different layers, consider using a smaller penalty for wrong-way routing on Metal1 than on other metal layers.

3 Design Guidelines

This chapter provides recommendations to keep in mind while using Synopsys' DesignWare Logic libraries. It covers:

3.1 Technology Setup

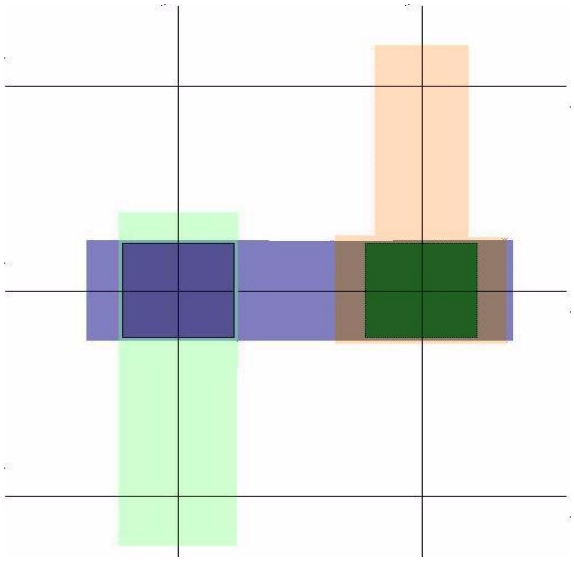
3.1.1 Setup for Synopsys' Apollo/Astro Physical Design Tools

The **.tf** technology rule file for Galaxy/Milkyway describes physical design rules such as minimum width, area, and spacing. However, it does not provide the primary routing directions for different routing layers. Care has been taken to ensure that the **contactCode** statements that define vias in the technology file are encoded correctly. For example, suppose the upper layer of the via is horizontal and the lower layer is vertical. Then **upperLayerEncWidth** is larger than **upperLayerEncHeight**, and **lowerLayerEncHeight** is larger than **lowerLayerEncWidth**. Appropriate directions for the 'first routing layer parallel to power rail' and 'first routing layer perpendicular to power rail' have been specified when generating cell frame views.

3.1.2 Setup for Cadence Design Systems Tools

In a Cadence Design Systems environment, the primary routing directions are specified in the technology section of the LEF file. An optimized technology LEF section is provided with each DesignWare Logic library. Among other things, the following considerations are taken into account. For each pair of consecutive layers there is a default via rule for which the orientations of its metal landing pads conform to their respective routing layers. Care has been taken to ensure that the minimum area rule for intermediate metal layers is satisfied for stacked vias, if they are permitted by the design rules. Likewise, the metal landing pads for non-stacked vias are large enough to satisfy the minimum area rule if the wire segment on an intermediate layer is only 1 grid long, as illustrated in Figure 3-1.

Figure 3-1



The default free track percentages for different routing layers used in SoC Encounter QPlace can be tuned to improve utilization. For libraries using NXT routing, the correct values are as shown in the table below.

Layer	High-Speed Standard Cells
Metal1	10%
Metal2	71%
Metal3 & above	(100%) – (% Used by Power Routing and Hard Macros)

Please note that on-grid via placement is a requirement for ensuring library compliance with technology design rules. Correct placement is achieved by use of commands within the SOC Encounter environments that result in the setting of properties used by Cadence WarpRoute and NanoRoute tools. Required commands are specified below.

```
NanoRouter (SOC Encounter):
            pdi set_option droute_via_on_grid_only {true}
WarpRouter (both SOC Encounter)

SET VAR WRoute.OnGrid.ViaThreshold 2 ;
SET VAR WRoute.Prefer.OnGrid TRUE ;
SET VAR WRoute.OnGrid.Via.Only TRUE ;
SET VAR Wroute.Pin.Access.Mode "upVia";
```

3.2 High-Level Architecture

3.2.1 Power and Electromigration

Power and electromigration are affected by a variety of factors that are not library-specific. To ensure that the design meets performance and reliability requirements, the user must ensure that power consumption, electromigration and slew rate limits are observed.

3.2.2 CMOS Power Consumption

CMOS power consumption is governed by the equation:

$$Power = K * F_c * C * V_{dd}^2$$

where,

K = Switching Factor

C = Load being Switched

F_c = Operating Frequency

V_{dd} = Operating Voltage

3.2.3 Operating Frequency

The operating frequency of a design directly affects its power consumption, with designs that operate at a higher frequency have a higher power consumption. The operating frequency of a design is generally governed by its logic design. As such, the amount of power strapping used within a design will depend on the operating frequency.

DesignWare Logic libraries are designed to ensure robust operation. While there is no limit to the maximum operating frequency, care must be taken to ensure that the electromigration limits of the technology are not exceeded. Please refer to Chapter 4, “Library Characteristics”, for further details.

3.2.4 Switching Factor

Switching factor, or how often the output of a cell switches relative to the frequency of operation, is a highly design-dependent parameter. DesignWare Logic libraries contain power consumption information on a per-cell basis. With this information, industry standard tools can be used to calculate the switching factor for a cell or for a design using a rule of thumb, a heuristic statistical method or a simulation-driven method.

3.3 Routing Directions and Power Grid Routing

3.3.1 Routing Power Straps on Grid

Power straps in NXT routing are more frequent to compensate for their narrower width. Off-grid straps can therefore potentially block a lot of tracks. Power straps must be routed on-grid; otherwise a *k*-track wide strap will block *k+1* tracks. (The term '*k* tracks wide' means that each strap occupies exactly *k* tracks.)

The actual width of a *k*-track wide strap is given by:

$$width = [(k-1) \times pitch] + [pitch - minspacing]$$

where,

pitch is the routing pitch of the strap layer, and

minspacing is the minimum spacing on the metal layer being used for the straps.

For example, the width of a 7-track wide strap with a routing pitch of 2.0 µm, and a minimum spacing of 1.0 µm is

$$[(7-1) \times 2.0] + [2.0 - 1.0] = 13.0\mu m$$

With 0.13 µm and smaller geometries, a larger spacing rule applies for such power straps. If the minimum spacing for the width corresponding to the intended power straps is *wspacing*, and the minimum spacing for single grid wires is *minspacing*, then the power straps must be shrunk on either side by an amount given by *wspacing-minspacing* to satisfy the spacing represented by *wspacing*. This spacing applies between the power straps and the wires that are routed on the tracks adjacent to them. The actual width of *k*-track wide straps in these technologies, therefore, should be

$$width = [(k-1) \times pitch] + [pitch - minspacing - 2 \times (wspacing - minspacing)]$$

For example, consider a routing *pitch* of 2.0 µm, a *minspacing* of 1.0 µm, and a *wspacing* of 1.2µm. The width of a 7-track wide strap would then be

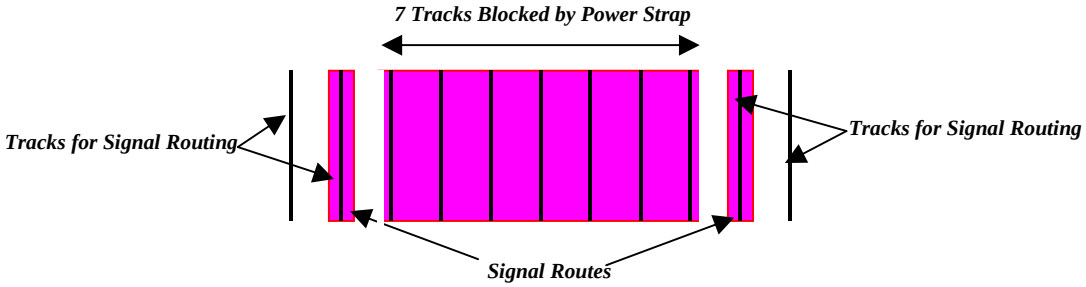
$$[(7-1) \times 2.0] + [2.0 - 1.0 - 2 \times (1.2 - 1.0)] = 12.60\mu m$$

Please refer to "Electrical Characteristics" on page 25, for specific process design rules.

Figure 3-2 illustrates on-grid power strapping. Note that:

- The power strap is placed exactly on the routing grid.
- The power strap has been trimmed on either side to reflect the wider spacing rule, so that both of the adjacent routing tracks are usable for signal routing.

Figure 3-2



3.3.2 Routing Power Straps to Balance Signal Routing Resources

Synopsys' DesignWare Logic libraries are designed with a narrow (1- or 2-track wide) P/G rail within the cell layouts, but the cells are generally designed so that the rail widths can be increased by 1 or 2 additional tracks during block-level P&R. This enables the designer to customize the power distribution at the cell level as needed to minimize the risk of electromigration problems that can arise in cell libraries with narrower cell P/G rails. Consequently, DesignWare Logic libraries can support wider spacing between successive vertical straps of the block- or chip-level power distribution grid.

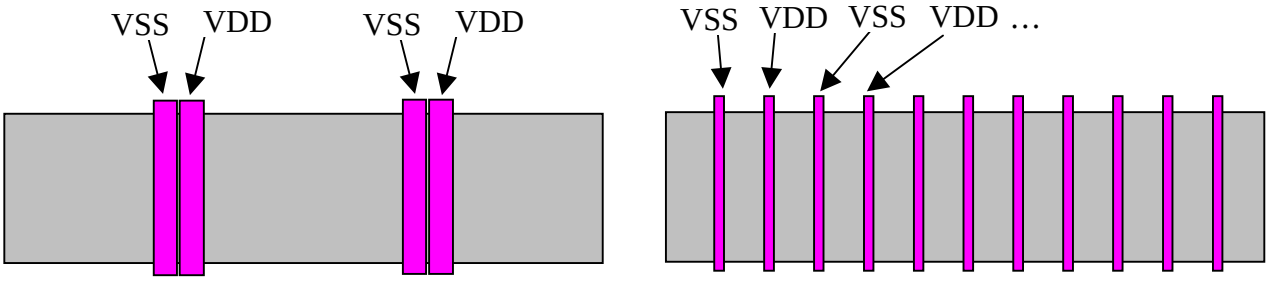
Tuning the power strapping can have a significant impact on die size by reducing the congestion in the bottleneck direction. The demand for vertical and horizontal routing resources varies from netlist to netlist. In addition, demand for horizontal and vertical routing is dependent on the pin location. As a result of this, the horizontal or the vertical direction could become the bottleneck as the case may be. If the power strapping requirements are flexible, power strapping can be tuned to improve routing congestion. For example, if the vertical layers are more congested, the amount of horizontal power strapping can be increased and thereby reduce the amount of vertical strapping.

3.3.3 Using a Conventional Power Routing Scheme

Users can choose to use a conventional power routing scheme, choosing to route power and ground straps close to each other. If choosing to use such a scheme, consider limiting the maximum strap width to 7 tracks.

The minimum strap width and the total amount of power strapping is based on the power grid requirements for a design. Exceeding the strap widths recommended here will also have an impact on routed area.

Figure 3-3



Conventional (metal3) power straps

NXT-style (metal3) power straps

3.4 Special Cells

3.4.1 Clock Driver Cell Usage

3.4.1.1 Clock Drivers, Symmetric Rise and Fall Cells

Clock and global signals need to maintain symmetric ramp times. Synopsys, Inc. provides cells with symmetric ramp times to be used as clock drivers.

These cells are named using the following scheme:

STN_INV_S_<*drive*>

Symmetric rise and fall inverter.

STN_BUF_S_<*drive*>

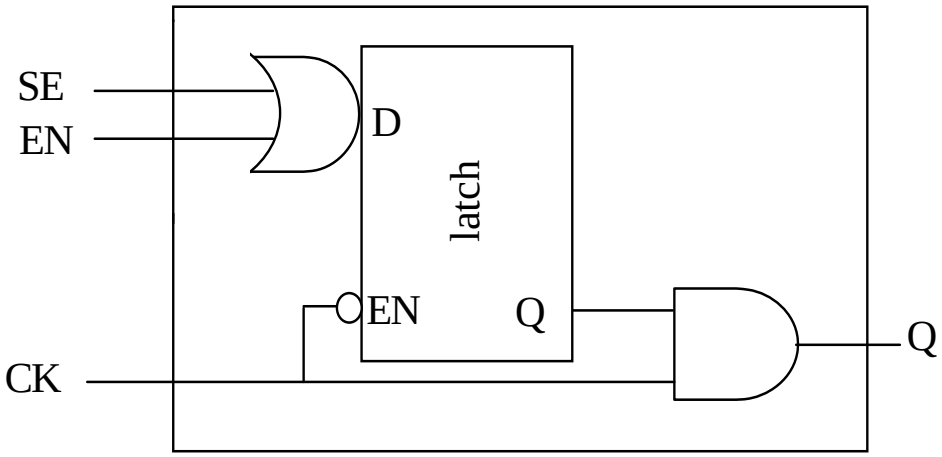
Symmetric rise and fall buffer.

3.4.1.2 Clock Gating Cells

Power dissipation is becoming an increasingly bigger challenge in chip design as the integration level of circuits becomes large while clock frequencies also continue to increase. Synopsys, Inc. provides clock gating cells to help automate the process of power dissipation management.

The latch positive-edge pre-control version of the clock gating cells, as shown in Figure 3-4 below, is supplied with the library.

Figure 3-4

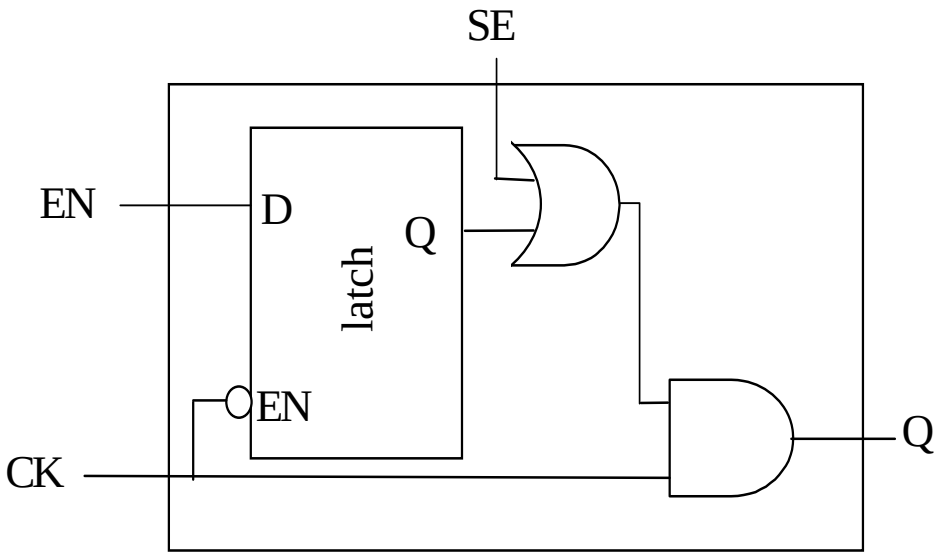


These cells are named using the following scheme:

STN_CKGTPLT_<*drive*>

The latch positive-edge post-control version of the clock gating cells, as shown in Figure 3-5 below, is also supplied with the library.

Figure 3-5



3.4.2 Delay Cell Usage

This DesignWare library includes delay cells to help to ensure that hold time requirements are met for difficult sections of the design. These delay cells have been designed with a delay which is a linear function of two-input NAND gate on which it is based.

These cells are named using the following scheme, and have the delay assumptions mentioned:

STN_DEL_R<*delayRatio*>_<*drive*>

Delay cells are designed with relative delay in mind. The baseline delay uses as a reference a single (1X) drive 2-input NAND gate. The actual delay of the delay cell is the baseline delay multiplied by the delay ratio.

3.4.3 Filler Cell Usage

To ensure that the design passes DRC and LVS checks, filler cells must be included when users stream out GDS for final verification. Users can insert filler cells into the design after all placement-related operations are complete, and before final signal routing is carried out. For Ultra High Density libraries, user must insert filler cells before the final signal routing.

The filler cells provided in this DesignWare Logic library come in several sizes, and range from a cell covering a single unused placement site up to cells covering multiple unused sites. These filler cells are named according to the number of placement sites that they “fill”, using the following scheme:

STN_FILL<*sites*>

The **<sites>** component of the name represents the number of grid positions the number of base cell widths covered by the cell, with the value of **<sites>** ranging from 1 to 64.

When adding filler cells to the design, start with the largest filler cell and work down to the single-site filler cell. This will ensure that all the unused sites are filled with the smallest possible number of filler cell instances.

In addition, when using SoC Encounter, the variable

```
PLACE.LLC.IGNORE.LAYER.2
```

must be set to **TRUE** before filler cells are added, and set to **FALSE** once the fill operations are completed. This is necessary in order to ensure proper gap filling within the array.

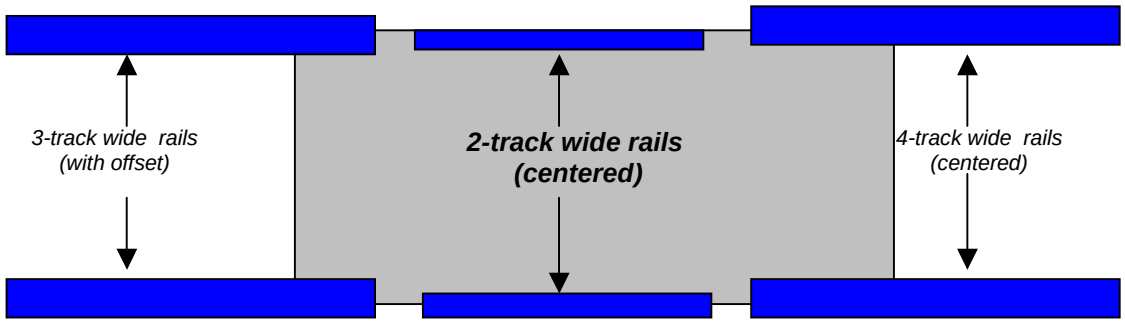
Here is an example sequence of SoC Encounter commands for filler cell insertion:

```
$filler_list=(list of all filler cells in the library starting with widest one)
  foreach cell_name $filler_cell_list {
     addFiller -cell $cell_name -prefix FILL -fillBoundary
  }
```

3.4.4 Variable Width Power and Ground Cell Rails

This DesignWare Logic library architecture permits the use of either 2-track, 3-track, or 4-track wide cell power and ground (P/G) rails, as needed for the application. Figure 3-6, below, illustrates the three different types of rails. All cells in the library have 2-track wide P/G rails; this is the default rail width.

Figure 3-6



Default-width P/G rails can be created as usual using 'follow pins' routing of the P/G pins in the cells. However, wider-than-minimum rails cannot be programmed in this way.

There are two ways to create wider-than-minimum rails. The first is to create them just as power grid straps are created; however, it's very important to ensure that the rail straps are precisely on-grid, and that they cover the corresponding P/G pins in the cells precisely as illustrated in above. Any deviation from this will result in a significant loss in routability, and possibly block access to signal pins on some cells.

All DesignWare Logic libraries provide a second, simpler, rail-programming methodology, which ensures that user can easily create rails of any supported width in a consistent and tool-independent manner. Each library contains several sets of special filler cells named.

STN_RAIL<*railWidth*>T[U/D]<*sites*>

where <railWidth> represents the P/G rail width, in tracks, defined by that filler cell set; [U/D] is an optional character indicating that the rail is offset up or down from the cell row; and <sites> describes the width, in placement sites, of the cell.

This library contains two sets of these special filler cells, **STN_RAIL3TU*** and **STN_RAIL3TD***. The first is used to define 2 or 4 track wide P/G rails of width equal to 1 track, the second is used to define 3-track wide rails, which must be offset from the top/bottom cell boundaries. Refer to "Filler Cell Usage" on page 18, for a complete listing of the available rail filler cells. The P/G rails for the design can be routed as follows:

- After placing the hard macros and defining the standard cell rows, but before placing the cells, fill all cell rows with the rail cell set(s) corresponding to the desired P/G rail width, using the procedure for filler cells described in "Filler Cell Usage" on page 18, above. For 2- or 4-track wide rails, use the STN_RAIL2T* or STN_RAIL4T* rail cell set(s), respectively. For 3-track wide rails, fill all flipped rows with STN_RAIL3TU*, and the remaining rows with STN_RAIL3TD*.
- Create the cell P/G rails as usual, using the P/G pin widths of the STN_RAIL* cell set with which the cell rows have been filled.
- Delete all STN_RAIL* cells inserted above.
- Perform cell placement and routing.

3.4.5 Tap Cell Placement

Cells in this library do not contain any substrate or well taps. A tap cell must be placed during floorplanning at locations and a frequency that satisfies design objectives as well as process design rule requirements. It consists of an nwell tap to VDD and a pwell tap to VSS. The tap is 2-grids wide.

The tap cell name is:

STN_TAP_DS - VSS and VSS connected to Nwell/Pwell

3.4.6 ENDCAP Cell and Triple Well Design

Triple well design requires the addition of the T3 (deep NW) implant layer covering the entire logic block, overlapped by an NW layer that forms a continuous ring around the entire logic block. This library contains both a dual well version and a triple well version of the library gds file. The cells of each library are identical except for the addition of the T3 layer to the cells in the triple well version. Dual well cells and triple well cells cannot be mixed in a logic block.

The following requirements must be satisfied by any chip or by a stand alone standard cell block that may be placed on a chip:

1. Fill the top and bottom rows of any standard cell block with top/bottom end cap cells provided in the library. These end cap cells ensure continuous NW ring around logic block to achieve PW isolation, plus ensure proper NW overlap of T3 (deep NW layer). This cell also contains the NW tiedown diode required by design rules.
2. The highest and lowest rows used for standard cells (logic rows) must be such that the NW in standard cells points outward towards the top and bottom of the block
   - The highest logic row is the one just below the top row. This row should be in the normal orientation
   - The lowest logic row is the 2nd row from the bottom. This row should be in the flipped orientation
3. In order to satisfy 2, the following constraints must be met when sizing a standard cell block
   - The height of the block must be an even number of rows
   - The top row (endcaps) must be in the flipped orientation, and the bottom row (endcaps) must be in the normal orientation
4. Exception: If all instances of a standard cell block have a standard cell row abutting its top/bottom row, then the block does not need top/bottom end cap cells. The abutting standard cell row may either belong to the parent block, or to any other standard cell block.
5. End-of-row end cap cells must be placed at the left and right edge of any row. These end cap cells help form a continuous NW ring around T3
   - Right endcap is placed in normal orientation
   - Left endcap is placed in the flipped orientation
6. An inner vertex end cap cell must be placed at any concave (inner) vertex of a non-rectangular block

- Right endcap is placed in normal orientation
- Left endcap is placed in the flipped orientation

Endcap cell names are:

STN_CAPLR10 -- End-of-row (left/right) endcap

STN_CAPNW1,2,4,8,16,32,64 -- Top/Bottom endcap must butt up against NW side of logic cell

STN_CAPCRNI10 -- Inner vertex endcap for non-rectangular blocks

The end-of-row endcap can also be used in dual well designs as an NW tiedown cell.

Figure 3-7 Triple Well Blocks with Endcaps

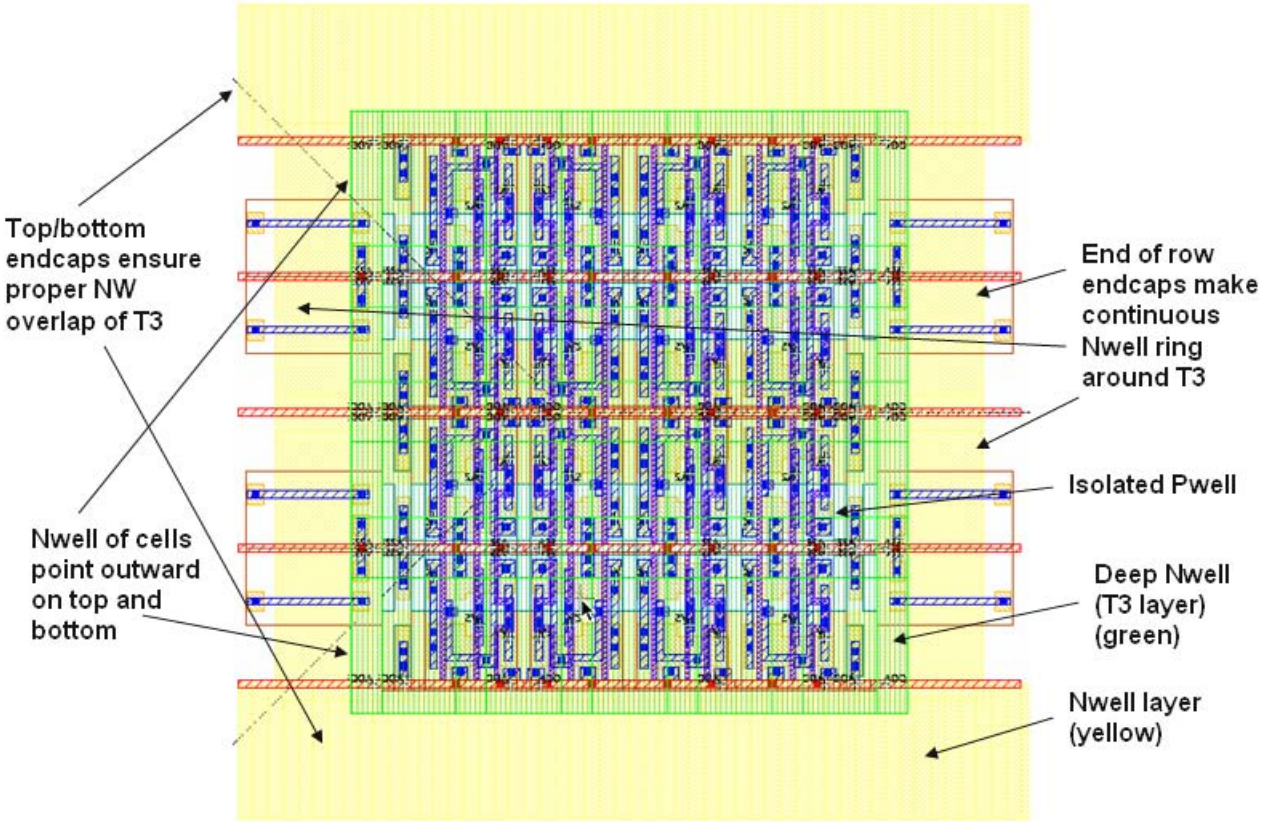
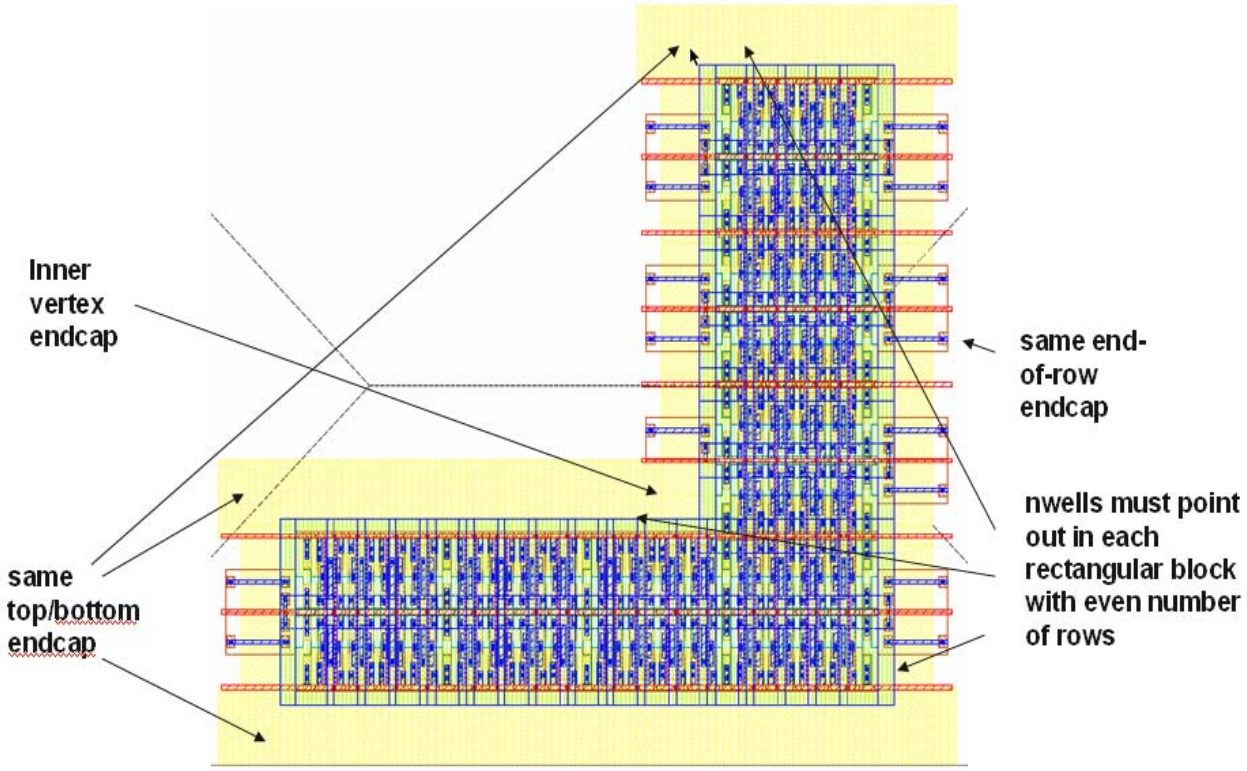


Figure 3-8 Non-Rectangular Triple Well Block



3.4.7 Decoupling Cells

Decap cells are used to reduce power supply noise. Power supply noise is generated due to short circuit currents caused by simultaneous switching of device on chip.

The library contains decoupling capacitors of the following type:

STN_DCAP (4/8/16/32/64) - Decap cells

4 Library Characteristics

This chapter describes the characteristics of the *DesignWare High Speed Standard Cell Tapless Logic Library Full for Common Platform 65nm LPe LowK Standard Vt Process*. It covers

4.1 Physical Characteristics

Table 4-1

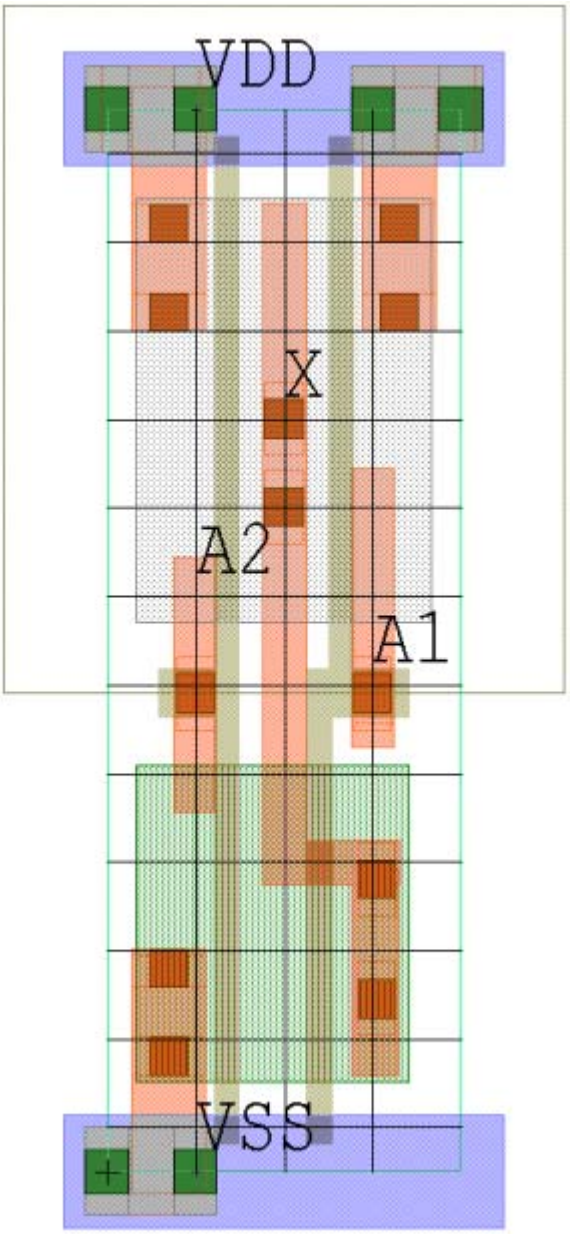
Characteristic	Value	Comment
X Grid	0.20 µm	No offset
Y Grid	0.20 µm	0.100 µm offset
Height	2.400 µm	12 M2 Routing Tracks
Width	Increments of 1 X-grid(s).	
Symmetry	X Y	
Power Bus Width	0.26 µm	M2 - shared between cell rows on cell boundary

Note In order to achieve maximum routing density in this technology, most of the Metal1 input pins pass DRC only when the router connects the input pin at the chip level. Therefore, it is possible for DRC to fail if it is run on an individual cell, but this will not be a problem when using the library in a design.

4.1.1 Example

Figure 4-1 shows an example of **STN_ND2_1**.

Figure 4-1



4.2 Electrical Characteristics

Table 4-2 provides a list of all PVTs supported in the library. However, only a subset of PVTs may be available to the user depending upon the library options licensed.

Table 4-2

Corner Name	Process	Temperature (C)	VDD (V)
ff1p1vn40c	ff	-40	1.1
ff1p1vn55c	ff	-55	1.1
ff1p32vn40c	ff	-40	1.32
ff1p32vn55c	ff	-55	1.32
ff1p1v125c	ff	125	1.1
ff1p32v125c	ff	125	1.32
ss0p9v125c	ss	125	0.9
ss0p9vn40c	ss	-40	0.9
ss0p9vn55c	ss	-55	0.9
ss1p08v125c	ss	125	1.08
ss1p08vn40c	ss	-40	1.08
ss1p08vn55c	ss	-55	1.08
tt1p0v25c	tt	25	1.0
tt1p2v25c	tt	25	1.2

Extra care should be taken when deviating from these criteria as designs might behave differently compared to what is documented here.

4.3 Electromigration Limit

The user must ensure that electromigration limits of the technology are not exceeded. DesignWare libraries are designed to ensure that the electromigration limit, for physical features in the cell, will be met if the following condition is satisfied.

Maximum cell output signal slew rate must be observed with the following relationship

$$\mathrm{MaximumSlew} \leq [1/(\mathrm{switchingfactor}) \times 1/(\mathrm{frequency}) \times 0.10]$$

4.4 Loading and Slew Rate

Excessive loading of signal nets can cause unpredictable timing behavior and excessive power consumption. DesignWare Logic libraries are designed to operate within specific slew-rate and load ranges.

Using cells in design scenarios that violate their operating ranges can cause problems for high toggle-rate and timing-critical nets.

DesignWare Logic libraries avoid such problems by incorporating the following in their design:

- Cell are designed with P/G rail fidelity in mind, taking electromigration into account for high-drive and clock-related cells:
  - P/G contact-to-via ratio set to 1:1.

    This ensures that the via does not become an electromigration bottleneck for high-speed designs. Note that within a given cell, this requires that there be twice as many vias as contacts, since vias are on the top and bottom cell boundaries, and therefore potentially shared between cells in vertically-adjacent rows.
  - Metal-segment widths increased as needed to meet electromigration specs.

    This ensures that metal segments that experience pulsed-DC current will not fail at the maximum frequency and load conditions specified for the library.
- Libraries undergo correct-by-construction cell characterization, which limits the maximum slew across the entire library.

  The maximum load for each output pin is determined from the maximum slew.

  The maximum slew, which varies for each PVT corner, is defined to be the output slew of a 1X inverter driving 64 1X inverters.

  Here are the maximum slews for this library (measured between 10% and 90% of VDD):

Table 4-3 Maximum Slew Table

PVT	Value (ps)
f1p1vn40c	1100
ff1p1vn55c	595
ff1p32vn40c	587
ff1p32vn55c	736
ff1p1v125c	1300
ff1p32v125c	490
ss0p9v125c	3999
ss0p9vn40c	2333
ss0p9vn55c	954
ss1p08v125c	1512
ss1p08vn40c	898
ss1p08vn55c	1095

Table 4-3 Maximum Slew Table

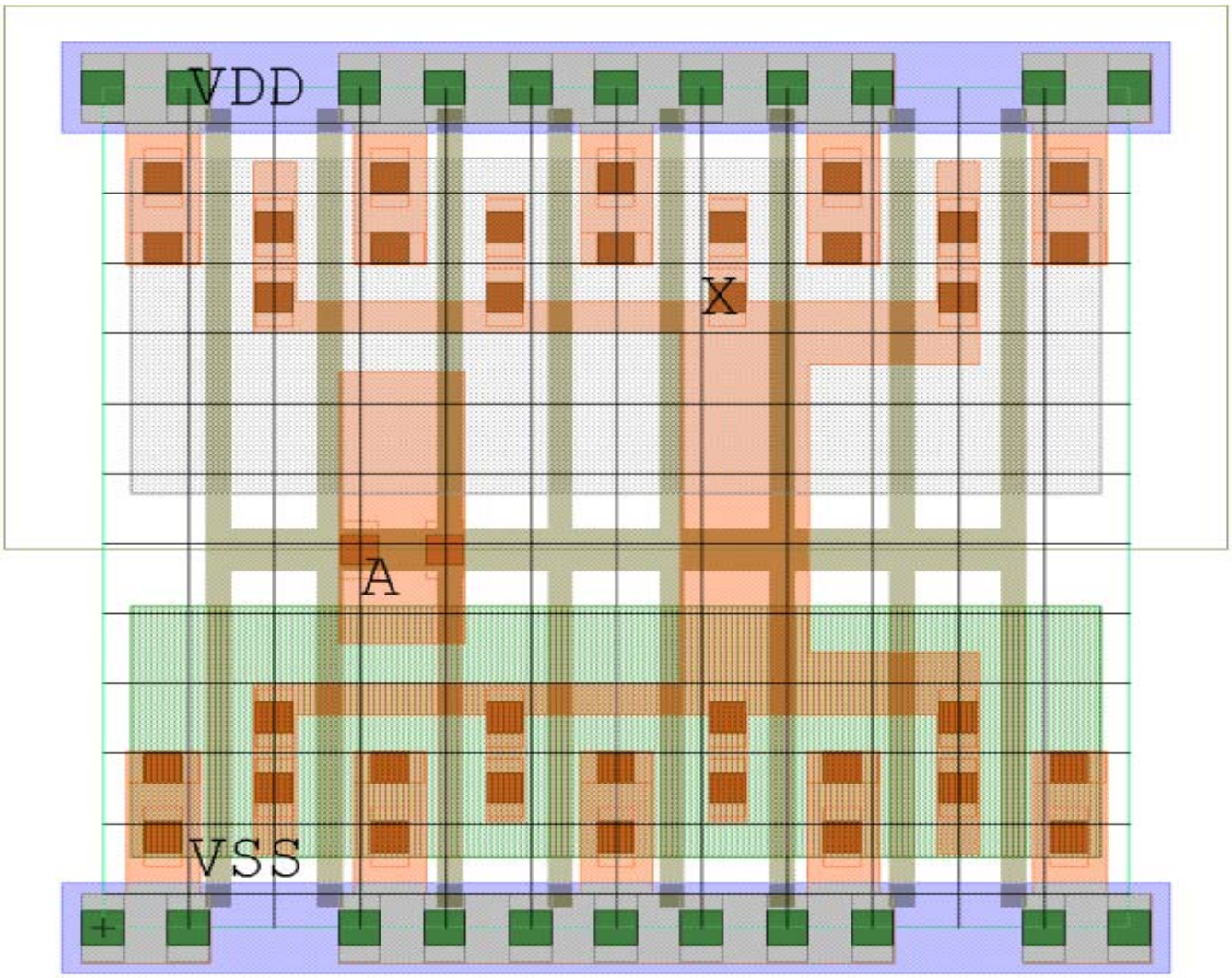
PVT	Value (ps)
tt1p0v25c	1982
tt1p2v25c	595

With the maximum slew rate set, the only determining factors are the operating frequency and switching factor, which users can control.

- Cell architecture designed for power rail fidelity to ensure that cell design takes electromigration into account for high drive and clock-related cells:
- Increased metal width connecting P and N transistors.

Figure 4-2 shows an example cell, **STN_INV_12**.

Figure 4-2



5 Datasheets

DesignWare Logic libraries are designed to optimize the metrics for area, power, and routability of ASIC designs. The *DesignWare High Speed Standard Cell Tapless Logic Library Full for Common Platform 65nm LPe LowK Standard Vt Process* contains 1127 cells and 205 functions.

This chapter provides data on the library and the cells it contains. It includes:

5.1 Cell Selection Philosophy

5.1.1 Synthesis Friendly

DesignWare Logic libraries have functional continuity, which is essential for optimal technology mapping during logic synthesis. This requires that whenever a complex gate is available in the library, gates for its Boolean sub-expressions are also available. This library contains

- Cells for different types of AOI, OAI, AO, and OA functions which can be implemented more efficiently as a single cell than a combination of individual AND, OR, NAND, NOR, INV gates.
- Several gates with bubble (inverted) inputs which can be implemented more efficiently as a single gate than using a combination of a simple gate with inverters to invert individual inputs.
- No complex flip-flops such as Set-Reset and JK, since modern synthesis tools prefer to build the equivalent logic with simpler building blocks.
- Numerous drive-strengths for frequently used functions to help with timing, as well as to further minimize power consumption.
- Flip-flops and latches with many drive-strengths to help with timing.

5.1.2 Place and Route Friendly

Better pin access is a basic feature of all DesignWare Logic libraries. Other features of these libraries that produce much better place and route results, include

- A minimum number of large cells, such as 8-bit counters, 4-bit latches. The absence of such gates enables better optimization during placement, and minimizes perturbations when handling engineering change orders (ECOs) or performing netlist changes for timing correction.
- A minimum number of gates (other than buffers and inverters) with very high drive-strengths, since they tend to encourage high fan-out nets during synthesis, causing wiring congestion. When such gates are required for performance considerations, their use is not recommended for initial random-logic synthesis.
- Carefully optimized pin-placements for all library cells, to provide maximum pin access, and consequently minimum local congestion.
- A minimal amount of Metal2 blockage. This minimizes restrictions on cell placement, leading to higher utilization and lower local congestion.
- Carefully adjusted wiring pitches for all metal layers, to optimize overall chip area, routability and wire length.

5.1.3 Optimized Performance and Power

The DesignWare Logic library provides a range of drive strength options for the more frequently used cells, such as inverters and buffers.

5.2 Cell Function Summary

This DesignWare Logic library consists of cells with a wide range of functions to fulfill the design needs:

- Standard combinational logic gates. Such as AND, OR, NAND, NOR.
- Standard combinational logic gates with inverted inputs.
- Exclusive-OR and exclusive-NOR.
- MUX, adders and comparators.
- AOI (and-or-inverted) and OAI (or-and-inverted) gates, and their counterparts with bubble inputs.
- AO (AND-OR) and OA (OR-AND) gates.
- Inverters, buffers, and delay elements, and clock buffers.
- Numerous latches and flip-flops, including scan flip-flops.
- High performance cells for use with Synopsys Module Compiler.

5.3 Naming Conventions

5.3.1 Cell Naming Syntax

<*prefix*>_<*baseName*>[_<*extension*>][_<*drive*>]

<prefix>

This is a three-letter prefix identifying the type of library the cell is a part of as well as the V_t corner.

<baseName>

This is an alphanumeric string conveying the function in commonly used form.

<extension>

This optional extension to the name appears where the general function of the cell is the same but the specific implementation requires a unique topology or other consideration.

<drive>

This optional suffix gives the drive strength, the driving capability of the cell, in the number format.

5.3.2 Examples

A cell with the name

```
STN_INV_S_1
```

is a standard cell symmetric rise and fall inverter with a 1x drive strength.

5.4 Datasheets

The following types of cells are included in this library:

- Combinational Cells
  - Buffers/Inverters
  - Simple Gates
  - Simple Gates
  - AO/OA Gates
  - Complex Gates
  - Adders
  - Multiplexers
- Sequential Cells
  - Flip-Flops
  - Flip-Flops with Scan
  - Latches
- Special Cells
  - Delay Buffers
  - Clock-Gater Cells
  - Miscellaneous Cells

The following sub-sections provide a list of the cells included in this library. Users can click on the name of each cell to view the associated Datasheet.

Note * refers to the drive option.

5.4.1 Combinational Cells

5.4.1.1 Buffers/Inverters

Table 5-1 Buffers/Inverters

Function	Drive Options	Description
STN_BUF_*	1,1.5,2,3,4,6,8,10,12,16,20,24	Non-inverting buffer
STN_BUF_AS_*	0.5,1,1.5,2,3,4,5,6,8,10,12,16	Symmetric rise/fall time buffer, antenna diode on input
STN_BUF_D_*	1,2,3,4,6,8,12,16	Non-inverting buffer
STN_BUF_S_*	1,2,3,4,6,8,12,16	Symmetric rise/fall delay buffer
STN_INV_*	0.5,0.65,0.8,1,1.25,1.5,2,2.5,3,4,5,6,8,10,12,16,20,24,32	Inverter
STN_INV_AS_*	0.5,1,1.5,2,3,4,5,6,8,10,12,16	Symmetric rise/fall time inverter, antenna diode on input
STN_INV_S_*	0.5,1,1.5,2,3,4,6,8,12,16,32	Symmetric rise/fall time inverter

5.4.1.2 Simple Gates

Table 5-2 Simple Gates

Function	Drive Options	Description
STN_AN2_*	1,1.5,2,3,4,5,6,8,12,16,24	2-Input AND
STN_AN2_DG_*	1,2,3,4,8,16	2-Input AND
STN_AN2_S_*	0.5,1,1.5,2,3,4,8,16	2-Input AND, symmetric rise/fall
STN_AN3B_*	0.5,1,2,4,8	3-Input AND (A inverted input)
STN_AN3_*	0.5,1,2,4,8,16	3-Input AND
STN_AN3_S_*	1,2,4,8	3-Input AND, symmetric rise/fall
STN_AN4B_*	0.5,1,2,4,8	4-Input AND (A inverted input)
STN_AN4_*	0.5,1,1.5,2,4,8	4-Input AND
STN_AN4_S_*	1,1.5,2,4,8	4-Input AND, symmetric rise/fall
STN_AN5_*	1,2,4,6	5-Input AND

Table 5-2 Simple Gates

Function	Drive Options	Description
STN_AN6_*	1,2,4,8	6-Input AND
STN_ND2B_*	1,2,3,4,6,8,12,16	2-Input NAND (A inverted input)
STN_ND2B_S_*	0.5,1,2,4,8,12	2-Input NAND (A inverted input), symmetric rise/fall
STN_ND2B_V1DG_*	1,2,4,8	2-Input NAND (A inverted input)
STN_ND2B_V1_*	1,2,3,4,6,8,16	2-Input NAND (A inverted input)
STN_ND2_*	0.5,0.65,0.8,1,1.5,2,3,4,6,8,12,16	2-Input NAND
STN_ND2_G_*	0.65,1,2,3,4,6,8,12,16,24	2-Input NAND
STN_ND2_S_*	0.5,0.65,0.8,1,1.5,2,3,4,6,8,12,16	2-Input NAND
STN_ND2_T_*	0.5,1,1.5,2,3,4,5,6,8,16	2-Input NAND
STN_ND3B_*	0.5,1,2,4,8	3-Input NAND (A inverted input)
STN_ND3B_V1DG_*	1,2,4,8	3-Input NAND (A inverted input)
STN_ND3_*	0.5,1,2,3,4,6,8	3-Input NAND
STN_ND3_S_*	0.5,1,2,4,8,12	3-Input NAND
STN_ND3_T_*	0.5,0.65,0.8,1,1.5,2,3,4,6,8,12,16	3-Input NAND
STN_ND4B_*	1,2,4,8	4-Input NAND (A inverted input)
STN_ND4_*	1,2,4,8	4-Input NAND
STN_ND4_S_*	0.5,1,1.5,2,3,4,6,8	4-Input NAND
STN_NR2B_*	1,2,3,4,6,8,12,16	2-Input NOR (A inverted input)
STN_NR2B_V1DG_*	1,2,3,4,6,8,12,16	2-Input NOR (A inverted input)
STN_NR2B_V1_*	1,2,4,8	2-Input NOR (A inverted input)
STN_NR2_*	0.5,0.65,0.8,1,2,3,4,6,8,16	2-Input NOR
STN_NR2_G_*	0.5,0.65,0.8,1,2,3,4,6,8,12,16,24	2-Input NOR
STN_NR2_S_*	0.5,0.65,0.8,1,2,3,4,5,8	Symmetric rise/fall time 2-input NOR
STN_NR2_T_*	0.5,1,1.5,2,3,4,5,6,8,12,16	Symmetric rise/fall time 2-input NOR
STN_NR3B_*	1,2,4	3-Input NOR (A inverted input)
STN_NR3B_DG_*	3,8	3-Input NOR (A inverted input)
STN_NR3_*	0.5,0.65,0.8,1,2,4,8	3-Input NOR
STN_NR3_G_*	1,2,3,4,8	3-Input NOR

Table 5-2 Simple Gates

Function	Drive Options	Description
STN_NR3_T_*	0.5,0.65,0.8,1,2,3,4,5,6,8,12	3-Input NOR, symmetric rise/fall
STN_NR4B_*	1,2,4	4-Input NOR (A inverted input)
STN_NR4_*	0.5,1,2,3,4,6,8	4-Input NOR
STN_OR2_*	1,1.5,2,2.5,3,4,5,6,8,10,12,16,24	2-Input OR
STN_OR2_DG_*	1,2,3,4,6,8	2-Input OR
STN_OR3B_*	0.5,1,2,4,8,12	3-Input OR (A inverted input)
STN_OR3_*	1,2,3,4,6,8	3-Input OR
STN_OR4B_*	1,2,4,8	4-Input OR (A inverted input)
STN_OR4_*	1,2,4,6,8	4-Input OR
STN_OR5_*	1,2,4	5-Input OR

5.4.1.3 AO/OA Gates

Table 5-3 AO/OA Gates

Function	Drive Options	Description
STN_AO21B_*	1,2,4,6,8,16	One 2-input AND into 2-input OR with inverted B
STN_AO21_*	1,2,4,8	One 2-input AND into 2-input OR
STN_AO21_DG_*	1,2,4	One 2-input AND into 2-input OR
STN_AO221_*	0.5,1,2,4	Two 2-input ANDs into 3 input OR
STN_AO2222_*	1,2,4	Four 2-input ANDs into 4-input OR
STN_AO222_*	1,2,4,6	Three 2-input ANDs into 3-input OR
STN_AO22_*	1,2,4,8	Two 2-input ANDs into 2-input OR
STN_AO22_DG_*	1,2,3,4,8,10	Two 2-input ANDs into 2-input OR
STN_AO2BB2_*	0.5,1,2,4,8	One 2-input NOR, and one 2-input AND, into a 2-input OR
STN_AO2BB2_DG_*	1,2,4,8	One 2-input NOR, and one 2-input AND, into a 2-input OR
STN_AO31_*	0.5,1,2,4,8	One 3-input AND into 2-input OR
STN_AO32_*	0.5,1,2,4	One 3-input AND one 2-input AND into 2-input OR
STN_AO33_*	1,2,4	Two 3-input ANDs into a 2-input OR

Table 5-3 AO/OA Gates

Function	Drive Options	Description
STN_AOA211_DG_*	1,2,4,8	One 2-input AND into 2-input OR into 2-input AND
STN_AOAI211_*	0.5,0.75,1,2,3,4,6,8	One 2-input AND into 2-input OR into 2-input NAND
STN_AOAI211_G_*	0.5,0.75,1,2,3,4	One 2-input AND into 2-input OR into 2-input NAND
STN_AOI211_*	0.5,0.75,1,2,3,4,6,8,12	One 2-input AND into 3-input NOR
STN_AOI211_G_*	0.5,0.75,1,2,3,4,5,8	One 2-input AND into 3-input NOR
STN_AOI21B_*	0.5,1,2,3,4,8	One 2-input AND into 2-input NOR (other input inverted)
STN_AOI21_*	0.5,0.75,1,2,3,4,6,8,16	One 2-input AND into 2-input NOR
STN_AOI21_G_*	0.5,0.75,1,2,3,4,6,8	One 2-input AND into 2-input NOR
STN_AOI21_S_*	0.5,1,2,4,8	One 2-input AND into 2-input NOR
STN_AOI21_T_*	0.5,1,1.5,2,3,4,6,8,12	One 2-input AND into 2-input NOR
STN_AOI221_*	0.5,1,2,4	Two 2-input ANDs into 3-input NOR
STN_AOI222_*	0.5,1,2,3,4,8	Three 2-input ANDs into 3-input NOR
STN_AOI22_*	0.5,0.75,1,2,3,4,5,6,8,12	Two 2-input ANDs into 2-input NOR
STN_AOI22_S_*	1,1.5,2,4,8	Two 2-input ANDs into 2-input NOR
STN_AOI22_T_*	0.5,1,1.5,2,3,4,6,8	Two 2-input ANDs into 2-input NOR
STN_AOI311_*	1,2,4	One 3-input AND into 3-input NOR
STN_AOI31_*	0.5,0.75,1,2,3,4,6,8,12	One 3-input AND into 2-input NOR
STN_AOI31_G_*	1,2,4,8	One 3-input AND into 2-input NOR
STN_AOI31_T_*	0.5,1,2,4,8	One 3-input AND into 2-input NOR
STN_AOI32_*	1,2,4,8	3-input AND and 2-input AND into 2-input NOR
STN_AOI33_*	1,2,4,6	Two 3-input ANDs into a 2-input NOR
STN_OA21B_*	0.5,1,2,3,4,8	One 2-input OR into 2-input AND (1 inverted input)
STN_OA21_*	1,2,4,8	One 2-input OR into 2-input AND
STN_OA222_*	0.5,1,2,3,4	Three 2-input ORs into a 3-input AND
STN_OA22_*	1,2,4,6,8	Two 2-input ORs into 2-input AND
STN_OA22_DG_*	1,2,4,8	Two 2-input ORs into 2-input AND
STN_OA2BB2_*	0.5,1,2,3,4,8	One 2-input OR with inverted inputs + 2-input OR into 2-input AND

Table 5-3 AO/OA Gates

Function	Drive Options	Description
STN_OA2BB2_DG_*	1,2,3,4,8	One 2-input OR with inverted inputs + 2-input OR into 2-input AND
STN_OA31_*	0.5,1,2,4,8	One 3-input OR into 2-input AND
STN_OA32_*	1,2,4	One 3-input OR + one 2-input OR into 2-input AND
STN_OA33_*	0.5,1,1.5,2,3,4	Two 3-input ORs into a 2-input AND
STN_OAI2111_*	0.5,1,2,4	One 2-input OR into 4-input NAND
STN_OAI211_*	0.5,1,2,3,4,8	One 2-input OR into 3-input NAND
STN_OAI21B_*	0.5,1,2,3,4,6,8	One 2-input OR into 2-input NAND (other input inverted)
STN_OAI21_*	0.5,0.75,1,2,3,4,5,6,8,16	One 2-input OR into 2-input NAND
STN_OAI21_G_*	0.5,1,2,4,6,8,12,16	One 2-input OR into 2-input NAND
STN_OAI21_S_*	0.5,1,1.5,2,3,4,8	One 2-input OR into 2-input NAND
STN_OAI21_T_*	0.5,1,2,3,4,6,8	One 2-input OR into 2-input NAND
STN_OAI221_*	0.5,1,2,4	Two 2-input ORs into 3-input NAND
STN_OAI222_*	0.5,0.75,1,2,3,4	Three 2-input ORs into 3-input NAND
STN_OAI22_*	0.5,0.75,1,2,3,4,6,8,12	Two 2-input ORs into 2-input NAND
STN_OAI22_S_*	0.5,1,2,3,4,8	Two 2-input ORs into 2-input NAND
STN_OAI22_T_*	0.5,1,1.5,2,3,4,8	Two 2-input ORs into 2-input NAND
STN_OAI31_*	0.5,0.75,1,2,3,4,6,8,12	3-input OR into 2-input NAND
STN_OAI31_G_*	0.5,1,2,4	3-input OR into 2-input NAND
STN_OAI31_T_*	0.5,1,2,4,8	3-input OR into 2-input NAND
STN_OAI41_*	1,2,4	4-input OR into 2-input NAND
STN_OAO211_DG_*	1,2,4,8	One 2-input OR into 2-input AND into 2-input OR
STN_OAOI211_*	0.5,1,2,3,4,8,12	One 2-input OR into 2-input AND into 2-input NOR
STN_OAOI211_G_*	1,2,4,8	One 2-input OR into 2-input AND into 2-input NOR

5.4.1.4 Complex Gates

Table 5-4 Complex Gates

Function	Drive Option	Description
STN_EN2_*	0.5,1,2,3,4,5,6,8	2-Input exclusive NOR
STN_EN2_DG_*	1,8,2,4,16	2-Input exclusive NOR
STN_EN2_S_*	0.5,1,2,3,4,6,8	2-Input exclusive NOR
STN_EN3_*	0.5,1,1.5,2,3,4,6	3-Input exclusive NOR
STN_EN3_DG_*	1,2,3,4,8	3-Input exclusive NOR
STN_EO2_*	0.5,1,2,3,4,5,6,8	2-Input exclusive OR
STN_EO2_DG_*	1,2,4,16,8	2-Input exclusive OR
STN_EO2_G_*	1,2,4,8	2-Input exclusive OR
STN_EO2_S_*	1,2,5,6,0.5,3,4,8	2-Input exclusive OR, symmetric rise/fall
STN_EO3_*	1,2,3,4,6	3-Input exclusive OR
STN_EO3_DG_*	1,2,4,8	3-Input exclusive OR
STN_EO4_DG_*	1,2,3,4	4-Input exclusive OR
STN_MAJ3_*	0.5,1,1.5,2,4,6	3-input majority
STN_MAJI3_*	1,2,3,4,5,6,8	3-input majority, inverted
STN_MAJI3_T_*	1,2,3,4,6	3-input majority, inverted

5.4.1.5 Adders

Table 5-5 Adders

Function	Drive Option	Description
STN_ADDF_*	0.5,1,2,4	Full adder
STN_ADDF_D_*	1,2,4,6	Full adder
STN_ADDF_P1_*	1,2,3,4	Full adder
STN_ADDH_*	0.5,1,2,4	Half adder
STN_ADDH_D_*	1,2,3,4,6	Half adder

5.4.1.6 Multiplexers

Table 5-6 Multiplexers

Function	Drive Option	Description
STN_MUX2AN2_DG_*	1,2,4,8	2-1 multiplexer w/and2 input
STN_MUX2OR2B_DG_*	1,2,4,8	2-1 multiplexer w/or2b input
STN_MUX2_*	0.5,1,2,3,4,6,8	2-1 multiplexer
STN_MUX2_DG_*	1,2,4,8,12,16	2-1 multiplexer
STN_MUX2_G_*	1,2,3,4,6,8,12,16	2-1 multiplexer
STN_MUX2_S_*	0.5,1,2,4,8	2-1 multiplexer, symmetric rise/fall
STN_MUX3_DG_*	1,2,3,4,6	3-1 multiplexer
STN_MUX4_*	0.5,1,2,3,4	4-1 multiplexer
STN_MUXI2_*	1,2,4,8	2-1 multiplexer with inverted output
STN_MUXI2_DG_*	0.75,1,2,3,4,5,6,8,10,12	2-1 multiplexer with inverted output
STN_MUXI2_S_*	0.5,1,2,3,4,6,8	2-1 multiplexer with inverted output, symmetric rise/fall

5.4.2 Sequential Cells

5.4.2.1 Flip-Flops

Table 5-7 Flip-Flops

Function	Drive Options	Description
STN_FDAO22PQ_*	1	D-Flip Flop, pos-edge triggered, lo-sync-clear, q-only, so pin
STN_FDNQ_*	1,2,4	D-Flip Flop, neg-edge triggered
STN_FDNQ_D_*	1,2,4	D-Flip Flop, neg-edge triggered
STN_FDNRBQ_*	1,2,4	D-Flip Flop, neg-edge triggered, lo-async-clear, q-only
STN_FDNRBQ_D_*	1,2,4	D-Flip Flop, neg-edge triggered, lo-async-clear, q-only
STN_FDNRBSBQ_*	1,2,4	D-Flip Flop neg-edge triggered, lo-async-clear/set, q-only
STN_FDPCBQ_*	1,1.5,2,3,4	D-Flip Flop, pos-edge triggered, lo-sync-clear, q-only
STN_FDPCBQ_D_*	1,2,4	D-Flip Flop, pos-edge triggered, lo-sync-clear, q-only
STN_FDPHQ_*	2	D-Flip Flop, pos-edge triggered, sync hold, q-only
STN_FDPHRBSBQ_*	2	D-Flip Flop, pos-edge triggered, lo-async-clear/set, sync hold, q-only

Table 5-7 Flip-Flops

Function	Drive Options	Description
STN_FDPMQ_*	1	D-Flip Flop, pos-edge triggered, 2-to-1 muxed data inputs, q-only
STN_FDPQB_*	1,2,3,4	D-Flip Flop, pos-edge triggered, qn-only
STN_FDPQB_D_*	1,2,3,4	D-Flip Flop, pos-edge triggered, qn-only
STN_FDPQ_*	1,2,4	D-Flip Flop, pos-edge triggered, q-only
STN_FDPQ_D_*	1,1.5,2,3,4	D-Flip Flop, pos-edge triggered, q-only
STN_FDPRBQ_*	1,2,4	D-Flip Flop, pos-edge triggered, lo-async-clear, q-only
STN_FDPRBQ_D_*	1,1.5,2,3,4	D-Flip Flop, pos-edge triggered, lo-async-clear, q-only
STN_FDPRBSBQ_*	1,2,4	D-Flip Flop pos-edge triggered, lo-async-clear/set, q-only
STN_FDPSBQ_D_*	1,2,4	D-Flip Flop pos-edge triggered, lo-async-set, q-only

5.4.2.2 Flip-Flops with Scan

Table 5-8 Flip-Flops with Scan

Function	Drive Options	Description
STN_FSDAO22PQO_*	1,2,4	D-Flip Flop w/scan, pos-edge triggered, lo-sync-clear, q-only, so pin
STN_FSDAO22PQ_D_*	1,2,4	D-Flip Flop w/scan, pos-edge triggered, lo-sync-clear, q-only
STN_FSDNQO_*	1,2,4	D-Flip Flop w/scan, pos-edge triggered, q-only, so pin
STN_FSDNQ_D_*	1,2,4	D-Flip Flop w/scan, neg-edge triggered, q-only
STN_FSDNRBQ_*	1,2,4	D-Flip Flop w/scan, neg-edge triggered, lo-async-clear, q-only
STN_FSDNRBQ_D_*	1,2,4	D-Flip Flop w/scan, neg-edge triggered, lo-async-clear, q-only
STN_FSDPC1BQO_*	1,2,4	D-Flip Flop w/scan, pos-edge triggered, lo-sync-clear, q-only, so pin
STN_FSDPC1BQ_D_*	1,1.5,2,3,4	D-Flip Flop w/scan, pos-edge triggered, lo-sync-clear, q-only
STN_FSDPHQO_*	2	D-Flip Flop w/scan, pos-edge triggered, sync hold, q-only, so pin
STN_FSDPHQ_D_*	2	D-Flip Flop w/scan, pos-edge triggered, sync hold, q-only
STN_FSDPMQO_*	1,2,4	D-Flip Flop w/scan, pos-edge triggered, 2-to-1 muxed data inputs, q-only, so pin

Table 5-8 Flip-Flops with Scan

Function	Drive Options	Description
STN_FSDPMQ_D_*	1,2,4	D-Flip Flop w/scan, pos-edge triggered, 2-to-1 muxed data inputs, q-only
STN_FSDPQB_*	1,1.5,2,3,4	D-Flip Flop w/scan, pos-edge triggered, qn-only
STN_FSDPQB_D_*	1,1.5,2,3,4	D-Flip Flop w/scan, pos-edge triggered, qn-only
STN_FSDPQO_*	1,2,4,6,8	D-Flip Flop w/scan, pos-edge triggered, q-only, so pin
STN_FSDPQ_DV1_*	1,2,4	D-Flip Flop w/scan, pos-edge triggered, q-only
STN_FSDPQ_D_*	1,1.5,2,3,4	D-Flip Flop w/scan, pos-edge triggered, q-only
STN_FSDPRBQO_*	1,2,4,6,8	D-Flip Flop w/scan, pos-edge triggered, lo-async-clear, q-only, so pin
STN_FSDPRBQ_D_*	1,1.5,2,3,4	D-Flip Flop w/scan, pos-edge triggered, lo-async-clear, q-only
STN_FSDPRBSBQO_*	1,2,4,6,8	D-Flip Flop w/scan, pos-edge triggered, lo-async-clear/set, q-only, so pin
STN_FSDPRBSBQ_D_*	1,2,4	D-Flip Flop w/scan, pos-edge triggered, lo-async-clear/set, q-only
STN_FSDPSBQO_*	1,2,4,6,8	D-Flip Flop w/scan, pos-edge triggered, lo-async-set, q-only, so pin
STN_FSDPSBQ_D_*	1,2,4	D-Flip Flop w/scan, pos-edge triggered, lo-async-/set, q-only
STN_FSDPTQO_*	1,2,4	D-Flip Flop w/scan, pos-edge triggered, hi-sync-set, q-only, so pin
STN_FSDPTQ_D_*	1,2,4	D-Flip Flop w/scan, pos-edge triggered, hi-sync-set, q-only

5.4.2.3 Latches

Table 5-9 Latches

Function	Drive Options	Description
STN_LDNQ_D_*	1,2,4	D-latch, neg-gate, q-only
STN_LDNRBQ_D_*	1,2,4	D-latch, neg-gate, lo-async-clear, q-only
STN_LDPQ_D_*	1,2,4	D-latch, pos-gate, q-only
STN_LDPRBQ_D_*	1,2,4	D-latch, pos-gate, lo-async-clear, q-only

5.4.3 Special Cells

5.4.3.1 Delay Buffers

Table 5-10 Delay Buffers

Function	Drive Options	Description
STN_DEL_L4_*	1,2,4,8	Delay buffer
STN_DEL_L6_*	1,2,4,8	Delay buffer

5.4.3.2 Clock-Gater Cells

Table 5-11 Clock-Gater Cells

Function	Drive Options	Description
STN_CKGTPLS_*	1,2,3,4,6,8,12,16	Clock Gater, positive clock, synchronous enable, post control
STN_CKGTPLT_*	1,2,3,4,6,8,12,16	Clock Gater, positive clock, synchronous enable, pre control

5.4.3.3 Miscellaneous Cells

Table 5-12 Miscellaneous Cells

Function	Drive Options	Description
STN_TIE0_*	1	Tie low
STN_TIE1_*	1	Tie high
STN_TIEDIN_*	1	Antenna diode with N+/Pwell Diode

5.4.3.4 De-coupling Filler Cells

Table 5-13 De-coupling Filler Cells

Function	Drive Options	Description
STN_DCAP*	4,8,16,32,64	Filler cell with De-coupling

6 Characterization Methodology

Each cell provided as part of the DesignWare Logic library design kit is characterized using a netlist extracted from the cell layout, and the characterization tool suite from Cell Builder. The following parameters are characterized:

1. Input pin capacitance
2. Combinational pin-to-pin delays
3. Output pin transition times
4. Dynamic (switching) power dissipation
5. Timing check constraints
6. Steady-state power dissipation.

A characterization database is generated for these parameters over a pre-determined set of environmental conditions, usually comprised of "worst case", "typical", and "best case" corners. Additional corners are included as needed. This database is used as the basis for creating synthesis, simulation, and timing models for various EDA tools.

6.1 Input Pin Capacitance

The cell pin being characterized is treated as a simple capacitive load and the rise and fall delays to charge or discharge this ‘capacitor’ are measured by simulation. These measured delays are then matched to actual pure-capacitive equivalent delays.

6.2 Combinational Pin-to-Pin Delays

Combinational cell delays are obtained for each distinct input-pin to output-pin path, by adding a load to the output pin, applying a transition on the input pin, and measuring the corresponding delay through a SPICE circuit simulator. This process is repeated for all desired combinations of input transitions and output loads. In order to model the delays as accurately as possible, the transition applied to the input pin is a realistic CMOS waveform, obtained by passing a piecewise-linear (PWL) transition through a pre-buffer circuit. This pre-buffer is usually one of the non-inverting buffer cells from the library itself.

6.3 Output Pin Transition Times

During the measurement of combinational delays, the rise/fall transition times on the output are also captured for each combination of input transitions and output loads.

6.4 Dynamic Power Dissipation

The power dissipated by a cell during state transitions is obtained by measuring the switching current through the circuit, from VDD to VSS, and multiplying by the rail-to-rail voltage difference. This switching current is measured during each of the pin-to-pin delay characterizations, as described in "Combinational Pin-to-Pin Delays" on page 43, above, as well as for input transitions which cause an internal node transition but have no corresponding output transition.

6.5 Timing Check Constraints

A reference simulation is performed for a given clock-to-Q transition, and then the clock and data edges (or two clock edges, in the case of minimum pulse width) are moved progressively closer together until failure (that is, the clock-to-Q transition increases in delay by prescribed amount prior to complete failure). The resulting time between the two edges is the setup, hold, or MPW. This process is repeated for all desired data-input-slope/clock-input-slope combinations.

6.6 Steady-State Power Dissipation

The steady-state, or static, power dissipated by a cell in a given state is determined by measuring the current through the circuit, from VDD to VSS, and multiplying by the rail-to-rail voltage difference. The current is only measured once the circuit has fully transitioned into that state, and all the nodes have settled. This process is repeated for all the distinct states of the circuit.

7 Modeling Methodology and Validation

All of the models provided in the DesignWare Logic library design kit are designed to provide accurate representations of the design and performance of each cell to all of the various EDA tools used in the design process, while at the same time optimizing the runtime performance in each of these tools.

The EDA views provided as standard deliverable with this DesignWare Logic library are listed in "Standard List of Models" on page 45, below. To request a custom set of views, please contact Synopsys as described in "Technical Assistance" on page 8.

7.1 Standard List of Models

The EDA views delivered with the standard DesignWare Logic library design kit are listed below. Please refer to the file named **`release.txt`** shipped with this product for the specific versions of the EDA views delivered.

- GDSII
- LVS
- LEF
- Verilog
- VHDL (VITAL-1 compliant)
- Synopsys Galaxy/Milkyway database (frames and CLF for each PVT corner)
- Synopsys Liberty DB (one per PVT corner)
- Synopsys Physical Liberty DB
- TLF (one per PVT corner)
- Mentor Graphics Fastscan ATPG models

7.2 Conditional compilation directive switches supported in verilog models

The following compilation options are available in the DesignWare Logic libraries:

- - _fv: Compile verilog models with +define+_fv for formal verification and DFT flows.
- - VIRL_functiononly: Compile verilog models with +define+VIRL_functiononly for unit simulations

7.3 Quality Assurance Methodology

Each DesignWare Logic product undergoes a variety of tests, focusing on accuracy, consistency, and usability in a wide variety of proprietary and industry tools. Unit level cell tests include Layout-versus-Schematic (LVS), Design Rule Checking (DRC), architectural checks that ensure proper design, logic functionality and equivalency, and characterization accuracy. A variety of leading industry design and analysis tools are run with these libraries and the results compared to each other to ensure consistent and accurate interpretation of the various library views. A sample RTL->GDS design methodology is run to ensure successful completion of a tape-out flow. This flow is almost identical to the sample flow included with each library.

All library validation is audited and documented in accordance with Synopsys requirements.

Coverage on the EDA views listed in "Standard List of Models" on page 45 is obtained with the following:

Library Construction Checks

Library Construction Checks are used to validate that the design is accurately represented in each EDA model.

Library View Cross-Checking

Library View Cross-Checking is performed to ensure that the information is consistent across the various model types.

Tool vs. Golden Data Checks

Tool vs. Golden Data Checks are used to correlate each EDA tool with the design area.

Tool vs. Tool Checks

Tool vs. Tool Checks are used to verify consistency between two analysis tools.

Methodology Checks

Methodology Checks are used in verifying the end-user flow and proactively identifying any problems with usage.

7.4 Explanation of Physical Models

Physical models come in two flavors: full layout and cell abstract. Full layouts are required in the final steps of design verification, while cell abstracts are used by place-and-route tools.

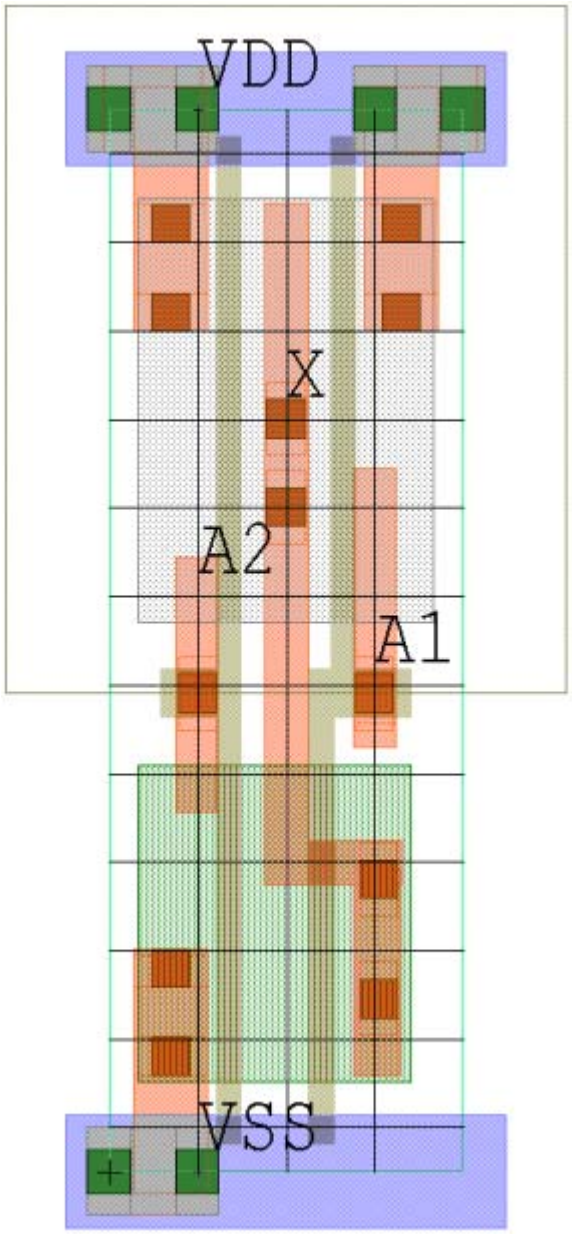
7.4.1 Full Layout Database

The full layout database contains all tapeout layers for all cells. These views are required for final design verification, when they are merged in with the routing geometries to run full-layer DRC, LVS, and other checks.

The full layout database in this library is provided both in GDSII format and in the Synopsys Galaxy/Milkyway "CEL" format.

Figure 7-1, below, shows an example of the full layout database for a 2-input NAND cell.

Figure 7-1



7.4.2 Cell Place-and-Route Abstract Model

Cell place-and-route abstract models, variously referred to as LEF, frame, phantom, footprint, PLIB or Vulcano, contain only the layout data required by the place-and-route tools. This data generally consists of the cell size and bounding box, the location and layer information for pins and blockages, pin names, and pin antenna information.

The place-and-route abstracts in this library are provided both in Cadence LEF format and in the Synopsys Milkyway "frame" format.

Figure 7-2, below, shows a representation of the information provided in the place-and-route abstract for the 2-input NAND cell shown in “Full Layout Database” on page 47.

Figure 7-2