VTVT TSMC 0.18 μm Standard Cell Library Documentation
(Release 1 – November 1, 2007)

Introduction

The VTVT (Virginia Tech VLSI for Telecommunication) group has developed a standard cell library kit based on the TSMC 0.18 μm technology. In this first release, the layout, symbols, and schematic libraries were developed and re-compiled using a modified version of the NCSU kit version 1.5.1. The layouts were developed using the MOSIS DEEP design rules, namely SCN6M_DEEP. The kit, as distributed here, includes:

- Layouts, symbols and schematic in Cadence dfII format
- Synopsys Synthesis (.db and .sdb) and VHDL simulation libraries, as well as the source code (.lib and .slib) for the synthesis libraries.
- LEF files for the PNR tools

  The layouts include only simple cells (2- and 3-input combinational cells, tristate buffers, flip-flops, latches).
- Symbols and Schematic libraries for improved user-interface during synthesis and import from synthesis tool to Cadence.
- Design supported by up-to-date testing and simulation tools:
    - Cadence SOC Encounter
    - Synopsys Design Vision
    - Synopsys Design Compiler
    - Synopsys VCS MX

There are some other things the VTVT lab plans to provide, but as of now (11/1/2007) has not yet provided, such as:

- Arithmetic macro cells
- TLF files, timing-driven placement
- Support for TSMC 0.13 μm technology

Cells Contained in the Library

The cell library vtvt_tsmc180 contains the cells listed in Table 1.

Table 1: Cells Contained in the Library vtvt_tsmc180

Cell Name	Function
buf_[1,2,4]	Noninverting buffer, drive strength 1, 2, or 4
inv_[1,2,4]	Inverter, drive strength 1, 2 or 4
and2_[1,2,4]	2-input AND gate, drive strength 1, 2, or 4
and3_[1,2,4]	3-input AND gate, drive strength 1, 2, or 4
and4_[1,2,4]	4-input AND gate, drive strength 1, 2, or 4
or2_[1,2,4]	2-input OR gate, drive strength 1, 2, or 4
or3_[1,2,4]	3-input OR gate, drive strength 1, 2, or 4
or4_[1,2,4]	4-input OR gate, drive strength 1, 2, or 4
nand2_[1,2,4]	2-input NAND gate, drive strength 1, 2, or 4
nand3_[1,2,4]	3-input NAND gate, drive strength 1, 2, or 4
nand4_[1,2,4]	4-input NAND gate, drive strength 1, 2, or 4
nor2_[1,2,4]	2-input NOR gate, drive strength 1, 2, or 4
nor3_[1,2,4]	3-input NOR gate, drive strength 1, 2, or 4
nor4_[1,2,4]	4-input NOR gate, drive strength 1, 2, or 4
xor2_[1,2]	2-input XOR gate, drive strength 1 or 2
xnor2_[1,2]	2-input XNOR gate, drive strength 1 or 2
mux2_[1,2,4]	2-to-1 multiplexer, drive strength 1, 2, or 4
mux3_2	3-to-1 multiplexer, drive strength 2
mux4_2	4-to-1 multiplexer, drive strength 2
ABnorC	(ip1*ip2+ip3)', drive strength 1
ABorC	ip1*ip2+ip3, drive strength 1
ab_or_c_or_d	ip1*ip2+ip3+ip4, drive strength 1
Not_ab_or_c_or_d	(ip1*ip2+ip3+ip4)', drive strength 1
fulladder	One-bit ripple-carry adder, drive strength 1
bufzp_2	noninverting tristate buffer, low-enabled, drive strength 2
invzp_[1,2,4]	inverting tristate buffer, low-enabled, drive strength 1, 2, or 4
cd_8	clock driver, drive strength 8
cd_12	clock driver, drive strength 12
cd_16	clock driver, drive strength 16
lp_[1,2]	high-active D latch, drive strength 1 or 2
Lrp_[1, 2, 4]	high-active D latch with asynchronous low-active reset and drive strength 1, 2, or 4
Lrsp_[1, 2, 4]	high-active D latch with asynchronous low-active reset and asynchronous high-active set, drive strength 1, 2, or 4
Dp_[1,2,4]	rising-edge triggered D flip-flop (with 1, 2, or 4 drive strength)
Drp_[1,2,4]	rising-edge triggered D flip-flop with asynchronous low-active reset

	(1, 2, or 4 drive strength)
drsp_[1,2,4]	rising-edge triggered D flip-flop with asynchronous low-active reset and asynchronous high-active set
dksp_1	rising-edge triggered D flip-flop with asynchronous active high set and extra inverted output.
dtsp_1	rising-edge triggered D flip-flop with asynchronous active high set input and serial scan input.
dtrsp_2	rising-edge triggered D flip-flop with asynchronous low-active reset, asynchronous high-active set, and serial scan input
jkrp_2	rising-edge triggered JK flip-flop with asynchronous active-low reset and extra inverted output, drive strength 2.
filler	filler cell (empty cell with power and ground rails and nwell)

Downloading the Cell Library

After you have downloaded the kit from the web and un-tarred it, you will have the following files listed in Table 2.

It is recommended that before the layout, symbols and schematic views (in Cadence_Libraries.tar) are used, a SCMOS_DEEP-based technology library for the TSMC 0.18 μm technology, named TSMC_CMOS018_DEEP, be created first. One way is by following the installation procedures on http://www.vtvt.ece.vt.edu/vlsidesign/tutorialCadence_unix_env.php#install.

Another way is to follow the instructions in the README file for the Cadence_Libraries directory of this release package.

Table 2: Contents of the Cell Library Tar file vtvt_tsmc180_release.tar.gz

File	Contents	Containing its own README?
cdk_modifications.tar.gz	Modifications to NCSU kit; in tar format	Y
Synopsys_Libraries.tar.gz	Synopsys synthesis layout and symbol libraries (.db and .sdb) and VHDL simulation libraries; in tar format	Y
Cadence_Libraries.tar.gz	Standard cell layout, symbol and schematic in dfII and GDSII formats; in tar format	Y
vtvt_tsmc180_lef.tar.gz	LEF files, GDSII-to-dfII map for Cadence SOC Ensemble and Virtuoso; in tar format	Y
tutorial_files.tar	Tutorials' input files for design flow; in tar format	Y

Installing the Cell Library Kit

A possible sequence for installing the cell library kit is as follows:

1. Download the kit and untar it.
2. Untar cdk_modifications.tar and make modifications to your NCSU kit. If you intend to generate a new LEF file, then, re-compile the NCSU Kit, following directions from the NCSU site (see http://www.cadence.ncsu.edu/wiki/Techfiles), or use the technology file (.tf file) provided in the release package.
3. Untar Cadence_Libraries.tar and copy the contained libraries to the user's working directory for layout, symbols and schematic tools (e.g. ICFB)
4. Untar vtvt_tsmc180_lef.tar and copy the contents to the user's working directory for placement-and-routing tools (e.g. SOC Encounter™ from Cadence).
5. Untar Synopsys_Libraries.tar and place it in a directory accessible by the users for use with Synopsys® Synthesis (Design Compiler, and Design Vision) and Simulation tools (Scirocco, Virsim).

Overview of Parts

The standard cell library `vtvt_tsmc180` is intended for use with Cadence SOC Encounter Placement-and-Routing (PNR) tool. The layouts were developed with Cadence Virtuoso custom layout tool (using `ICFB`). The MOSIS DEEP design rules were chosen since the I/O pads available from MOSIS (designed by Tanner Research – see http://www.mosis.org/Technical/Designsupport/pad-library-scmos.html) followed that particular set of rules. The symbols and schematic were developed with Cadence Composer schematic.

The Synopsys `.db` synthesis library was obtained by characterizing the layout using HSPICE. The Synopsys `.sdb` synthesis library was obtained by extracting the symbols library from Cadence to Synopsys Design Compiler using EDIF 2.0.0. The `.synopsys_dc.setup` and `.synopsys_vss.setup` files used by the VTVT group in creating the libraries are also included in the tar file.

Further Documentation

After you untar the tar files, some of them contain their own README files. They will provide further documentation for the specific directory. However, they do not contain guidelines or tutorials about the tools. For tutorials and guidelines to modify the standard cell library, and/or use the simulation and design tools, refer to our site at: http://www.vtvt.ece.vt.edu/vlsidesign/index.php.