Open ASIC EDA Tooling Random Diagrams

https://j.mp/openeda-diagram

These diagrams are an extension of the diagram found in the "Open FPGA EDA Tooling Interchange Formats" document to include ASIC.

Extended Diagram Including ASIC EDA Tools

Key

- **Flow Stage**
- File Format
- Open Source Tool
- Closed Source Tool
- *Currently non-existent*

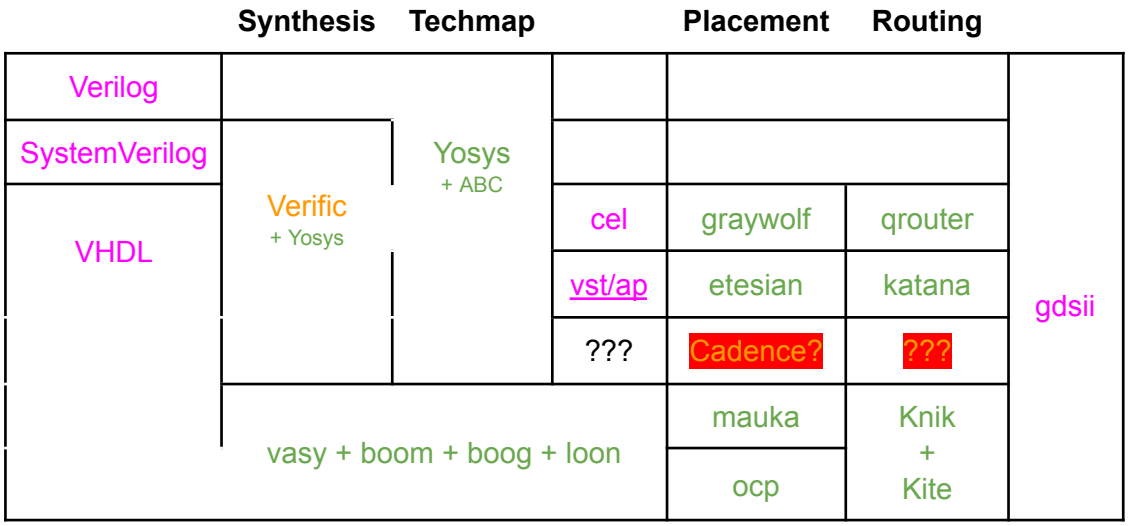


Table of Contents

File Formats

File Type	What does it do?	Open Format	Proprietary Format
Parameterized cells	Parameterized Cells primitive devices that have layouts determined by parameterization	Tcl scripts for Magic	Cadence PCells (Cadence Virtuoso)
DRC deck		Magic techfile	Mentor Graphics Calibre ?
LVS Deck		Magic techfile	Mentor Graphics Calibre ?
GDS	mask layout data	GDS	
LEF (Library Exchange Format) macros	abstract cell view	LEF	
Liberty (.lib) timing files		liberty	
SPICE or CDL netlists		Ngspice or CDL	
Schematic symbol		EDIF	Cadence virtuoso
Verilog		Verilog	
SKILL script (.il)	Rules for scribe lines and saw lanes	Tcl script for magic	Cadence SKILL
SKILL script (.il)	Fill structure rules	Tcl script or GDS output style	Cadence SKILL
SPICE or CDL device models		ngspice	Cadence Spectre

- Verification decks including:
    - Design Rule Checking
    - Layout Versus Schematic (i.e., extraction rules)
    - Antenna and Electrical rule check
    - Physical Extraction (Parasitic Extraction)
    - Fill structure rules
    - Rules for scribe lines and saw lanes
    - Rules for minimum/maximum fill density for each layer requiring such a rule.
    - Any other rule files needed to check manufacturability.

- Technology data
    - Layers, layer names, layer/purpose pairs
    - Colors, fills and display attributes
    - Process constraints
    - Electrical characteristics and maximum limits; including parasitic capacitances, sheet resistances, backend stack layer and insulator thicknesses and dielectrics, and electromigration limits
    - Technology LEF file
    - Device characteristics and device models (SPICE or CDL)
    - Monte Carlo distributions and models

- Technology documentation
    - Design Rule Manual - a user friendly representation of the process requirements
    - Manual of Electrical Specification and Device Parameters
    - ESD Guide
    - User Guide for each required library (see list below)
    - Wafer reticle preparation guide

- Cell libraries, to include the following:

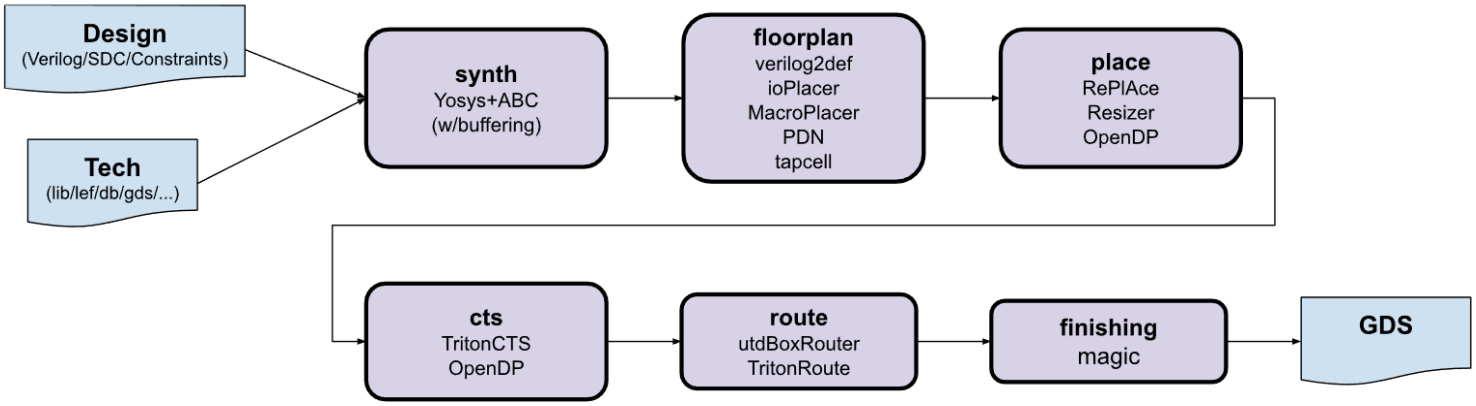
- Digital standard cell libraries
- Padframe I/O libraries
- SRAM cell library
- NVRAM cell library
- Primitive devices
- Analog libraries

OpenRoad Flow

https://theopenroadproject.org/

Parts

- Static Timing Analysis
- Logic Synthesis
- Floor Planning
- Placement
- Routing
- Clock Tree Synthesis



Name	Does	GitHub	
RePlAce	Global placement tool	Code	
ioPlacer	IO and Pin Placer for Floorplan-Placement Subflow		
OpenDP	Detailed Placement engine		
TritonCTS	Clock tree synthesis		
OpenSTA	Static Timing Analysis		
TritonRoute	Detailed Router		
TritonMacroPlace	Macro placement		
tapcell	Welltap and endcap cell insertion		
magic			
TritonFP	Floorplanning+Powerplanning		
TritonSizer	leakage/dynamic power recovery, timing recovery		
pdn	power grid insertion		

DATC RDF-2019

https://vlsicad.ucsd.edu/Publications/Conferences/373/c373.pdf

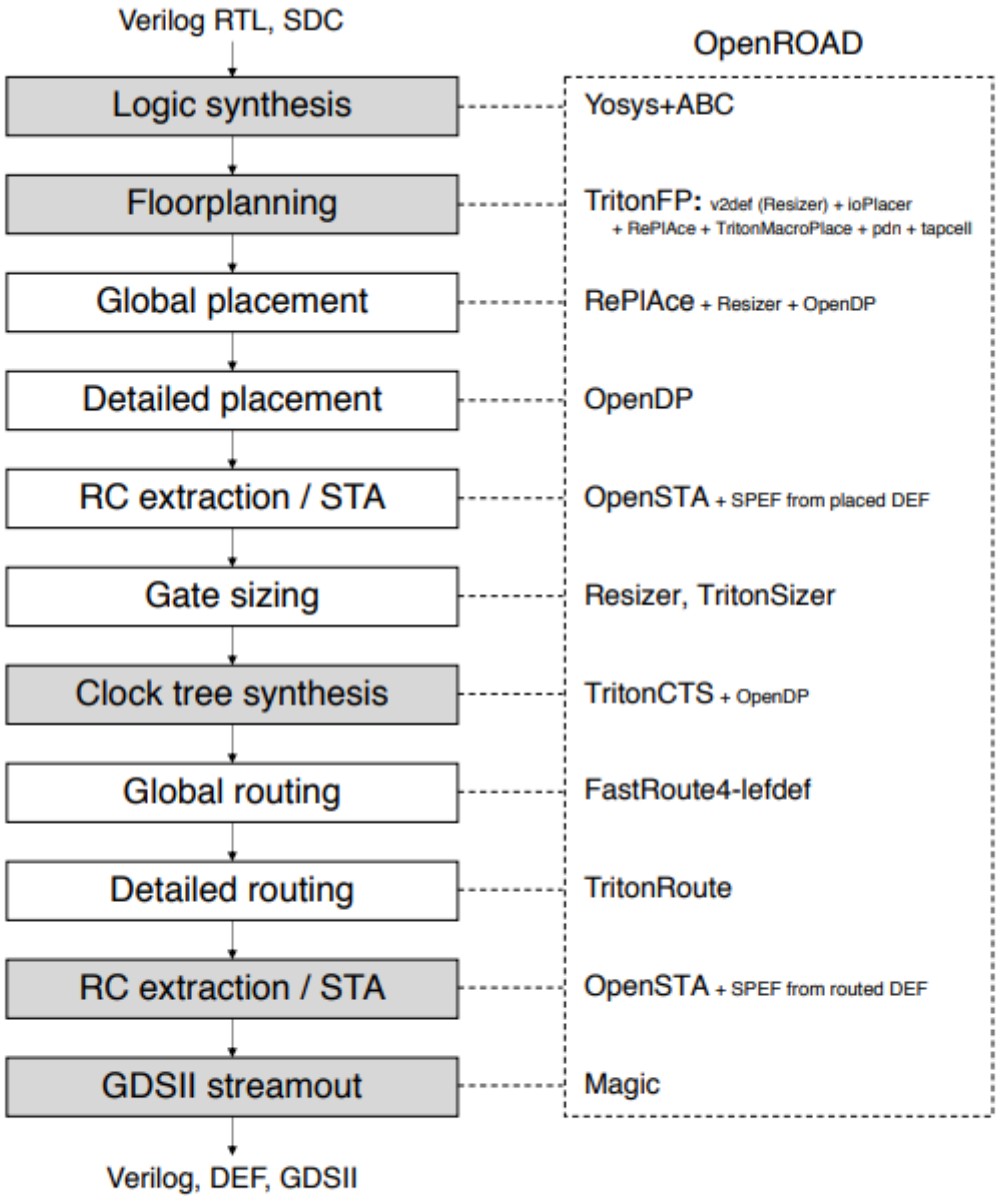


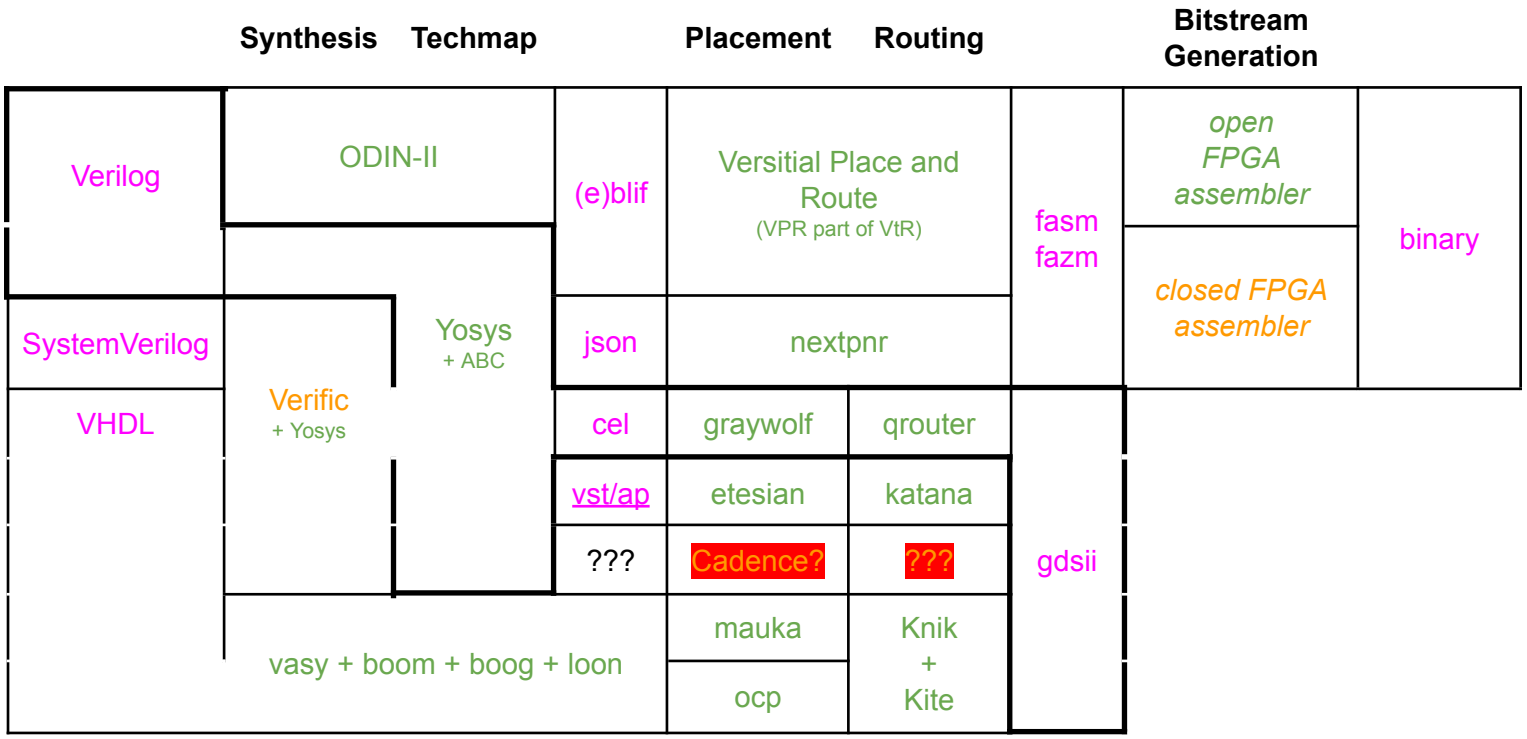
Fig. 1. Overview of RDF-2019 flow. It unifies RDF-2018, OpenROAD and additional tools. Vertical extensions made in this year are highlighted in gray.

Component	RDF-2018	RDF-2019 Extension
Logic synthesis (RTL)	-	Yosys + ABC
Logic synthesis (gate-level)	ABC	-
Floorplanning	-	TritonFP
Global placement	NTUPlace3, ComPLx, mPL5/6, FastPlace3-GP, Capo, Eh?Placer	FZUplace, RePlAce
Detailed placement	FastPlace3-DP, MCHL-T	OpenDP

Clock tree synthesis	-	TritonCTS
Global routing	NCTUgr, FastRoute4.1, BFG-R	FastRoute4-lefdef
Detailed routing	NCTUdr	Dr. CU, TritonRoute
GDSII streamout	-	Magic
Gate sizing	USizer2012, USizer2013	Resizer, TritonSizer
Parasitic extraction	-	OpenROAD Utilities (PEX)
Timing analysis	OpenTimer, iTimerC	OpenSTA
Libraries/Technologies	ISPD-2012/2013 Contests, ASAP-7nm	NanGate45, NCTUcell

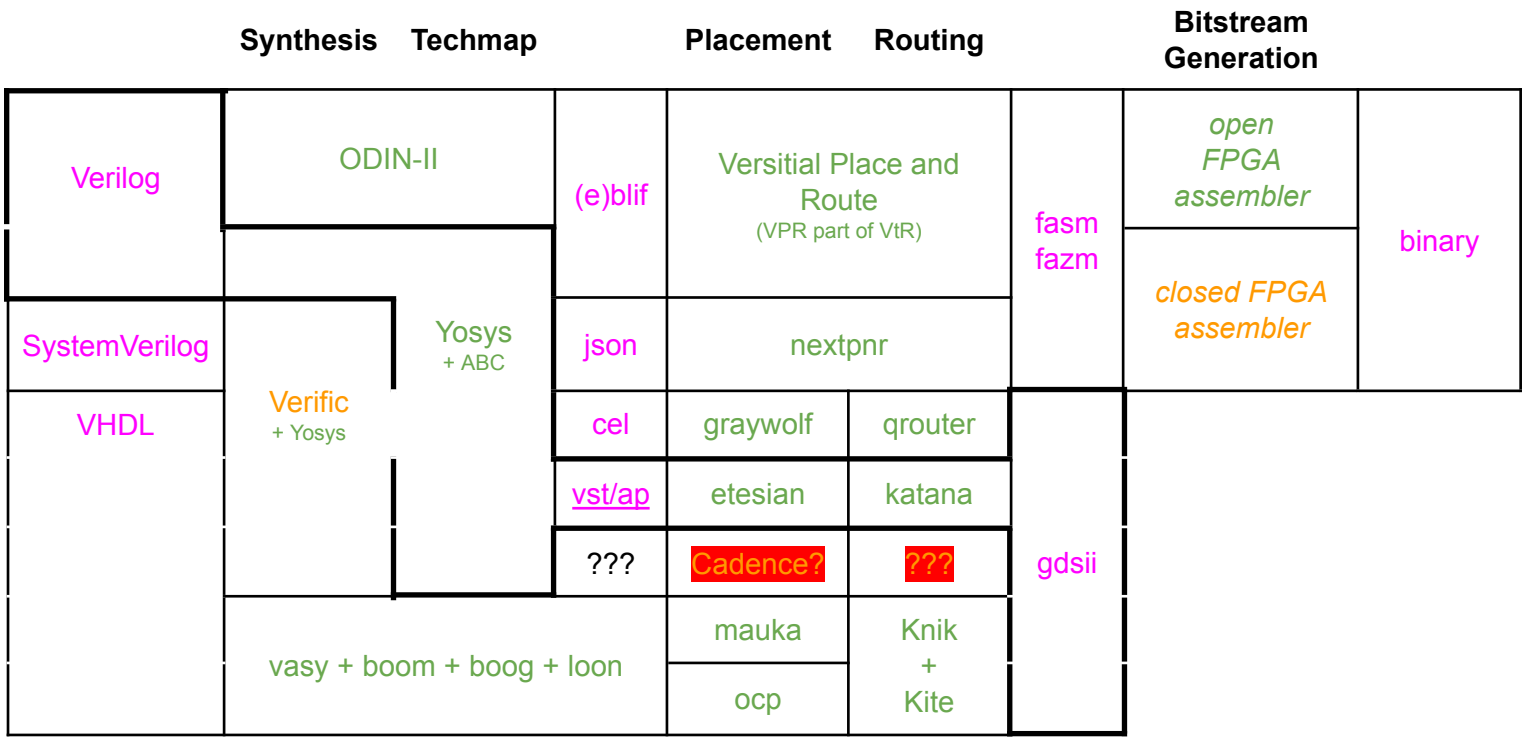
Open Source "qflow" Flow

- verilog → yosys → cel → graywolf → qrouter → gdsii



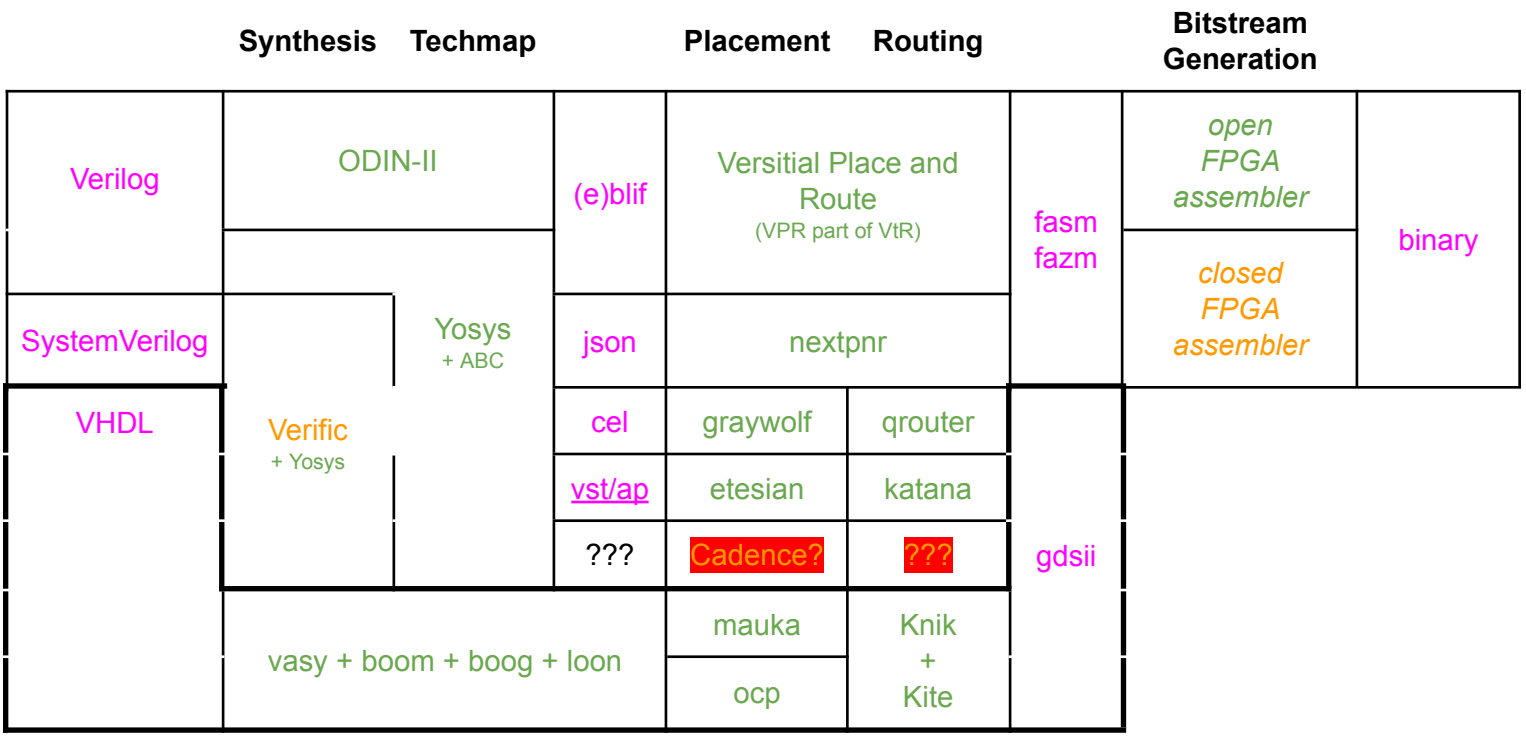
Open Source "Katana" Flow

- Replacing Coriolis / Alliance?
- verilog → yosys → ??? → etesian → katana → gdsii



Open Source "Alliance / Coriolis" Flow

Thing	**Alliance**	**Coriolis**	**qflow**
Language	"Custom" VHDL		Verilog
Synthesis			yosys
(?)	GenLib vasy + boom + boog + loon	stratus	
Placement	ocp || mauka	ocp || mauka || etesian?	graywolf
Routing	nero	Kite (and Knik)	qrouter



Libraries

Verilog Preprocessor

https://www.veripool.org/papers/Preproc_Good_Evil_SNUGBos10_paper.pdf - The Verilog Preprocessor: Force for `Good and `Evil

Verilog Parser

https://github.com/ben-marshall/verilog-parser
- A Flex/Bison Parser for the IEEE 1364-2001 Verilog Standard.
- Lacks SystemVerilog completely.

https://github.com/billswartz7/utd-SystemVerilog
- Part of the POSH / IDEA Program?

https://github.com/antlr/grammars-v4/tree/master/system_verilog
- Microsoft's, currently parked idle.

https://github.com/tree-sitter/tree-sitter-verilog
- **Incremental Parsing**

https://github.com/MikePopoloski/slang is an independent MIT-licensed SystemVerilog parser
- Hand-written, recursive descent parsing
    - Seems to cover SVA constructs
- Emphasis on good syntax diagnostics, incremental editing
- Analyzes beyond syntax (e.g. expressions)
- Looks active

Bison / Yacc / Lex

https://github.com/ben-marshall/verilog-parser
- A Flex/Bison Parser for the IEEE 1364-2001 Verilog Standard.
- Lacks SystemVerilog completely.

ANTLR

https://github.com/Nic30/hdlConvertor
- SystemVerilog and VHDL parser, preprocessor in C++. SystemVerilog support is WIP.

https://github.com/antlr/grammars-v4/tree/master/system_verilog
- Microsoft's, currently parked idle.

https://github.com/veriktig/ieee1800_2017
- SystemVerilog preprocessor, lexer and parser in Java. It uses ANTLR.

Verilog Test Suites

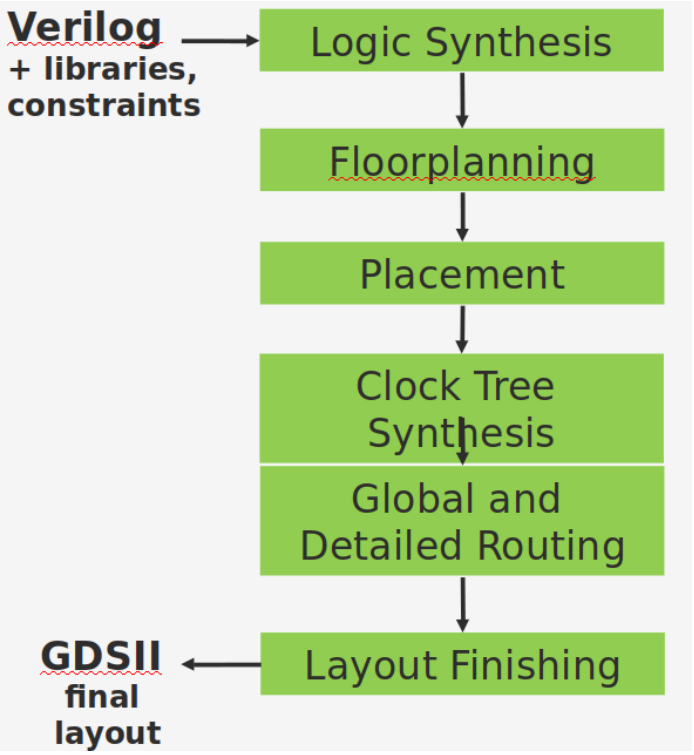
Having a quality test suite is vital for qualifying tools, open-source or otherwise.

- ivltest is a regression test suite for iverilog.
- https://github.com/verilator/verilator_ext_tests - Extended and external tests for Verilator testing

Other Projects

- cl-vhdl - lisp, Parser of VHDL into lisp-expressions
- HDL_ANTLR4 - C# projects that use ANTLR4 library to analyse VHDL and Verilog code
- hdlparse - vhdl/verilog parser in python
- ieee1800_2017 - Java, SystemVerilog preprocessor
- pyVHDLParser - python vhdl parser with 2008 support
- rust_hdl - rust vhdl 2008 parser
- slang - Parser and compiler library for SystemVerilog.
- systemc-clang - SystemC Parser using the Clang Front-end
- v2sc - vhdl to systemc
- veelox - Java+ANTLR, An experiment in SystemVerilog Preprocessing
- verilog-parser - A Flex/Bison Parser for the IEEE 1364-2001 Verilog Standard.
- verilog-parser - verilog parser, c
- Verilog-Perl
- vpp.pl - verilog preprocessor with integrated Perl
- sv2v- Haskell, SystemVerilog to Verilog

Tools



Logic Conversion

System Verilog to Verilog

zachjs/sv2v

https://github.com/zachjs/sv2v

sv2v converts SystemVerilog (IEEE 1800-2017) to Verilog (IEEE 1364-2005), with an emphasis on supporting synthesizable language constructs.

GitHub Link	Dependencies
umich-cadre/sv2v	Requires Synopsys Design Compiler
bespoke-silicon-group/bsg_sv2v	Requires Synopsys Design Compiler
zachjs/sv2v	No closed source requirements!

Logic synthesis (RTL)

	GitHub	Homepage
Yosys	YosysHQ/yosys	

ODIN-II		

Logic synthesis (gate-level) / Tech-mapping

	GitHub	
ABC	berkeley-abc/abc	

Floorplanning

Placement

Global

NTUPlace3, ComPLx, mPL5/6, FastPlace3-GP, Capo, Eh?Placer
FZUplace, RePlAce

Detailed

FastPlace3-DP, MCHL-T
OpenDP

Routing

Global

NCTUgr, FastRoute4.1, BFG-R
FastRoute4-lefdef

Detailed

NCTUdr
Dr. CU, TritonRoute

Clock Tree Synthesis

TritonCTS

Gate Sizing

USizer2012, USizer2013
Resizer, TritonSizer

Parasitic Extraction

OpenROAD Utilities (PEX)

Timing Analysis

OpenTimer, iTimerC
OpenSTA

Standard Cell Libraries

ISPD-2012/2013 Contests, ASAP-7nm
NanGate45, NCTUcell

An open source PDK should contain the following:

- Symbols in the format used by the Electric VLSI tool for all primitive devices and all library cells (see below for list of libraries).
- Parameterized cells (PCells) of primitive devices in TCL code for Magic
- Startup script for Magic
- Setup file for LVS using Netgen
- Setup file for digital synthesis using qflow
- The set of libraries provided by the foundry, at least including one or more digital standard cell sets, and an I/O padframe cell set. Each library consists of files copied directly from the Skywater data where the efabless stack tools use those files in the original format (such as liberty, LEF, verilog), and files which are in non-standard formats specific to the tools in the stack, which are derived from standard formats.
- Technology file for magic comprising the following:
    - Layout Design Rules
    - Antenna and Electrical rules
    - Device extraction rules
    - GDS generation rules
    - GDS input rules
    - Layer minimum/maximum density rules
    - LEF input rules
    - Electrical connectivity rules

- Layer drawing style assignments

DRC Checking

Open Source Tooling Owners

Tool	"Primary Owner"
qflow	Tim Edwards
magic	Tim Edwards
netgen	Tim Edwards
Electric	Steve Rubin
iverilog	Stephen Williams
ngspice	Holger Vogt and Paolo Nenzi
graywolf	Ruben Undheim
qrouter	Tim Edwards
vesta	Tim Edwards
OpenSTA	The OpenROAD project
OpenROAD	The OpenROAD project
OpenTimer	Tsung-Wei Huang
auto_pdks	Tim Edwards
xcircuit	Tim Edwards
gtkwave	Tony Bybell, Joel Wheeler
yosys	Symbiotic EDA
spectre2spice	Thomas Benz
OpenRAM	Matthew Guthaus
Open Galaxy (project management)	efabless
padring	Symbiotic EDA

Proprietary Tools

Tool	Vendor
AscentLint	Real Intent
IDesignSpec	Agnisys
Catapult HLS	Mentor
Conformal	Cadence
Design Compiler	Synopsys
Formality	Synopsys
Genus	Cadence
Innovus	Cadence
JasperGold	Cadence

Meridian CDC	Real Intent
RedHawk	Ansys
PrimeTime	Synopsys
Spyglass	Synopsys
Sting	Valtrix
Synplify	Synopsys
VCS	Synopsys
DVT/Verissimo	AMIQ
Virtualizer	Synopsys
Virtuoso	Cadence
VXE	Cadence
Xcelium	Cadence
Xtensa	Cadence