Synthesis and Optimization of Digital Systems

Logic Synthesis Flow using Synopsys®

LAB1: Technology Library & Logic Design

Andrea Calimera

andrea.calimera@polito.it

Valentino Peluso

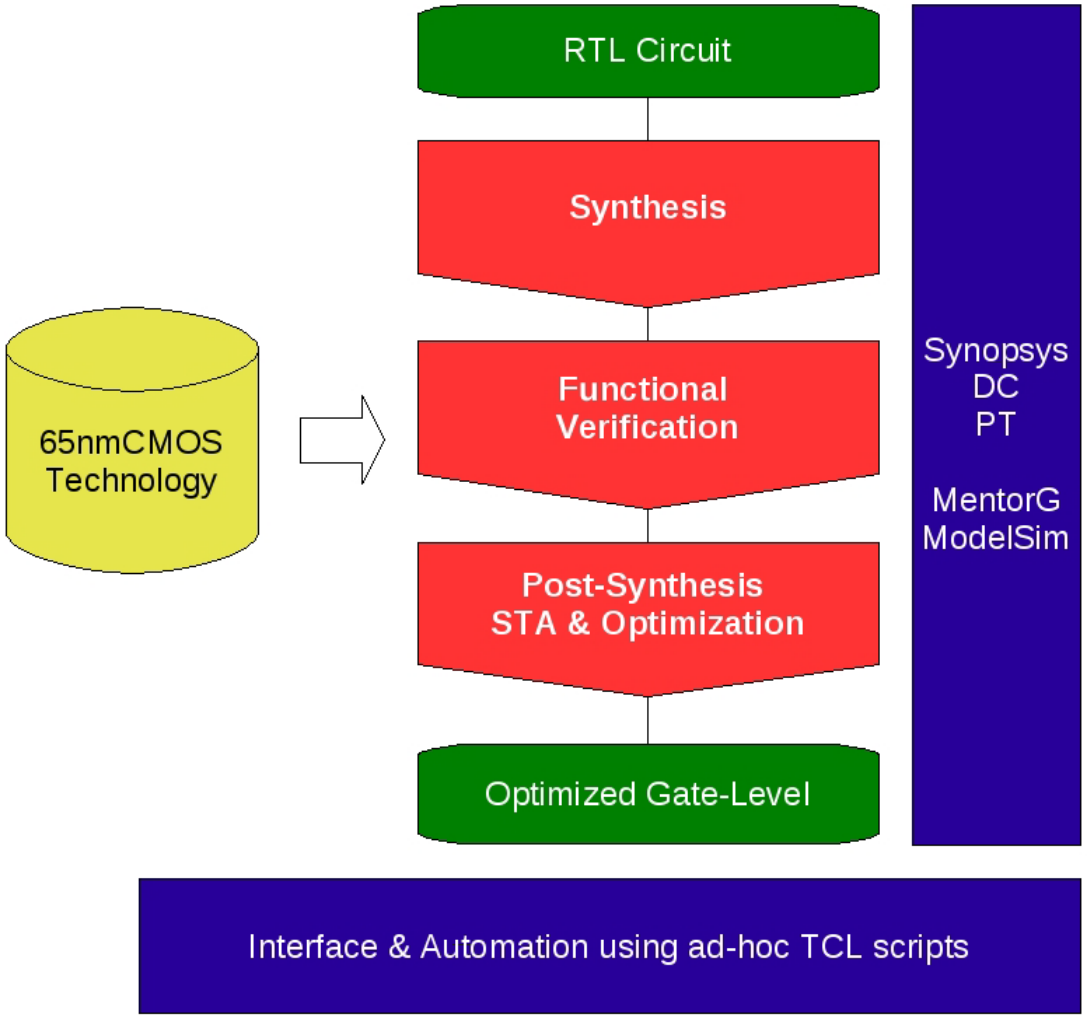
valentino.peluso@polito.it

Dipartimento di Automatica e Informatica

Politecnico di Torino

Technology Library

Logic Design

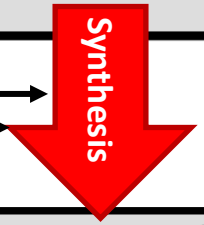


Logic Design

```
module counter(
     input clk, rstn, load,
     input  [1:0] in,
     output reg [1:0] out);
always @(posedge clk)
  if (!rstn) out <= 2'b0;
  else if (load) out <= in;
  else out <= out + 1;
endmodule
```

Technology Library

Design Constraints




```
module counter ( clk, rstn, load, in, out );
  input [1:0] in;
  output [1:0] out;
  input clk, rstn, load;
  wire   N6, N7, n5, n6, n7, n8;

  FFPQ1 out_reg_1 (.D(N7),.CK(clk),.Q(out[1]));
  FFPQ1 out_reg_0 (.D(N6),.CK(clk),.Q(out[0]));
  NAN2D1 U8 (.A1(out[0]),.A2(n5),.Z(n8));
  NAN2D1 U9 (.A1(n5),.A2(n7),.Z(n6));
  INVD1 U10 (.A(load),.Z(n5));
  OA211D1 U11 (.A1(in[0]),.A2(n5),.B(rstn),.C(n8),.Z(N6));
  OA211D1 U12 (.A1(in[1]),.A2(n5),.B(rstn),.C(n6),.Z(N7));
  EXNOR2D1 U13 (.A1(out[1]),.A2(out[0]),.Z(n7));
endmodule
```

Technology Library

- **100 to 1000 cells:**
  - Gates – IV, AND, NAND, NOR, …
  - Sequential – FFs, Latches, Scan Flops, …
  - I/O Cells – Input pads, Output Pads, ….
  - Variety of drive strengths for all cells.
    - E.g., IVX2, IVX4, IVX10…

Technology Library

- **High Quality Models of a Std. Cell**
  - Timing Behaviour
  - Power Behaviour
  - General Information (Area, Functionality, …)
- **Process Options**
- **Different Envinronmental Condition**
- **Used in multiple phase of the design-flow**
  - Synthesis
  - Post-synthesis Simulation
  - STA & Power Estimation
  - Place&Routing
  - ……

Library Operating Conditions

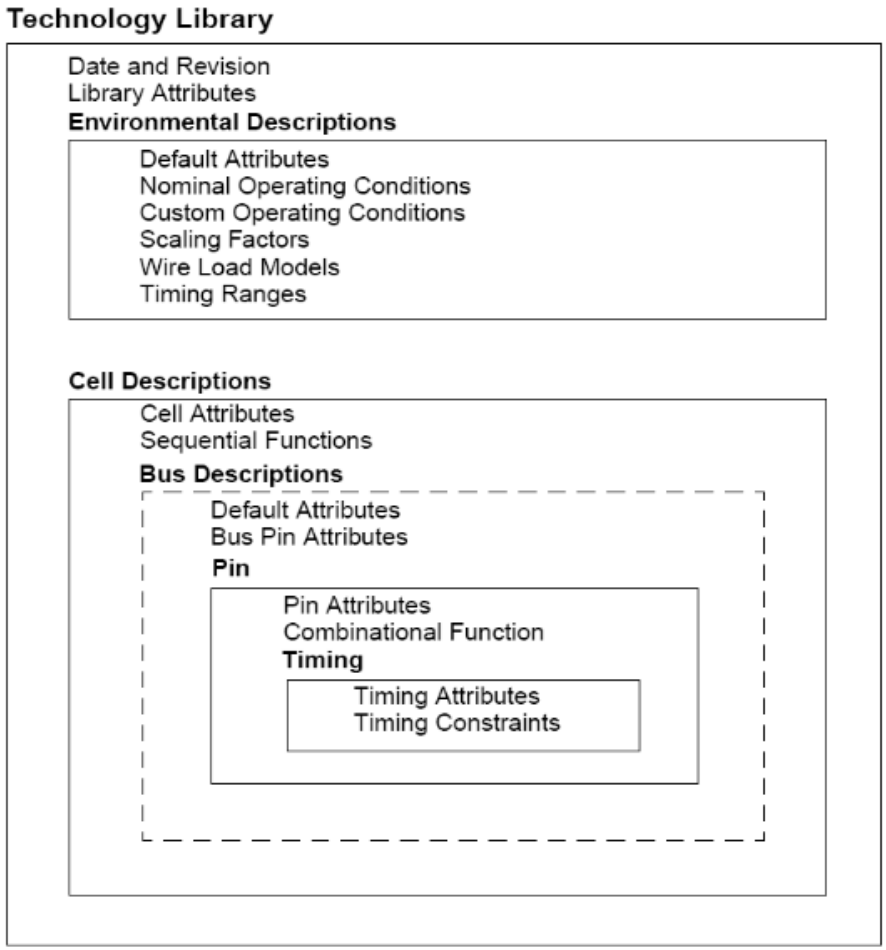
Each library contains the characterization of each standard cell for a particular PVT corner:

- ***Process*: fabrication yield**
- ***Voltage*: nominal supply voltage**
- ***Temperature*: on-chip operating temperature**

Library Name	*Process*	*Voltage*	*Temperature*
CORE65LPSVT_bc_1.35V_m40C.db	best (fast)	1.35V	-40°C
CORE65LPSVT_nom_1.00V_25C.db	nominal	1.00V	25°C
CORE65LPSVT_wc_1.05V_105C.db	worst (slow)	1.05V	105°C

Liberty File

- **Textual File containing information about std. cells**



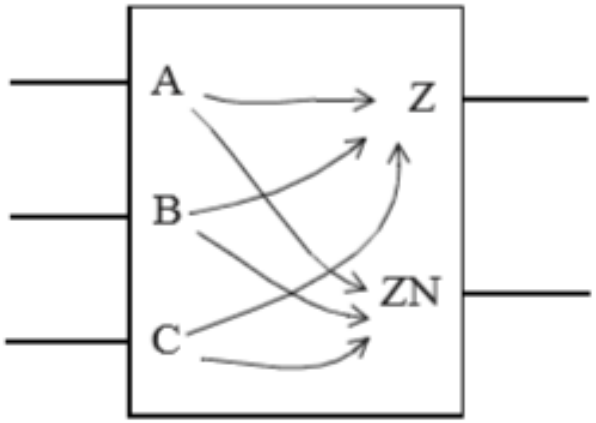
- Structural Information
- Functionality
- Timing
- Power
- Environmental Information

Liberty File

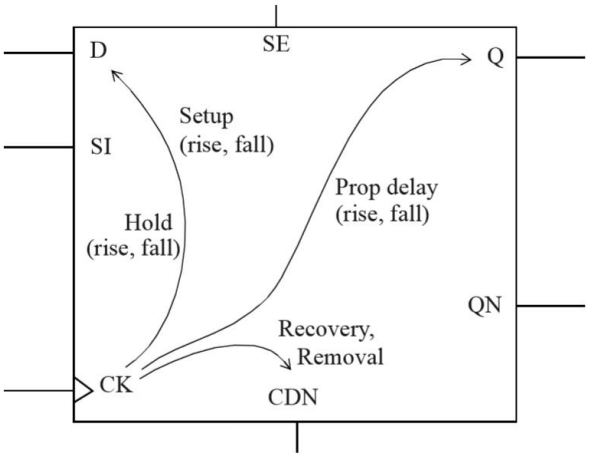
Cell Attributes in .lib File

- Structural Information
  - Pins
  - Bus
  - Power rails connections
- Functionality
  - Logic Function of each output pin
- Timing
  - Pin-to-Pin timing Information
- Power
  - State-dependent
  - Path-dependent
- Environmental Information
- Area
- Design Rules

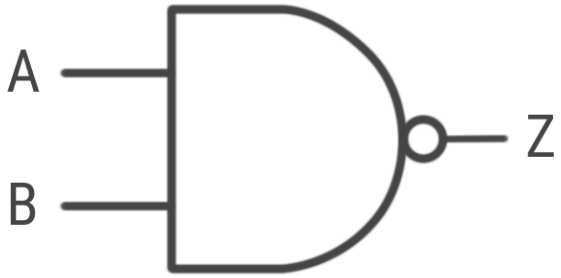
General Combinational Block

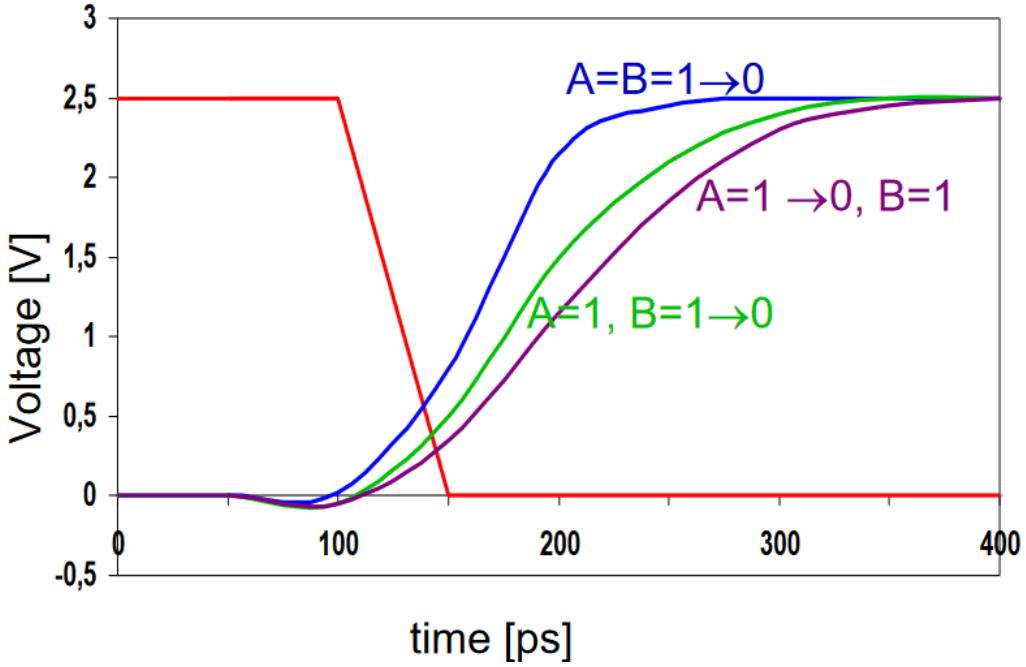


General Sequential Block



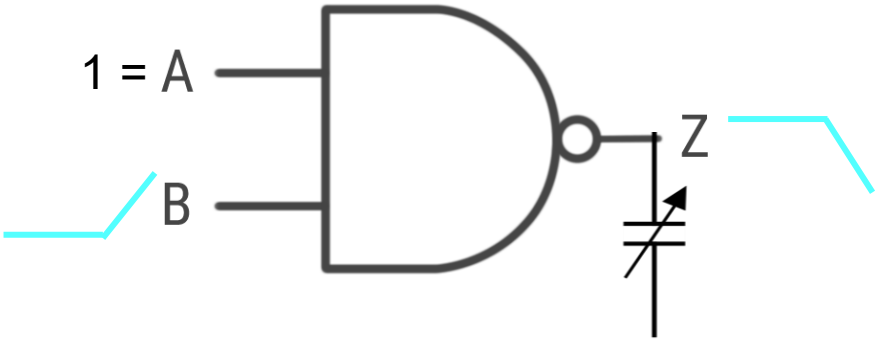
NAND2 Timing Behavior

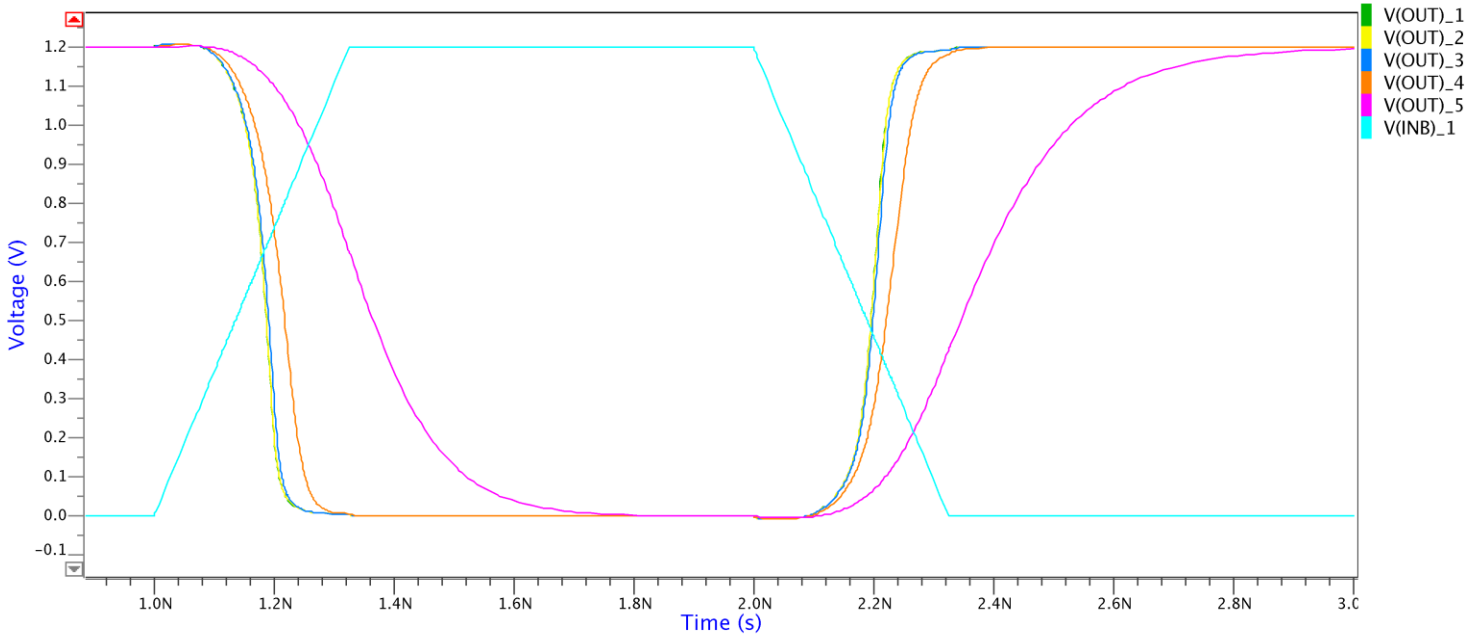




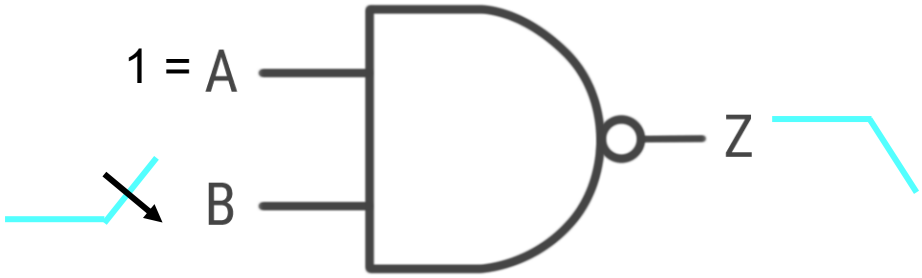
Inputs	Output	Delay (ps)
A=1, B=0->1	Z=1->0	69
A=1, B=1->0	Z=0->1	57
A=0->1, B=1	Z=1->0	62
A=1->0, B=1	Z=0->1	76

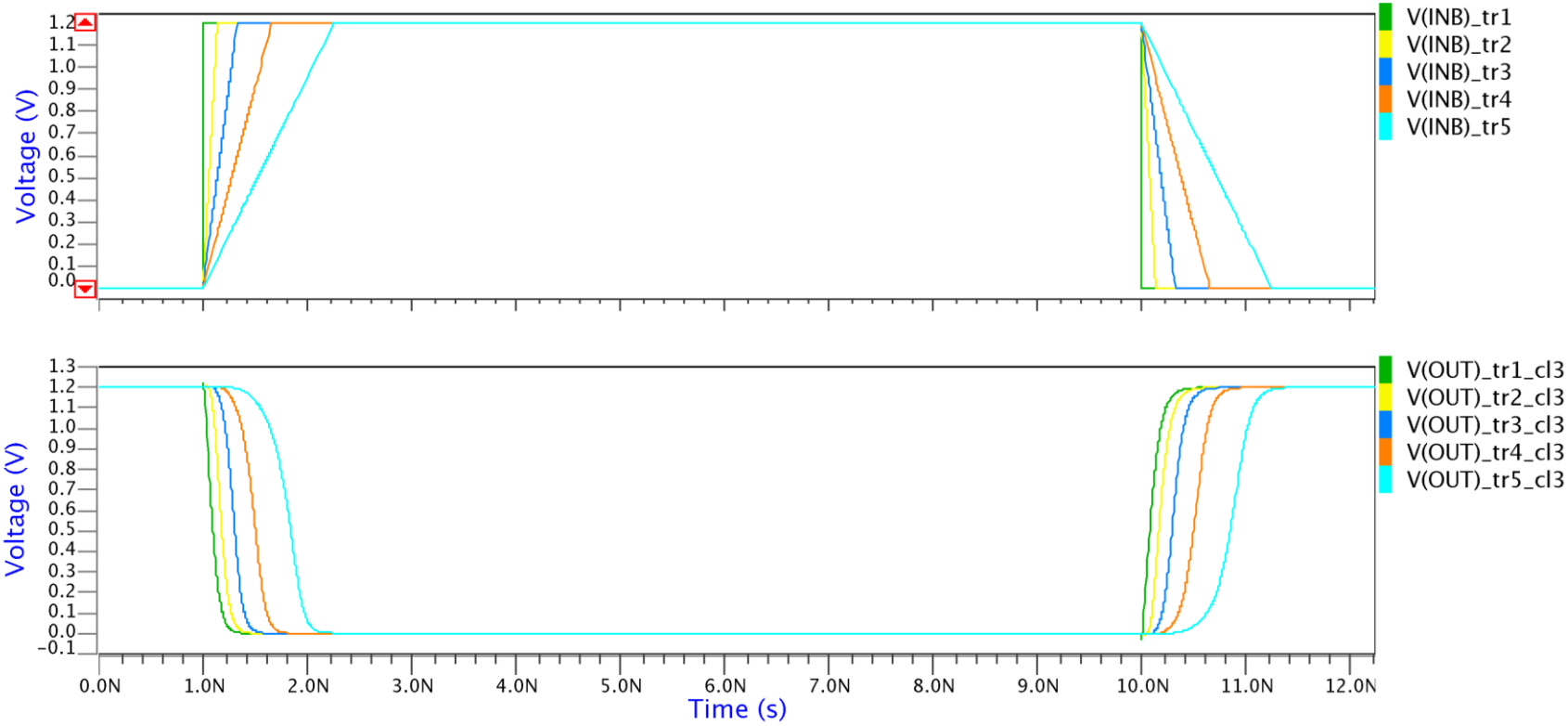
NAND2 Timing Behavior



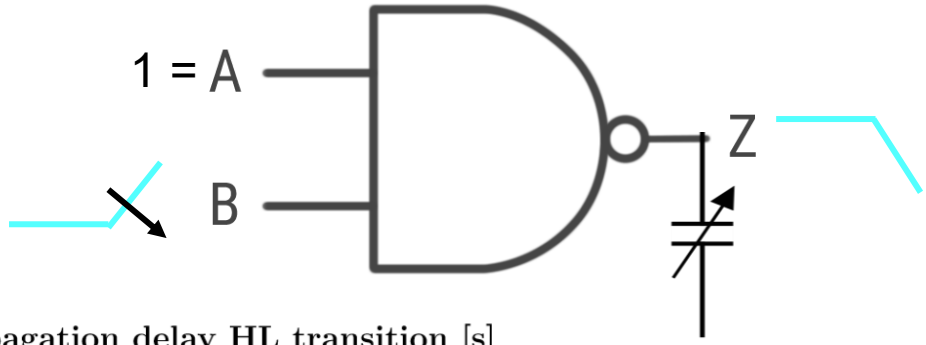


NAND2 Timing Behavior





NAND2 Timing Behavior



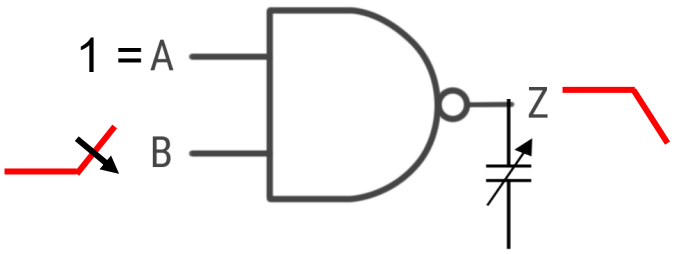
Propagation delay HL transition [s]

Transition time $[ns]$ \ $C_{load}[fF]$	0.004	0.012	0.028	0.08	0.16
0.0048	1.0425E-11	1.0446E-11	1.0486E-11	1.0627E-11	1.0841E-11
0.1088	1.8291E-11	1.8333E-11	1.8417E-11	1.8684E-11	1.9093E-11
0.2608	2.0517E-11	2.0575E-11	2.0691E-11	2.1067E-11	2.1634E-11
0.5248	1.9437E-11	1.9513E-11	1.9663E-11	2.0149E-11	2.0889E-11
0.1	1.2433E-11	1.2528E-11	1.2719E-11	1.3335E-11	1.4274E-11

Propagation delay LH transition [s]

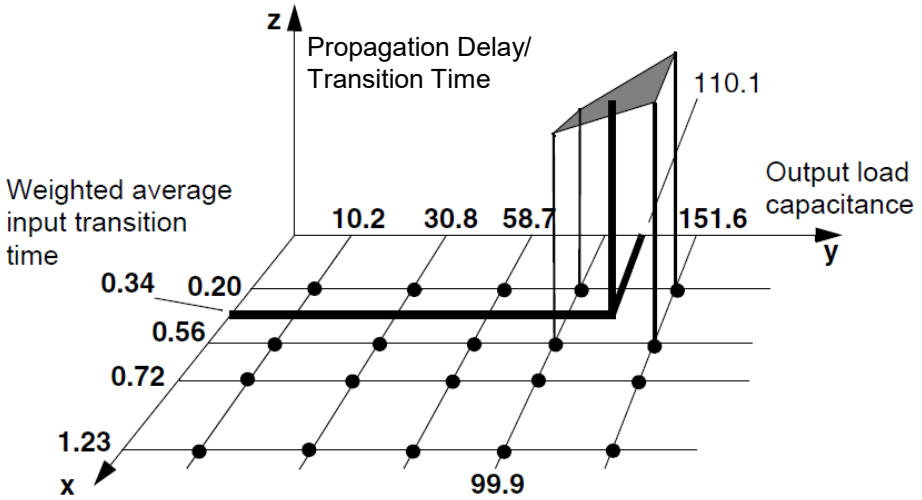
Transition time $[ns]$ \ $C_{load}[fF]$	0.004	0.012	0.028	0.08	0.16
0.0048	1.1445E-11	1.1469E-11	1.1520E-11	1.1690E-11	1.1946E-11
0.1088	2.5556E-11	2.5600E-11	2.5686E-11	2.5963E-11	2.6383E-11
0.2608	3.7941E-11	3.8001E-11	3.8121E-11	3.8509E-11	3.9098E-11
0.5248	5.4063E-11	5.4143E-11	5.4303E-11	5.4820E-11	5.5605E-11
0.1	7.7218E-11	7.7323E-11	7.7531E-11	7.8203E-11	7.9227E-11

Non-Linear Delay Model



foreach output_pin
  foreach input_pin
    foreach transition/delay
      foreach rise/fall

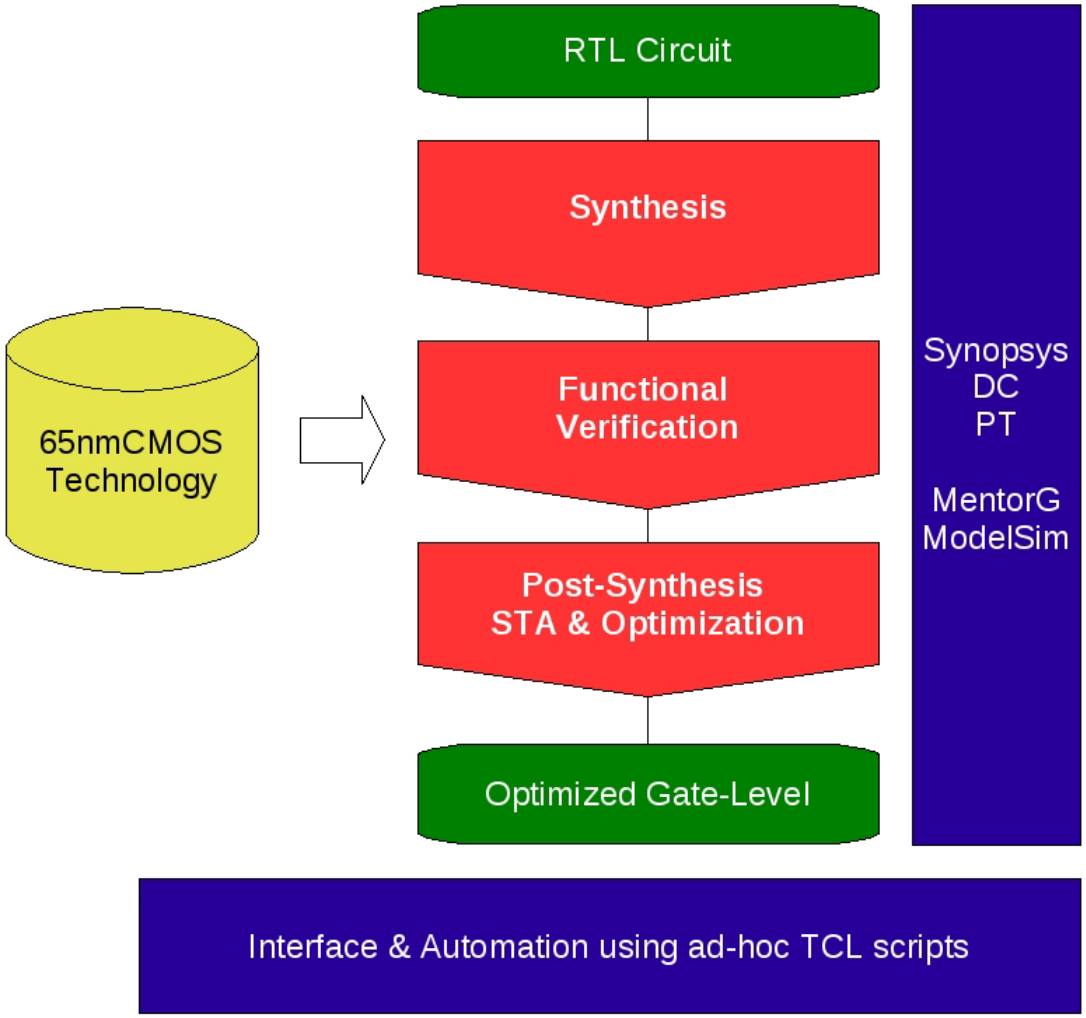
Input transition time \ Output capacitance				



$$T_{xy} = \frac{1}{(x_2 - x_1) \cdot (y_2 - y_1)} \cdot (T_{11} \cdot (x_2 - x) \cdot (y_2 - y) + T_{21} \cdot (x - x_1) \cdot (y_2 - y) + T_{12} \cdot (x_2 - x) \cdot (y - y_1) + T_{22} \cdot (x - x_1) \cdot (y - y_1))$$

Synopsys
Synthesis Flow

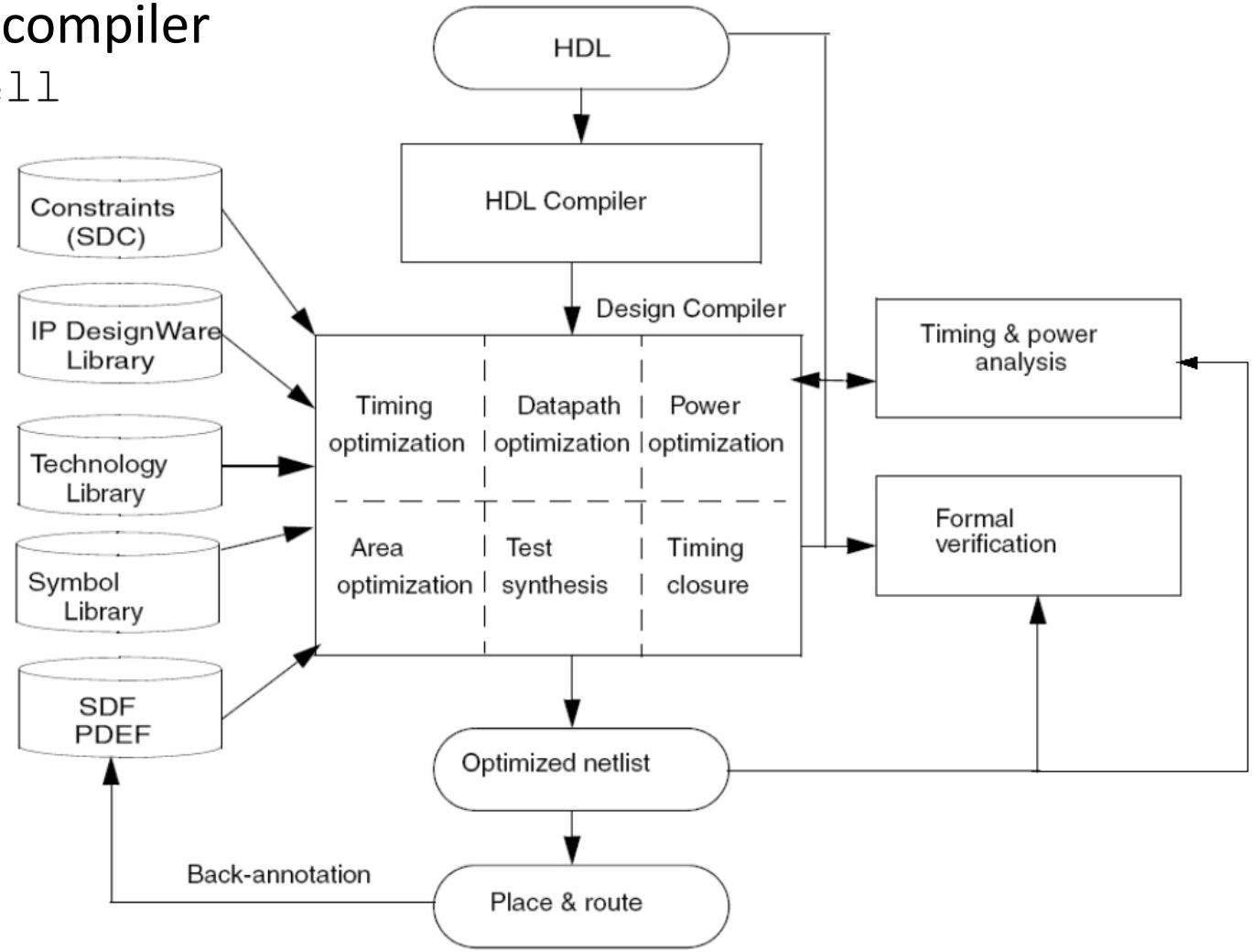
Logic Design



Logic Design Flow from Synopsys

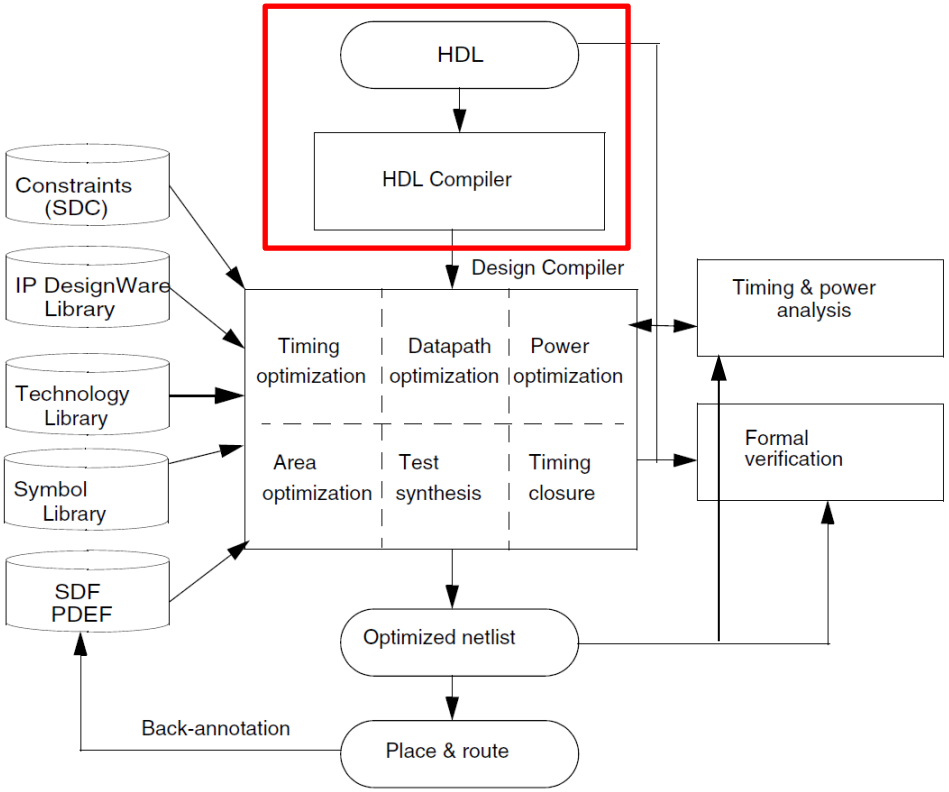
design-compiler

`dc_shell`



RTL Compiler (PRESTO)

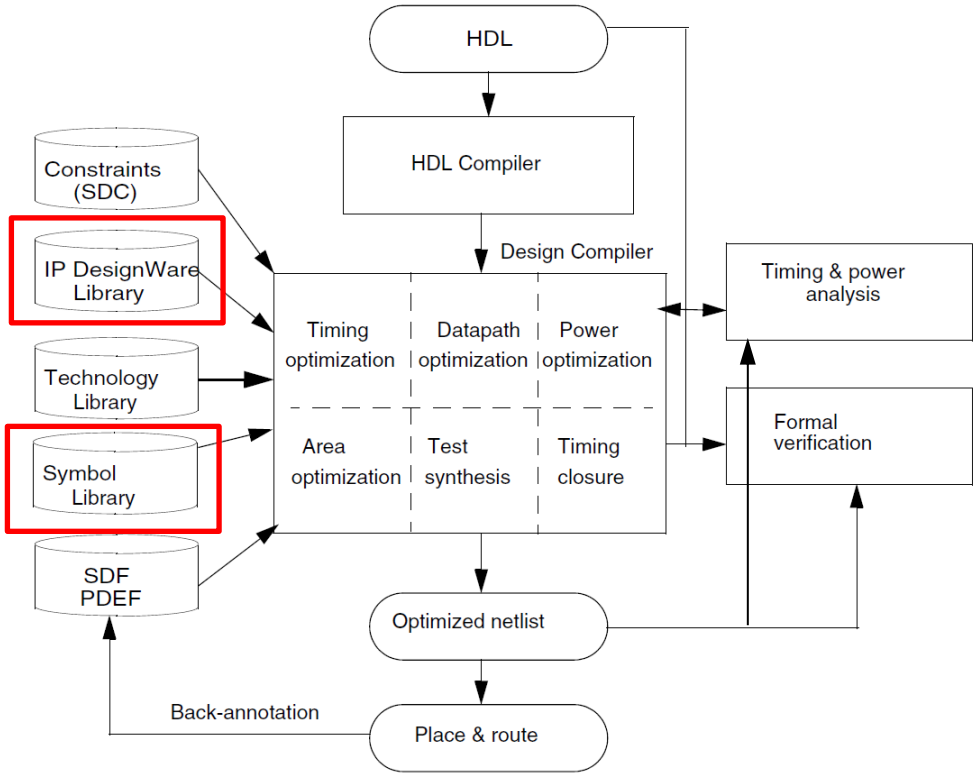
- **Logic synthesis and optimization is the process of converting a bhv. description written with some HDLs into an optimized gate-level netlist mapped with a specific technology library**



- *The input design files (bhv description) for Design Compiler are written using Verilog or VHDL*
- *HDL is compiled and translated into abstract internal models (logic networks, graphs and trees – AIG, BDD)*

Supporting Libraries

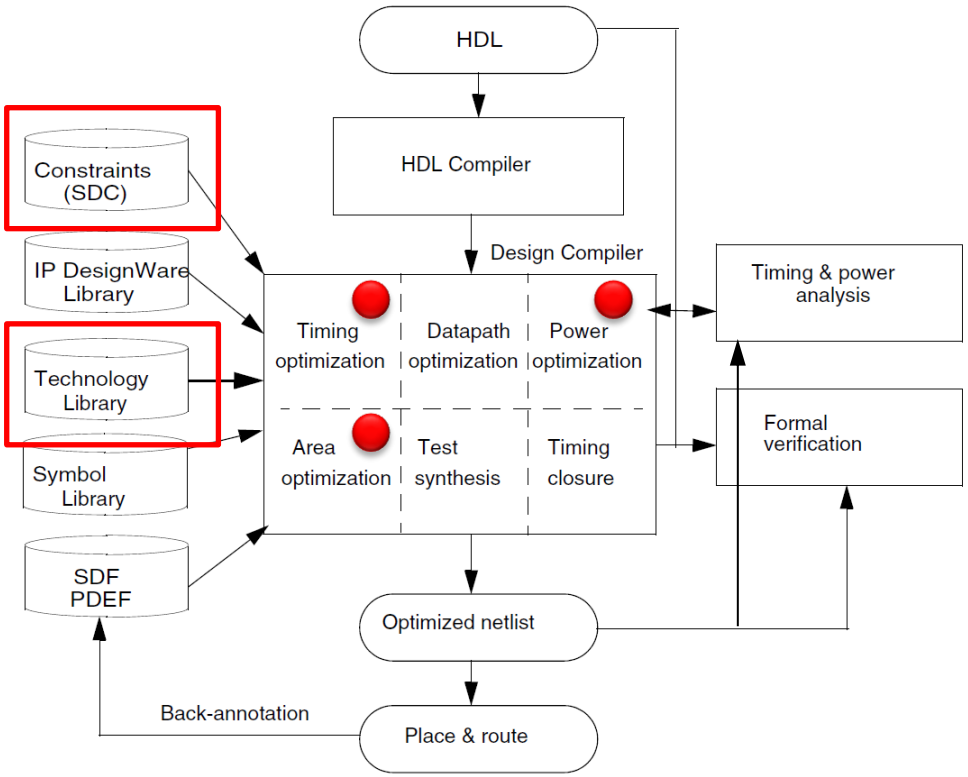
- **During the synthesis process, Design Compiler translates the abstract models into a generic gate-level netlist (gate taken from the generic library GTECH and the DesignWare library)**



- *The GTECH library consists of generic basic logic gates and flip-flops*
- *The DesignWare library contains more complex cells such as arithmetic cores*
- *The Symbol library is used to generate the design schematic*
- *The Technology library is used later*

Netlist Optimization

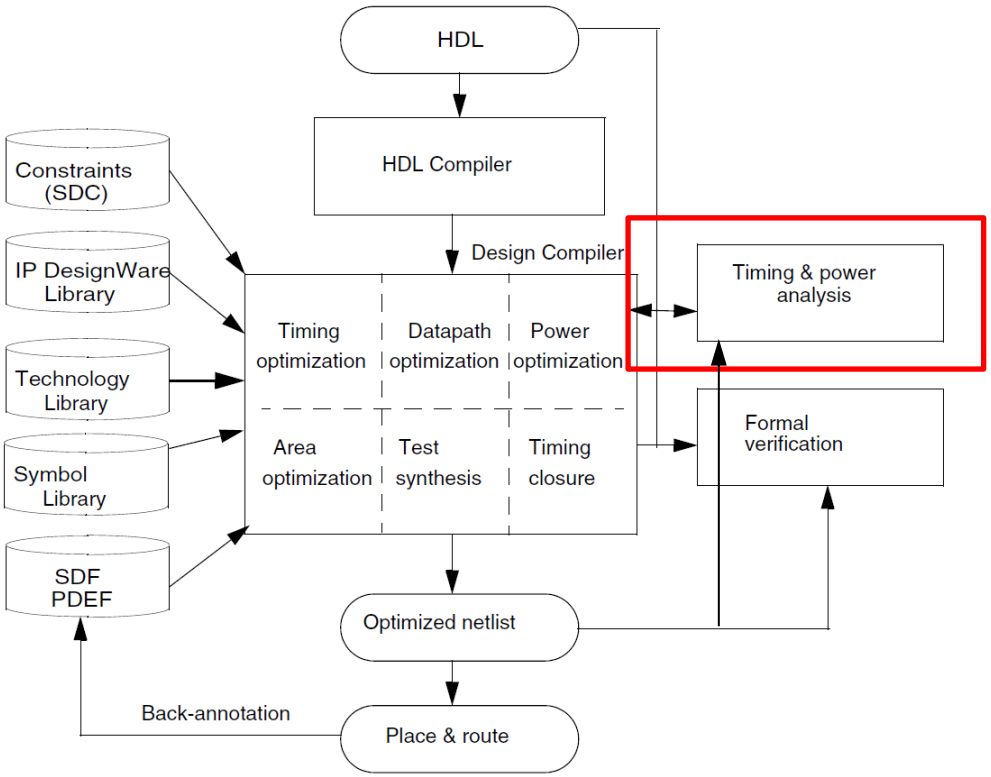
- **After the synthesis process, Design Compiler optimizes and maps the generic netlist into a specific technology library, known as the *target library*.**



- *The mapping process is constraint-driven (SDC file)*
- *The optimization phase refines the netlist in order to minimize*
  - *area*
  - *dynamic power*
  - *static power*
- *The Technology library is provided by silicon vendor*

Timing/Power Analysis

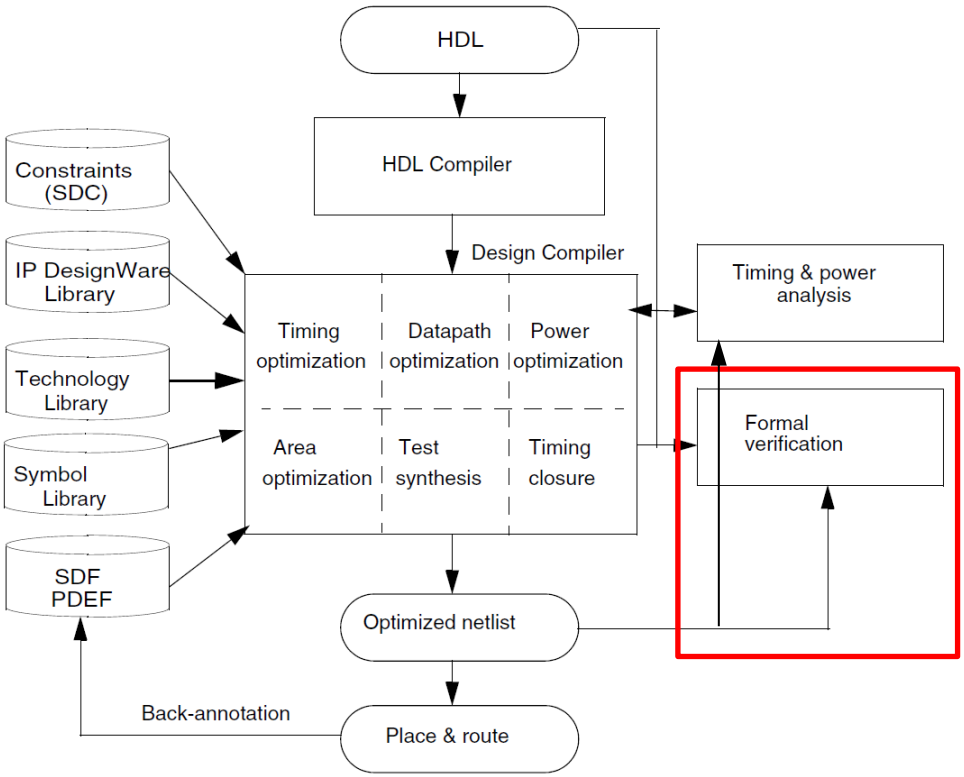
- **Static Timing/Power Analysis Engines help to verify timing/power compliance <u>during</u> the optimization process**



- *A dedicated static analysis engine is embedded in DC*
- *At each iteration it provides information about actual worst-case path delay and power-consumption*
- *If constraints are met the optimization process stops returning the optimized netlist*

Post-Synthesis Formal Verification

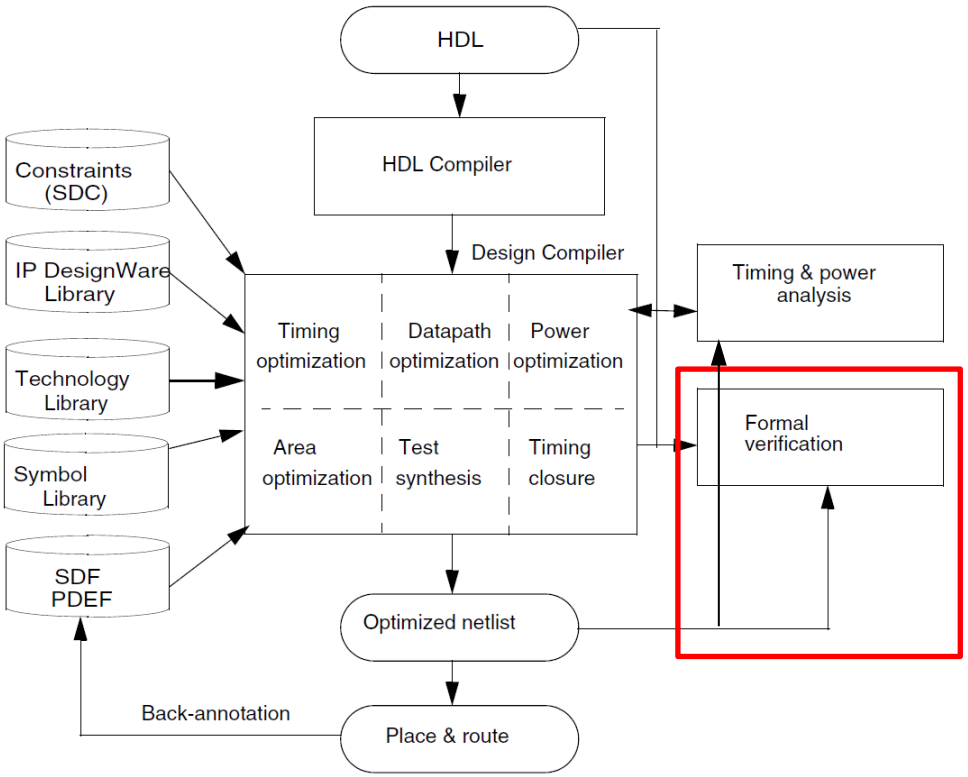
- **After optimization, the functionality of the gate-level netlist is validated**



- *Formal Verification*
  - *mathematical methods*
  - *the design is compared against a reference design model*
  - *Structural and Boolean equivalence check*
  - *Synopsys Formality*

Post-Synthesis Functional Verification

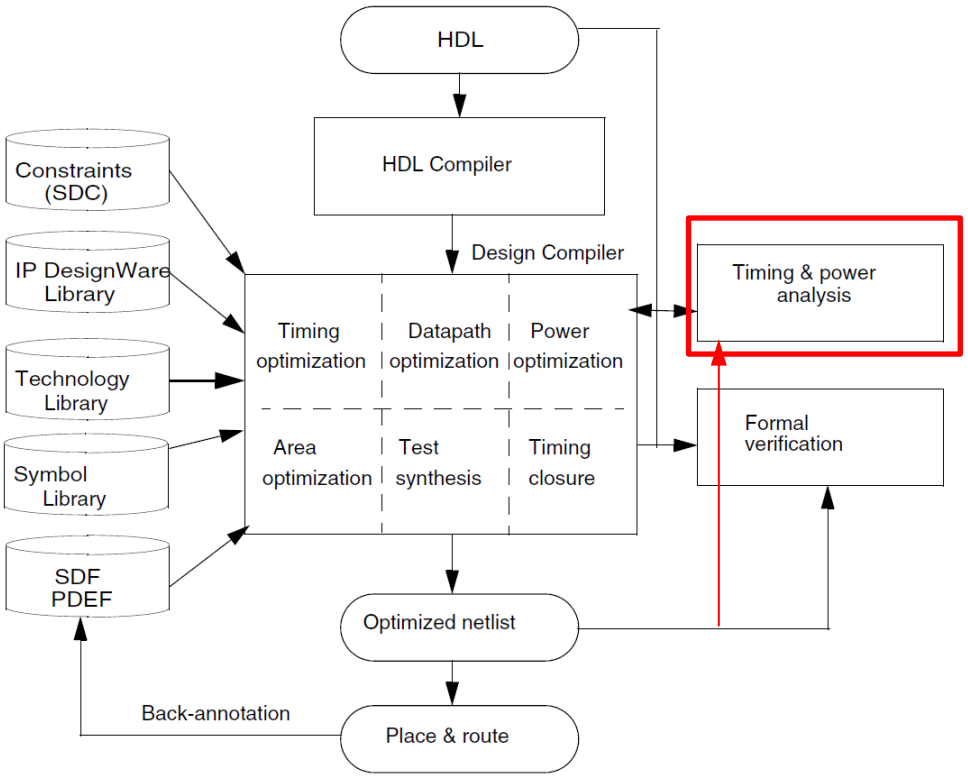
- **After optimization, the functionality of the gate-level netlist is validated**



- *Functional Verification*
  - *simulation based*
  - *time-consuming*
  - *gate-level accuracy slows down the simulation*
  - *coverage*
  - *post-synthesis simulation is used to estimate the switching activity of internal nodes under realistic work-load*

Post-Synthesis Static Timing Analysis

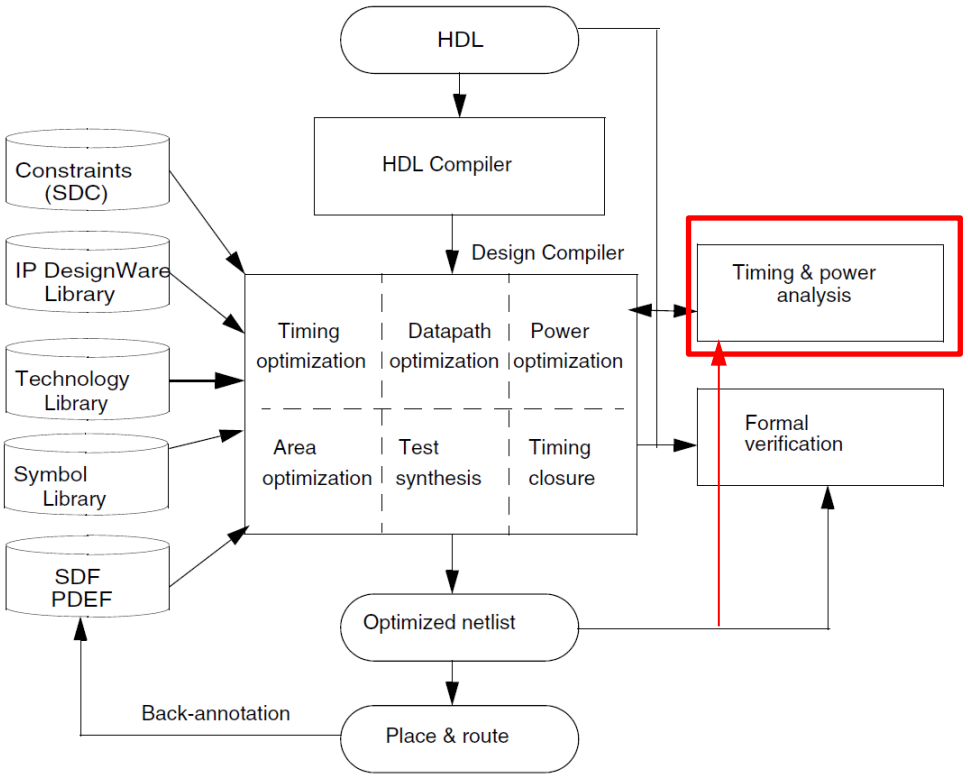
- **After obtaining the optimized gate-level netlist, we check the design performance through post-synthesis timing/power static analysis**



- *PrimeTime*
  - *stand-alone*
  - *sign-off accuracy (parasitic annotation)*
  - *manage full-chip*
  - *extremely fast*
- *PrimeTime allows the user to exhaustively analyze*
  - *critical paths distribution*
  - *fanout and cap. load of each net*
  - *.....*

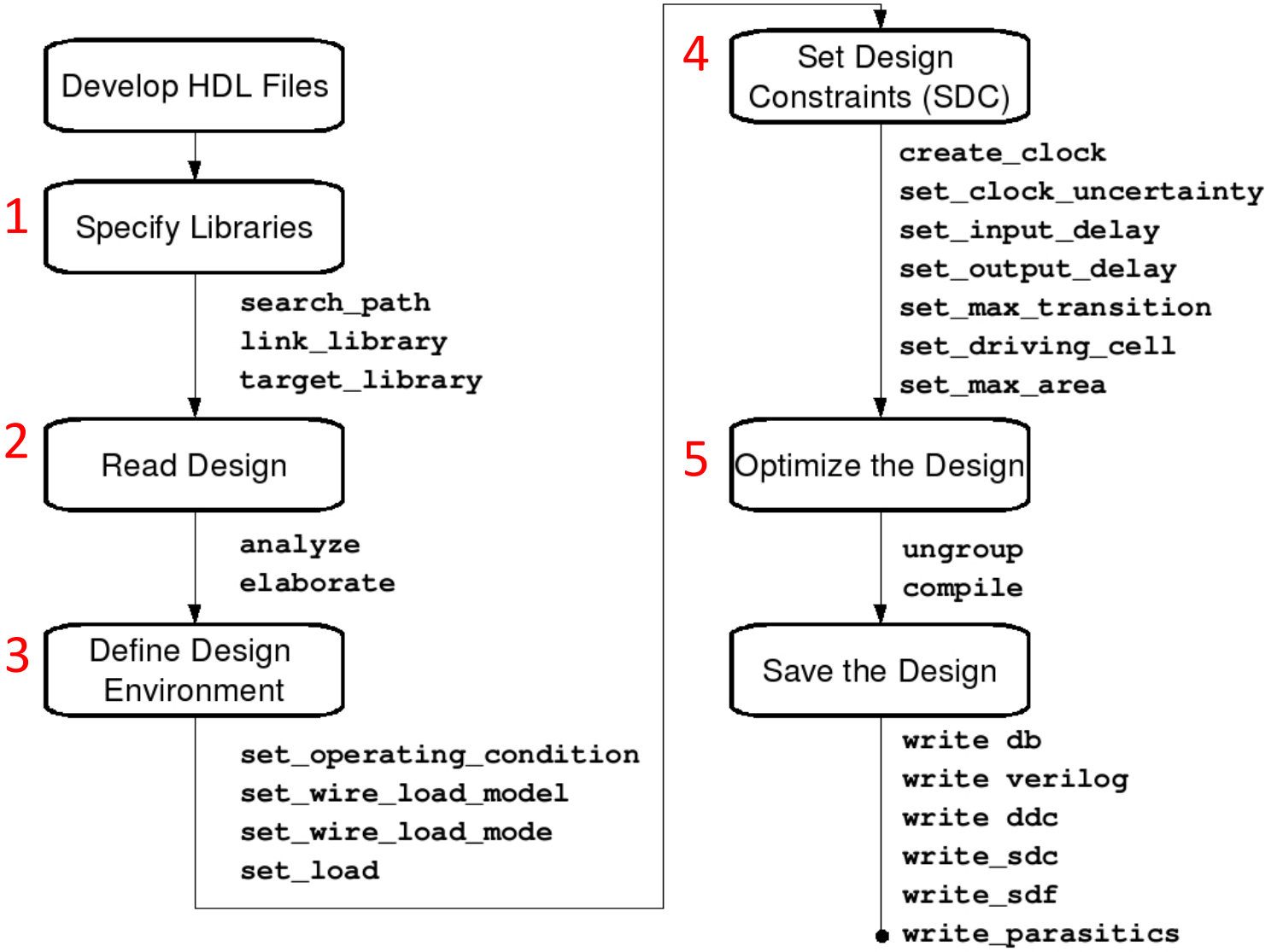
Post-Synthesis Static Power Analysis

- **After obtaining the optimized gate-level netlist, we check the design performance through post-synthesis timing/power static analysis**

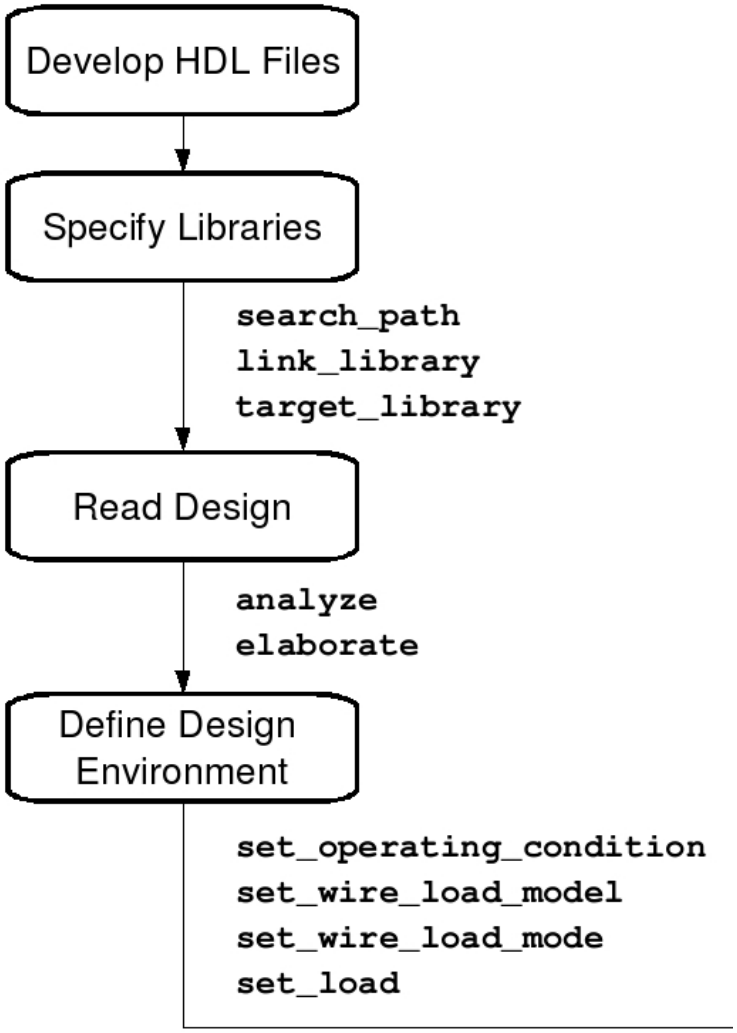


- *PrimePower is a static analysis tool*
  - *Embedded into PT*
  - *simulation of the effects of wires*
  - *manage full-chip power analysis*
  - *extremely fast*
- *Prime Power allows the user to exhaustively analyze*
  - *dynamic power using probabilistic models*
  - *leakage power analysis*

Synthesis: Main Flow and Commands



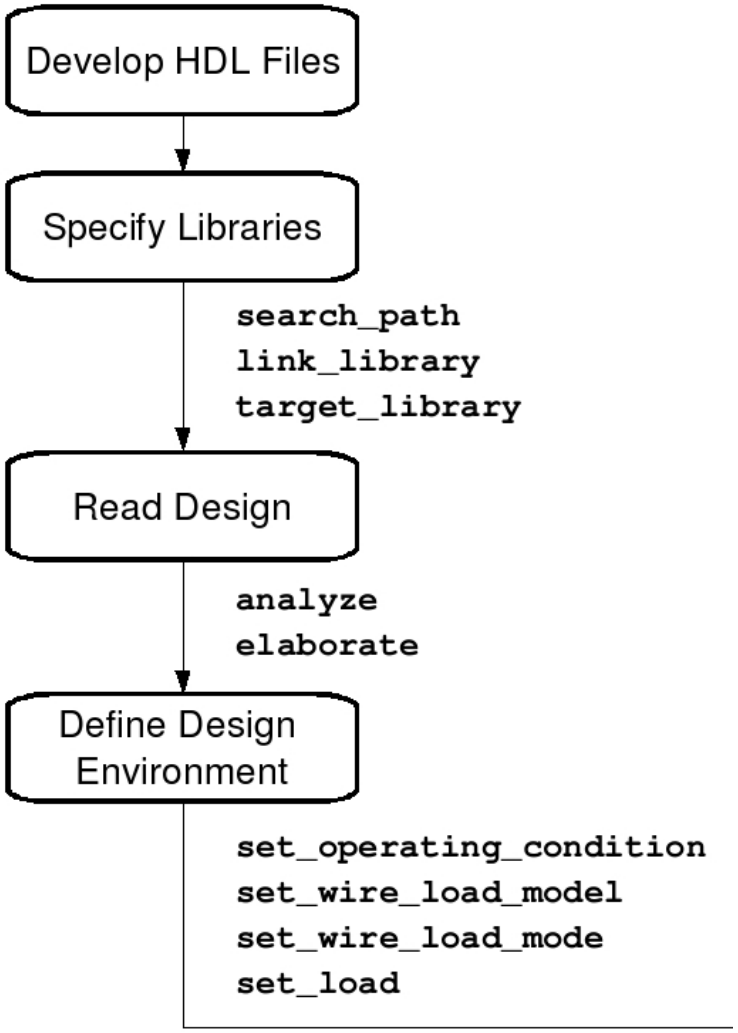
Specify Libraries



`search_path`

- Specifies directories that Design Compiler search for files specified without directory names
- The search includes looking for technology and symbol libraries, design files, and so on
- The value of this variable is a list of directory names and is usually set to a central library directory

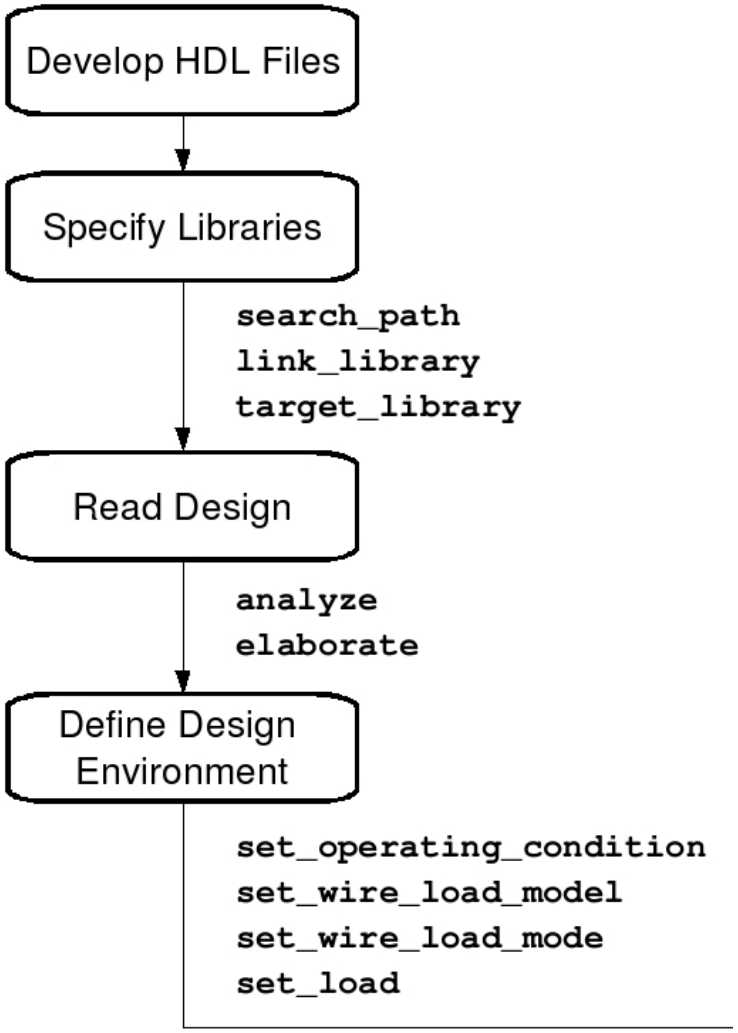
Specify Libraries



`link_library`

- **Specifies the list of design files and libraries used during linking**
- **The link command looks at those files and tries to resolve references in the order of specified files**

Specify Libraries



`target_library`

- **Specifies the list of technology libraries of components to be used when compiling a design**
- **The target library name should also be included in the `link_library` list**

Technology Libraries

CORE65LPSVT_nom_1.00V_25C

```
cell(HS65 LS IVX2){
area : 1.56;
rail_connection(vdd, vdd);
Leakage_power(){
    power_level : "vdd";
    value : 6.5078e-08;
    when : "!A"; }
pin(Z){
    capacitance : 0;
    direction : output;
    internal_power() {
    fall_power(power_table_7){
    values("-1.083150e-04, -9.550195e-05",
        "... ... ... ... ... ",
        "-1.224050e-04, -1.491749e-04");}
    timing(){
    cell_fall(table_6)}
    values("0.019924, 0.036924, 0.061583, 0.113250"
        "... ... ... ... ... ... ... ... ...");
        "0.220400, 0.299120, 0.370530, 0.474560");}
}
```

- Provided by silicon vendors
- Contain standard cells description and characterization:
  - reference name
  - logical function
  - pins name
  - area
  - leakage power
  - dynamic power
  - propagation delay

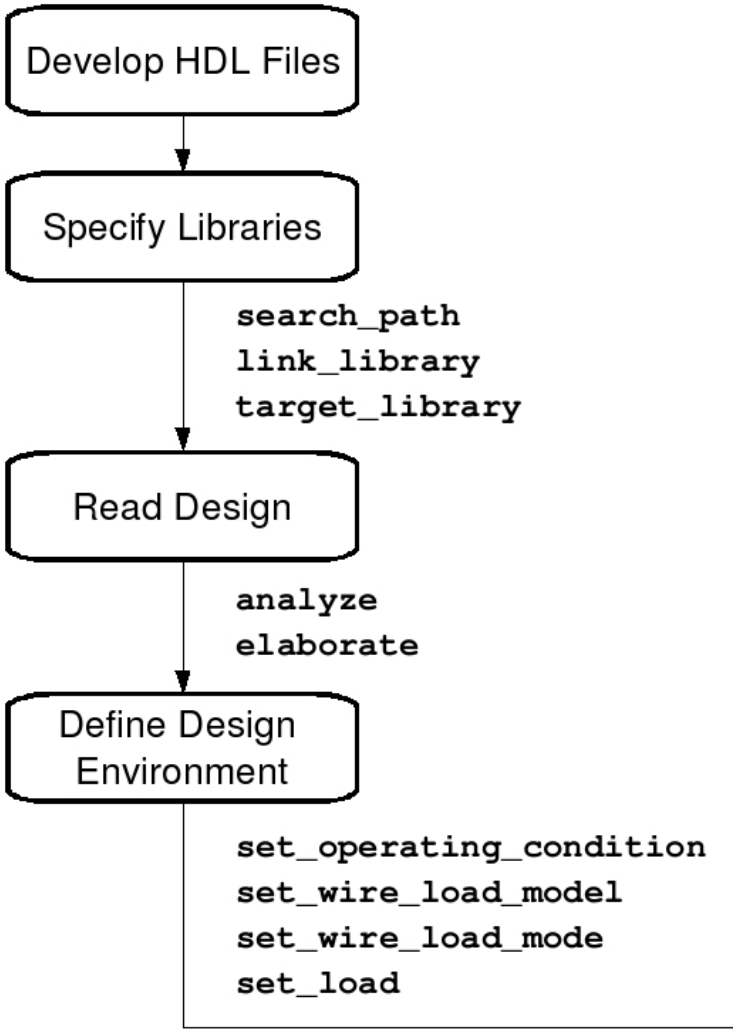
Library Operating Conditions

Each library contains the characterization of each standard cell for a particular PVT corner:

- ***Process*: fabrication yield**
- ***Voltage*: nominal supply voltage**
- ***Temperature*: on-chip operating temperature**

Library Name	*Process*	*Voltage*	*Temperature*
CORE65LPSVT_bc_1.35V_m40C.db	best (fast)	1.35V	-40°C
CORE65LPSVT_nom_1.00V_25C.db	nominal	1.00V	25°C
CORE65LPSVT_wc_1.05V_105C.db	worst (slow)	1.05V	105°C

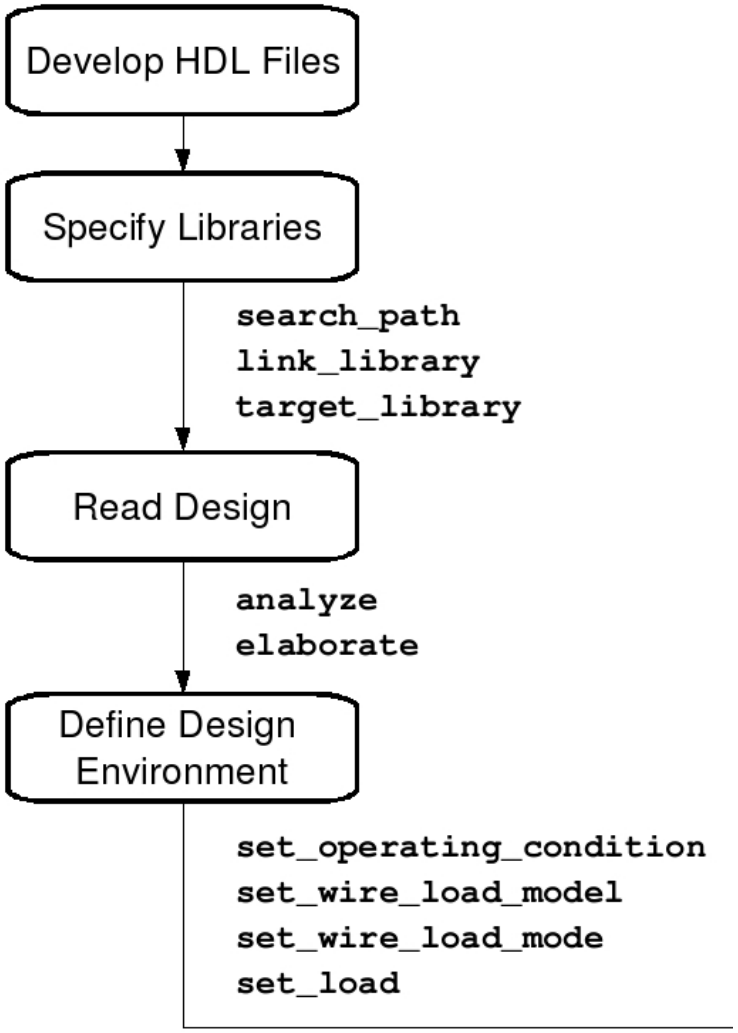
Read Design



`analyze/elaborate`

- **Initially analyzes the design for syntax errors**
- **Translates the RTL description using the generic logic**
- **The generic logic or GTECH components are part of the Synopsys generic technology independent library**

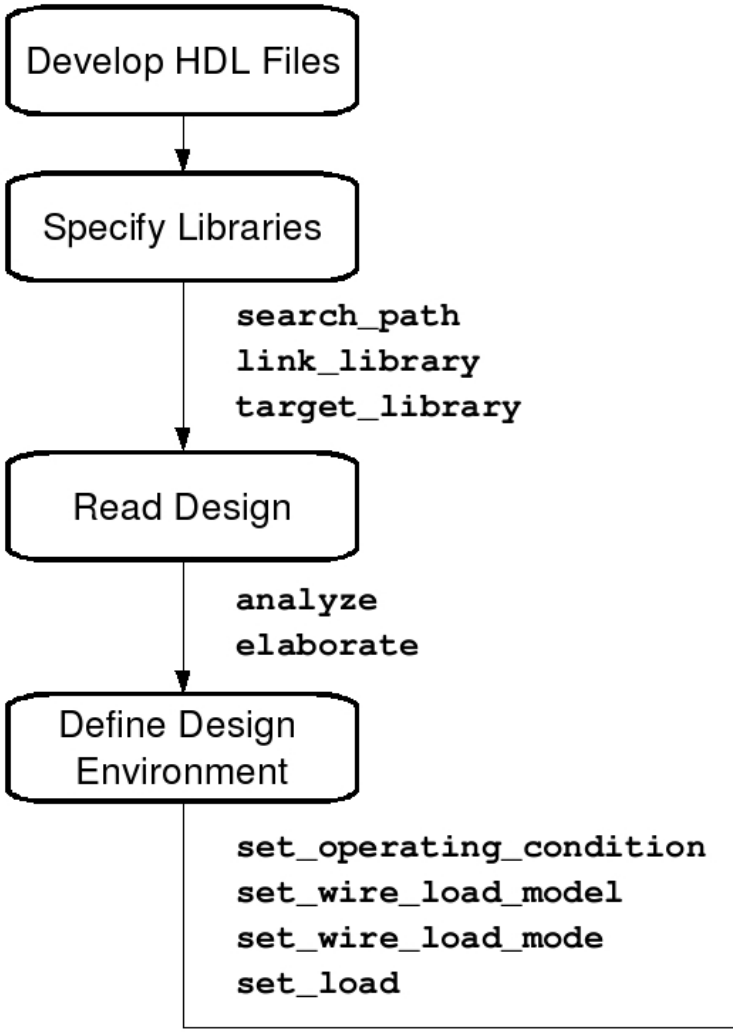
Define Design Environment



`set_operating_condition`

- **Describes the process, voltage and temperature (PVT) conditions of the design**
- **The technology library contains the description of these conditions**
- **The names of operating conditions are library dependent (e.g., nom_1.00V_25C)**
- **The operating conditions specified must be defined in lib or in one of the libraries in the link library**

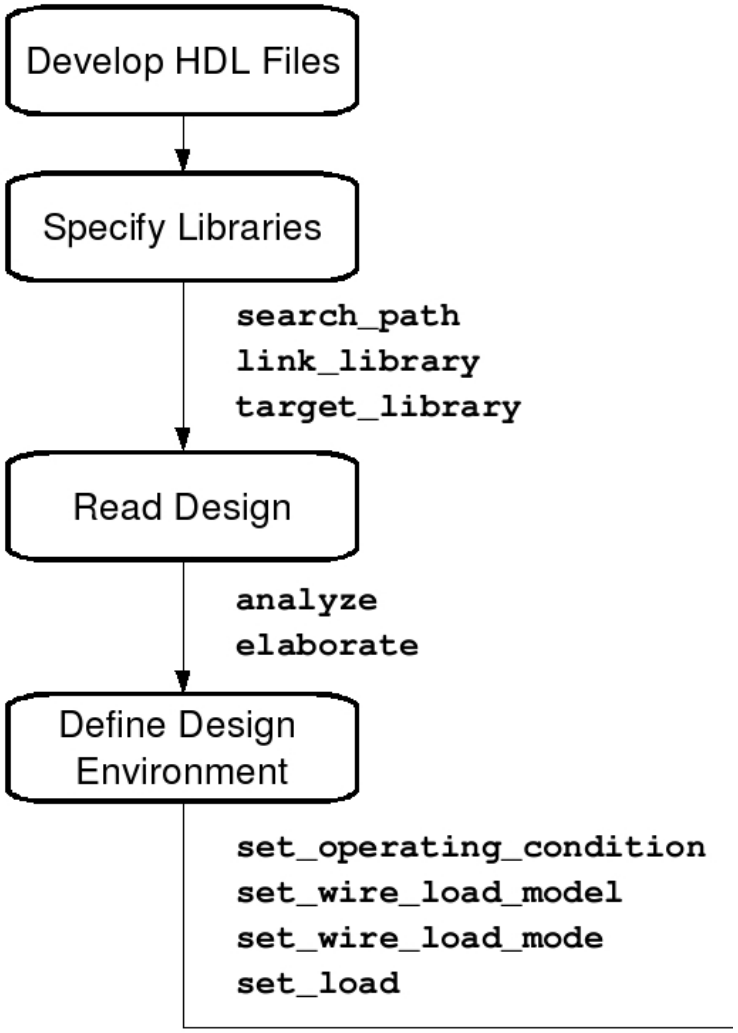
Define Design Environment



`set_wire_load_model`

- **This command is used to provide estimated statistical wire-load information to DC, which in turn, uses the wire-load information to model net delays as a function of loading**
- **Generally, a number of wire-load models are present in the technology library, each representing a particular size block (e.g., area_8k_to_12k)**

Define Design Environment



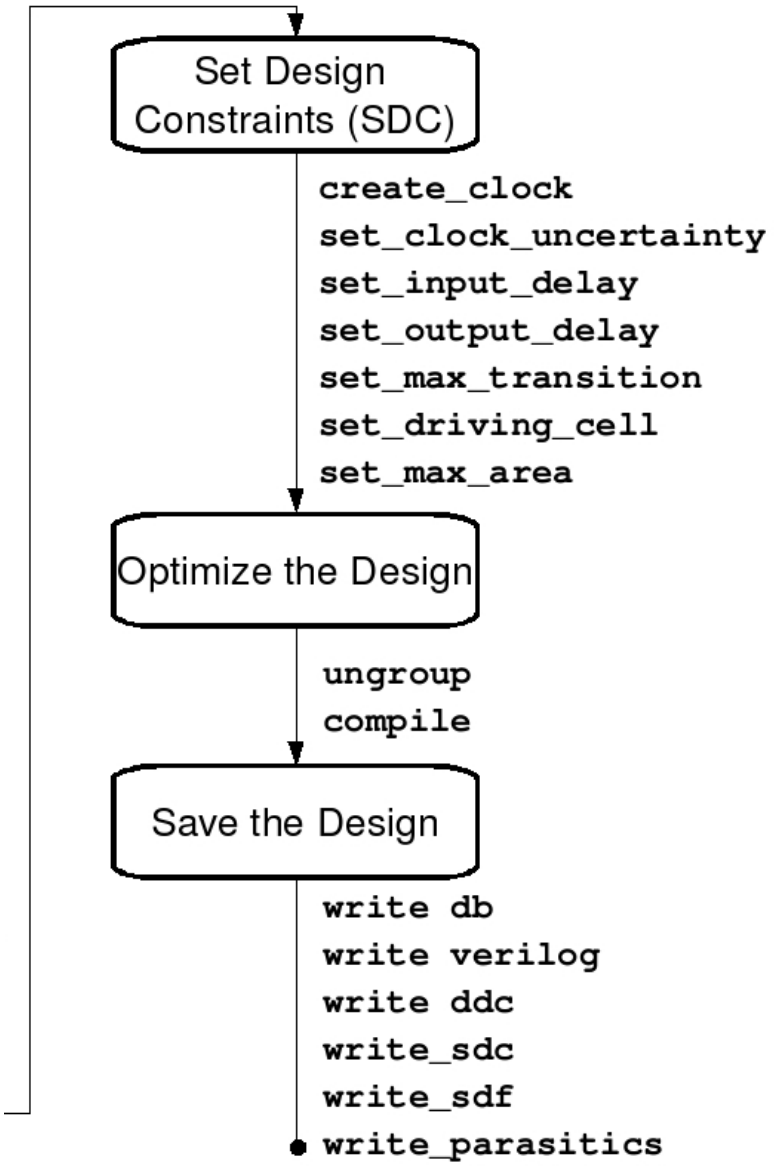
`set_load`

- **It sets the capacitive load in the units defined in the technology library (usually pico-farads, or pf), to the specified nets or ports of the design**
- **It typically sets capacitive loading on output ports of the blocks during pre-layout synthesis**

Set Design Constraints

`create_clock`

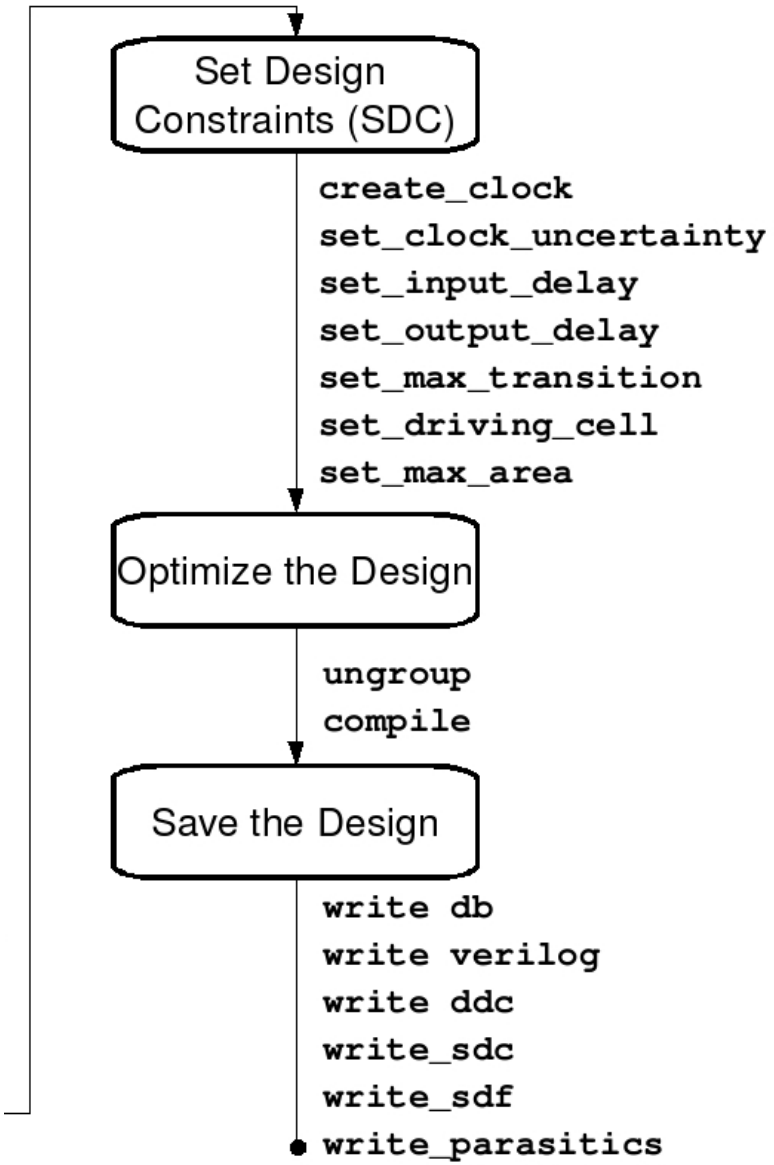
- Is used to define a clock object with a particular period and waveform
- A block may only contain combinational logic; to define delay constraints for this block, one can create a virtual clock and specify the input and output delays in relation to the virtual clock



Set Design Constraints

`set_clock_uncertainty`

- **Lets the user define the clock skew information**
- **Basically this is used to add a certain amount of margin to the clock, both for setup and hold times**

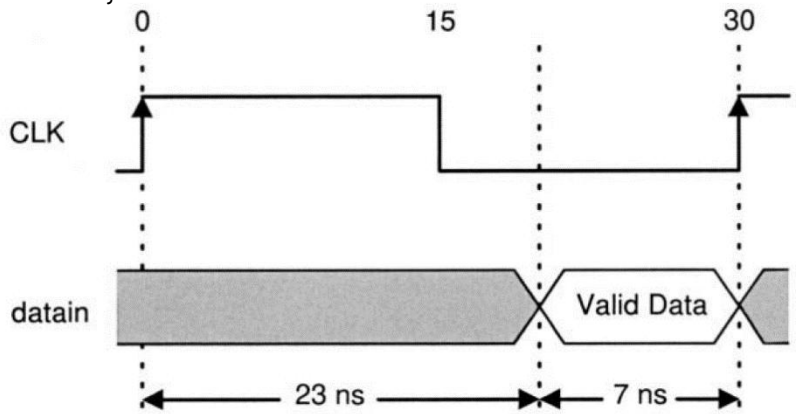


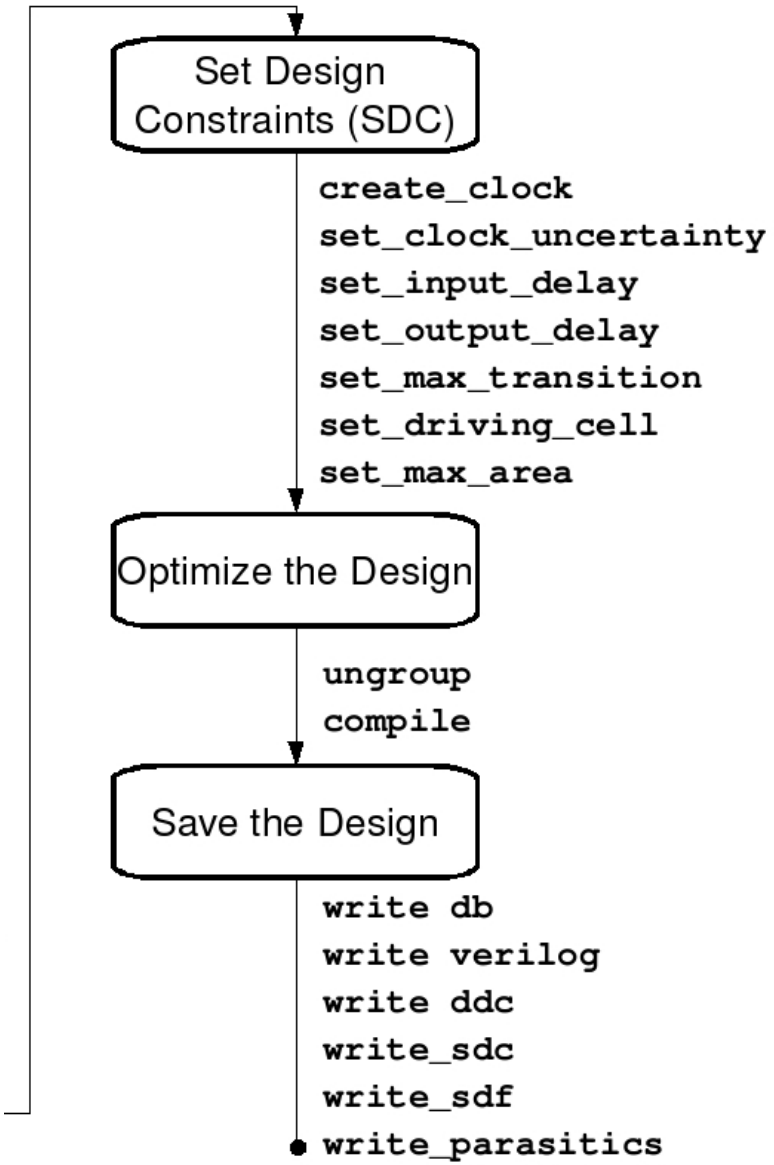
Set Design Constraints

`set_input_delay`

- **Specifies the input arrival time of a signal in relation to the clock**
- **It is used at the input ports, to specify the time it takes for the data to be stable after the clock edge**

```
set_input_delay max 23.0 clock CLK
{datain}
```



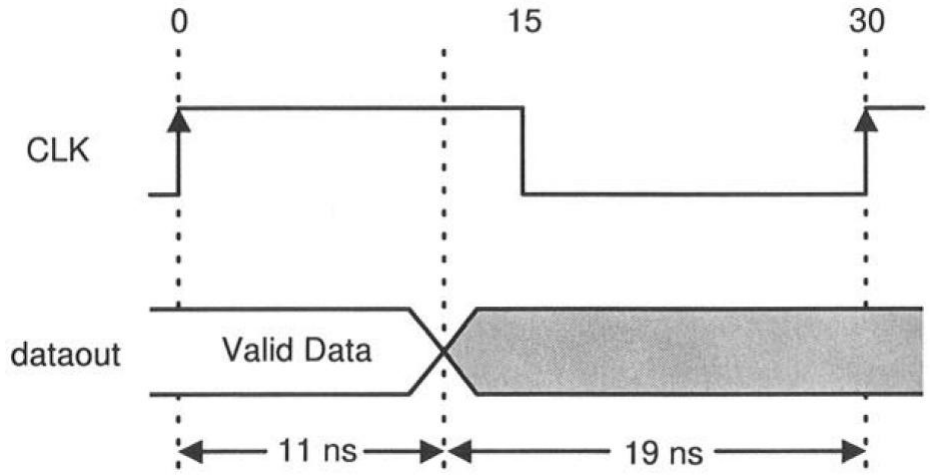


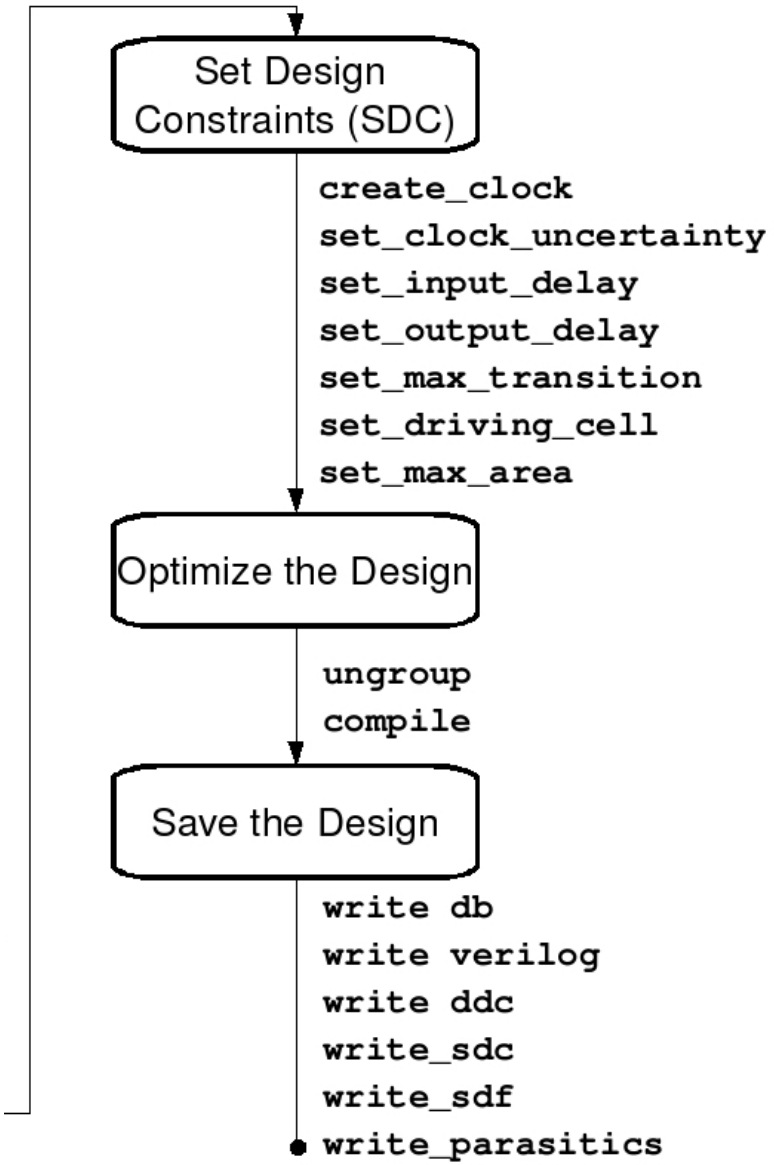
Set Design Constraints

`set_output_delay`

- **Is used at the output port, to define the time it takes for the data to be available before the clock edge**

```
set_output_delay max 19.0 clock CLK
{dataout}
```

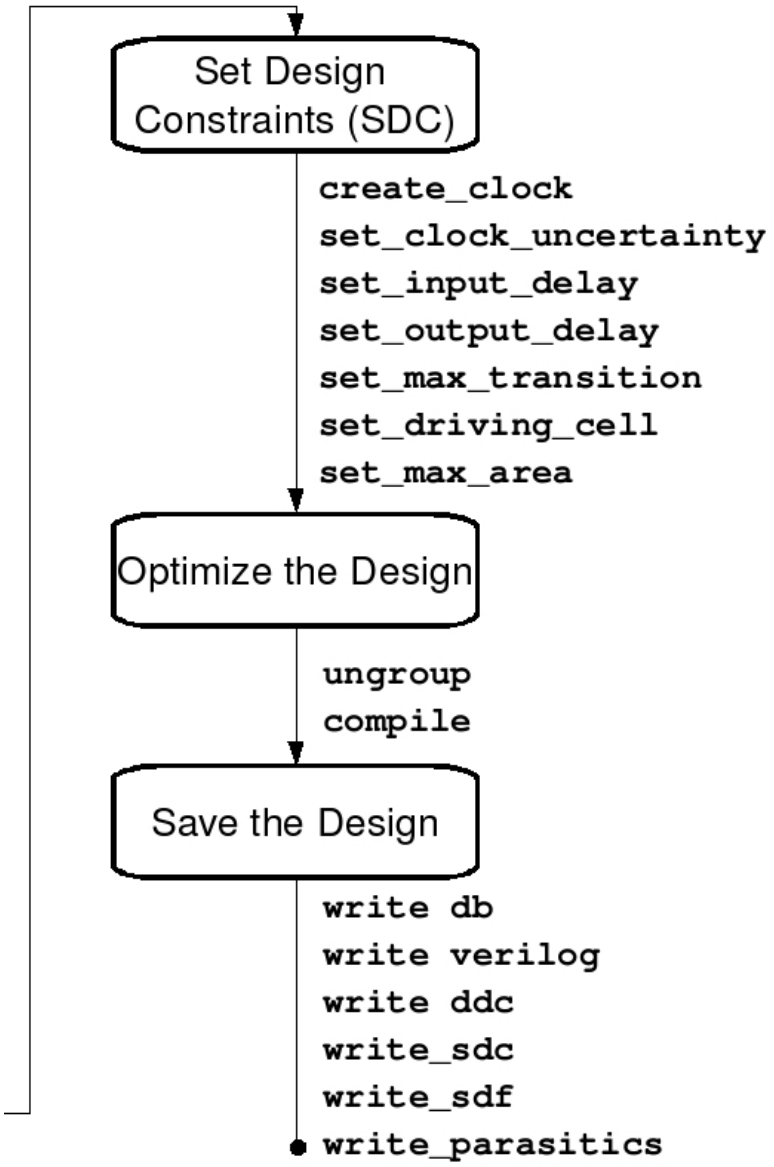




Set Design Constraints

`set_max_transition`

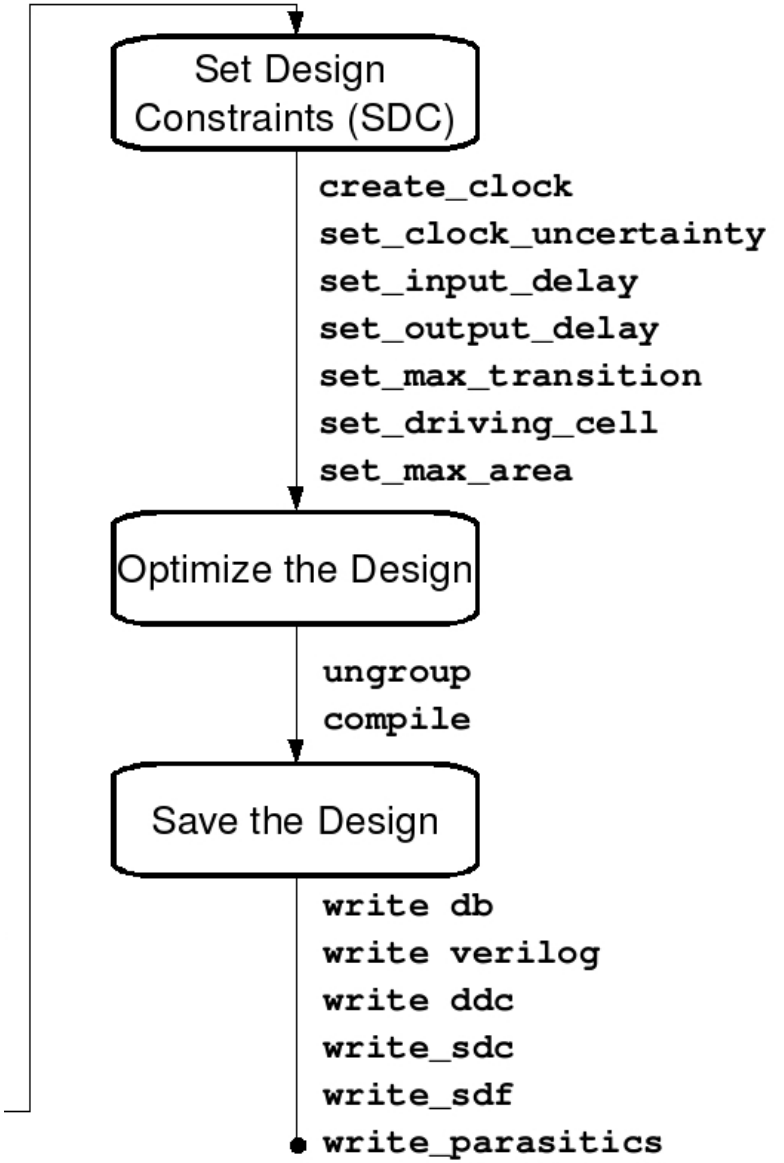
- Sets the max_transition attribute on the specified clock groups, ports or designs
- Compiler attempts to ensure that the transition time for a net is less than the specified value
- The max transition constraint is a design rule constraint



Set Design Constraints

`set_driving_cell`

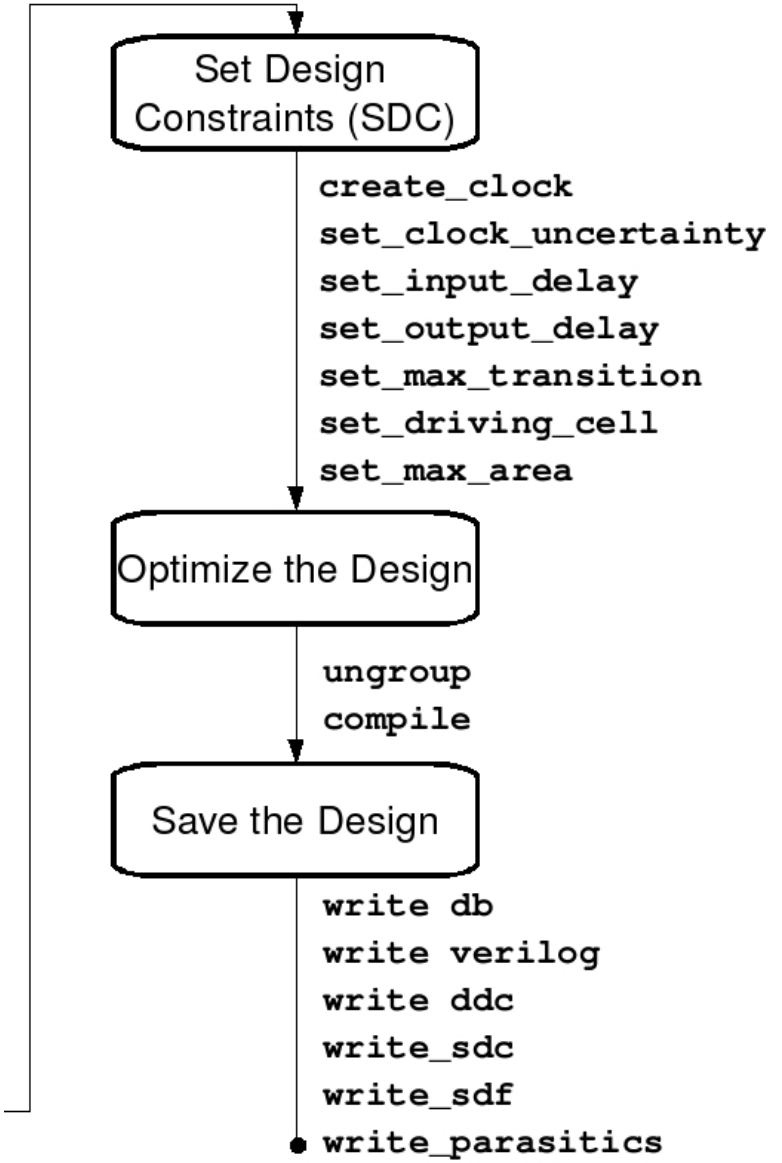
- Is used to model the drive resistance of the driving cell to the input ports
- This command takes the name of the driving cell as its argument and applies all design rule constraints of the driving cell to the input ports of the block



Set Design Constraints

`set_max_area`

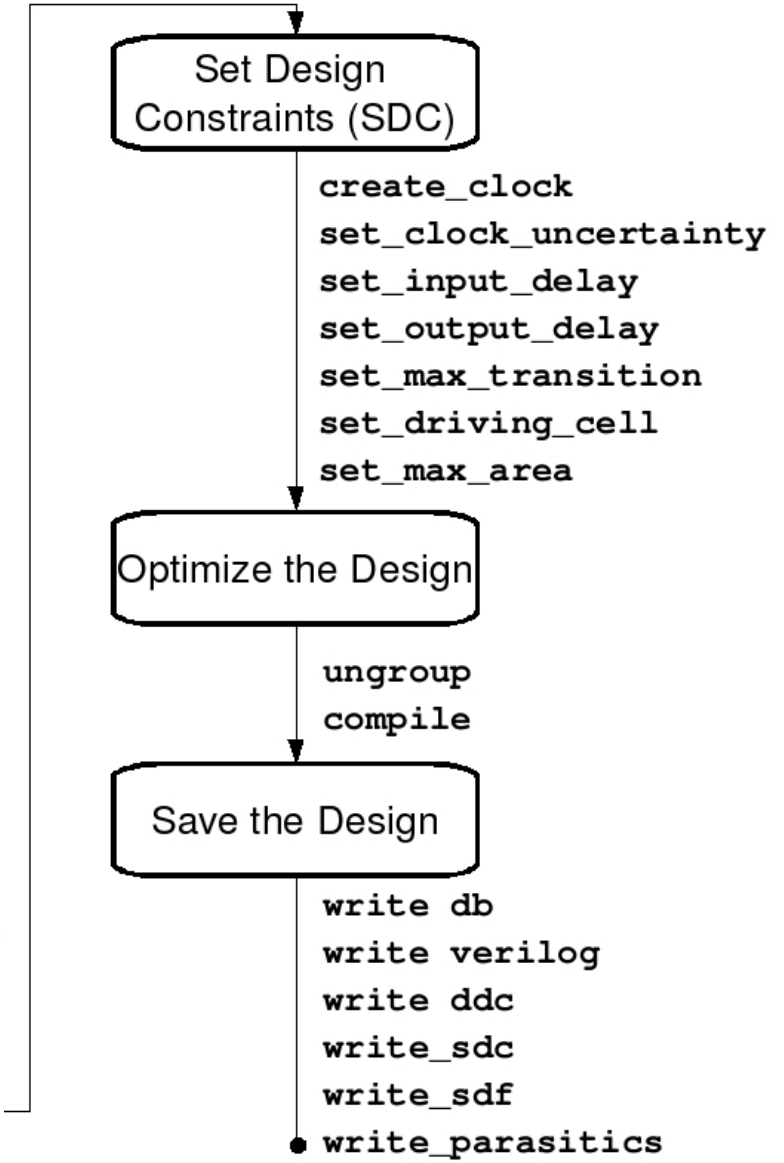
- **Sets the max area attribute to a specified value on the current design**
- **The max area attribute represents the target area of the design and is used by the compile command to calculate area cost of the design**



Optimize the Design

`ungroup`

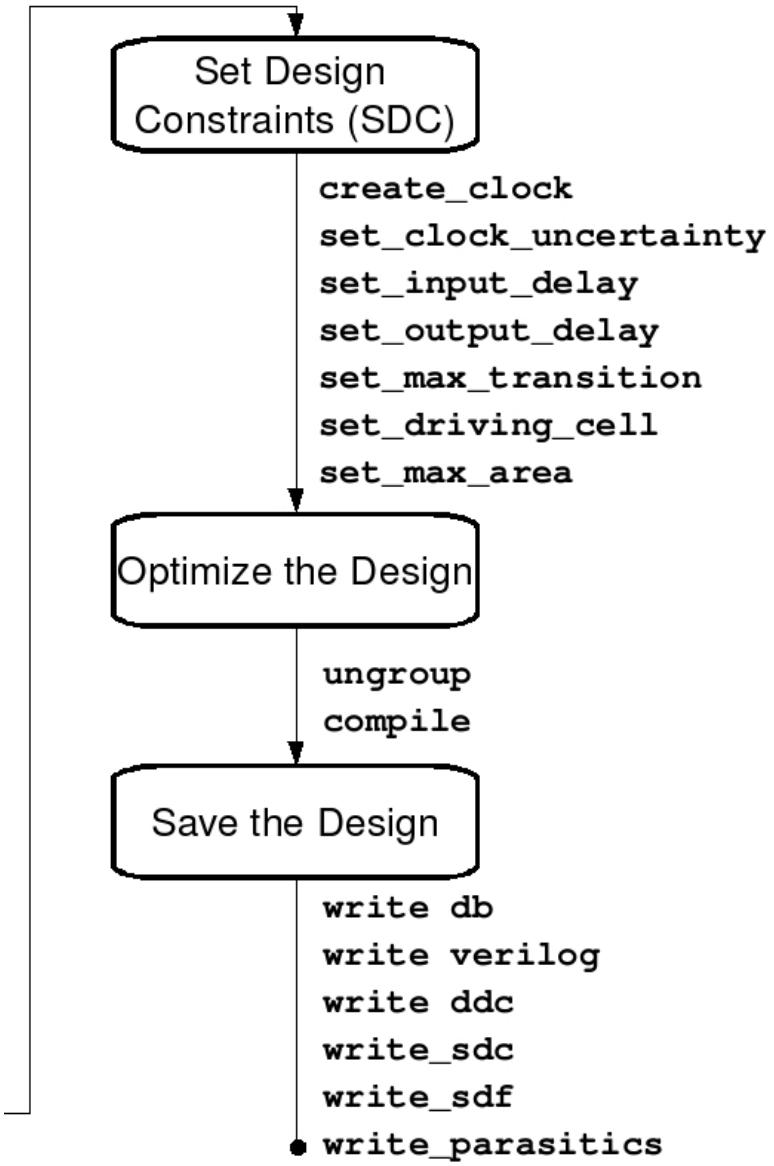
- **Provides the designer with the capability of altering the partitions**
- **It is used to remove the design hierarchy obtaining a flat netlist**



Optimize the Design

`compile`

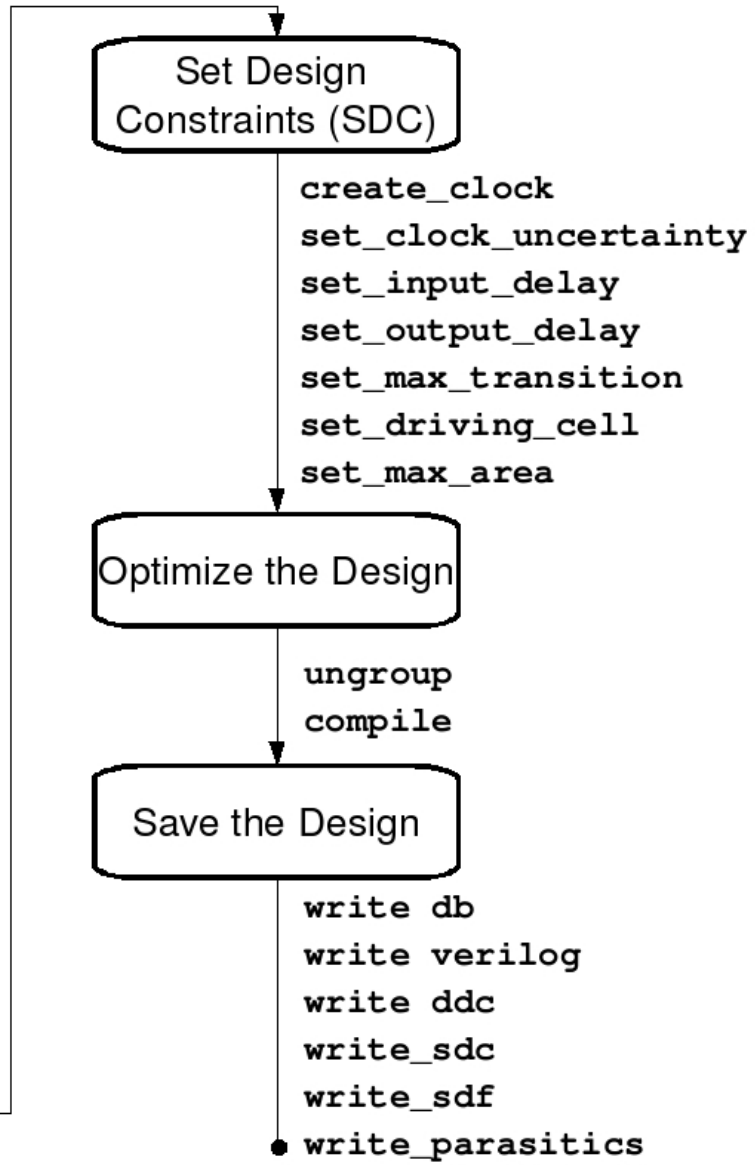
- **Performs logic-level and gate-level synthesis and optimization on the current design**
- **Optimization is controlled by user-specified constraints on the design**
- **These constraints describe goals for the optimization process, such as make the smallest circuit possible, or try to make specified outputs arrive by a specified time**



Save the Design

`write`

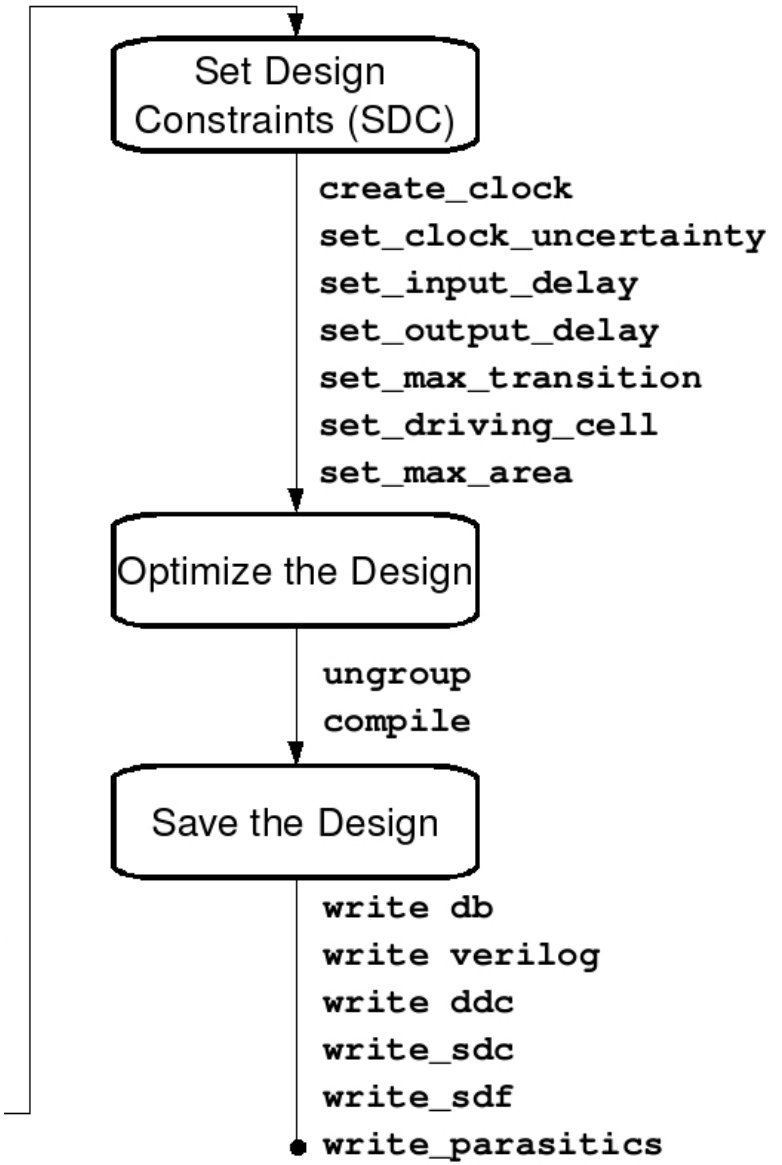
- **Writes a design netlist or schematic from dc_shell to a file (like a save)**
- **Available output format are:**
  - ***db*: Synopsys internal database format (no longer used)**
  - ***ddc*: dc_shell XG mode internal database format (XG mode is an enhanced memory management mode; it provides higher capacity and faster script runtime)**
  - ***verilog* : IEEE Standard Verilog**



Save the Design

`write_sdc`

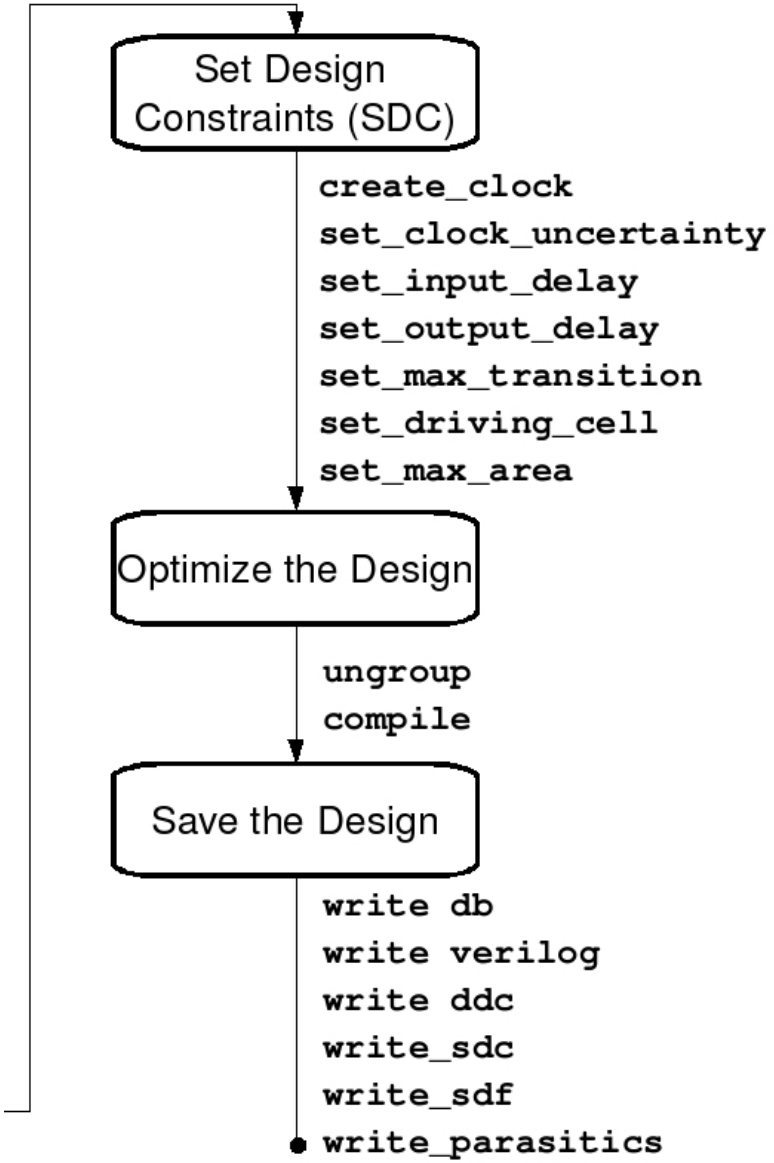
- **Writes out a script in Synopsys Design Constraints (SDC) format**
- **This script contains commands that can be used with PrimeTime to automatically load the design constraints**



Save the Design

`write_sdf`

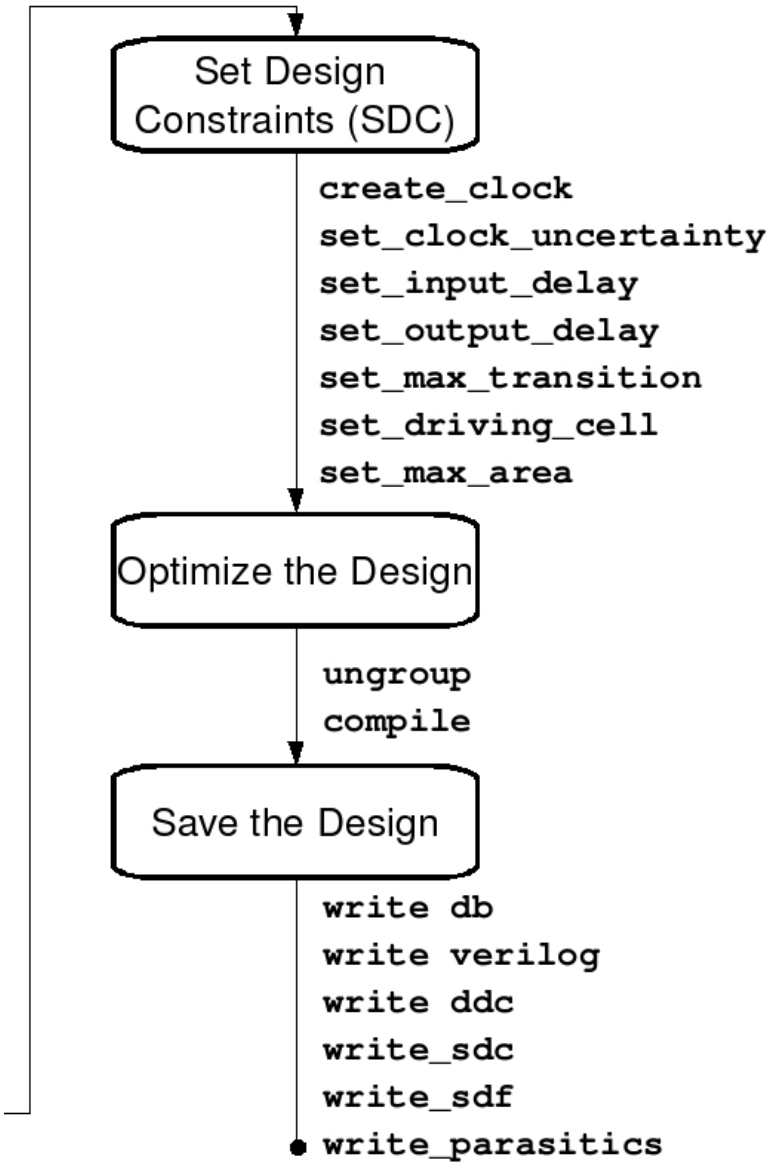
- Writes a Standard Delay Format (SDF) back-annotation file
- Timing information of each net is annotated in the output file
- Timing information is written in SDF format using version v1.0 or v2.1
- For more information please google "SDF format"



Save the Design

`write_parasitics`

- **Writes parasitics in SPEF format to a disk file for the delay calculation tools**
- **The output file can be used with PrimeTime to automatically annotate parasitics**
- **For more information please google "SPEF format"**



Demo