Topic 7.2

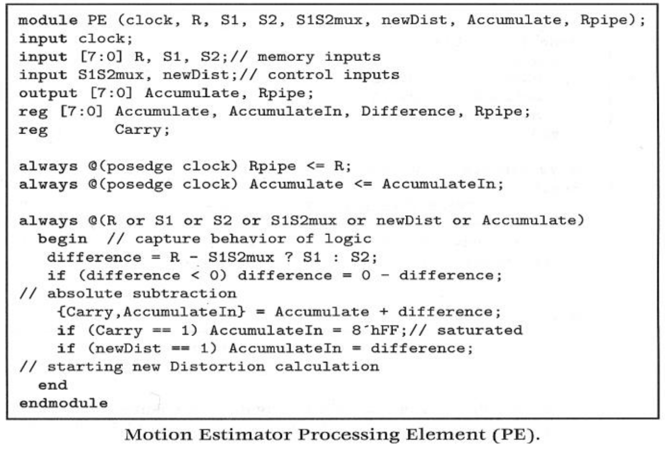
ASIC Design Flow II

Xinfei Guo
xinfei.guo@sjtu.edu.cn

November 4th, 2021

T7 learning goals

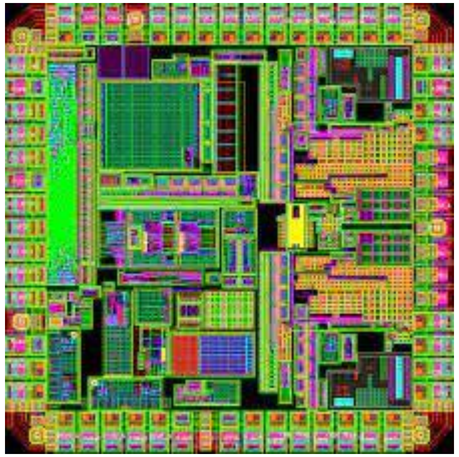
- How to design a Chip (SoC) from concept to silicon?
  - **Full design flow from RTL to Layout**
  - **How to make decisions at each step**



```
module PE (clock, R, S1, S2, S1S2mux, newDist, Accumulate, Rpipe);
input clock;
input [7:0] R, S1, S2;// memory inputs
input S1S2mux, newDist;// control inputs
output [7:0] Accumulate, Rpipe;
reg [7:0] Accumulate, AccumulateIn, Difference, Rpipe;
reg       Carry;

always @(posedge clock) Rpipe <= R;
always @(posedge clock) Accumulate <= AccumulateIn;

always @(R or S1 or S2 or S1S2mux or newDist or Accumulate)
  begin  // capture behavior of logic
   difference = R - S1S2mux ? S1 : S2;
   if (difference < 0) difference = 0 - difference;
// absolute subtraction
    {Carry,AccumulateIn} = Accumulate + difference;
    if (Carry == 1) AccumulateIn = 8´hFF;// saturated
    if (newDist == 1) AccumulateIn = difference;
// starting new Distortion calculation
  end
endmodule
```

Motion Estimator Processing Element (PE).

RTL

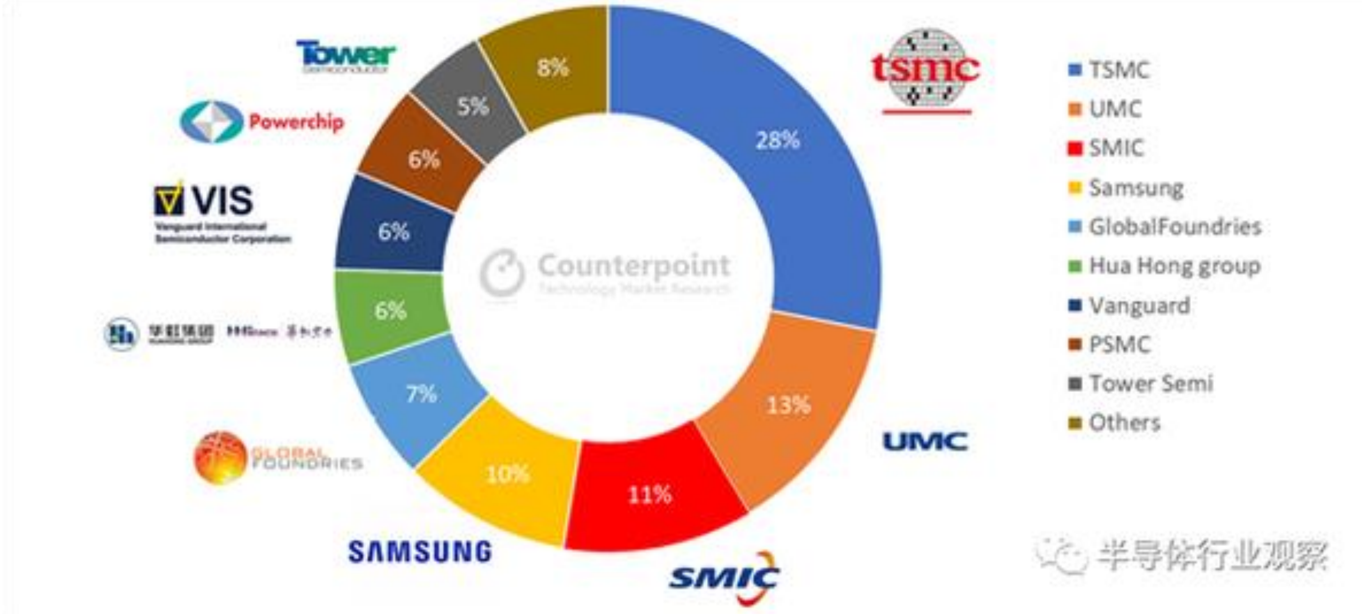


Layout

Figure: Synopsys

Fabless semiconductor company model

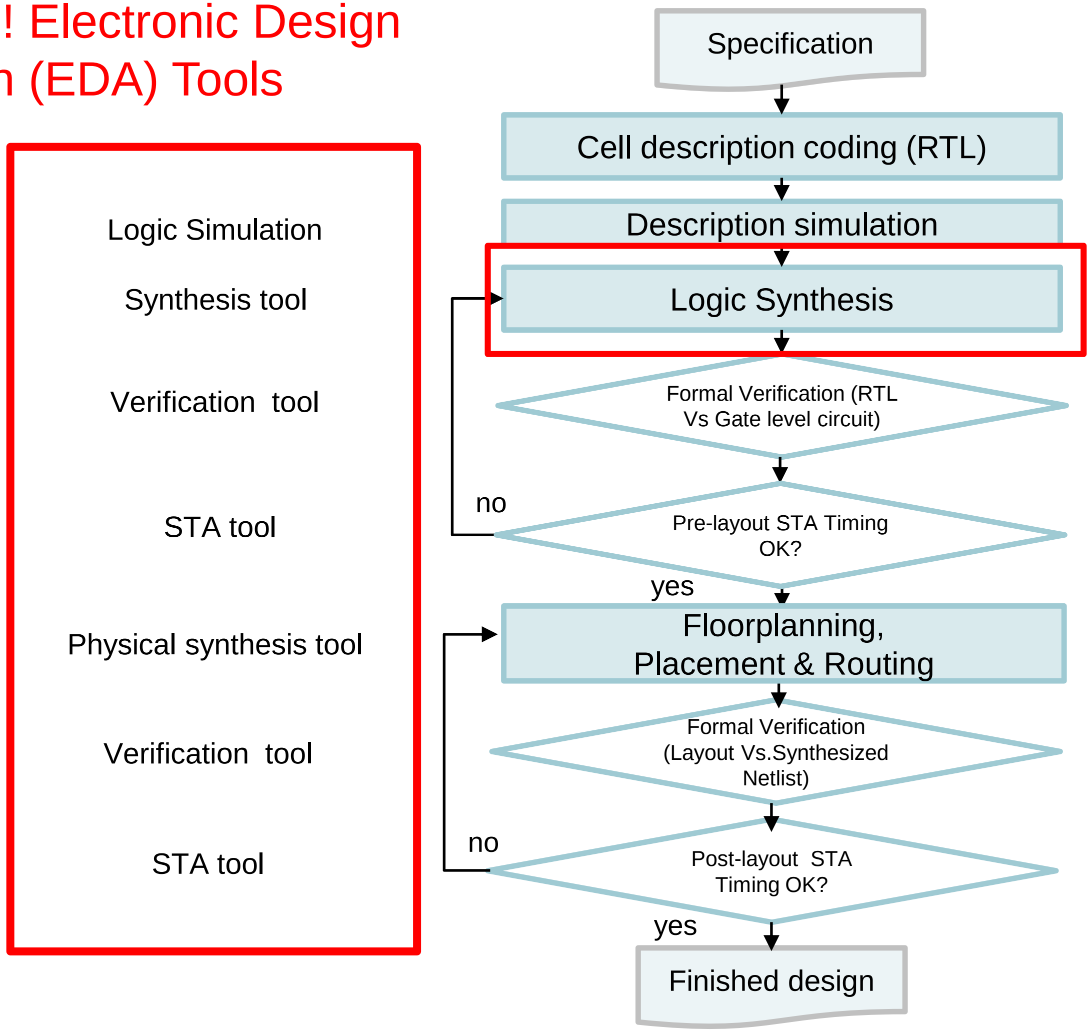
- Company does design only. Fab performed by another company (e.g. TSMC, UMC, Global Foundry, STMicron, SMIC).
- Back-end (place and route, etc.) might be performed at that company or with their assistance



List of Foundries

Digital IC Design Flow

Need tools! Electronic Design Automation (EDA) Tools


Logic Simulation
Synthesis tool
Verification tool
STA tool
Physical synthesis tool
Verification tool
STA tool
Specification
Cell description coding (RTL)
Description simulation
Logic Synthesis
Formal Verification (RTL Vs Gate level circuit)
no
Pre-layout STA Timing OK?
yes
Floorplanning, Placement & Routing
Formal Verification (Layout Vs.Synthesized Netlist)
no
Post-layout STA Timing OK?
yes
Finished design

source: Synopsys

Formal Verification

Are they equal?

Verification in the Design Flow

Verification is very important!!!

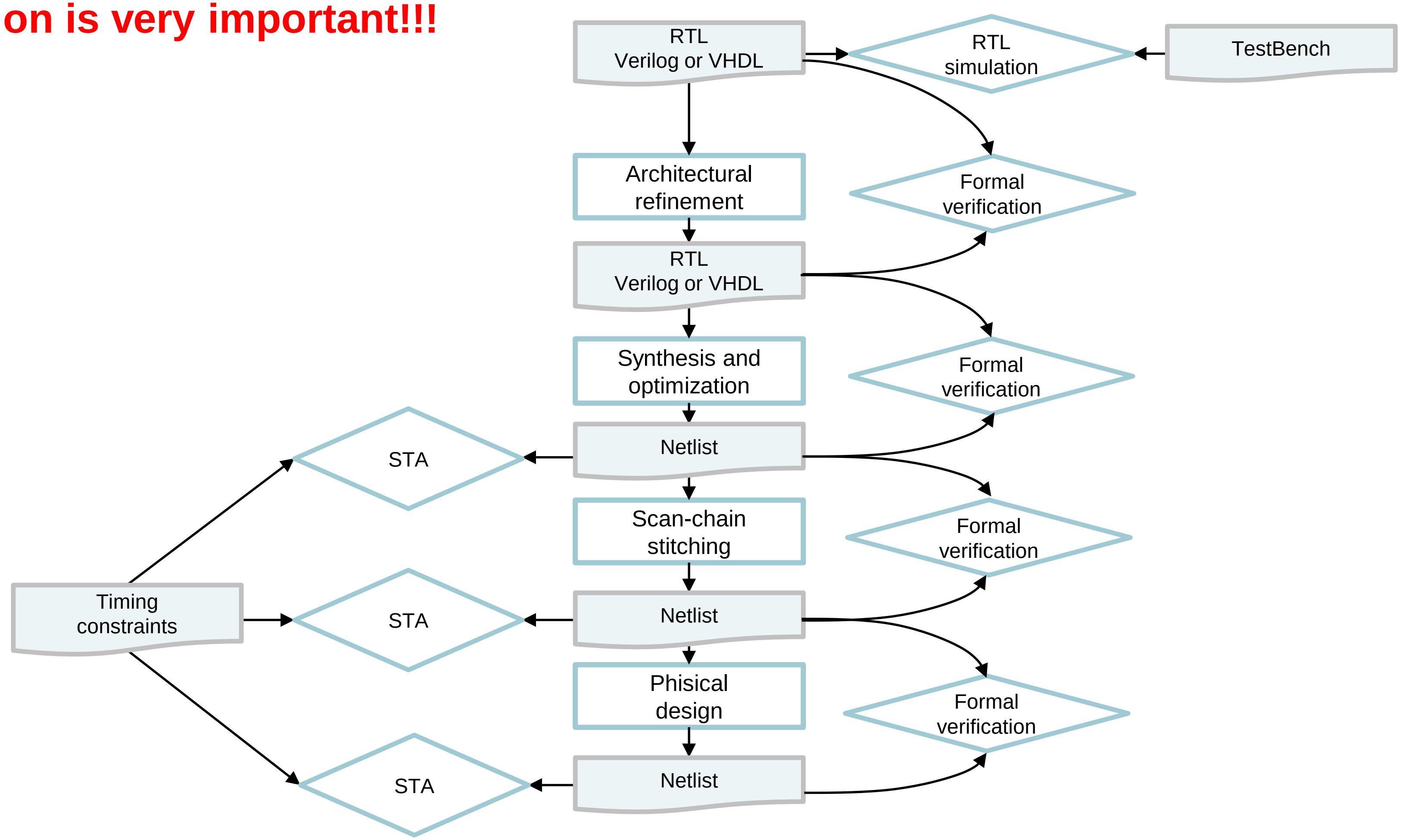


Figure: Synopsys

Traditional IC Simulation

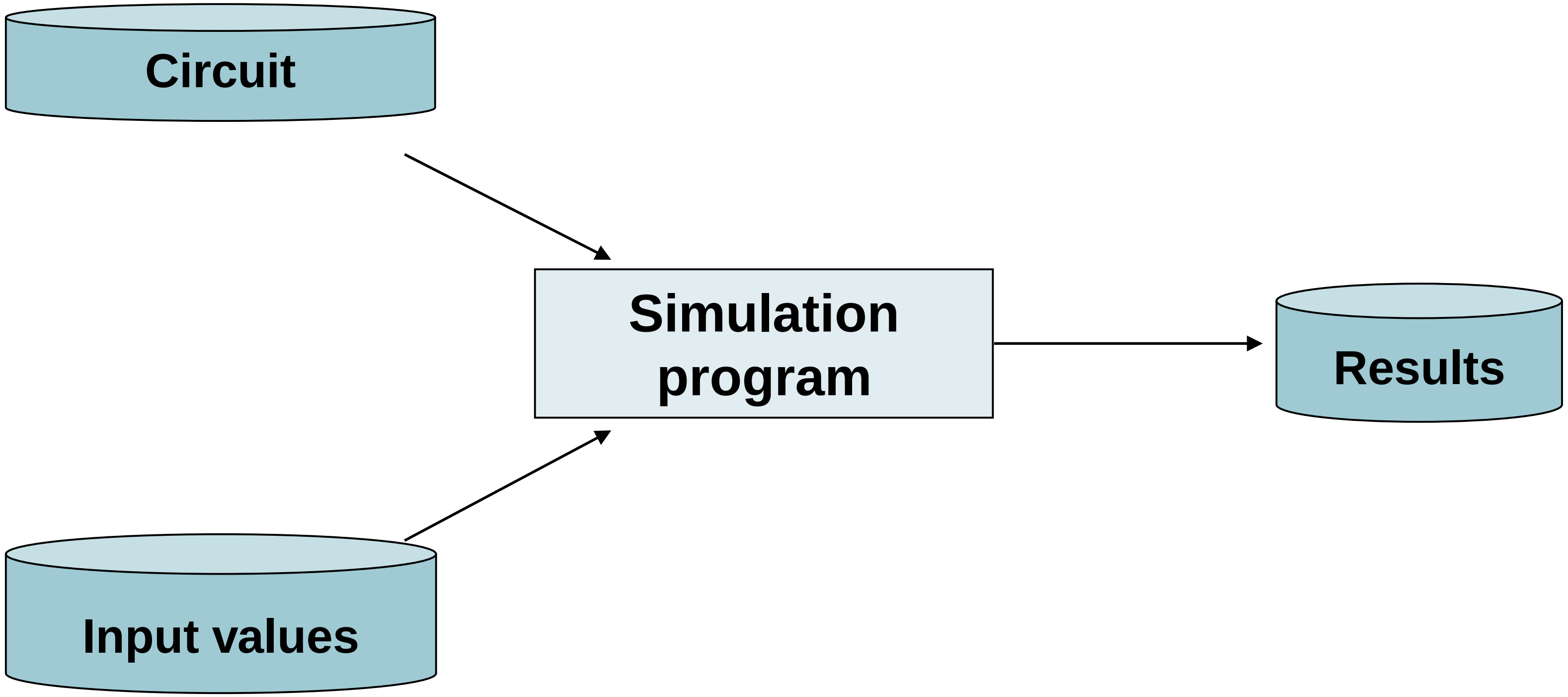


Figure: Synopsys

Formal Verification

- Formal verification checks whether two designs are functionally equivalent or not
- Its purpose is to detect unexpected differences that may have been introduced into a design during development

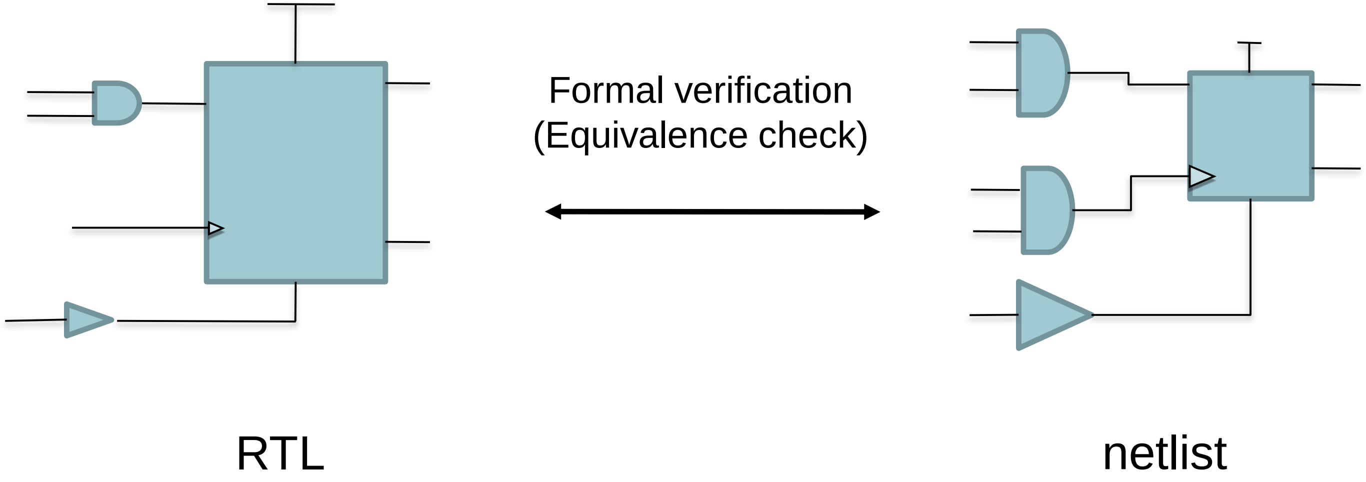
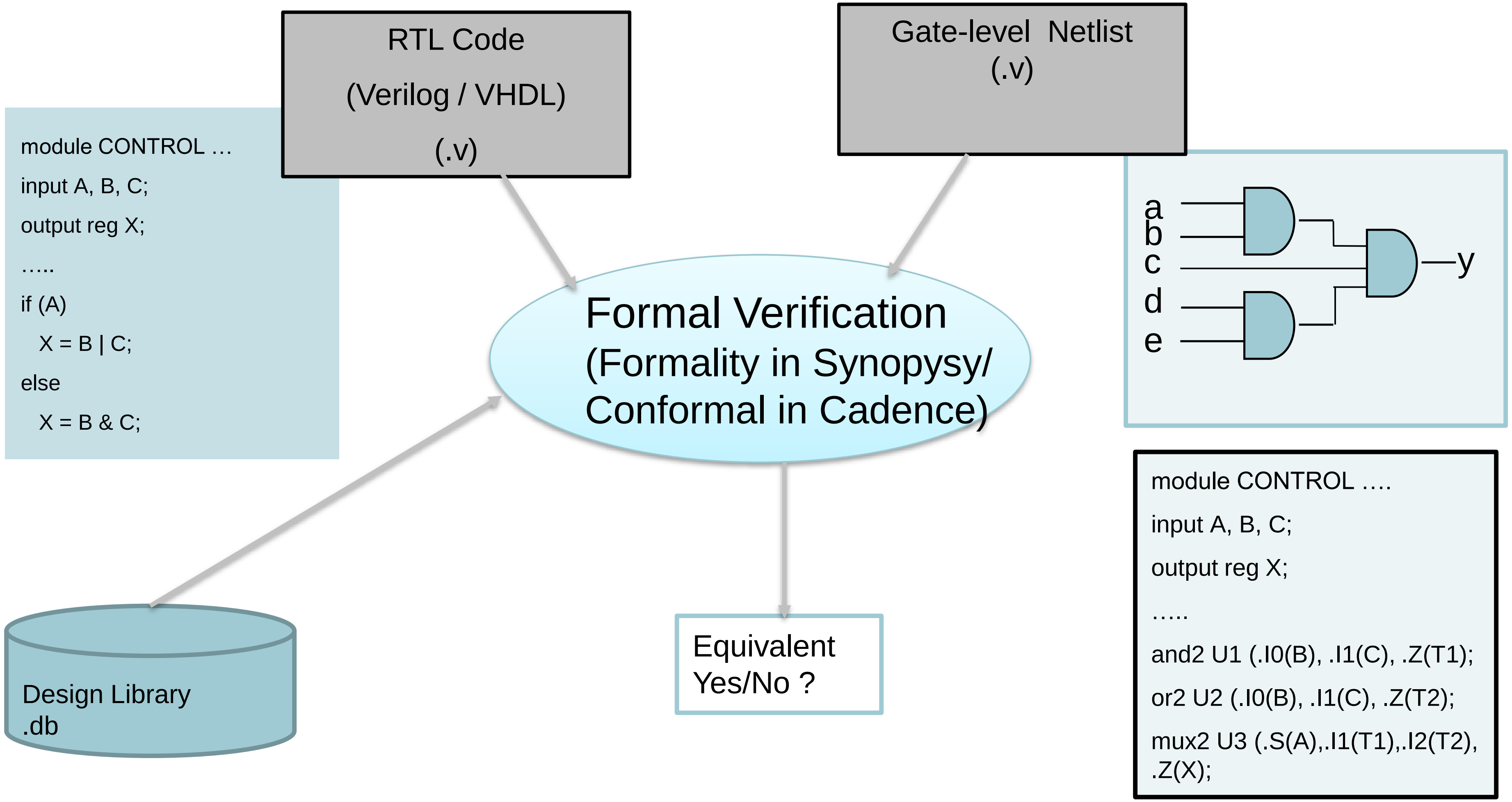


Figure: Synopsys

Formal Verification



RTL Code

(Verilog / VHDL)

(.v)

```
module CONTROL …
input A, B, C;
output reg X;
…..
if (A)
   X = B | C;
else
   X = B & C;
```

Gate-level Netlist
(.v)

```
module CONTROL ….
input A, B, C;
output reg X;
…..
and2 U1 (.I0(B), .I1(C), .Z(T1);
or2 U2 (.I0(B), .I1(C), .Z(T2);
mux2 U3 (.S(A),.I1(T1),.I2(T2),
.Z(X);
```

Formal Verification
(Formality in Synopysy/
Conformal in Cadence)

Design Library
.db

Equivalent
Yes/No ?

Figure: Synopsys

Key Concepts

- Main concepts in Formality are
  - Compare Point
    - Primary output of a circuit
    - Registers within a circuit
    - Input to black boxes within a circuit
  - Logic Cone
    - A block of combinational logic which drives a compare point

Equivalent Checking Verification Process

Equivalence checking is a four-phase process

- Reading and elaborating language descriptions into logical representations
- Setting up prompt for verification
- Mapping of corresponding compare points between pair of designs (matching)
- Comparison of logic cones that drive the compare points (verification)

Example: Input Files of Formality (Synopsys)

Formality supports following input formats:

Input formats	Options
Verilog (synthesizable subset)	- read_verilog
Verilog (simulation libraries)	- read_verilog -vcs
VHDL (synthesizable subset)	- read_vhdl
EDIF	- read_edif
Synopsys binary files	- read_db, read_ddc, read_mdb (*)

Source: Synopsys

Formality Flow Overview

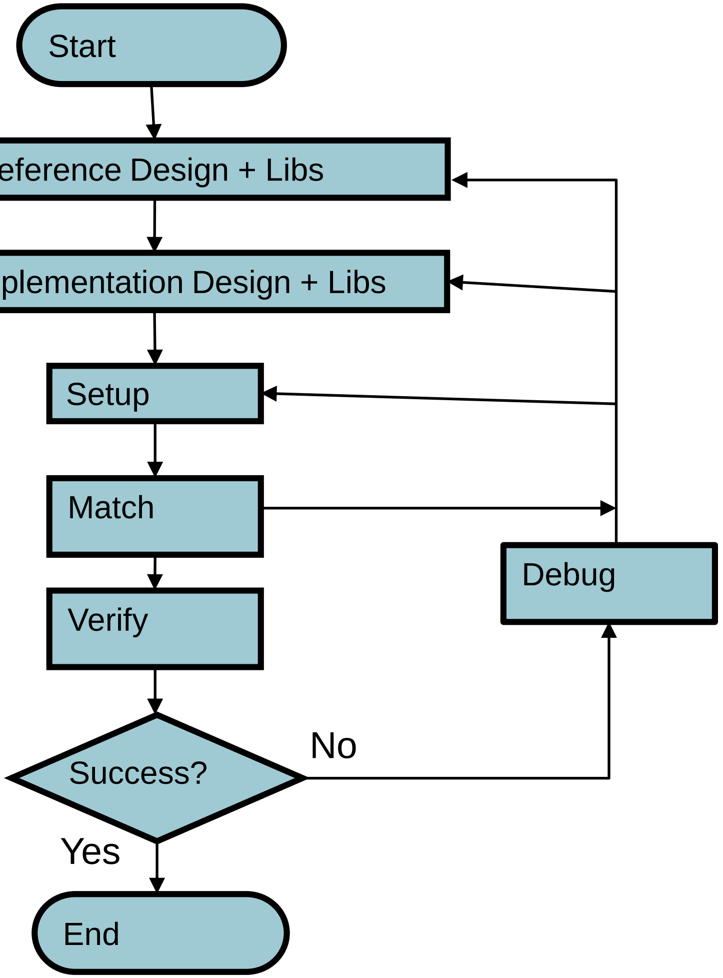
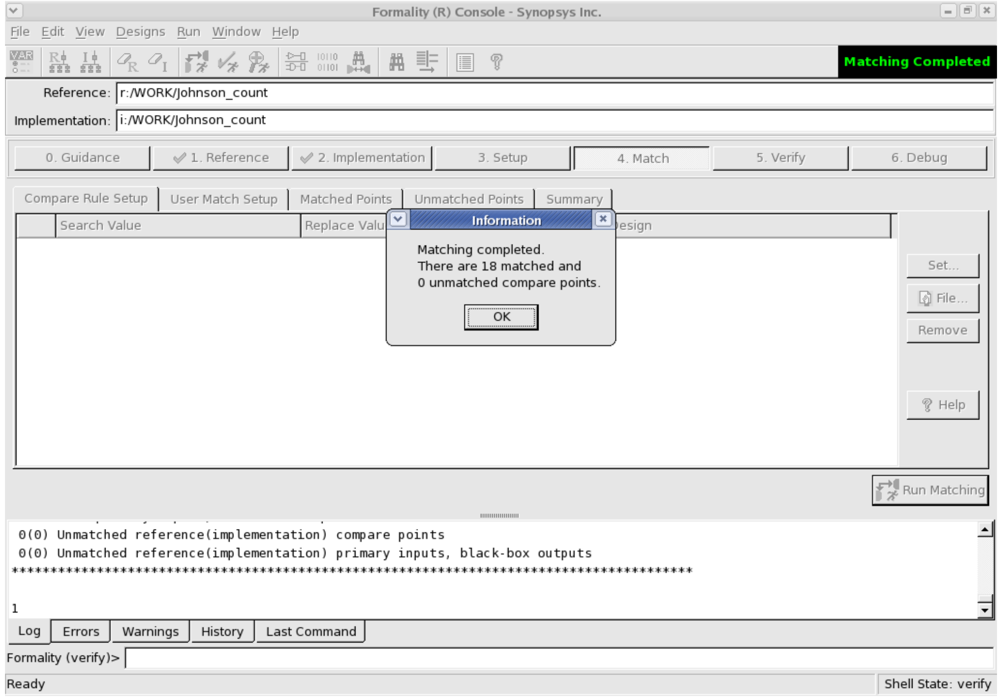

Start
Read Reference Design + Libs
Read Implementation Design + Libs
Setup
Match
Verify
Success?
No
Yes
End
Debug

Figure: Synopsys

Match (Matching Compare Points)

- Process of aligning compare points between two designs



Reference: r:/WORK/Johnson_count

Implementation: i:/WORK/Johnson_count

Matching completed.
There are 18 matched and
0 unmatched compare points.

```
 0(0) Unmatched reference(implementation) compare points
 0(0) Unmatched reference(implementation) primary inputs, black-box outputs
******************************************************************************************

1
```

Formality (verify)>

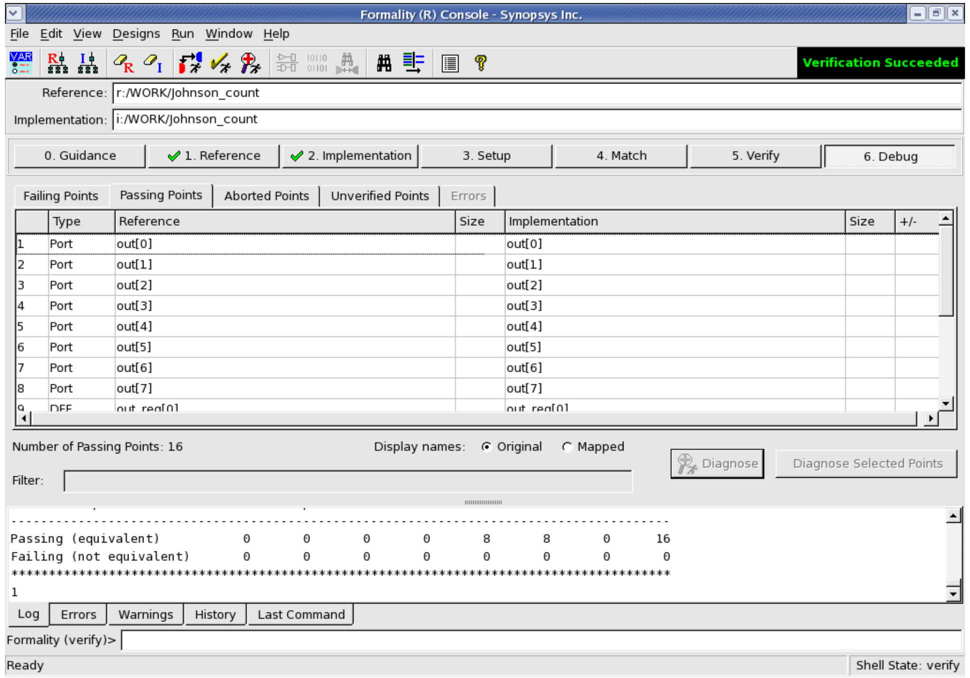
Ready

Shell State: verify

Figure: Synopsys

Debug

- Debugging is part of the process where verification results are used to pinpoint either failing or inconclusive results. During the debug step the user may determine where and possibly why the results were unsuccessful



Reference: r:/WORK/Johnson_count

Implementation: i:/WORK/Johnson_count

	Type	Reference	Size	Implementation	Size	+/-
1	Port	out[0]		out[0]		
2	Port	out[1]		out[1]		
3	Port	out[2]		out[2]		
4	Port	out[3]		out[3]		
5	Port	out[4]		out[4]		
6	Port	out[5]		out[5]		
7	Port	out[6]		out[6]		
8	Port	out[7]		out[7]		
9	DFF	out_reg[0]		out_reg[0]		

Number of Passing Points: 16

Display names: ● Original ○ Mapped

Filter:

```
---------------------------------------------------------------------------------
Passing (equivalent)        0       0       0       0       8       8       0      16
Failing (not equivalent)    0       0       0       0       0       0       0       0
*********************************************************************************
1
```

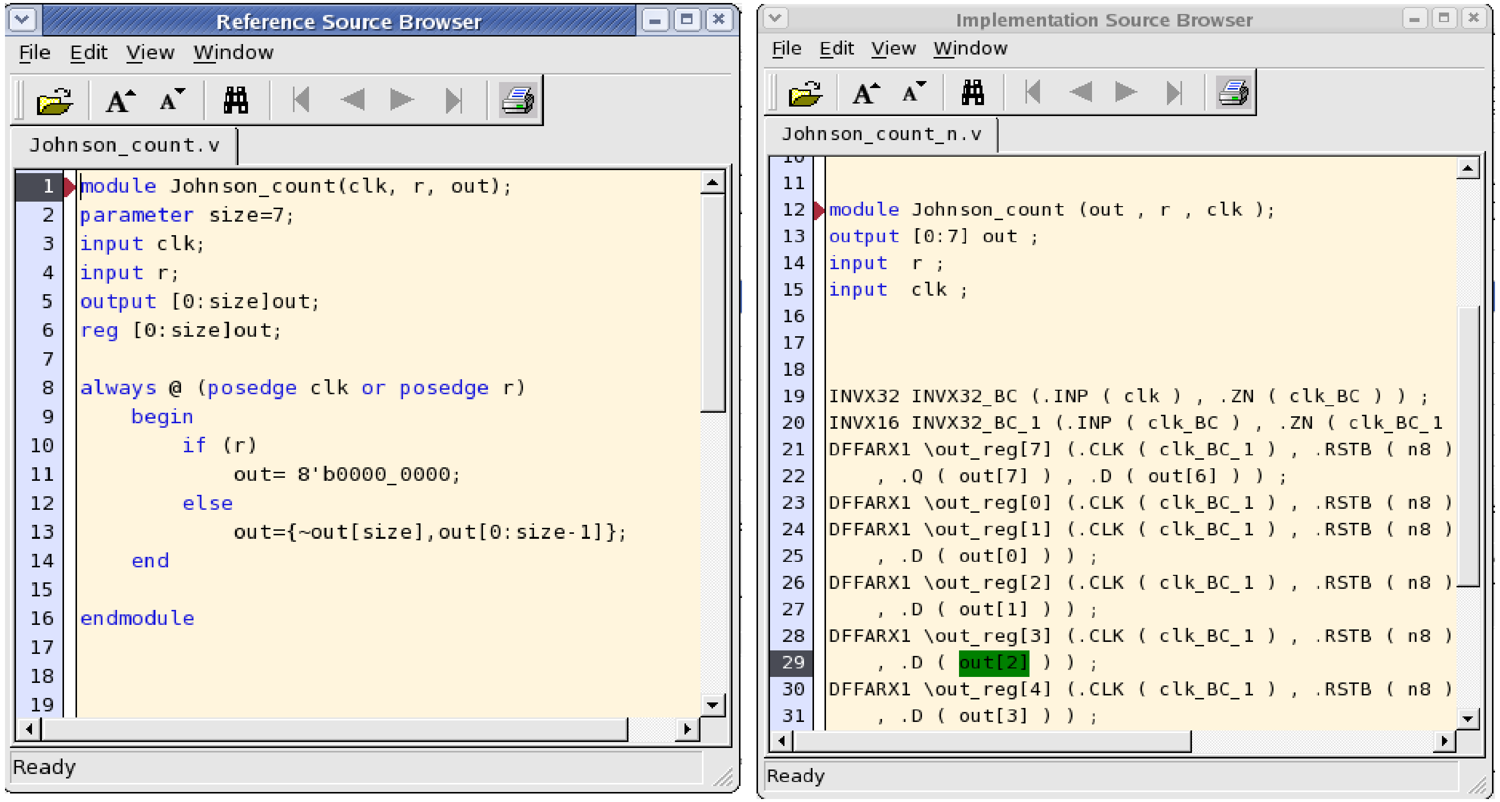
Formality (verify)>

Ready

Shell State: verify

Figure: Synopsys

Debug (2)



```
module Johnson_count(clk, r, out);
parameter size=7;
input clk;
input r;
output [0:size]out;
reg [0:size]out;

always @ (posedge clk or posedge r)
    begin
        if (r)
            out= 8'b0000_0000;
        else
            out={~out[size],out[0:size-1]};
    end

endmodule
```

```

module Johnson_count (out , r , clk );
output [0:7] out ;
input  r ;
input  clk ;



INVX32 INVX32_BC (.INP ( clk ) , .ZN ( clk_BC ) ) ;
INVX16 INVX32_BC_1 (.INP ( clk_BC ) , .ZN ( clk_BC_1
DFFARX1 \out_reg[7] (.CLK ( clk_BC_1 ) , .RSTB ( n8 )
    , .Q ( out[7] ) , .D ( out[6] ) ) ;
DFFARX1 \out_reg[0] (.CLK ( clk_BC_1 ) , .RSTB ( n8 )
DFFARX1 \out_reg[1] (.CLK ( clk_BC_1 ) , .RSTB ( n8 )
    , .D ( out[0] ) ) ;
DFFARX1 \out_reg[2] (.CLK ( clk_BC_1 ) , .RSTB ( n8 )
    , .D ( out[1] ) ) ;
DFFARX1 \out_reg[3] (.CLK ( clk_BC_1 ) , .RSTB ( n8 )
    , .D ( out[2] ) ) ;
DFFARX1 \out_reg[4] (.CLK ( clk_BC_1 ) , .RSTB ( n8 )
    , .D ( out[3] ) ) ;
```

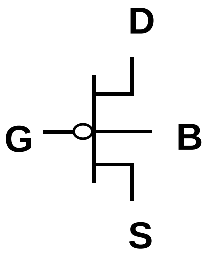
Figure: Synopsys

Floorplanning

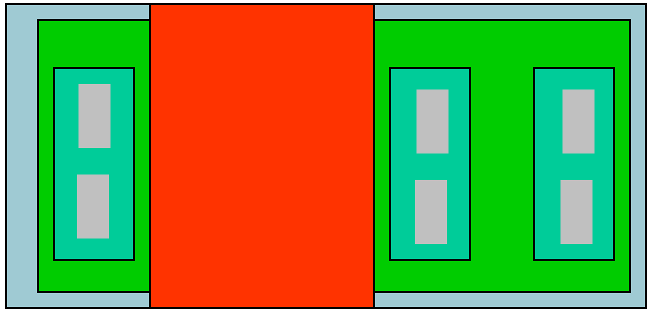
How should we place macros? Where should we

Concepts of the Circuit and Layout

Circuit



Layout



Resulting structure in manufactured IC

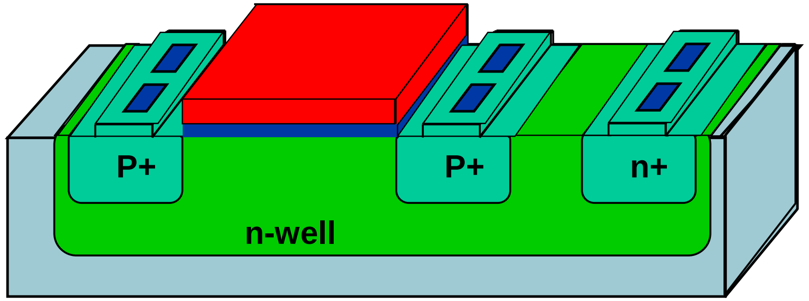
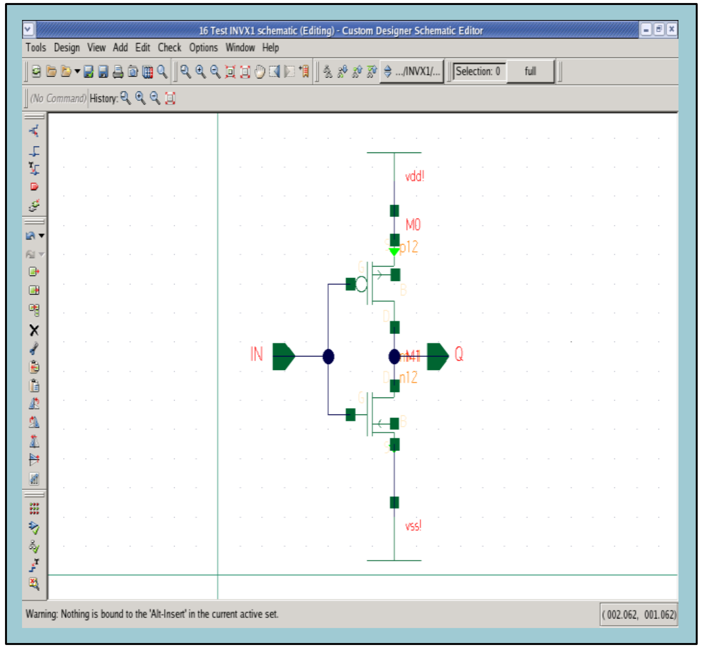


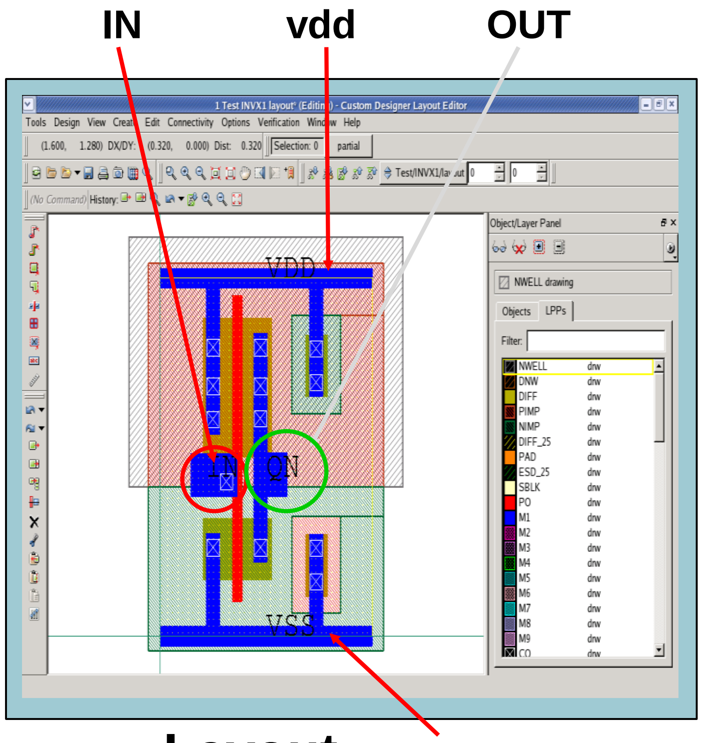
Figure: Synopsys

Circuit and Layout Editors


16 Test INVX1 schematic (Editing) - Custom Designer Schematic Editor
Tools Design View Add Edit Check Options Window Help
.../INVX1/...
Selection: 0
full
(No Command) History:
vdd!
M0
p12
IN
M1
n12
Q
vss!
(002.062, 001.062)

Warning: Nothing is bound to the 'Alt-Insert' in the current active set.

Circuit


IN
vdd
OUT
1 Test INVX1 layout* (Editing) - Custom Designer Layout Editor
Tools Design View Create Edit Connectivity Options Verification Window Help
(1.600, 1.280) DX/DY: (0.320, 0.000) Dist: 0.320
Selection: 0
partial
Test/INVX1/layout
(No Command) History:
VDD
IN
QN
VSS
Object/Layer Panel
NWELL drawing
Objects
LPPs
Filter:
vss

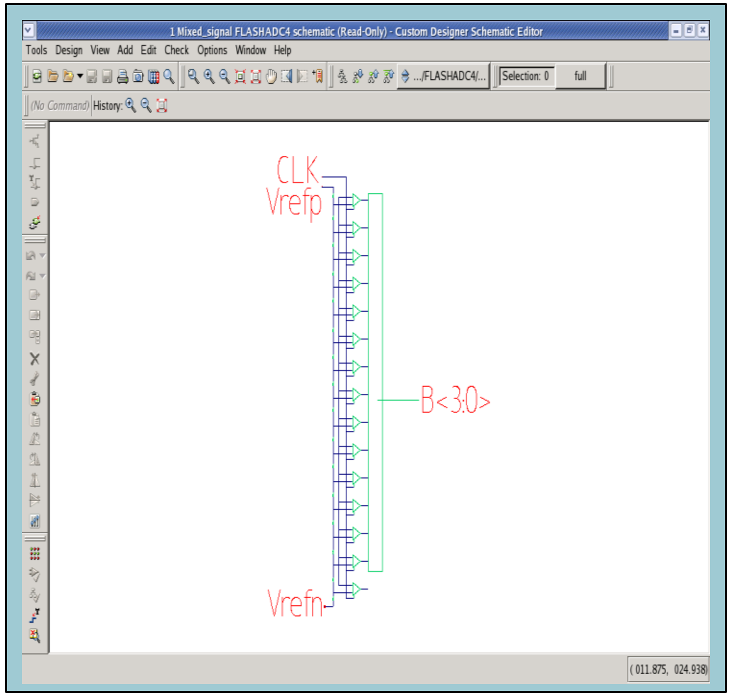
NWELL	drw
DNW	drw
DIFF	drw
PIMP	drw
NIMP	drw
DIFF_25	drw
PAD	drw
ESD_25	drw
SBLK	drw
PO	drw
M1	drw
M2	drw
M3	drw
M4	drw
M5	drw
M6	drw
M7	drw
M8	drw
M9	drw
CO	drw

Layout

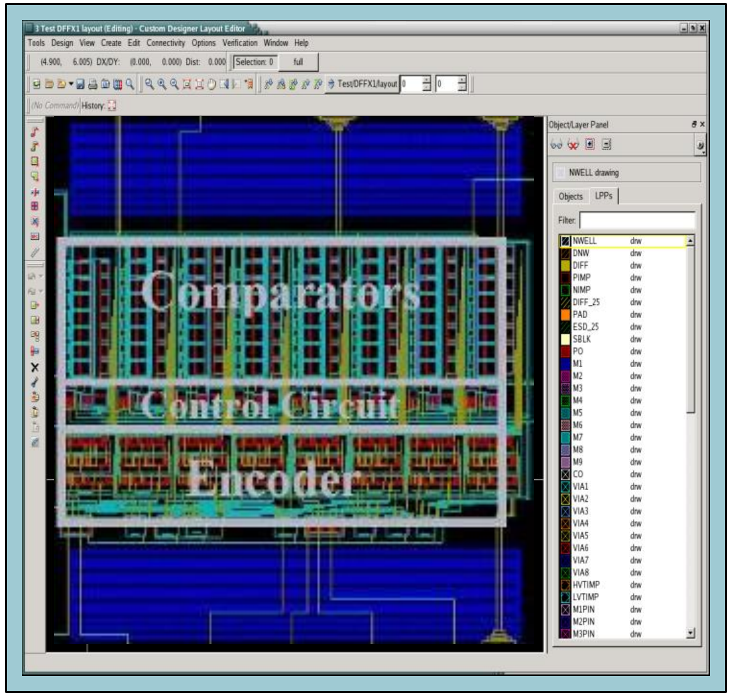
Figure: Synopsys

IP Example

FLASHADC4


1 Mixed_signal FLASHADC4 schematic (Read-Only) - Custom Designer Schematic Editor
Tools Design View Add Edit Check Options Window Help
.../FLASHADC4/...
Selection: 0
full
(No Command) History:
CLK
Vrefp
B<3:0>
Vrefn
(011.875, 024.938)

Circuit


3 Test DFFX1 layout (Editing) - Custom Designer Layout Editor
Tools Design View Create Edit Connectivity Options Verification Window Help
(4.900, 6.005) DX/DY: (0.000, 0.000) Dist: 0.000
Selection: 0
full
Test/DFFX1/layout
(No Command) History:
Comparators
Control Circuit
Encoder
Object/Layer Panel
NWELL drawing
Objects
LPPs
Filter:

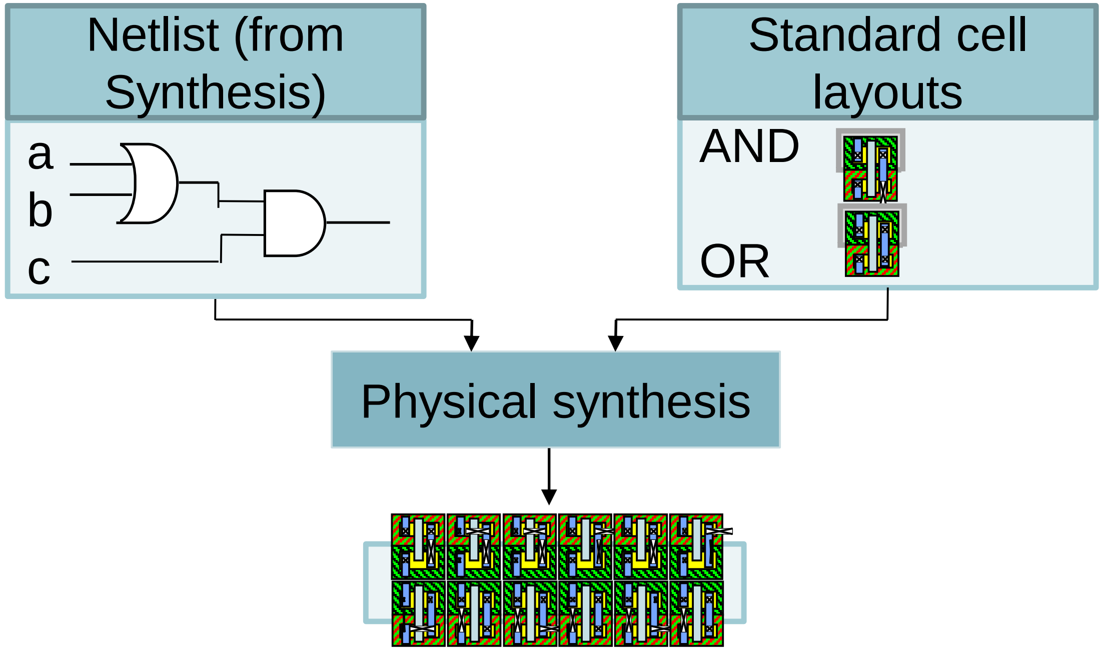
NWELL	drw
DNW	drw
DIFF	drw
PIMP	drw
NIMP	drw
DIFF_25	drw
PAD	drw
ESD_25	drw
SBLK	drw
PO	drw
M1	drw
M2	drw
M3	drw
M4	drw
M5	drw
M6	drw
M7	drw
M8	drw
M9	drw
CO	drw
VIA1	drw
VIA2	drw
VIA3	drw
VIA4	drw
VIA5	drw
VIA6	drw
VIA7	drw
VIA8	drw
HVTIMP	drw
LVTIMP	drw
M1PIN	drw
M2PIN	drw
M3PIN	drw

Layout

Figure: Synopsys

Physical Synthesis

Physical synthesis is the process that produces layout of logic circuit.



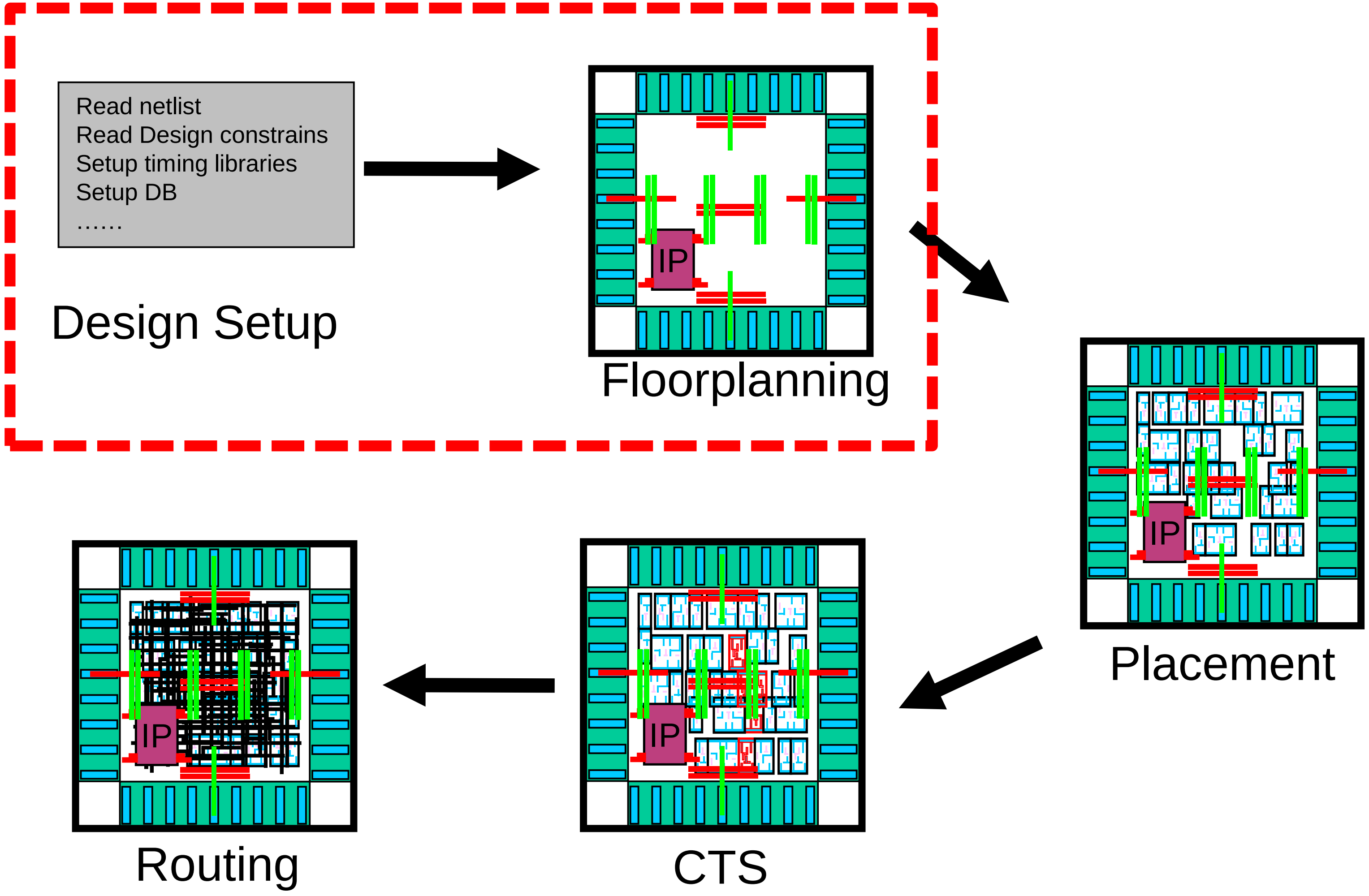
Netlist (from Synthesis)

Standard cell layouts

Physical synthesis

Figure: Synopsys

Physical Synthesis Flow



Read netlist
Read Design constrains
Setup timing libraries
Setup DB
……

Design Setup

Floorplanning

Placement

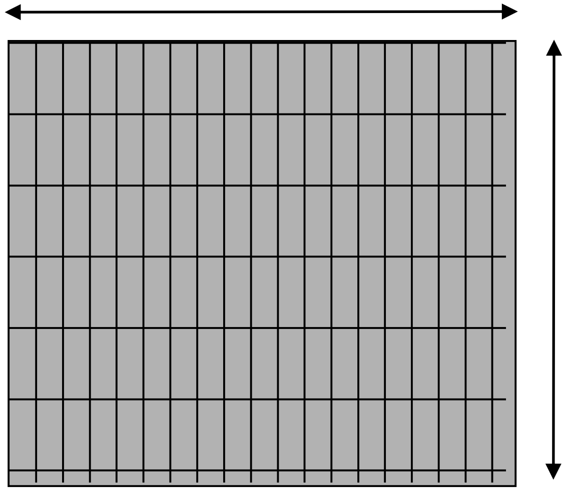
CTS

Routing

Figure: Synopsys

Floorplanning

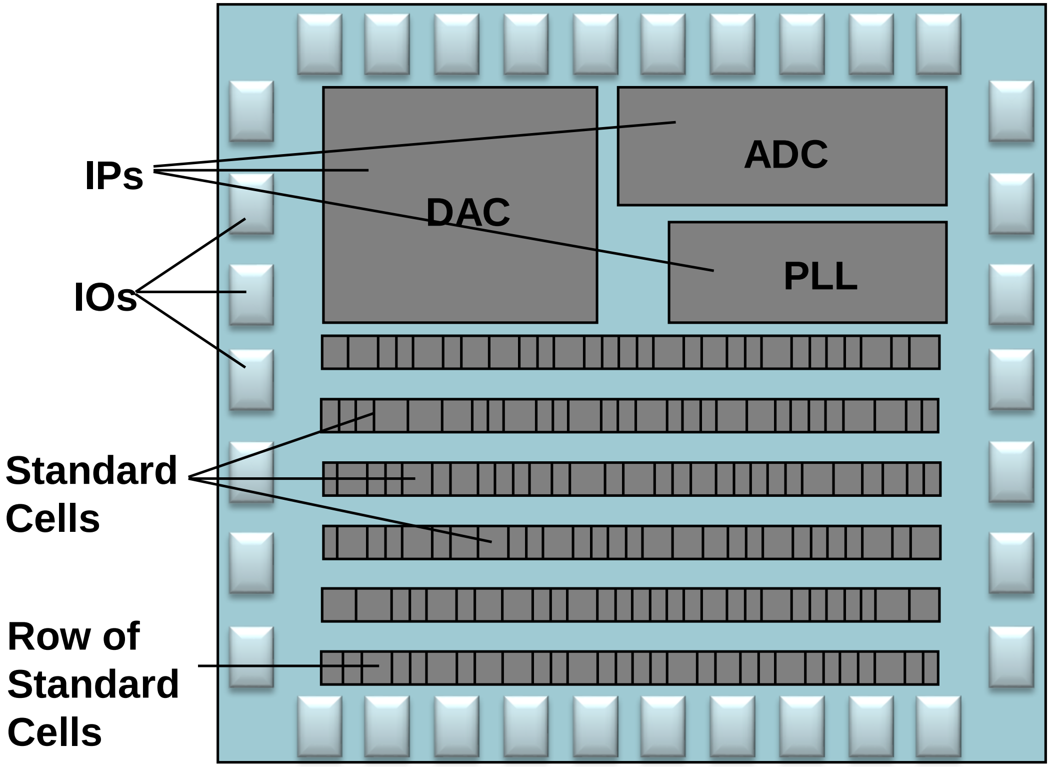
- Not needed in FPGA flow
- During the floorplanning step the overall cell is defined, including: cell size, supply network, etc.



Floorplan

Figure: Synopsys

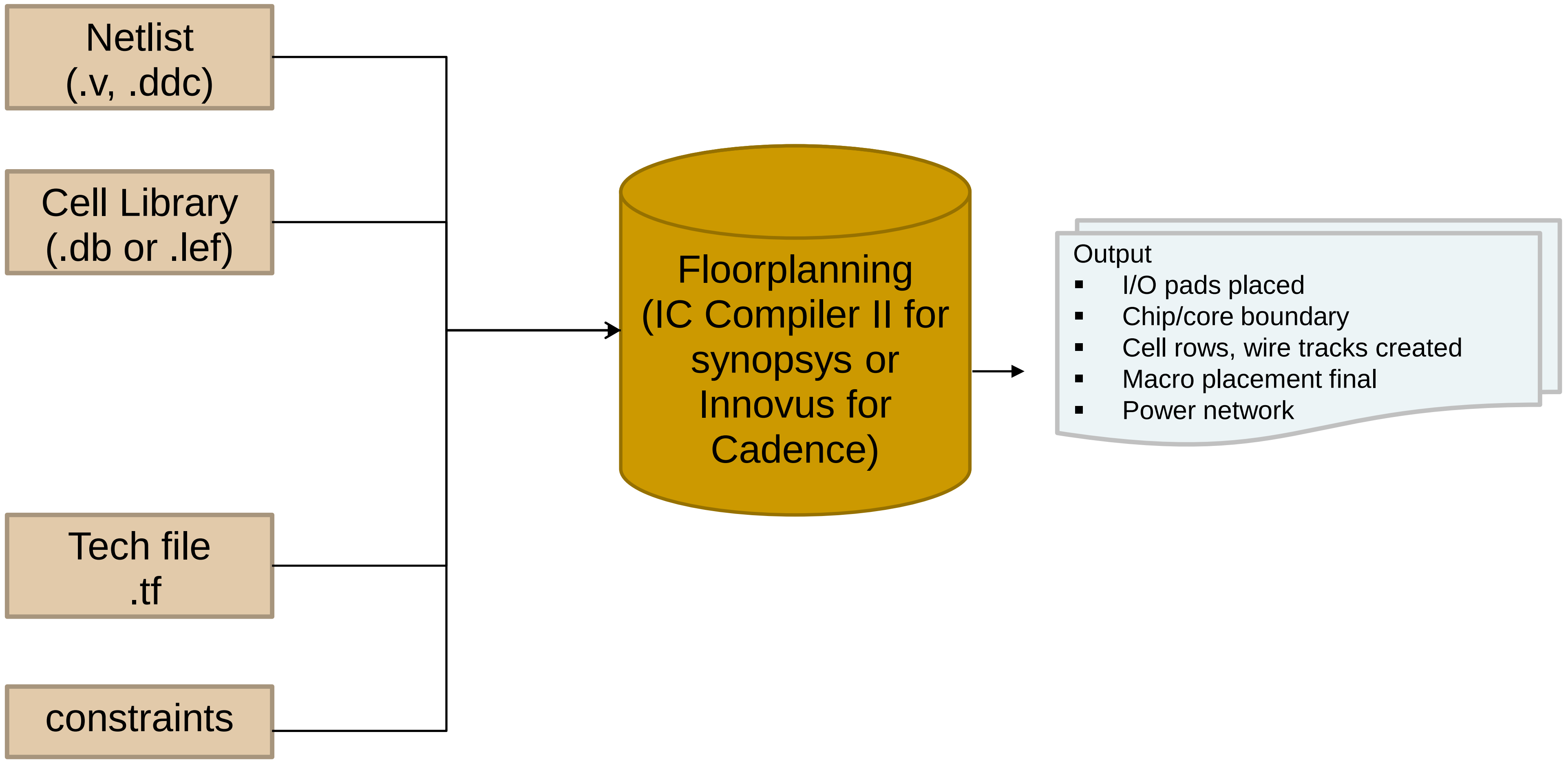
Floorplanning



- What are done in floorplanning?
  - IO Placement
  - Die Size and Aspect Ratio
  - Special Cell Pre-placemnt
  - IP/Macro pre-placement
  - Power grid generation
  - Blockage definition
  - Standard cells are **NOT** placed yet in this step.

Figure: Synopsys

Data Setup for Floorplan



Netlist (.v, .ddc)

Cell Library (.db or .lef)

Tech file .tf

constraints

Floorplanning (IC Compiler II for synopsys or Innovus for Cadence)

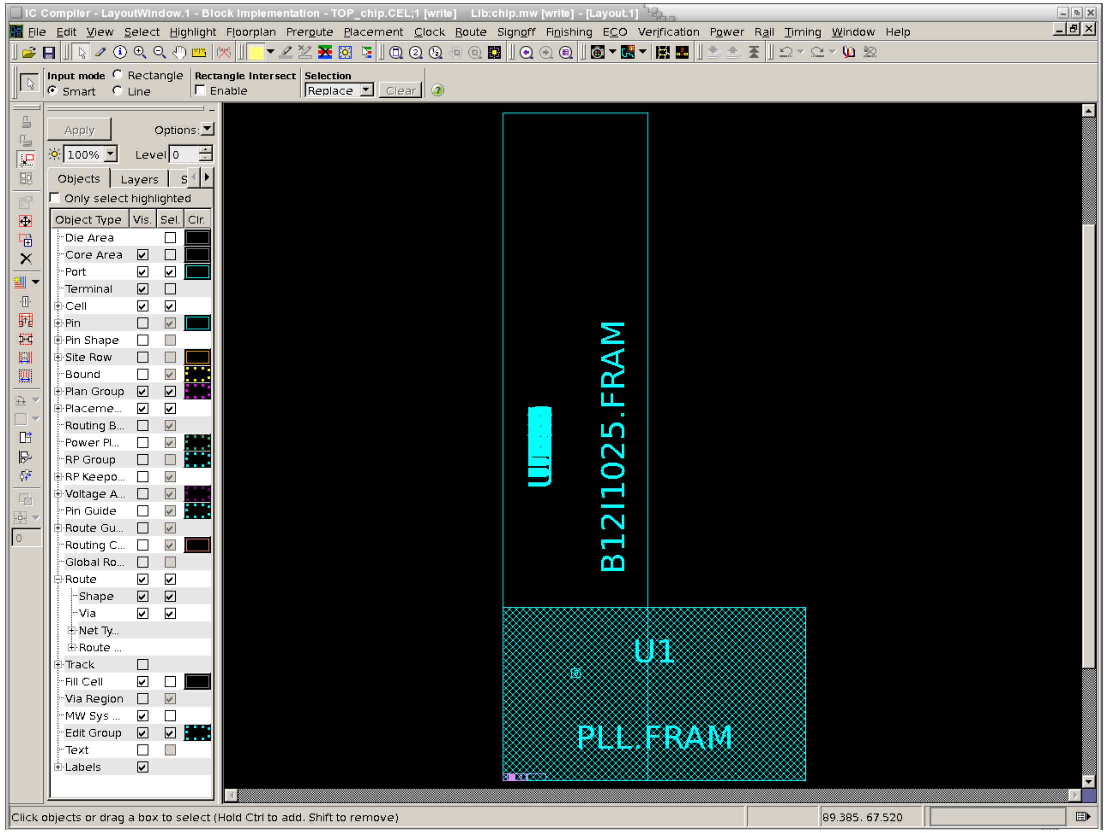
Output

- I/O pads placed
- Chip/core boundary
- Cell rows, wire tracks created
- Macro placement final
- Power network

Figure: Synopsys

Design Importing

- Example: Design view from ICC after importing the design into ICC


IC Compiler - LayoutWindow.1 - Block Implementation - TOP_chip.CEL;1 [write] Lib:chip.mw [write] - [Layout.1]
File Edit View Select Highlight Floorplan Preroute Placement Clock Route Signoff Finishing ECO Verification Power Rail Timing Window Help
Apply
Options:
Objects
Layers
B12I1025.FRAM
U1
PLL.FRAM

Input mode ● Smart ○ Rectangle ○ Line

Rectangle Intersect □ Enable

Selection: Replace

100% Level 0

□ Only select highlighted

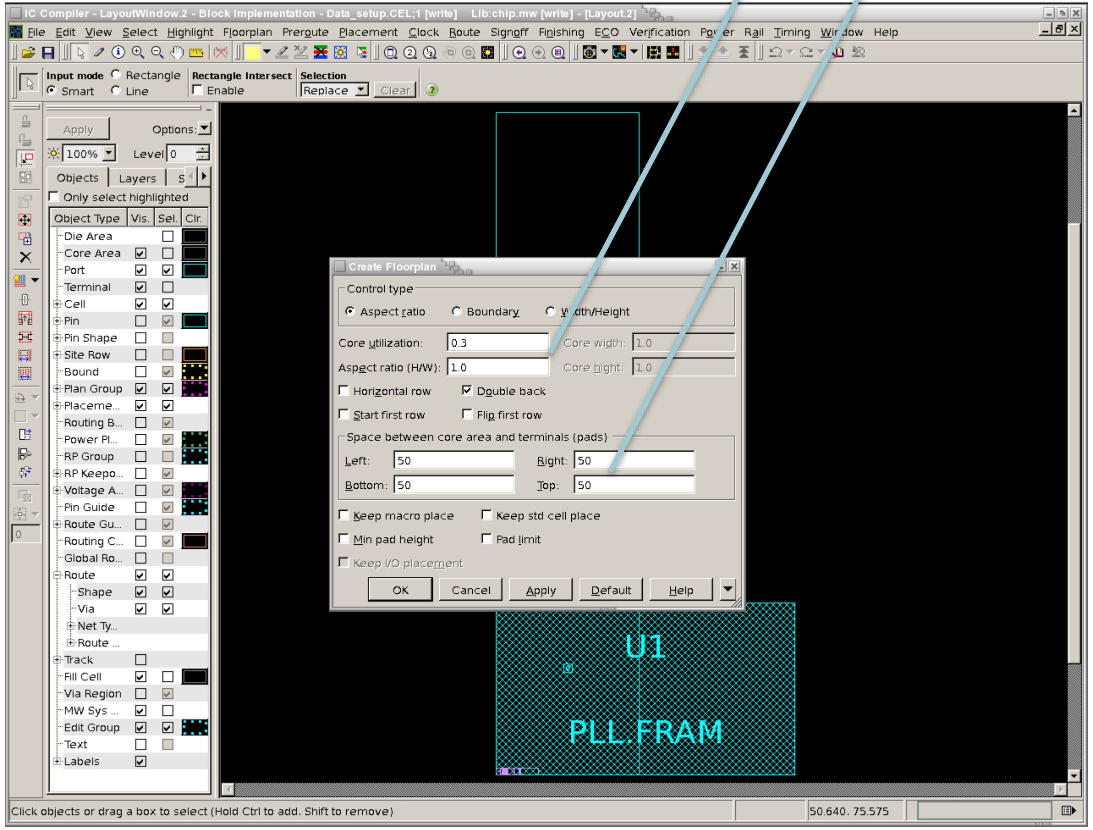
Object Type	Vis.	Sel.	Clr.
Die Area		□	■
Core Area	☑	□	■
Port	☑	☑	■
Terminal	☑	□	
Cell	☑	☑	
Pin	□	☑	■
Pin Shape	□	□	
Site Row	□	□	■
Bound	□	☑	▦
Plan Group	☑	☑	▦
Placeme...	☑	☑	
Routing B...	□	☑	
Power Pl...	□	☑	▦
RP Group	□	□	▦
RP Keepo...	□	☑	
Voltage A...	□	☑	▦
Pin Guide	□	☑	▦
Route Gu...	□	☑	
Routing C...	□	☑	■
Global Ro...	□	□	
Route	☑	☑	
Shape	☑	☑	
Via	☑	☑	
Net Ty...			
Route ...			
Track	□		
Fill Cell	☑	□	■
Via Region	□	☑	
MW Sys ...	☑	□	
Edit Group	☑	☑	▦
Text	□	□	
Labels	☑		

Click objects or drag a box to select (Hold Ctrl to add. Shift to remove)

89.385, 67.520

Figure: Synopsys

Initialize Floorplanning


IC Compiler - LayoutWindow.2 - Block Implementation - Data_setup.CEL;1 [write]   Lib:chip.mw [write] - [Layout.2]
File Edit View Select Highlight Floorplan Preroute Placement Clock Route Signoff Finishing ECO Verification Power Rail Timing Window Help
Input mode ● Smart ○ Rectangle ○ Line
Rectangle Intersect □ Enable
Selection Replace Clear
Apply
Options:
100%
Level 0
Objects Layers S
U1
PLL.FRAM

□ Only select highlighted

Object Type	Vis.	Sel.	Clr.
Die Area		□	■
Core Area	☑	□	■
Port	☑	☑	■
Terminal	☑	□	
Cell	☑	☑	
Pin	□	☑	■
Pin Shape	□	□	
Site Row	□	□	■
Bound	□	☑	■
Plan Group	☑	☑	■
Placeme...	☑	☑	
Routing B...	□	☑	
Power Pl...	□	☑	■
RP Group	□	□	■
RP Keepo...	□	☑	
Voltage A...	□	☑	■
Pin Guide	□	☑	■
Route Gu...	□	☑	
Routing C...	□	☑	■
Global Ro...	□	□	
Route	☑	☑	
Shape	☑	☑	
Via	☑	☑	
Net Ty...			
Route ...			
Track	□		
Fill Cell	☑	□	■
Via Region	□	☑	
MW Sys ...	☑	□	
Edit Group	☑	☑	■
Text	□	□	
Labels	☑		

Create Floorplan

Control type
● Aspect ratio ○ Boundary ○ Width/Height

Core utilization: 0.3   Core width: 1.0
Aspect ratio (H/W): 1.0   Core hight: 1.0

□ Horizontal row   ☑ Double back
□ Start first row   □ Flip first row

Space between core area and terminals (pads)
Left: 50   Right: 50
Bottom: 50   Top: 50

□ Keep macro place   □ Keep std cell place
□ Min pad height   □ Pad limit
□ Keep I/O placement

OK Cancel Apply Default Help

Click objects or drag a box to select (Hold Ctrl to add. Shift to remove)   50.640, 75.575

Set Core utilization

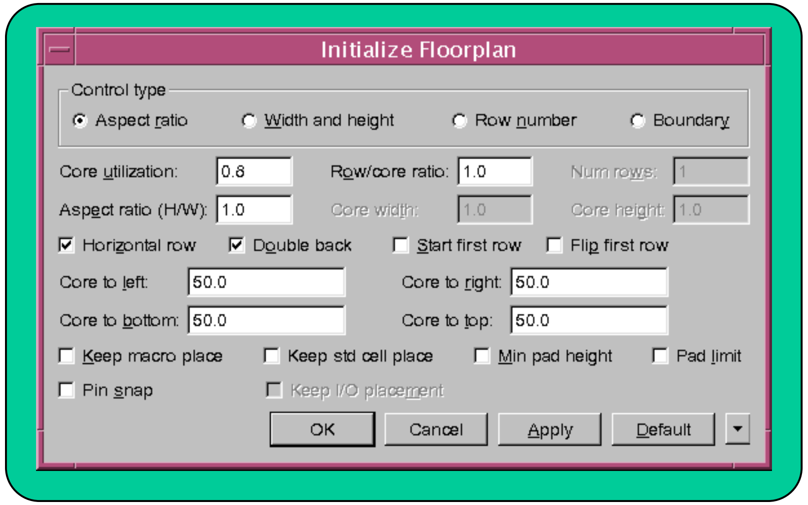
Space between Core area and terminals

Figure: Synopsys

Initialize the Floorplan

- Generates the basic elements of the FP
  - Place IO pads/pins
  - Create chip/core boundary
  - Create rows and tracks honoring user defined values
    - aspect ratio/width & height/row number boundary
    - core utilization
    - …

```
initialize_floorplan
```



Initialize Floorplan

Control type
● Aspect ratio ○ Width and height ○ Row number ○ Boundary

Core utilization: 0.8
Row/core ratio: 1.0
Num rows: 1
Aspect ratio (H/W): 1.0
Core width: 1.0
Core height: 1.0

☑ Horizontal row ☑ Double back □ Start first row □ Flip first row

Core to left: 50.0
Core to right: 50.0
Core to bottom: 50.0
Core to top: 50.0

□ Keep macro place □ Keep std cell place □ Min pad height □ Pad limit
□ Pin snap □ Keep I/O placement

Figure: Synopsys

What is a row?

- Cells are placed in rows, next to each other
- One cells structure continue previous one
- Cells on neighbor rows are flipped so that they can share same supply

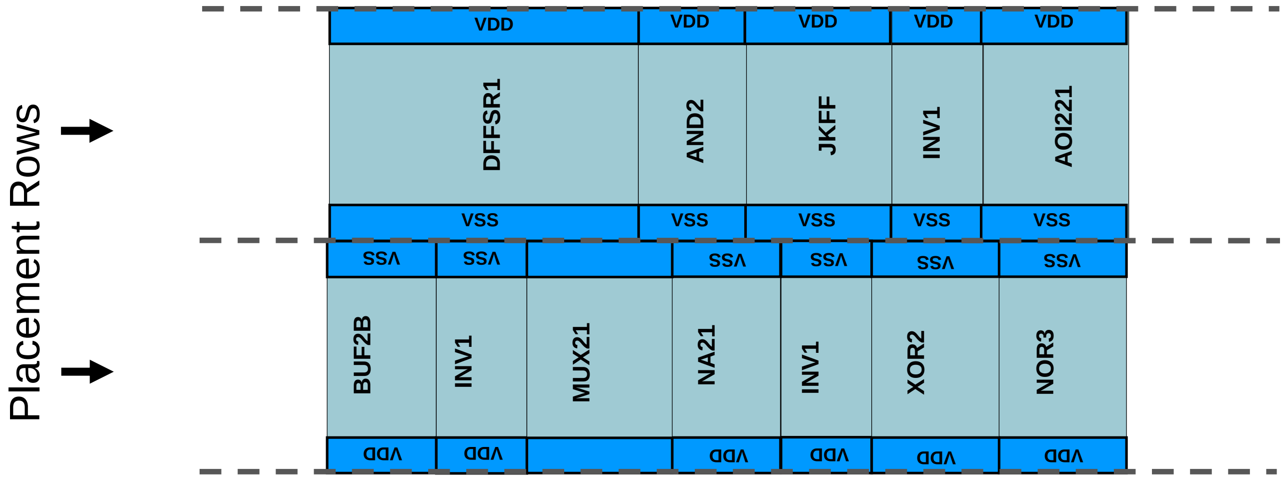
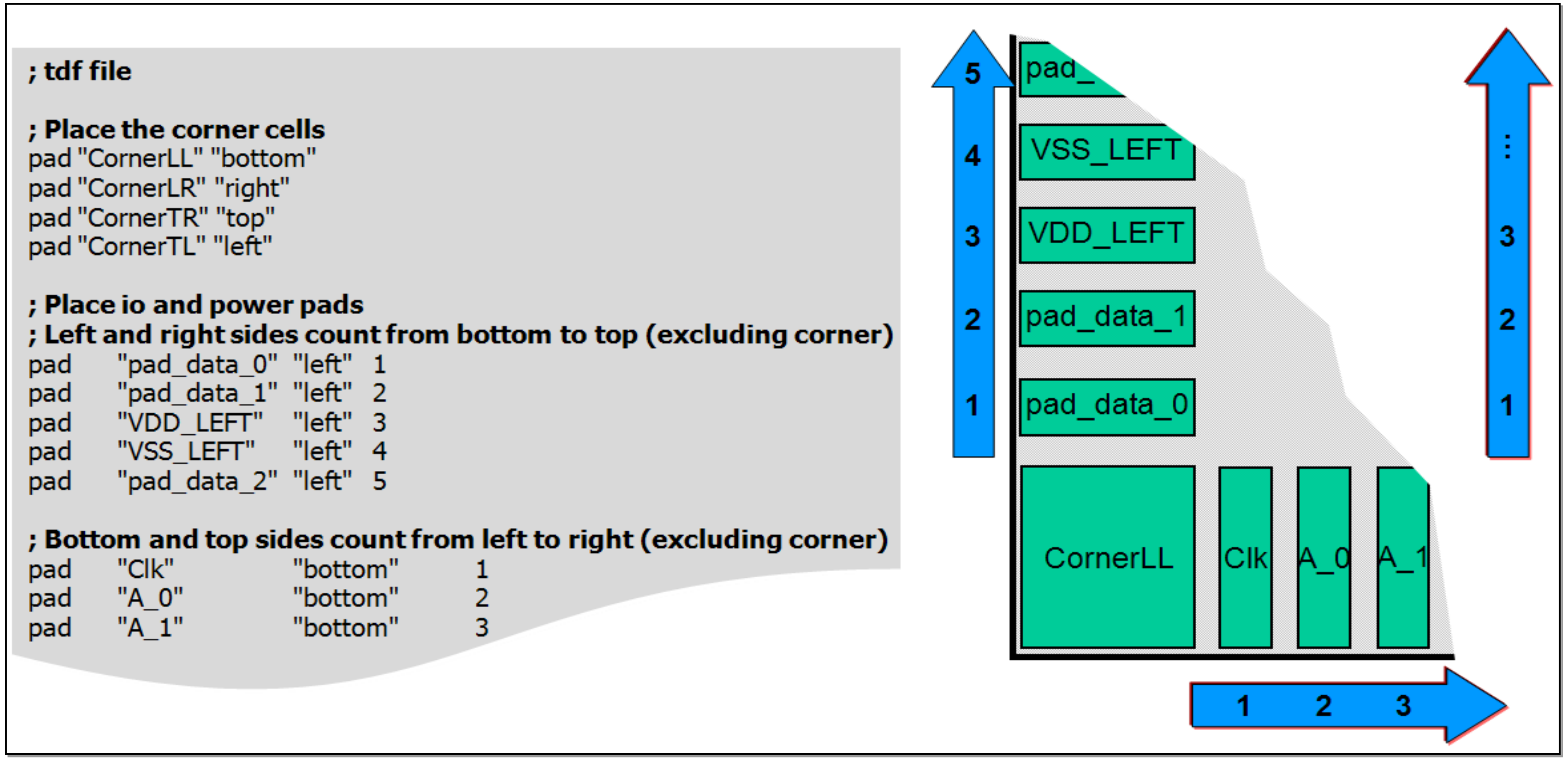


Figure: Synopsys

I/O Order Assignments

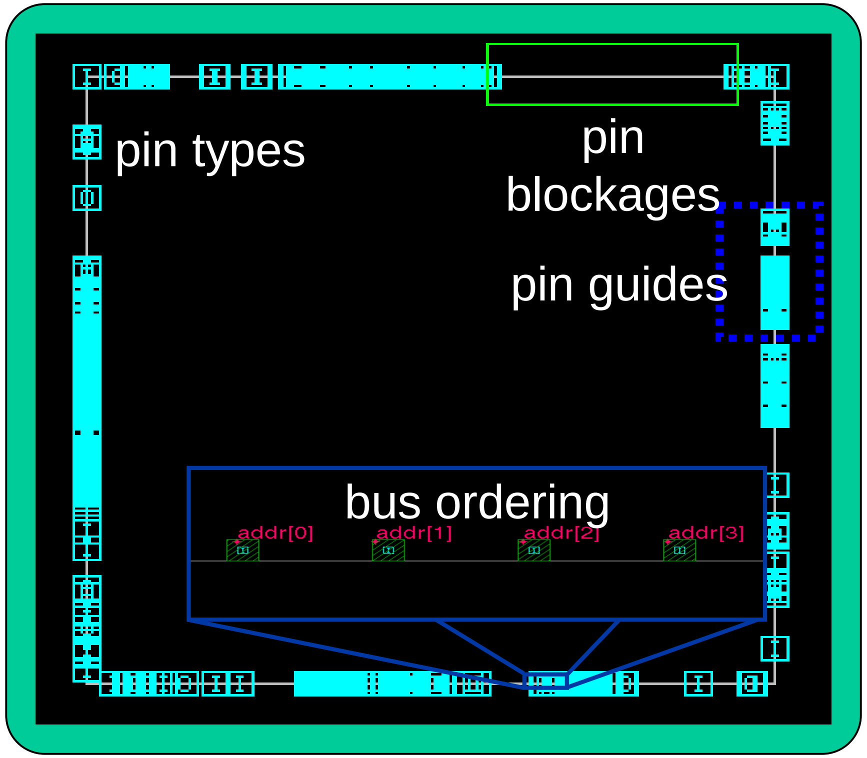


```
; tdf file

; Place the corner cells
pad "CornerLL" "bottom"
pad "CornerLR" "right"
pad "CornerTR" "top"
pad "CornerTL" "left"

; Place io and power pads
; Left and right sides count from bottom to top (excluding corner)
pad     "pad_data_0"  "left"   1
pad     "pad_data_1"  "left"   2
pad     "VDD_LEFT"    "left"   3
pad     "VSS_LEFT"    "left"   4
pad     "pad_data_2"  "left"   5

; Bottom and top sides count from left to right (excluding corner)
pad     "Clk"         "bottom"        1
pad     "A_0"         "bottom"        2
pad     "A_1"         "bottom"        3
```

By default, ports are evenly distributed on each side

Figure: Synopsys

Optimal Pin Placements For Block Level placement



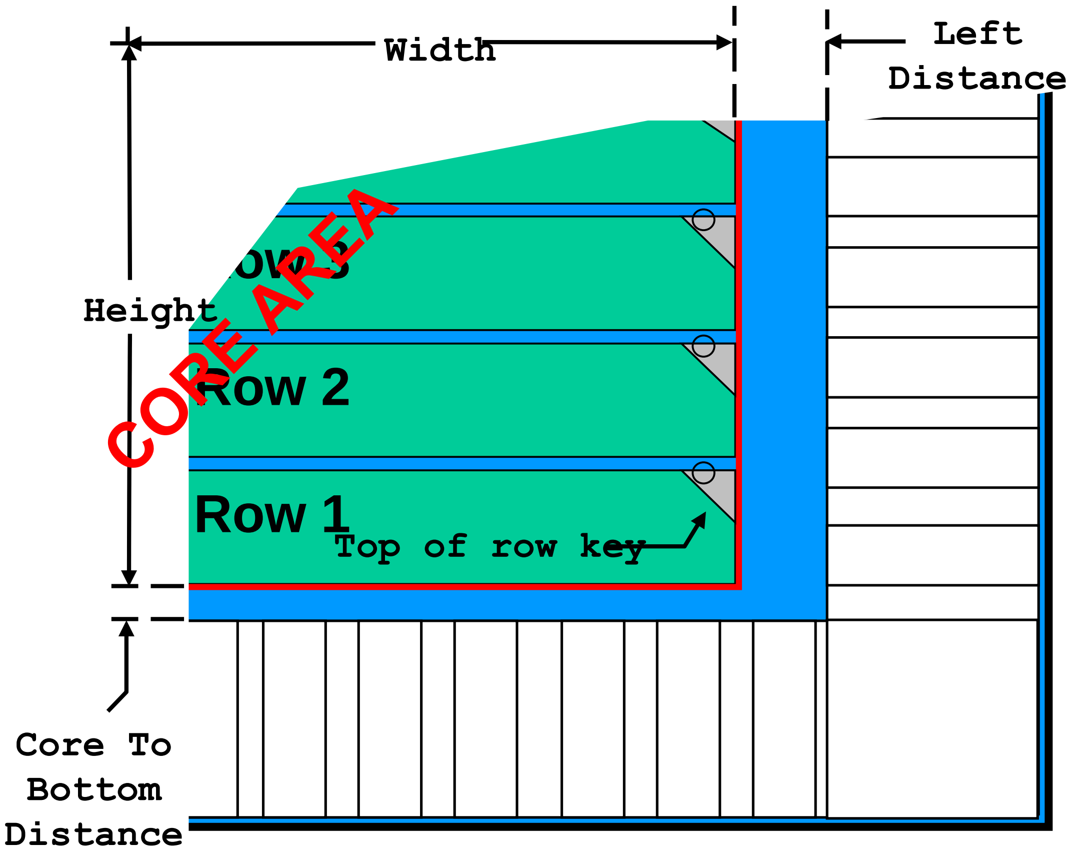
- Rectilinear and rectangular blocks
- Define relative pin locations
  - Drives cell placement
- Or, place cells first
  - Drives pin placement
- Define pin constraints
  - blockages, guides, bus ordering, layers…

Figure: Synopsys

Core Area

Control Parameters

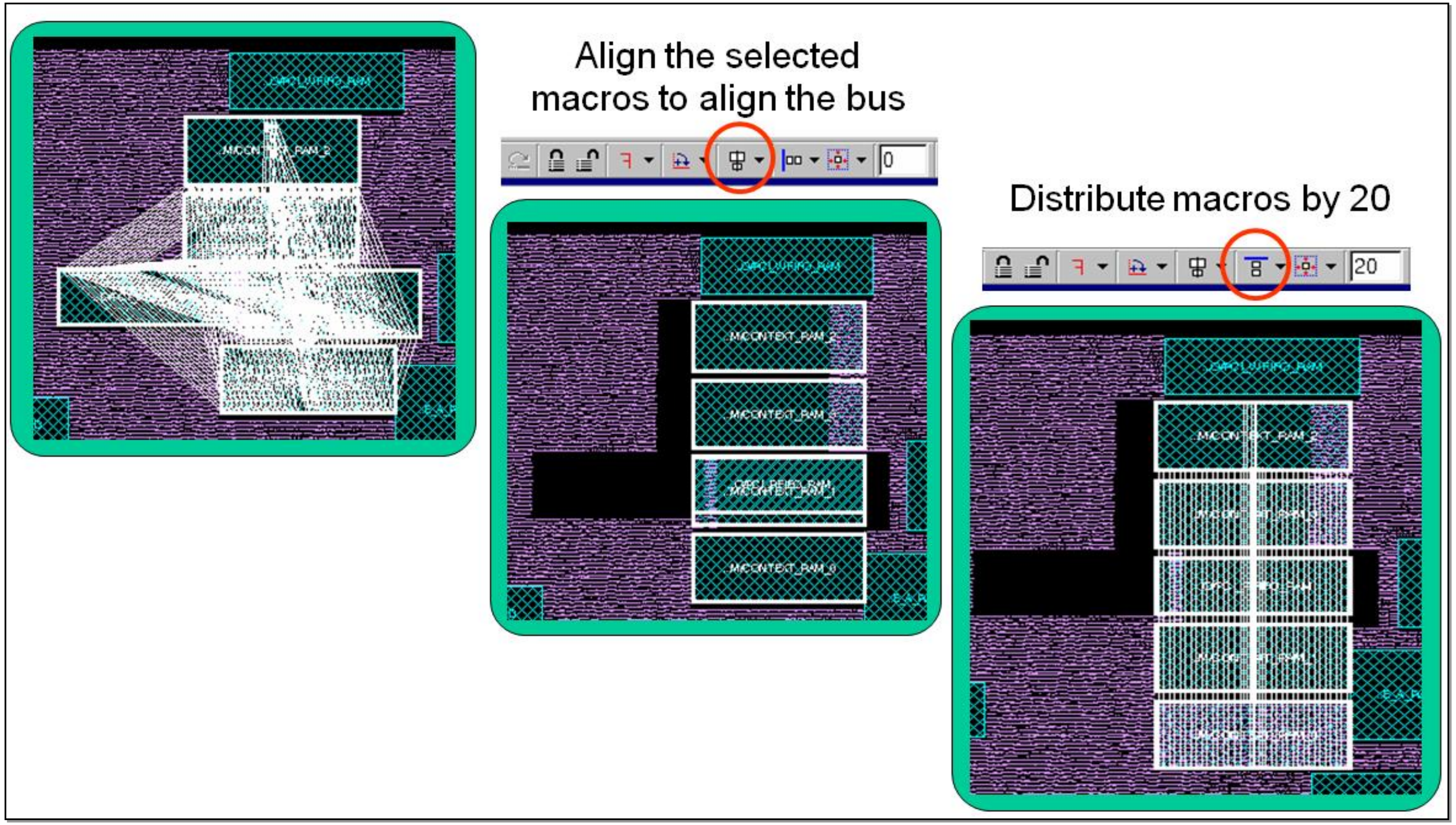
- Aspect Ratio
  - Utilization
  - Aspect ratio (H/W)
  - Row/core ratio
- Width & Height
  - Width
  - Height
  - Row/core ratio
- ……….
- ……….



Example of a horizontal, no double back, no-flip first row with Row/Core <1

Figure: Synopsys

Some operations done in Floorplanning

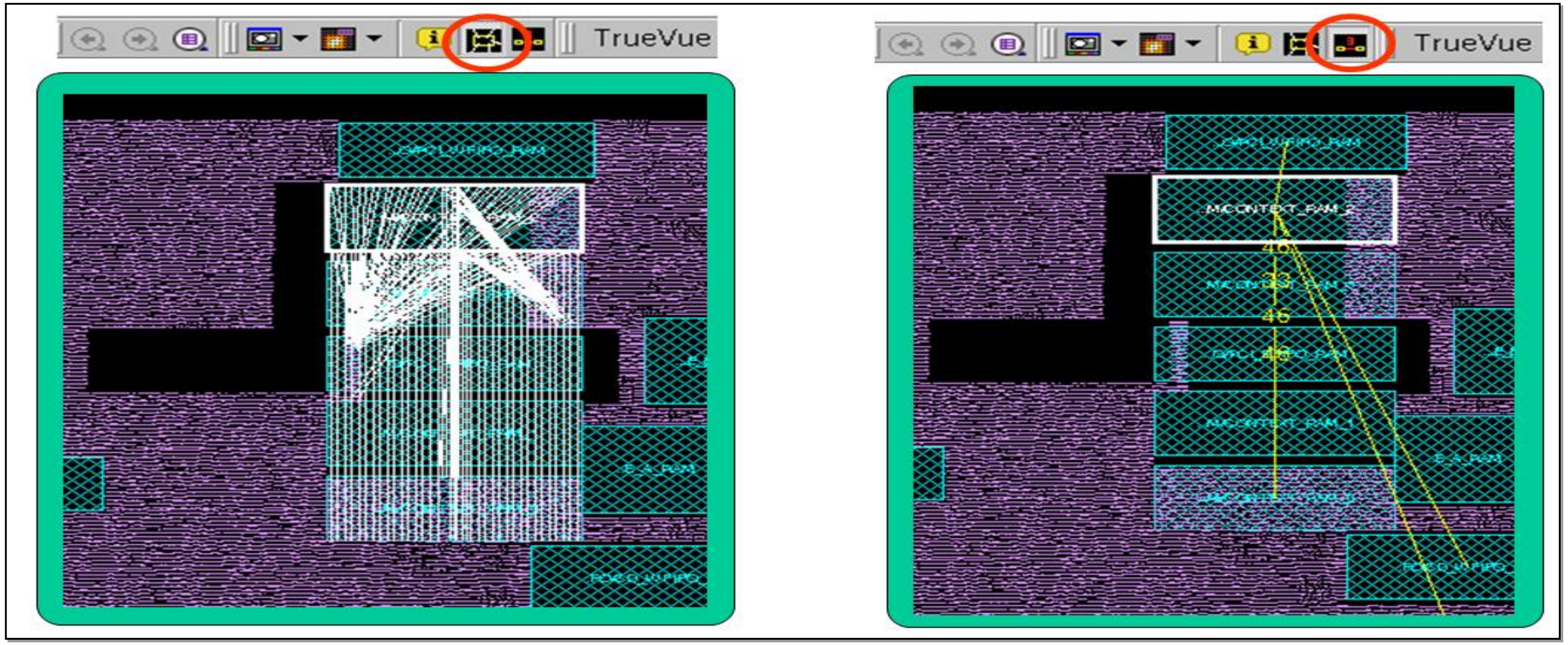


Align the selected macros to align the bus

Distribute macros by 20

Figure: Synopsys

FP Analysis: Flyline and Net Connectivity

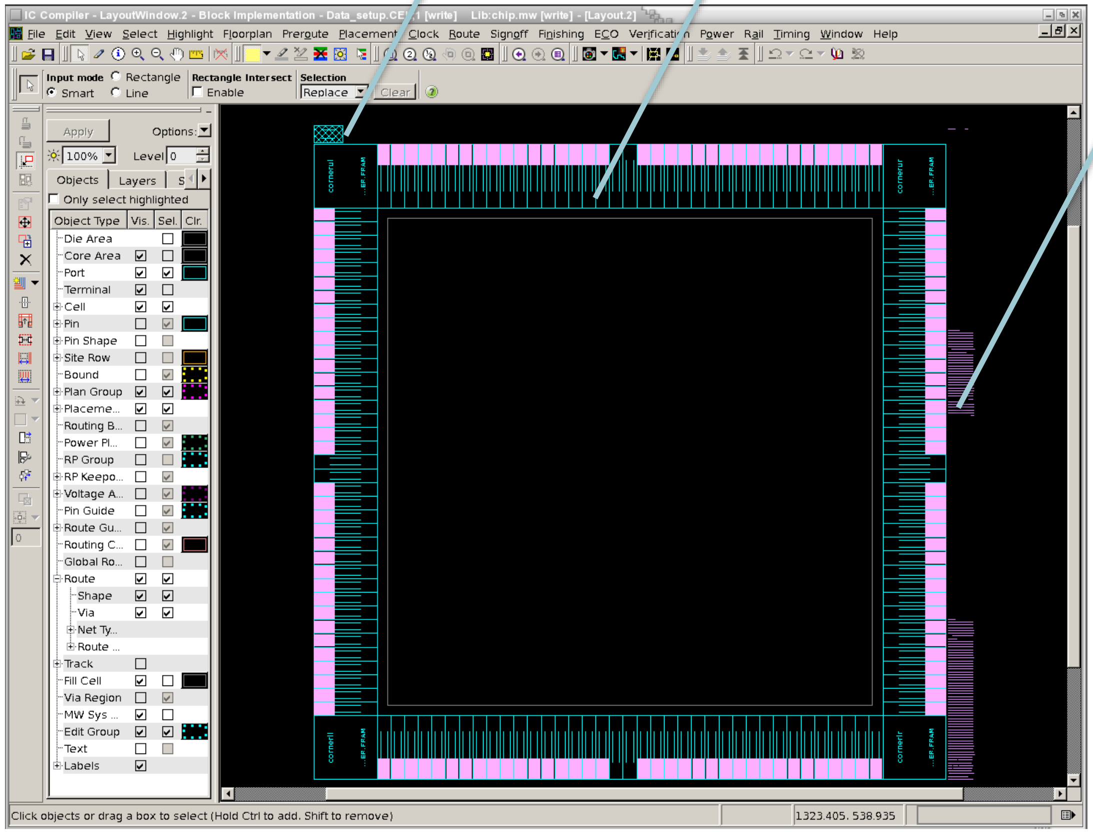


Flyline: a line representing a single net connection

- between two objects
- pins or cell instance objects

Figure: Synopsys

Design View After Floorplanning



IC Compiler - LayoutWindow.2 - Block Implementation - Data_setup.CEL;1 [write]   Lib:chip.mw [write] - [Layout.2]

File Edit View Select Highlight Floorplan Preroute Placement Clock Route Signoff Finishing ECO Verification Power Rail Timing Window Help

Input mode: ○ Rectangle ● Smart ○ Line

Rectangle Intersect: □ Enable

Selection: Replace

100%   Level 0

□ Only select highlighted

Object Type	Vis.	Sel.	Clr.
Die Area		□	■
Core Area	☑	□	■
Port	☑	☑	■
Terminal	☑	□	
Cell	☑	☑	
Pin	□	☑	■
Pin Shape	□	□	
Site Row	□	□	■
Bound	□	☑	▦
Plan Group	☑	☑	▦
Placeme...	☑	☑	
Routing B...	□	☑	
Power Pl...	□	☑	▦
RP Group	□	□	▦
RP Keepo...	□	☑	
Voltage A...	□	☑	▦
Pin Guide	□	☑	▦
Route Gu...	□	☑	
Routing C...	□	☑	■
Global Ro...	□	□	
Route	☑	☑	
Shape	☑	☑	
Via	☑	☑	
Net Ty...			
Route ...			
Track	□		
Fill Cell	☑	□	■
Via Region	□	☑	
MW Sys ...	☑	□	
Edit Group	☑	☑	▦
Text	□	□	
Labels	☑		

Click objects or drag a box to select (Hold Ctrl to add. Shift to remove)   1323.405, 538.935

PLL

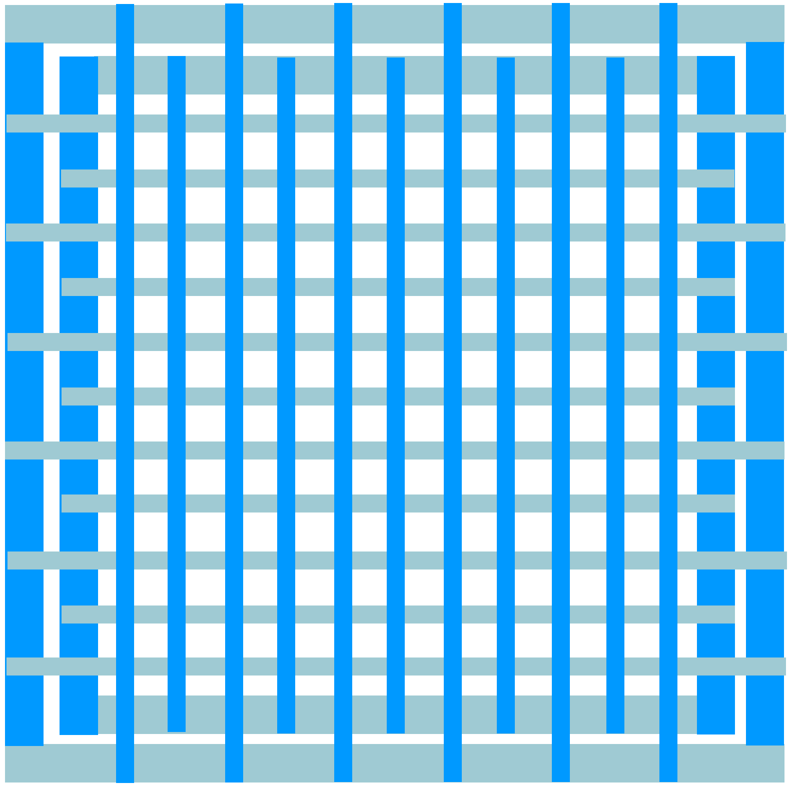
Placed I/O

Standard Cell

Figure: Synopsys

Power Network Synthesis (PNS)

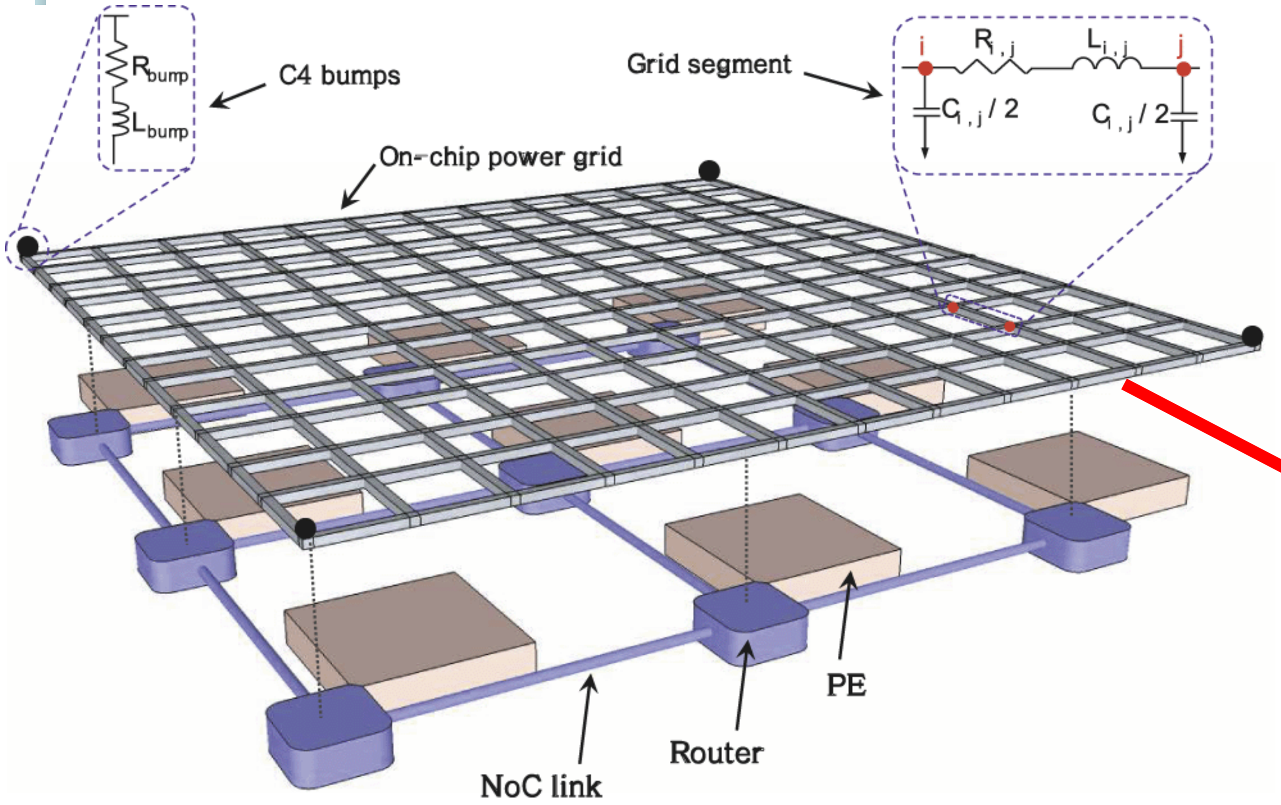
- Creating power delivery network
  - PNS currently creates a (rectilinear) power plan with/without a core ring connected to a power mesh
  - Designers need to specify
    - Number of straps: min, max
    - Width of straps: min, max
    - Width of ring
    - Layer
    - IR-drop constraint



Trunks are not shown here (they are outside the core rings, between the rings and the pads)

Figure: Synopsys

Example: Power Grid



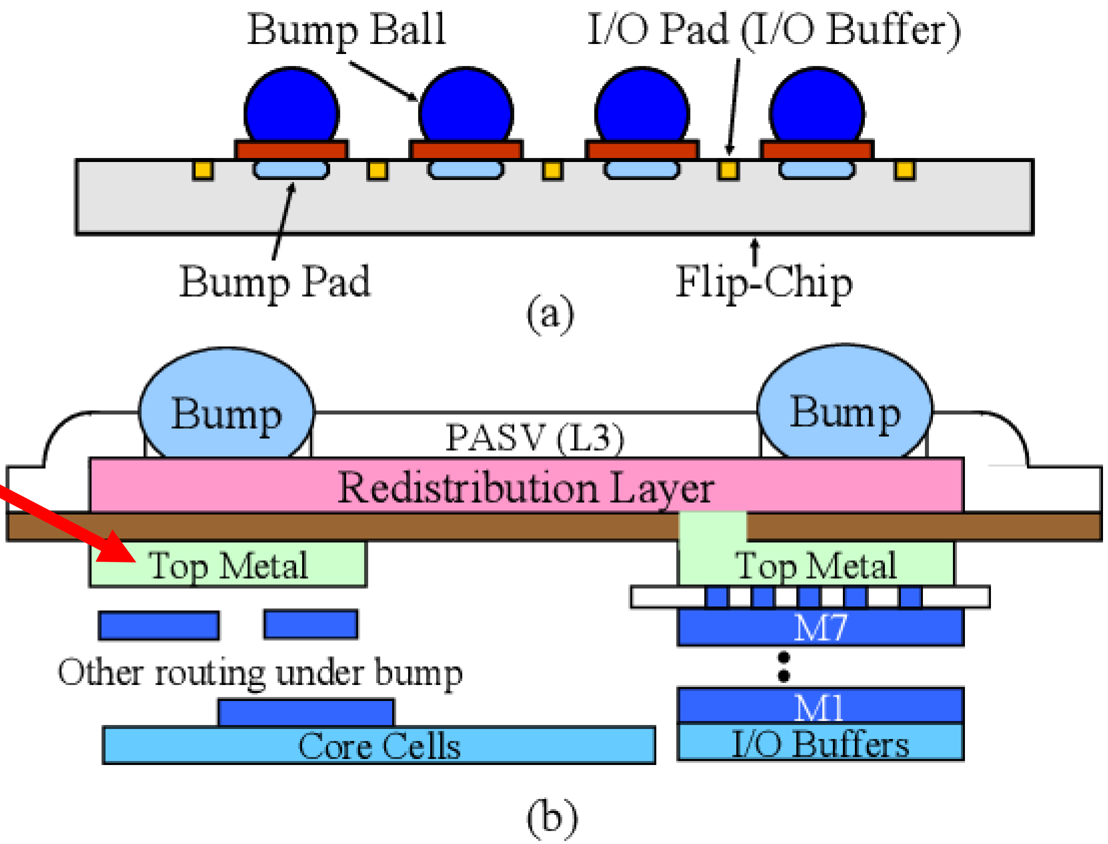


Fig. 1. (a) Flip-Chip Package. (b) Cross Section of RDL.

Source:

- Dahir N S, Mak T, Xia F, et al. Modeling and tools for power supply variations analysis in networks-on-chip[J]. IEEE Transactions on Computers, 2012, 63(3): 679-690.
- Fang J W, Chang Y W. Area-I/O flip-chip routing for chip-package co-design[C]//2008 IEEE/ACM International Conference on Computer-Aided Design. IEEE, 2008: 518-522.

RDL: extra metal layer on a chip that makes the IO Pads of an IC available in other locations of the chip, for better access to the pads where necessary

Macro Placement Constraints

- Macro placement constraints have an impact on placing and further global routing of standard cells
- Some basic constraints of macro placement
  - Alignment of macros by edges
  - Grouping macros in a way that for standard cells rectangular area should be left as much as possible
  - Alignment of macros with core boundary
- Some basic parameters of Macros Placement
  - Routability
  - Timing
  - Wire length
  - Area for standard cells

Creating a User-Defined Array of Hard Macros

Alignment to other macros, pins, and edge

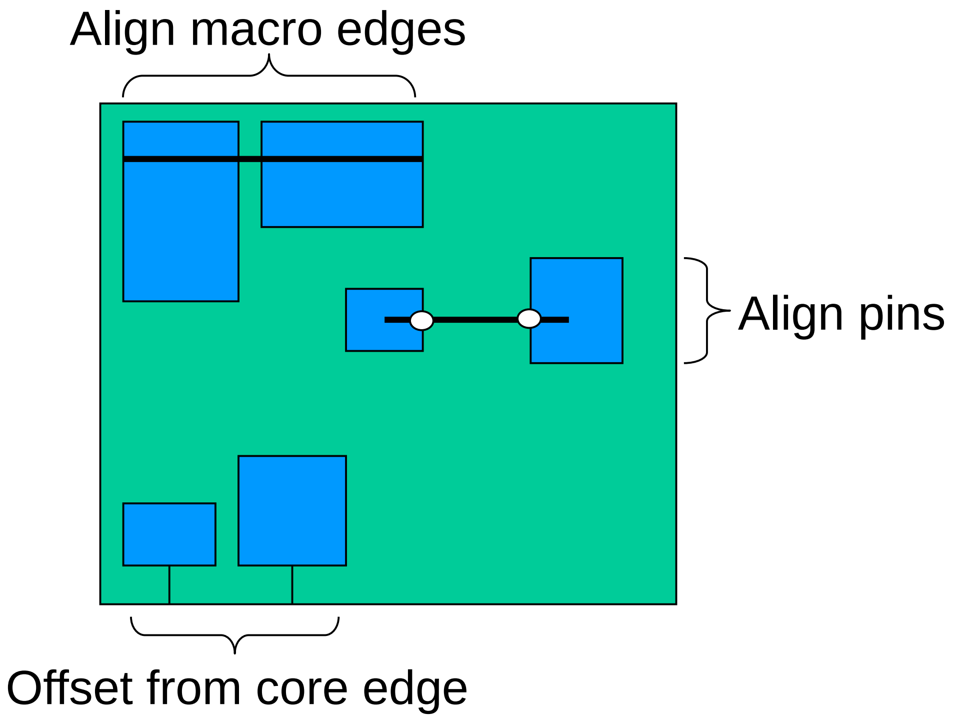
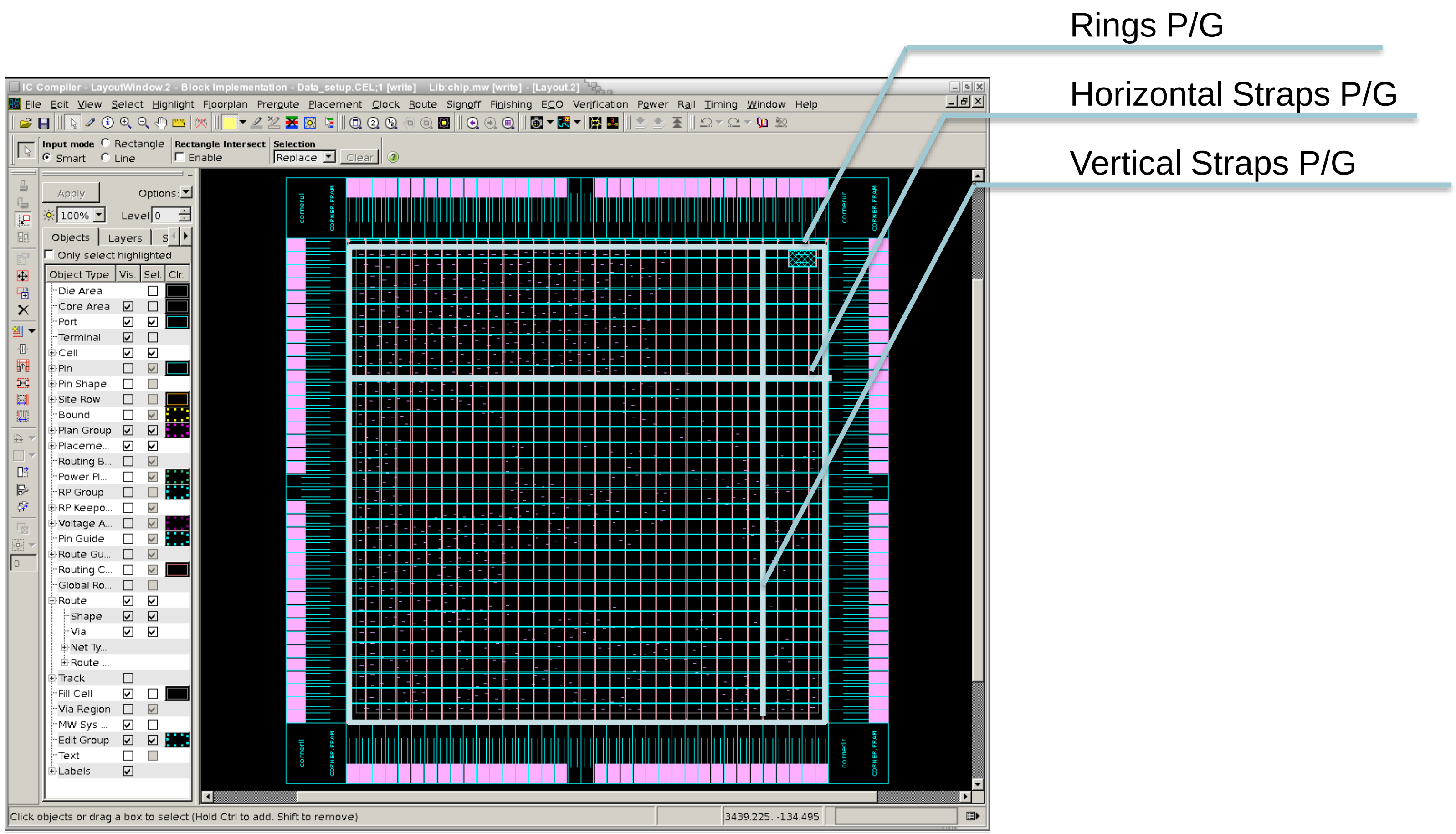


Figure: Synopsys

Rectangular Rings and Power Straps



IC Compiler - LayoutWindow.2 - Block Implementation - Data_setup.CEL;1 [write]   Lib:chip.mw [write] - [Layout.2]

Input mode ● Smart ○ Rectangle ○ Line

Rectangle Intersect ☐ Enable

Selection Replace

☐ Only select highlighted

Object Type	Vis.	Sel.	Clr.
Die Area		☐	■
Core Area	☑	☐	■
Port	☑	☑	■
Terminal	☑	☐	
Cell	☑	☑	
Pin	☐	☑	■
Pin Shape	☐	☐	
Site Row	☐	☐	■
Bound	☐	☑	▨
Plan Group	☑	☑	▨
Placeme...	☑	☑	
Routing B...	☐	☑	
Power Pl...	☐	☑	▨
RP Group	☐	☐	▨
RP Keepo...	☐	☑	
Voltage A...	☐	☑	▨
Pin Guide	☐	☑	▨
Route Gu...	☐	☑	
Routing C...	☐	☑	■
Global Ro...	☐	☐	
Route	☑	☑	
Shape	☑	☑	
Via	☑	☑	
Net Ty...			
Route ...			
Track	☐		
Fill Cell	☑	☐	■
Via Region	☐	☑	
MW Sys ...	☑	☐	
Edit Group	☑	☑	▨
Text	☐	☐	
Labels	☑		

Click objects or drag a box to select (Hold Ctrl to add. Shift to remove)

3439.225, -134.495

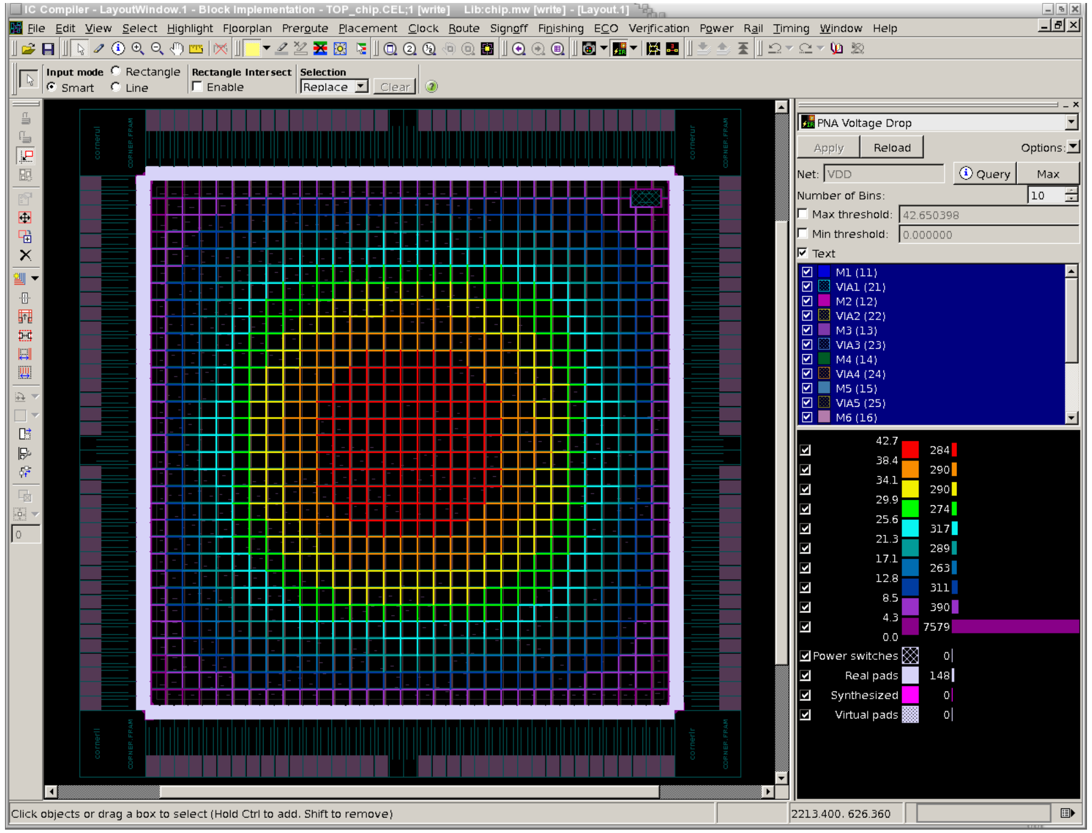
Rings P/G

Horizontal Straps P/G

Vertical Straps P/G

Figure: Synopsys

Analyze Voltage Drop



IC Compiler - LayoutWindow.1 - Block Implementation - TOP_chip.CEL;1 [write]   Lib:chip.mw [write] - [Layout.1]

File Edit View Select Highlight Floorplan Preroute Placement Clock Route Signoff Finishing ECO Verification Power Rail Timing Window Help
cornerul
CORNER.FRAM

Input mode: ● Smart ○ Rectangle ○ Line

Rectangle Intersect: □ Enable

Selection: Replace

Clear
PNA Voltage Drop
Apply
Reload
Options:
Query
Max

Net: VDD

Number of Bins: 10

□ Max threshold: 42.650398

□ Min threshold: 0.000000

☑ Text

☑ M1 (11)
☑ VIA1 (21)
☑ M2 (12)
☑ VIA2 (22)
☑ M3 (13)
☑ VIA3 (23)
☑ M4 (14)
☑ VIA4 (24)
☑ M5 (15)
☑ VIA5 (25)
☑ M6 (16)

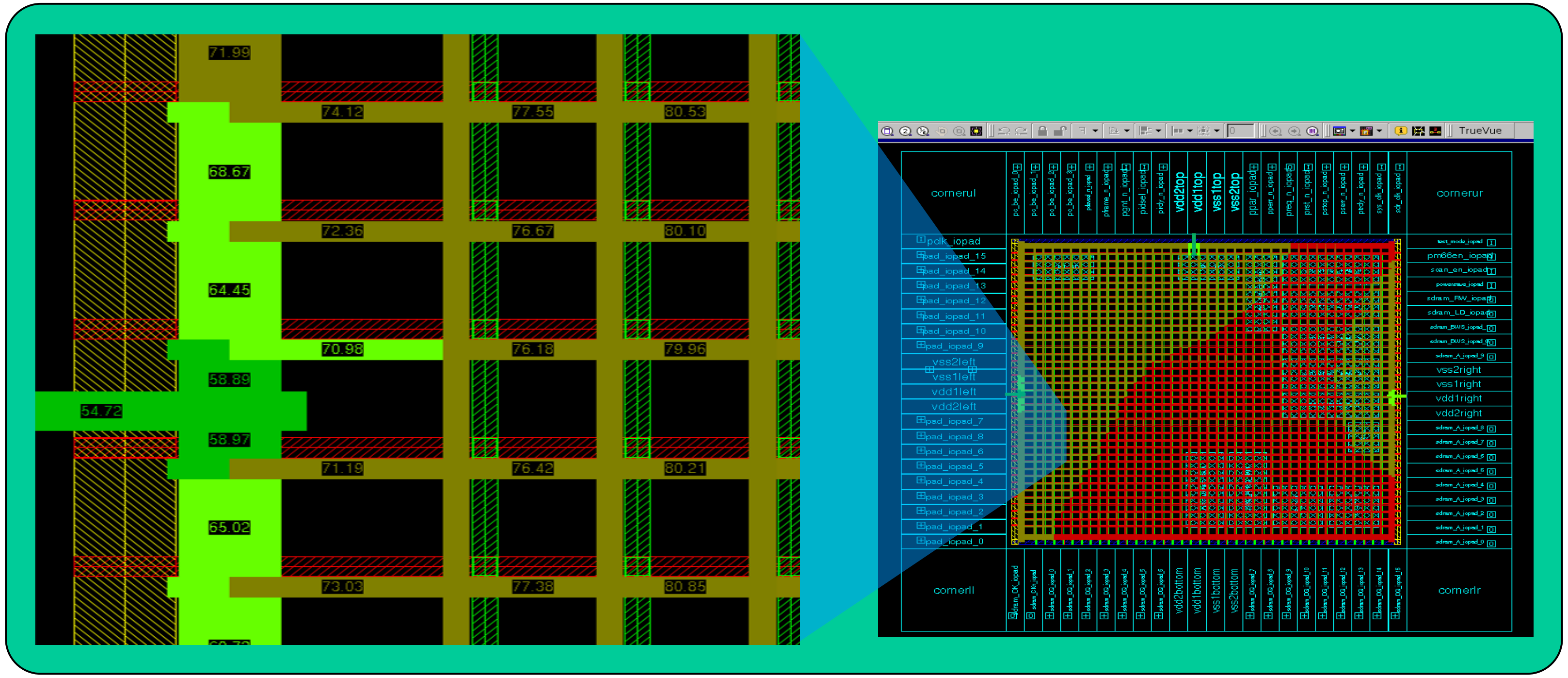
☑	38.4 – 42.7	284
☑	34.1 – 38.4	290
☑	29.9 – 34.1	290
☑	25.6 – 29.9	274
☑	21.3 – 25.6	317
☑	17.1 – 21.3	289
☑	12.8 – 17.1	263
☑	8.5 – 12.8	311
☑	4.3 – 8.5	390
☑	0.0 – 4.3	7579
☑	Power switches	0
☑	Real pads	148
☑	Synthesized	0
☑	Virtual pads	0

Click objects or drag a box to select (Hold Ctrl to add. Shift to remove)

2213.400, 626.360

Figure: Synopsys

Display Voltage (IR) Drop



Can continue to add/delete virtual power pads to see if voltage drop map gets better

Figure: Synopsys

Floorplanning problem

The floorplanning problem is to plan the positions and shapes of the modules at the beginning of the design cycle to optimize the circuit performance:

- chip area
- total wirelength
- delay of critical path
- routability
- others, e.g**., noise, heat** dissipation, etc.

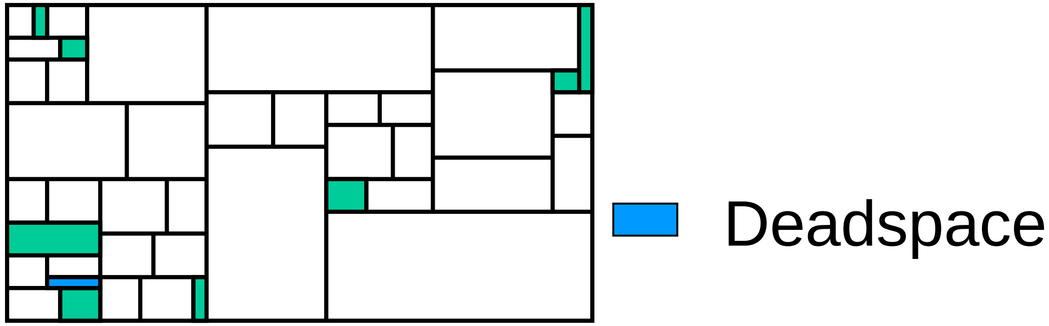
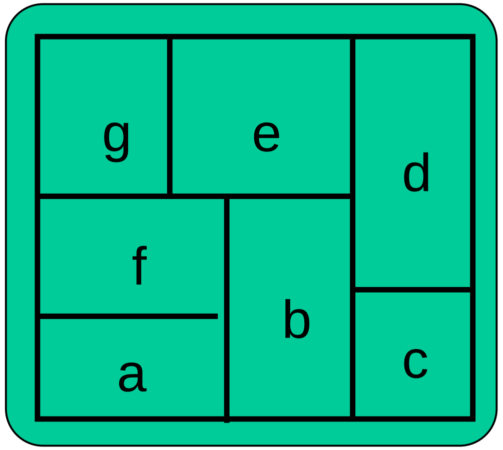


Figure: Synopsys

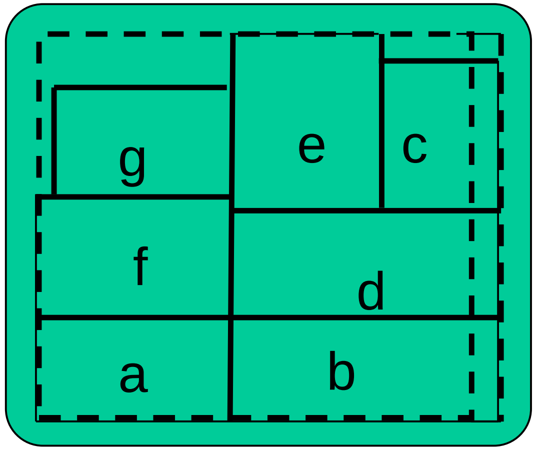
Floorplanning problem

Problem formulation

Input: n Blocks with areas A1, ... , An
Bounds ri and si on the aspect ratio of block Bi
Output: Coordinates (xi, yi), width wi and height hi for each block such that
hi wi = Ai and ri ≤ hi/wi ≤ si
Objective: Optimize the circuit performance.



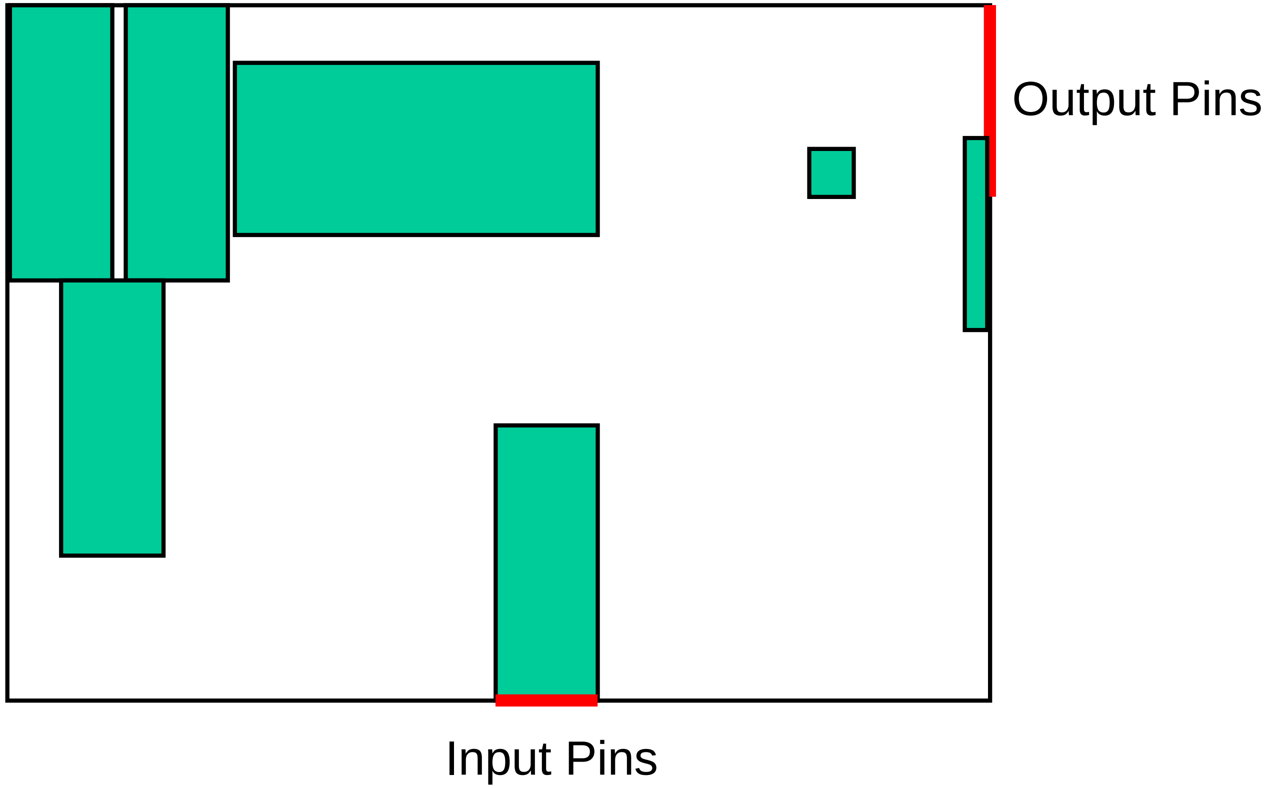
An optimal floorplan in terms of area



A non−optimal floorplan

Figure: Synopsys

A bad floorplan

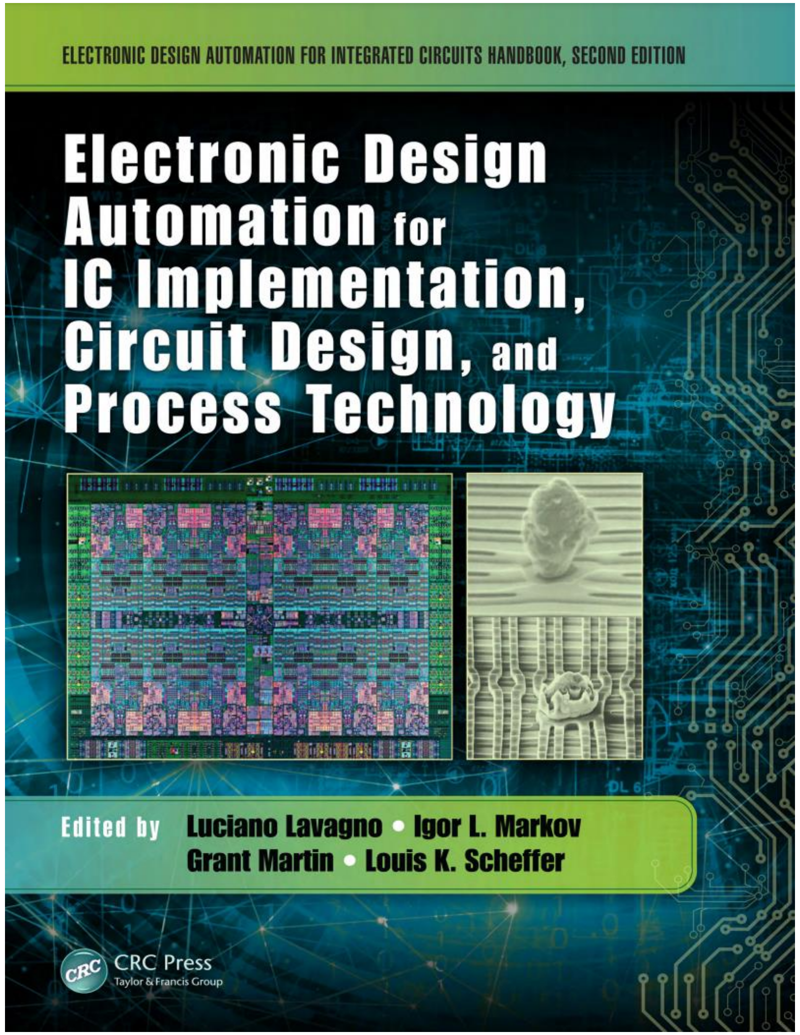


Where are we Heading?

- ASIC Design Flow III

Action Items

- HW#4 is coming!
- Reading Materials
  - Slides



ELECTRONIC DESIGN AUTOMATION FOR INTEGRATED CIRCUITS HANDBOOK, SECOND EDITION

Electronic Design Automation for IC Implementation, Circuit Design, and Process Technology

Edited by Luciano Lavagno • Igor L. Markov
Grant Martin • Louis K. Scheffer

CRC Press
Taylor & Francis Group

Acknowledgement

Slides in this topic are inspired in part by material developed and copyright by:

- Synopsys Courseware