
Digital VLSI system Design

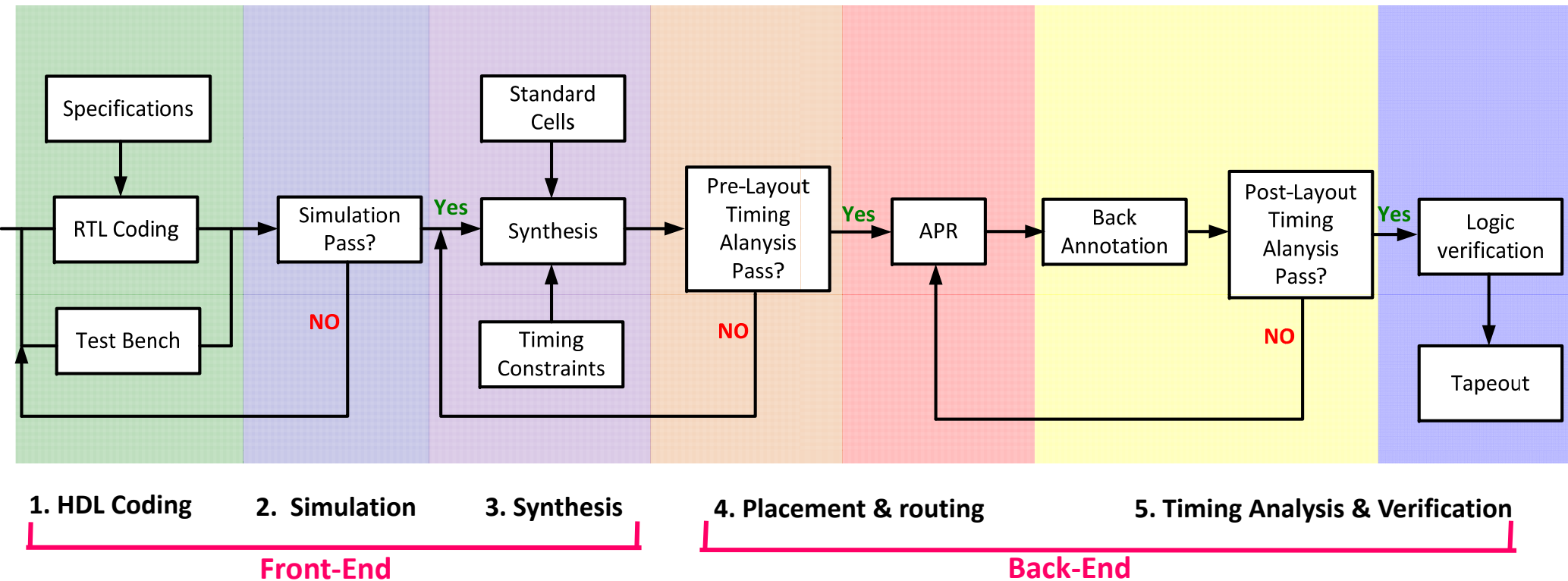
ASIC Design Flow

Mahdi Shabany

Department of Electrical Engineering
Sharif University of technology



ASIC Design Flow

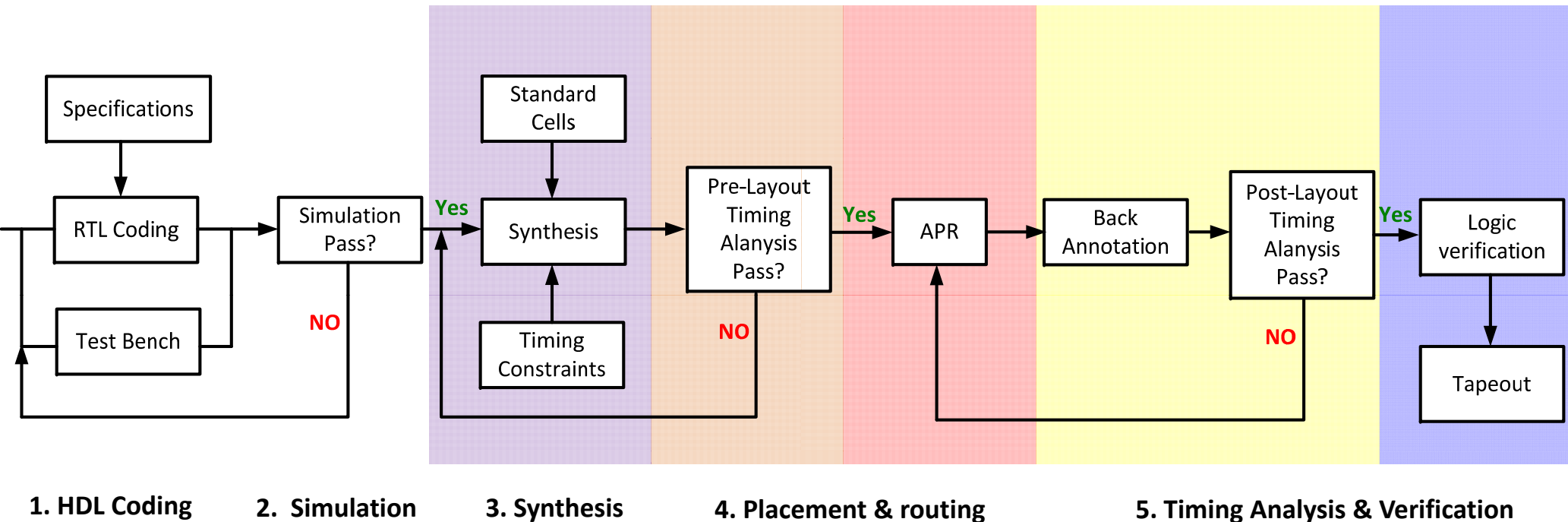


□ In this lecture we learn all the above steps in detail for

➤ ASIC Platform



ASIC Design Flow

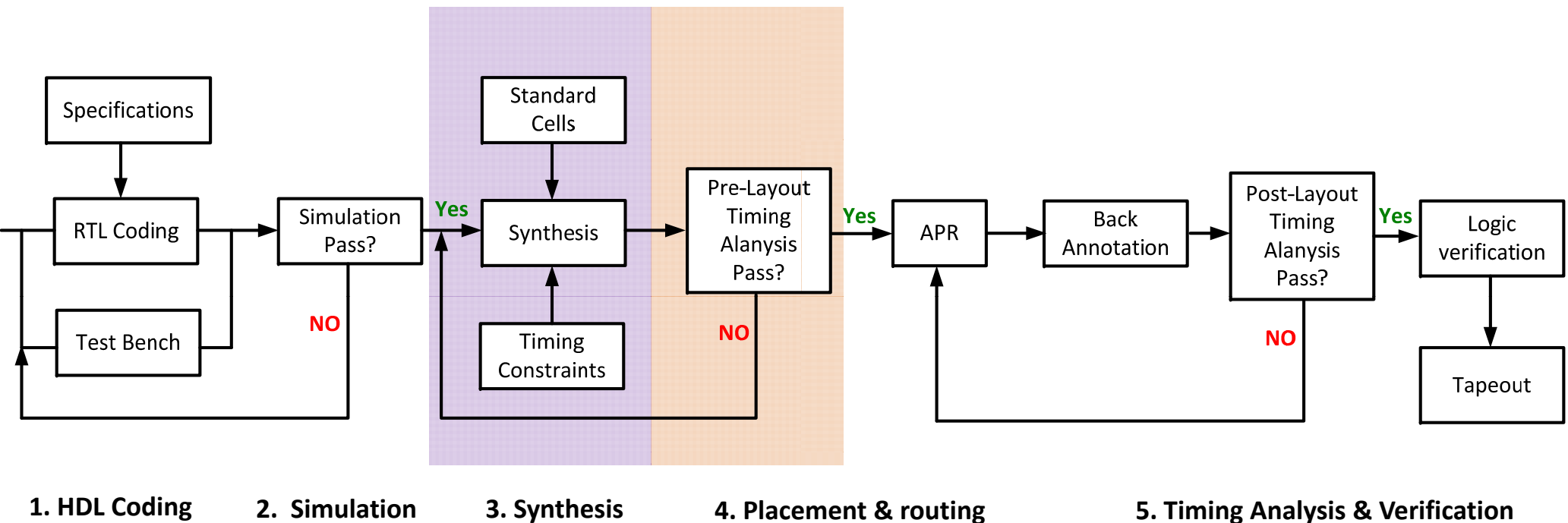


□ In this lecture we learn all the above steps in detail for

➤ ASIC Platform



ASIC Design Flow: Synthesis

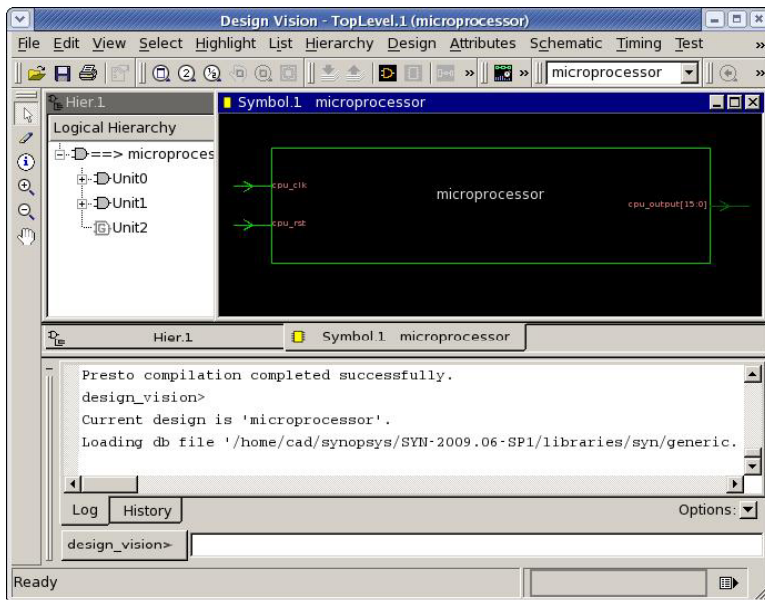
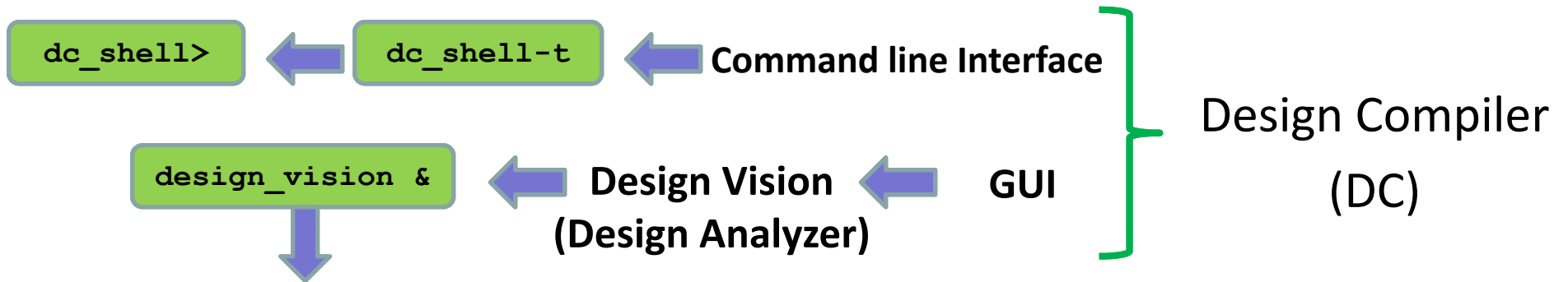


❑ “Synopsys Design Compiler” is a popular synthesis tool both in academia & industry



Design Compiler

ابزار سنتزی که ما در اینجا از آن استفاده می کنیم، نرم افزار Synopsys Design Compiler می باشد.



Design Compiler

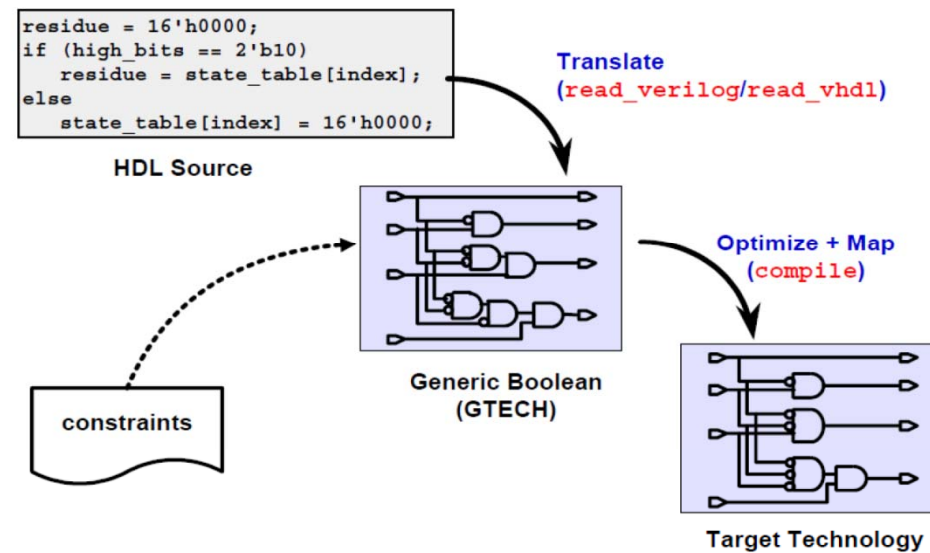
- ❑ “Design Compiler” can perform both the RTL and logical synthesis
- ❑ Design compiler also provides a framework to perform both flat and hierarchical synthesis

❑ Translation is done by the following commands in DC:

- read_verilog
- analyze/elaborate

❑ These commands perform

- Syntax checking
- Synth. with general elements (GTECH)



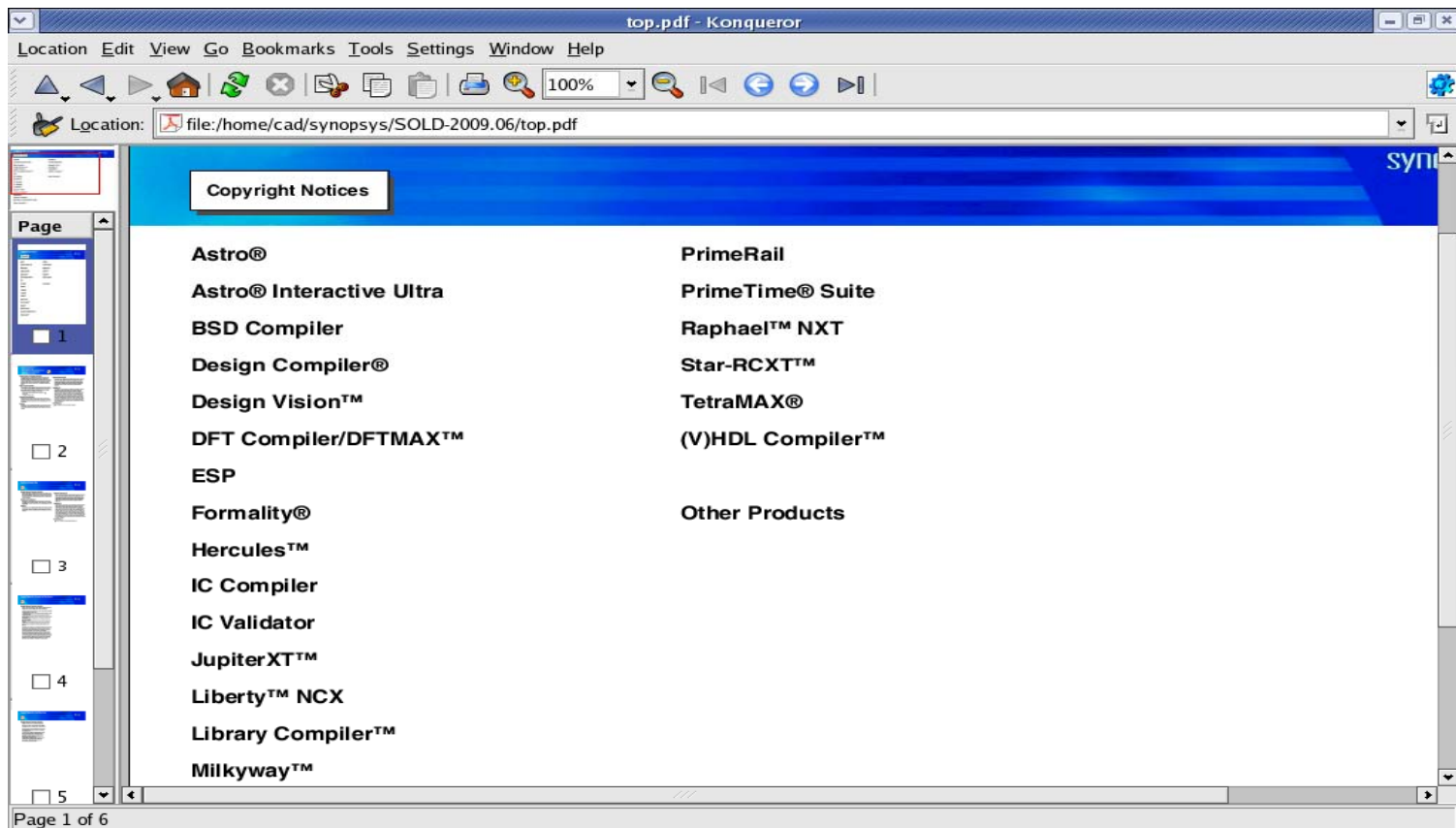
❑ GTECH are technology-independent cells



Design Compiler: Help

❑ To find help on any product of the Synopsys you need to find the SOLD file

`$$SYNOPSIS/SOLD-2009.06/top.pdf`



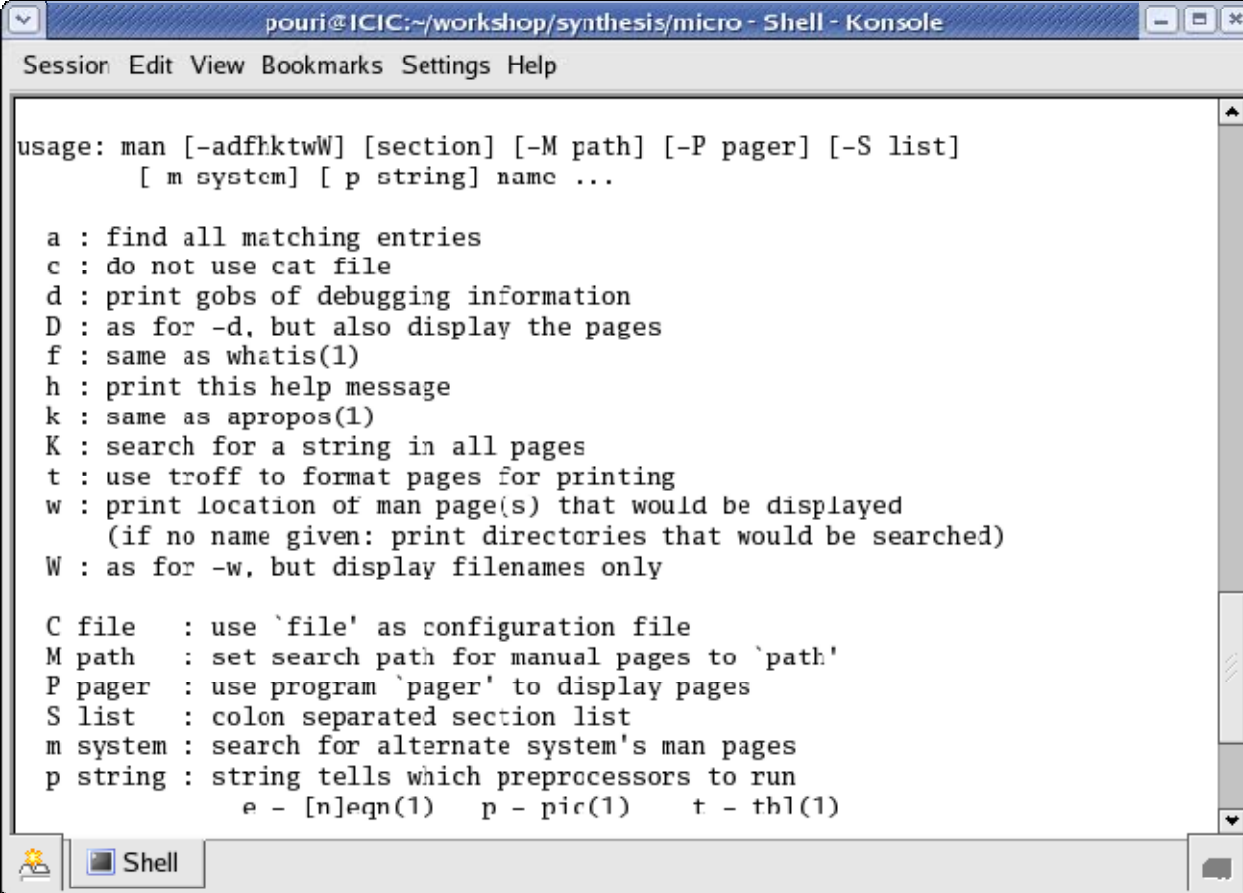
Help for all
Synopsys products



Design Compiler: Help

❑ To find help on any DC command:

```
dc_shell> man
```



```
pouri@ICIC:~/workshop/synthesis/micro - Shell - Konsole
Session Edit View Bookmarks Settings Help

usage: man [-adfkhktwW] [section] [-M path] [-P pager] [-S list]
        [ m system] [ p string] name ...

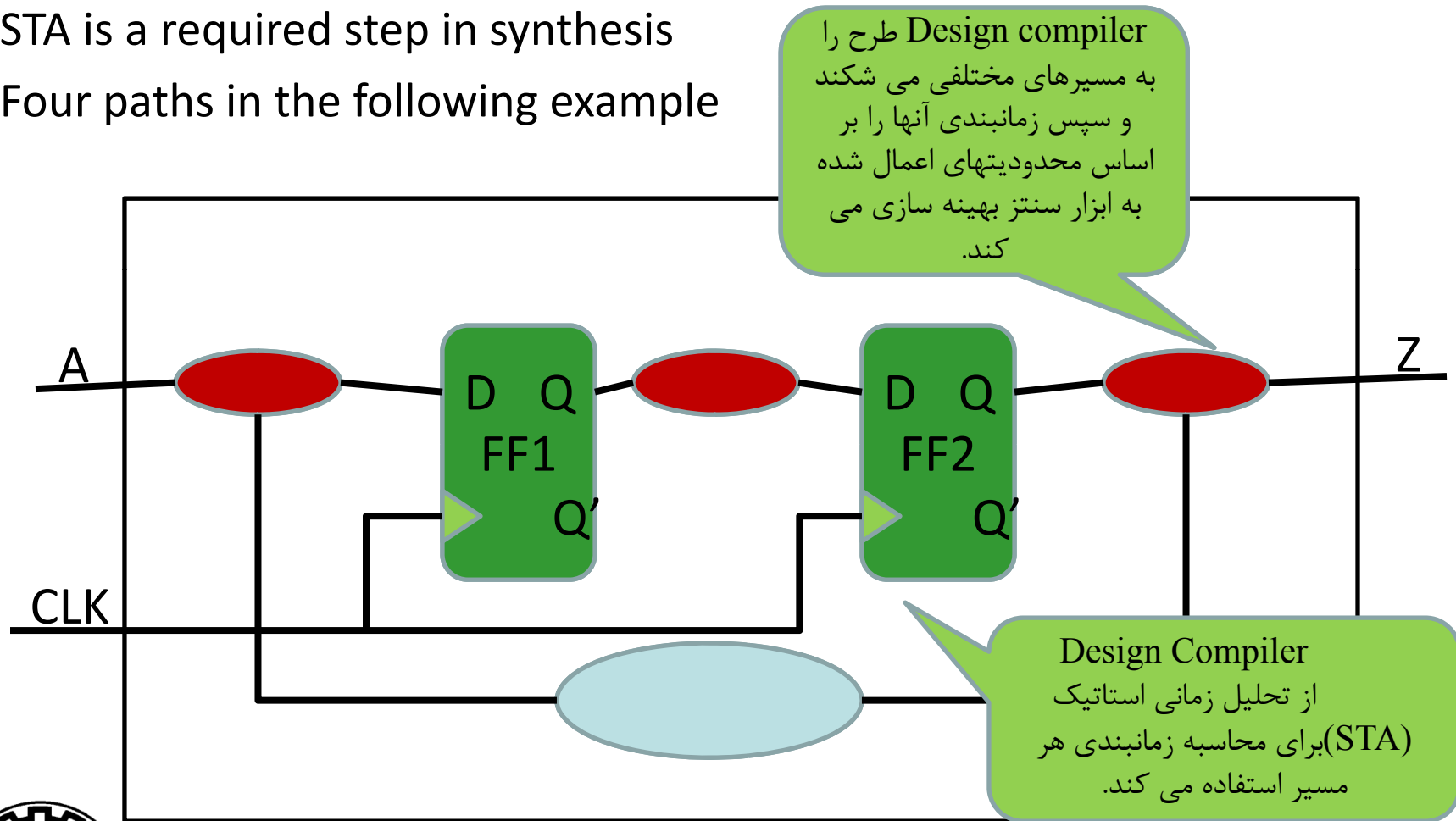
a : find all matching entries
c : do not use cat file
d : print gobs of debugging information
D : as for -d, but also display the pages
f : same as whatis(1)
h : print this help message
k : same as apropos(1)
K : search for a string in all pages
t : use troff to format pages for printing
w : print location of man page(s) that would be displayed
    (if no name given: print directories that would be searched)
W : as for -w, but display filenames only

C file   : use `file' as configuration file
M path   : set search path for manual pages to `path'
P pager  : use program `pager' to display pages
S list   : colon separated section list
m system : search for alternate system's man pages
p string : string tells which preprocessors to run
          e - [n]eqn(1)  p - pic(1)  t - tbl(1)
```



Design Compiler: Help

- ❑ Synthesis is a path-based process
- ❑ STA is a required step in synthesis
- ❑ Four paths in the following example



Design Compiler: Initial Setup

Technology
Libraries

DC Setup File



Design Compiler: Initial Setup

Technology
Libraries

DC Setup File



Design Compiler: Initial Setup

- ❑ Synopsys has a setup file called “.synopsys_dc.setup”
- ❑ Three libraries should be specified in this setup file.
 - **Technology Files**
 - Information about the functionality and characteristics of standard cells
 - Wireload models to calculate the wire delays
 - Provided and supported by semiconductor companies
 - in .lib format, which can be compiled to .db format by Synopsys Library Compiler
 - **Symbol Files**
 - Graphical symbols of each cell, used to show a cell symbol in design schematic
 - Provided and supported by semiconductor companies
 - **Design Ware Files**
 - Used to implement operations like “+ - * > =< “

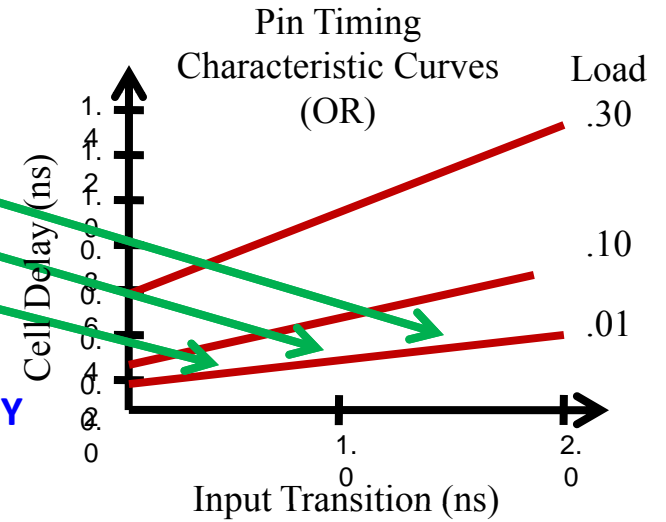


Design Compiler: Initial Setup : Technology Files

```

cell ( OR2_3 ) { ← Cell Name
  area : 8.000 ; ← Cell Area (μm²)
  pin ( Y ) {
    direction : output;
    timing ( ) {
      related_pin : "A" ;
      timing_sense : positive_unate ;
      rise_propagation (drive_3_table_1) {
        values ("0.2616, 0.2711, 0.2831,...")
      }
      rise_transition (drive_3_table_2) {
        values ("0.0223, 0.0254, ...")
      }
      . . . . .
      function : "(A | B)"; ← Pin Y Function
      max_capacitance : 1.14810 ;
      min_capacitance : 0.00220 ; ← Design Rules for Pin Y
    }
  }
  pin ( A ) {
    direction : input;
    capacitance : 0.012000; ← Electrical Characteristics of Pin A
    . . . . .
  }
}

```



Design Compiler: Initial Setup

Technology
Libraries

DC Setup File



Design Compiler: Initial Setup: DC Setup File

- Synopsys has a setup file called **“.synopsys_dc.setup”**

دایرکتوری که در `search_path` مشخص می شود، دایرکتوریهایی است که محل کتابخانه های مختلف را مشخص می کند.

کتابخانه نسبت داده شده به متغیر `target_library`، کتابخانه هدف است که توسط Design Compiler برای ساخت مدار بکار میرود و در طول عملیات نگاشت، Design Compiler گیتهای با عملکرد صحیح را از این کتابخانه انتخاب می کند.

```
# synopsys setup file
```

```
set search_path "$search_path ./unmapped"
```

```
set target_library core_slow.db
```

```
set link_library "* core_slow.db"
```

```
set symbol_library core.sdb
```

* یعنی DC ابتدا در حافظه خود (`search_path`) و سپس در `link_library` جستجو می کند.

`link_library` متغیری است که تمام ارجاعات به سلولهای طرح در آن مشخص شده است. بنابراین کتابخانه های موجود در `target_library` و در این متغیر مشخص می-شوند.

متغیر `symbol_library` کتابخانه سمبلهای عناصر طرح را مشخص می کند.



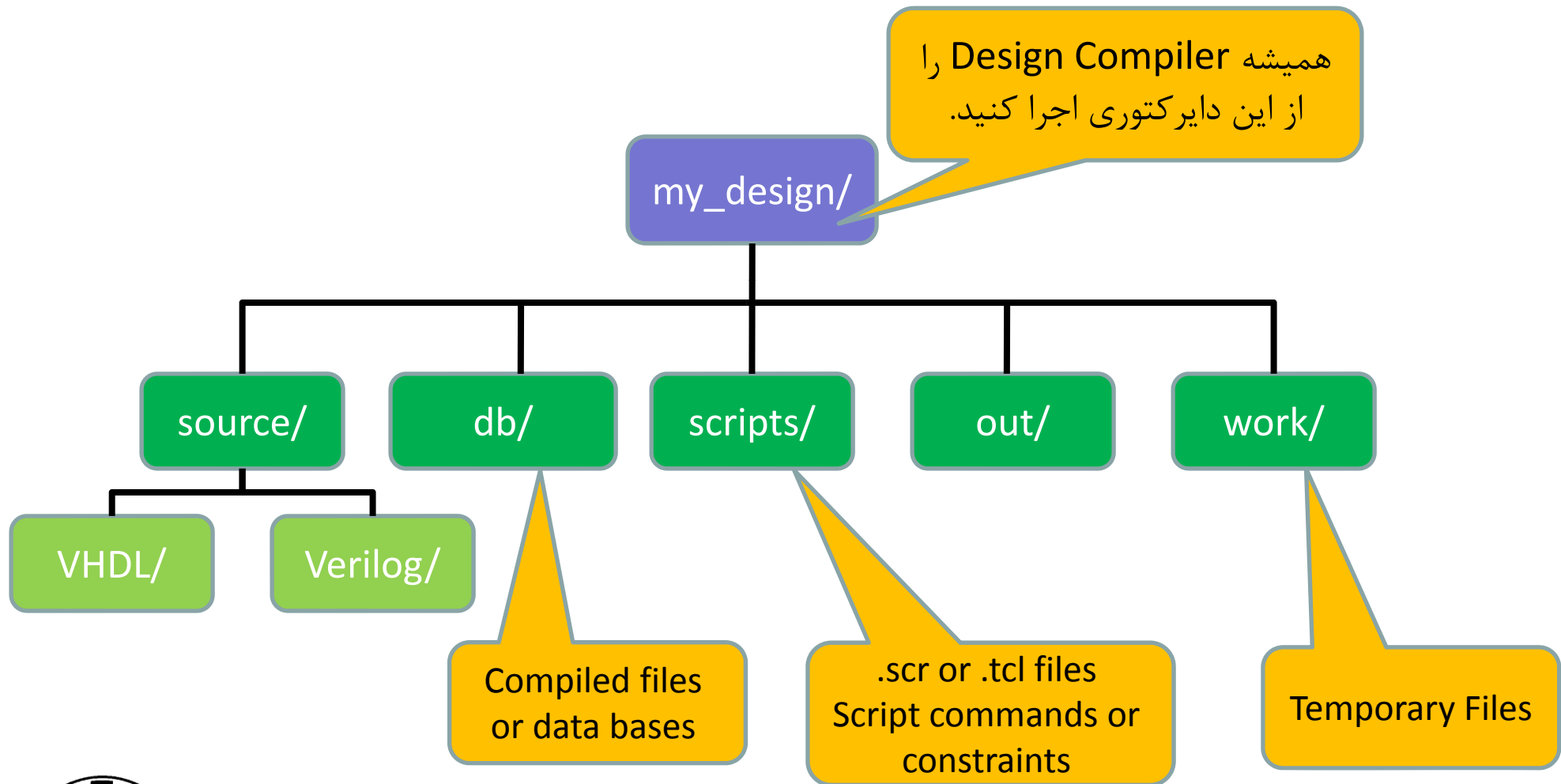
Design Compiler: Initial Setup: DC Setup File

- Summary: Setup file libraries

Library Type	Variable	Default	File Extension
Target Library	target_library	{“your_library.db”}	.db
Link Library	link_library	{“*”, “your_library.db”}	.db
Symbol Library	symbol_library	{“your_library.sdb”}	.sdb
Design ware Library	synthetic_library	{ }	.sldb

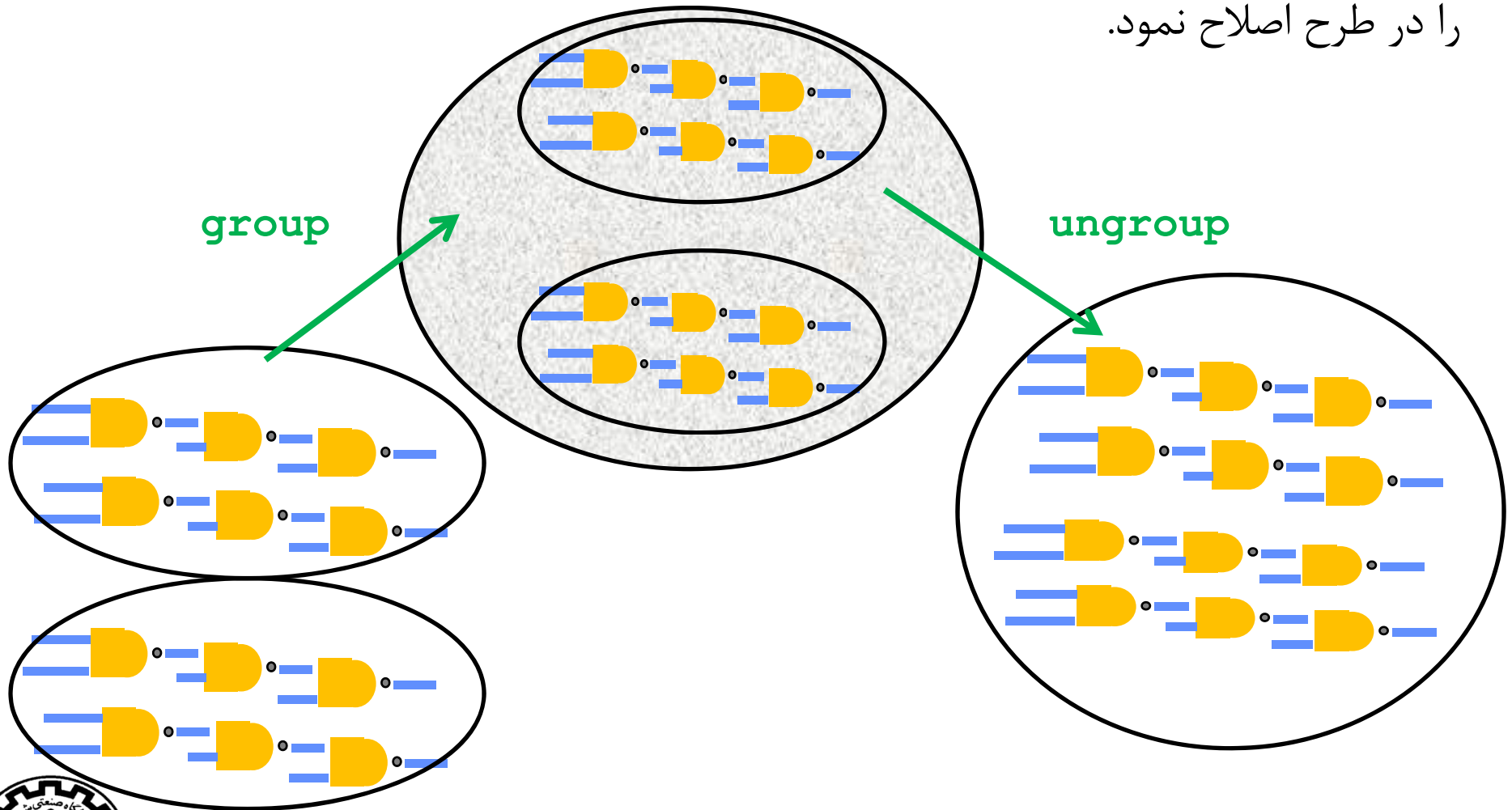


Design Compiler: Directory Structure



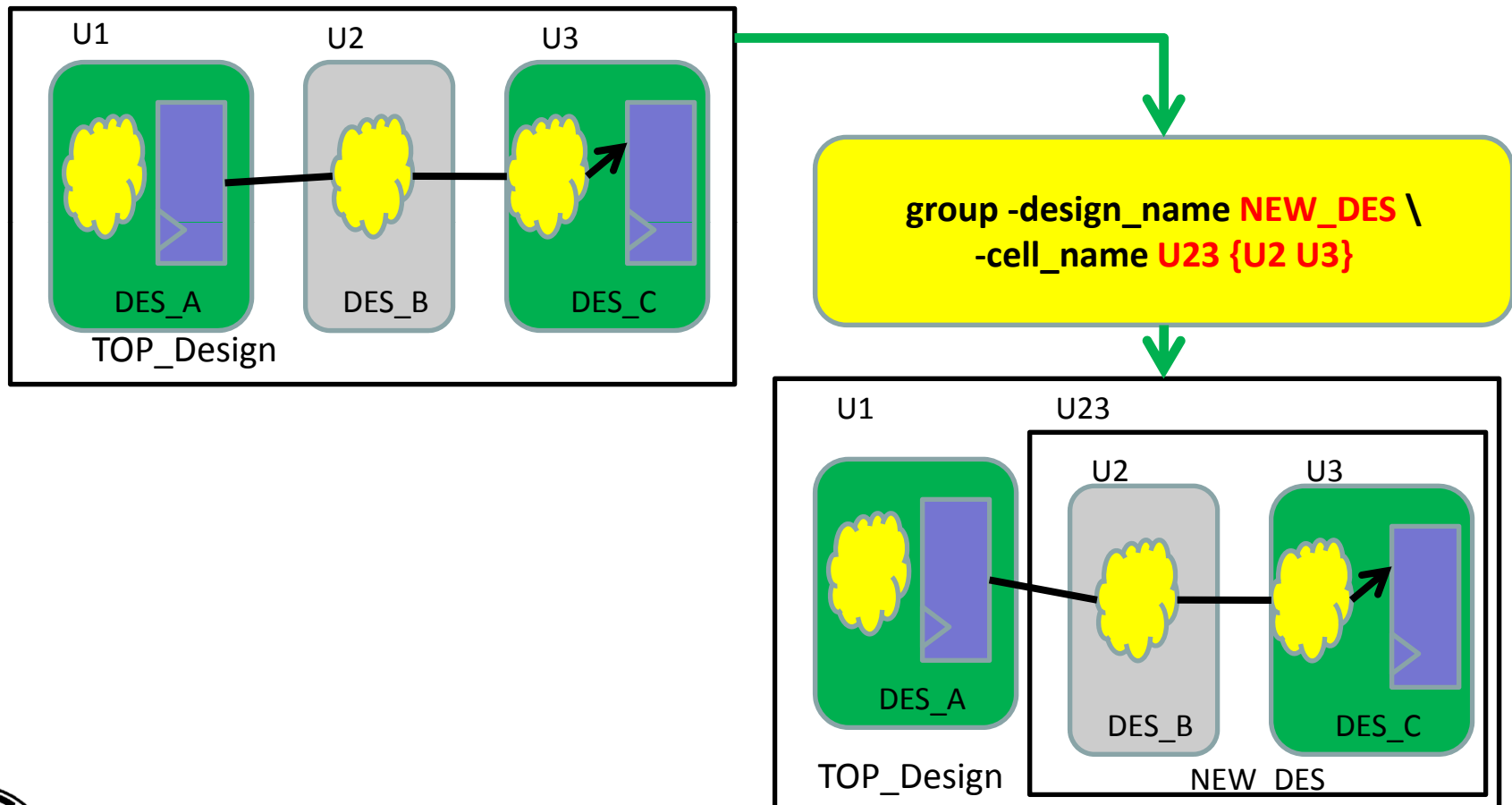
Partitioning in Design Compiler:

□ در Design Compiler با استفاده از دو دستور **group** و **ungroup** می توان جزءبندی را در طرح اصلاح نمود.



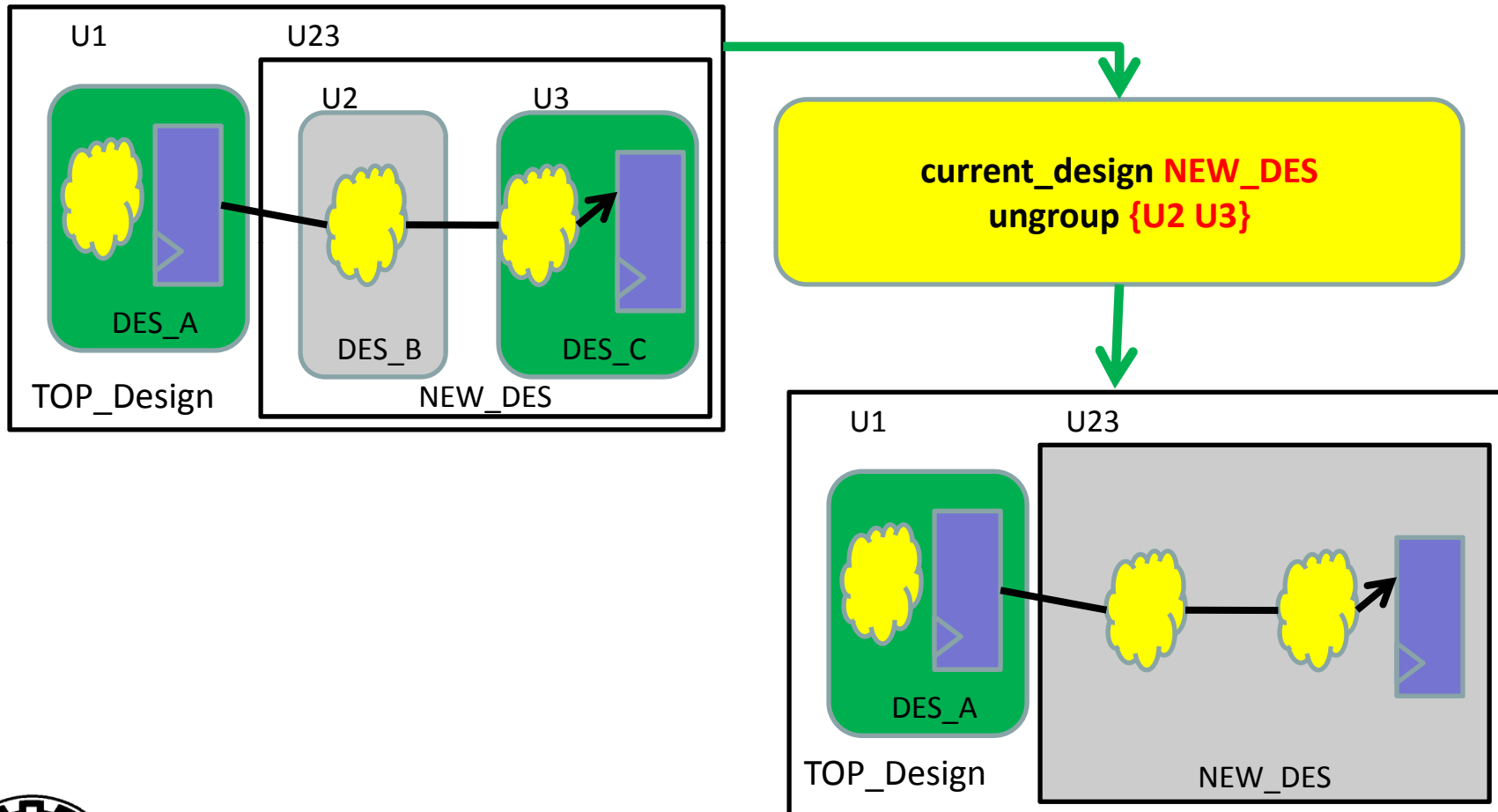
Partitioning in Design Compiler:

دستور **group** می تواند یک سلسله مراتب جدید در طرح ایجاد نماید. □



Partitioning in Design Compiler:

دستور **ungroup** تمام سطوح سلسله مراتبی را حذف می کند. □



Timing and Design Constraints in DC:

Timing and Area

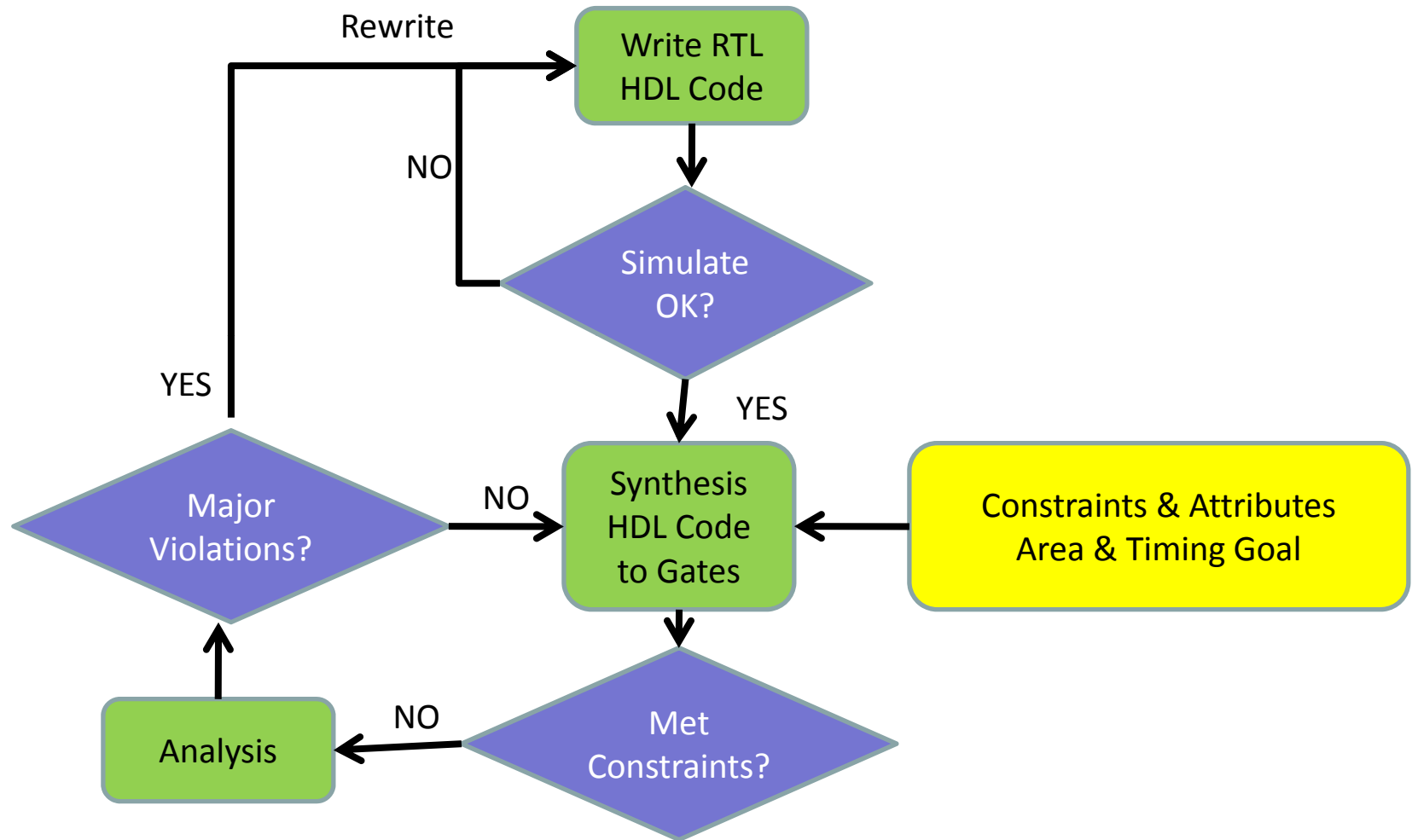
Environmental Attributes

Design Rules and Min Timing

Timing Analysis



Timing and Design Constraints in DC:



Timing and Design Constraints in DC:

Timing and Area

Environmental Attributes

Design Rules and Min Timing

Timing Analysis



Timing and Design Constraints in DC:

- ❑ The synthesizer is "lazy", if you don't set the proper constraints it will select constraints that will make him work less.

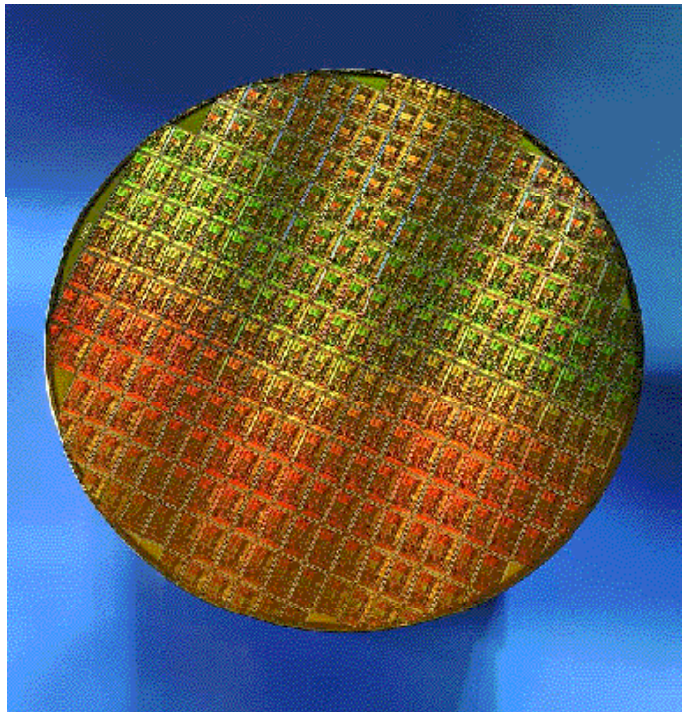
Always set proper constraints

- ❑ Timing Constraint
 - Max delay combinational delay
 - Max area total circuit area
 - Max power for power limitation
 - Setting the constraint does not guarantee the result!
- ❑ By default, timing constraints have higher priority over area constraint
 - "-ignore_tns" -> give area priority over timing



Area Constraint in DC:

□ به منظور کاهش هزینه های ساخت مدار، اعمال محدودیت مساحت در طرح بسیار ضروری به نظر می رسد.



```
current_design <my_design>  
set_max_area 100
```

100 μm

Minimum Area

```
current_design <my_design>  
set_max_area 0
```

When both area and delay constraints are set, design compiler will give speed priority



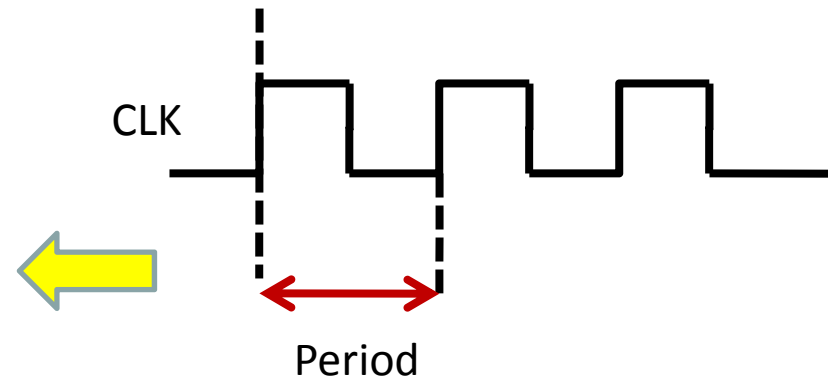
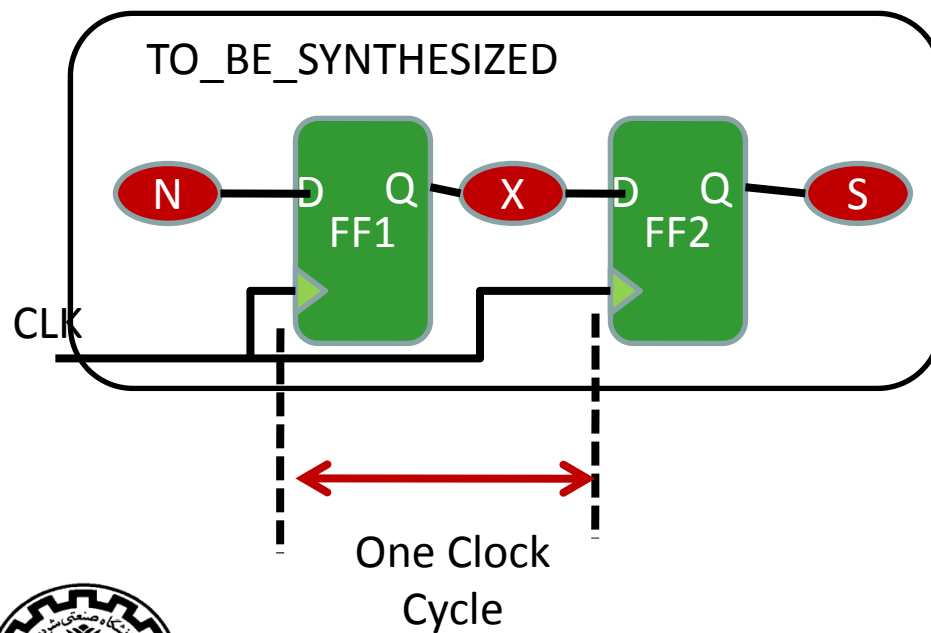
Timing Constraints in DC: Clock Definition

در تعریف کلاک مشخص نمودن
فاکتورهای زیر اختیاری است:

- Duty Cycle
- Skew
- نام کلاک

در تعریف کلاک مشخص نمودن
فاکتورهای زیر الزامی است:

- منبع کلاک (پورت یا پین)
- دوره تناوب کلاک



Timing Constraints in DC: Clock Definition

clk  TO_BE_SYNTESIZED

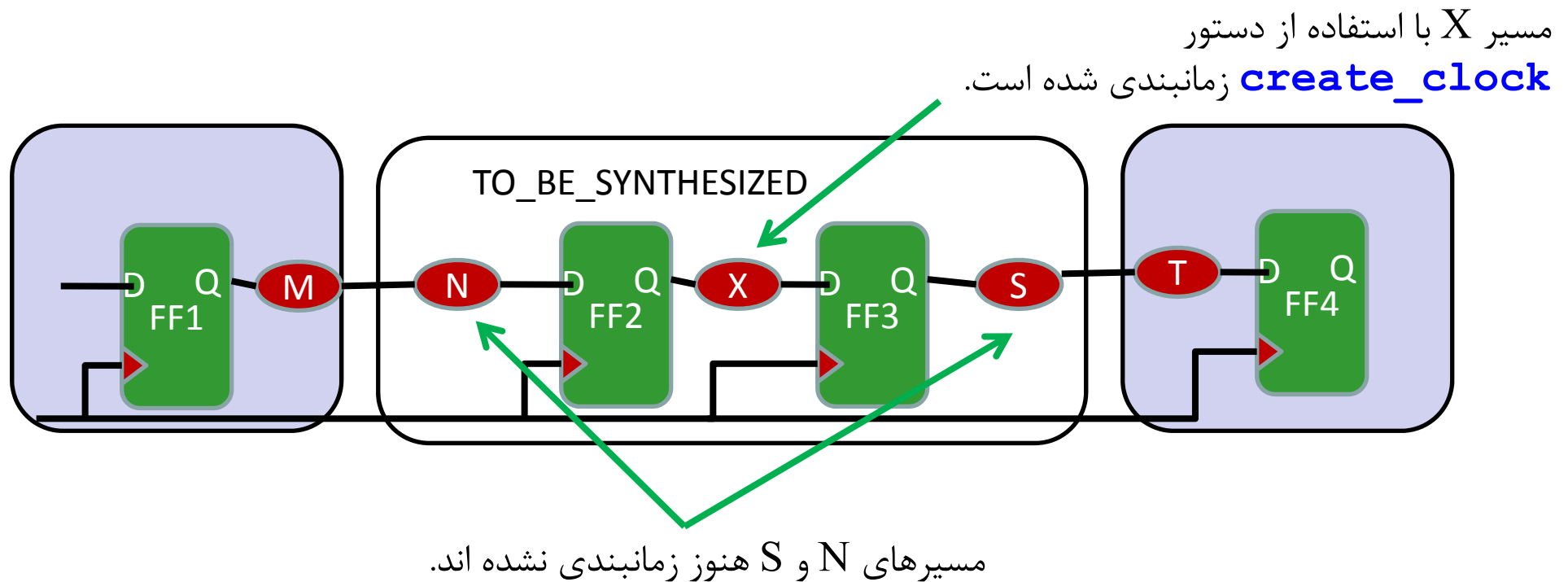
```
dc_shell-t> create_clock -period 10 [get_ports Clk]  
dc_shell-t> set_dont_touch_network [get_clocks Clk]
```

دستور "set_dont_touch_network" به Design Compiler می فهماند که سر راه مسیرهای کلاک بافر قرار ندهد حتی زمانیکه که بار روی فلیپ فلاپها زیاد باشد.



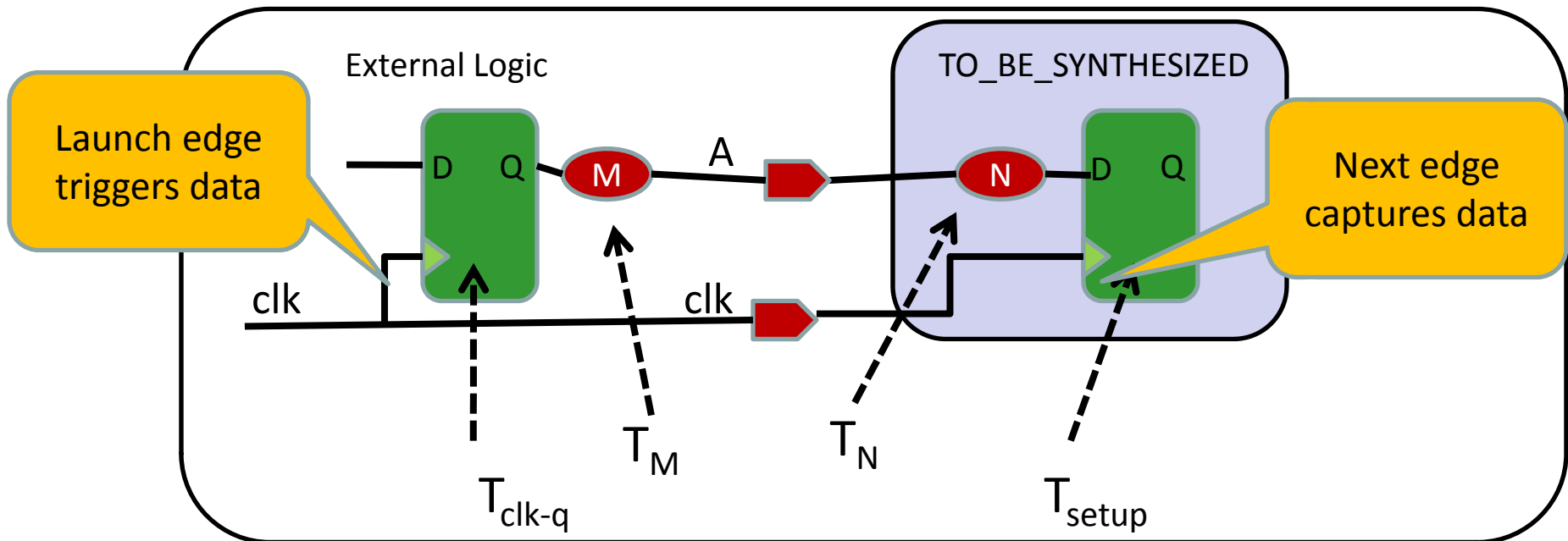
Timing Constraint: Input/Output Timing

- ❑ Paths that are between FFs are timely optimized with `create_clock` command
- ❑ The input/output paths, however, are not timely optimized this way.

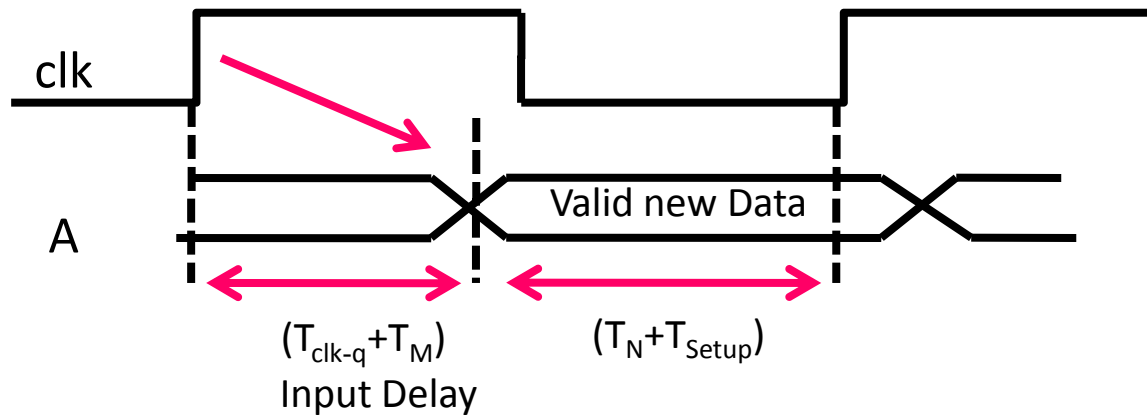


Timing Constraint: Input Path Timings

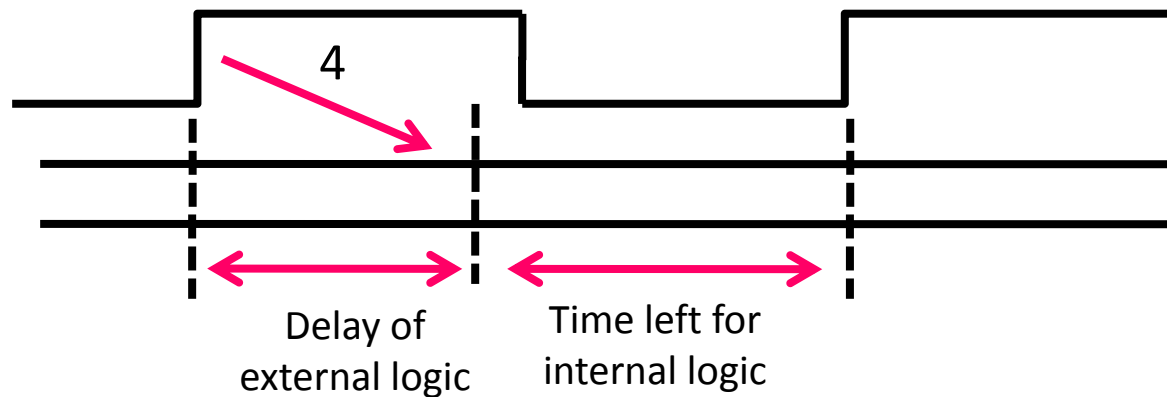
- ❑ To work properly with an external core sitting off the chip connected to the input



Timing Constraint: Input Path Timings

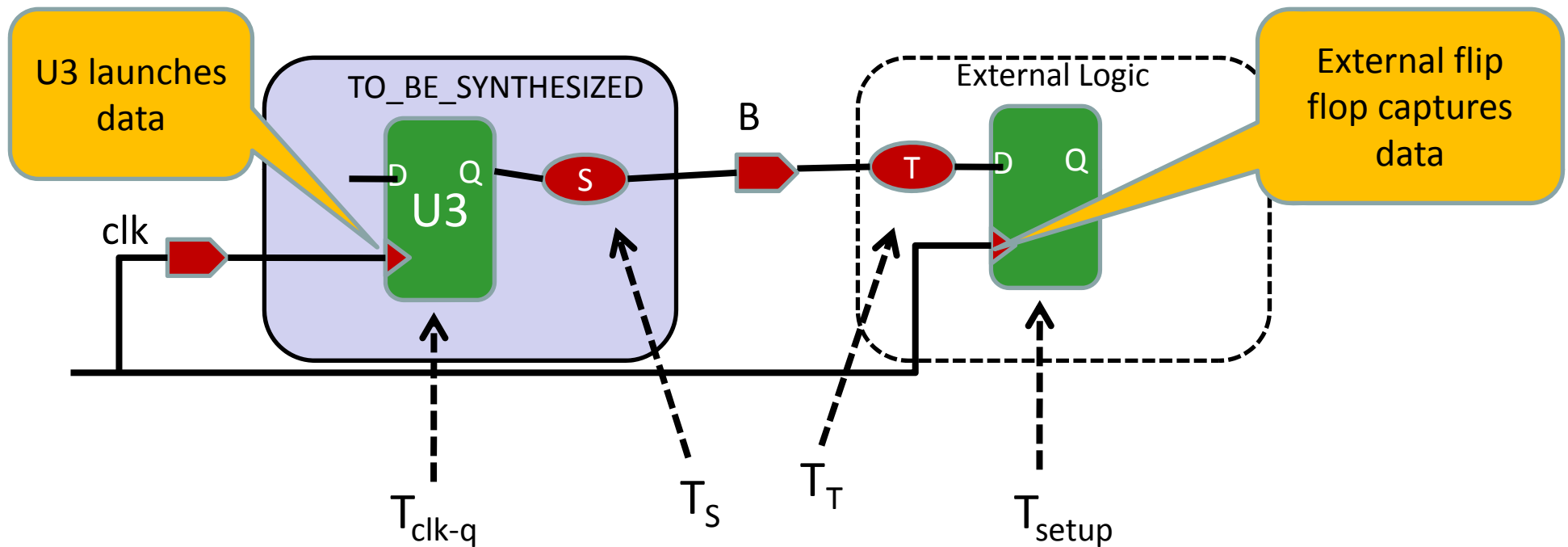


```
dc_shell-t> set_input_delay -max 4 -clock Clk [get_ports A]
```

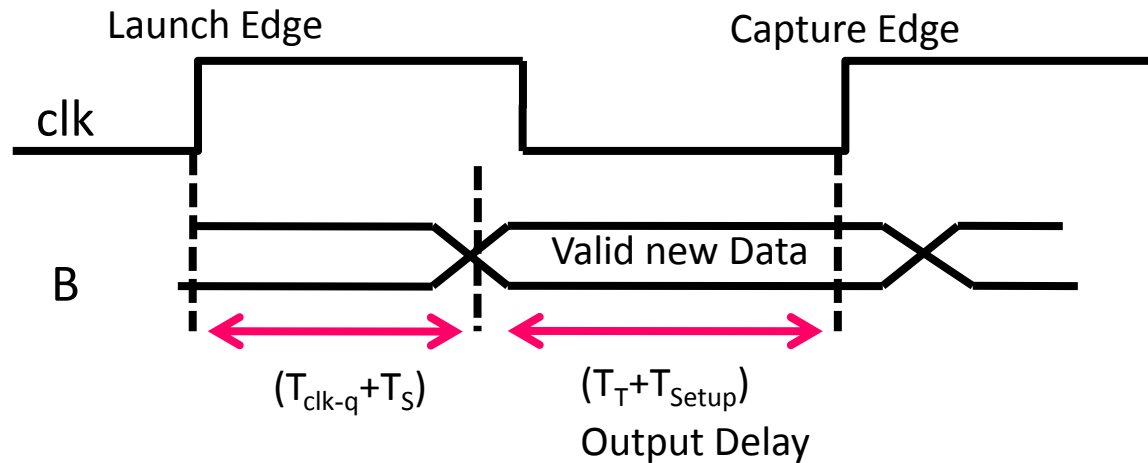


Timing Constraint: Output Path Timings

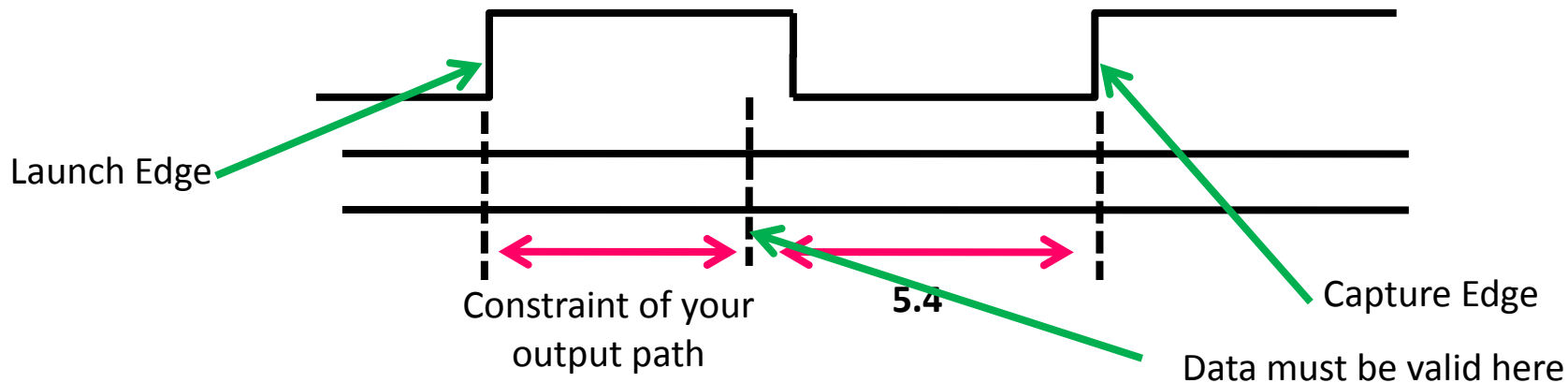
- ❑ To work properly with an external core sitting off the chip connected to the output



Timing Constraint: Output Path Timings



```
dc_shell-t> set_output_delay -max 5.4 -clock Clk [get_ports B]
```

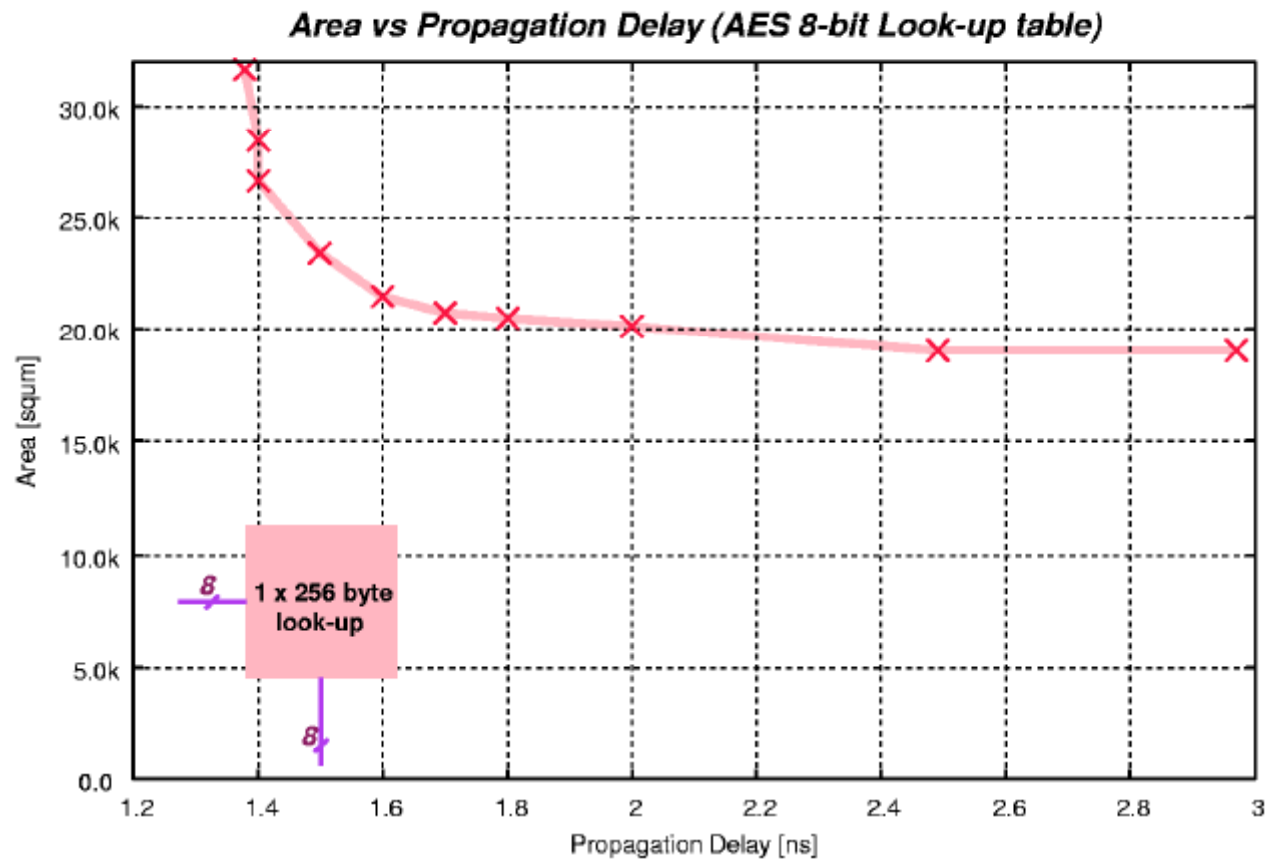


Timing Constraint: Summary

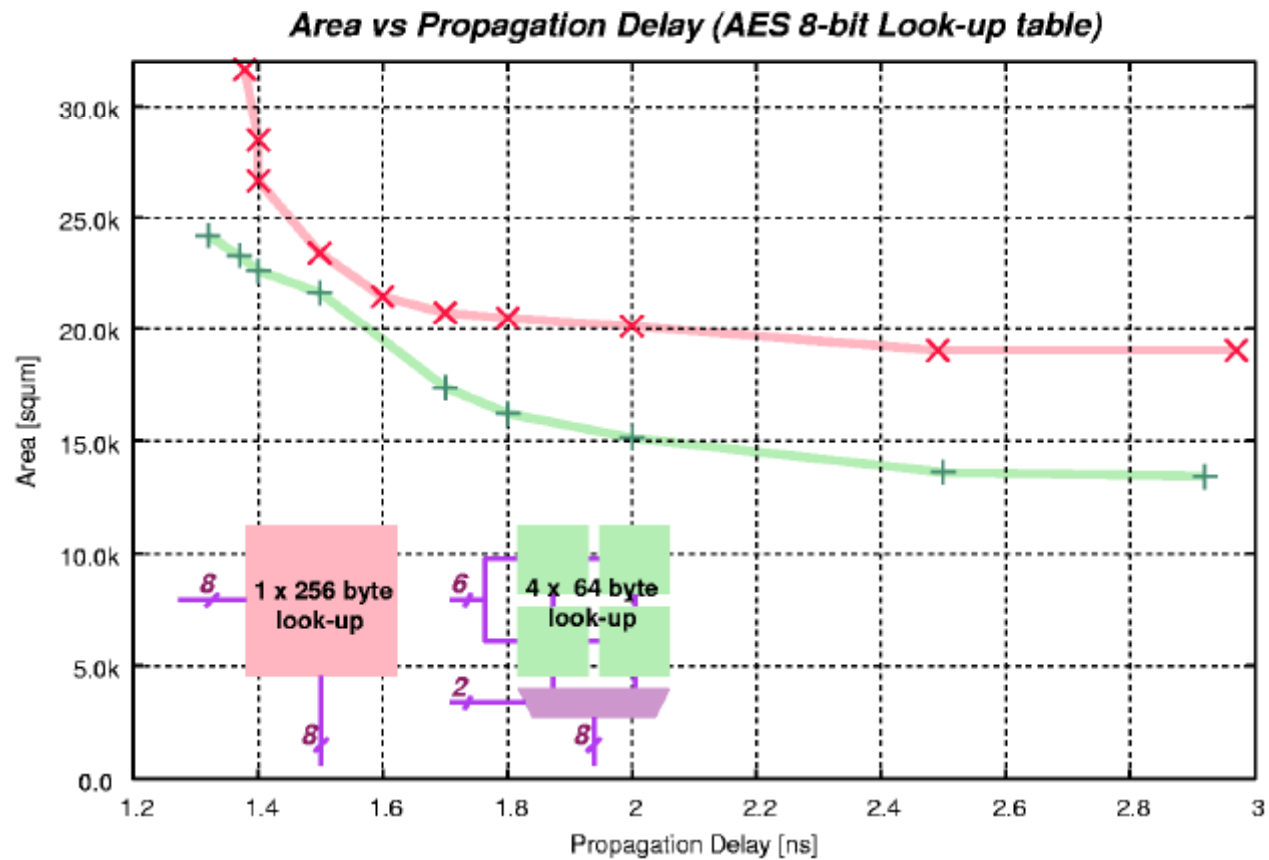
- ❑ Always have a “Time Budget”
- ❑ With the simplified timing assumption:
 - `dc_shell> create_clock “CLK” –period T –waveform { T/2 T } –name cn`
 - Delay of input signals (Clock-to-Q, Package etc.)
`dc_shell> set_input_delay 0 –clock cn all_outputs() – CLK`
 - Don’t forget! `Remove_input_delay [get_ports CLK]`
 - Reserved time for output signals (Holdtime etc.)
`dc_shell> set_output_delay 0 –clock cn all_outputs()`
 - SDC file (`write_sdc`)
 - Later STA & P&R tools need these constraints



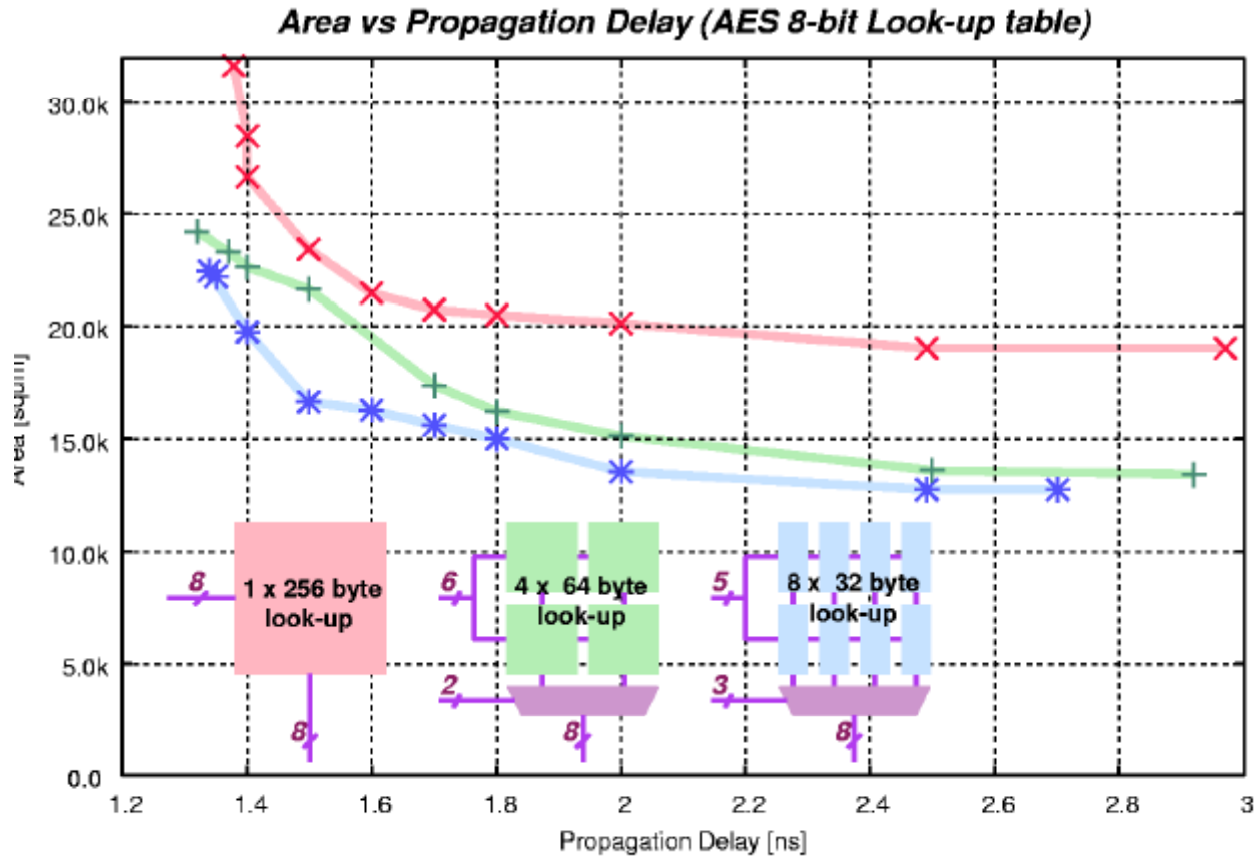
Don't trust the synthesizer too much



Don't trust the synthesizer too much



Don't trust the synthesizer too much



Environmental Attributes

Timing and Area

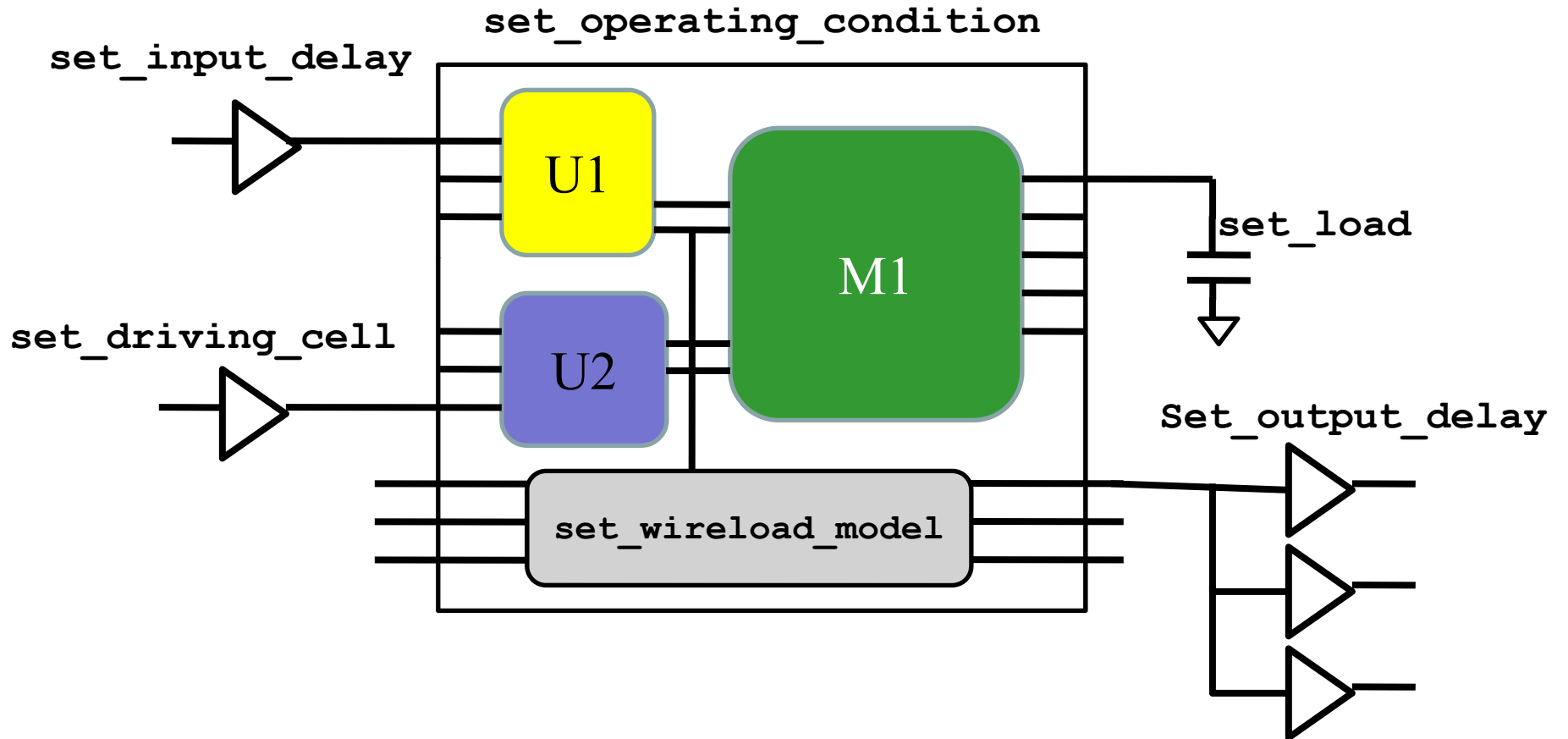
Environmental Attributes

Design Rules and Min Timing

Timing Analysis

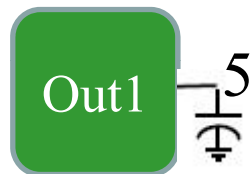


Environmental Attributes



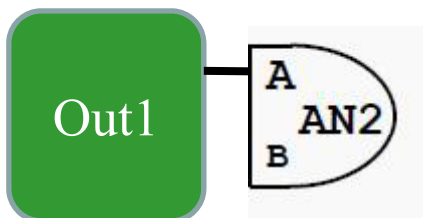
Environmental Attributes

- با توجه به محاسبه دقیق زمانبندی مدار خروجی، Design Compiler نیاز دارد که میزان بار خازنی سلولهای خروجی را بداند.
- دستور `set_load` به شما این امکان را می دهد که یک لود خازنی خارجی به پورتهای ورودی یا خروجی همانند لود پین یک سلول در کتابخانه تکنولوژی اعمال کنید.



```
dc_shell-t> set_load 5 [get_ports OUT1]
```

- دستور `load_of lib/cell/pin` میزان لود یک پین از گیت را از کتابخانه تکنولوژی به یک پورت اعمال می کند.

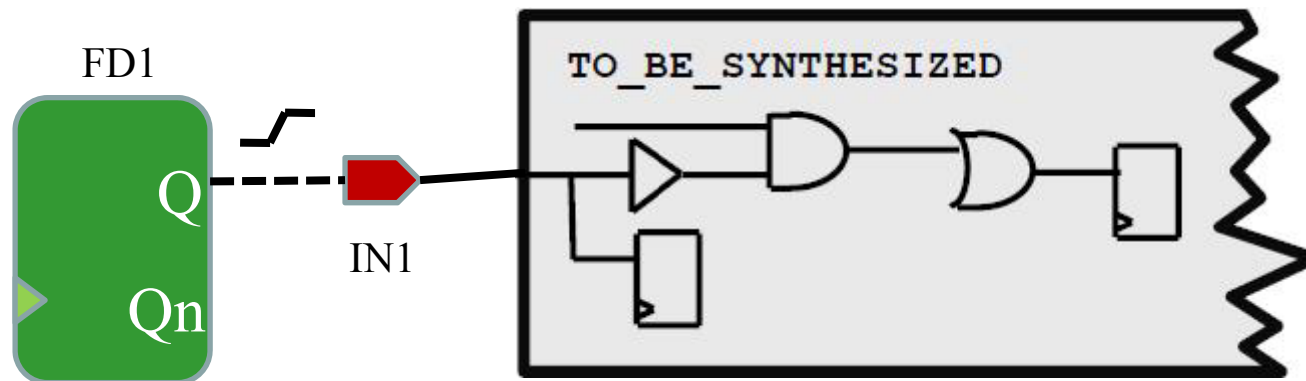


```
dc_shell-t> set_load [load_of my_lib/AN2/A] [get_ports OUT1]
```



Driving Cell at Input

- با توجه به محاسبه دقیق زمانبندی مدار ورودی، Design Compiler نیاز دارد که زمان گذار ورود یک سیگنال به یک پورت را بداند.
- دستور `set_driving_cell` به شما این امکان را می دهد که یک سلول راه انداز خارجی را برای یک پورت ورودی در نظر بگیرد.

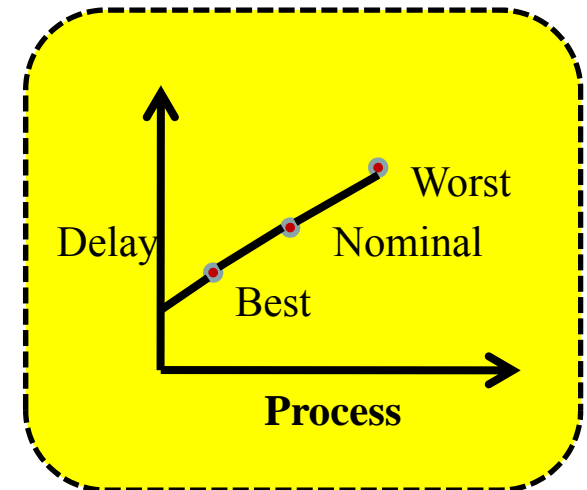
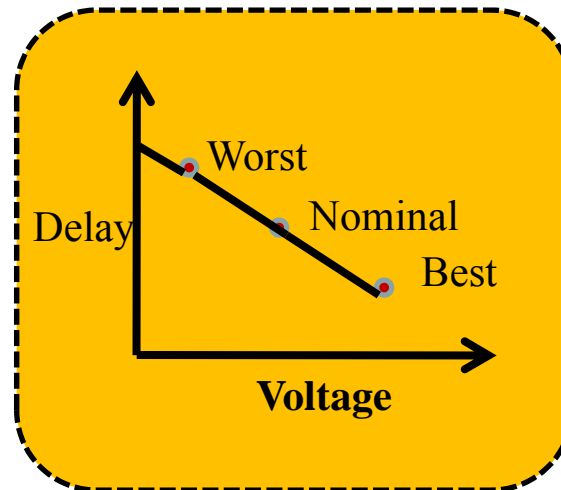
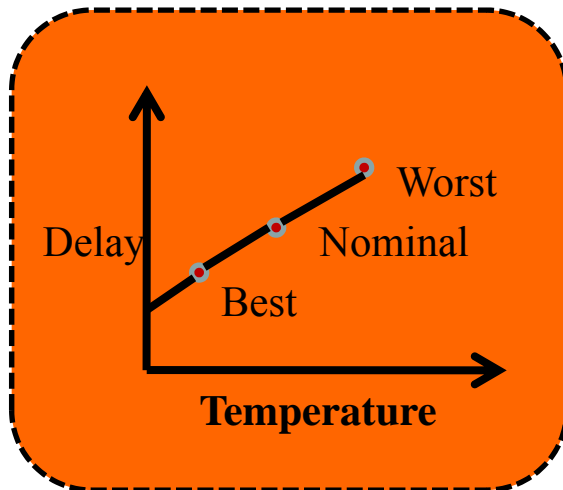


```
dc_shell-t> set_driving_cell max_lib_name -lib_cell FD1 -pin Q [get_ports IN1]
```



Environmental Attributes

- در حین عمل سنتز برای محاسبه تاخیر سیمها و سلولهای بکار رفته در طرح، شرایط خاصی از نظر ۳ ویژگی **دما**، **ولتاژ** و **نوع پروسه** در نظر گرفته می شود. این شرایط در کتابخانه تکنولوژی مشخص می گردد.
- شرایط مختلف عملیاتی با استفاده از دستور **set_operating_conditions** به طرح اعمال می شود.



Environmental Attributes

- با استفاده از دستور `report_lib libname` می توانید لیستی از شرایط عملیاتی را مشاهده کنید.

Operating Conditions:				
Name	Library	Process	Temp	Volt
typical	my_lib	1.00	25.00	1.80
slow	my_lib	1.05	125.00	1.62
fast	my_lib	0.93	0.00	1.98

For Setup Time

For Hold Time

- برای تعیین شرایط عملیاتی:

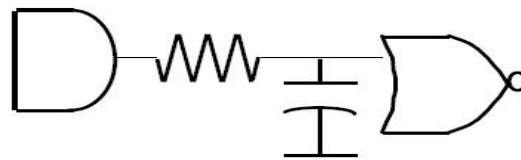
```
set_operation_conditions -min_library lib_name -min condition /  
-max_library lib_name -max condition
```

```
Ex: set_operating_conditions -min_library fast -min fast /  
-max_library slow -max slow
```



Wireload Model

- مدل Wireload تخمینی از RC پارازیتی یک نت بر اساس Fan-out به ما می دهد.
- مدل های Wireload توسط شرکتهای نیمه هادی در دسترس کاربر قرار می گیرد.



```
Name : 160KGATES
Location : ssc_core_slow
Resistance : 0.000271
Capacitance : 0.00017
Area : 0
Slope : 50.3104
```

Fanout	Length
1	31.44
2	81.75
3	132.07
4	182.38
5	232.68

R per unit length

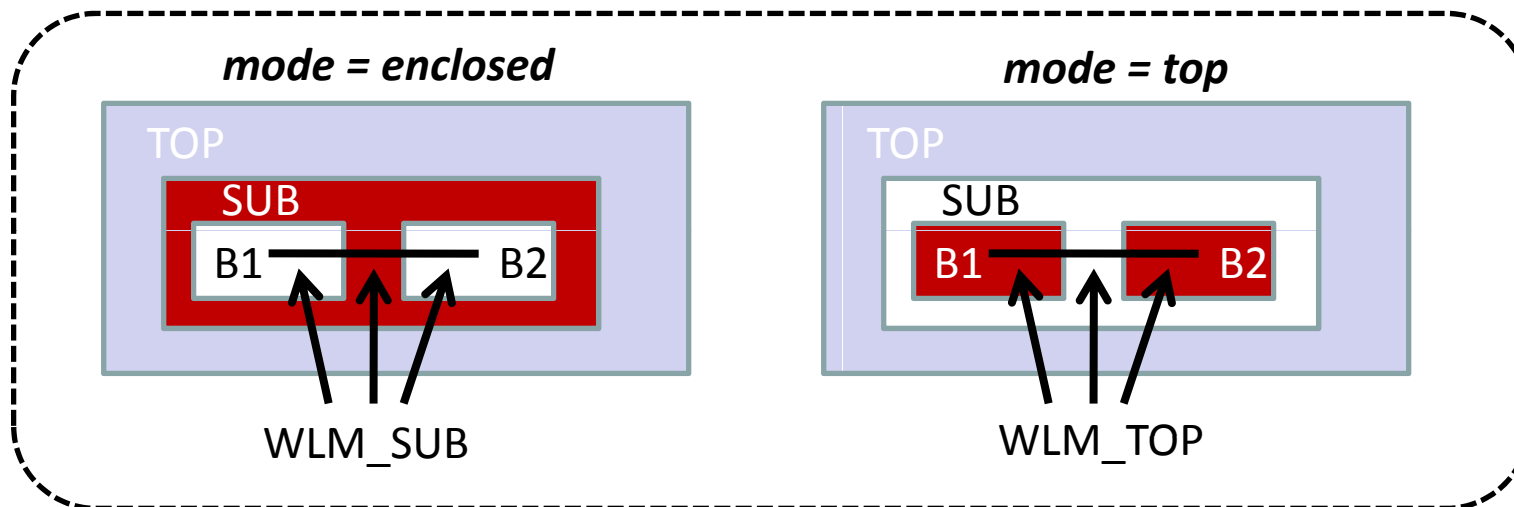
C per unit length

```
Time Unit : 1ns
Capacitive Load Unit : 1.000000pf
Pulling Resistance Unit : 1kilo-ohm
```



Wireload Model in DC

```
current_design my_design  
set_wire_load_model -name 160KGATES
```



در مد enclosed، DC برای یک نت
wireload بلوکی را در نظر می گیرد که نت
را احاطه کرده است (بد بینانه تر)

در مد top، DC تمام طرح را بصورت مسطح
می بیند و wireload مورد نیاز برای تمام
نتهای سطح بالای طرح را در نظر می گیرد

```
dc_shell-t> set_wire_load_mode top
```



Design Rules and Min Timing

Timing and Area

Environmental Attributes

Design Rules and Min Timing

Timing Analysis



Design Rules in DC

- ❑ شرکت‌های نیمه هادی قواعد طراحی را در نظر می‌گیرند تا محدودیتهایی برای سلول‌های متصل به هم از نظر بار خازنی، زمان گذار و Fanout ایجاد کنند.
- ❑ هنگامیکه که خطایی در حین فرایند بهینه سازی صورت می‌گیرد کامپایلر تلاش می‌کند تا طرح را به محدودیتهای هدف برساند.

مقدار مشخصی را برای بیشینه بار خازنی یک پورت یا یک نت در نظر می‌گیرد.

`max_capacitance`

زمان گذار یک نت، زمانی است که پین راه انداز آن نیاز دارد تا منطق سیگنال روی نت را عوض کند.

`max_transition`

- ❑ مهمترین محدودیتهای قواعد طراحی

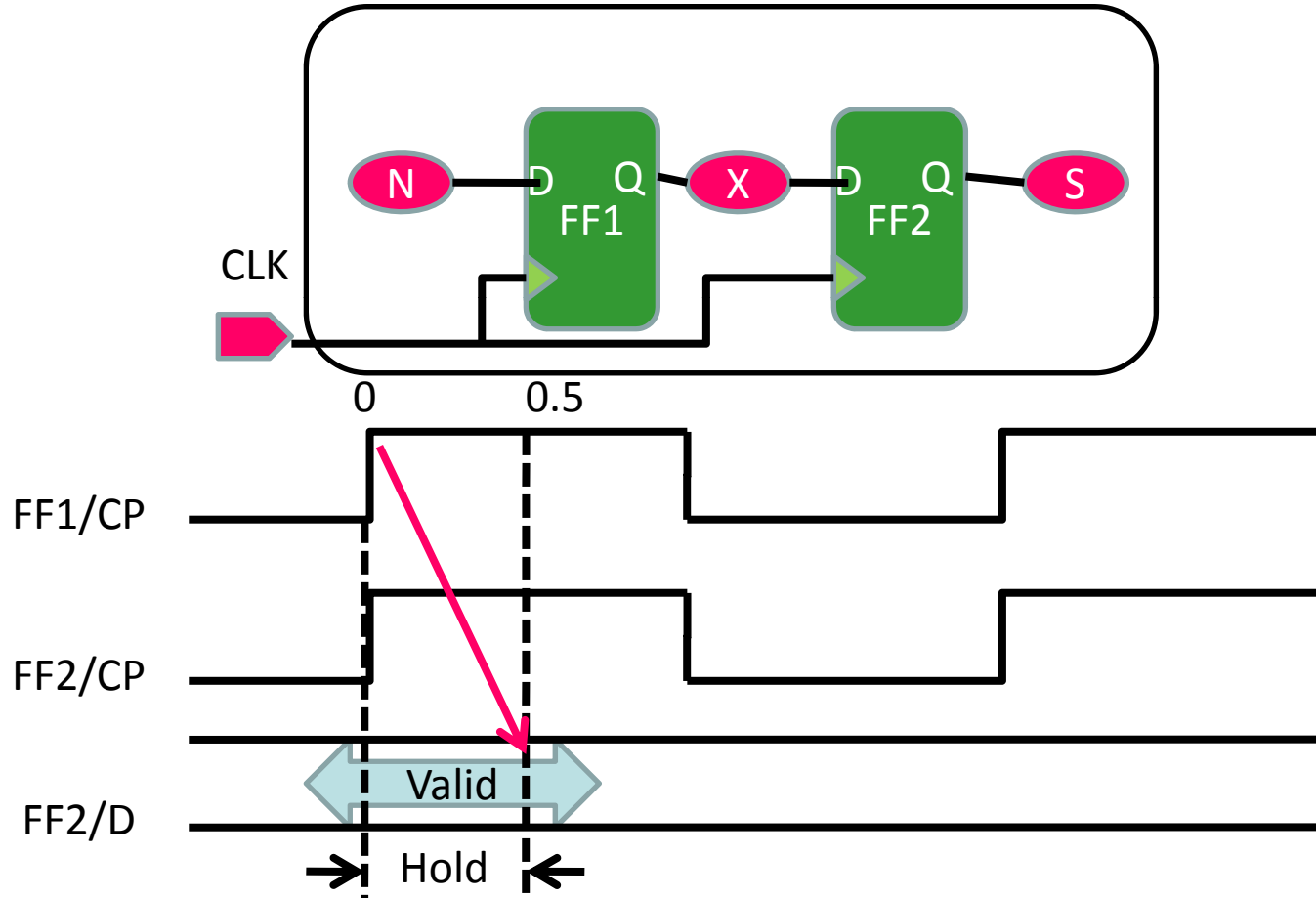
`max_fanout`

بیشینه fanout برای پورتهای ورودی یا برای تمام نتهای یک طرح را تنظیم میکند.



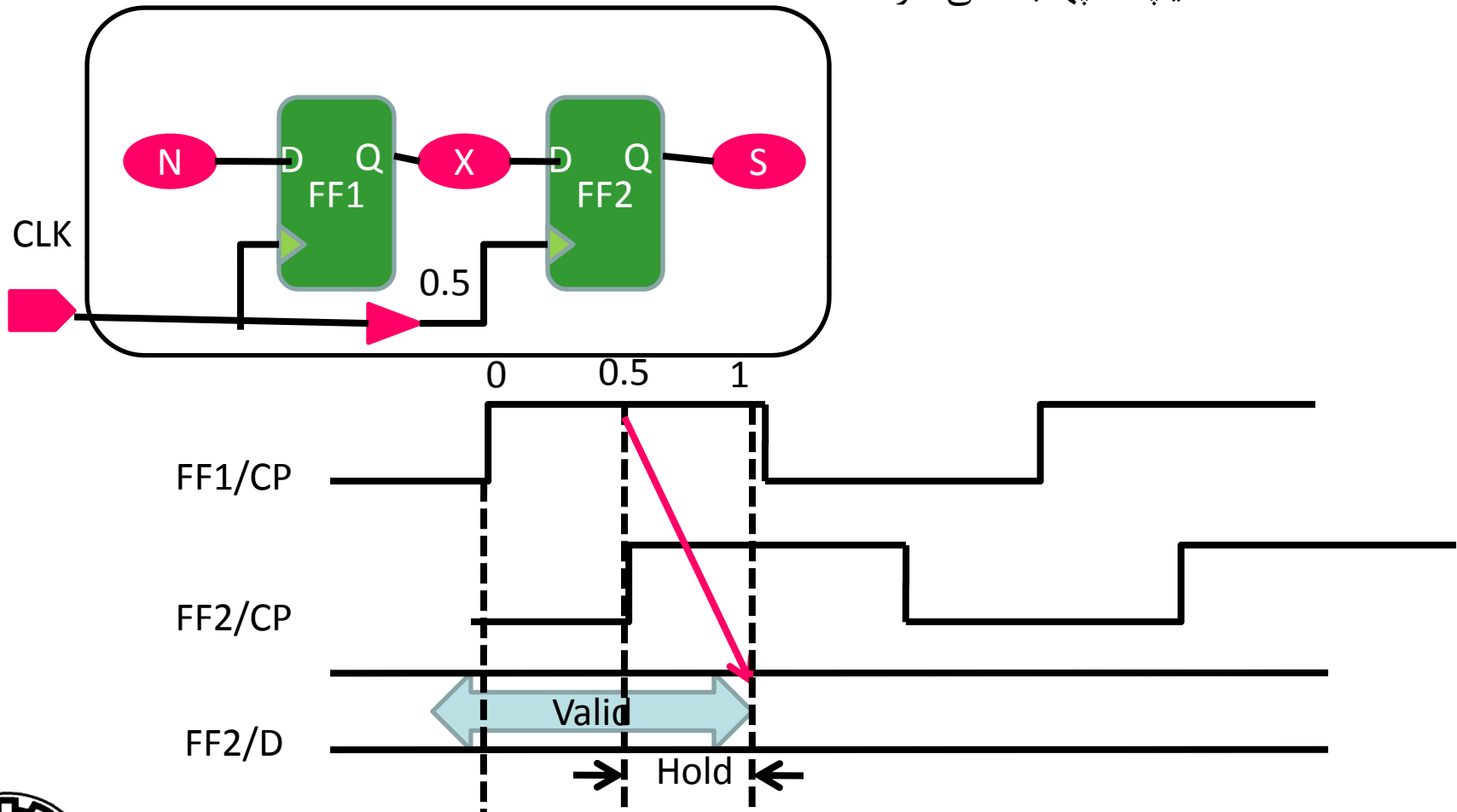
Hold Time Violation

□ Hold time مینیمم زمانی است که اطلاعات از FF1 به اندازه آن زمان باید منتظر ورود به FF2 بماند.



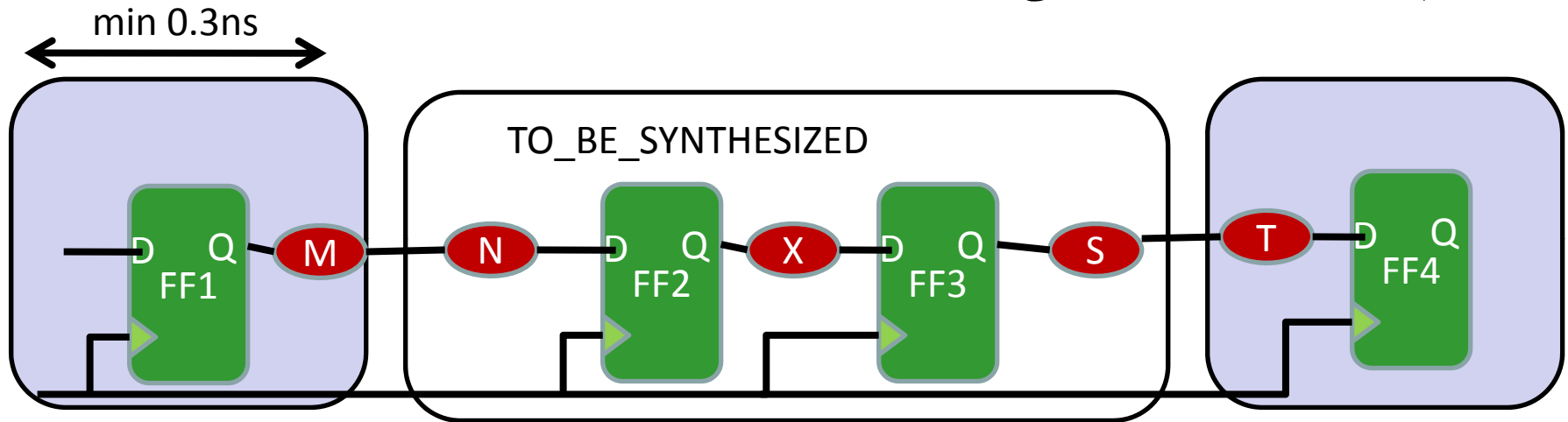
Hold Time Violation

□ میزان Hold time یک طرح به Clock Skew موجود بر روی شاخه های کلاک، شرایط کاری و Hold time فلیپ فلاپها بستگی دارد.



Hold Time Violation: Solution

□ دستور `set_input_delay -min` سریعترین زمان رسیدن یک سیگنال به یک پورت ورودی را در نظر می گیرد.



```
dc_shell-t> create_clock -period 10 [get_ports Clk]
dc_shell-t> set_input_delay -min 0.3 -clock Clk $all_in_ex_clk
```

اگر Hold Time فلیپ فلاپ FF2 برابر 1ns باشد، مسیر N حداقل به 0.7ns زمان نیاز دارد.



Timing Analysis

Timing and Area

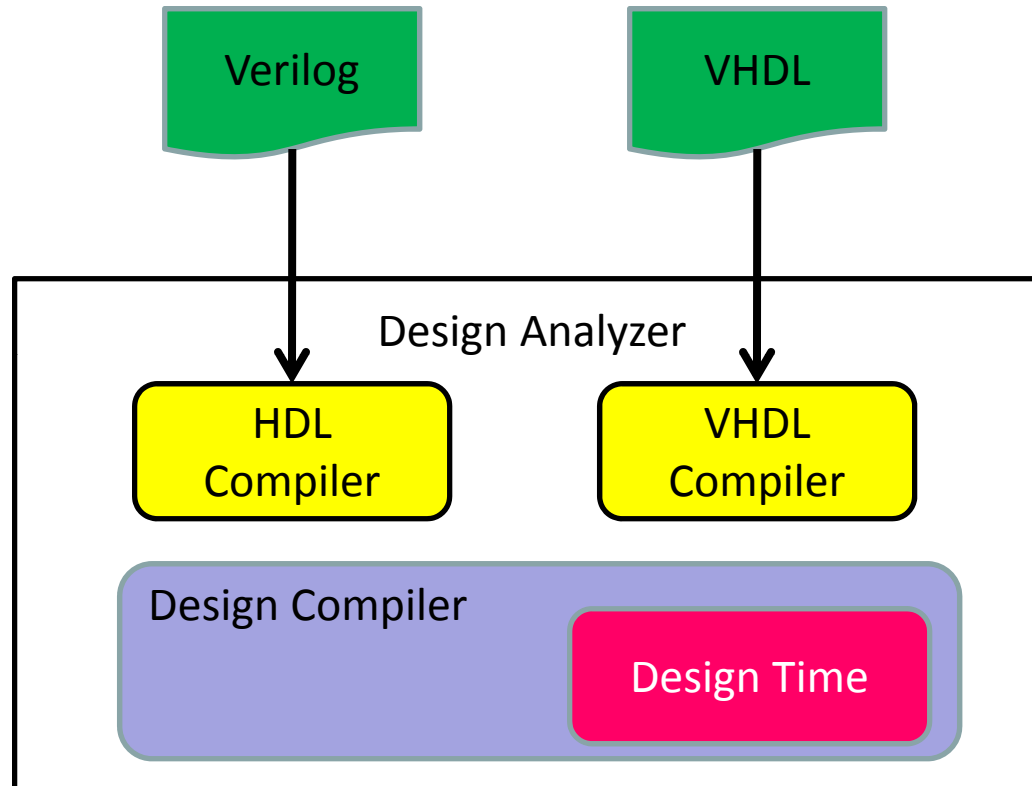
Environmental Attributes

Design Rules and Min Timing

Timing Analysis

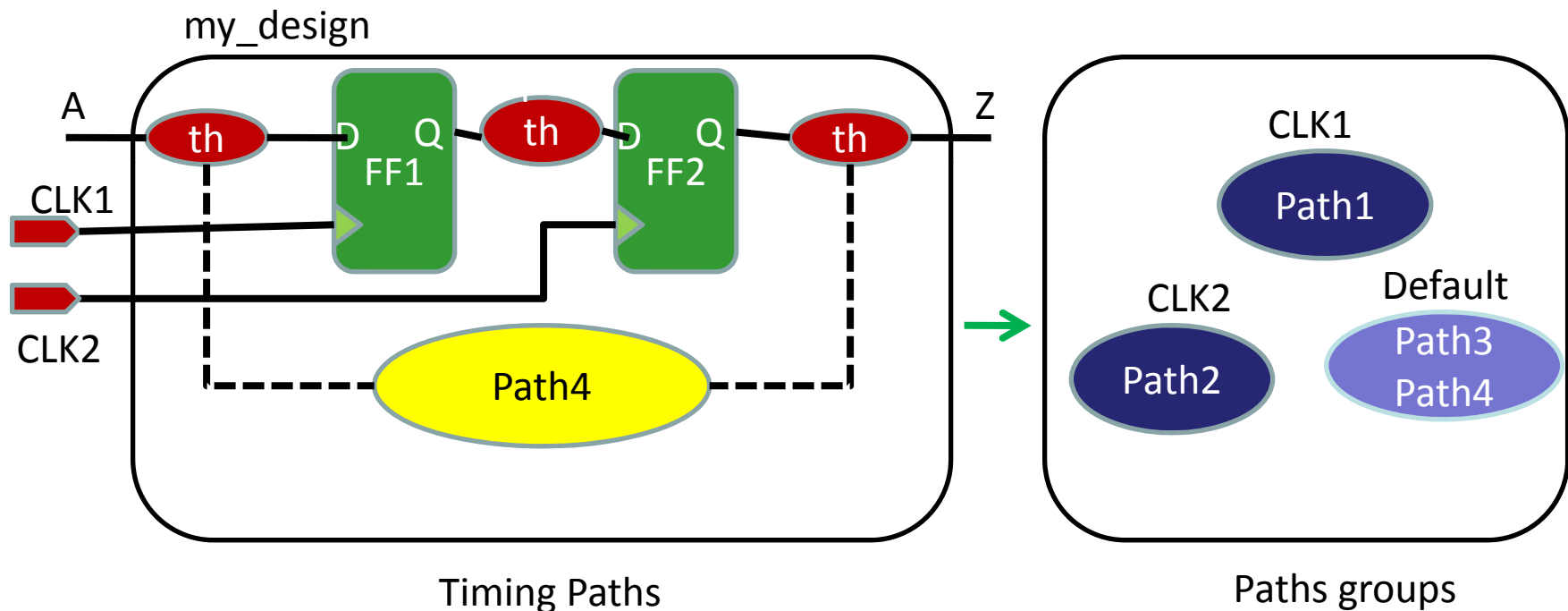


Timing Analysis



Timing Analysis

□ تحلیل زمانی استاتیک مشخص می کند که آیا مدار می تواند محدودیتهای اعمال شده به طرح را از نظر زمانی برآورد کند بدون اینکه شبیه سازی دینامیک بر روی آن انجام شود.



Timing Analysis

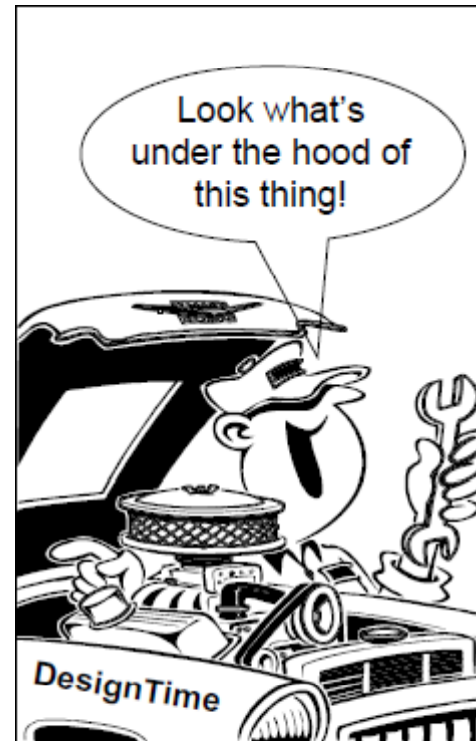
مدل تاخیر سلولها

مدل Wireload

مدل اتصالات داخلی

شرایط عملیاتی

المانهایی که در محاسبه تاخیر مسیرها استفاده می شوند



Compilation

```
dc_shell> compile
```



Change the Effort Level

```
dc_shell-t> compile -map_effort (low | medium | high)
```

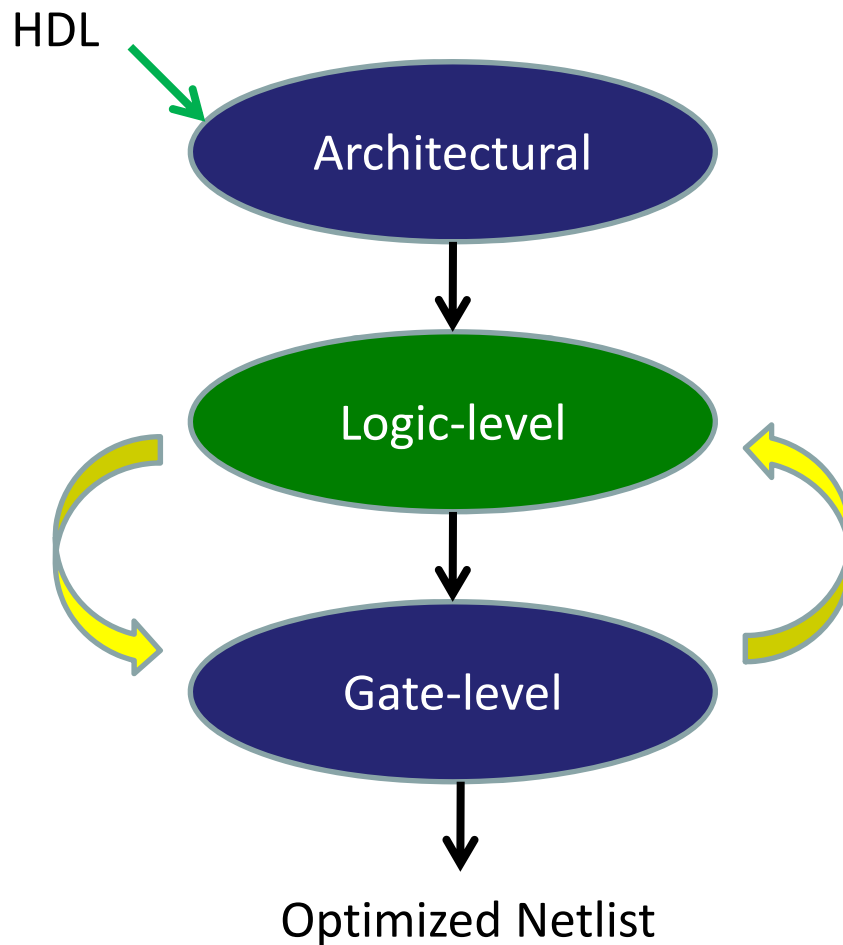
برای بدست آوردن نتایج جدی
از آن استفاده نمی شود.

حالت پیش فرض کامپایل است
و ممکن است نتایج خوبی به ما
بدهد.

از الگوریتمهای پیچیده تری
برای سنتز استفاده می کند.



Compilation



□ تمام طرح به المانهای GTECH تبدیل می شود.

□ ممکن است در یک اجرای کامپایل نتایج مطلوبی حاصل نشود مگر اینکه محدودیتهای طرح اصلاح شده و یا میزان تلاش کامپایل تغییر یابد.



Compilation Report

Beginning Delay Optimization Phase

ELAPSED TIME	AREA	WORST NEG SLACK	TOTAL NEG SLACK	DESIGN RULE COST	ENDPOINT
0:10:04	2761.7	1.38	3.20	18.1	Zro_Flag_reg/D
0:10:05	2761.7	1.38	3.20	18.1	Zro_Flag_reg/D
0:10:08	2761.7	1.28	3.10	18.1	Zro_Flag_reg/D

Critical Path
timing violations

Sum of all timing
violations



Compilation Report: Slack

□ **Arrival Time**: برای یک مسیر، $A(P)$ زمانی است که طول می کشد تا سیگنال از نقطه شروع یک مسیر به نقطه پایان برسد.

□ **Require Time**: $R(P)$ ماکزیمم زمانی است که یک سیگنال برای پیمایش طول مسیر در اختیار دارد.

□ **Slack**: با استفاده از **Arrival Time** و **Require Time** می توان **Slack** را محاسبه نمود.

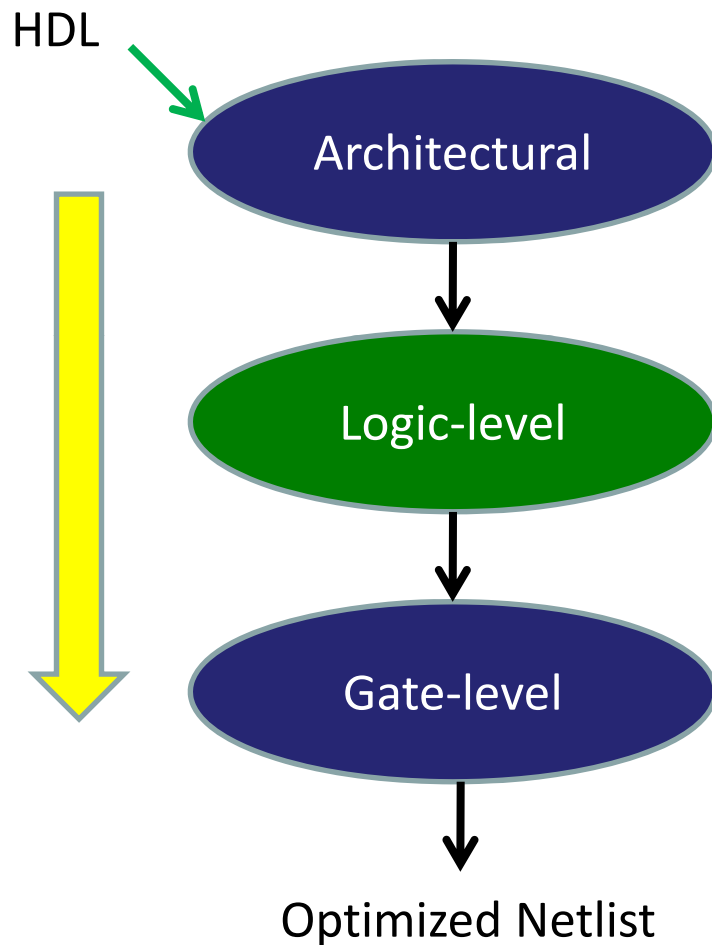
$$\text{Slack} = R(P) - A(P)$$

- میزان **Slack** بر بحرانی بودن یک نود تاثیر گذار است.
- **Positive Slack**: نود در مسیر بحرانی قرار ندارد و طرح به محدودیتهای زمانی خود می رسد.
- **Zero Slack**: نود در مسیر بحرانی است ولی به سختی طرح به محدودیتهای زمانی خود می رسد.
- **Negative Slack**: خطای زمانی وجود دارد.

بهینه سازی **Slack** یعنی بهینه سازی طرح از نظر زمانی



Compilation



□ در صورت بروز خطا بهینه سازی باید در تمام سطوح انجام شود.

□ اعمال تغییرات در کد HDL را نیز مد نظر داشته باشید.



Compilation Report

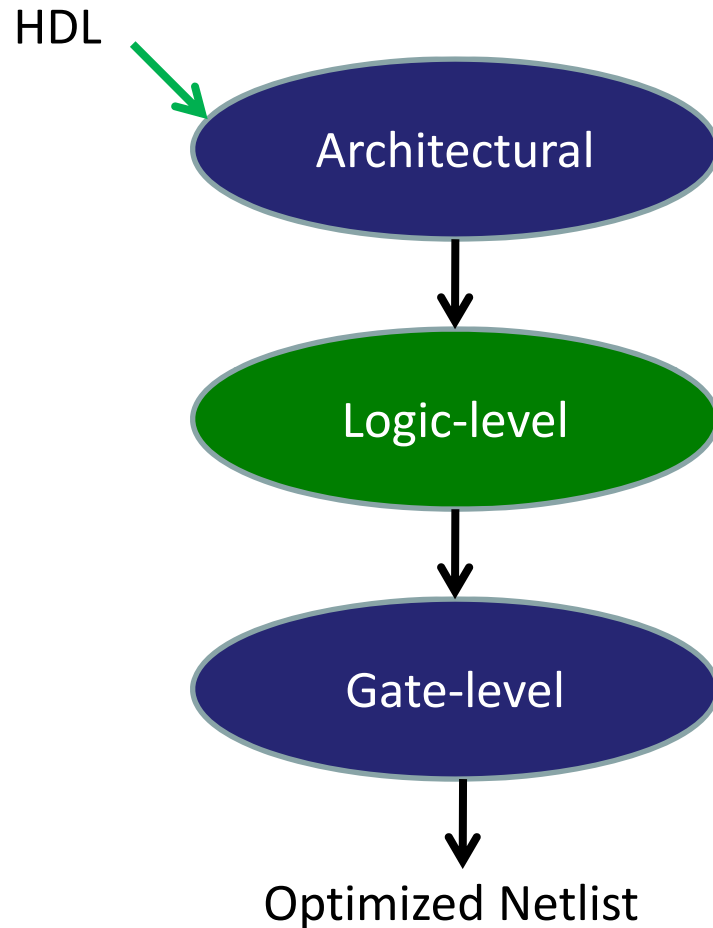
```
dc_shell-t> report_constraint -all
Information: Updating design information... (UID-85)
*****
Report : constraint
-all_violators
Design : RISC_CORE
Version: 2002.05
Date : Wed Jul 3 09:38:42 2002
*****
max_delay/setup ('Clk' group)
```

Endpoint	Required Path Delay	Actual Path Delay	Slack
RESULT_DATA[1]	1.20	1.30 r	-0.10 (VIOLATED)
RESULT_DATA[2]	1.20	1.26 r	-0.06 (VIOLATED)
RESULT_DATA[8]	1.20	1.26 r	-0.06 (VIOLATED)
RESULT_DATA[14]	1.20	1.22 r	-0.02 (VIOLATED)
RESULT_DATA[5]	1.20	1.22 r	-0.02 (VIOLATED)
RESULT_DATA[11]	1.20	1.22 r	-0.02 (VIOLATED)

فرض کنید محدودیتهای اعمال شده به طرح و جزءبندی به درستی انجام شده باشد. در این صورت با وجود خطا چه باید کرد؟



Incremental Compilation



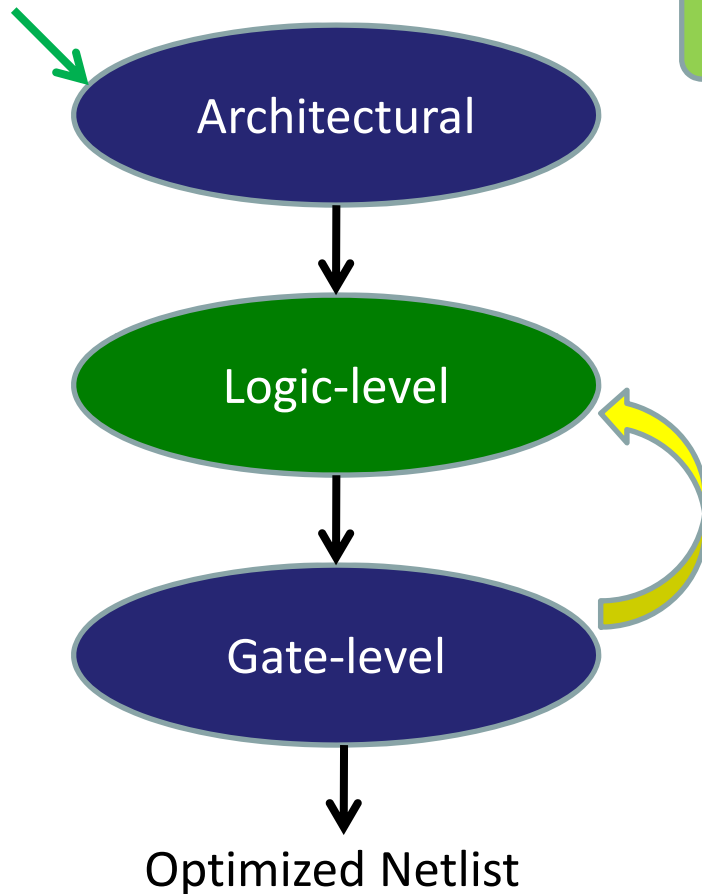
```
dc_shell-t> compile -incremental_mapping
```

- فقط بهینه سازی در سطح گیت انجام می شود. به همین دلیل طرح دوباره به المانهای GTECH تبدیل نخواهد شد.
- کامپایل افزایشی از کامپایل معمولی سریعتر است.



Incremental Compilation

HDL



```
dc_shell-t> compile -inc -map high
```

□ الگوریتم بهینه سازی فقط راهکارهایی را اعمال می کند که Slack مسیرهای بحرانی را کاهش می دهد.

□ کامپایل افزایشی می تواند نتایج بهتری را در بر داشته باشد و یا بهبودی در نتایج حاصل نشود.



Compilation

□ Usually, we have to perform 2 or 3 compiles

➤ **1st compilation** Rough compilation (timing only)

```
dc_shell> compile -map_effort medium
```

➤ **2nd compilation** Refine circuit area and timing

```
dc_shell> set_ultra_optimization true
```

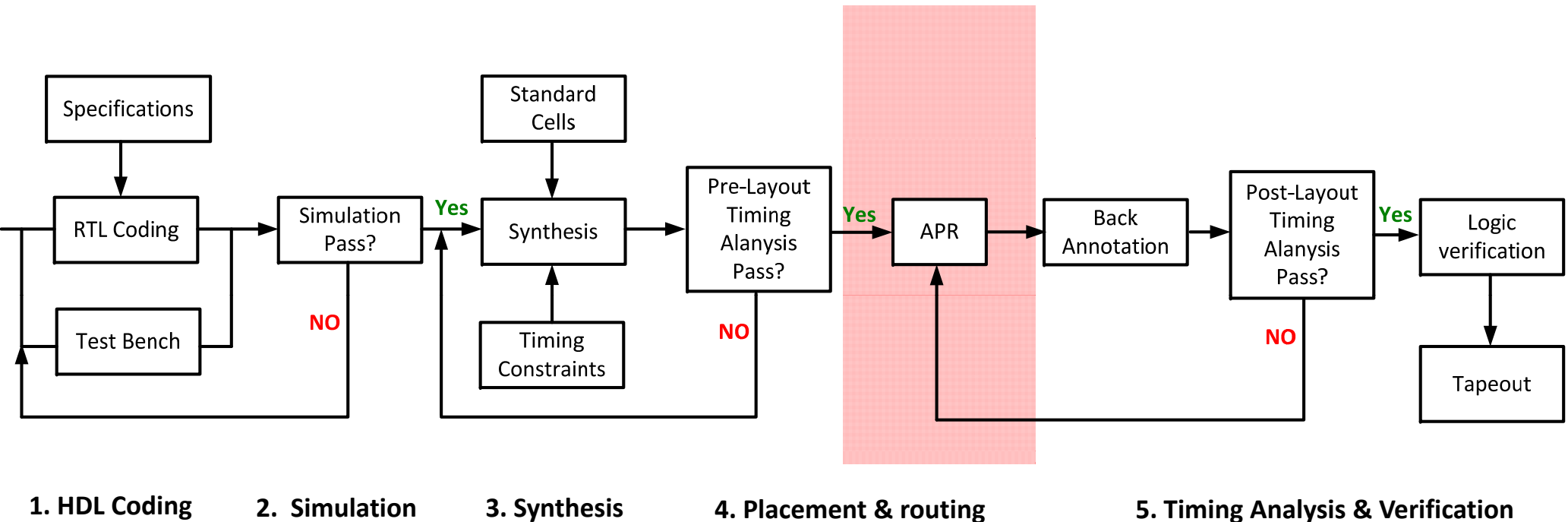
```
dc_shell> set_ultra_optimization -force
```

```
dc_shell> compile -map_effort high -incremental_map
```

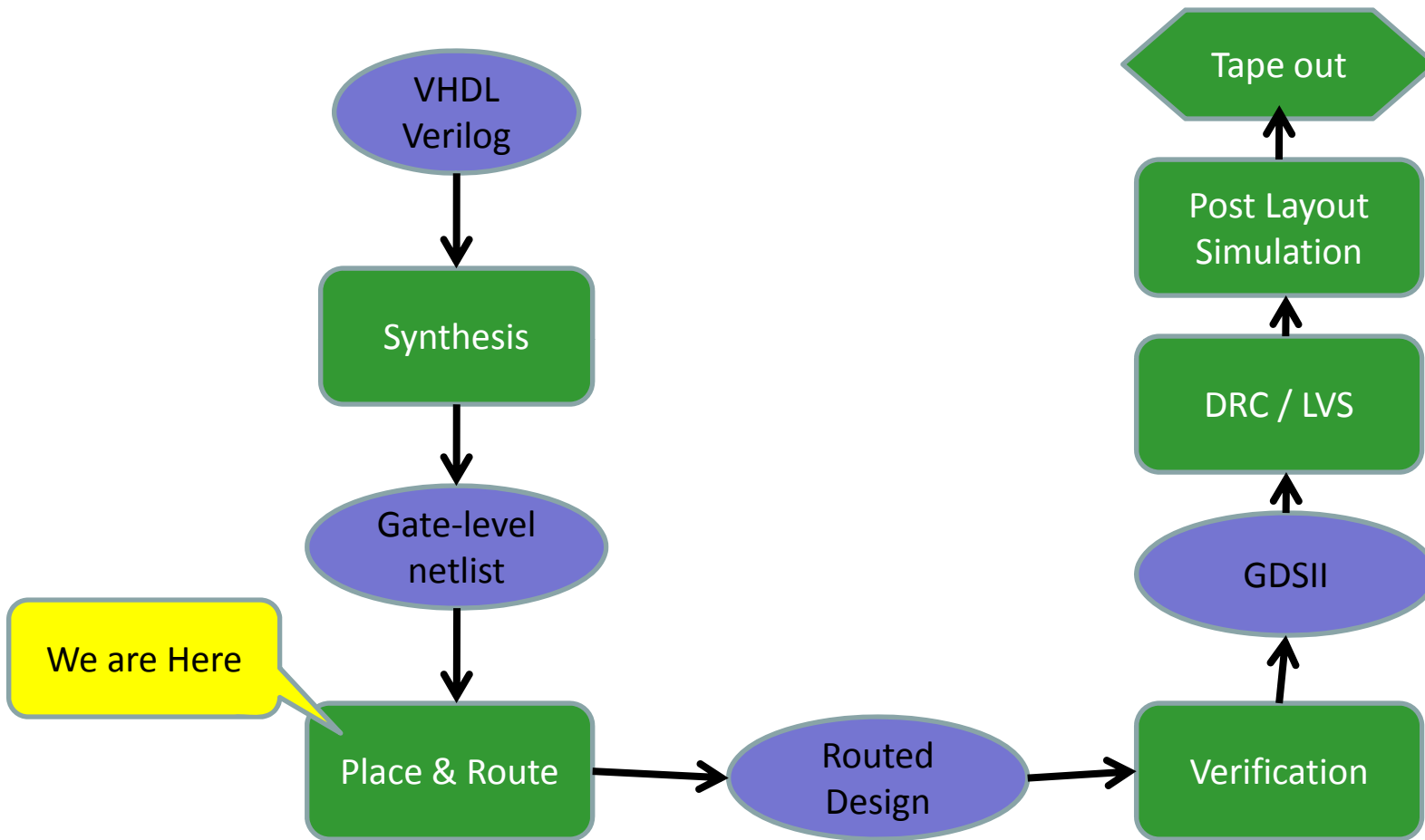
➤ **3rd compilation** Optimize power



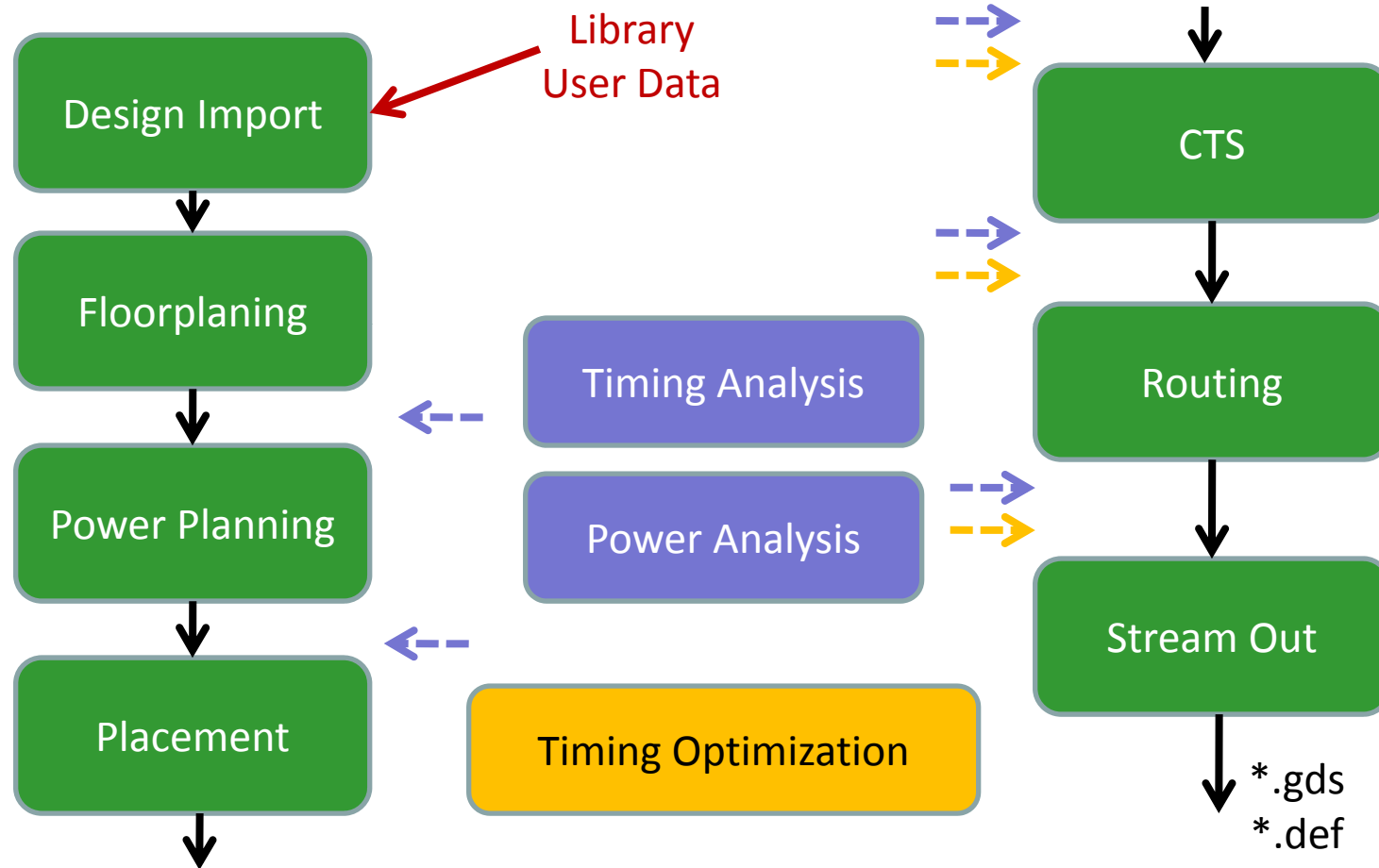
ASIC Design Flow: Revisited (Placement & Routing)



ASIC Design Flow: Revisited

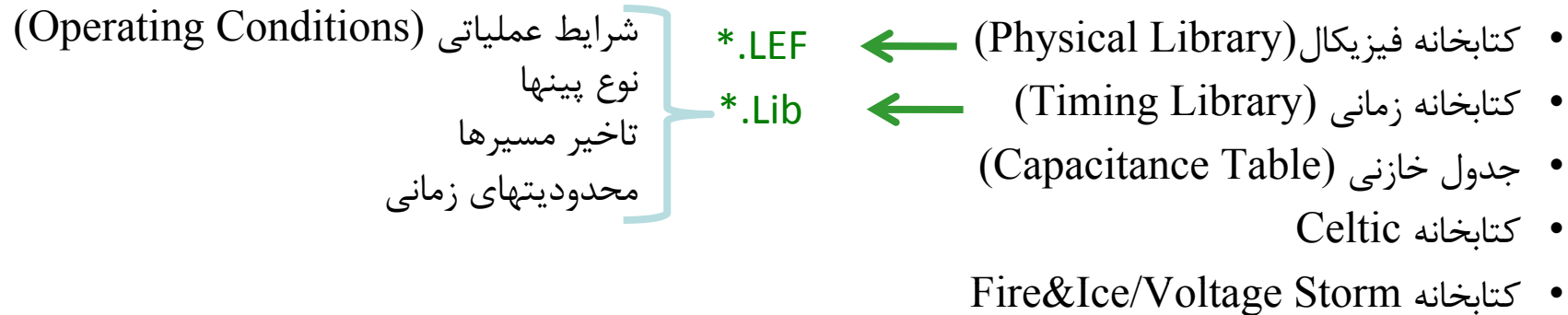


Placement & Routing Steps (SOC Encounter)

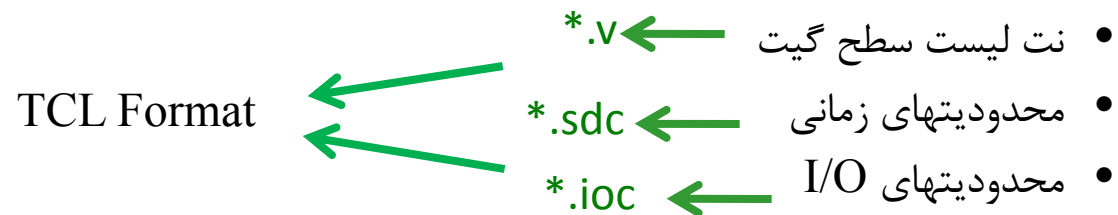


Placement & Routing: Setup

کتابخانه ها



اطلاعات کاربر



We use SOC Encounter from Cadence for P & R



Placement & Routing: Setup (LEF Files)

Layers

 POLY

 Contact

 Metal1

 Via1

 Metal2

Design Rule

- Net width
- Net spacing
- Area
- Enclosure
- Wide metal
- Slot
- Antenna
- Current Density

Parasitics

- Resistance
- Capacitance

Physical Rules

- Unit
- Site
- Routing pitch
- Default Direction
- Via generate
- Via stack



Placement & Routing: Setup (LEF Files)

Physical Rules

- Unit
- Site
- Routing pitch
- Default Direction
- Via generate
- Via stack

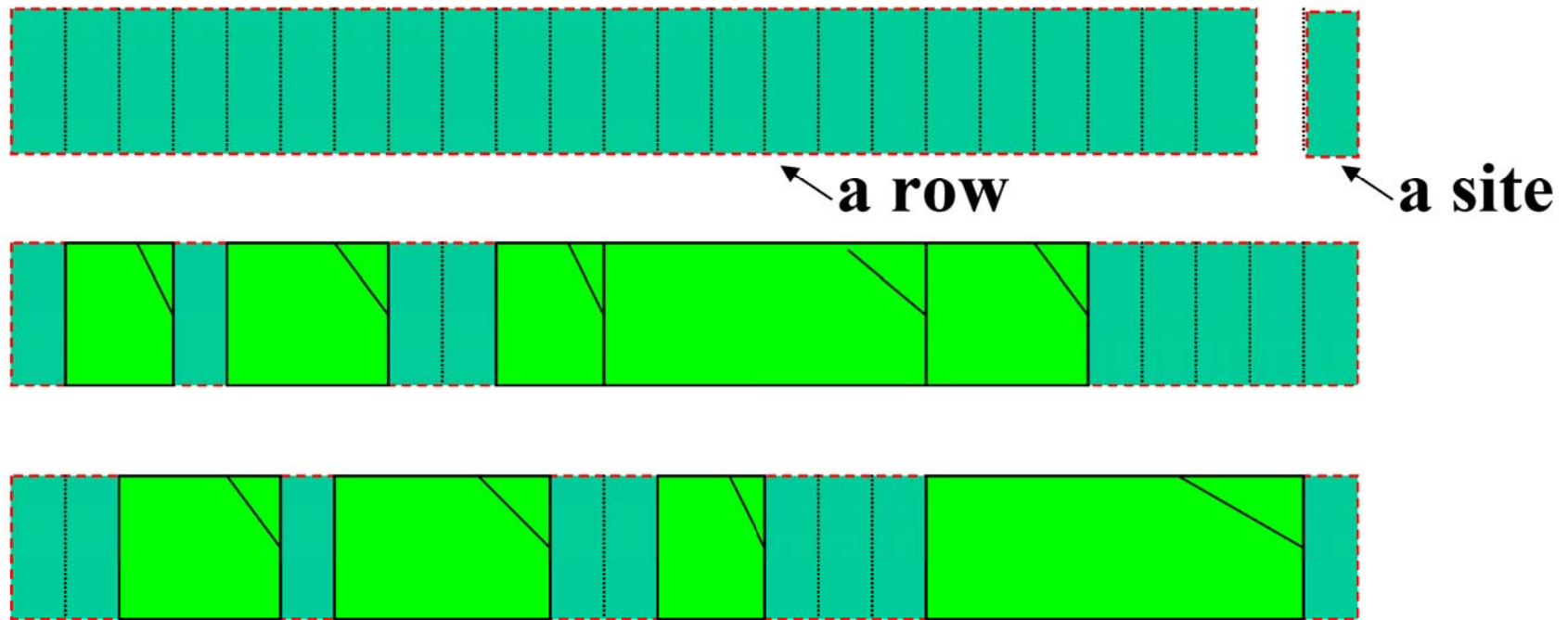
Physical Macros

- Define physical data for
 - Standard cells
 - I/O pads
 - Memories
 - other hard macros
- Describe abstract shape
 - Size
 - Class
 - Pin
 - Obstruction



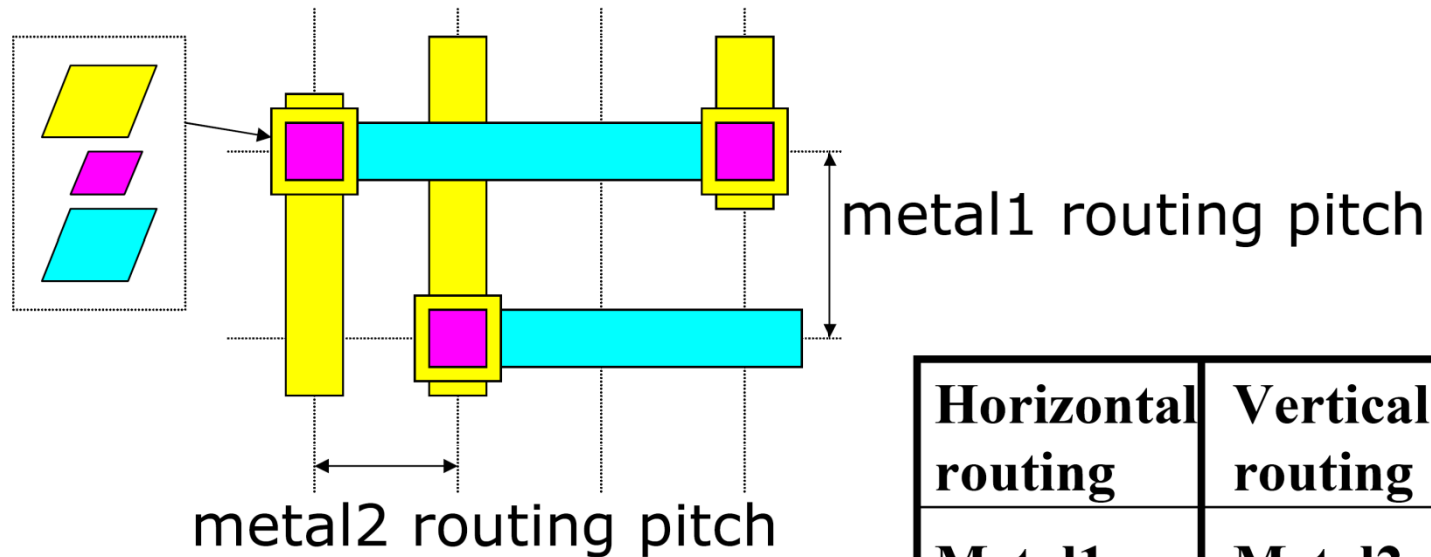
Placement & Routing: Setup (LEF Files)

□ Unit & Site



Placement & Routing: Setup (LEF Files)

Routing Pitch & Default Direction



Horizontal routing	Vertical routing
Metal1	Metal2
Metal3	Metal4
Metal5	Metal6

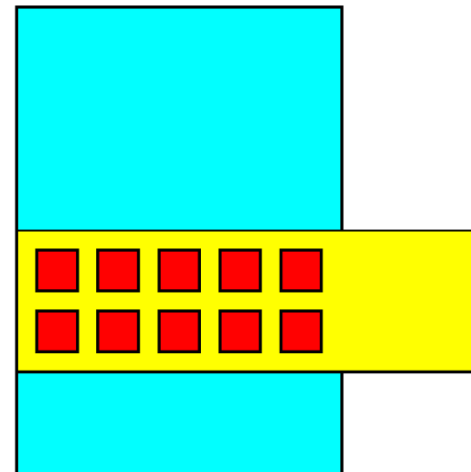


Placement & Routing: Setup (LEF Files)

□ Via Generate:

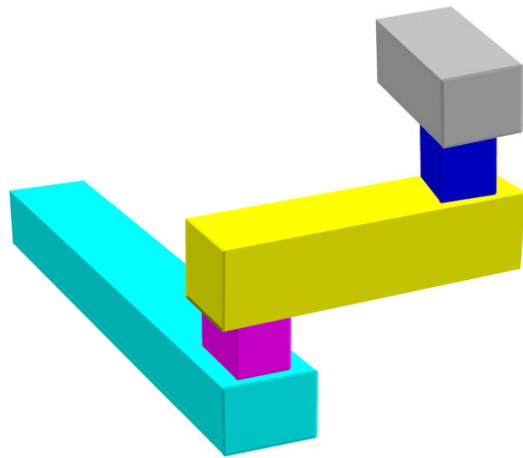
- To connect wide metal , create a via array to reduce via resistance
- Defines formulas for generating via arrays

```
Layer Metal1
  Direction HORIZONTAL
  OVERHANG 0.2
Layer Metal2
  Direction VERTICAL
  OVERHANG 0.2
Layer Via1
  RECT -0.14 -0.14 0.14 0.14
  SPACING 0.56 BY 0.56
```

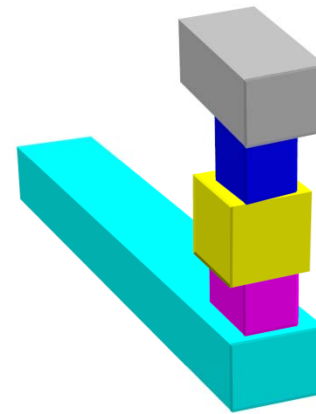


Placement & Routing: Setup (LEF Files)

□ Via Stacks



Without via stack



With via stack

- ◆ Higher density routing
- ◆ Easier usage of upper layer
- ◆ Must Follow minimum area rule



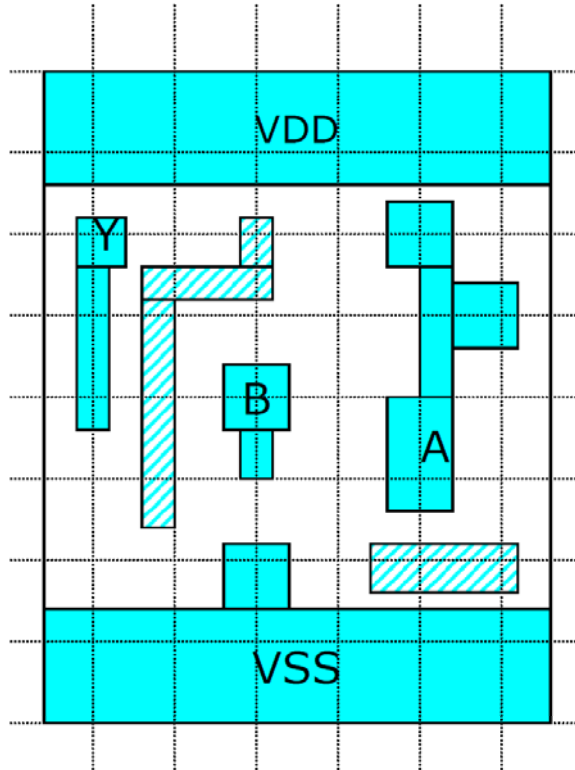
Placement & Routing: Setup (LEF Files)

Physical Macros

- Define physical data for
 - Standard cells
 - I/O pads
 - Memories
 - other hard macros
- Describe abstract shape
 - Size
 - Class
 - Pin
 - Obstruction



Placement & Routing: Setup (LEF Files)



```
MACRO ADD1
CLASS CORE ;
FOREIGN ADD1 0.0 0.0 ;
ORIGEN 0.0 0.0 ;
LEQ ADD ;
SIZE 19.8 BY 6.4 ;
SYMMETRY x y ;
SITE coresite
PIN A
  DIRECTION INPUT ;
  PORT
  LAYER Metal1 ;
  RECT 19.2 8.2 19.5 10.3
  .....
END
END A
OBS
  .....
END
END ADD1
```



Placement & Routing: Setup (LIB Files)

LIB Content

- Operating Conditions
 - Slow, fast, typical
- Pin type
 - input/output/inout
 - function
 - data/clock
 - capacitance
- Path delay
- Timing constraint
 - setup, hold, ...

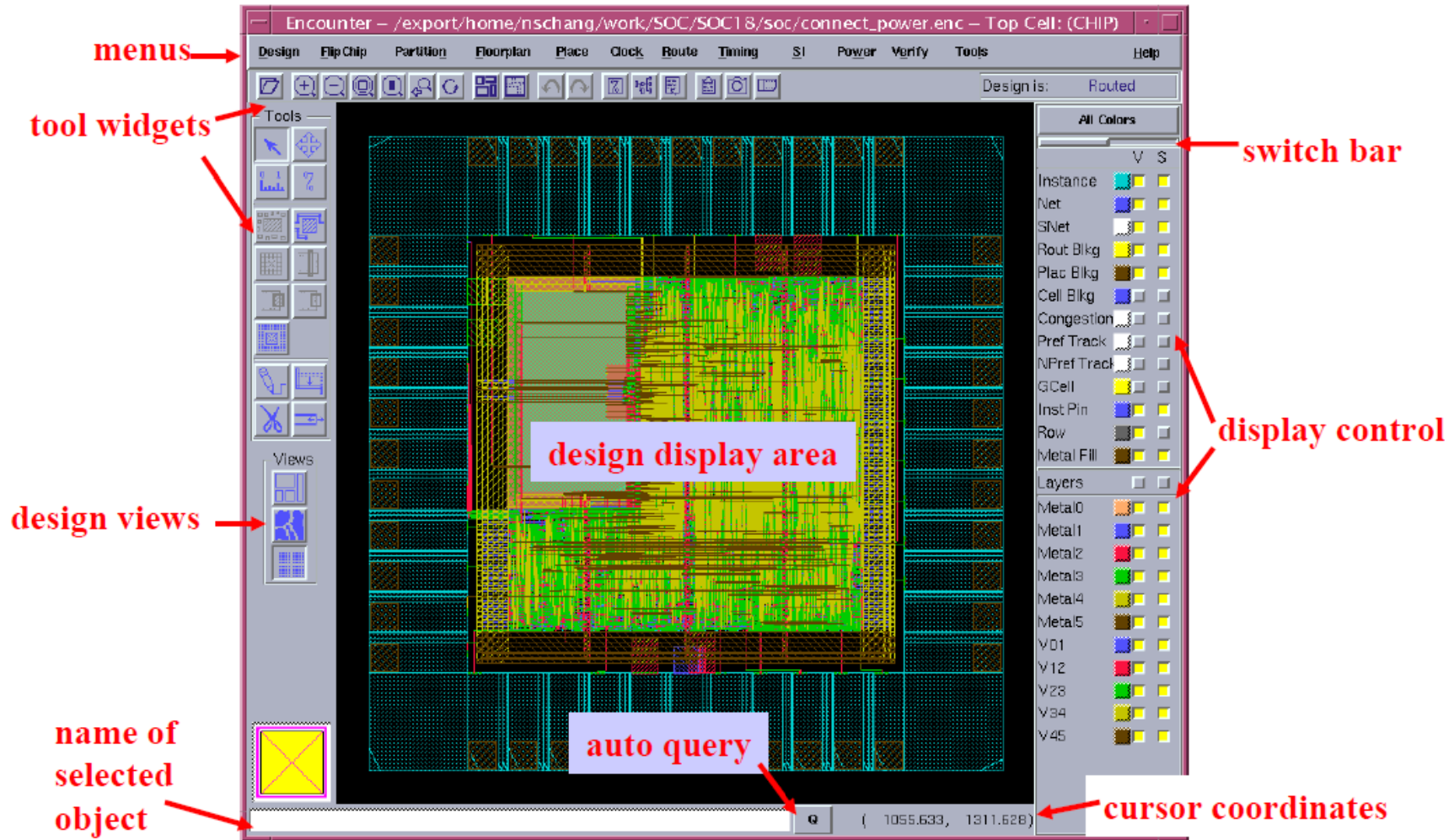


Placement & Routing: Setup

- ❑ If designing a chip , IO pads , power pads and Corner pads should be added before the netlist is imported.
- ❑ Make sure that there is no “assign” statement and no “ *cell* ” cell name in the netlist.
- ❑ Use the synthesis command below to remove assign statements
`set_boundary_optimization`
- ❑ Use the synthesis commands below to remove “*cell*” cell name
`define_name_rules name_rule –map {*cell* cell}`
`change_names –hierarchy –output name_rule`

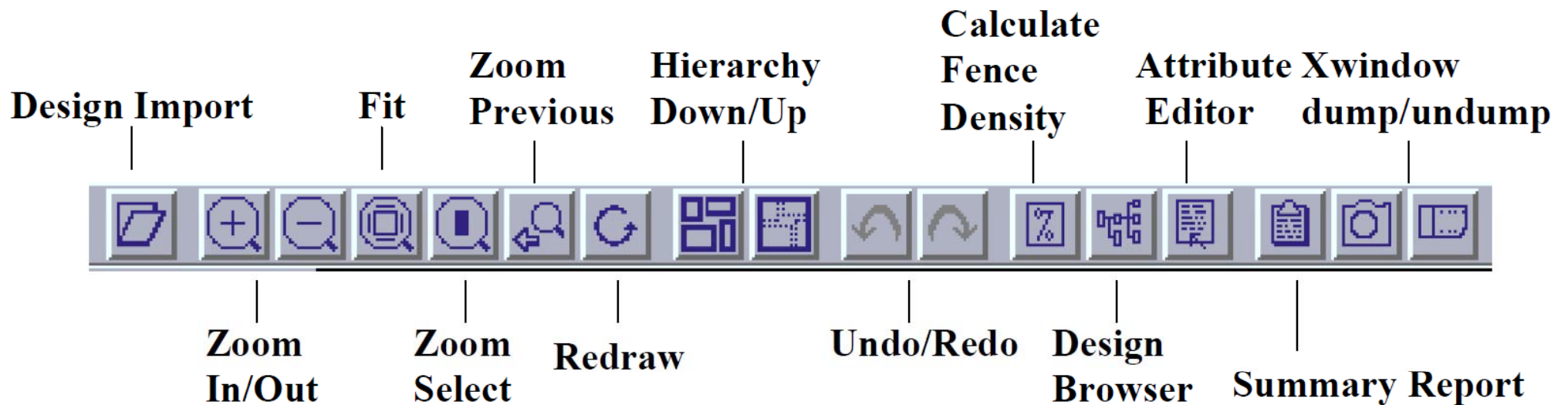


Placement & Routing: SOC Encounter GUI



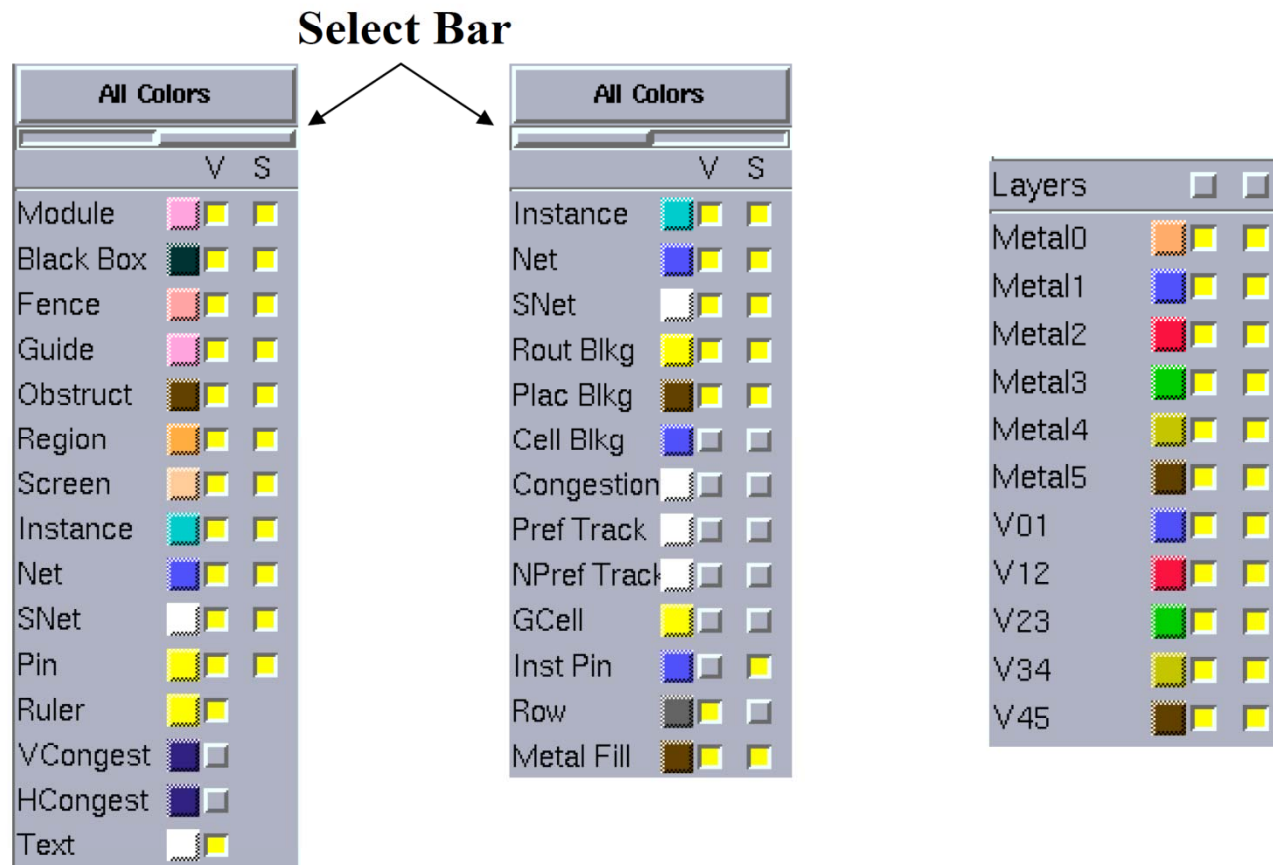
Placement & Routing: SOC Encounter GUI

☐ Tool Wedgits

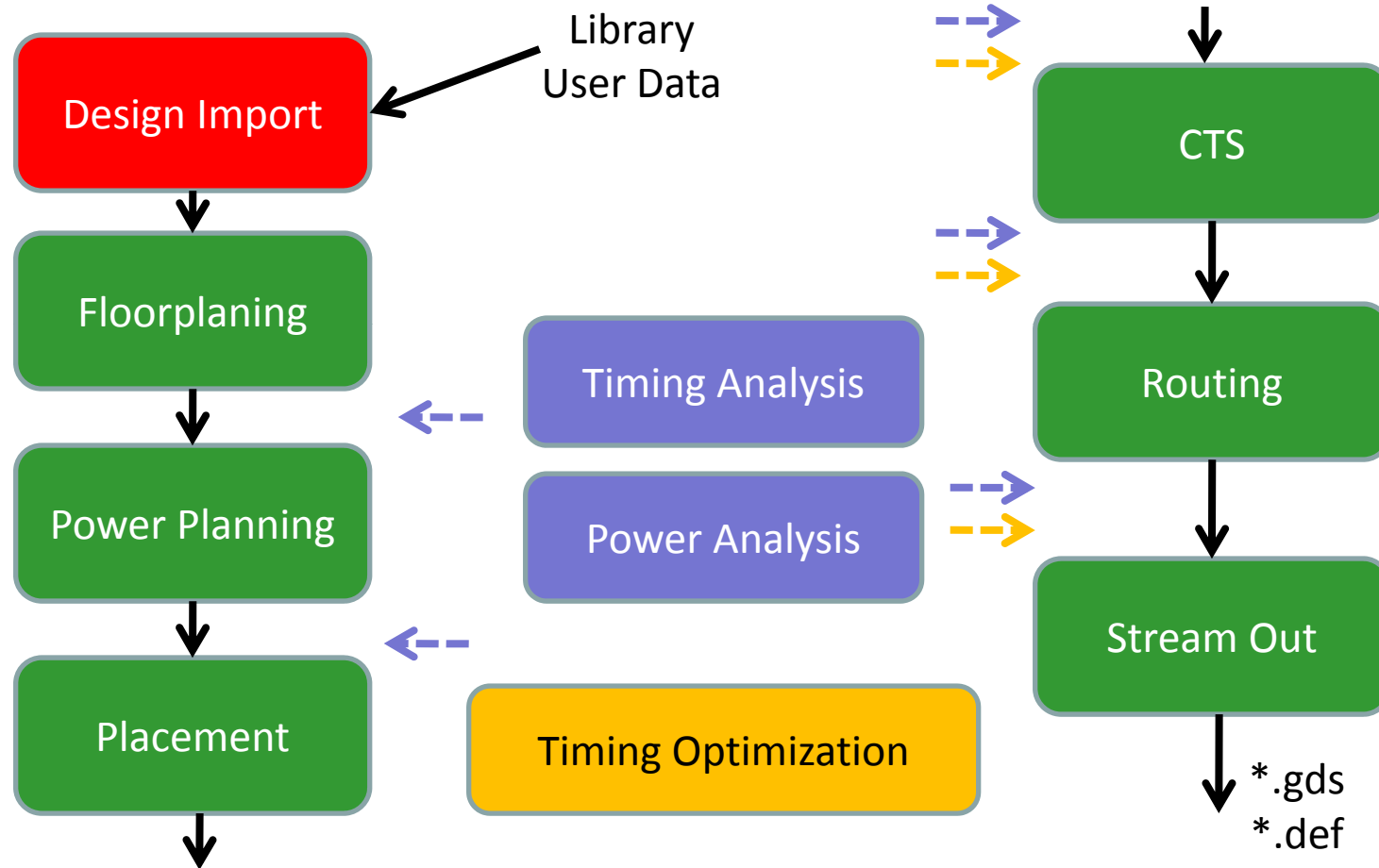


Placement & Routing: SOC Encounter GUI

☐ Display Control



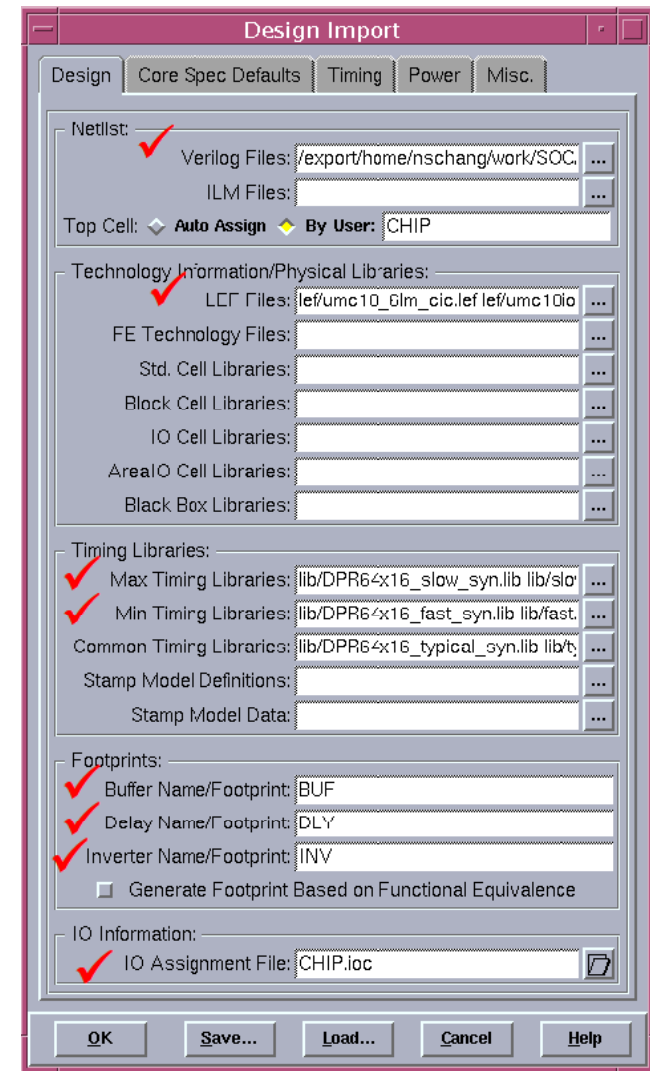
Placement & Routing Steps : Design Import



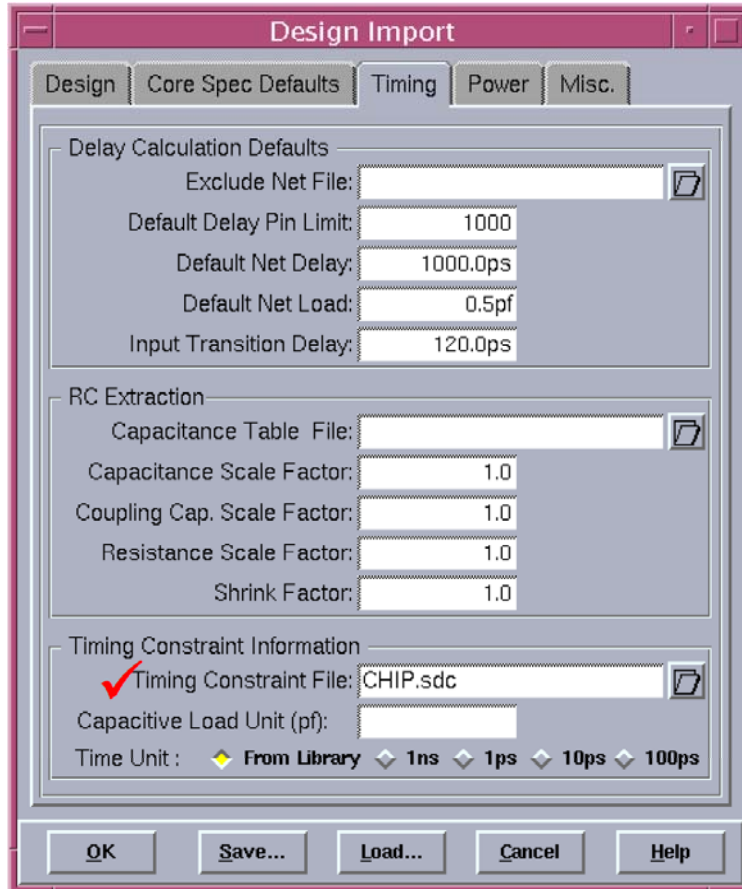
Placement & Routing Steps : Design Import

Design → Design Import

- Max Timing Libraries
 - Containing worst-case conditions for setup-time analysis
- Min Timing Libraries
 - Containing best-case conditions for hold-time analysis
- Common Timing Libraries
 - Used in both setup and hold analysis
- IO Assignment File:
 - Get a IO assignment template



Placement & Routing Steps : Design Import - Timing

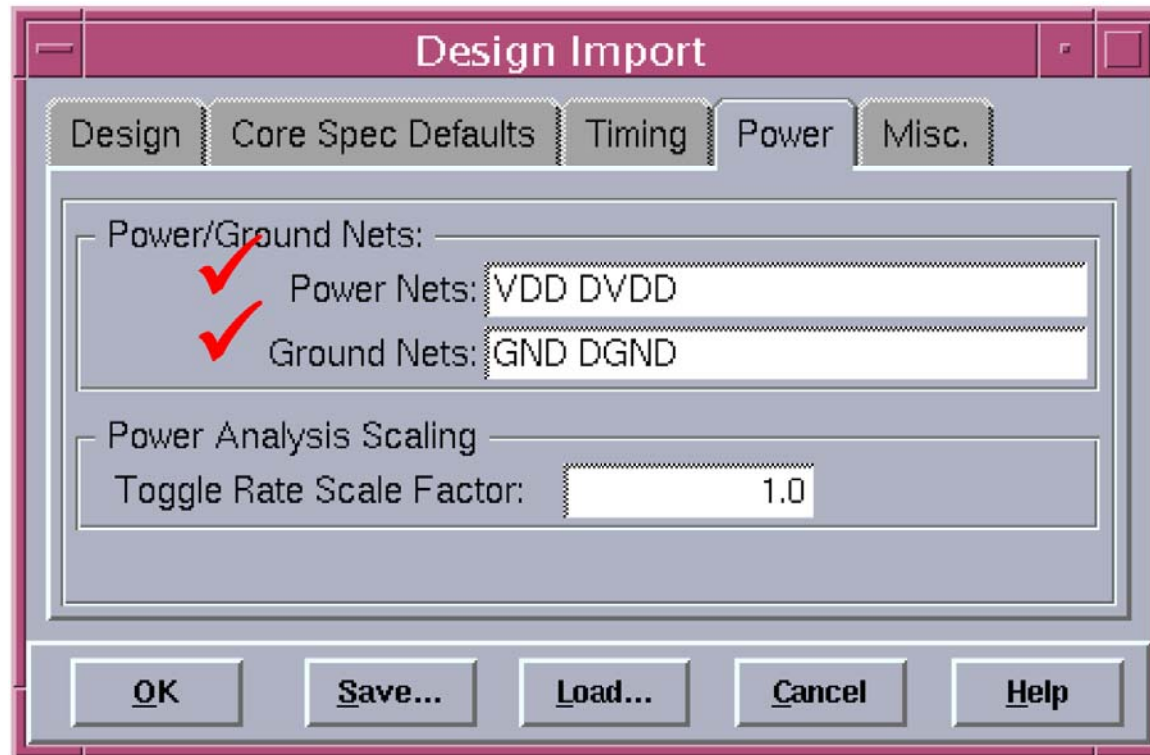


- ◆ Default Delay Pin Limit:
 - Nets with terminal counts greater than the specified value are assigned the default net delay and net load entries.
- ◆ Default Net Delay:
 - Set the delay values for a net that meets the pin limit default.
- ◆ Default Net Load:
 - Set the load for a net that meets the pin limit default.
- ◆ Input Transition Delay:
 - Set the Primary inputs and clock nets.



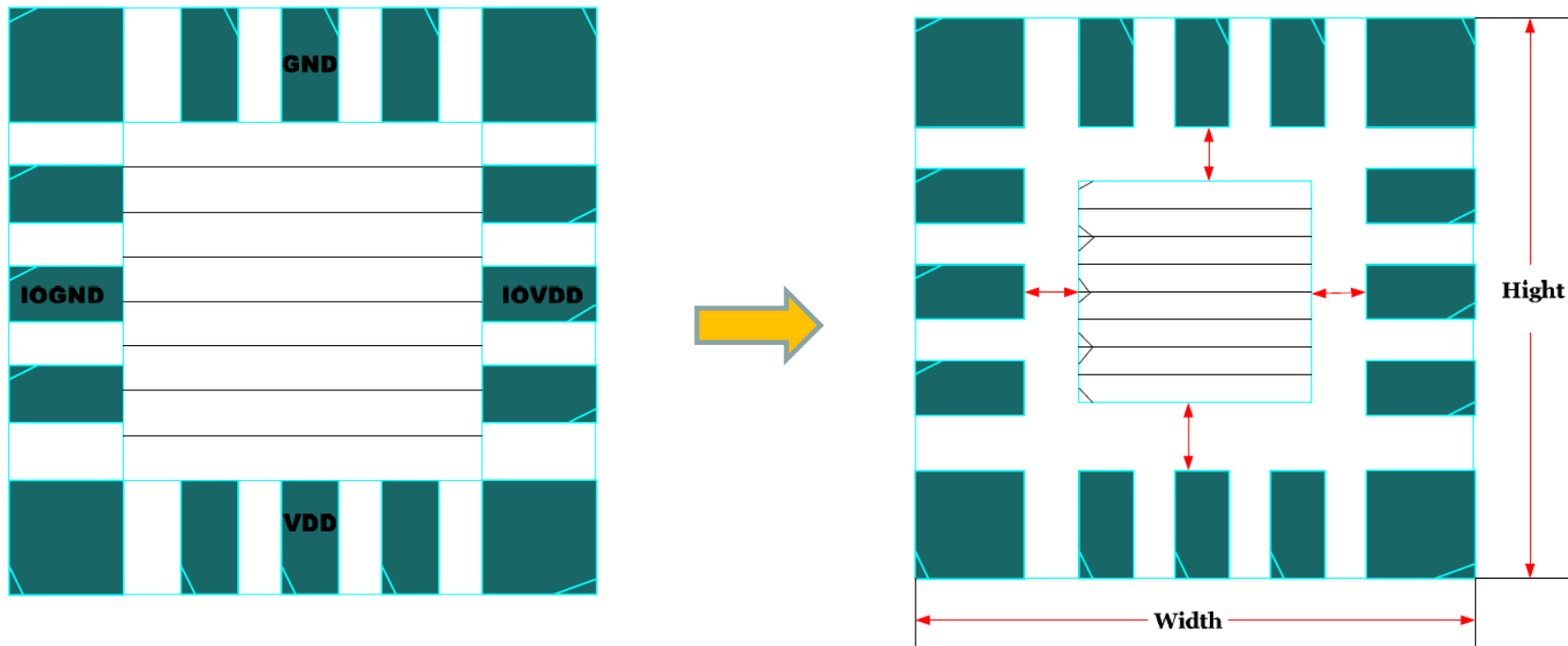
Placement & Routing Steps : Design Import - Power

- ❑ Specify the names of Power Nets and Ground Nets



Placement & Routing : IO Placement

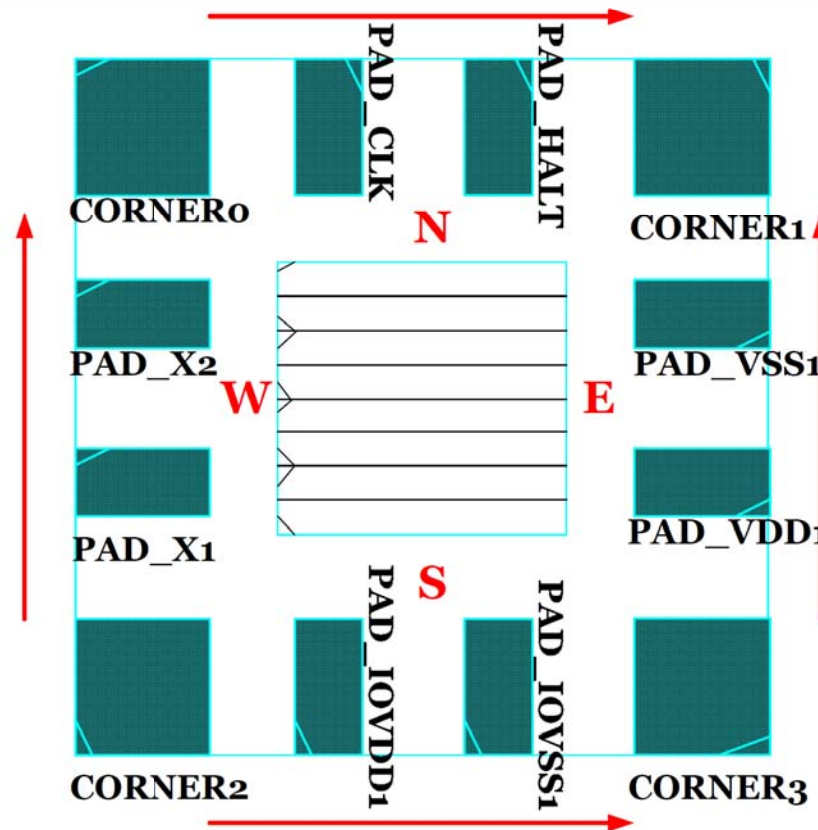
- ❑ Pads are inserted based on the specified IO files



Placement & Routing : IO Placement

- ❑ Pads are inserted based on the specified IO files

Version: 1	
Pad: CORNER0	NW
Pad: PAD_CLK	N
Pad: PAD_HALT	N
Pad: CORNER1	NE
Pad: PAD_X1	W
Pad: PAD_X2	W
Pad: CORNER2	SW
Pad: PAD_IOVDD1	S
Pad: PAD_IOVSS1	S
Pad: CORNER3	SE
Pad: PAD_VDD1	E
Pad: PAD_VSS1	E



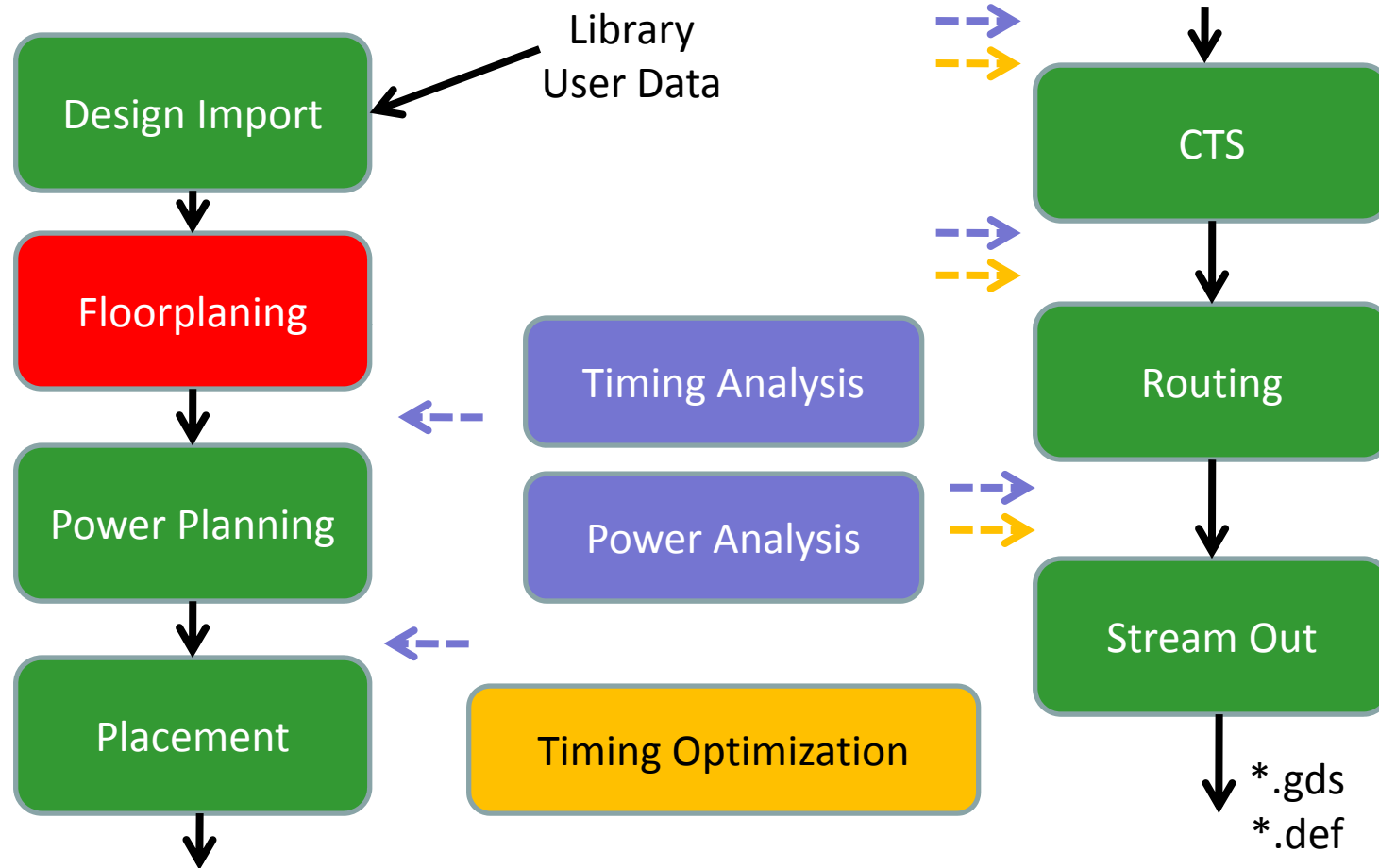
Placement & Routing : IO Placement

- ❑ Create an I/O assignment file manually using the following template:

```
MicronPerUserUnit: value  
Pin: pinName side |corner  
Pad: padInstanceName side|corner [cellName]  
Offset: length  
Skip: length  
Spacing: length
```

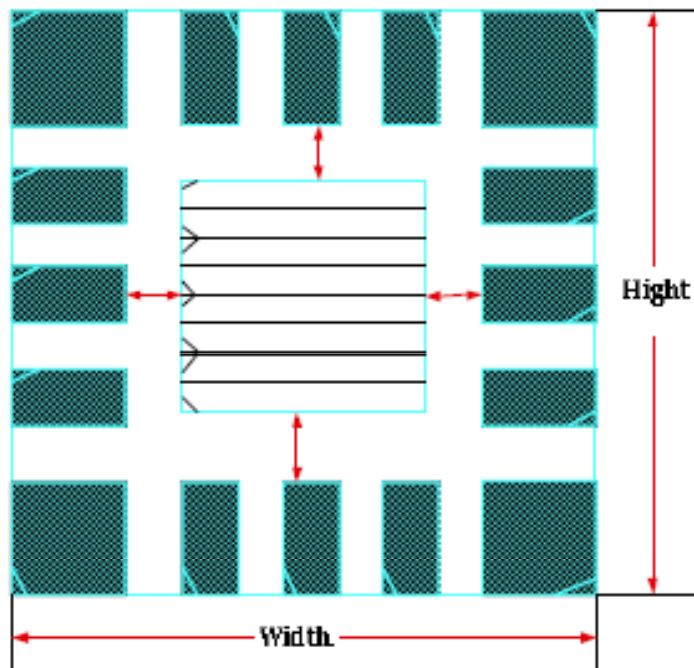


Placement & Routing Steps : Floorplaning



Placement & Routing Steps : Floorplaning

Floorplan → Specify Floorplan



Specify Floorplan

Design Dimensions

Specify Dimensions by:

- Size by:
 - Core Size by:
 - Aspect Ratio: Ratio (H/W): 0.9930020E
 - Core Utilization: 0.711806
 - Width and Height:
 - Core Height: 700.56
 - Core Width: 705.497
- Die Size by: Width and Height
 - Die Height: 1251.12
 - Die Width: 1255.617

Core Margins by:

- Core to IO Boundary
 - Core to Die Boundary
 - Core to Left: 80.32
 - Core to Right: 80.0
 - Core to Top: 80.7
 - Core to Bottom: 80.06

Die Size Calculation Use:

- Max IO Height
- Min IO Height

Floorplan Origin at:

- Lower Left Corner
- Center

Die/IO/Core Coordinates:

Die LL:	0.0	0.0	UR	1255.617	1251.12
IO LL:	194.9	194.9	UR	1060.717	1056.22
Core LL:	275.22	274.96	UR	980.717	975.52

unit: micron

Standard Cell Rows

- Double-back rows:
- Bottom row orient:
- Row Spacing: 0.0 um For Every 2 Row
- Row height: 5.04

IO Specifications

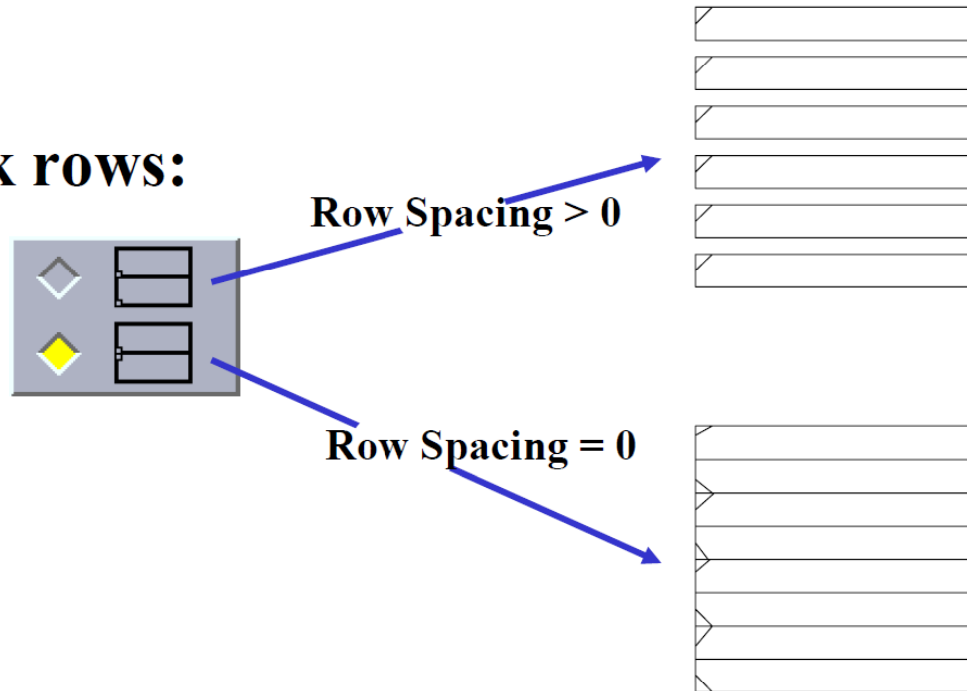
Bottom IO Pad Orientation:

OK Apply Cancel Help



Placement & Routing Steps : Floorplaning

Double-back rows:

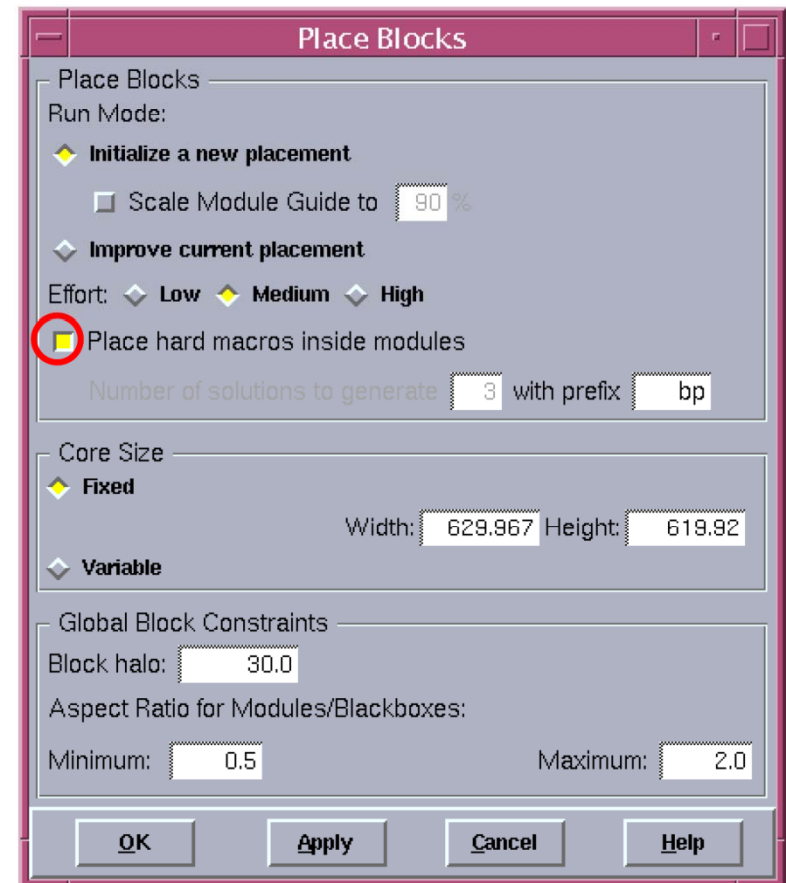


Placement & Routing Steps : Floorplaning

Floorplan → Place Blocks/Modules → Place ...

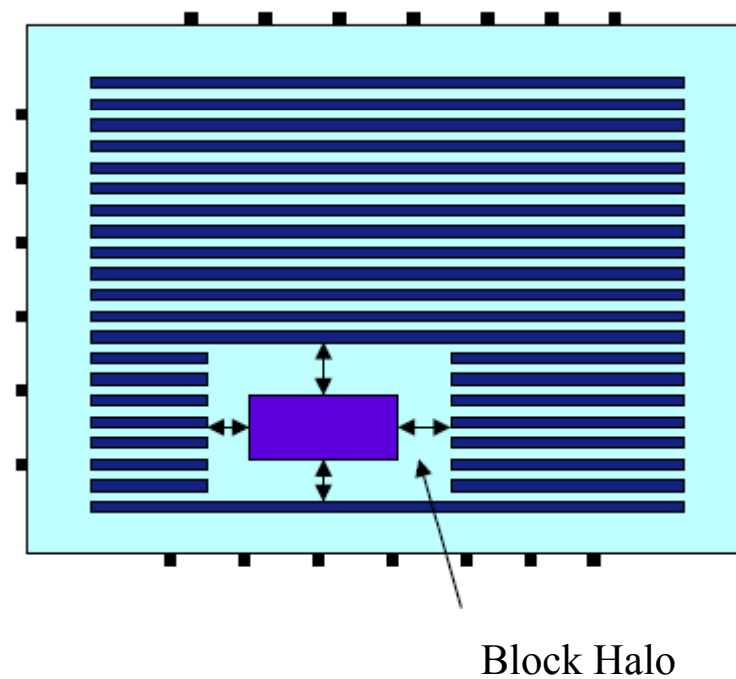
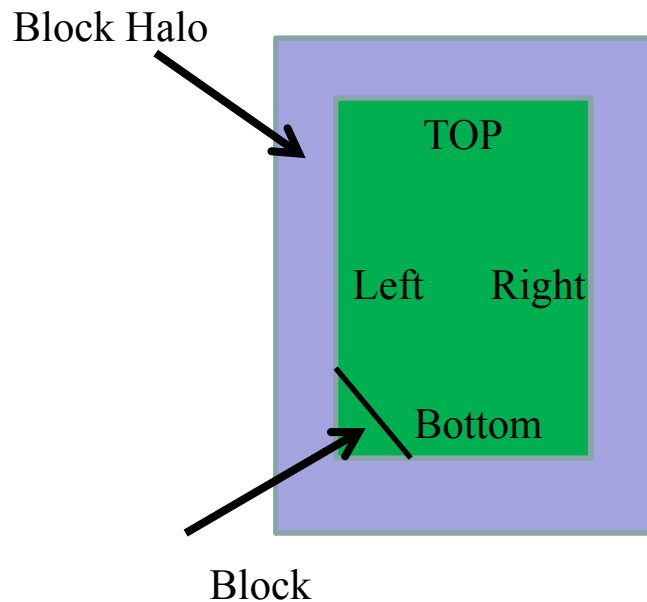
- Automatic place blocks
 - Black boxes and partitions and hard macros at the top-level design

- Block halo
 - Specifies the minimum amount of space around blocks that is reserved for routing.



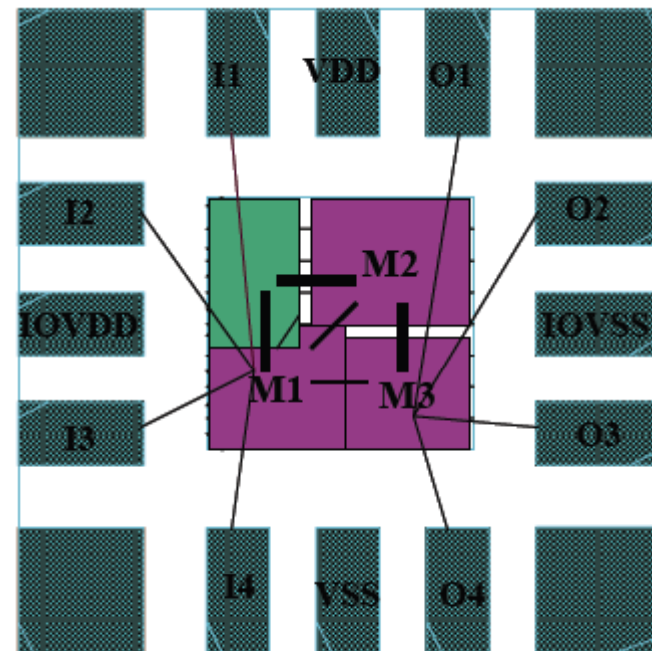
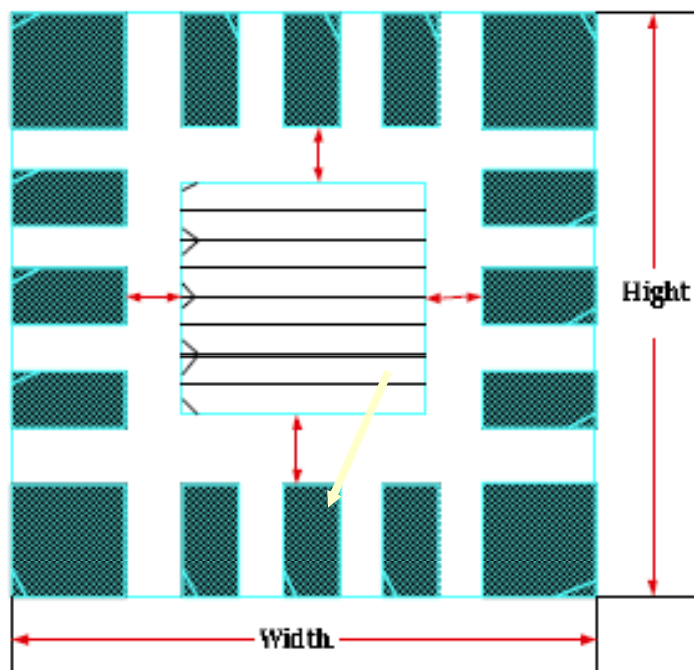
P&R Steps : Floorplaning (Black Halo)

□ Block Halo ابزاری است که به ما این امکان را می دهد که از بروز تراکم در مرز بلوکها و سلولهای استاندارد جانمایی شده در طرح جلوگیری کنیم.

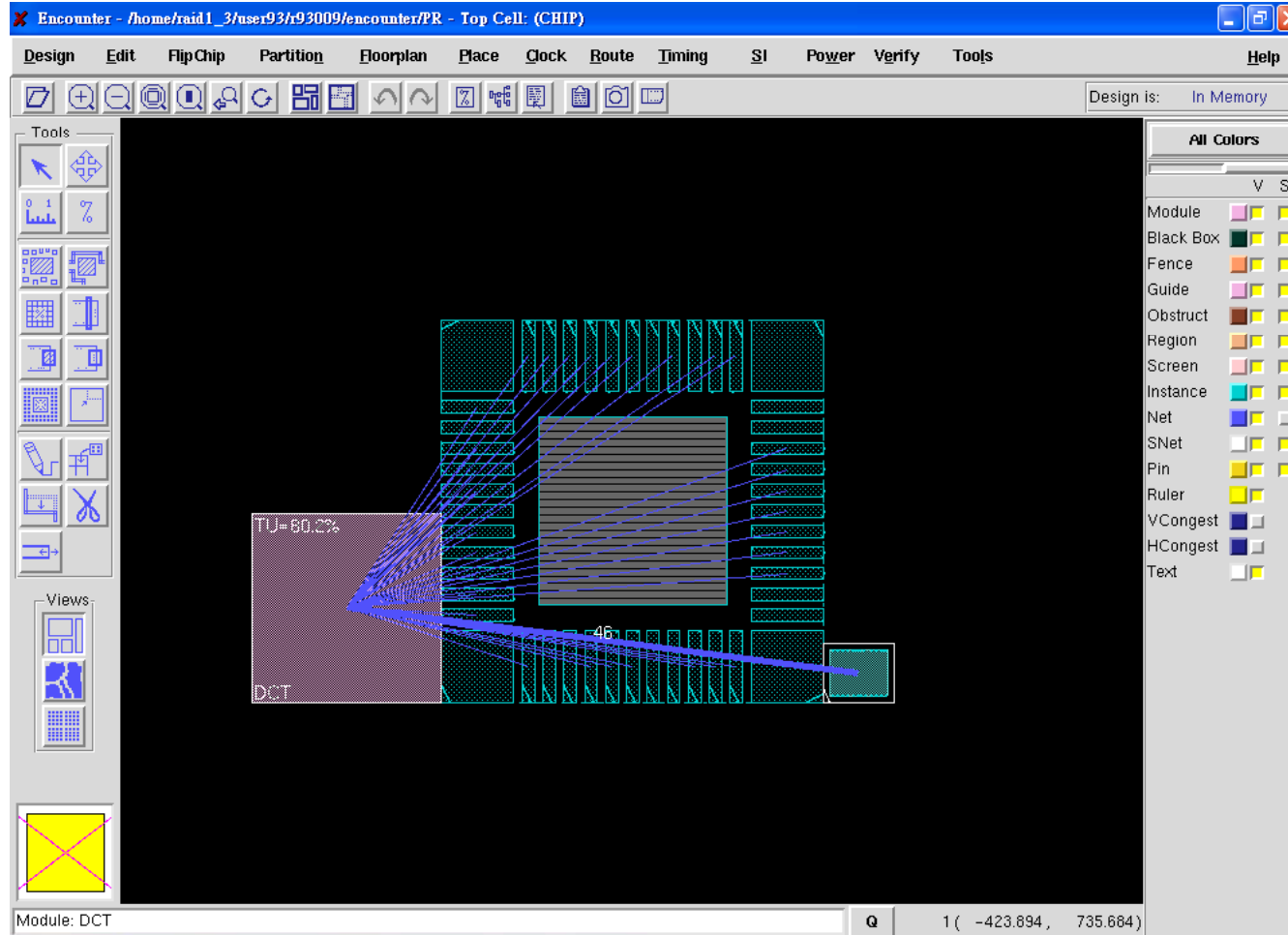


Placement & Routing Steps : Floorplaning

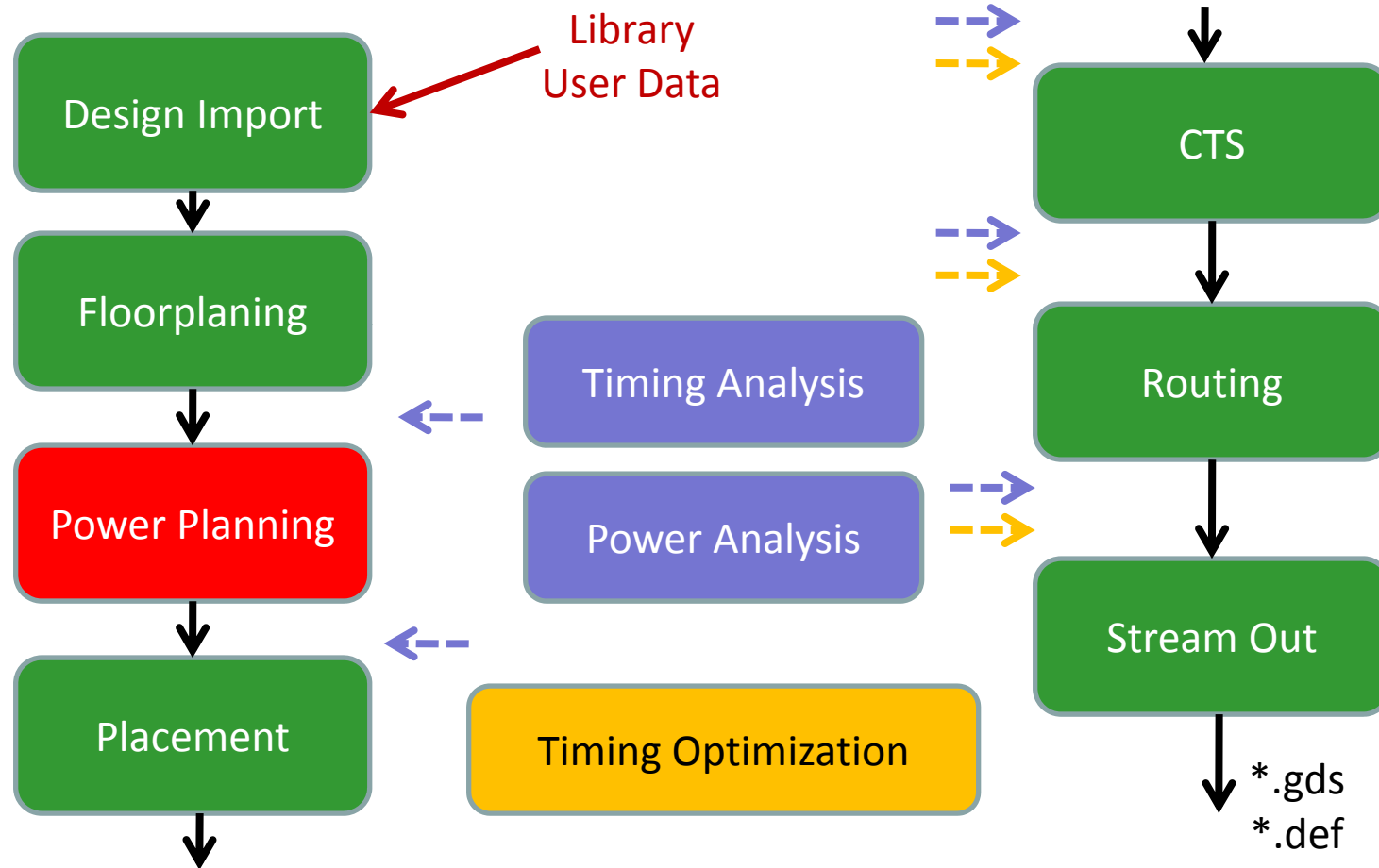
Floorplan → Specify Floorplan



Placement & Routing Steps : Floorplaning



Placement & Routing Steps: Power Planning



P&R Steps : Power Planning

Power → Power Planning → Add Rings

The screenshot shows the 'Power Planning' dialog box with the following settings:

- Net(s):** GND VDD
- Ring Type:**
 - Core Ring(s) contouring:
 - Around core boundary
 - Along I/O boundary
 - Exclude selected objects
 - Block ring(s) around
- Ring Configuration:**

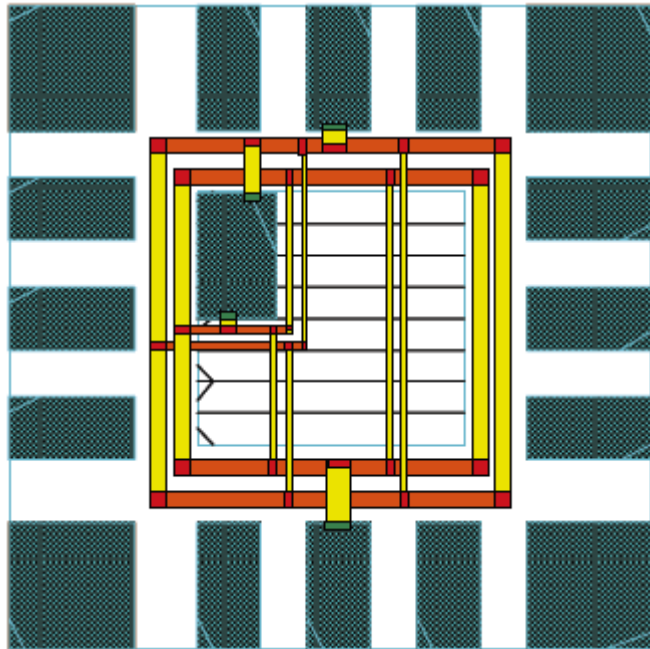
	Top:	Bottom:	Left:	Right:
Layer:	ME5	ME5	ME4	ME4
Width:	10	10	10	10
Spacing:	0.32	0.32	0.32	0.32
Offset:	<input checked="" type="checkbox"/> Center in channel	<input checked="" type="checkbox"/> Specify	<input checked="" type="checkbox"/> Center in channel	<input checked="" type="checkbox"/> Specify
	0.33	0.33	0.33	0.33
- Wire Group:**
 - Use wire group
 - Interleaving
 - Number of bits: 3

The 'Update' button is circled in red.

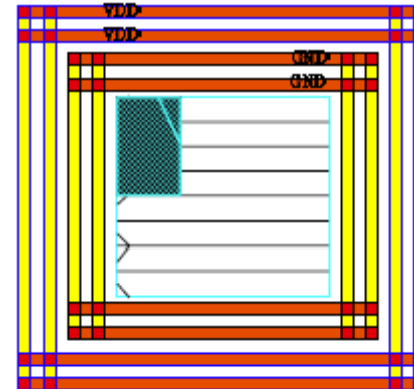


P&R Steps : Power Planning

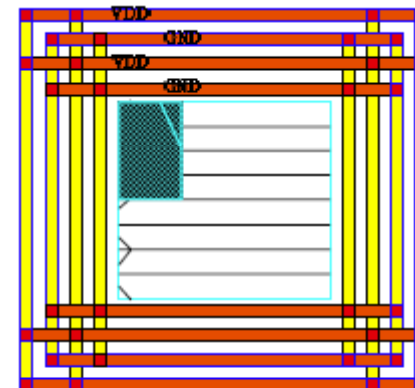
Power → Power Planning → Add Rings



Wire Group
No interleaving



Wire Group
interleaving



P&R Steps : Power Planning – Block Ring

Basic | Advanced | Via Generation

Net(s): GND VDD

Block ring(s) around

- Each block
- Selected power domain/fences
- Each selected block and/or group of core rows
- Clusters of selected blocks and/or groups of core rows

With shared ring edges

Layer:	Top:	Bottom:	Left:	Right:
	ME5	ME5	ME4	ME4
Width:	10	10	10	10
Spacing:	0.32	0.32	0.32	0.32
Offset:	<input checked="" type="checkbox"/> Center in channel	<input checked="" type="checkbox"/> Specify		
	0.33	0.33	0.33	0.33

Wire Group

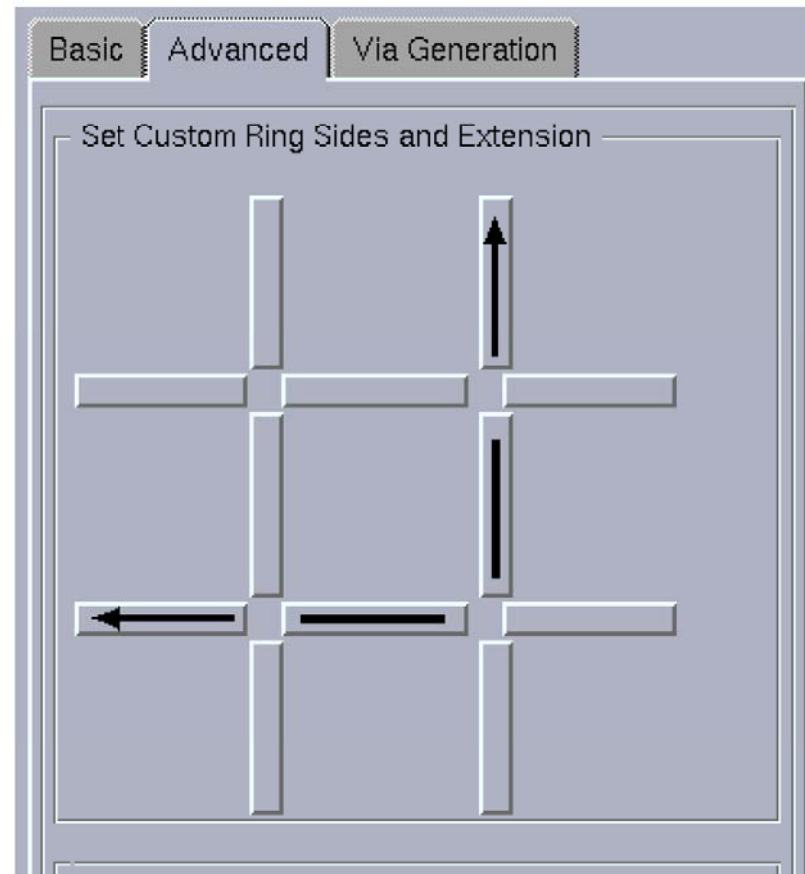
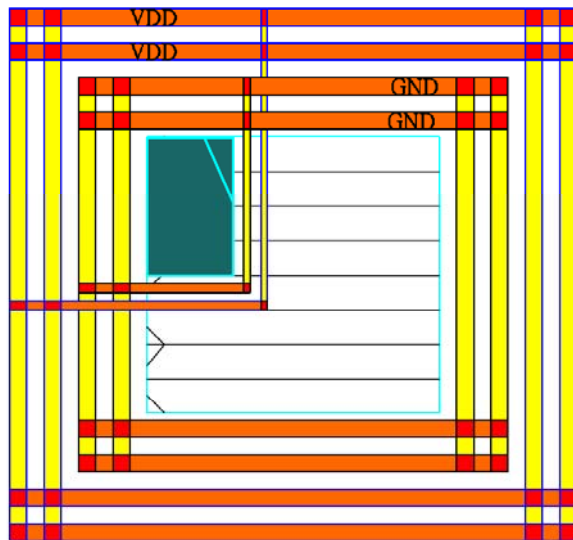
Use wire group

Interleaving

Number of bits: 3



P&R Steps : Power Planning – Block Ring



P&R Steps : Power Planning (Add Strip)

Power → Power Planning → Add Strips

Basic Advanced Via Generation

Set Configuration

Net(s): GND VDD

Layer: ME4

Direction: Vertical Horizontal

Width: 5

Spacing: 0.28

Set Pattern

Set-to-set distance: 100

Number of sets: 3

Stripe Offset Boundary

Selected power domain/fence

Relative from core or area:

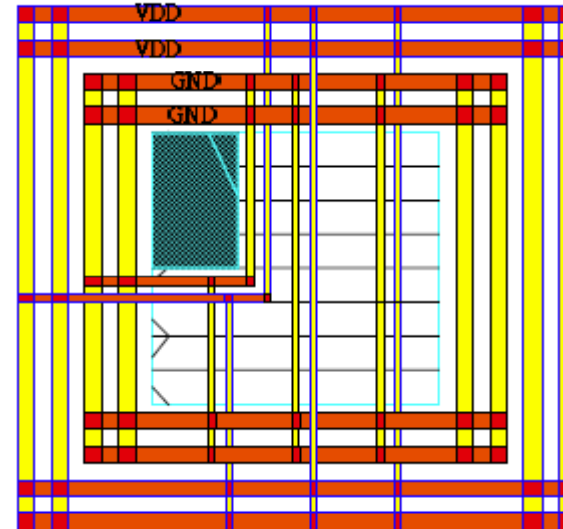
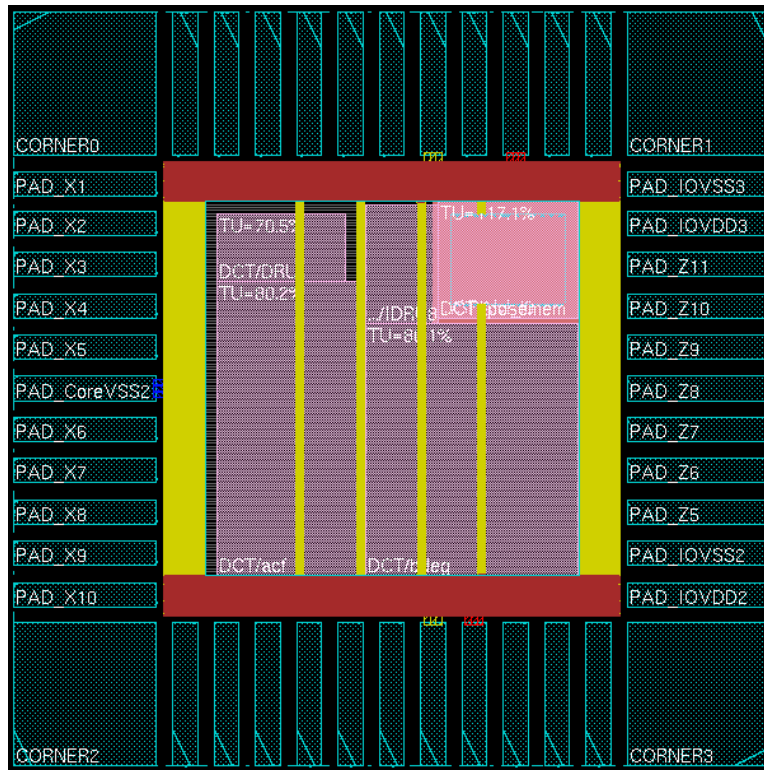
X from left: 150

X from right: 150

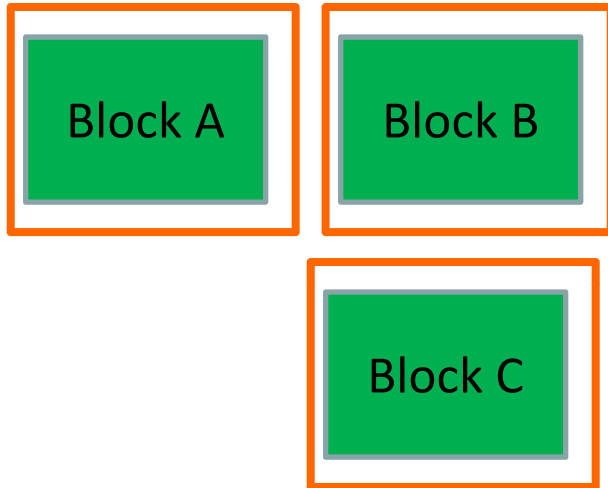


P&R Steps : Power Planning (Add Strip)

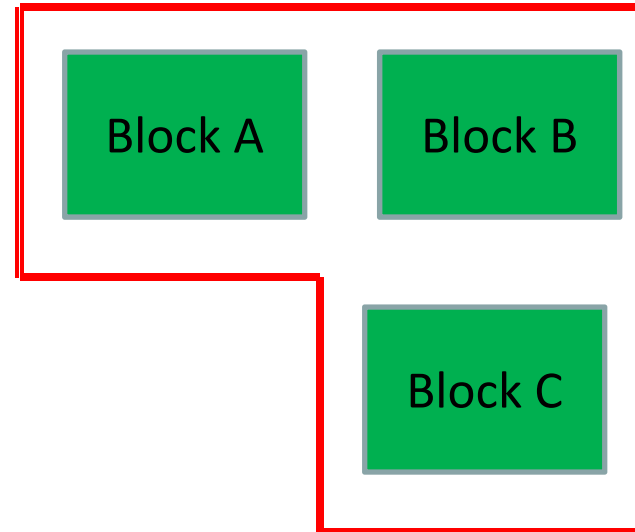
Power → Power Planning → Add Strips



P&R Steps : Power Planning



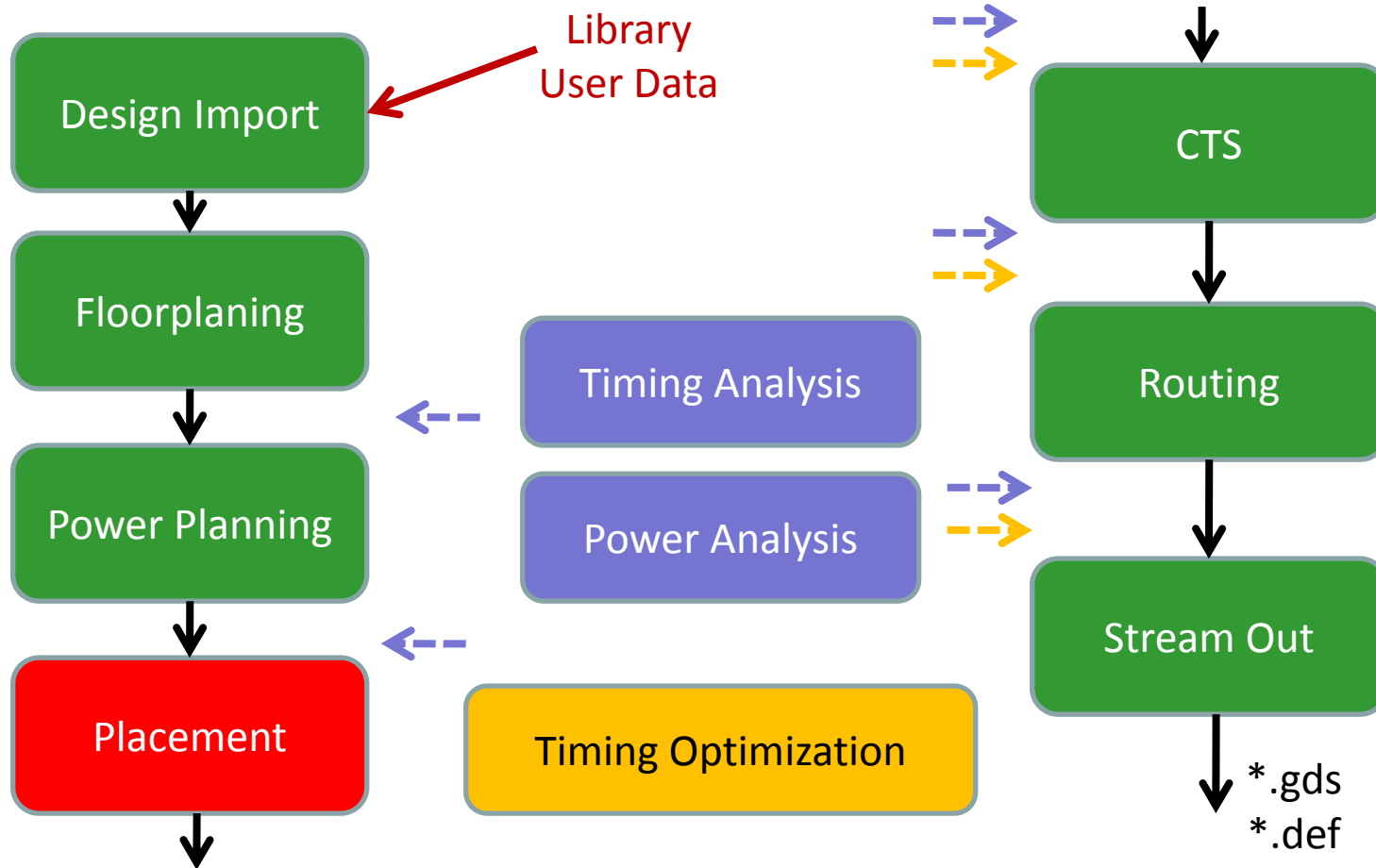
Power Planning without
Shared Ring Edge



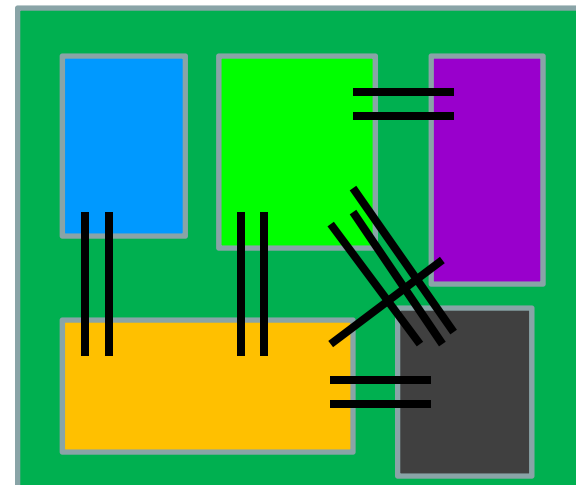
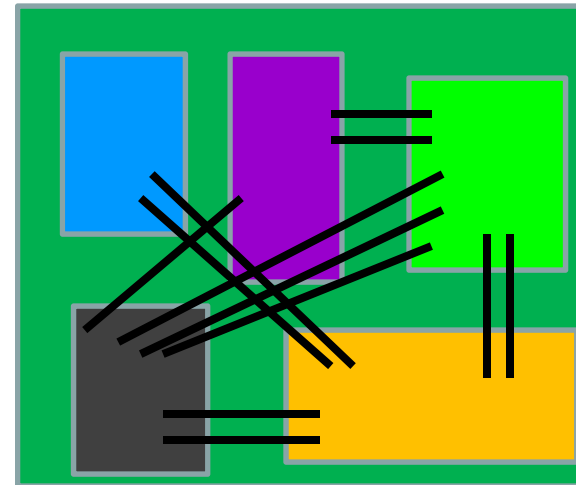
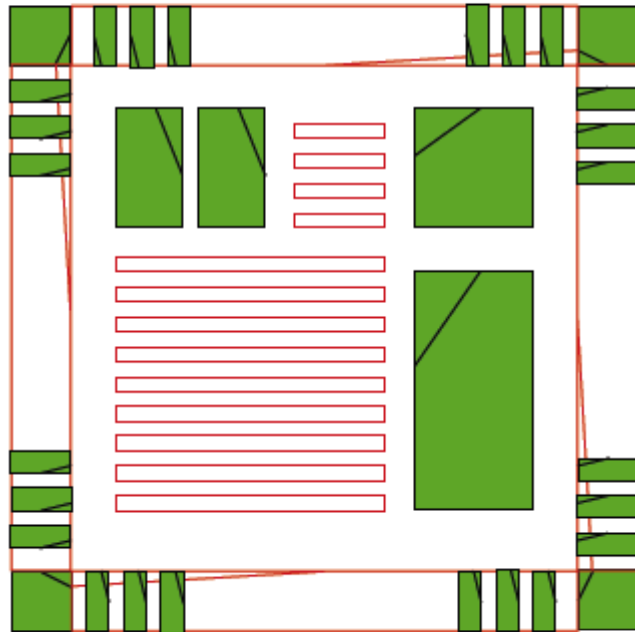
Power Planning with shared
Ring Edge



Placement & Routing Steps: Placement

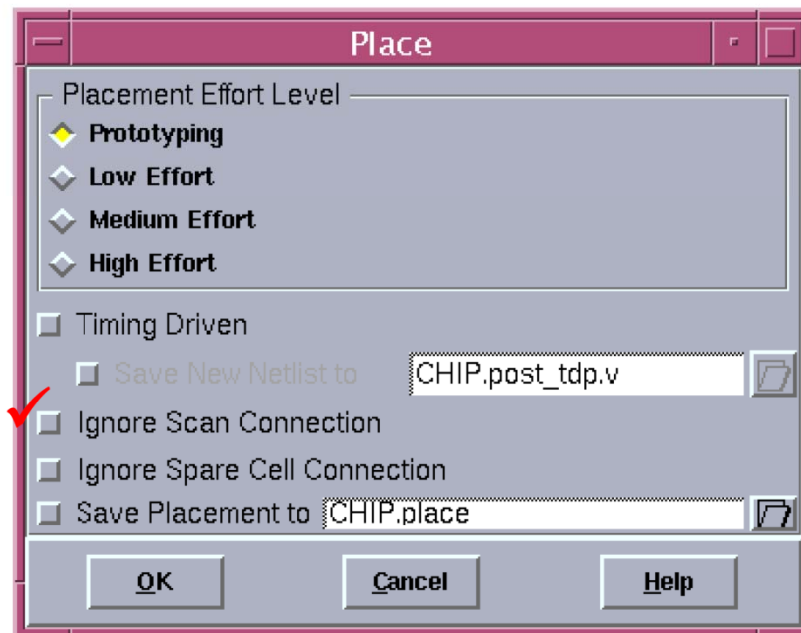


P&R Steps : Placement

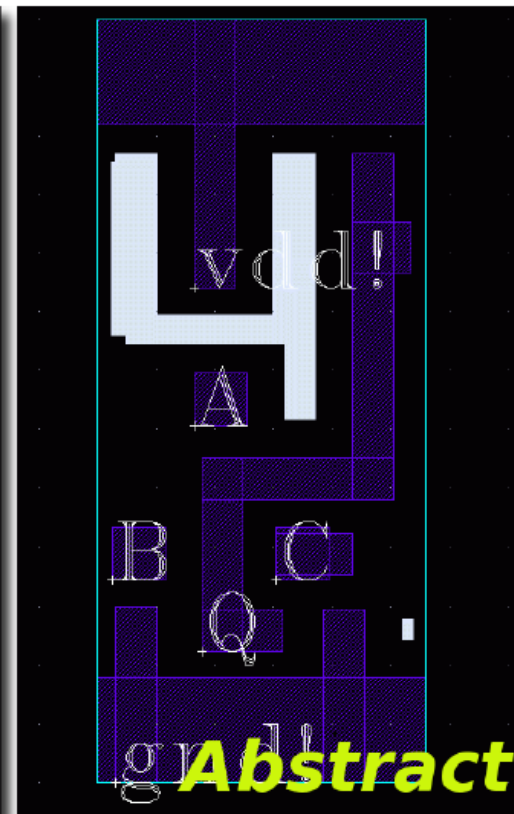
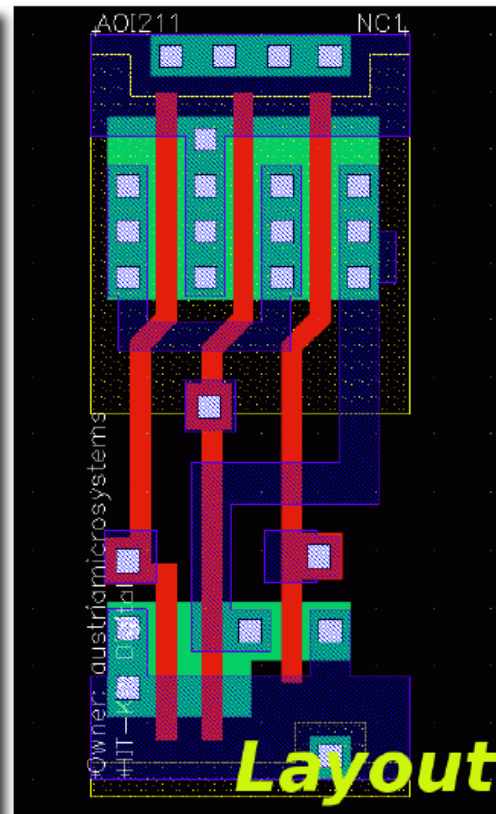
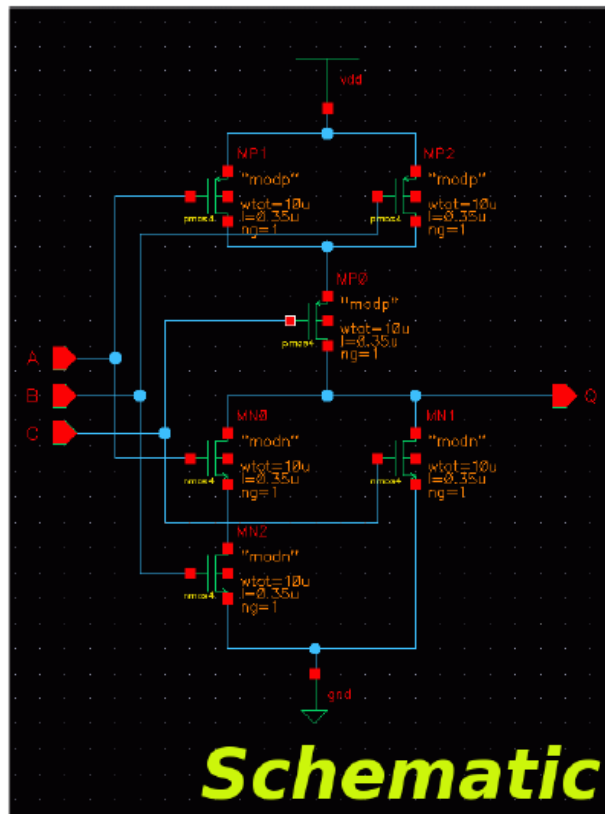


P&R Steps : Placement

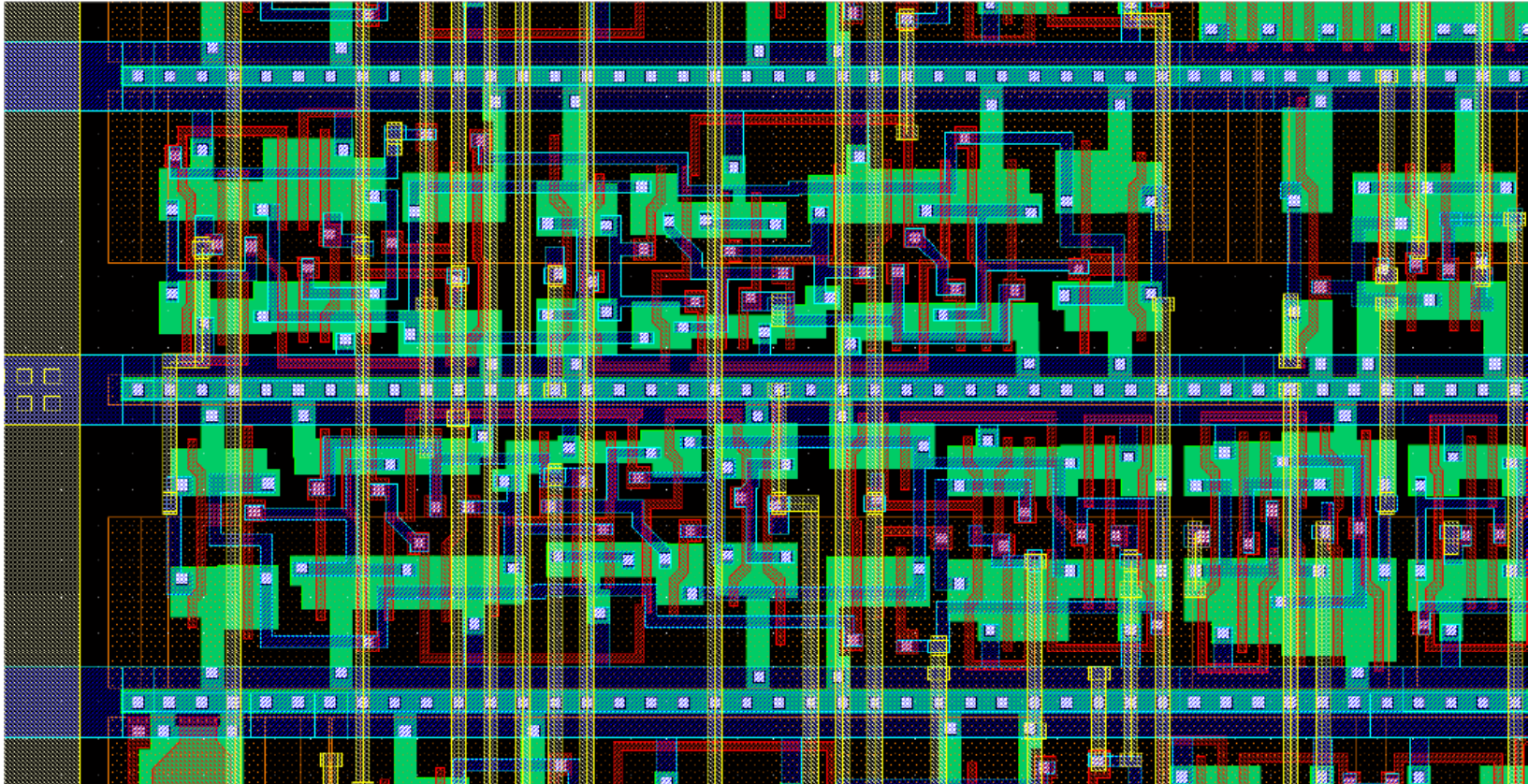
- ◆ Prototyping : Runs quickly, but components may not be placed at legal location.
- ◆ Timing Driven:
 - Build timing graph before place.
 - meeting setup timing constraints with routability.
 - Limited IPO by upsizeing/downsizing instances.
- ◆ Ignore Scan Connection
 - nets connected to either the scan-in or scan-out are ignored.
- ◆ Check placement after placed
 - *place* → *Check Placement*



P&R Steps : Placement (Standard Cells)



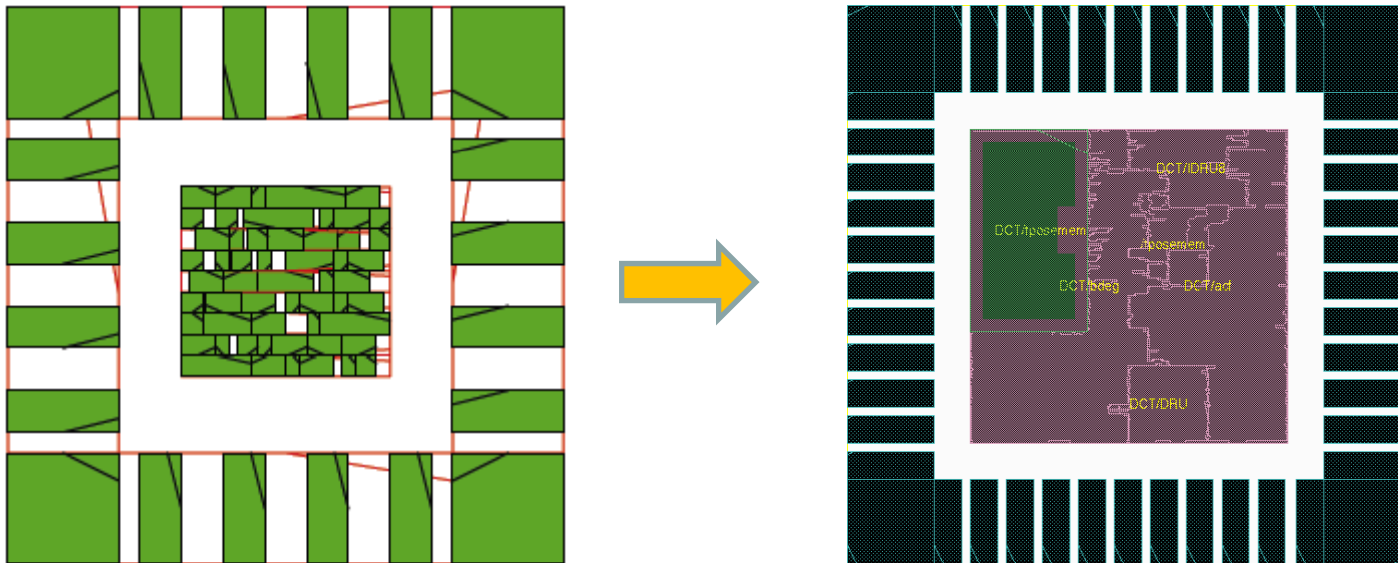
P&R Steps : Placement (Standard Cell Rows)



P&R Steps : Placement

- در این مرحله ما قادر خواهیم بود که سلولهای استاندارد را در سطح Core جانمایی کنیم.
- سلولهای استاندارد تا جایی که ممکن است در مکانهای نزدیک بهم قرار می گیرند.

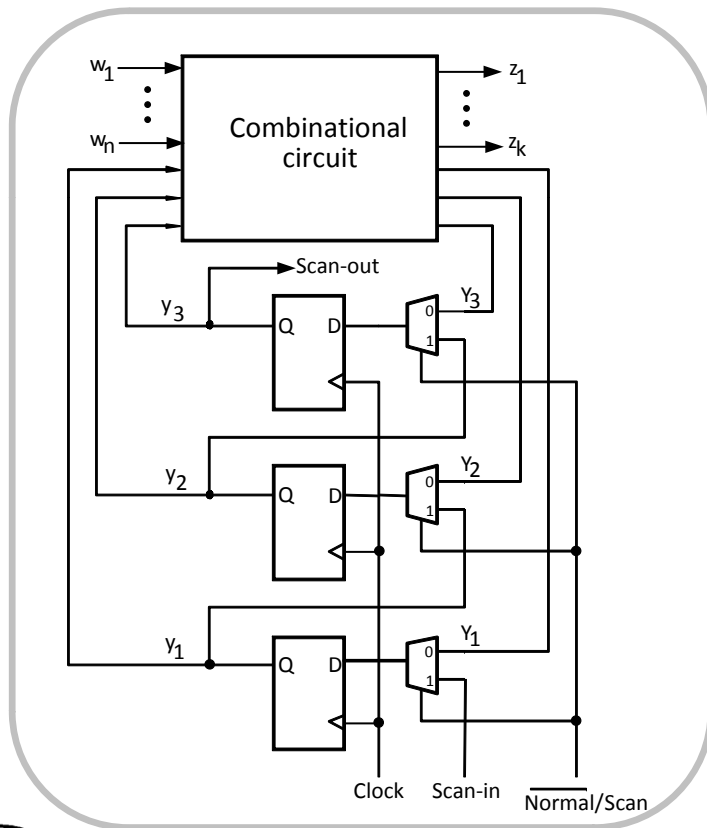
Place → Standard Cells...



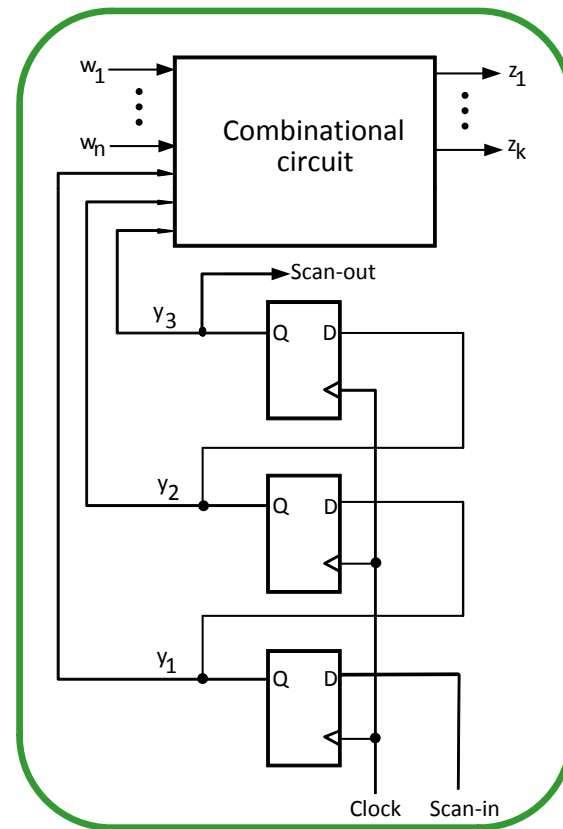
P&R Steps : Scan Chain

□ Scan chain is normally used for after tape-out verification

- Full Scan
- Partial Scan

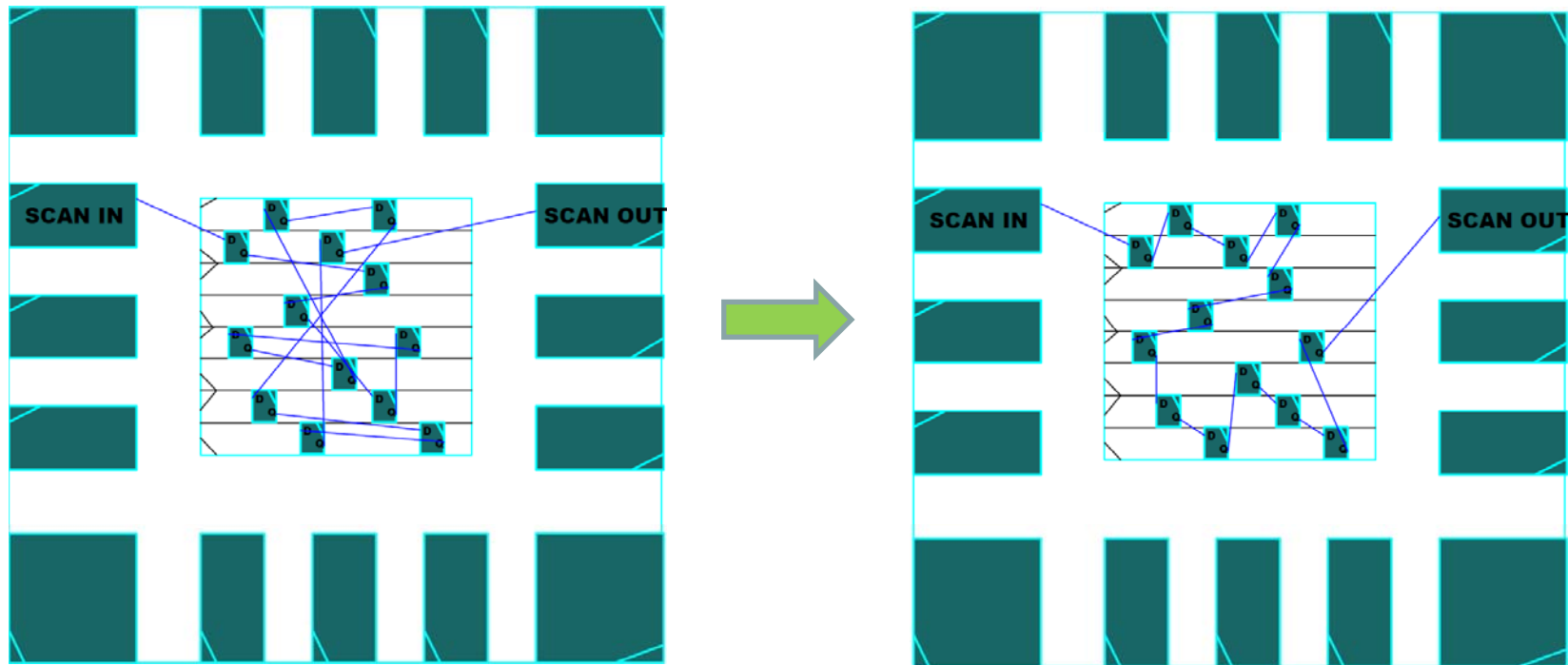


Scan
Mode
➔

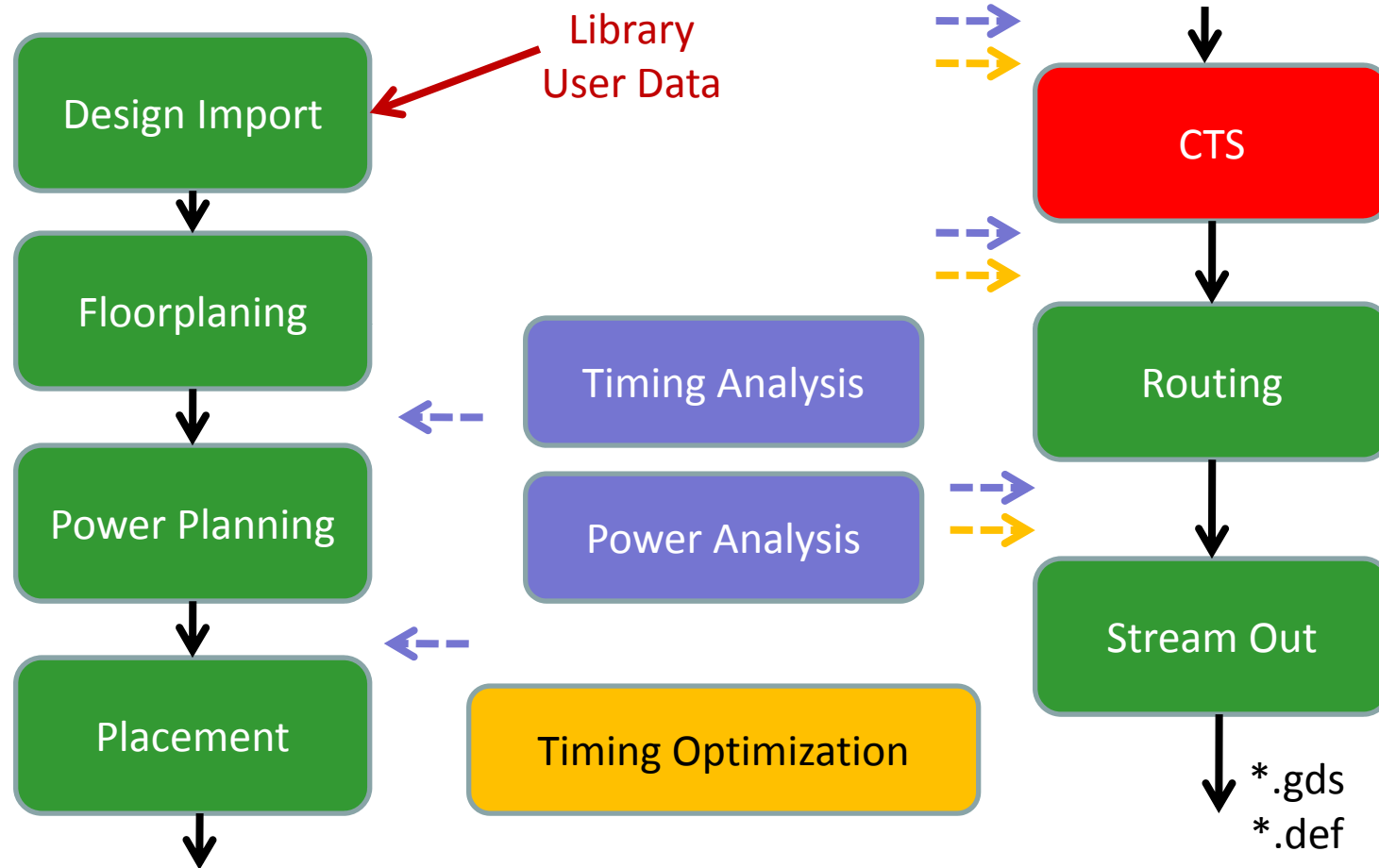


P&R Steps : Scan Chain

- ❑ Scan chain is normally used for after tape-out verification
 - Full Scan
 - Partial Scan



Placement & Routing Steps : Clock Tree Synthesis



Placement & Routing Steps : Clock Tree Synthesis

Clock Issues

- Heavy clock net loading
- Long clock insertion delay
- Clock skew
- Skew across clocks
- Clock to signal coupling effect
- Clock is power hungry
- Electromigration on clock net

Solutions of these problems may be conflict

Clock is one of the most important treasure in a chip, do not take it as others



Placement & Routing Steps : Clock Tree Synthesis

- به ماکزیمم تفاوت Arrival Time سیگنال کلاک به دو Component متفاوت Clock Skew گویند.
- Clock Skew طراح را وادار می کند تا از دوره تناوب بزرگتری برای کلاک استفاده کند. این امر باعث کند شدن سیستم می شود.
- برای اجتناب از این مشکل باید Clock Skew هنگام مسیریابی کلاک به مینیمم مقدار ممکن برسد.



Placement & Routing Steps : Clock Tree Synthesis

Skew

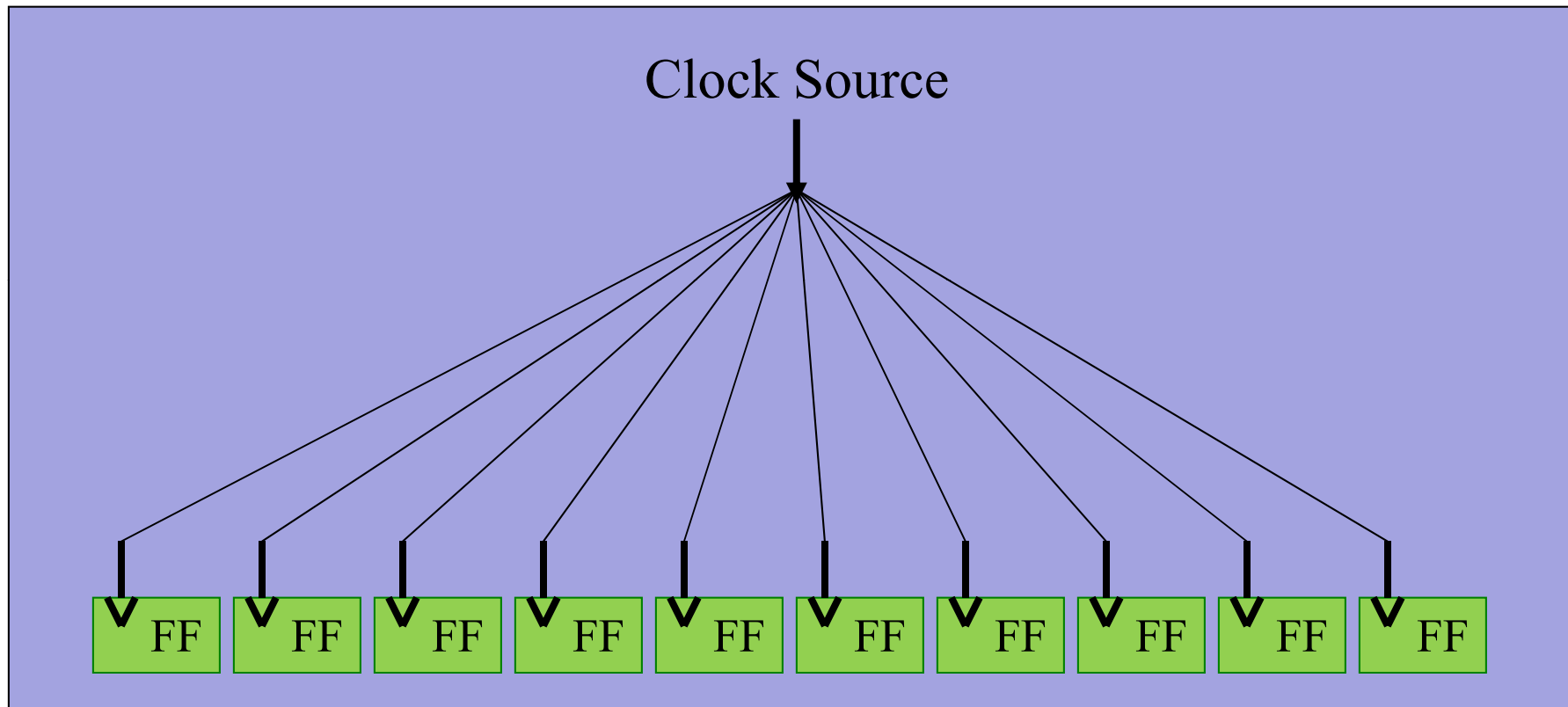
- مهمترین مشکل شبکه clock است.
- ممکن است ۱۰٪ دوره تناوب کلاک را به خود اختصاص دهد.

Power

- کلاک بیشترین مصرف کننده توان سیستم است.
- تعداد زیادی از نودهای Sink باید به منبع کلاک متصل شوند.
- چون کلاک یک سیگنال Global است در نتیجه نتهای کلاک بسیار طولانی هستند.



Placement & Routing Steps : Clock Tree Synthesis



Placement & Routing Steps : Clock Tree Synthesis

□ استفاده از متد Clock Tree می تواند به رفع مشکلات کلاک کمک کند زیرا:

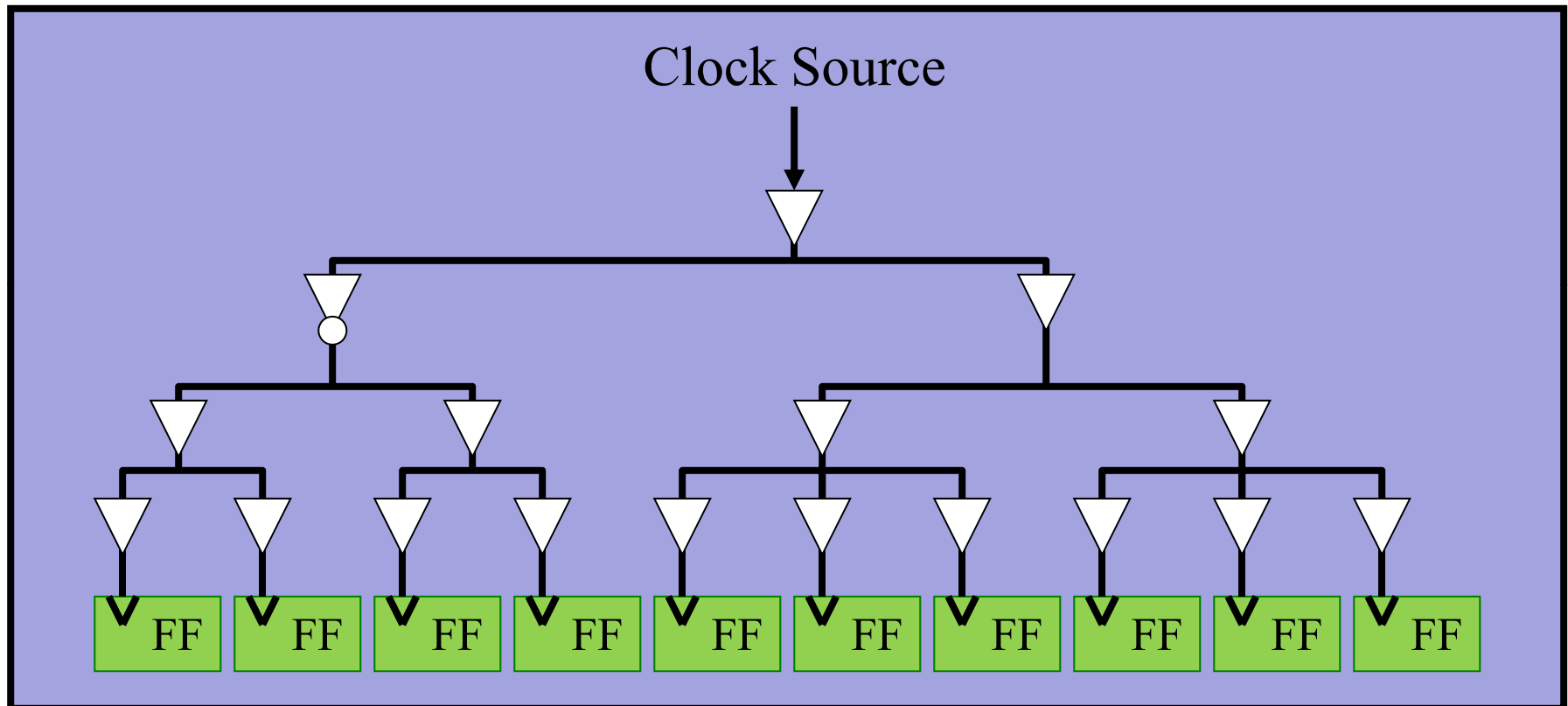
▪ Clock Skew و طول انتها بالانس می شوند.

▪ با استفاده از Buffer Insertion در Clock Tree می توان Clock

Skew و تاخیر را کاهش داد.



Placement & Routing Steps : Clock Tree Synthesis



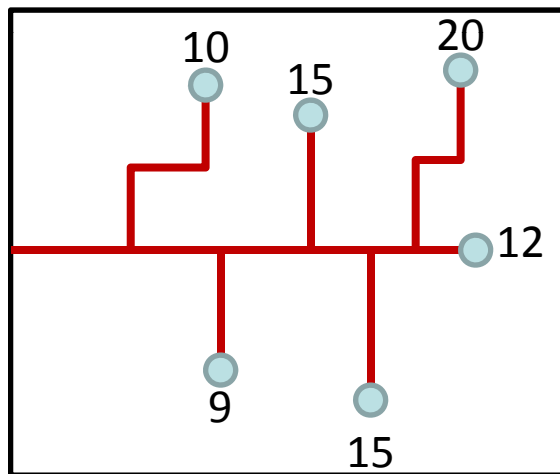
Placement & Routing Steps : Clock Tree Synthesis

$$\text{Clock Period} \geq t_{\text{logic}} + t_{\text{skew}} + t_{\text{su}}$$

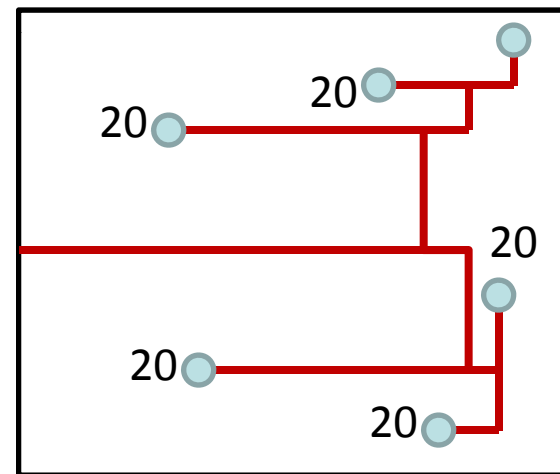
t_{logic} : بلندترین مسیر در Logic ترکیبی

Clock Skew

Setup Time : t_{su} المانهای سنکرون



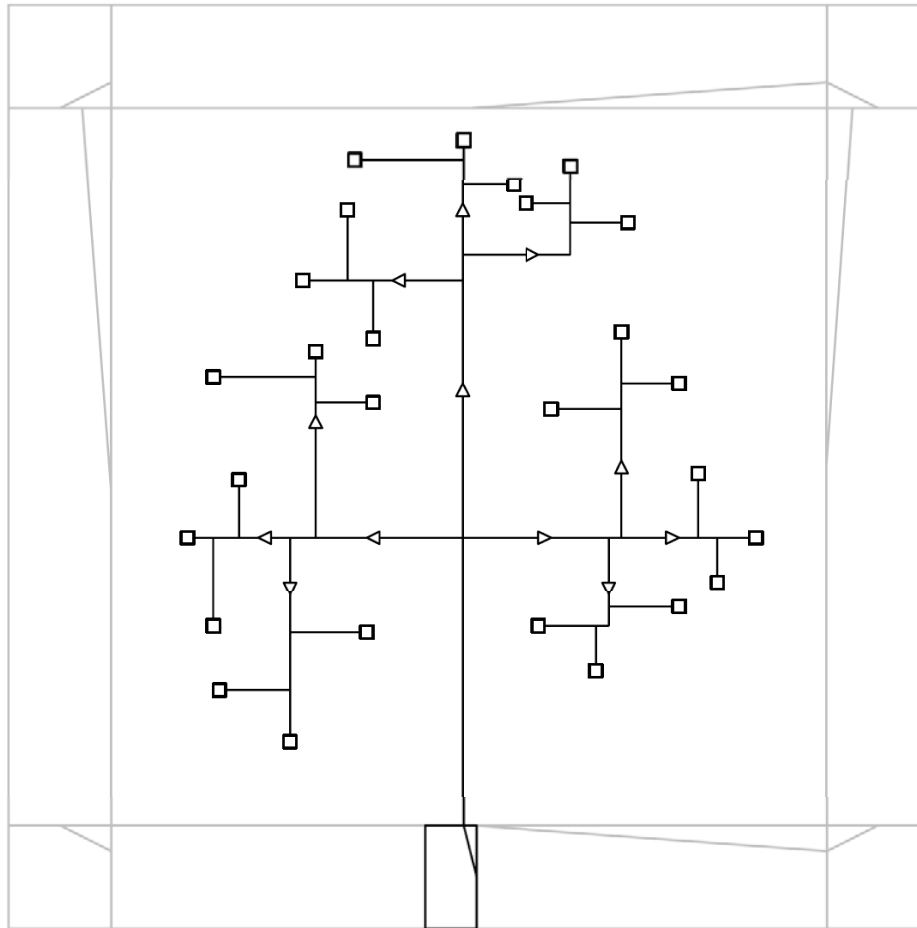
Clock Skew = $20 - 9 = 11$ unit



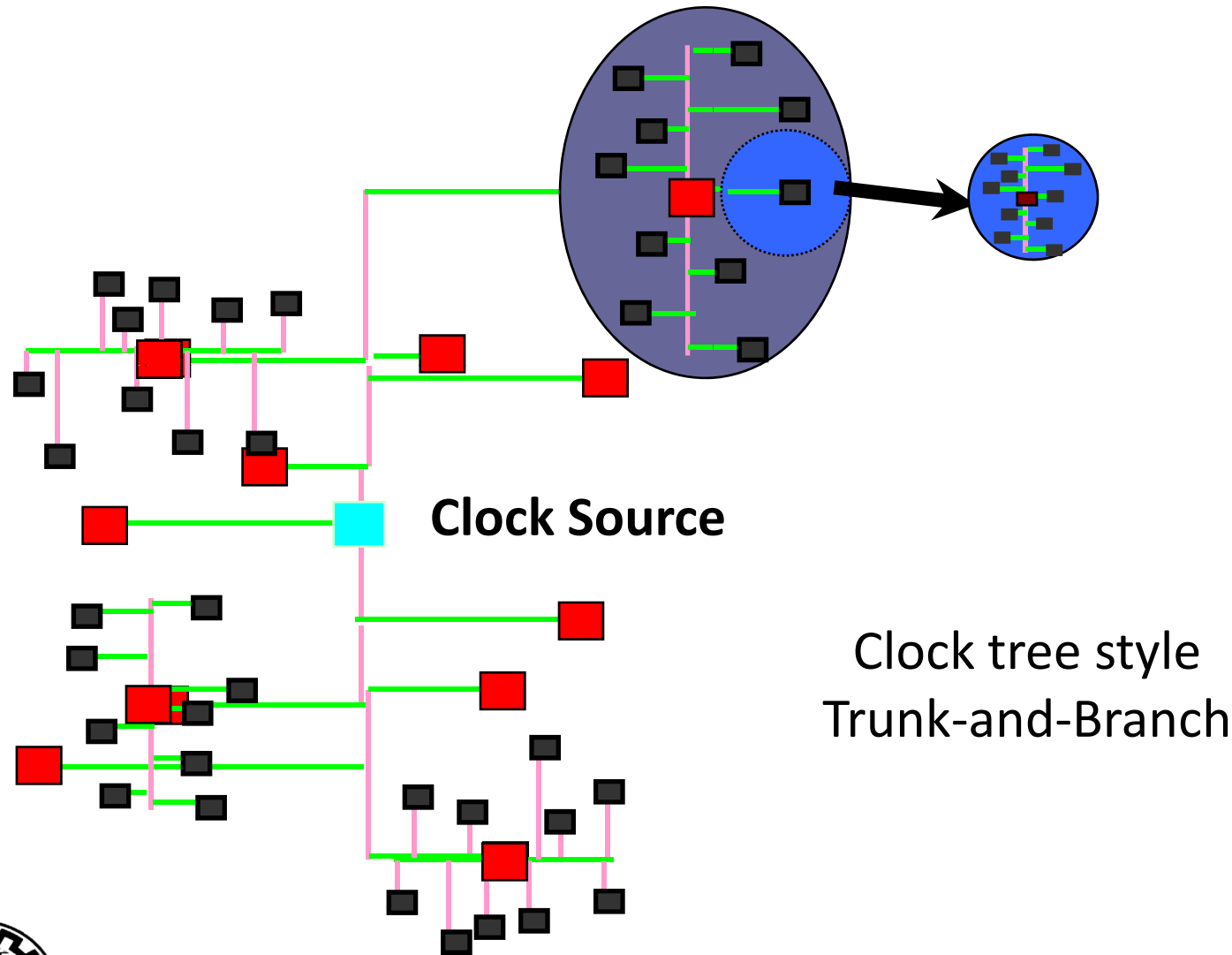
Clock Skew = 0



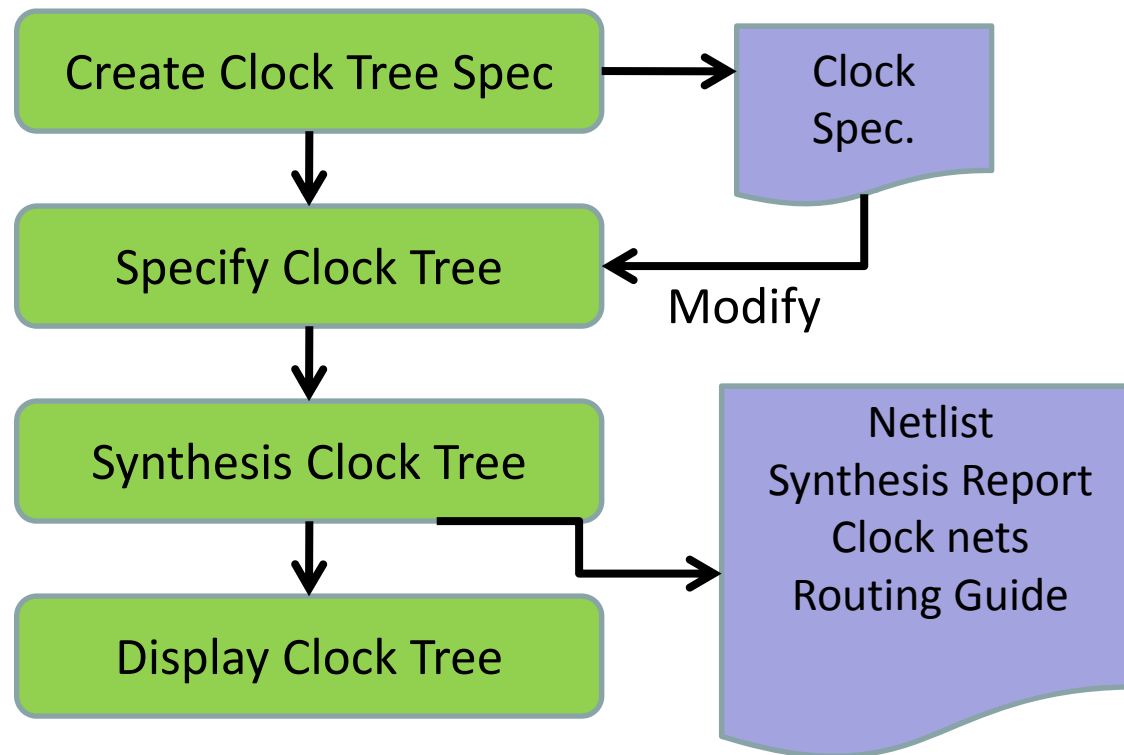
Placement & Routing Steps : Clock Topology



Placement & Routing Steps : Clock Tree Synthesis

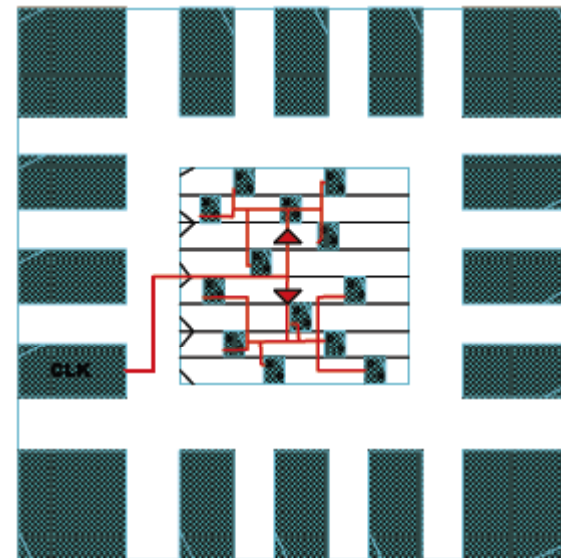
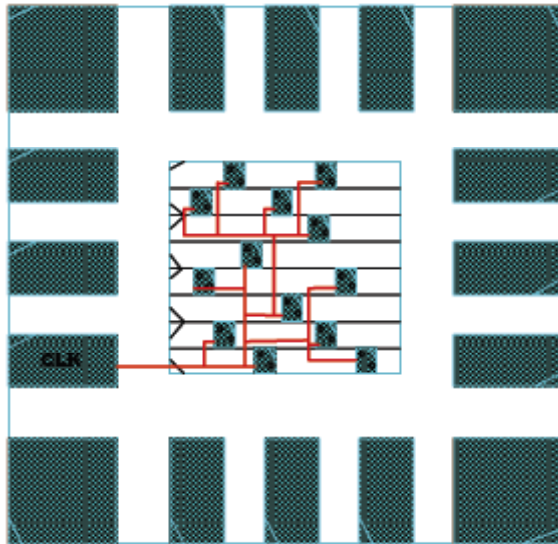
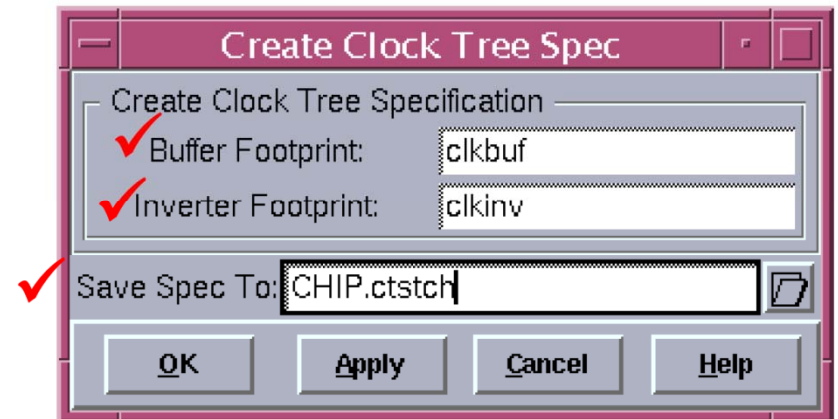


Placement & Routing Steps : Clock Tree Synthesis

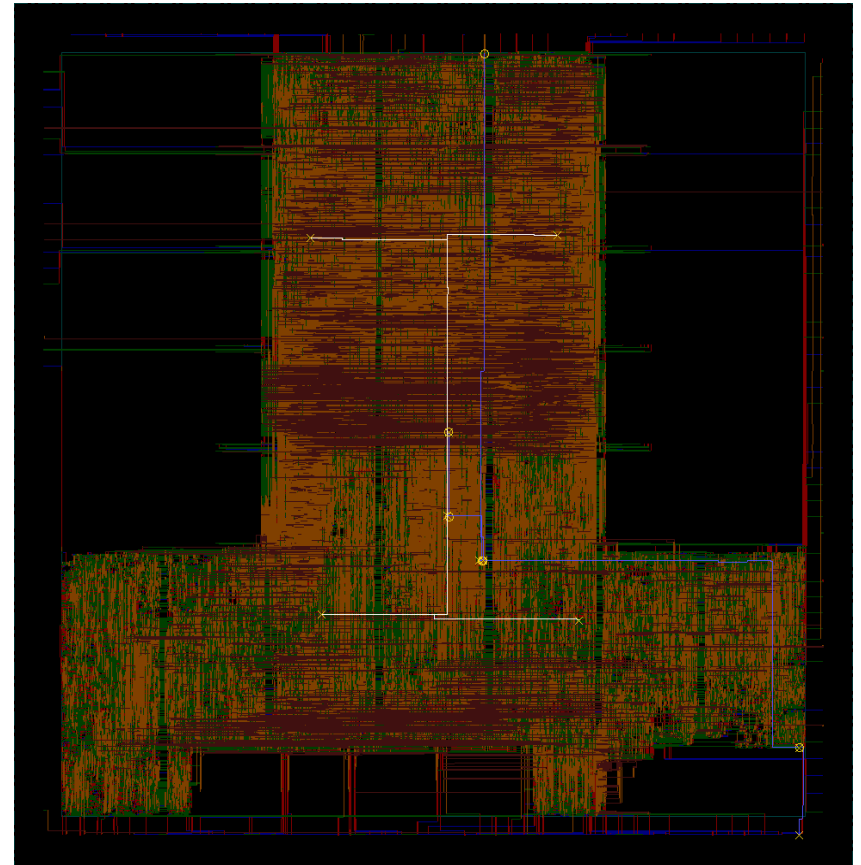
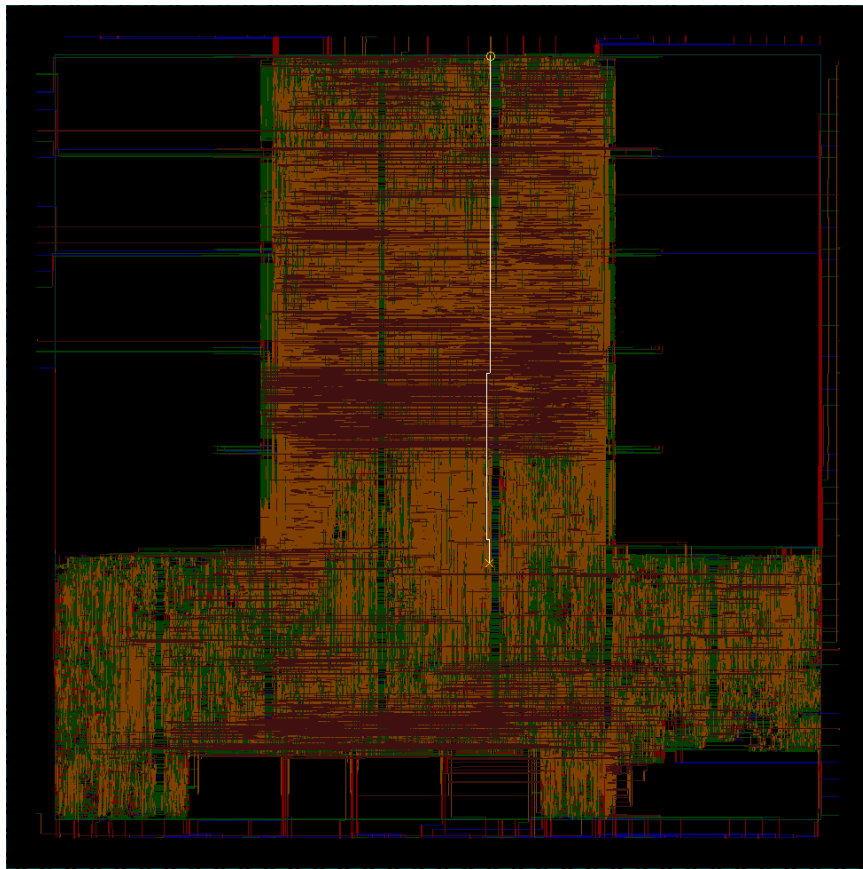


Placement & Routing Steps : Clock Tree Synthesis

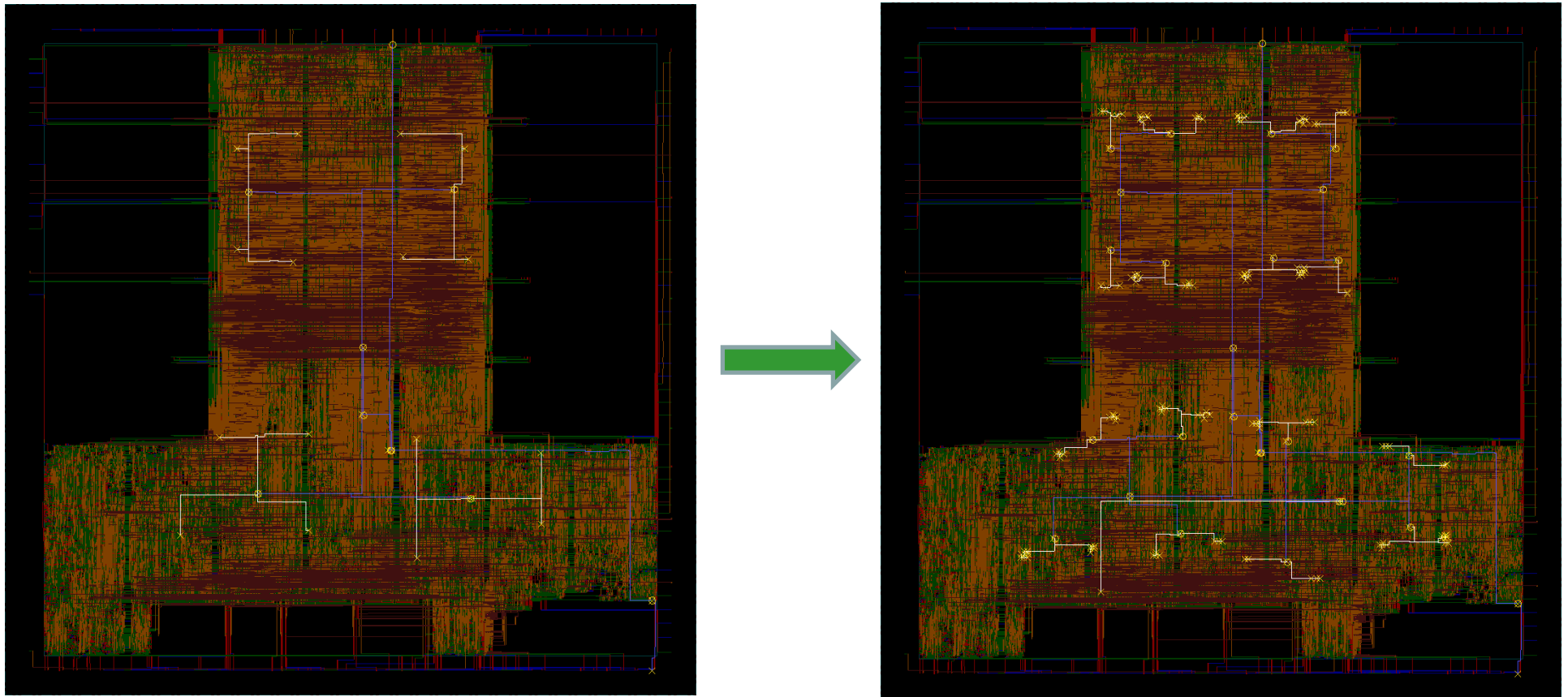
Clock → Design Clock



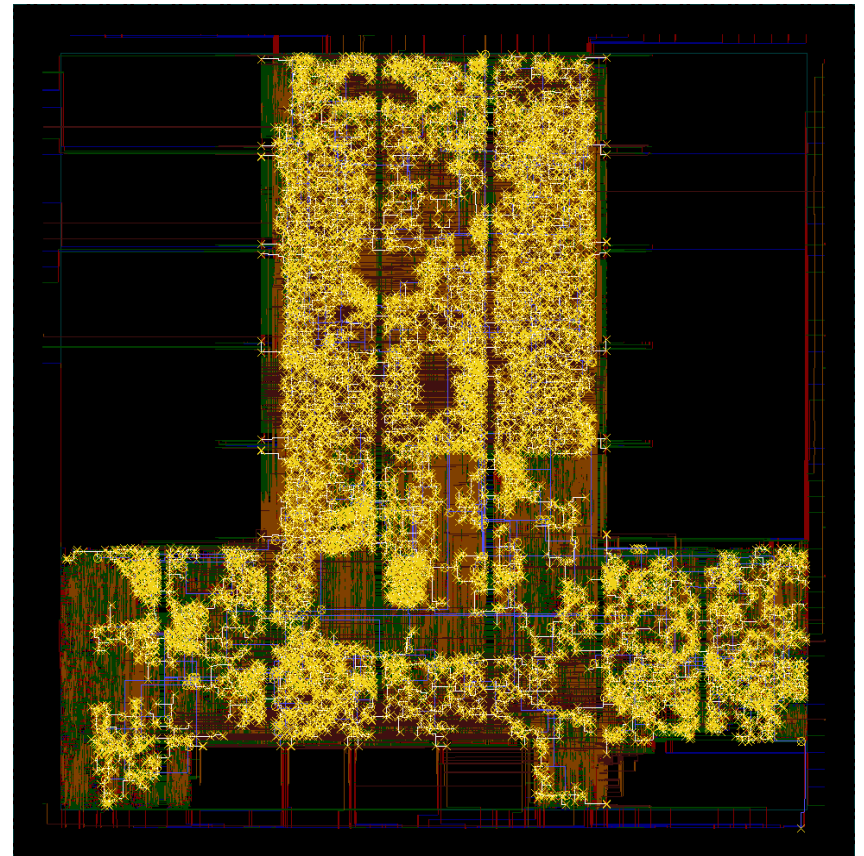
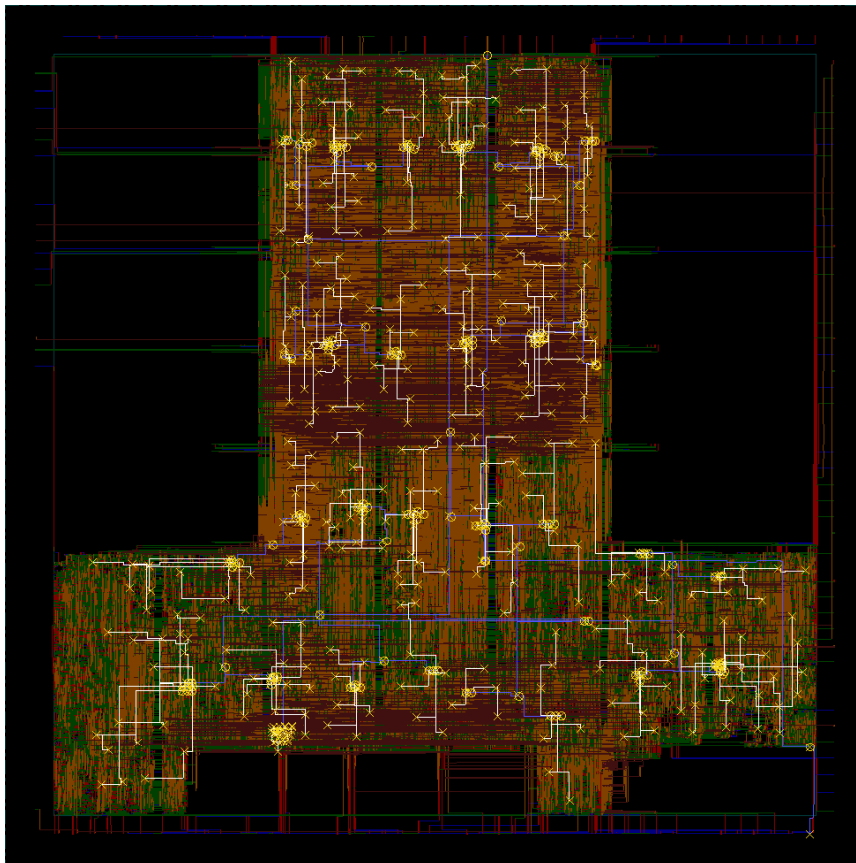
Placement & Routing Steps : Clock Tree Synthesis



Placement & Routing Steps : Clock Tree Synthesis

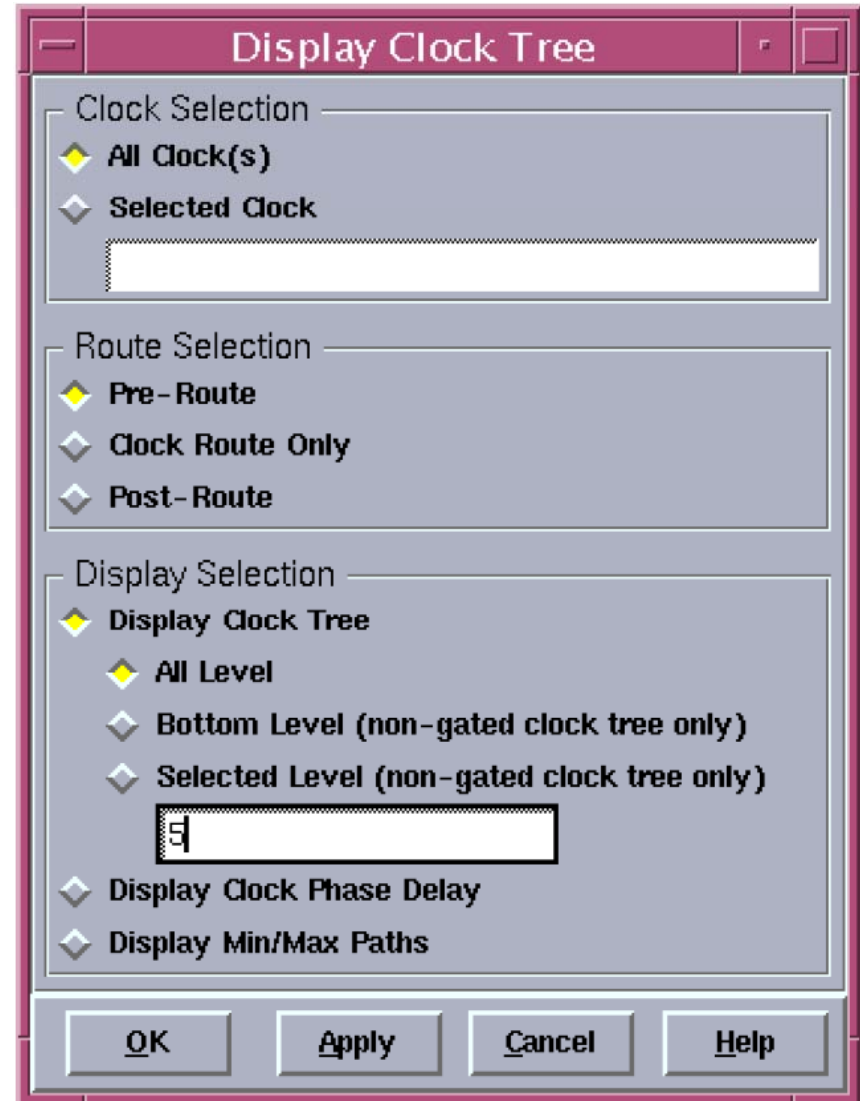


Placement & Routing Steps : Clock Tree Synthesis



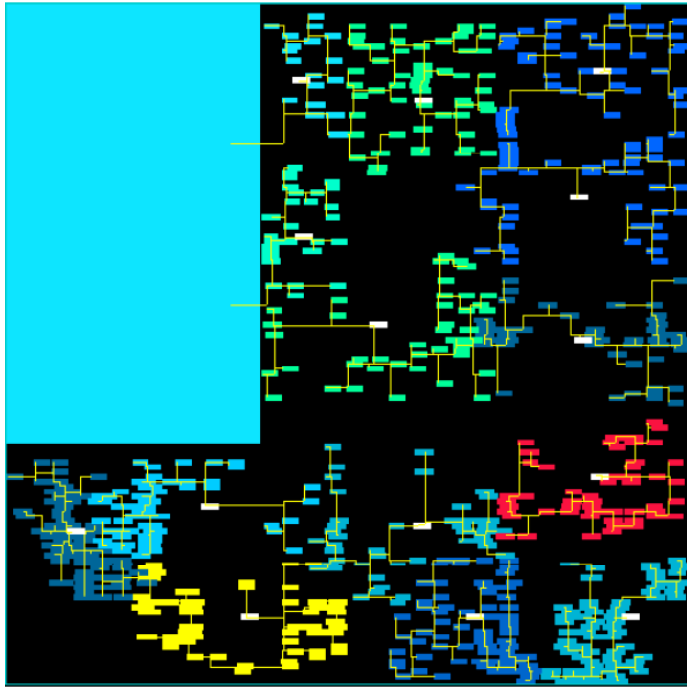
Placement & Routing Steps : Display Clock Tree

Clock → Display → Display Clock Tree

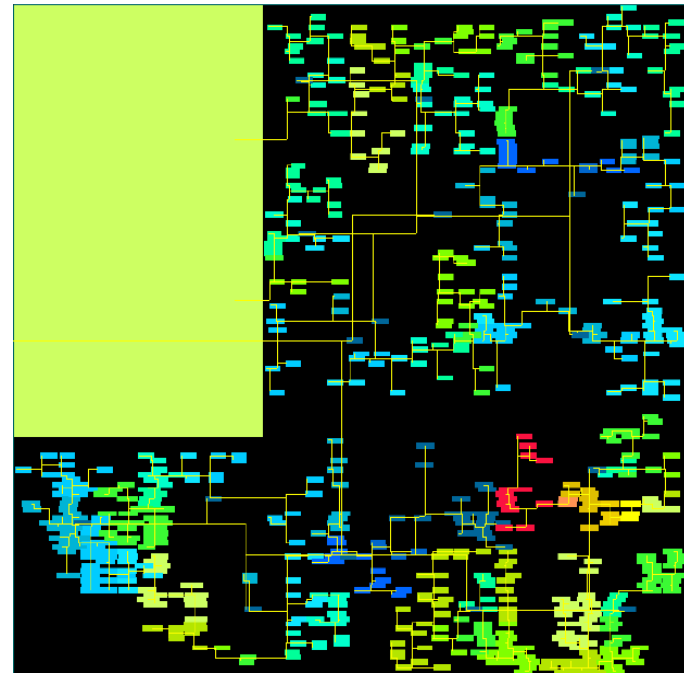


Placement & Routing Steps : Display Clock Tree

Clock → Display → Display Clock Tree



By Level



By Phase Delay

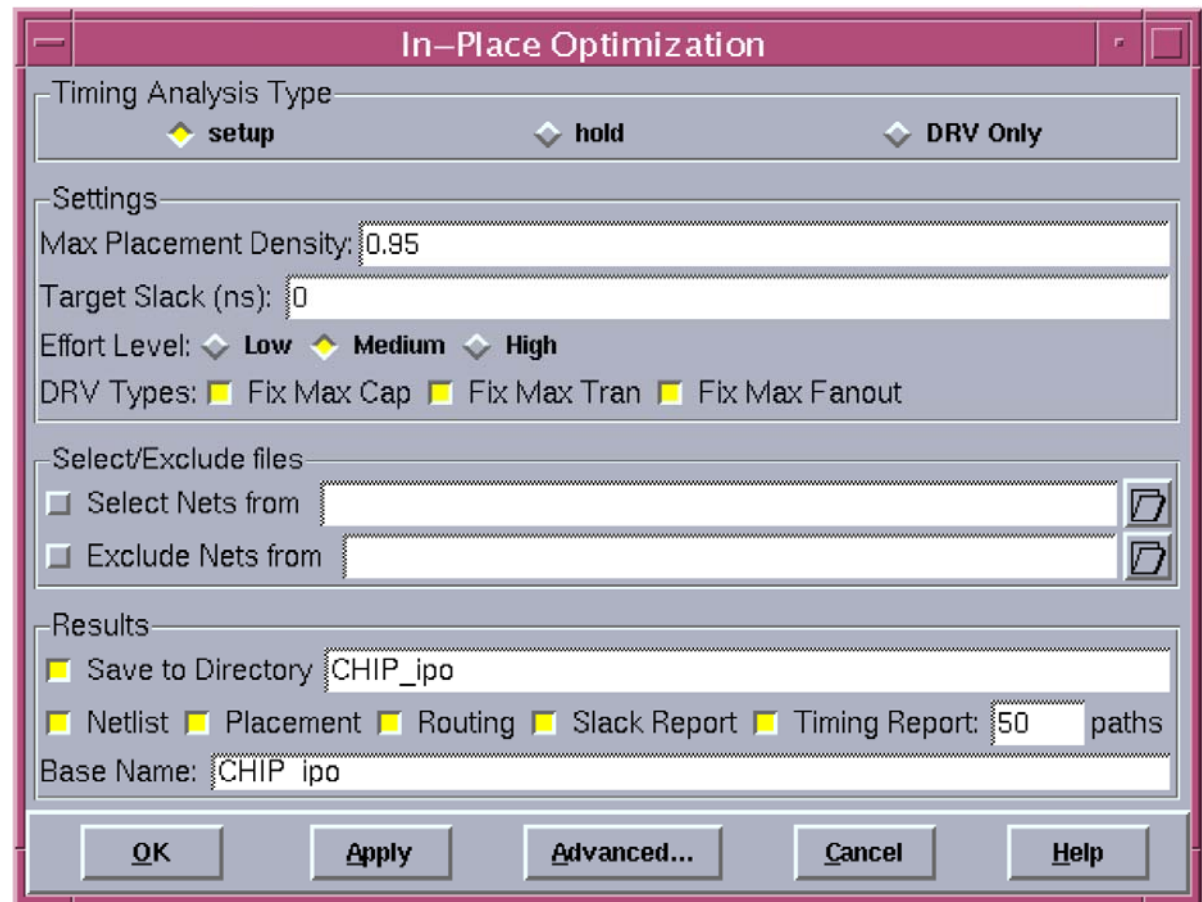


Placement & Routing Steps : Timing Optimization

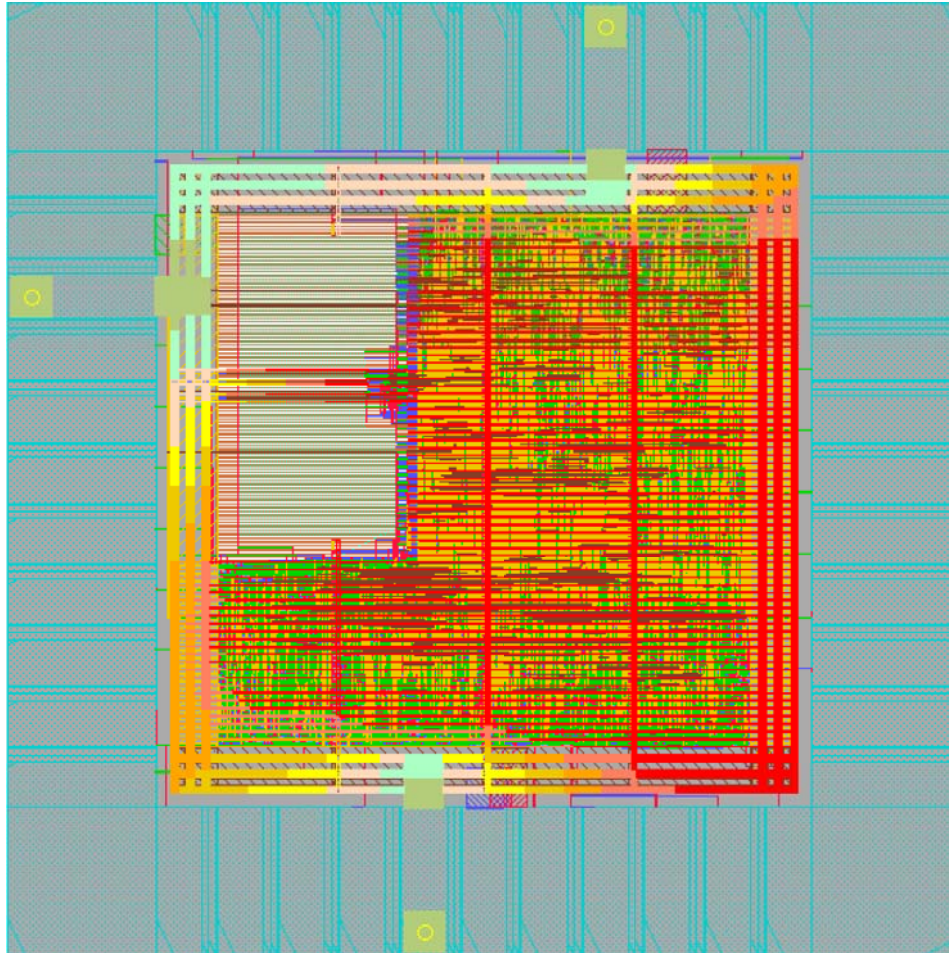
Timing → In-Place Optimization

Used for

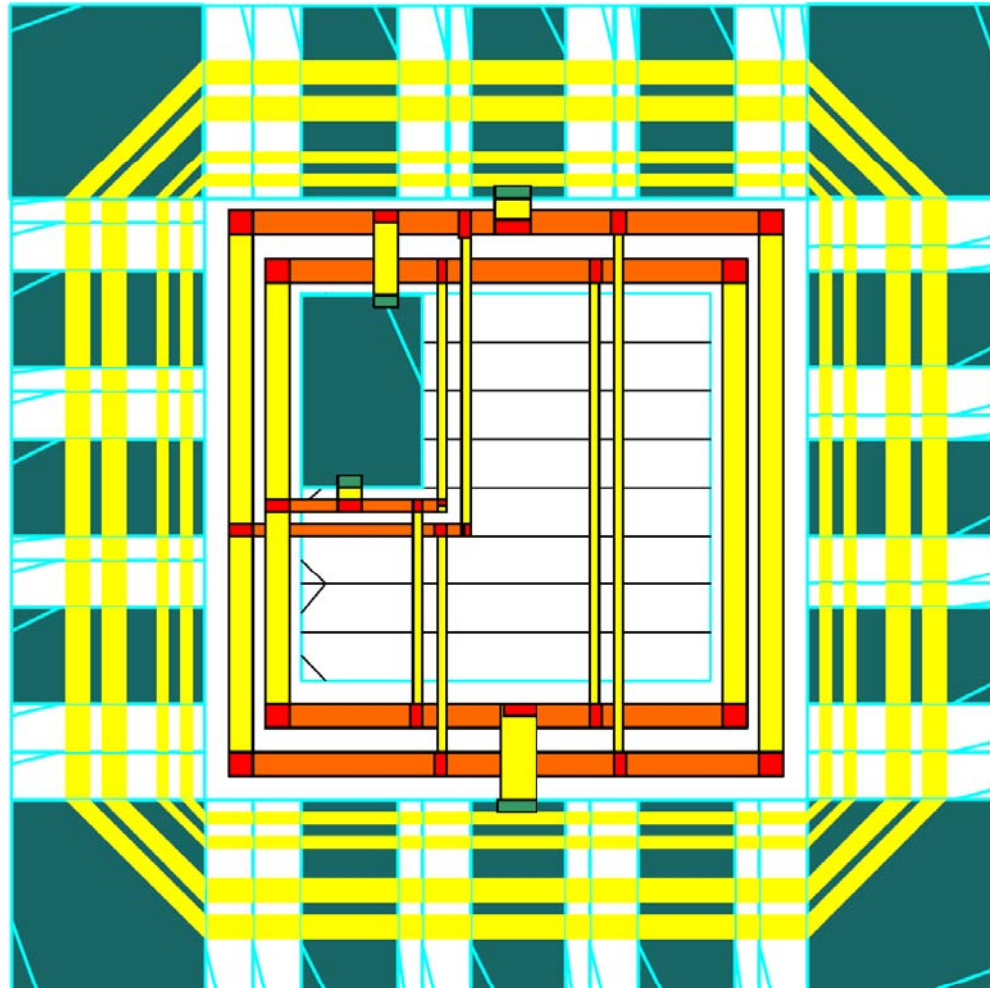
- Setup Time
- Hold Time
- Design Rule Violation



Placement & Routing Steps : Power Analysis

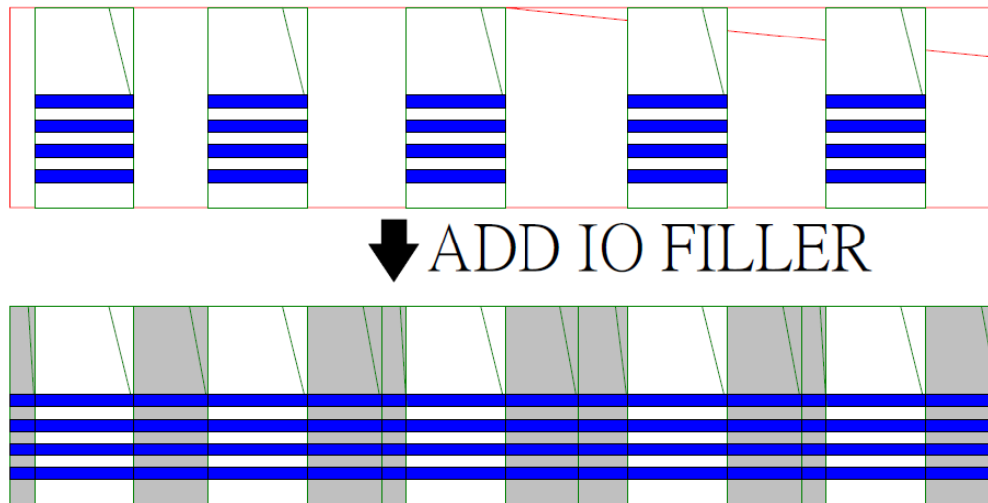


Placement & Routing Steps : Add Filler

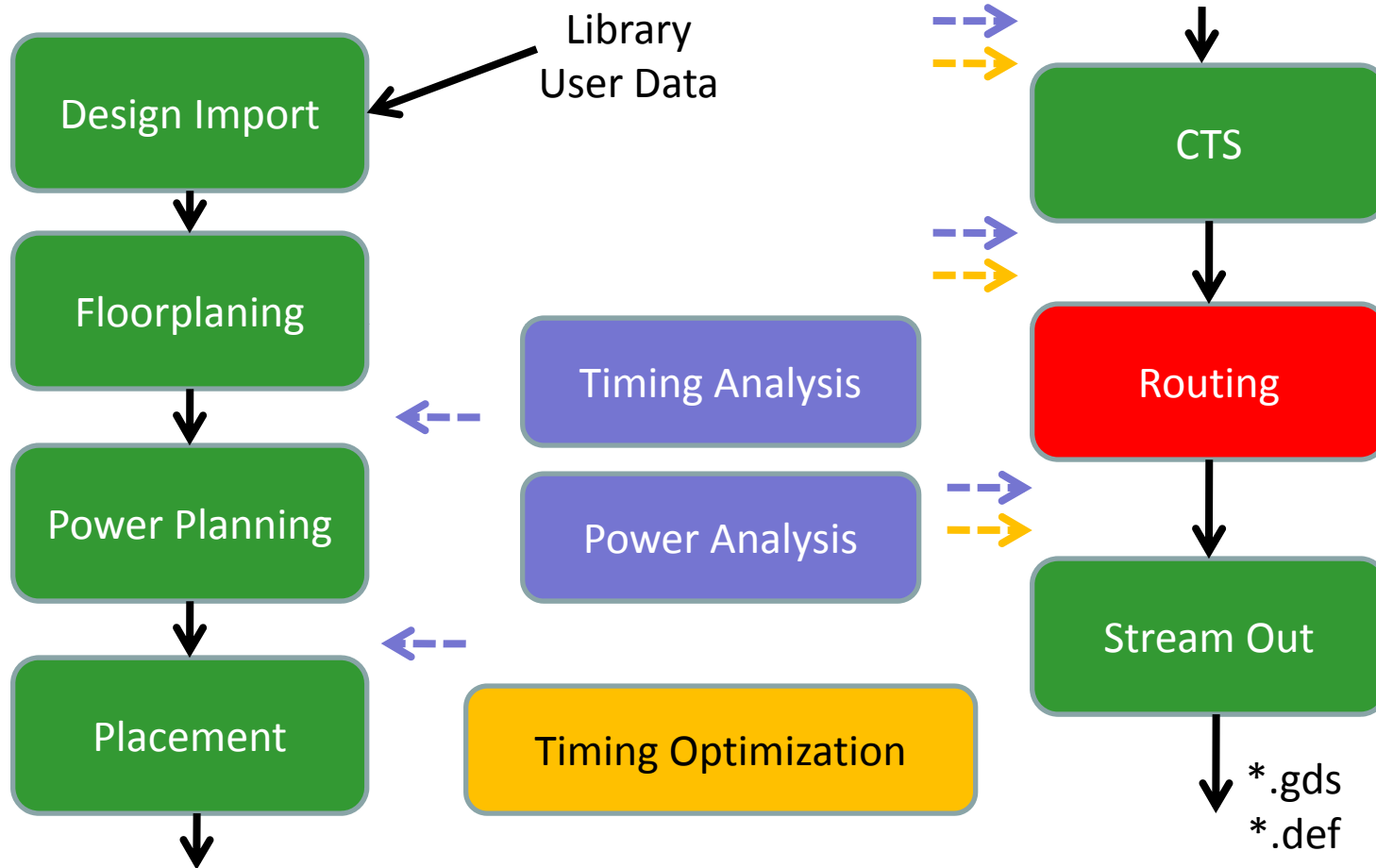


Placement & Routing Steps : Add Filler

- ❑ Connects IO pad power bus by inserting IO filler.
- ❑ Add from wider filler to narrower filler.



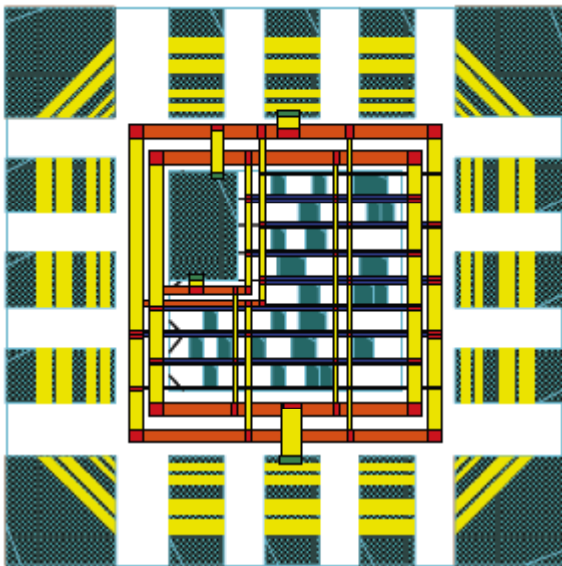
Placement & Routing Steps : Routing



Placement & Routing Steps : Sroute

- به مسیریابی نتهای Power و Ground اصطلاحاً Sroute یا Special Route گفته می شود که عبارتند از مسیریابی:

Route → SRoute

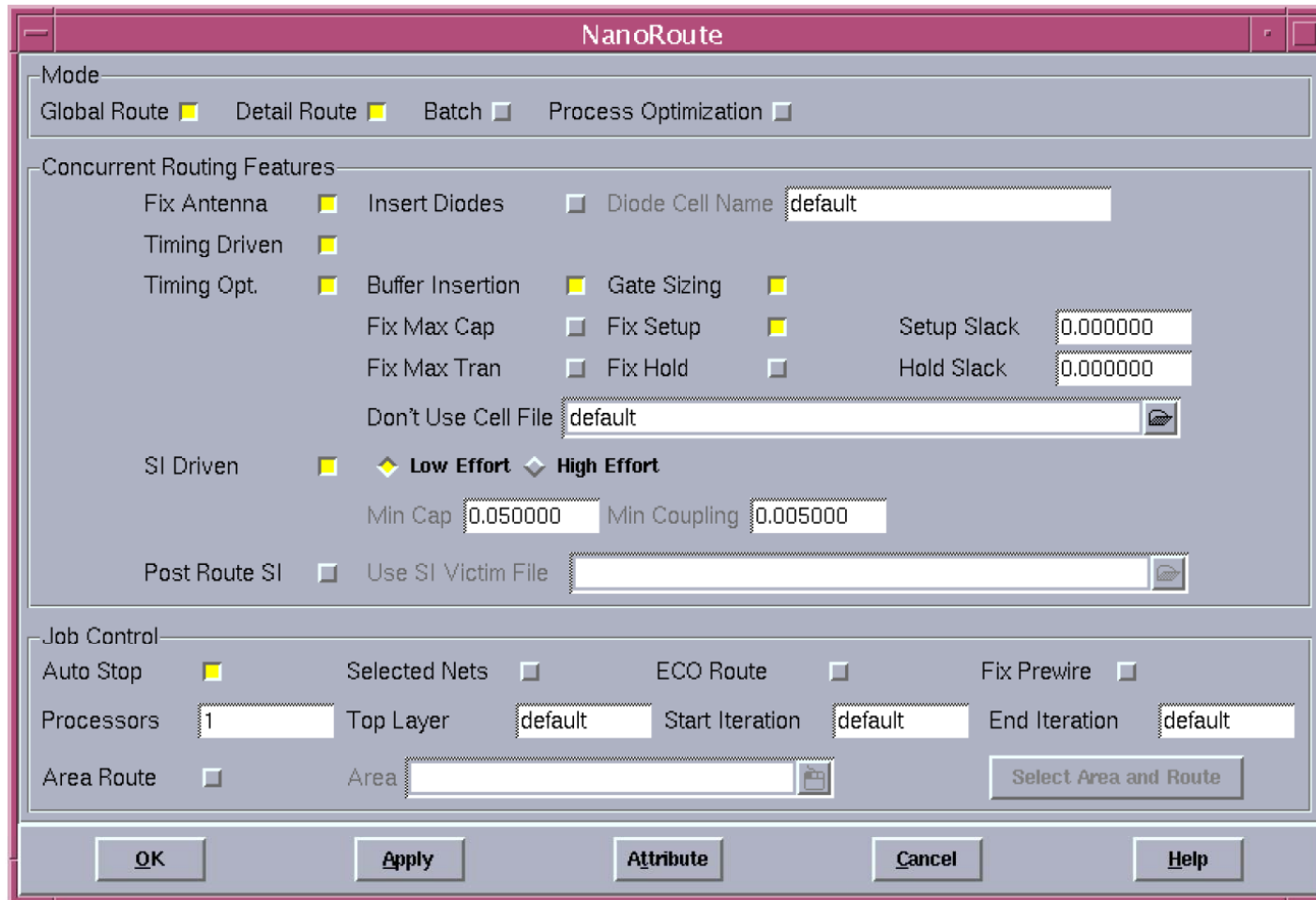


- پینهای توان بلوکهای طرح
- پینهای توان پدها
- پدهای توان
- پینهای توان Standard Cell ها
- Strips



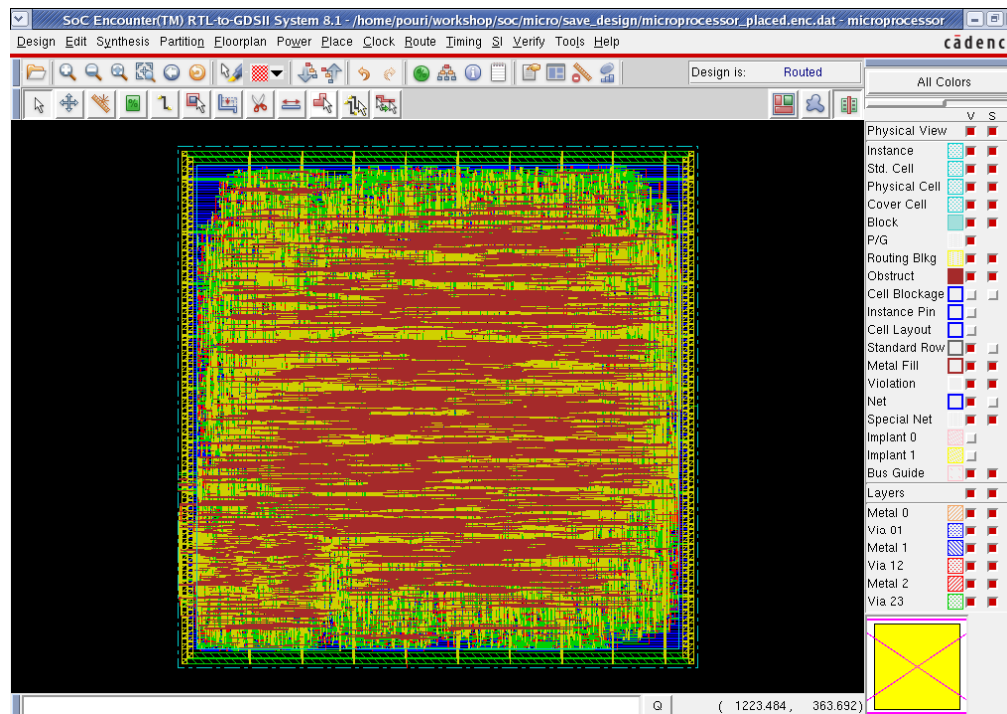
Placement & Routing Steps : Nanoroute

Route → NanoRoute → Route



Placement & Routing Steps : Nanoroute

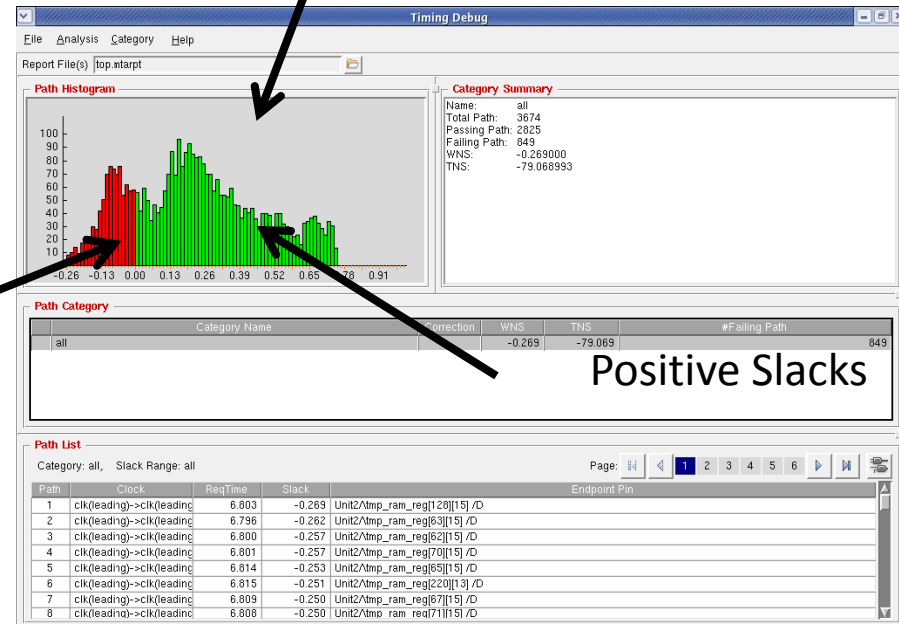
Route → NanoRoute → Route



Placement & Routing Steps : Timing Analysis

1. Timing → Extract RC...
2. Timing → Timing Analysis
3. Timing → Timing Debug → Slack Browser

Histogram Diagram

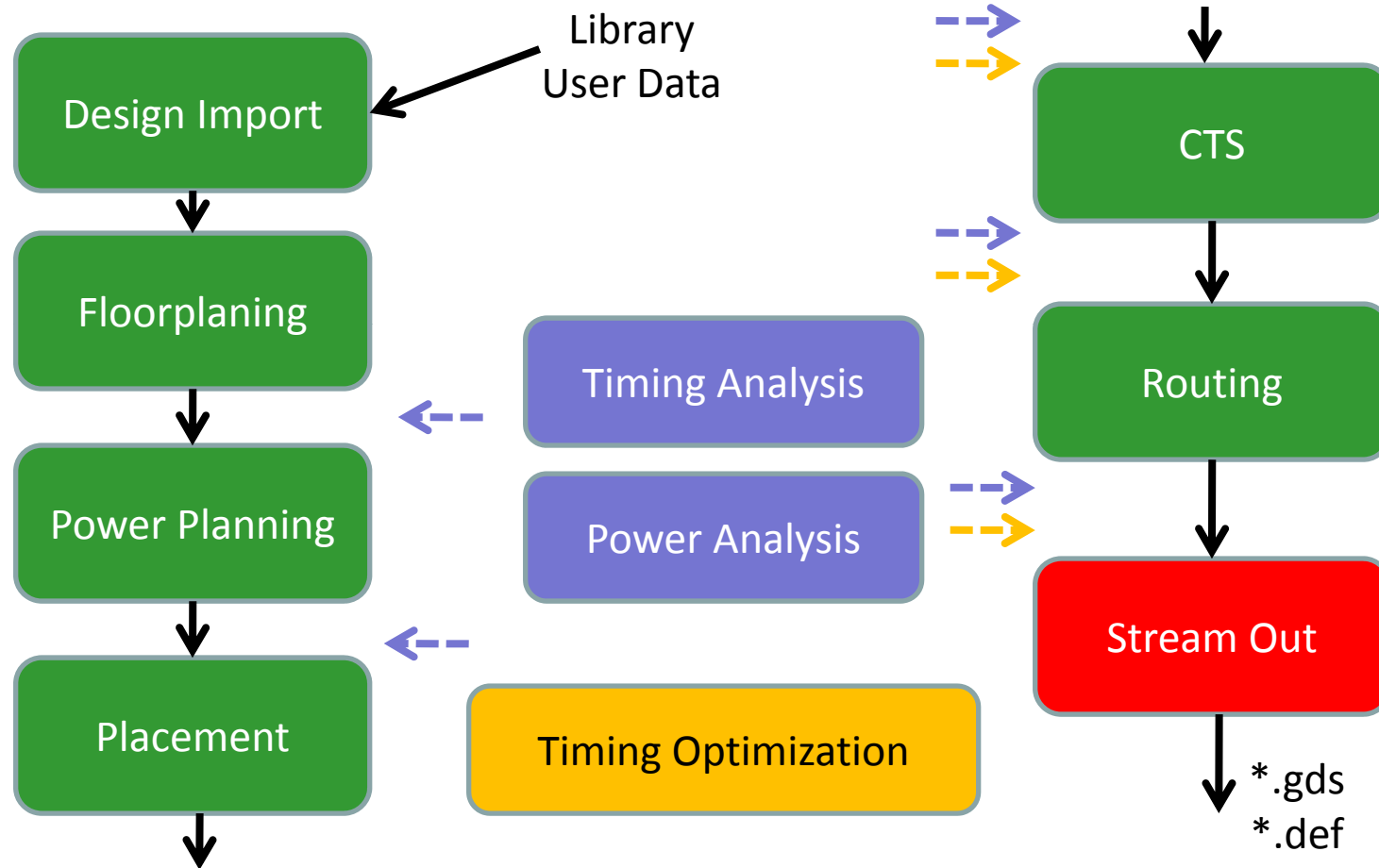


Negative Slacks

Positive Slacks



Placement & Routing Steps : Routing



Placement & Routing Steps : Timing Analysis

طرح لی اوت شده (جانمایی و مسیریابی شده) را به فرمت صنعتی لی اوت یعنی GDSII تبدیل می کند و برای انجام Verification مانند LVS و DRC استفاده می شود. برای Tape-out نیز فایل GDSII برای Fab فرستاده می شود.

1.Design → Save → *GDS*

2.Design → Save → *Netlist*

نت لیست طرح به فرمت v. برای انجام LVS و شبیه سازی Post-layout استخراج می شود.

