Placement timing closure using DCG SPG flow

Zhaoqing xi zqxi@marvell.com

Marvell

ABSTRACT

This article illustrate how to use DCG SPG flow for timing closure at placement stage in ICC and also provide timing correlation check between DCG and ICC.

1. Introduction

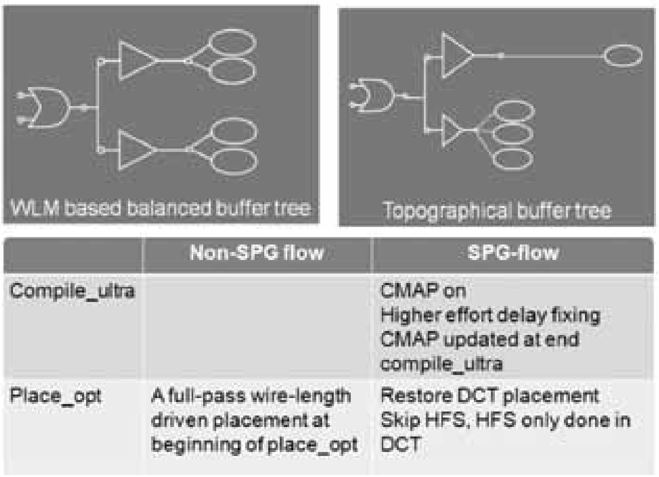
In traditional synthesis flow, there always has correlation problem between synthesis and physical placement. Timing after placement is not comparable with timing after synthesis. The reason is:

i. Wire load model is not precise enough for RC estimation during synthesis, this is more un-acceptable for 40nm and more advanced process;

ii. Once cells are placed, critical timing path may be different as critical path at synthesis stage. PnR tool can't optimize it well enough and synthesis tool don't take it as critical. So physical-aware synthesis is here to bridge this gap and make a total turnaround time shorter.

2. Physical aware synthesis using Synopsys physical guidance flow

Physical aware synthesis is not new to us, and SPG flow makes it better. It is fully aware of floorplan physical information through DEF: all hard macros' location, all placement blockages and so forth. It is jointly supported both in Design Compiler and IC Compiler in order to improve runtime and correlation. Placement is run at logic synthesis stage, not until physical synthesis in ICC. By doing so, the critical path is same as we see after placement done in ICC. For example, if some nets need buffering due to DRC violation, DC will add the buffer and optimize the whole path.

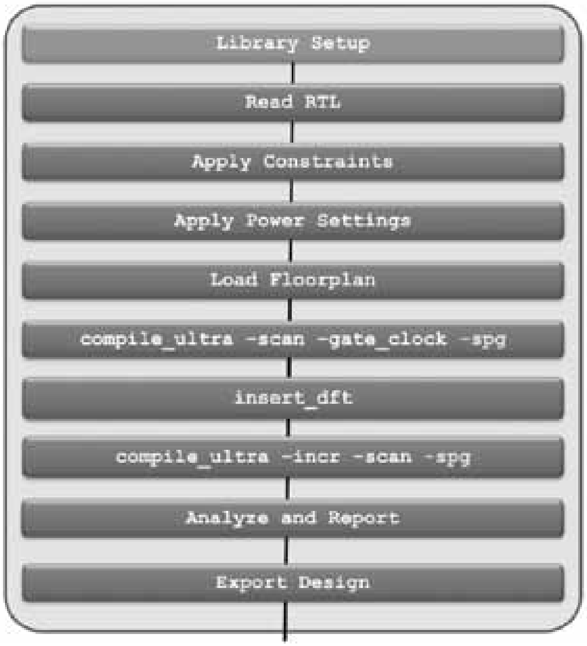


	Non-SPG flow	SPG-flow
Compile_ultra		CMAP on Higher effort delay fixing CMAP updated at end compile_ultra
Place_opt	A full-pass wire-length driven placement at beginning of place_opt	Restore DCT placement Skip HFS, HFS only done in DCT

Picture1 difference between wlm and topo based synthesis

2.1 DCG flow introduction:

We can see the flow is pretty similar with traditional DC flow. The UI is user-friendly. All we need to do is just adding –SPG option during compile. However, we need keep the ideal of physical guidance in mind. The followings are some notable places:


Library Setup
Read RTL
Apply Constraints
Apply Power Settings
Load Floorplan
compile_ultra -scan -gate_clock -spg
insert_dft
compile_ultra -incr -scan -spg
Analyze and Report
Export Design

Picture2 basic flow of the DCG-SPG

2.1.1 Database reference consistence

In DCG-SPG create milkyway with the same data and library for ICC:

i. ech file

ii. link library

iii. Milkyway reference library

iv. Tluplus file

2.1.2 Floorplan consistence

Floorplan information is dumped out by ICC using DEF format. So we need to finish a completed floorplan in ICC, than write DEF file. About each floorplan element:

i. Memories and IPs.

ii. Power switches.

iii. Endcaps, tap cell, and other physical only cells(power switch, etc.). They are part of floorplan for 40nm and more advanced technology node. Since there is no endcap or tap cell in RTL, we need add all these physical only cells as we did it in ICC. In DCG-SPG, once DEF is read, those physical only cells all having physical information the same as ICC.

iv. Use extract_physical_constraints for importing DEF. Check the log file to see any physical information missed during importing.

v. Create voltage area using tcl since VA info in not provide in DEF.

vi. Using create_placement_blockage -type soft and create_placement_blockage –type hard specifically for adding placement blockage. DEF may not differentiate hard/soft placement blockage.

2.1.3 Environment consistence

For the synthesis, must make the environments match.

i. Using exactly the same floorplan def and SDC constraints for ICC and

DCG-SPG

ii. Using ddc or milkyway as the start point of the ICC.

iii. Using place_opt -SPG in ICC flow respectively. And avoid using remove_buffer_tree and create_ buffer_tree before place_opt.

iv. extract_rc -estimate before report timing, to make DCG-SPG estimate RC delay. Keep the same variable setting on variable complete_mixed_mode_extraction to get better correlation.

v. Using the same dont_use lib-cell list and using the same dont_touch instance list.

vi. Using set_ignored_layers for routing layer constraints, it should consistence with ICC. Because it affects the global route and congestion evaluation.

Once all physical guidance information is provide into DC, just run compile flow as shown in previous figure.

2.2 data exchange

Once we have done DCG, we need to transfer data to ICC for placement. There are couples of choice:

i. Netlist/DEF

ii. ddc/DEF

iii. milkyway

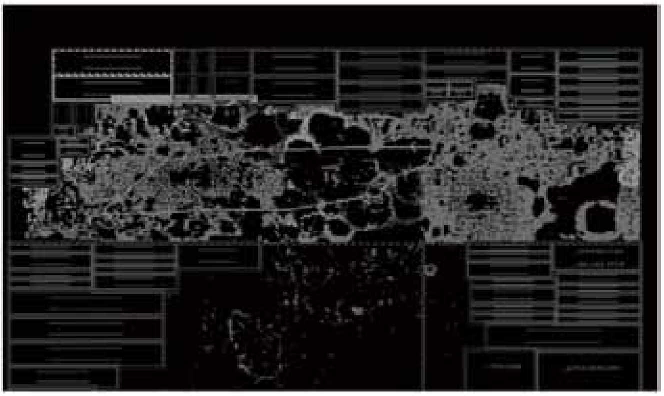
Here we choose Milkyway database as the start point, since it is easy to use. Launch the same ICC version in DC console using:

set_icc_dp_options -icc_executable

start_icc_dp

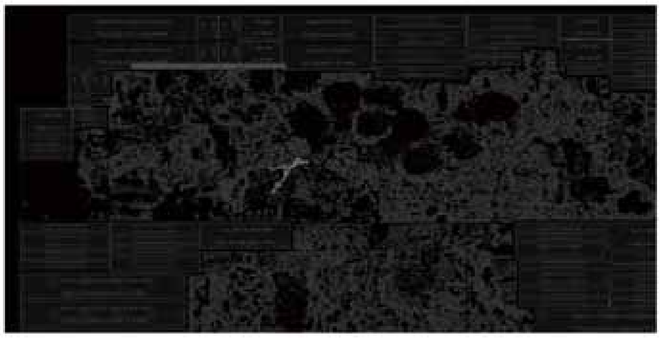
2.3 Why DCG?

For the first trail, use DC-ICC flow. The after place_opt WNS path is shown below. WNS path is impossible to close. Also, we can see cells along critical path are spread apart far away. There is no way we can move since even psynopt can't improve timing of paths with big WNS significantly.



Picture3 DC-ICC flow WNS path overview

Then switched to DCG-ICC flow for the trail, the WNS path after place_opt in ICC is shown below. We can see cells along critical path are put together. And WNS is reasonable to move on.

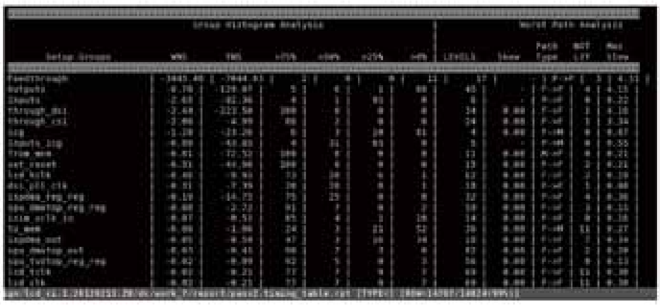


Picture4 DCG-SPG-ICC flow the same path overview

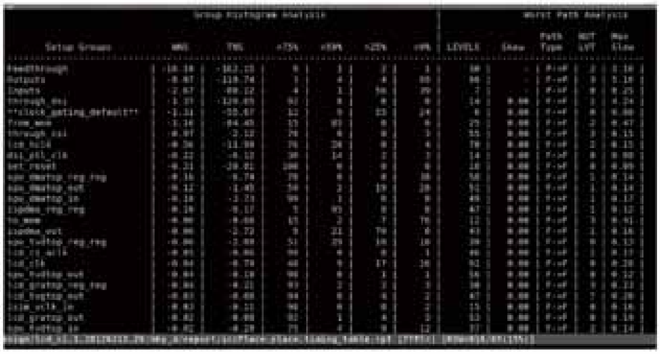
2.4 correlation check

2.4.1 WNS correlation

Here is the timing report after DCG-SPG and ICC synthesis



Picture5 timing QoR after DCG-SPG



Picture6 Timing QoR after place_opt in ICC

Now let's pick up some path groups of interest for WNS check. Criteria: delta slack/period

From_mem: -0.81-> -1.16

Correlation: (-1.16+0.81)/2.816 = -0.1243 (degrade 12%)

Ispdma_reg_reg: -0.19 -> -0.1

Correlation: (-0.1 + 0.19)/2.4 = 0.0375 (improve 3%)

Spu_dmatop_reg_reg: -0.08 -> -0.16

Correlation: (-0.16 + 0.08)/2.816=-0.0284 (improve 2%)

Spu_tvdtop_reg_reg: -0.02 -> -0.06

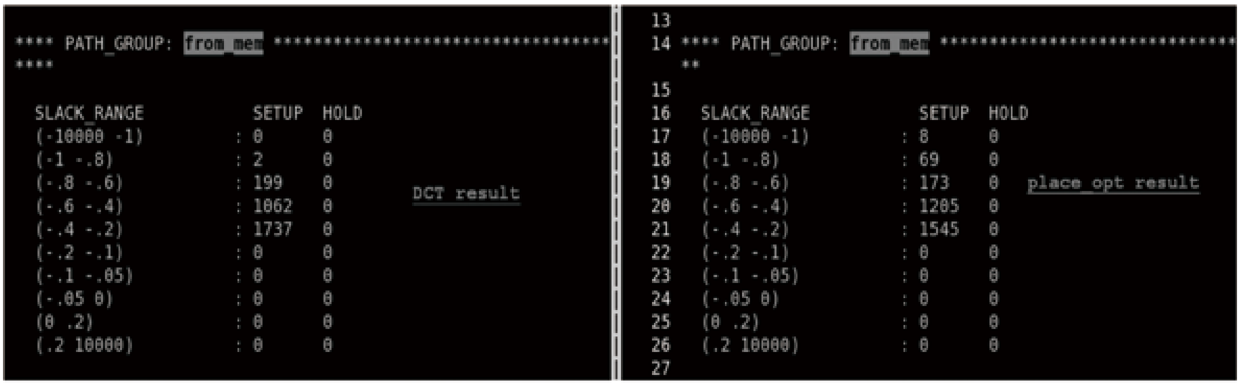
Correlation: (-0.06+0.02)/2.816=-0.0142 (degrade 1%)

Summary

There still being some differences between ICC placement timing and DCG synthesis result since DCG uses virtual placer and does not consider cell overlapping while ICC does real placement as well as checking cell legalization, but, we can see timing of most path group deviate not too much, within 5%. Only WNS of path group from_mem degrade about ~12%. But the total violating path number of From_mem is not changed significantly. So we take it acceptable and move on.

2.4.2 TNS correlation

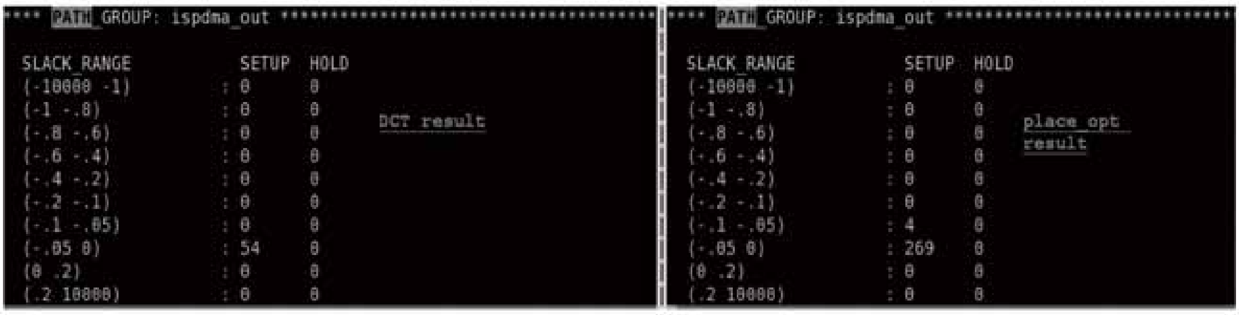
Here is comparison of TNS for path groups of interest. For each path group, we chose worst 3000 path for correlation check.



```
**** PATH_GROUP: from_mem ************************************
****

  SLACK_RANGE           SETUP  HOLD
  (-10000 -1)         : 0      0
  (-1 -.8)            : 2      0
  (-.8 -.6)           : 199    0
  (-.6 -.4)           : 1062   0          DCT result
  (-.4 -.2)           : 1737   0
  (-.2 -.1)           : 0      0
  (-.1 -.05)          : 0      0
  (-.05 0)            : 0      0
  (0 .2)              : 0      0
  (.2 10000)          : 0      0
```

```
**** PATH_GROUP: from_mem ********************************
**

  SLACK_RANGE           SETUP  HOLD
  (-10000 -1)         : 8      0
  (-1 -.8)            : 69     0
  (-.8 -.6)           : 173    0   place_opt result
  (-.6 -.4)           : 1205   0
  (-.4 -.2)           : 1545   0
  (-.2 -.1)           : 0      0
  (-.1 -.05)          : 0      0
  (-.05 0)            : 0      0
  (0 .2)              : 0      0
  (.2 10000)          : 0      0
```

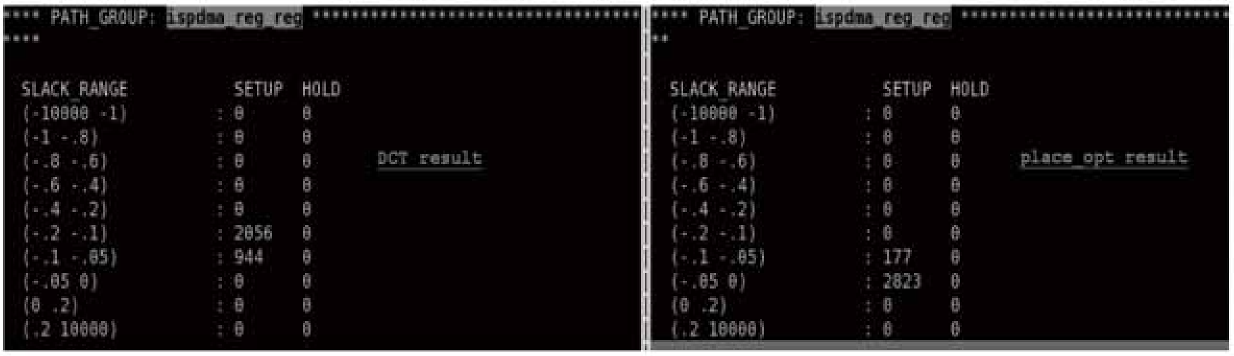
Picture7 Path group: from_mem



```
**** PATH_GROUP: ispdma_out ***************************************

 SLACK_RANGE           SETUP  HOLD
 (-10000 -1)         : 0      0
 (-1 -.8)            : 0      0
 (-.8 -.6)           : 0      0       DCT result
 (-.6 -.4)           : 0      0
 (-.4 -.2)           : 0      0
 (-.2 -.1)           : 0      0
 (-.1 -.05)          : 0      0
 (-.05 0)            : 54     0
 (0 .2)              : 0      0
 (.2 10000)          : 0      0
```

```
**** PATH_GROUP: ispdma_out *********************************

 SLACK_RANGE           SETUP  HOLD
 (-10000 -1)         : 0      0
 (-1 -.8)            : 0      0
 (-.8 -.6)           : 0      0    place_opt
 (-.6 -.4)           : 0      0    result
 (-.4 -.2)           : 0      0
 (-.2 -.1)           : 0      0
 (-.1 -.05)          : 4      0
 (-.05 0)            : 269    0
 (0 .2)              : 0      0
 (.2 10000)          : 0      0
```

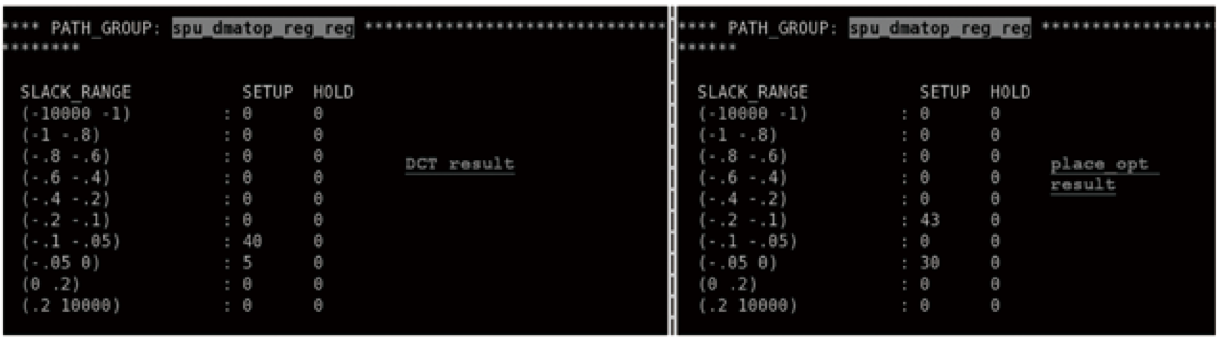
Picture8 Path group: ispdma_out



```
**** PATH_GROUP: ispdma_reg_reg ***********************************
****

 SLACK_RANGE           SETUP  HOLD
 (-10000 -1)         : 0      0
 (-1 -.8)            : 0      0
 (-.8 -.6)           : 0      0       DCT result
 (-.6 -.4)           : 0      0
 (-.4 -.2)           : 0      0
 (-.2 -.1)           : 2056   0
 (-.1 -.05)          : 944    0
 (-.05 0)            : 0      0
 (0 .2)              : 0      0
 (.2 10000)          : 0      0
```

```
**** PATH_GROUP: ispdma_reg_reg ******************************
**

 SLACK_RANGE           SETUP  HOLD
 (-10000 -1)         : 0      0
 (-1 -.8)            : 0      0
 (-.8 -.6)           : 0      0       place_opt result
 (-.6 -.4)           : 0      0
 (-.4 -.2)           : 0      0
 (-.2 -.1)           : 0      0
 (-.1 -.05)          : 177    0
 (-.05 0)            : 2823   0
 (0 .2)              : 0      0
 (.2 10000)          : 0      0
```

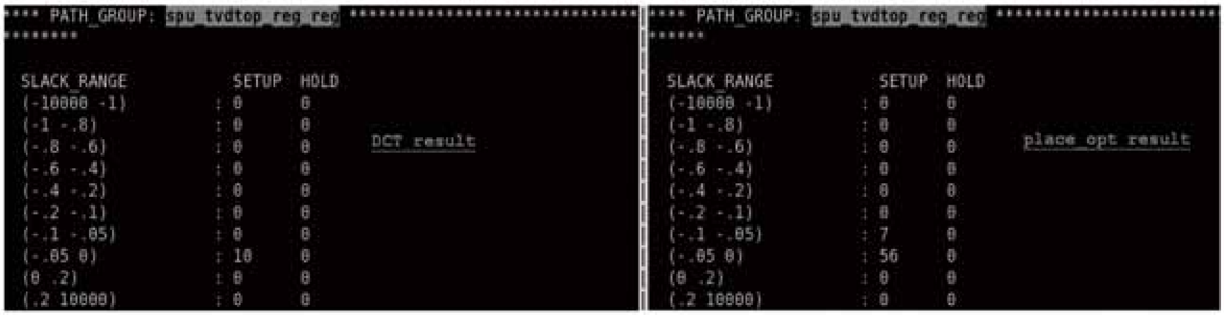
Picture9 Path group: ispdma_reg_reg



```
**** PATH_GROUP: spu_dmatop_reg_reg ********************************
********

 SLACK_RANGE            SETUP  HOLD
 (-10000 -1)          : 0      0
 (-1 -.8)             : 0      0
 (-.8 -.6)            : 0      0        DCT result
 (-.6 -.4)            : 0      0
 (-.4 -.2)            : 0      0
 (-.2 -.1)            : 0      0
 (-.1 -.05)           : 40     0
 (-.05 0)             : 5      0
 (0 .2)               : 0      0
 (.2 10000)           : 0      0
```

```
**** PATH_GROUP: spu_dmatop_reg_reg ******************
******

 SLACK_RANGE            SETUP  HOLD
 (-10000 -1)          : 0      0
 (-1 -.8)             : 0      0
 (-.8 -.6)            : 0      0
 (-.6 -.4)            : 0      0        place_opt
 (-.4 -.2)            : 0      0        result
 (-.2 -.1)            : 43     0
 (-.1 -.05)           : 0      0
 (-.05 0)             : 30     0
 (0 .2)               : 0      0
 (.2 10000)           : 0      0
```

Picture10 Path group: spu_dmatop_reg_reg



```
**** PATH_GROUP: spu_tvdtop_reg_reg ********************************
********

 SLACK_RANGE            SETUP  HOLD
 (-10000 -1)          : 0      0
 (-1 -.8)             : 0      0
 (-.8 -.6)            : 0      0        DCT result
 (-.6 -.4)            : 0      0
 (-.4 -.2)            : 0      0
 (-.2 -.1)            : 0      0
 (-.1 -.05)           : 0      0
 (-.05 0)             : 10     0
 (0 .2)               : 0      0
 (.2 10000)           : 0      0
```

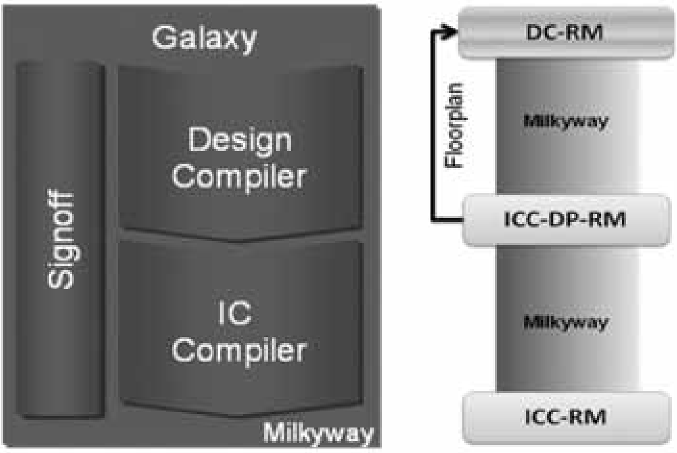
```
**** PATH_GROUP: spu_tvdtop_reg_reg ************************
******

 SLACK_RANGE            SETUP  HOLD
 (-10000 -1)          : 0      0
 (-1 -.8)             : 0      0        place opt result
 (-.8 -.6)            : 0      0
 (-.6 -.4)            : 0      0
 (-.4 -.2)            : 0      0
 (-.2 -.1)            : 0      0
 (-.1 -.05)           : 7      0
 (-.05 0)             : 56     0
 (0 .2)               : 0      0
 (.2 10000)           : 0      0
```

Picture11 Path group: spu_tvdtop_reg_reg

Summary:

Based on comparison of those path groups of interest, we can see good correlation of TNS (violation path number) between DCG result and place_opt result. There is also one thing to notice, for path group ispdma_reg_reg, a big improvement is observed after place_opt. After DCG, number of paths with slack which worse than -0.1ns is about 2000, but after place_opt, this number dropped dramatically to 0. Most violation path is having slack less than -0.05ns. This improvement is quiet impressive.

2. 5 flow recommendation



Picture12 recommended RM flow

In addition, if the design is build with RM-ICC RM-DCG flow, we can share the configuration and it save the runtime as well as reduce the possibility of the environment mismatch. So we recommended using this flow.

3. Conclusion and summary

Design Compiler Topographical synthesis solution perform virtual placement driven design mapping and optimization in order to achieve good QoR result that offer good correlation with IC Compiler. Although current DCG flow does not require user to provide any physical constraints, however DCG/ICC correlation can be further improved if user drive DCG with the same physical constraints being used in ICC.

It is obviously that DCG helps a lot on timing closure. For some cases, traditional flow like importing netlist and run place_opt just can't deliver acceptable QoR database. Getting a good correlative timing in synthesis is extremely important since it can reduce the iteration between front-end and back-end. And since we are moving forward to more advanced process and our floorplan get more complicated, physical-aware synthesis will be a grand trend.

4. References:

1) Design Compiler and IC Compiler Physical Guidance Technology Application Note.
2) Hierarchical Flow Support in Synopsys® Design Compiler Topographical Mode and IC Compiler Application Note.