OVERVIEW

Objective/Scope: For who in-charge CTS phase refer what to do (Project U2A6-chiptop flat/ U2B6/U2B10).

OVERVIEW

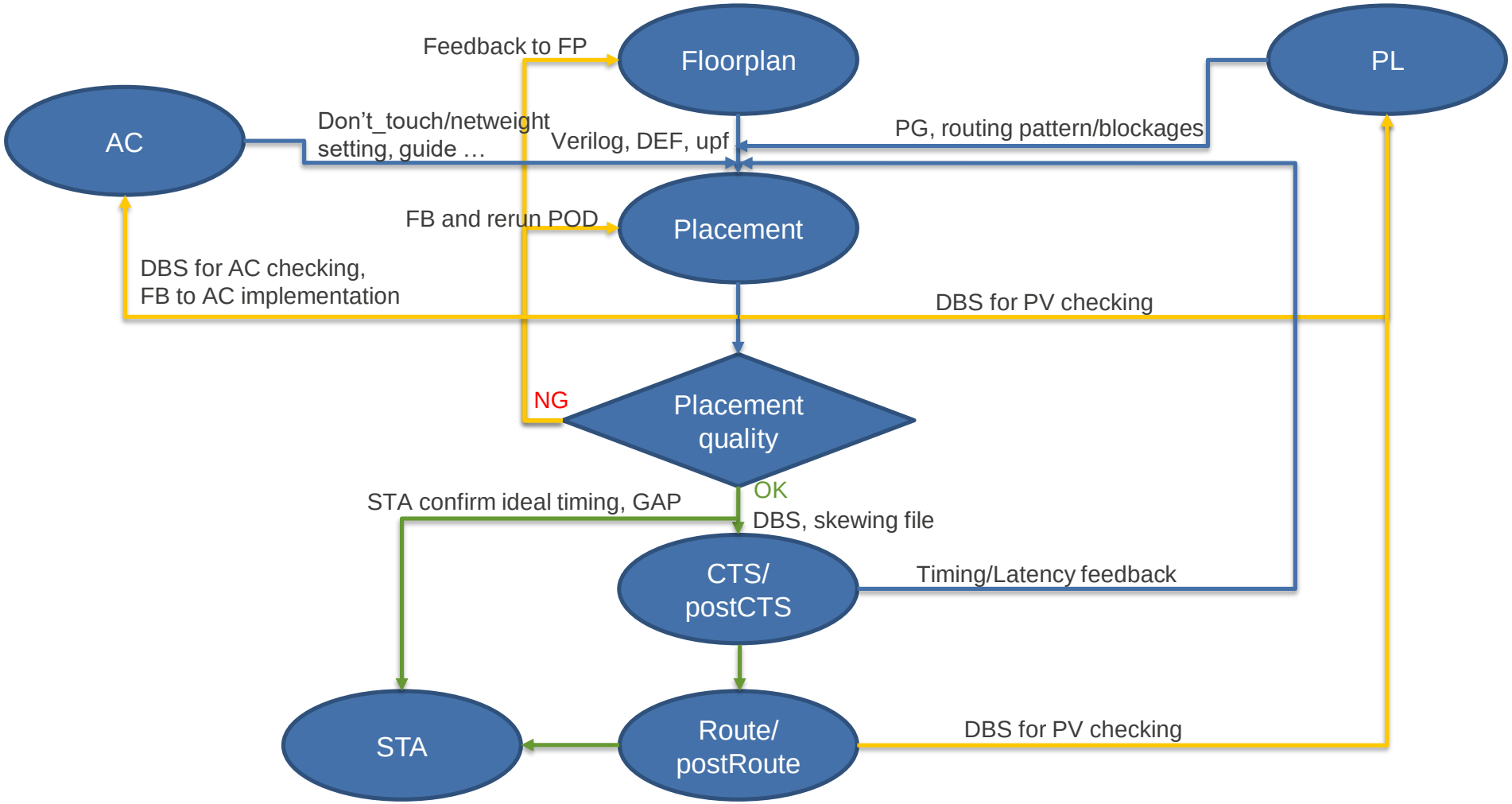
1. Physical Desgin Flow Overview
2. Clock Tree Synthesis Overview
3. Timing Path & Clock Structure
4. Clock Tree Synthesis Manual
5. Clock Tree Synthesis Flow

1. PHYSICAL DESGIN FLOW OVERVIEW

- PnR (Placement and Route) working flow
- Targets of PnR
- Chip PnR IN/OUTPUT

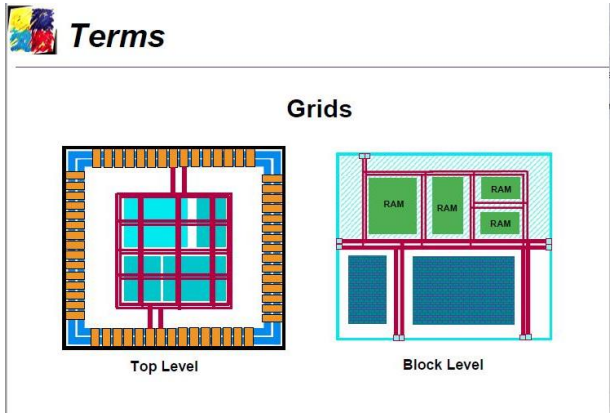
1. PHYSICAL DESGIN FLOW OVERVIEW

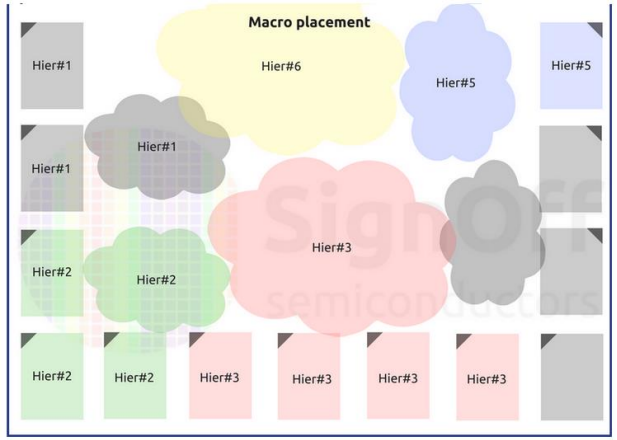
- **PnR (Placement and Route) working flow**

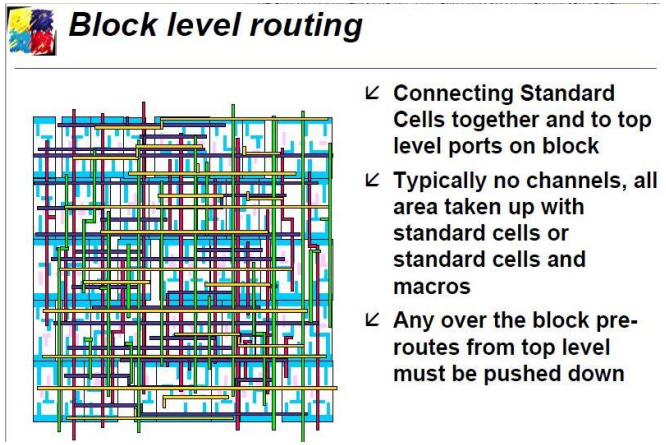


1. PHYSICAL DESGIN FLOW OVERVIEW

- **PnR (Placement and Route) working flow**



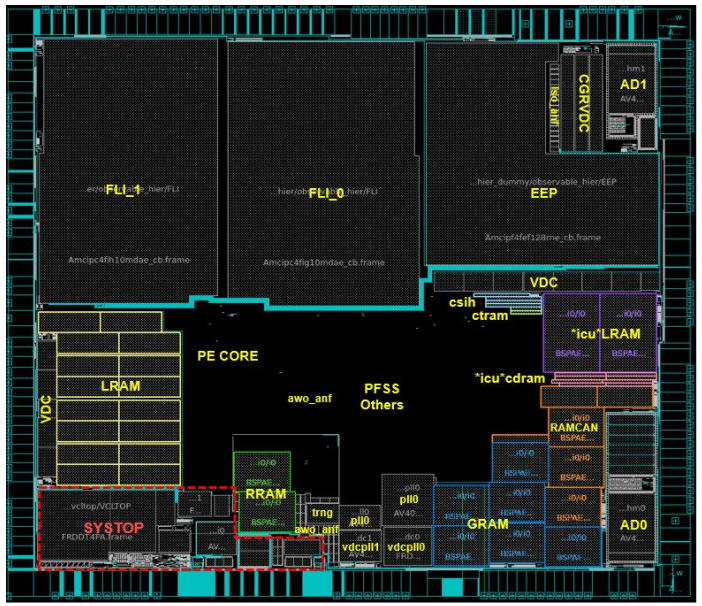


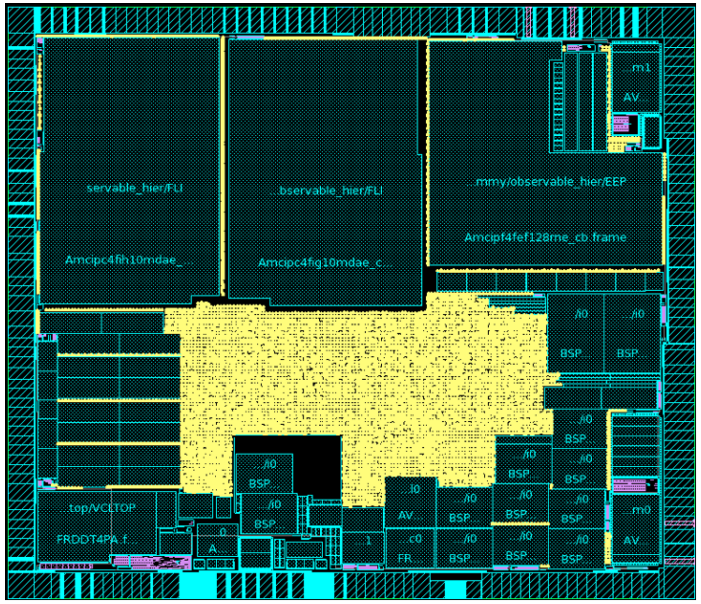


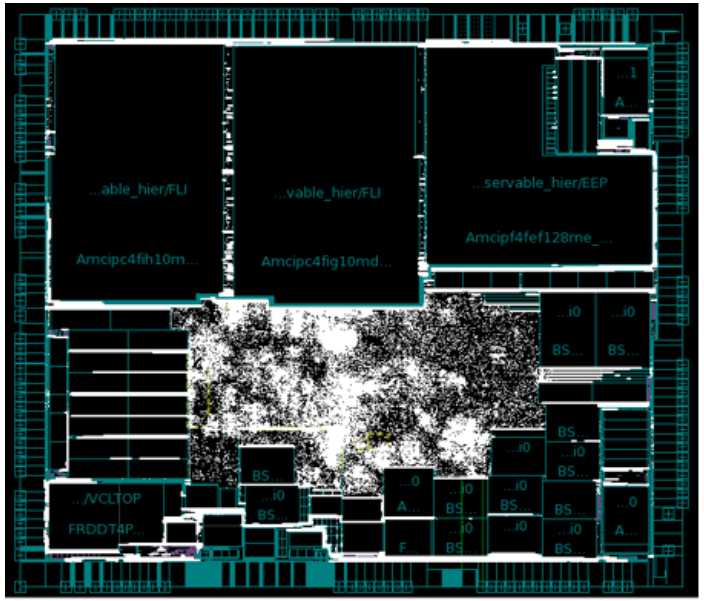
- **Floorplan:** performs placement of electronic components (excluding standard cells).
- **Placement:** performs placement of standard cells by using tools.
- **CTS/PostCTS:** performs clock tree synthesis and optimizations of clock tree.
- **Route:** performs metal wiring to connects signals between electronic components.
- **PostRoute Optimization:** performs final optimization of the designs (fix timing, drc, Xtalk, Noise …)

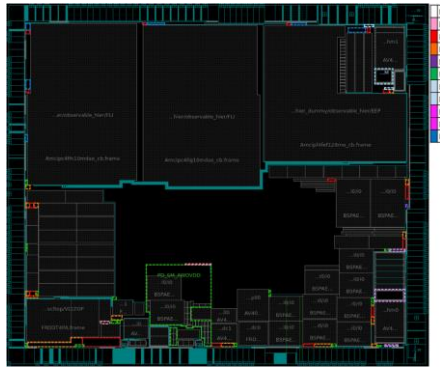
1. PHYSICAL DESGIN FLOW OVERVIEW

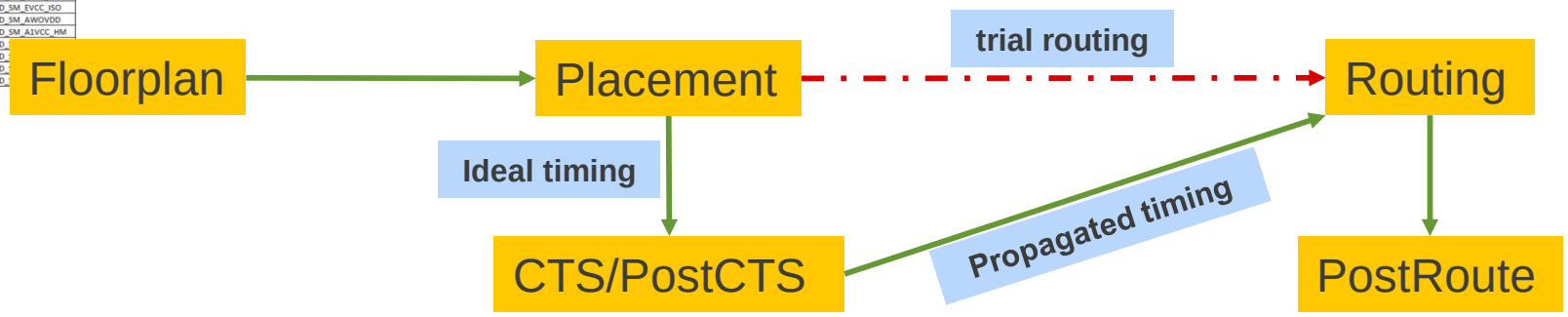
- PnR (Placement and Route) working flow





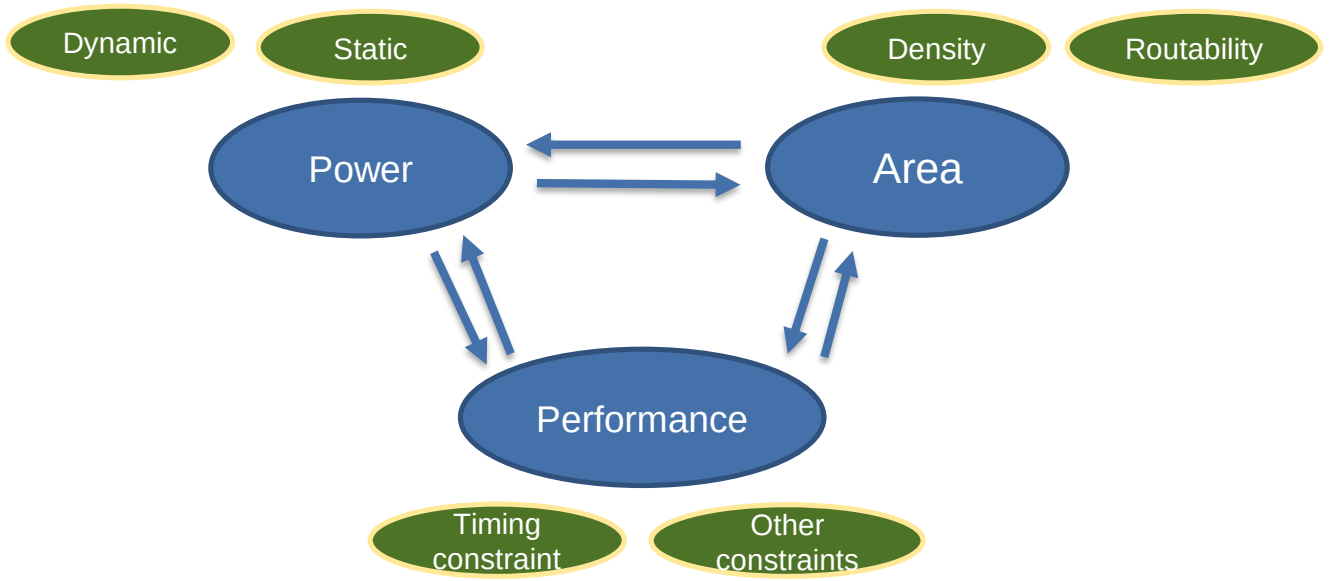






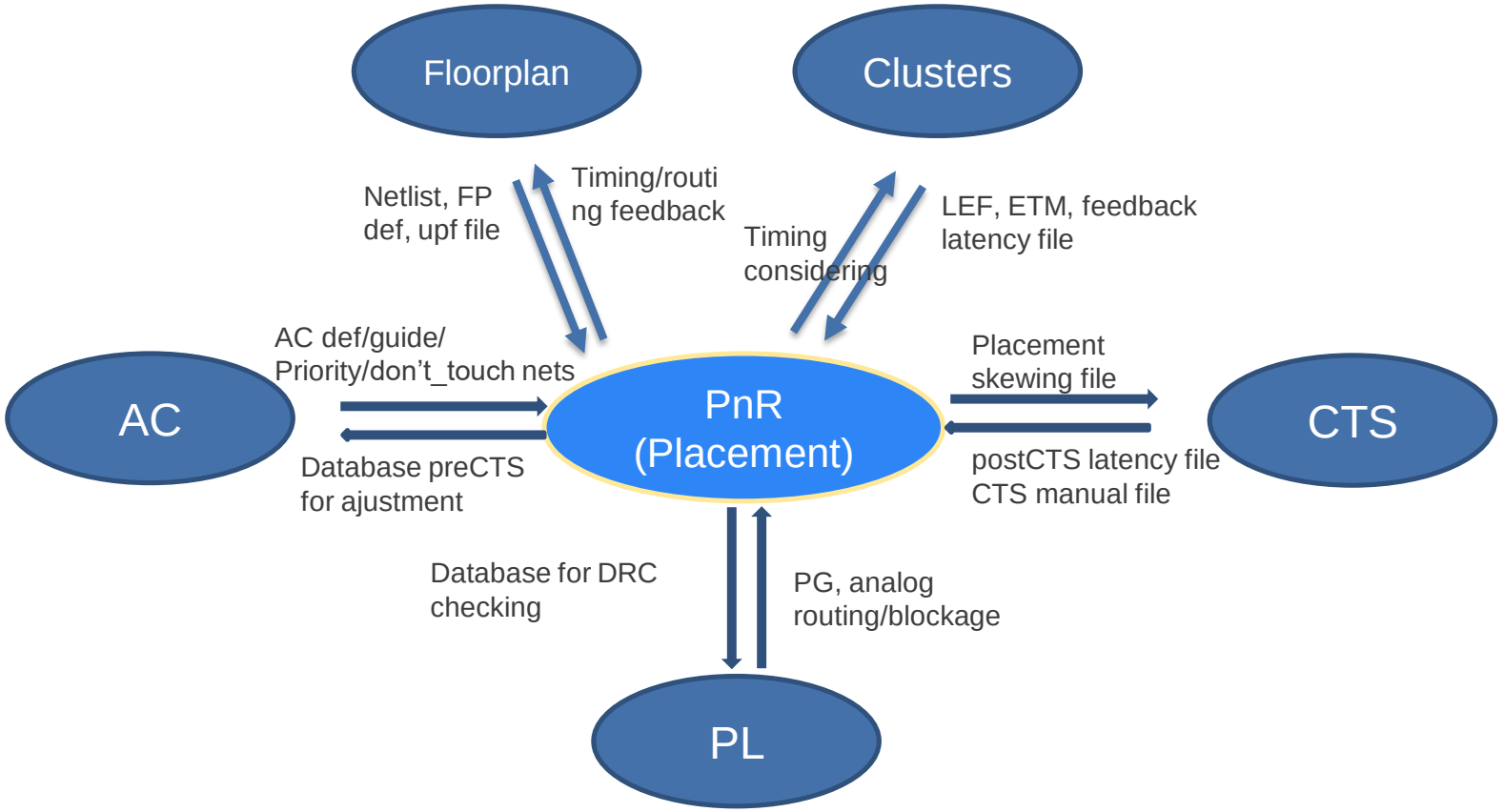
1. PHYSICAL DESGIN FLOW OVERVIEW

- **Targets of PnR**



1. PHYSICAL DESGIN FLOW OVERVIEW

- **Chip PnR IN/OUTPUT**



2. CLOCK TREE SYNTHESIS OVERVIEW

Thinking about Clock Tree Synthesis:

- What does CTS do?

After placement step which stardard cells are placed in design, we need to propagate clock to FFs. (at Placement step, the clock is treat as ideal => timing ideal). In CTS step, we will build a clock tree propagate from clock source to all sink/leaf (FFs).

Once we have created the clock, we need to propagate these clock in a way that all the clock elements present in the design need to switch at the same time. To achieve this, we need to balance these clocks and here comes the role of Clock Tree Synthesis (CTS) in physical design:

+ Clock propagate to all FFs.

+ Timing after propagate should be as same as possible at Placement phase (not jumpt to much)

+ All requirement adapt (HALO rules, NDR, tran/cap, noise/xtalk, Vth clock line)

To clearly about CTS and how to adapt these requirements, please refer next page.

2. CLOCK TREE SYNTHESIS OVERVIEW

The main task of CTS:

- Take care clock tree synthesize (care latency/skew which affect timing, care DRC, routing, tran/cap vio)
- Take care timing after CTS (timing met or not, check why: Do this violated timing come from CTS or others team? Do this violated be cover at postCTS?)
- Feedback ideal timing margin depend on CTS results to PnR

The sub task of CTS:

- Take care tran/cap clock line (which caused by long distance/big fanout)
- Take care Vth clock line
- Take care HALO rules
- Take care minpulse
- Take care NDR (Non Default Rule)
- Take care xtalk, noise.
- Take care 800Mhz clock lines

2. CLOCK TREE SYNTHESIS OVERVIEW

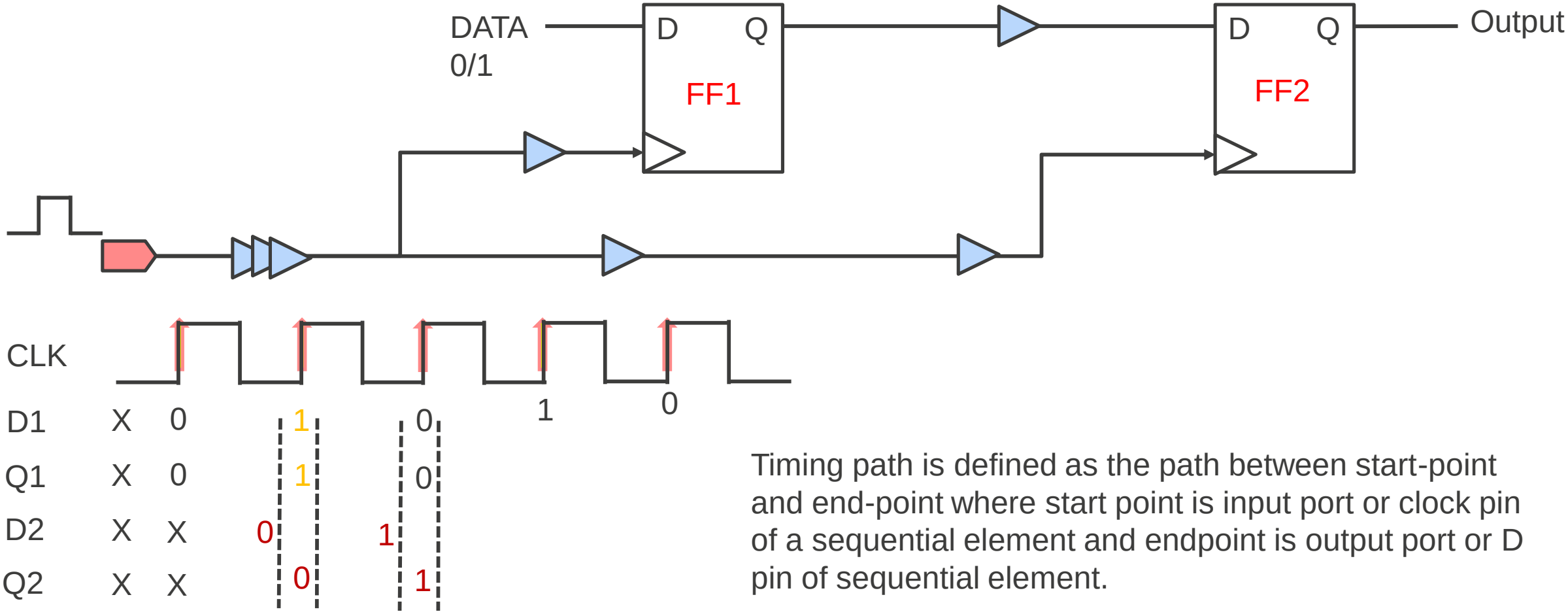
Before investigate what Clock Tree Synthesize, please understand what are **timing paths and clock structure** first. It's very important and basically to know what CTS do and quality of CTS.

-Timing paths: which is the most important quality in CTS step. At first stage of CTS, we will take care latency/skew whole design because it's directly affect timing. The most important target of CTS is timing (met or the number of violation is as small as possible), next page explain which is timing path, please refer it.

-Clock structure: CTS phase will propagate clock from source to all FFs, know clock structure to control the latency/skew group also debug easy if any.

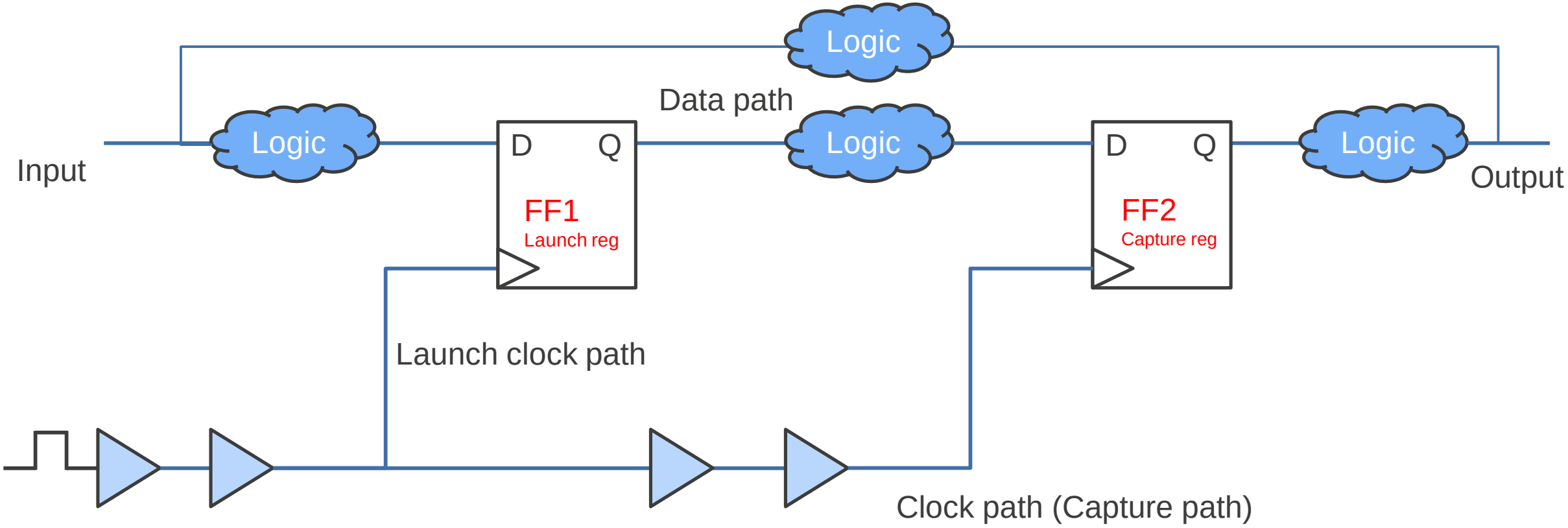
3. TIMING PATH & CLOCK STRUCTURE

- Flip-flops are used as data storage elements. A flip-flop is a device which stores a single *bit* (binary digit) of data.



Timing path is defined as the path between start-point and end-point where start point is input port or clock pin of a sequential element and endpoint is output port or D pin of sequential element.

3. TIMING PATH & CLOCK STRUCTURE



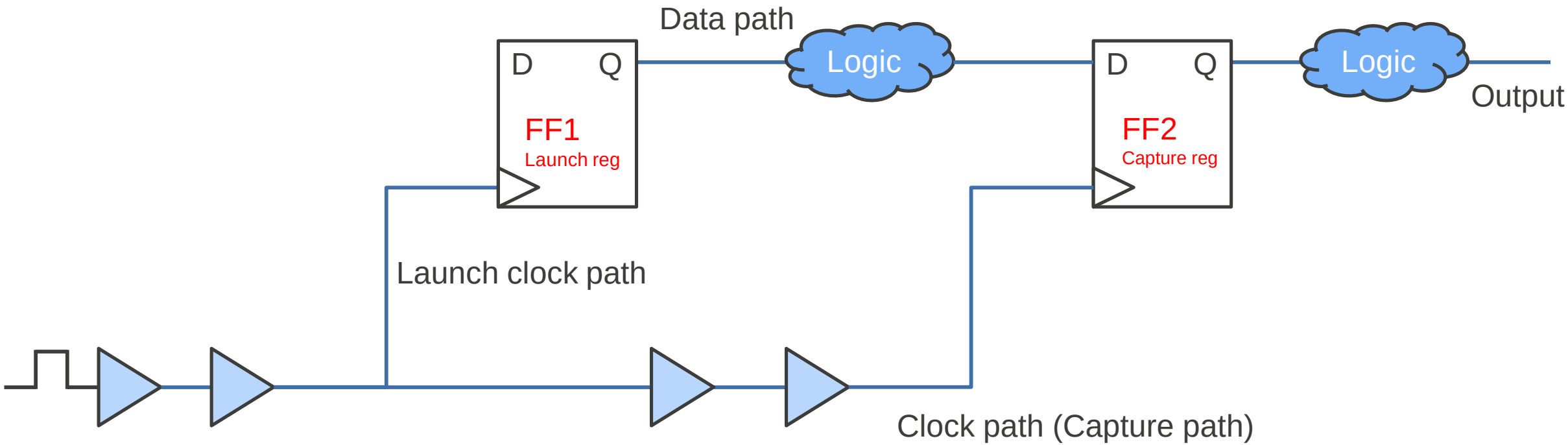
Input to register

Register to register

Register to output

Input to output

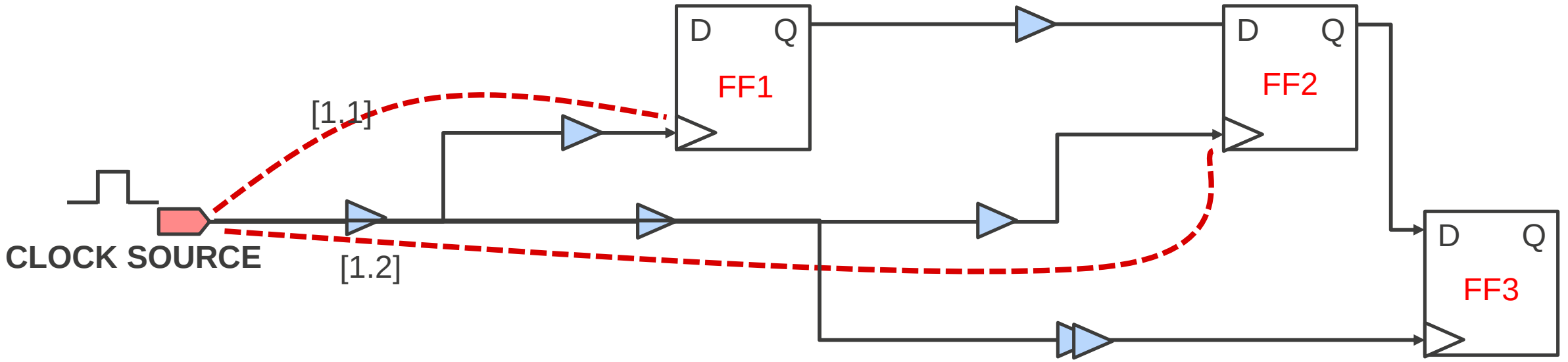
3. TIMING PATH & CLOCK STRUCTURE



Setup time: launch clock path + data path <= capture path – setup time + 1 cycle (Setup time is value define already by library)
Hold time: launch clock path + data path >= capture path + hold time (Hold time is value define already by library)

Violated Setup/Hold => data will transfer unstable/wrong => do not let any violated timing in design.

3. TIMING PATH & CLOCK STRUCTURE



[1.1/1.2] are latency. Latency is the amount of time a clock signal takes to propagate from the original clock source to the sequential elements in the design.
Local skew is the difference from clock timing path of only "related" FFs pairs. FFs are called "related" if one FF launches data when is captured by other (the difference between 1.1 and 1.2 called skew)
Global skew is the difference from clock timing path of all FFs in the design within the same clock source. And when global skew optimization is run, CTS tries to match the clock delays for all FFs' clock pins.

=> So lantency/skew affect directly timing.

3. TIMING PATH & CLOCK STRUCTURE

Please refer the detail timing path as link below:

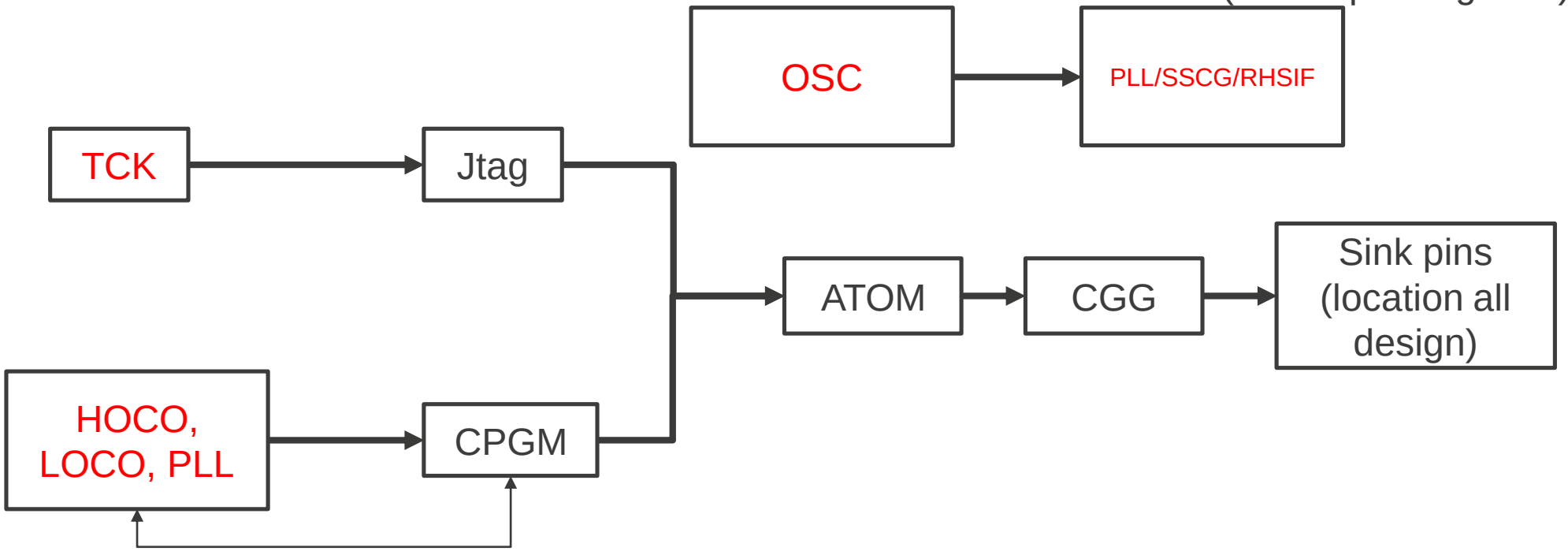
https://renesasgroup.sharepoint.com/:x:/r/sites/rvc-be2/_layouts/15/Doc.aspx?sourcedoc=%7BBD53AAE8-91CA-48D1-AB12-2770A46C396A%7D&file=Report_7_group_timing_path.xlsx&action=default&mobileredirect=true

https://renesasgroup.sharepoint.com/:x:/r/sites/rvc-be2/Shared%20Documents/General/Backend21/IMPROVEMENT_ACTIVITY/SKILL_IMPROVEMENT/DOCUMENT/MyNguyen/CTS%20U2A6/CTSinvTimingPath.xlsx?d=w1328edee6c9541d6a17c064681fe764e&csf=1&web=1&e=MGcmKU

3. TIMING PATH & CLOCK STRUCTURE

The overview structure of clock

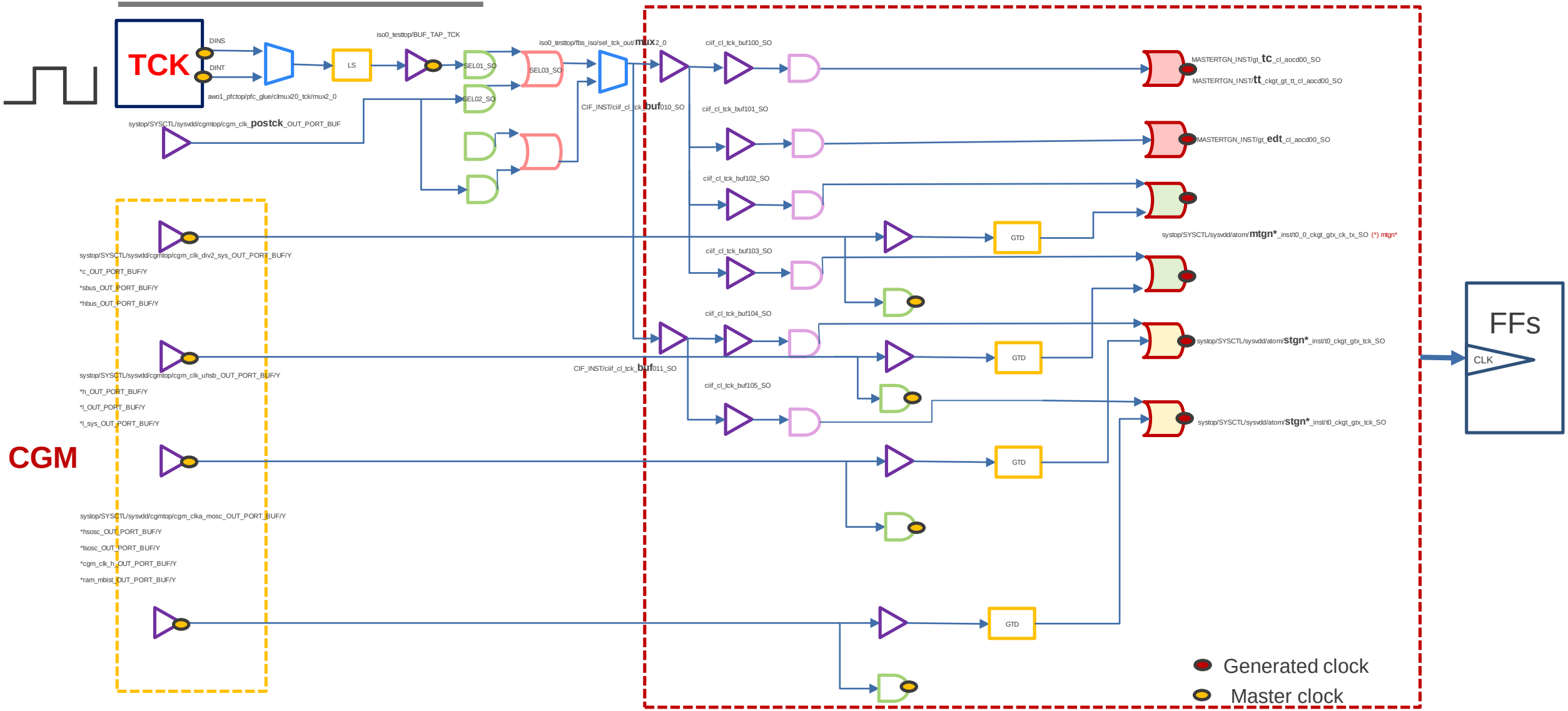
Please notice this structure (osc -> pll/sscg/rhsif)



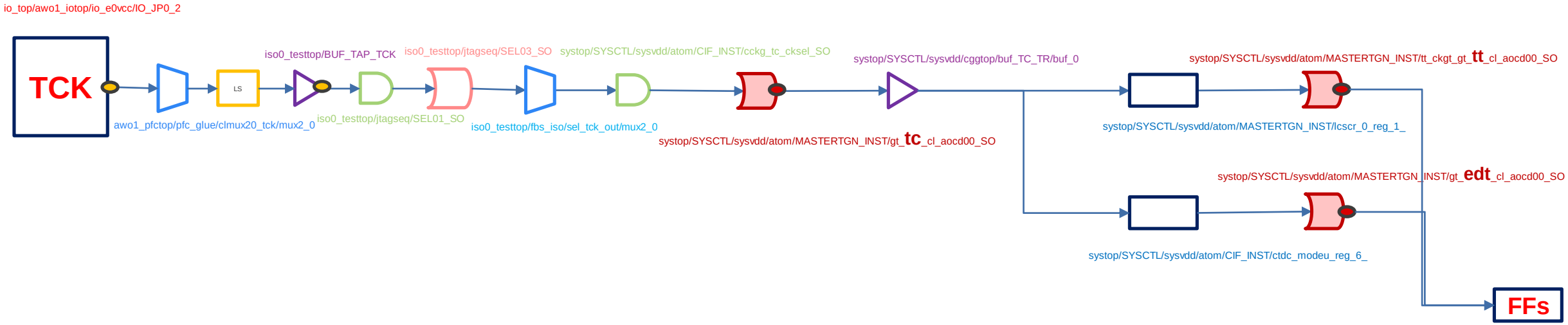
These source clock have frequency is 800Mhz, so we must take care initial (create module CPGM first of all, guide HOCO,LOCO, PLL manual with the line shortest from these hard macro to CPGM, OSC to PLL shortest). For more detail, please see in next page

3. TIMING PATH & CLOCK STRUCTURE

Exclusive buf/inver not balance
Trace to D : set false
No related timing not balance
Atom vs CPGM: have internal timing, not insert



3. TIMING PATH & CLOCK STRUCTURE



3. TIMING PATH & CLOCK STRUCTURE

More detail about clock structure, please refer link:

https://renesasgroup.sharepoint.com/:x:/r/sites/rvc-be2/Shared%20Documents/General/Backend21/IMPROVEMENT_ACTIVITY/SKILL_IMPROVEMENT/DOCUMENT/MyNguyen/CTS%20U2A6/CLOCK_STRUCTURE_DIVIDE%20.xlsx?d=w4174ed715ca546bab631b6bcbfa191da&csf=1&web=1&e=W8enRb

3. TIMING PATH & CLOCK STRUCTURE

Before you run clock tree synthesis, analyze each clock tree in the block and determine

•What the clock root is

•What the required clock sinks and clock tree exceptions are

•Whether the clock tree contains preexisting cells, such as clock-gating cells

•Whether the clock tree converges, either with itself (a convergent clock path) or with another clock tree (an overlapping clock path)

•Whether the clock tree has timing relationships with other clock trees in the block, such as interclock skew requirements

3. TIMING PATH & CLOCK STRUCTURE

What the clock root is

The ICC2 derives the clock trees by tracing through the transitive fanout from the clock roots to the clock endpoints. In general, the tracing terminates when it finds a clock pin of a sequential cell or macro; however, the tools traces through sequential cells if they are integrated clock-gating (ICG) cells or their fanout drives a generated clock.

+Clock roots: (create_clock) which can be either input ports or internal hier pins. For nested generated clocks (create_generated_clock), the tool consider the master-clock source to be the clock root, and the clock endpoints of the nested clock tree are considered endpoints of the master-clock source.

•If the tool cannot trace back to the master-clock source, it cannot balance the sink pins of the generated clock with the sink pins of its source.
•If the master-clock source is not a clock source defined by the create_clock or create_generated_clock command, the tool cannot synthesize a clock tree for the generated clock or its source.

3. TIMING PATH & CLOCK STRUCTURE

What the required clock sinks and clock tree exceptions are

When deriving the clock trees, the tool identifies two types of clock endpoints:

•Sink pins: are the clock endpoints that are used for delay balancing. The tool assigns an insertion delay of zero to all sink pins and uses this delay during delay balancing. During clock tree synthesis, the tool uses sink pins in calculations and optimizations for both design rule constraints and clock tree timing (skew and insertion delay). Sink pins are also referred to as balancing pins.

•Ignore pins: are clock endpoints that are **excluded** from clock tree timing calculations and optimizations. The tool uses ignore pins only in calculations and optimizations for design rule constraints. During clock tree synthesis, the tool isolates ignore pins from the clock tree by inserting a guide buffer before the pin. Beyond the ignore pin, the tool never performs skew or insertion delay optimization, but does perform design rule fixing.

=> There is another type that stop pins which are excluded from clock timing calculations and optimizations also perform design rule fixing => it maybe happened trans violated of these stop pins.

3. TIMING PATH & CLOCK STRUCTURE

What the required clock sinks and clock tree exceptions are

The tool identifies the following clock endpoints as sink pins:

•A clock pin on a sequential cell (a latch or flip-flip), unless that cell drives a generated clock

•A clock pin on a macro cell

The tool identifies the following clock endpoints as ignore pins:

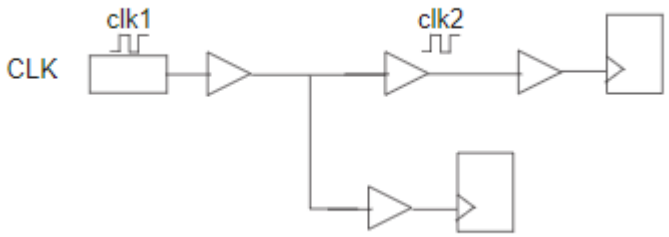
•Nonclock input pins of sequential cells

•Output ports

•Incorrectly defined clock pins (for example, the clock pin does not have trigger edge information or does not have a timing arc to the output pin)

•Input pins of combinational cells or integrated clock-gating cells that do not have any fanout or that do not have any enabled timing arcs

•Source pins of clock trees in the fanout of another clock



The source pin of the driven clock (clk2) is an ignore pin of the driving clock (clk1). Sinks of the driven clock are not considered sinks of the driving clock

5. CLOCK TREE SYNTHESIS FLOW

- Clock is a signal with constant rise and fall with ideally equal width (50% rise and 50% fall of the signal width) helps to control data propagation through the clock elements like FFs, Latches,… Once we have created the clock, we need to propagate these clock in a way that all the clock elements present in the design need to switch at the same time. To achieve this, we need to balance these clocks and here comes the role of Clock Tree Synthesis (CTS) in physical design.

- In CTS step, we will build a clock tree to supply clock signal for each flip-flop or hard/soft macro. The clock tree is very importance so that we need to build and route clock signal follow requirement to achieve the target that timing propagate is highest quality:

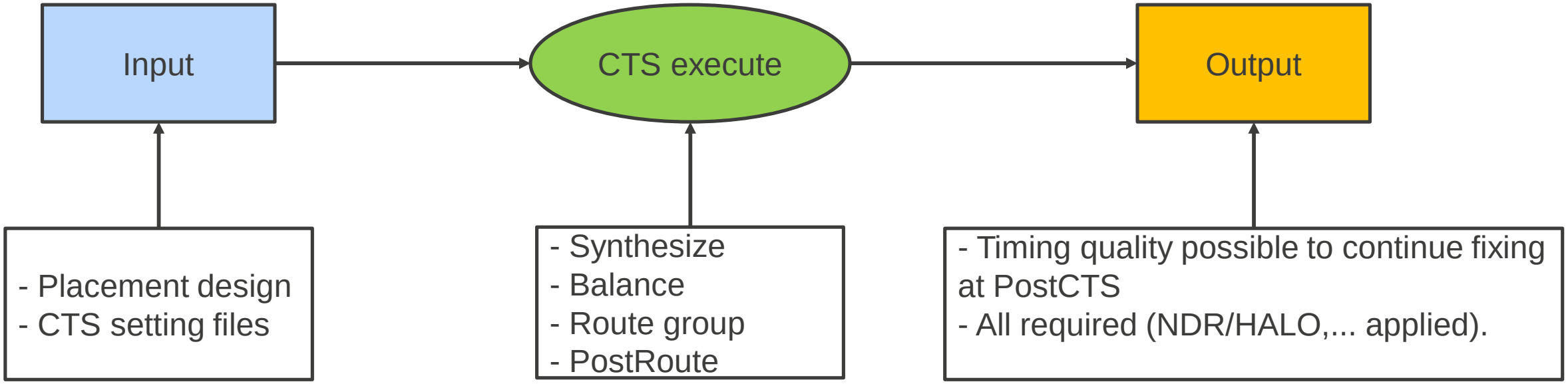
+ Trans/cap (DRC), Vth clock (Type cells), X-talk/Halo rules clean.

+ All constraint, NDR, Halo,... are applied.

+ Timing quality is highest (refer old design).

5. CLOCK TREE SYNTHESIS FLOW U2A6

CTS step:



5. CLOCK TREE SYNTHESIS FLOW

5.1 CTS step: INPUT

Before start CTS step, please ensuring design is ready for CTS:

- Placement database
- CTS setting files (spec files, library cells, constraint tran/cap/fanout, NDR rule/Halo rule definition, MCMM)

5. CLOCK TREE SYNTHESIS FLOW

5.1 CTS step: INPUT

- **Placement database:**

+ Which completed locating various circuit components (standard cell, macro, blockage, bound, tap cell, boundary cell,…) within the chip's core and optimizing a number of objectives to ensure that a circuit meets its performance demands.

+ Timing after placement (which will be compare with timing after CTS to know is it jump or not).

+ Congestion, density (which affect CTS synthesized if these are not good).

+ Useful skew if have

5. CLOCK TREE SYNTHESIS FLOW

5.1 CTS step: INPUT

- **CTS setting files**

+ CTS spec files

+ CTS library cells

+ Constraint tran/cap/fanout

+ NDR rule/Halo rule definition

5. CLOCK TREE SYNTHESIS FLOW

5.1 CTS step: INPUT

- **CTS Spec file:**

+ Clock definiton, CTS exceptions

+ Skew group, Target skew

+ MCMM setup file

Note: Chiptop need spec file, IP:SDC (which create/define clock, false path/timing)

MCMM: mode/corner(MAX) -> scenarios

* About mode/corner/scenarios, skew,... Please refer Synopsys book/website.

5. CLOCK TREE SYNTHESIS FLOW

5.1 CTS step: INPUT

MCMM run at CTS step: LT/HT: low temperature/high temperature

```
# MAX_HT
foreach c [list $corner1] {
    current_corner $c
    set_operating_conditions -analysis_type on_chip_variation
    set_voltage                0.945
    set_temperature        170
    set_process_number  1
}

##########################################################################
### derate or OCV
##########################################################################
###---
current_corner $corner1         ;# MAXHTSETUP
set_timing_derate  -corner  $corner1 -late  -net_delay  1.0
set_timing_derate  -corner  $corner1 -early -net_delay  1.0
set_timing_derate  -corner  $corner1 -late  -cell_delay 1.0
set_timing_derate  -corner  $corner1 -early -cell_delay 1.0

##########################################################################
### define voltage value
##########################################################################
foreach_in_col scenario [get_scenarios -corners [list $corner1]] {
  current_scenario $scenario
  source -c -v $UPF_SIDE_MAX_FILE
}
```

```
### Corner constraints; expand it as needed
#SETUP
set corner1                 "MAXHTSETUP" ;# name of corner1
set corner_constraints($corner1)        "" ;# for corner1 specific SDC constraints
```

```
##########################################################################
### define operation condition
##########################################################################

# MAX_LT
foreach c [list $corner1] {
    current_corner $c
    set_operating_conditions -analysis_type on_chip_variation
    set_voltage                                0.945
    set_temperature             -40
                set_process_number      1
}

#### MAX_HT
###foreach c [list $corner1] {
###    current_corner $c
###    set_operating_conditions -analysis_type on_chip_variation
###    set_voltage                                0.945
###    set_temperature          170
###    set_process_number       1
###}

##########################################################################
```

5. CLOCK TREE SYNTHESIS FLOW

5.1 CTS step: INPUT

```
************************************************************
* CTS_STEP: Design Initialization for Clock Synthesis
************************************************************
Information: CTS Exception cannot be applied with respect to generated clock: LB_DC_TCK_root2 and would be applied to its primary master clock. (UIC-084)
Information: CTS Exception cannot be applied with respect to generated clock: LB_DC_TT_TR and would be applied to its primary master clock. (UIC-084)
Information: All clock objects will be converted from ideal to propagated clock during CTS. (CTS-105)
Information: CTS will work on the following scenarios. (CTS-101)
    CTS_MAXLTSETUP       (Mode: CTS; Corner: MAXLTSETUP)
Information: CTS will work on the following clocks. (CTS-102)
  mosc40_oscout (Mode: CTS)
  mosc40_nfoosc (Mode: CTS)
  mosc40_oscout_div50   (Mode: CTS)
  mosc40_nfoosc_div50   (Mode: CTS)
  loco_xring    (Mode: CTS)
  ioscmax       (Mode: CTS)
  rngclk        (Mode: CTS)
  LB_DC_TT_TR   (Mode: CTS)
  LB_DC_TC_TR   (Mode: CTS)
  LB_DC_EDT_TR  (Mode: CTS)
  LB_DC_stgn0t0 (Mode: CTS)
  LB_DC_stgn0t1 (Mode: CTS)
  LB_DC_stgn1t0 (Mode: CTS)
  LB_DC_stgn2t0 (Mode: CTS)
  LB_DC_stgn3t0 (Mode: CTS)
  LB_DC_stgn4t0 (Mode: CTS)
  LB_DC_stgn4t1 (Mode: CTS)
  LB_DC_stgn5t0 (Mode: CTS)
  LB_DC_mtgn0t00        (Mode: CTS)
  LB_DC_mtgn0t10        (Mode: CTS)
  LB_DC_mtgn0t11        (Mode: CTS)
  LB_DC_mtgn0t12        (Mode: CTS)
  LB_DC_mtgn0t20        (Mode: CTS)
  LB_DC_mtgn0t30        (Mode: CTS)
  LB_DC_mtgn0t40        (Mode: CTS)
  LB_DC_mtgn0t50        (Mode: CTS)
  LB_DC_mtgn0t51        (Mode: CTS)
  LB_DC_mtgn0t52        (Mode: CTS)
  LB_DC_mtgn0t60        (Mode: CTS)
  LB_DC_mtgn0t61        (Mode: CTS)
  LB_DC_mtgn0t62        (Mode: CTS)
  LB_DC_mtgn0t70        (Mode: CTS)
  JTAG_NEXUS    (Mode: CTS)
  pllclk800     (Mode: CTS)
  nclkout_pll800        (Mode: CTS)
  pllclk800_tbo0        (Mode: CTS)
  pllclk800_tbo0_div2   (Mode: CTS)
  pllclk800_tbo0_div4   (Mode: CTS)
```

To make sure the scenario, clock defition CTS working on, in log file at CTS STEP have information

5. CLOCK TREE SYNTHESIS FLOW

5.1 CTS step: INPUT

- **CTS Spec file**:

Clock definition, CTS exceptions

- Clock definition: (clock structure: TCK, CPGM)

- CTS exceptions:

+ set_clock_balance_points (false: ignore, true: stop (delay 0/ delay $)).

+ set_sense (restricts unateness propagating forward for pins with respect to the clock source = ignore).

5. CLOCK TREE SYNTHESIS FLOW

5.1 CTS step: INPUT

- CTS Spec file:

Skew group, Target skew

```
create_clock_balance_group -name CPGM_all -objects { \
cgm_clk_c \
cgm_clk_sbus \
cgm_clk_hbus \
cgm_clk_l_sys \
cgm_clk_h \
cgm_clk_l \
cgm_clk_uhsb \
cgm_clka_mosc \
cgm_clka_hsosc \
cgm_clka_lsosc \
cgm_clka_ram_mbist \
cgm_clks_mosc \
cgm_clks_lsosc \
cgm_clks_hsosc \
cgm_clk_l_rhsif \
}

##cgm_clk_div2_sys \


report_clock_balance_groups
~
```

```
#################################################################################
### CTS target skew and latency
#################################################################################
#set_clock_tree_options -target_skew 0.00 -target_latency 0.00
set_clock_tree_options -target_skew 0.00
#################################################################################
```

5. CLOCK TREE SYNTHESIS FLOW

5.1 CTS step: INPUT

CTS library cells (buffer/inverter) to balance/ fix Trans-Cap/ minimize skew.

```
set cts_buf_cells [get_lib_cells {
*/TSHBUFCLX20
}]
##*/TSHBUFCLX120 */TSHBUFCLX100 */TSHBUFCLX80 */TSHBUFCLX60 */TSHBUFCLX40 */TSHBUFCLX30 */TSHBUFCLX20

set cts_inv_cells [get_lib_cells {
*/TSHINVCLX160 */TSHINVCLX120 */TSHINVCLX100 */TSHINVCLX80 */TSHINVCLX60 */TSHINVCLX40 */TSHINVCLX30 */TSHINVCLX20
}]

set cts_clock_gating [get_lib_cells {
*/TSHGTDPX20 */TSHGTDPX40 */TSHGTDPX80 */TSHGTDSBX20 */TSHGTDSBX40 */TSHGTDSBX80 */TSHGTDX20 */TSHGTDX40 */TSHGTDX80 */TWHGTDZHSX20 */TWHGTDZHSX40 */TWHGTDZHSX80
}]

set cts_clock_logic [get_lib_cells {
*/TSHXOR2CLX* */TSHMUX2CLX* */TSHAND2CLX* */TSHOR2CLX* */TSHXNOR2CLX*
}]

set_lib_cell_purpose -exclude cts [get_lib_cells */*]
set_dont_touch $cts_buf_cells false
set_dont_touch $cts_inv_cells false
set_dont_touch $cts_clock_gating false
set_dont_touch $cts_clock_logic false


set_lib_cell_purpose -include cts $cts_buf_cells
set_lib_cell_purpose -include cts $cts_inv_cells
set_lib_cell_purpose -include cts $cts_clock_gating
set_lib_cell_purpose -include cts $cts_clock_logic
```

5. CLOCK TREE SYNTHESIS FLOW

5.1 CTS step: INPUT

- Useful skew:

+ Most timing violations are fixed by data path optimization. With useful skew, you fix timing violations by adjusting clock arrival times at registers or latches.

```
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/cgc_cmem/il/q_reg/CLK
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/cgc_cmem/il/q_reg/CLK
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/i0/i0/CLK
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/i0/i0/CLK
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/i0/i0/CLK
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/i0/i0/CLK
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/i0/obs_nma_reg/CLK
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/i0/obs_nma_reg/CLK
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/i0/obs_nma_reg/CLK
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/i0/obs_nma_reg/CLK
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/i0/obs_bwn_reg_8_/CLK
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/i0/obs_bwn_reg_8_/CLK
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/i0/obs_bwn_reg_8_/CLK
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/i0/obs_bwn_reg_8_/CLK
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/i0/obs_bwn_reg_7_/CLK
set_clock_balance_points -delay -0.060 -consider_for_balancing true -balance_points dbg0_dbgtop/C0ERAM_wp/M00/ia_0_0/i0/obs_bwn_reg_7_/CLK
```

5. CLOCK TREE SYNTHESIS FLOW

5.1 CTS step: INPUT

Setting clock tran/cap/fanout: Adjust by sign-off. Can change but limit is depending on sign-off.

```
foreach_in_collection t [all_clocks] {
    set p [get_attr $t period]
    set name [get_attr $t full_name]

    set max_tran 0.2
    set max_cap 0.1
    puts "*** Setting tran/cap for clock: $name\($p\) with tran = $max_tran, cap = $max_cap"
    set_max_transition  -clock_path $max_tran $t
    set_max_capacitance -clock_path $max_cap $t
}

~

#control netlength
set_app_option -name cts.common.max_net_length -value 400
#fanout
set_app_options -list {cts.common.max_fanout 32}
```

5. CLOCK TREE SYNTHESIS FLOW

5.1 CTS step: INPUT

Halo rules (cell spacing):

/design04/u2a8_bed1_vf/u2a8/data/u2a8/5_layout/v100/Top/ICC2_MASTER/300_CMD_icc2/RV28F_clock_spacing_400MHz_0.40ns_0.18pF_v7r0s0_Pff_add_1129.tcl

```
set_clock_cell_spacing -x_spacing 1.325 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHAND2CLX10
set_clock_cell_spacing -x_spacing 2.810 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHAND2CLX20
set_clock_cell_spacing -x_spacing 4.261 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHAND2CLX30
set_clock_cell_spacing -x_spacing 5.713 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHAND2CLX40
set_clock_cell_spacing -x_spacing 6.205 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHAND2CLX60
set_clock_cell_spacing -x_spacing 6.152 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHAND2CLX80
set_clock_cell_spacing -x_spacing 1.397 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX10
set_clock_cell_spacing -x_spacing 2.887 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX20
set_clock_cell_spacing -x_spacing 4.339 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX30
set_clock_cell_spacing -x_spacing 5.809 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX40
set_clock_cell_spacing -x_spacing 5.971 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX50
set_clock_cell_spacing -x_spacing 6.335 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX60
set_clock_cell_spacing -x_spacing 6.286 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX70
set_clock_cell_spacing -x_spacing 6.256 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX80
set_clock_cell_spacing -x_spacing 6.163 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX100
set_clock_cell_spacing -x_spacing 6.076 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX120
set_clock_cell_spacing -x_spacing 5.968 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX140
set_clock_cell_spacing -x_spacing 5.891 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX160
set_clock_cell_spacing -x_spacing 5.885 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX180
set_clock_cell_spacing -x_spacing 5.830 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX200
set_clock_cell_spacing -x_spacing 5.666 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX240
set_clock_cell_spacing -x_spacing 5.349 -y_spacing 0.400 -lib_cell CLN28EFATHH_*/THHBUFCLX320
```

HALO is applied to prevent the EM (Electro-Migration) which affect the working time of chip => must have no violation HALO rules.
There’re 2 rules for clock (400Mhz & 800Mhz (a private rules for CPGM)).

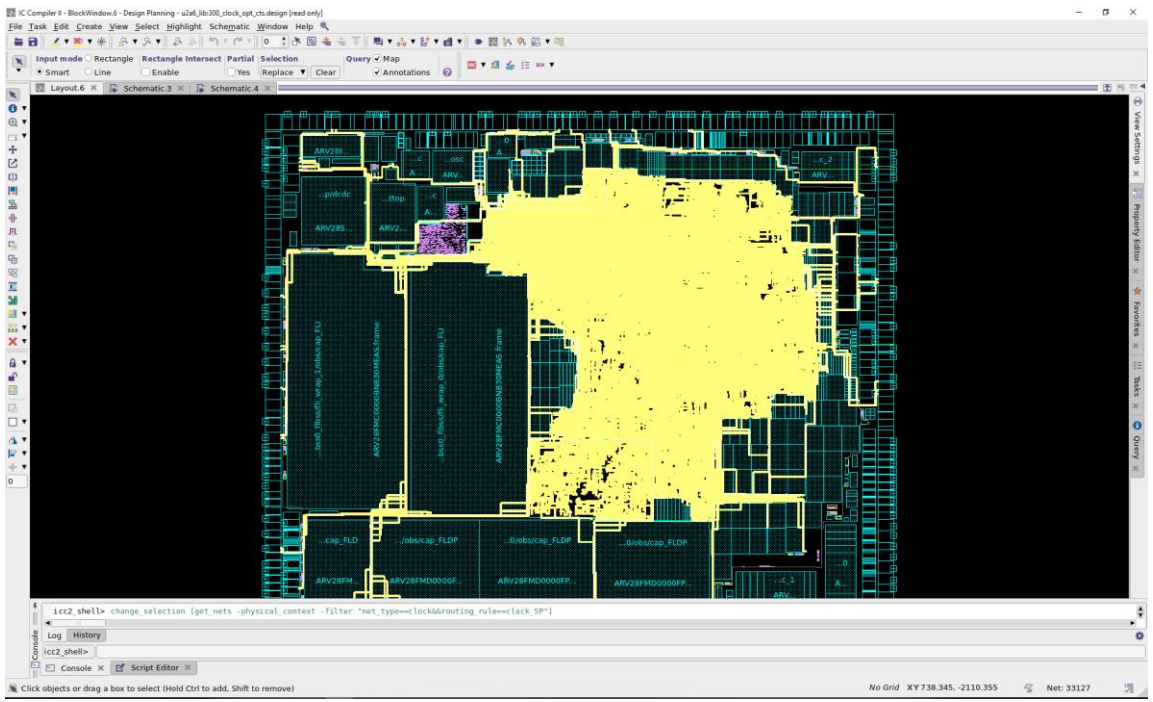
5. CLOCK TREE SYNTHESIS FLOW

5.1 CTS step: INPUT

NDR

```
if {[sizeof_collection [get_routing_rules -q clock_SP]] == 0} {
create_routing_rule clock_SP -default_reference_rule \
    -widths   { M1 0.05 M2 0.05 M3 0.05 M4 0.05 M5 0.05 M6 0.10 M7 0.10 } \
    -spacings { M1 0.05 M2 0.05 M3 0.15 M4 0.15 M5 0.15 M6 0.30 M7 0.30 } \
}

set_clock_routing_rules -rule clock_SP -net_type root     -min_routing_layer M6 -max_routing_layer M7
set_clock_routing_rules -rule clock_SP -net_type internal -min_routing_layer M6 -max_routing_layer M7
set_clock_routing_rules -default_rule  -net_type sink     -min_routing_layer M2 -max_routing_layer M5
```



5. CLOCK TREE SYNTHESIS FLOW

5.1 CTS step: INPUT

set_clock_routing_rules: Specifies routing rules used in clock tree synthesis.

net_type:

+sink: the nets that connect to one or more clock leaf pins

+root: the nets that are driven by the clock source. When synthesis inserts buffers, this type applies to the buffer chain before branching out.

+internal: the rest of the nets in the clock tree.

5. CLOCK TREE SYNTHESIS FLOW

5.1 CTS step: INPUT

To check NDR/HALO rules are applied yet, please read log file at CTS STEP:

```
INFO: auto-mv setup started.
CTS NDR rule list:
   Design Base; Net Type: sink;     Rule: default rule; Min Layer: M2; Max Layer: M5
   Design Base; Net Type: internal; Rule: clock_SP; Min Layer: M6; Max Layer: M7
   Design Base; Net Type: root;     Rule: clock_SP; Min Layer: M6; Max Layer: M7

Clock cell spacing rule list:
   Libcell: CLN28EFATWH_9lm4X2Y2RRDL/TWHGTDZHSX20; Spacing X 2.207000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATWH_9lm4X2Y2RRDL/TWHGTDZHSX40; Spacing X 5.115000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATWH_9lm4X2Y2RRDL/TWHGTDZHSX80; Spacing X 5.587000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAN21ZHX10; Spacing X 1.669000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAN21ZHX20; Spacing X 3.405000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAN21ZRHX20; Spacing X 3.405000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAN22ZHX20; Spacing X 3.405000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAN31X20; Spacing X 3.405000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAN31ZHX20; Spacing X 3.405000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAND2CLX10; Spacing X 1.592000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAND2CLX20; Spacing X 3.321000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAND2CLX30; Spacing X 5.019000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAND2CLX40; Spacing X 6.393000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAND2CLX60; Spacing X 6.278000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAND2CLX80; Spacing X 6.222000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAND2FX20; Spacing X 3.405000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAND2X20; Spacing X 3.405000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAO21ZFX80; Spacing X 6.341000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHAO31X20; Spacing X 3.405000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX10; Spacing X 1.669000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX100; Spacing X 6.275000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX120; Spacing X 6.186000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX140; Spacing X 6.104000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX160; Spacing X 6.029000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX180; Spacing X 6.060000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX20; Spacing X 3.405000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX200; Spacing X 6.001000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX240; Spacing X 5.862000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX30; Spacing X 5.108000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX320; Spacing X 5.615000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX40; Spacing X 6.466000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX50; Spacing X 6.441000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX60; Spacing X 6.417000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX70; Spacing X 6.371000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFCLX80; Spacing X 6.341000 ; Spacing Y 0.400000 ;
   Libcell: CLN28EFATSH_9lm4X2Y2RRDL/TSHBUFFX20; Spacing X 3.405000 ; Spacing Y 0.400000 ;
   ... (more spacing rules)
```

5. CLOCK TREE SYNTHESIS FLOW

5.2 CTS step: CTS execute

Clock tree synthesize: Builds and optimizes clock trees based on the clock definition.

Balance clock groups: Execute clock balance action among the clock balance groups which are defined by create_clock_balance_groups.

Route group: Route a specified group of nets.

Postroute: Runs post-route clock tree optimization. This option should be used after clock tree is synthesized and routed.

5. CLOCK TREE SYNTHESIS FLOW

5.2 CTS step: CTS execute

Clock tree synthesize: Builds and optimizes clock trees based on the clock definition.

-clocks: Specifies the clocks to compile. Without this option the command synthesizes all currently defined clocks in all active modes.

-propagate_only: Propagates all clocks in all active scenarios. This option should be used when more modes/scenarios are activated after CTS is done. It does not synthesize the clocks but only propagates the clocks.

-postroute: Runs post-route clock tree optimization. This option should be used after clock tree is synthesized and routed.

5. CLOCK TREE SYNTHESIS FLOW

5.2 CTS step: CTS execute

Clock tree synthesize: This command synthesizes clock trees and updates the design database with the compiled clock trees. The command will work on clocks in active scenarios. It will have clocks balanced if either setup or hold analysis is active. It will fix the transition violations on clock pins if max_transition analysis is active, and fix the capacitance violation on clock nets if max_capacitance analysis is active. At the end of this command it will invoke mark_clock_trees to mark clock synthesized attributes on clock objects.

The log file performs those CTS STEP which are derived from cmd: synthesizes_clock_tree:

Command: synthesize_clock_trees -postroute
- Design Initialization for Post-Route
- Clock Tree Initialization
- Post-Route Clock Tree Optimization

**For more detail, please investigate yourself*

Command: synthesizes_clock_trees
- Design Initialization for Clock Synthesis
- Existing Clock Tree Removal
- Clock Tree Initialization
- Gate-By-Gate Clock Tree Synthesis
- Latency Bottleneck Analysis
- DRC Fixing Beyond Exceptions
- Clock Net Global Routing
- Pre-Optimization DRC Fixing
- Skew Latency Optimization and Area Recovery
- Post-Optimization DRC Fixing
- Postlude
- Summary report

5. CLOCK TREE SYNTHESIS FLOW

5.2 CTS step: CTS execute

mark_clock_trees: This command marks clock attributes on clock objects to prevent the subsequent optimization programs to modify the built clock trees.

-clocks: Marks only the clock trees specified in clock_list

-synthesized: Marks clock synthesized attributes on clock objects, including cells and nets in the clock trees. On clock nets the net_type attribute updated as clock. The dont_touch attribute for CTS reason is also marked on clock nets. On clock network cells physical_status attribute set as application fixed. On sinks cts_size_in_place attribute set to true. This option cannot be used together with -dont_touch and -routing_rules.

-dont_touch: Sets dont_touch attribute on clock objects, including cells and nets in the clock trees. This option cannot be used together with -synthesized and -routing_rules.

-fix_sinks: Marks clock sink as fixed placement in order to avoid subsequent commands sizing or moving them.

5. CLOCK TREE SYNTHESIS FLOW

5.2 CTS step: CTS execute

Balance clock groups: This command will balance clocks insertion delay for all clock balance groups created by create_clock_balance_group.

The log file performs those CTS STEP which are derived from cmd: balance_clock_groups:

Command: balance_clock_groups
- Design Initialization for Clock Balance
- Inter-Clock Balance
- Postlude

create_clock_balance_group: create a clock balance group in which all clocks have related timing together.

5. CLOCK TREE SYNTHESIS FLOW

5.2 CTS step: CTS execute

Route group: Route a specified group of nets.

-all_clock_nets: routes all clock nets.

-reuse_existing_global_route: Controls whether the tool performs incremental global routing. If you *specify -reuse_existing_global_route true*, the global router considers and reuses the existing global routes. By the default, it's false.

-route_nondefault_nets_first: Specifies whether to route the nets with nondefault routing rules before the nets that use the default routing rule. When true, the tool performs global routing and track assignment first on nets with nondefault routing rules, and then on nets that use the default routing rule. By the default, it's false.

5. CLOCK TREE SYNTHESIS FLOW

5.2 CTS step: CTS execute

You can read log file to understand clearly what CTS does by search keyword: **CTS STEP**

This is an example about CTS STEP which show the delay jump can help to debug if have any latency strange.

```
Information: The run time for gate-by-gate clock tree synthesis is 10 hr : 44 min : 58.90 sec, cpu time is 21 hr : 28 min : 55.90 sec. (CTS-104)
************************************************************
* CTS STEP: Latency Bottleneck Analysis
************************************************************
* PD jump Gates (sorted by PD jump): Largest latency jumps (> 0.500000)

Clock         Phase     PD Jump   Buf/inv   Sinks     Gates     From Pin  To Pin
Root          Delay
-------------------------------------------------------------------------------------------------
AC_ETNB0_AVB_RX_CLK_T 1.0782 0.7411 9       4         1         iso0_eth/eth0/pif/clkrst/clock_mux_3/U1/mux2_0/Y iso0_eth/eth0/pif/clkrst/micg_gate_
AC_ETNB0_AVB_TX_CLK_T 2.7941 0.7396 12      0         2         iso0_eth/rmiic0/clksel/mmux_gate_1/mux2_0/Y iso0_eth/eth0/pif/clkrst/clock_mux_5/U1/
AC_ETNB0_AVB_TX_CLK_T 1.8858 0.5798 10      0         3         iso0_eth/eth0/pif/clkrst/clock_mux_5/U1/mux2_0/Y iso0_eth/eth0/pif/clkrst/clock_mux_
AC_SFMA0_PAD_0 1.0166   1.0166    14        2         9         iso0_sfma/spi_core/spibc_spictl/ECO_sfma_branch_point_AC_inst/Y iso0_sfma/spi_core/s
AC_clk_extfxr_S 4.1286  1.1311    14        0         1         io_top/iso0_iotop/iso_scaap_e1/uECOB_AC_iso_Buffer_P10_6_DINS_/Y io_top/iso0_iotop/i
AC_clk_extfxr_S 1.8978  0.9391    8         0         1         iso0_pfctop/porttop/pcm10/MODAMG_P10_6_CTCLKIN_BUFI/U4/YB iso0_flxtop/flx0/ip/produ
AC_mspi_10_sck_gen_1_0 3.0937 2.8839 40     0         1         io_top/iso0_iotop/iso_scaap_e0/scaap_bs_BSR0025_G/G4/Y io_top/iso0_iotop/iso_scaap_e
AC_mspi_10_sck_gen_3_1 0.8209 0.7260 10     0         1         io_top/iso0_iotop/iso_scaap_e1/scaap_bs_BSR0067_G/G4/Y io_top/iso0_iotop/iso_scaap_e
AC_mspi_10_sck_in_3_1_S 1.5839 0.6523 8     0         1         io_top/iso0_iotop/iso_scaap_e1/uECOB_AC_iso_Buffer_P22_4_DINS_/Y io_top/iso0_iotop/i
DCDCCKDIV8_AWO 3.3209   0.6494    2         0         1         systop/dcdc_top/Diode_buf_cell_156/Y systop/dcdc_top/dcdc_awo/CKDIV8_V11_scanmsk_MUX
GTMECLK0      4.5204    3.7538    50        0         1         iso0_pfctop/porttop/pcm17/PINMLT_P17_4_NML/U4/YB iso0_pfctop/porttop/pcm17/PINMLT_P1
GTMECLK1      2.0869    1.2806    14        0         1         iso0_pfctop/porttop/pcm17/PINMLT_P17_5_NML/U3/YB iso0_pfctop/porttop/pcm17/PINMLT_P1
GTMECLK2      1.5623    1.4854    18        0         1         io_top/iso0_iotop/iso_scaap_e1_P17_6/scaap_bs_BSR0175_G/G4/Y io_top/iso0_iotop/iso_s
LB_AC_mtgn0t11 1.8300   1.2954    297       5078      1530      systop/SYSCTL/sysvdd/cggtop/gck0a_3/gck/GCLK pfsstop/PFSS/CLWrapCL0/G4MHPE_PE0/L1RAM
LB_AC_mtgn0t11 1.2791   0.8132    235       897       3773      pfsstop/PFSS/PFSS_SYSTEM/PFSS_SYSTEM_CTRL/DBGSS_CLK_CTRL/micg_gate_0/gck/GCLK pfsstc
LB_AC_mtgn0t11 1.2675   0.6011    169       68        3085      pfsstop/PFSS/CLWrapCL0/G4MHPE_PE1/PECORE/sys_clk_FUSA_IN_PORT_BUF/Y pfsstop/PFSS/CLW
LB_AC_mtgn0t11 0.8053   0.5713    64        653       170       iso0_memtop/fsy0/flbss0_flbss/flbss_cgc/cgc1/gck/GCLK iso0_memtop/fsy0/flbss0_flbss/
LB_AC_mtgn0t11 0.6163   0.5563    6         0         1         systop/sysglue_isovdd/isengen_FBIST_ISO_1_616/sengen_and_clk_scan_cgc/Y systop/sysg
LB_AC_mtgn0t11 1.2714   0.5562    177       177       3217      pfsstop/PFSS/CLWrapCL0/G4MHPE_PE1/PECHKWP/sys_clk_c_FUSA_IN_PORT_BUF/Y pfsstop/PFSS/
LB_AC_mtgn0t11 0.8940   0.5202    81        946       853       pfsstop/PFSS/PFSS_SYSTEM/PFSS_SYSTEM_CTRL/system_pe_PE1m/micg_gate_1/micg_gate_0/gcl
LB_AC_mtgn0t30 1.6415   0.9715    345       8794      27        systop/SYSCTL/sysvdd/cggtop/gck1b_1/gck/GCLK iso0_eth/mmux_gate_01/mux2_0/D1
LB_AC_mtgn0t30 1.6067   0.9605    10        0         1         systop/SYSCTL/sysvdd/cggtop/gck1b_5/gck/GCLK dbg0_dbgtop/u_ECOB_CTS_C0ERAM_clk_mbist
LB_AC_mtgn0t30 1.6006   0.8554    20        116       31        systop/MSTBY_CG_ISO/gck_068/gck/GCLK iso0_flxtop/flx0/ip/product_if/clock_cntrl/micg
LB_AC_mtgn0t30 1.2296   0.7760    235       4021      1280      systop/sysglue_isovdd/clk_hbus_buff/buf_0/Y pfssglue/clk_gate_lockn_reg_latch/CLK
LB_AC_mtgn0t30 1.3586   0.5750    665       5273      11190     systop/sysglue_isovdd/clk_hbus_m_buff/buf_0/Y pfsstop/PFSS/DTS_TOP/RH85G2DTS/clk_FUS
```

5. CLOCK TREE SYNTHESIS FLOW

5.2 CTS step: CTS execute

From Atom:

```
#from ATOM
synthesize_clock_trees -clocks [remove_from_collection [all_clocks] [get_clocks {LB_DC_TCK* TCK* cgm_clk_postck cgm_clk* cgmawo_cgm_clka_hsosc_10m_awo}]]


cts_qor_check ${REPORTS_DIR} ${CLOCK_OPT_CTS_BLOCK_NAME}_after_synthesis_atom
save_block -as synthesis_atom

remove_clock_balance_groups -all
create_clock_balance_group -name ATOM_all -objects [get_object_name [remove_from_collection [get_clocks {LB_DC_TC_* LB_DC_EDT* LB_DC_TT* LB_DC_s* LB_DC_m* }] [get_clocks {LB_DC_mtgn0t00 }]]]

balance_clock_groups
cts_qor_check ${REPORTS_DIR} ${CLOCK_OPT_CTS_BLOCK_NAME}_after_balance_atom
save_block -as balance_atom

mark_clock_trees -synthesized -fix_sinks -clocks [remove_from_collection [all_clocks] [get_clocks {LB_DC_TCK* TCK* cgm_clk_postck cgm_clk* cgmawo_cgm_clka_hsosc_10m_awo}]]
mark_clock_trees -dont_touch -fix_sinks  -clocks [remove_from_collection [all_clocks] [get_clocks {LB_DC_TCK* TCK* cgm_clk_postck cgm_clk* cgmawo_cgm_clka_hsosc_10m_awo}]]

set_dont_touch_network -clock_only [get_pins {systop/SYSCTL/sysvdd/atom/MASTERTGN_INST/tt_ckgt_gt_tt_cl_aocd00_SO/Y}]
set_dont_touch_network -clock_only [get_pins {systop/SYSCTL/sysvdd/atom/MASTERTGN_INST/gt_tc_cl_aocd00_SO/Y}]
set_dont_touch_network -clock_only [get_pins {systop/SYSCTL/sysvdd/atom/MASTERTGN_INST/gt_edt_cl_aocd00_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/stgn0_inst/t0_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/stgn0_inst/t1_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/stgn1_inst/t0_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/stgn2_inst/t0_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/stgn3_inst/t0_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/stgn4_inst/t0_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/stgn4_inst/t1_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/stgn5_inst/t0_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/mtgn0_inst/t0_0_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/mtgn0_inst/t1_0_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/mtgn0_inst/t1_1_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/mtgn0_inst/t1_2_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/mtgn0_inst/t2_0_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/mtgn0_inst/t3_0_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/mtgn0_inst/t4_0_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/mtgn0_inst/t5_0_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/mtgn0_inst/t5_1_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/mtgn0_inst/t5_2_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/mtgn0_inst/t6_0_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/mtgn0_inst/t6_1_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/mtgn0_inst/t6_2_ckgt_gtx_ck_tx_SO/Y}]
set_dont_touch_network -clock_only [get_pins { systop/SYSCTL/sysvdd/atom/mtgn0_inst/t7_0_ckgt_gtx_ck_tx_SO/Y}]
```

5. CLOCK TREE SYNTHESIS FLOW

5.2 CTS step: CTS execute

From root

```
#synthesize_clock_trees
synthesize_clock_trees -clocks [get_clocks {LB_DC_TCK* TCK* cgm_clk_postck cgm_clk* cgmawo_cgm_clka_hsosc_10m_awo}]
cts_qor_check ${REPORTS_DIR} ${CLOCK_OPT_CTS_BLOCK_NAME}_after_synthesis
save_block -as synthesis

remove_clock_balance_groups -all
source -e -v -c ./CTS_env_u2a6_cpgm_all/balance_group.tcl
balance_clock_groups
cts_qor_check ${REPORTS_DIR} ${CLOCK_OPT_CTS_BLOCK_NAME}_after_balance
save_block -as balance

route_group -all_clock_nets -reuse_existing_global_route true -route_nondefault_nets_first true
cts_qor_check ${REPORTS_DIR} ${CLOCK_OPT_CTS_BLOCK_NAME}_after_routegroup
save_block -as routegroup

#ccs
set_app_options  -list {time.enable_ccs_rcv_cap true}
set_app_options  -list {time.delay_calc_waveform_analysis_mode full_design}

synthesize_clock_trees -postroute
cts_qor_check ${REPORTS_DIR} ${CLOCK_OPT_CTS_BLOCK_NAME}_after_routed
save_block -as routed

synthesize_clock_trees -postroute
cts_qor_check ${REPORTS_DIR} ${CLOCK_OPT_CTS_BLOCK_NAME}_after_routed2
save_block -as routed2
```

5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

There are following quality checks for the CTS:

+ Trans/cap (DRC), Vth clock (Type cells), no noise/xtalk, no HALO violation.

+ All constraint, NDR, Halo,... are applied.

+ Timing quality possible to continue fixing at PostCTS.

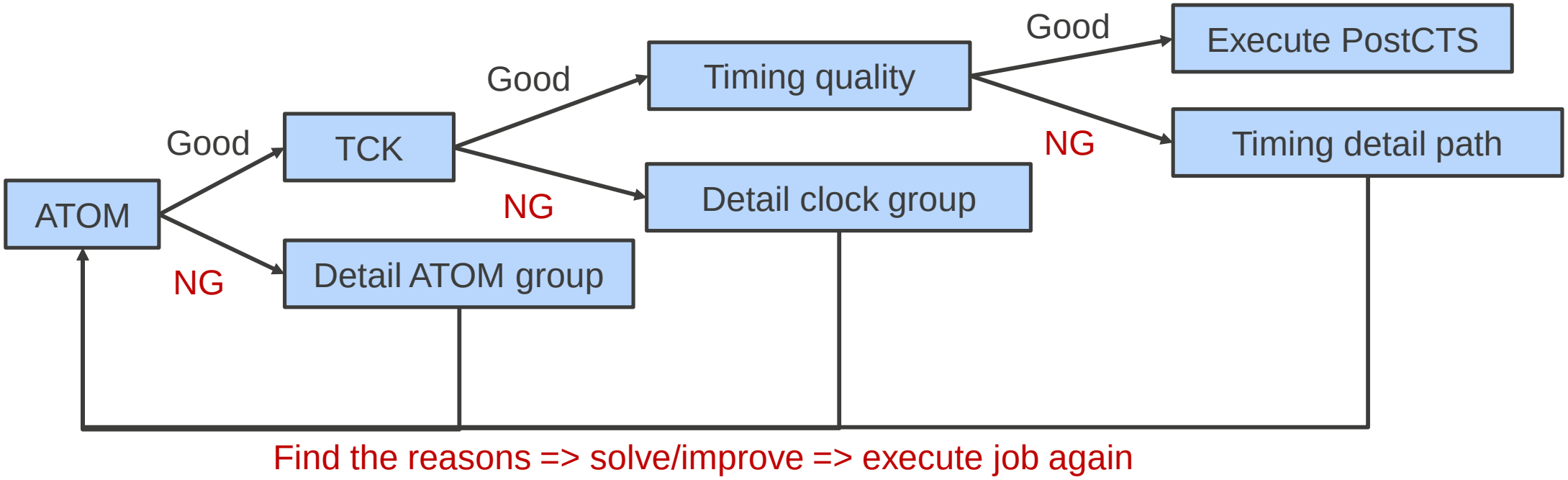
+ Check from file RPT to get WNS, TNS, detail path (make sure timing quality is achived)

Note: After CTS job's done, check reports latency and skew of design. If the results is not good (refer the old project to determine the target), check logs file and structure of the clock which cause large skew to find the reasons (large latency/skew affect timings).

5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

At synthesize 2 step, we need check reports **latency/skew** as this process:



5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

The violation timing maybe come from below reason:

- Clock skew propagated: check clock skew at reports latency/skew and find the reason why it large because skew from CTS ~ skew timing.
- Big delay cells: why cells had the big delay (fanout, placement, drive-ability,...).
- Zigzag/detour: check placement why it make zigzag/detour (cell fixed not right, intention from AC/CTS/PnR team,...).
- T.B.D

5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

Latency/skew from ATOM after synthesize:

Summary Table for Corner MAXHTSETUP

Clock / Skew Group	Attrs	Sinks	Levels	Clock Repeater Count	Clock Repeater Area	Clock Stdcell Area	Max Latency	Global Skew	Trans DRC Count	Cap DRC Count
### Mode: CTS, Scenario: CTS_MAXHTSETUP										
TCK_S	M,D	1060540	90	45287	61151.33	192556.42	2.30	2.19	9	10
LB_DC_TCK_root1	G	1060540	81	45287	61151.33	192546.23	2.30	2.42	9	10
LB_DC_TCK_root2	G	1058081	80	45177	60998.79	181920.36	2.30	2.34	9	10
LB_DC_TCK	G	1054993	75	45005	60766.16	191780.14	2.30	2.36	9	9
LB_DC_mtgn0t11	G	427576	47	15804	23756.43	65027.31	2.21	0.18	0	3
LB_DC_mtgn0t30	G	187168	34	6672	9101.34	35513.18	1.85	0.13	1	0
LB_DC_mtgn0t50	G	166862	65	7164	9468.82	38389.44	1.72	0.13	1	1
LB_DC_mtgn0t60	G	91049	36	3391	4518.53	26211.90	2.11	0.19	0	3
LB_DC_mtgn0t40	G	67048	36	3062	4324.66	10826.37	1.97	0.12	0	0
LB_DC_TT_TR	G	63286	39	5420	5475.01	7228.93	2.11	0.21	3	2
LB_DC_EDT_TR	G	14235	23	729	893.31	898.69	1.37	0.07	1	1
LB_DC_stgn0t0	G	12802	34	940	1208.26	2898.78	1.84	0.14	0	0
LB_DC_cgg_mosc40_div4	G	10323	25	413	581.17	2027.09	1.37	0.11	0	0
LB_DC_cgg_mosc40_div2	G	5254	25	284	393.34	1026.37	1.34	0.07	0	0
LB_DC_cgg_mosc40_div8	G	5069	14	127	188.27	1001.17	1.01	0.11	0	0
LB_DC_mtgn0t61	G	9664	29	491	568.51	1946.67	1.60	0.17	1	1
LB_DC_stgn1t0	G	9251	15	384	431.98	1979.04	0.76	0.06	0	0
LB_DC_TC_TR	G	9143	23	688	875.28	942.48	1.20	0.11	0	0
LB_DC_mtgn0t20	**G**	**8366**	**40**	**680**	**908.10**	**2198.56**	**2.30**	**0.21**	**0**	**0**
LB_DC_mtgn0t70	G	7894	17	519	764.51	1185.18	0.90	0.11	1	0
LB_DC_stgn0t1	G	5255	26	296	398.72	1037.68	1.38	0.07	0	0
LB_DC_stgn4t0	G	4513	33	321	425.26	546.90	2.08	0.18	0	0
LB_DC_stgn4t1	G	2182	21	199	215.26	324.91	1.24	0.18	1	0
LB_DC_stgn3t0	G	2173	34	197	173.38	502.43	1.54	0.15	1	1
LB_DC_mtgn0t62	G	2150	31	268	298.70	515.76	1.74	0.05	0	0
LB_DC_mtgn0t52	G	1122	34	430	522.26	680.51	1.84	0.07	0	0
LB_DC_stgn5t0	G	190	17	87	69.89	116.26	0.90	0.02	0	0
LB_DC_mtgn0t00	G	47	4	1	1.79	9.52	0.20	0.03	0	0
LB_DC_stgn2t0	G	45	6	10	13.78	19.60	0.36	0.02	0	0
LB_DC_mtgn0t12	G	5	7	1	1.79	12.10	0.40	0.04	0	0
LB_DC_mtgn0t51	G	1	3	126	186.48	1001.50	0.11	0.00	0	0
LB_DC_mtgn0t10	G	0	0	0	0.00	3.81	--	--	0	0

Confirm the largest latency/skew in the report and cases relative. Why is it largest, is it detour/zigzag?
You can see the longest path (which maybe the path is shown in the report latency or log file or you need trace in schematic).

5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

The report latency will shown you the longest and smallest path, you can depend on it and trace in schematic to see if it's detour or zigzag. (imported path pins or trace manual).

```
LB_DC_mtgn0t20
      L   pfsstop/PFSS/SAXI_Wrap/uabsdaxi3ic_saxiic_p2f2eval/core/axi3divrs_icum/uR/dat_reg_1__72_/CLK      2.30 r      2.30 r        --        --
      L   pfsstop/PFSS/SAXI_Wrap/uabsdaxi3ic_saxiic_p2f2eval/core/axi3divrs_icum/uR/dat_reg_1__59_/CLK      2.30 r      2.30 r        --        --
      L   pfsstop/PFSS/SAXI_Wrap/uabsdaxi3ic_saxiic_p2f2eval/core/axi3divrs_icum/uR/dat_reg_1__65_/CLK      2.30 r      2.30 r        --        --
      L   pfsstop/PFSS/SAXI_Wrap/uabsdaxi3ic_saxiic_p2f2eval/core/axi3divrs_icum/uR/dat_reg_1__77_/CLK      2.30 r      2.30 r        --        --
      L   pfsstop/PFSS/SAXI_Wrap/uabsdaxi3ic_saxiic_p2f2eval/core/axi3divrs_icum/uR/dat_reg_1__64_/CLK      2.30 r      2.30 r        --        --
      S   pfsstop/C3RAMM/C3RMMEM_W2/SRAM/L2CRAM_RS_32KB_LL_nomux/ia_0_0/i0/awo_SRAM/i0/CLK      2.09 r      2.09 r     -0.08     -0.08
      S   pfsstop/PFSS/SAXI_Wrap/uabsdaxi3ic_saxiic_p2f2eval/core/axi3divrs_sdmac1/uW/ptr_reg_1_/CLK      2.09 r      2.09 r        --        --
      S   pfsstop/PFSS/SAXI_Wrap/uabsdaxi3ic_saxiic_p2f2eval/core/axi3divrs_sdmac1/uW/ptr_reg_0_/CLK      2.09 r      2.09 r        --        --
      S   dbg0_dbgtop/C0ERAM_wp/M10/ib_0_6/i0/obs_enbl_reg/CLK      2.09 r      2.09 r     -0.08     -0.08
      S   dbg0_dbgtop/C0ERAM_wp/M10/ib_0_4/i0/obs_enbl_reg/CLK      2.09 r      2.09 r     -0.08     -0.08
```

There is a path called bottleneck (which is the root cause the large latency), tool will try to balance all sink but this path cause stack. It's ***maybe*** not the longest path is shown in the report, because tool will increase max size of cells in this clock path to decrease delay. We need to confirm this.

5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

You could find the longest path in the log file, which is:

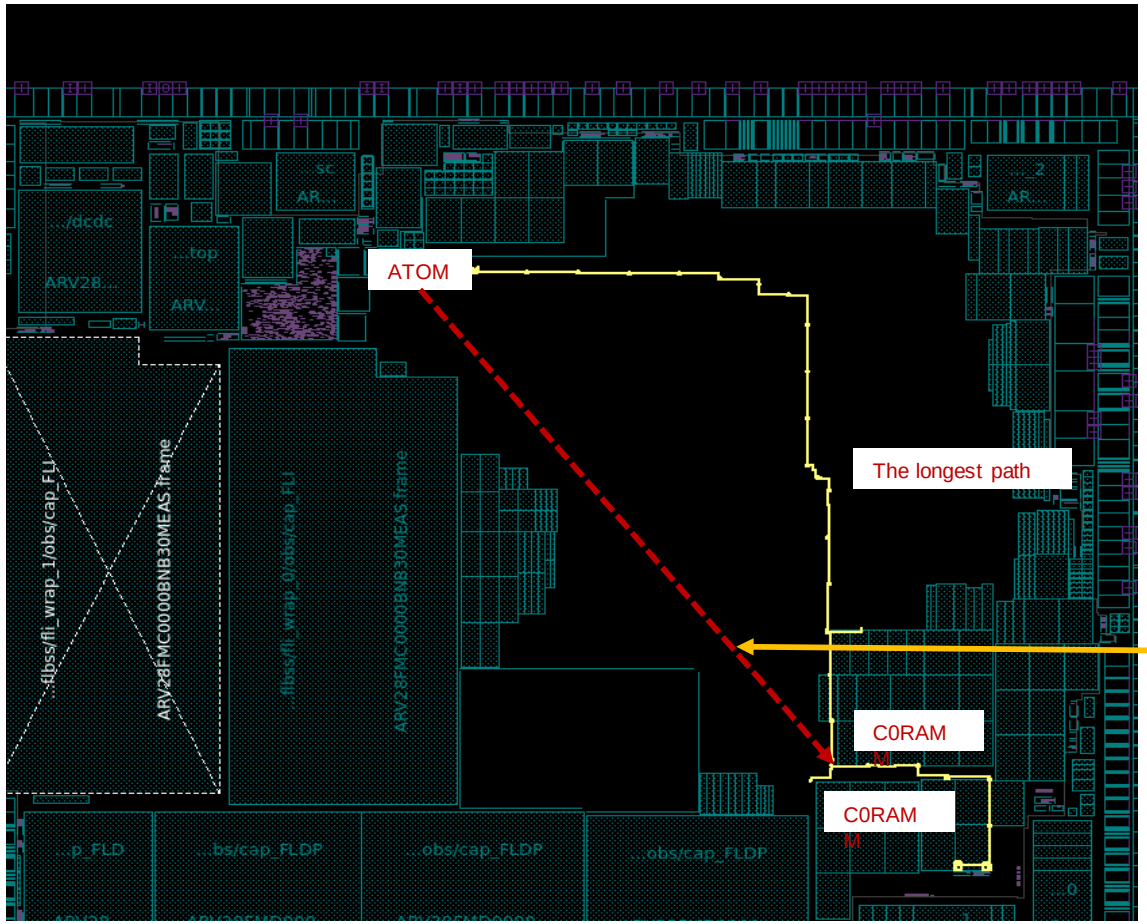
```
* CTS STEP: Latency Bottleneck Analysis
*********************************************************
* PD jump Gates (sorted by PD jump): Largest latency jumps (> 0.500000)

Clock          Phase     PD Jump    Buf/inv    Sinks      Gates      From Pin  To Pin
Root           Delay
-------------------------------------------------------------------------------------------------------
io_top/iso0_iotop/iso_scaap_e2_P24_10/scaap_bs_BSR0386_G/G4/D0
AC_mspi_20_sck_gen_1_2 0.8279 0.8279 16      1          0          io_top/iso0_iotop/iso_scaap_e1/scaap_tk_edt_channels_out_7_G/Y io_top/iso0_iotop/io_e1vcc/IO_P10_3/DOUT
AC_mspi_40_sck_in_1_3b 1.1393 0.6540 12      0          1          io_top/iso0_iotop/iso_scaap_e0/scaap_esd_P4_9_LVDS_I_COUTLV_ESD_G/Y iso0_pfctop/glue_lvds/U5/A
DCDCCKDIV8_AWO 1.0802    1.0802     20        1          0          io_top/iso0_iotop/iso_scaap_e1_P10_8/scaap_tk_edt_channels_out_11_G/Y io_top/iso0_iotop/io_e1vcc/IO_P10_8/P2BR0XRRFHLLY_R0/DOUT
DCDCCKDIV8_AWO 3.0522    0.5758     4         0          1          systop/dcdc_top/Diode_buf_cell_156/Y systop/dcdc_top/dcdc_awo/U22/B
GTMECLK2       1.0675    1.0675     20        1          0          io_top/iso0_iotop/iso_scaap_e1_P17_6/scaap_bs_BSR0175_G/G4/Y io_top/iso0_iotop/io_e1vcc/IO_P17_6/P2BR0XRRFHLLY_R0/DOUT
LB_AC_mtgn0t11 2.2946    1.1450     1299      2053       8171       systop/sysglue_isovdd/clk_cpu_buff/buf_0/Y
pfsstop/PFSS/PFSS_SYSTEM/PFSS_SYSTEM_CTRL/system_pe_PE1m/micg_gate_0/micg_gate_0/gck/CLK
LB_AC_mtgn0t11 0.9107    0.5838     242       1322       3381       pfsstop/PFSS/PFSS_SYSTEM/PFSS_SYSTEM_CTRL/DBGSS_CLK_CTRL/micg_gate_0/gck/GCLK
pfsstop/PFSS/Debug_Wrap/debugss/TCU_LTB/BUF_IN/clock_gate_i1d_reg_19/
LB_AC_mtgn0t11 0.9468    0.5800     59        787        36         iso0_memtop/fsy0/flbss0_flbss/flbss_cgc/cgc1/gck/GCLK
iso0_memtop/fsy0/flbss0_flbss/rbist_fli/br_1__inst_bridge/inst_bridge_u/clk_gate_misr_data_ff
LB_AC_mtgn0t20 2.2689    0.6627     16        23         3          systop/SYSCTL/sysvdd/cggtop/gck1a_0/gck/GCLK systop/sysglue_isovdd/clk_sbus_buff/buf_0/A
```

After synthesize clock tree, you can find the large jump at *CTS STEP: Latency Bottleneck Analysis* to trace in schematic to find the reason (why phase delay jump large? Why tool insert more repeater at this path?).

5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

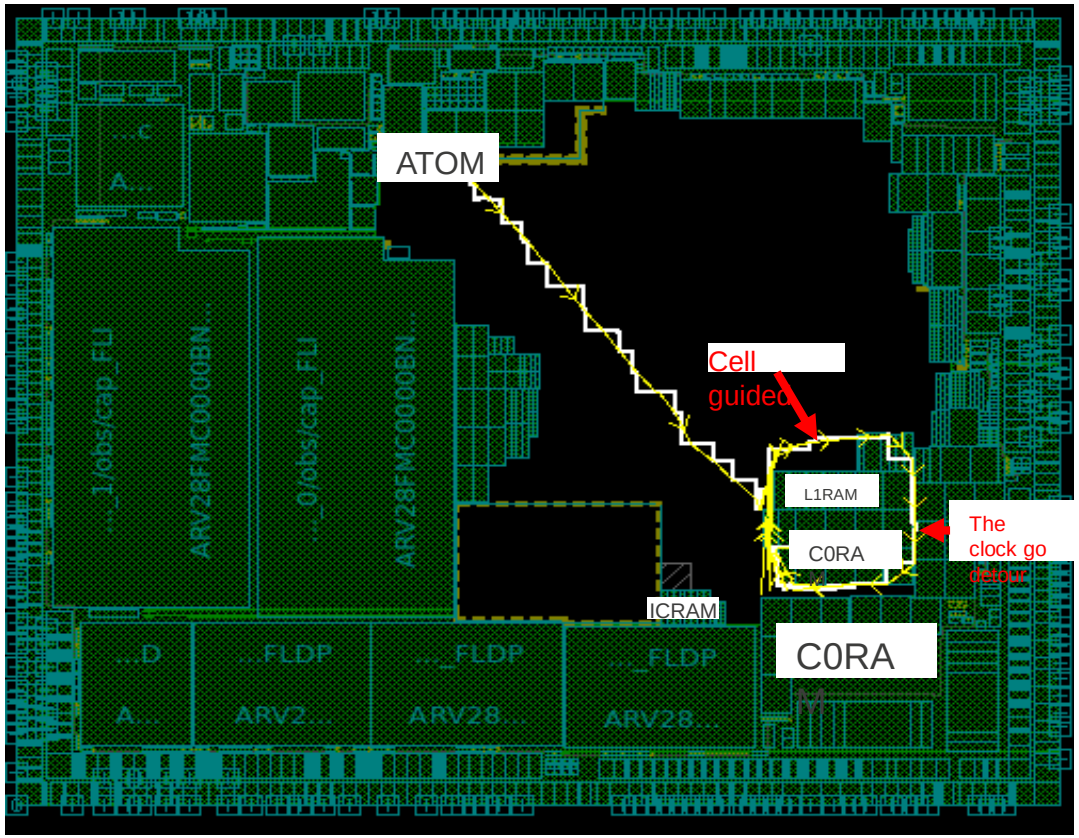


You can trace in schematic and see the path in layout, this longest path made latency is large, so we need consider to guide buffer to decrease the latency.

Consider to guide buffer as expeted

5. CLOCK TREE SYNTHESIS FLOW

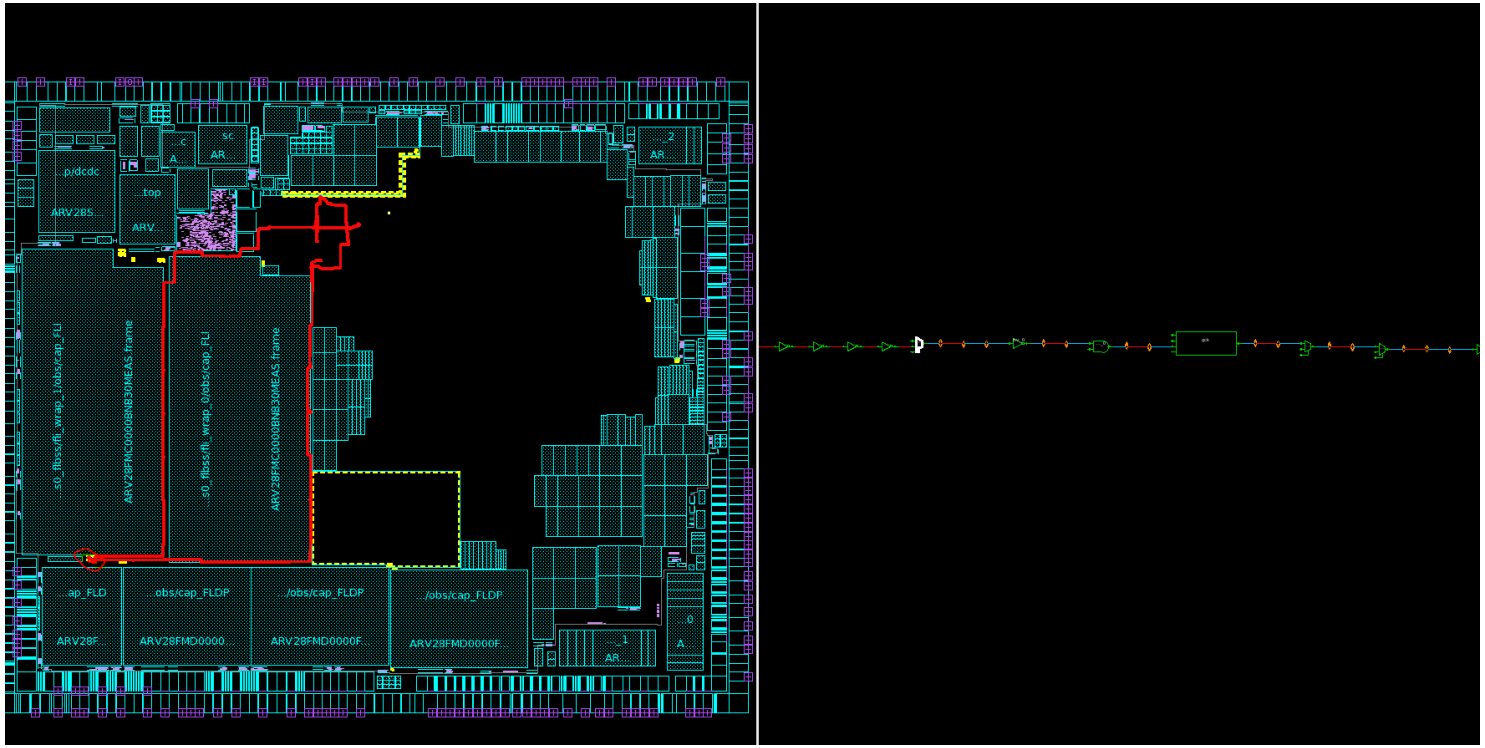
5.3 CTS step: Quality checks & debug



This path go detour due to the guide C0RAMM is not good. The path detour be the bottleneck lead tool stack to balance related path.
=> Re-guide buffer to C0RAMM, re-arrange CRAMM location,

5. CLOCK TREE SYNTHESIS FLOW

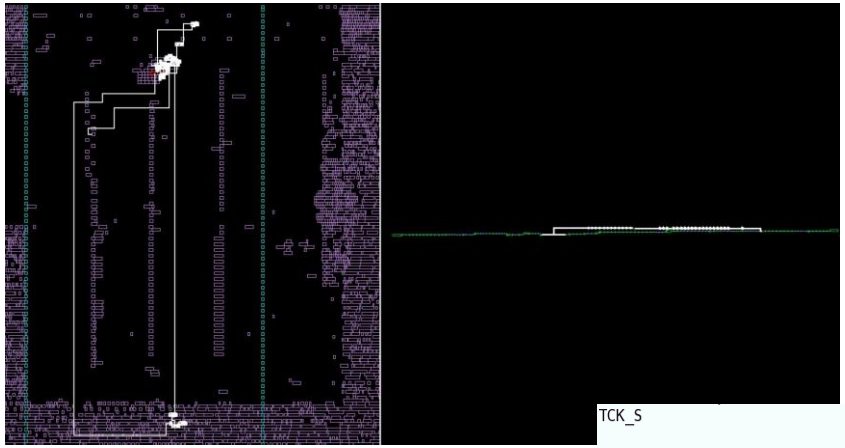
5.3 CTS step: Quality checks & debug



This mux fixed cause detour => unfixed for automic place/ ignore pin/ re-placed as expected.

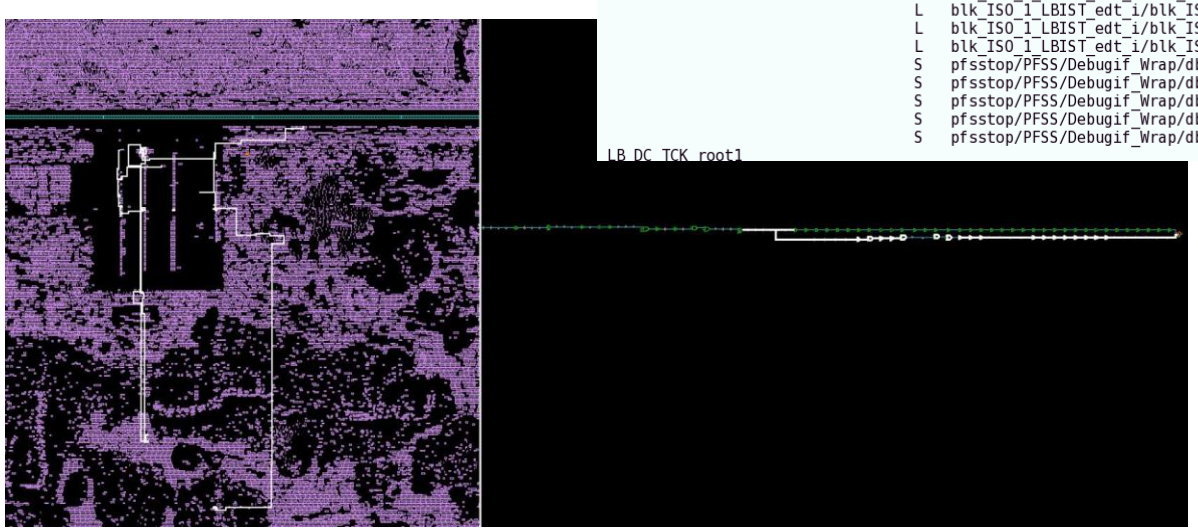
5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug



This path is derived from master clock and dive 2 path in the middle then combined one path to sink. So we need balance the up one and the down one.
However to the up one, the tool insert cells near together make the net delay is small => the skew between it is large (~0.4ns). So I insert 1 cell same as the down one for the up path, my intention is the tool will insert like the down one to balance this path. But the result is:

L	blk_ISO_1_LBIST_edt_i/blk_ISO_1_LBIST_edt_compactor_i/compactor1/lbist_misr_in_reg_18_/CLK	4.60 r	4.60 r	0.06	0.06
L	blk_ISO_1_LBIST_edt_i/blk_ISO_1_LBIST_edt_compactor_i/compactor1/lbist_misr_in_reg_20_/CLK	4.60 r	4.60 r	0.06	0.06
L	blk_ISO_1_LBIST_edt_i/blk_ISO_1_LBIST_edt_compactor_i/compactor1/lbist_misr_in_reg_22_/CLK	4.60 r	4.60 r	0.06	0.06
L	blk_ISO_1_LBIST_edt_i/blk_ISO_1_LBIST_edt_compactor_i/compactor1/lbist_misr_in_reg_23_/CLK	4.60 r	4.60 r	0.06	0.06
L	blk_ISO_1_LBIST_edt_i/blk_ISO_1_LBIST_edt_compactor_i/compactor1/lbist_misr_in_reg_38_/CLK	4.60 r	4.60 r	0.06	0.06
S	pfsstop/PFSS/Debugif_Wrap/dbgifss/JCU/JCU_JTAG_IF/bypass_reg/CLK	3.83 r	3.69 r	--	--
S	pfsstop/PFSS/Debugif_Wrap/dbgifss/JCU/JCU_JTAG_IF/rdy_hold1_reg/CLK	3.83 r	3.69 r	--	--
S	pfsstop/PFSS/Debugif_Wrap/dbgifss/JCU/JCU_JTAG_IF/sel_wrback_reg/CLK	3.83 r	3.69 r	--	--
S	pfsstop/PFSS/Debugif_Wrap/dbgifss/JCU/JCU_JTAG_IF/cycle_t2w_reg/CLK	3.83 r	3.69 r	--	--
S	pfsstop/PFSS/Debugif_Wrap/dbgifss/JCU/JCU_JTAG_IF/ir_reg_7_/CLK	3.84 r	3.69 r	--	--



=> Insert not right eventhough only 1 cell lead tool give up balance.

5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

CHECK TIMING AFTER CTS

First, to overview the timing after CTS, you can see in this file: CTS_propagate_report_qor.rep

```
****************************************
Report : qor
         -summary
Design : u2a6
Version: R-2020.09-SP4
Date   : Sun Apr 25 17:16:03 2021
****************************************
Information: Timer using 'AWP, CRPR'. (TIM-050)

Timing
--------------------------------------------------------------------------------
Context                                 WNS             TNS             NVE
--------------------------------------------------------------------------------
FBIST_MAXLTSETUP    (Setup)           -4.31          -83.04             201
MBIST_MAXLTSETUP    (Setup)           -0.57          -65.96             986
SCAN_MAXLTSETUP     (Setup)           -6.56       -11110.15          106920
USER_MAXLTSETUP     (Setup)           -8.79       -11830.41          116271
Design              (Setup)           -8.79       -12934.88          122000

FBIST_MINHTHOLD     (Hold)            -0.64         -409.58           10780
MBIST_MINHTHOLD     (Hold)            -2.29          -51.40             551
SCAN_MINHTHOLD      (Hold)            -0.64         -270.77            9293
USER_MINHTHOLD      (Hold)            -2.29         -449.97            3977
Design              (Hold)            -2.29         -829.31           12588
--------------------------------------------------------------------------------
```

Check if it could optimize at postCTS or not. Why is it violated?

5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

CHECK TIMING AFTER CTS

Check file: CTS_propagate_USER_MAXLTSETUP.slack to overview which group timing path is violated. CTS care internal path and AC team care in/out path. Check if it have room to fix or not. Why is it violated?

###	CLOCK	NegativeSlackPathNum	WorstNegativeSlack	TotalNegativeSlack
	in2reg_default	102	-8.7941	-106.0018
	reg2out_default	56	-5.3622	-118.0130
	REG2REG	10000	-6.7421	-2064.9998
	REG2MEM	2275	-0.4052	-151.9504
	MEM2REG	1372	-0.4259	-142.7311
	REG2ICG	3025	-3.0786	-1286.7205
	CGL_CGG	6	-0.2826	-0.8843
	CL0toCL0	10000	-1.8374	-2706.4158
		26836	-8.7941	-6577.7167

5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

CHECK TIMING AFTER CTS

First, short timing violated as skew, check why it have large skew,
is it intention from AC/CTS team or it the gap skew from synthesize clock tree?

```
TCK(6.2007)  TCK'(2.3983)  -3.8024  pfsstop/PFSS/Debugif_Wrap/dbgifss/JCU/JCU_JTAG_IF/d0tms_reg pfsstop/PFSS/Debugif_Wrap/dbgifss/JCU/JCU_JTAG_IF/d0tdo_reg/DATA        8       -1.0423
AC_mspi_40_sck_in_1_3b(6.6116)  AC_mspi_40_sck_in_1_3b'(3.5861)  -3.0255  iso0_mspi1top/mspi/comctrl/rx/rx_rxtglstate_r_reg     iso0_mspi1top/mspi/sinclone/clone3/sin1_f_reg/DATA      8       -2.2674
AC_mspi_40_sck_in_1_3b(6.6116)  AC_mspi_40_sck_in_1_3b'(3.5861)  -3.0255  iso0_mspi1top/mspi/comctrl/rx/rx_rxtglstate_r_reg     iso0_mspi1top/mspi/sinclone/clone3/sin2_f_reg/DATA      9       -2.6343
AC_mspi_40_sck_in_1_3b'(6.4983)  AC_mspi_40_sck_in_1_3b(3.6687)  -2.8296  iso0_mspi1top/mspi/comctrl/rx/rx_rxtglstate_r_reg     iso0_mspi1top/mspi/sinclone/clone3/sin1_r_reg/DATA      8       -2.0784
AC_mspi_40_sck_in_1_3b'(6.4983)  AC_mspi_40_sck_in_1_3b(3.6687)  -2.8296  iso0_mspi1top/mspi/comctrl/rx/rx_rxtglstate_r_reg     iso0_mspi1top/mspi/sinclone/clone3/sin2_r_reg/DATA      8       -2.4225
TCK'(2.7852)  TCK(0.0000)  -2.7852  pfsstop/PFSS/Debugif_Wrap/dbgifss/JCU/JCU_JTAG_IF/d0tdo_reg JP0_1   23      -0.0404
AC_mspi_40_sck_in_1_3b'(6.2975)  AC_mspi_40_sck_in_1_3b(3.6575)  -2.1871  iso0_mspi1top/mspi/comctrl/stx/r_stxdcnt_r_reg_3_     iso0_mspi1top/mspi/comctrl/stx/stxclone3/r_data0_r_reg/DATA     9       -1.0731
AC_mspi_40_sck_in_1_3b(6.2657)  AC_mspi_40_sck_in_1_3b(3.6575)  -2.1553  iso0_mspi1top/mspi/comctrl/stx/stx_setstate_r_reg      iso0_mspi1top/mspi/comctrl/stx/stxclone3/r_sout_r_reg/DATA      10      -1.5348
clk_cpu_pllclk800(5.2822)  clk_cpu_pllclk800(3.4189)  -1.6748  pfsstop/PFSS/PFSS_SYSTEM/PFSS_SYSTEM_CTRL/PFSS_U2A6_system/PE1_DISABLE_LF_reg_reg        pfsstop/PFSS/PFSS_SYSTEM/PFSS_SYSTEM_CTRL/system_pe_PE1m/m
clk_cpu_pllclk800(5.2794)  clk_cpu_c_pllclk800(3.6687)  -1.4570  pfsstop/PFSS/PFSS_SYSTEM/PFSS_SYSTEM_CTRL/PFSS_U2A6_system/PE1C_DISABLE_LF_reg_reg     pfsstop/PFSS/PFSS_SYSTEM/PFSS_SYSTEM_CTRL/system_pe_PE1c/m
clk_cpu_pllclk800(5.2777)  clk_cpu_c_pllclk800(3.7495)  -1.3745  pfsstop/PFSS/PFSS_SYSTEM/PFSS_SYSTEM_CTRL/PFSS_U2A6_system/PE0C_DISABLE_LF_reg_reg     pfsstop/PFSS/PFSS_SYSTEM/PFSS_SYSTEM_CTRL/system_pe_PE0c/m
clk_div2_sys_pllclk800(4.4545)  cgm_clk_hbus_pllclk800(3.0026)  -1.3662  systop/SYSCTL/sysvdd/cgltop/cgl_stbyckstp/cgl_ckstp_flxa_a_reg systop/SYSCTL/sysvdd/cggtop/gck1b_6/gck/CEN     3       -0.2275
clk_cpu_pllclk800(4.9172)  clk_cpu_pllclk800(3.4188)  -1.3496  pfsstop/PFSS/PFSS_SYSTEM/PFSS_SYSTEM_CTRL/DBGSS_CLK_CTRL/DCUDISABLE_reg_reg      pfsstop/PFSS/PFSS_SYSTEM/PFSS_SYSTEM_CTRL/DBGSS_CLK_CTRL/micg_gate
clk_div2_sys_pllclk800(4.4545)  cgm_clk_h_pllclk800(3.1091)  -1.2597  systop/SYSCTL/sysvdd/cgltop/cgl_stbyckstp/cgl_ckstp_flxa_a_reg    systop/SYSCTL/sysvdd/cggtop/gck2a_9/gck/CEN     3       -0.0855
clk_div2_sys_pllclk800(3.6227)  cgm_clk_sbus_pllclk800(2.3034)  -1.2336  systop/SYSCTL/sysvdd/cgltop/cgl_stbyckstp/cgl_ckstp_a_reg      systop/SYSCTL/sysvdd/cggtop/gck1a_0/gck/CEN     3       -0.1742
clk_div2_sys_pllclk800(4.1965)  cgm_clk_l_pllclk800(2.9144)  -1.1964  systop/SYSCTL/sysvdd/cgltop/cgl_stbyckstp/cgl_ckstp_adc_a_reg     systop/SYSCTL/sysvdd/cggtop/gck3_10/gck/CEN     7       -0.0722
clk_div2_sys_pllclk800(3.6227)  cgm_clk_c_pllclk800(2.3426)  -1.1944  systop/SYSCTL/sysvdd/cgltop/cgl_stbyckstp/cgl_ckstp_a_reg systop/SYSCTL/sysvdd/cggtop/gck0b_2/gck/CEN     5       -0.2826
clk_lsb_sys_pllclk800(5.6170)  clk_fli_cpu_pllclk800(4.3408)  -1.1905  systop/SYSCTL/sysvdd/rsstop/pre_fres_n/q_reg     systop/SYSCTL/sysvdd/rsstop/fres_c/q1_reg/RESET 6       -0.8034
clk_lsb_sys_pllclk800(5.6170)  clk_fli_cpu_pllclk800(4.3408)  -1.1905  systop/SYSCTL/sysvdd/rsstop/pre_fres_n/q_reg     systop/SYSCTL/sysvdd/rsstop/fres_c/q_reg/RESET  6       -0.7621
clk_div2_sys_pllclk800(3.6227)  cgm_clk_c_pllclk800(2.3688)  -1.1682  systop/SYSCTL/sysvdd/cgltop/cgl_stbyckstp/cgl_ckstp_a_reg systop/SYSCTL/sysvdd/cggtop/gck0a_0/gck/CEN     3       -0.0423
clk_cpu_pllclk800(4.9172)  clk_cpu_pllclk800(3.6004)  -1.1387  pfsstop/PFSS/PFSS_SYSTEM/PFSS_SYSTEM_CTRL/DBGSS_CLK_CTRL/DCUDISABLE_reg_reg      pfsstop/PFSS/PFSS_SYSTEM/PFSS_SYSTEM_CTRL/DBGSS_CLK_CTRL/micg_gate
clk_cpu_pllclk800(5.4432)  clk_cpu_pllclk800(4.3137)  -0.9514  pfsstop/PFSS/CLWrapCL0/FLISS/CCIBHV4/AXI2FLI_A/A2F_QBUF/regx_sig_QBUF0_ARADDR_Q_reg_13_  dbg0_dbgtop/C0ERAM_wp/ERAM_DP_glue/G3XSS10_ERAM_CKEFF_0/re
clk_cpu_pllclk800(5.4432)  clk_cpu_pllclk800(4.3142)  -0.9509  pfsstop/PFSS/CLWrapCL0/FLISS/CCIBHV4/AXI2FLI_A/A2F_QBUF/regx_sig_QBUF0_ARADDR_Q_reg_13_  dbg0_dbgtop/C0ERAM_wp/ERAM_DP_glue/G3XSS10_ERAM_CKEFF_1/re
clk_cpu_pllclk800(5.4434)  clk_cpu_pllclk800(4.3150)  -0.9503  pfsstop/PFSS/CLWrapCL0/FLISS/CCIBHV4/AXI2FLI_A/A2F_QBUF/regx_sig_QBUF0_ARADDR_Q_reg_12_  dbg0_dbgtop/C0ERAM_wp/ERAM_DP_glue/G3XSS10_ERAM_CKEFF_2/re
clk_cpu_pllclk800(5.6491)  clk_hbus_m_pllclk800(4.6139)  -0.9495  pfsstop/PFSS/SG0_APBIF/VCI2APB_SYNC/VQFIFO/regxe_vcq_sync_apb_Q_pre_reg_47_   pfsstop/PFSS/DTS_TOP/DTS_GUARD/guard_apbif/clock_gate_spiderr_r_re
clk_cpu_pllclk800(5.8138)  clk_cpu_c_pllclk800(4.7712)  -0.8889  pfsstop/PFSS/CLWrapCL0/G4MHPE_PE1/LSDELAYM/regxe_ICTramWAY3_WE_0_Q_pre_reg_0_   pfsstop/PFSS/CLWrapCL0/G4MHPE_PE1/PECHKWP/PECMPLS/delay_c/clock_ga
```

5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

CHECK TIMING AFTER CTS

```
  Path Group: REG2REG
  Path Type: max


Attributes
    b - black-box (unknown)
    s - size_only
    d - dont_touch
    u - dont_use
    g - generic
    h - hierarchical
    i - ideal
    n - noncombinational
    E - extracted timing model
    Q - Quick timing model


  Point                                                                                          Fanout    Cap      Trans    Derate     Incr      Path       Attributes
  ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------
  clock AC_mspi_10_sck_gen_0_1 (rise edge)                                                                                               0.0000    0.0000
  clock network delay (propagated)                                                                                                       5.7010    5.7010

  iso0_mspi0top/mspi/comctrl/master/mclone1/mst_sck_r_reg/Q (TSHSDFFQRX20)                                         0.0422    1.0560    0.0000    5.7010 r   s, d, n
  iso0_mspi0top/mspi/comctrl/master/mclone1/mst_sck_r (net)                                         2      0.0022                                                 d
  iso0_mspi0top/mspi/comctrl/master/mclone1/icc2_ctsmst_sck_r_reg_Q_btd2062986/A (TSHBUFCLX20)                     0.0420    1.0400    0.0000    5.7010 r   s
  iso0_mspi0top/mspi/comctrl/master/mclone1/icc2_ctsmst_sck_r_reg_Q_btd2062986/Y (TSHBUFCLX20)                     6.6784    1.0560    1.7876    7.4886 r   s
  iso0_mspi0top/mspi/comctrl/master/mclone1/cts0 (net)                                              3      0.8629
  iso0_mspi0top/mspi/comctrl/master/U410/B (THHOR4X10)                                                            16.9602    1.0400    6.8066   14.2952 r   s
  iso0_mspi0top/mspi/comctrl/master/U410/Y (THHOR4X10)                                                             0.1258    1.0560    1.2565   15.5517 r   s
  iso0_mspi0top/mspi/comctrl/master/n235 (net)                                                      1      0.0011
  iso0_mspi0top/mspi/comctrl/master/U412/A1 (TSHON21ZHX20)                                                         0.1258    1.0400    0.0000   15.5517 r
  iso0_mspi0top/mspi/comctrl/master/U412/YB (TSHON21ZHX20)                                                         0.7809    1.0560    0.5188   16.0705 f
  iso0_mspi0top/mspi/comctrl/master/mst_sck_r_d_3_ (net)                                            4      0.0404
  iso0_mspi0top/mspi/comctrl/master/mclone0/U3/D1 (THHMUX2X40)                                                     0.7816    1.0400    0.0253   16.0958 f   s
  iso0_mspi0top/mspi/comctrl/master/mclone0/U3/Y (THHMUX2X40)                                                      0.3604    1.0560    0.8574   16.9532 f   s
  iso0_mspi0top/mspi/comctrl/master/mclone0/n1 (net)                                                1      0.0635
  iso0_mspi0top/mspi/comctrl/master/mclone0/icc2_placeioptHFSBUF_321_811917/A (THHBUFZHHX30)                       0.3843    1.0400    0.0846   17.0378 f
  iso0_mspi0top/mspi/comctrl/master/mclone0/icc2_placeioptHFSBUF_321_811917/Y (THHBUFZHHX30)                       1.1721    1.0560    0.7251   17.7629 f
  iso0_mspi0top/mspi/comctrl/master/mclone0/icc2_placeioptHFSNET_14 (net)                           1      0.1412
  iso0_mspi0top/mspi/comctrl/master/mclone0/icc2_placefopt1ZINV_181_inst_1232281/A (THHINVZDX30)                   1.3196    1.0400    0.4470   18.2099 f
  iso0_mspi0top/mspi/comctrl/master/mclone0/icc2_placefopt1ZINV_181_inst_1232281/YB (THHINVZDX30)                  1.5493    1.0560    1.7326   19.9425 r
  iso0_mspi0top/mspi/comctrl/master/mclone0/icc2_placefopt1ZINV_181_4346 (net)                      1      0.1039
  iso0_mspi0top/mspi/comctrl/master/mclone0/icc2_placefopt1ZBUF_89_inst_1232279/A (THHBUFZHHX30)                   1.5815    1.0400    0.2292   20.1717 r
  iso0_mspi0top/mspi/comctrl/master/mclone0/icc2_placefopt1ZBUF_89_inst_1232279/Y (THHBUFZHHX30)                   0.9072    1.0560    1.1406   21.3124 r
  iso0_mspi0top/mspi/comctrl/master/mclone0/icc2_placefopt1ZBUF_89_4346 (net)                       1      0.0746
  iso0_mspi0top/mspi/comctrl/master/mclone0/icc2_placefopt1ZINV_23_inst_1232278/A (THHINVZHHX30)                   0.9191    1.0400    0.1082   21.4205 r
  iso0_mspi0top/mspi/comctrl/master/mclone0/icc2_placefopt1ZINV_23_inst_1232278/YB (THHINVZHHX30)                  0.7708    1.0560    0.9361   22.3566 f
  iso0_mspi0top/mspi/comctrl/master/mclone0/icc2_placefopt1ZINV_23_4346 (net)                       1      0.0672
  iso0_mspi0top/mspi/comctrl/master/mclone0/mst_sck_r_reg/DATA (TSHSDFFQRX20)                                      0.7792    1.0400    0.0704   22.4270 f   s, d, n
  data arrival time                                                                                                                                22.4270

  clock clka_mspi (rise edge)                                                                                                          11.3400   11.3400
  clock network delay (propagated)                                                                                                      5.1181   16.4581
  clock reconvergence pessimism                                                                                                         0.2128   16.6709
  iso0_mspi0top/mspi/comctrl/master/mclone0/mst_sck_r_reg/CLK (TSHSDFFQRX20)                                       0.0566    1.0000    0.0000   16.6709 r   s, d, n
  clock uncertainty                                                                                                                    -0.5000   16.1709
  library setup time                                                                                                         1.0000    -0.4860   15.6849
  data required time                                                                                                                               15.6850
  ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------
  data required time                                                                                                                               15.6850
  data arrival time                                                                                                                               -22.4270
  ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------
  slack (VIOLATED)                                                                                                                                 -6.7421
```

The slack is -6.75. Check the Incr that if size cell will it optimize the slack and meet timing?

5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

Fix minpulse violation

Pulse is the high edge/low edge in 1 period clock. It's can be undenstand as the time to keep at edge high/low in 1 period clock.

Minpulse is the target to keep high/low edge in 1 period clock to ensurance circuit will receive clock signal from source to sink.

Minpulse violation will make the circuit can't realize which is high/low edge => active wrong.

5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

Fix minpulse violation

Guide fixing min pulse:

Step 1: **restore_session** $*STA_release_dir**/work_minpulse_ss/LOAD**/$scenario_violated*

Step 2: report_constraint -min_pulse_width -all_violators (get all violators)

(**get detail violator**: *report_min_pulse_width [get_pin **systop/SYSCTL/sysvdd/cgmtop/cgmcmdiv_osc/cdiv2/pll_buf/buf_0/Y**] -path_type full_clock_expand -nosplit -all_violators)*

Source this script to get the proc at Ptime (same alias in ICC2):
/svhome/VCF_RVC_OffSite/mydiemnguyen_p2f2_vf/code_My/wkdir.forMy/aliasPT.tcl

rpm: get all violators minpulse

rpma: get a pin violation
Get violation 1 pins and report path violation => find the INV (if haven't got INV, find BUF) which is drive the violate pin and **size cell (note:** just size only cell and these cell are inserted by ECO/tools, don't have to size logic orginal)

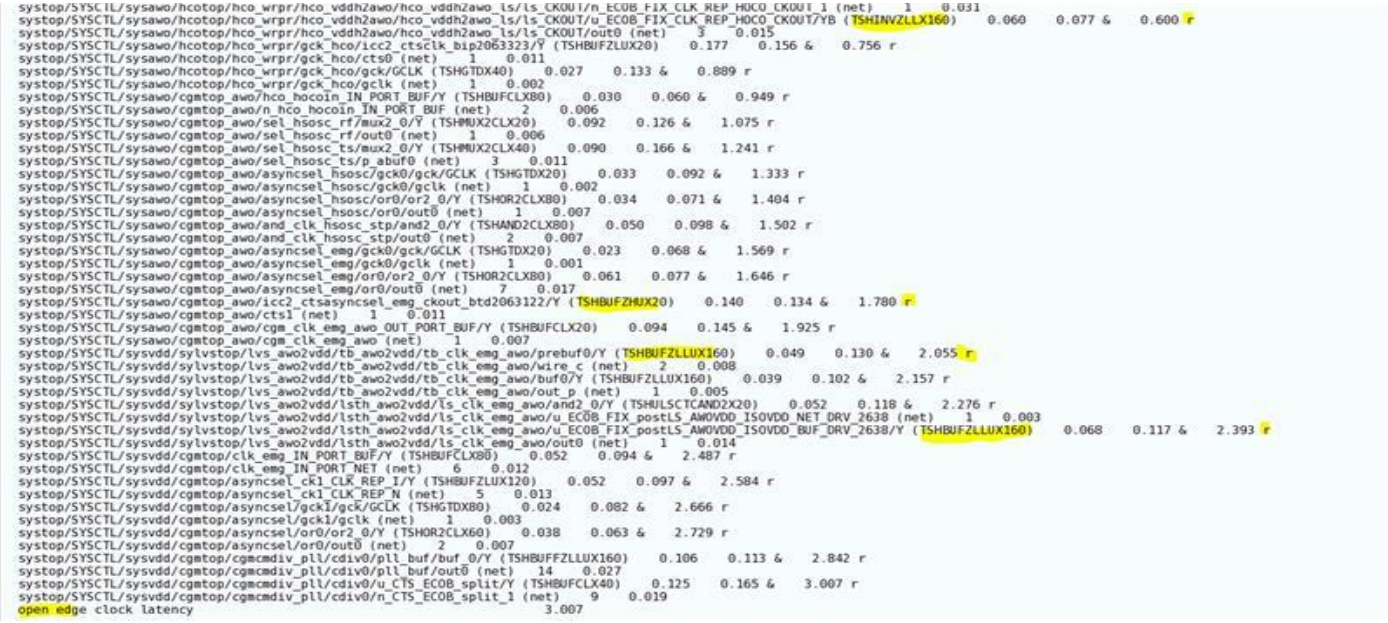
5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

Fix minpulse violation

```
clock reconvergence pessimism                                    0.000     3.284
clock uncertainty                                               -0.100     3.184
close edge clock latency                                                   3.184
-------------------------------------------------------------------------------
open edge clock latency                                                    3.007
close edge clock latency                                                   3.184
-------------------------------------------------------------------------------
actual pulse width                                                         0.177
-------------------------------------------------------------------------------
required pulse width (high)                                                0.200
actual pulse width                                                         0.177
-------------------------------------------------------------------------------
slack (VIOLATED)                                                          -0.023
```

You can down the open edge or up the close edge



Here's open edge, find the INV/BUF and size cell. Notice the last word of the row have character (r/f), that means this cell is better for rise/fall edge. You can report the library cell to choose cell to size.

The type cell good for fix minpulse have type name:

L: low

LL: low low

LU: low ultra

H: high

HH: high high

HU: high ultra

Note: please open 2 scenarios LT/HT to make sure you clean both.

5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

Fix tran/cap violation

Tran/Cap violation caused by long distance/big fanout. To fix tran/cap, you can insert buffer to split long distance, split fanout to divide the number fanout. You should check timing through these pins tran/cap vio due to insert/split fanout affect timing. If you split fanout, be careful about the logic, should check FV again.

5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

Fix Vth type cell

Using scripts: /svhome/VCF_RVC_OffSite/mydiemnguyen_p2f2_vf/code_My/wkdir.forMy/check_clock_cell.csh

5. CLOCK TREE SYNTHESIS FLOW

5.3 CTS step: Quality checks & debug

Fix HALO violation

T.B.D

ADDITION

APPLICATION OPTION SETTINGS CTS

```
################################################################################
### Core command
################################################################################
## NSGK2020
## ccd always disable at clock_opt_cts
set_app_options -name clock_opt.flow.enable_ccd -value false

set_app_options -name cts.common.user_instance_name_prefix -value icc2_cts
if {[get_app_option_value -name clock_opt.flow.enable_ccd]} {
        # If CCD is enabled, set both opt and cts user prefix as it can work on both data and clock paths
        set_app_options -name opt.common.user_instance_name_prefix -value icc2_cts_opt
}

redirect -tee -file ${REPORTS_DIR}/${CLOCK_OPT_CTS_BLOCK_NAME}.report_app_options.start {report_app_options -non_default}
```

```
#W.A
set_app_options -name route.common.track_width_constraint_relaxed_mode -value 1

# switch propagate mode
set_app_options -list {time.enable_clock_to_data_analysis false}
set_app_options -list { time.clock_reconvergence_pessimism  same_transition }
set_app_options -list { time.remove_clock_reconvergence_pessimism true}


#cts
set_app_option -name cts.compile.size_pre_existing_cell_to_cts_references -value true

#for routing congestion aware
set_app_option -name cts.compile.enable_global_route -value true

###Add setting to check reason of stacktrace

set_attribute [get_shapes -of_objects [get_nets iso0_memtop/fsy0/flbss0_flbss/EX_VDD3_C]] shape_use detail_route

########################################


#ccs
set_app_options  -list {time.enable_ccs_rcv_cap true}
set_app_options  -list {time.delay_calc_waveform_analysis_mode full_design}

#control CTS engine
set_app_options -name cts.compile.enable_cell_relocation -value none ;#all | leaf_only, all and none.
set_app_options -name cts.compile.power_opt_mode -value none ;# gate_relocation, low_power_targets, all and none.

cts_qor_check ${REPORTS_DIR} ${CLOCK_OPT_CTS_BLOCK_NAME}_before_build
```

APPLICATION OPTION SETTINGS CTS

```
##########################################################################################
## Enable AOCV (recommended after CTS is completed)
##########################################################################################
if {$AOCV_CORNER_TABLE_MAPPING_LIST != "" && ![get_app_option_value -name time.pocvm_enable_analysis]} {
    ## Enable the AOCV analysis
    set_app_options -name time.aocvm_enable_analysis -value true ;# default false

    ## Enable the AOCV distance analysis (optional)
    ## AOCV analysis will consider path distance when calculating AOCVM derate
    #   set_app_options -name time.ocvm_enable_distance_analysis -value true ;# default false

    ## Set the configuration for the AOCV analysis (optional)
    #   set_app_options -name time.aocvm_analysis_mode -value separate_launch_capture_depth ;# default separate_launch_capture_depth
}
```

```
# === propagated clock enable ===
foreach_in_collection sc [all_scenarios] {
 current_scenario $sc
 set_app_options -list { time.remove_clock_reconvergence_pessimism true}
 set_propagated_clock [all_clocks]
}

# switch propagate mode
set_app_options -list {time.enable_clock_to_data_analysis false}
set_app_options -list { time.clock_reconvergence_pessimism  same_transition }
set_app_options -list { time.remove_clock_reconvergence_pessimism true}

#ccs
set_app_options  -list {time.enable_ccs_rcv_cap true}
set_app_options  -list {time.delay_calc_waveform_analysis_mode full_design}
```