High Speed Cores Design convergence on Intel's 20A Process

Kamal Deep Rajput , Parbhat Kangra
Intel

Design SPEC And Sensitivity

Block Complexity

DESIGN SPEC

Block Complexity

- General Specifications
  - **Design : Block A**
  - **High Speed Design**
  - **Process Node**: Intel 20A
  - **Instance Count**: 1.3 million+
  - **Complex Memories**: 200+ memories with transparencies inside
  - **Clock Tree Synthesis (CTS)**: Multipoint CTS with multiple tap points

- FC Flow Runtime:
  - Total APR runtime from import design till route_opt : **~127 hrs (5+ days)**

DESIGN SENSITIVITY

Block Complexity

- Below table shows the sensitivity of the design w.r.t clock cycle time
- ~6.6% delta in the Clock Freq was leading to shift of ~92% TNS reg2reg , ~83% in2reg , and ~98% reg2out bus impacting other partitions

- **Disclaimer : Used normalized no.'s below for reference**
  - P^{int} = 20000
  - T^{int} = 400 ns
  - P^{ext} = 10000
  - T^{ext} = 300 ns
  - Crit_Reg_Out = 150 ns
  - Crit_A2B = 200 ns

corner	#paths (int)	TNS (int)	#paths (ext)	TNS (ext)	Crit_reg2out TNS	Path from Block 'A' impacting Block 'B' (Interface) TNS
max_high , CT	20000 (P_int)	400ns (T_int)	10000 (T_ext)	300ns (T_ext)	150ns (Crit_Reg_Out)	200 ns (Crit_A2B)
max_high , 1.06CT	2k (10%Pint)	36ns (8%Tint)	2200 (22%Pext)	54ns (17%Text)	16.5 ns (11%Crit_Reg_Out)	4ns (2% Crit_A2B)

Clock Design Strategy

Latency, Power and Timing

Clock Design Strategy

Clock tuning experiments to improve latency, timing, and power :

- The following experiments were done to achieve the optimal clock settings needed to improve the design :

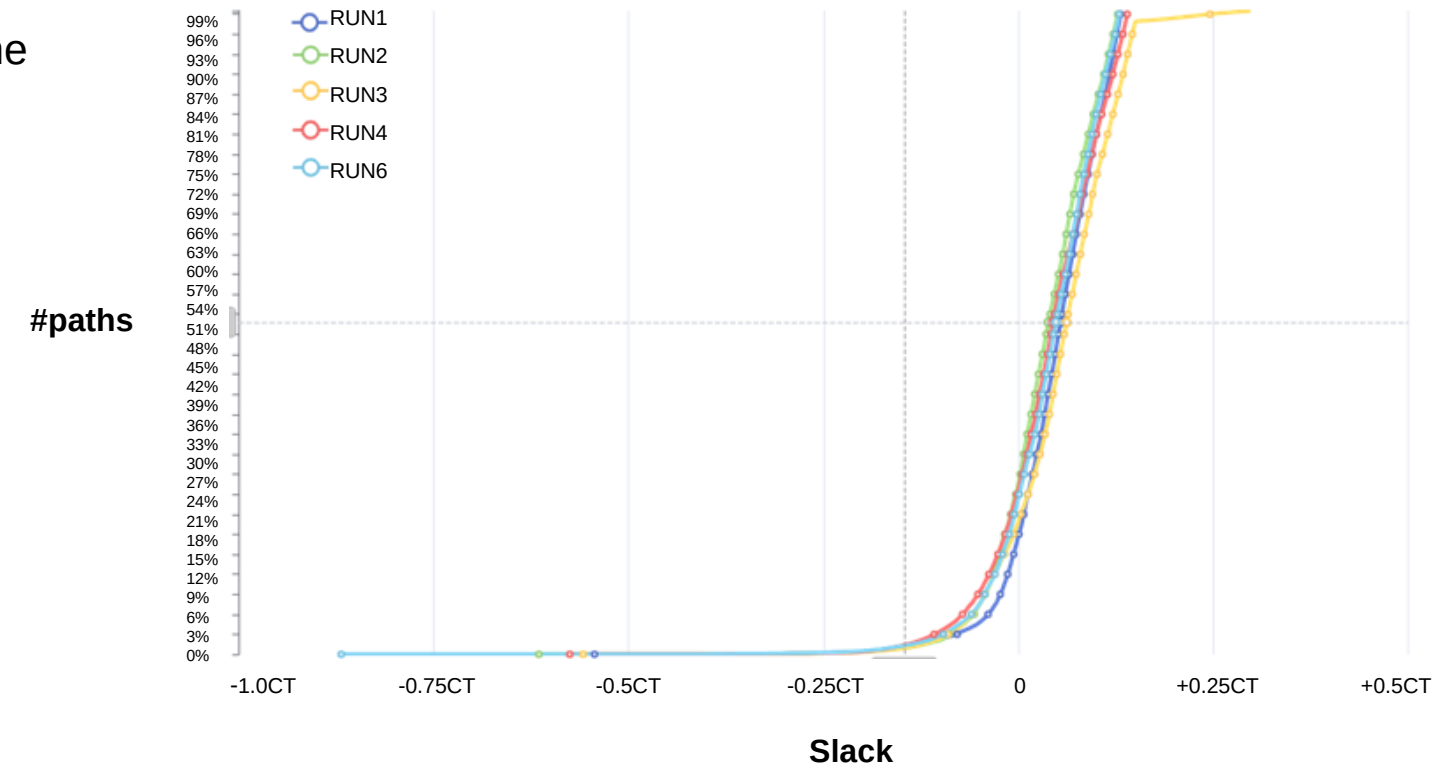
Run Name	Cell type	dg allowed	CTS primary Corner	tran limit	max repeater	cts Max fan out	metal layer allowed	Set Min_cap at clock o/p pin	Gskew (max_high %CT)	Med Lat (max_high %CT)	Max Lat (max_high %CT)	Min Lat (max_high %CT)
RUN1	all buf/inv	all dg used	max_nom	0.75(Y/Z)		no	Upto Top-2 layer		47	57	80	32
RUN2	Selected inv/buf templates	mid-range dg	max_high	Y/Z	4	no	Upto Top-2 layer		41	56	72	31
RUN3	Selected inv/buf templates	mid-range dg	max_high	Y/Z	4	no	Opened Top 2 layers also		51	53	77	25
RUN4	Selected inv/buf templates	mid-range dg	max_high	Y/Z	4	no	Opened Top 2 layers also	yes	50	54	76	27
RUN5	Selected inv/buf templates	Mid-range @ CTS, low+mid @ CRO	max_high	1.1(Y/Z)		42	Opened Top 2 layers also		45	57	77	31
RUN6	Selected inv/buf templates	Mid-range @ CTS, low+mid @ CRO	max_high	Y/Z	4	no	Opened Top 2 layers also		48	56	76	28

Clock Design Strategy

Clock tuning experiments to improve latency, timing, and power :

- **Normalized margin graph:**

- **RUN3** (in yellow) is shifted towards right w.r.t the reference shows better slack values

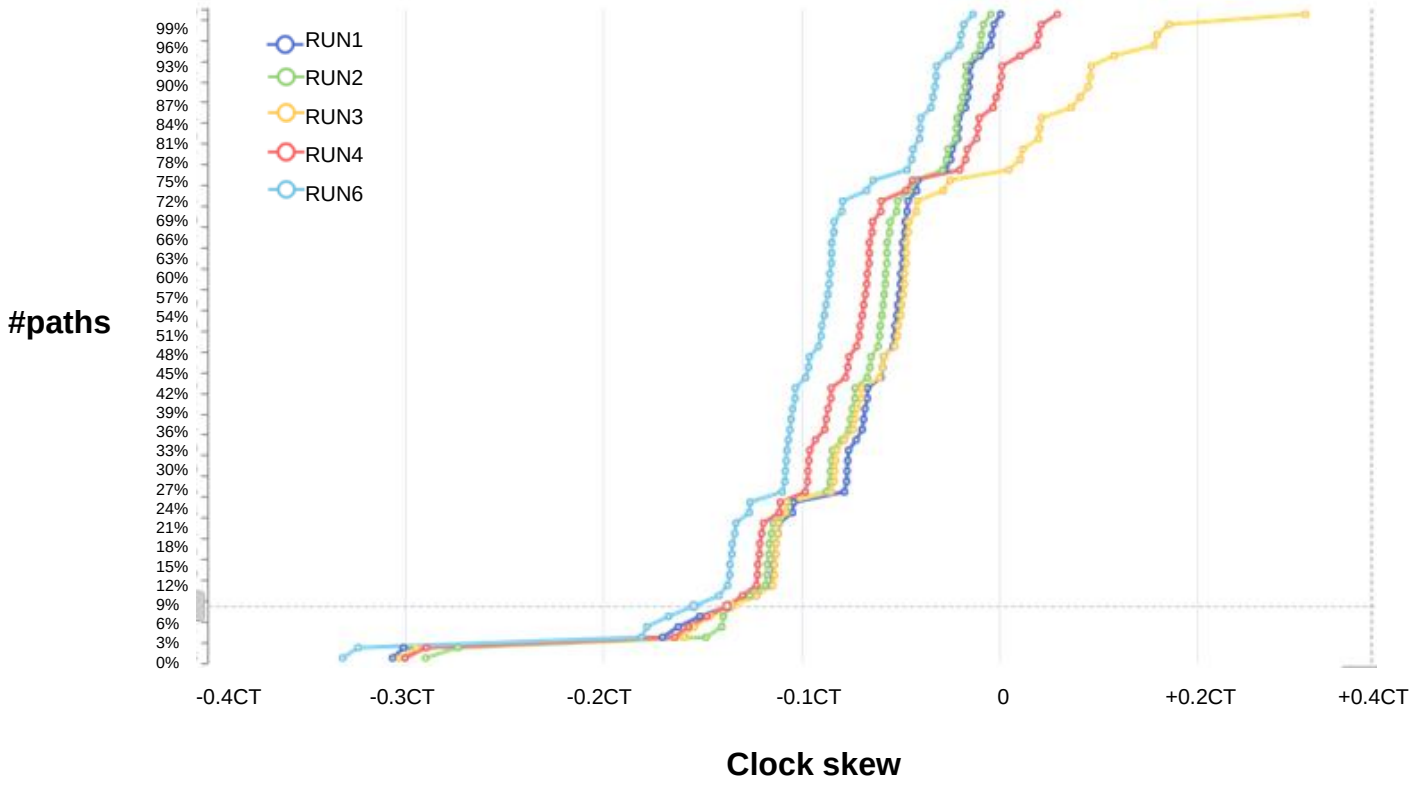


Clock Design Strategy

Clock tuning experiments to improve latency, timing, and power :

- **Clock skew graph:**

- **RUN3** (in yellow) is shifted towards right w.r.t the reference shows more +ve clock skew for more number of paths which helped improving the timing QoR

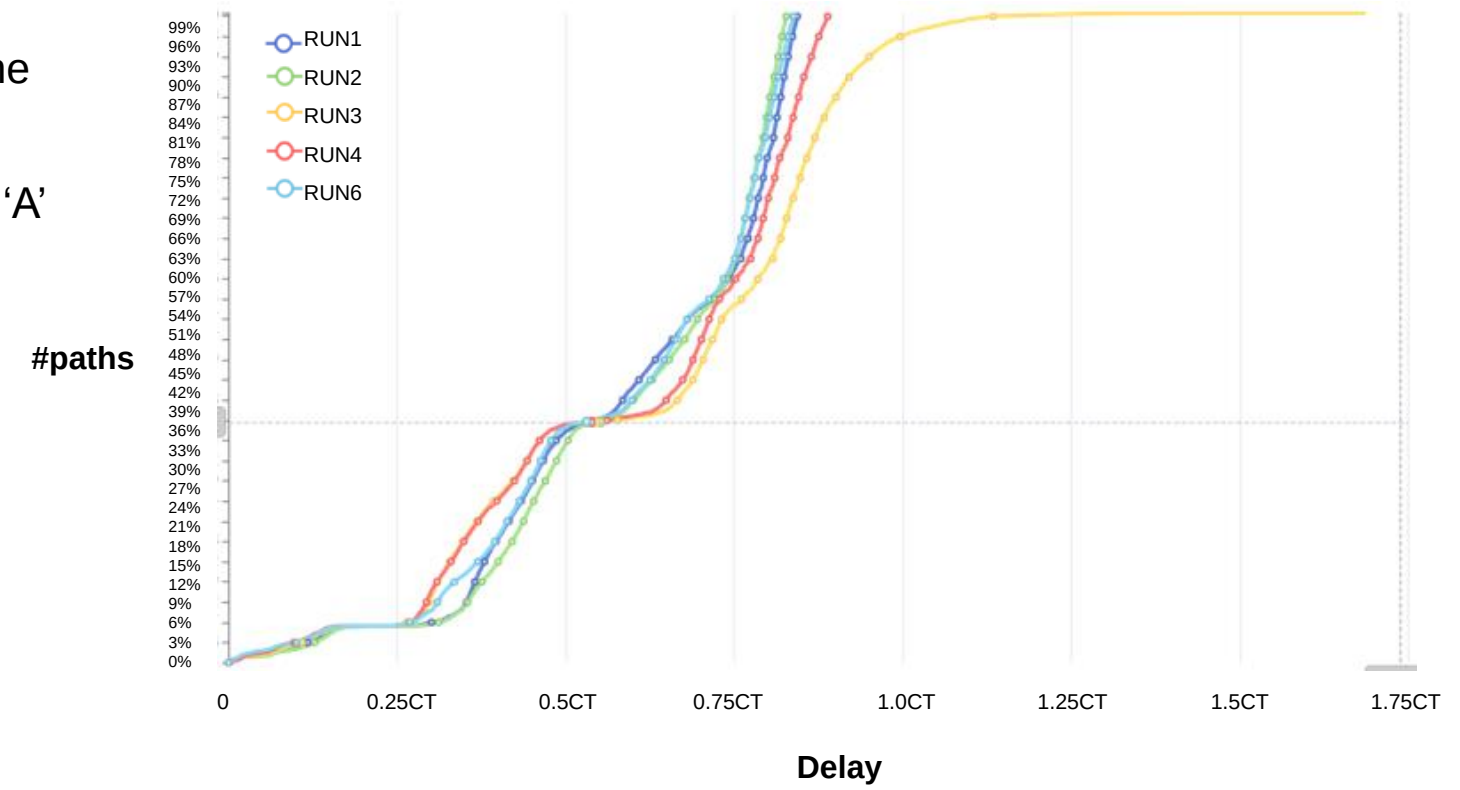


Clock Design Strategy

Clock tuning experiments to improve latency, timing, and power :

- **Total Delay graph:**

- **RUN3** (in yellow) is shifted towards right w.r.t the reference shows delay increased on datapath which helped in power improvement of the block ‘A’ without impacting timing QoR



Clock Design Strategy

Clock tuning experiments to improve latency, timing, and power :

- The following experiments were done to achieve the optimal clock settings needed to improve the design :

	Primetime				PTPX	
Run Name	Total (setup) (max_high)	R2R (Setup) (max_high)	In2reg (setup) (max_high)	R-->R(Hold) (max_nom)	Total pwr	clk pwr
RUN1	1	1	1	1	1	1
RUN2	0.8	0.8	1.2	1.4	0.98	0.95
RUN3	1.0	0.3	2.6	1.4	0.97	0.92
RUN4	1.3	0.7	3.0	1.4	0.96	0.92
RUN5	0.9	0.7	1.4	0.2	0.96	0.87
RUN6	1.0	0.7	1.4	1.5	0.97	0.90

Database	cell count	TOTAL
Ref	No' of clock cell	N^{total}
	No' of clock_inverter	N^{inv}
	No' of clock_buffer	N^{buf}

Database	cell count	TOTAL
Test	No' of clock cell	$0.9N^{total}$
	No' of clock_inverter	$0.8N^{inv}$
	No' of clock_buffer	$0.86N^{buf}$

- RUN3 QoR :
  - Enabling Top 2 layers for clock NDR increased clock net metal usage from ~1% to ~5% for those layers
  - better median latency w.r.t reference
  - It’s a trade-off between median latency improvement and in2reg setup paths which was handled with targeted clock-pushes in ECO phase
  - ~3% Total power gain and 8% clock power gain
  - Clock tree is inverter cell dominated
- While RUN5 recipe had better TNS and power but excess relaxation of maxcap and trans resulted in sign off logical DRC violations

Clock Tool Recipe

Clock Tool Recipe

CTS settings to improve timing and meet target latency:

- **CTS:**
  - Clock fine-tuning done to reach best on latencies based on criticality and validated during CTS to ensure less loop and least manual effort
  - **Challenges in clock tuning :**

    Problem statement :
    - Some of the critical in2reg paths latency was on lower side, while reg2out paths latency was on higher side
    - The median latency is high and there is a significant increase in median latency from the CTS stage to CRO
    - The balance point offset values due to CCD at compile was causing many iterations to settle on the required latency on critical paths

  - **Clock tuning settings used :**
    - To control the jump of latency from cts to clock_route_opt used below setting in CRO,

```
set_app_options -name clock_opt.flow.enable_ccd_clock_drc_fixing -value always_off
set_app_options -name ccd.max_postpone          -value $limit
set_app_options -name ccd.max_prepone           -value $limit
set_app_options -list {opt.common.hold_fixing_setup_margin $value}
```

Clock Tool Recipe

CTS settings to improve timing and meet target latency:

- **CTS:**
  - **Clock tuning settings used :** Steps followed to meet latency requirement for critical family of registers:
    - Below settings are common for all scenarios of clock tuning
      - Disable CCD optimization to critical family of registers from compile stage which resulted in '0' offset values as well

        set_app_options -name ccd.respect_cts_fixed_balance_pins -value true

        set_attribute [get_flat_pins $target_register/clk] -name cts_fixed_balance_pin -value true*

      - **For clock pushing**

        set_clock_balance_points -clock [get_clocks $clk_name] -delay -$push -balance_points $target_clk_pin –scenarios $scenario

      - **For clock pulling** , created skew groups for different families of sequential

        create_clock_skew_group -name $skew_gp_name -objects $target_clock_pin -clock $clk_name

        set_clock_balance_points -clock [get_clocks $clk_name] -delay $pull -balance_points $target_clk_pin –scenarios $scenario

Fine tunning of latencies was done w.r.t the median latency of the design.

Creating skew group resulted in latencies purely based on placement of the registers from tap drivers and not according to any other register out of this particular skew group. It was iterative process to decide on amount of push/pull for a skew group.

Clock Tool Recipe

CTS settings to improve timing and meet target latency:

- **CTS:**
  - **Clock Latency :**
    - FC requirement ,
      a) **Critical in2reg** family : **~65%CT**
      b) **Critical reg2out** family : **~47%CT**
  - **Latencies achieved through construction :**

Median latency
55%CT

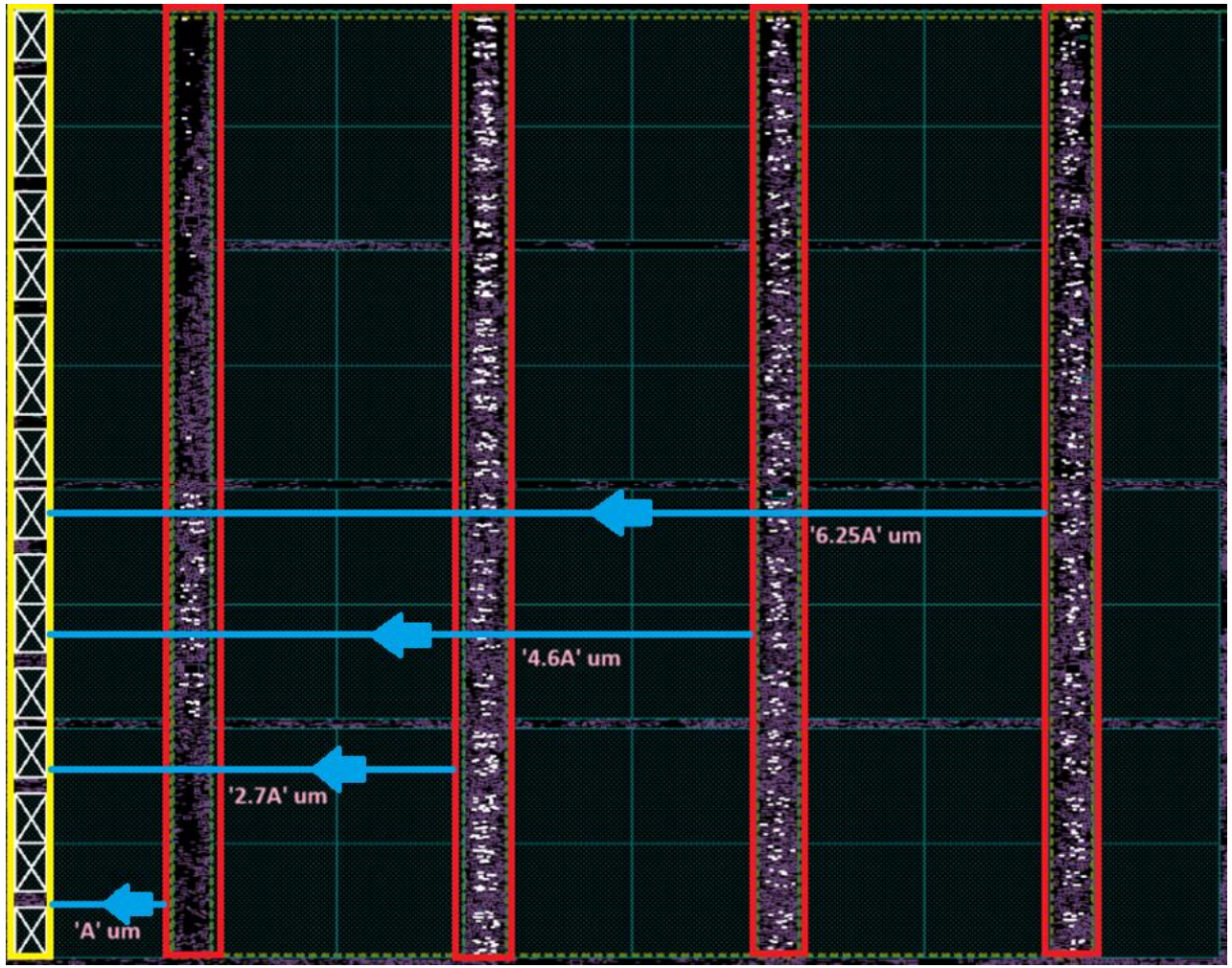
Crit in2reg family	Latency
Group 1	67%CT
Group 2	72%CT
Group 3	69%CT
Group 4	73%CT

Crit reg2out family	Latency
Group 1	49%CT
Group 2	45%CT
Group 3	44%CT
Group 4	47%CT

Placement & Routing Challenges

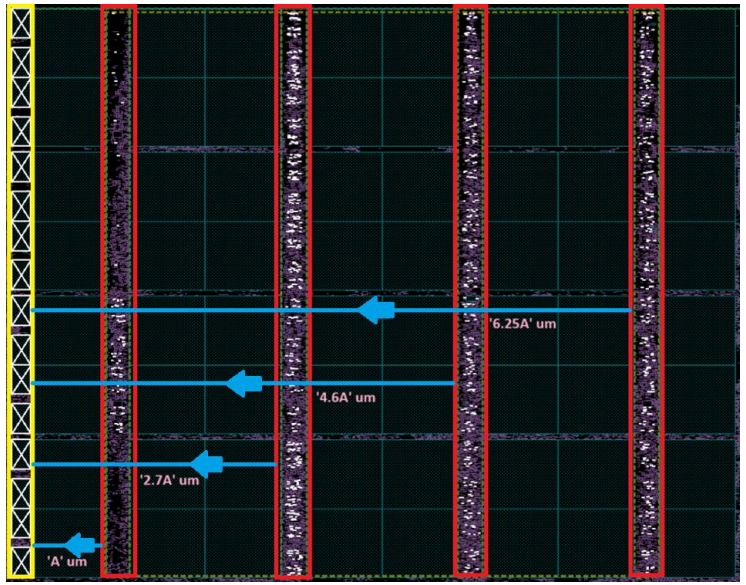
Placement & Routing Challenge

- We have timing paths going through 1k+ latches (in red) in transparency going through memories (in yellow) and ending at interface ports , as shown below,



Placement & Routing Challenge

- **Challenges :**
  - Jogs and lower metal layer usage for longer nets
  - Least level of repeaters (1-2)
  - Critical Seq alignment
    - Placement of the latches were not aligned as per the memory input pins
  - Crosstalk on good/straight routes because of no NDR



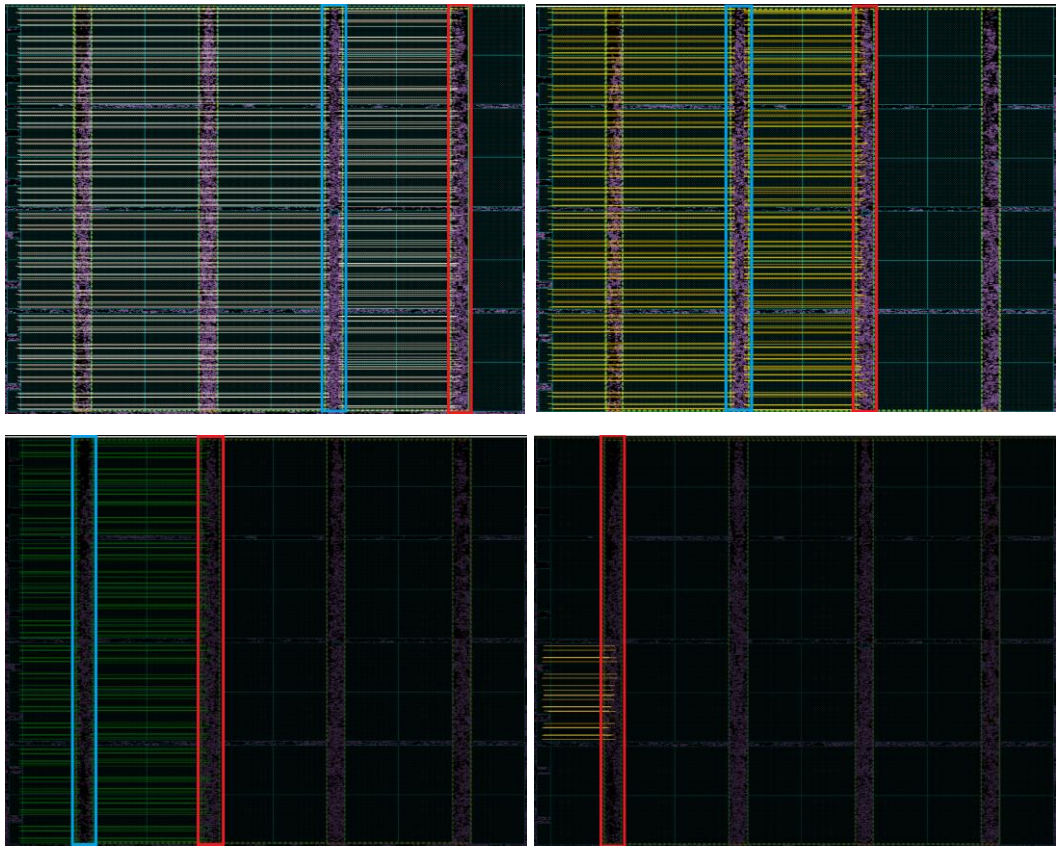
- **Target** : quality right from construction
  - Bounding logic : bounded 1k+ seq in the vertical channels based on location of fanin cone
  - Decision on repeater count
  - Optimized placement of seq and repeaters aligned with memory input pins
  - NDR to support crosstalk , congestion , and timing at same time
    a. Routing guide with 50% partial percentage in that region
    b. Adding routing blockages on alternate tracks

  - Achieved targeted results up to a certain level by above strategy and remaining were done by custom pre-route

Placement & Routing Challenge

- **Custom Pre-route Strategy :**
  - Drawn straight routes considering memory input pins as reference
  - NDR and one repeater on the route till the sequential
  - Different horizontal metal layers were used based on location to reduce congestion in one layer
  - Big size buffer was used to take full advantage of NDR with higher metal layer otherwise splitting it into inverter pair increased via res
  - Decided the optimal placement of the seq (in red) and its repeater (in blue)
  - Did via strapping also on repeater to reduce RC as much as possible

Distance	No. of buffers	Metal layer/NDR
'6.25A' um	1	Top Layer/NDR
'4.6A' um	1	Top Layer - 2/NDR
'2.7A' um	1	Top Layer - 4/NDR
'A' um	0	Top Layer - 6/NDR



- **Achieved RC improvement ~37% over reference DB**

Time Borrow, Pre-buffering, & Slack Based NDR

Time Borrow, Pre-Buffering, & Slack Based NDR

- **Time Borrow :**
  - Time Borrow limit on transparency paths through critical sequential gave **~5%CT (average) gain in datapath delay** by better optimization and reducing the number of levels

- **Slack Based NDR:**
  - **~4%CT gain (average) in RC delay**

FC Slack (ps)	vr length (um)	Metal layer/priority	Pre-buffering
M < worst slack	**L** > Max-limit	Top Layer/1st	Yes
M < worst slack	Max-limit > **L** > Mid-limit	Top Layer – 2/2nd	No
Mid slack > **M** > worst slack	**L** > Max-limit	Top Layer/3rd	Yes
Mid slack > **M** > worst slack	Max-limit > **L** > Mid-limit	Top Layer - 2/4th	No

Initial vs Final Timing QoR

Initial vs Final Timing QoR

- Crit_reg2out TNS improvement ~96%
- Full Chip TNS improvement ~93%
- Trade-off between median latency improvement and in2reg setup paths which was handled with targeted clock-pushes in ECO phase

- **FC timing comparison of Initial vs Final APR DB of block 'A' :** comparison is done with same constraint

Where , $T^{ext} << T^{int}$

DB Name	corner		TNS		TNS
Initial DB	max_high	External	T^{ext}	Internal	T^{int}
Final APR DB	max_high	External	$2.0T^{ext}$	Internal	$0.02T^{int}$
Initial DB	max_nom	External	T^{ext}	Internal	T^{int}
Final APR DB	max_nom	External	$2.0T^{ext}$	Internal	$0.01T^{int}$
Initial DB	min_pfff_high	External	T^{ext}	Internal	T^{int}
Final APR DB	min_pfff_high	External	$0.04T^{ext}$	Internal	$0.6T^{int}$
Initial DB	min_low_cold	External	T^{ext}	Internal	T^{int}
Final APR DB	min_low_cold	External	$0.09T^{ext}$	Internal	$1.6T^{int}$

- **Full Chip TNS comparison:**

Corner (max_high)	ext_tns	int_tns	crit_reg2out_tns	crit_in2reg_tns	FC TNS
Initial DB	T^{ext}	T^{int}	$T^{crit\_reg2out}$	T^{crit_in2reg}	T^{FC}
Final APR DB	$0.09T^{ext}$	$0.04T^{int}$	$0.04T^{crit\_reg2out}$	$1.25T^{crit\_in2reg}$	$0.07T^{FC}$


THANK YOU
Our Technology, Your Innovation™