Achieving sustained performance on next-generation GPUs.

Collaboration with Synopsys.

Abhishek Mukherjee, Senior Engineering Manager
Renuka Eadalada, Senior Applications Engineer
Imagination Technologies

Agenda

Presentation Overview: Delve into the physical design challenges GPUs face and introduce advanced solutions to ensure sustained performance.

- Introduction to Imagination Technologies
- Challenges in achieving sustained performance in GPUs
- Methodology and design overview
- Best recipe for power efficiency
- Summary of Results
- Conclusion
- Appendix

Introduction to Imagination Technologies

What do we do?

Imagination Technologies at a Glance

Global Leader in High-Performance Semiconductor IP Design

Market leader in IP since 1985:

- #1 GPU IP provider in automotive and mobile
- >13Bn cumulative chip shipments with Imagination IP

Key IP solutions for graphics and AI at the edge

- Graphics and GPU compute scaling across markets
- High-quality, performance-dense RISC-V processors

Business model:

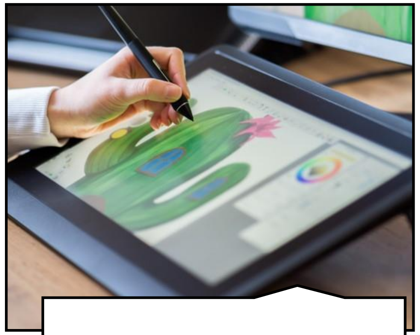
- IP licensing to customers and royalties per chip shipped

Global presence:

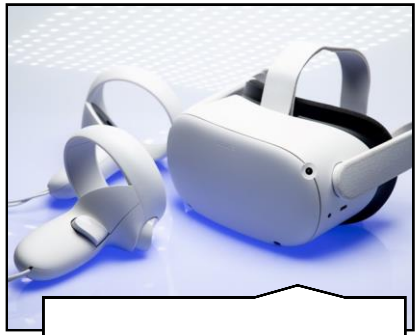
- UK-based with a global R&D team and leading customer relationships in US, Asia and Europe

Automotive

Desktop



Mobile



Consumer

Challenges in achieving sustained performance in GPUs.

What is the motivation?

Challenges in sustained GPU performance

Why does it matter?

- GPUs are ubiquitous (mobile, AI, virtual reality, automotive systems)
- Optimizing for better energy efficiency will help customers and end-users **manage costs** and **reduce operations carbon footprint** worldwide.

Complexity of Power Reduction

Rigorous Trial Process

Inherent Challenges

- Dense Integration
- High Frequency
- Technical obstacles : Leakage, performance vs power

Iterative Testing

- Refine design
- Tweak Parameters
- Test under various operating conditions

Marginal Gains

- Significant efforts, small gains
- Cumulative effect of small gains

Estimating Power savings

Emulation

- Lengthy and costly process
- No alternative for accuracy
- Low TAT, cannot be used for every trial

Benchmarks

- Different customers, different benchmarks
- Benchmarks affect optimizations and power reading.

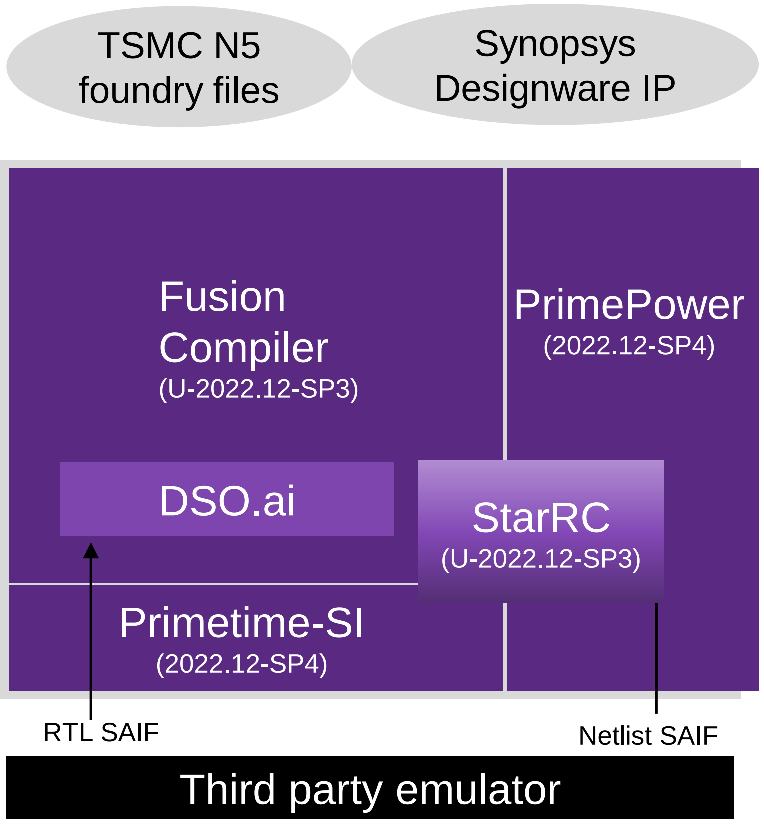
Power Efficient Graphics

The design, Tools, libraries and methodology

Power Efficient Graphics

The tools, process and methodology

- We process our design through the **PnR, timing closure, parasitic extraction, emulation,** and Primepower for accurate power calculation.
- RM QoR strategy for baseline was ‘**extreme_power**’ i.e. already power focussed.



TSMC N5 foundry files

Synopsys Designware IP

Fusion Compiler (U-2022.12-SP3)

PrimePower (2022.12-SP4)

DSO.ai

StarRC (U-2022.12-SP3)

Primetime-SI (2022.12-SP4)

RTL SAIF

Netlist SAIF

Third party emulator

Process Node	**N5**			
Libraries	Synopsys HPC Designware Libraries			
Metal Stack	15 (Top 2+ Layers for Power)			
PVT Corners	5 Scenarios			
	Process	SSGNP	TT	FFGNP
	Voltage	0.675	0.75	0.825
	Temperature	125c/m40c	85c	125c/m40c
	Parasitic corners	Cworst CCworst_T	typical	Cbest CCbest_T
OCV Strategy	POCV			
Power scenario	TT Nominal voltage (0.75V) by 15%			
Target Frequency	1 GHz			
# Instances	60 million			

Best recipe for power efficiency

Strategies to reduce dynamic power

Strategy to Improve Power Efficiency

1. Meticulous floorplan refinements to optimize power consumption.
2. Cell selection: Select the most appropriate standard cells within libraries.
   - Curating the don’t_use list using Power delay product (PDP).
   - Implementing double inverter flops for enhanced efficiency.
3. Selecting module activity-based voltage threshold (VT) and bounds.
4. Optimizing tool settings
   - Auto density control
   - Useful skew moderation
   - Register Retiming
   - Performance via ladders
   - Sequential fanin
   - Including datapath options
5. Conducting Regressions to determine optimal design inputs

   (Skew Limit, Clock Transition limits etc…)
6. DSO.ai and MLMP.

Floorplan refinements

Collaboration with designers to refine module placement
Floorplan shrink to reduce C(effective)

Floorplan refinement

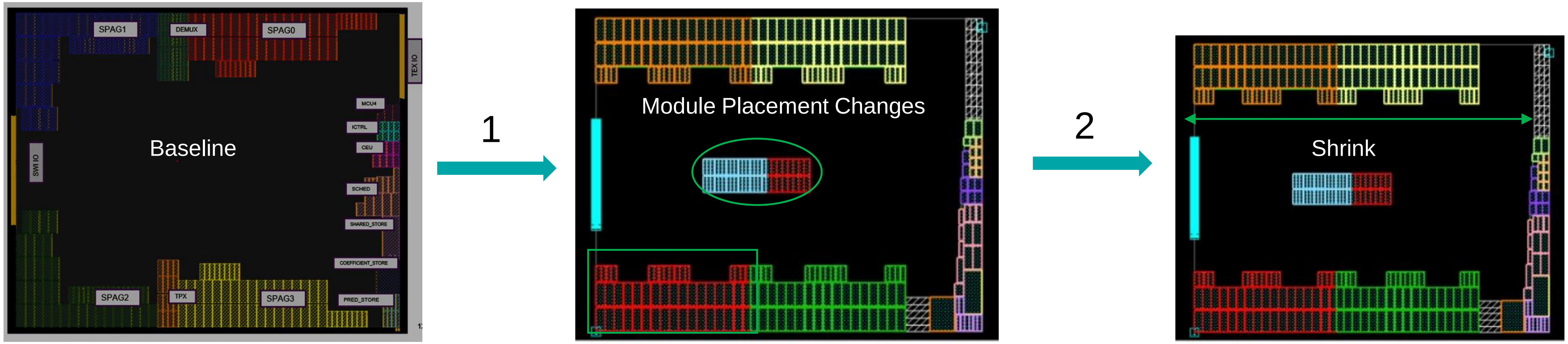
Block Level

1. **Module placement**: Reorder up to 4 levels of hierarchy to ensure module placement results in less wire length.
2. **Shrink** the floorplan but avoid congestion. (Final standard cell utilisation around 62%)

Objectives :

- Reduce the wire capacitance without causing routing issues.
- Study the TR of modules to ensure IR compliance.

Observations: This leads to a **2%** reduction in dynamic power.



Cell Selection

Using Power Delay Product (PDP)

Power efficient cells

- Compare PDP for “All Architecture Variants” for the function under typical design conditions (trans/load/toggle rate) and most used drive strength (D1) and VT (LVT).
- Set don’t use on variants for which the PDP is **higher than the minimum by a threshold**.
- Ex: For ICG, only V8 cells were selected, which aligns with the guidelines in Synopsys document
- **20%** more cells added to the new don’t use cell list
- PDP flow also selects the most power-efficient version for AOI/OAI and other combinational cells.

Example : ICG Cell
Function : CKGTPLT
variants : (V8, V7, V5)
Threshold : 5
Best variant : V8

CELL NAME	VARIANT	Power	Delay	PDP	% delta from min
HDBLVT06_CKGTPLT_CAQV8_1	CAQV8	3.32E-08	0.029	9.77E-10	**0**
HDBLVT06_CKGTPLT_CBQV7_1	CBQV7	3.63E-08	0.028	1.03E-09	6
HDBLVT06_CKGTPLT_CB3QV7_1	CB3QV7	3.63E-08	0.028	1.03E-09	6
HDBLVT06_CKGTPLT_CAQV7_1	CAQV7	3.63E-08	0.029	1.04E-09	7
HDBLVT06_CKGTPLT_CA3QV7_1	CA3QV7	3.63E-08	0.029	1.04E-09	7
HDBLVT06_CKGTPLT_CAQV7FC_1	CAQV7FC	3.61E-08	0.033	1.19E-09	22
HDBLVT06_CKGTPLT_CAQV5_1	CAQV5	3.86E-08	0.037	1.45E-09	48

	Delta
Stdcell Area	-1.0%
Total Dyn Power	-1.7%
Leakage Power	-2.60%

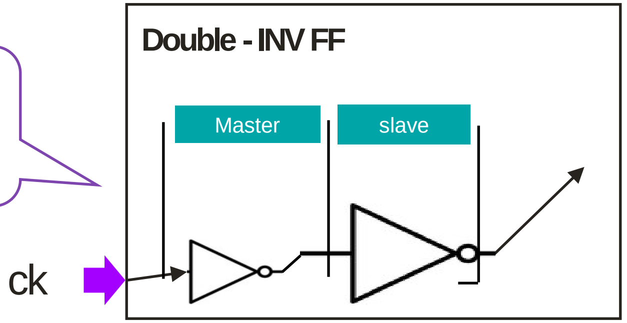
Cell Selection

Double Inverter Flops

Double inverter FF

- Some cells use double inverters on the clock, which isolates the register from the clock tree.
- Although slightly higher in the area, these cells help to reduce the dynamic power.
- Total power improved by **10.7%** with double inverter FF in CTS (**9%** post-route)
- WL , clock area reduced (less $C_{effective}$); Latency improved (better OCV)
- All other metrics (Logic Area / Setup TNS) have improved
- Hold TNS was easily recovered with a minor penalty to the area.

CK pin Load : 1/10th of normal FF
Avg o/p Tran : same as default FF



Clock QOR at cts stage					
AvgLncy	Area	Cell Count	Buf Area	Buf Count	wirelength
-13.10%	-38.18%	-55.02%	-266.57%	-279.73%	-2.52%

	Delta
Stdcell Area	-0.80%
Total Power	-10.70%
Leakage Power	-2.60%

Module Activity based VT Selection

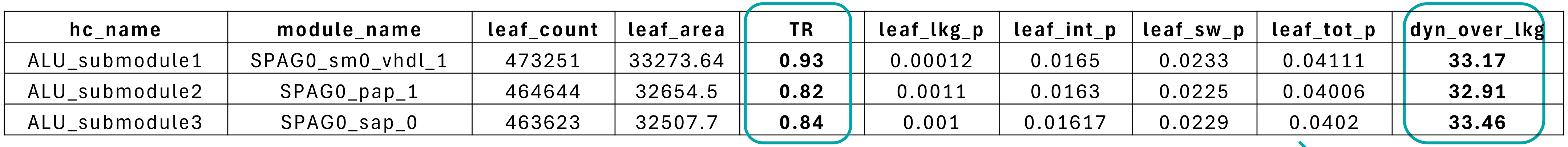
Module Activity based VT Selection

- Some ALU modules have a high average toggle rate (~ 1).
- Using ULVT/ULVTLL will reduce cell size (pin cap), making overall power better

1. Get list of hierarchy which meet criterion

```
source scripts/FC/process_hierarchical_power_info.tcl
set hier_tr_list [hier_power_info report_power.hier.rpt.new 30]
```

(The criterion is dyn2lkg ratio > 30)



hc_name	module_name	leaf_count	leaf_area	TR	leaf_lkg_p	leaf_int_p	leaf_sw_p	leaf_tot_p	dyn_over_lkg
ALU_submodule1	SPAG0_sm0_vhdl_1	473251	33273.64	**0.93**	0.00012	0.0165	0.0233	0.04111	**33.17**
ALU_submodule2	SPAG0_pap_1	464644	32654.5	**0.82**	0.0011	0.0163	0.0225	0.04006	**32.91**
ALU_submodule3	SPAG0_sap_0	463623	32507.7	**0.84**	0.001	0.01617	0.0229	0.0402	**33.46**

Hierarchical Average Toggle & DYN2LKG ratio info from Baseline runs.

2. Preserve the selected hierarchies (auto ungroup is ON by default)

```
set_ungroup [get_cells $hier_tr_list] false
```

3. Set Target lib subset

```
foreach cell $hier_tr_list {
if {[sizeof_coll [get_cell -q $cell] ] != 0} {
set_target_library_subset -objects $cell -only_here [get_lib_cells */HDBULTLL06*] }}
```

Results

1. With targeted ULVTLL **total area** was reduced by **2%**, and the area for the targeted modules shrunk by **3 to 4%**
2. Total ULVTLL usage is 7.14%, Targeted module ULVTLL usage is 16%
3. If dynamic to leakage ratio > 30, expected power saving > **2%**

	Logic Area Delta	ULVTLL %age
Block Level (ALU)	-2%	7%
Targeted Module (ALU_submodule1)	-3.20%	16%

Power Saving Estimate (for targeted modules)			
	Baseline	**Module Based VT**	**Delta**
Leakage	L	L*0.16*2.8 + L*(1-0.16) = 1.288L	
Dynamic	D = 30L	30L*(1-0.032) = 29.04L	
Total	31L	30.328L	-2%

*__Note__: We also experimented with Module activity-based bounds. While it leads to wirelength reduction, it results in routing issues later.
The concept for module selection is same.

Optimizing tool settings

Register Retiming
Include datapath options
Sequential fanin
Auto density control
Via Pillars

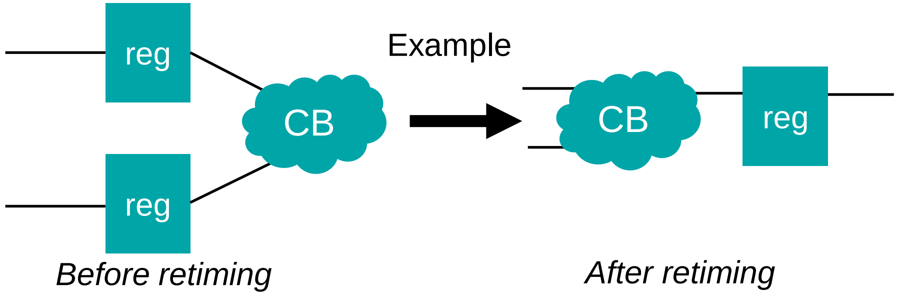
Retiming control

Current setting: **set_optimize_registers -modules $des true -delay_threshold 1.0**

Experiment setting: **set_optimize_registers -modules $des true -delay_threshold 1.2**

Observations:-

1. Relaxing delay threshold by 20%, doesn't degrade timing, area is smaller by **0.5%**
2. Tool inserted slightly higher number of retiming bits, but better banking among those cells
3. Recommendation: **Relax delay threshold to 1.2**



	Delta
Tot. Logic Area	-0.50%
Retiming FF Area	1.80%
Total Power	-1.00%

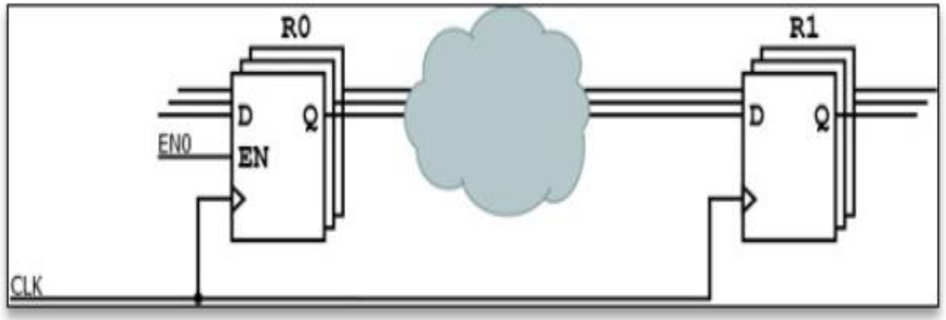
	1.0 (Baseline)	1.2 (relaxed)	Delta
Bit Per FF	6.05	6.2	2.50%
Retiming Reg count	14,180	13,637	-3.80%
Retiming Bit Count	78,460	79,860	1.80%
Setup TNS	-1.3	-1.3	0.00%

Fanin-based sequential clock gating

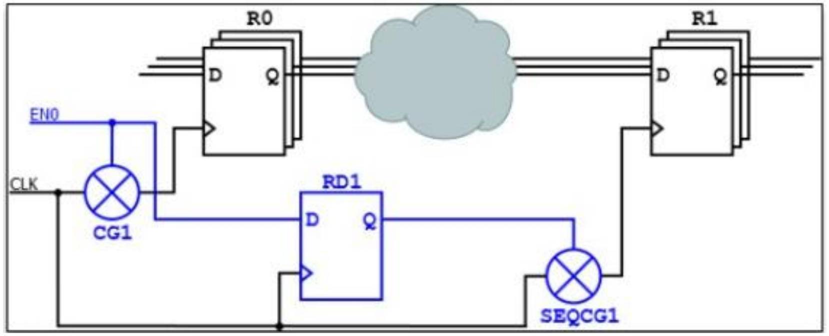
- Edit rm_user_plugin_scripts/compile_pre_script.tcl

```
set_app_options –list {compile.clockgate.fanin_sequential true}
```

- **Observation:** The sequential clock gating saves **0.5%** dynamic power.



Before clock gating



After clock gating

Stage	Setup		Hold		Netlist				
Route opt	r2rWNS	r2rTNS	HWNS	HTNS	StdCells	Hbufs	Util	Gated%	Bits/Flop
Baseline	-0.11	-2.07	-0.728	-2.9	3561388	174511	56	99.3	6.05
final	-0.133	-3.36582	-0.72072	-3.1	3671791	136642	56.73	99.3	6.01
Difference	20.90%	62.60%	-1.00%	6.90%	3.10%	-21.70%	1.30%	0.00%	-0.60%

	Delta
Stdcell Area	1.20%
Total Power	-0.50%
Leakage Power	0.90%

Include datapath options

Edit rm_user_plugin_scripts/compile_pre_script.tcl

set_datapath_gating_options –enable true –sequential true

set_datapath_architecture_options –power_effort medium

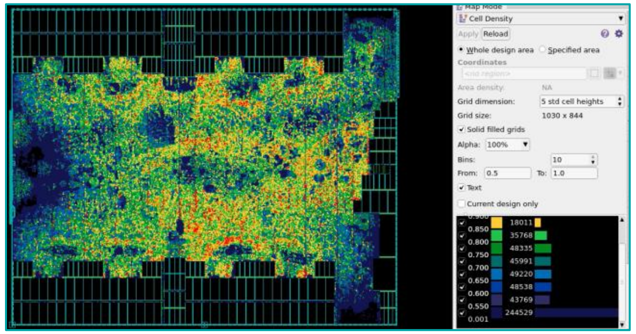
Observation: This option saves an additional **1%** dynamic power.

	Delta
Stdcell Area	-0.20%
Total Power	-1.20%
Leakage Power	0.50%

Stage	Setup		Hold		Netlist				
	r2rWNS	r2rTNS	HWNS	HTNS	StdCells	Hbufs	Util	Gated%	Bits/Flop
Baseline									
route_opt	-0.11	-2.07	-0.728	-2.9	3561388	174511	56	99.3	6.05
Final									
route_opt	-0.04	-0.74	-0.71	-2.7	3536458	201037	55.89	99.3	6.11
Difference	-64.60%	-64.10%	-1.90%	-6.90%	-0.70%	15.20%	-0.20%	0.00%	1.00%

Auto density control

Recipe: Let the tool clump (high TR) cells (don’t spread evenly)



Cell Density

● Whole design area ○ Specified area

Area density: NA

Grid dimension: 5 std cell heights

Grid size: 1030 x 844

☑ Solid filled grids

Alpha: 100%

Bins: 10

From: 0.5 To: 1.0

☑ Text

□ Current design only

	Range	Count
☑	0.900 – 0.850	18011
☑	0.850 – 0.800	35768
☑	0.800 – 0.750	48335
☑	0.750 – 0.700	45991
☑	0.700 – 0.650	49220
☑	0.650 – 0.600	48538
☑	0.600 – 0.550	43769
☑	0.550 – 0.001	244529

- **place.coarse.auto_density_control = true,**
- **place.coarse.max_density = 0**
- Visually, placement with enhanced option seems clumped.
- **7%** reduction in wire length and **1%** in cell area leads to a **1.13%** reduction in dynamic power

	Logic Area Delta	Wire Length Delta	Setup TNS	Congestion
Auto density (false)	-	-	-1.3ns	0.036/0.066
Auto density (True)	**-1%**	**-7%**	**-2.0ns**	0.027/0.071

Performance via ladders

- Setup performance via ladders for the high-density cells
  - 2 files are created which need to be edited in sidefile_setup.tcl
    - **set TCL_VIA_LADDER_DEFINITION_FILE "auto_perf_via_ladder_rule.tcl"**
    - **set TCL_SET_VIA_LADDER_CANDIDATE_FILE "auto_perf_via_ladder_association.tcl"**

    Edit design_setup.tcl
    - **set ENABLE_PERFORMANCE_VIA_LADDER true**
- **Observation:** **3.1%** power savings at the end of route_opt.

	Delta
Stdcell Area	-0.60%
Total Power	-3.10%
Leakage Power	-0.70%

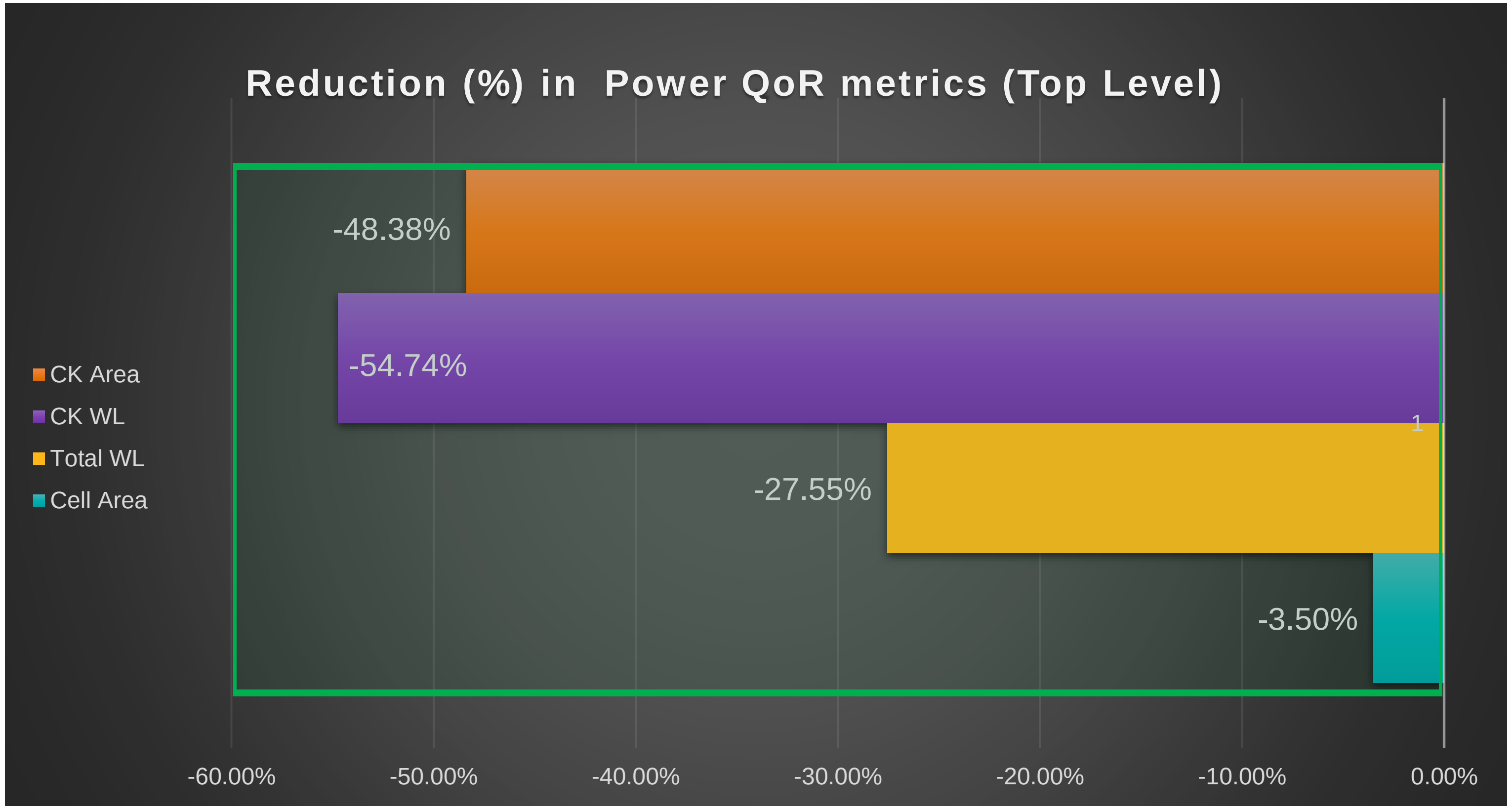
Stage	**Setup**		**Hold**		**Netlist**				
	r2rWNS	**r2rTNS**	**HWNS**	**HTNS**	**StdCells**	**Hbufs**	**Util**	**Gated%**	**Bits/Flop**
Baseline									
route_opt	-0.11	-2.07	-0.728	-2.9	3561388	174511	56	99.3	6.05
Final									
route_opt	-0.07	-2.38	-0.4	-2	3532897	163691	55.61	99.3	6.06
Delta	-38.20%	15.20%	-45.70%	-31%	-0.80%	-6.20%	-0.70%	0.00%	0.20%

Summary of results

Summary

Summary of the key points

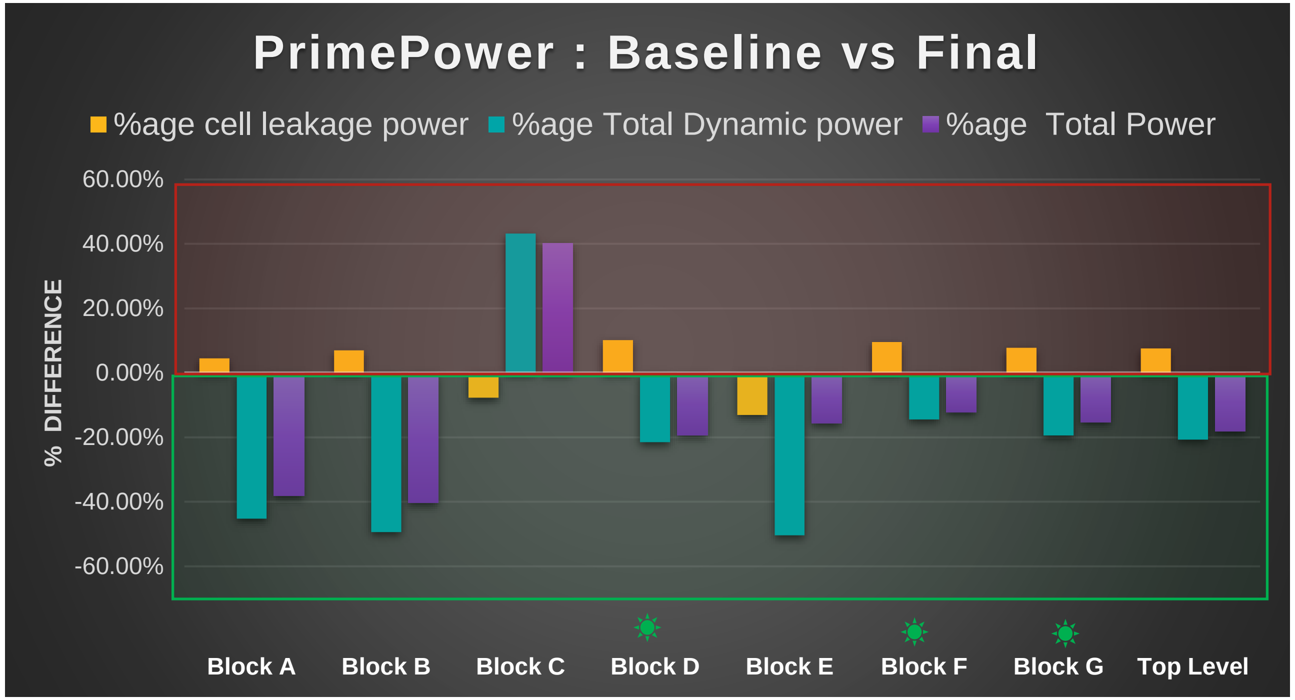
Trial	% Power improvement
Floorplan modifications	2.00%
PDP cell selection	2.50%
Double inverter FF	9.00%
TR based module VT placement	1.00%
Tool options	5.00%
Total Estimation (FC)	19.50%
Actual power savings (PrimePower)	18.0%
Correlation from estimate to actual power saving	93%

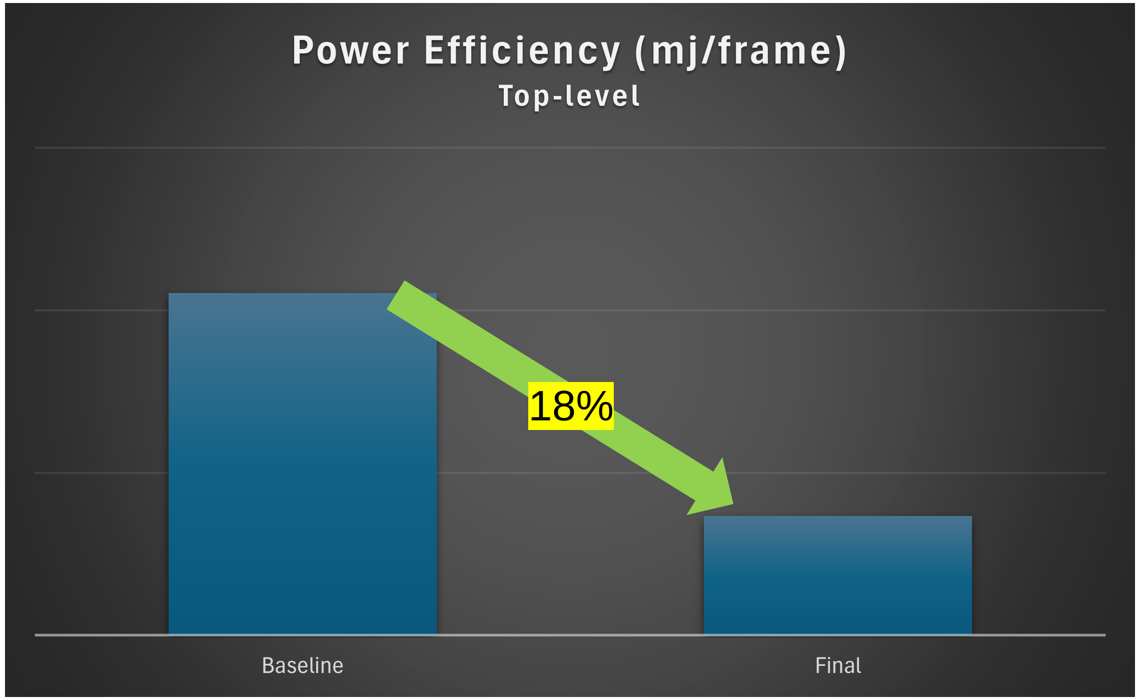


- Reduction in all **QoR parameters correlates** with the reduction in dynamic power.
- **Actual power reduction to estimation is <10%** - acceptable for time/resources saved.

Results (Power and Efficiency)

- **12-20%** power savings across critical blocks.
- Cell area reduction in all blocks except one. Overall, **4%** savings in area.
- No change in the performance of GPU.
- The power savings lead to a direct correlation = **18%** improvement in power efficiency.





☀ Represents critical blocks
*Timing after PT ECOs for both baseline and final were met.

Benchmark = Manhattan 3.1
Same performance (fps) for both runs.

Conclusion and the way forward

The work in progress

Conclusion

Retrospective and Way forward

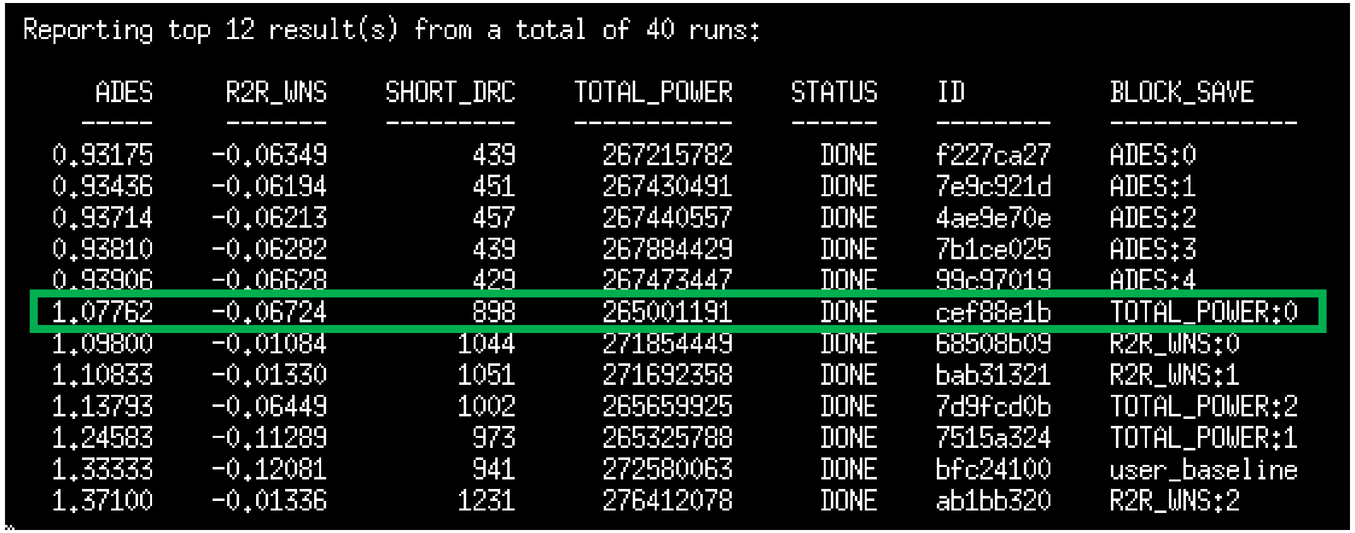
- Analysis of design for floorplan, CTS and module constraints is a crucial step.
- The library choice and cell selection can significantly impact power data.
- It is important to tune the tool settings for the design in use.
  - For ex: Retiming control, datapath options etc…
- Analysis and estimation of power is challenging.
  - Can be mitigated by comparing power QoR proxy metrics.
- Overall, an **18% improvement in power** efficiency with **4% less area** and **no impact on timing** is a massive gain.

- Our next step is to push the design with updated cell lists into DSO.ai.
  - This work is in progress.
  - Initial trials: to let the tool have full autonomy (i.e. baseline into DSO) – to compare against our manual work results.
  - Our results here are with cold start approach. We expect better results with warm start.
- As our blocks are macro-dominant, trials with MLMP will be beneficial – in progress.

DSO.ai

Initial trials

- Baseline upgraded with latest cell selection – as design input to DSO.ai
- 30 runs aimed at improving the total power of blocks
- Resources: 30 – 16 core machines. Runtime = 4 weeks.
- Comparison points: Register WNS, Shorts and Power
- **Cef88e1b** : Best results



```
Reporting top 12 result(s) from a total of 40 runs:

     ADES     R2R_WNS   SHORT_DRC   TOTAL_POWER   STATUS   ID         BLOCK_SAVE
    -----     -------   ---------   -----------   ------   --------   -------------
  0.93175    -0.06349         439     267215782     DONE   f227ca27   ADES:0
  0.93436    -0.06194         451     267430491     DONE   7e9c921d   ADES:1
  0.93714    -0.06213         457     267440557     DONE   4ae9e70e   ADES:2
  0.93810    -0.06282         439     267884429     DONE   7b1ce025   ADES:3
  0.93906    -0.06628         429     267473447     DONE   99c97019   ADES:4
  1.07762    -0.06724         898     265001191     DONE   cef88e1b   TOTAL_POWER:0
  1.09800    -0.01084        1044     271854449     DONE   68508b09   R2R_WNS:0
  1.10833    -0.01330        1051     271692358     DONE   bab31321   R2R_WNS:1
  1.13793    -0.06449        1002     265659925     DONE   7d9fcd0b   TOTAL_POWER:2
  1.24583    -0.11289         973     265325788     DONE   7515a324   TOTAL_POWER:1
  1.33333    -0.12081         941     272580063     DONE   bfc24100   user_baseline
  1.37100    -0.01336        1231     276412078     DONE   ab1bb320   R2R_WNS:2
```

DSO.ai

QoR Comparison for cef88e1b

- The best result shows a **15% improvement in dynamic power**.
  (Was 12% for similar manual trials) i.e. 3% extra power savings.
- Significant improvement in power, **no impact on area**.
- Considering this is an initial trial, we are confident that some tweaking of permutons will lead to more power savings.

Stage	Setup		Hold		Netlist		
	r2rWNS	r2rTNS	HWNS	HTNS	StdCells	Hbufs	Util
Baseline							
route_opt	-0.11	-2.07	-0.728	-2.9	3561388	174511	56
Final (Cef88e1b)							
route_opt	-0.048	-2.39	-0.334	-1.71	3696720	151999	56
Delta	-43.60%	15.90%	-46.00%	-41%	3.80%	-12.90%	0.00%

	Delta
Stdcell Area	0.10%
Total Power	-14.80%
Leakage Power	0.40%

THANK YOU

Our
Technology,
**Your
Innovation™**

Appendix

Additional information

Imagination Technologies (Links)

Company Website	https://www.imaginationtech.com/
About us	https://www.imaginationtech.com/about/
Products	https://www.imaginationtech.com/products/

Avoid useful skew moderation

Recipe: CTS CCD Settings

- **Current setting:** Tool decides max pre/post pone CCD values (set_app_options -name ccd.max_postpone -value auto)
- **Experiment setting:** (1) disable CCD (2) 10% of clock period as max pre/pone value

Why:

- FC seems to be inserting too many repeaters to fix timing using useful skew; clock power will be high
- By limiting useful skew, we can find a balance in clock area/power and timing.
- In our previous projects using Synopsys ICC2, controlling the clock skew showed benefits.

Avoid useful skew moderation

Recipe: CTS CCD Settings

1. By limiting CCD to 10% of clock period, the logic area increased by 0.7% and the clock area reduced only by 1.2%
2. Overall timing is also better with auto
3. **Recommendation: keep auto setting**

KPI (block level)	(auto)	(limit 0.1)	Delta
CK total area	2574	2544	-1.2%
CK repeater area	656	616	-6.1%
Logic Area (um2)			0.7%
CK wirelength	689K	587K	-14.8%
Level	40	19	-52.5%
Global Skew	0.529	0.254	-52.0%
Setup TNS	-8	-15	87.5%
Hold TNS	-9	-4	-55.6%

Clock transition trials

To extract any potential clock power savings

- Since the clock is responsible for almost 30% of GPU power, any fractional improvement will help reduce dynamic power.
- Some refinements and regressions helped us to extract another 0.5% power improvement.

1. Total power reduced by 0.4% for trial 1 setting and 0.5% for trial 2 setting
2. This data is from the clock_opt stage. The trend persists after route_opt.
3. Trial 2 setting is the best choice for all blocks.

	Default	Trial 1	%age	Trial 2	%age
Clock Total Area	3432	4587	33.7%	4073	18.7%
Clock Repeater Area	980	1999	104.0%	1561	59.3%
Clock Wirelength	851894	996791	17.0%	976960	14.7%
Logic Area			-2.3%		-1.8%