SNUG Singapore 2024

DSO.ai : A Paradigm Shift in digital design implementation

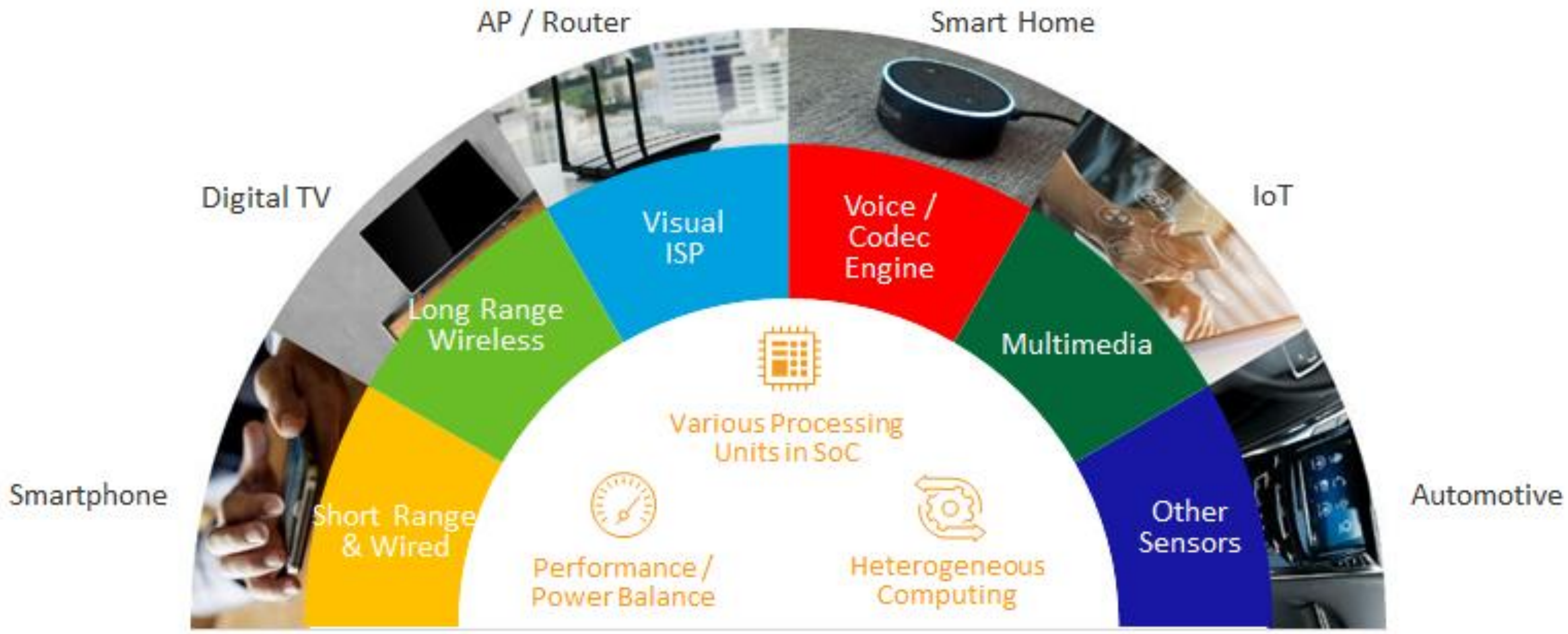
TrongNam Nguyen / Hoang ThienTran
MediaTek Singapore Pte Ltd

Agenda

- ➢ About MediaTek
- ➢ Problem Statement
- ➢ DSO.ai Introduction
- ➢ DSO.ai Setup
- ➢ Results
- ➢ Conclusion

MediaTek is the World’s fifth largest fabless IC design company by revenue. We enables nearly 2 billion connected devices a year.

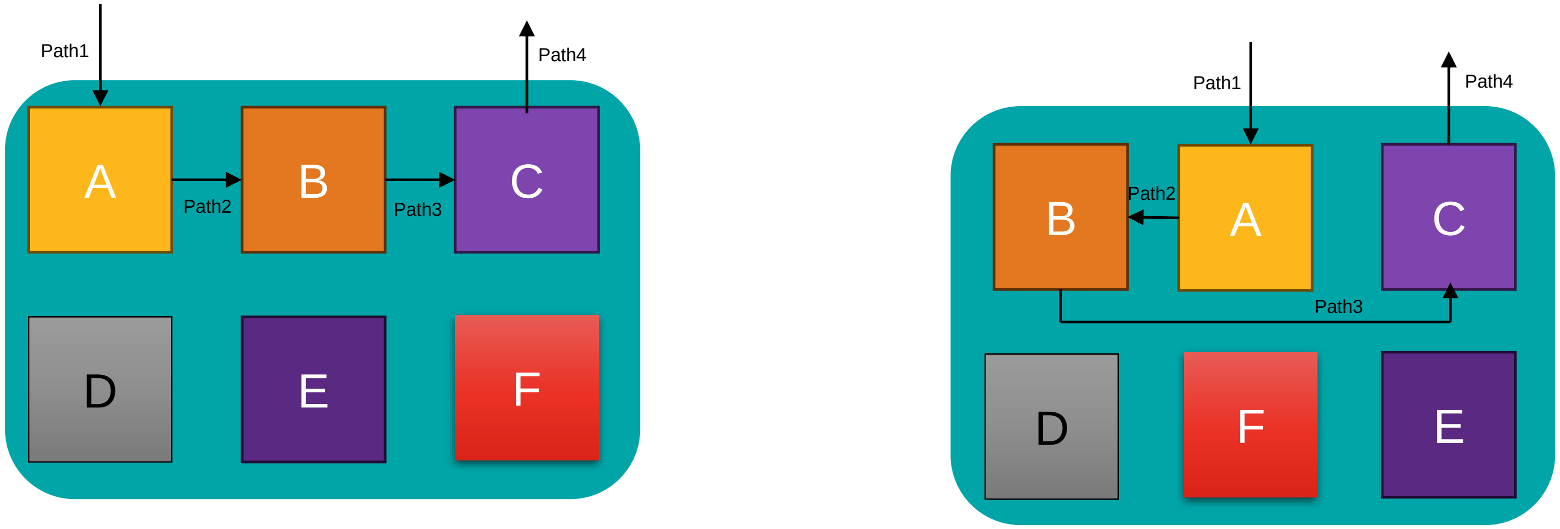
MediaTek is a market leader in developing innovative systems-on-chip (SoC) for mobile device, home entertainment, connectivity and IoT products. Our mission is to be a change catalyst, empowering our partners with smart technology solutions that will inspire them to connect with “next billion” people.



Problem Statement

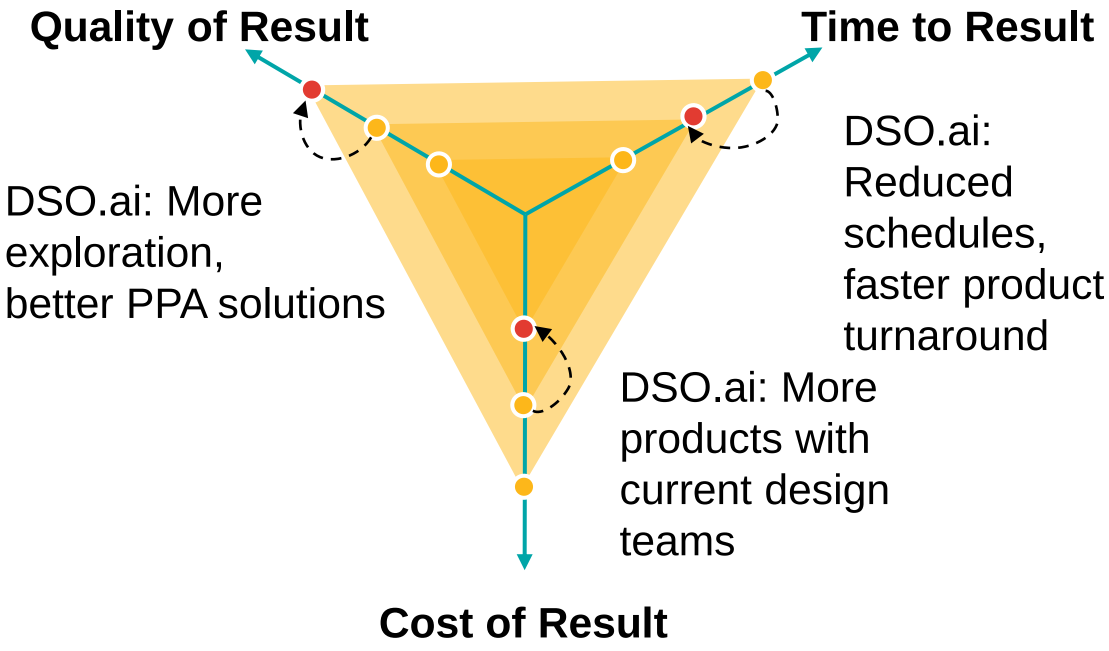
- Power saving is very important as this block is multiple instantiated at top level.
- Exploring power reduction opportunities without resynthesis
- Timing QoR and Total power is very sensitive to module placement
  - 8 set of input and output bus (interface timing) that interact with 6 module in the blocks
  - The first stage of the register is talking to multiple module right before it going out to the output
  - Tight input/output delay constraint

In both scenario below , timing path is clean with different kind of module placement, but total power varies



Why DSO.ai is chosen

- Meet Targets Faster, with Less Effort
- Less Human resources
- Scaling through Systematic Learning and Reuse



Quality of Result

Time to Result

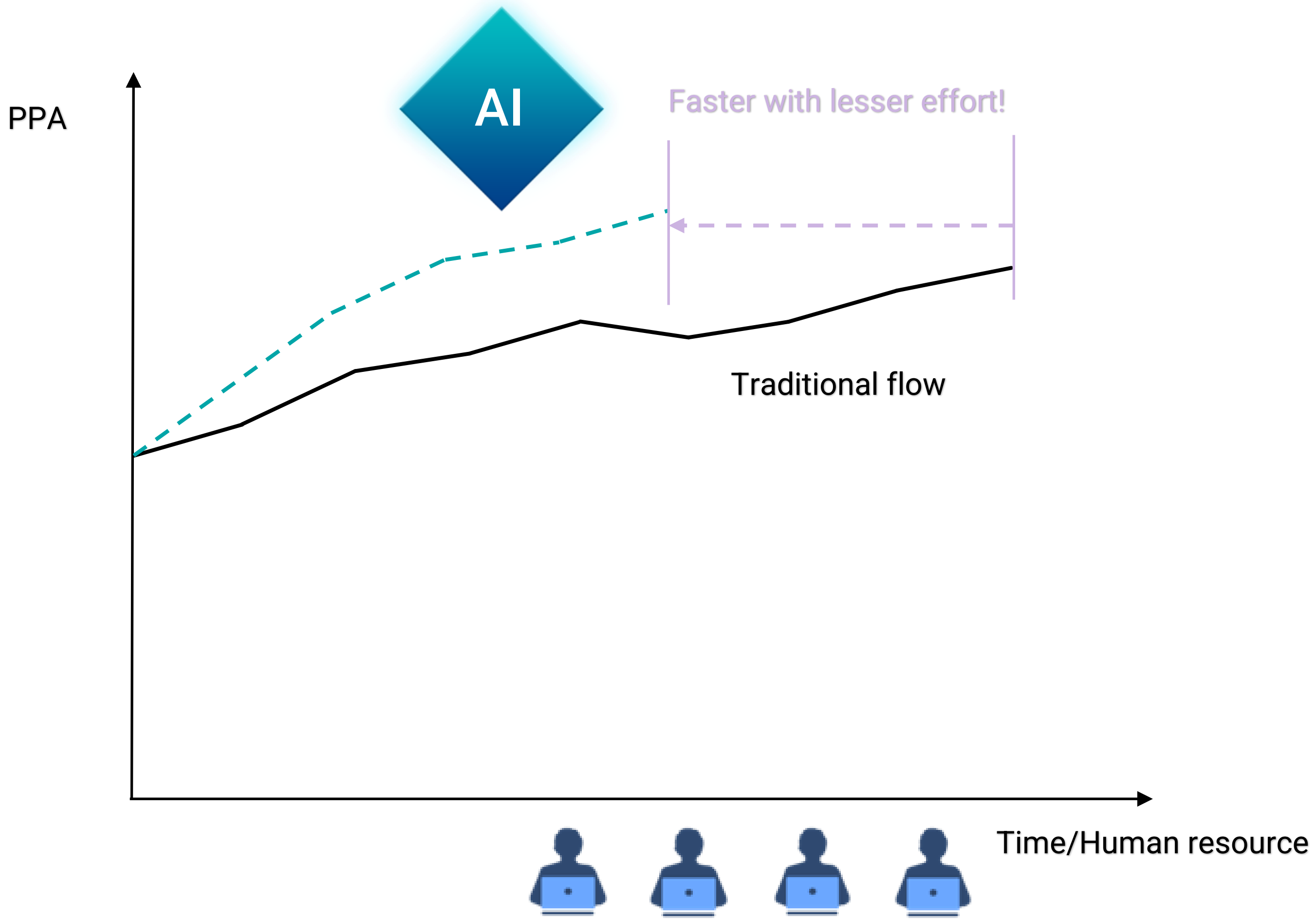
DSO.ai: More exploration, better PPA solutions

DSO.ai: Reduced schedules, faster product turnaround

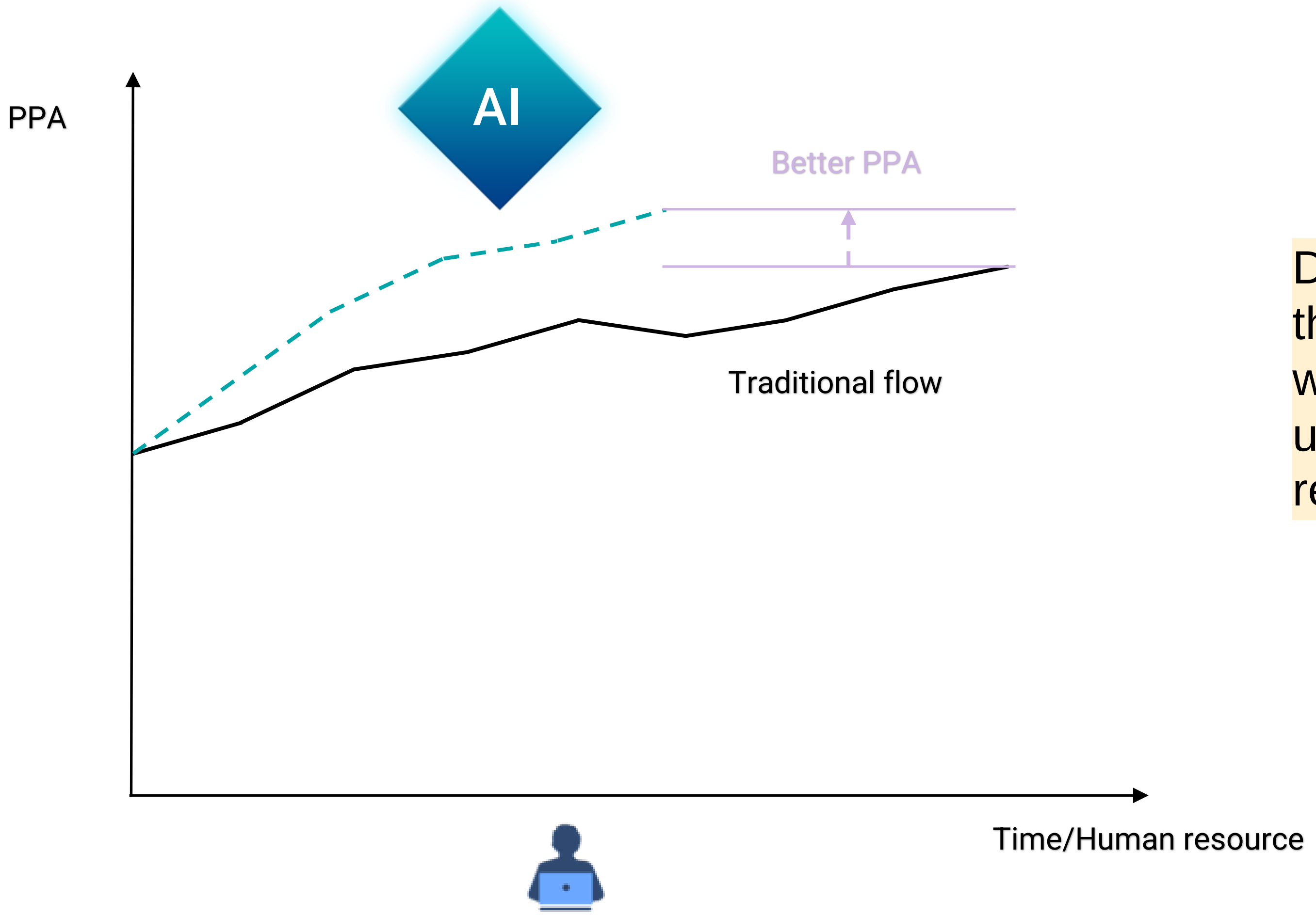
DSO.ai: More products with current design teams

Cost of Result

DSO.ai – Meet Targets Faster, with Less Effort



DSO.ai – Meet Targets Faster, with Less Effort



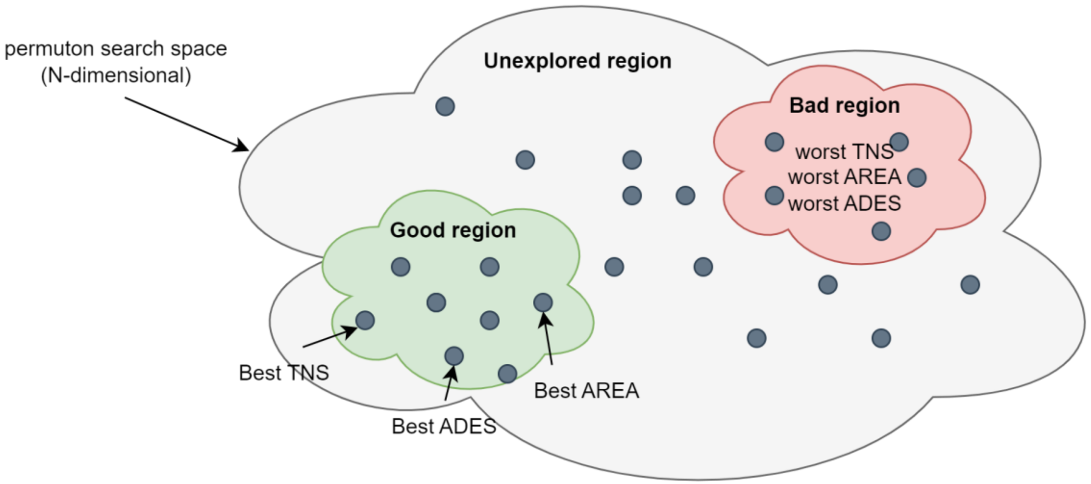
DSO.ai offers a scalable solution for accelerating the development of next-generation IC design while minimizing the design cycles and resource utilization which are very important factors to reduce the time to market.

DSO.ai Introduction

DSO.ai Introduction

What is DSO.ai

- The **Design Space Optimization** (DSO.ai) tool is a machine learning application that delivers better power, performance, and area (PPA) by exploring the search space for a design and evaluating the results across a set of user specified metrics.
- The search space consists of the inputs that impact PPA, such as tool application options and design settings. In the DSO.ai tool, these design inputs are called **permutons**.



DSO.ai Terminology

- Permutons
  - Input parameters DSO.ai explores (app options, flow variations, design settings, etc)
  - Permutons define the search space
- Metrics
  - Output metrics DSO.ai optimizes (TNS, WNS, AREA, ADES, etc)
- ADES (Aggregate Design score)
  - A scoring system that combines multiple output metrics to give the rating of the runs.
- Cold start
  - DSO.ai session started from scratch
  - Initial runs intelligently cover permuton space; true Machine Learning kicks in later in the session
- Warm start
  - DSO.ai learn from a prior session which immediately applied to initial runs (and continues for the rest of session)

Warm-Start Overview

- Speeds up the search process by learning from prior session which reduce the compute resources and potentially improve QoR
- Prior session must be similar to current session
- Warm start automatically retrains the optimizer and adjust the learning strategy based on prior session relevance

Prior Session Relevance	Learning Strategy
Low	Search only ‘good’ regions
Medium	Keep exploring entire space
High	Rerun specific best results

- Use warm start whenever you have a relevant prior session

DSO.ai Setup

DSO.ai Setup

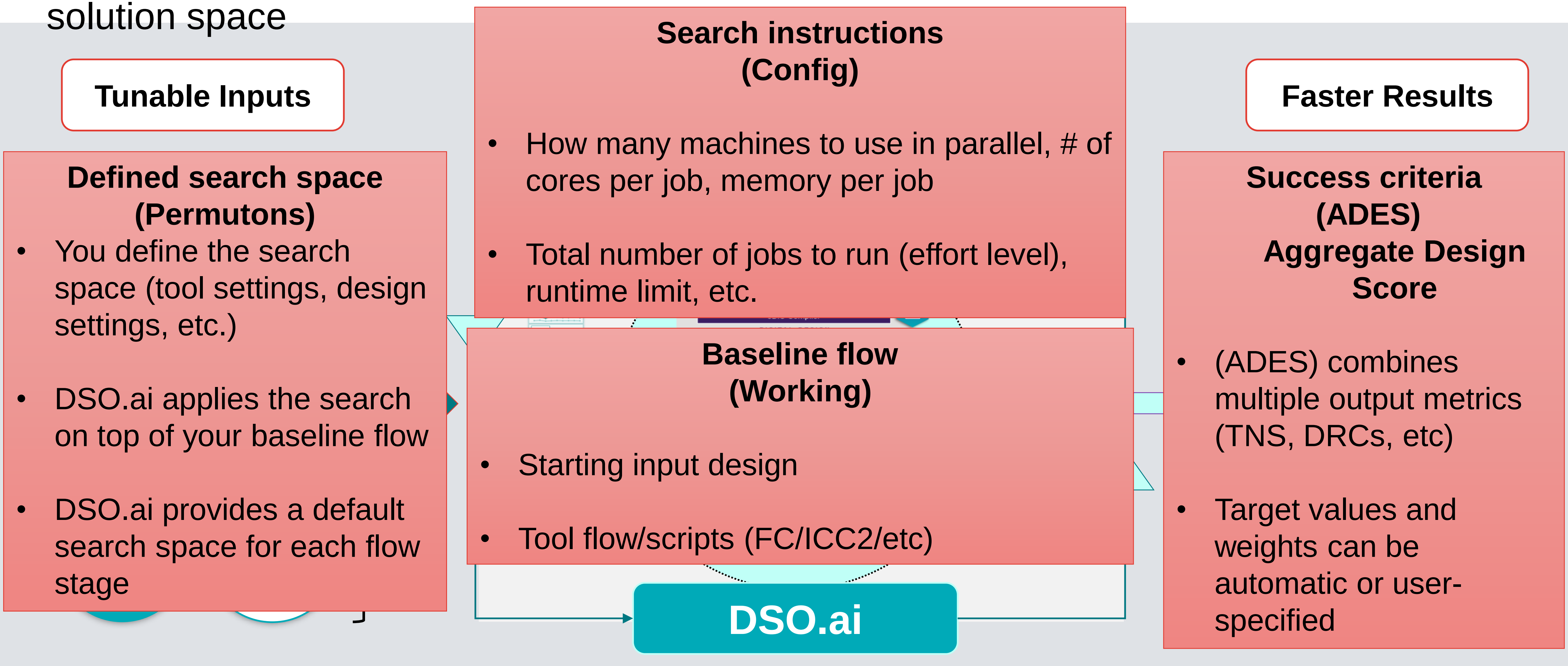
- Tool flow/scripts (FC/ICC2/etc)
  - – No special modification needed
  - – Some consideration for breaking up flow for DSO.ai

- Parameter space (input control)
  - - Default parameter space provided for each flow stage
  - - User can add specific tool settings, design parameters (like uncertainty, max_tran, etc)

- Success metrics (result scoring)
  - - Aggregate Design Score (ADES) combines multiple output metrics (like Wns ,TNS, DRCs, etc)
  - - Target values and weights can be automatic or user-specified

DSO.ai – Design Space Optimization Loop

Uses reinforcement-learning to navigate the design-technology solution space



Tunable Inputs

Defined search space (Permutons)

- You define the search space (tool settings, design settings, etc.)
- DSO.ai applies the search on top of your baseline flow
- DSO.ai provides a default search space for each flow stage

Search instructions (Config)

- How many machines to use in parallel, # of cores per job, memory per job
- Total number of jobs to run (effort level), runtime limit, etc.

Baseline flow (Working)

- Starting input design
- Tool flow/scripts (FC/ICC2/etc)

DSO.ai

Faster Results

Success criteria (ADES) Aggregate Design Score

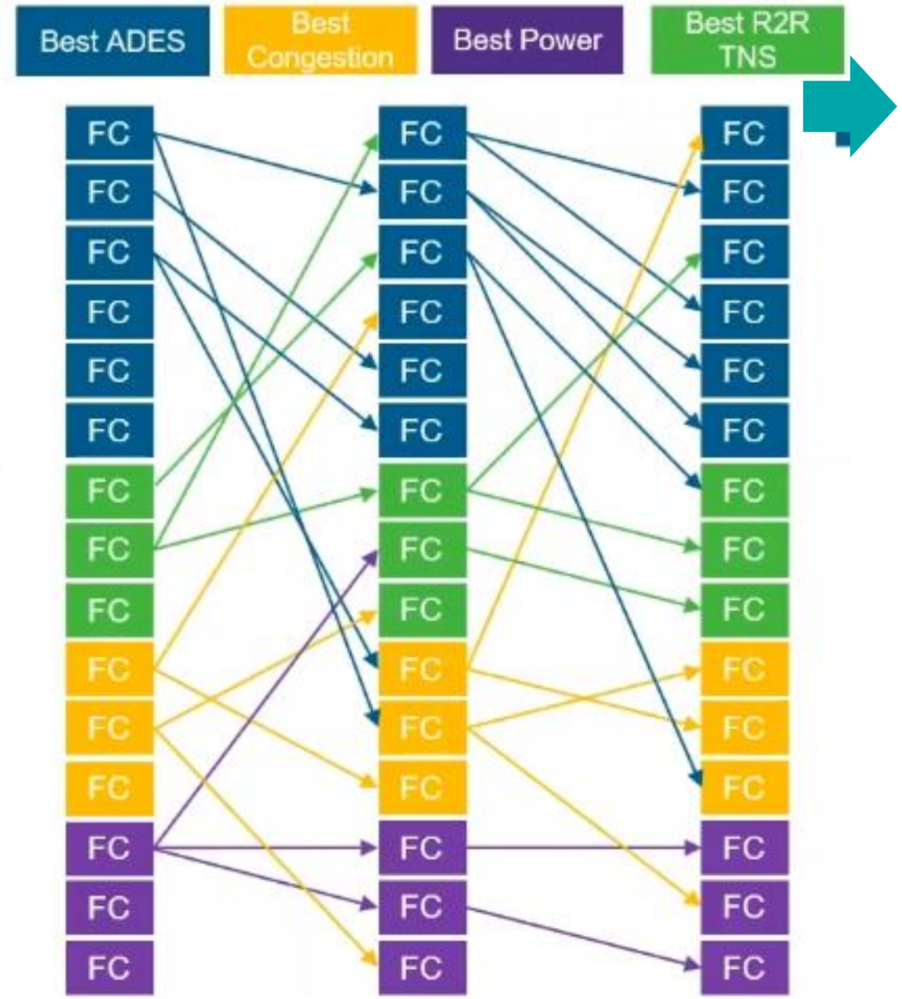
- (ADES) combines multiple output metrics (TNS, DRCs, etc)
- Target values and weights can be automatic or user-specified

Flow Slicing

Flow slicing

- Best run at preCTS stage may not be best at end of route_opt due to various design/tool/correlation factors
- Taking multiple runs forward from synthesis through route with different QOR profiles increases probability of achieving better convergence/PPA at the end
- Can explore more pemutons but no direct visibility to final results

“better” is determined using multiple metrics (TNS, area, power, etc.)



Final best run at the end of ***route_opt***

DSO Place slice

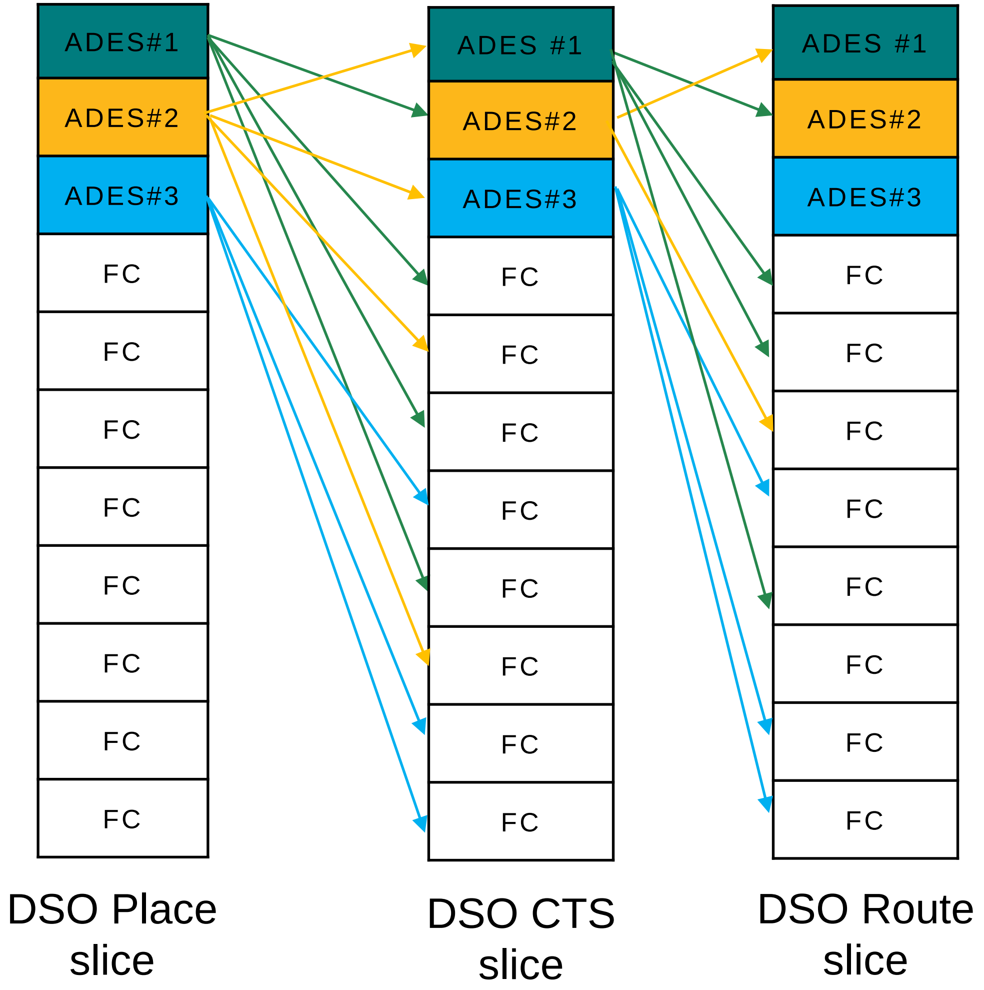
DSO CTS slice

DSO Route slice

MediaTek DSO Flow Slicing Methodology

- 30 database (DB) will be generated in place slice, only top 3 ADES DB will be fetched to the next slices.
- Similarly, same fetch rule is applied to clock and route slice to ensure the best PPA DB at the end of route_opt.

```
Set_run_fetch_rule \
        -session $previous_slice \
        -include_ties false \
        -metrics {ADES} \
        -num_runs 3
```



MediaTek “Place” DSO flow

- Resulting in multiple iterations, ad-hoc decisions

- Controls what output metrics DSO.ai uses to judge success
- We use a single aggregate metric called “ADES” (Aggregate DesignScore)
  - ADES can be automatic (no explicit Metric specification)
  - For more control, a full specification of the components can be added to the aggregate metric
- During DSO run ‘Baseline’ flow is run in parallel with Optimization runs, where no Permuting is performed
- Following ADES is created for MTK run:

```
create_aggregate_metric -name ADES \
        -component CONGESTION -weight 0.5 \
        -component dso_mtk_TNS -weight 1.0 \
        -component dso_mtk_WNS -weight 1.5 \
        -component total_power -weight 2.0
```

The ADES is created based on the customized metrics explained in the following slides

MediaTek “Place” DSO flow

Metric setup

Customize metrics is created to monitor the Fmax of the R2R and IF timing paths.

```
create_metric \
        -name dso_mtk_setup_WNS \
        -direction maximize \
        -procedure “dso_gpu_setup_qor wns”
        -procedure_file {./dso_metrics.tcl}
```

MTK procedure to collect the WNS from all the scenarios

```
create_metric \
        -name dso_mtk_setup_TNS \
        -direction maximize \
        -procedure “dso_gpu_setup_qor tns”
        -procedure_file {./dso_metrics.tcl}
```

MTK procedure to collect the TNS from all the scenarios

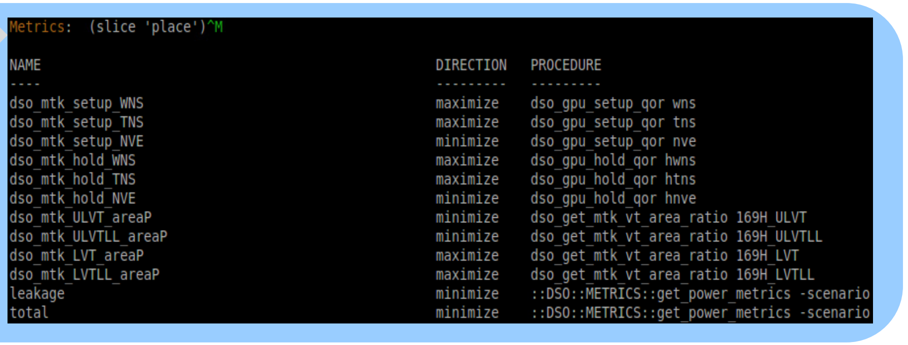
Customize metrics is created to monitor the extra leaky standard cell area.

```
create_metric \
        -name dso_mtk_ULVT_areaP \
        -direction minimize \
        -procedure “dso_get_mtk_vt_area_ratio 169H_ULVT”
        -procedure_file {./dso_metrics.tcl}
```

MediaTek Metrics Setting for Each Slice

Place slice

Metrics Setting

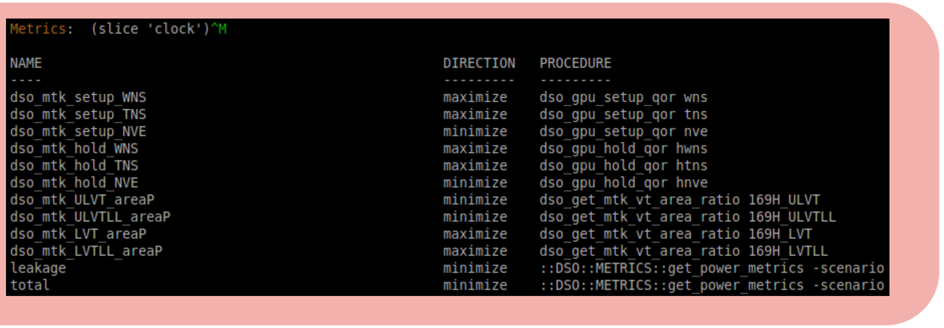


```
Metrics:  (slice 'place')^M

NAME                                                    DIRECTION   PROCEDURE
----                                                    ---------   ---------
dso_mtk_setup_WNS                                       maximize    dso_gpu_setup_qor wns
dso_mtk_setup_TNS                                       maximize    dso_gpu_setup_qor tns
dso_mtk_setup_NVE                                       minimize    dso_gpu_setup_qor nve
dso_mtk_hold_WNS                                        maximize    dso_gpu_hold_qor hwns
dso_mtk_hold_TNS                                        maximize    dso_gpu_hold_qor htns
dso_mtk_hold_NVE                                        minimize    dso_gpu_hold_qor hnve
dso_mtk_ULVT_areaP                                      minimize    dso_get_mtk_vt_area_ratio 169H_ULVT
dso_mtk_ULVTLL_areaP                                    minimize    dso_get_mtk_vt_area_ratio 169H_ULVTLL
dso_mtk_LVT_areaP                                       maximize    dso_get_mtk_vt_area_ratio 169H_LVT
dso_mtk_LVTLL_areaP                                     maximize    dso_get_mtk_vt_area_ratio 169H_LVTLL
leakage                                                 minimize    ::DSO::METRICS::get_power_metrics -scenario
total                                                   minimize    ::DSO::METRICS::get_power_metrics -scenario
```

Clock slice

Metrics Setting

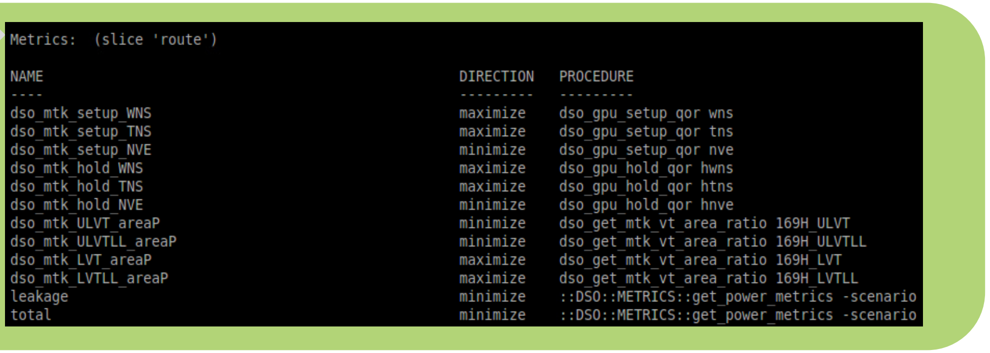


```
Metrics:  (slice 'clock')^M

NAME                                                    DIRECTION   PROCEDURE
----                                                    ---------   ---------
dso_mtk_setup_WNS                                       maximize    dso_gpu_setup_qor wns
dso_mtk_setup_TNS                                       maximize    dso_gpu_setup_qor tns
dso_mtk_setup_NVE                                       minimize    dso_gpu_setup_qor nve
dso_mtk_hold_WNS                                        maximize    dso_gpu_hold_qor hwns
dso_mtk_hold_TNS                                        maximize    dso_gpu_hold_qor htns
dso_mtk_hold_NVE                                        minimize    dso_gpu_hold_qor hnve
dso_mtk_ULVT_areaP                                      minimize    dso_get_mtk_vt_area_ratio 169H_ULVT
dso_mtk_ULVTLL_areaP                                    minimize    dso_get_mtk_vt_area_ratio 169H_ULVTLL
dso_mtk_LVT_areaP                                       maximize    dso_get_mtk_vt_area_ratio 169H_LVT
dso_mtk_LVTLL_areaP                                     maximize    dso_get_mtk_vt_area_ratio 169H_LVTLL
leakage                                                 minimize    ::DSO::METRICS::get_power_metrics -scenario
total                                                   minimize    ::DSO::METRICS::get_power_metrics -scenario
```

Route slice

Metrics Setting



```
Metrics:  (slice 'route')

NAME                                                    DIRECTION   PROCEDURE
----                                                    ---------   ---------
dso_mtk_setup_WNS                                       maximize    dso_gpu_setup_qor wns
dso_mtk_setup_TNS                                       maximize    dso_gpu_setup_qor tns
dso_mtk_setup_NVE                                       minimize    dso_gpu_setup_qor nve
dso_mtk_hold_WNS                                        maximize    dso_gpu_hold_qor hwns
dso_mtk_hold_TNS                                        maximize    dso_gpu_hold_qor htns
dso_mtk_hold_NVE                                        minimize    dso_gpu_hold_qor hnve
dso_mtk_ULVT_areaP                                      minimize    dso_get_mtk_vt_area_ratio 169H_ULVT
dso_mtk_ULVTLL_areaP                                    minimize    dso_get_mtk_vt_area_ratio 169H_ULVTLL
dso_mtk_LVT_areaP                                       maximize    dso_get_mtk_vt_area_ratio 169H_LVT
dso_mtk_LVTLL_areaP                                     maximize    dso_get_mtk_vt_area_ratio 169H_LVTLL
leakage                                                 minimize    ::DSO::METRICS::get_power_metrics -scenario
total                                                   minimize    ::DSO::METRICS::get_power_metrics -scenario
```

MediaTek “Place” DSO flow

Permutons setup

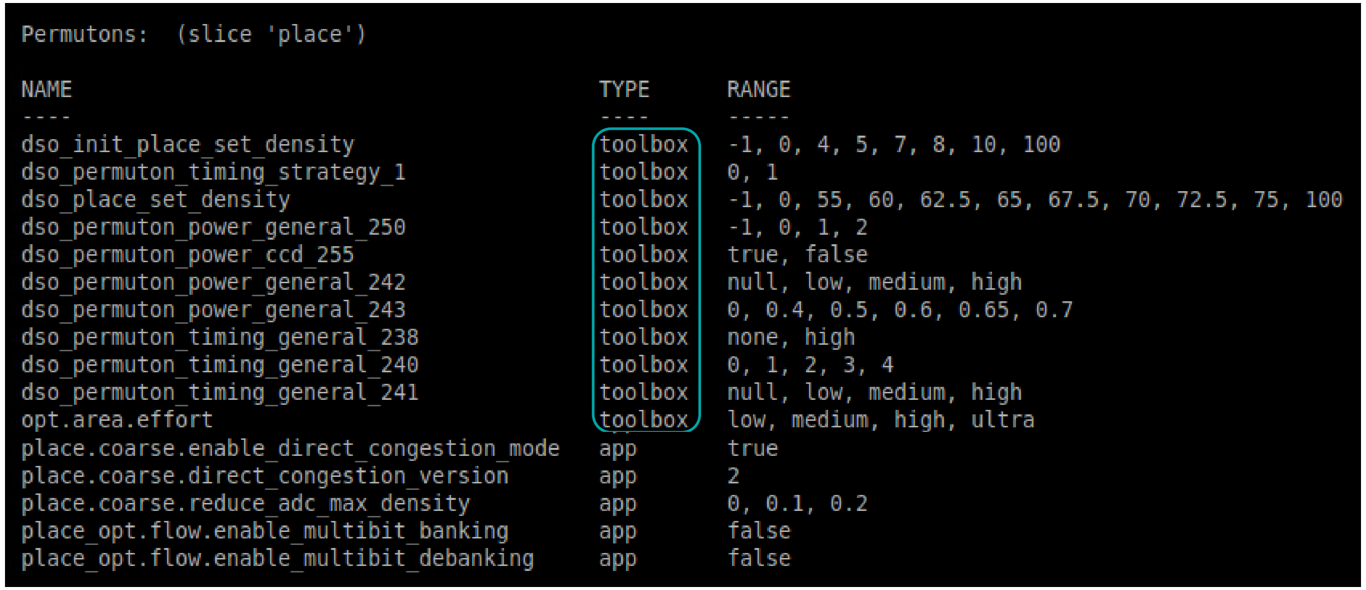
Use the new low power permutons from the default toolboxes:

add_permutons [get_place_power_default_permutons]

Improved ease of use

You can also define your own permutons:

```
create_permuton \
        -name place.coarse.enable_direct_congestion_mode \
        -type app \
        -range {true} \
        -datatype categorical
create_permuton \
        -name place.coarse.direct_congestion_version \
        -type app \
        -range {2} \
        -datatype categorical
```



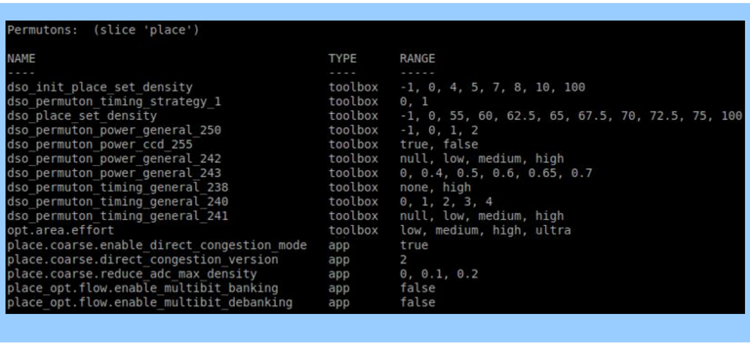
```
Permutons:  (slice 'place')

NAME                                          TYPE      RANGE
----                                          ----      -----
dso_init_place_set_density                    toolbox   -1, 0, 4, 5, 7, 8, 10, 100
dso_permuton_timing_strategy_1                toolbox   0, 1
dso_place_set_density                         toolbox   -1, 0, 55, 60, 62.5, 65, 67.5, 70, 72.5, 75, 100
dso_permuton_power_general_250                toolbox   -1, 0, 1, 2
dso_permuton_power_ccd_255                    toolbox   true, false
dso_permuton_power_general_242                toolbox   null, low, medium, high
dso_permuton_power_general_243                toolbox   0, 0.4, 0.5, 0.6, 0.65, 0.7
dso_permuton_timing_general_238               toolbox   none, high
dso_permuton_timing_general_240               toolbox   0, 1, 2, 3, 4
dso_permuton_timing_general_241               toolbox   null, low, medium, high
opt.area.effort                               toolbox   low, medium, high, ultra
place.coarse.enable_direct_congestion_mode    app       true
place.coarse.direct_congestion_version        app       2
place.coarse.reduce_adc_max_density           app       0, 0.1, 0.2
place_opt.flow.enable_multibit_banking        app       false
place_opt.flow.enable_multibit_debanking      app       false
```

MediaTek Permutons Setting for Each Slice

Place slice

Permutons Setting

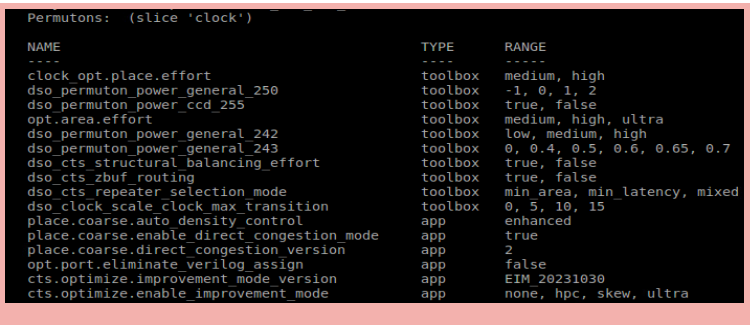


```
Permutons:  (slice 'place')

NAME                                  TYPE      RANGE
----                                  ----      -----
dso_init_place_set_density            toolbox   -1, 0, 4, 5, 7, 8, 10, 100
dso_permuton_timing_strategy_1        toolbox   0, 1
dso_place_set_density                 toolbox   -1, 0, 55, 60, 62.5, 65, 67.5, 70, 72.5, 75, 100
dso_permuton_power_general_250        toolbox   -1, 0, 1, 2
dso_permuton_power_ccd_255            toolbox   true, false
dso_permuton_power_general_242        toolbox   null, low, medium, high
dso_permuton_power_general_243        toolbox   0, 0.4, 0.5, 0.6, 0.65, 0.7
dso_permuton_timing_general_238       toolbox   none, high
dso_permuton_timing_general_240       toolbox   0, 1, 2, 3, 4
dso_permuton_timing_general_241       toolbox   null, low, medium, high
opt.area.effort                       toolbox   low, medium, high, ultra
place.coarse.enable_direct_congestion_mode   app       true
place.coarse.direct_congestion_version       app       2
place.coarse.reduce_adc_max_density          app       0, 0.1, 0.2
place_opt.flow.enable_multibit_banking       app       false
place_opt.flow.enable_multibit_debanking     app       false
```

Clock slice

Permutons Setting

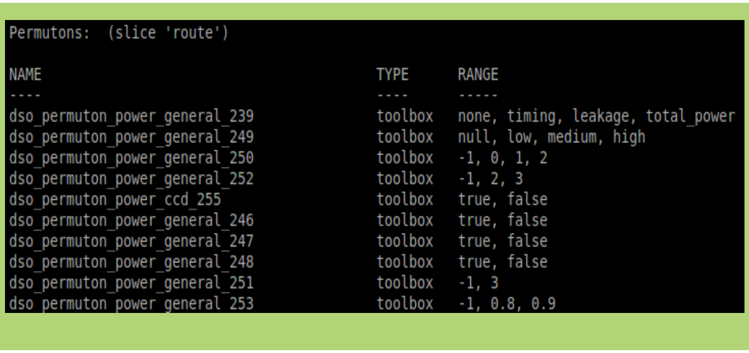


```
Permutons:  (slice 'clock')

NAME                                        TYPE      RANGE
----                                        ----      -----
clock_opt.place.effort                      toolbox   medium, high
dso_permuton_power_general_250              toolbox   -1, 0, 1, 2
dso_permuton_power_ccd_255                  toolbox   true, false
opt.area.effort                             toolbox   medium, high, ultra
dso_permuton_power_general_242              toolbox   low, medium, high
dso_permuton_power_general_243              toolbox   0, 0.4, 0.5, 0.6, 0.65, 0.7
dso_cts_structural_balancing_effort         toolbox   true, false
dso_cts_zbuf_routing                        toolbox   true, false
dso_cts_repeater_selection_mode             toolbox   min_area, min_latency, mixed
dso_clock_scale_clock_max_transition        toolbox   0, 5, 10, 15
place.coarse.auto_density_control           app       enhanced
place.coarse.enable_direct_congestion_mode  app       true
place.coarse.direct_congestion_version      app       2
opt.port.eliminate_verilog_assign           app       false
cts.optimize.improvement_mode_version       app       EIM_20231030
cts.optimize.enable_improvement_mode        app       none, hpc, skew, ultra
```

Route slice

Permutons Setting



```
Permutons:  (slice 'route')

NAME                              TYPE      RANGE
----                              ----      -----
dso_permuton_power_general_239    toolbox   none, timing, leakage, total_power
dso_permuton_power_general_249    toolbox   null, low, medium, high
dso_permuton_power_general_250    toolbox   -1, 0, 1, 2
dso_permuton_power_general_252    toolbox   -1, 2, 3
dso_permuton_power_ccd_255        toolbox   true, false
dso_permuton_power_general_246    toolbox   true, false
dso_permuton_power_general_247    toolbox   true, false
dso_permuton_power_general_248    toolbox   true, false
dso_permuton_power_general_251    toolbox   -1, 3
dso_permuton_power_general_253    toolbox   -1, 0.8, 0.9
```

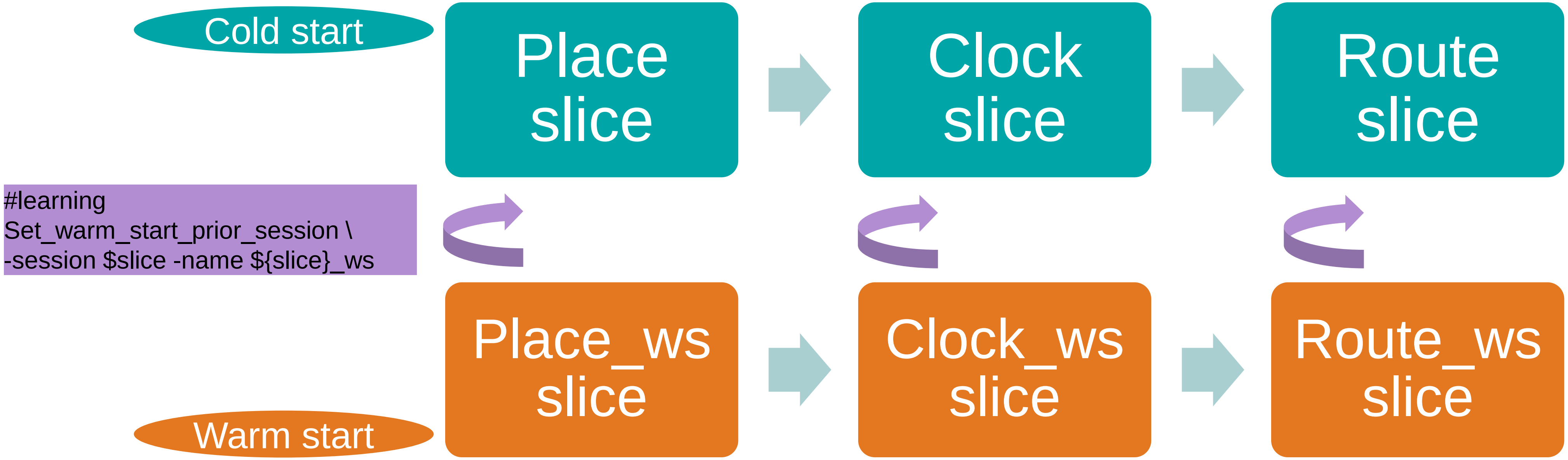
MediaTek Warm-Start Setup

- Changes always happen until tapeout
- Warm start is designed to work in this environment

```
Set_compute_options –parallel_effort 5

#Replay top 2 ADES runs from prior session and generate 3 new suggestions

Set_warm_start_prior_session -name place_ws -type dir -session ./place -relevance medium -num_runs 2
```



```
#learning
Set_warm_start_prior_session \
-session $slice -name ${slice}_ws
```

Results

DSO Final Result comparison

Compare b/t baseline and cold start

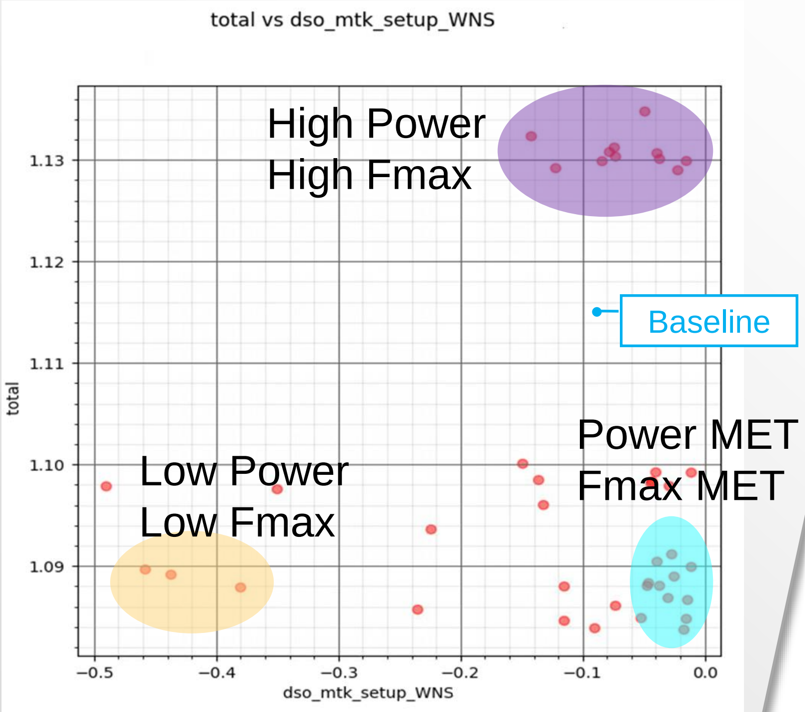
Stage		Baseline	DSO Cold start (90runs)
	Std cell area	100%	92.60%
IP Power (mW) (PTPX run)	Logic Dyn. Pwr	100%	90.45%
	Logic Leakage	100%	93.82%
	Total	100%	91.01%
Total Setup Timing (WNS/TNS/NVP)		-0.080/-100/5544	-0.005/-60/3879
Total Hold Timing (WNS/TNS/NVP)		-0.019/-0.168/30	-0.02/-0.03/6
Shorts		30	10
VT area ratio (%)	VT1	100%	100%
	VT2	100%	85.28%
	VT3	100%	98.63%
	VT4	100%	100.00%

Summary

- **2.8% fmax improvement**
- **8% std area improvement**
- **9% Max total power improvement**

AI-Driven Optimization
AI-driven RTL-to-GDSII design implementation

AI-Driven Optimization



All Targets MET
1 human x 10 days (90 runs)

	Perf.	Power	Area
Target User Expectation	100%Ghz	100%	MET
Baseline Reference flow	89%Ghz	109%	MET
AI	**10 days, 90 runs**		
DSO.ai Ai-driven	**99%Ghz**	**99%**	**MET**

DSO Final Result comparison

Compare b/t cold start and warm start

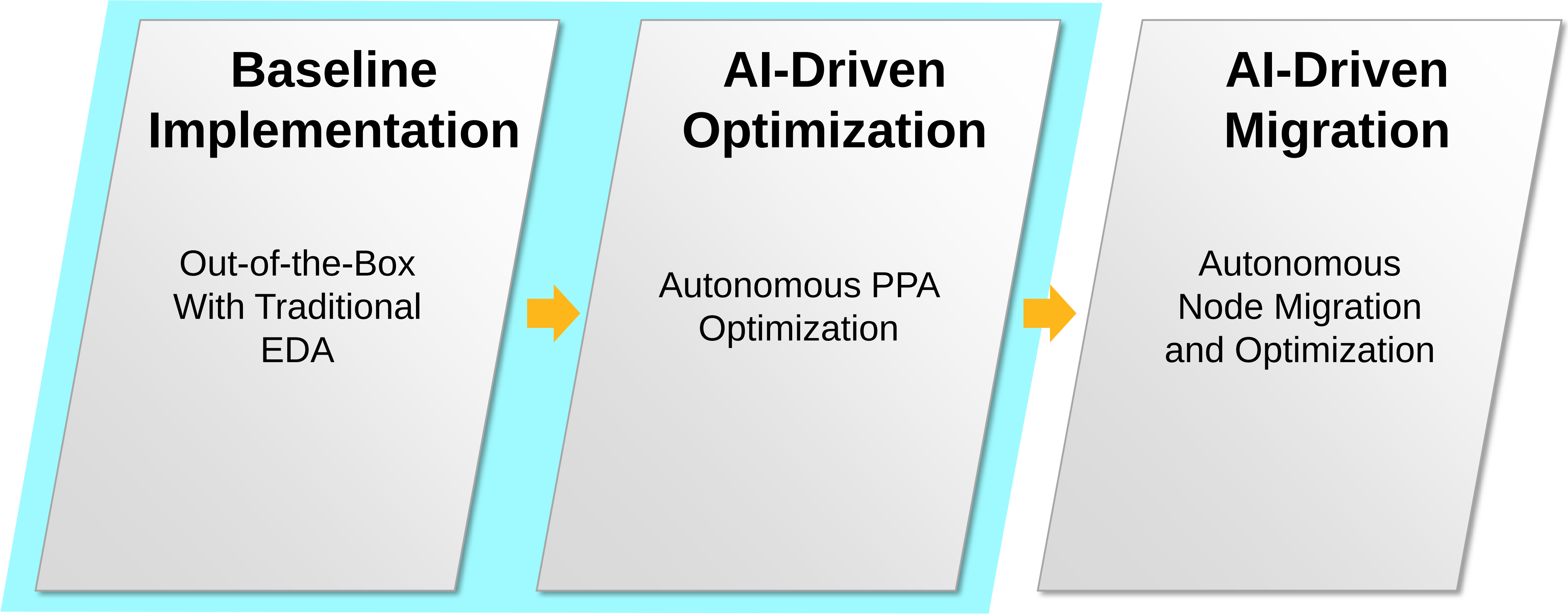
Stage		DSO Cold start (90runs)	DSO warm start (15runs)
	Std cell area	92.60%	92.58%
IP Power (mW) (PTPX run)	Logic Dyn. Pwr	90.45%	90.60%
	Logic Leakage	93.82%	93.80%
	Total	91.01%	91.01%
Total Setup Timing (WNS/TNS/NVP)		-0.005/-60/3879	-0.006/-60/3888
Total Hold Timing (WNS/TNS/NVP)		-0.02/-0.03/6	-0.02/-0.03/6
Shorts		10	10
VT area ratio (%)	VT1	100%	100%
	VT2	85.28%	85%
	VT3	98.63%	98.6%
	VT4	100.00%	100%

- DSO Warm Start with only 5 workers per slice (compared to 30 workers per slice in cold start) was deployed to achieve PPA convergence after Functional ECO, optimize machine resources by leveraging on training model from cold start run.

Conclusions

Achieving PPA Targets with DSO.ai

AI-driven Digital Design PPA Optimization



Baseline Implementation

Out-of-the-Box With Traditional EDA

AI-Driven Optimization

Autonomous PPA Optimization

AI-Driven Migration

Autonomous Node Migration and Optimization

Summary – Targets Achieved Autonomously with DSO.ai

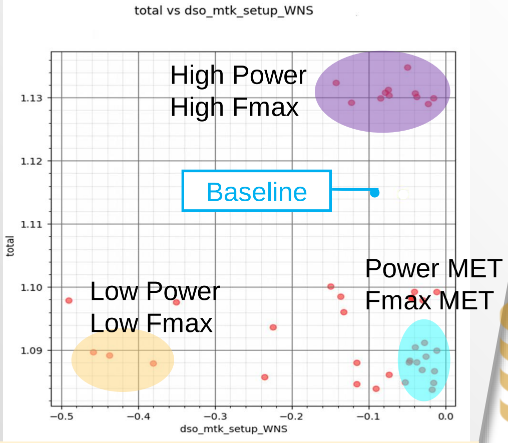
AI-driven Digital Design PPA Optimization

Baseline Implementation

Perf.: ~89%
Power: ~109%
Area: **MET**

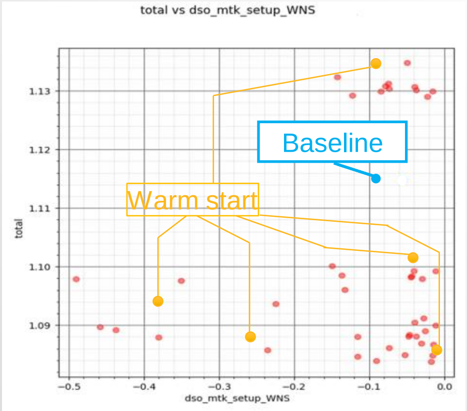
Estimated Time-to-Target
2 experts x 1 months

AI-Driven Optimization



All Targets MET
1 human x 10 days (90 runs)

AI-Driven Migration



All Targets MET
1 human x 7 day (15 runs)

DSO.ai Achieves PPA Targets Autonomously in Days vs. Weeks of Human Effort!

Conclusions

DSO.ai has been deployed for multiple MTK project with significant total power and Fmax improvement

The AI driven optimization engine save additional effort and engineer resources for PPA push run.

THANK YOU

Our
Technology,
Your
Innovation™