Power Optimization Flows in Deep Sub-Micron Technologies for Computer Vision Systems

A Comprehensive Framework for Energy-Efficient Design

Ajay Mehta (Staff Engineer)
Bhanusekhar Kalaga (Senior Staff Engineer)
Vamsi Chakradhar (Staff Engineer)
MediaTek Bangalore Pvt Ltd

AGENDA

- INTRODUCTION
- FLOW PRACTICES
- SAIF BASED POWER OPTIMIZATION
- IN-DESIGN PRIME POWER
- RESULTS
- CONCLUSION
- Q&A

INTRODUCTION

CHALLENGES

Computer Vision System

Low Power 4nm chip, critical metric dynamic power

Quality of Results

When focus on power then QoR gets impacted

Turn Around Time

Multiple experiment to improve QoR & for signoff PTPX

SOLUTIONS

Flow Practices

Dynamic Power Optimization using Fusion Compiler app options

SAIF FLOW

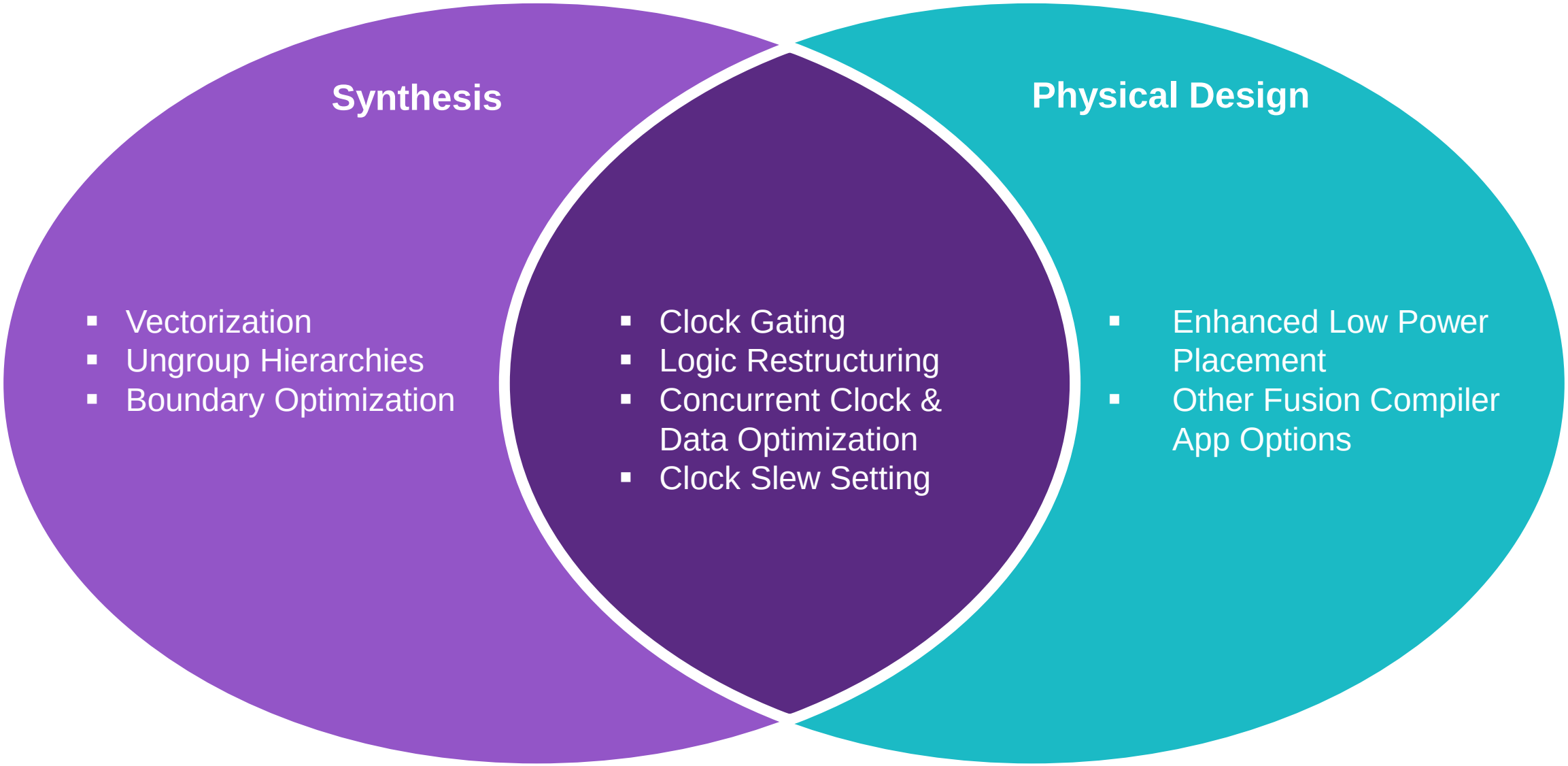
Dynamic Power Optimization using RTL SAIF file

In-Design Prime Power

Dynamic Power Optimization using In-Design Prime Power

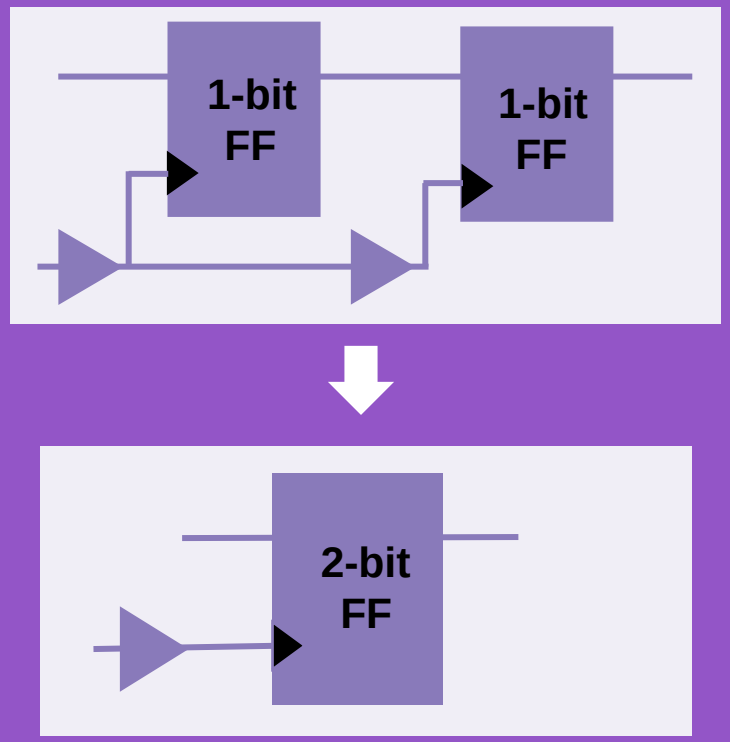
FLOW PRACTICES

Flow Practices For Power Optimization



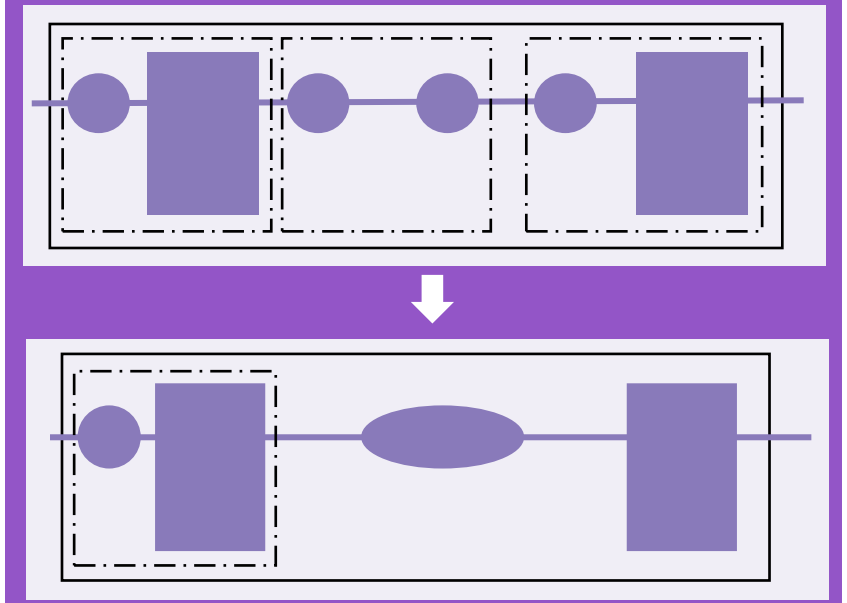
Flow Practices For Power Optimization

Vectorization



- Reduce clock tree cell & net length
- Reduce reg area
- App Option:
  compile.flow.enable_multibit true

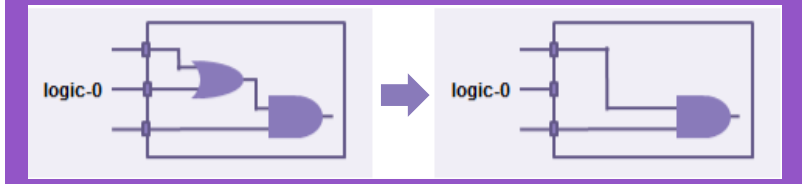
Ungroup Hierarchies



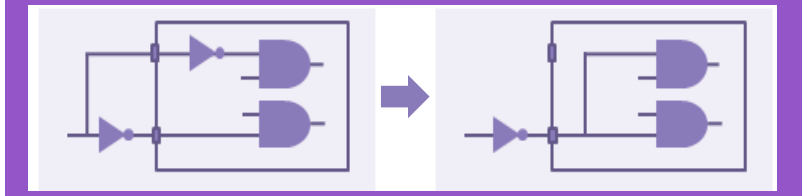
- Regrouping at flat or hierarchical level
- App Option:
  compile.flow.autoungroup true

Boundary Optimization

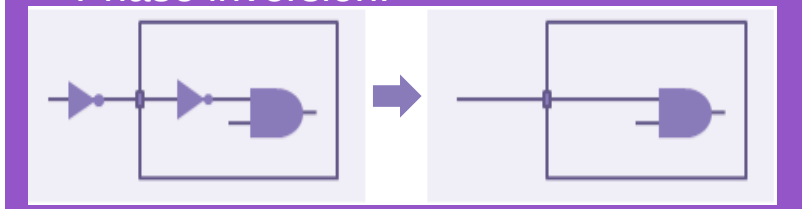
- Propagation of constant:



- Propagation of equal/opposite:



- Phase inversion:

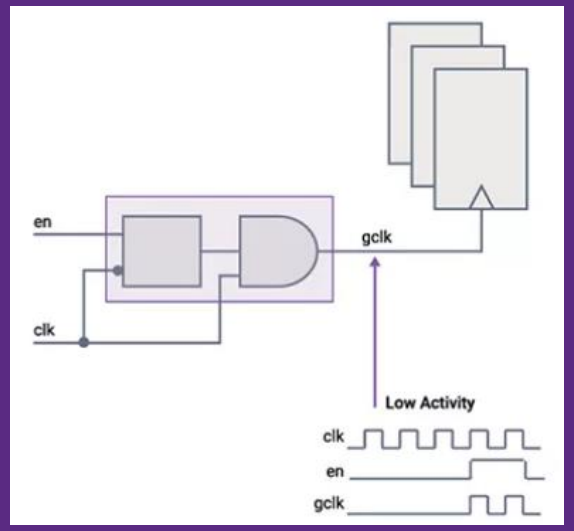


- Propagate unconnected port information.
- App Option:
  compile.flow.boundary_optimization true

- Synthesis

Flow Practices For Power Optimization

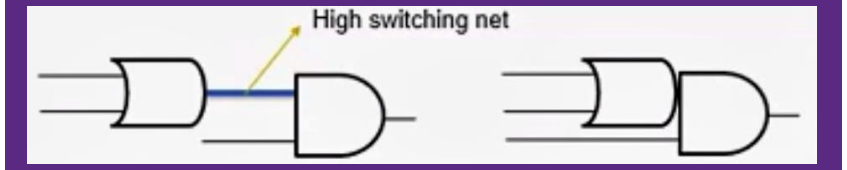
Clock Gating



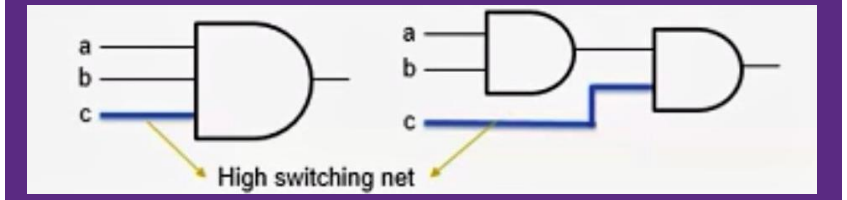
- Cutoff high frequency functional clocks during ideal state.
- PnR controls placement of clock gating cell.

Logic Restructuring

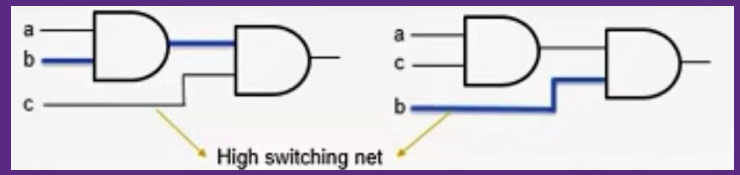
- Composition: Absorb in complex gate.



- Decomposition: Driven by smaller gate

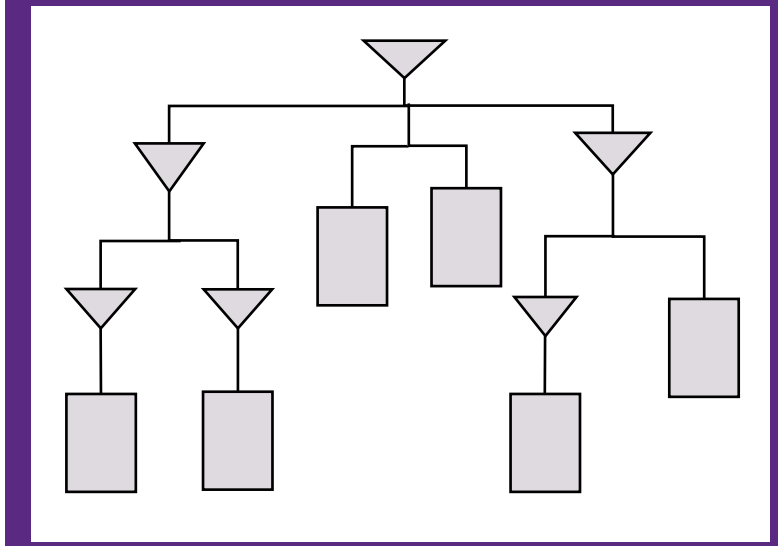


- Rewiring: Feed at last stage



- App Option: opt.common.advanced_logic_ restructuring_mode –value <power or timing_power>

Concurrent Clock & Data Optimization



- Fix timing violation with useful skew
- App Option: compile/clock_opt/route_opt.flow. enable_ccd true clock_opt.flow.enable_clock_power_ recovery -value auto

Flow Practices For Power Optimization

Clock Slew Setting

- Need to set an optimum value
- Too relaxed: clock slope increase short circuit power
- Too tight: over splitting of clock tree, more cells added
- App Option:
set_max_transition -clock_path <slope> [get_clocks *] -scenarios [get_scenarios *]

Enhanced Low Power Placement (eLPP)

- Apply net weights to direct the placer to shorten nets of high activity
- Allow placement to get better balance between power and timing
- App Option:
place.coarse.enhance_low_power_effort -value <low/medium/high>

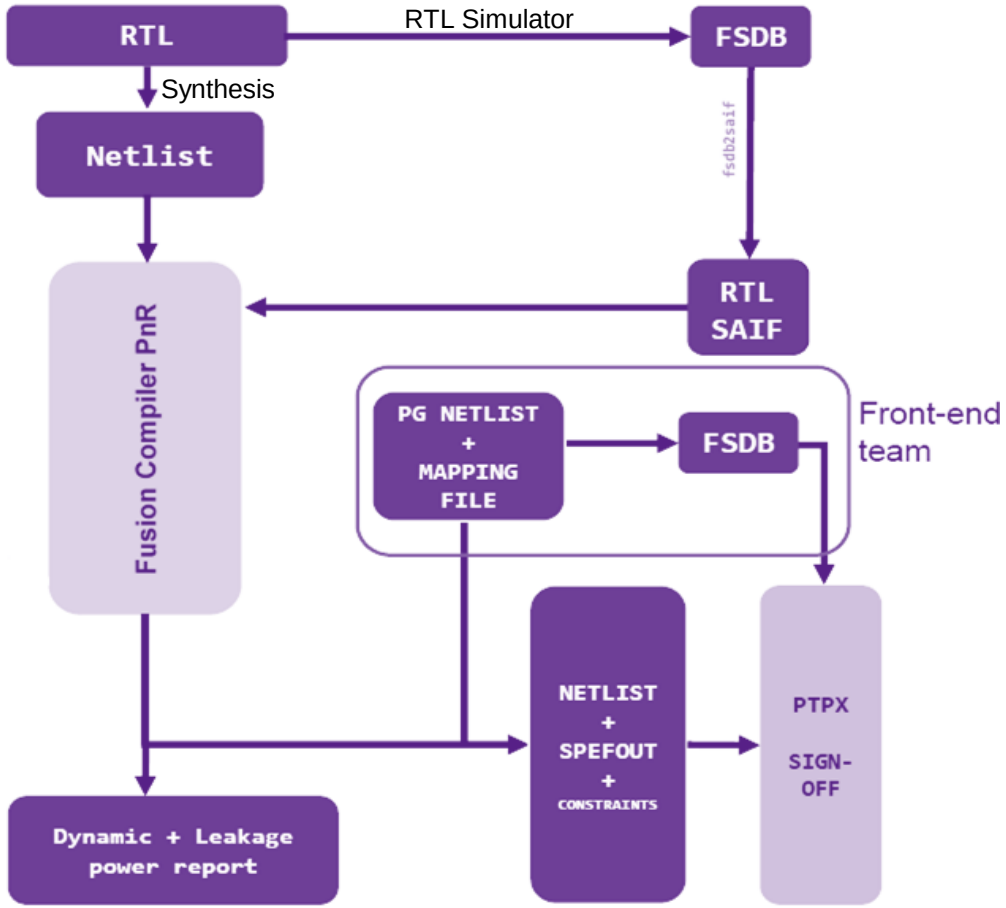
Other Fusion Compiler App Options

- Enable power optimization:
place_opt/ clock_opt/ route_opt.flow.enable_power -value true
- Enable power recovery:
clock_opt/route_opt.flow.enable_clock_power_recovery -value auto
- Configure app options and tool flow:
set_qor_strategy -stage pnr -metric total_power -mode extreme_power

- Synthesis & Physical Design - Physical Design

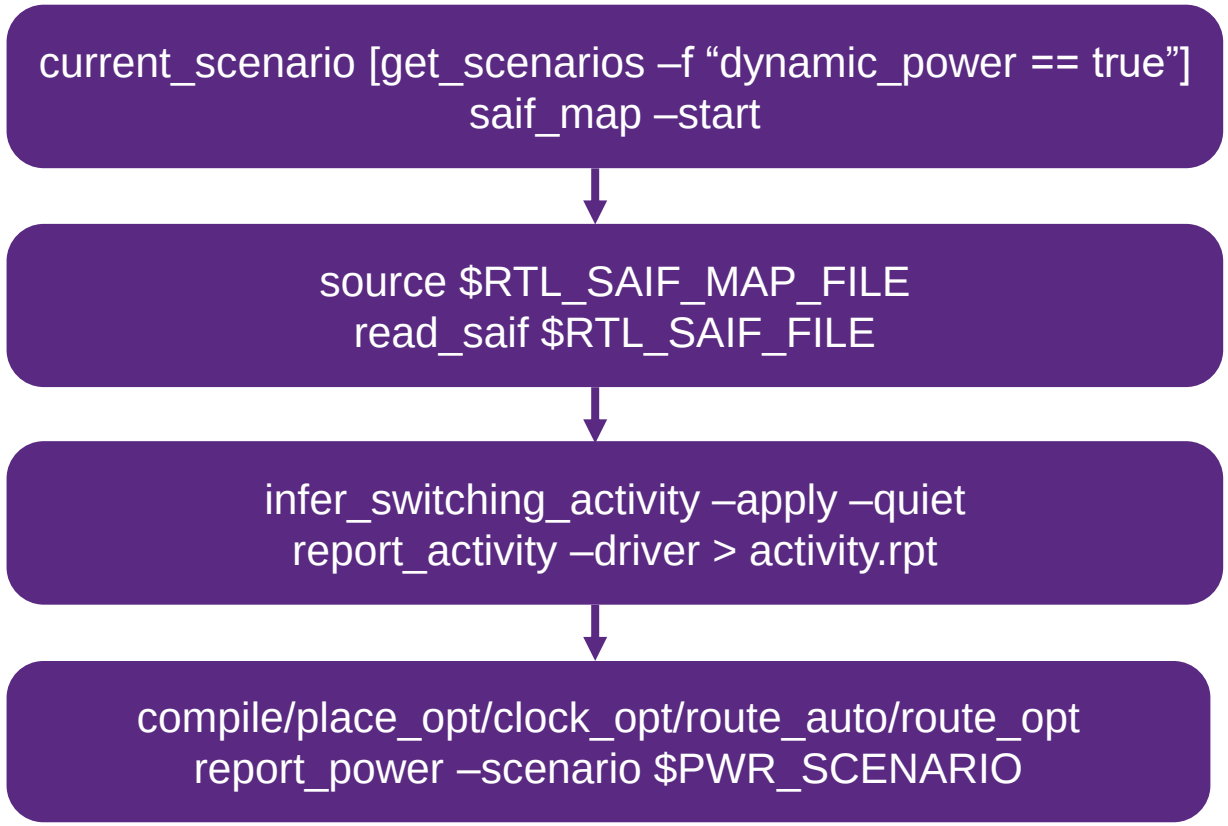
SAIF BASED POWER OPTIMIZATION

RTL SAIF Flow



- For RTL SAIF flow, fusion compiler optimize dynamic power more accurately than for without SAIF flow.
- Challenge: RTL SAIF cannot cover whole netlist's changes throughout optimization steps. So, need to release multiple PnR experiments for PTPX, as power report is not enough to finalize trial before PTPX signoff.

RTL SAIF Setup



- set scenario
- record netlist change, to update saif map file

- read mapping points between RTL & netlist
- read RTL SAIF file

- apply switching activity to activity points
- write activity report

- run optimization step
- write power report

RTL SAIF Sanity Check

- Check coverage:

report_activity –driver > activity.rpt

```
Essential activity is complete
Activity Type    primary-input          seq-pin          icg-pin          comb-pin          no-func     total
-------------------------------------------------------------------------------------------------------------
simulated          817 ( 99.9%)   325614 ( 93.2%)     7422 (100.0%) 2386664 ( 65.6%)     8066 ( 81.2%)   2728583
annotated            0 (  0.0%)        0 (  0.0%)        0 (  0.0%)       0 (  0.0%)        0 (  0.0%)         0
inferred             0 (  0.0%)        0 (  0.0%)        0 (  0.0%)       0 (  0.0%)        0 (  0.0%)         0
derived              1 (  0.1%)        0 (  0.0%)        0 (  0.0%)   83136 (  2.3%)        0 (  0.0%)     83137
calculated           0 (  0.0%)        0 (  0.0%)        0 (  0.0%)       0 (  0.0%)        0 (  0.0%)         0
default              0 (  0.0%)    23764 (  6.8%)        0 (  0.0%) 1167750 ( 32.1%)     1865 ( 18.8%)   1193379
-------------------------------------------------------------------------------------------------------------
total              818 (100.0%)   349378 (100.0%)     7422 (100.0%) 3637550 (100.0%)     9931 (100.0%)   4005099
```

Better coverage, better power optimization by the tool

Missing Objects : report_activity -rtl –print_objects {default {seq-cell tri-cell}} > saif_missing.rpt

Usually includes DFT/TMBIST registers added in synthesis

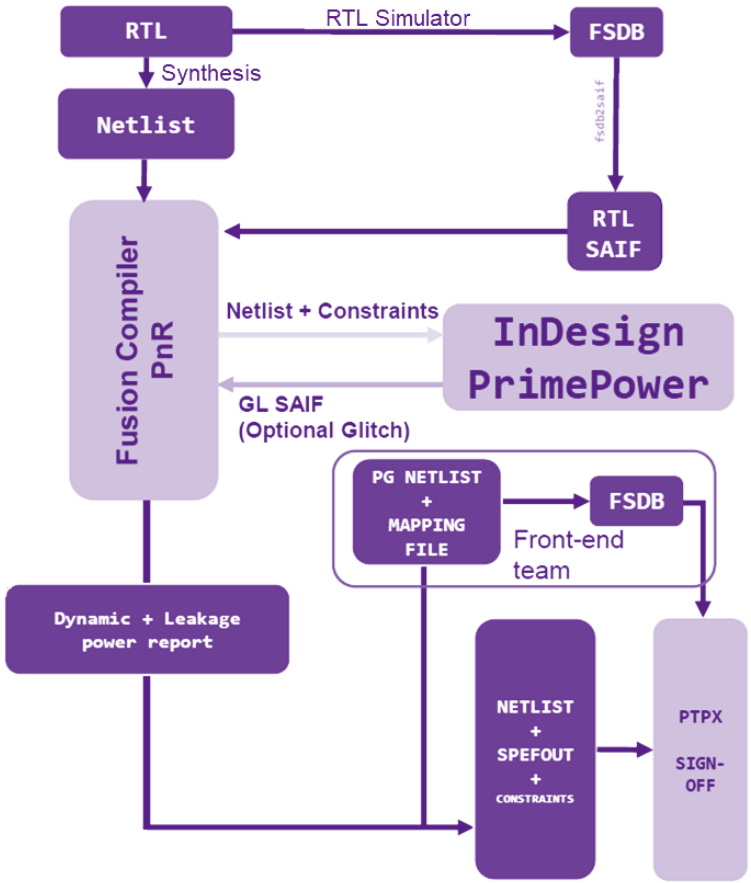
IN-DESIGN PRIME POWER

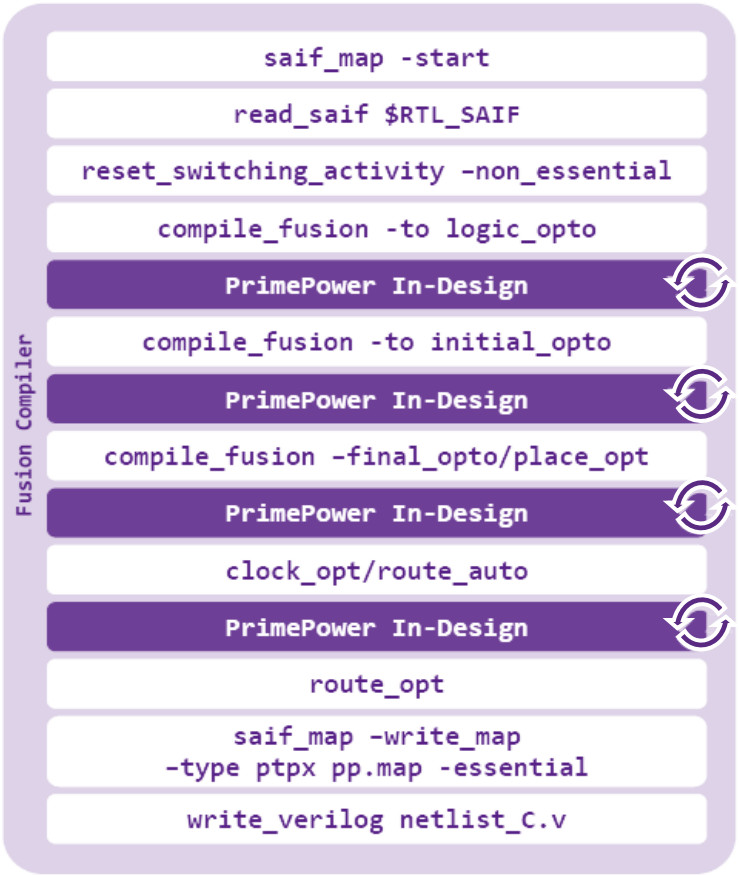
In-Design Prime Power (IDPP)

- Need:

  As netlist change during PnR optimization stage, due to which fusion compiler cannot optimize dynamic power correctly based on RTL SAIF flow.

- Flow:





In-Design Prime Power Setup

```
saif_map –start
read_saif $RTL_SAIF
```

- record netlist change, to update saif map file
- read RTL SAIF file

```
set_indesign_prime_power_options  \
-fsdb {{ $FSDB_FILE –weight 1  \
-strip_path $RTL_FSDB_STRIP_PATH  \
-format systemverilog  \
-analyse_scenarios $PWR_SCENARIO  \
-back_annotate_scenarios $PWR_SCENARIO}}  \
-output_dir $OUT_DIR  \
-max_core 32  \
-pwr_shell $PT_VER  \
-script idpp_setup.tcl
```

- Define fsdb file, weight associated for it, strip path
- Define power scenarios, output directory, max_core, PT version
- Change flow app_var in idpp_setup.tcl to control flow

```
update_indesign_activity –keep all
report_activity –driver –scenarios $PWR_SCENARIO
```

- Invoke In-Design Prime Power to generate GL SAIF & read back in fusion compiler, then write activity report

```
compile/place_opt/clock_opt/route_auto/route_opt
report_power –scenario $PWR_SCENARIO
```

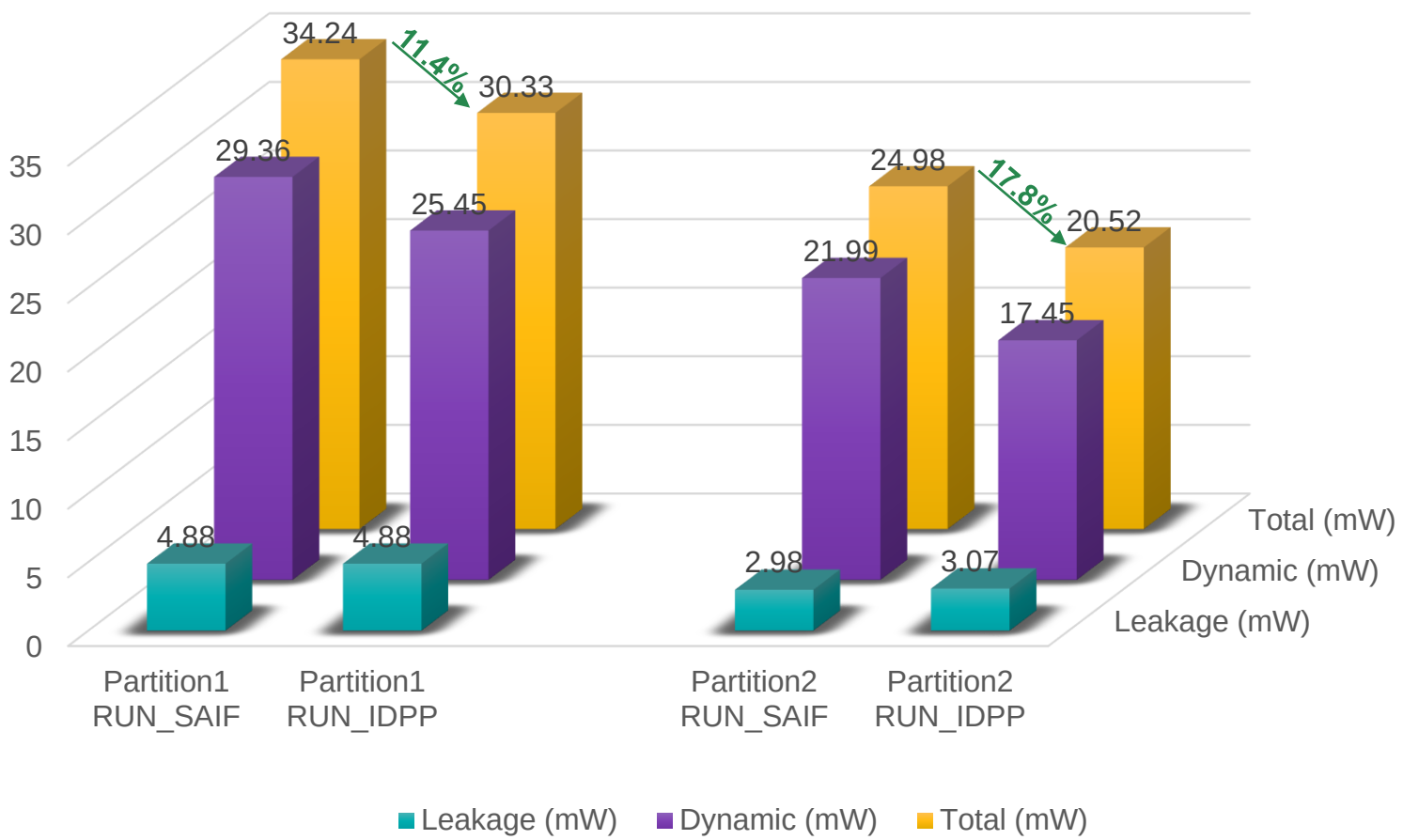
- run optimization step
- write power report

RESULTS

Results

Seq-pin Coverage

Power Comparison



Results

- Partition1 QoR Summary:

	Strategy	Std Cell Count	Congestion	Setup (WNS/TNS/NVP)	Hold (WNS/TNS/NVP)
APR RESULT	RUN_IDPP	3962254	H: 0.298% V: 0.458%	-0.033/-3.97/1560	-0.079/-15.47/5696
	RUN_SAIF	3958351	H: 0.175% V: 0.382%	-0.038/-3.67/1526	-0.105/-20.54/7627

- Partition2 QoR Summary:

	Strategy	Std Cell Count	Congestion	Setup (WNS/TNS/NVP)	Hold (WNS/TNS/NVP)
APR RESULT	RUN_IDPP	1920688	H: 0.11% V: 0.13%	-0.151/-49.79/1940	-0.063/-38.353/3001
	RUN_SAIF	1906054	H: 0.11% V: 0.17%	-0.151/-28.064/1246	-0.077/-21.150/2734

CONCLUSION

Conclusion

1. For run based on In-Design Prime Power flow, dynamic & total power has improved by 10-20%
2. QoR are comparable for both SAIF-based & InDesign Prime Power run
3. So, for power critical design, In-Design Prime Power based power optimization flow is recommended

Q&A


THANK YOU
Our Technology, Your Innovation™