# Maximizing Efficiency: Achieving Full Flat STA Run Overnight with Primetime Hypergrid

Anju KC, Rajiv Girdhar, Shardul Kishor Dixit
Intel

# Motivation

**Increasing Design Sizes**

**Longer run times associated with flat runs**

**Larger machines for flat runs**

**Management of multiple runs with hierarchical flow**

# Hypergrid- Distributed STA Feature with Primetime

### Distributed compute workers in Manager-Worker configuration

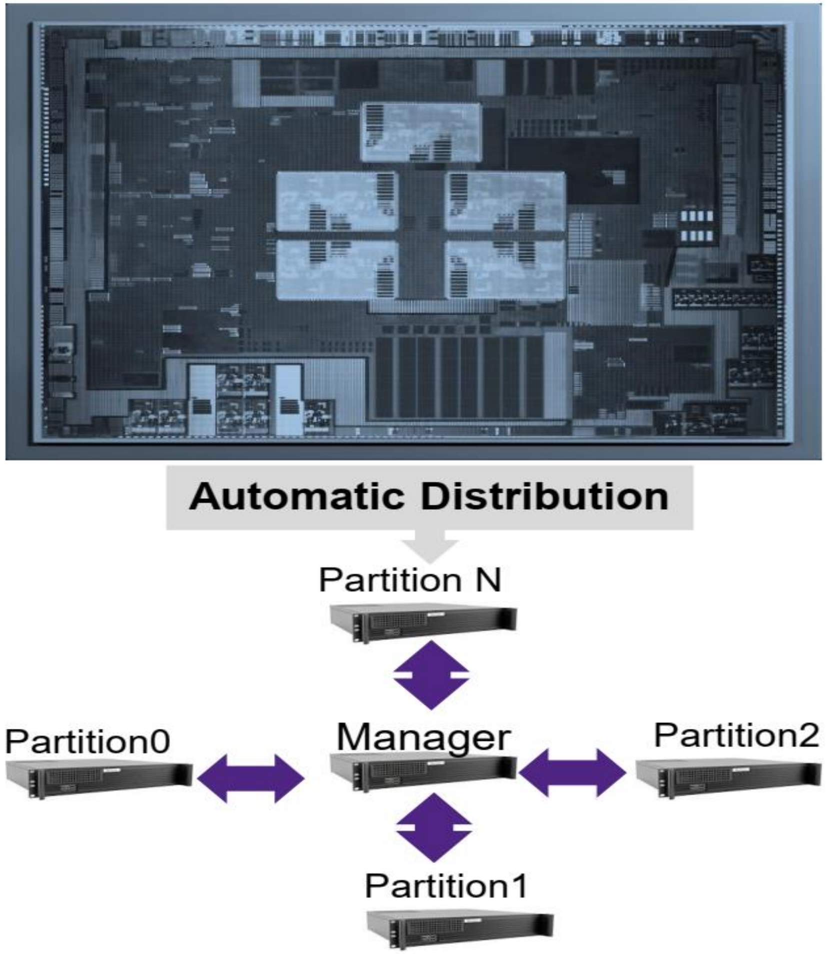
- Manager handles inter-process communication
- Workers/partitions analyzes part of design
- Manager-Worker interacts to keep the timing data up to date for the entire design

### Design is partitioned based on timing cones

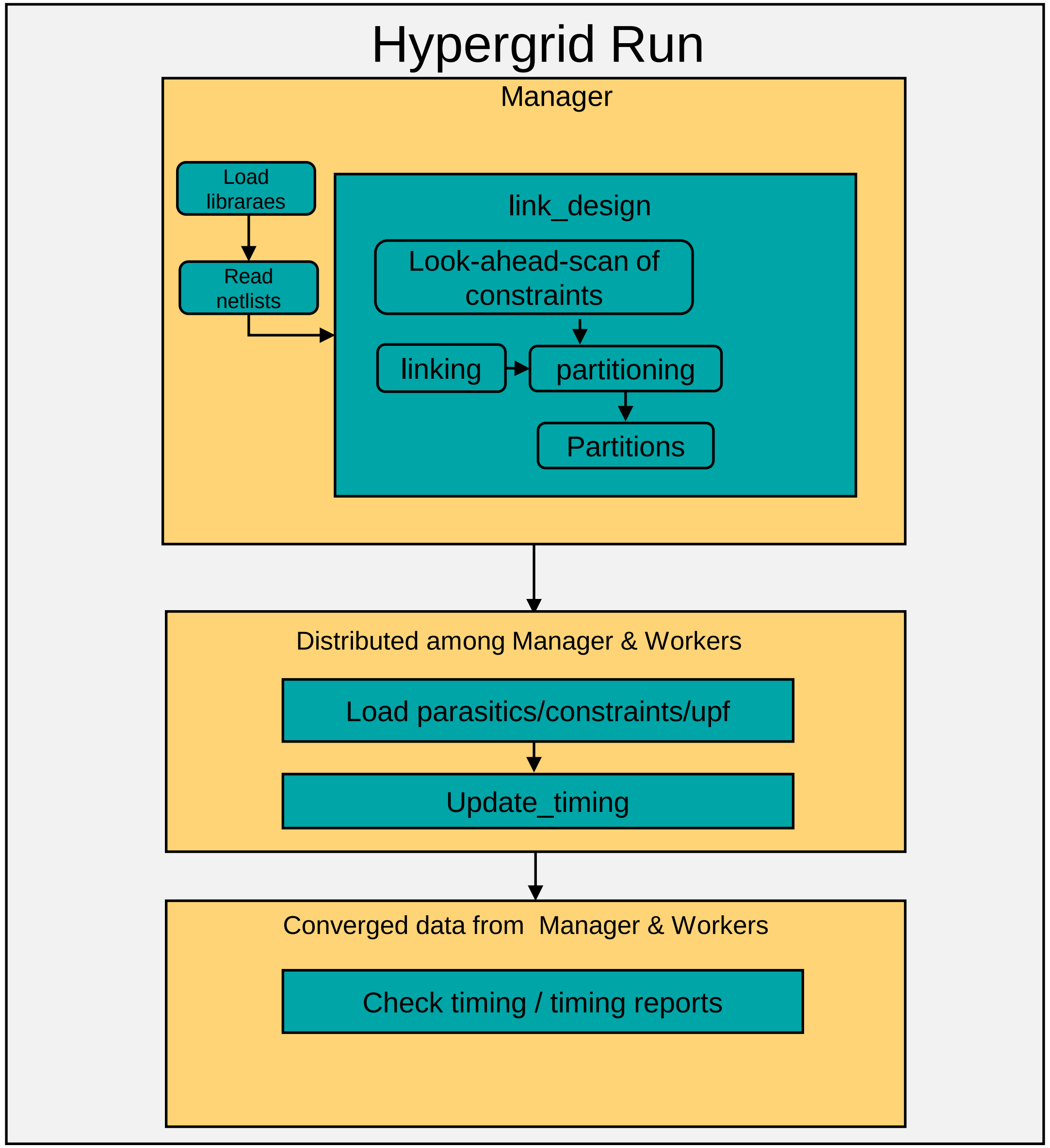
- Timing cones derived based on topological connectivity
- Constraints and Parasitics aware
- No dependency on physical hierarchies

### Better run time with lower configuration machines

- Utilizes multiple smaller configuration machines in parallel
- Significant improvement on run time
- No dependency on high configuration machines

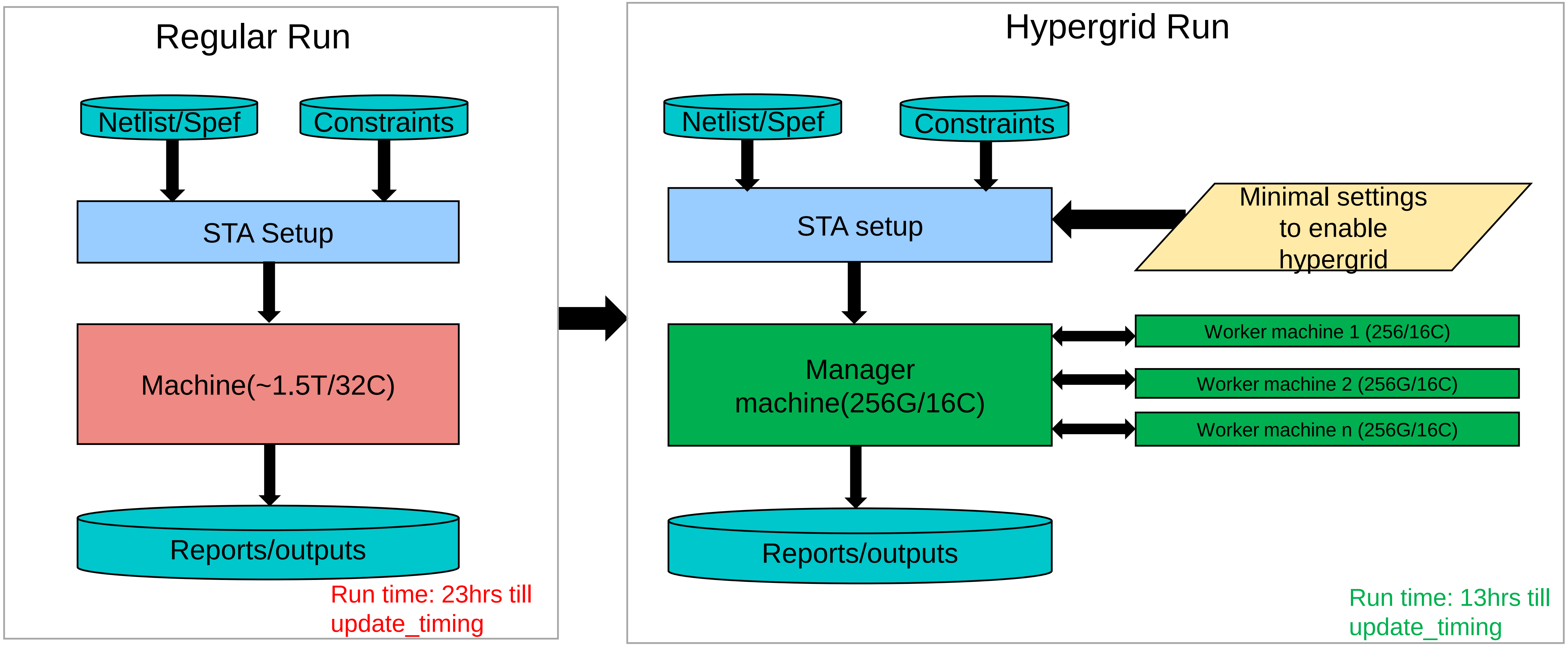


# Hypergrid Flow


Hypergrid Run
Manager
Load libraraes
Read netlists
link_design
Look-ahead-scan of constraints
linking
partitioning
Partitions
Distributed among Manager & Workers
Load parasitics/constraints/upf
Update_timing
Converged data from Manager & Workers
Check timing / timing reports

# Hypergrid Flow

| | |
|---|---|
| Design Size | 144M instances |
| No. of clocks | 834 |
| No of workers | 4 |
| PT Version | U-2022.12-SP5-2 |



Regular Run

Netlist/Spef

Constraints

STA Setup

Machine(~1.5T/32C)

Reports/outputs

Run time: 23hrs till update_timing

Hypergrid Run

Netlist/Spef

Constraints

STA setup

Minimal settings to enable hypergrid

Manager machine(256G/16C)

Worker machine 1 (256/16C)

Worker machine 2 (256G/16C)

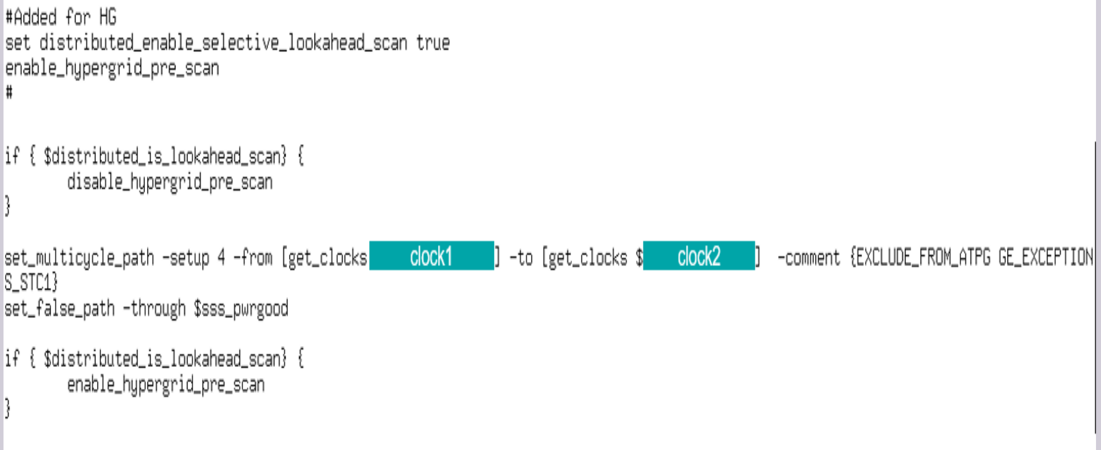
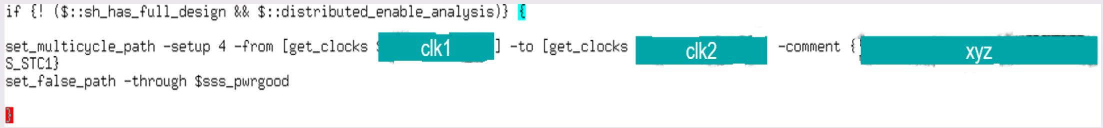
Worker machine n (256G/16C)

Reports/outputs

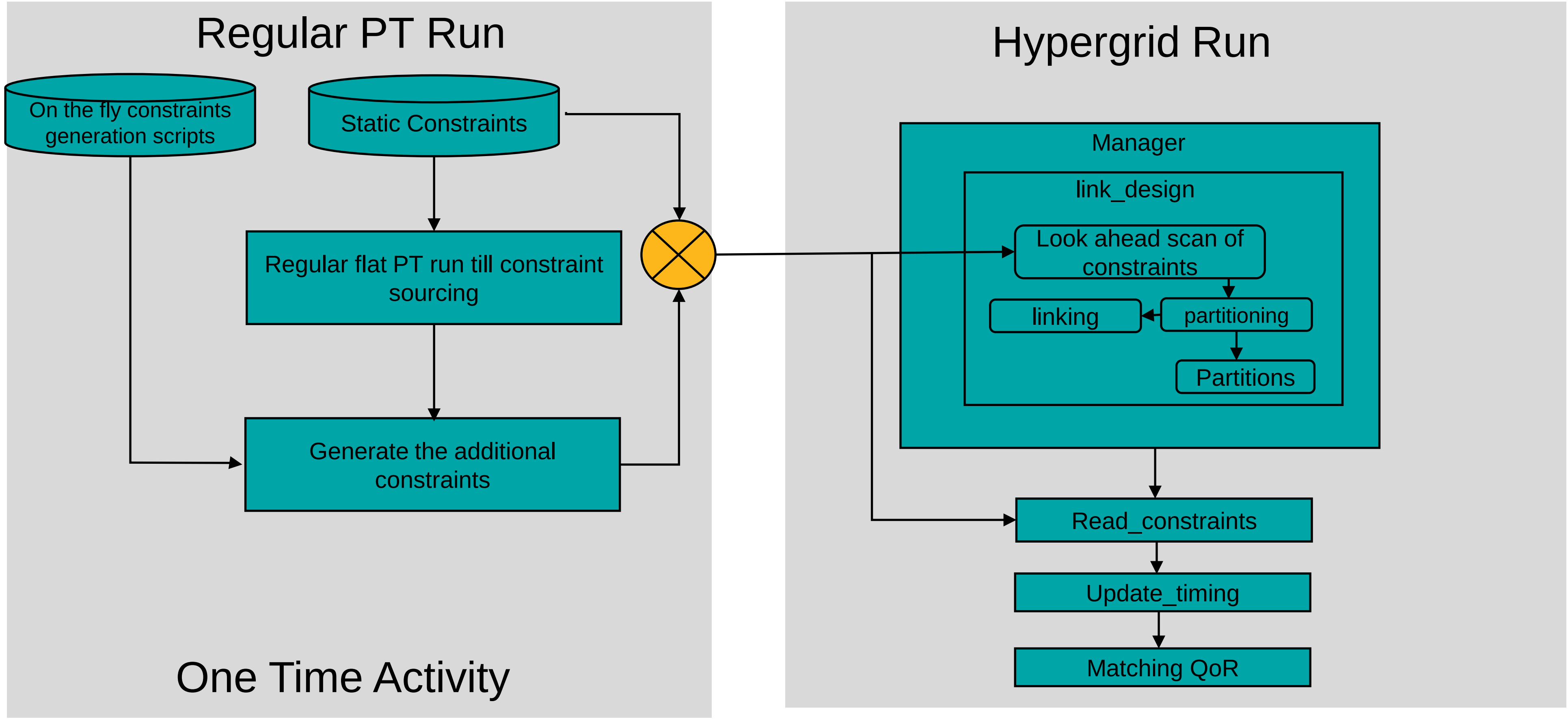
Run time: 13hrs till update_timing

Expectation: Run time improvement by utilizing lower memory machines

# Run Time Optimization

| Issues | Reason | BKM from SNPS | Update |
|---|---|---|---|
| Increase in link_design run time | HG Manager glances through the constraints to identify the clock propagation for proper partitioning of workers | Selective look-ahead-scan to avoid HG Manager wasting time on huge number of exceptions present | <br>#Added for HG<br>set distributed_enable_selective_lookahead_scan true<br>enable_hypergrid_pre_scan<br>#<br>if { $distributed_is_lookahead_scan} {<br>        disable_hypergrid_pre_scan<br>}<br>set_multicycle_path -setup 4 -from [get_clocks clock1 ] -to [get_clocks $ clock2 ]  -comment {EXCLUDE_FROM_ATPG GE_EXCEPTIONS_STC1}<br>set_false_path -through $sss_pwrgood<br>if { $distributed_is_lookahead_scan} {<br>        enable_hypergrid_pre_scan<br>} |
| Higher constraint sourcing run time at Manager | Sourcing of huge number of exceptions at the Manager results in higher run time | Selective disabling of exceptions which doesn't impact clock definition & propagation at the Manager | <br>if {! ($::sh_has_full_design && $::distributed_enable_analysis)} {<br>set_multicycle_path -setup 4 -from [get_clocks $ clk1 ] -to [get_clocks clk2 -comment { xyz<br>S_STC1}<br>set_false_path -through $sss_pwrgood<br>} |

# On-the-Fly Constraints Management for Hypergrid


Regular PT Run
On the fly constraints generation scripts
Static Constraints
Regular flat PT run till constraint sourcing
Generate the additional constraints
One Time Activity
Hypergrid Run
Manager
link_design
Look ahead scan of constraints
linking
partitioning
Partitions
Read_constraints
Update_timing
Matching QoR

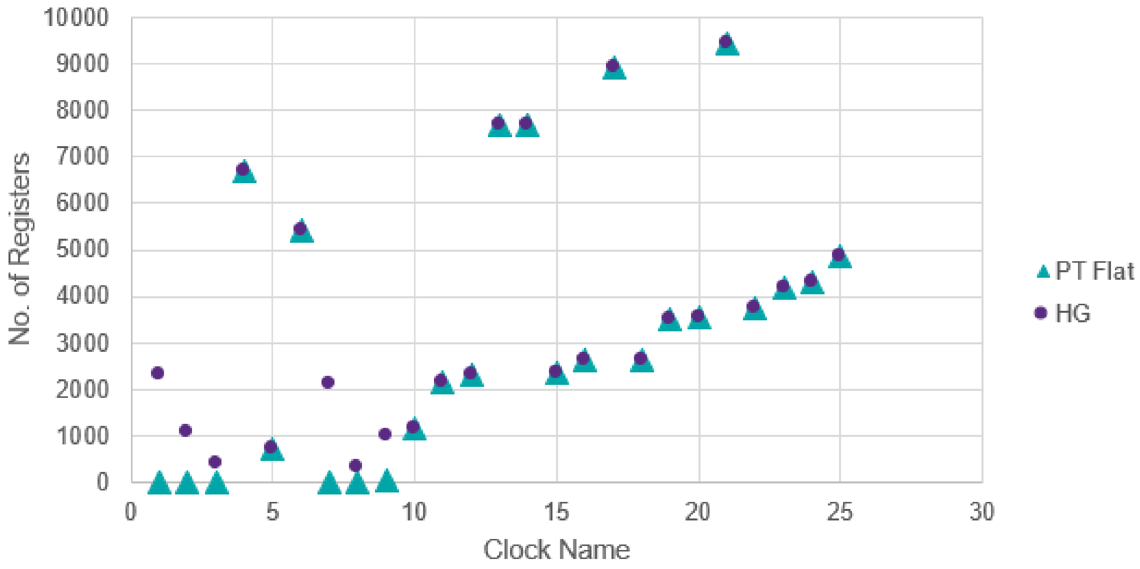
- All clock related constraints to be present during partitioning
- On-the-fly constraints to be converted to static first and then merged with other constraints for Hypergrid

# Quality Checks (Initial)

## Initial Run

| Check | Regular Run | HG Run | Comments |
|---|---|---|---|
| No Clocks | 70560 | 72679 | Delta is due to case_analyis getting excluded at the Manager causing high UCR in HG |
| Unconstrained Endpoints | 76884 | 76884 | Matching |
| Timing Loops | 17 | 25 | Delta is due to case_analyis getting excluded at the Manager |
| Parasitic Annotation(un-annotated) | 243 | 243 | Matching |
| Clocks | 835 | 834 | Un-used virtual clock is less in HG |
| all_registers | 12175580 | 12175580 | Matching |

Regular flat Vs Hypergrid

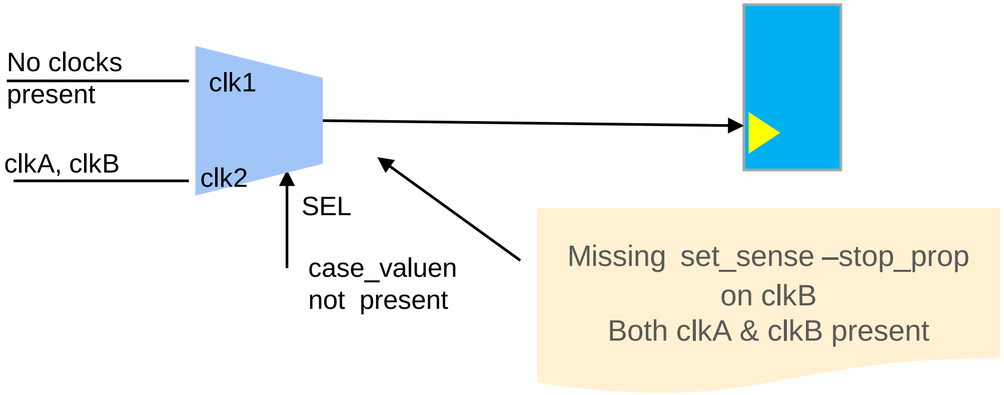


Clock Sinks Comparison

- Differences in quality checks during the initial runs due to clock propagation issues
- Few set_case_analysis and set_sense commands got excluded from manager while excluding exceptions from look ahead scan and exception exclusion

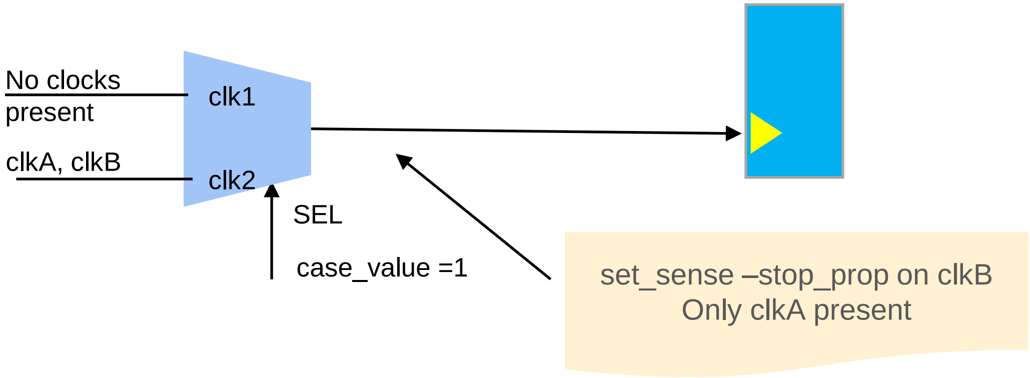
# Reason for High UCR

- Exclusion of some case analysis & clock stop propagation constraints in the manager impacted clock propagation



Missing set_sense –stop_prop on clkB
Both clkA & clkB present

- Missing case_analysis from the manager lead to no case value at the clock mux select pin
- clk1 input of the clock mux is reported as no clock
- Set_sense to stop the clock clkB was missing



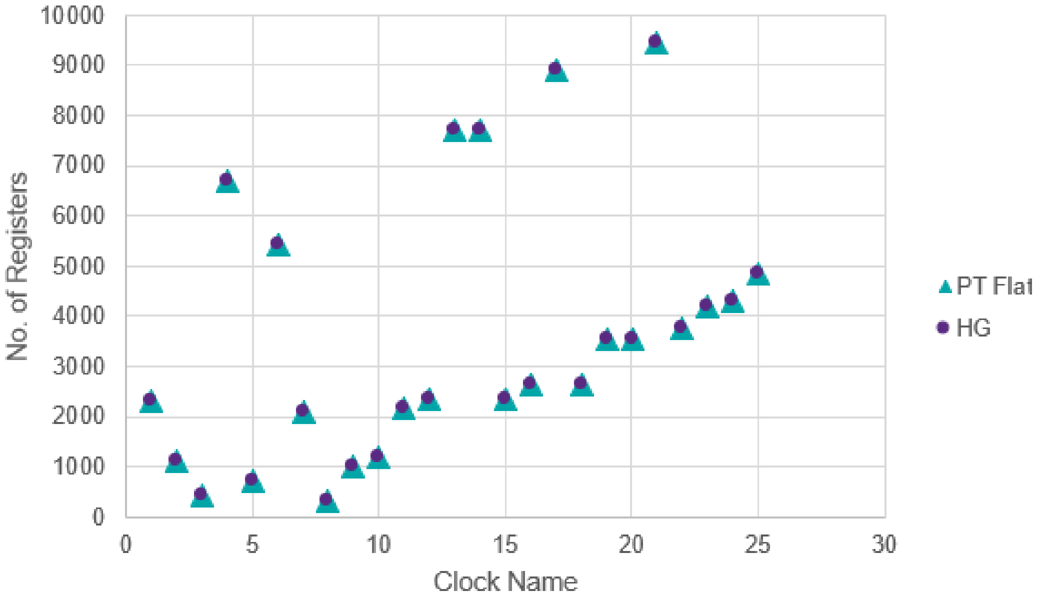
set_sense –stop_prop on clkB
Only clkA present

- No no-clock or UCE reported on the cell with the case value present
- Set_sense stops the unwanted clock

**Keep timing constraints modular to ease out exceptions exclusion from look-ahead-scan and Manager**

# Quality Checks (Final)

| check_timing | PT | HG | comment |
|---|---|---|---|
| no_input_delay | 0 | 0 | Matching |
| no_driving_cell | 43375 | 43375 | Matching |
| unconstrained_endpoints | 76696 | 76696 | Matching |
| unexpandable_clocks | 2 | 2 | Matching |
| no_clock | 70560 | 70560 | Matching |
| loops | 19 | 19 | Matching |
| generated_clocks | 5 | 5 | PTE-103 & PTE-075 |
| voltage_level | 613 | 613 | Matching |
| Clocks | 835 | 834 | Un-used virtual clock is less in HG |
| all_registers | 12175580 | 12175580 | Matching |



Clock Sinks Comparison

- Check_timing matches with regular flat runs
- The run also passed all other extensive internal quality checks

# Timing Correlation

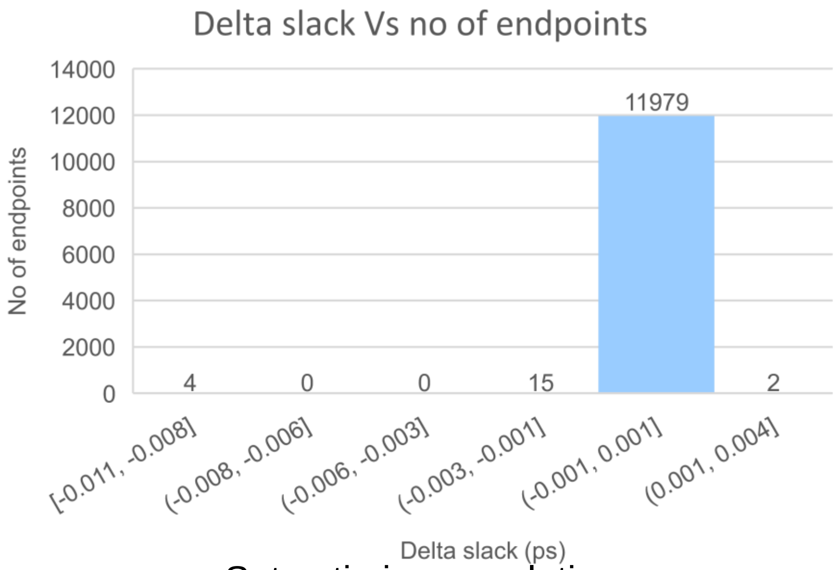
## Regular Flat Vs Hypergrid

| SETUP | | | |
|---|---|---|---|
| | Flat | HG | Diff |
| WNS (ps) | -14410 | -14410 | 0 |
| TNS (ps) | -262953910 | -262958579 | 4668 |
| NUM | 1329545 | 1329597 | 52 |
| TNS per endpoint | 197.8 | 197.8 | 0.004 |

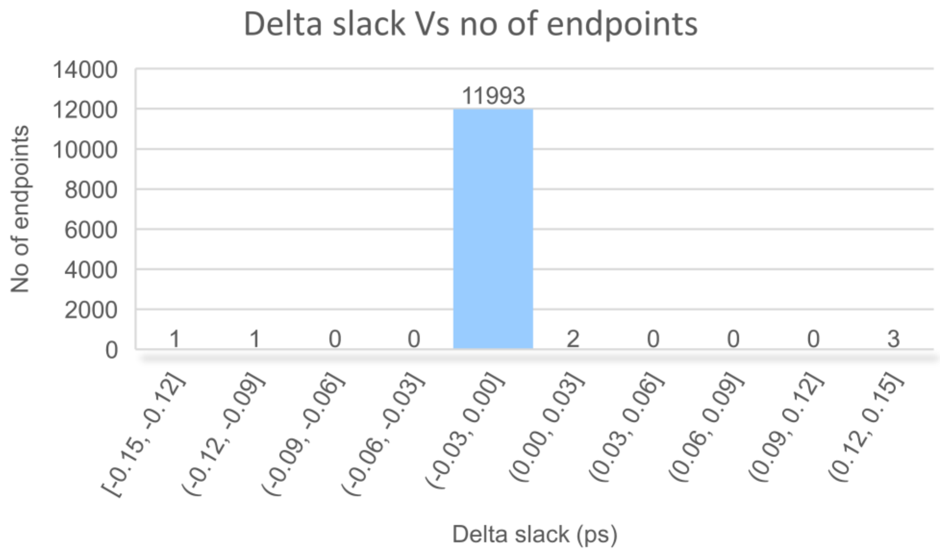
Setup Timing QoR (PBA)

| HOLD | | | |
|---|---|---|---|
| | Flat | HG | Diff |
| WNS (ps) | -8052 | -8052 | 0 |
| TNS (ps) | -300606 | -300596 | 10 |
| NUM | 29324 | 29325 | 1 |
| TNS per endpoint | -10.251 | -10.251 | 0.001 |

Hold Timing QoR (PBA)



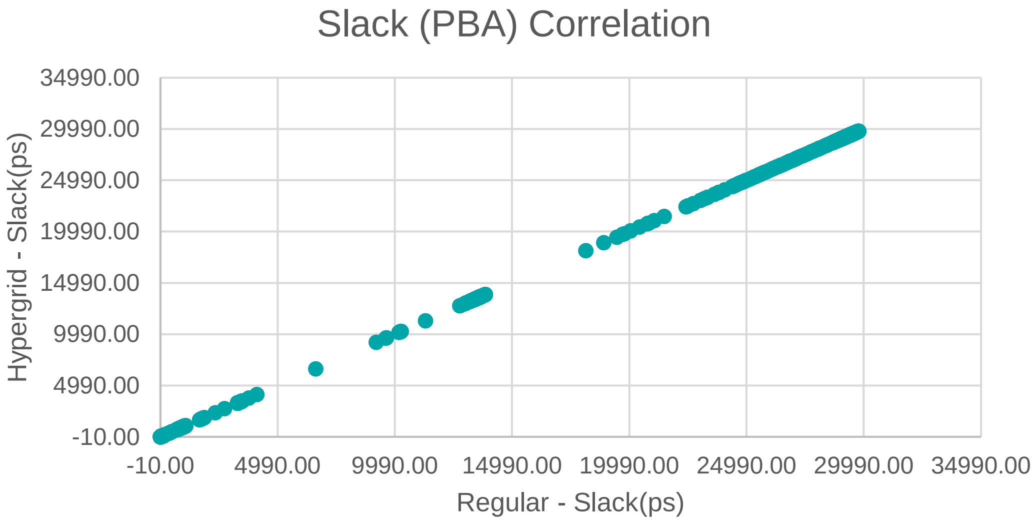
Setup timing correlation

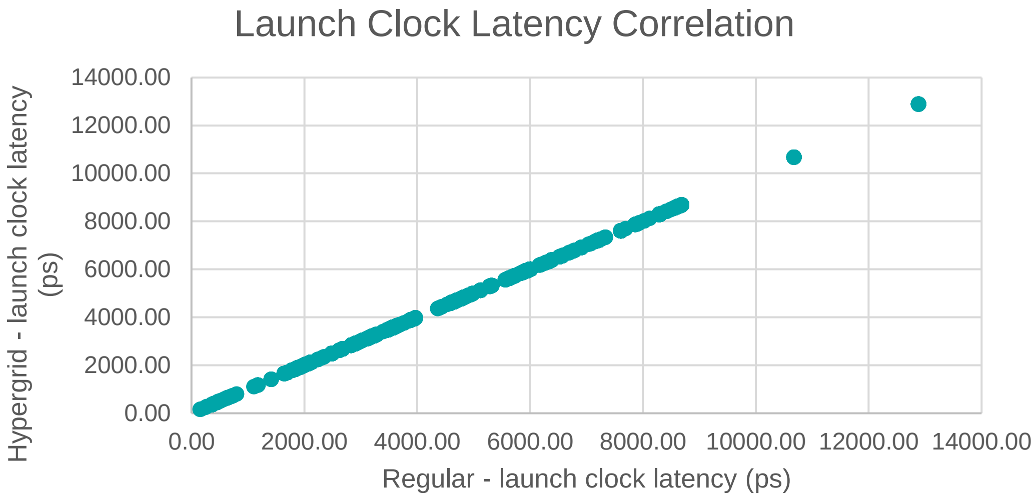


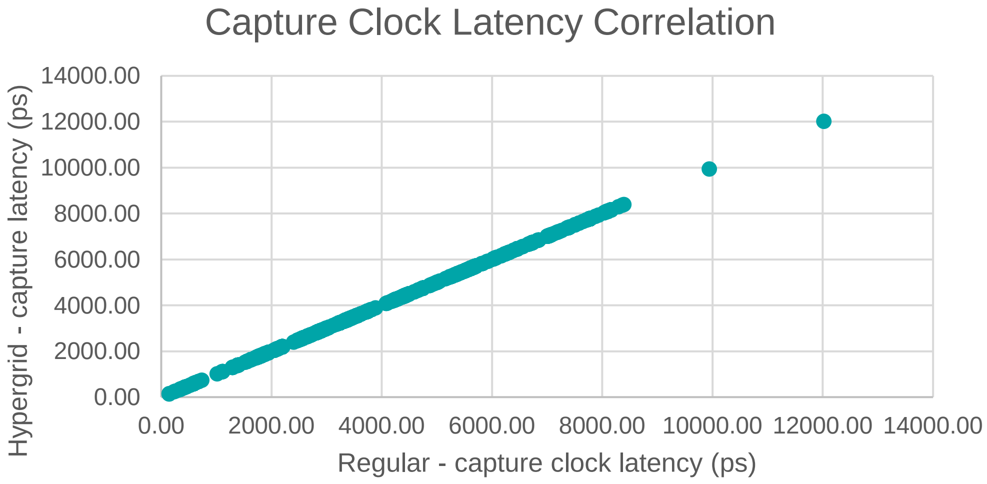
Hold timing correlation

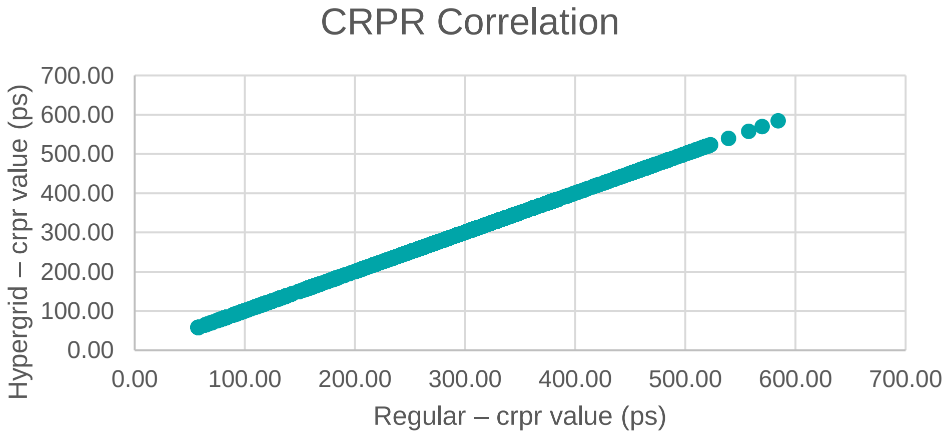
# Timing Correlation

## Regular Flat Vs Hypergrid







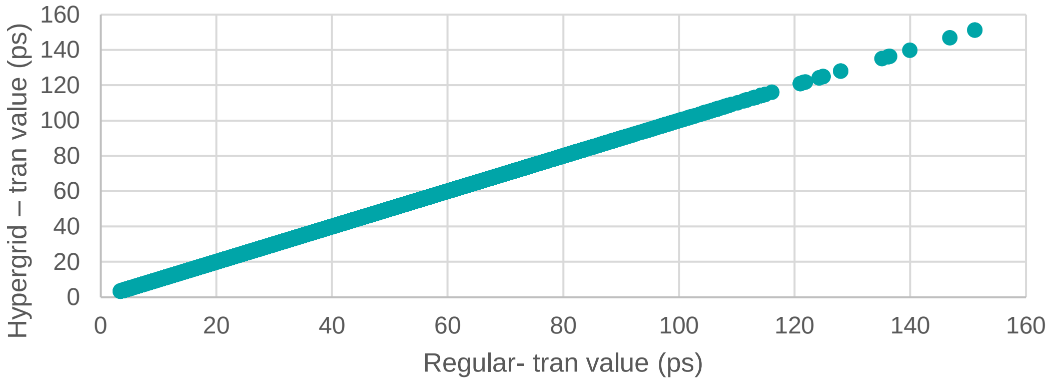


Correlation done for 12K endpoints

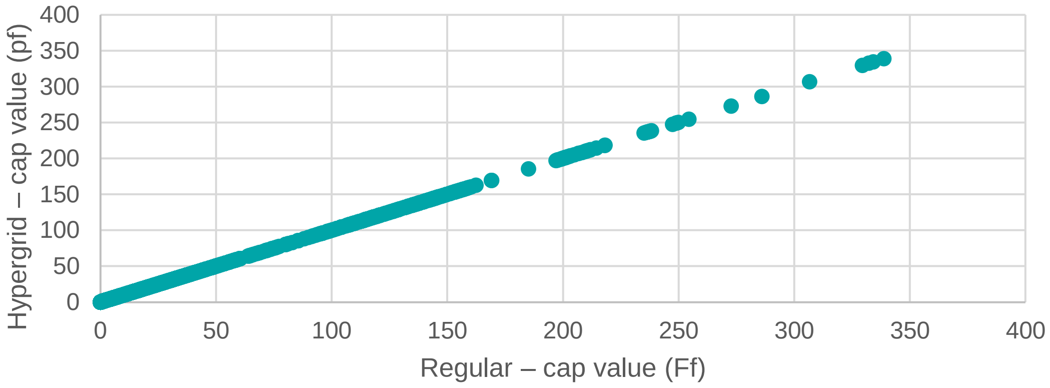
# Timing Correlation

## Regular Flat Vs Hypergrid

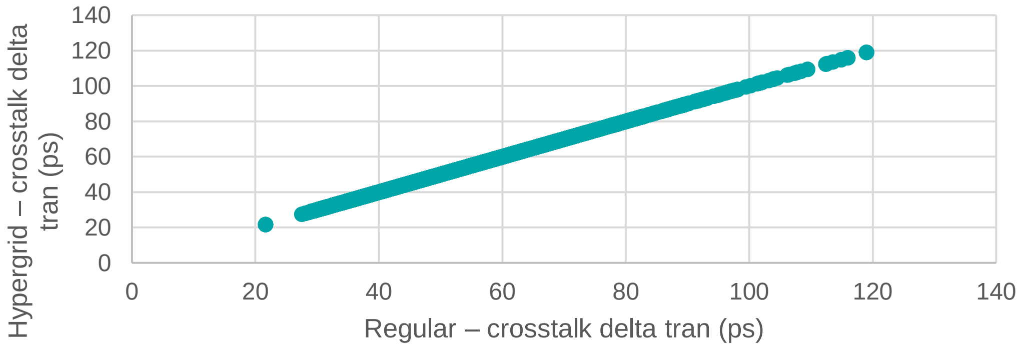
Transition Correlation



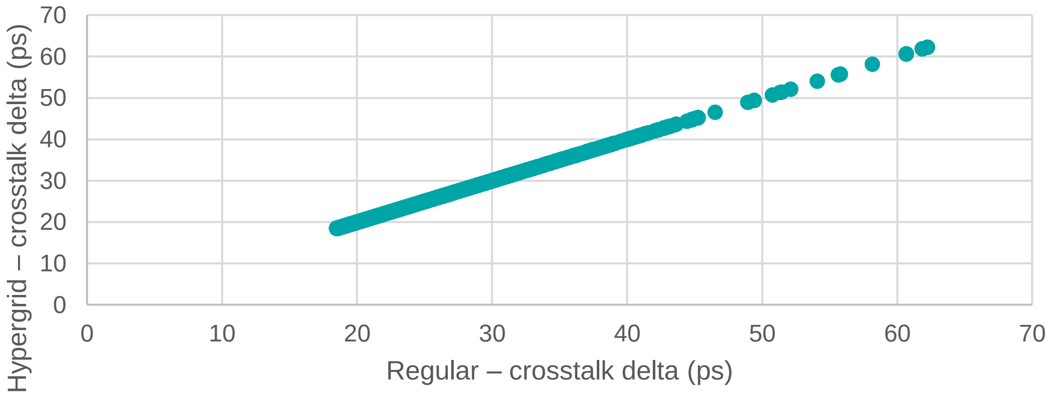
Capacitance Correlation



Crosstalk delta-tran Correlation



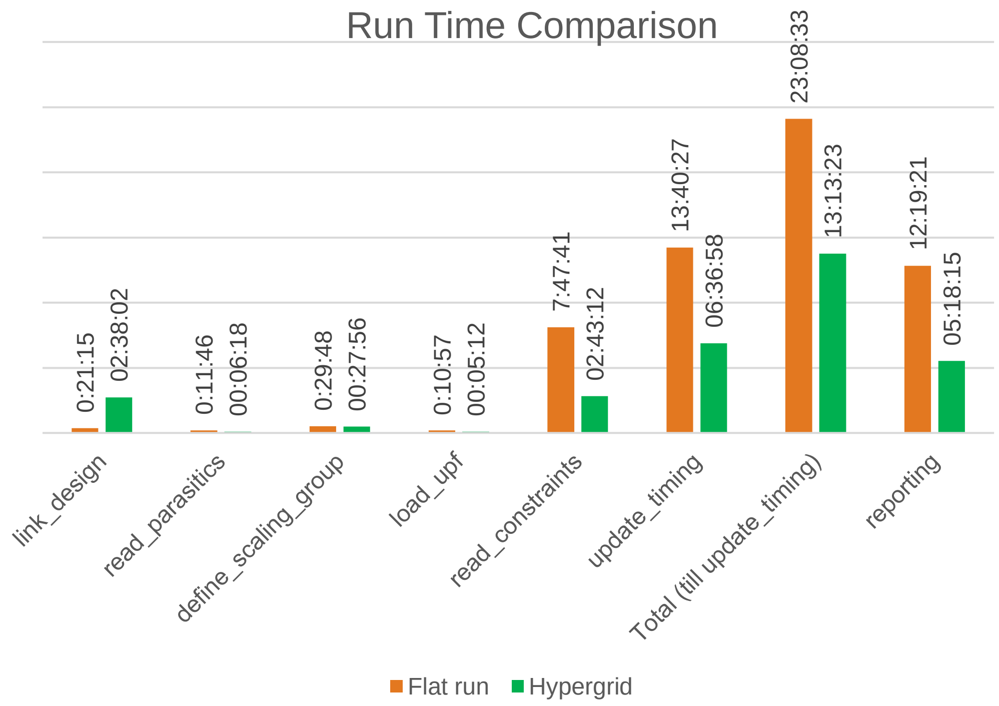
Crosstalk Correlation

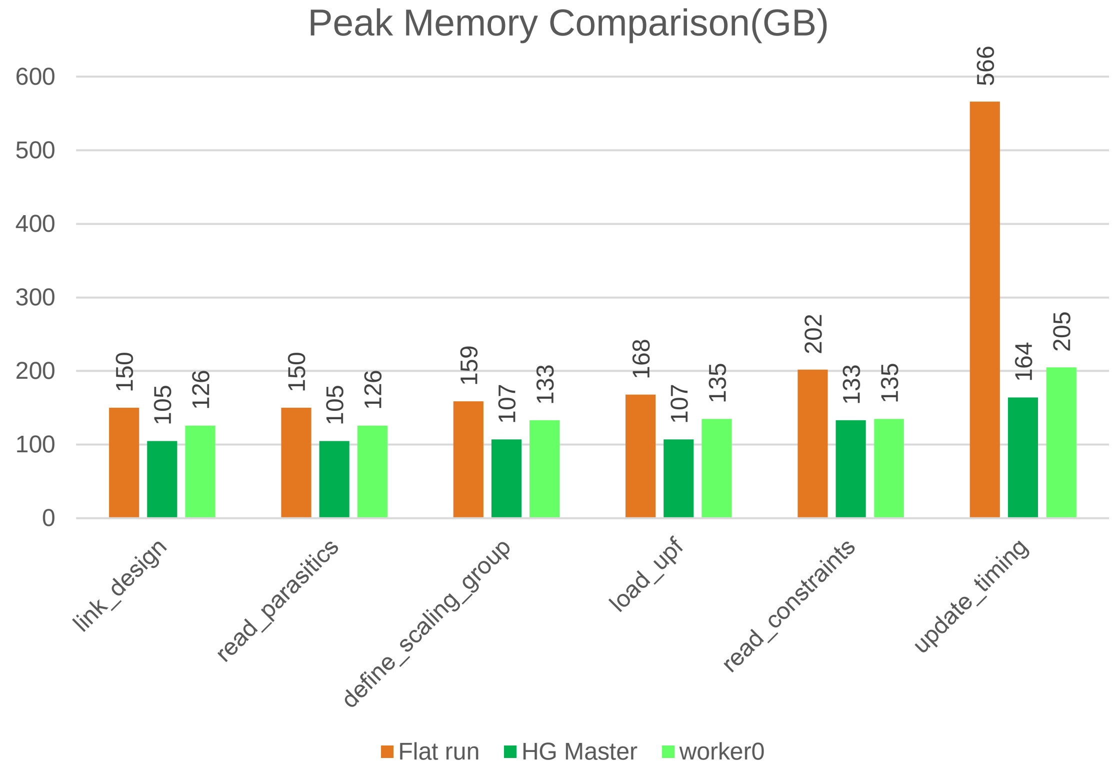


Correlation done for 12K nodes

# Run Time / Memory Comparison

## Regular Flat Vs Hypergrid



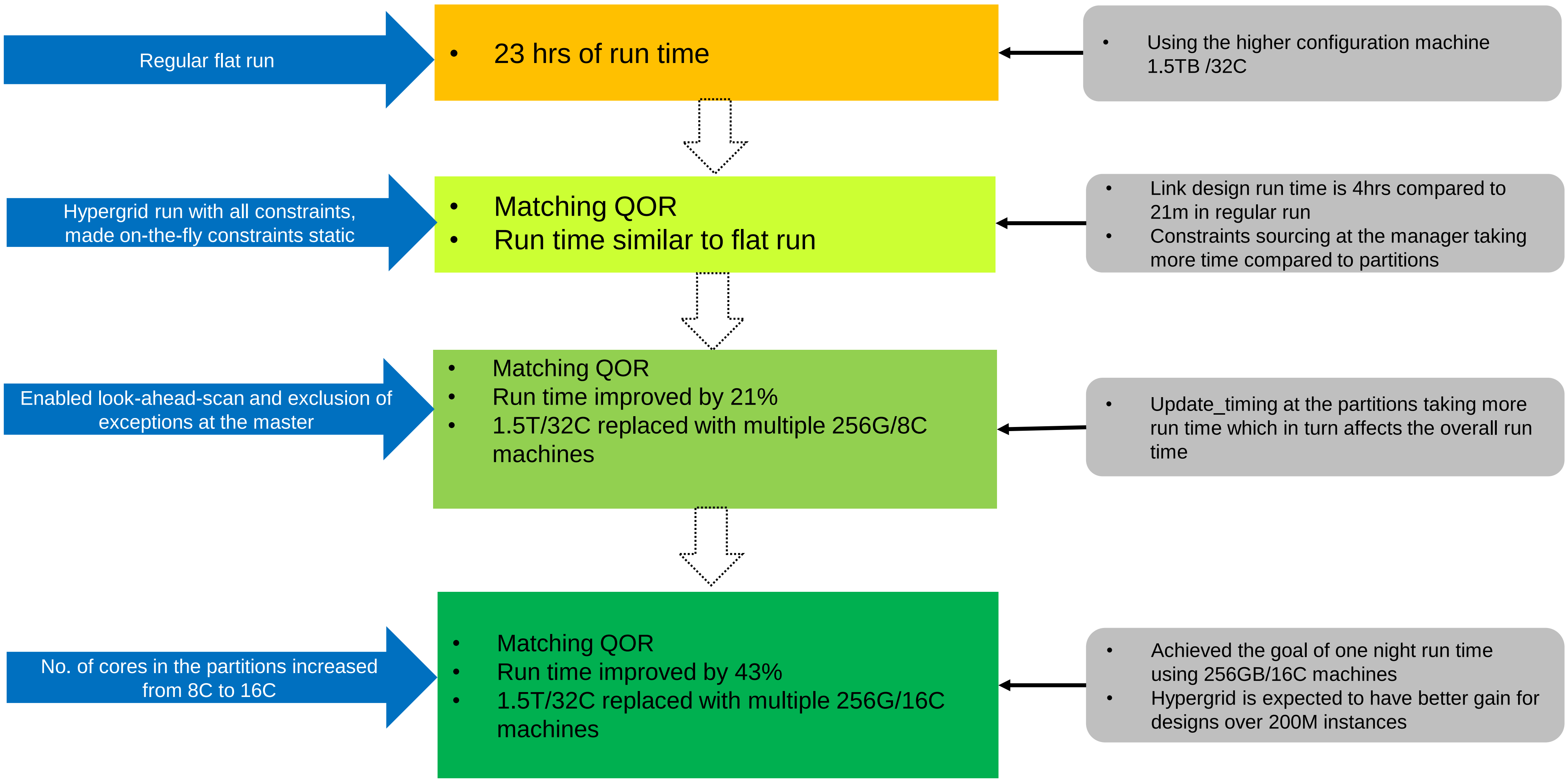


- Considerable improvement in run time of constraint reading & update timing stage
- Overall Hypergrid run time till update timing reduced from 23hrs to 13hrs
- Run time improved by ~43% than flat run time
- Peak memory usage reduced to < 256GB with 16C compared to >500GB with 32C in flat runs

# Current Limitations & Future Scope

- On-the-fly constraint generation
  - All constraints including clock definitions and others affecting clock propagation must be present during link_design while partitioning occurs
  - **Tool can be enhanced to handle this automatically**
- Exceptions (FP & MCP) to be manually excluded from manager
  - To be excluded from look-ahead-scan and constraints sourcing to achieve an optimal run time
  - Adds extra overhead in managing the constraints effectively
  - **Tool can be enhanced to handle the exception exclusion automatically**
- Minor reporting differences observed on few PARA, RC & UITE warnings
  - Debugged to be reporting issues
  - **To be fixed in tool**
- Evaluate quality of the outputs e.g. sdf, context, hyperscale etc.

# The Hypergrid Journey



Regular flat run

- 23 hrs of run time

- Using the higher configuration machine 1.5TB /32C

Hypergrid run with all constraints, made on-the-fly constraints static

- Matching QOR
- Run time similar to flat run

- Link design run time is 4hrs compared to 21m in regular run
- Constraints sourcing at the manager taking more time compared to partitions

Enabled look-ahead-scan and exclusion of exceptions at the master

- Matching QOR
- Run time improved by 21%
- 1.5T/32C replaced with multiple 256G/8C machines

- Update_timing at the partitions taking more run time which in turn affects the overall run time

No. of cores in the partitions increased from 8C to 16C

- Matching QOR
- Run time improved by 43%
- 1.5T/32C replaced with multiple 256G/16C machines

- Achieved the goal of one night run time using 256GB/16C machines
- Hypergrid is expected to have better gain for designs over 200M instances

# Timing Model Comparisons

| Parameter | Flat Timing Model | Hierarchical Timing Model | HyperGrid based Timing Model |
|---|---|---|---|
| Timing Signoff | Only at SoC level | At multiple levels | Only at SoC level |
| Signoff Accuracy | High | Needs more efforts to establish correlation | High |
| Constraints Generation, Management & Signoff | Only at SoC level | At multiple levels | Only at SoC level |
| Xtalk & DCD impact modeling | Not Required | Required | Not required |
| Run Time | High | Low | Highly Optimized |
| Memory Requirement | High | Low | Low |
| HyperScale/Lib Generation & Management | Not required | Needs more efforts, Required at various level | Not required |
| Quality Check | Only at SoC level | At multiple levels | Only at SoC level |

# Summary

- Achieved over night run time for SoC level flat timing runs with HyperGrid
- Reduces dependency on limited high configuration machines which reduces overall wait time
- HyperGrid based timing run QoR matches with regular flat timing runs.

snug

# *THANK YOU*

Our Technology,
**Your Innovation**™