Tempus Block Scope for Faster Timing Signoff

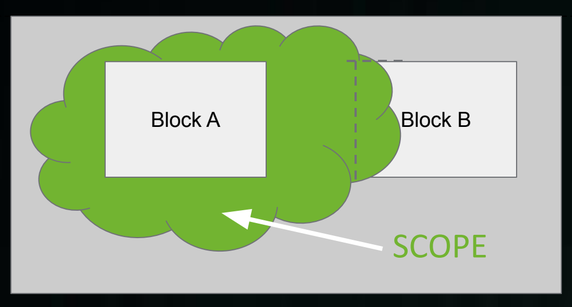
averasemi

Tim Helvey, Parth Lakhiya,
Jay Gaudani, Tim Sterczyk

What is Block Scope?

Block scope is a timing abstract created from full chip timing for use by the block

Scope contains a verilog netlist of all paths touching the block boundary and all the necessary timing information to see an accurate timing analysis for the block



Why Scope?

Big Chip Challenges

1. Block owners need to know what to work on
2. Making sign-off accurate boundary constraints is difficult (impossible)
   a. Constraint differences
   b. Inaccuracy across corners
   c. Stage counting or statistical differences
3. Inaccuracy can lead to over/under fixing
4. Relying on full chip timing analysis is expensive (IT and schedule)

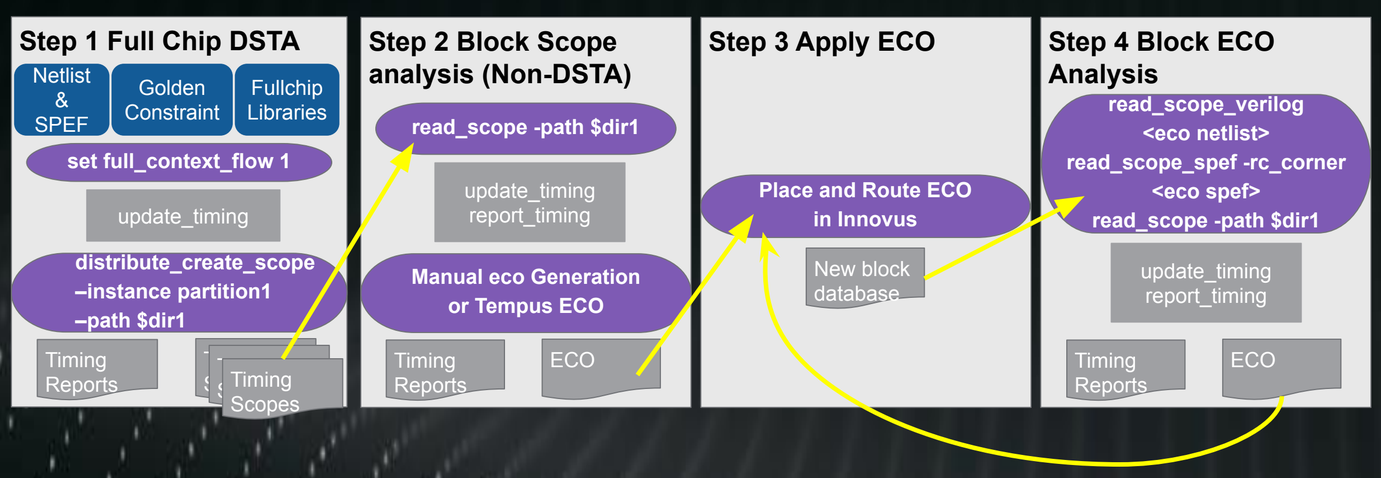
We need an accurate environment for block owners to test their timing fixes quickly

Scope Advantages

1. Block owners see what they need
2. Accurate boundary path information
3. Signoff quality modeling
4. Reduced dependency on top level run

Scope provides an accurate environment for block owners to test their timing fixes quickly

Using Block Scope

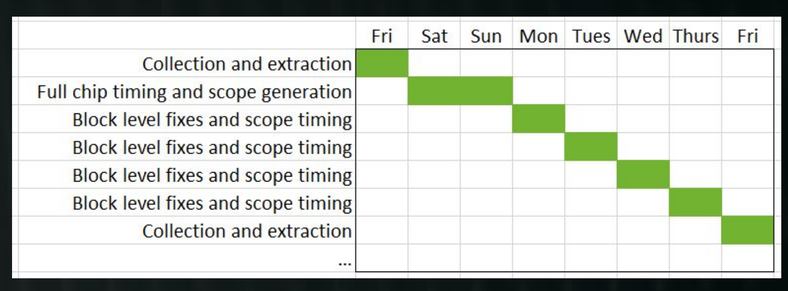


Flow

Full chip timing generates “scope” for block and chiplet environments

Block and chiplets iterate repeatedly in that environment

7 days for 1 full chip and 4 block scope iterations



Design Example 1

Die size : “very big”
Total unique blocks : ~100
Levels of Hierarchy: 3
Total sub-partitions : ~5

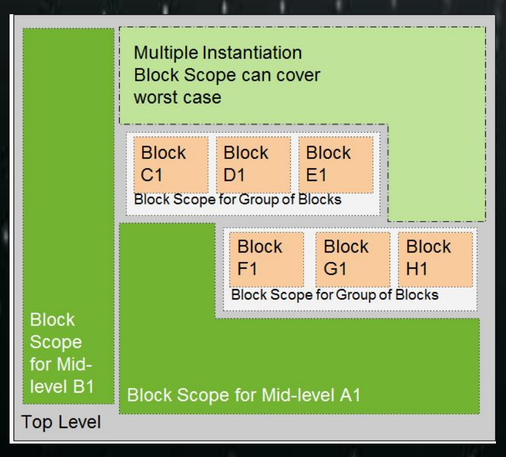


Image is for representation purposes only.

blocks = 1-4M instance

mid-level = 10-13M instances

scope = 20-70M instances

Generated scope for mid- and leaf- levels

Multiple blocks in one scope

Design Example 2

Die size : “very big”
Total unique blocks : ~100
Levels of Hierarchy: 2
Reuse of a unique block : >10

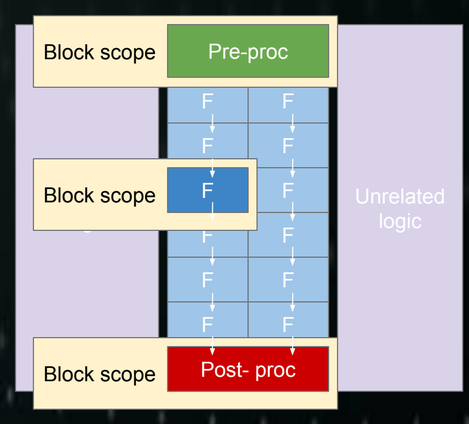


Image is for representation purposes only.

“F” blocks talk to themselves

Scope for all instances is too big

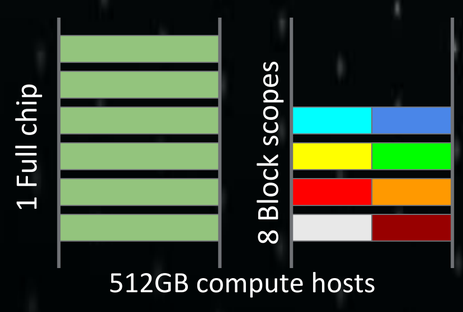
- We need to use one representative instance
- Pre- and Post- interfaces are unique and need to be handled by those blocks

For F to F paths, we need to choose to work on either the input side or output side, but not both

Runtime and Memory

Full chip (distributed): 3TB, 8 hours / view

Block with scope: 250GB, 8 hours / view



Results

More iterations per week

Faster closure

Reduction in IT demand peaks

Summary - challenges

Not currently supported in the P&R tool

Large amounts of reuse

Multi-instance module talking to itself

Long spef reading for MiM

Scope generation time

Scope gets stale due to ECOs

Scope data volume >30GB*

19.1 Improved but not verified

Summary - Advantages

Eliminates SDC inaccuracy in block level runs

Reduced dependency

Improved turnaround time

Reduced IT demand

Reduced need for constraint budgets

Support for Tempus ECO optimization

No requirement to align SDC between block and top