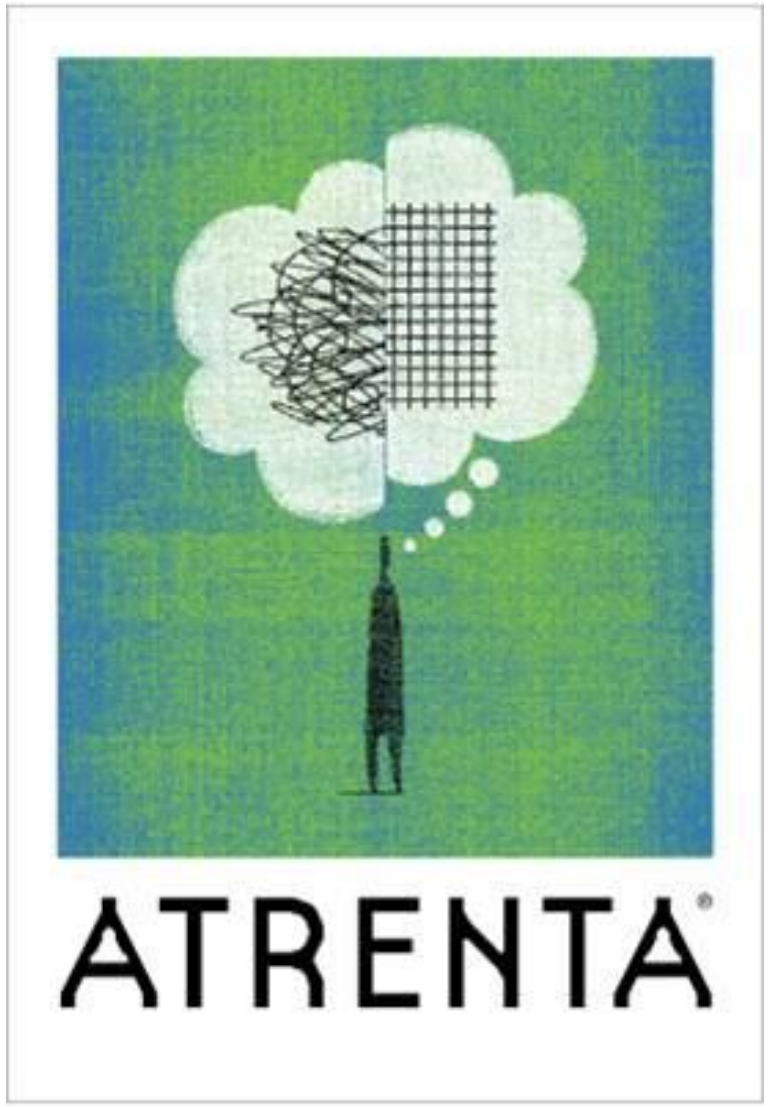


SpyGlass®-CDC Training

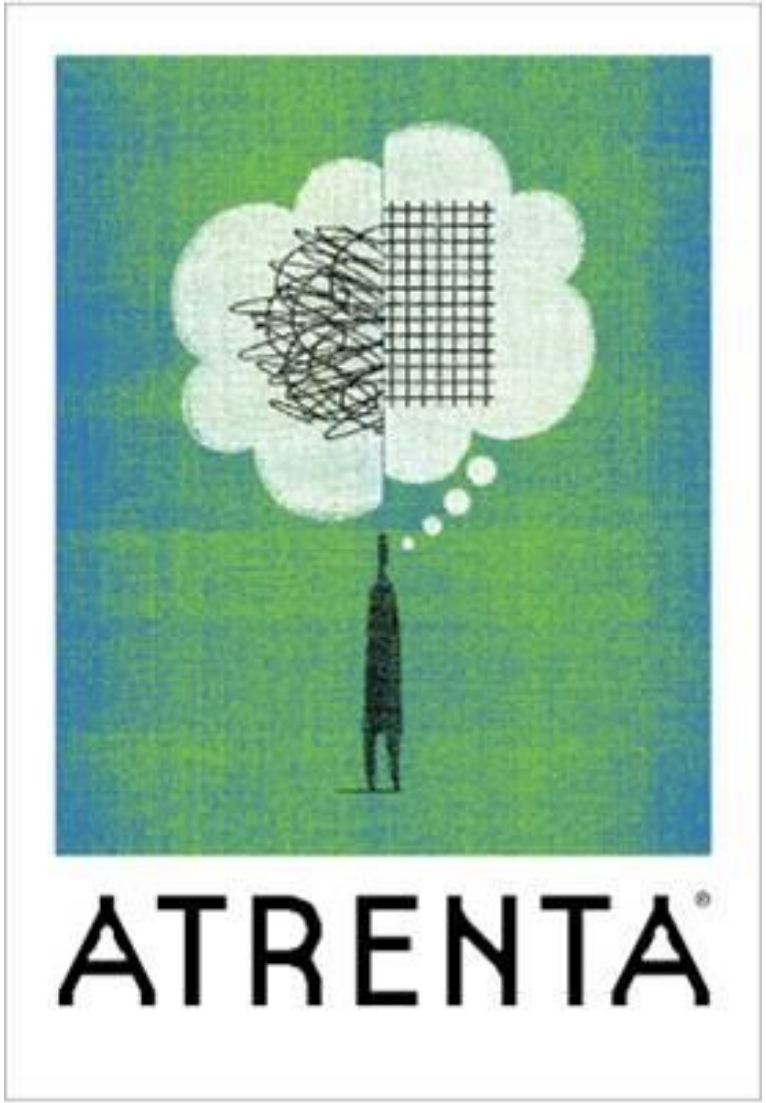
THINK EARLY DESIGN CLOSURE THINK ATRENTA®

Atrenta Confidential © 2008 Atrenta Inc.

Agenda

- **Overview of Atrenta SpyGlass Platform and SpyGlass Environment**
  - Lab 1: Getting familiar with SpyGlass
- **Introduction to the CDC issues**
- **SpyGlass-CDC flow and methodology**
- **Setting up SpyGlass-CDC using CDC Setup Manager**
  - Lab 2: Setup for CDC Verification
- **Checking Clock/Reset Integrity**
  - Lab 3: Clock/Reset Integrity check
- **Running SpyGlass-CDC analysis**
  - Lab 4: Identifying and correcting for Metastability Issues in the Design
- **Functional Verification of synchronization structures and other CDC rules**
  - Lab 5: Identifying advanced CDC issues
- **Appendix**
- **Where to look for more**

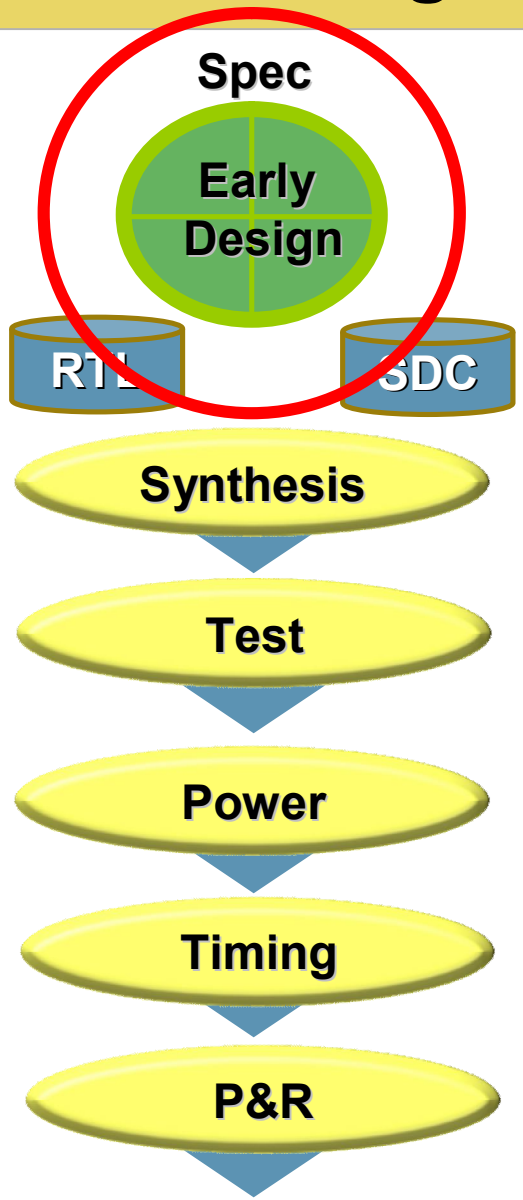
Atrenta Confidential © 2008 Atrenta Inc.



SpyGlass Platform Overview

Atrenta Confidential © 2008 Atrenta Inc.

The SoC Design Problem

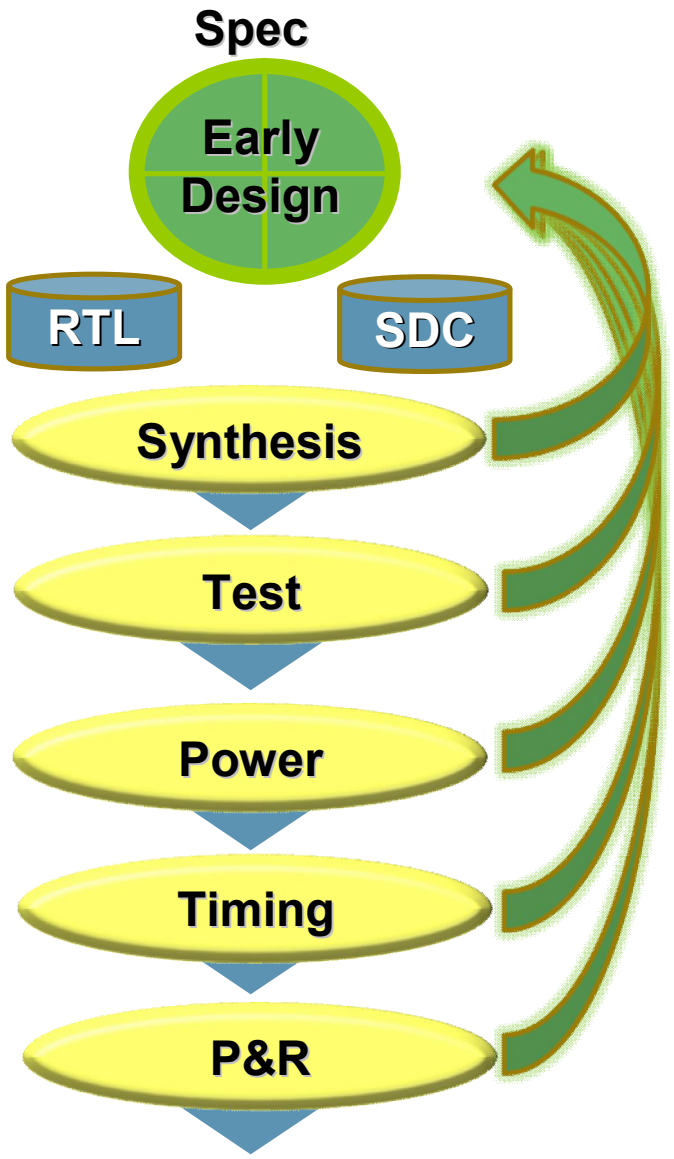


Design created with little feedback/visibility

- RTL – SDC conflicts
- Simulation – Synthesis mismatch
- Low fault coverage
- Power out of budget
- Can't meet timing
- Routing congestion, will not fit die

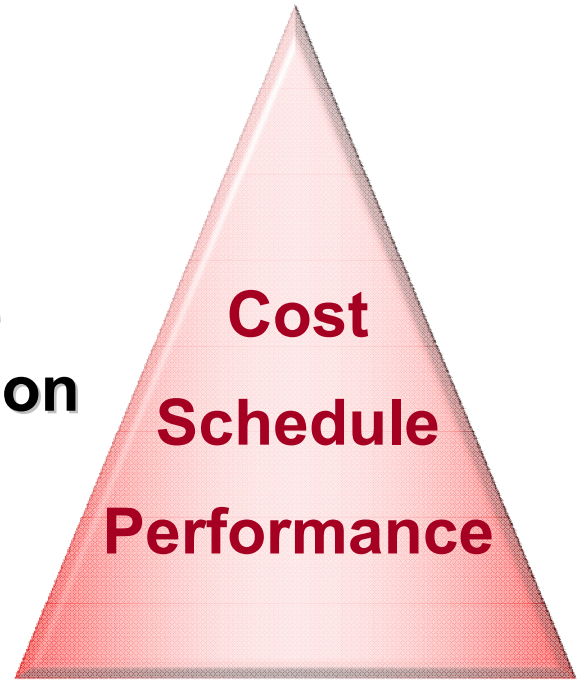
Atrenta Confidential © 2008 Atrenta Inc.

The Current Design Process



Iterative Correction

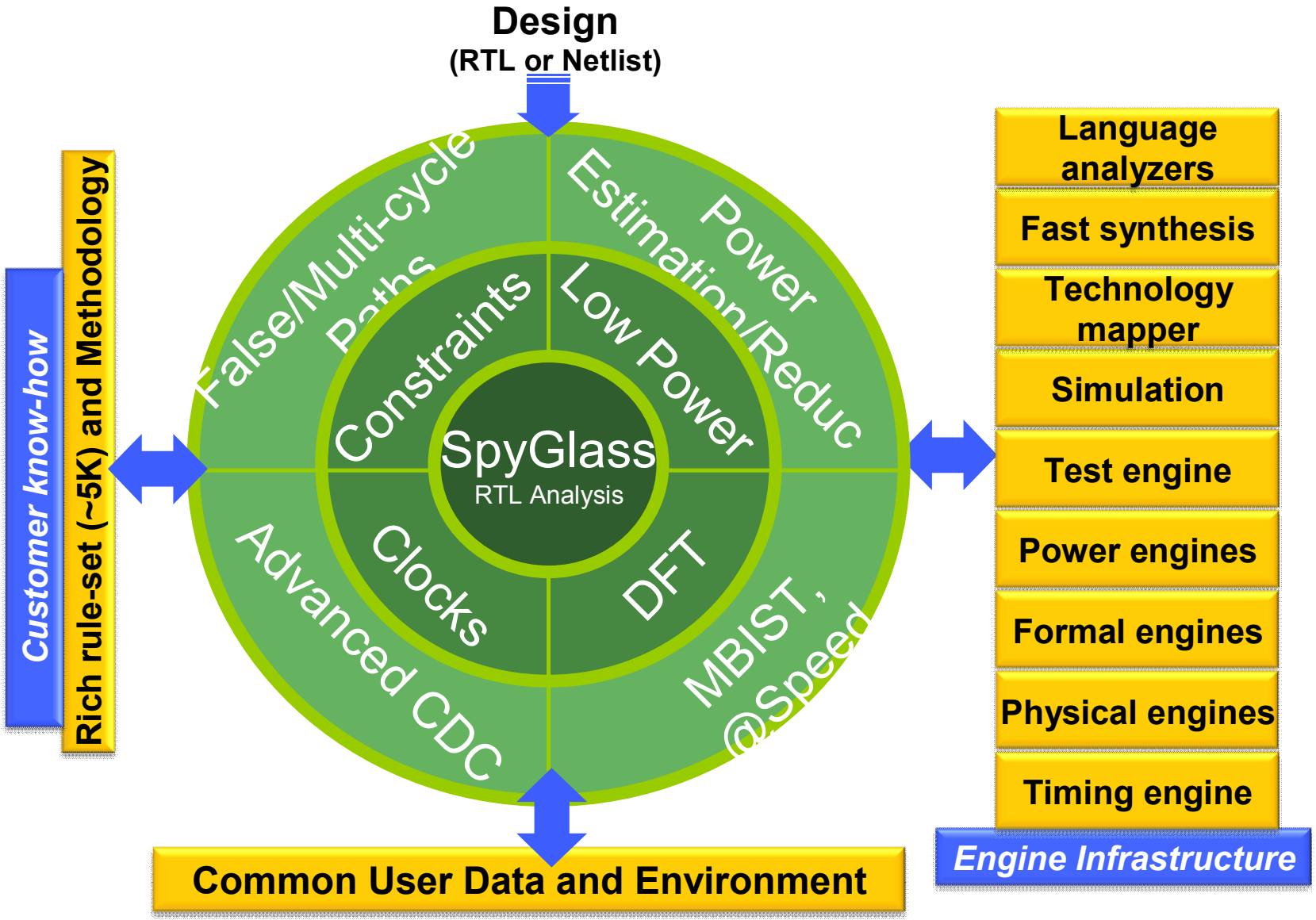
Long Iterations Result In Negative Impact On:



Divergent Process…
Getting Worse as Technology Advances

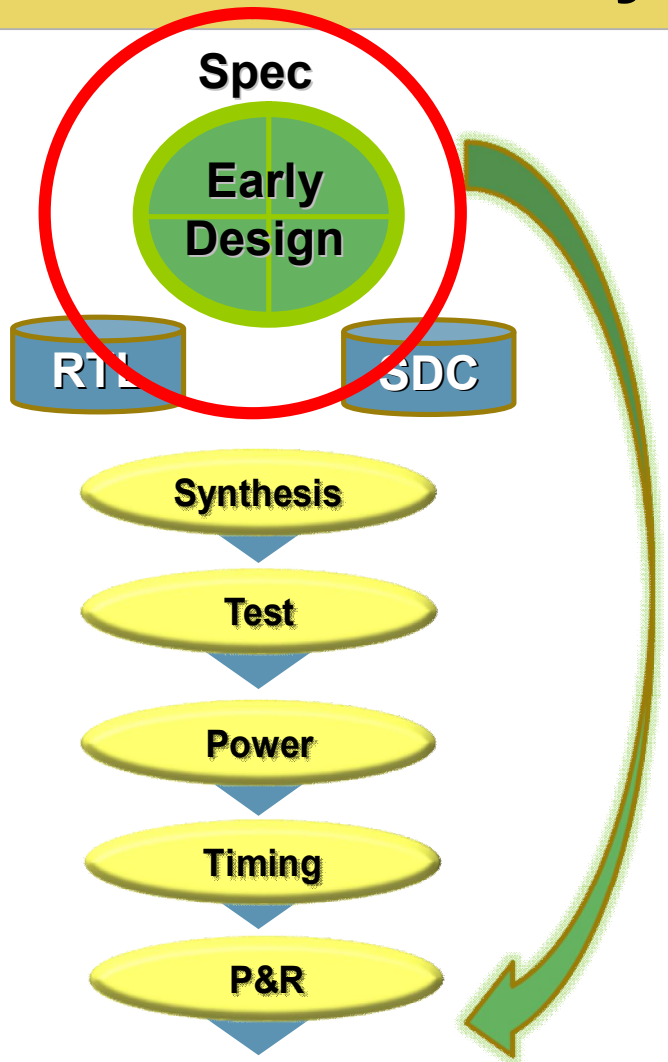
Atrenta Confidential © 2008 Atrenta Inc.

SpyGlass - Analyze & Optimize Design Early



Atrenta Confidential © 2008 Atrenta Inc.

The Power of Early Design Closure®



Recognize Issues at RTL
Fix and Optimize

Higher Productivity, Predictability

Lower Cost, Risk

Convergent Process
Faster Time to Market

Atrenta Confidential © 2008 Atrenta Inc.

GuideWare – A Structured Methodology for SoC Design

Architecture and Micro Architecture

Block Budget and Targets/Constraints

VERIFICATION

Field of Use-1 – New-RTL

Initial RTL	Detailed RTL	RTL Handoff

Field of Use-2 – IP Reuse

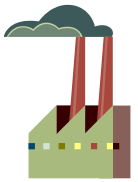
IP Exploration	Inherent Risks	Adaptation Risks

IP Development **IP Acceptance**

Field of Use-3 – SoC

RTL Integ	Netlist Integ	Pre Layout	Post Layout

SoC Integration

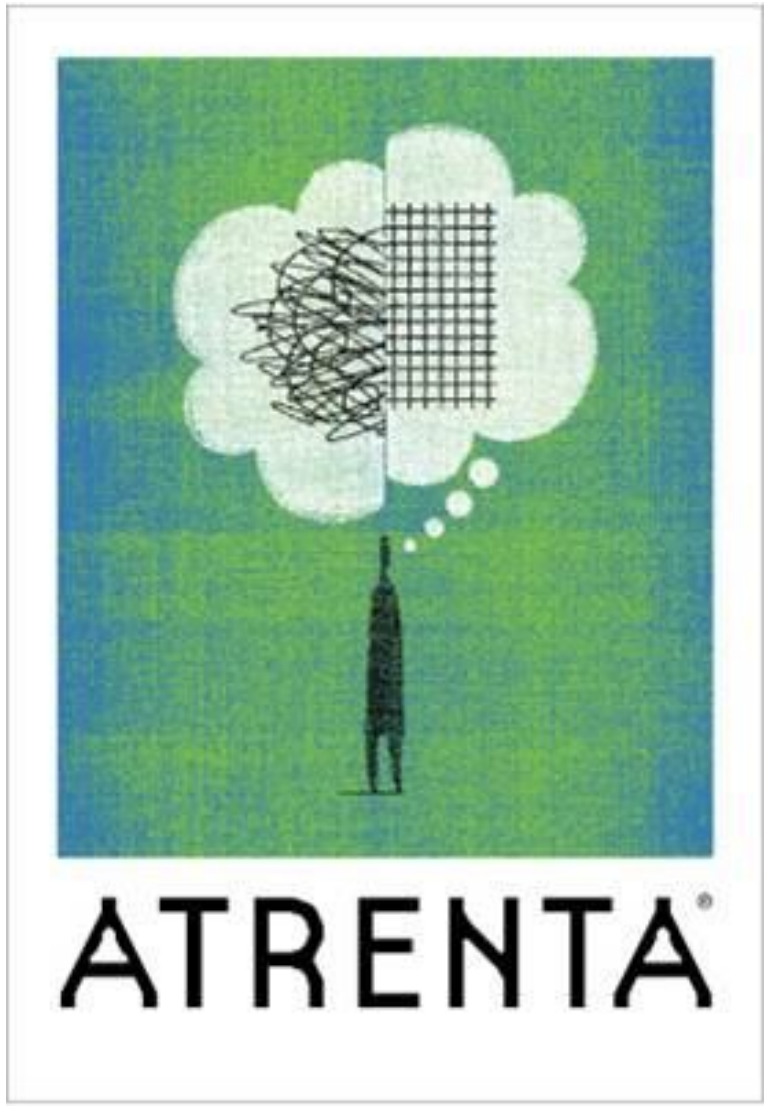


Codifies best-in-class methods – capture/propagate design intent

Enhances existing methodology – improve productivity with better handoff

Enables communication of methodologies between partners

Atrenta Confidential © 2008 Atrenta Inc.



SpyGlass Environment Overview

Atrenta Confidential © 2008 Atrenta Inc.

Some SpyGlass Definitions

- **Methodology**

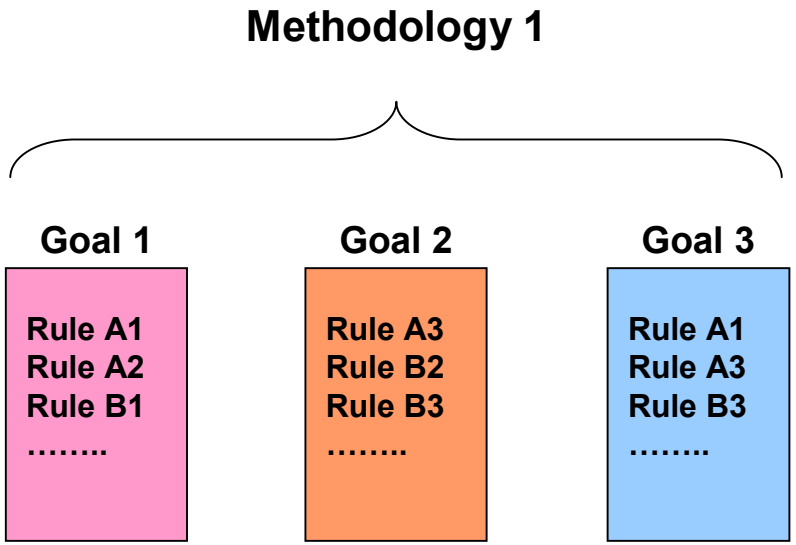
  A set of related Goals (templates)

- **Goal (template)**

  A file containing rules and rule setup

- **Rule**

  Individual check for specific issues



Atrenta Confidential © 2008 Atrenta Inc.

More SpyGlass Definitions and Acronyms

- **SpyGlass Design Constraint or SGDC**
  - SpyGlass Design Constraint file – additional design information that is not apparent in the RTL
- **Atrenta Console**
  - The analysis and debug environment, with session management, analysis guidance and enhanced reporting capabilities
- **Black-Box**
  - Is a term for a device or system or object when it is viewed primarily in terms of its input and output characteristics

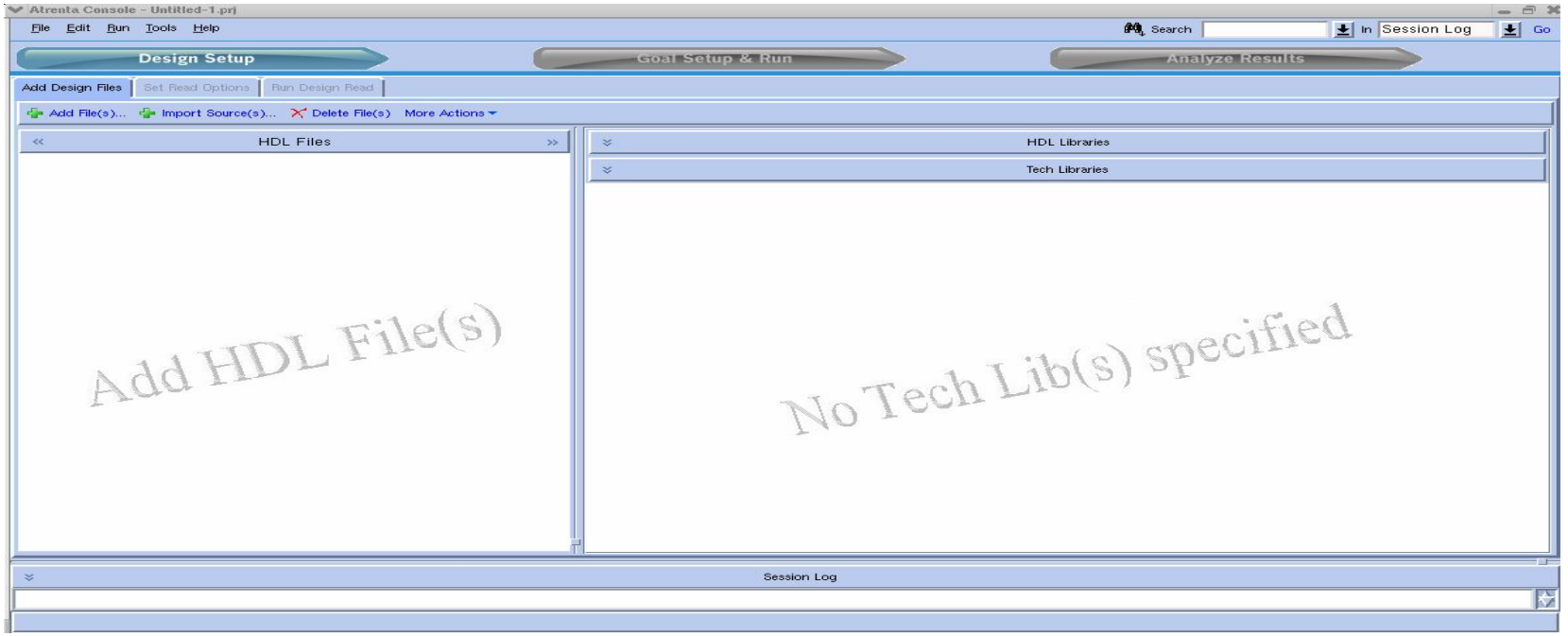
Atrenta Confidential © 2008 Atrenta Inc.

Invoking Atrenta Console

- **Atrenta Console can be run in either batch or GUI mode. The default is GUI mode.**
- **To run SpyGlass in the default, GUI mode, type**

  `spyglass` **or** `spyglass -project <project name>`
- **To run SpyGlass in batch mode, type**

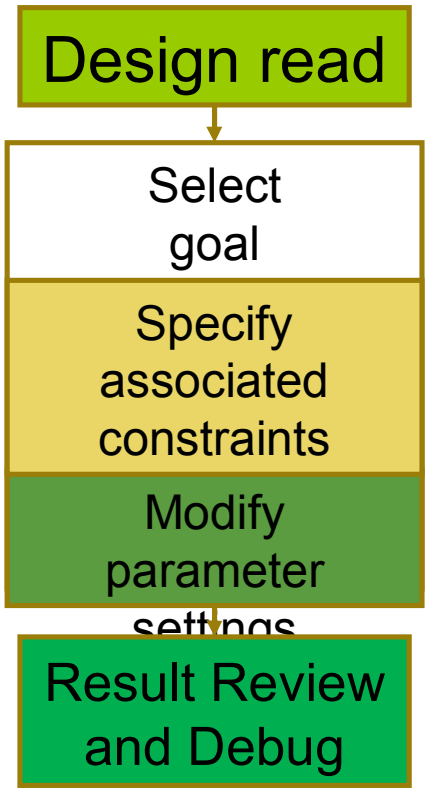
```
spyglass -batch -project <project name> \
        [-designread|-goals|-showgoals][-group]
```



Atrenta Confidential © 2008 Atrenta Inc.

Atrenta Console: Overview

- **Atrenta Console is built on two key principles**
  - In-line help
  - Guided Flow
- **The user interface is structured into three distinct steps**



- Load the design into SpyGlass for Analysis.
- Choose a goal from the selected methodology
- Certain goals require constraints to be specified in addition to the RTL/netlist/library information
- Parameters associated with the selected goal rules may be configured as necessary

Atrenta Confidential © 2008 Atrenta Inc.

Atrenta Console: Session Management

- **Built-in session management**
  - Allows a user to store the results of multiple analysis runs, and return to a saved session at any time
    - Facilitates session management, which allows the user to restore the results of previous analysis runs
    - A project file (<name>.prj) is an ASCII text file containing pointers to results and session status
  - Projects are created in two situations
    - When reading a design into Console for the first time
    - At any other time via 'File'->'New Project' or 'File'->'Save Project As…'
  - A project <name>.prj will have an associated directory <name> to store results of previously completed runs

Atrenta Confidential © 2008 Atrenta Inc.

Atrenta Console: Aggregate Reports

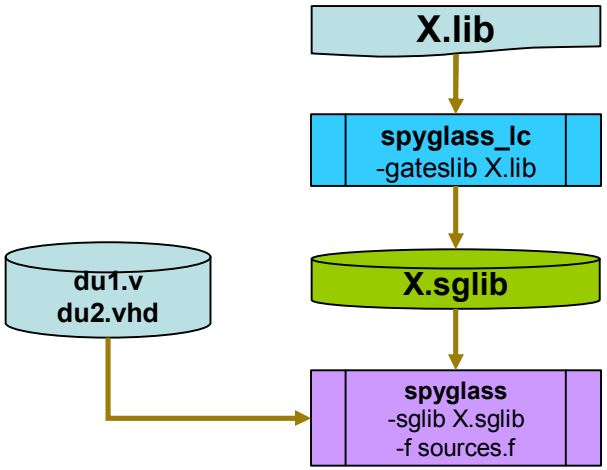
- **Improved means of providing high level summary reports in graphical (HTML) format and spreadsheet (CSV) format**
- **Different levels of report granularity – block level and multi-block (chip-level) summaries**
- **Aggregate reports are designed to work in conjunction with existing SpyGlass reports to give single/multi-block status summaries**
  - A block owner can generate summary reports for their own block
  - A design lead can generate multi-project summaries

Atrenta Confidential © 2008 Atrenta Inc.

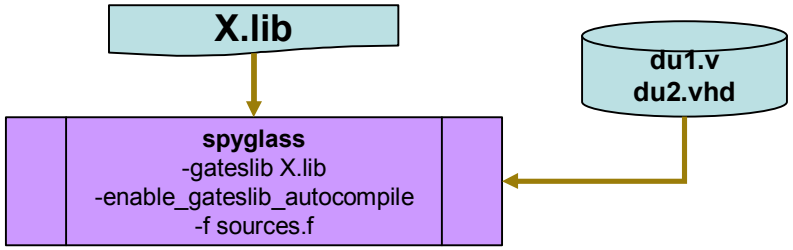
Use of Library Cells

- **Simplify user library handling by providing a unified Synopsys library and SpyGlass library to RTL**

Option 1: If the library rarely changes, then simply precompile it for later use



Option 2: If the library is small or changes frequently, consider a single-step precompile flow



- **Use compressed libraries for large files:**

  spyglass_lc –gateslib X.lib.gz

Atrenta Confidential © 2008 Atrenta Inc.

Blackboxes and Unsynthesizable Modules

- **Blackbox: A design unit with no structural definition**
  - May have no definition of any kind
  - May have its internal logic hidden by translate_off/translate_on pragmas
  - May be unsynthesizable
- **Unsynthesizable Modules: Restricts structural analysis**
  - For example, testbenches and simulation models
  - Treated as blackboxes in structural analysis. Therefore –
    - Combinatorial loops are not detected
    - Clock domain crossings are not detected
    - Estimated fault coverage is low

Atrenta Confidential © 2008 Atrenta Inc.

Reporting black boxes

- **The AnalyzeBBox rule group reports several kinds of black box, described by the sub-rules below**
  - The *InfoAnalyzeBBox* rule reports the following scenarios:
    - '-stop' command line option used
    - 'assume_path' constraint specified
  - This *WarnAnalyzeBBox* rule reports the following scenarios:
    - functional view is missing in the given .sglib files
    - definition is empty or masked completely by pragmas
  - The *ErrorAnalyzeBBox* rule reports the following scenarios:
    - definition is missing
    - DesignWare components used but '-dw' option is not specified
    - could not be synthesized due to synthesis errors
  - The *FatalAnalyzeBBox* rule reports the messages generated by above WarnAnalyzeBBox and ErrorAnalyzeBBox rules and '-nobb' switch is specified.

Atrenta Confidential © 2008 Atrenta Inc.

Working with SGDC

- **SGDC's are used to specify information about the design beyond what can be determined from the RTL/netlist**
- **Examples include clock definitions, DFT setup, Power setup, timing exceptions etc.**
  - More detail in later modules of the training
- **There are two ways of applying SGDC constraints in Atrenta Console**
  1. When performing analysis via GuideWare where constraints are required as input, guidance is provided via setup wizards and in-line help to assist in the SGDC creation and fine-tuning process
  2. Pre-existing SGDC constraints can easily be applied when running a goal requiring them

Atrenta Confidential © 2008 Atrenta Inc.

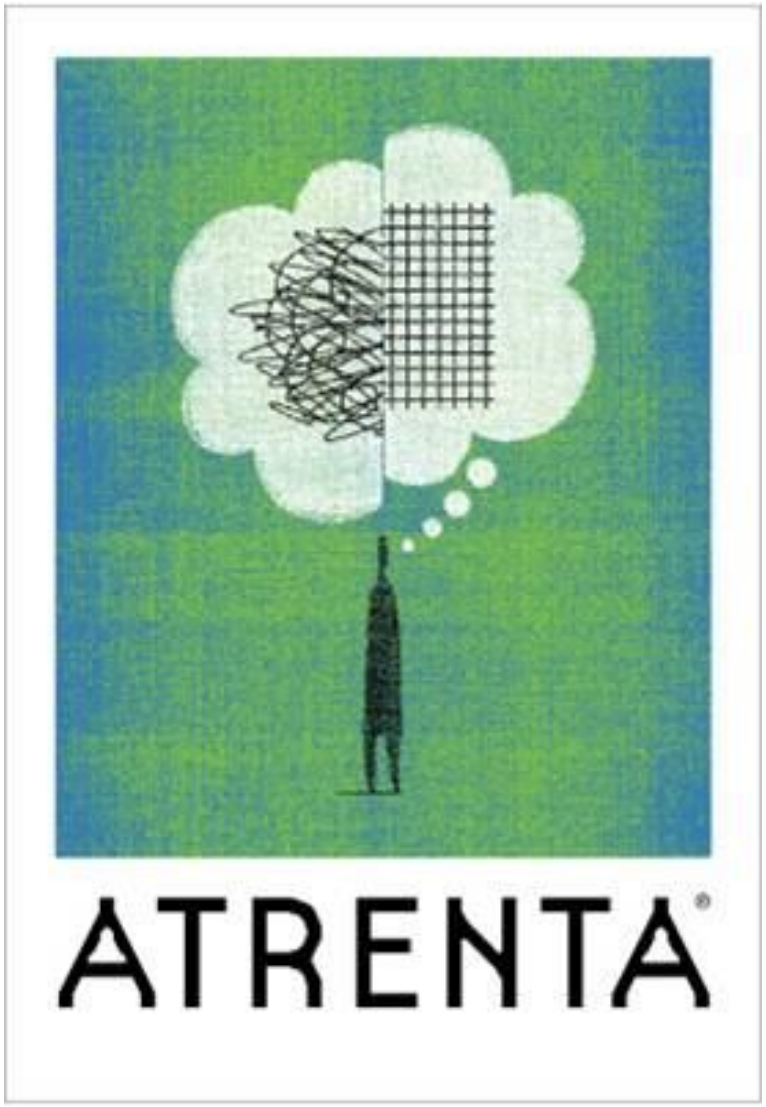
Lab 1: Getting familiar with SpyGlass

Lab duration: 30 minutes

After completing this lab, you should be able to:

- **Read the design into SpyGlass**
- **Run some GuideWare goals on a sample design**
- **Review the results and explore different debug methods**
- **Create SGDC for selected goals**
- **Waive a selected violation, and save the waivers for later use**

Atrenta Confidential © 2008 Atrenta Inc.

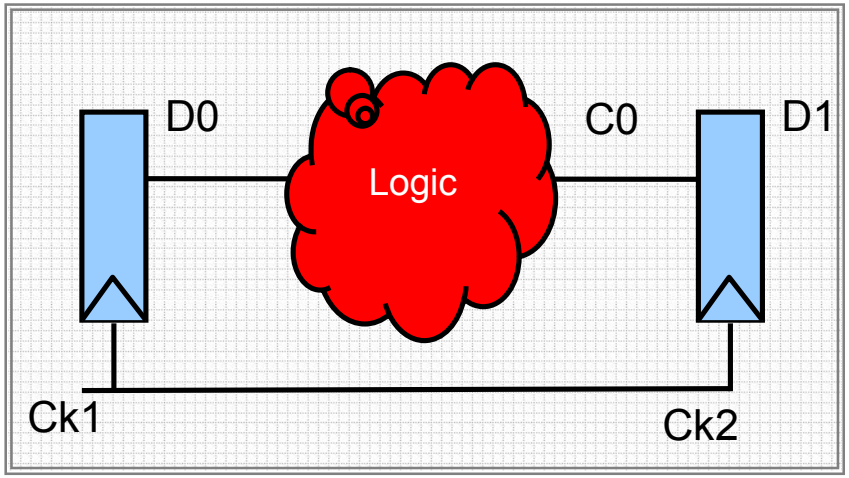


Introduction to Clock Domain Crossing Issues

THINK EARLY DESIGN CLOSURE THINK ATRENTA®

Atrenta Confidential © 2008 Atrenta Inc.

Principle of Synchronous Design



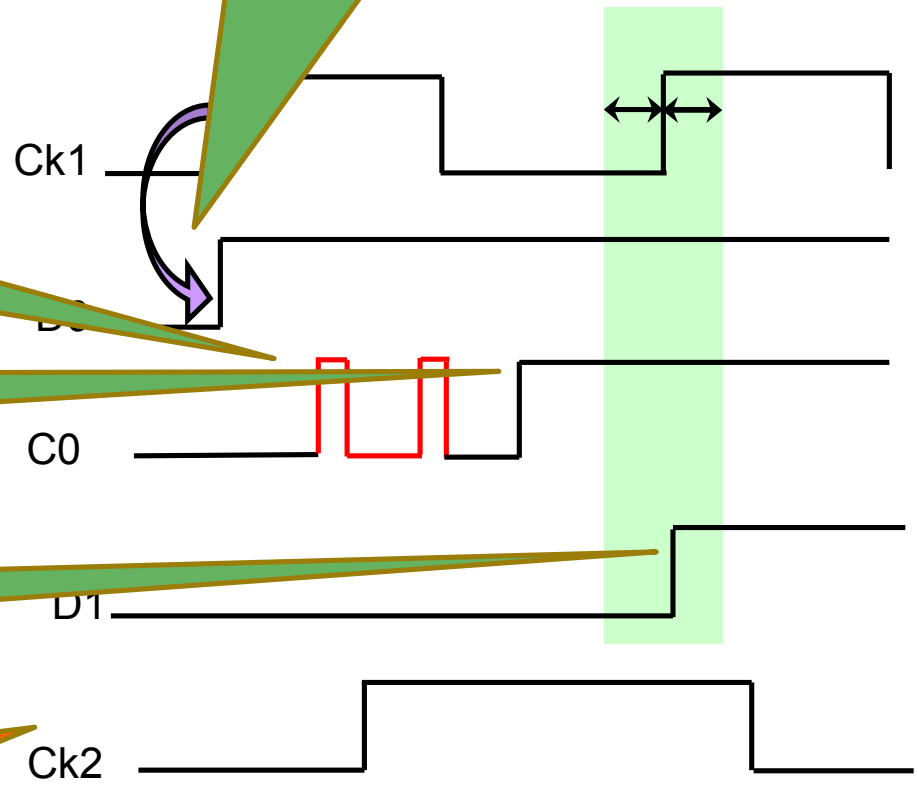
Input to combination logic changes upon clock edge

Combo logic computation → transient glitches

Computation settles before next edge of clock

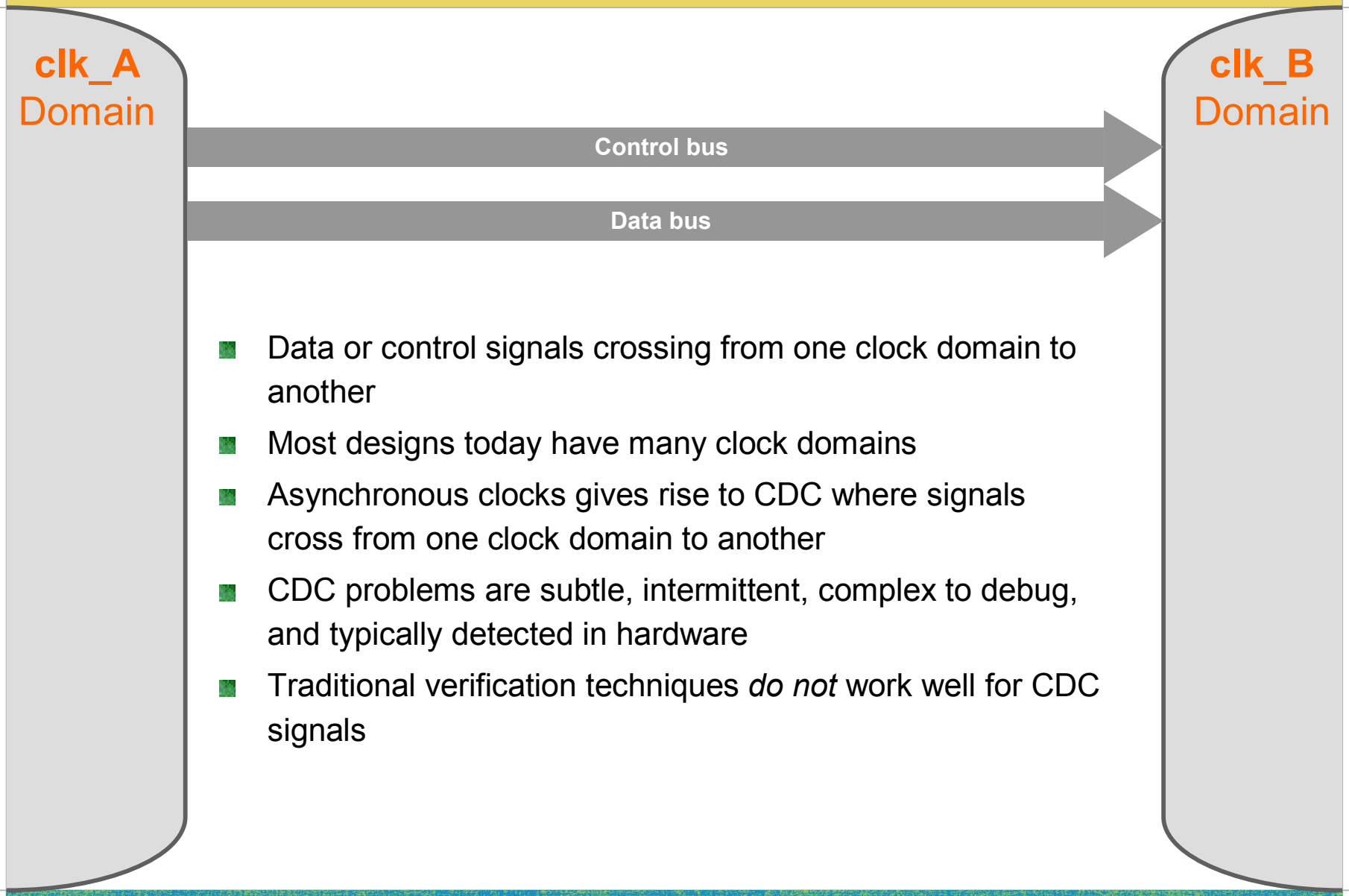
D1 captures a stable data honoring setup/hold time

But what if D1 is clocked by another asynchronous clock?



Atrenta Confidential © 2008 Atrenta Inc.

What is a Clock Domain Crossing (CDC)?



- Data or control signals crossing from one clock domain to another
- Most designs today have many clock domains
- Asynchronous clocks gives rise to CDC where signals cross from one clock domain to another
- CDC problems are subtle, intermittent, complex to debug, and typically detected in hardware
- Traditional verification techniques *do not* work well for CDC signals

Atrenta Confidential © 2008 Atrenta Inc.

Traditional Timing Analysis and their Limitations

- Timing verification
  - STA is not applicable to asynchronous interfaces, only within synchronous modules/blocks
  - Paths crossing clock domains are normally set to be false paths
  - Late in the design cycle at netlist level

- Functional simulation
  - Black-box testing is too cumbersome and not exhaustive
  - White-box testing requires user to write assertions
  - Requires writing test-benches for all the crossings
  - Is very reliant on luck (since simulation is not exhaustive), and will only detect a small subset of errors
  - Detected late in the design cycle

Clearly, existing techniques are inadequate for CDC

Atrenta Confidential © 2008 Atrenta Inc.

Types of CDC Problems

- **Problem #1: Meta-stability**

  How do you ensure that you have isolated metastability across clock domains so it does not affect design functionality?

- **Problem #2: Data hold problem (data loss)**

  How do you ensure that data from one clock domain is held long enough to be captured by the other clock domain? (Fast to slow clock, data enable sequencing)

- **Problem #3: Re-convergence (correlation)**

  How do you ensure that a group of converging synchronized control signals are in sync at a particular clock cycle?

- **Problem #4: Design intent across clock domains**

  How do you ensure that more sophisticated synchronization schemes (such as handshake, FIFO) behave according to your specifications?

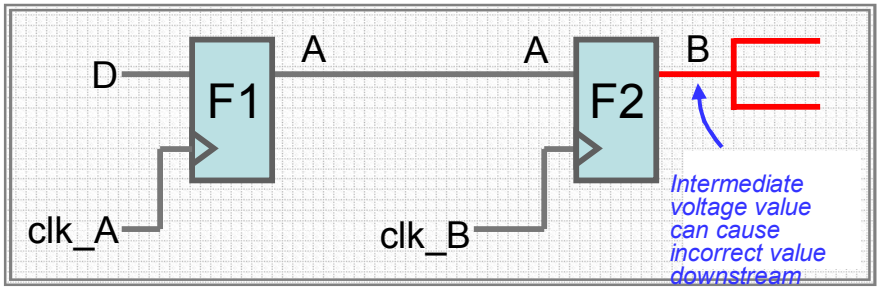
- **Problem #5: Reset synchronization**

  How do you ensure that an asynchronous set/reset is properly synchronized for a set of flops and the reset is asynchronously asserted and synchronously de-asserted

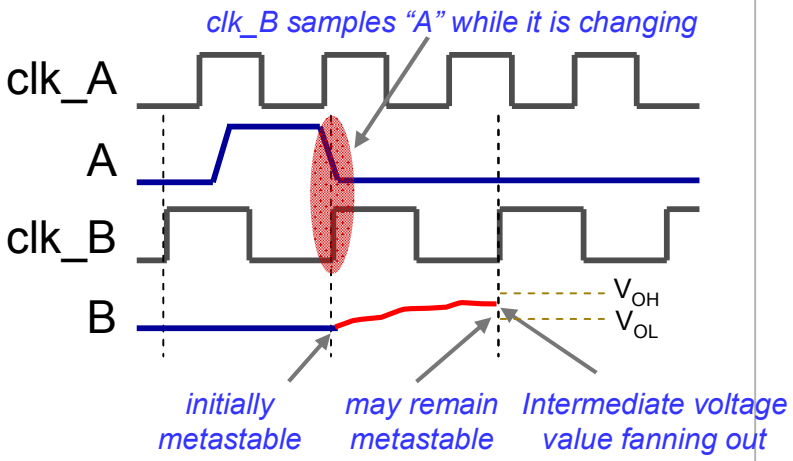
Atrenta Confidential © 2008 Atrenta Inc.

Problem #1: Meta-stability

Meta-stability problem



Output "B" becomes meta-stable whenever input "A" violates setup & hold time



Why is this bad?

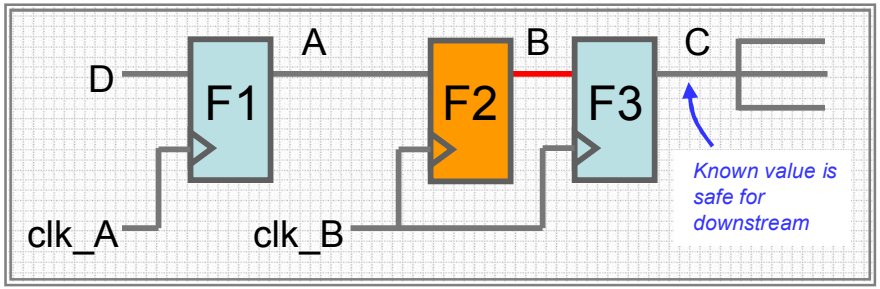
- The output of flop F2 is unpredictable

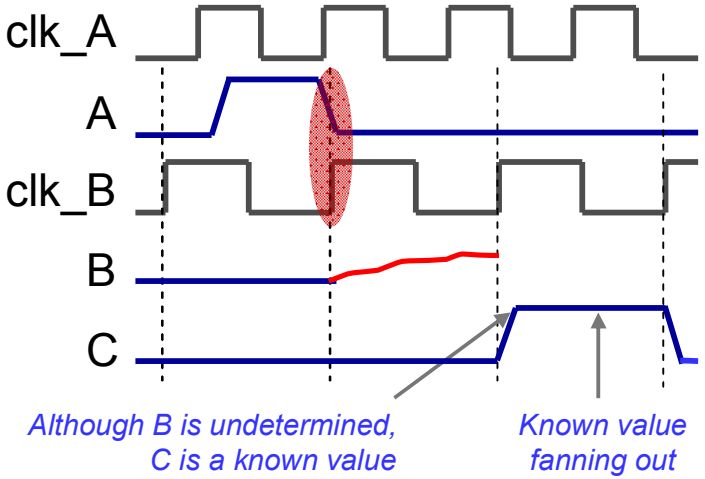
How can SpyGlass-CDC Help?

- SpyGlass-CDC identifies cases where synchronization is missing in the destination domain

Atrenta Confidential © 2008 Atrenta Inc.

Problem #1: Meta-stability





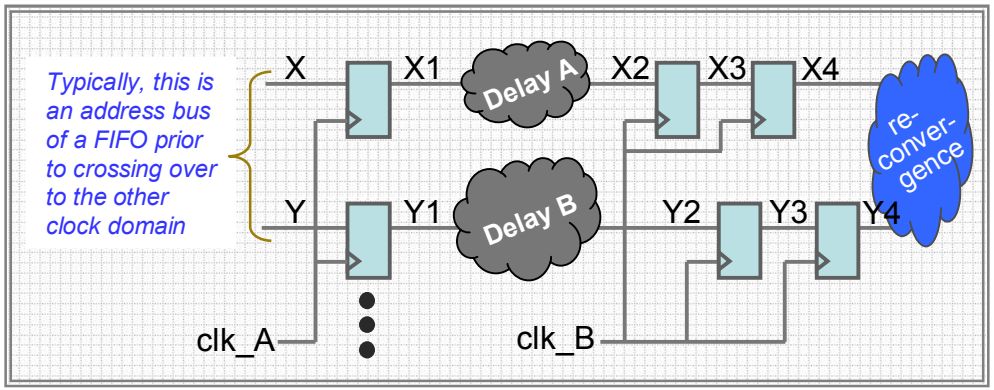
What designers typically do:

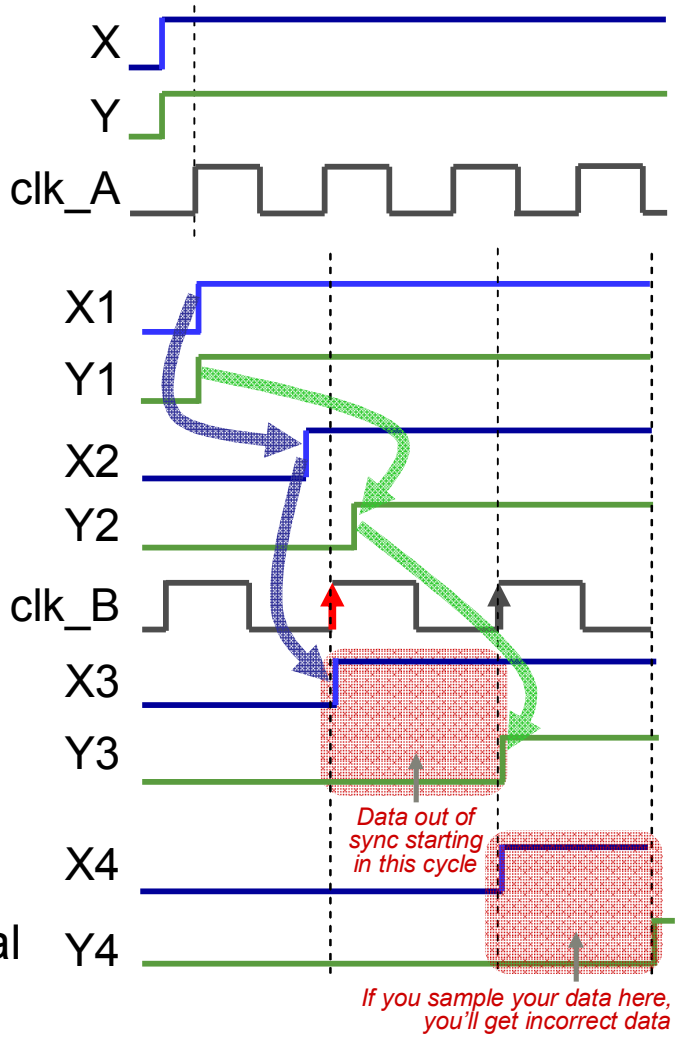
- Add an additional flop in the destination domain
- Add a mux recirculation flop in the destination domain, or
- Add a synchronization cell in the destination domain

Atrenta Confidential © 2008 Atrenta Inc.

Problem #2: Re-convergence

Re-convergence problem





Why is this bad?

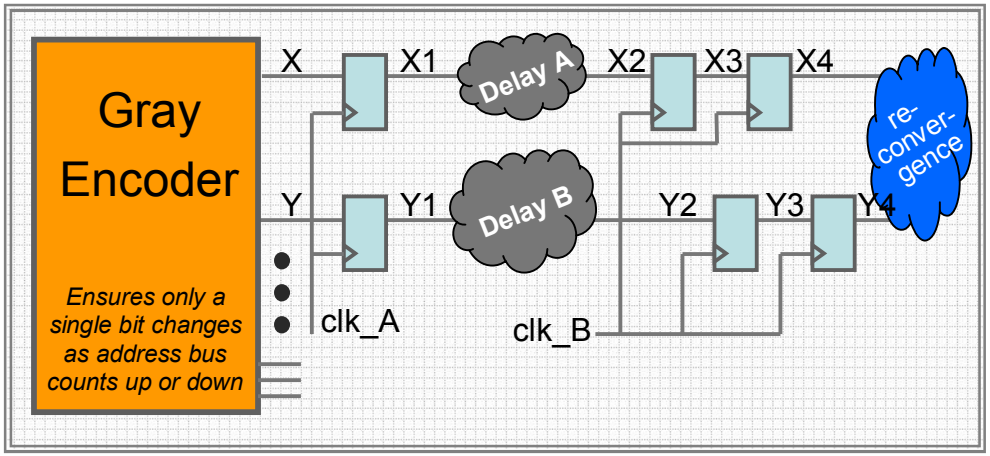
- Due to branch latencies incorrect data can be processed at the re-convergent point

How can SpyGlass-CDC help?

- Flags crossings where different pairs of synchronizers converge on some combinational logic

Atrenta Confidential © 2008 Atrenta Inc.

Problem #2: Re-convergence



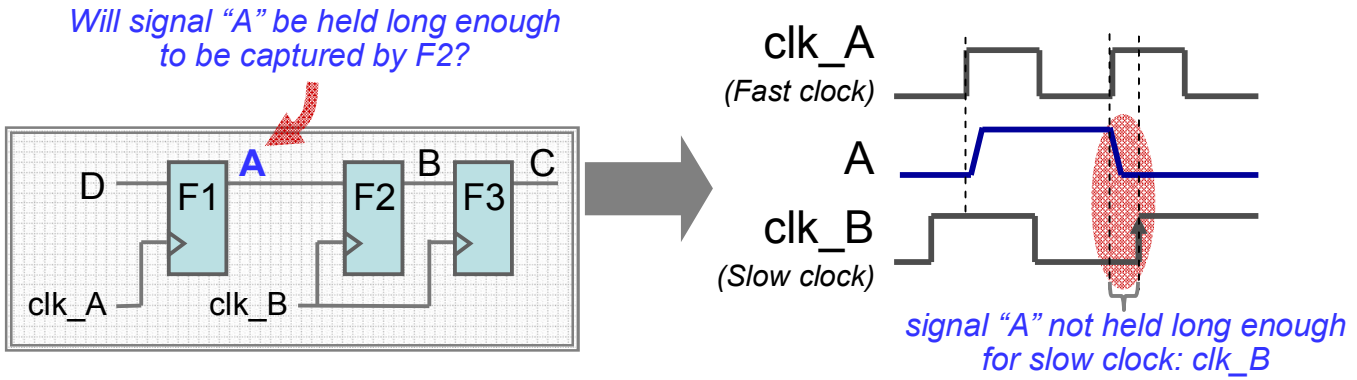
What designers typically do:

- Implement a gray encoder before the clock crossing and not between the clock crossing
  - Gray code ensures only one bit on the converging bus changes within one clock cycle

Atrenta Confidential © 2008 Atrenta Inc.

Problem #3: Data Hold

Data Hold problem *(Signal crosses from a fast clock domain to a slow clock domain)*



Why is this bad?

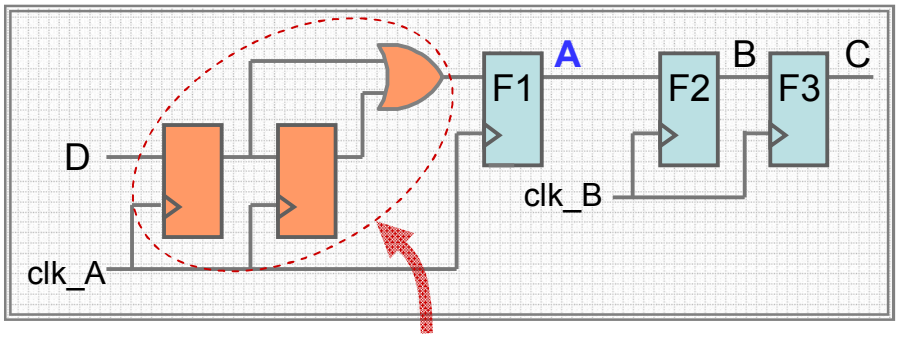
- Possible that the signal in the faster clock domain of F1 is not held long enough to be captured in the much slower clock domain of F2

How can SpyGlass-CDC help?

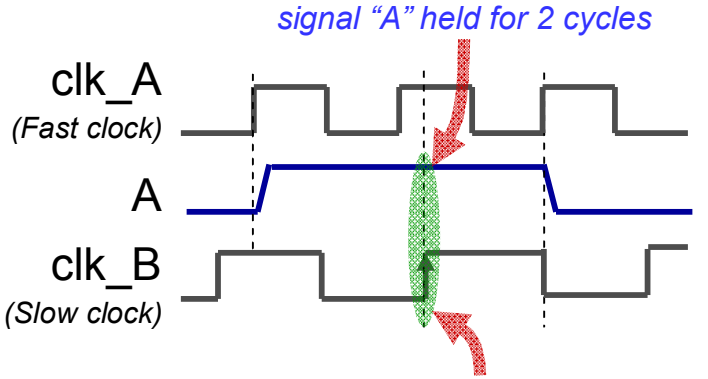
- For every fast-to-slow CDC crossing, a formal check is automatically carried out. If the data crossing the domain crossing is not captured, SpyGlass-CDC reports a violation has occurred.
  - A waveform trace is generated.

Atrenta Confidential © 2008 Atrenta Inc.

Problem #3: Data Hold



Simple 2-cycle Extender



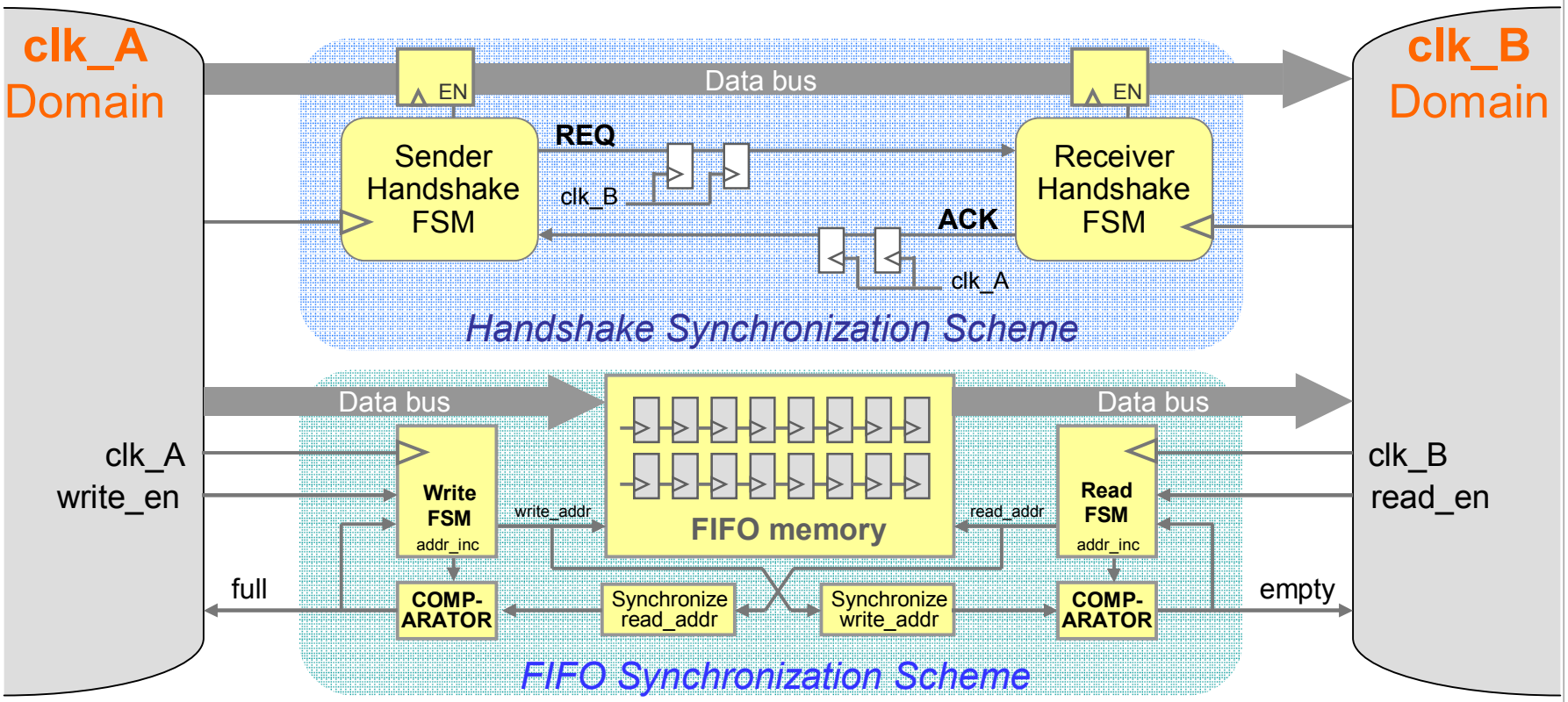
What designers typically do?

- Implement pulse extenders - To ensure a good margin between the data being stable and the clock edge being active

Atrenta Confidential © 2008 Atrenta Inc.

Problem #4: Design Intent (Handshake, FIFO)

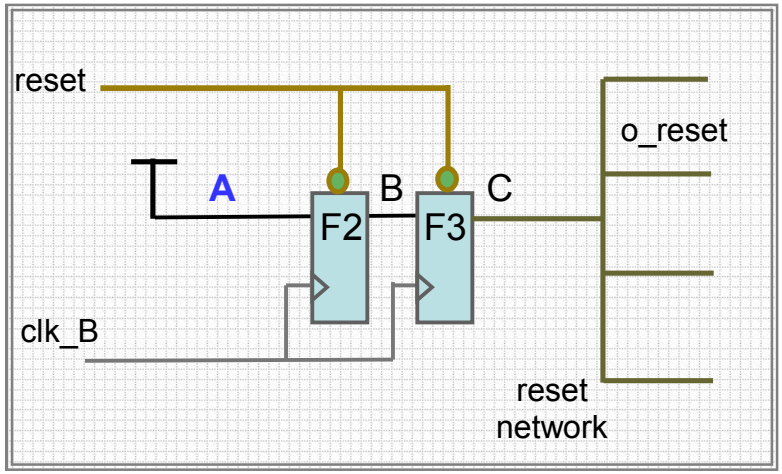
- Sometime more sophisticated synchronization schemes are employed
- Handshake synch scheme: good for portability, IP, reuse
- FIFO synch scheme: good for high volume data transfer



Atrenta Confidential © 2008 Atrenta Inc.

Problem #5: Reset Synchronization

- Asynchronous reset must be deasserted synchronously
- Asychronous deassertion of resets may put design into unintended state



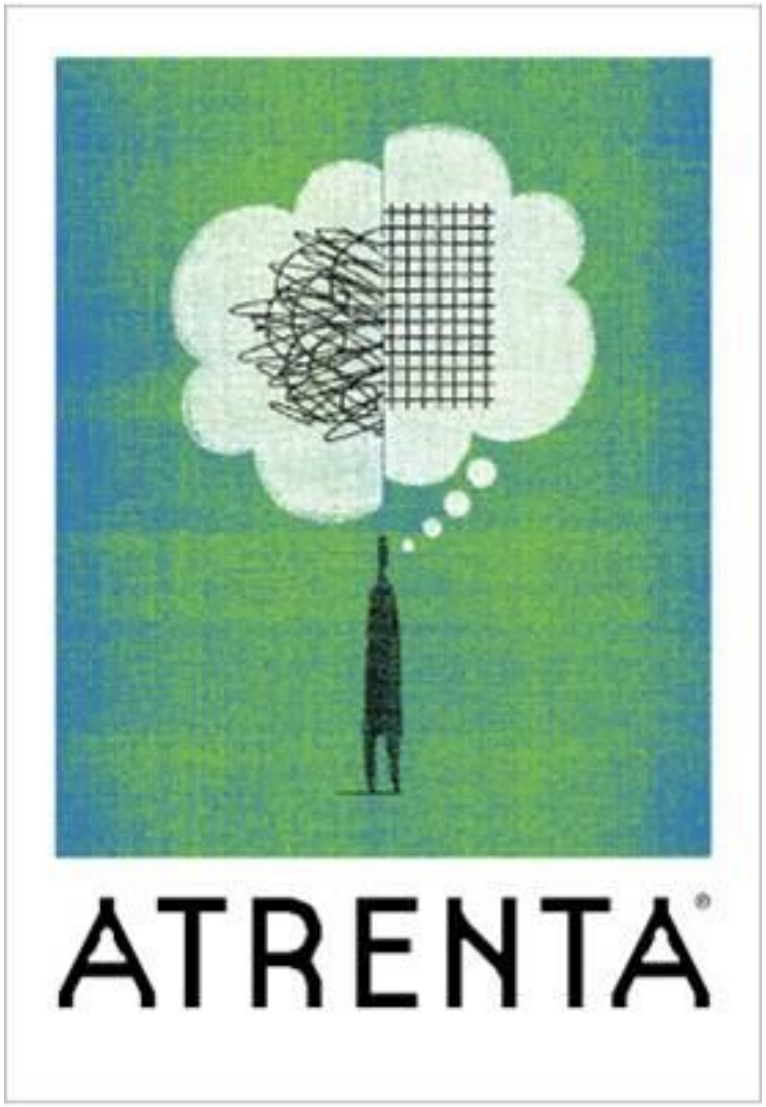
Reset Synchronizer

Atrenta Confidential © 2008 Atrenta Inc.

Quiz

1. What is the difference between Synchronous clock domains and Asynchronous clock domains?
2. Are Synchronous clock domain crossings safe?
3. Name two main problem caused by Asynchronous clock domain crossings.
4. Does Metastability problem depends upon the frequency of destination clock?
5. Can static timing tool like Synopsys Prime Time detect Asynchronous Clock domain crossings?
6. Can Clock Tree Synthesis (CTS) fix clock domain crossing issue?

Atrenta Confidential © 2008 Atrenta Inc.

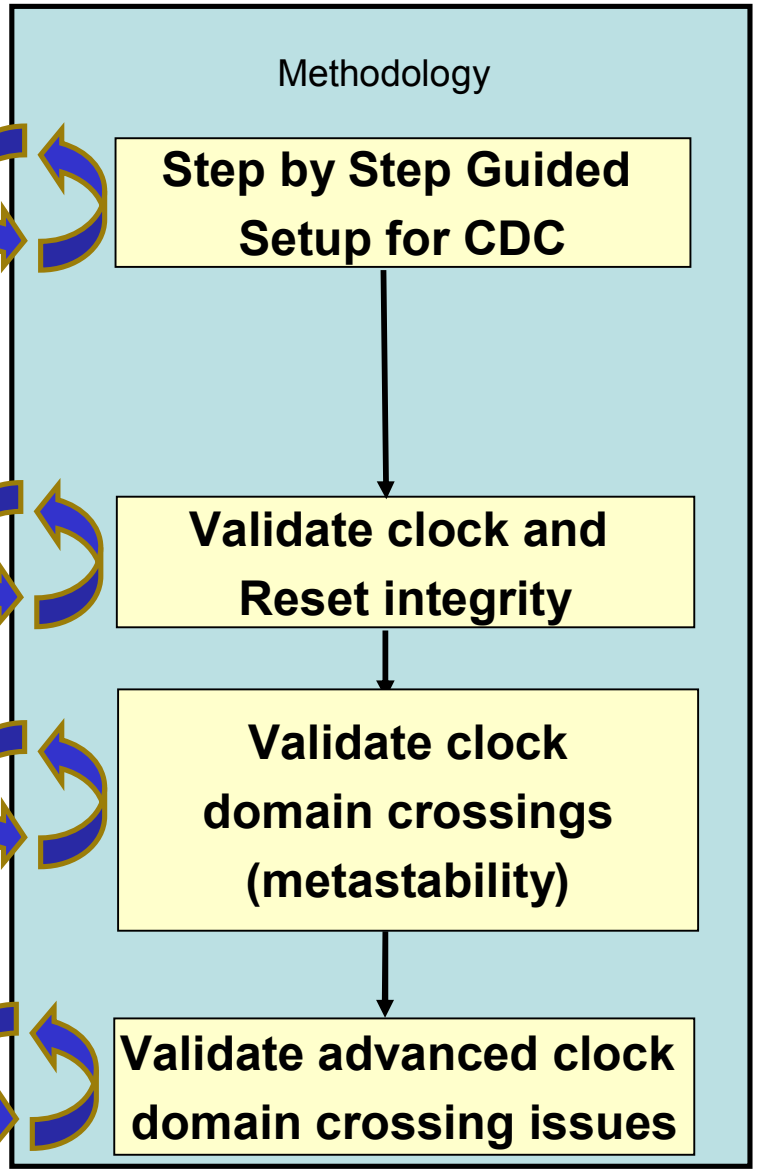


SpyGlass-CDC Flow and Methodology

THINK EARLY DESIGN CLOSURE THINK ATRENTA®

Atrenta Confidential © 2008 Atrenta Inc.

SpyGlass-CDC Tool Flow



Template

CDC-Setup-Manager
Review abstracted clock tree and finalize clocks in design
Verify Clocks Setup
Report resets in the design
Black-box setup/IO setup

ClockReset_Integrity
Checks clocks and reset trees for integrity issues

CDC-VerificationBase
Checks clocks/resets synchronization status

CDC-Verification
Reports reconvergence, I/O sampled by multiple clocks, data loss
FIFO/Handshake verification
Reset synchronization

Rules

- **Clock_info01**
- **Reset_info01**
- **Setup Clock01**
- **Clock_info03a**
- **Clock_info05a**
- **Clock_glitch02**
- **Clock_info05b**
- **Clock_sync01**
- **Clock_sync02**
- **Ac_conv01/02**
- **Clock_sync05/06**
- **Ac_cdc01**
- **Ac_fifo01**
- **Reset_sync01**

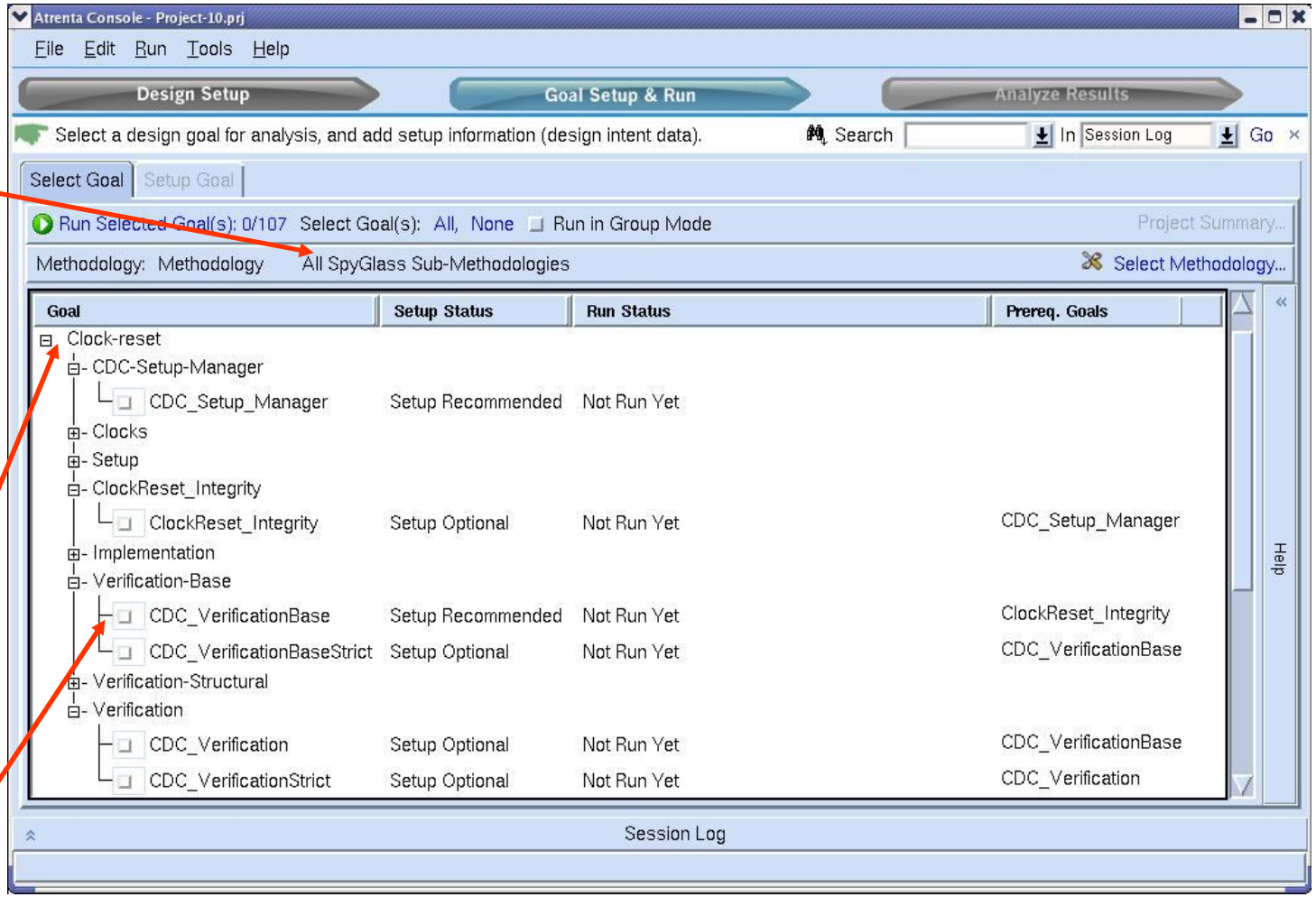
Atrenta Confidential © 2008 Atrenta Inc.

The Template/Goals Window

templates

Policy

Goals



Atrenta Confidential © 2008 Atrenta Inc.

SpyGlass-CDC Recommended Methodology

- **If possible, use a divide-and-conquer approach**
  - Perform analysis at the block/ leaf level
    - Remove violations
    - Apply waivers
    - Apply false paths
- **Integrate blocks one at a time and move up the hierarchy**
  - While doing so, add appropriate constraints at the block boundaries
    - IP_block constraint
    - input/output constraint
  - Include block level SGDC constraints, such as exceptions while moving up in hierarchy
  - Import waivers
    - `waive -import <block_name> <block-waive-file1>`
- **Exclude the complex clock generation/selection logic by black-boxing this logic. Alternatively, apply a `-stop` constraint on these modules**
  - This logic is better verified using simulation

Atrenta Confidential © 2008 Atrenta Inc.