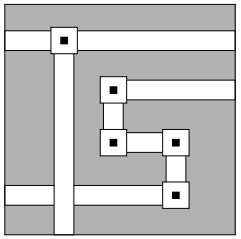


TRILOBYTE
SYSTEMS

Consistent Timing Constraints with PrimeTime

Steve Golson
Trilobyte Systems

http://www.trilobyte.com

Physical implementation

Rule #1

Do not change the functionality

Rule #2

Meet the specified timing constraints

Physical Implementation Tools

automatic test pattern generation (ATPG)

cell placement

clock tree synthesis (CTS)

design-for-manufacturing/design-for-yield (DFM/DFY) tools

detail router

floorplanning

Physical Implementation Tools

gate-level power analysis

global router

RTL power analysis

scan insertion

static timing analysis

synthesis from RTL to gate-level netlist

voltage drop analysis

Timing-driven Physical Implementation Tools

Q: What do all these *timing-driven* tools have in common?

A: They require *correct* and *complete* timing constraints

Correct: constraints agree with the chip spec

Complete: all paths are constrained

What's the big deal?

Just write

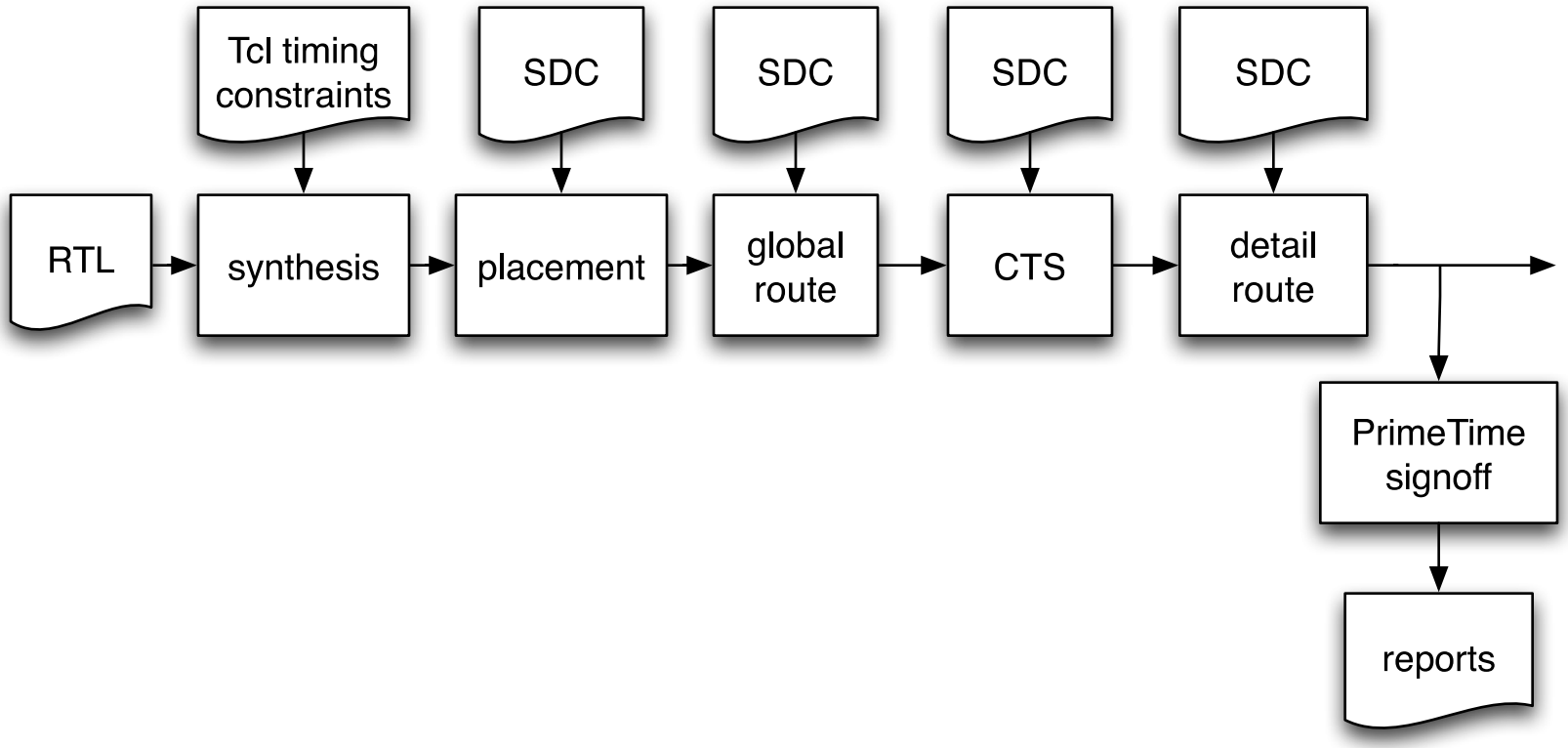
"the SDC file"

and use it for every tool

Why tools require different files

- File formats
- Versions
- Netlist status
- Features

Flow


Tcl timing constraints
SDC
SDC
SDC
SDC
RTL
synthesis
placement
global route
CTS
detail route
PrimeTime signoff
reports

Problems

Problem #1

How to *manage* this multiplicity of files

Problem #2

How to ensure *consistency* across tools

Consistency

Timing-driven tools have

consistent timing constraints

if, given the same netlist,

they identify the same critical path

for a given path group.

Consistency

1. For tool A, for each path group, report the critical path.

2. In tool B, report the identical path (startpoint, endpoint, through points, clocks).

3. Tool B path must report same constraints (e.g., launch clock time, capture clock time, clock latencies, input/output delay).

4. Tool B must report the same slack.

Consistency

1. For tool A, for each path group, report the critical path.

2. In tool B, report the identical path (startpoint, endpoint, through points, clocks).

3. Tool B path must report same constraints (e.g., launch clock time, capture clock time, clock latencies, input/output delay).

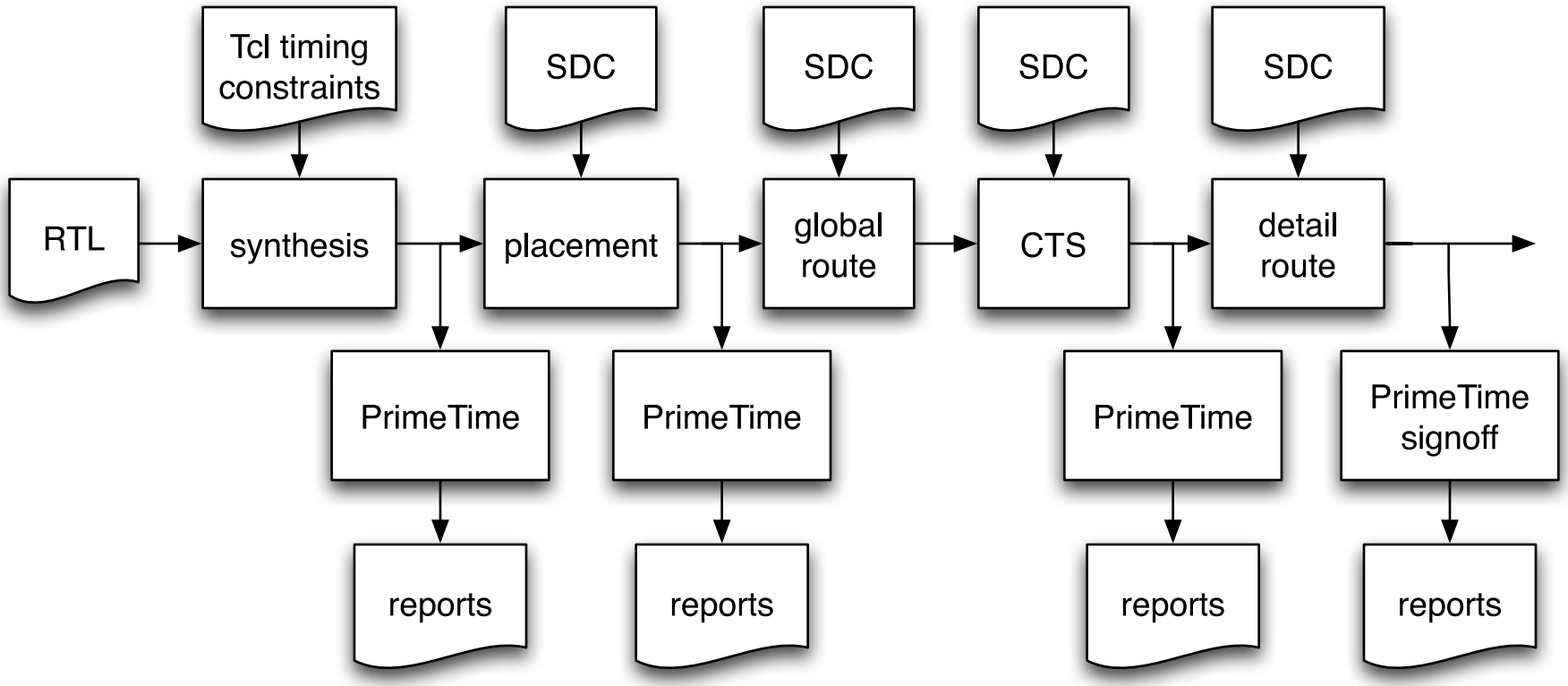
4. Tool B must report the same slack, within some reasonable margin.

Solution #2: Use PrimeTime throughout the flow

Switches to add to “signoff” PrimeTime scripts:

- High-fanout nets? (yes or no)
- Back-annotated parasitics? (yes or no)
- Propagate clocks? (yes or no)
- Crosstalk analysis? (yes or no)
- Netlist status (for example, scan inserted or not)

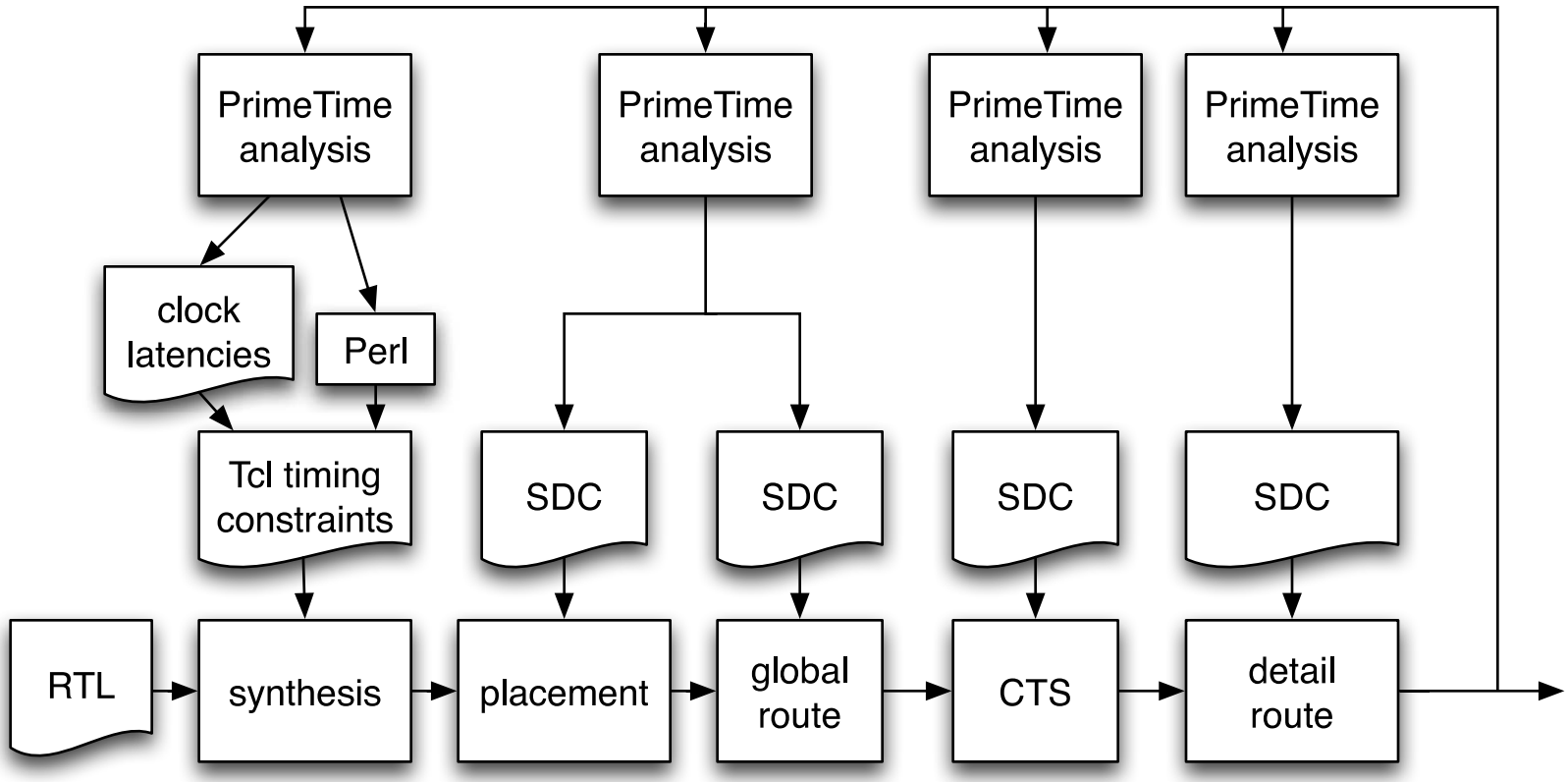
Flow


Tcl timing constraints
SDC
SDC
SDC
SDC
RTL
synthesis
placement
global route
CTS
detail route
PrimeTime
PrimeTime
PrimeTime
PrimeTime signoff
reports
reports
reports
reports

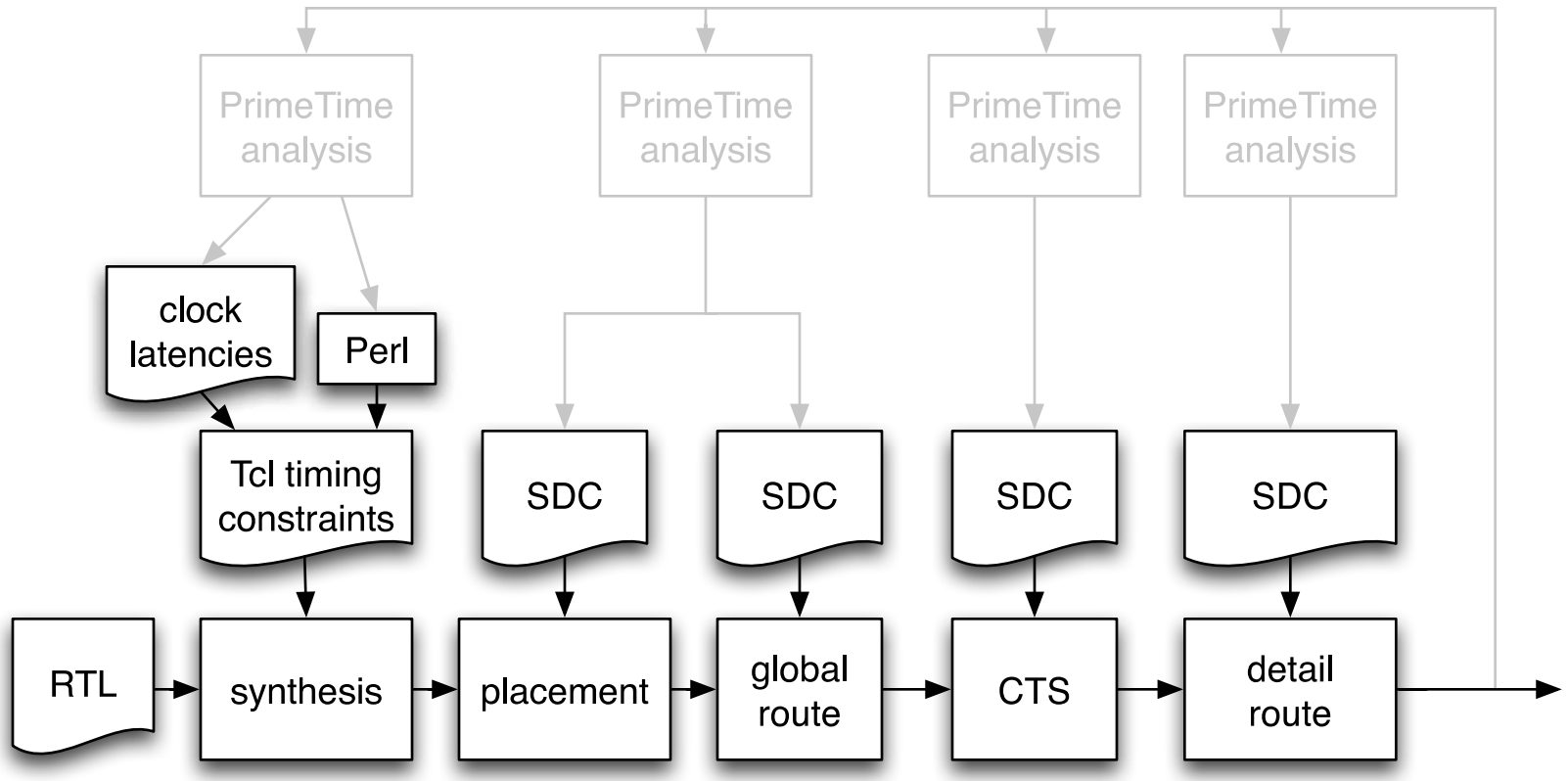
Solution #1: Use PrimeTime to generate *all* needed files

- **`write_sdc`**
- **`write_sdc`** and post-process the SDC (Perl script)
- custom PrimeTime Tcl to generate needed files

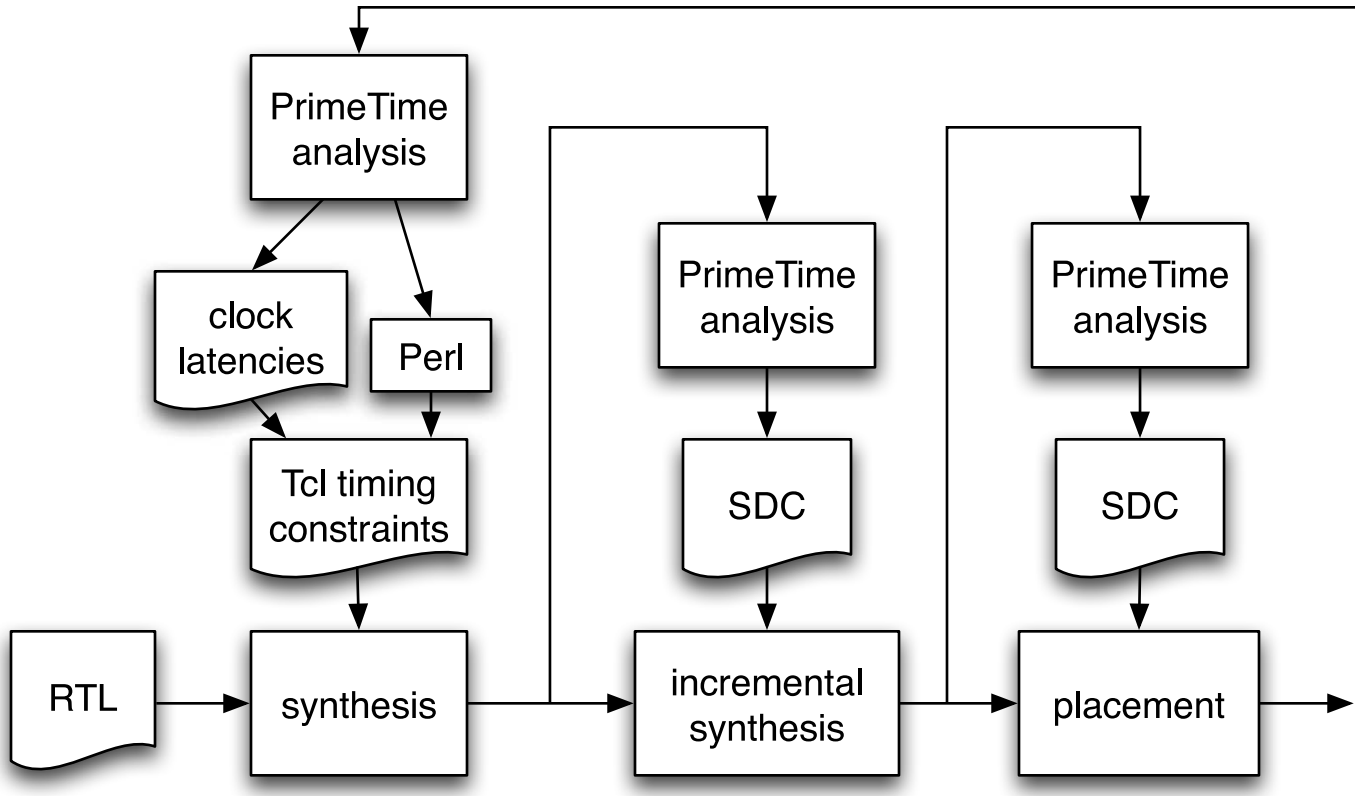
Flow


PrimeTime analysis
PrimeTime analysis
PrimeTime analysis
PrimeTime analysis
clock latencies
Perl
Tcl timing constraints
SDC
SDC
SDC
SDC
RTL
synthesis
placement
global route
CTS
detail route

Flow


PrimeTime analysis
PrimeTime analysis
PrimeTime analysis
PrimeTime analysis
clock latencies
Perl
Tcl timing constraints
SDC
SDC
SDC
SDC
RTL
synthesis
placement
global route
CTS
detail route

Flow


PrimeTime analysis
clock latencies
Perl
Tcl timing constraints
RTL
synthesis
PrimeTime analysis
SDC
incremental synthesis
PrimeTime analysis
SDC
placement

Problem: SDC command interpretation

SDC specification defines *syntax*, not *behavior*

If the SDC says

```
set_multicycle_path 2 -from [get_clocks clkA]

set_multicycle_path 3 -from [get_clocks clkA]  \
                      -to   [get_clocks clkB]

set_multicycle_path 4 -to   [get_clocks clkB]
```

What is the multicycle value for a path from clkA to clkB?

Problem: Hierarchy and budgeting

- Maintain consistent constraints across hierarchy
- Block-level constraints
  must be *automatically* generated
  from full-chip constraints
- Min delays (hold) are as important as max delays
- Use default budgets, or simple slack allocation
- Accurate clock latencies are critical
- This is really hard

Problem: Hierarchy and promotion

- Block-level constraints must be promoted to full-chip

  example: IP block with supplied block-level path exceptions

- Must *automate* this process

- “Identical” blocks may have different constraints

  example: multiple copies of CPU, or IO interface

  perhaps different modes, with different path exceptions

  perhaps operating at different voltages

Example: extract clock latencies

Given a netlist with propagated clocks,
how to describe the equivalent ideal clock timing?

What we want is a Tcl file that looks like this:

```
########## cpuA_clk
set clock_source_latency(cpuA_clk:SLOW:early:rise) 1.987
set clock_source_latency(cpuA_clk:SLOW:late:rise)  2.305
set clock_source_latency(cpuA_clk:SLOW:early:fall) 1.959
set clock_source_latency(cpuA_clk:SLOW:late:fall)  2.272
set clock_network_latency(cpuA_clk:SLOW:early:rise) 1.228
set clock_network_latency(cpuA_clk:SLOW:late:rise)  1.405
set clock_network_latency(cpuA_clk:SLOW:early:fall) 1.257
set clock_network_latency(cpuA_clk:SLOW:late:fall)  1.438
set clock_global_skew(cpuA_clk:SLOW)  0.187
```

Then source this into Design Compiler, PrimeTime…

PrimeTime Tcl script to extract clock latencies

foreach clock that is propagated and has a defined source
    get source latency attributes
    print source latency values
    foreach register using this clock
        find the max delay critical path to this capture register
            get capture clock latency => total early latency
            network early latency = total early – source
        find the min delay critical path to this capture register
            get the capture clock latency => total late latency
            network late latency = total late – source
    simple statistics on all network latency values
    print network latency values
    print global skew

Summary

Our goals:

- A methodology/flow which is
  - automated
  - flexible
  - consistent
- Repeatable, reliable results from our tools
- Manually edit and maintain constraints in *one* place: full-chip PrimeTime Tcl scripts
- Methodology which is not design-specific

Thank you

- Mark Sprague, AMD
- Jerry Frenkil, Sequence Design
- Many previous clients for examples, good and bad…

Questions?