ESNUG

(ESNUG 479 Item 2) --- [02/05/09]

Subject: (ESNUG 477 #6) User Benchmark of PrimeTime vs. GoldTime for STA

> Signal Integrity. We did all our timing analysis with GoldTime signal
> integrity turned on. (i.e. GT 1.5 hr, PT 3.0 hr, ETS 6.0 hrs was with
> SI.) I was impressed by how little time GoldTime took to perform the
> full-chip, SI-aware timing analysis with propagated clocks.
>
>   - [Dr. Evil]

From: [The Swedish Chef]

Hi, John,

We use Extreme-DA's GoldTime as our static timing analysis and timing
sign-off tool for our chip designs. We have had 4 tape-outs with it. The
following points were our driving reasons to adopt GoldTime:

  1. During our eval, GoldTime had excellent run times for our large
     designs. (See data below).

  2. GoldTime uses a small memory footprint. (See data below).

This allowed us to use cheap hardware to run our static timing analysis;
which enhanced productivity, since our design team has more hardware to
run STA. (Designers can run GoldTime on Linux desktops, rather than
requiring a machine with larger memory capacity.)

  3. Extreme-DA understands Statistical Static Timing Analysis (SSTA).
     With smaller technologies, corner based timing closure is problematic
     due to overly pessimistic guardbands; we can use SSTA to determine a
     more realistic timing analysis.

SSTA is part of Extreme DA's core competency, so we believe transitioning
to SSTA from corner-based STA will be a smooth process for us.

GoldTime's inputs are: design netlist, timing constraints, lib info (NDLM,
CCS, or ECSM) for cells, and parasitic information for nets (SPEF).

GoldTime's output is a set of timing reports that informs the user whether
his design passes the user specified design constraints.

RUNTIME:

Our experience with GoldTime has been very positive. Below is our data.

Tool versions used: GoldTime 2007.3.r1 vs. PrimeTime 2006.06

The design had 5.582 million instances.

	GoldTime		PrimeTime	
	time	mem	time	mem

```
     To read in the netlists:    2 min    3.6 G       5 min    5.0 G
           To annotate SPEF:   28 min    3.6 G     380 min   10.0 G
          To update timing:   27 min    4.8 G      90 min   13.0 G
    Generate timing reports:    8 min    5.1 G       2 min   13.0 G
```

Our total time to get a timing analyzed design in GoldTime was 1 hr, as
opposed to 8 hrs in PrimeTime. GoldTime was efficient enough for us
to debug design problems. PrimeTime wasn't.

In addition, GoldTime memory usage maxed at 5.1 GB. We can use our cheap
computer hardware for STA.

ACCURACY:

We selected a subset of timing paths in our designs, and compared GoldTime's
delay calculation with SPICE. It was within a 5% spread of SPICE.

CAPACITY:

In the timing analysis of our current chips (which are over 5 M instances),
we do not need to use Interface Logic Macros (ILMs) in order to expedite
runs in GoldTime. In PrimeTime, ILM's are essential.

CONSTRUCTIVE CRITICISM:

  - The GoldTime docs are ad hoc at best. We had to learn from our
    AE what it's commands and its options were. A "best practices"
    doc would be nice to have, too.

  - We need a catalogue of GoldTime's warning messages. The warnings
    and messaging are sometimes intuitive, sometimes strange. To
    investigate the more obscure cases, it would help if we had
    in-depth documentation.

We adopted GoldTime in June 2007. The TCL-based command set made it easy
to customize the tool setup.

    - [The Swedish Chef]