Pushing the Envelope

Use (and Abuse) of VLSI Technology Tools for Full-Custom Design

VLSI Technology Users Group
May 11, 1990

Steve Golson, Trilobyte Systems
Scott Griffith, Sun Microsystems

CAVEATS

- our experiences are based on our design of a 2Kbyte fully-associative cache memory with on-chip DRAM control (full-custom, 172k transistors, 2μ CMOS, 374x383 mils, 180 pins)
- for more details see the paper we presented at CICC 1989
- this design took two engineers from 4/87 to 12/87 (1st pass), 1/88 to 9/88 (2nd pass)
- tools versions used: v7r2 (1st pass), v7r3, v7r4 (2nd pass)
- we have been using VLSI Technology design tools since 1982
- regardless of all our problems, first silicon worked (well, almost...)
- we speak for ourselves, NOT our employers

COMPOSE

- with compose and a gutfull design, you can REALLY pack small modules (but this makes for lots of problems you may not expect...)
- large number of connectors breaks it (e.g. 128x5 ram periphery to cam)
- large number of objects breaks it (e.g. 128x128 ram array)
- any kind of autorouting breaks it (power and padring: hah!)
- some tools generate nodenames you can't edit (e.g. top.core.cell.$123.node)
- interactive selection of connectors on large objects is hard/impossible
- solution: use VIP for top level (you do know what VIP is, don't you?)
- chipcompiler did NOT fix these large design problems ∴ still must use compose

LAYOUT

- still the tool of choice, generally does the right thing
- edit in place is nice
- sometimes produces geometry that MEBES can't handle (you may have to drop your logo)
- bring back macros!

NETCOMPARE

- EQUIVs for power, ground, and well is a nightmare
- *GND*, *VDD* will find cellnames and EQUIV things you don't want
- every bitline and wordline in the ram needed STARTID before netc would converge at all
- had to STARTID every ram cell (built init file with shell script -- yuk!)
- other than that it did the right thing

EXTRACT

- one of the few tools you can rely on
- unless you use non-Manhattan geometries...
- extracted capacitances of gutfull cells did NOT agree with phantom cells
- huge temp files (~1Gbyte), huge run times (3 days on Sun 3/280), huge virtual memory requirements (~210Mbyte swap)

DRC

- generally reliable
- except be sure you rot 90 and check it again
- huge temp files (~1Gbyte), huge run times (6 days on Sun 3/280), huge virtual memory requirements (~210Mbyte swap)

SCHEMATIC

- icons are not your friend
- vectored instances would be nice

STATE MACHINE COMPILER

- clock trees are bogus
- not good at buffering heavily-loaded signals (why isn't smachn as smart as tv?)
- generates names you can't edit in schematic (e.g. instance $123)
- state machine compiler + makeschematic + timing verifier = successful design

SIM

- VLSI makes enhancements to QSIM, but you must use SIM for transistor designs
- has problems simulating some circuit constructs (dynamic logic, n-channel pullups, ram cells, sense amps)
- use switch to bring in netlists that would actually sim
- tweak capacitances to get sim behavior the same as spice
- hnl2spice was good for quick&dirty spice deck
- problems with system modelling (hard to do best/worst case system simulation)
- back annotation was a problem: physical and schematic hierarchies didn't match exactly (load extracted netlist, dump caps, make 'set cap' file, rename nodes, etc.)

BROWSER

- would like to have multiple tools sessions simultaneously accessing a single database
- sometimes loses things you check in (retains the old version, but the new version is gone...)
- sometimes you must hand edit the checkout file
- why not use SCCS instead?

GENERAL STUFF

- needs lots of virtual memory so Mainsail won't use its pager (else this causes an order of magnitude slowdown)
- each rev of tools increases capabilities by 10x, but designers needs increase by 50x...
- as long as files are text based, awk+sed+perl+csh can make anything work (e.g. netcompare inits, makeschematic hacks, sim backannotation)
- performance degradation of ALL tools seems to be exponential, but limits are *a priori* unknown
- by the way, what tools does VLSI Technology use for their full-custom designs?