ASIC Flow Engine for Timing Closure (AFETC),

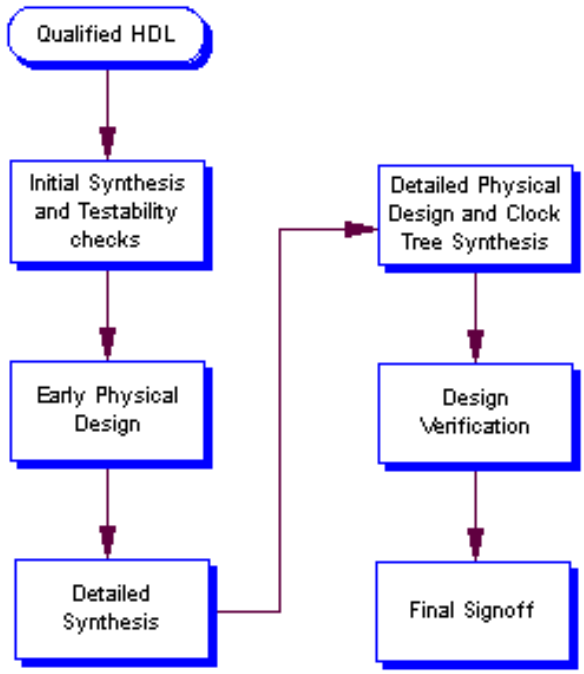
a Makefile Generator to Automate the Design Budgeter Methodology

Thomas D. Tessier,
t2design Incorporated
Marvin L. Anderson, StorageTek

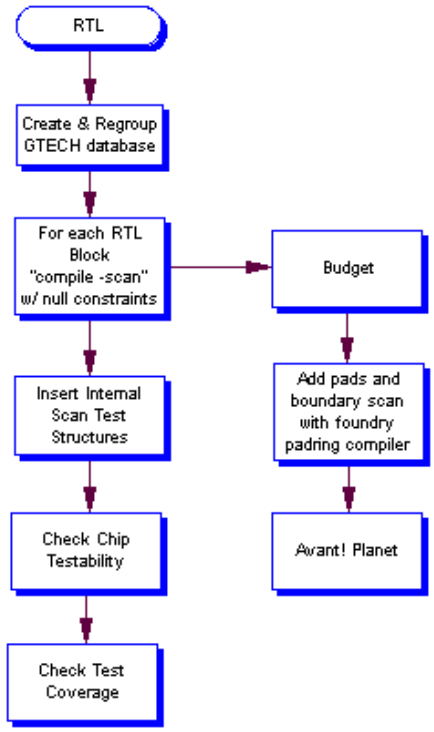
Discussion Outline

- Why Another Makefile Generator?
- What did we automate?
- Why bite off such a large piece of pie?
  - RTL Source
  - Design Budgeting
  - Floorplanning
  - Timing Closure
- Lessons Learned!

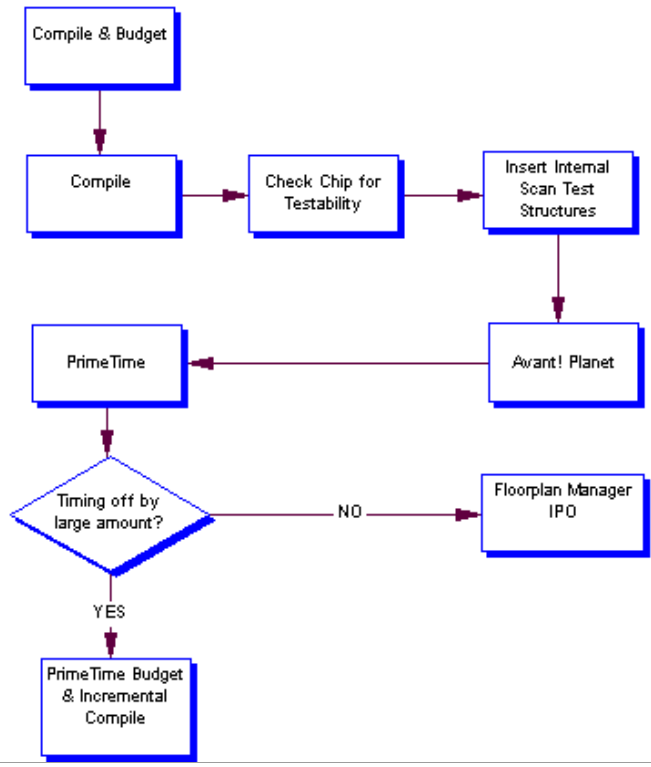
What we Automated?



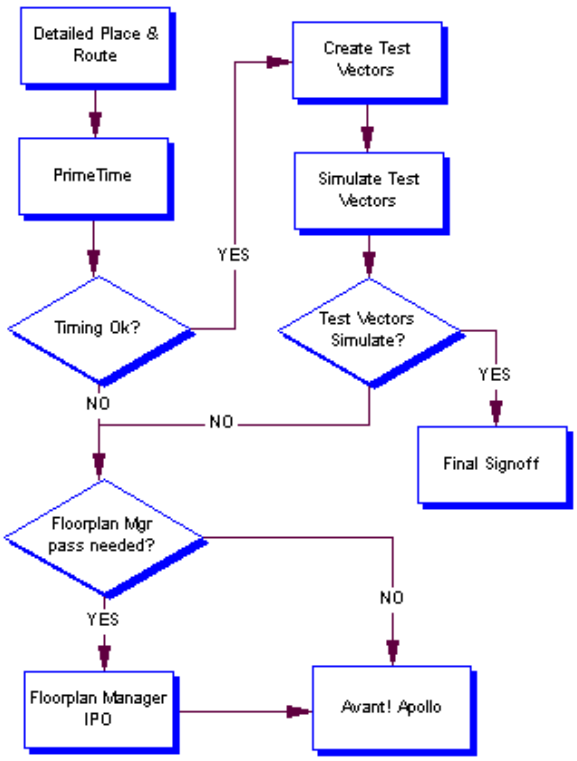
RTL to GTECH


RTL
Create & Regroup GTECH database
For each RTL Block "compile -scan" w/ null constraints
Budget
Insert Internal Scan Test Structures
Add pads and boundary scan with foundry padring compiler
Check Chip Testability
Avant! Planet
Check Test Coverage

GTECH to Gates



Layout to Timing Closure



Why another Makefile Generator?

- Using Design Budgeter thus many iterations through Design Compiler.
  - Bottom-up constraints were not going to be feasible?
- Using Avant! P&R tools
  - Exploit in-house layout tools and experts.
  - Use the tools to their fullest advantage.
- Isn’t every environment unique?
  - “Not Invented Here”

Why Bite off Such a Large Piece of the Pie?

- Desired to control RTL to finished gates.
- Desired to provide design engineers more control and access to the layout process.
- Desired to be able to reach timing closure.
  - Over the wall net list was not going to be sufficient for current 0.20u designs and in the future.

RTL Source Reference

- We wanted a method to control which RTL version built the gates. It was important to be able to recreate at any version.
- Discovered that RTL processing was a very small part of the problem.
- Desired to give the engineers early visibility into the synthesis process and if RTL built good designs.

Floorplanning, Wireloads and Timing Closure

- What do you learn and what can you use from the Floorplan?
  - Large buses and interconnect blockages.
- Wireloads; do we really want them and when do we create them?
  - Early for synthesis and post route.
- Timing Closure is getting to an acceptable margin that the design group feels confident they can sign off on the ASIC.
- Timing closure involves many tools; we desired repeatability of our results, thus we wanted even this flow automated.

Lessons Learned

- What have we learned from developing a tool of this size?
  - Scripting takes time and experimentation. We could not have developed this tool until we proved the flow. We would not have developed the flow without a tool to automate it.
- Logical vs. Physical.
- Physical Tools Don't Always Give you what you Need!
- The Make Circle.
- Successful Timing Closure.
- Why Don't we have a Central Script Location?

Logical vs. Physical

- Early in the development we kept the Logical design team focused on Logic Design and Verification.
- Physical structure was handled with the ungroup commands.
- Our designers wrote small modules which did not exploit the tools capacities.
- Keep Logical and Physical close. Only use the ungroup command to make blocks bigger to utilize the greater tool capacity.

Physical Tools Don't Always Give you what you Need (Wireloads).

- Wireloads were desired early to give Synthesis knowledge of the length and loading of wires.
  - Avant! Planet wrote out table form.

    Developed a method to read in the wireloads in memory and to keep them applied throughout optimization.
  - Avant! Apollo provides very complete data as long as you used the resulting SPEF, PDEF, "set_load" and Verilog.

    Use FPM "create_clusters" command.
- Making wireloads stick?
  - Beware, Budget_shell constraints replaced the wireload model and selection. Need to force DC to use the custom wireloads.

Physical Tools Don't Always Give you what you Need (Location Based Opt).

- Location Based Optimization, used for timing closure, has a very stringent data requirements.
  - Avant! Planet did not provide everything.

    After reading in SDF file, write it back out from DC, reset timing then read it back in fixed most of the problems we were having.
  - Avant! Apollo provides very complete data as long as you used the resulting SPEF, PDEF, "set_load" and Verilog.

    You must use the layout tools Verilog, don't even try to link to your source DB.
- Be prepared to handle Gbyte files and to use Gbyte files as temporary's
  - Everything takes time with big files.

The Make Circle and other Scripting Issues

- Managing Inputs to Make seems straight forward.
  - Beware of creating intermediate files which Make believes are inputs. It will try to find its sources and refuse to build.
- Make within Perl can you say Slash (\) and (\\)?
- Tcl within Perl, can you say confused; does the variable have a $ or is it { or [?
- Variable expansion always a problem when using multiple languages.
- A generator is responsible for the repeatability of the process.

Successful Timing Closure?

- Cooperation of many Disciplines
  - RTL designers writing good designs

    Linting, early synthesis and an understanding of timing always helpful
  - Synthesis Engineer understanding both the RTL (logical) and the Chip (Physical) design requirements.
  - Layout Engineer, willingness to try some of the Synthesis engineers wild ideas.
- Willingness to experiment
  - You must start practicing your synthesis and timing closure loops early.

Why we Don't have a Central Script Location?

- Desired independence of each ASICs which used the tool.
- Minimum Impact if an ASIC team took a new approach.
- Only one tool to copy over for all the support scripts and template scripts.
- Central location would require testing on existing ASICs. The overhead was not wanted.

Conclusions and Future Directions

- Repeatability is the key to any tool.
- Extensibility is the next most important feature for the tool writer to consider.
- Make provides a vehicle to generate repeatability.
- Should we have a GUI?
- Investigation of Synopsys ACS, how much does it do for us? Can we extend it to do what we need?
- Without Data Management repeatability is not possible, see Steve Remme's presentation "Migrating a Large Multi-team Simulation Environment" next.