



Ground Truths

MARCO GSRC Workshop
September 24, 1999

1



Review

- ◆ **Theme: “Calibrating Achievable Design”**
- ◆ **Ground Truths: facts and data, composable into inference chains and technology extrapolations, predictions of what matters**
- ◆ **Bookshelf: “Publication medium for implementations and supporting infrastructure” -- removes barriers to entry, evaluation, and adoption for research at leading edge**
- ◆ **Metrics: Value of tool, innovations unclear if tools are incorrectly applied -- the design process itself must be instrumented, diagnosed, and optimized**
- ◆ **“Life is short!”**

2

Agenda



- ◆ 9:10-9:50 A perspective on ground truths (D. Kirkpatrick)
- ◆ 9:50-10:30 GTX system and ground truths updates (ABK)
- ◆ 10:30-10:45 (break)
- ◆ 10:45-11:00 Group Discussion on Ground Truths
- ◆ 11:00-11:30 Bookshelf update (ABK and/or Igor Markov)
- ◆ 11:30-noon Metrics update (S. Fenstermaker, B. Thielges, ABK)

3

What is a “Ground Truth” ?



- ◆ A ground truth is an established fact or data point
 - anchors the **process** of bounding the achievable envelope of design
- ◆ Can be with respect to:
 - manufacturing process, materials, physical phenomena
 - specific CAD optimizations of circuit topology/embedding
 - system architecture and packaging
- ◆ Properly extrapolated via:
 - "inference chains"
 - response surface analysis, parameter optimization
- ◆ Drives the EDA community's vision of future design issues :
 - current distribution, inductance extraction, UDSM testing, ...
 - fundamental limits

4

Why Seek Ground Truths?



- ◆ **Improved understanding**
 - which techniques / technologies will achieve what, which won't
 - which issues deserve attention, which don't
 - which models are critical to get right, which aren't
- ◆ **Foundation for roadmaps and predictions**
 - process, system design, design technology
- ◆ **Improved focus, reduced “solution space”**
 - in what areas will new design technology and methodology have the greatest impact ?
- ◆ **Reduce the design productivity gap, if only by reducing wasted effort and attention**
 - even storing, cataloguing ground truths is immensely valuable!

5

Example Ground Truths



- ◆ How many clock cycles to get across die in given technology?
- ◆ What portion of power dissipation will be useful/useless in a given technology? (Synopsys)
- ◆ Do (BIST, functional vector sets) really detect a chip's failing timing spec? (Tim Cheng)
- ◆ Which fault model (resistive short/open, delay faults) better captures the behaviors of DSM defective devices? (Tim Cheng)
- ◆ Will supply-noise jitter force new clock distribution styles?
- ◆ What design tradeoffs must be made to maintain reasonable supply currents?
- ◆ At what geometries, supply voltages will domino lose most advantages over static CMOS?

6



GTX: A System for Ground Truths and Technology Extrapolation

7



Related Work

- ◆ **Estimators of chip size, power dissipation, clock frequency**
 - SUSPENS (Bakoglu, Stanford)
 - Sai-Halasz, IBM
 - GENESYS (Meindl, GaTech)
 - RIPE (Rose, RPI)
 - BACPAC (Sylvester, UCB)
- ◆ **Limitations**
 - **cannot easily validate, explore the model**
 - static, fixed studies (i.e., “inference chains”)
 - alternative rules or relations generally not provided (e.g., can’t check sensitivity to choice of RC delay model)
 - **not extensible in nearly any sense (without going back to developers)**
 - **not portable**
 - users unlikely to enter sensitive data into an offsite application
- ◆ **AI literature: general frameworks for constraint propagation, parameter optimization and trade studies**
 - Rockwell Design Sheet, TkSolver, UniCalc

8

Attributes of GTX



- ◆ **Flexible sets of rules, parameters; flexible rules chains**
 - user can add/modify at runtime
 - automatic rule discovery (?)
- ◆ **Flexible, user-specified studies (specific use of rules chains)**
 - optimization capability is planned
 - flexible display / plotting options
- ◆ **Portable, shippable application**
 - wxWindows / SciChart on Linux, NT, Solaris; C++ (/Java)
- ◆ **Comprehensive: sensitivity to particular modeling choices**
- ◆ **Quality: seek to include as much “best practice” as possible**
- ◆ **Understands concept of “optimized” (not just random models)**
- ◆ **Scales to use of “tool rules” (estimators can use Bookshelf!)**
- ◆ **Common naming convention, initial rules modules, ... in place**

9

GTX Architecture



- ◆ **Two main components**
 - GUI
 - Engine
- ◆ **Separation allows:**
 - client-server architecture with remote GUI
 - potential for web-based GUI
 - separate development
 - development of alternate GUIs for different environments/goals

10

Engine Architecture



◆ Rules and Parameters

- represent domain-specific knowledge
- parameters describe the system being modeled
 - cache size, doping density, ILD permittivity, Rent parameter, etc.
- rules describe relationships between parameters
 - $V = I \times R$; $t_{\text{stg}} = r_{\text{dr}} \times (c_{\text{int}} + c_{\text{load}}) + r_{\text{int}} \times (c_{\text{int}} / 2 + c_{\text{load}})$; etc.
- together, rules and parameters may be viewed as directed hypergraph
 - parameters = vertices, rules = hyperedges
- multiple rules may compute the same parameter by different relationships among possibly different parameters

◆ Rulechain (RC)

- connected, acyclic subhypergraph of rules (no two rules compute same param)
- inputs to the RC = input parameters to rule in the chain, not computed by any rule in the chain
- outputs from the RC = output parameters of any rule in the chain not used as input to any rule in the chain

11

Rules and Parameters -

“ASCII” and “Code” Nodes



◆ ASCII rules and parameters specified in ASCII files

- system reads files and adds them to the hypergraph
- each rule contains a parse tree (rules essentially interpreted)
- allows runtime addition/update of rules/parameters
- supports user-defined datatypes
- supports using other executables (even wrappedCAD tools!)
 - e.g., grammar extension to allow specification of path, command line, format of expected result
 - “rule” = single-node parse tree (command line, order of arguments)
 - system must execute shell command, get data from executed command
 - issues: OS portability (\ vs. -), finding executables, custom wrappers (“make tools look like rules”), tool portability

◆ Table rules also allowed (with specified interpolation method)

◆ Code rules and parameters specified in C++ code compiled into GTX

- allows more complex rules/datatypes

◆ GTX supports all types; nearly all rules are ASCII

12

Evaluating a RuleChain



- ◆ **Given:** RC with (possible multiple) values for each RC input
- ◆ **Output:** (set of) value(s) for all RC outputs
- ◆ **Apply rules in topological order**
 - outputs of one rule used as inputs to the next
- ◆ **If inputs have multiple values, iterate through cross-product**
 - produce a set of output values
 - display results according to user's specification of the study (find minimizer s.t. constraints, plot a response surface, ...)
- ◆ **Possibilities for more intelligent evaluation**
 - optimal reuse of intermediate computation results
 - approximate evaluation via hierarchical processing, interval math

13

Team Status



- ◆ **GTX Engine and GUI**
 - Mike Oliver (design, implementation)
 - Andy Caldwell and Igor Markov (design)
- ◆ **GTX Rules**
 - Farinaz Koushanfar, Hua Lu, Dr. Dirk Stroobandt
- ◆ **Theme PIs**
 - Dai: models of block packing
 - Cong: executable “rules” for BIS/WS interconnect optimization, via Dr. Wangning Long (based on TRIO package)
 - Keutzer: new student to take over device / scaling module ?

14



Soft Block Packing Based on BSG

Linda Huaizhi Wu
Wayne W.-M. Dai
University of California, Santa Cruz

15



Soft Block Packing

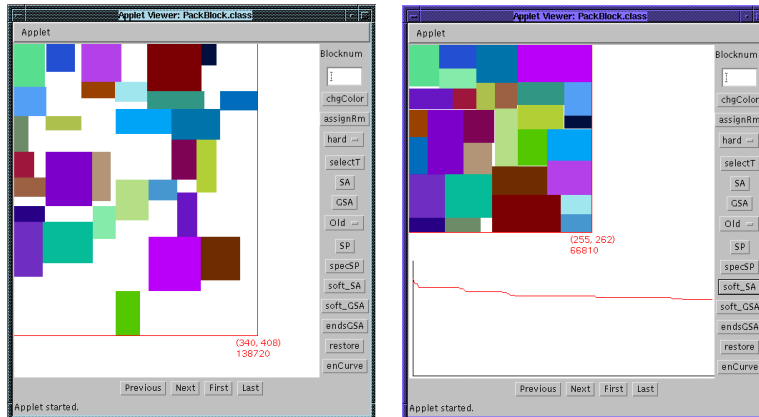
- ◆ **Soft block packing problem is to find a packing topology as well as block shapes such that the total area is minimized.**
- ◆ **The optimization of block shapes for a given packing topology is obtained by alternately reducing the overall height and width.**

16

Results



- ◆ A Result of Soft Block Packing Using Simulated Annealing with Cost Function: Cost = Total Packing Area

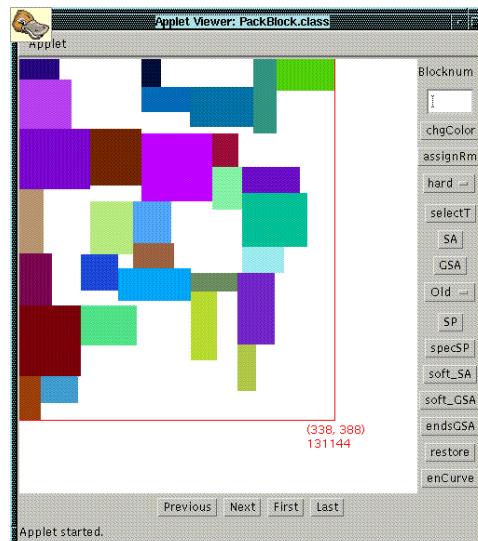


17

Hard Block Packing



- ◆ An Animated Process of Hard Block Packing Using Simulated Annealing with Cost Function:
Cost = Total Packing Area + Total Wiring Length

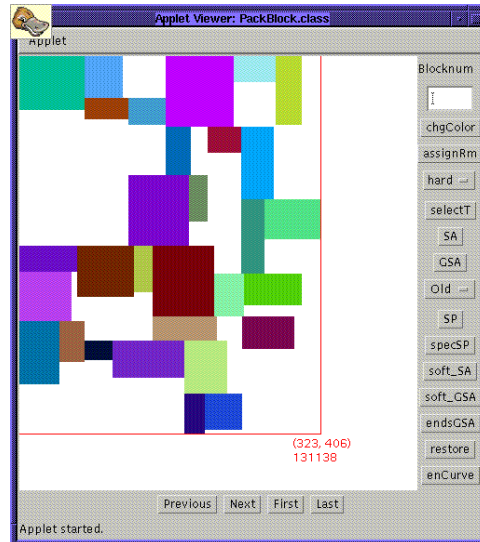


18

Soft Block Packing



- ◆ An Animated Process of Soft Block Packing Using Simulated Annealing with Cost Function:
 $\text{Cost} = \text{Total Packing Area} + \text{Total Wiring Length}$



19

Resolutions of Ground Truths



- ◆ Anticipate delivery of systems / white papers
- ◆ Maly
 - “optimum design strategy from manufacturing cost point of view” (e.g., monolithic vs. hybrid, 180nm vs. 150nm, etc.)
 - working with Jan (what-if studies for various technologies), IMEC
 - developing model for 250nm (10+ process recipes), will distribute within GSRC by December
- ◆ Kahng
 - “sensitivity analysis of the ITRS cycle time projections”
 - “wireability analysis and interconnect process optimization for multi-terminal nets, repeaters and explicit vertical interconnect”
 - “impact of 1- and 2-exposure altPSM on speed, layout density”
 - “scaling projections for jitter and skew (wrt noise, variability)”
- ◆ (Sponsors’ requests here)

20

Wireability Analysis



◆ Observations

- a priori WL estimates (e.g., Donath-type) do not take physical metal layers into account (unlimited wiring capacity assumed)
- choice of wire pitches at layers independent of considering wire lengths on these layers
- effects of vias and repeaters on WL left unstudied

◆ Given:

- # layers (or, layer types)
- wire pitch at each layer (at each layer type)
- estimated wirelength distribution
- **Which interconnects will be routed on which layer?**

◆ Given:

- allowed WL intervals for each layer type
- **How many layers of each type are needed to handle all wires ?**

21

Wireability Analysis



◆ Given:

- # tier types, wire pitch at each tier type
- estimated wirelength distribution
- interconnect dimensions and electrical properties
- **How many layers of each type are needed to accommodate all wires such that the max-length wire at each tier has same delay for all tier types?**

◆ Given:

- estimated wirelength distribution, and maximal delay
- **What is the optimum number of tiers, and what are the optimal interconnect dimensions for each tier?**

◆ These questions are now addressed in GTX via code rules

◆ Still to do: improved model of vertical interconnect

22

Ground Truths Discussion



- ◆ **Key functionality for GTX**
- ◆ **Key ground truths to investigate**
- ◆ **(How to avoid boiling the ocean...)**
- ◆ **Volunteers for ownership or advisory role for modules**
- ◆ **How might industry partners/sponsors wish to participate?**
 - e.g., Shehkar Borkar, Sani Nassif, Lars Liebmann, Paolo Gargini, ...
 - what is view of EDA vendors ?