Fast Simulation and Equivalence Checking using OpenAccess

University of Michigan

Kai-hui Chang, **David A. Papa**, Igor L. Markov and Valeria Bertacco

{changkh, iamyou, imarkov, valeria}@umich.edu

Outline

- IWLS 2006 Programming Challenge
- OpenAccess Gear
- Fast simulation
  - Oblivious vs. Event Driven
- Equivalence checking with simulation signatures
- Incremental Verification
- Custom vs. Native implementations
- Graphical User Interface extensions
- Plug-In interface

IWLS 2006 Programming Challenge

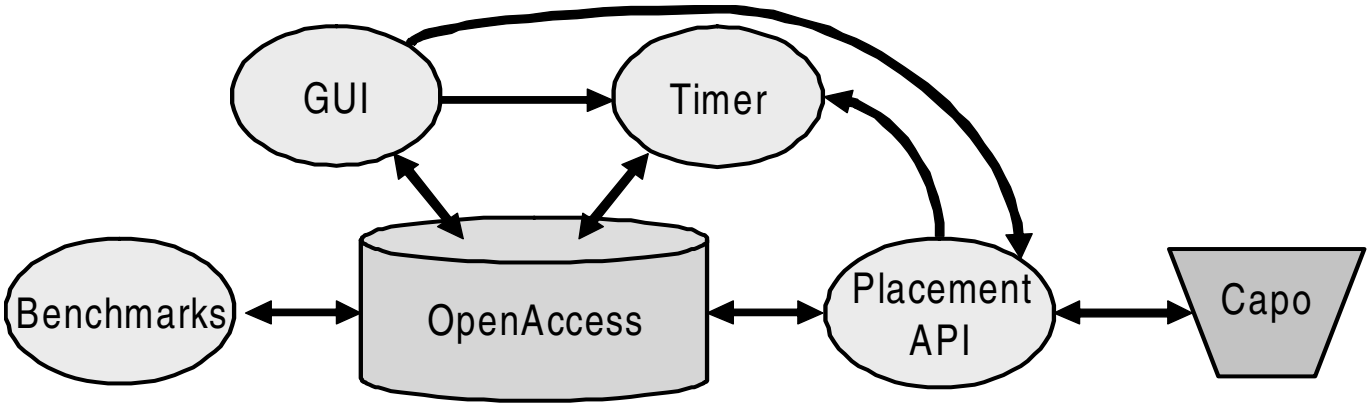
- Logic optimization student programming competition
- Must be implemented on OpenAccess and should make use of OAGear infrastructure
- Judged according to correctness, efficiency, importance, design, coding style, etc.
- 1st place was given to two teams
  - Sat Sweeping package
  - Fast simulation and equivalence checking
- Both entries are now part of OAGear

OpenAccess Gear

- Release useful tools and libraries to enable research
- Make OpenAccess a useful platform for academia
- Provide common infrastructure for research and benchmarking
- Adopt an open source development model
- Initiated and supported by Cadence Design Systems

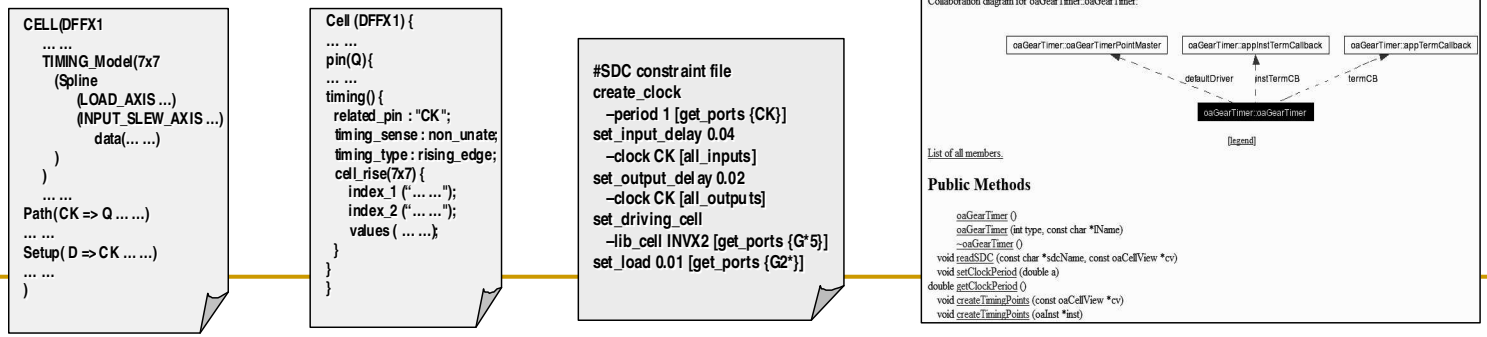
Early OAGear Overview

- **Focus on four main components**
  - **GUI: Layout and Schematic Viewer**
  - **Static Timing Analysis**
  - **Generic Standard Cell Placement Interface: Capo API**
  - **Benchmarks in OpenAccess Format**



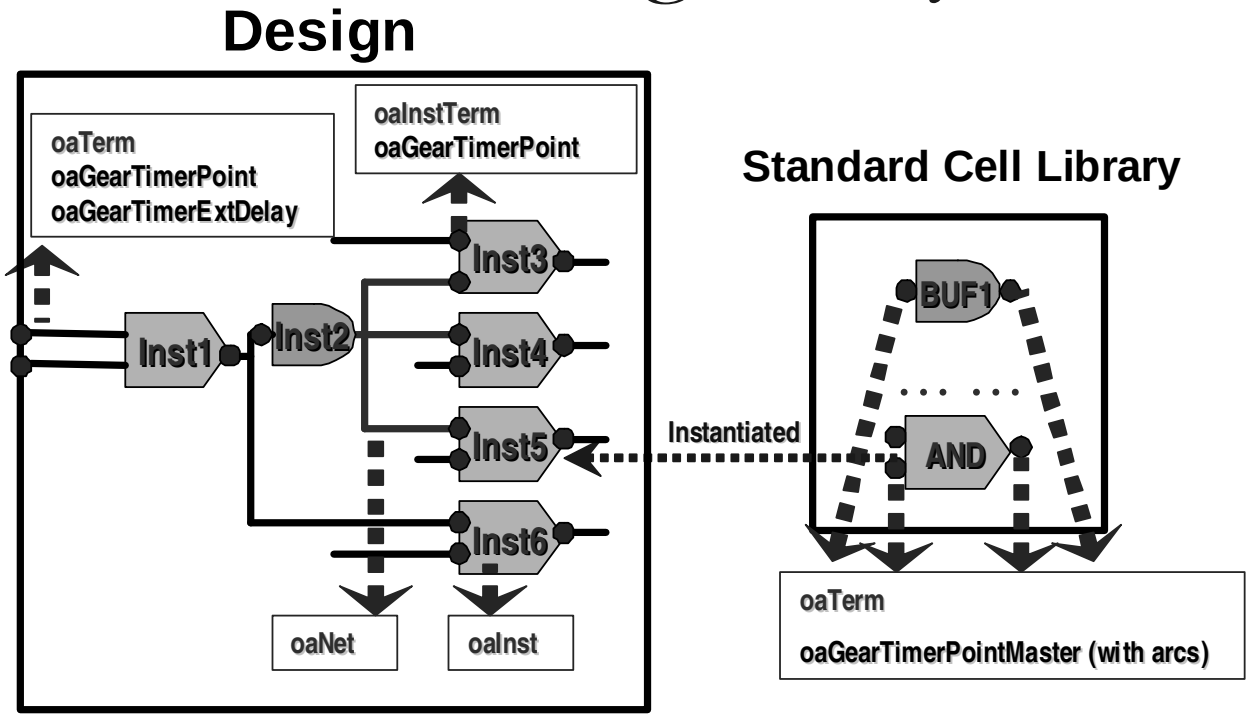
Static Timing Analysis: OAGear Timer

- **Built on OpenAccess for integration into other tools, e.g. placement**
- **Two modes: Full timing analysis and incremental timing analysis**
- **Different models for wires: No wire delay, bounding box model; can be extended easily to more accurate models**
- **Library formats: Cadence .tlf and Synopsys .lib**
- **Timing constraints: Subset of .sdc constraints**
- **Standardized timing reports**
- **Detailed documentation**



.tlf .lib .sdc Documentation

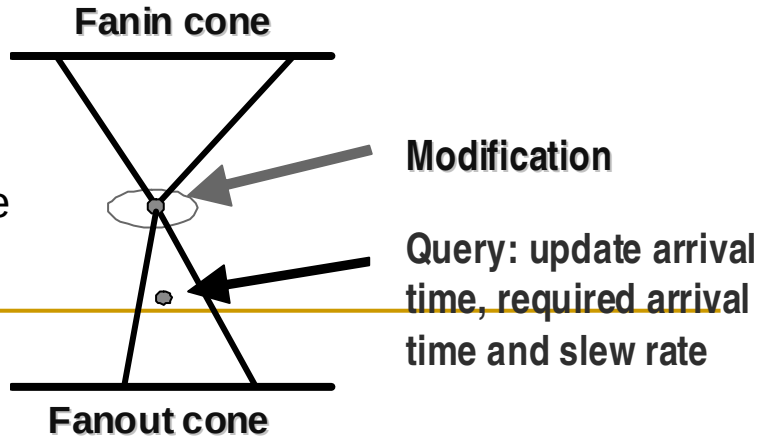
Static Timing Analysis: OAGear Timer



The timing information is stored using the OpenAccess extension mechanism (oaGearTimerPoint, …)

Incremental timing analysis

- **When a modification occurs:**
  - **Mark the** required arrival time **of nodes in the** fan-in cone **invalid**
  - **Mark the** arrival time **of nodes in the** fan-out cone **invalid**
- **Later if there is a query, update the timing information**



Capo Placement API

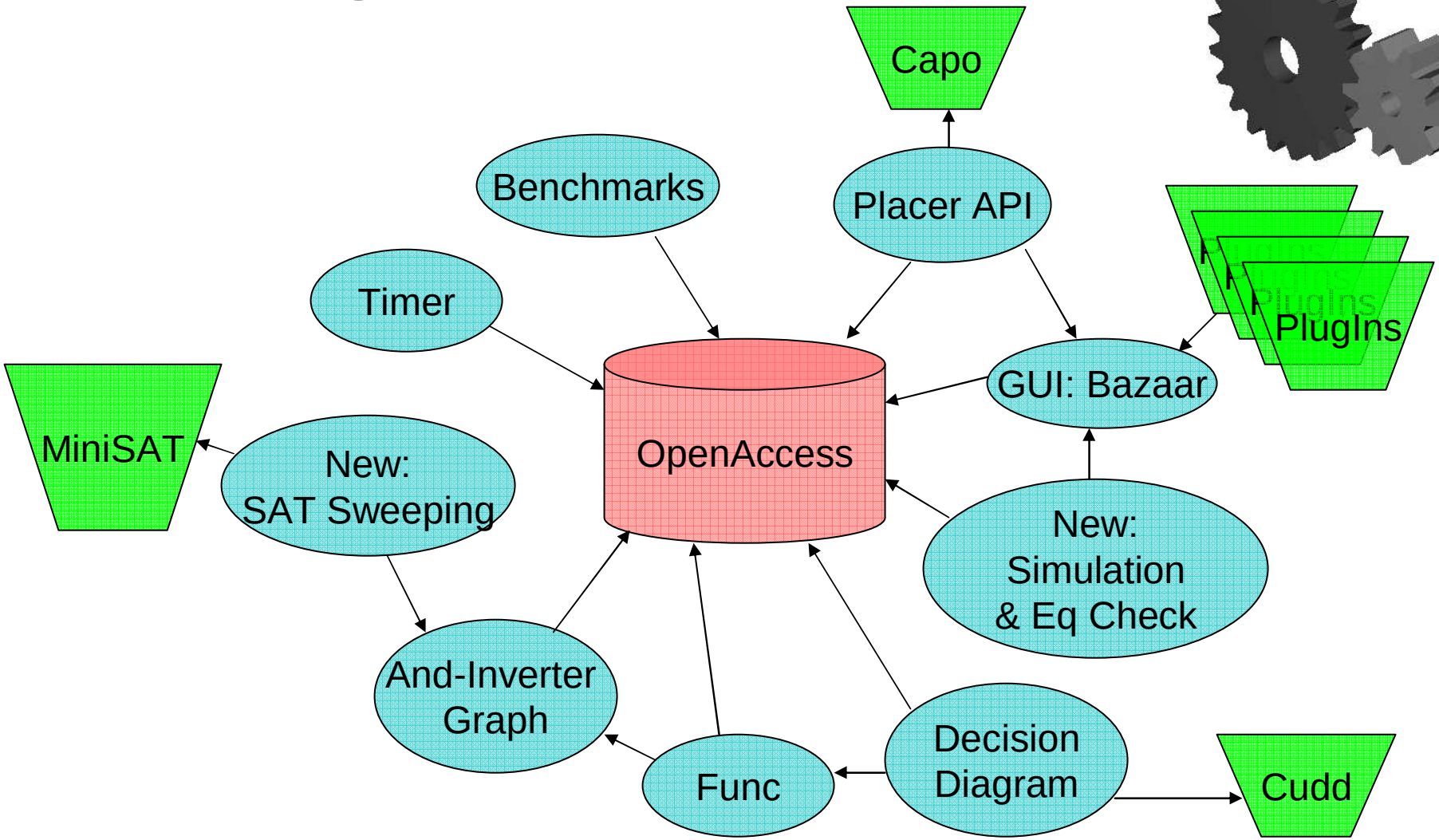
- Open source placement tool
  - Maintained at U. of Michigan
  - http://vlsicad.eecs.umich.edu/BK/PDtools/
- Extended to use OpenAccess by the OAGear CapoWrapper package
  - Reads design data directly from OpenAccess
  - Builds appropriate Capo data structures in memory
  - Reads back Capo data structures and writes results to OA
- Example of integrating large mature programs
  - Porting to a native codebase would require extraordinary effort

GUI: Bazaar

Technical Capabilities

- Easy to read and extend, built on Qt and OpenGL
- In the style and spirit of the OpenAccess standard
- Layout Editor displays block domain design data directly from database
- Schematic Editor displays module domain design's logical connectivity
- Controller operates Capo API for on-demand placement
- Fast OpenGL rendering scales to very large designs
- "oaRegionQuery" accesses only relevant portions of the design

Growing Picture



Fast Simulation

- Bit-parallel simulation
  - 32 or 64 patterns simulated simultaneously
- Special cases for common gate types
  - Compiler can use CPU instructions to implement AND, OR, etc.
- Levelize the circuit for faster topological traversals

Simulation Algorithms

- Event-driven algorithm
  - Evaluates only gates with events
  - Suitable when the number of events is small
- Oblivious algorithm
  - Evaluates all the gates
  - Avoids overhead of event scheduling
  - Suitable for random simulation

Equivalence Checking With Simulation Signatures

- Easy to disprove equivalence with counter-examples from simulation
  - Signatures mismatch => not equivalent
  - Signatures match => need more testing
- SAT-based equivalence checking is performed when signatures match
- Counter-examples are returned
  - To help understand the discrepancy

[J. Zhang et al. “Simulation and Satisability in LogicSynthesis”, *IWLS* 2005]

Incremental Verification

- User specified set of gates to EQ check
  - Still need to add GUI support for this feature
- Use fast simulation to define Similarity Factor between two netlists
  - (Matching signals) / (number of signals)
  - Signals are matching if simulation signature appears in both circuits
  - Small Similarity Factor means potential discrepancy
  - Reported in the GUI equivalence checker

Custom Vs. Native Implementation

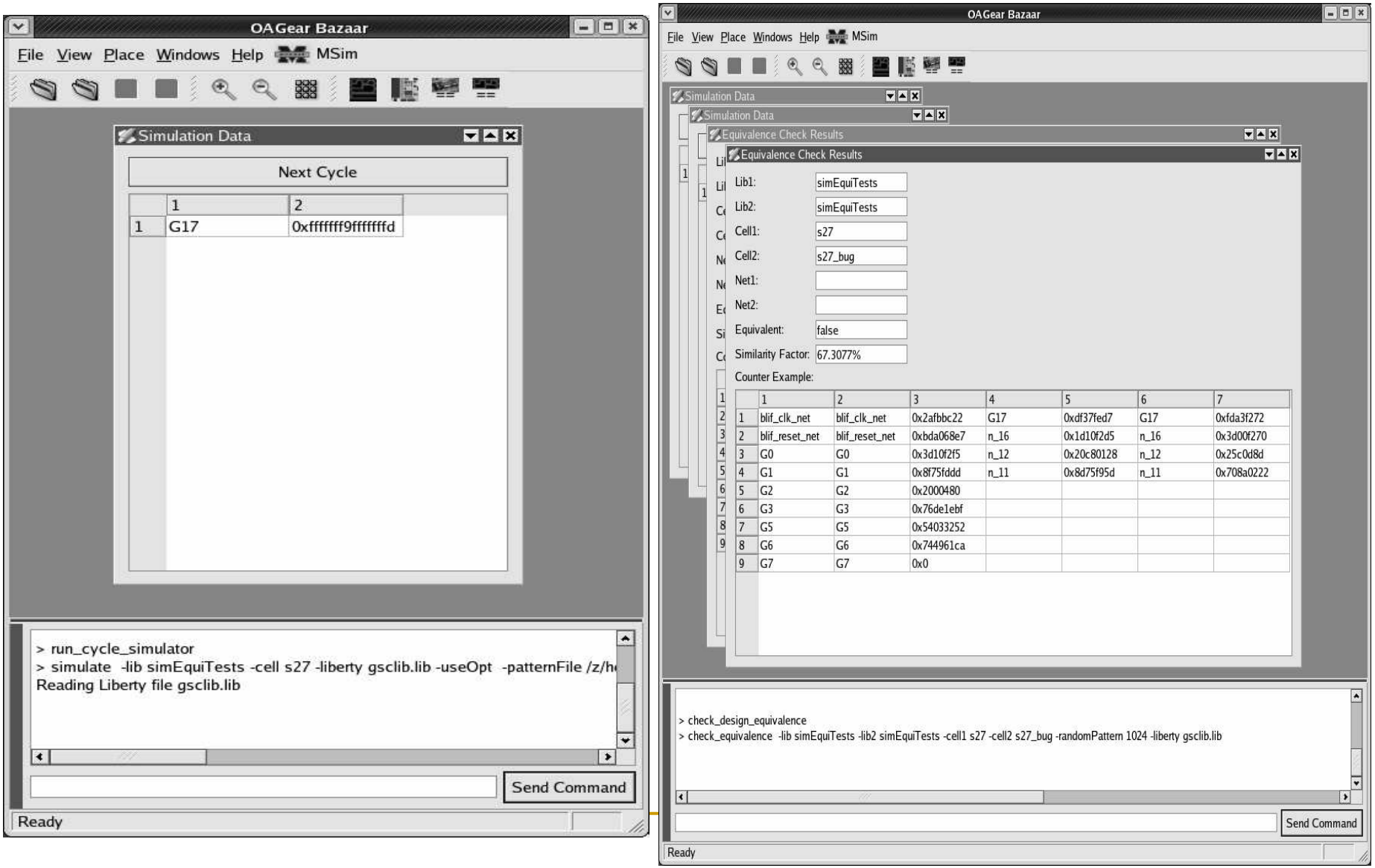
- Simulator originally used custom data structures and file I/O
- Ported it to run natively on OpenAccess
- Observed significant slowdown in native impl.
  - Primarily due to oaAppDef lookup time
- Spent significant effort optimizing native impl.
  - Converted oaAppDef uses to std::hash_map<oaNet*>
- Unable to match custom impl. performance

Custom vs. Native Experimental Data

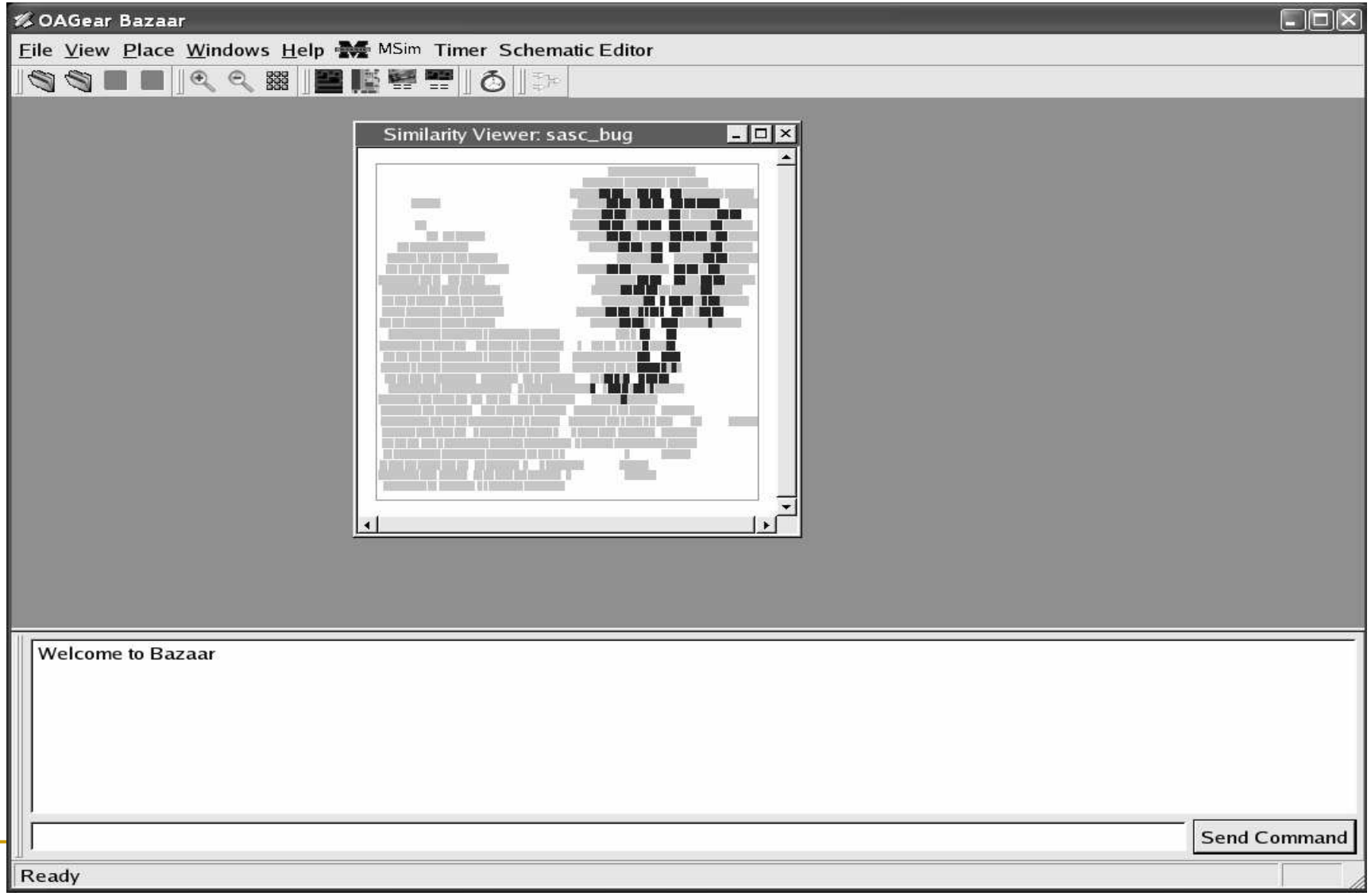
- Asymptotic improvement over original simulator
- Both new simulation algorithms scale linearly
- Both scale to realistic circuit sizes
- Custom data structures perform better for large inputs

Benchmark	Gate count	Simulator runtime (sec)			EQcheck runtime (sec)
		OAGear	Our simulators		
		orig.	custom	native	
s27	19	9.8e0	0.4	0.4	0.3
s344	132	4.0e1	0.4	0.7	0.6
s1196	483	9.5e1	0.5	1.7	1.6
s15850	685	5.0e3	0.6	2.0	1.8
s9234_1	974	2.4e3	0.7	2.9	2.8
s13207	1218	1.7e4	0.8	3.3	3.4
s38417	8278	3.0e5	2.0	21.0	1.4e2
vga_lcd	124031	time-out	20.2	3.3e2	2.5e4

Graphical User Interface Elements

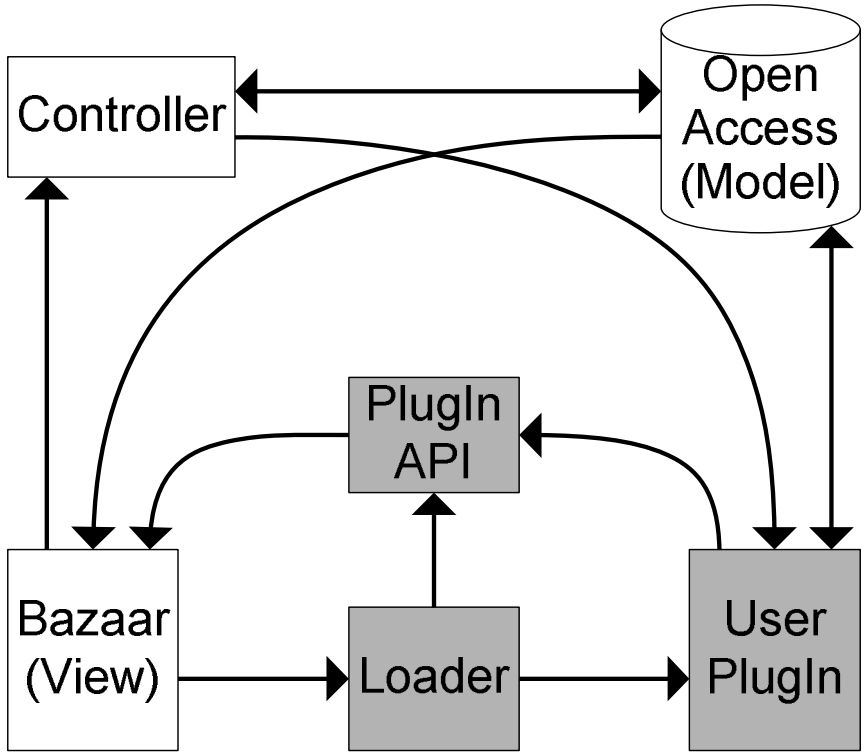


Similarity Layout View



Plug-In Interface for OAGear Bazaar

- Encapsulates Bazaar's interface
- Greatly simplifies extending Bazaar
- Facilitates dynamic loading of user code
- Decouples user code from Bazaar



Conclusions

- Identified a major component of OAGear with poor scalability
- Wrote new logic simulation engine using different algorithms
  - Reduces runtime by up to 100 times in our experiments
  - Asymptotic improvement
- Leveraged simulator to speed up equivalence checking
- Defined new metric of circuit similarity useful in incremental verication and debugging
- Extended OAGear's graphical user interface, Bazaar
  - Implemented and evaluated several use-cases
  - Designed new infrastructure for creating user plug-ins

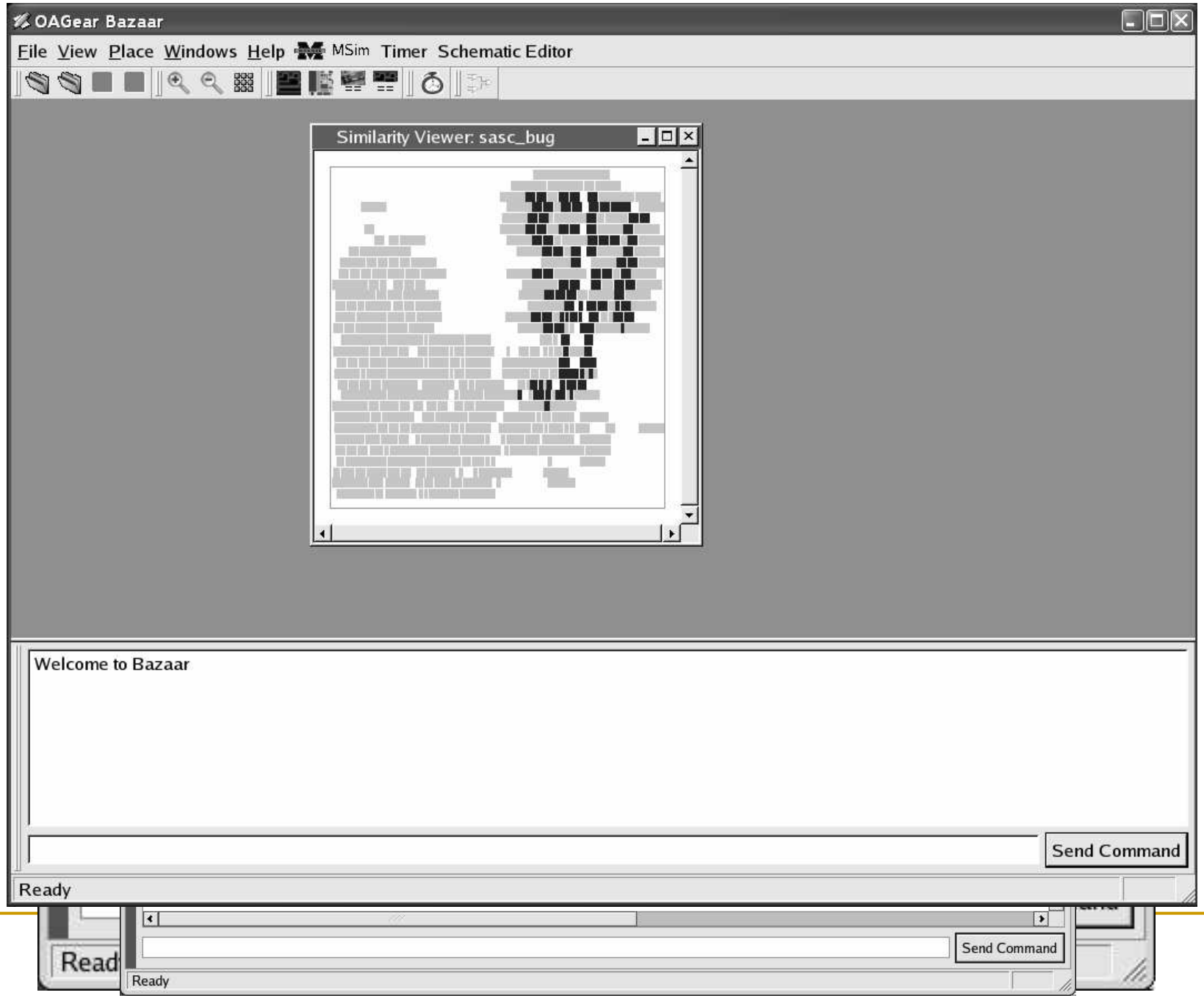
Future Work in OAGear

- Convert existing GUI tools to PlugIns
  - Layout Editor
  - Schematic Editor
- More PlugIns!
  - Waveform viewer for our logic simulator
  - User contributions… (hint, hint!)
- Develop a communication mechanism between PlugIns
- Further integration of OAGear Tools into the GUI
  - Buffer Insertion, Timer, Sat Sweeping, etc.
- More complete Tcl API for OAGear utilities
- Possibly an OAGear Router
- Ease of use improvements

Thank You!

Questions?

More screen shots…


OAGear Bazaar
File View Place Windows Help MSim Timer Schematic Editor
Similarity Viewer: sasc_bug
Welcome to Bazaar
Send Command
Ready
Send Command
Ready