LOSP SYNTHESIS SYSTEM: EMPIRICAL RESULTS

William Bricken

July 2004

Boundary logic is a new method of designing, optimizing, and verifying semiconductor circuits. The advantages of boundary logic include more succinct representations and more powerful transformations. These advantages apply to many aspects of semiconductor development, potentially impacting the efficiency and quality of logic design during specification, synthesis, verification, and technology mapping. To demonstrate these pragmatic advantages, the Losp implementation of boundary logic was applied to some of the most difficult logic problems found in commerce, specifically logic optimization of large semiconductor circuit designs.

SUMMARY OF EMPIRICAL RESULTS

Compared to the leading Electronic Design Automation (EDA) commercial tools, the Losp implementation of logic synthesis has been measured to improve a diversity of designs in these ways:

ASIC DESIGNS

- 17% less area when delay is held constant
- 14% lower delay at a cost of 20% more area
- 10% lower delay at a cost of 10% more area

FPGA DESIGNS

- 25% fewer logic gates
- 34% less routing

LOGIC SYNTHESIS

> "Research in logic synthesis is a very satisfying enterprise because improvements in algorithms can immediately translate into smaller or faster circuits, and smaller and faster circuits have substantial commercial impact."
>
> [Devadas, *Logic Synthesis*, p.xiv]

> "The need of practical synthesis and optimization algorithms for multiple-level circuits has made this topic one of utmost importance in CAD [Computer-Aided Design]."
>
> [DeMicheli, *Synthesis and Optimization of Digital Circuits*, p.344]

Semiconductor circuitry consists of millions, and recently tens of millions, of transistors configured as a logic network (called a *netlist*) and connected by microscopic wires that conduct electric currents. *Logic synthesis* is the optimization of logic, wires, and other specified criteria in a semiconductor design.

Logic synthesis is a technical, highly mathematical field. The EDA industry is fragmented over dozens of commercial products in order to address the multiple problems of design. While semiconductor chips have grown rapidly in size and complexity, EDA software has lagged behind, still relying on approaches and techniques from a decade ago. Boundary logic, as implemented in Losp, provides a comprehensive formal framework for updating and improving significant sections of the $4 billion EDA/CAD market.

Losp Logic Synthesis System

Losp is a *logic synthesis software system* that provides an integrated and complete approach to network transformation, including area and delay optimization. Losp takes a netlist circuit design as input and returns a functionally equivalent netlist with improved delay and area performance. Losp performance improvement can be applied at any step of the design tool-chain for which a formal description is available, and is general across all types of designs.

The Losp system consists of hundreds of boundary logic algorithms, and dozens of parameterizations. The netlist manipulation algorithms and all core reduction algorithms are complete, validated, and extensively tested.

Losp synthesis improves upon the current state-of-the-art by incorporating innovative mathematical techniques and algorithms that are unlike any found in other commercial products. The Losp logic synthesis system is a commercial prototype, and include several capabilities other than synthesis, such as logic partitioning and layout, equivalence checking, library mapping, and abstraction tools to reduce complexity. The Losp implementation is expected to improve significantly with increased development effort.

One unique capability of Losp, for example, is *steerable Boolean optimization*, the capability to dynamically identify circuit designs that meet desired criteria such as area and speed. Commercial tools offer optimization, but require the designer to iterate many times, making fine-grain adjustments to reach a preferred configuration. In comparison, Losp permits the designer to steer the design optimization process in real-time, exploring choices without iteration.

COMPARATIVE PERFORMANCE DATA

Losp performance was measured and compared to the best performance of leading EDA industry tools on a diversity of widely used industry benchmark designs. The benchmarks include common design modules, such as error correction, lookup tables, arithmetic-logic unit (ALU), adder/comparator, and digital signal processing (DSP).

Performance comparison is presented for several optimization capabilities:

- *Optimization of ASIC designs*
    - -- area reduction, regardless of delay
    - -- delay reduction, regardless of area
    - -- combined area and delay reduction
    - -- area and delay trade-off

- *Optimization of FPGA designs*
  - -- logic resources
  - -- routing resources

OPTIMIZATION OF ASIC DESIGNS

Benchmark ASIC designs were selected from the ISCAS'91/MCNC benchmark collection, and technology mapped using the TSMC logic cell library.

BEST AREA, delay held constant 17% improvement

Losp improves upon the best area performance of the leading commercial synthesis tool by an average of 17%, when delay is not an important design consideration and is held constant.

BEST DELAY 14% improvement with 20% more area

Losp improves upon the best commercial delay performance by an average of 14%, at a cost of an additional 20% in area.

BEST AREA and DELAY 10% improvement with 10% more area

Many designs require a balance between delay and area; considering both, Losp improves upon the best commercial combined area and delay performance by an average delay reduction of 10% at an average area increase of 10%.

MORE ROBUST RESULTS

Commercial tools were unable to process one complex benchmark that Losp successfully reduced in area by 95% and in delay by 93%. In addition, two designs reduced by commercial tools contained fanout design faults (i.e. violations of pre-specified structural requirements). An engineer must then iterate the design synthesis using different parametric settings in order to bring the design into conformity. In contrast, the Losp engine always respects achievable design specifications, returning designs that do not contain faults.

Area Reduction Performance

For reduction in design area, the delay of each comparison design was held constant for both Losp and the commercial tools. The best area results were measured using the TSMC fabrication library. The constant delay for these results is quite high, about two to three times the minimal achievable delay. Thus, the area improvement results apply only when delay is not a significant design consideration.

For eight benchmark designs, the average area improvement by Losp over commercial synthesis tools was 17%. This result shows the performance of the Losp area reduction algorithms only, and does not include application of Losp delay reduction tools.

Area reduction for different designs is highly variable, since different types of circuit functionality have vastly different structural characteristics. The Losp engine for this study was fully automated, no fine-tuning to particular design types occurred. Losp and other EDA algorithms are quite sensitive to the netlist structure of a design. Structure is effected by the type of functionality, by the original high-level design coding, by parsers from high-level design languages such as Verilog into netlists, by selection of technology mapping cell sets, and by several other factors.

Combined Delay and Area Performance

When the Losp delay reduction algorithms are included during area optimization, Losp can produce similar area results but with a significantly lower delay, thus improving the combined delay/area performance. For example, for an industrial DSP design, the commercial system was scripted to produce its best area reduction. Losp then produced its best area reduction while holding delay to the same value as produced by the commercial tool, and its best delay reduction while holding area to that of the commercial tool. In the chart below, delay is measured in picoseconds (ps) and area is measured in square microns (u^2):

	Commercial		***Losp***		***Losp***
	ps-delay	***u^2-area***	***ps-delay***	***u^2-area***	***Improvement***
DSP design	10513	44971	10232	36154	20% area reduction
			5856	44896	44% delay reduction

For this design, Losp combined delay and area reduction decreased area by 20% while maintaining approximately the same delay as the commercial tool. Losp combined delay and area reduction decreased delay by 44% while maintaining approximately the same area as the commercial tool.

Delay Reduction Performance

Design optimization is a careful trade-off between delay and area. The above result shows comparative performance when delay is relatively high. In most designs, however, delay is significantly more important than area, given that area is not greatly increased.

For the following measurements, the best *delay* output from a leading commercial tool was post-processed by Losp. The Losp algorithms further reduced the design delay at a small cost of additional design area. In this comparison, only the Losp delay reduction algorithms were used, the Losp logic and area reduction engines were not used. As a consequence, Losp transformations that decreased delay did so by increasing the design area, in effect trading-off delay for area.

Design Function	***Commercial Output***		***Losp Best Delay***		***Losp Delay+Area***	
	ps-delay	***u^2-area***	***%delay***	***%area***	***%delay***	***%area***
interrupt controller	2020	6200	11	-26	8	- 6
error correction	2000	11690	10	- 4	10	- 4
9-bit ALU	2400	25720	13	- 6	12	- 6
adder/comparator	2310	39390	13	-12	7	- 2
vector rotation	1040	2280	8	-22	6	-14
DSP core	4610	94340	27	-47	15	-30
Average (6 designs)			**14**	**-20**	**10**	**-10**

Above, the Losp comparative performance is expressed as the percentage *decrease* in delay, for a percentage *increase* in area. The ***Commercial Output*** columns show initial best commercial measures of delay and area. The results in the ***Losp Best Delay*** columns were achieved by beginning with the commercial tool output and then applying Losp delay reduction algorithms. On average, Losp decreased the commercial tool's delay by 14% at a cost of 20% more area.

Losp dynamically generates many data points for selecting the best balance between area and delay. The ***Losp Delay+Area*** columns show the results for one data point in each design, selected due to a good ratio of delay improvement to area cost. On average, Losp improved the commercial tool's best performance by 10%, at a cost of 10% in additional area.

Steerable Delay/Area Trade-off

Losp provides the opportunity to steer fine-grain reduction results, to find different delay/area trade-offs. The range of choices is illustrated below using the DSP design; they are suggestive, and not based on a statistical sampling. The chart shows the raw performance data from a commercial tool, and the relative performance of Losp for three different design objectives: lowest area, lowest delay, and good results for both area and delay. Both tools applied one iteration of their global optimization routine to the same netlist. The commercial tool provides the results of three runs.

DESIGN	**Commercial best**		**Losp Area**		**Losp Delay+Area**		**Losp Delay**	
OBJECTIVE	***ps-delay***	***u^2-area***	***delay/area***		***delay/area***		***delay/area***	
Low Area	10513	44971	-2%	10%	44%	0%	-52%	20%
Low Delay+area	5260	64700	0%	3%	9%	-14%	6%	-1%
Low Delay	4612	94931	0%	22%	4%	15%	27%	-47%

For each design objective, three designs generated by Losp are compared to the best commercial results for that design objective. Each column presents the Losp comparative percentage improvement for delay and area, given the design goal. **Losp Area** shows area improvement for each design goal; **Losp Delay** shows delay improvement; and **Losp Delay+Area** shows combined delay and area improvement.

Area Improvement

Area design objective row: When area reduction is the primary objective, Losp can produce a design with 10% less area at the cost of increasing delay by a negligible 2%, or a design with 20% less area at a cost of a much greater delay (here 52%). Should the commercial tool's area reduction be acceptable, Losp can produce a design that is 44% faster for the same area.

Losp Area column: When both low delay and low area are desirable, Losp matches the commercial tool's performance (area is decreased by a negligible 3%). Should low delay be the primary objective, the Losp design requires 22% less area to achieve the same delay as the best produced by the commercial tool.

Delay+Area Improvement

Often a design requires good performance on both metrics. Since this is the most common case in modern designs, commercial software is fine-tuned to provide the best performance here. The Losp performance advantage is not large, however Losp differentially permits designers a wider range of dynamic trade-off choices, while the commercial results are significantly less flexible.

Delay+Area design objective row: When both delay and area are important, Losp can provide a range of trade-off choices, either meeting the commercial tool's performance (with a negligible 3% area improvement), or improving delay slightly (6%) with negligible area cost (-1%), or improving delay moderately (9%) with a moderate area cost (14%).

Losp Delay+Area column: Given the desirability of both low delay and low area, when area is minimized, the Losp design is 44% faster. When delay is minimized, the Losp design is 4% faster and 15% smaller.

Delay Improvement

Delay design objective row: With delay paramount, the Losp design is 27% faster. In this case, the 47% greater area is generally acceptable in order to gain improved delay performance. Without significantly improving delay, the Losp design is 22% smaller. Alternatively, Losp can improve delay slightly (4%) while reducing area by 15%.

OPTIMIZATION OF AN FPGA DESIGN

We applied Losp as a post-process to a large industrial design that had been optimally mapped to an FPGA architecture by the leading FPGA synthesis tool. FPGAs are characterized by large logic cells called Look-Up Tables (nLUTs, where n refers to the number of inputs into the LUT).

The chart below indicates Losp reduction gains by counting the number of four different structural forms:

- Number of two-input NAND gates: before and after logic reduction.

- Number of wire nets: reduction of routing requirements for a design.

- Raw number of 4LUTs: the logic resource requirements for placing the design into a 4LUT silicon architecture.

- Weighted number of LUTs: lower input LUTs (i.e. 2LUTs and 3LUTs) require less routing and power resources.

The counts of structural forms below were taken of the best commercial performance (before) and the Losp improvement (after).

	NAND GATES		***4LUTS***		***WIRE NETS***		***WEIGHTED LUTS***	
	Before/After		***Before/After***		***Before/After***		***Before/After***	
Count	71744	54490	15213	15419	106420	70253	13027	8252
Losp gain	24%		-1%		34%		37%	

When measured by the number of two-input NAND logic gates, Losp reduces the structural netlist by 24%, from 72K to 55K gates of the same type. (The design incorporates around 2000 registers that are not included in the gate count.)

Losp reduction is highly sensitive to optimization goals. Losp LUT mapping did not reduce the number of 4LUTs compared to the 4LUT map generated by the commercial software (-1%) when only 4-input LUTS are permitted. Losp does convert many 4LUTs in the design into 2LUTs and 3LUTs, thus reducing the wires required to route the design logic. Thus, Losp reduces the routing required by the 4LUTs by 34%. It is widely acknowledged that routing, and not logic resources, is the design bottleneck for FGPA architectures.

When LUT mapping is weighted using a common gate counting metric that combines logic and wiring requirements, Losp reduces the design complexity by 37%.

LOSP COMPETITIVE ADVANTAGES

Why Losp Out-Performs Conventional EDA Software

Losp is based upon and implemented using a new model of logic that differs significantly from product terms (SoP), binary decision diagrams (BDDs), and all other commercial and academic methods of logic computation. Discovery and implementation of new mathematical techniques is not unusual, rather, it is the common way that mathematical knowledge and tools grow. In the 1990s, BDDs fundamentally changed the synthesis algorithms used in CAD tools. Boundary logic is another fundamental change.

The computational model used in Losp is unary, based on one concept, rather than binary, which is based on two concepts (such as 0 and 1). The single concept is a boundary, or container.

Conventional logic distinguishes between two values (i.e. 0 and 1, True and False). In contrast, containers distinguish between two spaces (i.e. inside and outside), using only one value (i.e. the presence of a boundary). Although thinking of logic as containment relations is new and unfamiliar, everything that can be done with conventional logic forms can also be done using container boundaries.

Summary of Demonstrated Competitive Advantages

Losp demonstrates these performance characteristics:

- produces designs that are out of reach to commercial tools, providing designs with delay and area performance that are superior to designs produced by existing commercial tools.
- produces significantly better delay (14%) when area does not matter and significantly better area (17%) when delay does not matter.
- produces a wider range of preferable designs with concurrently good area and delay performance.
- reduction applies to the full variety of design types.
- does not return results that contain design faults.
- optimizes at least one design that commercial tools could not process.
- provides responsive parametric settings that allow a designer to specify a detailed configuration of constraints that guide engine performance.
- requires far fewer design iterations to achieve specified performance, providing more stable, more controllable, and more predictable parametric performance, thus improving time-to-market.