ILOC/SYNOPSYS COMPARATIVE PERFORMANCE

William Bricken
April 2004

CONTENTS

This memo contains selected technical details and performance comparison results between the ILOC alpha-level code and the Synopsys logic synthesis and technology mapping product. Both area and delay results are included. This comparison project used only combinational designs to make comparison easier.

SUMMARY

	ILOC improvement over Synopsys	
BEST AREA	17%	delay held constant
BEST DELAY	15%	with 20% more area
COMPETITIVE DELAY	10%	with 10% more area

ILOC has been measured to reduce Synopsys' best delay performance by an average of 15% for a diversity of designs, at a cost of an additional 20% in area. ILOC also reduces Synopsys' best area performance by an average of 17% when delay is not an important design consideration and is held constant. Many designs require a balance between delay and area; considering both, ILOC provides an average delay reduction of 10% at an area cost of 10%. Unlike other commercial systems, ILOC permits a designer to steer the engine performance through a diversity of delay and area trade-off points, selecting the ratio that is most desirable for a particular design. The ILOC implementation will improve significantly when a more robust control structure for making transformation choices is added.

PERFORMANCE COMPARISONS

Benchmark Designs

ILOC performance was measured and compared to Synopsys best performance on eight benchmark designs. Table I identifies characteristics of these designs, including functionality, input/output count, gate and net count, and area and delay as measured by Synopsys best performance. The comparison designs range over a diversity of structural and functional types, including ALUs, lookup tables, arithmetic, error correction, and other functionalities.

Name	*I/O*	*Gates*	*Nets*	*Area*	*Delay*	*Function*
C432	36/7	212	355	3338	3782	interrupt controller
9SYMML*	9/1	222	159	2184	2021	count 1s
C1355	41/32	518	949	9213	3705	error correcting
TABLE5*	17/15	1180	718	10738	2008	lookup table
C5315	178/123	1926	3101	30132	5987	9-bit ALU
C7552	207/107	2238	4343	40549	7075	adder/comparator
CORDIC	23/2	2890	2079	29578	2475	vector rotation
DMC54	77/41	5239	8812	94931	4612	DSP

Table I: Eight Comparison Designs

Synopsys was unable to process one benchmark, not included above, that ILOC successfully reduced in both area and delay. DMC54 requires a slightly different methodology and has been discussed separately above.

Table I includes two designs (9SYMML and TABLE5) for which Synopsys produced reduced versions that contained fanout design faults. Apparently the Synopsys tools optimize as best they can, but at the same time permit design faults to occur. An engineer must then iterate the analysis using different parametric settings in order to bring a design into conformity. In contrast, the ILOC engine always respects design specifications when they can be achieved. The ILOC results reported for these two design incorporate no design faults.

Four other benchmarks were included in the original comparison study, and were excluded from this analysis:

Name	*Function*	*Reason for omission*
CM85A	magnitude comparator	too simple
MAJTAUT	simple tautologies	used to test logic reduction only
TOO_LARGE	FPGA routing map	Synopsys is unable to process this design
DES	encryption	saved for validation purposes

Area Reduction Performance

For reduction in design area, we held the delay of each comparison design constant, and measured the area using the Synopsys TSMC mapping tools. ILOC and Synopsys area measurements are very highly correlated. The constant delay in these results is quite high, representing two to three times the minimal achievable delay. Thus, these area results apply only when delay is not a significant design consideration.

Table II shows ILOC and Synopsys design areas, expressed in square microns, at a constant delay, expressed in picoseconds. Cell sizes are from the TSMC logic library. The average ILOC area reduction is 17% better than Synopsys.

Competitive area reduction is highly variable, as would be expected. The ILOC engine for this study was fully automated, no fine-tuning to particular design structures occurred. ILOC (and other EDA) algorithms are quite sensitive 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, and by selection of technology mapping cell sets.

	Synopsys		ILOC		area
Name	*delay*	*area*	*delay*	*area*	*reduction*
9SYMML	3750	2096	3750	1790	16%
C432	6330	1939	6330	1673	14%
C1335	5550	5914	5550	3672	38%
TABLE5	5100	5489	5100	5216	5%
C5315	6820	14047	6820	12301	12%
C7552	7770	18392	7770	15813	14%
CORDIC	3130	2202	3130	1144	48%
DMC54	13700	43076	13700	48086	-12%
				Average	**17%**

Table II: Area Performance with Delay Held Constant

Synopsys and ILOC area models are in close agreement, however their delay models differ by 15-20%. The above delay measures use the Synopsys delay model, and are not comparable to results reported in other parts of this report.

Results with higher percentage competitive reduction (such as CORDIC and C1335) are generally due to ILOC's superior logic minimization capabilities. Results with lower competitive reduction (such as DMC54 and TABLE5) are generally due to minimal availability of redundant logic to eliminate. The DMC54 result, for example, swings to 20% in favor of ILOC when delay is increased even more, to around 20 nanoseconds. This delay was considered unacceptable.

More importantly, these measures were taken prior to the development of an ILOC delay reduction capability. When the ILOC delay reduction algorithms are included, ILOC can produce similar area results but with a significantly lower delay, thus improving its relative performance. For example, using ILOC metrics, the DMC54 comparative area reduction is:

	Synopsys		ILOC		reduction
	delay	*area*	*delay*	*area*	
DMC54	10513	44971	10232	36154	20% area reduction
			5856	44896	44% delay reduction

This DMC54 comparison shows that ILOC area reduction augmented with delay reduction can decrease area given the same delay by 20%, or can decrease delay given the same area as Synopsys by 44%. When the ILOC control structures integrate both delay and area performance, similar gains can be expected for the other designs in the Table II above.

Design optimization is a careful trade-off between delay and area. When delay is set higher, both Synopsys and ILOC can produce better area reduction. However, a preferable comparison would be when delay is moderately low, since this is preferred by designers. These results are presented next.

Combined Area and Delay Performance

In most designs, delay is significantly more important than area, given that area is not greatly increased. We have compared ILOC performance to Synopsys performance using the ILOC internal delay model. The comparison methodology was to begin ILOC reduction with the best output from Synopsys, using ILOC as a post-processor. The ILOC algorithms further reduced the design delay at an additional cost of design area. In this study, the ILOC logic and area reduction engines were not used. As a consequence, ILOC transformations that decreased delay did so by increasing the design area, in effect trading-off delay for area.

Name	*Function*	*Synopsys Output*		*ILOC Best Delay*		*ILOC Competitive*	
		psDelay	*u^2Area*	*%delay*	*%area*	*%delay*	*%area*
9SYMML*	count 1s	1300	4340	9	55	4	41
C432	interrupt controller	2020	6200	11	26	8	6
C1355	error correction	2000	11690	10	4	10	4
TABLE5*	lookup table	1560	16590	10	32	5	14
C5315	9-bit ALU	2400	25720	13	6	12	6
C7552	adder/comparator	2310	39390	13	12	7	2
CORDIC	vector rotation	1040	2280	8	22	6	14
DMC54	DSP core	4610	94340	27	47	15	30
Average (8 designs)				**13**	**26**	**8**	**15**
without design faults (6 designs)				**15**	**20**	**10**	**10**

Table III: ILOC and Synopsys Performance on Eight Designs

Table III presents area and delay results for Synopsys and for ILOC for the eight benchmark designs. Comparison data is expressed as the percentage improvement in delay for ILOC over Synopsys, at an additional percentage area cost. ***Synopsys Output*** provides raw measures of area and delay. ***ILOC Best Delay*** was achieved by beginning with the Synopsys best performance and then applying ILOC delay reduction algorithms. Localized ILOC transformations generate many data points for trading area for delay. The ***ILOC Competitive*** performance above shows one of those data points, selected due to a good ratio of delay gain to area loss.

For *best delay performance*, ILOC decreased Synopsys delay by 13% at a cost of 26% more area. These figures are not fair, since they include two designs for which the Synopsys version contained design faults. Design faults can be eliminated by increasing delay and/or area. For a fairer comparison, when these two designs are not included in the averages, ILOC improved Synopsys delay performance by 15%, at a cost of 20% more area.

The *competitive comparison* shows a reduction result that is sensitive to both delay and area, optimizing both concurrently. For the competitive comparison, on average ILOC improved Synopsys' best delay performance by 8%, at a cost of 15% in additional area. Eliminating the two faulted designs, ILOC improved Synopsys delay by 10% with an average cost in area of 10%.

The current ILOC architecture applies area and delay reduction algorithms separately. Future ILOC releases will integrate these tools, potentially providing competitive delay and area results concurrently. For now, the ILOC delay reduction algorithms trade area for delay, so that lower delay is always accompanied by increase design area.

Comparison Using the Synopsys Delay Metric

ILOC and Synopsys each have a model for measuring delay and thus identifying critical paths. The Synopsys model predicts physical timing within 1-2%; the ILOC model is less refined, predicting physical timing within about 10%. Roughly, the Synopsys model uses 50 table entries per internal pin in a network, while ILOC uses one.

Measurement of delay performance is technically difficult, particularly because delay accumulates over cells in a multiplicative, non-linear manner. A consequence of differences in models is that fair comparison of delay performance across different reduction engines is technically not achievable. Both ILOC and Synopsys use their own internal models to guide reduction decisions. Very small differences in the behavioral models of cells and wires can lead to large differences in delay performance modeling and in the identification of critical paths.

The above delay results use the internal ILOC delay model. The comparison methodology first uses Synopsys to generate its best delay version using its own internal metrics to guide reduction choices. This result is then measured by the ILOC delay model. ILOC further reduces these results, using the ILOC internal model to guide choices and to assess the results. The ILOC measurement of both Synopsys and ILOC performance is then compared.

Alternatively the Synopsys delay model can be used to assess ILOC reductions. The idea is to use Synopsys' measurement of its original performance, then to use ILOC as a reduction engine, and finally to return to Synopsys for measurement of ILOC reductions. The flaw in this approach is that ILOC does not use the Synopsys models as a basis for reduction decisions, thus the ILOC reduction decisions do not reflect what Synopsys measures.

The Synopsys measurement of ILOC delay reduction results shows a 3% increase in delay, rather than a 15% decrease. ILOC reductions are essentially random when measured using Synopsys metrics, the delay of some designs decreased slightly and some increased slightly. This result is best understood as an inappropriate application of a measuring device to a process that does not address the metric of that device.

Fine-Grain Differences in the Two Delay Models

Indications as to the differences between the ILOC and the Synopsys delay models can be found by examining the delay path cell by cell. The following observations are ordered in importance:

-- Synopsys incorporates raising and falling input signals in its more elaborate model.

-- The most critical path in designs with multiple outputs does not match across metrics.

-- The Synopsys model for XOR and XNOR cells is substantively different, resulting in cell measurements that diverge by several hundred picoseconds.

-- Synopsys uses more elaborate modeling than does ILOC for selection of delay cells, buffers and inverters.

-- The Synopsys model includes some hidden modeling parameters, since the same cell in the same path with the same rising/falling signal has slightly different delays for the two models.

-- ILOC exposed obvious opportunities for improvement in the Synopsys version, that Synopsys did not implement.

-- Currently, ILOC reduction decisions are not integrated, do not include specialized treatment of specific design structures, and are relatively blind about the context within which they are applied.

Comparative Performance on the DMC54 Design

ILOC provides an opportunity for a designer to steer reduction results to different area/delay trade-offs. The range of choices and how they might effect performance results are illustrated using the DMC54 design.

	Delay reduction		*Area reduction*		*Competitive reduction*	
Synopsys best	4612	94931	10513	44971	5260	64700
ILOC comparable	0%	22%	-2%	10%	0%	3%
ILOC best	27%	-47%	-52%	20%	6%	- 1%
ILOC competitive	4%	15%	44%	0%	9%	-14%

Table IV: ILOC and Synopsys Performance on DMC54

Table IV presents ILOC's relative performance improvement over Synopsys for the DMC54 design, for three different design objectives: low delay, low area, and both low delay and low area.

Area Comparison

When delay is held constant, ILOC requires 22% less area than Synopsys during delay reduction, and a range from 3% (negligible) to 10% less area for other reduction goals. Synopsys biases area optimization toward very good delay performance even when scripted to emphasize area reduction. The more that area is favored in a reduction process, the more ILOC out-performs Synopsys. However for this data, ILOC area reduction is strongest when delay is most important. This advantage aligns with the common preference for faster designs at a moderate expense to area.

When delay is completely flexible, ILOC can produce a design with 20% less area (here delay is 50% greater). ILOC can also produce a design that is 44% faster when area is emphasized but not to an extreme.

Another advantage is that Synopsys violates fanout constraints to achieve the above low area results, whereas ILOC does not.

Delay Comparison

ILOC improves upon Synopsys' best delay performance by 27% at a cost of around 50% more area. When delay is important, this trade-off is quite acceptable. When area is more critical during delay reduction, ILOC can still improve delay by 4% while also improving area by 15%.

ILOC did not generate a comparison for delay reduction when area is held to approximately 95000 u^2. Starting from an area minimal design, ILOC improves delay at the cost of area, but only up to around 80000 u^2. Starting from a delay minimal design, ILOC improves area at the cost of delay, but only down to around 115000 u^2.

Competitive Comparison

Often a design requires fairly good performance on both metrics. When neither delay nor area is minimal, Synopsys returns results that favor delay reduction; an example competitive result is 15% greater than Synopsys' best achievable delay, while being 44% greater than its best achievable area. The closest ILOC data point to this competitive Synopsys result is negligibly different in both metrics. ILOC however can also produce results that improve delay slightly (6%) at negligible area cost, moderately (9%) at a moderate area cost (14%), and significantly (27%) at a substantial area cost (47%).

This competitive comparison emphasizes ILOC's capability to provide a superior range of competitive design choices directly from parametric specification, without the overhead of design iteration.

ADDITIONAL TECHNICAL DISCUSSION

Difficulties with Measurement of Delay

Delay Measurement

The critical path of a design is the longest modeled path between primary input and output. ILOC and Synopsys each have a model for measuring delay and thus identifying critical paths. The Synopsys model predicts physical timing within 1-2%; the ILOC model is less refined, predicting physical timing within about 10%. Roughly, the Synopsys model uses 50 table entries per internal pin in a network, while ILOC uses one. The DMC54 design, for example, has close to ten thousand internal pins, requiring half a million data points to achieve the accuracy of the Synopsys model.

Measurement of delay performance is technically difficult. A consequence of differences in models is that fair comparison of delay performance across different reduction engines is technically not achievable.

Very small differences in the behavioral models of cells and wires can lead to large differences in delay performance modeling and in the identification of critical paths. DMC54 has over 500 million branches passing through 36 levels of cells on one given critical path alone; any of these branches could support a different critical path, depending on small differences in the delay model of each cell or wire.

It is not possible to use the Synopsys model to measure ILOC reduction results, since ILOC uses a less refined model that less accurately identifies paths and selects transformation choices. As a consequence, ILOC will sometimes identify and optimize different critical paths than Synopsys. By iteration, ILOC will usually optimize those critical paths identified by Synopsys, so that reductions converge. However, when ILOC results are measured by Synopsis, the paths being compared are often different paths.

Synopsys results are therefore measured by the ILOC model, reducing the accuracy of the Synopsys results but not changing the actual network structure generated by Synopsys. Roughly, the resulting comparison is accurate within 10%. This approach still suffers from a possible mis-measurement of actual Synopsys reduction results, however it is always more appropriate to use less refined measures when accuracy is lacking.

Interpretation of Results

Several factors must be considered in interpreting comparative results:

1) Reduction engine performance is neither linear nor continuous nor deterministic.

2) Identification of critical paths is meta-stable; very small changes in a model can lead to largely different critical paths.

3) Each different *initial network* for a given functionality leads to a different result. These results can vary by around 20%.

4) Each *functionally different* design elicits very different performance for a given engine.

5) No algorithms can be tuned to converge upon good results for all designs.

6) ILOC has up to 100 different parametric settings that can each vary measurement results significantly.

7) Slight variations between the measurement models in two different engines can lead to greatly different measurements.

The extreme variation of measurement results means, technically, that standard descriptive statistics such as average performance, do not provide accurate information. Further, any comparative statements of performance will have large numbers of exceptions. To date, technical improvement of delay measurement in EDA tools has focused on accuracy of timing *prediction*. Delay *optimization* is still a relatively unorganized and poorly understood field.

For example, Synopsys performance for the same design and the same initial network can vary by 15% in delay and 20% in area for two different specified timing goals that are both equally unachievable . This requires engineers to iterate timing targets, somewhat by guesswork, until an acceptable design that meets specifications is found. In Synopsys, the balance between design delay and design area cannot be specified by the designer, rather a delay is specified and the area is then selected by the engine's algorithms. Small parametric differences can lead to large design differences.

Thus, the complexity and instability of delay reduction algorithms makes design guidance unmanageable. This is a primary contributor to the costly practice of timing iteration to reach convergence on design goals.

One significant competitive advantage is that ILOC's simple formal models can largely eliminate design iteration during logic synthesis and delay modeling.

Delay Reduction Comparison Methodologies

Data for two methodologies is presented below. The first methodology illustrates the cost of not having an integrated delay and area reduction

engine. The second methodology illustrates the potential within ILOC when its delay and area minimization capabilities are integrated.

Methodology I: Beginning from a non-optimized version of each design, both systems independently optimize that design for delay. ILOC first minimizes the design for area and then independently minimizes that result for delay. A consequence is that some area gains are lost.

Methodology II: Synopsys first optimizes a non-optimized design for delay. This result is then optimized again for delay by ILOC. The ILOC area optimization capabilities are not used.

All results are measured by the ILOC delay model.

Methodology I

From an original non-optimized design specification, both Synopsys and ILOC generated their best delay reduction version. These versions were measured by ILOC. The results are reported in Table V, both as raw data points and as a percentage of Synopsys performance. These results were obtained without changing the ILOC control algorithm and without eliminating poor transformation decisions. In particular, the ILOC area results do not incorporate area minimization for cells that are not on the critical path. All of these results can be improved upon by fine-tuning the ILOC engine parameters.

Name	*Synopsys best*		*ILOC performance*		*Improvement* *Delay*	*Area*
9SYMML*	1411	4341	1268	7286	10%	- 68%
C432	2018	6204	2410	6513	-20%	- 5%
C1335	2001	11691	1989	14271	1%	- 22%
TABLE5*	1564	16590	1507	35768	4%	-216%
C5315	2404	25718	2615	35024	- 9%	- 36%
C7552	2311	39392	2701	48989	-17%	- 24%
CORDIC	1036	2276	1071	3330	- 3%	- 46%
DMC54	4612	94931	4368	128808	5%	- 36%
				Average	**- 4%**	**- 56%**

Table V: Comparison For Independent Delay Optimization

Asterisks identify two designs (9SYMML and TABLE5) for which Synopsys violated design fanout constraints. Synopsys uses a conglomerate objective function, attempting to meet delay, area, fanout, wire capacitance, and other design objectives at the same time, in an integrated fashion. A consequence is that Synopsys will sometimes fail to meet design criteria. The engineer must then iterate the design specifications until Synopsys returns a satisfactory result. One would expect that iterating the design parameters would eventually lead Synopsys to respect fanout constraints and that the resulting design would have a slower delay measurement and a higher area measurement, leading to a better ILOC comparative performance. The ILOC designs do meet fanout and all other design constraints

ILOC lacks a comprehensive control structure, and is not a refined tool for delay reduction. Thus the above ILOC results are highly variable, more or less hit-or-miss depending upon the fit between reduction algorithms and design structure. As an example, when one parametric setting for TABLE5 is changed, ILOC generates a result of -8% delay and -25% area. This single data point would decrease the overall delay average slightly (from -4% to -5%), but would greatly reduce the overall area average (from -56% to -32%) by removing the outlying 216% area increase.

As another illustration of instability of delay comparison data, the above Synopsys results were generated using a delay specification that was approximately 300ps faster than what Synopsys is able to achieve. The 300ps target was measured through Synopsys design iteration. Changing this performance goal to a more stringent (and more uninformed) expectation, approximately 2000ps better than Synopsys can achieve, generates significantly different performance from Synopsys. The different Synopsys results change the average comparative ILOC delay from -4% to -2%, and the average area (adjusted for TABLE5) from -32% to -17%.

More fundamentally, ILOC generated these results by applying area optimization independently of delay optimization. A reduction strategy that separates area and delay transforms is inherently poor, since good results require that fine-grain transformation decisions take into account the effect of each transformation on both area and delay concurrently. As well, this ILOC strategy degrades the off critical path area during delay optimization, significantly lowering the area reduction performance. All of the above results can be significantly improved by adding an ILOC control strategy that considers area and delay concurrently.

Methodology II

To address the lack of ILOC control refinement, our primary comparison strategy for delay reduction is to measure the ILOC capability to improve upon designs that had already been maximally optimized for delay by Synopsys. This piggy-back strategy can be of independent value when ILOC is seen as a

Synopsys post-process, however this strategy is also an expedient used to expose the potential of a more fully developed ILOC engine. Thus the following results show how much ILOC can improve upon Synopsys, given that ILOC is further developed by including existing and understood Synopsys techniques.

This methodology requires that ILOC increase area in exchange for delay improvements. The ILOC logic and area optimization engine is not used. Table VI shows comparative raw data points and expresses ILOC performance as a percentage of Synopsys performance.

	Synopsys		*ILOC*		*Improvement*	
Name	*best*	*output*	*post-processing*		*Delay*	*Area*
9SYMML*	1411	5177	1242	6266	12%	-21%
C432	2018	6204	1814	7055	10%	-14%
C1335	2001	11691	1799	14508	10%	-24%
TABLE5*	1564	16590	1488	18874	5%	-14%
C5315	2404	25718	2108	27166	12%	- 6%
C7552	2311	39392	2101	41763	9%	- 6%
CORDIC	1036	2276	948	2777	8%	-22%
DMC54	4612	94931	4298	98492	7%	- 4%
			3941	123241	15%	-30%
			3344	139130	27%	-47%
				Average	**9%**	**-14%**

Table VI: Comparison for Post-Processing Delay Optimization

These results show that ILOC can achieve post-processing delay reduction of an average of 9% for a cost of 14% increase in area. These results can be significantly improved, since they too were generated using a blind automated control strategy that applied the same reduction algorithm to all designs.

Three DMC54 results are listed, one using the same algorithms as the other designs and included in the averages, and two that are recapitulated from Table VI, one for competitive delay reduction and one for maximal delay reduction. The competitive and the maximal DMC54 delay reductions suggest what is achievable for the other designs, given an evolved ILOC control strategy. In particular, beta-release ILOC performance improvement over Synopsys is expected to be around 20% delay reduction for around 30% area increase.

ILOC Control Structure

Synopsys, being a mature tool, was run only once on each design. During development of algorithms and reduction strategies, ILOC generated literally thousands of versions of each design. Thus we have available a rich mapping of ILOC reduction results over many different parameter settings. All reported results are from entirely automated, "push button" algorithms.

The best reduction strategy for a given design depends significantly upon the network structure of that design. Since several ILOC versions are available for each design, the best performing version is reported. Since ILOC techniques are based upon structural network patterns, we assume that pattern templates to identify and match specific patterns with specific reduction strategies can be built for a wide collection of commercial designs and will be incorporated into a forthcoming ILOC beta-release. Such a top-level reduction control strategy has not yet been implemented because it is dependent upon a large database of commercial designs.

The ILOC reduction strategy at this time is fairly rigid. About 30 local cell reduction transformations are organized hierarchically into five sequentially applied groups. Whenever a transformation group succeeds in making a reduction step, the reduction control algorithm begins again with the new design incorporating the small change. Some cell transformations are filtered to allow only local delay improvement, while others are applied at all times. None of these algorithms simultaneously considers both delay and area effects. The function of each transformation group follows:

Group One:	Greatly reduce the number of cell types.
Group Two:	Condense adjacent cells that always benefit.
Group Three:	Condense adjacent cells pending delay improvement.
Group Four:	Reduce fanout loads.
Group Five:	Terminal critical path changes that always benefit.

Future versions will optimize the selection of particular transformations without hierarchical organization.

COMPARATIVE PERFORMANCE FOR DMC54

We have extensive measurements of ILOC delay and area performance for several designs. DMC54 is one such design that has been mapped extensively; some data points generated by changing ILOC parameters to achieve desirable performance for this design are listed below. This data has been selected from over ten thousand measurements, and is intended to illustrate several performance characteristics.

In Table VII, data pairs list DELAY in picoseconds followed by AREA in square microns. Rows are ordered from lowest delay to highest delay. Three columns are distinguished by the following criteria:

Column 1: ILOC performance showing an apparent inverse relationship between delay reduction and area reduction.

Column 2: Exceptions to this apparent relationship.

Column 3: Measurements of Synopsys performance, optimizing delay and optimizing area.

Column 1		*Column 2*		*Column 3*	
delay	*area*	*delay*	*area*	*delay*	*area*
3344	139130				
3832	137027				
3941	123241				
		4087	128681		
4366	116976				
4448	80790				
				4612	94931
4615	73979				
		4884	113615		
4921	65028				
				5260	64700
5261	62908				
		5396	73577		
5492	61154				
5609	49820				
				5718	59173
		5856	44896*		
6119	49412				
		6322	90801		
7144	44958				
		8781	75185		
		9606	44958		
				10513	44971
10720	40283				
16030	36154				

Table VII: Comparison Performance Data for DNC54

Column 1: Inverse Relationship

Column 1 shows that the lowest achievable delay requires the largest area, and that the lowest achievable area requires the largest delay. In between these extremes, the data points are not at all linear. The ratio of delay gain to area cost varies widely, as illustrated below using the data points from Column 1.

COLUMN 1		***Ratio of change-in-delay to change-in-area***	
		picoseconds per 1000 microns^2	
3344	139130		
3832	137027	488/ 2103	232
3941	123241	109/13786	8
4366	116976	425/ 6265	68
4448	80790	82/36186	2
4615	73979	167/ 6811	25
4921	65028	306/ 8951	34
5261	62908	340/ 2120	160
5492	61154	231/ 1754	132
5609	49820	117/11334	10
6119	49412	510/ 408	1250
7144	44958	1025/ 4454	230
10720	40283	3576/ 4675	765
16030	36154	5310/ 4129	1286

Table VIII: Slope of Delay/Area Trade-Offs

Table VIII shows the ratio of change-in-delay to change-in-area (the slope) for successive delay data points. The third column of Table VIII expresses this ratio as the number of picoseconds gain in delay for every 1000 microns of additional area. Table VIII shows that trading area for delay is a highly irregular process. A linear relationship would exhibit a constant delay gain.

These measurements were drawn from ILOC reduction runs using several different parametric settings. Some of the non-linearity of delay improvement per unit area is due to the inclusion of ILOC's best results from these runs. ILOC uses a fine-grain transformation approach, with many (thousands) small iterative steps. Despite the curvilinear nature of delay/area points identified during reduction, ILOC algorithms can identify network structures that correspond to preestablished design goals as they wander over the possible structures for a specific functional design.

Table VIII is also intended to illustrate the sensitive dependence between control algorithm settings and results. The trajectory of a reduction is guided by several design parameters, such as the desired timing and area performance, the cell fanout limitations, and the capacitance on each wire. Within a given ILOC reduction run, convergence is quite smooth. The algorithms explore many different delay/area ratios, sometimes heading in disadvantageous directions but never failing to converge upon a final result.

The above results are generated by a single algorithmic approach. It is possible to turn on or off several dozen fine-grain transformations dynamically, providing a much richer set of generated design options, and providing a much greater degree of control in guiding the engine's exploration of options.

The top-level control structure for ILOC has not yet been written, although the ILOC software architecture can accommodate it easily. Control decisions need to be guided by performance information over dozens or hundreds of commercial designs. When the ILOC control structure is calibrated to a wide diversity of commercial designs, ILOC should be able to achieve unprecedented rapid convergence upon particular specified design goals.

Column 2: Exceptions

Column 2 shows that the apparent inverse relationship in Column 1 is even more unstable than it appears. These data points primarily illustrate poor parametric settings, for which results are clearly inferior. The single exception to this, (5856 44896), marked with an asterisk, shows an excellent low area result with a relatively good delay.

This data is intended to emphasize the "wandering" nature of the ILOC algorithms, and the highly non-linear performance of the ILOC engine. Using information that is purely local to each cell in a design, each fine-grain transformation can be tightened parametrically to make its application more or less likely. By selectively turning off some available transformations, engine performance can be steered toward design objectives, although the non-linearity of the engine behavior cannot be suppressed.