# ILOC COMMERCIAL AND COMPETITIVE ADVANTAGES

William Bricken
April 2004

Over the last two years, we have assembled several lists of the ILOC commercial and competitive advantages. This memo collects those lists.

## CONTENTS



## TECHNICAL ADVANTAGES

Iconic Logic is exceedingly elegant and efficient, providing ILOC with a competitive advantage for all network transformations. Network logic, area and delay reduction are currently implemented for circuit designs. The technical advantages of Iconic Logic can be summarized as:

1. Structural patterns are constructed from a single component type, rules for different cell types are not necessary.

2. Gates and wiring are a single concept, permitting path-oriented transformation and fluidity in meeting logic and connectivity design goals.

3. Less computational effort is required to achieve conventional objectives, including simpler representations that permit identification of advantageous transformations, and more powerful and efficient transformations.

4. A simple, integrated, and comprehensive theory guides all possible network transformations. This advantage provides a substantive competitive advantage since current commercial and academic techniques consist of a haphazard collect of unrelated ideas and algorithms.

## SUMMARY OF COMPETITIVE ADVANTAGES

-- ILOC can produce designs that are out of reach to Synopsys, improving either area reduction about 20% or delay reduction about 10%.

-- ILOC produces significantly better delay (25%) when area does not matter and significantly better area (20%) when delay does not matter.

-- ILOC produces a wider range of preferable designs with concurrently good area and delay performance.

-- ILOC delay reduction is consistent over a variety of design types.

-- ILOC does not return results that contain design faults.

-- ILOC requires far fewer design iterations to achieve specified performance, providing more stable, more controllable, and more predictable parametric performance.

-- ILOC provides responsive parametric settings that allow a designer to specify a detailed configuration of constraints that guide engine performance.

-- ILOC is able to optimize at least one design that Synopsys could not process.

## COMPETITIVE POSITIONING

Designers seek a solution that meets cost and speed specifications; designers are attracted to solutions that improve either area or delay without sacrificing performance of the other specification. Our competitive position based on performance of the current alpha-version of ILOC, is as follows:

1. *Cases where Synopsys meets both specifications:*

   -- ILOC can provide approximately 20% less area for a given delay when that delay is not close to Synopsys' best achievable delay.
   -- ILOC can provide approximately 10% lower delay for a given area when ILOC logic reduction is significantly better than that of Synopsys.
   -- ILOC can generally reach specified design goals on the first design iteration. ILOC is also more flexible and more accurate in fine-tuning design specification trade-offs.

2. *Cases where Synopsys fails to meet delay specifications:*

   -- ILOC can provide approximately 10% lower delay as a post-process to Synopsys, for an area cost of approximately 15%.

3. *Cases where Synopsys fails to meet area specifications:*

-- ILOC can provide about 15% lower area when delay is not critical.

4. *Cases where Synopsys fails to meet both specifications:*

-- For some designs, ILOC can meet both specifications, particularly those designs for which ILOC has superior logic reduction.

## ADDRESSING PERFORMANCE COMPARISONS

A common technique used to describe optimization performance is to list a large diversity of commercial designs or design components, and then to show the relative improvement for each design. Group averages are eschewed in favor of a pictorial sampling of specific results, while performance variability is acknowledged implicitly in the range of specific comparative results. The complexities of parametric settings, multiple iterations, etc. are simply not mentioned. Instead, corporate technical white papers present what is known about the technical issues.

Another prevalent technique is advocacy by respected professional peers and experts. This approach finesses completely the discussion of technical performance details.

Perhaps most importantly, BTC software development practices must include extensive benchmark measurement and validation. One might expect over half of the software development effort to be invested in comparative performance testing and refinement.

Delay data instability is an inherent characteristic of the delay optimization problem. Apparently, Synopsys de-emphasizes this issue, passing convergence of design specifications onto the designer as "timing iteration". Due to the rather immature state of delay minimization techniques and the rapid evolution of design requirements, engineers do not yet expect a tool that simplifies achievement of design specifications, including timing.

## *EASE-OF-DESIGN*

Ease-of-design has always been ILOC's strength. This report establishes that ease-of-design is not incompatible with superior improvement of design performance. One approach to conveying the desirability of ILOC performance is to begin initially with the idea of improved performance, and then strongly emphasize the features that make the tool easy to use, in particular:

-- generality of superior performance over most designs
-- one or few iterations to reach design specifications
-- course-grain and fine-grain controllability of engine performance
-- comprehensive and integrated logic synthesis capabilities
-- predictable output
-- fast processing time
-- ease of extensibility to new architectures and feature scales
-- complete compatibility with existing techniques
-- complete integration with existing skills
-- no new design skills to learn
-- less design work to do, leading to faster time to market

Most of these interaction advantages must be built into the ILOC interface. The potential of these performance characteristics is inherent in the core engine, but there is considerable development work to bring them to the interface and put a designer in control of engine behavior.

## *FORMAL VERIFICATION AND EXTENSIBILITY*

The primary mathematical advantage of ILOC is that the underlying boundary logic algorithms greatly simplify transformation of network structure. This advantage is entirely general and can be applied to benefit all network architectures and all modeled network performance characteristics (delay, area, power, noise, etc). It has been quite consistent that roughly six months of directed effort to any specific application issue has resulted in prototype software with formal underpinnings that improves upon the current state-of-the-art. One clear example is that the ILOC delay reduction tools appear to organize and condense a field that at the moment consists of a disconnected set of semi-efficient partial techniques.

**SOFTWARE VALUE PROPOSITIONS** (September 2002)

[This memo was originally prepared to describe the ILOC value propositions to potential investors. The ISSUES is based on interviews with designers.]

***ISSUE:*** Hardware design is costly, time-consuming, and difficult
***SOLUTION:*** Pragmatic benefits provided by a unified tool model

- incremental design development
- faster time-to-market
- fewer, faster design iterations
- fewer design errors
- controlled trade-off of key design parameters
- cost savings and productivity gains
- preserves reliability of delivery schedules
- integrated verification prevents design change errors

***ISSUE:*** Larger, new designs press current tools to their limits
***SOLUTION:*** New synthesis capabilities

- new, more efficient model of logic functionality
- multilevel logic optimization
- path optimization
- partial simulation
- logic/interconnect trade-off
- small number of logic patterns give comprehensive design coverage
- integrated front-end back-end tools and model
- designs generated automatically to meet performance specifications

***ISSUE:*** Synthesis and design iteration require equivalence checking
***SOLUTION:*** Built-in formal verification and goal-directed logic synthesis

- redundancy and reconvergent path removal
- false path identification
- automated test pattern generation
- automated test bench validation

***ISSUE:*** Large designs are hard to manage and to modify
***SOLUTION:*** Powerful abstraction tools for management of complexity

- identification and abstraction of recurrent functional elements
- bit-width abstraction
- library module identification and extraction
- parametric generator functions
- hierarchical partitioning
- control of critical path length
- retiming and register relocation
- design reuse and integration
- identification and isolation of essential components
- sequential/parallel decomposition
- branching and path explosion management
- structure sharing

***ISSUE:*** Functional specification is hard to convert to hardware design
***SOLUTION:*** Design space exploration tools guided by user goals

- time/space trade-off
- modular and localized design trade-offs
- datapath synthesis
- top-down and bottom-up design
- localized changes under parametric control
- fan-in and fan-out control
- pipelining

***ISSUE:*** New tools require new skills
***SOLUTION:*** Compatibility with standard design practices and toolchains

- standard HDL and netlist input
- netlist output at any design stage
- integrates with conventional tool chain whenever desired
- handles all types of logic with equal efficiency