RECOMMENDATIONS for an EMPIRICAL PROGRAM to EVALUATE ILOC SOFTWARE

William Bricken
May 2003

Our objective is to generate substantive and representative data that will provide sufficient evaluative information to make informed business decisions. In particular we wish to demonstrate a competitive advantage of the ILOC software for configuring LUT-based FPGA architectures.

OVERVIEW OF RECOMMENDATIONS

1. Establish a methodology, metrics, and procedures for gathering dynamic performance data for ILOC and for Synplicity configurations of identical designs running in a placed-and-routed Virtex-II architecture.

2. Gather performance data for non-complicated designs with known and well-understood structural characteristics.

3. Then gather performance data for substantive and complex industrial designs of interest.

CONTENTS

Sections of this memo present considerations about our experience with the SP700 design, constraints on existing ILOC software, suggestions for an evaluation methodology, recommendations for an empirical process, and suggested designs for evaluation.

This memo also includes two appendices that add detail to the recommendations.

> SPECIFICATIONS for a DESIRABLE INITIAL ILOC TEST DESIGN
> SELECTION and ANALYSIS of SEVERAL DESIRABLE DESIGNS

Two separate documents provide a contextual background for the recommendations herein:

> SP700 REPORT, May 2003
> PRESENTATION AND NOTES, November 2002

CURRENT STATUS

The SP700 study provided at least three benefits:

1) Proof of principle that ILOC can be applied to a complex industrial design.
2) Extension of ILOC capabilities to handle limited aspects of commercial designs, such as structural VHDL parsing, Xilinx cell libraries, and naive LUT-mapping.
3) Preliminary suggestive data that ILOC is at least competitive with Synplicity in several synthesis functions, such as logic minimization, path reduction, and hierarchical decomposition.

The SP700 study was not appropriate for the competitive evaluation of ILOC for several substantive reasons:

1) ILOC does not include a competitive LUT-mapping capability,
2) The SP700 analysis did not include test vectors and other validity checks,
3) The study was based on abstract metrics rather than on silicon behavior,
4) ILOC capabilities do not currently address the design as a whole.

The SP700 study assessed a very limited range of the scope and capabilities of ILOC. In particular, only logic cell (either NANDs gates or 4LUTs) reduction and incidental critical path reduction were evaluated. ILOC, however, provides many design environment tools that were not exercised, including:

-- critical path length reduction
-- abstraction of functional blocks and signal vectors
-- localized transformations to reach design goals
-- parametric exploration of a design space

Many ILOC features that would enhance an interactive design and development tool for FPGA CLB placement and routing were not assessed by optimization and performance measures of the SP700 study. We suggest that performance metrics of area and speed first be assessed, prior to examining other features of ILOC.

The ILOC value propositions include:

-- pragmatic benefits provided by a unified tool model
-- new synthesis capabilities
-- built-in formal verification and goal-directed logic synthesis
-- powerful abstraction tools for management of complexity
-- design space exploration tools guided by user goals
-- compatibility with standard design practices and toolchains

These capabilities currently exist as algorithms and concepts without a user-interface or a specific application to LUT-mapping, thus they all require significant development effort to evaluate empirically. To assess the broad potential of ILOC, it would be necessary to work with the ILOC software and the mathematics directly. This in turn would require a development and evaluation effort with supported resources.

The EDA capabilities of ILOC are fully integrated into the CoMesh architecture, a reconfigurable silicon platform with none of the inadequacies of LUT-based FPGAs. It is widely acknowledged that EDA synthesis strongly interacts with silicon architecture. LUT-based architectures exhibit inherently poor performance, requiring specialized hardware features such as non-LUT carry chains and hardwired multipliers, as well as a routing overhead that dwarfs the effective silicon area used for logic. Naturally ILOC cannot fix these architectural problems, however it is possible to design a new silicon architecture that fully utilizes the new EDA capabilities of ILOC. CoMesh is such an architecture. On the SP700 design CoMesh was shown to improve logic density by at least 250%, while increasing simulated system speed to 180 MHz.

CURRENT CONSTRAINTS

ILOC is a new technology for logic applied to synthesis of ASIC designs, and specifically to configuration of the CoMesh architecture. As a technology, ILOC has an excellent potential to improve any application of logic, including technology mapping to LUT-based FPGA architectures. We intend to patent Intellectual Property rights to any and all applications of ILOC.

The ILOC code is currently developed and maintained by one person. Thus, without additional resources, the existing code cannot be significantly extended with new capabilities or for new applications.

Currently ILOC does not support LUT-mapping or the features of modern FPGA architectures, such as the Virtex-II. A direct comparison between ILOC and Synplicity performance is not possible since the two tools do not address the same task.

ILOC does not support many features of a complete commercial design, such as memory, a diversity of FF types, multiple clocks, specialize i/o and pin placement, and specialized hardware features of FPGAs. We believe that the only barrier to extending all ILOC capabilities to commercial designs and to FPGA technology mapping is further development resources.

ILOC is not commercial code, and is not sufficiently developed to support commercial testing and evaluation. It can be viewed as proof of principle for a wide diversity of new and unique EDA capabilities.

SUGGESTED METRICS, METHODOLOGY and PROCESSES

We suggest an approach to the competitive evaluation of ILOC for FPGA technology mapping that conforms to current constraints and also provides substantive comparative information.

Metrics

We recommend that ILOC performance be evaluated only in the form of actual dynamic behavior of designs placed and routed in a Virtex chip. For an operational design running in the Virtex environment, Xilinx tools provide both timing and LUT-usage information. We do not recommend statistical measurement based on theoretical or modeled behaviors.

Methodology

Two comparative methodologies are available:

1) Designs optimized by ILOC compared directly to the same designs optimized by Synplicity

2) Designs processed by Synplicity compared to the same design optimized by ILOC and also processed by Synplicity

The head-to-head comparison of Method 1 is most desirable, however, design processing by Synplicity, in the form of LUT-mapping and placement is unavoidable since ILOC does not include a competitive LUT-mapper or a Virtex router. Thus, in any event, ILOC output will inevitably also be processed by Synplicity/Xilinx tools.

Method 2 could indicate directly any improvement of Synplicity synthesis added by ILOC, particularly when applied within the context of a controlled experimental evaluation process..

We recommend trying both methodologies, but modifying designs and cell libraries for improved comparability. Method 1 results may be indicative but not evaluative; Method 2 results can provide the ground-work necessary to guide decisions regarding further development effort.

Process

We see three general processes to achieve competitive evaluation; all use ILOC output in the form of EDIF netlists, and all are converted into Virtex-based LUTs by Synplicity, and then placed and routed into the Virtex environment by Synplicity/Xilinx tools. The three differ in the type of EDIF output generated by ILOC processing:

1) LUT-output using ILOC's simple LUT-mapper
2) Xilinx cell library output using ILOC's pattern matcher
3) Simple cell library output using ILOC's native data structures

ILOC Output of LUTs (Not Recommended)

The ILOC LUT-mapper is not competitive and does not have the capability to use Virtex special features. We believe that utilizing Virtex special features gives an incomparable advantage to Synplicity. Thus in order to reach comparability, any ILOC LUT output would need to be post-processed by Synplicity in order to access Virtex special features. Partial processing by Synplicity is probably unavoidable, leading to the conclusion that LUT-based output from ILOC only serves to emphasize the ILOC LUT-mapper, and thus obscures the actual developed ILOC optimization performance.

This process is thus not recommended. However, since the ILOC LUT-mapper does confer ILOC optimization to the routing complexity of LUTs, it may be of interest to measure the timing and ease of routability of an ILOC LUT-based design. The process of post-processing ILOC LUT output by Synplicity in order to achieve Virtex placement and routing provides evaluation information only for ILOC improvement of routing, and not for overall LUT usage.

ILOC Output of Xilinx Library Cells (Not Recommended)

Original designs expressed in the Xilinx cell library can be fully processed by Synplicity. Similarly, the same original design can be optimized by ILOC and then passed through Synplicity for placement and routing. For comparability, Synplicity will at least have to convert the Xilinx cell-based design into LUTs that incorporate Xilinx special CLB features, so that at least some Synplicity processing is unavoidable.

Two versions of the suggested process are possible: ILOC optimized output that is LUT mapped, and then placed and routed by Synplicity with either Synplicity optimization turned on or turned off. In the case of Synplicity optimization turned on, both speed and area optimization by Synplicity should be considered. Optimization settings for Synplicity may also be varied equally for ILOC and non-ILOC processed input. The comparative performance assumption is that ILOC reduction provides a better initial design than can be achieved solely by Synplicity reduction.

A further complexity with this approach is that we do not know which Xilinx cells are preferred by the Synplicity mapper. Without this information, the selection of Xilinx cells used by ILOC may not be the best. This concern is mitigated by the assumption that a designer using the Xilinx cell library may not select preferred cells either. However, increasing emphasis on the special case capabilities of the Synplicity LUT mapper only decreases the

quality of the evaluative data for ILOC. The additional complexity of cell selection can be mitigated by the next process

ILOC Output of Simple ASIC Cells (Recommended)

This process is similar to the one above, but incorporates a prior processing step of "leveling the play field". A design is first reduced to simple gate structure, such as N-input NANDs or NORs. It is submitted to Synplicity optimization. The same design expressed in simple logic cells is submitted to ILOC processing and then submitted to Synplicity for place and route. The process would measure head-to-head improvement of a design prior to place and route. The optimized initial design that results in a preferred place and route, and consequently better FPGA performance, is empirically better.

To assess ILOC capabilities without the added benefit of Synplicity, the above process can be followed using two conditions: with and without Synplicity optimization engaged. A design expressed in a simple gate library would be placed and routed by Synplicity/Xilinx, and the same design expressed in the same simple gate library would be first optimized by ILOC and then place-and-routed by the same Synplicity/Xilinx process.

This process is recommended. It measures the dynamic performance of several versions of a design, all placed and routed by the same Synplicity/Xilinx processes after reduction. Performance of the following processed designs, all placed and routed by Synplicity, would be measured:

A. not reduced by either tool
B. reduced solely by ILOC
C. reduced solely by Synplicity
    C1. using speed optimization settings
    C2. using area optimization settings
D. reduced by ILOC and then by Synplicity (using two settings)
E. reduced by Synplicity (using two settings), then by ILOC

Process A provide baseline performance measures, while Processes B and C provide measures of the performance gain conferred by either tool individually. It should be expected that both tools provide similar gains. Process D provides the pre-processing advantage of ILOC, while Process E provides the differential benefit of ILOC added to Synplicity reduction.

In this approach, pre-processing the original design into a simple gate library minimizes non-comparative customization advantages, while the placement and routing by Synplicity of all designs assures a far comparison of ILOC's existing capabilities.

For ILOC to demonstrate a competitive advantage, these conditions might be met:

1. (B - A) better than (C - A)
2. B better than C
3. D better than C
4. E better than C

SUGGESTED DESIGNS

For the methods and processes recommended above, the goal is to assess competitive performance, not to assess ILOC performance for complex commercial designs. To separate these two independent goals, it is important to assess both tools on as non-complicated designs as possible. This removes potential contamination of results due both to aspects of a complex design that ILOC does not yet handle, and to advantages differentially incorporated into the commercial Synplicity tools.

The SP700 design study provides sufficient proof that ILOC can handle commercial designs, so we recommend that decisions concerning a more detailed assessment of ILOC's real-world capabilities be based on the firm foundation of performance comparison for designs that are controlled for complexity and measurement complications.

Thus, we recommend a suite of standard well-understood design fragments for initial assessments, enriched by designs that meet the selection criterion of absence of complicating factors. These factors are listed in Appendix I: SPECIFICATIONS for a DESIRABLE INITIAL ILOC TEST DESIGN.

A list of 20 designs that meet all selection criteria is included as Appendix II: SELECTION and ANALYSIS of SEVERAL DESIRABLE DESIGNS. All of these relatively clean design fragments are used in complete commercial designs, and in fact components that have the identical functionality of several of them occur in the SP700 design. We emphasize that real-world commercial designs are "messy", but this additional complexity only obscures comparative performance measurements. ILOC competitive performance on real-world designs can be assessed independently and at a later time, assuring that further business decisions are based first on clearly interpretable data. Since ILOC is not yet sufficiently developed to be accurately assessed for real-world designs, such assessment should naturally come later in the evaluation process, and be built on data gathered from more tidy design components, data that is both less resource expensive and easier to interpret.

The suggested suite of 20 designs provide several baseline circuits to assure process comparability, several circuits that are known to be difficult to optimize, circuits that ILOC does well on and those that ILOC does poorly on, and several large circuits that are included in pre-optimized IP packages from major vendors. All have extensive test-vectors, published performance results, and all are in common usage. All meet the clearly interpretable

criterion, and all are challenging to any FPGA synthesis and place and route tool.

SUMMARY OF THE RECOMMENDED EVALUATION METHOD AND PROCESS

To address the goal of determining ILOC competitive advantage over Synplicity for FPGA optimization:

1. Select several designs that can provide clearly interpretable results, that represent a wide variety of common design structures, and that are not contaminated by components that are not addressed by ILOC tools.

2. Convert these designs into a simple gate library to achieve initialization comparability.

3. Use Synplicity/Xilinx to place and route the following versions of the initialized designs:

   A. no optimization by either ILOC or Synplicity
   B. optimized by ILOC only
   C. optimized by Synplicity only (using area and speed settings)
   D. optimized by ILOC and then by Synplicity
   E. optimized by Synplicity and then by ILOC

All file transfers are to use simple logic gates in EDIF 2 0 0 format. Appendix IV: A SAMPLE VERY SIMPLE EDIF DESIGN SPECIFICATION provides an example.

4. Measure dynamic behavior for all of the above cases, in terms of achieved number of LUTs used, and achieved design timing.

5. Should results warrant, invest development resources to extend ILOC to completely optimize complex real world designs, and then compare the extended ILOC to Synplicity on several commercial designs using the same process as above.

6. Should the further results warrant, extend ILOC capabilities to cover the entire EDA capabilities of Synplicity, such as tools for FPGA place and route, interactive user design, and customization preferred silicon platforms. Also extend EDA capabilities by new and unique tool functionality available only with ILOC, such as dynamic critical path customization, automated design abstraction, and design space exploration.

APPENDIX I:
SPECIFICATIONS for a DESIRABLE INITIAL ILOC TEST DESIGN

The i/o constraints of the current prototype ILOC implementation follow:

Input: EDIF 2 0 0 format. Recently extended to limited cases of structural VHDL. Limited parsing capabilities also exist in ILOC for BLIF, KISS, and VERILOG. ILOC can expand hierarchical EDIF modules when necessary.

Output: EDIF 2 0 0 format. Recently extended to limited cases of structural VHDL.

Gate types: all standard logic gates, any number of inputs. No non-standard logic such as memory blocks or tri-state devices.

Register types: ILOC currently uses only D flip-flops. It is possible to extend ILOC to other FF types, however extending the ILOC simulator to incorporate a diversity of FF behaviors would require substantial development effort. The CoMesh architecture dictates how registers are structured; ILOC currently converts all register types to those consistent with the CoMesh model.

Circuit size: ILCO can process any size that best answers evaluation questions without introducing unnecessary complexities. Something under ~20K two-input ASIC gates would be most convenient.

Performance targets: It would be quite helpful to know the performance characteristics of any test circuits, as well as differential performance goals. ILOC has many different optimization processes, each optimizing to a different type of criteria (speed, density, technology map, modularization, etc.).

Validation: Test vectors are mandatory.

APPENDIX II:
SELECTION and ANALYSIS of SEVERAL DESIRABLE DESIGNS

Twenty benchmark circuits have been identified to provide comparative performance measurements for FGPA LUT-based optimization. They are listed below, ordered by size and separated by type. Gate counts marked by an asterisk are for a two-level logic PLA version of the circuit functionality. "Pins" refers to the number of internal input pins to all gates.

			PINS		ASIC GATES	
NAME	*FUNCTION*	*I/O/REG*	*ORIG*	*RED*	*ORIG*	*RED*
COMBINATIONAL						
cm85a	mag comp	11/ 3	120	42	39	22
9symml	count	9/ 1	372	214	221	151
c5315	ALU+select	178/123	4558	1516	1807	475
dalu	dedicated ALU	75/ 16	4458	1489	1809	863
cordic	2-level	23/ 2	4046	120	2888*	51
c6288	16-mult	32/ 32	7152	2562	2337	1860
des	encrypt	256/245	8735	4092	6630	2040
too_large	logic	38/ 3	15563	634	14461*	460
SEQUENTIAL						
mult16a	mult	18/ 1/ 16	488	188	271	93
s838	fract mult	36/ 2/ 32	862	305	310	170
s1423	logic	18/ 5/ 74	1640	640	546	400
keyb	FSM	9/ 2/ 5	804	314	548*	198
sbc	bus control	41/ 56/ 27	1611	870	680	576
mm30a	minmax	34/ 30/ 90	3339	1375	1690	541
dsip	encrypt	229/197/ 224	5755	2992	2747	1603
bbara_bbtas	FSM	6/ 2/ 7	3595	1001	2925*	676
s298	PLD FSM	5/ 1/ 68	7164	2154	6142*	1665
bigkey	encrypt	263/197/ 224	11325	5334	8888*	2163
s38417	logic	29/106/1465	33170	10830	10634	5924
clma	bus interface	383/ 82/ 33	44872	26051	24216	16367

The rationale for inclusion of each circuit follows. In general, selections possess structural features known to challenge logic optimization, and place and route algorithms. Features emphasized by different selections include:

-- base cases to establish consistency of comparisons
-- large count of inputs, outputs, and/or registers
-- common and frequent usage of circuit functionality
-- common function types such as multipliers, FSMs and ALUs.
-- highly regular/irregular logic structure
-- low/high branchiness of logic structure, primarily XOR and MUX gates

-- exceptionally wide and/or deep functions
-- large fanin/fanout
-- irregular and difficult fanin/fanout shapes over function depth
-- long critical paths between registers
-- many register feedback paths
-- poor/good reduction performance by ILOC
-- known poor/good optimization performance by EDA tools

NAME	*SELECTION RATIONALE*
cm85a	base
9symml	base, little structure, one output, pyramid, poor reduction
c5315	ALU, high i/o, very many xor/mux
dalu	ALU, wide and deep with tail, poor reduction
cordic	common, reduction, 2-level pyramid
c6288	arith, very deep and wide with tail, no xor/mux
des	common, large, high i/o, many mux, wide and shallow
too_large	large, reduction, no xor/mux, 2-level
mult16a	arithmetic, deep and narrow
s838	arithmetic, mid bulge, high feedback
s1423	deep and narrow
keyb	small FSM, 2-level
sbc	common, bus, wide and shallow with tail, pyramid
mm30a	arithmetic, very deep at both ends, very narrow in middle
dsip	high i/o, high registers, very wide and shallow, high feedback
bbara_bbtas	common FSM, 2-level, wide and shallow, muxed regs, high feed
s298	2-level FSM, very wide, huge ors
bigkey	large, many mux, many registers, 2-level, wide and shallow
s38417	large, many xor/mux, many registers, very wide, shallow
clma	large, bus, very wide and very deep with tail

RANK ORDER of IMPORTANCE

A selection of several of these proposed benchmark circuits would be appropriate for the comparative performance evaluation between Synplicity and ILOC; evaluating relative performance on all would be ideal. In the case of a limited selection, the preferred prioritization follows:

NAME	*TYPE*	*ASIC GATES*
des	C	6630
cordic	C	2888
clma	S	24216
dsip	S	2747
bbara_bbtas	S	2925
mm30a	S	1690
dalu	C	1809
sbc	S	680
c6288	C	2337
bigkey	S	8888
c5315	C	1807
mult16a	S	271
s38417	S	10634
too_large	C	14461
s298	S	6142
s1423	S	546
s838	S	310
keyb	S	548
9symml	C	221
cm85a	C	39