TECHNICAL DESCRIPTION OF LOSP/PUN (short)

William Bricken
April 2001

Boundary Logic (BL)

The essential mathematical innovation in BL is *void-based reasoning*. The idea is to use *non-representation* (i.e. the void, recording nothing at all) as a logical element. For example, if True were represented by a particular token, then False could be "represented" by the absence of that token, that is, by nothing at all

We have developed trade secret understanding, algorithms, implementations, and methods for void-based computation. The attendant contribution is an improvement of computational algorithms and data structures using logic.

Losp, a suite of computer programs for logic optimization

Losp implements BL. The Losp algorithms improve the implementation of applications involving logical deduction and manipulation, including

- circuit specification and design
- theorem proving and logic verification
- relational database queries and management
- knowledge-base optimization and minimization
- declarative programming
- procedural program control structure optimization
- planning
- data mining
- formal specifications.

As well Losp has unique application advantages for

- parallel processing
- visual, visceral, and experiential programming languages.

Losp includes tools and methods for:

- logic evaluation and case analysis
- logic standardization
- equivalence checking and tautology identification
- redundancy removal and logic minimization
- Boolean factoring and network decomposition
- Boolean pattern abstraction
- Boolean constraint reasoning.

Although BL is inherently parallel, the current Losp algorithms are implemented on a sequential architecture. Techniques for reducing the Losp software algorithms to hardware are understood, but have not yet been explored. These could conceivably provide an alternative model of silicon computation with distinct technical advantages.

Pun, a suite of Losp application programs for circuit minimization

Pun is a substantive collection of unique trade secret data structures and transformations which are particularly tuned for Losp manipulation, reduction, and minimization of circuit structure.

Pun development has focused on optimization of combinational circuitry, a domain originally selected for the complexity of its logical structures. Efficient Boolean minimization and factorization are unsolved problems in the logic synthesis community. No tools exist for circuit optimization across multiple criteria.

Pun includes tools and methods for

- redundancy removal and minimization
- don't care analysis and minimization
- critical path modification
- network decomposition and transformation
- specification of maximum fanin and fanout
- structure abstraction
- library mapping
- structure sharing
- test vector generation and BDD generation
- technology mapping.

The Losp and Pun algorithms perform efficient Boolean minimization on over 200 benchmark circuits. These algorithms provide automated design capabilities with competitive advantages which include:

- comprehensive coverage of logical and arithmetic circuits
- formal correctness and verification
- multi-criteria minimization
- around 15% improvement on most optimization criteria, relative to existing tools
- shorter design times, from weeks to hours (one order of magnitude improvement).

The tools and techniques implemented in Pun for combinational circuit logic are readily generalizable to *timed Boolean logic* (sequential circuits), providing automated time/space tradeoff and optimization for sequential circuit design.