Synthesis Flow Framework for IC Design

Tech ID: 34293 / UC Case 2016-858-0

BACKGROUND

Digital integrated circuit design has evolved significantly over the past several decades, with synthesis becoming increasingly automated and sophisticated. The traditional synthesis flow emerged in the 1980s when commercial logic synthesis packages from companies like Cadence and Synopsys revolutionized chip design by automatically converting hardware description languages (HDL) into gate-level netlists. Electronic design automation (EDA) tools evolved from simple netlist extraction to complex optimization processes, progressing through gate-level optimization, register-transfer-level synthesis, and eventually algorithmic synthesis. However, as designs have grown exponentially in complexity, synthesis times have become a major bottleneck, with full synthesis often taking hours or days for large designs, significantly impacting designer productivity and iteration cycles. Long synthesis runtimes prevent designers from rapid iteration, with typical synthesis taking 3+ days for complex designs, forcing designers to carefully consider when to submit jobs and wait for delayed feedback. The traditional register-transfer level (RTL) design flow suffers from critical limitations including the inability for RTL engineers to identify and resolve top-level timing issues early in the design process, routing congestion problems that cannot be detected until placement is completed, and insufficient feedback on power consumption during early architectural phases. Additionally, even small design changes trigger full re-synthesis of large blocks, wasting computational resources on unchanged portions of the design, while inter-module optimization requirements often degrade quality-of-results (QoR) when designs are artificially partitioned.

TECHNOLOGY DESCRIPTION

To help address these challenges, researchers at UC Santa Cruz (UCSC) have developed approach to design partitioning known as functionally-invariant boundaries (FIBs) which automatically identify regions whose functionality remains unchanged during synthesis rather than relying on artificial or hierarchical boundaries. The invariant cone methodology enables intelligent partitioning of designs into regions containing 1000-5000 gates based on functional invariance rather than arbitrary module boundaries, ensuring optimal synthesis quality while maintaining manageable partition sizes. The interactive incremental synthesis framework achieves near real-time feedback (within seconds) by synthesizing only the small regions affected by changes, representing a 10x improvement in synthesis speed over traditional approaches. The system employs a sophisticated netlist differential algorithm that performs structural comparison between elaborated netlists to precisely identify which invariant cones have been modified, avoiding unnecessary traversal of the entire design graph. Moreover, the netlist stitching technique seamlessly reintegrates synthesized modified regions back into the original netlist while managing gate count tracking and removing unused components, ensuring design integrity throughout the incremental process.

APPLICATIONS

- ASIC design and development
- FPGA design and development
- AI/ML hardware accelerator design and development

FEATURES/BENEFITS

- FIBs eliminate arbitrary partitioning decisions and preserves inter-module optimizations, maintaining design quality while enabling targeted incremental synthesis,
- Real-time synthesis feedback (< 30 seconds) matches human short-term memory cycles, enabling continuous design optimization and immediate validation of design changes without workflow disruption.
- Automatic invariant cone partitioning reduces synthesis time by 10x on average by synthesizing only affected regions (typically < 5% of total design), improving designer productivity.

CONTACT

Marc Oettinger
marc.oettinger@ucsc.edu
tel: 831-502-0253.

OTHER INFORMATION

KEYWORDS

integrated circuit, IC, ICs, integrated circuit design, EDA, electronic design automation, register-transfer level, RTL, hardware description language, Verilog, VHDL, LiveHD, synthesized netlist, netlist, synthesis, hardware design, partitioning, FPGA, ASIC

CATEGORIZED AS

- **Computer**
  - Hardware
  - Software
- **Semiconductors**
  - Design and Fabrication

RELATED CASES

2016-858-0

▶ Netlist differential analysis algorithm provides more precise identification of changed regions through structural comparison, avoiding full design traversal.

INTELLECTUAL PROPERTY INFORMATION

Country	Type	Number	Dated	Case
United States Of America	Issued Patent	10,614,188	04/07/2020	2016-858

Additional Patent Pending

RELATED MATERIALS

▶ Wang, S. H., Possignolo, R. T., Skinner, H. B., & Renau, J. (2020). Livehd: A productive live hardware development flow. IEEE Micro, 40(4), 67-75. - 05/22/2020