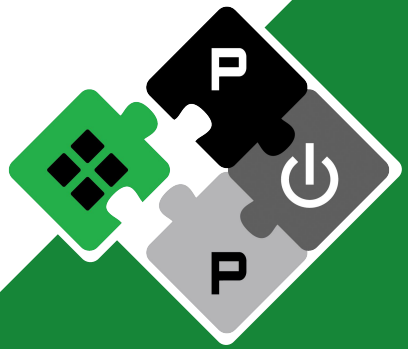
ETH zürich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Open-Source Logic Synthesis: ABC, mockturtle, and Beyond

Integrated Systems Laboratory (ETH Zürich)

Yukio Miyasaka ymiyasaka@iis.ee.ethz.ch



pulp-platform.org

@pulp_platform

company/pulp-platform

youtube.com/pulp_platform

PULP Platform

Open Source Hardware, the way it should be!

Self Introduction

Name: Yukio MIYASAKA

Position: Postdoc at ETHZ

- **We're now maintaining mockturtle (originally developed at EPFL)**

Education:

- **University of Tokyo (BE, ME)**
- **University of California, Berkeley (PhD)**
  - Supervised by Alan (Developper of ABC)

Research interests:

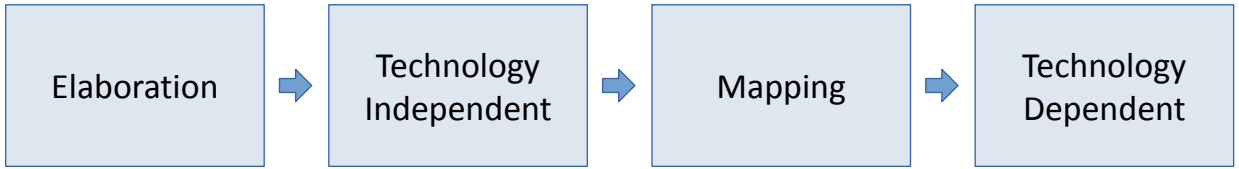
- **Logic Synthesis**
- **Formal Verification**

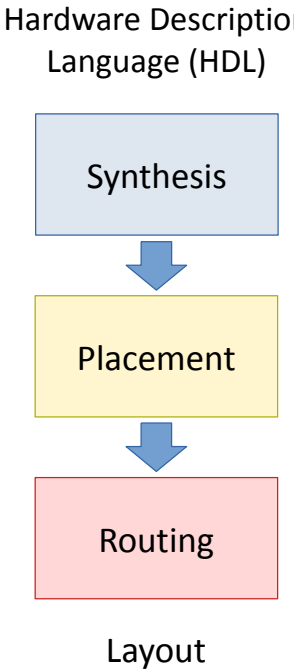
Outline

- **Overview of open-source logic synthesis tools**
- **Logic representations**
- **Optimization algorithms**
- **Current gap and future directions**

Logic Synthesis

- **First step in the electronic design automation flow**
  - Generates a topology of logic circuit
- **Further divided into 4 substeps**
  - Elaboration is a general preprocessing step
  - Technology-independent step optimizes circuits in a simple form
  - Mapping maps the circuits to technology library cells
  - Technology-dependent step further optimizes them in the mapped form





Open-Source Logic Synthesis Tools

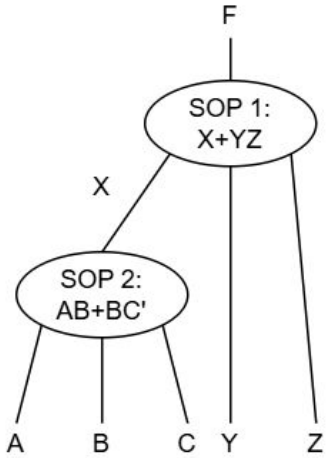
- **ESPRESSO/MIS/SIS** (R. K. Brayton, A. Sangiovanni-Vincentelli, at al. since 1982/1987/1992)
  - Historic logic synthesis tools
  - Based on Sum Of Products (SOP) and Binary Decision Diagram (BDD)
- **ABC** (A. Mishchenko, R. K. Brayton, et al. since 2005)
  - Technology-independent optimization & mapping
  - Centered on And-Inverter Graph (AIG) and Boolean satisfiability (SAT) solvers
  - (Also known as a formal verification backend e.g. pdr)
- **Yosys** (C. Wolf et al. since 2012)
  - Frontend/Elaborator
  - Handles complex sequential elements, macros, and other building blocks
- **mockturtle** (M. Soeken, H. Riener, G. De Micheli, et al. since 2018)
  - Technology-independent optimization redesigned to accommodate various representations such as XOR and Majority gates
  - Enhanced mapping and technology-dependent optimization

Open-Source Logic Synthesis Tools

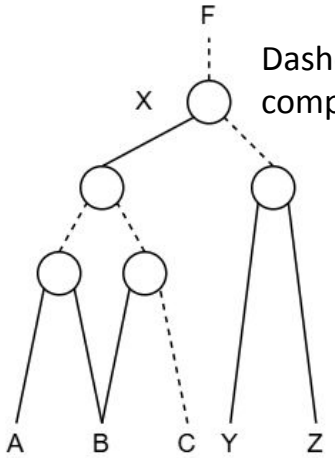
- **ESPRESSO/MIS/SIS** (R. K. Brayton, A. Sangiovanni-Vincentelli, at al. since 1982/1987/1992)
  - Historic logic synthesis tools
  - Based on Sum Of Products (SOP) and Binary Decision Diagram (BDD)
- **ABC** (A. Mishchenko, R. K. Brayton, et al. since 2005)
  - Technology-independent optimization & mapping
  - Centered on And-Inverter Graph (AIG) and Boolean satisfiability (SAT) solvers
  - (Also known as a formal verification backend e.g. pdr)
- **Yosys** (C. Wolf et al. since 2012)
  - Frontend/Elaborator
  - Handles complex sequential elements, macros, and other building blocks
- **mockturtle** (M. Soeken, H. Riener, G. De Micheli, et al. since 2018)
  - Technology-independent optimization redesigned to accommodate various representations such as XOR and Majority gates
  - Enhanced mapping and technology-dependent optimization

Logic Representations

- **AIG** is the modern synthesis standard, taking over **SOP network**
- **Look-Up Table (LUT) network** is an FPGA-native representation but used during optimization even for ASIC

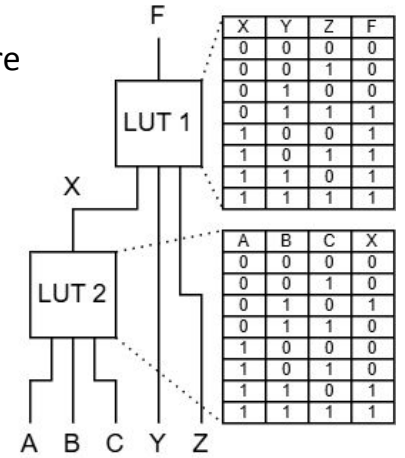


SOP Network



Dashed edges are complemented

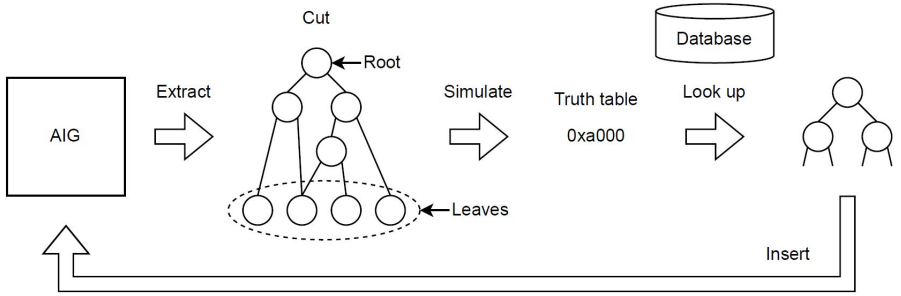
AIG



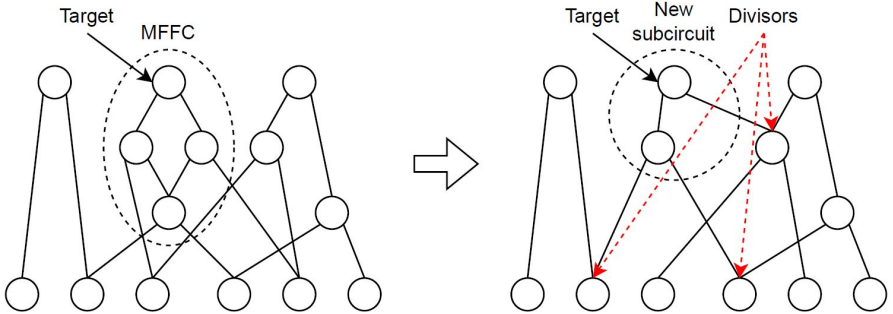
LUT Network

Optimization Algorithms

- **Rewriting**
  - Based on a precomputed database
  - Replace a target subcircuit (cut) with the one recorded in the database
- **Resubstitution**
  - Dynamically create a new subcircuit that can replace a target node
  - Many different heuristics exist
  - SAT sweeping (0-node resubstitution)
- **Delay (depth) optimization**
  - Tree balancing
  - Decomposition methods
  - Lazy man's synthesis (memorization)



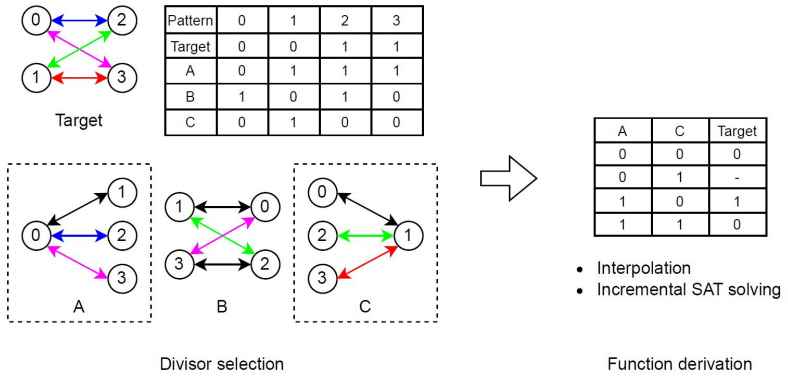
Rewriting

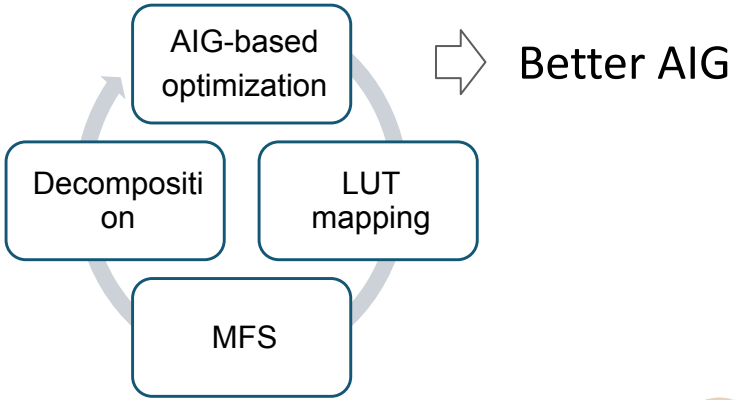


Resubstitution

High-Effort Optimization

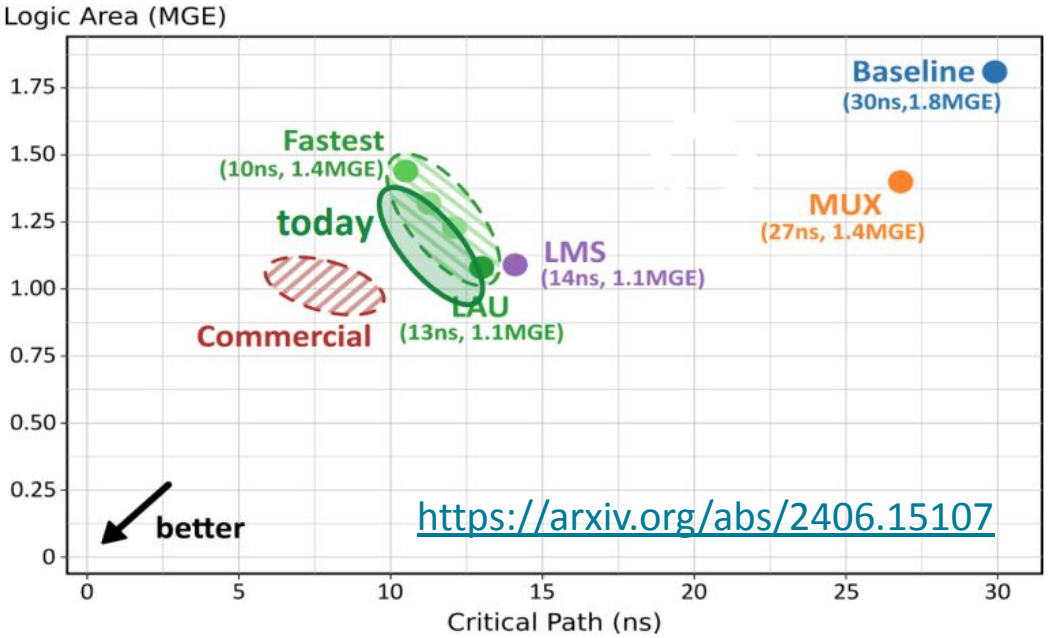
- **MFS**
  - LUT-based resubstitution using don’t-cares
  - Find divisors to distinguish every onset-offset pair using the SAT solver
- **Deepsyn**
  - Niklas Een proposed iterating mapping and unmapping for AIG optimization
  - Deepsyn interleaves MFS with AIG-based optimization
- **Transduction/rewiring**
  - Edge addition and removal
  - Uses don’t-cares more eagerly





Existing Gap

- In previous study, our group improved area (1.6 ×) and delay (2.3×) by using better selector/arithmetic circuits and applying Lazy Man's Synthesis (LMS)
- Still off by ~20% compared to commercial tools



Future Directions

- **Hierarchy management**
  - Grouping/ungrouping
  - Partitioning
- **Datapath optimization**
  - Arithmetic restructuring
  - Sharing/unsharing
- **Sequential optimization**
  - Retiming
  - Sequential don’t-cares
- **Physical awareness**
  - Timing analysis
    - Buffering/Sizing
  - Fake placement

- **PULP benchmarks**
  - Under preparation
  - 16 designs including RISC-V cores, accelerators, SoCs, and many-core systems
  - 200k-500k cells for a core
  - 500k-10M for an SoC/accelerator
  - 10M-100M for a many-core systems
- **Let EDA tools tackle real designs!**

Thank you!

Institut für Integrierte Systeme – ETH Zürich
Gloriastrasse 35
Zürich, Switzerland

DEI – Università di Bologna
Viale del Risorgimento 2
Bologna, Italy