PRO-ME:

Postroute Resource Optimization with Merging ECOs of Multiple Instantiated Blocks

Ashkan Vakil, Ramesh Murugesan
NVIDIA

Outline

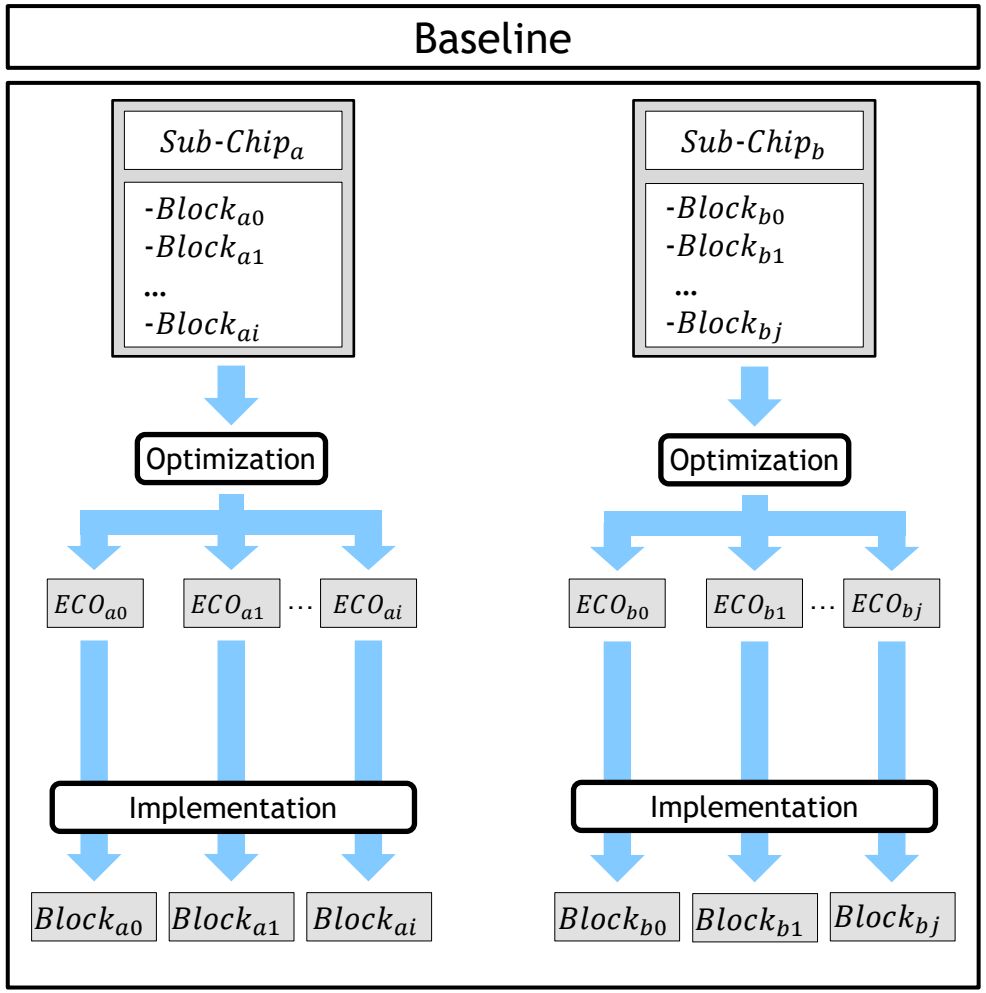
Outline

- Introduction
- Challenges
- Proposed Model
- Results

Introduction

Introduction

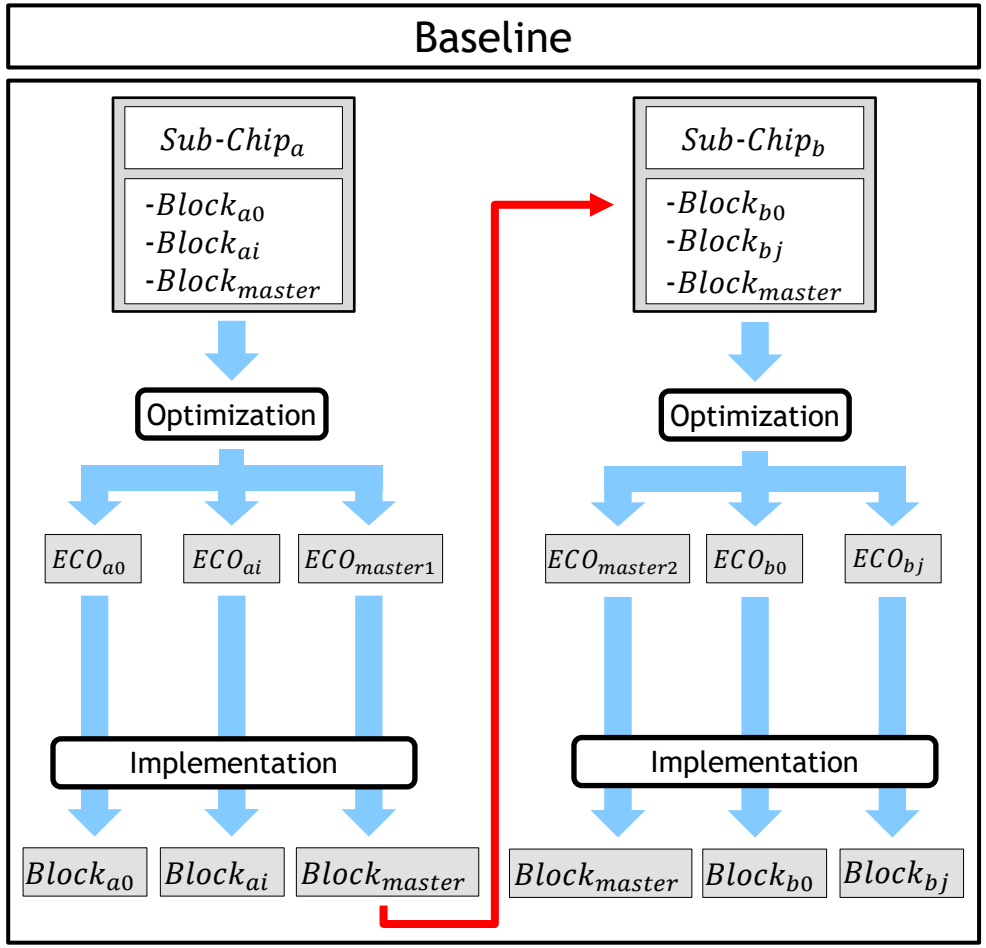
- Lower technology node:
  - Design complexity
  - Increased runtime
  - increased resource usage
- Break chips into Sub-Chips
  - Introducing pipelining and parallel PPA optimization
  - Reduce runtime of each job



Challenges

Challenges

- block instantiation in different Sub-Chips
  - $Block_{master}$ in the figure
  - Introduces serial dependency
    - Turnaround time increases because of wait-time
  - Implementing same block multiple times
    - Unnecessary resource usage

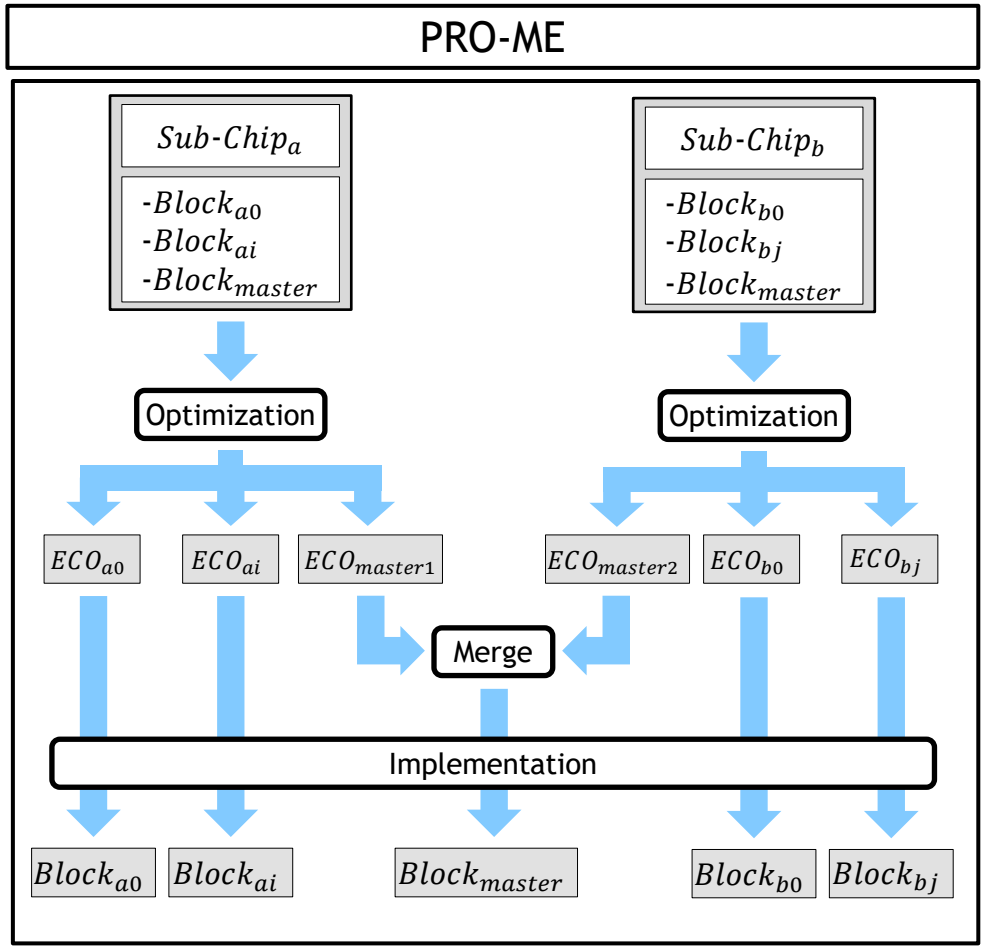


Proposed Model

Proposed Model: PRO-ME

Case 1: with one shared block

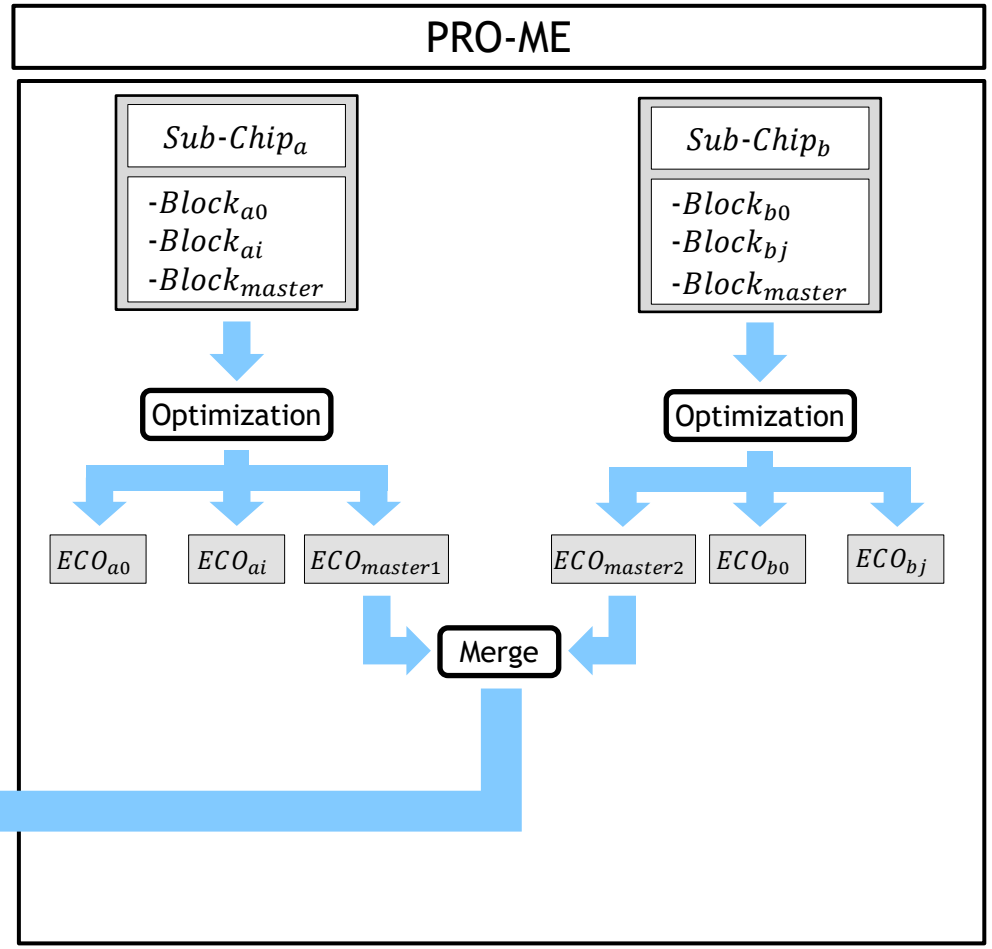
- merge step
  - Merge ECO feature in Tweaker tool
- Remove serial dependency
- Remove unnecessary implementation job



Proposed Model: PRO-ME

Case 1: with one shared block

- merge step
  - Merging ECOs of a block from different optimization sessions
  - Conflict analysis:
    - Logical connectivity conflicts
    - Physical conflicts

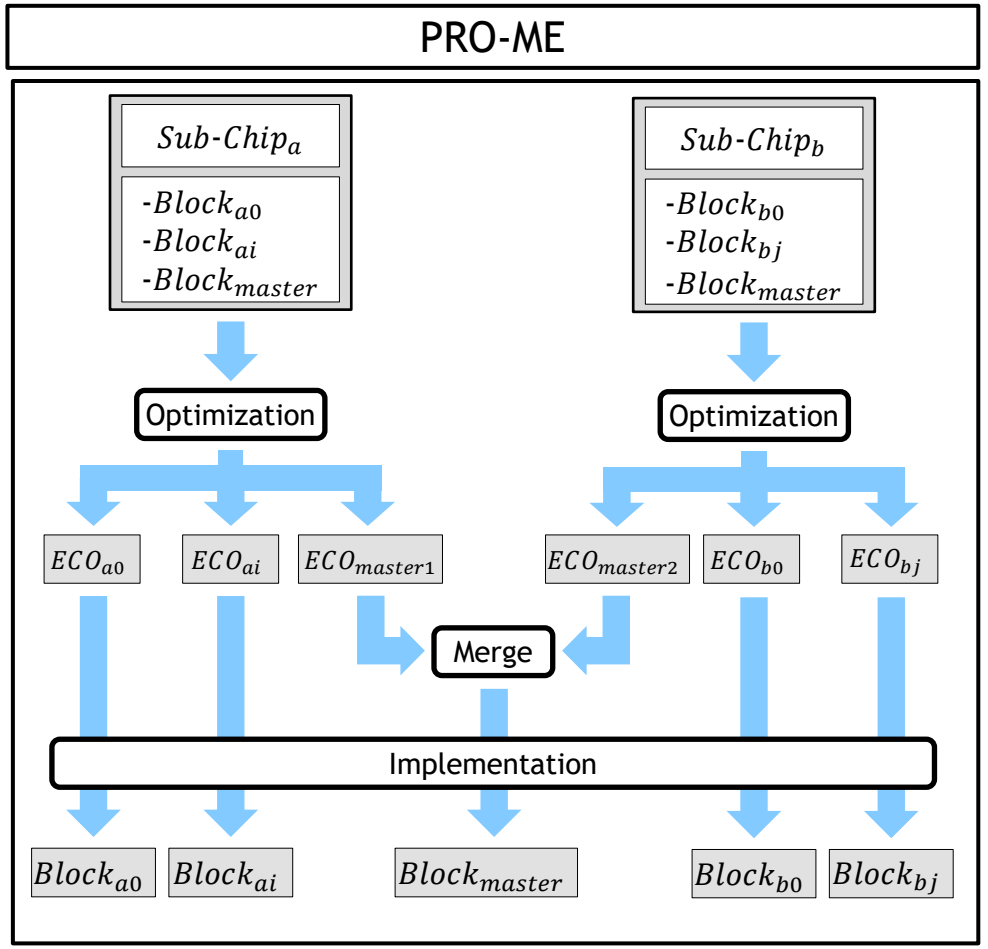


tweaker > load netlist collaterals
tweaker > slackin –nlcmd $ECO_{master1}$
tweaker > slackin –nlcmd $ECO_{master2}$
tweaker > source –skip_conflict $ECO_{master1}$
tweaker > source –skip_conflict $ECO_{master2}$

Proposed Model: PRO-ME

Case 1: with one shared block

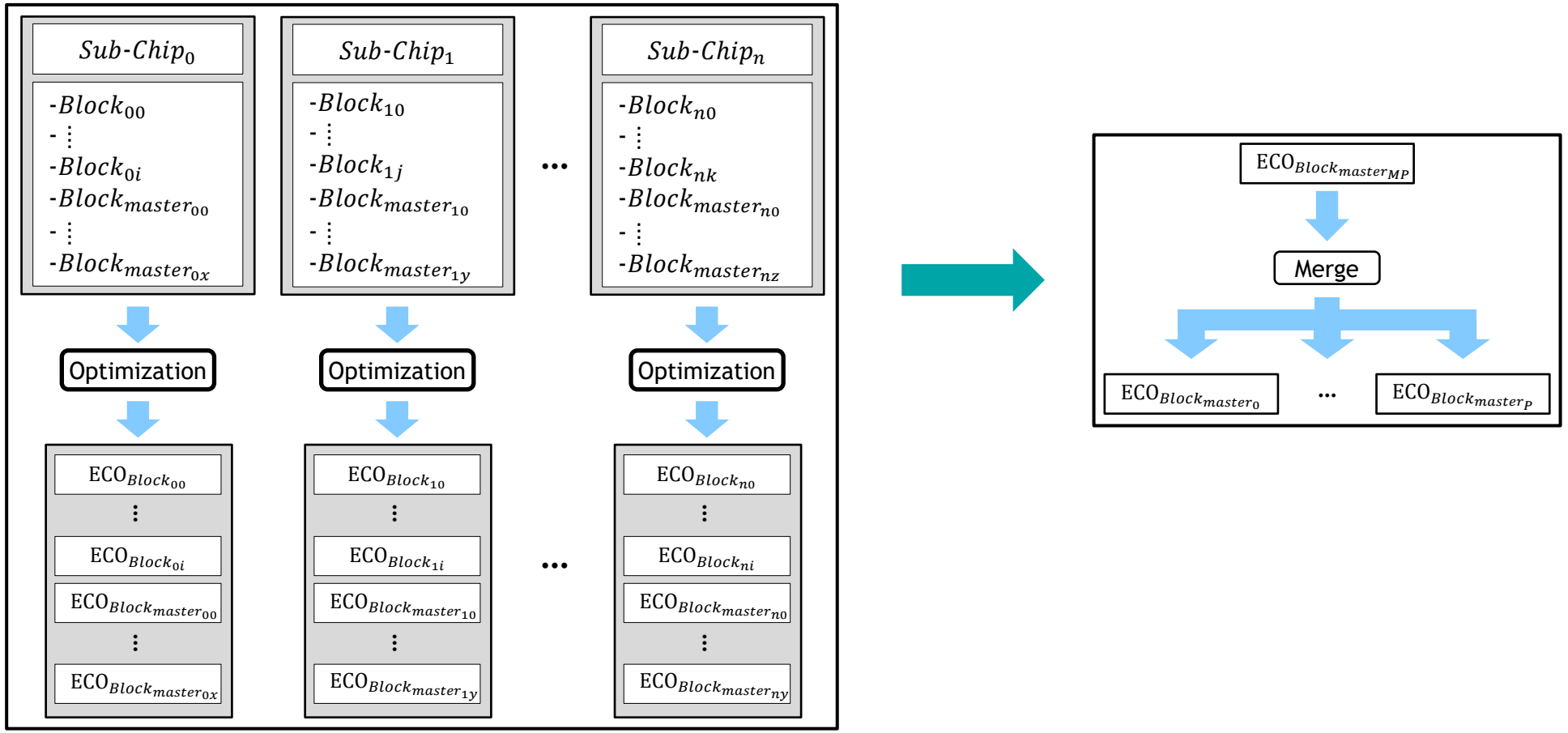
- merge step
  - Merge ECO feature in Tweaker tool
- Remove serial dependency
  - $Sub\text{-}Chip_a$ and $Sub\text{-}Chip_b$ use same netlist for PPA
  - Run PPA optimization simultaneously
- Remove unnecessary implementation job
  - One ECO file for $Block_{master}$
  - Only one implementation



Proposed Model: PRO-ME

Case 2: multiple shared blocks

- Multiple instantiation



Results

Results

- Testcase:
  - Two sub-chips, each with 12 blocks
  - 2 blocks are instantiated in both sub-chips
- Runtime improvement
- Better Quality of Results in timing

Metric	Value
% Runtime Improvement	47.32%
% Resource Reduction	4.35%

Metric	Setup % Improvement	Hold % Improvement	Transition % Improvement
Worst Negative Slack	58.66%	45.27%	0.02%
Total Negative Slack	97.54%	98.06%	86.87%
Endpoints	89.50%	95.13%	39.12%


THANK YOU
YOUR
INNOVATION
YOUR
COMMUNITY