APEX: A Framework for Automated Processing Element Design Space Exploration using Frequent Subgraph Analysis

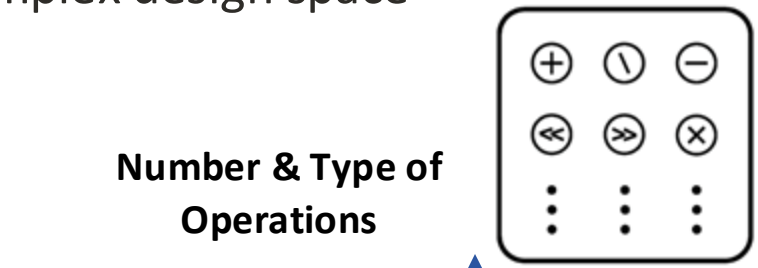
Jackson Melchert

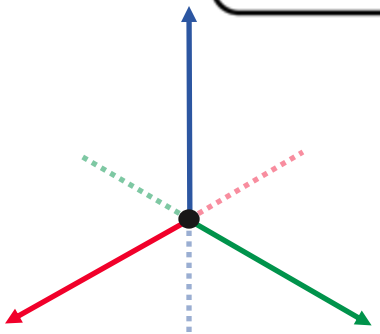
Stanford University

Processing Element Design Space

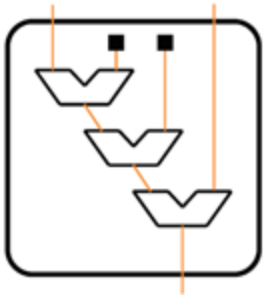
CGRA PEs have a complex design space

Number & Type of Operations





Intraconnect
(Connections between operations inside the PE)



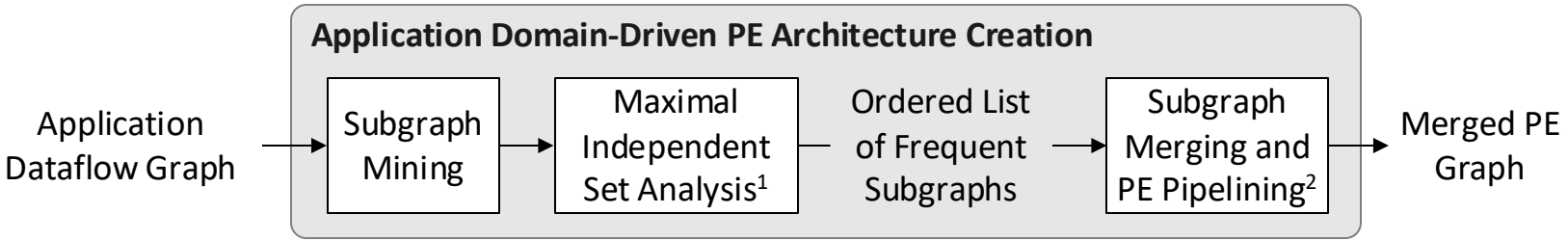
I/O to Interconnect

Key Challenges and Our Solutions

Challenge 1	**Challenge 2**	**Challenge 3**
A naïve exploration of the design space leads to too many candidate PEs	Risk of overspecialization towards the applications chosen for analysis	Evaluating the efficacy of PE designs on applications requires a functioning compiler
Generate candidate PEs by finding frequently occurring operations in the applications themselves	Merge frequent operations into a baseline PE with a general-purpose instruction set	Leverage the Peak DSL to automatically synthesize rewrite rules for the compiler for each PE

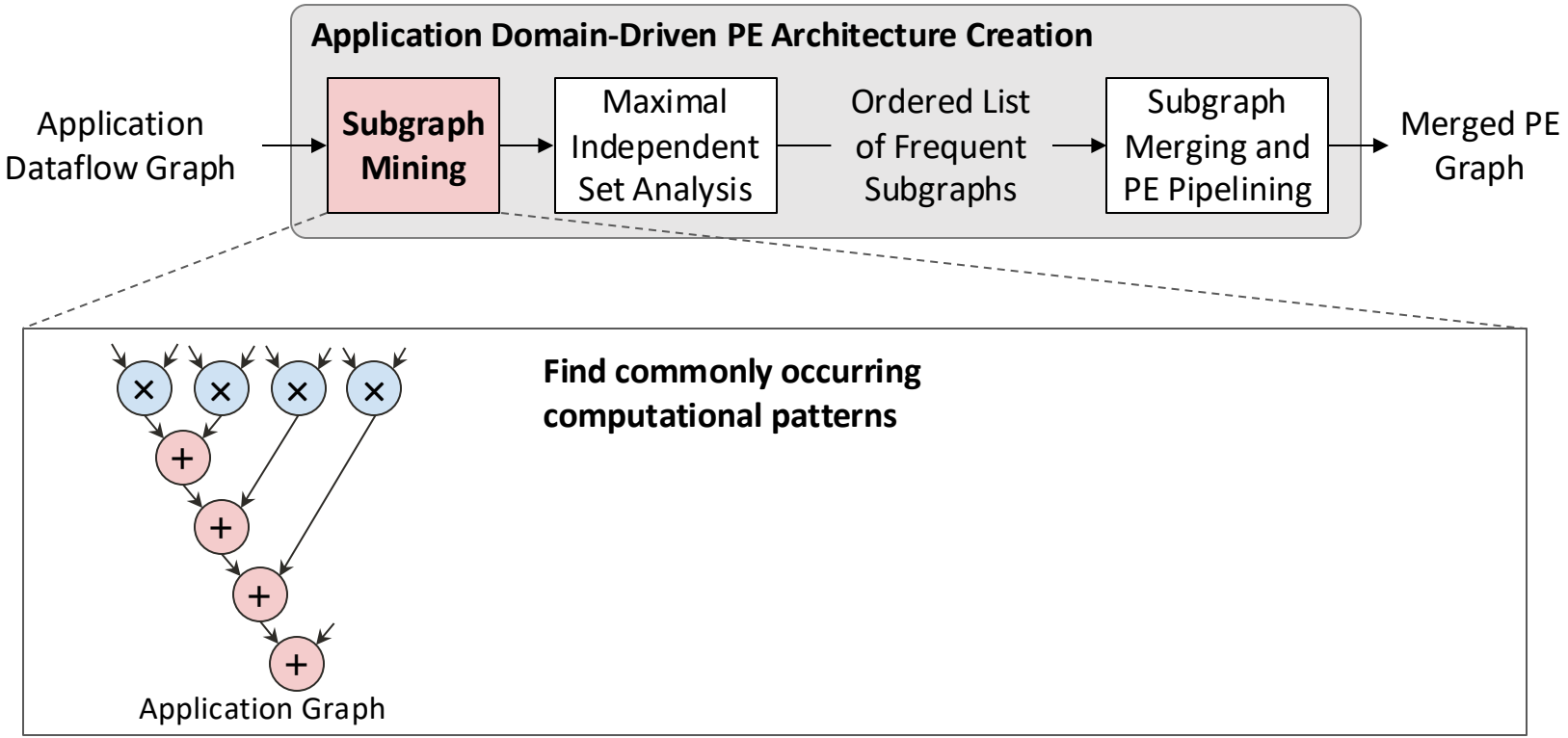
APEX (Automated PE Exploration) encompasses application analysis, PE specification, and CGRA hardware and compiler generation to create an end-to-end flow for PE DSE

APEX: Application-Driven PE Exploration

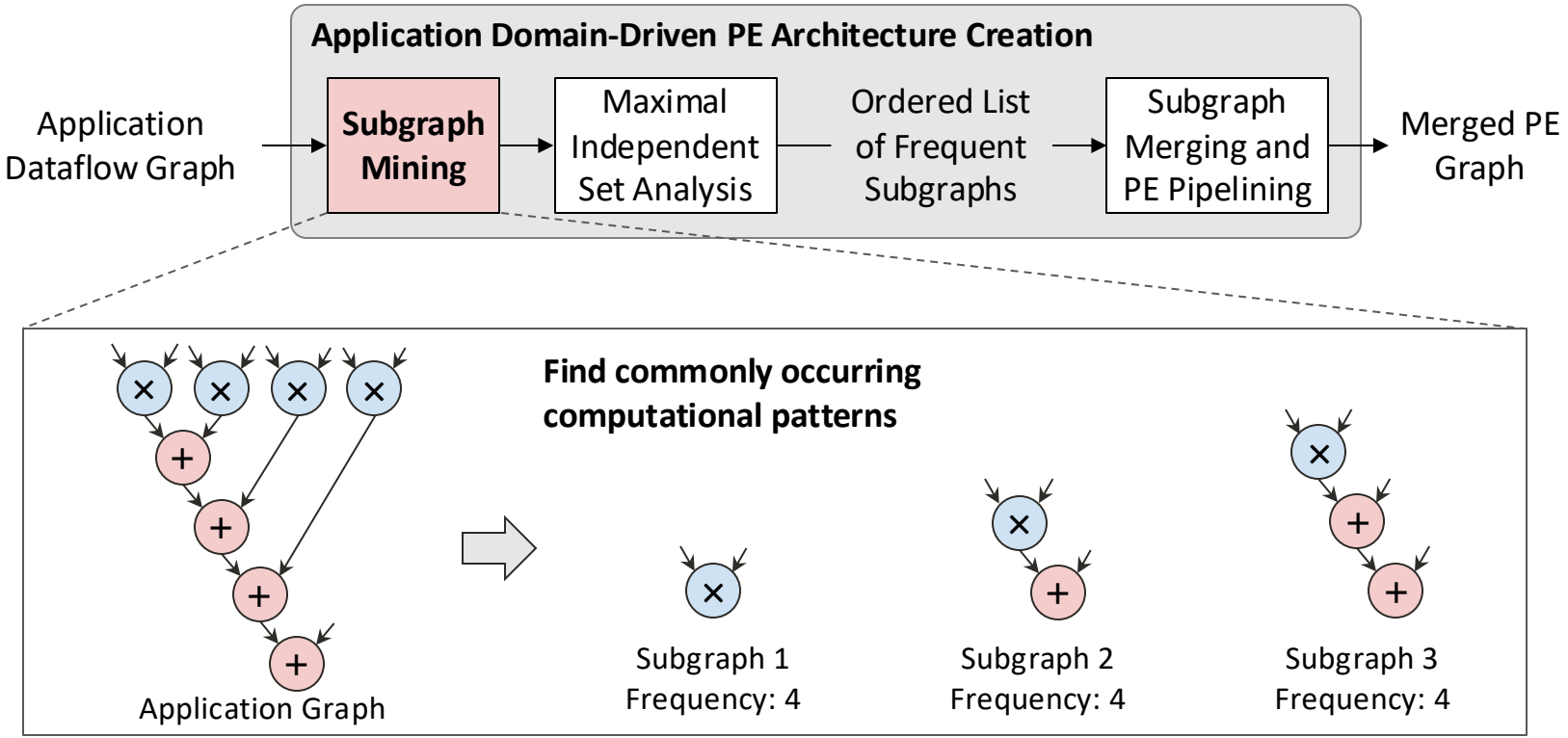


[1] Mining Graph Patterns, Hong Cheng, Xifeng Yan, and Jiawei Han. Springer, 2010.
[2] Efficient Datapath Merging for Partially Reconfigurable Architectures. Nahri Moreano, Edson Borin, Cid C. de Souza, and Guido Araujo IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2005.

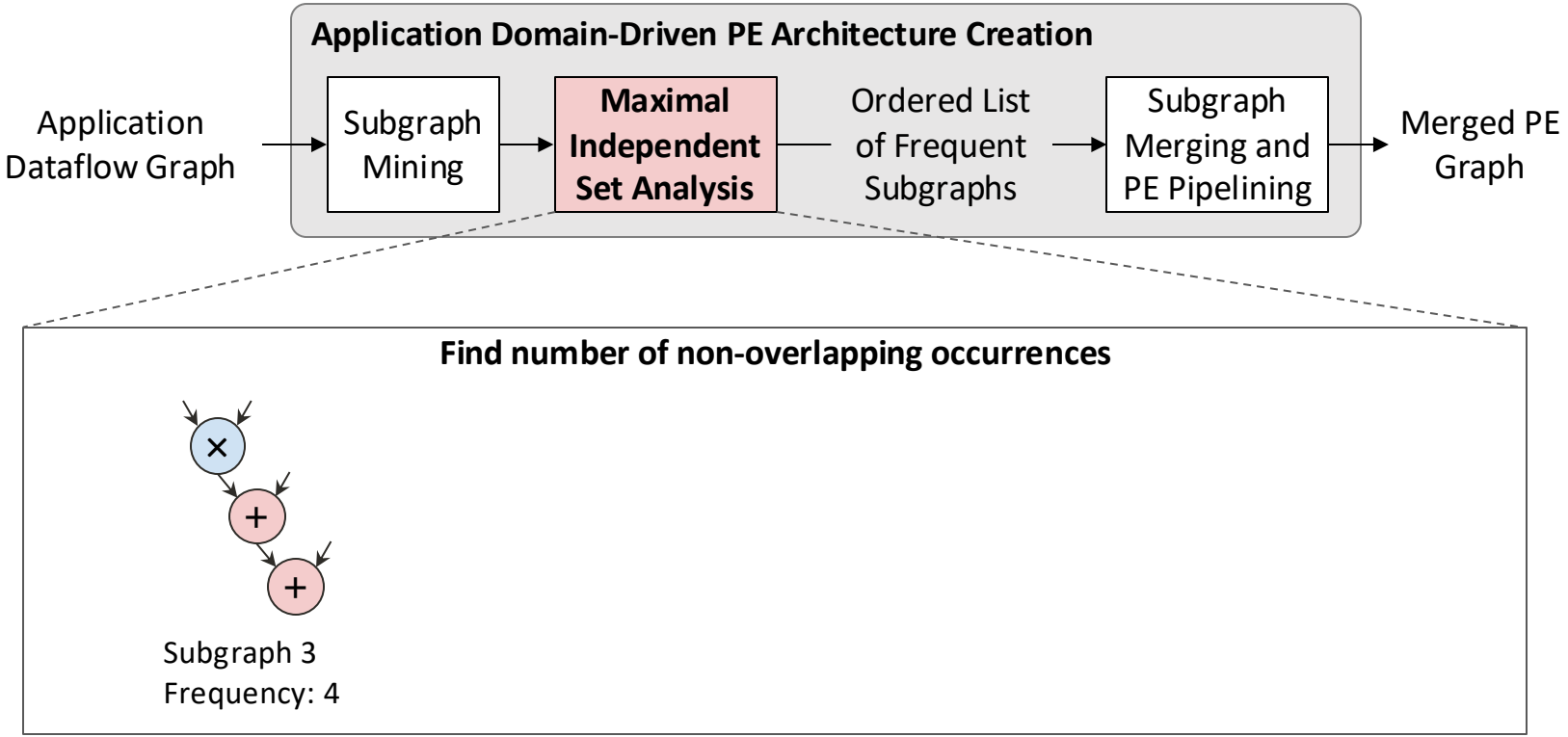
Subgraph Mining



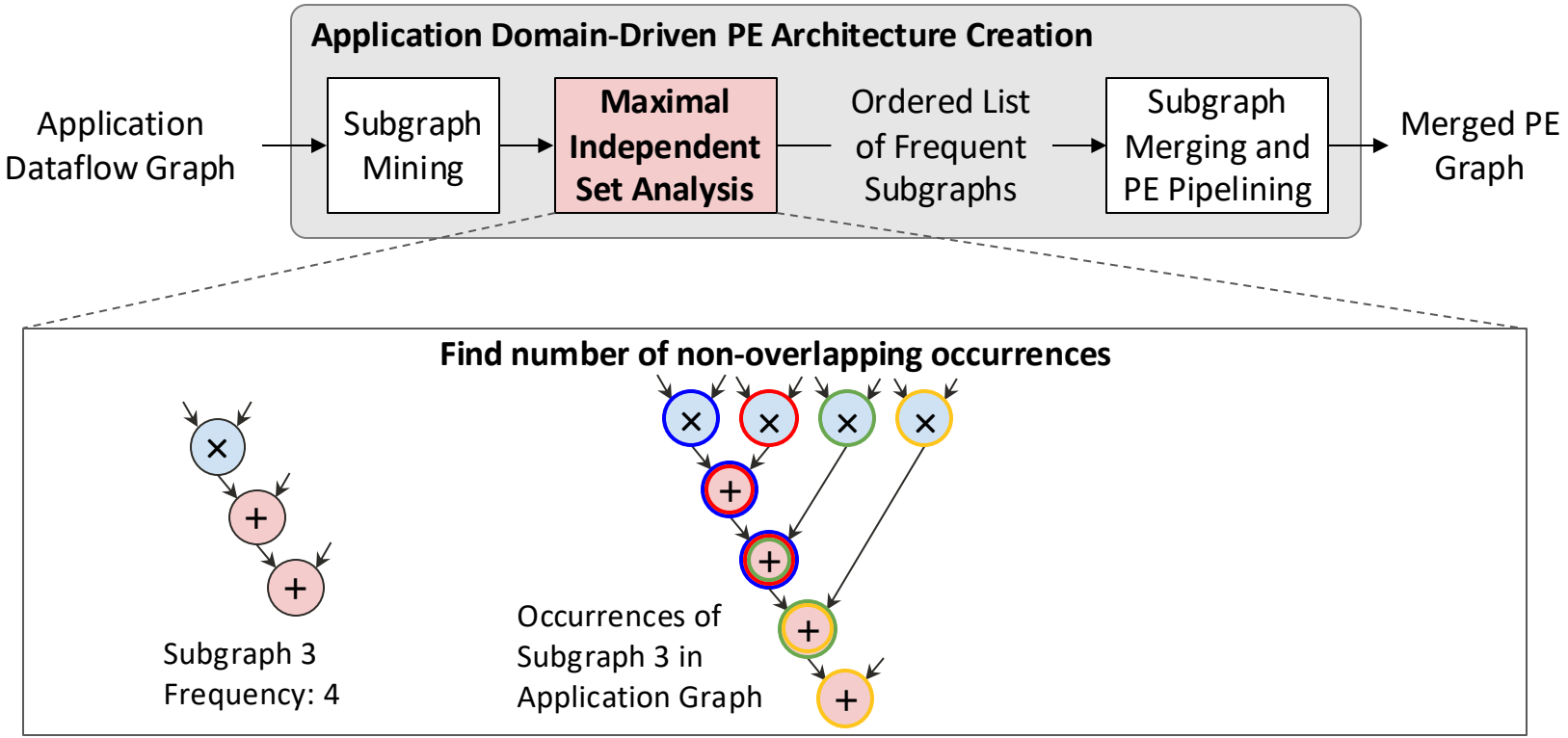
Subgraph Mining



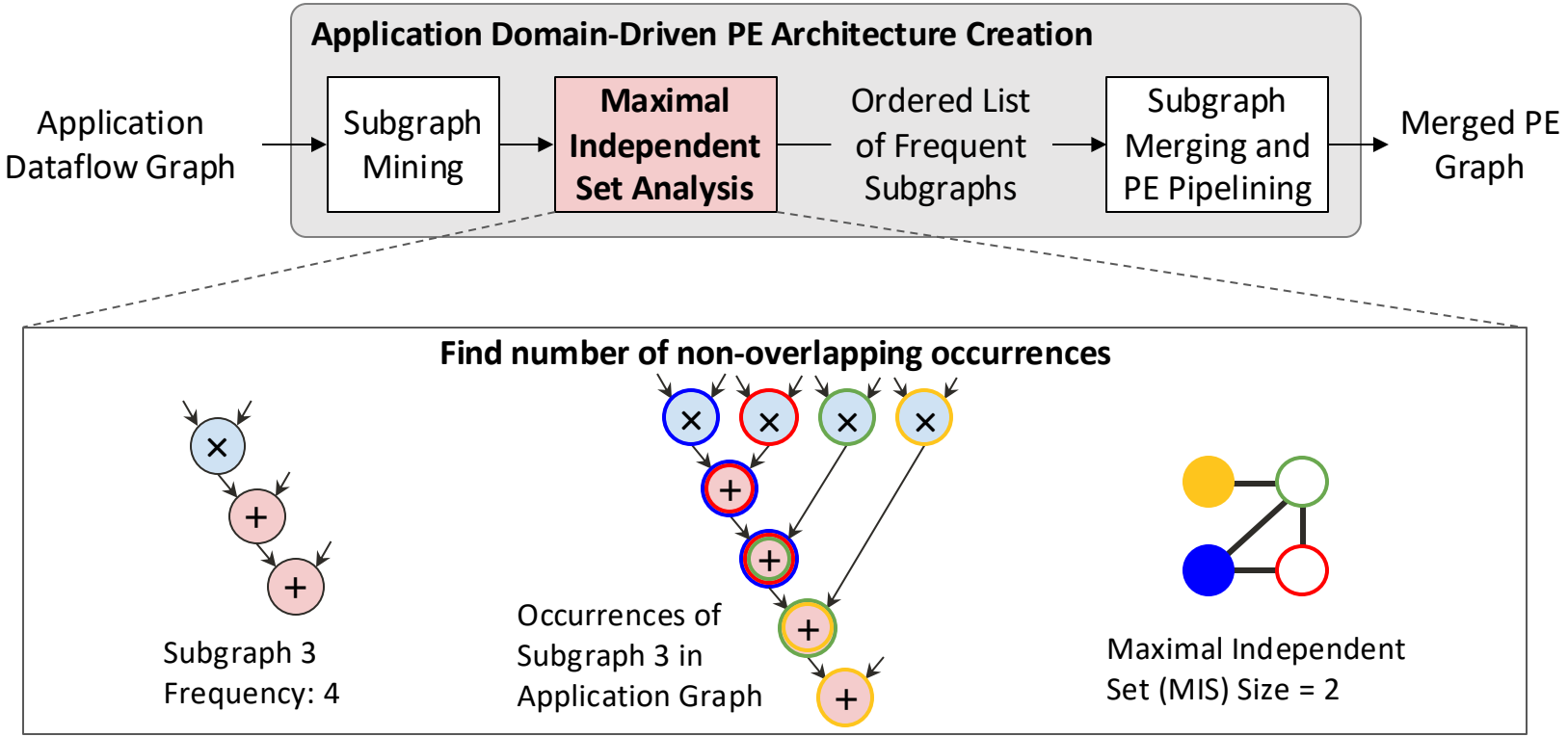
Finding Non-Overlapping Occurrences



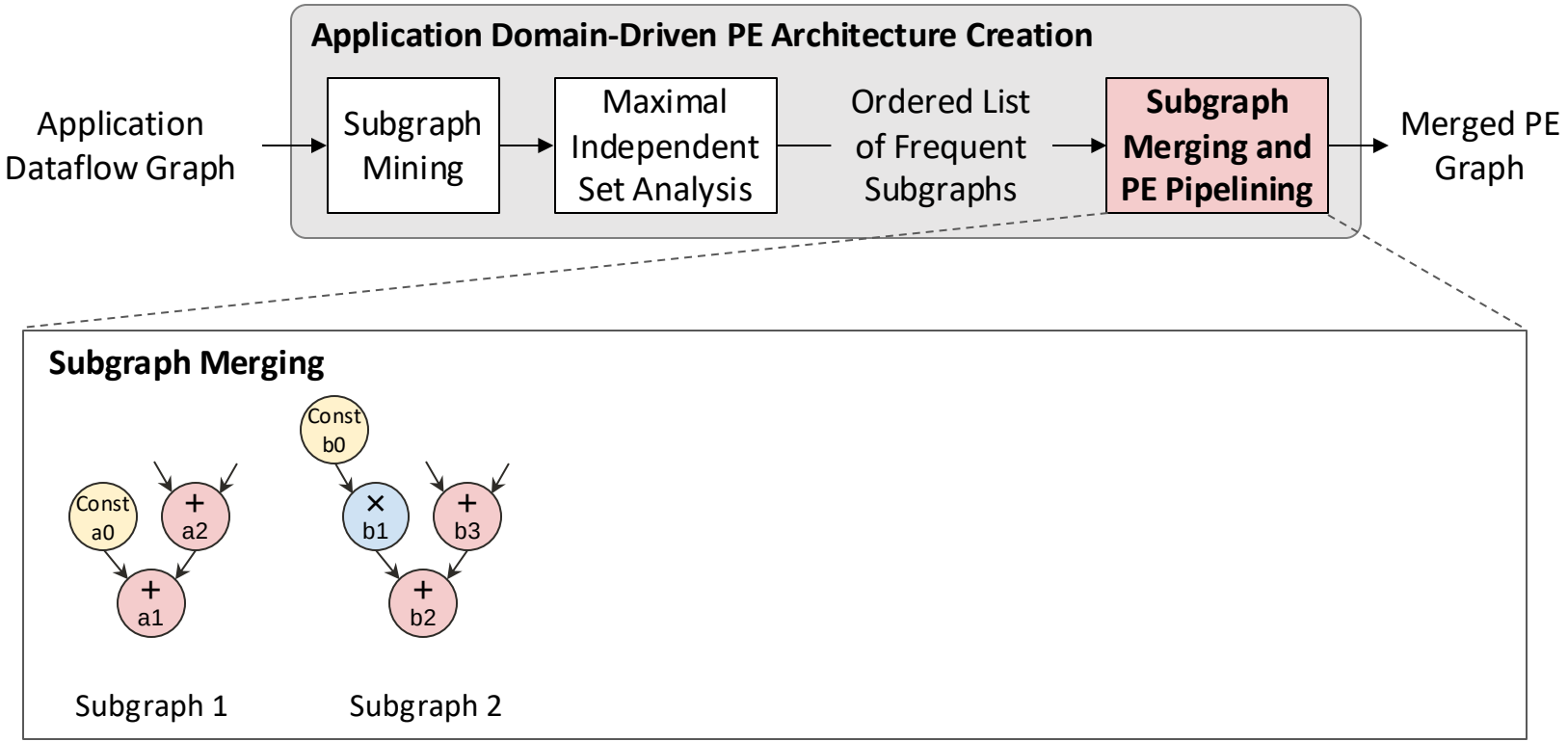
Finding Non-Overlapping Occurrences



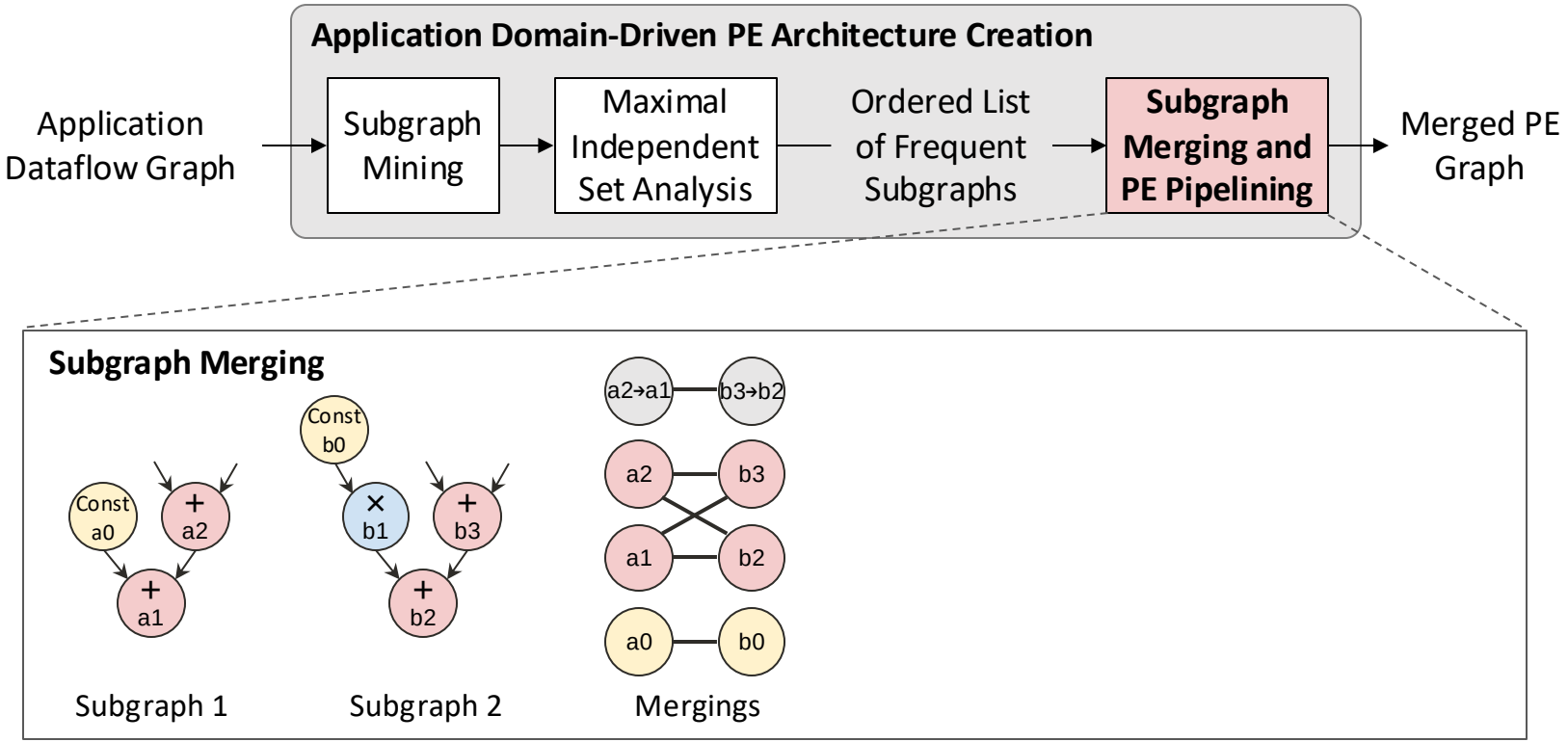
Finding Non-Overlapping Occurrences



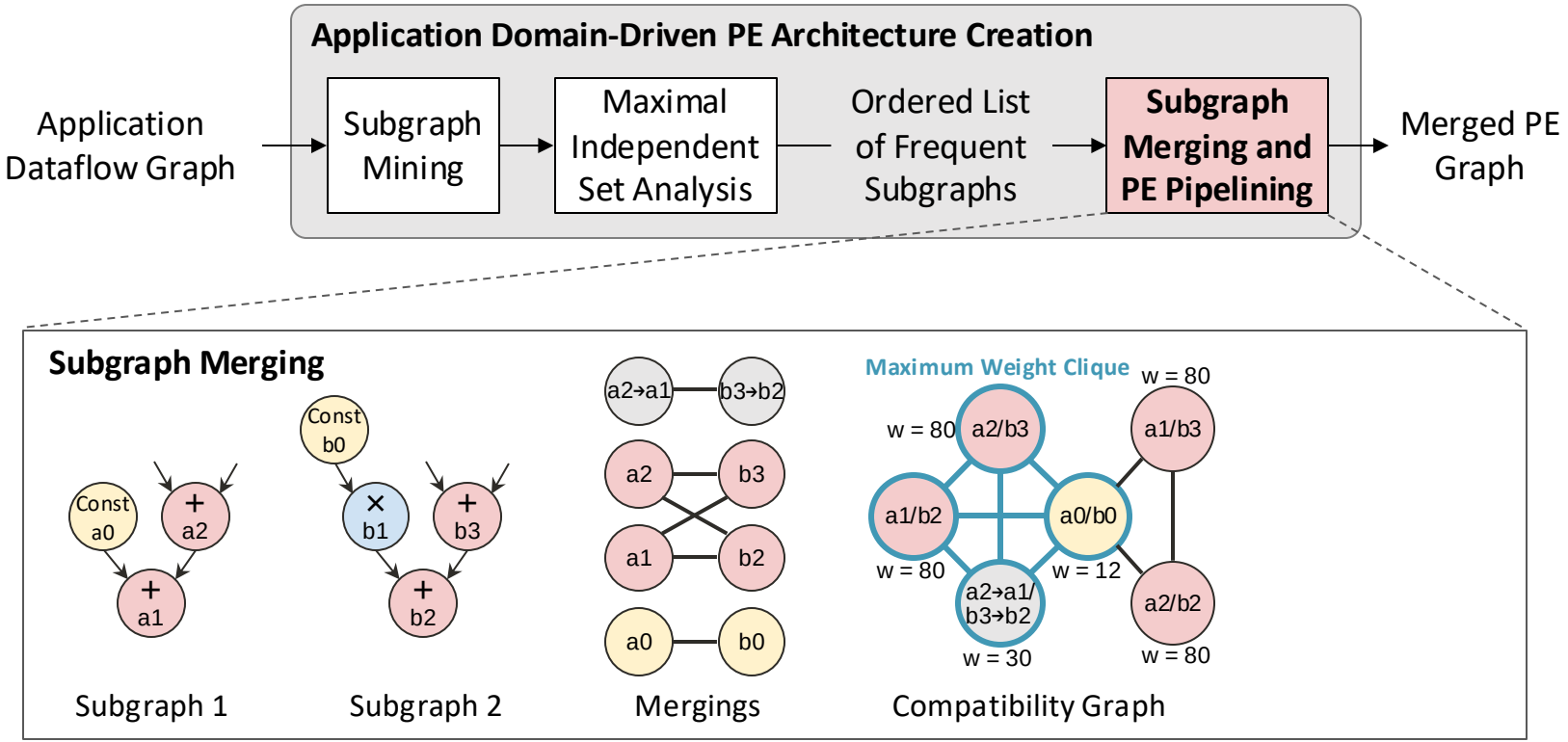
Subgraph Merging



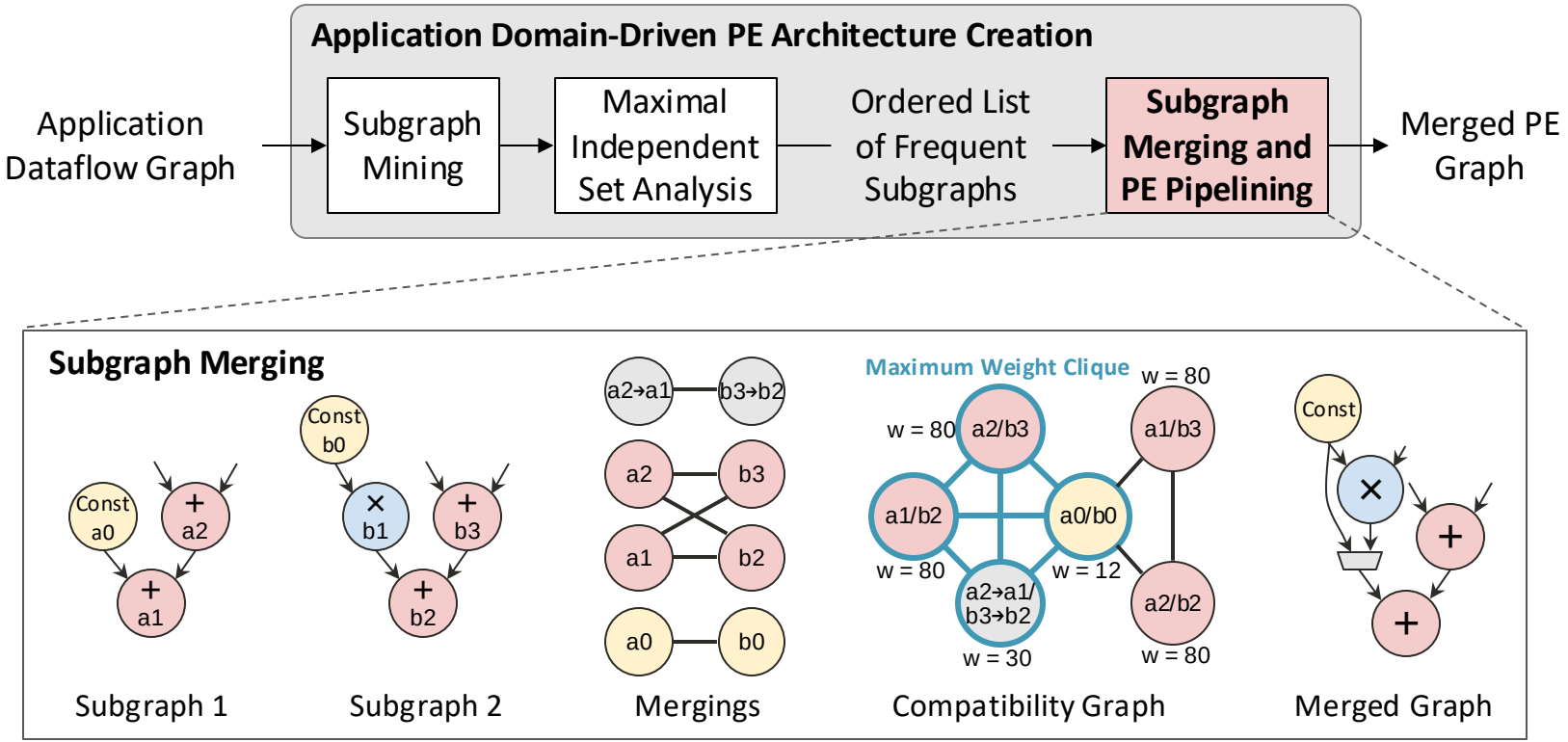
Subgraph Merging



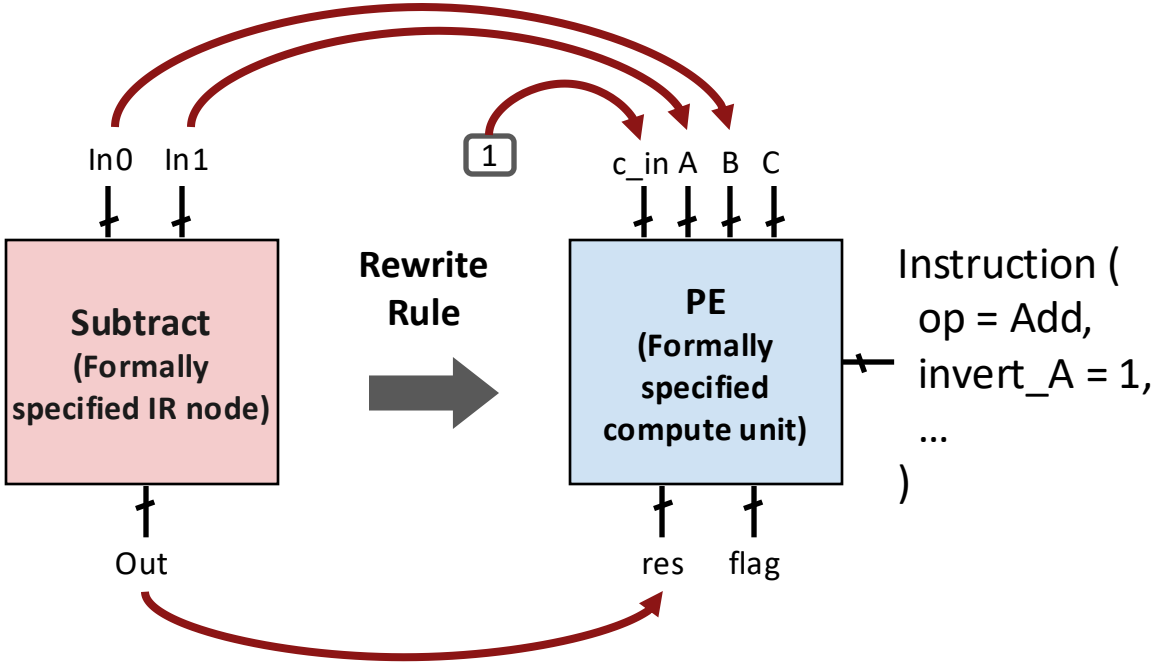
Subgraph Merging



Subgraph Merging



Automatic Rewrite Rule Synthesis Using SMT

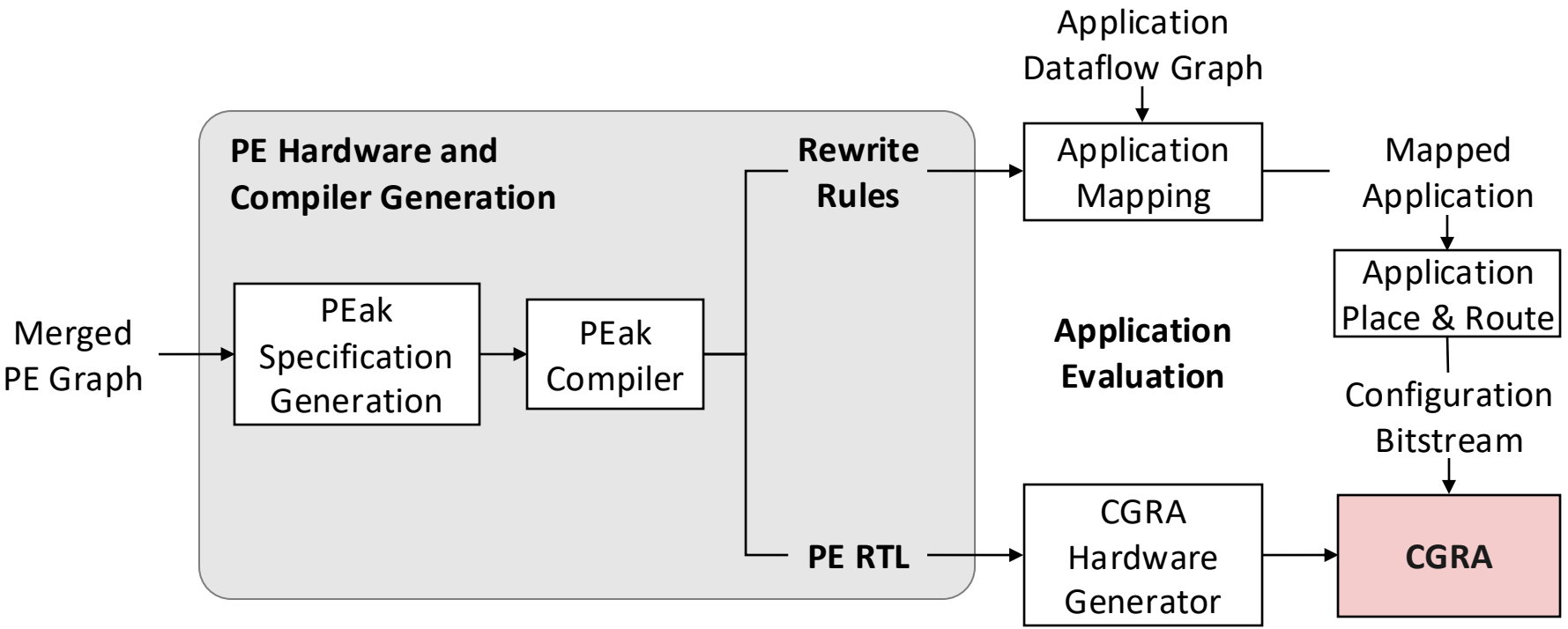


$\exists inst \; \forall inputs : Subtract(inputs) == PE(inst, inputs)$

Solve using an SMT solver

Synthesizing Instruction Selection Rewrite Rules from RTL Using SMT. Ross Daly, Caleb Donovick, Jackson Melchert, Rajsekhar Setaluri, Nestan Tsiskaridze Bullock, Priyanka Raina, Clark Barrett, and Pat Hanrahan. Formal Methods in Computer-Aided Design (FMCAD), 2022

APEX Design Space Exploration Framework



Demo

- In this demo, we will take a look at an example application or two and generate a PE specialized for those applications
- First, we will run two applications through the Halide to Hardware compiler:
- `aha map apps/harris`
- `aha map apps/gaussian`

Demo

- Next, we will load them into the APEX tool and do some analysis:
- `bash apex_demo.sh mine`
- We can see a visualization of the application compute graph and the mined subgraphs in /aha/APEX/pdf/

Demo

- Finally, we can generate a customized PE including all the subgraphs that you might want with the following command:
- `bash apex_demo.sh specialize`
- The visualization of the resulting PE is in /APEX/arch_graph.pdf
- The Verilog of the resulting PE is in /APEX/outputs/verilog/PE.v

Evaluation - Baseline PE

- One ALU
- One multiplier
- Two registers for integer operands
- Bit registers and LUT for bitwise operations

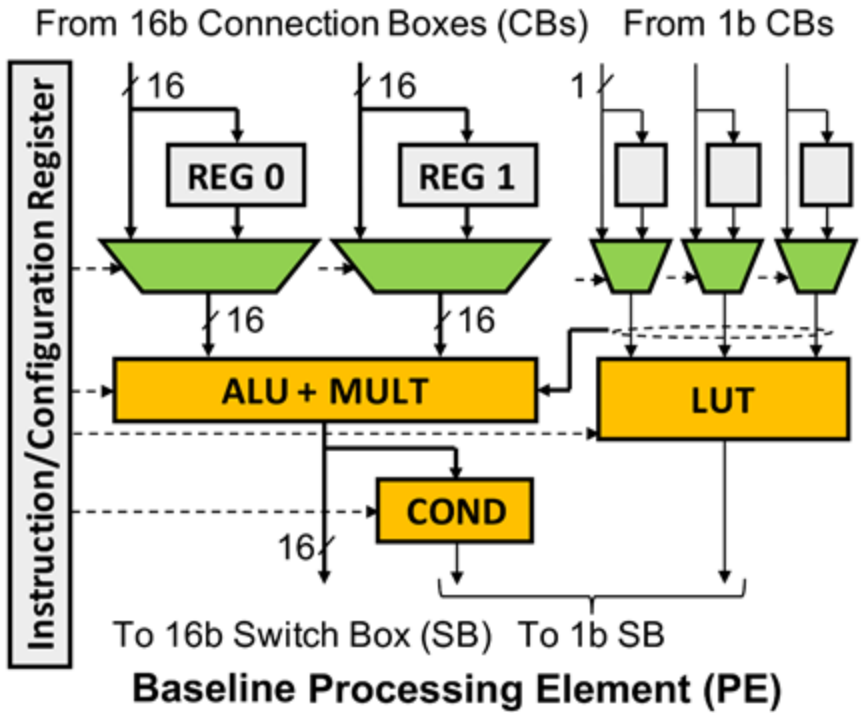
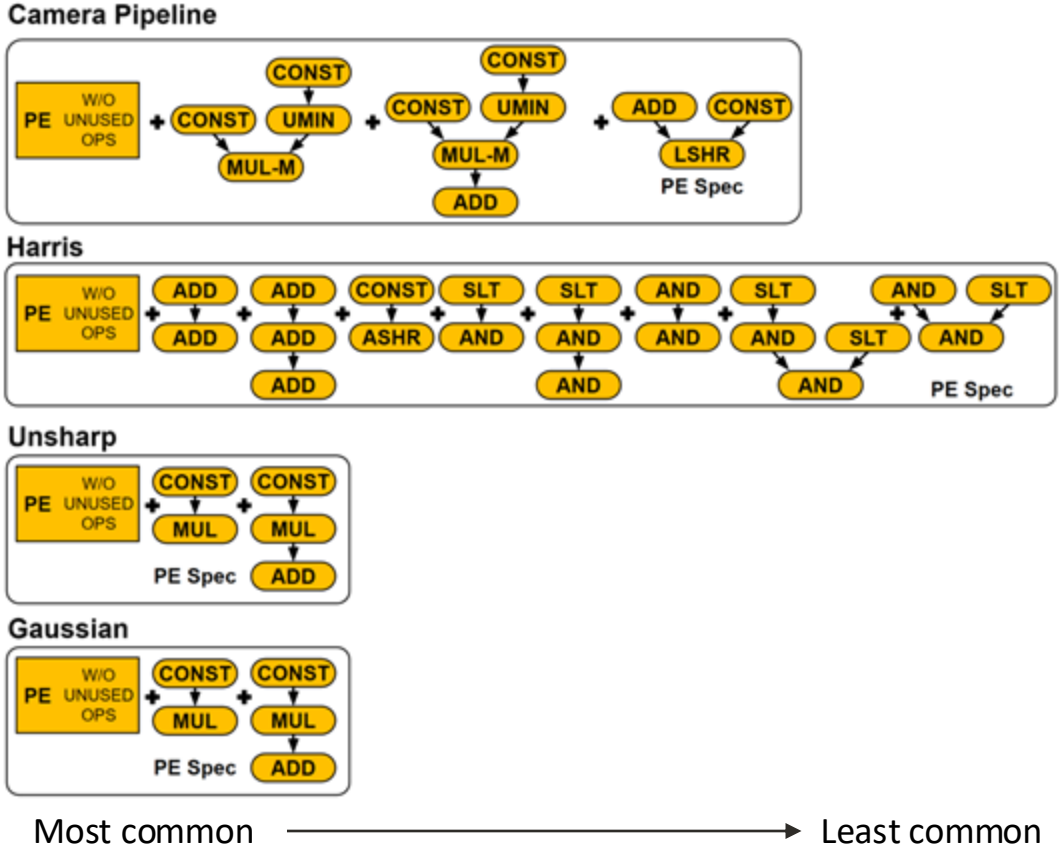


Image Processing Specialization

- APEX generates a specialized PE for each application
- Each PE Spec contains the most common operations from those applications



Architecture of Camera Pipeline PE Spec

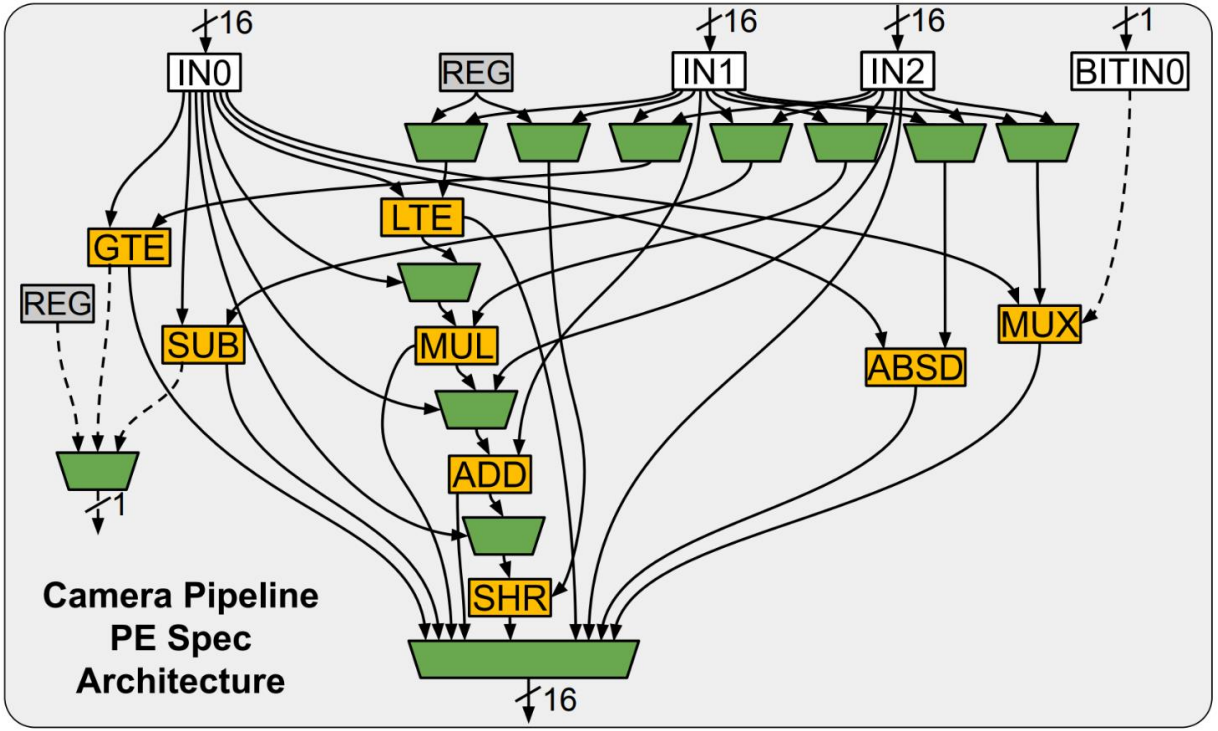
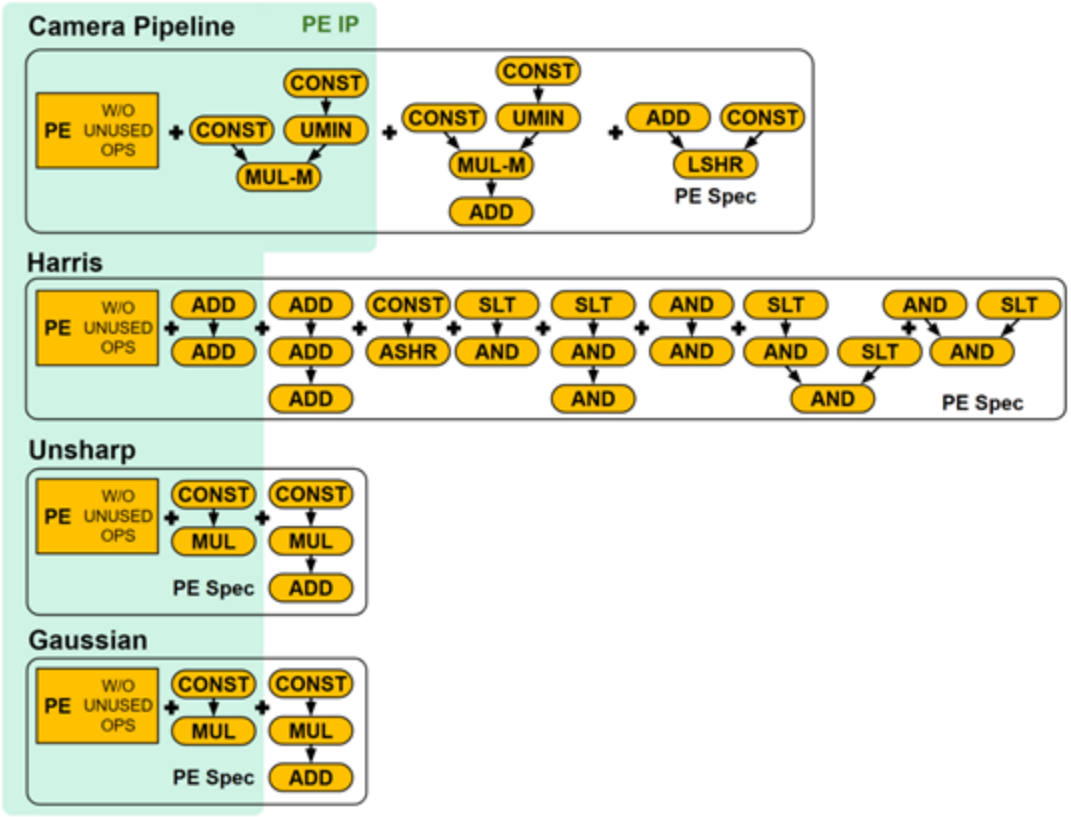


Image Processing Specialization

- APEX also generates PE IP, specialized across all image processing applications



Architecture of PE IP

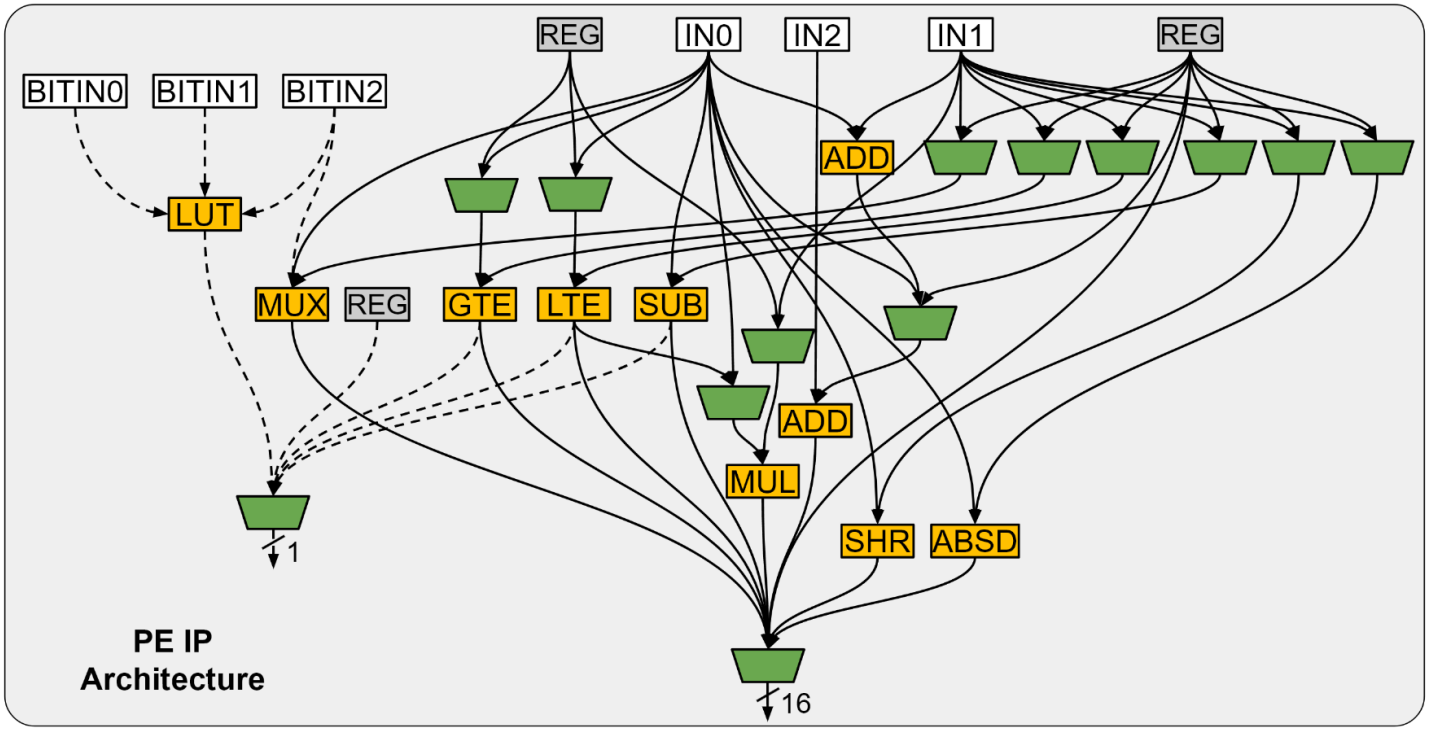
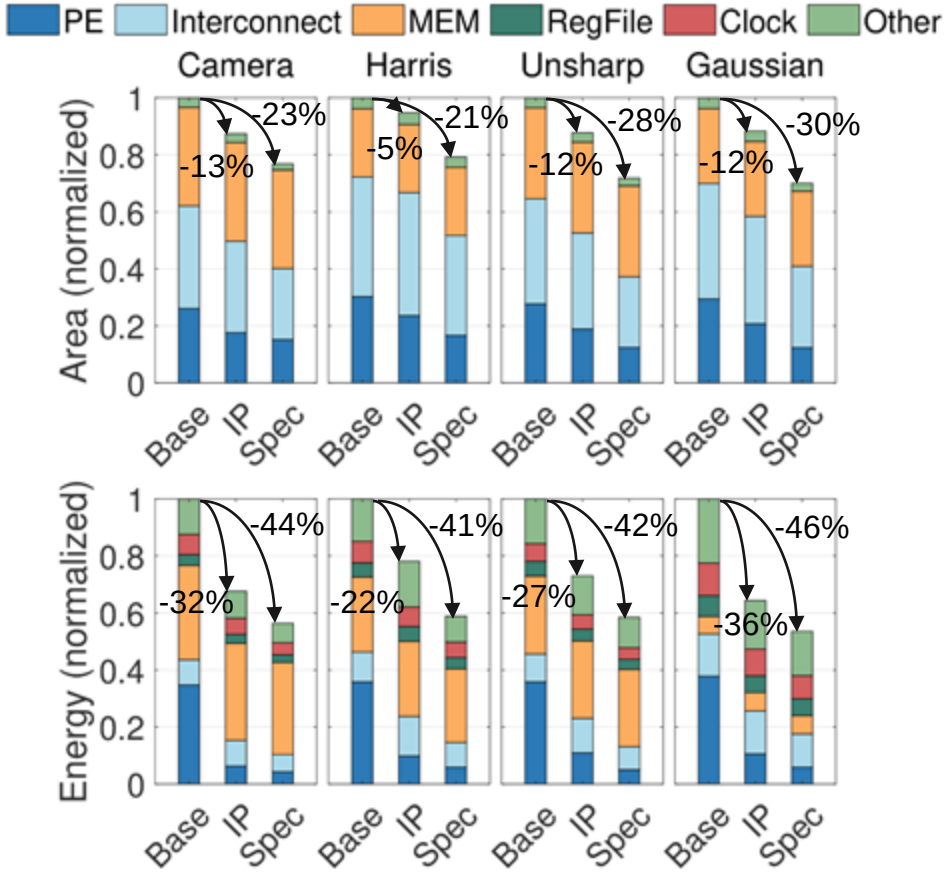


Image Processing Specialization

- CGRA with PE IP provides significant area and energy improvements over the baseline PE
- CGRA with PE Spec provides significant improvement over PE IP



APEX Summary

- Developed APEX: a framework for automated design space exploration of CGRA PEs
  - Allows for application domain-driven specialization of CGRAs using subgraph mining and merging
  - Includes automated hardware and compiler generation
  - Generates specialized CGRAs that are more area and energy efficient compared to a baseline CGRA