A High-Level Synthesis Flow for Custom Instruction Set Extensions for Application-Specific Processors

Nagaraju Pothineni
Google, India

Philip Brisk
UC Riverside

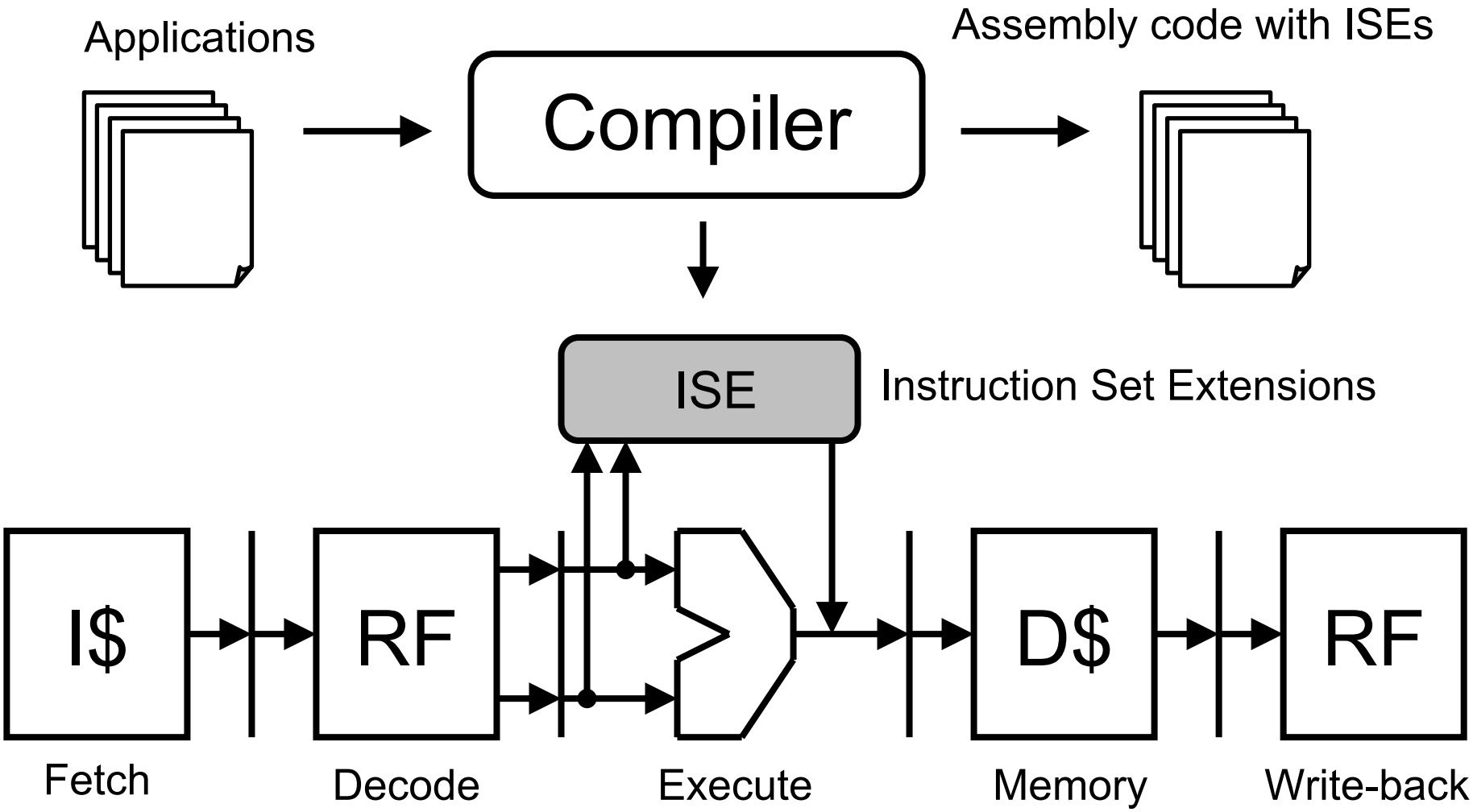
Paolo Ienne
EPFL

Anshul Kumar
IIT Delhi

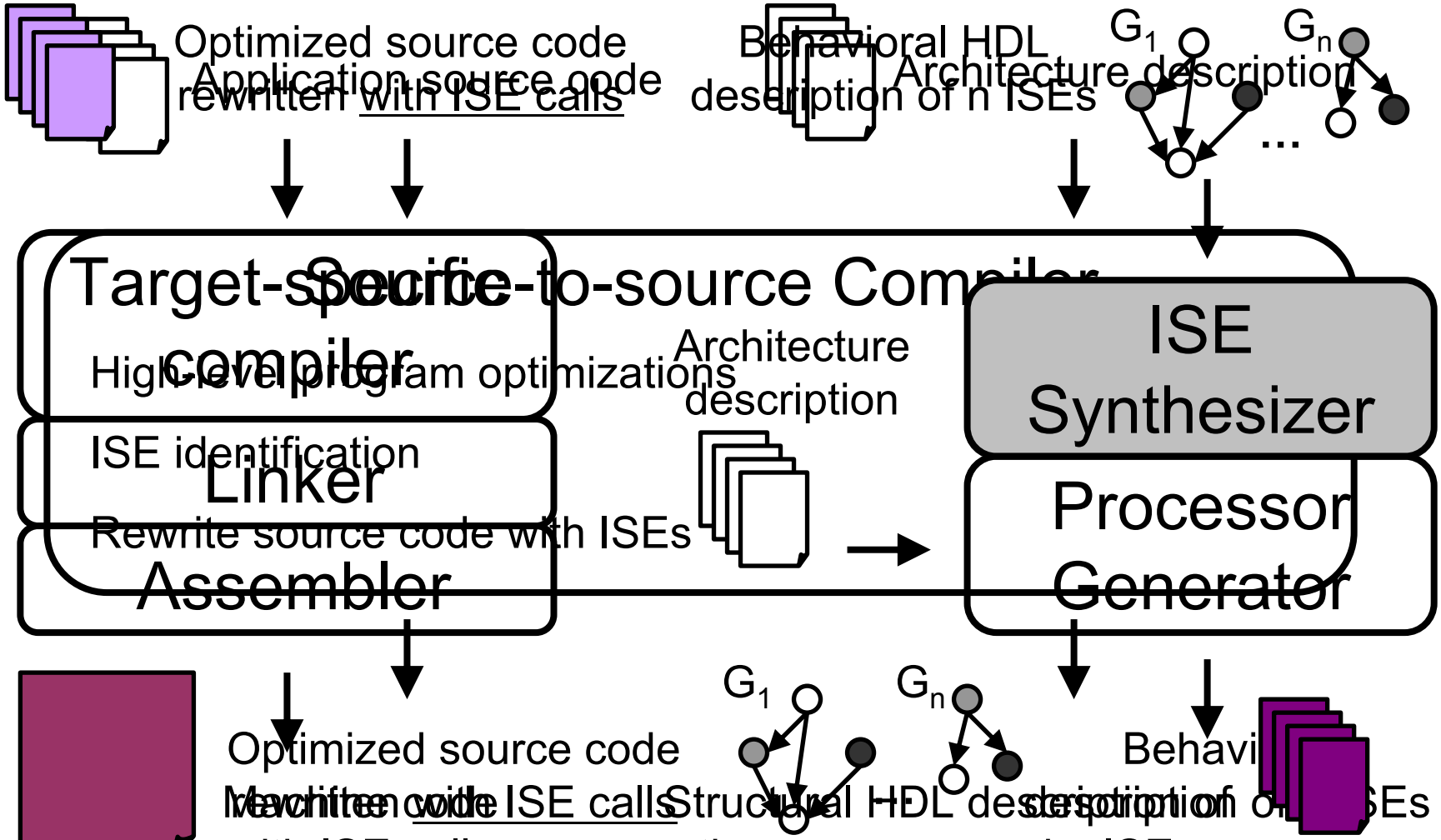
Kolin Paul
IIT Delhi

Asia and South Pacific Design Automation Conference
Taipei, Taiwan R.O.C. January 21, 2010

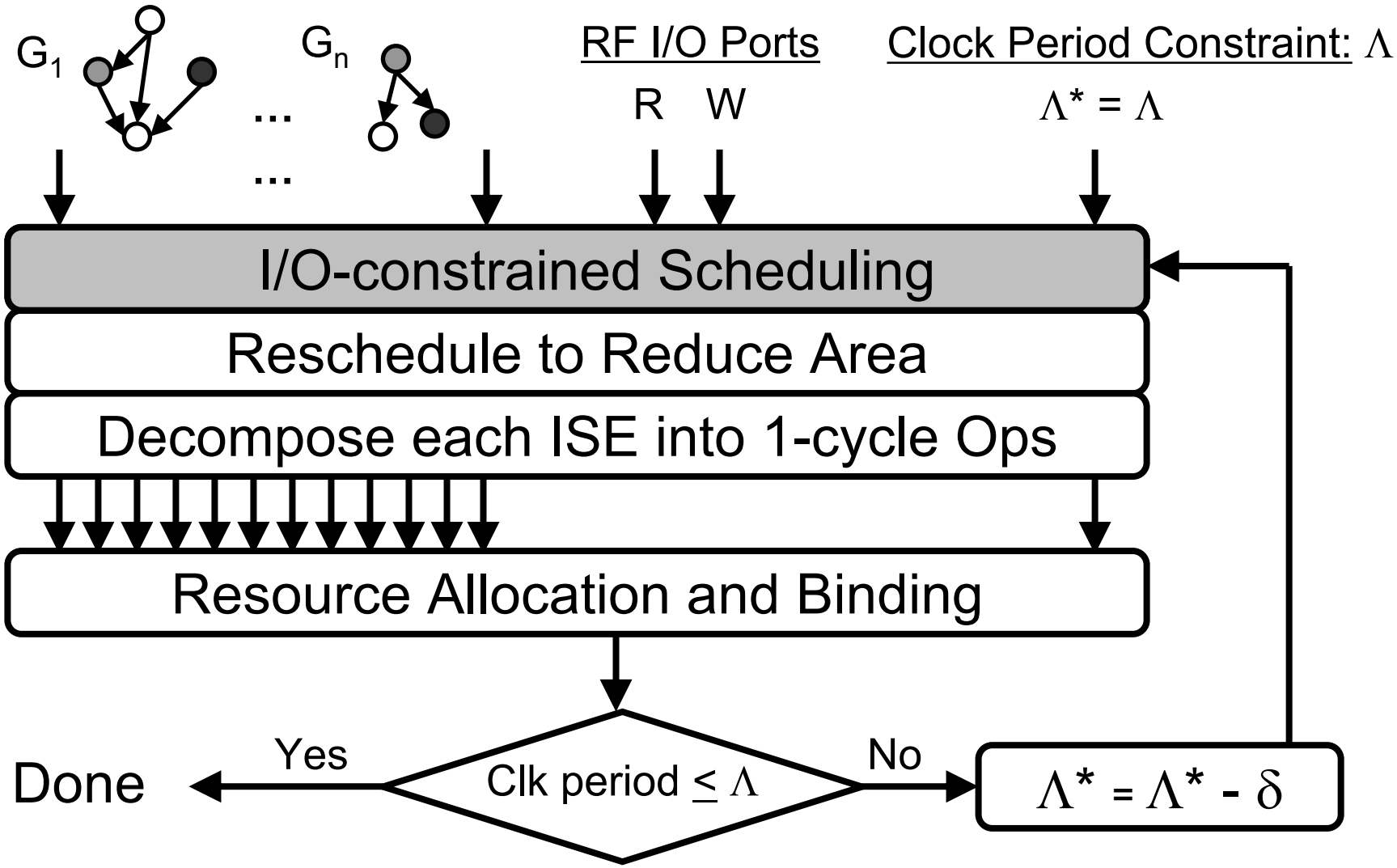
Extensible Processors



Compilation Flow

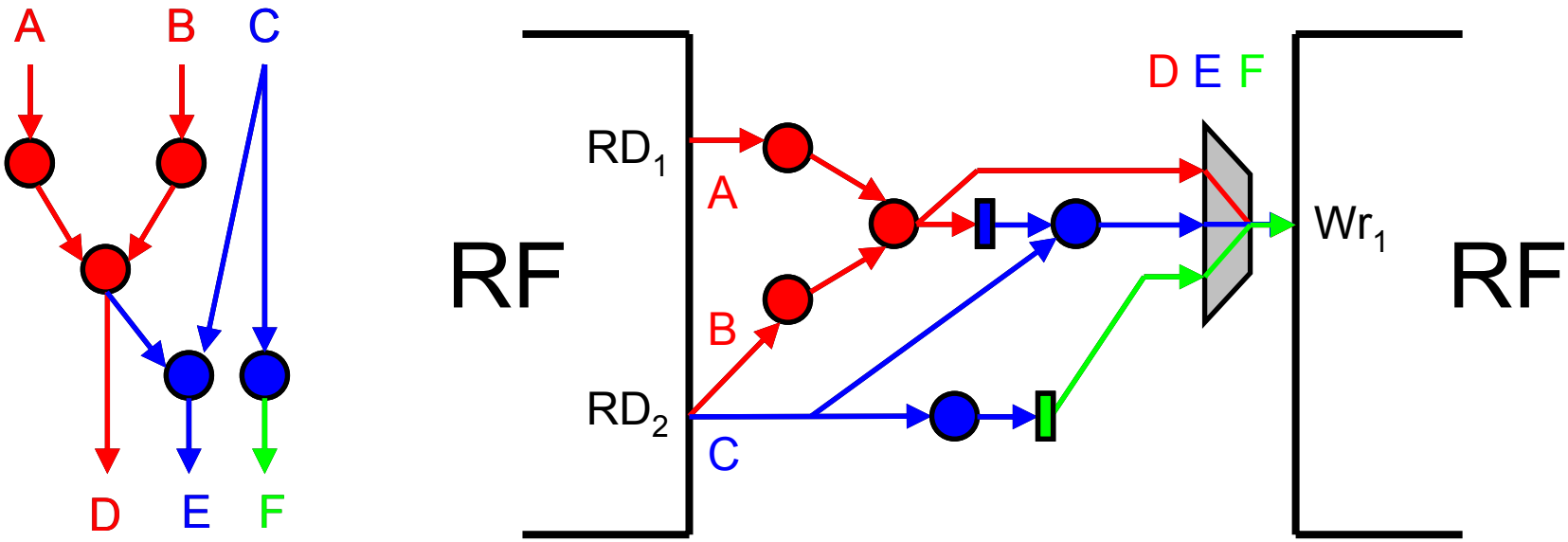


ISE Synthesis Flow

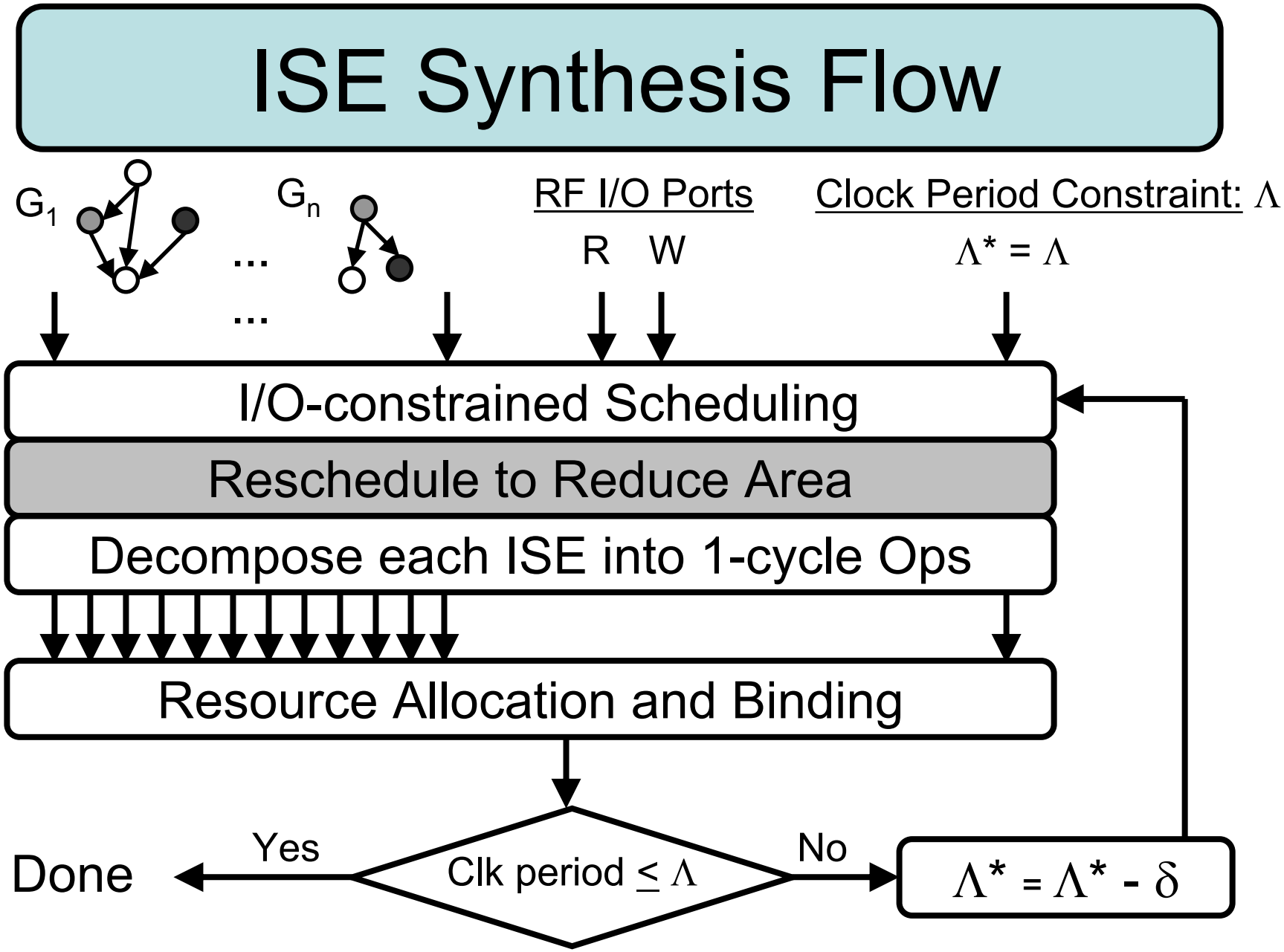


I/O-constrained Scheduling

- I/O ports are a resource constraint
  - Resource-constrained scheduling is NP-complete
  - Optimal algorithm [Pozzi and Ienne, CASES '05]

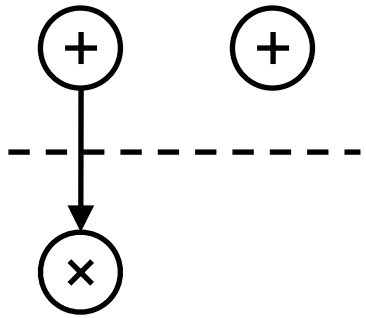


ISE Synthesis Flow

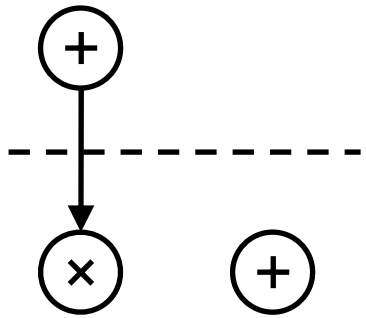


Reschedule to Reduce Area

- Goal:
  - Minimize area
- Constraints:
  - Do not increase latency or clock period
  - I/O constraints
- Implementation:
  - Simulated annealing (details in the paper)

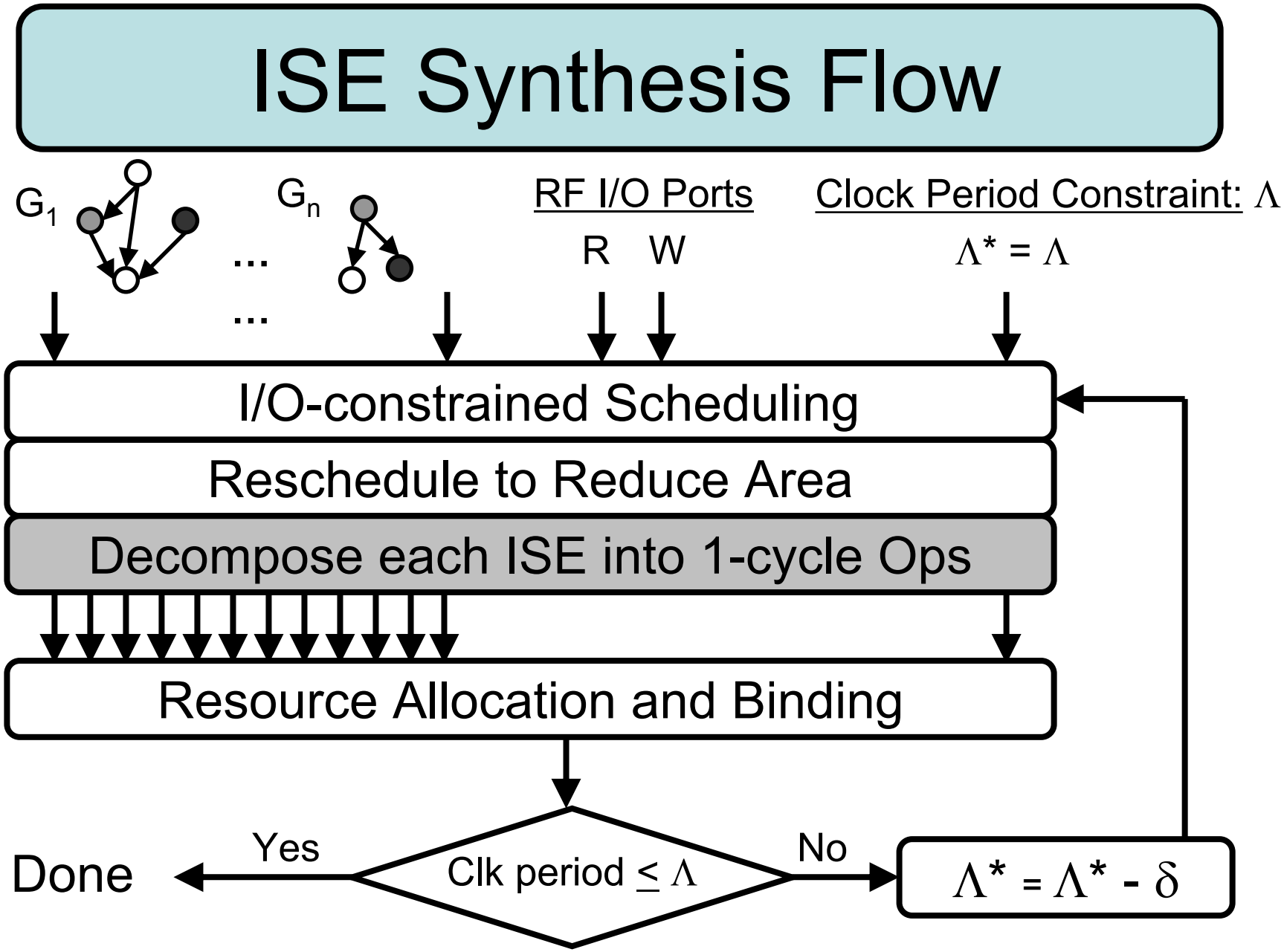


2 Adders, 1 Multiplier



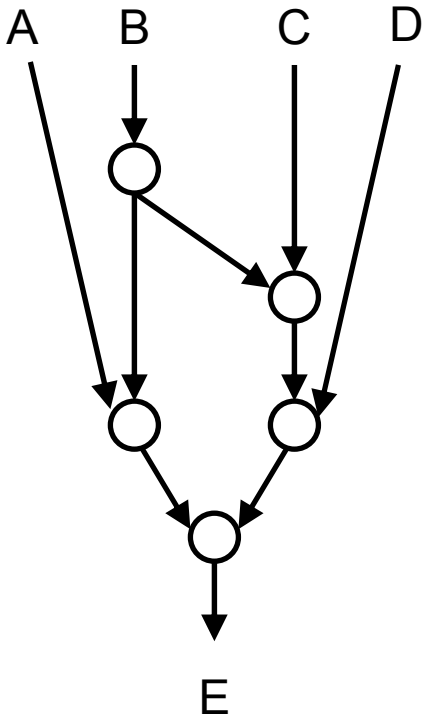
1 Adder, 1 Multiplier

ISE Synthesis Flow

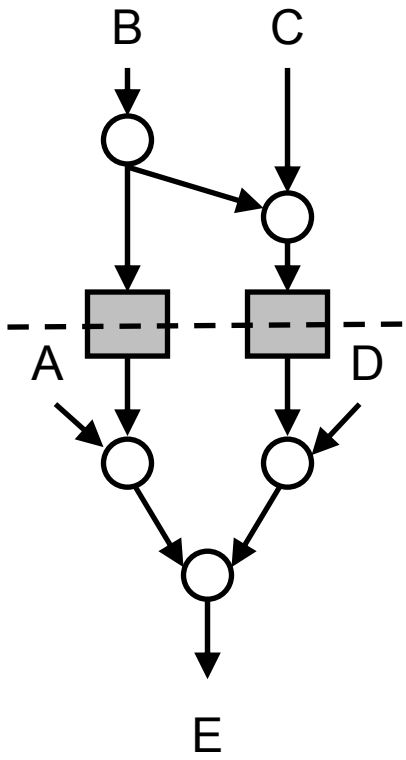


Decompose each ISE into Single-Cycle Operations

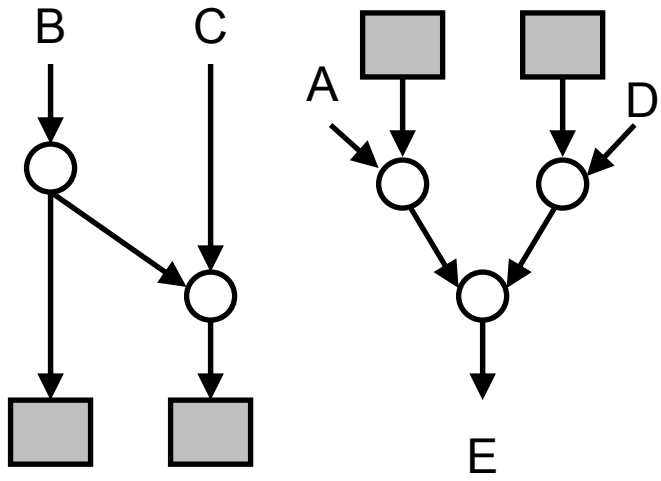
ISE



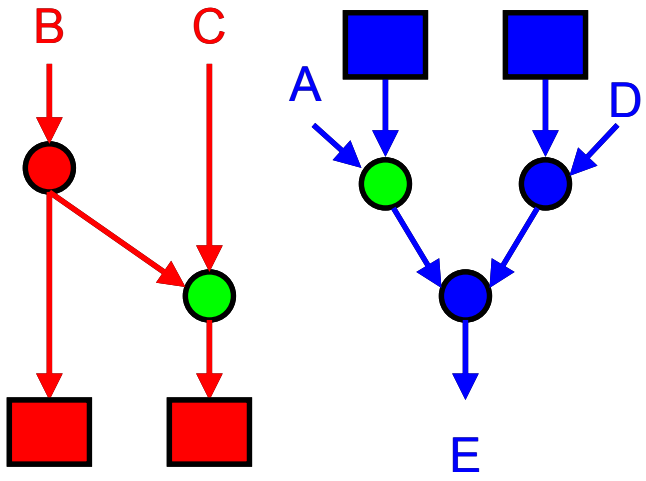
After Scheduling

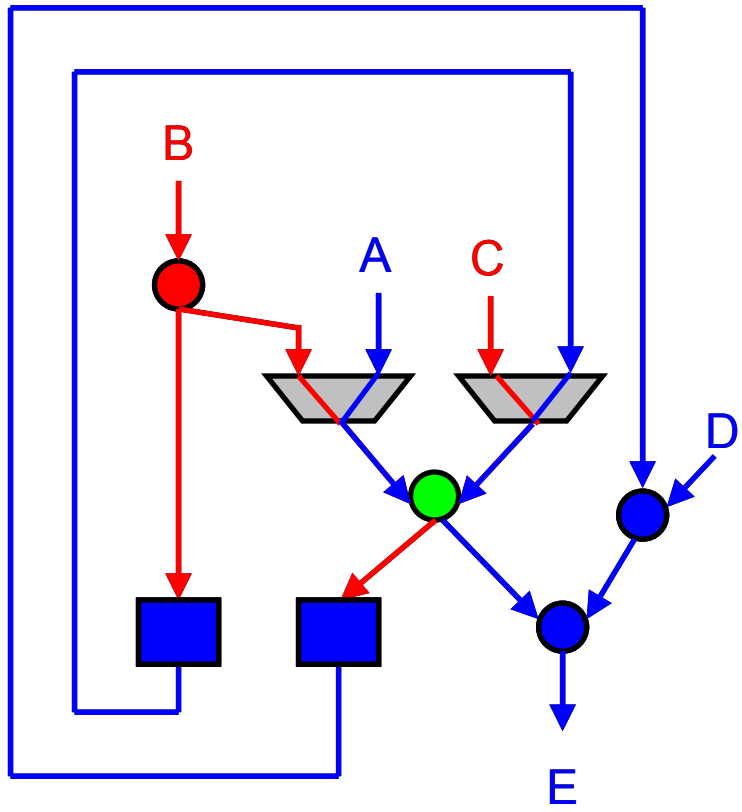


1-cycle Ops

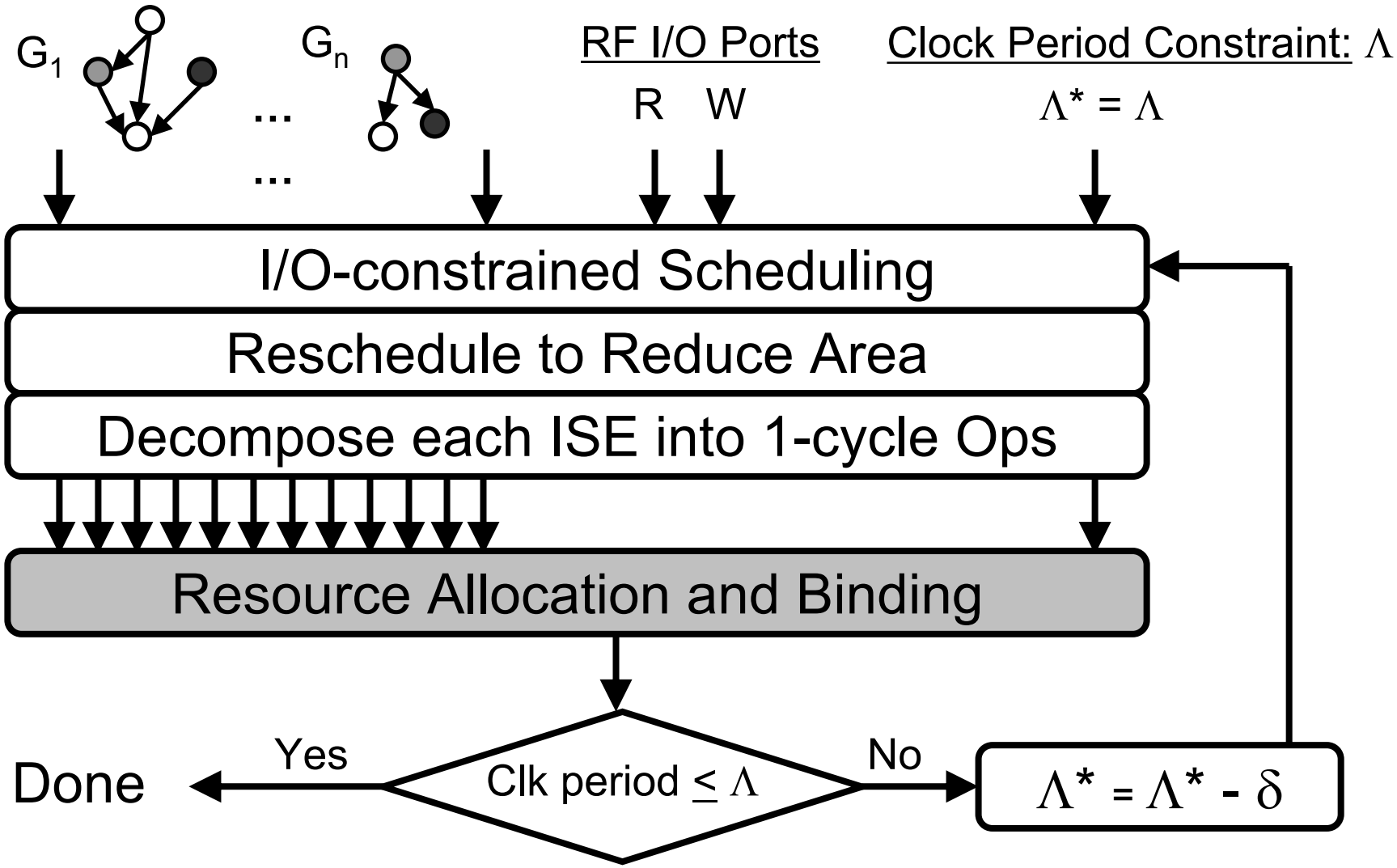


Decomposition Facilitates Resource Sharing within an ISE

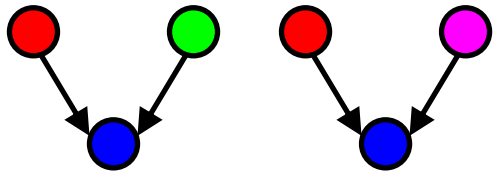




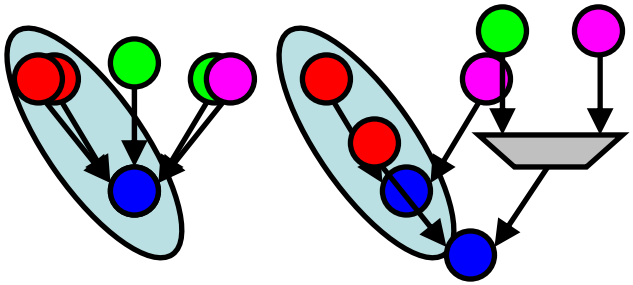
ISE Synthesis Flow



Resource Allocation and Binding



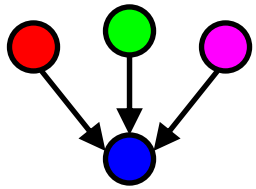
Two 1-cycle ISEs



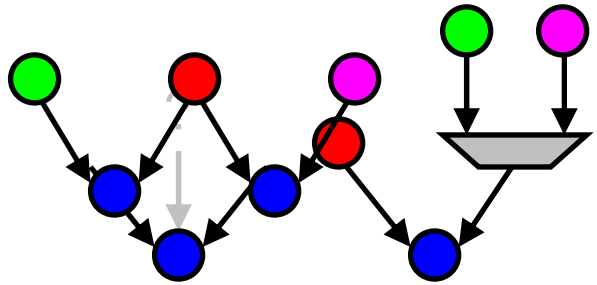
Minimum cost common supergraph

Maximum-cost weighted common isomorphic subgraph problem

Requires a multiplexer



Weighted minimum-cost common supergraph (WMCS) problem



Higher cost common supergraph 2-input operation

No multiplexers needed (Multiplexers required)

NP-c

(Grap LSI)

Which solution is better?

Depends on the cost of the multiplexers compared to the merged operations!

Datapath Merging (DPM)

- Old problem formulation
  - Based on WMCS problem (graph theory)
    - NP-complete [Bunke et al., ICIP '02]
  - Share as many operations/interconnects as possible
    - [Moreano et al., TCAD '05; de Souza e al., JEA '05]
  - Optimize port assignment as a post-processing step
    - NP-complete [Pangrle, TCAD '91]
    - Heuristic [Chen and Cong, ASPDAC '04]

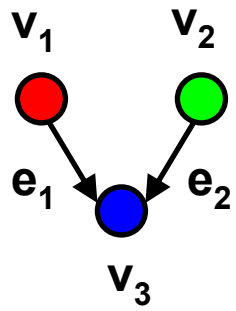
- Contribution: New problem formulation
  - Accounts for multiplexer cost and port assignment

New DPM Algorithms

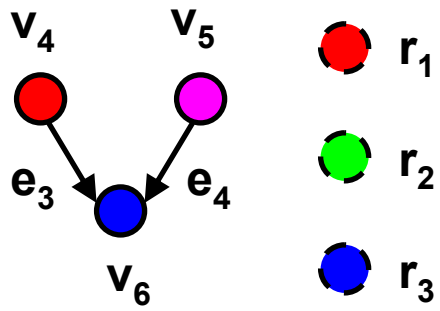
- ILP Formulation
  - See the paper for details
- Reduction to Max-Clique Problem
  - Extends [Moreano et al., TCAD ’05]
    - Solve Max-Clique problem optimally using “Cliquer”
  - Identify isomorphic subgraphs up-front
    - Merge isomorphic subgraphs rather than vertices/edges
    - (Details in the paper)

Example

New ISE fragment to merge



Partial merged datapath



Edge mappings:

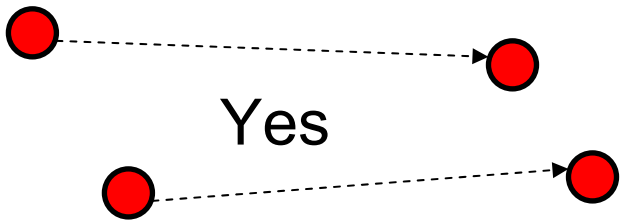
1. Map e_1 onto e_3
2. Must be compatible with vertex mappings: (v_1, v_4), (v_2, v_5), (v_3, v_6)

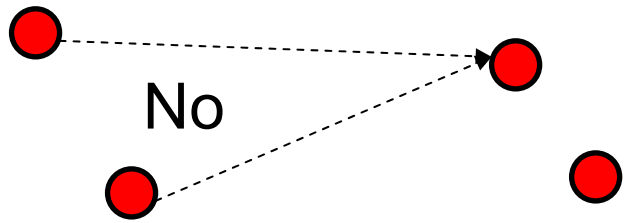
Vertex mappings:

1. Map v_1 onto v_4: (v_4, r_3), (r_1, v_5), (r_1, r_3)
2. Allocate a new resource r_4 and map v_5 onto r_4: (v_4, r_3), (r_4, v_5), (r_4, r_3)
3. Must be compatible with vertex mappings: r_1

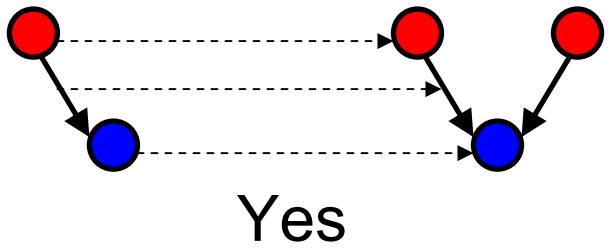
Compatibility

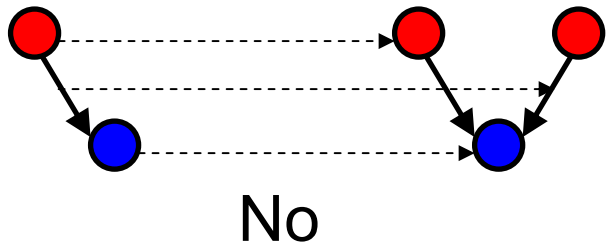
– Vertex/vertex compatibility



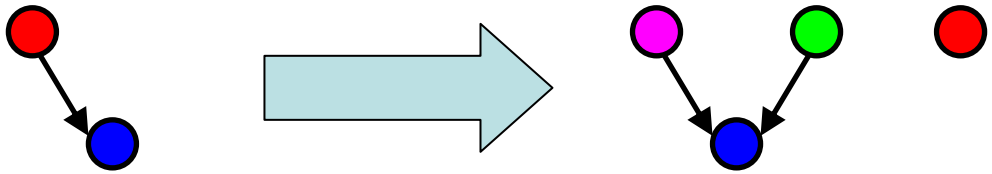


– Vertex/edge compatibility

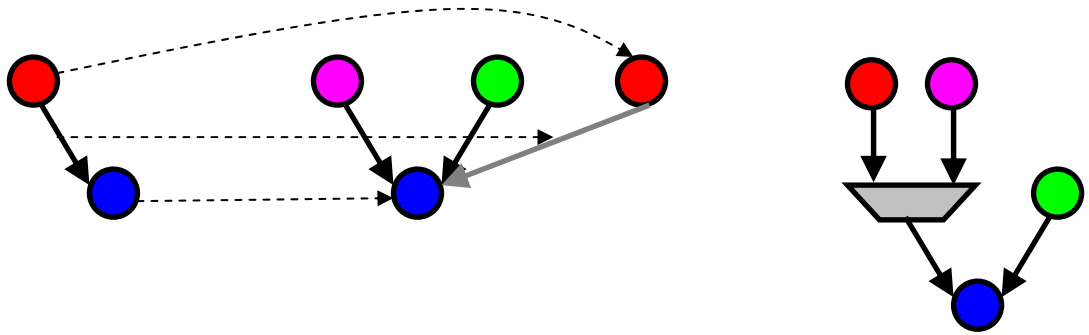




Why Edge Mappings?

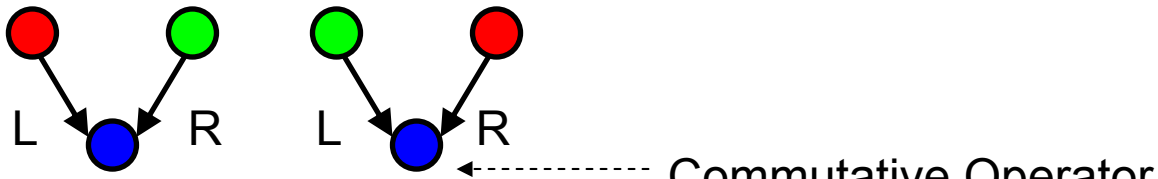


- Allocate an edge in the merged datapath
- May require a multiplexer

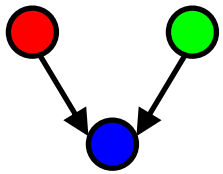


Port Assignment

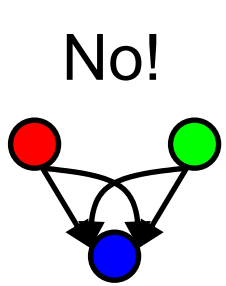
- Deterministic for non-commutative operators
- NP-complete for <u>every</u> commutative operator
  - [Pangrle, TCAD '91]

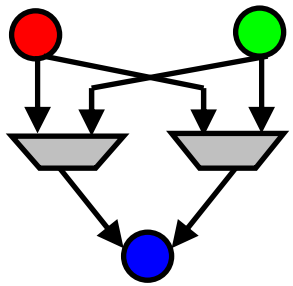


We want this!

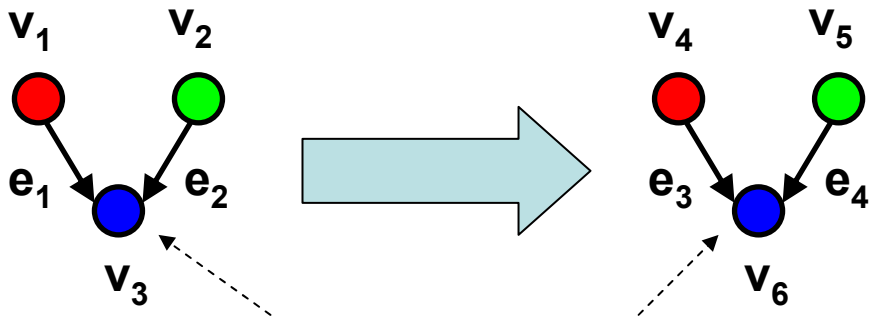


No!

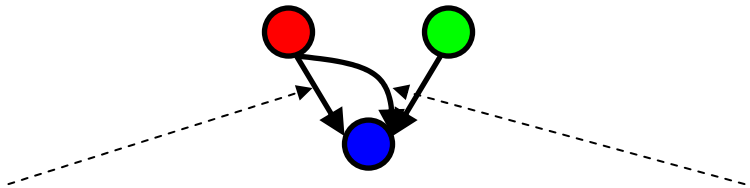




Edge Mappings = Port Assignment



Commutative Operator



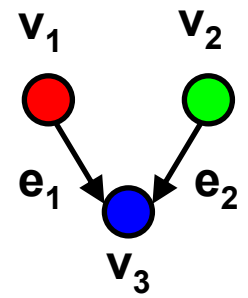
Mapping:
e_1: (v_4, v_6, L)

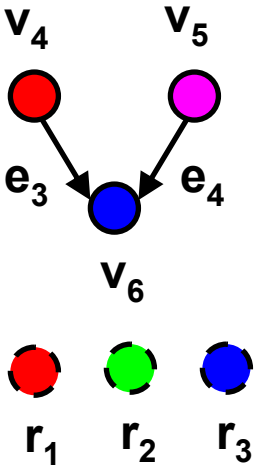
Mapping:
e_1: (v_4, v_6, R)

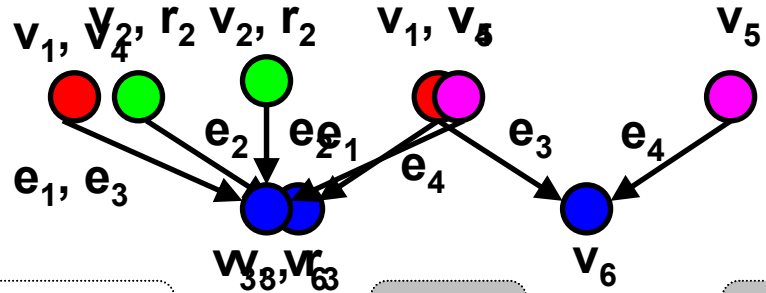
Compatibility Graph

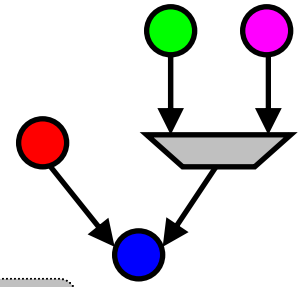
- Vertices correspond to mappings
  - Vertex mappings
    - Weight is 0 for vertex → vertex
    - Weight is resource cost for vertex → new resource
  - Edge mappings, including port assignment
    - Weight is 0 if edge exists in merged datapath
    - Weight is estimated cost of increasing mux size by +1 otherwise
- Place edges between compatible mappings
  - Each max-clique corresponds to a complete binding solution
  - Goal is to find max-clique of minimum weight

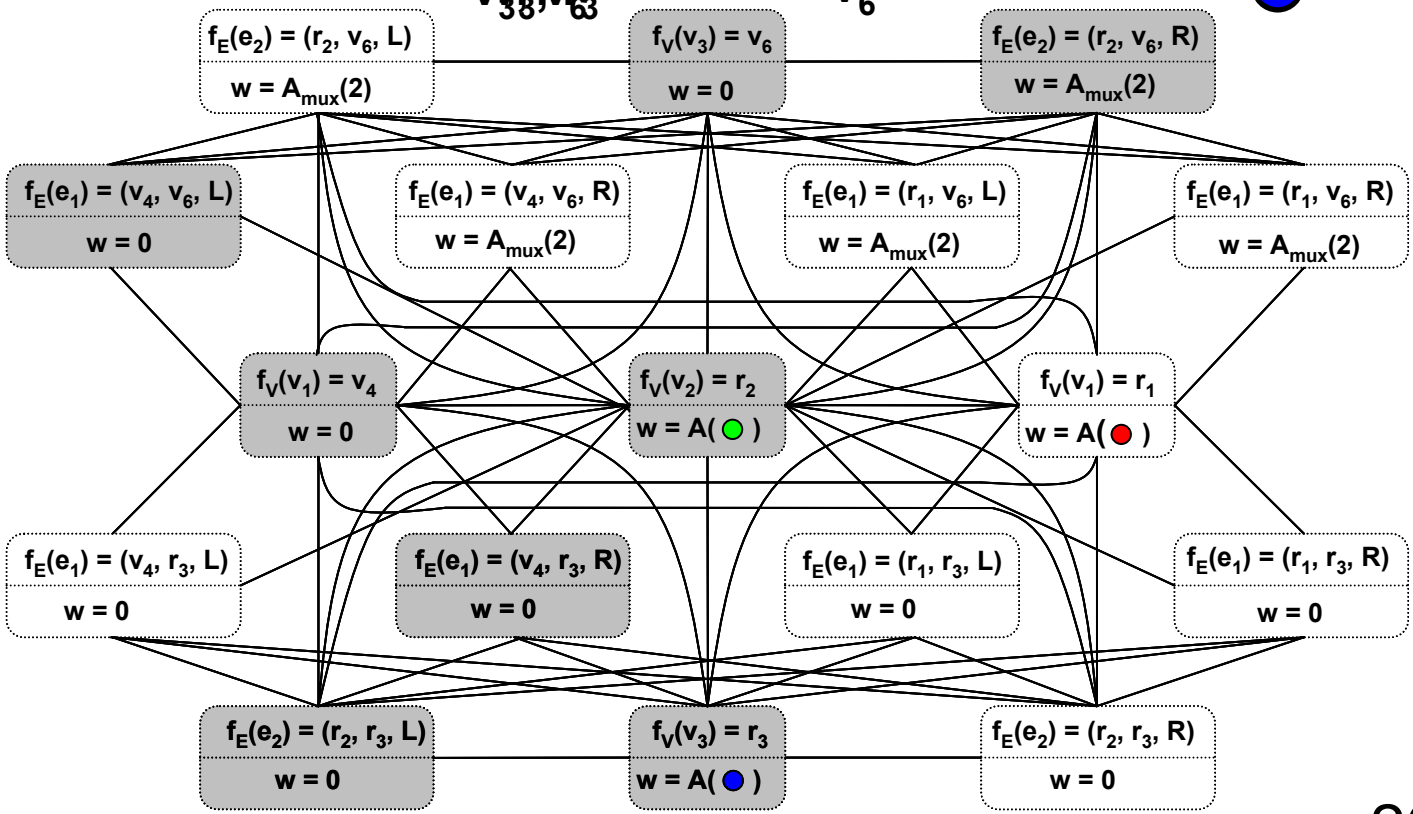
Compatibility Graph











Experimental Setup

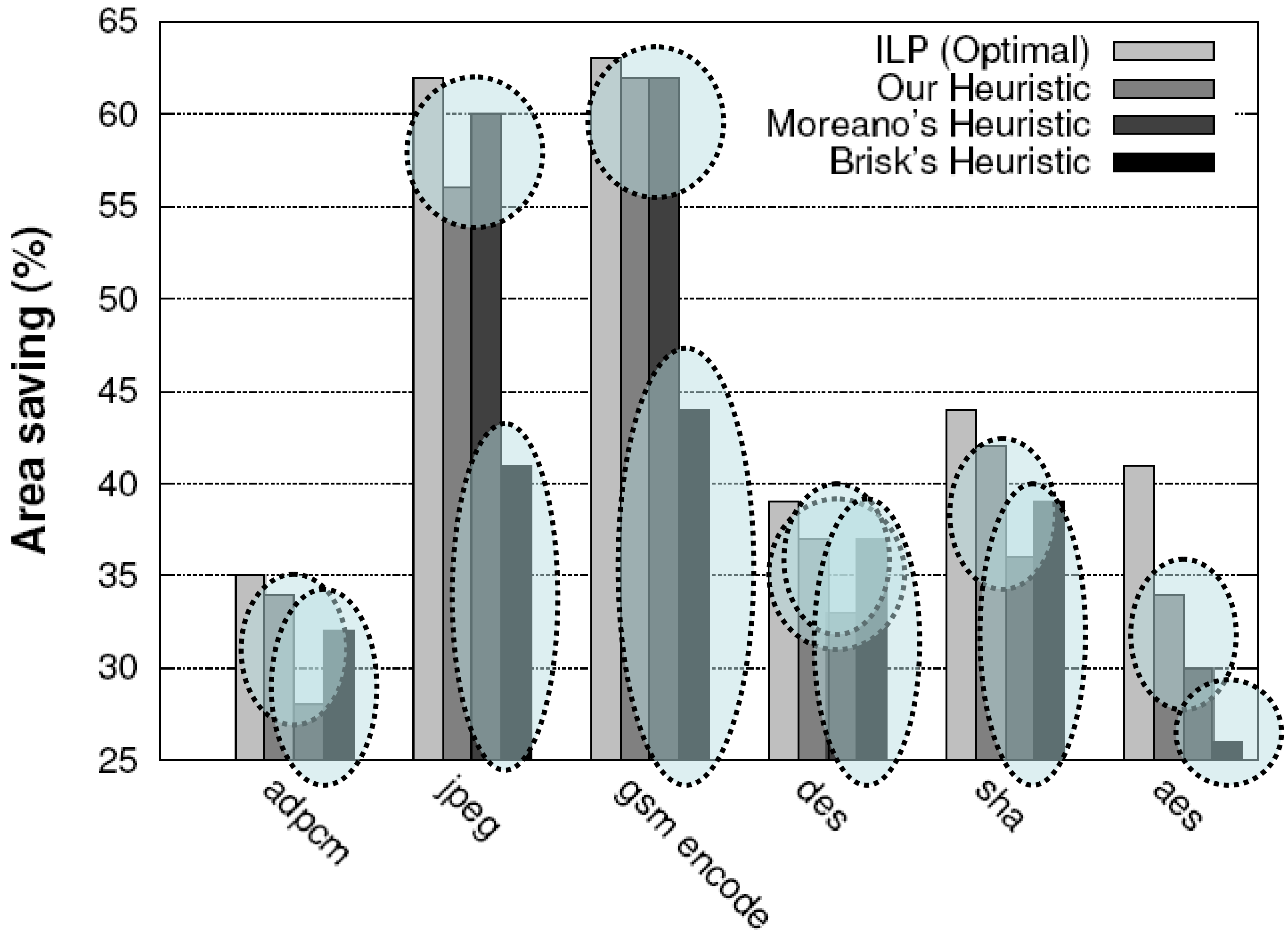
- Internally developed research compiler
  - 1-cycle ISEs [Atasu et al., DAC '03]
  - RF has 2 read ports, 1 write port
  - Standard cell design flow, 0.18μm technology node

- Five DPM algorithms

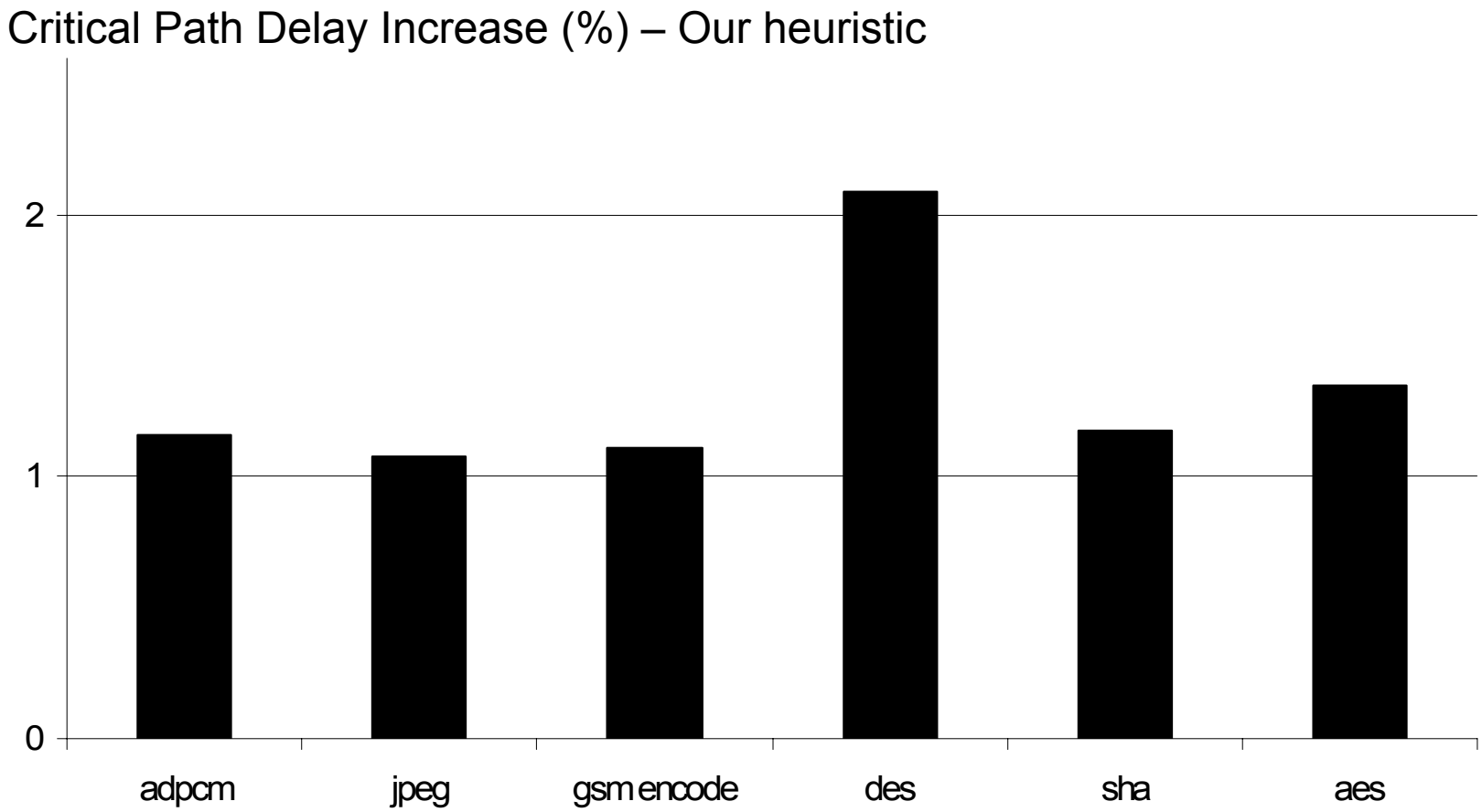
– Baseline	No resource sharing
– ILP (Optimal)	[This paper]
– Our heuristic*	[This paper]
– Moreano's heuristic*	[Moreano et al., TCAD '05]
– Brisk's heuristic	[Brisk et al., DAC '04]

* Max-cliques found by "Cliquer"

1-cycle ISE Area Savings



Critical Path Delay Increase



Runtimes

• Baseline	0
• Optimal (ILP)	3-8 hours
• Our heuristic	2-10 minutes
• Moreano's heuristic	~1 minute
• Brisk's heuristic	$\leq$ 5-10 seconds

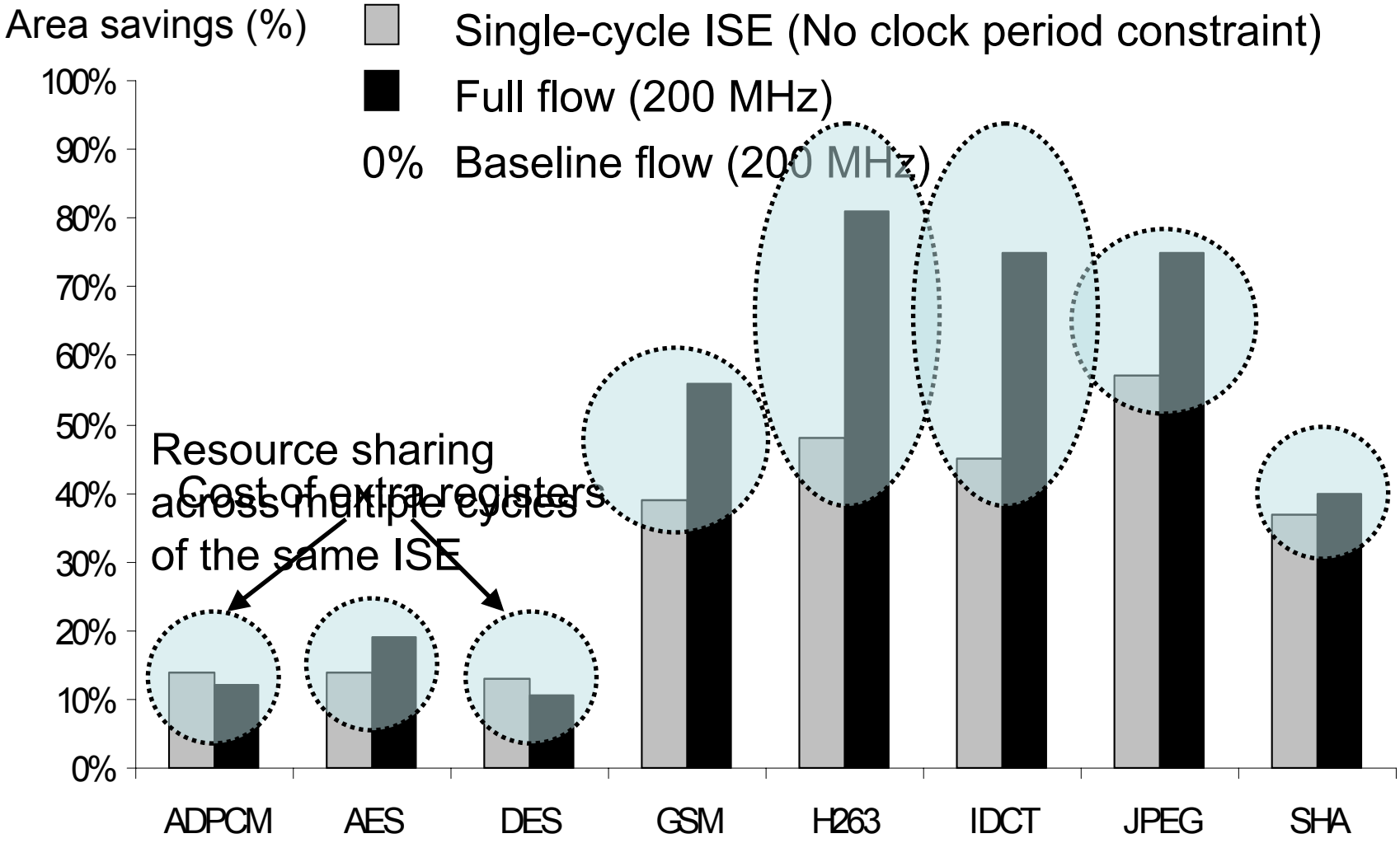
Experimental Setup

- Internally developed research compiler
  - Multi-cycle ISEs [Pozzi and Ienne, CASES '05]
  - RF has 5 read ports, 2 write ports
  - Standard cell design flow, 0.18μm technology node

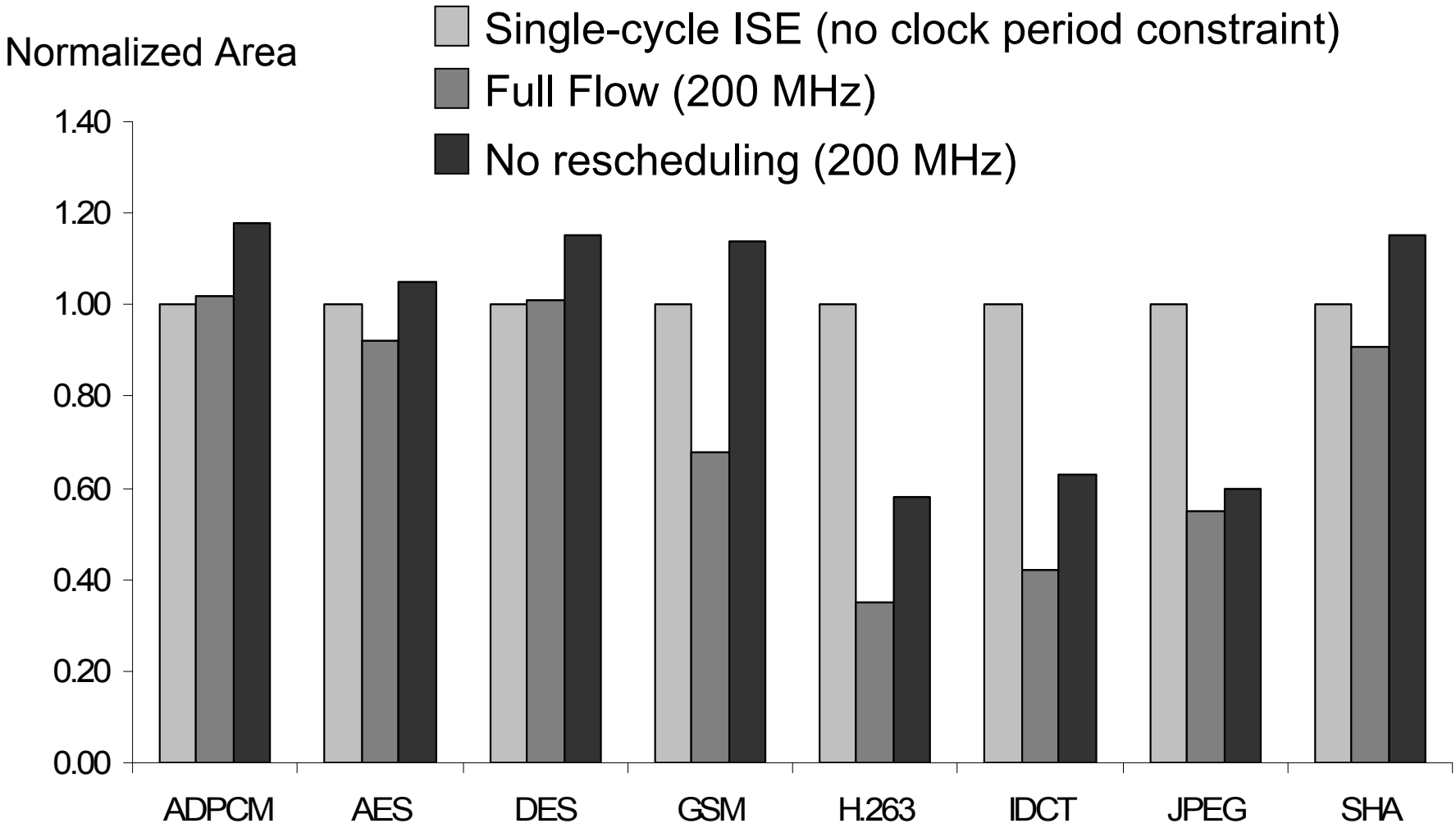
- Four versions of our flow
  - Single-cycle ISE (None) (1-cycle)
  - Full flow (200 MHz) (Multi-cycle)
  - No rescheduling (200 MHz) (Multi-cycle)
  - Baseline flow (200 MHz) (Multi-cycle)

    (Resource sharing and binding step disabled)

Single vs. Multi-cycle ISEs



Impact of Rescheduling



Conclusion

- HLS Flow for ISEs
  - RF I/O constraints
    - Min-latency scheduling is NP-complete
  - Requires two scheduling steps
    - Rescheduling is important for area reduction
- Resource allocation and binding
  - Modeled as a datapath merging problem
  - New problem formulation
    - Multiplexer cost
    - Port assignment