



Communication Based Architecture Design for Systems-On-Chip

Umit Y. Ogras, Radu Marculescu

Electrical and Computer Engineering
Carnegie Mellon University

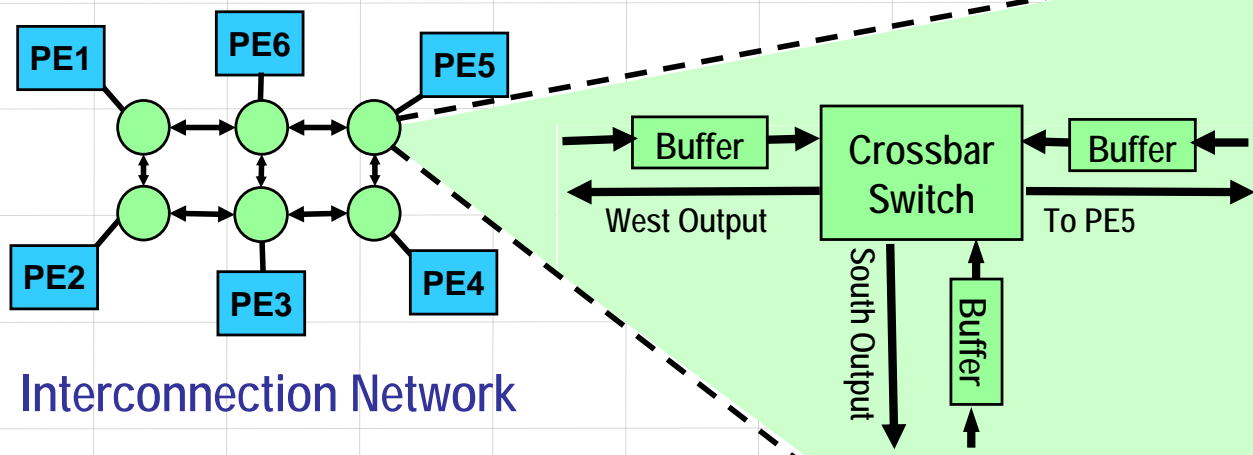


Outline

- Networks-on-Chip
 - ◆ Design space
 - ◆ Design trade-offs
- Communication-based customized architecture synthesis
 - ◆ Design methodology
 - ◆ Graph decomposition algorithm
- Experimental results
- Conclusion

Networks-on-Chip (NoCs)

- Set of communicating nodes
 - ◆ Each node consists of a core and an embedded router
 - ◆ Nodes are connected through an interconnection network
 - ◆ Nodes communicate by sending packets



3

NoC Design Space

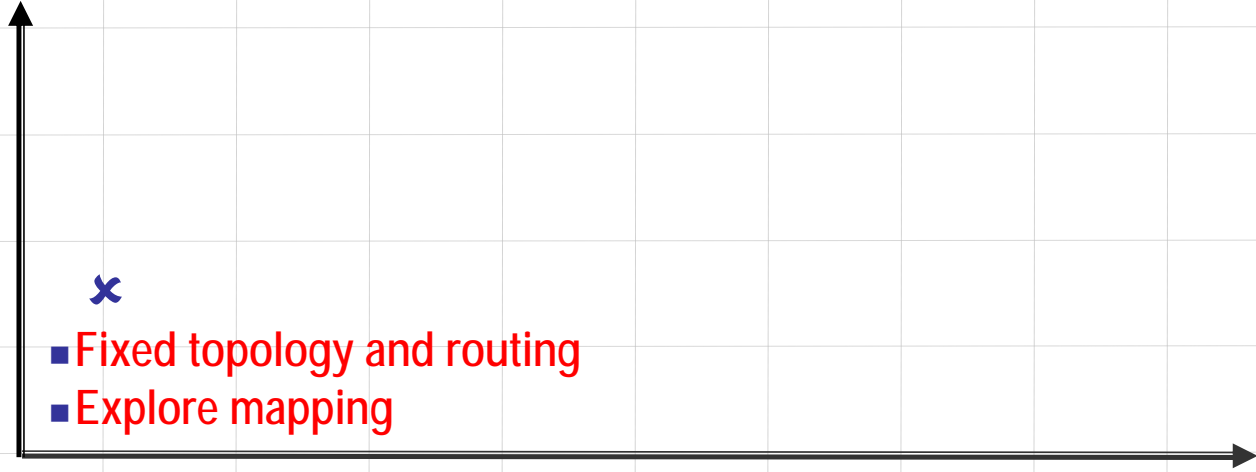
- Communication infrastructure:
Topology (Mesh, hypercube,...)
- Communication paradigm:
Routing Decision (Adaptive, deterministic,...)
- Communication traffic
Application (Uniform, bursty,...)



4

NoC Design Approaches

Design effort



Design quality

NoC Design Approaches

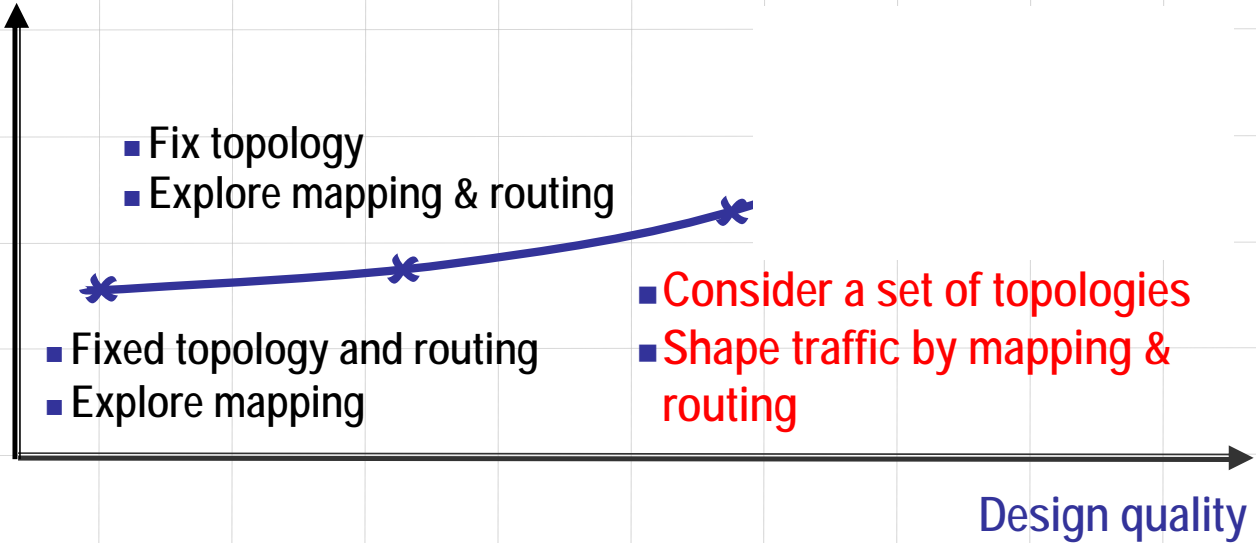
Design effort



Design quality

NoC Design Approaches

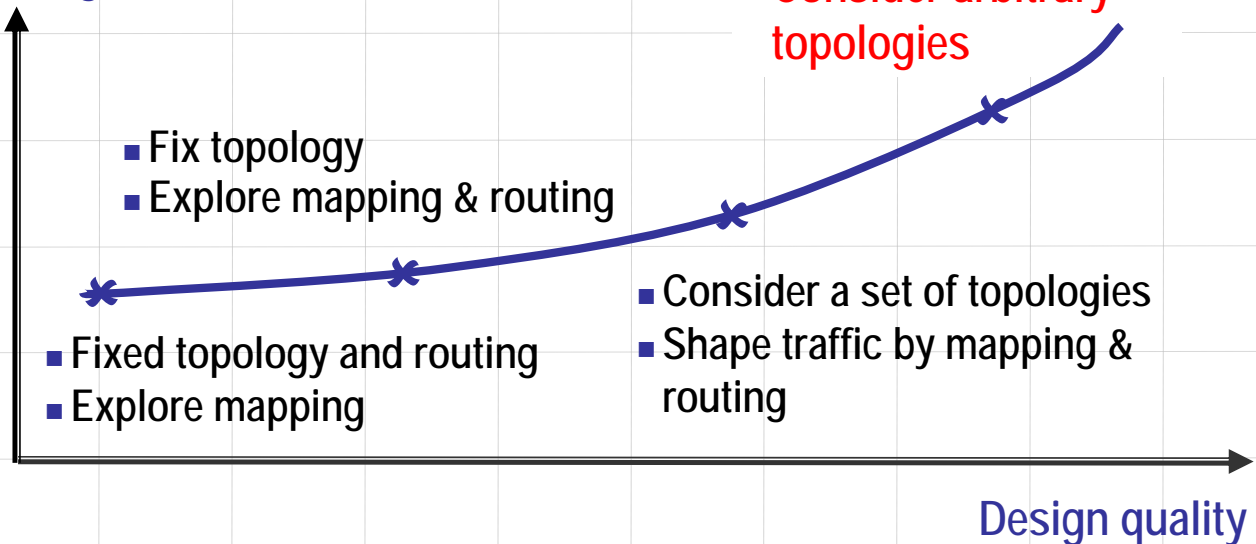
Design effort



5

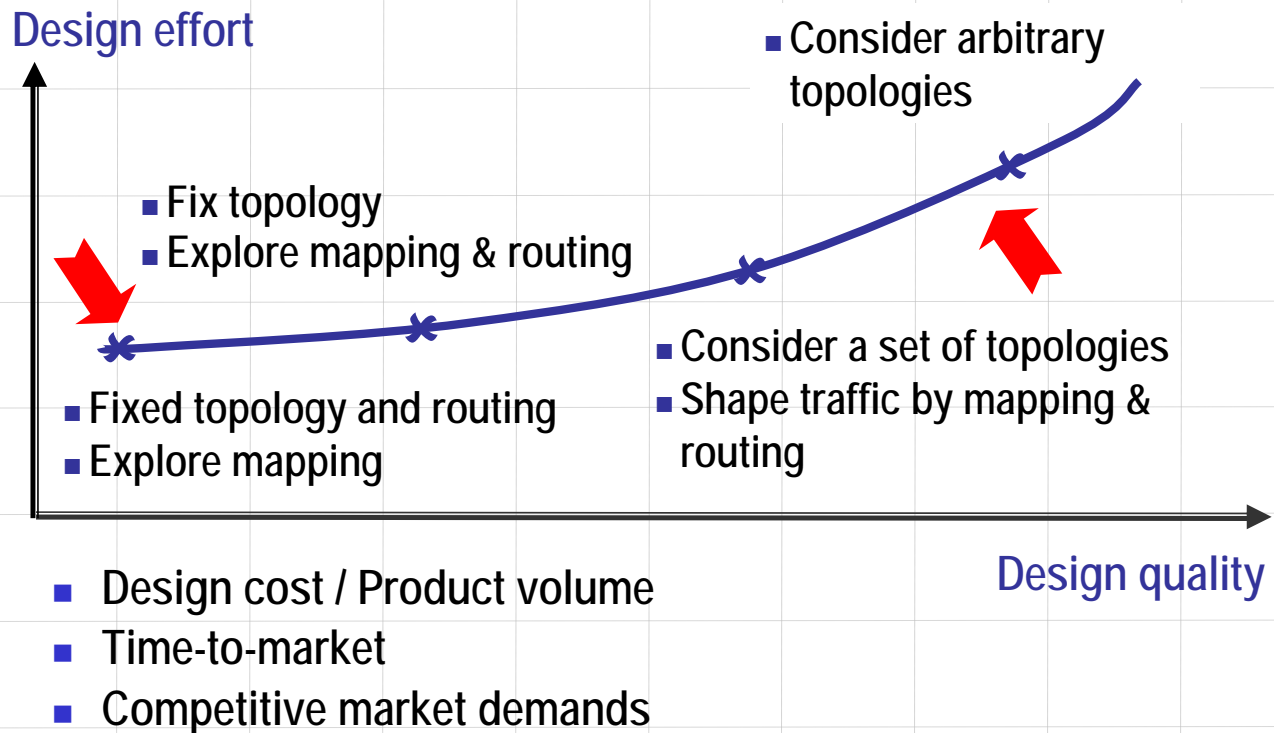
NoC Design Approaches

Design effort



5

NoC Design Approaches



5

Customized Architecture Synthesis

- Express a general communication problem in terms of the primitives
 - ◆ Primitives are the alphabet
 - ◆ Like the standard cells in ASIC design
- Generic patterns are
 - ◆ One-to-all or Broadcast,
 - ◆ One-to-many or Multicast
 - ◆ All-to-all or Gossiping
- Gossiping between 8 nodes

7

Customized Architecture Synthesis

- Express a general communication problem in terms of the primitives
 - ◆ Primitives are the alphabet
 - ◆ Like the standard cells in ASIC design
- Generic patterns are
 - ◆ One-to-all or Broadcast,
 - ◆ One-to-many or Multicast
 - ◆ All-to-all or Gossiping
- Gossiping between 8 nodes

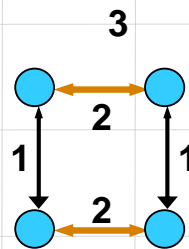


$$t_{opt} = \lceil \log_2 N \rceil$$

7

Customized Architecture Synthesis

- Express a general communication problem in terms of the primitives
 - ◆ Primitives are the alphabet
 - ◆ Like the standard cells in ASIC design
- Generic patterns are
 - ◆ One-to-all or Broadcast,
 - ◆ One-to-many or Multicast
 - ◆ All-to-all or Gossiping
- Gossiping between 8 nodes

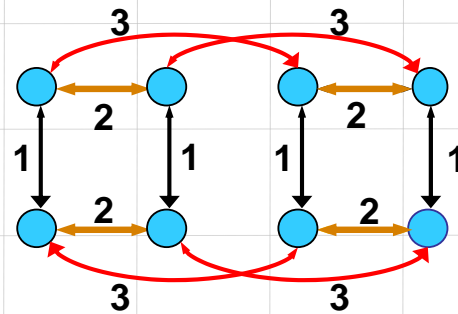


$$t_{opt} = \lceil \log_2 N \rceil$$

7

Customized Architecture Synthesis

- Express a general communication problem in terms of the primitives
 - ◆ Primitives are the alphabet
 - ◆ Like the standard cells in ASIC design
- Generic patterns are
 - ◆ One-to-all or Broadcast,
 - ◆ One-to-many or Multicast
 - ◆ All-to-all or Gossiping
- Gossiping between 8 nodes



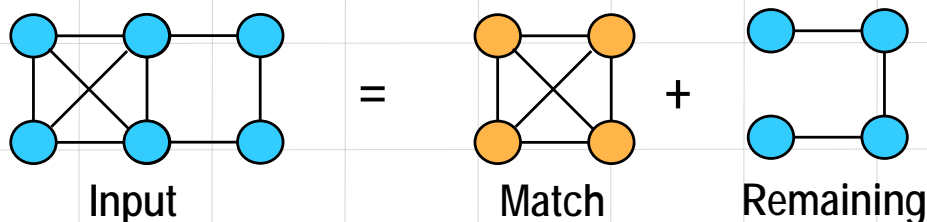
$$t_{opt} = \lceil \log_2 N \rceil$$

7

Problem Formulation

- **Given** the target application represented by an Application Characterization Graph G
- **Decompose** G into a set of communication primitives

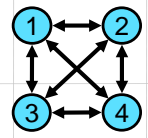
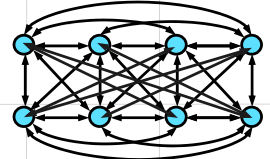
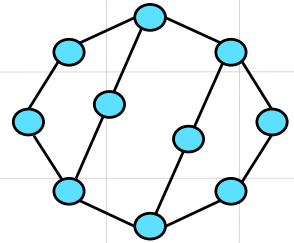
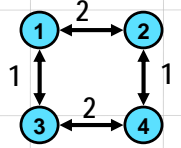
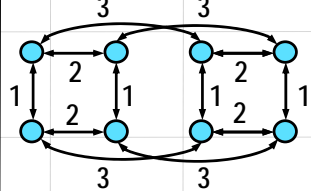
$$G = \sum_{L_i \in D} M_i(L_i) + R(V_R, E_R)$$



- ◆ Such that *total energy consumption* is minimized, with a constraint on the *total wire length*

8

Communication Library

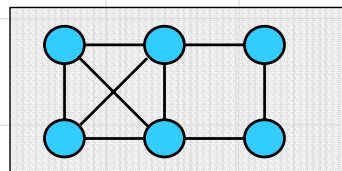
	Gossip Graph of size 4	Gossip Graph of size 8	Broadcast Graph of size 10
Representation Graph (Used in analysis)			
Implementation Graph (Used in synthesis)			Implementation

- Representation graph: Searched in the input graph
- Implementation graph: Optimal implementation of the communication primitive

9

Graph Decomposition Algorithm

Input Graph

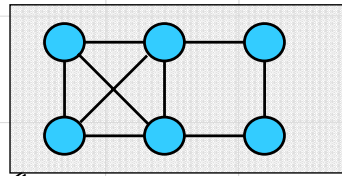


- Match: A subgraph isomorphism from a communication primitive to the input graph

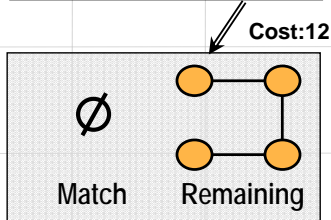
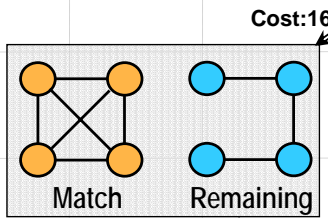
10

Graph Decomposition Algorithm

Input Graph



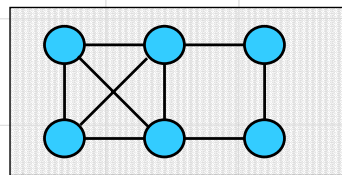
- Match: A subgraph isomorphism from a communication primitive to the input graph



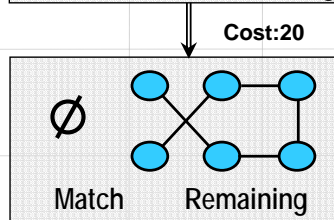
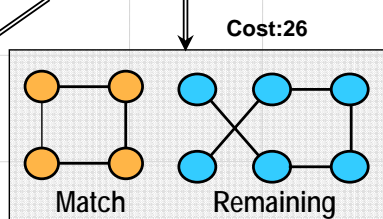
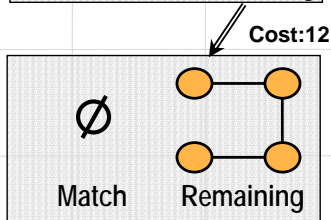
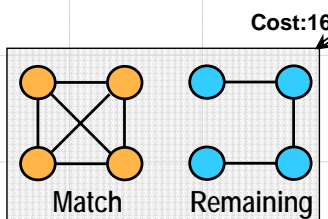
10

Graph Decomposition Algorithm

Input Graph

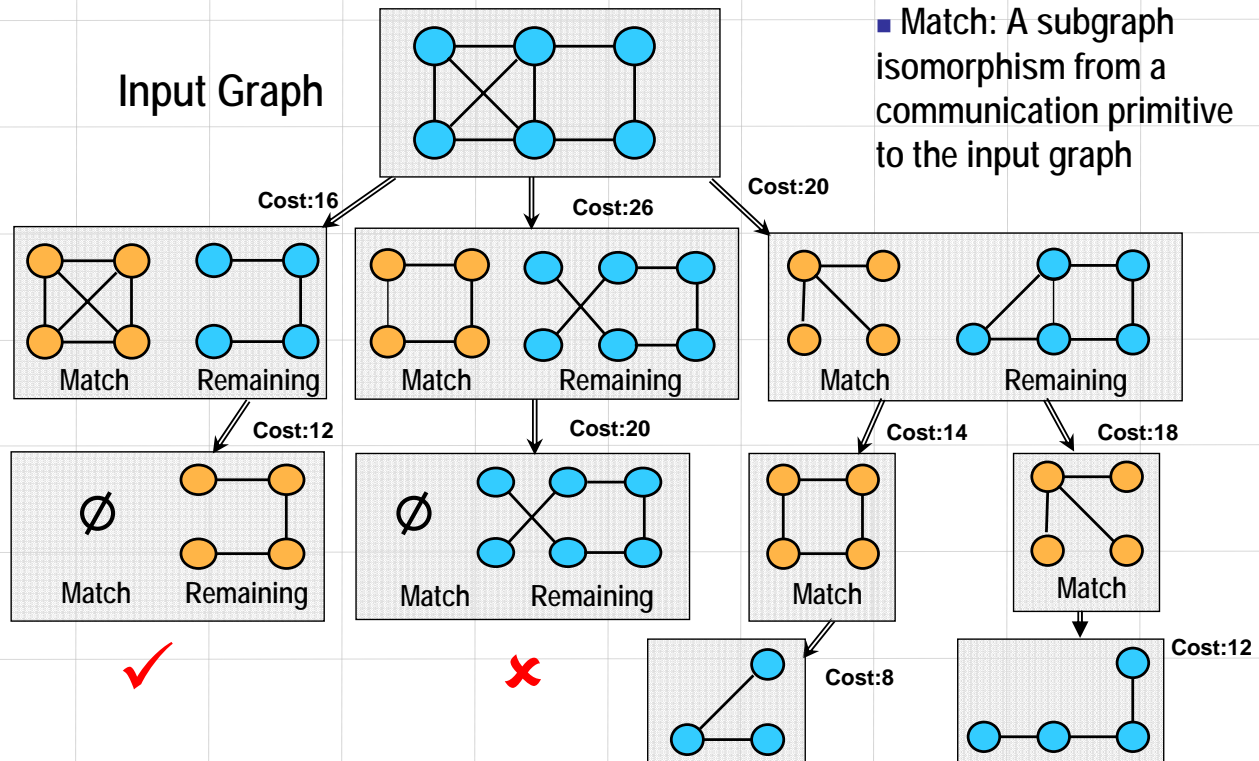


- Match: A subgraph isomorphism from a communication primitive to the input graph

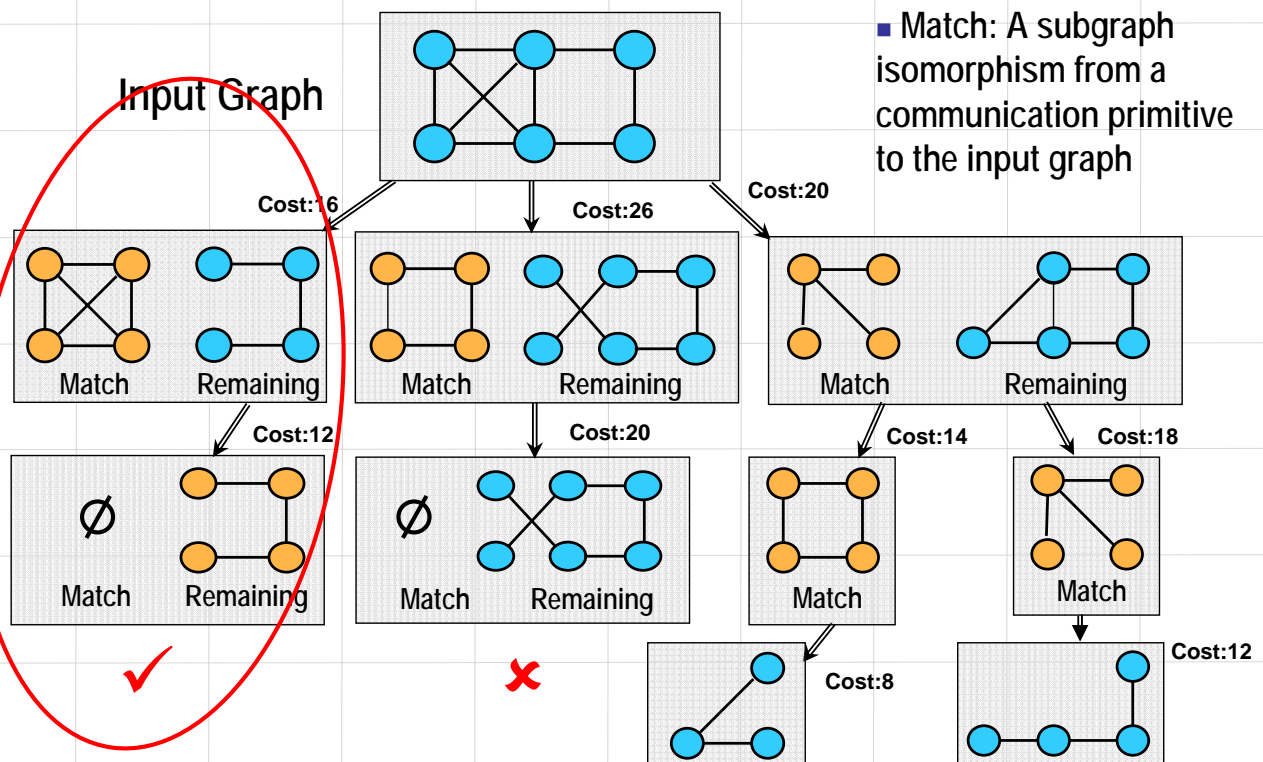


10

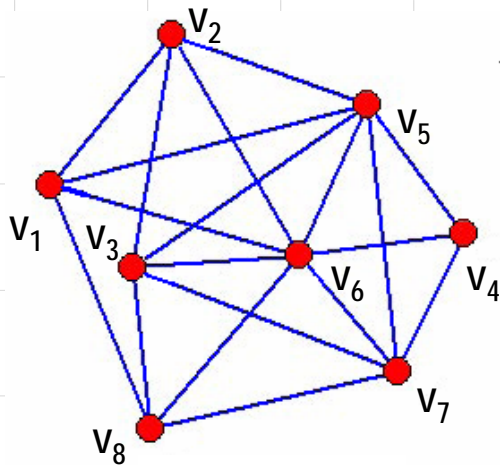
Graph Decomposition Algorithm



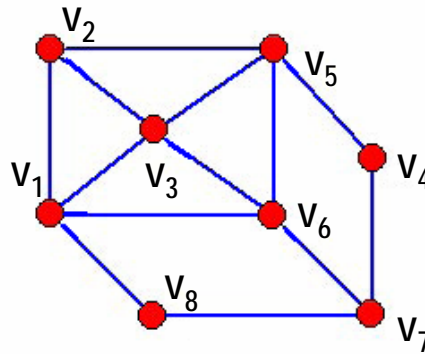
Graph Decomposition Algorithm



Experimental Results – Illustration



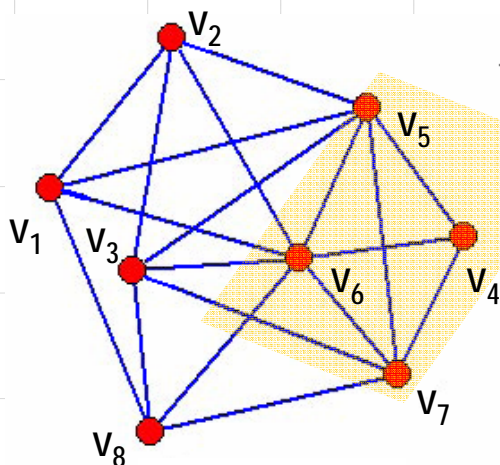
Input graph



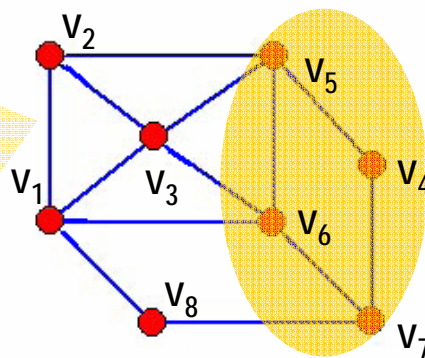
Custom Implementation

* The graphs are generated and drawn using Pajek

Experimental Results – Illustration



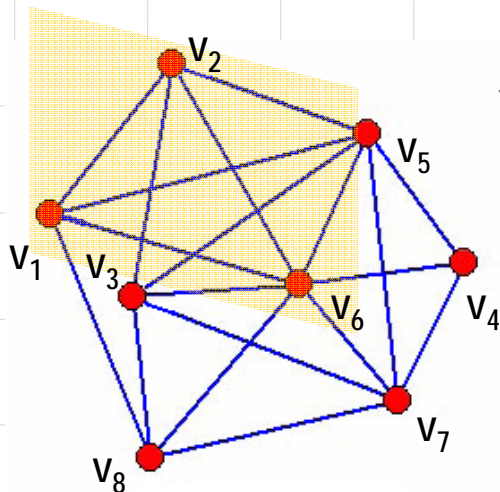
Input graph



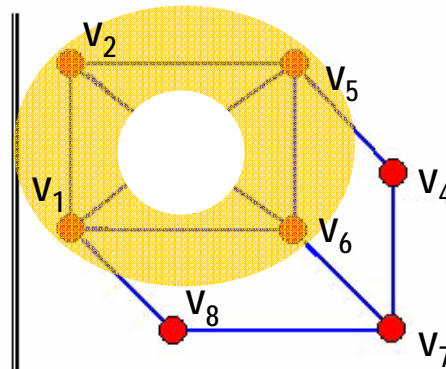
Custom Implementation

* The graphs are generated and drawn using Pajek

Experimental Results – Illustration



Input graph

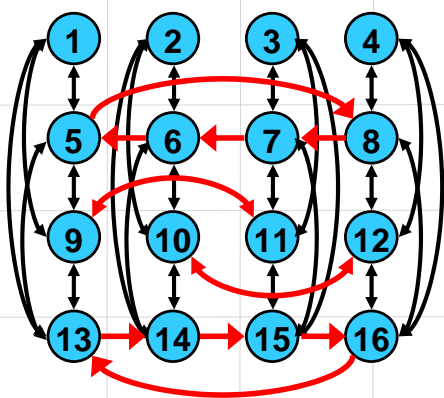


Custom Implementation

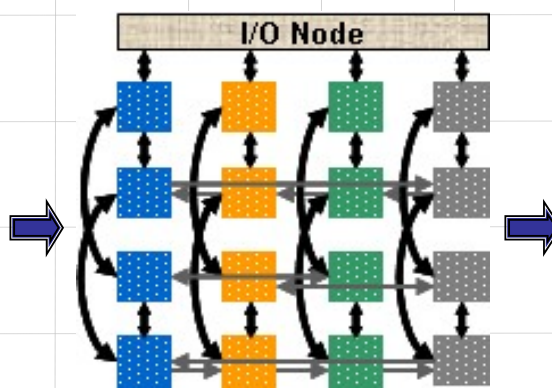
* The graphs are generated and drawn using Pajek

Experimental Results – AES

- Advanced Encryption Standard is distributed to 16 nodes
- Identical cores process 1 byte of the 128-bit input block



Application Characterization



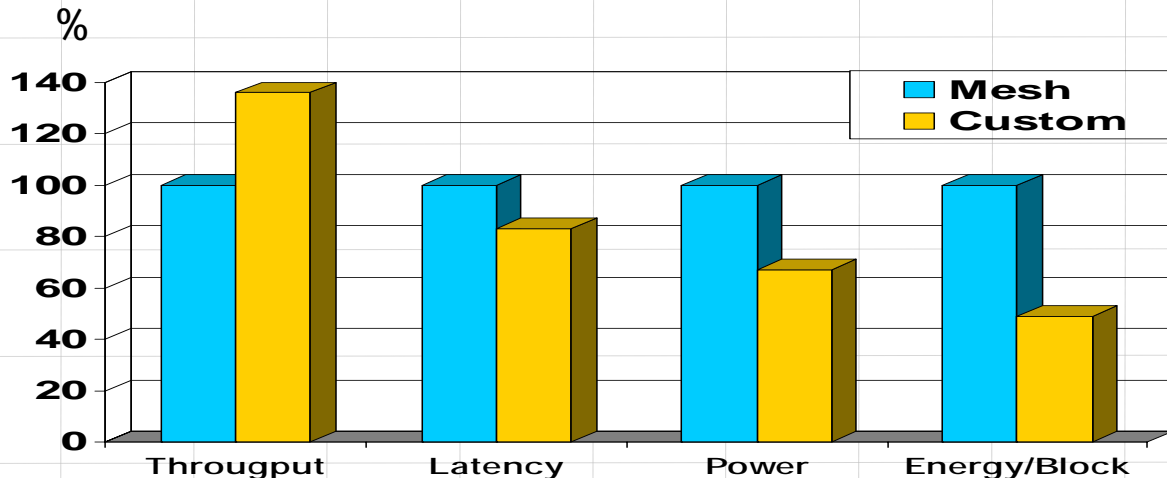
Communication Architecture



Prototype

Experimental Results – AES

- Advanced Encryption Standard is distributed to 16 nodes
- Identical cores process 1 byte of the 128-bit input block



* Normalized values

12

Summary and Conclusion

- Introduce a design methodology to synthesize communication-based customized architecture
- Use graph decomposition algorithm for automated architecture synthesis
- Demonstrate the approach on a real driver application

Thanks, for more questions:

<http://www.ece.cmu.edu/~sld/>

13