Constraint-Driven Bus Matrix Synthesis for MPSoC

Sudeep Pasricha†, Nikil Dutt† and Mohamed Ben-Romdhane‡

†ACES (Architectures and Compilers for Embedded Systems) Lab
Center for Embedded Computer Systems (CECS)
University of California, Irvine
{sudeep,dutt}@cecs.uci.edu

‡Conexant Systems Inc
Newport Beach, CA
m.benromdhane@conexant.com

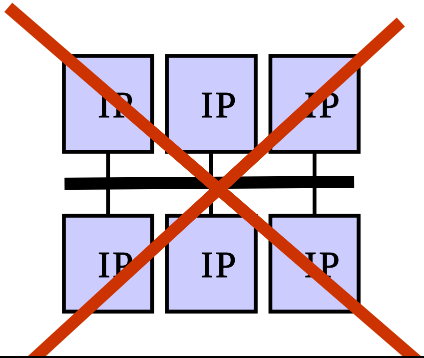
Outline

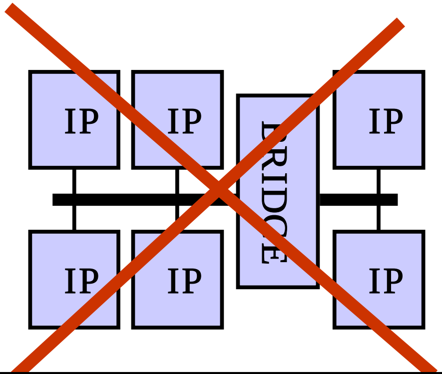
- Motivation
- Related Work
- Problem Formulation
- Bus Matrix Synthesis (BMSYN) Approach
- Case Studies
- Conclusion

Importance of Communication Architectures

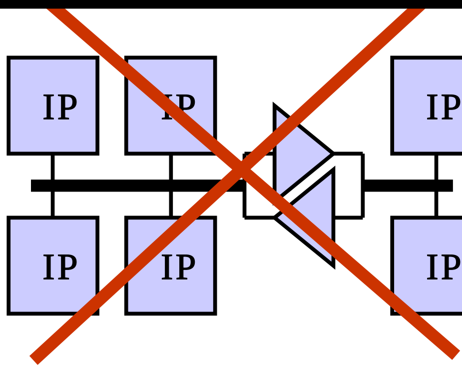
- Improving process technology has led to increasing number of cores being integrated on a single SoC
  - Tens to hundreds of cores in today's MPSoCs
- Sharp increase in overall on-chip communication
  - Next generation of multimedia, broadband and networking apps
  - Communication is fast becoming a major design bottleneck!
- Standard bus architectures such as AMBA, CoreConnect and STBus are popular choices for handling on-chip communication
  - Relatively simple to design
  - Low area overhead

Typical Bus-based Communication Architectures for SoCs

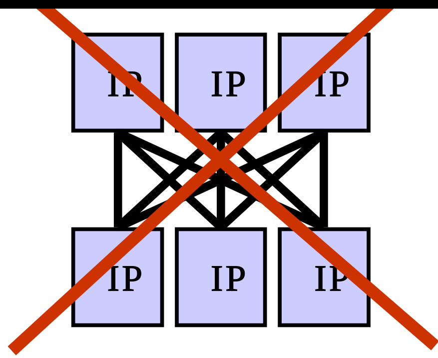




Not scalable to meet the communication demands of modern high performance MPSoC systems!



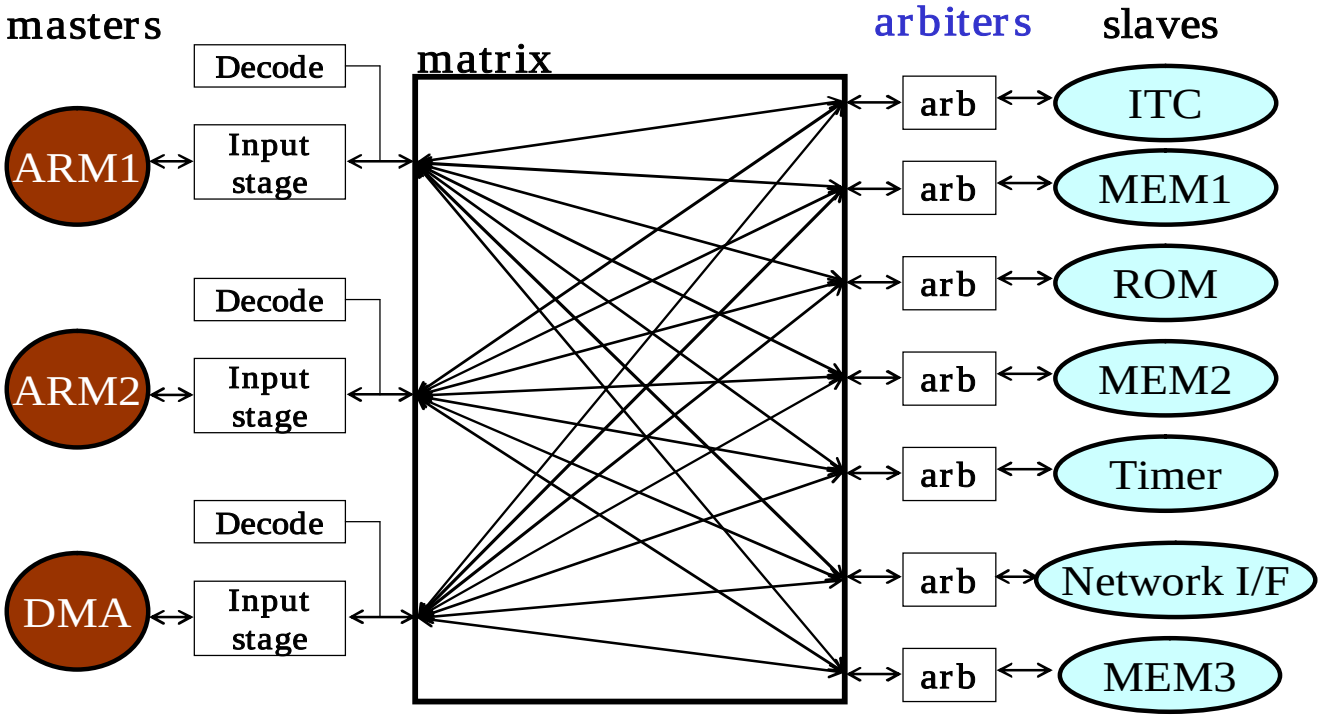
c) split-bus



d) point-to-point bus

Bus Matrix Communication Architectures

- Recent trend has been to use Bus Matrix communication architectures to support high bandwidths for modern MPSoC systems
  - AMBA, CoreConnect, STBus all support matrix configurations



Full Bus Matrix

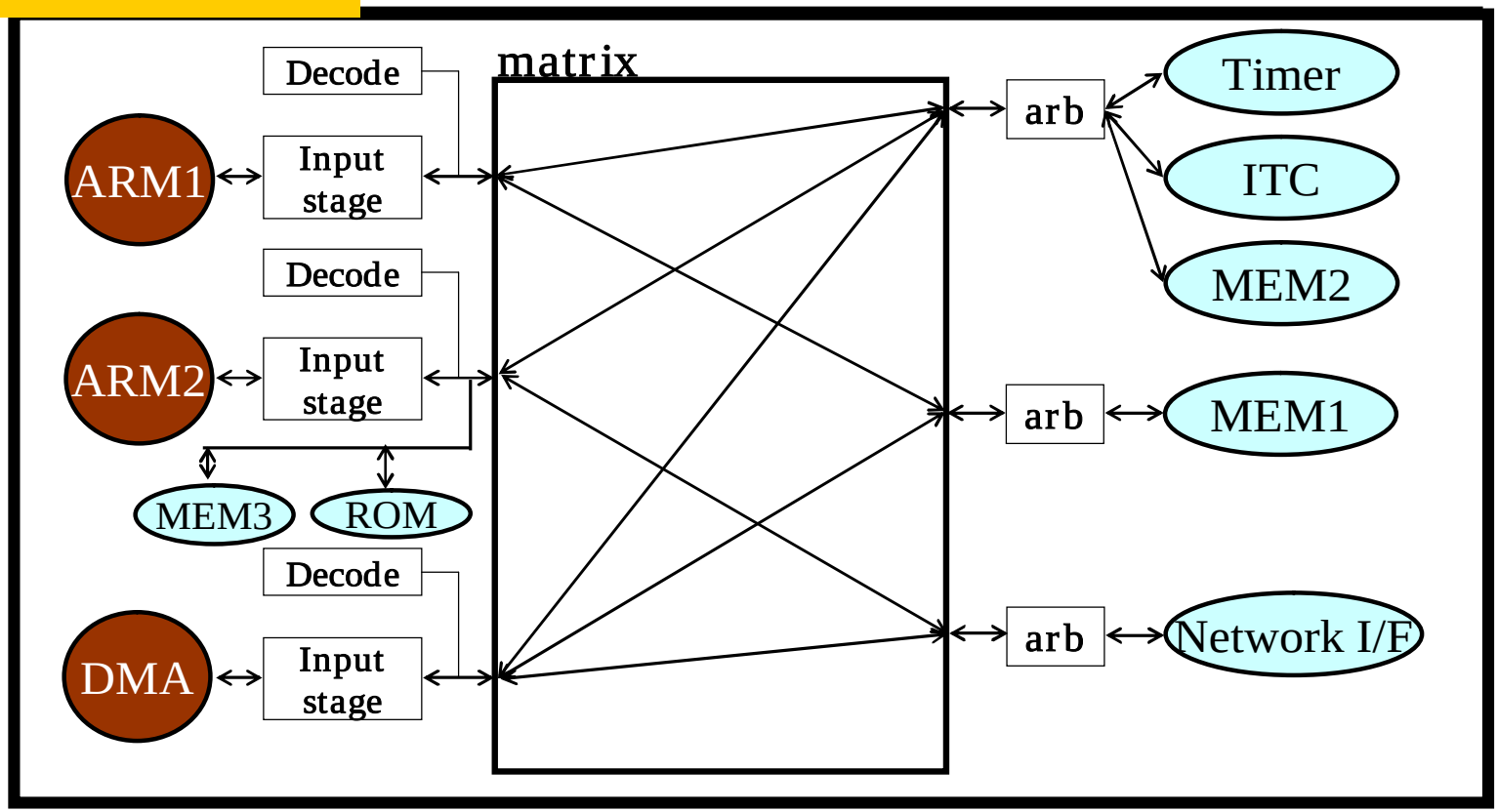
Bus Matrix Communication Architectures

- A major drawback of a full bus matrix architecture is that it connects every master to every slave with a bus
  - Results in prohibitively large number of busses
  - High cost of implementation!
  - Practically impossible to route and achieve timing closure

- One solution is to “tailor” the bus matrix according to the application, to create a partial bus matrix which still meets application performance requirements
  - Has fewer busses
    - Consequently fewer arbiters, decoders, buffers
  - Maximizes bus utilization
  - Reduces implementation cost, area and power dissipation

Bus Matrix Communication Architectures

partial bus matrix



- Goal is to automatically synthesize a partial bus matrix with minimal number of buses, and which meets all performance requirements of the application

Outline

- Motivation
- Related Work
- Problem Formulation
- Bus Matrix Synthesis (BMSYN) Approach
- Case Studies
- Conclusion

Related Work

- Need for Bus Matrix Communication Architectures
  - Ryu et al. [DSS 2001], Lahtinen et al. [ISCAS 2003] compared bus matrix with other bus based topologies
    - bus matrix outperformed the other choices due to its superior parallel response
  - Loghi et al. [DATE 2004] presented exploration studies with the AMBA and STBus shared bus, full matrix and partial matrix topologies
    - matrix topologies are much better suited for high throughput systems requiring frequent parallel accesses
    - partial matrix schemes can perform just as well as the full matrix architectures, if designed carefully

Related Work

- Plenty of work in area of shared/hierarchical bus synthesis
  - Lahiri et al [ICCAD 2000], Lyonnard et al [DAC 2001]
  - Pinto et al [DAC 2002], Ryu et al [DATE 2003]
  - Pasricha et al [ASPDAC 2005], [DAC 2005]

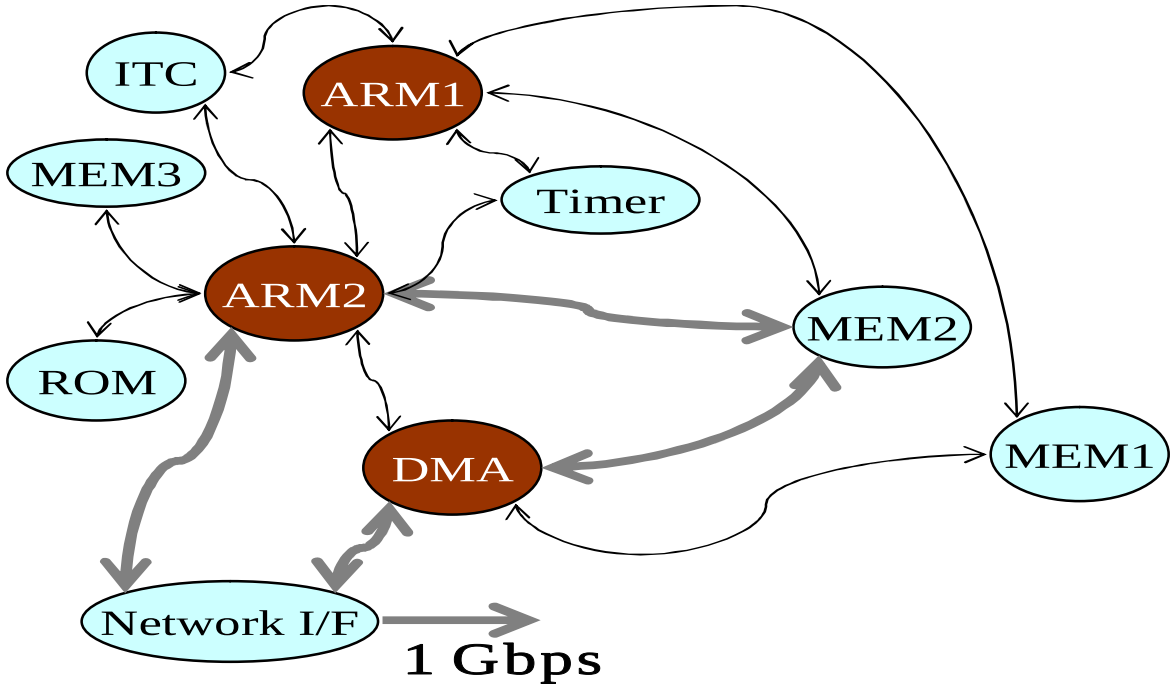
- However, very few research efforts have looked at bus matrix synthesis
  - Ogawa et al. [DATE 2003] proposed a transaction based simulation environment to explore and design a bus matrix
    - manually specify topology, arbitration scheme, memory mapping
    - too time consuming
  - Murali et al. [DATE 2005] come closest to our goal of automated application specific bus matrix synthesis. However,
    - work focuses on automated matrix topology synthesis
    - communication parameters which considerably influence system performance are not synthesized
    - our approach synthesizes both topology AND parameter values

Outline

- Motivation
- Related Work
- Problem Formulation
- Bus Matrix Synthesis (BMSYN) Approach
- Case Studies
- Conclusion

MPSoC Performance Constraints

- MPSoC designs have performance constraints that can be represented in terms of Data Throughput Constraints
- Communication Throughput Graph, CTG = G(V,A) incorporates SoC components and throughput constraints
- Throughput Constraint Path (TCP) is a CTG sub-graph



Problem Formulation

- Given:
  - an MPSoC with performance constraints
  - a target bus matrix communication architecture (e.g. AMBA, STBus)

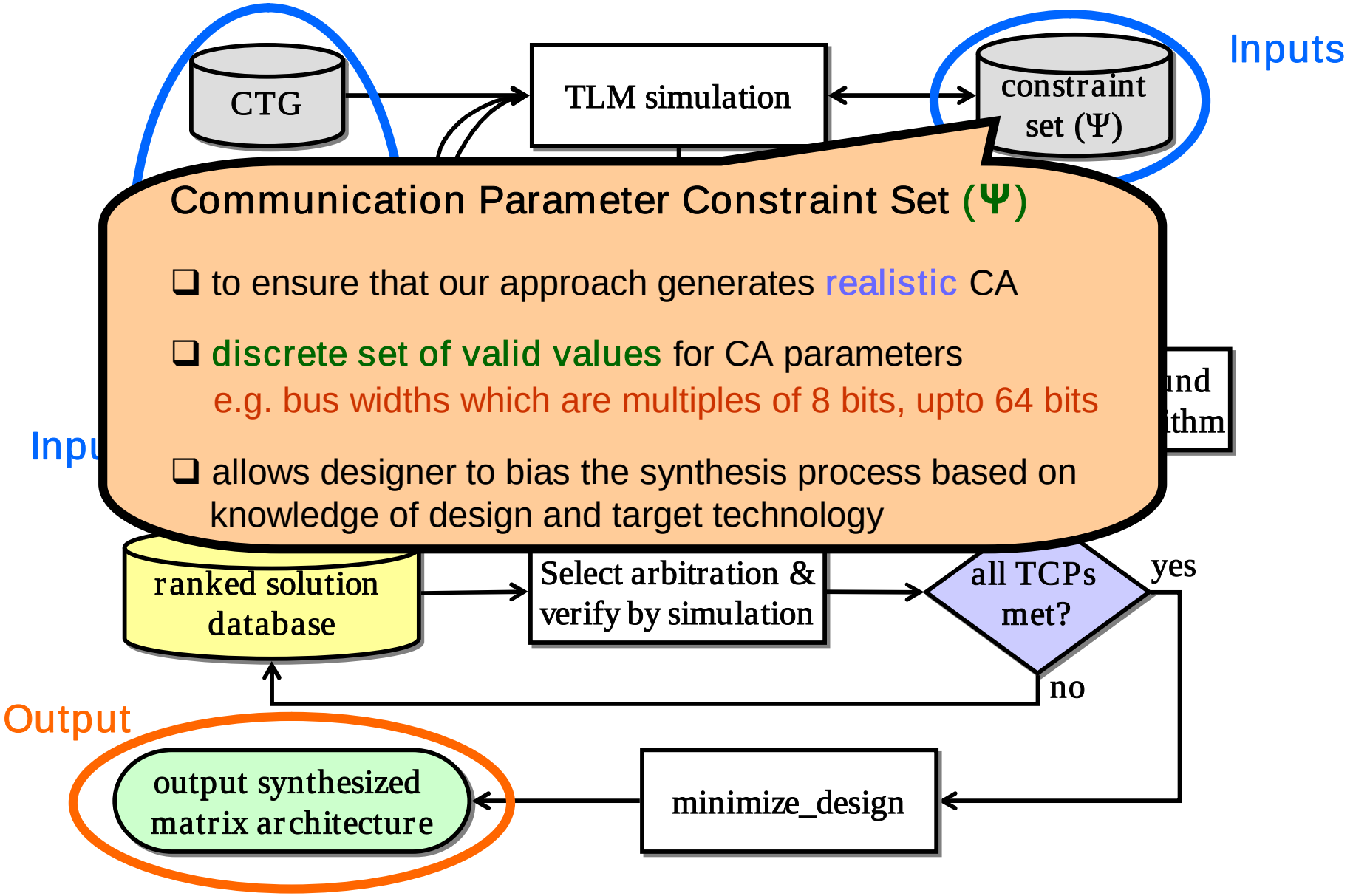
- Assumptions:
  - hardware-software partitioning has been done already
  - IPs are standard non-modifiable “black box” components
  - memories can be split and modified
  - busses within a bus matrix have the same data bus width,
    - typically depends on number of data interface pins of the IPs in the design

- Goals:
  - automatically synthesize bus matrix topology AND parameter values
  - minimize number of busses in matrix
  - satisfy all throughput constraints in the design

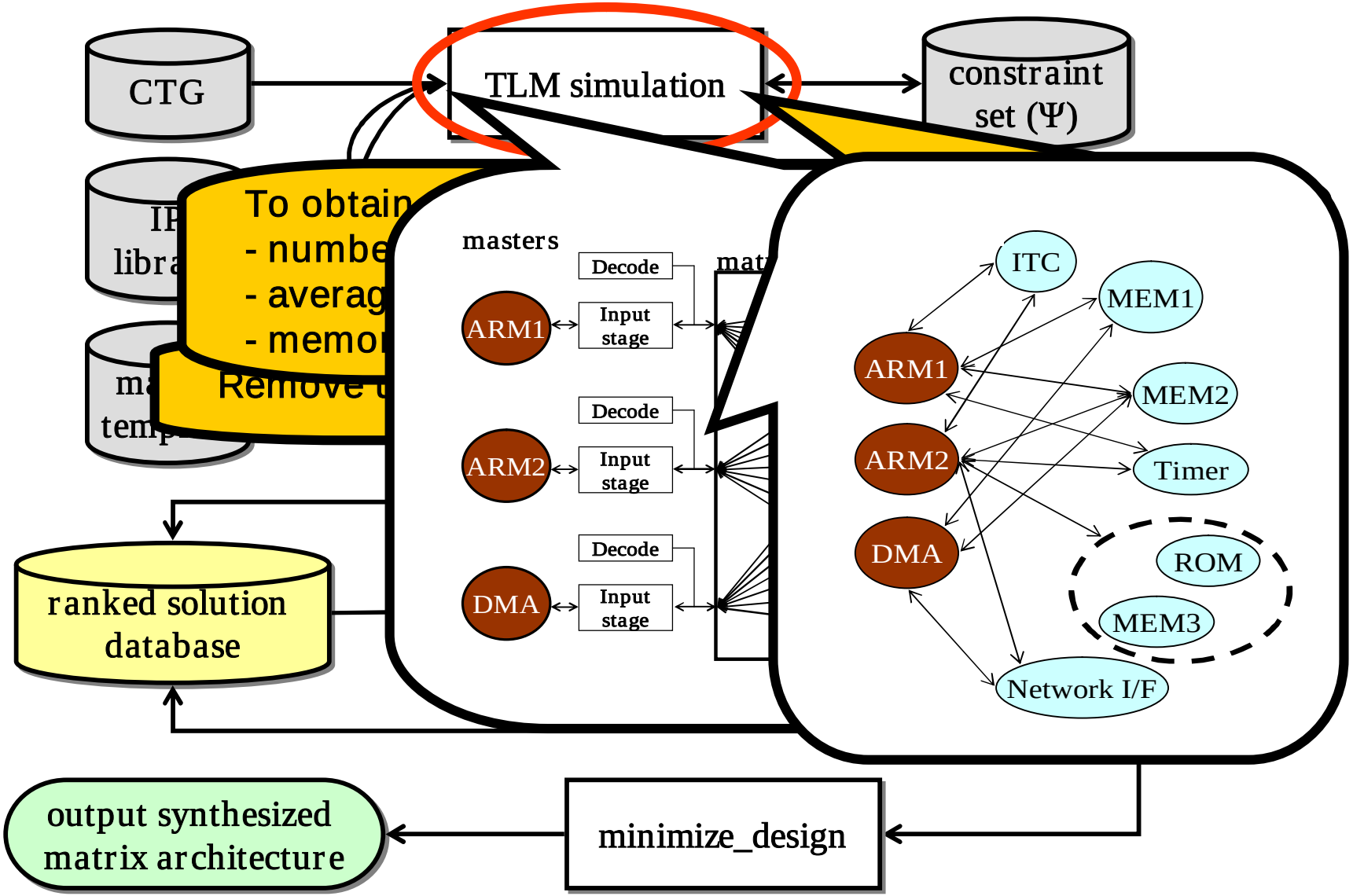
Outline

- Motivation
- Related Work
- Problem Formulation
- Bus Matrix Synthesis (BMSYN) Approach
- Case Studies
- Conclusion

Bus Matrix Synthesis (BMSYN) Approach



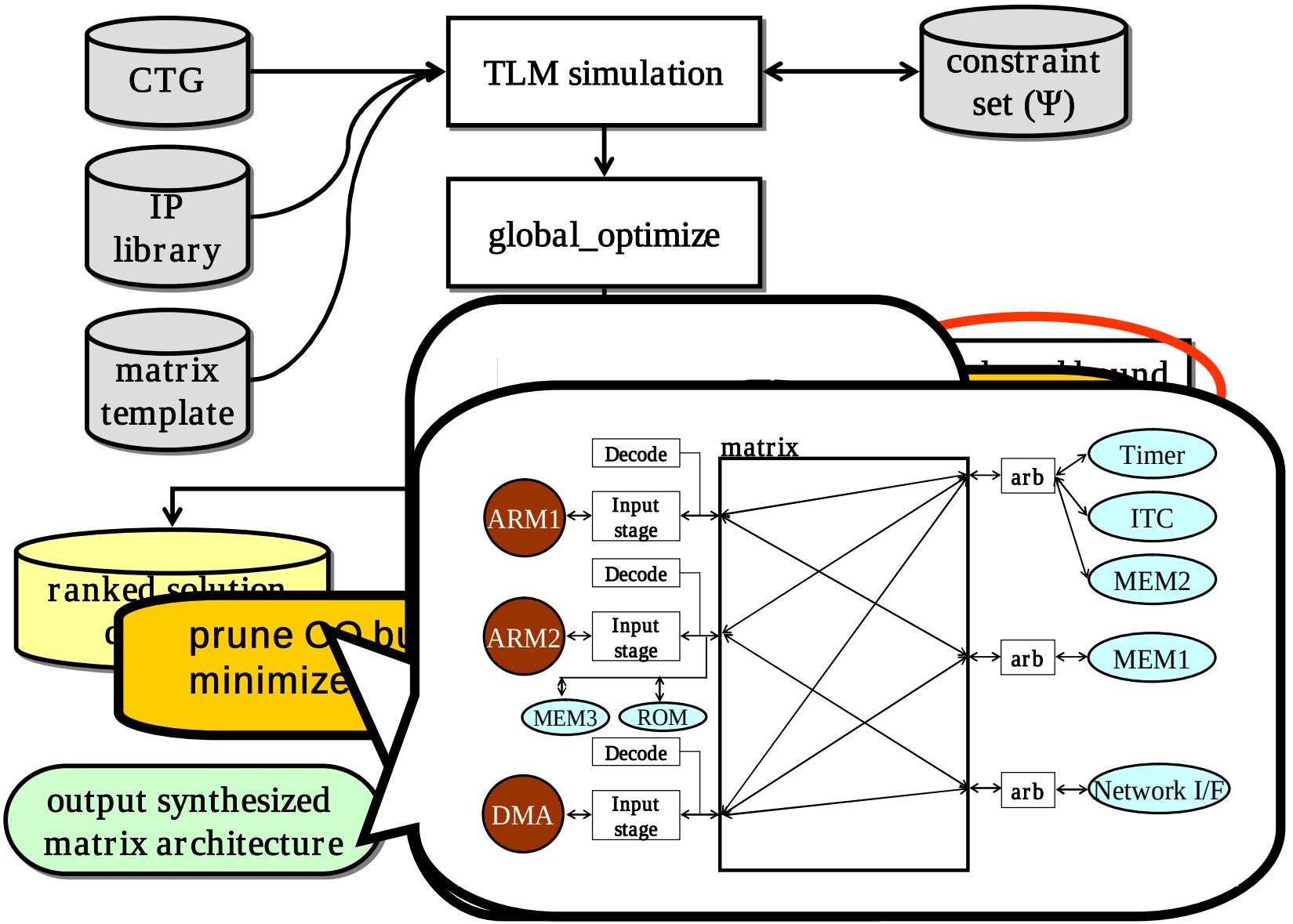
Bus Matrix Synthesis (BMSYN) Approach



Branch and Bound Clustering Algorithm

- Goal: cluster slave modules to minimize matrix cost
- Start by clustering two slave clusters at a time
  - Initially, each slave cluster has only one slave
- However, the total number of clustering configurations possible for n slaves is $(n! \times (n-1)!)/2^{(n-1)}$
  - Extremely large number for even medium sized SoCs!
- Solution: use a powerful Bounding function
  - Called after every clustering operation
  - Uses lookup table to discard duplicate clustering ops
  - Discards non-beneficial clustering (i.e. no savings in no. of busses)
  - Discards incompatible clustering
    - **e.g. mergers of busses with conflicting bus speeds**
  - Discards clustering which violates b/w requirements

Bus Matrix Synthesis (BMSYN) Approach



Outline

- Motivation
- Related Work
- Problem Formulation
- Bus Matrix Synthesis (BMSYN) Approach
- Case Studies
- Conclusion

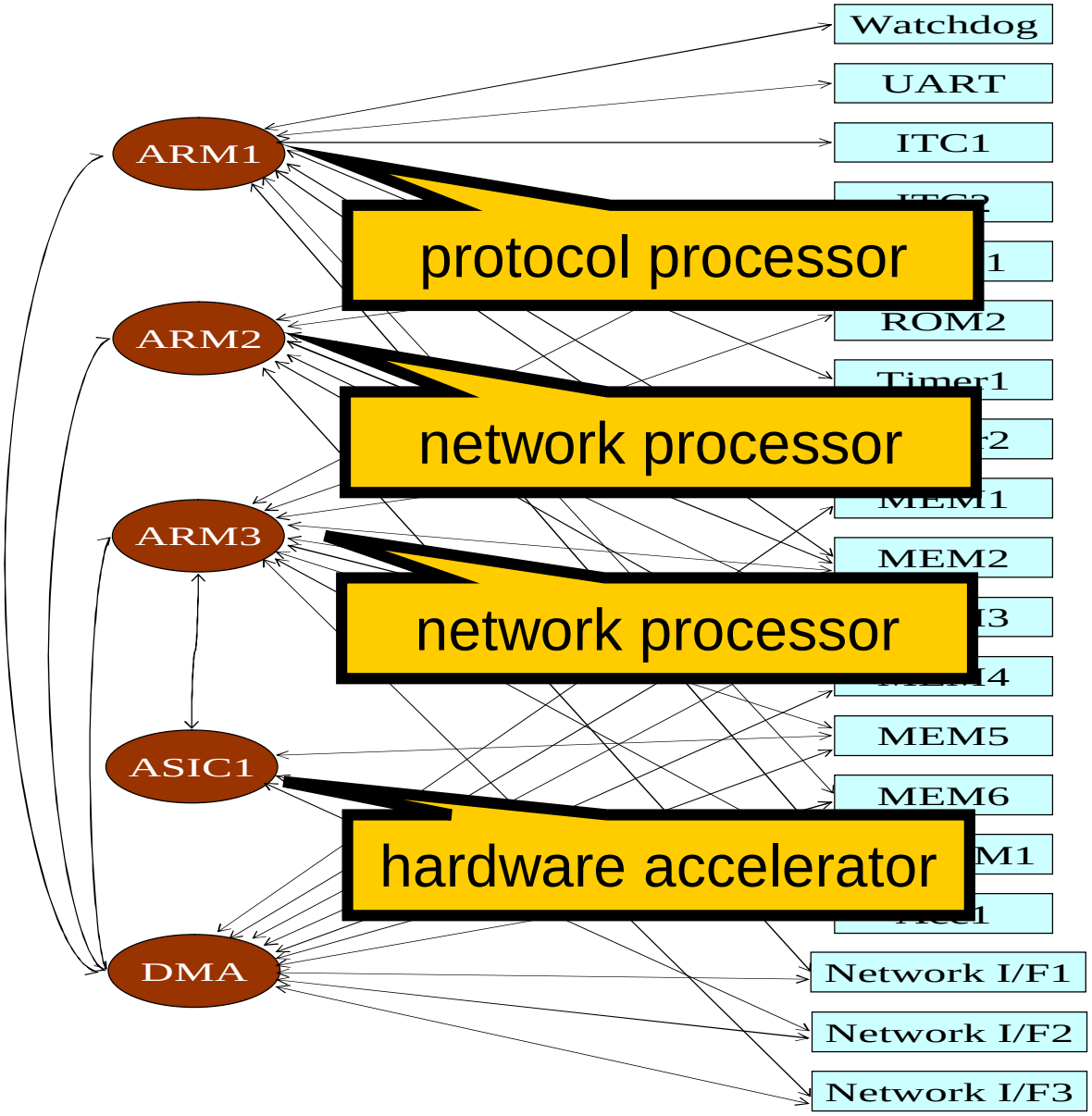
Case Studies

- To evaluate effectiveness of our synthesis approach, we applied it to 4 MPSoC applications from networking domain
  - VIPER, SIRIUS – variants of existing industrial strength applications
  - ORION4, HNET8 – larger systems derived from next-gen MPSoCs

Number of cores in MPSoC applications

Applications	Processors	Masters	Slaves
VIPER	2	4	15
SIRIUS	3	5	19
ORION4	4	8	24
HNET8	8	13	29

SIRIUS MPSoC



SIRIUS MPSoC

Throughput Constraint Paths (TCPs)

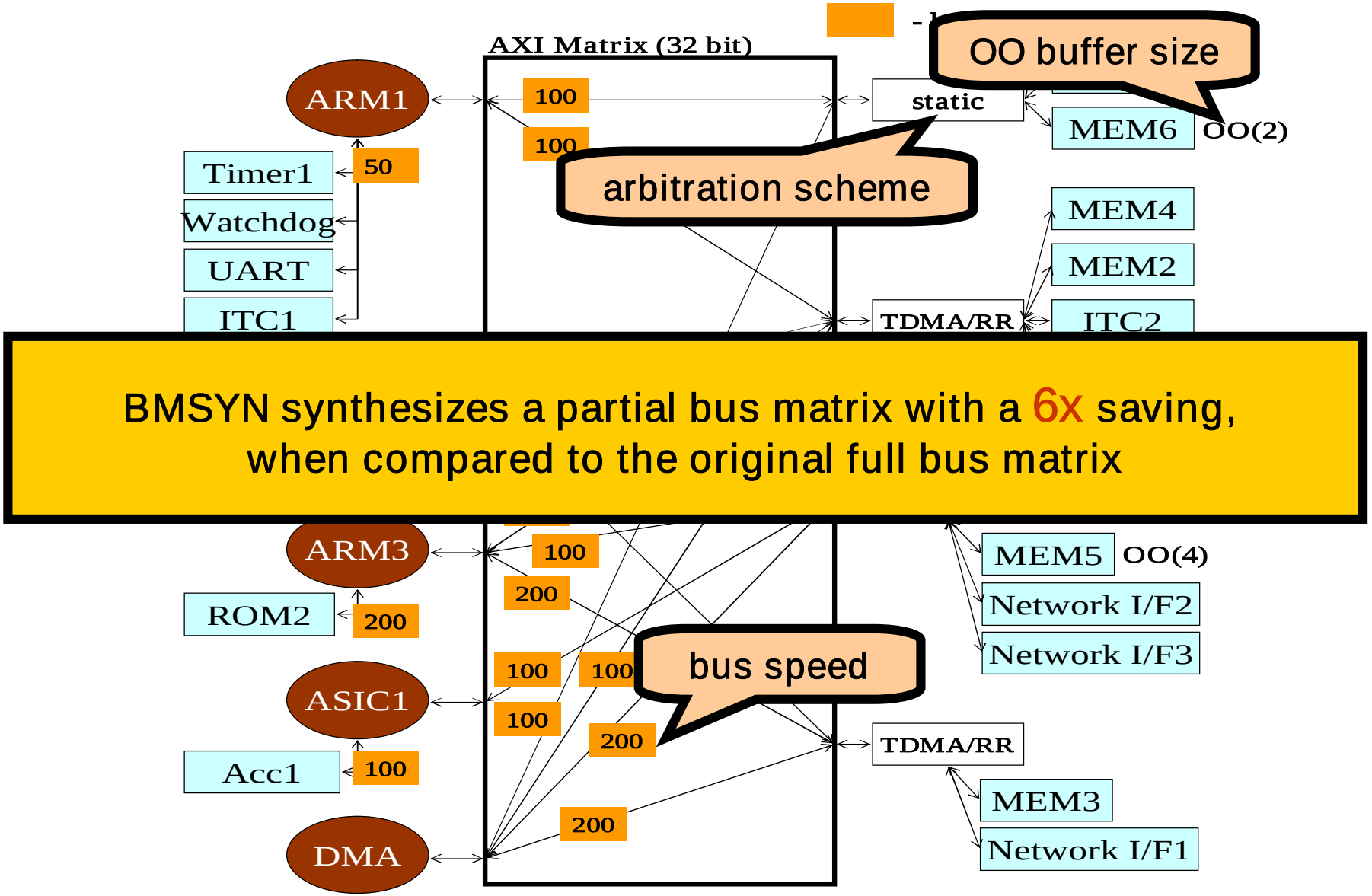
IP cores in Throughput Constraint Path (TCP)	Throughput Requirement
ARM1, MEM1, DMA, SDRAM1	640 Mbps
ARM1, MEM2, MEM6, DMA, Network I/F2	480 Mbps
ARM2, Network I/F1, MEM3	5.2 Gbps
ARM2, MEM4, DMA, Network I/F3	1.4 Gbps
ASIC1, ARM3, SDRAM1, Acc1, MEM5, Network I/F2	240 Mbps
ARM3, DMA , Network I/F3, MEM5	2.8 Gbps

Communication Parameter Constraint Set

Set	Values
bus speed	25, 50, 100, 200, 300, 400
arbitration strategy	static, RR, TDMA/RR
OO buffer size	1 – 8

Target bus matrix architecture: AMBA3 AXI bus matrix

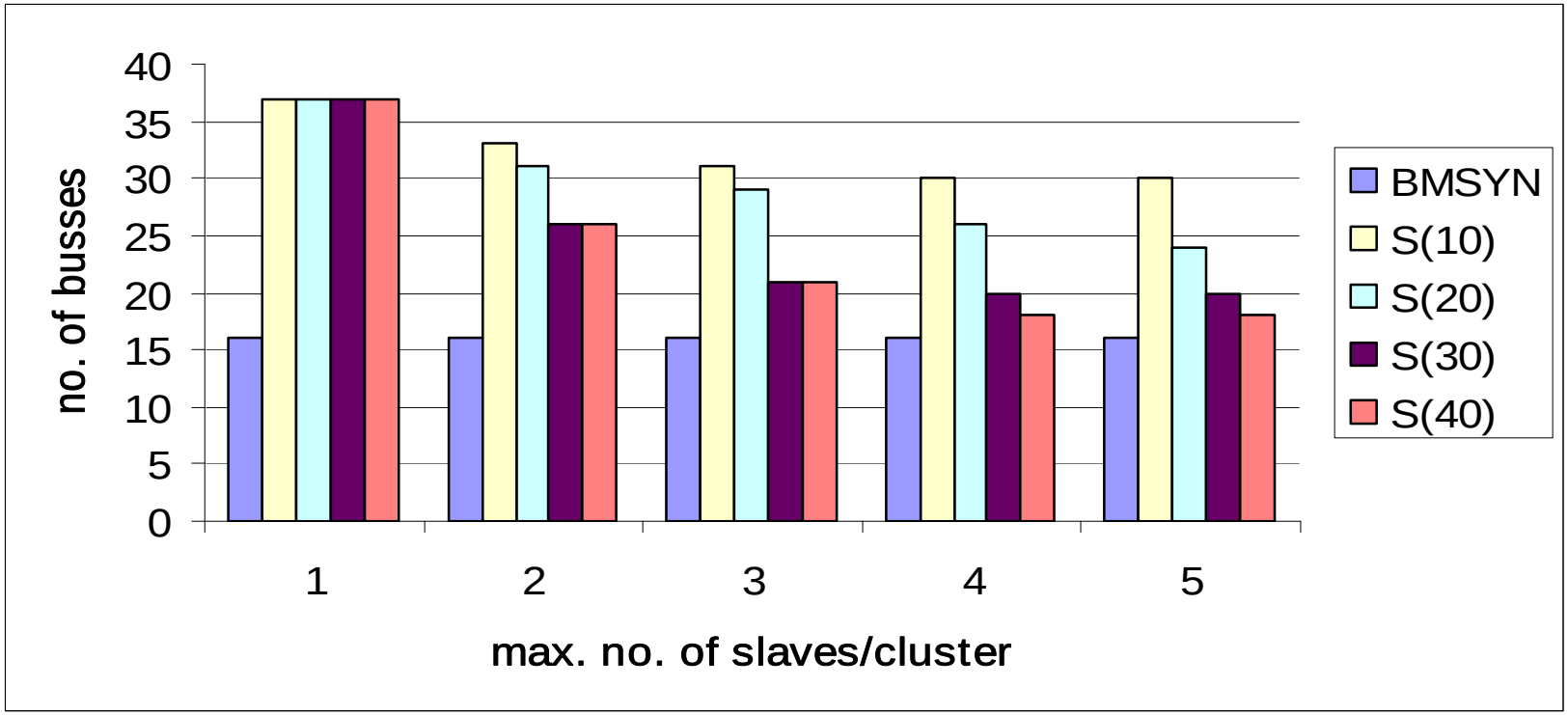
SIRIUS Synthesized Output



Comparison with Related Work

- To compare quality of our synthesis results, we chose the closest existing piece of work, by Murali et al. [DATE 2005]
  - deals with automated matrix synthesis with the aim of minimizing number of busses
- Since their approach only generates matrix topology, we restricted our comparison to the number of busses in the final synthesized design
  - Our approach generates both matrix topology and parameter values
- Their “threshold-based” approach requires the designer to statically specify
  - maximum number of slaves per cluster
  - traffic overlap threshold
    - **which if exceeded prevents two slaves from being assigned to the same bus cluster**

Comparison with Related Work

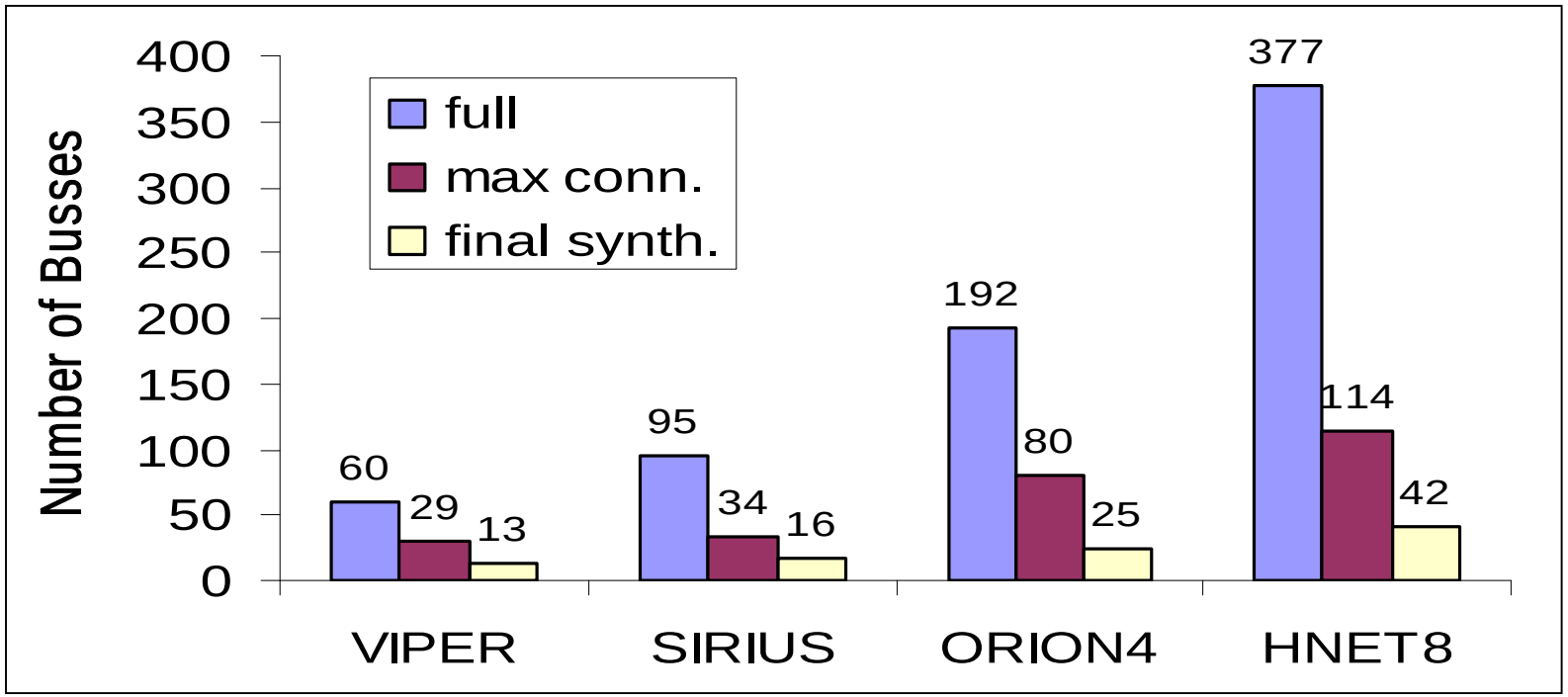


BMSYN produces a lower cost system (having lesser number of busses) than an approach which requires the designer to statically approximate application characteristics

Comparison of Number of Busses

- We compared the number of busses in a
  - the final synthesized bus matrix, with BMSYN

2.1x to 3.2x savings when compared to maximally connected bus matrix
4.6x to 9x savings when compared with full bus matrix



Outline

- Motivation
- Related Work
- Problem Formulation
- Bus Matrix Synthesis (BMSYN) Approach
- Case Studies
- Conclusion

Conclusion

- We presented an approach for the automated synthesis of bus matrix communication architectures (BMSYN)
- BMSYN satisfies all performance constraints and generates
  - topology for bus matrix, having a minimal number of busses
  - values for matrix parameters
    - **bus speeds, OO buffer sizes and arbitration strategies**
- Results from synthesis for 4 industrial strength MPSoC applications show a significant reduction in bus count
  - 9X reduction vs. full bus matrix
  - 3.2X reduction vs. maximally connected reduced matrix
- In the present and near future, bus matrix communication architectures can efficiently support MPSoC systems
  - with tens to hundreds of cores
  - several data throughput constraints in the multiple gigabits/sec range

Thank you!

sudeep@cecs.uci.edu

Branch and Bound Clustering Illustration