Virtual Channel Flow Control across Mesochronous Clock Domains

IEEE International Conference on Modern Circuits & Systems Technologies - 2022

Giorgos Dimitrakopoulos, Anastasios Psarras
Electrical and Computer Engineering
Democritus University of Thrace, Greece

Chrysostomos Nicopoulos
Electrical and Computer Engineering
University of Cyprus, Cyprus

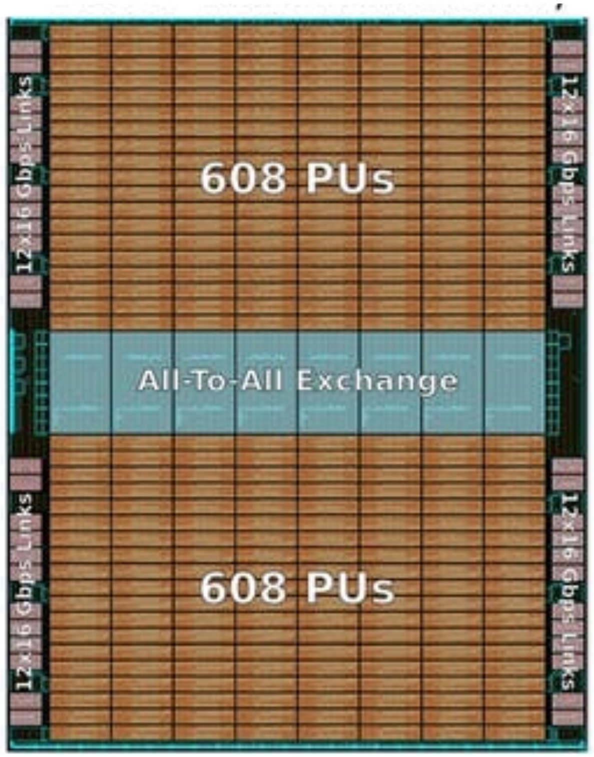
Era of extreme integration

Multi-Cores are found on **all kinds of devices** & **sizes**

(Mobile, Automotive, Data Center, Wearables, IoT sensors, Desktop, …)

CMPs *(Chip Multi-Processors)*

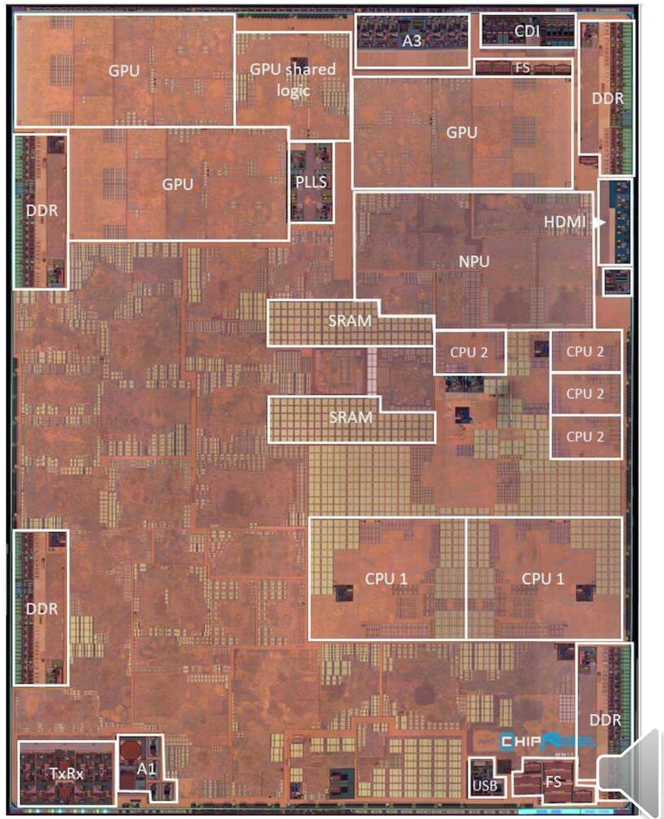
- Identical cores (CPU+L1+L2/Scratchpad)
- Homogeneous, regular, Tiled
- Driven by ML applications



[GraphCore IPU]

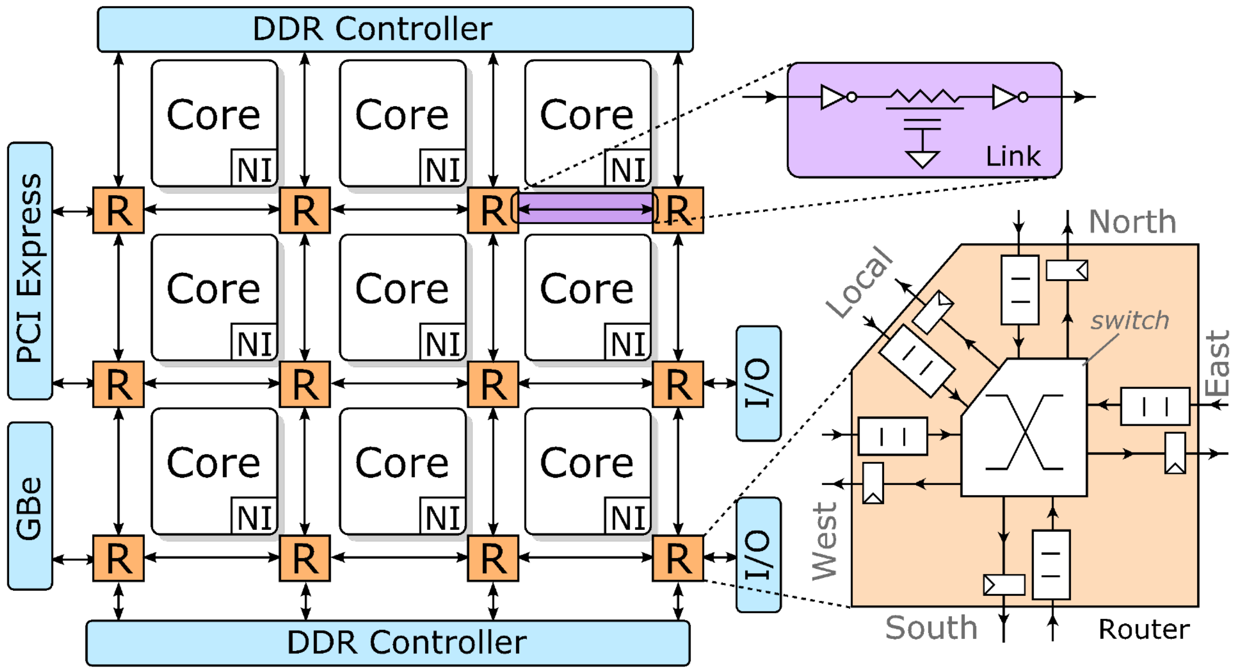
MPSoCs *(Multi-Processor System-on-Chips)*

- Diverse IP cores
- Heterogeneous, irregular



[Apple A11]

Network-on-Chip: established communication medium

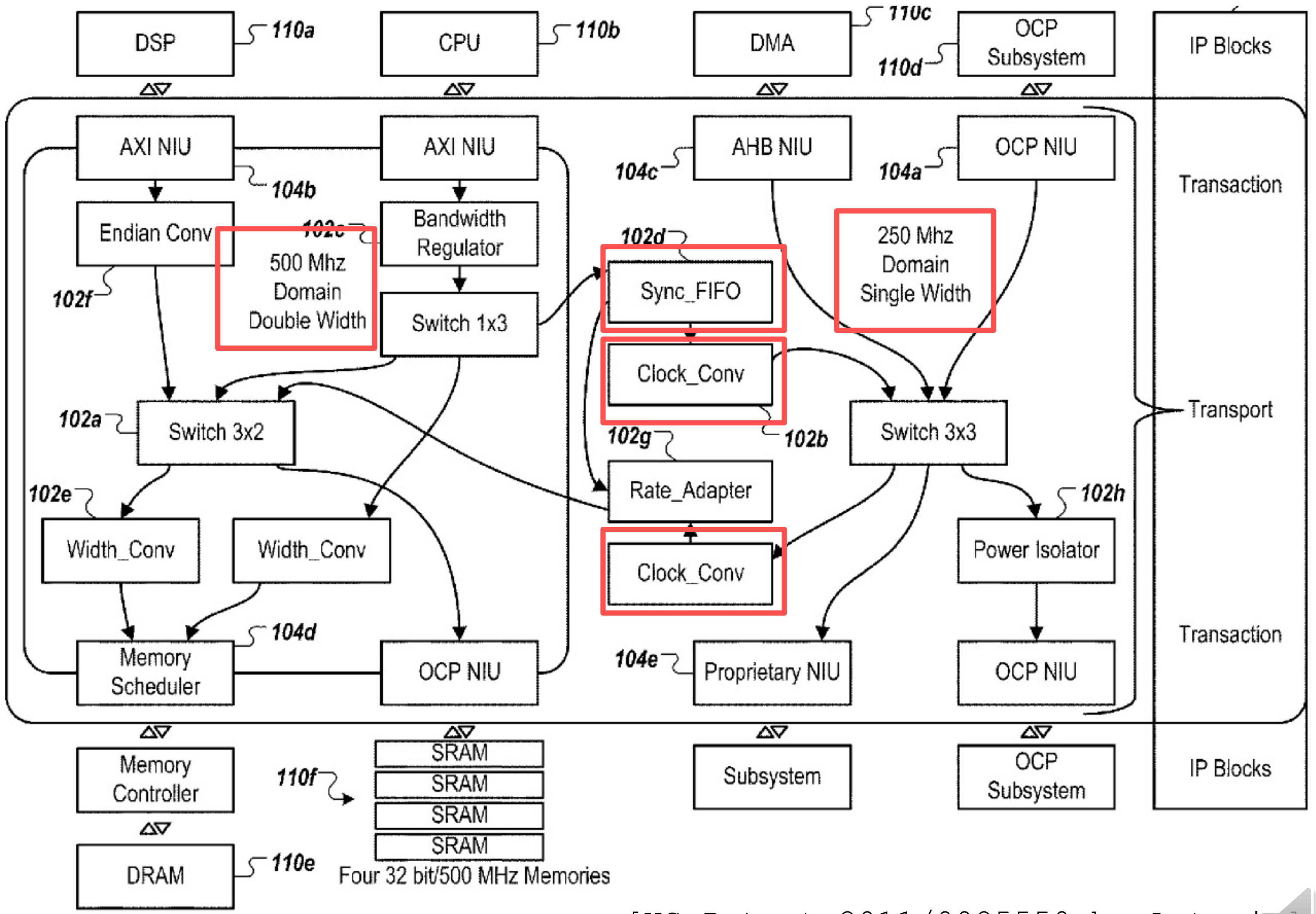


Messages generated by Cores/Memory/IPs…

- …are converted to **packets** & injected to the NoC (through **Network Interfaces – NI**)
- …contending with other packets (in **Routers**)
- …traversing physical distances (in **Links**)

NoCs follow a distributed architecture by construction

- NoC connects all kinds of IPs…
- …**spread** across the **whole chip**…
- …operating under all possible **clock frequencies**
- Mainstream approach:
- **Synchronous NoC**
- **CDC points at the NIs**

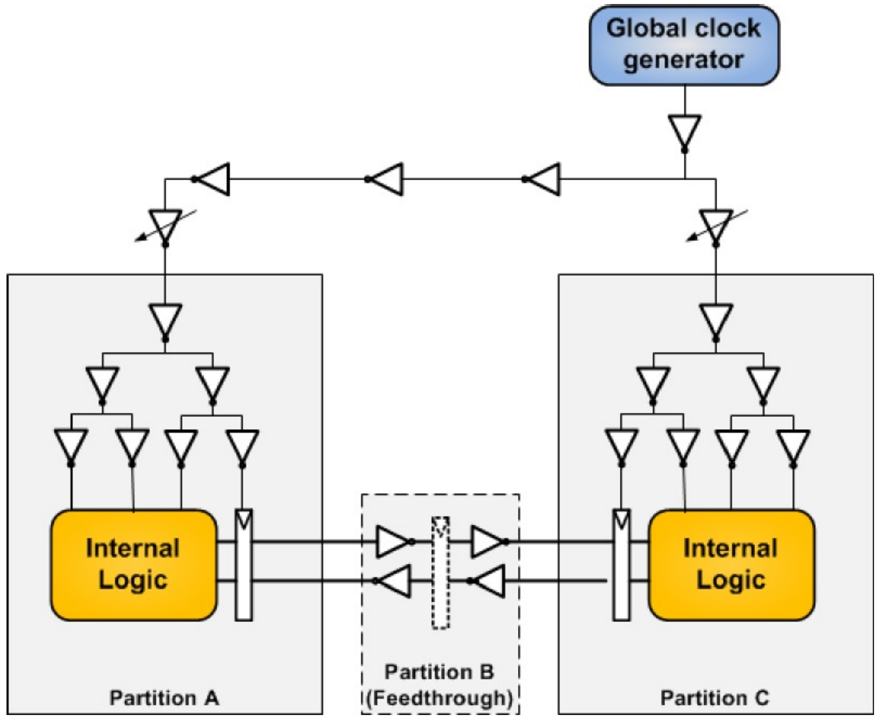


[US Patent 2011/0085550 by Arteris]

NoC Clocking Challenges

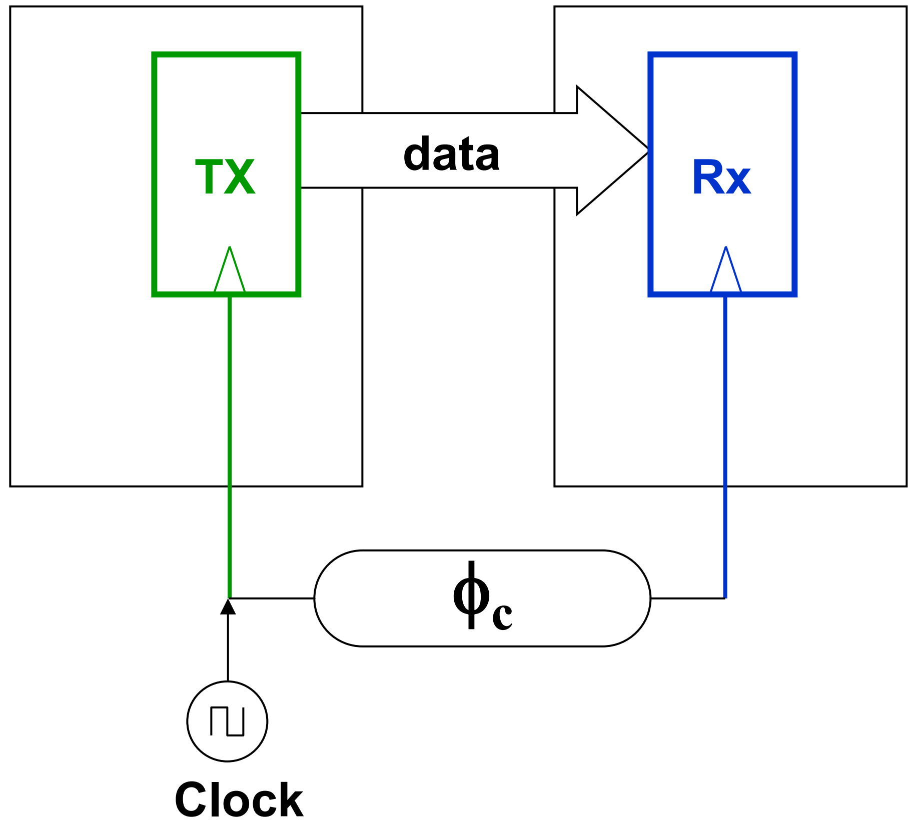
Fully-Synchronous NoC approach not scaling

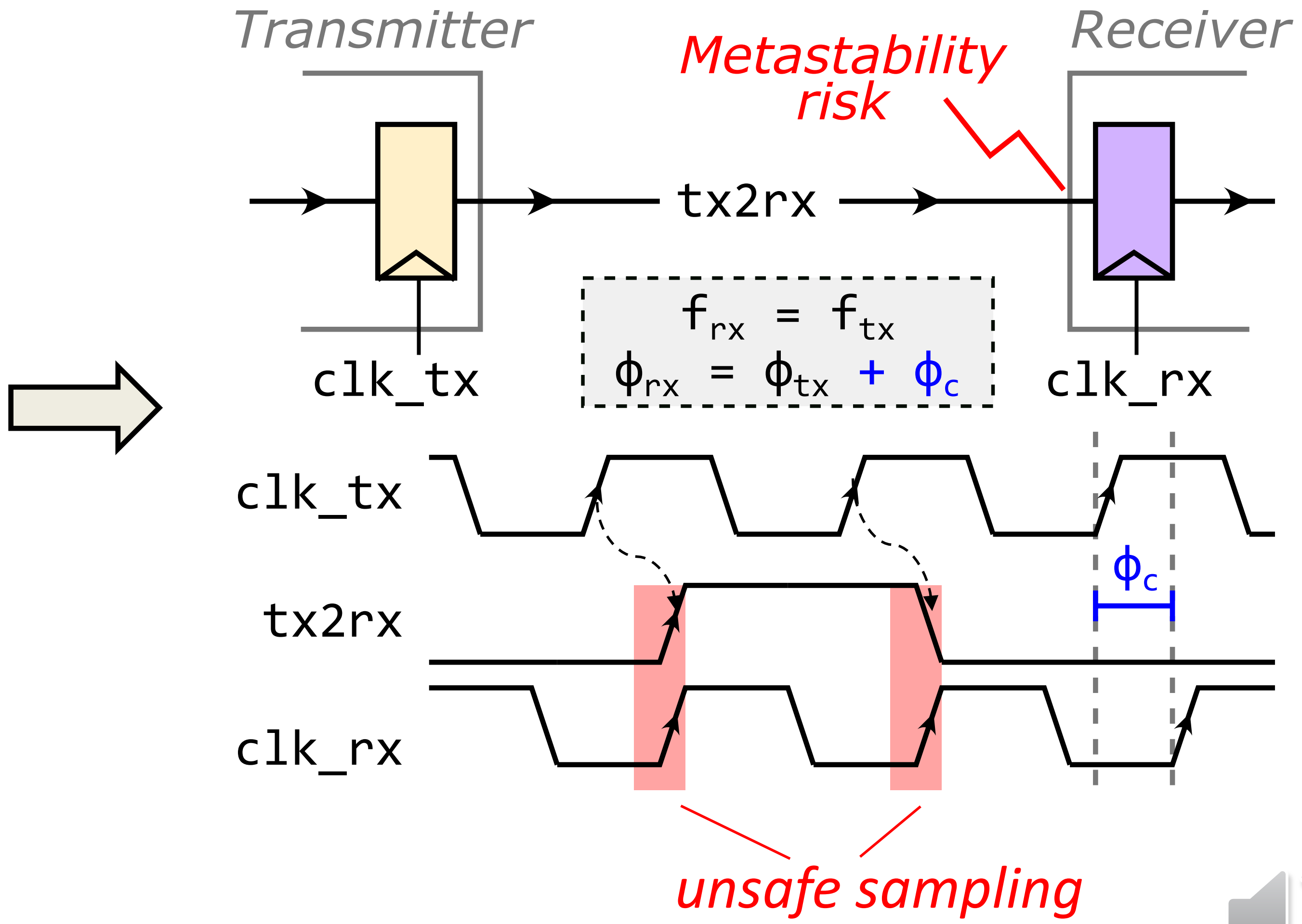
- **Clock Distribution – Clock Tree Synthesis**
  - Global timing closure becomes challenging as technology scales
  - Clock must reach every chip corner overcoming PVT variations
- **Clock frequency may vary within the NoC**
  - not just at its borders
- **Regular / Tiled chips can follow a mesochronous clocking discipline to simplify CTS and timing closure**



Mesochronous Clocking

Clock net with large *constant* skew

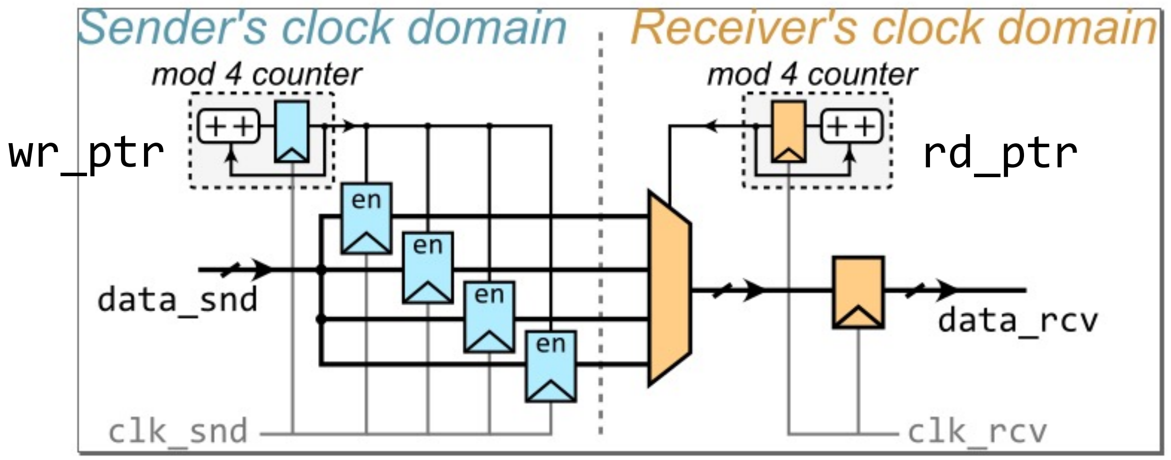


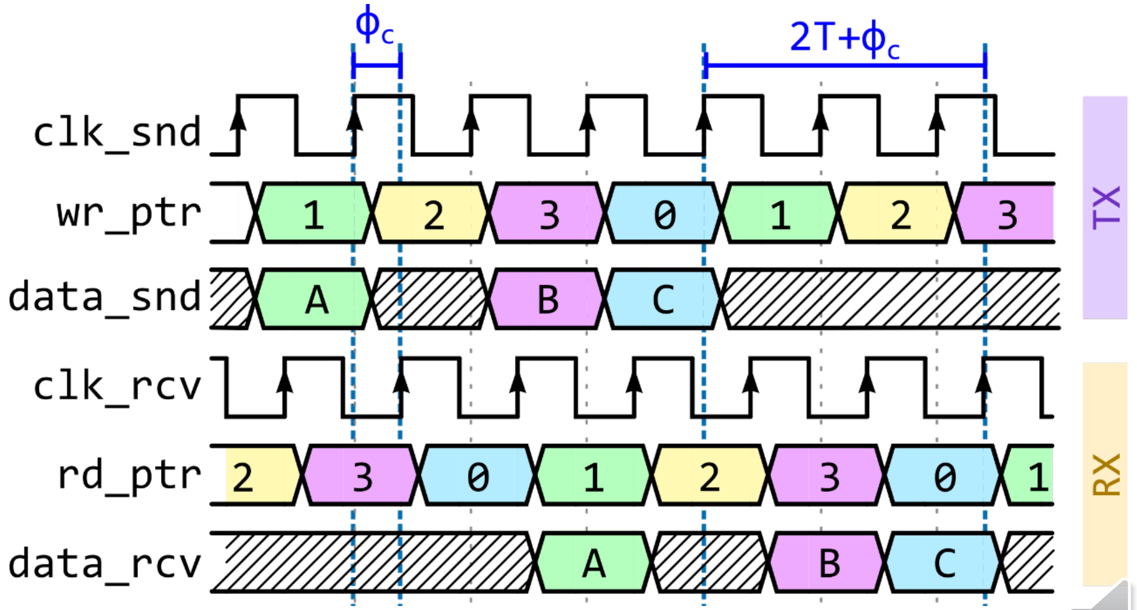


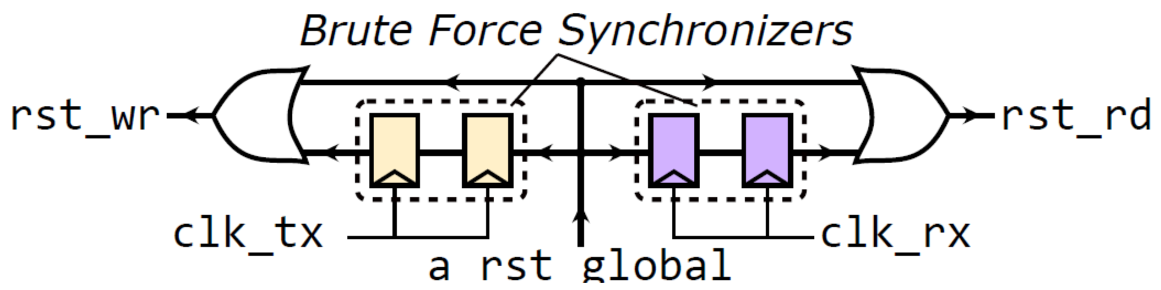
Mesochronous Synchronizer

- FIFO moves synchronization elsewhere
- Sample data in using *clk_tx*
- Mux data out using *clk_rx*
- Constant latency determined at reset
- High throughput

- How do we initialize the pointers?
  - Use brute force reset synchronizer

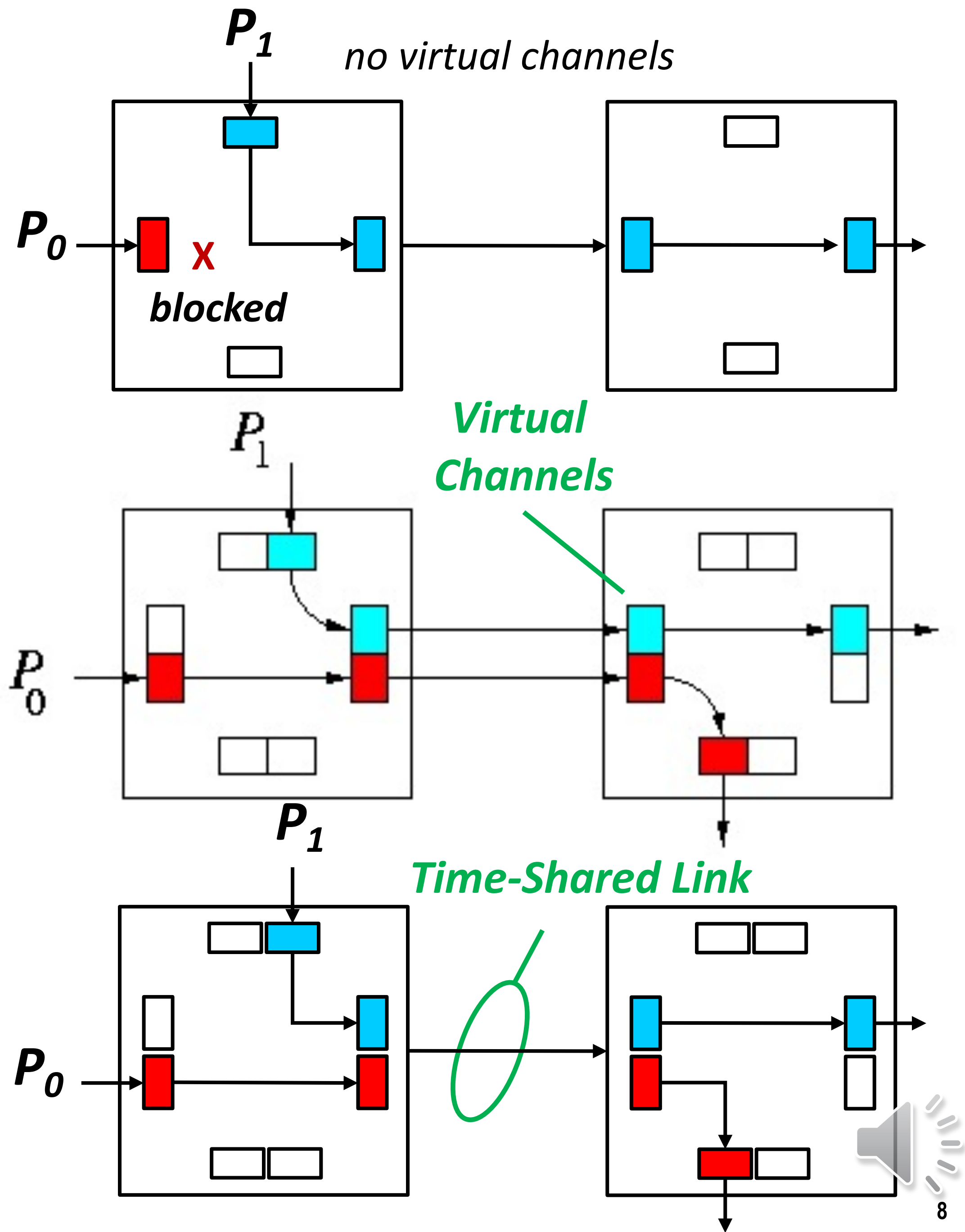






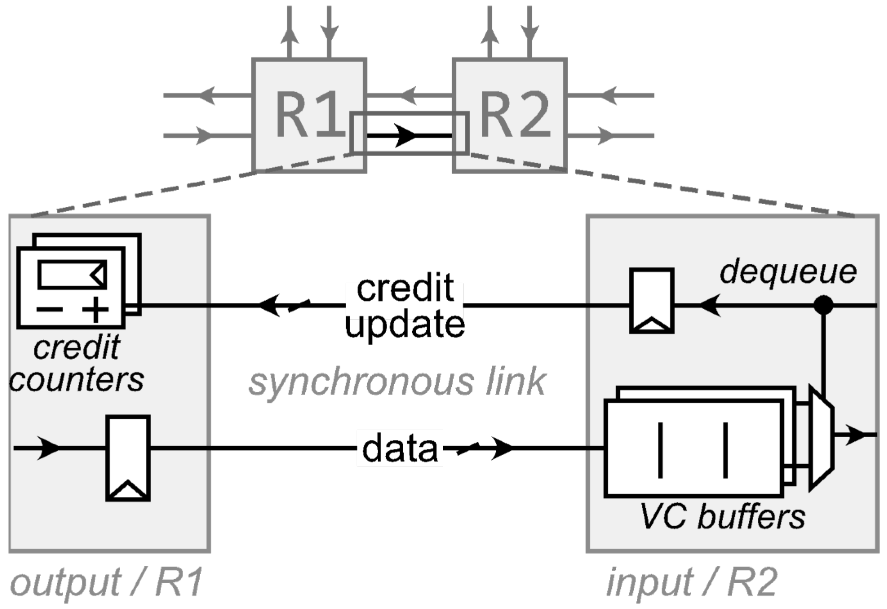
Virtual Channels

- **Share channel capacity between multiple data streams**
  - Interleave flits from different packets
- **Provide dedicated buffer space for each virtual channel**
  - Decouple channels from buffers
- **"The Swiss Army Knife for Interconnection Networks"**
  - Prevent deadlocks
  - Reduce head-of-line blocking
  - Also useful for providing Quality-of-Service

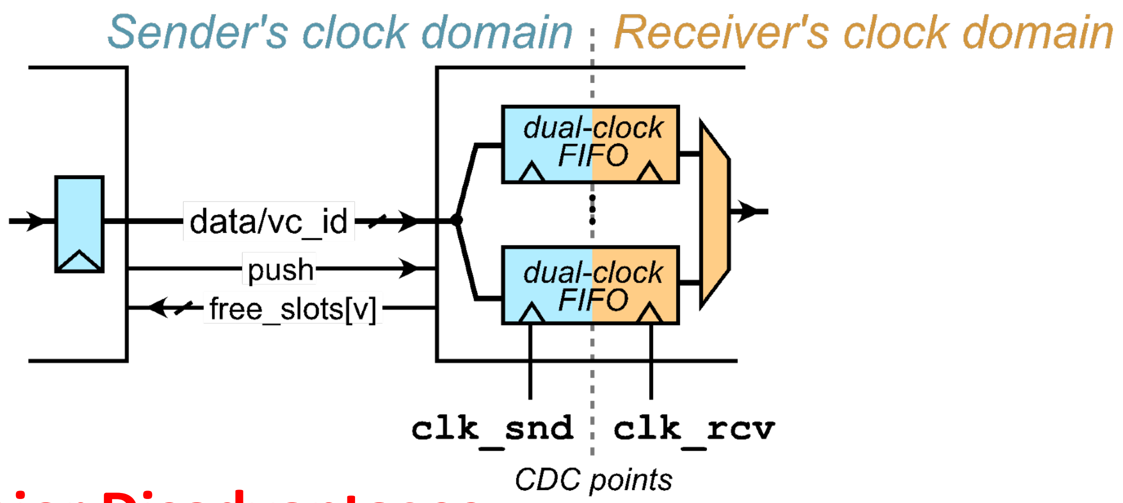


Synchronous Virtual-Channel flow-controlled link

- Sender maintains credit counters whose values correspond to the free slots of downstream Virtual Channel buffers
- Forward Flow Control
  - A flit can be transmitted for VC #v only if credit_count[v]>0
  - When a flit is transmitted: credit_count[v]--
- Backward Flow control
  - Receiver frees up a slot on VC #v: a credit update and the VC ID is transmitted to the sender
  - When the sender receives a credit update for VC #v: credit_count[v]++
- **Sender always in sync** with downstream buffer availability
- How to transform a synchronous multi-VC credit-based link to a **mesochronous** one?



Mesochronous VC flow-controlled link: The tightly-coupled approach



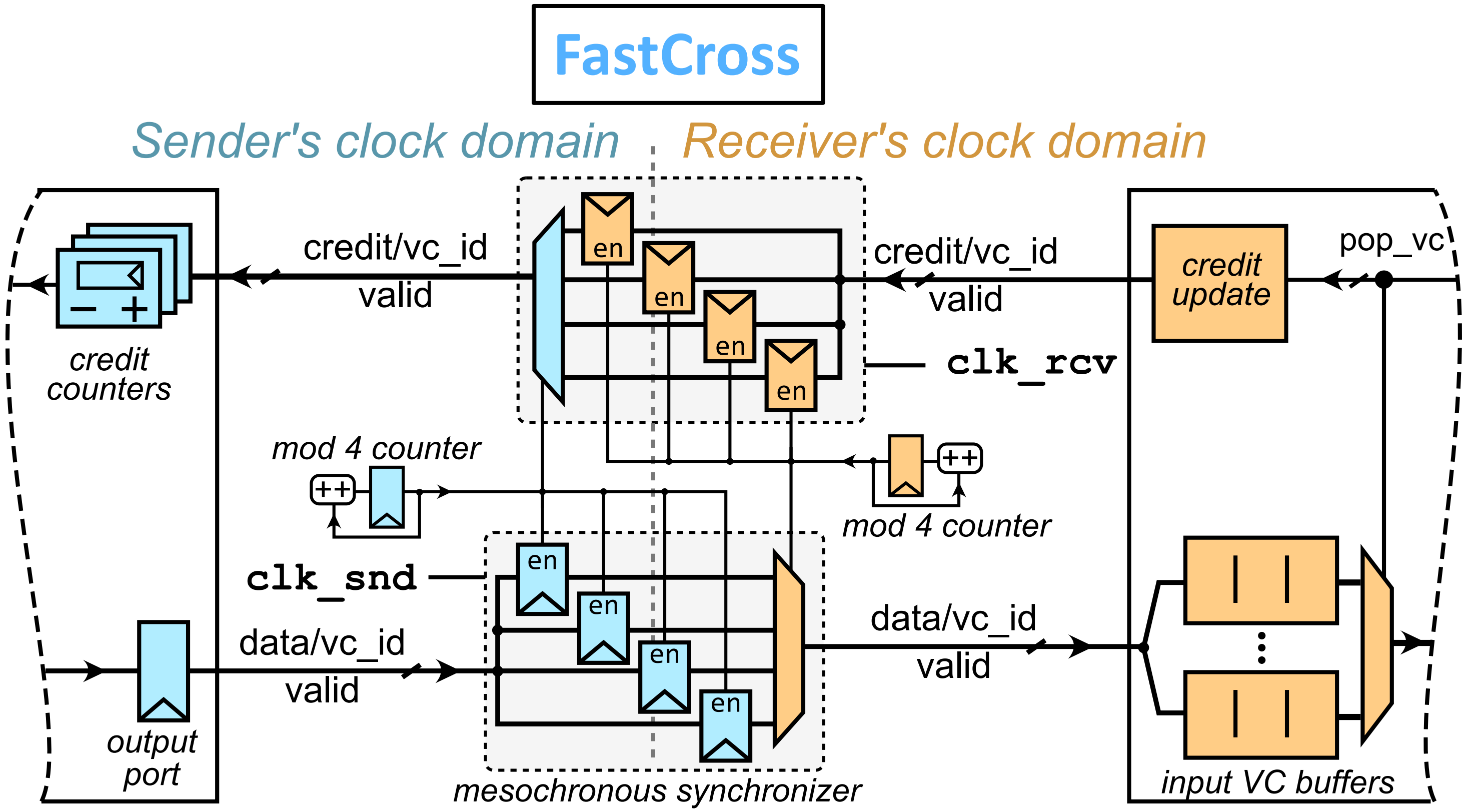
- VC buffers implemented as V× **parallel mesochronous dual-clock FIFOs**
- Backward flow-control altered to fit the dual-clock FIFO implementation
- **V× Clock Domain Crossing Points**
- Possible, but **inefficient** & **impractical**

Major Disadvantages

- **Increased verification effort**
  - Multiple CDC points considered bad practice
- **No buffer sharing** across VCs
  - Sharing is highly desirable since it minimizes total buffering
- **Increased buffering**
  - FIFO synchronizer buffering must be paid per VC due to increased RTT for full throughput

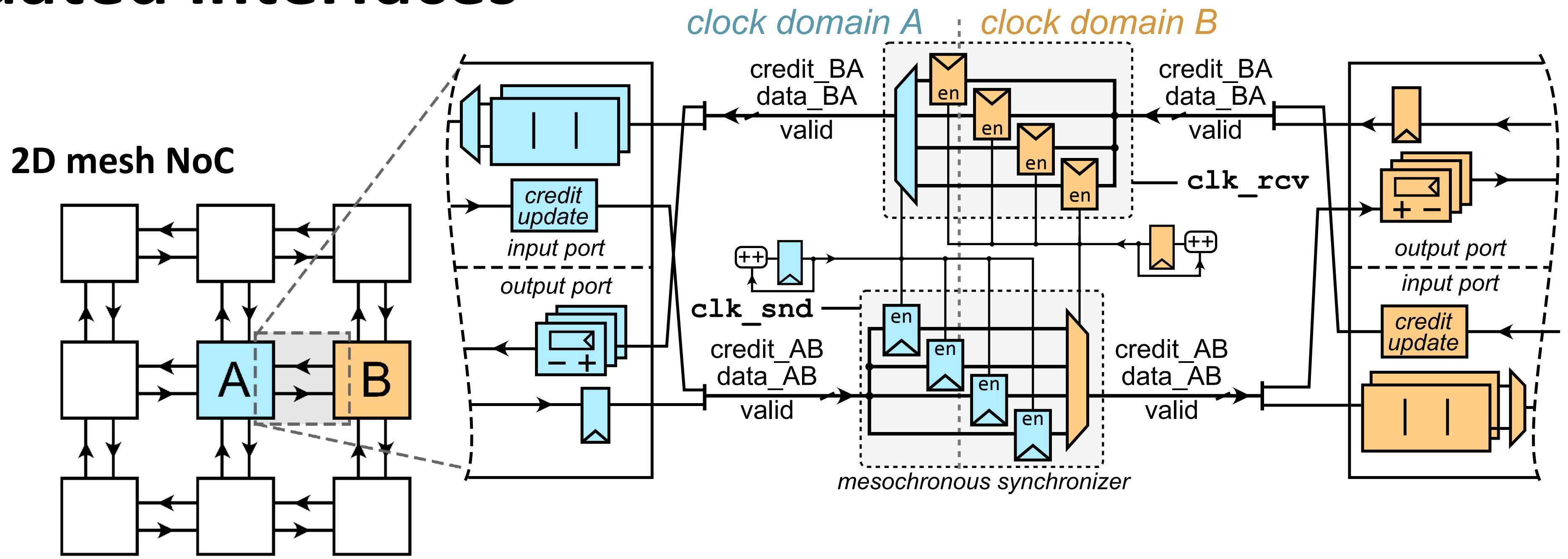
Mesochronous VC flow-controlled link: The loosely-coupled approach

- **2 mesochronous synchronizers** pass signals on both directions
  - Forward path
    data (flit, VC ID, head/tail etc.)
  - Backward path
    credit updates & credited VC IDs
  - One CDC point per signal direction
  - **Synchronizers share counters!**



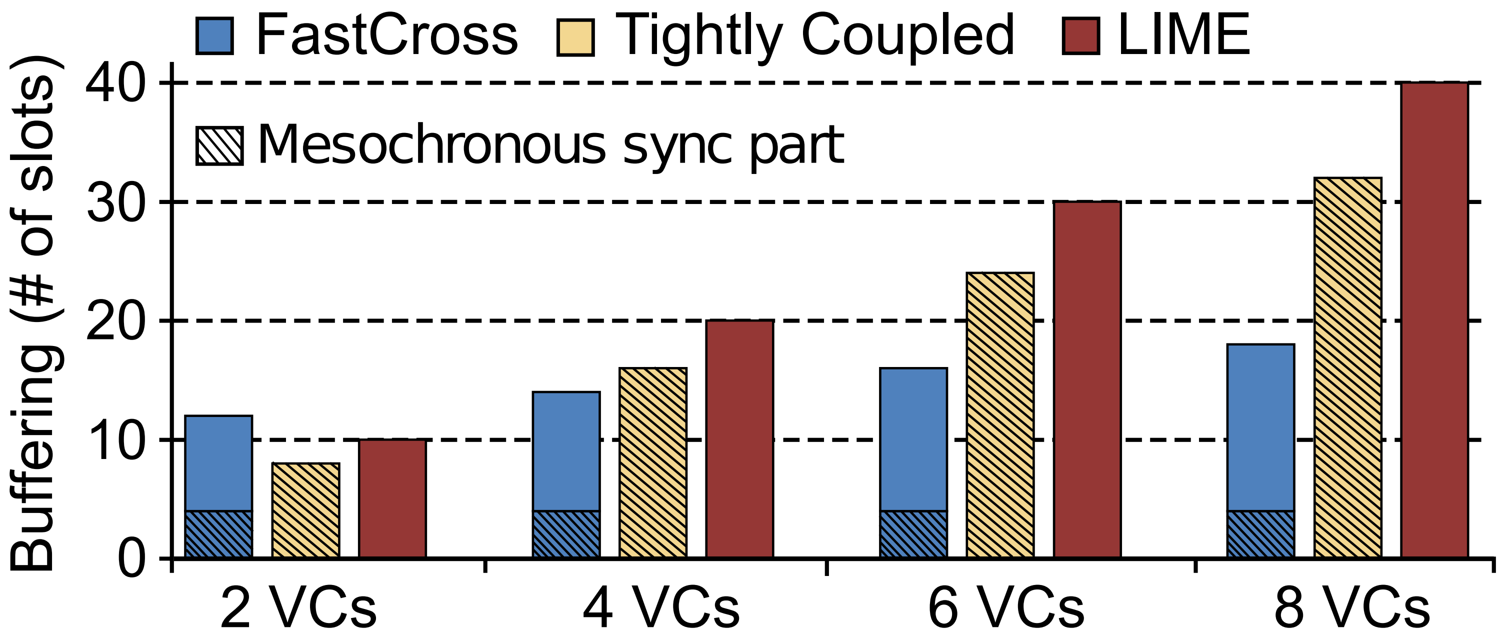
- Forward flow control similar to synchronous case
- Data and credits arrive after paying the necessary synchronization latency

Consolidated interfaces



- **Bidirectional links** are very common in NoC topologies (meshes, rings, trees etc.)
- **One synchronization point** for each domain's incoming signals is desirable
  - Maximize design safety and minimize verification effort
- **Consolidated Interfaces: merge synchronizers** that synchronize **same-direction** signals
  - E.g. lower part mesochronous synchronizer syncs data from A's *output* to B's *input* as well as credits from A's *input* to B's *output*

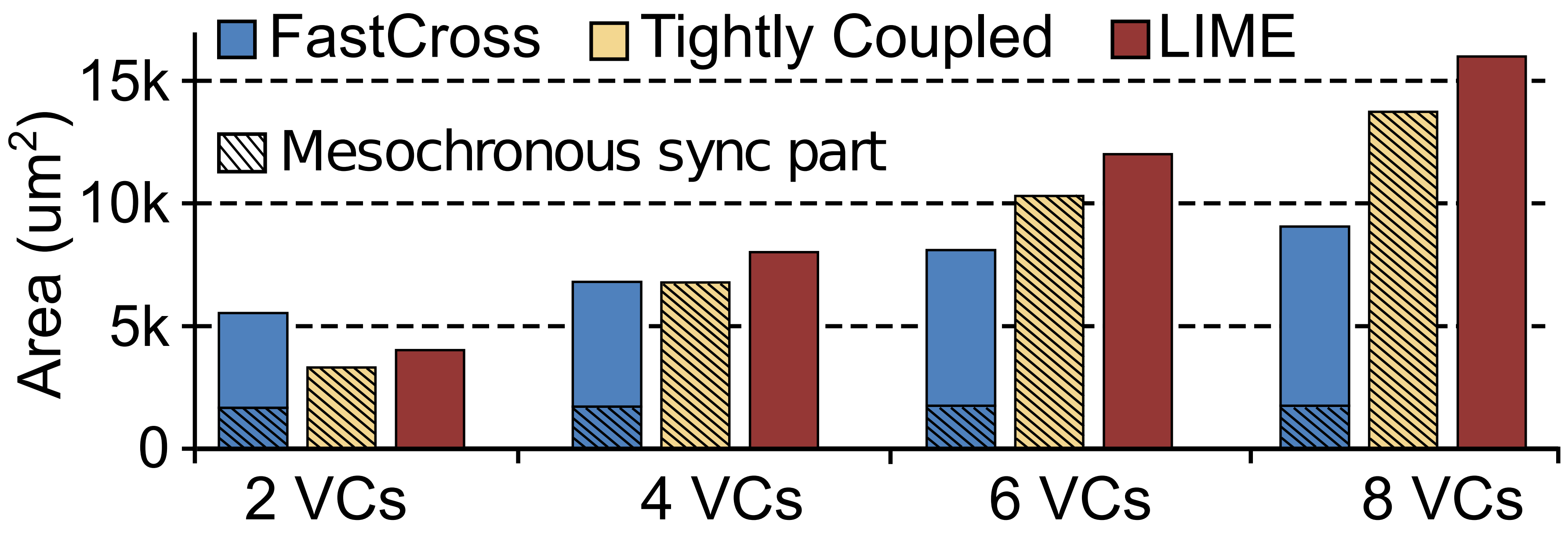
Minimum buffering required for full (100%) throughput



- In all cases FIFOs deep enough to cover Round-Trip Time (RTT)
  - RTT in FastCross is increased due to the loosely coupled mesochronous synchronization
- **FastCross** use VC **buffer sharing to amortize the increased RTT**
  - Tightly coupled approaches (LIME is also tightly coupled) does not

Silicon area comparisons

SystemVerilog RTL
Synthesis, P&R using standard EDA flow
Standard cell library (45nm, 0.8V, 120°C)



- Tightly coupled approaches that integrate input buffering and flow control is effective when the number of employed VCs is small
- FastCross decouples synchronization from the VC flow-control semantics
  - This decoupling enables the use of shared buffering at the receiver that saves hardware area

Conclusions

- As NoCs are turning GALS CDC at the link level become a necessity
  - Mesochronous interfaces a viable alternative for tiled/regular architectures
  - Clock domain crossing should be smoothly combined with virtual-channel flow control
- FastCross mesochronous VC flow-controlled link offers:
  - Credit-based VC flow control under mesochronous clock domains
  - An efficient & low-cost implementation with Design Safety & Verification Effort in mind
- FastCross leverages a loosely coupled approach that separates synchronization from buffering
  - Sharing of VC buffers amortizes the cost of separate sync + functional buffering