Addressing the System-on-a-Chip Interconnect Woes Through Communication-Based Design

J. Rabaey,

M. Sgroi, M. Sheets, A. Mihal, K. Keutzer, S. Malik, J. Rabaey, A. Sangiovanni-Vincentelli

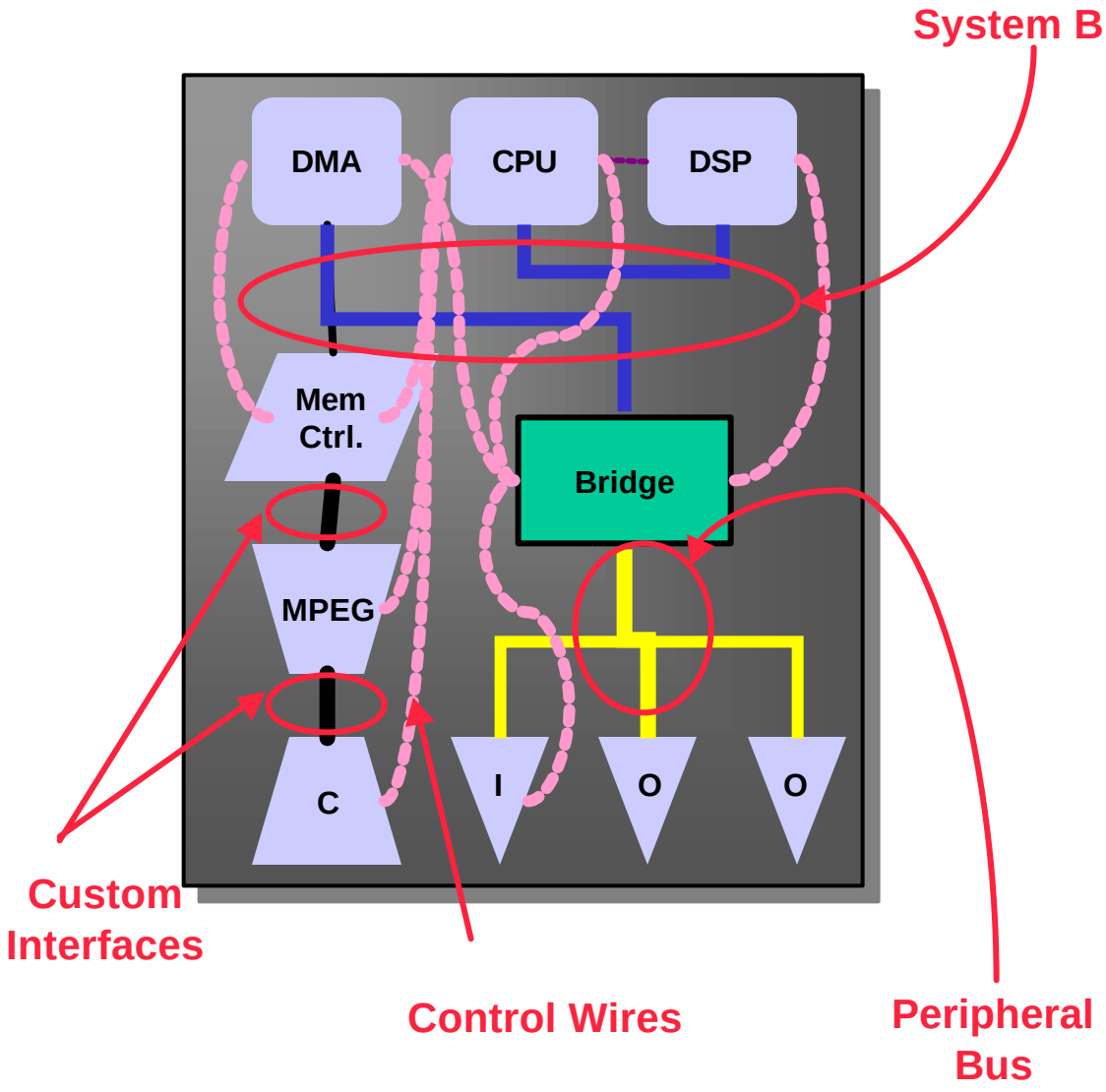
University of California, Berkeley and Princeton University

The SOC Interconnect Challenge

The SOC Interconnect Challenge

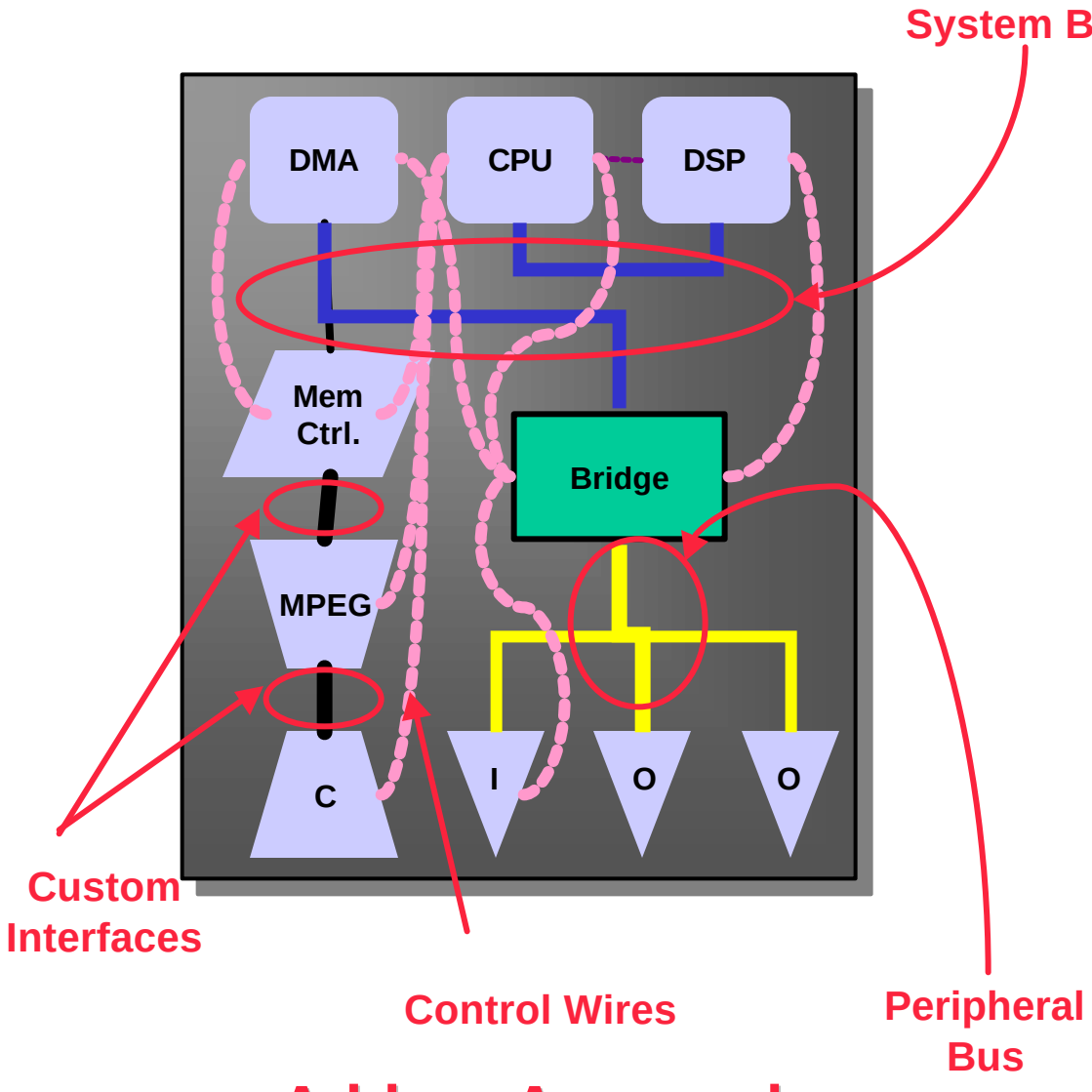
“Femme se coiffant”
Pablo Ruiz Picasso
1940

The SOC Interconnect Challenge



Ad-hoc Approach

The SOC Interconnect Challenge



Ad-hoc Approach

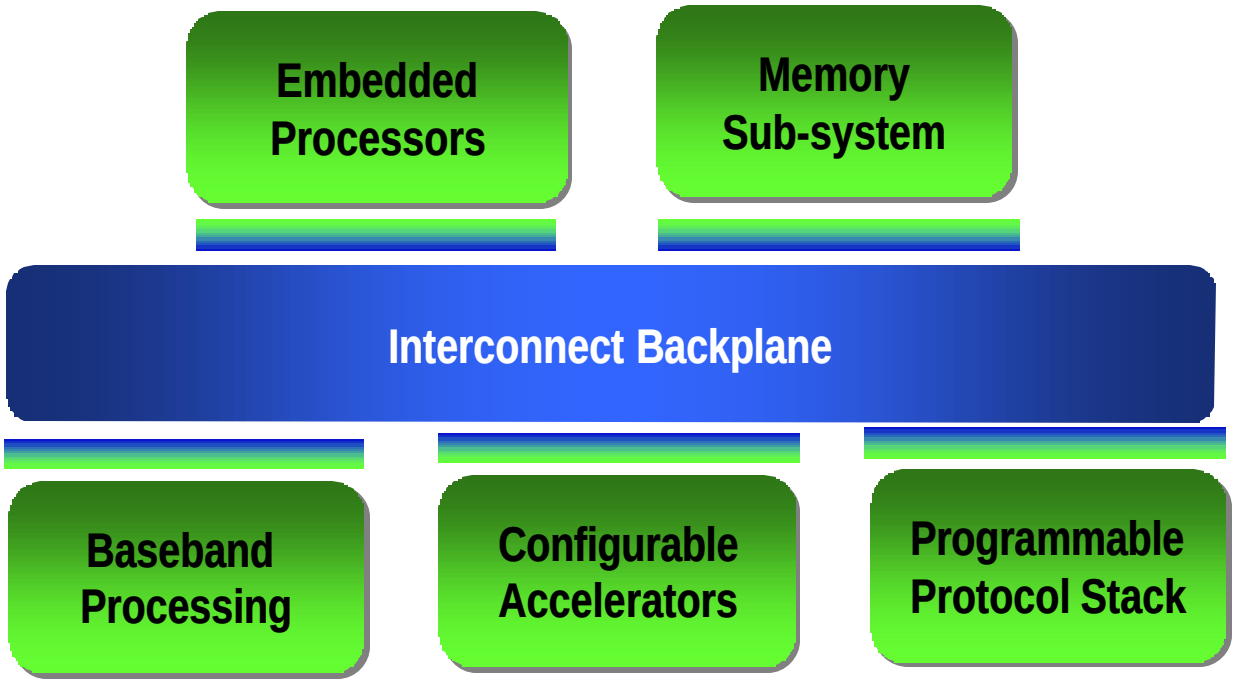
Alternative:

A *disciplined SOC interconnect design approach* that addresses:

- reliability
- predictability
- performance
- power dissipation

concerns caused by deep-submicron effects and complexity considerations, and exploits *advanced communication techniques*

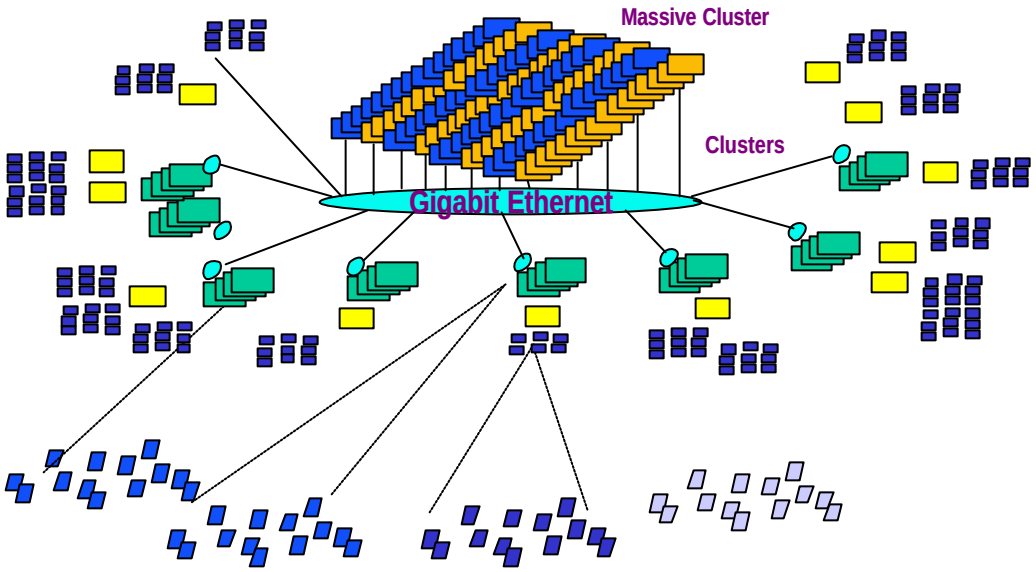
The Network-on-a-Chip (NOC) Approach



Communication-based Design

- Orthogonalizes function and communication
- Builds on well-known models-of-computation and correct-by-construction synthesis flow
- Parallels layered approach exploited by communications community

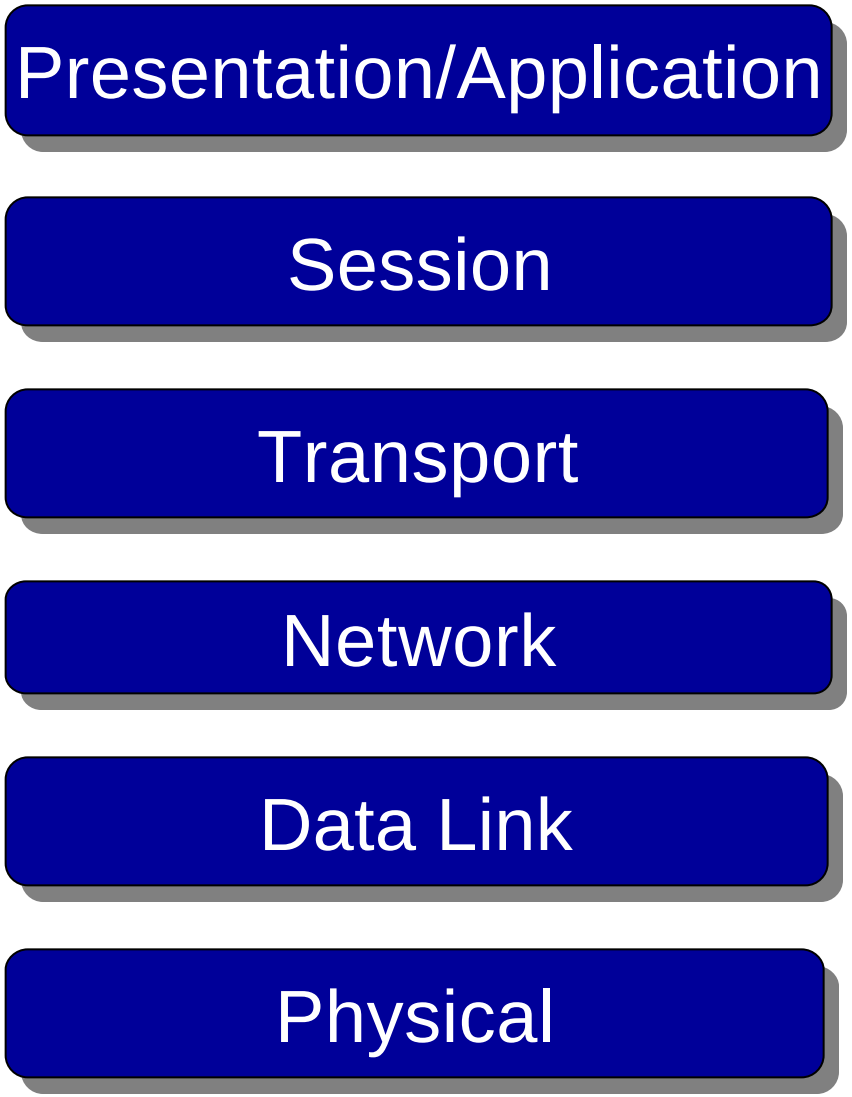
How Does the Communication Network World Deal with these Problems?



- Scalable clusters of heterogeneous networks
- Wide range of data units at different levels of abstraction (streams, packets, bits)
- With varying throughput, latency and reliability requirements

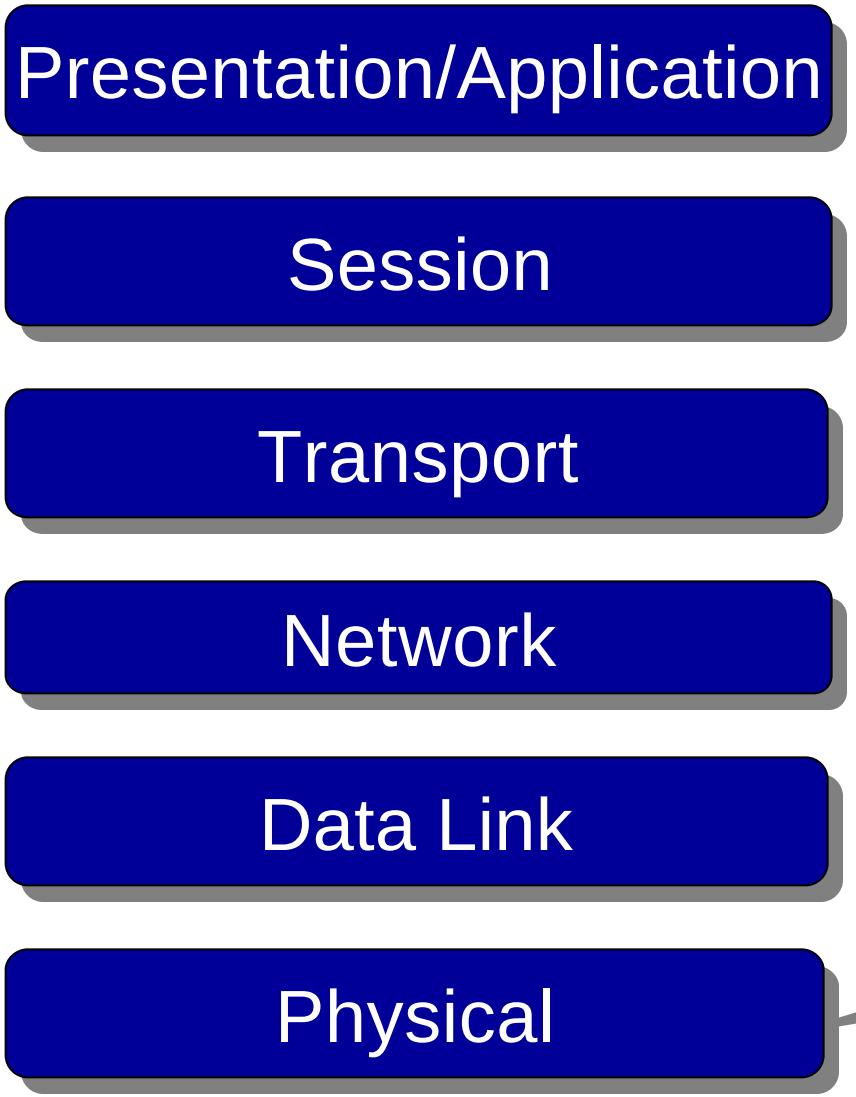
Central tenet: *Layered approach* standardized as the ISO-OSI Reference Model.

The ISO Protocol Stack



- **Reference model for wired and wireless protocol design —Also useful guide for for conception and decomposition of NOCs**
- **Layered approach allows for orthogonalization of concerns and decomposition of constraints**
- **Not required to implement all layers of the stack**
  - depends upon application needs and technology
- **Layered structure must not necessarily be maintained in final implementation**
  - e.g., multiple layers can be merged in implementation optimization

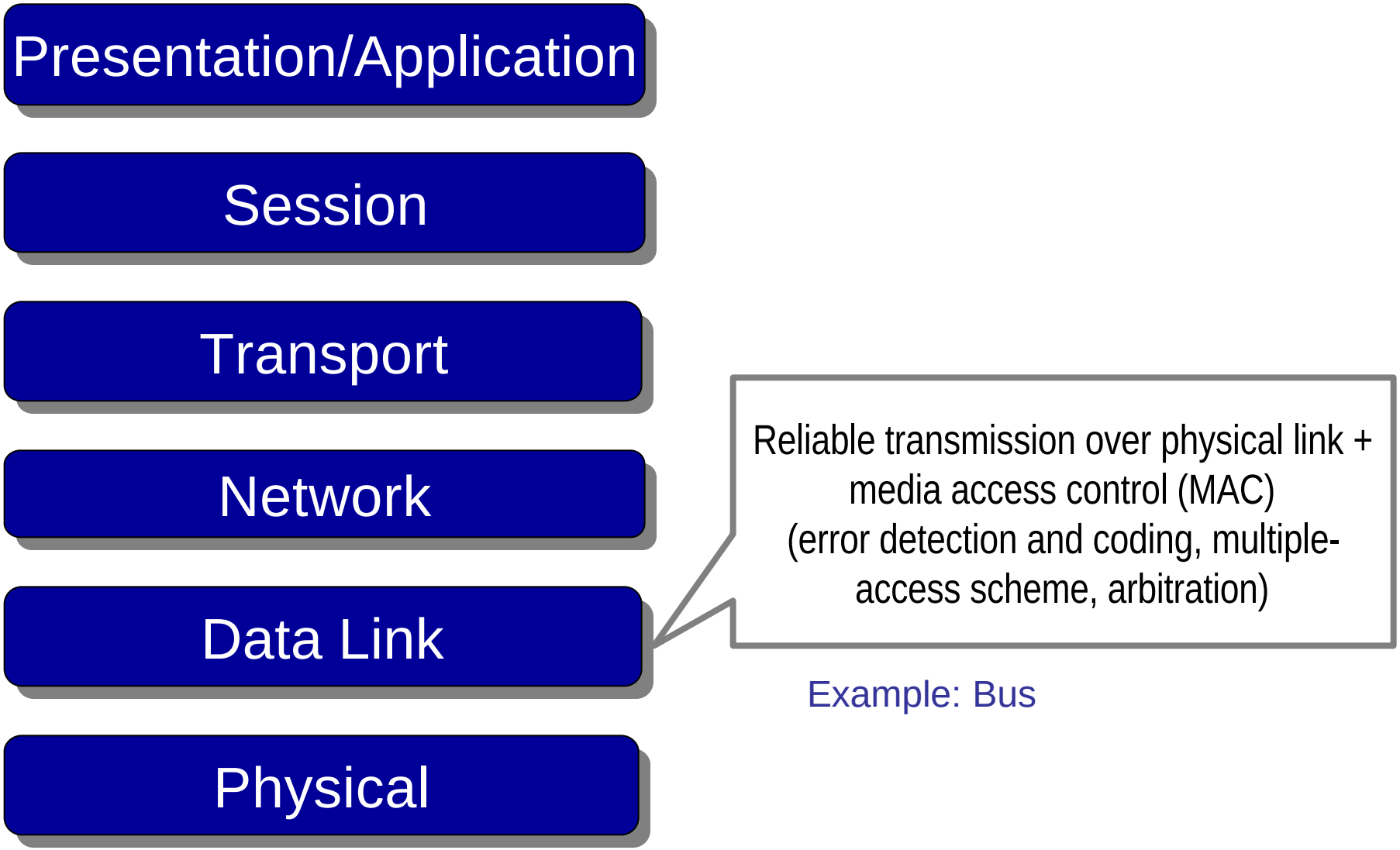
The ISO Protocol Stack



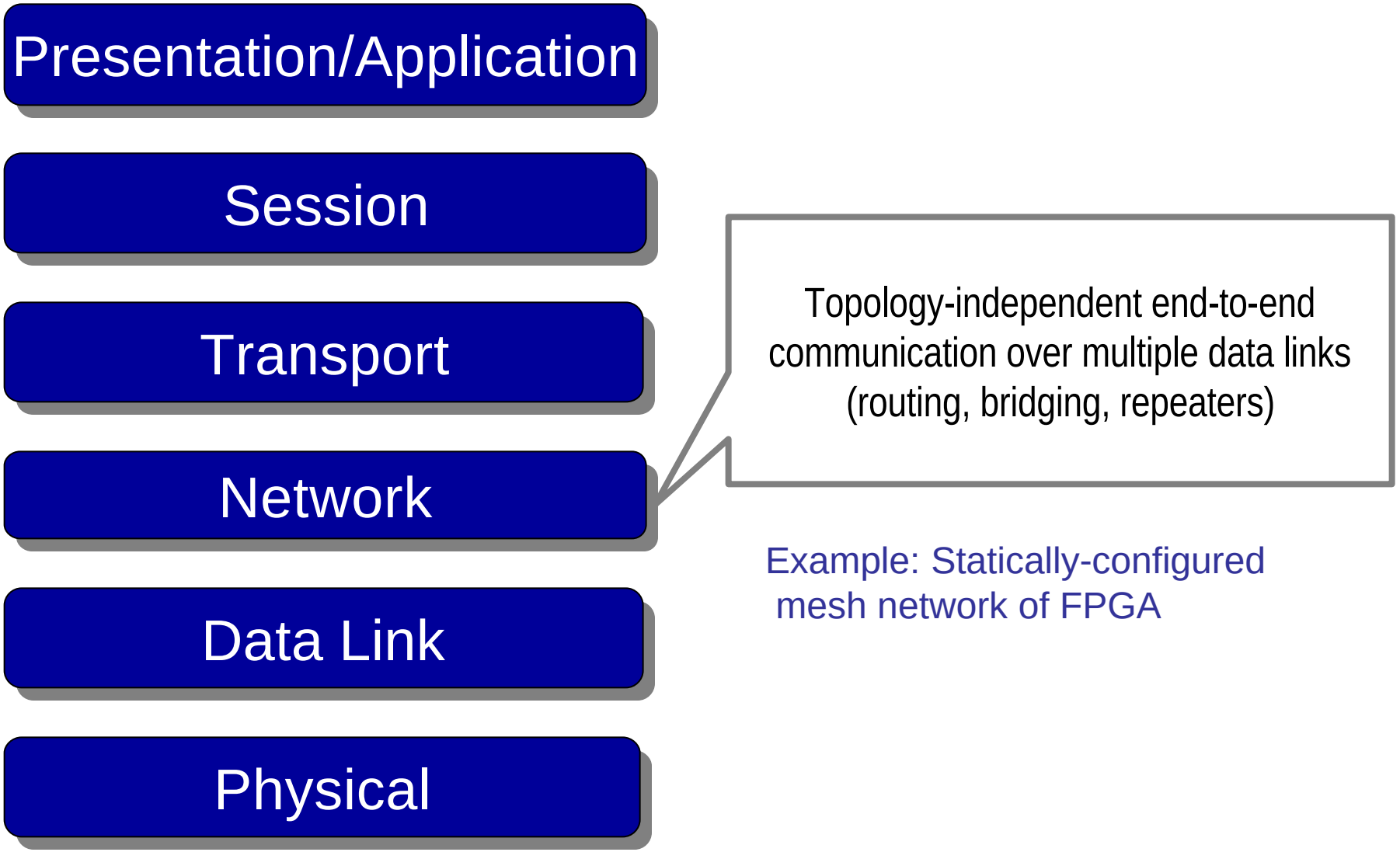
Transmit bits over physical interconnect medium (signal waveform, voltages, timing, synchronization)

Example: synchronous reduced-swing pulse-based signaling

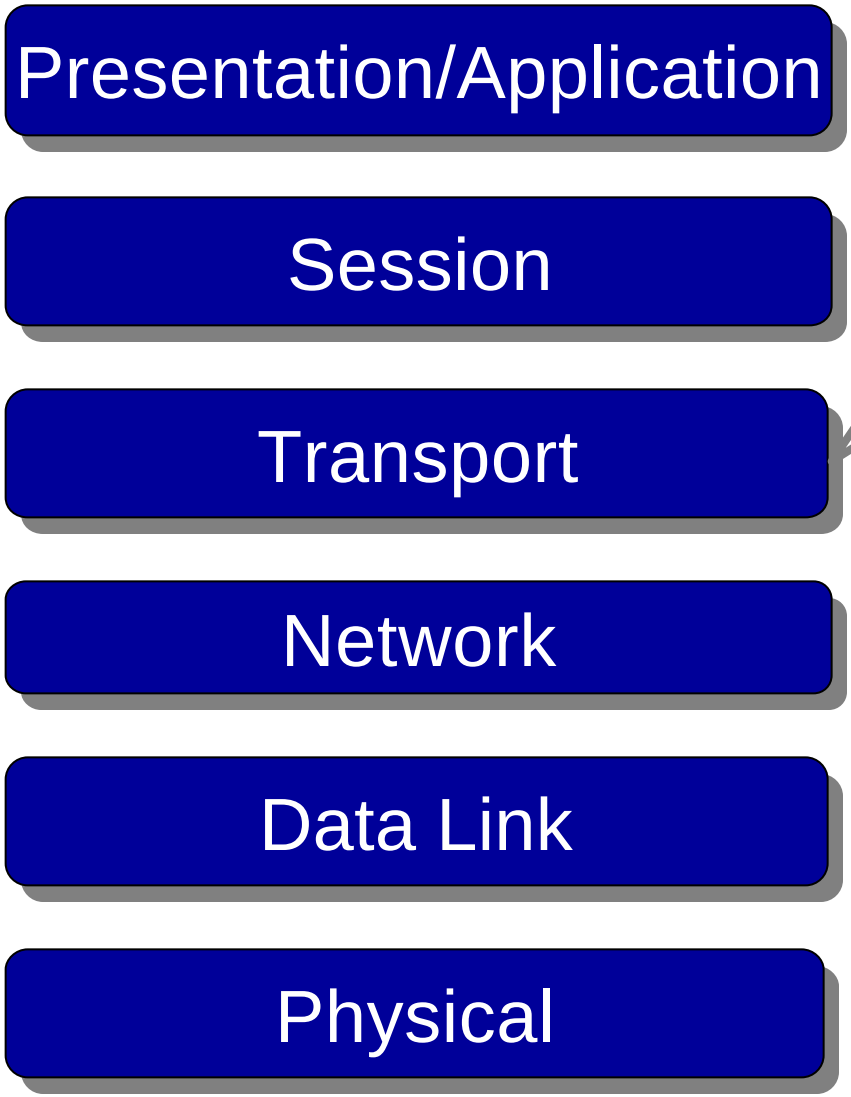
The ISO Protocol Stack



The ISO Protocol Stack



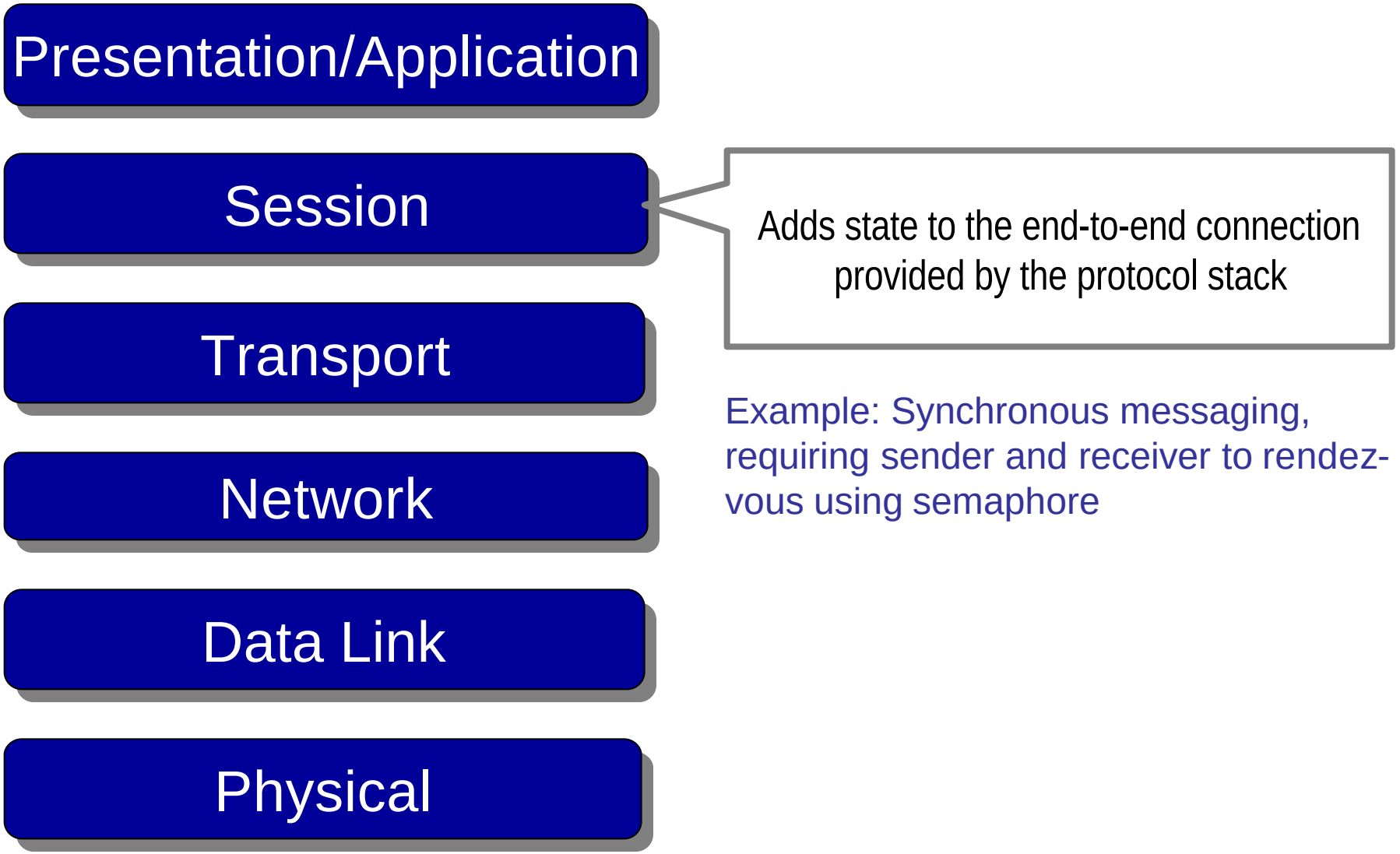
The ISO Protocol Stack



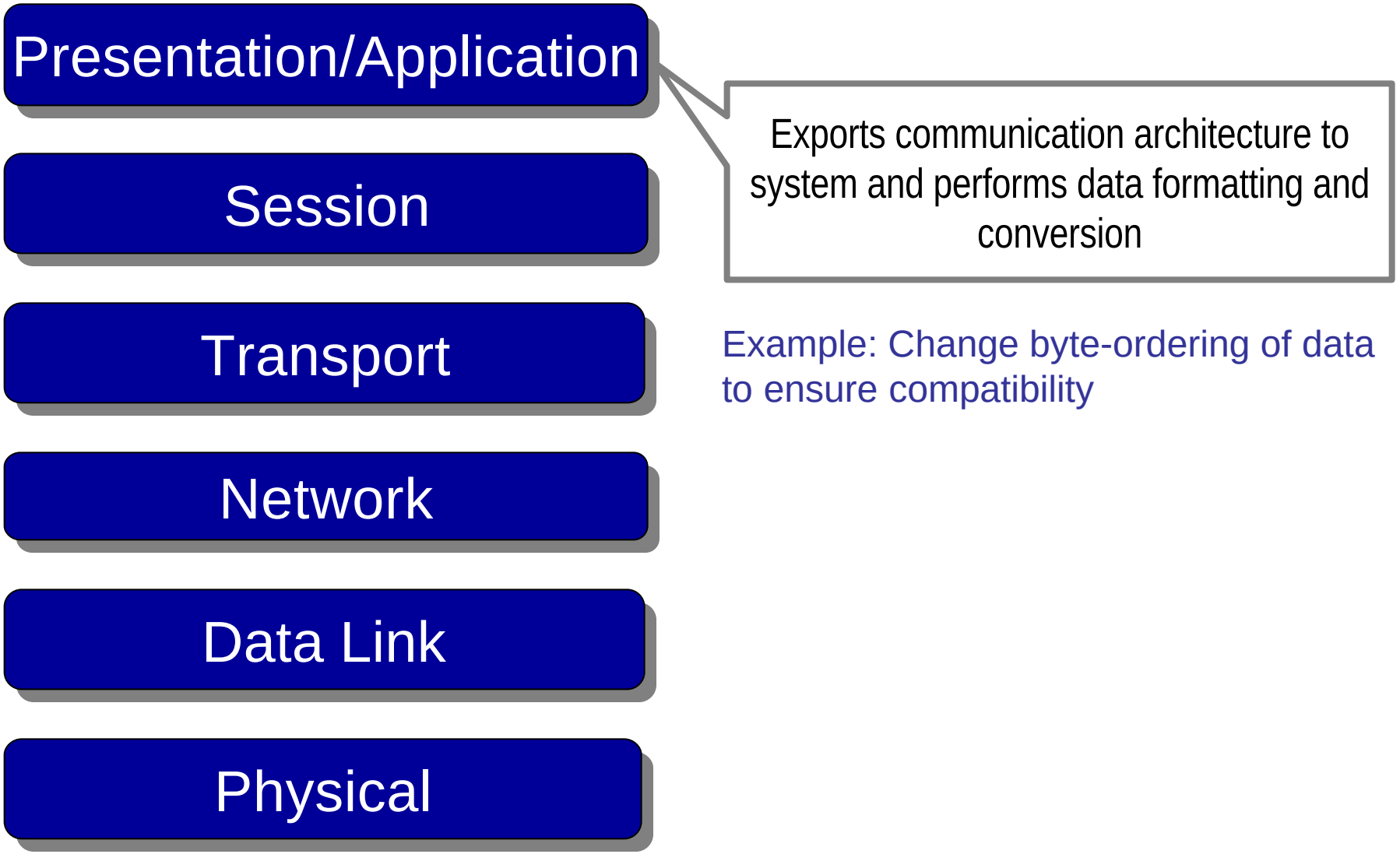
Establish and maintain end-to-end communications (flow control, message reordering, packet segmentation and reassembly)

Example: Establish, maintain and rip-up connections in dynamically reconfigurable SOCs

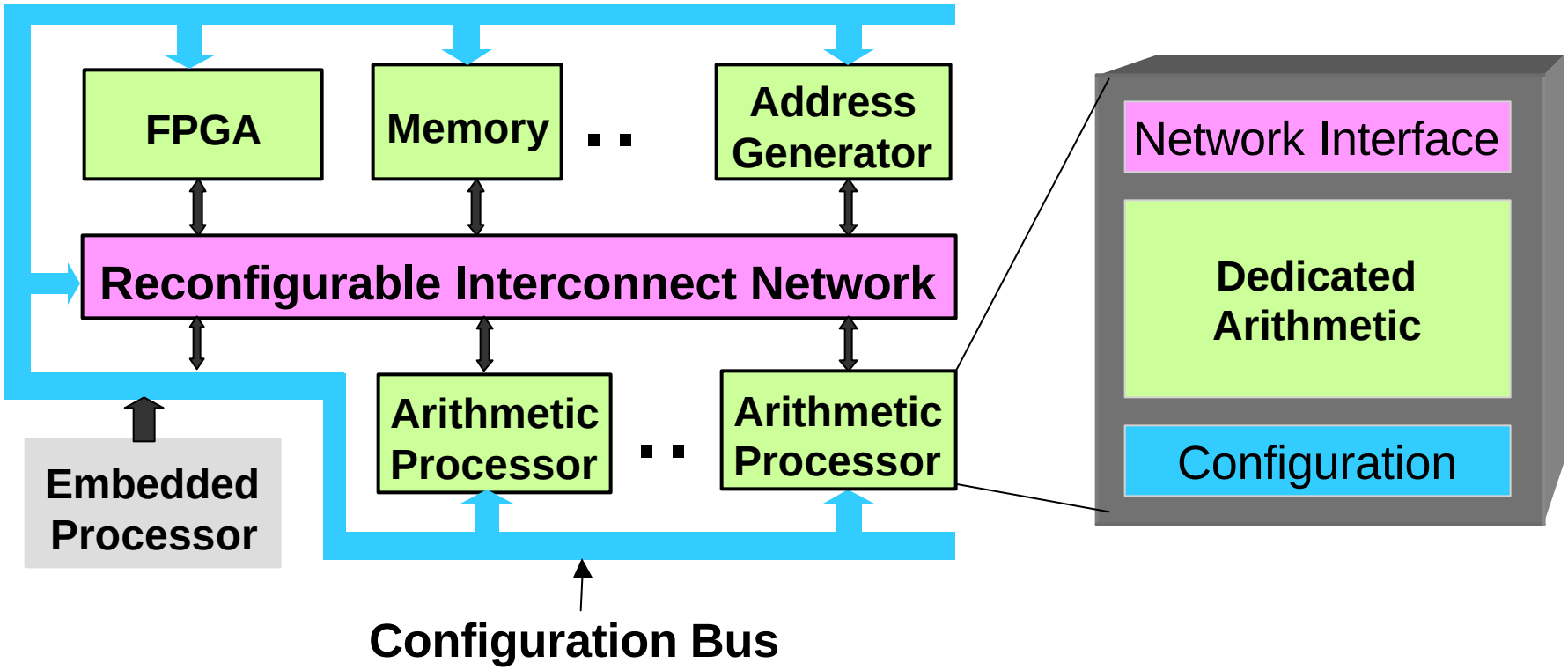
The ISO Protocol Stack



The ISO Protocol Stack



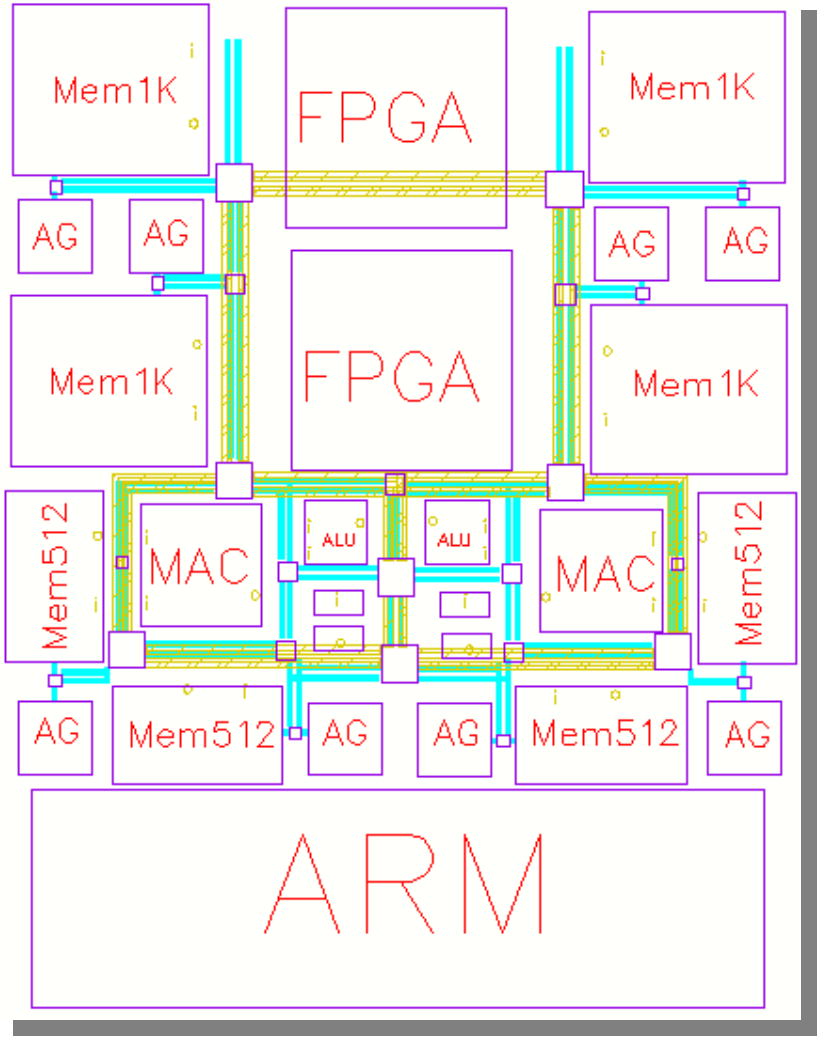
Example: The Pleiades Network-on-a-Chip

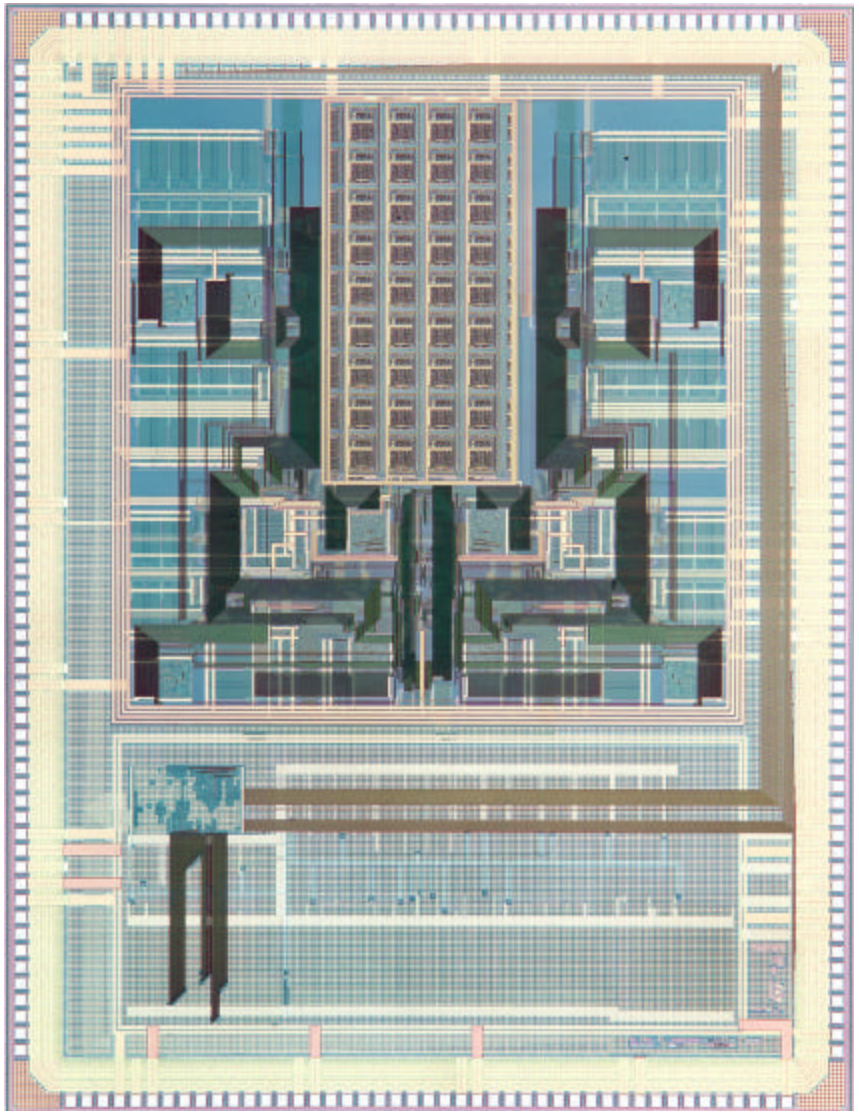


- Programmable/configurable platform intended for low-energy communication and signal-processing applications (wireless, media)
- Allows for dynamic task-level reconfiguration of large-granularity modules into dedicated “data-flow” accelerators

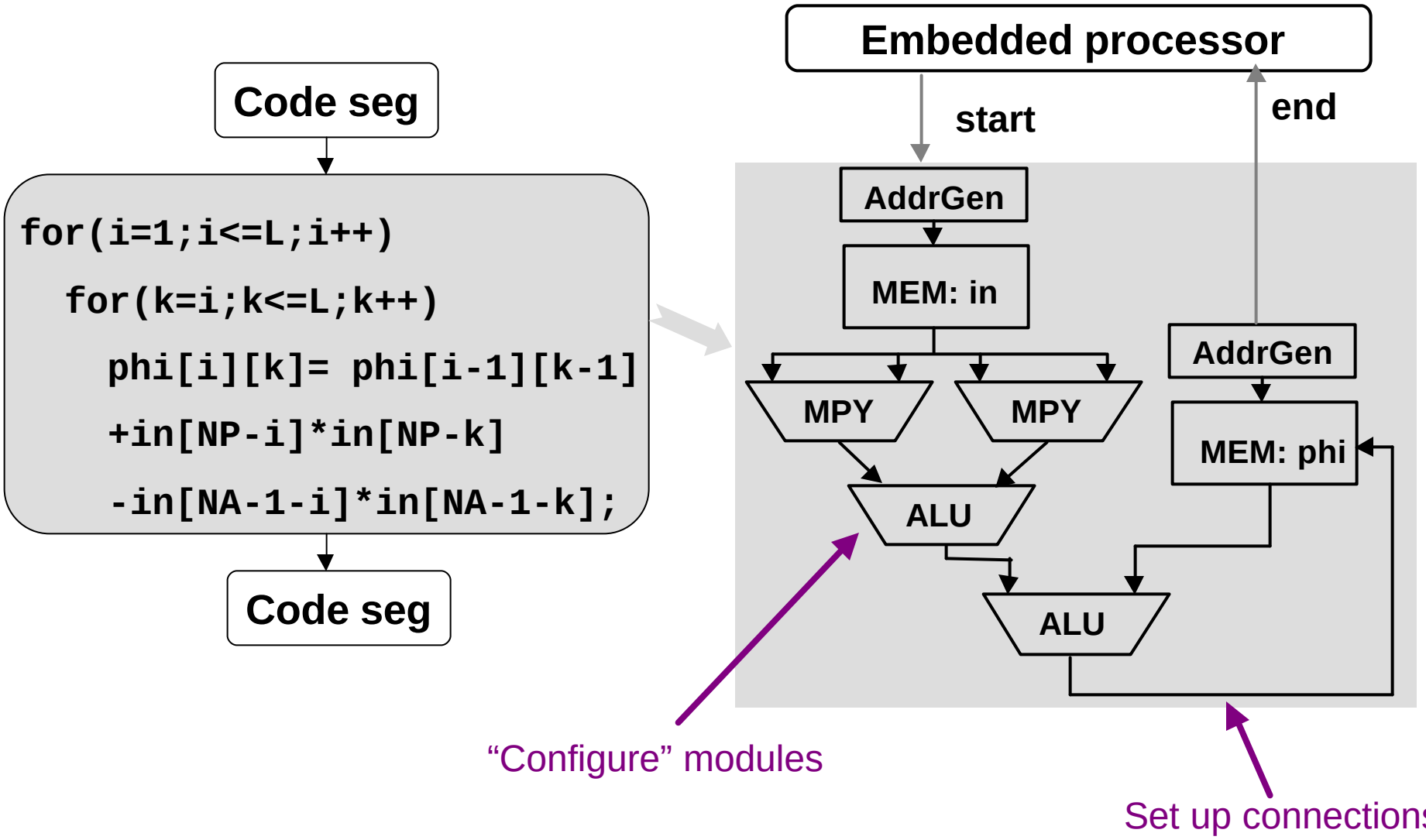
[Zhang, ISSCC 00]

Maia: Reconfigurable Baseband Processor for Wireless



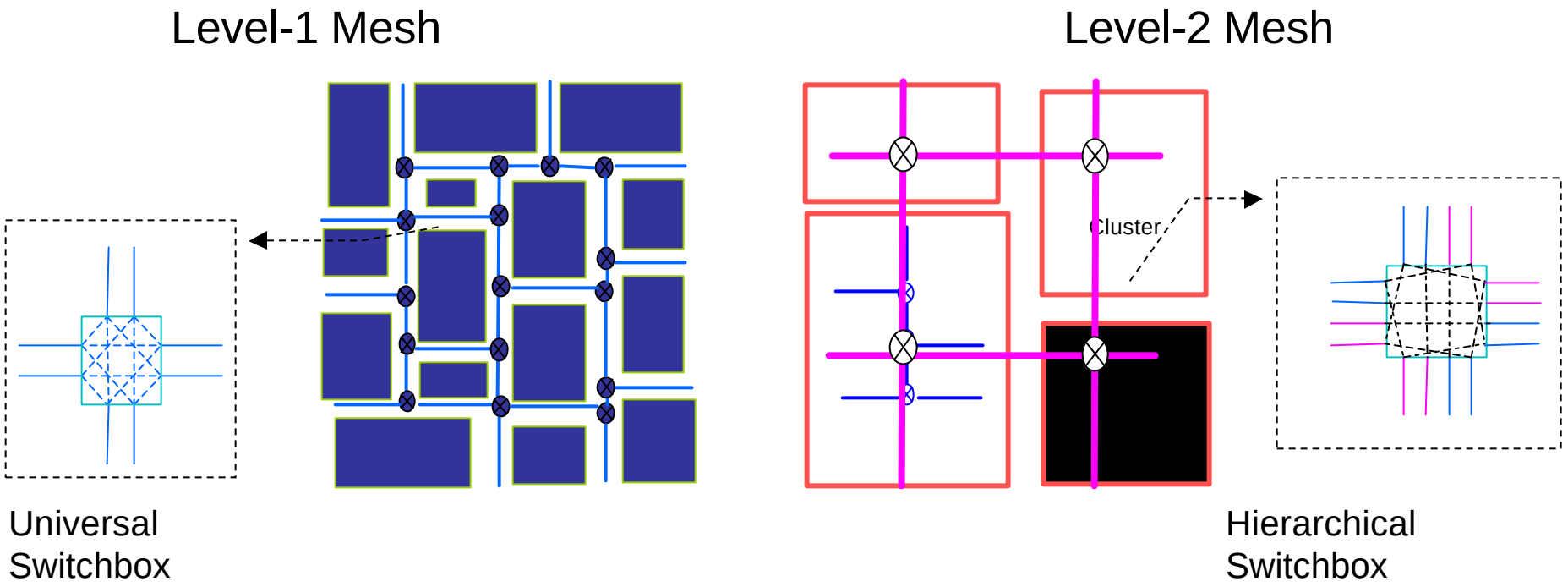


A Session-level Perspective



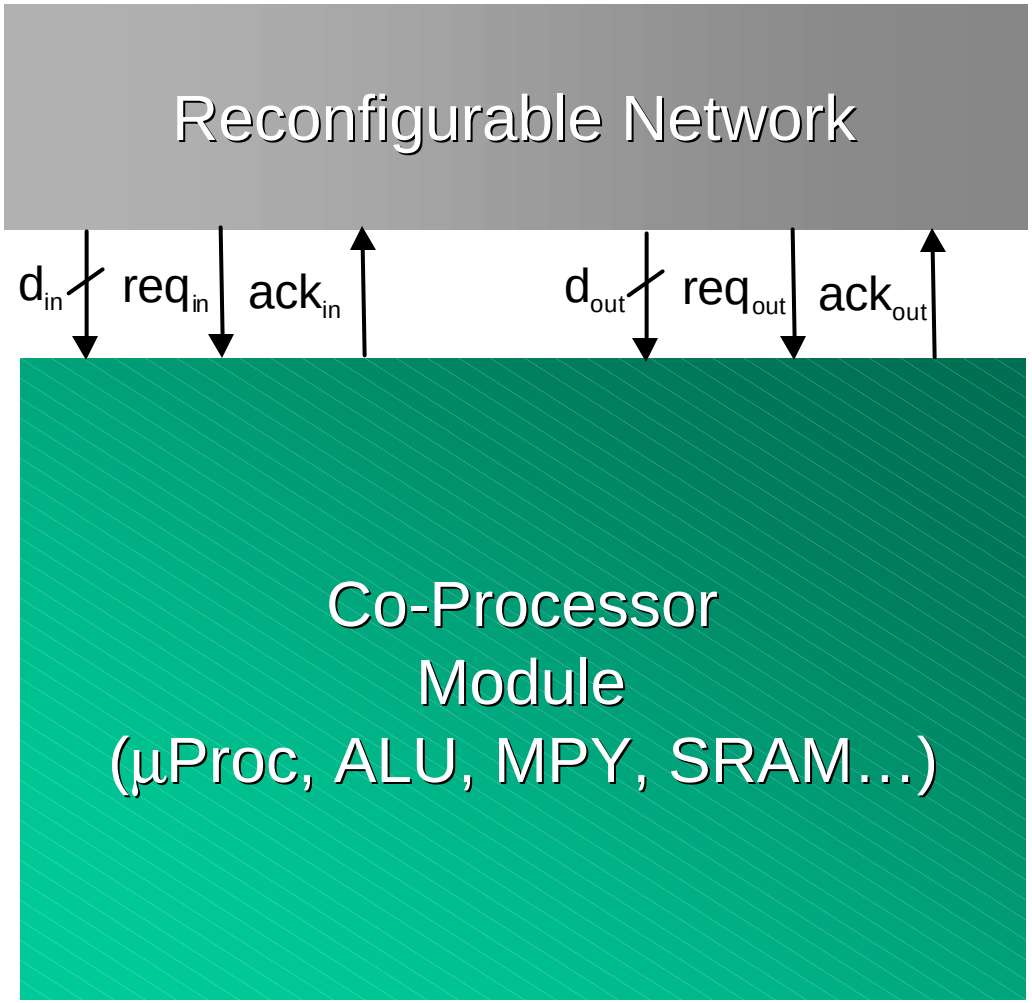
The Network Layer

Hierarchical reconfigurable mesh network



- Network statically configured at start of session and ripped up at end
- Structured approach reduces interconnect energy with factor 7 over straightforward cross-bar

The Physical Layer

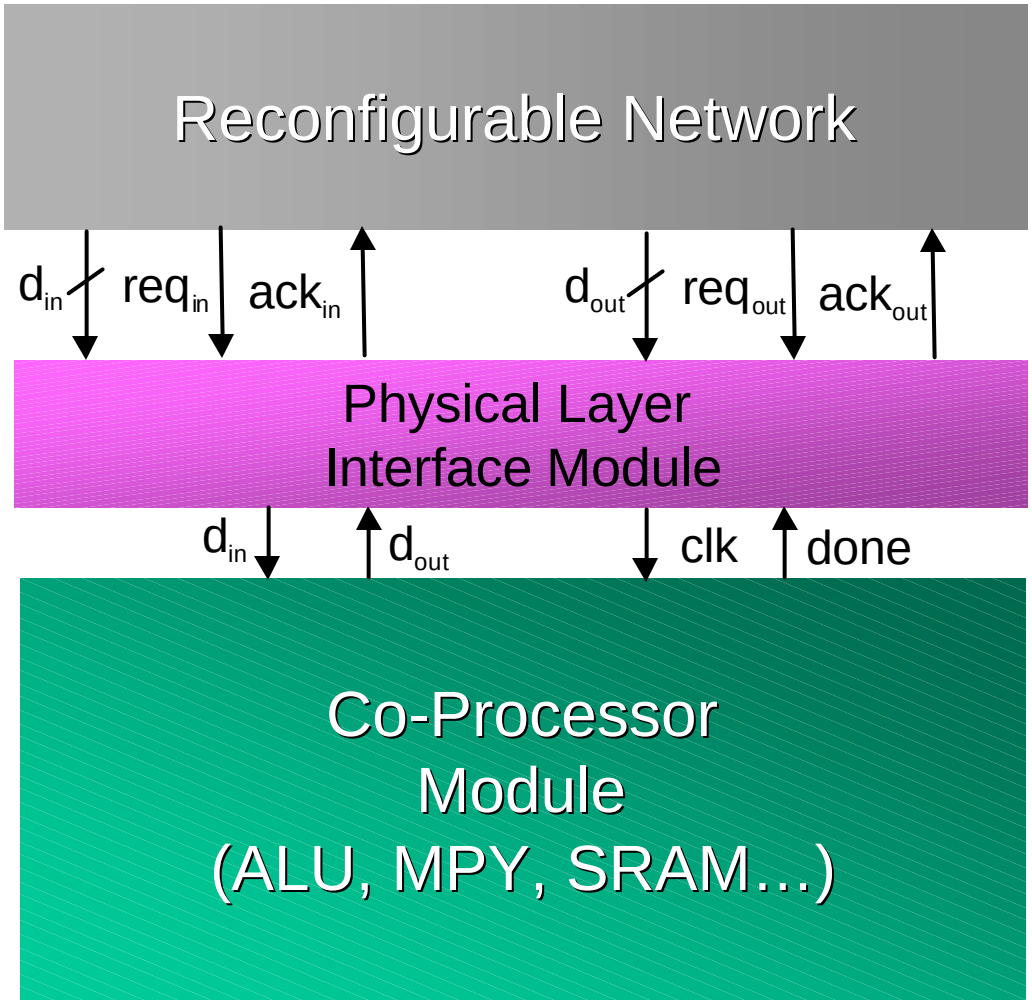


Globally Asynchronous

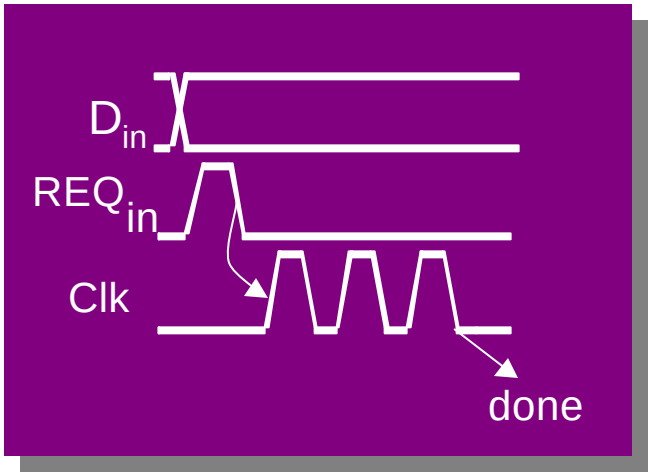
2-phase self-timed handshaking protocol

Allows individual modules to dynamically trade-off performance for energy-efficiency

The Physical Layer

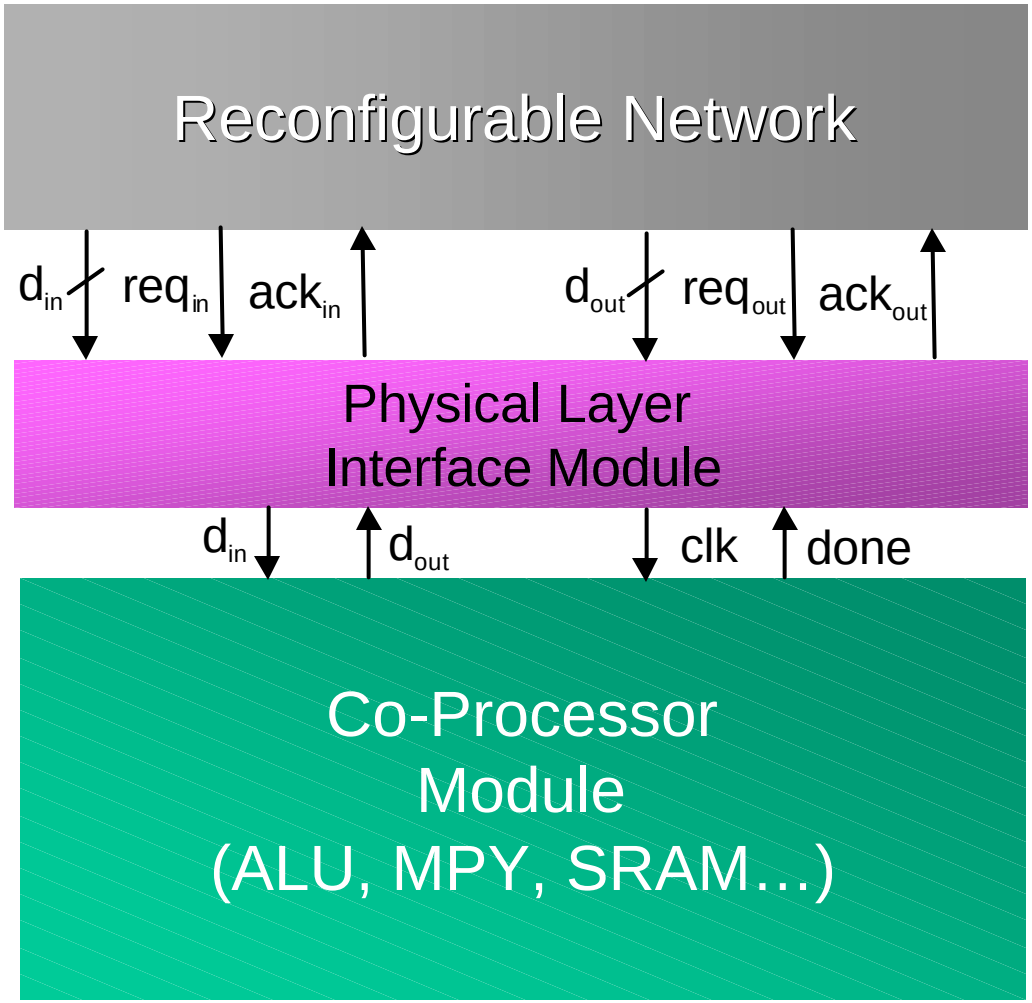


Globally Asynchronous



Locally synchronous

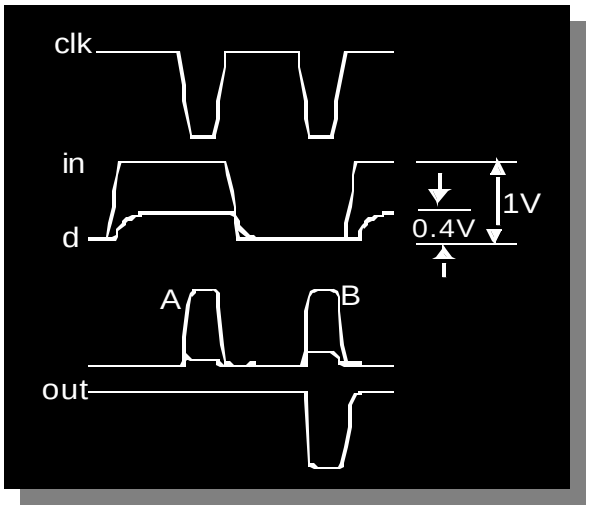
The Physical Layer



Reduced voltage swing on interconnect reduces energy by factor 3.4

0.4 V

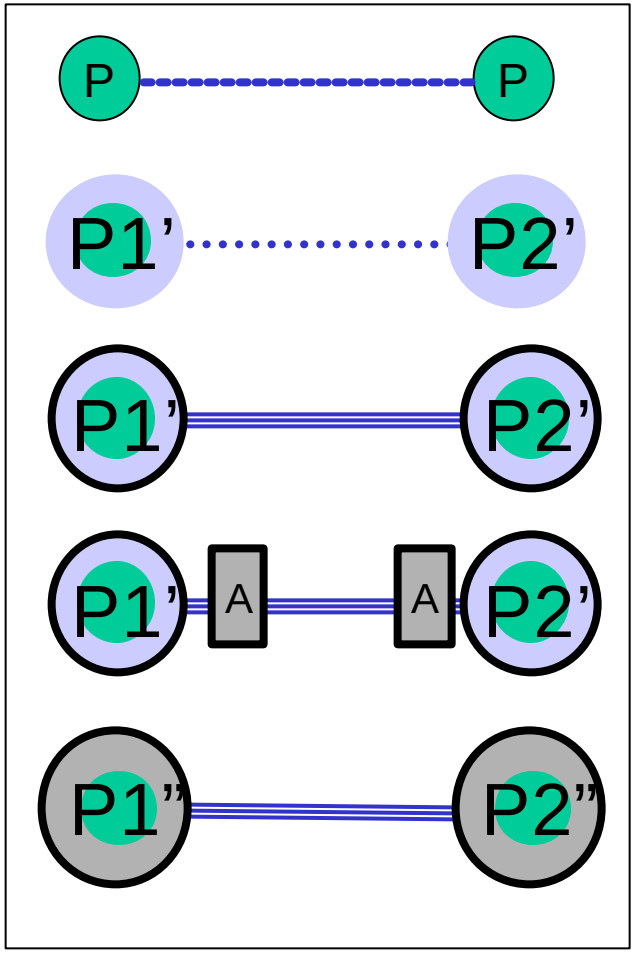
1 V



level-converters

Metropolis Design Methodology

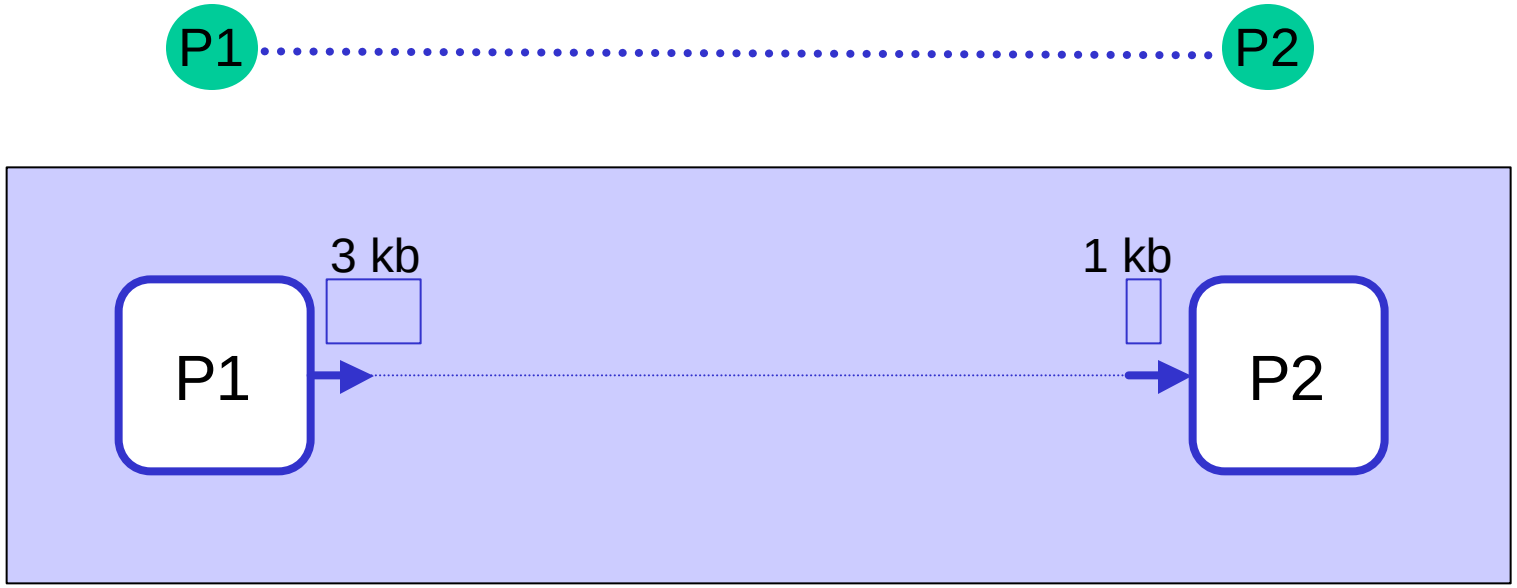
- **Orthogonalization of concerns: separation of communication and computation**
- **Formal system representation (supporting multiple Models of Computation)**
- **Formal Methodology for Communication Refinement: sequence of adaptation steps between objects (processes and channel) with incompatible behaviors**



Metropolis Design Methodology

Behavior Adapter:

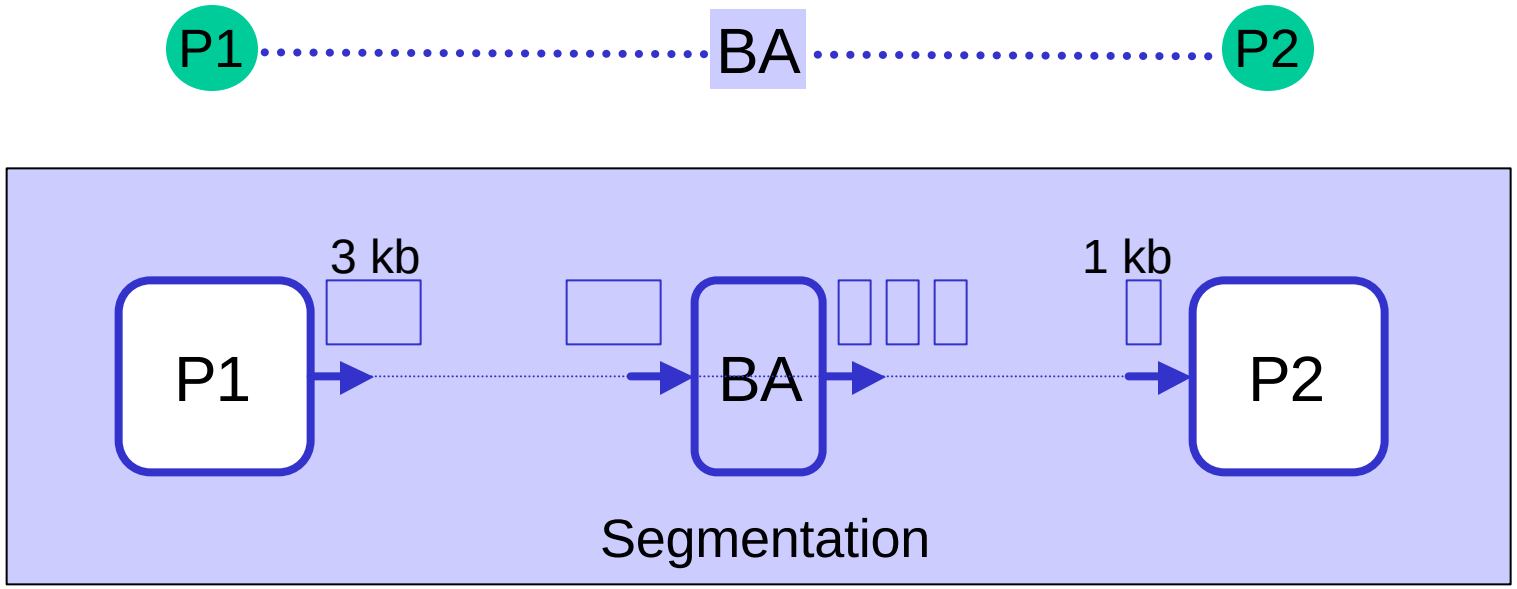
Adapt communicating processes with incompatible behaviors



Metropolis Design Methodology

Behavior Adapter:

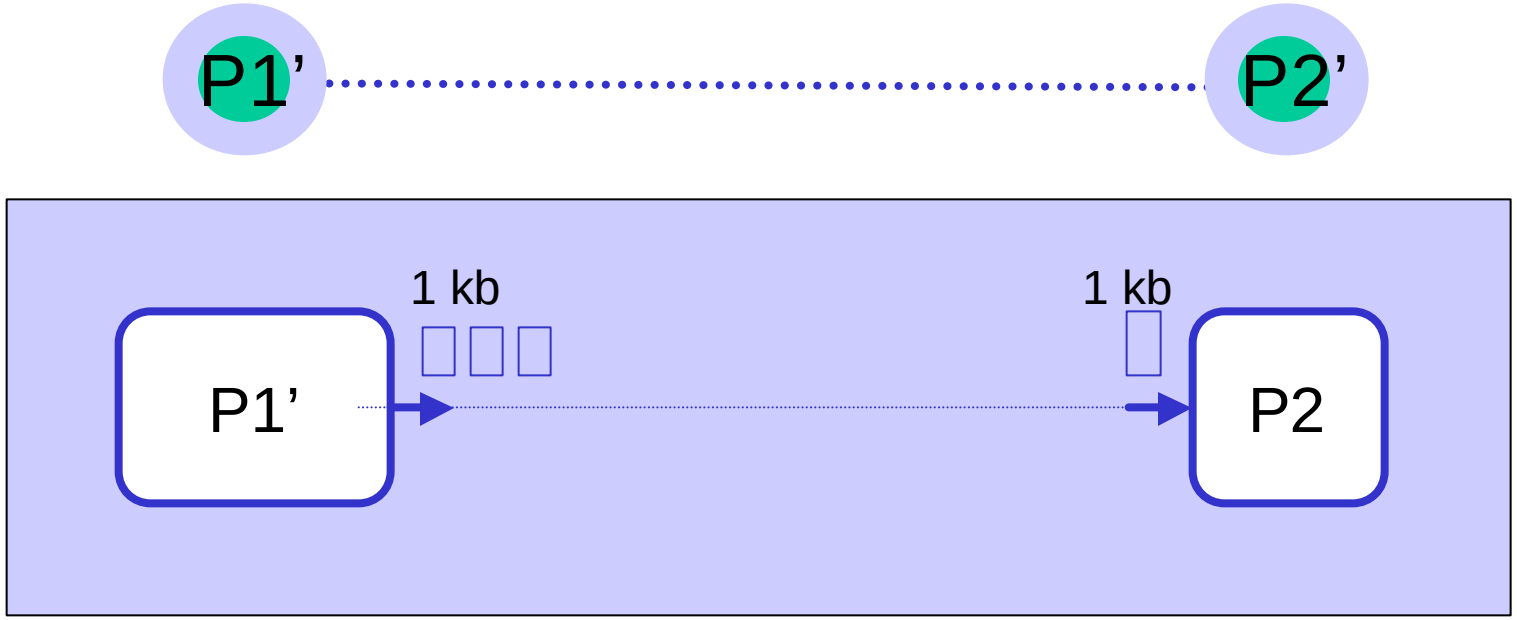
Adapt communicating processes with incompatible behaviors



Metropolis Design Methodology

Behavior Adapter:

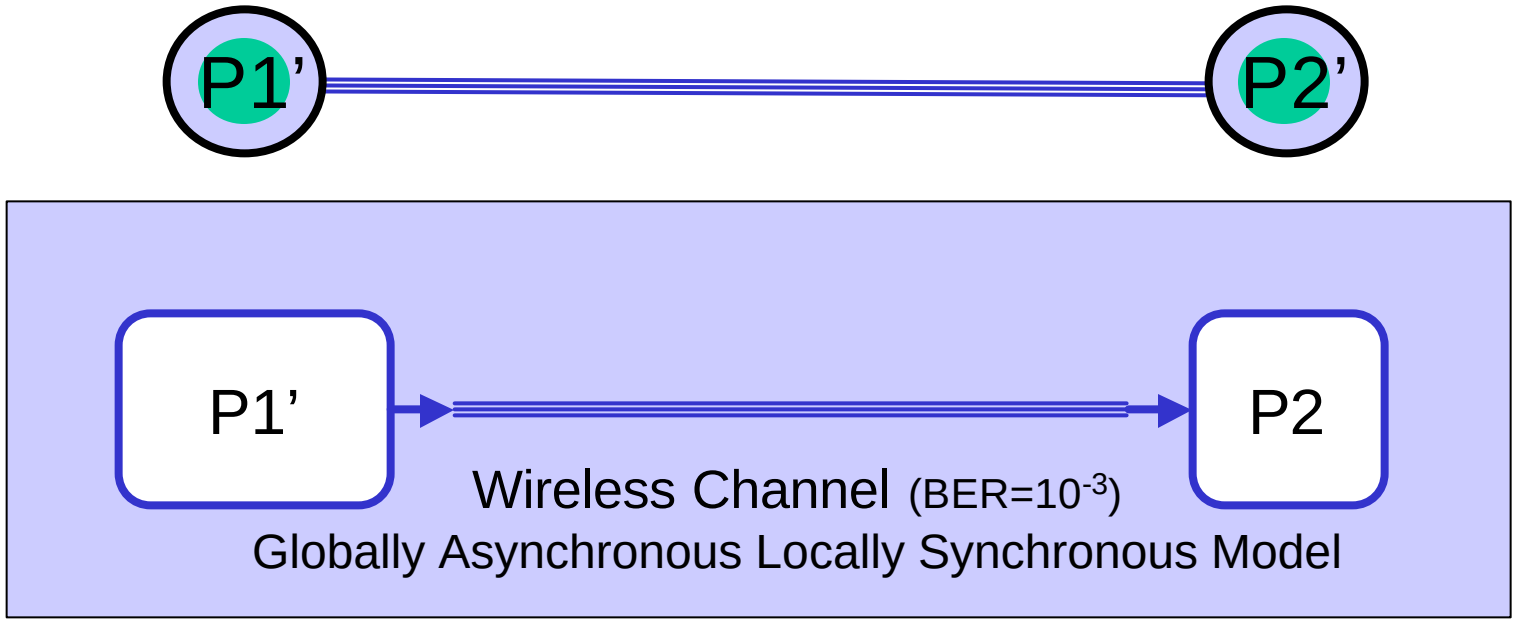
Adapt communicating processes with incompatible behaviors



Metropolis Design Methodology

Channel Selection:

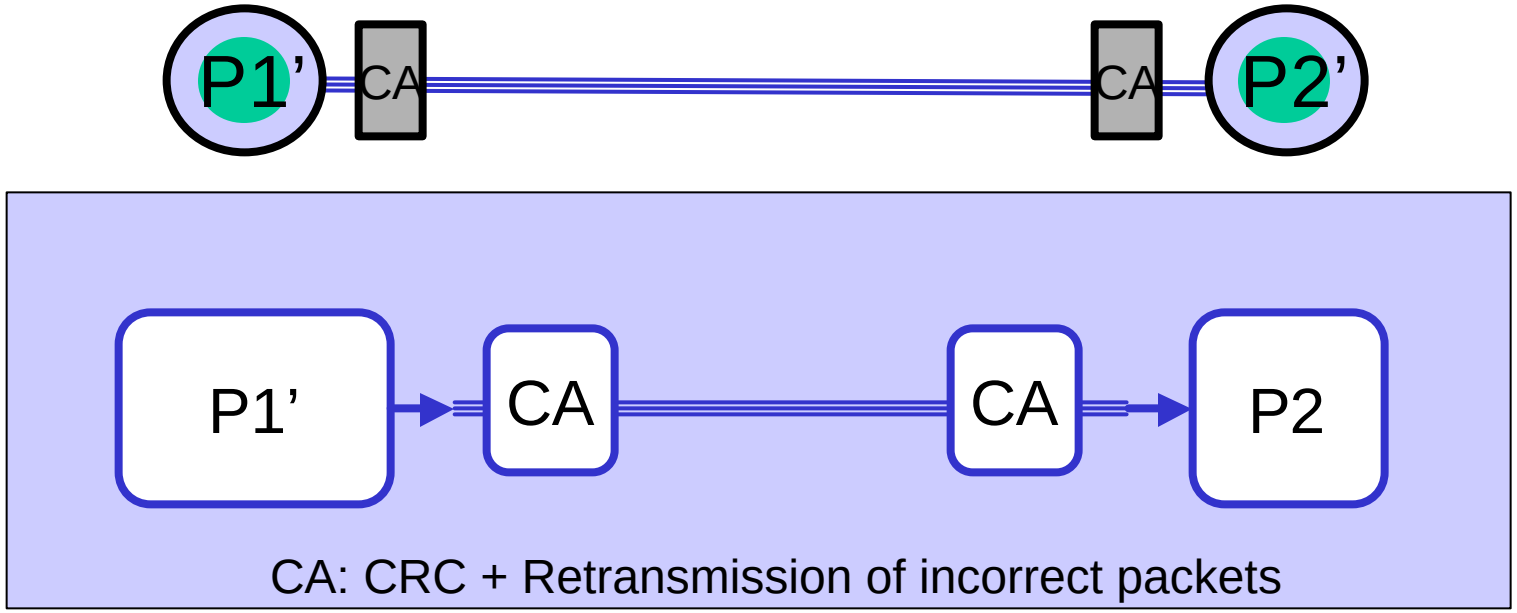
Select a (non-ideal) channel that physically transports messages



Metropolis Design Methodology

Channel Adaptation:

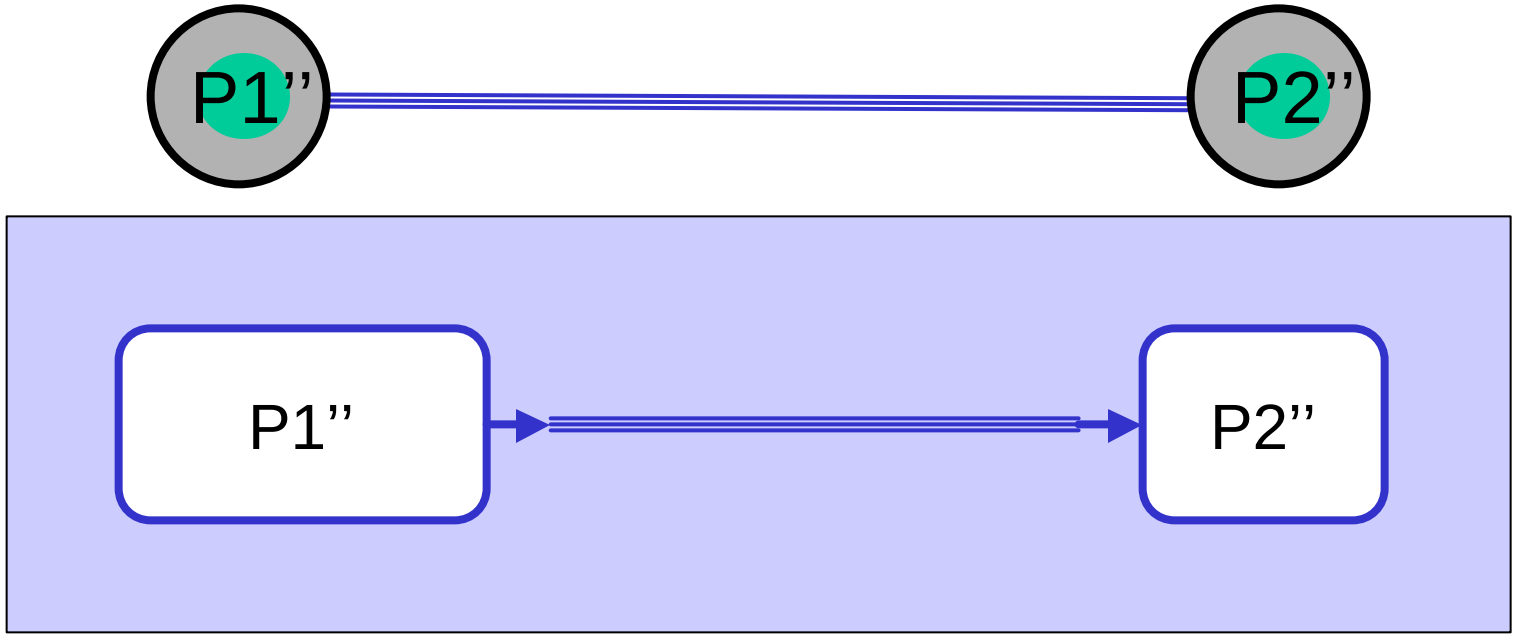
Adapt the behaviors of processes and channel to meet communication requirements



Metropolis Design Methodology

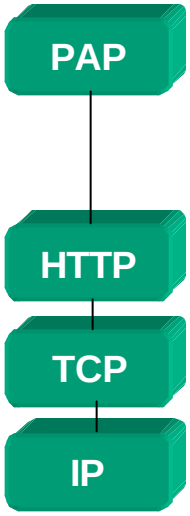
Optimization:

Merge adapters and processes



Example: Wireless Application Protocol

“push-application protocol“

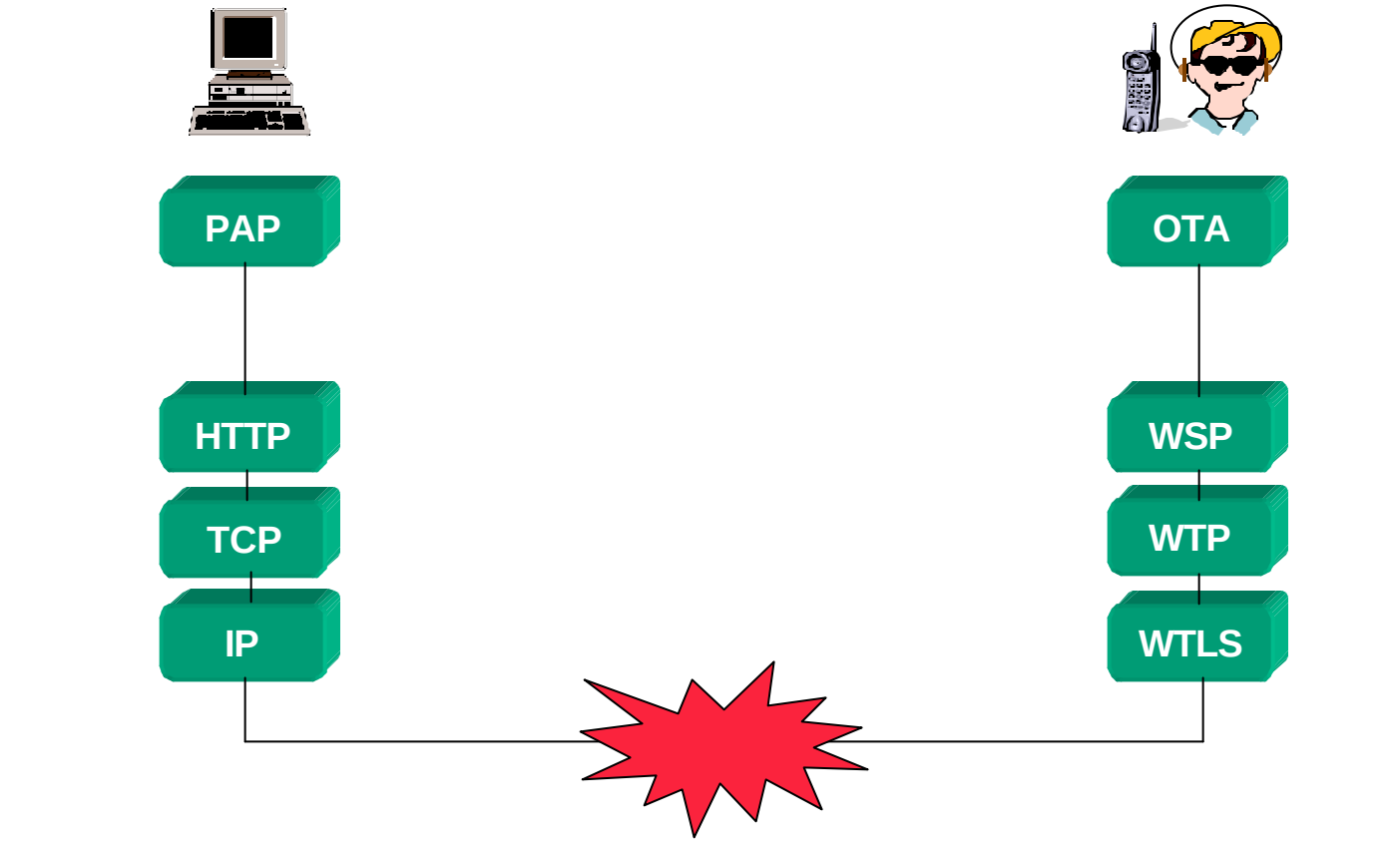


“over-the-air “

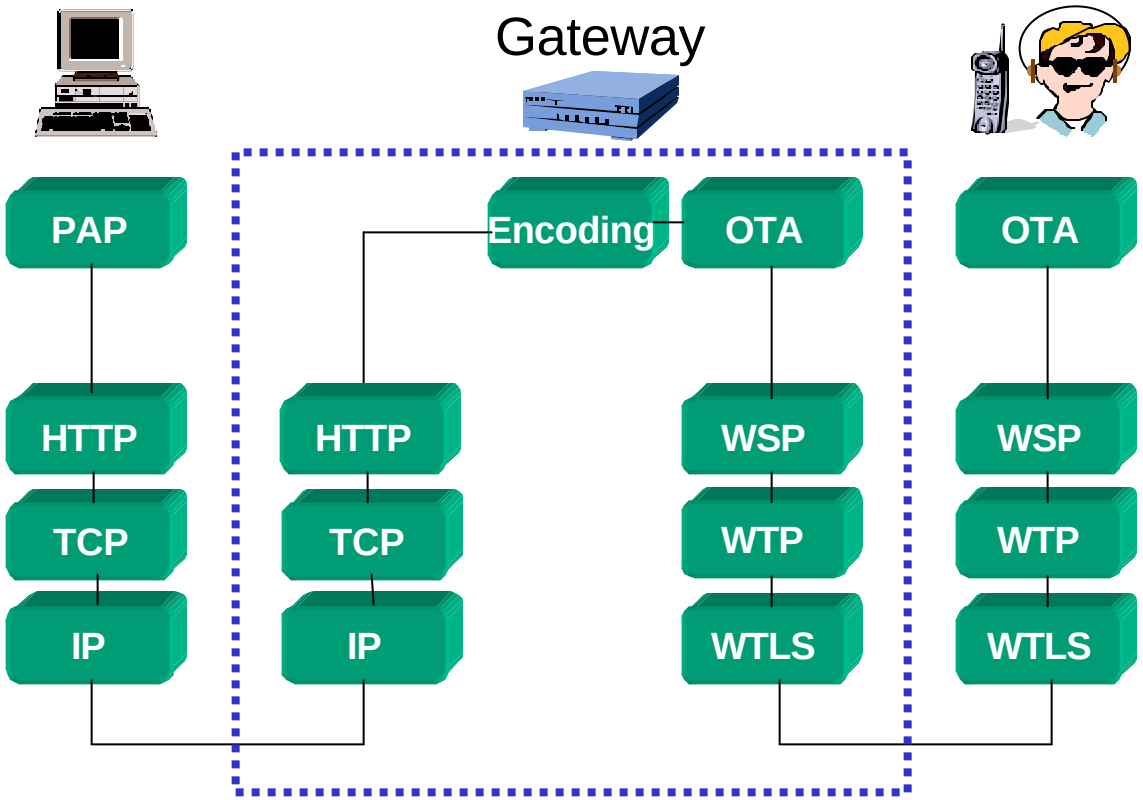
Internet Protocol Stack

WAP Protocol Stack

Example: Wireless Application Protocol

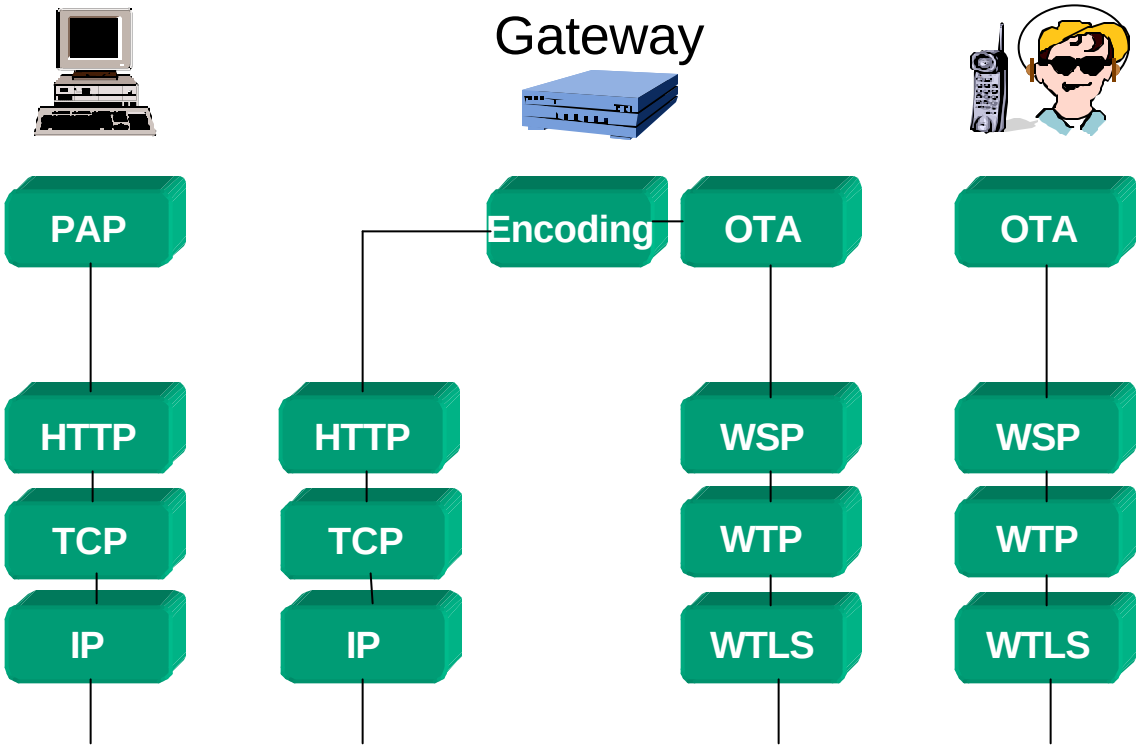


Example: Wireless Application Protocol



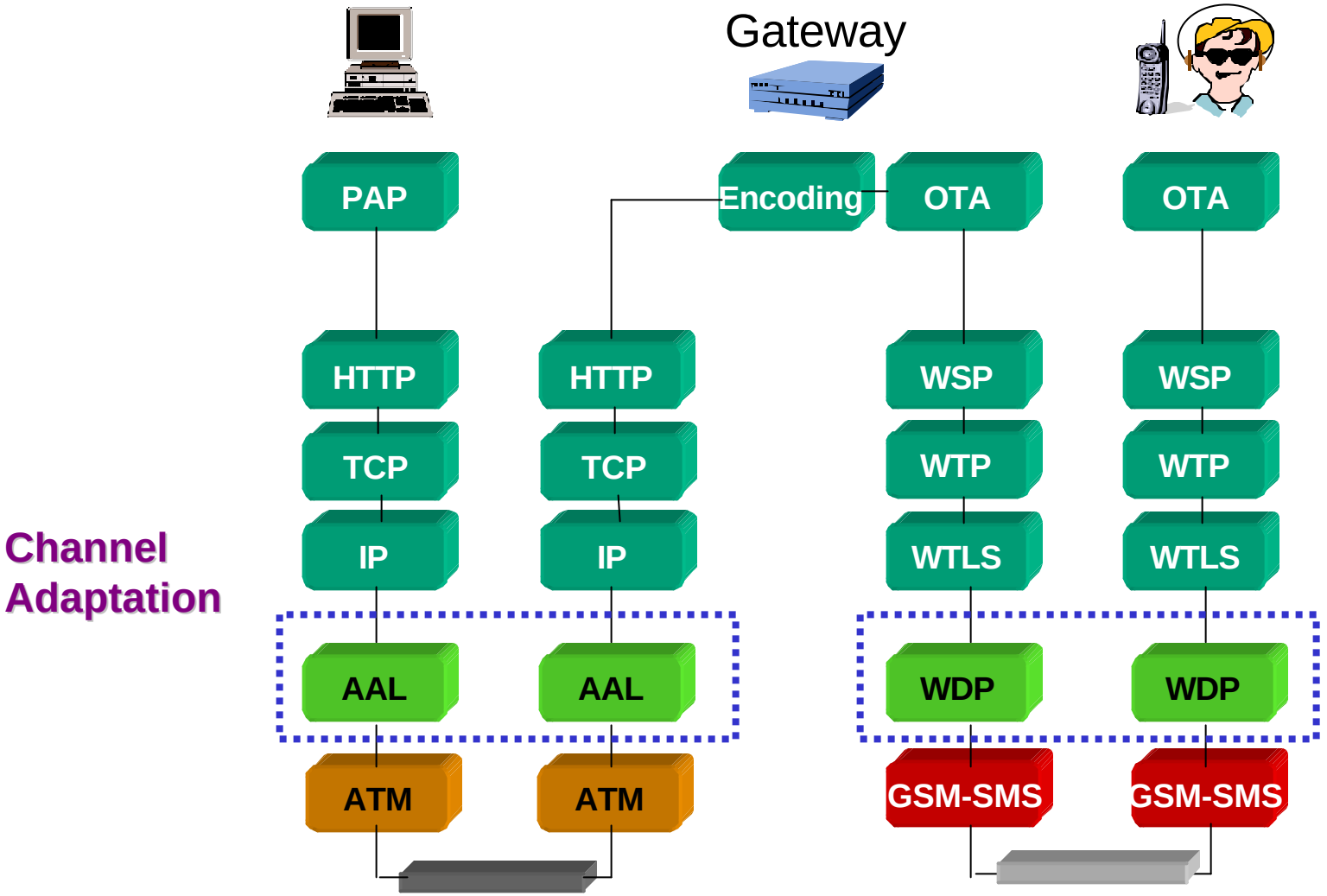
Behavior Adaptation

Example: Wireless Application Protocol

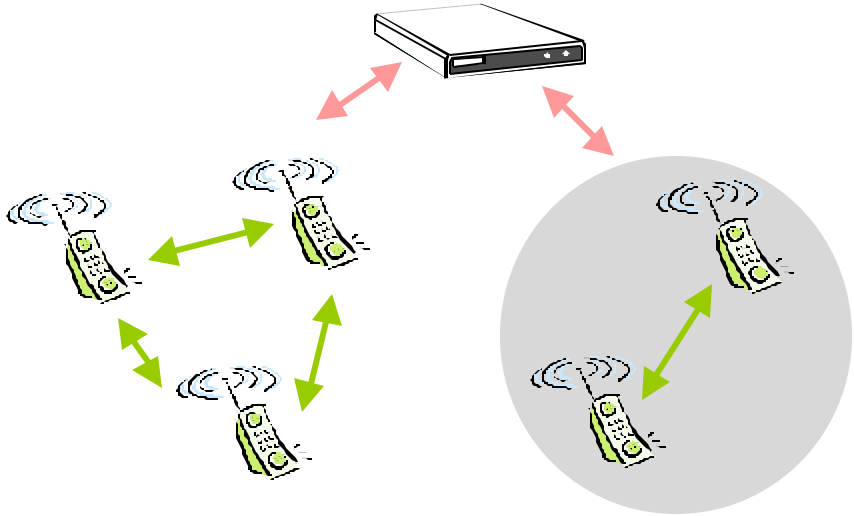


Channel Selection

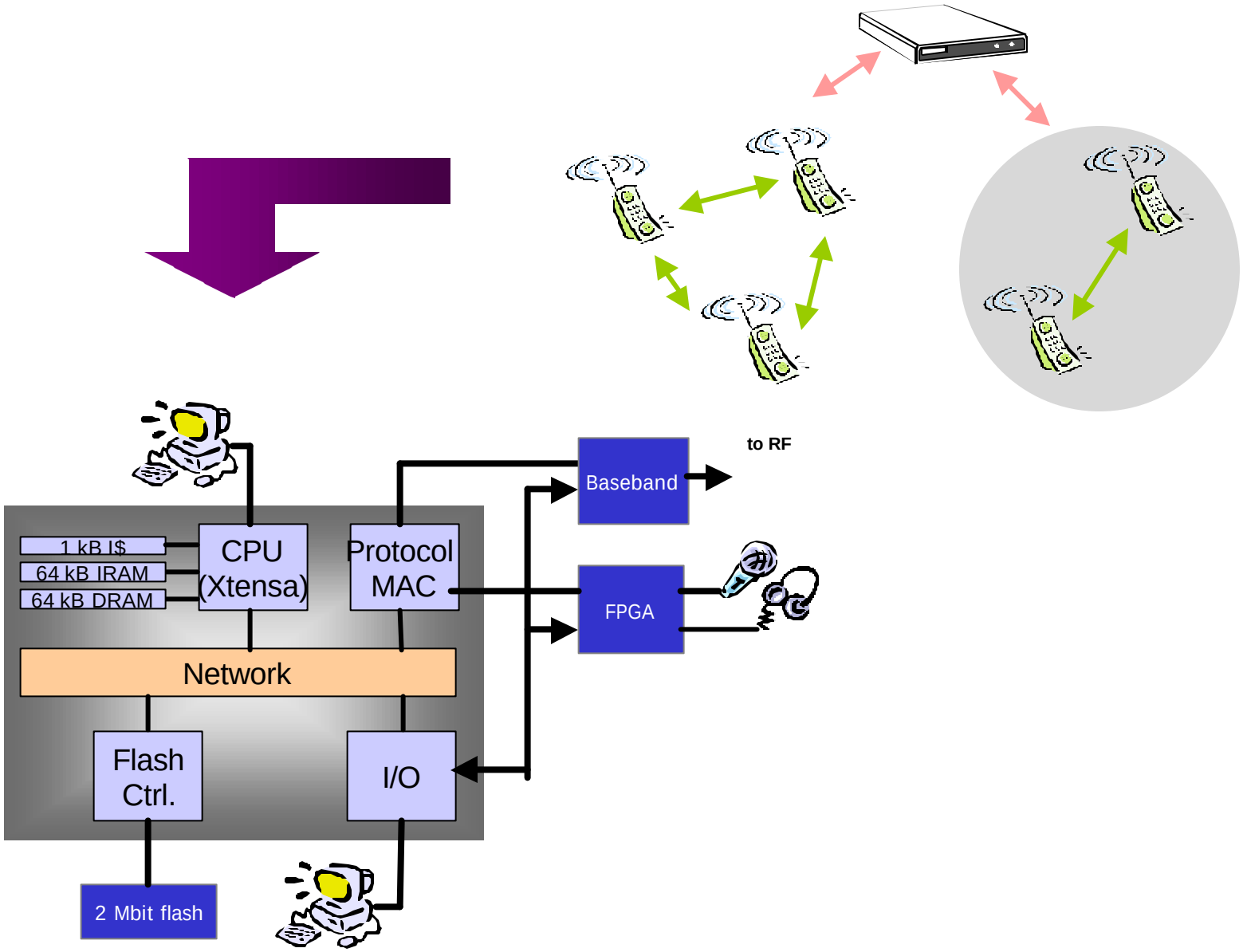
Example: Wireless Application Protocol



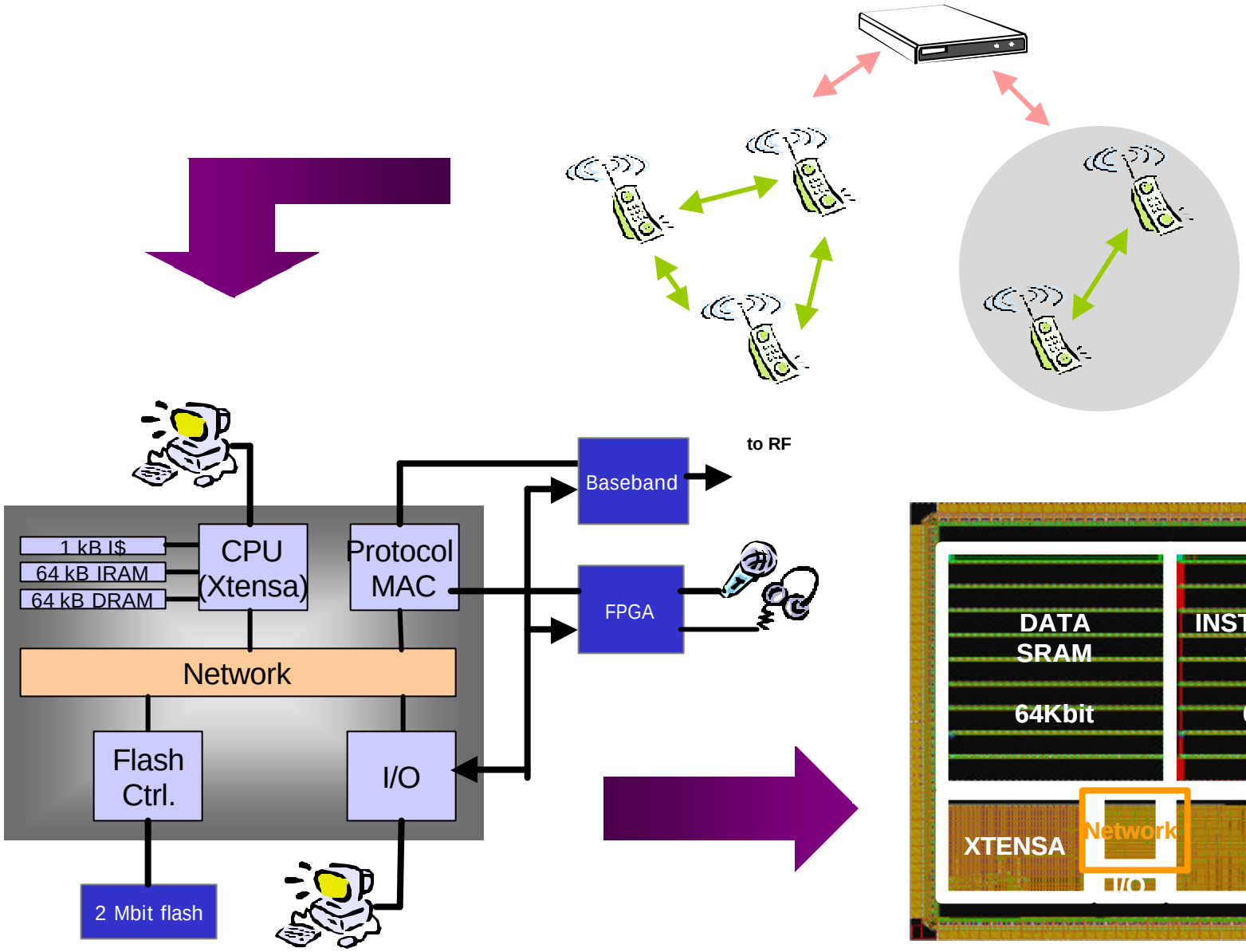
Example: The PicoRadio II (TCI) Design



Example: The PicoRadio II (TCI) Design

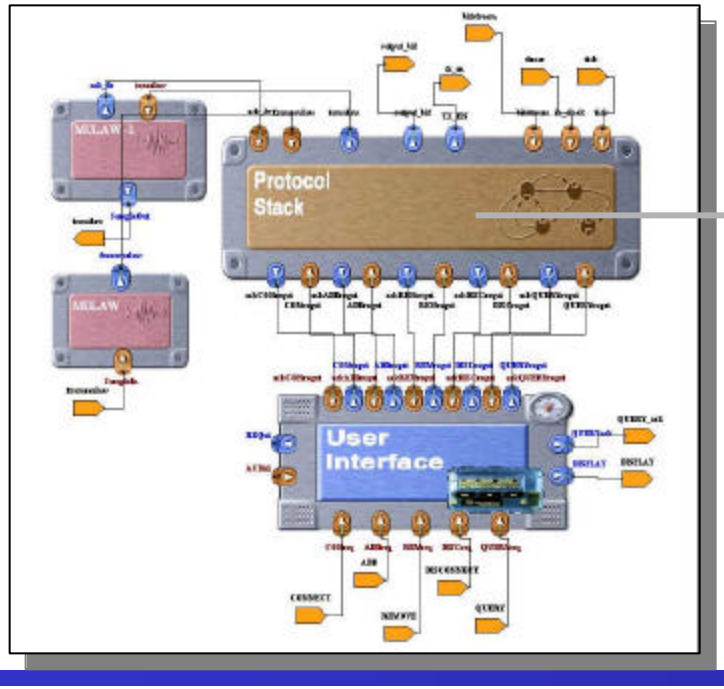


Example: The PicoRadio II (TCI) Design

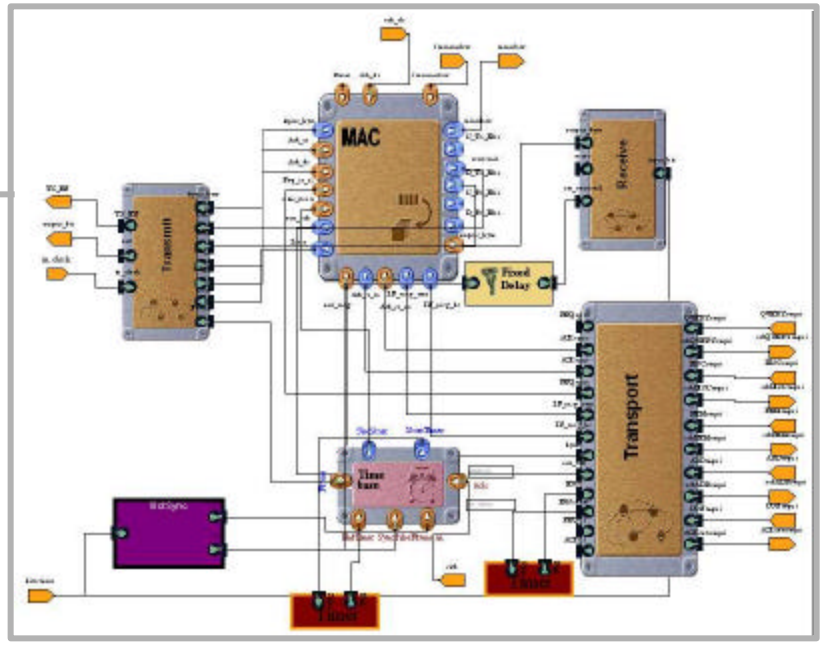


Mapping Behavior onto Architecture*

Behavior described using CFSMs



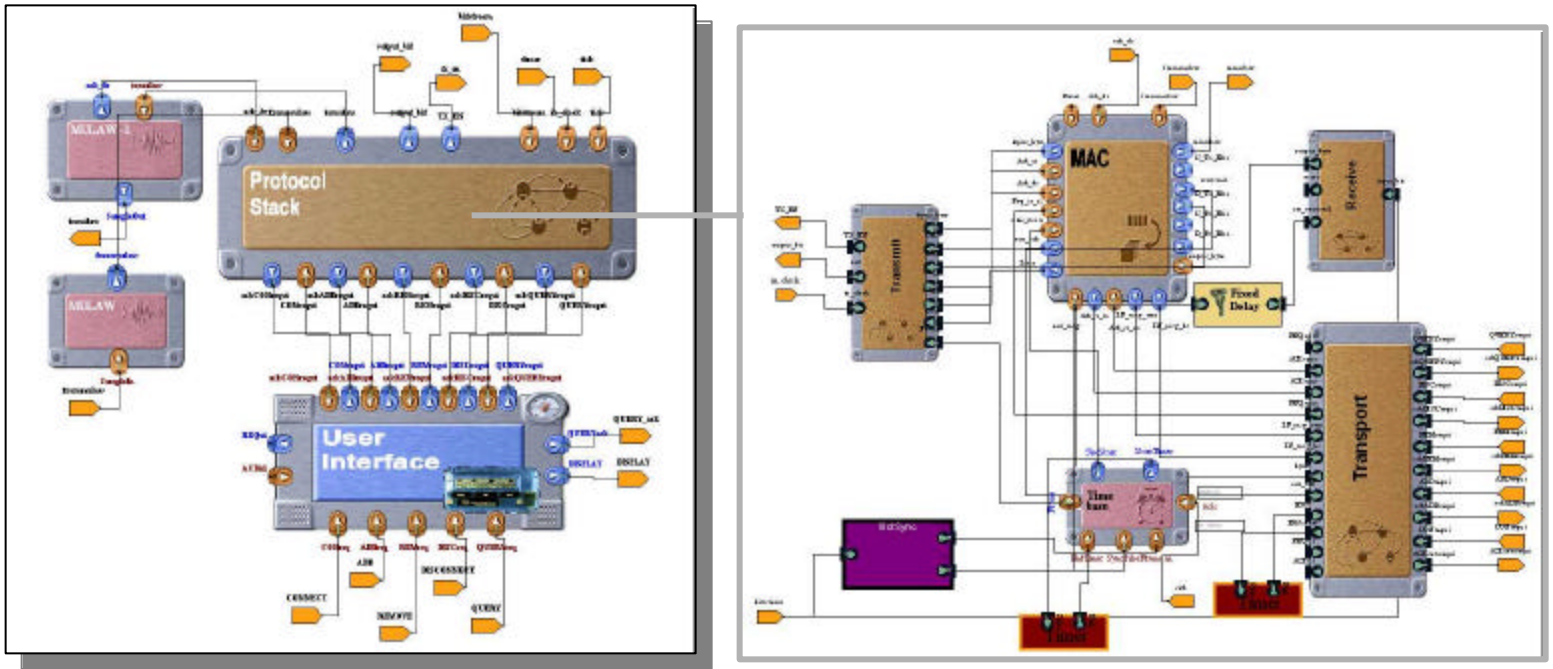
Behavior



*Using the VCC tools from Cadence Design Systems

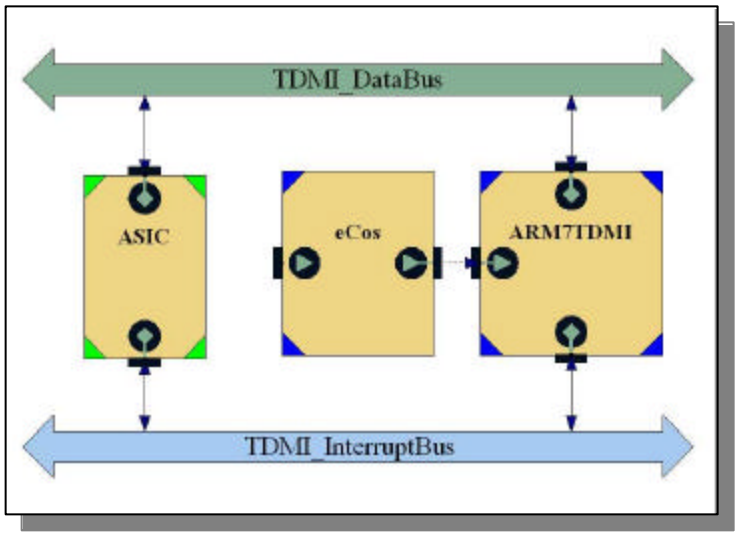
Mapping Behavior onto Architecture*

Behavior described using CFSMs



Behavior

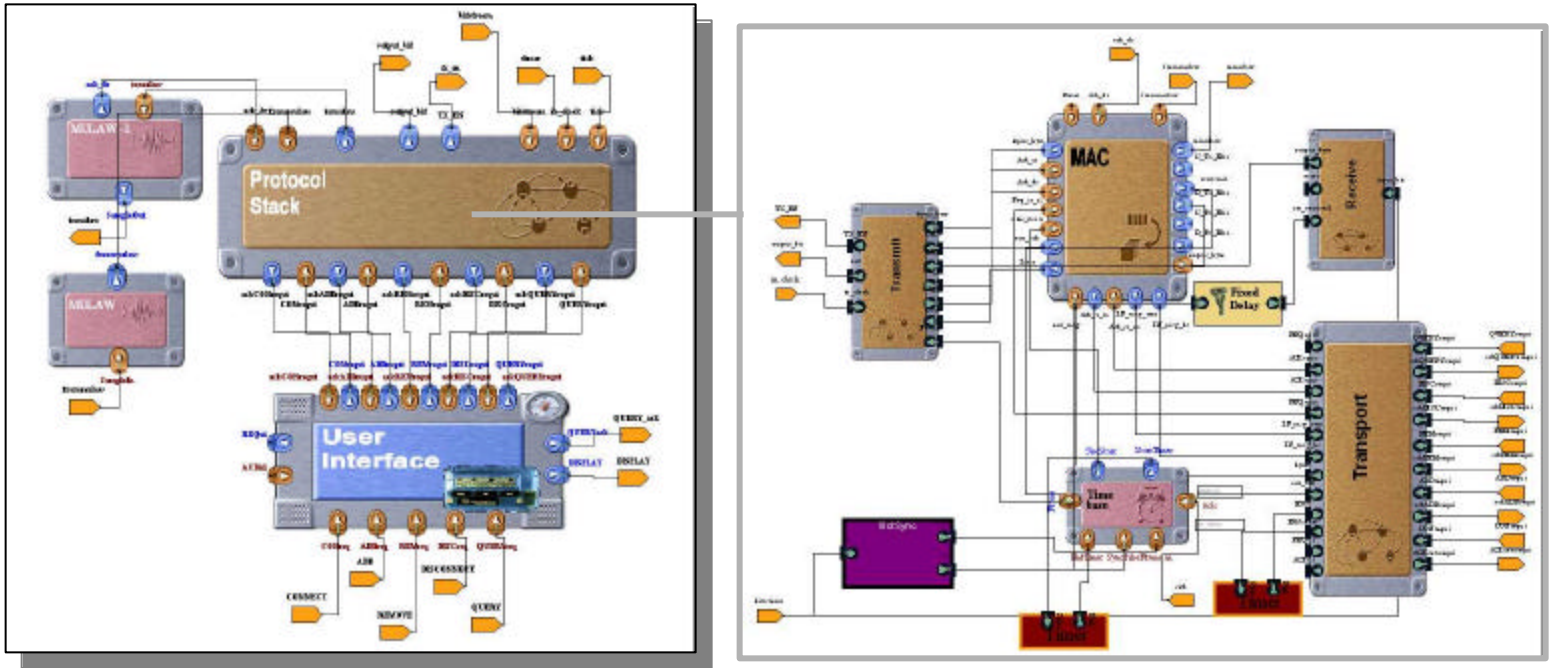
Architecture



*Using the VCC tools from Cadence Design Systems

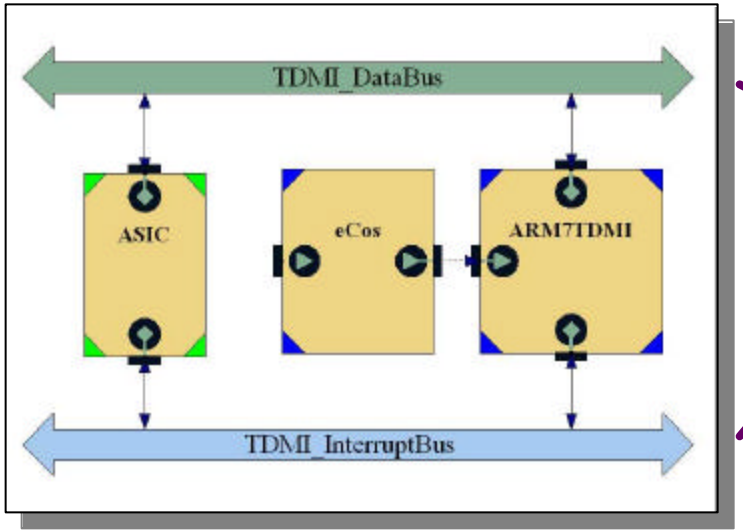
Mapping Behavior onto Architecture*

Behavior described using CFSMs



Behavior

Architecture



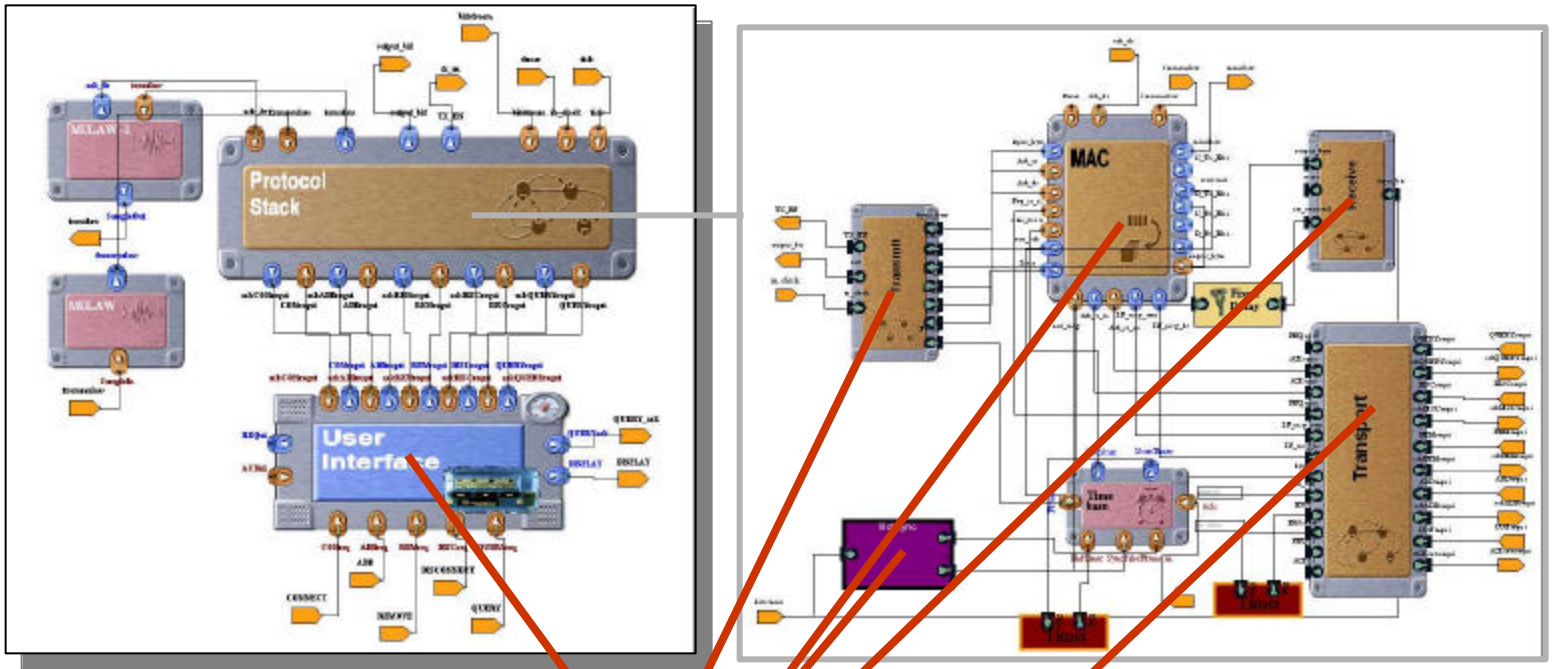
Collection of interconnect models enables exploration

*Using the VCC tools from Cadence Design Systems

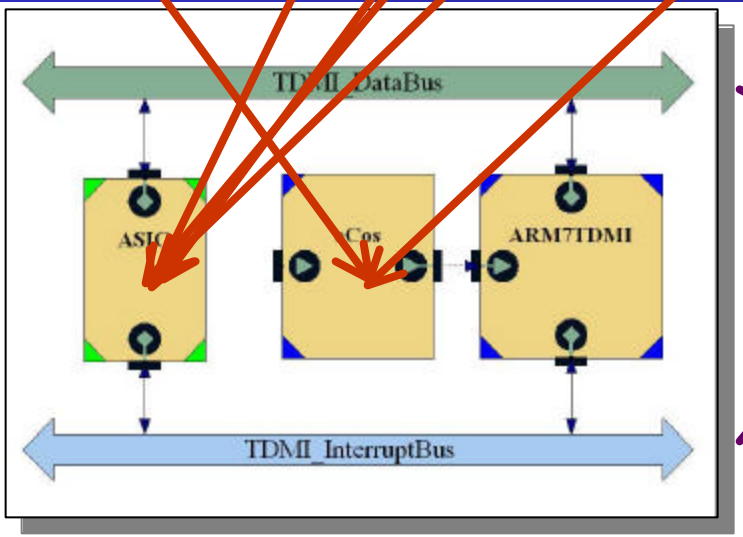
Mapping Behavior onto Architecture*

Behavior described using CFSMs

Behavior



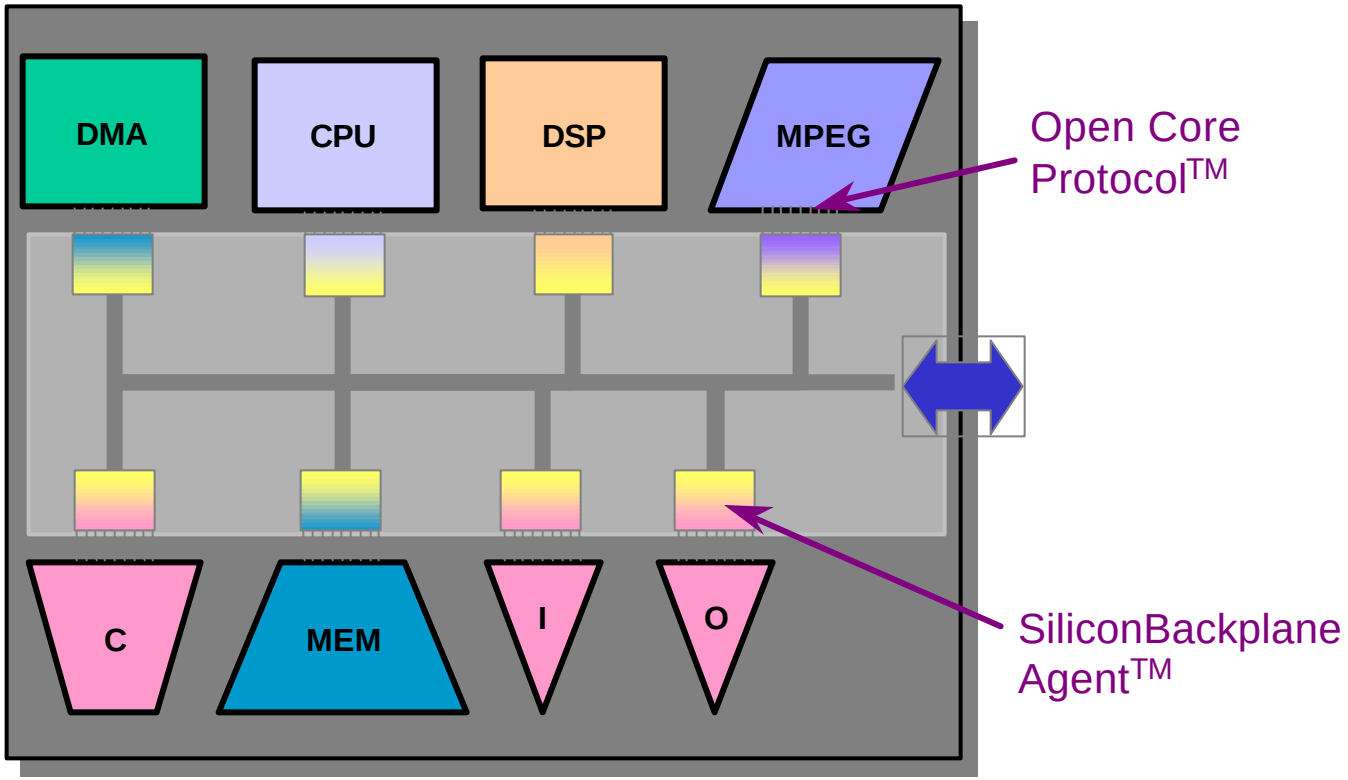
Architecture



Collection of interconnect models enables exploration

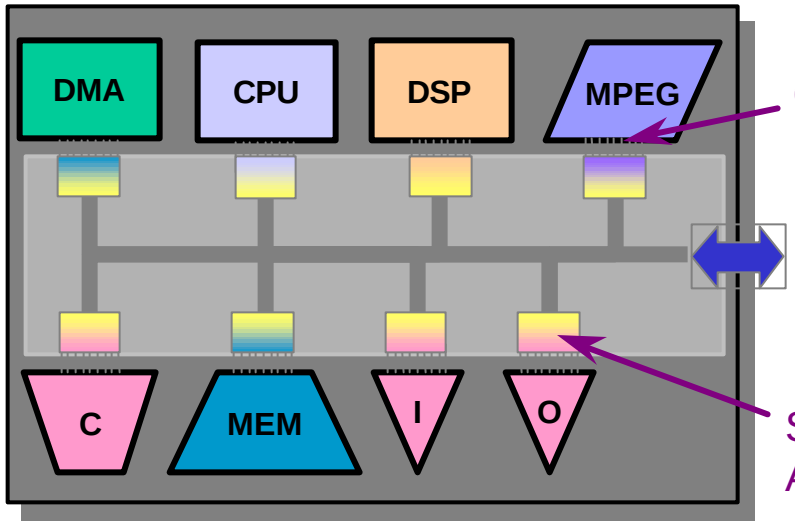
*Using the VCC tools from Cadence Design Systems

Choosing the Interconnect Architecture

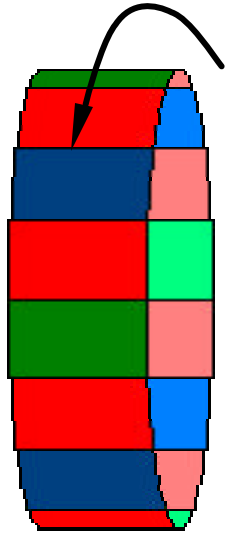


"The Silicon Backplane"
(Courtesy Sonics, Inc)

Choosing the Interconnect Architecture

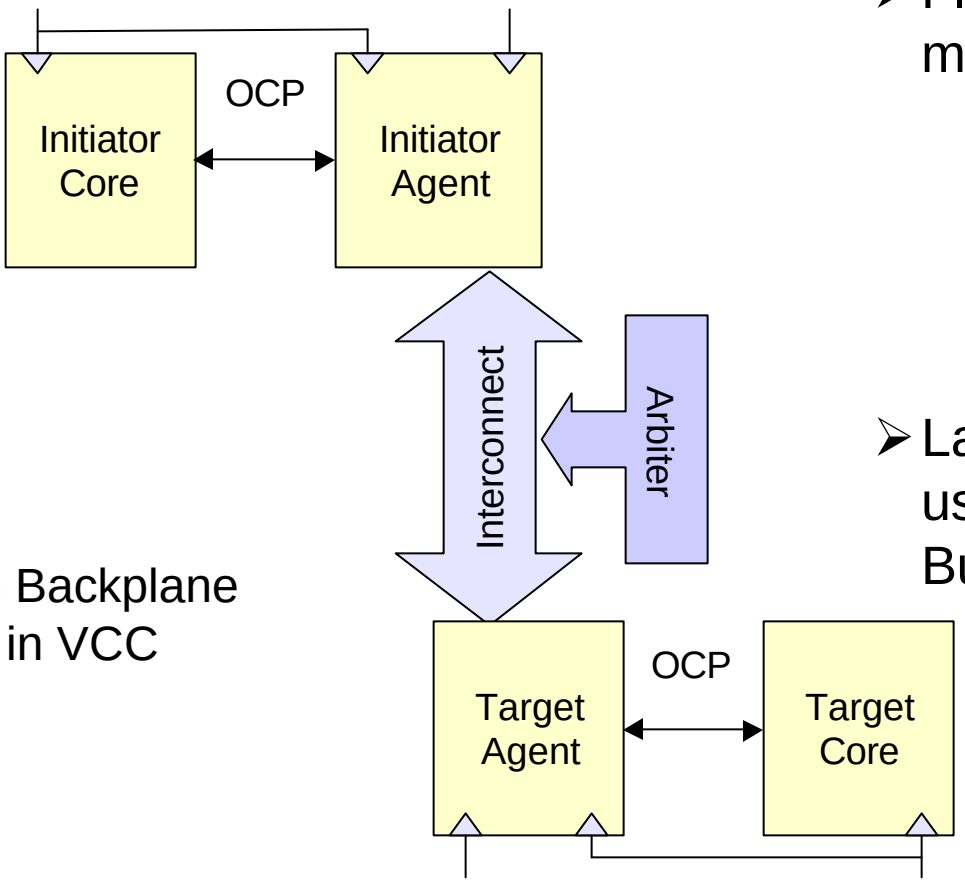


"The Silicon Backplane" (Sonics, Inc)



TDMA Multiple Access Scheme guarantees bandwidth for time-critical links; Combined with contention slots for other communications

Modeling the Impact of Interconnect Choice

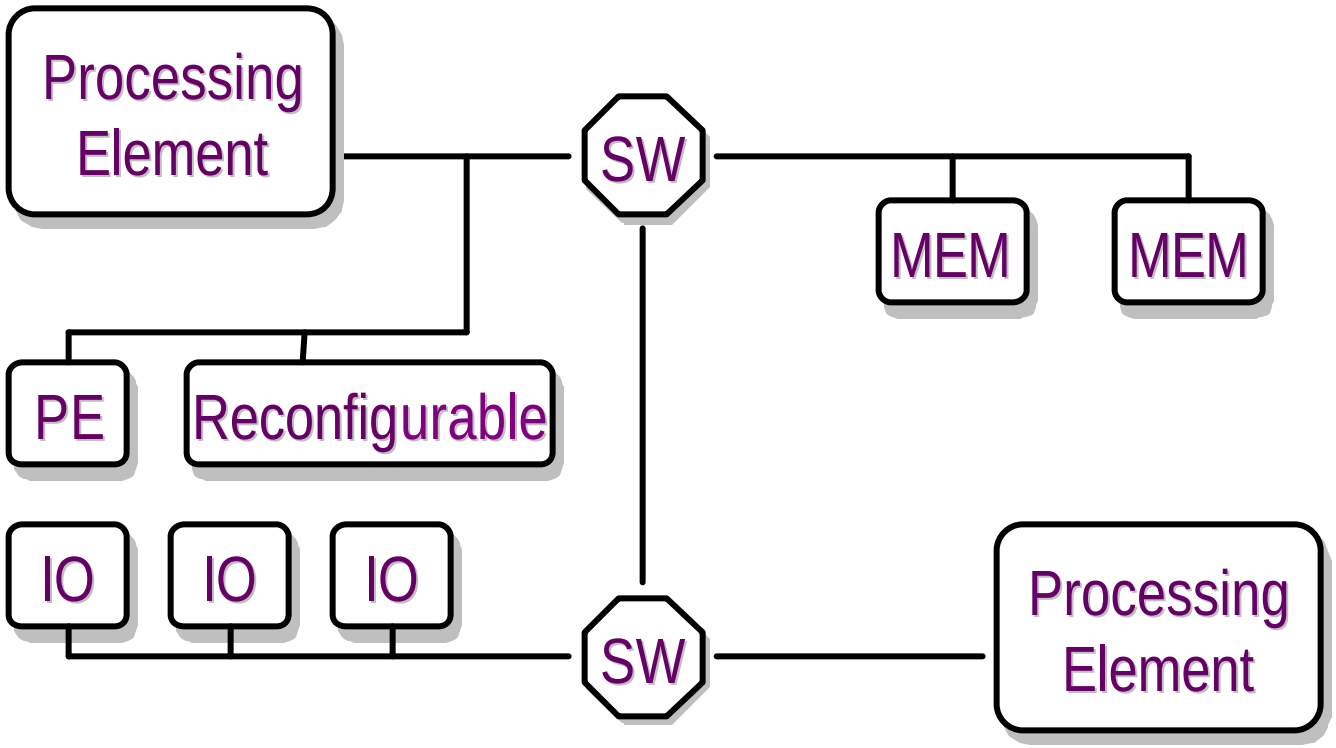


Silicon Backplane Model in VCC

- Flexible bandwidth arbitration model
  - TDMA slot map gives slot owner right of refusal
  - Unowned/unused slots fall to round-robin arbitration
- Latency after slice granted is user-specified between 2-7 Bus Clock cycles

Example: The MESCAL Architecture*

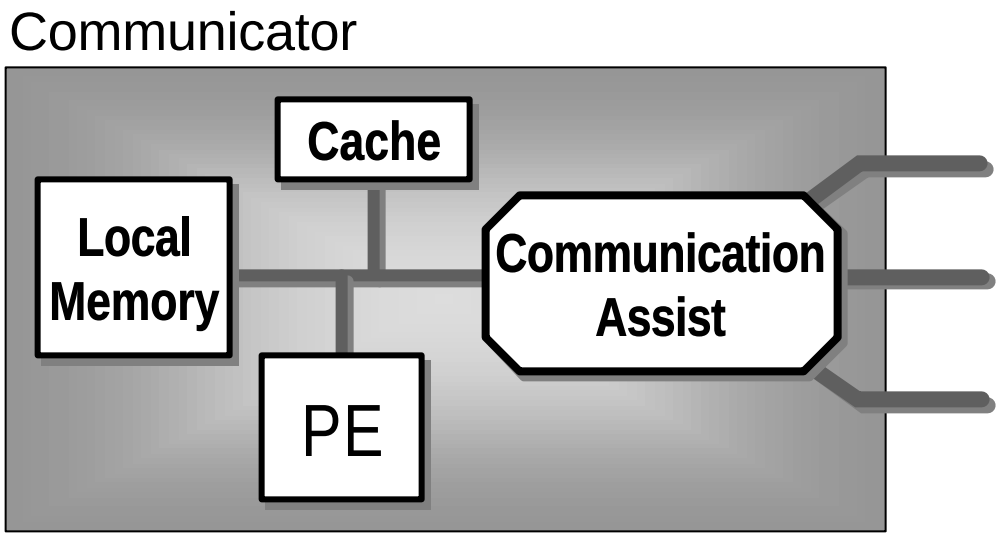
- **A MESCAL Communication Architecture is a general, coarse-grained interconnection scheme for system components**
- **Communicators are Processing Elements, I/Os, Memories, Switches, Reconfigurable fabrics**

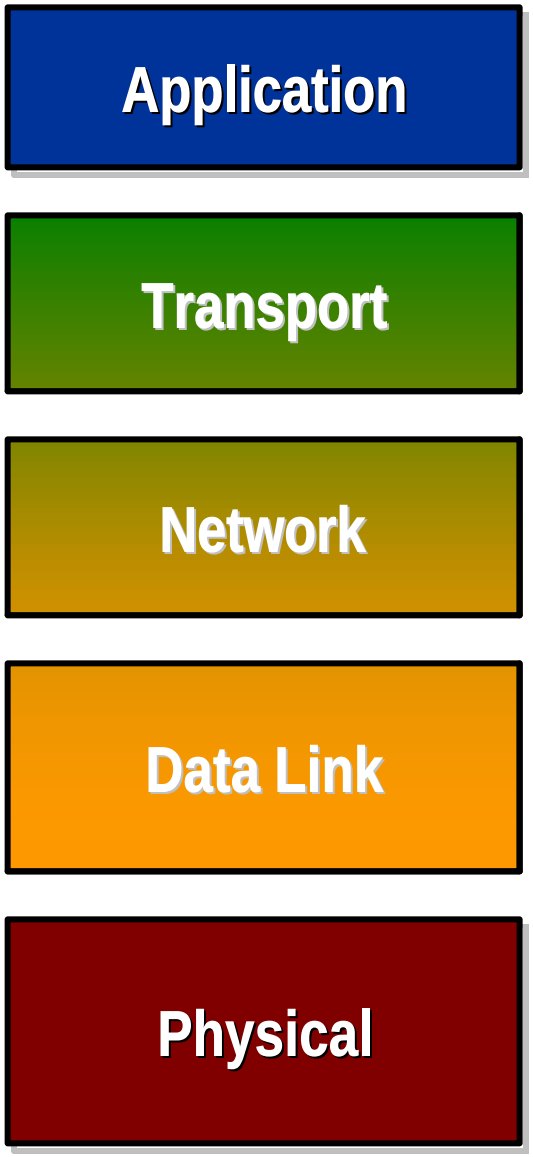


*MESCAL is a GSRC project, targeting full-programmable SOC design

The MESCAL Interconnect Architecture

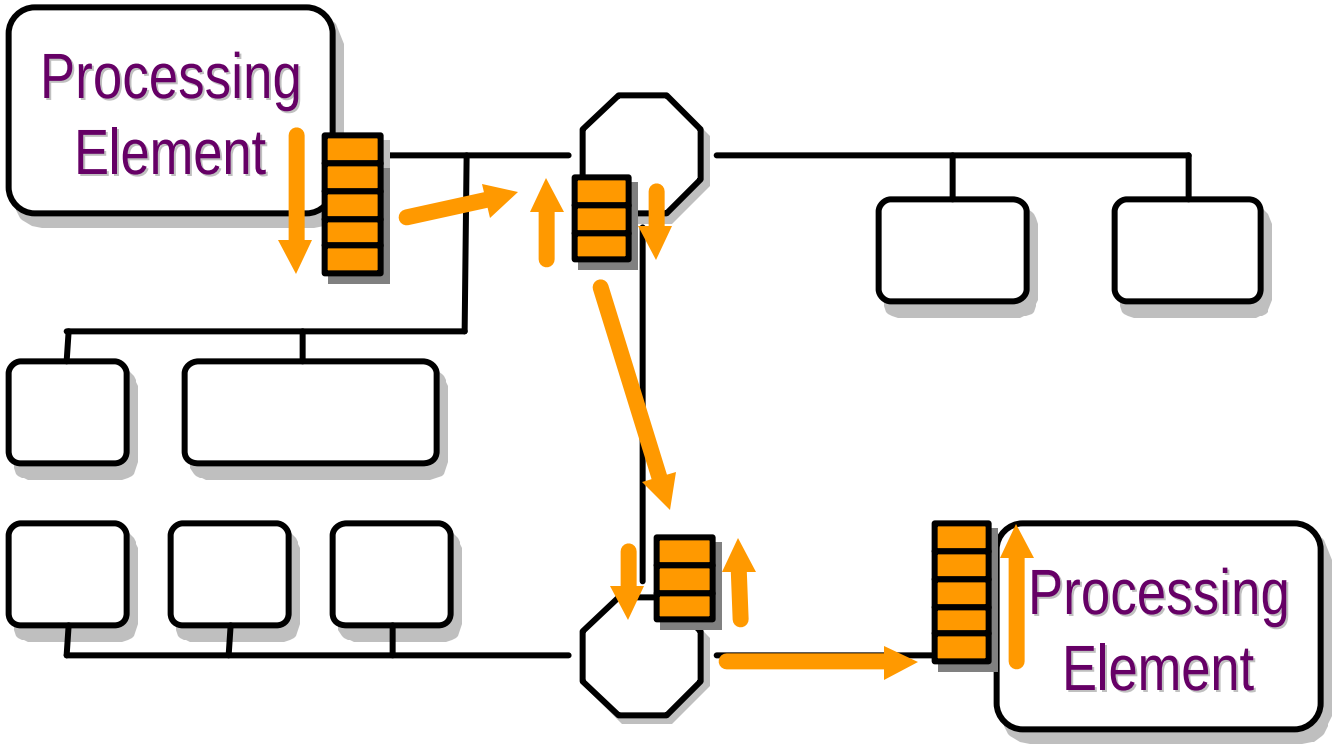
- **Stack layers map to software and hardware components:**
  - PE software
  - CA software
  - CA hardware
  - Channel Hardware





Communication Architecture Design

- **Describe a stack at each node using a formal Ptolemy model**
- **Describe the interconnect topology**
- **Use a correct-by-construction synthesis approach to implement on a programmable platform**



Summary

- **Designing a SOC has become a communications-design problem**
- **Refinement-based formal methodology, inspired by OSI protocol stack, leads to predictable, verifiable and testable solution**
- **Methodology opens the door for innovative solutions to the interconnect problem**
  - Ultra-low swing signals with error-correction and retransmission
  - Data compression for high-rate links
  - Globally asynchronous design
  - Dynamic routing of data (see talk of Bill Dally)