Networks on Chip

Giovanni De Micheli

CSL - Stanford University

Outline

- Introduction and motivation
- Physical limitations of on-chip interconnect
- Communication-centric design
- On-chip networks and protocols
- Software aspects of on-chip networks

Electronic systems




- Systems on chip are everywhere


- Technology advances enable increasingly more complex designs

Challenge: ***how to design complex systems efficiently?***

Component-based design

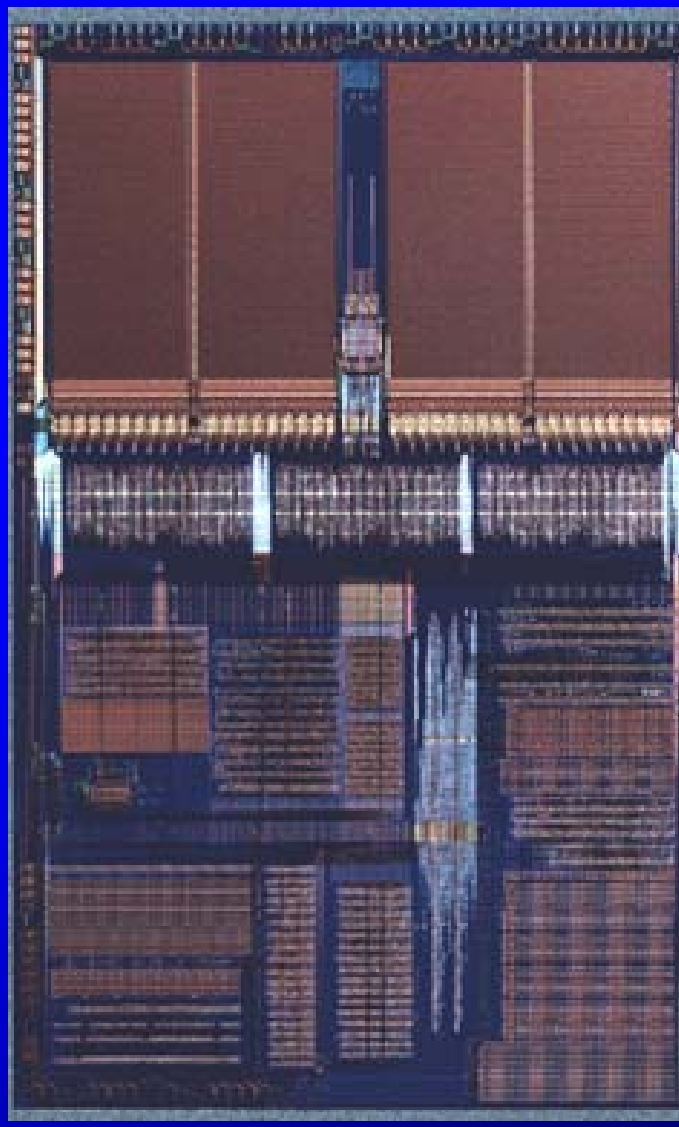
- SoCs are designed (re)-using large macrocells
  - E.g., processors, controllers, memories, …
  - Goal: plug and play methodology
- Design objectives:
  - Provide a functionally-correct, reliable operation of the interconnected components
  - Achieve high quality of service (QoS)
  - Contain energy consumption

Challenge: ***how to connect the components effectively?***

SoC trends

- Computation
  - More, faster blocks
- Storage
  - Most chip area
  - Heavy, unpredictable data traffic
- Communication
  - Speed, energy are critical

SoCs are interconnect dominated



Outline

- Introduction and motivation
- Physical limitations of on-chip interconnect
- Communication-centric design
- On-chip networks and protocols
- Software aspects of on-chip networks

Qualitative roadmap trends

- Continued gate downscaling
- Increased transistor density and frequency
  - Power and thermal management
- Lower supply voltage
  - Reduced noise immunity
- Increased spread of physical parameters
  - Inaccurate modeling of physical behavior

Silicon technology roadmap

Year	Gate length (*nm*)	Transistor density (*million/cm^2*)	Clock rate (*GHz*)	Supply voltage (*V*)
2002	75	48	2.3	1.1
2007	35	154	6.7	0.7
2013	13	617	19.3	0.5

Propagation delays (Example)

- Light speed in vacuum 300 micron/psec
- Chip diagonal
  - 30 mm (edge 22mm)
  - 100 psec
  - 1 clock cycle @ 10GHz
  - 5-10 clock cycles with realistic assumptions
- Wire pipelining
- Delay variation and delay control

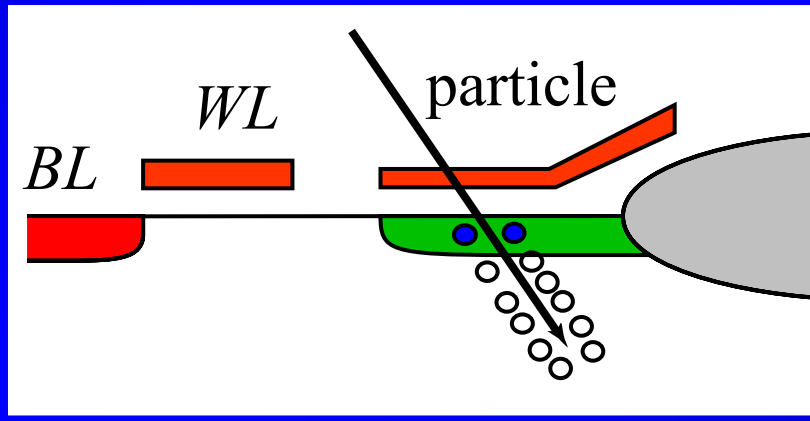
Physical design

- Limitations come from interconnect physics
  - Delay on global wires and delay uncertainty
  - Crosstalk due to capacitive coupling among wires
- Electric signalling techniques
  - Trade-off noise immunity vs. energy vs. speed
  - Sense small swings -> low energy and fast transitions
- Synchronization across large chips
  - Is synchronization possible at high clock rates?
  - What is the probability of synchronization failure?

Reliability of information

- Information transfer is inherently unreliable at the electrical level, due to:
  - Timing errors
  - Cross-talk
  - Electro-magnetic interference (EMI)
  - Soft errors
- The problem will get increasingly more acute as technology scales down

Soft errors



- Due to charge injection:
  - Charged ion:
    - Alpha particle
    - Neutron scattering (from cosmic rays)
- Soft error rate increase with:
  - Environment (e.g., altitude, latitude)
  - Decrease of node critical charge (capacitance*voltage)
- Used to be a problem for DRAMs
  - Now important also for SRAM, registers, etc.

Noise and transient malfunctions

- SoC will operate in presence of noise
  - Data may be delayed or corrupted
  - Malfunctions modeled as single/multiple upsets
- Present design methods reduce noise
  - Physical design (e.g., sizing, routing)
- Future methods must tolerate noise
  - Push solution to higher abstraction levels

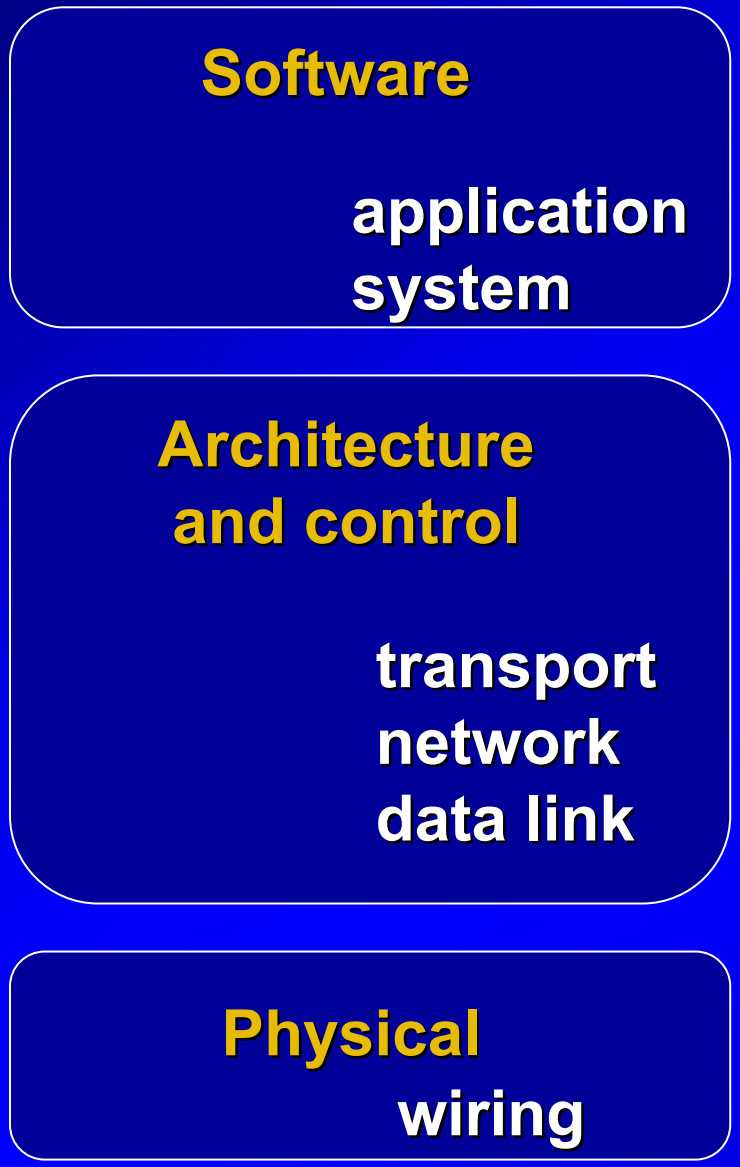
Outline

- Introduction and motivation
- Physical limitations of on-chip interconnect
- Communication-centric design
- On-chip networks and protocols
- Software aspects of on-chip networks

Systems on chips:
a communication-centric view

- Design component interconnection under:
  - Uncertain knowledge of physical medium
  - Incomplete knowledge of data traffic
- Design interconnection as a micro-network
  - Leverage network design technology
  - Manage information flow
    - To provide for performance
  - Power-manage components based on activity
    - To reduce energy consumption

Network Architectures and control



- Architectures
  - Shared medium
  - Direct/indirect
  - Hybrid
- Control protocols
  - Layered
  - Architecture dependent
  - Implemented in Hw or Sw

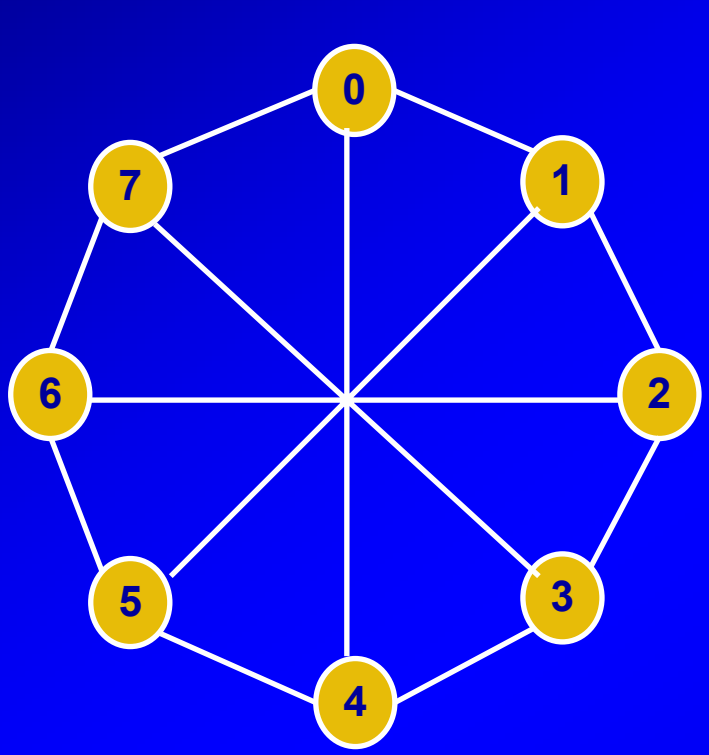
Micro-networks on SoCs

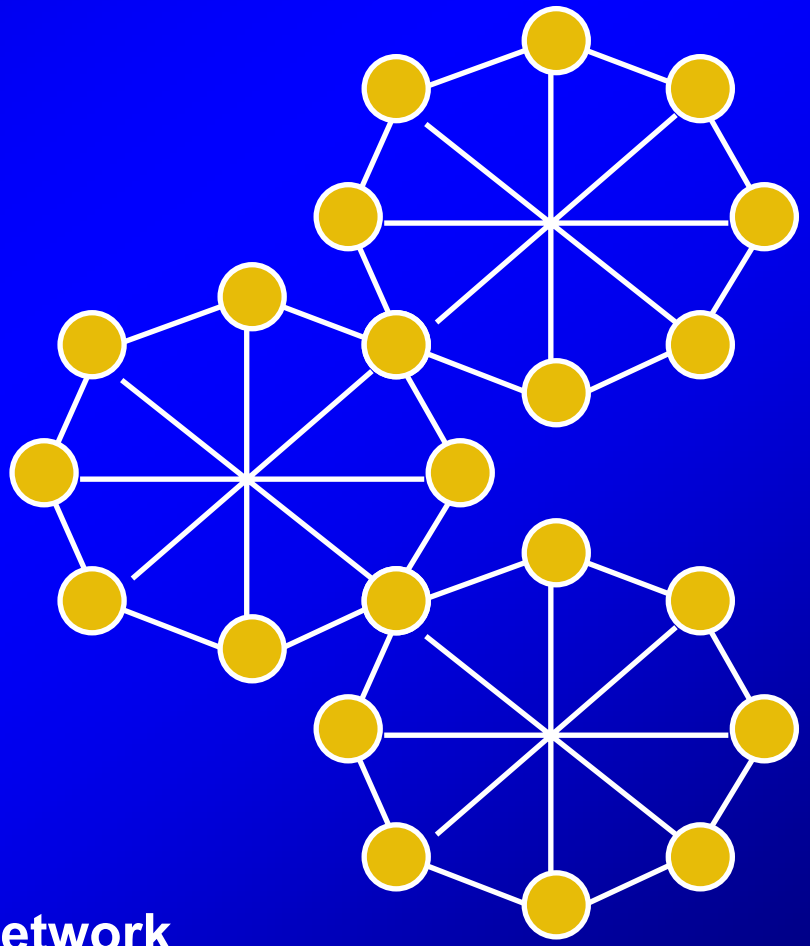
- Multi-processors on a chip
  - Use different network architectures
  - Designed for performance
- Application-specific SoCs
  - Micro-network can be tailored to application
  - Low-energy communication
    - Satisfying Quality of Service (QoS) requirements
      - Performance
      - Reliability
- Field-programmable systems
  - Large-scale, new FPGAs

Example
RAW architecture

- Fully programmable SoC
  - Homogenous array of tiles
    - Computational processing cores with local storage
  - Each tile has a router
- The raw architecture is exposed to the compiler
  - Cores and routers are programmable
  - Compiler determines which wires are used at each cycle
  - Compiler pipelines long wires
- Direct network over a homogeneous fabric

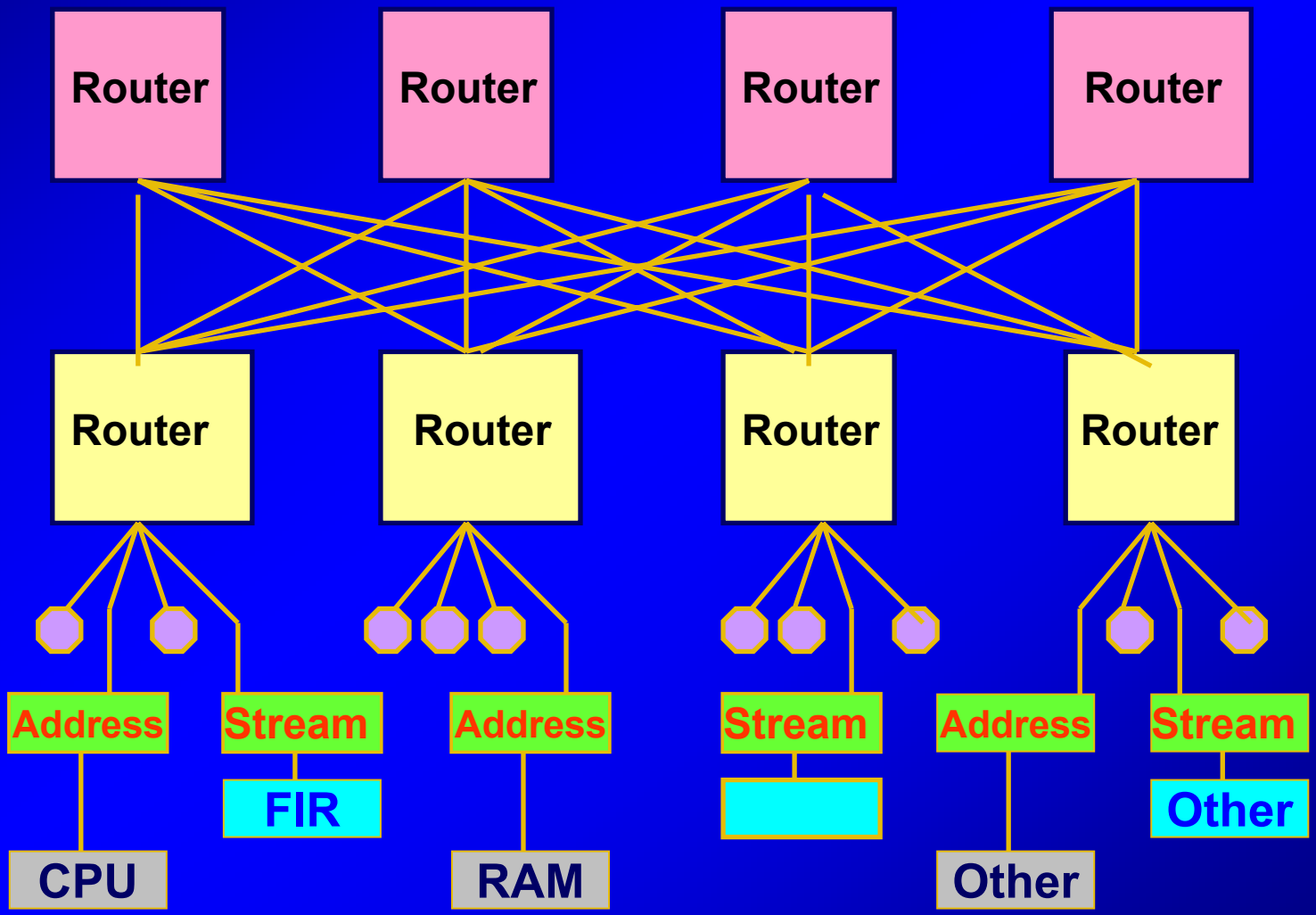
Example of direct micro-network STM *Octagon*





The network is scaleable:
a node can be a port processor
or an interface node to another network

Example of indirect micro-network SPIN



Examples FPGAs

- Example 1: Xilinx Spartan II
- Indirect network over homogeneous fabric
  - Many simple processing elements
  - CLBs interconnected by switches
- Example 2: Xilinx Virtex II
- Indirect network over heterogeneous fabric
  - Elements are CLBs, RAMs, multipliers and clock managers interconnected by switches

Network design objectives

- Low communication latency
  - Streamlined control protocols
- High communication bandwidth
  - To support demanding SW applications
- Low energy consumption
  - Wiring switched capacitance dominates
- High system-level reliability
  - Correct communication errors, data loss

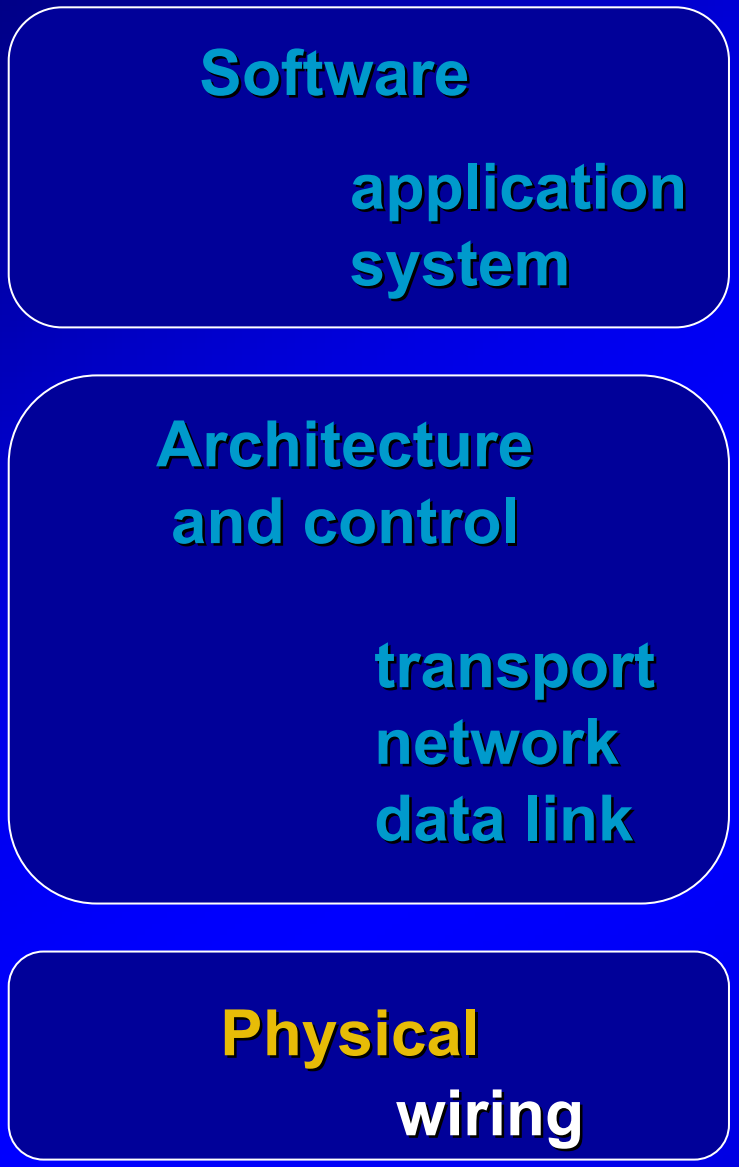
Guaranteed vs Best-Effort Service

- Guaranteed service
  - QoS of communication is guaranteed
  - Necessary for real-time communication
    - E.g. 4Mb/s for an MPEG video stream
  - Resource utilization may be low
- Best-effort service
  - QoS of communication is not guaranteed
  - Conventional *priority-based on-chip buses* offer best-effort service
  - Suitable for non-real time communication
  - Communication resources are shared
    - Use idle communication resources
    - Good utilization of communication resource

Outline

- Introduction and motivation
- Physical limitations of on-chip interconnect
- Communication-centric design
- On-chip networks and protocols
- Software aspects of on-chip networks

Physical layer



Physical design:

- Voltage levels
- Driver design
- Sizing
- Physical routing

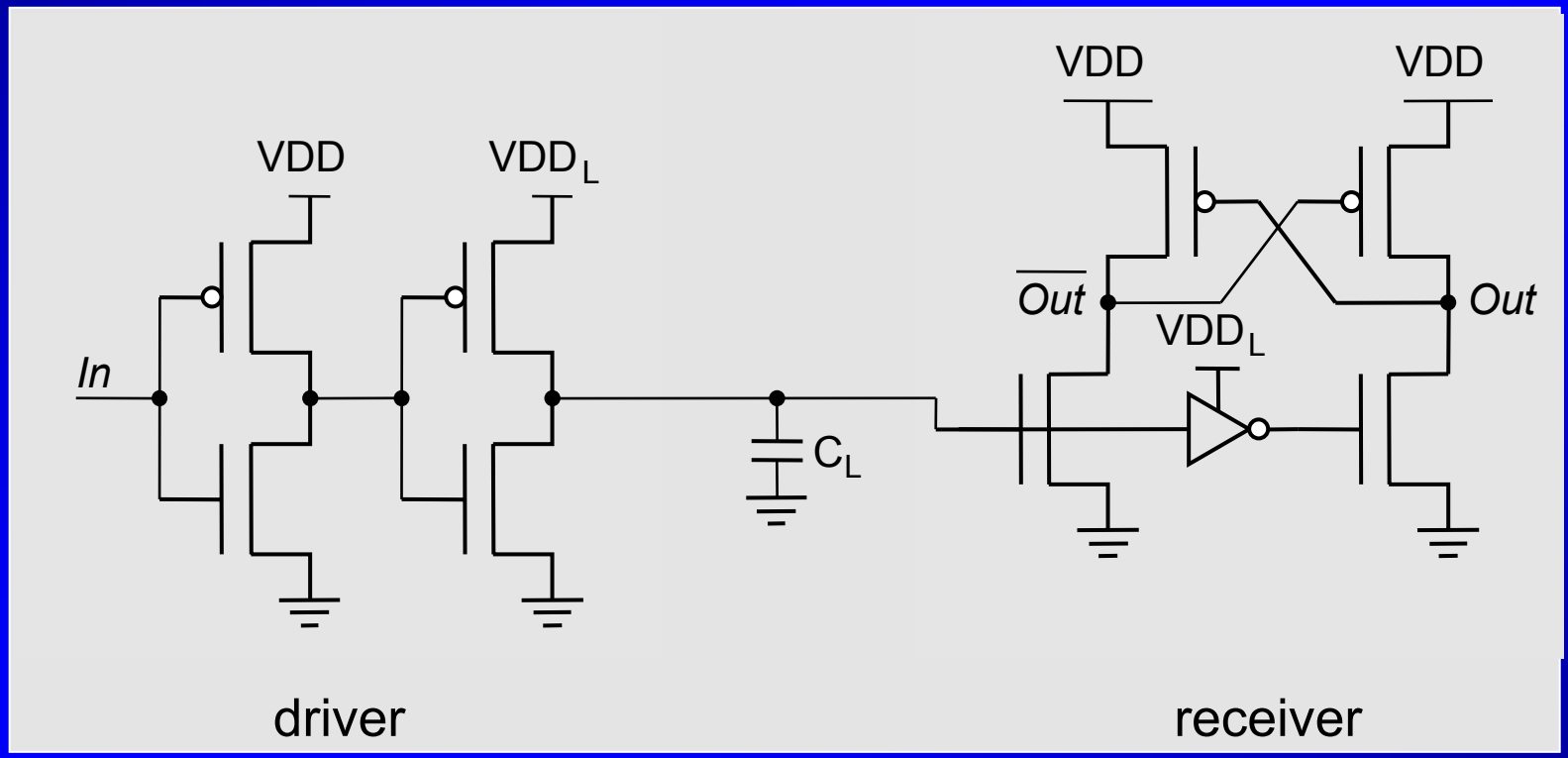
Physical layer design

- Multiple-voltage, dynamic voltage supplies
  - Multiple, adjustable threshold voltages
- Bus/wire drivers:
  - Large/small driver swings
  - Receivers with sense-amplifiers
- Device and wire sizing
  - Tapered wires
- Physical routing
  - Cross-talk avoidance

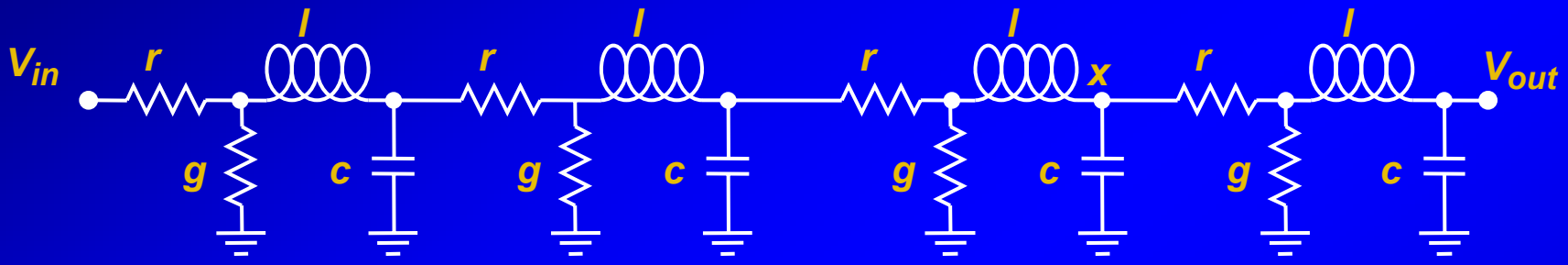
Reduced swing transmission

- Sensing small swings improves performance
  - As in the case of RAMs
- Propagation delay is insensitive to supply voltage
  - Small delay reduction for increased supply
- Power dissipation decreases with voltage swing
- Transmission can be single-ended or differential
  - Many possible schemes

Single-Ended Static Driver and Receiver



Coping with inductive effects



- Transmission-line effects:
  - Considered when rise/fall time of input signal are smaller than the time of flight on the transmission line
  - Total wire resistance much less than 5 Z_0
- Design considerations
  - Use matched termination to avoid reflection

Limitations of physical design

- Large scale design:
  - 1 billion transistors
- Small scale features
  - Sub 100 nm geometric features
- Nearly impossible to generate ideal layout
  - Too difficult to find hot spots
  - Design and timing closure problem

Challenge: make imperfect layout to work by making corrections at higher levels

Architecture and control

Software

application
system

Architecture and control

transport
network
data link

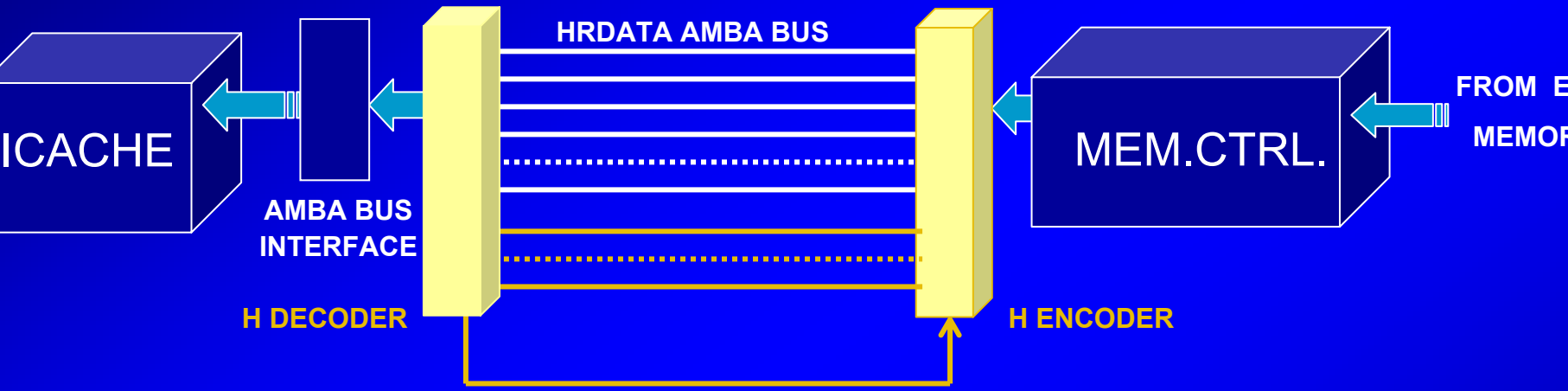
Physical

wiring

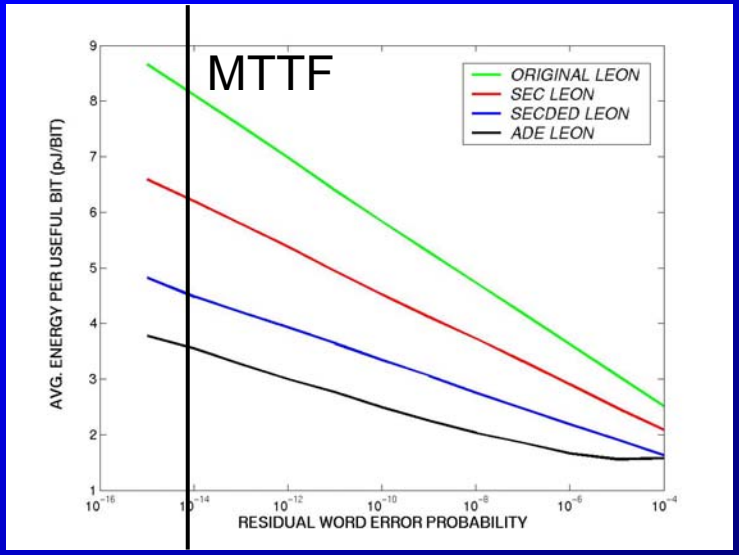
Data link layer

- Provide reliable data transfer on an unreliable physical channel
- Access to the communication medium
  - Dealing with contention and arbitration
- Issues
  - Fairness and safe communication
  - Achieve high throughput
  - Error resiliency

Data-link protocol example: error-resilient coding

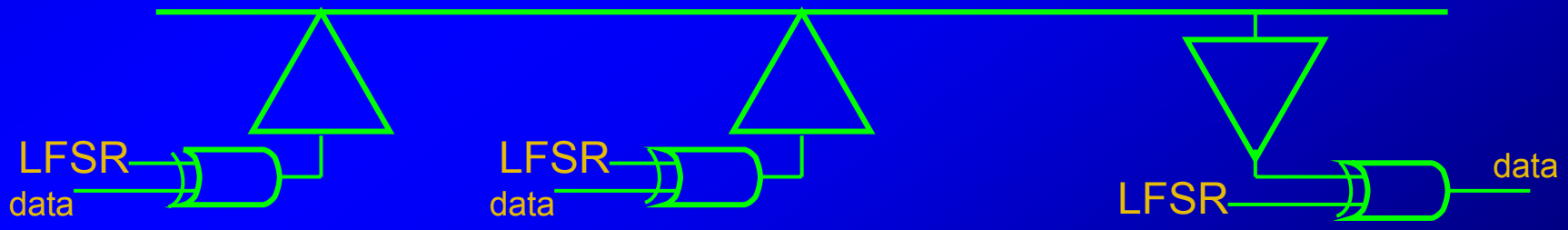


- Compare original AMBA bus to extended bus with error detection and correction or retransmission
  - SEC coding
  - SEC-DED coding
  - ED coding
- Explore energy efficiency



Advanced bus techniques: CDMA on bus

- Motivation: many data sources
  - Support multiple concurrent write on bus
  - Discriminate against background noise
- Spread spectrum of information
  - Driver/receiver multiply data by random sequence generated by LFSR
    - LFSR signature is key for de-spreading



Going beyond buses

Buses:

- Pro: simple, existing standards
- Contra: performance, energy-efficiency, arbitration

Other network topologies:

- Pro: higher performance, experience with MP
- Contra: physical routing, need for network and transport layers

Challenge: exploit appropriate network architecture and corresponding protocols

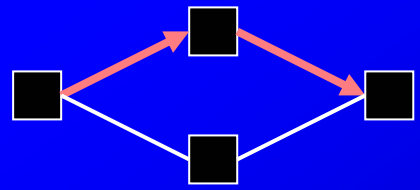
Network layer

- Network switching
  - Connection-oriented switching
    - A path from source to destination is reserved prior to communication
    - Useful for traffic with infrequent and long messages
    - Circuit switching, virtual circuits
  - Connection-less switching
    - The communication path is determined dynamically
    - Datagram
- Network routing
  - Unicast, multicast
  - Source routing, distributed routing
  - Deterministic, adaptive

Challenge: which models and what parameter values are effective for micro-networks?

Connection-oriented schemes

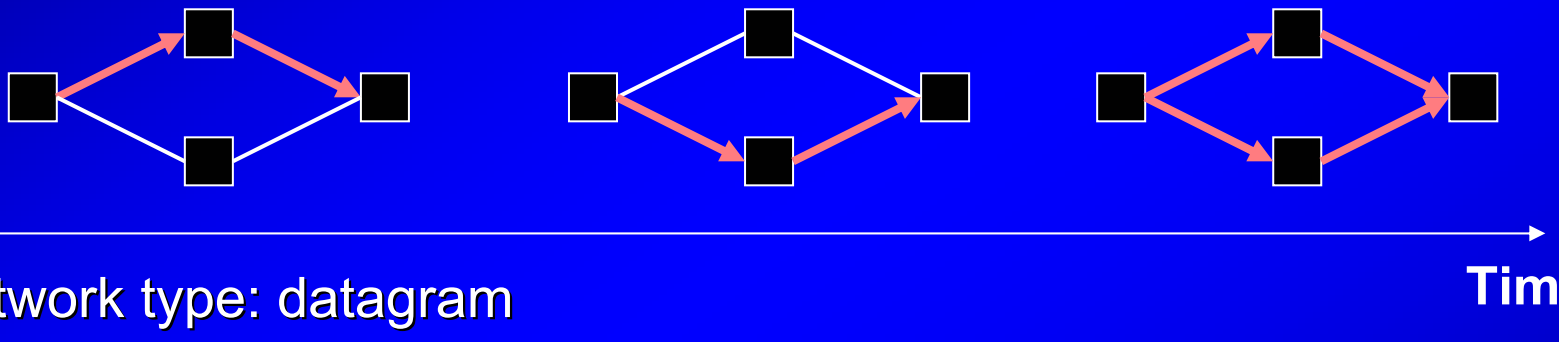
- Communication path is fixed before data transmission starts
- Network types: circuit switch, virtual circuit
  - Basic operations
    - Connection setup
    - Data transfer
    - Connection teardown



- Advantages
  - QoS (e.g. bandwidth, delay, jitter) guarantee through resource reservation over the fixed path
  - Suited to real-time, constant BW communication
- Disadvantages
  - Resource utilization is worse than connection-less communication (i.e. datagram).
  - Connection setup overhead

Connection-less communication

- Depending on network traffic, the communication path is determined dynamically during data (packet) transmission



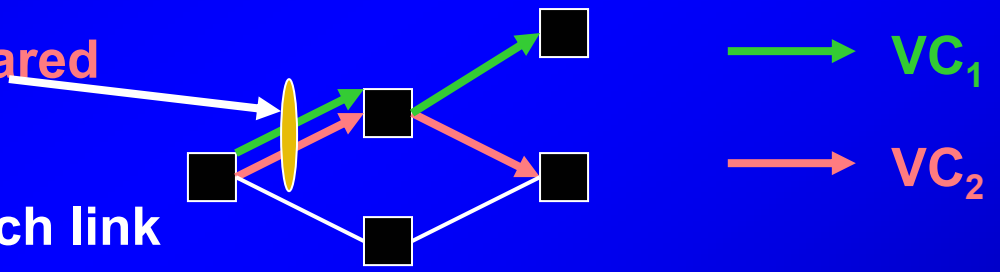
- Network type: datagram
- Advantages
  - Better adaptation of communication to the varying network traffic
  - Better utilization of network resource
  - Suited to variable bit rate communication
    - (e.g. encoded voice, MPEG2,4, etc.)
- Disadvantage
    - Poor QoS support (← no resource reservation)

Packet-based communication Virtual circuit

- Fixed end-to-end communication path
  - Packets are transferred over the VC

Each link can be shared by several VC's

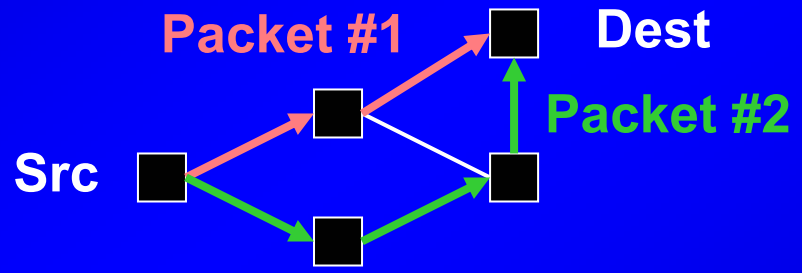
* **Multiplexing on each link**



- QoS guarantee:
  - Through resource reservation when the connection is set up

Packet-based communication datagram

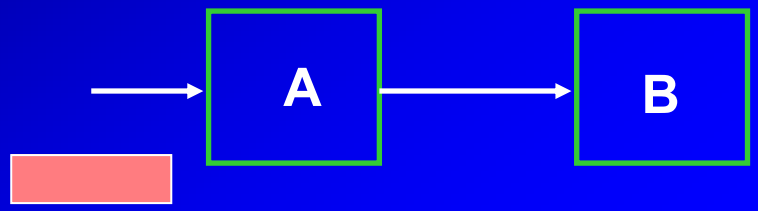
- Connection-less
- Routers route packets independently.
  - Packets can take any paths to the destination.



  - Routers manage routing tables to find paths to any destinations
- Non-deterministic communication delay due to communication resource (buffer and link) contention
- No QoS guarantee!
- Flow and congestion control is needed

Packet Forwarding Schemes

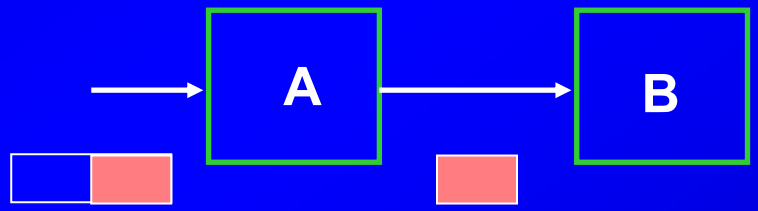
- Store-and-forward



(1) After A finishes receiving the entire packet, A asks B if B is ready for the entire packet.
(2) B → A, ack
(3) A sends the packet to B.

The same buffer space is needed

- Virtual-cut-through

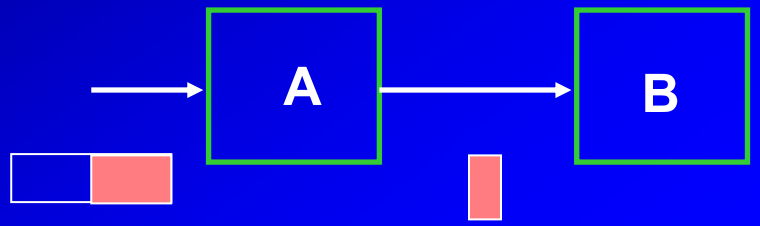


Pipelining!

(1) While A receives a part of the packet, A asks B if B is ready for the entire packet.
(2) B → A, ack
(3) A starts to send the packet to B even when A has not yet received the entire packet.

Packet Forwarding Schemes

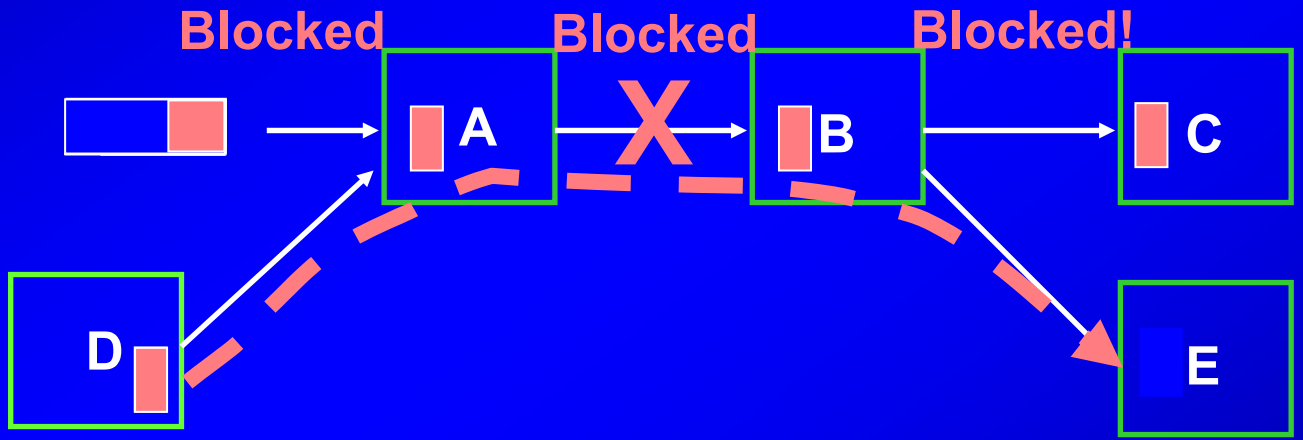
- Wormhole



Pipelining
on a flit (flow control unit
basis

(1) After A receives a flit of the packet,
A asks B if B is ready to receive a flit
(2) B → A, ack
(3) A sends a flit to B.

flit size < packet size
Smaller data space
is needed than
store-and-forward



C cannot send it
and has no enough
space for a new flit

Head-of-line
blocking problem

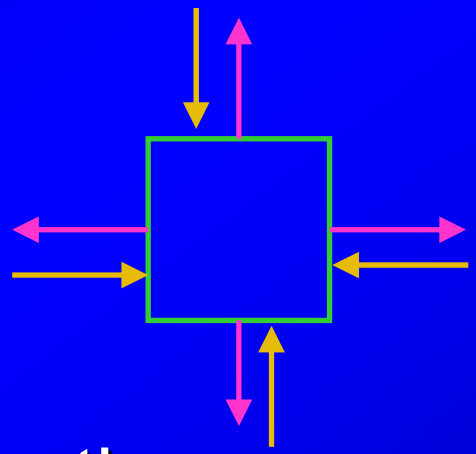
Packets cannot pass from D to E
due to the blocked link between A and B

Example
SPIN Micro-network

- 36-bit packets
  - header: destination
  - trailer: checksum
- Fat-tree network architecture
- Wormhole switching
- Credit-based control flow

Example
Nostrum network

- Mesh network topology
- Memoryless switch
- Packet-based switching
- Contention-aware deflection routing (*hot potato*)
  - Local information used to distribute packets and avoid hot spots
  - Effective in removing contention at network center



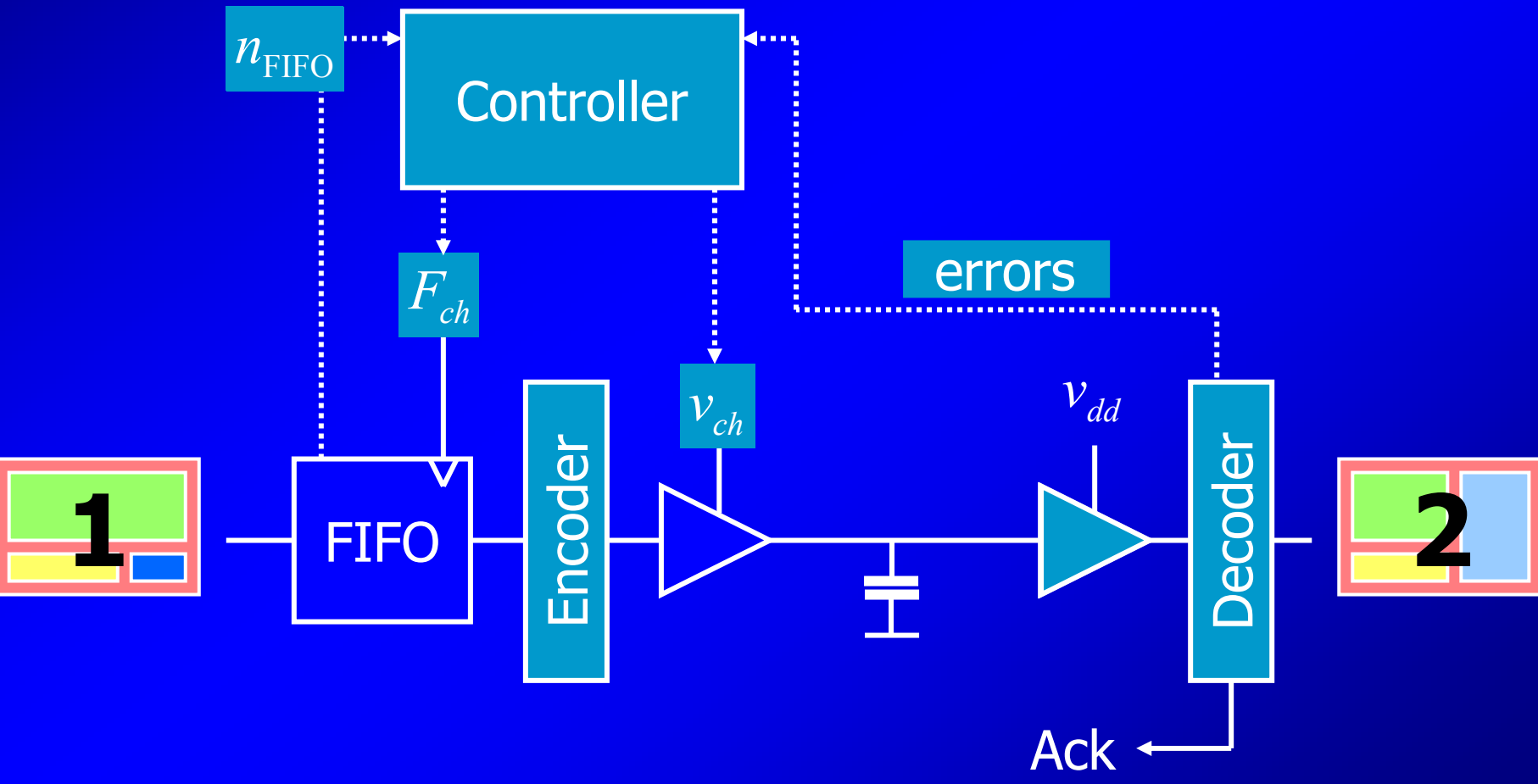
Transport layer

- Decompose and reconstruct information
- Important choices
  - Packet granularity
  - Admission/congestion control
  - Packet retransmission parameters
- All these factors affect heavily energy and performance
- Application-specific schemes vs. standards

Benefits of packets

- Reliable error-control mechanism
  - With small overhead
- Exploit different routing paths
  - Spread information to avoid congestion
- Several user-controllable parameters
  - Size, retransmission schemes, …
- Use retransmission rate for calibrating parameters

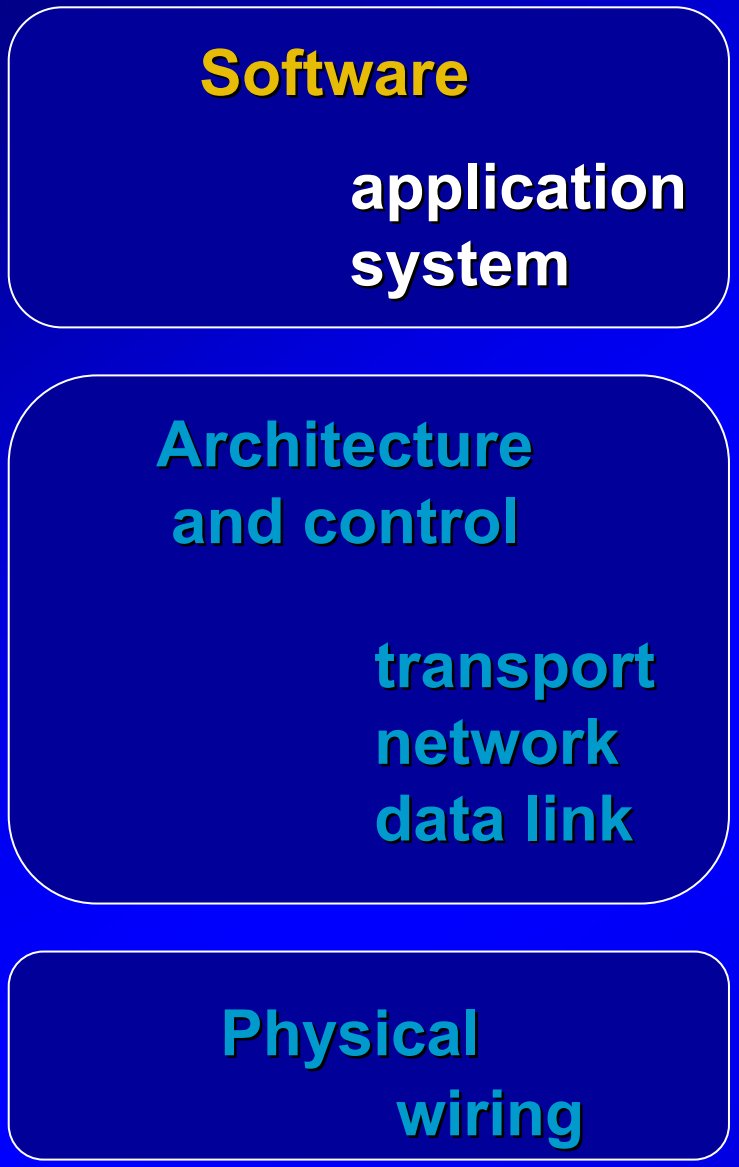
Adaptive Low-Power Transmission Scheme



Outline

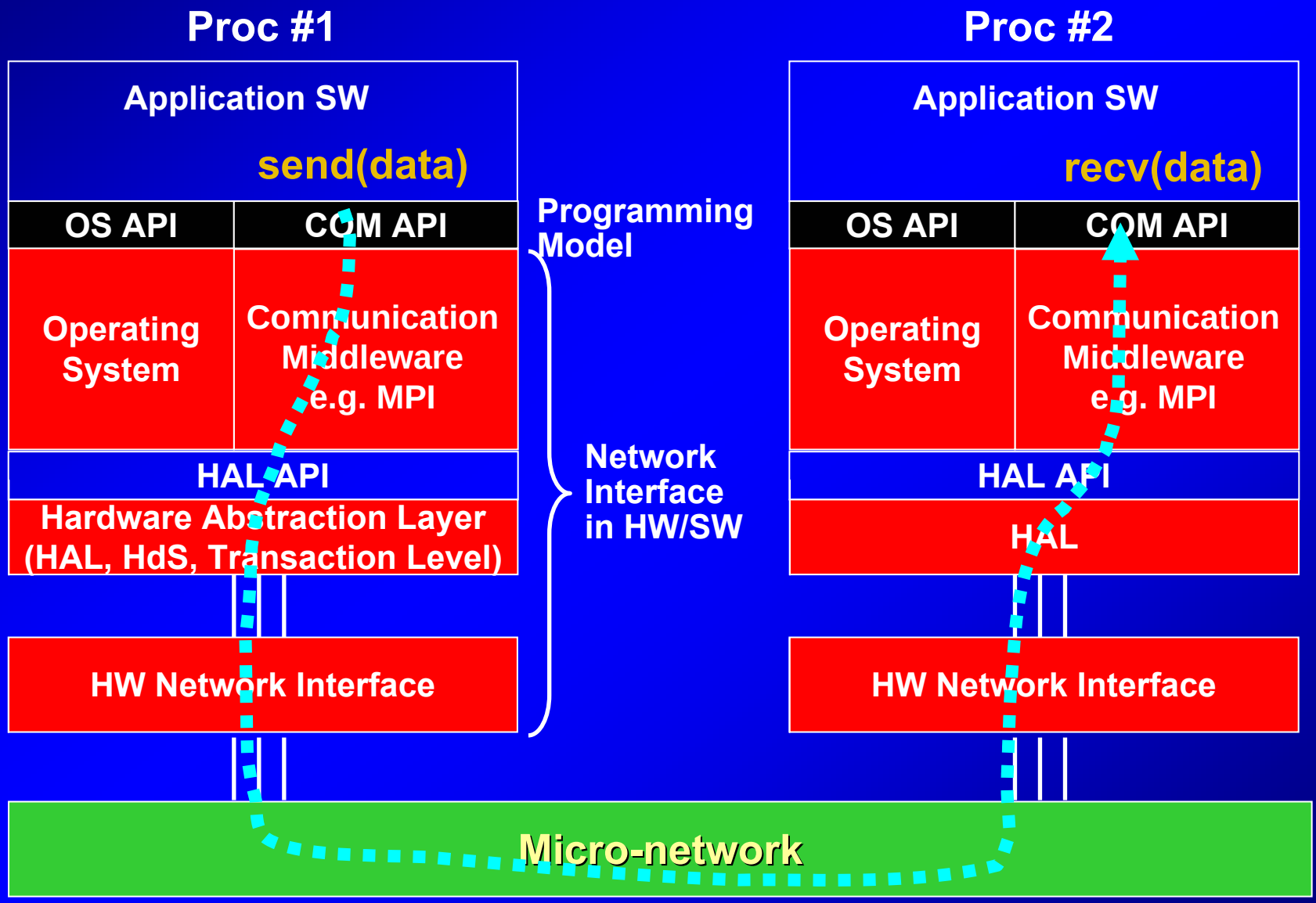
- Introduction and motivation
- Physical limitations of on-chip interconnect
- Communication-centric design
- On-chip networks and protocols
- Software aspects of on-chip networks

Software layers



- System software
  - OS, RTOS, run-time scheduler
  - Component and network dependent
- Application software
  - User and standard applications

System view of communication



Programming models

- Abstraction:
  - Hides hardware complexity
  - Enhances code longevity and portability
- Programming paradigms
  - Shared memory
    - Parallel tasks access shared memory locations
  - Message passing
    - Tasks have separate address spaces and communication is done via explicit messages

Programming paradigms

- Shared memory
  - Easier to write code
  - More hw needed to support high performance
- Message passing
  - Scalable
  - Makes communication needs explicit
  - More predictable

Message passing is likely to be the paradigm of choice for programming networked SoCs

Middleware

- Communication middleware is the software layer between sw programs and communication hw
- Requirements:
  - Provide safe and balanced resource access
  - Scheduling communication
  - Task synchronization
  - Management of peripherals
- Modeling and abstraction
  - Sw applications communicate via APIs
  - Hw abstraction layer (HAL) is the hw view to the programmer

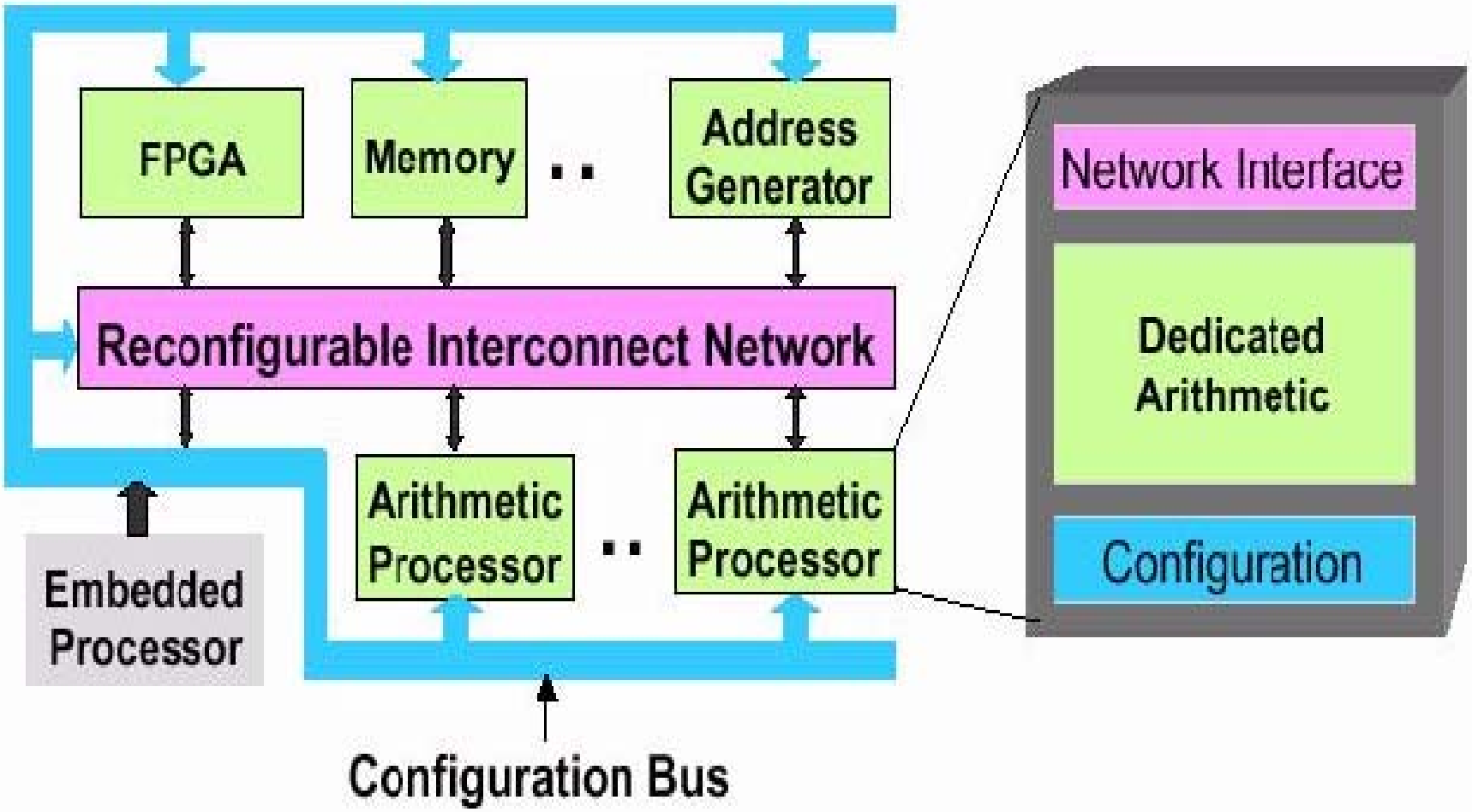
Middleware design challenges

- Trade off hw and sw support for communication
  - Hw/Sw co-design issue
- Balance *guaranteed* and *best-effort* services
- Support high-throughput, low-latency communication
  - Few sw layers
  - Low-overhead transactions
- Simple, scaleable, portable, robust, …

Software control of networked SoCs

- Exploit widely-varying loads on components and on communication links
- Power-manageable components:
  - Dynamic voltage scaling (DVS)
    - Adjust frequency and voltage
  - Dynamic power management (DPM)
    - Set idle components into sleep states
- Dynamic information-flow management
  - Reconfigure network and protocols dynamically

Dynamic network-flow management (Pleiades)



Application layer

- Given a platform, the performance and energy to realize a function depends on software
  - Different algorithms to embody a function (e.g., sorting)
  - Different coding styles
  - Different instruction streams (e.g. assembly)
- Software production tools need to address both performance and energy goals

Application software development tools

- Software synthesis
  - Source-code generation
  - Source-level optimizing transformations
- Application-specific compilation
  - Choice of instructions, registers, schedule
- Software design tools need network awareness to be effective
  - Balance computation, storage and communication

Summary
Problem analysis

- Electronic embedded systems require SoCs with high QoS and low energy consumption
- The challenge of SoC design is in interconnecting high-level components
- Design has to cope with non-determinism
  - High-level abstraction
  - Physical properties of material
- The physical interconnection is unreliable

Summary
Design strategies

- Reliable communication is achieved by layered design methods:
  - Learn from network and MP design
  - For application-specific SoCs, the network and protocols can be tailored to the application
- Encoding, packet switching and routing provide a new view of interconnect design
- The system and application software design are key to manage components and networks

To probe further

- Ackland et al., A Single Chip, 1.6 Billion, 16b MAC/s Multiprocessor DSP, IEEE JSSC, March 2000
- Agrawal, Raw Computation, Scientific American, August 1999
- Benini and De Micheli, Networks on Chip: A New SoC Paradigm, IEEE Computer, January 2002
- Benini and De Micheli, Powering Networks on Chip, Proceedings ISSS, October 2001
- Bertozzi, Benini and De Micheli, Low-Power Error-Resilient Codes for On-Chip Data Busses, DATE 2002
- Dally and Towles, Route Packets, not Wires, DAC 2001
- Guerrier and Grenier, A Generic Architecture for On-Chip Packet Switched Interconnections, DATE 2000
- Ho, Mai and Horowitz, The Future of Wires, IEEE Proceedings, January 2001
- Hu and Marculescu, Energy Aware Mapping for Tile-Based NoC Architectures, ASPDAC 2003
- Rijpkema et al., Trade off in the Design of a Router with Both Guaranteed and Best-Effort Services for Networks on Chip, DATE 2003
- Yoshimura et al., DS-CDMA Wired Bus with Simple Interconnection Topology for Parallel Processing System LSIs, ISSC 2000
- Worm, Ienne, Thiran and De Micheli, An Adaptive, Low-Power Transmission scheme for On-Chip Networks, ISSS 2002
- Ye, De Micheli, Benini, Packetized On-chip Interconnect Communication Analysis for MPSoCs, DATE 2003
- Zhang et al., A 1V Heterogeneous Reconfigurable DSP IC for Wireless Baseband Digital Signal Processing, JSSC, November 2000


Thank You