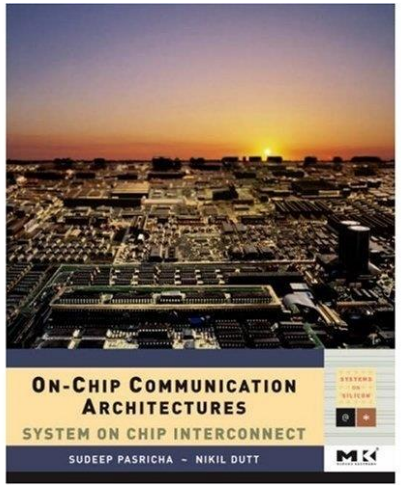
Topic 3

SoC Interconnect Architecture I*

Xinfei Guo
xinfei.guo@sjtu.edu.cn

September 30th, 2024

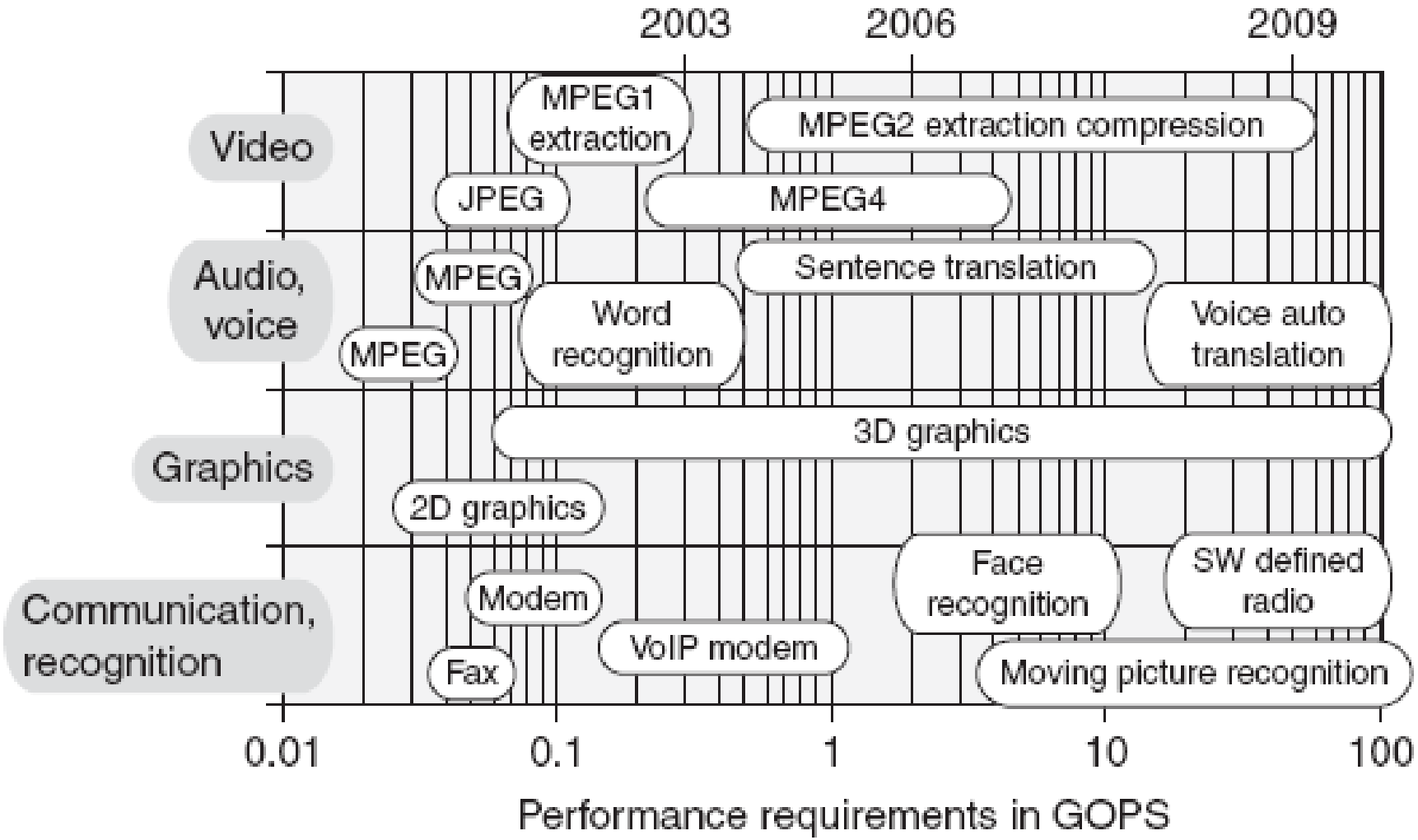


*Many materials are adapted from *"On-Chip Communication Architectures" by Sudeep Pasricha & Nikil Dut, Morgan Kaufmann, 2008*

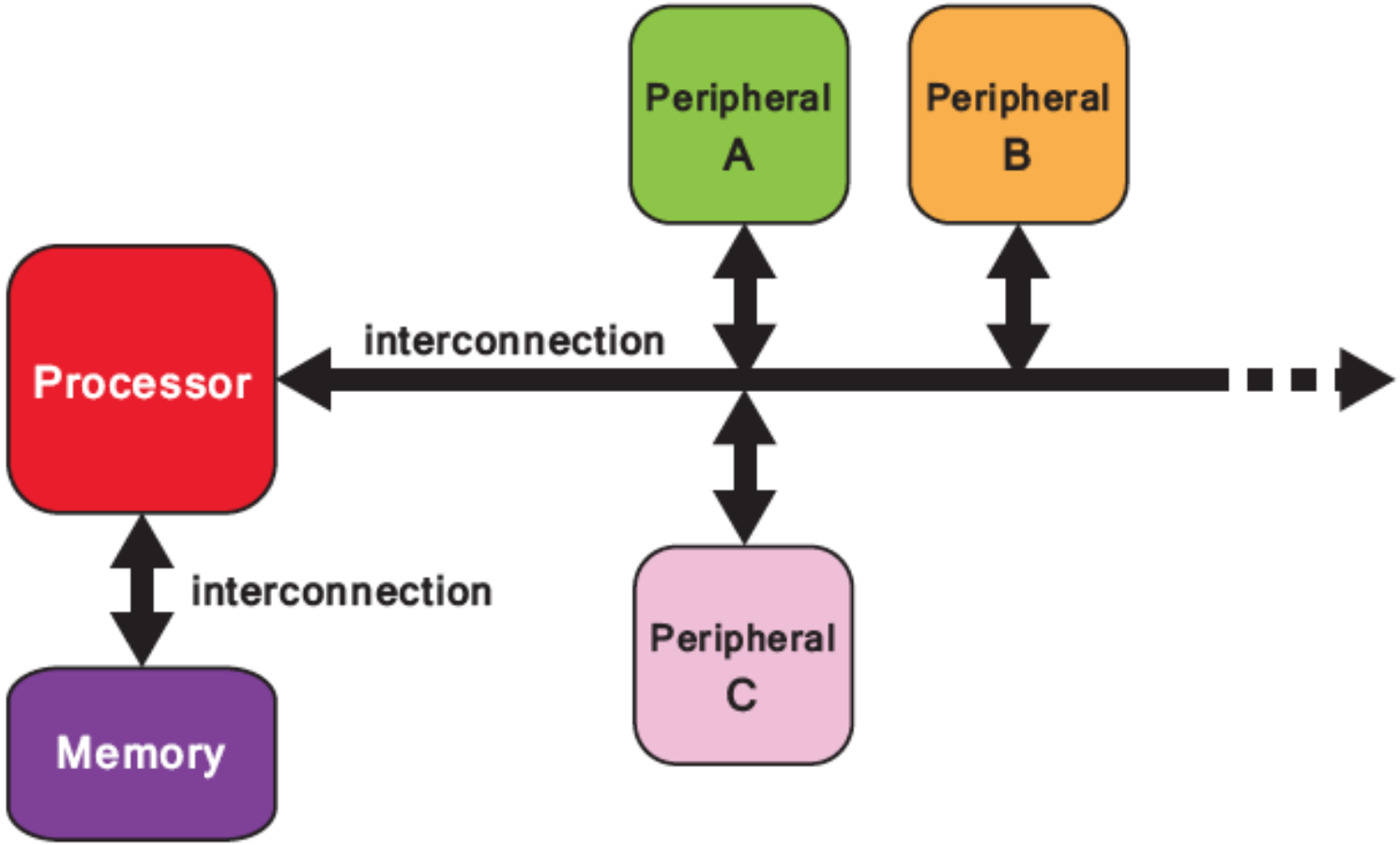
T3 learning goals

- How SoC components are connected together…
  - I. Introduction & Motivation
  - II. Bus
  - III. Network on chip (NoC)

Emerging Application Requirements

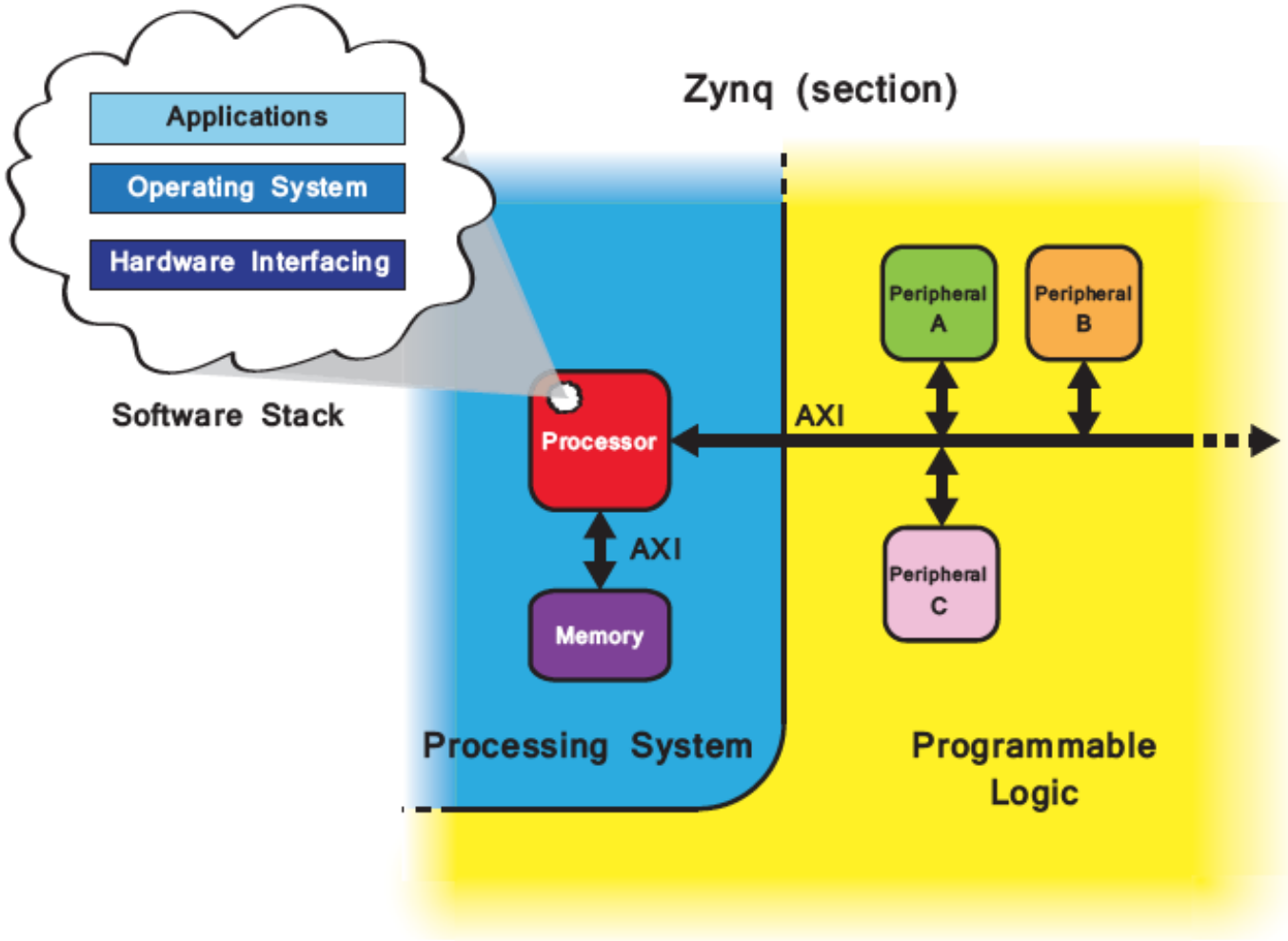


SoC Architecture: Interconnect



Source: The Zynq Book

Interconnect on Zynq SoC



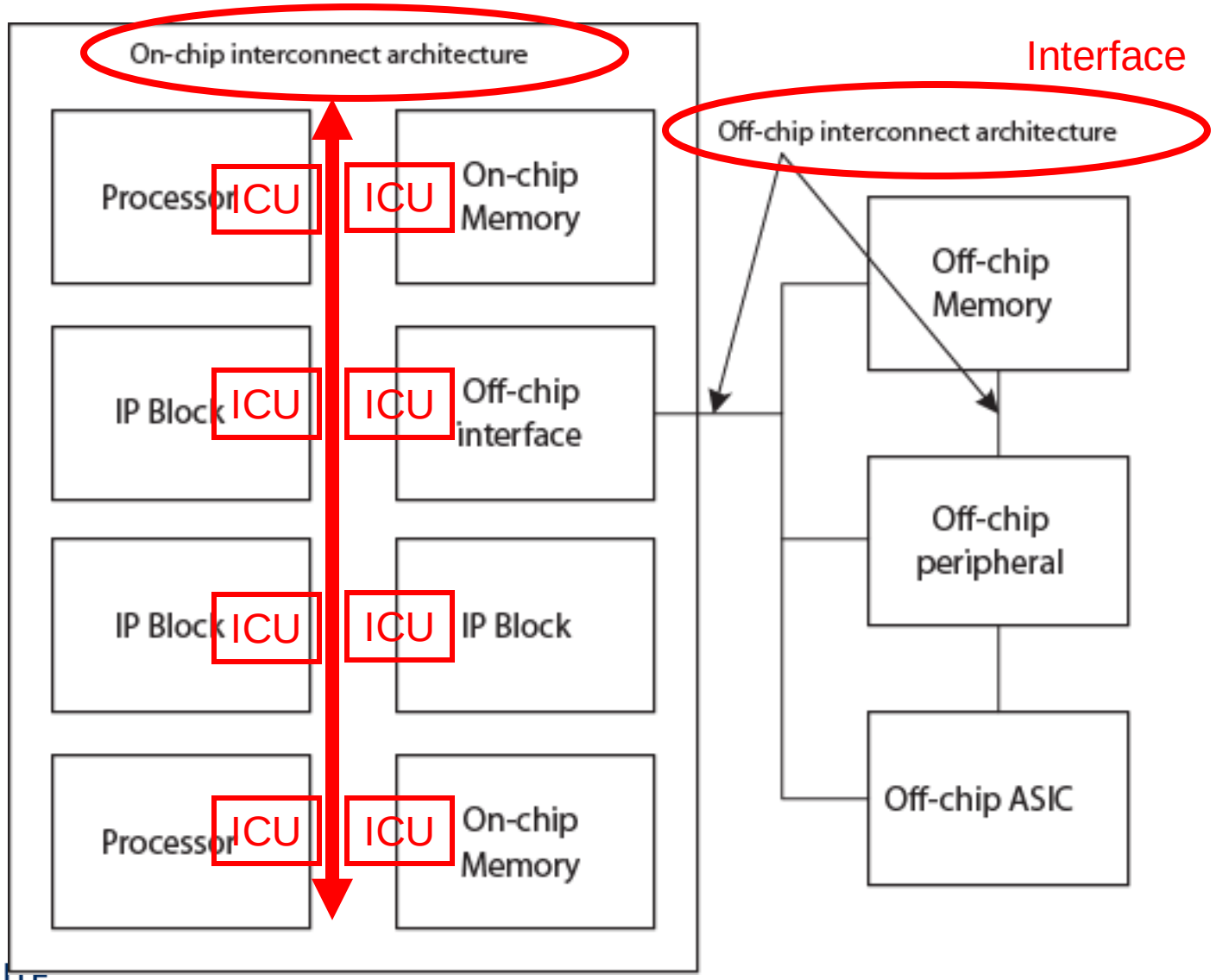
Source: The Zynq Book

Interconnect: An Analogy

highway interchange in Chongqing

source: https://k.sina.cn/article_6433573200_17f788950001004pux.html

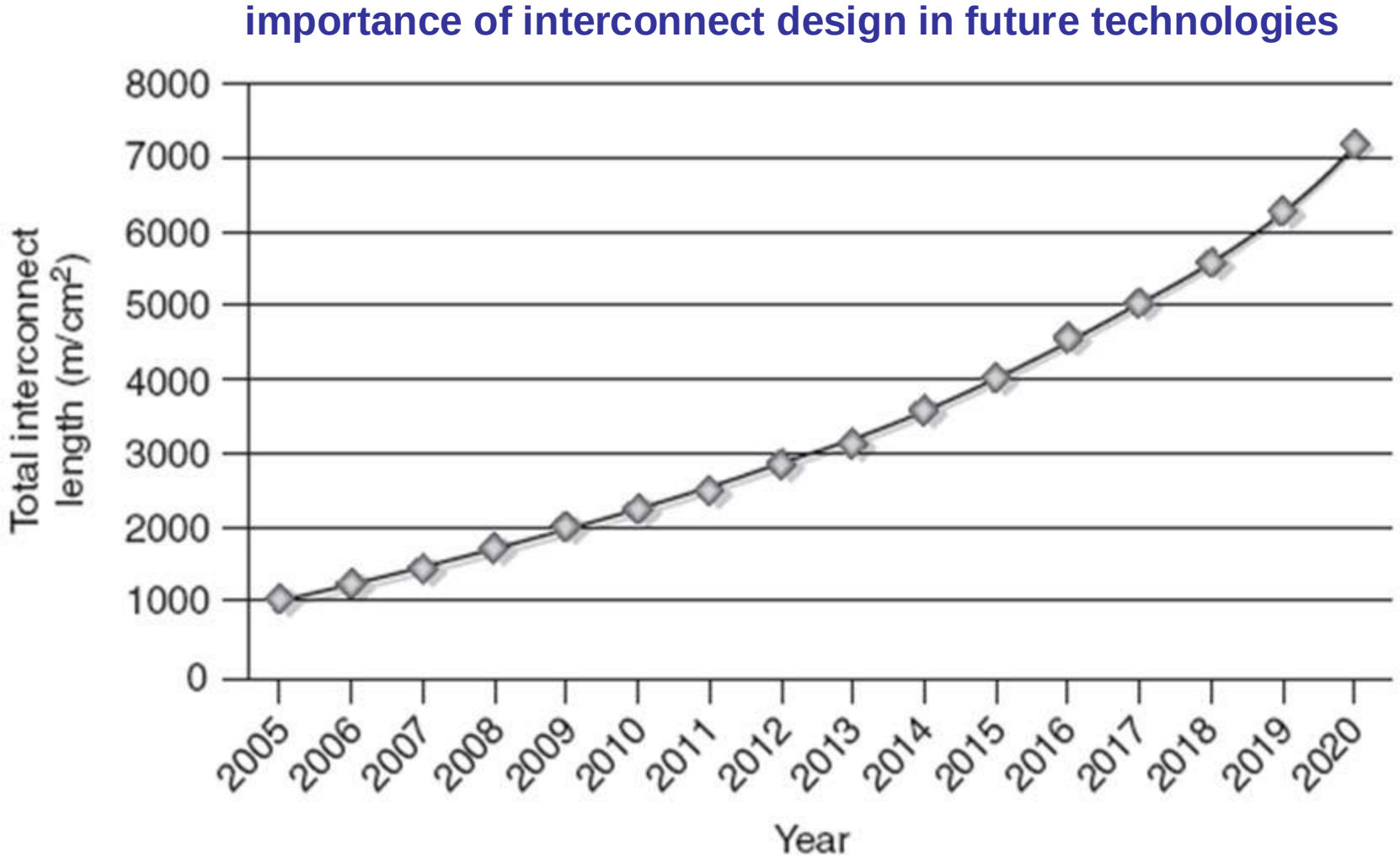
SOC module with interconnect



The cost depends on both on-chip and off-chip interconnect!

ICU: Interconnect interface unit – enable a common interface protocol for all SoC Modules

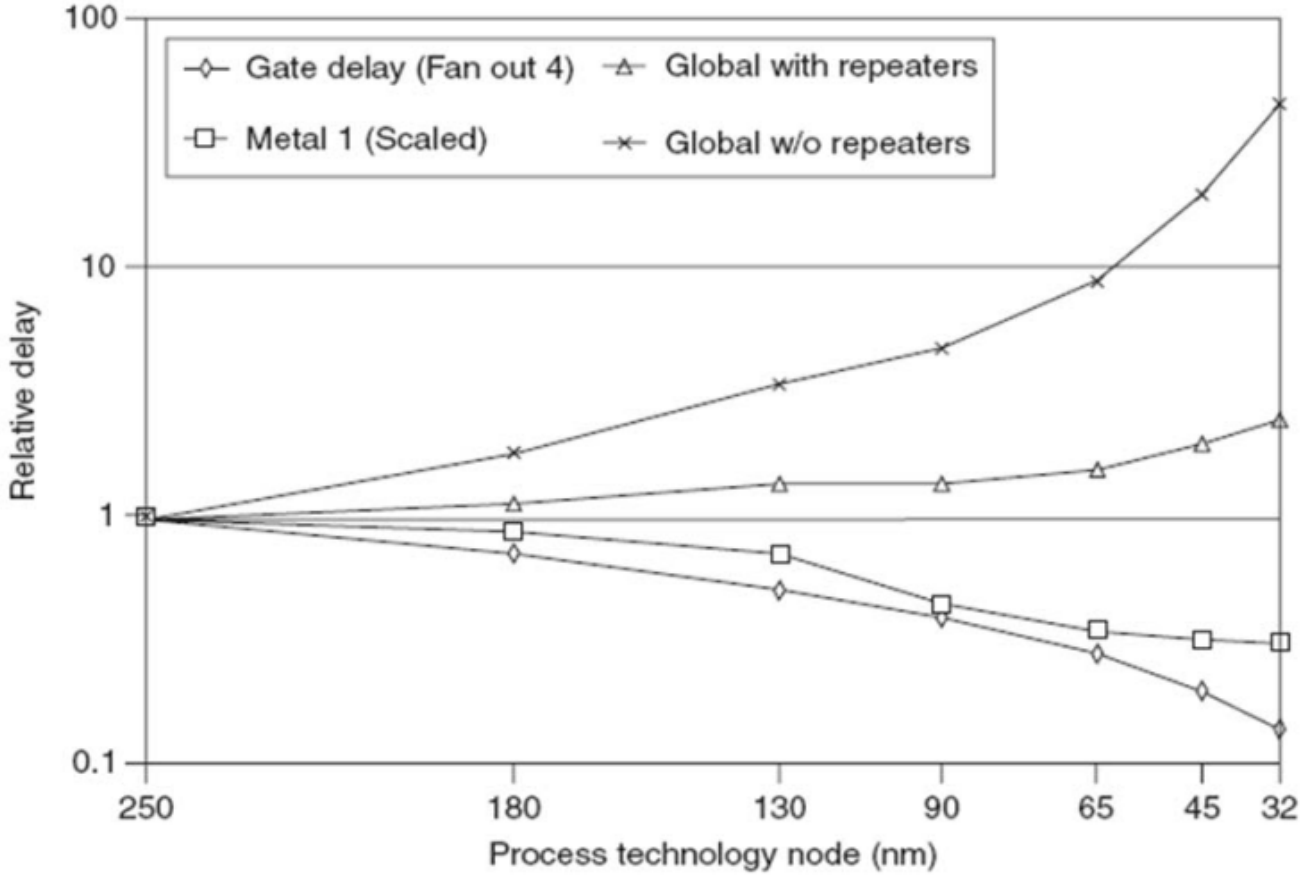
Total Interconnect Length on a Chip



source: UT Austin, EE382M.20: System-on-Chip (SoC) Design

Interconnect Scaling Trends

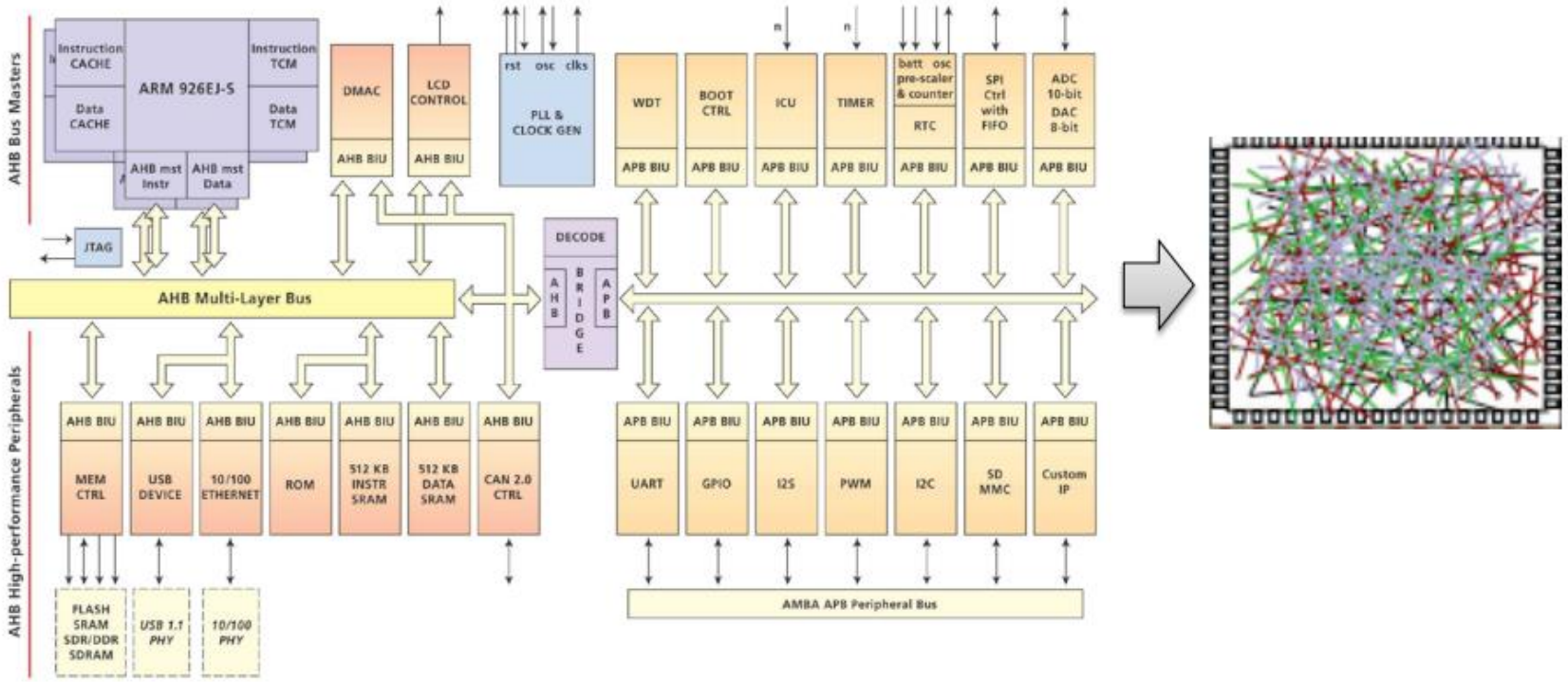
- Global wires scale slower than transistors/gates, why?
- In nano-scale technologies, interconnects greatly impact performance and power consumption of SoCs!
- Increasing wire delay limits achievable performance



Relative delay comparison of wires vs. process technology

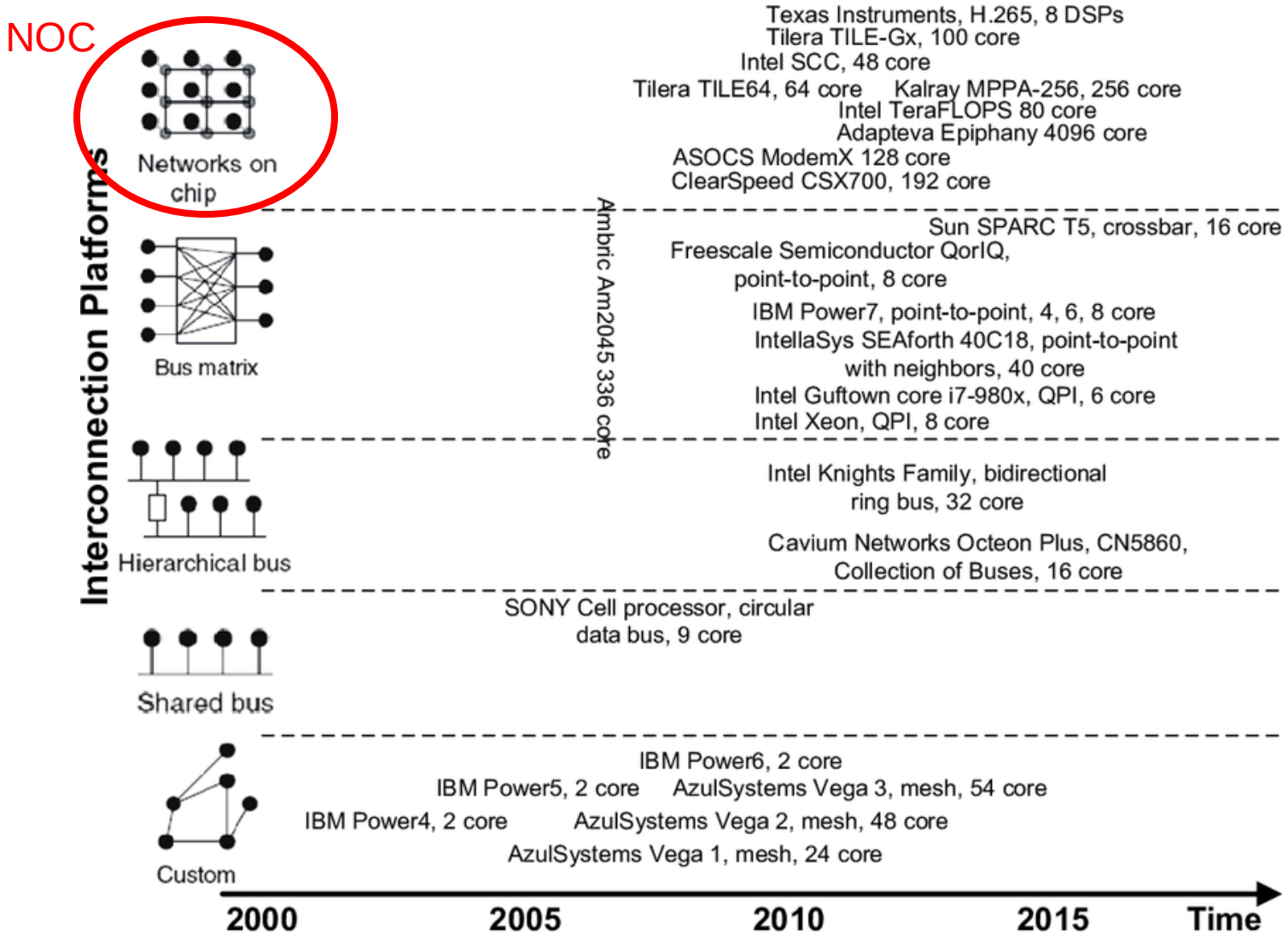
System-level Trends

- Heterogeneity among components that need to be interconnected
- Increasing volume and diversity of traffic



source: https://nanohub.org/courses/ECE695R/

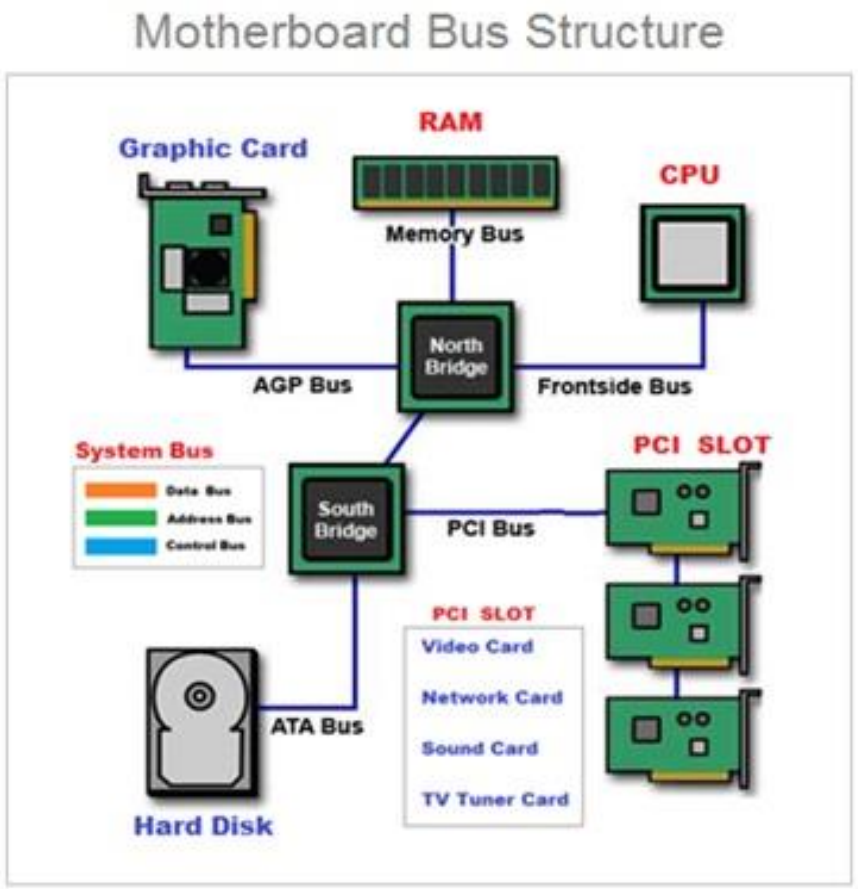
On-chip Interconnect Evolution



Latif K. Design space exploration for MPSoC architectures[J]. 2013.

Recall: VE370

- I/Os and Their Interfaces



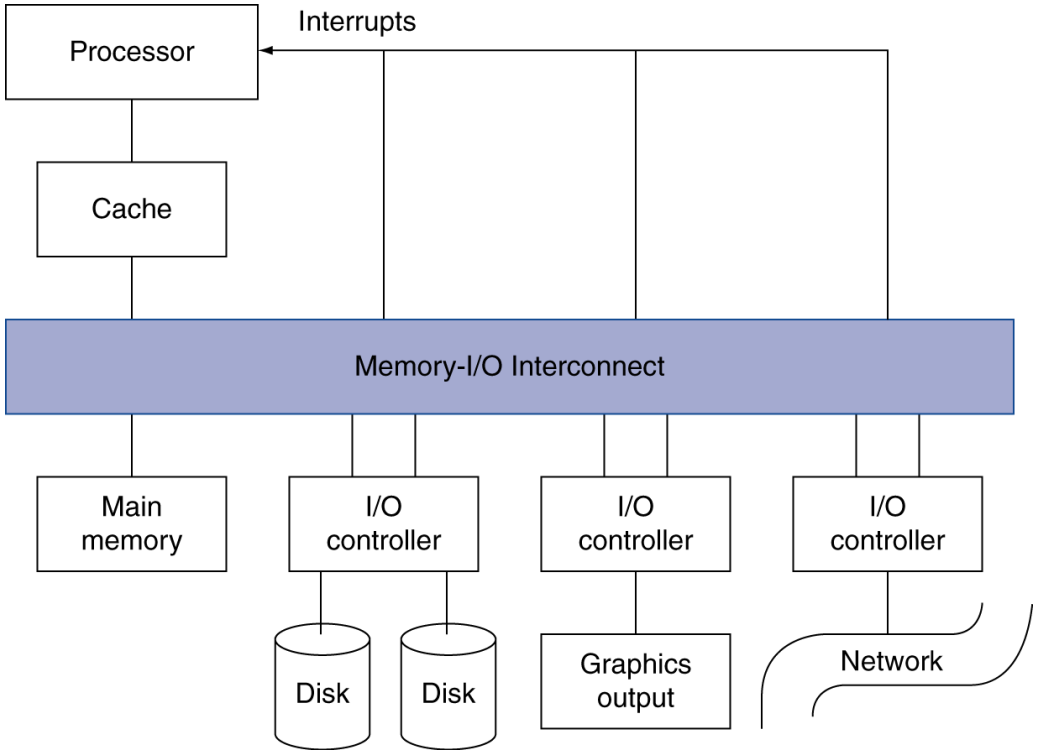
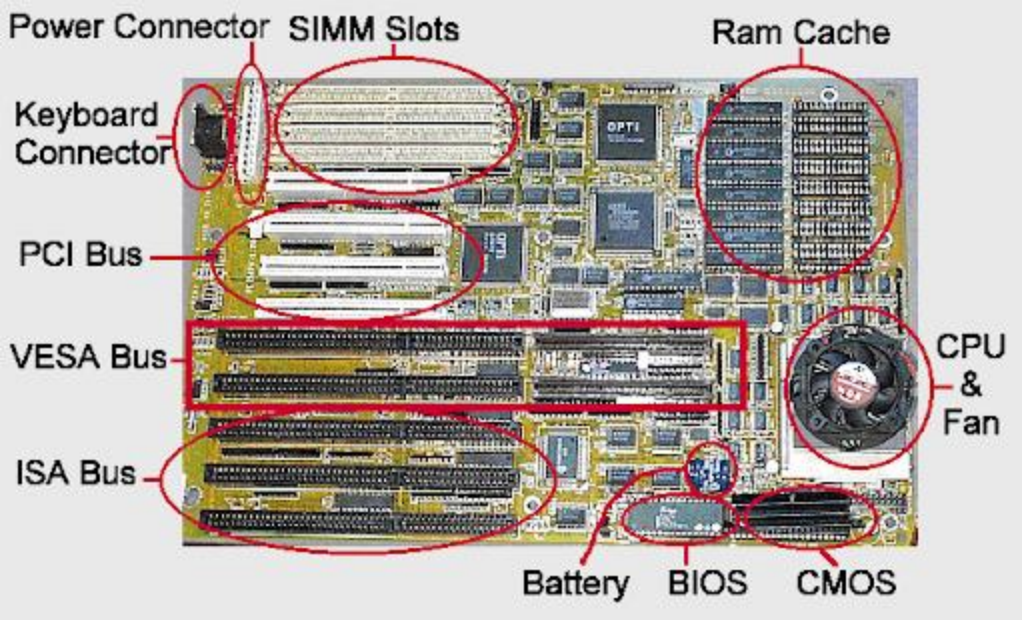


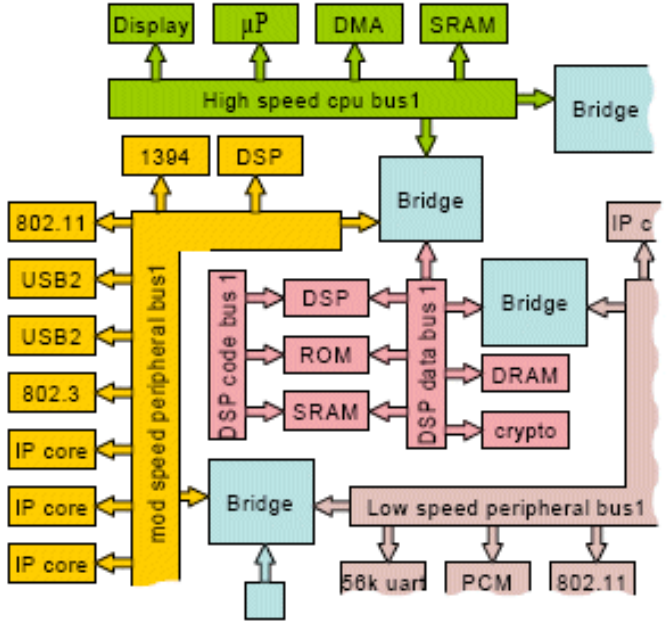
Image: https://www.learncomputerscienceonline.com/computer-bus/

Traditional Bus

Traditional bus vs. SoC on-chip bus



Traditional Bus



On-chip Bus

source: http://www.c-jump.com/CIS24/Slides/Hardware/Hardware.html
https://www.design-reuse.com/articles/14561/system-on-chip-design-using-self-timed-networks-on-chip.html

Differences between traditional bus vs. SoC on-chip bus

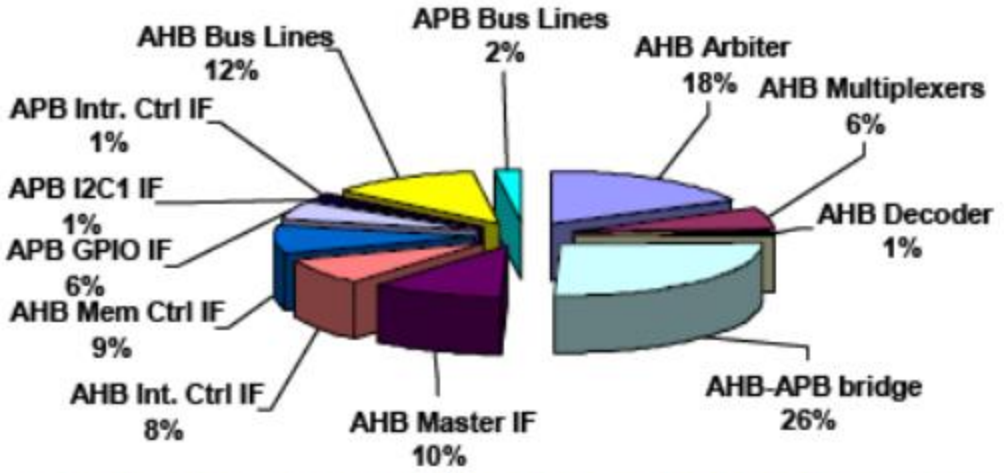
- Bus location: PCB vs. Chip
- The root: I/O pins are limited and fixed (by package)
- Drivers: drivers for SoC bus are smaller, less area and power
- Speed: traditional bus has high capacitive loads and large contact resistance, thus can be slower
- For on-chip buses, bandwidth and latency are very important.

On-chip buses are expensive!

- Complexity of logic can easily compare to a small microprocessor!
- Huge impact on SoC power

System Component	Part Name	Power (mW)
Embedded Processor	ARM946E-S[1]	60
Memory Controller	DW_ahb_memctl[2]	29.1
On-Chip Bus	DW_amba[2]	22.6
Cache	ARM946E-S[1]	36
Interrupt Controller	DW_ahb_ictl[2]	2.6
UART	DW_apb_uart[2]	4.1

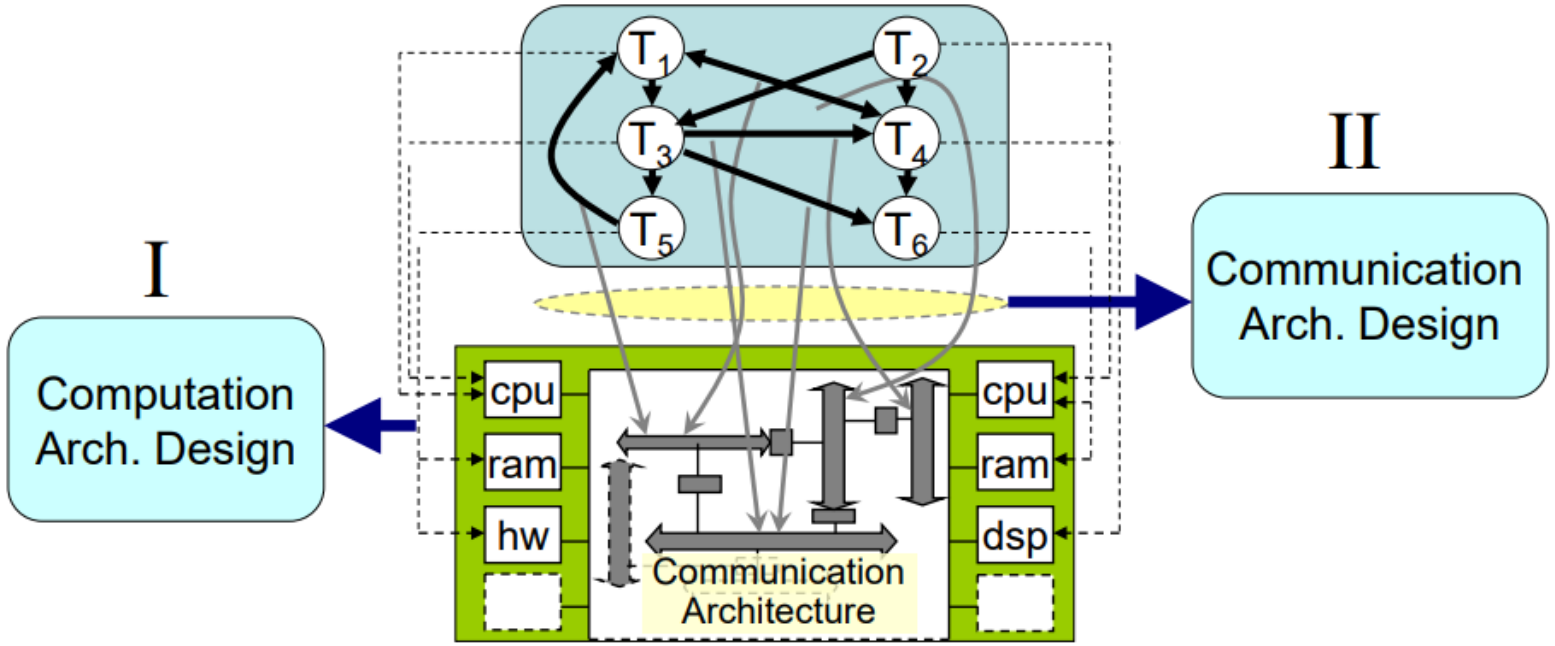
AMBA: one type of on-chip bus



Power breakdown for AMBA AHB (NEC 0.15um)

source: Power Analysis of System-Level On-Chip Communication Architectures Kanishka Lahiri and Anand Raghunathan, ACM/IEEE CODES, Oct. 2004.

Evolution of SoC Design Methodology



Computation-centric system design
- Focus : optimizing computation (e.g., HW/SW partitioning)
- Standard communication architecture

Communication-aware system design
- Incorporate communication architecture into design process
- Evolutionary changes to comm. architecture

Communication -centric system design
- Extensive effort on optimizing communication

+ interconnection architecture

Communication architecture design, customization, exploration, verification and implementation takes up the largest chunk of a design cycle…

source: https://nanohub.org/courses/ECE695R/

Interconnect design

- find the cost and performance of alternatives
- iterated to find the least expensive design that meets the requirements
- consider the larger issues: reliability, scalability, design costs, availability of IP

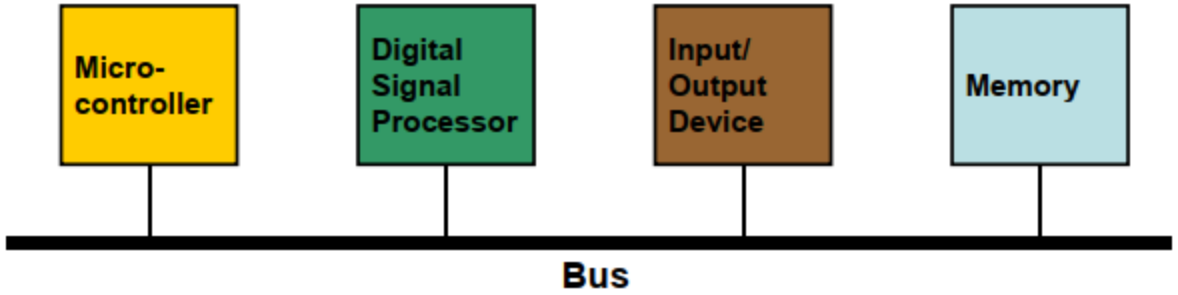
Scalability: plug-in any IP -> keep the performance optimal

Many alternatives

- Find requirements: number of nodes, performance requirements, marginal and development cost
- Bus based: purchased IP or proprietary
- NOC based: static vs dynamic

Bus-based Architectures

- Buses are the simplest and most widely used SoC interconnection networks
  - a collection of signals (wires) to which one or more IP components (which need to communicate data with each other) are connected
- Only one component can transfer data on the shared bus at any given time…



source: UT Austin, EE382M.20: System-on-Chip (SoC) Design

Bus Terminology

- **Master (or Initiator)**
  - IP component that initiates a read or write data transfer
- **Slave (or Target)**
  - IP component that does not initiate transfers and only responds to incoming transfer requests
- **Arbiter**
  - Controls access to the shared bus
  - Uses arbitration scheme to select master to grant access to bus

There can be more than one master

- **Decoder**
  - Determines which component a transfer is intended for
- **Bridge**
  - Connects two busses
  - Acts as *slave* on one side and *master* on the other

Buffer structure / to synchronize data

- **Synchronization**
  - clock management
- **Bus wrapper**: manages multiple protocols

Bus Signal Lines

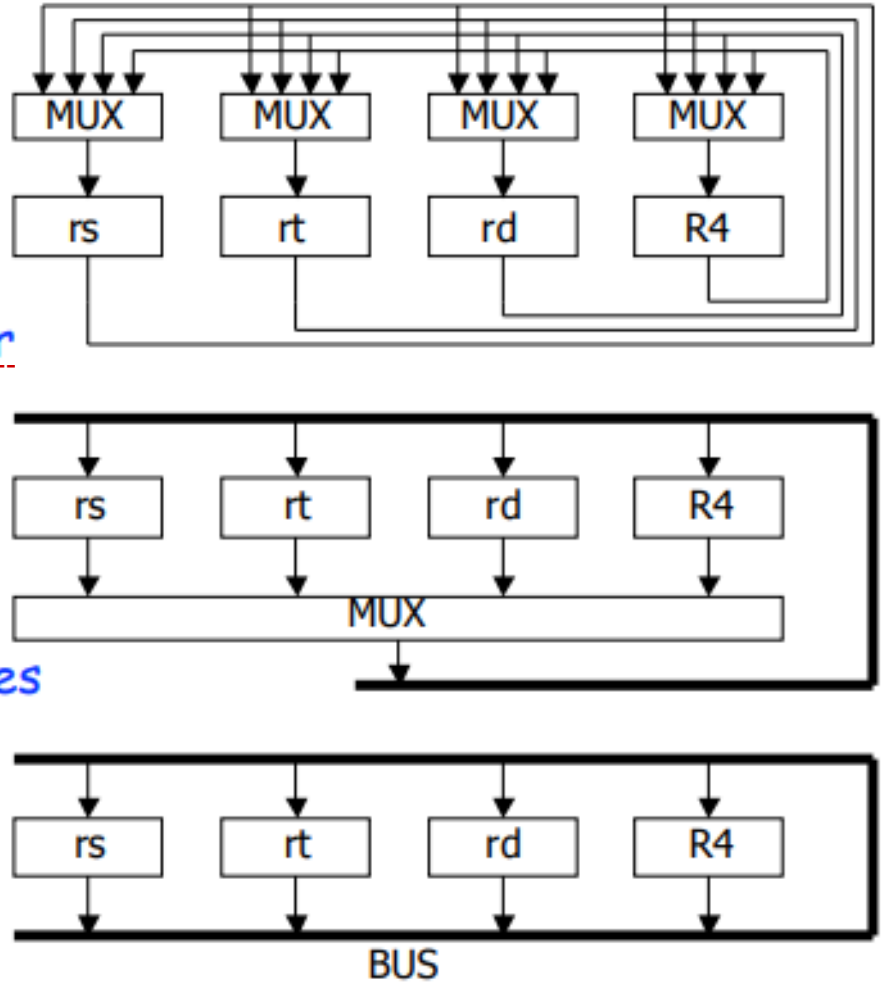
- **A bus typically consists of three types of signal lines**
  - Address
    - Carry address of destination for which transfer is initiated
    - Can be shared or separate for read, write data
  - Data
    - Carry information between source and destination components
    - Can be shared or separate for read, write data
    - Choice of data width critical for application performance
  - Control
    - Requests and acknowledgements
    - Specify more information about type of data transfer
    - Byte enable, burst size, cacheable/bufferable, write-back/through, …

Important Metrics

- Communication Bandwidth (bytes/s)
- Latency (delay between request and response)
- Concurrency (number of independent simultaneous communication channels operating in parallel)
- Master or Slave
- Clock domains (can be multiple clock and data rates across IP modules)
  - IP and processor might operate at different clocks
  - crossing clock domain can cause deadlock and synchronization problems without careful design
- Packet or bus transaction (size and definition of the info. transmitted in a single transaction, e.g. address with control bits for a bus, for NoC it refers to packet)

Interconnect Strategies

- Point-to-point connection
  - Dedicated wires
  - Muxes on inputs of each register
- Common input from multiplexer
  - Load enables for each register
  - Control signals for multiplexer
- Common bus with output enables
  - Output enables and load enables for each register



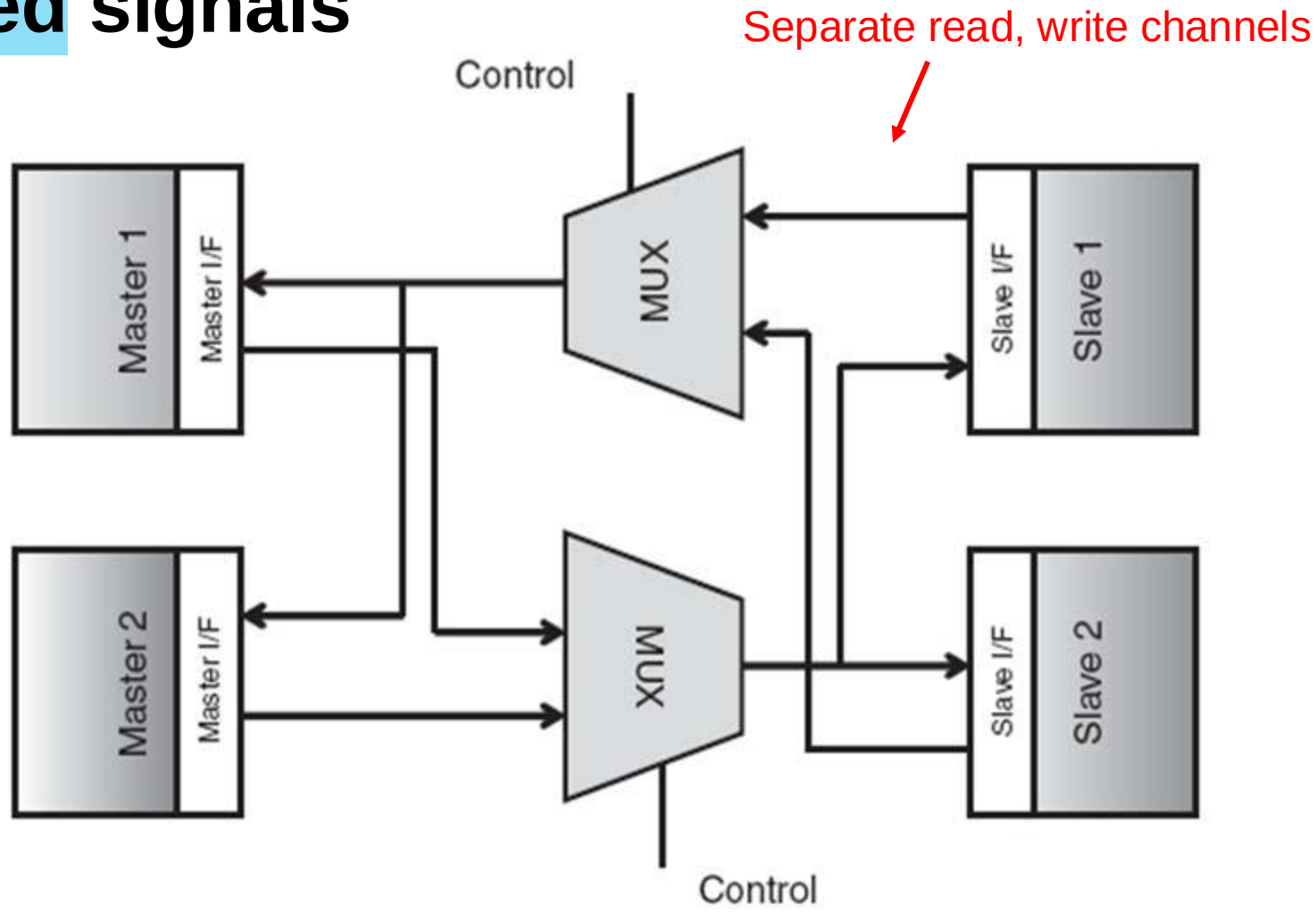
A master connects directly to a slave

Single bus

source: https://inst.eecs.berkeley.edu/~cs150/fa05/Lectures/13-CompOrgIx6.pdf

Physical Bus Structure

- **MUX-based signals**

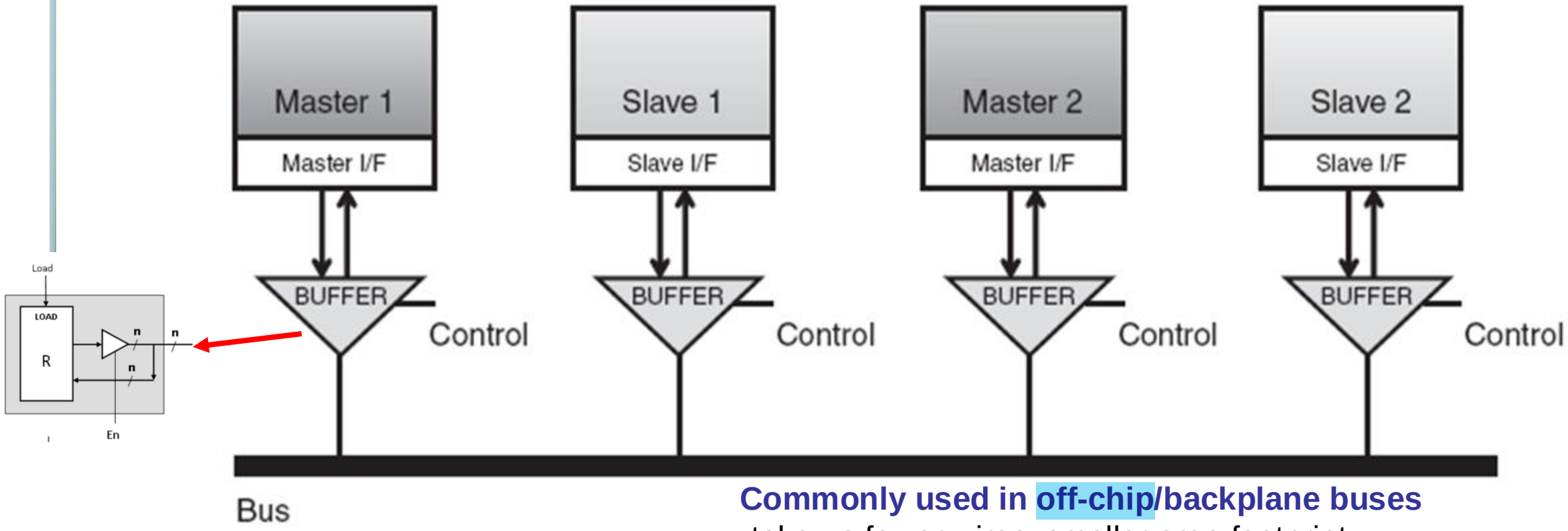


source: UT Austin, EE382M.20: System-on-Chip (SoC) Design

Physical Bus Structure

- **Tri-state buffer based bidirectional signals**

Cheaper than a Mux / because it is some transistors



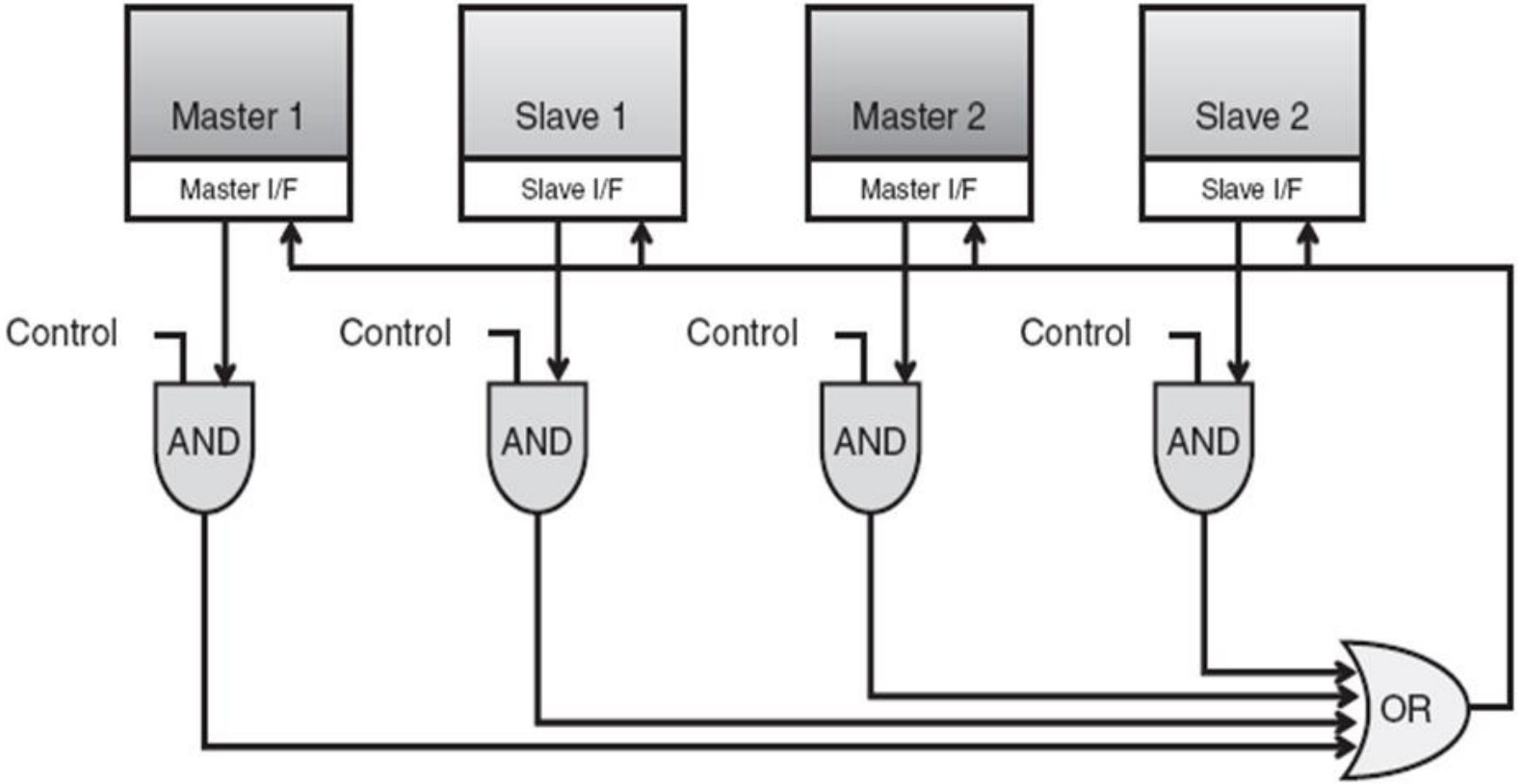
Commonly used in off-chip/backplane buses

- take up fewer wires, smaller area footprint
- higher power, higher delay, hard to debug

Physical Bus Structure

- **AND-OR based signals**

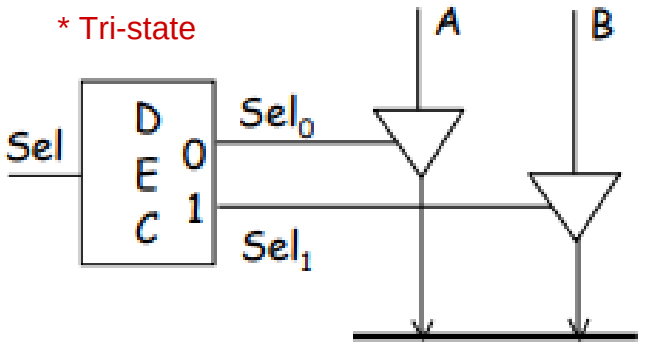
Replace buffer with AND gates + single OR gate (to select signal)



Slightly faster

MUX vs. Tri-state

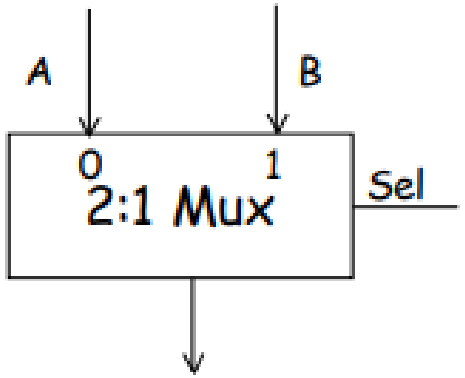
* Tri-state



Takes less area + less costly

Buffer circuits simple!
Scales nicely for high fan-in
and wide bit widths!

* Mux



Scales poorly for high fan-in
or wide bit widths

source: https://inst.eecs.berkeley.edu/~cs150/fa05/Lectures/13-CompOrgIx6.pdf

Where are we Heading?

- More buses…
- NoC

Action Items

- Lab #1 is due!
- Reading Materials
  - Ch. 5.1 – 5.3, literature on canvas

On-chip Interconnect Schemes for Reconfigurable System-on-Chip

Andy S. Lee, Neil W. Bergmann.
School of ITEE, The University of Queensland, Brisbane Australia
{andy, n.bergmann} @itee.uq.edu.au

ABSTRACT

On-chip communication architectures can have a great influence on the speed and area of System-on-Chip designs, and this influence is expected to be even more pronounced on reconfigurable System-on-Chip (rSoC) designs. To date, little research has been conducted on the performance implications of different on-chip communication architectures for rSoC designs. This paper motivates the need for such research and analyses current and proposed interconnect technologies for rSoC design. The paper also describes work in progress on implementation of a simple serial bus and a packet-switched network, as well as a methodology for quantitatively evaluating the performance of these interconnection structures in comparison to conventional buses.

Keywords: FPGAs, Reconfigurable Logic, System-on-Chip

1. INTRODUCTION

System-on-chip (SoC) technology has evolved as the predominant circuit design methodology for custom ASICs. SoC technology moves design from the circuit level to the system level, concentrating on the selection of appropriate pre-designed IP Blocks, and their interconnection into a complete system.

However, modern ASIC design and fabrication are expensive. Design tools may cost many hundreds of thousands of dollars, while tooling and mask costs for large SoC designs now approach $1million. For low volume applications, and especially for research and development projects in universities, reconfigurable System-on-Chip (rSoC) technology is more cost effective. Like conventional SoC design, rSoC involves the assembly of predefined IP blocks (such as processors and peripherals) and their interconnection. However, here the fabrication technology uses mega-gate FPGAs, rather than custom ASICs. First generation commercial rSoC products are now offered by most FPGA vendors.

rSoC has re-programmability that allows designers to implement and test theories in real hardware many times on a single device. The turnaround time is also relatively short, allowing almost instantaneous hardware implementation, with the lengthy delay for silicon fabrication eliminated from the design cycle.

However, simply mapping ASIC designs on to reprogrammable devices will not yield efficient and optimized results. Due to the underlying architectural differences between reconfigurable devices and ASICs[1], such as much more constrained wiring channels, conventional SoC design methods may not always be appropriate for rSoC.

There is, therefore a need to revisit conventional SoC design techniques, and analyze their applicability for rSoC, and also to examine new rSoC design techniques that can make good use of the special characteristics of rSoC.

A key component in any SoC design is the interconnection fabric that is used for inter-module communication. The most common interconnection strategy is a parallel system bus. A number of different "standard" buses are already being used in custom SoC designs. Other interconnection strategies for SoC have also been proposed, such as cross-bar switches, packet-switch network, and on-chip LANs.

For rSoC, a number of different bus "standards" have also been proposed. It is not clear that these bus structures, which have been used more or less unchanged from custom SoC designs, are the most appropriate for rSoC. To date, little work has been done on alternate rSoC interconnection networks.

This paper presents our early work on evaluating the applicability of existing and proposed interconnection structures for rSoC designs. We first examine existing rSoC bus structures, viz. the IBM CoreConnect bus used by Xilinx, the ARM AMBA bus used by Altera, and the "Open Source" Wishbone standard. We compare these buses and evaluate their applicability to rSoC designs.

Microelectronics: Design, Technology, and Packaging, edited by Derek Abbott, Kamran Eshraghian, Charles A. Musca, Dimitris Pavlidis, Neil Weste, Proceedings of SPIE Vol. 5274 (SPIE, Bellingham, WA, 2004) · 0277-786X/04/$15 · doi: 10.1117/12.523334

Acknowledgement

Slides in this topic are inspired in part by material developed and copyright by:

- Dr. Wayne Luk (Imperial College)
- Dr. Gul N. Khan (Ryerson University)
- Dr. Andreas Gerstlauer (UT Austin)
- Dr. Anand Raghunathan (Purdue)
- Dr. Sudeep Pasricha (Colorado)
- Dr. Nikil Dutt (UCI)