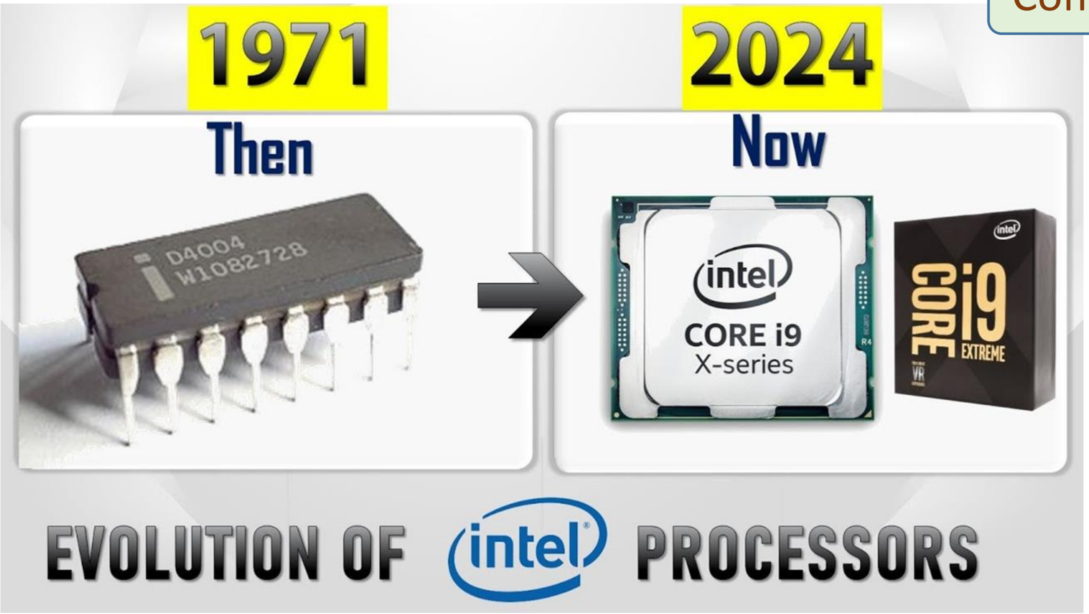
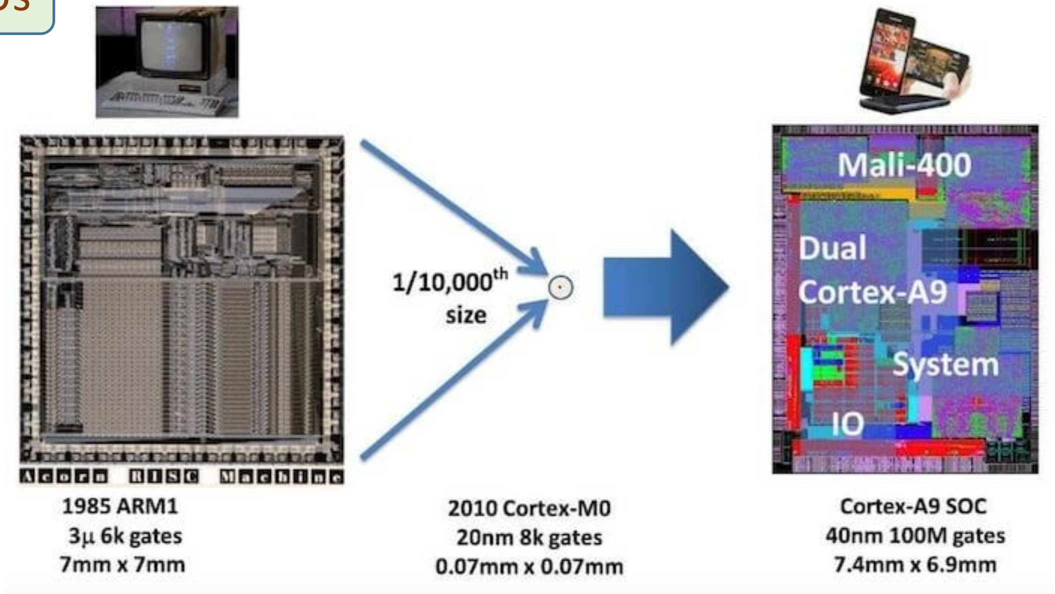
Instruction Set Architecture (ISA - Ch 2)

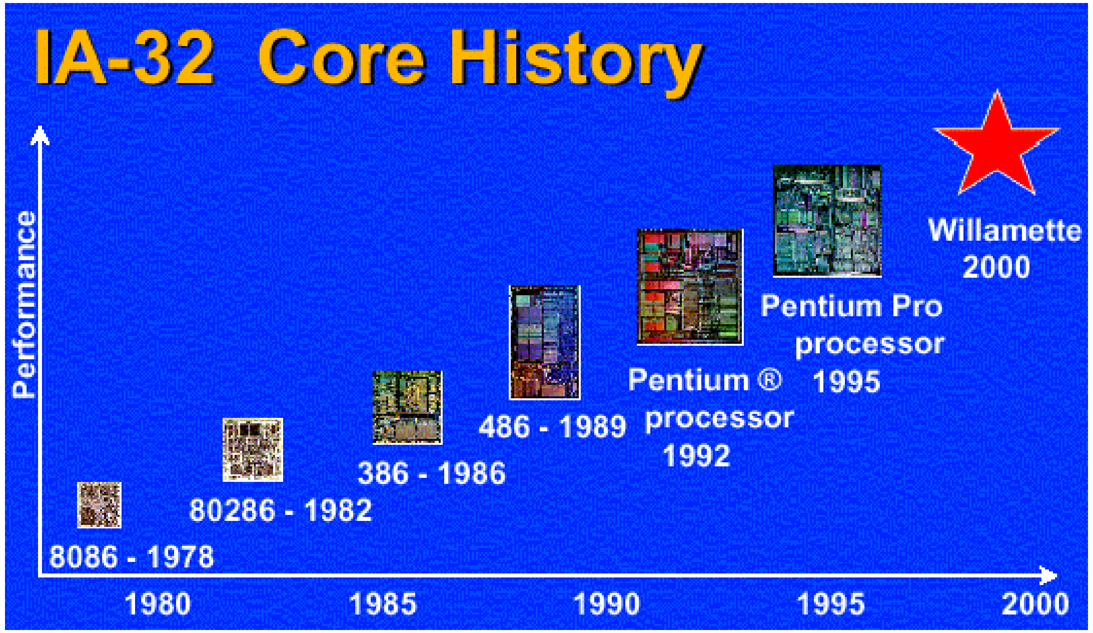
Dr. Rajib Ranjan Maiti (Mtech and PhD at IIT Kharagpur)

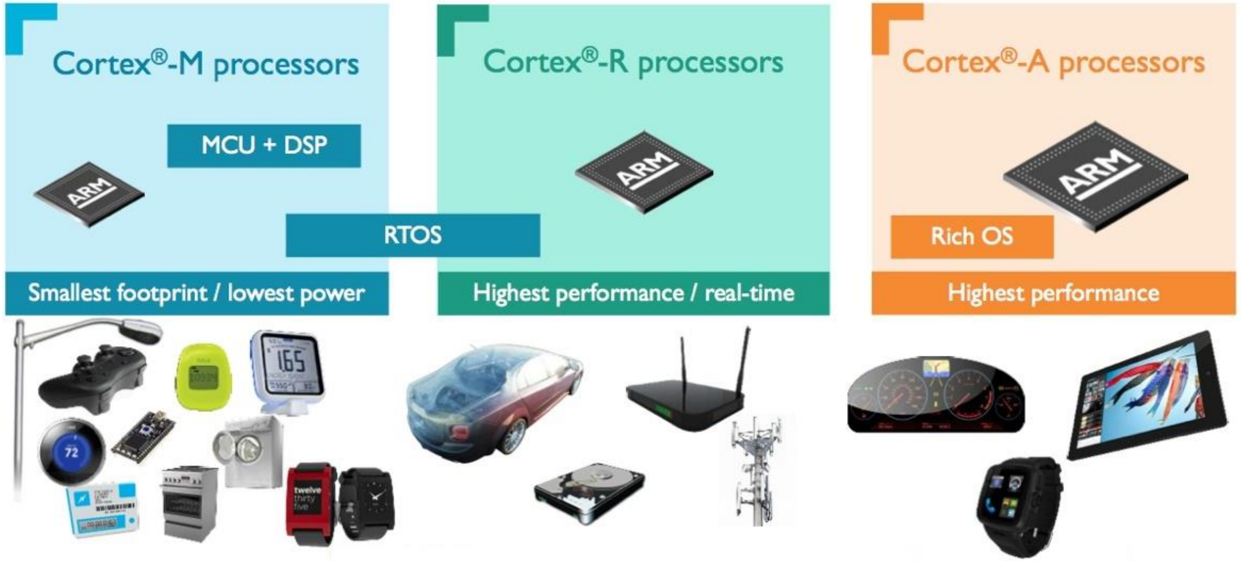
CSIS, BITS-Pilani, Hyderabad

Commercial CPUs

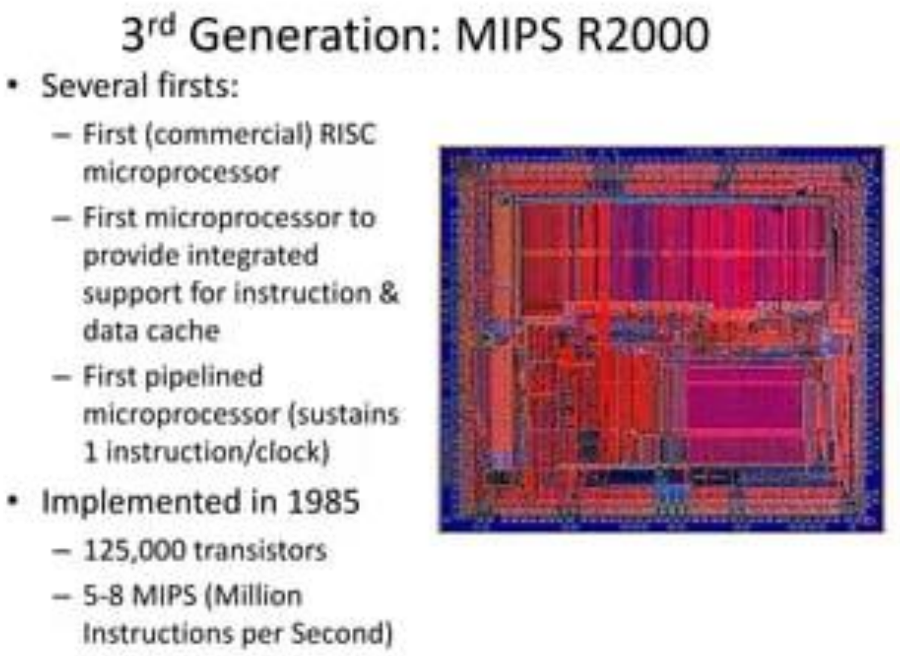






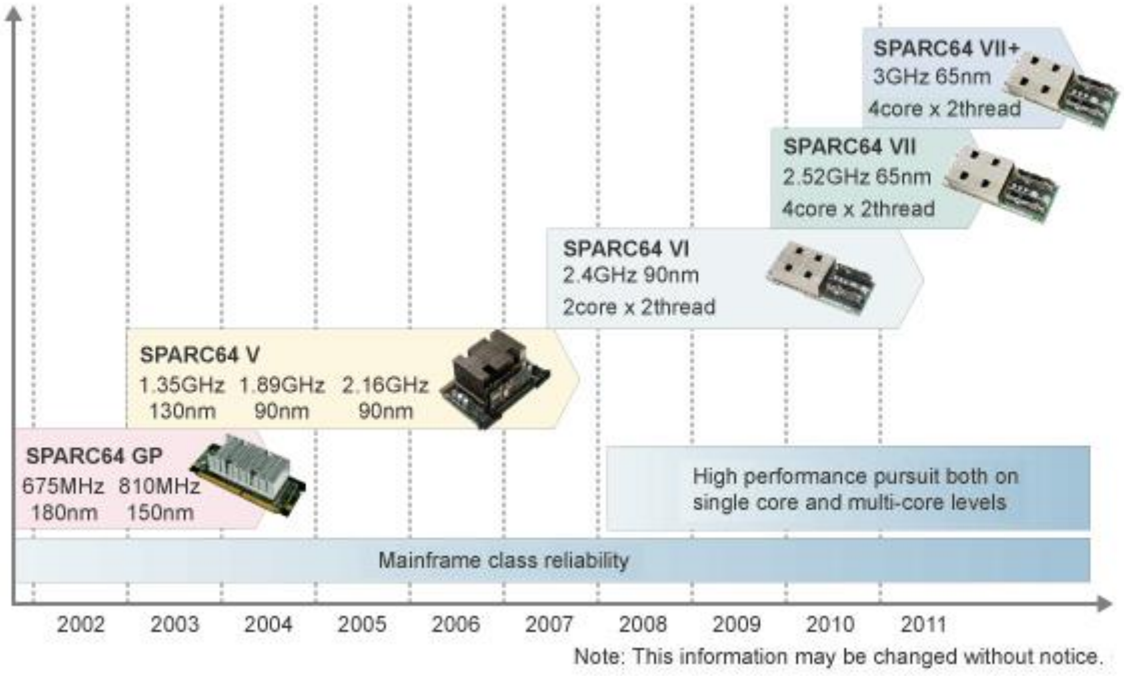


Commercial CPUs



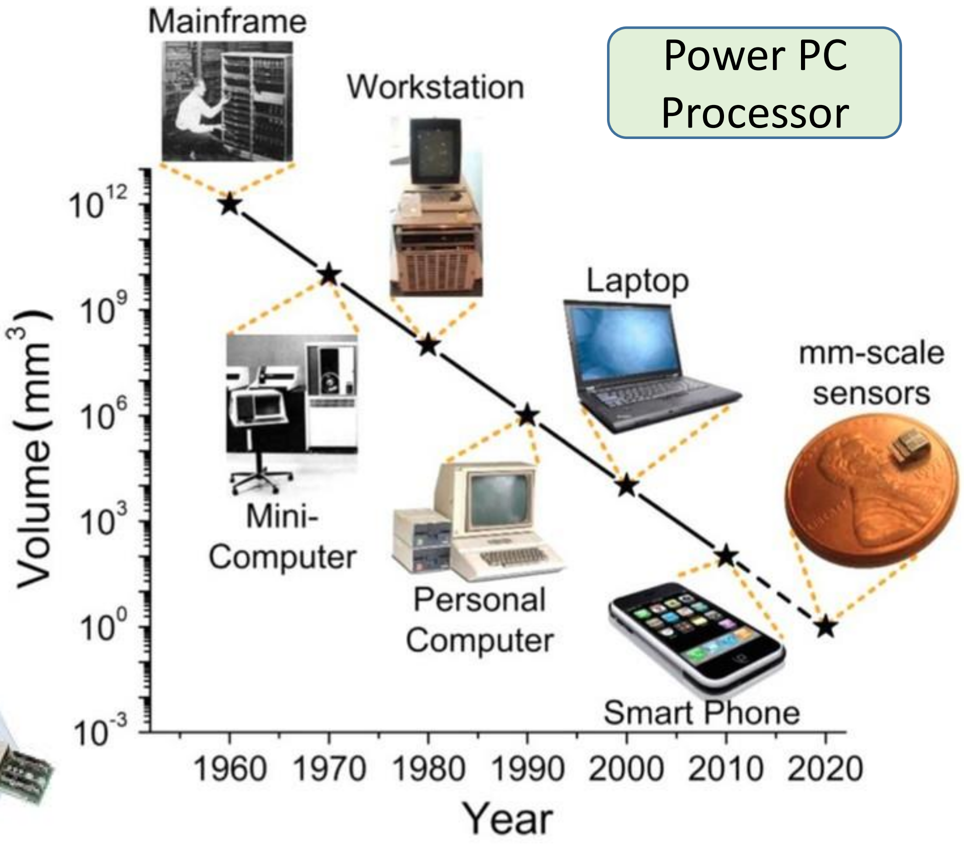
3rd Generation: MIPS R2000

- Several firsts:
  - First (commercial) RISC microprocessor
  - First microprocessor to provide integrated support for instruction & data cache
  - First pipelined microprocessor (sustains 1 instruction/clock)
- Implemented in 1985
  - 125,000 transistors
  - 5-8 MIPS (Million Instructions per Second)

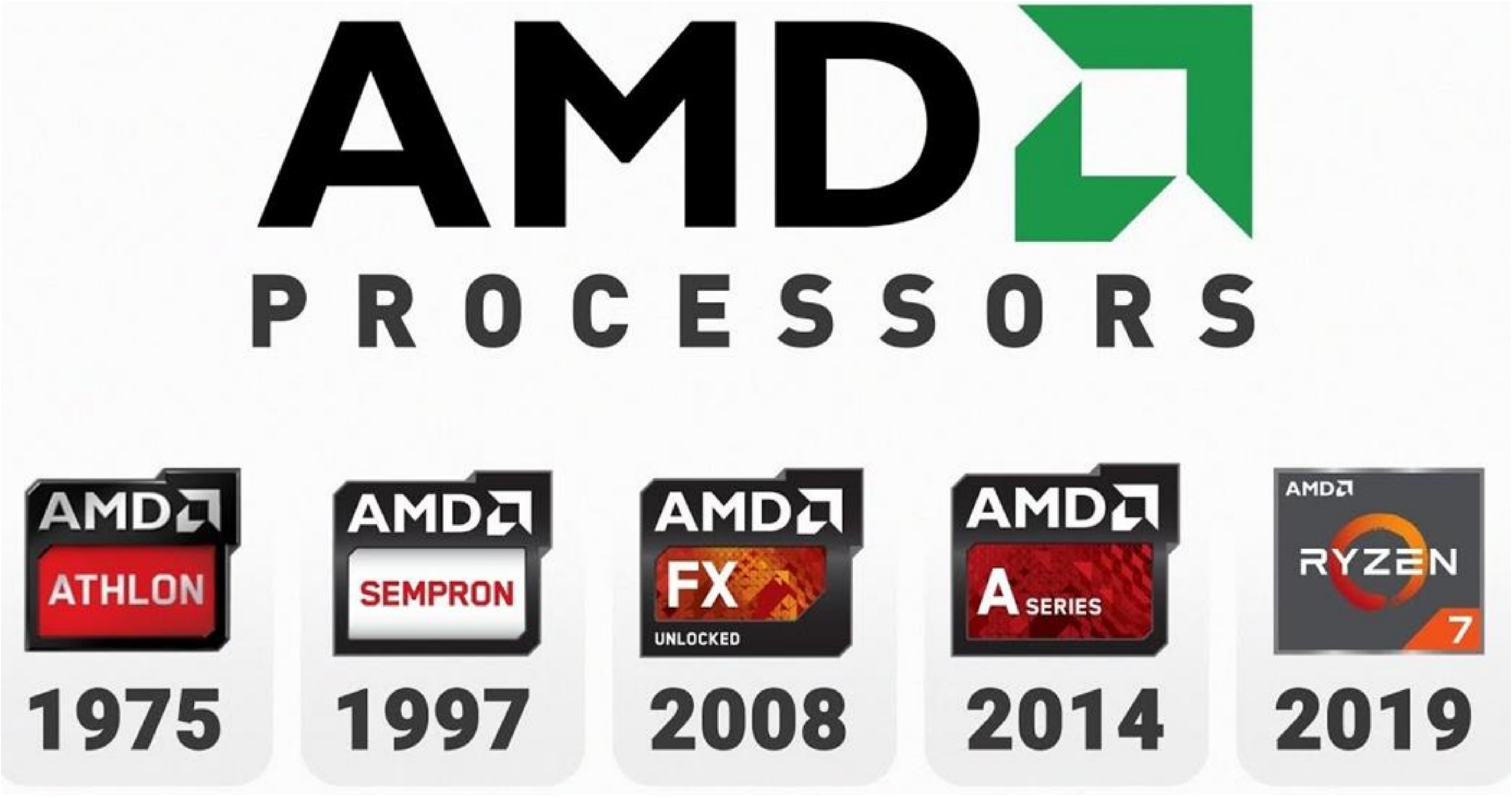


Note: This information may be changed without notice.

Power PC Processor



Commercial CPUs



Top Industries interested in **Comp Arch** Specialists

Quadric

HUAWEI

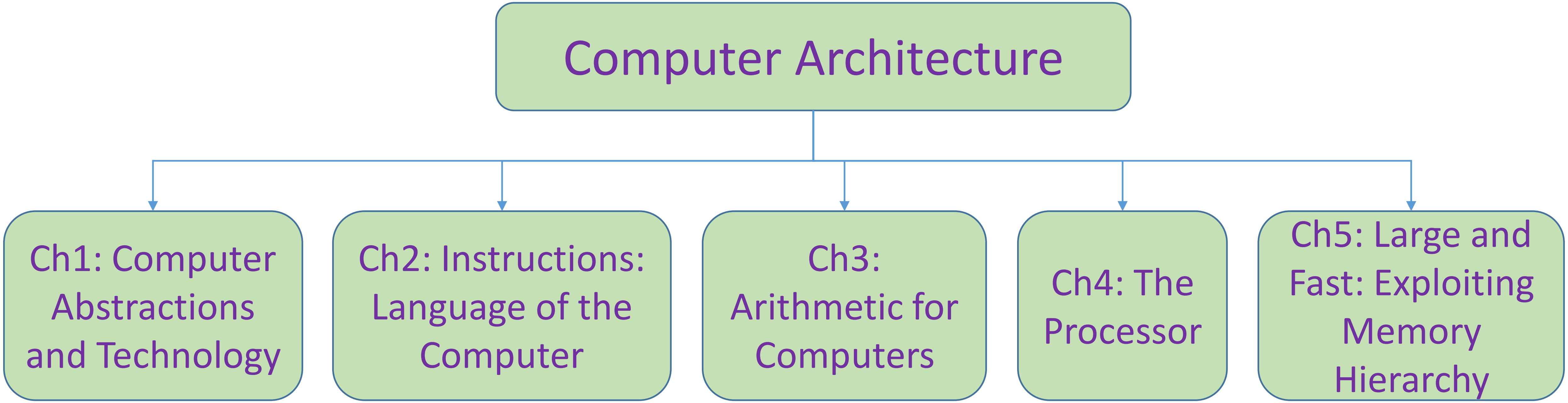
MIRABILIS design

Top Industries interested in **Comp Arch** Specialists

tenstorrent

SILICON LABS

SiFive



Computer Architecture

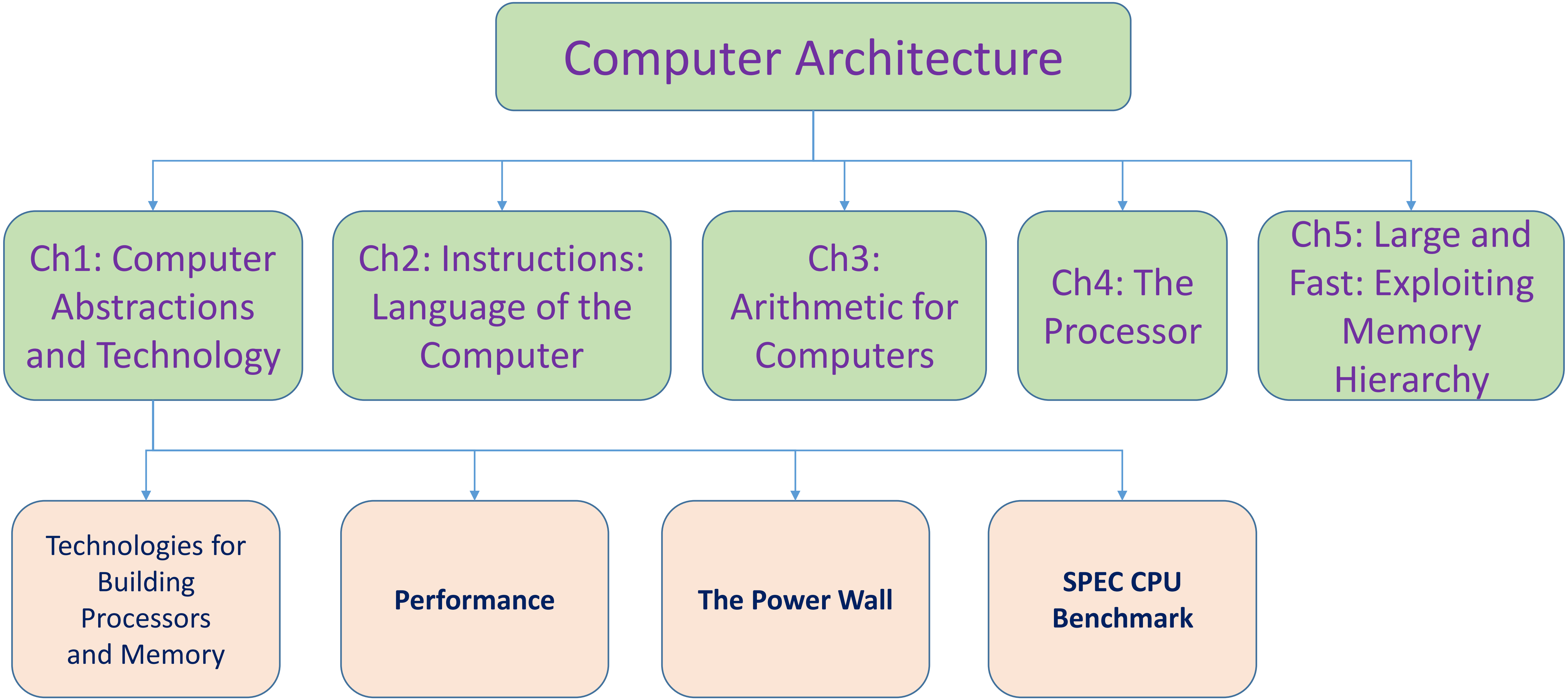
Ch1: Computer Abstractions and Technology

Ch2: Instructions: Language of the Computer

Ch3: Arithmetic for Computers

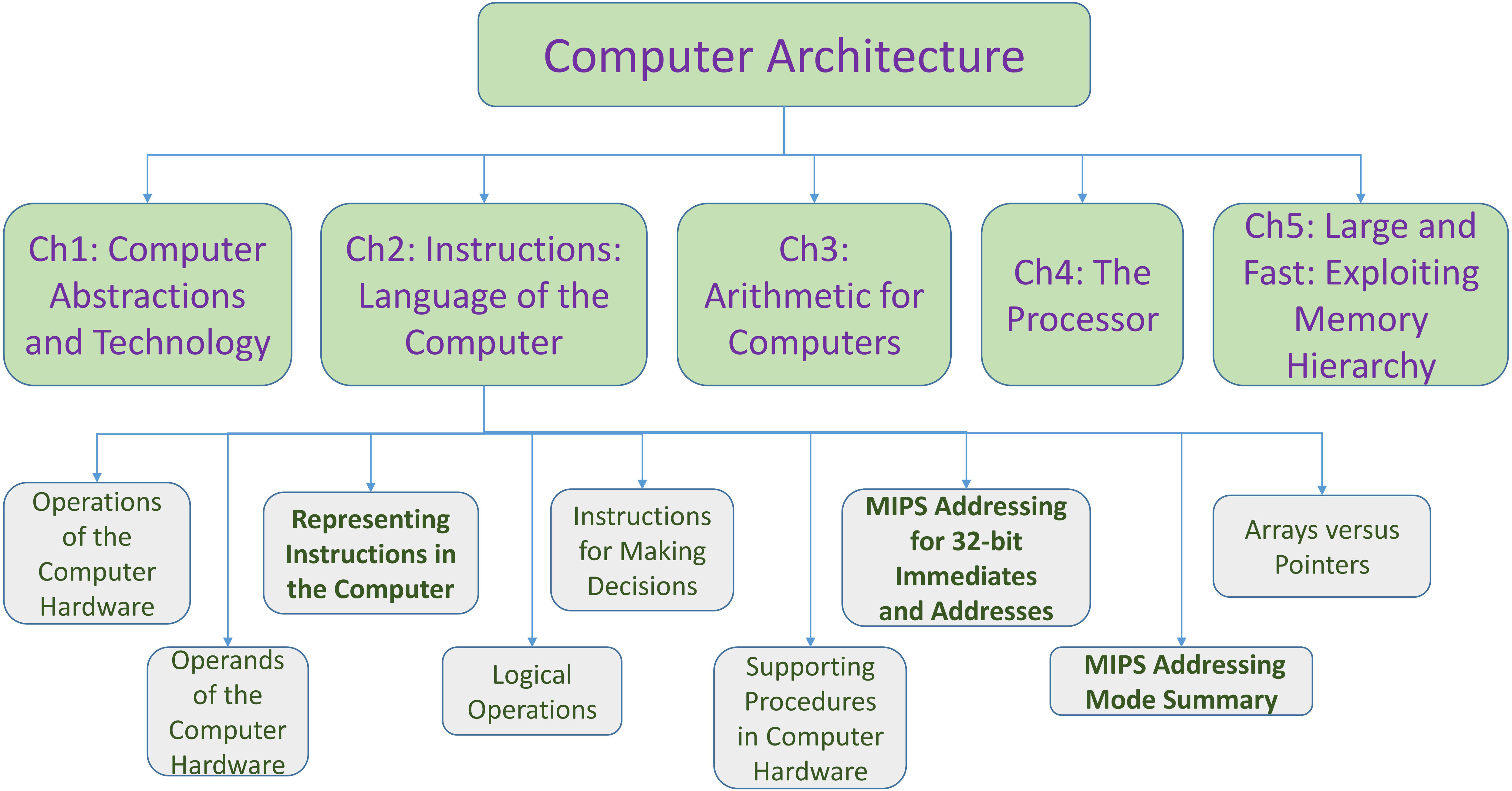
Ch4: The Processor

Ch5: Large and Fast: Exploiting Memory Hierarchy



Computer Architecture

- Ch1: Computer Abstractions and Technology
  - Technologies for Building Processors and Memory
  - **Performance**
  - **The Power Wall**
  - **SPEC CPU Benchmark**
- Ch2: Instructions: Language of the Computer
- Ch3: Arithmetic for Computers
- Ch4: The Processor
- Ch5: Large and Fast: Exploiting Memory Hierarchy



Computer Architecture

Ch1: Computer Abstractions and Technology

Ch2: Instructions: Language of the Computer

Ch3: Arithmetic for Computers

Ch4: The Processor

Ch5: Large and Fast: Exploiting Memory Hierarchy

Operations of the Computer Hardware

Operands of the Computer Hardware

Representing Instructions in the Computer

Logical Operations

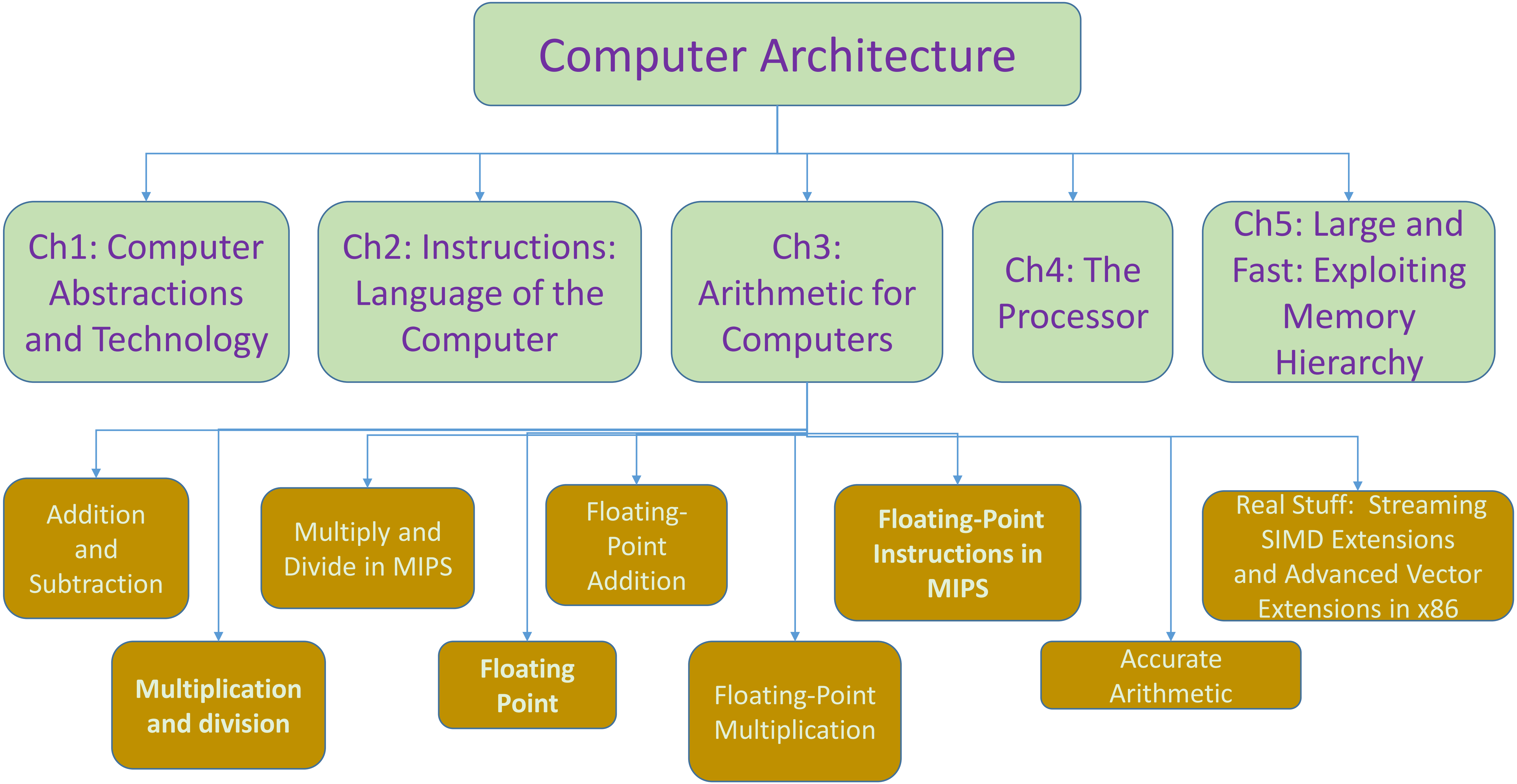
Instructions for Making Decisions

Supporting Procedures in Computer Hardware

MIPS Addressing for 32-bit Immediates and Addresses

MIPS Addressing Mode Summary

Arrays versus Pointers



Computer Architecture

Ch1: Computer Abstractions and Technology

Ch2: Instructions: Language of the Computer

Ch3: Arithmetic for Computers

Ch4: The Processor

Ch5: Large and Fast: Exploiting Memory Hierarchy

Addition and Subtraction

Multiplication and division

Multiply and Divide in MIPS

Floating Point

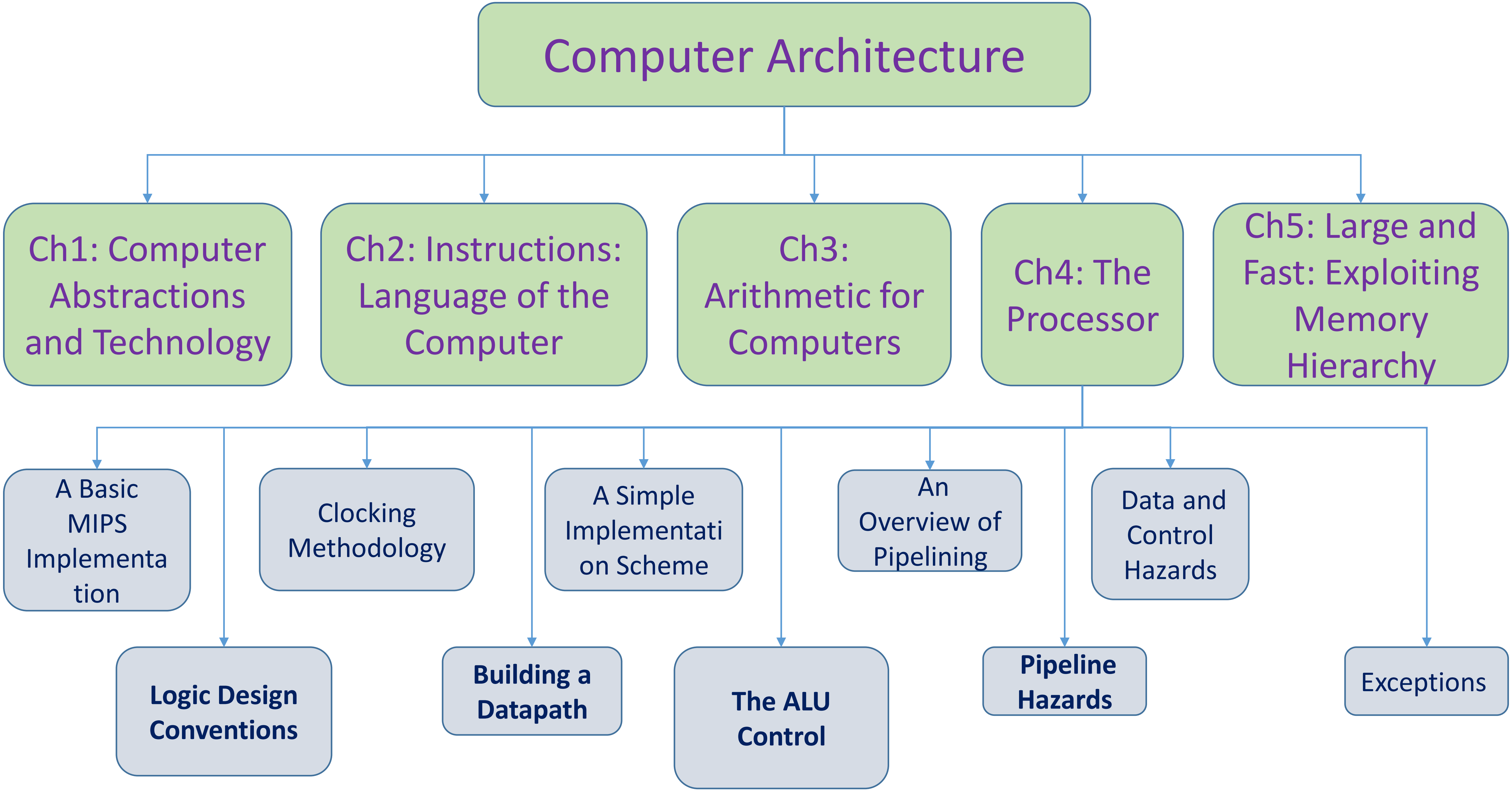
Floating-Point Addition

Floating-Point Multiplication

Floating-Point Instructions in MIPS

Accurate Arithmetic

Real Stuff: Streaming SIMD Extensions and Advanced Vector Extensions in x86



Computer Architecture

Ch1: Computer Abstractions and Technology

Ch2: Instructions: Language of the Computer

Ch3: Arithmetic for Computers

Ch4: The Processor

Ch5: Large and Fast: Exploiting Memory Hierarchy

A Basic MIPS Implementation

Logic Design Conventions

Clocking Methodology

Building a Datapath

A Simple Implementation Scheme

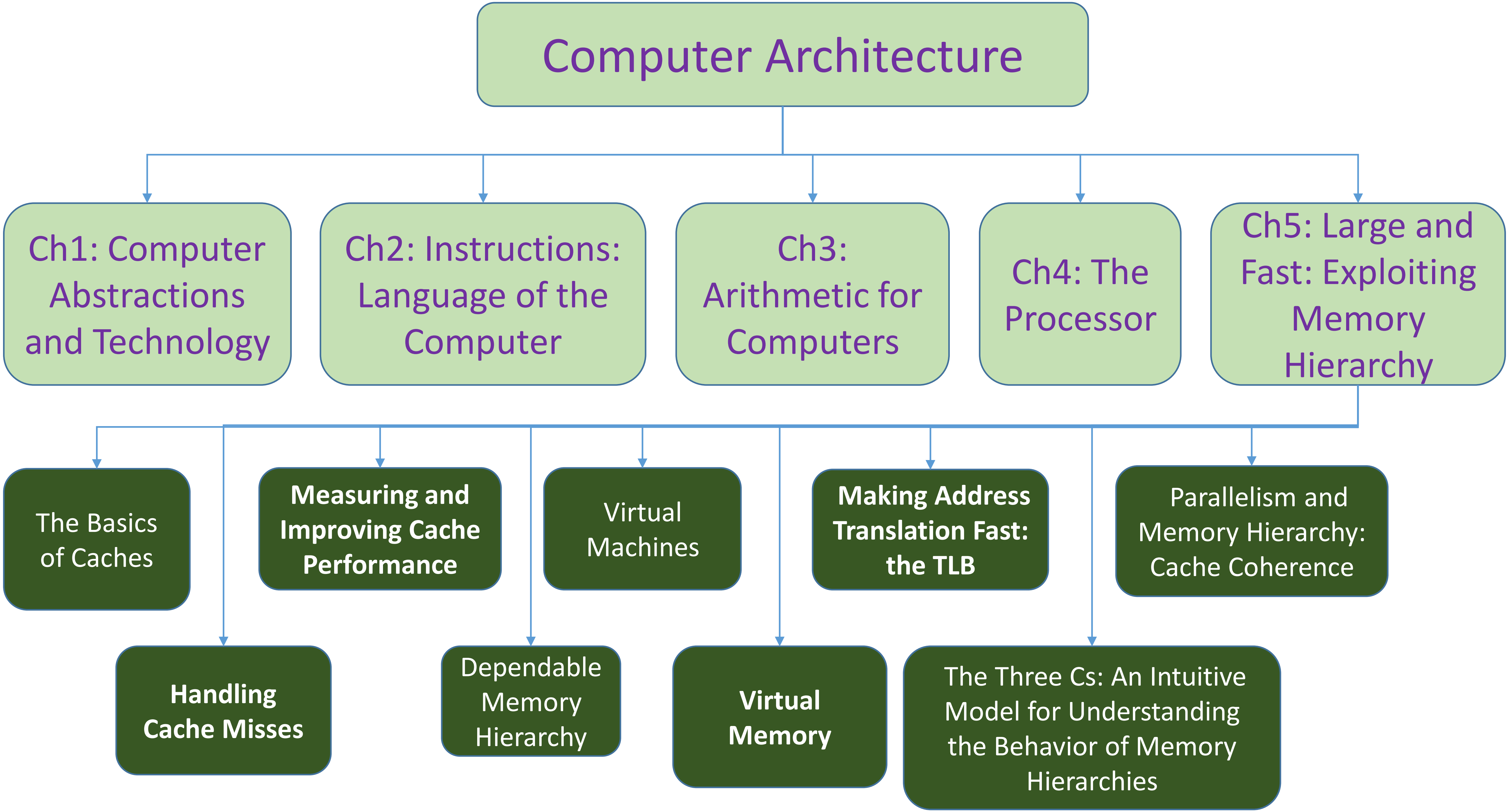
The ALU Control

An Overview of Pipelining

Pipeline Hazards

Data and Control Hazards

Exceptions



Computer Architecture

- Ch1: Computer Abstractions and Technology
- Ch2: Instructions: Language of the Computer
- Ch3: Arithmetic for Computers
- Ch4: The Processor
- Ch5: Large and Fast: Exploiting Memory Hierarchy
  - The Basics of Caches
  - Handling Cache Misses
  - Measuring and Improving Cache Performance
  - Dependable Memory Hierarchy
  - Virtual Machines
  - Virtual Memory
  - Making Address Translation Fast: the TLB
  - The Three Cs: An Intuitive Model for Understanding the Behavior of Memory Hierarchies
  - Parallelism and Memory Hierarchy: Cache Coherence

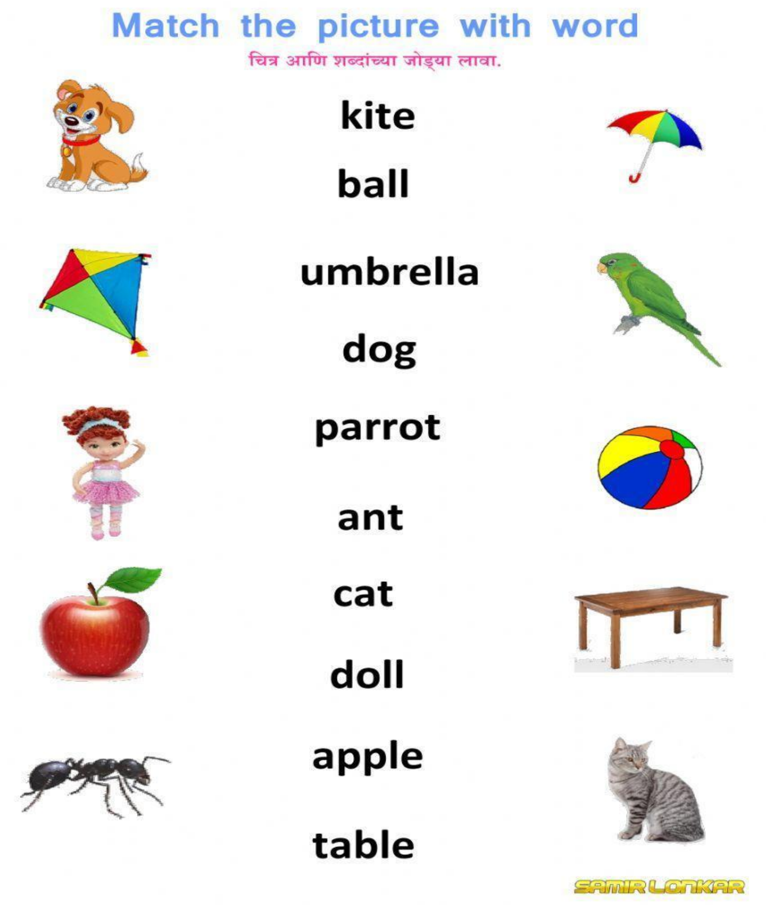
Teaching Policy and Logistics

- **Lab Evaluation** (In-Class)
  - Before Mid Sem: Best of Test 1 and Test 2 = 10%
  - After Mid Sem: Best of Test 3 and Test 4 = 10%
- **Quiz Evaluation** (In-Tut-Class)
  - Before Mid Sem: Best of Test 1 and Test 2 = 5%
  - After Mid Sem: Best of Test 3 and Test 4 = 5%
- **MidSem Exam** (08/10 - 1.30 - 3.00PM)
  - Question Pattern: Mostly numerical (will be discussed in lec class)
- **Compre Exam** (06/12 FN)
  - Question Pattern: Mostly numerical (will be discussed in lec class)

- Answer Key release
  - Immediately after exam
  - Key verification time : 24 hours after key release
- Make up :
  - **no make up for Quiz and Lab Test**
  - Prior permission required for midsem and compre

To become an efficient speaker, be strong in vocabulary of a native language

English: started learning letters and then words



Match the picture with word

चित्र आणि शब्दांच्या जोड्या लावा.

kite

ball

umbrella

dog

parrot

ant

cat

doll

apple

table

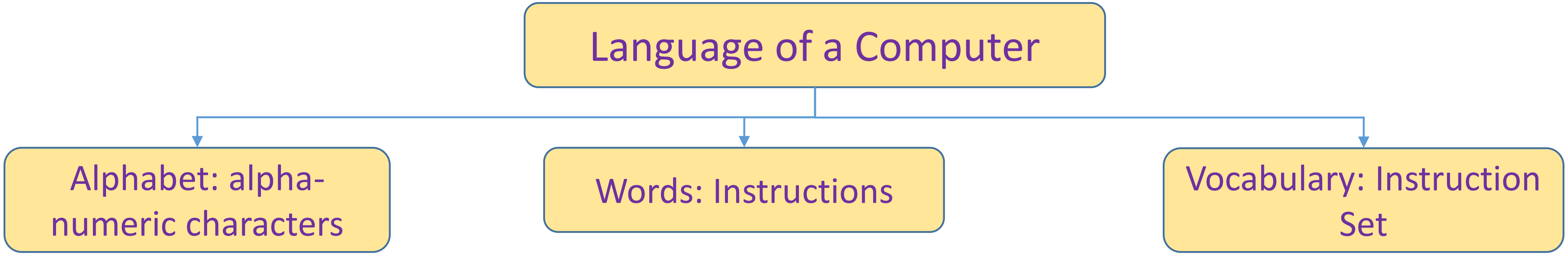
SAMIR LONKAR

To become an efficient speaker, be strong in vocabulary of a native language

English: Learn forming your own sentence and story

Language of the Computer

- To whom we need to speak?
  - Computer?
  - No, its hardware
  - So, we need to speak its language

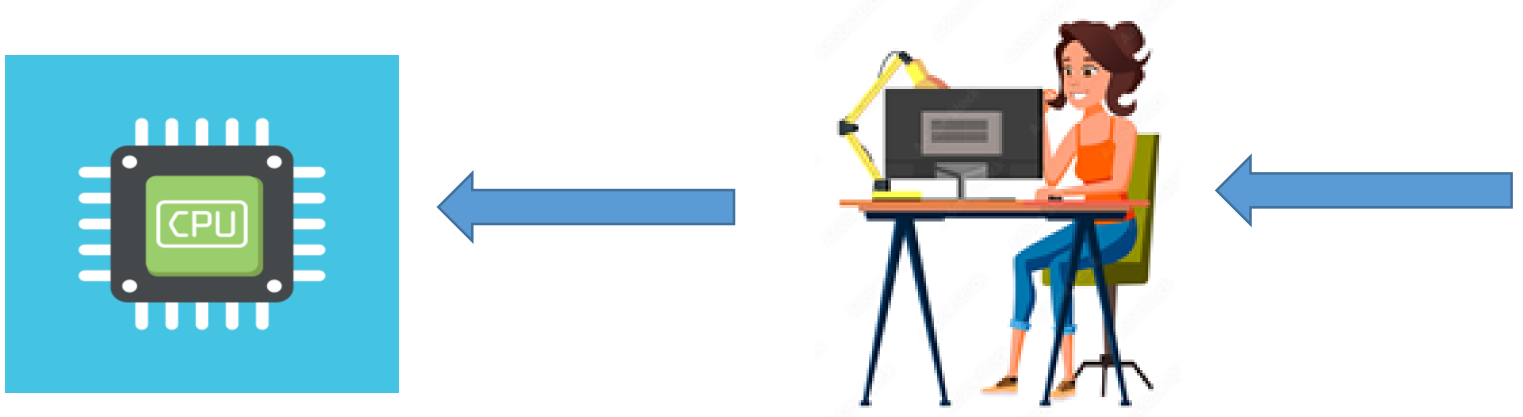


Language of a Computer

Alphabet: alpha-numeric characters

Words: Instructions

Vocabulary: Instruction Set



Own story: Assembly language program

Language of the Computer

- We choose
  - The instruction set developed by MIPS Technologies
  - Designed in the 1980s

- Other instructions sets are
  - ARMv7 : similar to MIPS
  - Intel x86 : used in both the PC and the cloud
  - ARMv8 : extends the address size of the ARMv7 from 32 bits to 64 bits

Classes of Parallelism and Parallel Architectures

- **Parallelism** - primary driving force for the design of computers
  - Other considerations: energy consumption and cost reduction

- **Two broad kinds** of parallelism:
  - Data-level parallelism (DLP): many data items can be operated on at the same time
  - Task-level parallelism (TLP): certain tasks of a work can operate independently

- How can we exploit these parallelisms to take advantage?

Classes of Parallelism and Parallel Architectures

- Four ways to exploit
  - **Instruction-level parallelism:**
    - exploits data-level parallelism, using ideas like **pipelining** and **speculative** execution
  - **Vector architectures, graphic processor units (GPUs), and multimedia instruction sets:**
    - exploit data-level parallelism, using ideas like applying **a single instruction to a collection of data** in parallel
  - **Thread-level parallelism:**
    - exploits either data-level parallelism or task-level parallelism, using ideas like interaction between **parallel threads**
  - **Request-level parallelism:**
    - exploits parallelism among **largely decoupled tasks specified by the programmer** or the operating system.

Classes of Parallelism and Parallel Architectures

- **Flynn** (1966) studied the **parallel computing efforts** in the 1960s,
  - He found **a simple classification** that we follow till date

- Four categories:
  - Single instruction stream, single data stream (**SISD**)—
    - The standard sequential computer,
    - Yet, ILP using **superscalar (more than one execution unit)** and **speculative execution (predicts outcome of condition check)**.
  - Single instruction stream, multiple data streams (**SIMD**)—
    - The same instruction executed by multiple processors using different data streams,
    - DLP using **vector architectures, multimedia extensions** to standard instruction sets, and GPUs.

Classes of Parallelism and Parallel Architectures

- **Flynn** (1966) studied the **parallel computing efforts** in the 1960s,
  - He found **a simple classification** that we follow till date

- Four categories:
  - Single instruction stream, single data stream (**SISD**)—
    - The standard sequential computer,
    - Yet, ILP using **superscalar (more than one execution unit)** and **speculative execution (predicts outcome of condition check)**.
  - Single instruction stream, multiple data streams (**SIMD**)—
    - The same instruction executed by multiple processors using different data streams,
    - DLP using **vector architectures, multimedia extensions** to standard instruction sets, and GPUs.
  - Multiple instruction streams, single data stream (**MISD**)—
    - **No commercial multiprocessor** of this type has been built to date
  - Multiple instruction streams, multiple data streams (**MIMD**)—
    - Multiple processor and each fetches its own instructions and operates on its own data
    - exploits **task-level parallelism**.

Defining Computer Architecture

Designing a computer

- Tasks include
  - **instruction set design**, functional organization, logic design, and implementation.
- Traditionally,
  - **Computer architecture ≈ only instruction set design**
- But, **other considerations**
  - Memory organization
  - the design of CPU
- For example, two processors **AMD Opteron** and the **Intel Core i7** use
  - the same instruction set architectures
    - (**80x86 instruction set,** i386, i486 and i686 architectures are grouped into **x86**)
  - but **different organizations**
    - (in terms of **pipeline and cache organizations**)

Seven dimensions of an ISA

- Class of ISA
    - Nearly all ISAs today are classified as **general-purpose register architectures**, where the operands are either registers or memory locations
- Memory addressing
    - Virtually all desktop and server computers use **byte addressing to access memory operands**
- Addressing modes
    - specify the address of **a memory object**, e.g., Register, Immediate, Displacement
- Types and sizes of operands
    - Supported operand sizes in bits, e.g., **8-bit, 16-bit, 32-bit, 64-bit, 80-bit (extended double)**
- Operations
    - data transfer, arithmetic logical, control, and floating point
- Control flow instructions
    - support for **conditional branches, unconditional jumps, procedure calls and returns**
- Encoding an ISA
    - Two basic choices of encoding: **fixed length and variable length**

Thank You