BCT 2408 Computer Architecture

Fundamentals of Computer Design

Jeff Gicharu – SCT212-0053/2021

Introduction to Computer Architecture

Rapid Progress in Computer Technology

- Computer technology has made incredible progress in the roughly 65 years since the first general-purpose electronic computer was created.
- Example: Today, less than $500 will purchase a personal computer that has more performance, more main memory, and more disk storage than a computer bought in 1985 for 1 million dollars.
- This progress is driven by both technological advancements and innovative computer design.

Microprocessors and Performance Improvement

- The emergence of microprocessors in the late 1970s led to significant performance gains.
- The ability of the microprocessor to ride the improvements in integrated circuit technology led to a higher rate of improvement which was roughly 35% growth per year in performance.
- This growth, combined with cost advantages, led to the dominance of microprocessors in the computer industry.

Introduction to Computer Architecture

Impact of New Architectures

- The development of RISC (Reduced Instruction Set Computer) architectures in the early 1980s raised the performance bar.
- The RISC-based machines focused the attention of designers on two critical performance techniques, the exploitation of instruction-level parallelism (initially through pipelining and later through multiple instruction issue) and the use of caches (initially in simple forms and later using more sophisticated organizations and optimizations).
- This innovation forced other architectures to adapt or become obsolete.

Evolution of Computer Architecture

Historical Performance Growth

- From the mid-1980s to 2002, processor performance grew at an unprecedented rate of over 50% per year.
- This growth was driven by architectural and organizational enhancements, such as pipelining and sophisticated cache designs.
- By 2002, high-performance microprocessors were seven times faster than what could have been achieved with technology alone.

Changing Landscape of Computing

- The dramatic improvement in performance led to new classes of computers, such as personal computers and workstations.
- Minicomputers were replaced by microprocessor-based servers, and mainframes were replaced by multiprocessors.
- Even high-end supercomputers began to be built using collections of microprocessors.

Evolution of Computer Architecture

Modern Challenges

- Since 2002, performance improvement has slowed to about 20% per year due to power limitations, memory latency, and reduced ILP.
- The industry shifted focus from ILP to thread-level parallelism (TLP) and data-level parallelism (DLP) to achieve further performance gains.

Classes of Computers – Desktop Computing

Desktop Computing Overview

- Desktop computing is the largest market in dollar terms, ranging from low-end netbooks to high-end workstations.
- Key characteristics include price-performance optimization, graphics performance, and responsiveness.
- The market is driven by consumer needs for balanced performance across various tasks, including interactive applications.

Market Drivers

- The newest, highest-performance microprocessors often debut in desktop systems due to the demand for high performance at a reasonable cost.
- Applications are well-characterized, though the increasing use of web-centric applications poses new challenges in performance evaluation.
- Example: A high-end workstation might be used for professional graphics design, requiring powerful processors and graphics cards.

Classes of Computers – Servers

Server Computing Overview

- Servers provide large-scale, reliable file and computing services, replacing traditional mainframes.
- Key characteristics include dependability, scalability, and efficient throughput.
- Servers must operate 24/7, making reliability and availability critical.

Market Drivers

- Dependability is crucial, with significant financial losses associated with downtime.
- Scalability is essential as server demands grow, requiring the ability to increase computing capacity, memory, storage, and I/O bandwidth.
- Example: Google's servers handle billions of search queries daily, requiring high availability and scalability.

Classes of Computers – Embedded Systems

Embedded Systems Overview

- Embedded computers are found in everyday devices, from microwaves and printers to cars and cell phones.
- Key characteristics include low cost, low power consumption, and application-specific performance.
- Performance requirements are often real-time, with strict constraints on execution time.

Market Drivers

- Price is a critical factor, with a focus on meeting performance needs at the lowest possible cost.
- Energy efficiency is driven by battery life and heat dissipation concerns.
- Example: A digital set-top box must process video frames in real-time, requiring efficient use of limited resources.

Embedded Systems – Characteristics and Applications

Characteristics of Embedded Systems

1. **Wide Range of Processing Power**: From simple 8-bit processors to high-end 64-bit processors capable of executing billions of instructions per second.
2. **Cost Sensitivity**: Design often focuses on meeting performance needs at the minimum cost.
3. **Real-Time Performance**: Many applications require real-time execution, with strict constraints on maximum execution time.
4. **Power and Memory Optimization**: Minimizing power consumption and memory usage are critical, especially in battery-powered devices.

Applications of Embedded Systems

1. **Consumer Electronics**: Microwaves, washing machines, printers, and digital set-top boxes.
2. **Mobile Devices**: Smartphones and tablets with multimedia capabilities.
3. **Automotive**: Modern cars contain multiple embedded systems for various functions, from engine control to infotainment.
4. **Networking**: Routers, switches, and other network infrastructure devices.

Embedded Systems – Design Considerations

Design Challenges

1. **Low Power Consumption**: Essential for battery-operated devices and to reduce heat generation.
2. **Minimized Memory Usage**: Optimizing code size and data structures to fit within limited memory resources.
3. **Real-Time Constraints**: Ensuring that critical tasks meet their deadlines, even under varying load conditions.
4. **Reliability**: Ensuring robust operation in potentially harsh environments.

Future Trends

1. **Increased Integration**: More functions integrated into a single chip to reduce cost and improve performance.
2. **Energy Efficiency**: Continued focus on reducing power consumption to extend battery life and reduce environmental impact.
3. **Security**: Enhanced security measures to protect against vulnerabilities in connected devices.

Defining Computer Architecture

Role of the Computer Architect

- The computer architect's task is to design a computer that maximizes performance while staying within constraints of cost, power, and availability.
- This involves understanding a wide range of technologies, from compilers and operating systems to logic design and packaging.

Components of Computer Design

- **Instruction Set Architecture (ISA)**: The boundary between software and hardware, defining the instructions visible to the programmer.
- **Organization**: The high-level design aspects, including memory systems, interconnects, and processor design.
- **Hardware**: The detailed logic design and packaging technology.

Instruction Set Architecture (ISA)

ISA Dimensions

- **Class of ISA**: General-purpose register architectures, with operands as registers or memory locations.
- **Memory Addressing**: Byte addressing, with alignment requirements for some architectures.
- **Addressing Modes**: Various ways to specify memory addresses, including register, immediate, and displacement modes.
- **Types and Sizes of Operands**: Support for multiple operand sizes, including integers and floating-point numbers.
- **Operations**: Categories include data transfer, arithmetic/logical operations, control flow, and floating-point operations.

Encoding

- Fixed-length (e.g., MIPS) or variable-length (e.g., x86) instructions, affecting instruction decoding complexity.

Organization and Hardware in Computer Design

Organization

- **Memory System**: Design of main memory, caches, and memory hierarchy.
- **Processor Design**: Pipeline design, cache organization, and instruction scheduling.
- **Interconnects**: Design of buses and networks for communication between components.

Hardware

- **Detailed Logic Design**: Implementation of specific logic circuits and control units.
- **Packaging Technology**: Physical packaging and cooling solutions to manage power dissipation.
- **Performance Optimization**: Techniques to improve performance, such as speculative execution and out-of-order processing.

Challenges Beyond ISA Design

Beyond ISA: The Bigger Picture

- While ISA design is crucial, modern computer architecture involves much more.
- **Microarchitecture**: Detailed design of the CPU, including pipeline stages, cache design, and execution units.
- **System Organization**: High-level design aspects such as memory hierarchy, interconnects, and I/O systems.
- **Hardware Implementation**: Detailed logic design, packaging, power management, and cooling solutions.

Modern Architectural Challenges

- **Power Management**: Balancing performance with power consumption, especially in mobile and embedded devices.
- **Thermal Design**: Ensuring efficient heat dissipation to prevent overheating and maintain performance.
- **Scalability**: Designing systems that can scale up in terms of performance, memory, and storage.
- **Reliability**: Ensuring system dependability, especially in critical applications like servers and embedded systems.

Functional Requirements in Computer Architecture

Key Functional Requirements

- **Application Area**: Tailoring the architecture to specific use cases like desktops, servers, or embedded systems.
- **Software Compatibility**: Ensuring compatibility with existing software ecosystems through instruction set design.
- **Operating System Support**: Providing necessary features to support modern operating systems.
- **Address Space**: Defining the size of addressable memory space.
- **Memory Management**: Implementing efficient memory management techniques like paging and segmentation.
- **Protection**: Ensuring security and isolation between different processes and users.

Standards and Interfaces

- Adhering to industry standards for floating-point arithmetic, I/O interfaces, and networking.
- Supporting various programming languages and development frameworks.

Trends in Technology

Rapid Technological Advancements

- Computer architecture must adapt to rapid changes in technology to remain relevant.
- Key technologies include integrated circuit logic, DRAM, Flash memory, magnetic disks, and networking.

Moore’s Law and its Impact

- **Transistor Density**: Transistor density increases by about 35% per year, doubling every 18 to 24 months.
- **Performance Improvements**: Device speed scales more slowly, but improvements in density lead to significant performance gains.
- **Cost Reduction**: Technological advancements reduce the cost of components, making high-performance computing more accessible.

Trends in Memory and Storage

DRAM Trends

- **Capacity Growth**: DRAM capacity has been increasing by about 25% to 40% per year, doubling every two to three years.
- **Performance**: Improvements in bandwidth and latency, though latency improvements are slower compared to bandwidth.
- **Cost**: DRAM remains a cost-effective solution for main memory, with prices dropping over time.

Flash Memory

- **Rapid Growth**: Flash memory capacity is increasing rapidly, making it a popular choice for mobile and embedded devices.
- **Performance**: Significant improvements in read/write speeds and overall performance.
- **Cost**: Flash memory is becoming increasingly cost-effective, challenging traditional storage solutions.

Trends in Networking and Future Directions

Networking Trends

- **Bandwidth Improvements**: Network bandwidth has seen significant increases, from 10 Mbps to 100 Gbps.
- **Latency Reduction**: Latency improvements, though not as dramatic as bandwidth, are still significant.
- **Technological Advances**: Adoption of new standards and technologies like Ethernet, InfiniBand, and wireless networking.

Future Directions

- **Energy Efficiency**: Increasing focus on energy-efficient designs to reduce power consumption.
- **Scalability**: Designing systems that can scale efficiently to meet growing demands.
- **Integration**: Continued integration of multiple processors and functionalities into single chips.
- **Reliability**: Ensuring system reliability and fault tolerance in the face of increasing complexity.

Performance Trends in Technology

Performance Metrics

- **Bandwidth vs. Latency**: Bandwidth measures the total amount of work done in a given time, while latency measures the time between the start and completion of an event.
- **Improvement Rates**: Bandwidth improvements have outpaced latency improvements significantly across microprocessors, memory, networks, and disks.
- **Rule of Thumb**: Bandwidth grows by at least the square of the improvement in latency.

Impact on Design

- **Microprocessors**: Focus on increasing throughput and parallelism.
- **Memory**: Emphasis on increasing capacity and bandwidth, with moderate latency improvements.
- **Disks**: Continued growth in storage density, with improvements in both bandwidth and latency.
- **Networks**: Significant increases in bandwidth, with latency improvements driven by advancements in switch and transmission technologies.

Scaling of Transistor Performance and Wires

Transistor Performance

- **Feature Size Reduction**: Transistor density increases quadratically with a linear decrease in feature size.
- **Performance Improvement**: Transistor performance improves linearly with decreasing feature size.
- **Impact**: Higher transistor density and performance enable more complex and efficient designs.

Wire Delay

- **Signal Delay**: Wire delay scales poorly compared to transistor performance, increasing proportionally to the product of resistance and capacitance.
- **Challenges**: Wire delay has become a significant design limitation, often more critical than transistor switching delay.
- **Technological Enhancements**: Introducing materials like copper has provided one-time improvements in wire delay.

Trends in Power and Energy in Integrated Circuits

Power Consumption

- **Dynamic Power**: Dominant energy consumption in CMOS chips, proportional to the product of capacitive load, voltage squared, and switching frequency.
- **Static Power**: Increasingly significant due to leakage current, even when transistors are off.
- **Voltage Scaling**: Reducing voltage significantly lowers dynamic power and energy consumption.

Energy Efficiency

- **Energy vs. Power**: Energy is a better metric for comparing processors, as it ties performance to a specific task and execution time.
- **Techniques for Efficiency**: Dynamic Voltage-Frequency Scaling (DVFS), power gating, and designing for typical use cases.

Trends in Cost

Cost Factors

- **Manufacturing Costs**: Decrease over time due to the learning curve and yield improvements.
- **Volume**: Higher volumes reduce cost through increased efficiency and purchasing power.
- **Commoditization**: Standardized components lead to lower costs and increased competition.

Integrated Circuit Costs

- **Wafer Cost**: Dominant factor in the cost of producing integrated circuits.
- **Die Yield**: Number of good dies per wafer, influenced by defect density and process complexity.
- **Mask Costs**: Significant fixed cost for low-volume production, affecting prototyping and debugging.

Cost vs Price

- **Manufacturing Margins**: Narrowing due to commoditization, with R&D and marketing costs impacting final price.
- **Operational Costs**: Significant for large-scale data centers and warehouse-scale computers, emphasizing energy efficiency.