An Introduction to Computer Architecture

Bruce Shriver
University of Tromsø, Norway

Architecture

- refers to the instruction set, resources, and features of a processor that are visible to software programs running on the processor
- *architecture* as defined here is often called the instruction set architecture (ISA)
- the architecture determines what software the processor can directly execute and essentially forms a specification for the microarchitecture

Microarchitecture

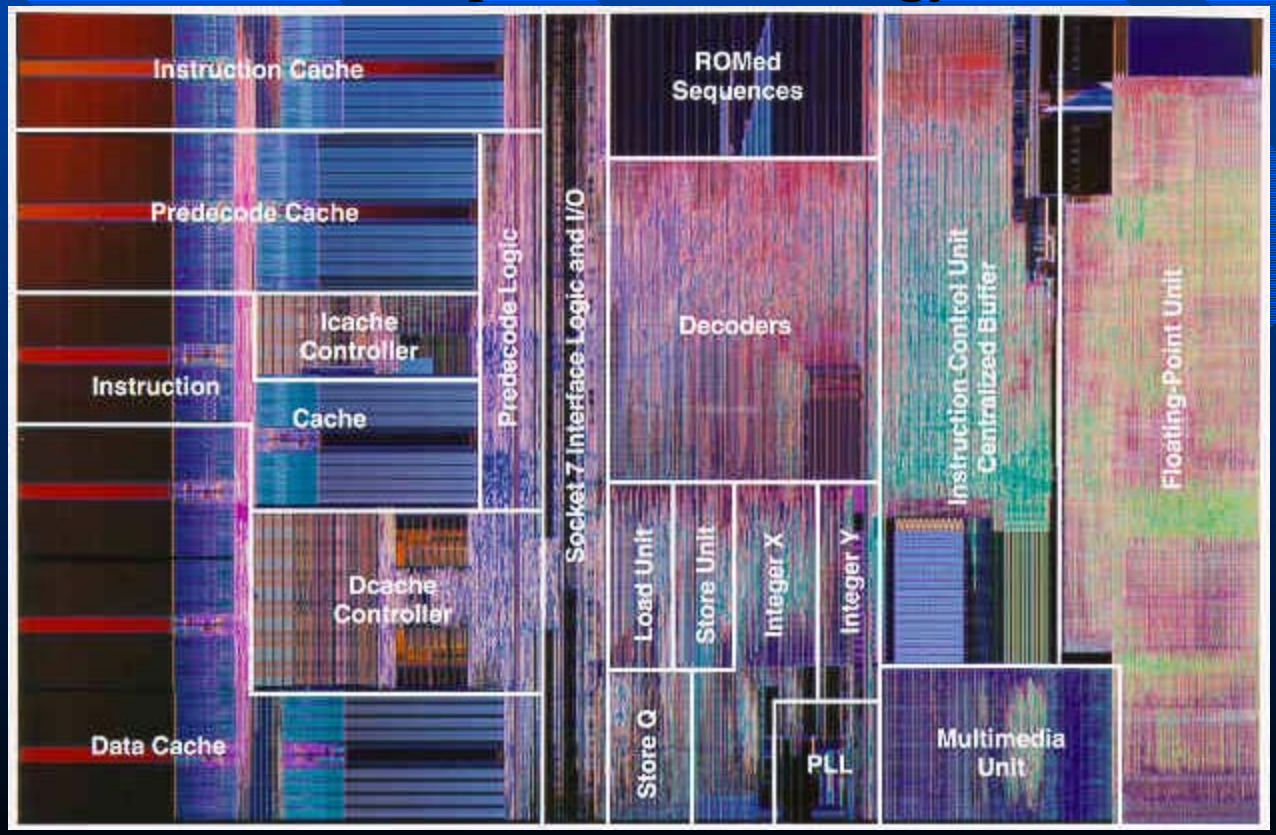
- refers to the set of resources and methods used to realize the architecture specification. The term typically includes the way in which these resources are organized as well as the design techniques used in the processor to reach the target cost and performance goals
- the microarchitecture essentially forms a specification for the logical implementation

Logical Design

- refers to the actual logic and circuit designs used to realize the microarchitecture specifications
- also called logical implementation
- these designs essentially form a specification for the microprocessor chip itself

Microprocessor Chip

- the physical implementation of the logical design in a given semiconductor process technology



A Platform

- consists of a number of key components and interconnections on a motherboard and typically includes a high-performance peripheral bus and ports, main memory, an I/O module, a processor module and appropriate BIOS code
  - the processor module typically includes the processor, processor local bus, optional external cache, and a main memory and bus controller
  - the I/O module includes bus controllers and ports for standardized and optional peripherals

A System

- consists of a platform extended with essential and optional peripherals, an operating system, device drivers including BIOS extensions, other configuration and power management software, and a basic set of applications software
  - examples are a Microsoft Windows 95 desktop system and a Unix-based workstation system

Subject Material in Architecture and Microprocessor Textbooks

- the structure and control of basic building blocks (combinational logic, sequential logic, adders, shifters, decoders, multiplexers, etc.)
- basic control mechanisms (finite state machines, field programmable gate arrays, microcode, state machines, etc.)
- instruction set design (register and memory addressing, highly/ minimally encoded opcodes, compiler issues, operating system issues, architecture state, etc.)
- input/output systems (interrupt, handlers, masking, device controllers, channels, etc.)
- parallelism, synchronization, and consistency issues (pipelining, hazards, multiple functional units, instruction level parallelism, task switching, multi-computer and multi-processor support, etc.)
- cache and memory design (cache organizations and protocols, memory hierarchies, burst and pipeline accesses, virtual memory, coherency, etc.)
- the microarchitecture of contemporary microprocessors
- interconnection and network technology (buses, switches, crossbars, routers, hubs, etc.)

Using the Material

- designing, implementing, and testing
  - architecture
  - microarchitecture
  - logical design
  - chip
- and then
  - platforms
  - systems

A Discipline of Trade-Offs

- computer architecture and platform design both involve balancing a large collection of highly dependent trade-offs.
- computer architects, computer engineers, and platform designers employ a variety of tools to resolve the trade-offs:
  - iterative design techniques
  - simulators, emulators, prototyping models and systems
  - measurement tools
  - formal analysis methods

Iterative Design Techniques

- based on performance and cost trade-offs
  - the trade-offs occur along a number of dimensions, e.g.
    - » the complexity, engineering resource requirements, and impact on the software a particular solution has
    - » the amount of area, power, and pins required

Simulators, Emulators, Prototyping Models and Systems

- verify behavioral and functional specifications, compatibility requirements, and adherence to standards
- explore and analyze alternative solutions, such as impact of the static or dynamic binding of a specific aspect of the processor
- trace-driven detailed performance models have become an increasingly important component of tools in this category

Measurement Tools

- collect data generated by the simulators, emulators, prototype systems or models
  - determine what systems actually do when executing either real or synthetic programs
  - data from address and instruction traces that yields information about the frequency of occurrence of branches and other types of instructions, address modes, exception conditions, and interrupts

Formal Analysis Methods

- attempt to predict one or more aspects of the architecture or platform
  - such as performance, critical speed paths, latency, hit-rates and line-sizes, bandwidth requirements, or availability
    - » of a specific system component
    - » of overall system behavior

Co-Design

- close working relationship among design teams
  - hardware-software co-design
    - » microarchitecture, compiler writers, operating systems implementors
  - hardware-system co-design
    - » extend the above teams to include core logic chipset, BIOS, bus and memory architectures, the motherboard, and any special device controller chips that will be integrated on the motherboard
- co-evolution, testing, and integration of the microprocessor and the software

Summary

- conventional computer architecture course should be extended to include the design, implementation, and testing of platforms (and possibly even systems)
  - the performance of the microarchitecture can be significantly impacted by the characteristics of the platform components that it interact with
- hardware-system co-design is required as contemporary platforms increase in complexity