CPU Structure and Instruction Set Architecture

Introduction to Computer Architecture

Introduction to Computer Architecture

- Computer architecture refers to the design and organization of a computer system's components and their interactions.
- It encompasses both hardware and software aspects, focusing on how these elements work together to execute instructions and process data efficiently.

Computer Architecture vs Computer Organization

- Computer Architecture:
  - Focuses on functional behavior visible to programmers
  - Examples: data type sizes, instruction set

- Computer Organization:
  - Deals with structural relationships not visible to programmers
  - Examples: clock frequency, physical memory size

Computer Architecture vs Computer Organization

- Computers have wide variety of types
  - Single chip microcomputers
  - Supercomputers
- The variation is not only on cost but in size, performance and application
- There are various definitions
  - Computer Architecture: the attributes of the computer system that are visible to programmers (i.e the attributes that have a direct impact on the logical execution of a program)
  - Computer Organization: the operational units and their interconnection that realize the architectural specifications

Language of Instructions

- Humans understand complex languages (e.g., English, Japanese)
- Computers understand simple instructions (e.g., Add A,B; Mul A,B)
- Instruction Set Architecture (ISA): The complete set of instructions a processor can execute
- The semantics of all the instructions supported by a processor is known as its Instruction Set Architecture (ISA)

Features of an Instruction Set Architecture (ISA)

1. Complete: Able to implement all user-written programs

2. Concise: Limited size, typically 32-1000 instructions

3. Generic: Instructions should not be overly specialized

   Eg: adds a number with 15 - instruction is too specialized

4. Simple: Instructions should not be unnecessarily complicated

Designing an ISA: Key Considerations

- Number of instructions
- Functionality of instructions
- Complexity level
- Two main paradigms:
  - RISC (Reduced Instruction Set Computer)
  - CISC (Complex Instruction Set Computer)

RISC vs CISC

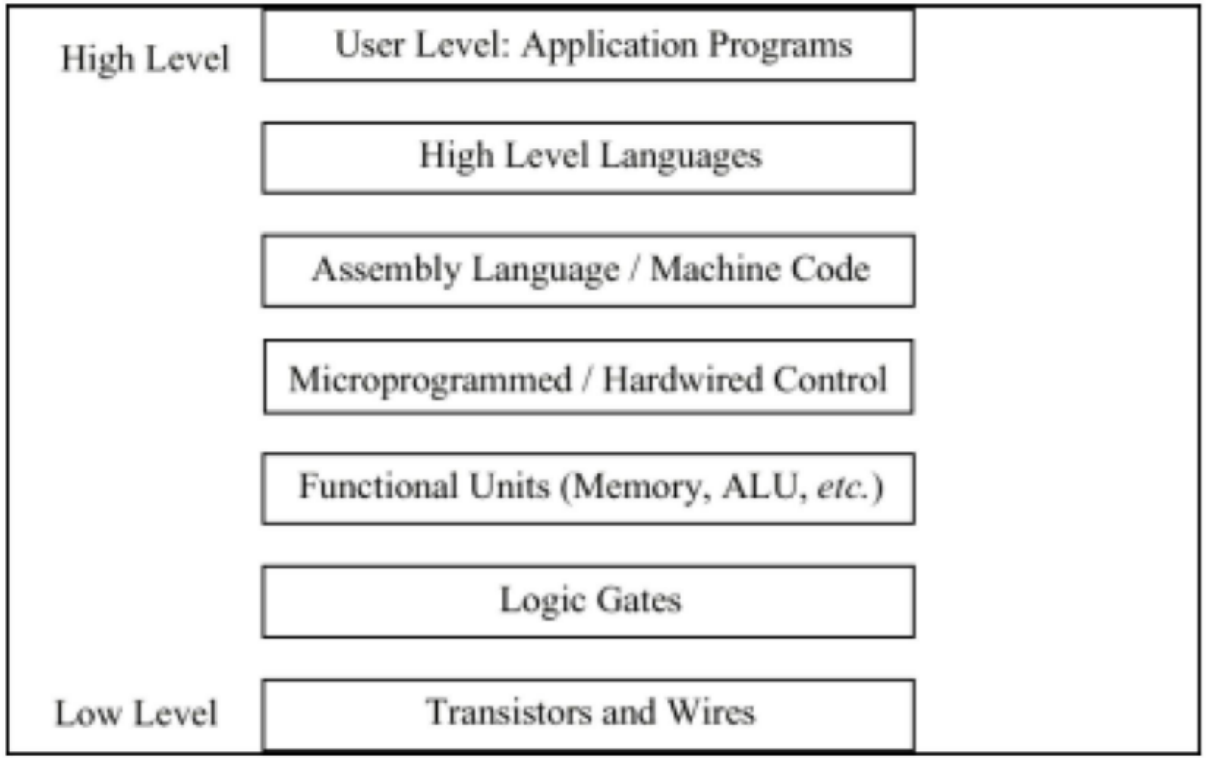
RISC:

- Simple, regular instruction structure
- Fewer instructions (64-128)
- Example: ARM

CISC:

- Complex, irregular instructions
- Multiple operands
- Larger instruction set
- Example: Intel x86

Levels of Machine



Levels of Machine Architecture

- User ISA level: Instructions visible to programmer
- System ISA level: Privileged instructions for OS
- Hardware level: Actual implementation of instructions

Computer Architect's Role

- Design ISA for optimal programming utility and performance
- Design hardware for best instruction implementation
- Use performance measurement tools (e.g.benchmarks)
- Balance performance of CPU, memory, I/O devices, and interconnections
- Meet performance goals at lowest cost

Computer Architect's Role

- Computer architects must design a computer to meet functional requirements as well as price, power, performance, and availability goals.
- Different functional requirements
  - Application area
  - Level of software compatibility
  - Operating system requirements

What is a "better" computer? What is the "best" computer?

- One machine may perform well for a particular aspect but not for another.

- There is no such “better” or “best” computer, It depends on user’s requirements and cost.

Computer cost factors:

- Cost of hardware design
- cost of software design (OS, applications)
- cost of manufacture
- cost to end purchaser

Computer Performance Factors

- Program characteristics (size, frequency of execution)
- Number of users and their expertise
- I/O device requirements
- Technological advancements (more transistors, faster switching)
- Innovative architectures and implementations

Defining Performance: Response Time

- Response time: Time between start and completion of an event
- Execution time is the reciprocal of performance
- Performance comparison: "X is n times faster than Y"

$$\frac{ResponseTimeY}{ResponseTimeX} = n$$

Performance Metrics

For users:

- Response time: Elapsed time for program execution

For computer center managers:

- Throughput: Number of jobs completed per unit time

Performance Metrics

- Is throughput = 1/average response time?
  - Only if NO overlap
  - Otherwise, throughput > 1/av. response time
  - E.g. car assembly factory
    - 4 hours to produce a car (response time)
    - 6 cars per an hour produced (throughput)
    - In general there is no relationship between these 2 metrics
    - Throughput of the factory may increase to 18 cars per an hour without changing time to produce one car.
    - How?

CPU Time: True Measure of Processor Performance

- Time spent running a program
- It doesn't include the time waiting for I/O and running other programs
- CPU time is a true measure of processor performance

$$performace = \frac{1}{CPUTime}$$

$$ElapsedTime = CPUTime + I/OWait$$

Cycles Per Instruction (CPI)

- All computers are constructed using a clock to operate its circuits.
- A “cycle” is technically a pulse synchronization by an internal oscillator, but for our purpose, they are a basic unit that helps understand a CPU’s speed. During each cycle, billions of transistors within the processor open and close.
- The clock speed measures the number of cycles your CPU executes per second, measured in Hz (GHz, MHz)

$$CPI = \frac{CPU Clock Cycles For A Program}{Instruction Count}$$

$$Clock Cycles = IC * CPI$$

CPU Time Formula

$$CPUTime = InstructionCount * CyclesPerInstruction * ClockCycleTime$$

$$CPUTime = \frac{Instructions}{Program} * CPI * ClockCycleTime$$

Factors Affecting Processor Performance

1. Clock cycle (clock rate)
2. Cycles Per Instruction (CPI)
3. Instruction Count (IC)

All three factors equally affect performance and are interdependent.

$$CPUTime = \frac{I}{P} * CPI * ClockCycleTime$$

Performance Improvement Considerations

- Changing one factor often affects others
  - Clock cycle: Influenced by hardware technology and organization
  - CPI: Affected by organization and ISA
  - IC: Impacted by ISA and compiler technology

Performance Calculation Example

Problem:

A program takes 10s on a 400 MHz computer. We want it to run in 6s, but increasing clock rate will increase total clock cycles by 1.2 times.

What's the minimum required clock rate?

Performance Calculation Solution

Given:

$$T_A = 10s, ClockRate_A = 400MHz, T_B = 6s, CPI_B = 1.2CPI_A$$

Using the formula:

$$\frac{T_A}{T_B} = \frac{CPI_A}{CPI_B} * \frac{ClockRate_B}{ClockRate_A}$$

Solving for $ClockRate_B$:

$$ClockRate_B = \frac{10}{6} * 1.2 * 400 = 800MHz$$

Example

Our program takes 10s to run on computer A, which has 400 MHz clock.We want it to run in 6s. The designer says that the clock rate can be increased, but it will cause the total number of clock cycles for the program to increase to 1.2 times the previous value.

What is the minimum clock rate required to get the desired speedup ?

Example

Our program takes 10s to run on computer A, which has 400 MHz clock.We want it to run in 6s. The designer says that the clock rate can be increased, but it will cause the total number of clock cycles for the program to increase to 1.2 times the previous value.

What is the minimum clock rate required to get the desired speedup ?

$$T_A = 10s$$

$$ClockRate_A = 400MHz$$

$$T_B = 6s$$

$$CPI_B = 1.2CPI_A$$

Example

Suppose we have 2 implementations of the same instruction set architecture.

Computer A has a clock cycle time of 10 nsec and a CPI of 2.0 for some program, and computer B has a clock cycle time of 20 nsec and a CPI of 1.2 for the same program.

Which machine is faster for this program?

Key Takeaways

- Computer architecture involves designing both hardware and software components
- ISA is crucial in defining a computer's capabilities
- Performance measurement considers various factors like response time and throughput
- CPU time is a key metric for processor performance
- Improving performance requires balancing multiple interdependent factors