Computer Architecture
CSL3020

Deepak Mishra

http://home.iitj.ac.in/∼dmishra/
Department of Computer Science and Engineering
Indian Institute of Technology Jodhpur

Timings

- Slot: U
- Timing: Mon, Wed, Fri - 12:00 - 12:50 PM

Syllabus

Introduction: Basic computer organization, Components of computer systems, information representation. (3 Lectures)
Central Processing Unit: Arithmetic and Logic Unit; Instruction sets; RISC, CISC, and ASIC/ASIP paradigms; Various addressing modes; Assembly language programming; Instruction interpretation: micro-operations and their RTL specification; CPU design, Hardwired and microprogrammed, Performance issues: Parallel processing, Pipelining, Hazards, Advanced parallelization techniques. Cache Coherence protocols, Multicore Architecture (16 Lectures)
Memory Hierarchy: Memory organization, Various levels of memory architecture and their working principles, Cache memory, Writing strategy, Coherence, Performance issues and enhancement techniques for memory design. (14 Lectures)
Interfacing: I/O transfer techniques: Program controlled, Interrupt controlled and DMA; Introduction to computer buses, Peripherals and current trends in architecture. (9 Lectures)

Books

Text Books

D.A. PATTERSON, J.L. HENNESSY (2008), Computer Organization and Design, Morgan Kaufmann, 4th Edition.

W. STALLINGS (2015), Computer Organization and Architecture: Designing for Performance, Pearson Education India, 10th Edition.

Evaluation Scheme (tentative)

- Quizzes - 15%
- Lab - 25%
- Minor (2 hours) - 20%
- Major (3 hours) - 40%

Attendance

- Attendance policy of the institute will be followed.

Malpractices

- Zero tolerance against plagiarism

LMS

- Google Classroom: doik2ad
- Will be used for sharing material, announcements, discussion etc.

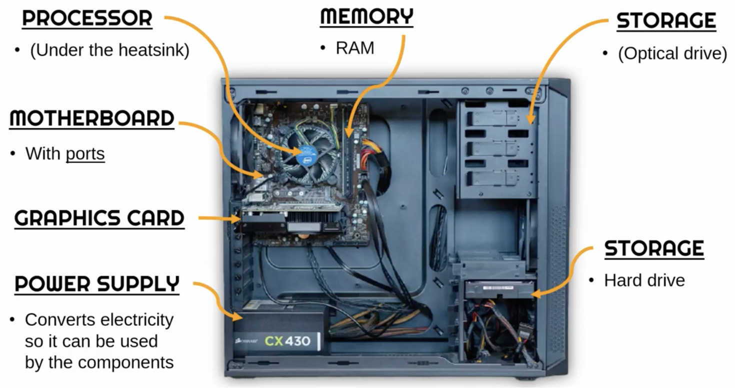
Why should we study Computer Architecture?

Introduction

- *Computer Architecture*: The view of a computer as presented to software designers.
- *Computer Organization*: The actual implementation of a computer in hardware.

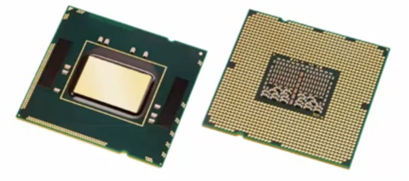
Source: Basic Computer Architecture by SR Sarangi

Computer components

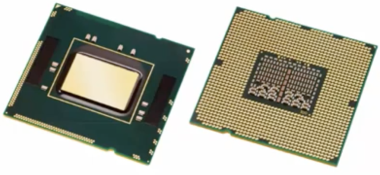


Source: CTEC IT Fundamentals

Computer components: Processor



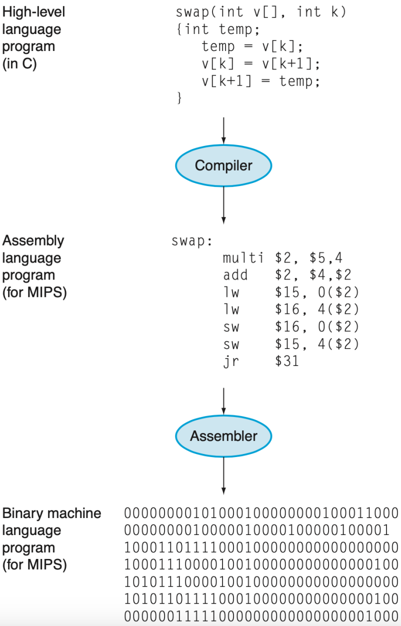
Computer components: Processor



- Processor which is often called as Central Processor Unit (CPU) follows the instructions of a program to perform arithmetic and logical operations, interact with I/O devices etc.

Source: CTEC IT Fundamentals

Introduction



ISA

A processor's electronic circuitry executes machine language (language of bits) instructions.

A processor's electronic circuitry executes machine language (language of bits) instructions.

– It can support a fix set of instructions.

ISA

A processor's electronic circuitry executes machine language (language of bits) instructions.

– It can support a fix set of instructions.

This set of instructions is known as Instruction Set Architecture (ISA).

More formally, ISA is an abstract interface between hardware and software that encompasses all the information necessary to write a machine language program.

Performance

How do we compare two computers?

How do we compare two computers?

Two notions of performance

- *Response time*: Also called execution time. The total time required for the computer to complete a task.
- *Throughput*: The number of tasks completed per unit time.

Performance

From architecture point of view we are more interested in CPU (execution) time – The actual time the CPU spends computing for a specific task.

Performance

From architecture point of view we are more interested in CPU (execution) time – The actual time the CPU spends computing for a specific task.

The Classic CPU Performance Equation

$$\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{Clock cycle time}$$

Performance

From architecture point of view we are more interested in CPU (execution) time – The actual time the CPU spends computing for a specific task.

The Classic CPU Performance Equation

$$\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{Clock cycle time}$$

- Instruction count - The number of instructions executed by the program
- CPI - Average number of clock cycles per instruction
- Clock cycle time - The time for one clock period, usually of the processor clock.

Example

Example 1: Suppose we have two implementations of the same instruction set architecture. Computer A has a clock cycle time of 250 ps and a CPI of 2.0 for some program, and computer B has a clock cycle time of 500 ps and a CPI of 1.2 for the same program. Which computer is faster for this program and by how much?

Example

Example 1: Suppose we have two implementations of the same instruction set architecture. Computer A has a clock cycle time of 250 ps and a CPI of 2.0 for some program, and computer B has a clock cycle time of 500 ps and a CPI of 1.2 for the same program. Which computer is faster for this program and by how much?

Ans. Computer A is 1.2 times faster than Computer B.

Example

Example 2: A compiler designer is trying to decide between two code sequences for a particular computer. The hardware designers have supplied the following facts: The instructions can be divided into three classes according to their CPI (class A, B, and C). Their respective CPI values are 1, 2, and 3. For a particular high-level language statement, the compiler writer is considering two code sequences that require the following instruction counts:
Code seq. 1: A - 2, B - 1, C - 2
Code seq. 1: A - 4, B - 1, C - 1

Which code sequence executes the most instructions? Which will be faster?What is the CPI for each sequence?

Performance

Example 2 Ans. Sequence 2 executes more instructions. Sequence 2 is faster. The CPI values are 2.0 and 1.5, respectively.

Language of Bits

A computer understands the language of bits.

Bit	0 or 1
Byte	8 bits
Word	4 bytes
kiloByte	1024 bytes
megaByte	10^6 bytes

Basic logical operations

AND
OR
NAND
NOR
NOT
XOR

De Morgan's Laws

$$\overline{A+B} = \overline{A}.\overline{B}$$

$$\overline{A.B} = \overline{A} + \overline{B}$$

Consensus Theorem

$$X.Y + \overline{X}.Z + Y.Z = X.Y + \overline{X}.Z$$

Number Systems

Binary
Decimal
Octal - starts with 0
Hexadecimal - starts with 0x

Convert 111000101111 to the hex format : 1110 0010 1111 = 0xE2F

Negative Numbers

Binary Representation negative numbers

Sign-Magnitude based representation
3: 0011
-3: 1011

1's Complement representation
3: 0011
-3: 1100

Representing negative numbers

2's Complement representation
$F(-u) = 2^n - F(u)$
3: 0011
-3: 1101

Advantages:

- Unique representation of 0
- Subtraction is easy

Overflow

Real Numbers

Floating point representation

3.14
1.0×10^{-9}
3.45×10^{10}

Floating point representation

3.14
1.0×10^{-9}
3.45×10^{10}

Generic form $N = \sum_{-n}^{n} x_i 10^i$

$9.56 = 9 \times 10^1 + 5 \times 10^{-1} + 6 \times 10^{-2}$

Floats

Floating point representation

Generic form in binary $N = \sum_{-n}^{n} x_i 2^i$

$0.75 \Rightarrow 0.11 \Rightarrow 1 \times 2^{-1} + 1 \times 2^{-2}$

$6.375 \Rightarrow 110.011 \Rightarrow 2^2 + 2^1 + 2^{-2} + 2^{-3}$

Normalized Form

In base 2 the normalized form is

$$1.xxxxxx \times 2^{yyyy}$$

$$N = (-1)^S \times F \times 2^E$$

$S \rightarrow$ Sign bit
$F \rightarrow$ Significand
$F = 1 + M$, $M \rightarrow$ Mantissa
$E \rightarrow$ exponent

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	exponent								fraction																						
1 bit	8 bits								23 bits																						

Double Precision

A double number is represented using 2 words (64 bits)

$S \rightarrow 1$ bit
$E \rightarrow 11$ bits
$M \rightarrow 52$ bits

IEEE 754 Format

No need to waste 1 bit representing (1.) in the significand, instead we can just save the mantissa bits

$S \rightarrow 1$ bit
$E \rightarrow 8$ bits
$M \rightarrow 23$ bits

Biased representation

E = exponent + bias For float bias is 127 as

$$N = (-1)^S \times (1 + M) \times 2^{(E-127)}$$

Floating-Point Addition

Steps:

- Compare the exponents of the two numbers; shift the smaller number to the right until its exponent would match the larger exponent
- Add the significands
- Normalize the sum, either shifting right and incrementing the exponent or shifting left and decrementing the exponent