

**Book** Computer Organisation and Design

David A Patterson

4th Edition

Tajmior

2015331572

1st chapter - Computer Abstraction and technology.

moore's law : every two years transistors number increases by the powers 2.

- ① Desktop
- ② Server - cloud computing. (storage)
- ③ Super computer - NASA, genome.
- ④ Embedded computer - mobile, washing machine.  
handi case use.

TOP:  
Assembly language, performance, parallel processing, MIPS code.

Q: What is computer in device it is called

o If we embed a computer in device, it is called embedded computer.

→ How programs are translated into the machine language.

→ the hardware/software interface

→ program performance.

→ Hardware improve performance.

→ parallel processing.

Performance:

Algorithm: Determine number of operations

Programming language, compiler, architecture:

Processor and memory system:

I/O system:-

→ Application software

→ written in high level language

K. Here Ax is  
cation Ax can  
be remapped.

H L  
00 05  
00 0A →  
00 00  
00 02

2 Ax is multi  
Ax carries -  
(is a 16 bit

E  
H  
G  
D  
S  
00

→ System software!

Compiler → HLL → machine

Operating system → Handling I/O

memory/storage

Scheduling tasks and  
sharing resources.

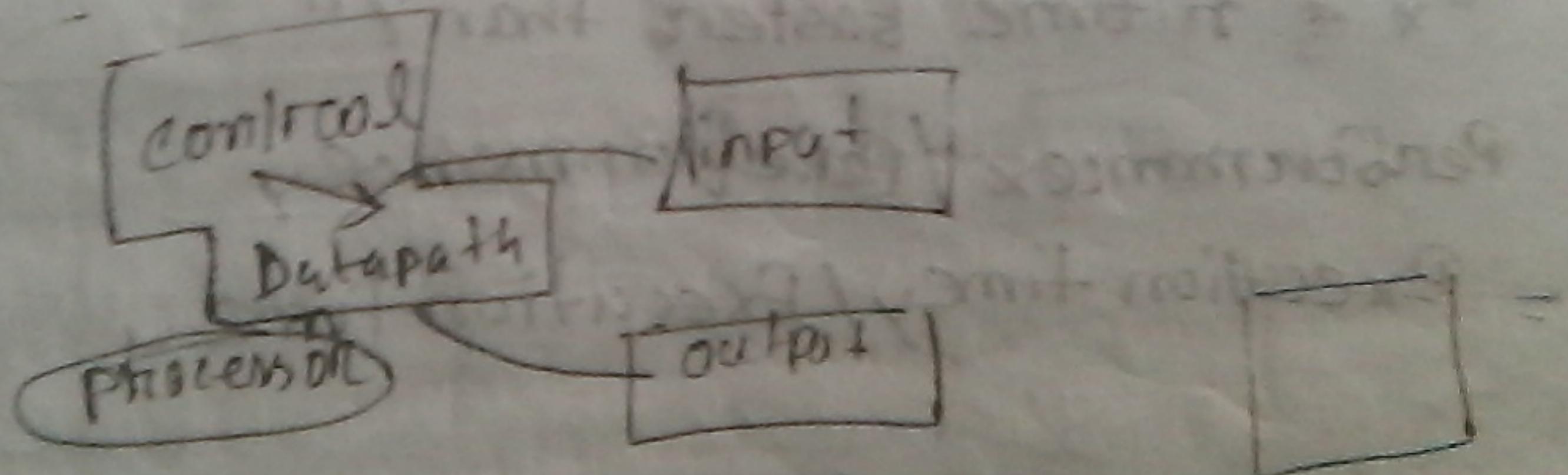
Hardware - Processor, memory, I/O  
controllers.

System of system → Software of system → OS

WORA.

- \* Level of program  $\rightarrow$  high.
  - ⇒ ~~Lower~~ Machine Language: Binary string as instructions.
  - ⇒ Assembly language: ADD SUB
    - uses assembler.
  - ⇒ High level: Makes the programme independent enough to run anywhere.

Device:



Processor :-

Processor :-

Data path: performs operation on data.

Control: Sequence data path, memory

Cache memory Frequently used instruction

Small part static random access

Memory

Abstraction :-

\* helps us deal with complexity

1 Instruction set architecture

\* Application binary interface

\* Implementation

Memory:

Volatile → Losses instruction and data

when power off

→ Non-volatile memory:

Magnetic disk

Ram memory

Optical disk (CDROM, DVD)

Performance:

- \* Response time?
- \* throughput?

Relative Performance:

\* Define performance =  $1/\text{Execution time}$   
"x is n times faster than y"

Performance<sub>x</sub> / Performance<sub>y</sub>

$$= \frac{\text{Execution time}_y}{\text{Execution time}_x}$$

## Measuring Execution Time:

④ Elapsed time:

- total response time, including all aspects
- processing, I/O, OS overhead, idle time
- determines system performance.

⑤ CPU time:

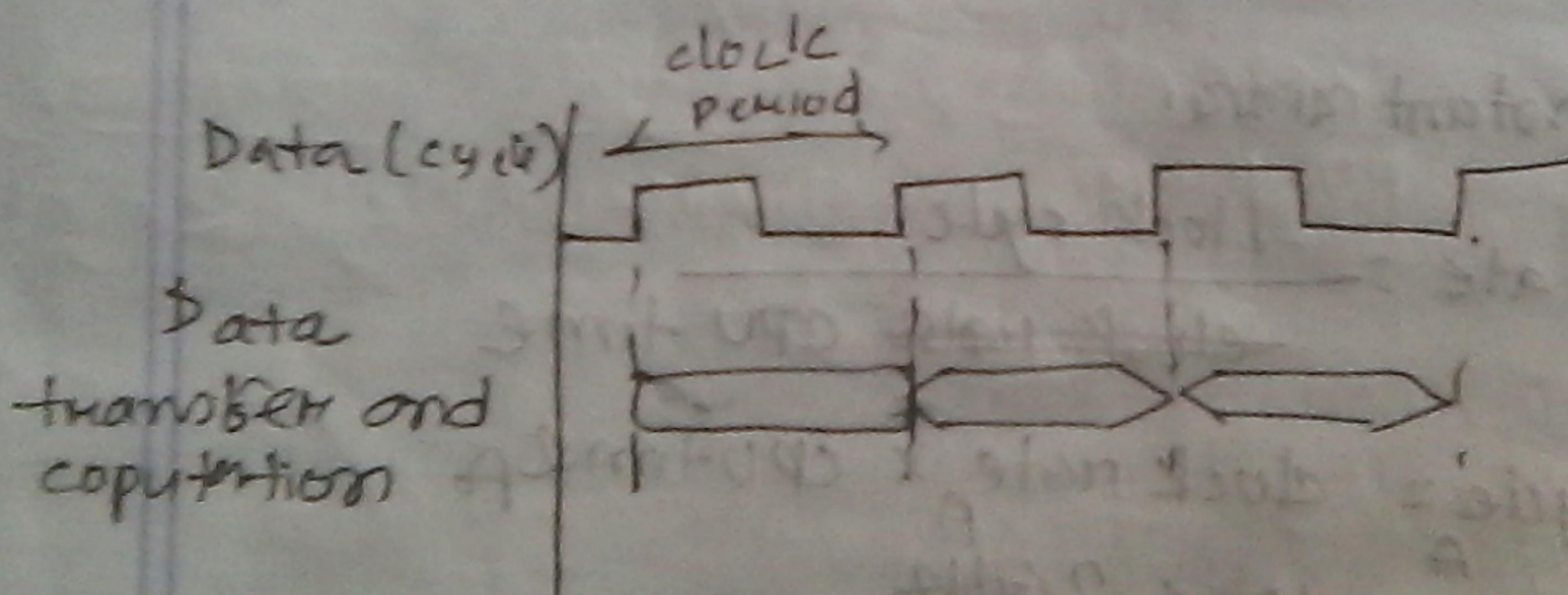
- time spent processing a given job
- Discounts I/O operations, other job share.
- Comprises user CPU time and system CPU time
- Different programs are affected differently by user and system performance.

CPU time = user CPU + system CPU

elapsed time = + I/O

~~clock~~ ~~clocking~~

Operation of digital hardware governed by a constant rate clock.



\* Clock period: duration of a clock cycle

$$250 \text{ ps} = 0.25 \text{ ns} = 250 \times 10^{-12} \text{ s}$$

CPU time:

CPU time = CPU clock cycles  $\times$  Clock cycle time

$$= \frac{\text{CPU clock cycle} - \text{Overhead}}{\text{Clock Rate} \rightarrow \text{Overhead}} \quad \left. \begin{array}{l} \text{Performance} \\ \text{overhead} \end{array} \right\}$$

Performance improved by:

~~= 40 MHz~~  
Instruction Count and CPI:

Clock cycle = Instruction count \* cycle per instruction

CPU time = Instruction count \* CPI \* clock cycle time

$$= \frac{\text{Instruction count} \times \text{CPI}}{\text{clock Rate.}}$$

## CPI in more detail

If different instruction classes take different number of cycles

clock cycles:

A	B	C
1	2	3
2	1	2
4	4	1

Sequence 1: SC=5

clock cycles:

$$2 \times 1 + 1 \times 2 + 2 \times 3$$

$$= 10$$

$$\text{Avg} = 10/5 = 2.0$$

Sequence 2: SC=6

clock cycles

$$= 4 \times 1 + 1 \times 2 + 1 \times 3$$

$$= 9$$

$$\text{Avg CPI} = 9/6 = 1.5$$

## Performance summary

$$\text{CPU time} = \frac{\text{Instructions}}{\text{program}} \times \frac{\text{clock cycle}}{\text{Instruction}} \times \frac{\text{seconds}}{\text{clock cycle}}$$

performance depend on:

- ① Algorithm
- ② Programming language
- ③ Compiler
- ④ Instruction set architecture

2-10-2018

## Power Trends

$$\text{Power} = \text{capacitive load} \times \text{voltage}^2 \times \text{frequency}$$

CMOS - Technology - stages

power trend  $\propto \text{area} \propto n^2$

## Multiprocessing %

One program uses multiple core processor.  
Some drawbacks are:

Manufacturing %:

Pfaffall: Amdahl's Law → 50% - mother example

MIPS as performance metric:

~~MIPS~~

Performance → math

TOT %

Chapter 2 (principal), convert  
Instruction language of the computer.

performance  $\rightarrow$  math

TOP:

chapter 2 (principal), convenient  
Instruction language of the computer -

\* Arithmetic Operation question

MIPS code.

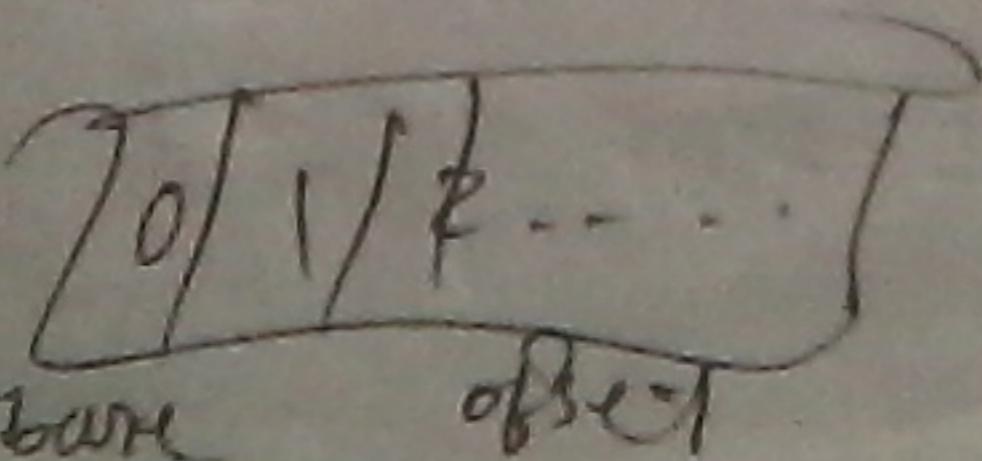
add a, b, c  
 $a = b + c$

design principle

$\rightarrow$  similar example.

mips code conversion

base, offset



memory position

$$= \text{base} + \text{offset}$$

3-11-1618

operand : registers operand  
memory operand }  
immediate operand }.

add \$50, \$1, 5

\* make + the common case first.

The constant zero 0

add \$50 , \$51, \$ zeros

Sign extension .

Sign extension :

Representing instructions

add \$50 \$51 \$zero



numerical  
value.

Converting numerical value

$2^5$       32 bit

11111  
10101010

32 bit register

your register or want more help contact me

3 bit 45 convert into 10 !

Format register:

Binary Hexa

in format field MIPS code (or machine code)

MIPS 1 format Instructions:

Format: