

Computer Architectures

Zhu Wen-Jun

Faculty of Information Technology

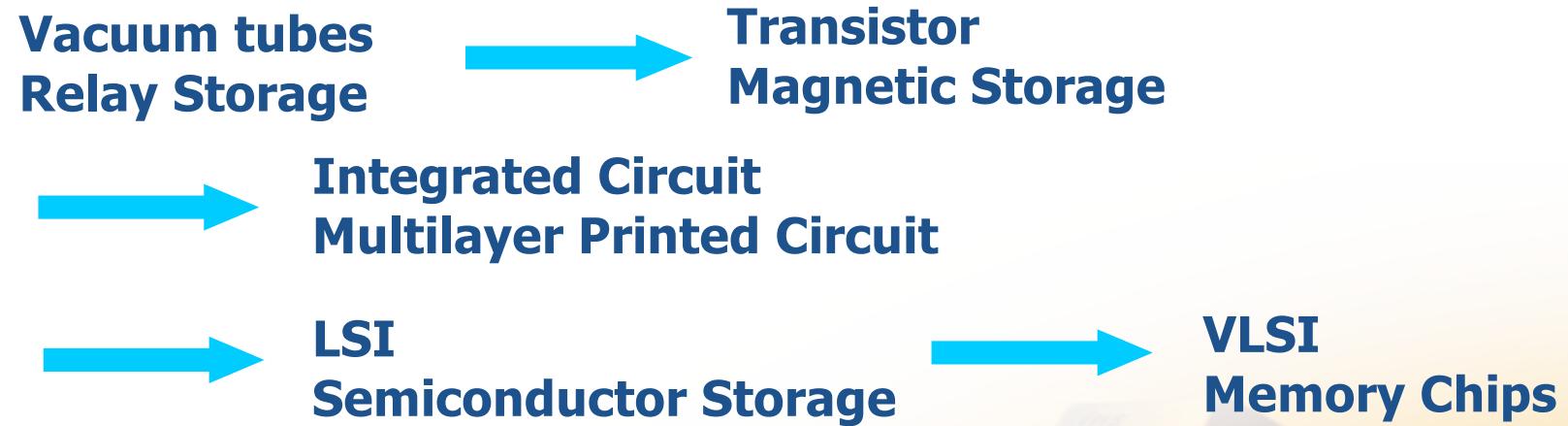
References

- | **John L. Hennessy, David A. Patterson, Computer Architecture: A Quantitative Approach. Fifth Edition.**
- | **David A. Patterson, John L. Hennessy, Computer Organization & Design : The Hardware/Software Interface, Fourth Edition.**
- | 方娟, 计算机系统结构, 清华大学出版社

Chapter1 Concepts of Computer Architecture

I Development of Computer

Device



Structure

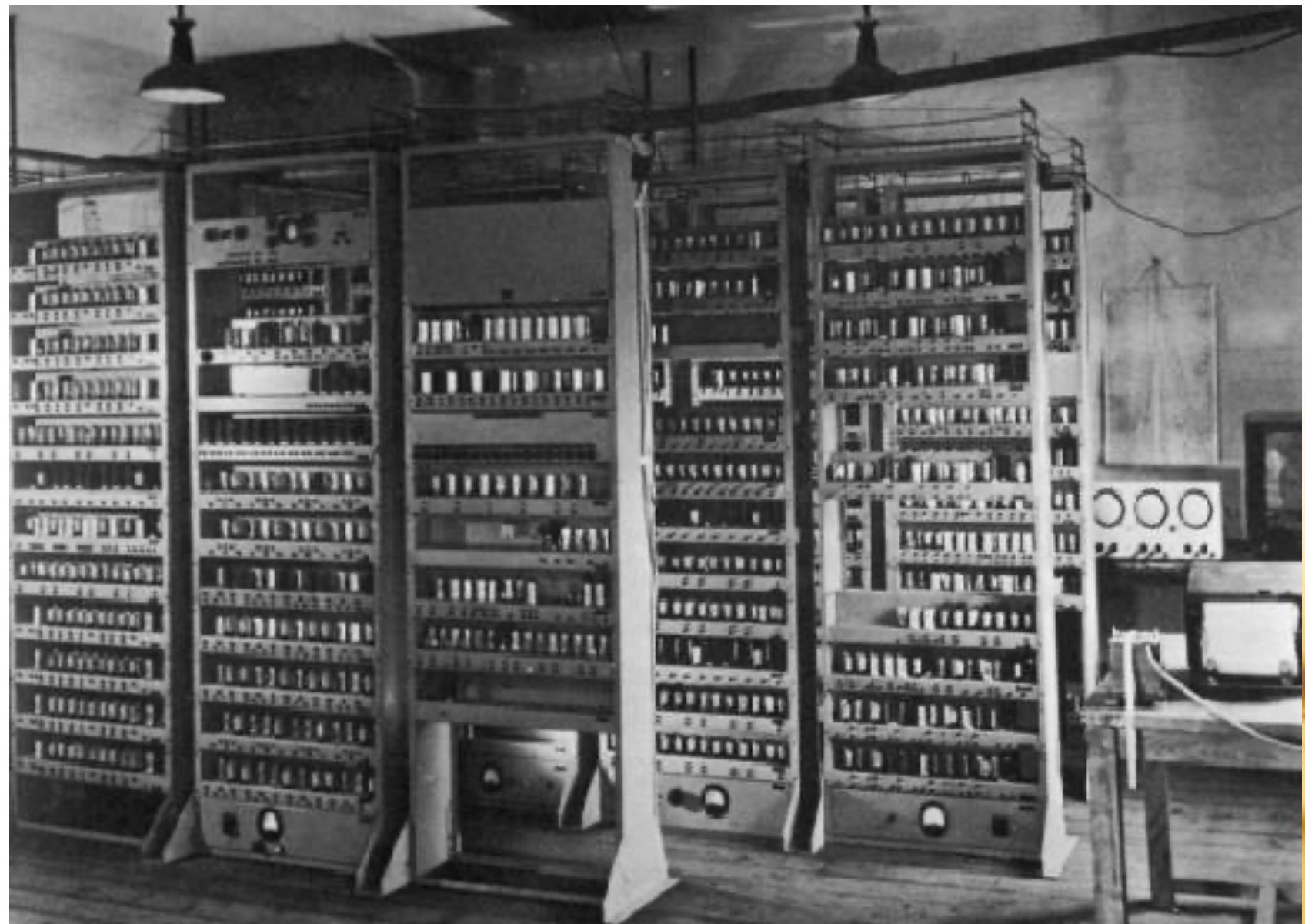


Chapter1 Concepts of Computer Architecture

Technology



Computing Devices Then...



EDIAC, University of Cambridge, UK, 1949

Computing Systems Today

❖ The world is a large parallel system

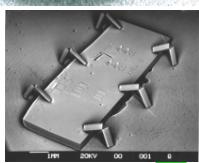
- Microprocessors in everything
- Vast infrastructure behind them



Internet
Connectivity



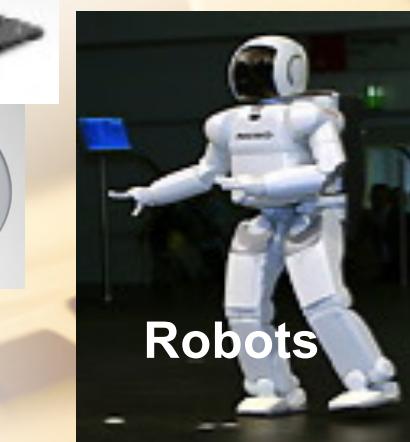
Sensor
Nets



MEMS for
Sensor Nets



Databases
Information Collection
Remote Storage
Online Games
Commerce



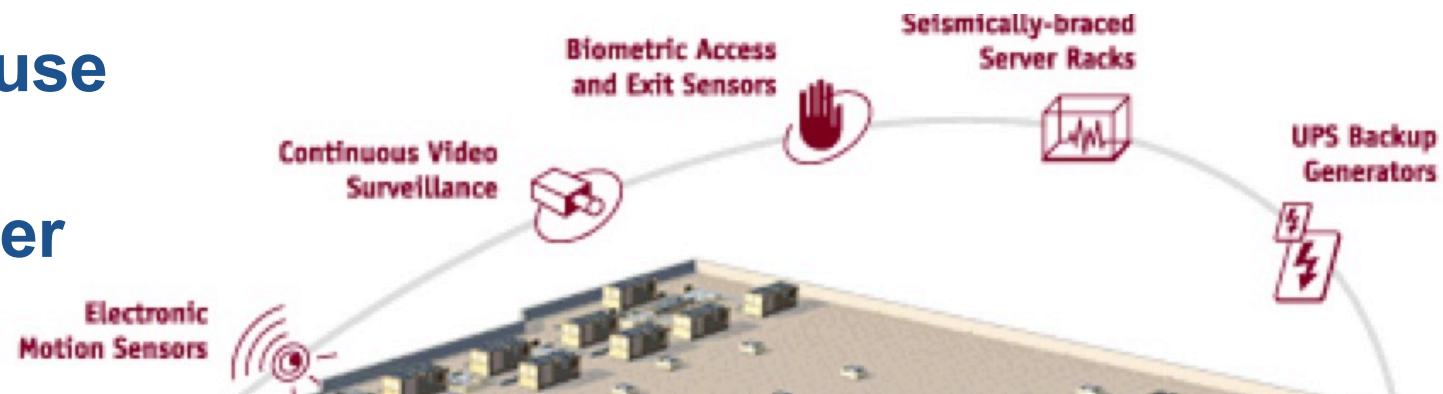
New “Great Ideas”

Devices

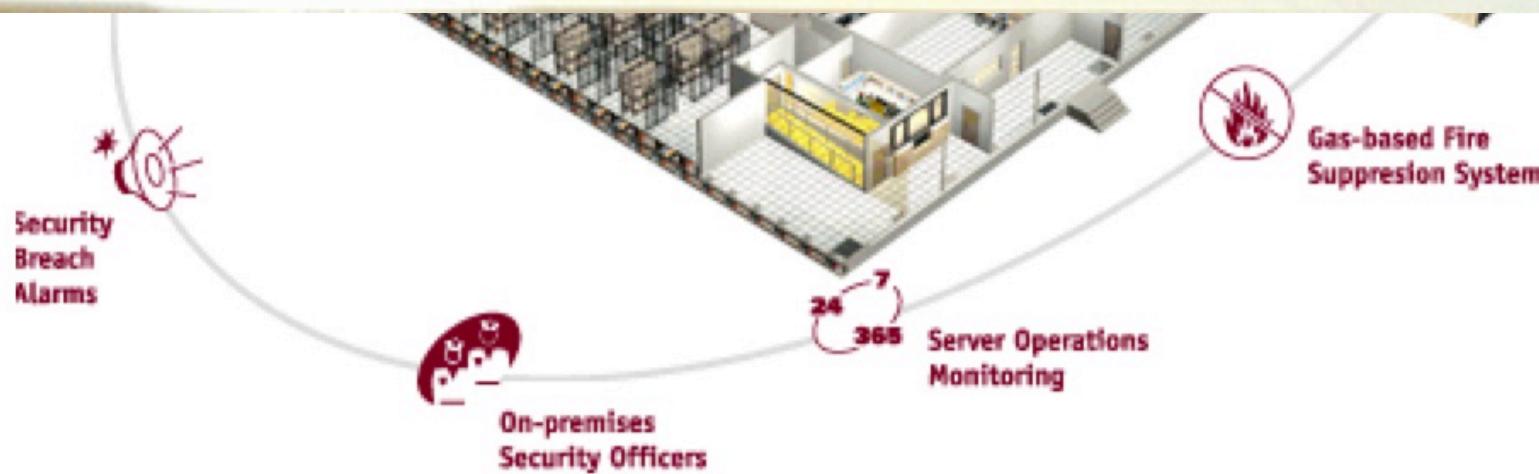


© 2011 SAS International

Warehouse Scale Computer



My other computer is a data center



New “Great Ideas”

Software

- ❖ Parallel Requests
Assigned to computer
- ❖ Parallel Threads
Assigned to core
e.g., Lookup
- ❖ Parallel Instruction
>1 instruction @ one time
e.g., 5 pipelined instructions
- ❖ Parallel Data
>1 data item @ one time
e.g., Add of 4 pairs of words
- ❖ Hardware Descriptions
All gates functioning in parallel at same time
- ❖ Programming Languages

Hardware

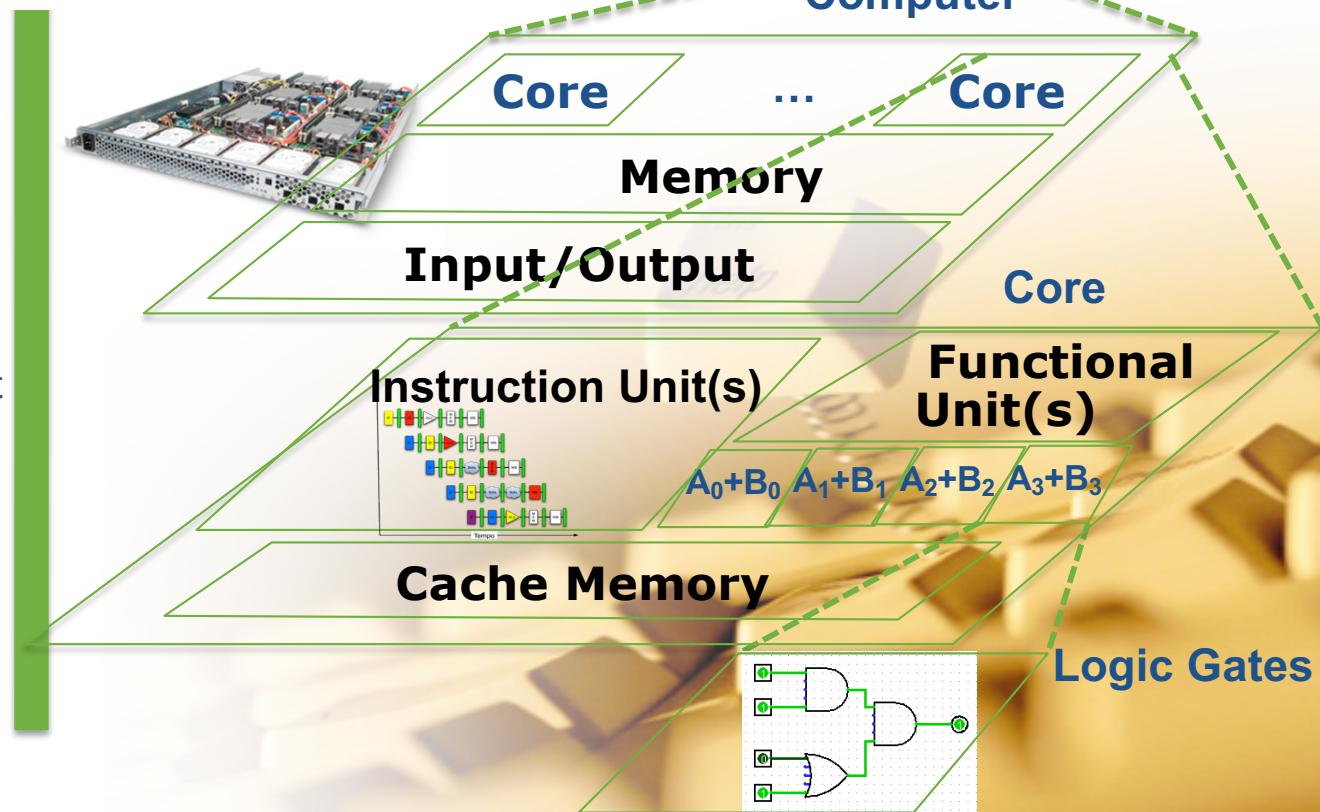
Warehouse Scale Computer



Smart Phone

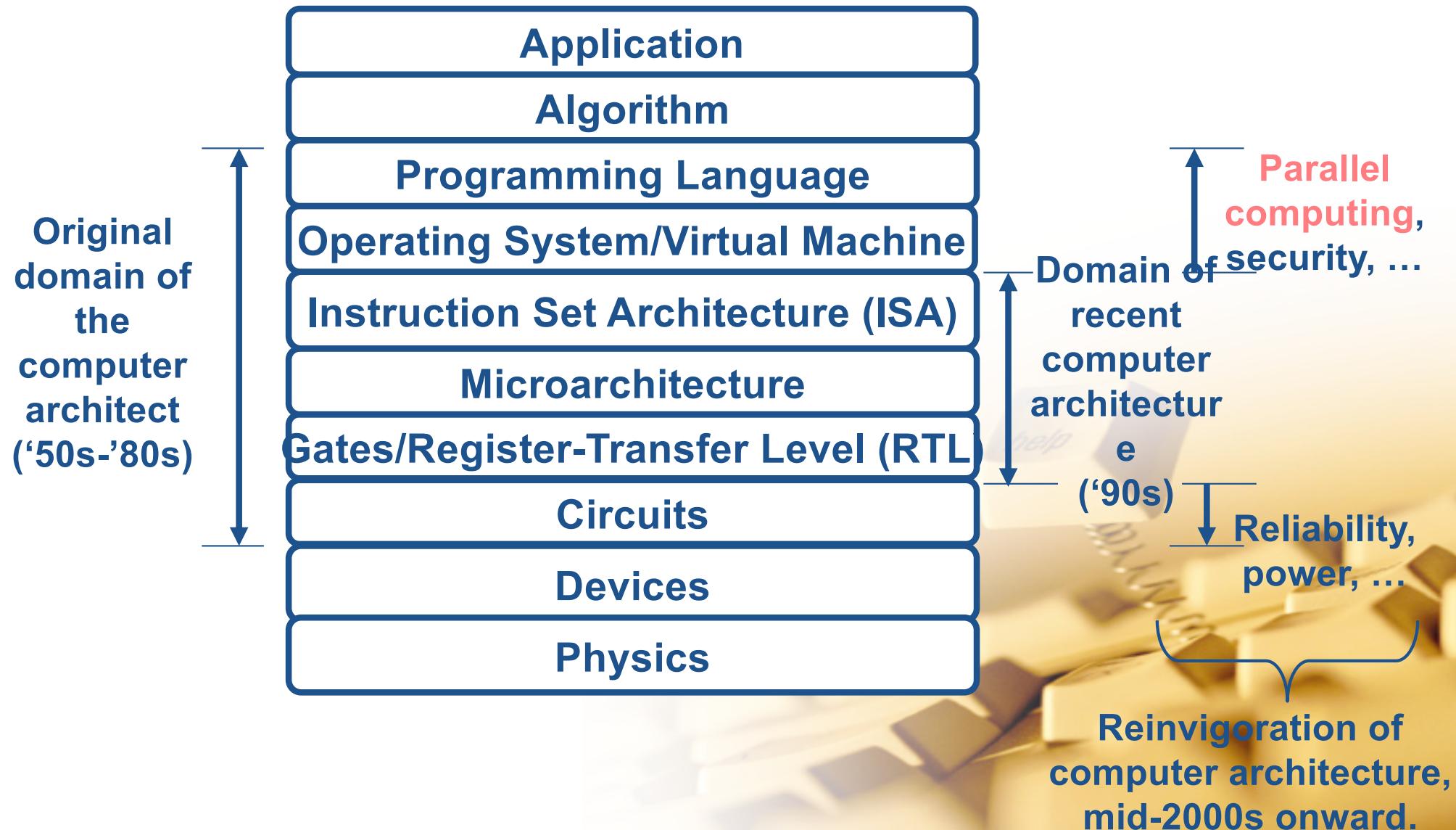


*Leverage
Parallelism &
Achieve High
Performance*



Chapter1 Concepts of Computer Architecture

I Hierarchy of Computer System



Chapter1 Concepts of Computer Architecture

■ Definition of Computer Architecture

Architecture

the attributes of a [computing] system as seen by the programmer, *i.e.* the conceptual structure and functional behavior, as distinct from the organization of the data flows and controls , the logic design, and the physical implementation.

– Amdahl, Blaaw, and Brooks, 1964

Computer Architecture

the attributes of a computing system as seen by the Machine Language Designer or Compile Program Designer.



Chapter1 Concepts of Computer Architecture

- The difference between application programmer and machine language designer**

Application Programmer—The function of computer is to identify and execute application language.

Machine language designer—The Special function of hardware device



Chapter1 Concepts of Computer Architecture

■ Transparency

A kind of thing and attribute which originally exists many differences, from a certain perspective the differences can not be seen, referred to as "transparency".



Chapter1 Concepts of Computer Architecture

■ Computer Architecture, Organization and Implementation

Computer Architecture – the attributes of a computing system as seen by the Machine Language Designer

Computer Organization – Logical Implementation of Computer Architecture

Computer Implementation – Physical Implementation of Computer Organization



Chapter1 Concepts of Computer Architecture

■ Computer Architecture

- ❖ **Data Representation**
- ❖ **Register Definition**
- ❖ **Instruction System**
- ❖ **Interrupt System**
- ❖ **Memory System**
- ❖ **Input / Output Structure**
- ❖ **Operating Mode**
- ❖ **Information Protection**



Chapter1 Concepts of Computer Architecture

■ Computer Organization

- ❖ **Data Path Width**
- ❖ **Special Components**
- ❖ **Functional Components Parallelism**
- ❖ **Control Mechanism**
- ❖ **Buffering Technology**
- ❖ **reliability technology**



Chapter1 Concepts of Computer Architecture

■ Computer Implementation

- ❖ Physical Implementation of Logical Design**

■ Difference and Relation among the above three

Ex1: Instruction System

Ex2: Multiply Instruction

Ex3: Main Memory System



Chapter1 Concepts of Computer Architecture

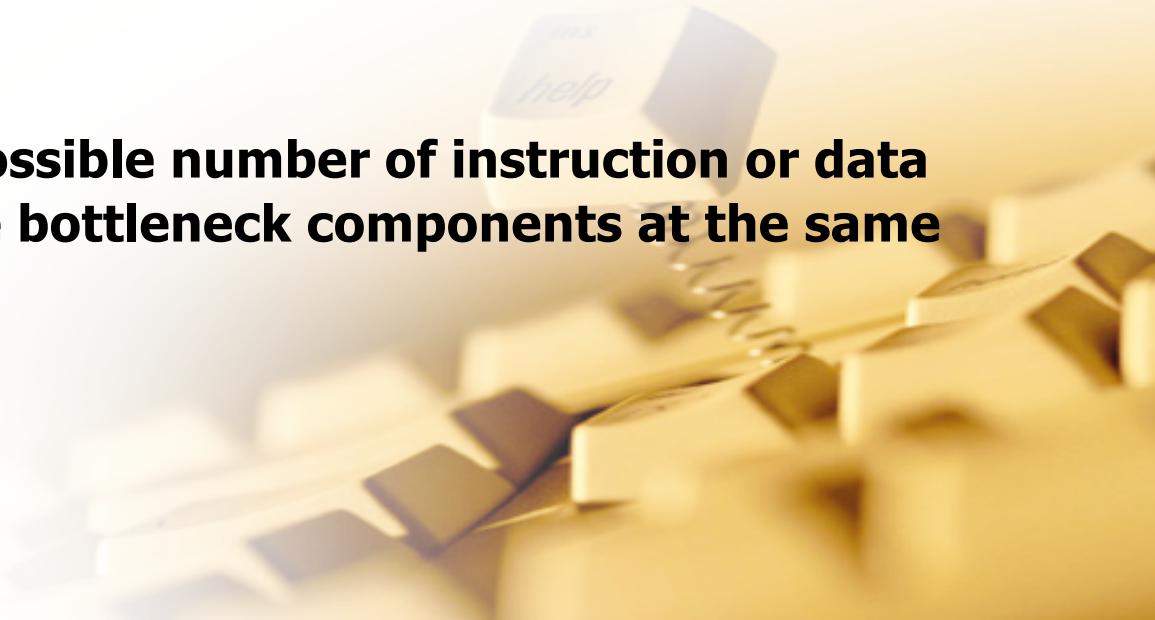
■ Classification of Computer Architecture

1. FLYNN Classification (Michael J.Flynn,1966)

Instruction Stream

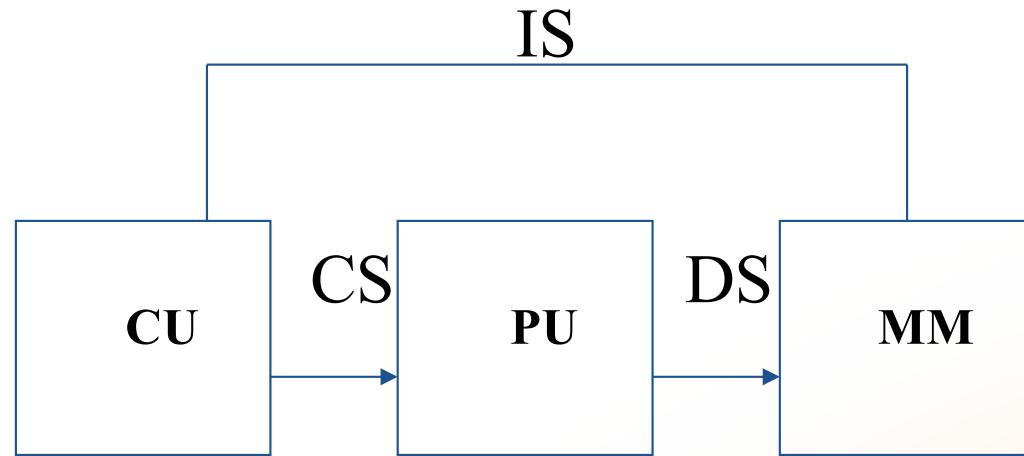
Data Stream

Multiplicity: the largest possible number of instruction or data on the system performance bottleneck components at the same execution stage



Chapter1 Concepts of Computer Architecture

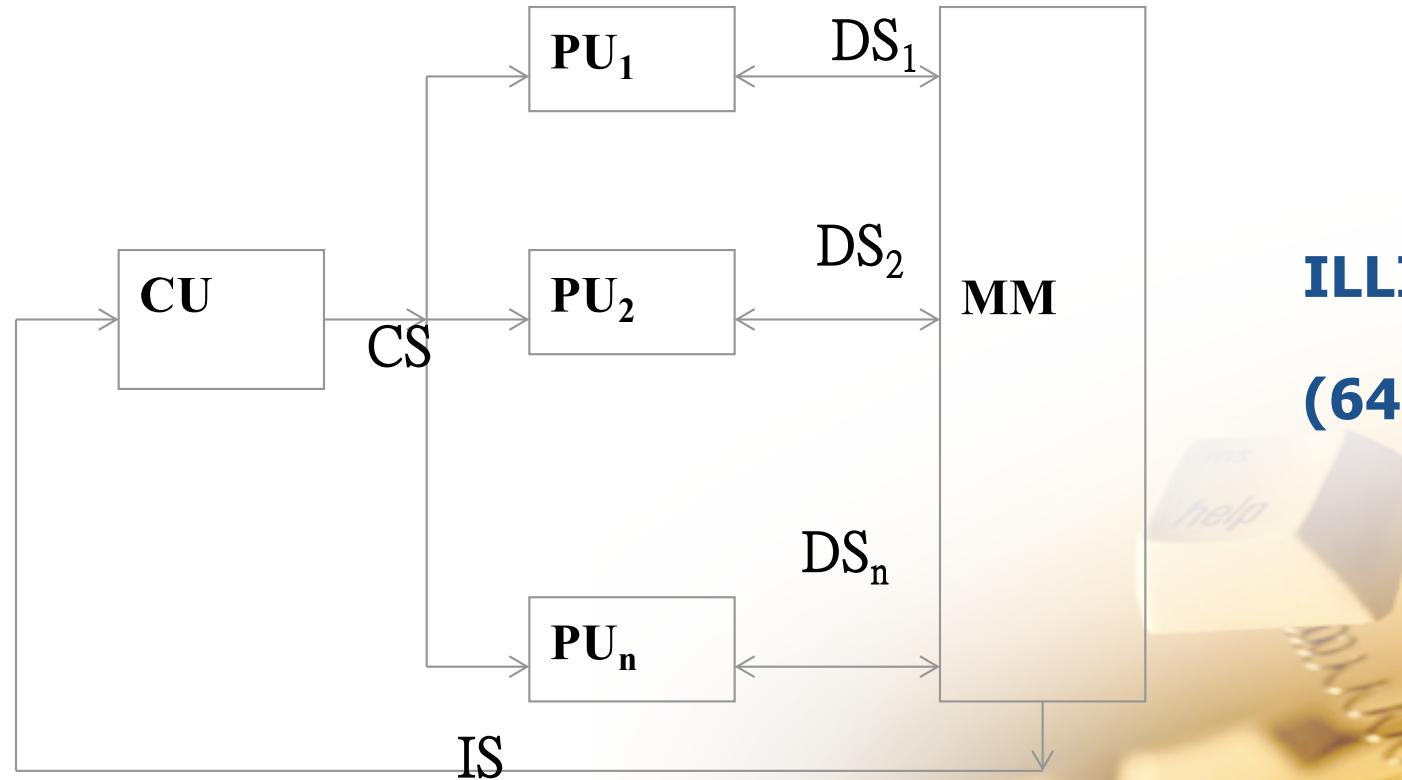
■ SISD-Single Instruction stream Single Data stream



Example: PDP-11、IBM-360/370、PC 8086、Z-80

Chapter1 Concepts of Computer Architecture

I SIMD-Single Instruction stream Multiple Data stream

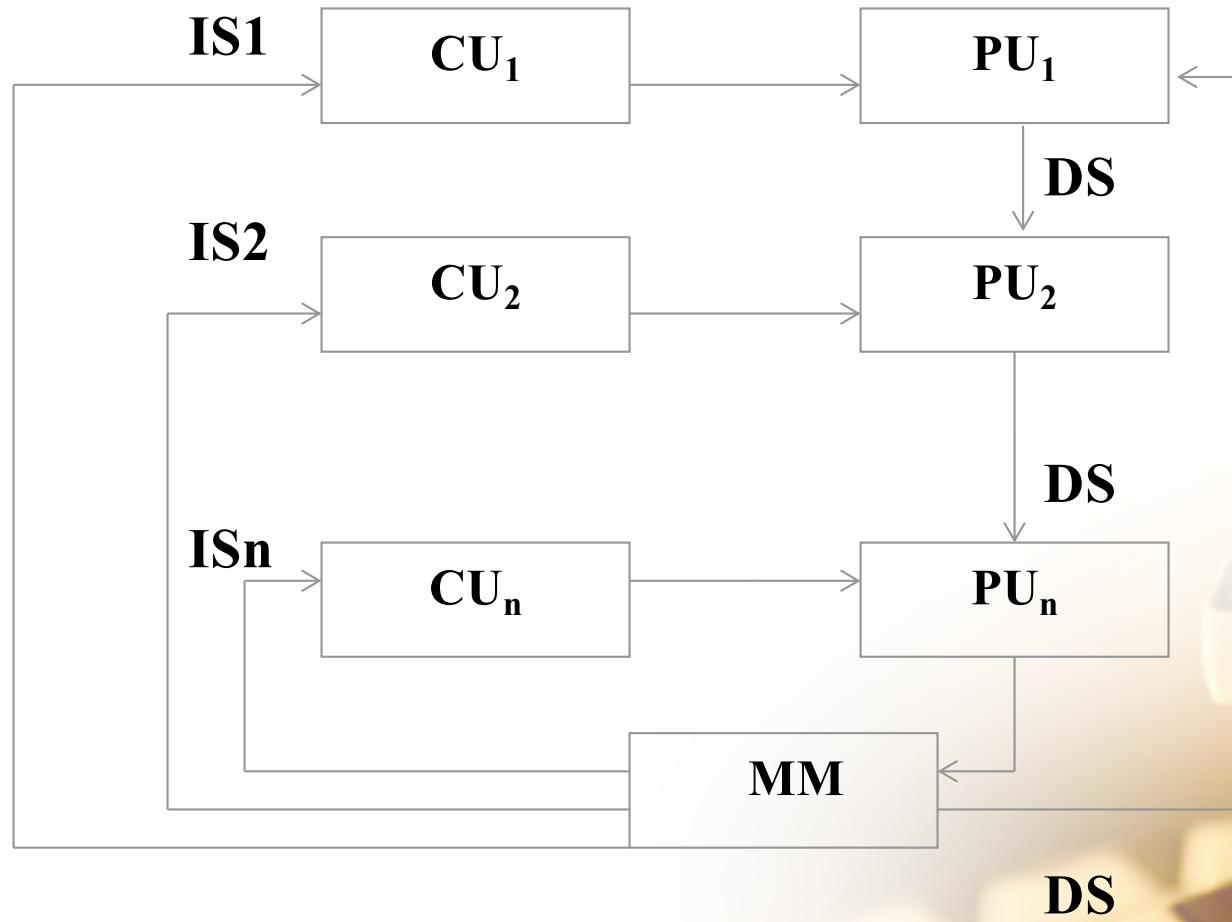


ILLIAC IV
(64 Units)

❖ **Array Processor, Parallel Processor**

Chapter1 Concepts of Computer Architecture

■ MISD-Multiple Instruction stream Single Data stream

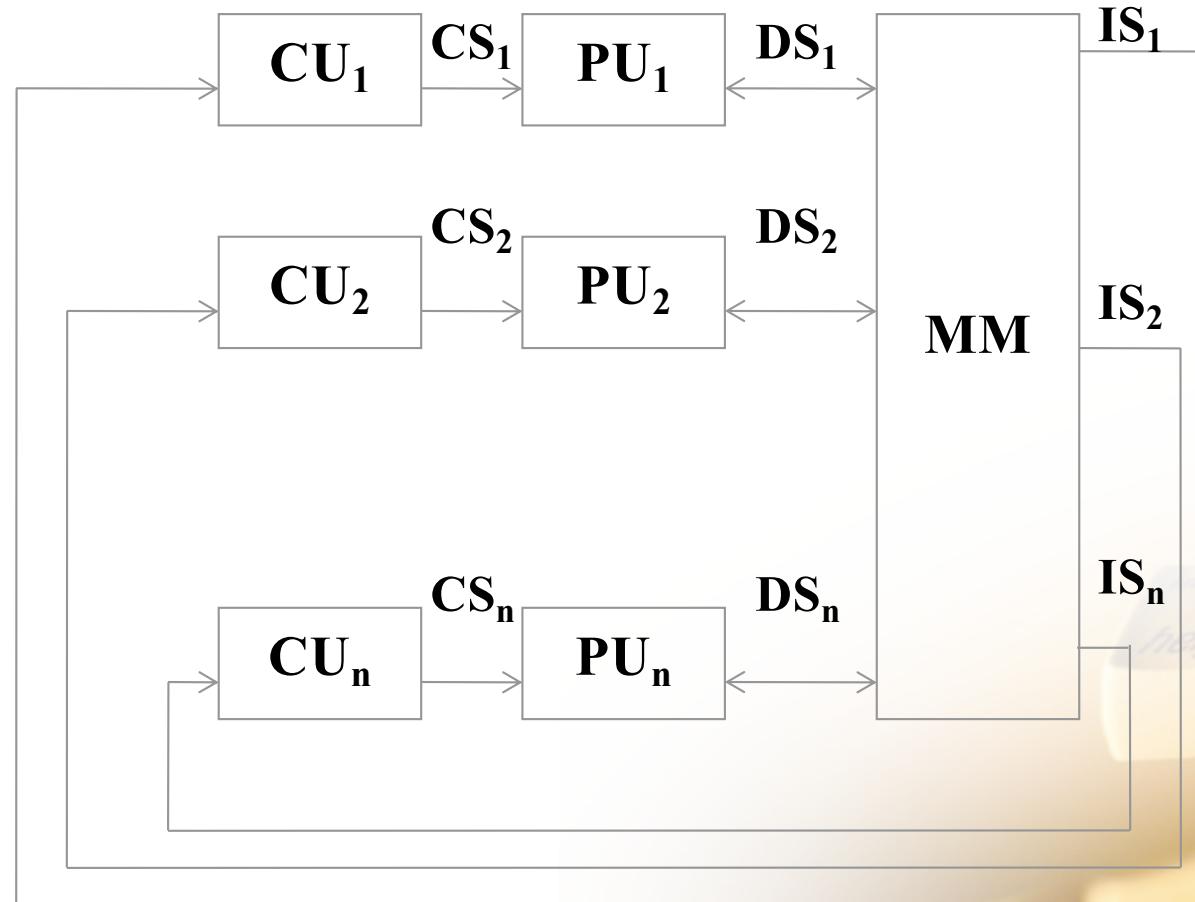


RISC, Vector
Machine

DS

Chapter1 Concepts of Computer Architecture

I MIMD-Multiple Instruction stream Multiple Data stream



Example: IBM 3081/3084, Univac 1100/80, Cray-2

❖ Multiprocessor

Chapter1 Concepts of Computer Architecture

2. Feng's Classification

Maximum Parallelism Degree

the largest binary digits which can be handled by Computer per unit time

n: the binary digits in a word which can be handled at the same time

m: the number of words which can be handled by a functional unit at the same time

- (1) WSBS, n=1,m=1
- (2) WSBP , n>1,m=1
- (3) WPBS, n=1,m>1
- (4) WPBP , n>1,m>1



Chapter1 Concepts of Computer Architecture

| The Design Principle of the Computer System

- 1. To Accelerate High Frequency Used Components**

- 2. Amdahl's Law**

- 3. Program Access Locality Principle**



Chapter1 Concepts of Computer Architecture

Amdahl's Law

The performance of a part of a system is improved by some ways, thus improves the performance of the whole system.

The measurement is speedup ratio. (Sp)

$$\text{Speedup}_{\text{overall}} = \frac{\text{ExTime}_{\text{old}}}{\text{ExTime}_{\text{new}}} = T_e / T_0 = \frac{1}{(1 - f_e) + f_e / r_e}$$

$$T_0 = T_e(1 - f_e + f_e / r_e)$$

Chapter1 Concepts of Computer Architecture

$$Sp = T_e / T_0 = \frac{1}{(1 - fe) + fe / r_e}$$

$f_e \rightarrow 0, Sp \rightarrow 1$

$r_e \rightarrow \infty, \text{ Then}$

$$S_p = \frac{1}{1 - f_e}$$



Chapter1 Concepts of Computer Architecture

Ex1:

A test program contains a large number of floating point data processing operations, in order to improve performance we can use two options:

One is to use hardware implementation for floating-point square root (FPSQR) operation, this method can make the operation speed increased by 10 times;

Another solution is to improve the speed of all floating-point data manipulation (FP), to make it speed up 2 times.

It is also known that FPSQR operation time accounts for 20% of the entire test program execution time, and FP operation accounts for 50% of the entire execution time.

Chapter1 Concepts of Computer Architecture

$$Sp(FPSQR) = \frac{1}{(1-0.2) + \frac{0.2}{10}} = 1.22$$

$$Sp(FP) = \frac{1}{(1-0.5) + \frac{0.5}{2}} = 1.33$$

Chapter1 Concepts of Computer Architecture

Ex2:

If we make a function's operation speed increased by 10 times, but this function's operation time accounts for 40% of the entire system execution time. After this enhancement method is adopted, how many can be improved the performance of the whole system?

Answer: $f_e=0.4$, $r_e=10$, then $Sp=1.56$

Chapter1 Concepts of Computer Architecture

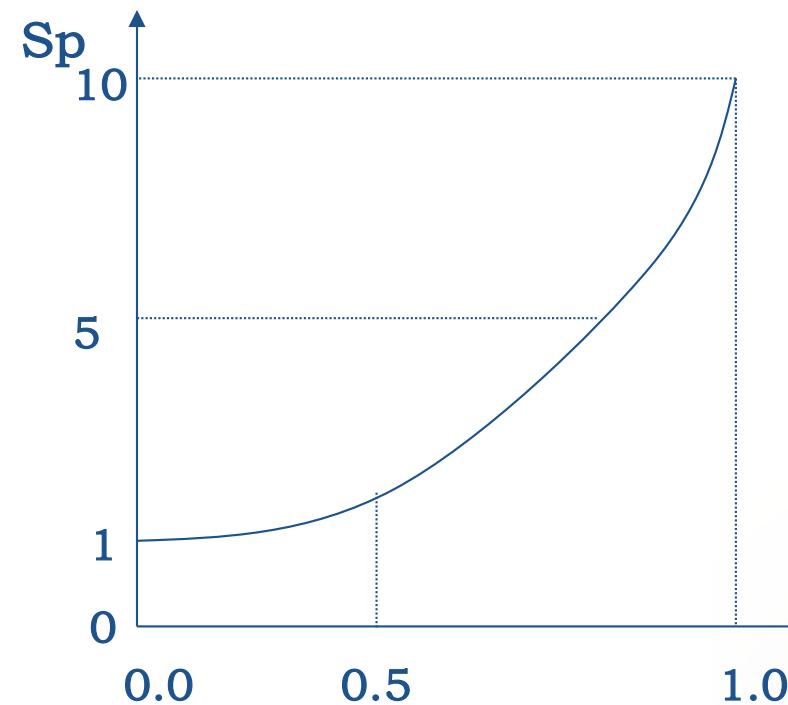


Figure1 The relation between Sp and f_e

Chapter1 Concepts of Computer Architecture

Program Access Locality Principle

Program tends to reuse data and instructions which are just used.

Temporal locality

Recently accessed code, probably will soon be accessed again

Spatial locality

The code with adjacent address may be continuous accessed

Chapter1 Concepts of Computer Architecture

I Computer Performance Assessment

Plane	DC to Paris	Speed	Passengers	Throughput (pmph)
Boeing 747	6.5 hours	610 mph	470	286,700
BAD/Sud Concorde	3 hours	1350 mph	132	178,200

- Time to do the task (Execution Time)
 - execution time, response time, latency
- Tasks per day, hour, week, sec, ns... (Performance)
 - throughput, bandwidth

Chapter1 Concepts of Computer Architecture

I Computer Performance Assessment

CPU time = CPU clock cycles for a program \times Clock cycle time

$$\text{CPU time} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

$$\text{CPI} = \frac{\text{CPU clock cycles for a program}}{\text{Instruction count}}$$

$$\frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}} = \frac{\text{Seconds}}{\text{Program}} = \text{CPU time}$$

$$\text{CPU time} = \text{Instruction count} \times \text{Cycles per instruction} \times \text{Clock cycle time}$$

Chapter1 Concepts of Computer Architecture

Let CPI_i = clocks per instruction for class i of instructions

Let IC_i = instruction count for class i of instructions

$$\text{CPU cycles} = \sum_{i=1}^n (CPI_i \times IC_i)$$

$$CPI = \frac{\sum_{i=1}^n (CPI_i \times IC_i)}{\sum_{i=1}^n IC_i}$$

$$Feq_i = \frac{IC_i}{\sum_{i=1}^n IC_i}$$

$$CPI = \sum_{i=1}^n (CPI_i \times Feq_i)$$

Chapter1 Concepts of Computer Architecture

I CPI Example

Base Machine

Op	Freq	CPI _i	CPI _i *F _i	(% Time)
ALU	50%	1	.5	(33%)
Load	20%	2	.4	(27%)
Store	10%	2	.2	(13%)
Branch	20%	2	.4	(27%)

1.5

Chapter1 Concepts of Computer Architecture

Ex1:

A kind of computer only uses the Load/Store instructions to read or write memory. According to the results of the program tracking experiment, the proportion of each instruction and CPU clock cycles are as follows:

Please try to get the value of average CPI.

Chapter1 Concepts of Computer Architecture

Instruction Type	Proportion	CPI
ALU	43%	1
LOAD	21%	2
STORE	12%	2
J	24%	2

$$\begin{aligned} \text{CPI} &= 1 \times 0.43 + 2 \times 0.21 + 2 \times 0.12 + 2 \times 0.24 \\ &= 0.43 + 0.42 + 0.24 + 0.48 = 1.57 \end{aligned}$$

Chapter1 Concepts of Computer Architecture

Ex2:

If FP operation ratio is 25%, $CPI_{FP} = 4$, $CPI_{other} = 1.33$. FPSQR operation ratio is 2%, $CPI_{SQR} = 20$.

There are two options for improvement.

- ❖ One is to improve the FP operation speed, make it double, namely $CPI_{FP} = 2$;
- ❖ The other is to improve the speed of FPSQR 10 times, namely $CPI_{SQR} = 2$;
- ❖ Try to compare the two solutions according to the CPI result.

$$\begin{aligned} CPI &= \sum_{i=1}^n (CPI_i \times \frac{I_i}{I_N}) \\ &= (4 \times 0.25) + (1.33 \times 0.75) = 2 \end{aligned}$$

Chapter1 Concepts of Computer Architecture

Option1: $CPI_{FP1} = 2$

$$CPI_1 = (2 \times 0.25) + (1.33 \times 0.75) = 1.5$$

$$= CPI - 0.25 \times (CPI_{FP} - CPI_{FP1})$$

$$= 2.0 - 0.25 \times (4 - 2) = 1.5$$



Chapter1 Concepts of Computer Architecture

Option2: $CPI_{SQR2}=2$

$$CPI_2 = CPI - 0.02 \times (CPI_{SQR} - CPI_{SQR2})$$

$$= 2 - 0.02 \times (20 - 2) = 1.64$$

$CPI_1 < CPI_2$, Option1 is better!



Chapter1 Concepts of Computer Architecture

■ MIPS and MFLOPS

$$\text{MIPS} = \frac{I_N}{T_e \times 10^6} = \frac{R_c}{CPI \times 10^6} \text{ (MIPS)}$$

$$\text{MFLOPS} = \frac{I_{FN}}{T_e \times 10^6}$$



Chapter1 Concepts of Computer Architecture

■ Performance Evaluation

Benchmark Test

- ❖ **Actual Program: CL**
- ❖ **Kernels: Livermore Loops, LINPACK**
- ❖ **Synthetic benchmarks: Whetstone、Dhrystone**



Chapter1 Concepts of Computer Architecture

Processing Performance Evaluation Result

	A(s)	B(s)	C(s)
P1	1	10	20
P2	1000	100	20
Total Time	1001	110	40
Am	500.5	55	20

Chapter1 Concepts of Computer Architecture

✓ Arithmetic Performance Average

$$A_m = \frac{1}{n} \sum_{i=1}^n T_i = \frac{1}{n} \sum_{i=1}^n \frac{1}{R_i}$$

$$= \frac{1}{n} \left(\frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_n} \right)$$

Chapter1 Concepts of Computer Architecture

✓ Geometric Performance Average

$$G_m = \sqrt[n]{(\prod_{i=1}^n T_i)} = \sqrt[n]{\prod_{i=1}^n \frac{1}{R_i}}$$

✓ Harmonic Performance Average

$$H_m = \frac{n}{\sum_{i=1}^n \frac{1}{R_i}} = \frac{n}{T_1 + T_2 + \dots + T_n}$$

