National Institute of Technology Karnataka Surathkal
Department of Information Technology

IT 301 Parallel Computing

Introduction

Dr. Geetha V

Assistant Professor

Dept of Information Technology

NITK Surathkal

Index

- 1: Moore's Law and Need for Parallel Processing
- 2. Parallel Processing in Uniprocessor System
  - Bit Level , Instruction level, Task level and Data level parallelism
- 3: Multiprocessor and Multicores System

1: Introduction

Course Plan: Theory:

Part A: Parallel Computer Architectures

Week 1,2,3: ***Introduction to Parallel Computer Architecture:***

Parallel Computing,

Parallel architecture,

bit level, instruction level , data level and task level parallelism.

Instruction level parallelisms: pipelining(Data and control instructions),

scalar and superscalar processors,

vector processors.

Parallel computers and computation.

1: Moore's Law and Need for Parallel Processing

- Chip performance doubles every 18-24 months
- Power consumption is proportional to frequency of the system
- Limitations of serial Computing
  - Heating issues
  - Limit to transmissions seeds
  - Leakage currents
  - Limit to miniaturization

- Smaller transistor = faster processors
- Faster processor = increased power consumption
- Increased power consumption = increased heat
- Increased heat = unreliable processors

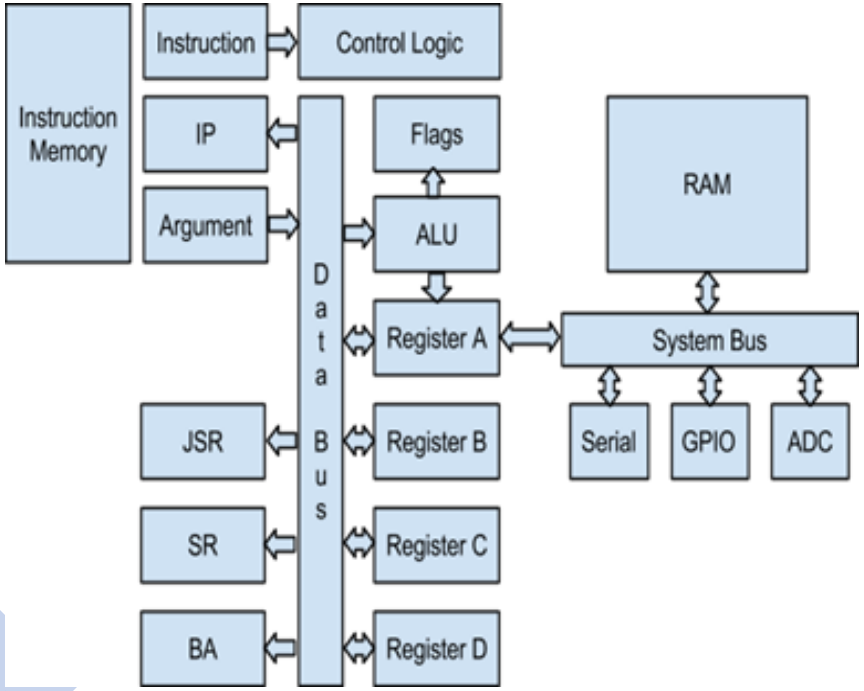
- Parallel processing in Uniprocessor systems
  - Bit level parallelism, Instruction level parallelism (pipelining, Superscalar processors), Data parallelism (vector processors) and task level parallelism (threads)
- Parallel processing in Multiprocessor and Multicore systems
  - Multi core processor, clusters, grids

2. Parallel Processing in Uniprocessor System

- Bit level parallelism
- Instruction level parallelism (pipelining, Superscalar processors)
- Data parallelism (vector processors)
- Task level parallelism (threads)

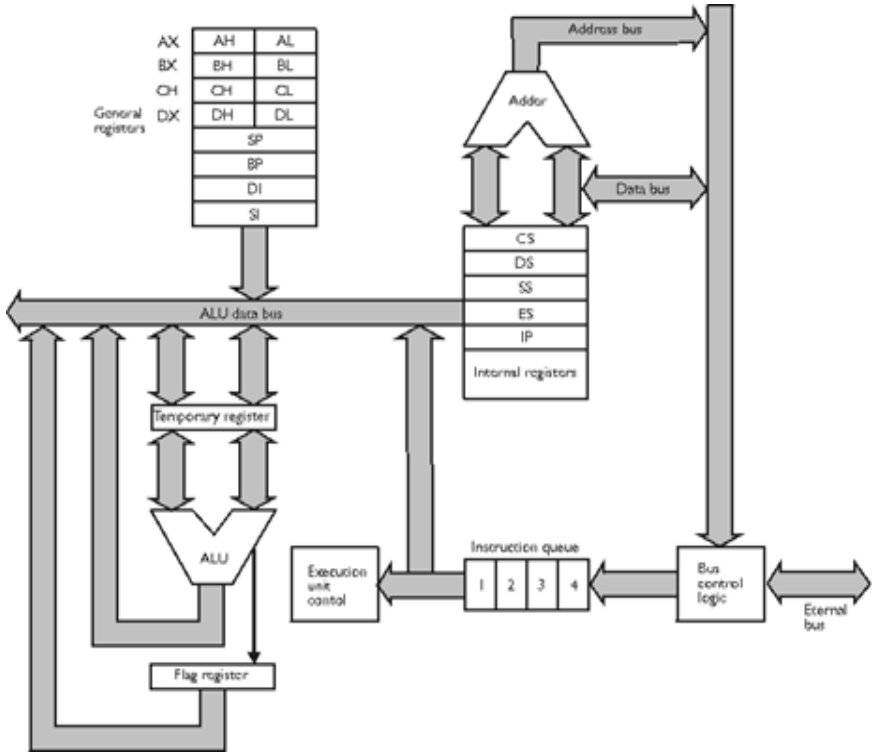
2. Parallel Processing in Uniprocessor System

- Bit level parallelism
- Adding 16-bit number in 8-bit processor



http://ryanohs.com/2015/09/high-level-architecture/

- Adding 16-bit number in 16-bit processor



http://www.c-jump.com/CIS77/CPU/VonNeumann/lecture.html#V77_0010_von_neumann

2. Parallel Processing in Uniprocessor System

- Bit Level Parallelism (eg. 1516H +2829H)
  - Adding 16-bit number in 8-bit processor

```
MVI A, #16H
MVI B, #29H
ADD B
MVI A, #15H
MVI B, #28H
ADC B
MOV D, A
```

  - Adding 16-bit number in 16-bit processor

```
MVI AX, #1516H
MVI BX, #2829H
ADD AX, BX
MOV CX, AX
```

2. Parallel Processing in Uniprocessor System

- Bit Level Parallelism
  - 8-bit processor (1972-1974)

Intel 8008, Intel 8080

  - 16-bit processor (1979 – 1982)

Intel 8086, Intel 286

  - 32-bit processor (1985 -2006)

Intel 386, Intel 486, P5 (Pentium)

Pentium Pro, Pentium II, Pentium III, Pentium M, Intel Core, Dual-core Xeon LV

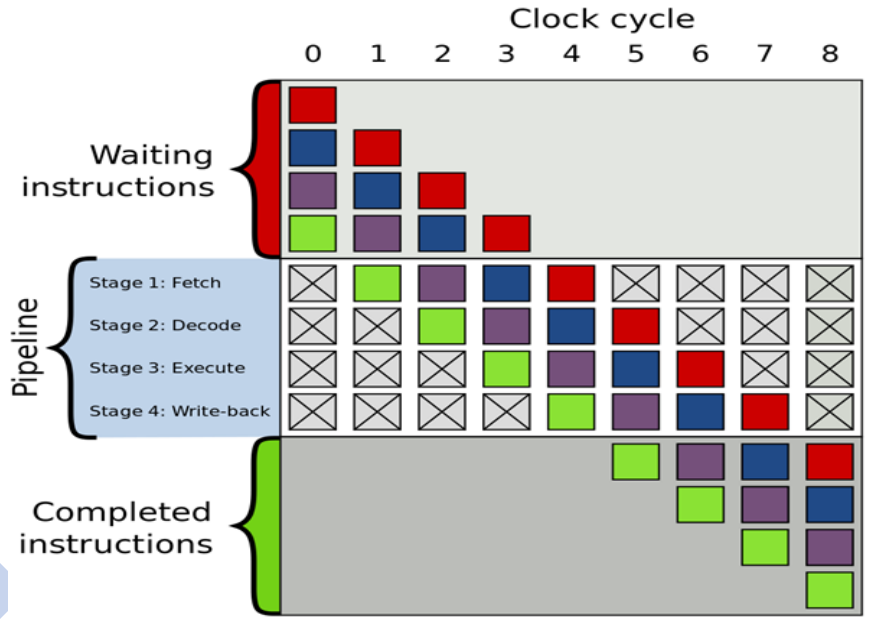
  - 64-bit processor (2004 onwards)

Itanium , Itanium 2 (IA-64), Pentium 4F, Pentium D, Intel 6: Xeon (NetBurst), Intel Core 2, Intel Pentium Dual Core, Celeron, Celeron M; Nehalem: Intel Pentium, Core i3, i5, i7, Xeon…..etc

2. Parallel Processing in Uniprocessor System

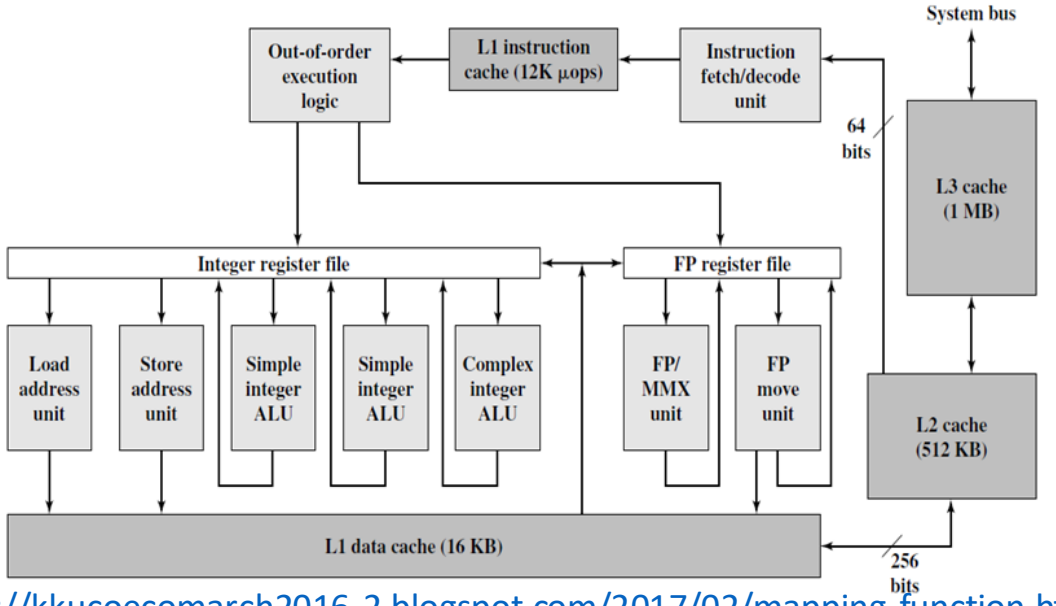
- Instruction level parallelism (pipelining, Superscalar processors)

- Scalar Processors
  - Goal towards one instruction execution per cycle



https://en.wikipedia.org/wiki/Instruction_pipelining

- Superscalar Processors
  - Multiple Instruction per cycle



http://kkucoecomarch2016-2.blogspot.com/2017/02/mapping-function.html

2. Parallel Processing in Uniprocessor System

- Instruction level parallelism (pipelining, Superscalar processors)
  - Scalar Processors
    - •Goal towards one instruction execution per cycle
    - •Reduced Instruction Set Computer (RISC) Scalar processors
      - Intel i860, Motorola MC8810, SUN's SPARC CY7C601 etc
    - •Complex Instruction Set Computer (CISC) scalar Processors
      - Intel 386, 486; Motorola's 68030, 68040; etc
  - Superscalar Processors
    - •Multiple Instruction per cycle
      - Pentium , Pentium Pro, Pentium II, Pentium III Motorola 88110 etc.

2. Parallel Processing in Uniprocessor System

- Task level parallelism

Task parallelism or function level parallelism is a form of parallelization of computer code across multiple processors in parallel computing environment

- Models
  - •Task dependency graph model
  - •Master Slave Model
  - •Pipeline/Producer Consumer model

- Data level parallelism

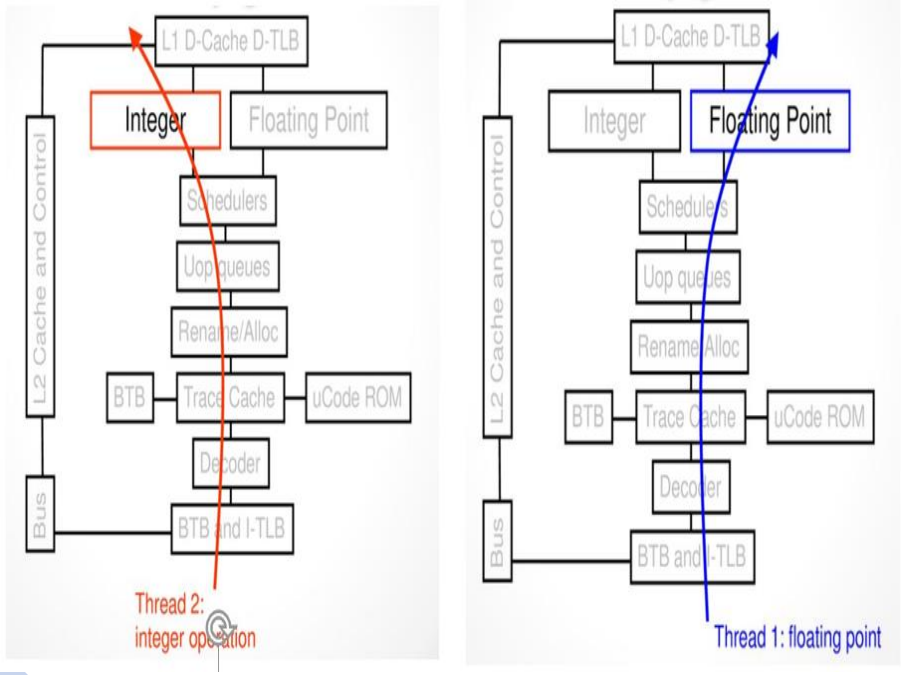
Data parallelism is a form of parallelization of computing across multiple processors in parallel computing environment.

- Architectures
  - •Vector Processors
  - •Single Instruction Multiple Data (SIMD)

2. Parallel Processing in Uniprocessor System

- Task level parallelism

Intel Xeon hyperthreading

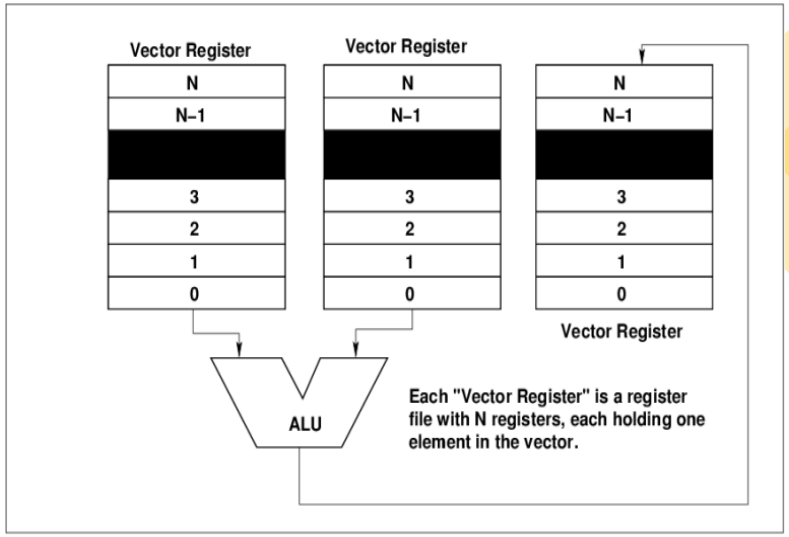


https://slideplayer.com/slide/14851286/

- Data level parallelism

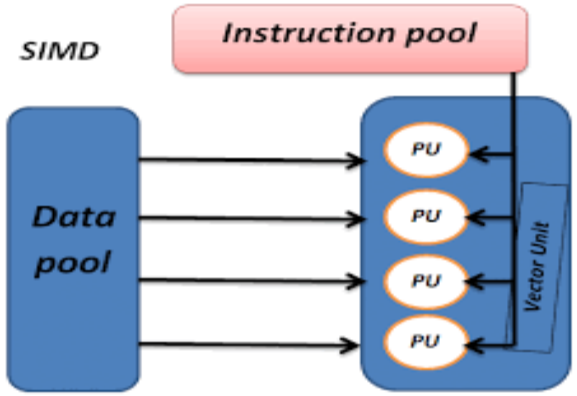
- Vector Processor

Eg. Cray 1



- SIMD

Eg. ILLIAC IV

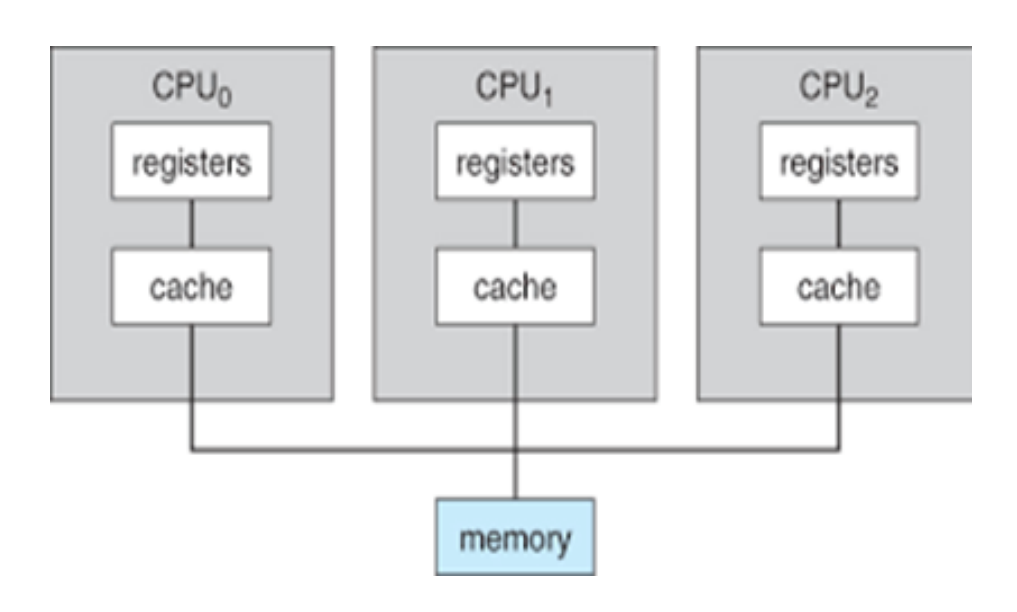


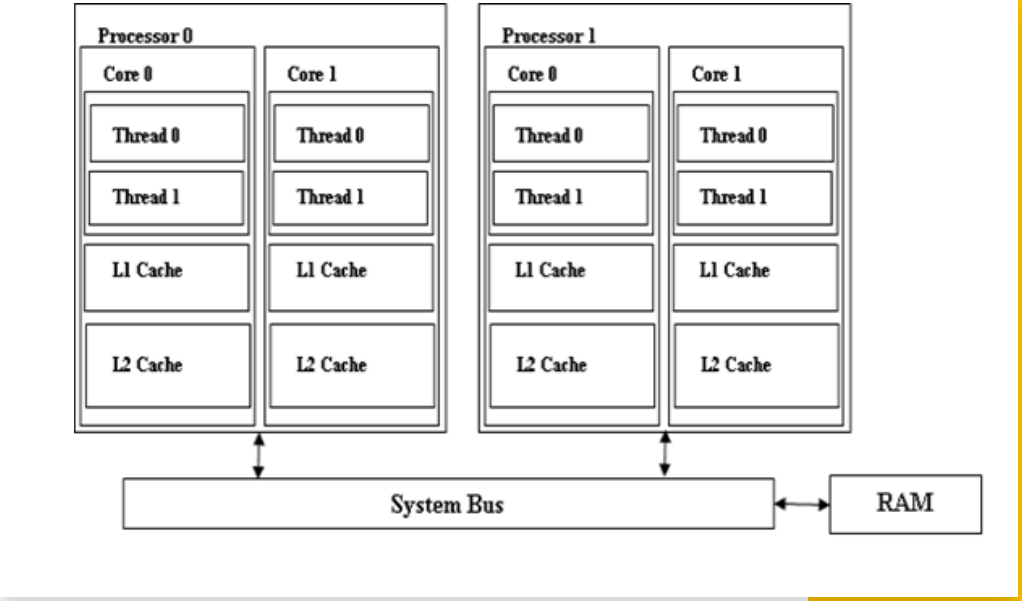
https://www.researchgate.net/profile/Ahmed_Al-Araji/publication/342107319/figure/fig1/AS:901279303864320@1591893139023/Figure-1-Big-picture-of-SIMD-architecture.ppm
https://www.ausairpower.net/XIMG/vector-1.png

3: Multiprocessor and Muticore system

- Multiprocessor system: Two or more CPU within single computer system

- Multicore processor: Multiple Execution units (cores) o the same chip.

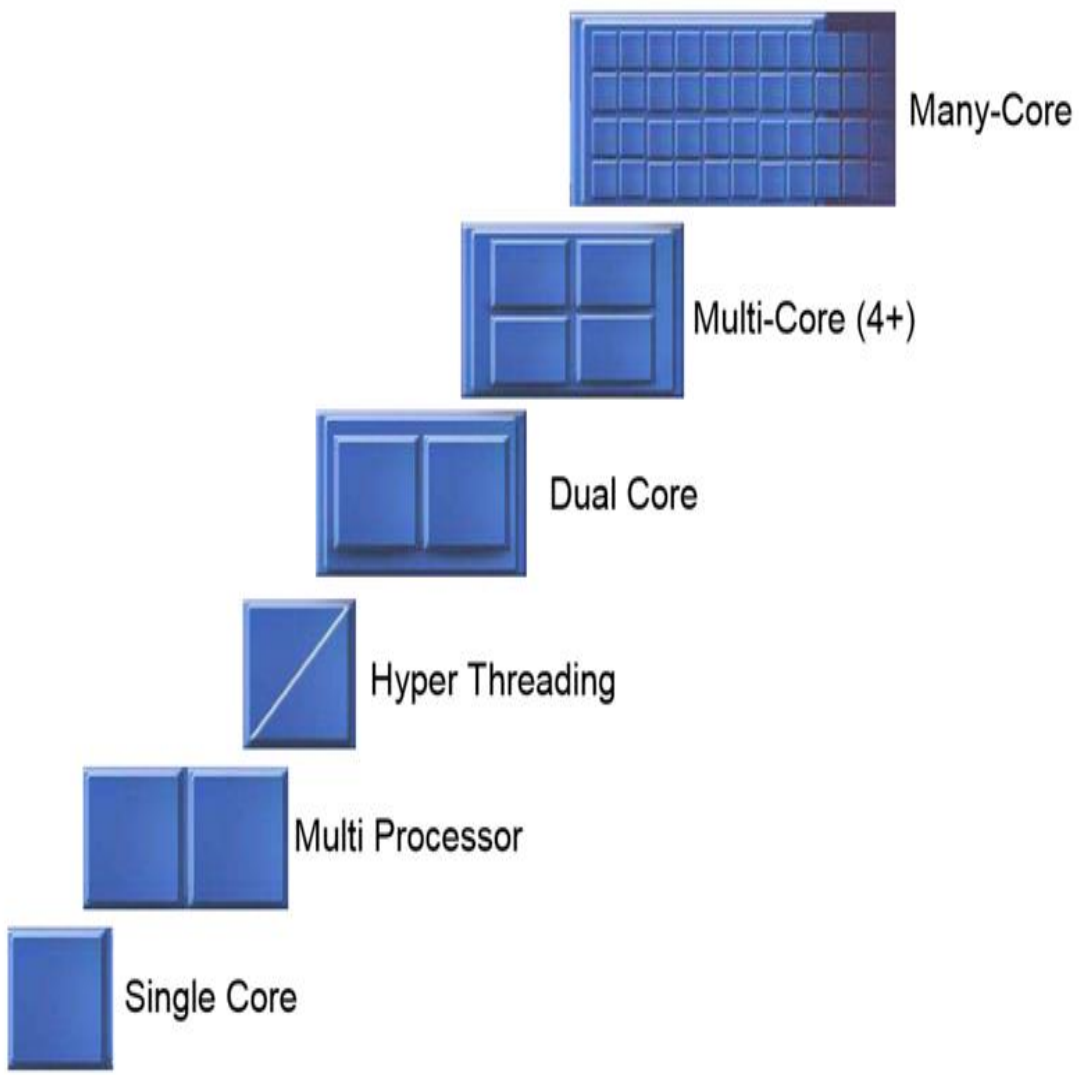




3: Multiprocessor and Muticore system

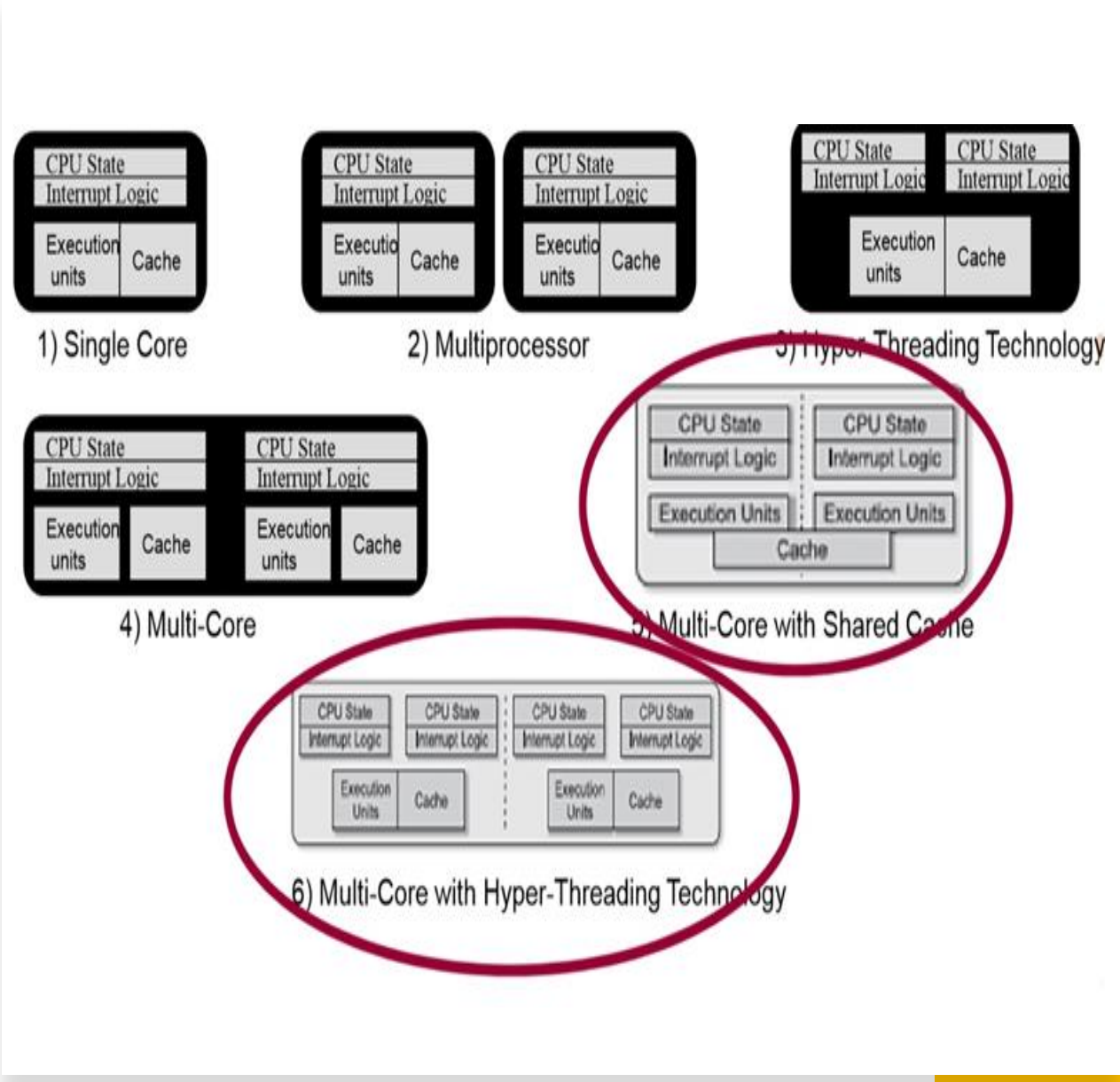
- Multiprocessor system: Two or more CPU within single computer system

- Multicore processor: Multiple Execution units (cores) o the same chip.



3: Multiprocessor and Muticore system

- Single Core: One execution unit in one chip
- Multiprocessor : Two or more CPU within single computer system
- Hyperthreading : Task level parallelism
- Multicore processor: Multiple Execution units (cores) on the same chip.
- Multicore with shared cache: All core shares same cache
- Multicore with Hyperthreading: task level parallelism



Reference

Textbooks and/or Reference Books:

1. Professional CUDA C Programming – John Cheng, Max Grossman, Ty McKercher, 2014
2. Wilkinson, M. Allen, ”Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers”, Pearson Education, 1999
3. I. Foster, ”Designing and building parallel programs”, 2003
4. Parallel Programming in C using OpenMP and MPI – Micheal J Quinn, 2004
5. Introduction to Parallel Programming – Peter S Pacheco, Morgan Kaufmann Publishers, 2011
6. Advanced Computer Architectures: A design approach, Dezso Sima, Terence Fountain, Peter Kacsuk, 2002
7. Parallel Computer Architecture : A hardware/Software Approach, David E Culler, Jaswinder Pal Singh Anoop Gupta, 2011 8. Introduction to Parallel Computing, Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Pearson, 2011

Thank You