CS531: Memory Systems and Architecture
Jan-Apr 2022

Dr. Shirshendu Das
Assistant Professor,
Department of CSE
IIT Ropar.

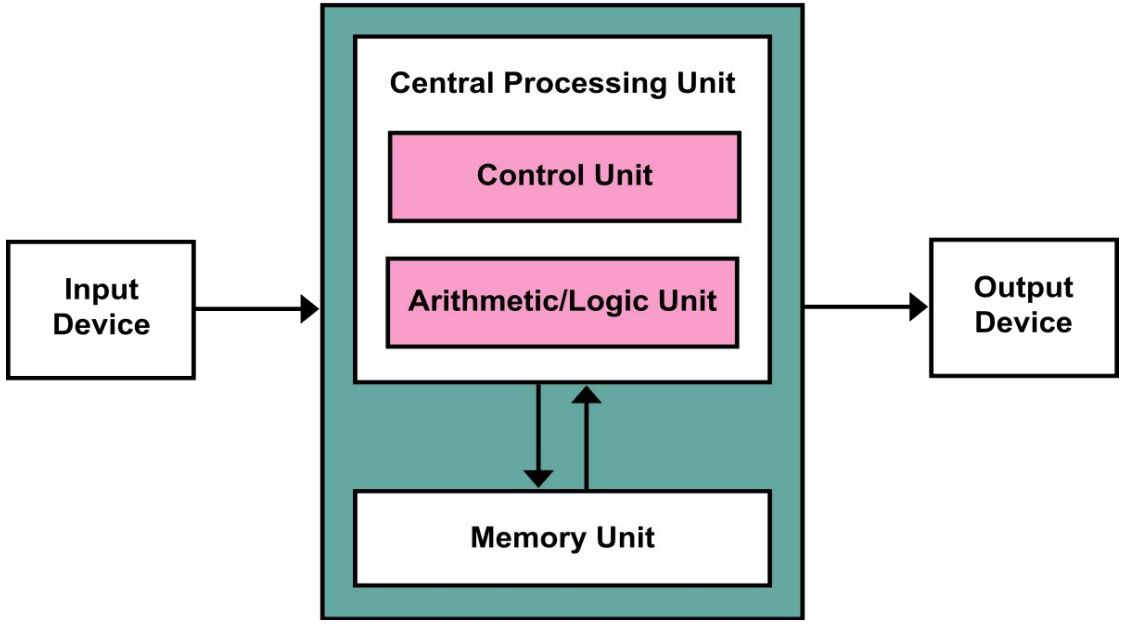
Topic: Introduction to Memory

History of Computers:

Universal Turing Machine
(1936)

Alan Turing
(1912-1954)

History of Computers:



Von Neumann Architecture

Basic Layout of Computer

Computer

1. CPU
2. Memory
3. I/O
4. Interconnects

a) Register
b) ALU
c) Control Unit
d) Internal CPU interconnect

a) Cache Memory
b) Main Memory
c) Secondary memory

Chip Designing:

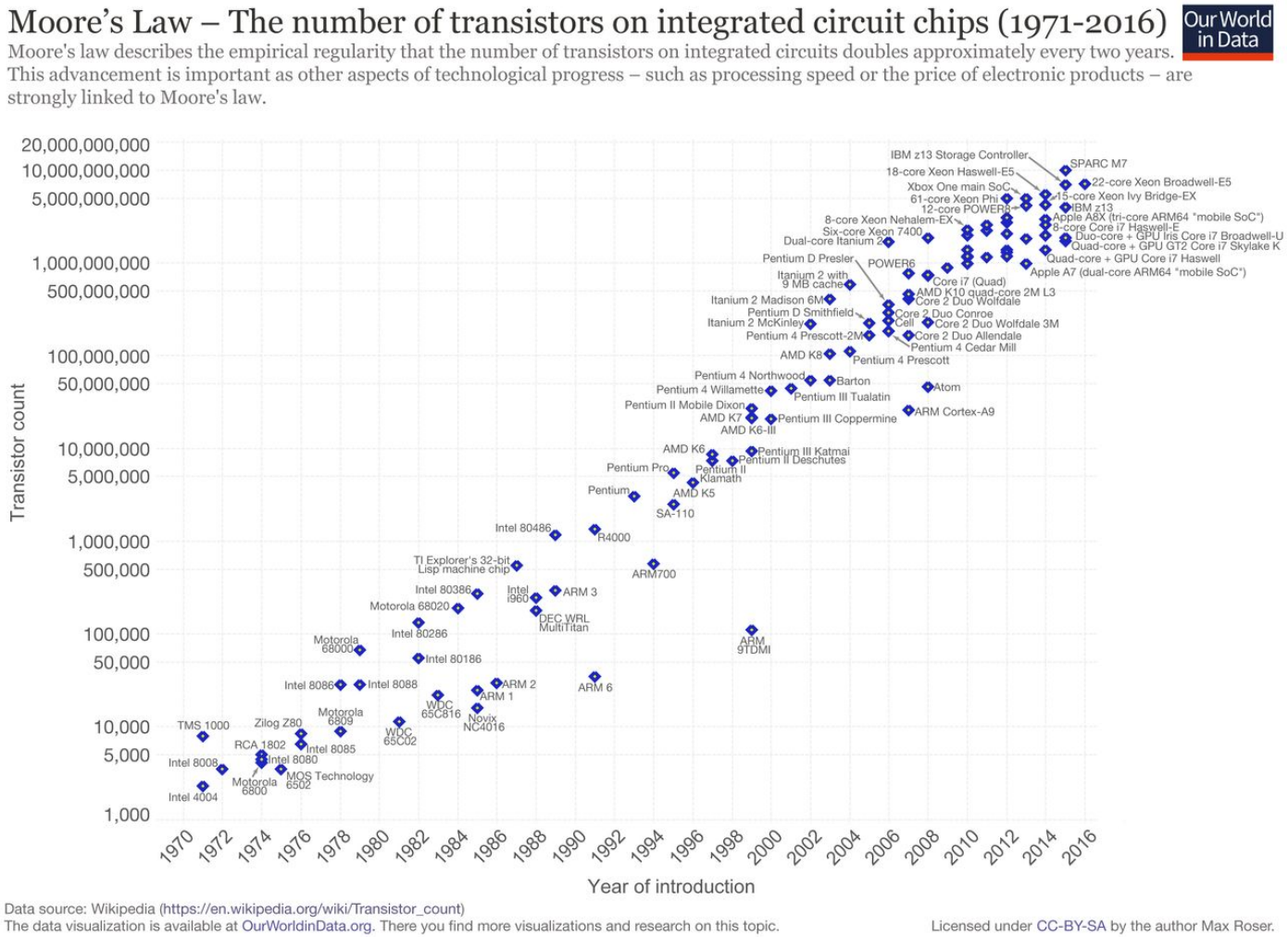
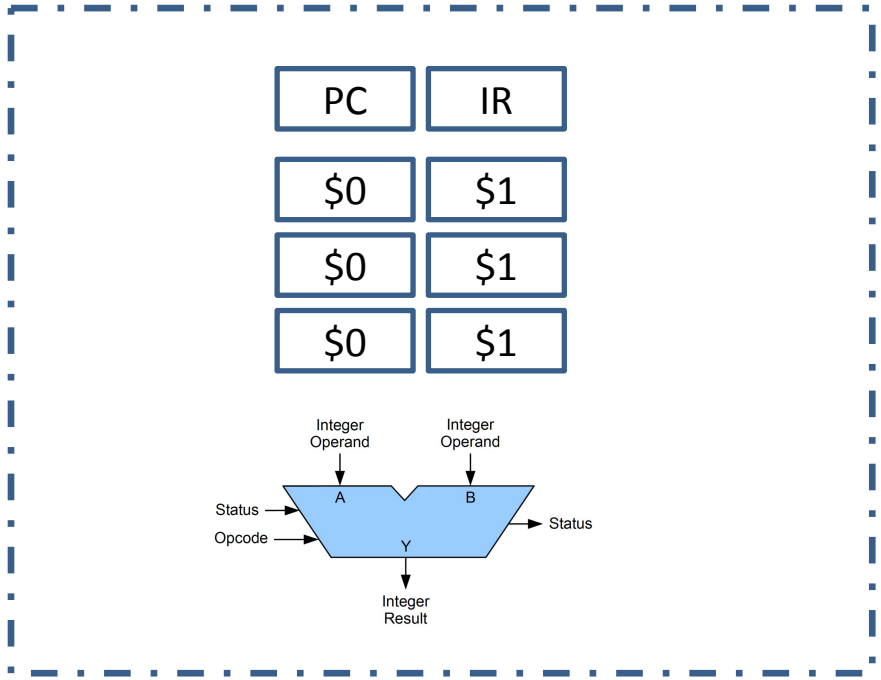
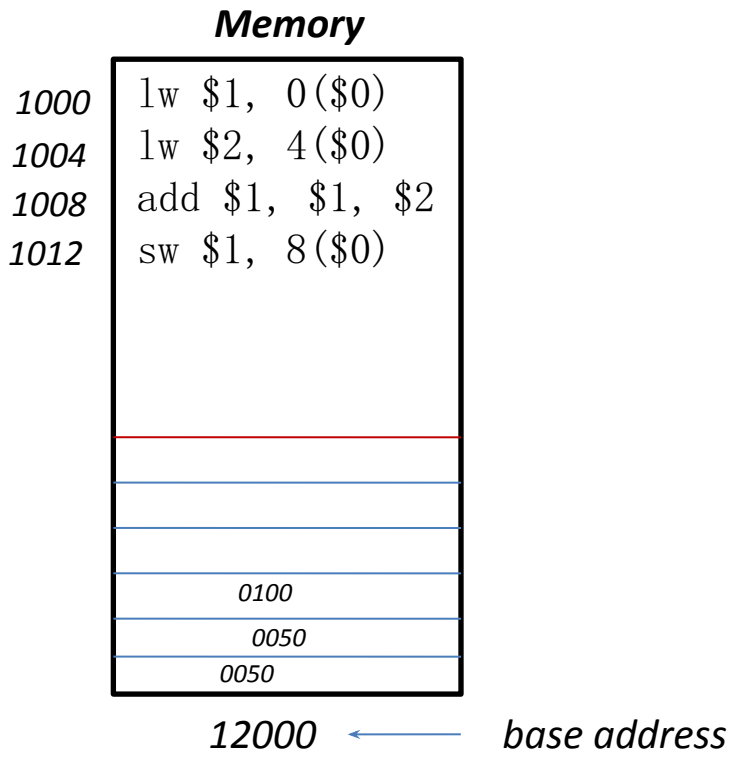


Image source: Wikipedia.

How Computer Works



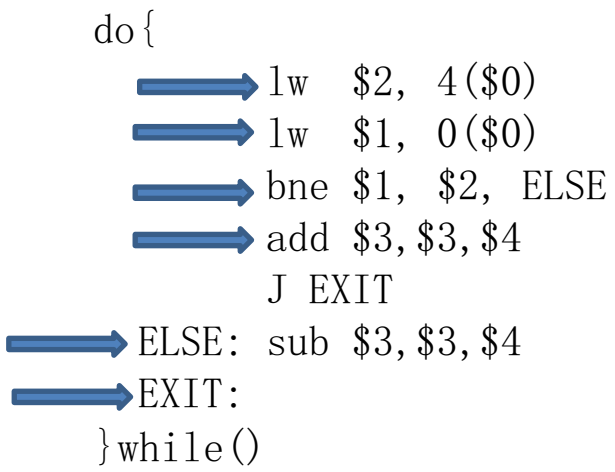


Fetch (IF) | Decode (ID) | Execute (EX) | Memory access (MEM) | Write-back (WB)

Program Counter (PC)

Increment:

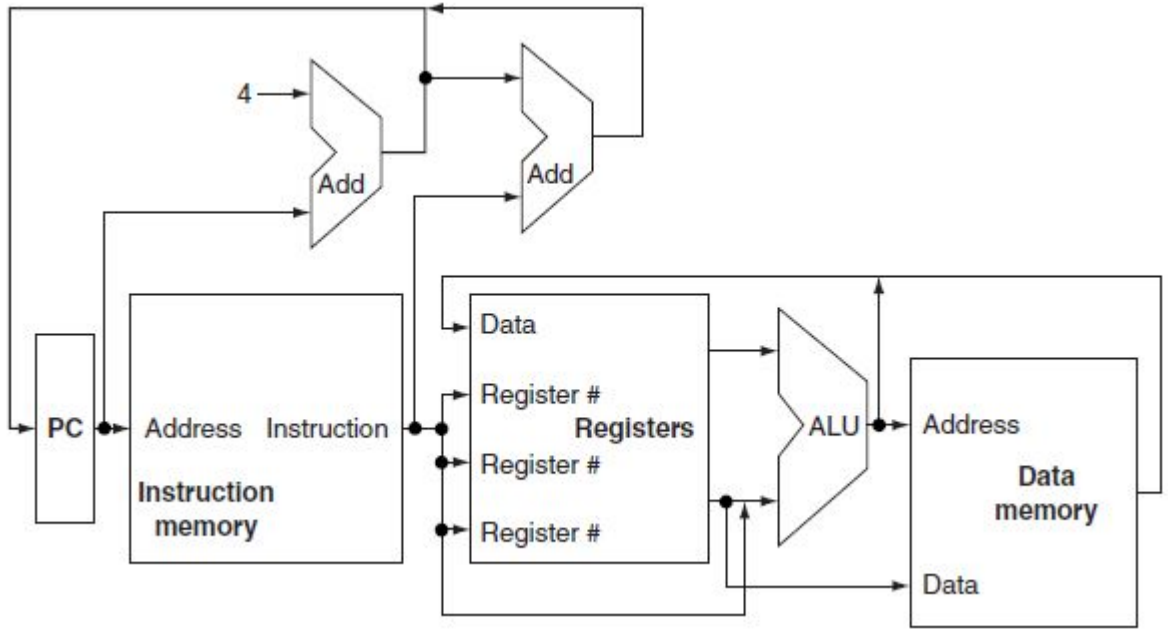
- PC will not increment linearly every time.
- Predicting the next instruction after executing a branch instruction is a major area of research.
- In case of loop, the next instruction can be predicted easily.



PC

Note: *do-while* statement is not a part of MIPS. It is added here to easily show an example of loop. However loops can be easily implemented in MIPS.

How Computer Works



MIPS implementation in Computer. Source: Patterson and Hennessy, "***Computer Organization and Design"***, Third edition, 2005.

Memory

```
add $1, $1, $2
adi $1, $1, #12
lw $1,  0($0)
add $1, $1, $2
sw $1,  8($0)
```

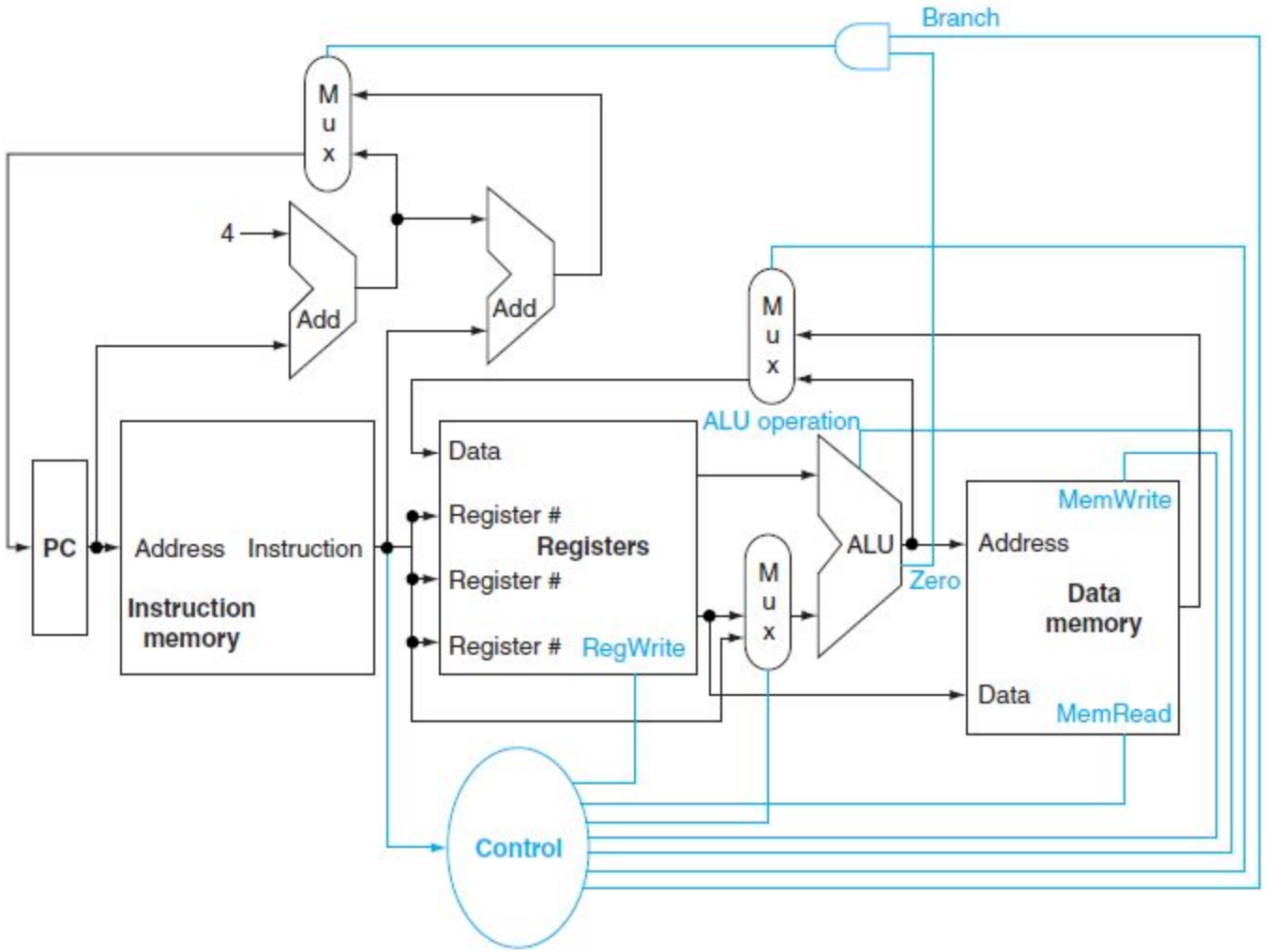
0105
0050
0055

12003

base address

For more detail read **Chapter 5** of Patterson and Hennessy, "***Computer Organization and Design"***, Third edition, 2005.

How Computer Works



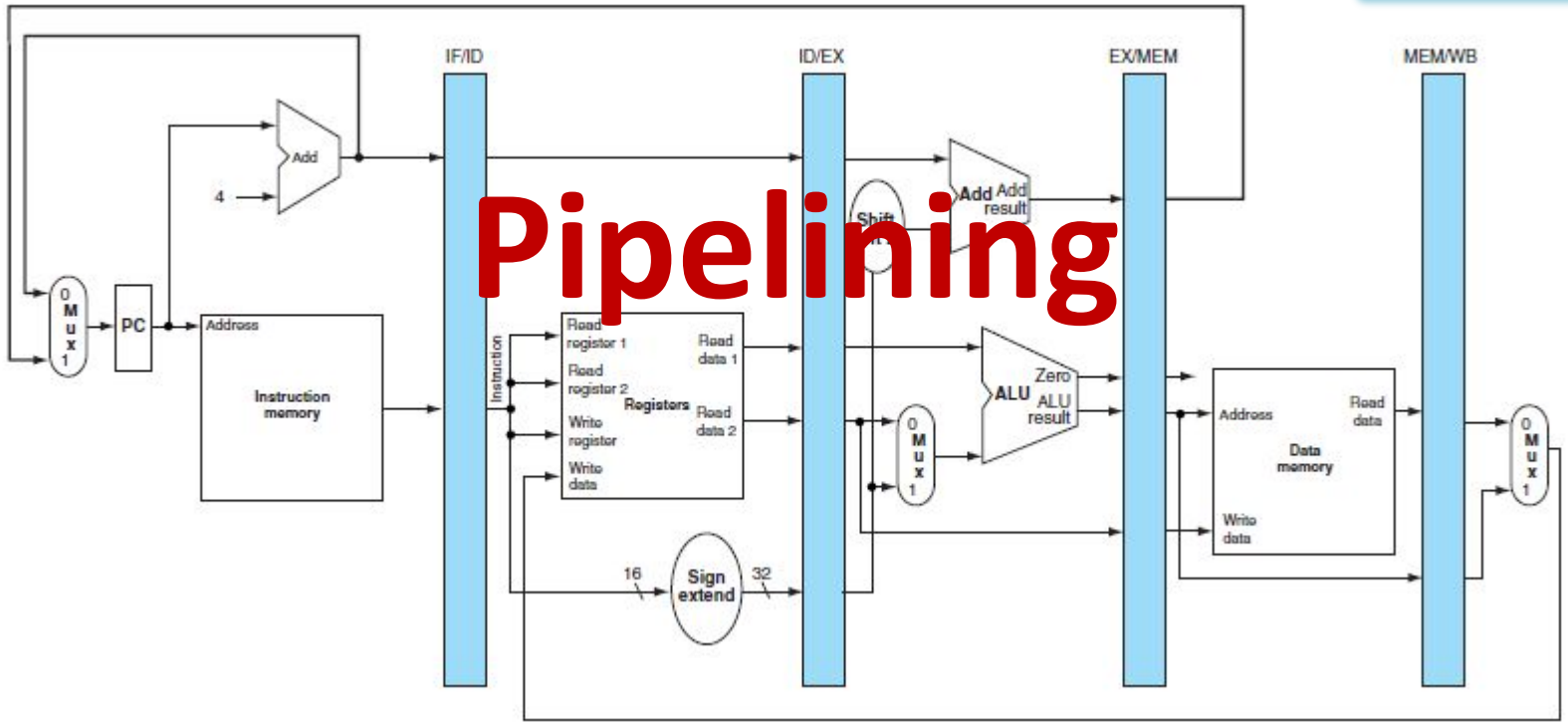
MIPS implementation in Computer. Source: Patterson and Hennessy, "***Computer Organization and Design"***, Third edition, 2005.

For more detail read **Chapter 5** of Patterson and Hennessy, "***Computer Organization and Design"***, Third edition, 2005.

Multi-cycle Instruction Execution:

IF: Instruction fetch	ID: Instruction decode/ register file read	EX: Execute/ address calculation	MEM: Memory access	WB: Write back

```
lw $1, 0($0)
lw $2, 4($0)
add $1, $1, $2
sw $1, 8($0)
```



Pipelining

MIPS implementation in Computer. Source: Patterson and Hennessy, "***Computer Organization and Design***", Third edition, 2005.

For more detail read **Chapter 5** of Patterson and Hennessy, "***Computer Organization and Design***", Third edition, 2005.

Performance of Computer

$$\text{CPU time} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}$$

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

Instruction Count means number of instructions executed by the computer.
Cycles Per Instruction (CPI) means average cycles required to execute an instruction.

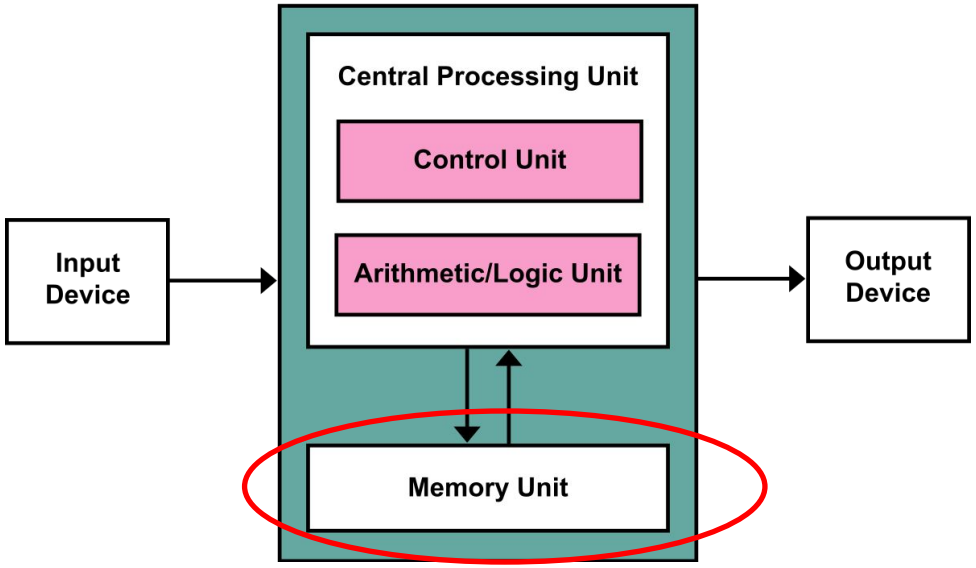
$$\text{CPI} = \frac{\text{CPU clock cycles for a program}}{}$$

$$\text{CPU time} = \text{Instruction count} \times$$

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

For more detail read **Chapter 1 (1.9)** of Hennessy and Patterson, "***Computer Architecture A Quantitative Approach"***, Fourth edition, 2007.

Memory Hierarchy



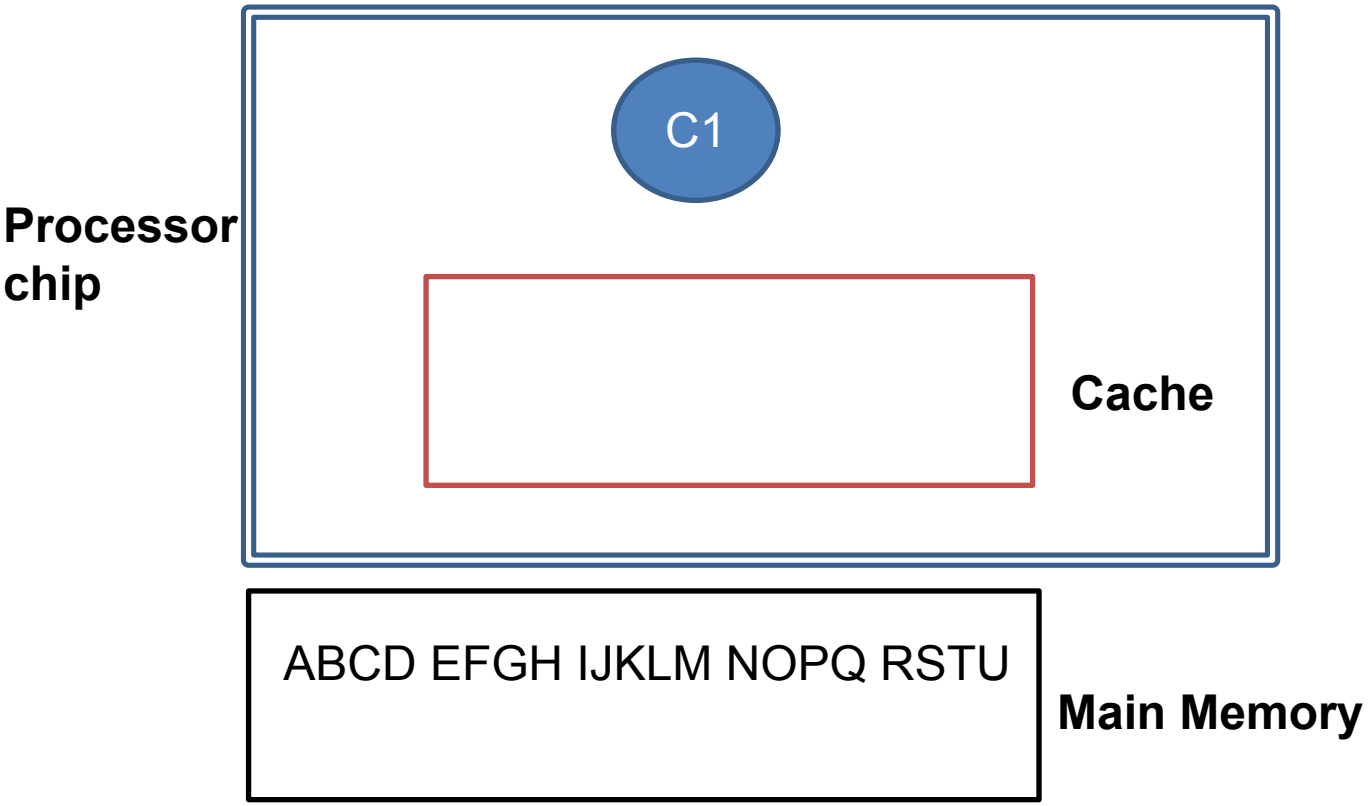
- ❑ Cache Memory: Placed on-chip.
  - ✔ L1-I, L1-D
  - ✔ L2
  - ✔ Last Level Cache (LLC). *** The levels of cache varies in different processors.*
- ❑ Main Memory
- ❑ Secondary memory.

How cache memory work.
Structure of cache memory.

Principle of locality

All the basic concepts starting from this slide must be cleared to continue with this course.

Memory Hierarchy



C1: A, D, A, F, L, G, F……

- ❑ Cache Memory: Placed on-chip.
  - ✔ L1-I, L1-D
  - ✔ L2
  - ✔ Last Level Cache (LLC). *** The levels of cache varies in different processors.*
- ❑ Main Memory

Principle of locality:

- **Temporal Locality**
- **Spatial Locality**

Introduction to Computer: Cache Memory

Cache memory is an on-chip fast memory placed very near to the core. It improves the performance of the computer.

- ✔ Design technology: SRAM Technology
- ✔ Block size: The basic unit of cache storage. Core requests for word. Multiple words stored in a block.
- ✔ Some important terms:
  - *Cache hit, Cache miss.*
  - *Miss rate: Miss per memory access.*
  - *Miss penalty:* Cost to fetch a block from main memory when the block is not found in cache.

- ✔ **Data management policy of cache:** block searching, block placement, block replacement.

Direct mapped cache, **fully associative cache**, **set-associative cache**.

Memory Hierarchy: *Some important terms*

$$\text{CPU execution time} = (\text{CPU clock cycles} + \text{Memory stall cycles}) \times \text{Clock cycle time}$$

$$\text{Memory stall cycles} = \text{Number of misses} \times \text{Miss penalty}$$

$$= \text{IC} \times \qquad \times \text{Miss penalty}$$

$$= \text{IC} \times \frac{\text{Memory accesses}}{\text{Instruction}} \times \qquad \times \text{Miss penalty}$$

$$\begin{aligned}\text{Memory stall clock cycles} = {} & \text{IC} \times \text{Reads per instruction} \times \text{Read miss rate} \times \text{Read miss penalty} \\ & + \text{IC} \times \text{Writes per instruction} \times \text{Write miss rate} \times \text{Write miss penalty}\end{aligned}$$

$$\frac{\text{Misses}}{\text{Instruction}} = \frac{\text{Miss rate} \times \text{Memory accesses}}{\text{Instruction count}} = \text{Miss rate} \times \frac{\text{Memory accesses}}{\text{Instruction}}$$

$$\text{Average memory access time} = \text{Hit time} + \qquad \times \text{Miss penalty}$$

AMAT

For more detail read **Appendix C** of Hennessy and Patterson, "***Computer Architecture A Quantitative Approach***", Fourth edition, 2007.

Introduction to Computer: Cache Memory

Direct mapped cache, **fully associative cache**, **set-associative cache**

Data management policy of cache: block searching, block placement, block replacement.

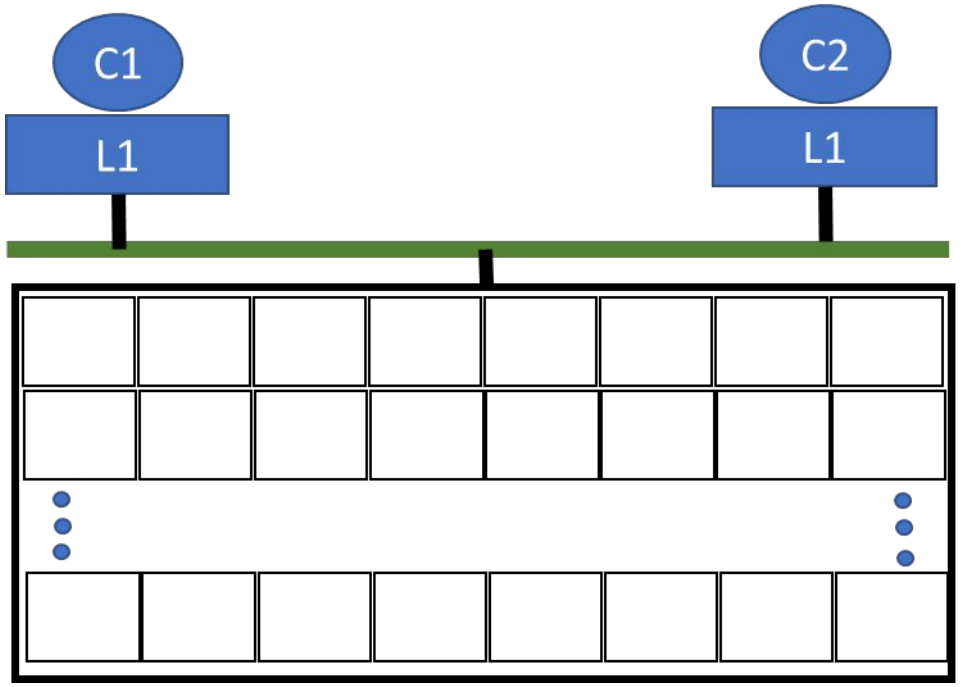
- Direct mapped cache:
  - Advantage: *Easy to implement. Easy to search.*
  - Disadvantage: *More collision of blocks.*

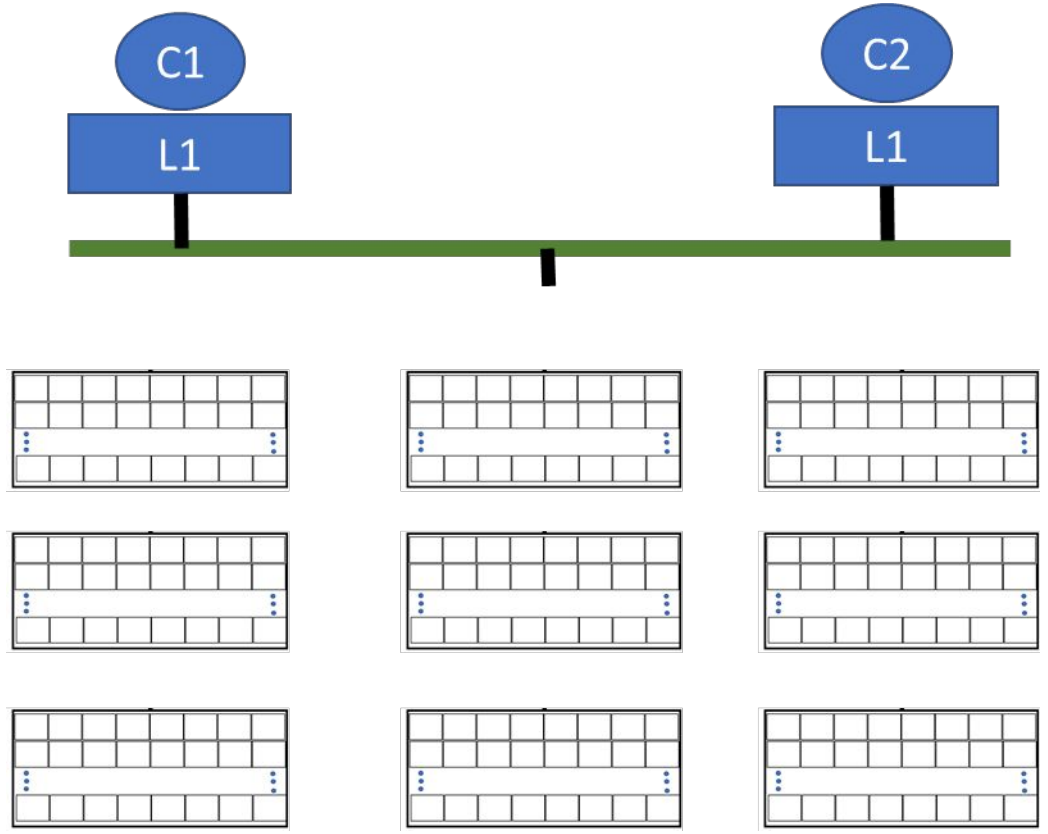
- Fully associative cache:
  - Advantage: *Less number of misses. Best utilization of the cache.*
  - Disadvantage: *Expensive to implement in hardware. Search time is also high.*

- Set associative cache:
  - Advantage:
    - *Less hardware overhead (compared to fully associative).*
    - *Less number of collisions (compared to direct mapped).*
    - *Less searching time (compared to fully associative).*
  - Disadvantage: ??

Which one is better?

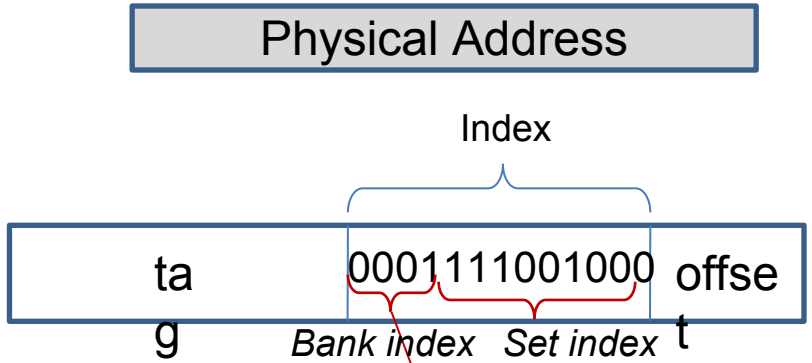
- **Cache in multicore processors**
  - Chipmultiprocessor (CMP).
  - Last Level Cache (LLC)
  - Shared vs Private





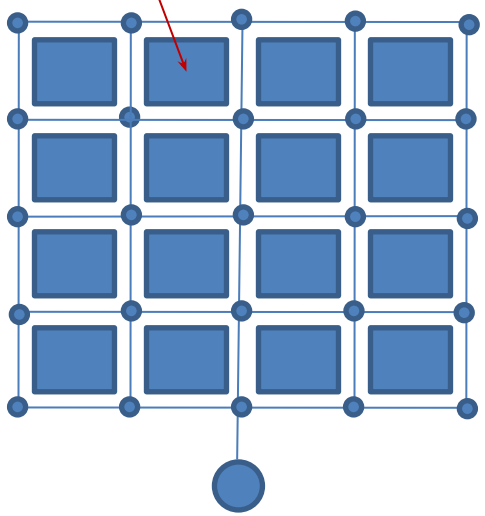
Non-Uniform Cache Access (NUCA)

Static NUCA: SNUCA:



Advantages:

- Easy to search.
- Can have more number of banks than SNUCA-1.
- Channels consumes only 5.9% of total bank area.



Disadvantages:

- Heavily used data may be mapped into a bank far from the core.
- Data cannot be migrated from one bank to another.

[1] R. Balasubramonian, N. P. Jouppi, and N. Muralimanohar, "Multi-Core Cache Hierarchies," *Morgan Claypool Publishers, 2011.*
[2] C. Kim, D. Burger, and S. W. Keckler, "An adaptive, non-uniform cache structure for wire- delay dominated on-chip caches," *SIGOPS Oper. Syst. Rev.*, vol. 36, 2002.

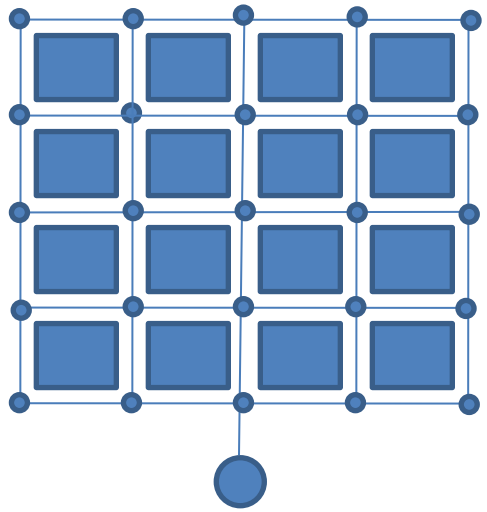
❖ D-**NUCA:**

The data management of D-NUCA is based on the following three concepts:

1. **Mapping:** How the blocks are mapped to the banks.

2. **Search:** How the set of possible locations are searched to find a block.

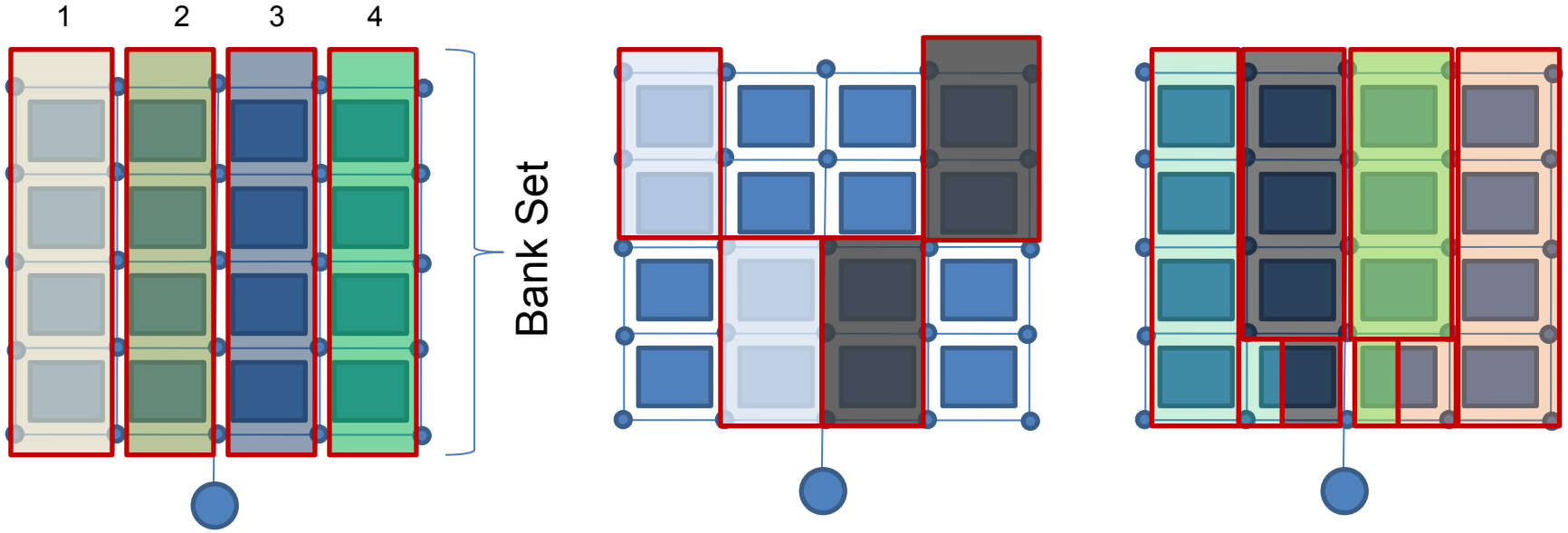
3. **Movement:** Under what conditions the data should be migrated from one bank to other.



[1] R. Balasubramonian, N. P. Jouppi, and N. Muralimanohar, "Multi-Core Cache Hierarchies," *Morgan Claypool Publishers, 2011.*
[2] C. Kim, D. Burger, and S. W. Keckler, "An adaptive, non-uniform cache structure for wire- delay dominated on-chip caches," *SIGOPS Oper. Syst. Rev.*, vol. 36, 2002.

❖ D-**NUCA Data Mapping:**

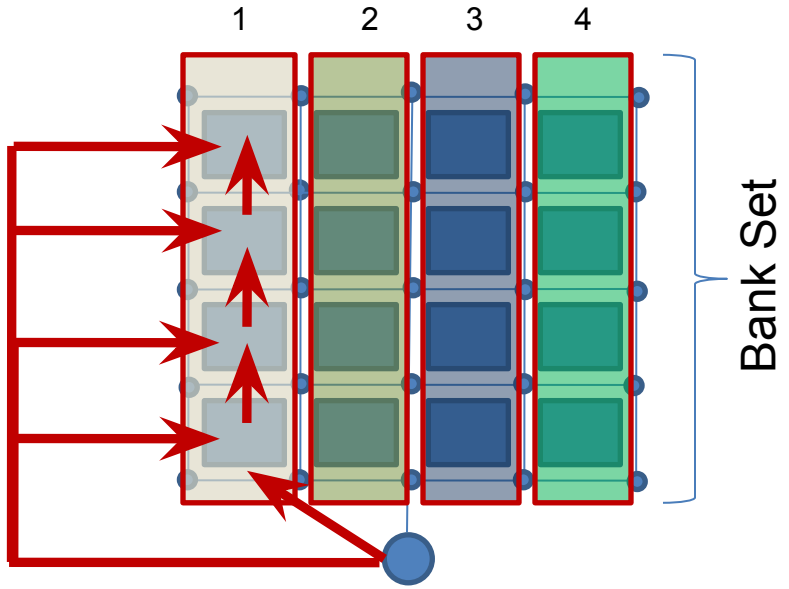
1. Data can be mapped to a single bank (S-NUCA).
2. Data can be mapped in any bank (extreme).
3. An intermediate solution called *bank-sets*.



Simple Mapping | Fair mapping | Shared Mapping

❖ D-**NUCA Searching:**

- Incremental Search.
- Multicast Search.
- Limited Multicast Search.
- Smart Search



Advantages:

- Minimum number of messages.
- Reduces access latency.
- Low energy consumption.
- Fewer banks are accessed when result is a early hit.

Disadvantages:

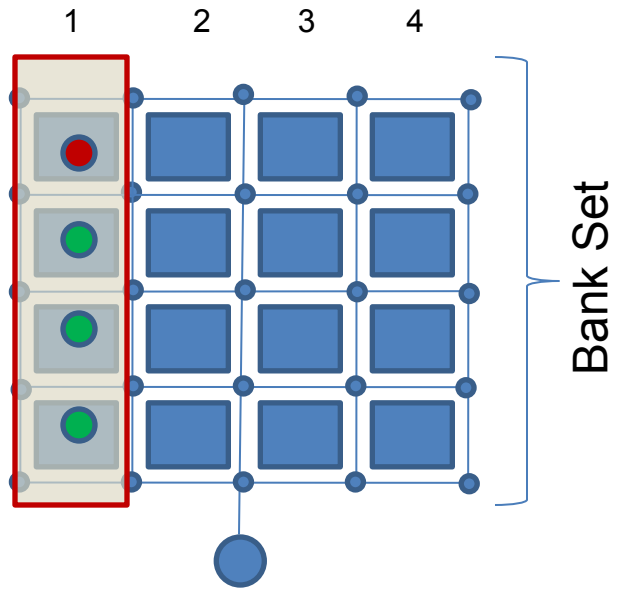
- High energy consumption.
- Increases network contention.

Read about the D-NUCA searching techniques from:

[1] Z. Sravanthi Kottapalli Ramonian, N. P. Jouppi, and N. Muralimanohar, "Multi-Core Cache Hierarchies," *Morgan Claypool Publishers, 2011.*
[2] C. Kim, D. Burger, and S. W. Keckler, "An adaptive, non-uniform cache structure for wire- delay dominated on-chip caches," *SIGOPS Oper. Syst. Rev.*, vol. 36, 2002.

❖ D-**NUCA Migration:**

- Heavily used blocks are gradually migrated towards the closer banks.
- A data is initially placed into the farthest bank and gradually move closer.

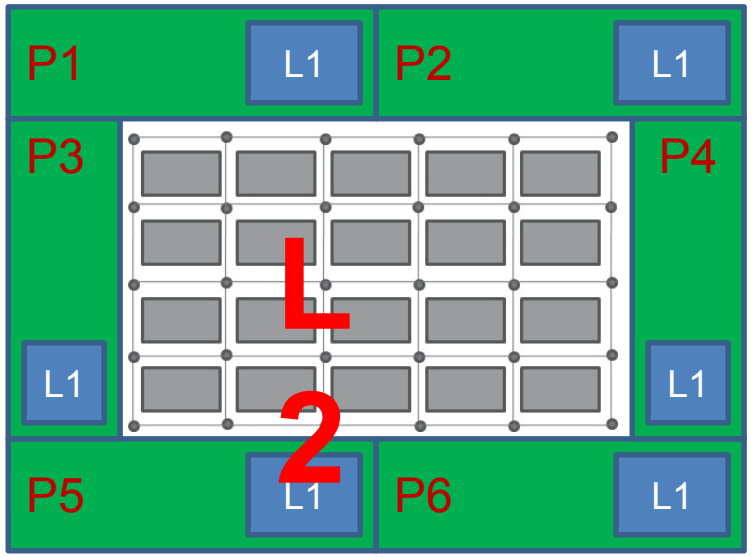


D-NUCA Migration Example

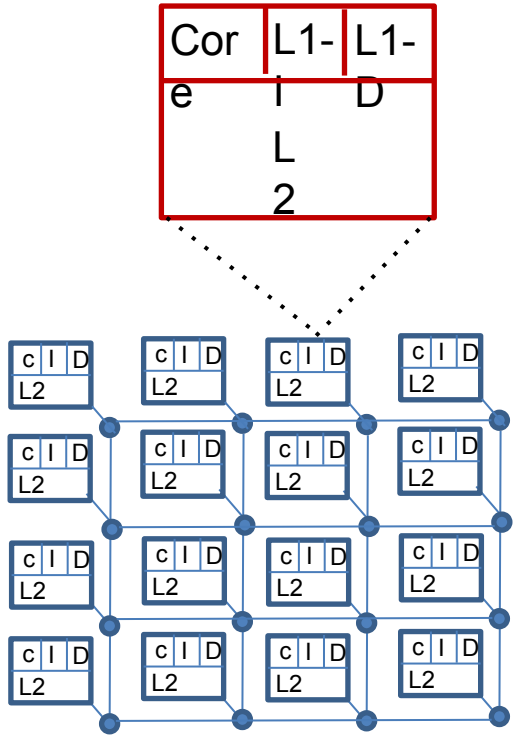
D-NUCA performs 18% better than SNUCA-2 and 20% better than monolithic shared LLC.

NUCA in CMP

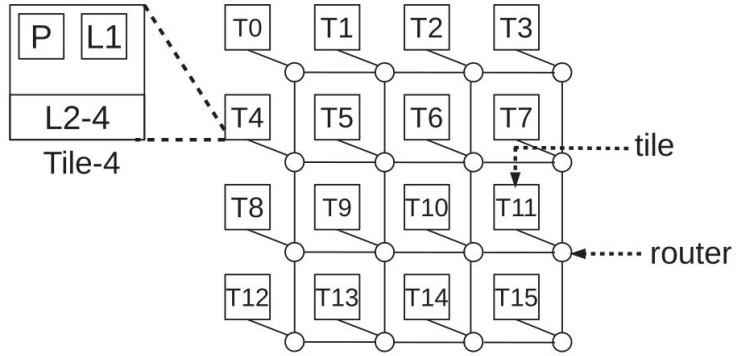
- **Chipmultiprocessor (CMP)**



CMP with Centralized LLC



Tile based CMP (TCMP)

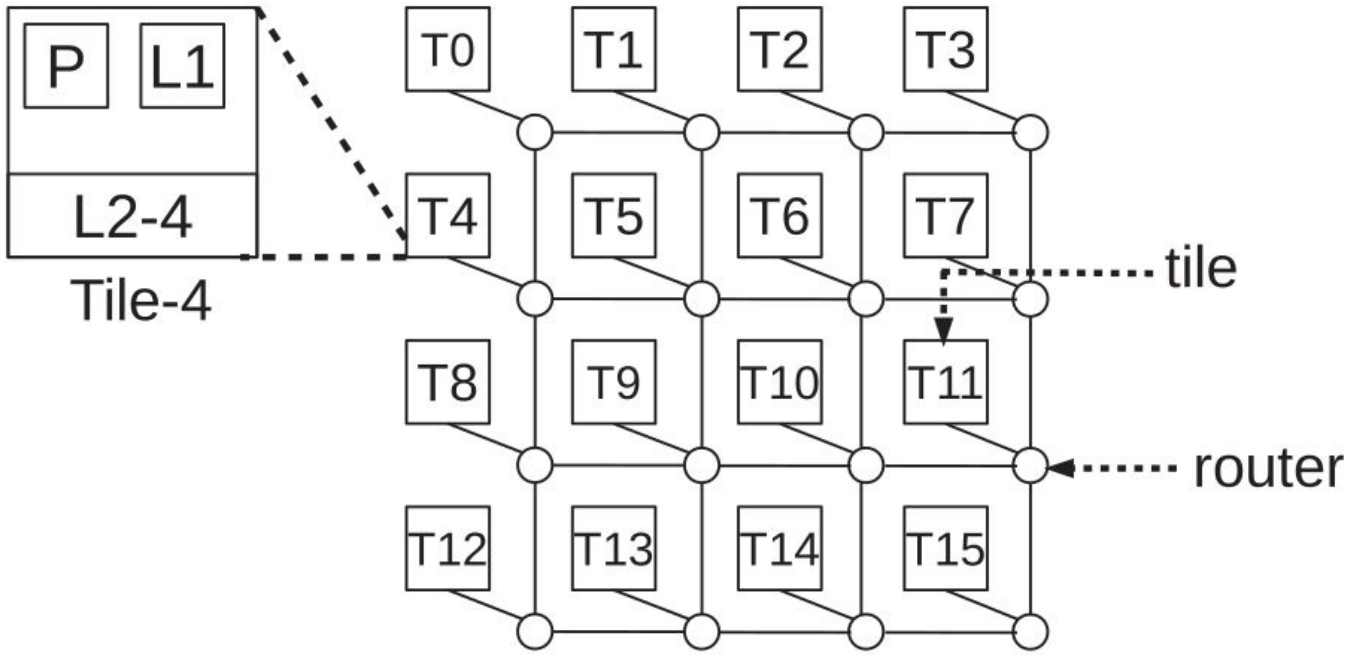


Possible Attack: Discussed later.

[1] R. Balasubramonian, N. P. Jouppi, and N. Muralimanohar, "Multi-Core Cache Hierarchies," *Morgan Claypool Publishers, 2011.*

[2] C. Kim, D. Burger, and S. W. Keckler, "An adaptive, non-uniform cache structure for wire- delay dominated on-chip caches," *SIGOPS Oper. Syst. Rev.*, vol. 36, 2002.

TCMP


P
L1
L2-4
Tile-4
T0
T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12
T13
T14
T15
tile
router

For more detail please read

1. **Appendix C** of Hennessy and Patterson, "***Computer Architecture A Quantitative Approach***", Fourth edition, 2007.
2. **Multicore Cache Hierarchies** by Rajeev Balasubramonian, Norman Jouppi and Naveen Muralimonohar.

Thank You