Microprocessor & Computer Architecture (µpCA)

UE19CS252

Dr. D. C. Kiran

Department of
Computer Science and Engineering

Microprocessor & Computer Architecture (µpCA)

Unit 4: Cache Optimization

Dr. D. C. Kiran
Department of Computer Science and Engineering

Microprocessor & Computer Architecture (µpCA)

Syllabus

~~**Unit 1: Basic Processor Architecture and Design**~~

~~**Unit 2: Pipelined Processor and Design**~~

~~**Unit 3: Memory**~~

Unit 4: Input/Output Device Design

~~3 C~~

~~Introduction to Cache Optimization~~

~~Reduce Miss Rate~~

~~Reduce Miss Penalty~~

~~4th Optimization: Multilevel Caches to Reduce Miss Penalty~~

~~5th Optimization: Giving priority to Read misses over Write misses to~~

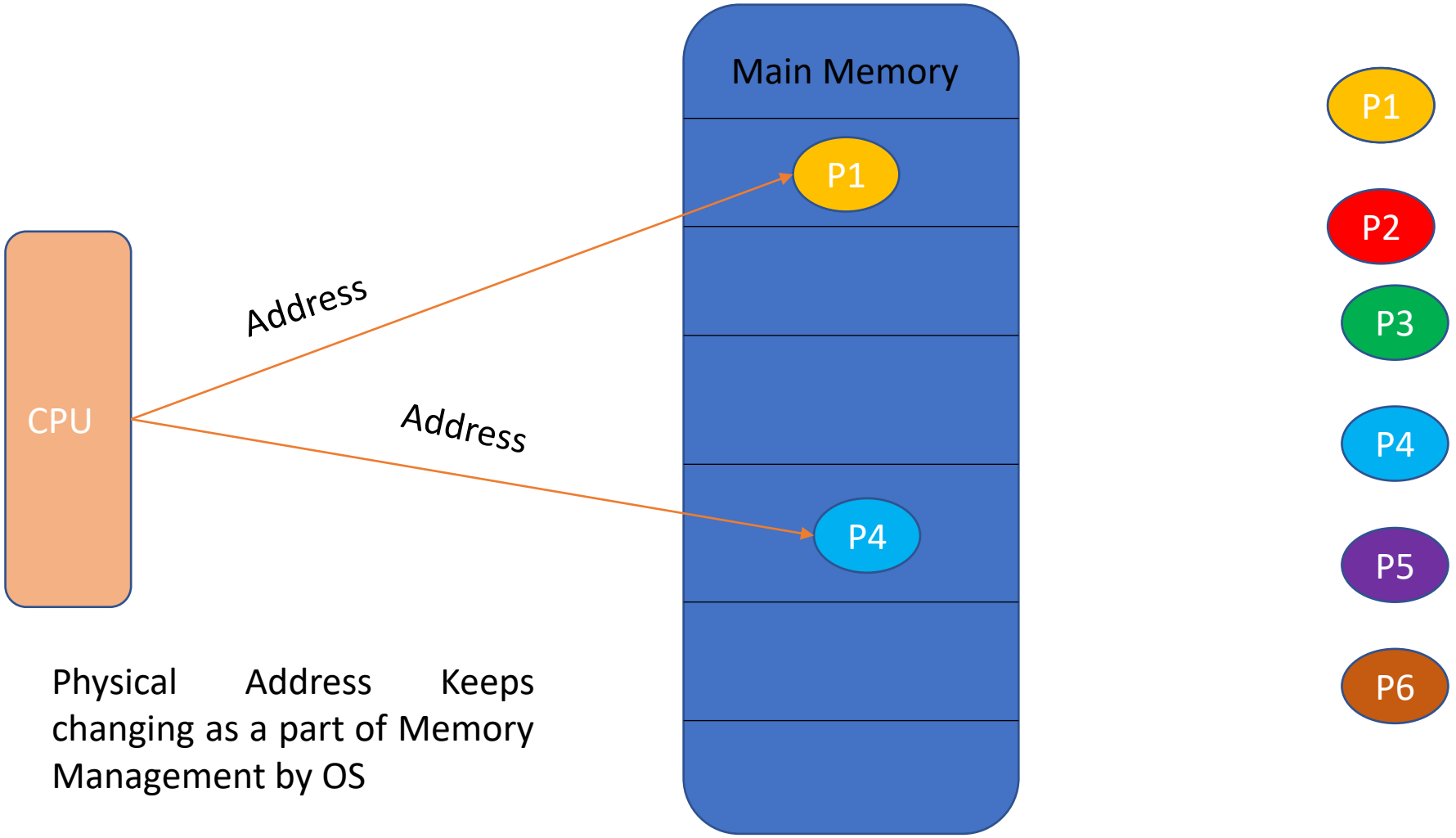
Reduce Hit Time

Unit 5: Advanced Architecture

Microprocessor & Computer Architecture (µpCA)

6th Optimization: Reduce Hit Time

Avoid Address Translation In Cache Indexing To Reduce Hit Time

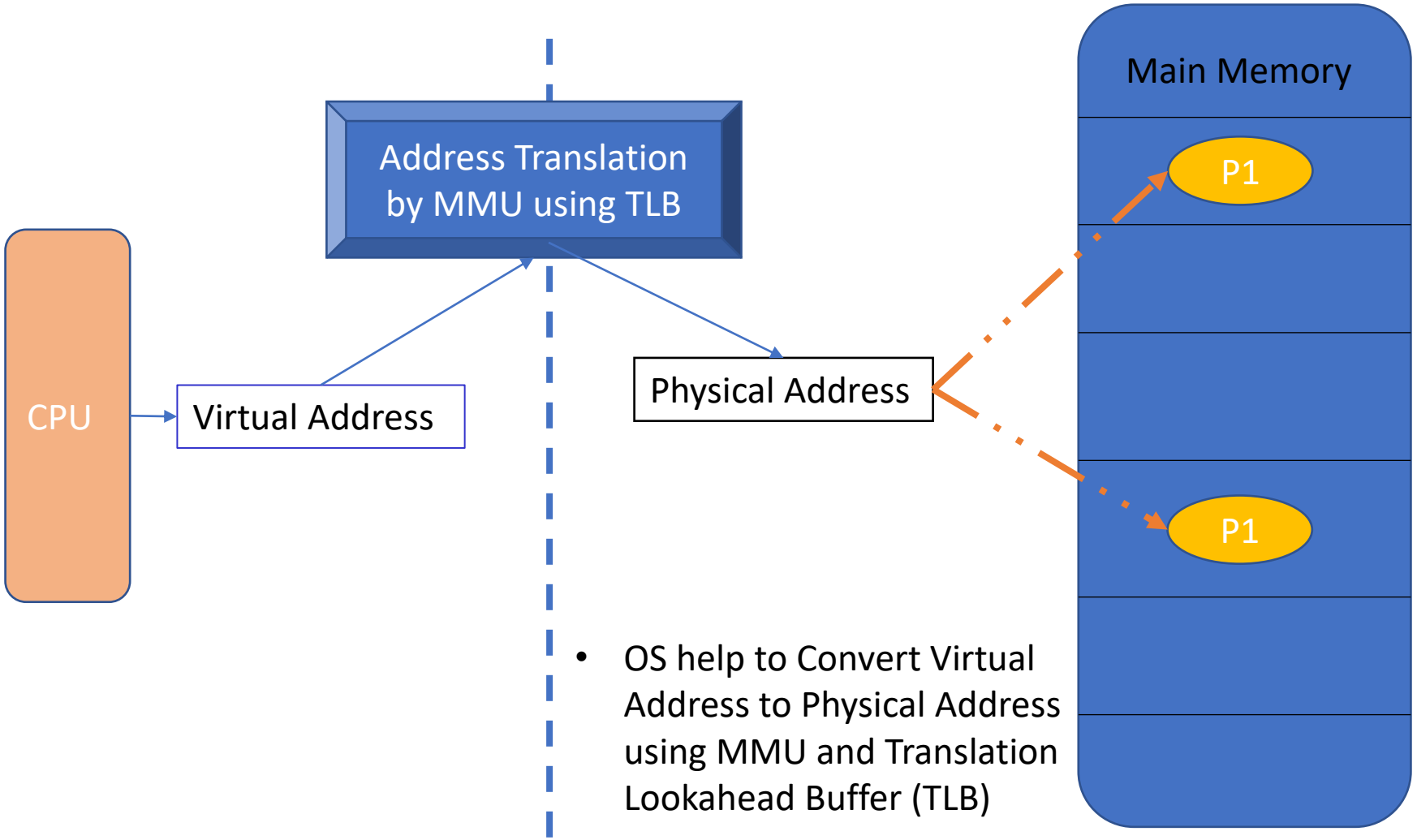


Physical Address Keeps changing as a part of Memory Management by OS

Microprocessor & Computer Architecture (µpCA)

6th Optimization: Reduce Hit Time

Avoid Address Translation In Cache Indexing To Reduce Hit Time



- OS help to Convert Virtual Address to Physical Address using MMU and Translation Lookahead Buffer (TLB)

What is the Problem?

Accessing data from Cache involve:

- Indexing
- Tagging

Converting Virtual Address to Physical Address take some time.

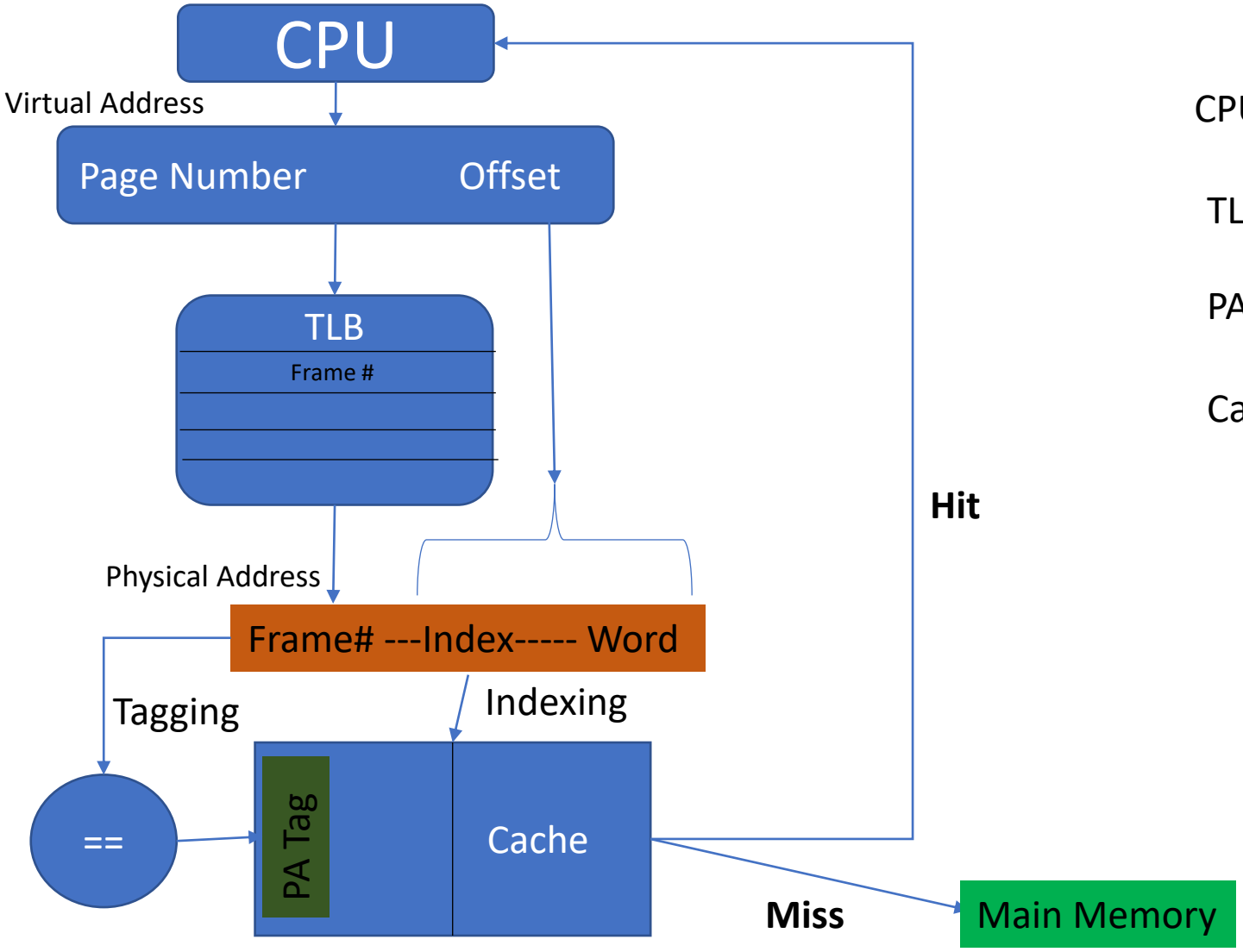
Solution ☺

Virtually Indexed & Physically Tagged

- Do not wait for VA to PA translation.
- Extract Index information from Virtual Address
- Extract Tag Information from the Physical Address.

Microprocessor & Computer Architecture (µpCA)

Physically Indexed and Physically Tagged (PIPT)



CPU generate VA

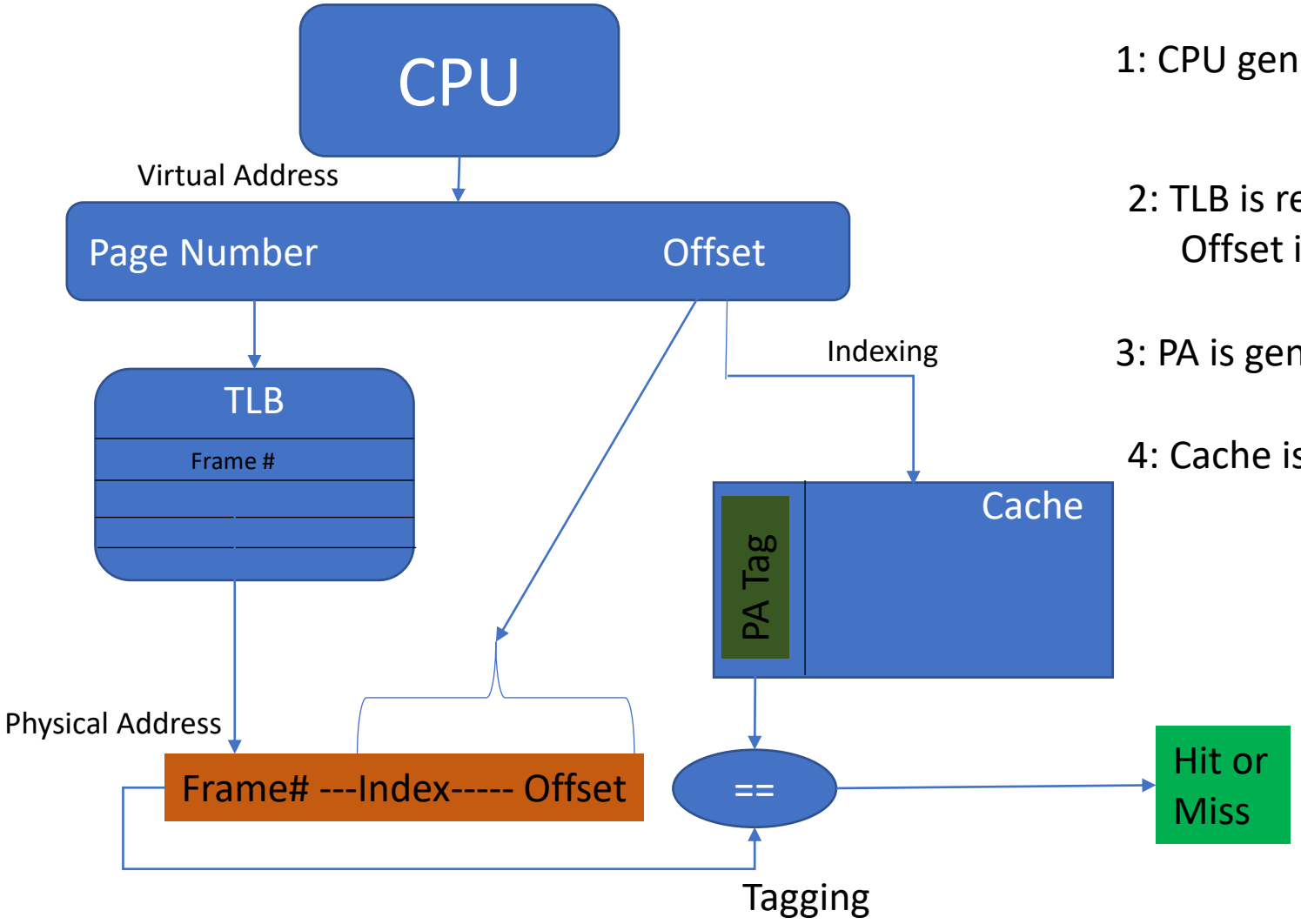
TLB is referred to check if required Frame present or not

PA is generated

Cache is Indexed and Tagged using PA

Microprocessor & Computer Architecture (µpCA)

Virtually Indexed and Physically Tagged



1: CPU generate VA

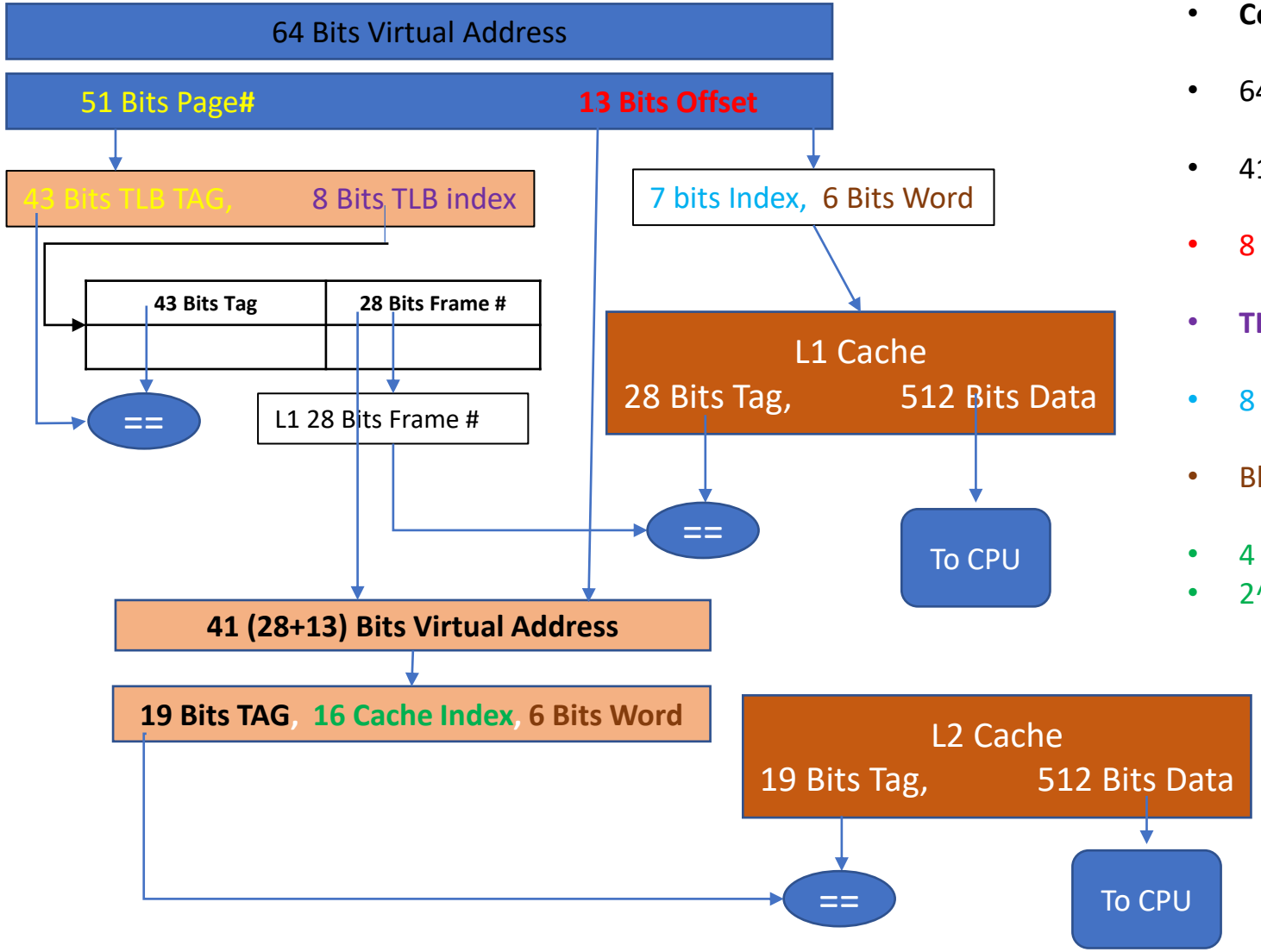
2: TLB is referred to check if required Frame present or not & Offset is used to Index Cache

3: PA is generated

4: Cache is Tagged using PA

Microprocessor & Computer Architecture (µpCA)

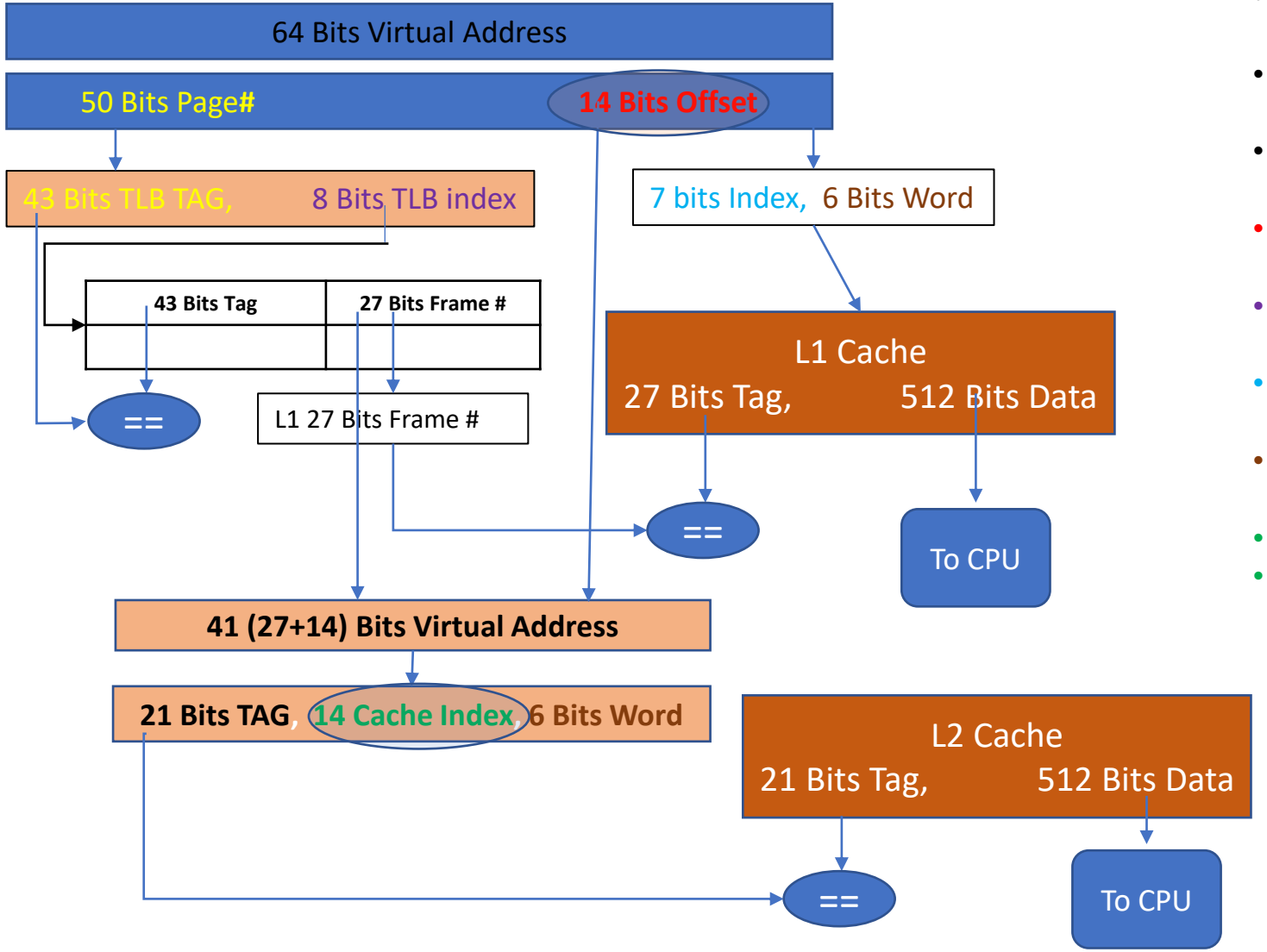
6th Optimization : Case Study



- **Consider the following Specification**
- 64 bit Virtual Address
- 41 bit Physical Address
- 8 KB (2^13)Page Size
- **TLB is Direct Mapped with 256 Entries**
- 8 KB (2^13) Direct Mapped L1 Cache
- Block Size = 64 Bytes (2^6 words)
- 4 MB Direct Mapped L2 Cache
- 2^22/2^6= 2^16

Microprocessor & Computer Architecture (µpCA)

6th Optimization : Optimized to Improve the Hit Time



- **Modified Specification for Cache**
- 64 bit Virtual Address
- 41 bit Physical Address
- 16 KB (2^14)Page Size
- **2 Way, 256 Entry TLB**
- 16 KB (2^13) Direct Mapped L1 Cache
- Block Size = 64 Bytes (2^6 words)
- 4 MB, 4 Way, L2 Cache
- 2^22/2^6x 2^2= 2^14

Think About It?

Why Can’t we Virtually Index and Virtually TAG?

Reference

Next Session

Input & Output devices

THANK YOU

Dr. D. C. Kiran
Department of Computer Science and Engineering

dckiran@pes.edu

9829935135

What is the Problem?

Accessing data from Cache involve:

- Indexing
- Tagging

Converting Virtual Address to Physical Address take some time.

Solution ☺ Example:

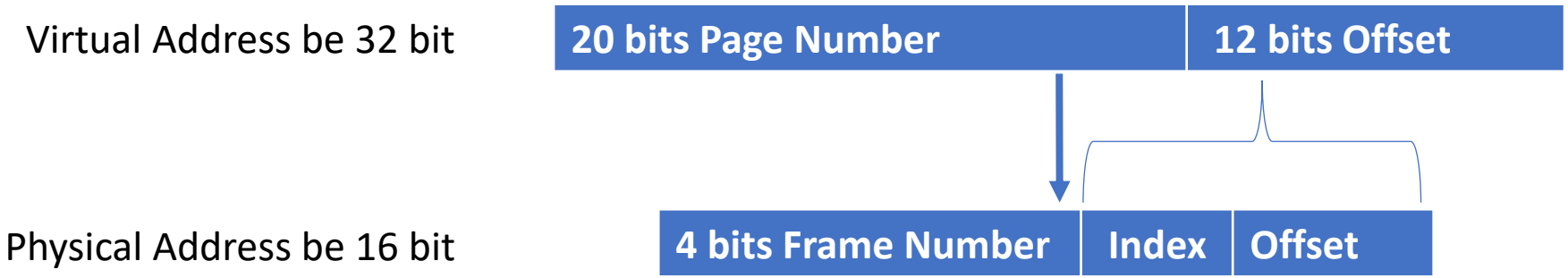
Virtual Address be 32 bit

Physical Address be 16 bit

Microprocessor & Computer Architecture (µpCA)

6th Optimization: Reduce Miss Time

Avoid Address Translation In Cache Indexing To Reduce Hit Time



Solution ☺

Do not wait for VA to PA translation.

Extract Index information from Virtual Address

Extract Tag Information from the Physical Address.

6th Optimization facilitate Virtual Indexing and Physically Tagging (VIPT)