

# COL216

# Computer Architecture

Memory Organization

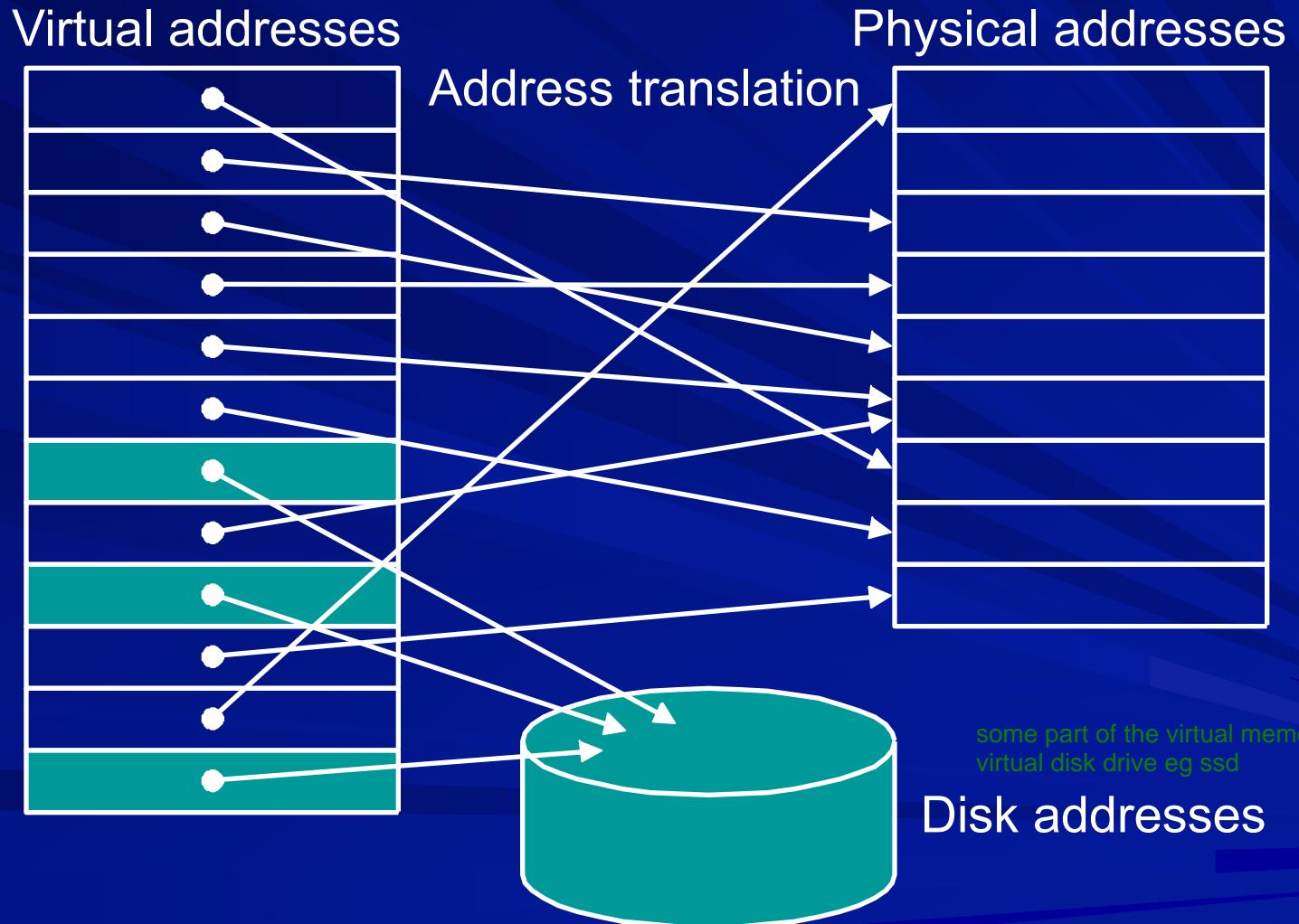
Virtual Memory

12th March 2022

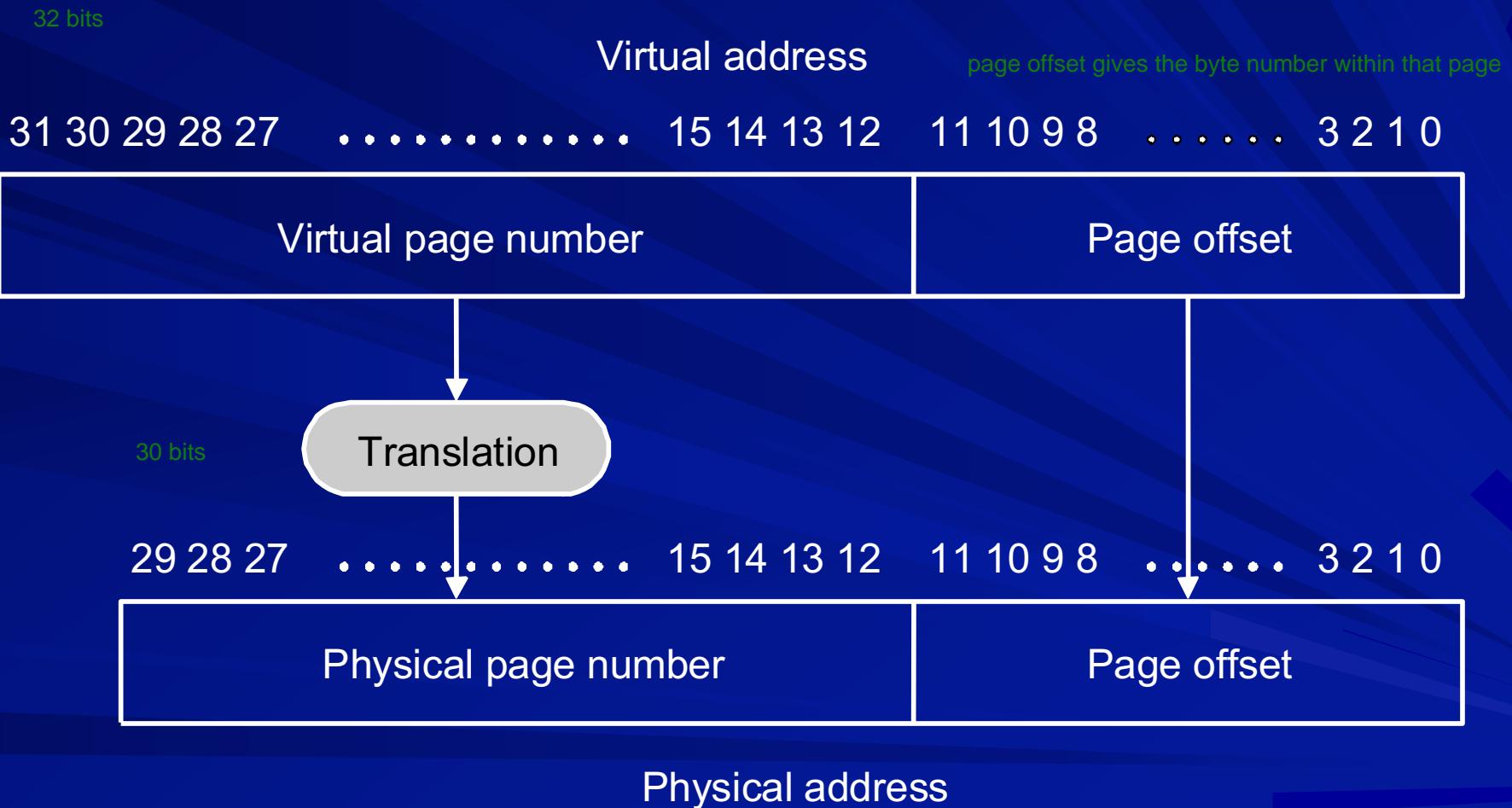
# Virtual memory objectives

- To overcome the limitation of the physical memory size
- To allow multiple programs to share memory
  - Protection don't want other programs to mess up with the memory assigned for a particular program
  - Program relocation machine code can be placed anywhere

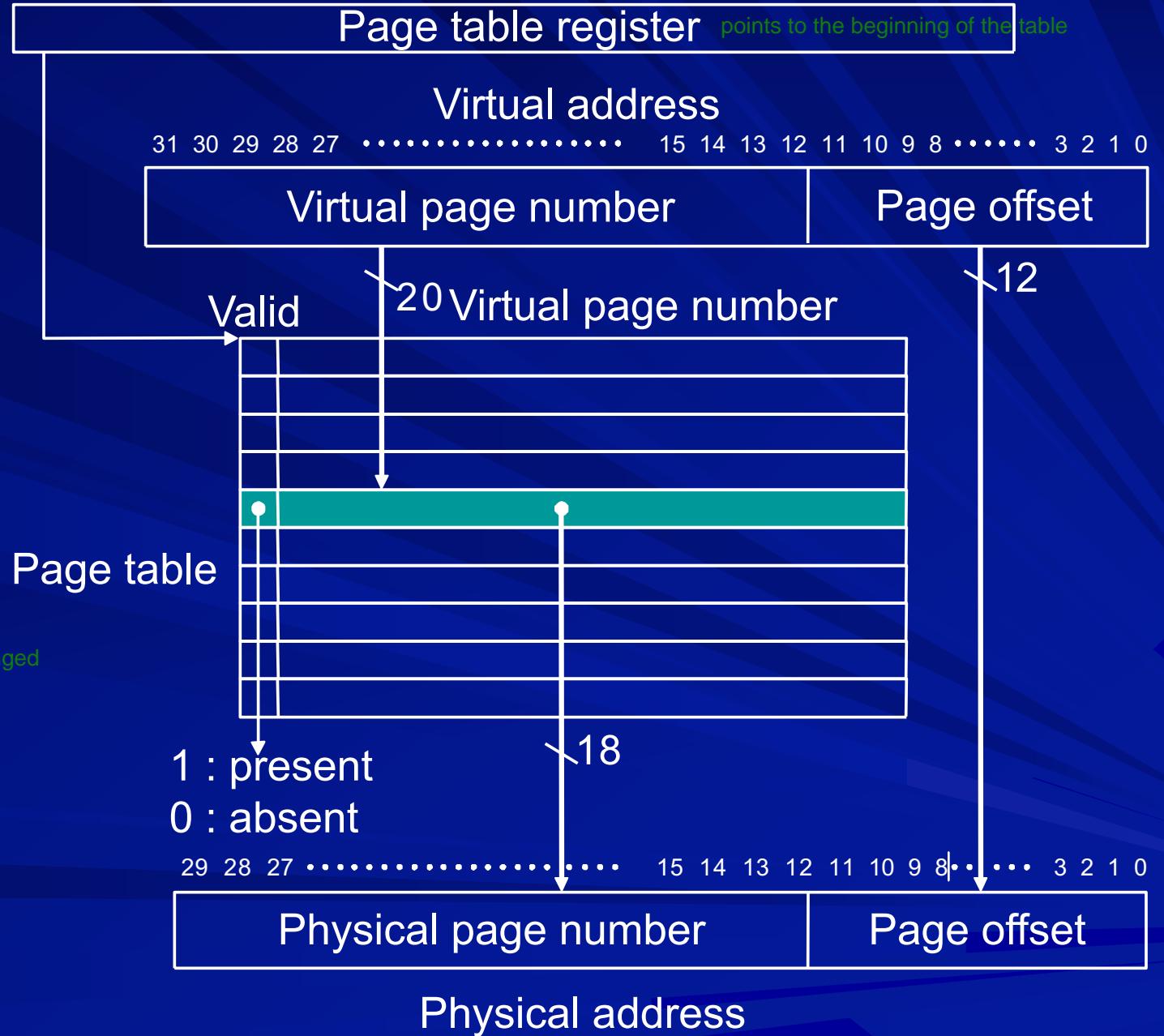
# Virtual to physical mapping



# Address translation



# Page table



# Mapping with page table

virtual page number



Page table:

v physical page  
or disk addr

1	•
1	•
1	•
1	•
0	•
1	•
1	•
0	•
1	•
1	•
0	•
1	•

Physical memory



Disk storage



0 means see in disk storage

1 means see in physical memory

# Virtual memory vs cache : similar but important differences

## ■ Speed difference

- cache and main memory : one order of magnitude
- main memory and **HDD**: several orders of magnitude

## ■ Response to a miss

- cache : must be handled by h/w, can't afford to switch context
- VM : can't keep CPU waiting, must switch context, can be conveniently handled by s/w

VM handled by S/W since we need to do other tasks while stalls / info is being fetched

## ■ Terminology differences

- **page** vs block(cache line), **page fault** vs miss, page table vs cache directory etc

page not found

# Differences contd.

## ■ Miss rate

- VM can only afford extremely low miss rates
- main memory much larger than cache, implies much lower miss rate

## ■ Policies

bring in a page so larger but better

- pages much larger than blocks (~4KB - 16 KB) : capture large spatial locality, amortize transfer time
- fully flexible mapping (assoc mem not used) no parallel computation
- always write back dont write one word in disk, no write through
- good approximation to LRU

# Back to page table

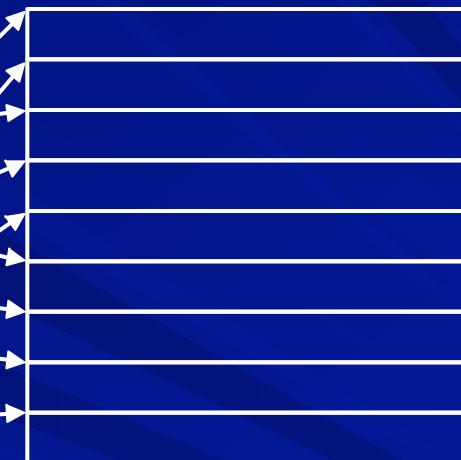
virtual page number



Page table:  
physical page  
or disk addr

1	•
1	•
1	•
1	•
0	•
1	•
1	•
0	•
1	•
1	•
0	•
1	•

Physical memory



Disk storage



# Where is page table stored?

How big is it?

- Suppose virtual address = 32 bits, page size = 4 KB, page table entry = 4 B
- then number of page table entries = number of pages =  $2^{32} / \cancel{2^{12}} = 2^{20}$
- page table size =  $2^{20} \times 2^2 = 4 \text{ MB}$
- hundreds of processes

Store in dedicated memory? main memory?

# Handling large page tables

- Bound the page table size
- Exploit sparseness
- Use multiple levels
- Page the page table

# Bounding the page table size

- Keep a bound register
  - Allocate as much space as necessary
  - Address space expands in one direction
- Split the virtual space into two parts, each with its own bound, e.g. heap + stack
  - Two segments can grow independently
- Sparseness not exploited

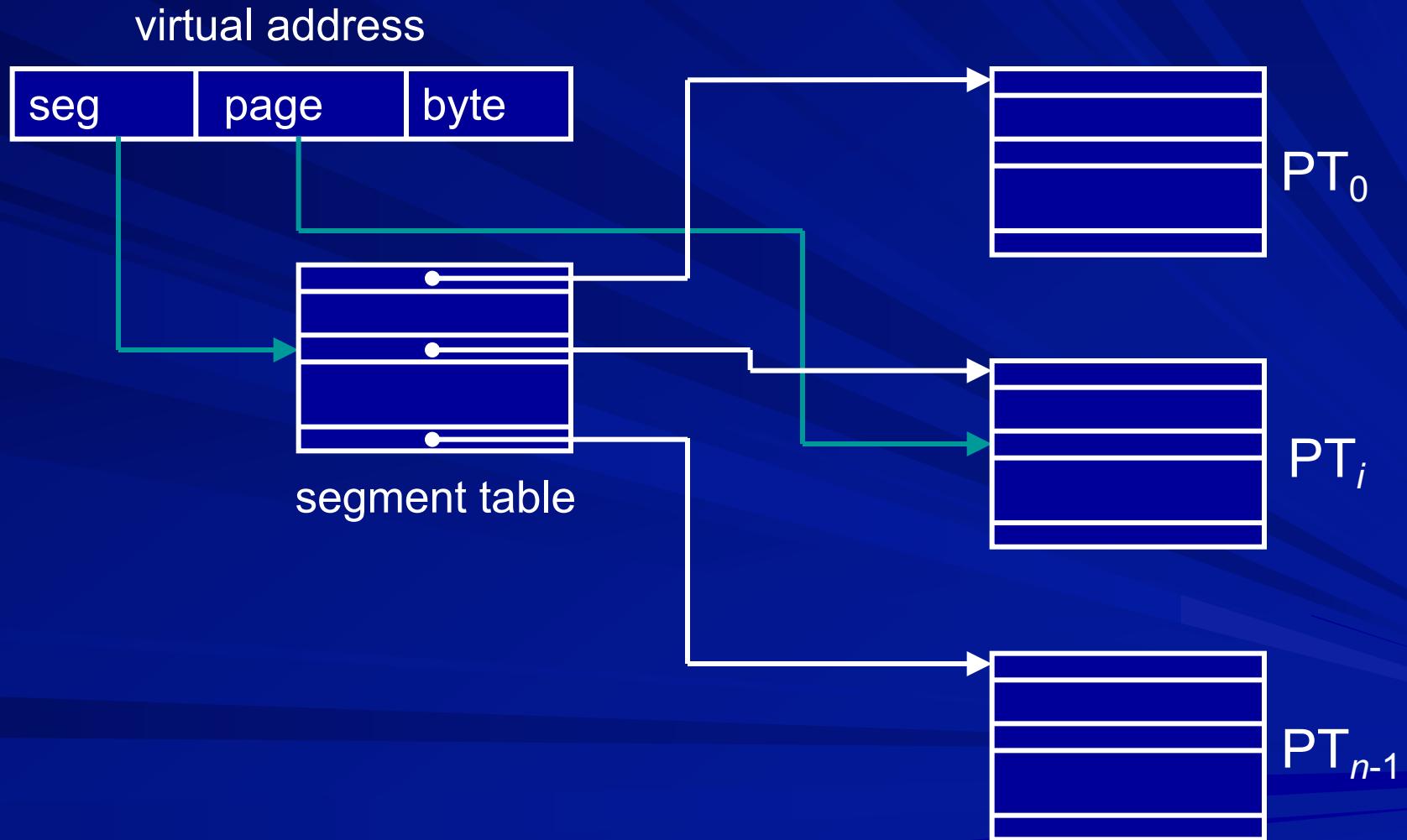
# Exploit sparseness

- Page table: one entry per virtual page,  
Cache: one entry per cache block
  - Associative memory impractical for page table
- Hashing can be used - inverted page table
  - Access is more complex than simply indexing into the page table

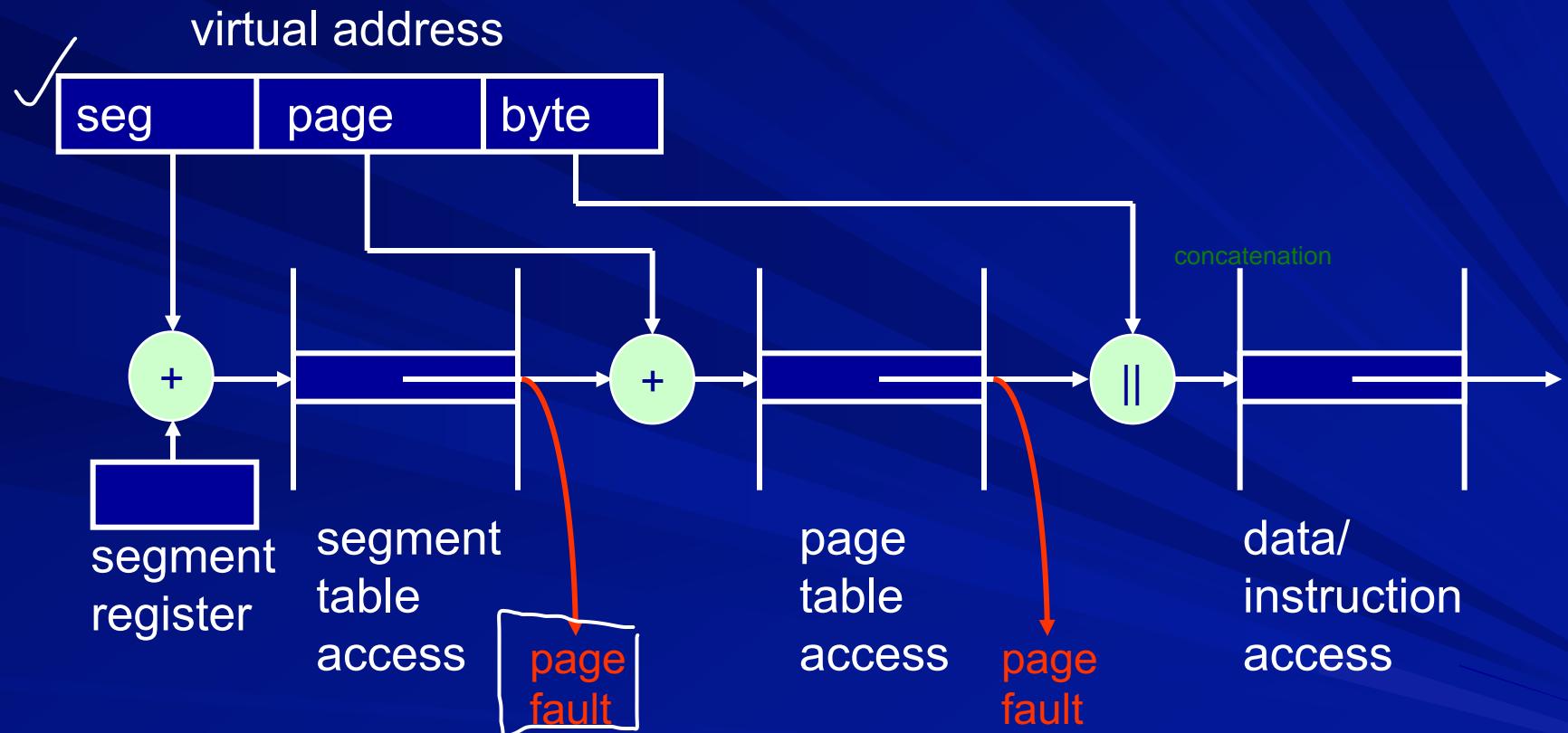
# Two level page tables

- Virtual space divided into segments
- A smaller page table for each segment
- All page tables need not be in physical memory
- A segment table keeps track of where the page tables of different segments are located

# 2 Level page table structure



# Memory access with 2 level page table



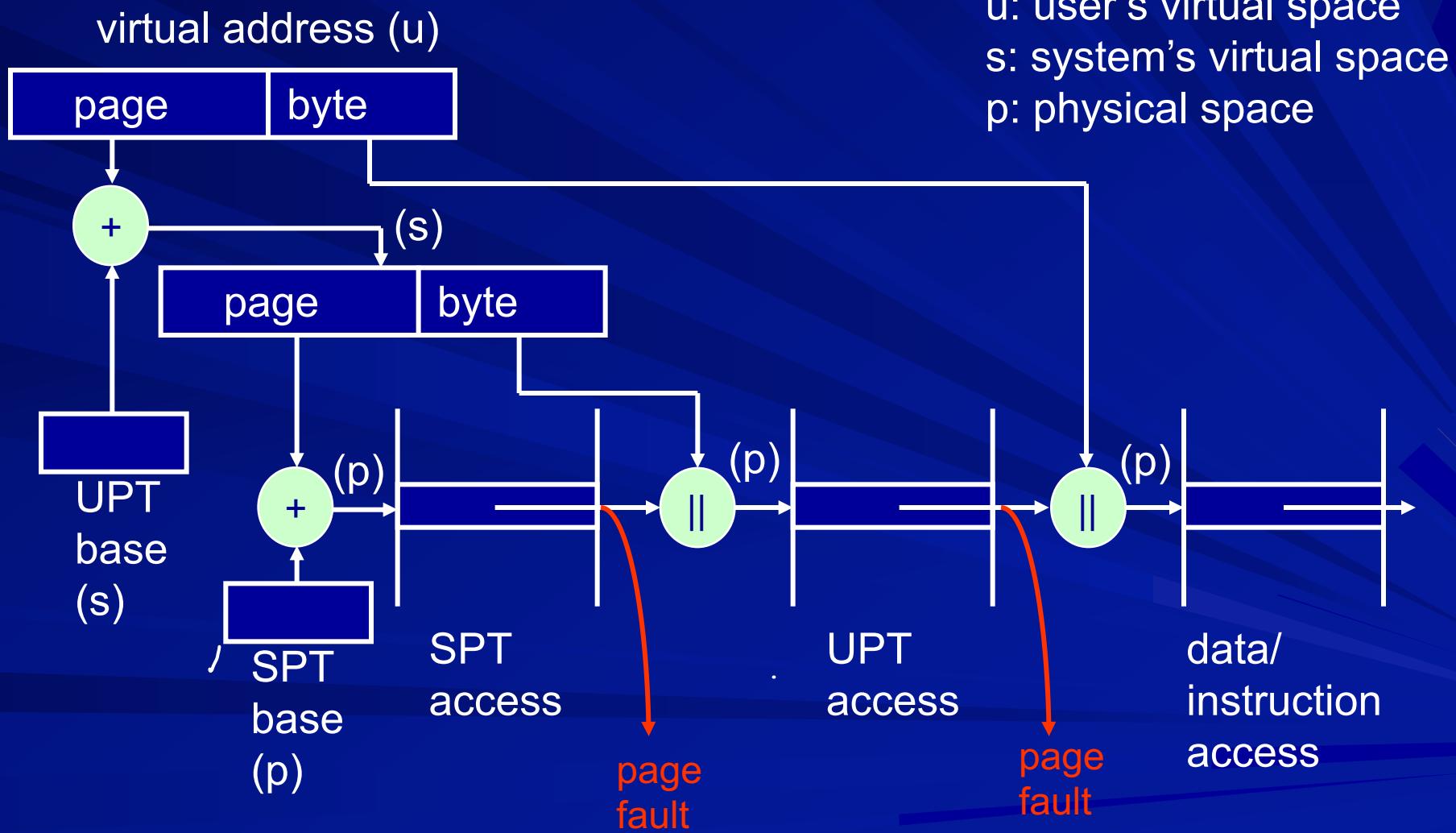
segment table is in main memory

so at the segment table level, if you get a page fault you bring the entire page table into the memory else if you get a page fault at the page

# Keep page table in virtual space

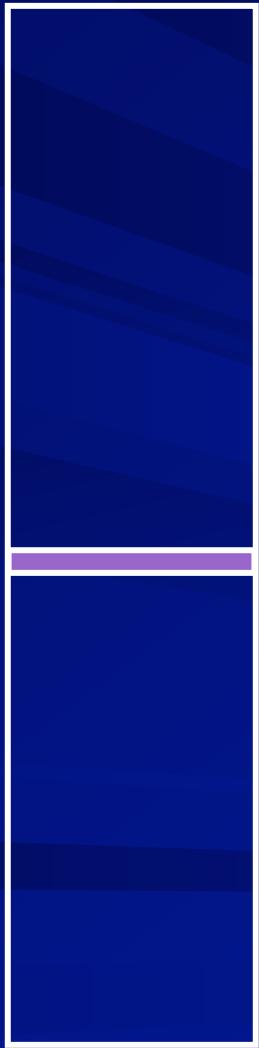
- Page table is paged!
- Only active pages of page table are brought into physical memory  
not doing segmentation here
- How do you locate these pages? - need another page table!
- User page tables are kept in system's virtual space  
User and OS have different page tables
- System's page table (or at least a part of it) must be kept in physical memory and accessed directly  
Imagine system's page table is in the physical memory

# Memory access with paged page table



user  
virtual space

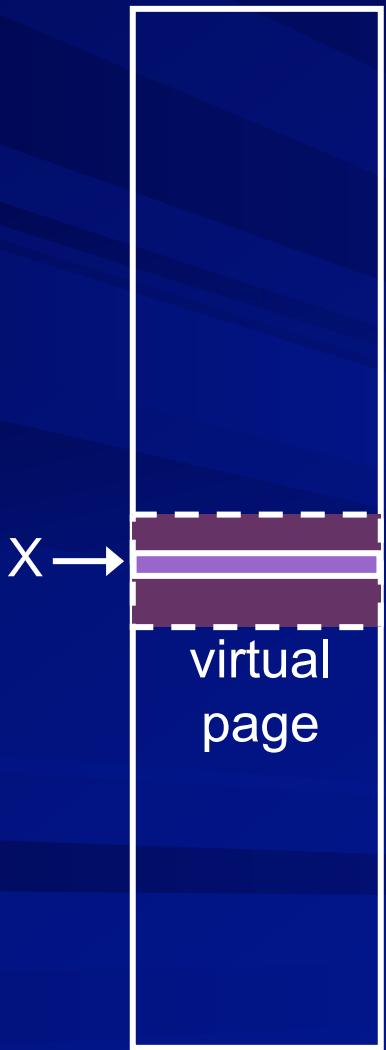
X →



physical  
space



## user virtual space



## physical space

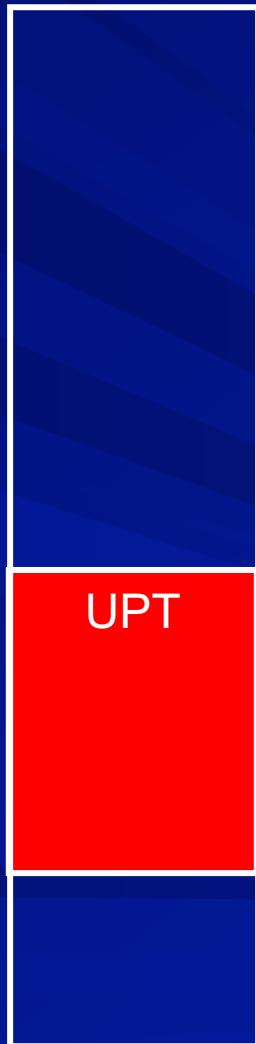


mapping is page to page  
look at the entire page

user  
virtual space



system  
virtual space

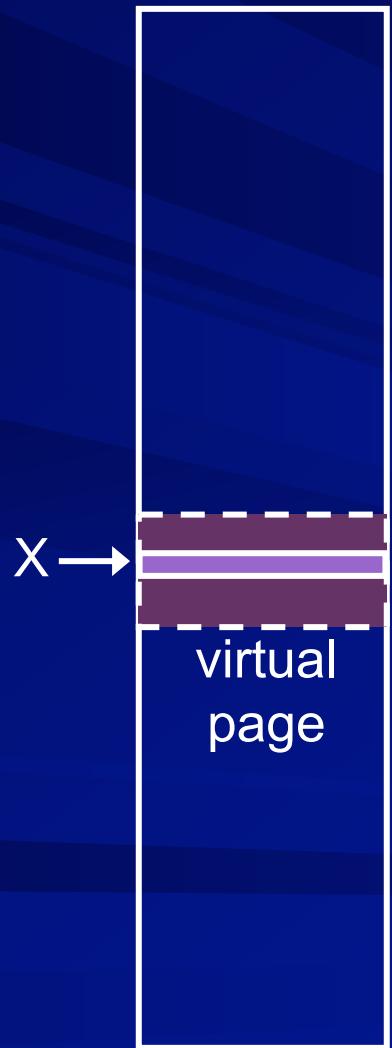


physical  
space

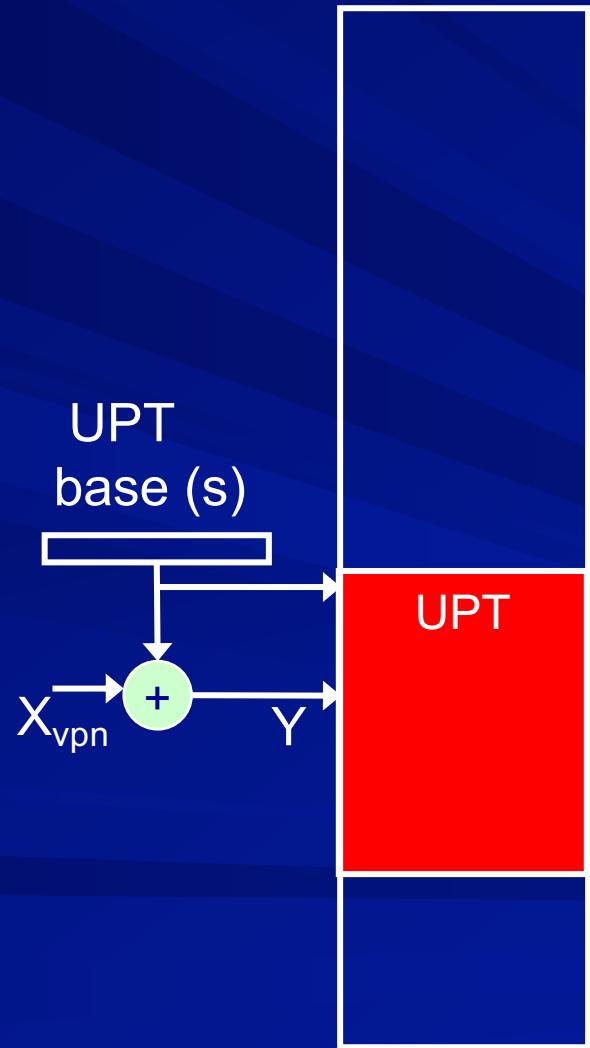


physical  
page

user  
virtual space



system  
virtual space



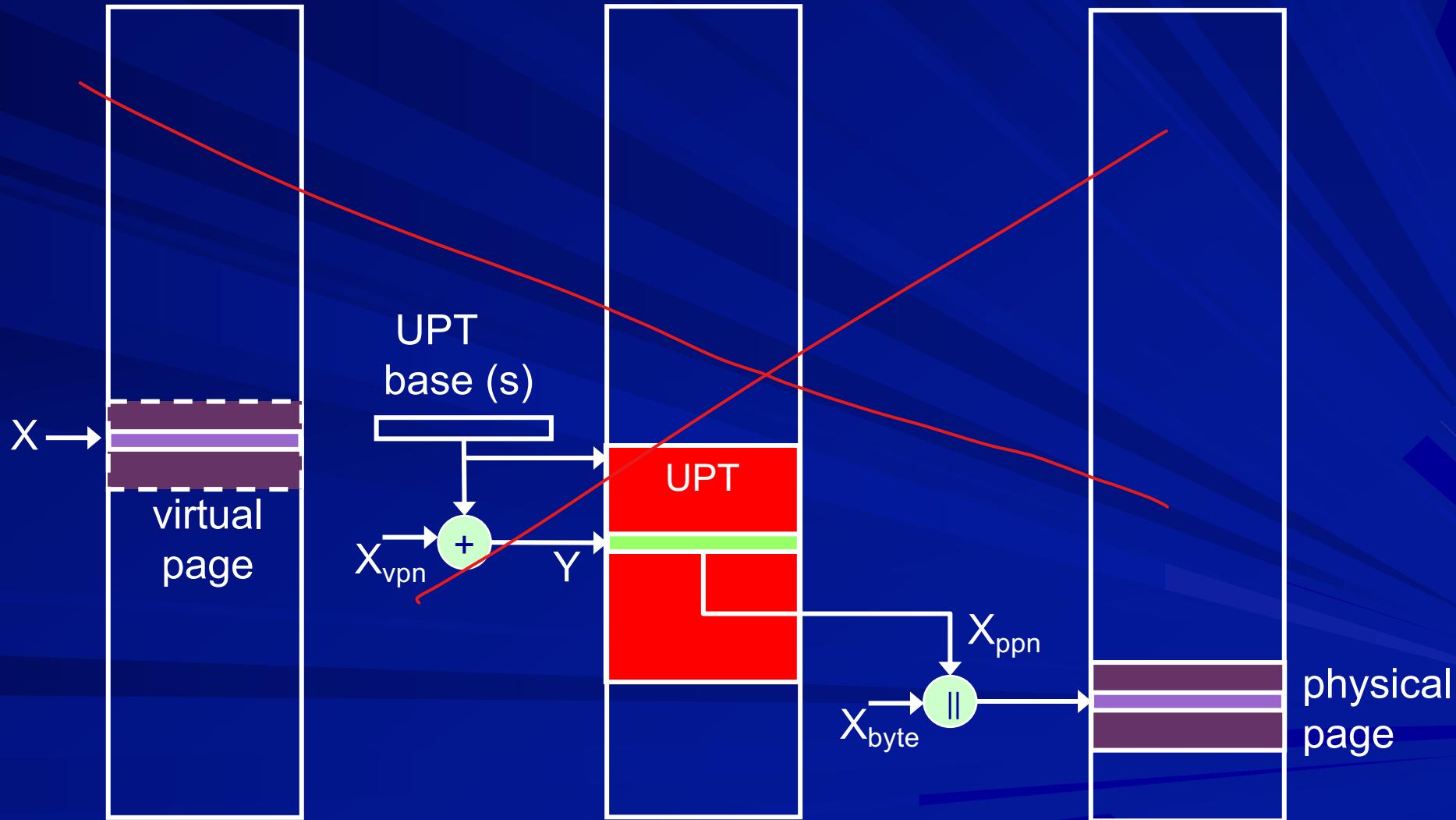
physical  
space



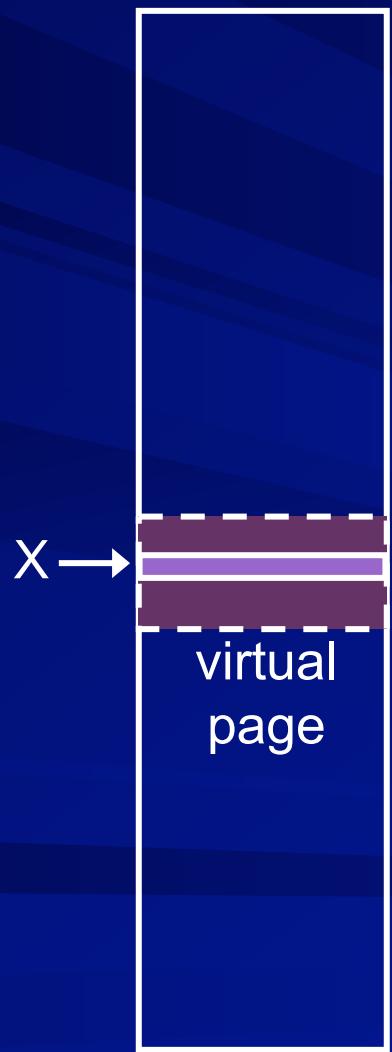
user  
virtual space

system  
virtual space

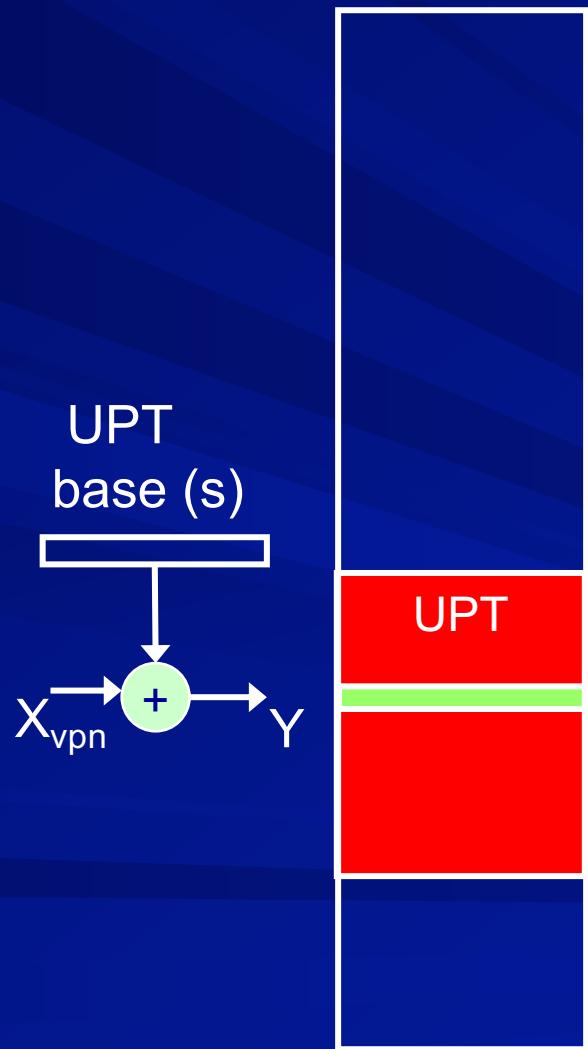
physical  
space



user  
virtual space



system  
virtual space



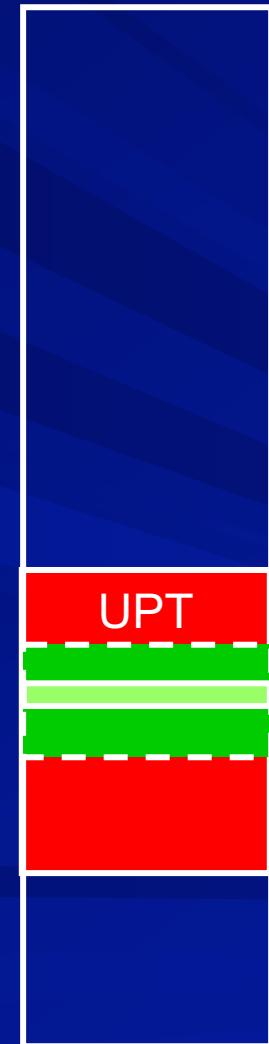
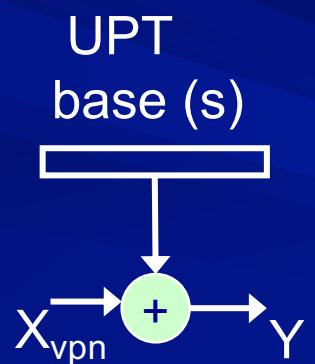
physical  
space



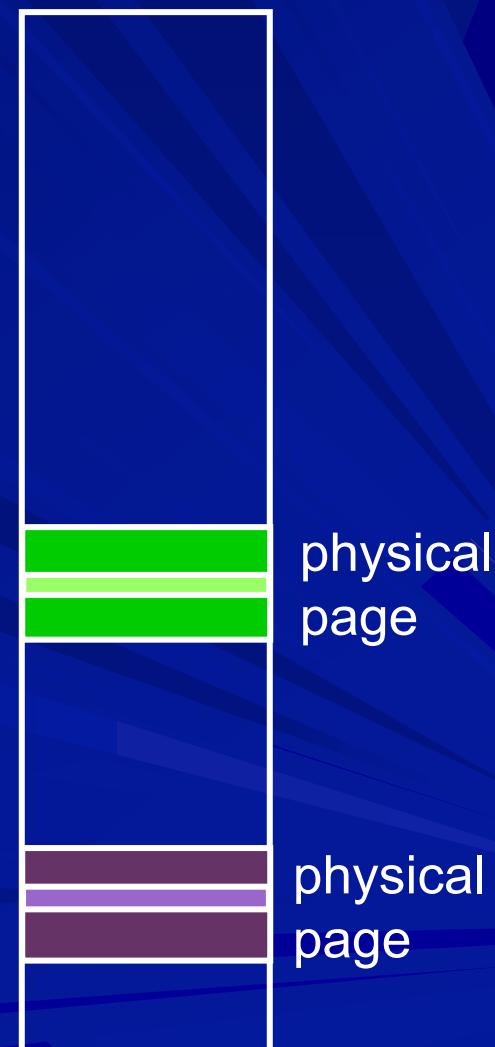
user  
virtual space



system  
virtual space



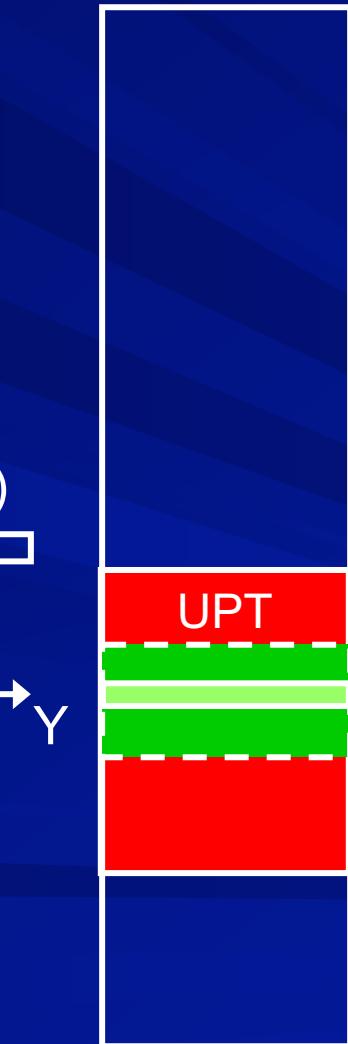
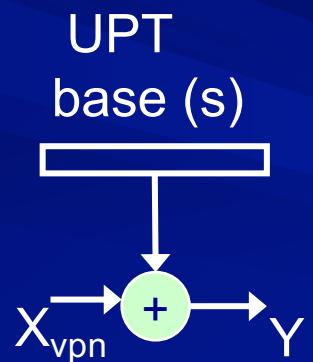
physical  
space



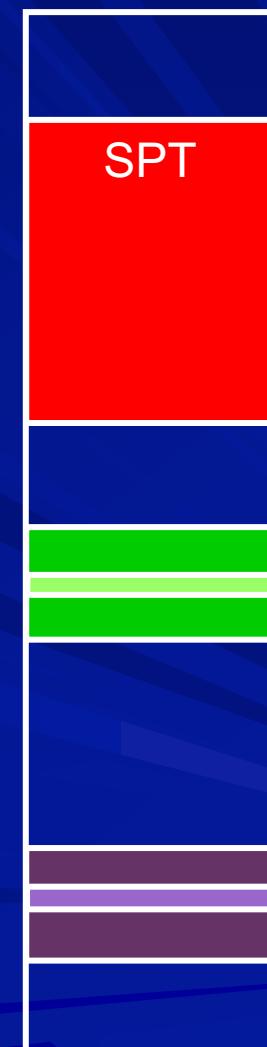
user  
virtual space



system  
virtual space



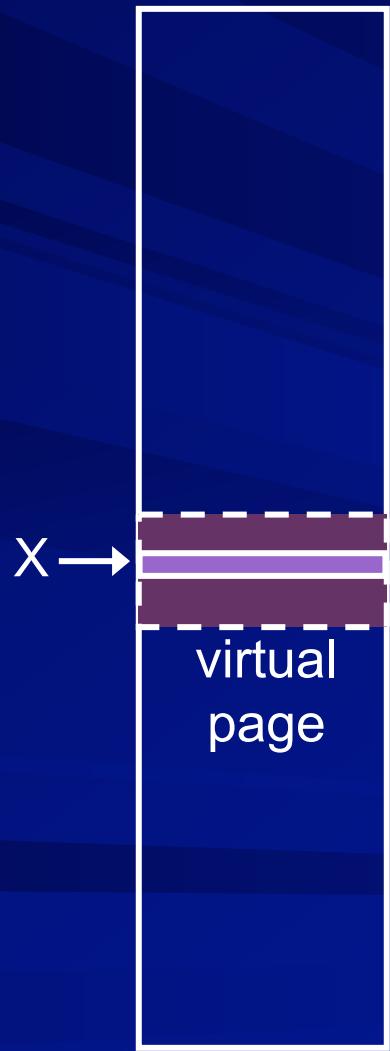
physical  
space



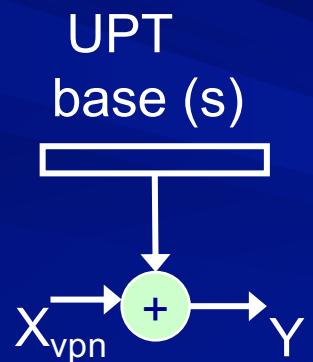
physical  
page

physical  
page

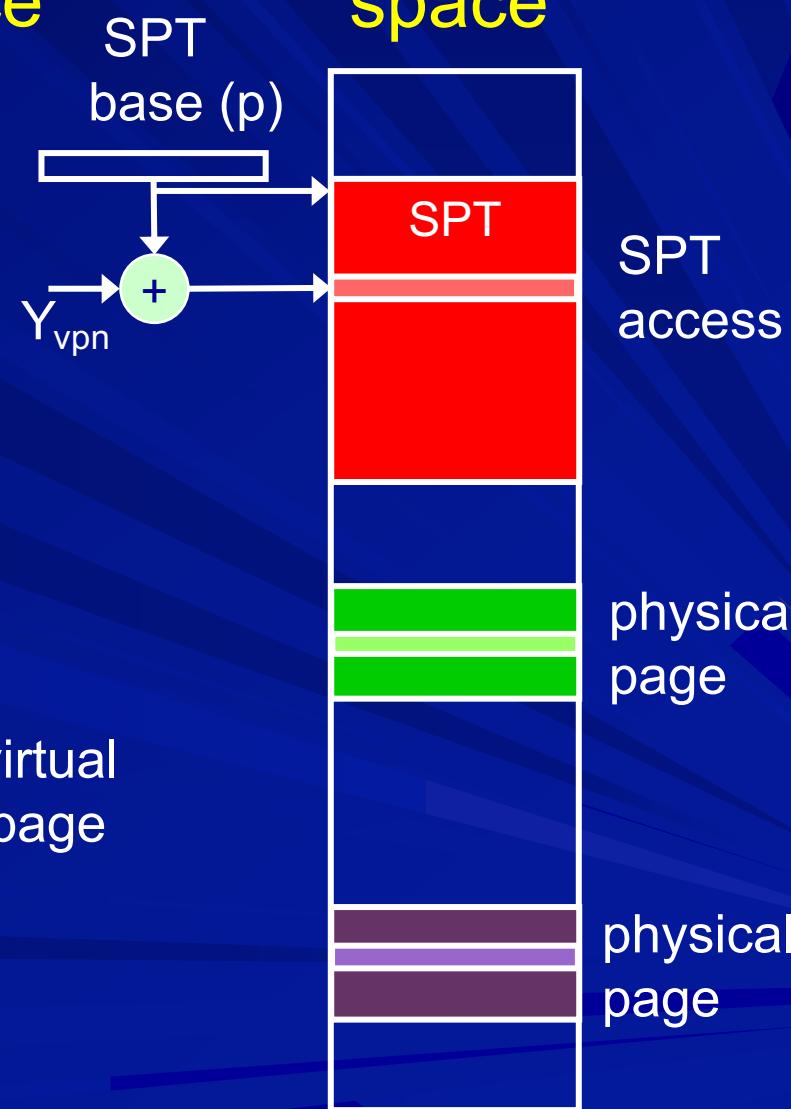
## user virtual space



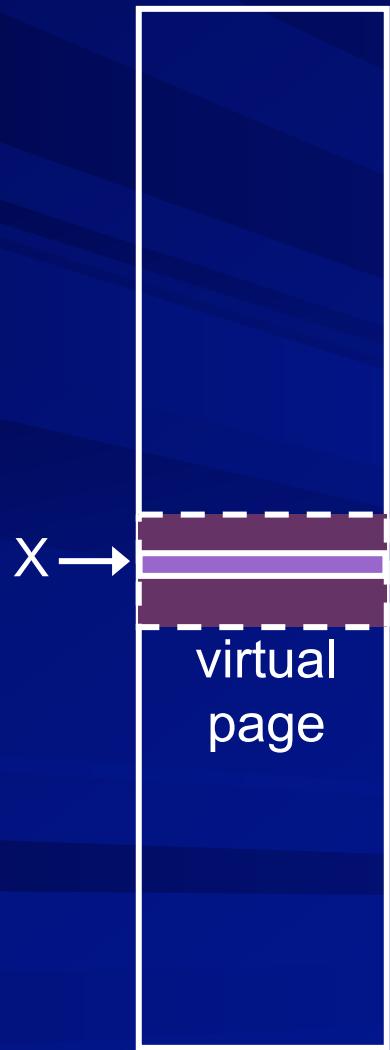
## system virtual space



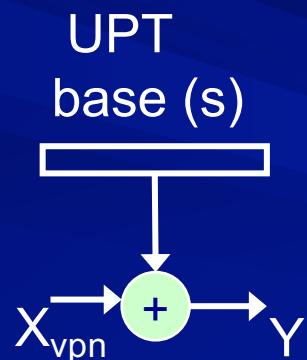
## physical space



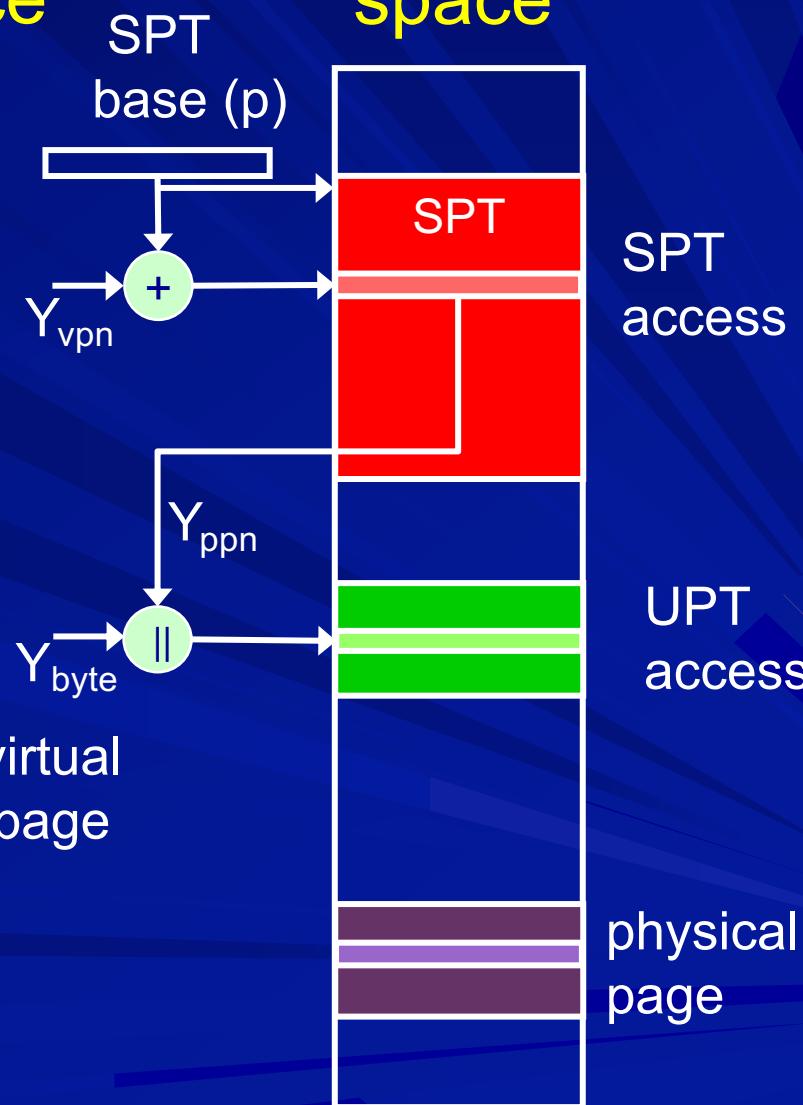
## user virtual space



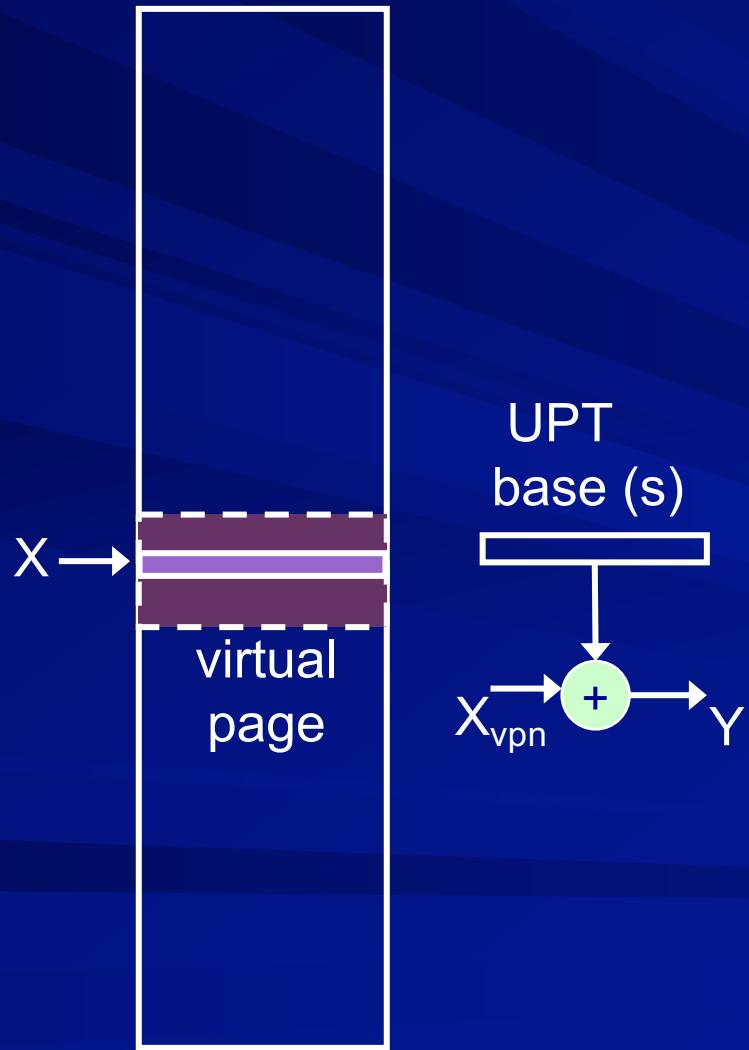
## system virtual space



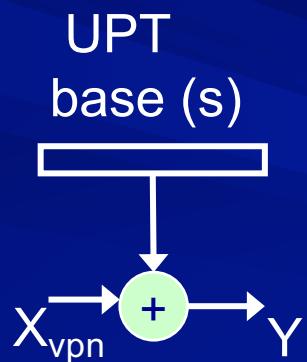
## physical space



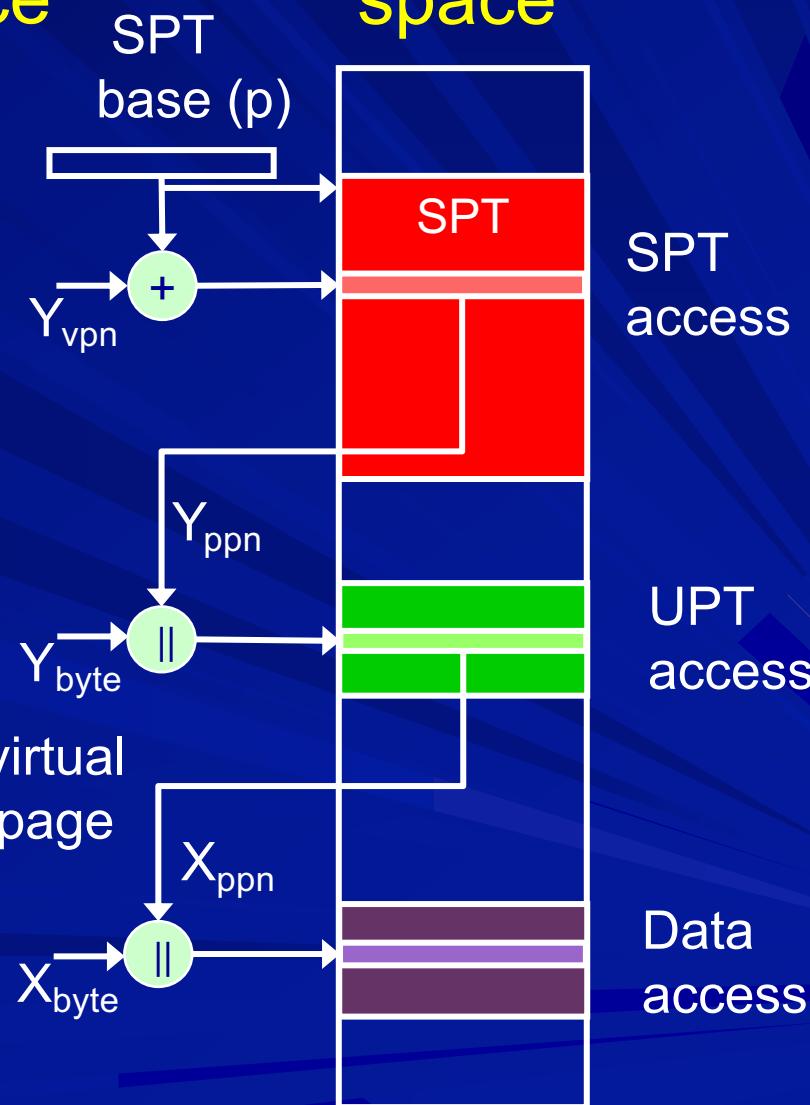
## user virtual space



## system virtual space



## physical space



SPT  
access

UPT  
access

Data  
access

# Memory protection with VM

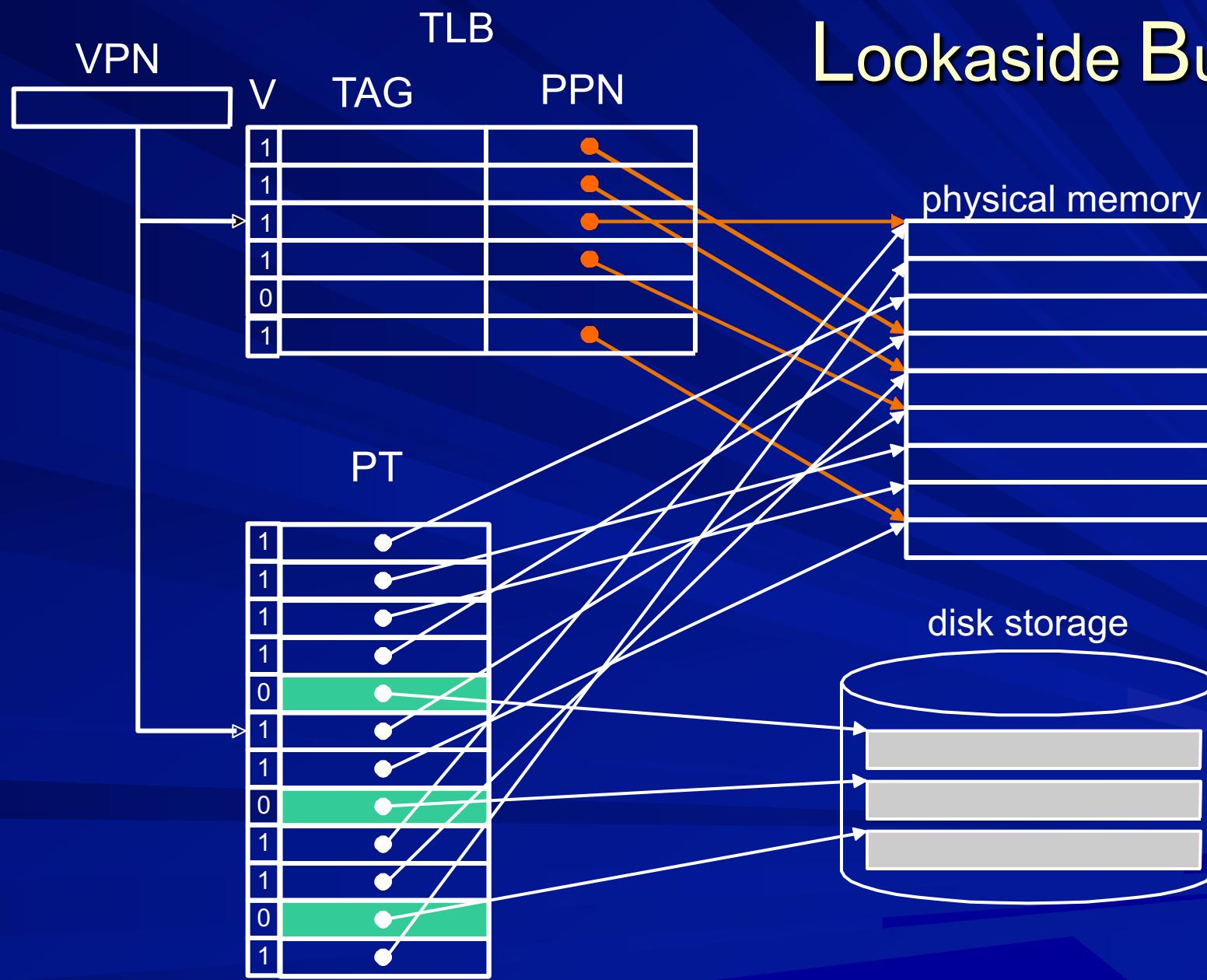
- A process can access only those physical pages which are pointed to by its page table
- Page tables are initialized by the OS and are updated automatically. User processes can't modify these.
- Programs run in user mode or privileged mode. Certain operations are possible in privileged mode only.

# Virtual memory hit time

- simple page table :  
     $2 \times$  physical memory access time
- 2 level page table :  
     $3 \times$  physical memory access time
- paged page table :  
     $3 \times$  physical memory access time

Can this be better??

# Translation Lookaside Buffer



# How is TLB Miss handled?

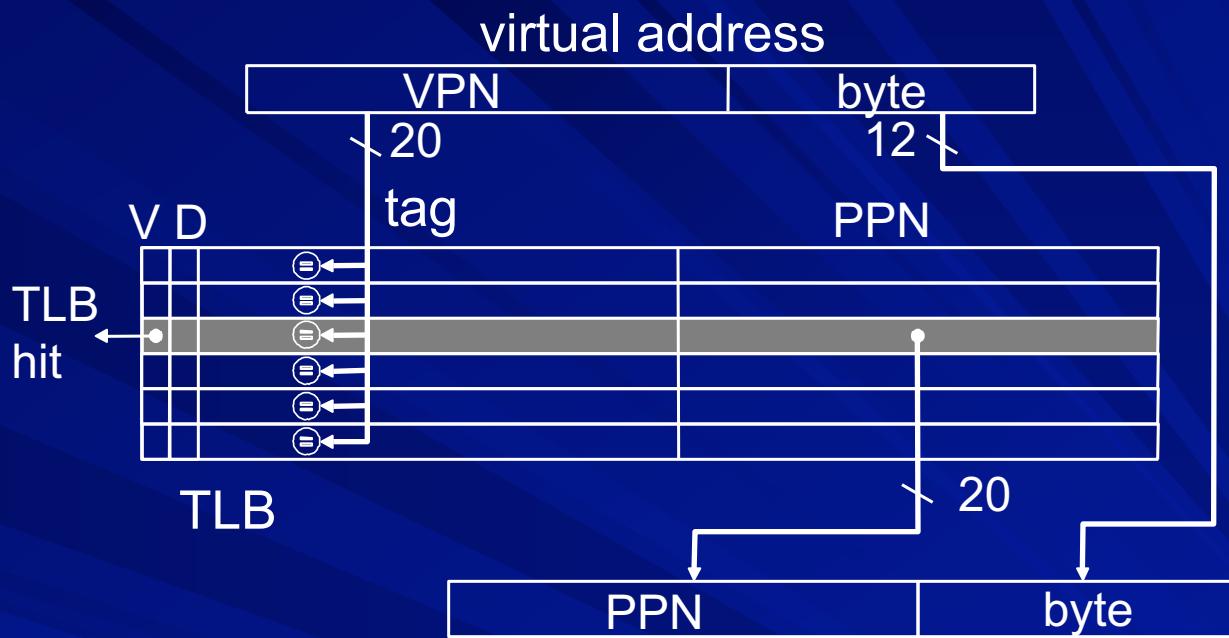
## ■ Hardware

- Memory Management Unit accesses Page Table
- Extra hardware, but efficient

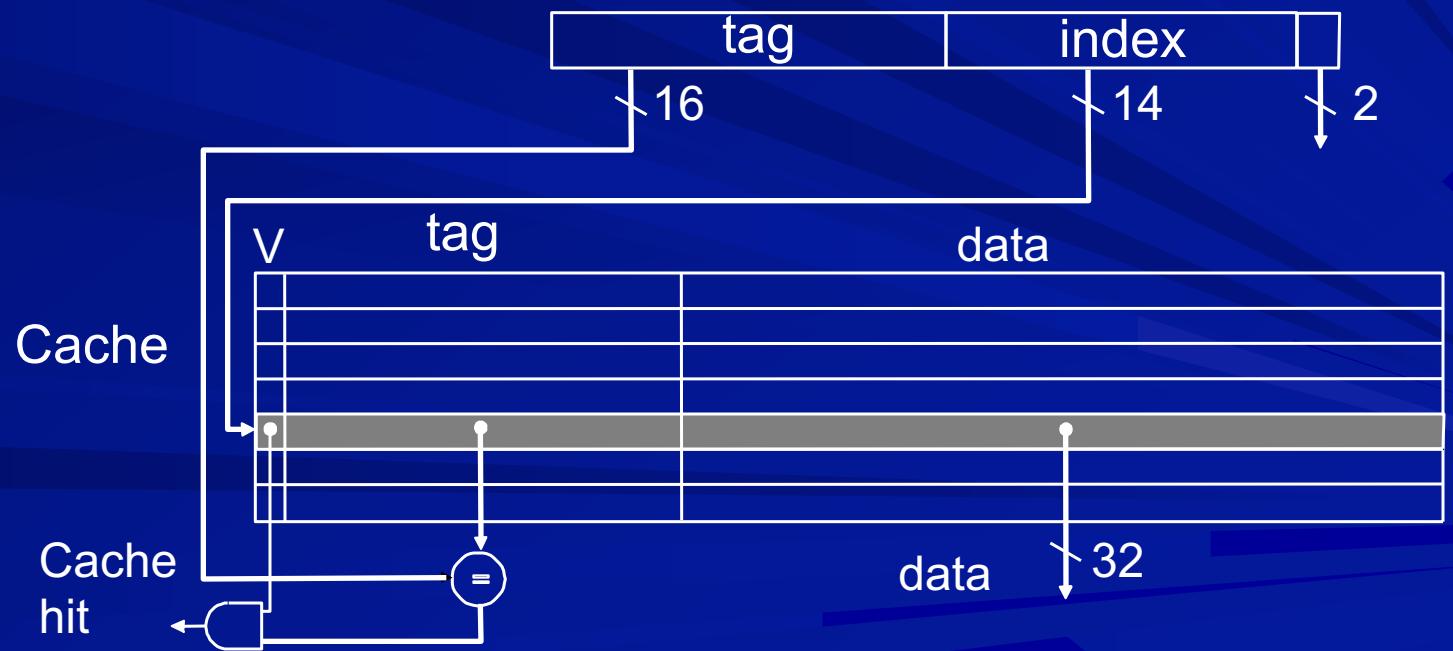
## ■ Software

- Exception is caused
- OS (exception handler) accesses Page Table
- Page Table structure defined by OS, not fixed by hardware

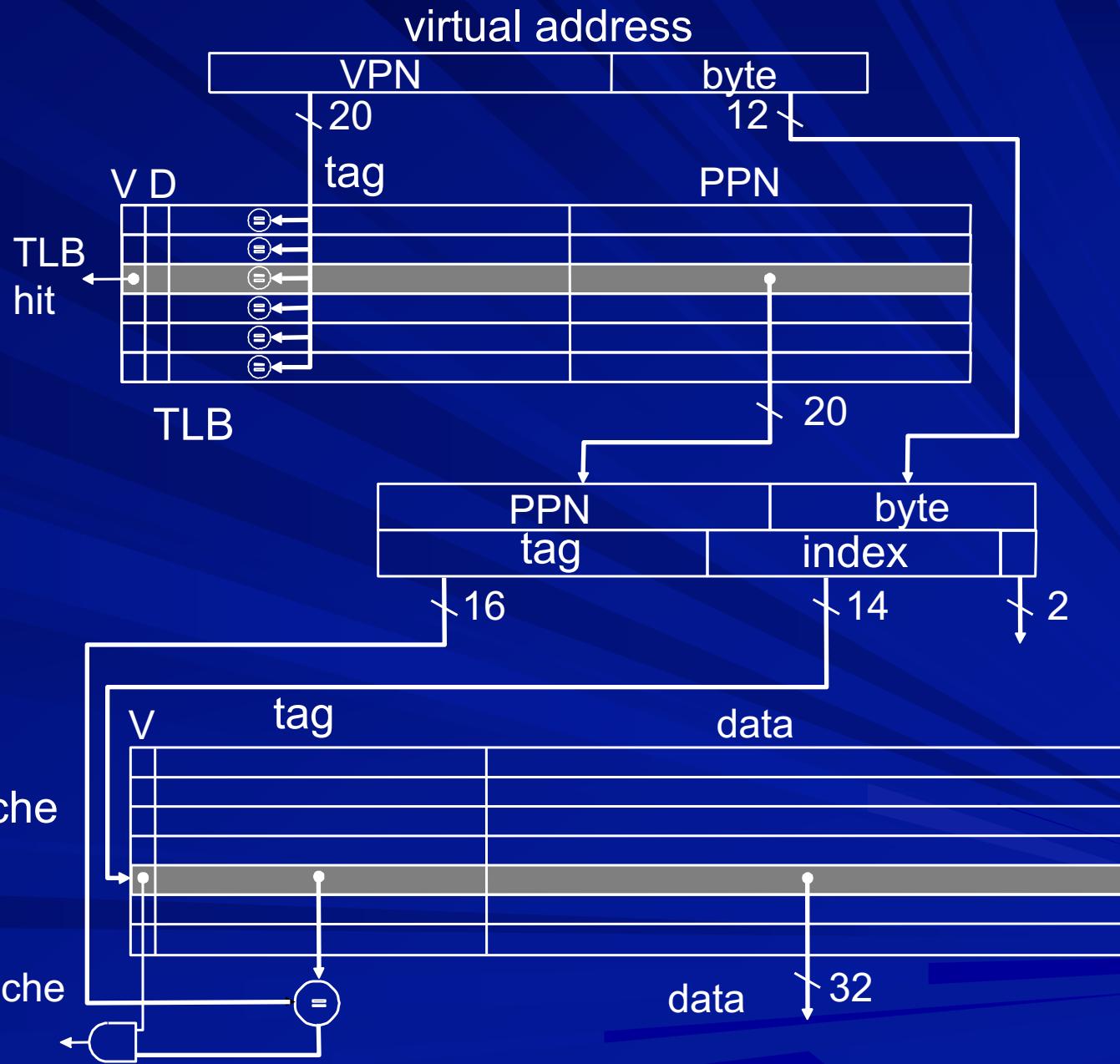
# TLB with Cache



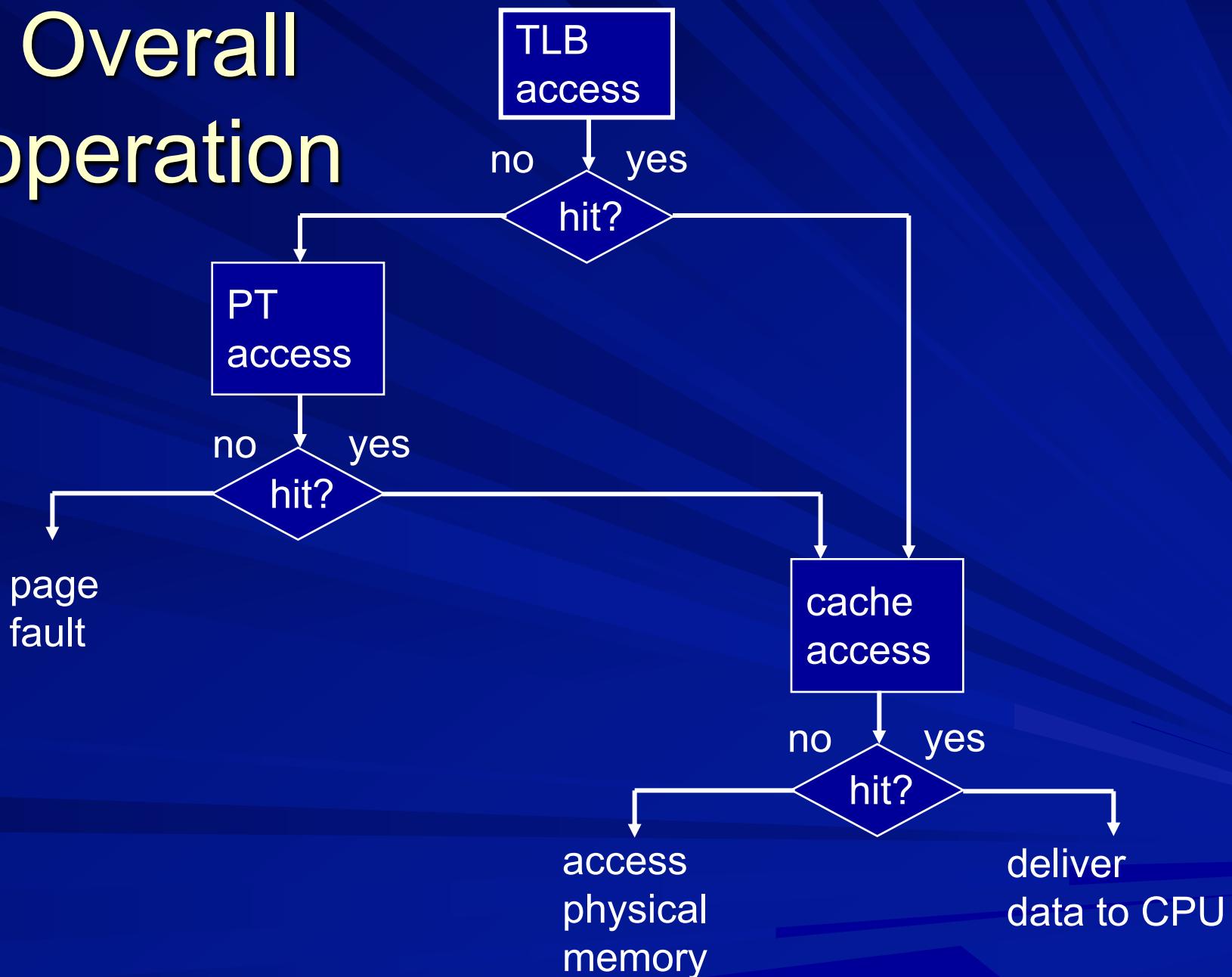
# TLB with Cache



# TLB with Cache



# Overall operation



# Virtually addressed cache

- Tags in the cache are from virtual addresses
- Cache access is made first
- TLB accessed only after cache miss
- Problem of aliasing - two copies of a shared object in physical memory
- Physically tagged and virtually indexed cache may be used

# Indexing before address translation

- Virtually indexed, physically tagged cache
  - simple and effective approach
  - indexing overlaps with address translation, tag matching after address translation
  - valid only if index field does not change during address translation (possible for caches that are not too large)

Virtual Page No.	Offset in page
Tag	Index

# COL216

# Computer Architecture

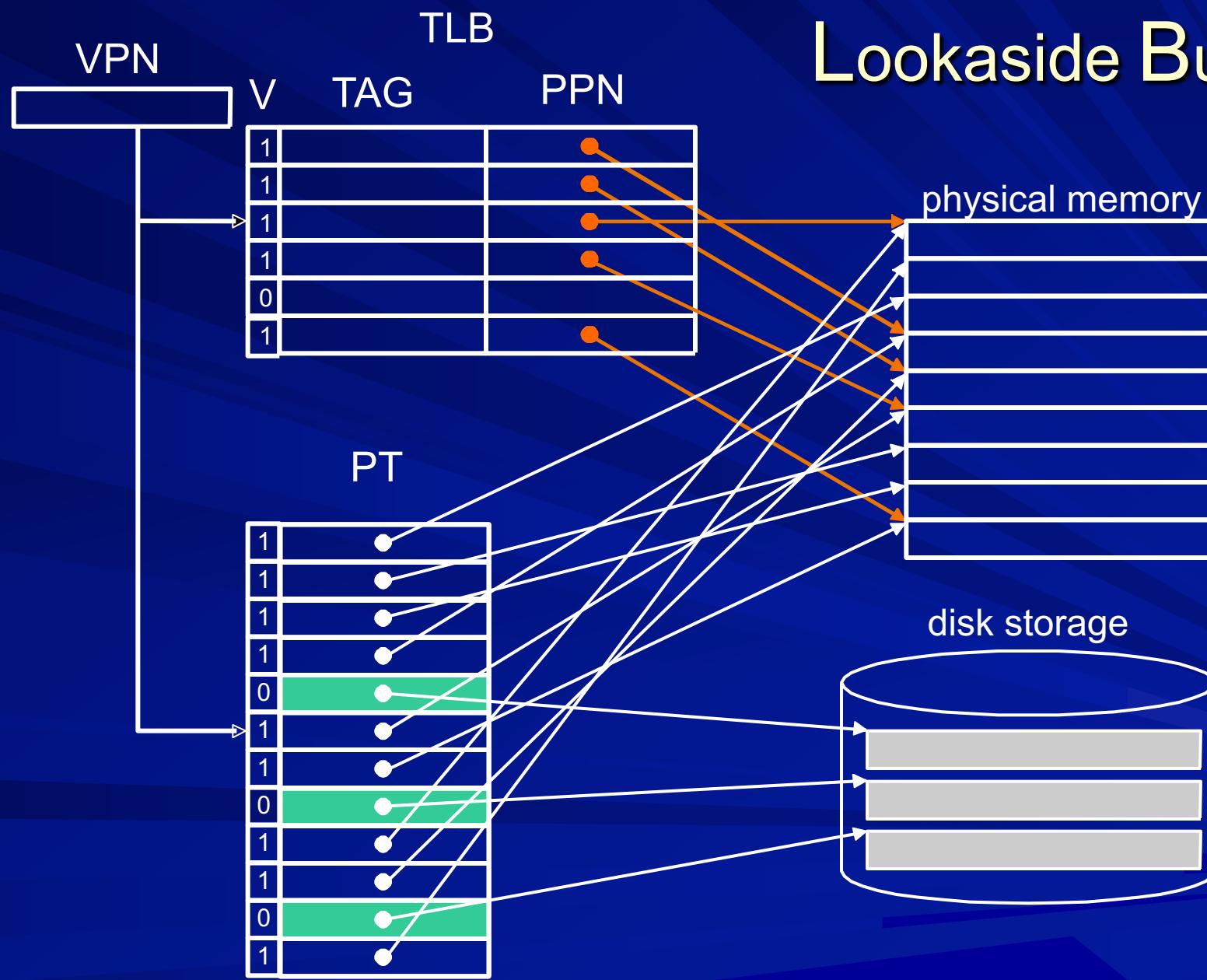
Virtual Memory  
TLB illustration  
14th March 2022

# Virtual memory hit time

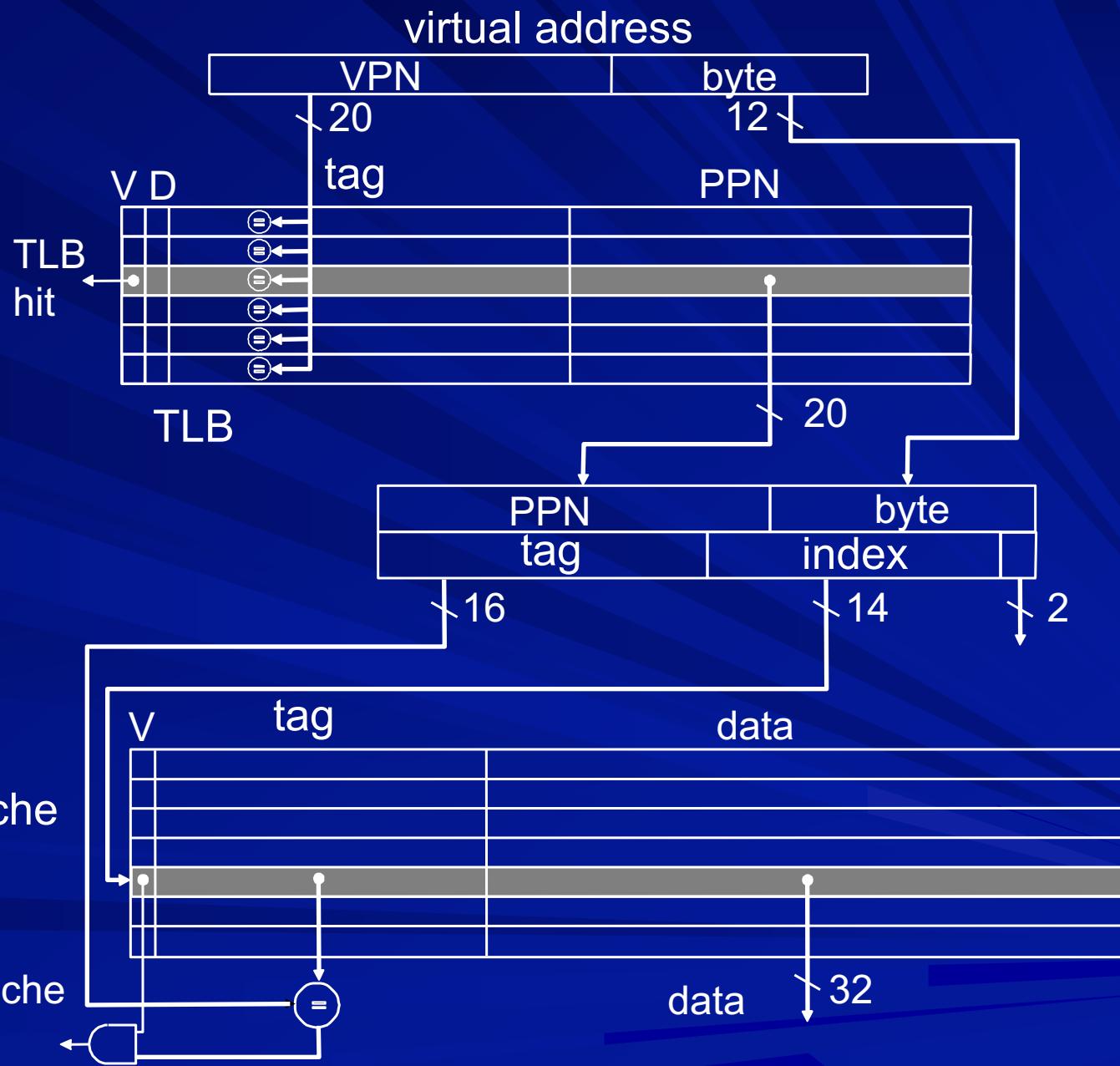
- simple page table :  
     $2 \times$  physical memory access time
- 2 level page table :  
     $3 \times$  physical memory access time
- paged page table :  
     $3 \times$  physical memory access time

Can this be better??

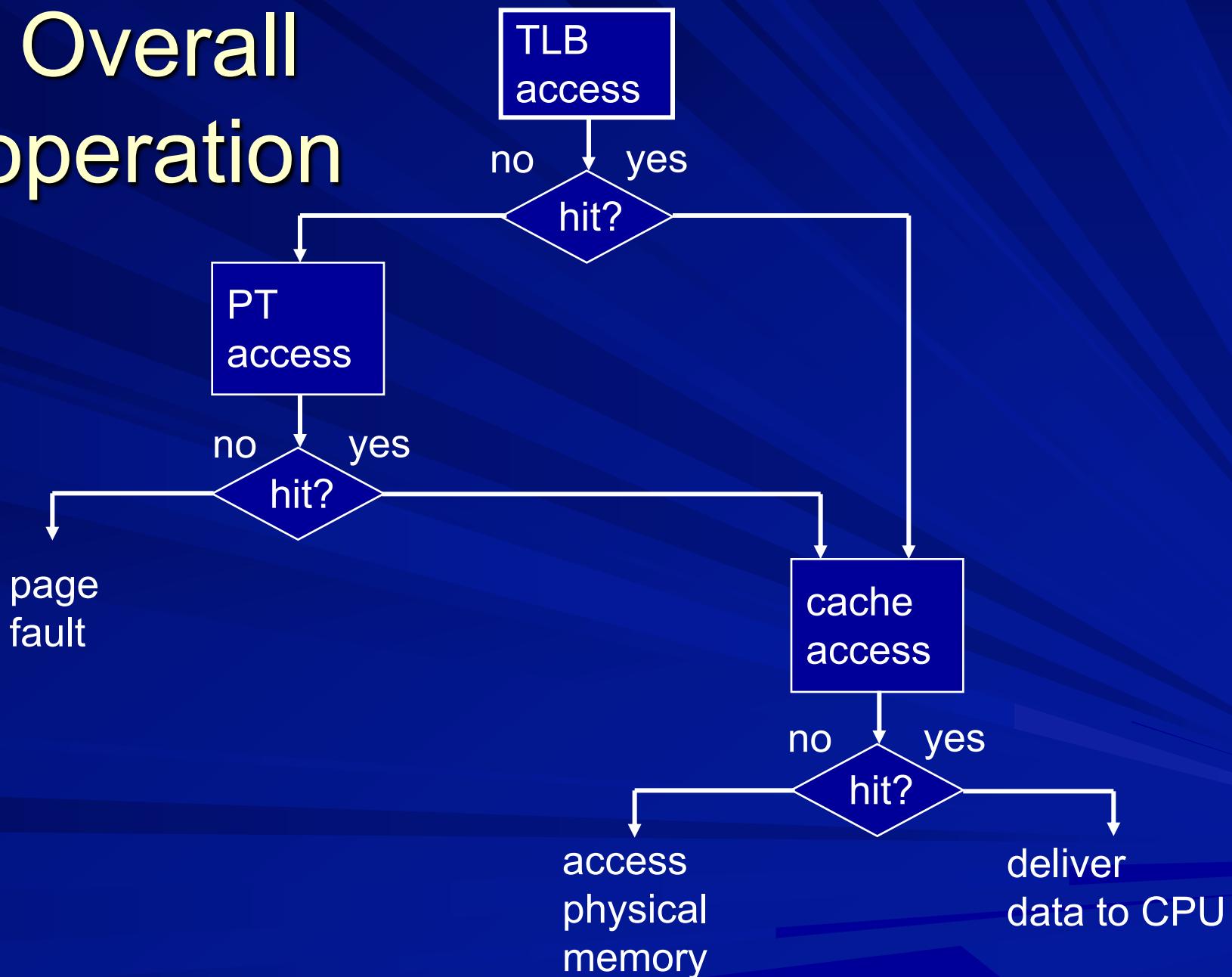
# Translation Lookaside Buffer



# TLB with Cache



# Overall operation



# Virtually addressed cache

- Tags in the cache are from virtual addresses
- Cache access is made first
- TLB accessed only after cache miss
- Problem of aliasing - two copies of a shared object in physical memory
- Physically tagged and virtually indexed cache may be used

# Indexing before address translation

- Virtually indexed, physically tagged cache
  - simple and effective approach
  - indexing overlaps with address translation, tag matching after address translation
  - valid only if index field does not change during address translation (possible for caches that are not too large)

Virtual Page No.	Offset in page
Tag	Index

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

	V	Tag(VPN)	PPN
	1	11	12
	1	7	4
	1	3	6
	0	4	9

fully associative

Page size = 4 KB

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

	V	Tag(VPN)	PPN
	1	11	12
	1	7	4
	1	3	6
	0	4	9

fully associative

Page size = 4 KB

Virtual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

	V	Tag(VPN)	PPN
	1	11	12
	1	7	4
	1	3	6
	0	4	9

fully associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

	V	Tag(VPN)	PPN
	1	11	12
	1	7	4
	1	3	6
	0	4	9

fully associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M

Page fault:

No

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	11	12
1	7	4
1	3	6
1	0	5

fully associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M

Page fault:

No

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	11	12
1	7	4
1	3	6
1	0	5

fully associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M      H      H      H

Page fault:

No

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	11	12
1	7	4
1	3	6
1	0	5

fully associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M      H      H      H      M

Page fault:

No

Yes

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	1	13
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	11	12
1	7	4
1	3	6
1	0	5

fully associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M      H      H      H      M

Page fault:

No

Yes

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	1	13
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	1	13
1	7	4
1	3	6
1	0	5

fully associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M      H      H      H      M

Page fault:

No

Yes

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	1	13
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	1	13
1	7	4
1	3	6
1	0	5

fully associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M      H      H      H      M      H      M

Page fault:

No

Yes

Yes

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	1	13
2	1	14
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

	V	Tag(VPN)	PPN
1	1	1	13
1	1	7	4
1	1	3	6
1	1	0	5

fully associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M      H      H      H      M      H      M

Page fault:

No

Yes

Yes

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	1	13
2	1	14
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	1	13
1	7	4
1	3	6
1	2	14

fully associative

vpn values are obtained by  
dividing virtual address stream  
by page size

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M      H      H      H      M      H      M

Page fault:

No

Yes

Yes

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

	V	Tag(VPN)	PPN
	1	11	12
	1	7	4
	1	10	3
	0	4	9

2-way set associative

Page size = 4 KB

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	11	12
1	7	4
1	10	3
0	4	9

2-way set associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	11	12
1	7	4
1	10	3
0	4	9

2-way set associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M

Page fault:

No

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	11	12
1	7	4
1	10	3
1	0	5

2-way set associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M

Page fault:

No

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	11	12
1	7	4
1	10	3
1	0	5

2-way set associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M      H      M

Page fault:

No      No

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	3	6
1	7	4
1	10	3
1	0	5

2-way set associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M      H      M

Page fault:

No      No

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	3	6
1	7	4
1	10	3
1	0	5

2-way set associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M      H      M      H      M

Page fault:

No      No

Yes

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	1	13
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	3	6
1	7	4
1	10	3
1	0	5

2-way set associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M      H      M      H      M

Page fault:

No      No

Yes

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	1	13
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	3	6
1	1	13
1	10	3
1	0	5

2-way set associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M      H      M      H      M

Page fault:

No      No

Yes

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	1	13
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	3	6
1	1	13
1	10	3
1	0	5

2-way set associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M      H      M      H      M      H      M

Page fault:

No

No

Yes

Yes

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	1	13
2	1	14
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	3	6
1	1	13
1	10	3
1	0	5

2-way set associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M      H      M      H      M      H      M

Page fault:

No

No

Yes

Yes

# Example: TLB & PT operation (Ex 5.10)

PT

	V	PPN
0	1	5
1	1	13
2	1	14
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

TLB

V	Tag(VPN)	PPN
1	3	6
1	1	13
1	2	14
1	0	5

2-way set associative

Page size = 4 KB

Vitual address stream:

4095, 31272, 15789, 15000, 7193, 4096, 8912

VPN:

0,      7,      3,      3,      1,      1,      2

TLB hit/miss:

M      H      M      H      M      H      M

Page fault:

No

No

Yes

Yes