

# Multi-Level Page Tables

## ■ Suppose:

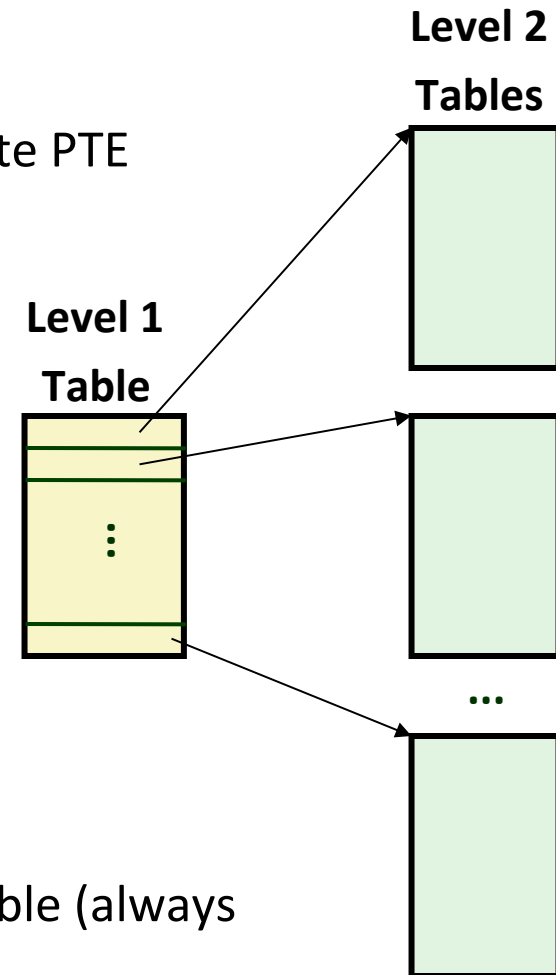
- 4KB ( $2^{12}$ ) page size, 48-bit address space, 4-byte PTE

## ■ Problem:

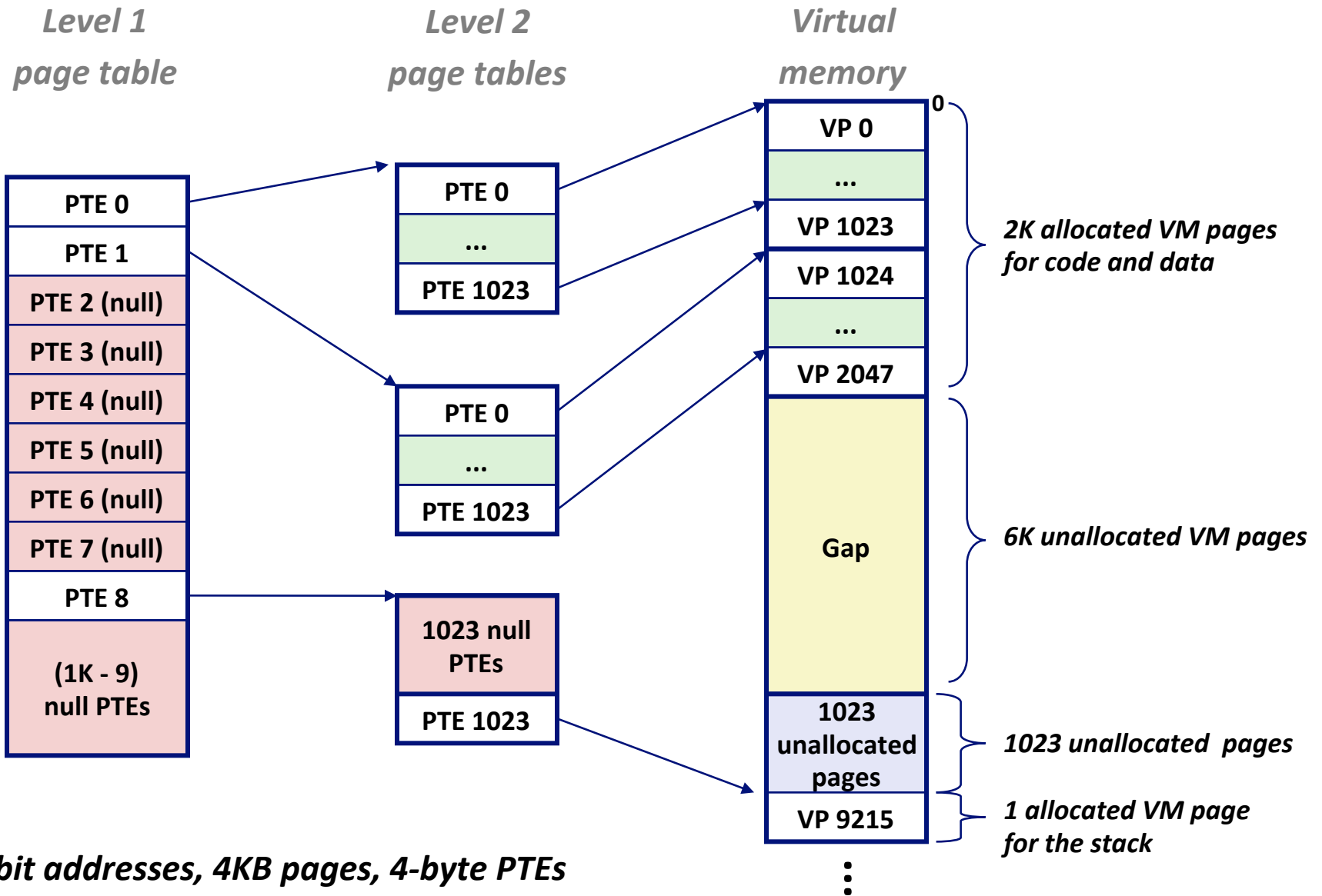
- Would need a 256 GB page table!
  - $2^{48} * 2^{-12} * 2^2 = 2^{38}$  bytes

## ■ Common solution:

- Multi-level page tables
- Example: 2-level page table
  - Level 1 table: each PTE points to a page table (always memory resident)
  - Level 2 table: each PTE points to a page (paged in and out like any other data)



# A Two-Level Page Table Hierarchy



# Review of Symbols

## ■ Basic Parameters

- $N = 2^n$  : Number of addresses in virtual address space
- $M = 2^m$  : Number of addresses in physical address space
- $P = 2^p$  : Page size (bytes)

## ■ Components of the virtual address (VA)

- TLBI: TLB index
- TLBT: TLB tag
- VPO: Virtual page offset
- VPN: Virtual page number

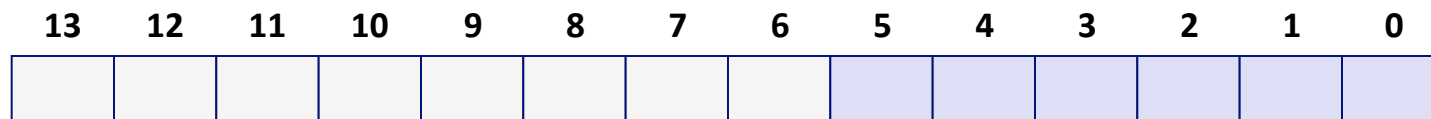
## ■ Components of the physical address (PA)

- PPO: Physical page offset (same as VPO)
- PPN: Physical page number
- CO: Byte offset within cache line
- CI: Cache index
- CT: Cache tag

# Simple Memory System Example

## ■ Addressing

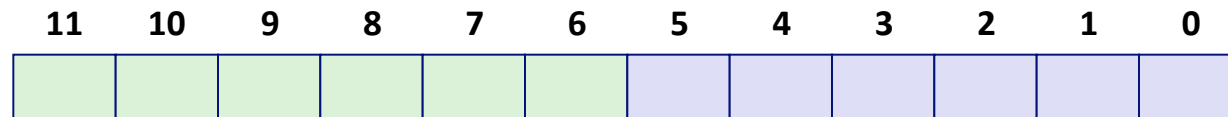
- 14-bit virtual addresses
- 12-bit physical address
- Page size = 64 bytes



Virtual Page Number

Virtual Page Offset

Because  
of the  
page size  
 $2^6 = 64$



Physical Page Number

Physical Page Offset

# Simple Memory System Page Table

Portion of the page table.

<i>VPN</i>	<i>PPN</i>	<i>Valid</i>
<b>00</b>	<b>28</b>	<b>1</b>
<b>01</b>	<b>–</b>	<b>0</b>
<b>02</b>	<b>33</b>	<b>1</b>
<b>03</b>	<b>02</b>	<b>1</b>
<b>04</b>	<b>–</b>	<b>0</b>
<b>05</b>	<b>16</b>	<b>1</b>
<b>06</b>	<b>–</b>	<b>0</b>
<b>07</b>	<b>–</b>	<b>0</b>

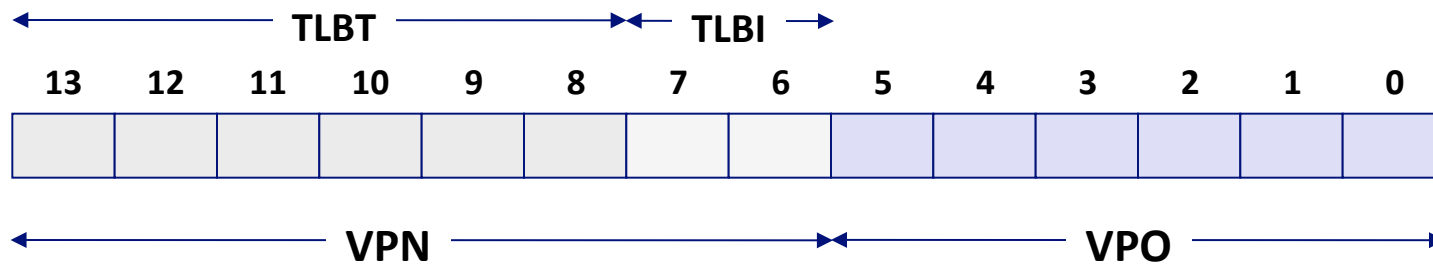
<i>VPN</i>	<i>PPN</i>	<i>Valid</i>
<b>08</b>	<b>13</b>	<b>1</b>
<b>09</b>	<b>17</b>	<b>1</b>
<b>0A</b>	<b>09</b>	<b>1</b>
<b>0B</b>	<b>–</b>	<b>0</b>
<b>0C</b>	<b>–</b>	<b>0</b>
<b>0D</b>	<b>2D</b>	<b>1</b>
<b>0E</b>	<b>11</b>	<b>1</b>
<b>0F</b>	<b>0D</b>	<b>1</b>

# Simple Memory System TLB

- 16 entries
- 4-way associative

Distinguishes different VPNs that map to same TLB set

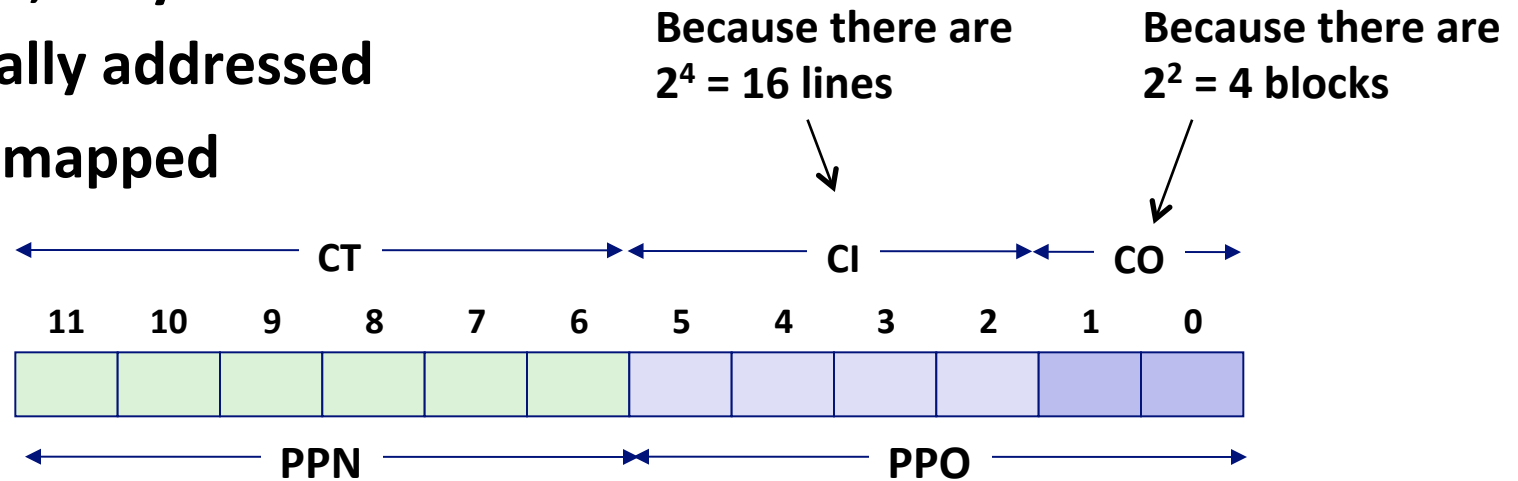
Because the TLB is 4-way set assoc.



Set	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid
0	03	-	0	09	0D	1	00	-	0	07	02	1
1	03	2D	1	02	-	0	04	-	0	0A	-	0
2	02	-	0	08	-	0	06	-	0	03	-	0
3	07	-	0	03	0D	1	0A	34	1	02	-	0

# Simple Memory System Cache

- 16 lines, 4-byte block size
- Physically addressed
- Direct mapped



<i>Idx</i>	<i>Tag</i>	<i>Valid</i>	<i>B0</i>	<i>B1</i>	<i>B2</i>	<i>B3</i>
0	19	1	99	11	23	11
1	15	0	-	-	-	-
2	1B	1	00	02	04	08
3	36	0	-	-	-	-
4	32	1	43	6D	8F	09
5	0D	1	36	72	F0	1D
6	31	0	-	-	-	-
7	16	1	11	C2	DF	03

<i>Idx</i>	<i>Tag</i>	<i>Valid</i>	<i>B0</i>	<i>B1</i>	<i>B2</i>	<i>B3</i>
8	24	1	3A	00	51	89
9	2D	0	-	-	-	-
A	2D	1	93	15	DA	3B
B	0B	0	-	-	-	-
C	12	0	-	-	-	-
D	16	1	04	96	34	15
E	13	1	83	77	1B	D3
F	14	0	-	-	-	-

Page Table

<i>VPN</i>	<i>PPN</i>	<i>Valid</i>
00	28	1
01	-	0
02	33	1
03	02	1
04	-	0
05	16	1
06	-	0
07	-	0

<i>VPN</i>	<i>PPN</i>	<i>Valid</i>
08	13	1
09	17	1
0A	09	1
0B	-	0
0C	-	0
0D	2D	1
0E	11	1
0F	0D	1

TLB

<i>Set</i>	<i>Tag</i>	<i>PPN</i>	<i>Valid</i>	<i>Tag</i>	<i>PPN</i>	<i>Valid</i>	<i>Tag</i>	<i>PPN</i>	<i>Valid</i>	<i>Tag</i>	<i>PPN</i>	<i>Valid</i>
0	03	-	0	09	0D	1	00	-	0	07	02	1
1	03	2D	1	02	-	0	04	-	0	0A	-	0
2	02	-	0	08	-	0	06	-	0	03	-	0
3	07	-	0	03	0D	1	0A	34	1	02	-	0

Cache

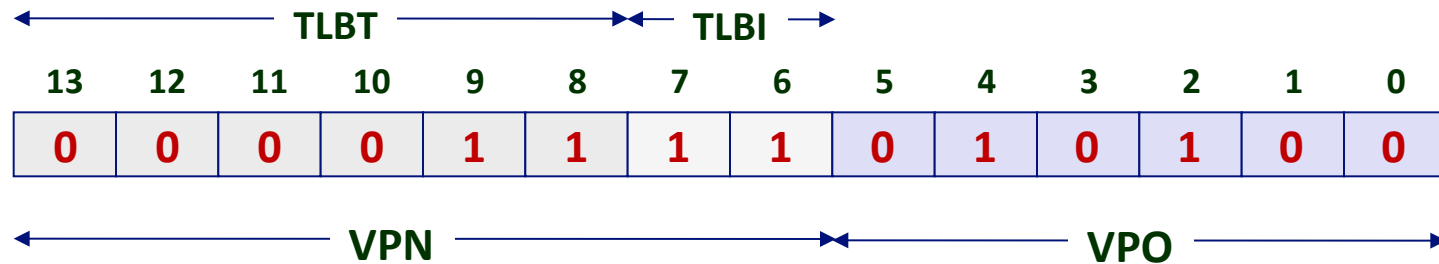
<i>Idx</i>	<i>Tag</i>	<i>Valid</i>	<i>B0</i>	<i>B1</i>	<i>B2</i>	<i>B3</i>
0	19	1	99	11	23	11
1	15	0	-	-	-	-
2	1B	1	00	02	04	08
3	36	0	-	-	-	-
4	32	1	43	6D	8F	09
5	0D	1	36	72	F0	1D
6	31	0	-	-	-	-
7	16	1	11	C2	DF	03

<i>Idx</i>	<i>Tag</i>	<i>Valid</i>	<i>B0</i>	<i>B1</i>	<i>B2</i>	<i>B3</i>
8	24	1	3A	00	51	89
9	2D	0	-	-	-	-
A	2D	1	93	15	DA	3B
B	0B	0	-	-	-	-
C	12	0	-	-	-	-
D	16	1	04	96	34	15
E	13	1	83	77	1B	D3
F	14	0	-	-	-	-



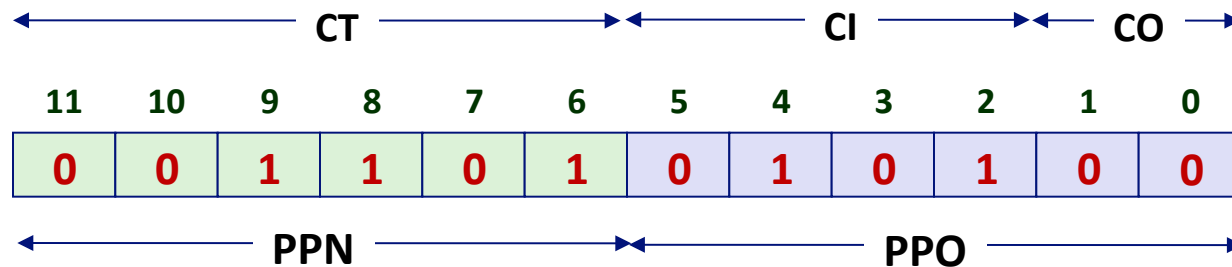
# Address Translation Example #1

Virtual Address: 0x03D4 (0000001111010100)



VPN 0x0F TLBI 0x3 TLBT 0x03 TLB Hit? Y Page Fault? N PPN: 0x0D

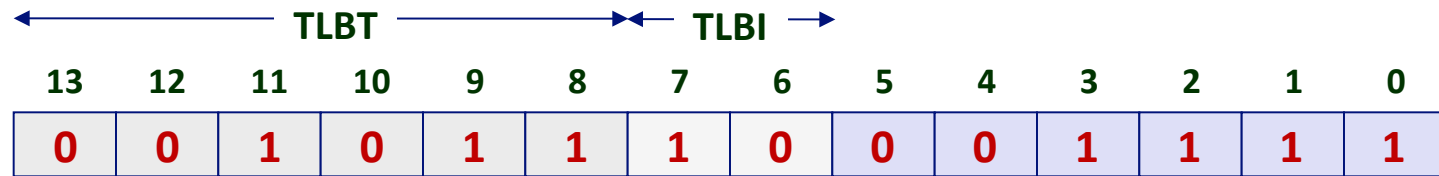
## Physical Address



CO 0 CI 0x5 CT 0x0D Hit? Y Byte: 0x36

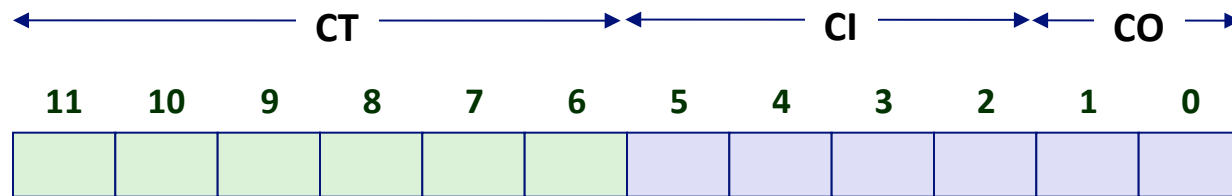
# Address Translation Example #2

Virtual Address: 0x0B8F



VPN 0x2E    TLBI 2    TLBT 0x0B    TLB Hit? N    Page Fault? Y    PPN: TBD

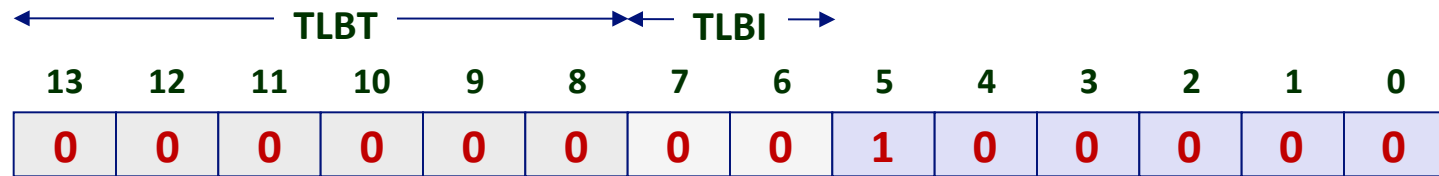
## Physical Address



CO \_\_\_    CI \_\_\_    CT \_\_\_    Hit? \_\_\_    Byte: \_\_\_

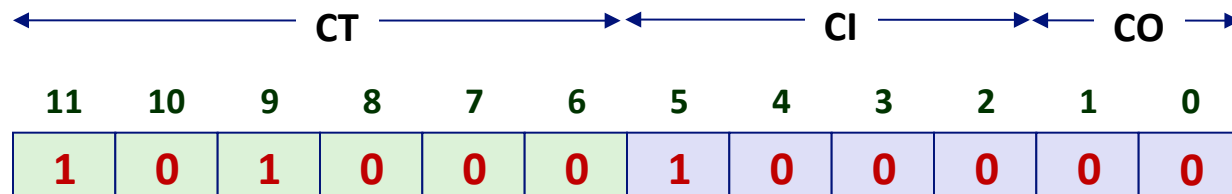
# Address Translation Example #3

Virtual Address: 0x0020



VPN 0x00    TLBI 0    TLBT 0x00    TLB Hit? N    Page Fault? N    PPN: 0x28

## Physical Address

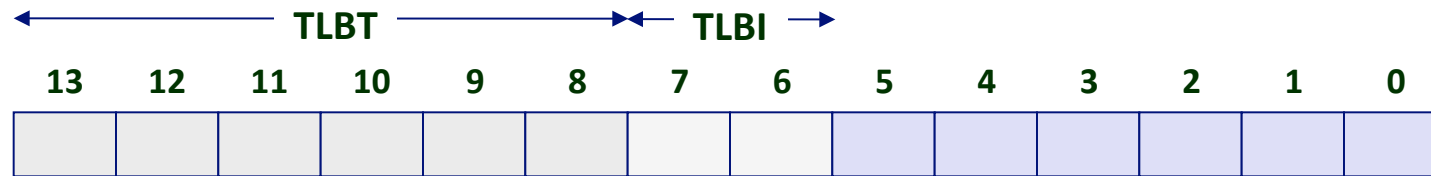


CO 0    CI 0x8    CT 0x28    Hit? N    Byte: Mem

# Quiz 33

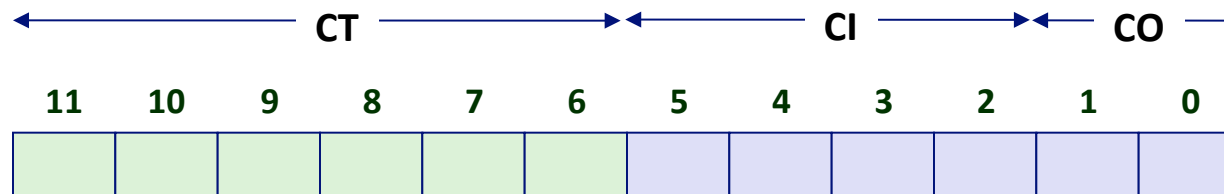
Name \_\_\_\_\_ Mailbox \_\_\_\_\_

Virtual Address: 0x0394



VPN \_\_\_\_ TLBI \_\_\_\_ TLBT \_\_\_\_ TLB Hit? \_\_\_\_ Page Fault? \_\_\_\_ PPN: \_\_\_\_

## Physical Address



CO \_\_\_\_ CI \_\_\_\_ CT \_\_\_\_ Hit? \_\_\_\_ Byte: \_\_\_\_