CS-200
Computer Architecture
—
Part 3d. Memory Hierarchy
Simple Virtual Memory Examples

Paolo Ienne
<paolo.ienne@epfl.ch>

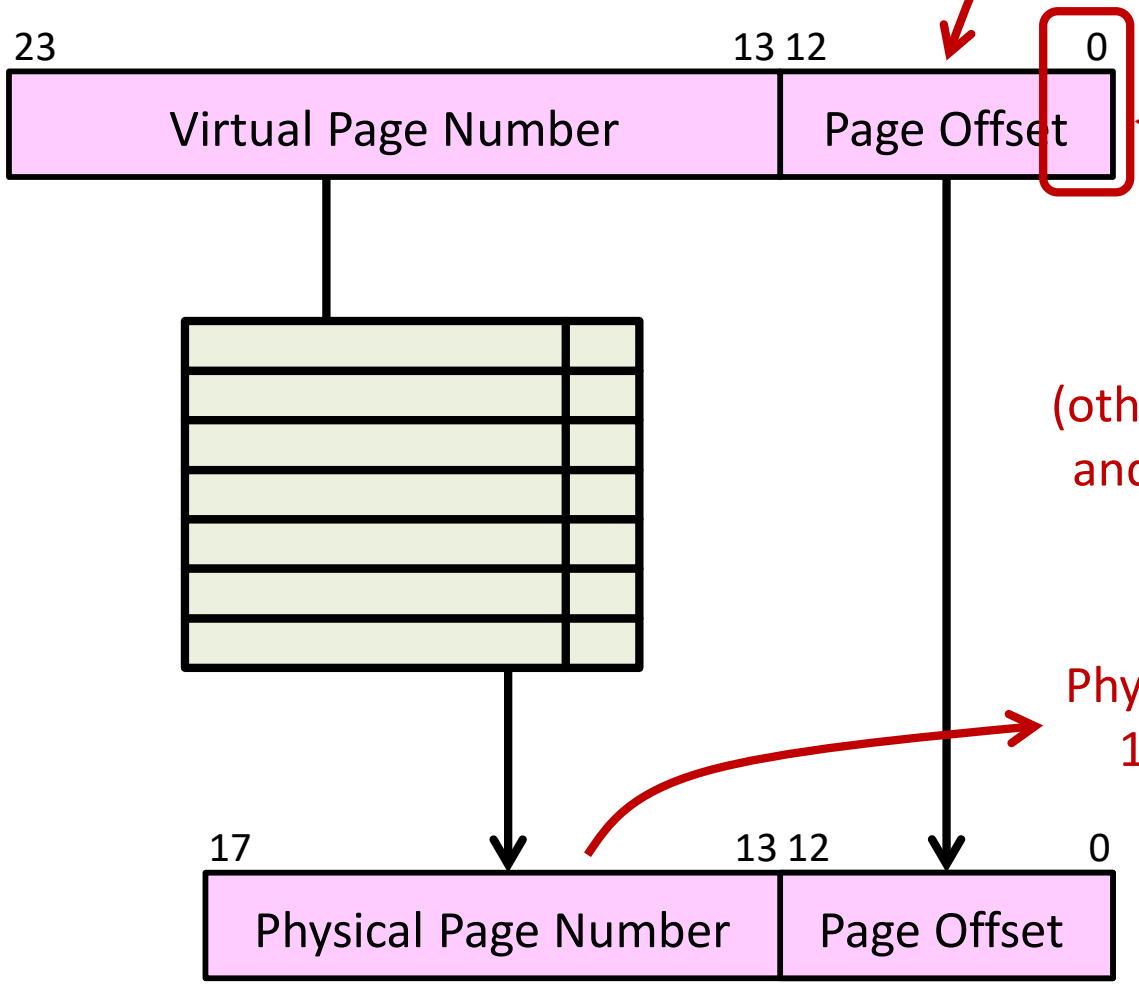
1. Simple Translation Scheme

- Consider a **byte-addressable** virtual memory system that uses linear page tables with **8-KiB pages**, **24-bit virtual addresses**, **18-bit physical addresses**, and **3 control bits** per page table entry.
- One word is **4 bytes**.
- What is the width of the physical page number field, in bits?

Solution

Page size =
8 KiB = 2^{13} B

Virtual Address



Byte-addressable
→ page size in B
(otherwise page size 2^{11} W
and page offset = 11 bits)

Physical Page Number =
18 – 13 bits = **5 bits**

Physical Address

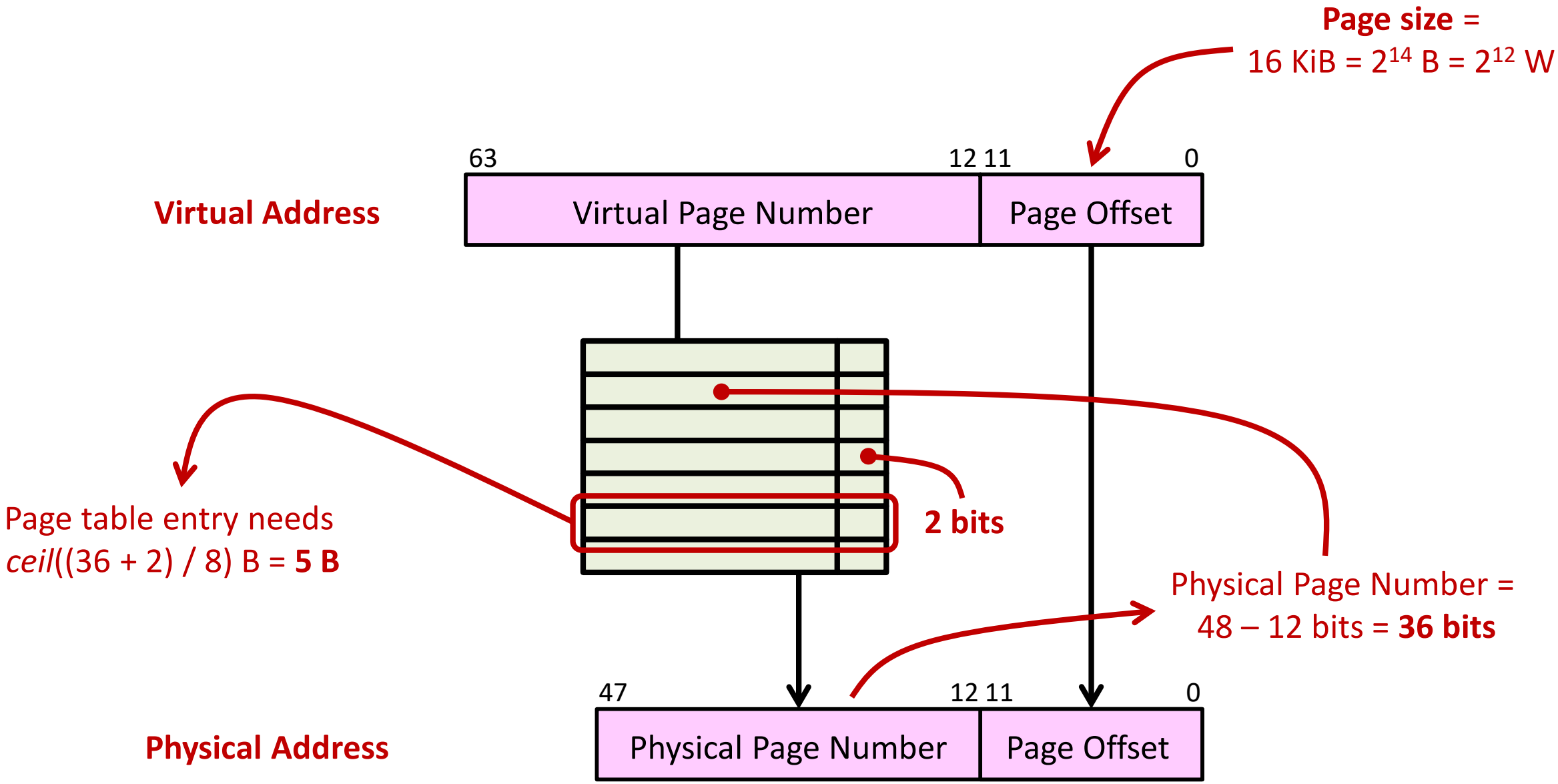
The word size is irrelevant

The number of bytes per page table entry is irrelevant

2. Page Table Entry Size

- Consider a **word-addressable** virtual memory system that uses linear page tables with **16-KiB pages**, **64-bit virtual addresses**, **48-bit physical addresses**, and **2 control bits** per page table entry.
- One word is **4 bytes**.
- Page table entries are **byte-aligned**.
- What is the corresponding minimum size of each page table entry, in bytes?

Solution

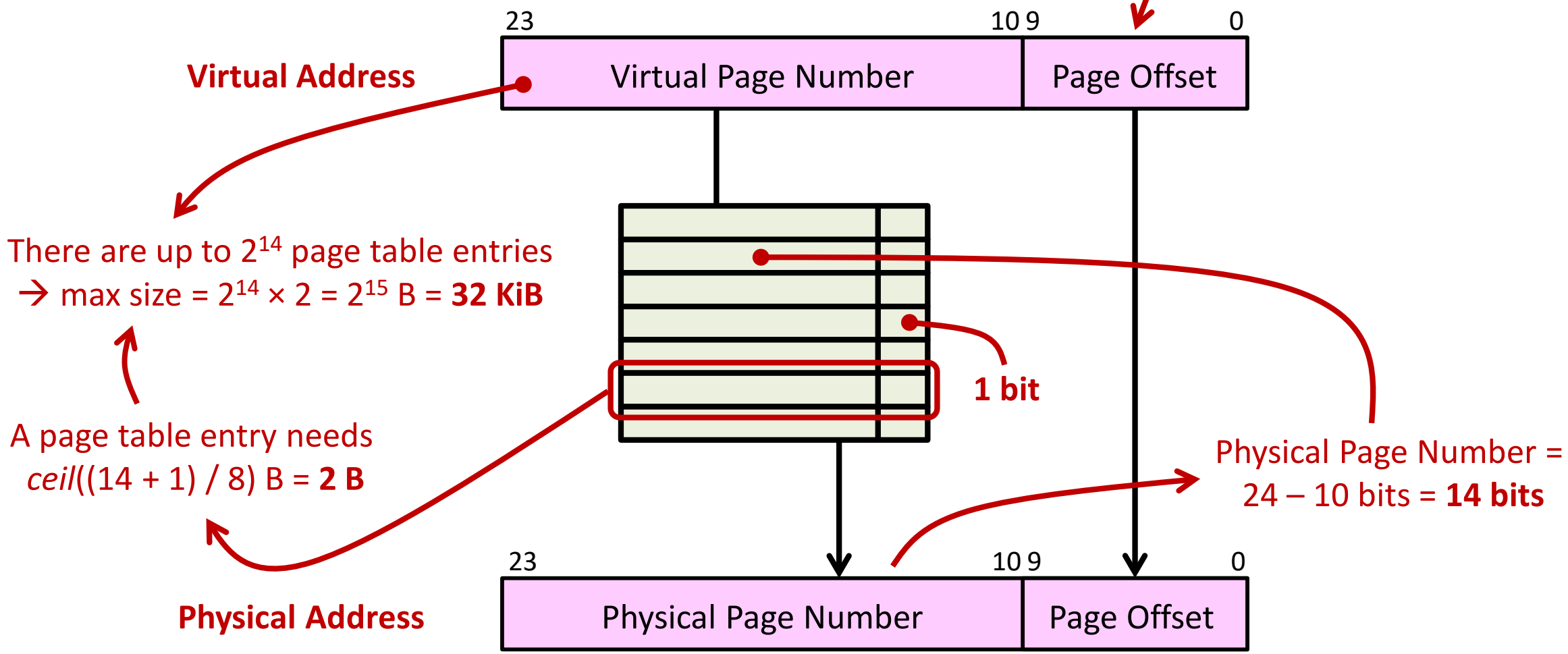

Page size =
16 KiB = 2^14 B = 2^12 W
Virtual Address
63
12 11
0
Virtual Page Number
Page Offset
Page table entry needs
ceil((36 + 2) / 8) B = 5 B
2 bits
Physical Page Number =
48 – 12 bits = 36 bits
47
12 11
0
Physical Address
Physical Page Number
Page Offset

3. Total Page Table Size

- Consider a **word-addressable** virtual memory system that uses linear page tables with **4-KiB pages**, **24-bit virtual addresses**, **24-bit physical addresses**, and **1 control bit** per page table entry.
- One word is **4 bytes**.
- Page table entries are **byte-aligned**.
- Assuming that the page table contains all possible translations, what is going to be its total size?

Solution

Page size =
4 KiB = 2^{12} B = 2^{10} W



4. Total Addressable Physical Memory

- Consider a **word-addressable** virtual memory system that uses linear page tables with **4-KiB pages**.
- The **physical page number** is encoded on **19 bits**.
- One word is **2 bytes**.
- What is the total size in words of the addressable physical memory?

Solution

- There can be up to 2^{19} physical pages of 4 KiB.
- Total addressable memory = $2^{19} \times 2^{12}$ B = 2 MiB = **1 MiW**

5. Address Translation

- Consider a **byte-addressable** virtual memory system that uses linear page tables with **8-KiB pages**, **32-bit virtual addresses**, **32-bit physical addresses**.
- The table to the right contains the first 16 elements of a **linear page table**.
- The **Valid** bit indicates that the page is allocated and in memory.
- What is the translation of `0x19A60`?
- And of `0x14C48`?

Index	Physical Page	Valid
0	0x6235	TRUE
1	0x22BB4	FALSE
2	0x2DE8	FALSE
3	0x34120	FALSE
4	0x1BE42	FALSE
5	0x2D5FF	FALSE
6	0xCC56	TRUE
7	0x6C7B	TRUE
8	0x2ABA	TRUE
9	0xFDFB	TRUE
10	0x3990B	FALSE
11	0x1AB4F	TRUE
12	0x8F0D	TRUE
13	0x1ACE	TRUE
14	0x3465B	FALSE
15	0xB586	FALSE

Solution

Index	Physical Page	Valid
…	…	…
10	0x3990B	**FALSE**
11	0x1AB4F	TRUE
12	**0x8F0D**	TRUE
…	…	…

Page size = 8 KiB = 2^{13} B

offset = 13 bits

0x00019A60 = 0b **0000 0000 0000 0001 100**1 1010 0110 0000

virtual page **#12** → physical page **0x8F0D**

0b **1000 1111 0000 1101** 1 1010 0110 0000

0b 000**1 0001 1110 0001 101**1 1010 0110 0000 = **0x11E1BA60**

offset = 13 bits

0x00014C48 = 0b **0000 0000 0000 0001 010**0 1100 0010 1000

virtual page **#10** → **Invalid!**