CMPE 200
Computer Architecture & Design

Lecture 4.
Memory Hierarchy (7)

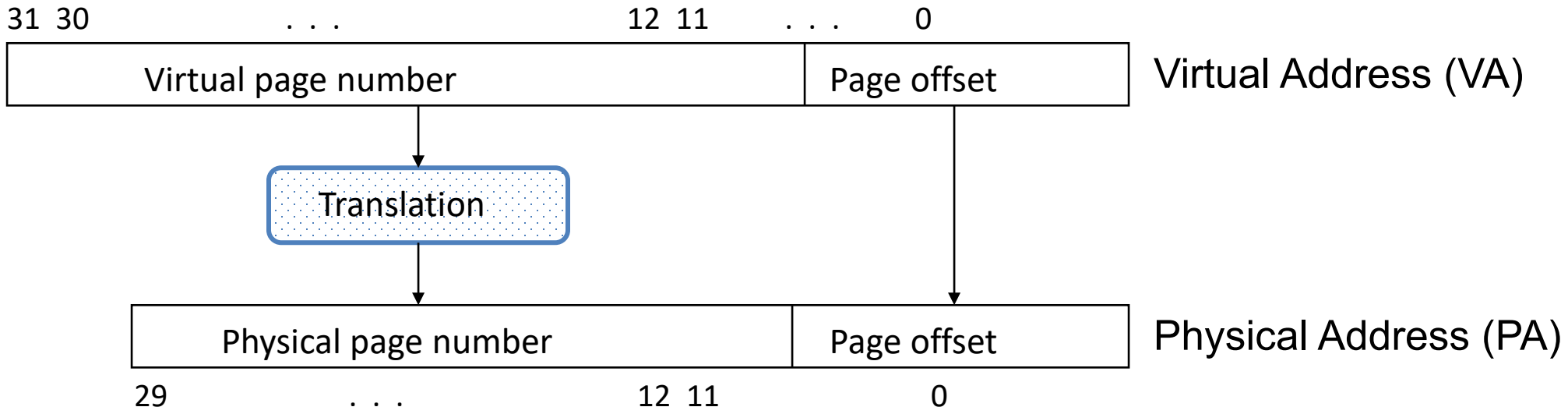
Haonan Wang

SAN JOSÉ STATE
UNIVERSITY

How is the Hierarchy Managed?

- **registers ↔ memory**
  - by compiler (or programmer)
- **registers ↔ cache ↔ main memory**
  - by the cache & memory controller hardware
- **main memory ↔ external storage (flash, disk)**
  - Static: by the programmer with OS support (files)
  - Dynamic: by the operating system (virtual memory)
    - virtual address to physical address mapping
    - assisted by the hardware (TLB, page tables)

Virtual Memory

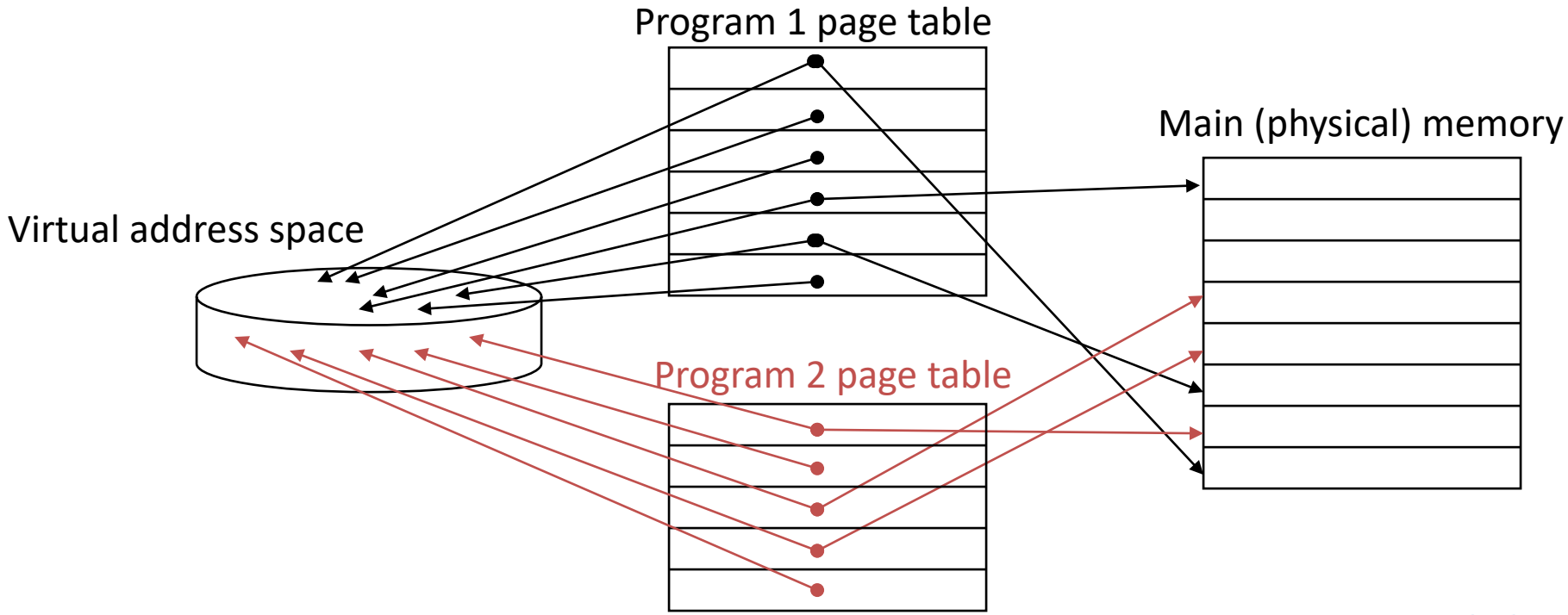


- **A virtual address must first be translated to a physical address to access memory**
  - Basic unit: page (e.g., 1KB to 64KB)

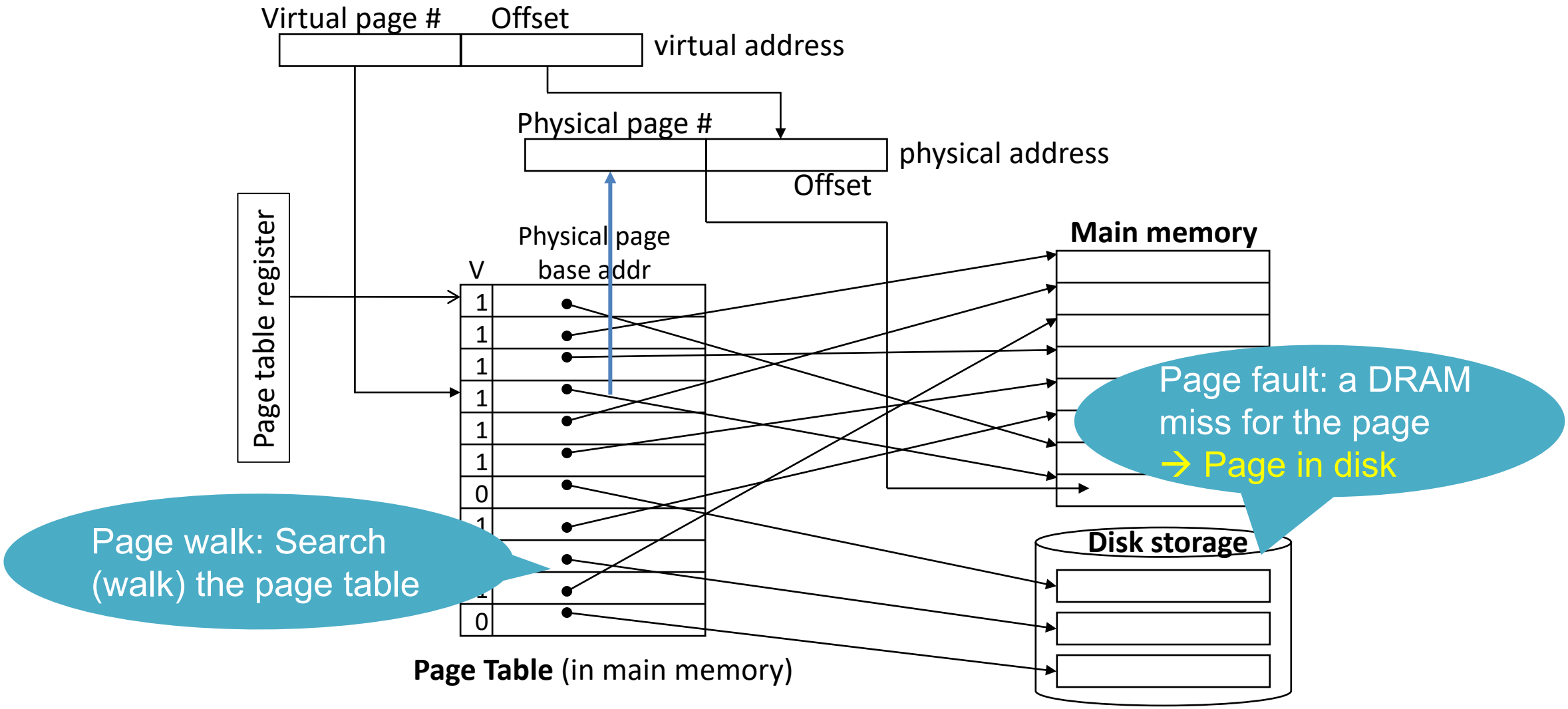
- **Virtual memory size can be larger than physical memory size**
  - Pages can be stored in the secondary storage

Address Space Isolation

- **Allows efficient and safe sharing of main memory among multiple processes**
  - The starting location of each page is contained in the program's page table
  - Create the illusion that each program has a large consecutive memory space
  - Improving memory utilization: code can be loaded anywhere the OS can find space

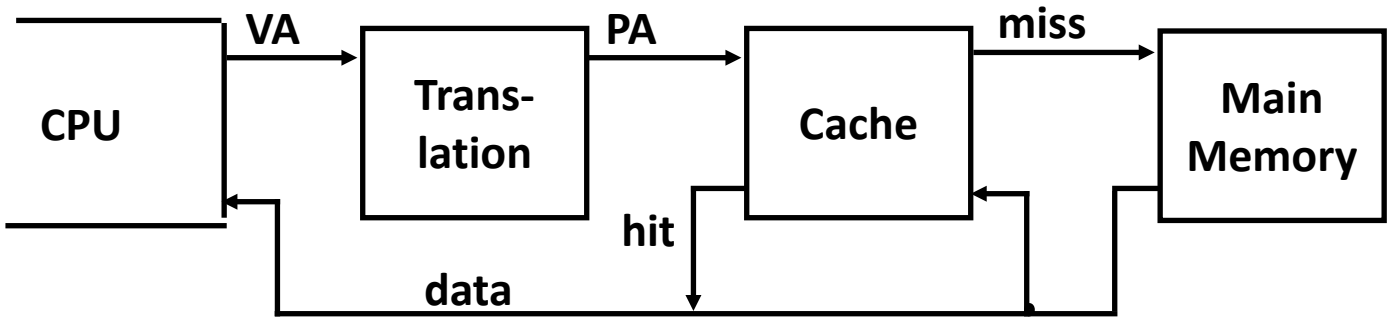


Address Translation Mechanisms



Virtual Addressing with a Cache

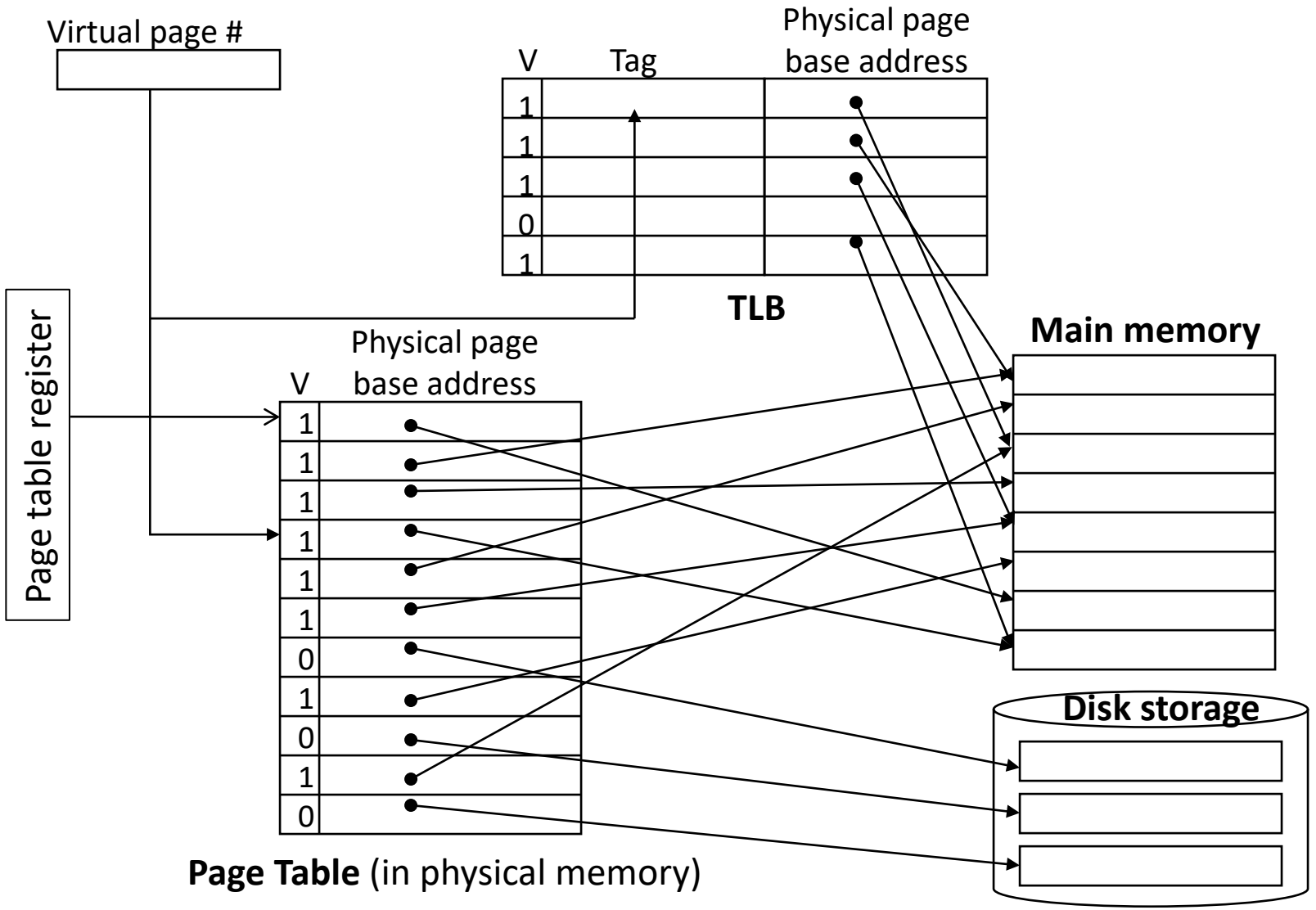
- **It takes an extra memory access to translate a VA to a PA**



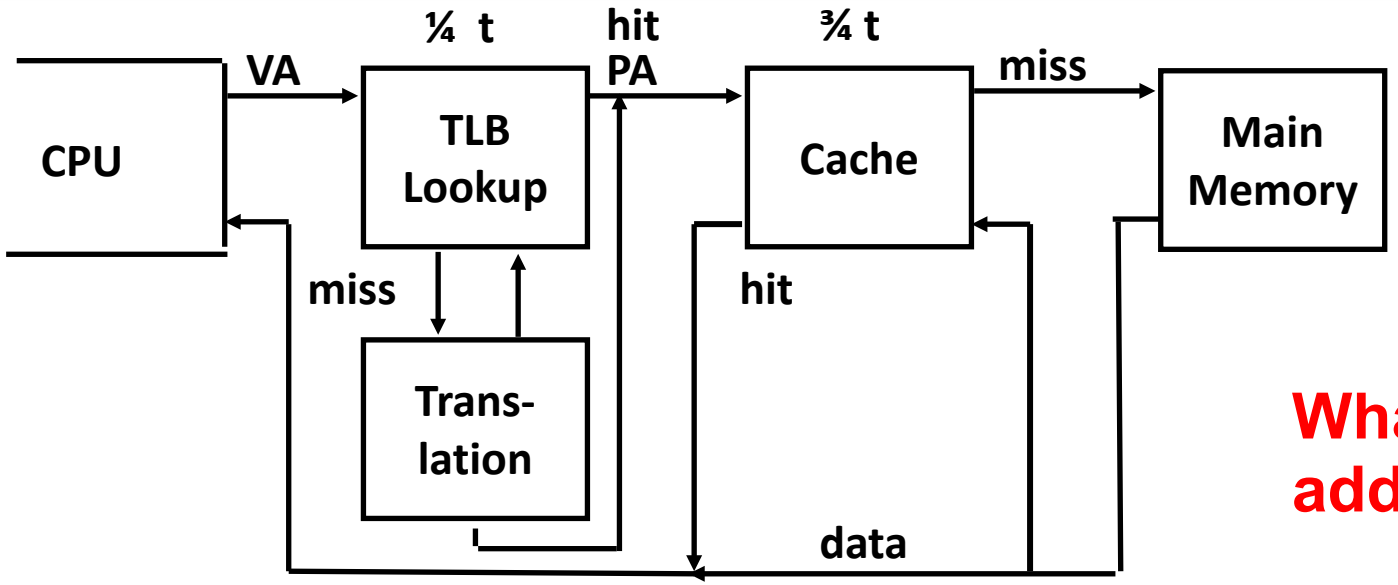
- **This makes memory (cache) accesses very expensive**

- **The hardware solution is to use a Translation Lookaside Buffer (TLB)**
  - A small cache that keeps track of recently used address mappings to avoid having to do a page table lookup in main memory

Making Address Translation Fast



A TLB in the Memory Hierarchy



What about virtually addressed caches?

- **A TLB miss – is it a page fault or merely a TLB miss?**
  - If the page is loaded into main memory, then the TLB miss can be handled by loading the translation information from the page table into the TLB (10's of cycles)
  - If the page is not in main memory, then it's a true page fault (1,000,000's)

- **TLB misses are much more frequent than true page faults**

TLB Event Combinations

TLB	Page Table	Cache	Possible? Under what circumstances?
Hit	Hit	Hit	Yes – this is what we want!
Hit	Hit	Miss	Yes – although the page table is not checked after the TLB hits
Miss	Hit	Hit	Yes – TLB missed, but PA is in page table and data is in cache
Miss	Hit	Miss	Yes – TLB missed, but PA is in page table, data not in cache
Miss	Miss	Miss	Yes – page fault
Hit	Miss	Miss/ Hit	No – TLB translation is not possible if the page is not present in main memory
Miss	Miss	Hit	No – data is not allowed in the cache if the page is not in memory

Measuring Performance with Caches

- **Assuming cache hit costs are included as part of the normal CPU execution cycle, then**

$$\text{CPU time} = \text{IC} \times \text{CPI} \times \text{CP}$$

$$= \text{IC} \times (\underbrace{\text{CPI}_{\text{ideal}} + \text{Memory-stall cycles}}_{\text{CPI}_{\text{stall}}}) \times \text{CP}$$

Memory-stall cycles come from cache m… write-stalls)

Note: this is miss ratio with regard to all instructions (not only LW) = read ratio * cache miss rate

$$\text{Read-stall cycles} = \text{read miss ratio} \times \text{read miss penalty}$$

$$\text{Write-stall cycles} = \text{write miss ratio} \times \text{write miss penalty} + \text{write buffer stalls}$$

For write-through caches, we can simplify this to

$$\text{Memory-stall cycles} = \text{miss ratio} \times \text{miss penalty}$$

Impacts of Cache Performance

- **Relative cache penalty increases as processor performance improves (faster clock rate and/or lower CPI)**
  - The memory speed is unlikely to improve as fast as processor cycle time. When calculating CPI_{stall}, the cache miss penalty is measured in *processor* clock cycles needed to handle a miss
  - The lower the CPI_{ideal}, the higher the impact of stalls

- **Example: A processor with a CPI_{ideal} of 2, a 100 cycle miss penalty, 36% load/store instructions, and 2% I$ and 4% D$ miss rates**

  Memory-stall cycles = 2% × 100 + 36% × 4% × 100 = 3.44

  CPI_{stalls} = 2 + 3.44 = 5.44

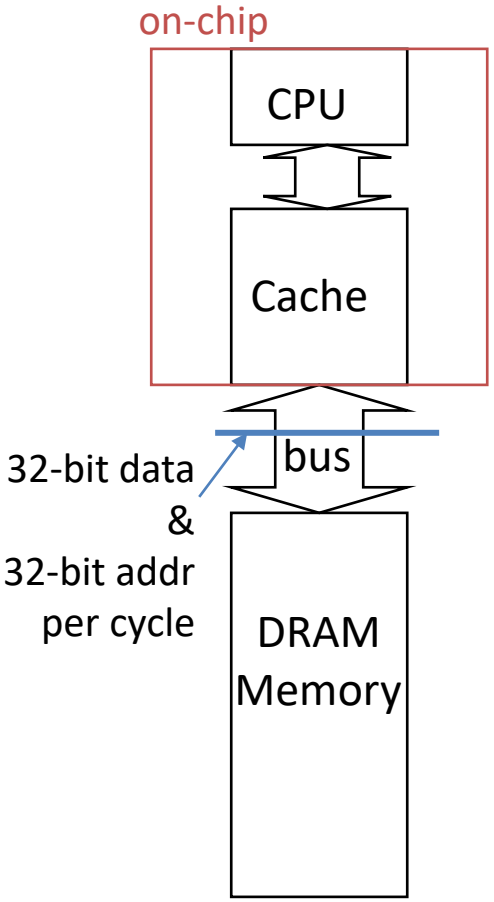
- **What if the CPI_{ideal} is reduced to 1? Or the processor clock rate is doubled (doubling the miss penalty)?**

Memory (DRAM) + Caches

- **It is important to match the cache characteristics**
  - Caches want information provided to them one block at a time (and a block is usually more than one word)

- **Memory design considerations:**
  - With the main memory characteristics
    - Use DRAMs that support fast multiple word accesses, preferably ones that **match the block size** of the cache
  - With the memory-bus characteristics
    - Make sure the memory-bus can support the DRAM access rates and patterns
    - With the goal of increasing the Memory-Bus-to-Cache bandwidth

Memory Systems that Support Caches



One word wide organization
(one word wide bus & memory)

The off-chip interconnect and memory architecture can affect overall system performance in dramatic ways

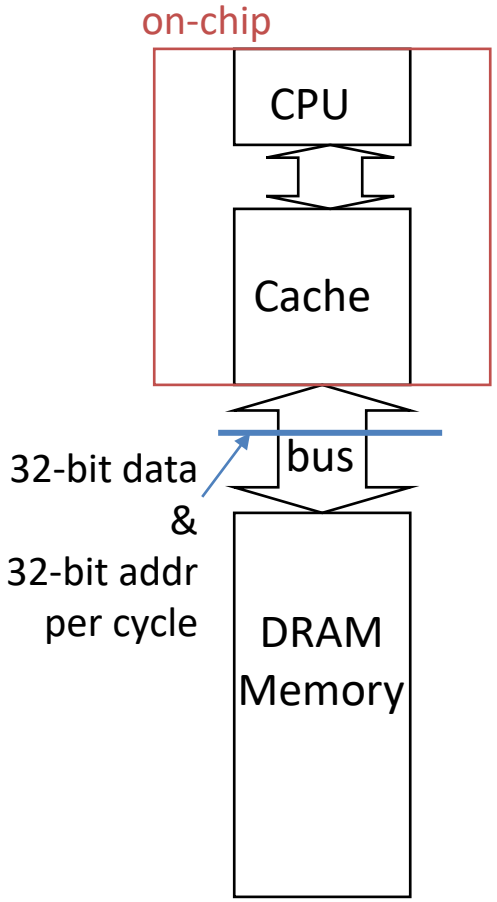
Assume:

- 1 memory bus clock cycle to send the address
- 15 memory bus clock cycles to get the 1st word in the block from DRAM (row cycle time), 5 memory bus clock cycles for 2nd, 3rd, 4th words (column access time)
- 1 memory bus clock cycle to return a word of data

Memory-Bus to Cache bandwidth

- number of bytes accessed from memory and transferred to cache/CPU per memory bus clock cycle

One Word Wide Bus, One Word Blocks



One word wide organization
(one word wide bus & memory)

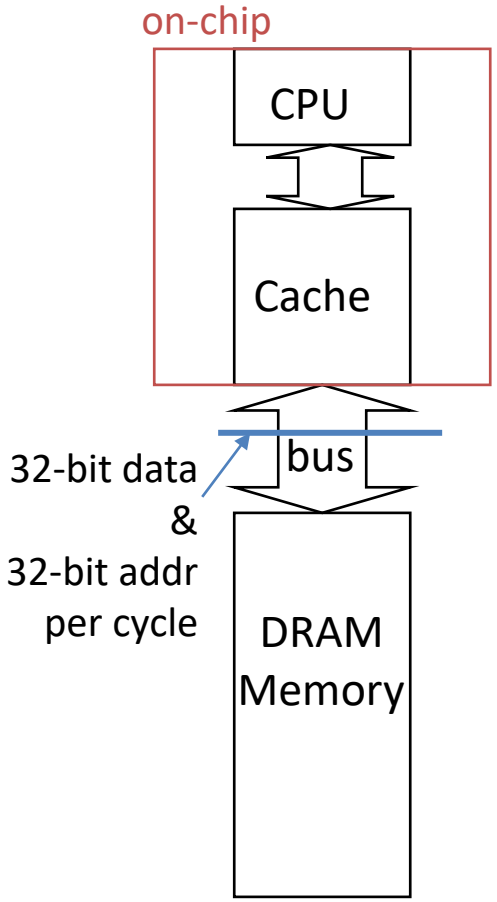
- **If the block size is one word, then for a cache miss, the pipeline will have to stall for:**

  1 memory bus clock cycle to send address
  15 memory bus clock cycles to read DRAM
  1 memory bus clock cycle to return data
  17 total clock cycles miss penalty

- **Number of bytes transferred per clock cycle (bandwidth) for a single miss is**

  4 / 17 = 0.235 bytes per memory bus clock cycle

One Word Wide Bus, Four Word Blocks



One word wide organization
(one word wide bus & memory)

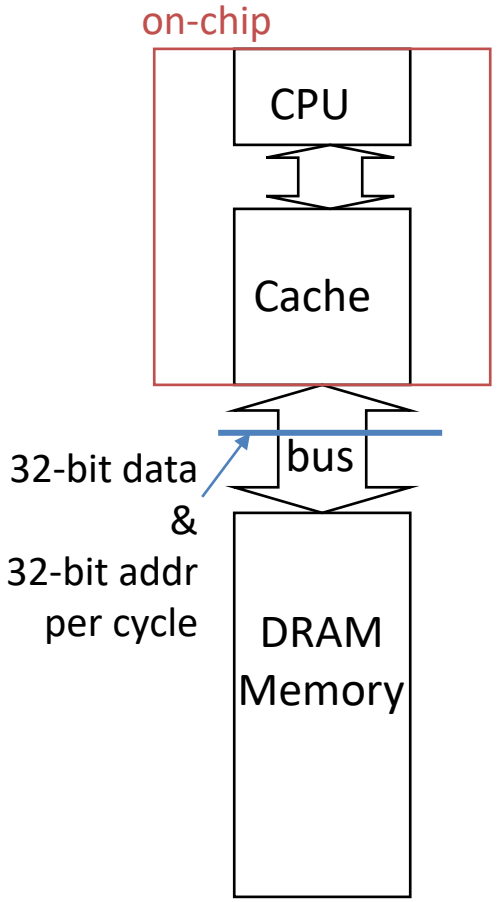
- **What if the block size is four words and each word is in a different DRAM row?**

1	cycle to send 1st address
4 x 15 = 60	cycles to read DRAM
1	cycles to return last data word
62	total clock cycles miss penalty

- **Number of bytes transferred per clock cycle (bandwidth) for a single miss is**

(4 x 4) / 62 = 0.258 bytes per memory bus clock cycle

One Word Wide Bus, Four Word Blocks



One word wide organization
(one word wide bus & memory)

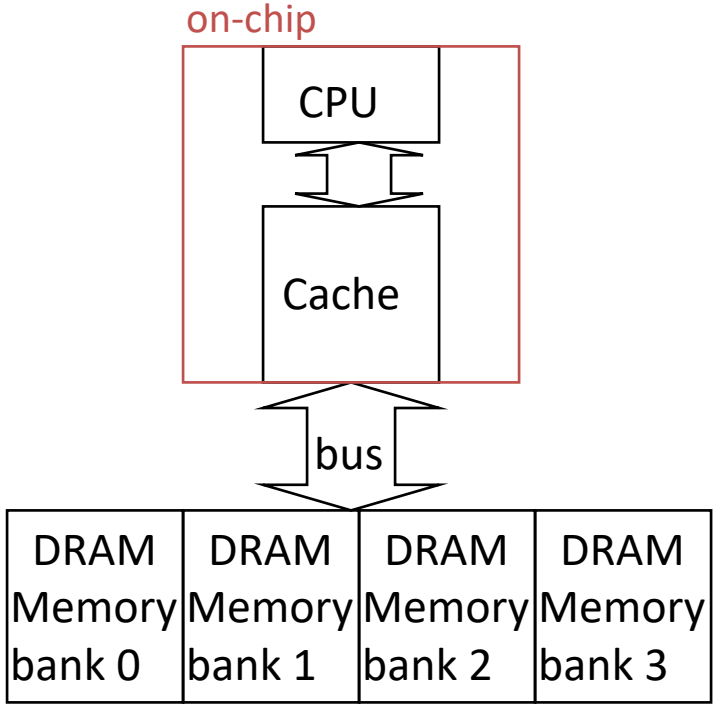
- **What if the block size is four words and all words are in the same DRAM row?**

1	cycle to send 1st address
15 + 3 x 5 = 30	cycles to read DRAM
1	cycles to return last data word
32	total clock cycles miss penalty

- **Number of bytes transferred per clock cycle (bandwidth) for a single miss is**

(4 x 4) / 32 = 0.5 bytes per memory bus clock cycle

Interleaved Memory, One Word Wide Bus



- **For a block size of four words**

1	cycle to send 1st address
15	cycles to read DRAM banks
4 x 1 = 4	cycles to return last data word
20	total clock cycles miss penalty

- **Number of bytes transferred per clock cycle (bandwidth) for a single miss is**

(4 x 4) / 20 = 0.8 bytes per memory bus clock cycle

What about interleaving channels?

Conclusion Time

What is TLB?

Cache for page table

What are the three cases (in terms of delay) for address translation?

TLB hit, page table hit, page fault

SAN JOSÉ STATE UNIVERSITY *powering* SILICON VALLEY