Computer Organization & Software

Anjali Agarwal
Concordia University

Acknowledgements

Material used in this set of slides based on:

- P. E. Livadas and Ch. Ward, “Computer organization and the MC68000
- C. Hamacher, Z. Vranesic and S. Zaky, “Computer Organization”,
- D. A. Patterson and J. L. Hennessy, “Computer Organization & Design”

Memory - Primary Storage

- ***Primary Storage*** - fast memory operating at electronic speed
- One of user wishes is to have unlimited amount of fast memories
  - » *Not feasible, however computer memory organization oriented towards creating "reliable illusion" of unlimited fast memory*
  - » *How to create such illusion*
    - Imagine preparing research project on some broad engineering topic
    - You can find partial material in several library books, while none of the book contains the comprehensive description of all data needed for your project
    - To save time, you want to gather all the relevant books on your desk at the same time instead of keeping only one book, and returning to library any time you need additional information from other books
      - Notice, that you still use one book at the time, however, in the case you need to have a look at the other book, you have it just at hand instead of going to library
  - » *How the above example translates into illusion of having unlimited fast computer memory?*
    - As much as you did not access all needed library books at once (you accessed each of them with equal probability), computer program does not access all lines of code at the same time
      - Parts of program accessed with equal probability

Memory Locality

- ***Def. Principle of locality***: Program accesses relatively small portion of memory address space at any time
  - » *Programs tend to reuse data and instructions near those they have used recently, or that were recently referenced themselves*
  - » *Temporal locality: Recently referenced items are likely to be referenced in the near future*
  - » *Spatial locality: Items with nearby addresses tend to be referenced close together in time*
- Locality in computer programs comes from their most often assumed structure

Example:

- » *Majority of numerical algorithms are programmed in terms of loops*
  - – Instructions and data accessed in one loop iteration will most likely be accessed in iterations to follow - temporal locality
- » *Most of the program body written in sequential form (with exceptions for jumps)*
  - – exhibiting spatial locality property

Memory Hierarchy

Hierarchical computer memory structure based on principal of locality

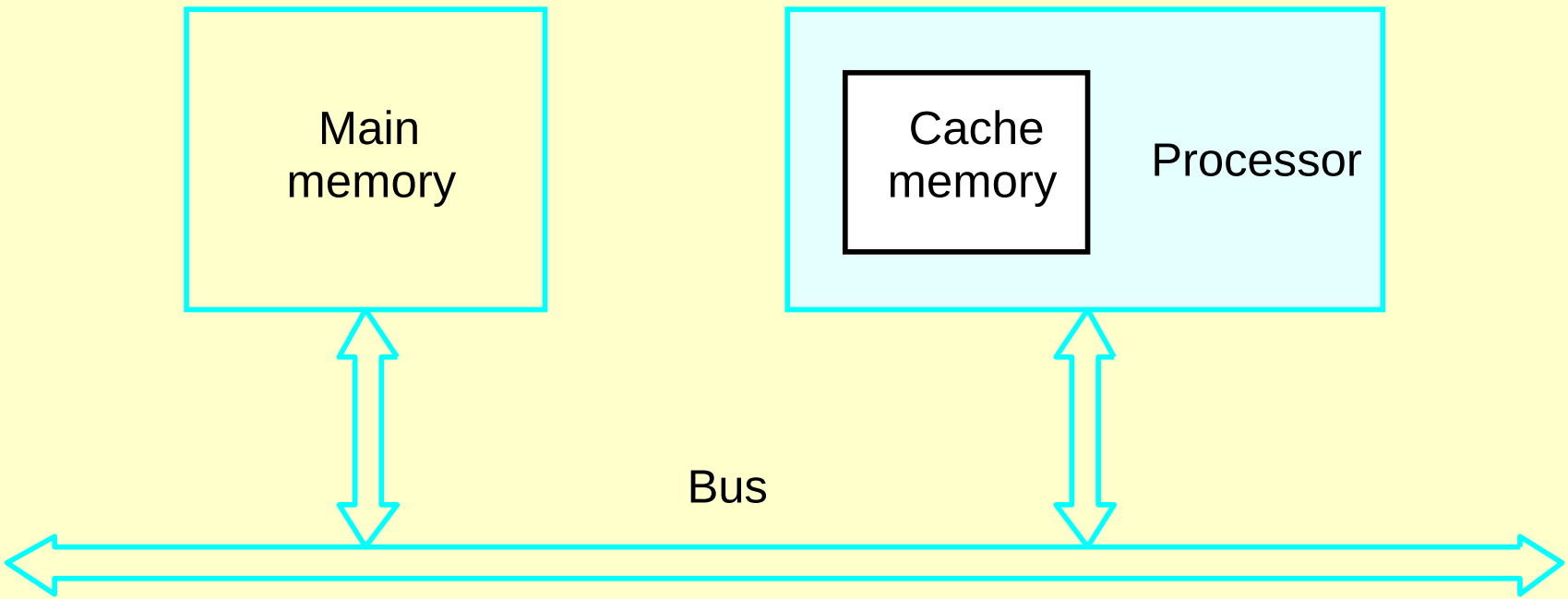
- » *Memory hierarchy structured from levels of memory differing in speed and size*
  - The faster memory, the more expensive, tends to be smaller
  - The further level from processor the more time it takes to access it
- » *Three mainstream technologies used to build memory hierarchy*
  - Main memory implemented from DRAM
  - Cache used SRAM (more costly than DRAM but faster)
  - Largest and slowest level of memory hierarchy (secondary storage) implemented using cheap magnetic tapes

♦ Majority of program execution time spent on accessing memory

- » *Memory system a necessary factor contributing to overall computer performance*
  - ***Good understanding of memory hierarchy and application of this knowledge to the structure of the created program can make a big difference in the execution time, particularly of large programs***

Memory Hierarchy (cont.)

- Memory hierarchy consists of multiple levels
  - » *Data copied only between two adjacent levels at the time*
    - Level closer to processor smaller and faster than lower level
      - Built out of more expensive technology
    - Information present at two adjacent levels divided into blocks
      - If requested by processor data located in some block in upper level then we call it a ***hit***
      - If requested data not found in any block in upper level memory, then we call it a ***miss***
        - » The lower level is accessed to get the requested information
      - ***Hit ratio***: fraction of memory accesses found in upper level used as measure of memory performance
      - ***Miss ratio***: 1- *hit ratio*
      - ***Hit time****: time* needed to access data in upper level
      - ***Miss penalty****:* time needed to replace missing block in upper level with data from lower level plus time needed to deliver the block to requesting processor

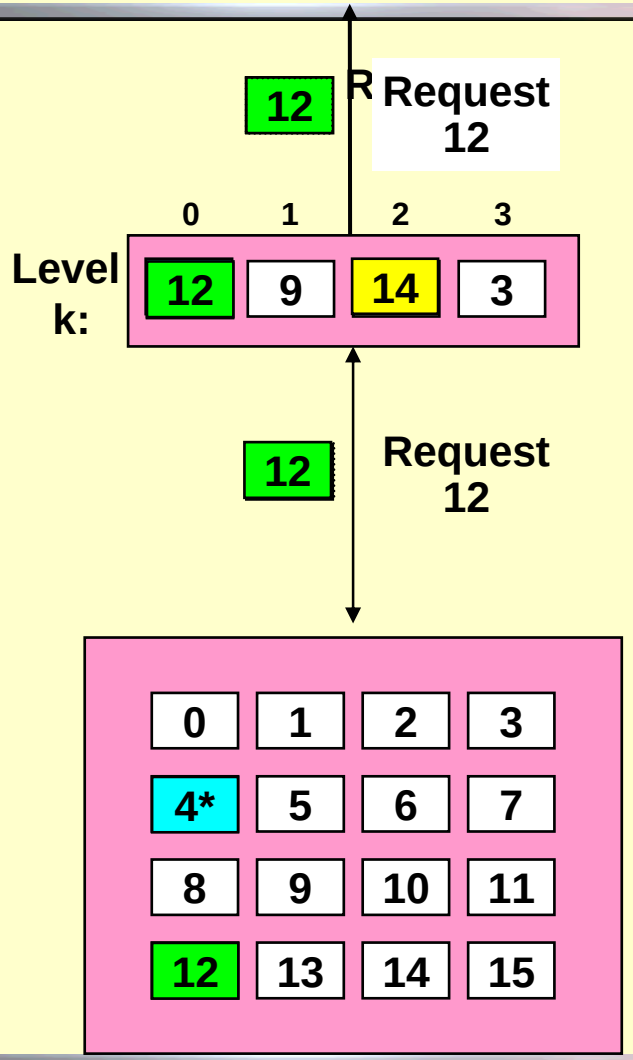


The processor cache.

Cache Memory

- ***Cache* -** initially referred to a level of memory hierarchy between CPU and main memory
  - » *Nowadays the term extended to any storage taking advantage of locality of access*
- Cache first introduced in 1960s, today every computer has one
- How cache operates?
  - » *Cache supposed to be a fast memory, i.e., access to cache for a block requested by processor should be as fast as possible*
  - » *Programs tend to access the data at level k (e.g. cache) more often than they access the data at level k+1 (e.g. Main memory).*
  - » *Thus, the storage at level k+1 can be slower, and thus larger and cheaper per bit.*

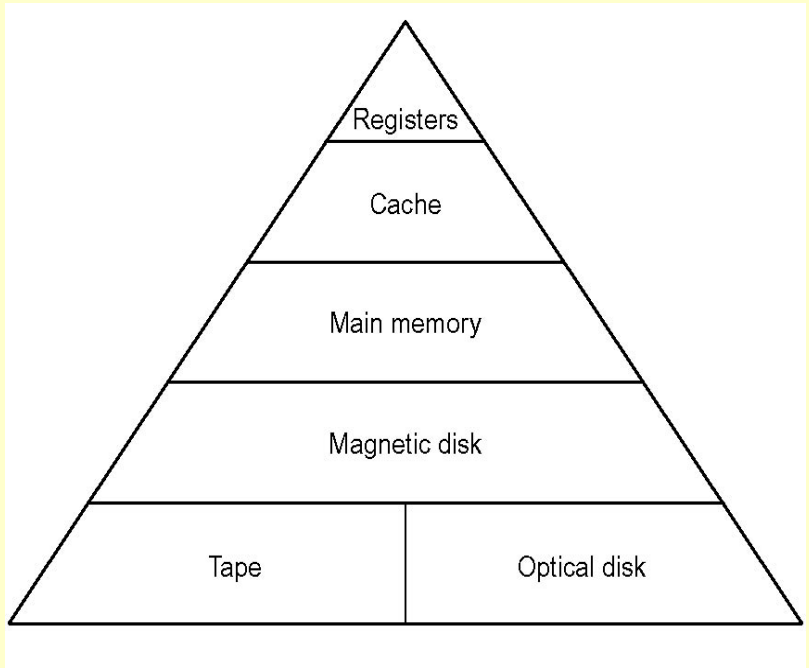
General Caching Concepts



- Program needs object *d*, which is stored in some block *b*
- Cache hit
  - *Program finds* b *in the cache at level* k *e.g., block 14*
- Cache miss
  - b *is not at level* k, *so level* k *cache must fetch it from level* k+1 *e.g., block 12.*
  - *If level* k *cache is full, then some current block must be replaced (evicted). Which one is the "victim"?*
    - Placement policy: where can the new block go? e.g., b mod 4
    - Replacement policy: which block should be evicted?
- More details in COEN 316 course

Secondary Storage

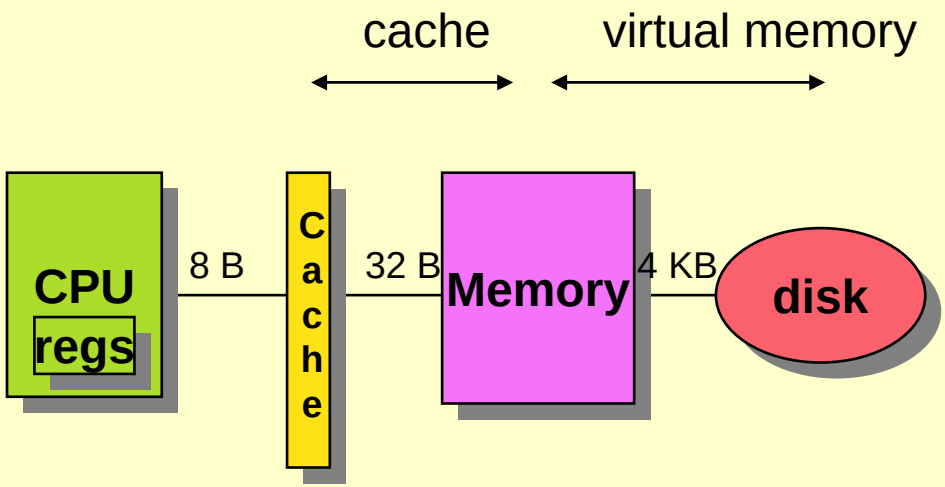
- Regardless of main memory size, it is always too small to host all programs and data
  - » *Secondary storage used when large amounts of data and many programs are to be stored and accessed infrequently (long access time)*
- Traditional storage pyramid: CPU registers accessing CPU at speed, cache memory (from 32K to few MB), main memory (MB – GB), secondary storage
- Secondary storage, size limited by budget:
  - » *Magnetic disks and tapes, floppy disks, magneto-optical disks*
- Down pyramid storage capacity increases, speed of accessing data decreases



Virtual Memory

- Cache used to "speed up" main memory by allowing fast access to recently used portion of program/data from main memory
  - » *As cache is built in much faster technology than main memory, its size is smaller than main memory (cost issue)*
- Main memory although large, often not sufficient to run programs and applications for many users
  - » *Need to use secondary storage to keep all programs/data*
  - » *Access time to secondary storage much longer than to main memory - overall program execution would be significantly slower*
  - » **Solution***: introduction of cache concept to speed up interaction between main memory and secondary storage*
- ***Virtual memory*** concept: main memory acts as "cache" for secondary storage
  - » *Main memory can be more efficiently shared among many users*
    - Total amount of memory needed to run simultaneously all required applications may be way larger than size of main memory
      - In reality only fraction of main memory containing active program section is actively used at given time
    - ***Requirement***: memory allocated to one program must be protected from overwriting by other simultaneously running applications

Levels in Memory Hierarchy



	Register	Cache	Memory	Disk Memory
size:	32 B	32 KB-4MB	1024 MB	100 GB
speed:	1 ns	2 ns	30 ns	8 ms
$/Mbyte:		$125/MB	$0.20/MB	$0.001/MB
line size:	8 B	32 B	4 KB	

larger, slower, cheaper