**CHAPTER 4**

**4.1- Computer Memory System OverviewCharacteristics of Memory Systems**

**Location**

Refers to whether memory is internal and external to the computerInternal memory is often equated (make equal) with main memoryProcessor requires its own local memory, in the form of registersCache is another form of internal memoryExternal memory consists of peripheral storage devices that are accessible to the processor via I/O controllers

**Capacity**Memory is typically expressed in terms of bytes

**Unit of transfer**For internal memory the unit of transfer is equal to the number of electrical lines into and out of the memory module

**Method of Accessing Units of Data-Sequential access-Direct access-Random access-Associative**

**Direct Access**Location of each sector is idenfified by a unique numberT1: seek time, time for moving the head to the accessed trackT2: Rotational delay, time for rotating the disk to position the head to the beginning of the accessed sectorT3: Transfer time, time for rotating the disk to access all the accessed sectorAccess time = T1 + T2 + T3

**Each sector is accessed using different access time**

**Random AccessThe time to access a given location is independent of the sequence of prior accesses and is constant**

**Capacity and Performance:The two most important characteristics of memoryThree performance parameters are used:**

**+Access time (latency)**

**+Memory cycle time**

**+Transfer rate**

**MemoryThe most common forms are: Semiconductor memoryMagnetic surface memory OpticalMagneto-optical**

**Volatile memory** Information decays naturally or is lost when electrical power is switched off

Nonvolatile memory Once recorded, information remains without deterioration until deliberately changed.No electrical power is needed to retain information

Magnetic-surface memories : Are nonvolatile

Semiconductor memory : May be either volatile or nonvolatile

Nonerasable memoryCannot be altered, except by destroying the storage unitSemiconductor memory of this type is known as read-only memory (ROM)

**For random-access memory the organization is a key design issueOrganization refers to the physical arrangement of bits to form words**

**Memory HierarchyDesign constraints on a computer’s memory can be summed up by three questions:How much, how fast, how expensiveThere is a trade-off among capacity, access time, and costFaster access time, greater cost per bitGreater capacity, smaller cost per bitGreater capacity, slower access time**

**A great capacity memory but cheap and low speed + one or some small capacity memory but fast and more expensive (cache) .**

**4.2- Cache Memory PrinciplesCache:** A small size, expensive, memory which has high-speed access is located between CPU and RAM (large memory size, cheaper, and lower-speedMemory).

**Cache/Main Memory Structure**Main memory is divided into the same size blocks. Some blocks will be loaded to cache.

Address in cache is different from those in main memory ->A mapping is needed.Cache Addr ->Main Mem Addr

**4.3- Elements of Cache Design**A blue and white background with black text

AI-generated content may be incorrect.

**Main Memory Address Specifications1.Physical Address:** When the operating system is upgraded, the OS needs more memory ( ex: 4000 bytes), old applications are not compatible.Address must be specified by an other way to ensure that old programs can be run in new OS.**2.Abtract Address:** **All addresses in an application will be specified by compilers using an offset (difference) from the base address (position at which the app. is loaded)-A register (base register) must be added to maintain the base address of the process->An address is specified by <base, offset)**

**3.Virtual Address:Opcode:** **Paging- SegmentationModern OSs enable multiple programs to run concurrently despite limited memory by dividing program content into pages (fixed size, e.g., 4KB) or segments (variable size). Only required pages/segments are loaded into memory.**

**->Compilers translate addresses into virtual addresses in the format:  
⟨page/segment, offset⟩.**

**Opcode :**(viết tắt của operation code) là phần của lệnh trong ngôn ngữ máy xác định thao tác cần thực hiện. Nó được lưu trữ trong bộ nhớ chính (RAM) như một phần của lệnh khi chương trình được nạp và thực thi bởi CPU.

* **Physical Address: Địa chỉ thực tế trong bộ nhớ RAM.**
* **Abstract Address: Địa chỉ khái niệm do lập trình viên sử dụng, chưa được ánh xạ cụ thể.**
* **Virtual Address: Địa chỉ do hệ điều hành tạo ra, được ánh xạ sang địa chỉ vật lý khi cần.**

**Ánh xạ (Mapping) là quá trình chuyển đổi một giá trị hoặc địa chỉ từ một không gian này sang không gian khác.**

**Virtual memory** allows programs to use memory logically, ignoring physical limitations. **Only necessary parts of a program are loaded into RAM**, enabling large programs to run on limited memory. The MMU (Memory Management Unit) translates virtual addresses into physical addresses during memory access.(Lưu tạm trên ổ đĩa)

**Logical and Physical Caches**A diagram of a computer network

AI-generated content may be incorrect.

**Mapping Function**Because there are fewer cache lines than main memory blocks, an algorithm is needed for mapping main memory blocks into cache lines**Three techniques can be used:DirectAssociativeSet Associative Direct Mapping: Mỗi block bộ nhớ chính chỉ ánh xạ vào một cache line duy nhất.**

** Associative Mapping: Mỗi block có thể nằm ở bất kỳ cache line nào trong cache.**

** Set-Associative Mapping: Bộ nhớ cache được chia thành các tập hợp, mỗi block có thể ánh xạ vào một tập hợp cụ thể.** **Thỏa hiệp kết hợp điểm mạnh của cả direct mapping và associative mapping, đồng thời giảm bớt nhược điểm của chúng.**

**Victim CacheVictim Cache: Giảm xung đột trong direct-mapped cache mà vẫn giữ tốc độ truy cập nhanh. Là fully associative cache, thường có 4–16 lines, nằm giữa L1 cache và bộ nhớ cấp tiếp theo.**

**Replacement Algorithms**Two situations: **Cache hit: Accessed address exists in cacheCache miss: Accessed address does not exist in cache. The memory block containing it must be loaded to the cache**Once the cache has been filled, when a new block is brought into the cache, one of the existing blocks must be replaced**For direct mapping** there is only one possible line for any particular block and **no choice is possibleFor the associative and set-associative techniques a replacement algorithm is neededTo achieve high speed, an algorithm must be implemented in hardware**

**The four most common replacement algorithms are:Least recently used (LRU) (Most effective)**

**First-in-first-out (FIFO)Least frequently used (LFU)**

**Write Policy**Write policy defines how data modifications in the cache are handled with respect to main memory. There are two main strategies:

**Write Through and Write BackWrite through- Ghi thẳng**Simplest techniqueAll write operations are made to main memory as well as to the cacheThe main disadvantage of this technique is that it generates substantial (heavy) memory traffic and may create a bottleneck**Write back-Ghi ngầm**Minimizes memory writesUpdates are made only in the cachePortions of main memory are invalid and hence accesses by I/O modules can be allowed only through the cacheThis makes for complex circuitry and a potential bottleneck

**Line SizeLarger line size -> More data -> Cache hit increases, but expensive and more data in cache but not used (Normal: 64-128 bytes)**

Line size là lượng dữ liệu được chuyển giữa cache và bộ nhớ chính trong một lần truy cập.

* Line size lớn → Giảm số lần truy cập, tận dụng tính cục bộ không gian, nhưng dễ gây lãng phí dữ liệu.
* Line size nhỏ → Ít lãng phí hơn, nhưng tăng số lần truy cập bộ nhớ, gây tăng độ trễ.

**Multilevel CachesTwo-level cache:Internal cache designated as level 1 (L1)External cache designated as level 2 (L2)** On-chip cache improves performance by reducing external bus activity and speeding up execution.

 L1 cache (internal) is faster, while L2 cache (external) reduces memory access delays.

 Multilevel cache effectiveness depends on hit rates in both L1 and L2.

 Design complexity increases with cache size, replacement policy, and write policy.

**Unified Versus Split CachesHas become common to split cache:**One dedicated to instructionsOne dedicated to dataBoth exist at the same level, typically as two L1 caches

**Advantages of unified cache: Higher hit rate**Balances load of instruction and data fetches automaticallyOnly one cache needs to be designed and implemented

**Trend is toward split caches at the L1 and unified caches for higher levelsAdvantages of split cache:Eliminates cache contention (tranh chấp)** between instruction fetch/decode unit and execution unitImportant in pipelining (**cơ chế đường ống, output của xử lý này là input của xử lý kế tiếp)**