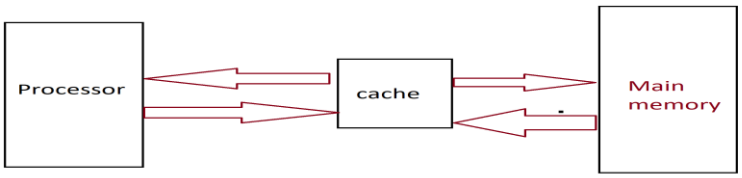
Computer Science Dept. / 2nd year 2nd Course (2024-2025)
Computer Architecture
Dr.Haleema Essa

Cache Memory

A cache memory is a small fast memory placed between a processor and main memory as illustrated in the figure:



- The cache is the ***fastest*** component in the memory hierarchy. It can be viewed as a buffer memory for the main memory.
- Caches are used in different forms to reduce the effective time required by a processor to access addresses, instructions, or data that are normally stored in main memory.
- Sometimes a cache is used to store instructions but not data, in this case the term ***instruction cache or instruction look aside buffer*** are used.
- T***he advantage of restricting a cache to instruction is that***, unlike data, instructions do not change, so the contents of an instruction cache need never be written back to main memory.

❖ Cache Design

- The ***performance goal*** of adding a cache memory to a computer is to make the average memory access time seen by the processor as close as possible to that of the cache.
- ***To achieve this***, a high percentage of all memory reference should be satisfied by the cache,

  i.e., the cache hit ratio should be close to 1.

- This is possible because of ***the locality of-reference property*** of programs.

- **Hit**: means the information are in cache.
- **Miss**: means the information are in main.

$$Cache\ Hit\ Ratio = \frac{\text{No.of Hits}}{\text{No.of Hits+No.of Miss}} * 100$$

$$Cache\ Miss\ \ Ratio = \frac{\text{No.of Miss}}{\text{No.of Hits+No.of Miss}} * 100$$

❖ Performance of Cache Memory

1- $Cs = \frac{\text{Cc Sc+Cm Sm}}{\text{Sc+Sm}}$

Cs: average cost per byte for main plus cache (Cost of System).

Cc: Average cost per byte for cache.

Cm: Average cost per byte for main.

Sc: Size of cache.

Sm: size of main.

2- Ts = H * Tc + (1-H) * Tm

Ts: Average system access time.

Tc: Cache access time.

Tm: main access time.

H: Hit Ratio (1- Miss Ratio)

(1- H): Miss Ratio.

Hit ratio + Miss ratio = 1

Example 1: (Find the Hit ratio) if you have 51 cache hits and three misses over a period of time.

Then that would mean:
you would divide 51 by 54. The result would be a hit ratio of 0.944. The 0.944 result would be multiplied by 100 to get a hit ratio of 94.4%.

No. of Hits = 51
No. of Miss = 3
Cache Hit Ratio = (51/ (51+3)) * 100 = 94.4%

Example 2: (Find the Miss ratio) if the misses of cache was 11, and the total number of requests (Hit + Miss) was 48.

Then you would:
divide 11 by 48 to get a miss ratio of 0.229 and multiply the result by 100.

No. of Miss = 11
No. of requests (Hits + Miss) = 48
Cache Miss Ratio = (11/ 48) * 100 = 22.9%

***Example 3*:** Find the average access time of a system if the hit rate is 80%. The memory access takes 12ns on a hit and 100ns on a miss.
Hit rate =80%
Miss rate=20%
Tc =12 ns
Tm=100ns
Ts = (hit rate * Tc) + (miss rate * Tm) = (0.8*12ns) + (0.2*100ns) = 29.6ns

HW 1: Find the average access time of a system if the hit rate is 90%. A hit takes 0.5ns and a miss takes 10ns.

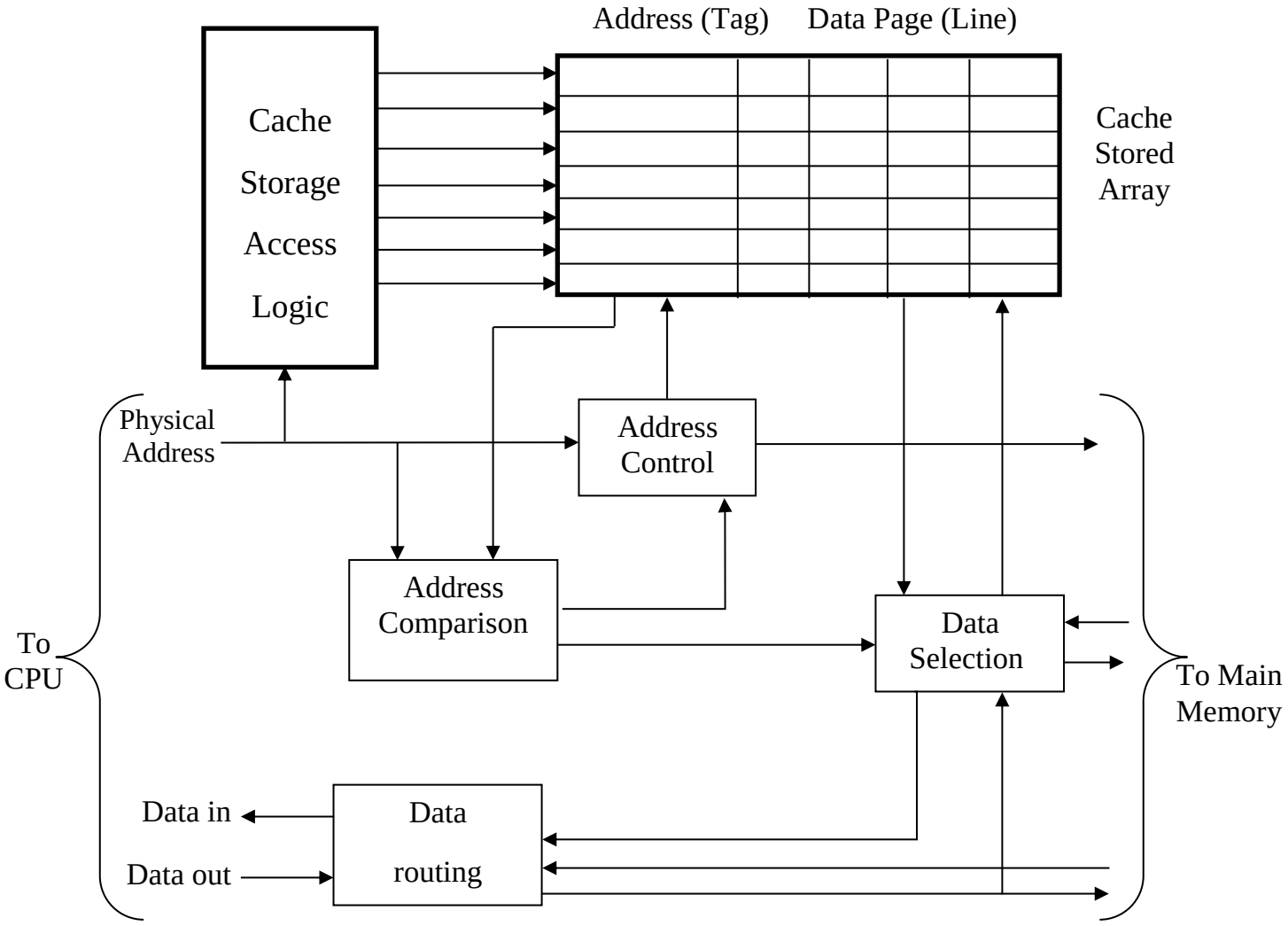
HW 2: Consider a memory system having the following specifications. Find its total cost and cost per byte of the system.

Memory Type	Size	Cost per Byte
SRAM (Cache)	256KB	30$ per MB
DRAM (Main Memory)	128MB	1$ per MB

➢ Principle of locality-of-reference

Over any short period of time execution may be confined to a small section of the program.

The structure of a cache memory unit is outlined in the figure below:



Basic Design of a Cache.

- It stores a set of main memory address A_i and the corresponding (data) words $M(A_i)$.
- The data entries are grouped into blocks (***cache pages, or called "lines"***), caches of which is a sub block of some main-memory page; the corresponding stored address is therefore a block address.
- The contents of the cache array are thus copies of a set of small noncontiguous main memory blocks tagged with addresses.

A cache typically operates as follows:

1. A physical address A is sent to the cache from the CPU at the start of a read (load) or write (store) memory access cycle.

2. The cache compares the relevant part of A, sometimes called the address tag, to all the addresses it currently stores. If there is a match, i.e., a cache hit, then the cache selects the desired word M(A) from the data entry corresponding to A.

3. It completes the memory cycle by transferring data from the CPU to its copy of M(A) and routing it to the CPU (write operation).

4. If A fail to match any of the stored addresses, i.e., a cache miss occurs, then the cache usually initiates a sequence of one or more main-memory read cycles to copy into the cache the main-memory block P(A) that contains the desired item M(A).