Chapter 09: Caches

Lesson 01:

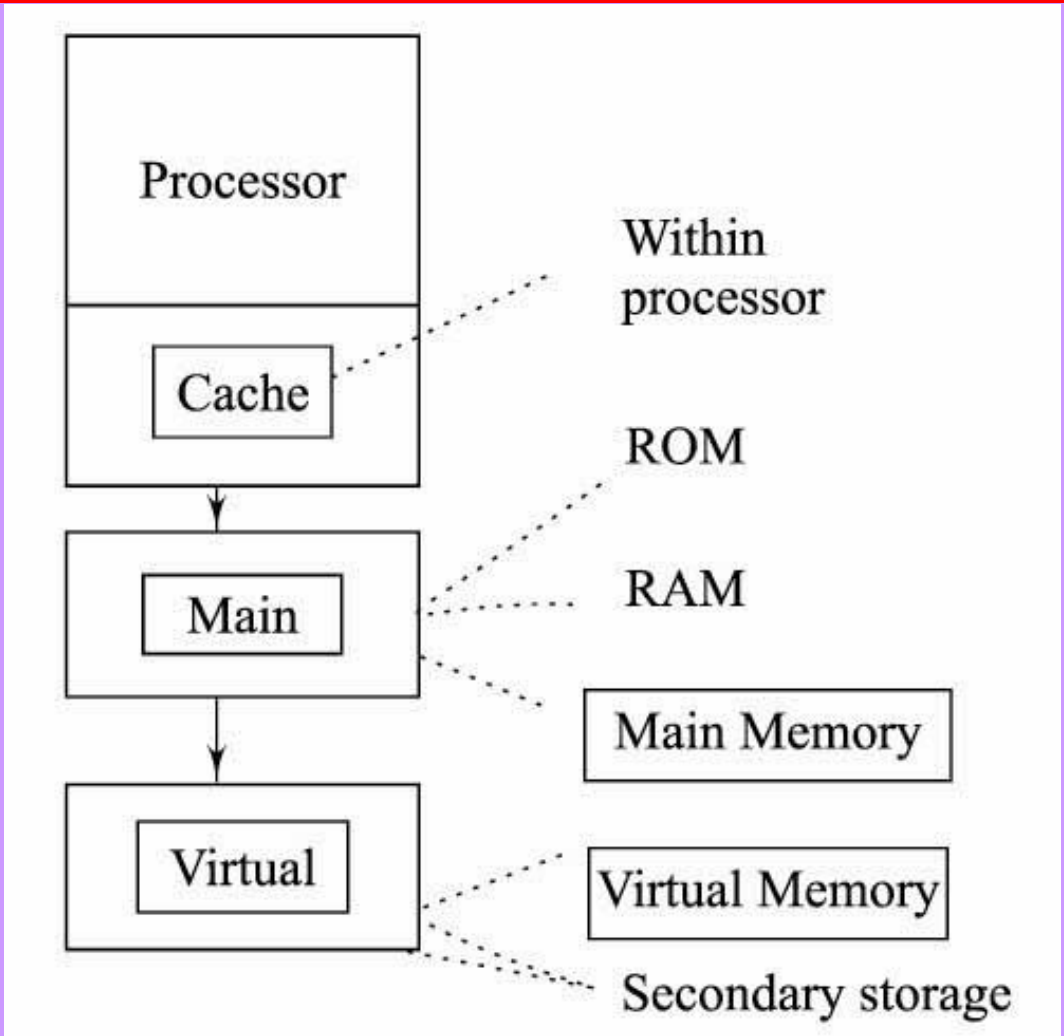
Cache and its Access

Objective

- Understand the cache block and cache access for a data-read from cache or write to cache
- Learn Harvard and Princeton cache memory architectures

Cache Block Diagram

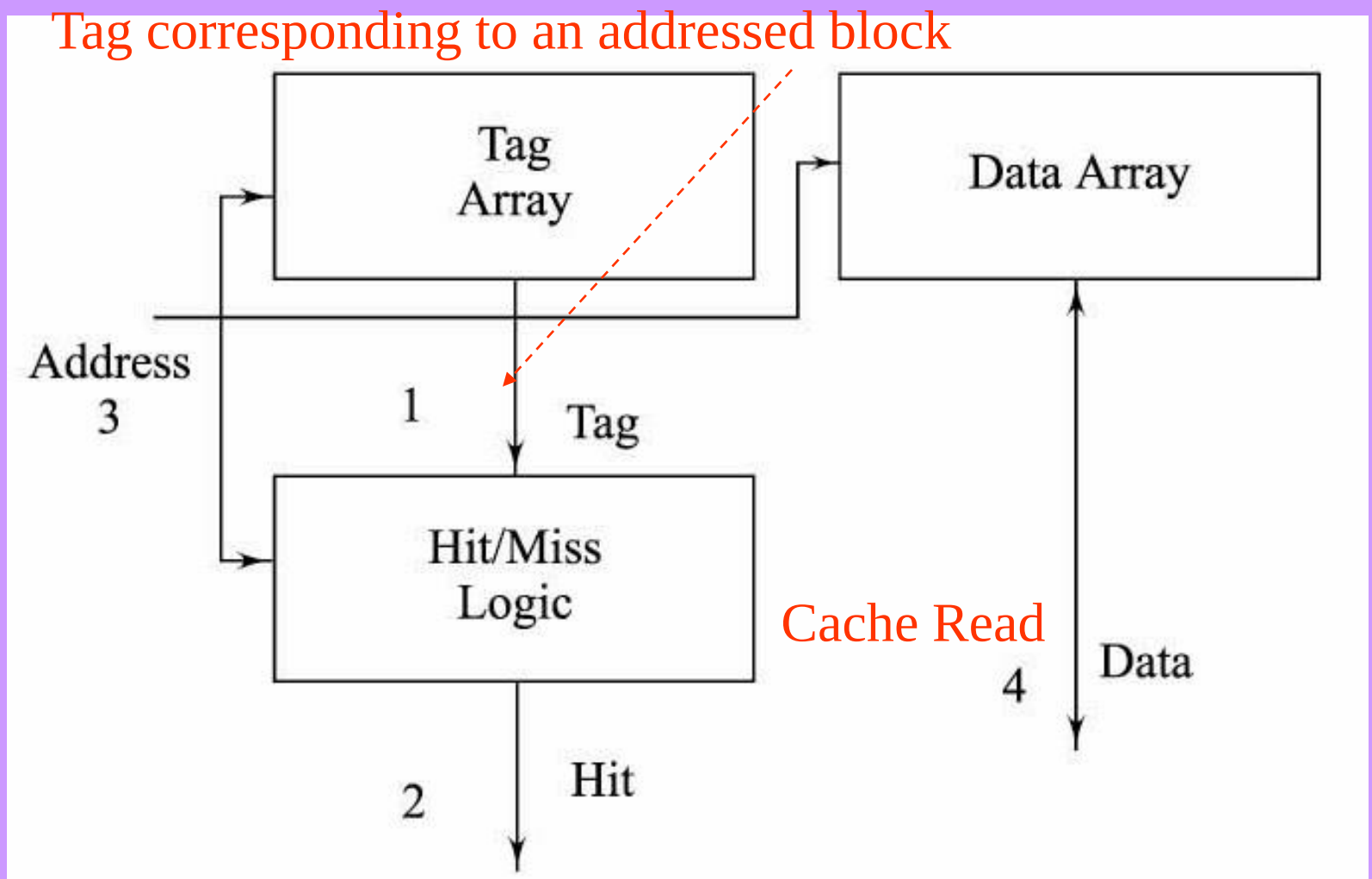
Cache within the processor



Tag and Data Arrays

- A tag array contains the addresses of the data contained in the cache
- The data array contains the data itself

Cache block diagram



Steps 1-4

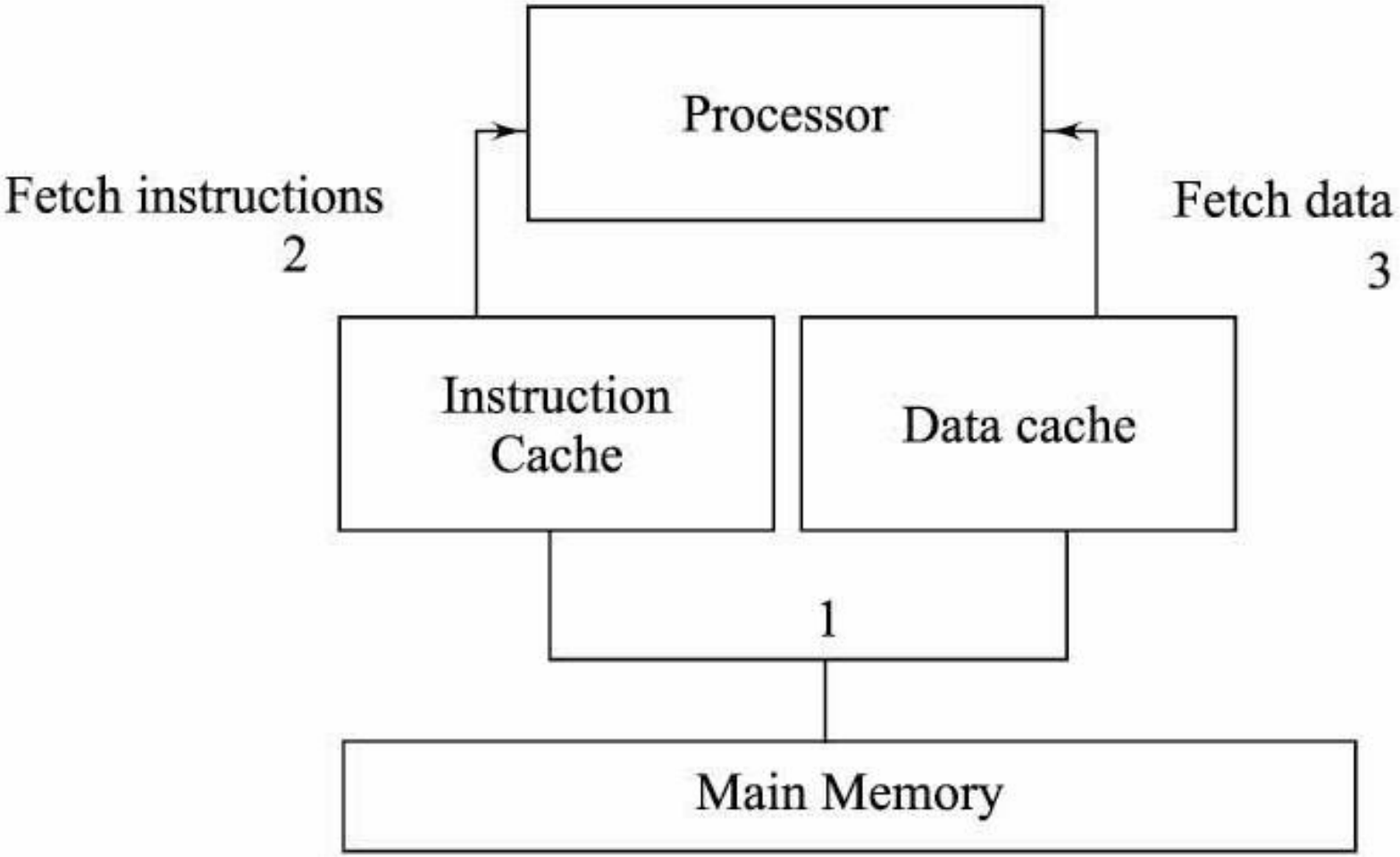
- Step 1 — Processor accesses tag array
- Step 2 — Once the tag array has been accessed, its output must be compared to the address of the memory reference to determine if a hit has occurred
- Step 3 — Using address access data array
- Step 4 — read or modify the addressed data at the data array and get the final result

Advantage of Tag and Data Arrays

- Dividing the cache into separate tag and data arrays reduces the access time of the cache
- The tag array typically contain many fewer bits than the data array
- Can therefore be accessed more quickly than either the data array or a single combined tag/data array

Data cache, Instruction cache and Unified cache

Harvard architecture in cache



Instruction and Data Addresses

- Instructions and data share address space within each level of the memory hierarchy
- For the main memory and the virtual memory, this is generally true

Cache

- Memory access cycle time is greater than the processing unit for an instruction execution
- Therefore, instruction cache memory inside the processor is used for issuing instructions to the execution unit at a fast rate

Harvard Cache

- However, for caches, data and instructions are often stored in separate data and instruction caches
- Harvard architecture allows the processor to fetch instructions from the instruction cache and data from the data cache simultaneously

Harvard Cache Advantages

- Programs do not in general modify their own instructions
- Thus, instruction caches can be designed as read-only devices that do not allow modification of the instructions they contain
- An instruction cache can simply discards any blocks that have to be evicted from it without writing them back to the main memory, since the data they contain is guaranteed not to have changed since it was brought into the cache

Harvard Cache Advantages

- Finally, keeping the instruction and data caches separate prevents conflicts between blocks of instructions and data that might map into the same storage locations in a unified cache

System's instruction cache significantly smaller

- Often a system's instruction cache will be significantly smaller (two to four times) than its data cache
- This is because the instructions for a program generally take up much less memory than the program's data

System's instruction cache significantly smaller

- Also most programs spend the majority of their time in loops that use the same instructions multiple times
- As a result of this, its data cache made larger and programs have the same hit rate, so designers often choose to devote more chip area to the data cache than the instruction cache

Disadvantage

- When a program modifies its own instructions, those instructions are treated as data and are stored in the data cache, not the instruction cache

Princeton Unified Cache

- When a cache contains both instructions and data, it is called a *unified* cache (or Princeton architecture cache).
- Self Modifying Programs easy to execute

Unified Cache (Princeton Architecture)

- Keeping the instruction and data caches not separate
- Instructions and data that might map into the same storage locations in a unified cache

Modified instructions execution

- For the, the program must use special cache flush operations to ensure that the original versions of the instructions are not present in the instruction cache, forcing the program to fetch the modified version from the main memory before they can be executed

Modified instructions execution

- If the data cache is write back additional cache flush operations may be required to ensure that the modified instructions have been written out to the main memory before they are read into the instruction cache
- The additional overhead imposed by these cache flush operations reduces the performance benefit of self modifying code, making it less useful

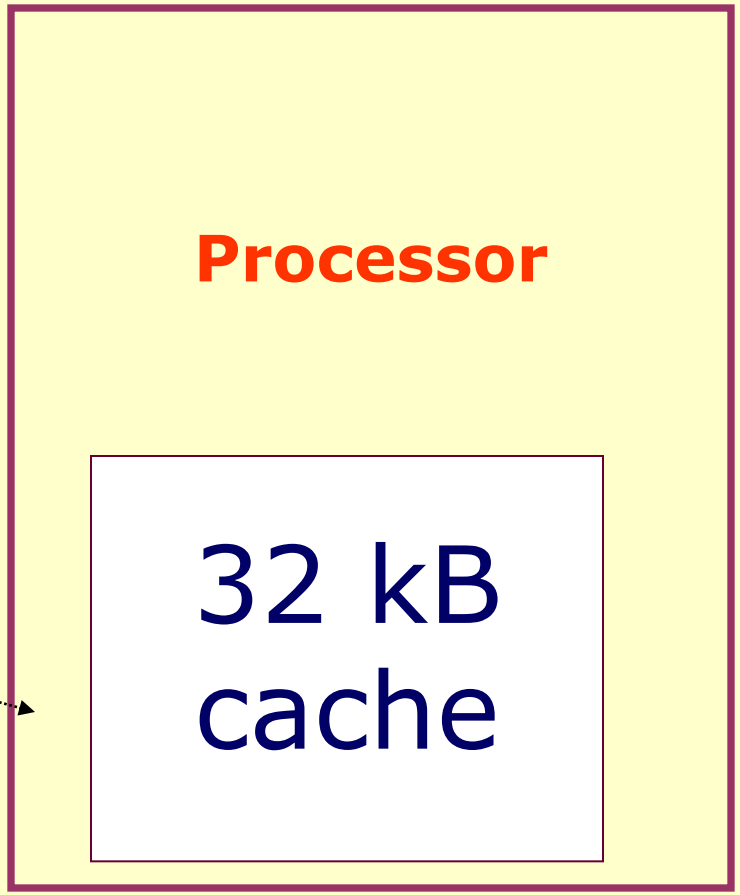
Outline

- Cache Block Diagram
- Data cache, Instruction cache and Unified cache
- **Features describing a cache - Capacity of cache**

Cache Features

- Capacity,
- Line-length,
- Associativity (number of places a given address can reside)
- Replacement policy, and
- Whether the cache is write back or write through.

Capacity



32 kB Data

Array

Capacity

- The *capacity* of a cache is simply the amount of data that .can be stored in the cache,

Summary

We learnt

- Cache Access is first by comparing a tag in tag array and find if referenced address is hit, if yes, then access data from data array at the referenced address
- Instructions and data do not share address space in Harvard Architecture

We learnt

- Instruction and Data Caches separate
- Instruction and Data Caches not separate in Unified Princeton architecture caches

End of Lesson 01on

Cache and it Access