**Classification of Microprocessors based on the architecture:**

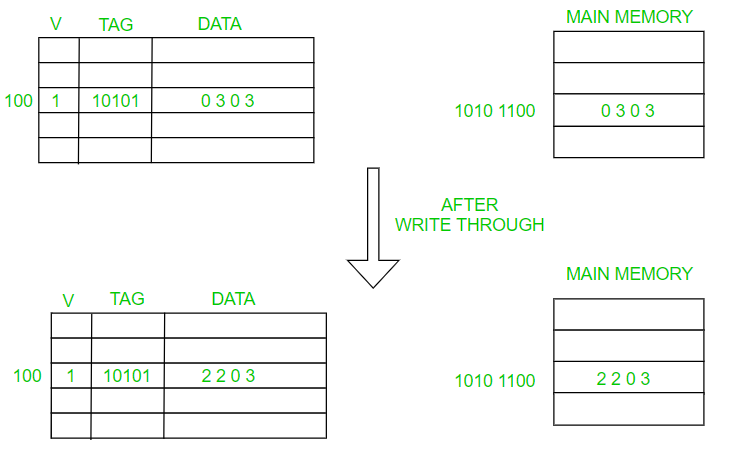
|  |  |
| --- | --- |
| CISC | RISC |
| 1. It stands for Complex Instruction Set Computer. | 1. It stands for Reduced Instruction Set Computer. |
| 1. A large number of instructions are present in the architecture. | 1. Very few instructions are present. The number of instructions is generally less than 100. |
| 1. Some instructions with long execution times. | 1. No instruction with a long execution time due to a very simple instruction set. |
| 1. CISC supports array. | 1. RISC does not support an array. |
| 1. Arithmetic and logical operations can be applied to both memory and register operands. | 1. Arithmetic and logical operations only use register operands. |
| 1. Condition codes are used. | 1. No condition codes are used. |
| 1. The stack is being used for procedure arguments and returns addresses. | 1. Registers are being used for procedure arguments and return addresses |
| 1. Implementation programs are hidden from machine-level programs. | 1. Implementation programs exposed to machine-level programs. |
| 1. Multiple formats are supported for specifying operands. | 1. Simple addressing formats are supported. |
| 1. **Examples:**Intel architecture, AMD | 1. **Examples:** SPARC, POWER PC, etc. |

**Write Policy**

A cache’s write policy is the behavior of a cache while performing a write operation. A cache’s write policy plays a central part in all the variety of different characteristics exposed by the cache. Let’s now take a look at three policies:

* write-through
* write-around
* write-back

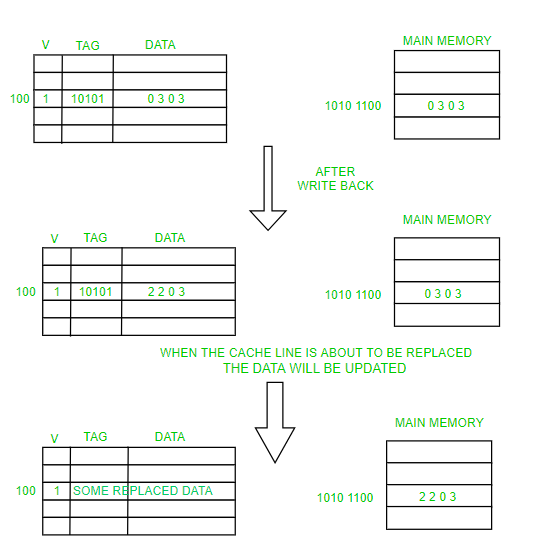
1. **Write Through:**



In write-through, data is **simultaneously updated to cache and memory**. This process is simpler and more reliable. This is used when there are no frequent writes to the cache(The number of write operations is less).

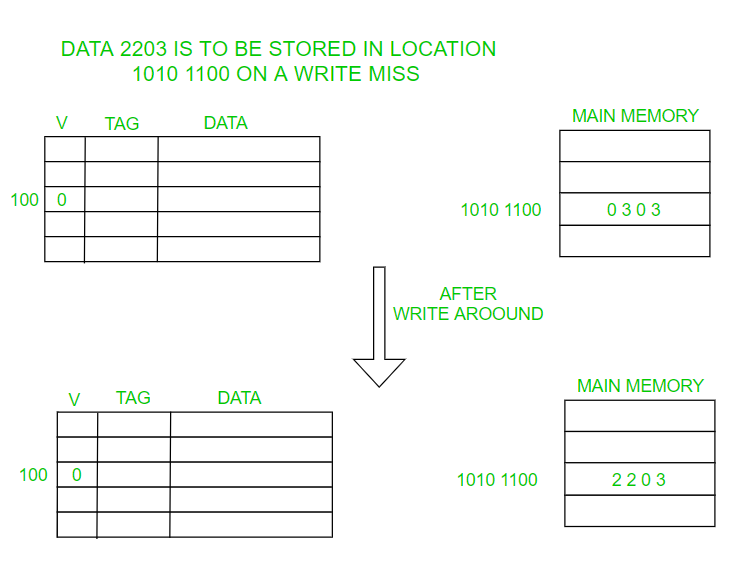
It helps in data recovery (In case of a power outage or system failure). A data write will experience latency (delay) as we have to write to two locations (both Memory and Cache). It Solves the inconsistency problem. But it questions the advantage of having a cache in write operation (As the whole point of using a cache was to avoid multiple access to the main memory)

1. **Write Back:**



The data is updated only in the cache and updated into the memory at a later time. Data is updated in the memory only when the cache line is ready to be replaced (cache line replacement is done using Belady’s Anomaly, Least Recently Used Algorithm, FIFO, LIFO, and others depending on the application).   
Write Back is also known as Write Deferred.

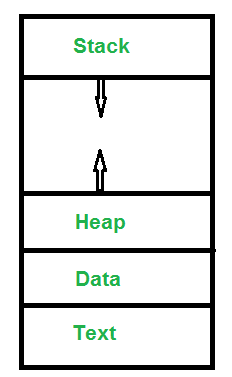
1. **Write Around:**



Here data is Directly written/updated to the main memory without disturbing the cache. It is better to use this when the data is not immediately used again.

**Process:**A process is a program in execution. For example, when we write a program in C or C++ and compile it, the compiler creates binary code. The original code and binary code are both programs. When we actually run the binary code, it becomes a process.

**What does a process look like in memory?**



**States of Process:**

A process is in one of the following states:

**1. New:** Newly Created Process (or) being-created process.

**2. Ready:** After creation process moves to Ready state, i.e. the

process is ready for execution.

**3. Run:** Currently running process in CPU (only one process at

a time can be under execution in a single processor).

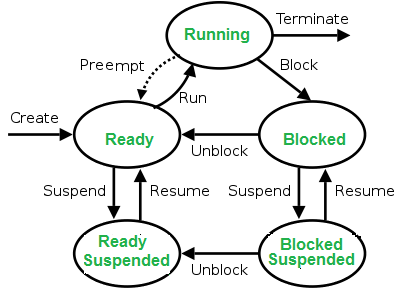
**4. Wait (or Block):** When a process requests I/O access.

**5. Complete (or Terminated):** The process completed its execution.

**6. Suspended Ready:** When the ready queue becomes full, some processes

are moved to suspended ready state

**7. Suspended Block:** When waiting queue becomes full.



**Process Control Block (PCB):**

 A process control block (PCB) contains information about the process, i.e. registers, quantum, priority, etc. The process table is an array of PCB’s, that means logically contains a PCB for all of the current processes in the system.

