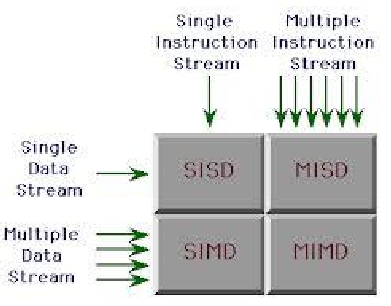
CLASSIFICAZIONE DI COMPUTER PARALLELI

La classificazione di questi computer si può effettuare in tre modi:

* **La classificazione secondo Flyn.**
* **La classificazione basata sulla memoria.**
* **La classificazione basata sui “network interconnection”**

Flyn

Divide le macchine parallele, a seconda del flusso di istruzioni e di dati che queste possono gestire, in quattro categorie:

* **SISD:** le istruzioni di un singolo flusso vengono eseguite una alla volta in modo sequenziale su un singolo flusso di dati preso da input.
* **MISD:** più flussi di istruzioni lavorano contemporaneamente su un unico flusso di dati. Sui dati vengono effettuati più istruzioni.
* **MIMD:** più istruzioni vengono eseguite contemporaneamente su più dati diversi. VENGONO ESEGUITI PIU’ PROGRAMMI IN PARALLELO.
  + **MPMD: Architettura in cui ogni** processore esegue diversi programmi. Di solito esiste un nodo host che distribuisce i dati e infine li raccoglie dopo le varie computazioni.
  + **SPMD:** Architettura costituita di più processori che **eseguono** lo stesso programma in parallelo sebbene in modo indipendente ed asincrono. OVVIAMENTE IL PROGRAMMA CONTERRA’ DIRETTIVE SPECIFICHE IN MODO TALE CHE OGNI PROCESSO ESEGUA SOLO DETERMINATE ISTRUZIONI SU UNO SPECIFICATO SOTTINSIEME DI DATI. [**BRANCHING**].
* **SIMD:**
* Architettura composta da più processori che eseguono contemporaneamente la stessa istruzione però, su dati diversi ad ogni clock. Dunque descrivono macchine parallele che processano solo un’istruzione per volta attraverso un ciclo di **lockstep**, che fa eseguire la stessa istruzione ad ogni processo.
  + L’architettura SIMD ha bisogno anche di meno memoria rispetto alla MIMD poiché solo una copia del programma ha bisogno di essere salvata.

# **CLASSIFICAZIONE IN BASE ALLA MEMORIA**

In un sistema parallelo sono disponibili due configurazioni di memoria:

* **MEMORIA CONDIVISA (o multiprocessore):** 
  + - Più core o unità di elaborazione che accedono alla stessa memoria centrale.
    - Gli scambi di messaggi avvengono attraverso la scrittura in memoria centrale.
    - Quando vi sono più processi che vogliono accedere in memoria si va incontro al ‘**bottleneck’** da risolvere con la **sincronizzazione** (fenomeno che si verifica nel momento in cui le prestazioni di un sistema o le proprie capacità computazionali sono vincolate ad un singolo componente, infatti un solo processore alla volta può accedere e modificare i dati in memoria centrale.
* **MEMORIA DISTRIBUITA (o multi computer):**
  + Si accede a più memorie dislocate anche su più computer collegati tra loro tramite una rete.
  + La comunicazione avviene per via di scambio di messaggi esplicito
  + MACCHINE A PRESTAZIONI MIGLIORI.
  + Dunque ogni processore ha un proprio spazio di indirizzamento logico o fisico privato.

Le **architetture a memoria condivisa** possono essere ulteriormente distinte:

* **UMA**: tipo di architettura in cui tutti i processori condividono la memoria fisica ed il tempo d’accesso a quest’ultima è uniforme poiché indipendente dal processore che fa richiesta e dal chip che contiene i dati di comunicazione.
* **NUMA**: architettura di memoria sviluppata per i sistemi multiprocessore in cui la memoria è condivisa a livello logico ma a livello fisico non è proprio così e dove il tempo di accesso alla memoria centrale dipende dalla posizione del processo rispetto a quest’ultima.
  + La soluzione a questo problema è rappresentata dall’**utilizzo della cache** (= un tipo di memoria dalle dimensioni ridotte che mantiene copie di dati di cui si fa spesso uso o di cui si è fatto recentemente uso.
  + Attraverso questa soluzione si possono ottenere elevate prestazioni limitando di fatto l’accesso alla memoria centrale.

Quando si parla di cache però è importante parlare del problema che quest’ultima implica: la gestione della **CACHE COERENCE**: visto che nella cache sono presenti copie locali dei dati presenti in memoria centrale, ogni volta che un processore modifica un dato nella propria memoria locale, questo deve essere modificato anche nelle altre cache e in memoria centrale. Questa operazione può essere fatta attraverso due procedure:

* **Protocollo di aggiornamento**: aggiorna le altre copie
* **Protocollo invalidante**: invalida le altre copie in modo da non poter più essere utilizzate dagli altri processi in modo NON COERENTE.

# Architecture pipeline

* È una variante dell’architettura SIMD anche se non è propriamente così in quanto nella SIMD tutti I processor eseguono la stessa istruzione, mentre nella PipeLine no.
* Precisamente la pipeline è una tecnologia applicate all’architettura dei microprocessori per incrementare il **throughput** ovvero, la quantità di istruzioni eseguibili in parallelo per incrementare al Massimo le performance della macchina.
* La differenza sostanziale che sta nell’architettura pipeline è che ad ogni clock viene eseguita un istruzione completa e non più un passo di un istruzione.
  + Lettura
  + Decodifica
  + Esecuzione istruzione
  + Attivazione memoria
  + Scrittura in memoria