

## Capitolo 1. Introduzione alle Pipeline Grafiche 3D

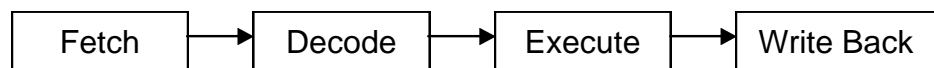
L'oggetto della presente tesina è la descrizione in linguaggio VHDL di due processori, detti *Vertex Shader* e *Pixel Shader*. Si tratta di processori matematici aventi un set di istruzioni specificatamente dedicato all'implementazione di algoritmi comunemente utilizzati nel campo della Computer Graphics.

Allo scopo di rendere i capitoli che seguiranno maggiormente comprensibili, si ritiene opportuno iniziare la trattazione con un capitolo dedicato all'introduzione dei concetti basilari sulla grafica 3D e sulle pipeline grafiche.

### 1.1. Introduzione

#### 1.1.1 Significato di Pipeline

Il primo e fondamentale concetto da illustrare riguarda il significato di pipeline grafica, che cosa sia e come funzioni. Per pipeline si intende una struttura particolare di un processore che permette l'elaborazione di nuove istruzioni indipendentemente dal fatto che le precedenti abbiano o meno completato la loro elaborazione. Pensiamo ad un microprocessore, in linea generale un qualsiasi Intel o AMD; nella elaborazione di codice fornito da un particolare programma il processore esegue quattro stadi indipendenti: Fetch, Decode, Execute e Write Back.



Nello stadio di Fetch una istruzione viene letta dalla memoria e memorizzata in una particolare cache detta *cache istruzioni*. Da lì passa nello stadio di Decode, nel quale il processore interpreta l'istruzione e determina eventuali altri accessi alla memoria. Una volta decodificata, nello stadio di Execute l'istruzione verrà eseguita per ottenere un risultato che sarà scritto in memoria attraverso l'operazione di Write Back (l'ultimo stadio). Nella situazione reale l'esecuzione delle istruzioni avviene in maniera più complessa rispetto quella ora presentata per garantire migliori prestazioni, nondimeno questa sequenza schematizza in modo generale il processo.

La caratteristica fondamentale di questo schema risiede nel fatto che durante l'operazione di Decode non ha senso lasciare inattivo il primo stadio (quello di Fetch) e per questo (come può essere visto nella figura a pagina seguente), mentre l'istruzione segue il suo ciclo naturale, il microprocessore si preoccupa di leggere una nuova istruzione dalla memoria che verrà poi fatta passare nell'intera sequenza di stadi.