



Architettura Von Neumann vs. Harvard

Benvenuti alla lezione odierna sulla differenza tra l'Architettura Von Neumann e l'Architettura Harvard nei computer.

Vediamo come si comportano i due modelli di architettura, il "Modello di Von Neumann" e il "Modello di Harvard", quando eseguono le seguenti due istruzioni:

1. `ldr r0, adr_var1`
2. `ldr r1, adr_var2`

Architettura Von Neumann

L'Architettura Von Neumann, chiamata così in onore del famoso matematico e informatico John von Neumann, è una delle architetture di computer più comuni. In questo modello, il flusso di istruzioni e il flusso di dati condividono la stessa memoria.

Nell'Architettura Von Neumann, un'istruzione come "ldr r0, adr_var1" richiede l'accesso sequenziale alla memoria per leggere sia l'istruzione stessa che i dati da essa referenziati.

Architettura Harvard

L'Architettura Harvard, chiamata così in riferimento all'Università di Harvard, utilizza una memoria separata per le istruzioni e una per i dati. Questo approccio consente l'accesso simultaneo e indipendente alle istruzioni e ai dati.

Nell'Architettura Harvard, quando eseguiamo "ldr r0, adr_var1," il processore preleva l'istruzione dalla memoria delle istruzioni e il dato dalla memoria dei dati in parallelo.

Ora, confrontiamo le due architetture chiave in termini di separazione delle memorie, flusso di istruzioni e complessità di controllo.

Modello di Von Neumann

Il Modello di Von Neumann è uno dei modelli di architettura dei computer più diffusi. Vediamo ora come funziona nel dettaglio:

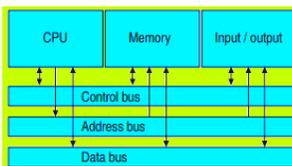
Supponiamo che il processore esegua l'istruzione ``ldr r0, adr_var1``:

- Il processore preleva l'istruzione dalla memoria, dove si trova ``ldr r0, adr_var1``.
- Successivamente, il processore decodifica l'istruzione e identifica che deve caricare il valore da un indirizzo di memoria denominato ``adr_var1`` nel registro ``r0``.
- Il processore accede quindi alla memoria, utilizzando l'indirizzo contenuto in ``adr_var1``.
- Il valore situato all'indirizzo specificato (il valore di ``adr_var1``) viene quindi letto dalla memoria dei dati e caricato nel registro ``r0``.

Il processo è simile quando il processore esegue l'istruzione ``ldr r1, adr_var2``.

In questo modello, entrambe le istruzioni seguono un flusso di esecuzione sequenziale, e il processore deve accedere alla memoria due volte: una volta per leggere l'istruzione e una volta per leggere i dati.

Nella memoria sono caricati sia i dati che il programma.



Modello di Harvard

Nel Modello di Harvard, il flusso di istruzioni e il flusso di dati sono gestiti separatamente. Vediamo ora come funziona questo modello:

Supponiamo che il processore esegua l'istruzione ``ldr r0, adr_var1``:

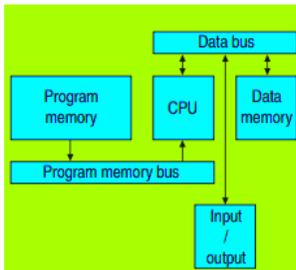
- Il processore preleva l'istruzione ``ldr r0, adr_var1`` dalla memoria delle istruzioni. Questa memoria contiene solo istruzioni e non dati.
- Dopo aver decodificato l'istruzione, il processore determina che deve caricare il valore da un indirizzo di memoria denominato ``adr_var1`` dalla memoria dei dati.
- Il processore accede quindi alla memoria dei dati, utilizzando l'indirizzo contenuto in ``adr_var1``,

per leggere il valore e caricarlo nel registro `r0`.

- Il processore mentre aspetta il dato letto dalla memoria dati potrebbe velocizzare l'esecuzione e leggere la prossima istruzione da eseguire nella memoria delle istruzioni.

Il processo è simile quando il processore esegue l'istruzione `ldr r1, adr_var2`.

In questo modello, le istruzioni e i dati vengono gestiti separatamente, consentendo al processore di accedere alle istruzioni e ai dati in parallelo. Questo approccio può portare a prestazioni migliori in termini di velocità di esecuzione.



(CC BY-NC-SA 3.0) lezione - by tankerino.com

<https://www.tankerino.com>

Questa lezione e' stata realizzata grazie al contributo di:



Risorse per la scuola

<https://www.baobab.school>



Siti web a Varese

<https://www.francescobelloni.it>