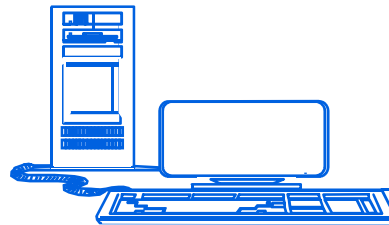


Architetture Parallele

**CALCOLATORI
ELETTRONICI LM**



Architettura

- **Architettura:** comportamento funzionale di un computer. Esempio: un processore sul quale si possa eseguire il codice DLX
- **Implementazione:** rete logica che realizza l'architettura. E' detta anche microarchitettura. Esistono molte implementazioni di una stessa architettura. Esempi: DLX sequenziale e DLX pipelined
- **Realizzazione:** una realizzazione fisica dell'implementazione. Esistono molte realizzazioni fisiche della stessa realizzazione. Di un DLX pipelined possono esistere molte realizzazioni (esistono infinite realizzazioni di una stessa funzione logica) ma anche della stessa rete logica diverse realizzazioni tecnologiche

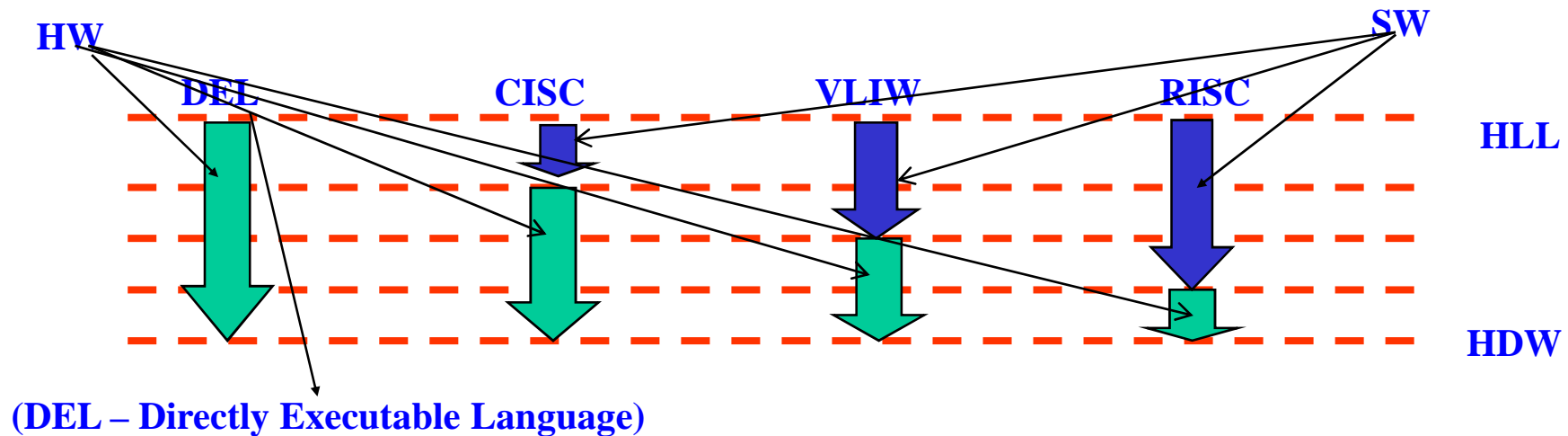
L'architettura è di fatto definita dal linguaggio macchina o meglio dal set di istruzioni (assembly language): **Instruction Set Architecture, ISA**

L'ISA tende a variare molto lentamente mentre varia rapidamente l'implementazione. Normalmente si hanno processi di accrescimento (IA8, IA16, IA32...). Più permane, più si sviluppano applicazioni e più è difficile sbarazzarsene ..

Dynamic Static Interface

Dove finisce il software e inizia l'hardware ?

Possibile posizionamento della Dynamic Static Interface



Più si sale più i compiti sono affidati all'hardware. Più si scende più i compiti sono affidati al compilatore. Nelle differenti implementazioni dello stesso ISA il livello di confine si sposta

Diverse architetture per parallelismo

Instruction level parallelism

- Sequenziali
Singola istruzione
- Pipelined
Istruzioni multiple contemporaneamente in esecuzione
- Superpipelined
*Più stadi per ogni singola operazione (EX, MEM etc.)
per aumentare la frequenza del clock (esempio Pentium IV)*
- Scalari
Una sola pipeline
- Superscalari
*Più pipelines; più istruzioni iniziate contemporaneamente
Possibile esecuzione fuori ordine (scelta a **run time**)*
- VLIW
*Più pipelines; più istruzioni iniziate contemporaneamente
Scelta dell'ordine a **compile time***
- Superscalari superpipelined
.....