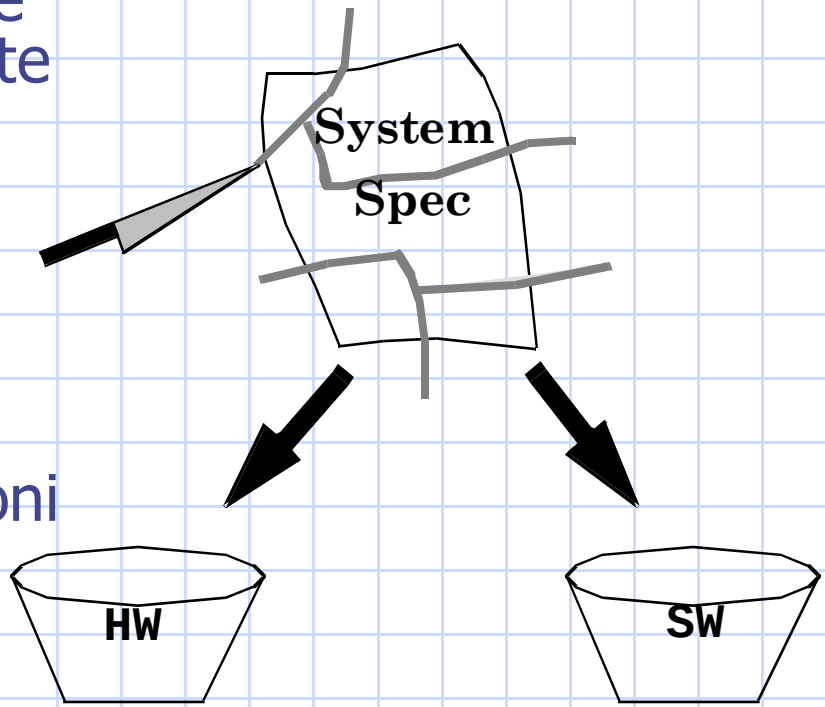
Introduzione a HW/SW codesign

Origini: sintesi dell' HW

- La sintesi logica risulta essere relativamente assestata a partire dagli anni 90
- Flusso di progetto:
  - algoritmo => FSM => RTL=> equazioni booleane => gate => transistor
- Sistemi complessi (embedded o SOC)
  - esempio: DSP + interfaccia di rete + CPU per la gestione di periferiche
  - richiesta di operare a livelli più alti

Metodologie tradizionali

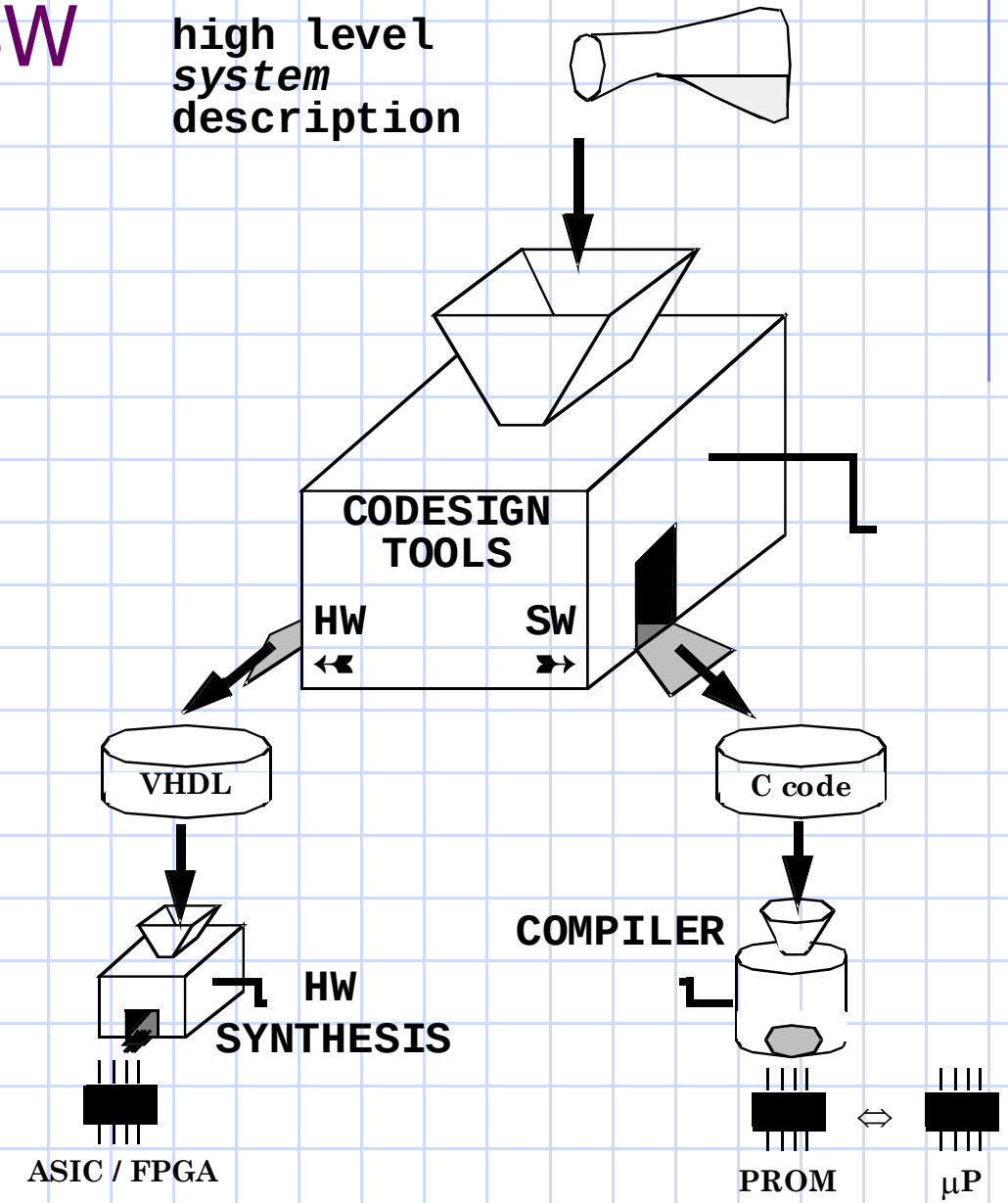
- Il partizionamento fra HW e SW viene deciso inizialmente su basi euristiche
- Difficoltà di integrazione
- Scarsa esplorazione dello spazio delle possibili soluzioni
- E' richiesto un approccio automatico



Progetti indipendenti

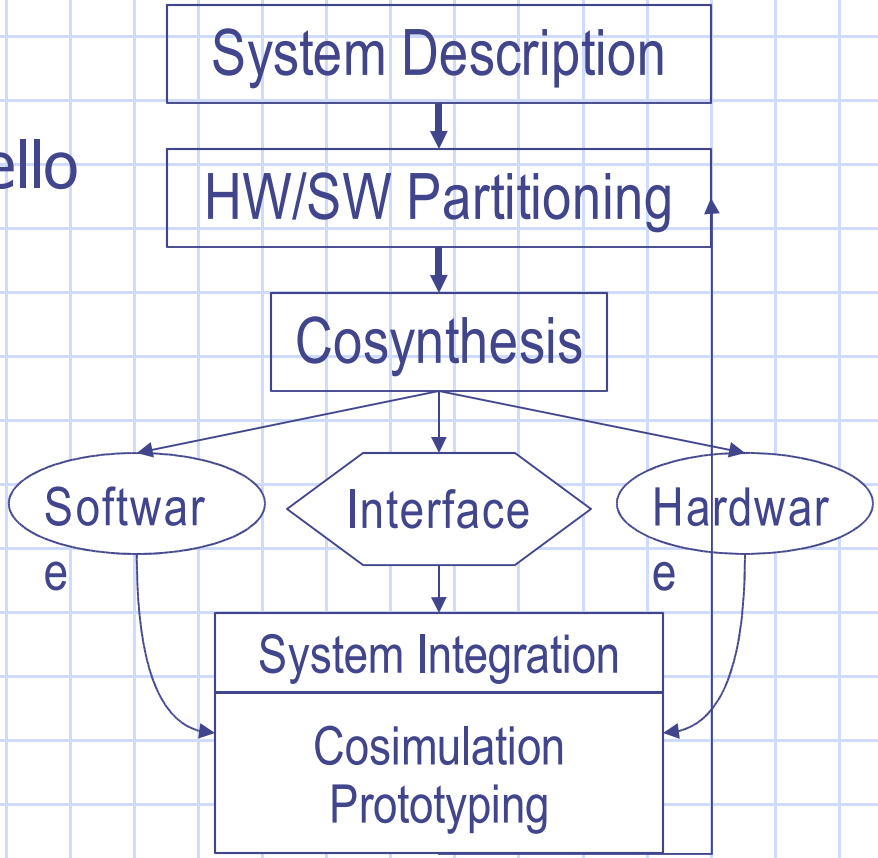
Concetti dell'HW/SW Codesign

Visione utopistica



Flusso di progetto

- Rende possibile una rapida valutazione dello spazio di progetto
- Aree di ricerca:
  - Cospecification
  - Partitioning
  - Cosynthesis
  - Cosimulation



Due categorie di sistemi

- Dominati dai dati
  - Transformano flussi di dati in ingresso in flussi di uscita (dataflow processors)
  - Digital signal processing (DSP)
    - DSP, ASIP e relativo supporto in termini di compilazione
  - Diagrammi di tipo dataflow (DFG)
- Dominati dal controllo
  - Sistemi reattivi: stimoli rari e risposte
  - Modellabili con STG
- Le soluzioni risultano essere specifiche al dominio applicativo

Sottoproblemi

- Cospecification
- Partitioning
- Cosynthesis
- Cosimulation

Cospecification

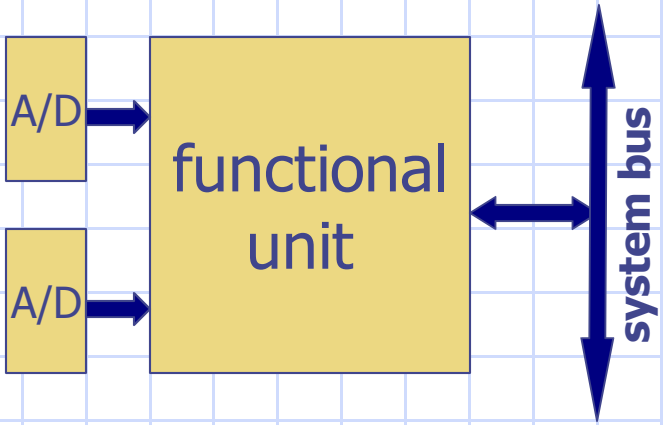
- Come modellare i sistemi?
  - Devono essere descritte specifiche funzionali ed algoritmiche ad alto livello
  - Sintassi e semantica devono aiutare il progettista e fornire un ingresso ai tool di CAD
- Linguaggi SW? (biased)
- Linguaggi HW? (biased)
- Linguaggi neutrali (algebra dei processi) ?
- Linguaggi formali ? (vantaggi nella verifica)
- Attualmente: un misto di linguaggi

Contengono già un modello computazionale implicito

Partitioning

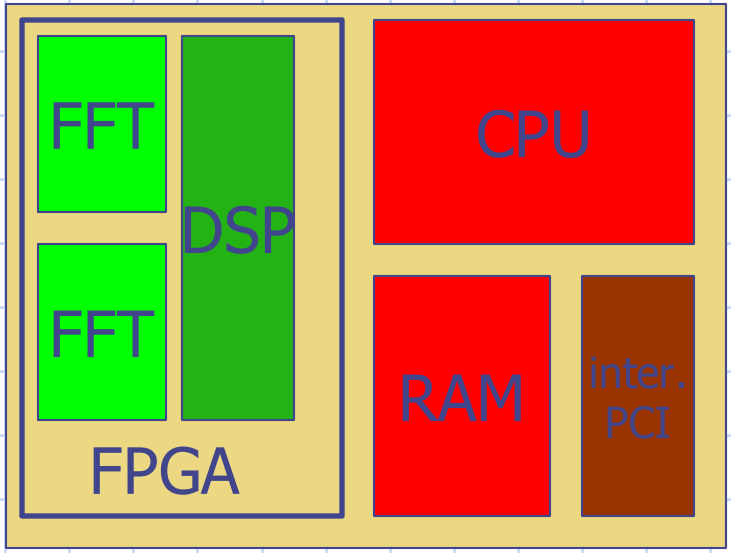
- Consiste nel determinare quali funzioni dvono essere realizzate in HW e quali in SW
  - il software è meno costoso, ma presenta minori prestazioni
- Vincoli di progetto
  - Timing, costo, power
- Si tratta di un problema di ottimizzazione
  - lo spazio delle soluzioni può essere troppo grande
- Approcci
  - Si inizia con tutto nel SW, spostando funzionalità nell'hardware fino a soddisfare i vincoli
  - Si inizia con tutto nell' HW, si spostano funzioni nel SW fino a che i vincoli non vengono violati
- E' evidente che entrambi gli approcci penalizzano un aspetto del problema e non garantiscono l'ottimalità della soluzione

Esempio



Compito dell'unità funzionale è di verificare costantemente alcune caratteritiche spettrali dei due flussi dati di ingresso, confrontarle con quelle rilevate in precedenza e segnalare eventuali eccezioni sul bus di sistema

- Si possono riconoscere diverse funzionalità:
  - FFT dei due flussi di ingresso e ulteriori operazioni in tempo reale (HW)
  - realizzazione dell'algoritmo di memorizzazione dei dati e di confronto (SW)
  - gestione dell'interfaccia con il bus (HW)



Cosynthesis

- Genera i componenti HW e SW
- Non dovrebbe essere separata dal partitioning
- Sintesi HW realizzata su tool di CAD esistenti
  - utilizzando linguaggi tipo il VHDL, Verilog
- Sintesi SW tipicamente in C
- Comunicazione fra HW e SW
  - Bus, reti supportati da protocolli
- Scheduling dei processi SW

Cosimulation

- Valutazione e verifica del progetto sintetizzato
- Simulazione interazioni fra HW e SW
  - correttezza funzionale
  - valutazione delle prestazioni
- Si può anche tornare al partitioning
- Tempi di simulazione molto elevati

Tendenze attuali

- Miglioramento della sintesi al livello behavioral (ad es. possibilià di tenere in conto processi concorrenti)
- Supporto al partitioning piuttosto che algoritmi ideali di partitioning
  - strumenti che consentano di valutare rapidamente le scelte del progettista
- SystemC
  - utilizzabile sia per l'hw che per il sw