

Introduzione al corso

M. Favalli

Engineering Department in Ferrara



Obbiettivi

Introdurre il progetto di sistemi digitali (ovvero basati su una codifica delle informazioni che utilizza un numero finito di valori)

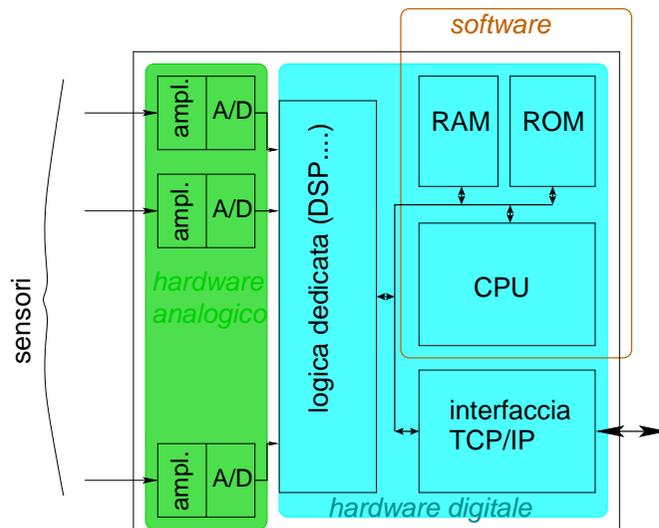
- Applicazioni
 - Calcolatori
 - Controllo
 - Telecomunicazioni
 -
- Specifiche:
 - funzionali
 - costo
 - prestazioni
 - consumo di potenza
 -

Tutti questi sistemi svolgono una funzione di "calcolo"

Per progettarli abbiamo bisogno di un modello matematico (come vedremo, si tratta di più modelli).

Esempio di sistema digitale

Invece di un esempio classico come un calcolatore, consideriamo un di sistema realizzabile con un unico circuito integrato che svolge applicazioni specifiche di calcolo



Come descrivere un sistema digitale

- Sistemi molto complessi che contengono milioni di dispositivi "elementari" ⇒ non é possibile utilizzare un singolo modello matematico per descriverli
- Gerarchia di livelli di descrizione che, in maniera molto semplificata, possiamo iniziare a descrivere procedendo dall'alto (utente) verso il basso (tecnologia)
 - livello software (se esiste)
 - livello hardware
- Entrambi i livelli si suddividono in una serie di sottolivelli
- Questa gerarchia consente di gestire la complessità in quanto a ciascun livello nascondiamo alcuni dettagli di quello sottostante ottenendo un modello più efficiente

- Come esempio di astrazione da un livello a quello superiore, consideriamo un caso riguardante il software
- Nel passaggio dal linguaggio assembly a un linguaggio di programmazione (es. C), si nascondono diversi dettagli:
 - allocazione delle variabili nei registri
 - il meccanismo di riferimento diretto a un'etichetta nelle istruzioni di salto condizionato
 - il protocollo di chiamata delle funzioni

- I livelli piú alti si identificano con le specifiche e quelli piú bassi con l'implementazione

Specifiche di sistema



Realizzazione fisica

Un esempio introduttivo

- Partendo da alcune conoscenze acquisite nel corso di Fondamenti di Informatica, si cercherà di inquadrare il ruolo delle reti logiche nel progetto dell'hardware
- Si assume una conoscenza di base dell'architettura (Von Neumann) di una semplice CPU e dei suoi componenti fondamentali (ALU, registri)
- Come si vedrà, si possono descrivere sistemi digitali piú generali di una CPU

Livelli dell'hardware: sistema - architettura

- Si descrive il comportamento (behavior) del sistema, che può essere molto complesso, mediante processi di elaborazione interagenti
- Si forniscono le specifiche su costi e prestazioni
- Si fanno le prime scelte decidendo quali parti del sistema possono essere realizzate via software (utilizzando una o piú CPU) e quali via hardware (reti dedicate)
- Le prestazioni determinano questo partizionamento: ciò che viene implementato direttamente via hardware è piú veloce di quanto realizzato via software

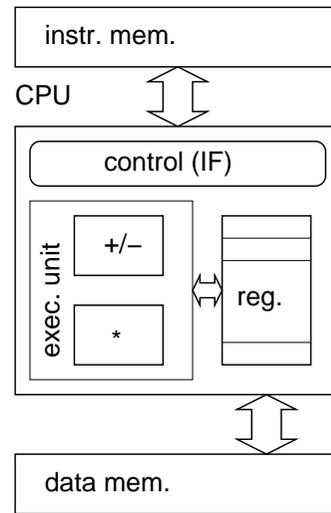
Esempio (semplice) di specifica

Un semplice sistema che calcola: $x = a * b + c^2 + d * e$ (dove a, b, c, d, e sono interi)

- Traduzione dell'espressione in un algoritmo (una sequenza di operazioni) e sua esecuzione mediante una semplice CPU (sw+hw)

```
u=a*b;
v=c*c;
u=u+v;
v=d*e;
u=u+v;
```

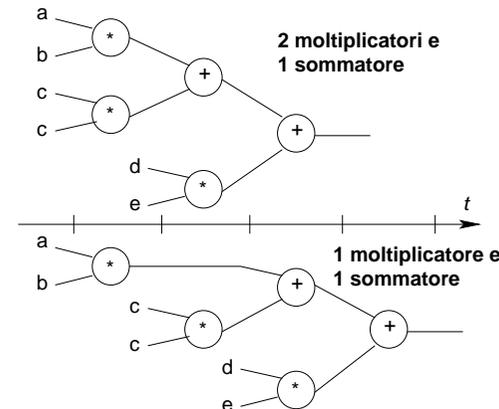
- Questo approccio, molto generale, può risultare inadeguato dal punto di vista delle prestazioni



- Sono possibili diverse soluzioni con costi e prestazioni diversi
- Per arrivare a una realizzazione bisogna porsi alcune domande:
 - che tipo e che numero di blocchi funzionali (sommatori, moltiplicatori ...) sono disponibili
 - quando si esegue una certa operazione e quale risorsa la esegue
 - quali sono i costi e le prestazioni di ciascuna risorsa utilizzata
- Come vedremo sono possibili diverse soluzioni

- L'hw dedicato non ha le limitazioni della CPU (sequenzialità nelle operazioni, numero limitato di risorse) rinunciando (in parte) alla programmabilità
- Realizziamo un componente dedicato supponendo di disporre di sottosistemi in grado di calcolare somma e prodotto, e di sottosistemi in grado di memorizzare risultati parziali
- Dall'espressione si può notare che è possibile eseguire in parallelo i diversi prodotti nel caso in cui siano disponibili più moltiplicatori
- Questo consente di calcolare il risultato in un tempo inferiore rispetto alla CPU

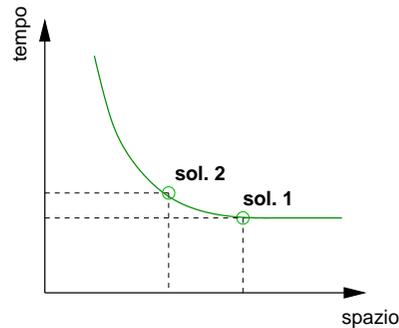
Confrontiamo due soluzioni: la prima (sol. 1) utilizza 2 moltiplicatori e la seconda (sol. 2) 1 moltiplicatore.



Come si vede dal grafo che illustra le operazioni eseguite nel tempo (nell'ipotesi che ciascuna operazione impieghi un "intervallo" di tempo) la prima soluzione è più veloce ma è anche più costosa

Servirebbero 3 moltiplicatori? quanti intervalli di tempo impiegava la CPU?

- Si tratta di un aspetto ricorrente nel progetto dell'hardware (non solo ...)
- Lo spazio si identifica con il costo (nelle tecnologie attuali l'area occupata da un sottosistema é proporzionale al suo costo)
- Per tempo si intende quello necessario per calcolare i risultati (prestazioni)
- Non esiste quasi mai una soluzione che minimizzi sia lo spazio che il tempo, si hanno diverse soluzioni con diversi compromessi fra costi e prestazioni



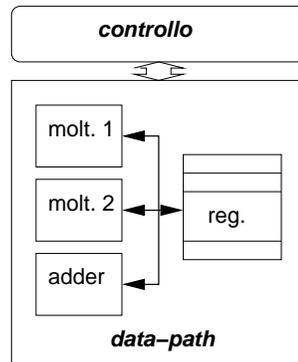
Livelli dell'hardware: RTL

Register Transfer Level

É il livello al quale sono descritte in dettaglio le scelte di progetto fatte al livello superiore:

- quale sottosistema (sommatore, moltiplicatore) esegue un'operazione e quale ne esegue un'altra
- dove memorizzare i risultati parziali
- come controllare che le operazioni si svolgano in maniera corretta
- come interconnettere fra loro questi sottosistemi

Vediamo la soluzione 2



```

molt. 1  molt. 2  adder
-----
r1:=a*b; r2:=c*c;
-----
r3:=d*e;          r1:=r1+r2;
-----
                      r1:=r1+r3;
-----
    
```

Come esempio consideriamo ancora il caso del software con due funzioni che eliminano gli elementi ripetuti da un vettore ordinato

```

int f1(int *v, int n, int *vout)
{
int i, *vout, nout;

vout=(int *)malloc(n*sizeof(int));
nout=0;
for (i=0; i<n; i++)
    if ((i==0) || (v[i]!=v[i-1]))
        {
            vout[nout]=v[i];
            nout++;
        }
return nout;
}
    
```

```

int f2(int *v, int n, int *vout)
{
int i, *vout, nout;

nout=0;
for (i=0; i<n; i++)
    if ((i==0) || (v[i]!=v[i-1]))
        nout++;
vout=(int *)malloc(nout*sizeof(int));
nout=0;
for (i=0; i<n; i++)
    if ((i==0) || (v[i]!=v[i-1]))
        {
            vout[nout]=v[i];
            nout++;
        }
return nout;
}
    
```

Livelli dell'hardware: logico - gate

- Un ulteriore problema é quello di realizzare:
 - 1 i componenti che svolgono le operazioni (sommatori, moltiplicatori) e le unità che contengono i risultati parziali (data-path)
 - 2 il sottosistema che si occupa di eseguire le varie operazioni nella sequenza corretta (controllo)
 - 3 le interconnessioni fra questi sottosistemi
- Questi sistemi devono essere progettati mediante i blocchi elementari (gate) che la tecnologia ci mette a disposizione
- **Eeguire queste operazioni in modo da creare sistemi funzionalmente corretti, che soddisfino criteri di costo e prestazioni é il ruolo del livello logico di progetto**
- Al di sotto di questo livello entrano in gioco livelli tecnologici in cui compaiono le caratteristiche fisiche dei componenti elementari

Interfaccia fra un mondo architetturale (fino al livello RTL) indipendente dalla tecnologia e un mondo tecnologico che porta alla realizzazione fisica del sistema.

Informazioni dai livelli piú alti

specifiche di progetto e obiettivi di progetto

↓
Analisi e sintesi dei circuiti digitali al livello logico
↑

Informazioni dai livelli "tecnologici"

caratteristiche dei componenti elementari che vengono utilizzati per costruire le reti logiche

Modello di un sistema di calcolo

Il modello piú generale di un qualsiasi sistema di calcolo é quello di funzione: $f : I \rightarrow O$ dove I é l'insieme dei possibili ingressi e O quello delle possibili uscite.



- descrizione di tipo **comportamentale** (ci dice cosa fa, ma non come lo fa)
- ci sono molte cose non specificate, come ad esempio il modo in cui vengono forniti gli elementi di I e O

Determinismo

Cosa succederebbe se invece di una funzione avessi una relazione? Se allo stesso elemento di I corrispondono piú elementi di O , si ha un comportamento non deterministico

- Tecnologie che ci consentono di realizzare un sistema digitale
- Requisiti generali di queste tecnologie
- Bisogna prima specificare cosa intendiamo per sistema di calcolo digitale

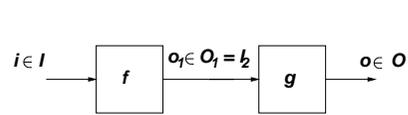
Sistemi di calcolo non digitali

Un processo di calcolo puó essere realizzato in molti modi e non necessariamente mediante tecnologie digitali. Ad esempio, le reti neurali che simulano in maniera semplificata il comportamento del cervello possono essere di tipo non digitale e consentono di risolvere problemi complessi come il riconoscimento di caratteri

Modelli comportamentali e strutturali

- Un modello di tipo comportamentale (behavioral) descrive il sistema come una "scatola nera"
- Si parla invece di descrizione **strutturale** quando un sistema viene descritto come insieme di componenti interconnessi (in questo caso si specifica come é fatto il sistema invece del cosa fa)
- Nel caso strutturale la funzione svolta dal sistema puó essere dedotta a partire dalle funzioni dei singoli componenti e dal modo in cui sono interconnessi
- Le descrizioni dei singoli componenti possono essere sia di tipo strutturale (gerarchico) che comportamentale

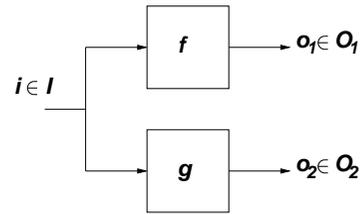
Connessione in cascata



$$\begin{aligned} f &: I_1 \rightarrow O_1 \\ g &: I_2 \rightarrow O_2 \\ I &= I_1 \quad I_2 = O_1 \end{aligned}$$

$$O = O_2 = g(f(I))$$

Connessione in parallelo

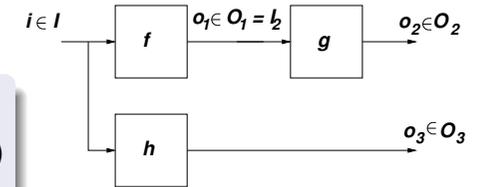


$$\begin{aligned} f &: I_1 \rightarrow O_1 \\ g &: I_2 \rightarrow O_2 \\ I &= I_1 = I_2 \end{aligned}$$

$$O = O_1 \cup O_2 = f(I) \cup g(I)$$

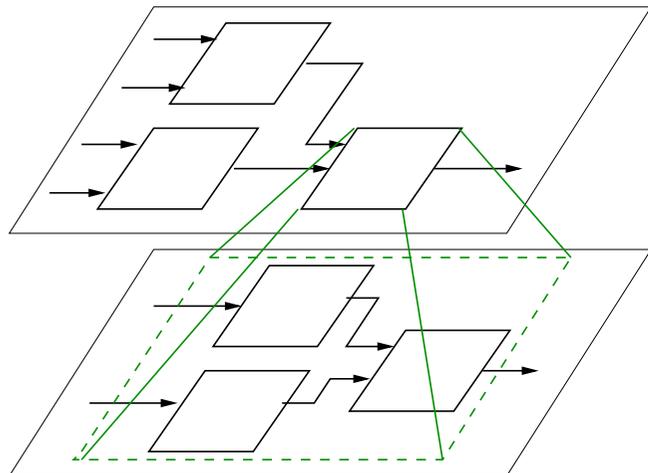
$$\begin{aligned} f &: I \rightarrow O_1 \\ g &: I_2 = O_1 \rightarrow O_2 \\ h &: I \rightarrow O_3 \end{aligned}$$

$$O = O_2 \cup O_3 = g(I_2) \cup h(I) = g(f(I)) \cup h(I) \quad (1)$$



Descrizioni di tipo gerarchico

Si tratta di un accorgimento utilizzato per gestire la complessità



Realizzazione

Una tecnologia deve mettere a disposizione:

- 1 Supporto fisico per le informazioni: gli elementi di I e O devono essere rappresentati tramite una qualche grandezza fisica
- 2 Supporto fisico per l'elaborazione: un sistema fisico in grado di elaborare le grandezze di ingresso producendo i valori di uscita

Nota

Consistenza fra il tipo di grandezze utilizzato per rappresentare ingresso e uscita e il sistema fisico utilizzato per elaborarle

Sistema fisico	Grandezze fisiche utilizzate per rappresentare l'informazione
Calcolatrice meccanica (basata su ingranaggi)	Posizioni angolare
Centrale elettromeccanica di commutazione	Posizioni angolari dei selettori e correnti
Calcolatore elettronico basato su circuiti integrati	Valori di tensione alle interconnessioni
Calcolatore elettronico basato su Quantum Cellular Automata	Posizione di una coppia di elettroni