

Esercitazione 11

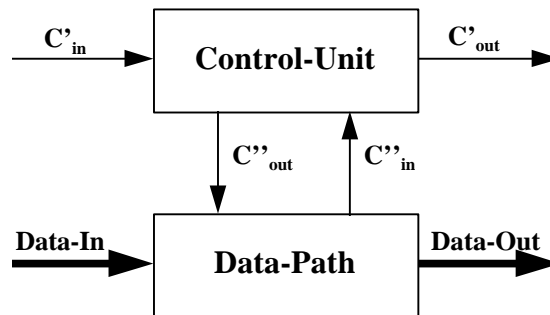
Sommario

- Unità di controllo cablate

1. Unità di controllo

L'architettura interna di una CPU può essere modellata attraverso una struttura costituita da 2 unità interagenti:

- percorso dati (data path): contiene i registri, le unità aritmetico-logiche, le memorie, ecc;
- unità di controllo: genera i segnali di controllo per il data-path, sulla base dei segnali provenienti dal data-path stesso e dall'esterno.



La Control Unit è una macchina a stati che riceve segnali di Controllo dall'esterno (C'_{in}) e dal Data-Path (C''_{in}). Sulla base del valore di tali segnali e dello stato corrente essa produce i segnali di controllo per l'esterno (C'_{out}) e per il Data-Path (C''_{out}).

La complessità della Control-Unit è quindi proporzionale al prodotto tra il numero di segnali di controllo (dall'esterno e dal Data-Path) ed il numero di variabili di stato.

Segnali C'_{out} sono segnali generati dall'Unità di Controllo ed agiscono sul Data Path (Percorso Dati).

Segnali C'_{in} sono relativi ai risultati che modificano l'evoluzione successiva delle operazioni (flag, risultati parziali di una determinata operazione). Effetto di reazione da parte dei dati sull'unità di controllo in funzione dei risultati parziali ottenuti.

Segnali C'_{in} e C'_{out} sono i segnali di controllo che agiscono sull'esterno, sono segnali che possono comandare l'inizio delle operazioni ed indicare ad un'unità esterna che le operazioni sono terminate.

1.1. Unità di controllo in una CPU

Una CPU è formata da:

- un'unità di esecuzione;
- un'unità di controllo che agisce a 2 livelli di astrazione diversi:

- controllo della sequenza delle istruzioni;
- controllo dell'esecuzione delle istruzioni, ogni istruzione scatena una sequenza di comandi che nel loro insieme svolgono la singola istruzione.

Per ogni istruzione c'è un insieme di micro-istruzioni che insieme permettono di eseguire l'istruzione.

1.1.1. Unità di controllo dell'esecuzione delle istruzioni

Esistono due tecniche di implementazione dell'unità di controllo dell'esecuzione delle istruzioni:

- controllo cablato (*hardwired*): l'unità di controllo viene considerata come una normale macchina a stati a cui applicare i metodi tradizionali di progetto, oppure metodi *ad hoc*;
- controllo a microprogramma: ogni operazione che l'unità di controllo deve eseguire viene descritta da una microistruzione; le microistruzioni sono immagazzinate in una memoria, in maniera analoga a quanto accade per i programmi.

Un controllo cablato è un circuito sequenziale speciale che realizza la funzione desiderata.

In un controllo a microprogramma, le microistruzioni contengono direttamente le indicazioni per l'attivazione dei segnali: ogni istruzione scatenerà un sottoprogramma a livello di insieme di microistruzioni ed all'interno dell'unità di elaborazione saranno contenuti tutti i sottoprogrammi che definiscono lo svolgimento delle singole istruzioni.

Il contrasto tra queste due tecniche implementative è molto forte.

Nel caso di controllo cablato occorre progettare un circuito sequenziale particolare. L'approccio è meno costoso e più conveniente per sistemi di limitata complessità mentre è di difficile progettazione per sistemi complessi. L'unità di controllo cablato è tipico delle CPU RISC.

Nel caso di unità di controllo a microprogramma si può effettuare in modo concorrente la progettazione della parte operativa e dell'unità di controllo. La progettazione della struttura di controllo è strutturata. La facilitazione nel meccanismo progettuale si paga con una complessità aggiunta nell'hardware (è necessaria un numero di porte superiore ed un'area superiore di silicio). Complessità superiore comporta una minore velocità di esecuzione delle operazioni. L'unità di controllo a microprogramma è tipico delle CPU CISC.

2. Tecniche di progettazione dei controlli cablati

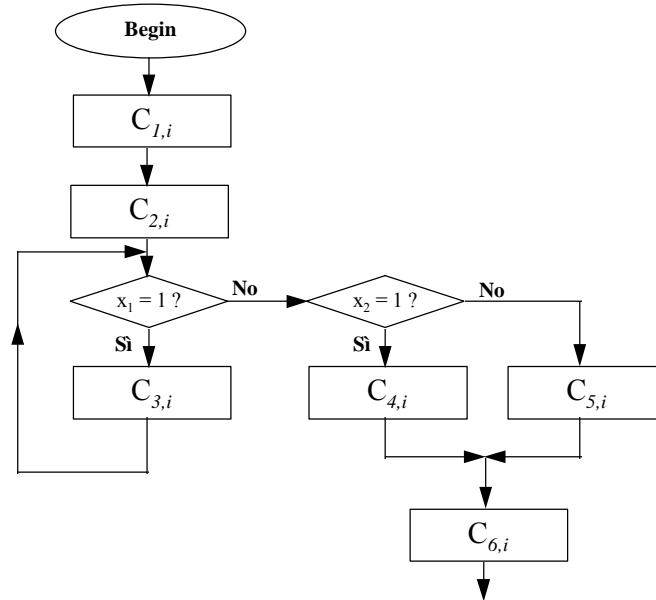
Metodi di progetto:

- generatori di sequenze basati su:
 - elementi di ritardo
 - contatori;

- macchina a stati finiti (progetto tradizionale di circuiti sequenziali).

Punto di partenza per la progettazione dell'unità di controllo è il *diagramma di flusso* che definisce il comportamento della parte operativa.

A partire dal diagramma di flusso si deve generare una sequenza temporale di vettori (*insieme di segnali di controllo*) $c_{1j}, c_{2j}, c_{3j}, c_{4j}, \dots$ a tempi definiti $t_1, t_2, t_3, t_4, \dots$, che agiscono sull'unità di percorso dati. La sequenza sarà in parte regolata dai risultati parziali delle operazioni.



I primi vettori generati corrispondono ai vettore c_1 e c_2 . Il terzo vettore dipende dai segnali x_1 ed x_2 che derivano dall'unità del percorso dei dati (sono risultati parziali delle operazioni). Tale esempio mostra l'interazione tra l'unità di controllo e l'unità di percorso dati.

2.1. Progetto di un'unità di controllo quale circuito sequenziale

L'unità di controllo è un circuito sequenziale tradizionale, può essere progettato con le tecniche già viste a partire da una tabella di commutazione tra stati.

Tabella di commutazione tra stati descrive l'evoluzione degli stati del sistema a partire dai valori degli ingressi. Ad ogni combinazione stato presente-valore degli ingressi occorre precisare lo stato futuro ed il valore dei segnali di controllo.

2.2. Progetto basato su elementi di ritardo

Questa tecnica si basa sull'uso di un numero di Flip Flop pari al numero di stati della Unità di Controllo: ogni Flip Flop contiene il valore 1 solo quando l'Unità di Controllo si trova nello stato corrispondente.

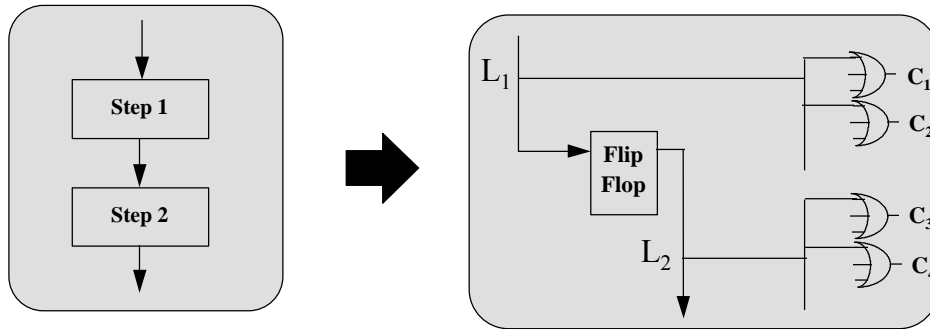
Il passaggio da uno stato al successivo avviene tramite il passaggio del valore 1 da un Flip Flop al successivo.

Il valore di ciascun Flip Flop viene utilizzato come abilitazione per i segnali di controllo che devono essere attivati in un certo stato.

È possibile passare in maniera automatica dal flow-chart dell'Unità di Controllo al corrispondente hardware. Il problema è quella di costruire una struttura circuitale che genera la stessa sequenza di comandi generati dal diagramma di flusso.

Si analizzano gli elementi base che occorre realizzare in hardware per permettere l'implementazione del diagramma di flusso.

2.2.1. Sequenze

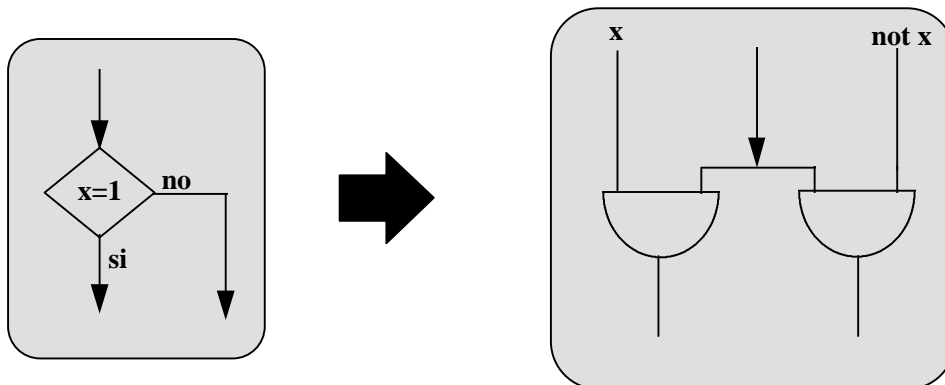


Al tempo t_1 la linea L_1 assume valore logico 1, tale linea porta (al tempo t_1) il valore logico 1 su una serie di segnali di controllo. Dopo un certo ritardo la linea L_2 assume il valore 1.

I segnali di controllo sono collegati a porte logiche C_i . Se al tempo t_i il segnale C_i deve assumere il valore logico 1, si collega la linea L_i al segnale C_i .

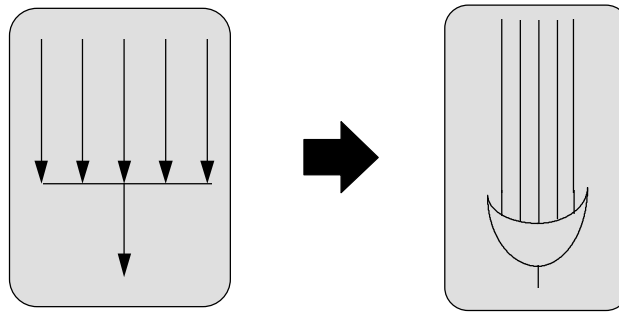
Per realizzare un'Unità di Controllo cablata utilizzando gli elementi di ritardo ci vogliono tanti elementi di ritardo quanti sono il numero di sequenze complessive.

2.2.2. Condizioni



Il flusso di controllo si trasferisce in base al valore della variabile x .

2.2.3. Convergenza

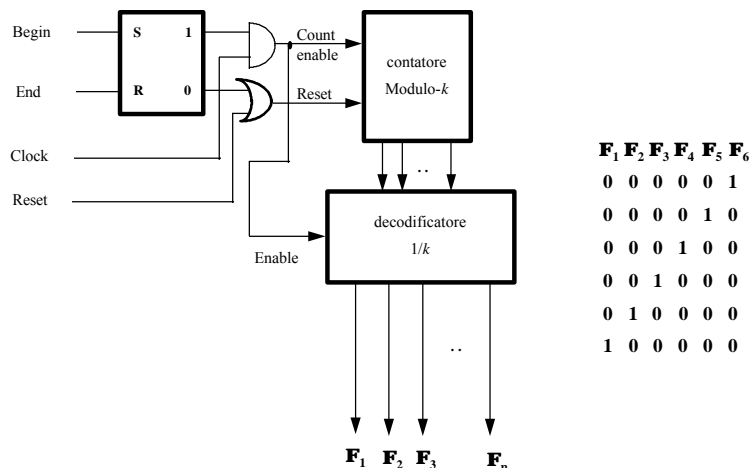


La convergenza viene realizzata attraverso una porta logica OR.

2.3. Progetto basato su contatori

Il metodo è particolarmente adatto per implementare un'Unità di Controllo il cui flow-chart sia privo di condizioni e sia basato sulla ripetizione di una stessa serie di operazioni. Tale caso si presenta ad esempio nell'unità di Controllo di una CPU.

Attraverso un generatore di sequenze vengono generati impulsi su linee diverse (*Fasi F_i*), ciascuno in corrispondenza di un diverso colpo di clock. Tali segnali attivano gli opportuni segnali di controllo.



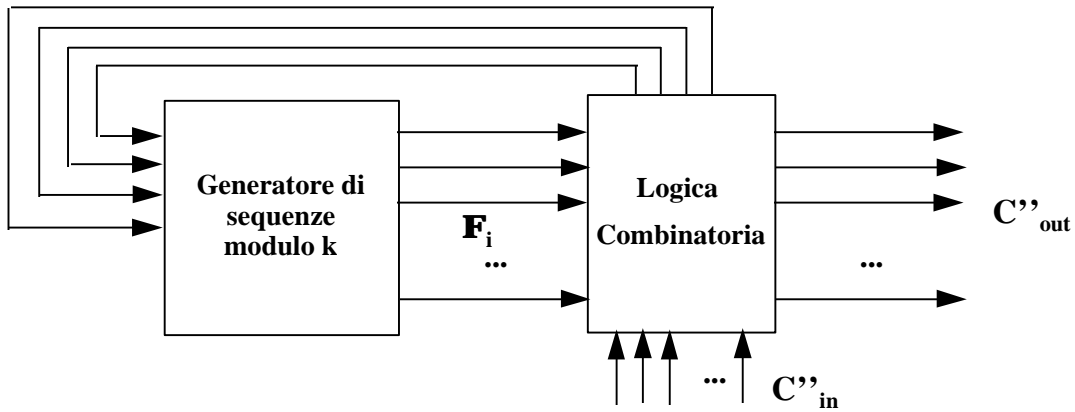
In un decodificatore $1/k$, una sola uscita assume il valore 1 in funzione della parola di ingresso. Attraverso un contatore modulo k ed un decodificatore $1/k$ è possibile realizzare un generatore di sequenza modulo k . La presenza di un valore logico alto si propaga lungo le varie fasi.

I segnali di fase sono generati attraverso un contatore modulo k ed un decodificatore $1/k$.

Il contatore è pilotato da un segnale di clock. Il segnale di *begin* permette di attivare il contatore. I segnali di *reset* e di *end* interrompono il conteggio.

A partire dai segnali di fase occorre generare i comandi all'unità di percorso dati. In taluni casi i segnali di controllo da attivare in corrispondenza di una certa fase dipendono anche da segnali di controllo esterni (C'' in).

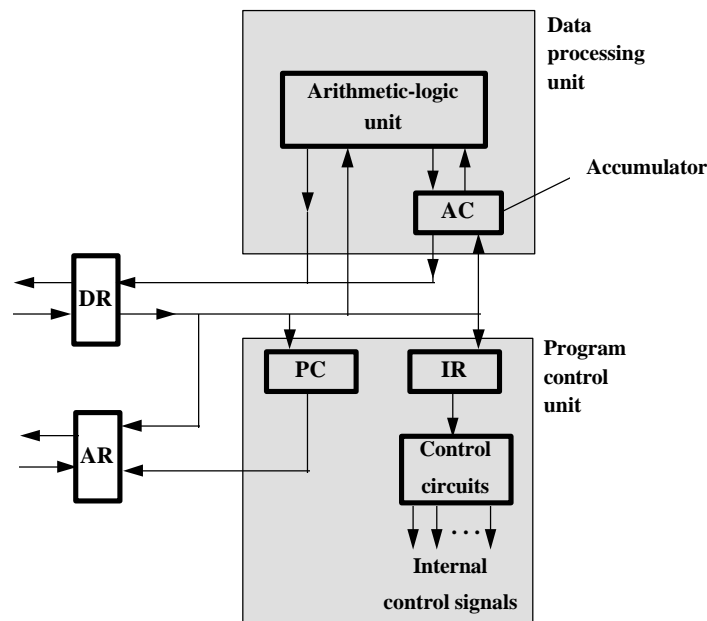
Viene allora inserita una logica aggiuntiva di selezione.



Il generatore di sequenza modulo k genera i segnali di fase che vanno in ingresso ad un circuito combinatorio che permette di generare i segnali C''_{out} che pilotano il percorso dati. Il circuito combinatorio genera alcuni segnali che possono modificare la sequenza dei segnali di fase. I segnali C''_{in} sono generati dal percorso dati in funzione dei risultati parziali e sono segnali di ingresso al circuito combinatorio.

2.4. Unità di controllo di una CPU

Si assuma di considerare una CPU semplificata il cui schema è il seguente:

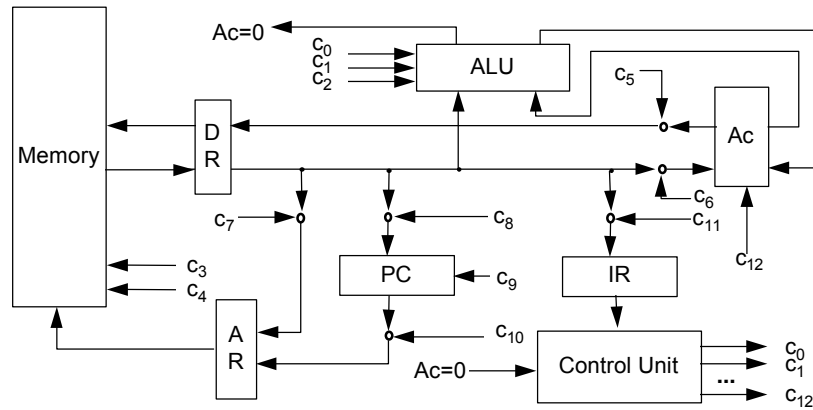


Si consideri una semplice CPU avente le seguenti caratteristiche:

- 8 istruzioni ad un operando:
 - LOAD X ; $AC \leftarrow M(X)$
 - STORE X ; $M(X) \leftarrow AC$
 - ADD X ; $AC \leftarrow AC + M(X)$
 - AND X ; $AC \leftarrow AC \wedge M(X)$
 - JUMP X ; $PC \leftarrow X$
 - JUMPZ X ; If $AC = 0$ then $PC \leftarrow X$

- COMP ; $AC \leftarrow \wedge AC$
- RSHIFT ; Scorrimento a destra di AC

- unità aritmetica svolge le funzioni di ADD, AND e CMP.



I segnali di controllo permettono di regolare il funzionamento della CPU.

- c_0 $AC \leftarrow AC + DR$
- c_1 $AC \leftarrow AC \wedge DR$
- c_2 $AC \leftarrow AC$
- c_3 $DR \leftarrow M(AR)$ %% lettura M
- c_4 $M(R) \leftarrow DR$ %% scrittura M
- c_5 $DR \leftarrow AC$
- c_6 $AC \leftarrow DR$
- c_7 $AR \leftarrow DR$
- c_8 $PC \leftarrow DR$
- c_9 $PC \leftarrow PC + 1$
- c_{10} $AR \leftarrow PC$
- c_{11} $IR \leftarrow DR$
- c_{12} scorrimento a destra AC

L'esecuzione di un'istruzione può essere suddivisa in una serie di passi elementari:

- trasferimento del PC in AR;
- lettura dell'istruzione dalla memoria (*fetch*);
- incremento di PC e decodifica dell'istruzione;
- indirizzo di operando in AR;
- lettura dell'operando dalla memoria;

- svolgimento dell'operazione.

I primi 3 passi sono comuni a tutte le istruzioni. I restanti passi variano da istruzione ad istruzione. Lettura e scrittura richiedono una durata maggiore (2 unità di tempi). In totale si avrà una sequenza di 6 passi per un totale di 8 unità di tempo.

Si vede ora nel dettaglio gli specifici passi che devono essere effettuati da alcune istruzioni.

2.4.1. Istruzione LOAD

- $AR \leftarrow PC$
- $DR \leftarrow M(AR)$; Lettura dell'istruzione
- $PC \leftarrow PC + 1$, IR DR(OP), Decodifica OP
- $AR \leftarrow DR(ADR)$
- $DR \leftarrow M(AR)$; Lettura del dato
- $AC \leftarrow DR$

2.4.2. Istruzione ADD

- $AR \leftarrow PC$
- $DR \leftarrow M(AR)$; Lettura dell'istruzione
- $PC \leftarrow PC + 1$, IR DR(OP), Decodifica OP
- $AR \leftarrow DR(ADR)$
- $DR \leftarrow M(AR)$; Lettura del dato
- $AC \leftarrow DR + AC$

Cambia unicamente l'ultimo passo.

2.4.3. Istruzione JUMP

- $AR \leftarrow PC$
- $DR \leftarrow M(AR)$; Lettura dell'istruzione
- $PC \leftarrow PC + 1$, IR DR(OP), Decodifica OP
- $PC \leftarrow DR(ADR)$

Viene comunque incrementato il contatore di programma al terzo passo e poi viene caricato il PC con il valore a cui si deve saltare dopo aver decodificato l'istruzione.

2.4.4. Progetto dell'unità di controllo

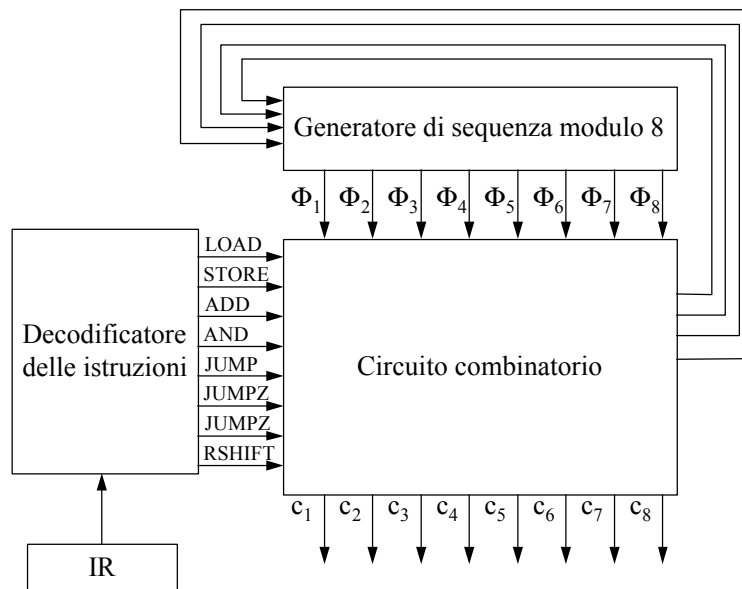
Si suddivide l'esecuzione di ogni singola istruzione in 8 fasi. Dal diagramma di flusso di controllo si ricavano i segnali da attivare per ogni istruzione in ogni fase.

Ogni segnale è definito da una particolare funzione in base alla fase corrente ed al tipo di istruzione. Esempio (C_3 : $DR \leftarrow M(AR)$):

$$C_3 = F_2 + F_6 (\text{LOAD} + \text{ADD} + \text{AND})$$

Ogni segnale di controllo è definito per ogni fase in corrispondenza delle istruzioni in cui è necessario attivarlo.

Tutti i segnali di controllo possono essere descritti da un'espressione generale di questo genere: $C_j = \sum_j (F_j \sum_m I_m)$.



Segnali di fase e le uscite del decodificatore delle istruzioni sono gli ingressi del circuito combinatorio secondo le funzioni specifiche di ogni singolo segnale di controllo.

Il progetto dell'unità di controllo cablata si riduce ad un progetto di un circuito combinatorio.