

Realizzazione del controllo

La generazione della corretta sequenza di segnali di controllo per l'esecuzione di una data istruzione avviene da parte dell'Unità di Controllo.

La sequenza generata è funzione:

- dell'istruzione (codice operativo, operandi, modi d'indirizzamento)
- dei flag
- di segnali di stato (MFC, ...)

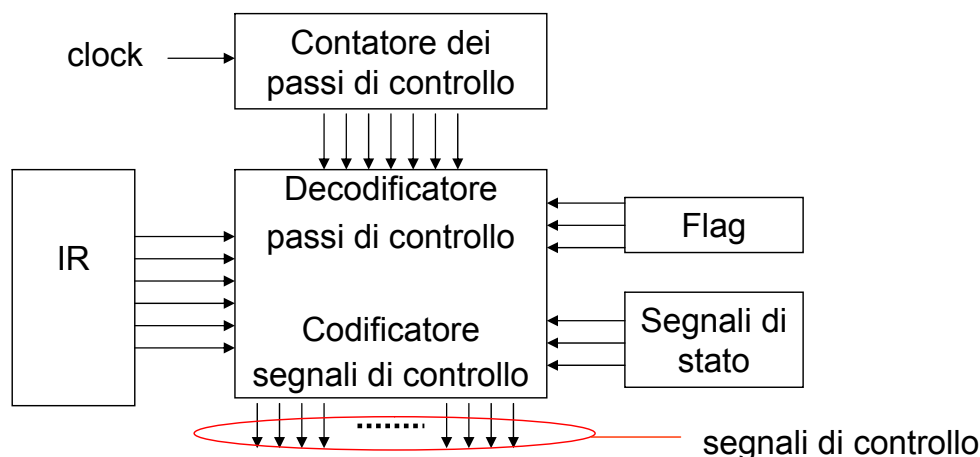
Due possibili tecniche per la gestione del controllo:

- **controllo cablato**
- **controllo microprogrammato**

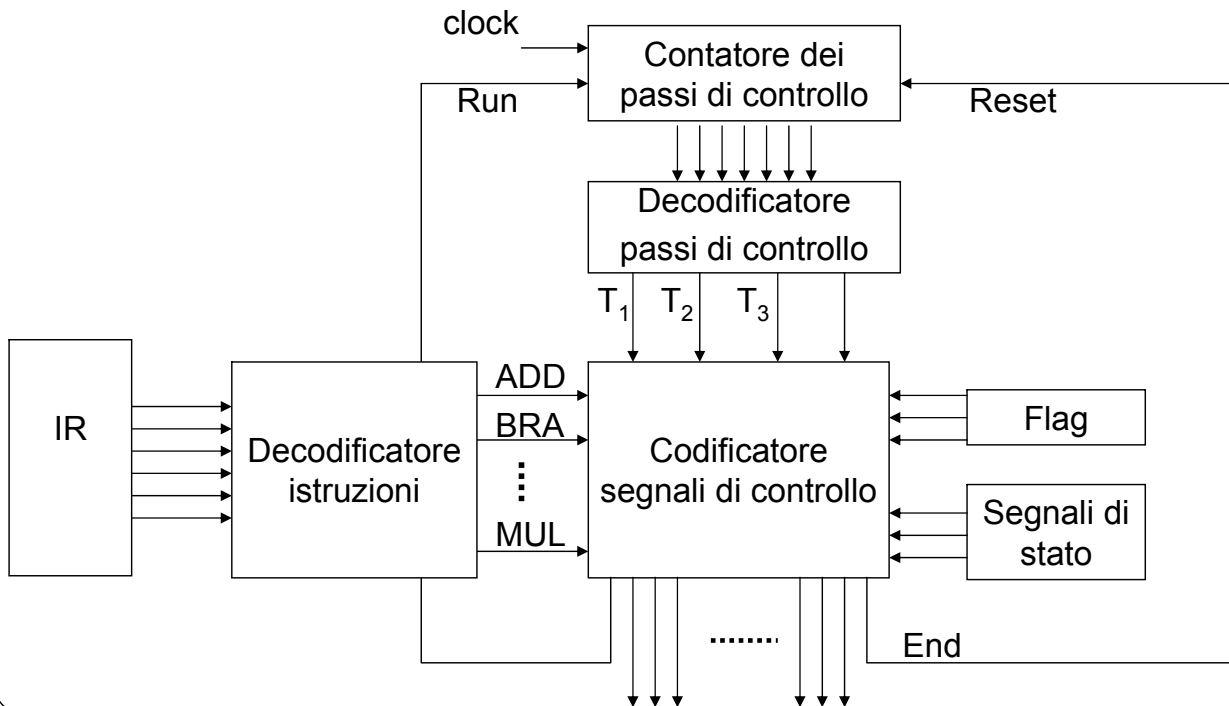
Controllo cablato

Da un punto di vista logico, l'unità di controllo può essere realizzata con una **rete sequenziale** che ha:

- **stati** coincidenti con i passi da realizzare per eseguire l'istruzione
- **uscite** coincidenti con i segnali di controllo e funzioni dello stato corrente, dell'istruzione, dei flag e di segnali di stato



Rete di controllo (1/2)



F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi
di Cassino

Rete di controllo (2/2)

Realizzazione dei segnali di controllo

$$\text{Zin} = T_1 + T_6 * \text{ADD} + T_5 * \text{BMI} + \dots$$

$$\text{PCin} = T_2 + T_6 * \text{N} * \text{BMI} + \dots$$

$$\text{End} = T_7 * \text{ADD} + (T_4 * \text{N} + T_6 * \text{N}) * \text{BMI} + \dots$$

- Il segnale End opera il reset sul contatore dei passi di controllo, facendo ripartire il conteggio dal primo passo (inizio instruction fetch).
- Il segnale Run abilita la lettura del clock da parte del contatore e quindi la sequenza dei passi. Se posto a 0, trattiene indefinitamente il contatore nello stato presente: in tal modo si può gestire il colloquio con l'unità di memoria.

F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi
di Cassino

Controllo microprogrammato

“... I realized that the solution was to turn the control unit into a computer in miniature by adding a second (diode) matrix to determine the flow of control at the microlevel and by providing for conditional micro-instructions.” (Wilkes, 1953)

Gli insiemi dei segnali di controllo che vengono attivati in ogni passo di controllo possono essere organizzati in una struttura formata da una *parola di controllo* (**control word**), i cui bit forniscono i valori dei segnali.

	PCin	Pcout	MARin	Read	MDRin	MDRout	IRin	Yin	Clear Y	Carry in	Add	Zin	Zout	R1in	R1out	R3in	R3out	WMFC	End
1	0	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
3	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0
6	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

F. Tortorella

Corso di Calcolatori Elettronici

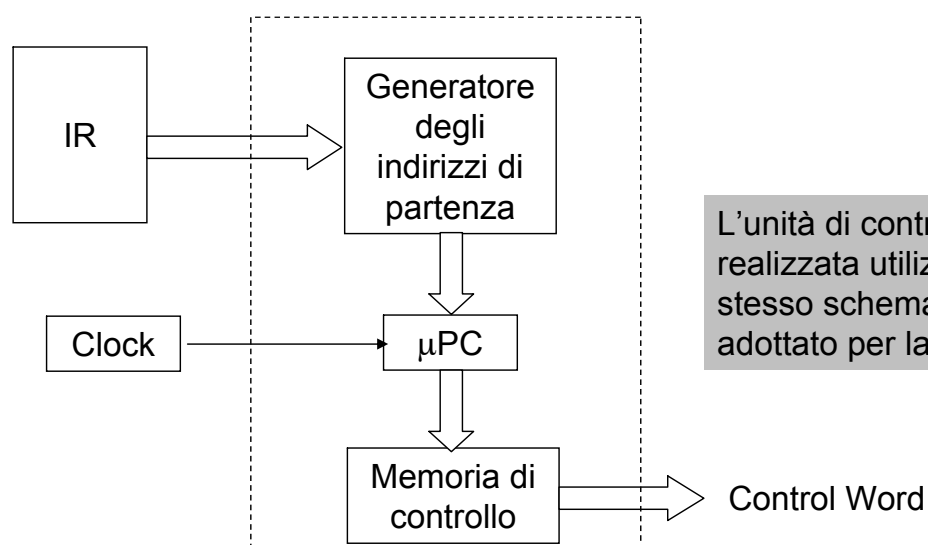
Università degli Studi
di Cassino

Organizzazione dell'unità di controllo (1/3)

Control Word ↔ **microistruzione**

Esecuzione di un'istruzione ↔ esecuzione di una sequenza di CW

→ **microprocedura**



L'unità di controllo viene realizzata utilizzando lo stesso schema di principio adottato per la CPU

F. Tortorella

Corso di Calcolatori Elettronici

Università degli Studi
di Cassino

Organizzazione dell'unità di controllo (2/3)

Problema: come gestire i salti condizionati ?

→ **microistruzioni di salto condizionato**

Nell'esecuzione di microprocedure è necessario eseguire un salto:

- quando si incontra una microistruzione End
- quando si avvia l'esecuzione di una nuova istruzione (dopo la fase di fetch)
- quando si incontra una microistruzione di salto condizionato

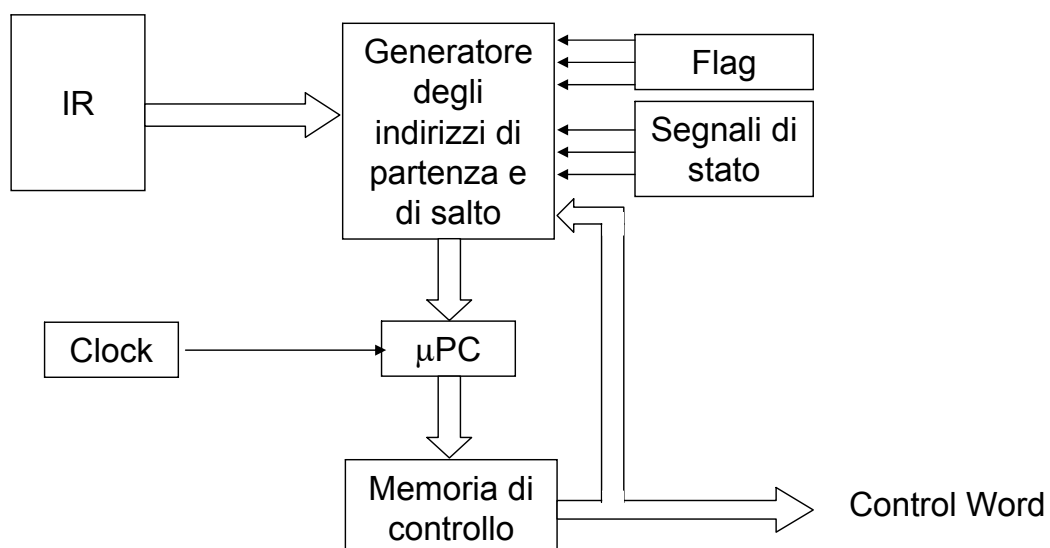
Realizzazione dei salti nella microprocedura

ampliamento dell'unità per generare gli indirizzi di partenza



generatore di indirizzi di partenza e di salto

Organizzazione dell'unità di controllo (3/3)



Formato delle microistruzioni (1/3)

Rappresentazione esplicita dei segnali di controllo nella CW

- microistruzioni molto lunghe (con pochi bit a 1)
- memoria di controllo molto ampia

E' possibile codificare i segnali di controllo con un numero minore di bit ?

Esaminiamo le combinazioni di segnali possibili

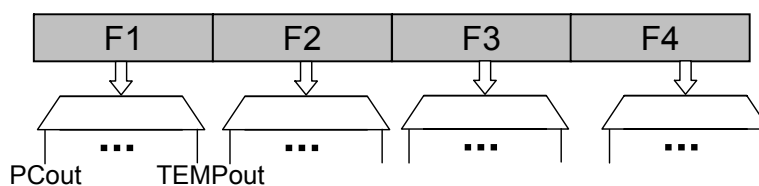
- gruppi di segnali non necessari contemporaneamente (PCout, Zout, R1out, ...)
- gruppi di segnali mutuamente esclusivi (ADD, SUB, ...)

Formato delle microistruzioni (2/3)

Microistruzioni organizzate in campi di bit

F1	F2	F3	F4	F5	...	F8	F9
0000: No Trasn	000: No Trasn	000: No Trasn	0000: ADD	00: No Trasn		0: None	0: Continue
0001: PC out	001: PC in	001: MAR in	0001: SUB	01: Read		1: WMFC	1: End
0010: MDR out	010: IR in	010: MDR in	0010: CMP	10: Write			
0011: Z out	011: Z in	011: TEMP in	...				
0100: R0 out	100: R0 in	100: Yin	1111: XOR				
...	...						
1010: TEMP out	111: R3 in						

- diminuisce la dimensione della control memory
- aumenta la complessità dell'hardware



→ ad ogni campo deve essere associato un decodificatore !

Formato delle microistruzioni (3/3)

Sulla base delle caratteristiche del formato delle microistruzioni, si distinguono:

Organizzazione orizzontale

formato lungo

capacità di esprimere un alto grado di parallelismo

codifica non spinta dei segnali di controllo

Organizzazione verticale

formato corto

basso grado di parallelismo

codifica spinta dei segnali di controllo