

L'Unità di Controllo

M. Sonza Reorda

Politecnico di Torino
Dipartimento di Automatica e Informatica

1

M. Sonza Reorda - a.a. 2001/02

Sommario

- **Introduzione**
- **Le unità di controllo**
- **Unità di controllo cablate**
- **Unità di controllo microprogrammate.**

2

M. Sonza Reorda - a.a. 2001/02

Introduzione

La CPU (*Central Processing Unit*) è l'unità che si occupa del caricamento e dell'esecuzione delle istruzioni.

Ciascuna delle istruzioni macchina eseguite dalla CPU è a sua volta composta da una serie di passi elementari, noti come *microistruzioni*.

3

M. Senza Reorda - a.a. 2001/02

Il processo di esecuzione delle istruzioni

Ciascuna istruzione può essere suddivisa in due fasi:

- caricamento dalla memoria del codice macchina dell'istruzione, il cui indirizzo è contenuto nel registro Program Counter ed aggiornamento del PC (fase di *fetch*)
- esecuzione delle azioni previste dall'istruzione (fase di *esecuzione*).

4

M. Senza Reorda - a.a. 2001/02

Operazioni elementari

Tutte le operazioni svolte dalla CPU possono essere ricondotte a 4 tipologie elementari:

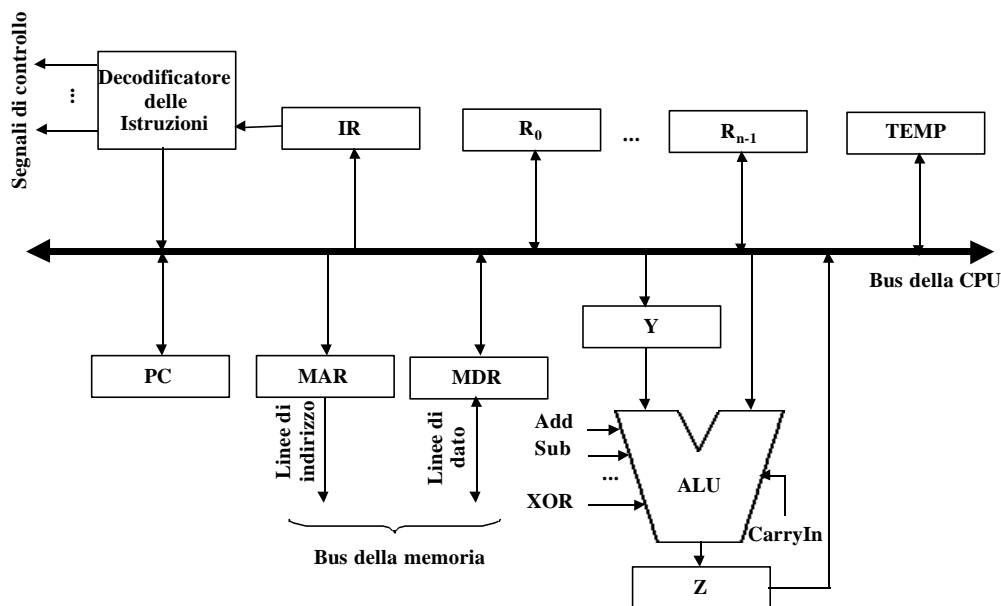
- prelievo di un dato o di una istruzione dalla memoria e caricamento in un registro
- scrittura in memoria di un dato contenuto in un registro
- trasferimento di un dato da un registro ad un altro
- esecuzione di un'operazione aritmetica o logica e memorizzazione del risultato in un registro.

Le 4 operazioni verranno descritte con riferimento al seguente modello della CPU, semplificato rispetto a quello reale.

5

M. Sonza Reorda - a.a. 2001/02

Architettura della CPU



6

M. Sonza Reorda - a.a. 2001/02

Trasferimenti tra registri

Si può supporre che ogni registro R_i connesso al bus possieda un'interfaccia controllata da due segnali:

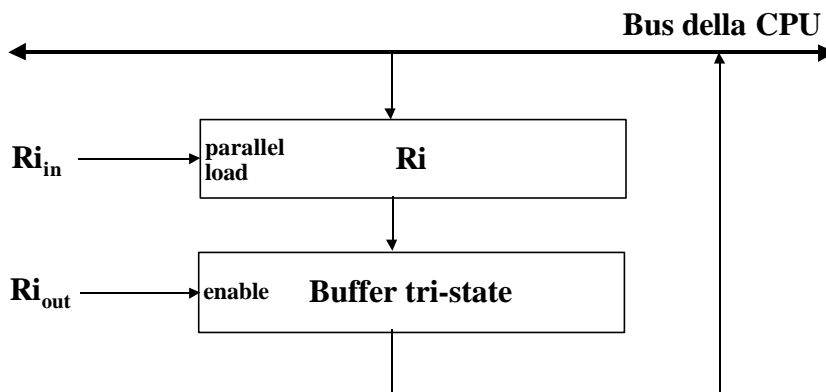
- $R_{i_{in}}$, che fa sì che il registro carichi il valore presente sul bus
- $R_{i_{out}}$, che fa sì che il contenuto del registro venga trasferito sul bus; quando il segnale non è attivo, il registro forza sul bus il valore Z.

Per il corretto funzionamento del bus è essenziale che ad ogni istante uno ed uno soltanto dei segnali $R_{i_{out}}$ sia attivo.

7

M. Sonza Reorda - a.a. 2001/02

Realizzazione dell'interfaccia



8

M. Sonza Reorda - a.a. 2001/02

Trasferimenti tra registri

Avvengono agendo opportunamente sui segnali di interfaccia dei registri connessi al bus interno della CPU.

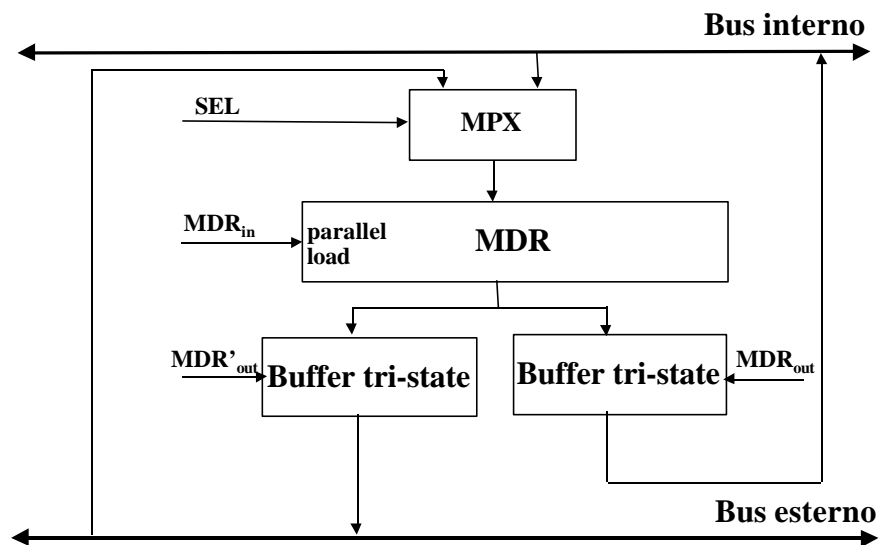
Ad esempio, per trasferire il contenuto del registro R1 in R4 si deve:

- attivare $R1_{out}$: in tal modo il valore di R1 viene posto sul bus
- attivare $R4_{in}$: in tal modo R4 memorizza il valore presente sul bus.

9

M. Sonza Reorda - a.a. 2001/02

Connessione di MDR



10

M. Sonza Reorda - a.a. 2001/02

Prelievo di una parola dalla memoria

- MAR \rightarrow R1
- Attivazione dei segnali di lettura
- Aspetta sino al segnale MFC
- MDR \rightarrow Bus esterno
- R2 \rightarrow MDR

Quando la memoria, dopo aver percepito i segnali di lettura, ha posto sul bus di dato il dato richiesto, attiva il segnale MFC (*Memory Function Completed*).

Tale meccanismo presuppone un bus di tipo asincrono.

11

M. Sonza Reorda - a.a. 2001/02

Memorizzazione di una parola in memoria

- MAR \rightarrow R1
- MDR \rightarrow R2, attivazione dei segnali di scrittura
- Aspetta sino al segnale MFC.

12

M. Sonza Reorda - a.a. 2001/02

Esecuzione di un'operazione logica o aritmetica

Per eseguire la somma tra i registri R1 ed R2 e mettere il risultato in R3 si devono attivare i seguenti segnali:

- $R1_{out}$, Y_{in}
- $R2_{out}$, Add, Z_{in}
- Z_{out} , $R3_{in}$.

13

M. Senza Reorda - a.a. 2001/02

Temporizzazioni

Ciascun segnale deve rimanere attivo per il tempo necessario a completare l'operazione da esso pilotata, tenuto conto dei ritardi dei vari componenti attivati.

14

M. Senza Reorda - a.a. 2001/02

Esempio

Si consideri l'istruzione

Add (R3), R1

che somma ad R1 il contenuto della cella di memoria il cui indirizzo è contenuto in R3, e mette il risultato in R1.

I segnali che devono essere attivati sono:

- PC_{out} , MAR_{in} , Read, Clear Y, Set Carry In to ALU, Add, Z_{in}
- Z_{out} , PC_{in} , aspetta MFC
- MDR_{out} , IR_{in}
- $R3_{out}$, MAR_{in} , Read
- $R1_{out}$, Y_{in} , aspetta MFC
- MDR_{out} , Add, Z_{in}
- Z_{out} , $R1_{in}$, End

15

M. Sonza Reorda - a.a. 2001/02

Istruzioni di salto

I segnali che devono essere attivati per eseguire un'istruzione di salto incondizionato sono:

- PC_{out} , MAR_{in} , Read, Clear Y, Set Carry In to ALU, Add, Z_{in}
- Z_{out} , PC_{in} , aspetta MFC
- MDR_{out} , IR_{in}
- PC_{out} , Y_{in}
- (Campo offset di IR)_{out}, Add, Z_{in}
- Z_{out} , PC_{in} , End.

16

M. Sonza Reorda - a.a. 2001/02

Salto condizionato

I segnali che devono essere attivati per eseguire un'istruzione di salto condizionato sono:

- PC_{out} , MAR_{in} , Read, Clear Y, Set Carry In to ALU, Add, Z_{in}
- Z_{out} , PC_{in} , aspetta MFC
- MDR_{out} , IR_{in}
- PC_{out} , Y_{in} ; se $N=0$ allora End
- (Campo offset di IR) $_{out}$, Add, Z_{in}
- Z_{out} , PC_{in} , End.

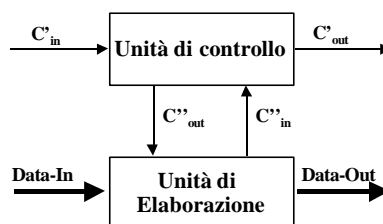
17

M. Sonza Reorda - a.a. 2001/02

Le Unità di Controllo

I processori sono strutturati in 2 parti:

- l'*unità di elaborazione* (o *data-path*): contiene i registri, le unità aritmetico-logiche, le memorie, etc.
- l'*unità di controllo* (UC): genera i segnali di controllo per l'unità di elaborazione, sulla base dei segnali provenienti dalla stessa unità e dall'esterno.



18

M. Sonza Reorda - a.a. 2001/02

Funzionamento dell'Unità di Controllo

L'Unità di Controllo è una macchina a stati che riceve segnali dall'esterno (C'_{in}) e dall'Unità di Elaborazione (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 l'Unità di Elaborazione (C''_{out}).

La complessità dell'Unità di Controllo è quindi proporzionale al prodotto tra il numero di segnali di controllo (dall'esterno e dall'Unità di Elaborazione) ed il numero di variabili di stato.

19

M. Sonza Reorda - a.a. 2001/02

Progetto dell'Unità di Controllo

Si parte generalmente da una descrizione del suo funzionamento, basata ad esempio su un diagramma a stati.

Si esegue poi la trasformazione in hardware.

Sono state definite 2 strategie:

- Unità di Controllo *cablate* (o *hardwired*): la UC viene considerata come una normale macchina a stati a cui applicare i metodi tradizionali di progetto, oppure metodi *ad hoc*
- Unità di Controllo *microprogrammate*: 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.

20

M. Sonza Reorda - a.a. 2001/02

Obiettivi

- **Minimizzare la quantità di hardware**
- **Massimizzare la velocità di esecuzione**
- **Ridurre il tempo di progetto (*flessibilità*).**

21

M. Senza Reorda - a.a. 2001/02

Unità di Controllo Cablate

Approcci:

- **progetto di una FSM**
- **architettura tradizionale**
- **elementi di ritardo.**

22

M. Senza Reorda - a.a. 2001/02

Progetto di una FSM

Lo stato dell'Unità di Controllo è caratterizzato dai valori presenti negli elementi di memoria in corrispondenza del colpo di clock.

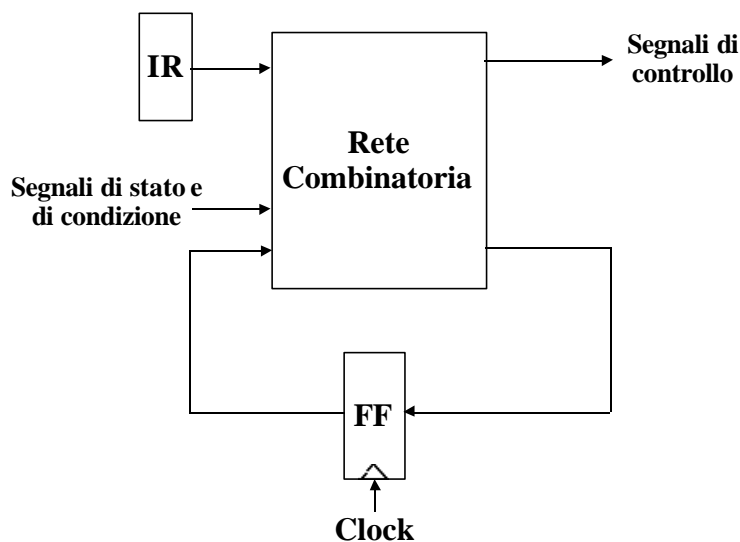
Per ogni stato, in corrispondenza di ciascuna possibile combinazione di ingressi (provenienti dall'esterno e dall'Unità di Elaborazione), si definiscono le combinazioni di uscita (segnali di controllo verso l'Unità di Elaborazione) e lo stato futuro.

Si costruisce così una tabella degli stati, che viene poi trattata con i normali metodi di sintesi, e da essa si genera l'hardware corrispondente.

23

M. Sonza Reorda - a.a. 2001/02

Architettura



24

M. Sonza Reorda - a.a. 2001/02

Temporizzazione

Si assume che esista un clock in grado di scandire la tempistica di generazione dei segnali di controllo verso l'Unità di Elaborazione.

25

M. Senza Reorda - a.a. 2001/02

Svantaggi

- Le dimensioni della tabella possono renderla intrattabile
- Alcune informazioni legate al funzionamento dell'Unità di Controllo (ad esempio la presenza di cicli) vengono perse
- Una volta che la tabella è stata trasformata nell'hardware corrispondente, questo non ha una struttura regolare, e la sua modifica è in genere estremamente complessa.

26

M. Senza Reorda - a.a. 2001/02

Passi di progetto

Il progetto avviene nei seguenti passi:

- stesura del progetto con un Hardware Description Language
- stesura di un Flow-Chart
- identificazione dei segnali di controllo (individuazione delle operazioni contemporanee)
- identificazione degli stati e loro etichettamento
- costruzione della tabella degli stati
- individuazione e fusione degli stati compatibili
- costruzione della tabella delle transizioni
- sintesi rete combinatoria.

27

M. Sonza Reorda - a.a. 2001/02

Sintesi Rete Combinatoria

Può venire effettuata a mano o in modo automatico.

Il risultato può essere

- logica sparsa (*glue logic*)
- una PLA
- una ROM.

28

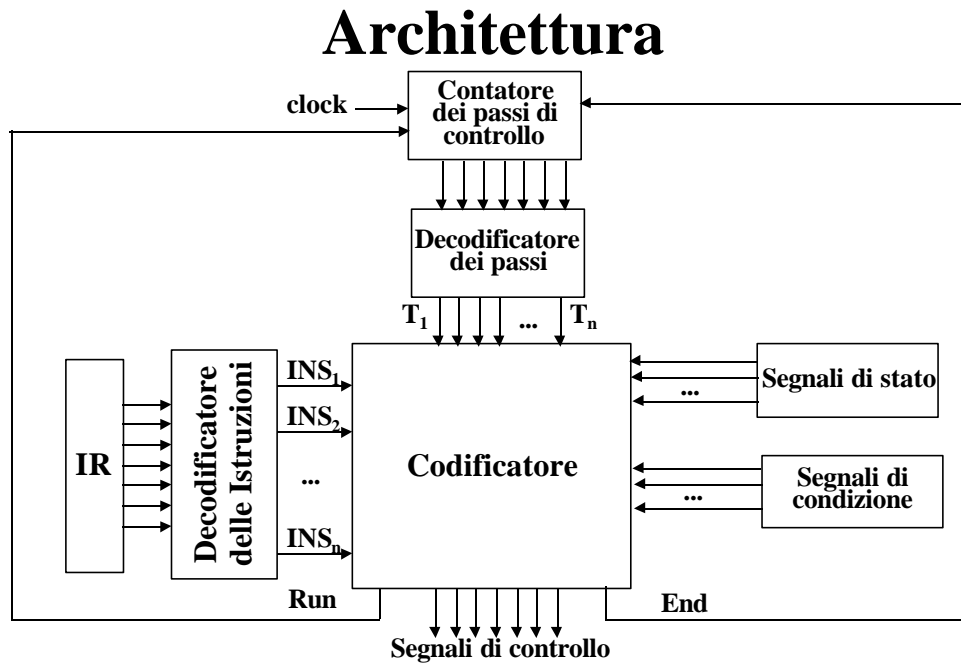
M. Sonza Reorda - a.a. 2001/02

Architettura tradizionale

Anziché ricondurre il progetto dell'Unità di Controllo a quello di una generica FSM, si definisce in questo caso uno schema specifico per l'Unità di Controllo di un processore.

29

M. Sonza Reorda - a.a. 2001/02



30

M. Sonza Reorda - a.a. 2001/02

Codificatore

Il codificatore genera i segnali di controllo nei modi e tempi richiesti.

Ad esempio, con riferimento all'architettura del processore presentato all'inizio, il segnale Z_{in} deve essere generato:

- nel primo ciclo, per ogni istruzione
- nel quinto ciclo, per l'istruzione BR
- nel sesto ciclo, per le istruzioni ADD.

La funzione implementata dal codificatore per quanto riguarda tale segnale sarà quindi

$$Z_{in} = T_1 + T_6 \cdot ADD + T_5 \cdot BR + \dots$$

Elementi di Ritardo

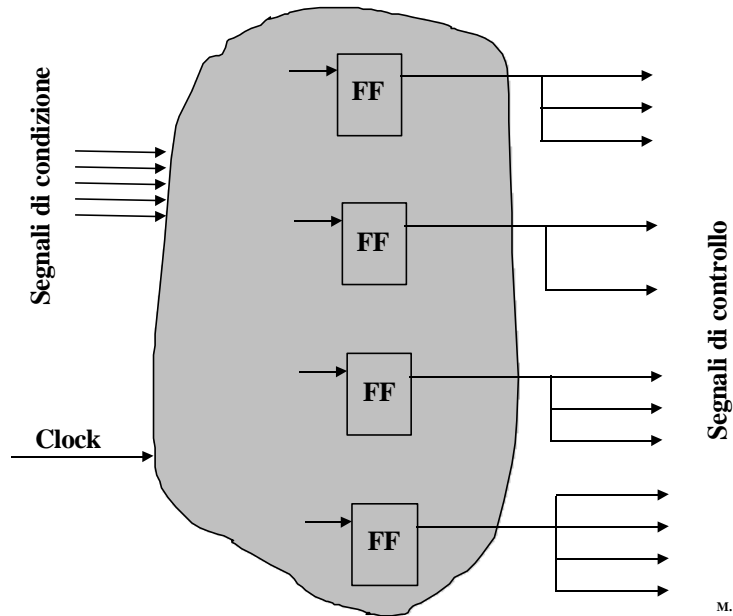
Questa tecnica si basa sull'uso di un numero di Flip Flop pari al numero di stati della UC: ogni Flip Flop contiene il valore 1 solo quando la UC si trova nello stato corrispondente.

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 c_i che devono essere attivati in un certo stato.

È possibile passare in maniera automatica dal flow-chart della UC al corrispondente hardware.

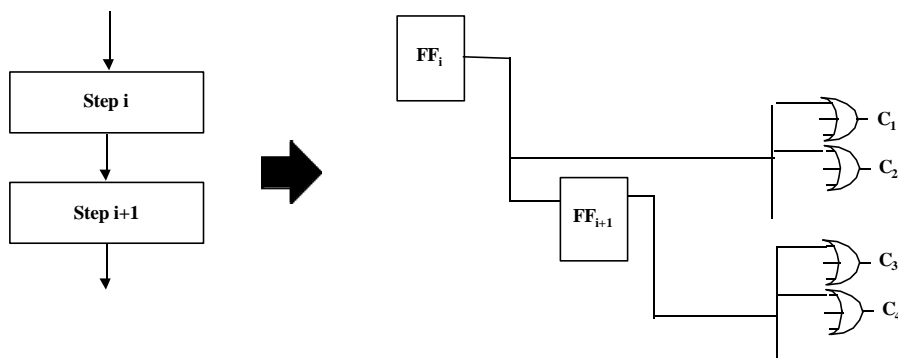
Architettura dell'UC



33

M. Sonza Reorda - a.a. 2001/02

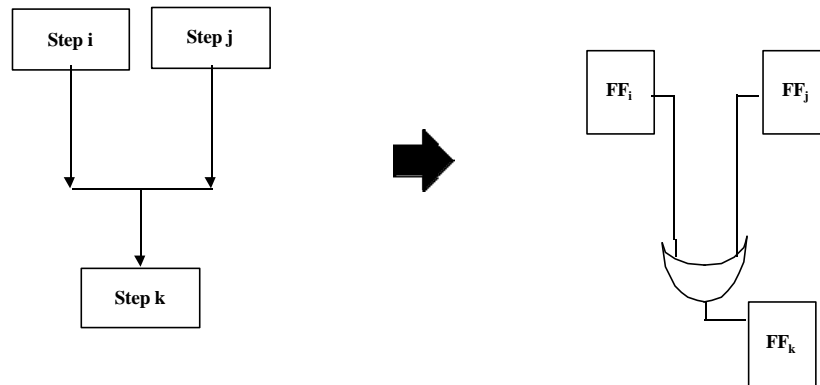
Regole di Trasformazione (I)



34

M. Sonza Reorda - a.a. 2001/02

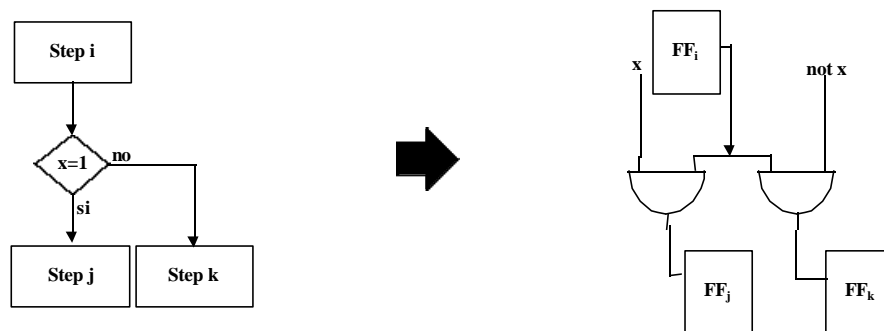
Regole di Trasformazione (II)



35

M. Sonza Reorda - a.a. 2001/02

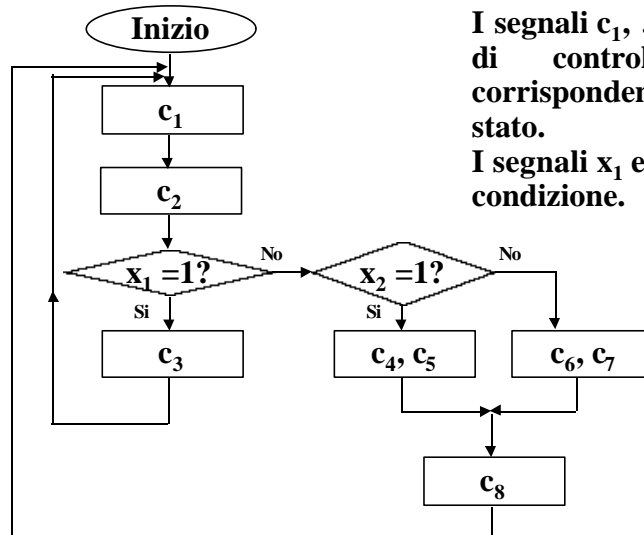
Regole di Trasformazione (III)



36

M. Sonza Reorda - a.a. 2001/02

Esempio



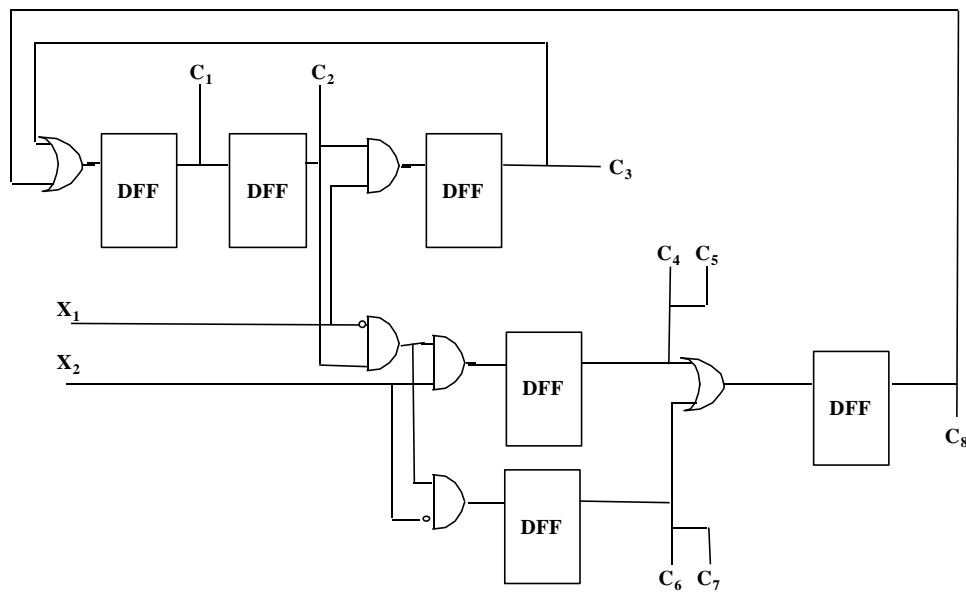
I segnali c_1, \dots, c_8 sono i segnali di controllo attivati in corrispondenza di ciascuno stato.

I segnali x_1 ed x_2 sono segnali di condizione.

37

M. Sonza Reorda - a.a. 2001/02

Implementazione

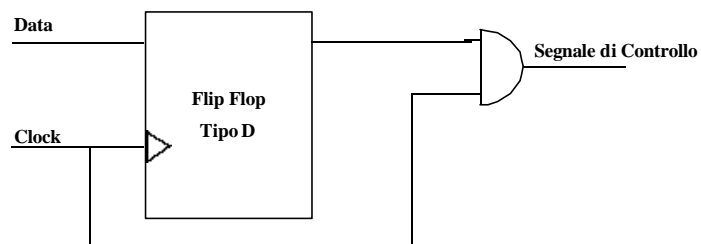


38

M. Sonza Reorda - a.a. 2001/02

Elemento di Ritardo

Nel caso di segnali di controllo impulsivi, lo schema seguente garantisce che la durata dei segnali di controllo sia pari a quella di un impulso di clock.



39

M. Sonza Reorda - a.a. 2001/02

Problemi

- Ci vogliono n elementi di ritardo (contro $\log n$ richiesti progettando una FSM)
- È difficile propagare il clock a tutti gli elementi di ritardo.

40

M. Sonza Reorda - a.a. 2001/02