

## Reti sequenziali

- Rappresentano l'implementazione di funzioni con stato, ossia di **automi a stati finiti**
- Strumenti di espressione (specifica) della **parte controllo** e della **parte operativa** di unità di elaborazione

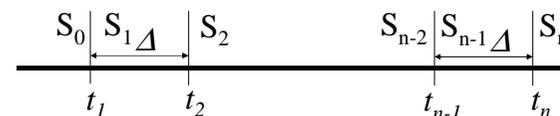
Struttura di un elaboratore

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 30

## Reti sequenziali (segue)

- In un modello **ideale** di rete sequenziale di tipo **sincrono**, le variazioni di stato ( $S_i \rightarrow S_{i+1}$ ) avvengono in corrispondenza degli istanti di una sequenza temporale **discreta** ( $t_1, t_2, \dots, t_n, \dots$ ) di intervallo **costante** ( $\Delta = t_n - t_{n-1}$ )



Struttura di un elaboratore

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 32

## Reti sequenziali (segue)

- Un automa a stati finiti è una macchina caratterizzata da:
  - ✓ **n** variabili logiche di **ingresso** ( $2^n$  stati di ingresso  $X_i$ )
  - ✓ **m** variabili logiche di **uscita** ( $2^m$  stati di uscita  $Z_i$ )
  - ✓ **k** variabili logiche di **stato interno** ( $2^k$  stati interni  $S_i$ )
  - ✓ una **funzione  $\sigma$  di transizione dello stato interno**
    - $\sigma : X \times S \rightarrow S$
  - ✓ una **funzione  $\omega$  delle uscite**
    - $\omega : X \times S \rightarrow Z$

Struttura di un elaboratore

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 31

## Reti sequenziali (segue)

- 2 modelli matematici di automa
  - ✓ **Modello di Mealy**
    - $S(t_{i+1}) = \sigma(X(t_i), S(t_i))$
    - $Z(t_{i+1}) = \omega(X(t_i), S(t_i))$
  - ✓ Sia lo stato interno **successivo**  $S(t_{i+1})$  che lo stato di uscita successivo  $Z(t_{i+1})$  dipendono tanto dallo stato di ingresso **presente**  $X(t_i)$  che dallo stato interno presente  $S(t_i)$

Struttura di un elaboratore

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 33

## Reti sequenziali (segue)

### ✓Modello di Moore

- $S(t_{i+1}) = \sigma(X(t_i), S(t_i))$

- $Z(t_{i+1}) = \omega(S(t_i))$

✓La sequenza di uscita è **ritardata** di un intervallo  $\Delta$  rispetto a quella di un modello di Mealy:

- $Z(t_{i+1}) = \omega(S(t_i)) = \omega(\sigma(X(t_{i-1}), S(t_{i-1}))) = \omega(X(t_{i-1}), S(t_{i-1}))$

Struttura di un elaboratore

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 34

## Reti sequenziali (segue)

### • Esempio

- $y = S(t_i) ; Y = S(t_{i+1}) ; x = X(t_i) ; z = Z(t_{i+1})$

		$\omega$	$\sigma$
$y$	$x$	$z$	$Y$
0	0	0	1
1	0	1	1
0	1	0	0
1	1	1	0

Struttura di un elaboratore

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 36

## Reti sequenziali (segue)

- La realizzazione delle funzioni  $\sigma$  ed  $\omega$  mediante reti combinatorie dà luogo alla **parte combinatoria** delle reti sequenziali

- Il modo classico di rappresentare (specificare) è mediante **tabelle di verità**

Struttura di un elaboratore

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 35

## Reti sequenziali (segue)

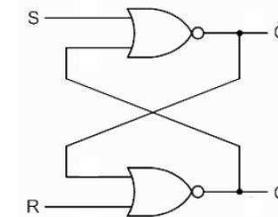
### • Esempio: Latch SR

– permette di memorizzare 1 bit

Ingressi: S e R

Stato:  $\bar{Q}$  e Q

Uscite:  $\bar{Q}$  e Q



$$S(t_{i+1}) = \sigma(X(t_i), S(t_i)) = \sigma(S_t, R_t, \bar{Q}_t, Q_t) \\ = \{ \bar{Q}_{t+1} = \text{nor}(S_t, Q_t), Q_{t+1} = \text{nor}(Q_t, R_t) \}$$

$$Z(t_{i+1}) = \omega(X(t_i), S(t_i)) = \omega(S_t, R_t, \bar{Q}_t, Q_t) = \sigma(S_t, R_t, \bar{Q}_t, Q_t)$$

Struttura di un elaboratore

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 37

### Reti sequenziali (segue)

input $t$		stato $t$		stato $t+1$	
S	R	Q	$\bar{Q}$	Q	$\bar{Q}$
0	0	0	0	1	1
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	0	0
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

Esistono 3 “stati stabili” per alcune configurazioni in ingresso:

- 00 01 → 01 (stato stabile “1”)
- 00 10 → 10 (stato stabile “0”)
- 01 10 → 10
- 10 01 → 01
- 11 00 → 00 (stato stabile da evitare)

A noi interessa interpretare  $\bar{Q}$  come il complemento di Q, cioè  $Q = \text{not}(\bar{Q})$   
 Quindi gli stati 00 e 11 li consideriamo non coerenti (output non coerente)

Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 38

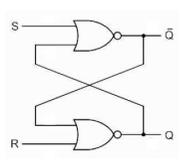
### Reti sequenziali (segue)

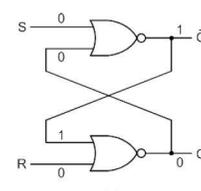
- Nel modello **ideale** le funzioni  $\sigma$  ed  $\omega$  hanno un tempo di stabilizzazione **nullo**
- Nel modello **reale** occorre un ritardo **non nullo** per la stabilizzazione delle uscite, a partire da quando gli ingressi sono stabili
- Gli ingressi di tipo  $\{y\} = S(t_i)$  variano, in principio, in tempi **diversi** rispetto agli ingressi di tipo  $\{x\} = X(t_i)$

Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 40

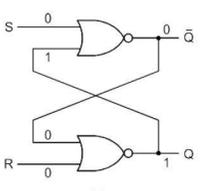
### Reti sequenziali (segue)

- Esempio: Latch SR**
  - permette di memorizzare 1 bit
  - ha 2 stati stabili (bit a 0, e bit a 1)





(a)  
Stato stabile 0



(b)  
Stato stabile 1

S (setting) imposta lo stato 1  
 R (reset) imposta lo stato 0  
 Q è l'output  
 $\bar{Q}$  è il complemento di Q

Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 39

### Reti sequenziali (segue)

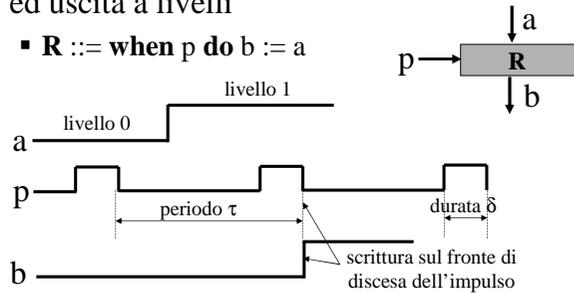
- Occorre dunque ripristinare l'ipotesi che le le variabili di ingresso alla rete varino tutte simultaneamente, ad intervalli temporali discreti  $\Delta$

Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 41

### Reti sequenziali (segue)

- Rete sequenziale impulsata **R** con ingresso ed uscita a livelli

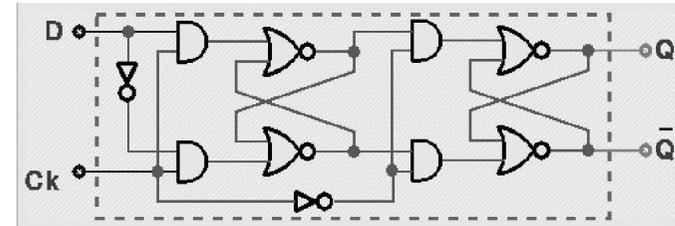
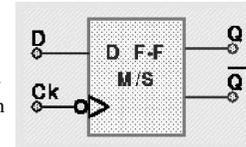
▪  $R ::= \text{when } p \text{ do } b := a$



### Reti sequenziali (segue)

- Master/Slave Edge-triggered:

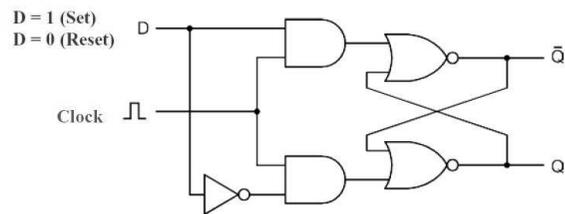
– la variazione dell'output si ha quando cambia la variabile di controllo (edge-triggered) e non direttamente dal valore (level-triggered)



### Reti sequenziali (segue)

- Latch D sincronizzato:

- evita che S e R siano uguali
- sincronizzato



### Reti sequenziali (segue)

- Importanza della sincronia: problema delle alee

- in un circuito non sincronizzato (asincrono) si possono creare problemi di stabilizzazione del segnale, o meglio fenomeni transitori, detti alee di commutazione, che possono determinare un comportamento non atteso della rete sequenziale
- in particolare, una alea di commutazione è determinata dal ritardo di propagazione della variazione dei segnali all'interno del circuito

### Reti sequenziali (segue)

- Esempio di alea di commutazione

Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 46

### Reti sequenziali (segue)

- Esempio di alea di commutazione  
 $t = 2$

Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 48

### Reti sequenziali (segue)

- Esempio di alea di commutazione  
 $t = 1$

Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 47

### Reti sequenziali (segue)

- Esempio di alea di commutazione  
 $t = 3$

Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 49

### Reti sequenziali (segue)

- Esempio di alea di commutazione

$t = 4$

Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 50

### Reti sequenziali (segue)

- Esempio di alea di commutazione

$t = 6$

Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 52

### Reti sequenziali (segue)

- Esempio di alea di commutazione

$t = 5$

Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 51

### Reti sequenziali (segue)

- Esempio di alea di commutazione

$t = 7$

Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 53

### Reti sequenziali (segue)

- Esempio di alea di commutazione  
 $t = 8$

Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 54

### Reti sequenziali (segue)

- Registro di tipo D (a 4 bit):
  - è il circuito sincrono piu' semplice che realizza un registro
  - memorizzazione (store): dati presentati in ingresso e clock da 0 a 1 (uscita riproduce ingresso)
  - mantenimento (hold): clock da 1 a 0 (poi costante); l'uscita rimane invariata indipendentemente dal valore degli ingressi

Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 56

### Reti sequenziali (segue)

- Rete sequenziale sincrona 'Level input Level output Clocked' (LLC), modello di Mealy
  - {x} e {y} sono a livelli; p (clock) è impulsivo

Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 55

### Reti sequenziali (segue)

- Registro a scorrimento:
  - C=0  
scorrimento verso l'alto dei bit, e valore "0" inserito sul bit meno significativo
  - C=1  
store degli ingressi
  - permette la conversione "parallelo/seriale": si memorizzano i dati in parallelo e si "trasmettono" uno alla volta su  $Q_3$

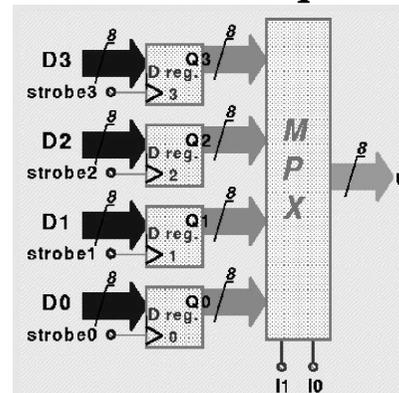
Struttura di un elaboratore      Architettura degli Elaboratori - 1 - A. Sperduti      Pagina 57

## Reti sequenziali (segue)

- Memorie (RAM):
  - insieme di celle identificate da un numero (**indirizzo**)
  - ogni cella è in grado di codificare n bit (es. n=8 [byte], n=16, ...)
  - ogni cella deve preservare nel tempo il suo contenuto (**memoria**) a meno di operazioni di scrittura
  - una cella deve poter essere letta fornendo l'indirizzo della stessa (**lettura**)
  - la lettura di una cella non deve cancellare l'informazione letta
  - una cella deve poter essere scritta fornendo sia l'indirizzo della stessa che l'informazione da memorizzare (compatibile con il valore di n) (**scrittura**)

Struttura di un elaboratore Architettura degli Elaboratori - 1 - A. Sperduti Pagina 58

## Reti sequenziali (segue)



- Lettura
  - 4 celle di 8 bit realizzate tramite registri sincroni di tipo D
  - 2 bit di indirizzo ( $I_0, I_1$ )
  - strobe è il segnale di clock
  - MPX è un multiplexer

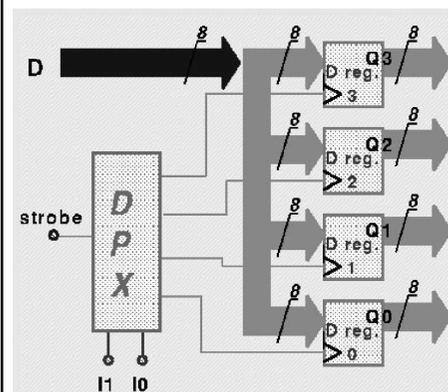
Struttura di un elaboratore Architettura degli Elaboratori - 1 - A. Sperduti Pagina 60

## Reti sequenziali (segue)

- RAM statiche:
  - implementate tramite registri di tipo D
  - Problema: realizzare un meccanismo per consentire l'accesso in scrittura e lettura di una singola cella di memoria alla volta (specificata dall'indirizzo in ingresso alla memoria)
  - gli indirizzi della memoria saranno specificati come numeri interi progressivi rappresentati in base 2 (rappresentazione binaria; es.  $ind=0$  a  $7$ , richiede 3 bit:  $ind=5$  è rappresentato da  $101$ ,  $ind=6$  è rappresentato da  $110$ , ...)
  - vediamo come realizzare separatamente i meccanismi per implementare la **lettura** e quelli per implementare la **scrittura**

Struttura di un elaboratore Architettura degli Elaboratori - 1 - A. Sperduti Pagina 59

## Reti sequenziali (segue)

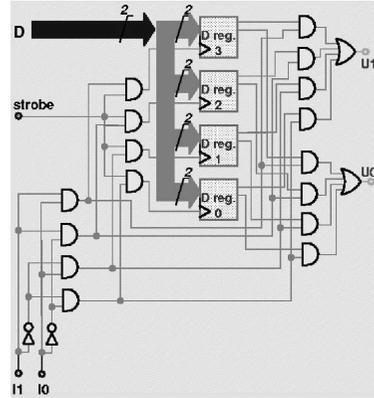


- Scrittura
  - 4 celle di 8 bit realizzate tramite registri sincroni di tipo D
  - D è il dato da memorizzare all'indirizzo specificato da  $I_1 I_0$
  - strobe è il segnale di clock
  - DPX è un demultiplexer

Struttura di un elaboratore Architettura degli Elaboratori - 1 - A. Sperduti Pagina 61

## Reti sequenziali (segue)

- RAM ottimizzata
  - combina il multiplexer con il demultiplexer
  - 4 celle di 2 bit
  - la lettura avviene sempre
  - la scrittura avviene come in precedenza

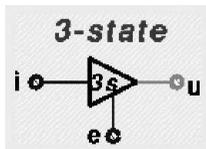


Struttura di un elaboratore

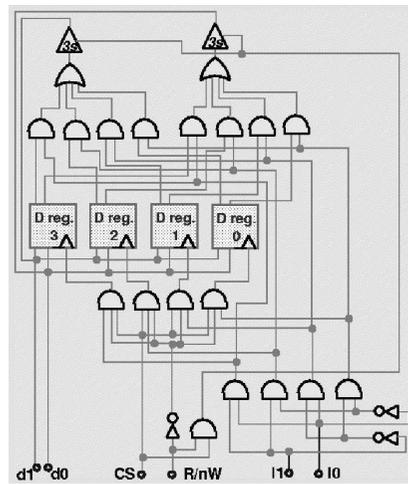
Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 62

- RAM modulari
  - usano dispositivo a 3 stati



i	e	u
0	0	non connesso
0	1	0
1	0	non connesso
1	1	1



Struttura di un elaboratore

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 63