Livello Elettrico

Il sistema viene modellato come interconnessione di componenti quali resistori, condensatori, induttori, etc.

Il calcolo dei valori di corrente e tensione nei diversi punti comporta la risoluzione di un sistema di equazioni differenziali.

Sul mercato esistono vari pacchetti SW (ad es. *SPICE*) in grado di simulare circuiti di dimensioni comunque ridotte.

Sta al progettista trovare un soddisfacente compromesso tra precisione e tempo di calcolo.

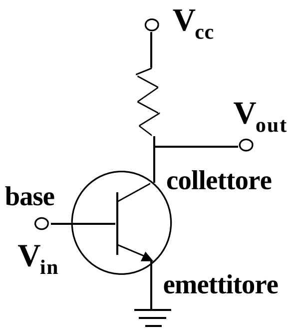
1

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

Livello Transistor

Viene utilizzato solo per alcune parti critiche dal punto di vista elettrico o delle prestazioni.

Il circuito viene visto come una interconnessione di *transistor*, considerati come interruttori pilotati dal segnale di base:



Se $V_{in}<V_{soglia}$:

$V_{out}=V_{cc}$

Se $V_{in}>V_{soglia}$:

$V_{out}=V_{ground}$

La velocità di commutazione di un transistor è dell'ordine dei nanosecondi.

2

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

Livello Porte Logiche

A questo livello vengono progettati circuiti di tipo combinatorio o sequenziale; i componenti sono le porte logiche (*gate*).

Per questa ragione è anche noto come *livello logico*.

Le porte logiche sono elementi operanti su variabili binarie che possono assumere i due valori 0 e 1.

Le informazioni trattate a questo livello sono *segnali* binari; ogni linea del circuito può cioè assumere 2 soli valori di tensione, corrispondenti ai 2 valori logici 0 e 1.

Circuiti Combinatori

Implementano funzioni combinatorie.

Una *funzione combinatoria* è una trasformazione

z: B^n ® B,

dove B={0,1}.

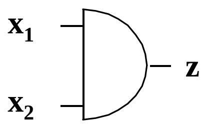
Le funzioni combinatorie non coinvolgono il tempo (a differenza di quelle sequenziali) e possono essere descritte (se il numero di variabili è piccolo) attraverso *tavole di verità*.

Porte Logiche (I)

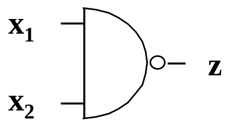
Compongono i circuiti combinatori.

L’insieme delle porte logiche utilizzate è formato dai seguenti componenti

AND



NAND



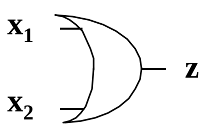
$$z = x_1 x_2 = x_1 \wedge x_2$$

$$z = \overline{x_1 x_2} = \overline{x_1 \wedge x_2}$$

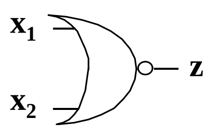
5

Porte Logiche (II)

OR



NOR



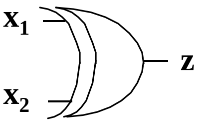
$$z = x_1 + x_2 = x_1 \vee x_2$$

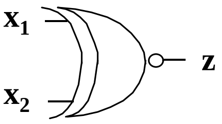
$$z = \overline{x_1 + x_2} = \overline{x_1 \vee x_2}$$

6

Porte Logiche (III)

EXOR **EXNOR**





$z = x_1 \oplus x_2$

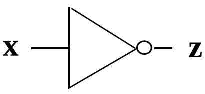
$z = \overline{x_1 \oplus x_2}$

7

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

Porte Logiche (IV)

NOT

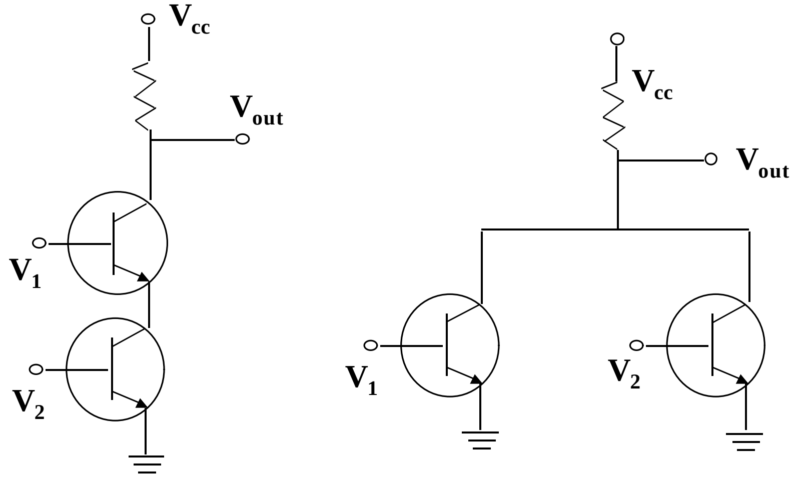


$z = \overline{x}$

8

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

Implementazione delle Porte Logiche tramite Transistor



9 **Porta NAND** **Porta NOR**

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

Insieme di Porte Completo

Un insieme di tipi di porte logiche si dice *completo* se utilizzando i soli tipi di porte in esso contenuti si può realizzare qualsivoglia funzione combinatoria.

Sono insiemi completi:

- **{NAND}**
- **{NOR}**
- **{AND, NOT}**
- **{OR, NOT}**
- **{AND, OR, NOT}**

Nella pratica sono frequenti i circuiti realizzati con sole porte NAND o NOR.

10

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

Circuiti Combinatori *Ben Formati*

Si dicono *Circuiti Combinatori Ben Formati* (ccbf) i circuiti combinatori che soddisfano le seguenti regole:

- **una singola linea o una singola porta è un ccbf**
- **la giustapposizione di 2 ccbf è un ccbf**
- **se C_1 e C_2 sono due ccbf, il circuito ottenuto connettendo un insieme di linee di uscita di C_1 ad un insieme di linee di ingresso di C_2 è un ccbf**
- **se x_i ed x_j sono due linee di ingresso ad un ccbf, il circuito ottenuto connettendo insieme x_i ed x_j è un ccbf.**

I circuiti che soddisfano tali condizioni sono *aciclici*.

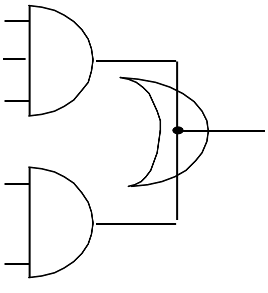
11

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

Wired-Logic

Alcune tecnologie per la produzione di IC permettono la connessione diretta delle linee di uscita di 2 o più porte logiche (violando le regole sui ccbf), in quanto si garantisce che il valore assunto dalla nuova linea è definito.

Tali logiche sono di tipo *wired-or* o *wired-and*.



12

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

Ritardi

L'uscita di una porta logica assume il valore corretto solo dopo che è trascorso un certo tempo *k* dalla stabilizzazione dei segnali di ingresso.

Conoscendo i ritardi associati alle porte logiche componenti è possibile calcolare il tempo di risposta di un circuito combinatorio.

Assumendo che i ritardi delle porte siano uguali, tale tempo è proporzionale alla *profondità* del circuito, ossia al massimo numero di porte che si incontrano lungo un qualsiasi cammino da un ingresso ad una uscita.

Cammino Critico

Il cammino lungo il quale il ritardo con il quale una variazione in ingresso si propaga sulle uscite è massimo si definisce *cammino critico* (*critical path*).

Ridurre la lunghezza del cammino critico permette quindi di migliorare le prestazioni (in termini di velocità) dei circuiti.

Livelli

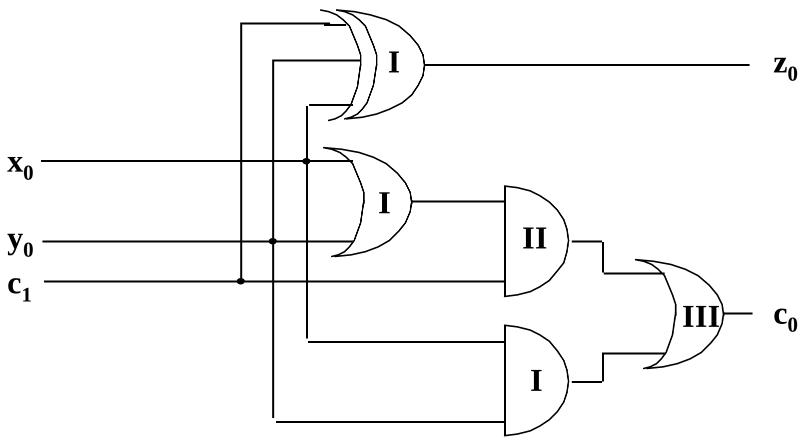
Ad ogni porta logica in un circuito può essere assegnato un *livello* nella seguente maniera:

- il livello delle porte che hanno come ingressi solo linee di ingresso è 1
- il livello delle altre porte è pari al livello della porta di ingresso avente il livello massimo, più 1.

La *profondità* di un circuito è pari al livello della porta di livello massimo. Questa alimenterà sicuramente una linea di uscita.

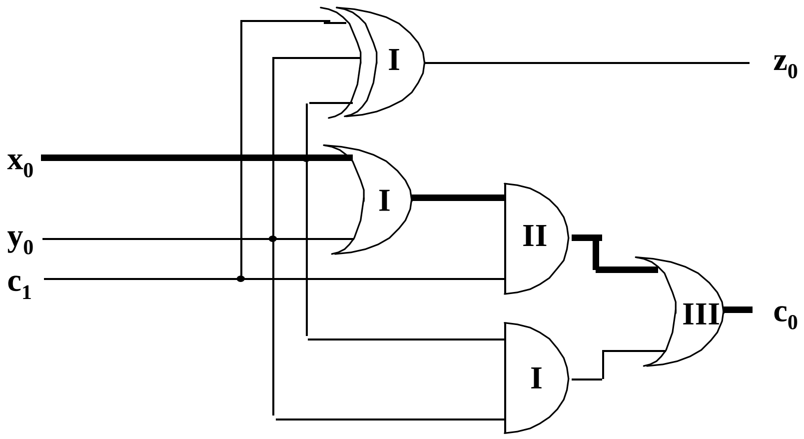
15

Livelli: esempio



16

Cammino critico



Algebra Booleana

È definita sui seguenti elementi ed operatori:

- ***elementi*: 0, 1**
- ***operatori*: AND (·), OR (+), NOT**
- ***assiomi*: (Postulati di Huntington)** **K={0,1}**
  - **chiusura:** $a \in K, b \in K \Rightarrow ab \in K, a + b \in K$
  - **identità: a+0=a, a · 1=a**
  - **commutatività: a+b=b+a, a · b=b · a**
  - **distributività: a(b+c)=a · b+a · c, a+(b · c)=(a+b) ·(a+c)**
  - **inverso:** $a\overline{a} = 0, a + \overline{a} = 1$

Leggi dell'Algebra Booleana

- ***associatività*:**
  a+(b+c)=(a+b)+c
  a(bc)=(ab)c
- ***idempotenza*:**
  a+a=a
  aa=a
- ***De Morgan*:** $\overline{a+b} = \bar{a}\bar{b}$

  $\overline{ab} = \bar{a} + \bar{b}$
- ***involuzione*:**

  $\bar{\bar{a}} = a$

19

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

Progetto di Circuiti Combinatori

Consiste nell'individuare quel circuito combinatorio che:

- **realizza un dato insieme di funzioni booleane date**
- **soddisfa determinati vincoli quali:**
  - **la profondità del circuito è minore di un certo valore**
  - **il *fan-in* (numero di porte di ingresso) massimo è minore di un certo valore**
  - **il *fan-out* (numero di porte alimentate) massimo è minore di un certo valore.**

20

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

Circuiti a 2 livelli

Una soluzione comune al problema del progetto dei circuiti combinatori è rappresentata dai circuiti a 2 livelli.

Il metodo di Quine-McCluskey permette di passare dalla tavola di verità ad un circuito a 2 livelli.

Esso consiste nel trasformare la tavola di verità in una espressione booleana di uno dei 2 tipi seguenti:

- somma di prodotti

$$f(x_1, x_2, \ldots, x_n) = \sum_i \dot{x}_{i1}\, \dot{x}_{i2} \ldots \dot{x}_{in}$$

- prodotto di somme

$$f(x_1, x_2, \ldots, x_n) = \prod_i \dot{x}_{i1} + \dot{x}_{i2} + \ldots + \dot{x}_{in}$$

Circuiti a 2 livelli (II)

Se tutte le variabili di ingresso sono disponibili affermate e negate, tali espressioni sono direttamente trasformabili in circuiti a 2 livelli composti esclusivamente di porte AND e OR.

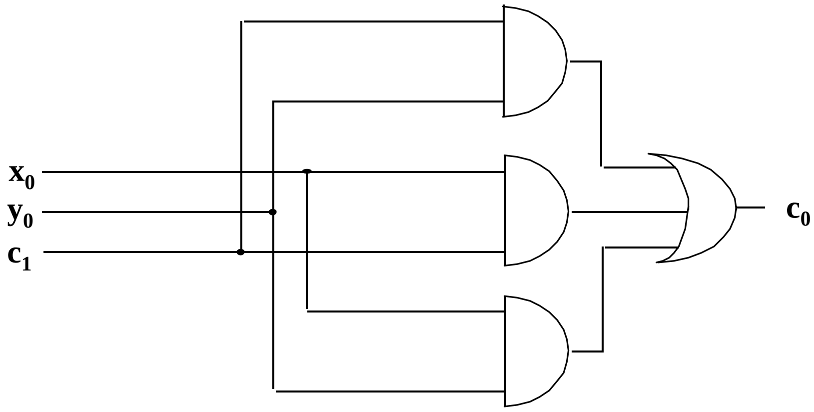
Esempio

La funzione svolta da un sommatore a 2 bit può essere scritta come

$$c_0 = x_0 c_1 + y_0 c_1 + x_0 y_0$$

che si può trasformare nel circuito seguente:

Circuiti a 2 livelli (III)

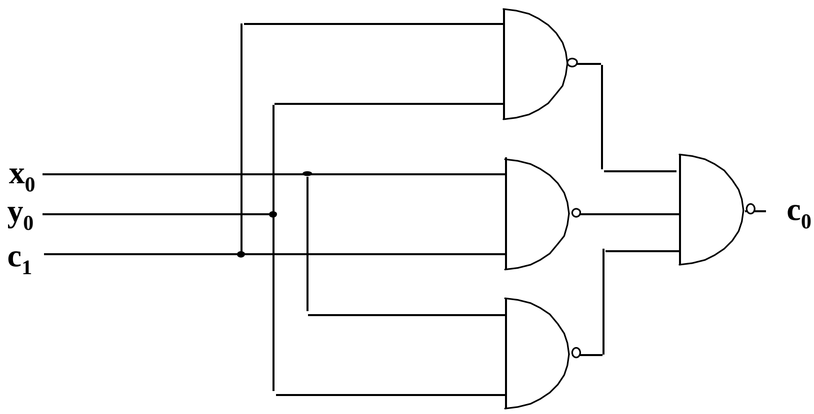


$c_0 = x_0 c_1 + y_0 c_1 + x_0 y_0$

23

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

Circuiti a 2 livelli (IV)



Utilizzando la legge di De Morgan: $c_0 = \overline{\overline{x_0 c_1} \bullet \overline{y_0 c_1} \bullet \overline{x_0 y_0}}$

24

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

Terminologia

letterale	$\dot{x}_i$
minterm	$\dot{x}_1 \, \dot{x}_2 \ldots \dot{x}_n$
maxterm	$\dot{x}_1 + \dot{x}_2 + \ldots + \dot{x}_n$
cubo	$\dot{x}_i \ldots \dot{x}_j \ldots \dot{x}_k$

Canonicità

Qualsiasi funzione booleana possiede due possibili forme canoniche:

- **come *somma di prodotti*, ove tutti i prodotti sono minterm**
- **come *prodotto di somme*, ove tutte le somme sono maxterm.**

Minimizzazione

Un circuito combinatorio si dice *minimale* se è vera una delle seguenti condizioni (tra loro equivalenti):

- **il circuito è composto dal minimo numero possibile di porte logiche**
- **il circuito implementa una funzione espressa in forma di somma di prodotti, nella quale**
  - **il numero di prodotti è minimo**
  - **nessun letterale può esssere cancellato da un prodotto senza cambiare la funzione**
- **una analoga condizione relativa alla forma prodotto di somme.**

27

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

Minimizzazione (II)

Il problema della minimizzazione di un circuito combinatorio viene normalmente risolto con strumenti automatici (ad es. *ESPRESSO*).

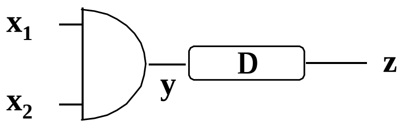
Per circuiti combinatori di piccole dimensioni (fino a 5 variabili di ingresso) il metodo delle *Tavole di Karnaugh* permette di passare dalla tavola della verità al circuito minimizzato.

28

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

Circuiti Sequenziali

- Implementano funzioni dipendenti dal tempo.
- Sono in grado di memorizzare informazioni.
- Sfruttano i ritardi delle porte:



$z(t+D)=x_1(t)x_2(t)$

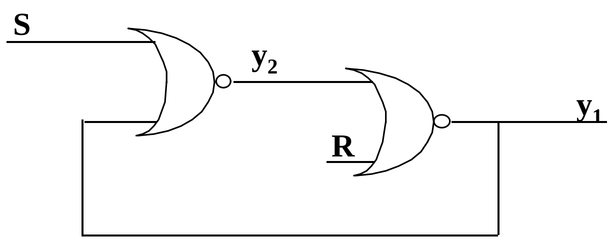
Il ritardo associato alla linea *y* persiste per un tempo D.

29

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

Flip-Flop

È in grado di mantenere l'informazione per un periodo illimitato di tempo.



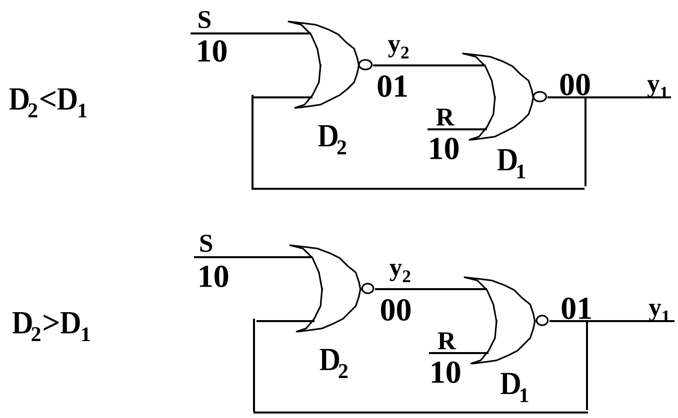
Comportamento:

- S=R=0: mantiene il valore su y_1
- S=0, R=1: forza uno 0 su y_1
- S=1, R=0: forza un 1 su y_1
- S=1, R=1: configurazione vietata

30

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

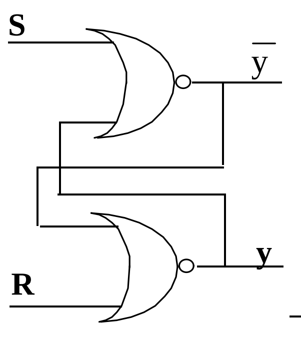
Transizione S=R=1 Þ S=R=0



31

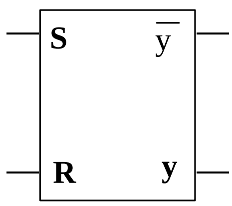
M. Sonza Reorda - a.a. 2001/02

Flip-Flop SR asincrono



Schema

Simbolo



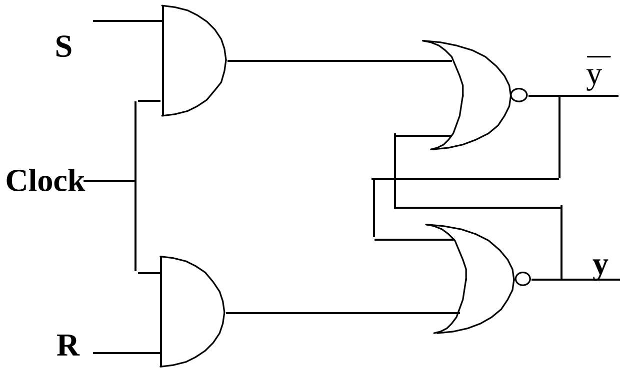
	SR 00	01	10
y 0	0	0	1
1	1	0	1

Tavola degli Stati

32

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

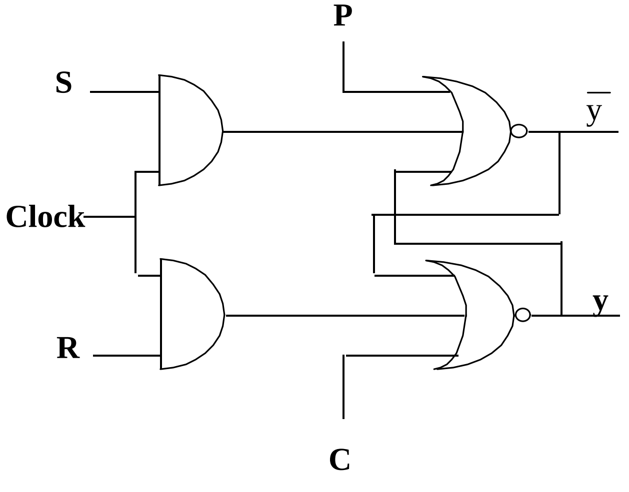
Flip-Flop Sincrono



33

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

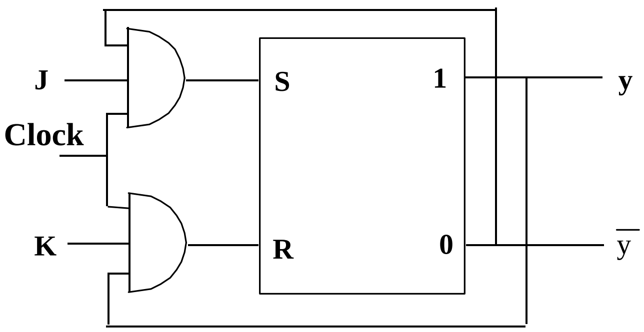
Flip-Flop SR con Clock, Preset e Clear asincroni



34

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

Flip-Flop JK



Comportamento:

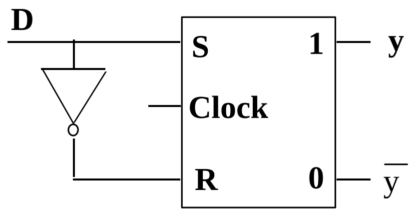
- **J=K=0: mantiene il valore di y**
- **J=0,K=1: clear**
- **J=1,K=0: preset**
- **J=K=1: complementa il valore di y.**

La combinazione J=K=1 non è più proibita.

35

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

Flip-Flop D



		D = 0	D = 1
y	0	0	1
	1	0	1

Tavola degli Stati

Ad ogni colpo di clock, il valore di D viene memorizzato.

36

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

Circuiti Sincroni

Il funzionamento di un circuito sequenziale dipende dai valori relativi dei ritardi delle porte logiche nella rete combinatorie (*corse* o *race*).

Questo può essere evitato facendo in modo che la memorizzazione dei valori delle Y_i avvenga per tutti i Flip-Flop in uno stesso istante, determinato dal valore di un segnale comune (*clock*).

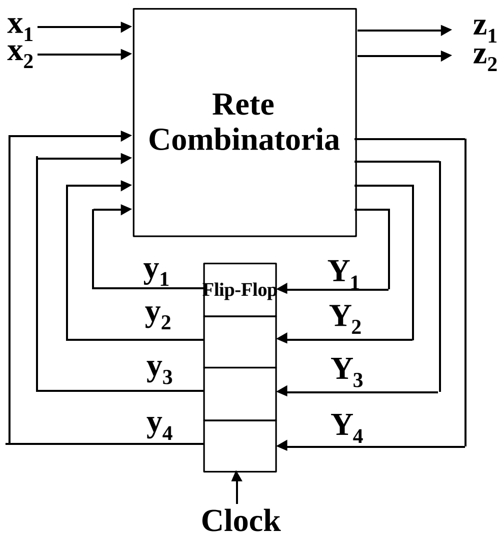
Tale segnale ha un andamento ad onda quadra la cui frequenza dipende dal ritardo massimo attraverso la rete combinatoria.

I circuiti dotati di clock si dicono *sincroni*.

37

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

Modello di Huffman



38

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

Tabella degli Stati

Un circuito sequenziale sincrono evolve attraverso *stati*, determinati dai valori presenti nei Flip-Flop.

Il suo comportamento può quindi essere descritto attraverso una *tabella degli stati*, che descrive le transizioni tra stati in funzione del valore sugli ingressi.

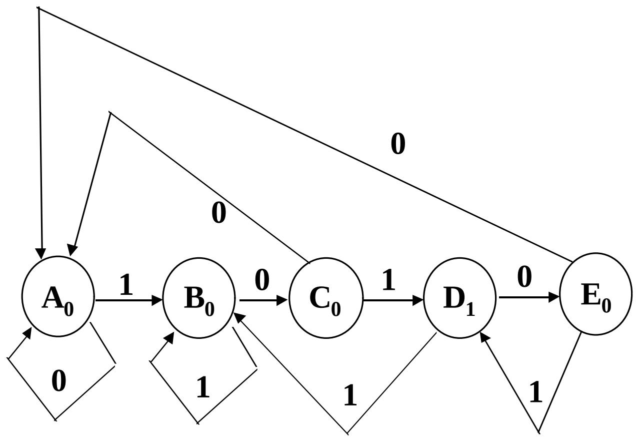
Progetto di Circuiti Sequenziali Sincroni

- **Costruzione Tavola degli Stati**
- **Minimizzazione (Eliminazione Stati Equivalenti)**
- **Assegnazione degli Stati**
- **Costruzione Tavola della Verità Rete Combinatoria**
- **Sintesi Rete Combinatoria**

Esempio

Consideriamo il riconoscitore della sequenza 1-0-1.

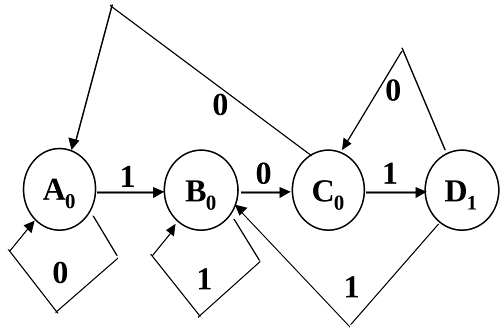
Il diagramma degli stati è



41

Esempio (II)

Lo stato E è equivalente allo stato C; il diagramma può quindi essere così semplificato:



42

Esempio (III)

Una possibile codifica degli stati è:

A	**00**
B	**01**
C	**10**
D	**11**

Esempio (IV)

La funzione di transizione degli stati è:

f(0,00) ® 00

f(1,00) ® 01

f(0,01) ® 10

f(1,01) ® 01

f(0,10) ® 00

f(1,10) ® 11

f(0,11) ® 10

f(1,11) ® 01

Il valore dell'uscita può essere facilmente generato tenendo conto che esso è 1 solo quando il circuito è nello stato 11:

h(-,00) ® 0

h(-,01) ® 0

h(-,10) ® 0

h(-,11) ® 1

Esempio (V)

Si possono ora sintetizzare le 3 funzioni f_1, f_2 e h, ottenendo:

$$f_1=y_1'y_2i'+y_1y_2'i+y_1y_2i'$$

$$f_2=i$$

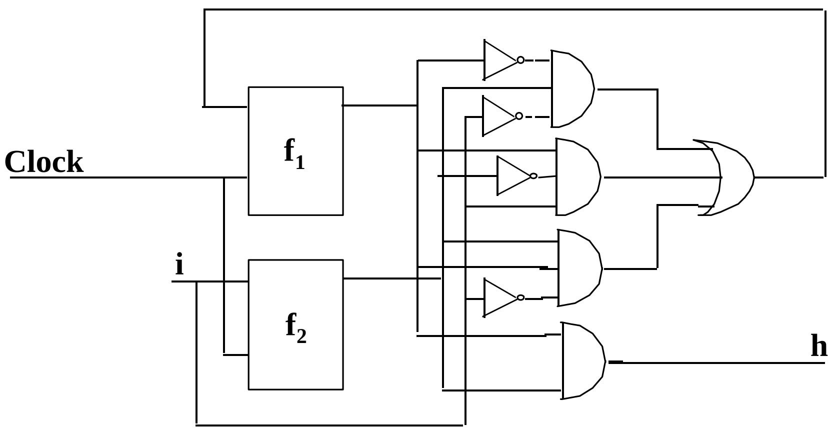
$$h=y_1y_2$$

45

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

Esempio (VI)

Il circuito che implementa il riconoscitore di sequenza è:



46

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