

Calcolatori Elettronici A

a.a. 2008/2009

RETI SEQUENZIALI: ESERCIZI

Massimiliano Giacomini

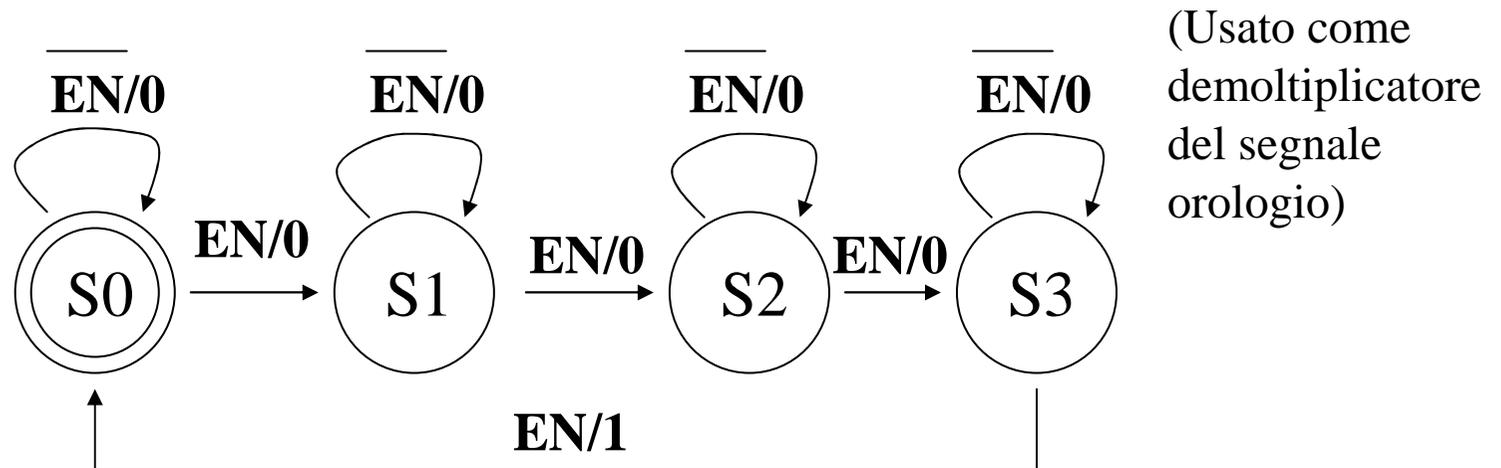
Esercizio 1: implementazione di contatori

- Un contatore è un dispositivo sequenziale che aggiorna periodicamente il suo **stato** secondo una regola che riprende il susseguirsi dei numeri naturali, ad esempio:

0000 – 0001 – 0010 – 0011 – 0100 - ...

- Prendiamo qui in considerazione solo contatori sincroni, ovvero che aggiornano il loro stato **ad ogni ciclo di clock**
- **Contatore modulo k**: conta da 0 a k-1 e pone l'uscita a 1 quando arriva a k-1
⇒ un contatore modulo k può essere usato come 'demoltiplicatore' del segnale orologio: produce un segnale di uscita pari a 1 ogni k cicli di clock

ESERCIZIO 1A: contatore modulo k



Stati = {S0, S1, S2, S3}

S0 : conteggio = 0

S1 : conteggio = 1

S2 : conteggio = 2

S3 : conteggio = 3

Ingressi = {EN} segnale di abilitazione

Uscite = {0, 1}

Sono necessari 2 bit per codificare gli stati e le combinazioni corrispondono esattamente a 0, 1, 2, 3 in binario

S0 \Rightarrow 00

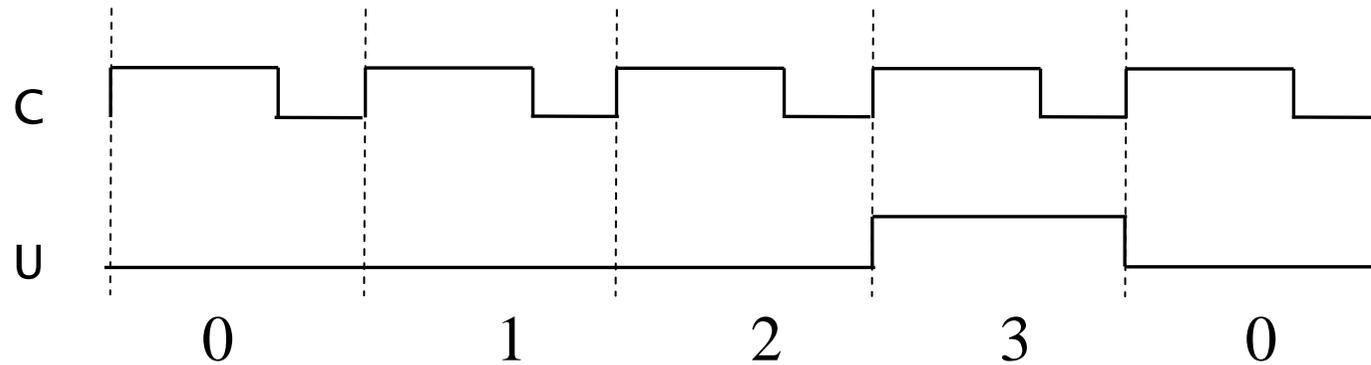
S1 \Rightarrow 01

S2 \Rightarrow 10

S3 \Rightarrow 11

Realizzazione circuitale lasciata per esercizio

Diagramma temporale con EN sempre attivo



Domanda: cosa succede durante il ciclo 3 se EN passa a 0?
E con il modello di Moore?

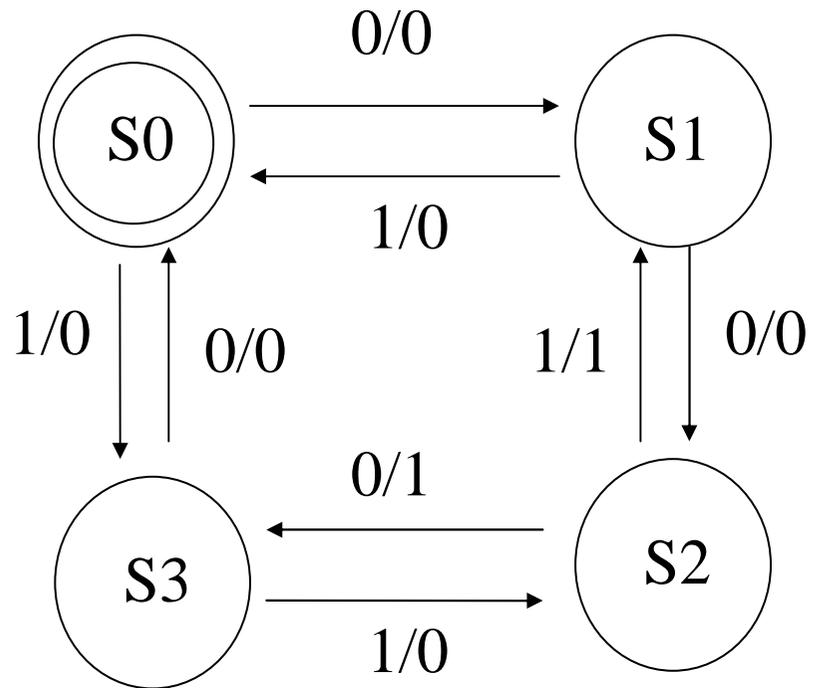
ESERCIZIO 1B: contatore modulo 4 bidirezionale

Progettare un contatore modulo 4 che conti sia in avanti sia all'indietro, a seconda del valore della variabile di ingresso x :

- Se $x = 0$ il circuito conta in avanti, se $x = 1$ il circuito conta all'indietro
- Se il valore del conteggio è 2 la variabile di uscita z viene posta a 1, altrimenti viene posta a 0

La transizione del conteggio avviene ad ogni impulso di clock (a seconda del valore di x e del valore attuale del conteggio)

Macchina a stati finiti e assegnazioni



4 stati \rightarrow 2 flip-flop

| Stato | y_1 | y_0 |
|-------|-------|-------|
| S0 | 0 | 0 |
| S1 | 0 | 1 |
| S2 | 1 | 0 |
| S3 | 1 | 1 |

Tabella delle transizioni e delle uscite

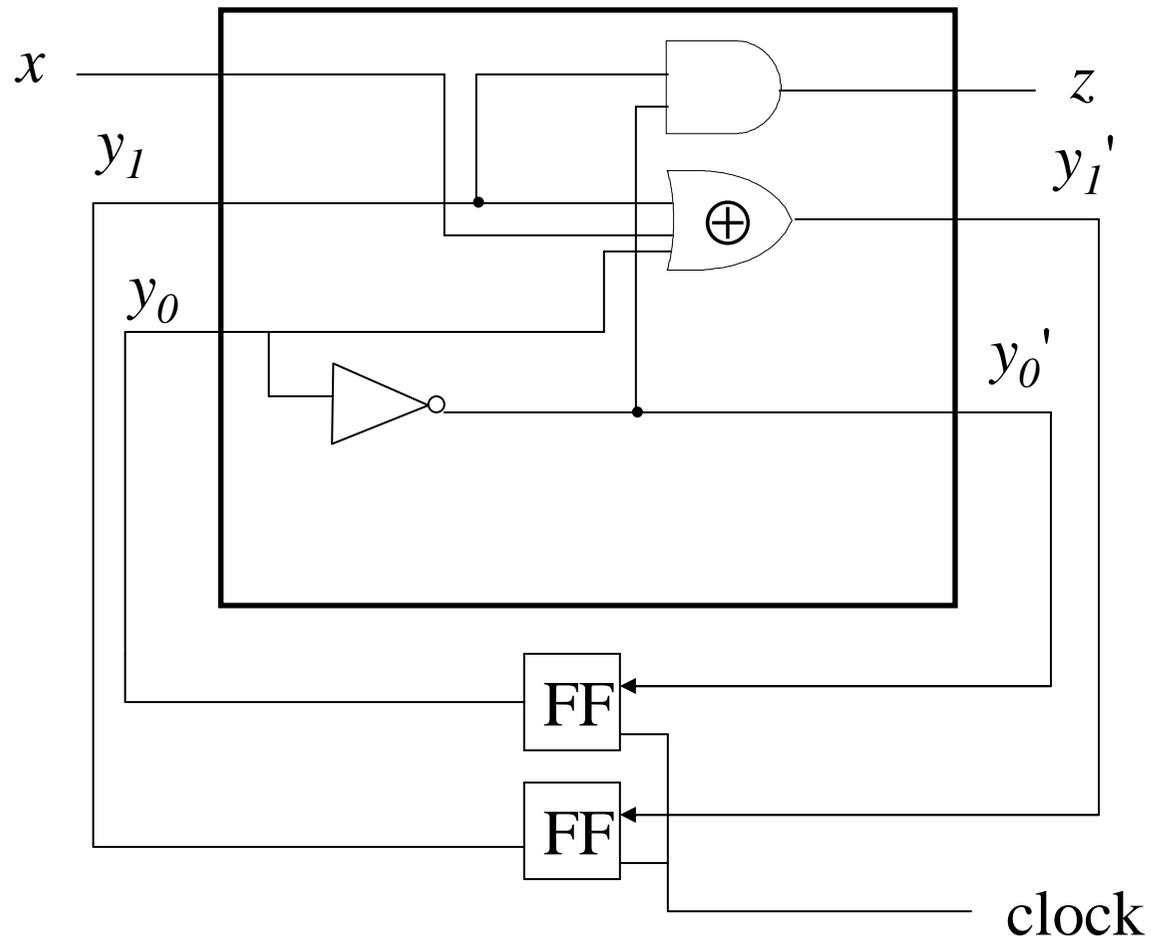
| <i>Stato presente</i> | | <i>Ingresso</i> | <i>Stato futuro</i> | | <i>Uscita</i> |
|-----------------------|-------|-----------------|---------------------|--------|---------------|
| y_1 | y_0 | x | y_1' | y_0' | z |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |

$$z = y_1 \bar{y}_0$$

$$y_0' = \bar{y}_1 \bar{y}_0 + y_1 \bar{y}_0 = \bar{y}_0$$

$$y_1' = \bar{y}_1 \bar{y}_0 x + \bar{y}_1 y_0 \bar{x} + y_1 \bar{y}_0 \bar{x} + y_1 y_0 x = y_1 \oplus y_0 \oplus x$$

Macchina sequenziale per il contatore bidirezionale



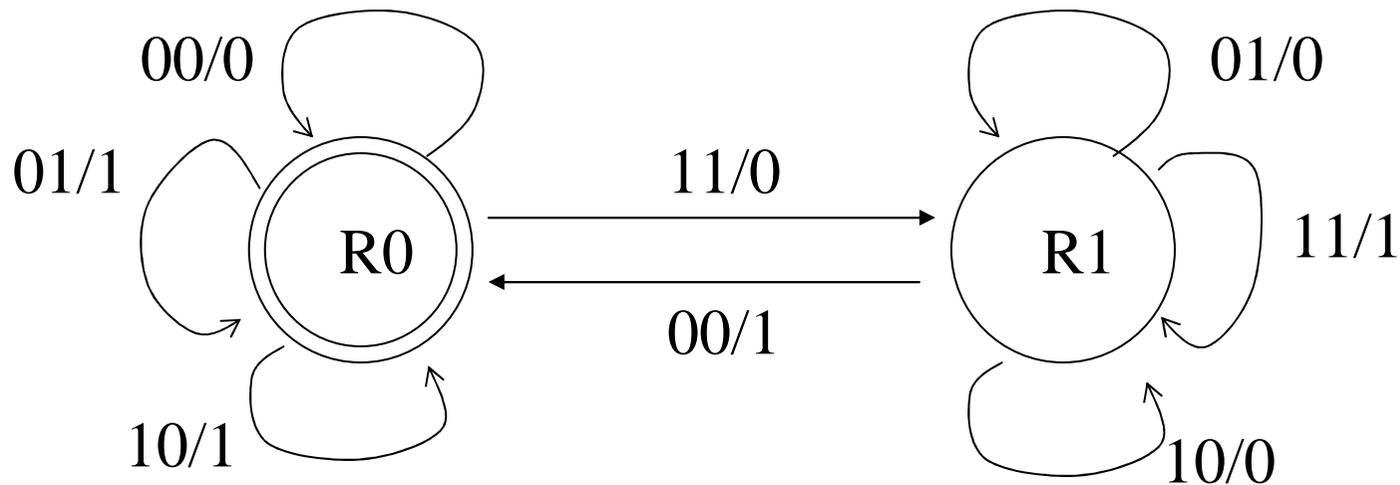
Esercizio 2: implementazione di un sommatore sequenziale

Progettare un sommatore che ad ogni ciclo di clock riceva in ingresso la coppia dei bit corrispondenti di due operandi da sommare, a partire dalla posizione meno significativa.

Il sommatore deve restituire in uscita, nello stesso ciclo di clock, la cifra corrispondente del risultato

Ingressi, uscite, automa

- Vengono sommate coppie di bit a_i e b_i di due operandi
 $I = \{00, 01, 10, 11\}$
- L'uscita c_i appartiene a $U = \{0, 1\}$
- Si distinguono 2 stati: “riporto presente” (R1) e “riporto assente” (R0)



| Tabella degli stati corrispondente all'automa | stato | $a_i b_i$ | | | |
|---|-------|-----------|------|------|------|
| | | 00 | 01 | 10 | 11 |
| | R0 | R0,0 | R0,1 | R0,1 | R1,0 |
| | R1 | R0,1 | R1,0 | R1,0 | R1,1 |

Assegnamento degli stati

| Stato | y | |
|-------|-----|--|
| R0 | 0 | Usiamo un flip-flop di tipo D che memorizza il valore della variabile di stato y nell'intervallo compreso fra due fronti (di salita) successivi del clock |
| R1 | 1 | |

Tabella delle transizioni e delle uscite

| a_i | b_i | y | y' | c_i |
|-------|-------|-----|------|-------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Formula per il flip-flop D

$$y' = b_i y + a_i \bar{y} + a_i b_i$$

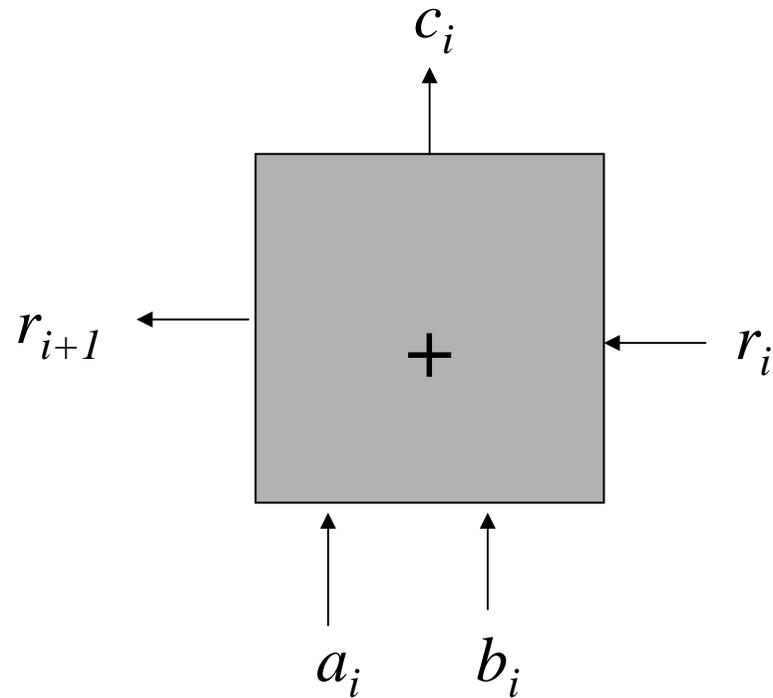
Formula per la variabile di uscita

$$c_i = \bar{a}_i \bar{b}_i y + \bar{a}_i b_i \bar{y} + a_i \bar{b}_i y + a_i b_i y$$

In pratica, c_i è la somma di a_i , b_i ed y , y' è il corrispondente riporto

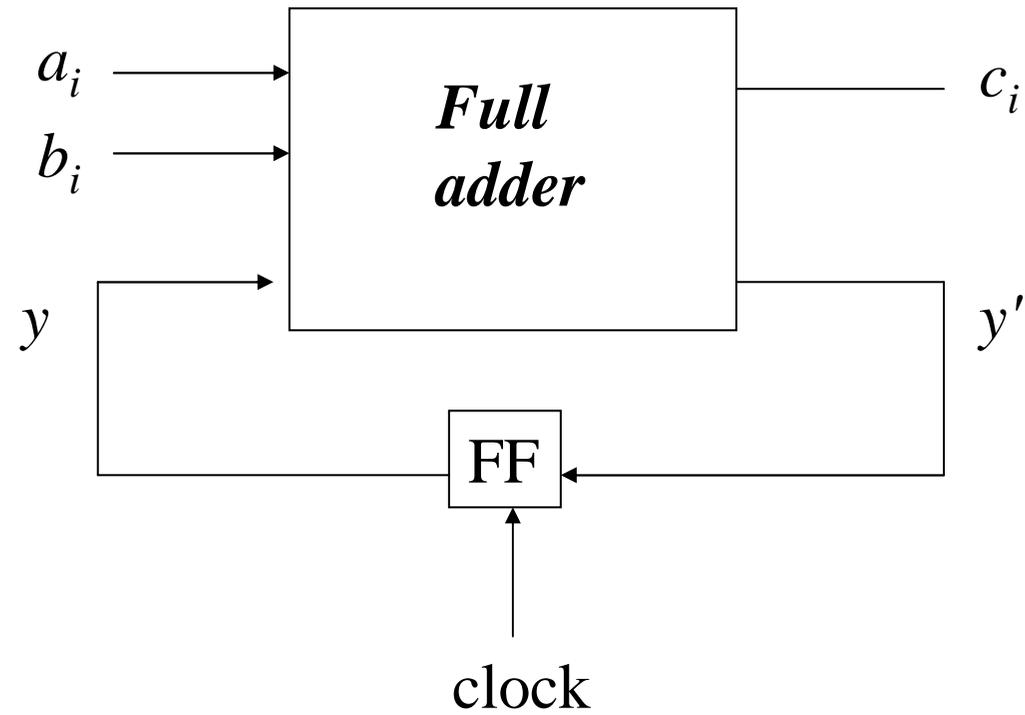
➡ *Sono le espressioni delle uscite di un full adder !!!*

Full Adder



In ingresso due cifre da sommare e il riporto,
in uscita la cifra e il riporto risultanti

Macchina sequenziale per il sommatore a 2 stati



Esercizio 3: riconoscitore di sequenze

Progettare una macchina che:

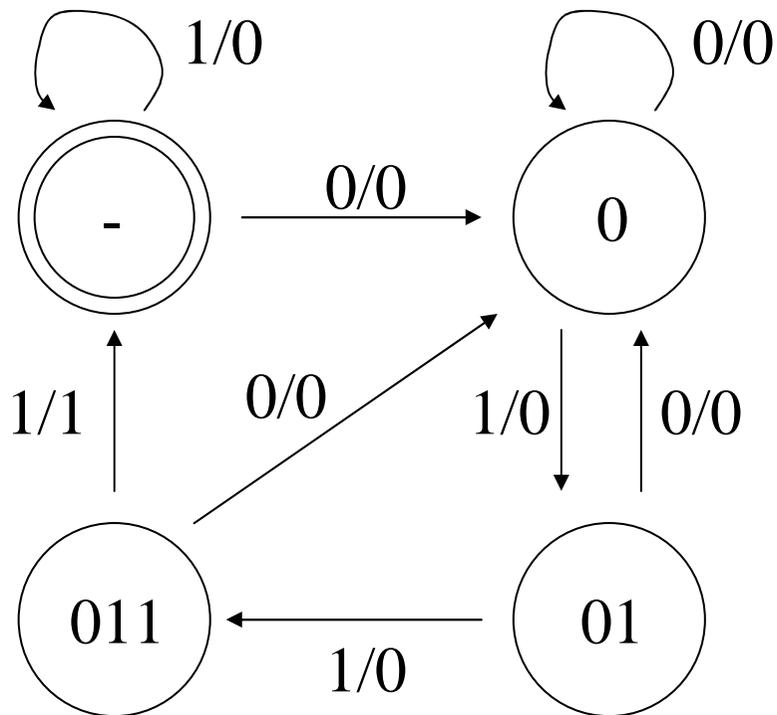
- Legge sequenze di 0 e 1 di lunghezza arbitraria
(le cifre sono lette successivamente ad ogni ciclo di clock)
- Produce in uscita 1 ogni volta che ha riconosciuto la sequenza

0111

(durante il ciclo di clock in cui viene ricevuto l'ultimo 1)

$$I=\{0,1\}, U=\{0,1\}$$

- Identifichiamo **4 stati** che indichiamo con “-”, “0”, “01” e “011”: i nomi rappresentano la parte della sottosequenza finora riconosciuta
- Lo stato “-” è lo stato iniziale: nessun carattere della sottosequenza cercata è stato riconosciuto



Per 4 stati bastano 2 flip-flop (tipo D)

Facciamo il seguente assegnamento degli stati

| Stato | y_1 | y_0 |
|-------|-------|-------|
| “-” | 0 | 0 |
| “0” | 0 | 1 |
| “01” | 1 | 0 |
| “011” | 1 | 1 |

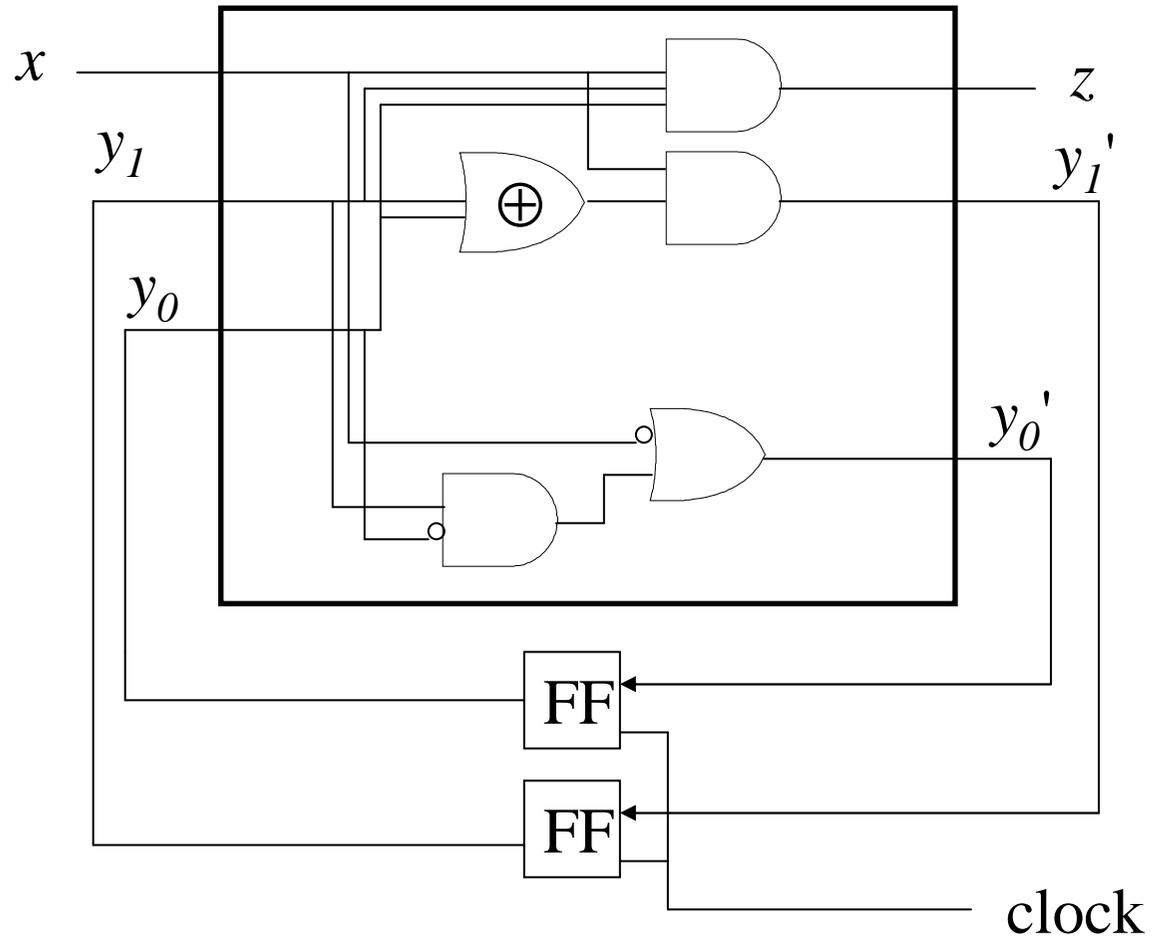
Tabella delle transizioni e delle uscite

| <i>Stato presente</i> | | <i>Ingresso</i> | <i>Stato futuro</i> | | <i>Uscita</i> |
|-----------------------|-------|-----------------|---------------------|--------|---------------|
| y_1 | y_0 | x | y_1' | y_0' | z |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |

$$y_1' = \bar{y}_1 y_0 x + y_1 \bar{y}_0 x = x(y_1 \oplus y_0) \qquad z = y_1 y_0 x$$

$$y_0' = \bar{y}_1 \bar{y}_0 \bar{x} + \bar{y}_1 y_0 \bar{x} + y_1 \bar{y}_0 \bar{x} + y_1 \bar{y}_0 x + y_1 y_0 \bar{x} = \bar{y}_1 \bar{x} + y_1 \bar{y}_0 + y_1 \bar{x} = \bar{x} + y_1 \bar{y}_0$$

Realizzazione circuitale



Esercizio 4: vantaggi nel tennis

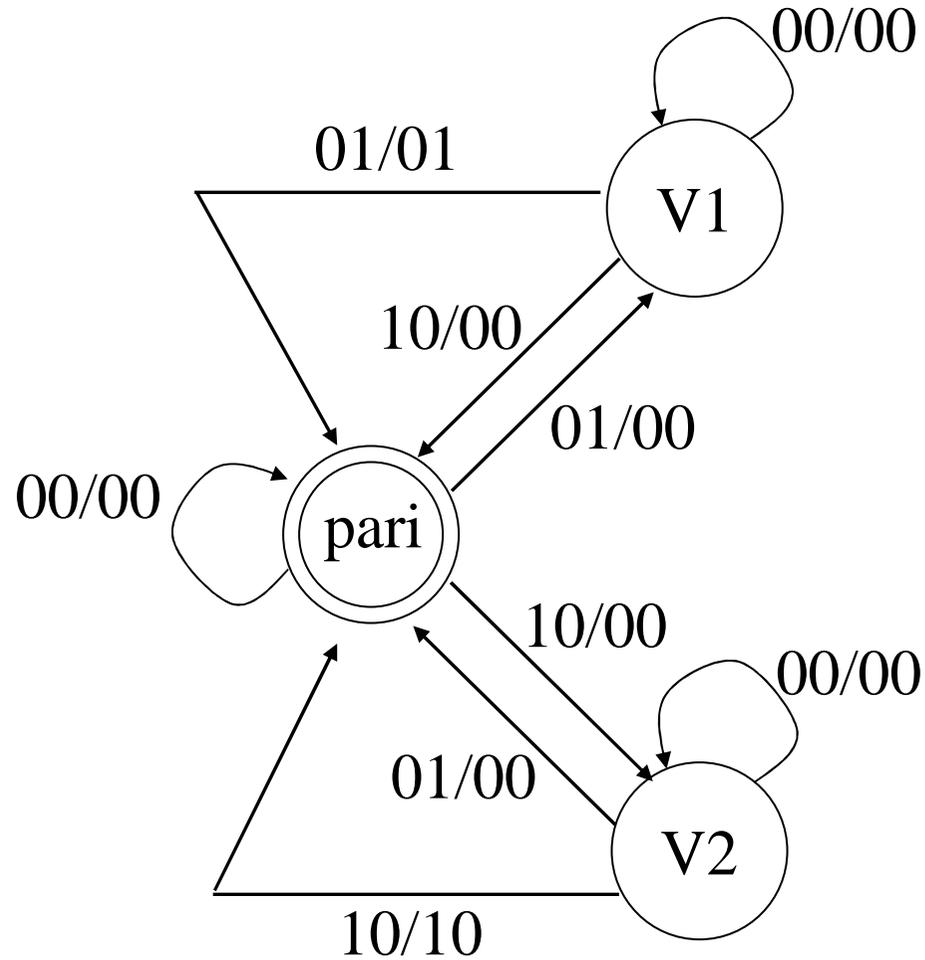
Progettare una macchina sequenziale che sia in grado di segnalare il vincitore di un game di tennis. Per semplicità, si assume di partire dal punteggio di 40 pari (e quindi il game viene vinto dal giocatore che supera l'altro di due punti).

La macchina riceve ad ogni ciclo di clock:

- 00 se nessuno dei due giocatori ha realizzato un nuovo punto
- 01 se il punto è realizzato dal giocatore 1
- 10 se il punto è realizzato dal giocatore 2

La macchina deve segnalare il vincitore ponendo la sua uscita a 01 (giocatore 1) o 10 (giocatore 2) e riportarsi nello stato iniziale.

$I = \{00, 01, 10\}$, $U = \{00, 01, 10\}$



| Stato | y_1 | y_0 |
|-------|-------|-------|
| pari | 0 | 0 |
| V1 | 0 | 1 |
| V2 | 1 | 0 |

Per esercizio:

- 1) proseguire fino alla realizzazione circuitale
- 2) scrivere l'automa che gestisce l'intero game (a partire da 0-0)

Esercizio 5: macchina distributrice

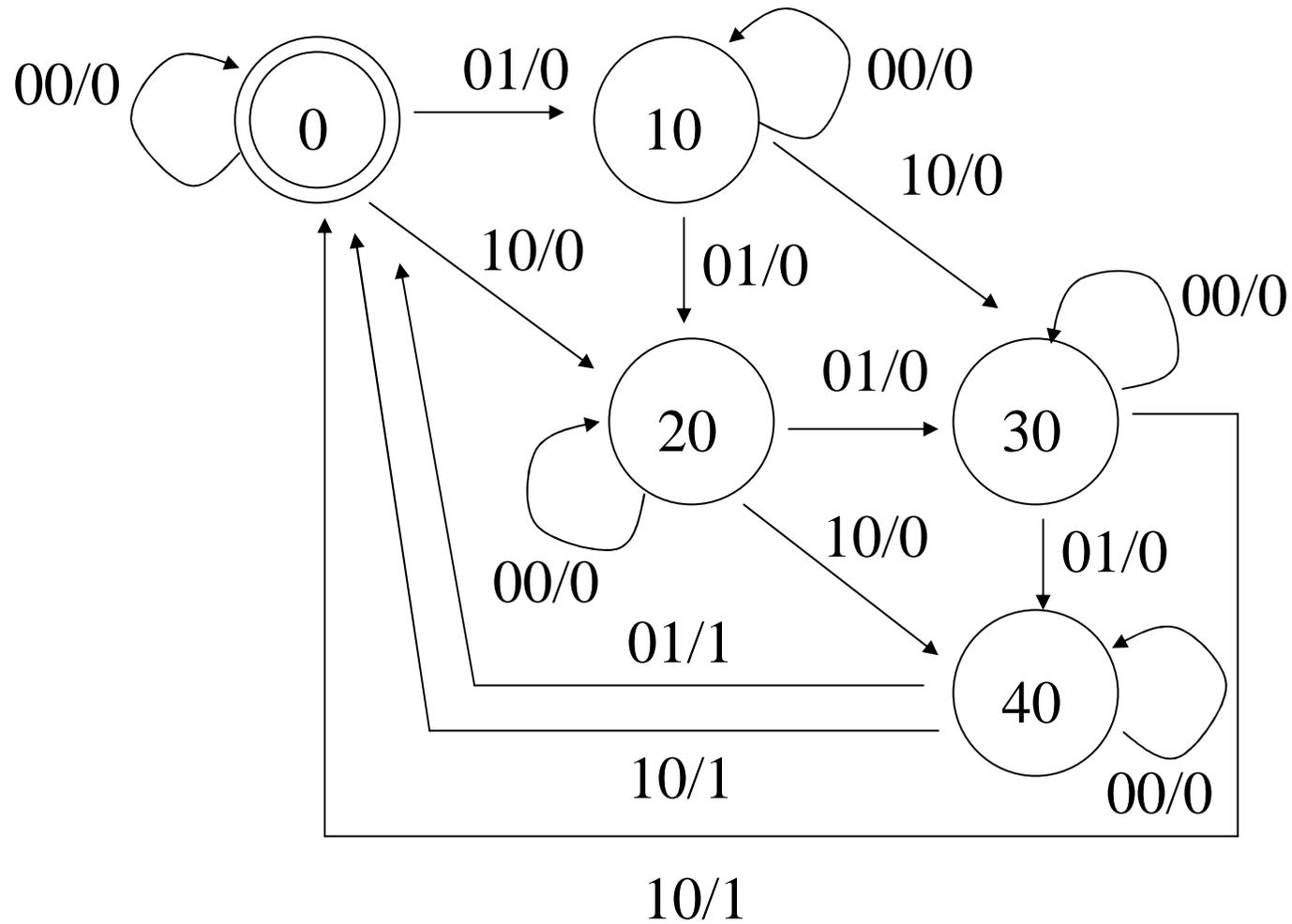
- La macchina accetta monete da 10 e 20 centesimi
- **Codifichiamo gli ingressi** con le variabili x_1 e x_2

| x_2x_1 | |
|----------|----------------|
| 0 0 | nessuna moneta |
| 0 1 | moneta da 10 |
| 1 0 | moneta da 20 |

$$I = \{00, 01, 10\}$$

- Fornisce il prodotto quando viene raggiunto l'importo di 50 (ponendo l'uscita 1)
- Convenzione: chiamiamo ogni stato della macchina con l'importo raggiunto

Automa per la macchina distributrice



Assegnamento degli stati

- Per 5 stati sono necessari 3 flip-flop (tipo D): y_2, y_1, y_0
- Facciamo il seguente **assegnamento degli stati**

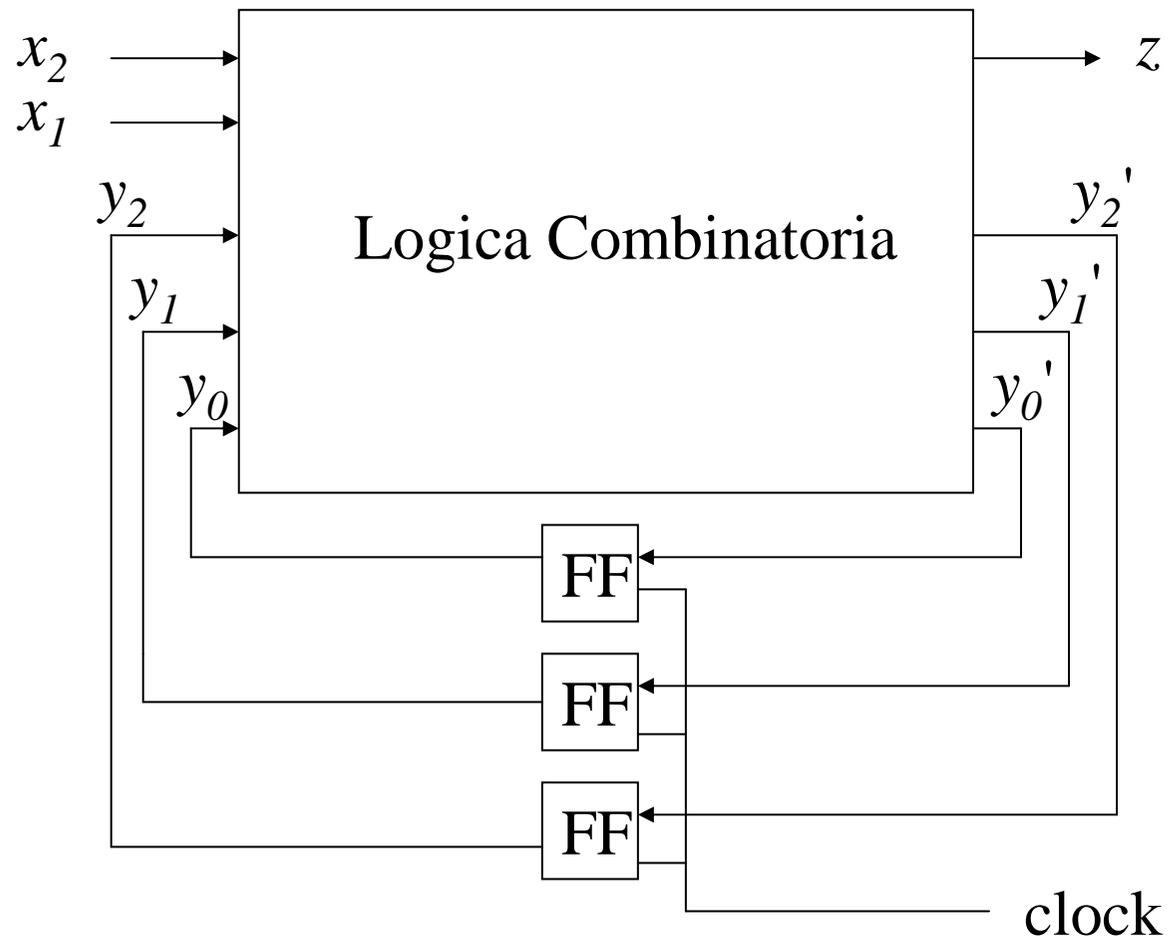
| Stato | y_2 | y_1 | y_0 |
|-------|-------|-------|-------|
| “0” | 0 | 0 | 0 |
| “10” | 0 | 0 | 1 |
| “20” | 0 | 1 | 0 |
| “30” | 0 | 1 | 1 |
| “40” | 1 | 0 | 0 |

- Deriviamo dall'automata la tabella delle transizioni e delle uscite

Tabella delle transizioni e delle uscite

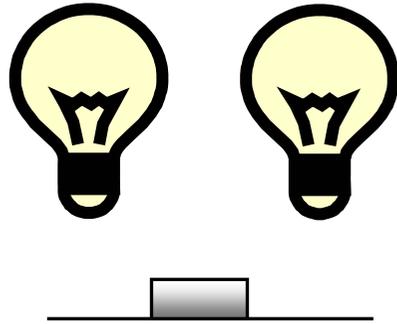
| Stato presente | | | Ingressi | | Stato prossimo | | | Uscite |
|----------------|-------|-------|----------|-------|----------------|--------|--------|--------|
| y_2 | y_1 | y_0 | x_2 | x_1 | y_0' | y_1' | y_2' | z |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

Macchina sequenziale risultante



Per esercizio: realizzare con PLA la logica combinatoria

Esercizio proposto: doppia lampada con tre modalità di accensione



Nella situazione iniziale entrambe le lampade sono spente. Poi:

- se si preme l'interruttore si accende la lampadina sinistra;
- se si preme di nuovo l'interruttore si accende la lampadina destra e si spegne quella sinistra;
- se si preme l'interruttore ancora una volta le due lampadine si accendono entrambe;
- infine, con un ultimo giro, le lampadine si spengono entrambe.

NB: disegnare gli automi con il modello di Mealy e Moore