

# Reti sequenziali

## RAM, register file, FSM

Esercitazione  
Architettura degli elaboratori

# Esercizio 1

- Considerando una RAM di dimensione uguale a 4096 Kbit e una parola di memoria di 4 byte, qual è l'altezza di tale memoria?
- Da quanti bit è composto un indirizzo per ciascuna parola di tale memoria?

# Esercizio 1 – soluzione

- Considerando una RAM di dimensione uguale a 4096 Kbit e una parola di memoria di 4 byte, qual è l'altezza di tale memoria?
- Da quanti bit è composto un indirizzo per ciascuna parola di tale memoria?



- $4096 \text{ Kbit} = 4 * 2^{10} \text{ Kbit} = 2^{22} \text{ bit}$
- L'altezza della memoria è:  $2^{22} / 4 * 8 \text{ bit} = 2^{22} / 2^2 * 2^3 = 2^{22} / 2^5 = 2^{17}$
- Ogni indirizzo di questa memoria è rappresentato su 17 bit (perché con 17 bit posso avere fino a  $2^{17}$  configurazioni distinte di bit dove ciascuna potrà far riferimento ad una parola di memoria distinta)

## Esercizio 2

Qual è la capacità (in MB) di una memoria con indirizzi di parola da 30 bit e parole di memoria di 64 bit?

## Esercizio 2 – soluzione

Qual è la capacità (in MB) di una memoria con indirizzi di parola da 30 bit e parole di memoria di 64 bit?

- $2^{30}$  = numero di indirizzi di memoria
- 64 bit =  $2^6$  bit = dimensione di una parola di memoria
- Capacità della memoria (memoria indirizzabile) =  $2^{30} * 2^6$  bit
- Trasformo in MB dividendo per  $2^3$  (bit in 1 byte) \*  $2^{20}$  (bit in 1 Mbit)
- Dimensione di memoria in MB =  $2^{10} * 2^3 = 8 * 1024$  MB

## Esercizio 3

In relazione al tempo d'accesso di un dato a una memoria di tipo SRAM, quali delle seguenti affermazioni sono corrette?

1. È dipendente dalla posizione del dato in memoria
2. È dipendente dalla dimensione della memoria
3. È dipendente dal tipo di dato rappresentato

## Esercizio 3 – soluzione

In relazione al tempo d'accesso di un dato a una memoria di tipo SRAM, quali delle seguenti affermazioni sono corrette?

1. È dipendente dalla posizione del dato in memoria
2. È dipendente dalla dimensione della memoria
3. È dipendente dal tipo di dato rappresentato

Nessuna affermazione è corretta

## Esercizio 4

In relazione alle memorie RAM, quali delle seguenti affermazioni sono corrette? (rif. Pag. B-58)

1. Le memorie SRAM hanno generalmente dimensioni minori rispetto alle DRAM
2. I dati in una memoria SRAM sono disponibili solo mentre il computer è acceso
3. Le memorie DRAM richiedono un refresh periodico
4. Le memorie DRAM sono 5-10 volte più lente delle SRAM
5. In una memoria DRAM, il refresh ritarda di circa 1%-2% gli accessi a memoria



## Esercizio 4

In relazione alle memorie RAM, quali delle seguenti affermazioni sono corrette?

1. Le memorie SRAM hanno generalmente dimensioni minori rispetto alle DRAM
2. I dati in una memoria SRAM sono disponibili solo mentre il computer è acceso
3. Le memorie DRAM richiedono un refresh periodico
4. Le memorie DRAM sono 5-10 volte più lente delle SRAM
5. In una memoria DRAM, il refresh ritarda di circa 1%-2% gli accessi a memoria

Tutte le affermazioni sono corrette

## Esercizio 5

Considerando la seguente memoria e considerando che indirizzi e contenuto proseguano le sequenze esadecimali indicate, quale sarà il valore memorizzato all'indirizzo 0xFFFF0100?

**Indirizzi:**

0xFFFF0000

0xFFFF0020

0xFFFF0040

...

**Valori:**

0xFFFFFFFF

0xFFFFFFFFE

0xFFFFFFFFD

...

# Esercizio 5

**Indirizzi:**

0xFFFF0000

0xFFFF0020

0xFFFF0040

0xFFFF0060

0xFFFF0080

0xFFFF00A0

0xFFFF00C0

0xFFFF00E0

0xFFFF0100

**Valori:**

0xFFFFFFFF

0xFFFFFFFFE

0xFFFFFFFFD

0xFFFFFFFFC

0xFFFFFFFFB

0xFFFFFFFFA

0xFFFFFFFF9

0xFFFFFFFF8

0xFFFFFFFF7

## Esercizio 6

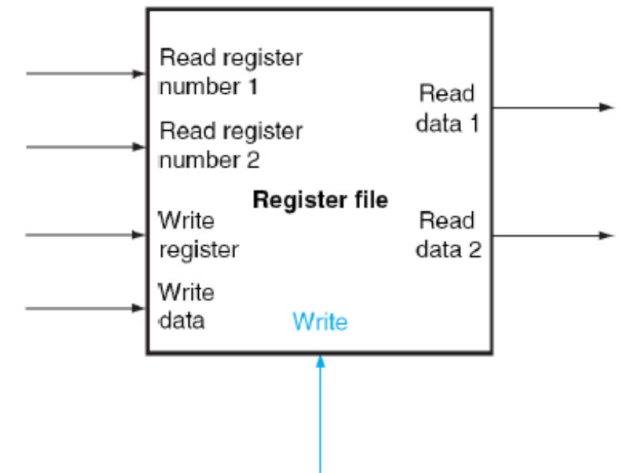
Quanti ingressi e quante uscite ha un register file che può leggere due valori alla volta e scriverne uno?

# Esercizio 6 – soluzione

Quanti ingressi e quante uscite ha un register file che può leggere due valori alla volta e scriverne uno?

- 2 uscite per i dati letti dai registri
- 5 ingressi per indicare
  - i due registri da leggere
  - il registro da scrivere
  - il valore da scrivere
  - il segnale di scrittura del registro

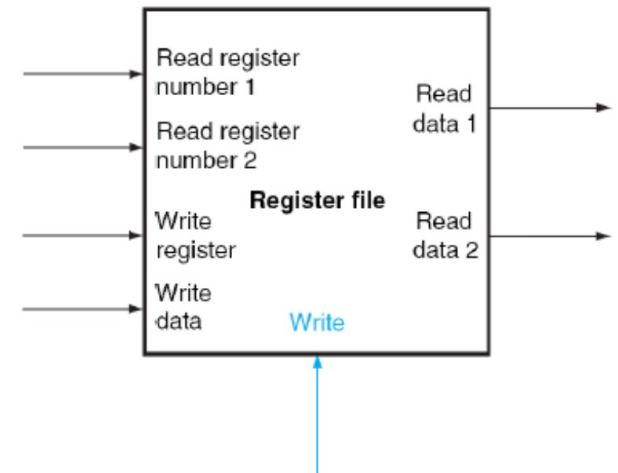
**FIGURE C.8.7**



# Esercizio 7

- Con riferimento al register file MIPS32 si supponga che gli ingressi siano così configurati
  - Read Register number 1 = 01000
  - Read Register number 2 = 01001
  - Write Register number = 01001
  - WriteData abbia valore 0x00000032
  - Write abbia valore 0
- Si consideri inoltre che i valori contenuti nei registri siano
  - \$8 = 0x0000001A
  - \$9 = 0x00000010
  - \$10 = 0x0000001B
  - \$11 = 0x00000009
- Successivamente al fronte attivo del clock, quando i segnali logici saranno ormai divenuti stabili, che valore si avrà su Read Data 1 e Read Data 2?

**FIGURE C.8.7**



# Esercizio 7 – soluzione

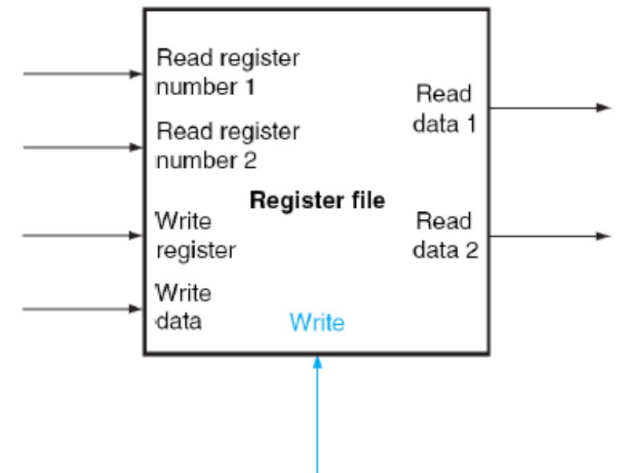
- Essendo Write = 0, non ci interessa la parte relativa alla scrittura
- Essendo
  - Read Register number 1 = 01000 (\$8)
  - Read Register number 2 = 01001 (\$9)
  - \$8 = 0x0000001A
  - \$9 = 0x00000010

successivamente al fronte attivo del clock, avremo i valori

- Read Data 1 = 0x0000001A
- Read Data 2 = 0x00000010
- Se invece **Write fosse = 1** il valore del registro \$9 sarebbe aggiornato con il valore di WriteData e quindi, successivamente al fronte attivo del clock, avremmo
  - \$8 = 0x0000001A
  - \$9 = 0x00000032

e quindi anche Read Data 2 = 0x00000032

FIGURE C.8.7



# Esercizio 8

Cos'è un segnale di clock?

1. Un segnale (onda sinusoidale) con un periodo predeterminato e costante
2. Un segnale (onda quadra) con un periodo non costante
3. Un segnale (onda sinusoidale) con un periodo non costante
4. Un segnale (onda quadra) con un periodo predeterminato e costante



# Esercizio 8 – soluzione

Cos'è un segnale di clock?

1. Un segnale (onda sinusoidale) con un periodo predeterminato e costante
2. Un segnale (onda quadra) con un periodo non costante
3. Un segnale (onda sinusoidale) con un periodo non costante
4. Un segnale (onda quadra) con un periodo predeterminato e costante

## Esercizio 9

Cosa viene misurato in Hertz?

1. Il periodo del segnale di clock
2. Il tempo fra un picco ed il successivo in un segnale di clock
3. Il valore massimo raggiunto da un segnale di clock
4. La frequenza del segnale di clock

## Esercizio 9 – soluzione

Cosa viene misurato in Hertz?

1. Il periodo del segnale di clock
2. Il tempo fra un picco ed il successivo in un segnale di clock
3. Il valore massimo raggiunto da un segnale di clock
4. La frequenza del segnale di clock

## Esercizio 10

- a) Considerando un segnale di clock con il periodo uguale a 0,02 microsecondi, qual è la sua frequenza in MHz?
- b) E per un segnale di clock con il periodo uguale a 10 nsec?

## Esercizio 10 – soluzione

- a) Considerando un segnale di clock con il periodo uguale a 0,02 microsecondi, qual è la sua frequenza in MHz?
- b) E per un segnale di clock con il periodo uguale a 10 nsec?

frequenza = 1/periodo                      1 hz = 1/sec

**1 sec =  $10^6$  microsecondi =  $10^9$  nanosecondi**

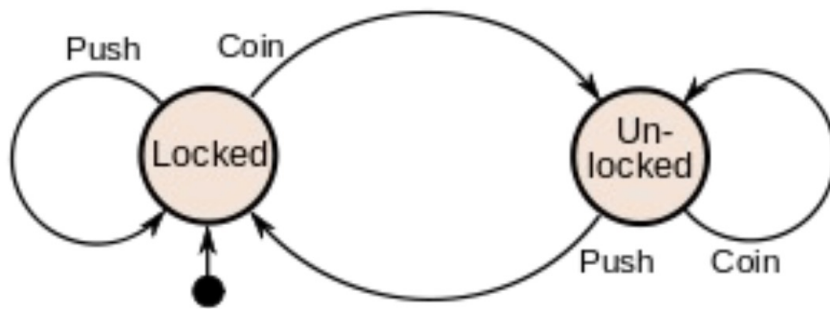
a) frequenza in Hz =  $1/(0,02 * 10^{-6}) = 50 * 10^6 \text{ Hz}$

→ frequenza in MHz = 50 MHz              essendo **1MHz =  $10^6 \text{ Hz}$**

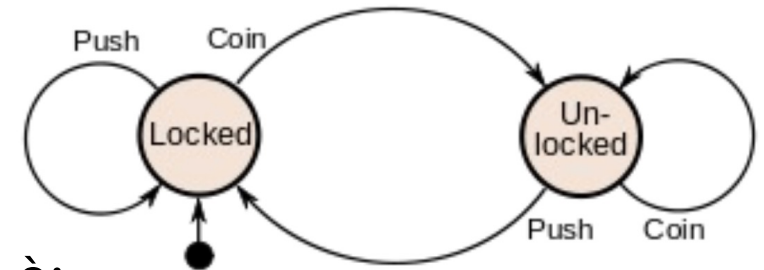
b) frequenza in MHz =  $1/(10 * 10^{-9}) / 10^6 = 10^8 / 10^6 \text{ MHz} = 100 \text{ MHz}$

# Esercizio 11

Qual è il comportamento di un **armadietto a moneta (coin locker)**, sapendo che la FSM che lo rappresenta è quella qui rappresentata?



# Esercizio 11 – soluzione



Una possibile descrizione del comportamento è:

- Un armadietto di tipo coin locker può trovarsi in uno di **due stati**: chiuso o aperto
- Lo stato dell'armadietto può essere modificato con **due possibili input**: l'inserimento di una moneta (coin) o la pressione sullo sportello (push)
- Quando l'armadietto è chiuso (stato locked)
  - si apre (passa allo stato unlocked) se viene inserita una moneta
  - resta chiuso se viene premuto lo sportello
- Quando l'armadietto è aperto (stato unlocked)
  - non cambia il suo stato se viene inserita una moneta
  - passa dallo stato unlocked allo stato locked se viene fatta pressione sullo sportello

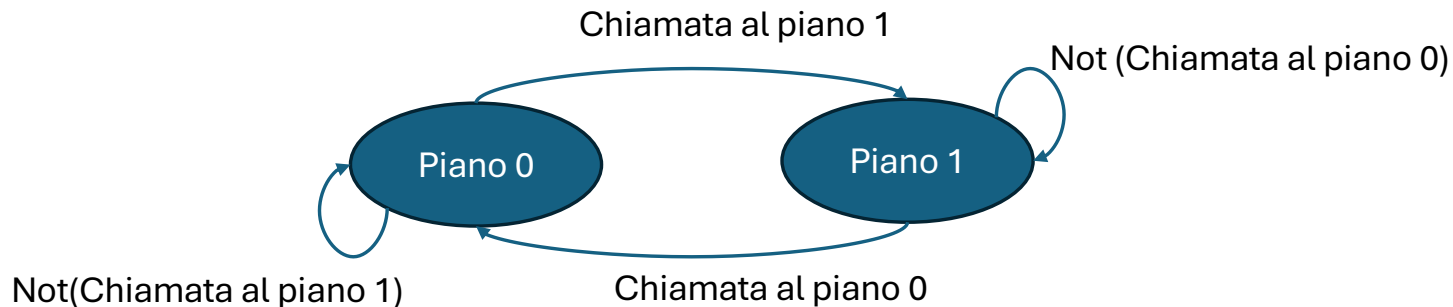
## Esercizio 12

Disegnare la FSM e la funzione next-state per descrivere il comportamento di un **ascensore che collega 2 piani**. Si supponga che, quando l'ascensore si trova ad un piano e viene premuto il pulsante di chiamata all'altro piano, l'ascensore si sposta da un piano all'altro. Quando è chiamato contemporaneamente ai due piani, prevale la chiamata al piano dove `_non_` si trova



# Esercizio 12 – soluzione (1/2)

- Stati: ascensore al piano 0, ascensore al piano 1
- Input: chiamata al piano 0, chiamata al piano 1
- Next-state-function: descrizione
  - se ascensore al p0 e chiamata al p1 (con chiamata o non chiamata al p0) → ascensore al p1
  - se ascensore al p0 e NON chiamata al p1 (con chiamata o non chiamata al p0) → ascensore al p0
  - se ascensore al p1 e chiamata al p0 (con chiamata o non chiamata al p1) → ascensore al p0
  - se ascensore al p1 e NON chiamata al p0 (con chiamata o non chiamata al p1) → ascensore al p1



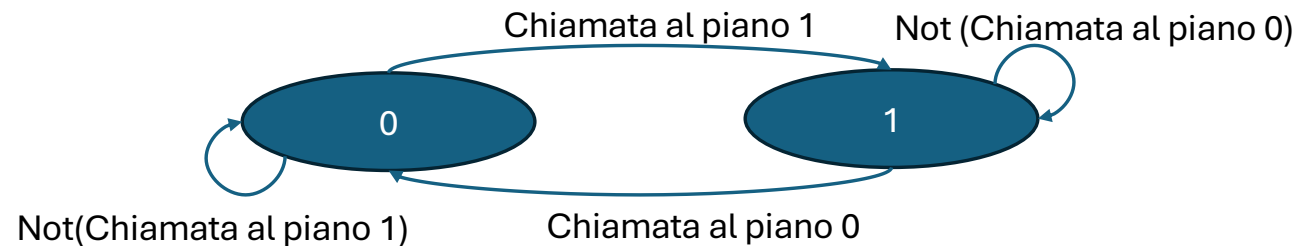
# Esercizio 12 – soluzione (2/2)

1 indica chiamata  
0 indica NON chiamata

Piano (state)	Chiam piano 0	chiam piano 1	Next state
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1



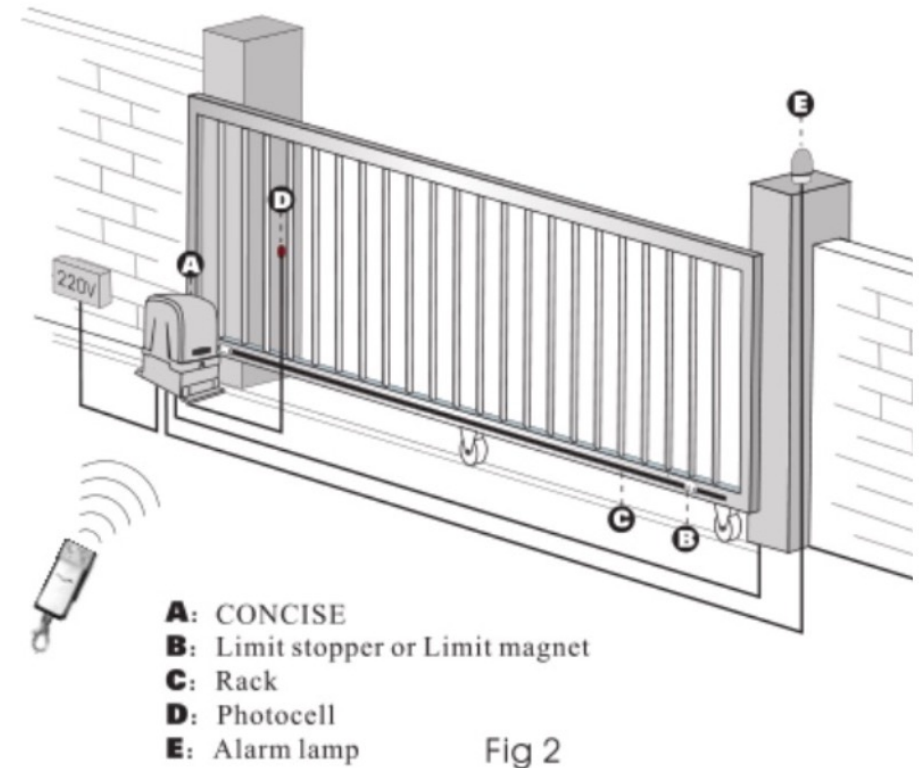
$$\begin{aligned}
 \text{Next\_state}(p, \text{ch0}, \text{ch1}) &= p * \text{not}(\text{ch0}) * \text{not}(\text{ch1}) + \\
 &\quad p * \text{not}(\text{ch0}) * \text{ch1} + \\
 &\quad \text{not}(p) * \text{not}(\text{ch0}) * \text{ch1} + \\
 &\quad \text{not}(p) * \text{ch0} * \text{ch1} = \\
 &= p * \text{not}(\text{ch0}) + \text{not}(p) * \text{ch1}
 \end{aligned}$$



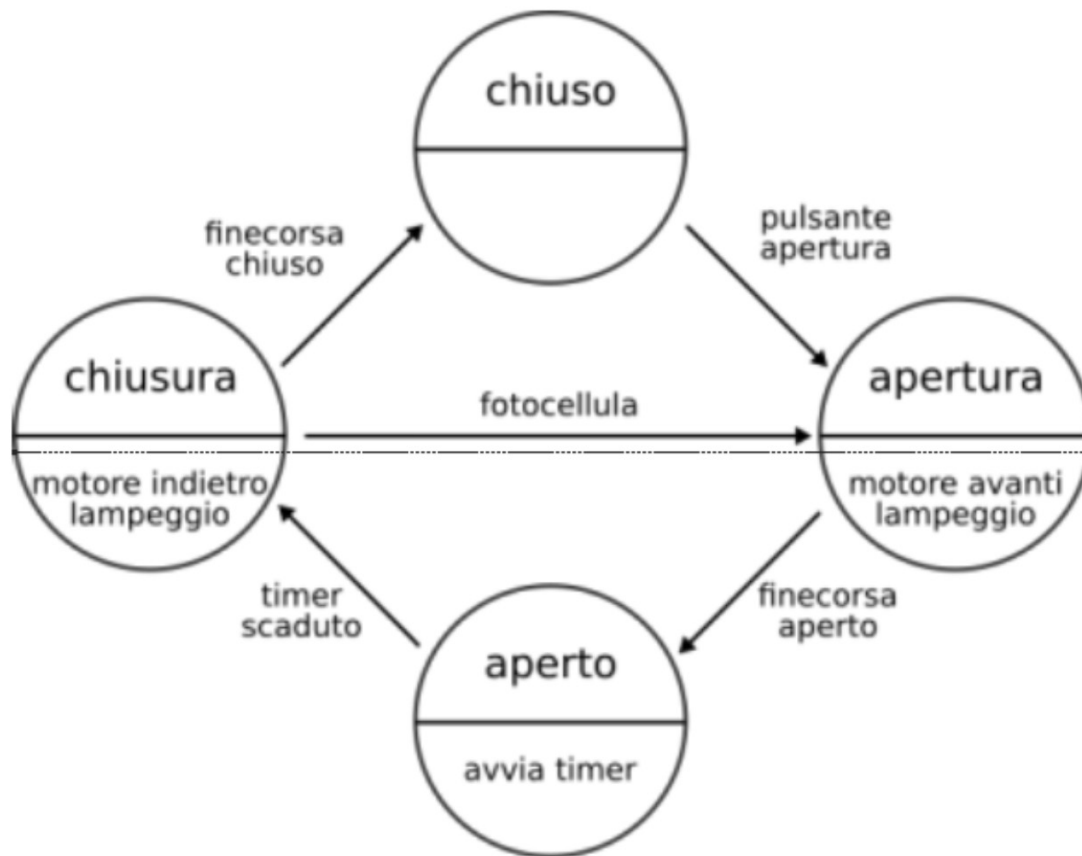
# Esercizio 13

Disegnare la FSM che descrive il comportamento di un **cancello automatico scorrevole** mosso da un motore in cui:

- Il movimento del motore avviene in due direzioni (apertura e chiusura) e si arresta quando si attivano due **finecorsa** (Fcc o Fca) che segnalano la completa chiusura o apertura
- L'apertura del cancello viene comandata da un **pulsante**
- La chiusura avviene automaticamente dopo lo scadere di un **timer** impostato a 10 secondi
- Una **fotocellula** apre automaticamente il cancello se rileva un ostacolo durante il movimento di chiusura
- Un segnalatore **lampeggiante** si accende durante il movimento del cancello



# Esercizio 13 – soluzione



FSM con 4 stati 5 input 3 output

stato	input	Next_state(stato,input)
chiuso	pulsante	apertura
apertura	FCA	aperto
aperto	timer_scaduto	chiusura
chiusura	fotocellula	apertura
chiusura	FCC	chiuso

stato	Output(stato)
Chiusura	Motore_indietro e segnalatore_luminoso
Chiuso	-
Apertura	Motore_avanti e segnalatore_luminoso
Aperto	Avvio_timer

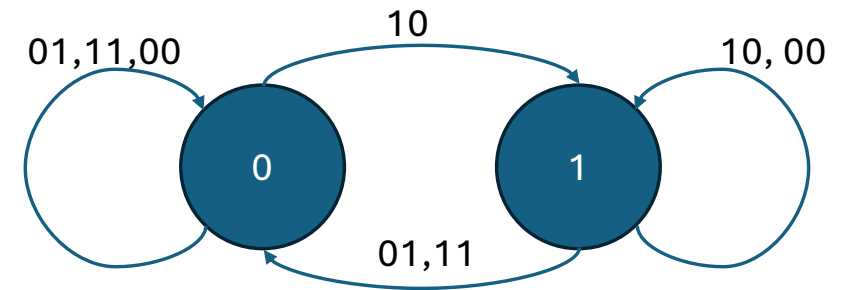
# Esercizio 14

Disegnare la macchina a stati finiti (FSM) e la tabella di verità per il controllo di un **motore elettrico** che:

- riceve i segnali da due pulsanti A e S
  - $A=1 \Rightarrow$  accendi
  - $S=1 \Rightarrow$  spegni
- in caso di pressione simultanea, S prevale
- se il motore è acceso (o spento) e arriva un altro segnale di accensione (o spegnimento), il segnale è ignorato
- l'output è un segnale binario F
  - $F = 0 \Rightarrow$  motore spento
  - $F = 1 \Rightarrow$  motore acceso

# Esercizio 14 – soluzione

- Macchina a stati finiti con 2 input e 2 stati (coincidenti ai possibili output):
  - F=0 : motore spento
  - F=1: motore acceso
- Per ogni stato, 4 transizioni possibili date dalle 4 possibili configurazioni dei 2 input AS
  - Se il motore è spento (stato 0)
    - si accende solo se A=1 e S=0
    - resta spento se S=1 (con A=0 o A=1) e anche se S=0 e A=0
  - Viceversa, se il motore è acceso (stato 1)
    - si spegne se S=1, quando A=0 e anche quando A=1 (perché S prevale su A)
    - resta acceso se S=0 (sia quando A=0 che quando A=1)



state	Accendi	Spegni	next
0	1	0	1
0	0	1	0
0	0	0	0
0	1	1	0
1	1	0	1
1	0	1	0
1	0	0	1
1	1	1	0

$$\text{next}(\text{state}, A, S) = \text{not}(\text{state}) * A * \text{not}(S) + \text{state} * A * \text{not}(S) + \text{state} * \text{not}(A) * \text{not}(S)$$

## Esercizio 15

Disegnare la FSM che descrive il comportamento di una **cassaforte a combinazione** che si apre solo digitando la sequenza  $2 \rightarrow 6 \rightarrow 5$

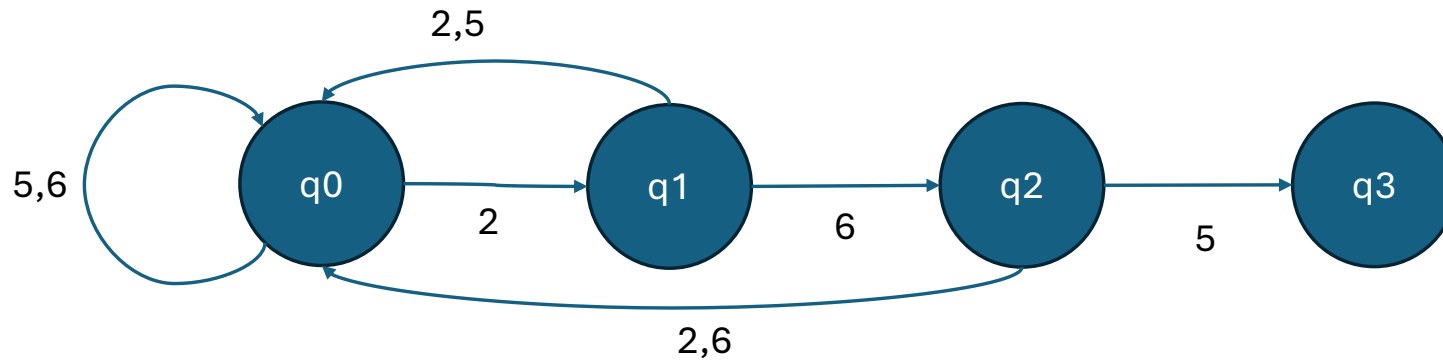
## Esercizio 15 – soluzione (1/2)

Disegnare la FSM che descrive il comportamento di una **cassaforte a combinazione** che si apre solo digitando la sequenza  $2 \rightarrow 6 \rightarrow 5$

- 10 possibili input (ma consideriamo solo 2, 5 e 6)
- 4 Stati:
  - q0 (stato iniziale)
  - q1 (digitato il 2)
  - q2 (digitato il 6 dopo il 2)
  - q3 (stato finale - combinazione corretta: digitati in sequenza 2, 6, 5)
  - se viene digitato un tasto diverso da 2,5,6 assumiamo che lo stato successivo sia q0



## Esercizio 15 – soluzione (2/2)



stato	0	1	2	3	4	5	6	7	8	9
q0	q0	q0	<b>q1</b>	q0	q0	q0	q0	q0	q0	q0
q1	q0	q0	q0	q0	q0	q0	<b>q2</b>	q0	q0	q0
q2	q0	q0	q0	q0	q0	<b>q3</b>	q0	q0	q0	q0
q3	q0	q0	q0	q0	q0	q0	q0	q0	q0	q0

Nella tabella sono indicati i next\_state per ciascuno dei possibili input

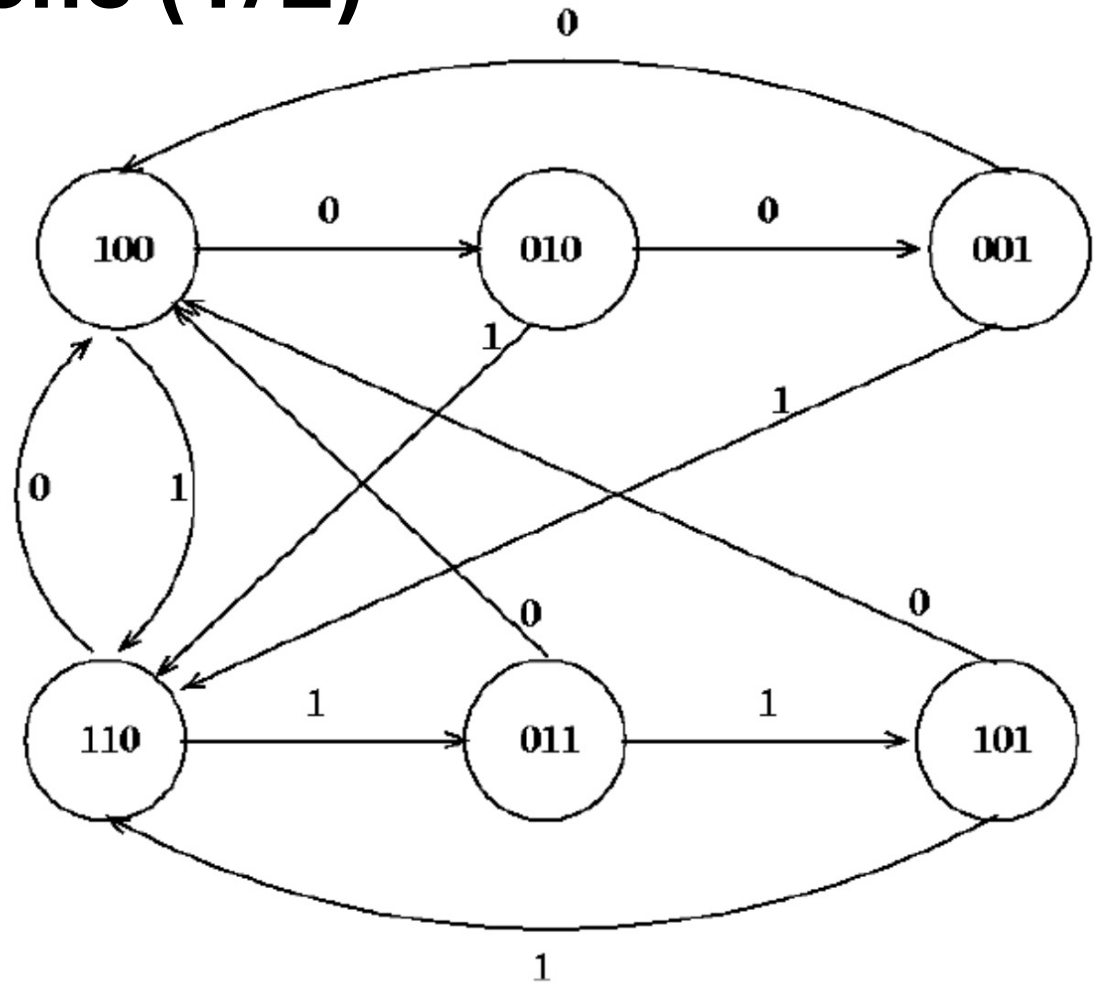
# Esercizio 16

- Disegnare FSM, funzione nextState() e funzione di output() relativa ad una rete sequenziale che comanda l'**accensione e lo spegnimento di tre lampadine** (LampS, LampC, LampD) in sequenza e il cui ritmo è determinato dal periodo di un clock
- L'output del circuito può essere rappresentato con tre bit. Quando tali bit sono uguali a 1, le lampadine corrispondenti sono accese
- La rete riceve un **segnale di ingresso I** e ha il seguente comportamento:
  - **se I = 0**  $\Rightarrow$  le lampadine si accendono in sequenza **una alla volta** partendo da S (100  $\rightarrow$  010  $\rightarrow$  001  $\rightarrow$  100  $\rightarrow$  ...)
  - **se I = 1**  $\Rightarrow$  le lampadine si accendono in sequenza **due alla volta** partendo da S e C (110  $\rightarrow$  011  $\rightarrow$  101  $\rightarrow$  110  $\rightarrow$  ...)

## Esercizio 16 – soluzione (1/2)

Macchina a stati finiti

- 6 stati ammissibili, corrispondenti alle possibili configurazioni SCD specificati dalle due sequenze di output
- per ogni stato si hanno 2 possibili transizioni in corrispondenza dei due possibili input



# Esercizio 16 – soluzione (2/2)

Tabella di verità per *NextState*

Scale document down

Non sono stati ammissibili

S0	S1	S2	I	S0*	S1*	S2*
0	0	0	0	X	X	X
0	0	1	0	1	0	0
0	1	0	0	0	0	1
0	1	1	0	1	0	0
1	0	0	0	0	1	0
1	0	1	0	1	0	0
1	1	0	0	1	0	0
1	1	1	0	X	X	X
0	0	0	1	X	X	X
0	0	1	1	1	1	0
0	1	0	1	1	1	0
0	1	1	1	1	0	1
1	0	0	1	1	1	0
1	0	1	1	1	1	0
1	1	0	1	0	1	1
1	1	1	1	X	X	X

• Tabella di verità per *Output*

S0	S1	S2	S	C	D
0	0	0	X	X	X
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	X	X	X