

Esercitazione del 30/03/2006 - Soluzioni

1. Bistabile asincrono SC (detto anche SR)

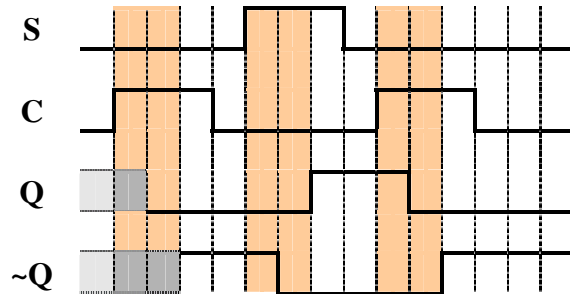
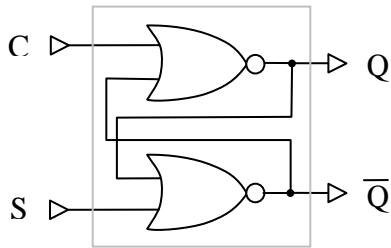


Tabella delle transizioni o *stato prossimo*:

| S | C | Q | Q* |
|---|---|---|----|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | X |
| 1 | 1 | 1 | X |

Configurazioni vietate: il circuito per queste configurazioni ha un comportamento indeterminato. E' possibile quindi impostare arbitrariamente il valore di queste configurazioni allo scopo di semplificare le funzioni se necessario.

Funzione stato prossimo:

$$Q^* = \sim S \sim C Q + S \sim C \sim Q + S \sim C Q$$

(sviluppo SOP, X=0)

Status quo

Rivoluzione

$$= \sim S \sim C Q + S \sim C (\sim Q + Q) = \sim S \sim C Q + S \sim C$$

(sviluppo con mappe di Karnaugh, X=1)

| Q* | | SC | | | |
|----|---|----|----|----|----|
| | | 00 | 01 | 11 | 10 |
| Q | 0 | 0 | 0 | X | 1 |
| | 1 | 1 | 0 | X | 1 |

$$Q^* = S + \sim C Q$$

2. Latch sincrono SC (positive level-triggered)

Un latch è un bistabile sincrono sensibile ai livelli del segnale di controllo

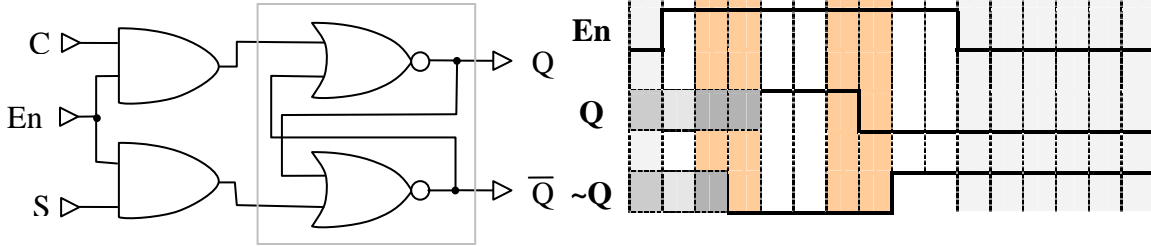
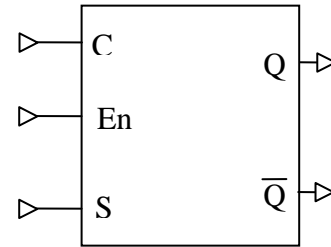


Tabella delle transizioni o stato prossimo:

| S | C | En=E | Q | Q* |
|---|---|------|---|----|
| X | X | 0 | 0 | 0 |
| X | X | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | X |
| 1 | 1 | 1 | 1 | X |

Configurazioni *ininfluenti*: il comportamento del circuito per queste configurazioni non dipende dalle variabili X. E' un modo per semplificare la scrittura della tabella di verità



Funzione stato prossimo:

(sviluppo SOP, X=0)

$$\begin{aligned}
 Q^* &= \sim EQ + \sim S \sim CEQ + S \sim CE \sim Q + S \sim CEQ \\
 &= \sim EQ + \sim S \sim CEQ + S \sim CE (\sim Q + Q) \\
 &= \sim EQ + \sim S \sim CEQ + S \sim CE = \sim EQ + E(\sim S \sim CQ + S \sim C)
 \end{aligned}$$

Status quo

Rivoluzione

(sviluppo con mappe di Karnaugh, X=1)

| Q* | | SC | | | |
|----|----|----|----|----|----|
| | | 00 | 01 | 11 | 10 |
| EQ | 00 | 0 | 0 | 0 | 0 |
| | 01 | 1 | 1 | 1 | 1 |
| | 11 | 1 | 0 | X | 1 |
| | 10 | 0 | 0 | X | 1 |

$$Q^* = \sim EQ + ES + E \sim C Q$$

3. Latch sincrono tipo D (positive level-triggered)

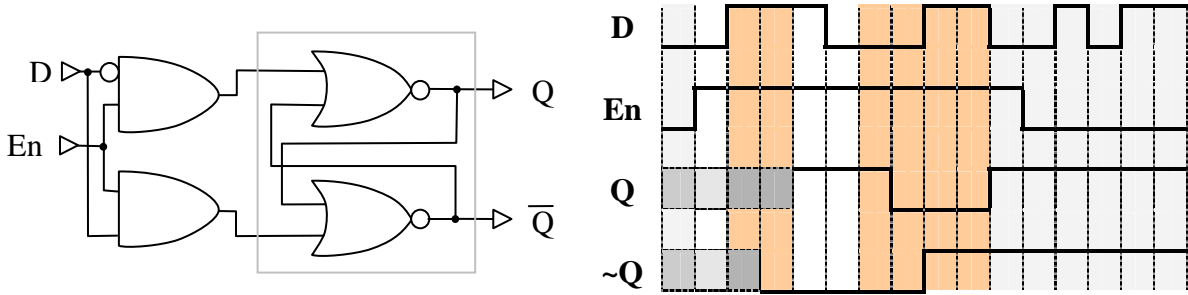
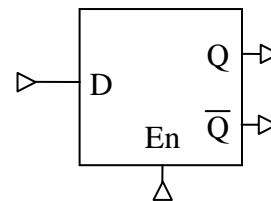


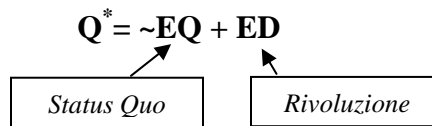
Tabella delle transizioni o *stato prossimo*:

| D | En=E | Q | Q* |
|---|------|---|----|
| X | 0 | 0 | 0 |
| X | 0 | 1 | 1 |
| 0 | 1 | X | 0 |
| 1 | 1 | X | 1 |

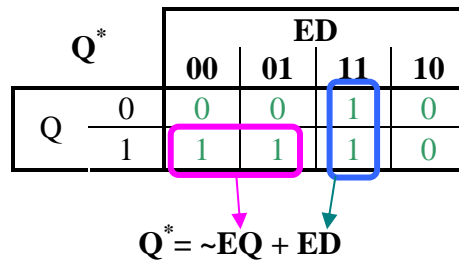


Funzione stato prossimo:

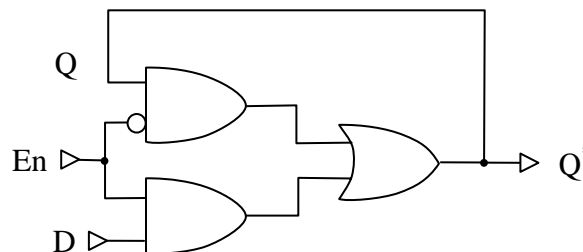
(sviluppo SOP)



(sviluppo con mappe di Karnaugh)



Circuito logico equivalente:



4. Flip-Flop sincrono D (positive edge-triggered). Hold e Set Time

Comportamento identico al latch sincrono level-triggered eccetto che l'attivazione avviene sui fronti di salita o di discesa del clock

Un flip-flop è un bistabile sincrono sensibile ai fronti del segnale di controllo

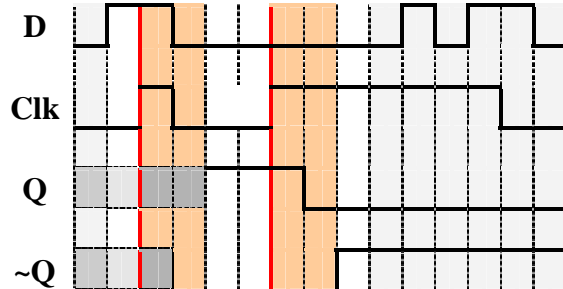
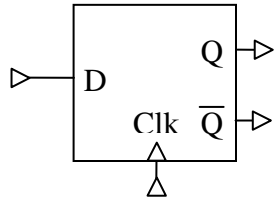


Tabella delle transizioni o stato prossimo (formato compatto):

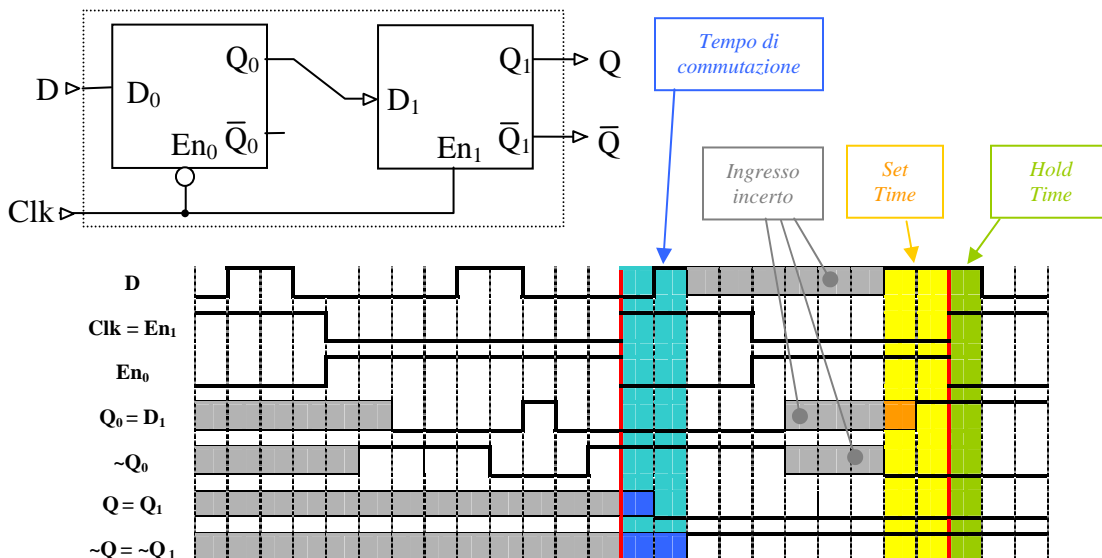
| D | Clk = E | Q | Q* |
|---|---------|---|----|
| X | - | q | q |
| 0 | ↑ | X | 0 |
| 1 | ↑ | X | 1 |

Funzione stato prossimo:

Caso edge-triggered: considero il fronte positivo come fosse un valore uguale ad 1, i restanti casi come uguali a 0.

$$Q^* = \sim EQ + ED \quad (\text{sviluppo SOP})$$

Una possibile realizzazione circuitale e grafico dei tempi:



5. Flip-flop sincrono J-K (positive edge-triggered)

In questo latch, la combinazione vietata del latch SC, entrambi gli ingressi ad 1, viene usata per invertire lo stato dell'uscita (chiaramente il circuito deve essere edge-triggered altrimenti l'uscita dovrebbe continuare a cambiare di stato finché persiste il segnale di controllo).

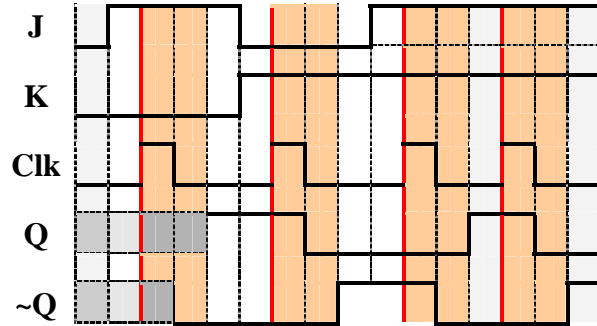
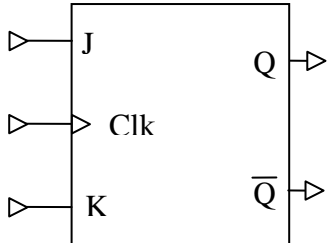
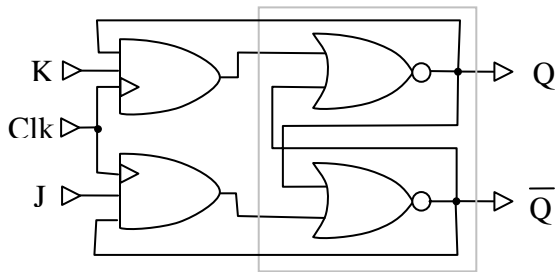


Tabella delle transizioni o stato prossimo:

| J | K | Clk=E | Q | Q* |
|---|---|-------|---|----|
| X | X | - | q | q |
| 0 | 0 | ↑ | q | q |
| 0 | 1 | ↑ | X | 0 |
| 1 | 0 | ↑ | X | 1 |
| 1 | 1 | ↑ | q | ~q |



Funzione stato prossimo:

Caso edge-triggered: considero il fronte positivo come fosse un valore uguale ad 1, i restanti casi come uguali a 0.

(sviluppo con SOP)

$$\begin{aligned}
 Q^* &= \sim EQ + \sim J \sim KEQ + J \sim KE + JKE \sim Q \\
 &= \sim EQ + E(\sim J \sim KQ + \mathbf{J \sim K} + JK \sim Q) \\
 &= \sim EQ + E(\mathbf{\sim J \sim KQ} + \mathbf{J \sim KQ} + \mathbf{J \sim K \sim Q} + \mathbf{JK \sim Q}) = \sim EQ + E(\sim KQ + \mathbf{J \sim Q})
 \end{aligned}$$

(sviluppo con mappe di Karnaugh)

| Q* | | JK | | | |
|----|----|----|----|----|----|
| | | 00 | 01 | 11 | 10 |
| EQ | 00 | 0 | 0 | 0 | 0 |
| | 01 | 1 | 1 | 1 | 1 |
| | 11 | 1 | 0 | 0 | 1 |
| | 10 | 0 | 0 | 1 | 1 |

$$Q^* = \sim EQ + \sim KQ + EJ \sim Q$$

6. Flip-flop sincrono T (positive edge-triggered)

Questo flip-flop costituisce una versione semplificata (con meno linee di controllo) del flip-flop JK.

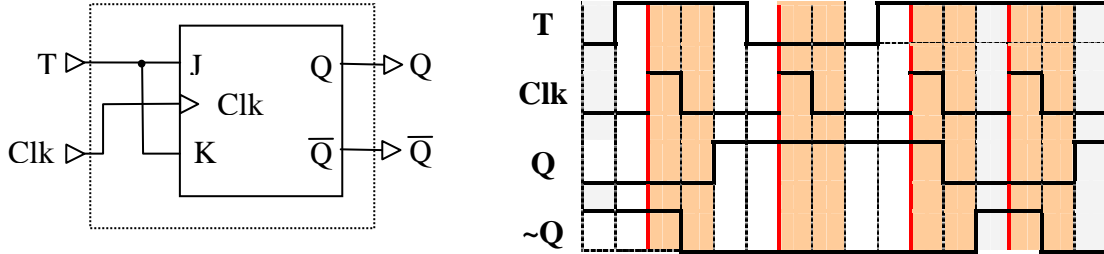


Tabella delle transizioni o *stato prossimo*:

| T | Clk=E | Q | Q* |
|---|-------|---|-----------|
| X | - | q | q |
| 0 | ↑ | q | q |
| 1 | ↑ | q | ~q |

Funzione stato prossimo:

Caso edge-triggered: considero il fronte positivo come fosse un valore uguale ad 1, i restanti casi come uguali a 0.

(sviluppo con SOP)

$$Q^* = \sim EQ + E\sim TQ + ET\sim Q$$

$$= \sim EQ + E(\sim TQ + T\sim Q) = \sim EQ + E(T \oplus Q)$$

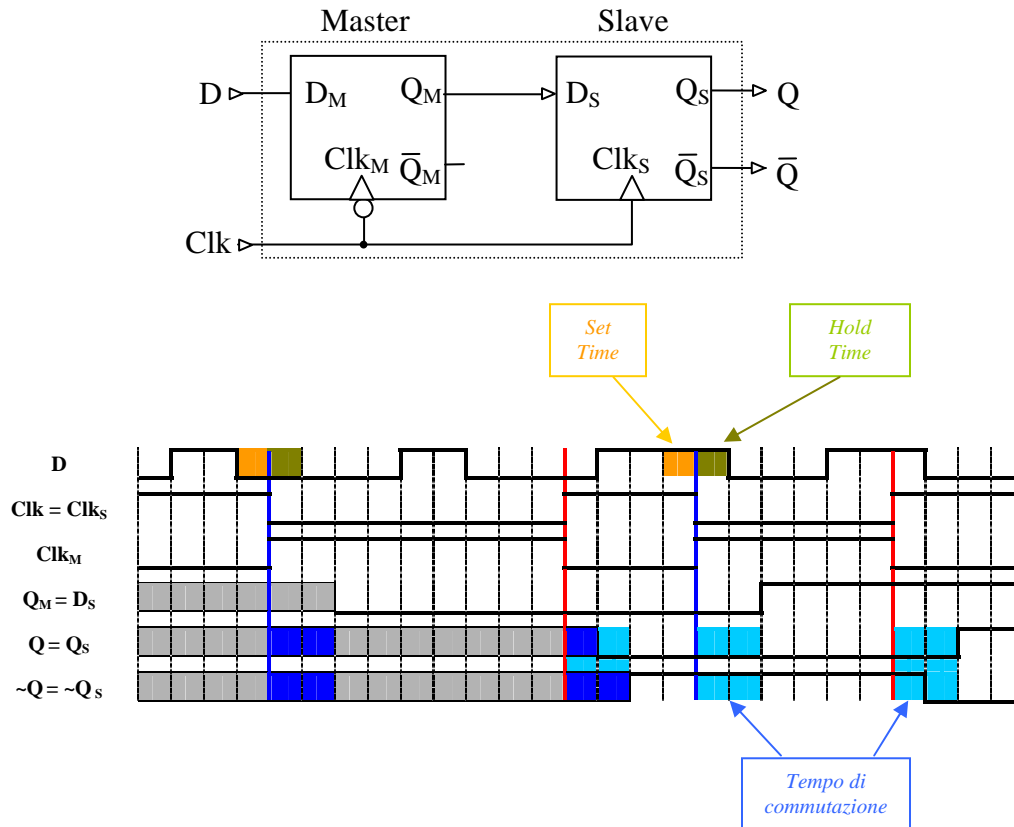
(sviluppo con mappe di Karnaugh)

| Q* | E | TQ | | | |
|----|---|----|----|----|----|
| | | 00 | 01 | 11 | 10 |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |

$$Q^* = \sim EQ + \sim TQ + ET\sim Q$$

7. Flip-Flop sincrono D Master-Slave (positive edge-triggered).

Comportamento identico al flip-flop sincrono di tipo D eccetto che la fase di memorizzazione e la fase di trasferimento in uscita vengono separate in due momenti distinti: il F/F Master-Slave (positive edge-triggered) memorizza il dato sul fronte di discesa mentre lo trasferisce sulle uscite solo sul successivo fronte di salita.



8. Reti sequenziali e contatori

Vedi Slides .