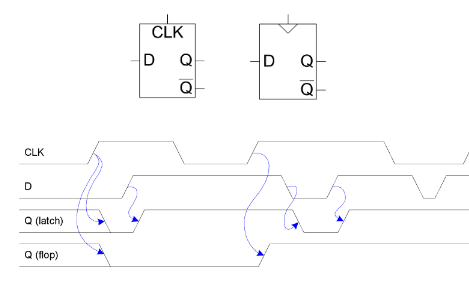
**Flip Flop**

Input : D, CLK

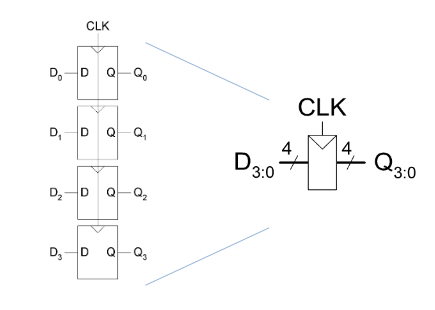
Output : Q

Ci sono oggetti simili ai latch ma che hanno un funzionamento leggermente diverso.

Quando CLK passa dal segnale 0 al segnale 1 (*fronte di salita*) Q assume il valore di D e lo ricorda finché non si presenterà un nuovo fronte di salita.



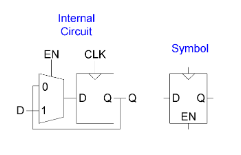
**Multi Flip Flop**



**Enabled Flip Flop**

Inputs : CLK, D, EN

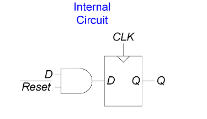
L’input EN controlla quando viene dato un nuovo valore a D Funzione:  
EN = 0 il flip-flop torna allo stato precedente  
EN = 1 D passa attraverso Q



**Resettable Flip Flop**

Inputs : D, CLK, Reset

Funzione   
Reset = 1 Q è forzato a 0  
Reset = 0 Si comporta come un normale flip flop

2 tipi:  
Sincrono : Si resetta solo sul fronte di salita  
Asincrono : si resetta quando Reset = 1  
Immagine che contiene testo, orologio, calibro

Descrizione generata automaticamente

**Settable Flip Flop**

Inputs : D, CLK, Set

Funzione   
Reset = 1 Q è forzato a 1  
Reset = 0 Si comporta come un normale flip flop

Immagine che contiene testo, orologio, antenna, calibro

Descrizione generata automaticamente

**Circuiti sincroni sequenziali**

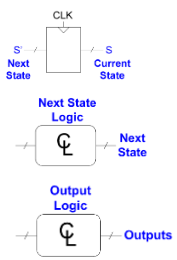
* Si rompe il ciclo inserendo registri
* I registri contengono lo stato del sistema
* Lo stato cambia sul fronte di salita

Regole per la composizione

* Ogni elemento o è un registro o è un circuito combinatorio
* Ci deve essere almeno un registro
* Tutti i registri ricevono lo stesso segnale di CLK
* Tutti i percorsi ciclici contengono almeno 1 registro

**Macchina a stati finiti**

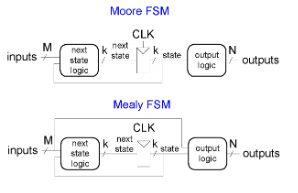
La macchina a stati finiti mantiene lo stato corrente, e carica il prossimo stato sul fronte di salita.



Il prossimo stato è determinato dallo stato corrente e dagli input, ci sono due tipi di macchine a stati e differiscono nella logica degli output :

Moore FSM : output dipendono solo dallo stato corrente

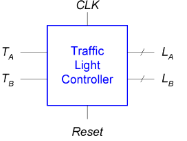
Melay FSM : gli output dipendono dallo stato corrente e dall’input



**FSM Block blox**

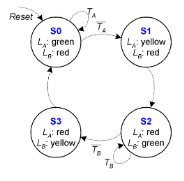
Inuts : CLK,Reset,

Output:



**FSM Diagramma di transizione**

Stati : cerchi  
Transizioni : archi



|  |  |  |  |
| --- | --- | --- | --- |
| CURRENT STATE | INPUTS | INPUTS | NEXT STATE |
| S | T L | T B | S’ |
| S0 | 0 | X | S1 |
| S0 | 1 | X | S0 |
| S1 | X | X | S2 |
| S2 | X | 0 | S3 |
| S2 | X | 1 | S2 |
| S3 | X | X | S0 |

**FSM Tavola codificata di transizione e stati**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CURRENT STATE | // | INPUTS | // | NEXT STATE | // |
|  |  |  |  |  |  |
| 0 | 0 | 0 | X | 0 | 1 |
| 0 | 0 | 1 | X | 0 | 0 |
| 0 | 1 | X | X | 1 | 0 |
| 1 | 0 | X | 0 | 1 | 1 |
| 1 | 0 | X | 1 | 1 | 0 |
| 1 | 1 | X | X | 0 | 0 |

|  |  |
| --- | --- |
| State | Encoding |
| S0 | 00 |
| S1 | 01 |
| S2 | 10 |
| S3 | 11 |

Questa codifica ci dice informazioni sugli stati precedenti, correnti e sulle variabili.

**FSM Tavola di output**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Current | State | Outputs | Outputs | Outputs | Outputs |
|  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |

|  |  |
| --- | --- |
| output | encoding |
| green | 00 |
| yellow | 01 |
| red | 10 |

**FMS REGISTRO STATI**

