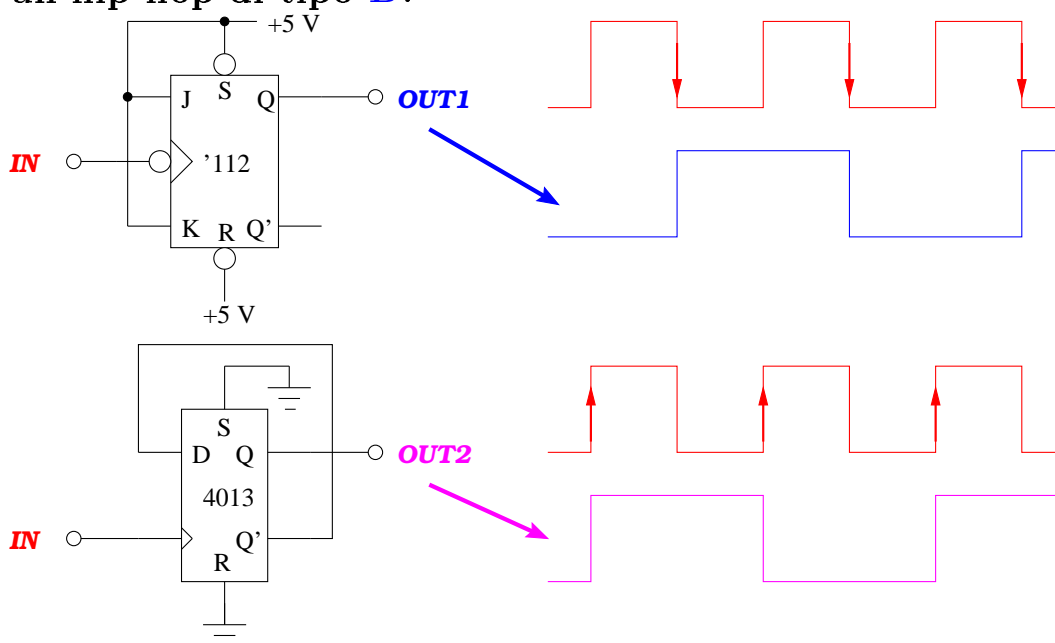
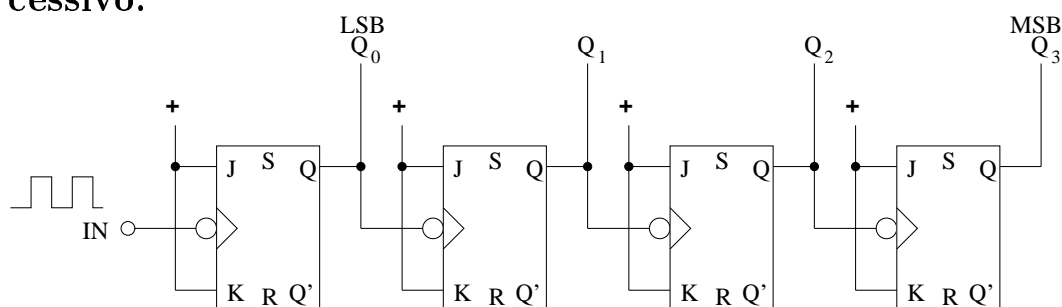


Prime applicazioni dei flip-flop

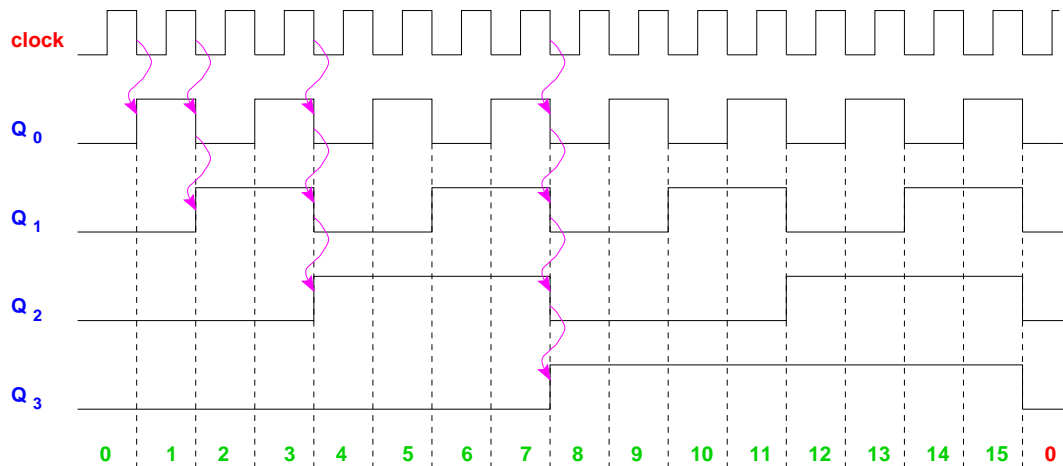
- Una semplice applicazione dei flip-flop è il circuito che **divide per 2** un treno di impulsi in ingresso: si può realizzare sia con un flip-flop di tipo **J-K** (sfruttando il fatto che con $J = K = 1$ si ha un'inversione ad ogni ciclo di clock, $Q_{n+1} = \overline{Q_n}$), sia con un flip-flop di tipo **D**.



- Il secondo circuito funziona correttamente anche se l'ingresso D cambia stato *quasi* immediatamente dopo il fronte del *clock*: ad es. il flip-flop 74HC74 richiede che l'ingresso sia stabile per $t_s = 20$ ns prima del fronte di salita del clock, e per ulteriori $t_h = 3$ ns dopo, e queste condizioni possono essere soddisfatte dato che il tempo di propagazione interno (dal clock alle uscite) è di 10 ns.
- Un altro circuito interessante è il **contatore binario** ovvero un circuito che divide per 2^n , realizzabile collegando in serie n flip-flop nella configurazione "alternante" vista sopra, con l'uscita di un flip-flop collegata all'ingresso di clock del successivo.



- Questo circuito produce in uscita (Q_3) una forma d'onda digitale di frequenza $\frac{1}{16}$ della frequenza in ingresso. Considerando globalmente le uscite Q_3 , Q_2 , Q_1 e Q_0 si vede che funge anche da contatore da 0 a 15.



- Il contatore binario appena visto è del tipo **asincrono** ovvero *ripple*, in cui ogni flip-flop fa transire il successivo, e può presentare problemi di temporizzazione: quando avviene la transizione tra due stati successivi, ad esempio tra 0111_2 e 1000_2 ($7_{10} \rightarrow 8_{10}$), si presentano momentaneamente sulle uscite alcuni stati “proibiti”:

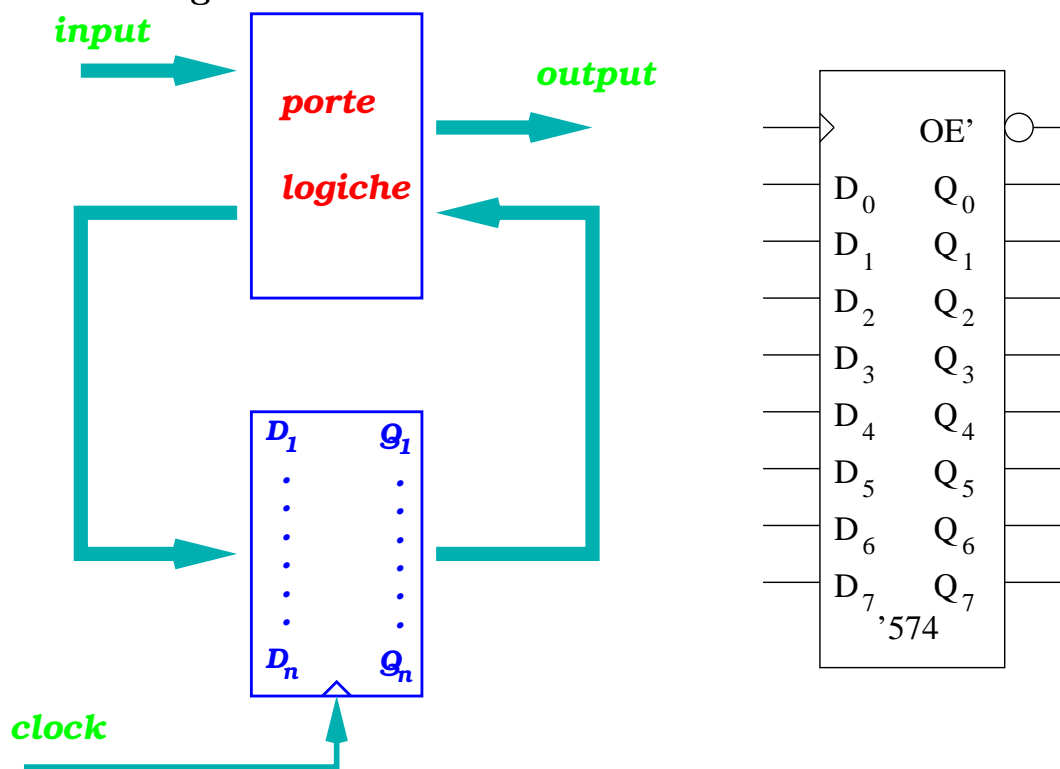
0111 7 (*legale*)
 $\rightarrow 0110$ 6 *illegale*
 $\rightarrow 0100$ 4 *illegale*
 $\rightarrow 0000$ 0 *illegale*
 $\rightarrow 1000$ 8 (*legale*)

per evitare errori, lo stadio successivo al contatore asincrono deve essere abilitato solo dopo che la transizione tra due valori “legali” del conteggio è stata completata.

- Un contatore **sincrono**, in cui tutti gli ingressi di *clock* sono collegati ad un unico segnale esterno, è generalmente più facile da usare.

Circuiti sequenziali

- La forma più generale di circuito digitale comprende un certo numero di flip-flop e una rete combinatoria; un **circuito sequenziale** generico può essere rappresentato dallo schema a blocchi seguente:



in cui è stato introdotto il **registro**, ovvero un insieme di flip-flop di tipo **D** con un unico ingresso di clock; è importante notare che il sistema è sincrono, cioè che i cambiamenti delle uscite avvengono solo in corrispondenza del fronte del clock (gli ingressi devono essere resi stabili con un certo anticipo rispetto al fronte).

- La rete combinatoria può essere realizzata nei diversi modi che abbiamo già esaminato, mentre un esempio di registro a 8 ingressi è il 74HCT574 indicato in figura. Lo **stato interno** del sistema è immagazzinato in un dato istante come contenuto degli n flip-flop.
- Uno strumento molto potente e flessibile per realizzare un circuito sequenziale è rappresentato dalla "PAL sequenziale", simile alla PAL (Programmable Array Logic) combinatoria, con l'aggiunta di registri di uscita.

Contatori

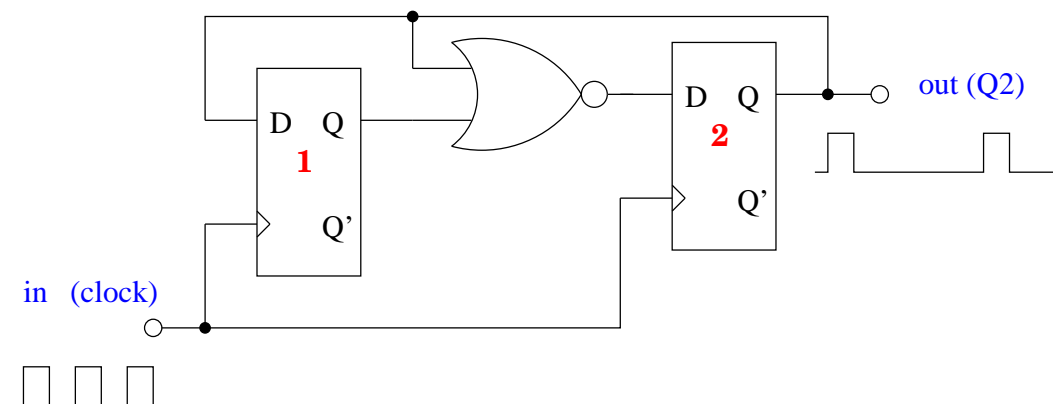
- I circuiti sequenziali più semplici sono quelli senza ingressi (a parte il clock, ovviamente), cioè i **contatori**, di cui abbiamo già visto un esempio. Possono essere descritti da un insieme di stati e da una funzione $NS = f(PS)$, che per ogni valore dello **stato presente** ($PS = \text{Present State}$) dà il valore del **prossimo stato** ($NS = \text{Next State}$).
- Vediamo come esempio un contatore sincrono che divide per 3; come per tutti i contatori conviene rappresentare ciascun stato mediante il numero binario Q_1Q_2 corrispondente al conteggio, per cui la tabella di transizione $PS \rightarrow NS$ è:

PS	Q_1	Q_2	NS	D_1	D_2
00	0	0	01	0	1
01	0	1	10	1	0
10	1	0	00	0	0

- La parte sequenziale consiste in due flip-flop (le cui uscite indichiamo con Q_1 e Q_2), mentre la parte combinatoria deve generare D_1 e D_2 che rappresentano lo stato successivo:

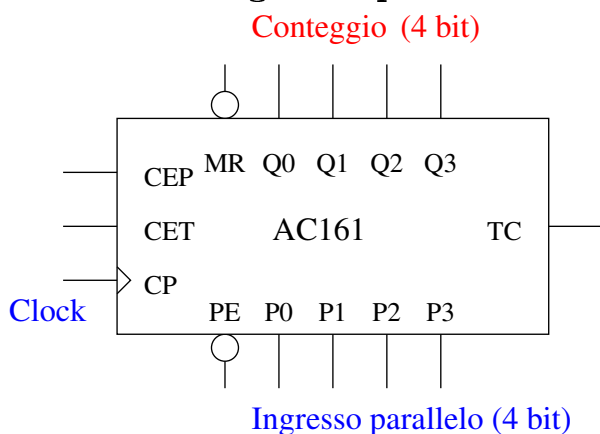
$$D_1 = Q_2$$

$$D_2 = \overline{Q_1 + Q_2}$$



- Infine, verifichiamo che il nostro contatore non resti bloccato nel quarto stato $Q_1 = 1, Q_2 = 1$ (all'accensione lo stato dei flip-flop è casuale): si vede che alla prima transizione $(1, 1) \rightarrow (1, 0)$, cioè il contatore si porta in uno degli stati ammessi.

- I contatori sincroni a n bit sono costituiti da n flip-flop con l'ingresso di clock in comune; se si usano flip-flop di tipo J-K, i due ingressi J e K di ciascuno stadio sono connessi insieme ad una combinazione delle uscite degli stadi precedenti; possono essere realizzati con il riporto in serie (ottenuto dai due stadi precedenti) oppure con il riporto in parallelo (ottenuto da tutti gli stadi precedenti).
- Sono disponibili nelle categorie MSI e LSI parecchi tipi di contatori, sia sincroni sia asincroni, a 4, 5, 7, 8, 12 e 14 bit; ad esempio il '161 è un contatore sincrono a 4 bit che conta sul fronte di salita del clock ed è dotato di azzeramento asincrono e di ingresso parallelo sincrono.



Esempio di contatore binario
sincrono a 4 bit: '161

- clock: su fronte di salita
- ingresso parallelo: sincrono
- azzeramento: asincrono
- direzione: UP

MR = azzeramento

CEP, CET = abilitazioni

PE = abilitazione ingresso parallelo

TC = Total Count (segnala massimo conteggio)

- Nello stesso gruppo troviamo il '163 ad azzeramento sincrono e i contatori BCD '160 (azzer. asincrono) e '162 (azzer. sincrono).
- Esistono contatori di tipo UP (crescenti) e di tipo DOWN (decrescenti: si ottengono collegando \bar{Q} invece di Q al successivo flip-flop), ed anche contatori UP/DOWN in cui la direzione di conteggio è selezionabile (ad es. i '190-'193).
- Le applicazioni comprendono, oltre al semplice conteggio:
 1. generazione di impulsi di durata multipla del periodo del clock
 2. generazione di treni di n impulsi
 3. misura della frequenza di un segnale periodico (utilizzando un oscillatore a cristallo)
 4. misura di tempo tra due impulsi
 5. controllo dell'esecuzione di istruzioni in sequenza in un elaboratore elettronico

Una macchina a stati

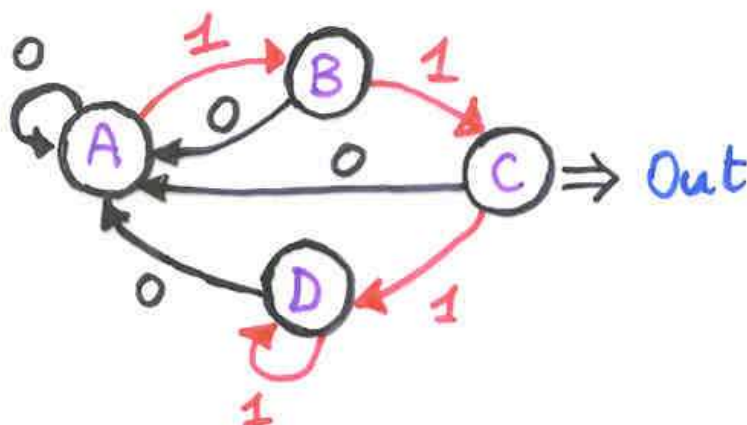
- I circuiti sequenziali con pochi ingressi (oltre al clock) possono essere facilmente realizzati mediante flip-flop di tipo D e logica combinatoria. Vediamo un esempio, un circuito con 1 bit di ingresso che rivela la presenza della successione binaria 011, concepito come “macchina a stati”. Innanzitutto dobbiamo definire l’insieme degli stati interessanti:

Stato	Definizione
A	l’ultimo ingresso era 0
B	gli ultimi due ingressi erano 0,1
C	gli ultimi tre ingressi erano 0,1,1
D	gli ultimi tre ingressi erano 1,1,1

- In seguito dobbiamo definire la funzione $NS = f(PS, In)$, che per ogni valore dello **stato presente** ($PS = \text{Present State}$) e dell’**ingresso** (In) ci dà il valore del **prossimo stato** ($NS = \text{Next State}$):

PS	NS		Out
	In=0	In=1	
A	A	B	0
B	A	C	1
C	A	D	0
D	A	D	0

- È utile rappresentare gli stati e le transizioni in forma grafica:



- Dobbiamo poi definire la codifica binaria degli stati mediante alcuni flip-flop (nel nostro caso ne bastano due): scegliamo $(Q_0, Q_1) = (0, 0), (0, 1), (1, 1), (1, 0)$ rispettivamente per A, B, C, D.
- Infine, per la parte combinatoria, possiamo usare il metodo della mappa di Karnaugh *separatamente* per Q_0 e Q_1 .