

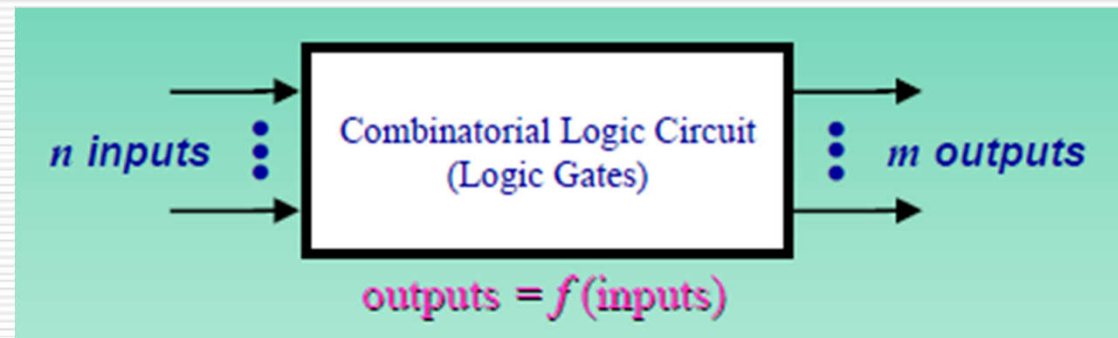
Logică digitală

-Curs 7-
Circuite logice
secvențiale

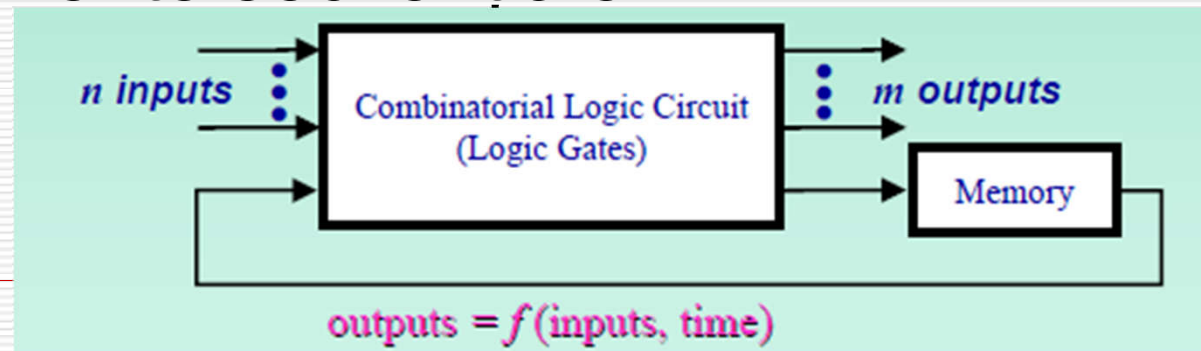
Clasificare componente digitale

□ Componente combinaționale

- Ușor de analizat, partiționat, verificat



□ Componente secvențiale



Circuite secvențiale

- ❑ Circuitele secvențiale sunt circuitele care prezintă feedback (legătură inversă), adică o legătură de la ieșiri spre intrări;
 - ❑ Denumirea de secvențiale provine de la faptul că acestea țin cont (rețin) secvența de intrări;
 - ❑ Ele sunt elemente care prezintă **memorie**;
 - ❑ **Ieșirea** unui circuit secvențial este determinată atât pe baza **intrării**, cât și pe baza **stării curente** (memorate);
-

Circuite secvențiale

- ❑ Exemplu: un robot care înregistrează mesaje după ce telefonul sună de 6 ori
 - ❑ Pentru a număra mesajele, modulul are nevoie de 3 biți
 - ❑ Când este memorat numărul spunem că modulul de numărare se află în starea 0
 - ❑ Pentru situația în care sunt memorate numerele: 1,2,3,4,5 spunem că modulul se află în starea 1,2,3,4, respectiv 5
 - ❑ Starea 5 denotă faptul că au sosit 5 apeluri, deci se poate înregistra mesajul, iar modulul de numărare trece în starea 0
-

Circuite secvențiale

- ❑ Exemplu: un robot care înregistrează mesaje după ce telefonul sună de 6 ori
- ❑ Pentru a număra mesajele, modulul are nevoie de 4 biți

Ieșirea numărătorului de apeluri, depinde atât de **intrare** (se generează un apel nou), cât și de **starea curentă** (care reține câte apeluri au fost până la momentul curent)!

- ❑ 1,2,3,4,5 spunem că modulul se află în starea 1,2,3,4, respectiv 5
 - ❑ Starea 5 denotă faptul că au sosit 5 apeluri, deci se poate înregistra mesajul, iar modulul de numărare trece în starea 0
-

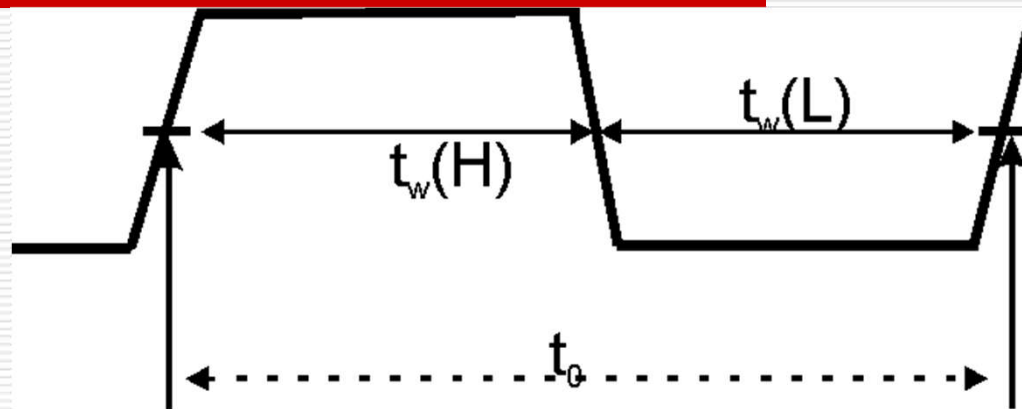
Circuite secvențiale

- **Circuitele secvențiale se clasifică:**
 - **Asincrone**
 - **Sincrone**

 - Componentele secvențiale asincrone își modifică starea și valorile de ieșire funcție de modificările semnalelor de la intrare (**oricând!**) se modifică acestea.

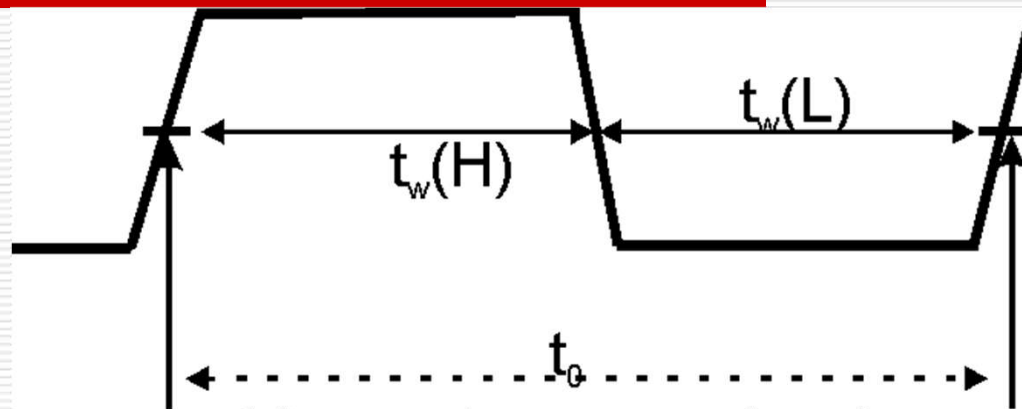
 - Componentele secvențiale sincrone își modifică valoarea funcție de valoarea semnalelor de intrare la **momente bine definite de timp**, dictate de un semnal (de intrare) care se numește tact (*clock*)
-

Semnalul de tact

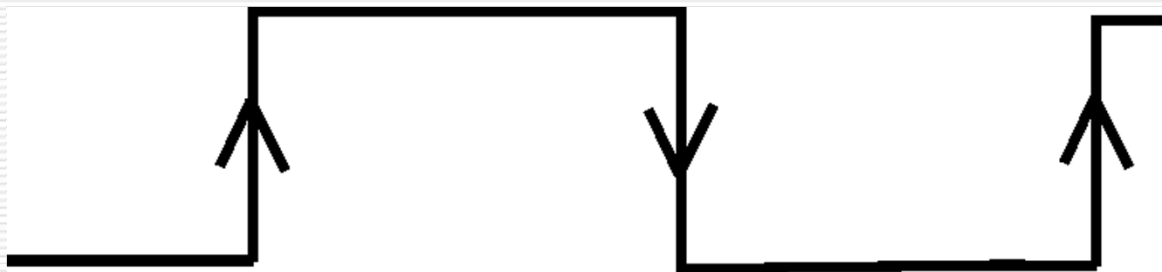


- ❑ **Palierul** unui semnal reprezintă porțiunea unde acesta rămâne constant 0 logic (palier negativ) și 1 logic (palier pozitiv).
- ❑ **Frontul crescător** se referă la porțiunea unde semnalul își modifică valoarea de la 0 logic la 1 logic (mai exact de la 10% din nivelul corespunzător lui 1 logic la 90% din nivelul corespunzător lui 1 logic)
- ❑ t_0 - perioada semnalului de tact,
- ❑ $t_w(H)$ și $t_w(L)$ reprezintă durata unui impuls de 1 respectiv 0 logic

Semnalul de tact

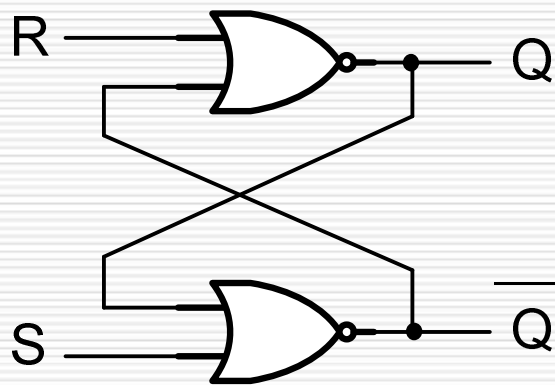


a) Semnal rectangular de tact

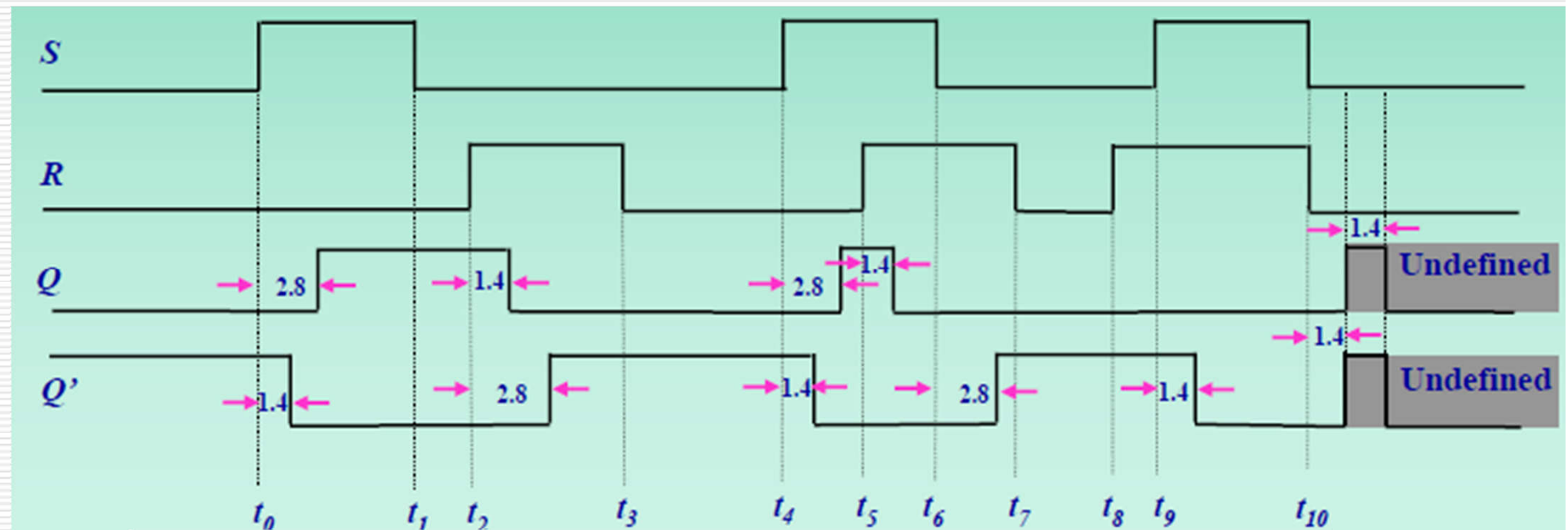


b) Formă idealizată a semnalului

S-R Latch (SAU-NU) - asincron



R	S	Q_{t+1}
0	0	Q_t
0	1	1
1	0	0
1	1	interzis



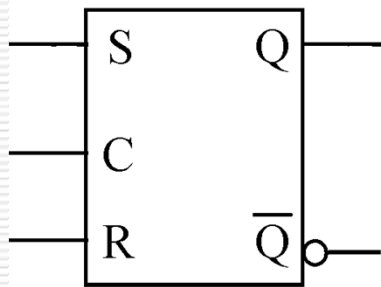
Observații:

- Dacă S, R – sunt active → ieșirile sunt 0
 - Dacă S, R – sunt dezactivate simultan:
 - Pt porți care au exact aceeași întârziere: Latch-ul intră într-o stare de oscilație ($t_{10}+1.4$ ns ieșirile sunt 1, $t_{10}+2.8$ ns ambele ieșiri sunt 0)
 - Pt porți cu întârzieri diferite: Latch-ul intră într-o stare nedefinită (poarta cu întârzierea mai mare își impune valoarea de ieșire la 1 logic)
 - SR-latch-ul pt. S,R dezactivate simulatan are un comportament nedeterministic
-

Observații:

- Regula ca S, R să nu fie dezactivate simultan e greu de impus unui design:
 - Multe căi cu întârzieri diferite – pentru unele situații greu de estimat (activitatea de comutare design time)
 - Se impune regula mai strictă ca S și R să nu fie activate simultan
-

Gated SR-latch

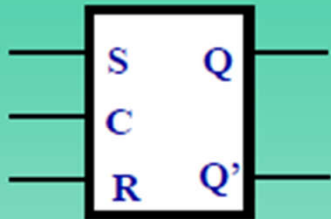


S	R	C	Q _{next}	\overline{Q}_{next}
0	0	1	Q	\overline{Q}
0	1	1	0	1
1	0	1	1	0
1	1	1	-	-
*	*	0	Q	\overline{Q}

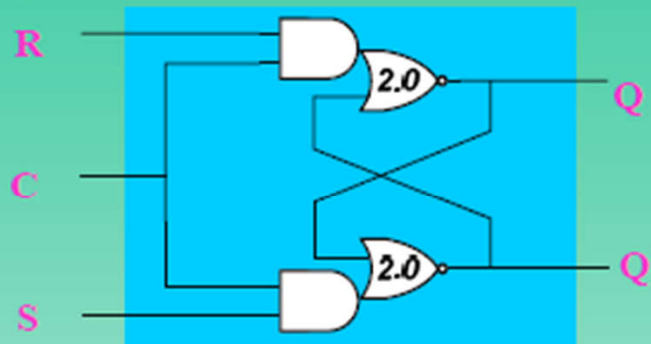
- ❑ Când semnalul C este activ valorile de la intrare sunt propagate prin latch
- ❑ Semnalele de intrare nu trebuie să se modifice în intervalul t_{setup} și t_{hold} al frontului descrescător

Gated SR-Latch

•Control signal C activates the latch



Graphic symbol

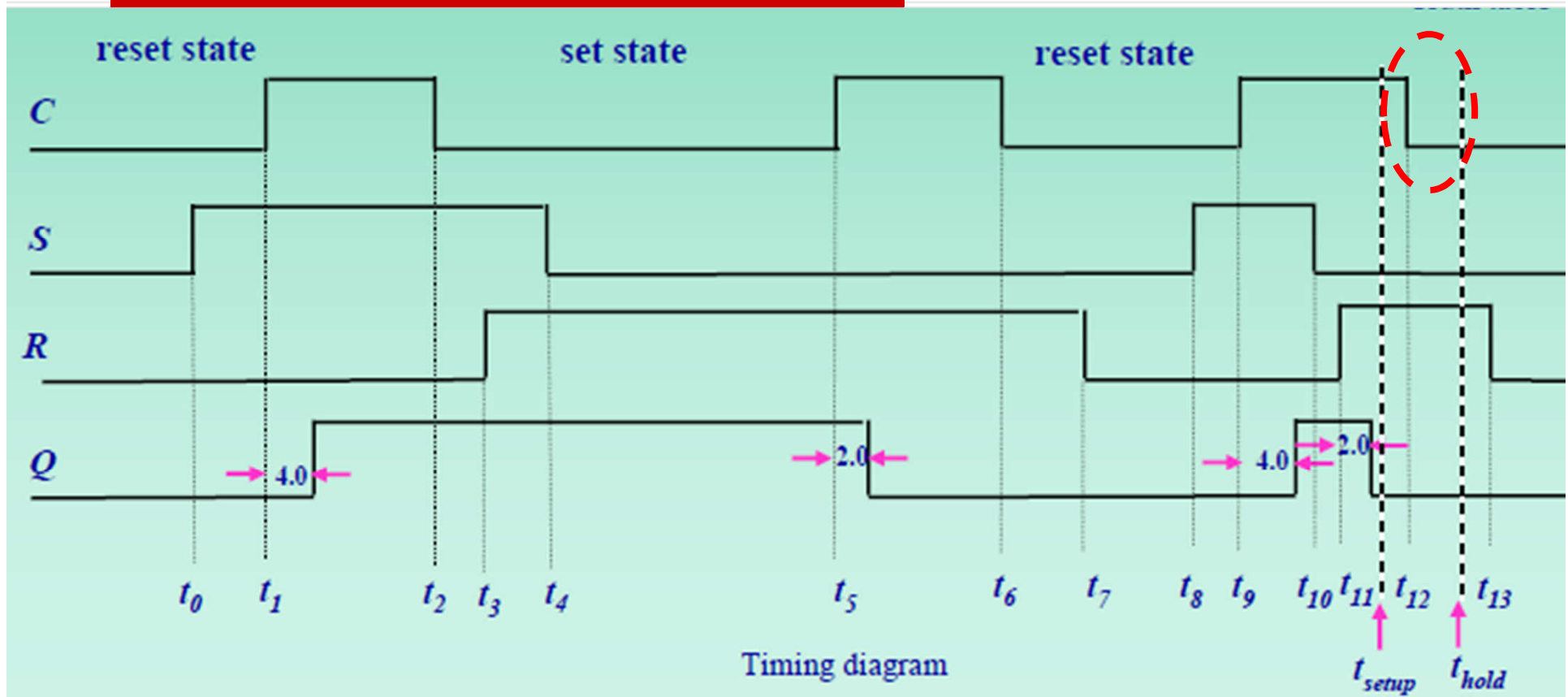


Logic schematic

<i>C</i>	<i>S</i>	<i>R</i>	<i>Q</i>	<i>Q(next)</i>	
0	X	X	0	0	(inactive)
0	X	X	1	1	(inactive)
1	0	0	0	0	(hold)
1	0	0	1	1	(hold)
1	0	1	X	0	(reset)
1	1	0	X	1	(set)
1	1	1	X	NA	(?)

Truth table

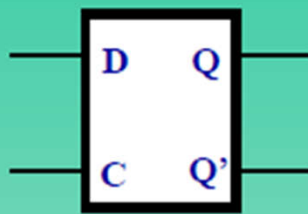
Gated SR-latch (SR-latch sincron)



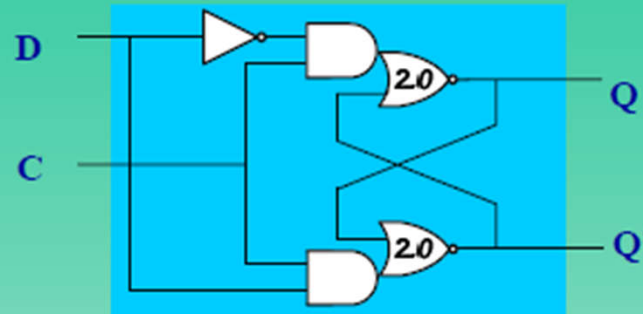
$$t_{12}-t_{11} \geq t_{setup}$$

$$t_{13}-t_{12} \geq t_{hold}$$

Gated D-latch



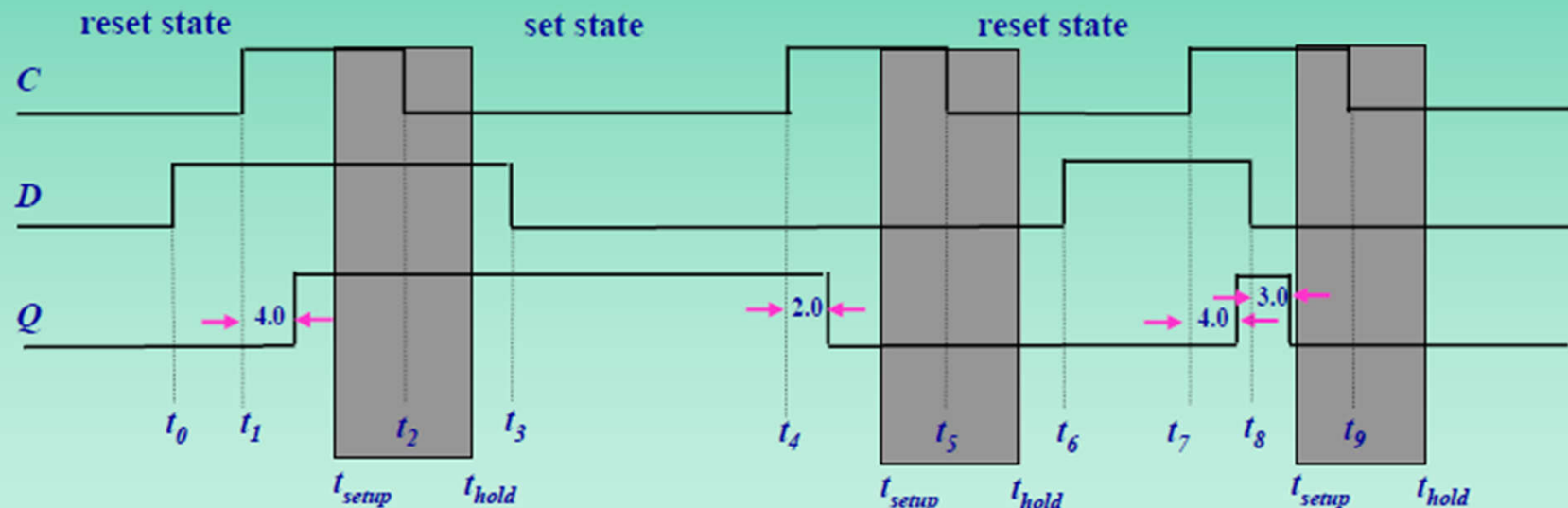
Graphic symbol



Logic schematic

<i>C</i>	<i>D</i>	<i>Q</i>	<i>Q(next)</i>
0	X	0	0
0	X	1	1
1	0	X	0
1	1	X	1

Truth table

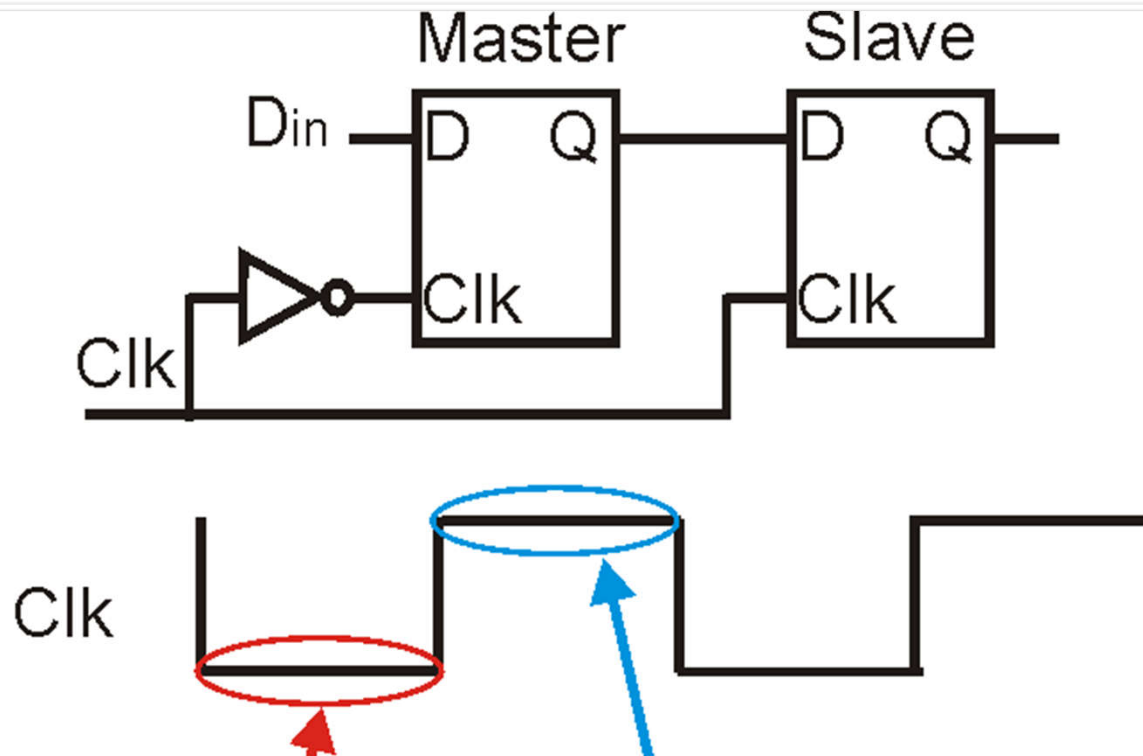


Timing diagram

Flip-flop-uri

- ❑ Se mai numesc și latch-uri sensibile pe frontul semnalului de tact;
 - ❑ Bascularea se face pe frontul semnalului de tact (!nu pe palier – latch-uri)
 - ❑ Două variante de arhitecturi:
 - Configurația master-slave
 - Edge-triggered FF
-

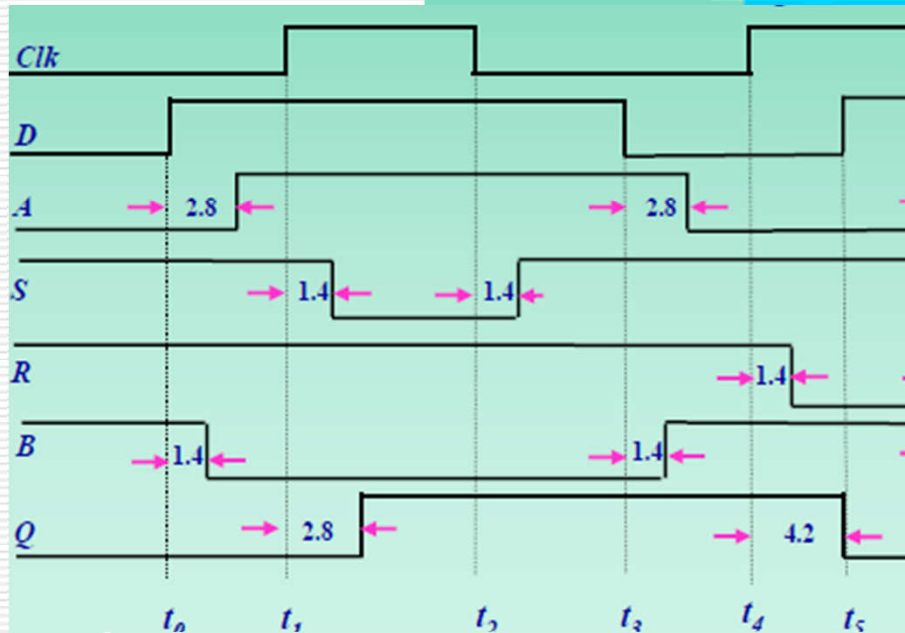
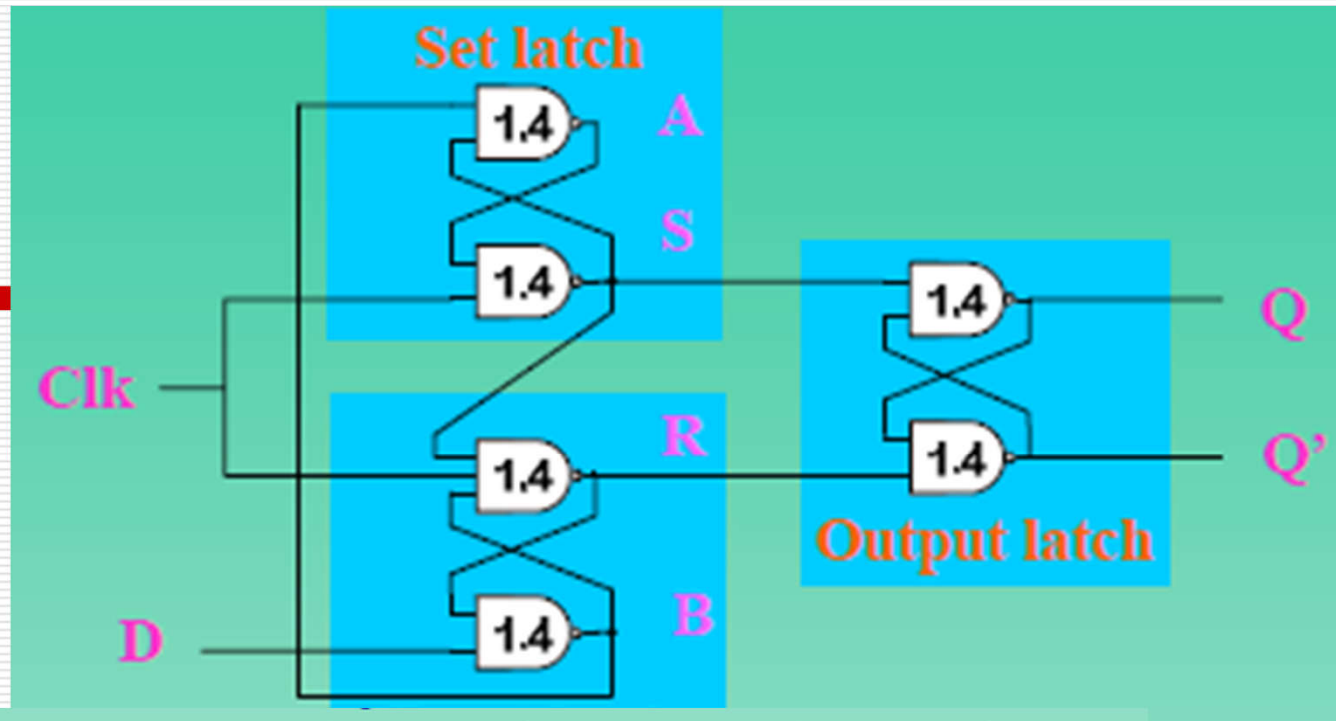
FF-Master-Slave



Incarcare Master
Latch cu D_{in}
Slave Latch
memoreaza
data anteriora

Incarcare Slave
Latch cu Q_{master}
Master Latch
memoreaza
valoarea D_{in}
dinaintea comutarii
semnalului Clk

Edge trigger FF



Clk = 0

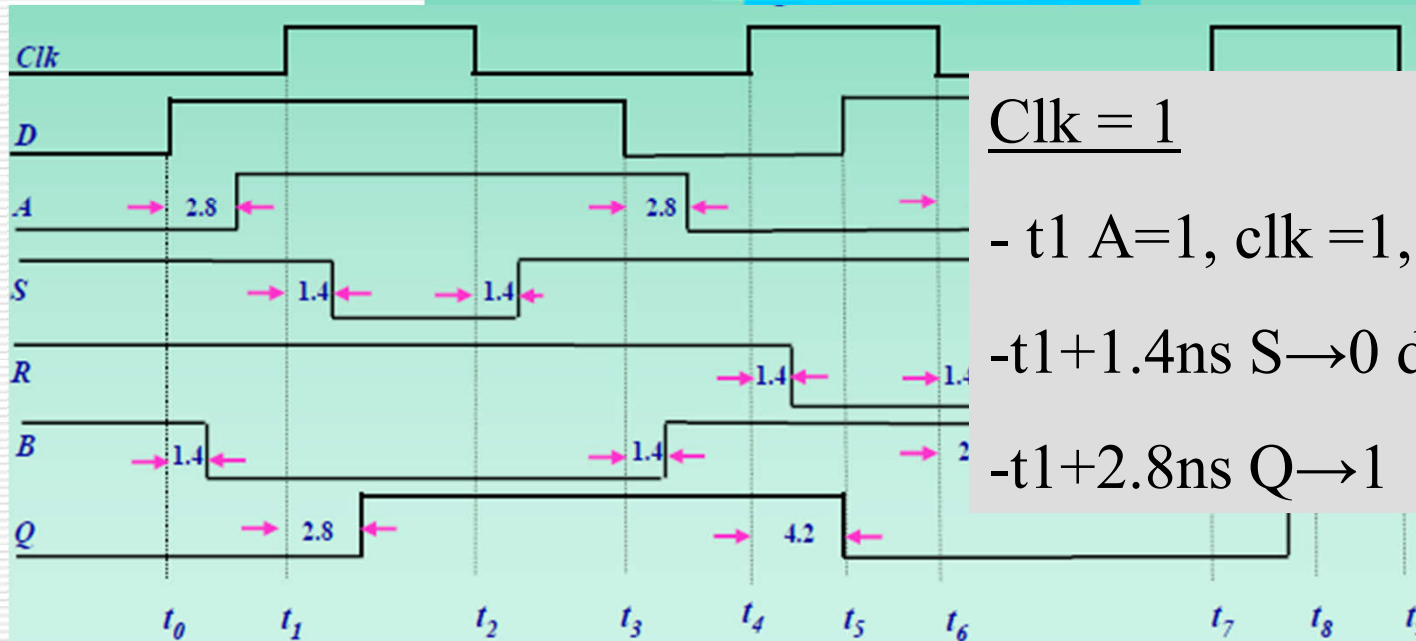
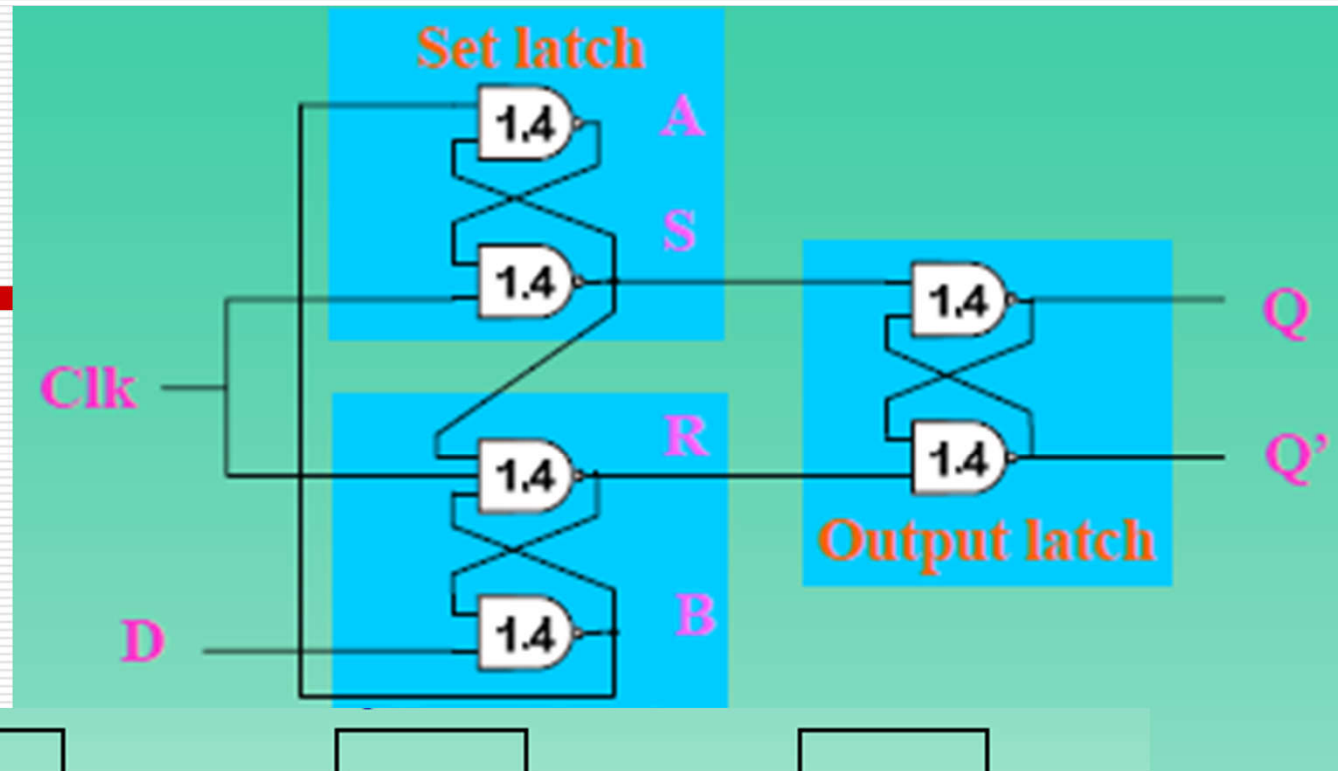
- t_0 $D \rightarrow 1$ determină

- $t_0 + 1.4\text{ns}$ $B \rightarrow 0$ determină

- $t_0 + 2.8\text{ns}$ $A \rightarrow 1$

$R, S = 1 \rightarrow Q$ nemodificat

Edge trigger FF



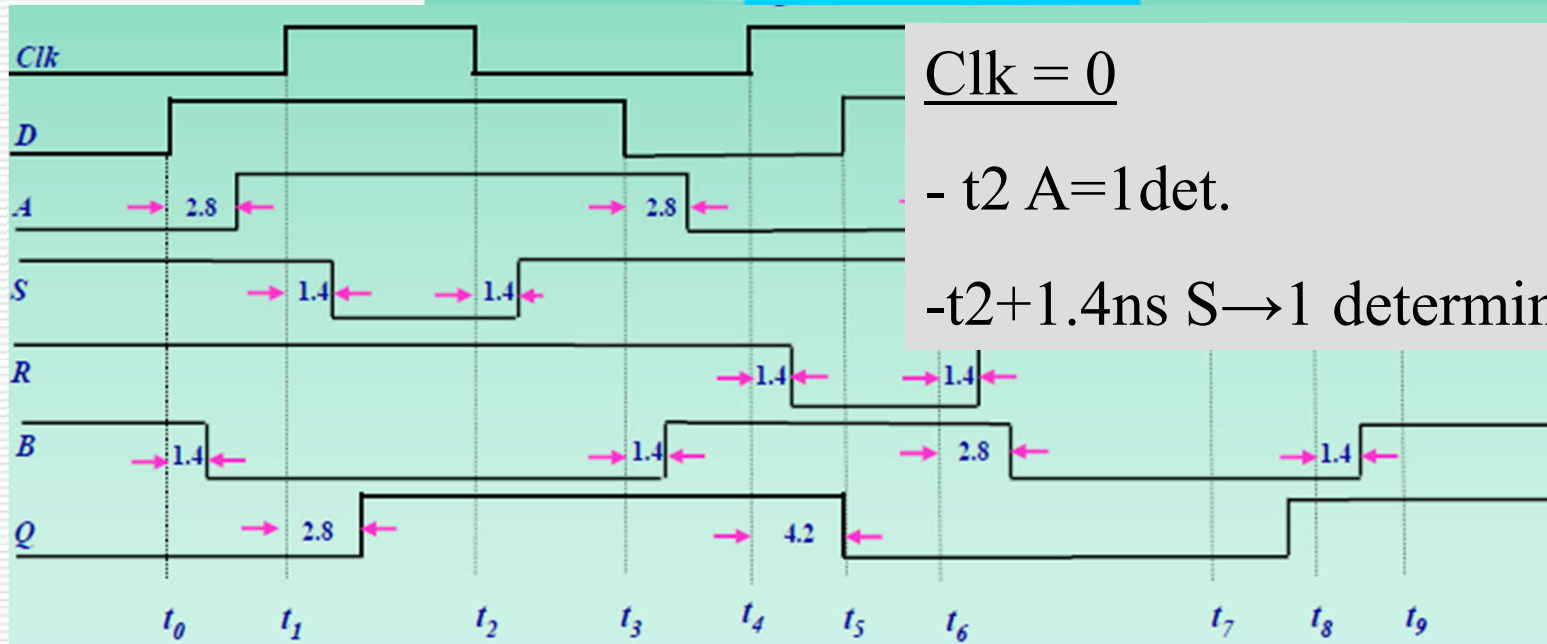
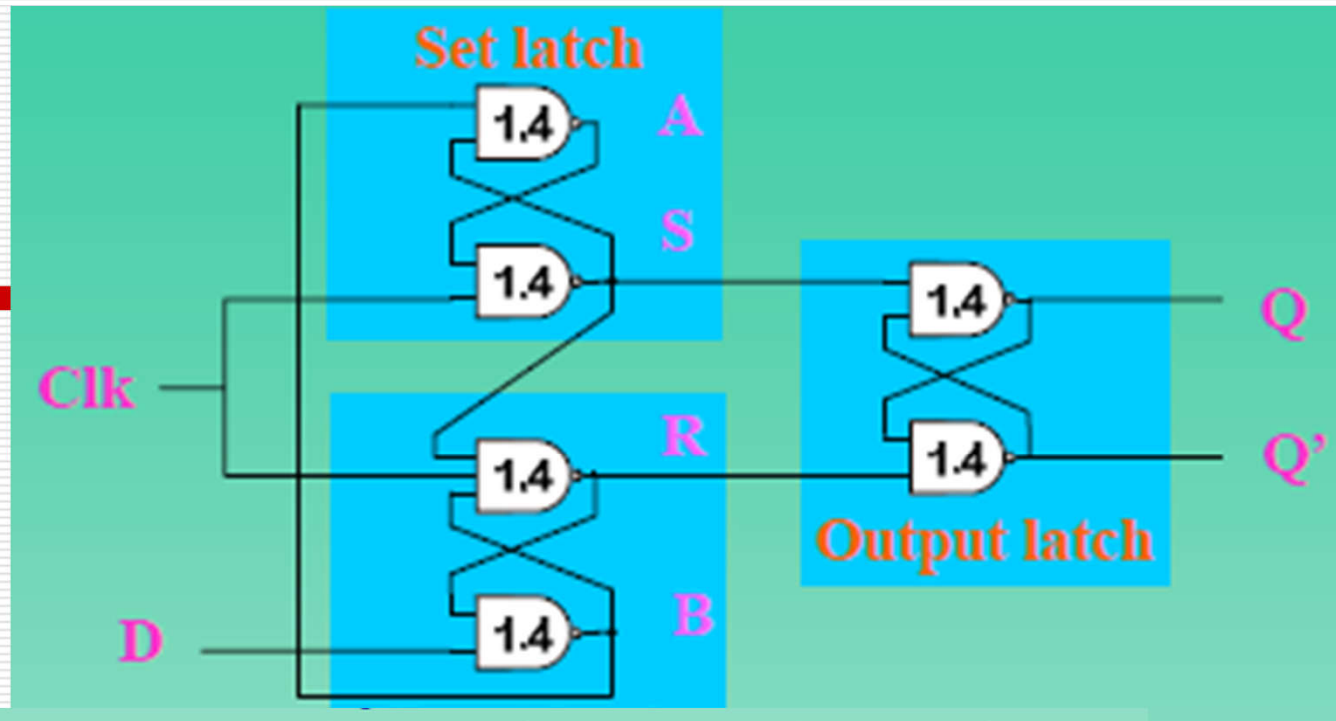
Clk = 1

- t_1 $A=1$, $clk = 1$, det.

- $t_1 + 1.4ns$ $S \rightarrow 0$ determină

- $t_1 + 2.8ns$ $Q \rightarrow 1$

Edge trigger FF

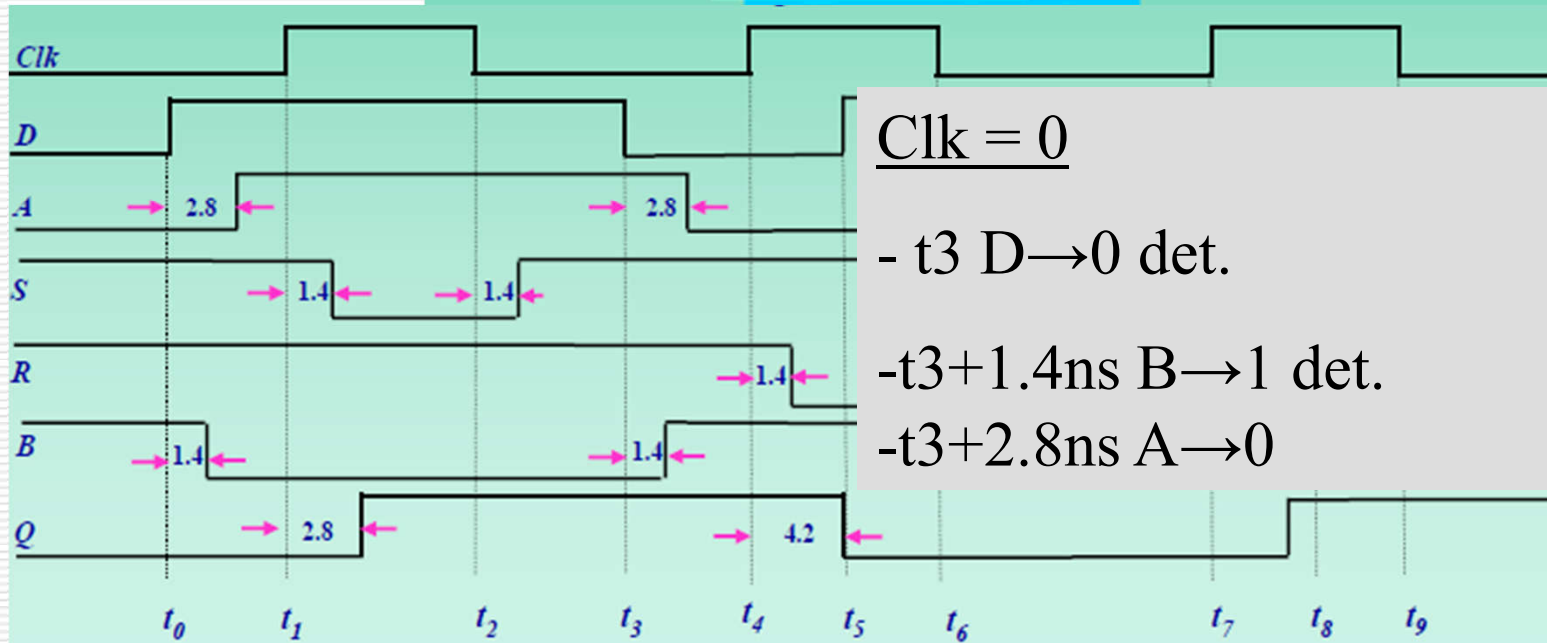
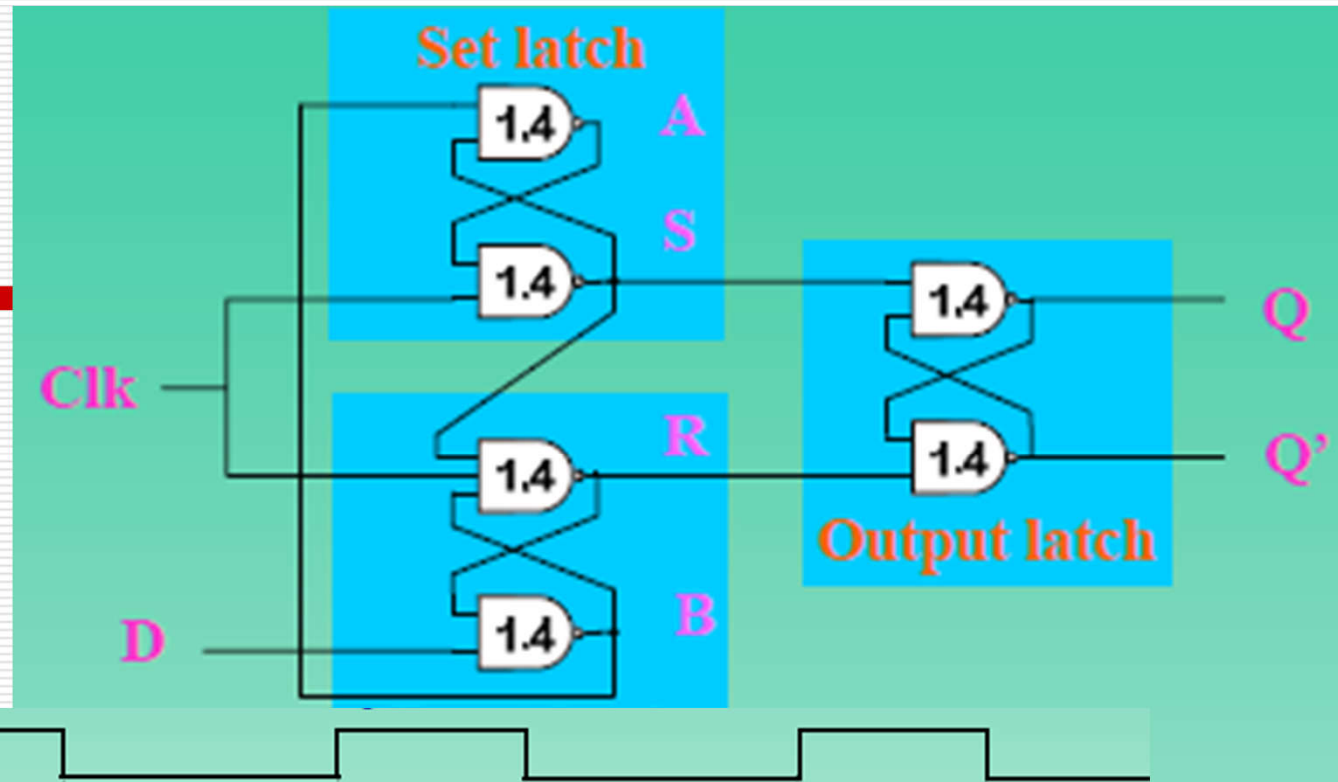


Clk = 0

- t_2 A=1 det.

- $t_2 + 1.4\text{ns}$ S \rightarrow 1 determină

Edge trigger FF



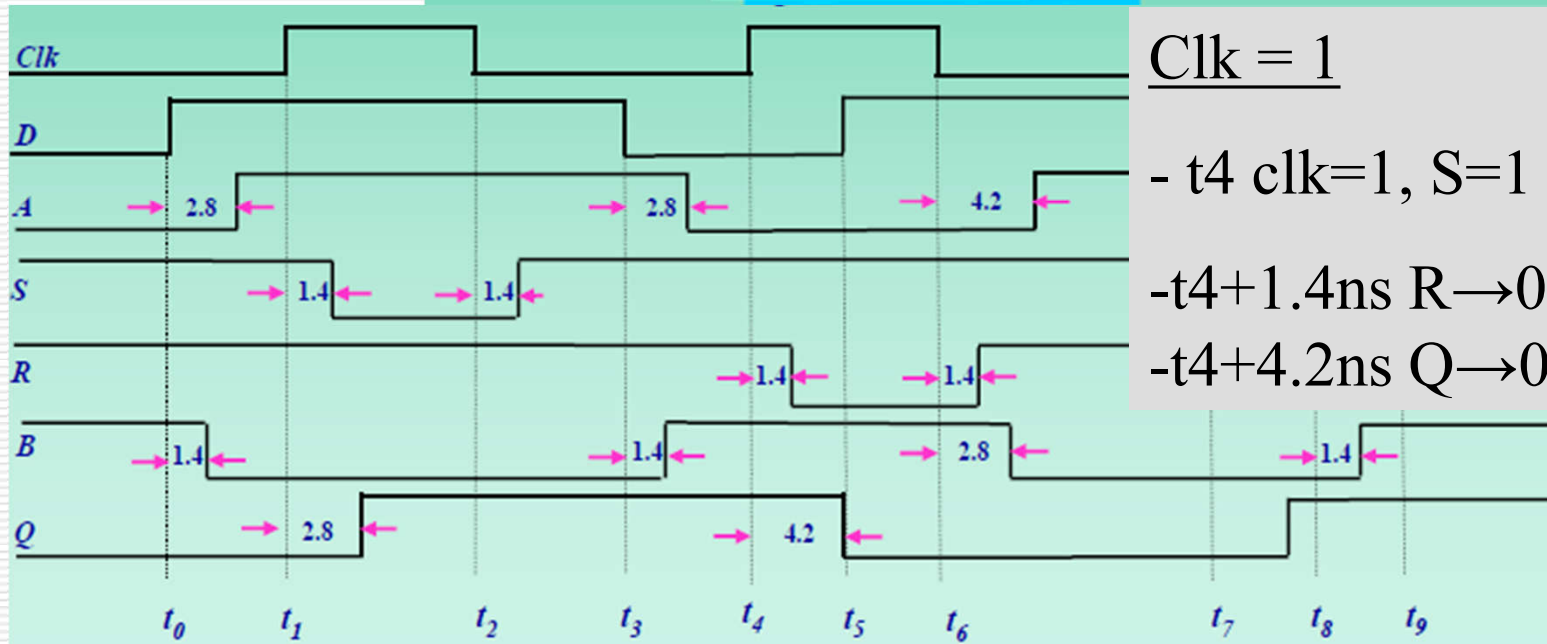
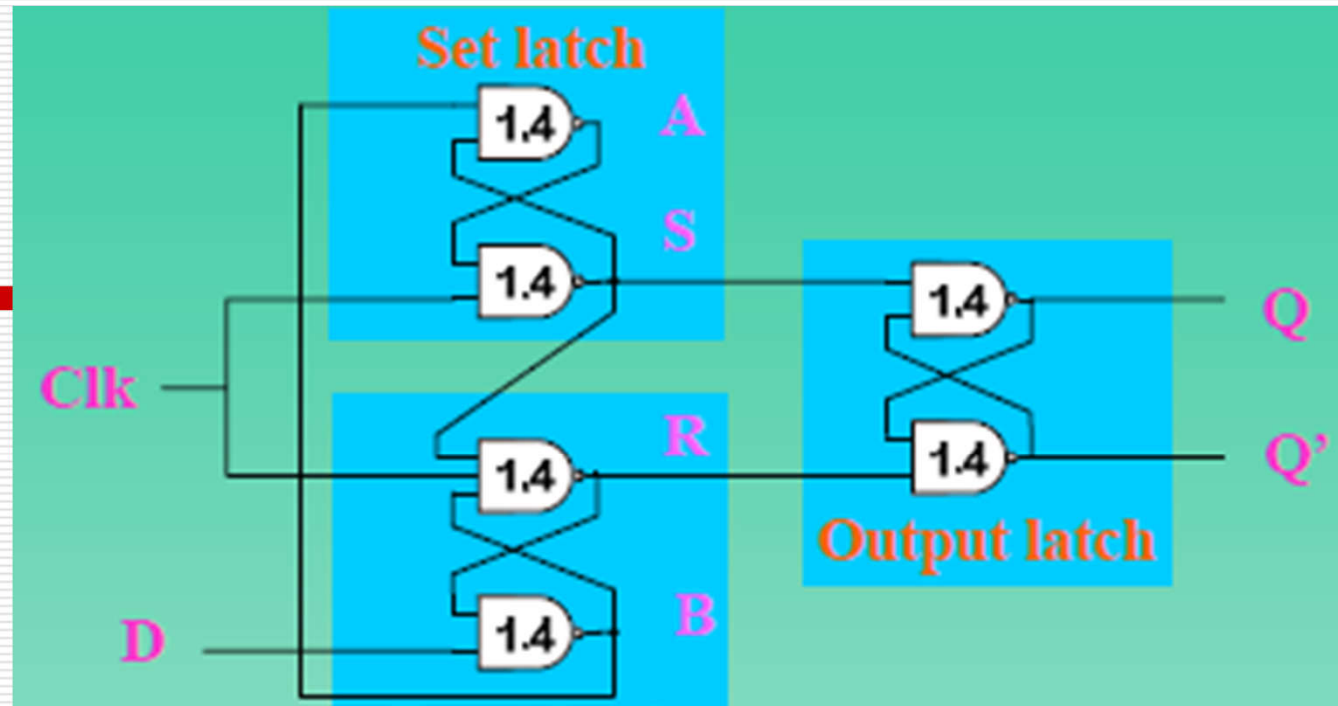
Clk = 0

- t_3 $D \rightarrow 0$ det.

- $t_3 + 1.4\text{ns}$ $B \rightarrow 1$ det.

- $t_3 + 2.8\text{ns}$ $A \rightarrow 0$

Edge trigger FF



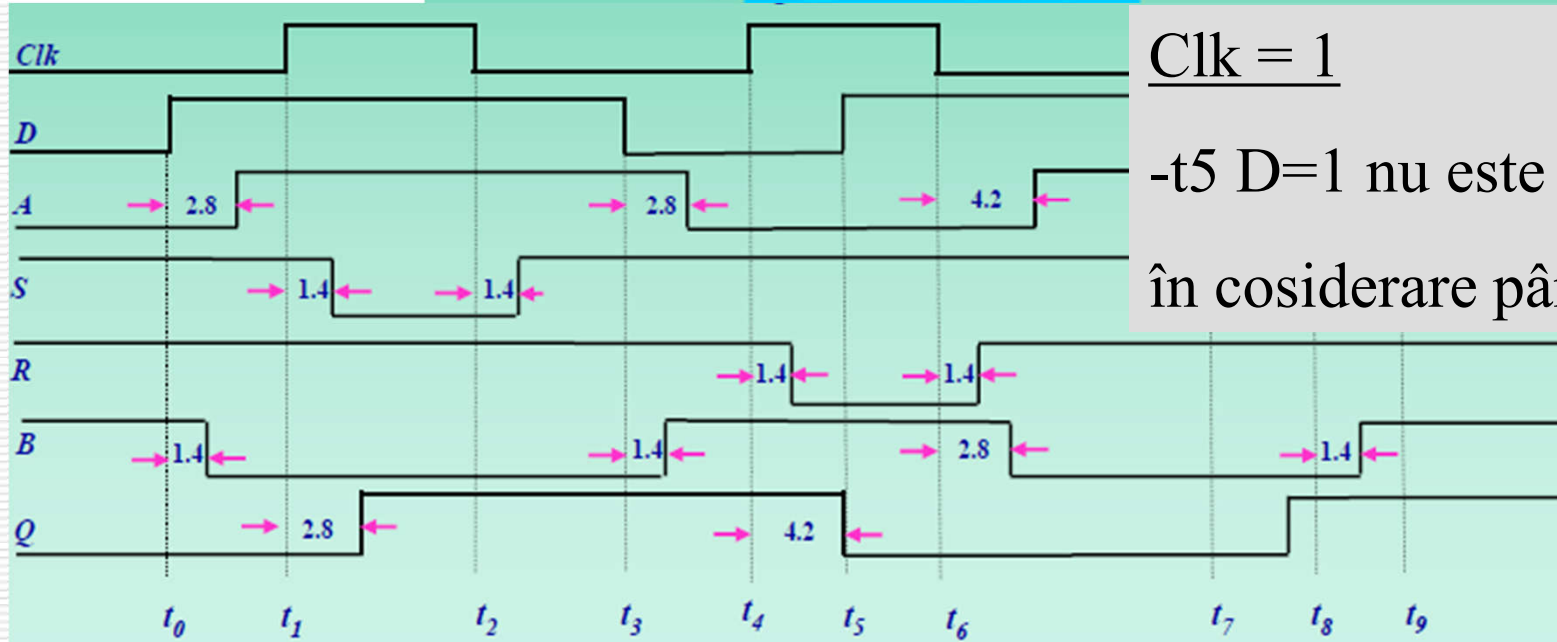
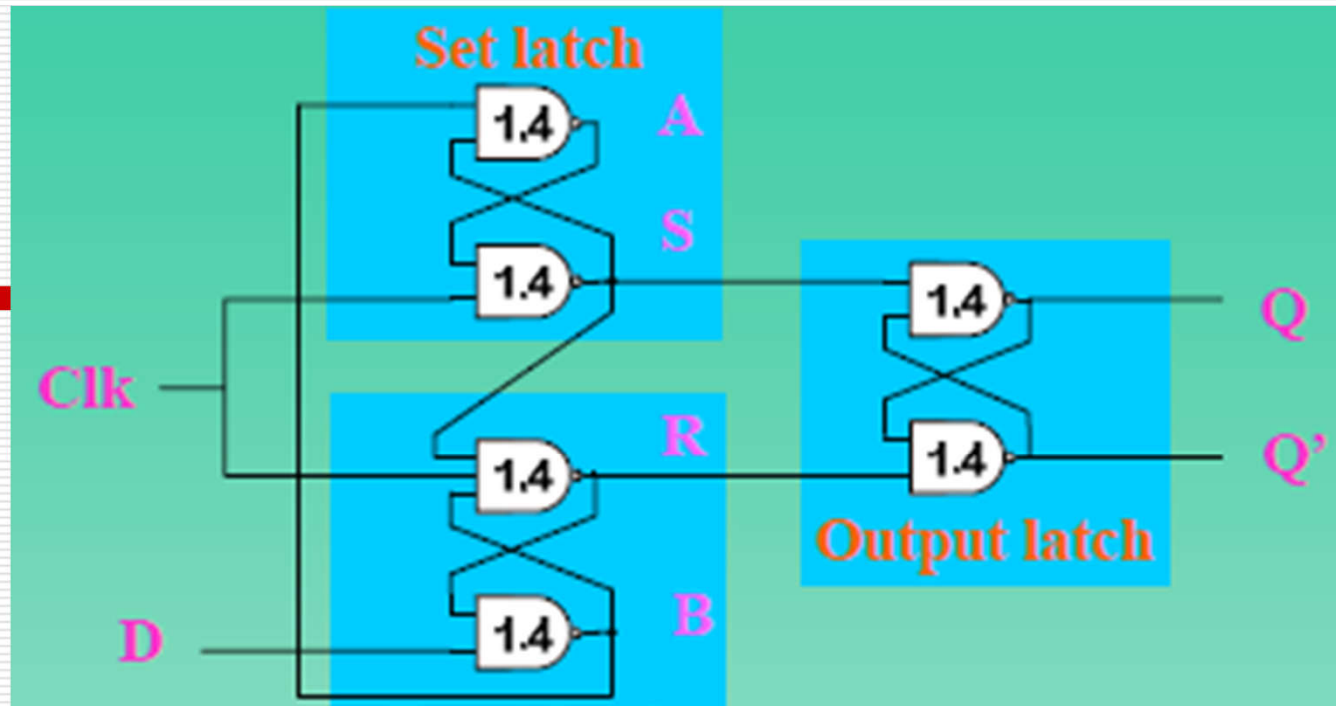
Clk = 1

- t_4 clk=1, S=1 det.

- $t_4 + 1.4\text{ns}$ R \rightarrow 0 det.

- $t_4 + 4.2\text{ns}$ Q \rightarrow 0

Edge trigger FF

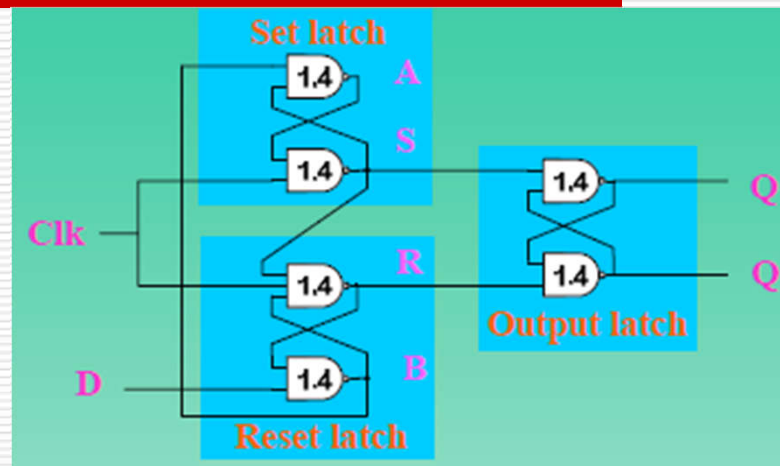


Clk = 1

-t5 D=1 nu este luat

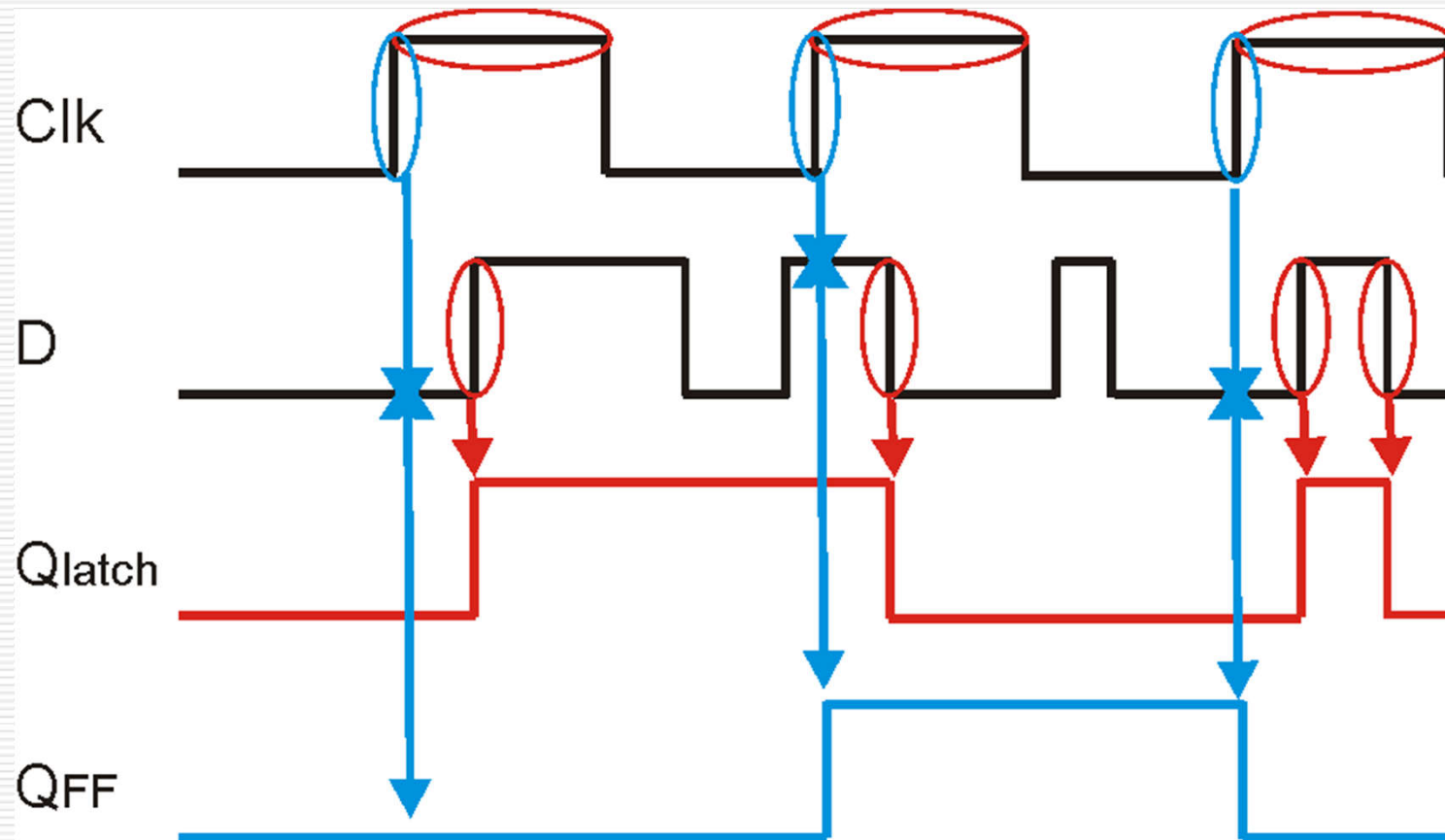
în considerare până la t6

Edge trigger FF



- Semnalele A, B memorează schimbările lui D pt. $\text{Clk} = 0$
- Semnalele S, R sunt 1 \rightarrow latch-ul de ieșire nu își modifică valoarea
- Pt. $\text{clk}=1$ fct. De R și S se resetează/setează și astfel se încarcă valoarea în latch-ul de ieșire

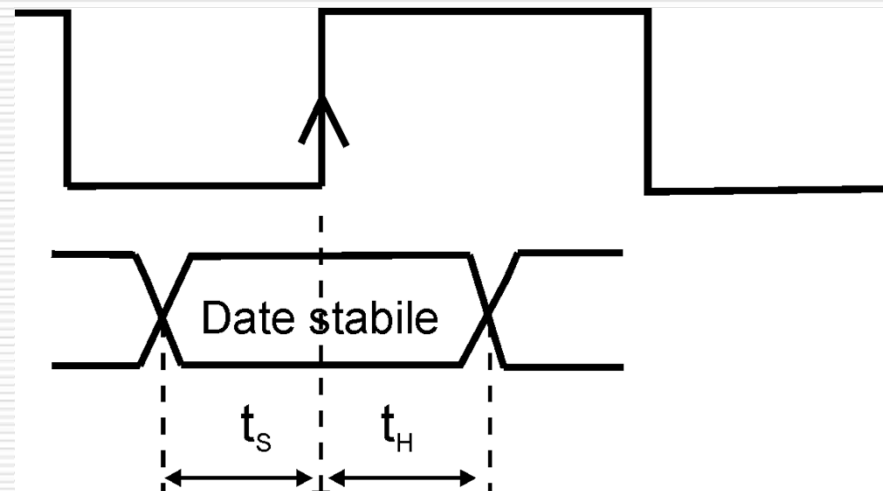
Latch sincron vs. FF sincron



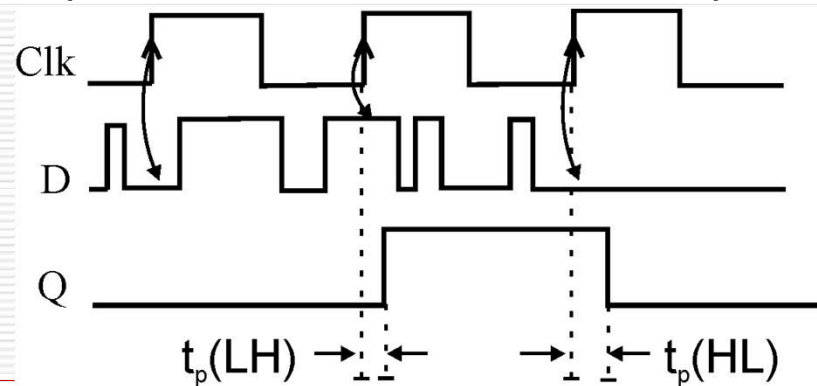
FF-uri

- contrângeri de timp in operarea FF-urilor - trebuie să aibe în vedere 3 timpi:
 - **Timpul de setup:** se referă la timpul necesar pentru ca semnalul de intrare (D) să rămână stabil înainte de apariția frontului semnalului de tact;
 - **Timpul de hold:** reprezintă timpul în care datele de intrare nu pot fi modificate după apariția frontului semnalului de tact în vederea încărcării corecte a acestora;
 - **Timpul aferent întârzierii datorate propagării (t_p):** constituie timpul necesar basculării FF-ului (clock to Q delay);
-

Prezentarea celor 3 timpi de propagare



Timpii de setup și hold pentru un FF care basculează pe front crescător al tactului



Timpi de propagare t_p

Tipuri de FF-uri

Flip-flop name	Flip-flop symbol	Characteristic table	Characteristic equation	Excitation table																																			
SR		<table> <tr> <th>S</th> <th>R</th> <th>Q(next)</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>NA</td> </tr> </table>	S	R	Q(next)	0	0	0	0	1	0	1	0	1	1	1	NA	$Q(next)=S+R'Q$ $SR=0$	<table> <tr> <th>Q</th> <th>Q(next)</th> <th>S</th> <th>R</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>X</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>X</td> <td>0</td> </tr> </table>	Q	Q(next)	S	R	0	0	0	X	0	1	1	0	1	0	0	1	1	1	X	0
S	R	Q(next)																																					
0	0	0																																					
0	1	0																																					
1	0	1																																					
1	1	NA																																					
Q	Q(next)	S	R																																				
0	0	0	X																																				
0	1	1	0																																				
1	0	0	1																																				
1	1	X	0																																				
JK		<table> <tr> <th>J</th> <th>K</th> <th>Q(next)</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>Q'</td> </tr> </table>	J	K	Q(next)	0	0	0	0	1	0	1	0	1	1	1	Q'	$Q(next)=JQ'+K'Q$	<table> <tr> <th>Q</th> <th>Q(next)</th> <th>J</th> <th>K</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>X</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>X</td> </tr> <tr> <td>1</td> <td>0</td> <td>X</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>X</td> <td>0</td> </tr> </table>	Q	Q(next)	J	K	0	0	0	X	0	1	1	X	1	0	X	1	1	1	X	0
J	K	Q(next)																																					
0	0	0																																					
0	1	0																																					
1	0	1																																					
1	1	Q'																																					
Q	Q(next)	J	K																																				
0	0	0	X																																				
0	1	1	X																																				
1	0	X	1																																				
1	1	X	0																																				
D		<table> <tr> <th>D</th> <th>Q(next)</th> </tr> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> </table>	D	Q(next)	0	0	1	1	$Q(next)=D$	<table> <tr> <th>Q</th> <th>Q(next)</th> <th>D</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	Q	Q(next)	D	0	0	0	0	1	1	1	0	0	1	1	1														
D	Q(next)																																						
0	0																																						
1	1																																						
Q	Q(next)	D																																					
0	0	0																																					
0	1	1																																					
1	0	0																																					
1	1	1																																					
T		<table> <tr> <th>T</th> <th>Q(next)</th> </tr> <tr> <td>0</td> <td>Q</td> </tr> <tr> <td>1</td> <td>Q'</td> </tr> </table>	T	Q(next)	0	Q	1	Q'	$Q(next)=TQ'+T'Q$	<table> <tr> <th>Q</th> <th>Q(next)</th> <th>T</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	Q	Q(next)	T	0	0	0	0	1	1	1	0	1	1	1	0														
T	Q(next)																																						
0	Q																																						
1	Q'																																						
Q	Q(next)	T																																					
0	0	0																																					
0	1	1																																					
1	0	1																																					
1	1	0																																					

Modalitati de descriere: circuite secvențiale

- Tabelul caracteristic:
 - Pentru fiecare combinație de intrare, funcție de starea curentă este precizată starea următoare
 - Ecuația caracteristică:
 - Ecuația rezultată în urma aplicării unei metode de minimizare
 - Tabelul excitațiilor:
 - Folosit la sinteza circuitului
 - Specifică intrările necesare pentru a trece din starea curentă în starea următoare
 - Diagrame/grafuri de stare:
 - Graf orientat, în care valorile posibile (stările) sunt reprezentate prin cercuri, iar tranzițiile prin arce
-

Circuite secvențiale reprezentare

□ Circuitele secvențiale:

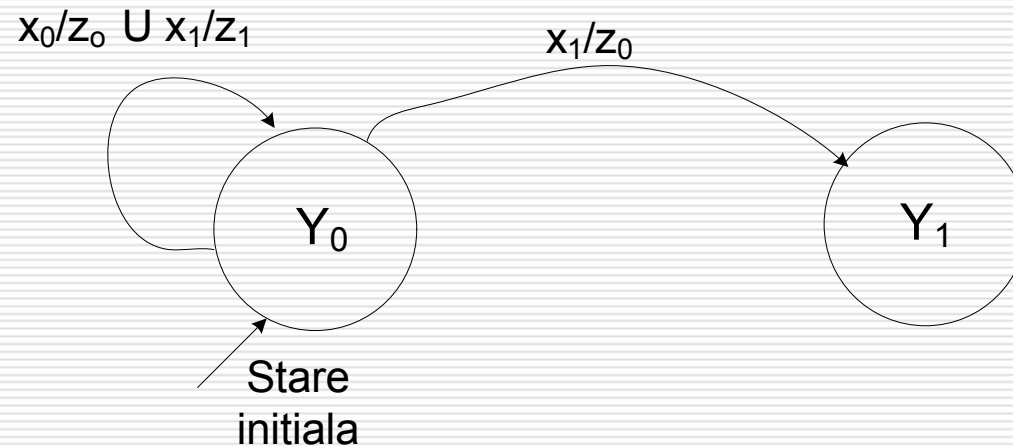
- **MEALY** sunt caracterizate prin faptul că starea următoare și ieșirea la un moment dat depind de starea ***prezentă*** și de ***intrarea prezentă***;
- **MOORE** sunt caracterizate prin faptul că ieșirea depinde **numai** de **starea circuitului**. Starea următoare depinde de intrarea prezentă;

- Modelele matematice ale circuitelor secvențiale se numesc în teoria comutațiilor **automate finite**.
-

Circuite secvențiale: diagrame e stare

□ circuite de tip Mealy:

- fiecare nod se notează cu simbolul stării pe care o reprezintă,
- arcul care pleacă din nod se notează cu perechea intrarea care a generat tranziția circuitului/ ieșirea generată în timpul tranziției.

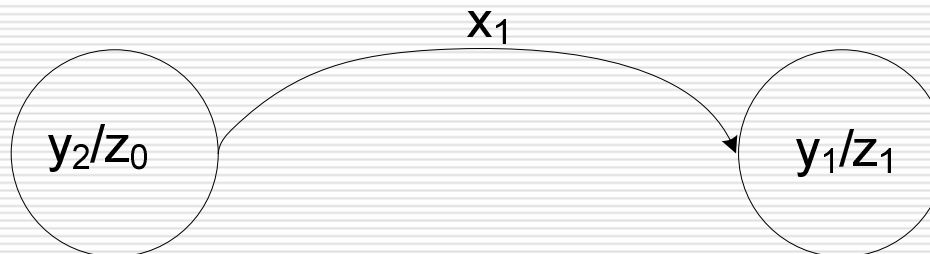


Starea inițială se marchează printr-o săgeată aplicată nodului respectiv.

Circuite secvențiale: diagrame e stare

□ **circuite de tip Moore:**

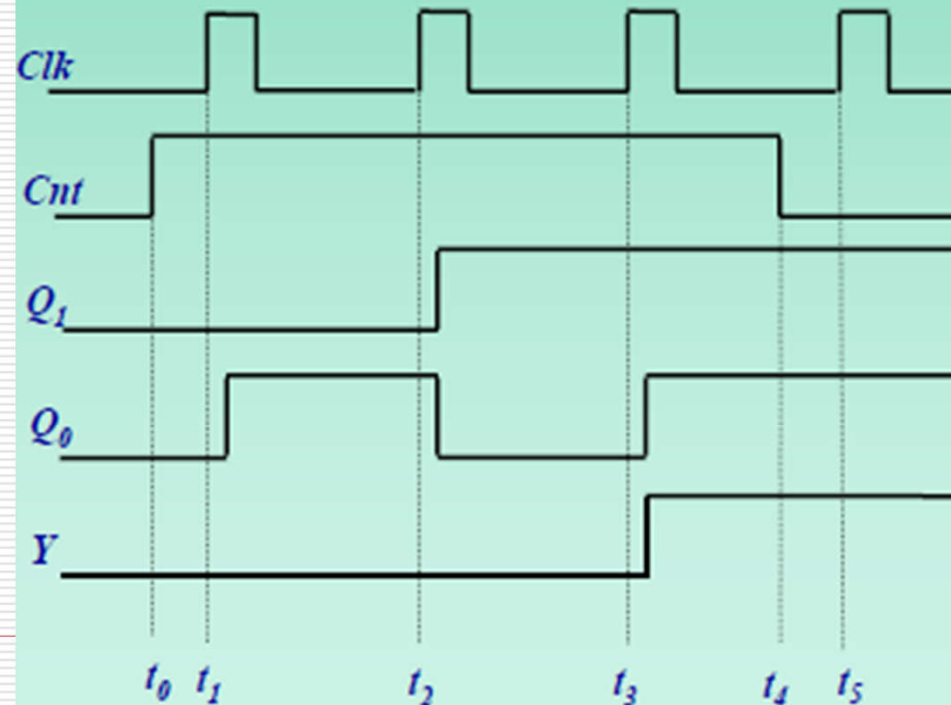
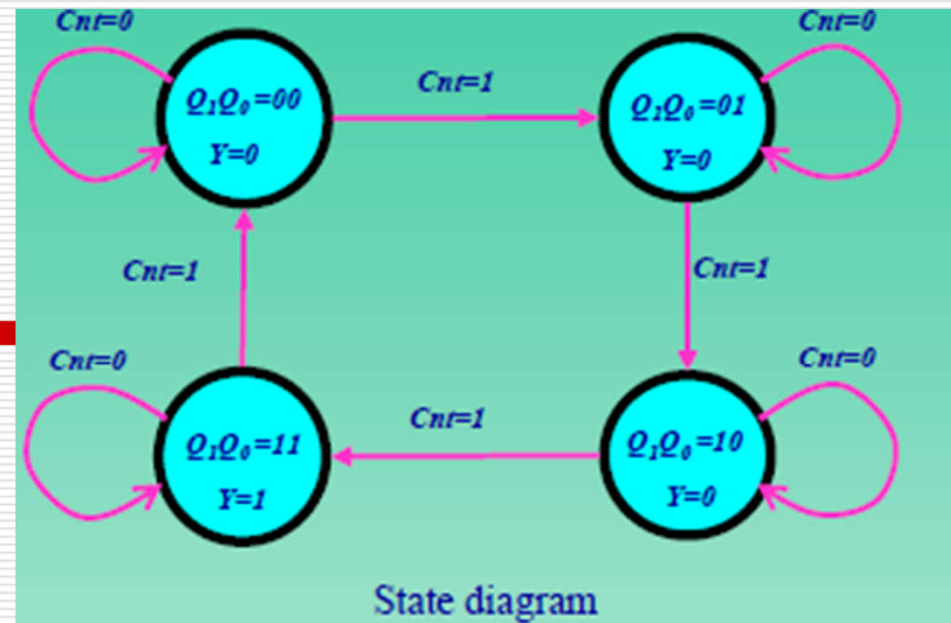
- nodurile diagramei de stări se notează simbolul stării corespondente și ieșirile
- arcul are notată intrarea care a generat tranziția.



Circuite secvențiale: diagrame e stare & tabelul tranzițiilor

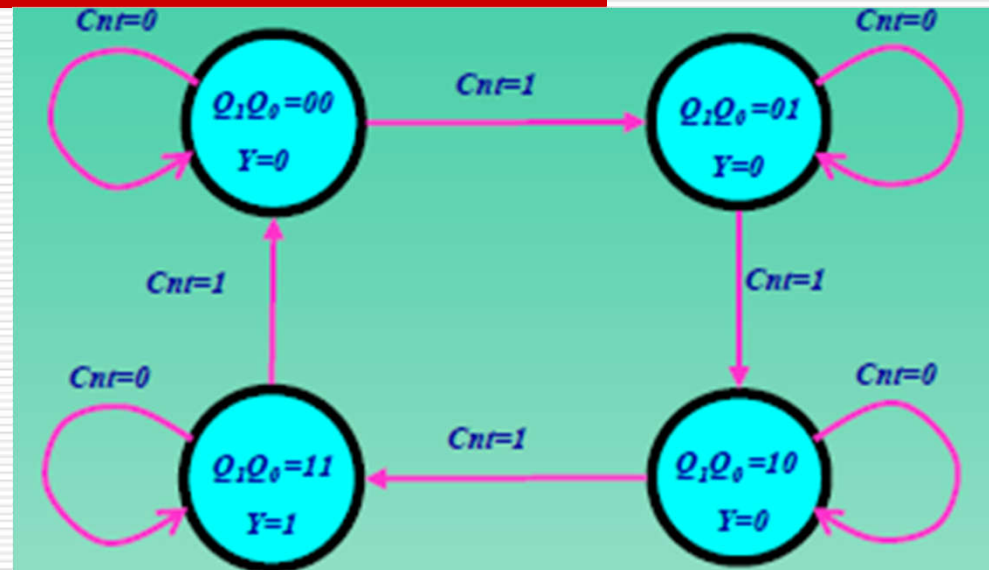
□ Moore

PRESENT STATE Q_1Q_0	NEXT STATE $Q_1(next) \ Q_0(next)$		OUTPUTS Y
	Cnt=0	Cnt=1	
00	00	01	0



Circuite secvențiale: diagrame de stare & tabelul tranzițiilor

□ Moore



PRESENT STATE Q_1Q_0	NEXT STATE $Q_1(next) Q_0(next)$		OUTPUTS Y
	Cnt=0	Cnt=1	
00	00	01	0
01	01	10	0
10	10	11	0
11	11	00	1

State and output table

Circuite secvențiale: diagrame e stare & tabelul tranzițiilor

<i>PRESENT STATE</i> Q_1Q_0	<i>NEXT STATE</i> $Q_1(next) Q_0(next)$		<i>OUTPUTS</i> Y
	Cnt=0	Cnt=1	
00	00	01	0
01	01	10	0
10	10	11	0
11	11	00	1

State and output table

$$Q_0(next) = D_0 = Cnt' Q_0 + Cnt Q_0'$$

$$Q_1(next) = D_1 = Cnt' Q_1 + Cnt Q_1' Q_0 + Cnt Q_1 Q_0'$$

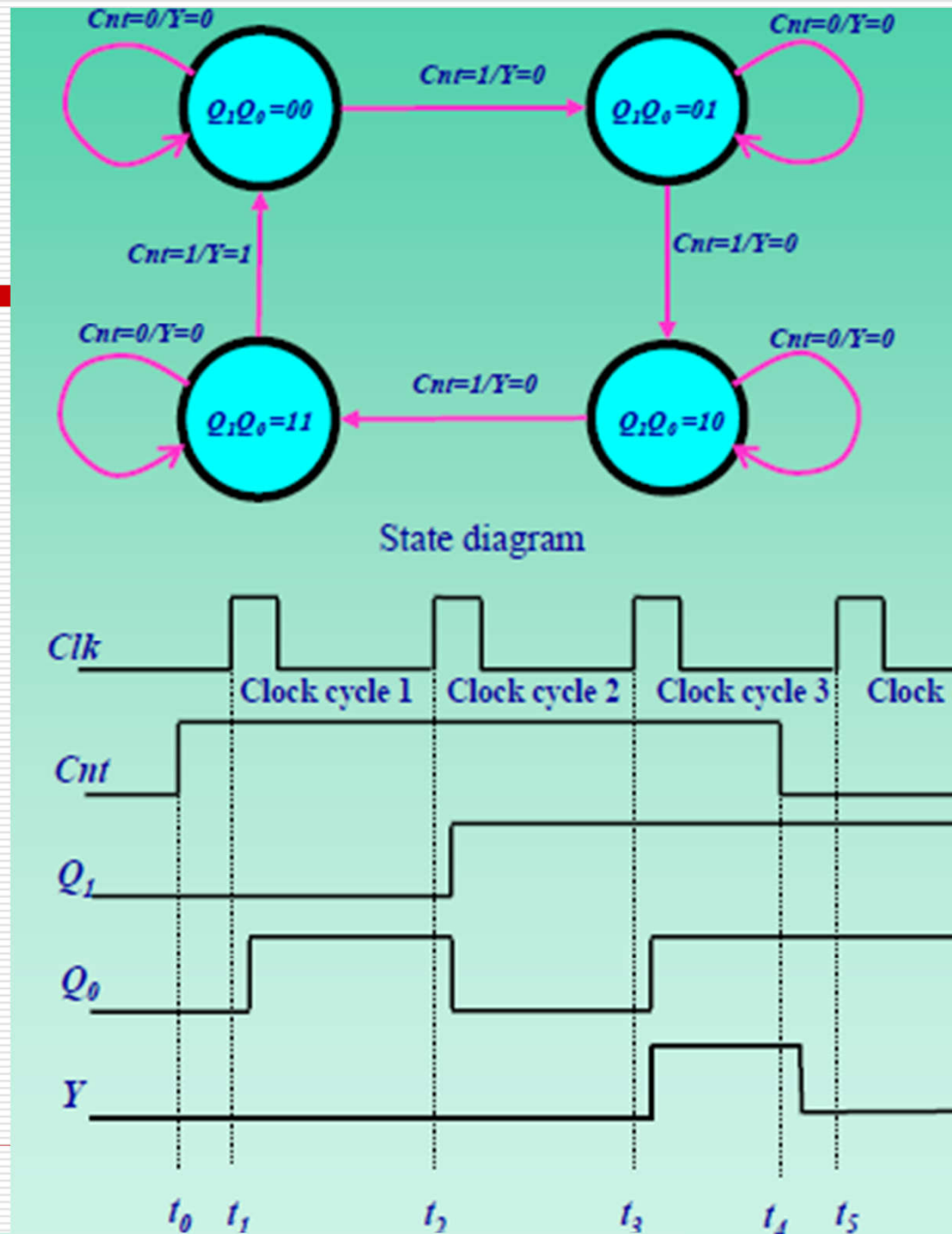
$$Y = Q_0 Q_1$$

Next-state and output equation

Circuite secvențiale: diagrame e stare & tabelul tranzițiilor

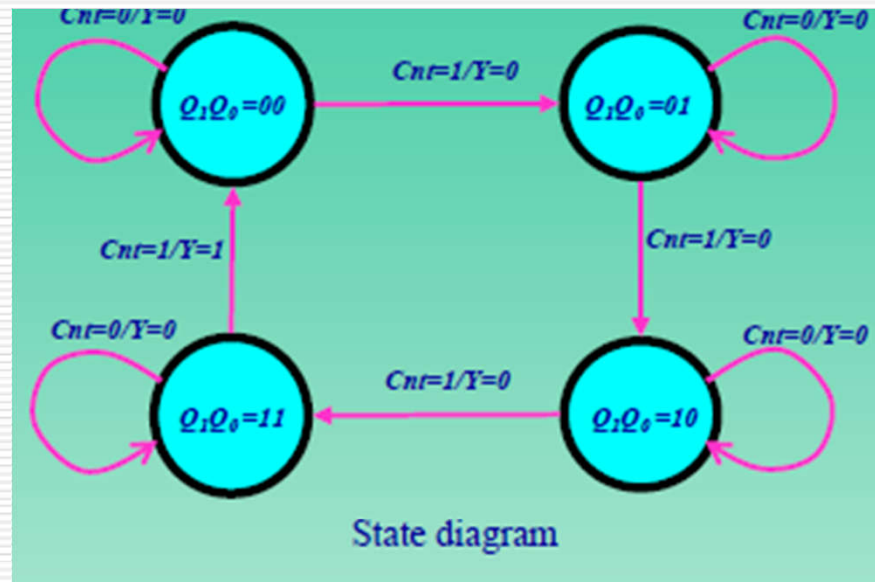
□ Mealy

PRESENT STATE Q_1Q_0	NEXT STATE /OUTPUTS $Q_1(next) Q_0(next)/Y$	
	Cnt=0	Cnt=1
00	01	00
01	10	01
10	11	10
11	00	11



Circuite secvențiale: diagrame e stare & tabelul tranzițiilor

□ Mealy



PRESENT STATE Q_1Q_0	NEXT STATE /OUTPUTS $Q_1(next) Q_0(next)/Y$	
	Cnt=0	Cnt=1
0 0	0 0 / 0	0 1 / 0
0 1	0 1 / 0	1 0 / 0
1 0	1 0 / 0	1 1 / 0
1 1	1 1 / 0	0 0 / 1

Circuite secvențiale: diagrame e stare & tabelul tranzițiilor

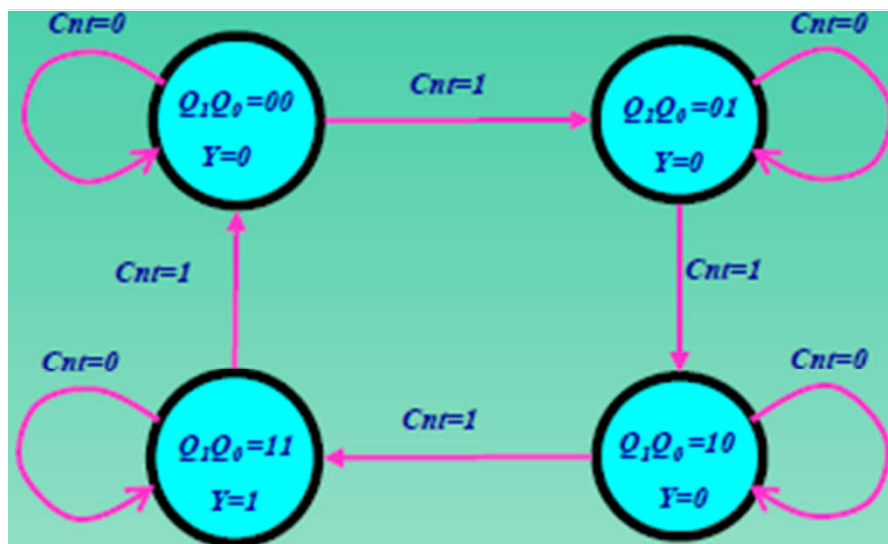
□ Mealy

<i>PRESENT STATE</i> $Q_1 Q_0$	<i>NEXT STATE /OUTPUTS</i> $Q_1(next) Q_0(next)/Y$	
	Cnt=0	Cnt=1
0 0	0 0 / 0	0 1 / 0
0 1	0 1 / 0	1 0 / 0
1 0	1 0 / 0	1 1 / 0
1 1	1 1 / 0	0 0 / 1

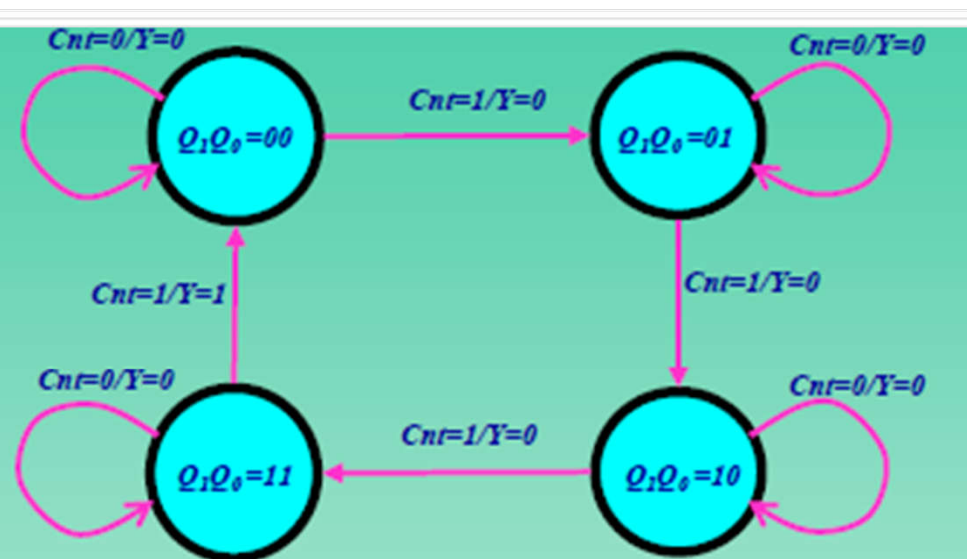
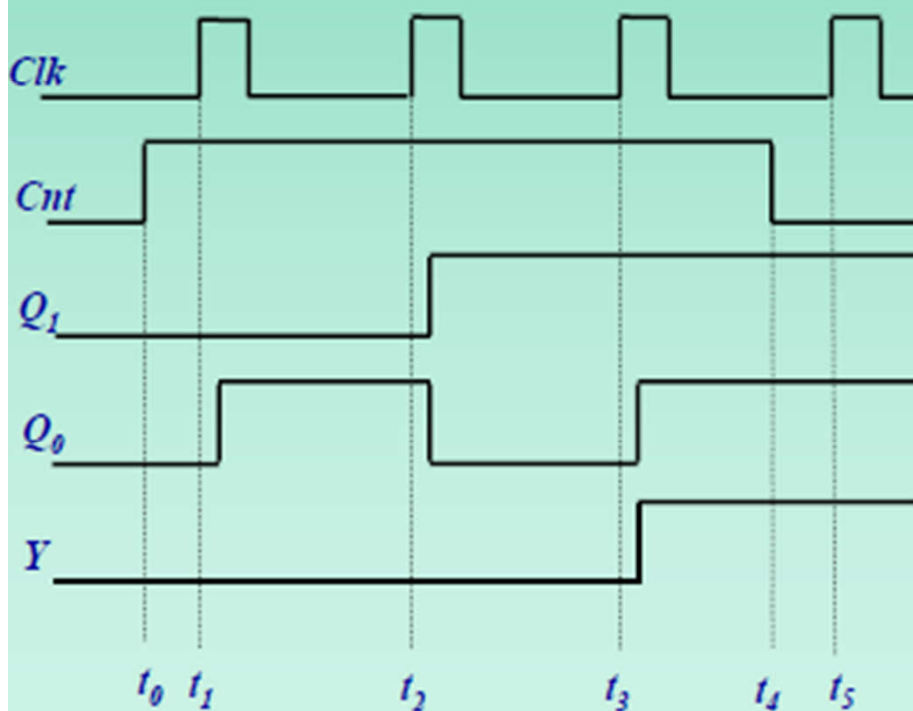
$$Q_0(next) = D_0 = Cnt' Q_0 + Cnt Q_0'$$

$$Q_1(next) = D_1 = Cnt' Q_1 + Cnt Q_1' Q_0 + Cnt Q_1 Q_0'$$

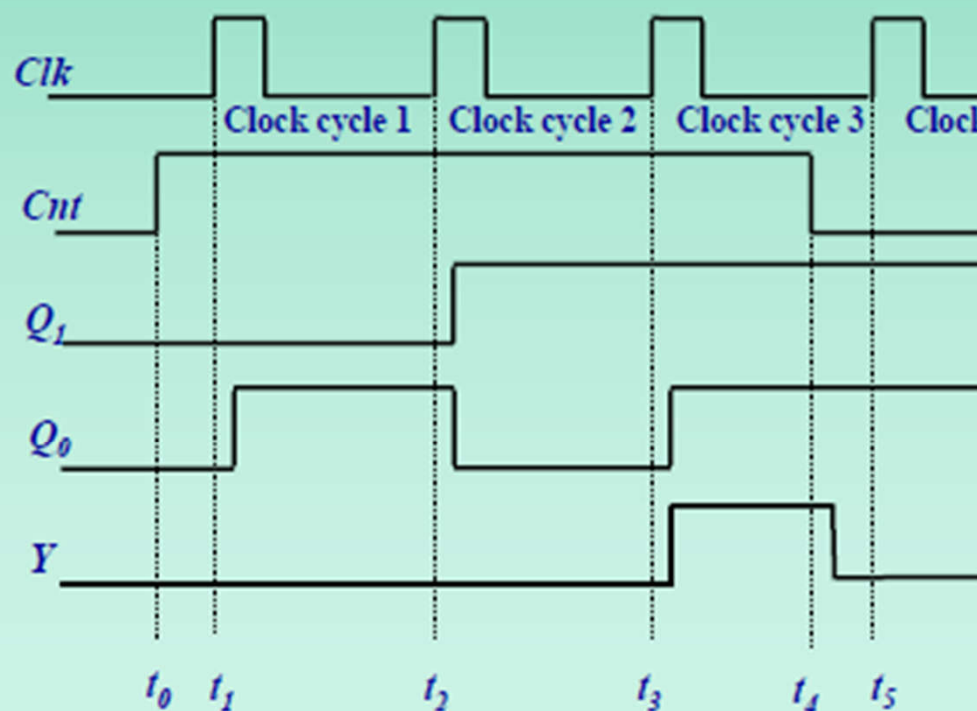
$$Y = Cnt Q_0 Q_1$$



State diagram

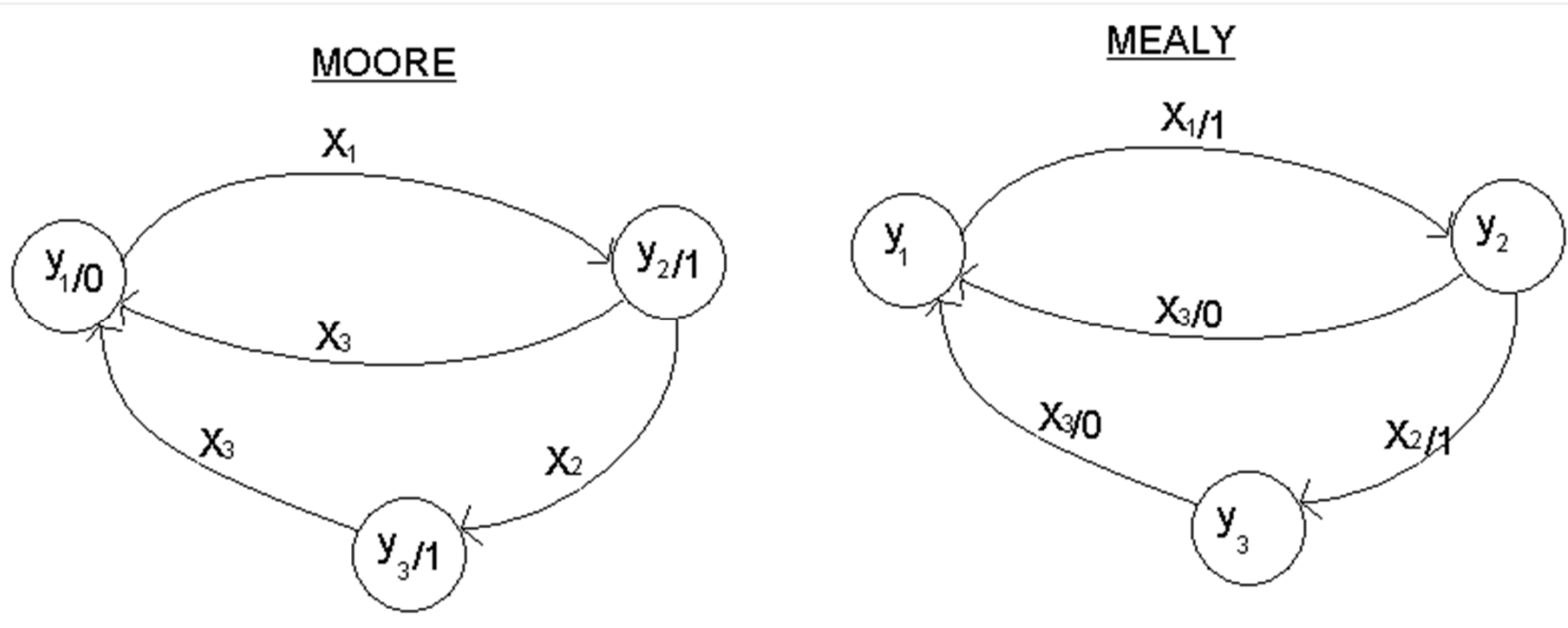


State diagram



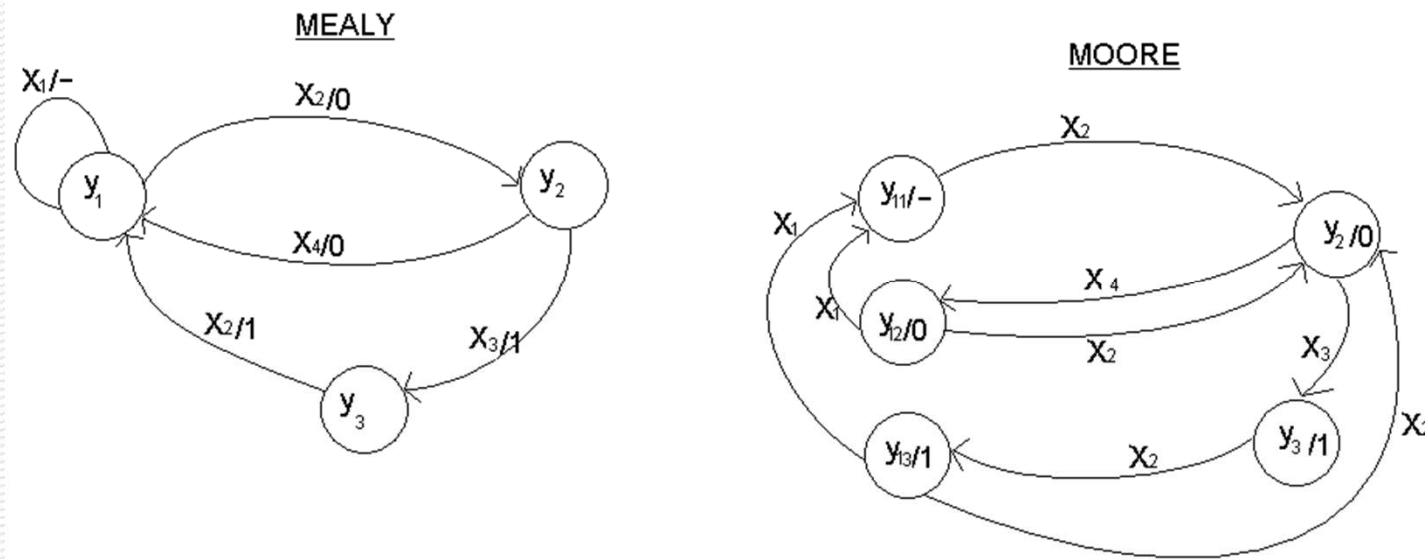
Transformarea diagramei de stări Moore în diagramă de stări Mealy

- se realizează transferând ieșirile din nodurile modelului Moore pe arcele ce conduc spre nodul respectiv.



Transformarea diagramei de stări Mealy în diagramă de stări Moore

- ❑ fiecărei stări din modelul Mealy i se asociază tot atâtea stări în modelul Moore echivalent, câte ieșiri diferite generează modelul Mealy în timpul tranziției spre starea respectivă, ieșirile fiind generate în noile stări ale modelului MOORE.



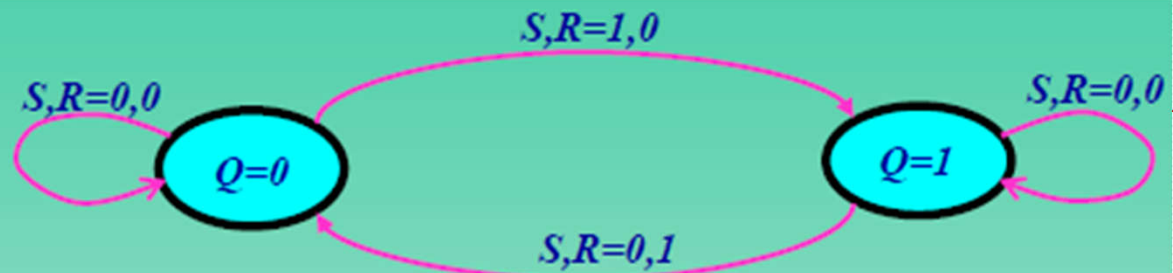
Tipuri de FF-uri

Flip-flop name	Flip-flop symbol	Characteristic table	Characteristic equation	Excitation table																																			
SR		<table> <tr> <th>S</th> <th>R</th> <th>Q(next)</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>NA</td> </tr> </table>	S	R	Q(next)	0	0	0	0	1	0	1	0	1	1	1	NA	$Q(next)=S+R'Q$ $SR=0$	<table> <tr> <th>Q</th> <th>Q(next)</th> <th>S</th> <th>R</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>X</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>X</td> <td>0</td> </tr> </table>	Q	Q(next)	S	R	0	0	0	X	0	1	1	0	1	0	0	1	1	1	X	0
S	R	Q(next)																																					
0	0	0																																					
0	1	0																																					
1	0	1																																					
1	1	NA																																					
Q	Q(next)	S	R																																				
0	0	0	X																																				
0	1	1	0																																				
1	0	0	1																																				
1	1	X	0																																				
JK		<table> <tr> <th>J</th> <th>K</th> <th>Q(next)</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>Q'</td> </tr> </table>	J	K	Q(next)	0	0	0	0	1	0	1	0	1	1	1	Q'	$Q(next)=JQ'+K'Q$	<table> <tr> <th>Q</th> <th>Q(next)</th> <th>J</th> <th>K</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>X</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>X</td> </tr> <tr> <td>1</td> <td>0</td> <td>X</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>X</td> <td>0</td> </tr> </table>	Q	Q(next)	J	K	0	0	0	X	0	1	1	X	1	0	X	1	1	1	X	0
J	K	Q(next)																																					
0	0	0																																					
0	1	0																																					
1	0	1																																					
1	1	Q'																																					
Q	Q(next)	J	K																																				
0	0	0	X																																				
0	1	1	X																																				
1	0	X	1																																				
1	1	X	0																																				
D		<table> <tr> <th>D</th> <th>Q(next)</th> </tr> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> </table>	D	Q(next)	0	0	1	1	$Q(next)=D$	<table> <tr> <th>Q</th> <th>Q(next)</th> <th>D</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	Q	Q(next)	D	0	0	0	0	1	1	1	0	0	1	1	1														
D	Q(next)																																						
0	0																																						
1	1																																						
Q	Q(next)	D																																					
0	0	0																																					
0	1	1																																					
1	0	0																																					
1	1	1																																					
T		<table> <tr> <th>T</th> <th>Q(next)</th> </tr> <tr> <td>0</td> <td>Q</td> </tr> <tr> <td>1</td> <td>Q'</td> </tr> </table>	T	Q(next)	0	Q	1	Q'	$Q(next)=TQ'+T'Q$	<table> <tr> <th>Q</th> <th>Q(next)</th> <th>T</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	Q	Q(next)	T	0	0	0	0	1	1	1	0	1	1	1	0														
T	Q(next)																																						
0	Q																																						
1	Q'																																						
Q	Q(next)	T																																					
0	0	0																																					
0	1	1																																					
1	0	1																																					
1	1	0																																					

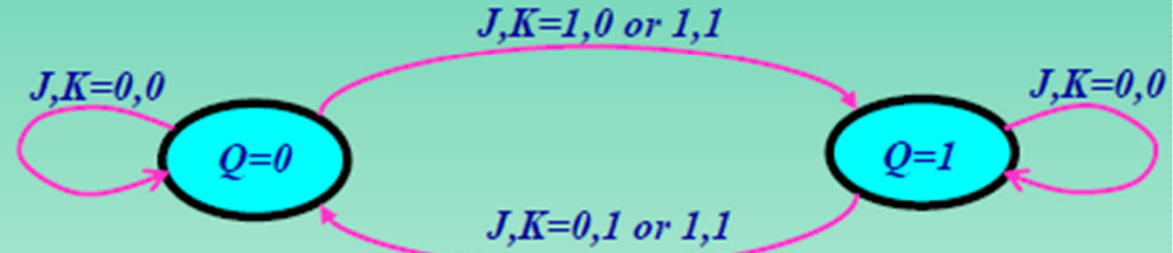
Flip-flop name

State diagram

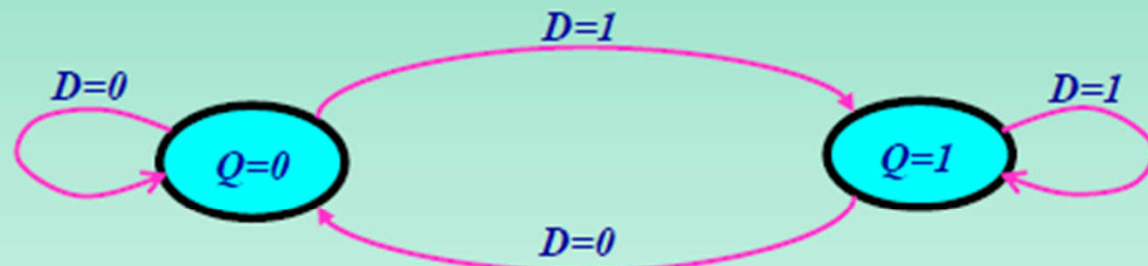
SR



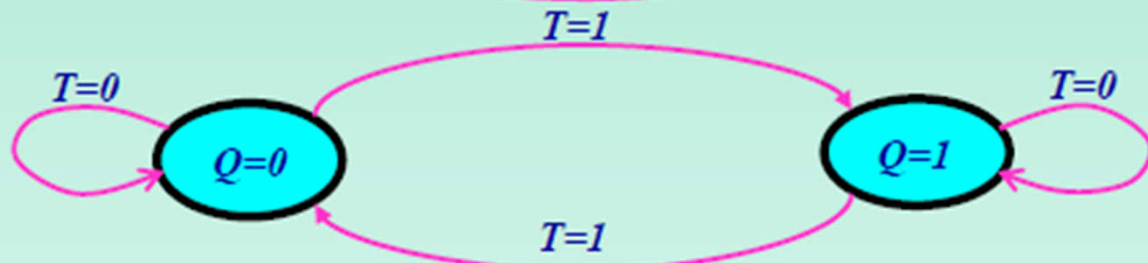
JK



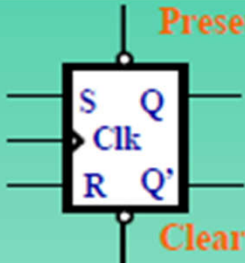
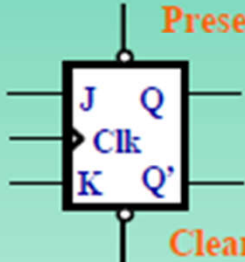
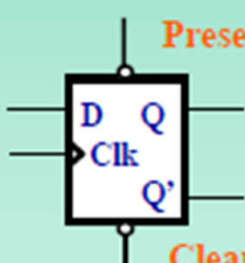
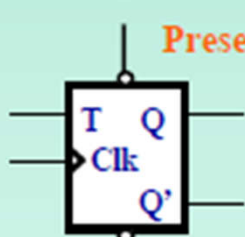
D



T



Simbol: FF-uri cu intrări de date asincrone

Flip-flop name	Flip-flop symbol
SR	 The SR flip-flop symbol is a square with a clock input (Clk) on the left. It has two data inputs: S (Set) and R (Reset). It has two outputs: Q and Q' (complement of Q). Asynchronous inputs are shown as 'Preset' (top) and 'Clear' (bottom), both with small circles indicating active-low connections.
JK	 The JK flip-flop symbol is a square with a clock input (Clk) on the left. It has two data inputs: J and K. It has two outputs: Q and Q'. Asynchronous inputs are shown as 'Preset' (top) and 'Clear' (bottom), both with small circles indicating active-low connections.
D	 The D flip-flop symbol is a square with a clock input (Clk) on the left. It has one data input: D. It has two outputs: Q and Q'. Asynchronous inputs are shown as 'Preset' (top) and 'Clear' (bottom), both with small circles indicating active-low connections.
T	 The T flip-flop symbol is a square with a clock input (Clk) on the left. It has one data input: T. It has two outputs: Q and Q'. Asynchronous inputs are shown as 'Preset' (top) and 'Clear' (bottom), both with small circles indicating active-low connections.

Avantajele logicii sincrone

- Simplifică analiza timpilor de propagare în cadrul unui circuit design care prezintă un semnal de tact comun pentru toate elementele de memorare.
 - Astfel analiza de timp (timpul de setup și timpul de hold, precum și timpul de propagare când frecvențele de lucru se apropie de frecvența maximă admisă) se reduce la **analiza în raport cu un singur element** de memorare.
 - Modelul sincron separă partea de design aferentă logicii combinaționale de cea secvențială.
 - De asemenea oferă protecție în ceea ce privește hazardul specific logicii combinaționale.
-

Întrebări?

**Enough Talking Let's Get To It
!!Brace Yourselves!!**

