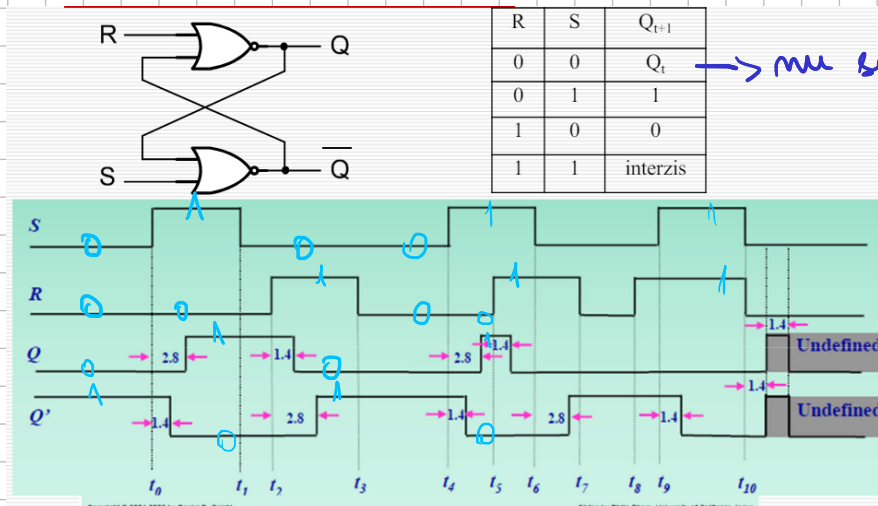


→ Si pe pozitive (front cre.)
Si pe pacientul pozitiv

b) Formă idealizată a semnalului

S-R Latch (SAU - NU) asincron

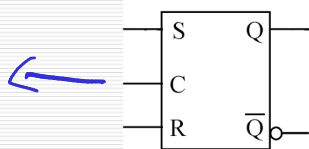


→ nu se produce nicio schimbare

- 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 simultan are un comportament nedeterministic
- 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 (de ex. activitatea de comutare la design time)
 - Se impune regula mai strictă ca S și R să nu fie activate simultan

Gated SR-latch

em.

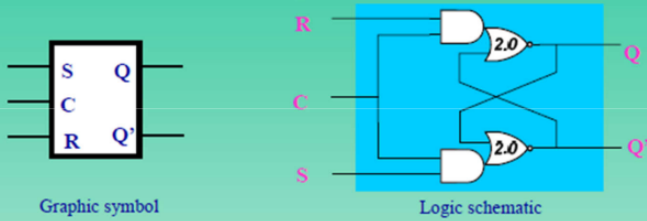


S	R	C	Q_{next}	\overline{Q}_{next}
0	0	1	Q	\overline{Q}
0	1	1	0	1
1	0	1	1	0
1	1	1	-	-
*	*	0	Q	\overline{Q}

→ se păstrează

- 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

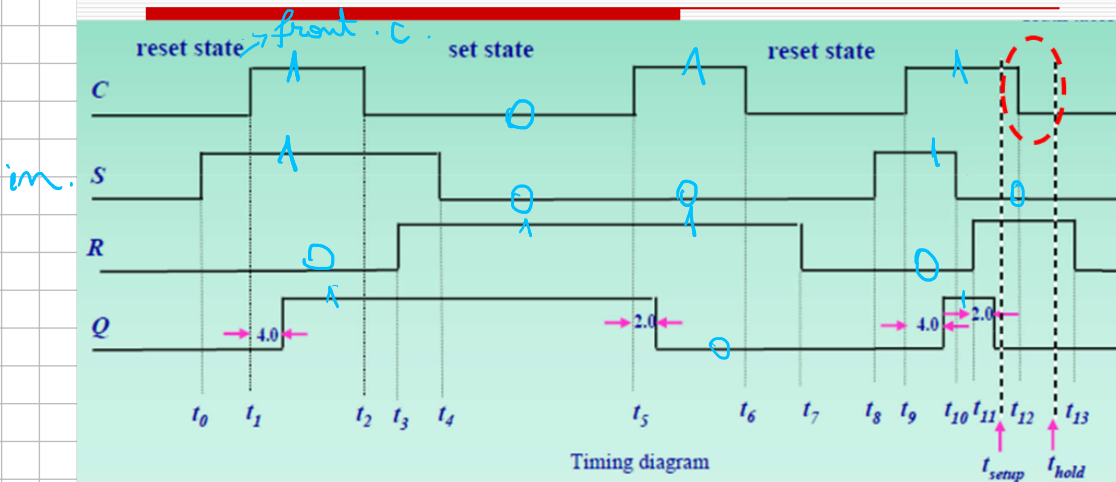
•Control signal C activates the latch



C	S	R	Q	Q(next)
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

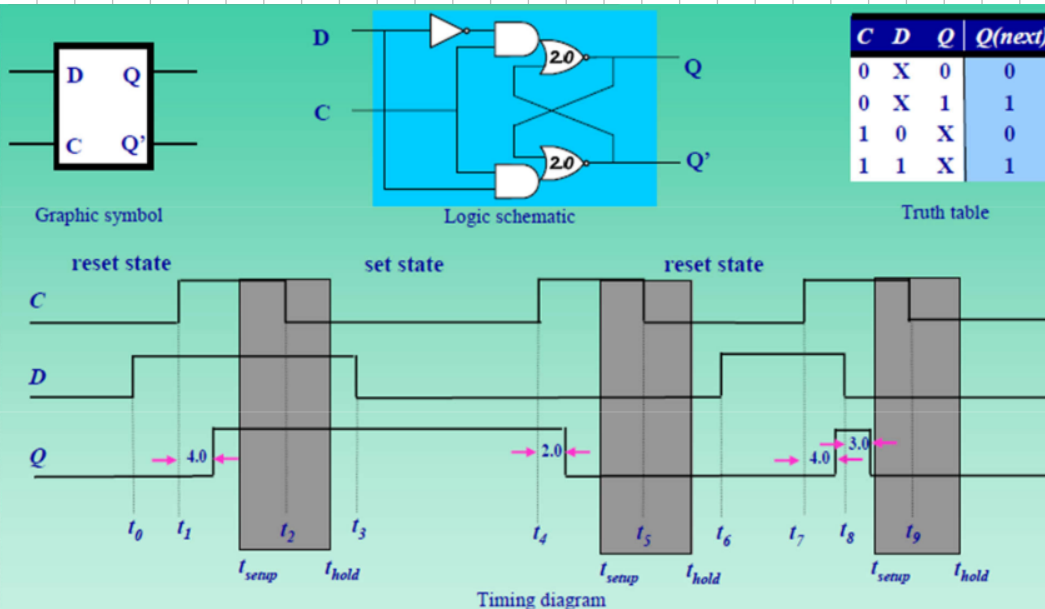
Gated SR-latch (SR-latch sincron)



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

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

Gated D-Latch



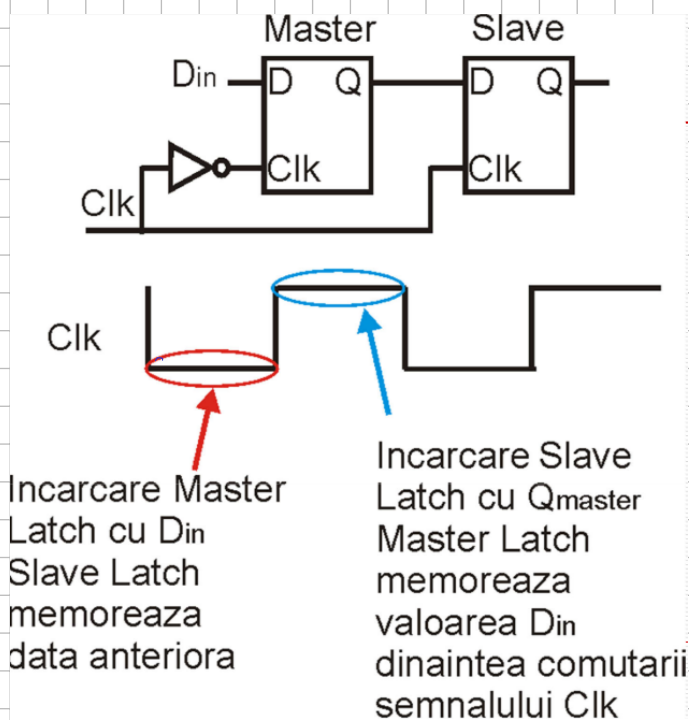
$$C = 0 \Rightarrow Q_m = Q$$

$$C = 1 \Rightarrow Q_m = D$$

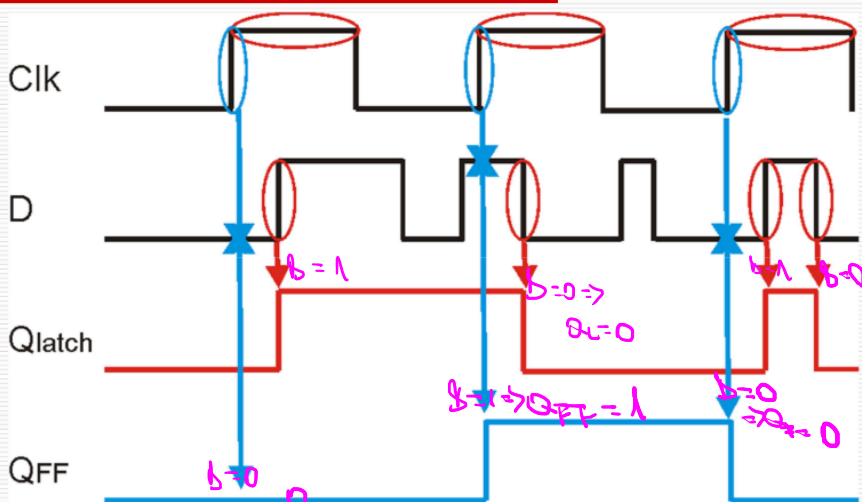
Flip-flop-uri

- 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



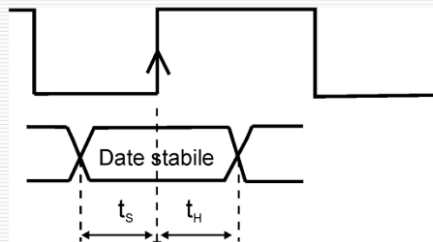
Latch sincron vs. FF sincron



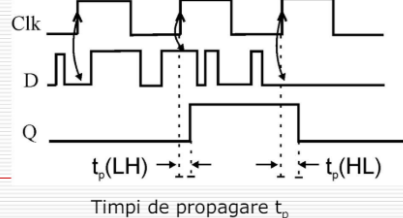
□ contrângeri de timp în operarea FF-urilor - trebuie să aibe în vedere 3 timpi:

- **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



Timpii de propagare t_p

Tipuri de FF-uri (P.S.: sunt de toate în gile)

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>Q</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	Q	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	Q																																					
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>Q</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	Q	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	Q																																					
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

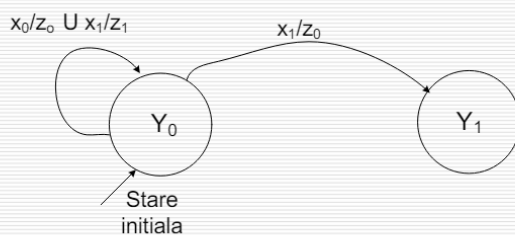
□ Circuitele secvențiale:

- **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 de tip Mealy:

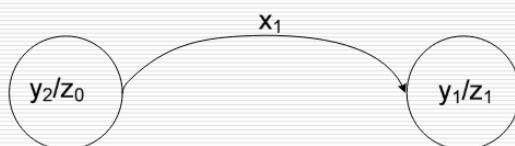
- fiecare nod se notează cu simbolul stării pe care o reprezintă,
- arcul care pleacă din nod se notează cu perechea intrarea care a generat tranziția circuitului/ ieșirea generată în timpul tranziției.



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

□ circuite de tip Moore:

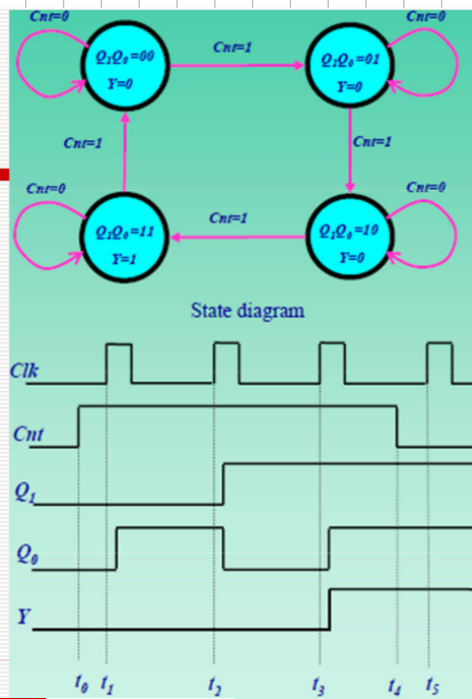
- nodurile diagramei de stări se notează simbolul stării corespundente și ieșirile
- arcul are notată intrarea care a generat tranziția.



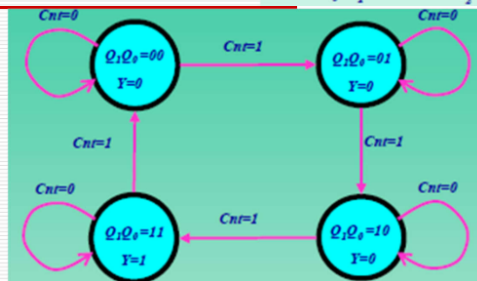
Circuite secvențiale: diagrame de stare & tabelul tranzițiilor

Moore

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



Moore



PRESENT STATE $Q_1 Q_0$	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

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

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

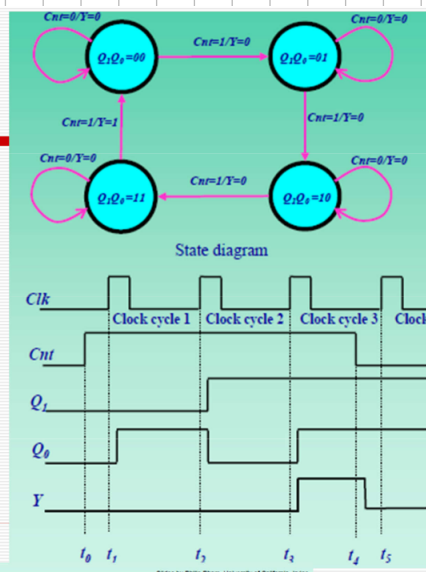
$$Y = Q_0 Q_1$$

Next-state and output equation

Circuite secvențiale: diagrame de stare & tabelul tranzițiilor

Mealy

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



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

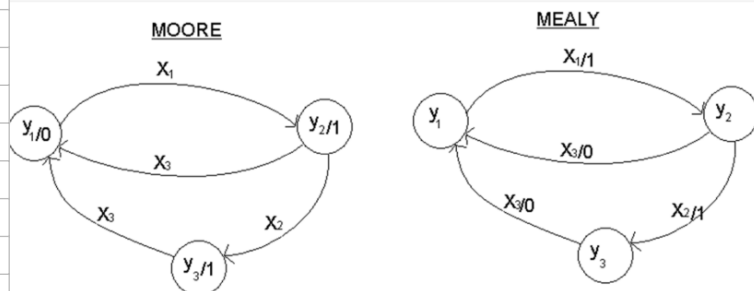
$$Q_0(next) = D_0 = Cnt' Q_0 + Cnt Q_0'$$

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

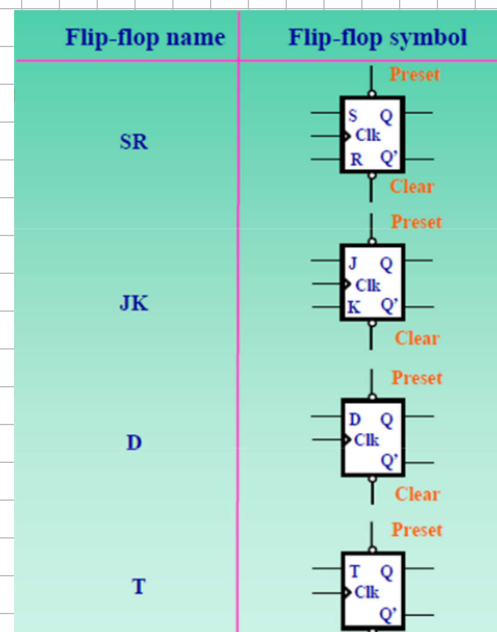
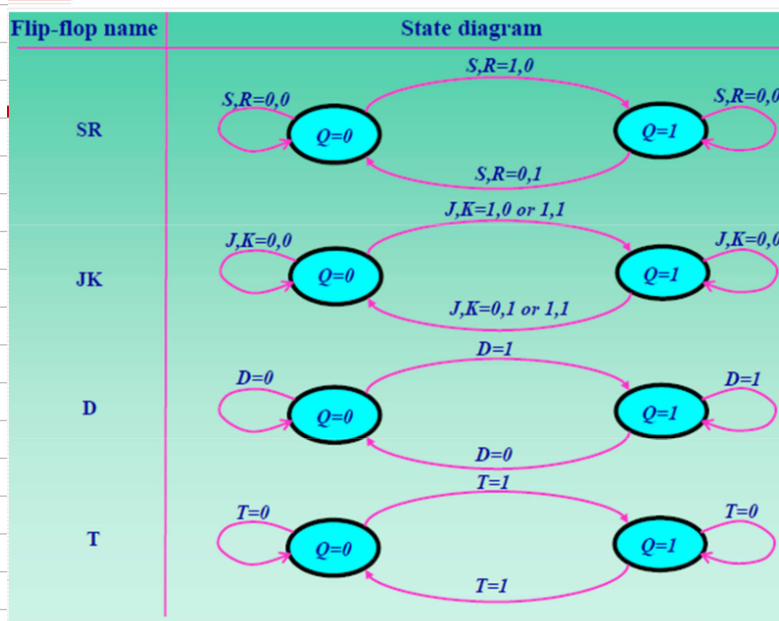
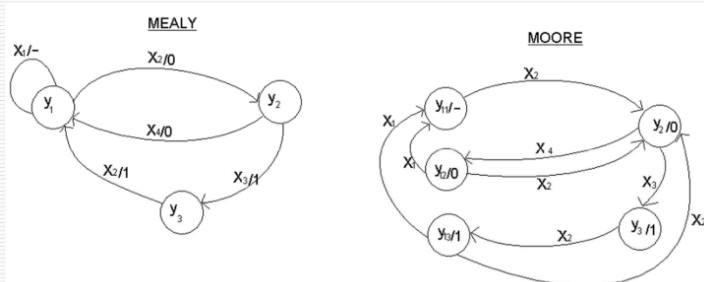
$$Y = Cnt Q_0 Q_1$$

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.



- 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.



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.