

Digital- och datorteknik

Synkrona sekvensnät

Analys

Syntes av räknare

Syntes av små sekvensnät

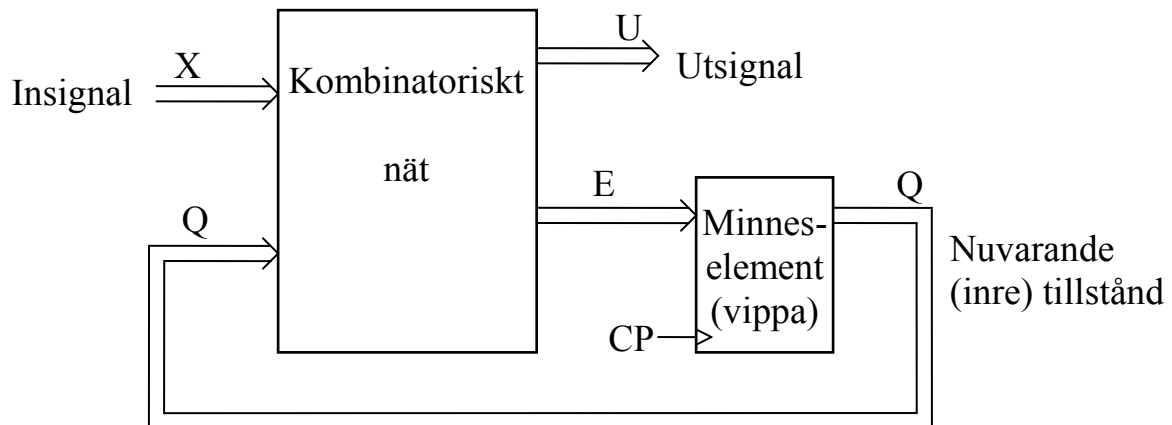
Extrauppgifter

X.1 - X.14

på analys och syntes

Analys av synkrona sekvensnät

Allmän modell av ett synkront sekvensnät



$$E = f(Q, X)$$

$$Q^+ = g(E) = \delta(Q, X)$$

$$U = \lambda(Q, X)$$

(Q^+ är Q -värdet efter nästa klockpuls.
 Q^+ är alltså nästa tillstånd.)

Analys innebär att man bestämmer **f -, g -, δ - och λ -funktionerna** som **booleska uttryck** och som **tabeller**.

Från **$\delta(\lambda)$ -tabellen** ritas man sedan **tillståndsgrafen**, som grafiskt visar hur övergångar mellan inre tillstånd sker. Eventuellt kompletteras tillståndsgrafen med pulsdigram, som visar klockpulser, tillståndsvariabler och utsignaler.

Arbetsgång:

1. Tag fram

$$E = f(Q, X)$$

$$Q^+ = \delta(Q, X)$$

$$U = \lambda(Q, X)$$
 i form av booleska uttryck.

2. Sammanställ

$$E = f(Q, X)$$

$$Q^+ = \delta(Q, X)$$

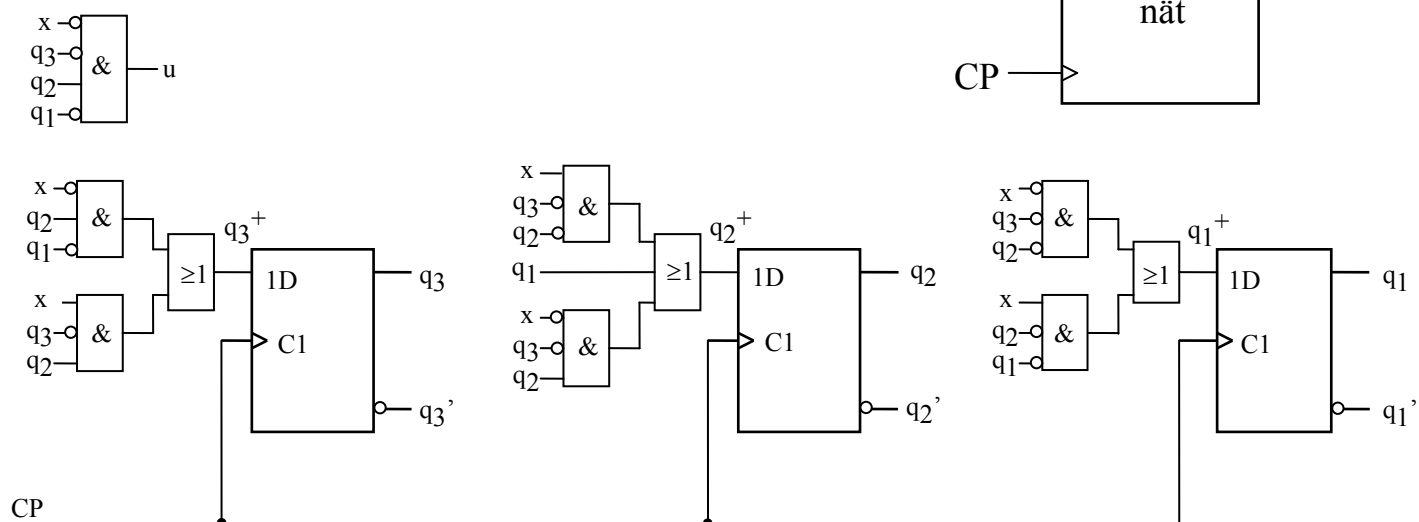
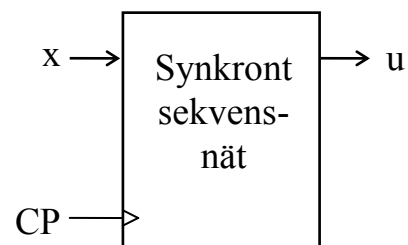
$$U = \lambda(Q, X)$$
 i form av en $\delta(\lambda)$ -tabell.

3. Rita en tillståndsgraf utgående från tabellen.

4. Rita eventuellt pulsdigram för CP, Q och U.

Exempel på analys:

Tillämpa analysstegen 1 - 3 på nedanstående nät.

**1.**

$q_1^+ =$

$q_2^+ =$

$q_3^+ =$

$u =$

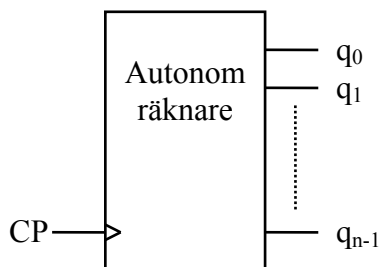
3.**2.**

x	q ₃	q ₂	q ₁	q ₃ ⁺	q ₂ ⁺	q ₁ ⁺	u
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

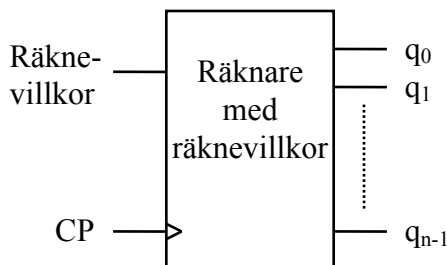
Syntes (konstruktion) av synkrona räknare

En (synkron) **räknare** registrerar antalet klockpulser som uppträder på klockingången. Räknaren är **synkron** om samtliga vippor är kopplade till samma klocksignal och triggar på samma flank. Räknarens utsignaler utgörs av Q-värdena hos räknarens samtliga vippor. En räknare sägs räkna **modulo-N** om den antar N olika tillstånd då $M > N$ klockpulser uppträder på klockingången. En modulo-N räknare antar alltså samma tillstånd för klockpuls nr i och nr $N + i$.

En **autonom** räknare saknar insignaler som styr hur räknaren räknar och följer därför alltid en och samma tillståndsssekvens. Se figuren nedan.



En räknare med **räknevillkor** har en eller flera insignaler som styr vilket nästa tillstånd räknaren antar. Med k styrsignaler kan man få en räknare att räkna med maximalt 2^k olika tillståndsssekvenser.



Övningsuppgifter:

- Konstruera en autonom synkron modulo-6 räknare med räknesekvens och vipptyp enligt nedan. Samtliga vipptyper nedan triggar på positiv flank. Dessutom får AND-, OR- och INVERTERAR-grindar användas.

a)	Räknesekvens: $q_1q_2q_3 = 000, 001, 101, 111, 011, 010, 000$.	Vipptyp: D.
b)	Räknesekvens: - " -	Vipptyp: SR.
c)	Räknesekvens: - " -	Vipptyp: JK.
d)	Räknesekvens: - " -	Vipptyp: T.
- Konstruera en synkron räknare med en styrsignal x , som bestämmer räknarens funktion på följande sätt

$x = 0$ Räknaren räknar enligt sekvensen $q_1q_2 = 00, 01, 11, 10, 00$

$x = 1$ Räknaren räknar enligt sekvensen $q_1q_2 = 00, 10, 11, 01, 00$

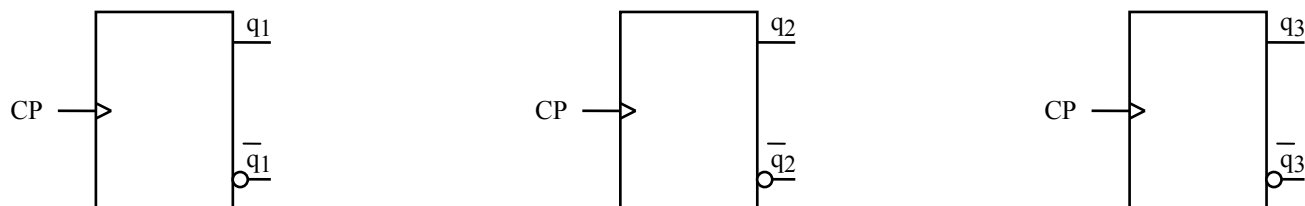
Av de två sekvenserna ovan framgår att räknaren är **reversibel** dvs den kan räkna en bestämd sekvens framåt eller bakåt.

JK-vippor som triggar på positiv flank, NAND-grindar och INVERTERARE får användas.

Lösningar på övningsuppgifter, syntes av synkrona räknare.

1. Antag att tillstånden skrivs som $q_1q_2q_3$.
Lösningen visas här för samtliga typer av vippor vi har gått igenom!

Räknaren har tre vippor med utsignalerna q_1 , q_2 och q_3 enligt figuren nedan.



Det gäller nu att bestämma insignalerna till vipporna så att räknesekvensen blir den rätta. Vi ställer därför upp en tabell som visar nuvarande (q_i) och nästa tillstånd (q_i^+) samt de insignalvärden som ger "rätt" nästa tillstånd för de fyra olika typerna av vippor. Dessa insignalvärden får man ur excitationstabellerna nedan.

Excitationstabeller:

D-vippan

q	q^+	D
0	0	0
0	1	1
1	0	0
1	1	1

SR-vippan

q	q^+	S	R
0	0	0	–
0	1	1	0
1	0	0	1
1	1	–	0

JK-vippan

q	q^+	J	K
0	0	0	–
0	1	1	–
1	0	–	1
1	1	–	0

T-vippan

q	q^+	T
0	0	0
0	1	1
1	0	1
1	1	0

q_1	q_2	q_3	q_1^+	q_2^+	q_3^+	D_1	D_2	D_3	S_1	R_1	S_2	R_2	S_3	R_3	J_1	K_1	J_2	K_2	J_3	K_3	T_1	T_2	T_3
0	0	0	0	0	1	0	0	1	0	–	0	–	1	0	0	–	0	–	1	–	0	0	1
0	0	1	1	0	1	1	0	1	1	0	0	–	–	0	1	–	0	–	–	0	1	0	0
0	1	0	0	0	0	0	0	0	0	–	0	1	0	–	0	–	–	1	0	–	0	1	0
0	1	1	0	1	0	0	1	0	0	–	–	0	0	1	0	–	–	0	–	1	0	0	1
1	0	0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
1	0	1	1	1	1	1	1	1	–	0	1	0	–	0	–	0	1	–	–	0	0	1	0
1	1	0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
1	1	1	0	1	1	0	1	1	0	1	–	0	–	0	–	1	–	0	–	0	1	0	0

För D-vippor gäller att nästa tillstånd (q^+) är lika med D-värdet före klockningen. Därför skall D-värdet för varje D-vippa väljas till motsvarande q^+ -värde, vilket framgår av tabellen ovan.

Karnaughdiagram för vippornas insignaler visas på nästa sida.

Ext-7 (Ver 2013-04-14)

a) D-vippor

$$q_2 q_3$$

D_1	00	01	11	10
0	0	1	0	0
1	-	1	0	-

$$D_1 = \overline{q_2} q_3$$

D ₂	00	01	11	10
0	0	0	1	0
1	-	1	1	-

$$D_2 = q_1 + q_2 q_3$$

D ₃	00	01	11	10
0	1	1	0	0
1	-	1	1	-

$$D_3 = q_1 + \overline{q_2}$$

b) SR-vippor

S_1	00	01	11	10
0	0	1	0	0
1	-	-	0	-

$$S_1 = \overline{q_2} q_3$$

S_2	00	01	11	10
0	0	0	-	0
1	-	1	-	-

$$S_2 = q_1$$

S_3

	00	01	11	10
0	1	-	0	0
1	-	-	-	-

$$S_3 = \overline{q_2}$$

R_1	00	01	11	10
0	-	0	-	-
1	-	0	1	-

$$R_1 = q_2$$

R ₂	00	01	11	10
0	-	-	0	1
1	-	0	0	-

$$R_2 = \overline{q_3}$$

R_3	00	01	11	10
0	0	0	1	-
1	-	0	0	-

$$R_3 = \overline{q_1} q_2$$

c) JK-vippor

J_1	00	01	11	10
0	0	1	0	0
1	-	-	-	-

$$J_1 = \overline{q_2} q_3$$

J_2	00	01	11	10
0	0	0	-	-
1	-	1	-	-

$$J_2 = q_1$$

J_3	00	01	11	10
0	1	-	-	0
1	-	-	-	-

$$J_3 = \overline{q_2}$$

K_1	00	01	11	10
0	-	-	-	-
1	-	0	1	-

$$K_1 = q_2$$

K ₂	00	01	11	10
0	-	-	0	1
1	-	-	0	-

$$K_2 = \overline{q_3}$$

K ₃	00	01	11	10
0	-	0	1	-
1	-	0	0	-

$$K_3 = \overline{q_1} q_2$$

d) T-vippor

T ₁	00	01	11	10
0	0	1	0	0
1	-	0	1	-

$$T_1 = \overline{q_1} \overline{q_2} q_3 + q_1 q_2$$

T ₂	00	01	11	10
0	0	0	0	1
1	-	1	0	-

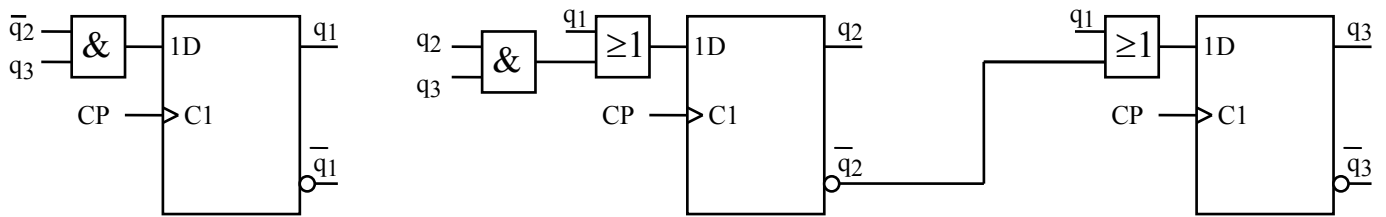
$$T_2 = q_1 \overline{q_2} + q_2 \overline{q_3}$$

T ₃	00	01	11	10
0	1	0	1	0
1	-	0	0	-

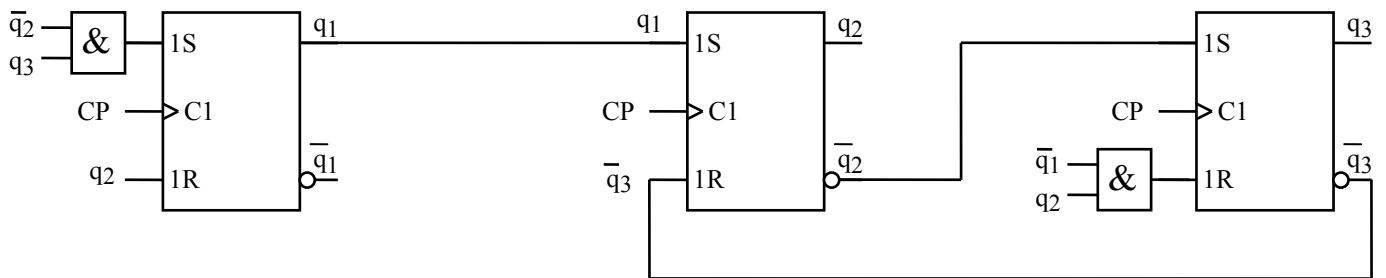
$$T_3 = \overline{q_2} \overline{q_3} + \overline{q_1} q_2 q_3$$

Nedan visas den kompletta räknaren med de olika vipptyperna.

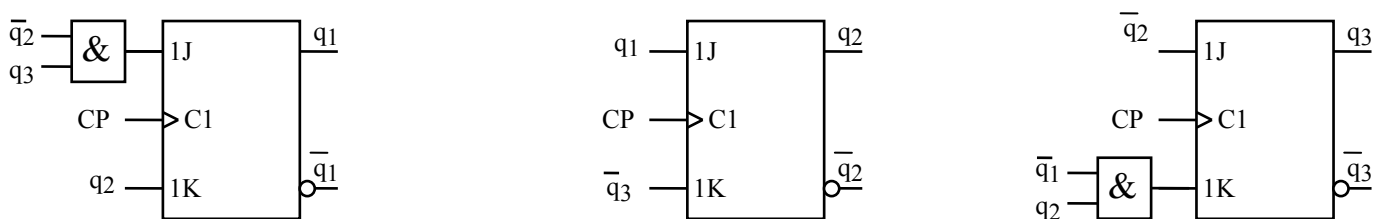
a)



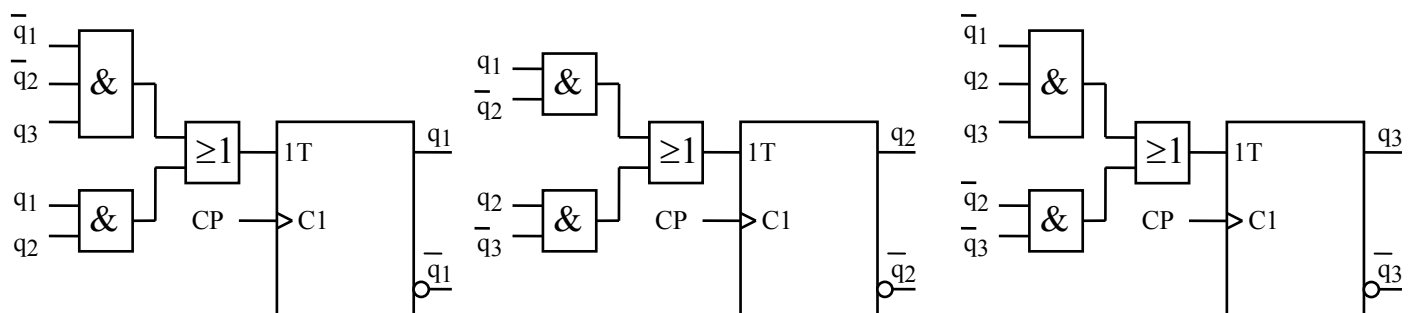
b)



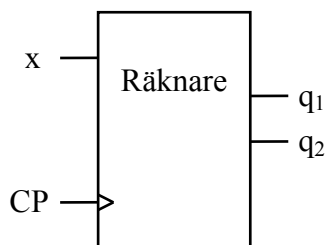
c)



d)



2.



Räknaren har en insignal x och två utsignaler q_1 och q_2 . Den innehåller alltså två JK-vippor.



Konstruktion av räknaren innebär att man skall bilda vippornas insignaler J_1 , K_1 , J_2 och K_2 av signalerna x , q_1 och q_2 så att rätt räknesekvenser bildas, dvs så att alla (q_1, q_2) följs av rätt (q_1^+, q_2^+) .

$x = 0$ Räknaren räknar enligt sekvensen $q_1 q_2 = 00, 01, 11, 10, 00$

$x = 1$ Räknaren räknar enligt sekvensen $q_1 q_2 = 00, 10, 11, 01, 00$

Tillståndstabell:

q_1	q_2	x	q_1^+	q_2^+	J_1	K_1	J_2	K_2
0	0	0	0	1	0	-	1	-
0	0	1	1	0	1	-	0	-
0	1	0	1	1	1	-	-	0
0	1	1	0	0	0	-	-	1
1	0	0	0	0	-	1	0	-
1	0	1	1	1	-	0	1	-
1	1	0	1	0	-	0	-	1
1	1	1	0	1	-	1	-	0

Excitationstabell:

q	q^+	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

Tillståndstabellen består av $2^3 = 8$ rader eftersom vi har tre variabler.

Man fyller först i kolumnerna q_1^+ och q_2^+ med hjälp av de givna insignalsekvenserna.

Därefter använder man excitationstabellen och fyller i J_i - och K_i -värden för varje par (q_i, q_i^+) för de två vipporna.

Med hjälp av tillståndstabellen ritas Karnaughdiagram för J_1 , K_1 , J_2 och K_2 . Se nästa sida!

	q_2x			
J_1 :	00	01	11	10
q_1				
0	0	1	0	1
1	-	-	-	-

$$J_1 = \overline{q_2}x + q_2\overline{x} = q_2 \oplus x$$

J_2 :	00	01	11	10
0	1	0	-	-
1	0	1	-	-

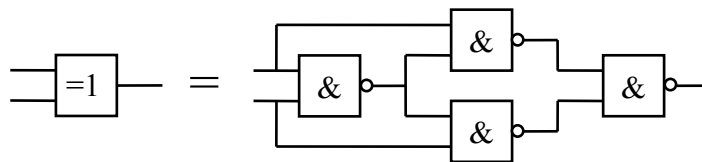
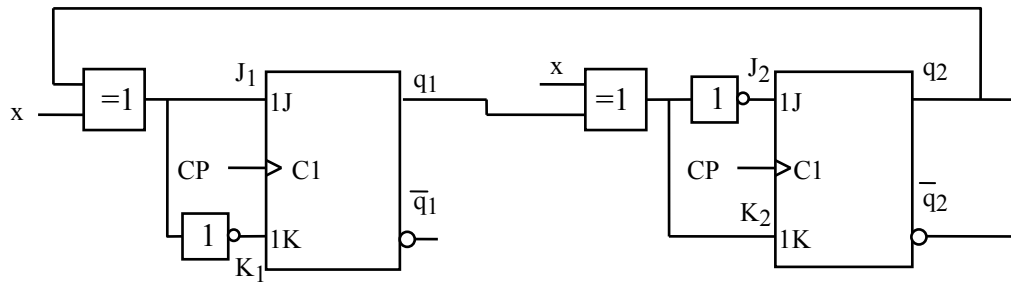
$$J_2 = q_1x + \overline{q_1}\overline{x} = \overline{(q_1 \oplus x)}$$

K_1 :	00	01	11	10
0	-	-	-	-
1	1	0	1	0

$$K_1 = \overline{q_2}\overline{x} + q_2x = \overline{(q_2 \oplus x)}$$

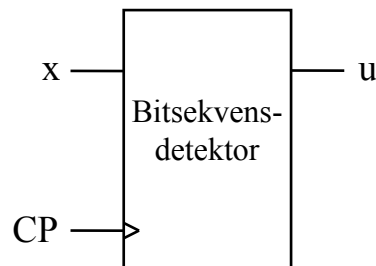
K_2 :	00	01	11	10
0	-	-	1	0
1	-	-	0	1

$$K_2 = \overline{q_1}x + q_1\overline{x} = q_1 \oplus x$$



Konstruktion av enkla synkrona sekvensnät

Metodiken vid konstruktion av enkla synkrona sekvensnät visas här genom att ett sekvensnät för detektering av en speciell bitsekvens, en bitsekvensdetektor, tas fram.



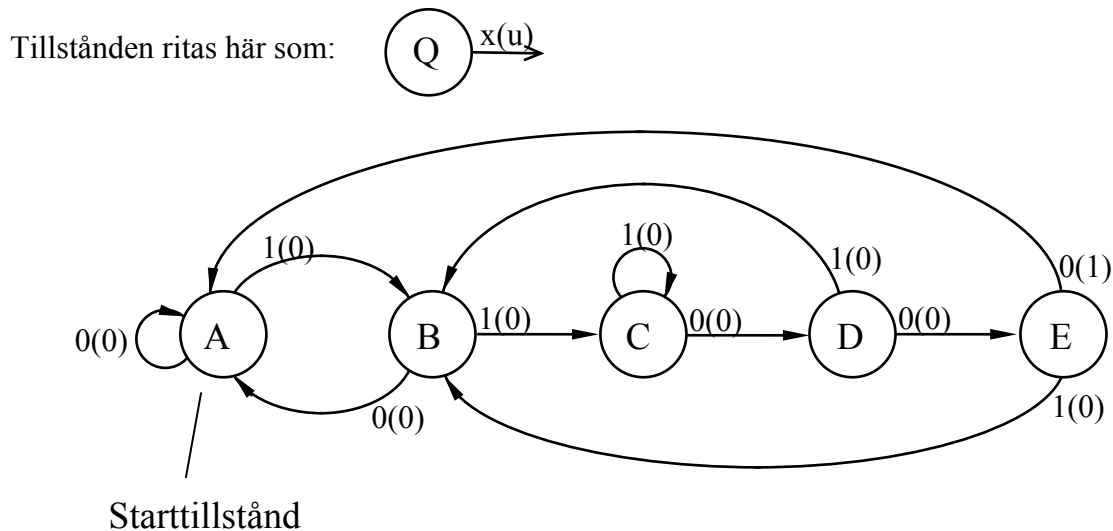
Bitsekvensdetektorn i figuren ovan skall uppfylla följande specifikation:

1. Bitsekvensdetektorn skall vara ett synkront sekvensnät med en insignal x och en utsignal u .
2. Utsignalen u skall anta värdet 1 om och endast om insignalen x haft värdena 11000 i de fem senaste klockpulsintervallen.
3. Utsignalen $u=1$ skall uppträda i samma klockpulsintervall som den sista nollan i en korrekt insignalsekvens.
4. Insignalen x är synkroniserad med klocksignalen CP så att om x ändrar värde så sker det direkt efter positiv flank på klocksignalen.
5. Sekvensnätet skall konstrueras med D-vippor (högst 3 st) med positiv flanktrigging samt ett minimalt antal standardgrindar.
6. Det får förutsättas att alla vippor i sekvensnätet har $Q=0$ från starten.

Konstruktionsarbetet

Steg 1:

Rita en tillståndsgraf som uppfyller specifikationen.



Denna tillståndsgraf uppfyller specifikationen och vi nöjer oss med detta i vår kurs.

Det finns metoder för att avgöra om en given tillståndsgraf innehåller onödiga tillstånd. Man kan därför första ta fram en icke minimal tillståndsgraf, som uppfyller en given specifikation. Därefter minimerar man grafen och får då en ny mindre tillståndsgraf, som också uppfyller specifikationen. Sådana metoder berörs inte i denna kurs.

Steg 2:

Motsvarande $\delta(\lambda)$ -tabell fylls i.

$\delta(\lambda)$		x	
		0	1
Q	A	A(0)	B(0)
	B	A(0)	C(0)
	C	D(0)	C(0)
	D	E(0)	B(0)
	E	A(1)	B(0)

Steg 3:

$\delta(\lambda)$ -tabellen kodas binärt, dvs tillstånden A, B, C, D och E binärkodas.

Eftersom antalet tillstånd i detta fall är 5 krävs det minst 3 tillståndsvariabler ($2^2 < 5 \leq 2^3 = 8$) för att varje tillstånd skall få ett unikt 3-bitars kodord.

Vilken binär kod som är bäst (optimal) i någon mening är ett svårt problem att knäcka.

Med våra begränsade kunskaper nöjer vi oss med en godtycklig binär kod, som ger en unik kombination av nollor och ettor för varje tillstånd. Vi hugger till med en kod enligt tabellen nedan, där kodordet för starttillståndet A har valts till 000 enligt punkt 6 i specifikationen!

Q	q ₂	q ₁	q ₀
A	0	0	0
B	0	1	0
C	0	1	1
D	0	0	1
E	1	0	0

Bokstäverna A, B, C, D och E i $\delta(\lambda)$ -tabellen byts nu ut mot de binära kodorden från tabellen ovan. Den resulterande $\delta(\lambda)$ -tabellen visas nedan.

$\delta(\lambda)$		x	
		0	1
q ₂ q ₁ q ₀	000	000(0)	010(0)
	010	000(0)	011(0)
	011	001(0)	011(0)
	001	100(0)	010(0)
	100	000(1)	010(0)

$\delta(\lambda)$ -tabellen visar alltså nu q_2^+ , q_1^+ , och q_0^+ och u som funktion av x, q_2 , q_1 , och q_0 eller annorlunda uttryckt

$$Q^+ = \delta(Q, X)$$

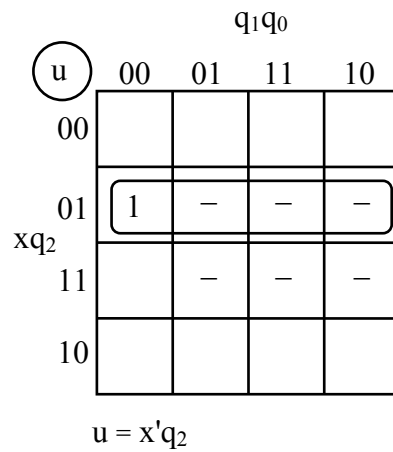
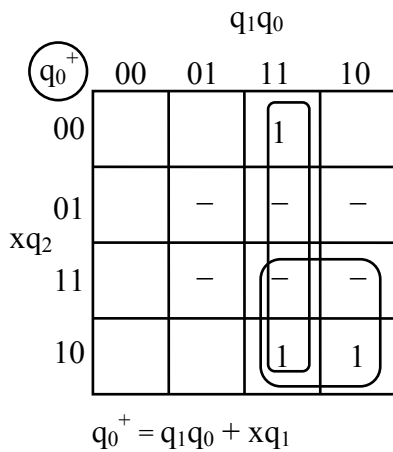
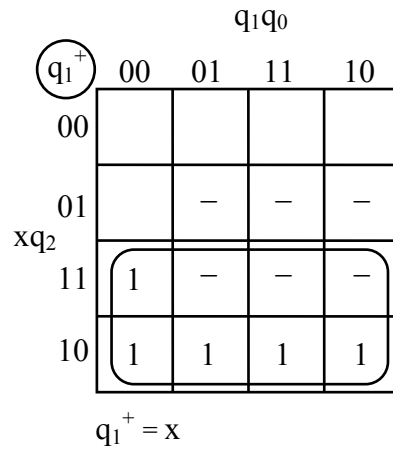
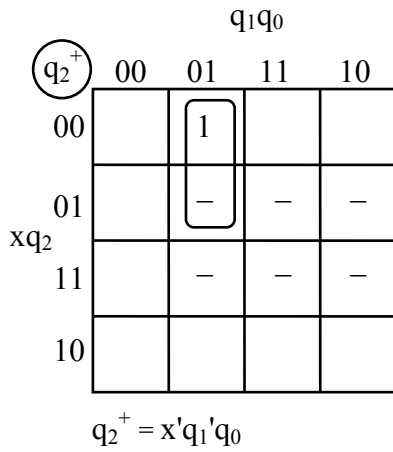
$$U = \lambda(Q, X)$$

(Jämför med analys av synkrona sekvensnät på sidan 2.)

Steg 4:

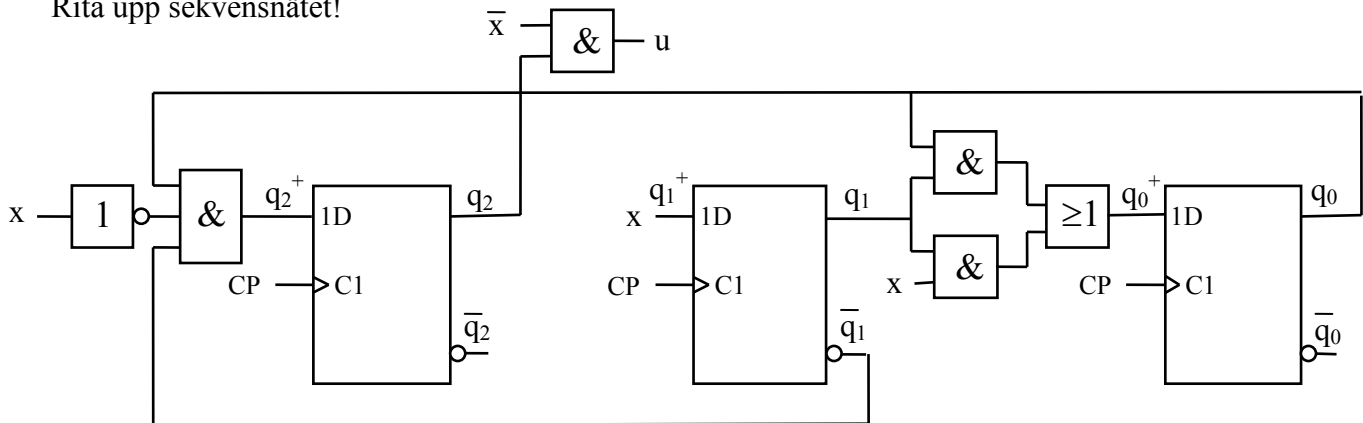
Rita nu Karnaughdiagram för q_2^+ , q_1^+ , och q_0^+ och u med hjälp av $\delta(\lambda)$ -tabellen.

Observera att endast 5 av 8 möjliga tillstånd används i $\delta(\lambda)$ -tabellen. De 3 oanvända tillstånden ger upphov till 6 st "don't care"-termer i Karnaughdiagrammen.



Steg 5:

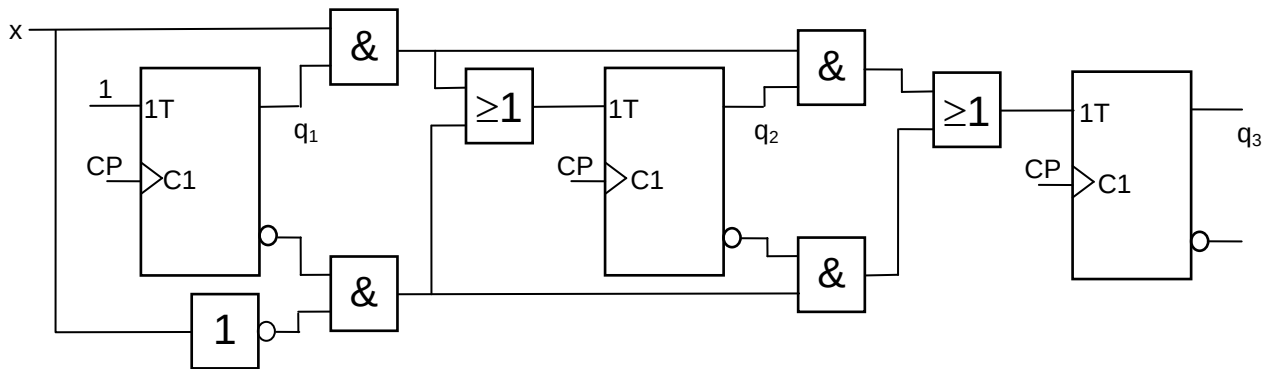
Rita upp sekvensnätet!



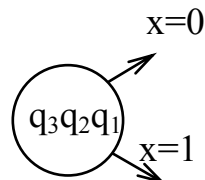
Extrauppgifter, synkrona sekvensnät.

Synkrona räknare. Analys.

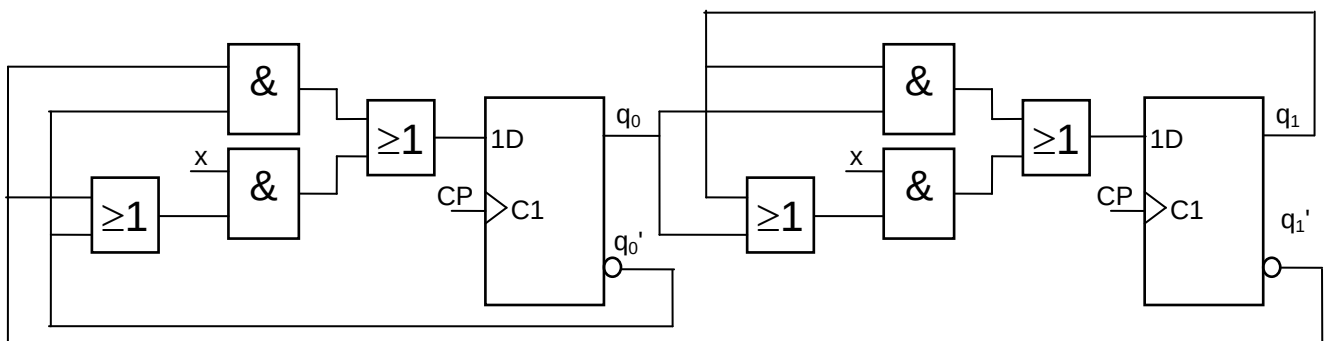
X.1 En räknare med räknevillkoret x visas nedan. Rita en tillståndsgraf för räknaren.



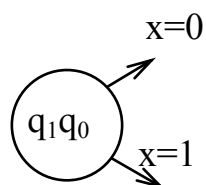
Rita tillstånden enligt:



X.2 En räknare med räknevillkoret x visas nedan. Rita en tillståndsgraf för räknaren.

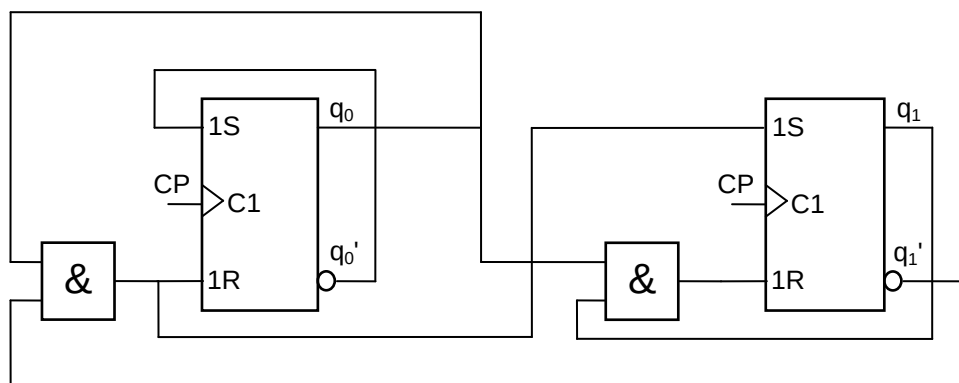


Rita tillstånden enligt:



Ext-7 (Ver 2013-04-14)

X.3 Rita en tillståndsgraf och rita pulsdigram för den autonoma räknaren nedan. Pulsdiagrammet skall utgå från tillståndet $Q = (q_1q_0)_2 = 0$

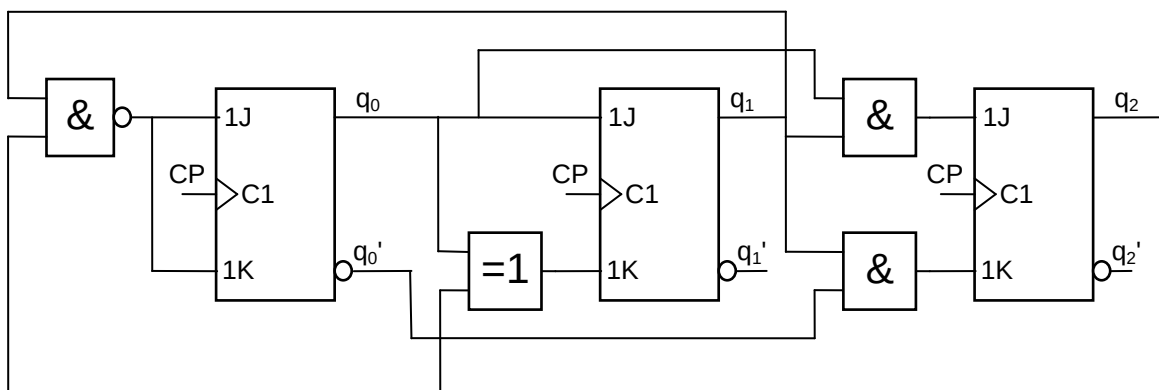


CP

q_0

q_1

X.4 Rita en tillståndsgraf och rita pulsdigram för den autonoma räknaren nedan. Pulsdiagrammet skall utgå från tillståndet $Q = (q_2q_1q_0)_2 = 0$



CP

q_0

q_1

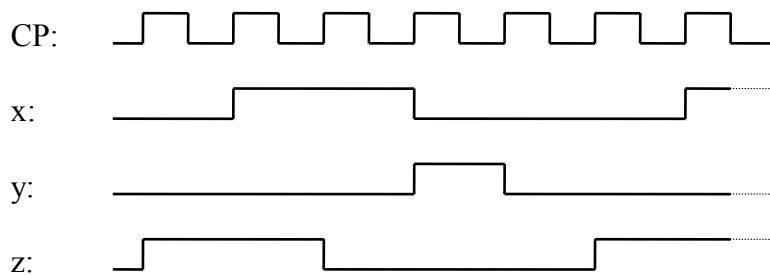
q_2

Synkrona räknare. Syntes.

Sekvensnät av denna typ måste starta från ett känt starttillstånd. Inga åtgärder för att försätta sekvensnäten i detta starttillstånd behöver vidtagas i uppgifterna nedan.

X.5 En autonom synkron räknare med räknesekvens enligt figuren nedan skall konstrueras. JK-vippor med positiv flanktriggning, NAND-grindar med valfritt antal ingångar samt INVERTERARE får användas. x, y och z i pulsdigrammen nedan är utsignaler (q) från var sin vippa.

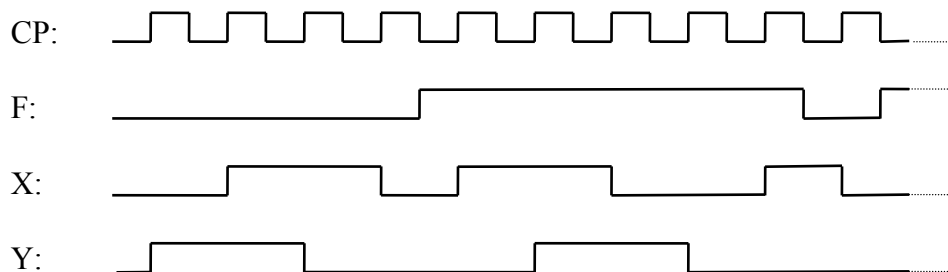
- Rita en tillståndsgraf enligt pulsdigrammen nedan som beskriver räknaren !
- Realisera räknaren!
- Komplettera grafen från a) så att räknarens samtliga tillstånd kommer med!



X.6 En sk stegmotor används för att styra en funktion på en maskin. Styrkretsen för stegmotorn är ett synkront sekvensnät, som förutom klocksignalen CP, har insignalen Fram/Back (F) samt utsignalerna X och Y.

Insignalen F är synkroniserad med klocksignalen CP.

Utsignalerna X och Y är kopplade till stegmotorns drivelektronik. Hur utsignalerna beror av klocksignalen CP och insignalen F framgår av figuren nedan.



X och Y ovan är utsignalen (q) från var sin vippa.

Konstruera sekvensnätet!

Två JK-vippor med positiv flanktriggning och NAND-grindar med valfritt antal ingångar samt INVERTERARE får användas.

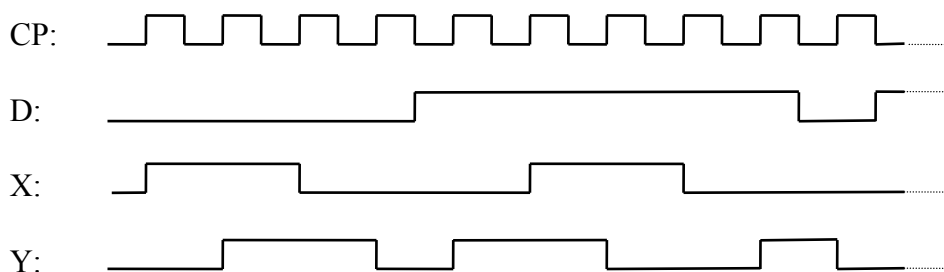
X.7 Konstruera en 2-bitars synkron räknare med insignalen x , som är synkroniserad med klocksignalen.

För $x = 0$ skall räknesekvensen (q_1q_0) vara 00, 01, 11, 00, ..

För $x = 1$ skall räknesekvensen (q_1q_0) vara 00, 11, 01, 00, ..

JK-vippor med trigging på positiv flank, NAND-grindar och INVERTERARE får användas.

X.8 En styrkrets för en stegmotor skall konstrueras. Styrkretsen skall utgöras av ett synkront sekvensnät, som förutom klocksignalen CP, har insignalen Direction (D) samt utsignalerna X och Y. Utsignalerna är kopplade till stegmotorns drivelektronik. Hur utsignalerna X och Y beror av klocksignalen CP och insignalen D framgår av figuren nedan.



X och Y ovan är utsignalen (q) från var sin vippa.

Konstruera sekvensnätet!

Två SR-vippor med positiv flanktrigging, NAND-grindar med valfritt antal ingångar, EXCLUSIVE-OR-grindar och INVERTERARE får användas.

X.9 En synkron tvåbitars (q_1q_0) uppräknare med ett räknevillkor x skall konstrueras. Då $x = 1$ skall räknaren räkna uppåt med vanlig binärkod. Då $x = 0$ skall räknaren stanna kvar i det tillstånd den befinner sig.

D-vippor med positiv flanktrigging, NOR-grindar med valfritt antal ingångar och INVERTERARE får användas.

a) Rita en tillståndsgraf som beskriver räknaren !

b) Realisera räknaren!

X.10 En synkron tvåbitars (q_1q_0) nedräknare med ett räknevillkor x skall konstrueras.

Då $x = 1$ skall räknaren räkna nedåt med vanlig binärkod, dvs sekvensen ..., 00, 11, 10, 01, 00, ...

Då $x = 0$ skall räknaren stanna kvar i det tillstånd den befinner sig.

T-vippor med positiv flanktrigging, NAND-grindar med valfritt antal ingångar och INVERTERARE får användas.

a) Rita en tillståndsgraf som beskriver räknaren !

b) Realisera räknaren!

Enkla synkrona sekvensnät. Syntes.

Sekvensnät av denna typ måste starta från ett känt starttillstånd. Inga åtgärder för att försätta sekvensnäten i detta starttillstånd behöver vidtagas i uppgifterna nedan.

- X.11** Ett synkront sekvensnät skall ha en insignal x och en utsignal u . Utsignalen u skall få värdet 1 för varje insignalsekvens som består av exakt en nolla följd av en etta. Med exakt en nolla menas att flera nollor följda av en etta inte skall ge utsignalen ett. Se exemplet nedan!

Exempel: $\sigma_x = 010010110000101 \dots$ (Insignalsekvens)

$\sigma_u = 010000100000001 \dots$ (Utsignalsekvens)

Utsignalen skall ges värdet ett när den sista biten i en korrekt insignalsekvens anländer på x -ingången och behålla värdet ett så länge x har kvar sitt värde under detta klockpulsintervall.

- Rita en tillståndsgraf för sekvensnätet.
- Konstruera sekvensnätet.

JK-vippor som triggar på positiv flank, NAND-grindar med valfritt antal ingångar och INVERTERARE får användas.

- X.12** Ett synkront sekvensnät med en insignal x , och en utsignal u skall konstrueras. Insignalen är synkroniserad med klocksignalen.

Utsignalen u skall få värdet ett varje gång två x -värden i rad har varit olika. Ettan på utgången skall uppträda i samma klockpulsintervall som det andra x -värdet på ingången. Se exemplet nedan!

Exempel: $\sigma_x = 001100011011100111100010110000 \dots$

$\sigma_u = 001010010110010100010011101000 \dots$

- Rita en tillståndsgraf som beskriver sekvensnätet!
- Realisera sekvensnätet!

JK-vippor som triggar på positiv flank, NAND-grindar med valfritt antal ingångar och INVERTERARE får användas.

- X.13** Ett synkront sekvensnät med en insignal x (synkroniserad med klocksignalen) och en utsignal u skall konstrueras.

Utsignalen u skall få värdet ett varje gång exakt två x -värden i rad har varit lika. Ettan på utgången skall uppträda i det klockpulsintervall som följer efter intervallen med de två lika x -värdena på ingången. Se exemplet nedan!

Exempel: $\sigma_x = 001100011011100111111000010110000 \dots$

$\sigma_u = 00101000010000010000000000000001000 \dots$

- Rita en tillståndsgraf som beskriver sekvensnätet!
- Realisera sekvensnätet!

JK-vippor som triggar på positiv flank, NAND-grindar med valfritt antal ingångar och INVERTERARE får användas.

Ext-7 (Ver 2013-04-14)

X.14 Ett synkront sekvensnät med en insignal x (synkroniserad med klocksignalen) och en utsignal u skall konstrueras.

Utsignalen u skall få värdet ett varje gång exakt tre x -värden i rad har varit lika. Ettan på utgången skall uppträda när den sista biten i en korrekt sekvens anländer på ingången. Se exemplet nedan!

Rita en tillståndsgraf som beskriver sekvensnätet! Nätet behöver ej konstrueras! Det räcker med tillståndsgraf.

Exempel: $\sigma_x = 00100000011011110011111100001010000\dots$
 $\sigma_u = 00000100000000100000100000100000010\dots$

Lösningar till X-uppgifter.

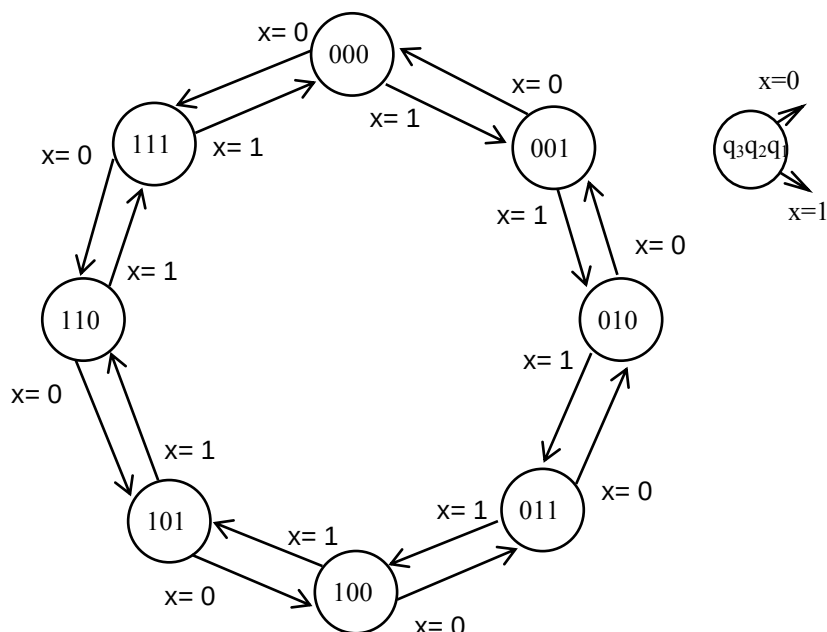
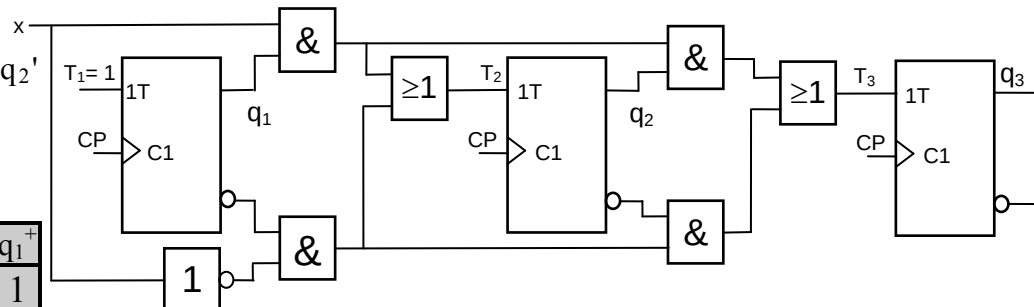
Synkrona sekvensnät

X.1 $T_3 = xq_1q_2 + x'q_1'q_2'$

$T_2 = xq_1 + x'q_1'$

$T_1 = 1$

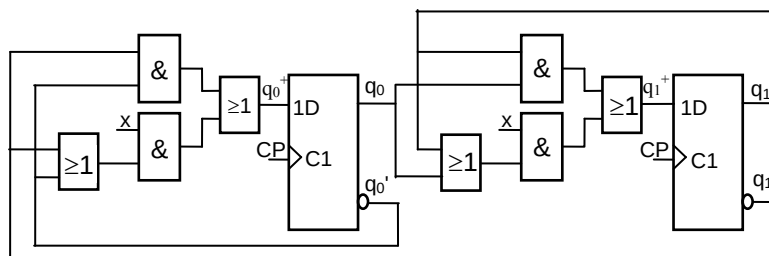
x	q ₃	q ₂	q ₁	T ₃	T ₂	T ₁	q ₃ ⁺	q ₂ ⁺	q ₁ ⁺
0	0	0	0	1	1	1	1	1	1
0	0	0	1	0	0	1	0	0	0
0	0	1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	0	1	0
0	1	0	0	1	1	1	0	1	1
0	1	0	1	0	0	1	1	0	0
0	1	1	0	0	1	1	1	0	1
0	1	1	1	0	0	1	1	1	0
1	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	1	0	1	0
1	0	1	0	0	0	1	0	1	1
1	0	1	1	1	1	1	1	0	0
1	1	0	0	0	0	1	1	0	1
1	1	0	1	0	1	1	1	1	0
1	1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	1	0	0	0



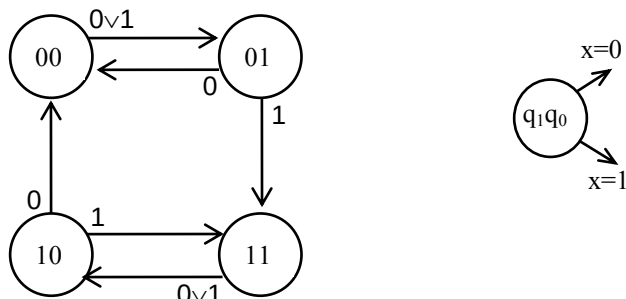
X.2 $q_0^+ = q_1'q_0' + xq_1' + xq_0'$

$q_1^+ = q_1q_0 + xq_1 + xq_0$

x	q ₁	q ₀	q ₁ ⁺	q ₀ ⁺
0	0	0	0	1
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	0



(0∨1 = 0 eller 1)



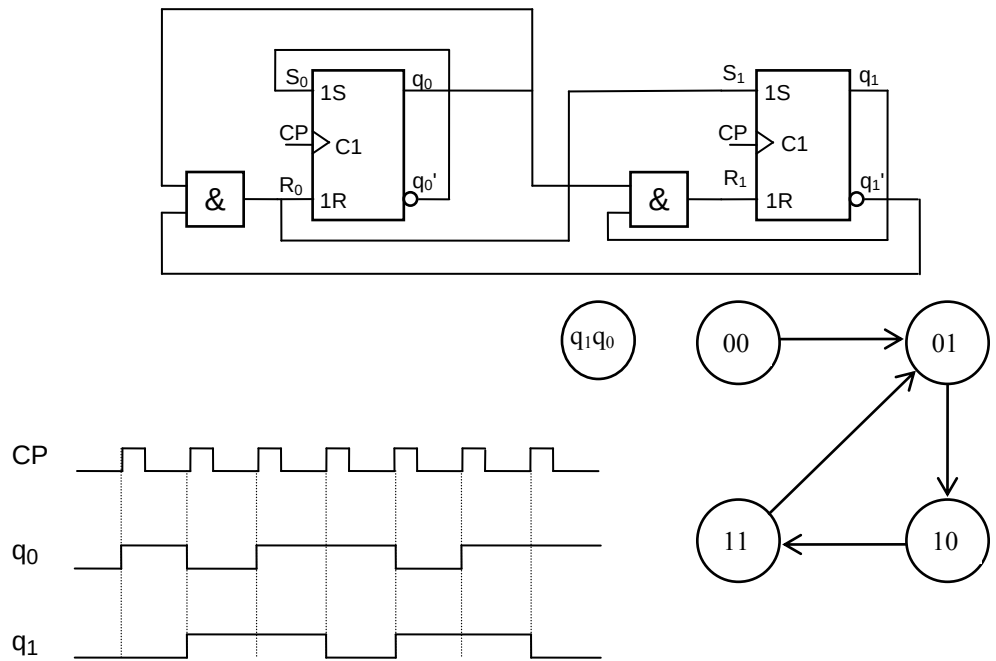
Ext-7 (Ver 2013-04-14)**X.3** $S_1 = q_1'q_0$

$$R_1 = q_1q_0$$

$$S_0 = q_0'$$

$$R_0 = q_1'q_0 (= S_1)$$

q_1	q_0	S_1	R_1	S_0	R_0	q_1^+	q_0^+
0	0	0	0	1	0	0	1
0	1	1	0	0	1	1	0
1	0	0	0	1	0	1	1
1	1	0	1	0	0	0	1

**X.4** $J_2 = q_1q_0$

$$K_2 = q_1q_0'$$

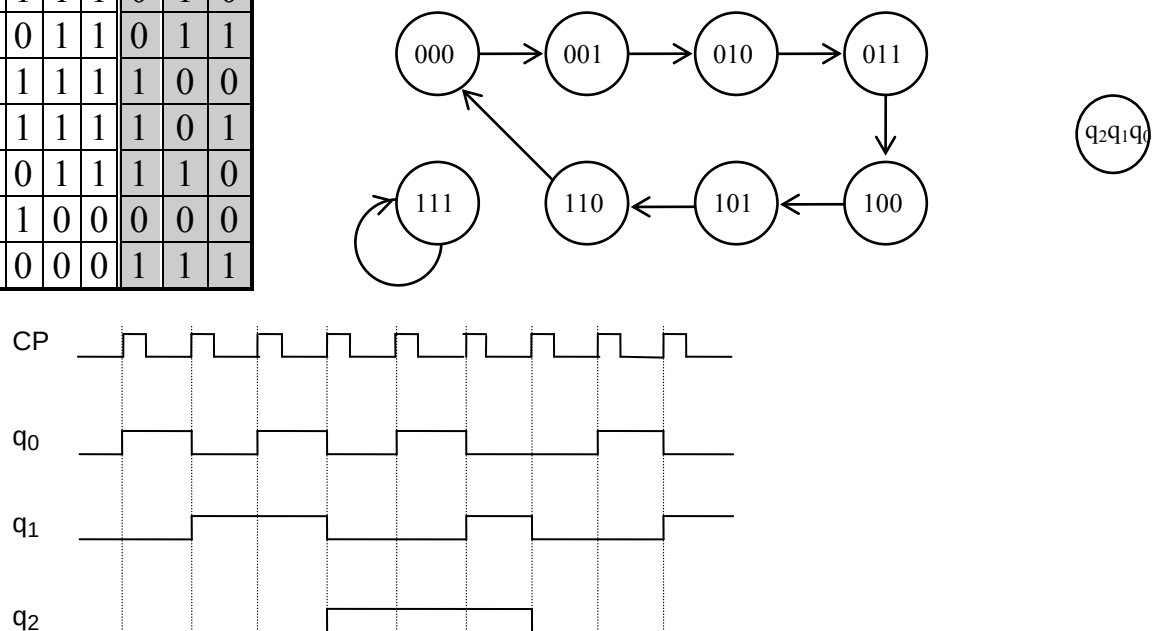
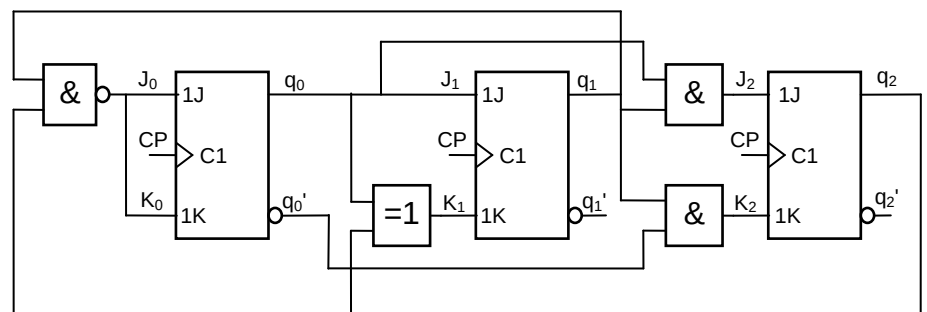
$$J_1 = q_0$$

$$K_1 = q_2 \oplus q_0$$

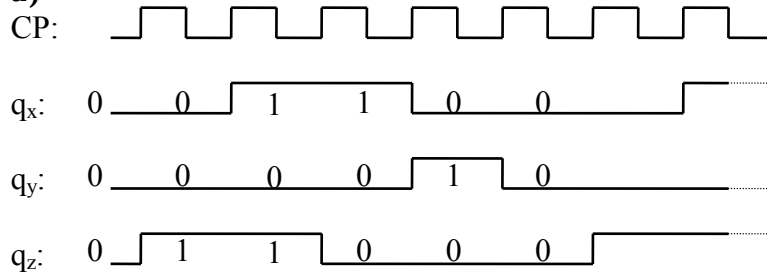
$$J_0 = q_2' + q_1'$$

$$K_0 = J_0$$

q_2	q_1	q_0	J_2	K_2	J_1	K_1	J_0	K_0	q_2^+	q_1^+	q_0^+
0	0	0	0	0	0	0	1	1	0	0	1
0	0	1	0	0	1	1	1	1	0	1	0
0	1	0	0	1	0	0	1	1	0	1	1
0	1	1	1	0	1	1	1	1	1	0	0
1	0	0	0	0	0	1	1	1	1	0	1
1	0	1	0	0	1	0	1	1	1	1	0
1	1	0	0	1	0	1	0	0	0	0	0
1	1	1	1	0	1	0	0	0	1	1	1



X.5 a)



q_x	q_y	q_z	q_x^+	q_y^+	q_z^+	J_x	K_x	J_y	K_y	J_z	K_z
0	0	0	0	0	1	0	-	0	-	1	-
0	0	1	1	0	1	1	-	0	-	-	0
0	1	0	0	0	0	0	-	-	1	0	-
0	1	1	-	-	-	-	-	-	-	-	-
1	0	0	0	1	0	-	1	1	-	0	-
1	0	1	1	0	0	-	0	0	-	-	1
1	1	0	-	-	-	-	-	-	-	-	-
1	1	1	-	-	-	-	-	-	-	-	-

	$q_y q_z$			
J_x	00	01	11	10
0	0	1	-	0
q_x	-	-	-	-

$$J_x = q_z$$

	$q_y q_z$			
K_x	00	01	11	10
0	-	-	-	-
1	1	0	-	-

$$K_x = q_z'$$

	$q_y q_z$			
J_y	00	01	11	10
0	0	0	-	-
1	1	0	-	-

$$J_y = q_x q_z'$$

	$q_y q_z$			
K_y	00	01	11	10
0	-	-	-	1
1	-	-	-	-

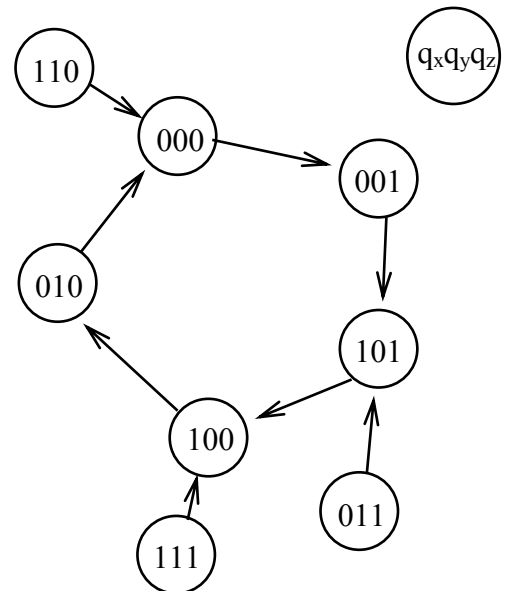
$$K_y = 1$$

	$q_y q_z$			
J_z	00	01	11	10
0	1	-	-	0
1	0	-	-	-

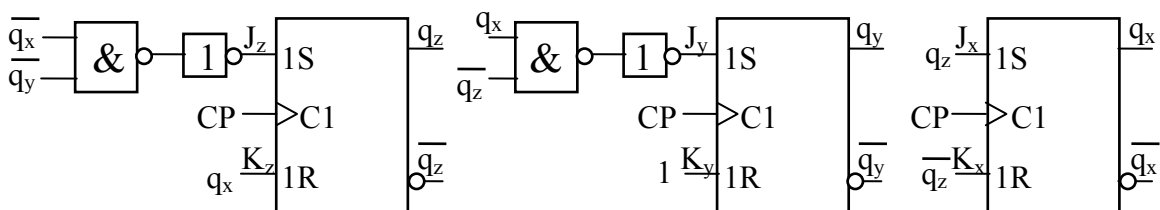
$$J_z = q_x' q_y'$$

	$q_y q_z$			
K_z	00	01	11	10
0	-	0	-	-
1	-	1	-	-

$$K_z = q_x$$



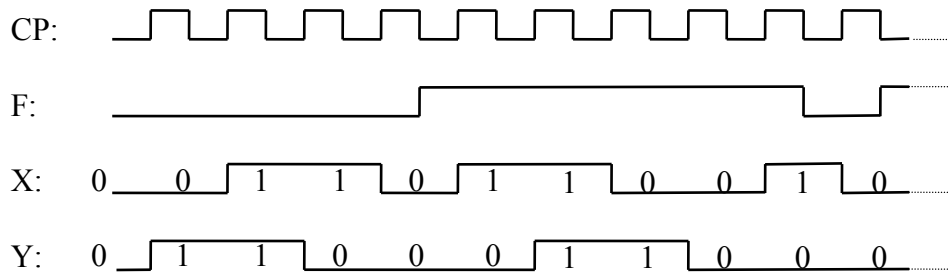
b)



c)

Inritade i graf ovan

q_x	q_y	q_z	q_x^+	q_y^+	q_z^+	J_x	K_x	J_y	K_y	J_z	K_z
0	0	0	0	0	1	0	1	0	1	1	0
0	0	1	1	0	1	1	0	0	1	1	0
0	1	0	0	0	0	0	1	0	1	0	0
0	1	1	1	0	1	1	0	0	1	0	0
1	0	0	0	1	0	0	1	1	1	0	1
1	0	1	1	0	0	1	0	0	1	0	1
1	1	0	0	0	0	0	1	1	1	0	1
1	1	1	1	0	0	1	0	0	1	0	1



F	q _X	q _Y	q _X ⁺	q _Y ⁺	J _X	K _X	J _Y	K _Y
0	0	0	0	1	0	-	1	-
0	0	1	1	1	1	-	-	0
0	1	0	0	0	-	1	0	-
0	1	1	1	0	-	0	-	1
1	0	0	1	0	1	-	0	-
1	0	1	0	0	0	-	-	1
1	1	0	1	1	-	0	1	-
1	1	1	0	1	-	1	-	0

q_Xq_Y

J _X	00	01	11	10
F	0	1	-	-
	1	1	0	-

$$J_X = F'q_Y + F q_Y' = F \oplus q_Y$$

K _X	00	01	11	10
F	0	-	-	1
	1	-	1	0

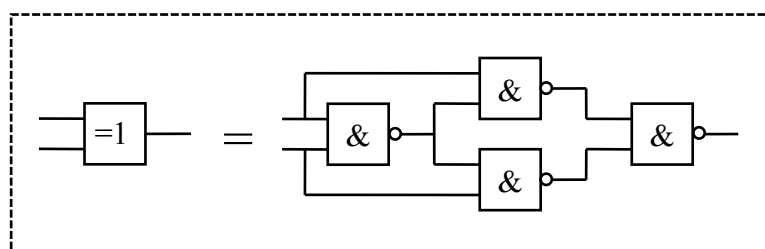
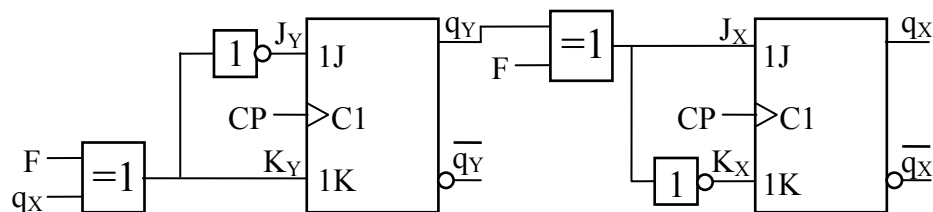
$$K_X = F'q_Y' + F q_Y = (F \oplus q_Y)'$$

J _Y	00	01	11	10
F	0	1	-	0
	1	0	-	1

$$J_Y = F'q_X' + F q_X = (F \oplus q_X)'$$

K _Y	00	01	11	10
F	0	-	1	-
	1	-	1	0

$$K_Y = F'q_X + F q_X' = F \oplus q_X$$



X.7

x	q ₁	q ₀	q ₁ ⁺	q ₀ ⁺	J ₁	K ₁	J ₀	K ₀
0	0	0	0	1	0	-	1	-
0	0	1	1	1	1	-	-	0
0	1	0	-	-	-	-	-	-
0	1	1	0	0	-	1	-	1
1	0	0	1	1	1	-	1	-
1	0	1	0	0	0	-	-	1
1	1	0	-	-	-	-	-	-
1	1	1	0	1	-	1	-	0

q₁q₀

J ₁	00	01	11	10
0	0	1	-	-
1	1	0	-	-

$J_1 = x'q_0 + xq_0' = x \oplus q_0$

K ₁	00	01	11	10
0	-	-	1	-
1	-	-	1	-

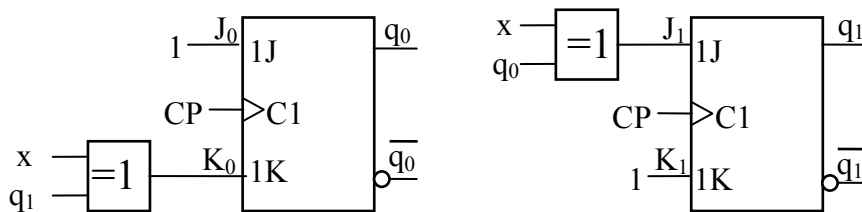
$K_1 = 1$

J ₀	00	01	11	10
0	1	-	-	-
1	1	-	-	-

$J_0 = 1$

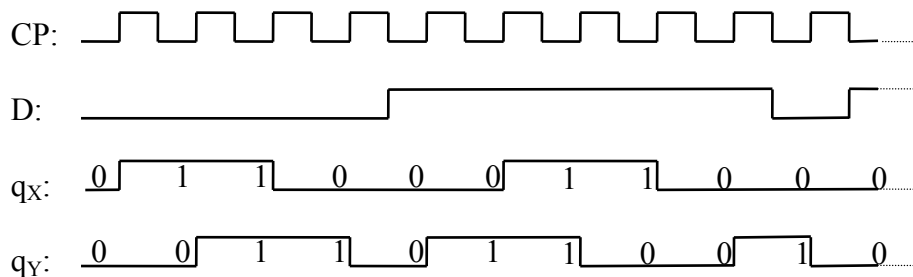
K ₀	00	01	11	10
0	-	0	1	-
1	-	1	0	-

$K_0 = x'q_1 + xq_1' = x \oplus q_1$



(Realisering av XOR-funktionen med NAND-grindar enligt uppgift X.6)

X.8



D	q _X	q _Y	q _X ⁺	q _Y ⁺	S _X	R _X	S _Y	R _Y
0	0	0	1	0	1	0	0	-
0	0	1	0	0	0	-	0	1
0	1	0	1	1	-	0	1	0
0	1	1	0	1	0	1	-	0
1	0	0	0	1	0	-	1	0
1	0	1	1	1	1	0	-	0
1	1	0	0	0	0	1	0	-
1	1	1	1	0	-	0	0	1

q_Xq_Y

S _X	00	01	11	10
0	1	0	0	-
1	0	1	-	0

$S_X = D'q_Y' + Dq_Y = (D \oplus q_Y)'$

R _X	00	01	11	10
0	0	-	1	0
1	-	0	0	1

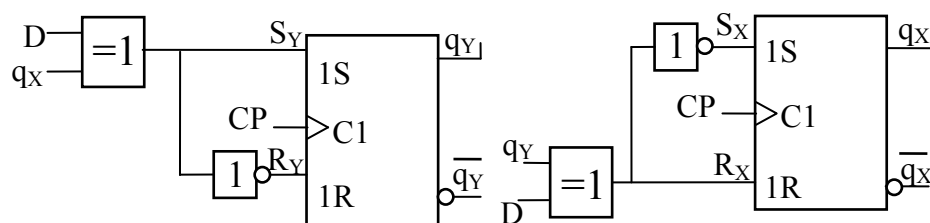
$R_X = D'q_Y + Dq_Y' = D \oplus q_Y$

S _Y	00	01	11	10
0	0	0	-	1
1	1	-	0	0

$S_Y = D'q_X + Dq_X' = D \oplus q_X$

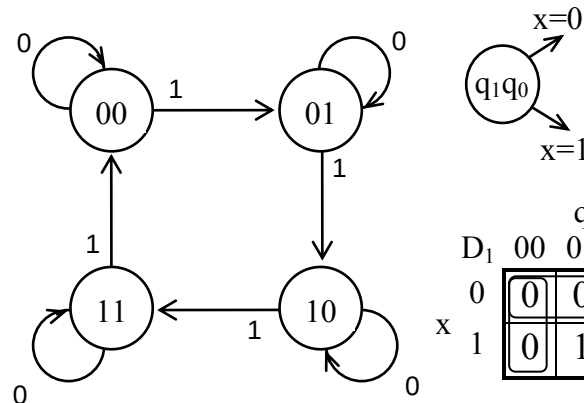
R _Y	00	01	11	10
0	-	1	0	0
1	0	0	1	-

$R_Y = D'q_X' + Dq_X = (D \oplus q_X)'$



X.9

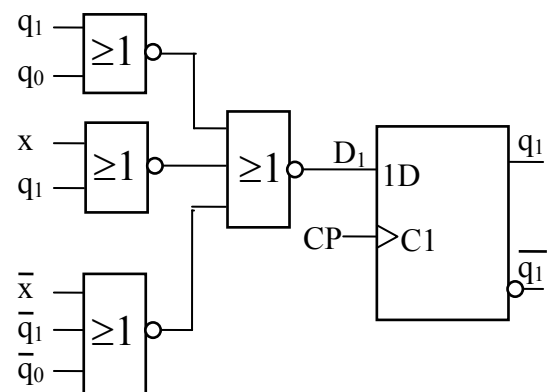
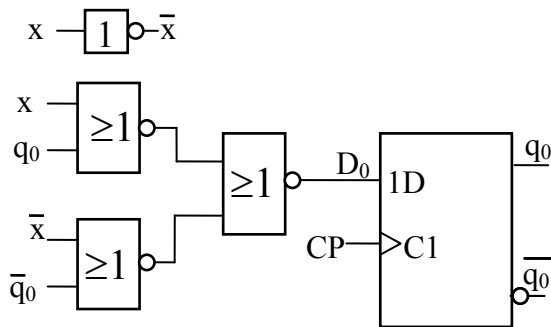
x	q ₁	q ₀	D ₁ =q ₁ ⁺	D ₀ =q ₀ ⁺
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0



	q ₁ q ₀	00	01	11	10
D ₁	0	0	0	1	1
	1	0	1	0	1

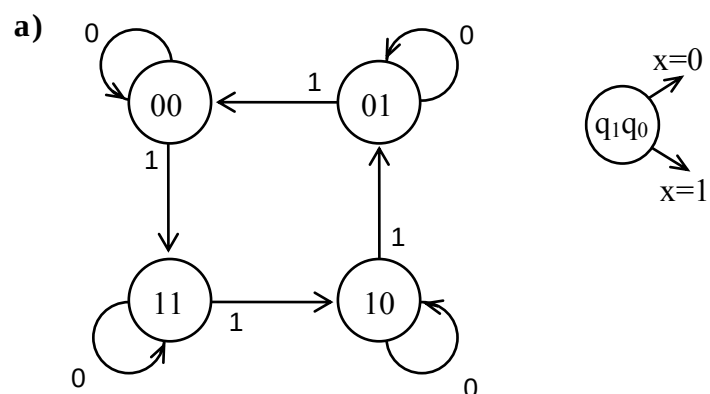
	q ₁ q ₀	00	01	11	10
D ₀	0	0	1	1	0
	1	1	0	0	1

$$D_1 = (q_1 + q_0)(x + q_1)(x' + q_1' + q_0') \quad D_0 = (x + q_0)(x' + q_0')$$



X.10

x	q ₁	q ₀	q ₁ ⁺	q ₀ ⁺	T ₁	T ₀
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	1	0	0	0
0	1	1	1	1	0	0
1	0	0	1	1	1	1
1	0	1	0	0	0	1
1	1	0	0	1	1	1
1	1	1	1	0	0	1



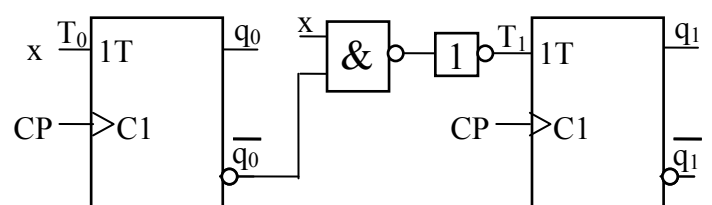
b)

	q ₁ q ₀	00	01	11	10
T ₁	0	0	0	0	0
	1	1	0	0	1

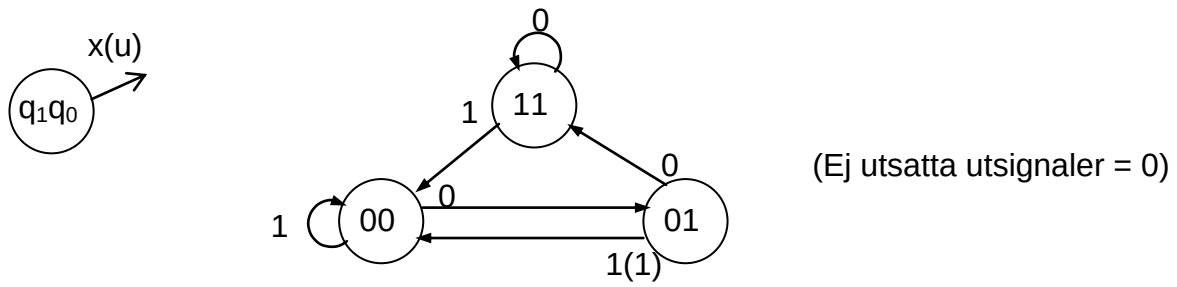
$$T_1 = x q_0'$$

	q ₁ q ₀	00	01	11	10
T ₀	0	0	0	0	0
	1	1	1	1	1

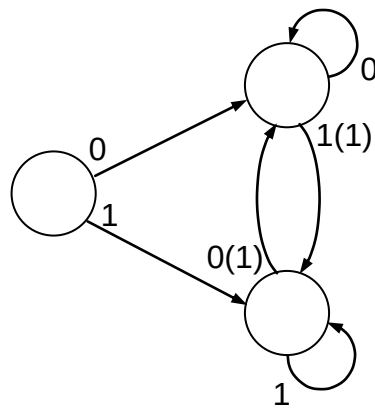
$$T_0 = x$$



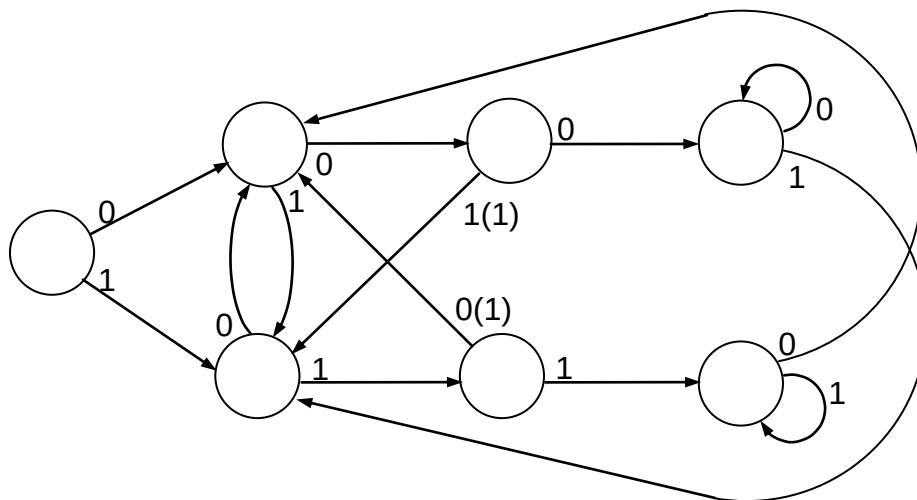
X.11



X.12



X.13



X.14

