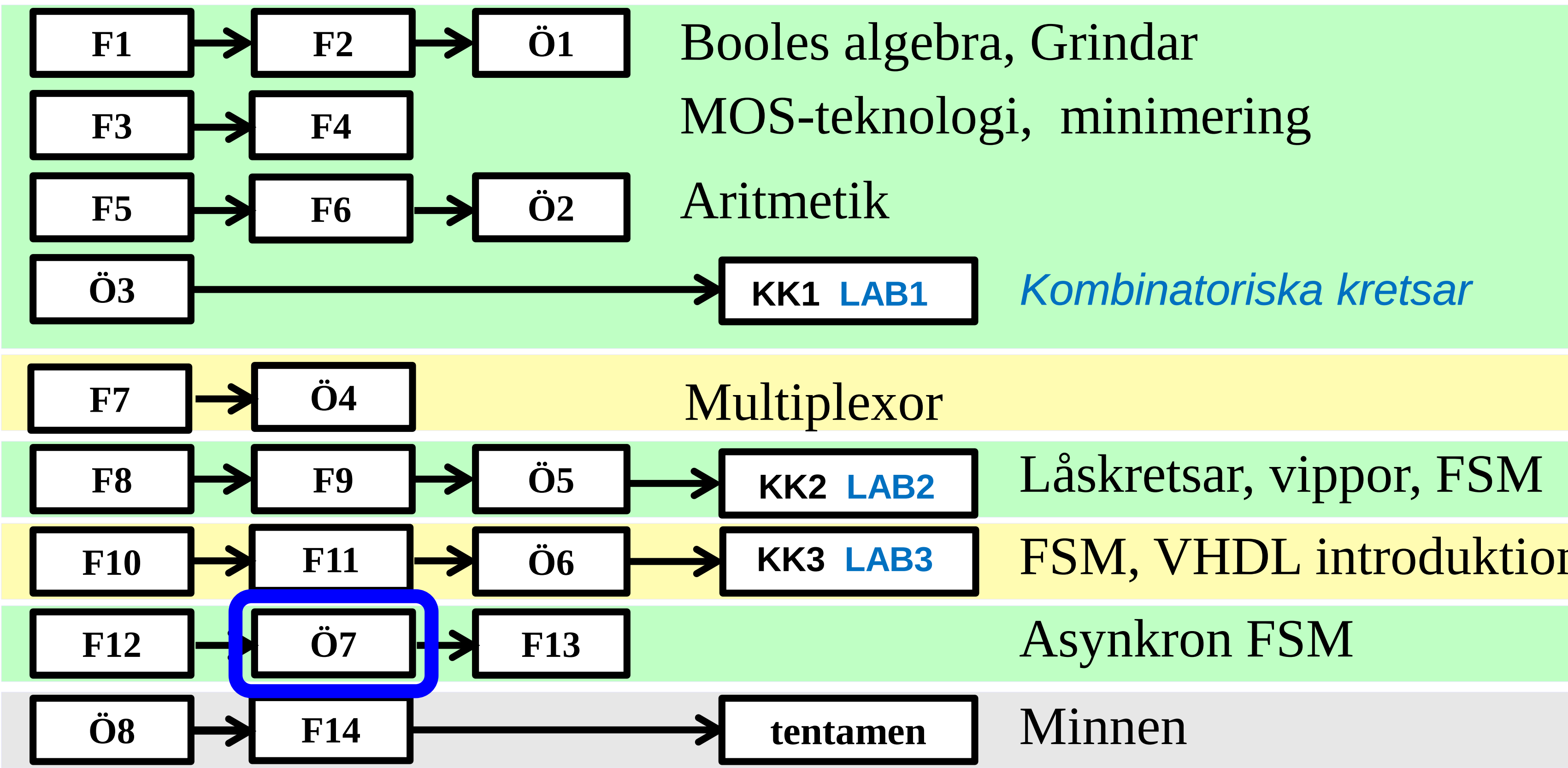
IE1204 Digital Design



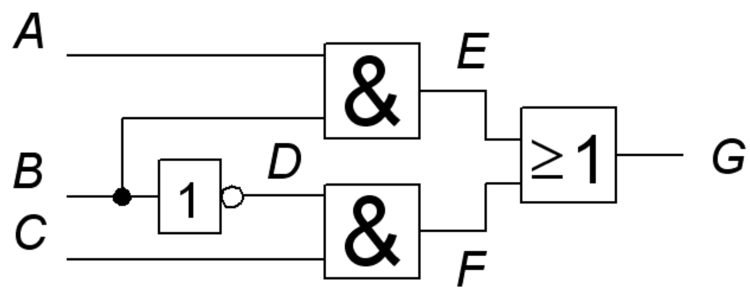
Föreläsningar och övningar bygger på varandra! Ta alltid igen det Du missat!
Läs på i förväg – delta i undervisningen – arbeta igenom materialet efteråt!

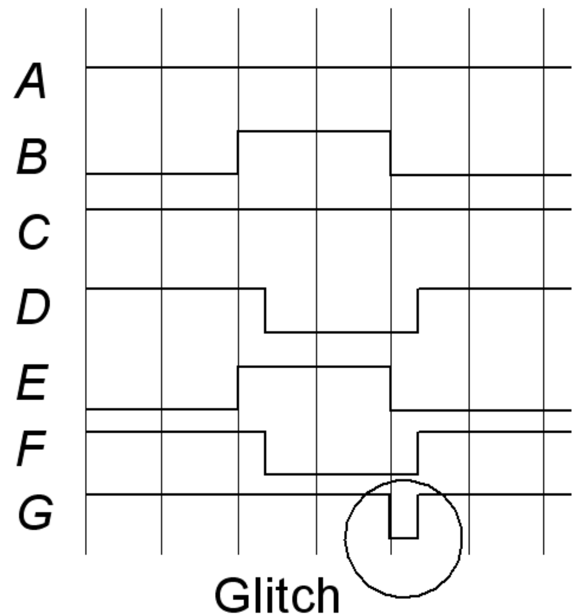
William Sandqvist william@kth.se

ÖH **11.1** ”Glitchar”

Om signaler passerar olika många grindsteg på vägen mot utgången kan kortvariga oönskade avvikelser från sanningstabellen uppkomma, så kallade ”glitchar”.

Visa i Karnaughdiagrammet hur man undviker dessa.





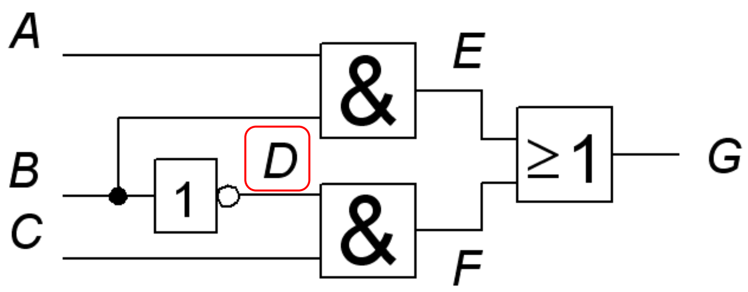
(*i figuren visas bara fördröjningen i inverteraren – övriga grindfördröjningar som inte påverkar ”glitchen” har inte tagits med*)

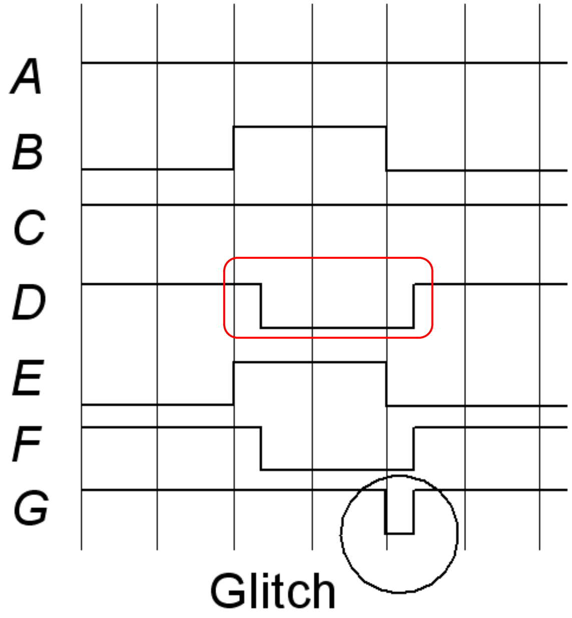
William Sandqvist william@kth.se

ÖH **11.1** ”Glitchar”

Om signaler passerar olika många grindsteg på vägen mot utgången kan kortvariga oönskade avvikelser från sanningstabellen uppkomma, så kallade ”glitchar”.

Visa i Karnaughdiagrammet hur man undviker dessa.

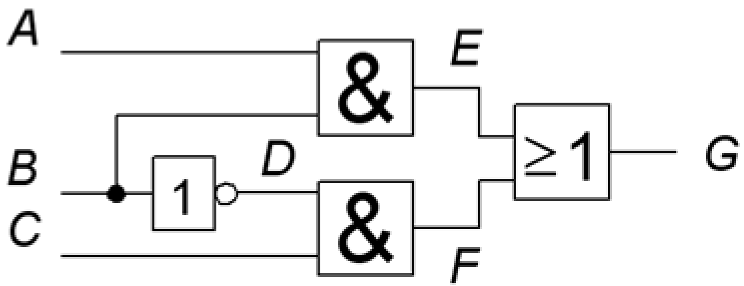


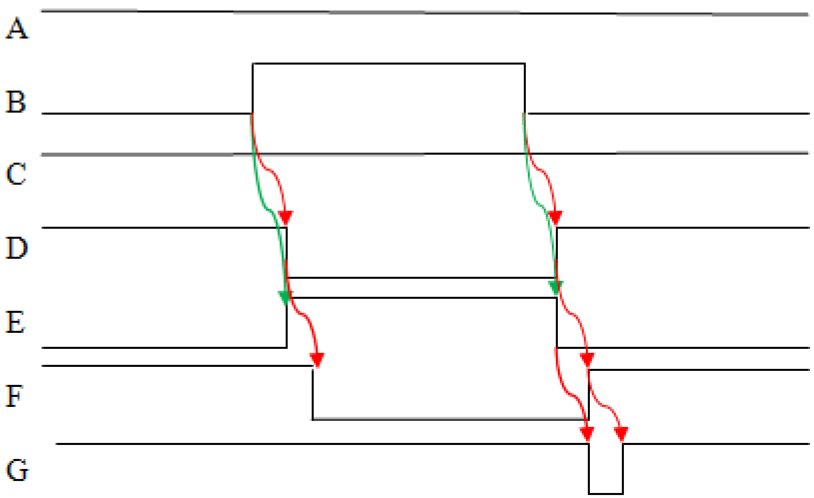


Signalen D är fördröjd i förhållande till A B C.

(*i figuren visas bara fördröjningen i inverteraren – övriga grindfördröjningar som inte påverkar ”glitchen” har inte tagits med*)

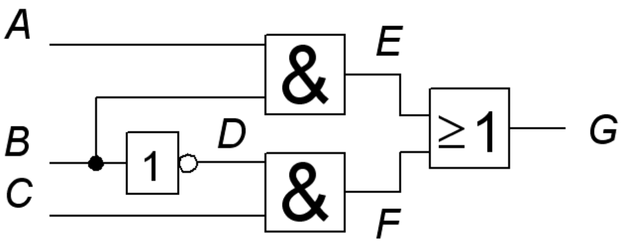
(med alla grindfördröjningar)





(Jan Andersson)

11.1

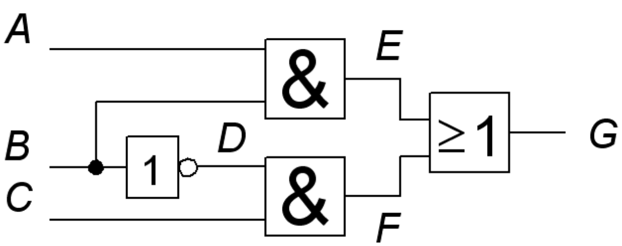


Nätet i ett Karnaughdiagram:

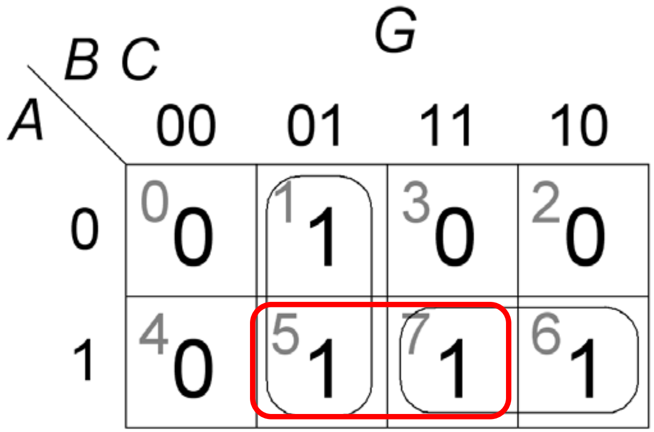
G

A \ BC	00	01	11	10
0	^{0} 0	^{1} 1	^{3} 0	^{2} 0
1	^{4} 0	^{5} 1	^{7} 1	^{6} 1

11.1

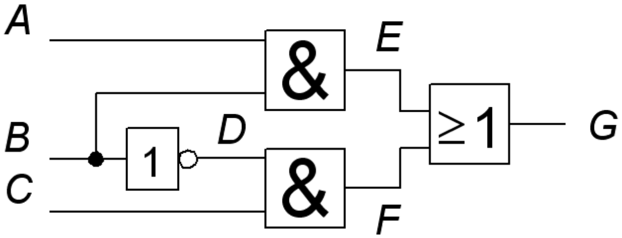


Nätet i ett Karnaughdiagram:

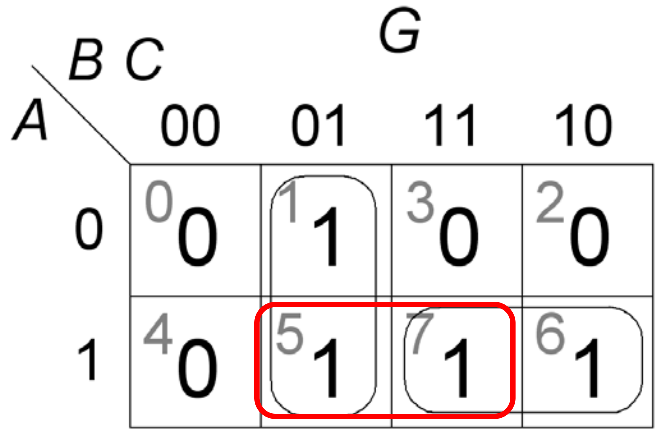


Se till att Karnaughdiagrammets hoptagningar bildar en sammanhängande ”kontinent” – inga öar! (Man tar med konsensus-termer så att man får funktionen på fullständig primimplikator form).

11.1



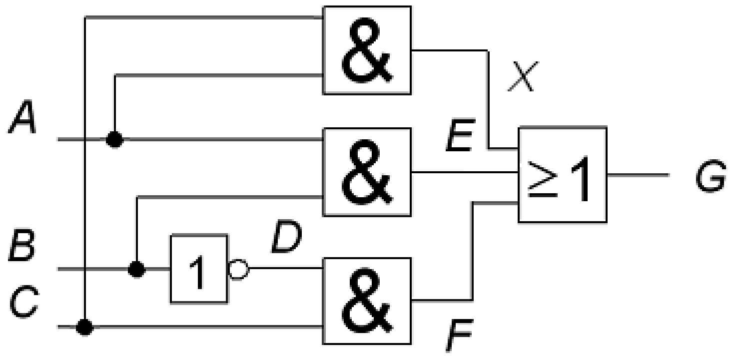
Nätet i ett Karnaughdiagram:



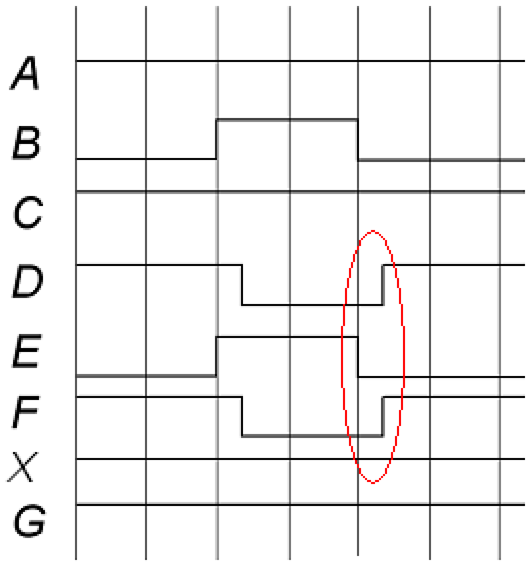
Se till att Karnaughdiagrammets hoptagningar bildar en sammanhängande ”kontinent” – inga öar! (Man tar med konsensus-termer så att man får funktionen på fullständig primimplikator form).

$$G = \overline{B}C + AB \quad \{Hazardfritt\} \quad G = \overline{B}C + AB + \boxed{AC}$$

11.1

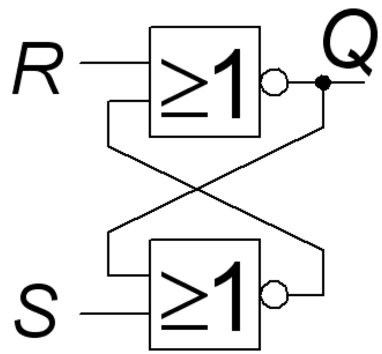


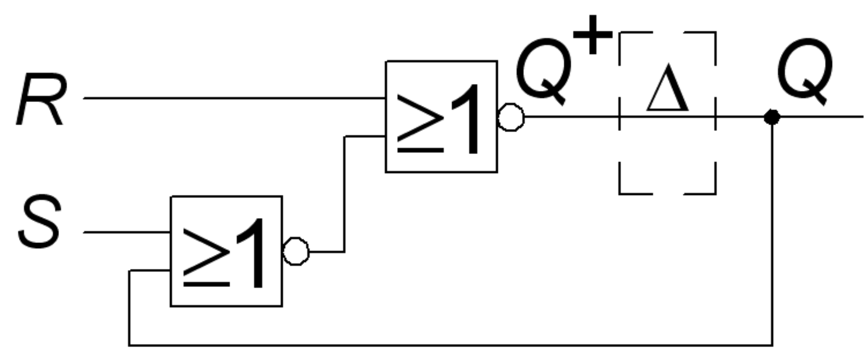
Vi ser att signalen *X* ”täcker upp” då det är risk för en ”glitch”, till priset av ett mer komplext nät!



ÖH **11.2** SR Asynkront sekvensnät

SR-låskretsen är ett asynkront sekvensnät.

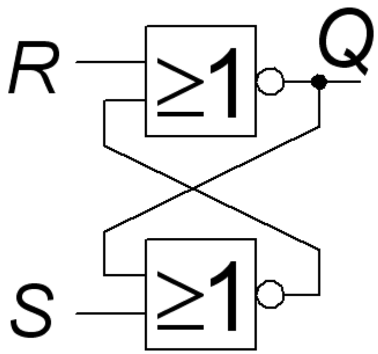


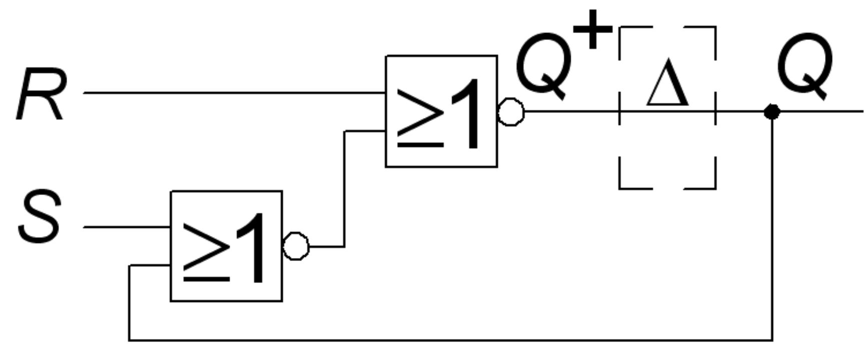


Alla grindfördröjningar som finns i nätet tänks placerade i symbolen Δ som får en liknande funktion som D-vippan i ett synkront sekvensnät.

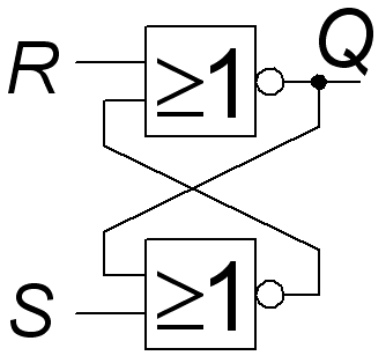
William Sandqvist william@kth.se

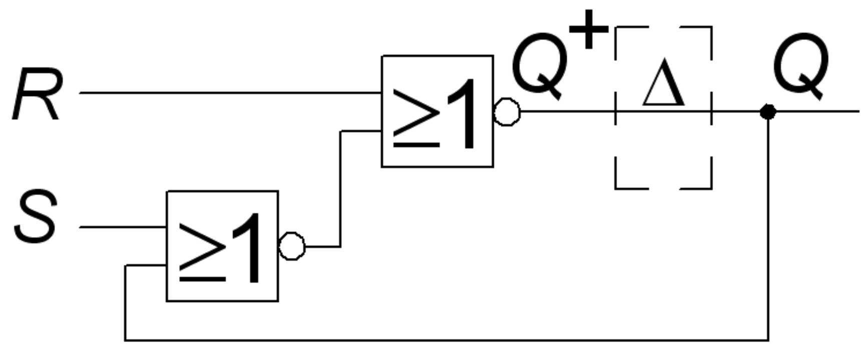
SR Analys:





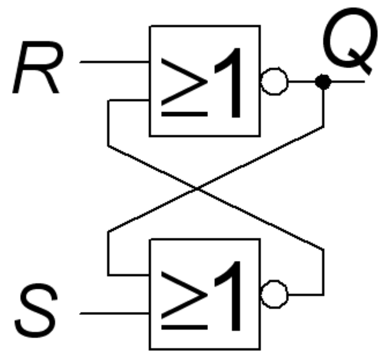
SR Analys:

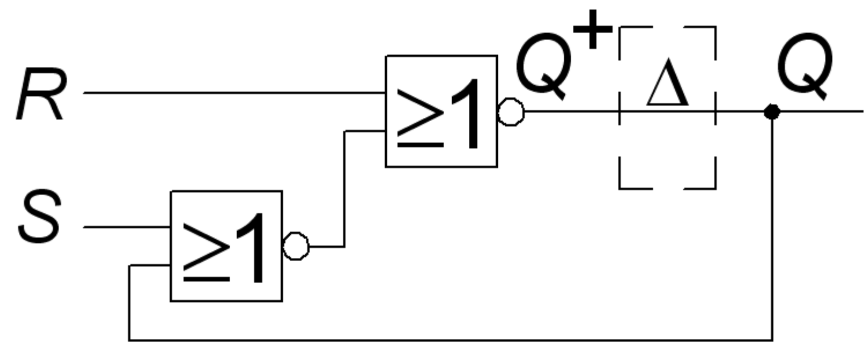




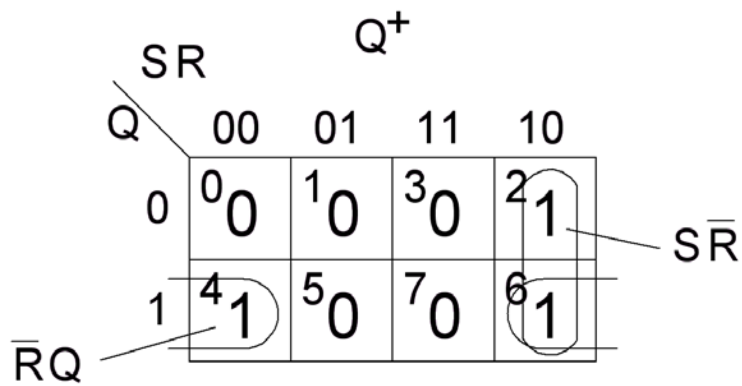
$$Q^{+} = \overline{R + \overline{S + Q}} = \overline{R}\cdot\overline{\overline{(S+Q)}} = \overline{R}\cdot(S+Q) = S\overline{R} + \overline{R}Q$$

SR Analys:



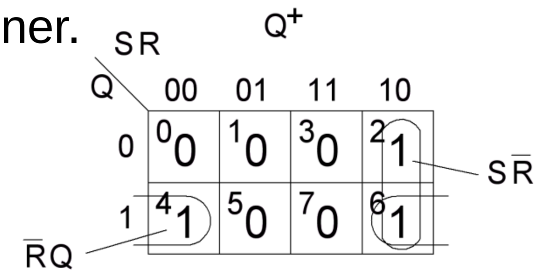


$$Q^{+} = \overline{R + \overline{S + Q}} = \overline{R} \cdot \overline{\overline{(S + Q)}} = \overline{R} \cdot (S + Q) = S\overline{R} + \overline{R}Q$$



SR Kodad tillståndstabell

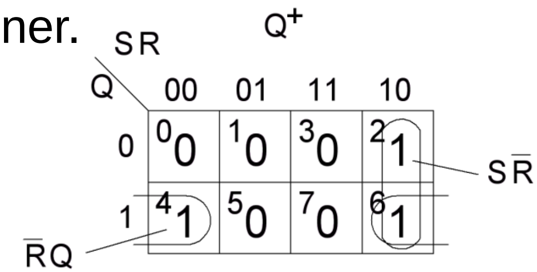
Den kodade tillståndstabellen brukar kallas för **excitationstabell** när man arbetar med asynkrona tillståndsmaskiner.



Nuvarande tillstånd Q	**Nästa tillstånd Q^+**			
	Insignaler SR			
	00	01	11	10
0	0	0	0	1
1	1	0	0	1

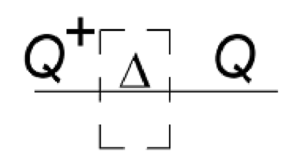
SR Kodad tillståndstabell

Den kodade tillståndstabellen brukar kallas för **excitationstabell** när man arbetar med asynkrona tillståndsmaskiner.



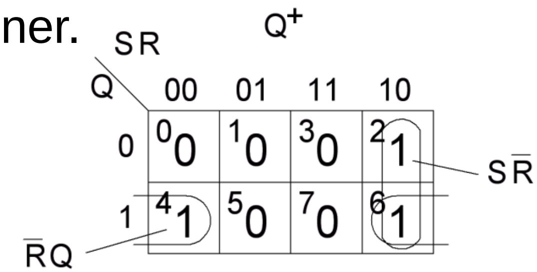
Nuvarande tillstånd Q	Nästa tillstånd Q^+			
	Insignaler SR			
	00	01	11	10
0	0	0	0	1
1	1	0	0	1

För varje insignal (kolumn) måste det finnas åtminstone något tillstånd där Q = Q^+. Sådana tillstånd är *stabila* och de brukar markeras genom att ringas in.



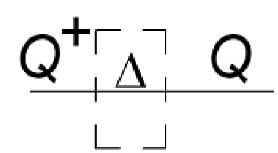
SR Kodad tillståndstabell

Den kodade tillståndstabellen brukar kallas för **excitationstabell** när man arbetar med asynkrona tillståndsmaskiner.



Nuvarande tillstånd Q	Nästa tillstånd Q^+			
	Insignaler SR			
	00	01	11	10
0	0	0	0	1
1	1	0	0	1

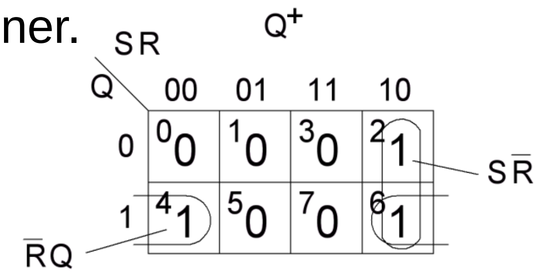
För varje insignal (kolumn) måste det finnas åtminstone något tillstånd där $Q = Q^+$. Sådana tillstånd är *stabila* och de brukar markeras genom att ringas in.



William Sandqvist william@kth.se

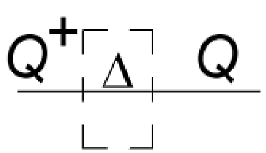
SR Kodad tillståndstabell

Den kodade tillståndstabellen brukar kallas för **excitationstabell** när man arbetar med asynkrona tillståndsmaskiner.



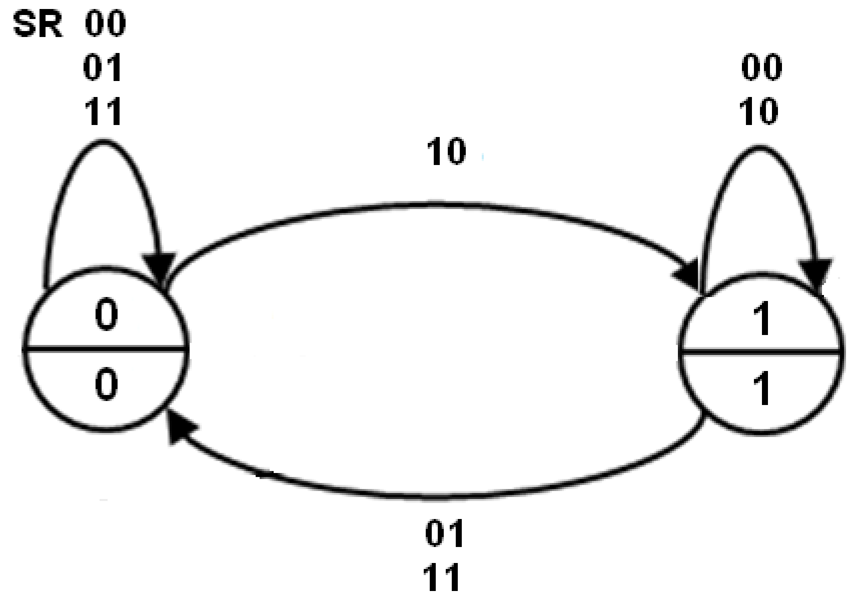
Nuvarande tillstånd Q	Nästa tillstånd Q^+			
	Insignaler SR			
	00	01	11	10
0	0	0	0	1
1	1	0	0	1

För varje insignal (kolumn) måste det finnas åtminstone något tillstånd där Q = Q^+. Sådana tillstånd är *stabila* och de brukar markeras genom att ringas in.



SR Tillståndsdiagram

Nuvarande tillstånd Q	**Nästa tillstånd Q^+**			
	Insignaler SR			
	00	01	11	10
0	0	0	0	1
1	1	0	0	1

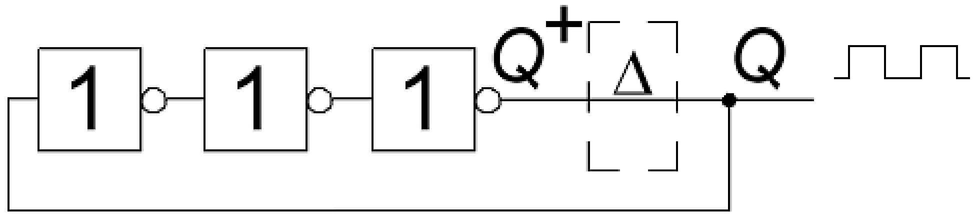


SR Tillståndstabell

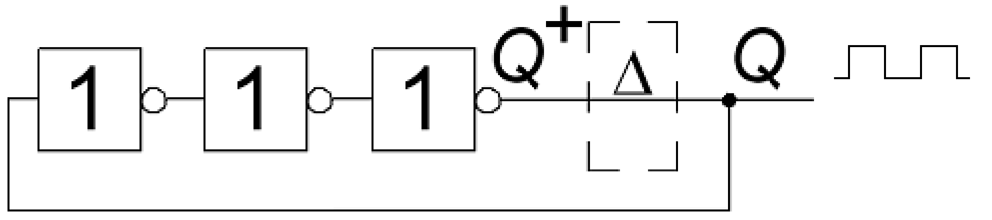
Tillståndstabellen brukar kallas för **flödestabell** när man arbetar med asynkrona tillståndsmaskiner.

Nuvarande tillstånd Q	**Nästa tillstånd Q^+**			
	Insignaler SR			
	00	01	11	10
A	A	A	A	B
B	B	A	A	B

ÖH **11.3** Oscillator?

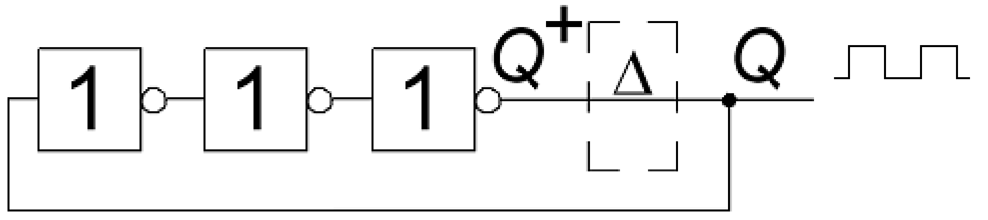


ÖH **11.3** Oscillator?



$$Q^{+} = \overline{Q}$$

ÖH **11.3** Oscillator?

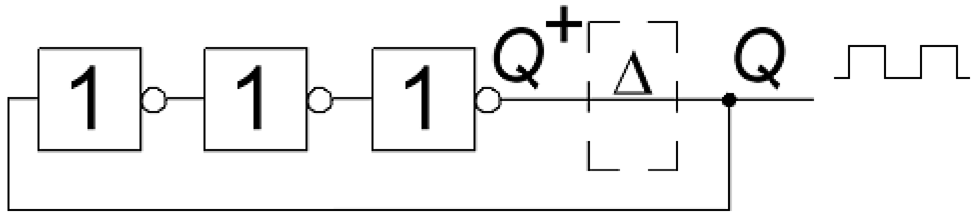


$$Q^{+} = \overline{Q}$$

Q	Q^{+}
0	1
1	0

Inga stabila tillstånd!

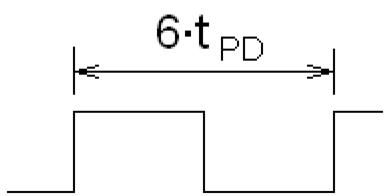
ÖH **11.3** Oscillator?



$$Q^{+} = \overline{Q}$$

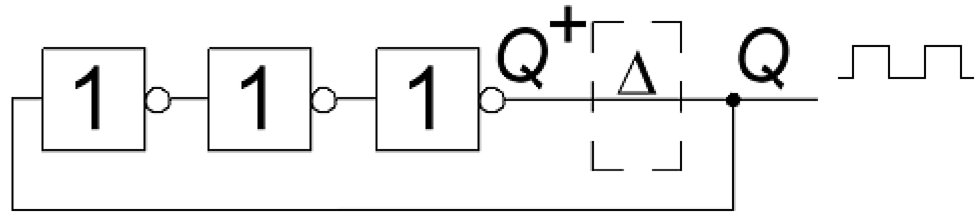
Q	Q^{+}
0	1
1	0

Inga stabila tillstånd!



$$T = 6 \cdot t_{PD} \quad \Rightarrow \quad f = \frac{1}{6 \cdot t_{PD}}$$

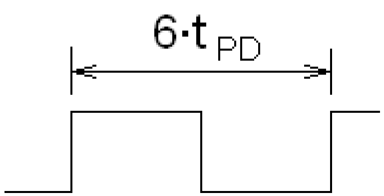
ÖH **11.3** Oscillator?



$$Q^+ = \overline{Q}$$

Q	Q^+
0	1
1	0

Inga stabila tillstånd!

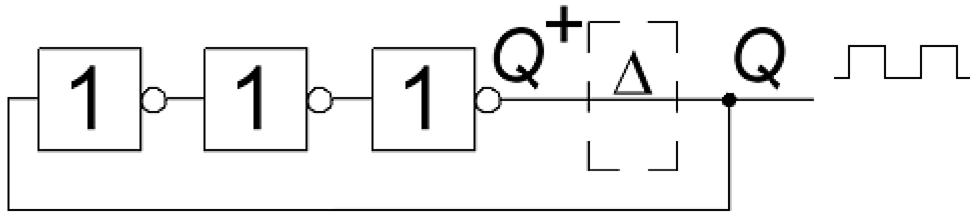


$$T = 6 \cdot t_{PD} \quad \Rightarrow \quad f = \frac{1}{6 \cdot t_{PD}}$$

Sifferexempel: $t_{pd} = 5 \cdot 10^{-9}$ $\quad f = \dfrac{1}{6 \cdot 5 \cdot 10^{-9}} = 33\,\mathrm{MHz}$

William Sandqvist william@kth.se

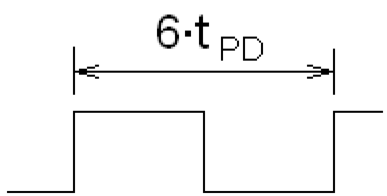
ÖH **11.3** Oscillator?



$$Q^{+} = \overline{Q}$$

Q	Q^{+}
0	1
1	0

Inga stabila tillstånd!



$$T = 6 \cdot t_{PD} \quad \Rightarrow \quad f = \frac{1}{6 \cdot t_{PD}}$$

Sifferexempel: $t_{pd} = 5 \cdot 10^{-9}$ $\quad f = \dfrac{1}{6 \cdot 5 \cdot 10^{-9}} = 33\,\mathrm{MHz}$

Kan användas för att indirekt mäta upp grindfördröjningen för logikkretsar.

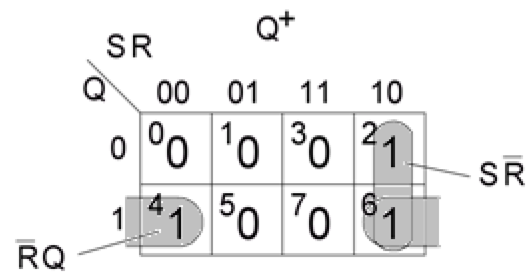
Speciellt för asynkrona nät

- Tillstånden måste kodas Kapplöpningsfritt (tex. Graykod).

SR-låskretsen är kapplöpningsfri eftersom det bara finns *en* tillståndssignal, som ju *inte* kan kapplöpa med sig själv.

- Nästa tillståndsavkodaren måste vara Glitchfri/Hazardfri
  (även konsensustermer tas med).

SR-låskretsens nät är sammanhängande i Karnaugh-diagrammet, det finns inga fler konsensustermer som behöver tas med.



Speciellt för asynkrona nät

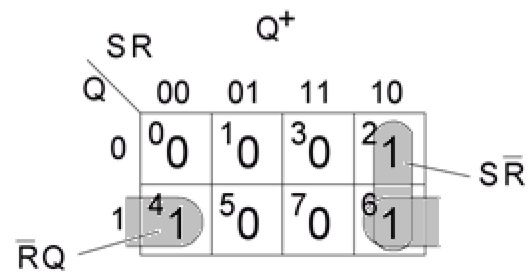
- Tillstånden måste kodas Kapplöpningsfritt (tex. Graykod).

SR-låskretsen är kapplöpningsfri eftersom det bara finns *en* tillståndssignal, som ju *inte* kan kapplöpa med sig själv.

- Nästa tillståndsavkodaren måste vara Glitchfri/Hazardfri (även konsensustermer tas med).

SR-låskretsens nät är sammanhängande i Karnaugh-diagrammet, det finns inga fler konsensustermer som behöver tas med.

SR-latchen är således en ”idiotsäker” konstruktion.
Större asynkrona sekvensnät är betydligt mera komplicerade att konstruera!

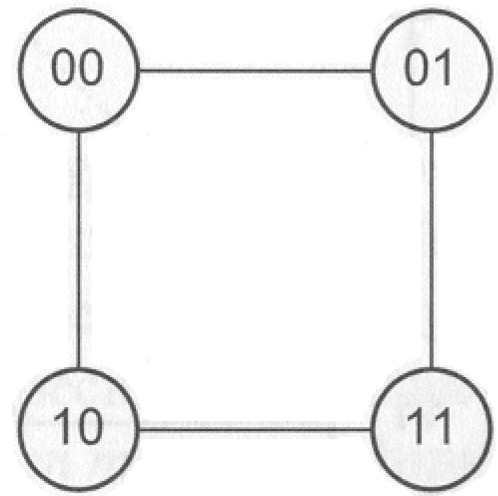


William Sandqvist william@kth.se

Tillståndsdiagram som hyperkuber

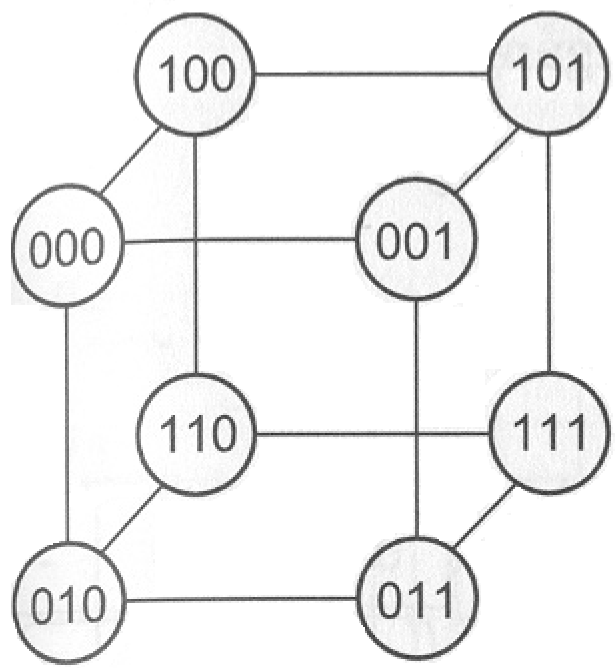
Tillståndsdiagrammet placeras ut på en hyperkub med Graykodade hörn.

För två tillståndsvariabler blir det en kvadrat.



Tillståndsdiagram som hyperkuber

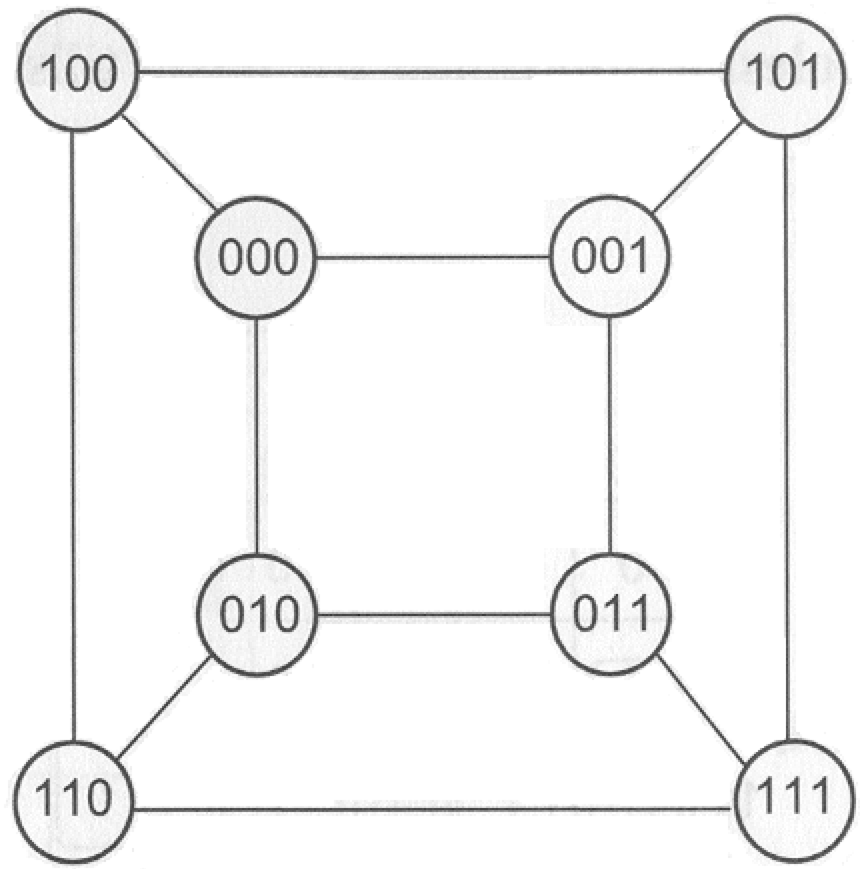
För tre tillståndsvariabler blir det en kub.



Tillståndsdiagram som hyperkuber

För tre tillståndsvariabler blir det en kub.

Det blir tydligare om man ”plattar till” kuben.

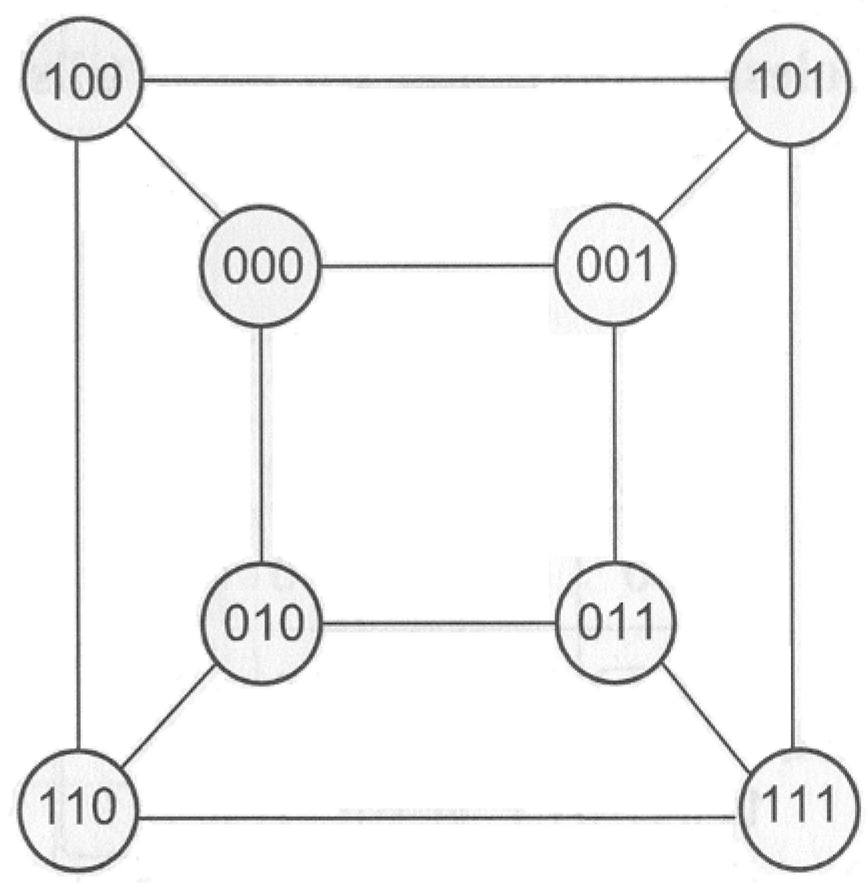


Tillståndsdiagram som hyperkuber

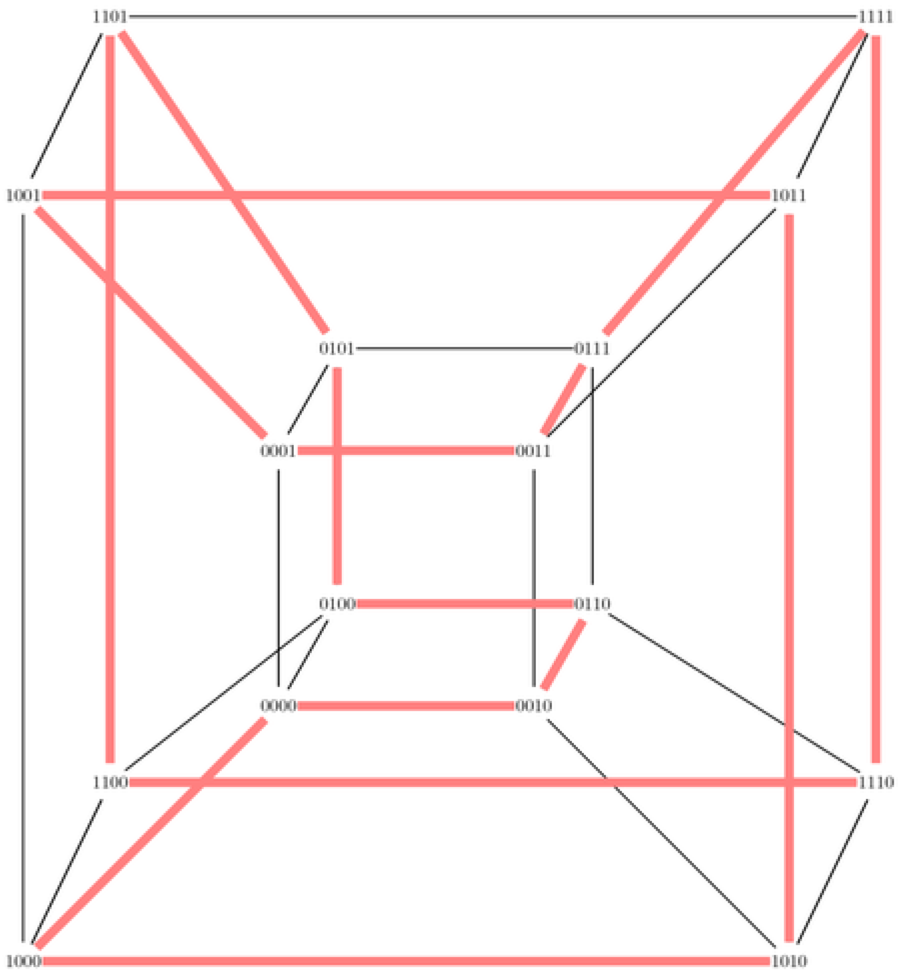
För tre tillståndsvariabler blir det en kub.

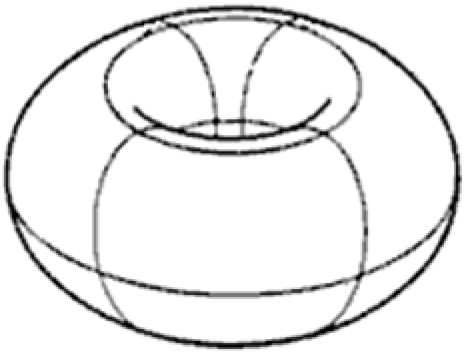
Det blir tydligare om man ”plattar till” kuben.

För fler variabler är principen densamma, men tillstånden placeras i hörnen av *hyperkuber* och det blir svårare att rita.



(Fyra variabler)

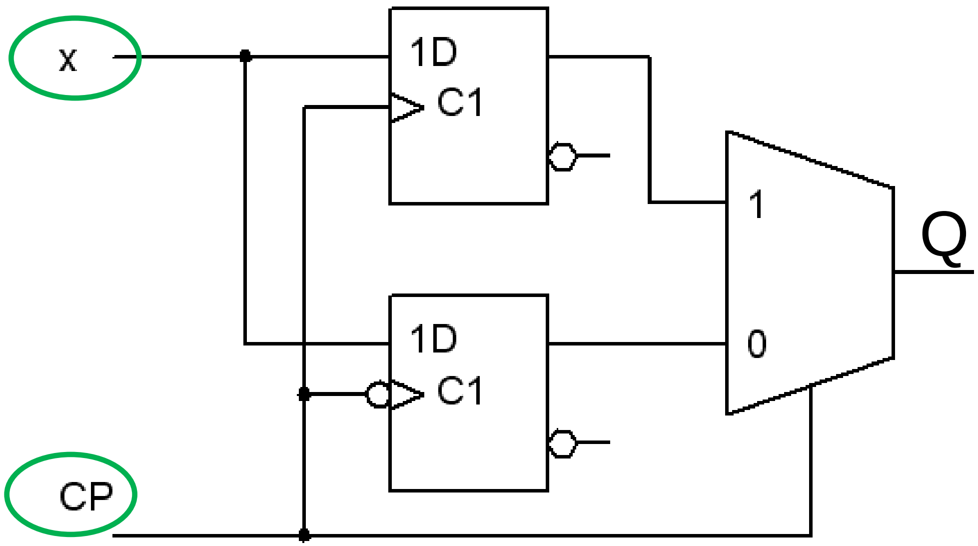




(Jfr. Karnaughdiagrammet.)

ab \ cd	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

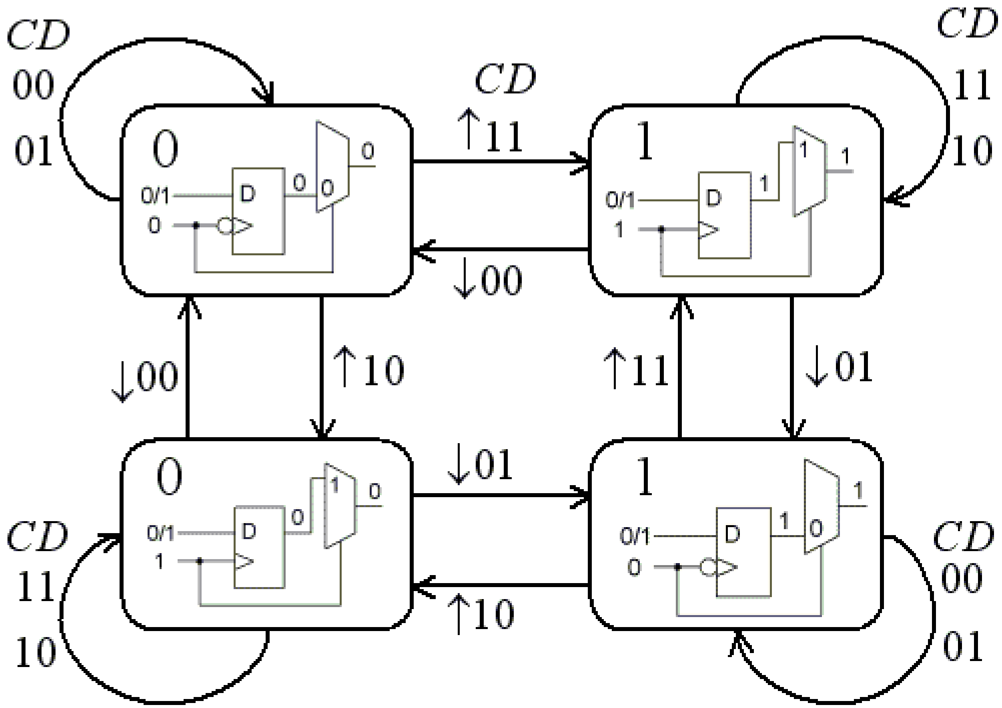
ÖH 11.4

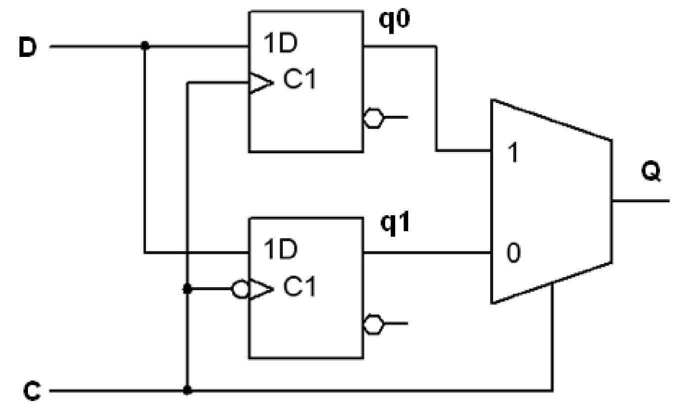


Analysera följande krets. Rita ett tillståndsdiagram.

Betrakta kretsen som ett asynkront sekvensnät där klockpulsingången är en av de asynkrona ingångarna. Vad har kretsen för funktion?

11.4 Positiv flank och negativ flank

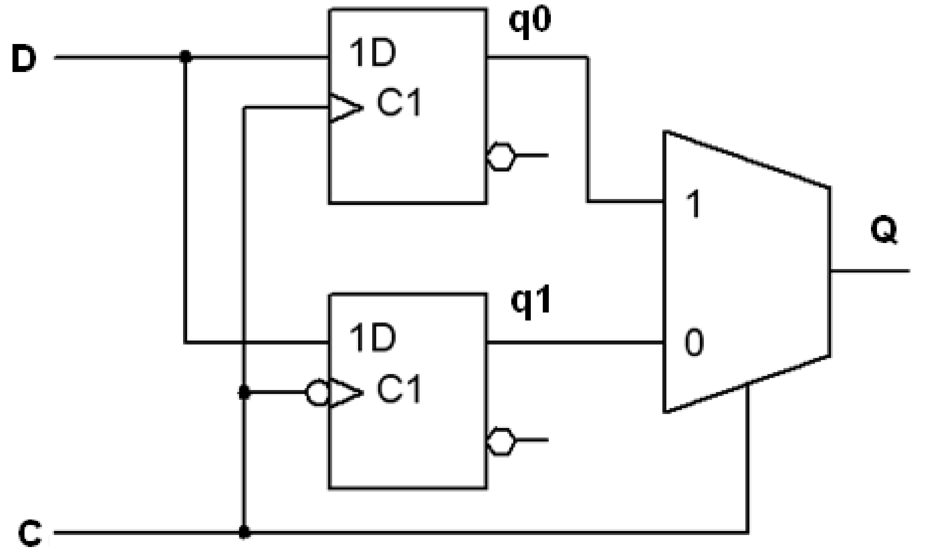


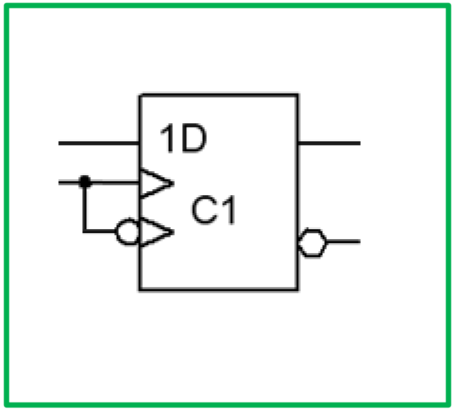


- Vid positiv flank ↑ går **C** från 0 till 1 och **C**=1 kopplar den övre **q0** vippan till utgången.
- Vid negativ flank ↓ går **C** från 1 till 0 och **C**=0 kopplar den undre **q1** vippan till utgången.

Resultatet blir en **D**-vippa som som reagerar på klockans *båda* flanker.

DETFF-vippan

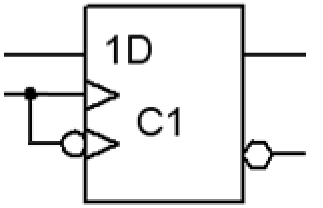




Dubbelflankvippan (DETFF, Double Edge Trigered Flip Flop) har fördelar vad gäller hastighet och effektförbrukning. Den kan i princip ge ett dubbelt så snabbt sekvensnät!
(Införande av DETFF-vippor skulle kräva nytänkande och omkonstruktion av den övriga logiken).

För att kunna dra nytta av fördelarna med DETFF-vippan måste den konstrueras som en egen komponent – dvs. som ett asynkront sekvensnät.

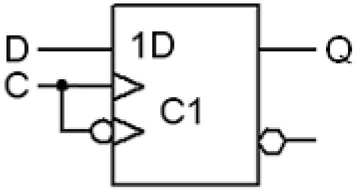
ÖH **11.5** DETFF



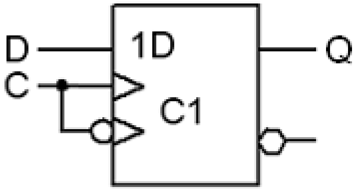
Konstruera en asynkron statemaskin som fungerar som en dubbelflankad D vippa (DETFF), dvs vippan skall ändra värde både på den *positiva* och den *negativa* flanken av klockan.

a) Härled FSMen.
b) Ta fram flödestabellen och minimera den.
c) Tilldela tillstånd (states), överför till Karnaughdiagram och härled de boolska uttrycken.
d) Rita kretsen.

11.5 Möjliga in/ut kombinationer



11.5 Möjliga in/ut kombinationer

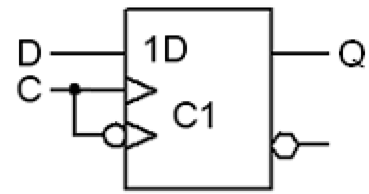


DETFF
Karakteristik

C	D	Q^+
0	-	Q
1	-	Q
↑	0	0
↑	1	1
↓	0	0
↓	1	1

11.5 Möjliga in/ut kombinationer

Det finns 4 ingångskombinationer (CD) och två utgångskombinationer (Q). Totalt 8 möjliga tillstånd (CD Q).

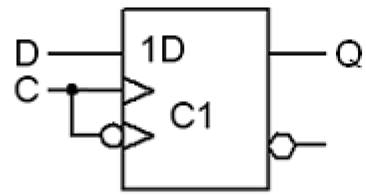


DETFF
Karakteristik

C D	Q^+
0 -	Q
1 -	Q
↑ 0	0
↑ 1	1
↓ 0	0
↓ 1	1

11.5 Möjliga in/ut kombinationer

Det finns 4 ingångskombinationer (CD) och två utgångskombinationer (Q). Totalt 8 möjliga tillstånd (CD Q).



Ett nytt nästa tillstånd får vi genom att ändra antingen C eller D. När C ändras får vi en positiv flank (↑) eller negativ flank (↓). För *båda* flankerna gäller att D kopieras till Q^+. (Enligt karakteristiken)

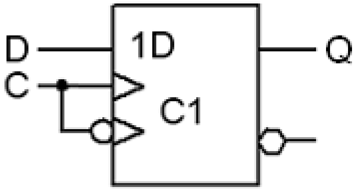
Nuvarande tillstånd		Nästa tillstånd
Namn:	(CD Q)	$(CD\ Q)^+$
A	00 0	
B	00 1	
C	01 0	
D	01 1	
E	10 0	
F	10 1	
G	11 0	
H	11 1	

DETFF Karakteristik

C D	Q^+
0 -	Q
1 -	Q
↑ 0	0
↑ 1	1
↓ 0	0
↓ 1	1

11.5 Möjliga in/ut kombinationer

Det finns 4 ingångskombinationer (CD) och två utgångskombinationer (Q). Totalt 8 möjliga tillstånd (CD Q).



Ett nytt nästa tillstånd får vi genom att ändra antingen C eller D. När C ändras får vi en positiv flank (↑) eller negativ flank (↓). För *båda* flankerna gäller att D kopieras till Q^+. (Enligt karakteristiken)

Nuvarande tillstånd		Nästa tillstånd
Namn:	(CD Q)	$(CD\ Q)^+$
A	00 0	01 0 C ↑ 10 0 E
B	00 1	01 1 D ↑ 10 0 E
C	01 0	00 0 A ↑ 11 1 H
D	01 1	00 1 B ↑ 11 1 H
E	10 0	↓ 00 0 A 11 0 G
F	10 1	↓ 00 0 A 11 1 H
G	11 0	↓ 01 1 D 10 0 E
H	11 1	↓ 01 1 D 10 1 F

DETFF Karakteristik

C D	Q^+
0 -	Q
1 -	Q
↑ 0	0
↑ 1	1
↓ 0	0
↓ 1	1

11.5 Flödestabell

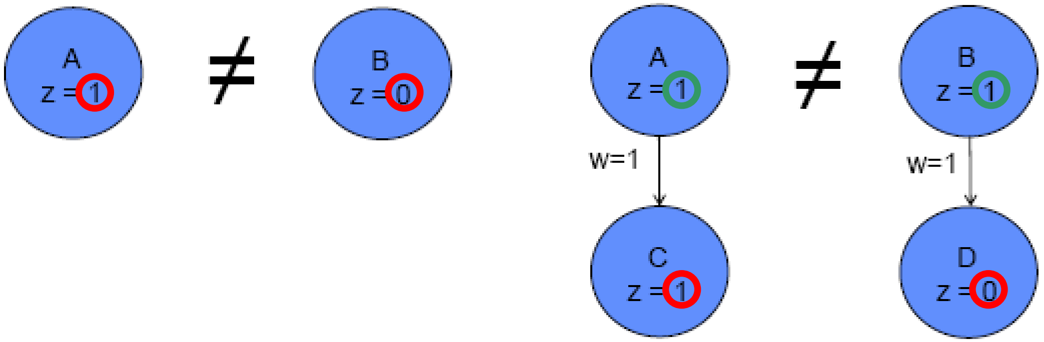
Nuvarande tillstånd		Nästa tillstånd
Namn:	(CD Q)	$(CD\ Q)^{+}$
A	00 0	01 0 C ↑ 10 0 E
B	00 1	01 1 D ↑ 10 0 E
C	01 0	00 0 A ↑ 11 1 H
D	01 1	00 1 B ↑ 11 1 H
E	10 0	↓ 00 0 A 11 0 G
F	10 1	↓ 00 0 A 11 1 H
G	11 0	↓ 01 1 D 10 0 E
H	11 1	↓ 01 1 D 10 1 F

Nuvarande tillstånd	Nästa tillstånd CD=				Output Q
	00	01	11	10	
A	A	C	-	E	0
B	B	D	-	E	1
C	A	C	H	-	0
D	B	D	H	-	1
E	A	-	G	E	0
F	A	-	H	F	1
G	-	D	G	E	0
H	-	D	H	F	1

Stabila tillstånd markeras med ring. Kontrollera att varje kolumn ”CD” innehåller minst ett stabilt tillstånd, annars får man ju ett ”oscillerande” nät för den insignalen. Don’t-care ”-” införs där insignalen ”CD” innehåller mer än *en* ändring från det stabila tillståndet på raden.

11.5 Tillståndsminimering

A och B är *inte* **ekvivalenta** om …



Ekvivalens innebär att tillstånden ska vara stabila för samma insignaler, och ha sina don’t care för samma insignaler – så att man inte förlorar flexibiliteten inför den fortsatta minimeringen.

Kompatibilitet blir olika för Moore eller Mealy. För Moore-kompatibla automater gäller att utsignalerna måste vara lika, och utsignalerna i efterföljar-tillstånden (alla, om flera) måste också vara lika. Annars är de två tillstånden *inte* kompatibla!

Tillståndsminimering

Nuvarande tillstånd	Nästa tillstånd CD= 00	01	11	10	Output Q
A	A	C	-	E	0
B	B	D	-	E	1
C	A	C	H	-	0
D	B	D	H	-	1
E	A	-	G	E	0
F	A	-	H	F	1
G	-	D	G	E	0
H	-	D	H	F	1

Vi startar med ett block med alla tillstånd
P_1 = (ABCDEFGH)

Inga **Ekvivalenta** tillstånd**,**
vi undersöker **Kompatibilitet**

Tillstånden delas först i två block efter utsignal. ACEG har utsignal 0, BDFH har utsignal 1.
P_2 = [ACEG][BDFH]

A och C har *samma* efterföljar-tillstånd
(eftersom don’t-care kan utnyttjas som H eller E)

```
AC-E
ACH-
```

P_3 = [(AC)…][BDFH]

(För **kompatibilitet** räcker det här med att *utsignalen* från efterföljar-tillstånden är samma, det behöver inte vara exakt samma tillstånd som det råkar vara i detta exempel.)

Tillståndsminimering

Nuvarande tillstånd	Nästa tillstånd CD= 00	01	11	10	Output Q
A	A	C	-	E	0
B	B	D	-	E	1
C	A	C	H	-	0
D	B	D	H	-	1
E	A	-	G	E	0
F	A	-	H	F	1
G	-	D	G	E	0
H	-	D	H	F	1

E och G har *samma* efterföljar-tillstånd
(eftersom don't-care kan utnyttjas som A eller D)

A-G**E**
-D**G**E

P_3 = [(AC)(EG)][BDFH]

B och D har *samma* efterföljar-tillstånd
(eftersom don't-care kan utnyttjas som H eller E)

BD-E
B**D**H-

P_3 = [(AC)(EG)][(BD)…]

F och H har *samma* efterföljar-tillstånd
(eftersom don't-care kan utnyttjas som A eller D)

A-H**F**
-D**H**F

$\boxed{P_3 = (AC)(EG)(BD)(FH)}$

Vi klarar oss med fyra tillstånd!

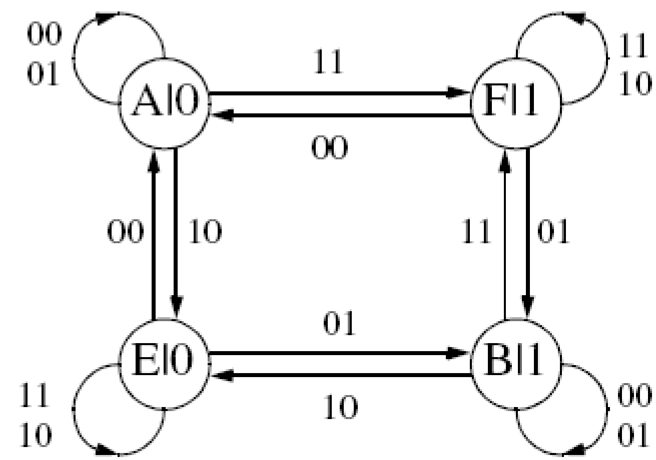
11.5 Ny Flödestabell

Nuvarande tillstånd	Nästa tillstånd CD= 00	01	11	10	Output Q
A	A	C	-	E	0
B	B	D	-	E	1
C	A	C	H	-	0
D	B	D	H	-	1
E	A	-	G	E	0
F	A	-	H	F	1
G	-	D	G	E	0
H	-	D	H	F	1

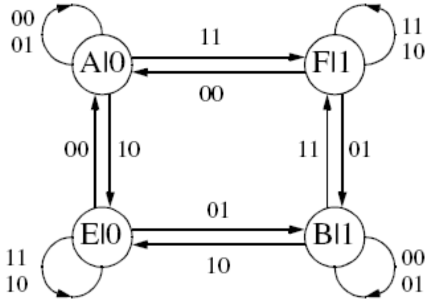
De nya tillstånden betecknas:
AC→**A**, EG→**E**, BD→**B**, FH→**F**.

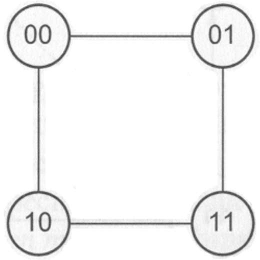
Nuvarande tillstånd	Nästa tillstånd CD= 00	01	11	10	Output Q
A	A	A	F	E	0
B	B	B	F	E	1
E	A	B	E	E	0
F	A	B	F	F	1

Tillståndsdiagram



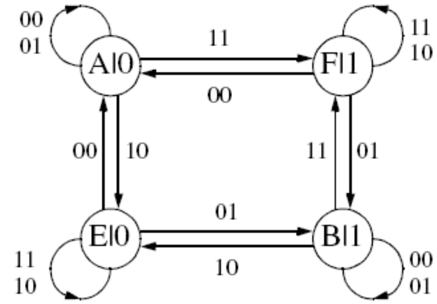
11.5 Tillståndskodning

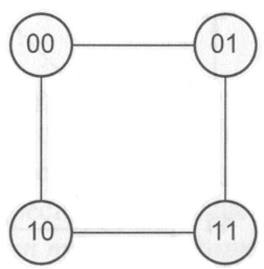




Tillstånden (q_1q_0), placeras i hörnen på en Gray-kodad kvadrat. Tex. A=00, F=01, B=11, E=10.

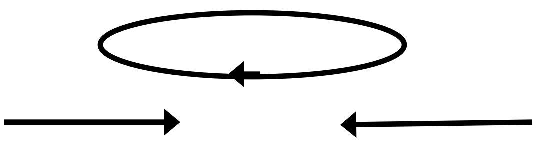
11.5 Tillståndskodning



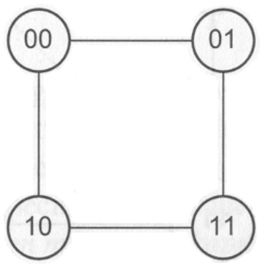


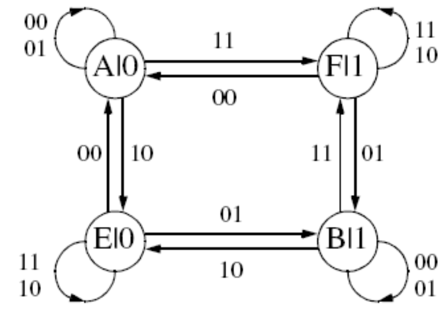
Tillstånden (q_1q_0), placeras i hörnen på en Gray-kodad kvadrat. Tex. A=00, F=01, B=11, E=10.

Även alla ”rotationer” och ”speglingar” av koden är giltiga tillståndskodningar.



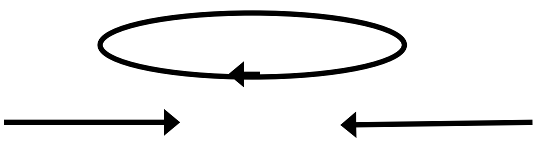
11.5 Tillståndskodning





Tillstånden (q_1q_0), placeras i hörnen på en Gray-kodad kvadrat. Tex. A=00, F=01, B=11, E=10.

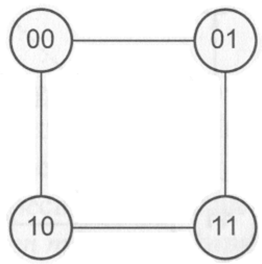
Även alla ”rotationer” och ”speglingar” av koden är giltiga tillståndskodningar.

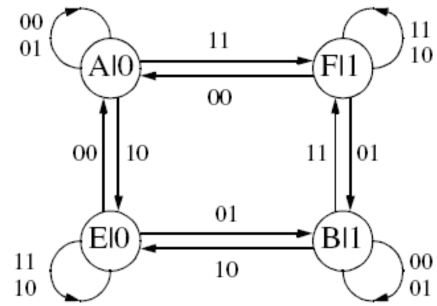


A	**F**	**B**	**E**	**A**	**F**	**B**	**E**
00	01	11	10	10	11	01	00
01	11	10	00	00	10	11	01
11	10	00	01	01	00	10	11
10	00	01	11	11	01	00	10

William Sandqvist william@kth.se

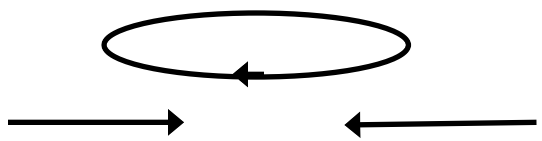
11.5 Tillståndskodning





Tillstånden (q_1q_0), placeras i hörnen på en Gray-kodad kvadrat. Tex. A=00, F=01, B=11, E=10.

Även alla ”rotationer” och ”speglingar” av koden är giltiga tillståndskodningar.

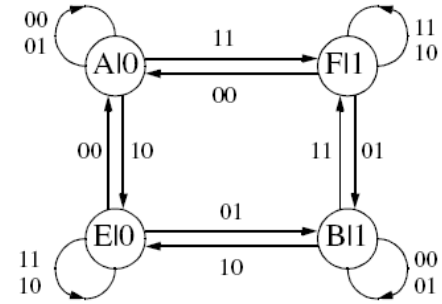


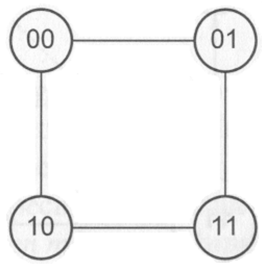
A	F	B	E
00	01	11	10
01	11	10	00
11	10	00	01
10	00	01	11

A	F	B	E
10	11	01	00
00	10	11	01
01	00	10	11
11	01	00	10

Detta blir vår godtyckligt valda tillståndskod.

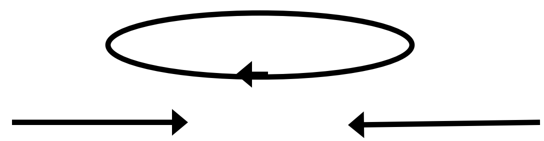
11.5 Tillståndskodning





Tillstånden (q_1q_0), placeras i hörnen på en Gray-kodad kvadrat. Tex. A=00, F=01, B=11, E=10.

Även alla ”rotationer” och ”speglingar” av koden är giltiga tillståndskodningar.

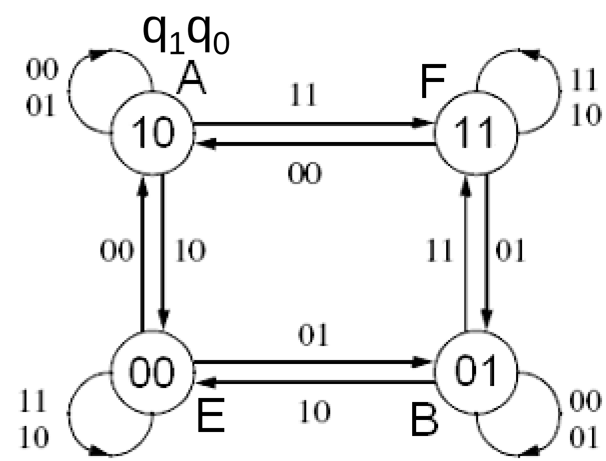


A	F	B	E	A	F	B	E
00	01	11	10	10	11	01	00
01	11	10	00	00	10	11	01
11	10	00	01	01	00	10	11
10	00	01	11	11	01	00	10

Detta blir vår godtyckligt valda tillståndskod.

Är detta bästa tillståndskodningen? Uttömmande sökning (=prova alla) är oftast enda lösningen för den som vill veta!

11.5 Exitationstabell



Nuvarande tillstånd	Nästa tillstånd CD= 00	01	11	10	Output Q
A	A	A	F	E	0
B	B	B	F	E	1
E	A	B	E	E	0
F	A	B	F	F	1

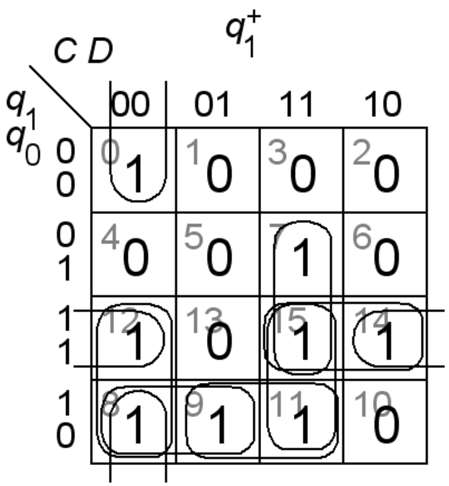
Nuvarande tillstånd q_1q_0	Nästa tillstånd CD= 00	01	11	10	Output Q
10	$\boxed{10}$	$\boxed{10}$	11	00	0
01	$\boxed{01}$	$\boxed{01}$	11	00	1
00	10	01	$\boxed{00}$	$\boxed{00}$	0
11	10	01	$\boxed{11}$	$\boxed{11}$	1

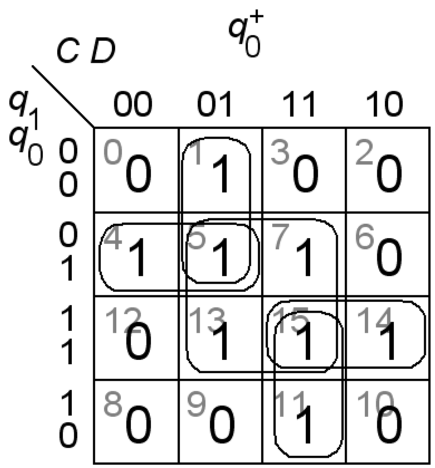
11.5 Karnaughdiagram

Nuvarande tillstånd q_1q_0	Nästa tillstånd CD= 00	01	11	10	Output Q
10	10	10	11	00	0
01	01	01	11	00	1
00	10	01	00	00	0
11	10	01	11	11	1

På K-map-form:

Nuvarande tillstånd q_1q_0	Nästa tillstånd CD= 00	01	11	10	Output Q
00	10	01	00	00	0
01	01	01	11	00	1
11	10	01	11	11	1
10	10	10	11	00	0



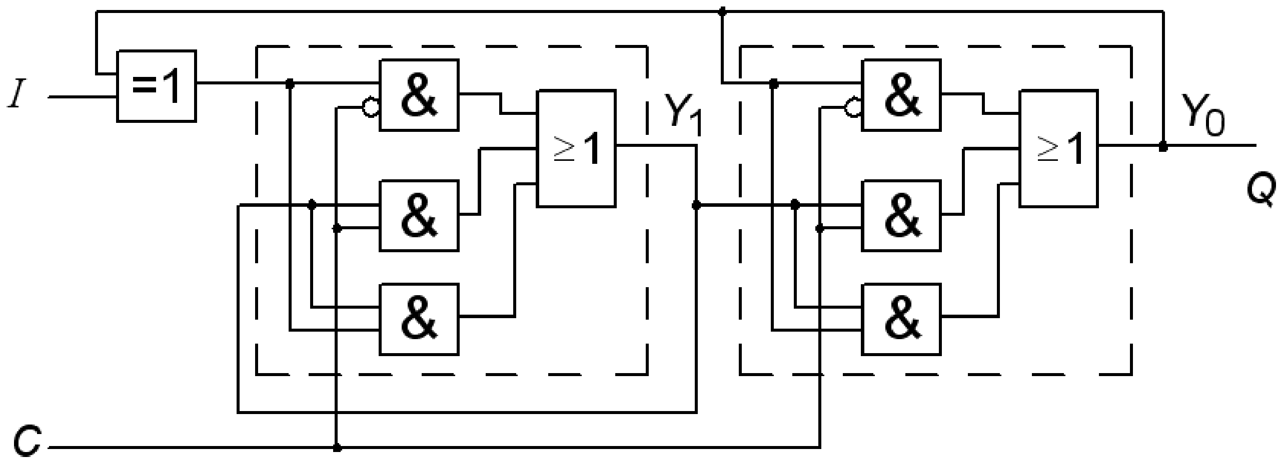


$$Q = q_0$$

$$q_1^+ = CDq_1 + CDq_0 + \overline{C}\,\overline{D}\,q_1 + \overline{C}\,\overline{D}\,\overline{q}_0 + q_1\,\overline{q}_0\,\overline{C} + q_1\,\overline{q}_0\,D + q_1q_0C + q_1q_0\overline{D}$$

$$q_0^+ = q_0D + \overline{q}_1q_0\overline{C} + \overline{q}_1CD + q_1CD + q_1q_0C$$

ÖH 11.6 Analysera



Analysera ovanstående krets.

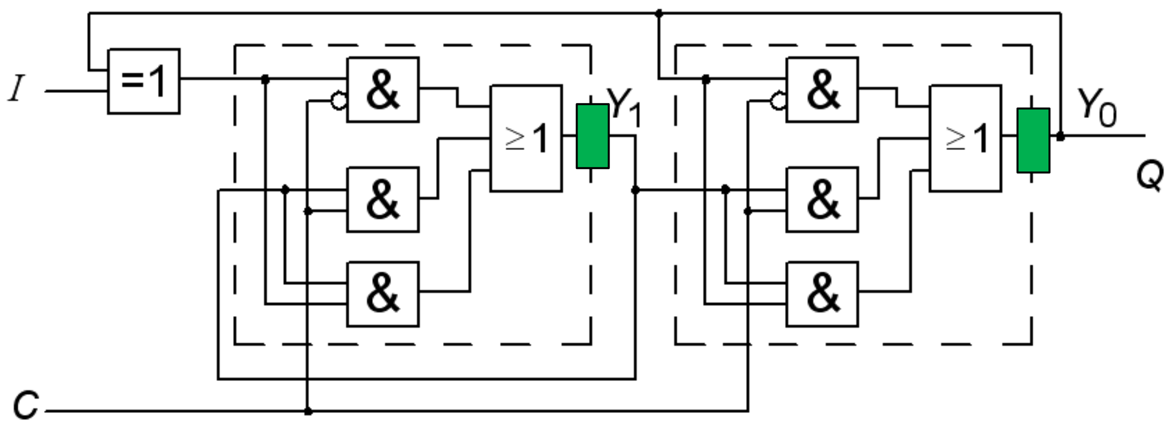
a) Härled de Boolska uttrycken för tillståndsvariablerna Y_1 och Y_0.

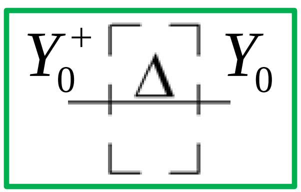
b) Härled exitationstabell. Ledning: Vilken funktion (inom streckat) finns i de inre looparna.

c) Härled flödestabell, tilldela symboliska states och rita FSM.

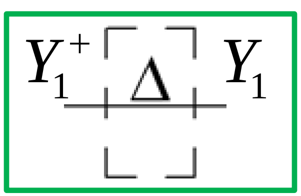
d) Vilken vippa motsvarar detta?

11.6 Boolska funktioner





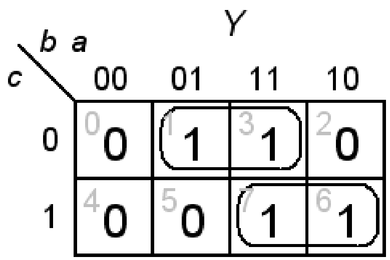
$$Y_0^+ = Y_0Y_1 + Y_0\overline{C} + Y_1C$$

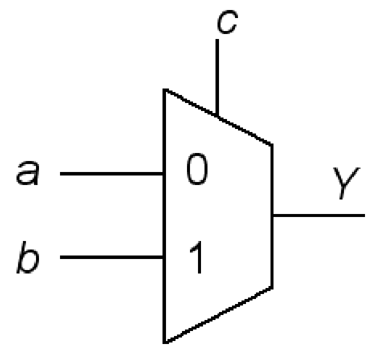


$$Y_1^+ = Y_1(Y_0 \oplus I) + (Y_0 \oplus I)\overline{C} + Y_1\,C$$

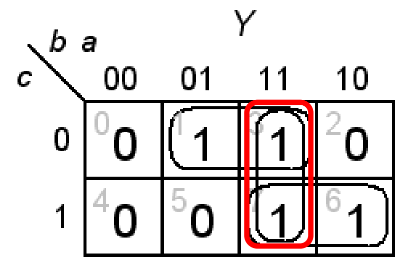
11.6 Glitch-fri MUX?

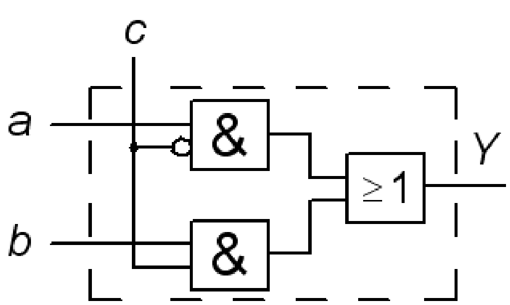
Vanlig MUX:

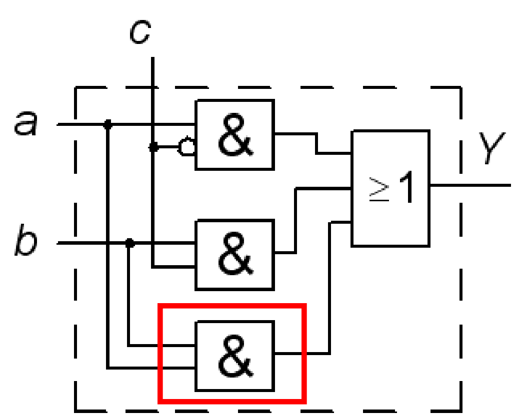




Glitch-fri MUX:

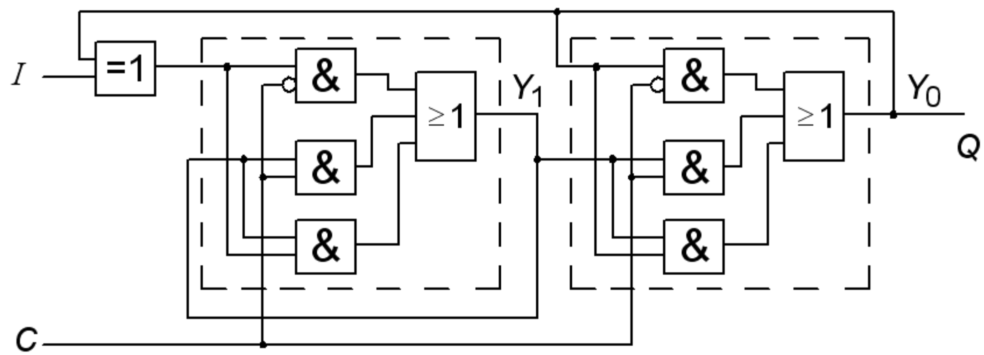


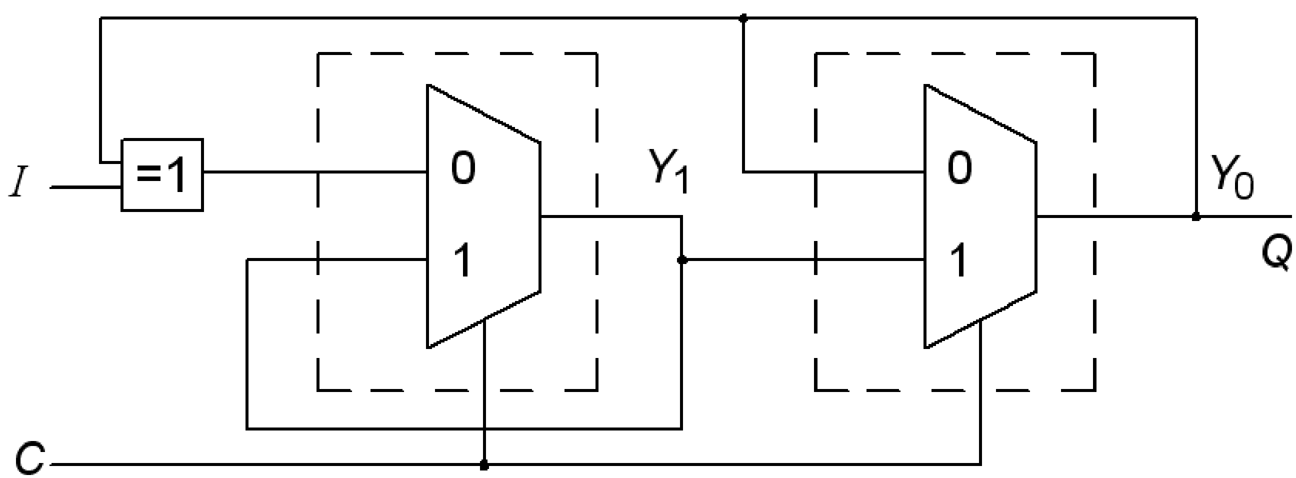




11.6 Två Glitch-fria MUXar

Nätet kan ses som sammansatt av *två* Glitch-fria MUXar. Detta faktum kan utnyttjas om man vill *resonera* sig fram till nätets funktion.





11.6 Boolska ekvationer

Vi använder de boolska funktionerna för att härleda funktionen.

$$Q = Y_0$$

$$\boxed{Y_1^+} = Y_1(Y_0 \oplus I) + (Y_0 \oplus I)\overline{C} + Y_1 C =$$

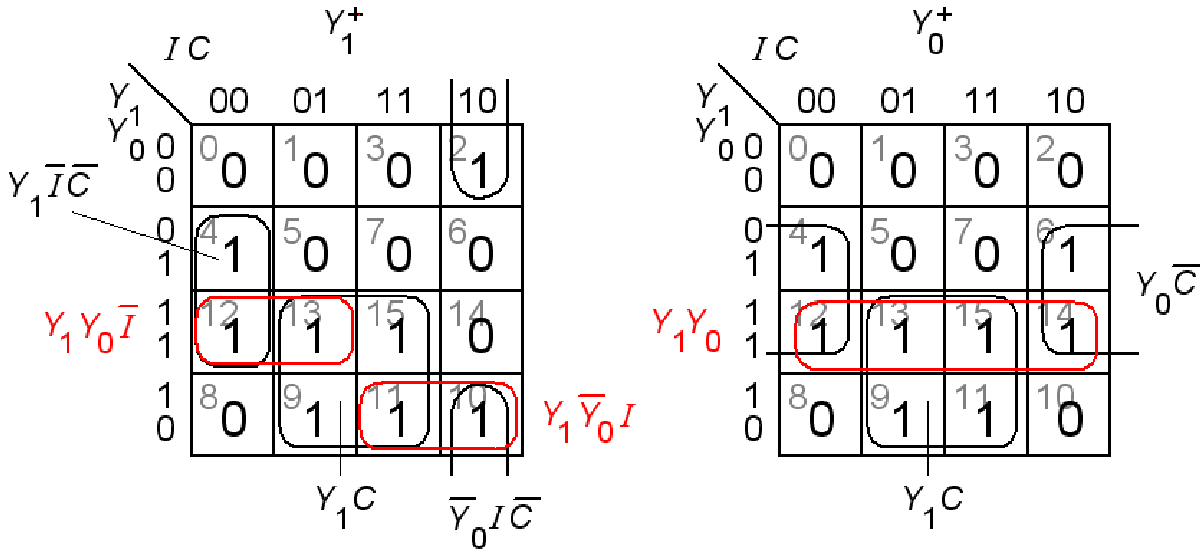
$$= Y_1(Y_0 \overline{I} + \overline{Y}_0 I) + (Y_0 \overline{I} + \overline{Y}_0 I)\overline{C} + Y_1 C =$$

$$= \boxed{Y_1 Y_0 \overline{I} + Y_1 \overline{Y}_0 I + Y_0 \overline{I}\,\overline{C} + \overline{Y}_0 I \overline{C} + Y_1 C}$$

$$\boxed{Y_0^+} = \boxed{Y_0 Y_1 + Y_0 \overline{C} + Y_1 C}$$

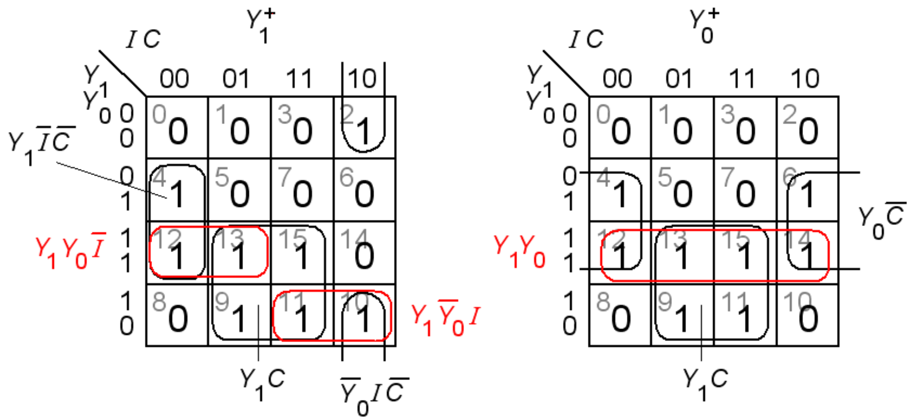
11.6 Excitationstabell

$$Y_1^+ = Y_1Y_0\overline{I} + Y_1\overline{Y}_0 I + Y_0\overline{I}\,\overline{C} + \overline{Y}_0 I\overline{C} + Y_1 C \qquad Y_0^+ = Y_0Y_1 + Y_0\overline{C} + Y_1C$$



Rödmarkerade inrigningar är kretsens Hazardcover

11.6 Exitationstabell



Nuvarande tillstånd Y_1Y_0	Nästa tillstånd IC= 00	01	11	10	Output Q
00	$\boxed{00}$	$\boxed{00}$	$\boxed{00}$	10	0
01	11	~~00~~	00	$\boxed{01}$	1
11	$\boxed{11}$	$\boxed{11}$	$\boxed{11}$	01	1
10	00	~~11~~	11	$\boxed{10}$	0

Omöjliga tillstånd betecknas som ~~genomstrukna~~. Det gäller tillstånd, som för att nås, skulle kräva *två* ändringar av insignalen från det stabila tillståndet på den aktuella raden.

$$Q = Y_0$$

IC 10→01 omöjlig samtidig ändring av ingångssignalerna.

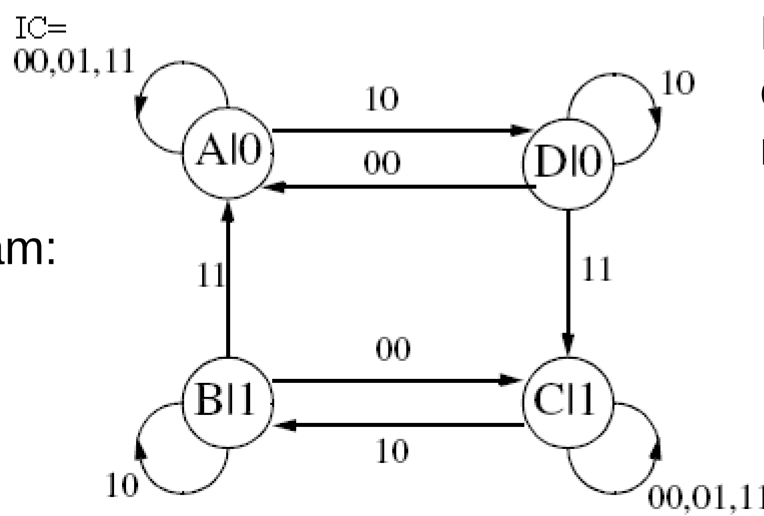
11.6 Flödestabell

Nuvarande tillstånd Y_1Y_0	Nästa tillstånd IC= 00	01	11	10	Output Q
00	$\boxed{00}$	$\boxed{00}$	$\boxed{00}$	10	0
01	11	~~00~~	00	$\boxed{01}$	1
11	$\boxed{11}$	$\boxed{11}$	$\boxed{11}$	01	1
10	00	~~11~~	11	$\boxed{10}$	0

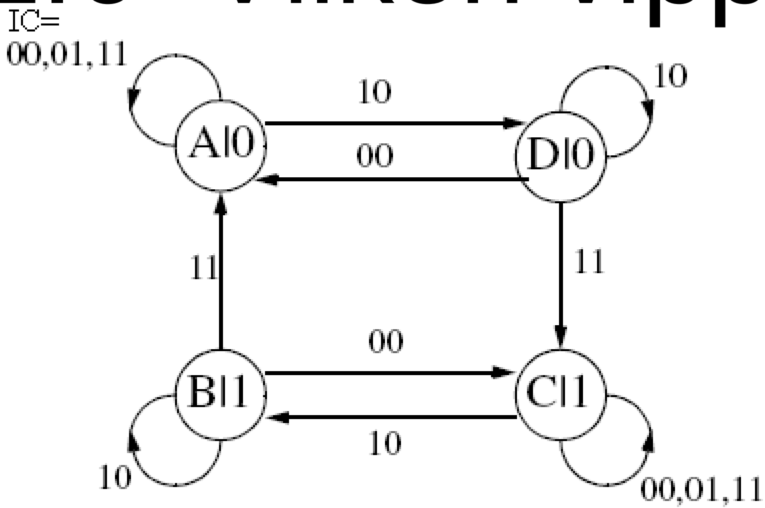
Nuvarande tillstånd Y_1Y_0	Nästa tillstånd IC= 00	01	11	10	Output Q
A	A	A	A	D	0
B	C	~~A~~	A	B	1
C	C	C	C	B	1
D	A	~~C~~	C	D	0

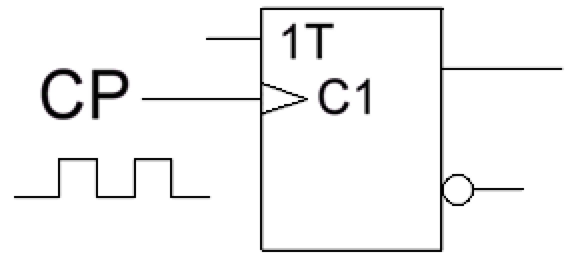
De omöjliga tillstånden (~~genomstrukna~~) skulle kunna utnyttjas som Don't-care om man vid ett annat tillfälle omkodar nätet.

Tillståndsdiagram:



11.6 Vilken vippa är det?





Om I = **1** och C är klockpulser 1,0,1,0… blir sekvensen:

IC: **1**0 **1**1 **1**0 **1**1, D-C-B-A-D-C-B-A Q: 0-1-1-0-0-1-1-0
Dvs. vippan togglar på positiv flank (↑) hos C.

Om I = 0 blir det i stället ”på stället marsch”
A$\rightarrow$A och D$\rightarrow$A   Q = 0
C $\rightarrow$C och B$\rightarrow$C   Q = 1

Vippan byter tillstånd vid övergångarna från C = 0 till C = 1, således en positivt flanktriggad (↑) T-vippa (I = *T*).