

Forelesning 7

Tilstandsmaskin

Hovedpunkter

- Tilstandsmaskin
- Tilstandstabell
- Tilstandsdiagram
- Analyse av D flip-flop basert tilstandsmaskin
- Reduksjon av antall tilstander
- Tilordning av tilstandskoder
- Designprosedyre for tilstandsmaskin basert på D flip-flops

Tilstandsmaskin

En **tilstandsmaskin** er et **sekvensielt system** som gjennomløper et sett med **tilstander** styrt av verdiene på inngangssignalene

Tilstanden systemet befinner seg i, pluss evt. **inngangsverdier** bestemmer **utgangsverdiene**

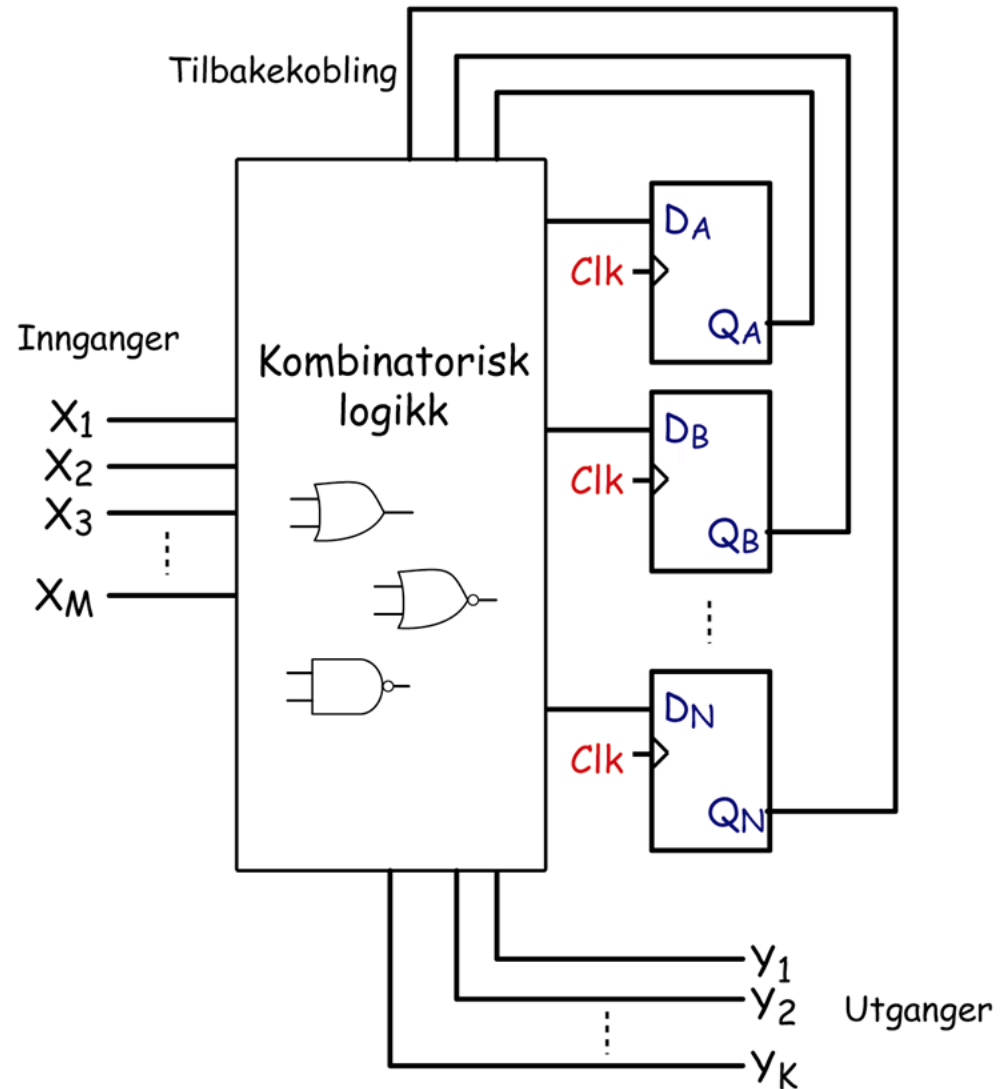
Tilstandsmaskins-konseptet gir en **enkel** og **oversiktlig** måte å designe **avanserte system** på

Tilstandsmaskin

Generell tilstandsmaskin basert på D flip-flops

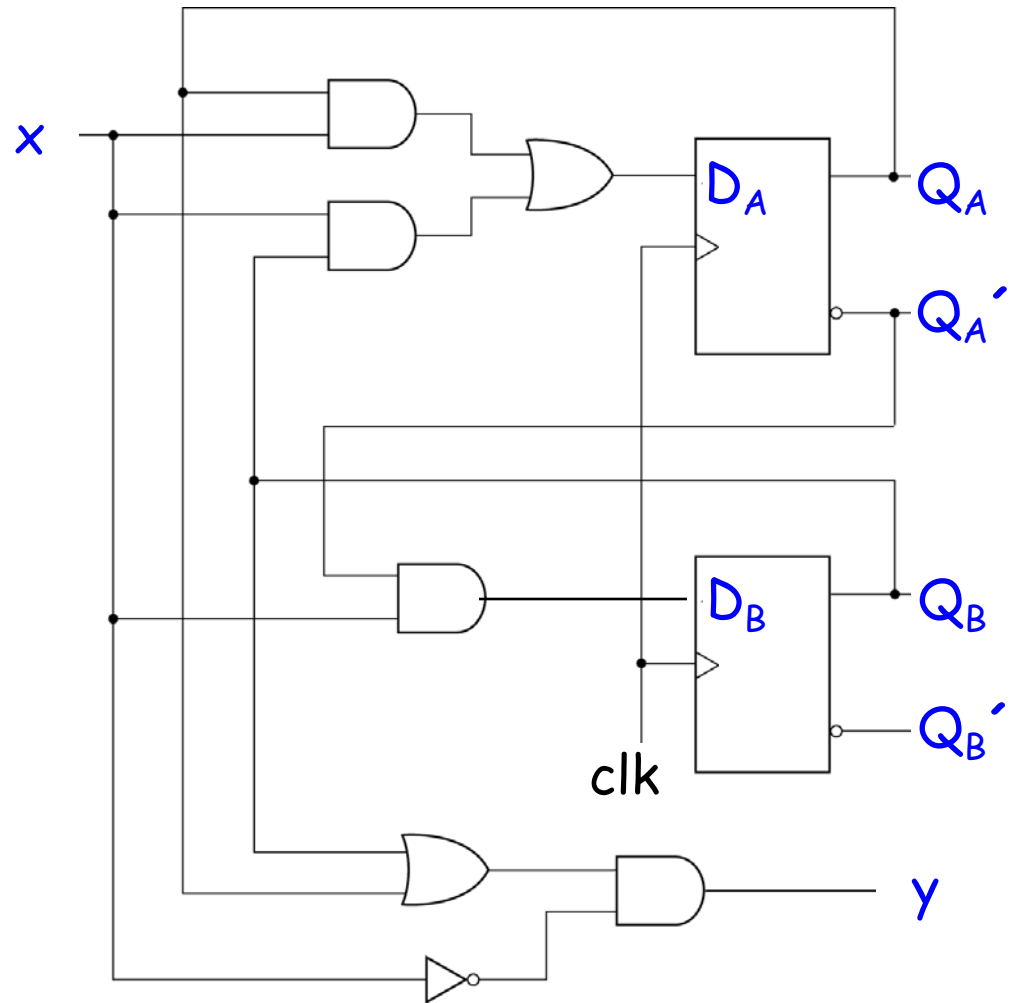
N-stk flip-flops gir 2^N forskjellige tilstander

Utgangssignalene er en funksjon av nåværende tilstand pluss evt. inngangsverdier



Eksempel nr.1

Tilstandsmaskin der
utgang y er en funksjon
av $tilstanden$ gitt av
verdiene til Q_A og Q_B ,
samt inngangen x



Tilstandstabell

Tilstandstabell = sannhetstabell for tilstandsmaskin

Eksempel nr.1: En inngang, en utgang og 2 stk. D flip-flops

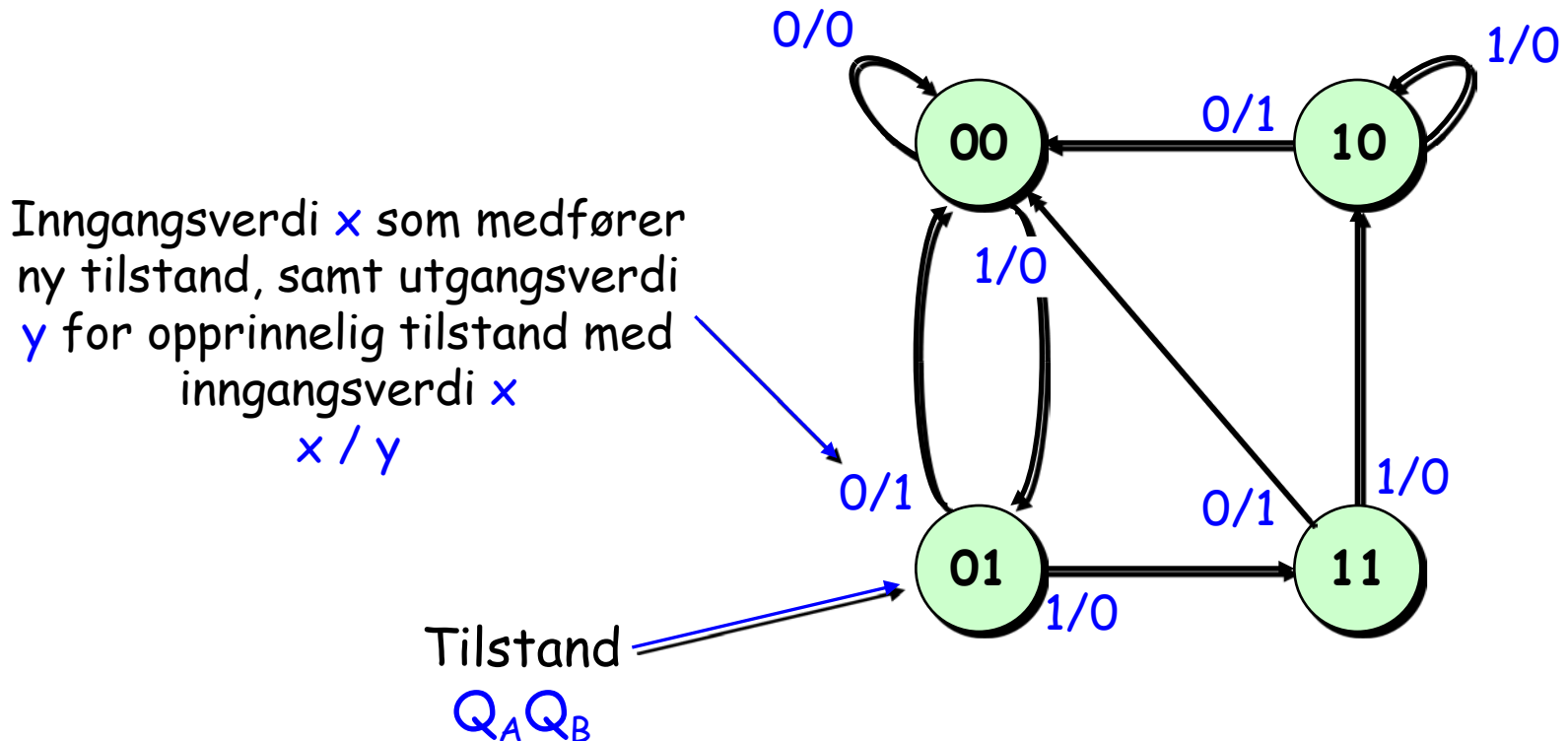
Nåværende tilstand			Utgang for neste nåværende tilstand		
Q_A	Q_B	Inngang x	Q_A	Q_B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

M.H

Tilstandsdiagram

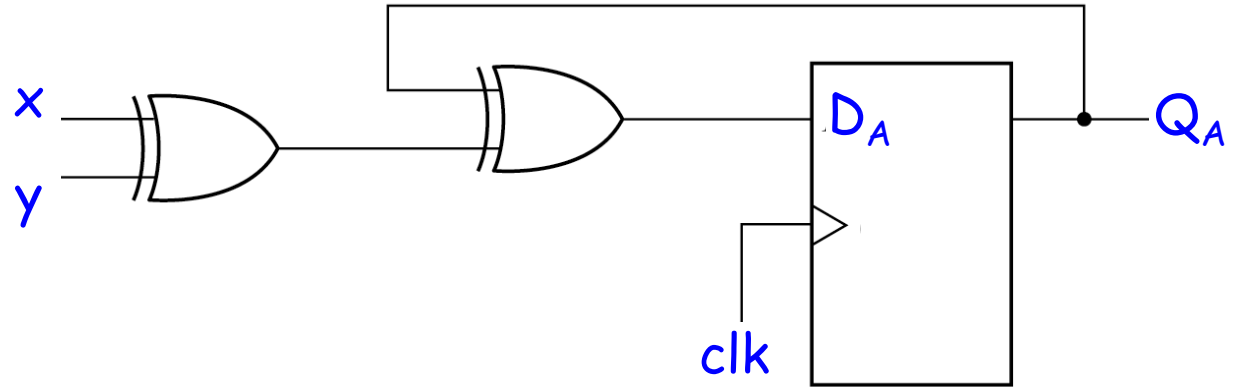
Tilstandsdiagram = grafisk illustrasjon av egenskapene til en tilstandsmaskin

Eksempel nr.1:



Eksempel nr.2

To innganger x
og y , en utgang
som bare er gitt
av tilstanden Q_A

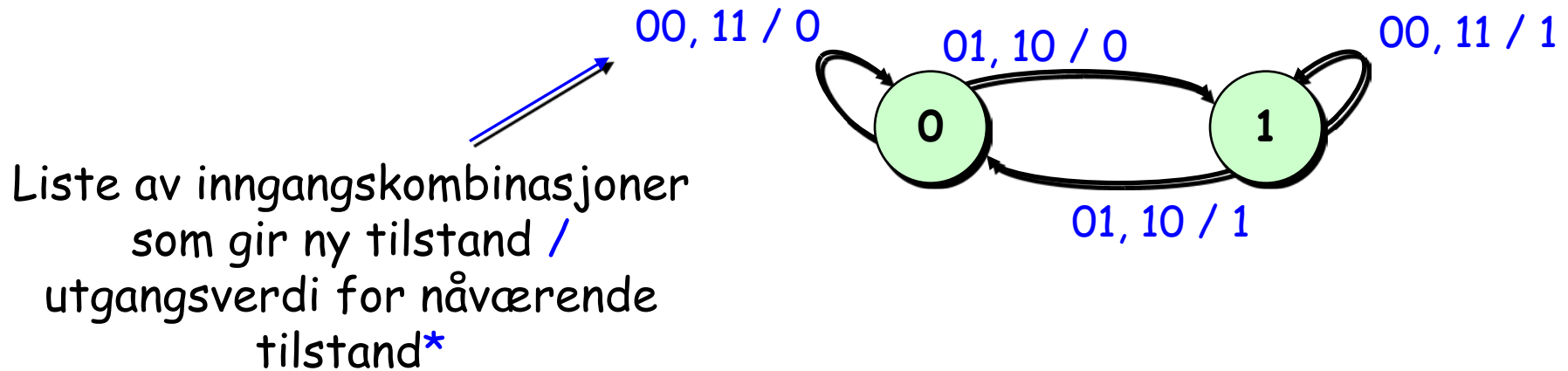


Nåværende tilstand Q_A	Innganger		Utgang for Neste nåværende tilstand	
	x	y	Q_A	Q_A
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

M.H

Eksempel nr.2

Tilstandsdiagram

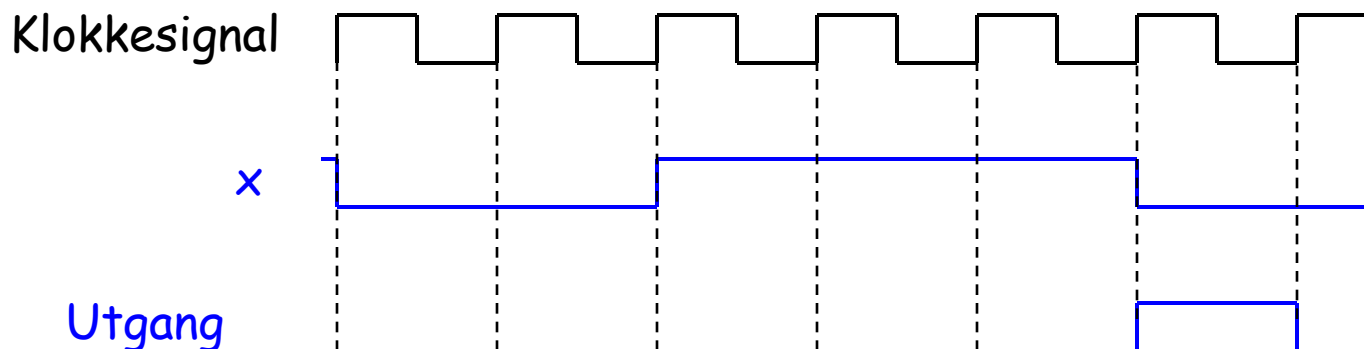


*Merk at i dette tilfelle er
utgangsverdien kun avhengig av
tilstanden (uavhengig av
inngangsverdiene)

Eksempel nr.3 - design av sekvensdetektor

Ønsker å lage en krets som finner ut om det har forekommet tre eller flere "1"ere etter hverandre i en klokke bit-sekvens x

Klokke bit-sekvens: Binært signal som kun kan skifte verdi synkront med et klokkesignal



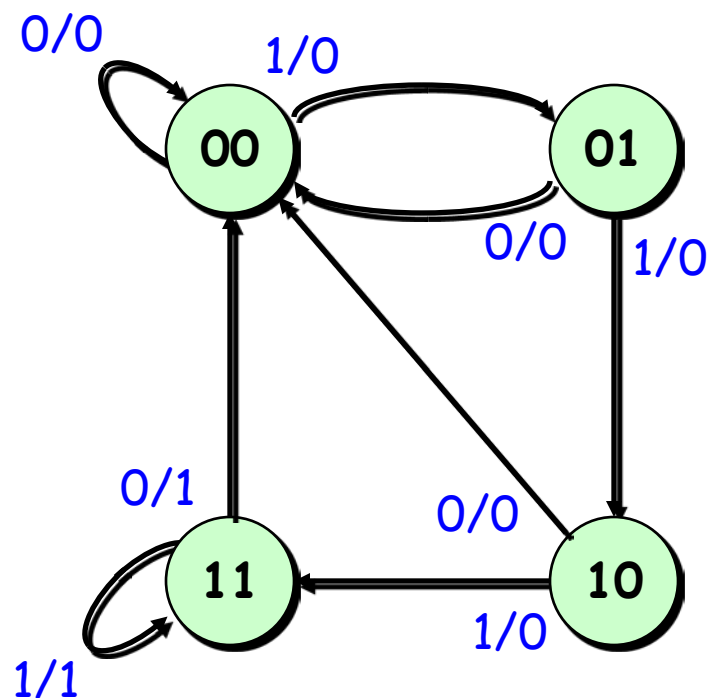
Eksempel nr.3 - design av sekvensdetektor

Tilstandsdiagram

Velger å ha 4 tilstander. Lar hver tilstand symbolisere antall "1"ere som ligger etter hverandre i bit-sekvensen.

Inngang: bit-sekvens x

Utgang: gitt av tilstanden, "0" for tilstand 0-2, "1" for tilstand 3



Eksempel nr.3

Bruker D flip-flops

D_A og D_B settes til de verdiene man ønsker at Q_A og Q_B skal ha i neste tilstand

$$D_A = Q_A' Q_B x + Q_A Q_B' x + Q_A Q_B x$$

$$D_B = Q_A' Q_B' x + Q_A Q_B' x + Q_A Q_B x$$

$$y = Q_A Q_B$$

Nåværende tilstand			Utgang for neste nåværende tilstand		
Inngang					
Q_A	Q_B	x	Q_A	Q_B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

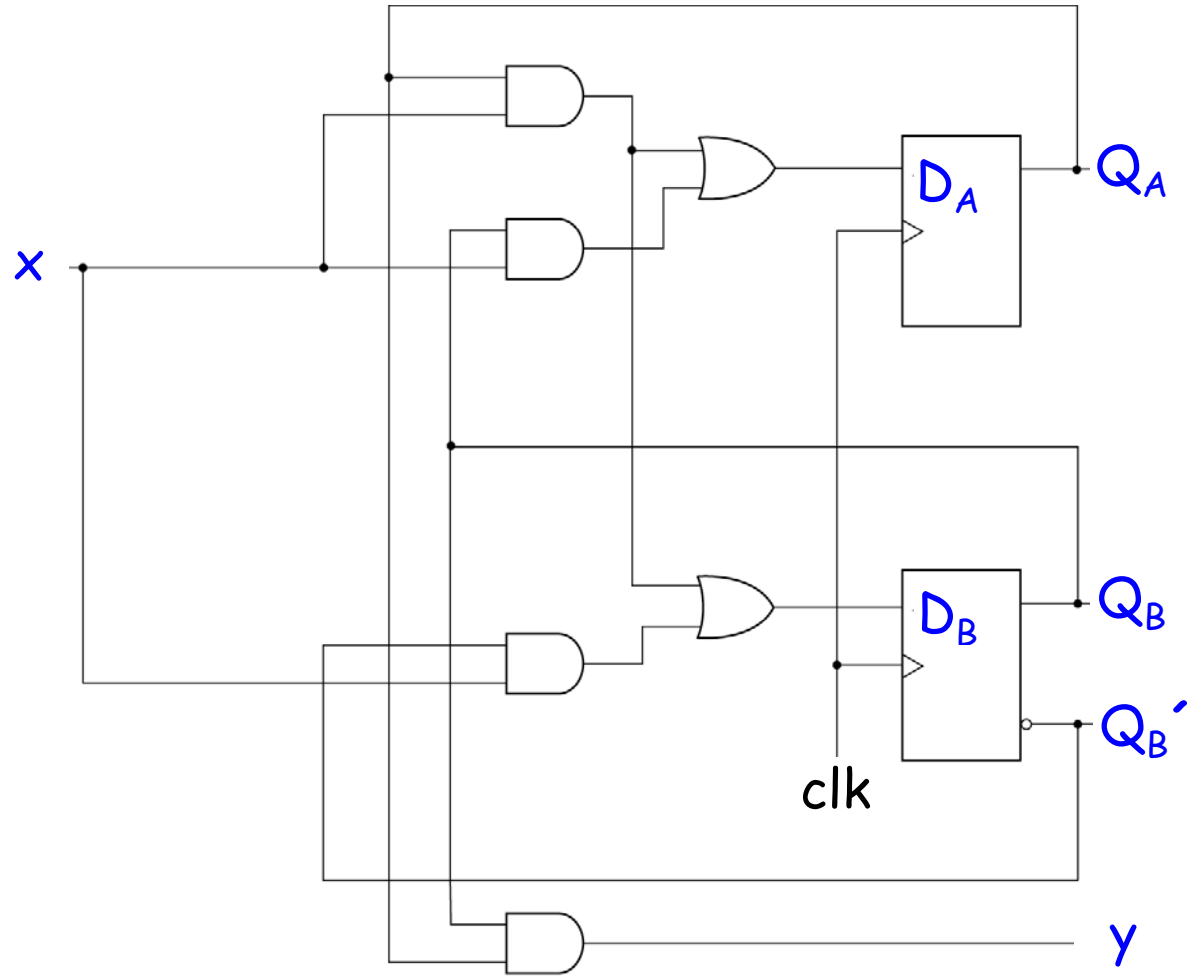
Eksempel nr.3

Forenkler
uttrykkene med
Karnaugh-diagram

$$D_A = Q_A x + Q_B x$$

$$D_B = Q_A x + Q_B' x$$

$$y = Q_A Q_B$$



Reduksjon av tilstander

En tilstandsmaskin gir oss en eller flere **utgangssignal** som **funksjon** av en eller flere **inngangssignal**

Hvordan dette implementeres internt i maskinen er uinteressant sett utenifra

I noen tilfeller kan man fjerne tilstander (forenkle designet) uten å påvirke inngangs/utgangs-funksjonene

Reduksjon av tilstander

Hvis **to tilstander** har samme utgangssignal, samt leder til de **samme nye tilstandene** gitt like inngangsverdier, er de to opprinnelige tilstandene **like**. En tilstand som er lik en annen tilstand kan fjernes.

Reduksjon av tilstander

Eksempel:

Tilstand **G** er lik tilstand **E**

Nåværende tilstand	Inngang	Neste tilstand	Utgang
A	0	B	0
A	1	B	0
B	0	C	0
B	1	D	0
C	0	A	0
C	1	D	0
D	0	E	0
D	1	F	1
E	0	A	0
E	1	F	1
F	0	G	0
F	1	F	1
G	0	A	0
G	1	F	1

Reduksjon av tilstander

Eksempel:

Fjerner tilstand

G. Erstatte

hopp til **G** med

hopp til **E**

Nåværende tilstand	Inngang	Neste tilstand	Utgang
A	0	B	0
A	1	B	0
B	0	C	0
B	1	D	0
C	0	A	0
C	1	D	0
D	0	E	0
D	1	F	1
E	0	A	0
E	1	F	1
F	0	E	0
F	1	F	1

Reduksjon av tilstander

	Nåværende tilstand	Inngang	Neste tilstand	Utgang
Eksempel:	A	0	B	0
	A	1	B	0
	B	0	C	0
	B	1	D	0
Nå er tilstand F	C	0	A	0
lik tilstand D	C	1	D	0
	D	0	E	0
	D	1	F	1
Fjerner tilstand	E	0	A	0
F	E	1	F	1
	F	0	E	0
	F	1	F	1

Reduksjon av tilstander

Nåværende tilstand	Inngang	Neste tilstand	Utgang
A	0	B	0
A	1	B	0
B	0	C	0
B	1	D	0
C	0	A	0
C	1	D	0
D	0	E	0
D	1	D	1
E	0	A	0
E	1	D	1

Eksempel:

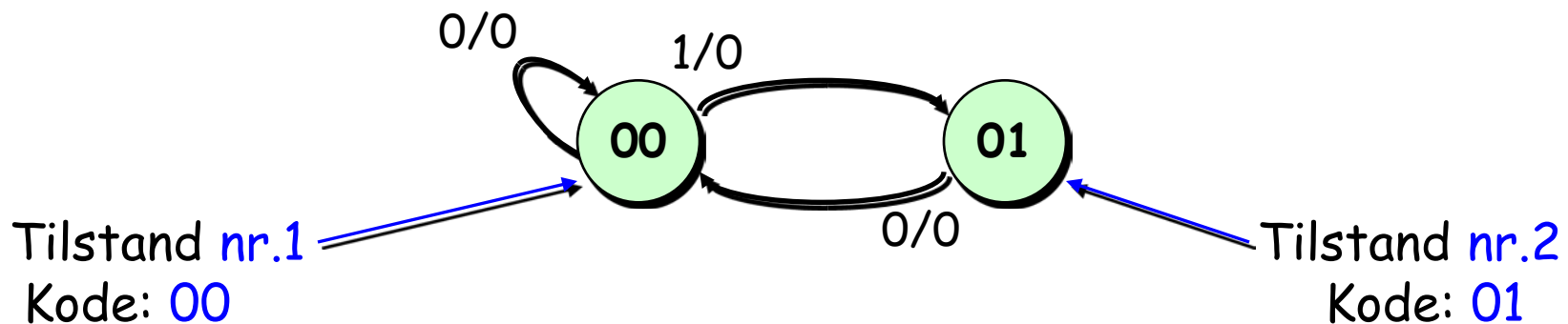
Har fjernet
tilstand F

Tilordning av tilstandskoder

I en tilstandsmaskin med M tilstander må hver tilstand tilordnes en kode basert på minimum N bit der $2^N \geq M$

Kompleksiteten til den kombinatoriske delen avhenger av valg av tilstandskode

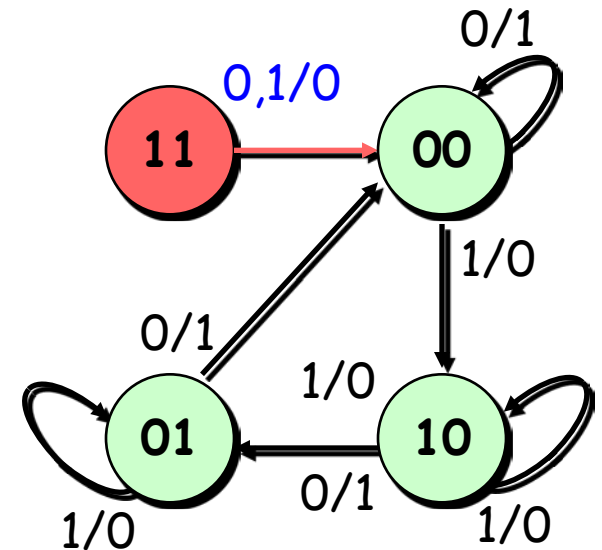
Anbefalt strategi for valg av kode: prøv-og-feil i tilstandsdiagrammet



Ubrukte tilstander

I en tilstandsmaskin med N flip-floppe vil det alltid finnes 2^N tilstander. Designer man for M tilstander der $M < 2^N$ vil det finnes ubrukte tilstander.

Problem: Under oppstart (power up) har man ikke full kontroll på hvilken tilstand man havner i først. Havner man i en ubrukt tilstand som ikke leder videre til de ønskede tilstandene vil systemet bli låst.



Løsning: Design systemet slik at alle ubrukte tilstander leder videre til en ønsket tilstand.

Generell designprosedyre basert på D flip-flops

- 1) Definer tilstandene, inngangene og utgangene
- 2) Velg tilstandskoder, og tegn tilstandsdiagram
- 3) Tegn tilstandstabell
- 4) Reduser antall tilstander hvis nødvendig
- 5) Bytt tilstandskoder hvis nødvendig for å forenkle
- 6) Finn de kombinatoriske funksjonene
- 7) Skjekk at ubrukte tilstander leder til ønskede tilstander
- 8) Tegn opp kretsen

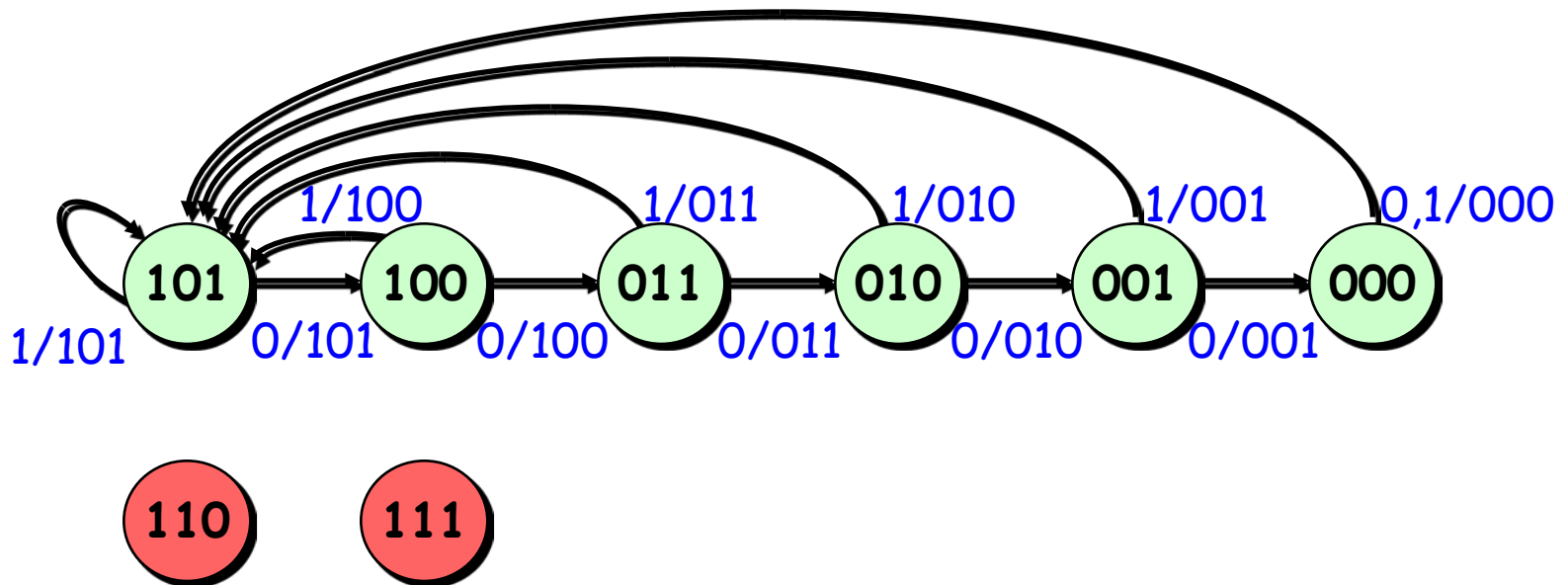
Design eksempel nr.4

Designer en teller som teller sekvensen 5,4,3,2,1,0. Etter 0 skal telleren gjenta sekvensen (telle rundt). Telleren skal kunne resettes til 5 med ett reset signal.

- 1) Velger en tilstand for hvert tall ut. Systemet har 1 reset inngang, og trenger 3 utganger for å representere tallene 5 til 0.
- 2) Velger tilstandskoder som direkte representerer tallene ut. Tallene ut blir gitt av tilstandene

Eksempel nr.4

2) Tegner tilstandsdiagram



Registrerer at vi har to ubrukte tilstander

Eksempel nr.4

3) Tegner tilstandstabell

4) Ingen reduksjons-
mulighet

5) Velger å ikke bytte
tilstandskoder da
utgangene i såfall må
omformes

Ubrukte
tilstander

Nåværende tilstand / utgang				Inngang	Neste tilstand		
Q_A	Q_B	Q_C		R	Q_A	Q_B	Q_C
0	0	0		0	1	0	1
0	0	0		1	1	0	1
0	0	1		0	0	0	0
0	0	1		1	1	0	1
0	1	0		0	0	0	1
0	1	0		1	1	0	1
0	1	1		0	0	1	0
0	1	1		1	1	0	1
1	0	0		0	0	1	1
1	0	0		1	1	0	1
1	0	1		0	1	0	0
1	0	1		1	1	0	1
1	1	0		0	X	X	X
1	1	0		1	X	X	X
1	1	1		0	X	X	X
1	1	1		1	X	X	X

M.H

Eksempel nr.4

- 6) Setter inn i karnaughdiagram og finner forenklede funksjoner

D_A

		$Q_C R$			
		00	01	11	10
$Q_A Q_B$	00	1	1	1	0
	01	0	1	1	0
	11	x	x	x	x
	10	0	1	1	1

D_B

		$Q_C R$			
		00	01	11	10
$Q_A Q_B$	00	0	0	0	0
	01	0	0	0	1
	11	x	x	x	x
	10	1	0	0	0

D_C

		$Q_C R$			
		00	01	11	10
$Q_A Q_B$	00	1	1	1	0
	01	1	1	1	0
	11	x	x	x	x
	10	1	1	1	0

$$D_A = R + Q_A' Q_B' Q_C' + Q_A Q_C$$

$$D_B = Q_B Q_C R' + Q_A Q_C' R'$$

$$D_C = Q_C' + R$$

Eksempel nr.4

- 6) Sjekk at ubrukte tilstander leder til ønskede tilstander - ok

Nåværende tilstand / utgang Inngang				Neste tilstand		
Q_A	Q_B	Q_C	R	Q_A	Q_B	Q_C
1	1	0	0	0	1	1
1	1	0	1	1	0	1
1	1	1	0	1	1	0
1	1	1	1	1	0	1

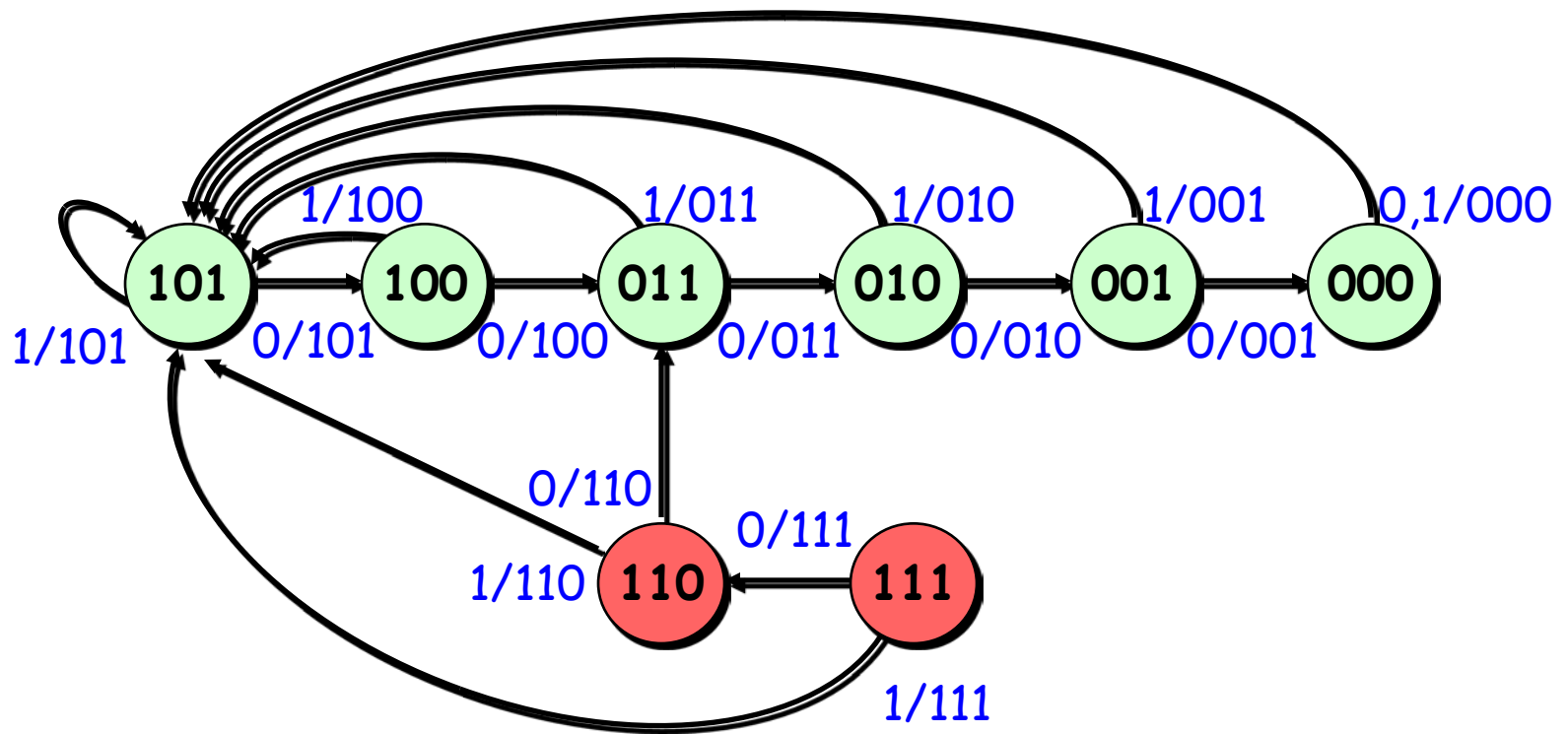
$$D_A = R + Q_A' Q_B' Q_C' + Q_A Q_C$$

$$D_B = Q_B Q_C R' + Q_A Q_C' R'$$

$$D_C = Q_C' + R$$

Eksempel nr.4

- 6) Alle ubrukte tilstander leder til ønskede tilstander, viser med diagram

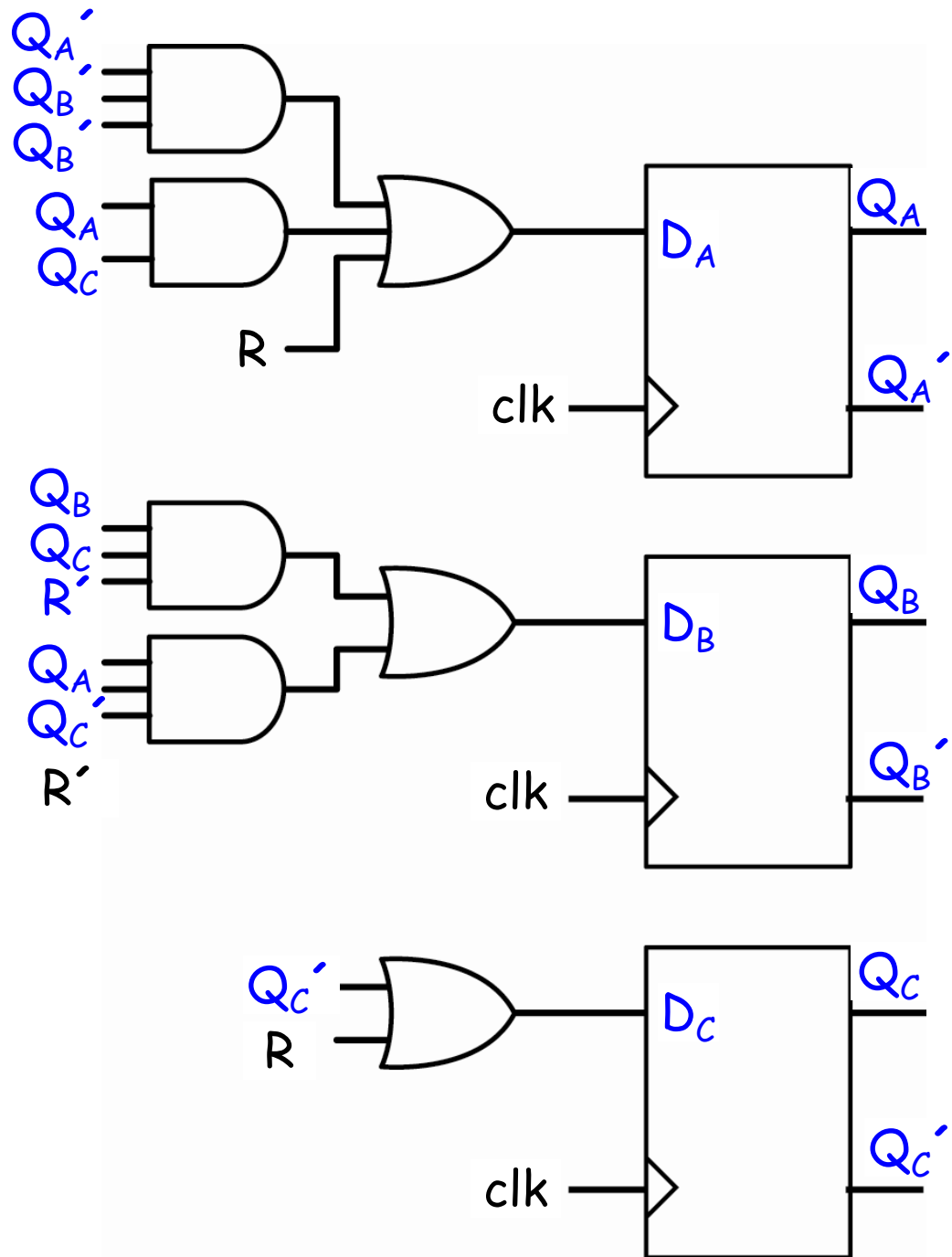


Eksempel nr.4

7) Tegner opp
krets

Q_A , Q_B og Q_C blir
tellerens utganger

Telleren resettes
ved å sette $R=1$



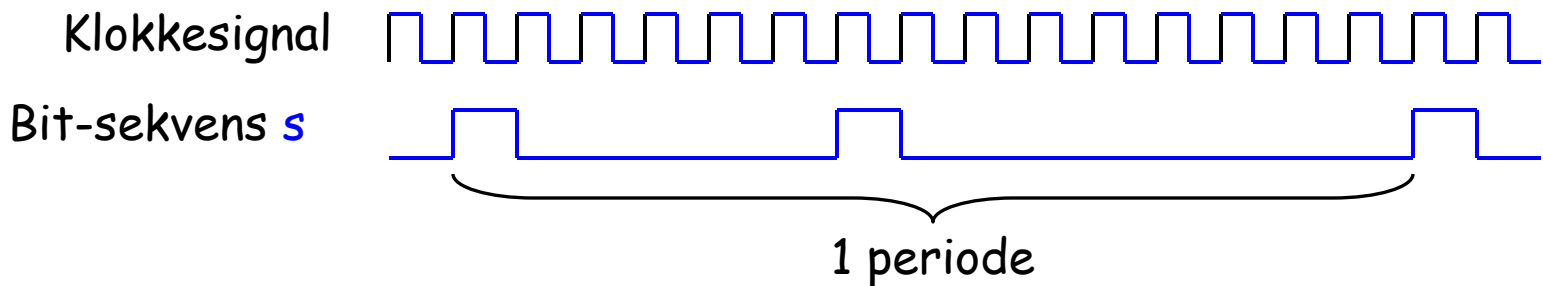
M.H

Eksempel nr.5 - trafikklys

Ønsker å bruker tilstandsmaskin for å styre trafikklys

Krysset har to vanlige trafikklys A og B . Disse styres med de binære signalene R_A , G_A , Gr_A samt R_B , G_B , Gr_B . Setter man G_A til "1" lyser det grønt i lys A osv.

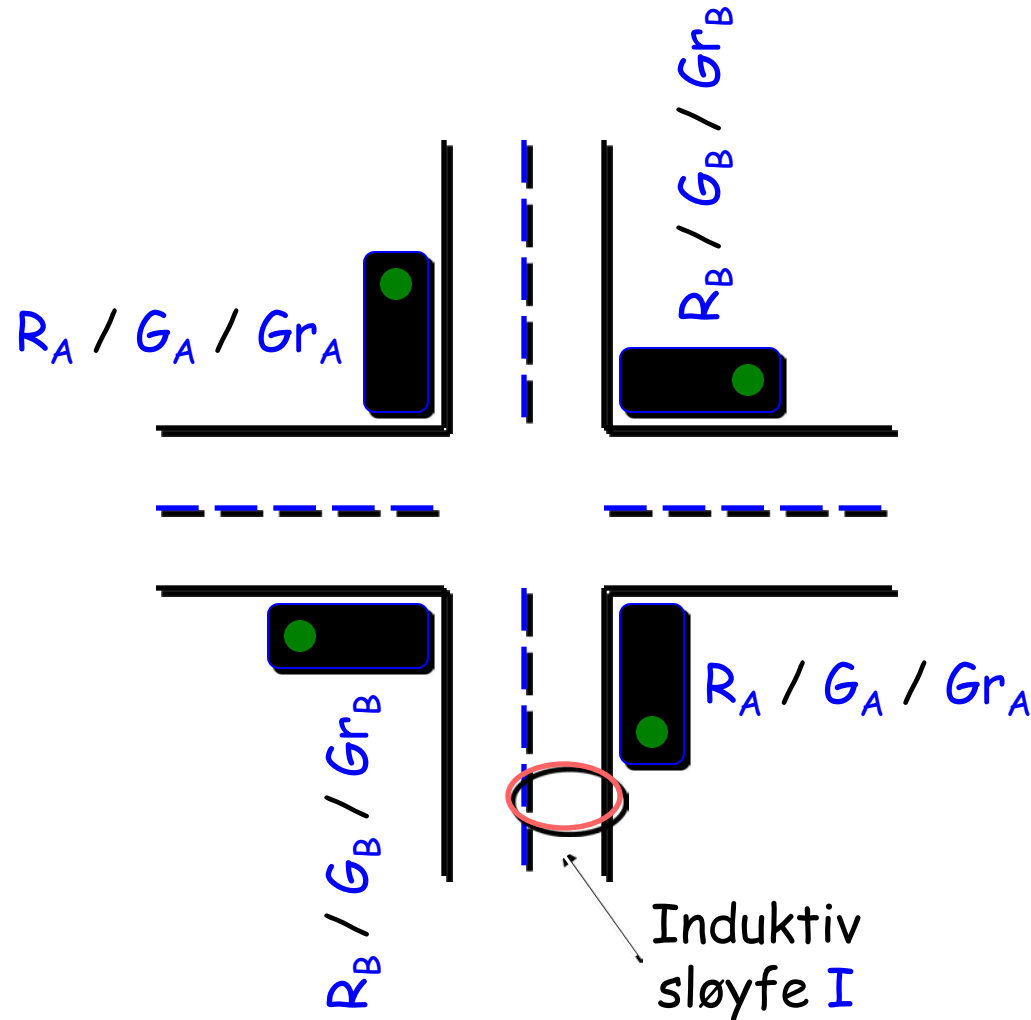
For å generere lyssekvensene bruker vi en repeterende bit-sekvens **s** som vist under. Avstanden mellom "1"er pulsene gir intervallene mellom skifte fra grønt i lys **A** til grønt i lys **B** og motsatt.



Eksempel nr.5 - trafikklys

Systemet har en induktiv sensor i bakken som registrerer biler den ene veien. Bil over sensoren gir $I=1$ ellers har vi $I=0$

Vi ønsker at bil registrert av sensoren skal gi grønt lys i A så fort som mulig



Eksempel nr.5

1,2) Velger følgende forenklede tilstander:

00 - Grønt lys i A, rødt lys i B

01 - Gult lys i A og B. Skifter mot grønt lys i B.

10 - Rødt lys A, grønt lys i B

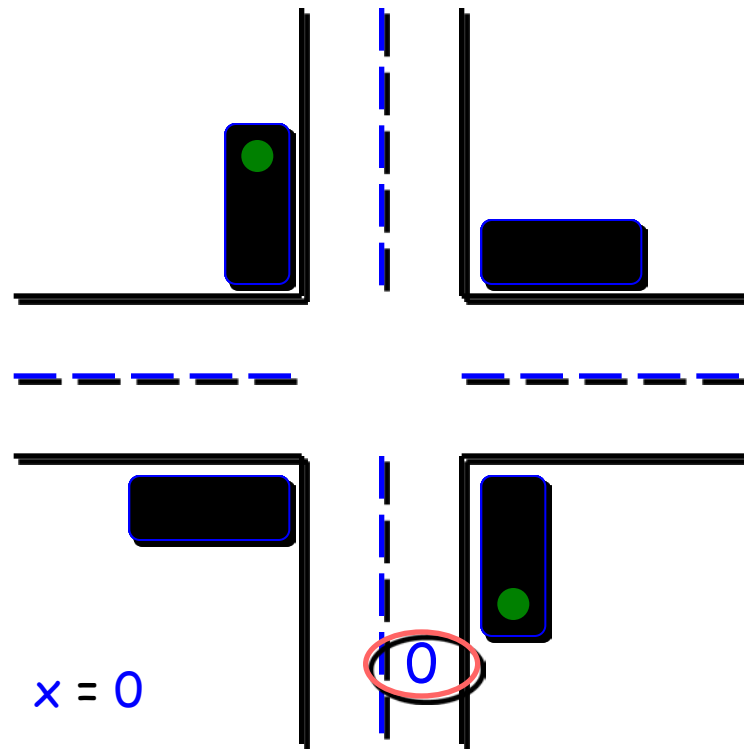
11 - Gult lys i A og B. Skifter mot grønt lys i A.

Innganger: s, I

Utganger: $R_A, G_A, Gr_A, R_B, G_B, Gr_B$

Lar utgangene kun være en funksjon av tilstanden

Eksempel nr.5

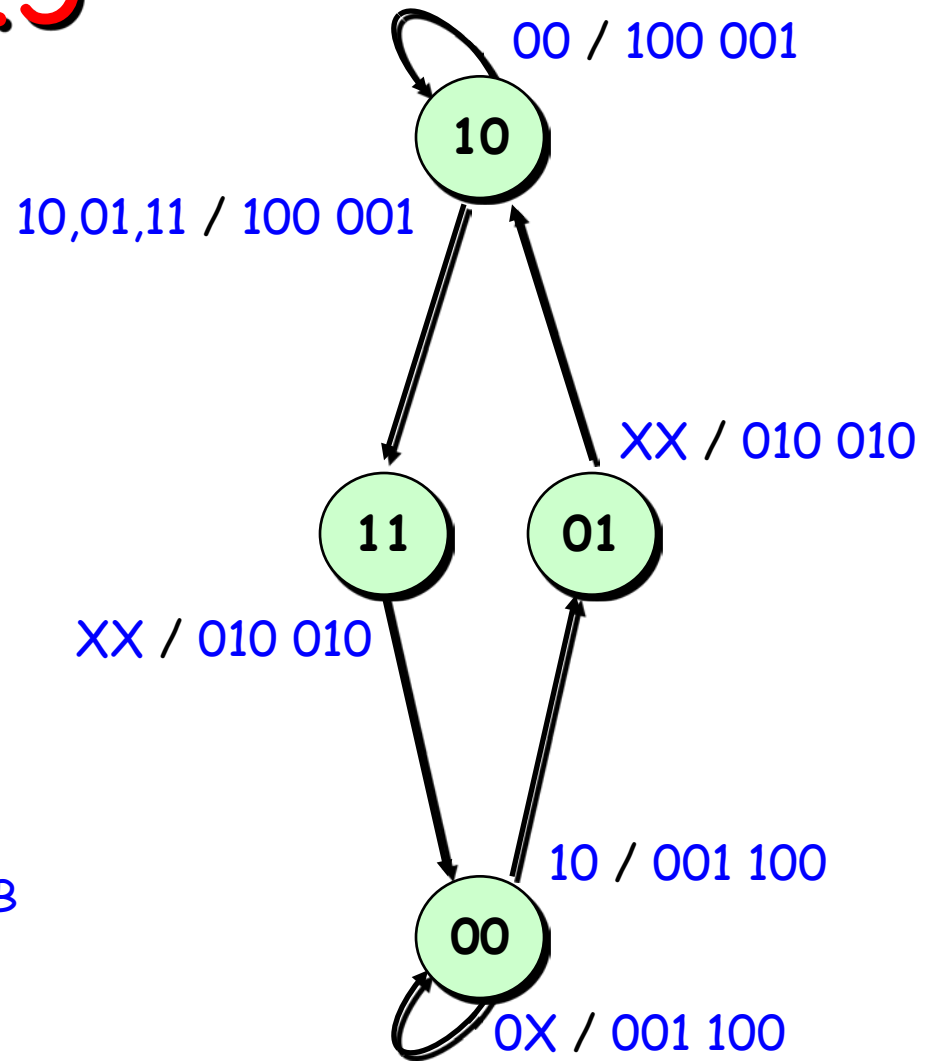


Eksempel nr.5

2) Tilstandsdiagram

X - don't care

$sI / R_A G_A Gr_A \quad R_B G_B Gr_B$



6) Finner kombinatoriske funksjoner

$$D_A = Q_A \oplus Q_B$$

$$D_B = Q_A Q_B' I + Q_B' s I'$$

$$R_A = Q_A Q_B'$$

$$G_A = Q_B$$

$$Gr_A = Q_A' Q_B'$$

$$R_B = Gr_A$$

$$G_B = G_A$$

$$Gr_B = R_A$$

Nåværende tilstand Innganger				Neste tilstand Utganger							
				Q_A	Q_B	R_A	G_A	Gr_A	R_B	G_B	Gr_B
I'	0	0	0	0	0	0	0	1	1	0	0
	0	0	0	1	0	0	0	1	1	0	0
	0	0	1	0	0	1	0	1	1	0	0
	0	0	1	1	0	0	0	1	1	0	0
	0	1	0	0	1	0	0	1	0	1	0
	0	1	0	1	1	0	0	1	0	1	0
	0	1	1	0	1	0	0	1	0	1	0
	0	1	1	1	1	0	0	1	0	1	0
	1	0	0	0	1	0	1	0	0	0	1
	1	0	0	1	1	1	1	0	0	0	1
	1	0	1	0	1	1	1	0	0	0	1
	1	0	1	1	1	1	1	0	0	0	1
	1	1	0	0	0	0	0	1	0	0	1
	1	1	0	1	0	0	0	1	0	0	1
	1	1	1	0	0	0	0	1	0	0	1
	1	1	1	1	0	0	0	1	0	0	1

M.H

Eksempel nr.5

7) Tegner opp krets

$$D_A = Q_A \oplus Q_B$$

$$D_B = Q_A Q_B' I + Q_B' s I'$$

$$R_A = Q_A Q_B'$$

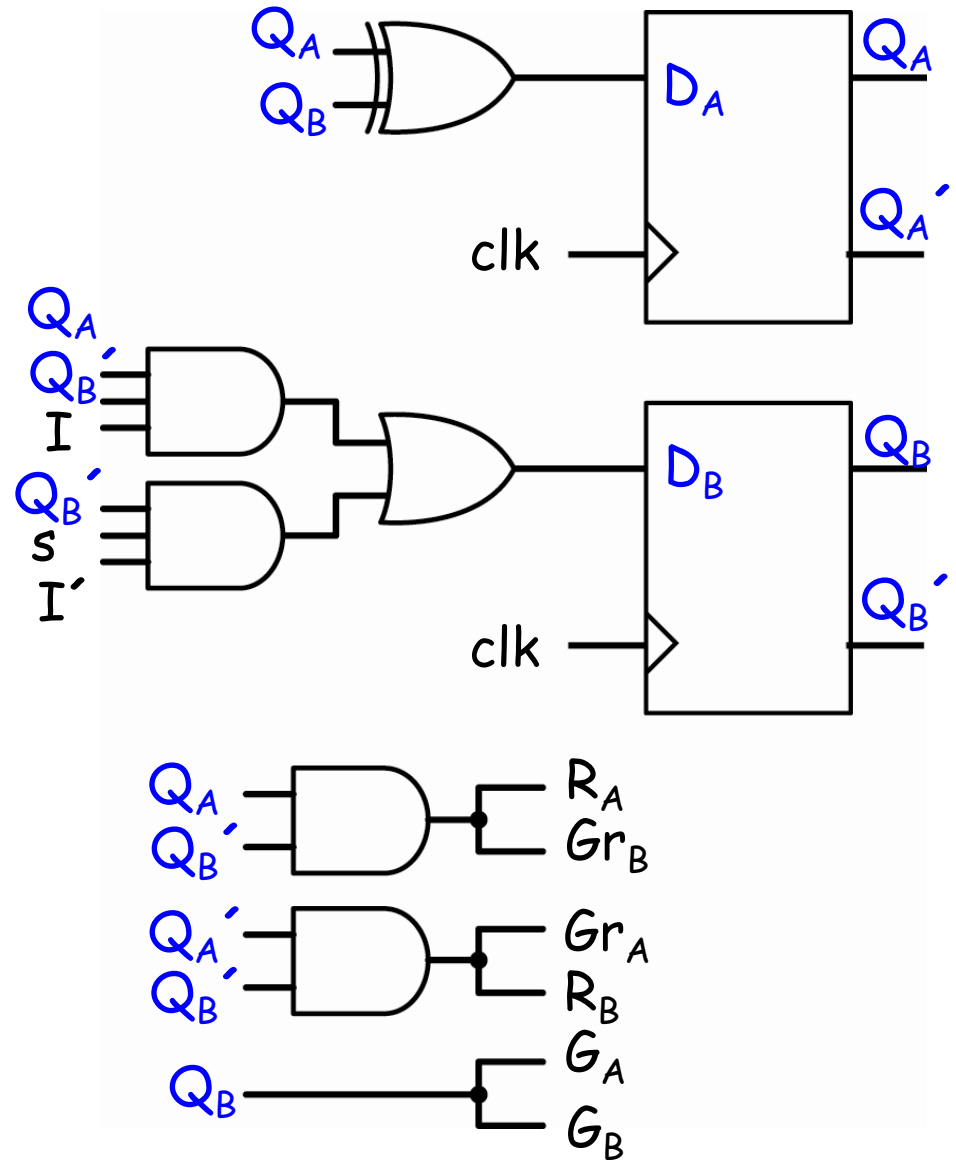
$$G_A = Q_B$$

$$Gr_A = Q_A' Q_B'$$

$$R_B = Gr_A$$

$$G_B = G_A$$

$$Gr_B = R_A$$



Oppsummering

- Tilstandsmaskin
- Tilstandstabell
- Tilstandsdiagram
- Analyse av D flip-flop basert tilstandsmaskin
- Reduksjon av antall tilstander
- Tilordning av tilstandskoder
- Designprosedyre for tilstandsmaskin basert på D flip-flops