

UiO : **Institutt for informatikk**

Det matematisk-naturvitenskapelige fakultet

INF1400

Tilstandsmaskin



Læringsutbytte

- Kunnskapsmål:
 - Kunnskap om hva en tilstandsmaskin er
- Ferdighetsmål:
 - Kunne implementere en tilstandsmaskin
 - Kunne forenkle antall tilstander
 - Kunne analysere en tilstandsmaskin
- Generelle kompetansemål:
 - Kunne forklare design prosedyre for tilstandsmaskiner med D-flip-floper

Hovedpunkter

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

Tilstandsmaskin

- Engelsk: Finite State Machine
- Tilstandsmaskiner er en metode til å beskrive systemer med logisk og dynamisk (tidsmessig) oppførsel.
- Brukes mye innen:
 - Logiske/digitale styresystemer
 - Sanntidssystemer
 - Telekommunikasjon
 - Kompilatorteknikk
 - Digitalteknikk

Tilstandsmaskin

Modellen av en tilstandsmaskin består av:

Tilstandsmaskin

En tilstandsmaskin er et

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

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

Sentrale begreper for tilstandsmaskin

Tilstand:

- er et begrep som benyttes til å beskrive systemets status / tilstand.
- er et verdisett/attributter som beskriver systemets egenskaper.

- **Hendelser:**

- er et begrep som benyttes om innganger/påvirkninger på systemet
- kan beskrives som en plutselig og kortvarig påvirkning av systemet.

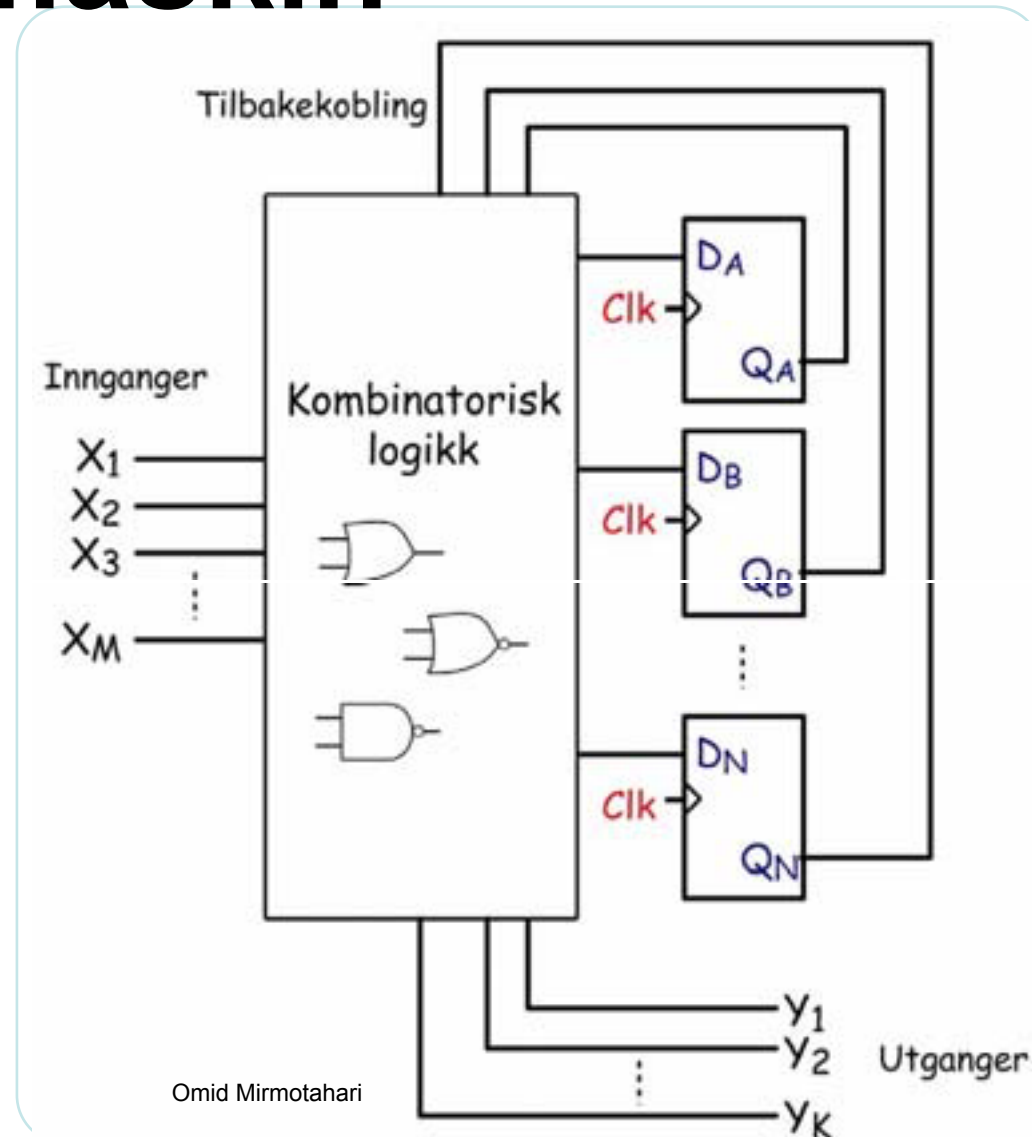
- **Aksjoner:**

- er det som kommer ut av systemet. Det vil si resultatet
- er en respons på en hendelse

Tilstandsmaskin

Generell tilstands-
maskin basert på D
flip-flops

Utgangssignalene er en
funksjon av **nåværende
tilstand** pluss evt.
inngangsverdier



Tilstandsdiagrammer

- For å visualisere oppførselen til systemer brukes gjerne tilstandsdiagrammer
 - Sirkler angir tilstander
 - Piler angir tilstandsending
 - Hendelse og aksjoner settes over piler som angir tilstandsendingen

Eksempel: Brusautomat

Eksempel: Brusautomat

Eksempel: Brusautomat

- Hendelser:

{

- Aksjoner:

{

- Tilstander:

{

Implementasjon og kretsdesign

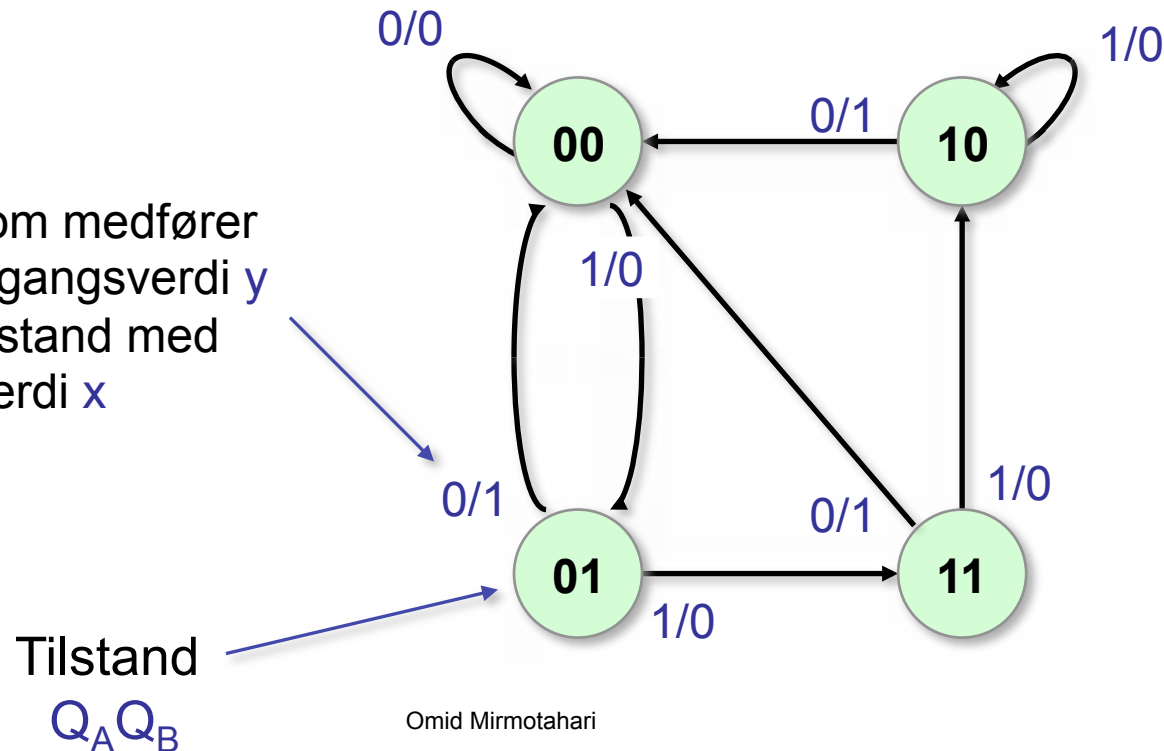
- Nå skal vi gjennom en rekke eksempler for implementasjon av tilstandsmaskiner.
- Vi skal lære om tilstandstabell.
- Vi skal se på noen forenklinger med hensyn på reduksjon av tilstander
- Vi skal ta hensyn til ubrukte tilstander

Tilstandsdiagram

Tilstandsdiagram = grafisk illustrasjon av egenskapene til en tilstandsmaskin

Eksempel nr.1:

Inngangsverdi x som medfører ny tilstand, samt utgangsverdi y for opprinnelig tilstand med inngangsverdi x
 x / y



Tilstandstabell

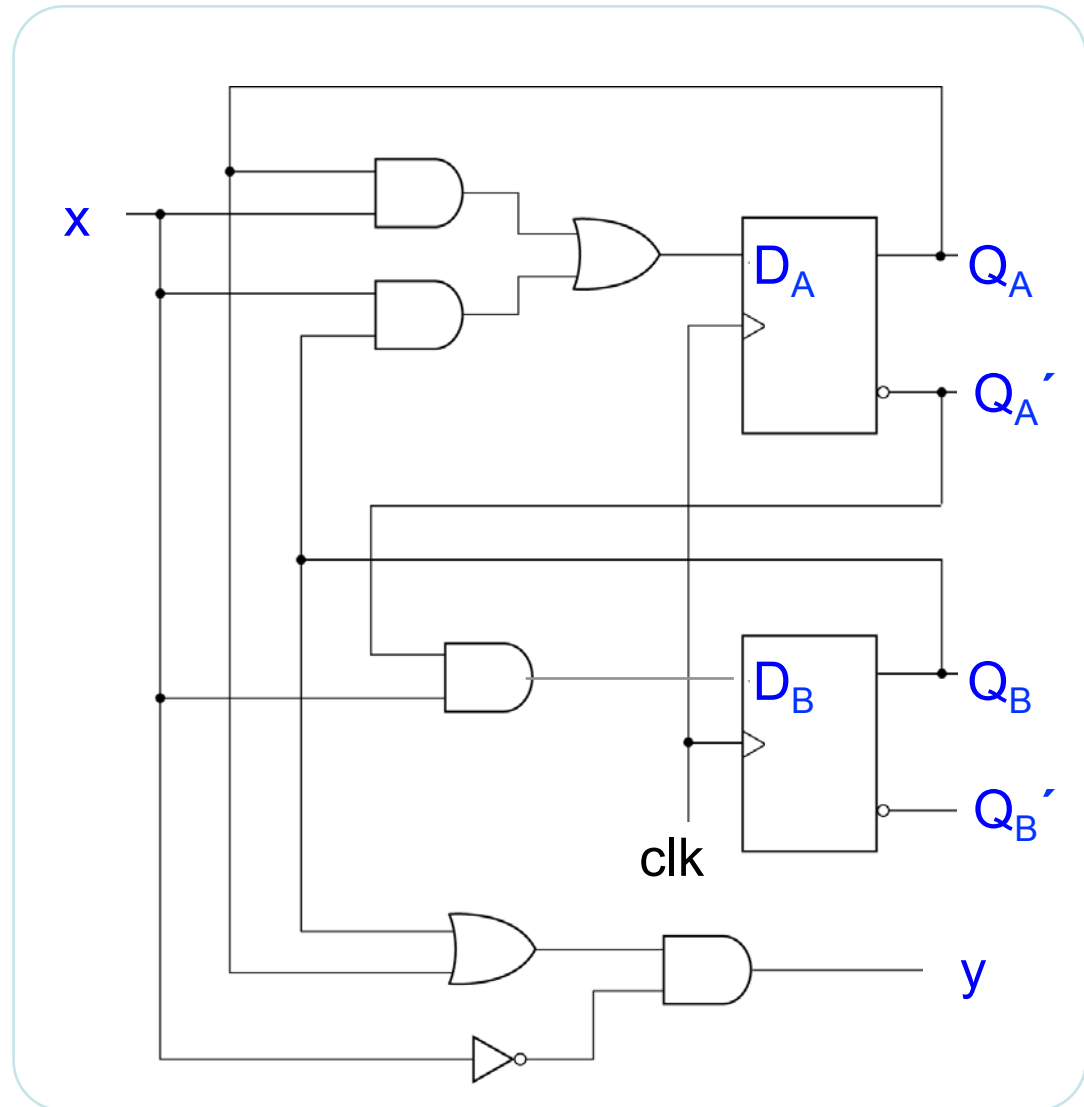
Tilstandstabell = sannhetstabell for tilstandsmaskin

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

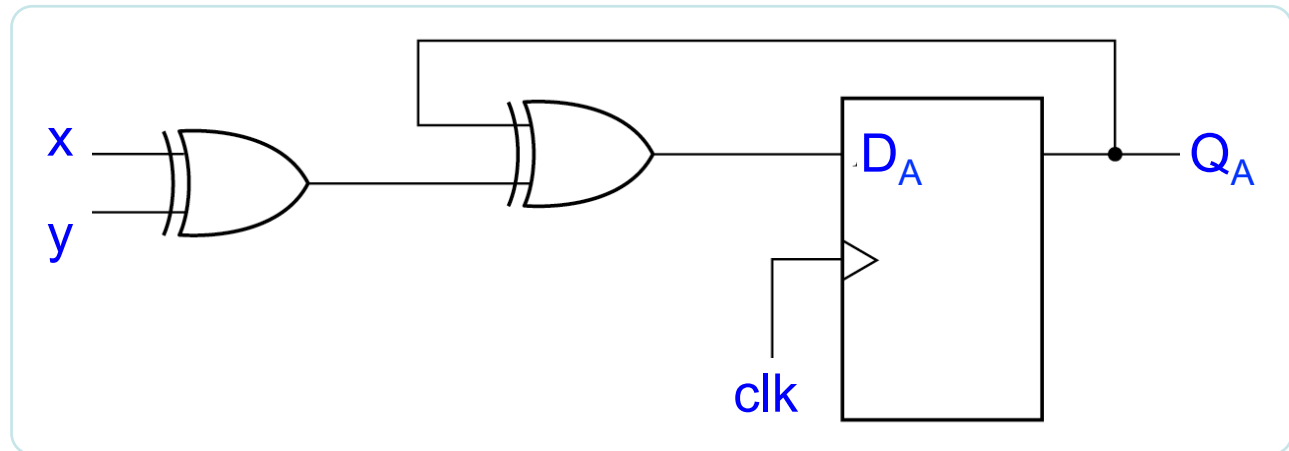
Nåværende tilstand			Inngang	Neste tilstand		Utgang for nåværende tilstand
Q_A	Q_B	x		Q_A	Q_B	y
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	0	1
0	1	1	0	1	1	0
1	0	0	0	0	0	1
1	0	1	0	1	0	0
1	1	0	0	0	0	1
1	1	1	0	1	0	0

Eksempel nr.1

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



Eksempel nr.2



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

Nåværende tilstand Q_A	Innganger		Neste tilstand Q_A
	x	y	
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Eksempel nr.2

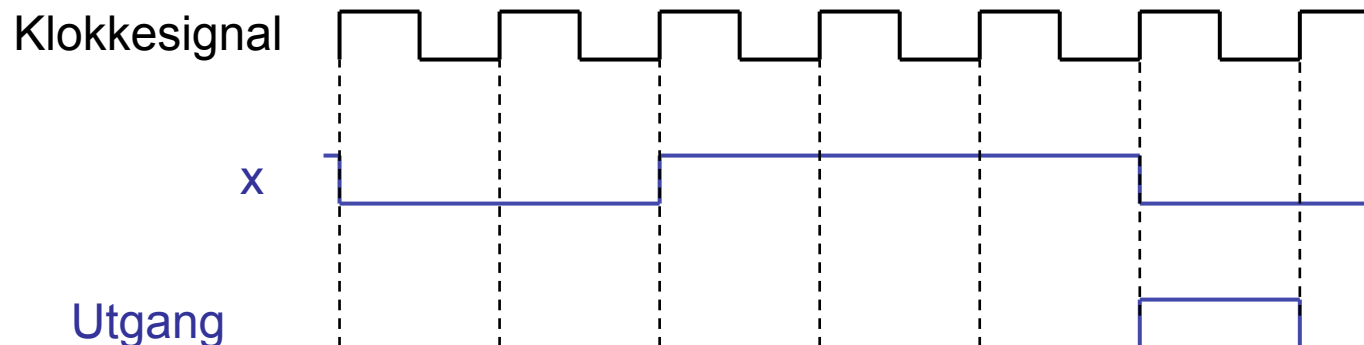
Tilstandsdiagram

*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

$D_A =$

$D_B =$

$y =$

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

Eksempel nr.3

Forenkler uttrykkene
med Karnaugh-diagram

$$D_A = Q_A X + Q_B X$$

$$D_B = Q_A X + Q_B \bar{X}$$

$$y = Q_A Q_B$$

x

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

		Nåværende tilstand		Neste tilstand		
		Inngang		Utgang		
Eksempel:		A	0	B	0	
		A	1	B	0	
		B	0	C	0	
		B	1	D	0	
	Tilstand G er lik tilstand E		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

	Nåværende tilstand		Neste tilstand	
	Inngang		Utgang	
Eksempel: Fjerner tilstand G. Erstatter hopp til G med hopp til E	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		Neste tilstand	
	Inngang		Utgang	
	A	0	B	0
Eksempel:	A	1	B	0
	B	0	C	0
	B	1	D	0
Nå er tilstand F lik tilstand D	C	0	A	0
	C	1	D	0
	D	0	E	0
	D	1	F	1
Fjerner tilstand F	E	0	A	0
	E	1	F	1
	F	0	E	0
	F	1	F	1

Reduksjon av tilstander

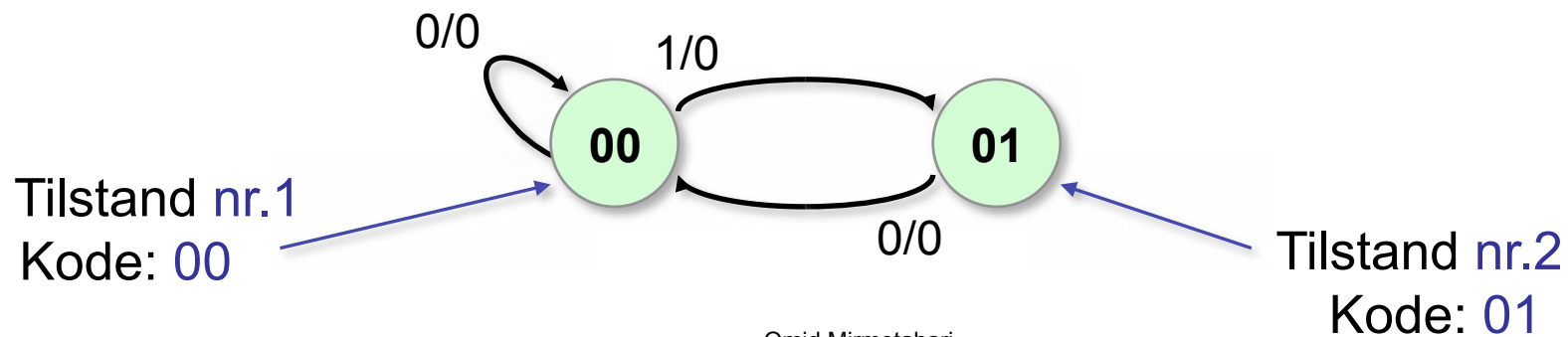
	Nåværende tilstand		Neste tilstand	
	Inngang		Utgang	
Eksempel: Har fjernet tilstand F	A	0	B	0
	A	1	B	0
	B	0	C	0
	B	1	D	0
	C	0	A	0
	C	1	D	0
	D	0	E	0
	D	1	D	1
	E	0	A	0
E	1	D	1	

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

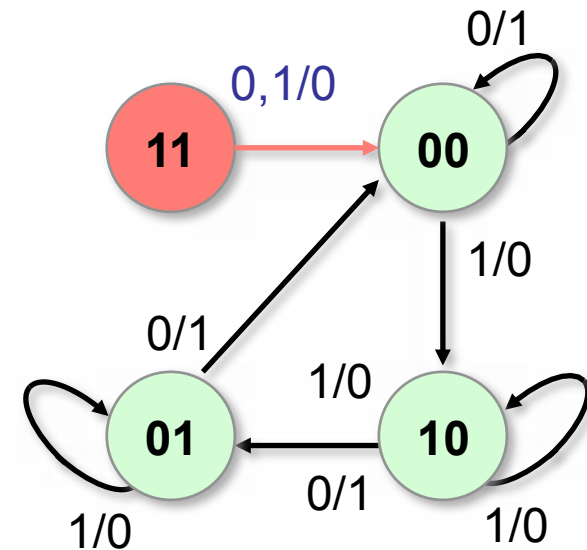
Anbefalt strategi for valg av kode: prøv-og-feil i tilstandsdiagrammet



Ubrukte tilstander

I en tilstandsmaskin med N flip-flopper 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) Sjekk 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

Eksempel nr.4

- 3) Tegner tilstandstabell
- 4) Ingen reduksjonsmulighet
- 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	0	0	0	1	0	1
0 0 0	0	0	0	1	1	0	1
0 0 1	0	0	1	0	0	0	0
0 0 1	0	0	1	1	1	0	1
0 1 0	0	1	0	0	0	0	1
0 1 0	0	1	0	1	1	0	1
0 1 1	0	1	1	0	0	1	0
0 1 1	0	1	1	1	1	0	1
1 0 0	1	0	0	0	0	1	1
1 0 0	1	0	0	1	1	0	1
1 0 1	1	0	1	0	1	0	0
1 0 1	1	0	1	1	1	0	1
1 1 0	1	1	0	0	X	X	X
1 1 0	1	1	0	1	X	X	X
1 1 1	1	1	1	0	X	X	X
1 1 1	1	1	1	1	X	X	X

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) Sjekker 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

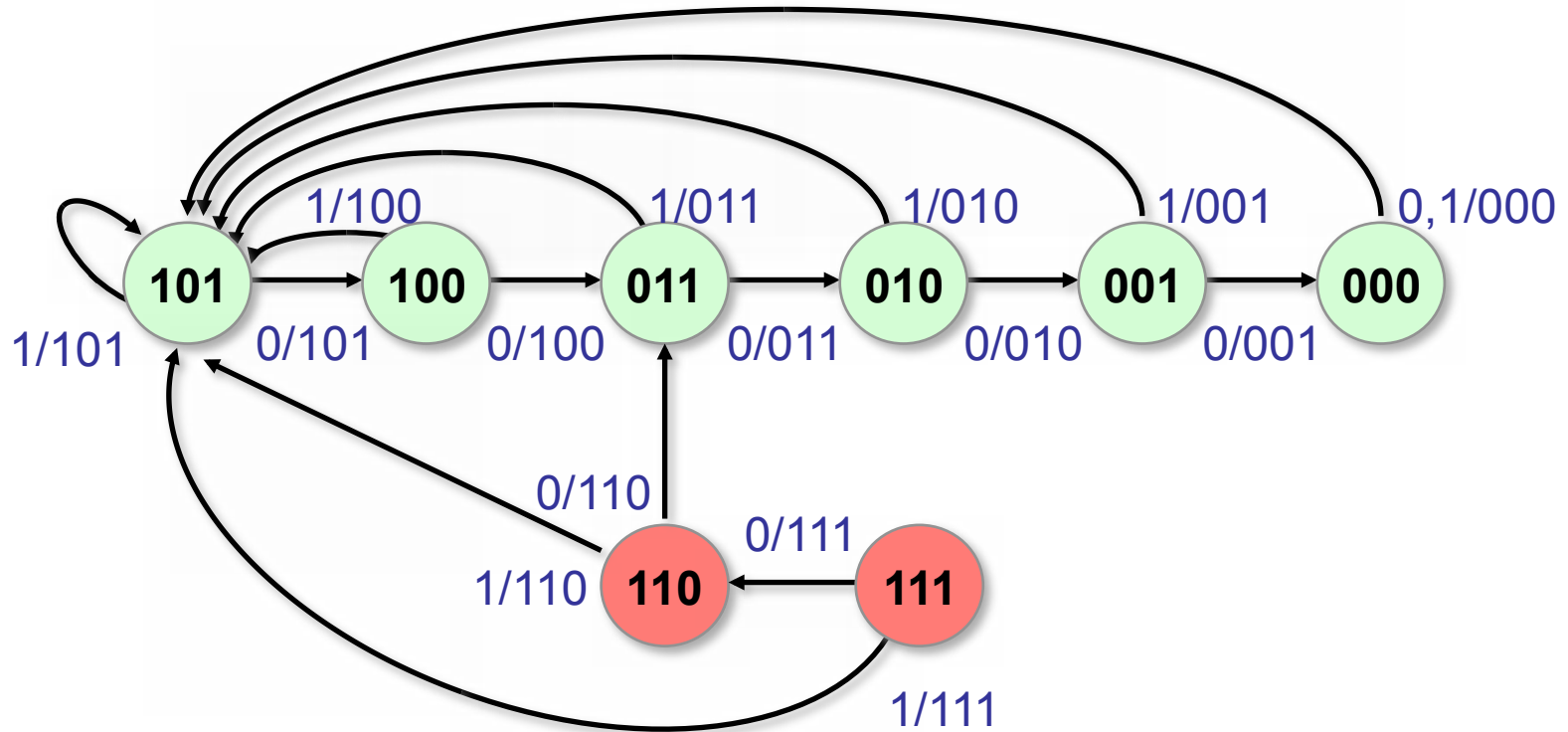
$$D_A = R + Q_A'Q_B'Q_C' + Q_A Q_C$$

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

$$D_C = Q_C' + R$$

Eksempel nr.4

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

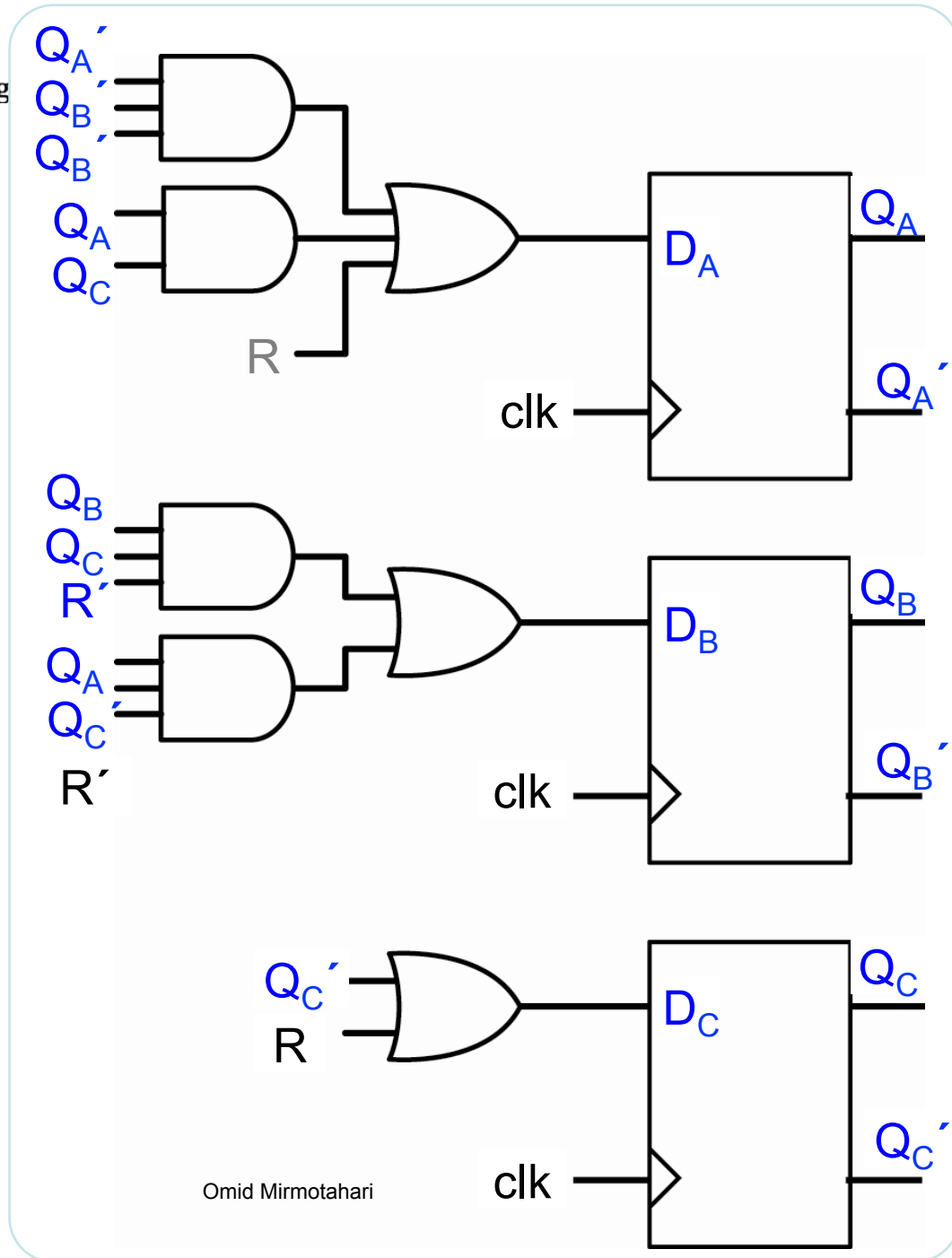


Eksempel nr.4

7) Tegner opp
krets

Q_A , Q_B og Q_C blir
tellerens utganger

Telleren resettes ved å
sette $R=1$

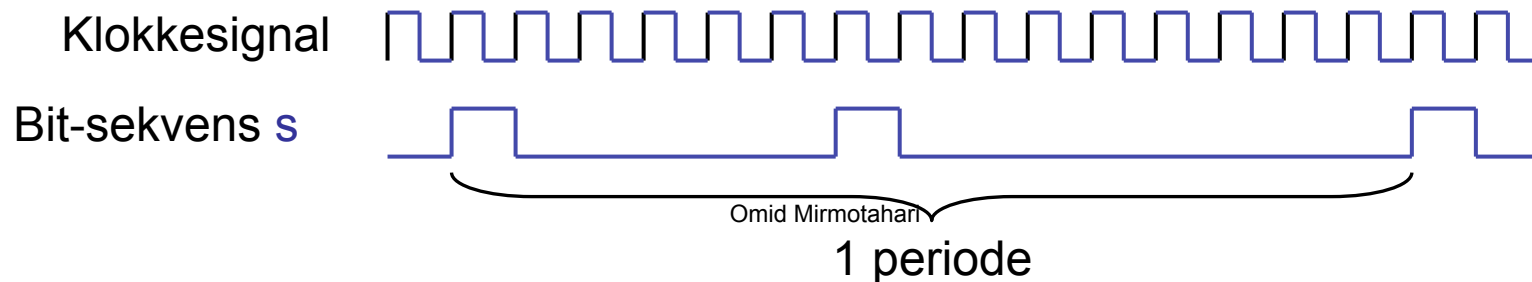


Eksempel nr.5 - trafikklys

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.

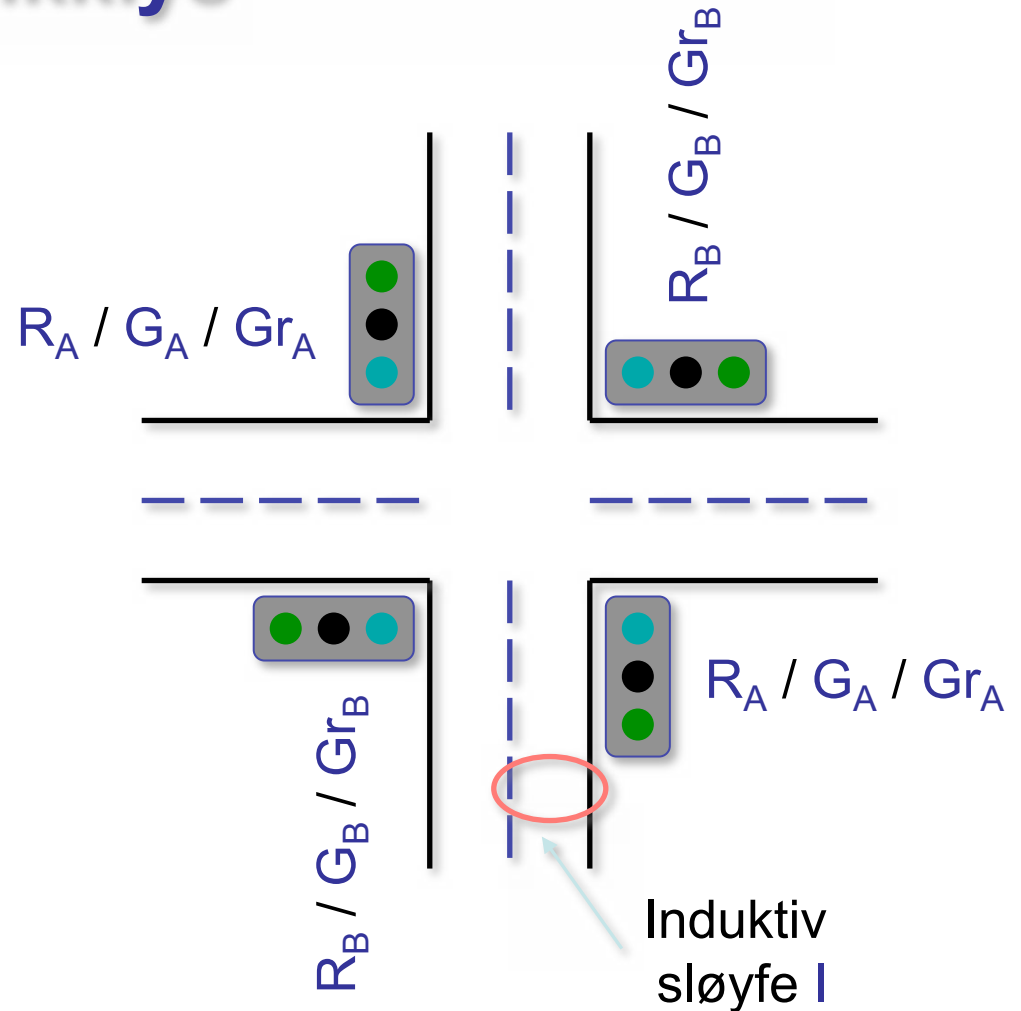
For å generere lyssekvensene bruker vi en repeterende bit-sekvens **s** som vist under. Avstanden mellom "1"er pulsene gir intervallene mellom skifte fra grønt i lys **A** til grønt i lys **B** og motsatt.



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 forenklete 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, l

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

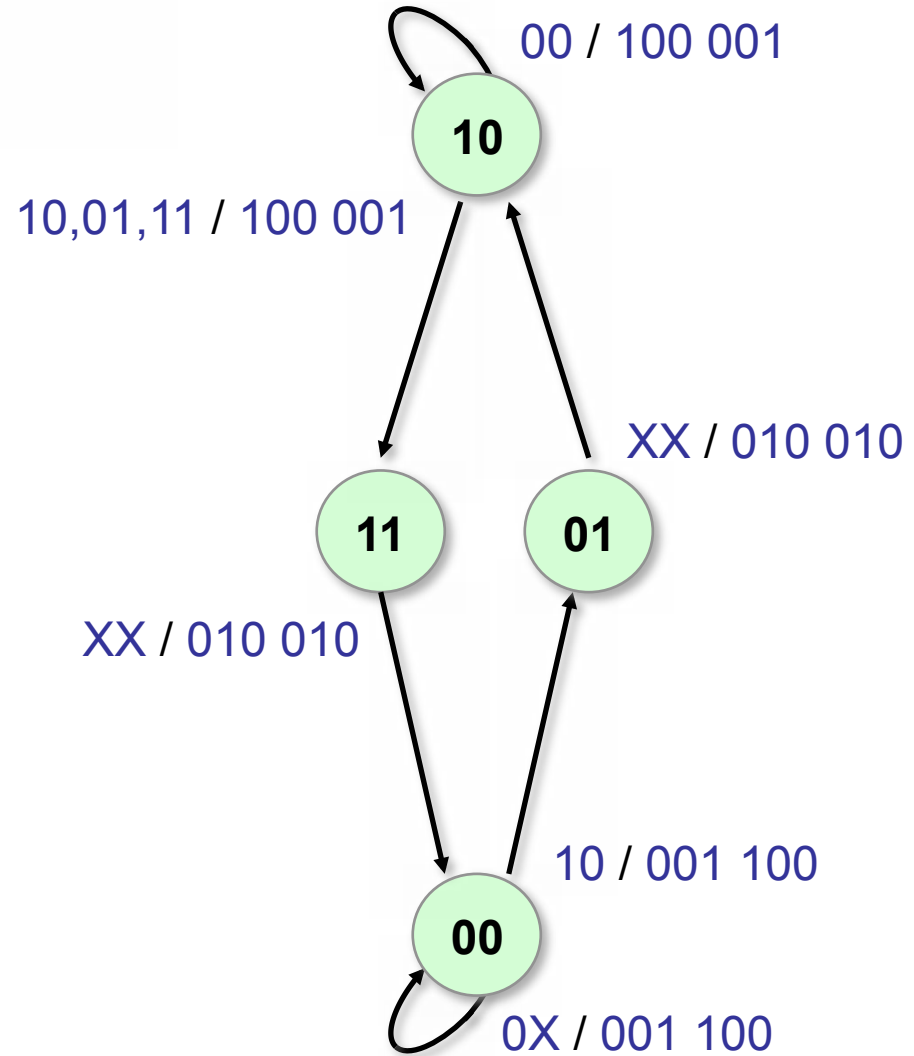
Lar utgangene kun være en funksjon av tilstanden

Eksempel nr.5

2) Tilstandsdiagram

X – don't care

sI / R_AG_AGr_A R_BG_BGr_B



Nåværende tilstand

Neste tilstand

Utganger

6) Finner kombinatoriske funksjoner

$$D_A = Q_A \oplus Q_B$$

$$D_B = Q_A Q_B '1 + Q_B 's1'$$

$$R_A = Q_A Q_B '$$

$$G_A = Q_B$$

$$Gr_A = Q_A 'Q_B '$$

$$R_B = Gr_A$$

$$G_B = G_A$$

$$Gr_B = R_A$$

Nåværende tilstand				Neste tilstand							
Innganger				Utganger							
Q_A	Q_B	s	l	Q_A	Q_B	R_A	G_A	Gr_A	R_B	G_B	Gr_B
0	0	0	0	0	0	0	0	1	1	0	0
0	0	0	1	0	0	0	0	1	1	0	0
0	0	1	0	0	1	0	0	1	1	0	0
0	0	1	1	0	0	0	0	1	1	0	0
0	1	0	0	1	0	0	1	0	0	1	0
0	1	0	1	1	0	0	1	0	0	1	0
0	1	1	0	1	0	0	1	0	0	1	0
0	1	1	1	1	0	0	1	0	0	1	0
1	0	0	0	1	0	1	0	0	0	0	1
1	0	0	1	1	1	1	0	0	0	0	1
1	0	1	0	1	1	1	0	0	0	0	1
1	0	1	1	1	1	1	0	0	0	0	1
1	1	0	0	0	0	0	1	0	0	1	0
1	1	0	1	0	0	0	1	0	0	1	0
1	1	1	0	1	0	0	1	0	0	1	0
1	1	1	1	0	0	0	1	0	0	1	0
1	1	1	1	1	0	0	1	0	0	1	0

Eksempel nr.5

7) Tegner opp krets

$$D_A = Q_A \oplus Q_B$$

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

$$R_A = Q_A Q_B'$$

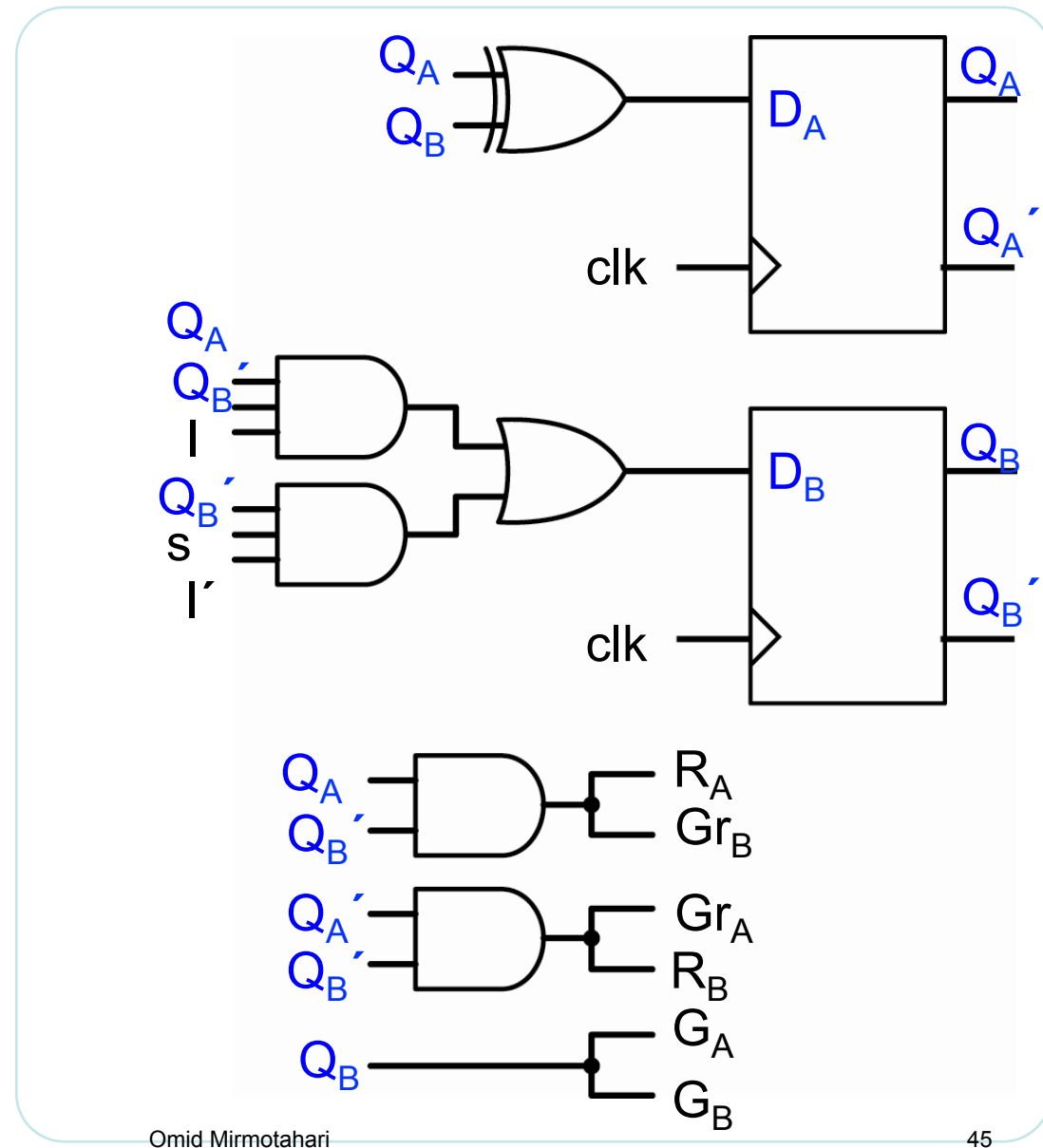
$$G_A = Q_B$$

$$Gr_A = Q_A' Q_B'$$

$$R_B = Gr_A$$

$$G_B = G_A$$

$$Gr_B = R_A$$



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