

UiO : **Institutt for informatikk**

Det matematisk-naturvitenskapelige fakultet

**INF1400**

**Sekvensiell logikk del 1**



# Læringsutbytte

- Kunnskapsmål:
  - Kunnskap om hva sekvensiell logikk er
  - Kunnskap om hva låsekretser er
  - Kunnskap om forskjellen mellom en latch og flip-flop
- Ferdighetsmål:
  - Kunne implementere en latch
  - Kunne implementere en flip-flop
- Generelle kompetansemål:
  - Kunne forklare forskjellen mellom sekvensiell logikk og kombinatorisk logikk
  - Kunne redegjøre for bruksområder for låsekretser

# Hovedpunkter

- Låsekretser (latch'er)
  - SR latch med NOR-porter
  - S'R' latch med NAND-porter
  - D-latch
- Flip-flop
  - Master-slave D-flip-flop
  - JK flip-flop
  - T-flip-flop



# Sekvensiell logikk

## Eksempel - låsekrets

Et eksempel på et vanlig behov innen digital design.

Ønsker å ha en to-inputs krets med følgende egenskaper:

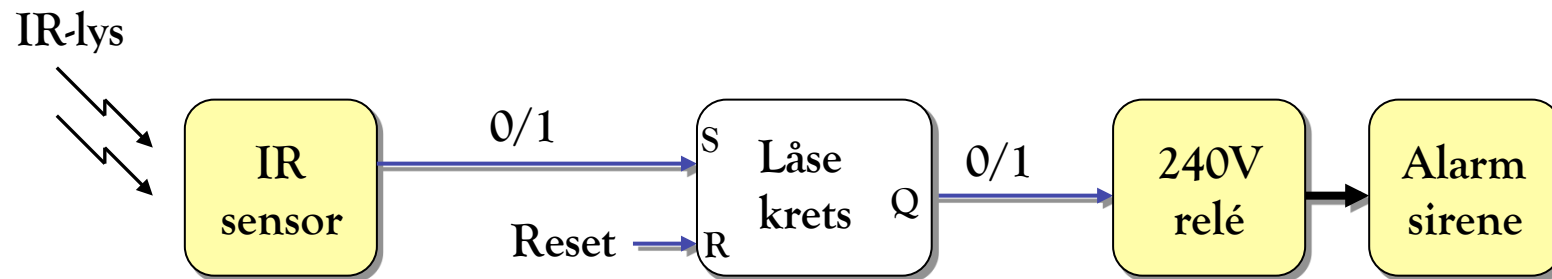
- 1) Kretsen skal sette utgangen Q til "1" hvis den får "1" på inngangen S. Når inngangen S går tilbake til "0" skal utgangen forbli "1"
- 2) Kretsen skal resette utgangen til "0" hvis den får "1" på inngangen R. Når inngangen R går tilbake til "0" skal utgangen forbli på "0".



# Praktiske eksempler

Logikk som behandler signaler fra fysiske sensorer:

For eksempel varmfølende persondetektor:

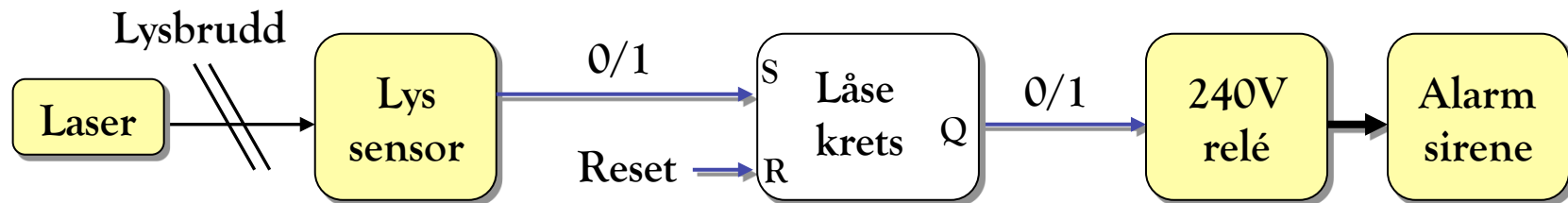


Når IR-lyset varierer mottar logikken et "ras" av kortvarige "1"er pulser (msek). Logikken skal sette sirenen permanent på på første mottatte puls



## Praktiske eksempler 2

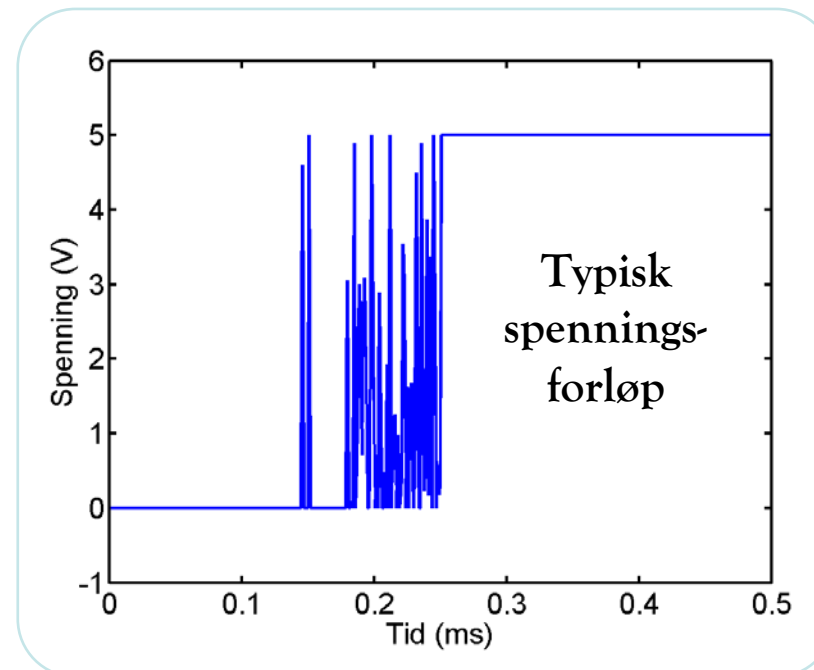
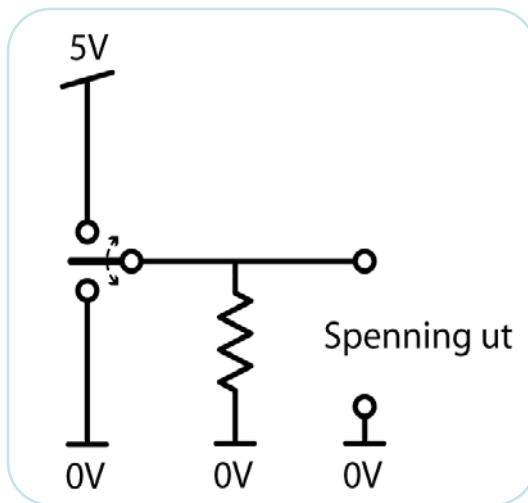
- Logikk som behandler signaler fra fysiske sensorer:
  - Laserbasert tyveridetektor



Når laserlyset blir brutt mottar logikken en eller flere '1' er pulser. Logikken skal sette sirenen permanent på første mottatte puls.

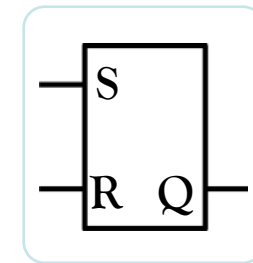
# Praktiske eksempler

- Kontaktprelling fra mekanisk bryter.
- Mekaniske brytere gir ikke “rene” logiske nivå ut i overgangsfasen. Slike signaler må ofte “renses” ved bruk av låsekretser.



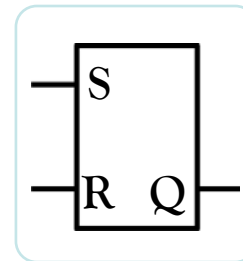
## SR-latch – funksjonell beskrivelse

- 1) Kretsen skal sette Q til "1" hvis den får "1" på inngang S. Når inngang S går tilbake til "0" skal Q forbli på "1"
- 2) Kretsen skal resette Q til "0" når den får "1" på inngang R. Når inngang R går tilbake til "0" skal Q forbli på "0"
- 3) Tilstanden "1" på både S og R brukes normalt ikke



SR		Q	
S	R	Q	
0	0		låst
0	1	0	
1	0	1	
1	1	0	

# SR-latch – funksjonell beskrivelse



SR		Q
S	R	Q
0	0	
0	1	0
1	0	1
1	1	0

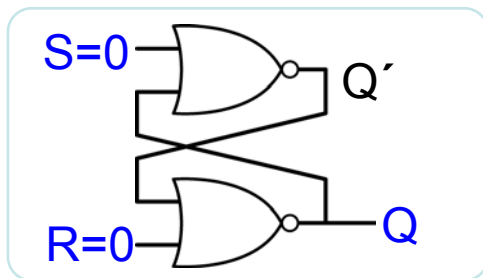
låst

# SR-latch – Portimplementasjon NOR

\*Signalet  $Q'$  er ikke invertert av  $Q$  for tilstand  $S=1, R=1$

# SR-latch – Analyse

- Tilstand  $S=0, R=0$ :



||

Øvre NOR

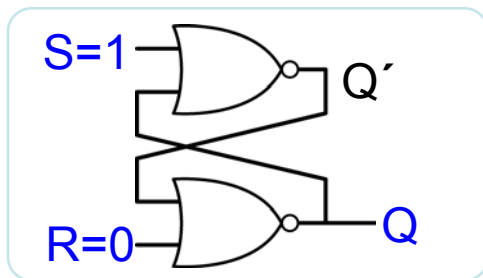
$S$	$Q$	$Q'$
0	0	1
0	1	0
1	0	0
1	1	0

Nedre NOR

$Q'$	$R$	$Q$
0	0	1
0	1	0
1	0	0
1	1	0

# SR-latch – Analyse

- Tilstand  $S=1, R=0$ :



||

Øvre NOR

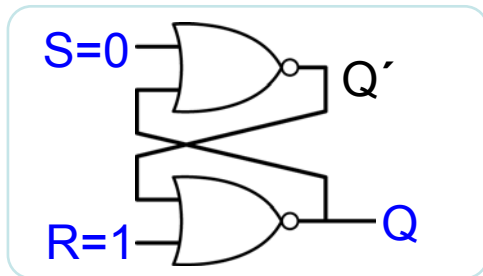
$S$	$Q$	$Q'$
0	0	1
0	1	0
1	0	0
1	1	0

Nedre NOR

$Q'$	$R$	$Q$
0	0	1
0	1	0
1	0	0
1	1	0

# SR-latch – Analyse

- Tilstand  $S=0, R=1$ :



||

Øvre NOR

S	Q	Q'
0	0	1
0	1	0
1	0	0
1	1	0

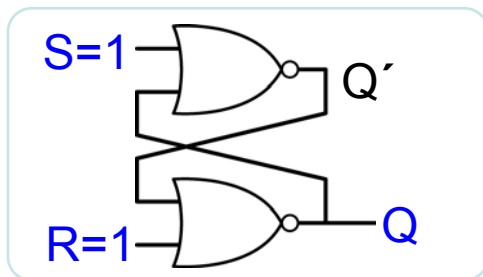
Nedre NOR

Q'	R	Q
0	0	1
0	1	0
1	0	0
1	1	0



## SR-latch – Analyse

- Tilstand  $S=1, R=1$



Øvre NOR

$S$	$Q$	$Q'$
0	0	1
0	1	0
1	0	0
1	1	0

Nedre NOR

$Q'$	$R$	$Q$
0	0	1
0	1	0
1	0	0
1	1	0

I denne tilstanden er utgang  $Q'$  ikke den inverterte av  $Q$ . Denne tilstanden brukes normalt ikke

# SR-latch oppsummering

Tilstand  $S=0, R=0$  gir

Tilstand  $S=0, R=1$  gir

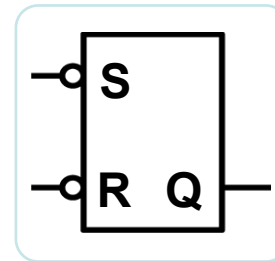
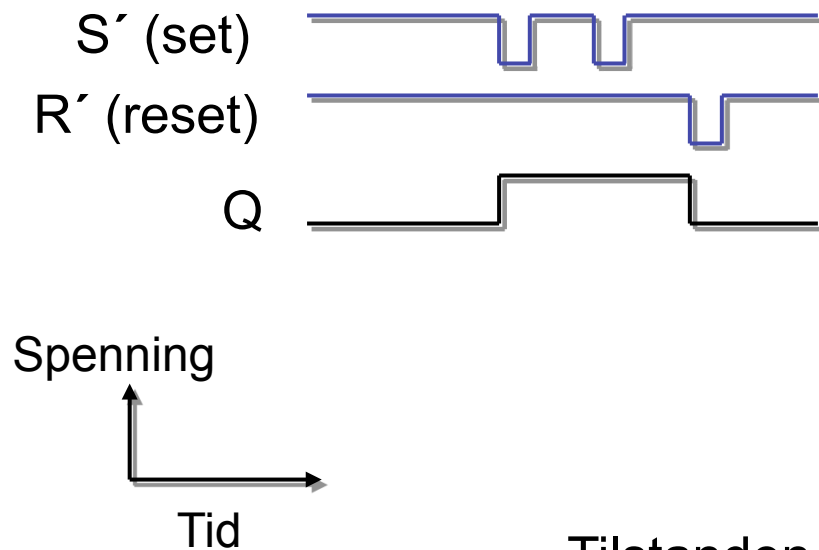
Tilstand  $S=1, R=0$  gir

Tilstand  $S=1, R=1$  gir

SR		Q
S	R	Q
0	0	
0	1	
1	0	
1	1	

# S'R'Latch

S'R' latch – lik funksjon som SR latch, men reagerer på "0" inn (negativ logikk)

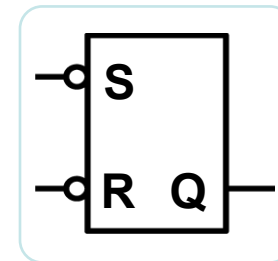
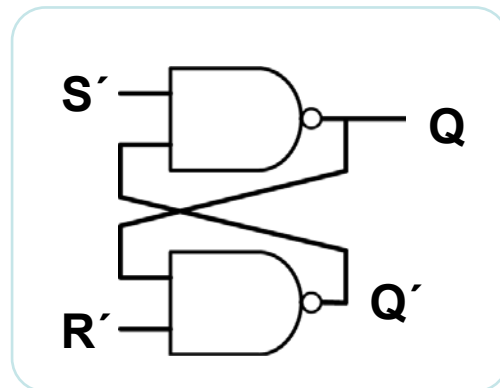


S'	R'	Q
1	1	låst
1	0	0
0	1	1
0	0	1

Tilstanden  $S' = 0, R' = 0$  brukes normalt ikke

## S'R'Latch – Portimplementasjon NAND

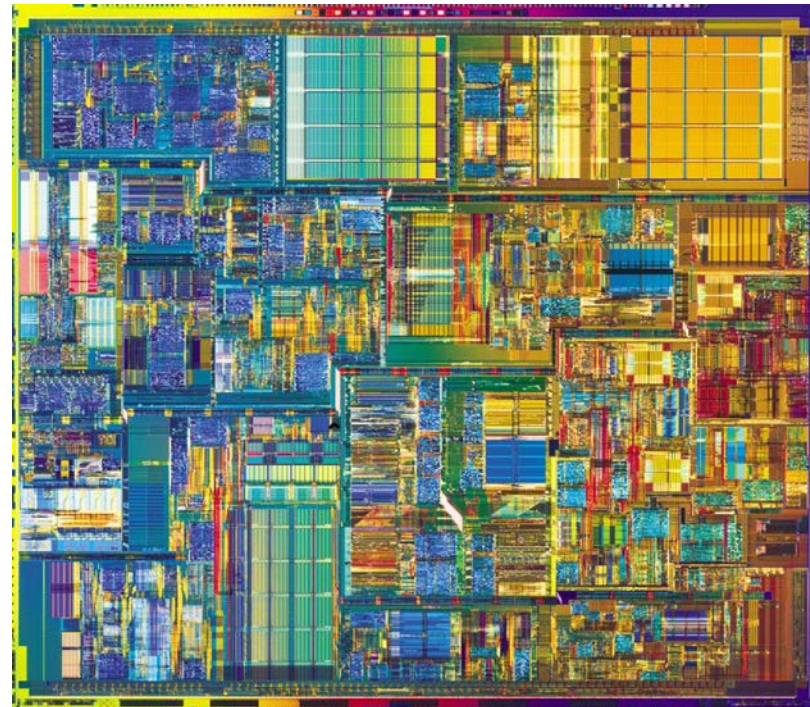
Kretsen kan analyseres på samme måte som for SR latch



$S'$	$R'$	$Q$
1	1	uforandret
1	0	0
0	1	1
0	0	1

# Synkron logikk

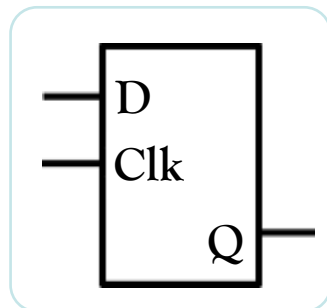
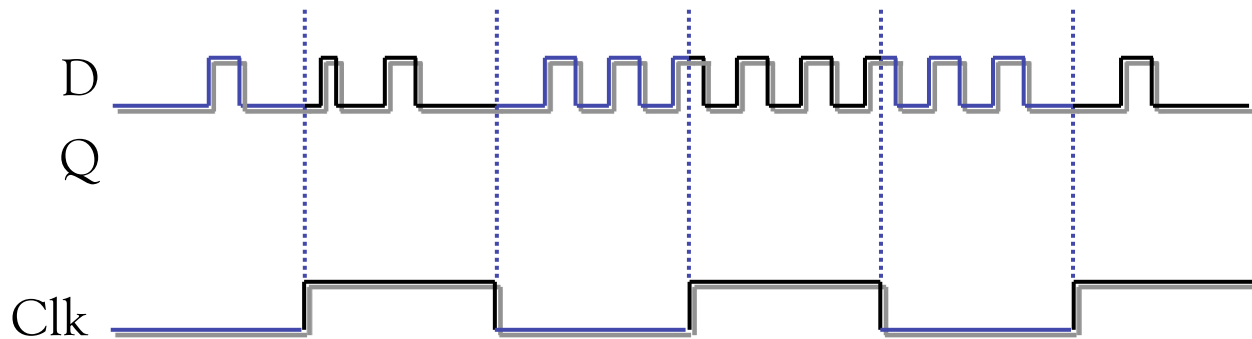
- I større digitale system har man behov for å synkronisere dataflyten. Til dette bruker vi et globalt klokkesignal
- Uten global synkronisering ville det vært total kaos



## D-Latch

- Dataflyten gjennom en D-latch kontrolleres av et klokkesignal
  - 1) Slipper gjennom et digital signal så lenge klokkeinngangen er "1" (transparent)
  - 2) I det øyeblikket klokkeinngangen går fra "1" til "0" låser utgangen seg på sin nåværende verdi. Forandringer på inngangen vil ikke påvirke utgangsverdien så lenge klokkesignalet er "0"

# D-Latch



Logisk verdi på D i det øyeblikk Clk går i fra "1" til "0" bestemmer verdien som holdes på Q

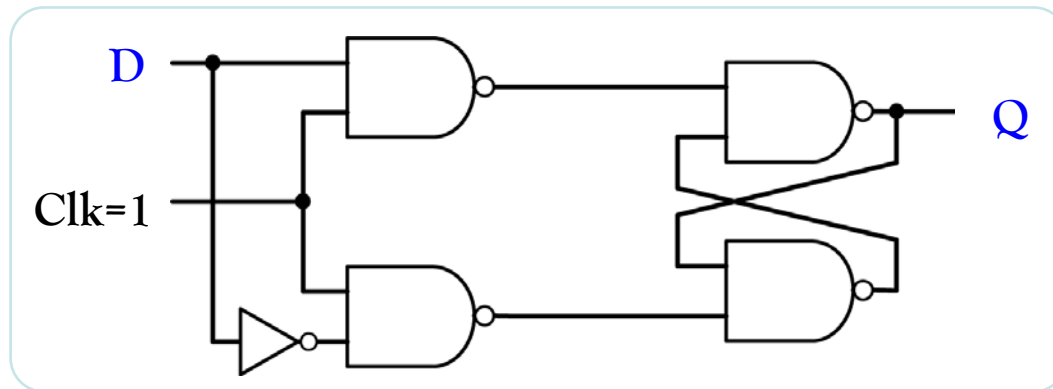
# D-latch – implementasjon





# D-latch – analyse

Tilstand 1: Clk = 1



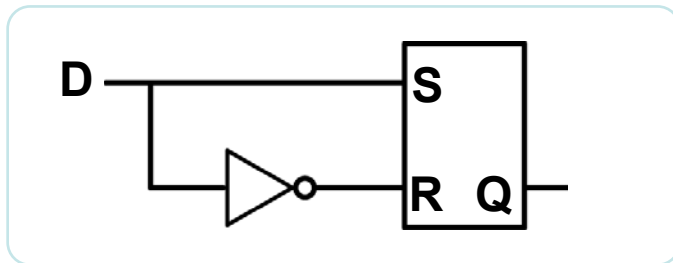
||

# D-latch – analyse tilstand 1 fortsetter

Dette er en  $S'R'$  latch med inverterte innganger

Tilstand  $S'=R'$  oppstår aldri, og kretsen er ekvivalent med en  $S''R''$  eller SR latch

## D-latch – analyse tilstand 1 fortsetter



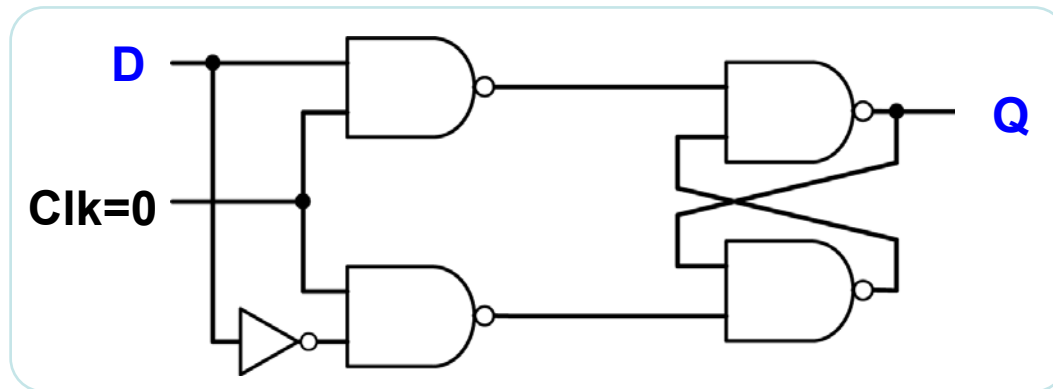
SR		
S	R	Q
0	0	uforandret
0	1	0
1	0	1
1	1	0

S og R vil alltid være forskjellige

Fra sannhetstabellen ser vi at  $Q=S=D$ , kretsen slipper signal D rett gjennom (transparent)

## D-latch – analyse tilstand 2

### Tilstand 2: Clk = 0



NAND med "1" på en inngang er ekvivalent med NOT. Får stabil låsekrets. Utgang holdes.

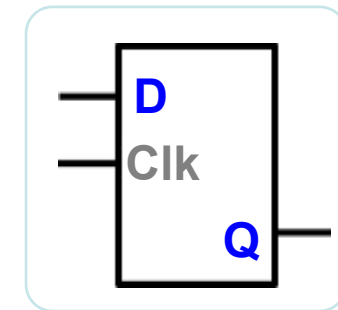
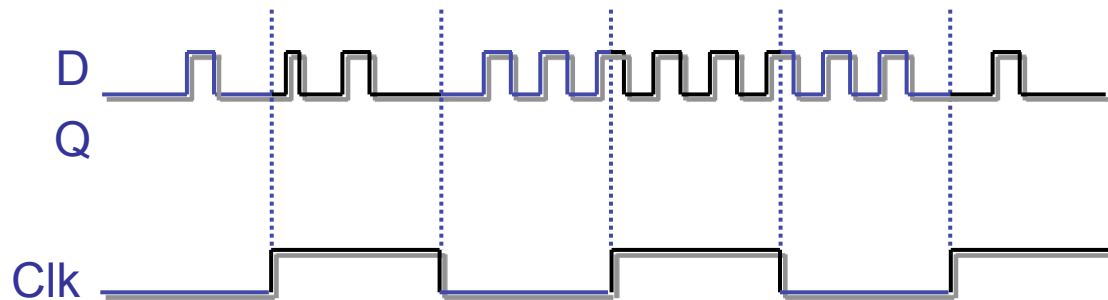
# Flip-flop

Flip-Flop'er kommer i to varianter:

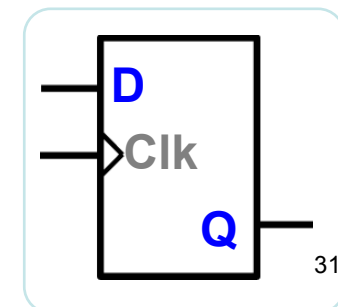
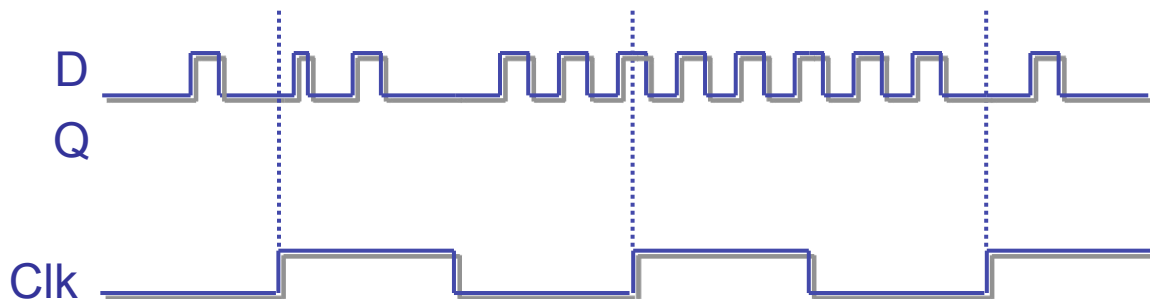
- Positiv flanketrigget
- Negativ flanketrigget

# D-Flip-Flop

En D latch er transparent for Clk=1

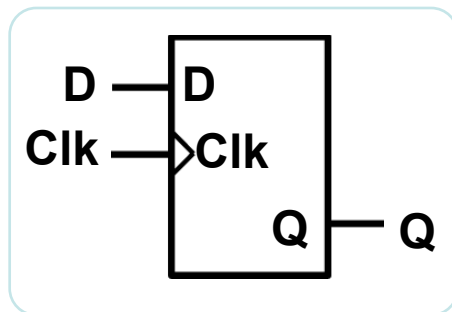


En positiv flanke-triggeret D flip-flop sampler verdien på D i det øyeblikk Clk går fra "0" til "1" (positiv flanke). Denne verdien holdes fast på utgangen helt til neste positive flanke



## Karakteristisk tabell/ligning

For flip-flop'er kan man generelt beskrive neste utgangsverdi  $Q(t+1)$  som funksjon av nåværende inngangsverdi(er), og nåværende utgangsverdi  $Q(t)$

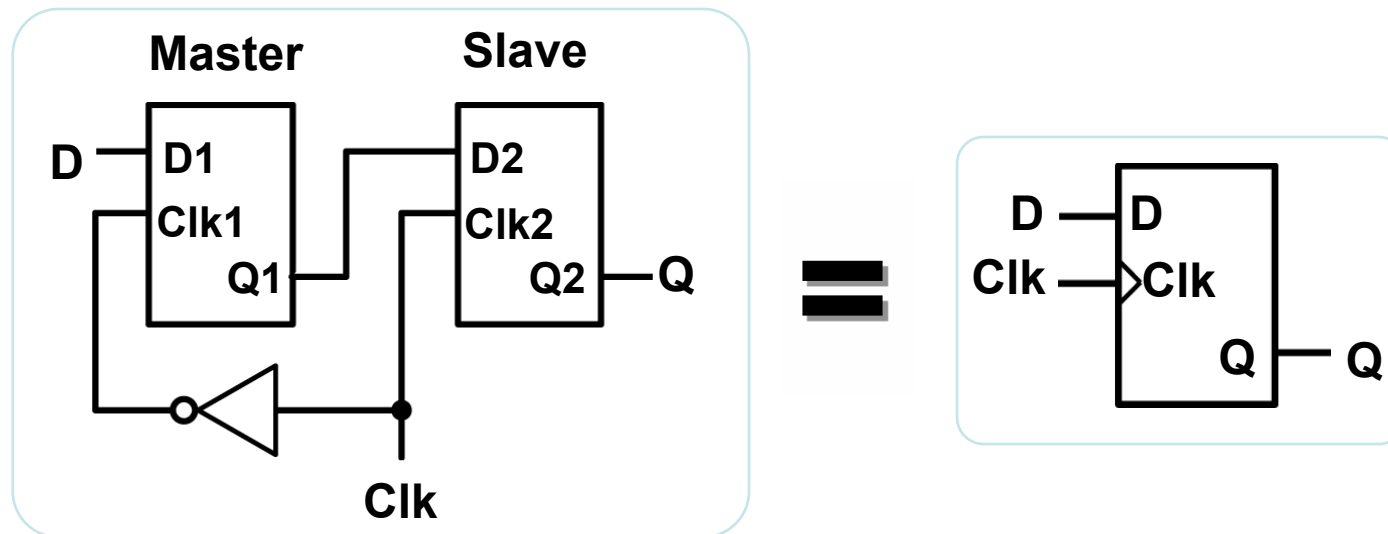


Karakteristisk tabell for D flip-flop

Karakteristisk ligning for D flip-flop

# D-Flip-Flop

En positiv flanketrigget D flip-flop kan lages av to D-latcher (Master-Slave)

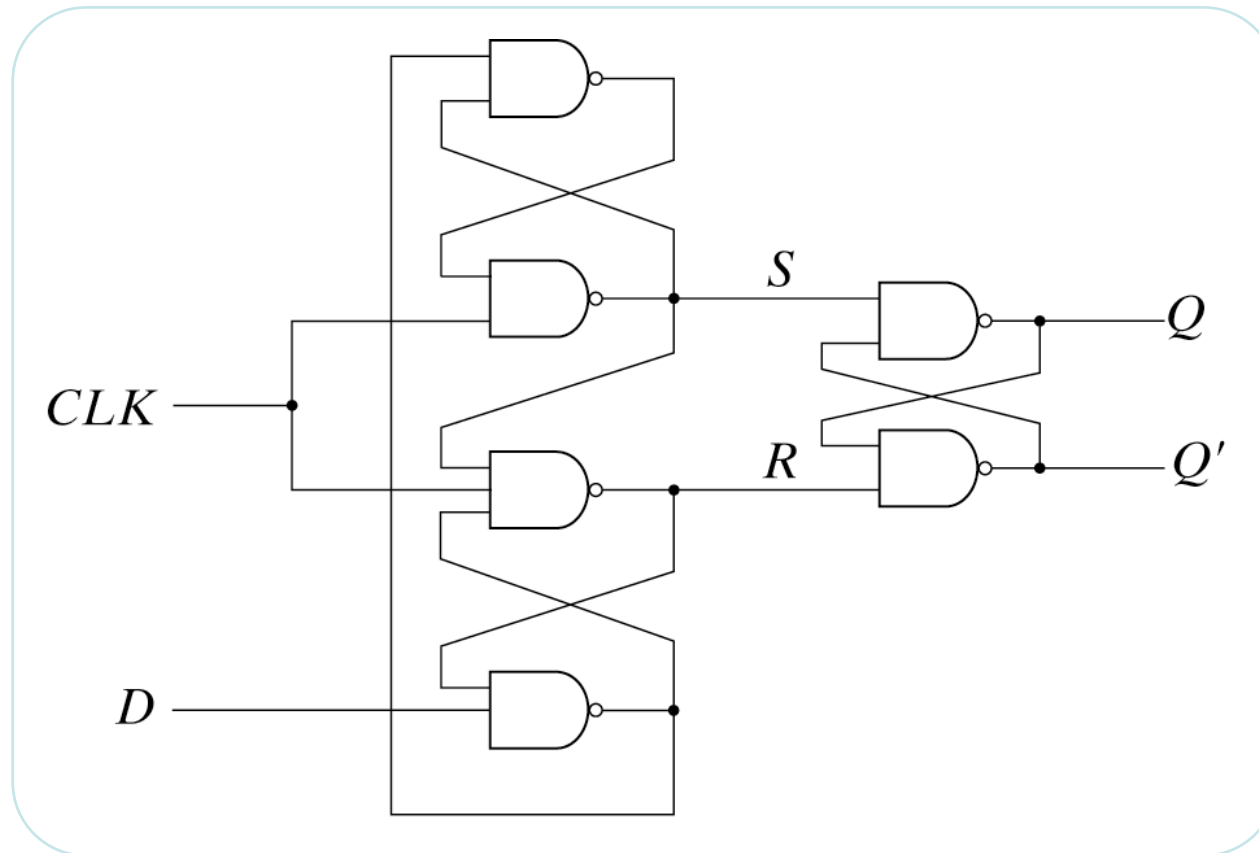


Under Clk=0 er første D latch (master) transparent

Under Clk=1 er siste D latch (slave) transparent



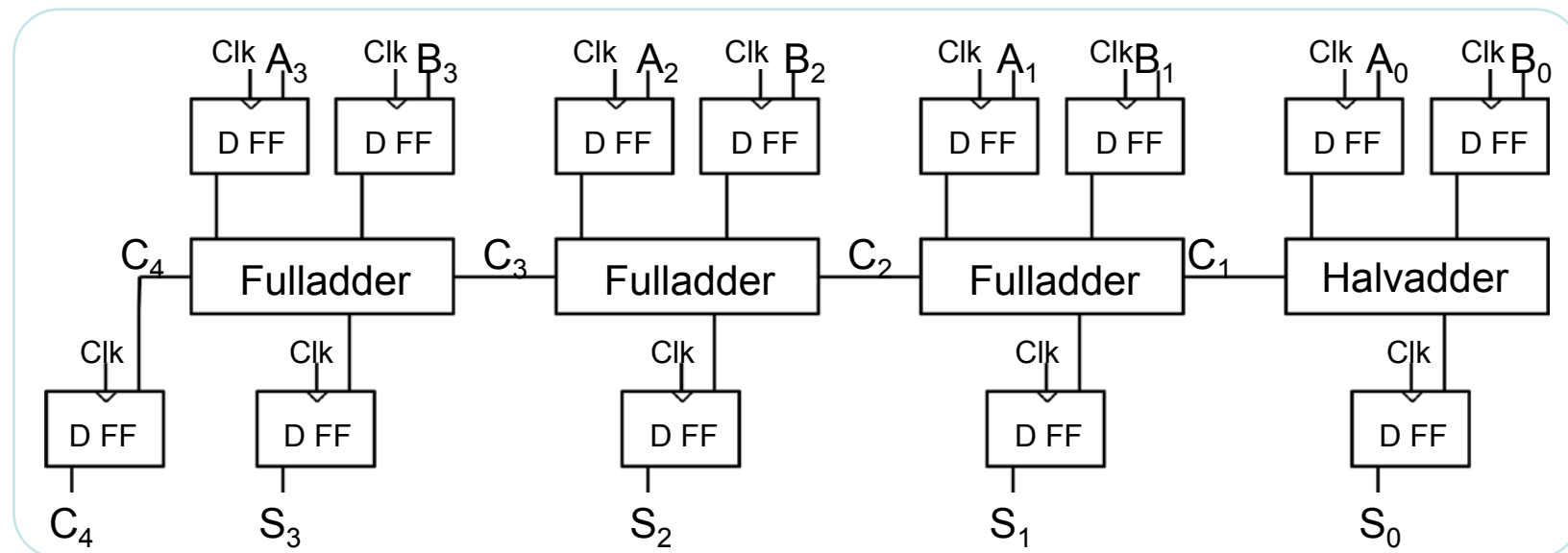
## D-Flip-Flop – kompakt versjon



## D-Flip-Flop, eksempel

En rippeladder vil i et kort tidsrom gi gal sum ut.

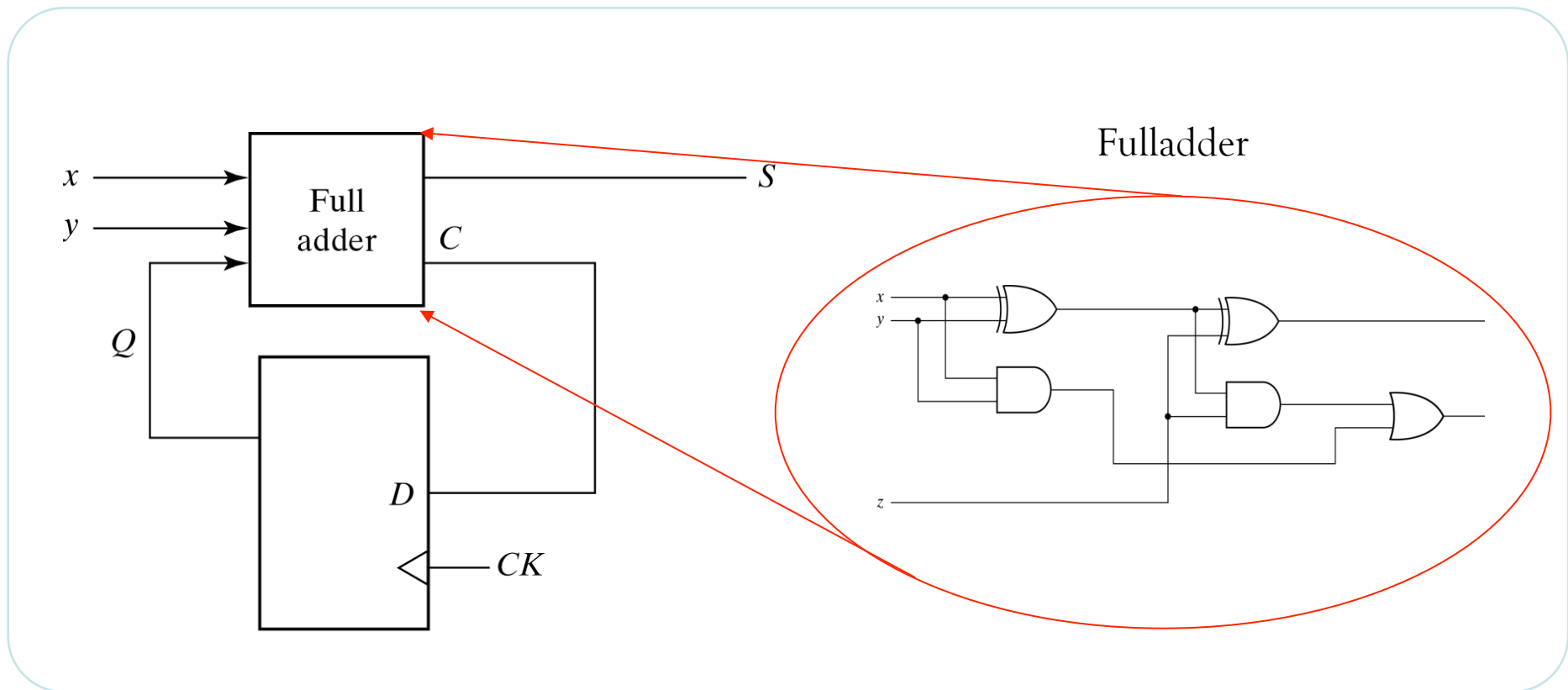
Styring av signalflyt med D flip-flops kamuflerer dette



På positiv Clk flanke kommer nye data inn til adderen. I samme øyeblikk leses forrige (stabiliserte) sum ut.

# D-Flip-Flop, eksempel

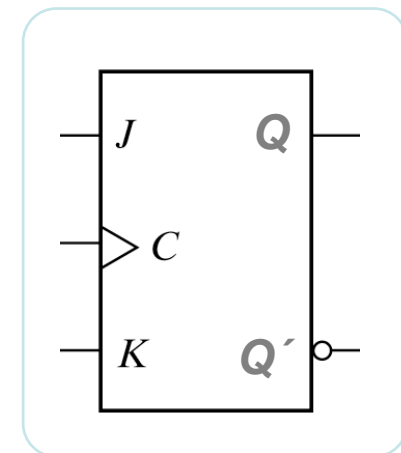
Seriell adder



# JK Flip-Flop

## Kretsoppbygging

Grafisk symbol



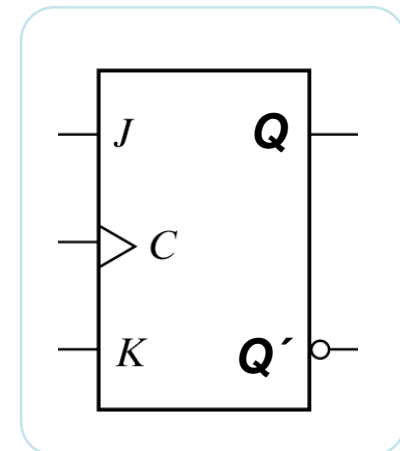
# JK Flip-Flop

En JK flip-flop har følgende egenskaper

- J=0, K=0:           Utgang låst
- J=0, K=1:           Resetter utgang til "0"
- J=1, K=0:           Setter utgang til "1"
- J=1, K=1:           Inverterer utgang  $Q \rightarrow Q'$

Utgangen kan kun forandre verdi på stigende klokkeflanke

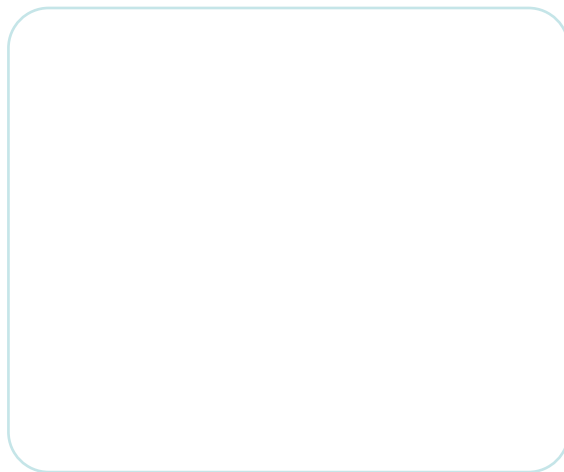
En JK flip-flop er den mest generelle flip-floppen vi har



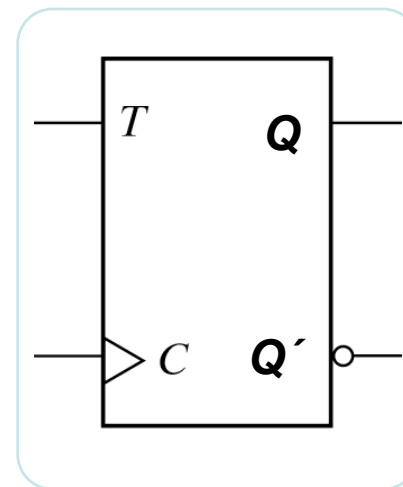
J	K	Q(t+1)
0	0	
0	1	
1	0	
1	1	

# T Flip-Flop

Kretsoppbygging



Grafisk symbol



# T Flip-Flop

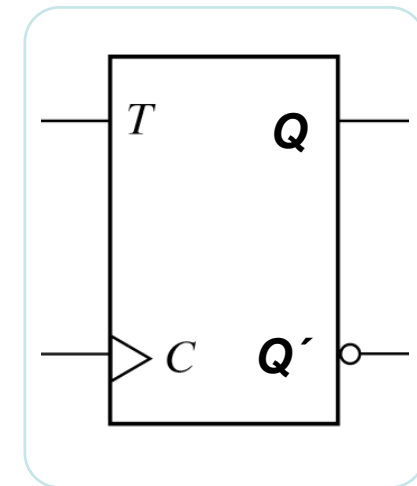
En T flip-flop har følgende egenskaper

$T=0$ ,    Utgang låst

$T=1$ ,    Inverterer utgang  $Q \rightarrow Q'$

Utgangen kan kun forandre verdi på stigende klokkeflanke

Det er lett å lage tellere av T flip-flop'er



T	Q(t+1)
0	
1	