

UiO : **Institutt for informatikk**

Det matematisk-naturvitenskapelige fakultet

INF1400

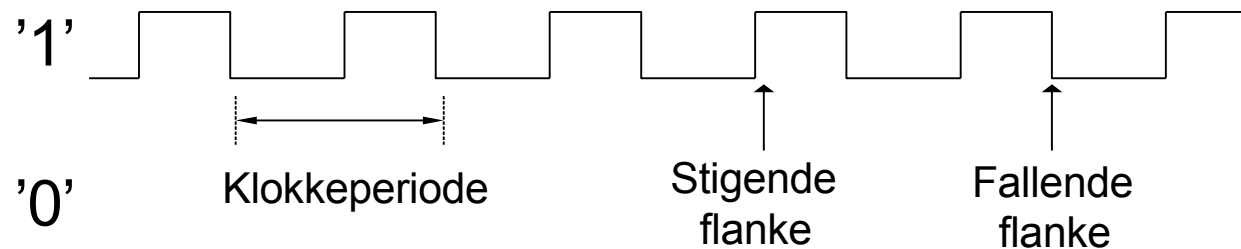
Sekvensiell logikk del 2



Oppsummering hittil

- 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

- I *synkrone* sekvensielle kretser skjer endringen(e) i output samtidig med endringen i et *klokkesignal*.
- I *asynkrone* sekvensielle kretser skjer endringen(e) i output uten noe *klokkesignal*.
- Nesten alle kretser er synkrone.
- Et klokkesignal er et digitalt signal som veksler mellom '0' og '1' med fast takt.

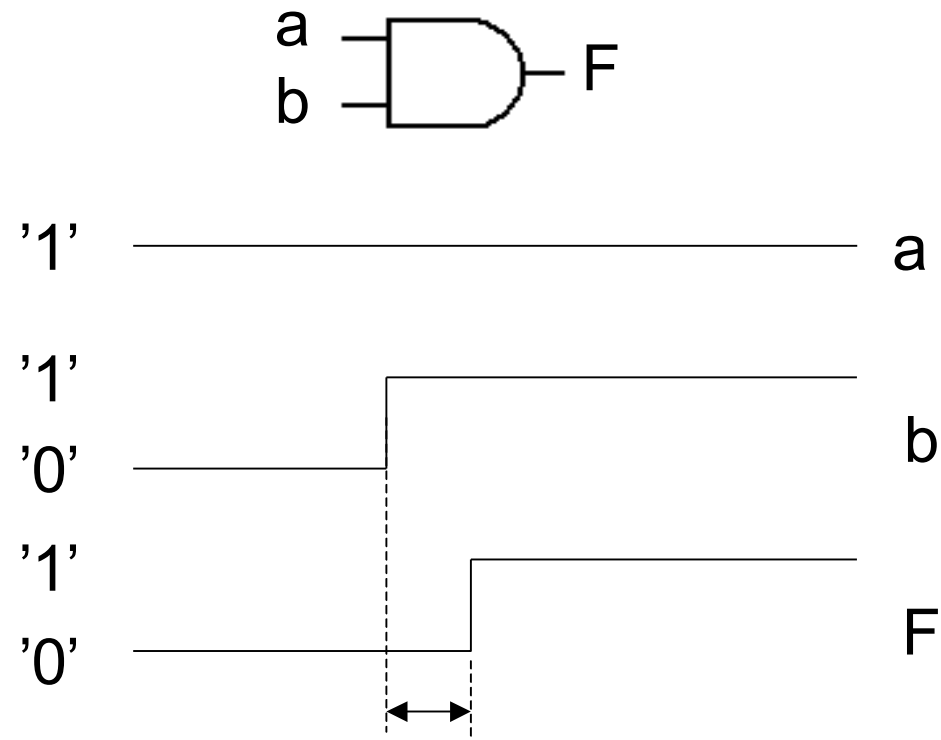


- Den omvendte av klokkeperioden kalles (klokke)frekvensen, altså

$$\textit{frekvens} = \frac{1}{\textit{klokkeperioden}}$$

- Ønsker så høy klokkefrekvens som mulig, fordi hver enkelt operasjon da bruker så kort tid som mulig.
- Maksimal klokkefrekvens bestemmes av flere faktorer, blant annet:
 - Lengde på signalveiene
 - Last
 - Forsinkelse gjennom porter (delay)
 - Teknologi.
- NB: Hastighet er ikke direkte proporsjonal med klokkefrekvens.

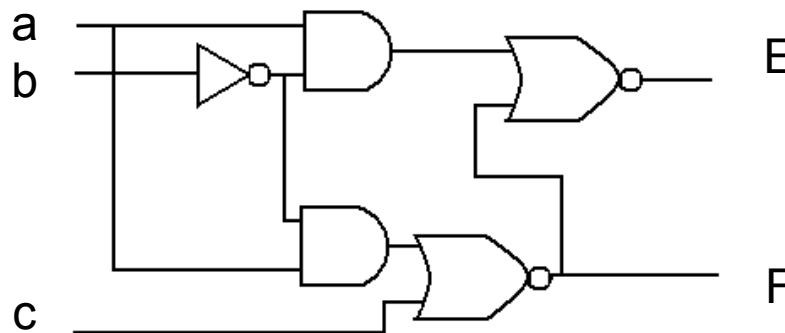
Portforsinkelse / tidsforsinkelse



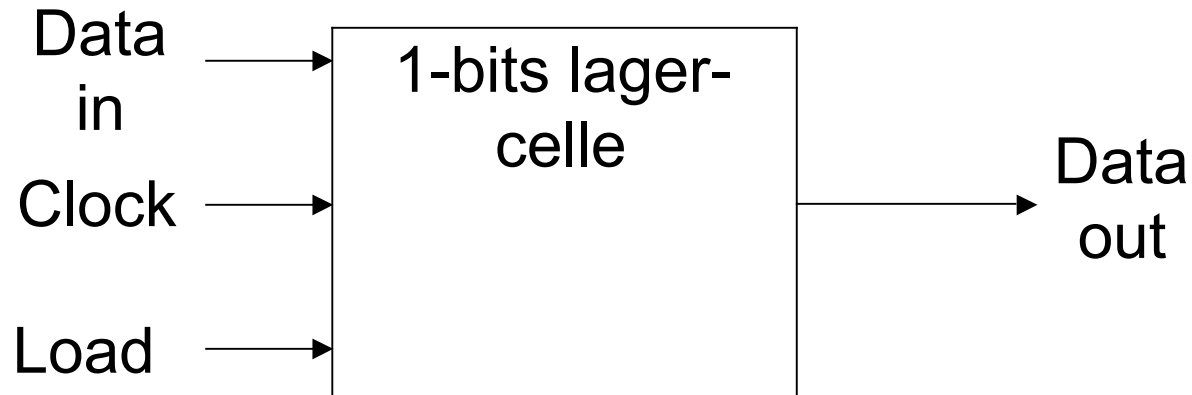
Logisk dybde

- Logisk dybde: Antall porter et signal passerer fra inngang til utgang.
- Ved å redusere logisk dybde reduseres forsinkelsen gjennom kretsen.

Eksempel:

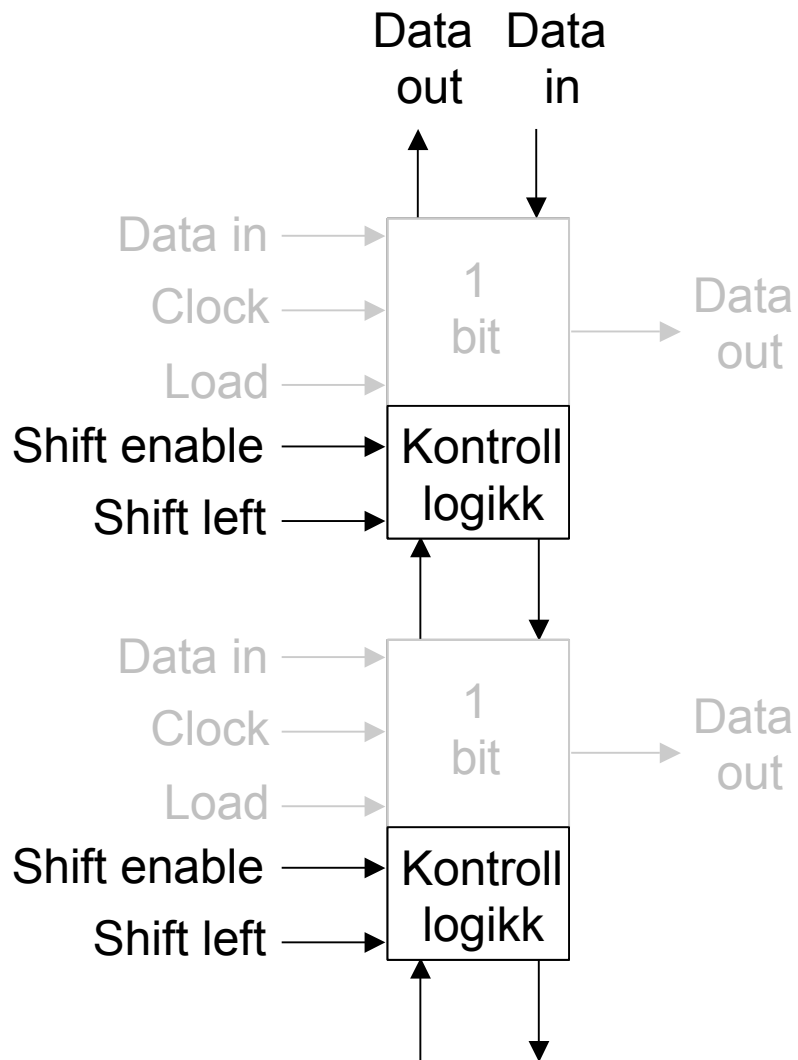


- Internt i en CPU trengs en fleksibel type lagercelle som kan lagre et bit.



- Som regel trenger man å lagre hele byte, halvord eller ord, og gjøre samme operasjon på alle bitene.
- Load-signalet bestemmer om ny verdi skal lastes inn eller ikke.
- Ved å sette sammen 1-bits celler i parallell, får man et register.
- Hvis man i tillegg kan laste data over i nabocellen, kalles det et skiftregister.

Shiftregister - enkel

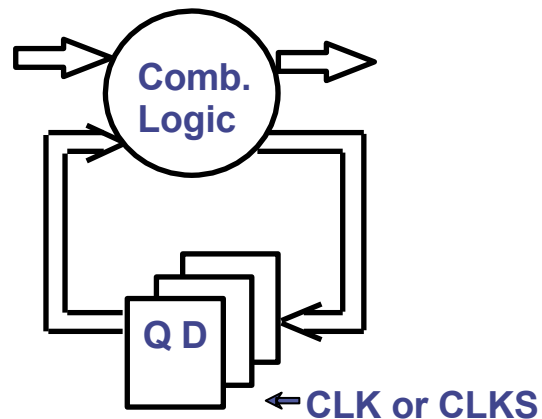
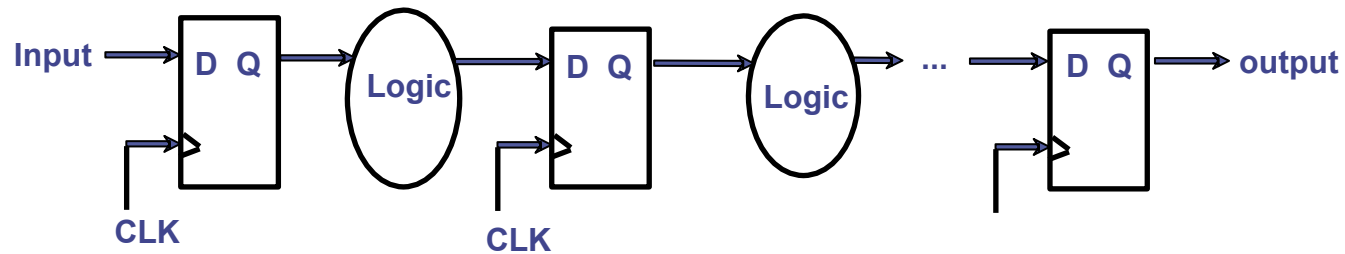


Load	Shift Enable	Shift Left	
'0'	'X'	'X'	Ingen endring
'1'	'0'	'X'	Data out := Data in
'1'	'1'	'0'	Shift right
'1'	'1'	'1'	Shift left

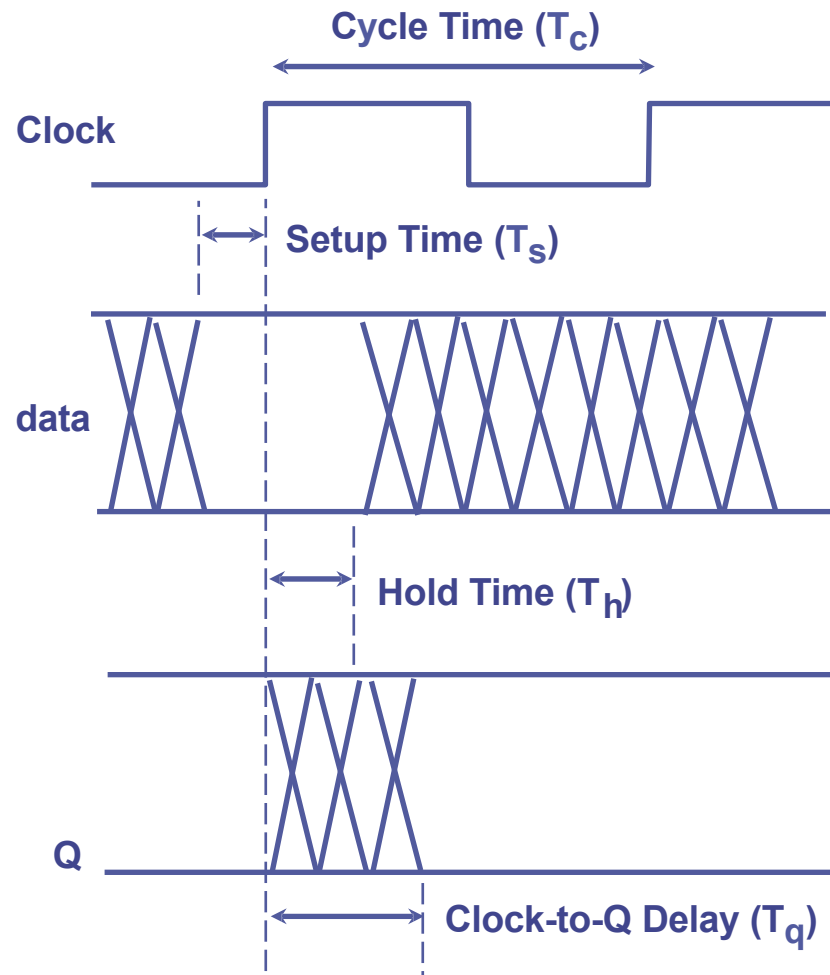
System (klokket)

De fleste VLSI systemer er en kombinasjon av:

- (a) Pipeline
- (b) tilstandsmaskiner (FSM)



1-fase klokking



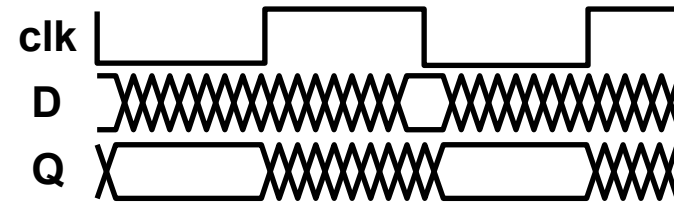
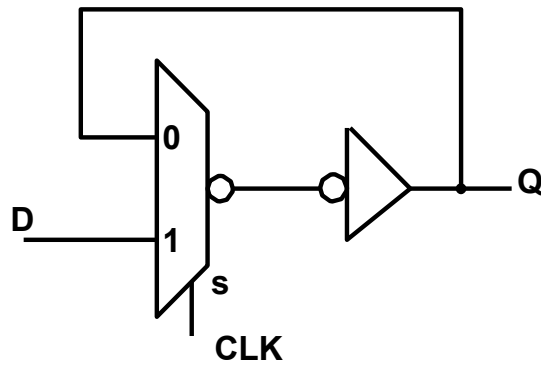
T_c = klokkeperioden

T_s = tiden før klokkeflanken hvor inngangen må være stabil og tilgjengelig

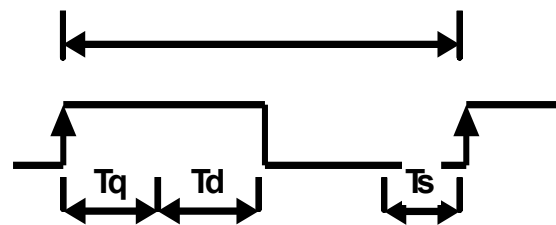
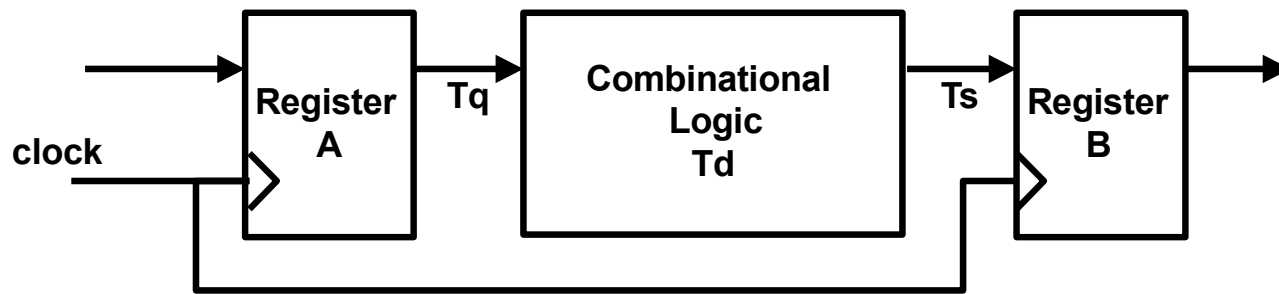
T_h = tiden etter klokkeflanken hvor inngangssignalet må fortsatt være stabil

T_q = tiden det tar fra klokkeflanken til utgangen er klar

1-fase klokking eksempel



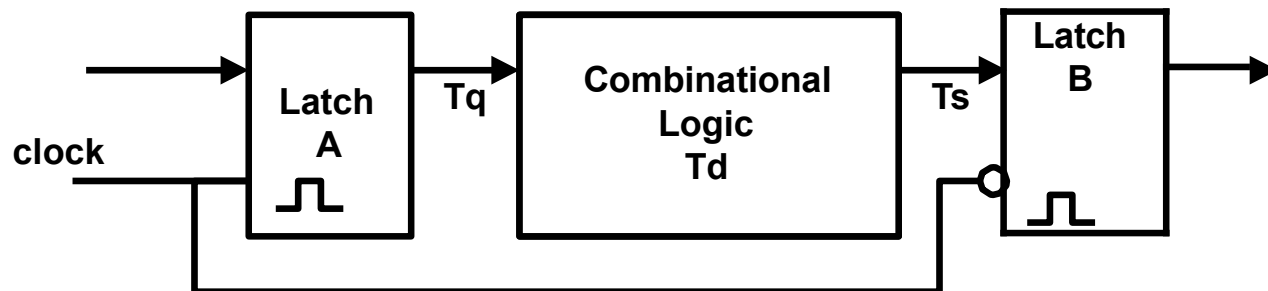
System timing



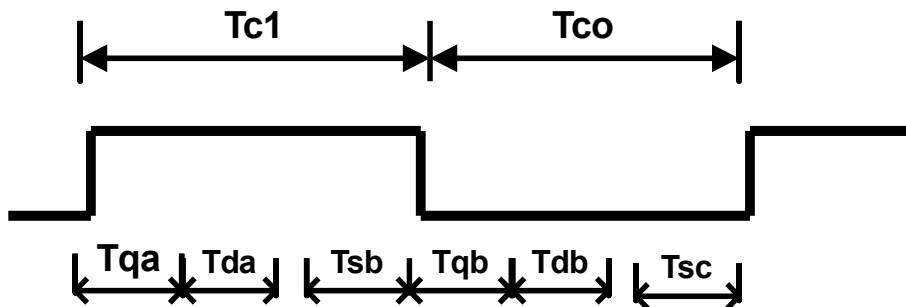
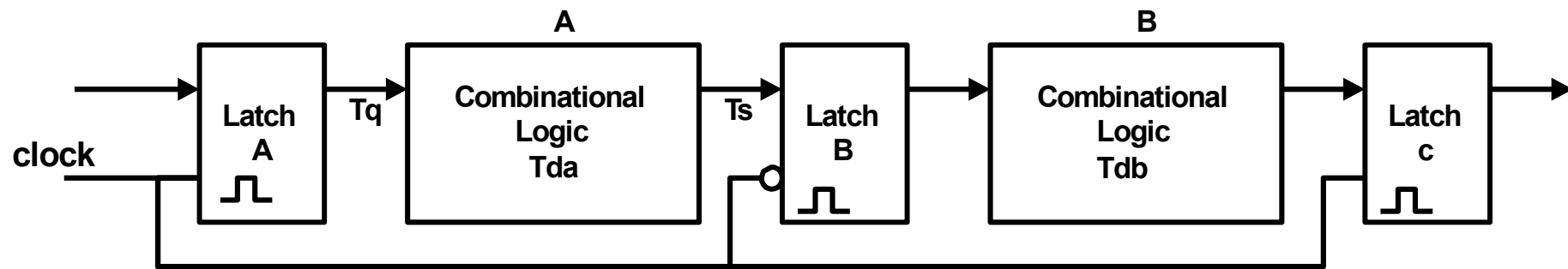
$$T_c > T_q + T_d + T_s$$

System timing (alternativ)

Alternativt, kan man bruke latcher som lagrings element for å spare plass



System timing



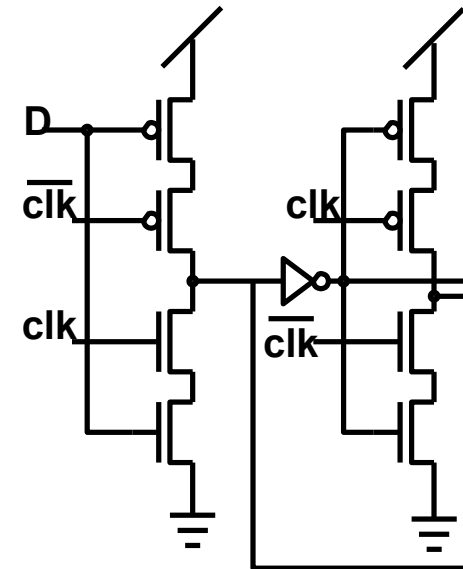
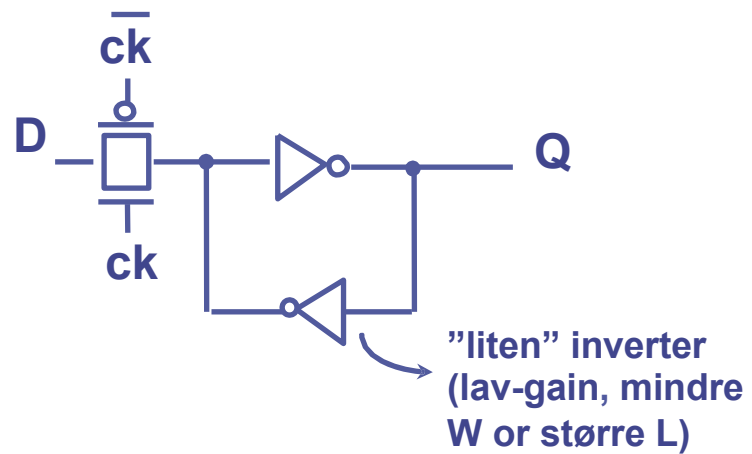
$$T_{c1} > T_{qa} + T_{da} + T_{sb} \quad T_{co} > T_{qb} + T_{db} + T_{sc}$$

If $T_c = T_{c1} + T_{co}$ and $T_{c1} = T_{co}$, $T_{qa} = T_{qb}$,

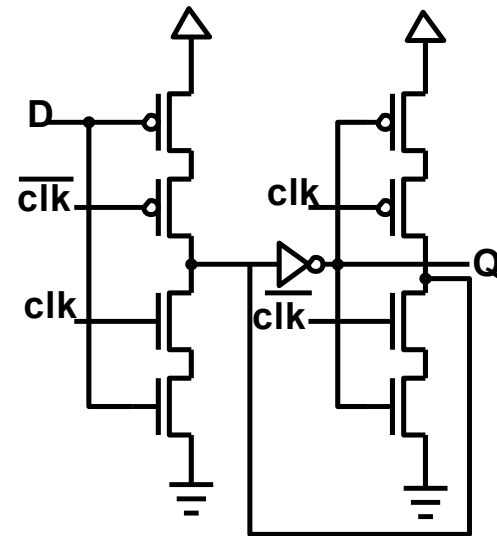
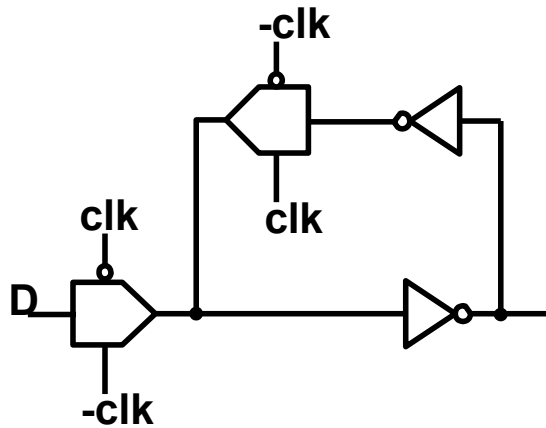
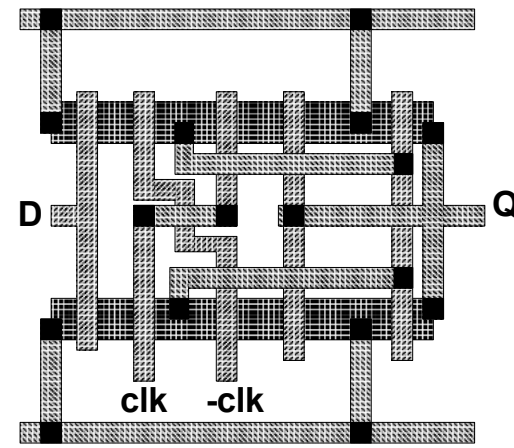
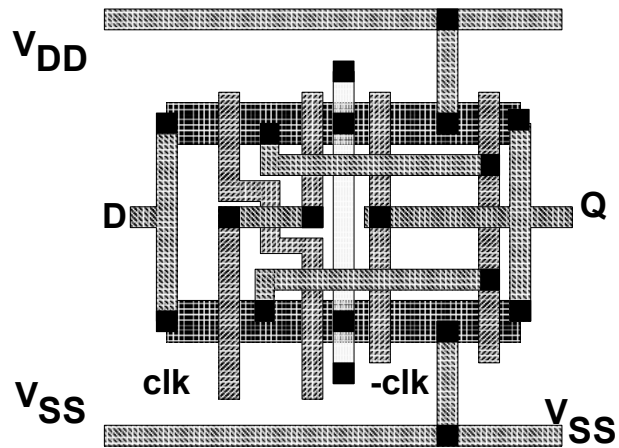
$T_{sb} = T_{sc}$

\Rightarrow The limit is $T_c = T_{da} + T_{db} + 2(T_q + T_s)$

Klokke latch i CMOS



Typisk utlegg for latch



Tilstandsmaskin - Teller

- Synkron 3-bits teller har en modus signal M
 - $M = 0$, teller oppover i binær sekvens
 - $M = 1$, teller oppover i Gray Code sekvens

Binært: 000, 001, 010, 011, 100, 101, 110, 111

Gray: 000, 001, 011, 010, 110, 111, 101, 100

Et utvalg inputkombinasjoner

Input M	Nåværende tilstand	Neste tilstand
0	000	001
0	001	010
1	010	110
1	110	111
1	111	101
0	101	110
0	110	111

Tilstandsdiagram

