INF3400/4400 Digital Mikroelektronikk Løsningsforslag DEL 11 Latcher og vipper Våren 2007

YNGVAR BERG

I. Oppgaver

A. Forklar hvordan en statisk latch virker

A.1 Løsningsforslag

Teori

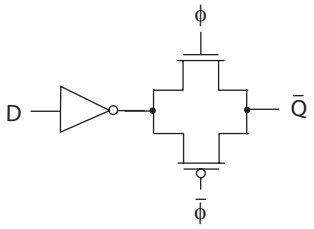

Fig. 1. *Dynamisk latch med inverter og transmisjonsport. (FIG7.17d)*

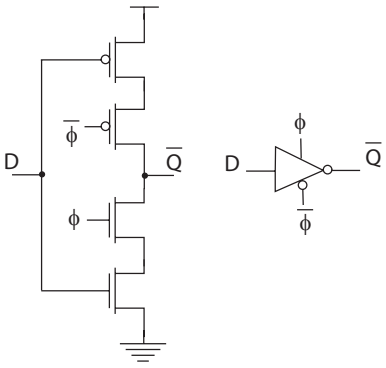

Fig. 2. *Klokket CMOS latch. (FIG7.18)*

I Fig. 1 er det vist en latch med inverter og transmisjonsport. En logisk ekvivalent krets som vist i Fig. 1, men med mindre arealbehov er en *klokket CMOS inverter* (C^2MOS) som er vist i Fig. 2. C^2MOS er noe tregere enn enn en inverter og en transmisjonsport fordi transistorene som styres av klokkesignalene aldri vil bidra i parallell. Det er derfor ikke vanlig å bruke klokket CMOS på inngangen til latchen.

Ved å kombinere latchene i Fig. 3 og 1 og klokke transmisjonsportene i motfase får vi en *statisk latch* som vist i Fig. 4. Det som nå mangler er gate terminal innngang.

I Fig. 5 har latchen fått en inverter på inngangen og utgangen blir dermed Q på grunn av to inverteringer. Utgangen lastes av C^2MOS inverteren i tillegg til eksterne kretser. En raskere latch der lasten på utgangen er redusert er vist i Fig. 6. Dette er en latch som ikke har noen av de begrensinger som ble beskrevet for pass transistor latchen. Vi ser imidlertid at latchen har blitt

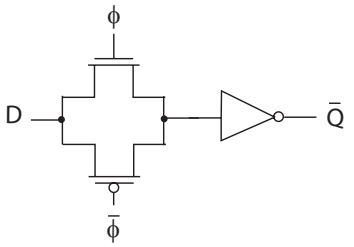

Fig. 3. *Dynamisk latch med transmisjonsport og inverter. (FIG7.17c)*

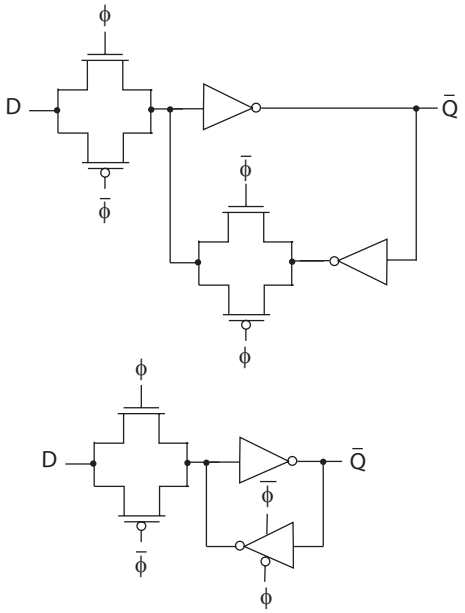

Fig. 4. *Statisk latch med transmisjonsport og inverter og tilbakekobling i motfase øverst. Tilbakekobling med C^2MOS nederst. (FIG7.17e)*

relativt kompleks, som medfører økt tidsforsinkelse, effektforbruk og økt areal (utlegg). Enklere latcher med gode elektriske egenskaper baseres på latchen i Fig. 6 med forenklinger som øker latchens ytelse og reduserer arealbehovet.

En enkel forenkling av latchen i Fig. 6 er vist i Fig. 7, der C^2MOS inverteren er erstattet med en svak[1] inverter. Når latchen sampler (latcher inn) vil inngangsignalet via inngangsinverteren og transmisjonsporten overstyre tilbakekoblingen slik at utgangen Q får ny verdi. Når transmisjonsporten er skrudd AV vil tilbakekoblingen være tilstrekkelig sterk til å holde verdien på Q. Denne latchen er derfor statisk.

[1]Med svak inverter menes en inverter som leverer lite strøm på grunn av lite W/L forhold for transistorene.

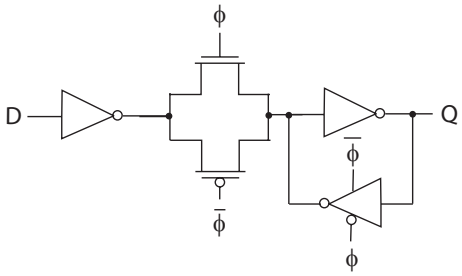

Fig. 5. *Statisk latch med inverter inmngang og utgang Q. (FIG7.17f)*

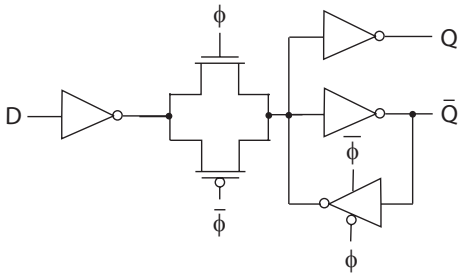

Fig. 6. *Statisk latch med utgang Q. (FIG7.17g)*

Timing for latcher

Timing for en latch er vist i Fig. 8. Samplingstidspunktet for latchen er ved fallende klokkeflanke. Vi må forutsette at inngangssignalet D kommer fra en latch styrt av et annet klokkesignal, for eksempel invertert klokke som gir klokkesignaler i motfase. Mellom latchene er det typisk kombinatorisk logikk.

Timingdetaljer for en latch er vist i Fig. 9. Når klokkesignalet CLK er lavt vil latchen fungere som et minneelement med tilbakekobling. Dersom vi ser på latchen og kombinatorisk logikk som prosesserer signalet fra latchen og neste latch får vi et system som vist i Fig. 10 med $CLK = \phi_1$ og $\overline{CLK} = \phi_2$. Fra stigende klokkeflanke får vi klokke til Q forsinkelse t_{ccq} og t_{pcq} som vist i Fig. 9. Latchen er transparent når klokkesignalet er høyt og enhver forandring på D vil påvirke Q. Vi må forutsette at D er stabil en liten stud før fallende klokkeflanke slik at latchen rekker å sample riktig verdi. Vi kaller denne tiden setup tid. For å være sikker på riktig sampling må D være stabil en stund etter at klokkesignalet har blitt 0. Vi kaller dette for hold tid.

Timingdetaljer i et sekvenseringssystem med transparente latcher som styres av tofase klokker er vist i Fig. 11. Vi antar at inngangen $D1$ ankommer latch 1 når $\phi_1 = 1$ og propagerer i kombinatorisk logikk $KL1$ fordi latch 1 er transparent når $\phi_1 = 1$. Kritisk signalvei i KL1 vil bestemme maksimal tids-

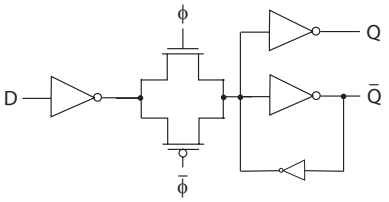

Fig. 7. *Statisk latch med utgang Q og svak uklokket tilbakekobling. (FIG7.17i)*

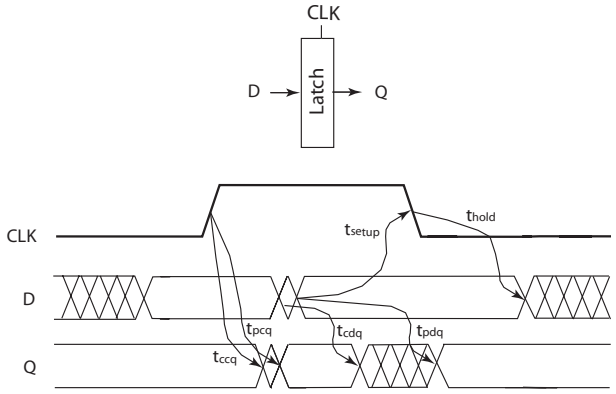

Fig. 8. *Timing for latch. (FIG7.4c)*

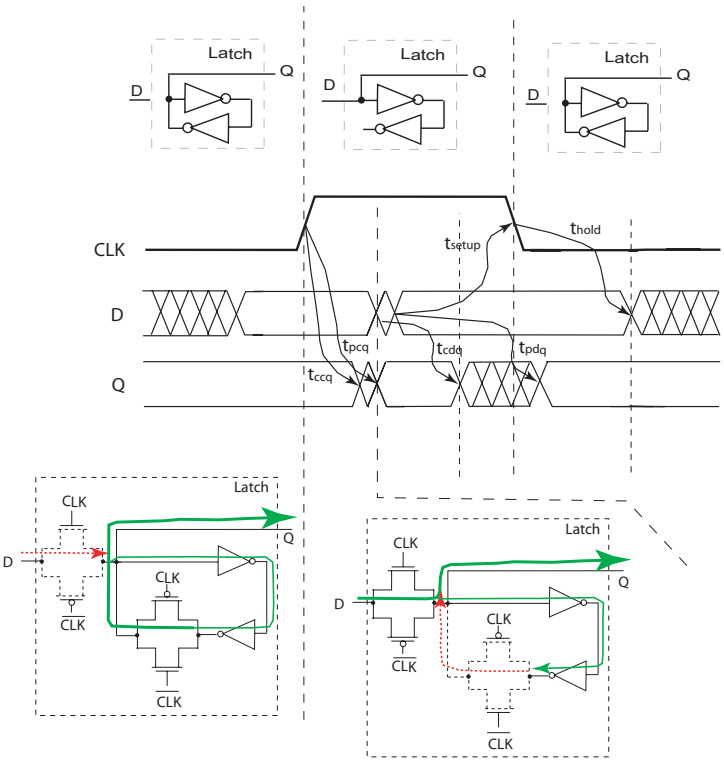

Fig. 9. *Timing for latch. (FIG7.4c)*

forsinkelse t_{pd1} for KL1. Vi må forutsette at $D2$ er stabil i god tid før ϕ_2 svinger fra 1 til 0 slik at vi får riktig verdi samplet (latchet) i latch 2. Tilsvarende argumentasjon gjelder for latch 3 osv. Vi kan uttrykke en klokkeperiode T_c som:

$$T_c \geq t_{pdq1} + t_{pd1} + t_{pdq2} + t_{pd2}. \quad (1)$$

Dersom vi løser med hensyn på total propageringsforsinkelse i hele klokkeperioden får vi:

$$\begin{aligned} t_{pd} &= t_{pd1} + t_{pd2} \\ &\leq T_c - (2t_{pdq}), \end{aligned} \quad (2)$$

der $(2t_{pdq2})$ er overhead gitt av propageringsforsinkelse i latchene, som vi antar er lik for de aktuelle latchene.

Eksempel på parameterverdier for to-fase latch er vist i tabell II-A.1. Vi kjenner klokkeperioden som er 500ps. Maksimal propageringsforsinkelse for to-fase transparent latch er gitt av

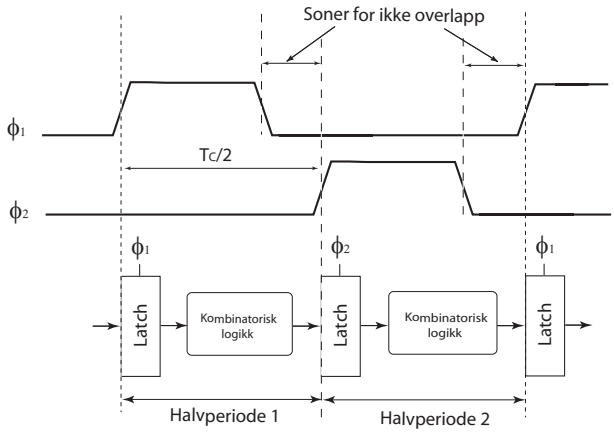

Fig. 10. *Sekvensering (synkronisering) med latcher. (FIG7.2)*

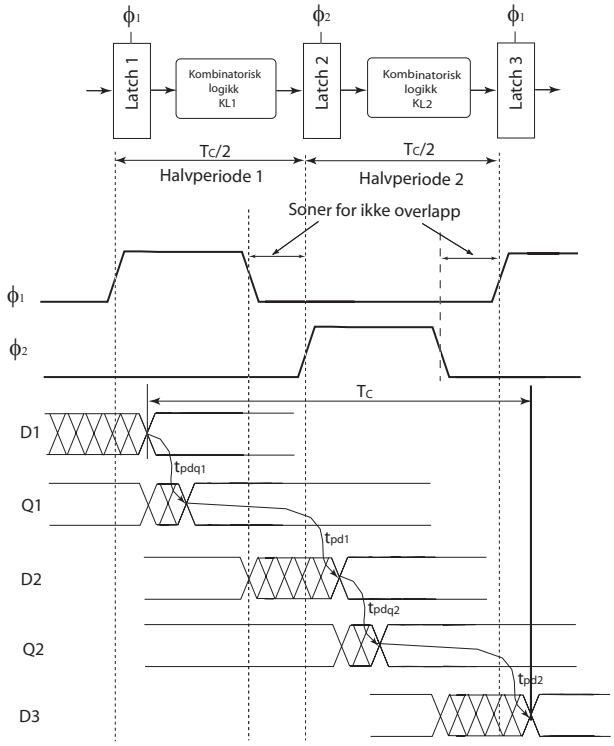

Fig. 11. *Begrensinger for maks forsinkelse i et sekvenseringssystem med latcher styrt av tofase klokker (FIG7.7).*

$$
\begin{aligned}
t_{pd} &= t_{pd1} + t_{pd2} \\
&\leq T_c - (2t_{pdq}) \\
&\leq 500ps - 2 \cdot 40ps \\
&\leq 420ps.
\end{aligned}
$$

Maksimal propageringsforsinkelse for to-fase transparent latch er 420ps.

I Fig. 12 er begrensninger på minimumsforsinkelse for latcher som er styrt av to fase lokker vist. Latchene styres av tofase ikkeoverlappende klokker som skal garantere at to latcher som styres av hver sin klokkefase ikke er åpne samtidig. Når begge klokkefasene ϕ_1 og ϕ_2 er lave samtidig skal begge latchene være lukket slik at utgangene ikke skal kunne påvirkes av inngangene. Ved stigende klokkeflanke på ϕ_1 åpner latchene som er

Term	Latch
t_{ccq}	35ps
t_{pcq}	50ps
t_{pdq}	40ps
t_{setup}	25ps
t_{hold}	30ps

TABLE I
Parameterverdier for to-fase latch.

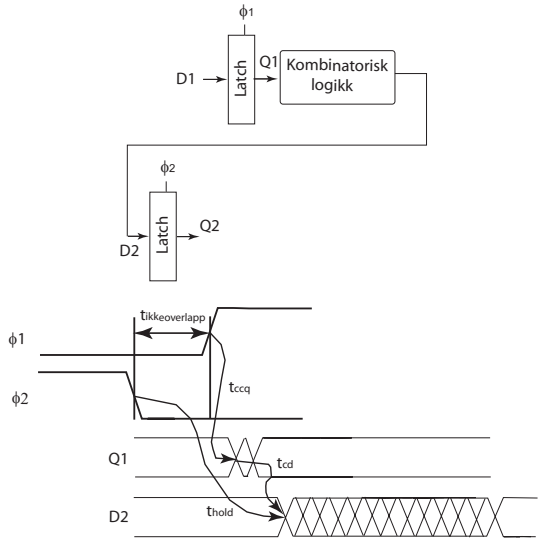

Fig. 12. *Begrensninger på minimumsforsinkelse for latcher som er styrt av to fase klokker. (FIG7.10).*

styrt av ϕ_1 slik at $D1$ latches inn til $Q1$. Merk at latchene er *nivåfølsomme*, dvs. utgangen på latchene vil påvirkes av inngangen så lenge latchen er åpen, i motsetning til en vippe som er *kantfølsom*. Vi forutsetter derfor at latchen som styres av ϕ_2 har en hold tid som går utover tiden når $\phi_2 = 1$. Vi kan anta at denne hold tiden t_{hold} er så lang at den kan påvirke utgangen $Q2$ etter at $Q1$ og $D2$ er endret som følge av latcing ved tidspunktet når ϕ_1 svinger fra 0 til 1.

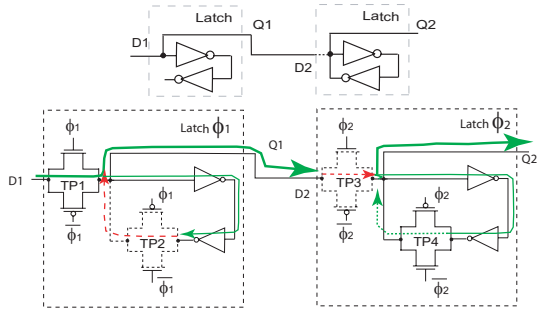

Fig. 13. *Begrensninger på minimumsforsinkelse for latcher som er styrt av to fase lokker. Detaljer ved stigende transisjon på ϕ_1. (FIG7.10).*

Timing detaljer for latcher som er styrt av to fase klokker ved stigende transisjon på ϕ_1 er vist i Fig. 13. Dersom hold tiden for latch styrt av ϕ_2 er for lang i forhold til tidsforsinkelse mellom latchene kan vi latche inn feil verdi. Vi har en situasjon der TP1 er PÅ slik at latch styrt av ϕ_1 er åpen og TP3 ikke er helt AV slik at latch styrt av ϕ_2 er delvis åpen. I denne situasjonen er den ene

latchen åpen og den andre delvis åpen slik at de to latchene satt sammen blir delvis transparent. Vi kan uttrykke betingelser for korrekt latching ved å sette en nedre grense for contamination forsinkelse for kombinatorisk logikk mellom latchene:

$$t_{cd1}, t_{cd2} \quad \geq \quad t_{hold} - t_{ccq} - t_{ikkeoverlapp}, \tag{3}$$

der t_{hold} er hold tid for latchene, t_{ccq} er klokke til Q contamination forsinkelse for latchene og $t_{ikkeoverlapp}$ er tiden der begge klokkefasene er lave. Alternativt kan vi utrykke dette som en begrensning på hold tiden:

$$t_{hold} \quad \leq \quad t_{ikkeoverlapp} + t_{ccq} + t_{cd}. \tag{4}$$

Dersom tiden der begge klokkefasene er lave og $t_{ikkeoverlapp}$ er tilstrekkelig lang vil vi ikke få problem med for liten tidsforsinkelse i kombinatorisk logikk mellom to latcher.

For to-fase transparente latcher har vi

$$\begin{aligned} t_{cd1}, t_{cd2} \quad &\geq \quad t_{hold} - t_{ccq} - t_{ikkeoverlapp} \\ &\geq \quad 30ps - 35ps - 0 \\ &\geq \quad 0. \end{aligned}$$

Minimum contamination forsinkelse for latchen er 0.

B. Forklar hvordan en konvensjonell statisk vippe med C^2MOS virker

B.1 Løsningsforslag

Teori

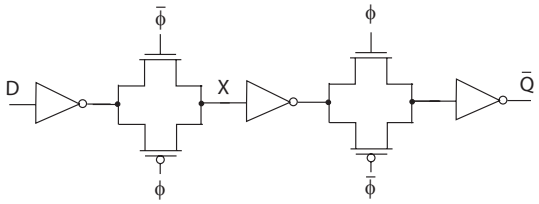

Fig. 14. *Dynamisk vippe. (FIG7.19a)*

En dynamisk vippe er vist i figur 14. Denne vippen er satt sammen av to dynamiske latcher som klokkes i motfase.

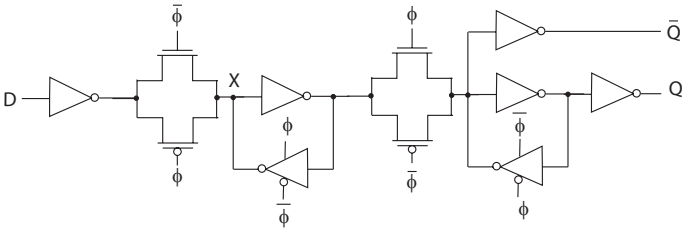

Fig. 15. *Statisk vippe. (FIG7.19b)*

En statisk vippe med to statiske latcher som er klokket i motfase er vist i Fig. 15. Denne vippen har både Q og $\overline{Q}$ utganger. Det er vanlig at vipper bare har en klokkeinngang ϕ og genererer invertert klokkesignal lokalt.

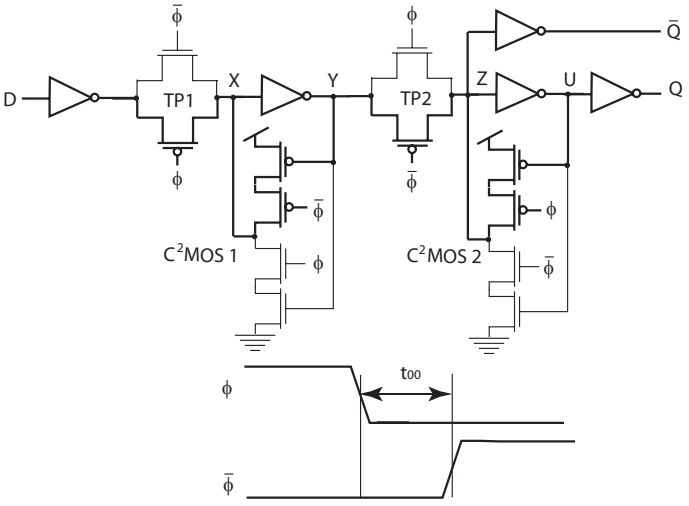

Fig. 16. *Statisk vippe ved negativ klokkeflanke og lokal generering av invertert klokke. (FIG7.19b)*

Ved lokal generering av invertert klokkesignal kan man få en liten forsinkelse for det inverterte klokkesignalet som vist i Fig. 16. Ved negativ klokkeflanke, dvs. ϕ skifter fra 1 til 0 vil det ta en viss tid t_{00} der begge klokkesignalene er lave. Signalveier som er markert med tykke linjer er da PÅ. Vi ser at den første transmisjonsporten $TP1$ er PÅ slik at inngangslatchen sampler inngangen. Tilbakekoblingen i inngangslatchen burde vært

skrudd av, men vil i perioden t_{00} ha ett opptrekk som er PÅ. Dette opptrekket er egentlig bare PÅ når $Y = 0$ som betyr at $X = 1$. I en situasjon der inngangen $D = 1$ får vi en konflikt i noden X fordi inngangen via en inverter og TP1 vil drive X til 0 mens tilbakekoblingen vil drive X til 1. Dette er bare et temporært problem fordi vi må forutsette at $\overline{\phi}$ endres til 1 før positiv klokkeflanke kommer. Etter t_{00} vil tilbakekoblingen skrus AV og nodene X og Y vil få riktig verdi drevet fra inngangen D. Problemet er mer betydelig enn man kan få inntrykk av ved bare å studere inngangslatchen i perioden t_{00}. Husk at utgangslatchen har samme klokkesignaler slik at i periden t_{00} vil transmisjonsporten for utgangslatchen $TP2$ også være feilaktig PÅ. Dette medfører at noden Z vil påvirkes av Y (direkte fra D) og via tilbakekoblingen i utgangslatchen C^2MOS 2. Vi ser at bare opptrekket i tilbakekoblingen er PÅ slik at tilbakekoblingen vil forsøke å precharge Z til 1. Den korrekte funksjonen til utgangslatchen er at $TP2$ er AV og tilbakekoblingen er PÅ. Vi ser at i perioden t_{00} er hele vippen transparent slik at D kan påvirke Q og $\overline{Q}$ direkte. En kritisk situasjon er når $Z = 0$ og $U = 1$ rett før t_{00} og $D = 1$ som betyr at Z drives mot 1 via Y og X fra D. I denne situasjonen er ikke tilbakekoblingen i utgangslatchen aktiv og Z kan derfor drives til 1, som igjen endrer U til 0 og bidrar til å holde $Z = 1$ feilaktig. Når perioden t_{00} er over vil $TP2$ stenge, men dette er for sent til å unngå en feilaktig endring av utgangene.

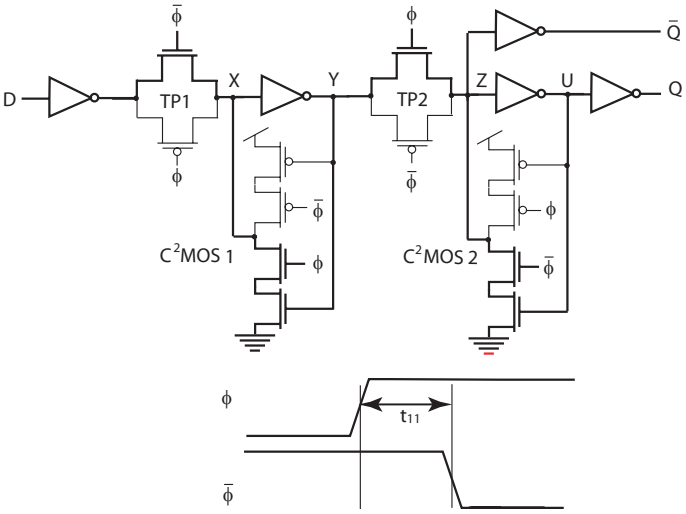

Fig. 17. *Statisk vippe ved positiv klokkeflanke og lokal generering av invertert klokke. (FIG7.19b)*

Vi får et tilsvarende problem ved positiv klokkeflanke som vist i Fig. 17. I perioden t_{11} vil begge klokkesignalene være høye slik at vippen blir temporært transparent. Riktig vippe funksjon er at inngangslatchen ikke sampler inngangen, men har en aktiv tilbakekobling. Dette betyr at $TP1$ skal være AV og C^2MOS 1 skal være PÅ. For utgangslatchen skal $TP2$ være PÅ og tilbakekoblingen C^2MOS 2 være av. I perioden t_{11} kan vi få en alvorlig situasjon dersom nodene X og Y endres på grunn av D og tilbakekoblingen C^2MOS 1 i inngangslatchen ikke kan overstyre $TP1$. I denne situasjonen blir vippen transparent.

En vanlig løsning på problemet med delvis transparente vipper er å bruke tofase ikke-overlappende klokker som vist i Fig. 18.

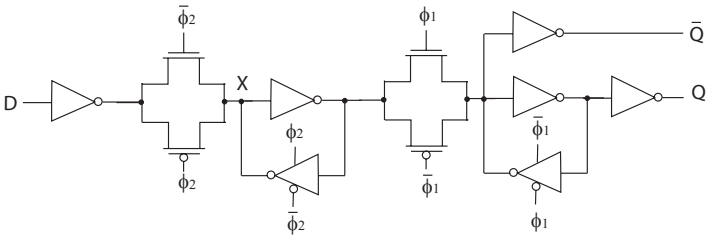

Fig. 18. *Statisk vippe med tofase ikke-overlappende klokker. (FIG7.21)*

Timing for vippe

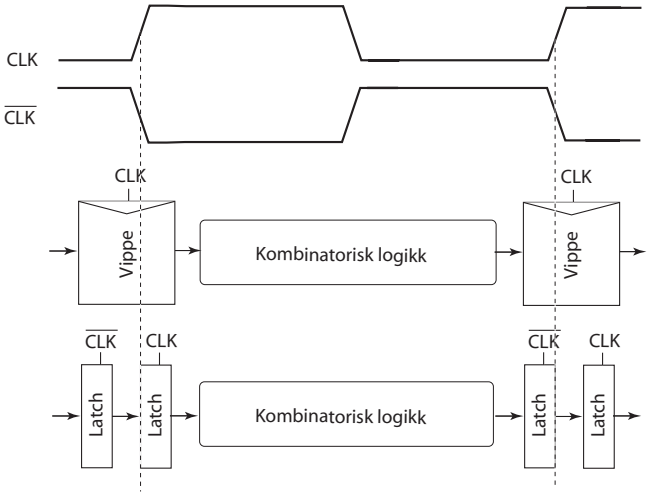

Fig. 19. *Sekvensering (synkronisering) med vipper realisert som latcher med klokke og invertert klokke. (FIG7.3)*

Sekvensering ved hjelp av vipper styrt av klokke (CLK) og invertert klokke ($\overline{CLK}$) er vist i figur 19. Her er to lacher plassert intill hverandre. Vi må da forutsette at de to latchene ikke er transparente samtidig, dvs. vi kan ikke tillate at CLK og $\overline{CLK}$ er høye samtidig. I praksis må vi sikre oss ved å ha en ikke-overlappende periode der CLK og $\overline{CLK}$ er lave samtidig. Et slikt system setter strenge krav til forholdet mellom CLK og $\overline{CLK}$.

Dersom vi har sekvensielt system bestående av vipper som klokkes med CLK og $\overline{CLK}$ må vi forutsette at disse signalene er presist i motfase.

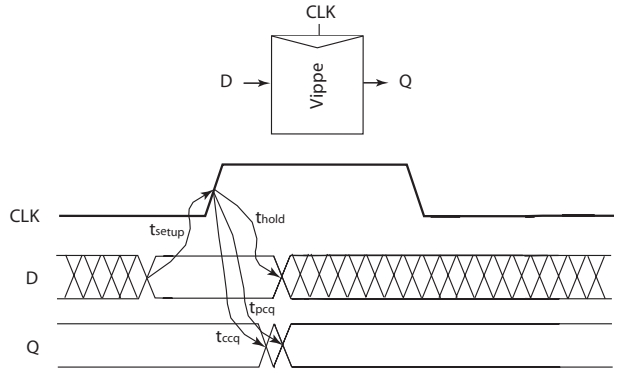

Fig. 20. *Timing for vippe. (FIG7.4b)*

Tidsforsinkelse i en vippe er vist i Fig. 20. Vippen vil holde en tilstand i hele perioden med unntak av lagring av ny verdi en kort stund etter at klokkesignalet skifter fra 0 til 1. Vippen er avhengig av *setup-* og *hold tid*. Dette betyr at inngangen D må være stabil en stund før klokkesignakel CLK endres fra 0 til 1 og forbli stabil tilstrekkelig lenge etter at klokkesignalet har blitt endret. Setup- og hold tid er sikkerhetsmarginer for å sikre riktig vippe funksjon.

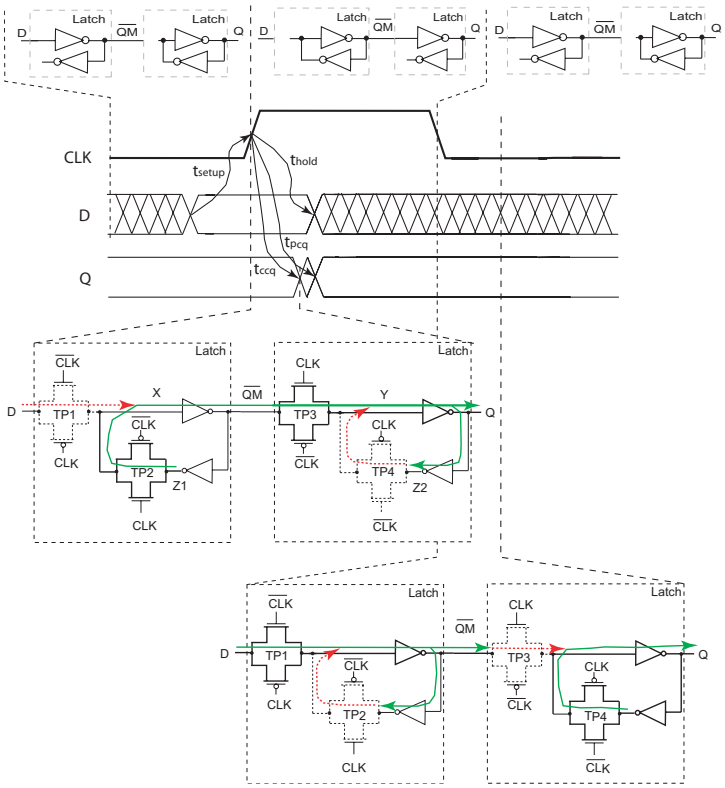

Fig. 21. *Timing for vippe. (FIG7.4b)*

Timing detaljer for vipper er vist i Fig. 21, der vippene er realisert som to latcher i motfase. Når klokkesignalet CLK er lavt vil inngangslatcher i vippenene være transparente og kontinuerlig sample inn $\overline{QM} = D$, samtidig med at utgangen Q oppfriskes ved hjelp av to invertere i tilbakekobling. Vippens funksjon er å sample inn D i slutten av tidsperioden når $CLK = 0$, dvs. vi ønsker å laste inn verdien D akkurat når CLK endres fra 0 til 1. Transmisjonsport 1 (TP1) stenger og TP2 åpner slik at signalet som ligger på $Z1 = QM$ vil påvirke X via TP2 og deretter $\overline{QM}$. Vi har en situasjon der $Z1 = X (= D)$. Dersom D forandres og påvirker X før TP2 åpner helt kan vi få en endring i X og deretter i $\overline{QM}$ som er uønsket. En hold tid på D vil sikre at $\overline{QM}$ ikke endres uønsket. I tillegg vil det være gunstig at TP1 stenger før TP2 åpner slik at X ikke kan drives via TP1 når tilbakekoblingen skal være aktiv. For latchen som kontrollerer utgangen på vippen (Q) har vi nå en situasjon der TP3 skal overstyre TP4. Det kan være gunstig at TP4 stenger før TP3 åpner for å redusere støy på utgangen. Vi ser at contamination forsinkelse og propageringsforsinkelse er tilnærmet like og er tidsforsinkelsen fra $\overline{QM}$ til Q når TP3 åpner, dvs fra stigende klokkeflanke. Ved fallende klokkeflanke vil det kunne oppstå en transparent situasjon dersom TP3 og TP1 er åpne samtidig. Det er viktig at TP3 stenger før TP1 åpner for å sikre at ikke inngangsignalet D påvirker Q. Vi ønsker i denne situasjonen når $CLK = 0$ at vippen skal holde sin lagrede verdi.

Ideelt vil vi ønske hele klokkeperioden tilgjengelig for prosessering av signaler. Sekvenseringen vil medføre en viss overhead som er knyttet til sekvenseringselementene. Dersom tidsforsinkelsen i kombinatorisk logikk er for stor vil en vippe som skal lagre resultatet få signalet for sent slik at situasjonen ikke tilfredstiller setup tid. Vi kaller en slik feil for set feil eller maks forsinkelse feil.

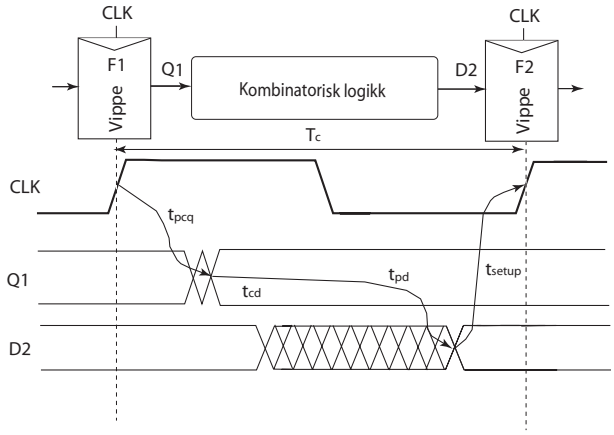

Fig. 22. *Begrensninger for maks forsinkelse. (FIG7.5)*

Begrensninger for maks forsinkelse er vist i Fig. 22. Dersom vi antar at vippene $F1$ og $F2$ styres av identiske klokkesignaler, dvs. at klokkesignalene har transisjoner på nøyaktig samme tidspunkt, er maksimal tid fra en vippe til neste vippe gitt av T_c:

$$T_c \geq t_{pcq} + t_{pd} + t_{setup}, \tag{5}$$

der t_{pcq} er propageringsforsinkelse for klokke til utgang (Q) for vippe, t_{pd} er propageringsforsinkelse i kombinatorisk logikk og t_{setup} er setup tid for vippe.

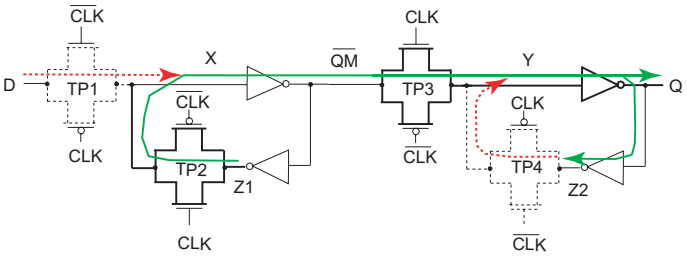

Fig. 23. *Propageringsforsinkelse for klokke til utgang (Q) for vippe.*

Propageringsforsinkelse for klokke til utgang (Q) for vippe er vist i Fig. 23. Når klokkesignalet er lavt vil inngangslatchen sample inn D til $\overline{QM}$ kontinuerlig. Når klokke signalet endres fra 0 til 1 lukker transmisjonsportene $TP1$ og $TP4$ mens $TP2$ og $TP3$ åpner. verdien som er ligger på $\overline{QM}$ (lagret) vil transmitteres via $TP3$ til Q. Propageringstidsforsinkelsen i dette tilfellet er gitt av en transmisjonsport og en inverter i kjede og last på Y og utgangen Q.

Propageringsforsinkelse i kombinatorisk nettverk mellom de to vippene i Fig. 22 er gitt av en den signalveien mellom $Q1$ og $D2$ som har størst tidsforsinkelse. Vi kaller den aktuelle signalveien en *kritisk signalvei*.

Setuptiden for vippen er en sikkerhetsmargin for å sample riktig verdi i inngangslatchen i vippen. Setuptid varierer med prosess og implementasjon av vippen.

Vi kan uttrykke begrensning for maks tidsforsinkelse på en annen form:

$$t_{pd} \quad \leq \quad T_c - (t_{setup} + t_{pcq}), \tag{6}$$

der $(t_{setup} + t_{pcq})$ er overhead ved sekvenseringen. Vi ser at ved å redusere t_{setup} og t_{pcq} til et minimum får vi mer av tiden som er tilgjengelig til å prosessere signaler i kombinatorisk logikk (gjøre beregninger).

Tiden tilgjengelig i en klokkeperiode T_c er direkte avhengig av t_{setup} og t_{pcq} for vipper og tidsforsinkelse i kritisk signalvei i kombinatorisk logikk mellom vipper. Klokkefrekvens, eller maksimal klokkefrekvens, er direkte knyttet til klokkeperioden:

$$f_{max} \quad = \quad \frac{1}{T_c}. \tag{7}$$

Term	Vippe	Latch
t_{ccq}	35ps	35ps
t_{pcq}	50ps	50ps
t_{pdq}		40ps
t_{setup}	65ps	25ps
t_{hold}	30ps	30ps

TABLE II
Parameterverdier for vipper.

Eksempel på parameterverdier for vipper er vist i tabell II. Vi kjenner klokkeperioden som er 500ps. Maksimal propageringsforsinkelse er gitt av

$$\begin{aligned} t_{pd} \quad &\leq \quad T_c - (t_{setup} + t_{pcq}) \\ &\leq \quad 500ps - (65ps + 50ps) \\ &\leq \quad 385ps. \end{aligned}$$

Maksimal propageringsforsinkelse for vippen er 385ps.

Vi har til nå forutsatt ideelle klokkesignaler uten *skew* eller forskyvning. I praksis vil klokkeflankene komme til litt forskjellige tider for ulike latcher og vipper. Dette skyldes at sekvenseringselementene vil være plassert ulike steder på en krets og dermed representere ulik last[2] for klokkesignalet.

I Fig. 24 er det vist et system med vipper som styres av et klokkesignal hvor tykk linje for CLK markerer det seneste tidspunktet for klokketransisjonene. Klokketransisjonene kan komme tidligere som vist i figuren. Den kritiske situasjonen for maksimal forsinkelse i et system med vipper er om vippen som sender et signal får klokketransisjonen sent og mottager vippen får klokketransisjonen tidlig. I dette tilfellet må klokkeskew trekkes fra den tiden systemet har tilgjenglig for å prosessere signaler i kombinatorisk logikk mellom vippene. Man regner da klokkeskew som en del av ovreheaden ved sekvenseringen.

Den kritiske situasjonen for minimum tidsforsinkelse har vi når sendervippen får klokketransjonen tidlig og mottager vippen får klokketransisjonen sent som vist i Fig. 25. I dette tilfellet vil den effektive holdtiden øke og vi får begrensingene:

$$t_{pd} \quad \leq \quad T_c - (t_{pcq} + t_{setup} + t_{skew}), \tag{8}$$

[2]Spesielt ulik kapasitans og motstand på grunn av interkonnekt.

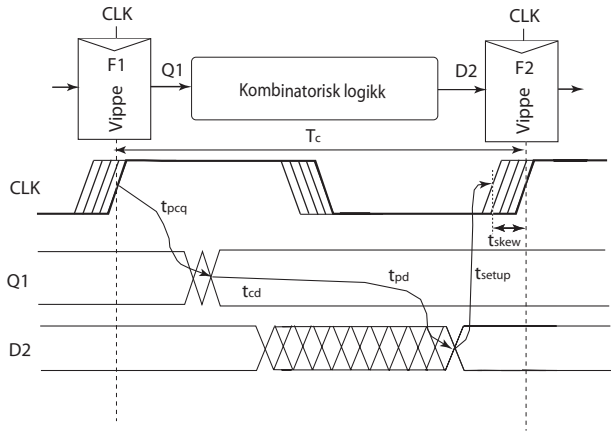

Fig. 24. *Klokkeskew og vipper. (FIG7.15a).*

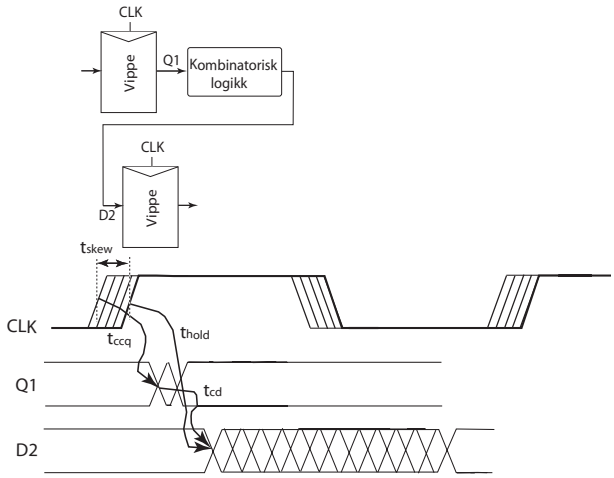

Fig. 25. *Klokkeskew og vipper. (FIG7.15b).*

der $(t_{pcq} + t_{setup} + t_{skew})$ er overhead i sekvenseringen. Vi har da:

$$t_{cd} \quad \geq \quad t_{hold} - t_{ccq} + t_{skew}. \tag{9}$$

Dersom vippen har en klokkeskew lik 50 ps får vi

$$\begin{aligned} t_{pd} \quad &\leq \quad T_c - (t_{pcq} + t_{setup} + t_{skew}) \\ &\leq \quad 500ps\,(50ps + 65ps + 50ps) \\ &\leq \quad 335ps. \end{aligned}$$

Maksimal propageringsforsinkelse for vippen med klokkeskew blir 335ps.

Begrensninger for minimum tidsforsinkelse for vipper er vist i Fig. 26 der vi antar at klokkesignalene til de to vippene er helt i fase (like).

Detaljer for et system med to vipper som er koblet sammen uten (minimalt) kombinatorisk logikk for $CLK = 0$ er vist i Fig. 27. Vi ser at så lenge $CLK = 0$ vil latch 1-1 i den første vippen følge inngangen D, dvs. vi latcher inn D i $\overline{Q1M}$. Utgangen på den første vippen $Q1$ holdes stabil ved hjelp av tilbakekobling i latch 1-2, og $Q1$ føres via eventuell kombinatorisk logikk til den neste vippen, nærmere bestemt latch 2-1 som latcher (sampler) inn $Q1$ til $\overline{Q2M}$. Vi ser at i slutten av perioden hvor $CLK = 0$ vil TP1, TP4, TP5 og TP8 være helt åpne og TP2, TP3, TP6 og

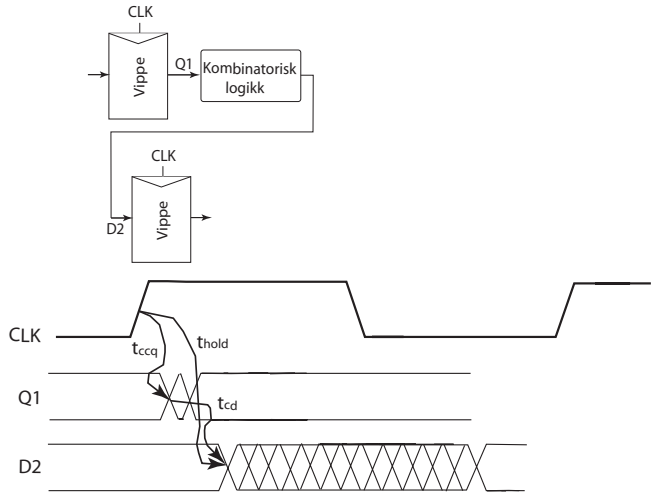

Fig. 26. *Begrensinger på minimumsforsinkelse for vipper. (FIG7.9).*

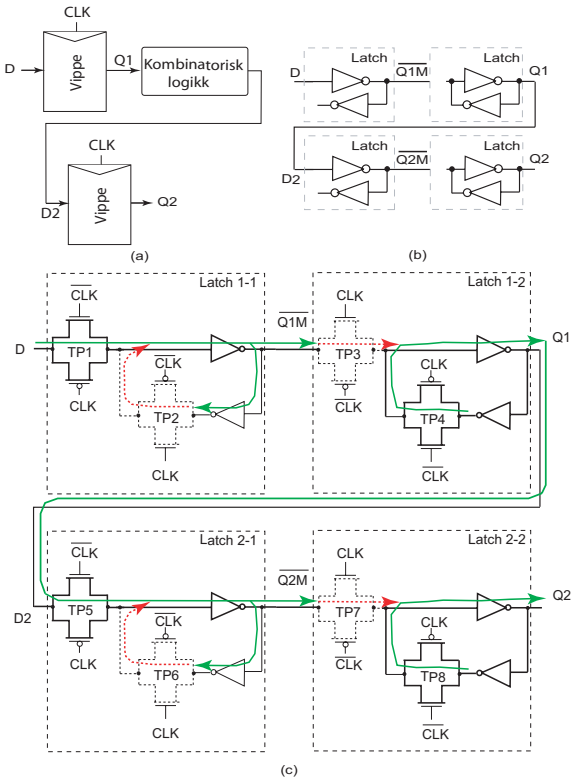

Fig. 27. *Begrensinger på minimumsforsinkelse for vipper. Detaljer i timing når $CLK = 0$. (FIG7.9).*

TP7 være helt lukket. Vi må forutsette at inngangen D er stabil en stund før stigende klokkeflanke (t_{setup}). En kritisk situasjon som medfører feil er dersom TP3 og TP5 er åpne samtidig slik at $\overline{Q1M}$ blir transmittert til $\overline{Q2M}$ via $Q1$. Vippenes funksjon er å lagre verdier i etterfølgende klokkeperioder.

Dersom vi ser på detaljene rundt stigende klokkeflanke, som er vist i Fig. 28, ser vi at $\overline{Q1M}$ skal endre $Q1$ med klokke til Q contamination forsinkelse t_{ccq}. Det vil si at vi får en endring, men ikke nødvendigvis stabil verdi, på $Q1$ ved tidspunktet t_{ccq} etter stigende klokketransisjon. Ideelt sett har nå den neste vippen lukket TP5 og kan ikke påvirkes av endringer på $D2 = Q1$. For å sikre at en vippe ikke endres feilaktig er det påkrevd

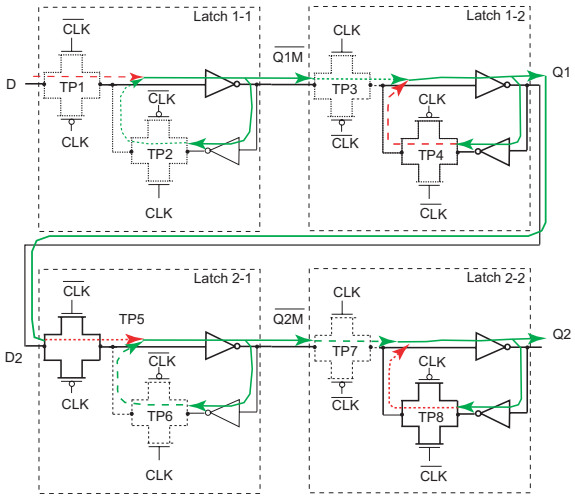

Fig. 28. *Begrensinger på minimumsforsinkelse for vipper. Detaljer i timing når $CLK = 0 \rightarrow 1$. (FIG7.9).*

at det defineres en hold tid for inngangen. I dette tilfellet betyr det at den siste vippen forutsetter at inngangen $D2$ er stabil en liten stund etter stigende klokkeflanke. Det er avgjørende at ikke $D2$ endres som følge av endring på $Q1$ idet ved stigende klokkeflanke før vippens setup tid er over. Vi kan uttrykke dette som

$$t_{cd} \geq t_{hold} - t_{ccq}, \tag{10}$$

der t_{cd} er contamination forsinkelse i kombinatorisk logikk[3] mellom vippene. Med andre ord, det er viktig at tidsforsinkelsen mellom vippene er så stor at inngangen til vippe nummer 2 ikke har har fått ny verdi fra latch 1-1 før setup tiden til vippe 2 er over. Dersom $D2$ endres før setuptiden er over vil latch 1-2 og latch 2-1 være transparente samtidig slik at $\overline{Q2M}$ blir lik $\overline{Q1M}$, som vil medføre at $Q2$ blir lik $Q1$ i neste omgang. Dersom contamination forsinkelse, dvs. klokke til Q forsinkelse, for vippen er større enn hold tid kan vippene plasseres helt inntil hverandre. I dette tilfellet vil ikke vippe nummer 2 rekke å reagere på endringer på inngangen for tidlig.

Vi har følgende uttrykk for vippe

$$\begin{aligned} t_{cd} &\geq t_{hold} - t_{ccq} \\ &\geq 30ps - 35ps \\ &\geq 0. \end{aligned}$$

Minimum contamination forsinkelse er 0.

[3] I eksemplet er det ikke kombinatorisk logikk mellom vippene slik at contamination forsinkelse blir minimum forutsatt at ikke avstanden mellom vippene er stor.

C. Forklar hvordan en Partovi puls latch virker

C.1 Løsningsforslag

Teori

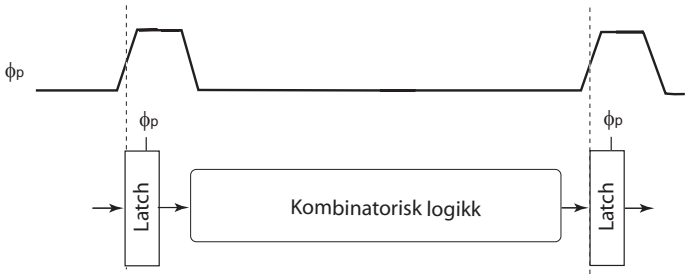

Fig. 29. *Latch som styres av klokkepuls.*

En latch som styres av klokkepulser minner om en konvensjonell transparent latch. Et slikt system er avhengig av forholdsvis stor tidsforsinkelse i kombinatorisk logikk mellom latchene som vist i Fig. 29. For at to latcher med kombinatorisk logikk mellom latchene ikke skal være transparent må det settes krav til tidsforsinkelse i den kombinatoriske logikken:

$$t_{cd} \quad \geq \quad t_{hold} - t_{ccq} + t_{pw}, \tag{11}$$

der t_{cd} er contamination (minimum) forsinkelse i kombinatorisk logikk, t_{hold} er hold tid for inngang fra negativ klokkeflanke, t_{ccq} er klokke til utgang contamination forsinkelse for latchen og t_{pw} er pulsbredden på klokkesignalet.

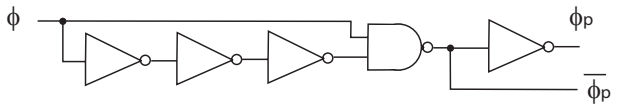

Fig. 30. *Enkel puls generator. (FIG7.22a)*

Med utgangspunkt i et klokke signal ϕ med dutycycle lik 50% kan vi generere en klokke med pulser $\leq 50\%$ som vist i Fig. 30. I dette tilfellet blir invertert klokkesignal også generert.

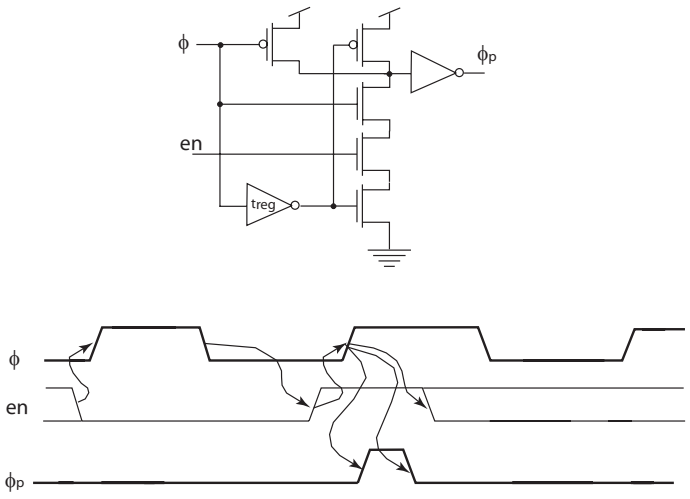

Fig. 31. *Puls generator. (FIG7.22b)*

En annen puls generator er vist i Fig. 31. Denne puls generatoren genererer pulser med meget kort bredde.

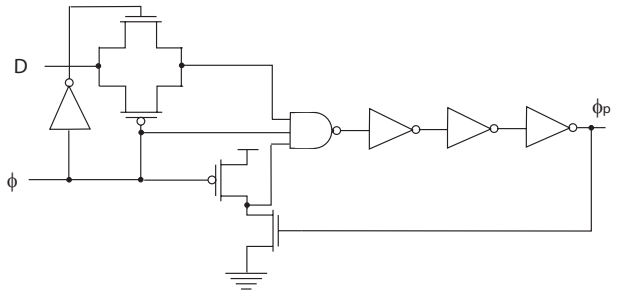

Fig. 32. *Puls generator. (FIG7.22d)*

En tredje pulsgenerator med betydelig bredere pulser er vist i Fig. 32.

Ulike pulsgeneratorer med forskjellig pulsbredder passer til ulike spesielle latcher.

D. Puls latch

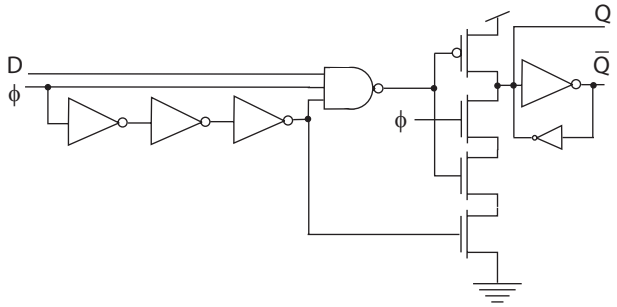

Fig. 33. *Partovi puls latch. (FIG7.23)*

Et eksempel på en latch som er styrt av klokkepulser er vist i Fig. 33. Dette er en såkalt Partovi puls latch som har pulsgeneratoren innebygd i selve latchen.

Timing for latcher styrt av pulser

Begrensninger for maks forsinkelse i et sekvenseringssystem med latcher styrt av pulser er vist i Fig. 34. Dersom pulsbredden t_{pw} er større enn setuptiden t_{setup} har vi situasjonen som vist øverst i figuren. Vi trenger i dette tilfellet ikke å forutsette at utgangen av kombinatorisk logikk $D4$ er stabil før klokkepulsen kommer. Dersom klokkepulsen er mindre enn setup tiden har vi situasjonen som vist nederst, og vi må da forutsette at inngangen til latchen er stabil før klokkepulsen kommer. Vi kan beskrive dette som:

$$T_c \quad \geq \quad \max\left(t_{pdq} + t_{pd}, t_{pcq} + t_{pd} + t_{setup} - t_{pw}\right), \tag{12}$$

som gir:

$$t_{pd} \quad \leq \quad T_c - \max\left(t_{pdq}, t_{pcq} + t_{setup} - t_{pw}\right), \tag{13}$$

der $\max\left(t_{pdq}, t_{pcq} + t_{setup} - t_{pw}\right)$ er overhead inkludert forsinkelse i en latch.

Eksempel på parameterverdier for vipper er vist i tabell III.

Vi kjenner klokkeperioden som er 500ps. Maksimal propageringsforsinkelse for latch styrt med pulser er gitt av

$$t_{pd} \quad \leq \quad T_c - \max\left(t_{pdq}, t_{pcq} + t_{setup} - t_{pw}\right),$$

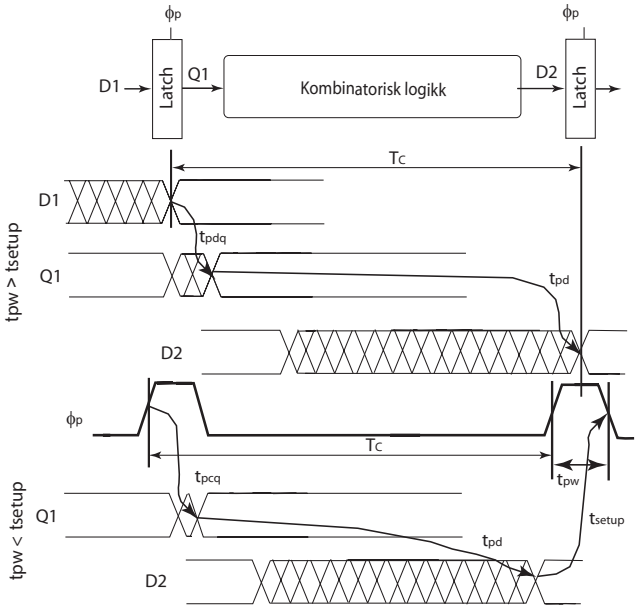

Fig. 34. *Begrensninger for maks forsinkelse i et sekvenseringssystem med latcher styrt av pulser. (FIG7.8).*

Term	Vippe	Latch
t_{ccq}	35ps	35ps
t_{pcq}	50ps	50ps
t_{pdq}		40ps
t_{setup}	65ps	25ps
t_{hold}	30ps	30ps

TABLE III
Parameterverdier for latch styrt av klokkepulser.

der pulsbredden t_{pw} er 80ps. Vi har da $t_{pdq} = 40ps$ og $t_{pcq} + t_{setup} - t_{pw} = 50ps + 25ps - 80ps = -5ps$, som gir

$$\begin{aligned} t_{pd} &\leq T_c - \max\left(t_{pdq}, t_{pcq} + t_{setup} - t_{pw}\right) \\ &\leq 500ps - 40ps \\ &\leq 460ps. \end{aligned}$$

Maksimal propageringsforsinkelse for latch styrt med pulser med bredde er 460ps.

Latcher som styres av klokkepulser er vist i Fig. 35. Det er tilsvarende begrensninger for minimum tidsforsinkelse i kombinatorisk logikk mellom latchene som for tofase latcher. Vi kan utrykke dette som:

$$t_{cd} \geq t_{hold} - t_{ccq} + t_{pw}. \tag{14}$$

For latcher styrt med pulser har vi

$$\begin{aligned} t_{cd} &\geq t_{hold} - t_{ccq} + t_{pw} \\ &\geq 30ps - 35ps + 80ps \\ &\geq 75ps. \end{aligned}$$

Som forventet blir minimum contamination forsinkelse for latcher styrt med pulser større enn 0, dvs. i dette tilfellet 75ps.

Klokke skew kan komme som et direkte tillegg i begrensing i maksimal propageringsforsinkelse for latch styrt med pulser. Vi har

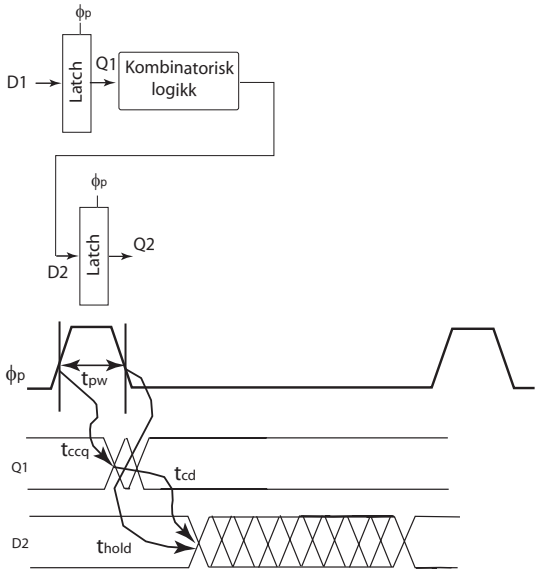

Fig. 35. *Begrensninger på minimumsforsinkelse for latcher som er styrt av klokkepulser. (FIG7.11).*

$$\begin{aligned} t_{pd} &\leq T_c - \max\left(t_{pdq}, t_{pcq} + t_{setup} - t_{pw} + t_{skew}\right) \\ &\leq 500ps - (-5ps + 50ps) \\ &\leq 455ps. \end{aligned}$$

Som vi ser påvirker klokkeskew maksimal propageringsforsinkelse lite i dette tilfellet.

E. Forklar hvordan en vippe med asynkron set og reset virker

E.1 Løsningsforslag

Teori

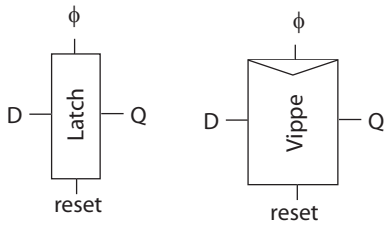

Fig. 36. *Symboler for latch og vippe med reset signal. (FIG7.24)*

Det er praktisk å kunne benytte et reset signal slik at tilstanden til et sekvenseringselement er kjent ved oppstart. Symboler for latch og vippe med reset signal er vist i Fig. 36. Det er to typer av reset:

- *Synkron reset.* Synkrone reset signaler må være stabile for setup- og hold tid ved klokkeflanker.

- *Asynkron reset.* Asynkrone reset signaler resetter et element uavhengig av klokkesignaler.

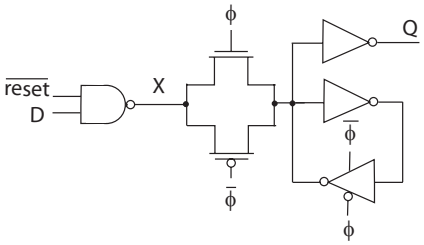

Fig. 37. *Synkron latch med reset signal. (FIG7.24)*

En latch med synkron reset er vist i Fig. 37. Som kjent er ikke latchen følsom for inngangen når $\phi = 0$. NAND porten på inngangen av latchen vil slippe gjennom D når $\overline{reset}$ er 1, vi har for en 2inngangs NAND port (NAND2) $X = \overline{D \cdot \overline{reset}}$ som gitt at $\overline{reset} = 1$ kan forenkles til $X = \overline{D}$. Når $\overline{reset} = 0$ kan uttrykket for NAND porten forenkles til $X = 1$. Når transmisjonsporten åpner for $\phi = 1$ vil latchen sample inn enten $\overline{D}$ eller 1. I det siste tilfellet skal latchen resettes slik at utgangen $Q = 0$ uavhengig av D. Vi legger merke til at latchen ikke resettes før $\phi = 1$.

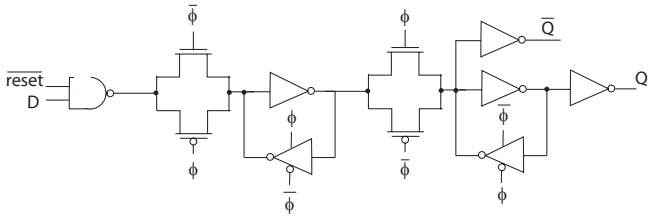

Fig. 38. *Synkron vippe med reset signal. (FIG7.24)*

En vippe med synkron reset er vist i Fig. 38. For inngangslatchen i vippen gjelder samme argumentasjon som for synkron reset av latch, men der inngangslatchen er klokket i motfase[4].

[4]Inngangslatchen kan resettes når $\phi = 0$.

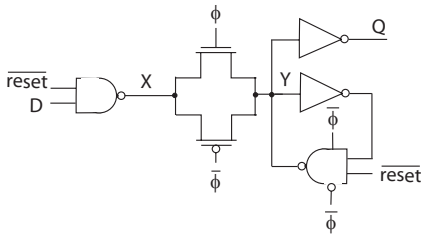

Fig. 39. *Asynkron latch med reset signal. (FIG7.24)*

En latch med asynkron reset er vist i Fig. 39. NAND porten på inngangen fungerer som beskrevet for latch med synkron reset, dvs. Q via Y og X settes til 0 når $reset = 1$ og $\phi = 1$. Latchen blir da resatt via transmisjonsporten på inngangen. Det er i tillegg plassert en dynamisk NAND port i tilbakekoblingen slik at noden Y kan settes til 1, og dermed utgangen Q settes til 0 når $reset = 1$ og $\phi = 0$. Dette betyr at utgangen Q settes til 0 når $reset = 1$ uavhengig av D og ϕ.

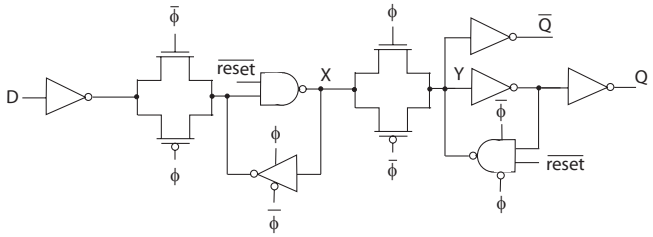

Fig. 40. *Aynkron vippe med reset signal. (FIG7.24)*

En vippe med asynkron reset er vist i FIG. 40. Inngangslatchen vil presse noden X til 1 når $reset = 1$ uavhengig av D, ϕ og (tidligere) verdi på X. Når $\phi = 0$ vil noden Y få verdien 1. Vi har da en situasjon der noden Y blir satt til 1 fra X via transmisjonsporten når $\phi = 1$ eller fra den dynamiske NAND porten når $\phi = 0$. Dette betyr at Y blir resatt til 1 uavhengig av ϕ og Q blir resatt til 1. Legg merke til at denne vippen resettes til 1 når $reset = 1$.

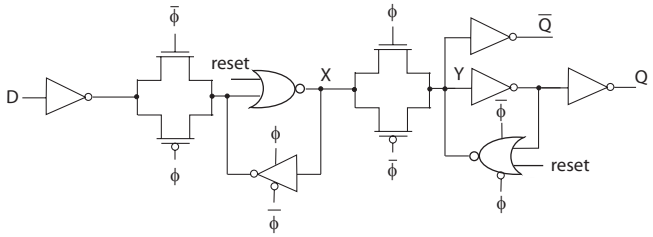

Fig. 41. *Aynkron vippe med reset signal.*

Vi kunne ha byttet ut NAND portene med NOR porter og $\overline{reset}$ med $reset$, som vist i Fig. 41, slik at noden Y ble resatt til 0 for å få resatt utgangen Q til 0 når $reset = 1$.

I Fig. 42 er en vippe med asynkron set og reset vist. Kretsen benytter to signaler set og $reset$ til å sette vippen i to ulike tilstander. Inngangslatchen har $\overline{set}$ signal som styrer NAND porten som setter noden X til 1 når $set = 1$. For utgangslatchen vil $\overline{set}$ signalet sette noden Y lik 1. $\overline{set}$ signalet setter utgangen på C^2MOS NAND porten i tilbakekoblingen i inngangslatchen til 1 når $reset = 1$ samtidig som NAND porten i utgangslatchen sette inngangen til inverteren til 1 og dermed utgangen Q til 0.

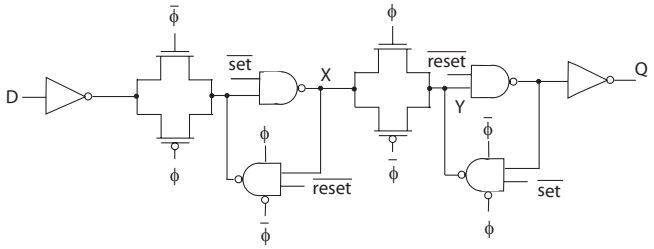

Fig. 42. *Vippe med asynkron reset og set signal.*

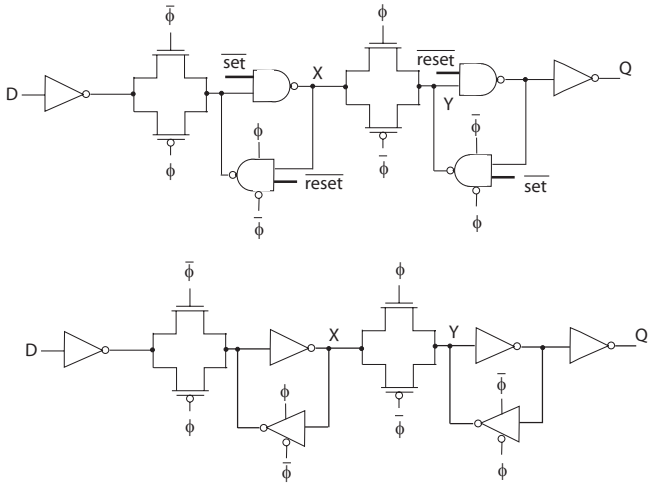

Fig. 43. *Vippe med asynkron reset og set signal.* $Set = reset = 0$

Vippen med $set = reset = 0$ er vist øverst i figur 43. For alle NAND portene vil det være en av inngangene som er 1, dvs. $\overline{set} = \overline{reset} = 1$. Forenklet port og logisk ekvivalent, men ikke elektrisk, ekvivalent, er en inverter som vist i den nederste kretsen i Fig. 43.

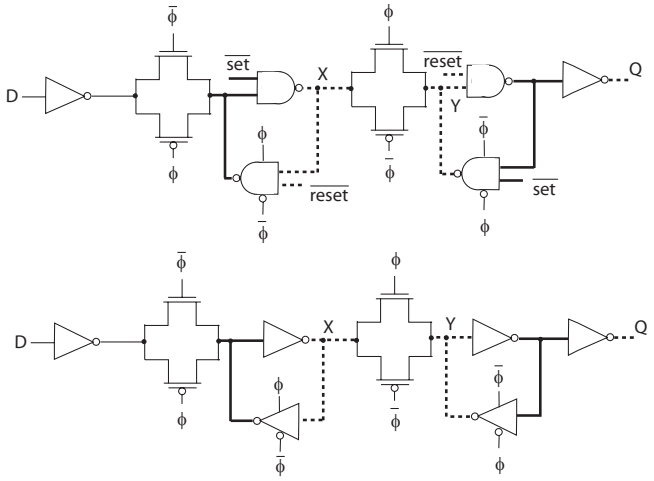

Fig. 44. *Vippe med asynkron reset og set signal.* $Reset = 1$

Vippen i reset funksjon er vist i Fig. 44 øverst. I dette tilfellet forutsetter vi at det andre kontrollsignalet $set = 0$. For inngangslatchen vil da utgangen på C^2MOS NAND porten i tilbakekoblingen bli satt til 1 slik at den andre NAND porten i inngangslatchen vil sette X til 0. Dette betyr at inngangslatchen vil bli resatt til 0 som er tilsvarende som om vi samplet inn 0 fra inngangen. For utgangslatchen vil NAND porten med $\overline{reset}$ som inngang sette inngangen til inverteren til 1 og dermed utgangen Q til 0. C^2MOS NAND porten i tilbakekoblingen i utgangslatchen vil sørge for at Y blir lik 0 (som er samme verdi som X). Dersom kontrollsignalet $reset$ settes til 0 etter at kretsen er korrekt resatt vil vippen være i tilstanden vist i Fig. 44 nederst[5] inntil vippen eventuelt sampler inn en ny verdi $D = 1$ når $\phi = 0$, eller vippen settes til 1 ved hjelp av kontrollsignalet set.

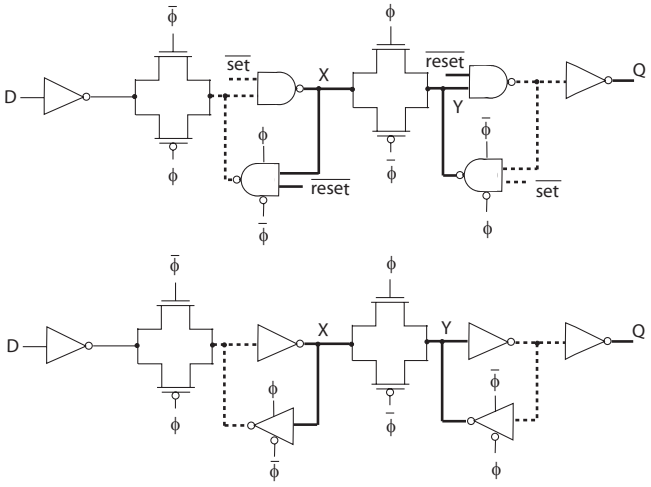

Fig. 45. *Vippe med asynkron reset og set signal.* $Set = 1$

Vippen i set funksjon er vist i Fig. 45 øverst. I dette tilfellet forutsetter vi at det andre kontrolsignalet $reset = 0$. Inngangslatchen vil sette noden X til 1 som igjen vil sette utgangen til C^2MOS NAND porten i tilbakekoblingen til 0. Denne verdien vil holde seg lik 0 gjennom forenklet logisk ekvivalent vist for inngangslatchen i Fig. 45 nederst. For utgangslatchen vil C^2MOS NAND porten i tilbakekoblingen sørge for at $Y = 1$ som vil sette inngangen til utgangsinverteren til 0 og dermed blir utgangen Q lik 1. Vippen nederst i figuren er logisk ekvivalent inntil inngangslatchen sampler inn ny verdi $D = 0$ når $\phi = 0$, eller vippen resettes ved hjelp av kontrolsignalet $reset$.

Vi har nå forutsatt at vippen kan ha tre ulike modi:

- **Vippe.** $set = reset = 0$. Vippen fungerer som en vanlig vippe som vist i Fig. 43 nederst.

- **Resett til 0.** $reset = 1$ og $set = 0$. Vippen resettes til 0, dvs. både utgangen og noden X resettes til 0. Når reset signalet endres til 0 vil kretsen operere som kretsenekvivalenten vist nederst i Fig. 44.

- **Sett til 1.** $set = 1$ og $reset = 0$. Vippen resettes til 1, dvs. både utgangen og noden X resettes til 1. Når reset signalet endres til 0 vil kretsen operere som kretsekvivalenten vist nederst i Fig. 45.

Det er en kombinasjon av kontrollsignalene som vi ikke har vurdert. Dersom vi antar at vippen har kontrollsignalene $reset = set = 1$ har vi en situasjon som ikke kan tillates. Vippen skal i denne situasjonen både settes til 1 og 0 som er selvmotsigende og meningsløst. For ordens skyld kan det være fornuftig å analysere vippen for å se hva som skjer dersom vi ved en feil påtrykker denne ulovelige kombinasjonen av kontrollsignaler.

[5]Kretsen nederst er logisk ekvivalent når $set = reset = 0$, men ikke elektrisk ekvivalent.

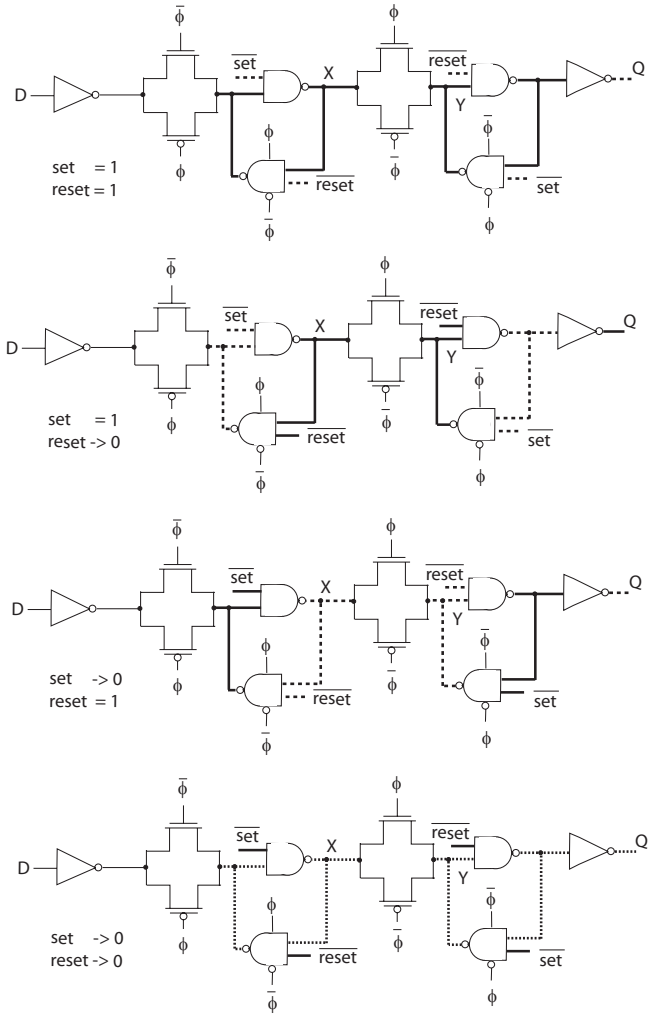

Fig. 46. *Vippe med asynkron reset og set signal. Set = Reset = 1*

I Fig. 46 øverst er det vist hvordan vippen virker når $set = reset = 1$. Vi ser at utgangen Q blir resatt til 0 som i utgangspunktet ligner en vanlig reset. Legg merke til at nodene X og Y blir satt til 1 samtidig. Dette samsvarer ikke med en vanlig reset. Vippens tilstand før eventuell ny sampling av inngangen er avhengig av hvilke av de to kontrollsignalene som skrus av først. Dersom $reset$ blir satt til 0 mens $set = 1$, som vist nest øverst i figuren, vil kretsen oppføre seg som om den ble satt til 1 slik at utgangen Q blir satt til 1. Dersom set blir satt til 0 mens $reset = 1$, som vist nest nederst i figuren, vil kretsen oppføre seg som om den ble satt til 0 slik at utgangen Q forblir 0. I den nederste vippen er det antatt at set og $reset$ endres fra 1 til 0 samtidig. Situasjonen vil da være ukjent, dvs. vi kan ikke forutsi om vippen blir satt til 0 eller 1.

F. Hvordan virker en differensiell sense-amplifier vippe?

F.1 Løsningsforslag

Teori

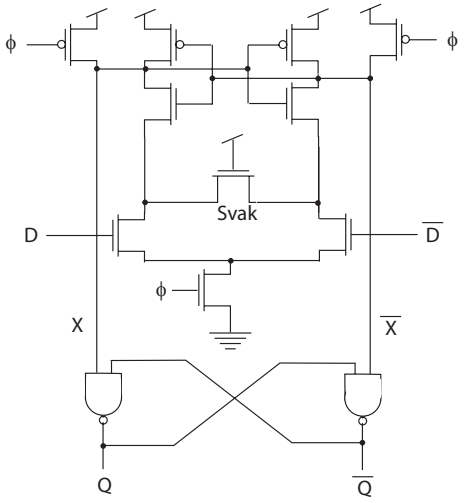

Fig. 47. *Differensiell sense-amplifier vippe. (FIG7.29a)*

En *differensiell* vippe er vist i Fig. 47. Vippen er basert på en såkalt *sense amplifier* som består at et inngangstrinn med D og $\overline{D}$ med en felles transistor med ϕ inngang ned mot GND. De to NAND portene brukes til å holde utgangene stabile.

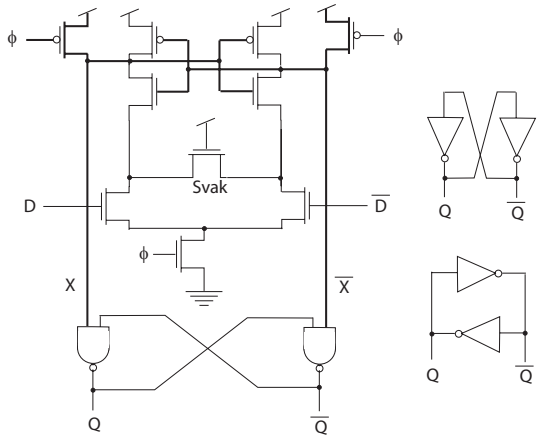

Fig. 48. *Differensiell sense-amplifier vippe med $\phi = 0$. (FIG7.29a)*

Den differensielle vippen med $\phi = 0$ er vist i Fig. 48. Når$\phi = 0$ vil nodene X og $\overline{X}$ precharges til 1 slik at de to NAND portene kan forenkles logisk som vist til høyre for vippen. De to kretsene med krysskoblete invertere er identiske og tilsvarer utgangslatchen på en vanlig vippe.

Den differensielle vippen med $\phi = 1$ er vist i Fig. 49. Vippen skal nå sample inn ny verdi. Som vi ser er vippen fullstendig symmetrisk, vi ser derfor på eksemplet der $D = 1$ som vist i figuren. Noden X blir trukket ned til 0 og dermed $\overline{X}$ opp til 1. Utgangen $Q = D$ som vil holdes til sampling ved neste positive klokkeklanke. Det er en viss tidsforsinkelse gjennom de to NAND portene på utgangene.

Den differenensielle vippen i Fig. 50 har en raskere respons enn vippen med krysskoblete NAND porter. Utgangen blir

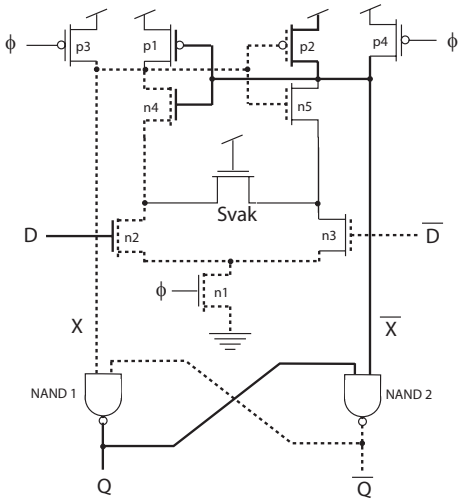

Fig. 49. *Differensiell sense-amplifier vippe $\phi = 1$. (FIG7.29a)*

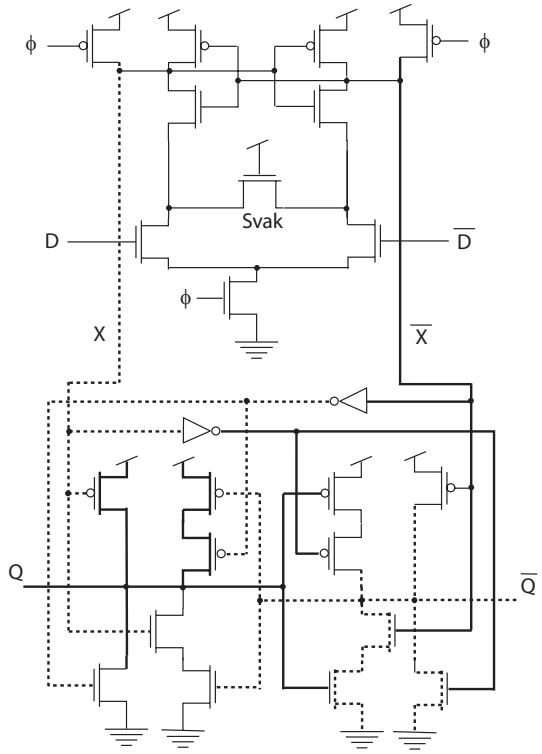

Fig. 50. *Differensiell sense-amplifier vippe. (FIG7.29b)*

trukket opp til via en pMOS transistor direkte fra X. De to krysskoblete portene bidrar til å holde verdien i vippen.

G. Forklar hvordan en ekte en-fase vippe virker

G.1 Løsningsforslag

Teori

Vanlige latcher og vipper benytter i tillegg til klokkesignal også invertert klokkesignal. I moderne CMOS blir typisk invertert klokkesignal generert lokalt ved latchene eller vippene.

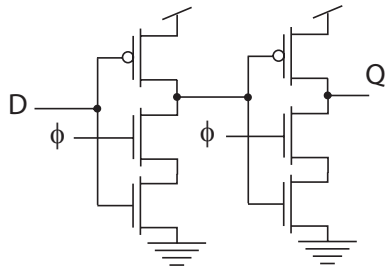

Fig. 51. *Ekte en-fase latch. (FIG7.30a)*

En latch som kun benytter ett klokkesignal er vist i Fig. 54. Vi kaller dette for en *ekte en-fase latch*.

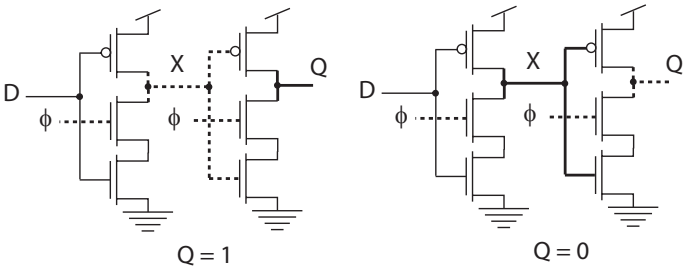

Fig. 52. *Ekte en-fase latch med $\phi = 0$. (FIG7.30a)*

En ekte en-fase latch med $\phi = 0$ er vist i Fig. 53. Latchen skal holde utgangen stabil så lenge $\phi = 0$. Vi ser at nedtrekkene er skrudd AV ved hjelp av ϕ. I utgangspunktet har vi to mulige tilstander; Q var 1 før ϕ skiftet fra 1 til 0 (som vist på venstre side) og Q var 0 opprinnelig (som vist på høyre side). En forutsetnig for at $Q = 1$ er at $X = 0$ som vist til venstre. Når nedtrekket koblet til utgangen ikke kan trekkes ned pga. ϕ vil ikke kretsen kunne endre utgangen så lenge $\phi = 0$. Legg merke til at latchen er dynamisk slik at lekkasje kan påvirke utgangssignalet etter en viss tid. Utgangen holdes høy ved hjelp av en pMOS transistor som er skrudd på forsi $X = 0$. Noden X er ikke drevet og kan endres som følge av lekkasje og dermed påvirge utgangen Q.

Til høyre er en tilstand der $Q = 0$ og $X = 1$. I dette tilfellet er hverken Q eller$X$ drevet og derfor utsatt for lekkasje.

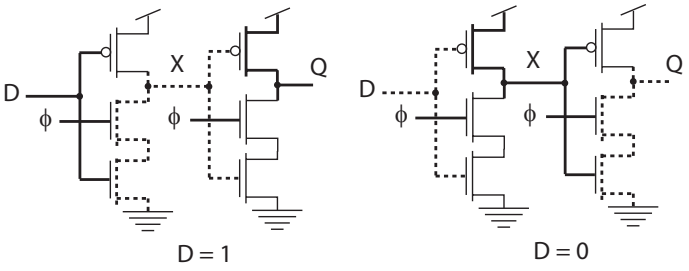

Fig. 53. *Ekte en-fase latch med $\phi = 1$. (FIG7.30a)*

Ved latching av ny verdi (sampling) er er $\phi = 1$ som vist i Fig. 53. I denne situasjonen kan vi (logisk) se bort i fra

transistorene styrt av klokke signalet ϕ. Kretsen vil da logisk være to invertere i serie slik at vi alltid får $Q = D$.

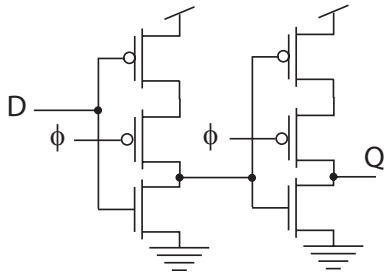

Fig. 54. *Ekte en-fase latch. (FIG7.30b)*

En latch som er følsom for motsatt klokkenivå er vist i Fig. 54. Istedet for å bruke samme latch med invertert klokke signal erstatter vi de klokkestyrte nMOS transistorene med pMOS transistorene og flytter utgangene mellom pMOS og nMOS transistorer.

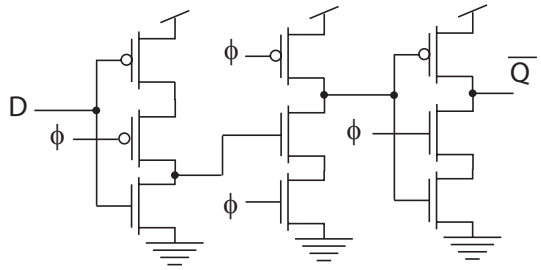

Fig. 55. *Ekte en-fase vippe. (FIG7.30c)*

En ekte en-fase vippe er vist i Fig. 55. Denne vippen er enda enklere en to en-fase latcher klokket i motfase.

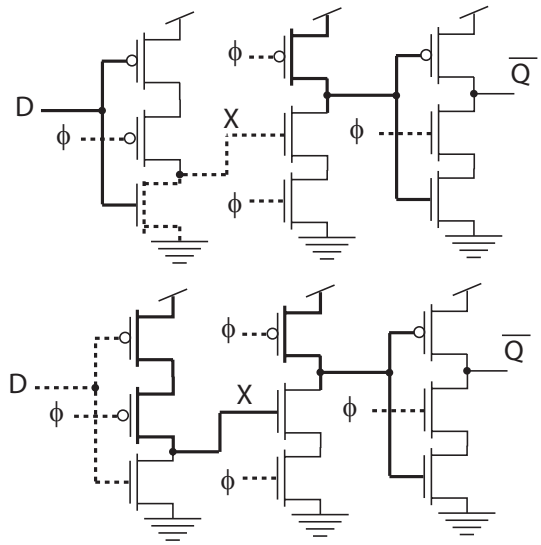

Fig. 56. *Ekte en-fase vippe når $\phi = 0$. (FIG7.30c)*

En-fase vippen nåt $\phi = 0$ og $\phi = 1$ er vist i henholdsvis Fig. 56 og 57.

En-fase latcher og vippe som er beskrevet i dette avsnittet er dynamiske.

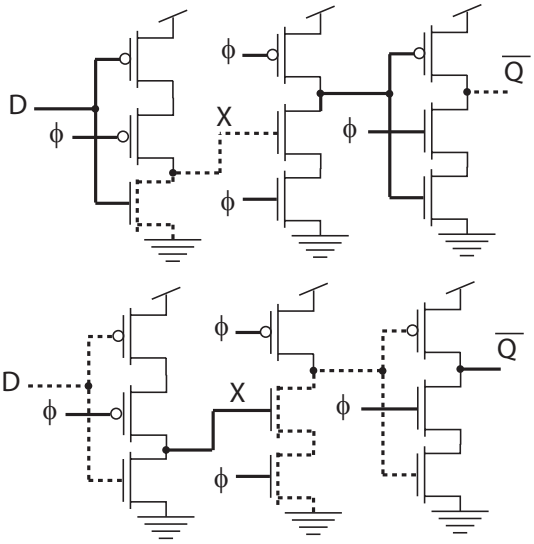

Fig. 57. *Ekte en-fase vippe når $\phi = 1$. (FIG7.30c)*

References

[1] Neil H.E. Harris og David Harris"CMOS VLSI DESIGN, A circuit and system perspective" tredje utgave 2005, ISBN: 0-321-26977-2, *Addison Wesley*,