

INF3400 Digital Mikroelektronikk

Løsningsforslag DEL 10

YNGVAR BERG

Del 10: Sekvensielle kretser

I. OPPGAVER

A. Oppgave 7.1

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

TABLE I

Parameterverdier for sekvenseringselementer.

Anta parameterverdier som gitt i tabell I. Finn maksimal propageringsforsinkelse innenfor en 500ps klokkeperiode for de følgende sekvenseringssystemene:

1. Vipper.
2. To-fase transparente latcher.

Anta at det ikke er klokke-skew.

A.1 Løsningsforslag

Teori

Det er ulike metoder for å synkronisere signaler eller ulike sekvenseringsmetoder som kan anvendes. Synkroniseringselementene eller sekvenseringselementene som benyttes er typisk latch eller vippe.

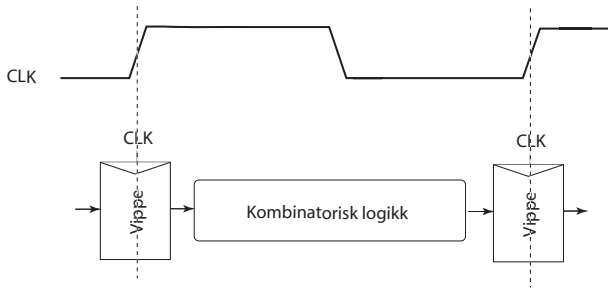


Fig. 1. Sekvensering (synkronisering) med vipper.

I Fig. 1 er det vist kombinatorisk logikk synkronisert ved hjelp av kantfølsomme vipper. Synkroniseringspunktet er gitt av en positiv klokkeflanke. Ved en positiv klokkeflanke vil vip-pene kopiere inngangssignalet til utgangen og holde verdien inntil neste positive klokkeflanke. Ved å invertere synkroniseringssig-nalene vil man få synkroniseringspunkt ved negative klokke-flanker.

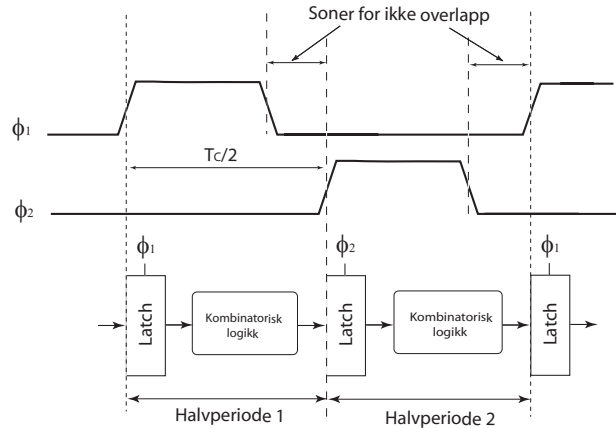


Fig. 2. Sekvensering (synkronisering) med latcher.

Synkronisering ved hjelp av latcher er vist i Fig. 2. I et to-fase system er det vanlig å bruke to-fase ikke-overlappende klokker, der klokkefasene (klokkesignalene ϕ_1 og ϕ_2) ikke er høye samtidig. Hele klokkeperioden er T_c . En halvperiode vil bestå av en tid der en av klokkesignalene er høye etterfulgt av en tid der ingen av klokkesignalene er høye (ikke-overlappende).

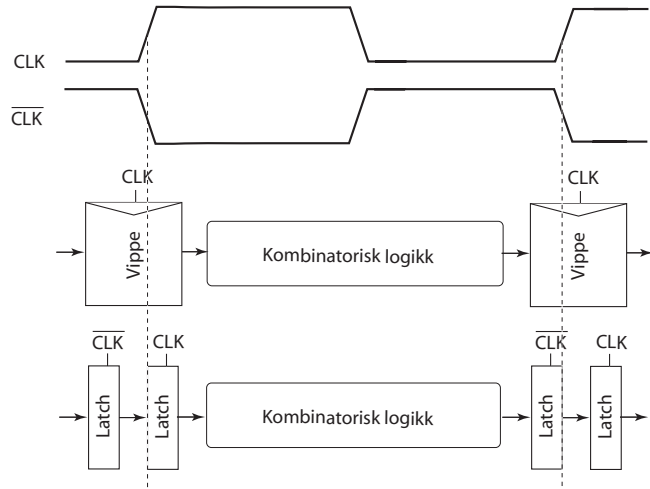


Fig. 3. Sekvensering (synkronisering) med vipper realisert som latcher med klokke og invertert klokke.

Sekvensering ved hjelp av vipper styrt av klokke (CLK) og invertert klokke (\overline{CLK}) er vist i figur 3. Her er to latch 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} .

I tabell II er de ulike uttrykkene (termene) for forsinkelse som legger begrensinger for timing i sekvensielle kretser.

Term	Kommentar
t_{pd}	Logisk propagering forsinkelse
t_{cd}	Logisk contamination forsinkelse
t_{pcq}	Latch/vippe klokke til Q propagering forsinkelse
t_{ccq}	Latch/vippe klokke til Q contamination forsinkelse
t_{pdq}	Latch D til Q propagering forsinkelse
t_{cdq}	Latch D til Q contamination forsinkelse
t_{setup}	Latch/vippe setup tid
t_{hold}	Latch/vippe hold tid

TABLE II

Notasjon for sekvenseringselementer.

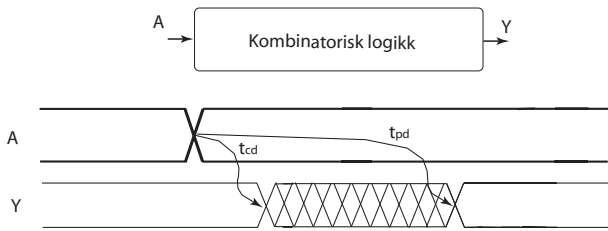


Fig. 4. Timing for kombinatorisk logikk.

Tidsforsinkelse i kombinatorisk logikk er gitt av *contamination forsinkelse* og *propagering forsinkelse* og vist i Fig. 4. Utgangen kan begynne å endre seg etter tiden gitt av contamination forsinkelse t_{cd} og utgangen vil bli stabil etter maksimal tidsforsinkelse fra inngang til utgang som er gitt av propageringsforsinkelse t_{pd} . Contamination forsinkelse og propageringsforsinkelse kan være forskjellig på grunn av forskjellige kjeder eller signalveier i logikken.

Timing for vippe

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.

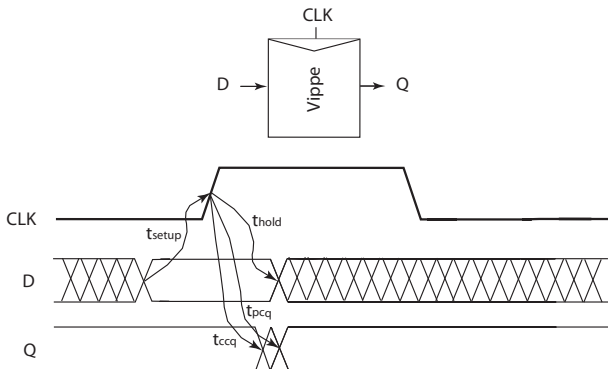


Fig. 5. Timing for vippe.

Tidsforsinkelse i en vippe er vist i Fig. 5. Vippet 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. Vippet er avhengig av *setup-* og *hold tid*. Dette betyr at inngangen D må være stabil en stund før klokkesignalet 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 vippefunksjon.

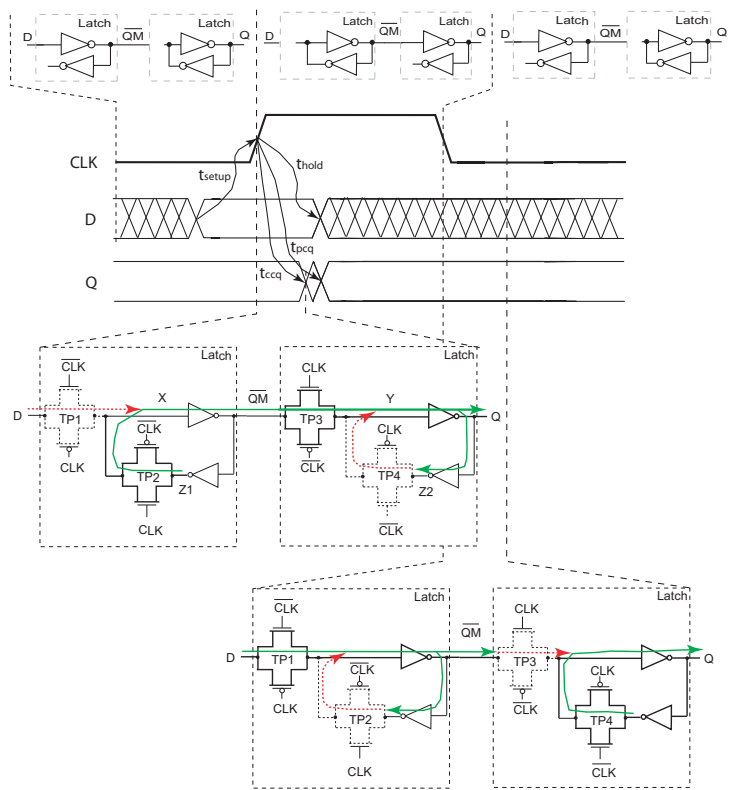


Fig. 6. Timing for vippe.

Timing detaljer for vipper er vist i Fig. 6, der vippene er realisert som to latches i motfase. Når klokkesignalet CLK er lavt vil inngangslatches i vippene 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 CLK 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 vippet (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 inngangssignalet D påvirker Q . Vi ønsker i denne situasjonen når $CLK = 0$ at vippet skal holde sin lagrede verdi.

Begrensninger for maks forsinkelse for vippe

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.

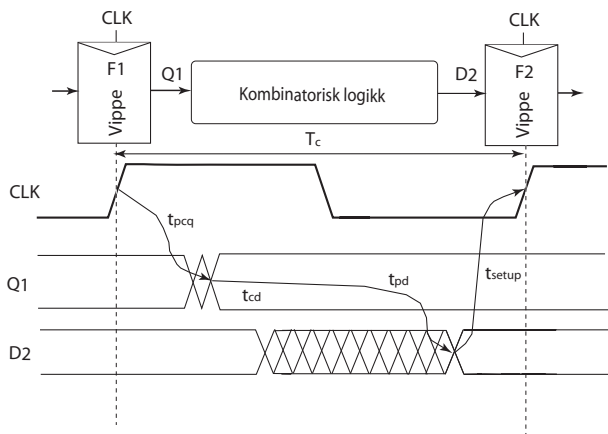


Fig. 7. Begrensninger for maks forsinkelse.

Begrensninger for maks forsinkelse er vist i Fig. 7. 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 :

$$T_c \geq t_{pcq} + t_{pd} + t_{setup}, \quad (1)$$

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.

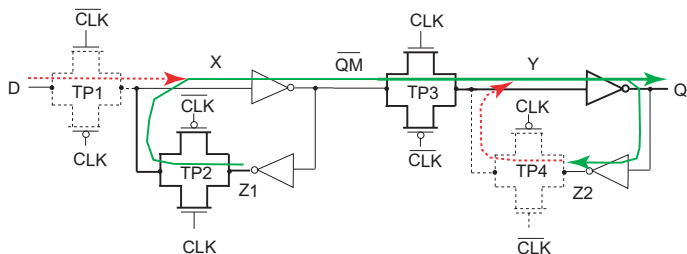


Fig. 8. Propageringsforsinkelse for klokke til utgang (Q) for vippe.

Propageringsforsinkelse for klokke til utgang (Q) for vippe er vist i Fig. 8. Når klokkesignalet er lavt vil inngangslatchen samle 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. 7 er gitt av en den signalveien mellom $Q1$ og $D2$ som har størst tidsforsinkelse. Vi kaller den aktuelle signalveien en *kritisk signalvei*.

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

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

$$t_{pd} \leq T_c - (t_{setup} + t_{pcq}), \quad (2)$$

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 vippene og tidsforsinkelse i kritisk signalvei i kombinatorisk logikk mellom vippene. Klokkefrekvens, eller maksimal klokkefrekvens, er direkte knyttet til klokkeperioden:

$$f_{max} = \frac{1}{T_c}. \quad (3)$$

Beregning av maksimal propageringsforsinkelsen for vippe

Vi kjenner klokkeperioden som er 500ps. Maksimal propageringsforsinkelse er gitt av

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

Maksimal propageringsforsinkelse for vippene er 385ps.

Timing for Latch

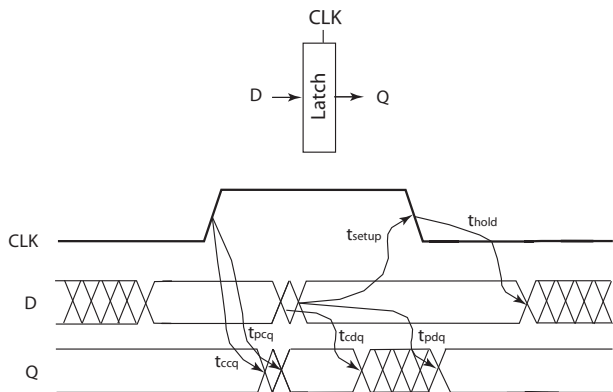


Fig. 9. Timing for latch.

Timing for en latch er vist i Fig. 9. 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.

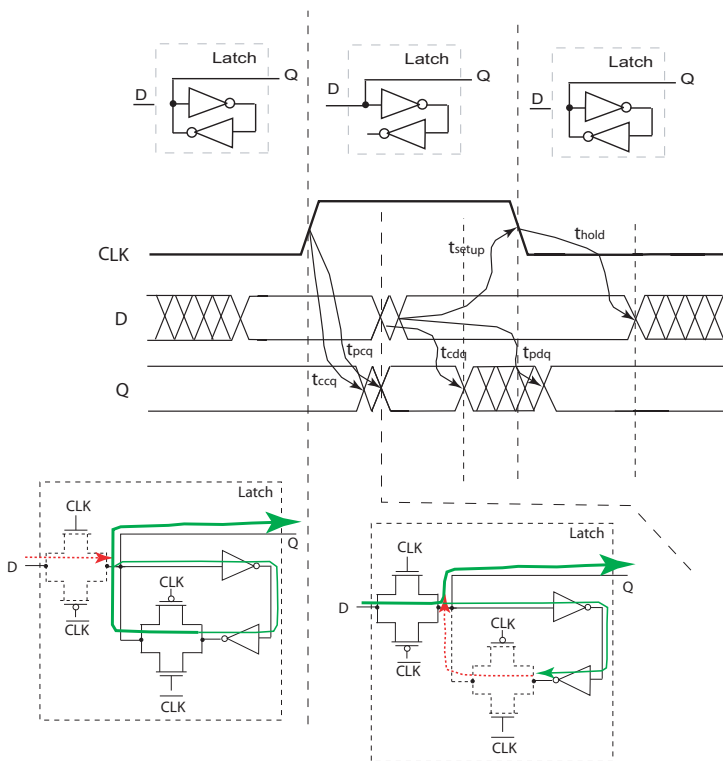


Fig. 10. Timing for latch.

Timingdetaljer for en latch er vist i Fig. 10. 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. 2 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. 10. Latchen er transparent når klokkesignalet er høyt og enhver forandring på D vil påvirke Q . Vi må forutsette at D er stabil en liten stund 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.

Begrensninger for maks forsinkelse for latcher

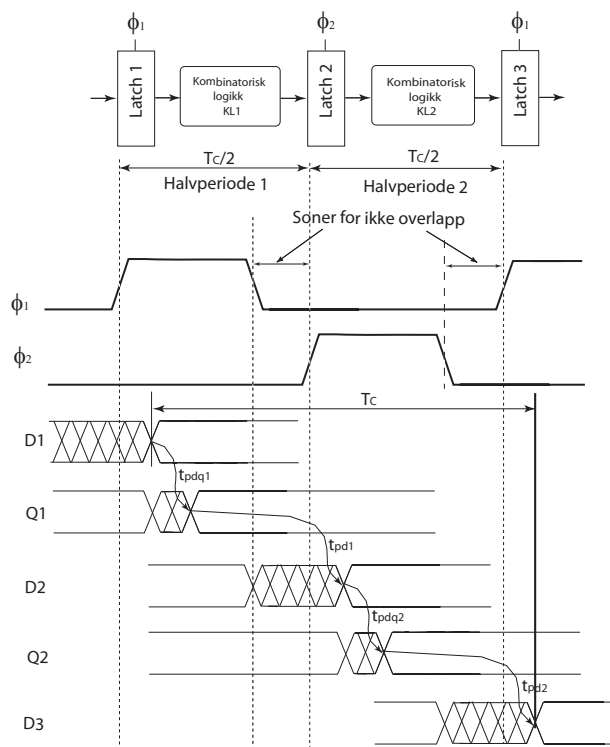


Fig. 11. Begrensninger for maks forsinkelse i et sekvenseringssystem med latcher styrt av tofase klokker.

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 tidsforsinkelse 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 samlet (latchet) i latch 2. Tilsvarende argumentasjon gjelder for latch 3 osv. Vi kan uttrykke en klokkeperiode T_c som:

$$T_c \geq t_{pdq1} + t_{pd1} + t_{pdq2} + t_{pd2}. \quad (4)$$

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

$$\begin{aligned} t_{pd} &= t_{pd1} + t_{pd2} \\ &\leq T_c - (2t_{pdq}), \end{aligned} \quad (5)$$

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

Beregning av maksimal propageringsforsinkelse for to-fase transparent latch

Vi kjenner klokkeperioden som er 500ps. Maksimal propageringsforsinkelse for to-fase transparent latch er gitt av

B. Oppgave 7.2

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

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

Gjenta forrige oppgave, men anta at klokke-skew mellom to elementer kan være 50ps.

B.1 Løsningsforslag

Teori

Vi har til nå forutsatt ideelle klokkesignaler uten *skew* eller forskyvning. I praksis vil klokkeflankene komme til litt forskjellige tider for ulike latches og vipper. Dette skyldes at sekvenseringselementene vil være plassert ulike steder på en krets og dermed representere ulik last¹ for klokkesignalet.

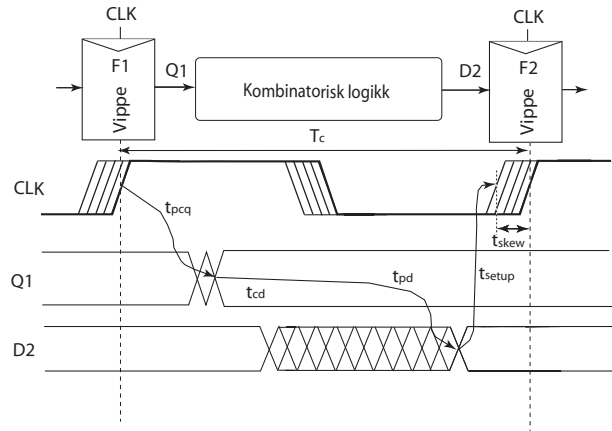


Fig. 12. Klokkeskew og vipper.

I Fig. 12 er det vist et system med vipper som styres av et klokkesignal hvor tykk linje for *CLK* markerer det seneste tidspunktet for klokkeovergangen. Klokkeovergangene kan komme tidligere som vist i figuren. Den kritiske situasjonen for maksimal forsinkelse i et system med vipper er om vippene som sender et signal får klokkeovergangen sent og mottager vippene får klokkeovergangen tidlig. I dette tilfellet må klokkeskew trekkes fra den tiden systemet har tilgjengelig for å prosessere signaler i kombinatorisk logikk mellom vippene. Man regner da klokkeskew som en del av overheaden ved sekvenseringen.

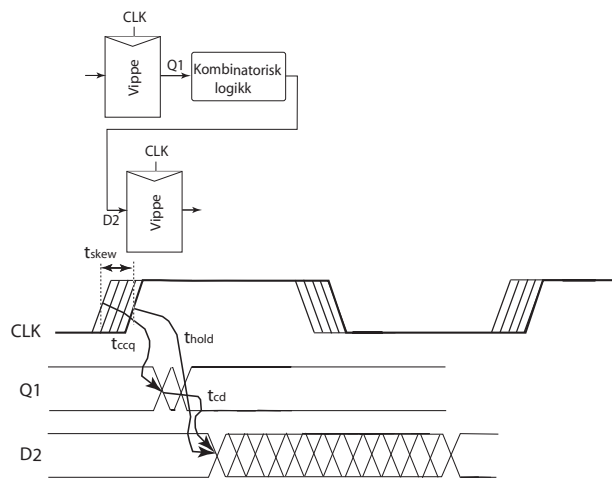


Fig. 13. Klokkeskew og vipper.

¹Spesielt ulik kapasitans og motstand på grunn av interkonnekt.

Den kritiske situasjonen for minimum tidsforsinkelse har vi når sendervippen får klokkeovergangen tidlig og mottager vippen får klokkeovergangen sent som vist i Fig. 13. I dette tilfellet vil den effektive holdtiden øke og vi får begrensningene:

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

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

$$t_{cd} \geq t_{hold} - t_{ccq} + t_{skew}. \quad (7)$$

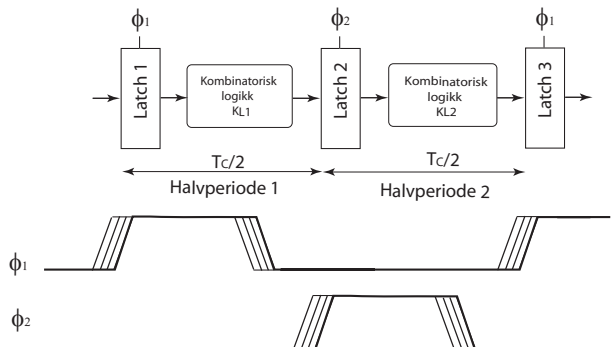


Fig. 14. Klokkeskew og transparente latches.

I et tofase system med transparente latches vil ikke klokkeskew redusere ytelsen, som vist i Fig. 14, så lenge de to klokkefasene er ikke-overlappende. Vi kaller et system med transparente latches med tofase ikke-overlappende klokker for *skew tolerant*. Muligheten for å fordele tid mellom klokkefasene vil imidlertid bli mer begrenset:

$$\begin{aligned} t_{pd} &\leq T_c - 2t_{pdq} \\ t_{cd1}, t_{cd2} &\geq t_{hold} - t_{ccq} - t_{ikkeoverlapp} + t_{skew}, \end{aligned} \quad (8)$$

der t_{cd} er contamination forsinkelse for kombinatorisk logikk mellom latches.

For latches som styres av klokkepulser vil fordeling av tid i kombinatorisk nettverk mellom latches bli svært begrenset fordi klokkeskew kan føre til at klokkepulserne blir svært smale.

Beregning av maksimal propageringsforsinkelse for vippe og klokkeskew lik 50ps

Dersom vippene har en klokkeskew lik 50 ps får vi

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

Maksimal propageringsforsinkelse for vippene med klokkeskew blir 335ps.

Beregning av maksimal propageringsforsinkelse for to-fase transparent latch og klokkeskew lik 50ps

Vi har

$$\begin{aligned} t_{pd} &\leq T_c - 2t_{pdq} \\ &\leq 420ps. \end{aligned}$$

Vi ser at for en to-fase transparent latch representerer ikke klokkeskew en ekstra begrensning i maksimal propageringsforsinkelse.

C. Oppgave 7.3

Bestem minimum logisk contamination forsinkelse for hver klokkeperiode (halve klokkeperioden for to-fase latches) for følgende sekvenseringsmetoder:

1. Vippen.
2. To-fase transparente latches med klokkesignaler med 50% duty cycle.
3. To-fase transparente latches med klokkesignaler med ikke-overlappende tidsperiode på 60ps.

Anta at det ikke er klokke-skew.

D. Løsningsforslag

Begrensninger for minimum forsinkelse for vippe

Sekvenseringselementer bør kunne plasseres inntil hverandre uten vesentlig kombinatorisk logikk mellom elementene. Eksempel på sekvensielle systemer med minimal kombinatorisk logikk mellom sekvenseringselementene er *pipeline systemer*.

Dersom hold tid er stor og contamination forsinkelsen er liten kan data propageres gjennom to sekvenseringselementer ved en klokkeflanke. En slik feil kalles *race feil*, *hold tid feil* eller *minimum-forsinkelse feil*.

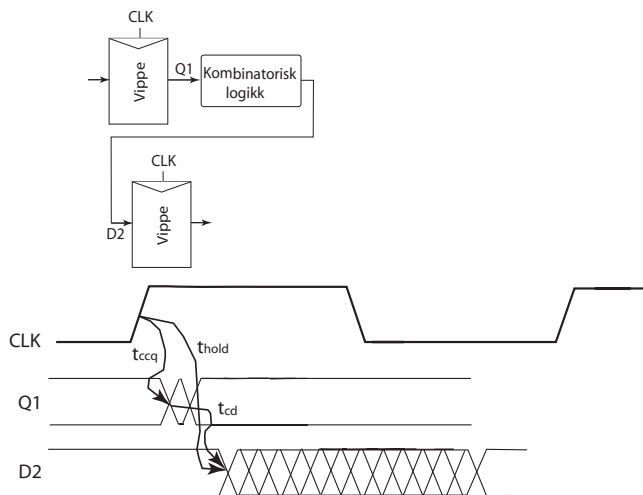


Fig. 15. Begrensninger på minimumsforsinkelse for vipper.

Begrensninger for minimum tidsforsinkelse for vipper er vist i Fig. 15 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. 16. Vi ser at så lenge $CLK = 0$ vil latch 1-1 i den første vippene følge inngangen D , dvs. vi latches inn D i $\overline{Q1M}$. Utgangen på den første vippene $Q1$ holdes stabil ved hjelp av tilbakekobling i latch 1-2, og $Q1$ føres via eventuell kombinatorisk logikk til den neste vippene, nærmere bestemt latch 2-1 som latches (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 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.

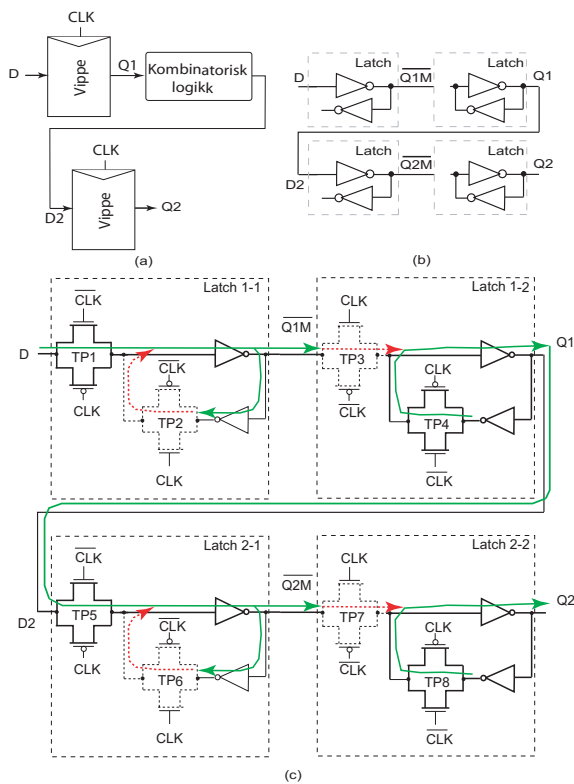


Fig. 16. Begrensinger på minimumsforsinkelse for vipper. Detaljer i timing når $CLK = 0$.

Dersom vi ser på detaljene rundt stigende klokkeflanke, som er vist i Fig. 17, 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 klokkeovergang. 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 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

$$t_{cd} \geq t_{hold} - t_{ccq}, \quad (9)$$

der t_{cd} er contamination forsinkelse i kombinatorisk logikk² mellom vippene. Med andre ord, det er viktig at tidsforsinkelsen mellom vippene er så stor at inngangen til vippe nummer 2 ikke 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 vippene 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.

Minimum logisk contamination forsinkelse for vippe

²I eksemplet er det ikke kombinatorisk logikk mellom vippene slik at contamination forsinkelse blir minimum forutsatt at ikke avstanden mellom vippene er stor.

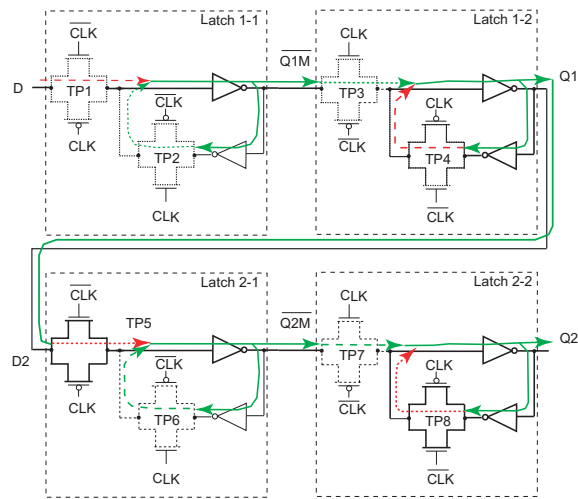


Fig. 17. Begrensinger på minimumsforsinkelse for vipper. Detaljer i timing når $CLK = 0 \rightarrow 1$.

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.

Begrensninger for minimum forsinkelse for latcher

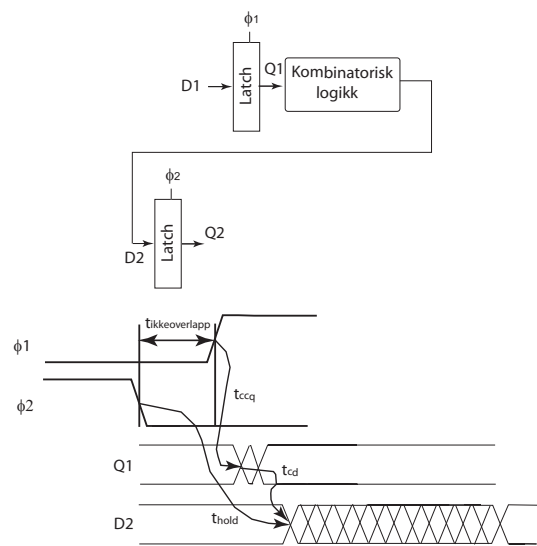


Fig. 18. Begrensninger på minimumsforsinkelse for latcher som er styrt av to fase klokker.

I Fig. 18 er begrensninger på minimumsforsinkelse for latcher som er styrt av to fase lokker vist. Latchene styres av to fase 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 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 latching ved tidspunktet når ϕ_1 svinger fra 0 til 1.

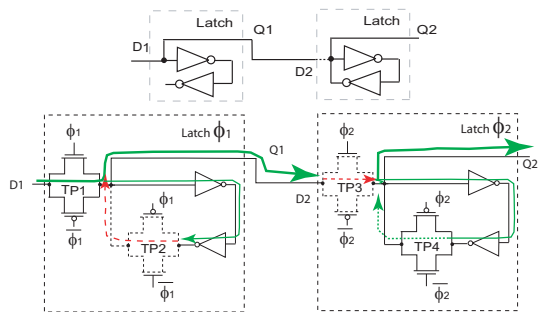


Fig. 19. Begrensninger på minimumsforsinkelse for latcher som er styrt av to fase klokke. Detaljer ved stigende transisjon på ϕ_1 .

Timing detaljer for latcher som er styrt av to fase klokke ved stigende transisjon på ϕ_1 er vist i Fig. 19. 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:

$$t_{cd1}, t_{cd2} \geq t_{hold} - t_{ccq} - t_{ikkeoverlapp}, \quad (10)$$

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 uttrykke dette som en begrensning på hold tiden:

$$t_{hold} \leq t_{ikkeoverlapp} + t_{ccq} + t_{cd}. \quad (11)$$

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.

Minimum logisk contamination forsinkelse for to-fase transparente latcher med klokkesignaler med 50% duty cycle

For to-fase transparente latcher har vi

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

Minimum contamination forsinkelse for latchen er 0.

Minimum logisk contamination forsinkelse for to-fase transparente latcher med klokkesignaler med ikke-overlappende tidsperiode på 60ps

For to-fase transparente latcher med klokkesignaler med ikke-overlappende tidsperiode på 60ps har vi

$$\begin{aligned} t_{cd1}, t_{cd2} &\geq t_{hold} - t_{ccq} - t_{ikkeoverlapp} \\ &\geq 30ps - 35ps - 60ps \\ &\geq 0. \end{aligned}$$

Minimum contamination forsinkelse for latchen er 0.

REFERENCES

- [1] Neil H.E. Harris og David M. Harris "Integrated Circuit Design" fjerde utgave 2010, ISBN 10: 0-321-69694-8, ISBN 13: 978-0-321-69694-6, Pearson.