

INF3400/4400 Digital Mikroelektronikk

Løsningsforslag DEL 12 og 13

Våren 2006

YNGVAR BERG

I. DEL 12 OG 13

Del 12 og 13: Passtransistor- og differensiell CMOS logikk og interkonnekt

II. OPPGAVER

A.

Tegn sjematikk for en 4:1 multiplekser med innganger D0, D1, D2 og D3, og to selekt signaler S0 og S1, i følgende logikkstiler:

1. CPL.
2. LEAP.
3. DPL.
4. EEPL.
5. PPL.
6. SRPL.
7. DCVSPG.
8. Statisk CMOS.

A.1 Løsningsforslag

Teori for CPL

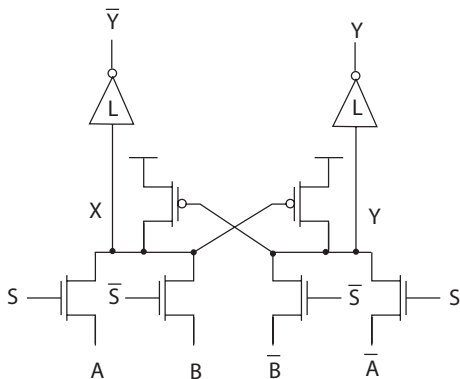


Fig. 1. Komplementær pass transistor logikk (CPL). (FIG6.51a)

Komplementær pass transistor logikk (CPL) er vist i Fig. 1. Vi ser at kretsen minner om CMOS TG. Inverterne på utgangen er lav skew der nedtrekket er prioritert.

CPL multiplekseren med $S = 0$ og $B = 1$ er vist i Fig. 2. Her er det to signalveier fra B og \bar{B} til henholdsvis \bar{Y} og Y . Vi ser først på signalveien fra \bar{B} til Y via Z . Signalet går via en nMOS transistor til Z som vil bli trukket helt ned til 0. X derimot kan ikke trekkes helt opp til 1 fordi nMOS transistoren som forbinder B med X vil ha et terskelfall. pMOS transistoren som er koblet til X og som har Z som gate signal vil bidra til å trekke X helt opp til 1.

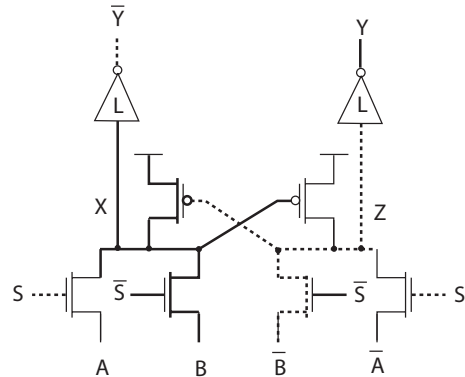


Fig. 2. Komplementær pass transistor logikk (CPL), $S = 0$ og $B = 1$. (FIG6.51a)

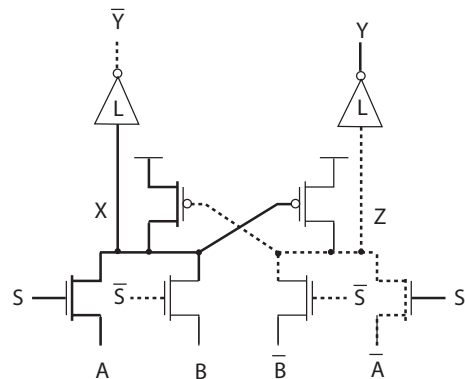


Fig. 3. Komplementær pass transistor logikk (CPL), $S = 1$ og $A = 1$. (FIG6.51a)

CPL multiplekseren med $S = 1$ og $A = 1$ er vist i Fig. 3. pMOS transistorene koblet til nodene X og Z bidrar til å trekke enten X eller Z til logisk 1, dvs. X i dette tilfellet. CPL kan implementeres uten de to pMOS transistorene. En av nodene X eller Z vil da bli trukket helt ned til 0, mens den andre vil bli trukket opp til $V_{DD} - V_t$. Denne reduserte logiske 1 vil bli invertert til en skikkelig logisk 0 etter utgangsinverteren. Utgangsinverteren som skal drive \bar{Y} til 0 vil da ikke bli skrudd helt på. Dette bidrar ikke bare til forsinkelse, men også til økt effektforbruk fordi inngangstransisjonen for utgangsinverteren blir treg.

4:1 CPL multiplekser

4:1 multiplekser implementert med komplementær pass transistor logikk (CPL) er vist i Fig. 4.

Eksempel på realisering av 4:1 multiplekser implementert med komplementær pass transistor logikk (CPL) er vist i Fig. 5.

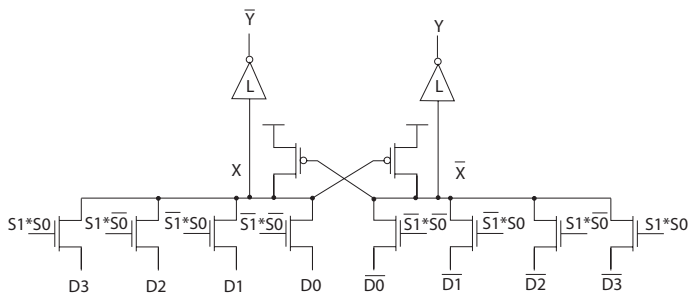


Fig. 4. 4:1 multiplexer implementert med komplementær pass transistor logikk (CPL).

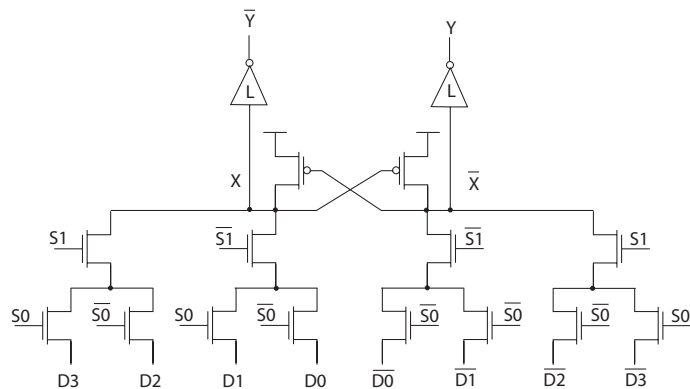


Fig. 6. 4:1 multiplexer implementert med komplementær pass transistor logikk (CPL).

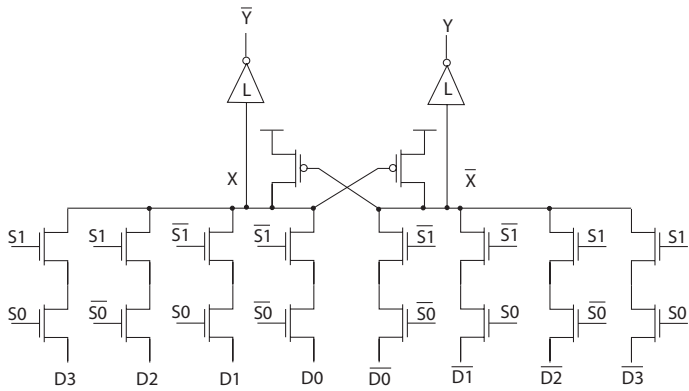


Fig. 5. 4:1 multiplexer implementert med komplementær pass transistor logikk (CPL).

Forenklet realisering i av 4:1 multiplexer implementert med komplementær pass transistor logikk (CPL) er vist i Fig. 6.

4:1 CPL XOR port

2inngangs XOR port implementert med komplementær pass transistor logikk (CPL) er vist i Fig. 7.

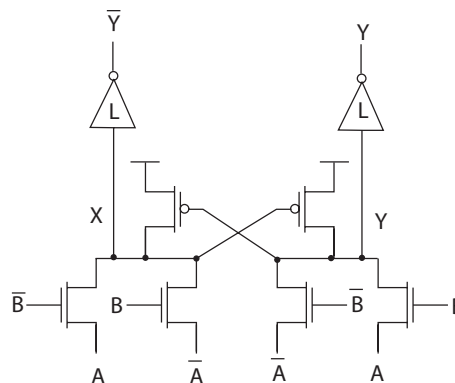


Fig. 7. Komplementær pass transistor logikk (CPL) XOR port. (FIG6.51a)

Teori for LEAP

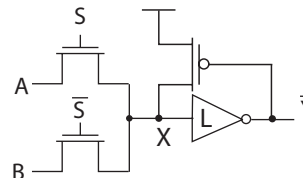


Fig. 8. Lean Integration med pass transistorer (LEAP) (FIG6.47)

En lean integration krets med pass transistorer (LEAP) er vist i Fig. 8. Transmisjonsportene er erstattet med passtransistorer. Node X blir presset opp til 1 når utgangen \bar{Y} er 0. På denne måten kan vi erstatte to pMOS transistorer ved inngangene med en transistor som er styrt fra utgangen.

LEAP multiplexer

4:1 multiplexer implementert med lean integration med pass transistorer (LEAP) er vist i Fig. 9.

LEAP XOR port

2inngangs XOR port implementert med lean integration med pass transistorer (LEAP) er vist i Fig. 10.

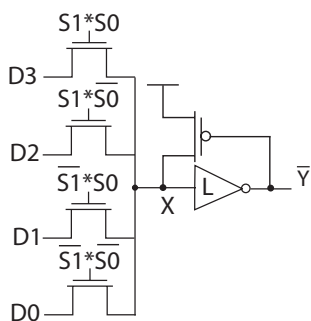


Fig. 9. 4:1 multiplexer implementert med lean integration med pass transistorer (LEAP) (FIG6.47)

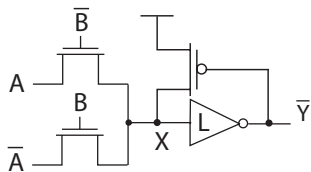


Fig. 10. 4:1 multiplexer implementert med lean integration med pass transistorer (LEAP) (FIG6.47)

Teori for DPL

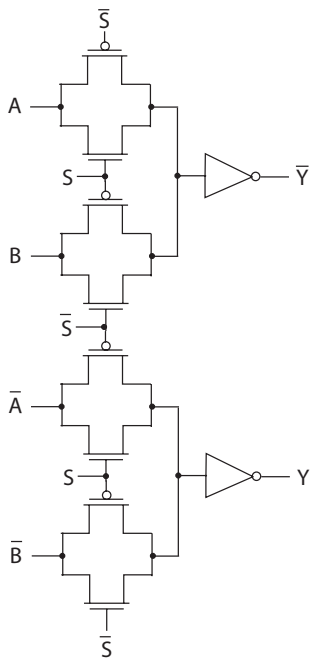


Fig. 11. Dobbel pass transistor logikk (DPL) (FIG6.47)

Dobbel pass transistor logic (DPL) er vist i Fig. 11. DPL ligner på CMOS, men består av dobbelt sett med transmisjonsporter når vi trenger inverterte utganger i tillegg til ordinære utganger. Kretsen trenger ikke tilbakekobling for å sikre gode logiske verdier.

DPL multiplexer

4:1 multiplexer implementert med DPL er vist i Fig. 12. Vi må nå inkludere mer logikk for å kunne anvende kontrollsignalene riktig.

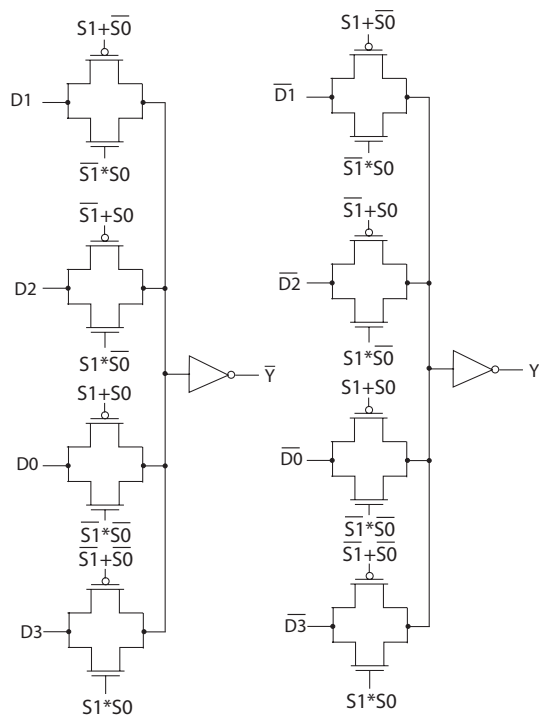


Fig. 12. 4:1 multiplexer implementert med dobbel pass transistor logikk (DPL).

DPL XOR port

En 2inngangs XOR port implementert med DPL er vist i Fig. 13.

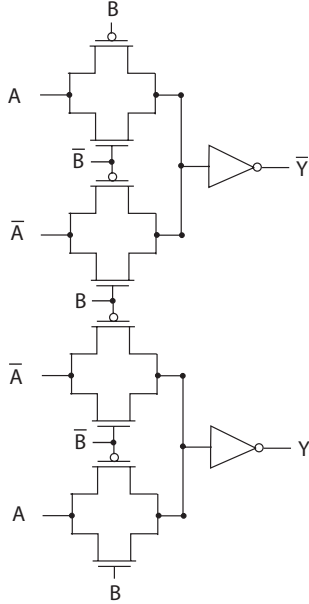


Fig. 13. XOR port med dobbel pass transistor logikk (DPL).

Teori for EEPL

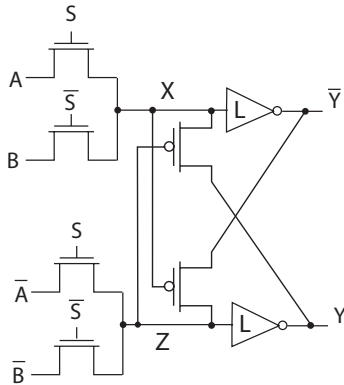


Fig. 14. Energi effektiv pass transistor logikk (EEPL). (FIG6.47)

Energi effektiv pass transistor logikk (EEPL) er vist i Fig. 14. pMOS transistorene som skal dra nodene X eller Z er ikke koblet til V_{DD} , men istedet koblet til utgangen. Dette betyr økt tidsforsinkelse, men redusert effektforbruk.

EEPL multiplekser

4:1 multiplekser implementert med energi effektiv pass transistor logikk (EEPL) er vist i Fig. 15.

EEPL XOR port

2inngangs XOR port implementert med energi effektiv pass transistor logikk (EEPL) er vist i Fig. 16.

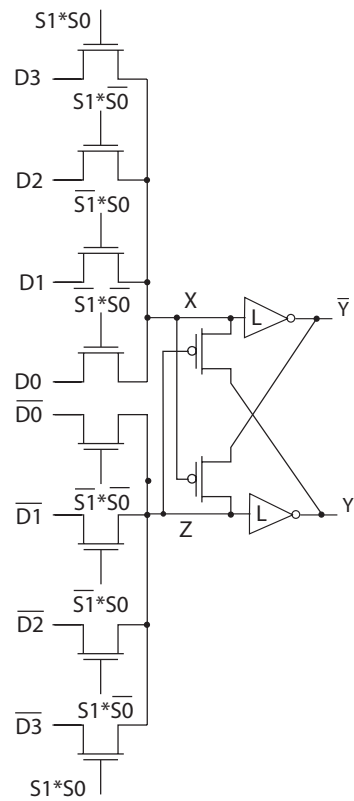


Fig. 15. 4:1 multiplekser implementert med energi effektiv pass transistor logikk (EEPL).

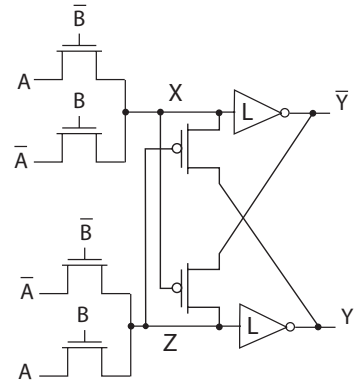


Fig. 16. XOR port implementert med energi effektiv pass transistor logikk (EEPL).

Teori for PPL

Push-pull pass transistor logikk (PPL) er vist i Fig. 17. I dette tilfellet er det to nMOS pass transistorer og to pMOS pass transistorer, der de to komplementære utgangene brukes til å trekke den andre utgangen til riktig verdi. Vi får et terskelfall over de to nMOS passtransistorene slik at Y bare kan trekkes opp til $V_{DD} - V_t$. Vi ser at når utgangen Y skal være logisk 1 så vil den andre utgangen \bar{Y} være logisk 0. I dette tilfellet vil \bar{Y} nær null skru på en pMOS transistor koblet med drain til Y slik at denne utgangen trekkes helt opp til 1. Tilsvarende vil en utgang Y nær 1 skru på nMOS transistoren med drain koblet til utgangen \bar{Y} trekke \bar{Y} helt ned til 0.

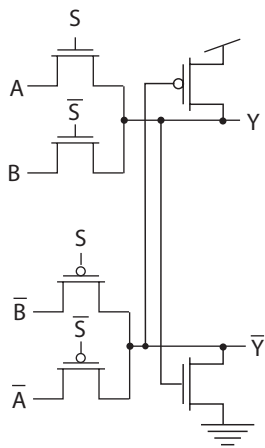


Fig. 17. Push-pull pass transistor logikk (PPL). (FIG6.47)

PPL multiplekser

4:1 multiplekser implementert med push-pull pass transistor logikk (PPL) er vist i Fig. 18.

PPL XOR port

2inngangs XOR port implementert med push-pull pass transistor logikk (PPL) er vist i Fig. 19.

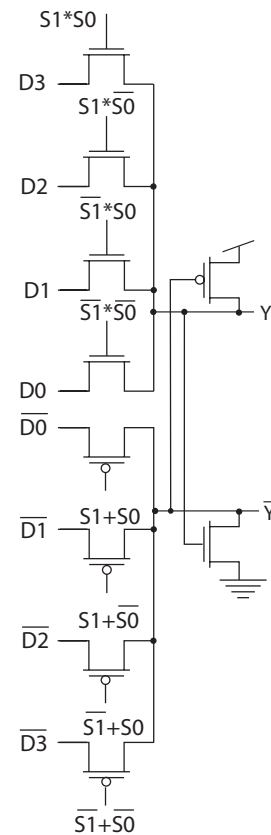


Fig. 18. 4:1 multiplekser implementert med push-pull pass transistor logikk (PPL).

Teori for SRPL

Sving-restored pass transistor logikk (SRPL) er vist i Fig. 20. Logikken ligner på EEPL, men er enklere. Her er pMOS transistorene som bidrar til å trekke nodene X og Z opp til 1 droppet. I stedet bidrar de to krysskoblede inverterne på utgangen til å trekke begge utgangene til skikkelige logiske verdier.

SRPL multiplekser

4:1 multiplekser implementert med sving-restored pass transistor logikk (SRPL) er vist i Fig. 21.

SRPL XOR port

2inngangs XOR port implementert med sving-restored pass transistor logikk (SRPL) er vist i Fig. 22.

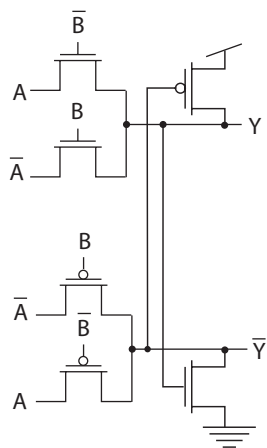


Fig. 19. XOR port implementert med push-pull pass transistor logikk (PPL).

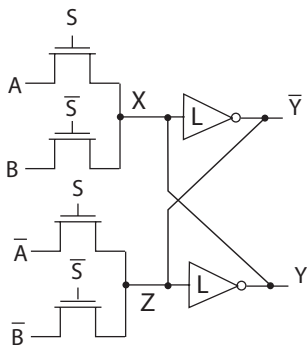


Fig. 20. Swing-restored pass transistor logikk (SRPL). (FIG6.47)

Teori for DCVSPG

Differensiell kaskode spenning svitsj med pass transistor logikk (DCVSPG) er vist i Fig. 23. Logikken er forholdsvis lik kaskode voltage svitsj logikk (CVSL). Inngangene er koblet til drain/source på transistorer slik at det blir pass transistor logikk i stedet for komplementære nedtrekk som CVSL har.

DCVSPG multiplekser

4:1 multiplekser implementert med differensiell kaskode spenning svitsj med pass transistor logikk (DCVSPG) er vist i Fig. 24.

DCVSPG XOR port

2inngangs XOR port implementert med differensiell kaskode spenning svitsj med pass transistor logikk (DCVSPG) er vist i Fig. 25.

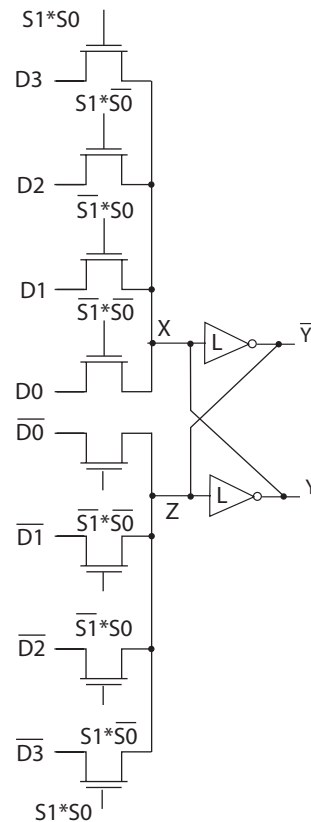


Fig. 21. 4:1 multiplekser implementert med swing-restored pass transistor logikk (SRPL).

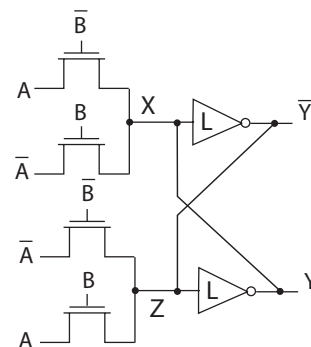


Fig. 22. XOR port implementert med swing-restored pass transistor logikk (SRPL).

Statisk CMOS multiplekser

4:1 multiplekser implementert med statisk CMOS er vist i Fig. 26.

To alternative implementasjoner i statisk CMOS er vist i Fig. 27 og 28.

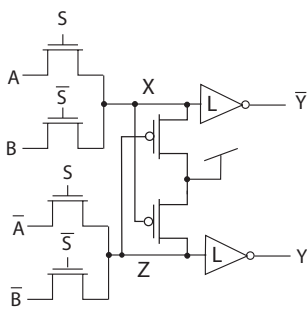


Fig. 23. Differensiell kaskode spenning svitsj med pass transistor logikk (DCVSPG). (FIG6.47)

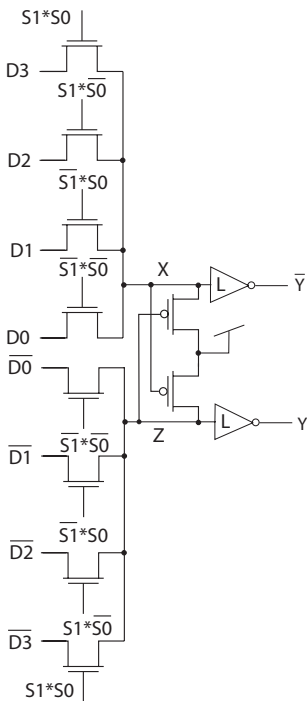


Fig. 24. 4:1 multiplekser implementert med differensiell kaskode spenning svitsj med pass transistor logikk (DCVSPG).

Statisk CMOS XOR port

Statisk CMOS XOR port er vist i Fig. 29.

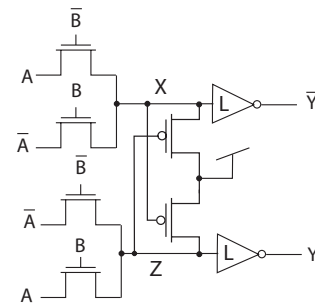


Fig. 25. XOR port implementert med differensiell kaskode spenning svitsj med pass transistor logikk (DCVSPG).

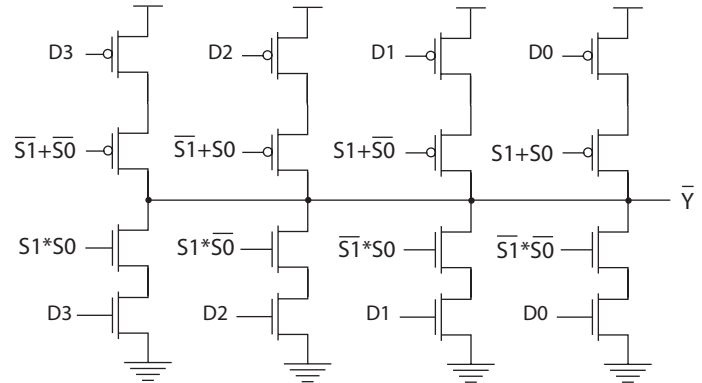


Fig. 26. 4:1 multiplekser implementert med statisk CMOS.

B.

Tegn skjematikk for en 3inngangs NAND port i følgende logikkstiler:

1. SSDL.
2. ECDL.
3. LCDL.
4. DCSL.
5. BiCMOS.

B.1 Løsningsforslag

Teori for SSDL

Sense-amplifier kretser er kretser som forsterker små differensielle signaler til store spenningsving på utgangen. Sense-amplifier kretser brukes ofte i hukommelser der bitlinjer ofte har stor kapasitiv last¹ og vil derfor ha stor forsinkelse og trege transisjoner. Kretser som reagerer på små spenningsendringer (transisjoner) vil da kunne reagere raskt når en bitlinje er i ferd med å få endret sin verdi. Sense-amplifier kretser er utsatt for ladningsdeling som kan påvirke kretsens utgang fordi kretsen reagerer på små forandringer. Dette påvirker robusthet negativt.

Sense-amplifier kretser er basert på CVSL kretser eller mer presist *dual-rail domino logikk* som er vist i Fig. 30. Vi kan betrakte sense-amplifier kretser som dual-rail domino logikk med en sense-amplifier for deteksjon og forsterkning av små differensielle spenninger.

¹Bitlinjene skal typisk drive svært mange transistorer.

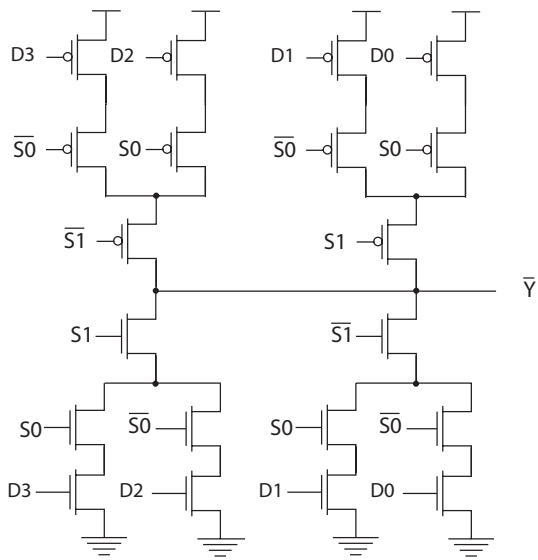


Fig. 27. 4:1 multiplexer implementert med statisk CMOS.

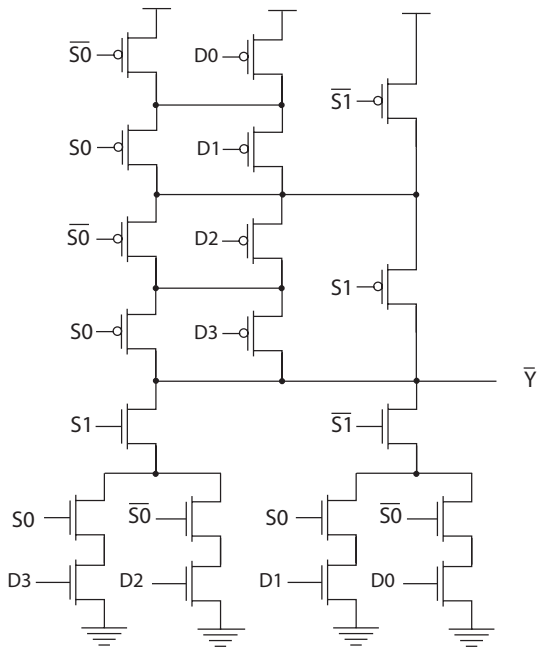


Fig. 28. 4:1 multiplexer implementert med statisk CMOS.

En generisk sense-amplifier krets er vist i Fig. 31 øverst. Den er mest nyttig for komplekse nedtrekk som representerer betydelig parasittisk forsinkelse. Inngangene er differensielle som eksempelet nederst i Fig. 31 viser. Det vil alltid være slik at bare et av nedtrekkene er PÅ.

Sample set differensiell logikk (SSDL) er vist i Fig. 32. Vi ser at SSDL skiller seg fra dual-rail domino logikk ved at SSDL ikke har en ren precharge fase. Legg merke til at nMOS transistoren som er brukt for å koble de komplementære nedtrekkene til GND er styrt av $\bar{\phi}$ og ikke ϕ som for dual-rail domino logikk. Dette betyr at porten ikke precharger, men *sample*. Vi må forutsette at nedtrekkene ikke kan overstyre pMOS transistorene som skal trekke både X og \bar{X} opp mot V_{DD} eller logisk 1.

Sample set differensiell logikk (SSDL) ved sampling er vist i Fig. 33. Vi antar at inngangene er stabile slik at ett av

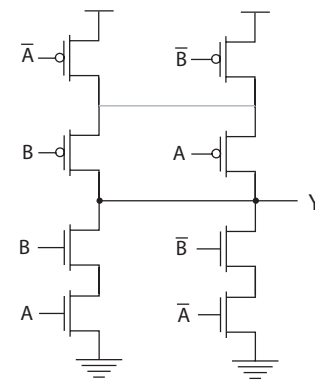


Fig. 29. XOR port implementert med statisk CMOS.

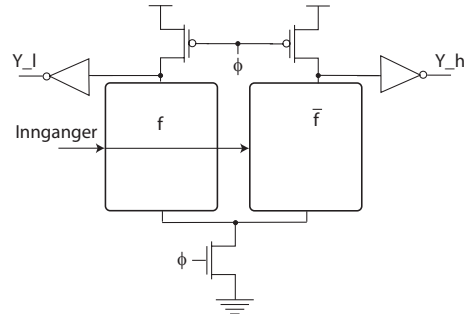


Fig. 30. Dual-rail domino logikk. (FIG6.63)

nedtrekkene er PÅ (f i dette eksemplet). Nedtrekket som er PÅ vil forsøke å trekke noden X ned til GND eller logisk 0. Vi kan anta at transistorene er dimensjonert slik at noden X vil få en spenning som er litt lavere enn V_{DD} , illustrert som stiplet linjer i figuren. Sense-amplifier kretsen som har en felles nMOS transistor i nedtrekket ned mot GND som er styrt av ϕ vil skru av nedtrekkene i sense-amplifieren. Poenget med sense-amplifere er å lage et alternativt nedtrekk for nodene X og \bar{X} som er mye mer effektivt enn de komplekse nedtrekkene (f og \bar{f}). Vi ser at porten sampler inngangene slik at en av nodene X og \bar{X} blir trukket noe ned fra V_{DD} istedet for å precharges til V_{DD} . I denne samplingsfasen er det statisk effektforbruk fordi en av utgangene X eller \bar{X} vil ha opptrekk og nedtrekk som er på samtidig.

Når ϕ skifter fra 0 til 1 vil porten gå over i en *set* fase (motsvarende evaluering for dual-rail domino logikk) som vist i Fig. 34. nMOS transistoren i det effektive nedtrekket i sense-amplifieren vil trekke \bar{X} raskt ned til GND fordi X og ϕ er begge 1. Poenget er at kretsen slipper å trekke en av utgangene X eller \bar{X} ned til 0 via de komplekse nedtrekkene. De grå transistorene antas å være svake og brukes til å motvirke reduksjon i den av utgangene X og \bar{X} som fortsatt skal være logisk 1.

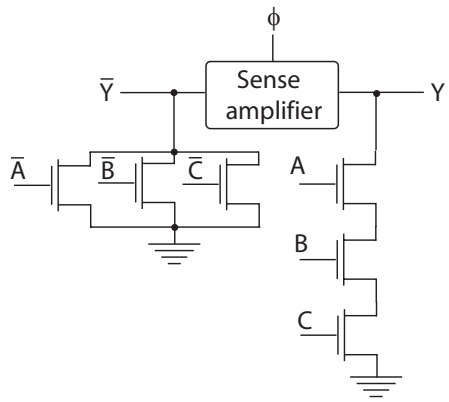
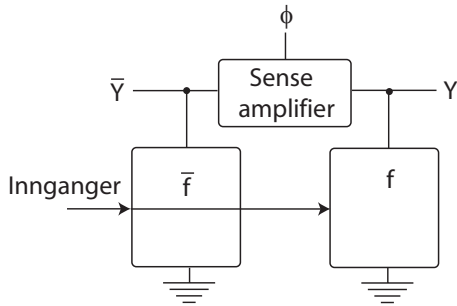


Fig. 31. Generisk sense-amplifier krets. (FIG6.64)

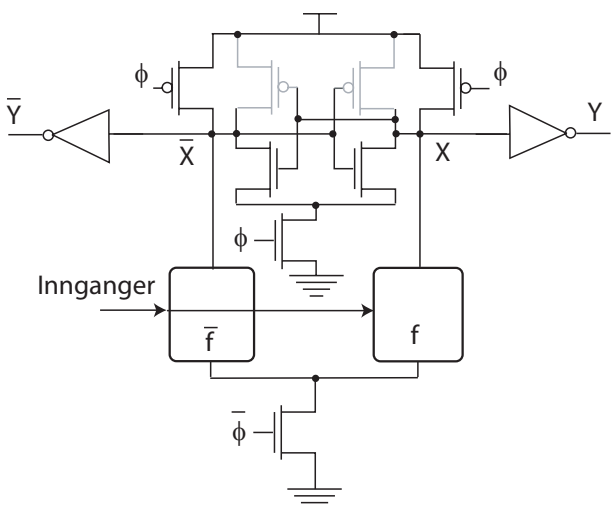


Fig. 32. Sample set differensiell logikk (SSDL). (FIG6.63)

3inngangs NAND SSDL port

3inngangs NAND port implementert i sample set differensiell logikk (SSDL) er vist i Fig. 35.

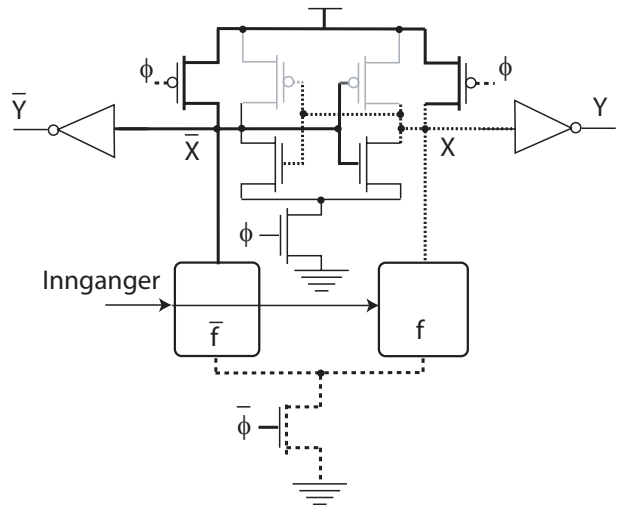


Fig. 33. Sample set differensiell logikk (SSDL) ved sample fase. (FIG6.63)

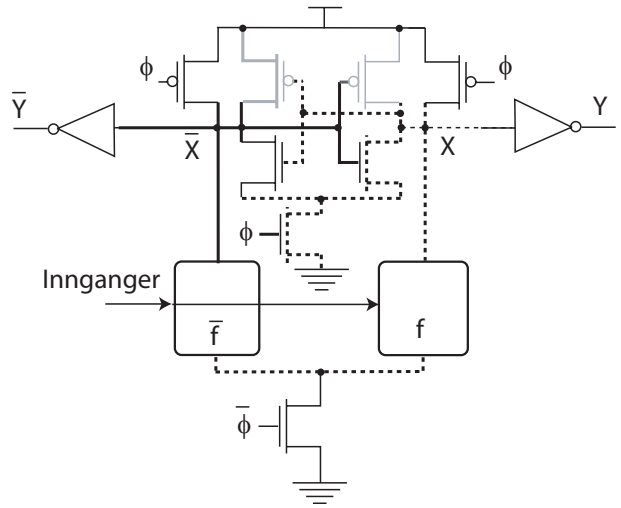


Fig. 34. Sample set differensiell logikk (SSDL) ved set fase. (FIG6.63)

Teori for ECDL

Enable/disable CMOS diffrenesiell logikk (ECDL) som er vist i Fig. 36 representerer en forbedring av SSDL logikk ved at statisk effektforbruk blir redusert.

ECDL logikk ved disable er vist i Fig. 37. Porten disables når phi = 1, slik at utgangene blir trukket ned til 0 som vist i Fig. 37. Her er "precharge" transistorene nMOS transistorer slik at de ikke vil danne en strømvei mellom spenningsreferansene VDD og GND og dermed bidra med statisk effekt. pMOS transistoren i toppen sørger for at det ikke er opptrekk som er PÅ. Legg merke til at de komplementære nedtrekkene kun eventuelt bidrar til å hjelpe med nedtrekket når porten skal disables.

ECDL logikk ved disable er vist i Fig. 38. Porten disables når phi = 0, slik at pMOS transistoren i toppen som er styrt av phi skrur av og "precharge" transistorene skrur av som vist i Fig. 38. De to pMOS transistorene i det som logisk er to krysskoblete invertere er i starten PÅ og vil forsøke å trekke begge utgangene til logisk 1. I figuren har vi antatt at nedtrekket f er PÅ og det vil derfor bidra til å holde utgangen Y lav. De to krysskoblete inverterne vil holde utgangene til riktig logiske verdier.

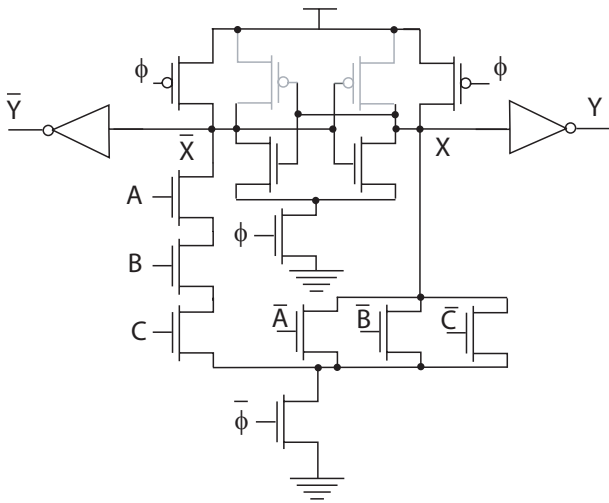


Fig. 35. 3inngangs NAND port implementert i sample set differensiell logikk (SSDL).

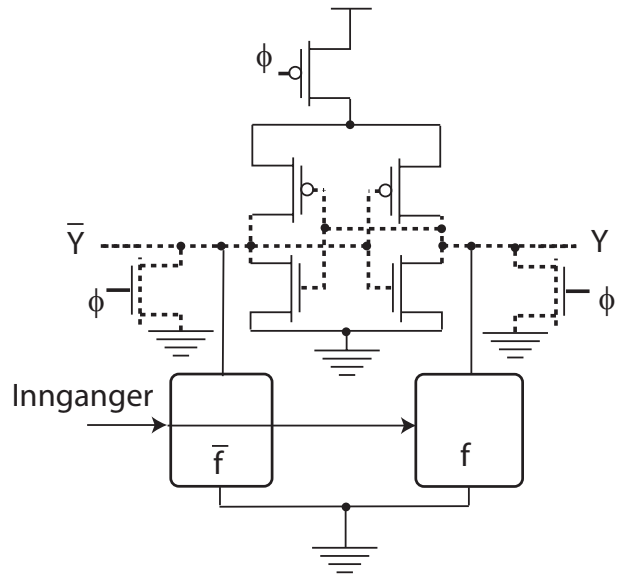


Fig. 37. Enable/disable CMOS differensiell logikk (ECDL) ved disable. (FIG6.63)

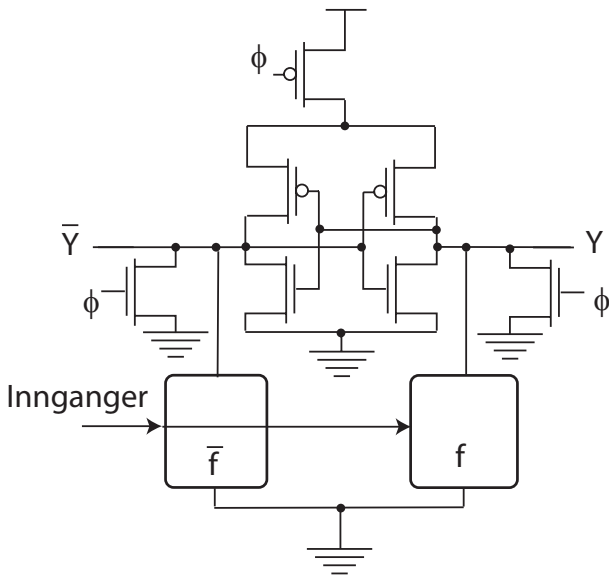


Fig. 36. Enable/disable CMOS differensiell logikk (ECDL). (FIG6.63)

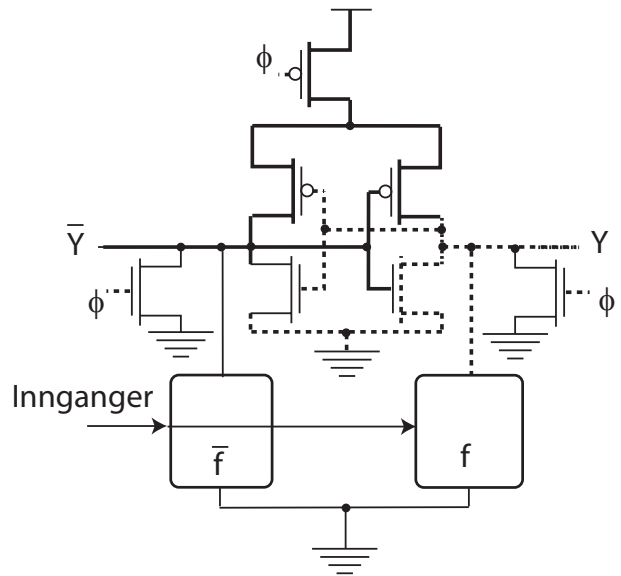


Fig. 38. Enable/disable CMOS differensiell logikk (ECDL) ved enable. (FIG6.63)

3inngangs NAND ECDL port

3inngangs NAND port implementert med nable/disable CMOS differensiell logikk (ECDL) er vist i Fig. 39.

Teori for LCDL

Latched CMOS differensiell logikk (LCDL), som er vist i Fig. 40, ligner på SSDL. nMOS transistoren under nedtrekkene er styrt av ϕ og ikke $\bar{\phi}$. På utgangene er det en latch som er likt utgangstrinnet på en ekte-en fase (TSPC) latch.

LCDL i precharge fasen er vist i Fig. 41 der $\phi = 0$. Nodene X og \bar{X} precharges til logisk 1. Begge nedtrekkene, f og \bar{f} , er skrudd av ved hjelp av ϕ som styrer en nMOS transistor. Legg merke til at utgangene ikke påvirkes av eventuelle endringer på X og \bar{X} ved precharge fordi utgangstrinnet er likt et utgangstrinn for en TSPC latch.

LCD i evalueringsfasen er vist i Fig. 41 der $\phi = 1$. Et av nedtrekkene vil nå trekke en av nodene X eller \bar{X} ned mot 0 og dermed trigge sense-amplifier kretsen som raskt bidrar med å trekke den aktuelle noden raskt helt ned til 0. Utgangstrinnene

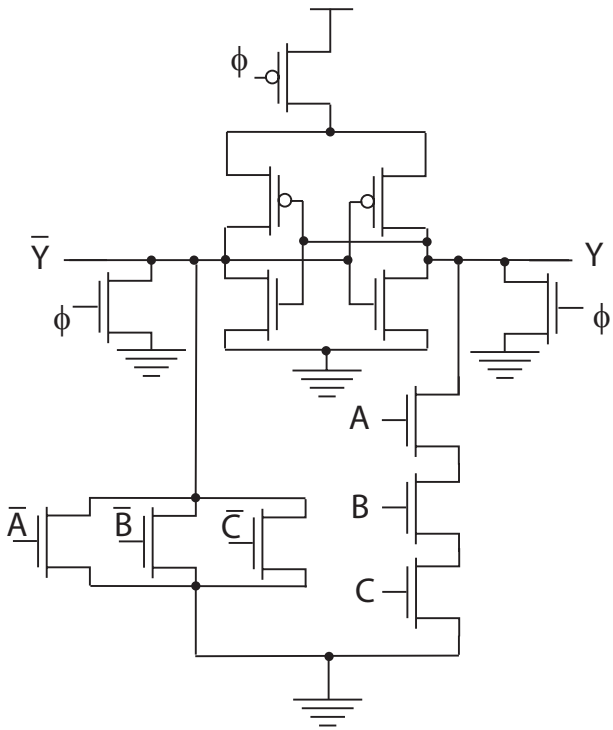


Fig. 39. 3inngangs NAND port implementert med enable/disable CMOS differensiell logikk (ECDL).

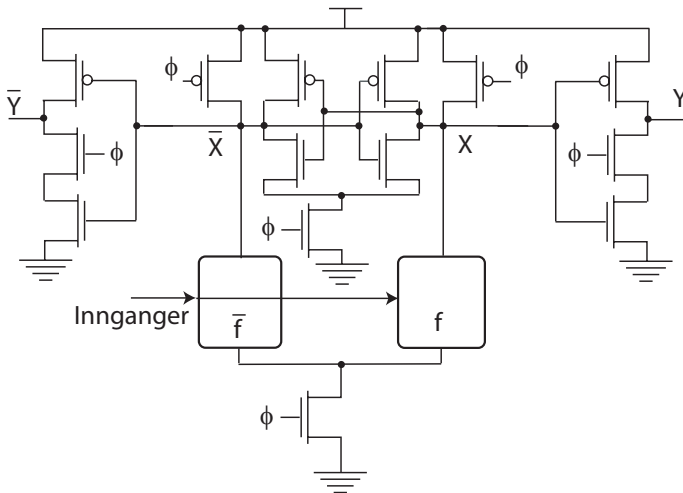


Fig. 40. Latched CMOS differensiell logikk (LCDL). (FIG6.63)

virker nå bare logisk som inverterte.

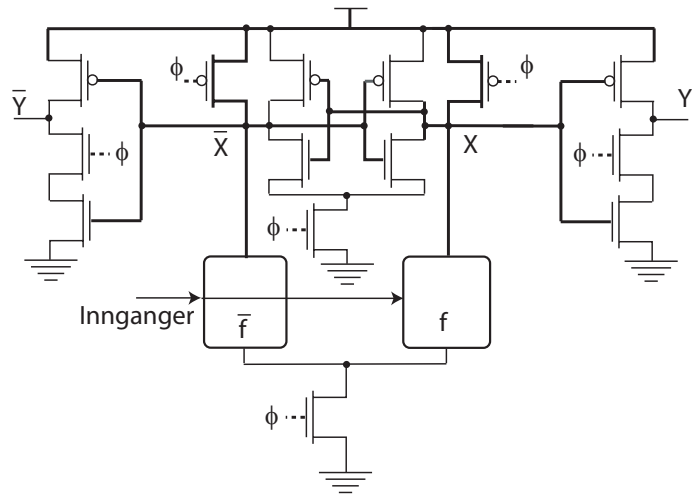


Fig. 41. Latched CMOS differensiell logikk (LCDL) ved precharge. (FIG6.63)

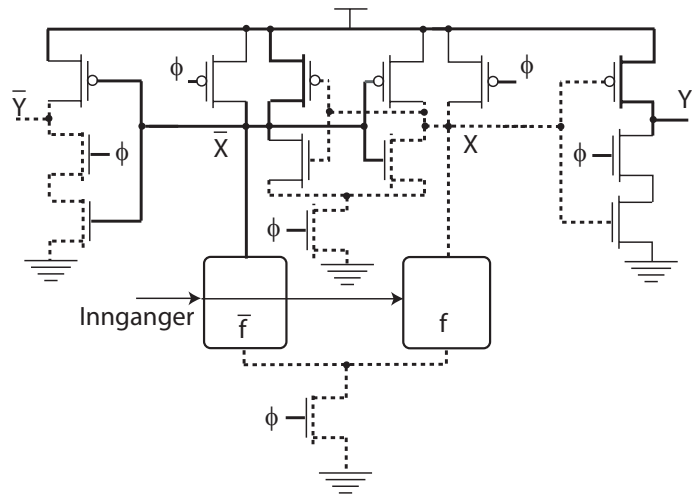


Fig. 42. Latched CMOS differensiell logikk (LCDL) ved evalering. (FIG6.63)

3inngangs NAND LCDL port

3inngangs NAND port implementert med catched CMOS differensiell logikk (LCDL) er vist i Fig. 43.

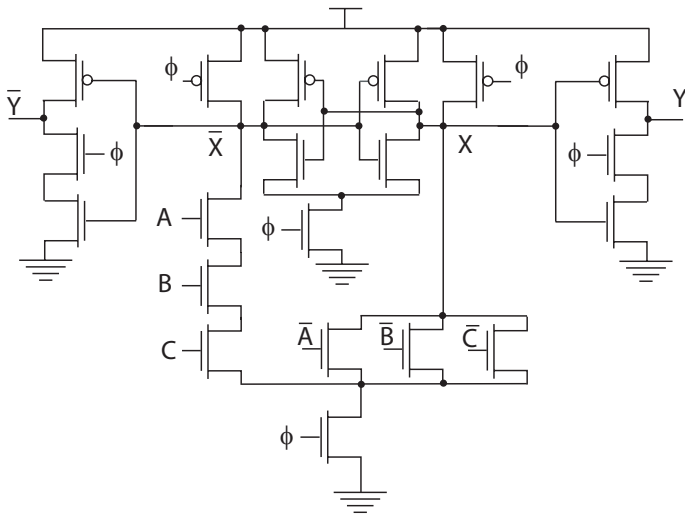


Fig. 43. 3inngangs NAND port implementert med latched CMOS differensiell logikk (LCDL).

Teori for DCSL

Differensielle kretser kan bruke mye effekt fordi en av utgangene vil ha en transisjon i hver klokkesykel.

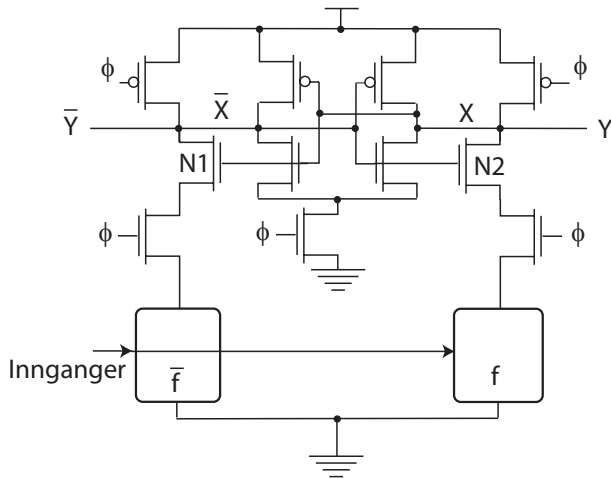


Fig. 44. Differensiell strøm svitsj logikk (DCSL). (FIG6.63)

Med differensiell strøm svitsj logikk (DCSL), som vist i Fig. 44 forsøker man å redusere effektforbruket relatert til interne noder og øke hastigheten ved å redusere spenningen som nedtrekket skal lade ut.

DCSL ved precharge er vist i Fig. 45, der vi antar at $\phi = 0$. Legg merke til at nMOS transistorene N1 og N2 er PÅ mens nMOS transistorene mellom disse transistorene og de komplekse² nedtrekkene er AV. Interne noder i nedtrekkene vil ikke lades opp under precharge.

DCSL ved evaluering er vist i Fig. 46, der vi antar at $\phi = 1$. Ved evaluering vil en av transistoren N1 eller N2 alltid stenge og sørge for at den av utgangene som ikke skal trekkes til 0 ikke vil ha strømveier ned i nedtrekket (\bar{f}). Det nedtrekket som er PÅ vil sørge for å dra en av utgangene ned til 0 (Y) og dermed interne (de fleste) nodene i dette nedtrekket også ned til 0.

²Antar at nedtrekkene har mange interne noder.

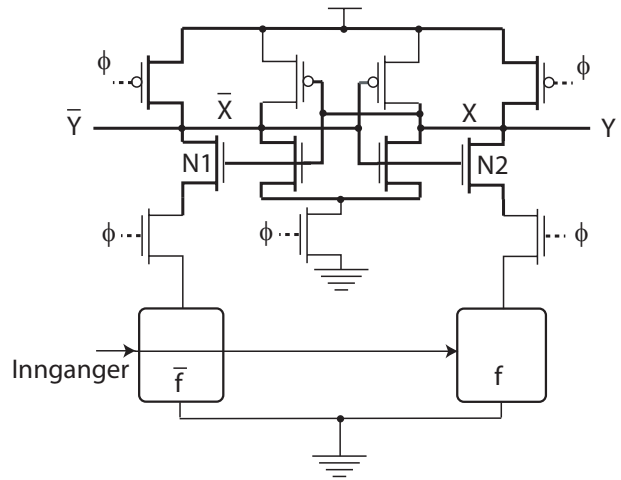


Fig. 45. Differensiell strøm svitsj logikk (DCSL) ved precharge. (FIG6.63)

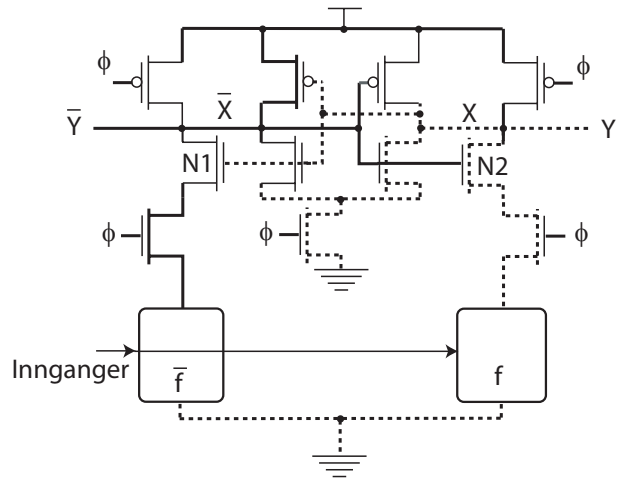


Fig. 46. Differensiell strøm svitsj logikk (DCSL) ved evaluering. (FIG6.63)

DCSL2 som er vist i Fig. 47 precharger utgangene til 0 tilsvarende ECDL. Transistorene N1 og N2 virker på tilsvarende måte som for DCSL1.

I DCSL3, som er vist i Fig. 48, er de to nMOS precharge transistorene ertattet med en pass transistor som kortslutter de to utgangene slik at begge utgangene blir precharged til 0 fordi pMOS transistorene i de to krysskoblete inverterne ikke kan levere strøm på grunn av pMOS transistorene styrt av ϕ som er koblet til logisk spenningsreferansen V_{DD} .

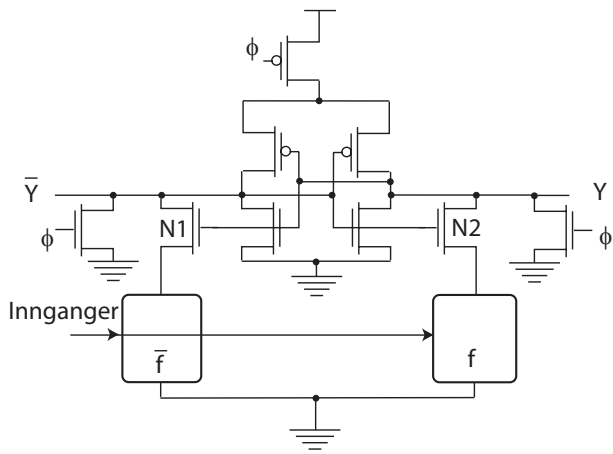


Fig. 47. Differensiell strøm svitsj logikk (DCSL2). (FIG6.63)

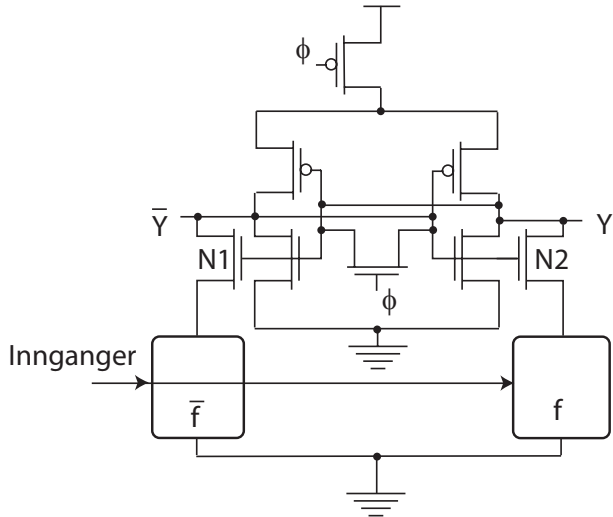


Fig. 48. Differensiell strøm svitsj logikk (DCSL3). (FIG6.63)

DCSL NAND port

3inngangs NAND port implementert med differensiell strøm svitsj logikk (DCSL) er vist i Fig. 49.

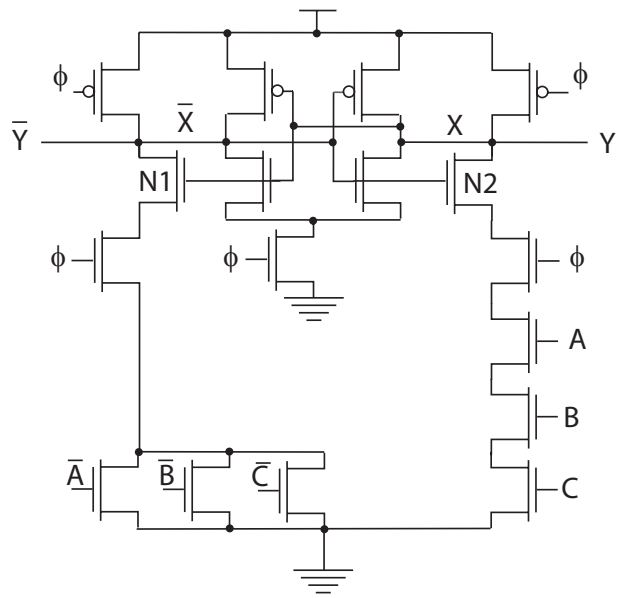


Fig. 49. 3inngangs NAND port implementert med differensiell strøm svitsj logikk (DCSL).

Teori for BiCMOS

Bipolare transistorer kan levere mye mer strøm enn CMOS transistorer og brukes i noen kretser der det er spesielt stort behov for sterke utgangsdrivere. Noen CMOS prosesser tillater implementasjon av bipolare transistorer. Slike prosesser kalles BiCMOS.

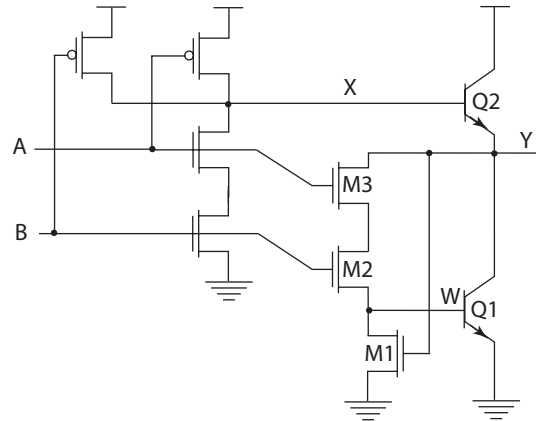


Fig. 50. BiCMOS NAND port. (FIG6.67)

En BiCMOS NAND port er vist i Fig. 50. Transistorene M2 og M3 tilsvarer nedtrekket for en CMOS NAND port. Transistoren M1 brukes for å trekke basen W på npn transistoren Q1 til 0 slik at nedtrekket for den bipolare utgangen blir mest mulig effektivt.

3-inngangs NAND BiCMOS port

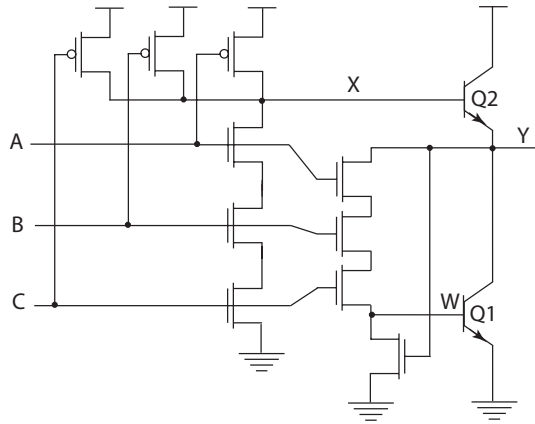


Fig. 51. BiCMOS 3 inngang NAND port.

BiCMOS 3 inngang NAND port er vist i Fig. 51.

C. Eksamensoppgave 2005

Hvorfor trengs buffere (repeaters) for å drive signaler over en viss avstand? Hvilke metallag er det vanlig å bruke til å distribuere spenningsreferanser og klokkesignaler?

C.1 Løsningsforslag

Motstanden og kapasitansen i en leder, typisk metall, er proporsjonal med lengden på lederen. Dette vil gi en stor tidsforsinkelse for signaler som skal transporteres langt. Det er vanlig å bruke de øverste metallag for distribusjon av forsyningspenninger og klokkesignaler, pga liten egenmotstand.

D. Eksamensoppgave 2005

Anta at en metalleder med lengde $l = 1800\mu m$ har egenmotstand $0.15\Omega/\mu m$ og egenkapasitans $0.2fF/\mu m$. Anta at lederen skal drives av en enhetsinverter, med $R = 3k\Omega \cdot \mu m$ og parasittisk kapasitans $C_p = 4.5fF/\mu m$, og buffres med invertere (repeaters). Hvor mange invertere trenger vi for å buffre? Finn størrelse på nMOS- og pMOS transistorene i buffrene.

D.1 Løsningsforslag

Vi har

$$\begin{aligned} \frac{l}{N} &= \sqrt{\frac{2RC_p}{R_w C_w}} \\ &= \sqrt{\frac{2 \cdot 3k\Omega \cdot \mu m \cdot 4.5 \frac{fF}{\mu m}}{0.15 \frac{\Omega}{\mu m} \cdot 0.2 \frac{fF}{\mu m}}} \\ &= 948.7 \\ &\approx 950\mu m \end{aligned}$$

Dette gir $N=2$, dvs. to buffere.

Størrelsen for nMOS transistoren blir

$$\begin{aligned} W &= \sqrt{\frac{RC_w}{R_w C_p}} \\ &= \sqrt{\frac{3k\Omega \cdot \mu m \cdot 0.2 \frac{fF}{\mu m}}{0.2 \frac{\Omega}{\mu m} \cdot 4.5 \frac{fF}{\mu m}}} \\ &= 25.8 \\ &\approx 26\mu m \end{aligned}$$

E. Forsinkelse i interkonnekt

Gitt en 3mm lang og $0.4\mu m$ bred leder i metall 2 i en 180nm prosess med egenmotstand $0.04\Omega/\square$ og kapasitans $0.25fF/cm$. Bruk π modell med tre segmenter (avdelinger) og lage en modell for lederen.

E.1 Løsningsforslag

Teori

Det er to grunner til at interkonnekt bidrar til å øke tidsforsinkelse i en krets:

1. Ruting av signaler (i metall) vil legge last til utgangen på en port.
2. Lange ledere har signifikant motstand.

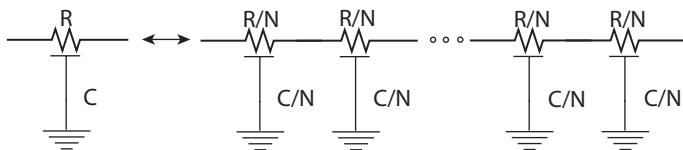


Fig. 52. Oppdeling av en leder i N deler. (FIG4.38)

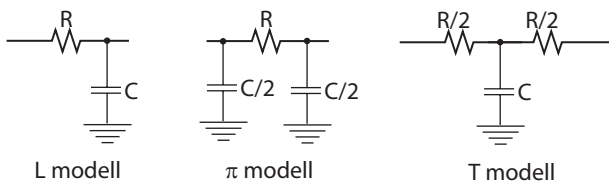


Fig. 53. Ulike modeller for forsinkelse i interkonnekt. (FIG4.38)

Det er enkelt å utvide Elmore forsinkelsesmodell med forsinkelse i interkonnekt. Motstand og kapasitans i en leder kan approksimeres ved å dele opp lederen i små avdelinger som vist i Fig. 52. Det er tre standard metoder for approksimasjon som benyttes; *L modell*, *π modell* og *T modell* som vist i Fig. 53. *L* modellen krever et høyt antall avdelinger for å produsere et nøyaktig resultat og anvendes derfor ikke så ofte. *π* modellen gir god nøyaktighet (3% avvik) for 3 eller flere avdelinger. *L* modellen kan sammenlignes med *π* modellen men vil være mer krevende å benytte fordi antallet elektriske noder er større. Vi ser at både kapasitans og motstand i en metalleder vil øke med lengde som medfører at forsinkelse i lederen øker kvadratisk.

Det er vanligst å bruke metallag til å rute signaler (interkonnekt) på grunn av liten egenmotstand.

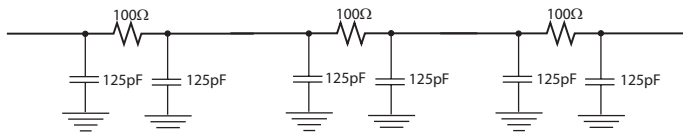


Fig. 54. *π* modell av leder.

3-segment *π* modell for leder er vist i Fig. 54. Lederen er $3000\mu\text{m}/0.4\mu\text{m}$ som utgjør 7500 arealenheter. Total motstand er $(0.04\Omega/\square \cdot 7500\square = 300\Omega)$. Total kapasitans er $(0.25\text{fF}/\mu\text{m}) \cdot (3000\mu\text{m}) = 750\text{fF}$. Hvert *π*-segment har en tredjedel av den totale motstanden og kapasitansen.

F. Crosstalk

To ledere med lengde 5mm har kapasitans $0.08\text{fF}/\mu\text{m}$ til jord og $0.1\text{fF}/\mu\text{m}$ til nabolederen. Hver leder blir drevet av en inverter med effektiv motstand lik $2\text{k}\Omega$.

F.1 Løsningsforslag

Teori

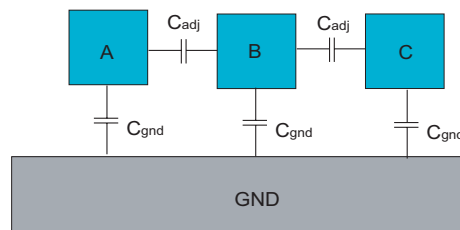


Fig. 55. Kapasitans mellom naboledere i samme lag og til GND. (FIG4.41)

I Fig. 55 er det vist kapasitans mellom naboledere i samme lag og til GND. Når A svitsjer³ vil dette påvirke nabolederen B som også vil få en spenningsendring i samme retning. Den kapasitive påvirkningen kalles *crosstalk*. *Crosstalk* kan påvirke naboen lederen slik at nabolederen får økt eller redusert sin egens svitsjetid. Påvirkningsgraden er avhengig av kapasitand mellom lederne og den totale kapasitans knyttet til lederen som påvirkes av *crosstalk*.

G. Forsinkelse

B	ΔV	$C_{eff}(A)$	MCF
Konstant	V_{DD}	$C_{gnd} + C_{adj}$	1
Svitsjing i samme retning	0	C_{gnd}	0
Svitsjing i motsatt retning	$2V_{DD}$	$C_{gnd} + 2C_{adj}$	2

TABLE I

Crosstalk avhengighet av svitsjeretninger.

Dersom en leder og nabolederen svitsjer i samme retning vil lederne påvirke hverandre positivt, dvs. redusert, med hensyn på forsinkelse. I tabell I er det vist hvordan *crosstalk* påvirkes av svitsjeretninger. Ladning som overføres til en koblingskondensator er gitt av

$$Q = C_{adj}\Delta V, \quad (1)$$

der ΔV er spenningsendringen mellom de elektriske nodene (ledere). Dersom for eksempel A svitsjer og B ligger fast blir $\Delta V = V_{DD}$. Dersom nodene A og B svitsjer i motsatt retning blir $\Delta V = 2V_{DD}$. Dette kalles *Miller effekt*. *Miller koblingsfaktor* (*MCF*) modellerer kapasitansen mellom to elektriske nodere (ledere). En vanlig verdi for MCF er 1.5.

En konservativ modell for MCF er 2 ved beregning av propageringsforsinkelse og 0 ved beregning av contamination forsinkelse.

Vi kan beregne contamination- og propageringsforsinkelse ved å finne de relevante kapasitansene; $C_{gnd} = (0.08\text{fF}/\mu\text{m}) \cdot (5000\mu\text{m}) = 0.4\text{pF}$ og $C_{adj} = (0.1\text{fF}/\mu\text{m}) \cdot (5000\mu\text{m}) = 0.5\text{pF}$.

³Transisjon fra 0 til 1 eller fra 1 til 0.

Tidsforsinkelsen er gitt av RC_{eff} . Contamination forsinkelse kan beregnes ved at vi antar at nodene svitsjer i samme retning slik at $C_{eff} = C_{gnd}$ og dermed $t_{cd} = (2k\Omega) \cdot (0.4pF) = 800ps$. Ved beregning av propageringsforsinkelse antar vi at lederne svitsjer i motsatt retning slik at $C_{eff} = C_{gnd} + 2C_{adj} = 1.4pF$ som gir $t_{pd} = (2k\Omega) \cdot (1.4pF) = 2.8ns$.

H. Hva er crosstalk støy?

H.1 Løsningsforslag

Teori

Når to ledere ligger forholdsvis nær hverandre vil de kunne påvirke hverandre elektrisk gjennom parasittiske (crosstalk) kapasitanser. En slik påvirkning er derfor kapasitiv. Anta et en leder B skal ligge på en fast spenningen og at en leder A svitsjer. Dersom A påvirker spenningen på B gjennom crosstalk kaller vi dette for *crosstalk støy*. I dette tilfellet kaller vi A for *aggressor* og B for *victim*.

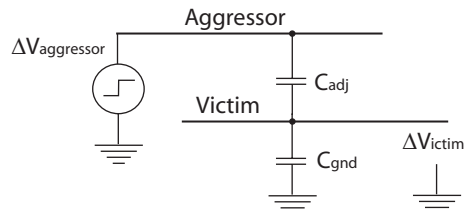


Fig. 56. Aggressor og victim. (FIG4.42)

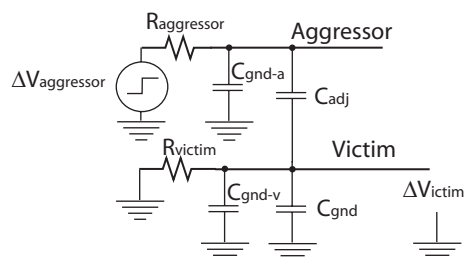


Fig. 57. Aggressor og victim med drivere. (FIG4.43)

I Fig. 56 ser vi to ledere med kapasitansen C_{adj} mellom lederne. Den ene lederen (victim) påvirkes av spenningsendring på den andre lederen (aggressor):

$$\Delta V_{victim} = \frac{C_{adj}}{C_{gnd} + C_{adj}} \Delta V_{aggressor}, \quad (2)$$

der $\Delta V_{aggressor}$ er spenningsendring på aggressor lederen. Dersom victim lederen drives vil strømmen som driveren leverer redusere crosstalk støy for victim. Dette kan modelleres som

$$\Delta V_{victim} = \left(\frac{C_{adj}}{C_{gnd} + C_{adj}} \right) \left(\frac{1}{1+k} \right) \Delta V_{aggressor}, \quad (3)$$

der

$$\begin{aligned} k &= \frac{\tau_{aggressor}}{\tau_{victim}} \\ &= \frac{R_{aggressor} (C_{gnd-a} + C_{adj})}{R_{aggressor} (C_{gnd-v} + C_{adj})}, \end{aligned} \quad (4)$$

der C_{gnd-a} og C_{gnd-v} er henholdsvis kapasitans for aggressor og victim til jord som vist i Fig. 57. Crosstalk støy er mest dominerende når victim er udrevet eller svakt drevet i forhold til aggressor, dette medfører at $k < 1$.

Effekten av crosstalk er vist i Fig. 58.

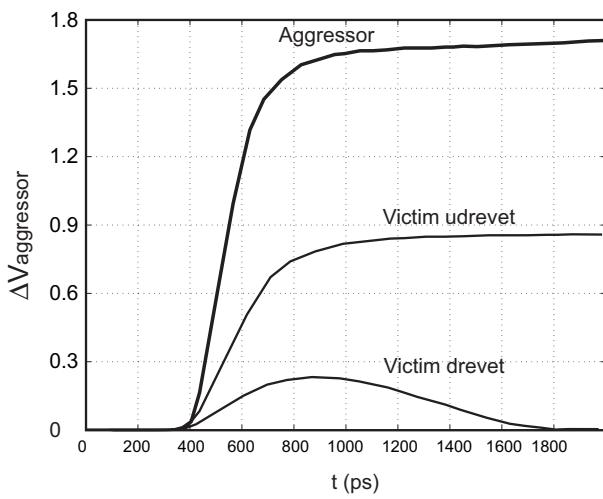


Fig. 58. Crosstalk. (FIG4.44)

I. Induktans

Anta et signal i metall 2 med en egenmotstand $0.04\Omega/\square$ og en bredde på $0.4\mu m$. Kapasitansen er $0.25fF/\mu m$ og induktansen er $0.5pH/\mu m$. Finn hastigheten til signalet og finn lengden der induktansen er signifikant (som funksjon av stigetid).

I.1 Løsningsforslag

Teori

Vanligvis sier vi at strømmer endrer spenninger i dynamiske noder ved å lade opp eller ut kapasitans knyttet til utganger på porter. I tillegg går det noen ganger strømmer mellom spenningsreferansene V_{DD} og V_{SS} (jord). I virkeligheten vil strømmer følge løkker i en integrert krets, typisk via spenningsreferansene. Strømmer som følger en løkke vil generere et magnetisk felt som er proporsjonalt med løkkens areal og strømstyrken. Endring av strømmen krever energi til å forandre det magnetiske feltet. Dette betyr at strømendringer inducerer en spenningsendring som er proporsjonal med endringsraten. Proporsjonalitetskonstanten kalles *induktans* L . Vi kan uttrykke den induserte spenningen som:

$$V = L \frac{dI}{dt}. \quad (5)$$

Induktans og kapasitans bestemmer lyshastigheten i et materiale. Selv om motstanden er 0, som vil gi $RC = 0$ dvs. ingen RC forsinkelse, kan vi uttrykke lysets tidsforsinkelse i en leder med lengde l og med kapasitans per enhetslengde C og induktans per enhetslengde L som

$$t_{pd} = l\sqrt{LC}. \quad (6)$$

Vi kan uttrykke signalthastighet v som

$$\begin{aligned} v &= \frac{1}{\sqrt{LC}} \\ &= \frac{1}{\sqrt{\epsilon_{ox}\mu_0}} \\ &= \frac{c}{\sqrt{k}}, \end{aligned} \quad (7)$$

der μ_0 er magnetisk permittivitet i vakum ($4\pi \cdot 10^{-7} H/m$) og c er lyshastigheten i vakum ($3 \cdot 10^8 m/s$). Det vil si at signaler oppnår ca. halve lyshastigheten. Dersom man velger et dielektrisk materiale med lav dielektrisk konstant eller relativ permittivitet $k < 3.9$ vil hastigheten øke.

Endringer i magnetfelt kan forårsake at strømveier endres og dette vil kunne redusere hastighet som følge av *induktiv crosstalk*.

Induktansen til en leder med lengde l og bredde w lokalisert i høyde h over et jordplan kan (forenklet) uttrykkes som:

$$L = l \frac{\mu_0}{2\pi} \ln\left(\frac{8h}{w} + \frac{w}{4h}\right), \quad (8)$$

dersom vi antar at $w < h$ og tykkelsen på lederen er neglisjerbar. Typiske induktanser på en integrert krets er i området $0.15 - 1.5pH/\mu m$ avhengig av nærhet av forsyningsplan (V_{DD} eller jord gnd).

Strømmer vil følge veier med lavest impedans $Z = R + j\omega L$. Ved høye frekvenser ω vil impedansen domineres av induktans. Induktansen reduseres når strømmen går nær overflaten av den nærmeste returveien for strømmen. Denne effekten kalles *skinn effekt* ("skin effect") og kan i praksis redusere det effektive tverrsnittet av den tykkeste lederen og dermed øke den effektive motstanden i lederen ved høye frekvenser. *Skinn dybde* kan uttrykkes som:

$$\delta = \sqrt{\frac{2\rho}{\omega\mu_0}}, \quad (9)$$

der ρ er egenmotstanden til lederen, og ω er frekvensen (klokkefrekvensen) i systemet.

Skinndybde blir et større problem i mer moderne teknologier (prosesser) pga. av høye frekvenser. Da vil nytten av å bruke tykke ledere bli redusert.

J. Betydning av induktans i integrerte kretser

Den viktige frekvensen er den høyeste frekvensen med signifikant effekt i signalets Fourier transform. Dette er en frekvens som er internt i en integrert krets og assosiert med prosessparameterverdier som gir høyest hastighet:

$$\omega = \frac{2\pi}{6t_{rf}}. \quad (10)$$

Ekstrahering av induktans er generelt et tredimensjonalt problem og ekstremt tidkrevende for komplekse geometrier. Inkludering av induktans i simuleringer er vanskelig og derfor er det vanlig å holde seg til designregler som gjør at man kan neglisjere induktans.

Induktans har alltid vært viktig for innkapsling av integrerte kretser fordi de fysiske størrelsene blir store i forhold til internt i de integrerte kretsene. Induktans internt i integrerte kretser er viktig for ledere der lyshastigheten er større enn RC forsinkelse for lederen. Lyshastigheten øker lineært med lederens lengde mens RC forsinkelse øker kvadratisk. Vi kan derfor estimere lengden på ledere der induktans ikke har betydning.

$$\frac{t_r}{2\sqrt{LC}} < l < \frac{2}{R} \sqrt{\frac{L}{C}}. \quad (11)$$

Hastigheten er gitt av

$$\begin{aligned}
v &= \frac{1}{\sqrt{LC}} \\
&= \frac{1}{\sqrt{(0.5pH/\mu m)(0.25fF/\mu m)}} \\
&= 0.8 \cdot 10^8 m/s \\
&\approx \frac{1}{4}c.
\end{aligned}$$

$0.8 \cdot 10^8 m/s$ tilsvarer $0.8 \cdot 10^{-4} m/ps = 80 \mu m/ps = 0.08 \mu m/fs$.
For at induktans skal ha betydning må lengden l på ledere tilfredstille

$$\begin{aligned}
l &< \frac{2}{R} \sqrt{\frac{L}{C}} \\
&< \frac{2}{\frac{0.04}{0.4} \Omega/\mu m} \sqrt{\frac{0.5pH/\mu m}{0.25fF/\mu m}} \\
&< 895 \mu m.
\end{aligned}$$

I tillegg har vi at

$$\begin{aligned}
l &> \frac{t_r}{2\sqrt{LC}} \\
&> \frac{v}{2} t_r \\
&> 0.4 \cdot 10^8 (m/s) \cdot t_r \\
&> 0.4 \cdot 10^2 (\mu m/s) \cdot t_r \\
&> 40 (\mu m/s) \cdot t_r,
\end{aligned}$$

der l er uttrykt i μm og t_r i ps . Vi ser at dersom $t_r > 25ps$ vil signalforsinkelsen være dominert av RC effekter (og stigetid). Induktans har bare betydning når stigetiden er svært liten.

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,