

Del 12: Passtransistor- og differensiell CMOS logikk

YNGVAR BERG

I. INNHOLD

CMOS med transmisjonsporter blir presentert, herunder komplementær pass transistor logikk (CPL), lean integration med pass transistorer (LEAP) og andre pass transistor logikk typer. Ulike differensielle logikkstiler blir introdusert inkludert sample set differensiell logikk (SSDL), enable/disable CMOS differensiell logikk (ECDL), latched CMOS differensiell logikk (LCDL) og differensiell strøm svitsj logikk (DCSL). BiCMOS logikk blir overfladisk presentert. Alle henvisninger til figurer er relevant for Weste & Harris [1].

1. *Innhold.*
2. *CMOS med transmisjonsporter.* Kapittel 6.2.5.1 side 347 - 348.
3. *Komplementær pass transistor logikk (CPL).* Kapittel 6.2.5.2 side 348 - 349.
4. *Lean integration med pass transistorer (LEAP).* Kapittel 6.2.5.3 side 349 - 350.
5. *Andre pass transistor familier.* Kapittel 6.2.5.4 side 350.
6. *Differensielle kretser.* Kapittel 6.4.1 side 359 - 360.
7. *Sense-amplifier kretser.* Kapittel 6.4.2 side 360 - 362.
8. *Sample set differensiell logikk (SSDL).* Kapittel 6.4.2.1 side 362 - 363.
9. *Enable/Disable CMOS differensiell logikk (ECDL).* Kapittel 6.4.2.2 side 363 - 364.
10. *Latched CMOS differensiell logikk (LCDL).* Kapittel 6.4.2.3 side 364.
11. *Differensiell strøm svitsj logikk (DCSL).* Kapittel 6.4.2.4 side 364.
12. *BiCMOS kretser.* Kapittel 6.4.3 side 365.
13. *Sammenligning av CMOS logikk typer.* Kapittel 6.6 side 367 - 369.

II. CMOS MED TRANSMISJONSPORTER (Kapittel 6.2.5.1 side 347 - 348)

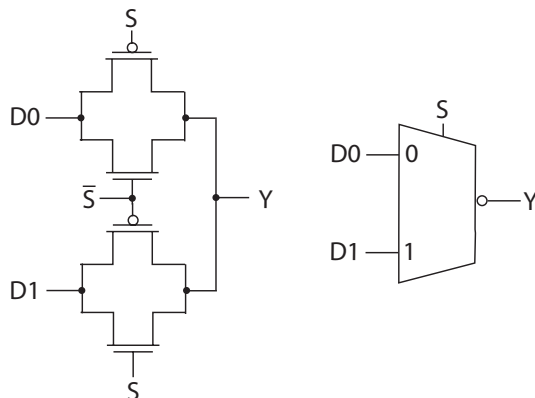


Fig. 1. *Transmisjons port multiplekser.*(FIG1.27)

Kretser slik som tristate, latcher og multipleksere er ofte tegnet med transmisjonsporter i tillegg til statisk CMOS logikk.

Et eksempel er transmisjons gate multiplekser som vist i figur 1.

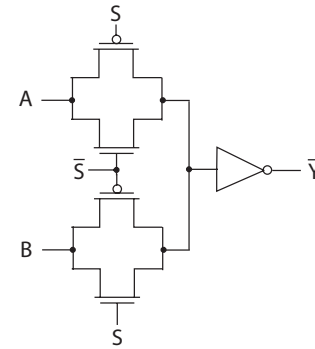


Fig. 2. *CMOS transmisjonsport med utgangsbuffer.*(FIG6.47)

En transmisjonsport kan utvides med et utgangsbuffer som vist i figur 2. Vi kaller en slik krets for *CMOSTG*. Utgangsbufferet er en inverter som vil sørge for at utgangen holder gode logiske verdier. I utgangspunktet kan man få inntrykk av at transmisjonsporter er fundamentalt forskjellige fra vanlig statisk CMOS. I realiteten er det ikke stor forskjell.

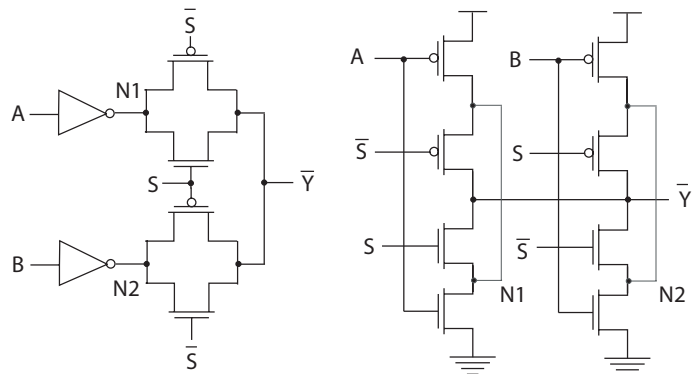


Fig. 3. *Alternative implementasjoner av .*(FIG6.48)

To ulike realiseringer av *CMOSTG* multipleksere er vist i figur 3. I kretsen til venstre er det brukt invertere og transmisjonsporter, mens kretsen til høyre er en statisk CMOS port. Dersom vi ser på de fire transistorene til venstre på kretsen til høyre i figuren ser vi at dette er en tristate¹ inverter med *A* som inngang. De fire transistorene til høyre utgjør en tristate inverter med *B* som inngang. Vi ser at dersom vi kobler sammen nodene mellom pMOS transistorene og nMOS transistorene i de to tristate inverterne, N1 og N2, vil vi få en helt identisk krets som til venstre i figuren. Kretsene blir da identiske, både logisk og elektrisk. Denne sammenkoblingen er unødvendig og vil bare bidra med økt motstand og kapasitans og derfor bidra med økt parasittisk tidsforsinkelse.

A. Mål

Kunne implemetere CMOS med transmisjonsporter.

¹Dette tilsvarer en klokke CMOS port *C²MOS*.

*Kaskode spennings svitsj logikk (CVSL). (INF3400 Del 8: Effektforbruk og statisk CMOS[2])

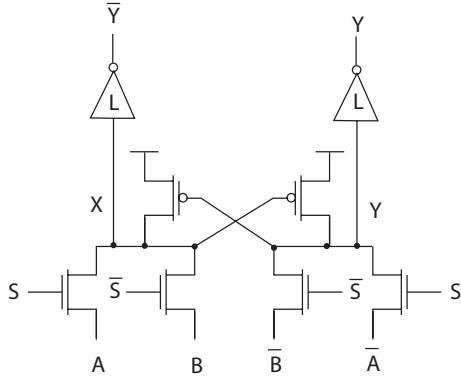


Fig. 4. Komplementær pass transistor logikk (CPL). (FIG6.51a)

Komplementær pass transistor logikk (CPL) er vist i figur 4. Vi ser at kretsen minner om CMOS TG. Inverterne på utgangen er lav skew der nedtrekket er prioritert.

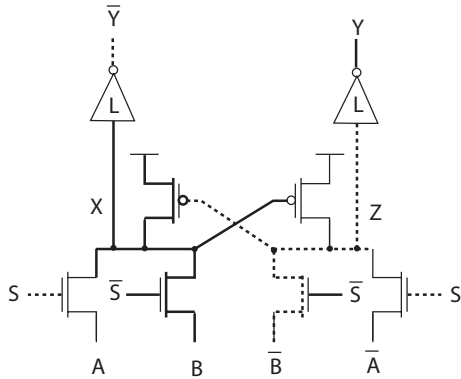


Fig. 5. Komplementær pass transistor logikk (CPL), $S = 0$ og $B = 1$. (FIG6.51a)

CPL multiplexer med $S = 0$ og $B = 1$ er vist i figur 5. 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.

CPL multiplexer med $S = 1$ og $A = 1$ er vist i figur 6. 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.

I figur 7 er utgangsinvertere til multiplkesere på inngangen vist. I dette tilfellet er utgangsinverterne droppet.

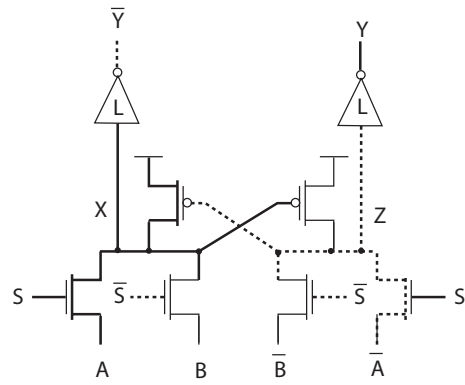


Fig. 6. Komplementær pass transistor logikk (CPL), $S = 1$ og $B = 1$. (FIG6.51a)

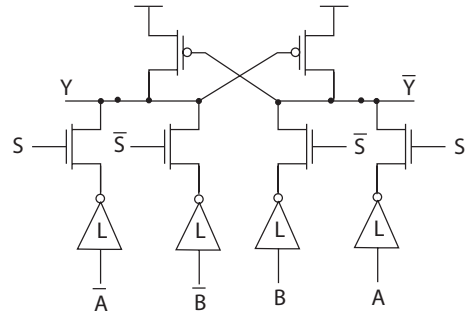


Fig. 7. Komplementær pass transistor logikk (CPL.) (FIG6.51b)

A. Mål

Kunne implementere multiplexere ved hjelp av komplementær pass transistor logikk (CPL).

B. Notater

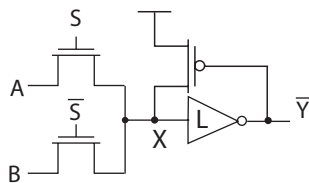


Fig. 8. Lean Integration med pass transistorer (LEAP) (FIG6.47)

En lean integration krets med pass transistorer (LEAP) er vist i figur 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.

A. Mål

Forstå hvordan lean integration krets med pass transistorer (LEAP) implementeres.

B. Notater

A. DPL

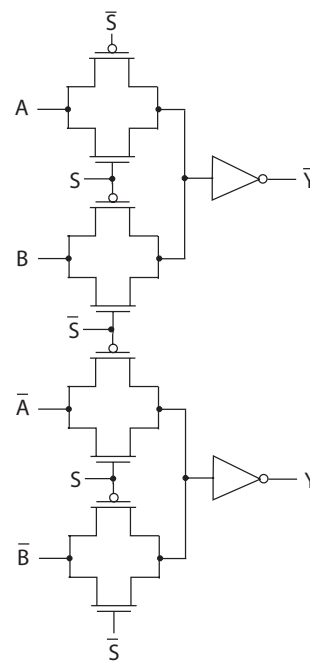


Fig. 9. Dobbelt pass transistor logikk (DPL) (FIG6.47)

Dobbelt pass transistor logic (DPL) er vist i figur 9. DPL ligner på CMOS, men består av dobbelt sett med transmosjonsporner når vi trenger inverterte utganger i tillegg til ordinære utganger. Kretsen trenger ikke tilbakekobling for å sikre gode logiske verdier.

B. EEPL

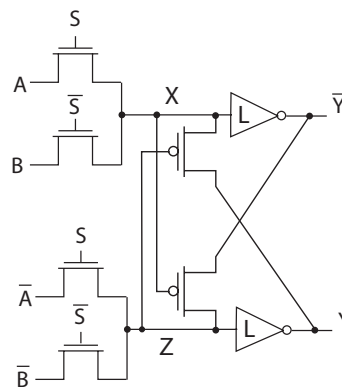


Fig. 10. Energi effektiv pass transistor logikk (EEPL). (FIG6.47)

Energi effektiv pass transistor logikk (EEPL) er vist i figur 10. 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.

C. PPL

Push-pull pass transistor logikk (PPL) er vist i figur 11. I dette tilfellet er det to nMOS pass transistorer og to pMOS

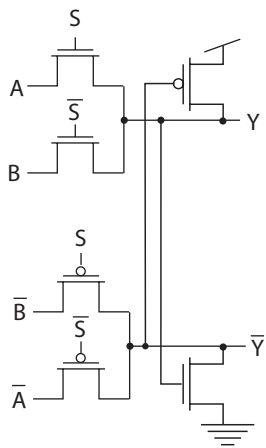


Fig. 11. Push-pull pass transistor logikk (PPL). (FIG6.47)

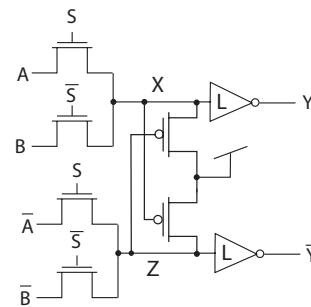


Fig. 13. Differensiell kaskode spenning svitsj med pass transistor logikk (DCVSPG). (FIG6.47)

G. Notater

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.

D. SRPL

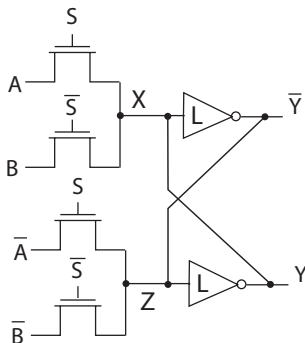


Fig. 12. Swing-restored pass transistor logikk (SRPL). (FIG6.47)

Swing-restored pass transistor logikk (SRPL) er vist i figur 12. 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 er det de to utgangsinverterne på utgangen krysskoblet som bidrar til å trekke begge utgangene til skikkelige logiske verdier.

E. DCVSPG

Differensiell kaskode spenning svitsj med pass transistor logikk (DCVSPG) er vist i figur 13. Logikken er til 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 ndetrekk som CVSL har.

F. Mål

Få et overblikk over ulike varianter av logikk med transmisjonsgater og pass transistorer.

* Kaskode spenning svitsj logikk (CVSL). (INF3400 Del 8: Effektforbruk og statisk CMOS[2])

Den vanligste CMOS logikken er statisk logikk. Dynamisk logikk som anvendes relativt ofte er domino-, pass-transistor- og pseudo nMOS logikk. I tillegg er det ikke uvanlig å anvende ulike typer *differensiell logikk* særlig i anvendelser der det er meget viktig å begrense digital støy. Typiske systemer er *mixed-mode* eller *mixed-signal* som er integrerte systemer der analoge og digitale delkretser deler samme substrat. Mange differensielle kretser er utviklet fra kaskode spenning svitsj logikk (CVSL).

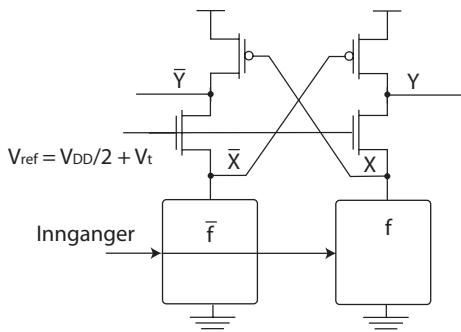


Fig. 14. *Differensiell split-level (DSL)*. (FIG6.61)

Differensiell split-level (DSL) er vist i figur 14. Nedtrekket består av komplementære nedtrekk lik CVSL, men i tillegg er det koblet nMOS transistorer i serie opp til utgangene. Disse nMOS transistorene har gate terminalen koblet til en spenningsreferanse $V_{ref} = V_{DD}/2 - V_t$, slik at utgangen av de komplementære nedtrekkene (X og \bar{X}) får redusert sving til $0 \rightarrow V_{DD}/2$. Dette reduserer parasittisk tidsforsinkelse ved nedtrekk som ble utnyttet til å redusere lengden på transistorene. I moderne design er imidlertid lengden ofte minimum slik at dette ikke er utnyttbart i praksis.

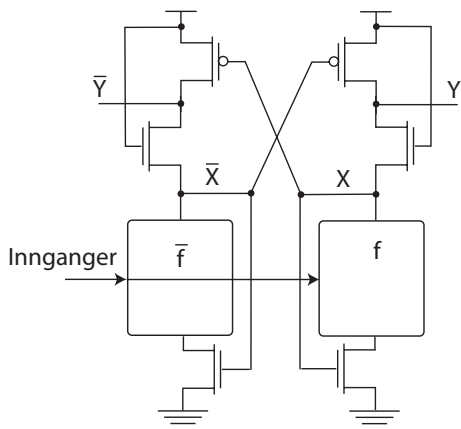


Fig. 15. *Kaskode nonthreshold logikk (CNTL)*. (FIG6.61)

Kaskode nonthreshold logikk (CNTL) er vist i figur 15. Logikken bygger på DSL og *nonthreshold logikk (NTL)* som er vist i figur 16. NTL logikk tilsvarer en pseudo nMOS logikk med tillegg en nMOS transistor og en kapasitans i parallell til GND . NTL er tregere enn pseudo nMOS logikk og bruker mer statisk effekt, og er derfor ikke vanlig å bruke. nMOS transistorene mot GND i CNTL logikk vil bidra til å redusere strømtrekket

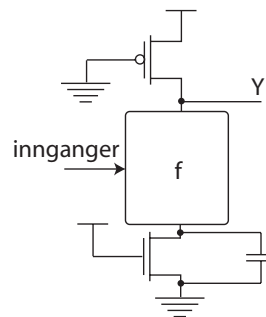


Fig. 16. *Nonthreshold logikk (NTL)*. (FIG6.62)

fordi transistorene vil stenge nedtrekket når utgangen på en av de komplementære blokkene trekkes ned mot GND .

Både DSL og CNTL er forholdsvis uvanlig å bruke i moderne CMOS.

A. Mål

Forstå hvordan DSL og CNTL logikk virker.

B. Notater

* Kaskode spenning svitsj logikk (CVSL). (INF3400 Del 8: Effektforbruk og statisk CMOS[2])

* Dual-rail domino logikk. (INF3400 Del 9: Dynamisk CMOS[3])

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 last² 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 endrer sin verdi. Sense-amplifier kretser er utsatt for ladningsdeling som kan påvirke kretsenes utgang fordi kretsen reagerer på små forandringer. Dette påvirker robusthet negativt.

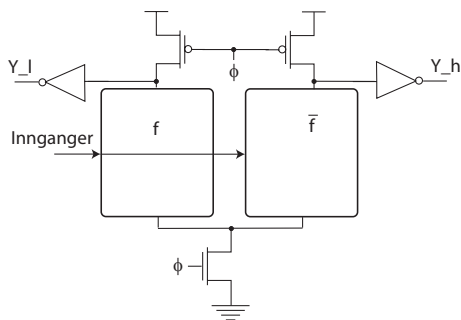


Fig. 17. Dual-rail domino logikk. (FIG6.63)

Sense-amplifier kretser er basert på CVSL kretser eller mer presist *dual-rail domino logikk* som er vist i figur 17. Vi kan betrakte sense-amplifier kretser som dual-rail domino logikk med en sense-amplifier for deteksjon og forsterkning av små differensielle spenninger.

En generisk sense-amplifier krets er vist i figur 18 øverst. Det er mest nyttig for komplekse nedtrekk som representerer betydelig parasittisk forsinkelse. Inngangene er differensielle som eksempelet nederst i figur 18 viser. Det vil alltid være slik at bare et av nedtrekkene er PÅ.

A. Mål

Forstå generisk sense-amplifier krets.

B. Notater

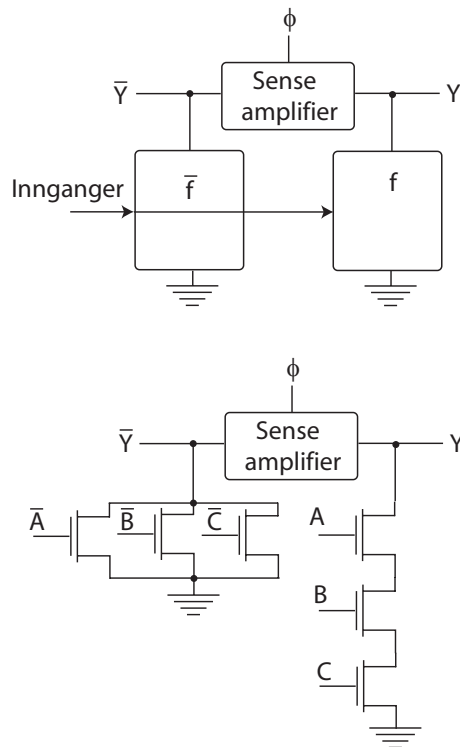


Fig. 18. Generisk sense-amplifier krets. (FIG6.64)

VIII. SAMPLE SET DIFFERENSIELL LOGIKK (SSDL) (Kapittel 6.4.2.1 side 362 - 363)

* Kaskode spenning svitsj logikk (CVSL). (INF3400 Del 8: Effektforbruk og statisk CMOS[2])

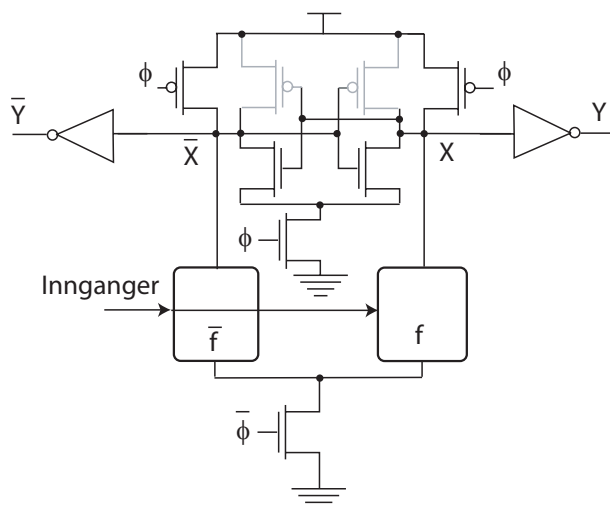


Fig. 19. Sample set differensiell logikk (SSDL). (FIG6.63)

Sample set differensiell logikk (SSDL) er vist i figur 19. 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 *sampler*. Vi må forutsette at nedtrekkene ikke kan overstyre pMOS transistorene som skal trekke både X og \bar{X} opp mot V_{DD} eller logisk

²Bitlinjene skal typisk drive svært mange transistorer.

1.

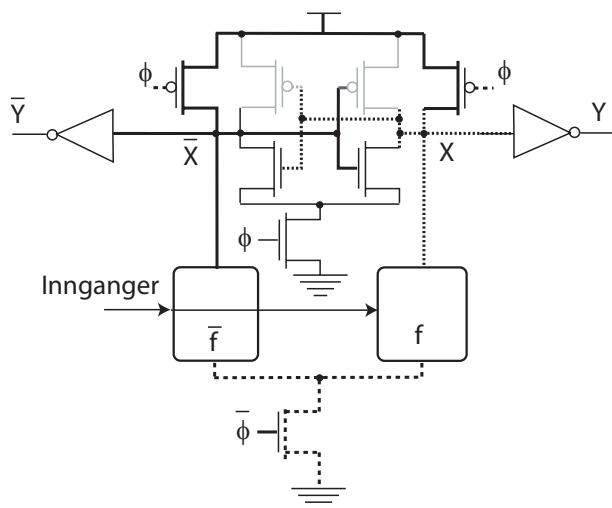


Fig. 20. *Sample set differensiell logikk (SSDL) ved sample fase.* (FIG6.63)

Sample set differensiell logikk (SSDL) ved sampling er vist i figur 20. Vi antar at inngangene er stabile slik at ett av nedtrekkene er PÅ (f i dette eksemplet). Dette 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-amplifiere 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 samler 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.

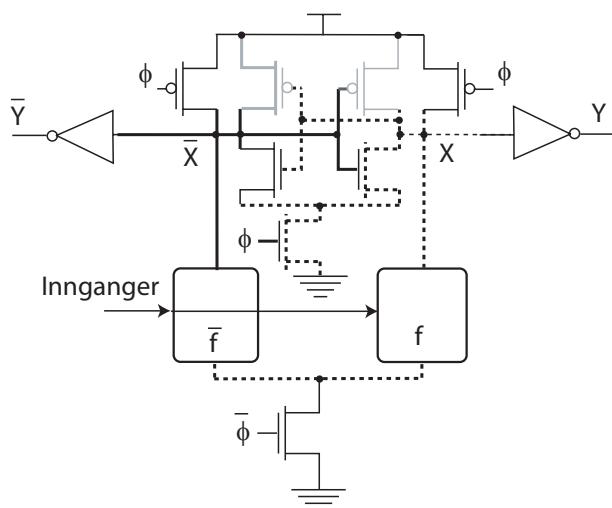


Fig. 21. *Sample set differensiell logikk (SSDL) ved set fase.* (FIG6.63)

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

figure 21. 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 holde motvirke reduksjon i den av utgangene X og \bar{X} som fortsatt skal være logisk 1.

A. Mål

Forstå hvordan SSDL logikk virker.

B. Notater

* Kaskode spenning svitsj logikk (CVSL). (INF3400 Del 8: Effektforbruk og statisk CMOS[2])

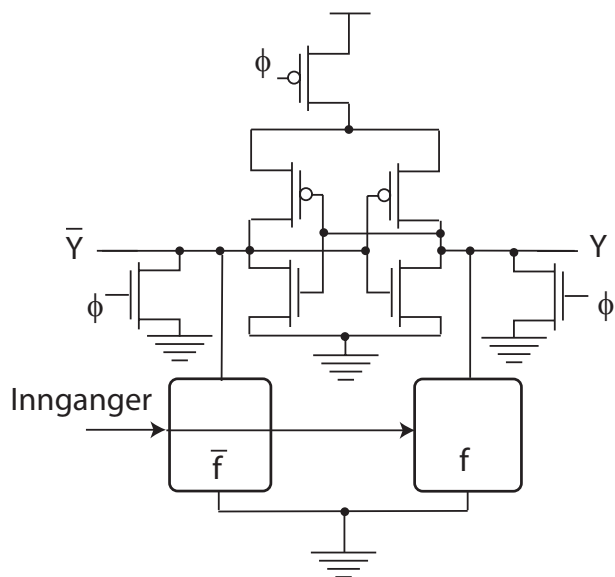


Fig. 22. Enable/disable CMOS differensiell logikk (ECDL). (FIG6.63)

Enable/disable CMOS differensiell logikk (ECDL) er vist i figur 22 representerer en forbedring av SSDL logikk ved at statisk effektforbruk blir redusert.

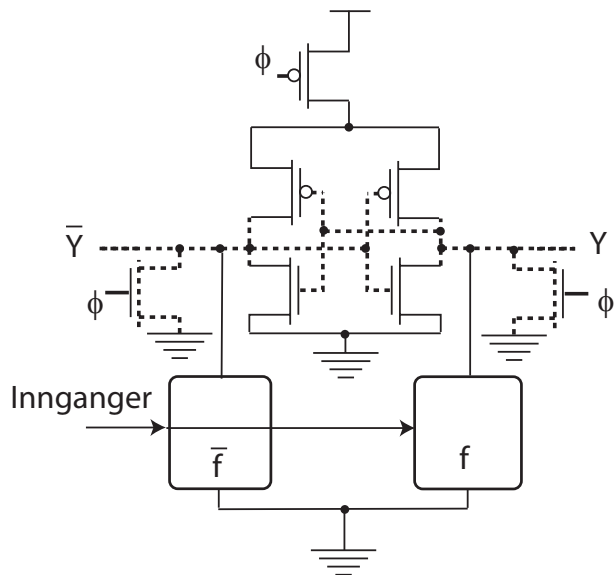


Fig. 23. Enable/disable CMOS differensiell logikk (ECDL) ved disable. (FIG6.63)

ECDL logikk ved disable er vist i figur 23. Porten disables når $\phi = 1$, slik at utgangene blir trukket ned til 0 som vist i figur 23. Her er "precharge" transistorene nMOS transistorer slik at de ikke vil danne en strømvei mellom spenningsreferansene V_{DD} 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.

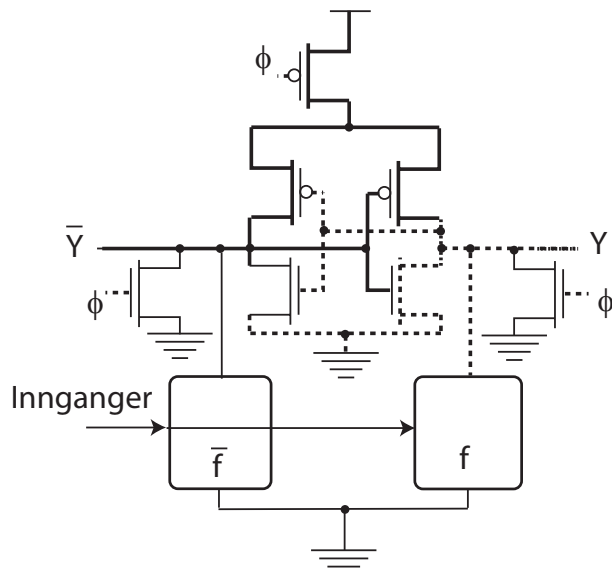


Fig. 24. Enable/disable CMOS differensiell logikk (ECDL) ved enable. (FIG6.63)

ECDL logikk ved disable er vist i figur 24. Porten disables når $\phi = 0$, slik at pMOS transistoren i toppen som er styrt av ϕ skrur av og "precharge" transistorene skrur av som vist i figur 24. 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.

A. Mål

Forstå hvordan ECDL virker.

B. Notater

X. LATCHED CMOS DIFFERENSIELL LOGIKK (LCDL)
(Kapittel 6.4.2.3 side 364)

* Kaskode spenning svitsj logikk (CVSL). (INF3400 Del 8: Effektforbruk og statisk CMOS[2])

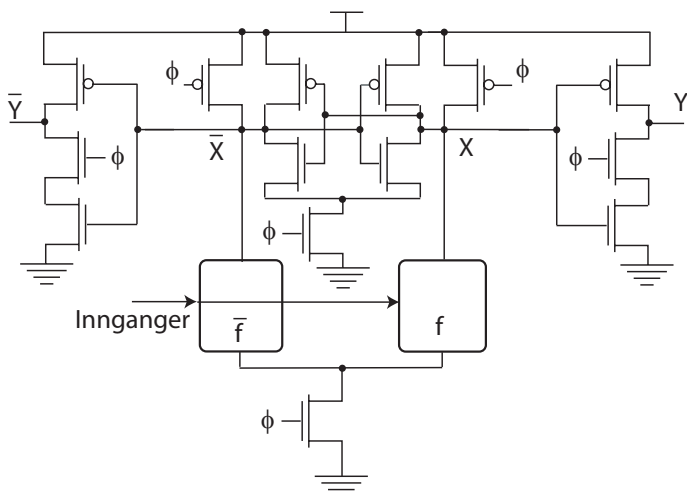


Fig. 25. Latched CMOS differensiell logikk (LCDL). (FIG6.63)

Latched CMOS differensiell logikk (LCDL) 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.

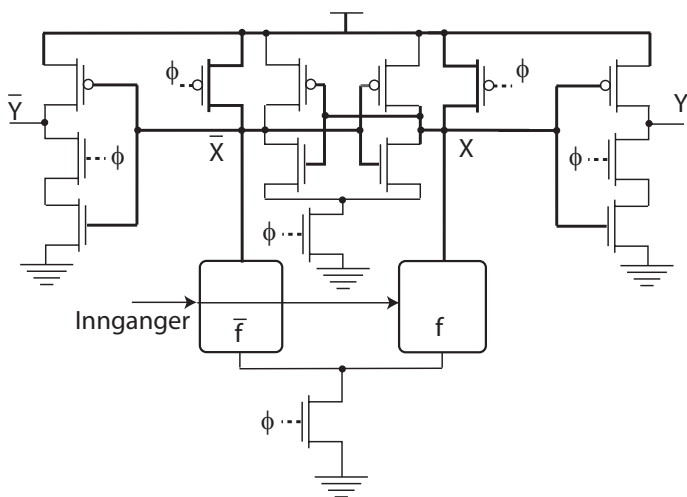


Fig. 26. Latched CMOS differensiell logikk (LCDL) ved precharge. (FIG6.63)

LCDL i precharge fasen er vist i figur 26 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 figur 26 der $\phi = 1$. Et av nedtrekkene vil nå trekke en av nodene X eller \bar{X} ned mot 0 og dermed trigge sense-amplifier kretsetn som raskt bidarr med å trekke den aktuelle noden raskt helt ned til 0. Utgangstrinnene virker nå bare logisk som invertere.

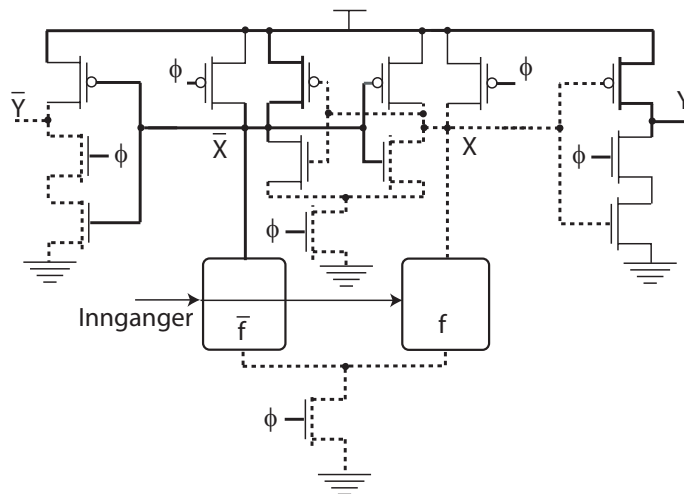


Fig. 27. Latched CMOS differensiell logikk (LCDL) ved evalering. (FIG6.63)

A. Mål

Forstå hvordan LCDL virker.

B. Notater

*Kaskode spenning svitsj logikk (CVSL). (INF3400 Del 8: Effektforbruk og statisk CMOS[2])

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

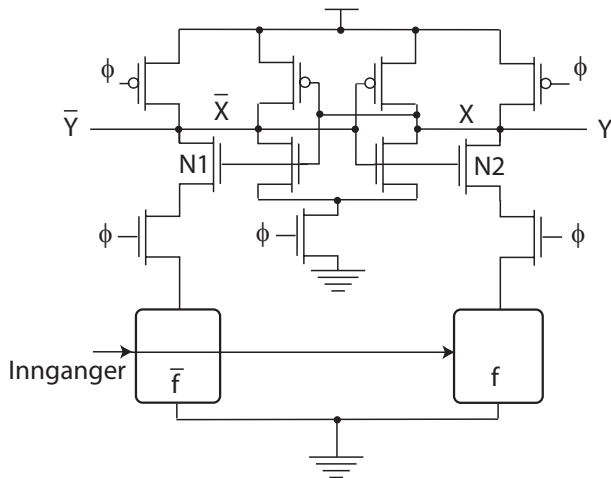


Fig. 28. Differensiell strøm svitsj logikk (DCSL). (FIG6.63)

Med differensiell strøm svitsj logikk (DCSL), som vist i figur 28 forsøker man å redusere effektforbruket relatert til interne noder og øke hastigheten ved å redusere spenningen som nedtrekket skal lade ut.

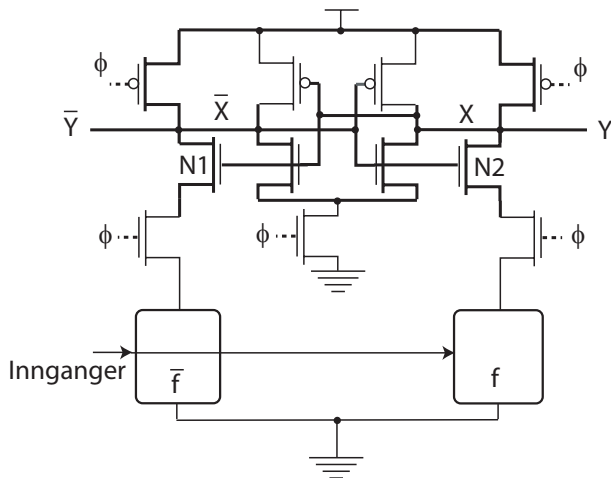


Fig. 29. Differensiell strøm svitsj logikk (DCSL) ved precharge. (FIG6.63)

DCSL ved precharge er vist i figur 29, 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 komplekse³ nedtrekkene. Interne noder i nedtrekkene vil ikke lades opp under precharge.

DCSL ved evaluering er vist i figur 30, 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

³Antar at nedtrekkene har mange interne noder.

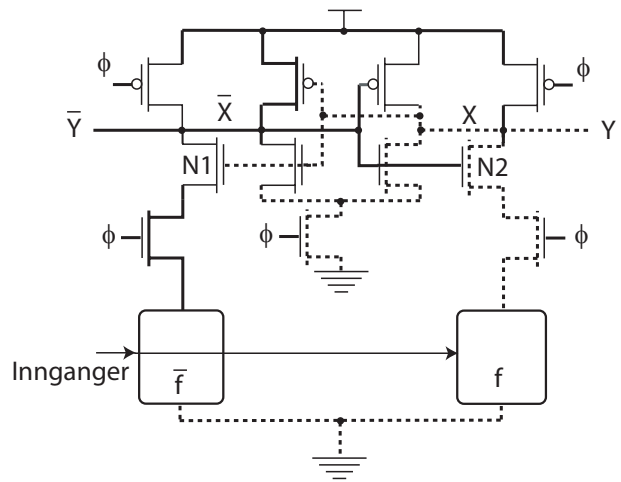


Fig. 30. Differensiell strøm svitsj logikk (DCSL) ved evaluering. (FIG6.63)

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.

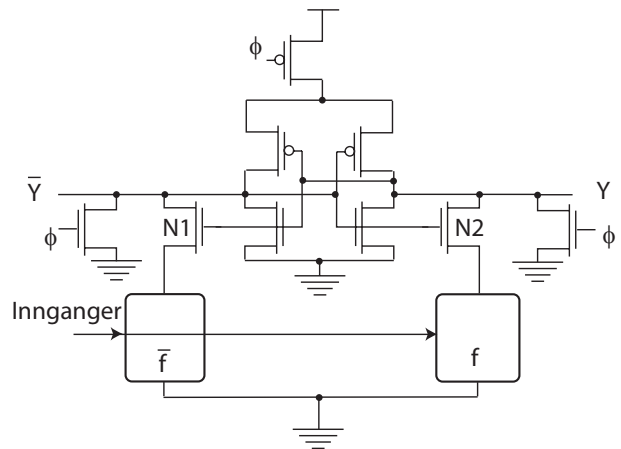


Fig. 31. Differensiell strøm svitsj logikk (DCSL2). (FIG6.63)

DCSL2 som er vist i figur 31 precharger utgangene til 0 tilsvarende ECDL. Transistorene N1 og N2 virker på tilsvarende måte som for DCSL1.

I DCSL3 er de to nMOS precharge transistorene ertattet med en pass transistor som kortslutter de to utgangene slik at begge utgangene blir precharge 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} .

A. Mål

Forstå hvordan ulike former for differensiell strøm svitsj logikk (DCSL) virker.

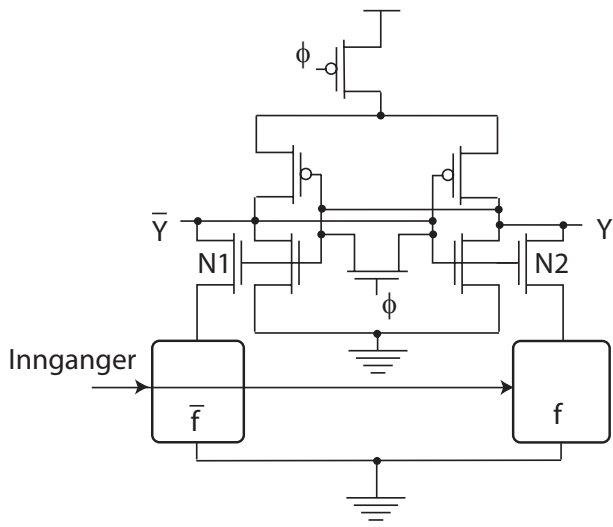


Fig. 32. Differensiell strøm svitsj logikk (DCSL3). (FIG6.63)

B. Notater

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.

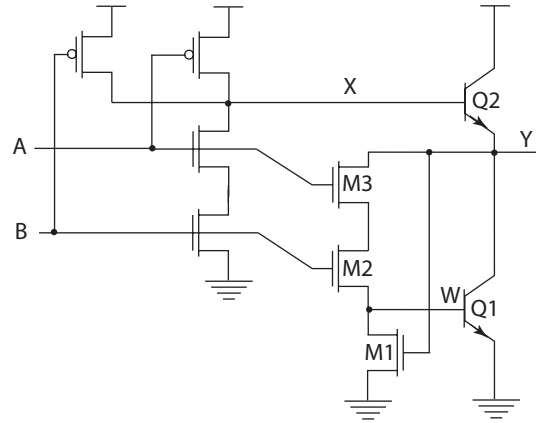


Fig. 33. BiCMOS NAND port. (FIG6.67)

En BiCMOS NAND port er vist i figur 33. 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.

A. Mål

Forstå hvordan enkle logiske porter kan implementeres i en BiCMOS prosess med bipolare transistor som utgangstrinn.

Logikk	nMOS	pMOS	Differens.	Stat. effekt	Sving	Dynamisk	Restore	Ratioed	Kaskade	Robust
Statisk	k	k	NEI	NEI	JA	NEI	JA	NEI	JA	Høy
pseudo nMOS	k	1	NEI	JA	NEI	NEI	JA	JA	JA	Middels
SFPL	2k+2	1	NEI	JA	NEI	NEI	JA	JA	JA	Middels
CVSL	2k	2	JA	NEI	JA	NEI	JA	NEI	JA	Høy
Dynamisk	k+1	1	NEI	NEI	JA	JA	JA	NEI	NEI	Lav
Domino	k+2	2	NEI	NEI	JA	JA	JA	NEI	JA	Lav
Dual-rail D.	2k+3	4	JA	NEI	JA	JA	JA	NEI	JA	Lav
CMOSTG	k	k	NEI	NEI	JA	NEI	JA	NEI	JA	Høy
LEAP	k	2	NEI	NEI	JA	NEI	JA	JA	JA	Middels
DPL	2k	2k	JA	NEI	JA	NEI	JA	NEI	JA	Høy
CPL	2k	4	JA	NEI	JA	NEI	JA	NEI	JA	Middels
EEPL	2k	4	JA	NEI	JA	NEI	JA	NEI	JA	Middels
SRPL	2k	2	JA	NEI	JA	NEI	JA	JA	JA	Lav
DCVSPG	2k-2	2	JA	NEI	JA	NEI	NEI	NEI	JA	Middels
PPL	k	k	JA	NEI	JA	NEI	NEI	NEI	JA	Lav
DSL	2k+2	2	JA	JA	NEI	NEI	JA	NEI	JA	Middels
CNTL	2k+4	2	JA	JA	NEI	NEI	JA	NEI	JA	Middels
NTL	k+1	2	NEI	JA	NEI	NEI	JA	JA	JA	Middels
SSDL	2k+6	6	JA	JA	JA	NEI	JA	NEI	NEI	LAV
ECDL	2k+4	3	JA	NEI	JA	NEI	JA	NEI	NEI	LAV
LCDL	2k+8	6	JA	NEI	JA	NEI	JA	NEI	NEI	LAV
D CSL1	2k+7	4	JA	NEI	JA	NEI	JA	NEI	NEI	LAV
BiCMOS	2k+1	k	NEI	JA	NEI	NEI	JA	NEI	JA	Middels

TABLE I
CMOS logikk.

XIII. SAMMENLIGNING AV CMOS LOGIKK TYPER (Kapittel 6.6 side 367 - 369)

Sammenligning av ulike typer CMOS logikk er gjort i tabell I. I tillegg til hastighet er det en rekke viktige faktorer som er knyttet til logikk typer; kompleksitet eller antall transistorer, om logikken er differensiell, statisk effektforbruk, utgangsving, dynamiske interne noder, signal restore, "ratioed" (transistor dimmensjonering i opptrekk/nedtrekk), mulighet for å koble i kaskade og robusthet.

A. Mål

Forstå grunnleggende forskjeller i ulike CMOS logikker.

B. Notater

XIV. INDEKS

CMOSTG 1
CNTL 5
CPL 2
D CSL 10
DCVSPG 4
Differensiell strøm svitsj logikk (D CSL) 10
Differensiell kaskode spenning svitsj logikk med pass transistor logikk (DCVSPG) 4
Differensiell logikk 5
Differensiell split-level (DSL) 5
Dobbel pass transistor logic (DPL) 3
DPL 3
DSL 5
Dual-rail domino logikk 6
ECDL 8
EEPL 3
Enable/disable CMOS differensiell logikk (ECDL) 8
Energi effektiv pass transistor logikk (EEPL) 3
Latched CMOS differensiell logikk (LCDL) 9
LCDL 9
LEAP 3
Lean integration med pass transistorer (LEAP) 3
Kaskode nonthreshold logikk (CNTL) 5
Komplementær pass transistor logikk (CPL) 2
Mixed-mode 5
Mixed-signal 5
Nonthreshold logikk (NTL) 5
NTL 5
PPL 3
Push-pull pass transistor logikk (PPL) 3
Sense amplifier kretser 6
SRPL 4

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*,
- [2] Yngvar Berg "INF3400 Del 8.XI"
- [3] Yngvar Berg "INF3400 Del 9.IV"