

UNIVERSITETET I OSLO
Institutt for informatikk

**Lavspennings
flernivålogikk
-En semi floating
gate inverter**

Hovedoppgave

Per Andreas Norseng

Mai 2007



Innhold

1	Forord	1
2	Innledning	3
3	Teori	5
3.1	Grunnleggende teoretisk bakgrunn	5
3.1.1	Flytende inngang (FG)	5
3.1.2	Lavspenning	5
3.1.3	Flernivåsignaler	6
3.2	Min implementasjon	6
4	Implementasjon	9
4.1	Kretsens virkemåte	9
4.1.1	Initialiseringsfasen	10
4.1.2	Evalueringsfasen	12
4.2	Flernivåsignaler	14
4.3	Dimensjonering	16
4.4	Utlegg	17
4.5	Hastighet, logisk dybde og forbruk	22
5	Resultater	29
5.1	Simuleringer	29
5.2	Målinger	35
6	Konklusjon og videre arbeid	45
6.1	Resultater	45
6.2	Videre arbeid	46
A	Programkode	49
A.1	bin.m	49
A.2	arb3ch.m	51
A.3	DumpScope.m	51

INNHold

B Instrumentoversikt	55
C Ordliste	57
Referanser	61

Kapittel 1

Forord

Denne hovedoppgaven for Candidatus Scientiarum-graden er skrevet ved avdeling for mikroelektronikksystemer, Institutt for informatikk, Universitetet i Oslo.

Professor Yngvar Berg har vært min veileder, og jeg vil takke ham for veiledning og gode diskusjoner. Videre vil jeg takke forsker Snorre Aunet for hans synspunkter og innspill, og 1. amanuensis Dag T. Wisland, stipendiat Johannes Goplen Lomsdalen, stipendiat Henning Gundersen, stipendiat René Jensen og sjefsingeniør Håvard Kolle Riis for god hjelp og veiledning på laboratoriet.

Til slutt vil jeg takke min familie for korrekturlesing og språkvask.

KAPITTEL 1. FORORD

Kapittel 2

Innledning

Tradisjonell binær logikk har lenge vært nærmest enerådende og har selvfølgelig sine fordeler og ulemper som alle andre typer logikk. De største fordelene er i hovedsak hastighet og nøyaktighet. Hastigheten har økt samtidig som transistorene har blitt mindre og dette har ført til økt varmeutvikling og effektforbruk. Et annet problem er at signalrout- ing tar større og større plass jo flere transistorer kretsene inneholder og kan faktisk ta opp så mye som 75% av brikkens areal. Vi vet nå også at minimumsgrensen for transistorstørrelse vil bli nådd, og når det skjer vil binær logikk ha få, hvis noen, muligheter til å øke ytelsen siden man ikke vil få plass til fler transistorer på en brikke.

Her er det altså to hovedproblemer som må løses i fremtidige applikasjo- ner: effektforbruk og hastighet. Når det gjelder effektforbruk kan dette blant annet reduseres ved å benytte seg av såkalt *ultra lavspenningslo- gikk (ULV)* som baserer seg på *floating gate* (FG eller flytende inngang) og lave forsyningsspenninger. FG-transistoren ble først introdusert av Shibata og Ohmi i 1991 [8], og er grunnlaget for mye av utviklingen som senere er gjort på området.

Senkes forsyningsspenningen, reduseres også hastigheten, men i kom- binasjon med *flernivålogikk (MVL)* kan man opprettholde og til og med øke ytelsen i et system, fordi flere operasjoner kan utføres med færre transistorer. Et resultat av dette er også at varmeutviklingen og effekt- forbruket går ned.

Denne oppgaven behandler, både teoretisk og praktisk, egenskapene til en *ULV semi floating gate*-inverter beregnet for å operere på lave spen- ninger med lavt effektforbruk. Funksjonaliteten demonstreres ved hjelp av simuleringer og målinger på to produserte kretser i hhv. $0,35\mu\text{m}$ og

KAPITTEL 2. INNLEDNING

0,13 μm CMOS-teknologi. Deler av dette arbeidet er publisert internasjonalt.

ULV-inverteren har blitt sammenlignet med såkalt *footed domino logic* (*FDL*) og simuleringer antyder at *ULV*-inverteren kan operere 10 ganger raskere enn *FDL*-inverteren, samtidig som effektforbruket reduseres når frekvensen øker på grunn av mindre statisk enregi. Det har også blitt vist at kretsen kan drives med så lave spenninger som 0,36V. Det har også blitt vist simuleringer av flernivåfunksjonalitet.

Kapittel 3

Teori

3.1 Grunnleggende teoretisk bakgrunn

Målet med oppgaven er å produsere og teste en inverter som er konstruert for å drives med lave spenninger for å redusere effektforbruket. Min veileder, professor Yngvar Berg, har konstruert kretsen. Den kombinerer deler av forskjellige konstruksjonsteknikker, og jeg derfor gå kort igjennom hva disse teknikkene/metodene går ut på.

3.1.1 Flytende inngang (FG)

FG-transistorer er en variant av transistoren som tradisjonelt har vært mye brukt i ikke-volatilt minne som flash, EPROM og EEPROM [3, 5]. En FG-transistor kan holde på en ladning i en begrenset tidsperiode uten å ha noen forsyning. Ladningen på den flytende inngangen kan endres ved å endre spenningen på source, drain eller FG-terminalen. Sagt på en annen måte er det slik at inngangen på transistoren er koblet av systemet med en kondensator slik at det ikke er selve nivået på inngangen som styrer transistoren, men eventuelle endringer av spenningen på inngangen. I løpet av de siste 17-18 årene har utviklingen av andre typer kretser basert på FG skutt fart [4]. FG har blant annet blitt brukt mye i kretser inspirert av nevralt nettverk og i den senere tid også til å løse problemer i både analoge og digitale systemer.

3.1.2 Lavspenning

Å operere elektronikk på lav spenning er en av de enkleste måtene å redusere effektforbruket på. Når vi i denne sammenheng snakker om

lavspenning (LV) og ultralavspenning (ULV), dreier det seg om å operere kretsene i området rundt og under terskelspenningen for kretsen. Når man kommer ned på så lave spenninger, oppfører transistorene seg annerledes og man må ta hensyn til og dra nytte av dette i måten man konstruerer kretsen på.

3.1.3 Flernivåsignaler

Når kompleksiteten i binære kretser øker, brukes store deler av arealet til routing av signaler, slik at effektforbruket øker, og man får dermed stor varmeutvikling. For å løse deler av dette problemet kan man benytte seg av flere logiske nivåer i motsetning til de binære "0" og "1". På denne måten reduseres antall signallinjer fordi man øker informasjonsinnholdet per linje.

En krets som benytter flernivålogikk vil redusere det totale antall porter som må brukes for å gjøre samme jobb som en binær krets. Et radix 8-signal (8 nivåer) tilsvarer 3 bit ($2^3 = 8$), og man vil da kunne redusere antall porter fra 3 til 1 for en gitt operasjon. I et stort system vil dette, som sagt, kunne ha stor betydning for arealet og effektforbruket og dermed varmeutviklingen.

3.2 Min implementasjon

Lavt effektforbruk kan blant annet bety lengere batterilevetid, noe som er viktig i elektronikk innenfor mange sektorer som for eksempel medisinsk utstyr som øreimplantater eller blodsuktermålere som implanteres i kroppen. Å redusere forsyningsspenningen er den enkleste og mest effektive måten å redusere effektbehovet på [7], og vi har prøvd å redusere en del av ulempene denne reduksjonen medfører.

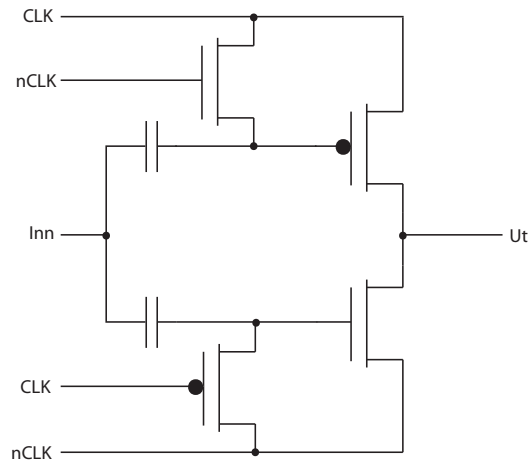
I denne oppgaven har jeg valgt å fokusere på binær funksjonalitet, og vil bare kort komme inn på flernivåfunksjonalitet da det rent designmessig kun er snakk om å benytte en kapasitiv tilbakekobling.

Når forsyningsspenningen senkes, vil dette påvirke hastigheten negativt. Det har vært foreslått flere teknikker for bruk i lavspenning/høytelessystemer, blant annet såkaldt fler- V_{dd} -teknikk [9] hvor man benytter en normal/høy forsyningsspenning i de kritiske delene av kretsen, mens man i mindre kritiske deler av systemet kan benytte en lav

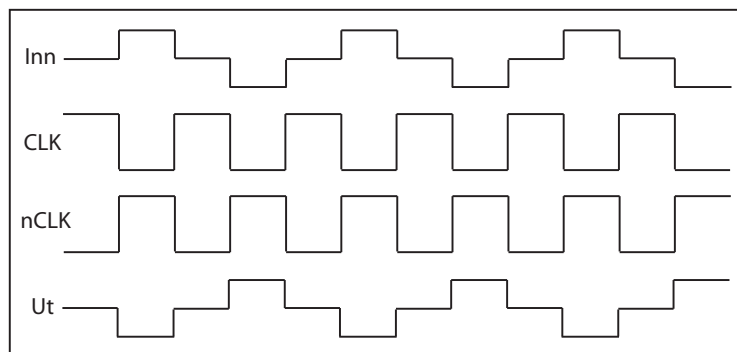
forsyningsspenning for å få ned det totale effektforbruket. I tillegg til dette kan man også skalere ned terskelspenningen, V_t , til transistorene for å opprettholde ytelsen, men dette gir en eksponentiell økning i lekkasjestrøm slik at det statiske effektforbruket øker. Dette har også blitt brukt på en lignende måte i [6] ved å redusere terskelspenningen i de ytelseskritiske delene av kretsen, mens man har beholdt en høy terskelspenning i resten av kretsen. Dette vil redusere problemene med økt lekkasje når forsyningsspenningen reduseres.

FG-teknikken går ut på at inngangen på transistoren er "koblet av" systemet med en kondensator slik at transistoren ikke reagerer direkte på selve spenningsnivået på inngangen til kretsen, men på eventuelle endringer på denne inngangen. Hvor mye af inngangssignalet som slipper igjennom er avhengig av størrelsen på inngangstransistorene i forhold til den totale kapasitansen på inngangen til transistoren. I en slik krets kan det være vanskelig å få utgangen til å bli helt riktig med en forsterkning på -1 og en nødvendig forutsetning er da nøyaktig transistor-match. Dette er i praksis vanskelig ettersom like transistorer ikke nødvendigvis blir helt identiske i produksjon. For å løse dette problemet kan man for eksempel gi systemet en initialverdi, men denne kan likevel endre seg over tid. Dette kan løses ved å initialisere kretsen én gang hver klokkeperiode. Transistorinngangen flyter da ikke helt på egenhånd og dette kalles derfor "semi floating gate" (*SFG*). Jeg vil i avsnitt 4.1 på side 9 komme nærmere inn på hvordan dette fungerer. På grunn av den store lekkasjen i CMOS-teknologiene egner ikke FG seg til ikke-volatile kretser som for eksempel RAM, FLASH og andre lagringsmidler som ikke skal miste informasjon. For å unngå tap av informasjon må portene ha hyppige reinitialiseringer/oppdateringer.

Figur 3.1 på neste side viser skjematikken for den binære varianten av kretsen, og figur 3.2 på neste side viser hvordan kretsen skal oppføre seg rent logisk.



Figur 3.1: Skjematikk for LV inverter



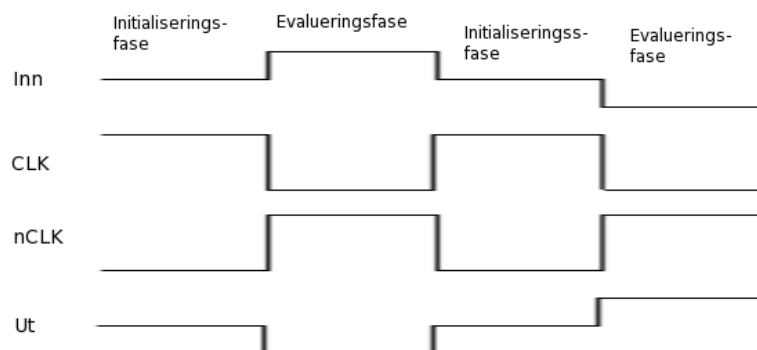
Figur 3.2: Signalrespons

Kapittel 4

Implementasjon

I første omgang velger jeg å undersøke en binær variant av kretsen (se figur 3.1 på forrige side), da det for å oppnå flernivåfunksjonalitet kun er behov for å benytte to tilbakekoblingskondensatorer fra utgangen til *SFG*-inngangene (se avsnitt 4.2 på side 14). For å gjøre illustrasjonene så enkle som mulig er også brønntilkoblingene utelatt.

4.1 Kretsens virkemåte

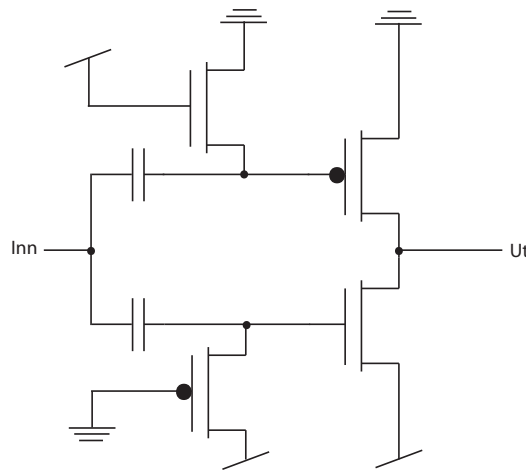


Figur 4.1: Signalrespons over to perioder med faser

For å se nærmere på kretsens virkemåte deler jeg opp kretsen i to faser: initialisering og evaluering. I figur 4.1 vises "ideell" signalrespons med fasene markert. Først ser vi på den halve klokkeperioden hvor kretsen initialiseres (recharge). I figurene benytter jeg *gnd* og V_{dd} som referanser i stedet for klokkene for at det skal være klarere om nivået er høyt

eller lavt. Man kan altså se det slik at gnd representerer en spenning som er lik eller lavere enn $0V$ og V_{dd} representerer en spenning som er lik eller høyere enn forsyningsspenningen. Nå er det også viktig å huske at kretsen ikke har en forsyningsspenning i ordets rette forstand: den drives bare av klokken, så med forsyningsspenning mener jeg et nivå som er likt det høye nivået til inngangssignalet. Inngangssignalet eller egentlig utgangssignalet begrenses oppad og nedad av henholdsvis det høye nivået til $nclk$ og det lave nivået til clk . Figur 3.2 på side 8 viser hvordan kretsen skal oppføre seg rent logisk.

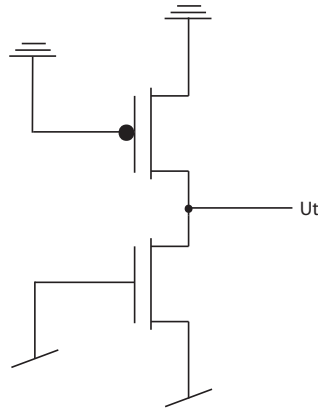
4.1.1 Initialiseringsfasen



Figur 4.2: Skjematikk for initialiseringsstadiet

Figur 4.2 viser hvordan initialiseringstransistorene leder strøm og setter hver sin SFG til hhv. gnd og V_{dd} . Inngangen får ingen betydning og man får den forenklete skjematikken i figur 4.3 på neste side.

De to transistorene i inverterdelen av kretsen er på, og at utgangen vil være source for begge transistorene. Det er imidlertid V_{gs} som styrer transistorene så dette vil være helt avhengig av hvilket nivå utgangen hadde i evalueringsfasen. For å få et bilde av hva som skjer, må man altså se på hvilket nivå utgangen lå på før initialiseringsfasen. Vi tenker oss at utgangen gikk lav i siste evalueringsfase, og at inngangen altså var høy. Fra artikkelen *Ultra Low Voltage CMOS gates* [2] har vi da at spenningen på SFG_n i et tenkt system kan uttrykkes slik:



Figur 4.3: Forenklet skjematikk for initialiseringsfasen

$$V_{SFG_n} \approx V_{dd} + (C_{inn_n}/C_{total_{SFG}}) \times V_{dd}/2 \approx (5/4)V_{dd}$$

Når klokkene vender kretsen over til ny initialiseringsfase, vil spenningen på inngangen til p-transistoren (V_{SFG_p}) faktisk være lavere enn null og V_{SFG_n} (for n-transistoren) være høyere enn V_{dd} . Dette betyr at nMOS-transistoren først vil operere i sterk inversjon fordi den effektive spenningen er:

$$V_{eff} = V_g - V_s - V_t \approx (5/4)V_{dd} - 0V - V_t \approx (1/4)V_{dd}$$

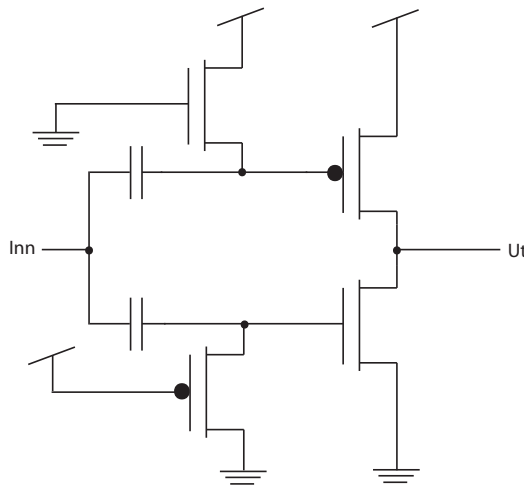
Kretsen vil nærmest få et sjokk, og siden utgangen er *source* for begge transistorene vil både p- og n-transistoren trekke utgangen fort oppover. Dermed vil V_{gs_n} synke raskt og V_{gs_p} vil stige raskt, men begge transistorene forblir på og balanserer utgangen på $V_{dd}/2$. For et binært system er det uproblematisk hvor nær $V_{dd}/2$ utgangen havner. Det er bare et spørsmål om hvor lik systemets responstid for opp- og nedtrekk blir i evalueringsfasen. Så lenge utgangen er i nærheten av $V_{dd}/2$, vil det ikke påvirke systemets robusthet, men det kan selvfølgelig påvirke hastigheten i negativ forstand. Jo lenger unna $V_{dd}/2$ utgangen er, jo større forskjell i responstid mellom opp- og nedtrekk og jo lavere vil systemets maksimale hastighet bli. Dette kan til dels kompenseres for ved å forandre forholdet mellom størrelsene på p- og n-transistorene, men dette kan i sin tur påvirke arealet og dermed effektforbruket.

Som nevnt over vil transistor-mismatch og variasjon i terskelspenning påvirke utgangsspenningen i initialiseringsfasen. Hvis for eksempel n-transistoren er sterk i forhold til p-transistoren vil utgangen stabilisere seg på en spenning som er høyere enn $V_{dd}/2$. Når porten så går over

i evalueringsfasen vil den sterke n-transistoren, hvis inngangen er høy, dra utgangen ned mot $gndog$ dermed redusere V_{SFG_n} og V_{SFG_p} gjennom parasittiske kapasitanser (C_{gd}/C_{gs}) i p- og n-transistorene. Det som faktisk skjer er at effekten av problemet med mismatch reduseres gjennom lokale, parasittiske kapasitanser.

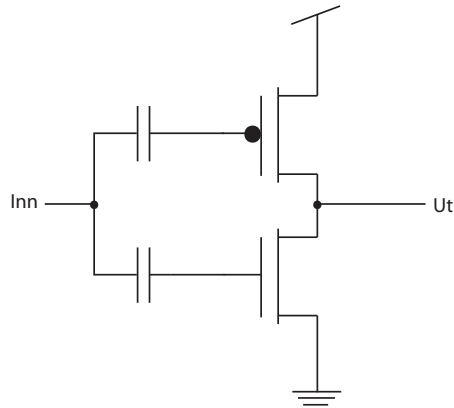
4.1.2 Evalueringsfasen

Når så klokkene vender, kommer vi over i fasen hvor kretsen evaluerer inngangssignalet. Figur 4.4 viser kretsen i evalueringsfasen og siden begge initialiseringstransistorene er slått av, kan man se bort fra dem. Skjematikken blir da som i figur 4.5 på neste side, og da ser vi at skjematikken for kretsen blir mye enklere å analysere. Det som skjer er at hvis inngangssignalet går eller har gått opp strupes p-transistoren og n-transistoren åpnes. Dermed trekkes utgangen ned. Hvis inngangssignalet derimot går ned, vil p-transistoren åpne seg og n-transistoren lukke seg. Utgangen vil dermed trekkes opp til V_{dd} , slik at kretsen vil få en respons som i figur 4.1 på side 9. Dette er selvfølgelig en stor forenkling av kretsens oppførsel, men viser hvordan den er ment å fungere rent logisk.



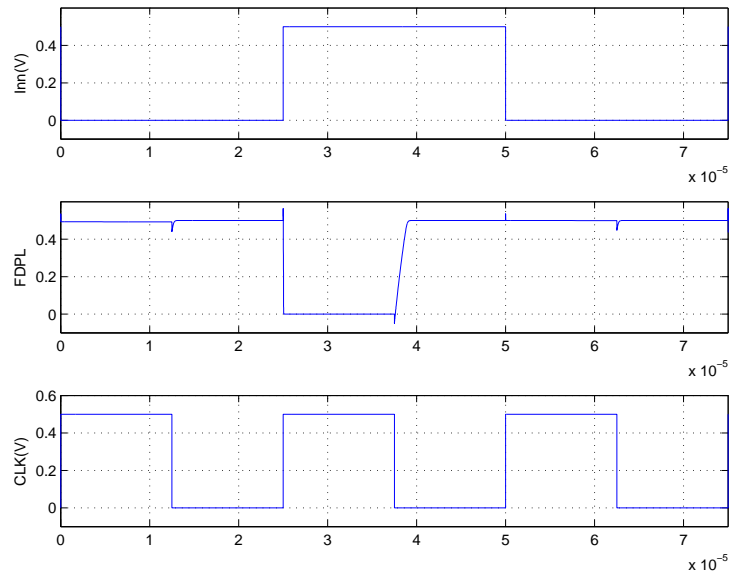
Figur 4.4: Skjematikk for evalueringsfasen

En av de store fordelene med denne kretsen, sammenlignet med *footed domino logic (FDL)* er at utgangen initialiseres til $V_{dd}/2$ og ikke til V_{dd} . Dette fører til at kretsen er like rask på både opptrekk og nedtrekk i



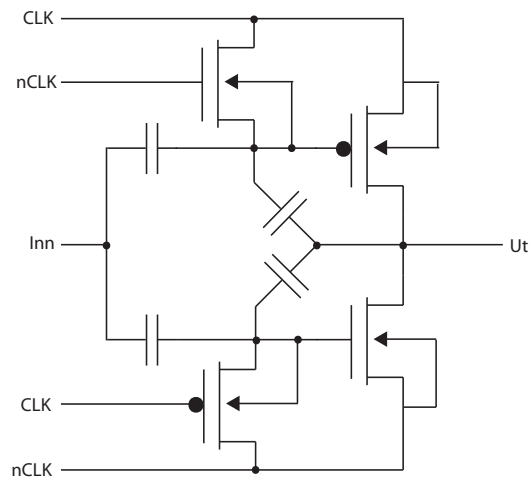
Figur 4.5: Forenklet skjematikk for evalueringsfasen

evalueringsfasen. Som man ser av simuleringen av *FDL*-inverteren i figur 4.6 på neste side, vil det være stor forskjell i responstiden for opptrekk og nedtrekk, og det vil være nedtrekkstiden som bestemmer den maksimale hastigheten en slik krets kan operere på.

Figur 4.6: Simulering av *FDL*-inverter

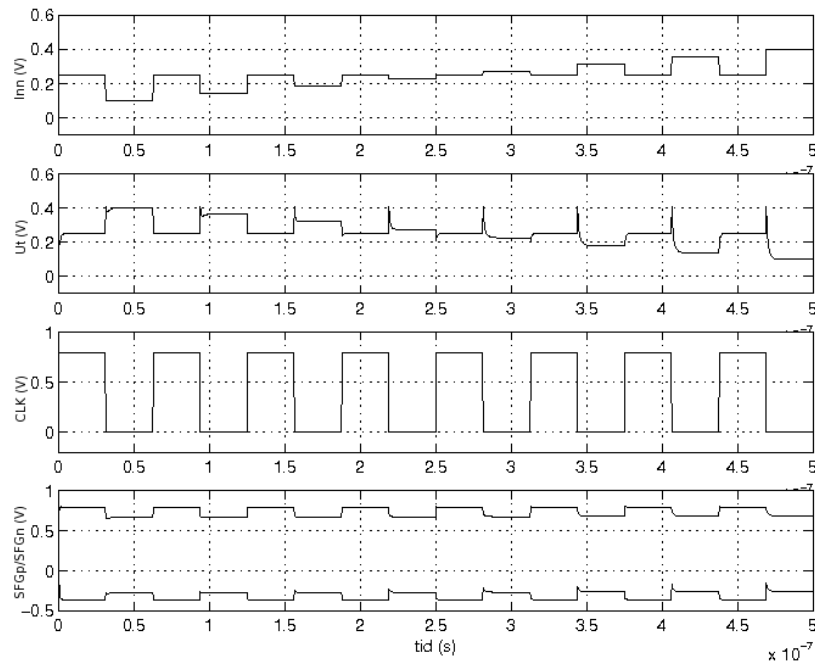
4.2 Flernivåsignaler

For flernivåsignaler er det i prinsippet bare snakk om to tilbakekoblingskondensatorer som vist i figur 4.7 på neste side. Disse må dimensjoneres i forhold til inngangskondensatorene, slik at man får utnyttet hele signalområdet. Når man opererer med flernivåsignaler, kan man ikke bruke hele spenningsområdet til forsyningene: utgangstransistorene må være i metning og inngangssignalet må derfor ha det som kalles *head room*. Hvis man opererer med 0,5V forsyningsspennning må man la signalet ligge mellom 0,05V og 0,45V. Dekke kan variere fra prosess til prosess, og disse tallene er godt innenfor marginene i prosessene jeg har brukt. Jeg har, som tidligere nevnt, valgt å ikke fokusere så mye på flernivåsignaler, og i utgangspunktet er mine to implementasjoner av kretsen uten tilbakekoblinger. En ting som er verdt å merke seg er at når man går ned i produksjonsstørrelse så øker blant annet de parasittiske kapasitansene relativt sett: på kretsen som er produsert i $0,13\mu\text{m}$ er det en kapasitiv kobling mellom utgang og *SFG*-nodene som er signifikant nok til at det faktisk er mulig å kjøre flernivåsignaler på denne kretsen. Dette oppdaget jeg ved en ren tilfeldighet og har dessverre ikke lagret målinger som viser dette. Figur 4.8 på side 16 viser en simulering av min $0,35\mu\text{m}$ -krets med et radix 8-signal med *head room* på 100mV slik at de 8 nivåene lig-



Figur 4.7: flernivå-inverter

ger mellom $100mV$ og $400mV$.

Figur 4.8: Simulering ved 0.5V, 8MHz, 0,35 μ m

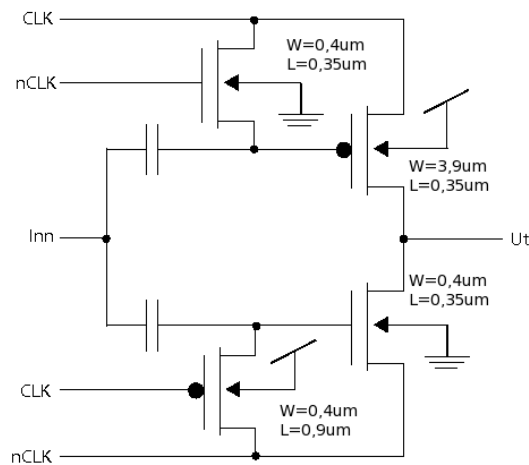
4.3 Dimensjonering

For å optimalisere kretsen har jeg, siden dette er en såpass liten krets, endret de forskjellige parametrene manuelt og simulert. Jeg har brukt en tradisjonell inverter for å finne forsterkningsforholdet mellom p- og n-transistorer og funnet at et forhold på 3,05 mellom β_n og β_p i AMS 0,35 μ m-prosess ved 0,5V gir likt opp- og nedtrekk. Jeg har likevel valgt å øke bredden på p-transistoren litt i forhold til dette for å kompensere for andre effekter i kretsen.

Det er også slik at forholdet mellom β_p og β_n varierer med forsyningsspenningen, og dette betyr at man må dimensjonere med tanke på den forsyningsspenningen man skal bruke. Beveger man seg i ettertid vekk fra denne forsyningsspenningen ved simuleringer eller målinger, vil forsterkningsforholdet mellom p- og n-transistorene bli feil. I mitt tilfelle hvor det er mest aktuelt å senke forsyningsspenningen vil p-transistorene bli svakere og opptrekket dermed bli dårligere. Dette kan sees på målingene i avsnitt 5.2 på side 35. Figur 4.9 på neste side og figur 4.10 på side 18 viser transistor- og kondensatorstørrelsene jeg har benyttet for kretsene i hhv. 0,35 μ m-prosessen og 0,13 μ m-prosessen.

4.4 Utlegg

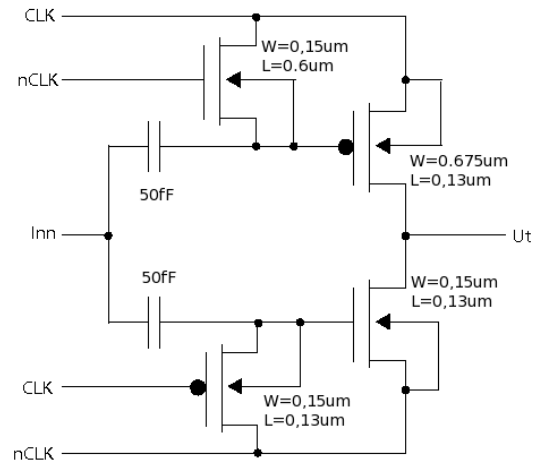
Standard-prosessen som ble benyttet til kretsproduksjon ved instituttet var en $0,35\mu\text{m}$ prosess fra AMS. Denne prosessen har imidlertid ikke mulighet for å isolere brønnen til n-transistoren slik at mitt design som vist i 4.10 på neste side ikke kunne produseres. Jeg valgte derfor å modifisere kretsen slik at den kunne produseres med denne prosessen. Brønnene ble koblet til statiske referanser som vist i figur 4.9.



Figur 4.9: LV inverter m/fast brønn.

Samtidig som jeg la ut den modifiserte kretsen og fikk den produsert, søkte vi også etter en prosess som hadde mulighet for å isolere alle brønnene. Jeg fant en $0,13\mu\text{m}$ "triple well"-prosess fra STM og jeg gikk i gang med å tilpasse den opprinnelige kretsen til den nye prosessen. Dette viste seg å være meget vanskelig da det manglet informasjon om design-kitet og oppbyggingen av transistorene og isoleringen av brønnene var komplisert. I tillegg var designkittet til denne prosessen dårlig tilpasset Cadence slik at det endte med at vi måtte kjøre en del tester som LVS og DRC på en tjener på NTNU. Figur 4.12 på side 20 viser utlegget for inverteren i $0,13\mu\text{m}$ -prosessen med brønnene på initialiserings-transistorene koblet til hhv. SFG_p og SFG_n . Her ser vi at n-transistorene tar mye plass på grunn av isoleringen av brønnen til hver transistor for seg. Utleppet for kretsen i $0,35\mu\text{m}$ som man kan se i figur 4.11 på side 19 er adskillig mer kompakt i så måte med felles brønn for hhv. p-transistorene og n-transistorene, men routingen er noe mer komplisert

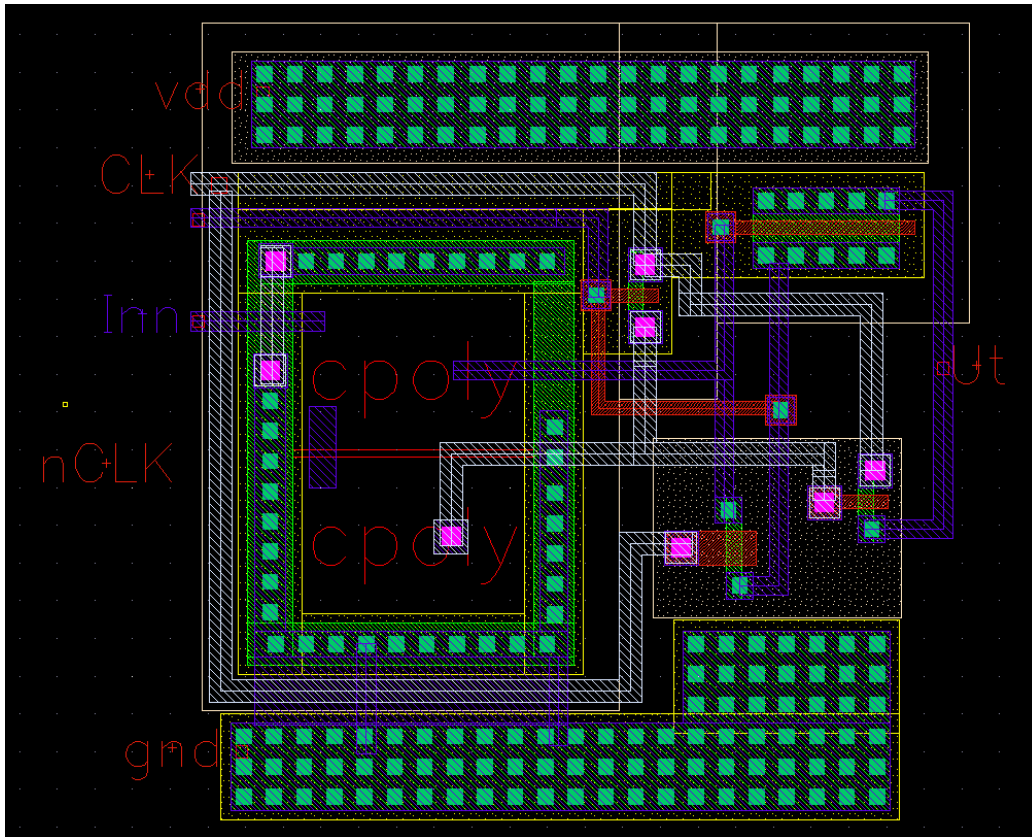
KAPITTEL 4. IMPLEMENTASJON

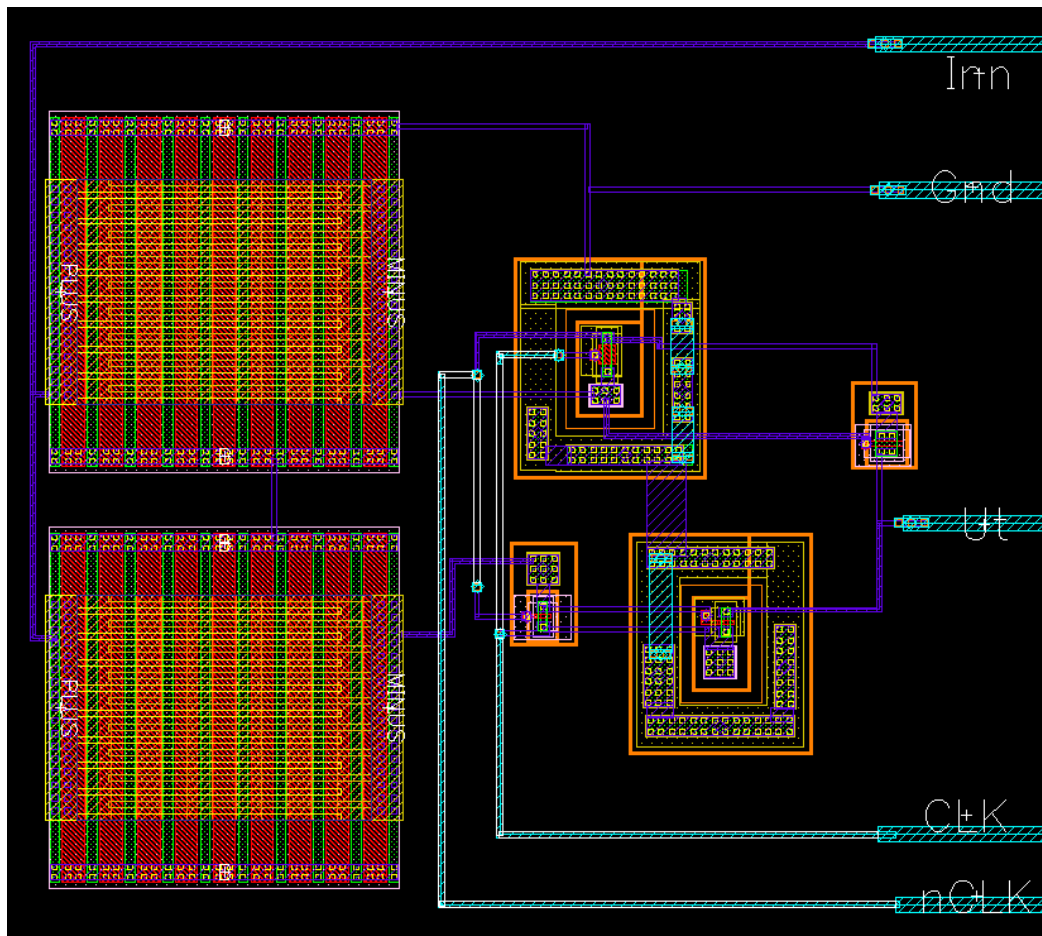


Figur 4.10: LV inverter.

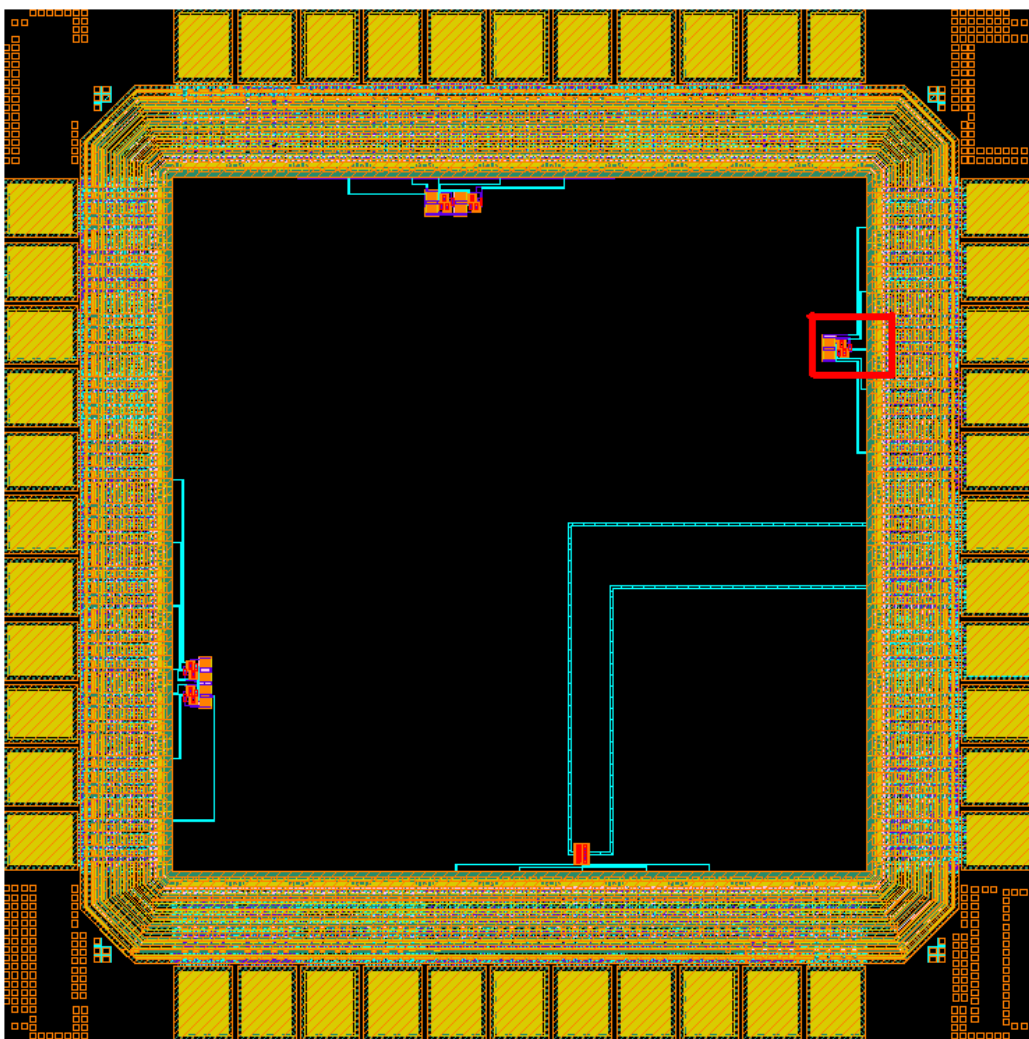
av samme grunn. Jeg har ikke lagt vekt på et effektivt utlegg, både fordi det på en såpass liten krets ikke vil ha så stor betydning for ytelsen og fordi mulighetene er ganske begrenset. I et større system vil selvfølgelig måten utlegget, med routing og plassbesparende transistorformer, lages på ha stor betydning for ytelsen og arealet.

Figur 4.13 på side 21 gir et inntrykk av hvor lite dette er: Hele brikken som vi ser i figuren måler $1.078 \times 1.078 \text{ mm}$ og i det røde rektangelet kan man se selve kretsen min som er vist i figur 4.12 på side 20.

Figur 4.11: Utlegg for 0,35 μ m-kretsen.



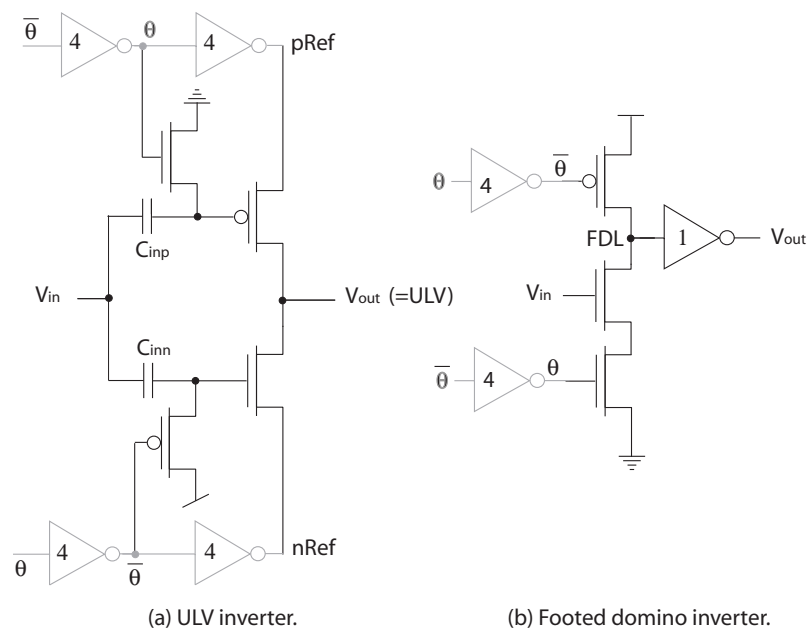
Figur 4.12: Utlegg for 0,13 μ m-kretsen.



Figur 4.13: 44-pinners brikke på $1,162\text{mm}^2$. I den røde ruten kan kretsen i figur 4.12 på forrige side sees.

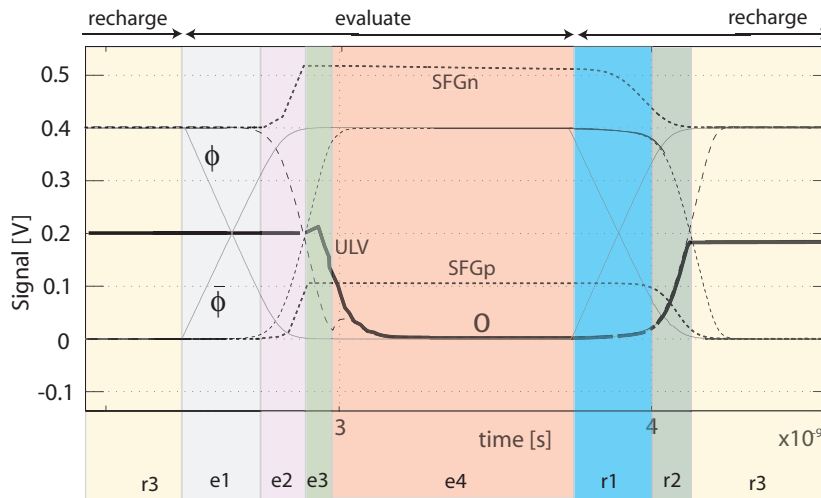
4.5 Hastighet, logisk dybde og forbruk

Når det gjelder kretsens ytelse, har mitt hovedfokus ligget på å verifisere at en slik kretskonstruksjon fungerer i praksis. Jeg har derfor ikke lagt så stor vekt på kretsens faktiske ytelse hverken i simuleringer eller målinger. I midlertid har vi gått inn på en del av disse forholdene i artikkelen “Ultra low voltage CMOS gates” [2].



Figur 4.14: De simulerte portene med klokke drivere

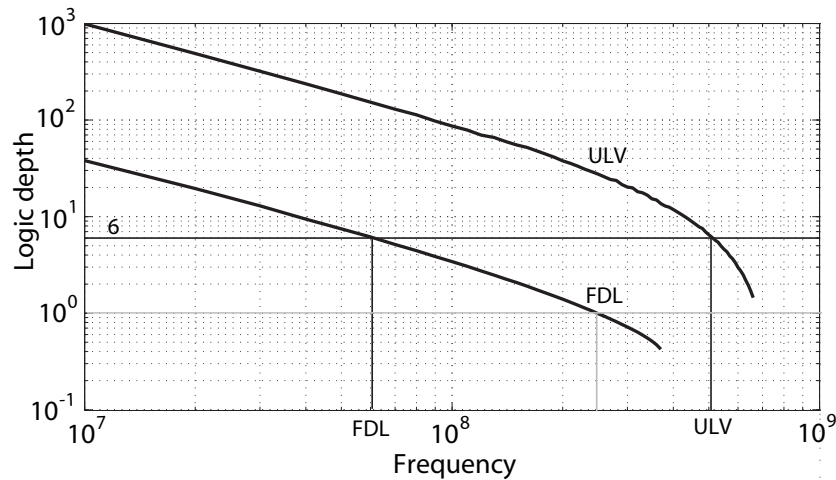
4.5. HASTIGHET, LOGISK DYBDE OG FORBRUK



Figur 4.15: Simulering av *ULV*- og *FDL*-inverterer

Hvis vi antar at vi har en CMOS-prosess hvor $V_t = V_{t_n} = |V_{t_p}| = 0.4V$ og matchede minimumstransistorer i kretsene, kan vi undersøke den potensielle forbedringen til en *ULV*-inverter i forhold til andre typer invertere. Det mest naturlige å sammenligne med er, som tidligere nevnt, såkalt Footed Domino logikk (*FDL*). Figur 4.14 på forrige side viser de to sammenlignede kretsene med klokke drivere som er fire ganger større enn minimum. Lasten brukt i simuleringene er lik lasten til en tilsvarende port, som vil gi en god indikasjon på mulig logisk dybde i et større system. Figur 4.15 viser en simulering av de to kretsene. *FDL*-inverteren har samme, høye inngangssignal i både "positive input transition" og "negative input transition" for å vise den faktiske forskjellen mellom kretsene. Ved et lavt inngangssignal ville utgangen til *FDL*-inverteren bare bli liggende på V_{dd} , mens det i praksis er nedtrekket ved et høyt inngangssignal som viser hvor rask kretsen vil være. I figuren ser man tydelig forsinkelsen som inverterne i klokke driverne skaper, og man ser også tydelig hvor treg *FDL*-inverteren er både ved nedtrekk i evalueringfasen og opptrekk i initialiseringsfasen i forhold til *ULV*-inverteren. Hvor stor forskjellen er kommer jeg tilbake til senere i avsnittet.

I figur 4.16 på neste side vises tilgjengelig logisk dybde for varierende frekvens. Den logiske dybden er ca. 20 ganger større for *ULV*-inverteren enn for *FDL*-inverteren. Hvis man ønsker en logisk dybde på 6, kan *ULV* operere på 500MHz, mens *FDL* bare kan operere på 60MHz. Dataene antyder at *ULV*-inverteren kan operere på ca. 10 ganger hastigheten til



Figur 4.16: Logisk dybde for *FDL* og *ULV*-inverterer.

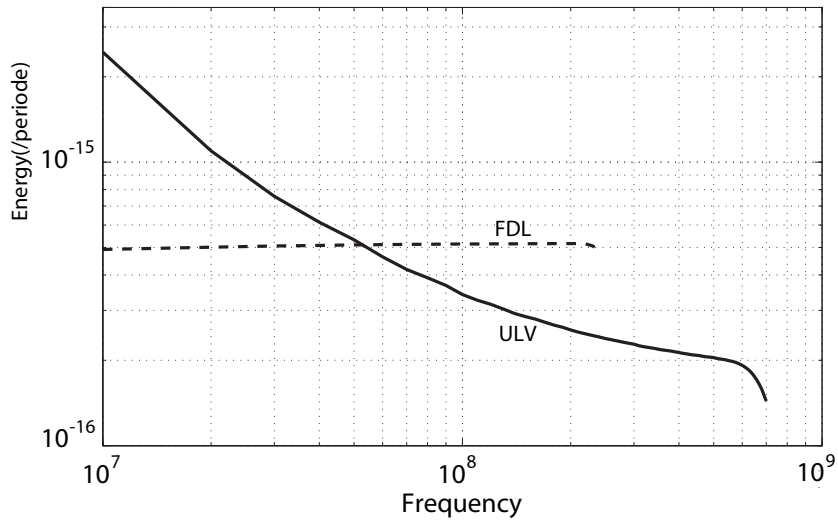
FDL-inverteren.

I figur 4.17 på neste side vises det totale energiforbruket for en periode (initialisering og evaluering) for frekvenser fra 10MHz til 700MHz. *FDL*-porten vil ikke virke riktig ved frekvenser over 250MHz og energiforbruket (med klokke drivere) er uavhengig av frekvensen, mens energiforbruket for *ULV*-porten avtar når frekvensen øker fordi den statiske energien reduseres. I et dynamisk system med høy aktivitet vil ikke den statiske strømmen påvirke det totale effektforbruket i særlig grad. Den statiske strømmen i *ULV*-inverteren etter en transisjon til evalueringsperioden er ca. 1uA, noe som er ganske stor sammenlignet med den statiske strømmen i initialiseringsfasen (2nA). Maksimal frekvens for *ULV* og *FDL*-logikk for varierende forsyningsspenning er vist i figur 4.18 på neste side. Stige- og falltiden for *ULV* og *FDL* vises i figur 4.19 på side 26, og her ser vi at stige-/falltiden til *FDL* er ca 30 ganger større enn for *ULV*.

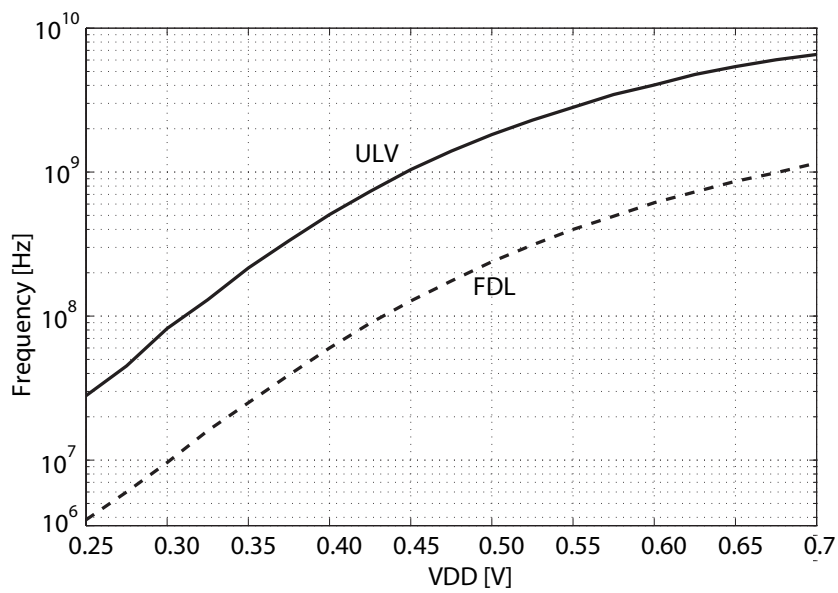
Figur 4.20 på side 26 viser energien over en klokkeperiode for *FDL*- og *ULV*-inverter for varierende forsyningsspenning. Her ser man at det er klokke driverne som dominerer og at den totale energien for *ULV*-inverteren er mindre enn for *FDL*-inverteren på grunn av den store forskjellen i energi for selve inverterne.

Et godt mål for å kunne sammenligne effektiviteten/ytelsen til forskjellige kretser, er å se på produktet av energi og forsinkelse (EDP). Dette gir et bilde av energiforbruket i forhold til hastighet/ytelse og gjør at man

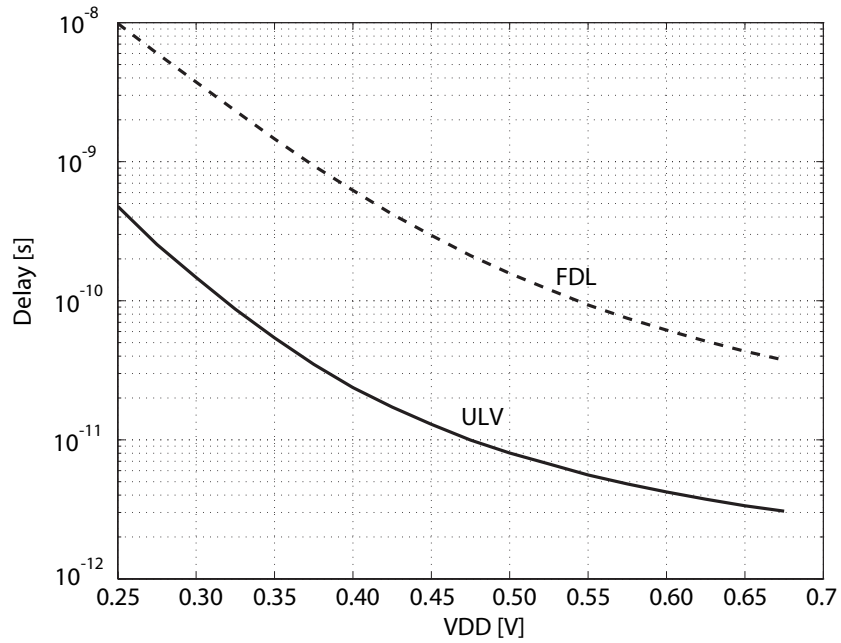
4.5. HASTIGHET, LOGISK DYBDE OG FORBRUK



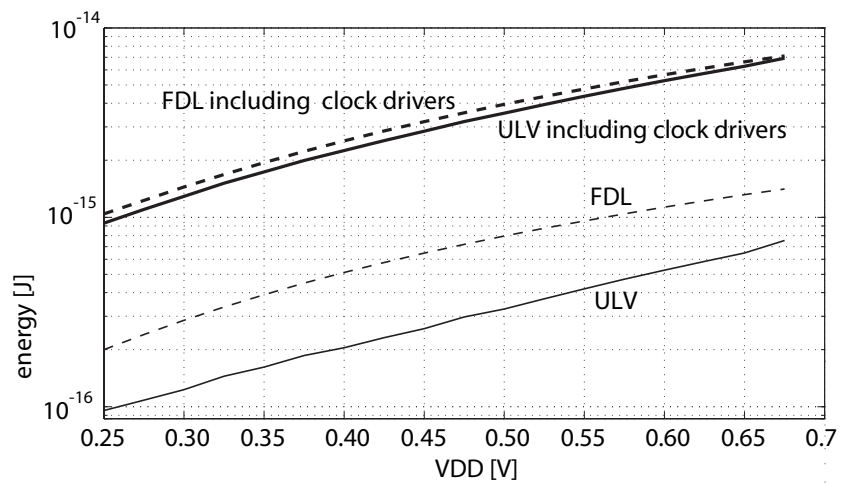
Figur 4.17: Energi over en periode for *FDLog ULV*.



Figur 4.18: Maksfrekvens gitt en dybde på 6 for *FDL* og *ULV*.



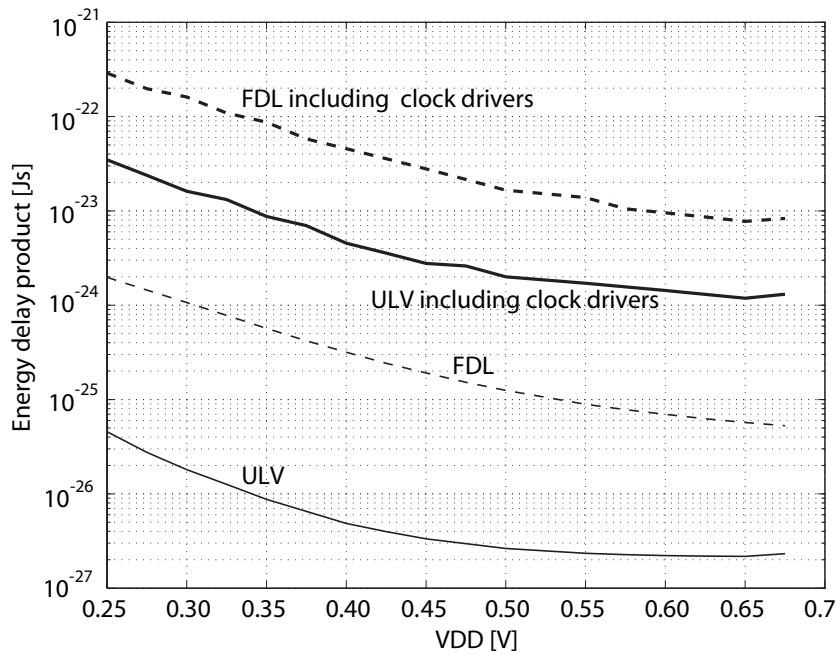
Figur 4.19: Stige- og falltid, 20%-80%, for *FDL* og *ULV*.



Figur 4.20: PDP/energi for *FDL* og *ULV* i en periode.

4.5. HASTIGHET, LOGISK DYBDE OG FORBRUK

kan sammenligne kretser med vidt forskjellige egenskaper. For å redusere EDP må man enten øke ytelsen eller redusere forbruket uten å påvirke hhv. forbruket eller ytelsen. Figur 4.21 viser EDP for de to kretsene med en logisk dybde på 6, og vi ser da at *ULV*-inverteren med klokke drivere er nærmere 10 ganger bedre enn *FDL*-inverteren.



Figur 4.21: EDP for *FDL* og *ULV* i en periode.

KAPITTEL 4. IMPLEMENTASJON

Kapittel 5

Resultater

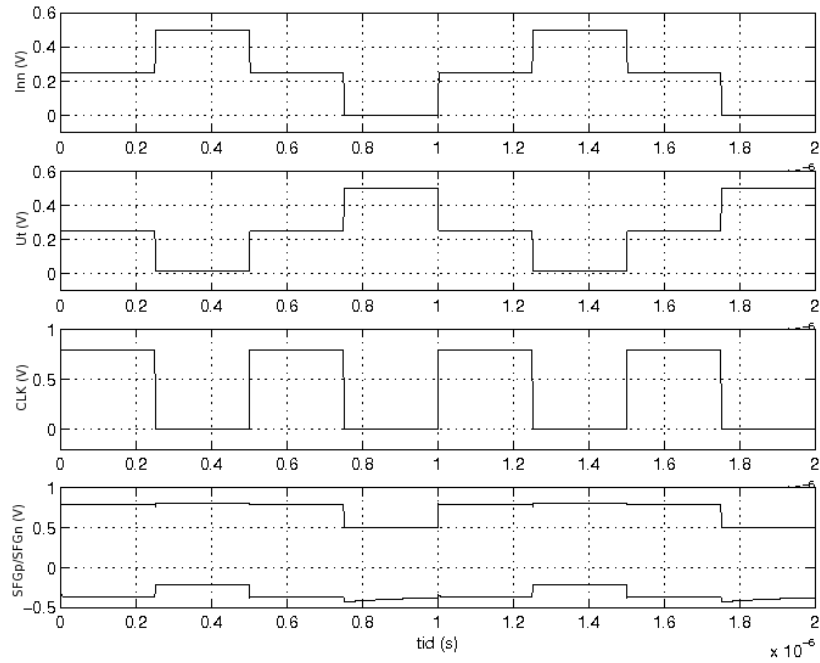
For testing av den designede kretsen har jeg benyttet *Spice*-simulatoren i *Cadence* for simuleringer og tilgjengelig måleutstyr på laboratoriet på Institutt for informatikk, Universitetet i Oslo. Se figur 5.8 på side 36 for en skisse av måleoppsettet og tabell B.1 på side 55 for en oversikt over benyttet utstyr. Kretsene har blitt simulert på 4 og 8MHz fordi dette er omtrent den maksimale hastigheten signalgeneratorene vi har tilgang på kan gi ut med programmerte flernivåsignaler.

Kretsene har blitt designet og simulert med asymmetriske klokkesignaler for å øke ytelsen uten at det går vesentlig ut over forbruket. Asymmetriske signaler vil si at klokkesignalene ikke ligger mellom 0 og V_{dd} , men at det ene klokkesignalet, clk , ligger mellom 0 og et nivå høyere enn V_{dd} og $nclk$ går fra under 0 og til V_{dd} . I simuleringene av $0,35\mu m$ -kretsen med V_{dd} på $0,5V$, har clk gått fra $0V$ til $0,8V$ og $nclk$ gått fra $-360mV$ til $0,5V$. Dette vil gi et ekstra pådrag til kretsen i initialiseringsfasen. Dette gjør at initialiseringen går raskere og ikke blir en så stor flaskehals som den potensielt kunne ha vært.

5.1 Simuleringer

I figur 5.1 på neste side vises en simulering av $0,35\mu m$ -kretsen på $4MHz$ ved $0,5V$ og i figur 5.2 på neste side en simulering av den modifiserte kretsen som jeg har produsert et eksemplar av i *AMS0*, $35\mu m$ -prosessen. Ser man nøye etter på flankene i de to simuleringene ser man at responsen er litt dårligere for den produserte kretsen. Dette kan virke litt påfallende med tanke på den ekstra lasten brønnene er for signalet. Den produserte kretsen vil altså ha en lavere teoretisk maksfrekvens enn det

KAPITTEL 5. RESULTATER



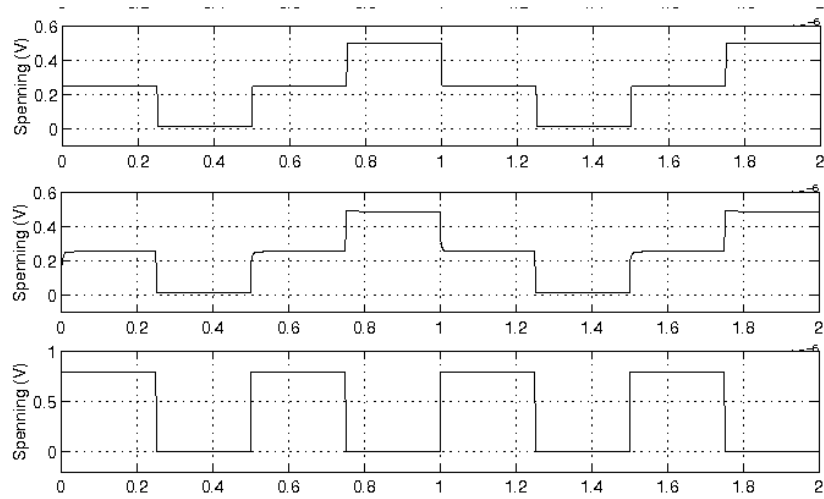
Figur 5.1: Simulering ved 0.5V, 4MHz



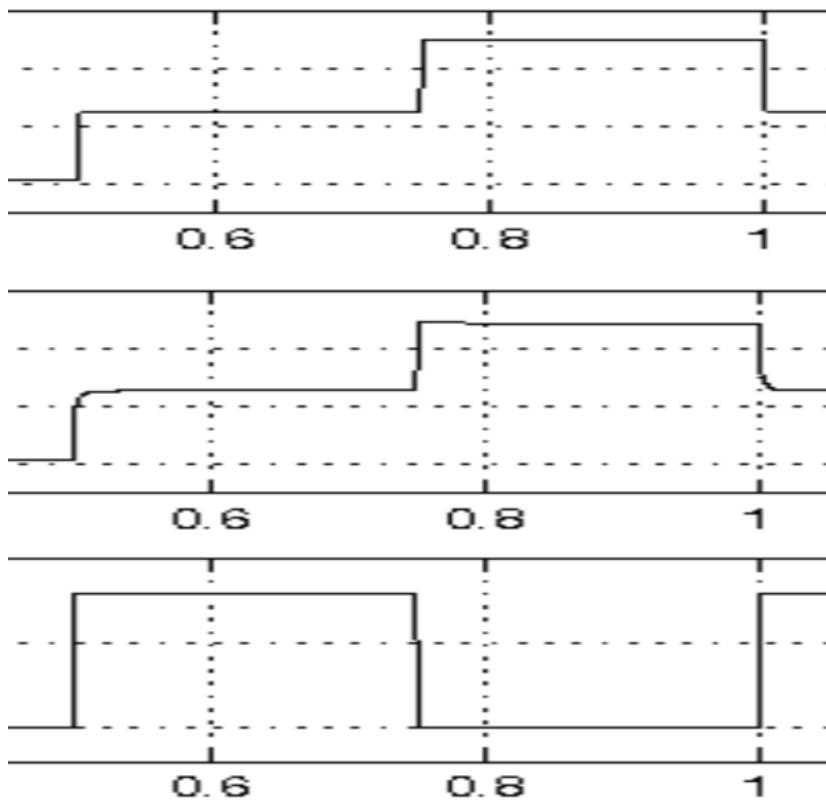
Figur 5.2: Simulering ved 0.5V, 4MHz m/fast brønn, 0,35 μ m

opprinnelige designet. Figur 5.3 på neste side viser simulering av de to variantene av kretsen, og figur 5.4 på side 33 gjengir et mindre utsnitt av denne simuleringen for å vise forskjellene tydeligere. Hvis man ser litt nærmere på dette, er det egentlig litt merkelig at jeg har fått de simuleringresultatene jeg har gjort. Teoretisk sett skulle det, ved å koble brønnene til gnd og V_{dd} i stedet for til klokkene og SFG -nodene, bli mindre last for inngangen og klokkene og dermed øke hastigheten til kretsen, men simuleringene viser at det ikke har noen innvirkning på responsen i evalueringsfasen og at initialiseringen faktisk går fortere med brønnene koblet til hhv. SFG_p og SFG_n . Dette kommer også tydelig fram i simuleringene med flernivåsignaler, figur 5.5 på side 34 og figur 5.6 på side 34.

Simuleringene viser at initialiseringen av kretsen blir raskere med brønnene på initialiseringstransistorene koblet til hhv. SFG_p og SFG_n . V_{SFG_p} er jo hele tiden lavere enn $gnd/0V$ slik at V_{gs_n} faktisk blir høyere enn om brønnen hadde vært koblet til gnd . På samme måte er V_{SFG_n} hele tiden høyere enn V_{dd} og dette gjør at V_{gs_p} blir høyere enn om brønnen på p-transistoren hadde vært koblet til V_{dd} . Samtidig, med de asymmetriske klokkesignalene, er også brønnene på transistorene i selve inverterledet koblet til et relativt sett høyere potensiale enn om de hadde vært koblet til hhv. gnd og V_{dd} , og dette er med på å gjøre initialiseringen raskere.

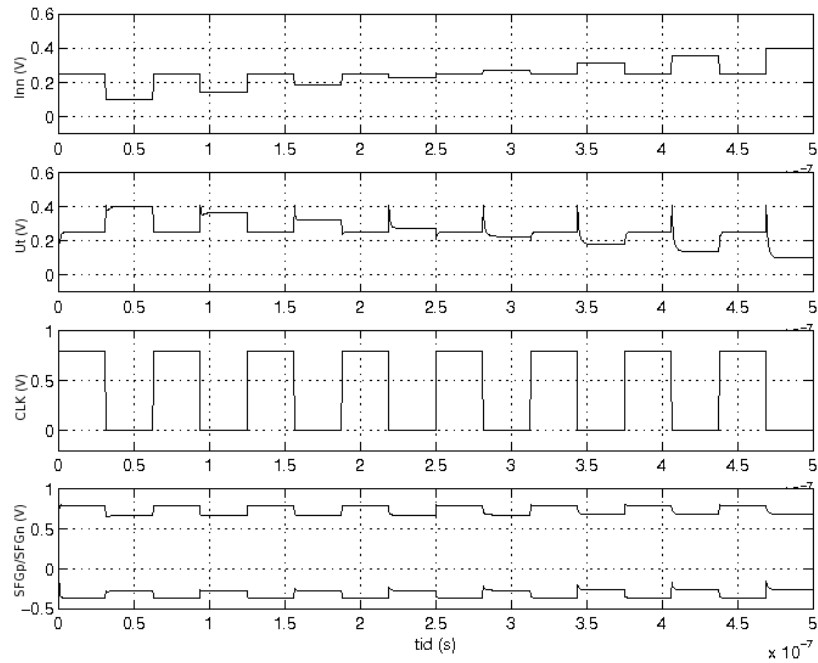


Figur 5.3: Ut 1 viser responsen til inverteren med brønnene koblet til hhv. klokkene og SFG -nodene, mens Ut 2 viser responsen med brønn koblet til hhv. gnd og V_{dd} .

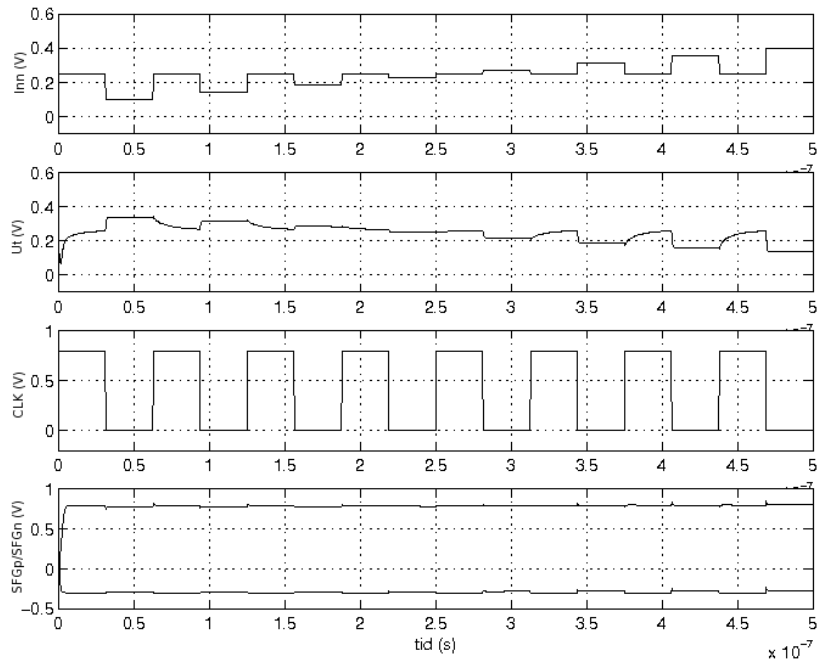


Figur 5.4: Utsnitt av figur 5.3 på forrige side

KAPITTEL 5. RESULTATER

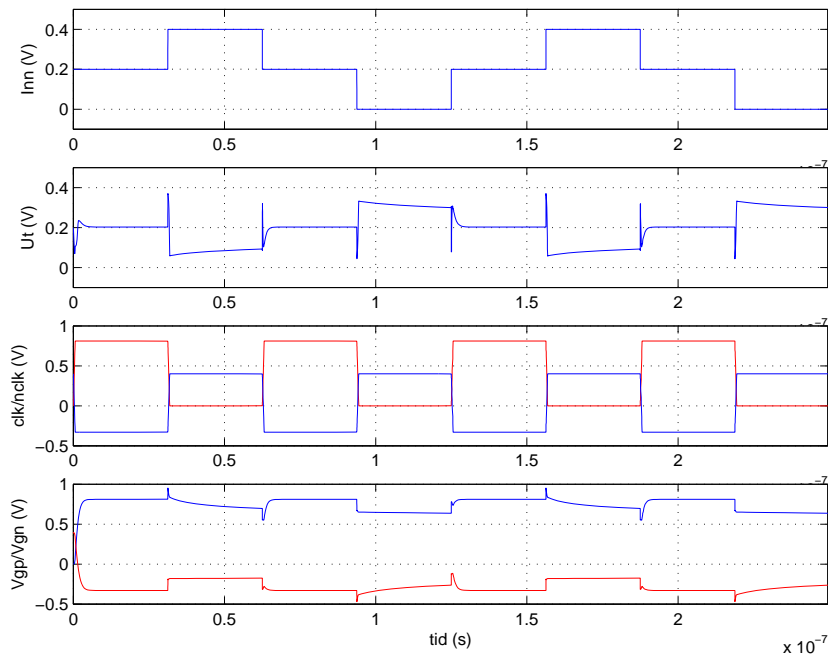


Figur 5.5: Simulering ved 0.5V, 8MHz, 0,35 μ m



Figur 5.6: Simulering ved 0.5V, 8MHz m/fast brønn på 0,35 μ m-brikke

Når det gjelder simuleringer på kretsen i $0.13\mu\text{m}$ -prosessen har vi ikke stolt helt på resultatene til spice-simulatoren i *Cadence* ettersom vi hadde flere problemer med designkitet til STM. Vi satset derfor på at kretsen ville virke bedre i virkeligheten og produserte den til tross for at simuleringene ga dårlige karakteristikker. Figur 5.7 viser simulering av $0.13\mu\text{m}$ -kretsen på 8MHz . Logisk sett gjør kretsen det den skal, men karakteristikken er slett ikke god.



Figur 5.7: Simulering ved 0.4V , 8MHz på $0.13\mu\text{m}$ -brikke

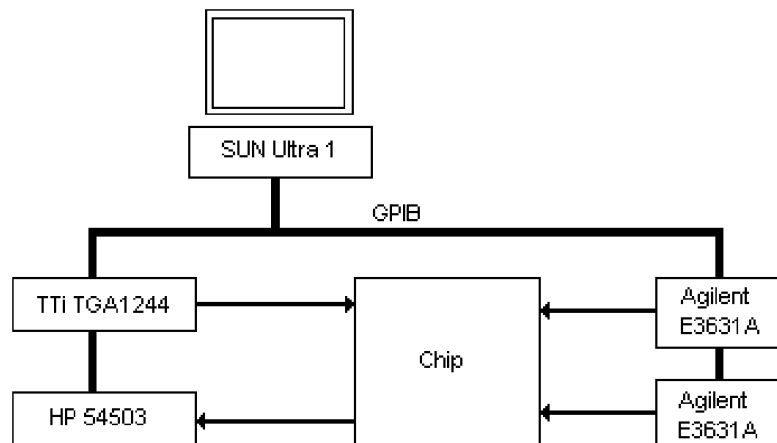
5.2 Målinger

Måleoppsettet jeg har brukt er som vist i figur 5.8 på neste side, og i vedlegg A.3 på side 55 er programkoden som er brukt for generering av signaler og uthenting av data fra måleinstrumentene. Nærmere beskrivelse av instrumentene finnes i vedlegg B på side 55. Av praktiske årsaker har jeg kjørt målingene på enda lavere frekvenser enn tidligere forespeilet. Selv om signalgeneratoren kan lage slike signaler i det lave MHz -området, har det vist seg at det er lettere å få signalene til å bli riktige i kHz -området. Dette skyldes kanskje en kombinasjon av instrumentets oppløsning og måten det har blitt programmert på. Det viktigste

KAPITTEL 5. RESULTATER

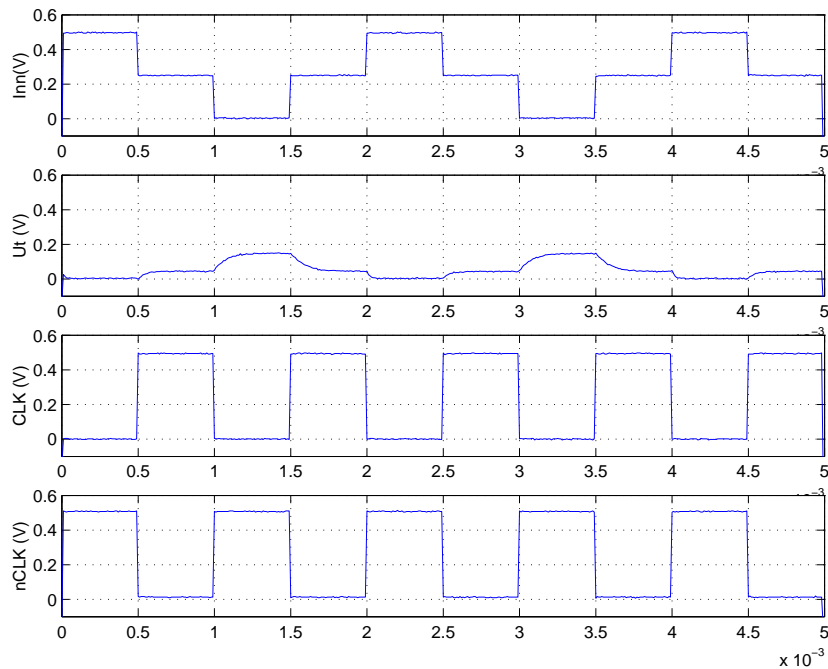
for meg er uansett å vise at logikken fungerer. Målingene er enkle og det er derfor en del støy fra måleutstyret, særlig på $0,13\mu m$ -kretsen.

Signalgeneratorene ga ingen mulighet til å lage de asymmetriske klokkesignalene slik jeg hadde planlagt, så jeg ble nødt til å bruke vanlige klokkesignaler mellom gnd og V_{dd} . Dette ville, som nevnt tidligere, gjøre initialiseringen av kretsen tregere, men utover dette ville det ikke påvirke kretsens funksjonalitet. Når det gjaldt $0,13\mu m$ -kretsen øynet jeg en mulighet til å omgå dette problemet, siden alle brønnene er isolert og inngangskondensatorene har et eget gnd -plan. Ettersom ingen andre deler av kretsen var koblet til gnd og jeg benyttet analoge padder uten beskyttelse, kunne jeg sette opp spenningen på gnd -planet til $360mV$ og, med $V_{dd} = 0,5V$, programmere $nclk$ mellom 0 og $860mV$ og clk mellom $360mV$ og $1,16V$ og la inngangssignalet ligge mellom $360mV$ og $860mV$. Målingene jeg har gjort er likevel kun med vanlige symmetriske klokkesignaler mellom $0V$ og V_{dd} .



Figur 5.8: Måleoppsett

Den eneste “fornuftige” målingen jeg har fått på $0.35\mu m$ -kretsen er vist i figur 5.9 på neste side. Her er nivåene langt fra riktige, og det kan se ut som utgangssignalet rett og slett er dempet. Dette kan blant annet skyldes for høy last i padden eller i deler av måleutstyret. Rent logisk ser det ut til at kretsen fungerer, men uten videre undersøkelser er det vanskelig å si om padden begrenser signalet nedad fordi det egentlig har en negativ verdi eller om det er andre grunner til nivåfeilen.

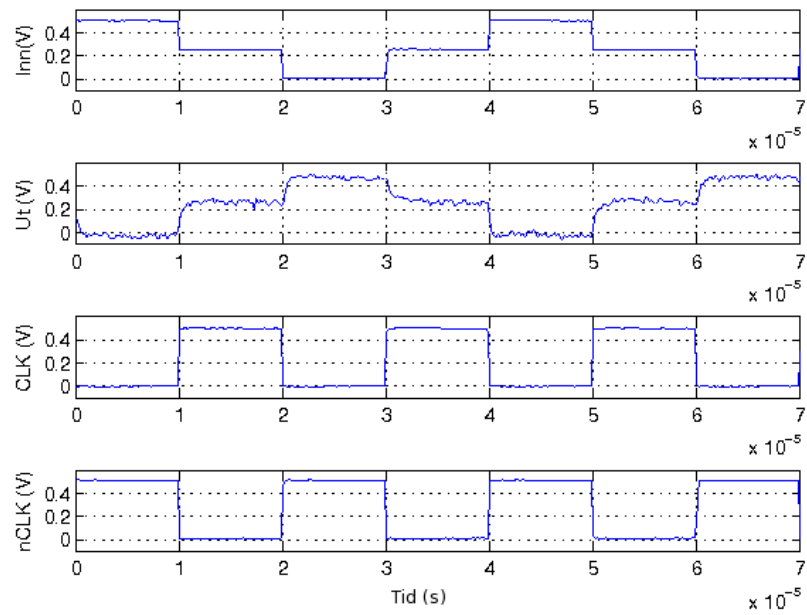


Figur 5.9: Måling ved $0.5V$, $1kHz$ på $0.35\mu m$ -krets

$0.13\mu m$ -kretsen har jeg dimensjonert med utgangspunkt i $0.4V$ forsyningsspenning. Målingene jeg har gjort starter med forsyningsspenning på $0.5V$ og så går jeg nedover for å se hvor lav spenning kretsen kan operere på. Dette betyr at i de første målingene vil p-transistorene være litt sterkere enn n-transistorene for så å bli svakere og svakere jo lavere forsyningsspenningen blir.

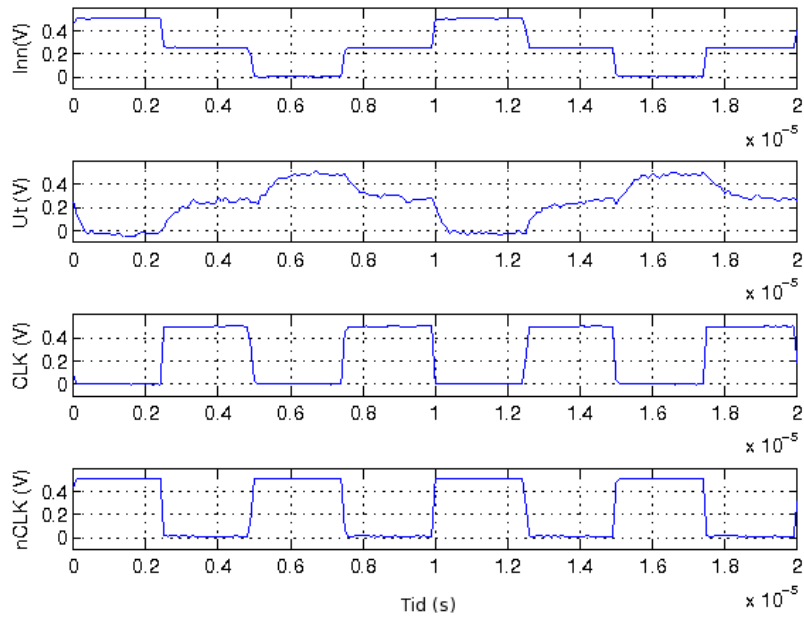
Figur 5.10 på neste side viser en simulering ved $50kHz$ og $V_{dd} = 0,5V$. Det kan se ut som om initialiseringsnivået er noe høyere enn $V_{dd}/2$, og dette kan skyldes at n-siden har blitt redusert i forhold til p-siden pga. manglende asymmetriske klokkesignaler. Under dimensjoneringsprosessen har n-brønnene blitt foret med $-360mV$, mens p-brønnene har fått $800mV$. Altså har n-siden fått mer pådrag enn p-siden for å rette opp en skjevhet som nok egentlig burde vært gjort i dimensjoneringen av transistorene. Dette gir nå en litt svak n-side, selv om jeg tidligere har skrevet at dimensjoneringen ikke har så stor betydning for nivået på utgangen i initialiseringsperioden. Dette går igjen i alle målingene på denne kretsen. I tillegg kan det virke som om p-siden, allerede ved $0,5V_{dd}$, er litt svak. Dette kan skyldes variasjoner i produksjonsprosessen.

KAPITTEL 5. RESULTATER



Figur 5.10: Måling ved 0.5V, 50kHz på 0.13 μ m-krets

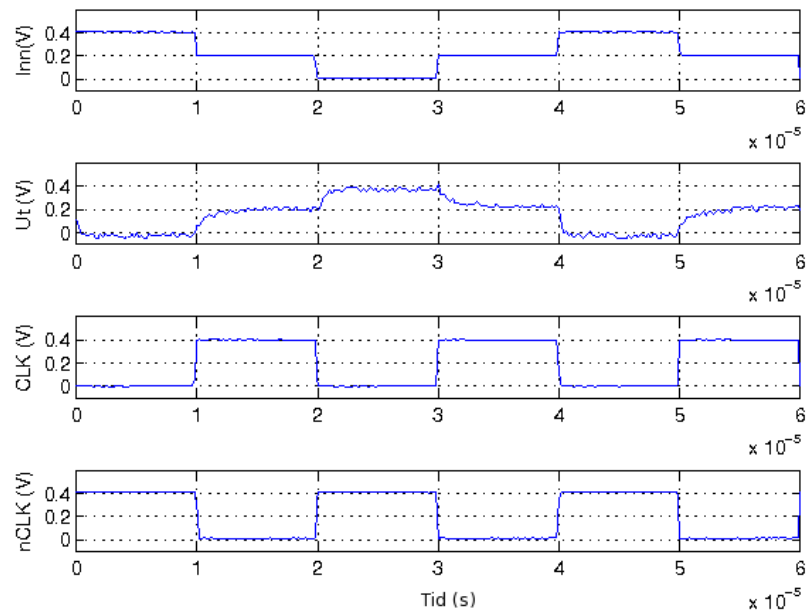
I figur 5.11 har jeg økt hastigheten til $200kHz$ og da ser man tydelig begrensningene i hvor fort denne kretsen kan kjøres på $0,5V_{dd}$. Maksimal hastighet anslår jeg til å være rundt $500kHz$.



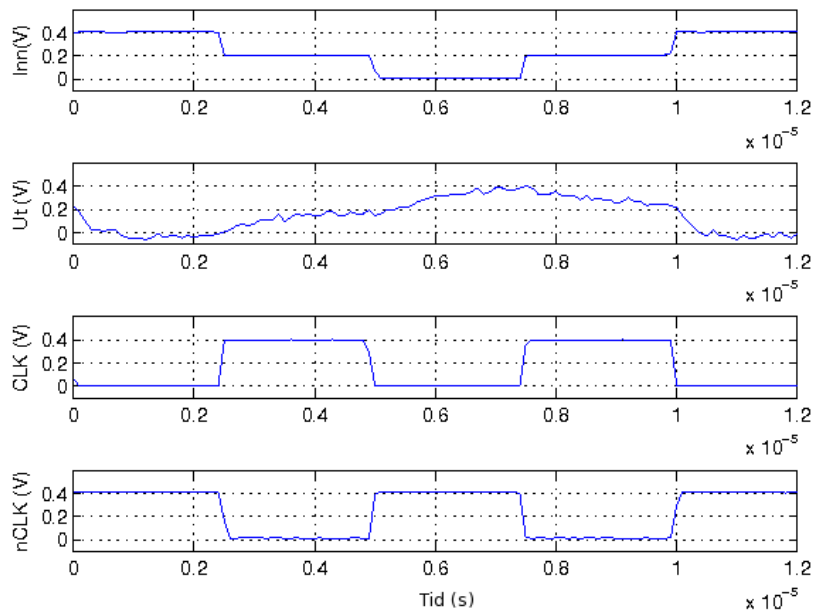
Figur 5.11: Måling ved $0,5V$, $200kHz$ på $0,13\mu m$ -krets

Når jeg reduserer forsyningsspenningen til $0,4V$ (figur 5.12 på neste side og 5.13 på side 41), ser man de samme tendensene med litt høyt initialiseringsnivå og ikke sterkt nok opptrekk i evalueringsfasen.

KAPITTEL 5. RESULTATER

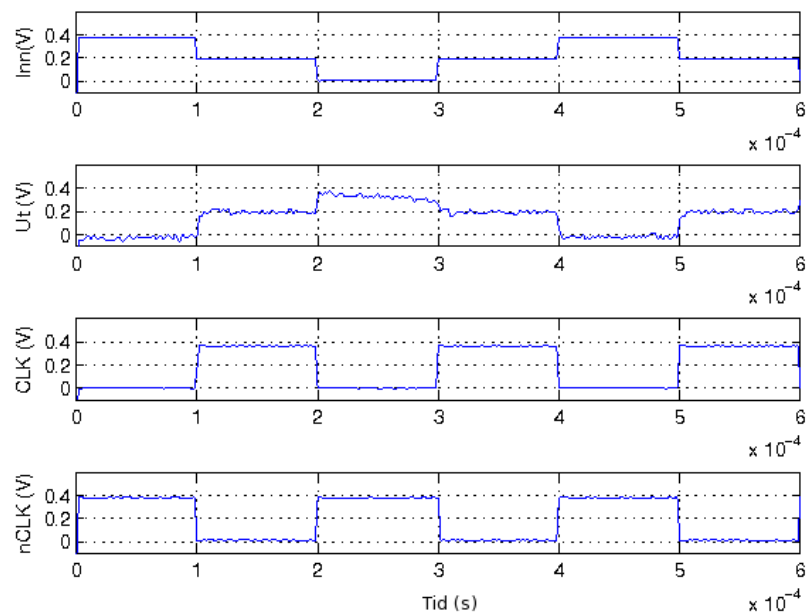


Figur 5.12: Måling ved 0.4V, 50kHz på 0.13 μ m-krets

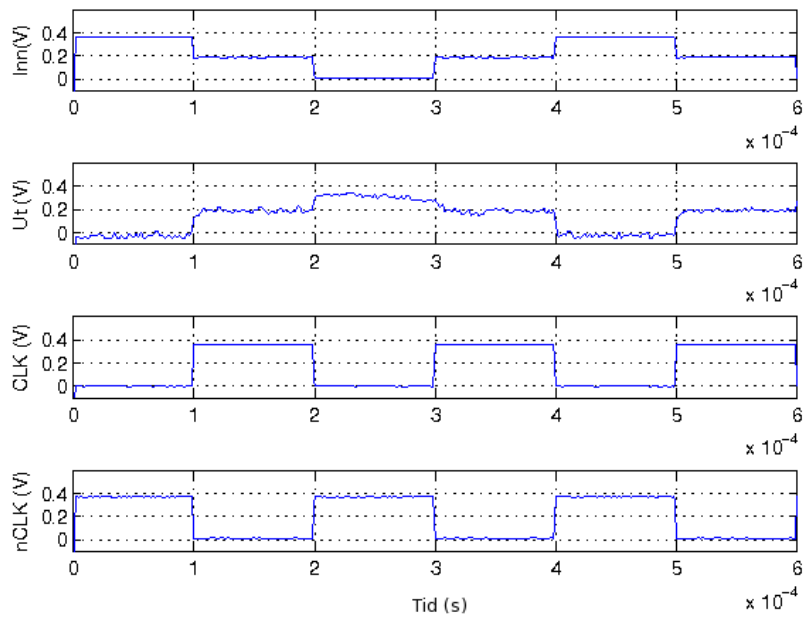
Figur 5.13: Måling ved 0.4V, 200kHz på 0.13 μ m-krets

KAPITTEL 5. RESULTATER

Senkes forsyningsspenningen ytterligere, må man enda lavere ned i frekvens for at kretsen skal virke. I figur 5.14 og 5.15 på neste side vises målinger på $5kHz$ ved hhv. $0,37V_{dd}$ og $0,36V_{dd}$. Her ser man tydelig at initialiseringsnivået ligger altfor høyt. Man må være klar over at hastighetene som kretsen opererer på er veldig lave fordi kretsen driver en stor ekstern last ubuffret. Internt på brikken vil man kunne oppnå hastigheter som forespeilet i 4.5 på side 22.



Figur 5.14: Måling ved $370mV$, $5kHz$ på $0.13\mu m$ -krets

Figur 5.15: Måling ved 360mV, 5kHz på 0.13 μ m-krets

KAPITTEL 5. RESULTATER

Kapittel 6

Konklusjon og videre arbeid

I denne oppgaven har en *ULV SFG*-inverter blitt presentert. To testkretser har blitt produsert og målt, og gjennom simuleringer har kretsen blitt sammenlignet med en *FDL*-inverter.

6.1 Resultater

Resultatene jeg har oppnådd i arbeidet med denne hovedoppgaven er på mange måter tilfredsstillende, og det har blitt vist at denne typen logikk har mange fordeler sammenlignet med *FDL*, som det er naturlig å sammenligne med på grunn av likhetstrekkene i design. Blant annet ser man en stor forbedring i hastighetspotensialet, og mulig logisk dybde er også sterkt forbedret. Det er dessuten verdt å merke seg at på grunn av at *ULV*-inverteren i hovedsak er dynamisk, vil effektforbruket reduseres når frekvensen øker. Den har noen statiske komponenter også, men disse vil ikke påvirke effektforbruket i særlig grad når aktiviteten i kretsen er høy. *FDL*-inverteren er til sammenligning mer eller mindre helt statisk og forbruket vil være konstant uavhengig av aktivitetsnivået.

Målingene jeg har gjort viser at kretsen er følsom i forhold til dimensjonering og forsyningsspenning. Forsterkningsforholdet mellom p- og n-transistorene endrer seg slik at hvis man skal redusere forsyningsspenningen, må man gjøre p-transistorene større for at de skal klare å trekke utgangen skikkelig opp. Dette gjelder selvfølgelig ikke for denne kretsen spesielt, men for alle kretser, og det vil også være variasjoner på dette mellom forskjellige prosesser.

Siden utgangen på kretsen ikke er buffret må kretsen drive pad-en og måleutstyret, og dette fører til at jeg har fått lite imponerende måler-

esultater rent ytelsesmessig, men rent funksjonsmessig er resultatene for $0,13\mu\text{m}$ -kretsen meget gode. Nivåene er noe lave, mest sannsynlig på grunn av litt svak p-side i utgangstrinnet, mens initialiseringsnivået er litt høyt. For $0,35\mu\text{m}$ -kretsen er måleresultatene dårlige. Nivåene er alt for lave, og siden kretsen ikke er buffret, kan det kanskje ha noe med demping av signalet i pad-en å gjøre og at et mulig negativt utgangsnivå ikke slipper igjennom pad-en.

Det som er litt overraskende er at spice-simuleringene på $0,35\mu\text{m}$ -kretsen er fine, mens den ikke fungerer riktig ved måling, og for $0,13\mu\text{m}$ -kretsen er det akkurat omvendt. For de dårlige simuleringene på $0,13\mu\text{m}$ -kretsen kan en mulig forklaring være at designkittet fra STM fungerte dårlig med designprogramvaren Cadence som ble benyttet.

Løsningen jeg har valgt med tilkobling av brønnene for initialiserings-transistorene mot de avkoblede inngangssignalene (*SFG*-nodene) ble valgt fordi dette ga den beste utgangsresponsen ved simuleringer sammenlignet med å koble brønnene mot hhv. V_{dd} og *gnd*. Simuleringene viste en forbedring av responsen i initialiseringsfasen og ingen påvirkning i evalueringsfasen, til tross for økt kapasitans i størrelsesorden et par fF for en minimumstransistor i $0,13\mu\text{m}$ -prosessen. Ettersom initialiseringsfasen er den potensielle flaskehalsen, vil faktisk løsningen jeg har valgt øke ytelsen.

6.2 Videre arbeid

Denne oppgaven har gitt mange svar, men har også reist en del spørsmål. En naturlig fortsettelese av dette arbeidet vil for eksempel kunne være å sammenligne med andre typer logikk, og å undersøke hvor lave spenninger kretsen kan operere på. Med riktig dimensjonering vil man nok kunne komme betydelig ned i forsyningsspenning. Det har blitt gjort målinger som viser inverterende karakteristikk helt nede på 100mV forsyningsspenning. Dette er presentert i [1].

En nærmere studie av hvordan kretsen oppfører seg med forskjellige varianter av tilkobling til brønnene kan også være interessant. Hva med å la brønnene til initialiseringstransistorene flyte, eller koble dem til klokkesignalene? Det siste alternativet vil, hvis man benytter asymmetriske klokker som foreslått, kanskje beholde fordelene med ekstra pådrag i initialiseringsfasen i forhold til om brønnene er koblet til hhv. V_{dd} og

gnd, samtidig som den ekstra lasten fjernes fra inngangssignalet slik at responsen i evalueringsfasen kan bli raskere.

Det vil selvfølgelig også være interessant å benytte denne konstruksjonsteknikken i større systemer som foreslått i [2] og sammenligne med samme system i et tradisjonelt design.

KAPITTEL 6. KONKLUSJON OG VIDERE ARBEID

Tillegg A

Programkode

A.1 bin.m

Kildekode for brukergrensesnitt, generering av inngangssignaler og lagring av målinger:

```
clear;
HP54503=17;
HP3631A=7;
HP3631B=8;
TTi1244A=6;
TTi1244B=5;

%-----%
% Innganger på kretsen %
%-----%
%div = input('Div:\n');
div=100;
f=10e6;
vdd = .5;
offs = vdd/2;
pv=4094/vdd;

clk_topp = .5; topp_clk = clk_topp*pv-2047;
clk_bunn = .0; bunn_clk = clk_bunn*pv-2047;
nclk_topp = .5; topp_nclk = nclk_topp*pv-2047;
nclk_bunn = 0; bunn_nclk = nclk_bunn*pv-2047;

topp_sig = nclk_topp*pv-2047;
bunn_sig = clk_bunn*pv-2047;
recharge = bunn_sig+(topp_sig-bunn_sig)/2;

%null = .0;
null = clk_bunn;

sig = [bunn_sig recharge topp_sig recharge];
CLK = [];
nCLK = [];

%plorange/offsets
xmax=6*div/f;
ymin = nclk_bunn-.1;
ymax = clk_topp+.2;
inn = -.4;
ut = -.4;
clk = -.8;
nclk = -.8;

%HP3631_Init(HP3631A);
HP3631_Init(HP3631B);
HP3631_SetILimit(1, 5e-3, HP3631A);
HP3631_SetILimit(2, 5e-3, HP3631A);
```

TILLEGG A. PROGRAMKODE

```
HP3631_SetVolt(1, vdd, HP3631A); %+6V
%HP3631_SetVolt(2, null, HP3631A); %+25V
HP3631_SetILimit(1, 5e-3, HP3631B);
HP3631_SetILimit(2, 5e-3, HP3631B);
HP3631_SetVolt(1, null, HP3631B); %+6V
HP3631_SetVolt(2, vdd/2, HP3631B); %+25V

%genererer klokkesignaler
%for j=1:2,
    for i=1:div,
        nCLK = [nCLK topp_nclk];
        CLK = [CLK bunn_clk];
    end
    for i=1:div,
        nCLK = [nCLK bunn_nclk];
        CLK = [CLK topp_clk];
    end
%end

%genererer inngangssignal
%Inn1 = vdd/2;
Inn1 = [];
for j=1:4,
    for i=1:div,
        Inn1 = [Inn1 sig(j)];
    end
end

%-----%
% Programmering av signalgeneratorer %
%-----%
arb3ch(TTi1244A,TTi1244B,Inn1,CLK,nCLK,f,vdd,offs);

%-----%
% Konfigurering av scope %
%-----%
 GPIB_Write('EOI ON', HP54503);
 GPIB_Write('BNC PROBE', HP54503);
for n=1:4,
    nn = int2str(n);
    tmp = strcat('CHANNEL', nn);
    GPIB_Write(strcat(tmp, ':OFFSET 0'), HP54503);
    GPIB_Write(strcat(tmp, ':RANGE .5'), HP54503);
    GPIB_Write(strcat(tmp, ':COUPLING DC'), HP54503);
end
%GPIB_Write('CHANNEL2:PROBE 10', HP54503);
GPIB_Write('AUTOSCALE', HP54503);
GPIB_Write('TIMEBASE:REFERENCE LEFT', HP54503);
GPIB_Write('DISPLAY:CONNECT ON', HP54503);

%% Dump oscilloscope values
d = input('Dump Scope(y/n)? ', 's');
pause(2);
%d = 'n';
if(d=='y');
    d = 'd';
    n = 1;
    while(d=='d');
        [x1,x2,x3,x4,y1,y2,y3,y4] = DumpScope(HP54503);
        figure(n);
        hold off;
        clf;
        subplot(4,1,1),plot(x1,y1+inn);axis([0 xmax ymin ymax]);grid on;ylabel('Inn(V)');
        subplot(4,1,2),plot(x2,y2+ut);axis([0 xmax ymin ymax]);grid on;ylabel('Ut (V)');
        subplot(4,1,3),plot(x3,y3+clk);axis([0 xmax ymin ymax]);grid on;ylabel('CLK (V)');
        subplot(4,1,4),plot(x4,y4+nclk);axis([0 xmax ymin ymax]);grid on;ylabel('nCLK (V)');

        d = input('dump(d)/lagre(l)? ', 's');

        if(d=='l');
            fname = strcat(strcat('/hom/perano/fag/oppgabe/figurer/013/mål/',input('lagre som: ', 's')),'.eps');
            if(length(fname)>=1);
                print('-depsc',fname)
            end
        end
        if(d=='d');
            d='d';
            n=n+1;
        end
    end
end
end
```

A.2 arb3ch.m

Kildekode for programmering av signalgenerator:

```
%/ifi/fenris/p23/mes/src/matlab/gpib/linux
function arb3ch(TTi1244A,TTi1244B,Inn1,CLK,nCLK,f,vdd,offs);

GPiB_Write('*CLS;*RST;',TTi1244A)
%GPiB_Write('*CLS;*RST;',TTi1244B)

GPiB_Write('SETUPCH 1',TTi1244A);
GPiB_Write('SYNCOU ON', TTi1244A);

TTi1244_LockStatus('OFF');
TTi1244_SetChannel(1);
TTi1244_ArbitraryChannelDelete('Inn1');
TTi1244_ArbitraryBackDelete('Inn1');
TTi1244_ArbitraryDef('Inn1',length(Inn1), Inn1);
TTi1244_ArbitrarySetOutput('Inn1');
TTi1244_SetTerm('OPEN');
TTi1244_LockMode('MASTER');
TTi1244_ArbitraryFrequency(f);
TTi1244_SetAmplitude(vdd);
TTi1244_DCOffset(offs);
TTi1244_ChannelEnable('ON', 1);

TTi1244_SetChannel(2);
TTi1244_ArbitraryChannelDelete('CLK');
TTi1244_ArbitraryBackDelete('CLK');
TTi1244_ArbitraryDef('CLK', length(CLK), CLK);
TTi1244_ArbitrarySetOutput('CLK');
TTi1244_SetTerm('OPEN');
TTi1244_LockMode('SLAVE');
TTi1244_ArbitraryFrequency(f);
TTi1244_SetAmplitude(vdd);
TTi1244_DCOffset(offs);
TTi1244_ChannelEnable('ON', 2);

TTi1244_SetChannel(3);
TTi1244_ArbitraryChannelDelete('nCLK');
TTi1244_ArbitraryBackDelete('nCLK');
TTi1244_ArbitraryDef('nCLK', length(nCLK), nCLK);
TTi1244_ArbitrarySetOutput('nCLK');
TTi1244_SetTerm('OPEN');
TTi1244_LockMode('SLAVE');
TTi1244_ArbitraryFrequency(f);
TTi1244_SetAmplitude(vdd);
TTi1244_DCOffset(offs);
TTi1244_ChannelEnable('ON', 3);

TTi1244_LockStatus('OFF');
%TTi1244_LockStatus('OFF',TTi1244B);

GPiB_Write('REFCLK MASTER', TTi1244A);
%GPiB_Write('REFCLK SLAVE', TTi1244B);
TTi1244_LockStatus('ON');
%TTi1244_LockStatus('ON',TTi1244B);
```

A.3 DumpScope.m

Kildekode for dumping av verdier fra oscilloskopet:

```
function [X1,X2,X3,X4,Y1,Y2,Y3,Y4]= DumpScope(dev)

HP54503 = dev;

format = [];
type = [];
nopoints = [];
count = [];
xinc = [];
xorg = [];
xref = [];
yinc = [];
yorg = [];
yref = [];
```

TILLEGG A. PROGRAMKODE

```
channelY = [];  
channelX = [];  
  
for channelNo = 1 : 4  
    cmdstr = sprintf('WAVEFORM:SOURCE CHANNEL%d', channelNo);  
    GPIB_Write(cmdstr, HP54503);  
    GPIB_Write('ACQUIRE:COMPLETE 30', HP54503);  
    % cmdstr = sprintf('DIGITIZE CHANNEL%d', channelNo);  
    % GPIB_Write(cmdstr, HP54503);  
  
    GPIB_Write('SYSTEM:HEADER OFF', HP54503);  
    GPIB_Write('WAVEFORM:PREAMBLE?', HP54503);  
    data = GPIB_Read(HP54503);  
  
    % vectorize comma-separated data in string.  
    % data - input string  
    % tmp - vectorized data  
  
    tmp = [0:9];  
    k = 1;  
    j = [];  
    for i = 1 : length(data)  
        if data(i) ~= ','  
            j = [j data(i)];  
        end  
        if data(i) == ','  
            tmp(k) = str2num(j);  
            k = k + 1;  
            j = [];  
        end  
    end  
    %tmp(k) = str2num(j);  
  
    format(channelNo) = tmp(1);  
    type(channelNo) = tmp(2);  
    nopoints(channelNo) = tmp(3);  
    count(channelNo) = tmp(4);  
    xinc(channelNo) = tmp(5);  
    xorg(channelNo) = tmp(6);  
    xref(channelNo) = tmp(7);  
    yinc(channelNo) = tmp(8);  
    yorg(channelNo) = tmp(9);  
    yref(channelNo) = tmp(10);  
  
    GPIB_Write('WAVEFORM:FORMAT ASCII', HP54503);  
    GPIB_Write('WAVEFORM:DATA?', HP54503);  
    b = GPIB_Read(HP54503);  
  
    array = [1:500];  
    k = 1;  
    j = [];  
    for i = 1 : length(b)  
        %b(i)  
        if b(i) ~= ','  
            j = [j b(i)];  
        end  
        if b(i) == ','  
            array(k) = str2num(j);  
            k = k + 1;  
            j = [];  
        end  
    end  
    %plot(array)  
  
    y = array;  
  
    for i = 1 : length(y)  
        y(i) = (y(i) - yref(channelNo))*yinc(channelNo) + yorg(channelNo);  
    end  
  
    channelY(channelNo, :) = y;  
  
    x = [0:499];  
    for i = 1 : length(y)  
        x(i) = (x(i) - xref(channelNo))*xinc(channelNo) + xorg(channelNo);  
    end
```



```
channelX(channelno, :) = x;  
end  
X1 = channelX(1, :);  
X2 = channelX(2, :);  
X3 = channelX(3, :);  
X4 = channelX(4, :);  
  
Y1 = channelY(1, :);  
Y2 = channelY(2, :);  
Y3 = channelY(3, :);  
Y4 = channelY(4, :);
```

TILLEGG A. PROGRAMKODE

Tillegg B

Instrumentoversikt

	Navn	Type	Nummer
1	Agilent E3631A	Triple output DC Powersuply	7
2	Agilent E3631A	Triple output DC Powersuply	8
3	TTi TGA1244	40MHz Arbitrary Waveform generator	6A
4	HP 5403A	500MHz Digitizing Oscilloscope	17
5	Agilent 54624A	100MHz Oscilloscope	

Tabell B.1: Instrumentoversikt

TILLEGG B. INSTRUMENTOVERSIKT

Tillegg C

Ordliste

TILLEGG C. ORDLISTE

AMS	Austria Microsystems
CMOS	Complementary Metal-Oxide Silicon
EDP	Energy Delay Product
FG	Floating Gate/Flytende inngang
LV	Low Voltage/Lavspenning
MV	Multi Valued/Flernivå
MVL	Multi Valued Logic/Flernivålogikk
PDP	Power Delay Product
SFG	Semi Floating Gate/Halvflytende inngang
STM	ST Microelectronics
ULV	Ultra Low Voltage/Ultralav spenning
UV	Ultra Violet/Ultraviolet

Tabell C.1: Ordliste

Figurer

3.1	Skjematikk for LV inverter	8
3.2	Signalrespons	8
4.1	Signalrespons over to perioder med faser	9
4.2	Skjematikk for initialiseringsstadiet	10
4.3	Forenklet skjematikk for initialiseringsfasen	11
4.4	Skjematikk for evalueringsfasen	12
4.5	Forenklet skjematikk for evalueringsfasen	13
4.6	Simulering av <i>FDL</i> -inverter	14
4.7	flernivå-inverter	15
4.8	Simulering ved 0.5V, 8MHz, 0, 35 μ m	16
4.9	LV inverter m/fast brønn.	17
4.10	LV inverter.	18
4.11	Utlegg for 0, 35 μ m-kretsen.	19
4.12	Utlegg for 0, 13 μ m-kretsen.	20
4.13	44-pinners brikke på 1, 162mm ² . I den røde ruten kan kretsen i figur 4.12 på side 20 sees.	21
4.14	De simulerte portene med klokke drivere	22
4.15	Simulering av <i>ULV</i> - og <i>FDL</i> -inverter	23
4.16	Logisk dybde for <i>FDL</i> og <i>ULV</i> -inverter.	24
4.17	Energi over en periode for <i>FDL</i> og <i>ULV</i>	25
4.18	Maksfrekvens gitt en dybde på 6 for <i>FDL</i> og <i>ULV</i>	25
4.19	Stige- og falltid, 20%-80%, for <i>FDL</i> og <i>ULV</i>	26
4.20	PDP/energi for <i>FDL</i> og <i>ULV</i> i en periode.	26
4.21	EDP for <i>FDL</i> og <i>ULV</i> i en periode.	27
5.1	Simulering ved 0.5V, 4MHz	30
5.2	Simulering ved 0.5V, 4MHz m/fast brønn, 0, 35 μ m	30
5.3	Ut 1 viser responsen til inverteren med brønnene koblet til hhv. klokken og <i>SFG</i> -nodene, mens Ut 2 viser responsen med brønn koblet til hhv. <i>gnd</i> og <i>V_{aa}</i>	32
5.4	Utsnitt av figur 5.3 på side 32	33
5.5	Simulering ved 0.5V, 8MHz, 0, 35 μ m	34
5.6	Simulering ved 0.5V, 8MHz m/fast brønn på 0, 35 μ m-brikke	34
5.7	Simulering ved 0.4V, 8MHz på 0.13 μ m-brikke	35
5.8	Måleoppsett	36
5.9	Måling ved 0.5V, 1kHz på 0.35 μ m-krets	37

FIGURER

5.10 Måling ved 0.5V, 50kHz på 0.13 μ m-krets	38
5.11 Måling ved 0.5V, 200kHz på 0.13 μ m-krets	39
5.12 Måling ved 0.4V, 50kHz på 0.13 μ m-krets	40
5.13 Måling ved 0.4V, 200kHz på 0.13 μ m-krets	41
5.14 Måling ved 370mV, 5kHz på 0.13 μ m-krets	42
5.15 Måling ved 360mV, 5kHz på 0.13 μ m-krets	43

Tabeller

B.1 Instrumentoversikt	55
C.1 Ordliste	58

TABELLER

Bibliografi

- [1] S. Aunet, B. Oelmann, P.A. Norseng og Y. Berg. Reconfigurable sub-threshold cmos perceptron. Universitetet i Oslo, Institutt for informatikk, Mikroelektronikksystemer. Innlevert til vurdering i IEEE Transactions on Neural Networks, Juni 2006.
- [2] Y. Berg, O. Mirmotahari, P.A. Norseng og S. Aunet. Ultra low voltage cmos gates. Universitetet i Oslo, Institutt for informatikk, Mikroelektronikksystemer, Mai 2006.
- [3] Takahiro Hanyu. Challenge of a multiple-valued technology in recent deep-submicron vlsi. I *Proceedings of the 31th ISMVL*, side 241-244, Mai 2001.
- [4] Paul Hasler, Bradley A. Minch og Chris Diorio. Floating-gate devices: They are not just for digital memories anymore. I *Proceedings of IEEE ISCAS '99*, bind 2, side 388-391, June 1999.
- [5] D. Kahng og S. M. Sze. A floating gate and its application to memory devices. *The Bell System Technical Journal*, 46(2):1288-1295, 1967.
- [6] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki og S. Shigematsu and J. Yamada. 1-v power supply high-speed digital circuit technology with multithreshold-voltage cmos. *IEEE Journal of Solide-State Circuits*, 30(8):847-854, 1995.
- [7] Shandrakasan A.P. Sheng og S. Brodersen R.W. Low-power cmos digital design. *IEEE Journal of Solide-State Circuits*, 27(4):473-484, 1992.
- [8] T. Shibata og T. Ohmi. An intelligent mos transistor featuring gate-level weighted sum and threshold operations. *Electron Devices Meeting, 1991. Technical Digest., International*, side 919-922, 1991.
- [9] K. Usami og M. Horowitz. Clustered voltage scaling technique for low-power design. *ISLPED*, side 3-8, 1995.