Kontinuerlig "backpropagation" nett i analog VLSI

Hovedoppgave til Cand. Scient. graden
i informatikk
av

Knut Soelberg

Institutt for Informatikk,
Universitetet i Oslo.
August 1992

Forord

Arbeidet med oppgaven ble påbegynt i september 1990 og er utført ved institutt for informatikk. Min veileder har vært Tor Sverre Lande ved institutt for informatikk.

Jeg vil spesielt takke Tor Sverre Lande som alltid har vært tilgjengelig for all hjelp og veiledning underveis. Jeg vil også takke Yngvar Berg også ved institutt for informatikk som også i stor grad har bidratt med veiledning underveis og som har gitt meg verdifull tilbakemelding av den skriftlige presentasjonen.

Blindern, 15. August, 1992.

Knut Soelberg

Innhold

Figurer

Kapittel 1

Innledning

1.1 Generelt om kunstige nevrale nett og implementasjoner av disse

En del problemer lar seg vanskelig om ikke umulig løse ved hjelp av tradisjonell databehandling generelt og bildebehandling spesielt på tradisjonelle datamaskiner. Klassifisering av en lysstolpe og et tre som to forskjellige objekter er et eksempel på et problem som det er vanskelig å løse med tradisjonell bildebehandling. For et menneske derimot, er det et meget trivielt problem. En slik klassifisering forgår hos mennesker naturligvis i sann tid. Det er derfor fristende å skjele til biologien for å finne inspirasjon til metoder for å løse slike problemer av fuzzy natur.

Syn, hørsel og hjernefunksjoner generelt, består bla. av nevrale nett. Det som kjennetegner slike nett, er at de består av en mengde enkle prosesserende elementer, nevroner som er koblet sammen ved hjelp av synapser. Prosesseringen er massiv parallell og lokal. Biologiske nevroner er langsomme i forhold til tradisjonell digital elektronikk, men med et betydelig mindre energiforbruk. Det er den massive parallelle prosesseringen som bidrar til at prosesseringen foregår i sann tid til tross for at hvert enkelte prosesserende element i seg selv er langsomt.

For å løse problemer slik som i eksemplet med å skille en lysstolpe fra et tre, har man skjelt til biologien og latt seg inspirere av den. Forskning innen såkalte Artificial Neural Networks (ANN) eller på norsk kunstige nevrale nettverk, startet så tidlig som i 50 årene, men først på åttitallet har denne forskningen virkelig økt i omfang.

Kunstige nevrale nettverk er bygd opp av et stort antall svært enkle beregnings- eller prosesseringselementer som i stor grad prosesserer lokalt og parallelt slik det er i biologien. Langsiktig kunnskap eller erfaring blir vanligvis lagret i form av vekter eller synapser på forbindelsene mellom de forskjellige nodene eller nevronene. Kunstige nevrale nettverksmodeller inneholder gjerne algoritmer for at nettet kan lære et passende sett med vektverdier eller tilstander basert på eksempler for ønsket oppførsel. Modeller for kunstige nevrale nett prøver vanligvis ikke å modellere alle detaljer til et virkelig nevron fra biologien. Spesielt opplæringsalgoritmene er som regel svært fjernt fra mekanismer i biologiske nevrale nett. Kunstige nevrale nettverksmodeller har fått en mengde anvendelser, f.eks. generell

mønstergjenkjenning, talegjenkjenning og -syntese, bildekomprimering og -segmentering, navigasjon, værvarsling og estimering av bankkunders kredittverdighet. Det er oppgaver som gjerne egner seg bedre å løse med anvendelse av kunstige nevrale nettverksmodeller enn med tradisjonell databehandling. De to mest vanlige kunstige nevrale nettverksmodellene er nett med en "feedforward" struktur og nett med en tilbakekoblingsstruktur ("feedback").

Hittil er kunstige nevrale nettverksmodeller i stor grad kun vært implementert som programvare for generelle datamaskiner. Det er i dag tilgjengelig et stor antall med programvare som simulerer kunstige nevrale nett, også for praktiske anvendelser.

Det har blitt gjort en del digitale VLSI-implementasjoner av forskjellige nevrale nett [Burr]. Noen digitale implementasjoner har en prosessor for hver synapse, mens de fleste har en prosessor for hvert nevron eller til og med en prosessor for flere nevroner. De fleste digitale implementasjoner så langt har en eller annen form for læring inne på selve den integrerte kretsen.

Det er rapportert om enkelte hybride kretser som kombinerer analog og digital design. En vanlig metode er å lagre vektene digitalt, mens beregningene ellers foregår analogt.

Intel har realisert en nevral nettverkskrets som de kaller "Electrically Trainable Artificial Neural Network" (ETANN) [Holler]. Her er både beregningene og vektene analoge, men opplæringen foregår eksternt.

Analog VLSI med transistoren operert i subterskelområdet gir mulighet for å realisere store analoge beregningssystemer slik som store nevrale nett. Begrensningen er primært størrelsen av det tilgjengelige silisiumarealet i den integrerte kretsen. Kopiering av signaler som er resultater innen beregningssystemet, kan nærmest utføres ubegrenset slik at signaler kan benyttes ved mange forskjellige delberegninger i et større beregningssystem.

1.2 Oppgavens mål

Oppgaven gikk ut på å realisere og karakterisere kretselementer og moduler for sammensetning av et vilkårlig nevralt nettverk av typen "feedforward net with backpropagation of error" [Rumelhart86], også bare kalt "backpropagation" nett, i analog VLSI hvor CMOS transistoren opereres i subterskelområdet. Modulene skulle også inneholder kretselementer for beregning av opplæringsdelen av algoritmen. For lagring av et netts vekter skulle jeg benytte analoge "floating gate" hukommelseselementer.

Som et eksempel har jeg realisert et lite nettverk hvor jeg har tatt i bruk de definerte modulene. Jeg har også gjort forsøk på å lære opp nettet.

En av grunnidéene var å unngå ekstra (digital) kontrollogikk og enhver form for klokking. En slik implementasjon vil utnytte den massive parallelliteten innebygd i algoritmene for nevrale nett. Et av problemene har tidligere vært langtidslagring av analoge vekter. Floating gate teknikken har gitt oss en løsning av problemet. Imidlertid kan det fremdeles være problemer med god nok oppløsning av hukommelseselementets arbeidsområde samt tilstrekkelig kort programmeringstid av hukommelseselementet.

De definerte modulene for sammensetning av et nevralt nett vil gi et nett som er helt analogt og uten noen form for klokking. Et slikt nett vil nok ha mer til felles med biologien enn det en digital implementasjon har. Avstanden til biologien er selvfølgelig fremdeles

stor, spesielt opplæringsalgoritmen (delta-regelen) er noe som er særegent for kunstige nevrale nett. De fleste VLSI implementasjoner av kunstige nevrale nett, spesielt digitale, vil inneholde ekstra kontrollogikk, klokking og gjerne adressering av vektene (synapsene). Dessuten kan prosessorene (prosesseringselementene) i digitale nett neppe betraktes som enkle i forhold til biologiske nett. Min implementasjonen har flere likhetstrekk med biologiske nett, f.eks. enkle prosesseringselementer som drar nytte av transistorens fysiske egenskaper, massiv parallellitet, signalstørrelse og -hastighet og energiforbruk.

Fagområdet kunstige nevrale nett er et stort tverrfaglig fagområde. Denne oppgaven er vinklet mot realisering av nevrale nett i analog mikroelektronikk. De to første kapitlene er til en viss grad en grunnleggende innføring i henholdsvis den aktuelle nevrale nettverksalgoritmen og i teori om CMOS transistoren i subterskelområdet.

Jeg har i oppgavebesvarelsen i stor grad holdt meg til norske termer. Jeg har imidlertid benyttet enkelte engelske termer hvor det har vært vanskelig å finne passende norske ord og uttrykk. De viktigste er **gate**, **source** og **drain** som er betegnelsen på CMOS transistorens tre terminaler. En annen term er **wide-range** som i **wide-range** transkonduktansforsterker, som er en transkonduktansforsterker uten sterke begrensninger i operasjonsområdet. Den siste termen er **floating gate** som i **floating gate** hukommelse, som er en teknologi for analog langtidshukommelse.

Kapittel 2

Nettarkitekturen

2.1 Feedforward nett

Nettarkitekturen i et lagdelt feedforward nett er vist i figur 2.1. Et slikt nett består av et sett noder eller nevroner arrangert i to eller flere lag [Rumelhart86]. Utgangen fra nodene i et lag går inn på inngangen til nodene i neste lag etter å ha blitt forsterket eller svekket av såkalte vekter. Bortsett fra nodene i inngangslaget, er hver nodes inngang summen av de veide utgangene fra nodene i underliggende lag. Komponentene i et inngangsmønster utgjør inngangssignalet til nodene i lag i slik som vist i figur 2.1. Utgangene fra disse nodene kan være identiske med inngangene, eller vi kan normalisere disse verdiene slik at de blir skalert innenfor et bestemt verdiområde.

Inngangssignalet til en node i lag j er gitt ved

$$net_j = \sum_i w_{ji} o_i, \tag{2.1}$$

hvor w_{ji} er vekter som veier inngangssignaler o_i. Utgangen fra noden i lag j er gitt ved

$$o_j = f(net_j),$$

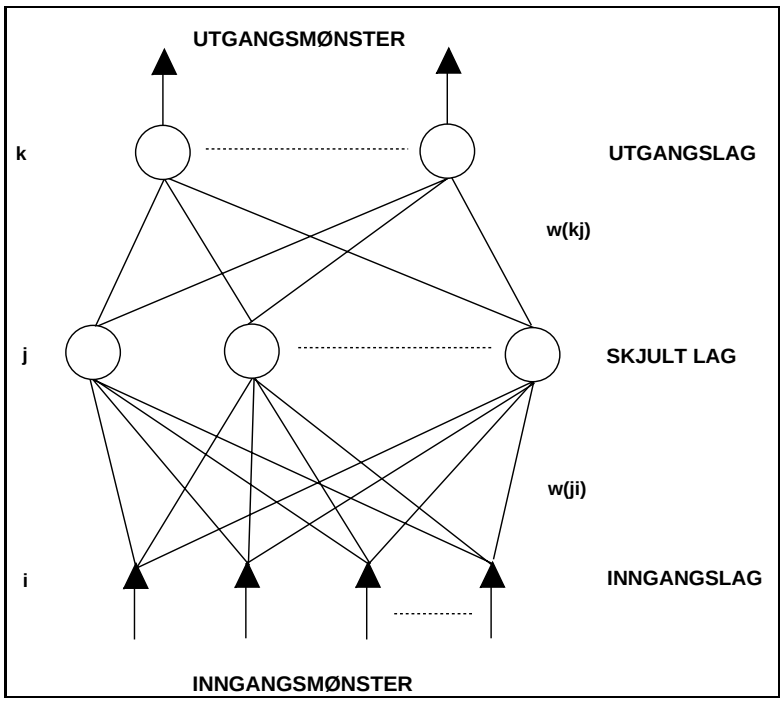
hvor f er aktiviseringsfunksjonen. Det er mest vanlig å bruke en sigmoid aktiviseringsfunksjon. I litteraturen brukes gjerne

$$o_j = \frac{1}{1 + e^{-(net_j + \Theta_j)/\Theta_0}}, \tag{2.2}$$

hvor Θ_j er en terskel. Θ_0 bestemmer steilheten på aktiviseringsfunksjonens overgang fra lavt til høyt nivå. En positiv Θ_j vil forflytte aktiviseringsfunksjonens overgang fra lavt til høyt nivå til høyre langs den horisontale aksen. Figur 2.2 viser en slik sigmoid aktiviseringsfunksjon og dens deriverte.

I analog VLSI kan vi implementere en annen sigmoid funksjon, nemlig tangenshypobolikus. Utgangsstrømmen fra en transkonduktansforsterker gir en tangenshypobolikus funksjon [Mead]. Strømmen I_{out} fra en transkonduktansforsterker er gitt ved

$$I_{out} = I_b \tanh(\kappa \frac{V_1 - V_2}{2V_T}),$$



Figur 2.1: Lagdelt nettstruktur.

som er en funksjon av differansen mellom inngangsspenningene V_1 og V_2, og hvor κ og V_T er konstanter. I motsetning til funksjonen i likning 2.2 og figur 2.2 som varierer fra 0 til 1, så varierer strømmen ut fra en transkonduktansforsterker mellom $-I_b$ og I_b, hvor I_b er strømmen gjennom forspenningstransistoren.

Fortsetter vi videre beregningene i nettet for nodene i lag k, er disse inngangene gitt ved

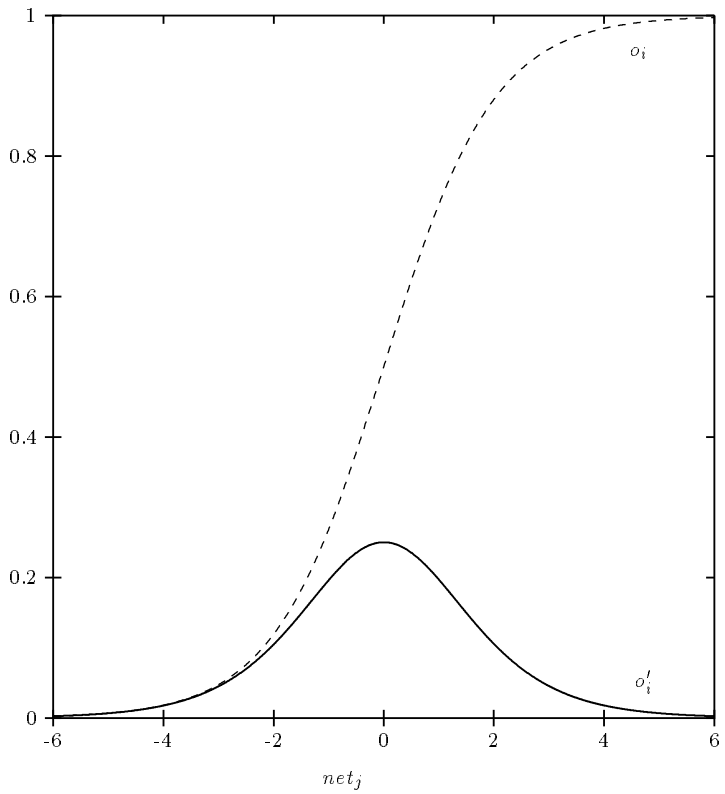
$$net_k = \sum_j w_{kj} o_j,$$

og de korresponderende utgangene er gitt ved

$$o_k = f(net_k).$$

2.2 Den generaliserte delta regelen

Det er flere metoder som kan benyttes ved en opplæring av et feedforward nett. En mulighet er å presentere et inngangsmønster $\underline{x}_p = \{i_{pi}\}$ og la nettet justere settet med vekter og terskler slik at den ønskete utgangsverdien t_{pk} fåes på utgangsnodene [Rumelhart86]. Når det er gjort, presenterer vi et nytt par med $\underline{x}_p$ og $\{t_{pk}\}$ og lar nettet lære også den assosiasjonen. Generelt lar vi nettet finne et sett med vekter og terskler som tilfredsstiller alle inngangs- og utgangspar den får presentert.



Figur 2.2: Sigmoidfunksjonen o og dens deriverte o', hvor $\Theta_j = 0$ *og* $\Theta_0 = 1$.

Generelt vil utgangsverdiene $\{o_{pk}\}$ ikke bli helt identiske med de ønskete utgangsverdiene $\{t_{pk}\}$. For hvert mønster p vil kvadratet av feilen for nettet være

$$E_p = \frac{1}{2}\sum_k (t_{pk} - o_{pk})^2,$$

og den gjennomsnittlige systemfeilen E, altså feilen etter at hele settet med mønstre $\underline{x}_p$ er presentert en gang, vil være

$$E = \frac{1}{2P}\sum_p \sum_k (t_{pk} - o_{pk})^2. \tag{2.3}$$

Den generaliserte delta regelen (GDR) som ble formulert av Rumelhart, Hinton og Williams for opplæring av vekter og terskler tar utgangspunkt i feilen. GDR varierer vektene på en slik måte at feilen E_p reduseres hurtigst mulig, kalt gradient søk. En gradient søk for minimum system feil bør baseres på en minimalisering av uttrykket i likning 2.3.

Selve utledningen er beskrevet i [Rumelhart86]. GDR beregner endringen av vekten. Endringen kan være positiv eller negativ og er for vekter w_{ji} gitt ved

$$\Delta_p w_{ji} = \eta \delta_{pj} o_{pi}, \tag{2.4}$$

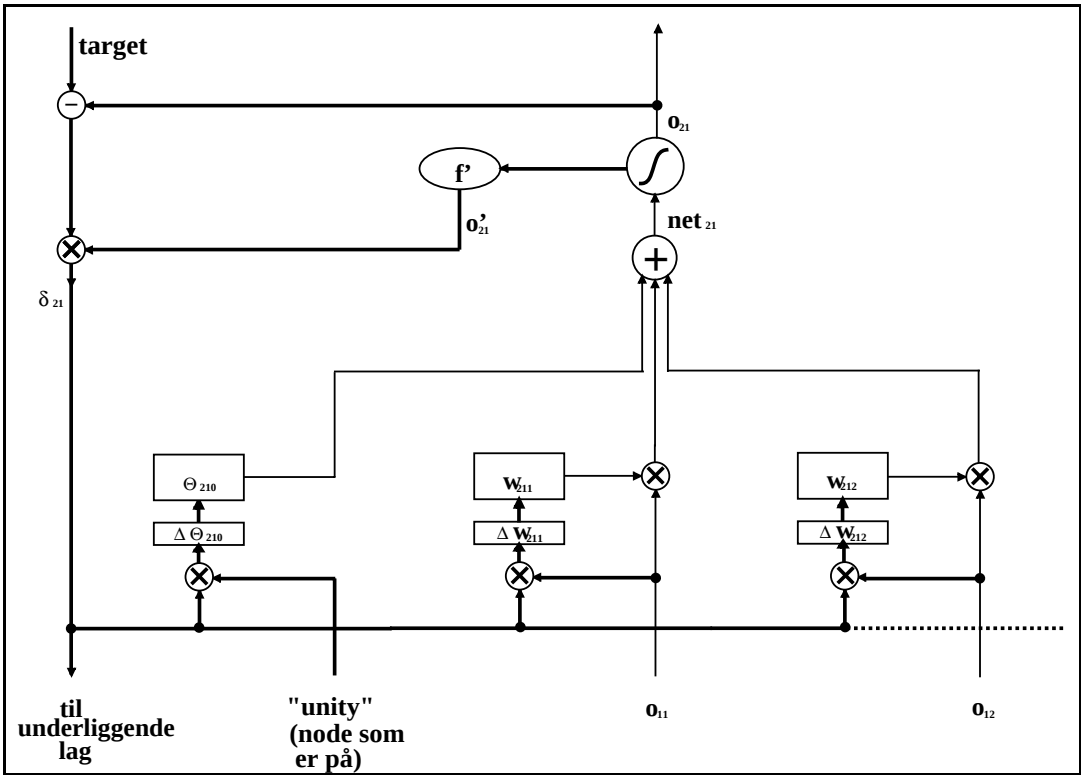
hvor η er en parameter som vil påvirke opplæringshastigheten. Dersom nodene j er noder i utgangslaget, er δ_{pj} gitt ved

$$\delta_{pj} = (t_{pj} - o_{pj}) f_j'(net_{pj}).$$

Dersom nodene j er interne noder, så evaluerer vi δ_{pj} uttrykt ved δ'er i overliggende lag:

$$\delta_{pj} = f'_j(net_{pj}) \sum_k \delta_{pk} w_{kj}. \tag{2.5}$$

Man bør merke seg at tersklene Θ_j læres på samme måte som andre vekter. Vi antar bare at Θ_j er en vekt fra en node som alltid er "på".



Figur 2.3: Skjematisk beskrivelse av en node i utgangslaget samt vekter som veier nodens inngangssignaler. De brede linjene representerer backpropagation delen av algoritmen, mens de vanlige linjene representerer feedforward delen.

I følge [Pao], kapittel 5.3, er det god praksis å beregne $\Delta_p w_{ji}$ for hvert av mønstrene i opplæringssettet av mønstre, og deretter beregne

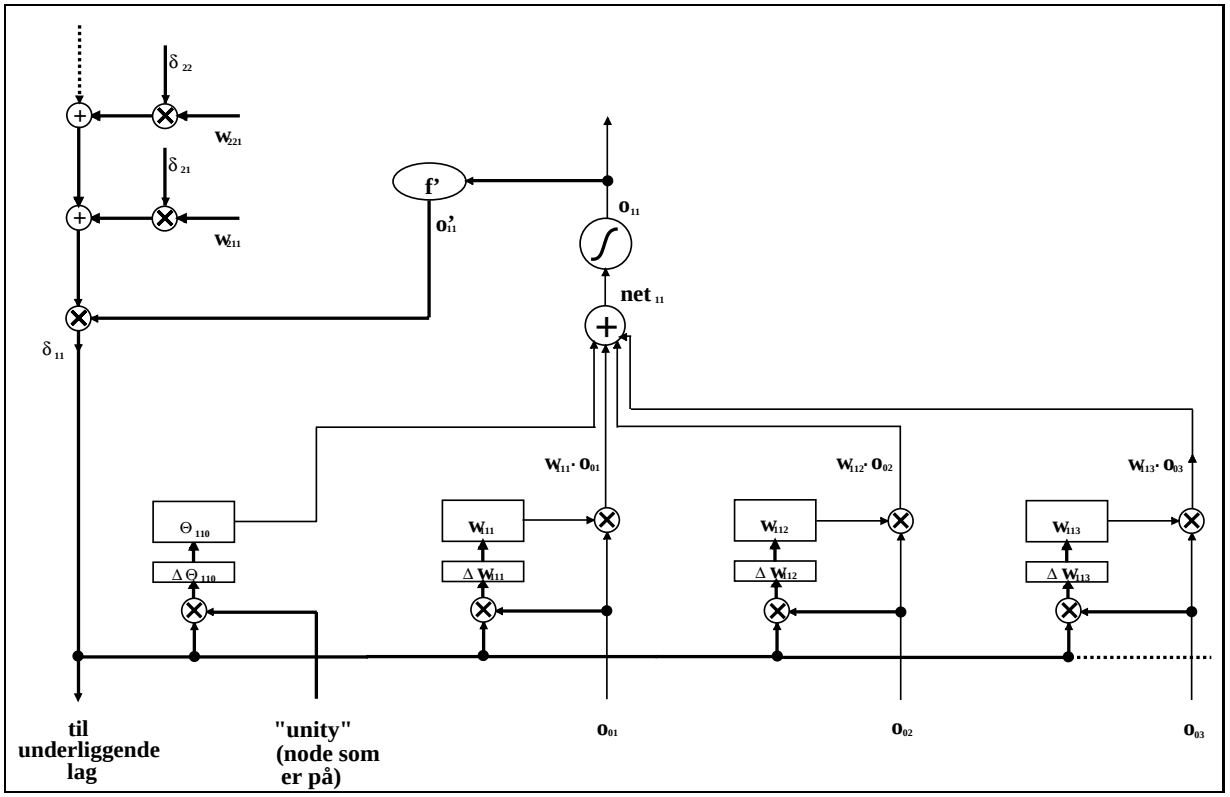
$$\Delta w_{ji} = \sum_p \Delta_p w_{ji}, \tag{2.6}$$

og så gjøre selve endringen av vekten w_{ji}.

Ved en vellykket opplæringsfase, vil systemfeilen E avta med antall gjentakelser mønstrene i opplæringssettet blir presentert, og verdiene av vektene vil stabliseres.

I en opplæringsfase er det flere hensyn som må taes, bla. valg av η. En stor η vil gi en hurtig opplæring, men kan også resultere i oscillasjon. [Rumelhart86] foreslår å inkludere en form for momentledd i likningen for endringen av vekten

$$\Delta w_{ji}(n+1) = \eta(\delta_j o_i) + \alpha \Delta w_{ji}(n), \tag{2.7}$$



Figur 2.4: Skjematisk beskrivelse av en node i et skjultlag samt vekter som veier nodens inngangssignaler. De brede linjene representerer backpropagation delen av algoritmen, mens de vanlige linjene representerer feedforward delen.

hvor n betyr presentasjonen nr n av opplæringssettet. α er en proporsjonalitetskonstant slik at det andre leddet i likning 2.7 sørger for en treghet i systemet. Passende valg av α vil dempe en oscillasjon, men den kan også dempe opplæringshastigheten.

Det vil i en kontinuerlig implementasjon være vanskelig (og unaturlig) å akkumulere endringen av en vekt etterhvert som hvert mønster i opplæringssettet blir presentert, for til slutt å gjøre selve endringen av vekten. Min implementasjon er kontinuerlig slik at endringer av vekten vil skje kontinuerlig. Jeg har ikke implementert beregningene i likning 2.6 og 2.7. Disse likningene er sannsynligvis i første rekke tenkt implementert i programvare for en generell datamaskin eller i skreddersydd digital maskinvare. En analog implementasjon av et nevralt nett vil ha innebygget flere tidskonstanter. Tregheten ved programmering av en vekt i en analog implementasjon kan tilsvare det andre leddet i likning 2.7 som også er et uttrykk for en treghet ved endringen av en vekt.

Når endringen av vektene skjer kontinuerlig ved presentasjon av hvert enkelt mønster, er det ikke en virkelig gradient søk. Konstanten η bør derfor være liten for å unngå for store sprang i endringen av vektene.

Nettstrukturen i figur 2.1 viser skjematisk kun feedforward-delen av algoritmen. Figur 2.3 og 2.4 viser skjematisk hele algoritmen slik jeg har implementert den for en enkelt

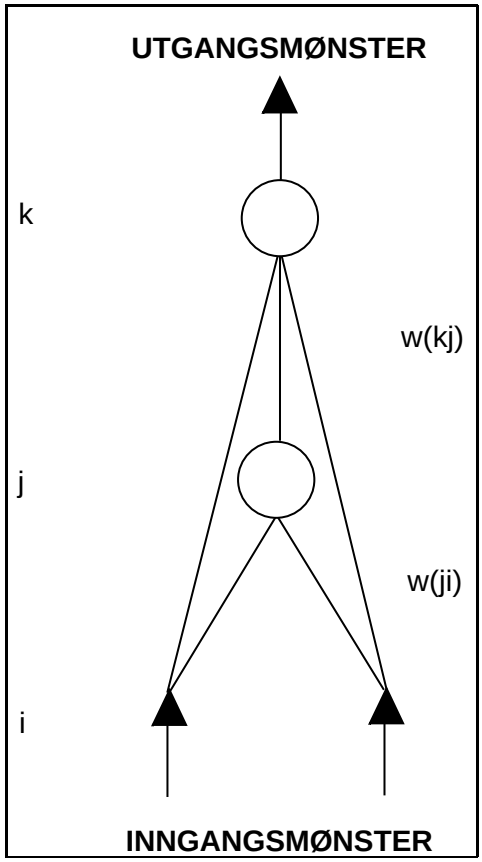
node for henholdsvis en node i utgangslaget og i et skjult lag. De ekstra tykke linjene i figur 2.3 og 2.4 representerer "back propagation of error"-delen av algoritmen, mens de vanlig tynne linjene representerer feedforward-delen av algoritmen. I begge figurene kan vi se at inngangen *net* til noden er summen av de veide innsignalene i tillegg til terskelen Θ. Dette er feedforward-delen av algoritmen. "Back propagation of error"-delen av algoritmen beregner endringen av vektene. I figur 2.3 ser vi at endringer av vektene for noder i utgangslaget beregnes på grunnlag av forskjellen mellom ønsket og faktisk utsignal (feilen), mens i figur 2.4 ser vi at endringene av vektene for noder i skjulte lag beregnes på grunnlag av forplantet feil fra overliggende lag.

2.3 En nettarkitektur som kan læres opp til å løse det klassiske xor-problemet

Jeg har valgt som et eksempel å realisere et backprop nett sammensatt av definerte moduler i analog VLSI som etter en programmering er tiltenkt oppgaven å kunne løse det klassiske xor-problemet. Et nevralt nett som er ment å kunne løse xor-problemet kan virke banalt. En vanlig digital xor-port med to innganger har som kjent et lavt utsignal når de to innsignalene er like, mens et høyt utsignal når de to innsignalene er forskjellige. Minsky og Papert viste i 1969 at de såkalte perceptronene som er nett uten interne lag, ikke kunne løse problemer som gikk ut på å lære et mønster av xor-typen. Det medførte mange stille år innen nevral nettverksforskning før det igjen ble ble svært aktivt innen området.

Nettarkitekturen som løser det klassiske xor-problemet er beskrevet i [Rumelhart86] og vist i figur 2.5. Det består av en skjult node og en node i utgangslaget. I tillegg har nettet to innganger og fem vekter. Nettarkitekturen i figur 2.5 har riktig nok ikke en helt lagdelt arkitektur. Selve algoritmen er derimot den samme. Vi kan dessuten betrakte nettet som et vanlig lagdelt nett dersom vi forestiller oss at det i signalveien for hvert av de to signalene som går direkte fra inngangen til utgangslag noden er plassert en vekt som alltid er lik 1 etterfulgt av en node som alltid er "på".

Xor-nettet er et lite nett, men det inneholder beregningselementer for alle beregninger som utføres i et vilkårlig backprop nett. Jeg har derfor som et eksempel valgt å realisere nettopp dette nettet i analog VLSI. Dessuten er det i forbindelse med testing, feilsøking og karakterisering av en realisert krets i VLSI en stor fordel at kretsen som utgjør nettet er lite.



Figur 2.5: En nettarkitektur som er i stand til å løse det klassiske xor-problemet. Det er riktignok ikke et lagdelt nett, men algoritmen forblir den samme.

Kapittel 3

CMOS transistoren og UV-strukturer

3.1 Subterskel analog CMOS

I subterskelområdet (svak inversjon) er drain-source strømmen I_{ds} gjennom en n-kanal CMOS transistor gitt ved

$$I_{ds} = I_0 e^{\frac{q}{kT}(\kappa V_g - V_s)}(1 - e^{-\frac{q}{kT}V_{ds}}), \tag{3.1}$$

hvor V_g er gate spenningen, V_{ds} er drain-source spenningen og I_0 er en fysisk avhengig konstant. Videre er k Boltzmann-konstanten, T er temperaturen, q er ladningen til de mobile partiklene (elektronene). $V_T = \frac{kT}{q}$ kalles den termiske spenningen og er omtrent lik $25mV$ ved romtemeratur. κ er et uttrykk for effektiviteten av gate spenningen V_g og er ≤ 1.

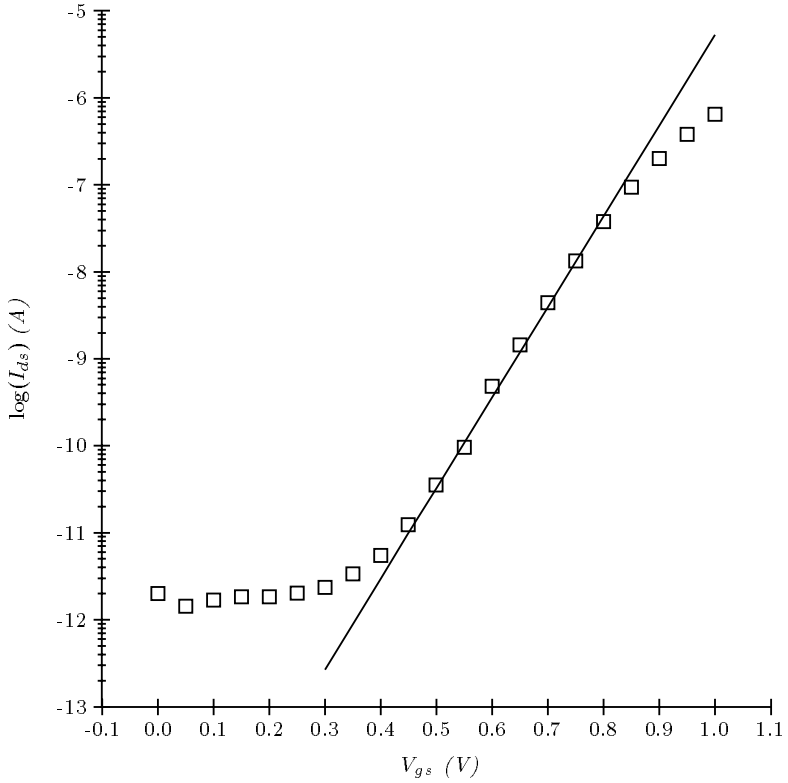
Strømmen I_{ds} gjennom transistoren som funksjon av gate spenningen V_{gs} er vist i figur 3.1 og plottet i en lin/log skala. Heltrukket kurve er beregnet etter likning 3.1. Firkanter representerer målte verdier. Det lineære området som er subterskelområdet, strekker seg frem til terskelspenningen V_{th}, som i figur 3.1 er omtrent 0.85V. For å være mer presis, er det en overgangssone kalt moderat inversjon mellom svak inversjon og sterk inversjon. Årsaken til at området hvor $V_g < 0.4V$ er flatet ut er særlig støy, men også unøyaktigheter i måleinstrumentene.

En p-kanal transistor er tilsvarende en n-kanal transistor bortsett fra at ladningsbærerne er "hull" i steden for elektroner. Strømlikningen for en p-kanal transistor er tilsvarende likning 3.1 bortsett fra at alle spenningene har motsatt fortegn.

I subterskelområdet er den mobile ladningen q_m pr. areal mye mindre enn utarmings (depelsjons) ladningen i substratet. Ved terskelspenningen V_{th} vil den mobile ladningen begynne å begrense diffusjonsstrømmen og en kanal vil oppstå mellom source og drain ettersom V_{gs} stiger over terskelspenningen V_{th}.

Transistorene operert i subterskelområdet fører til mange fordeler (og noen ulemper).

- Kretsene får et svært lavt effektforbruk.



Figur 3.1: Strømmen I_{ds} gjennom en transistor som funksjon av gatespenningen V_{gs} ved konstant spenning V_{ds} over transistoren, hvor source- og substratspenningen begge er lik GND. Firkanter representerer målinger, mens heltrukken kurve representerer teoretiske beregninger.

- I metningsområdet ($V_{ds} \geq 4V_T$) fungerer MOS transistoren som en spenningskontrollert strømkilde. Vi kan da forenkle likningen for strømmen I_{ds} gjennom transistoren til
$$I_{ds} = I_0 e^{\frac{\kappa V_g - V_s}{V_T}}.$$
- Transistorens eksponentielle egenskaper i subterskelområdet gjør den ideell til å bygge større kretser som utfører forbausende avanserte beregninger.

3.1.1 Transistorvariasjoner

Transistorer i en integrert krets av samme type og størrelse er ikke identiske. Den fysiske konstanten I_0 vil for tilsynelatende like transistorer variere. Forskjellen i verdien av I_0 for to vilkårlige og tilsynelatende identiske transistorer kan variere med en faktor opptil 2 [Mead]. For to nærliggende transistorer vil 20% avvik være et mer typisk tall. Noe som tilsvarer en forskjell i gatespenningen på $\pm 10mV$ i svak inversjon. Det er altså en randomisert variasjon mellom vilkårlig og tilsynelatende identiske transistorer i en silisiumbrikke.

3.1.2 Early effekt

Lengden l av en gitt transistorkanal er ikke konstant. Lengden l er avstanden mellom utarmingsområdene (deplesjonsområdene) for transistorens source og drain terminal. Utarmingsområdene rundt de to terminalene er en funksjon av source-substrat og drain-substrat spenningen. Størrelsen av utarmingslaget rundt drain terminalen vil øke med økende drain spenning som vil føre til at transistorkanalen forkortes. Forkortingen av kanalen vil føre til at strømmen i kanalen (drain-source strømmen I_{ds}) økes. Økningen av I_{ds} som følge av modulering av kanallengden kalles Early-effekten.

Drain-source strømmen I_{ds} som funksjon av drain-source spenningen V_{ds} ved forskjellige gate-source spenninger V_{gs} er vist i figur 3.2. Gate-source spenningen er i subterskelområdet. Strømmen I_{ds} vil være i metning etter få mV drain-source spenning V_{ds}.

Videre er drain-source strømmen når den er i metning omvendt proporsjonal med transistorlengden l. Drain konduktansen g_d i metningsområdet er da gitt ved [Mead]

$$g_d = \frac{\partial I}{\partial V_d} \approx -\frac{1}{V_0},$$

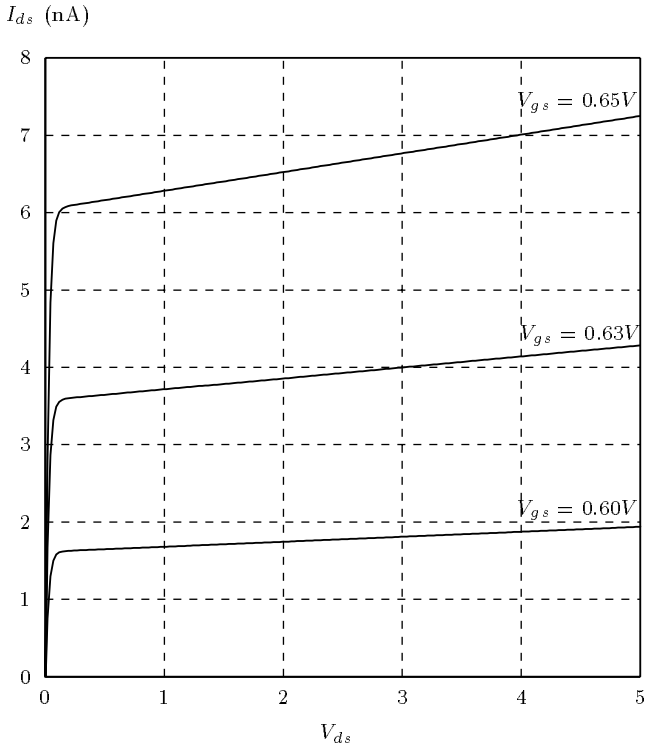
hvor V_0 er en konstant (Early-konstanten) for en gitt transistorstørrelse. Dersom vi ekstrapolerer I_{ds}-kurvene i metningsområdet for de ulike spenningene V_{gs}, vil alle kurvene krysse V_{ds}-aksen i samme punkt $-V_0$. For kurvene i figur 3.2 er $V_0 \approx -25V$.

Dersom vi tar hensyn til drain konduktansen vil likning 3.1 for strømmen I_{ds} bli

$$I_{ds} = I_0 e^{\frac{q}{kT}(\kappa V_g - V_s)}(1 - e^{-q\frac{V_{ds}}{kT}} + \frac{V_{ds}}{V_0}). \tag{3.2}$$

Dersom transistoren er i metning kan likning 3.2 forenkles til

$$I_{ds} = I_{sat}(1 + \frac{V_{ds}}{V_0}),$$



Figur 3.2: Source-drain strømmen I_{ds} gjennom en transistor som funksjon av drain-source spenningen V_{ds} ved ulike gatespenninger V_{gs} i subterskelområdet. Merk at transistoren mettes etter noen få mV slik at transistoren opereres som en strømkilde over det meste av operasjonsområdet.

hvor

$$I_{sat} = I_0 e^{\frac{q}{kT}(\kappa V_g - V_s)}.$$

Kopiering av strømmer gjøres lett ved bruk av strømspeil. Som nevnt vil transistorvariasjoner bidra til at strømkopien ikke blir helt lik orginalen. I tillegg vil Early-effekten ha en betydelig innvirkning i et strømspeil dersom det er stor forskjell i drain spenningene mellom de to transistorene.

3.1.3 Substrat effekt

Dersom vi øker source spenningen V_s for en n-kanal CMOS transistor, må vi øke gate-source spenningen V_{gs} enda mer for å holde strømmen I_{ds} konstant. Gate spenningen V_{gs} har mindre effekt på drain-source strømmen I_{ds} enn det source spenningen V_s har. Denne effekten kalles substrat effekt (body effekt, bulk effekt). Likning 3.2 er basert på en konstant κ, som er et uttrykk for effektiviteten av gate spenningen, utledet ved en lineærisering rundt operasjonspunktet for overflate potensialet.

3.1.4 Oppsummering

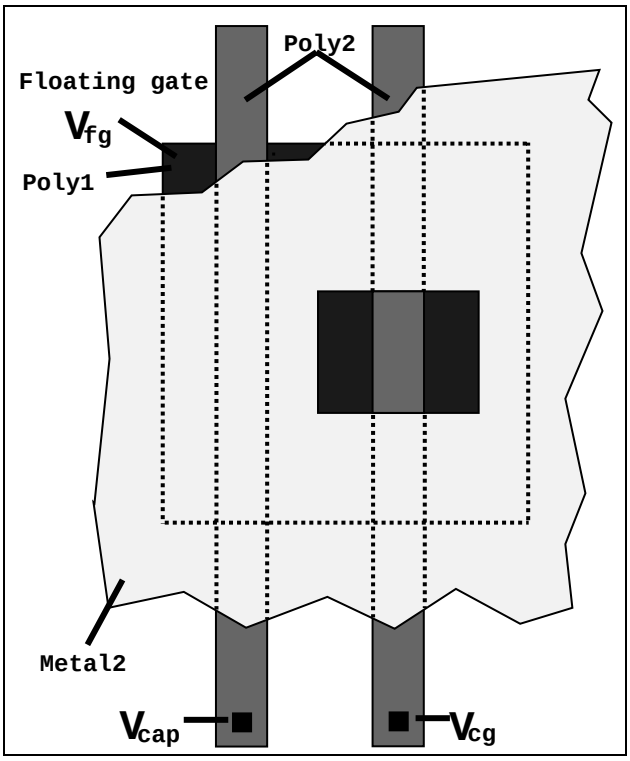
Det som man til en viss grad kan frykte, er at transistorvariasjoner og Early-effekten i værste fall vil være så dominerende at kretselementer ikke utfører beregningene tilfredstillende. Variasjoner er et fenomen som man må leve med i analog VLSI. Målet er til tross for variasjonene at de beregningene som skal utføres i en VLSI-implementasjon er robuste nok til at variasjonene kan tolereres. Det er også til en viss grad mulig å kompensere for variasjonene.

3.2 Analog langtids hukommelse

Analog hukommelse for langtidslagring kan med standard CMOS teknologi realiseres dersom vi kan plassere en bestemt ladning som representerer en viss spenning, på en node med minimal lekkasje. Vi kan benytte oss av at mellom hvert lag i en integrert krets så er det et lag med silisiumdioksyd som isolerer de ulike lagene fra hverandre. Dersom en integrert krets blir utsatt for ultrafiolett lys (UV-lys), vil laget med silisiumdioksyd bli ledende, riktignok med en høy motstand. En ladning kan dermed plasseres på en transistor gate terminal ved bruk av UV-lys dersom floating gate noden har et overlapp med et lag med en kontrollspenning. Når UV-lyset skrues av, vil ladningen forbli på floating gate noden.

Det vil mellom to overlappende noder (den ene i poly1 og den andre i poly2) være et isolerende lag med silisiumdioksyd som fører til en kapasistans mellom nodene. Den ene noden er floating gate noden og den andre er noden som får påtrykt en kontrollspenning. Dersom de to nodene blir bestrålt med UV-lys, vil det i tillegg til kapasistansen mellom dem oppstå en liten konduktans i parallell med kapasistansen. Den lille konduktansen kan da anvendes til å tilegne floating gate noden en ladning ved å påtrykke den andre noden en kontrollspenning.

3.2.1 UV-struktur

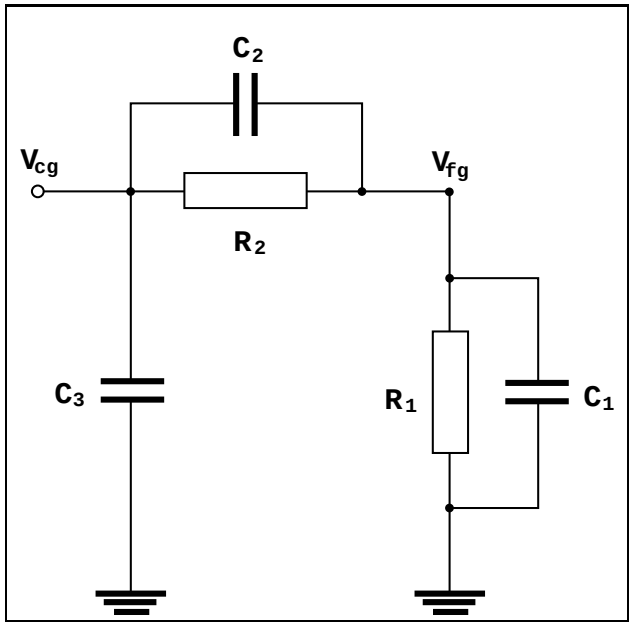


Figur 3.3: Utleggsmessig skisse av UV-strukturen. Floating gate noden V_{fg} er i poly1. To elektroder i poly2 krysser V_{fg}, henholdsvis V_{cg} som påtrykkes kontrollspenningen og V_{cap} som kan anvendes til å eliminere den totale last kapasistansen på floating gate noden. Over det hele ligger et dekklag i metall2, bortsett fra et lite vindu hvor programmering av V_{fg} etter V_{cg} vil foregå ved bruk av UV-lys.

Figur 3.3 viser en struktur for UV analog hukommelse [Maher]. Strukturen består av en flytende kondensator gjort i poly1 (merket V_{fg}) som krysses av to elektroder i poly2 (merket V_{cg} og V_{cap}). Hele strukturen er skjermet fra lys med et lag metall2, bortsett fra et lite vindu hvor programmering av V_{fg} etter elektrode V_{cg} vil foregå. Spenningen på floating gate noden kan betraktes/føres videre ved å koble V_{fg} til gate terminalen på inngangstransistoren til en spenningsfølger.

3.2.2 Elektrisk modell

Figur 3.4 viser en enkel elektrisk modell av UV-strukturen i figur 3.3 når den utsettes for UV-lys. R_2 modellerer konduktansen som oppstår mellom poly2 elektroden V_{cg} og floating gaten V_{fg}. I tillegg vil det gjerne være parasitt konduktans til andre lag slik som til substratet og metall2. Dersom disse to lagene begge er koblet til GND, kan parasitt konduktansen modelleres slik som R_1 i figuren. C_1 og C_3 modellerer kapasistansene til GND for henholdsvis floating gate noden og kontroll gate noden. C_2 modellerer kapasistansen mellom floating gate noden og kontroll gate noden. Vi antar at C_1 og C_3 vil være konstante



Figur 3.4: Elektrisk modell for UV-strukturen når den utsettes for UV-lys.

ved varierende spenning, mens C_2 vil være en funksjon av spenning fordi V_{fg} er koblet til gate terminalen på en CMOS transistor som er inngangstransistor til en spenningsfølger.

Den matematiske utledningen av modellen er beskrevet i [Maher]. For en gitt bølgelengde lys vil strømmen gjennom R_1 og R_2 være gitt som en funksjon av spenningen over dem [Maher], henholdsvis

$$I_1 = \alpha_1 (V_{fg})^{\frac{3}{2}}$$

og

$$I_2 = \alpha_2 (V_{cg} - V_{fg})^{\frac{3}{2}}$$

Den statiske konduktansen til elementene R_1 of R_2 er gitt ved

$$R_1 = \beta_1 (V_{fg})^{\frac{1}{2}}$$

og

$$R_2 = \beta_2 (V_{cg} - V_{fg})^{\frac{1}{2}}. \tag{3.3}$$

De dynamiske konduktansene er

$$r_1 = \frac{\partial I_1}{\partial V_{fg}}$$

og

$$r_2 = \frac{\partial I_2}{\partial (V_{cg} - V_{fg})}.$$

Videre er tidskonstanten gitt ved

$$\tau = \frac{r_1 r_2}{r_1 + r_2} \frac{1}{C_1 + C_2}.$$

α_1, α_2, β_1 og β_2 er konstanter som følge av foran nevnte antakelser.

3.2.3 Oppsummering

Modellen til [Maher] forteller oss at konduktansen mellom kontroll gate og floating gate (UV-hullet) er ikke-lineær og avhengig av spenningsforskjellen mellom V_{fg} og V_{cg}. Det medfører at konduktansen stadig vil bli mindre (motstanden vil øke) ved avtagende spenningsforskjell som betyr at floating gate spenningen V_{fg} aldri vil bli helt lik kontroll gate spenningen V_{cg}.

Dessuten vil det alltid være en kapasitiv kobling mellom floating gate og kontroll gate nodene. Det betyr at etter at programmeringen av floating gate noden er avsluttet, må det tas forhåndsregler for å hindre at floating gate noden blir forstyrret ved endringer av kontroll gate noden.

Kapittel 4

Grunnleggende kretselementer

Vi kan se hvilke typer beregninger som foregår i et backprop nett ved å betrakte figur 2.3 og 2.4. En beregning som går meget igjen i de to figurene er multiplikasjon. Dessuten utføres det addisjoner og en subtraksjon for hver node i utgangslaget . Hver node i et backprop nett gir en sigmoidfunksjon på utgangen i tillegg til den deriverte av sigmoidfunksjonen. Til slutt er det behov for lagring av vektene i nettet. Grunnleggende kretselementer for lagring av en vekt og de nevnte beregninger vil bli beskrevet etter valg av signalrepresentasjon. Jeg vil dessuten presentere målinger av utsignaler for hvert av de grunnleggende kretselementene.

Der hvor det ikke nødvendigvis kommer frem av sammenhengen, har jeg merket om et signal er en strøm eller en spenning ved å la selve signalnavnet være en indeks til henholdsvis I for et strømsignal og V for et spenningssignal, f.eks. I_{out} og V_{in}.

4.1 Signalrepresentasjon

Ved å undersøke figur 2.3 og 2.4, ser vi at de aller fleste beregninger som forgår i forbindelse med et vilkårlig nevron (node) i et backprop nett kan gi både positivt og negativt resultat når sigmoidfunksjonen ut fra nevronet er tangenshypobolikus. Eneste unntak er den deriverte av utgangssignalet fra nevronet. Siden utgangssignalet fra nevronet er en sigmoid, vet vi at den deriverte av dette signalet har store positive verdier når inngangssignalet til nevronet net_j er nær null. Ellers konvergerer den deriverte mot null (fra positiv side) slik som vist i figur 2.2.

For å representere både positive og negative signaler, har jeg valgt å benytte en differensiell signalrepresentasjon. Et signal vil da i praksis være representert ved hjelp av to komponenter, en negativ komponent og en positiv komponent.

Et signal vil ofte benyttes i flere beregninger. Speiling av strømsignaler ved bruk av strømspeil er enkelt og praktisk for å lage en eller flere kopier av et strømsignal. Strømsignalet som kopieres i et strømspeil har kun en retning. Et strømsignal som kan være både positivt og negativt vil i et strømspeil likerettes slik at kun den positive eller negative delen av signalet vil speiles. Ved bruk av en differensiell signalrepresentasjon vil de to signalkomponentene alltid ha en bestemt retning slik at speiling av de to signalkomponentene vil være enkelt.

Det er også mulig å ha en signalrepresentasjon med kun en komponent. Det vil imidlertid kreve et referansesignal for å angi nullpunktet. Referansesignalet må da distribueres til den negative inngangen på alle differensielle par slik som til f.eks den negative inngangen på en transkonduktansforsterker. Dersom referansesignalet som da settes eksternt, settes f.eks. til $2.5V$, vil et signal mindre enn $2.5V$ representere et negativt signal, mens et signal større enn $2.5V$ vil representere et positivt signal. En fordel med en slik signalrepresentasjon er at antall signallinjer reduseres betraktlig, og det hele tar mindre plass i et kretsutlegg. Problemet er valg av referansesignal. Dessuten må referansen distribueres dit det er behov for den. Det vil muligens også være behov for ikke en, men flere referanser Jeg har derfor valgt en differensiell signalrepresentasjon.

4.2 Transkonduktansforsterkeren

En vanlig transkonduktansforsterker gir en sigmoidfunksjon på utgangen. Nevronene i inngangslaget skal kun skalere inngangsverdiene. Vi kan derfor bruke en transkonduktansforsterker slik som vist i figur 4.1. Det er en transkonduktansforsterker med "wide-range"-egenskaper siden vi tar ut utgangssignalet i form av en positiv og en negativ komponent fra diodekoblinger. Mer nøyaktig er kretsen i figur 4.1 kun et differensielt par med en diodekobling i toppen av hvert ben i det differensielle paret.

Nevronene i skjulte lag og utgangslaget har behov for å beregne den deriverte av utgangssignalet i tillegg til selve utgangssignalet. Den deriverte av utgangssignalet fra et nevron benyttes i beregningene av δ i forbindelse med opplæring av nettet. Bumpkretsen beskrevet av [Delbrück], som er en variasjon av transkonduktansforsterkeren, har nettopp disse egenskapene.

Det er mulig å benytte utgangen fra en CMOS inverter som utgang fra nevronet. Å beregne den deriverte av signalet fra en CMOS inverter vil derimot by på problemer. [Borgstrom] har benyttet en vanlig CMOS inverter som utsignal fra nevronet. Opplæringen forgår derimot ikke inne på den integrerte kretsen.

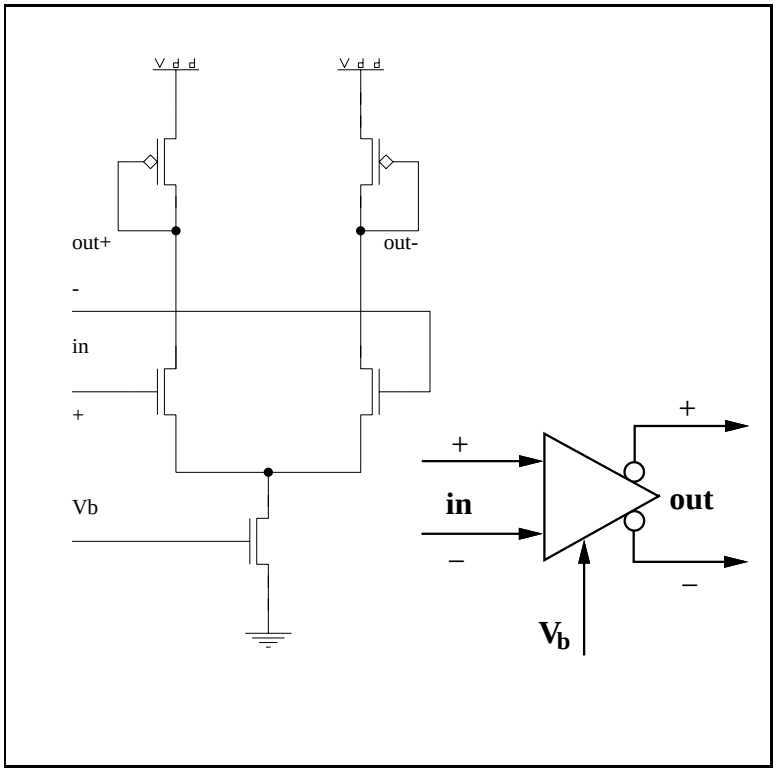
Bumpkretsen er derimot yppelig siden den gir oss derivasjonssignalet på en meget elegant måte. Den består av et differensielt par med en enkel korrelator på toppen, som vist i figur 4.2. Det er korrelatordelen i bumpkretsen som sørger for derivasjonen i utgangspunktet som et strømsignal. Når de to inngangssignalene er forskjellige vil derivasjonsstrømmen være tilnærmet null. Når inngangssignalene er like, altså når $in \approx 0$, vil derivasjonsstrømmen være "stor". Utgangsstrømmen for derivasjonssignalet er gitt ved [Delbrück]

$$I'_{out} = \frac{I_b}{2} sech^2(\kappa \frac{V_{in+} - V_{in-}}{2V_T}), \tag{4.1}$$

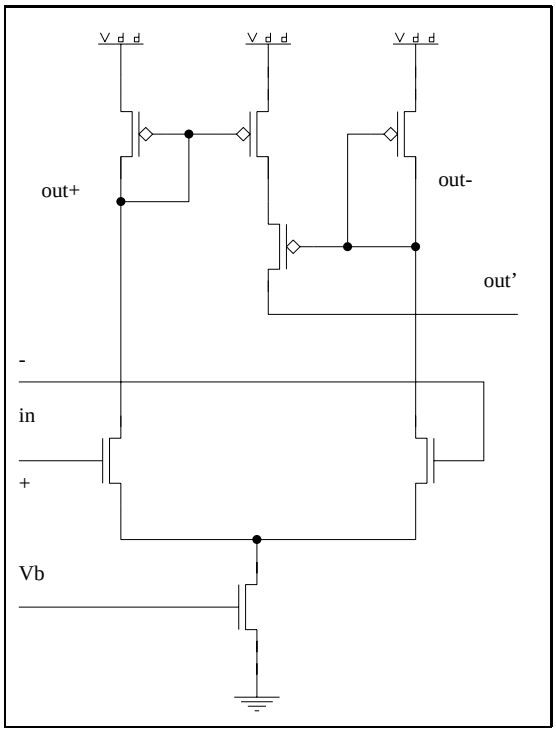
som gir en kurve sentret om $(V_{in+} - V_{in-}) = 0$ og med en form slik som kurven o' i figur 2.2. $Sech^2$ er nettopp den deriverte av $tanh$. V_T er den termiske spenningen og er omtrent lik 25mV.

Utgangsstrømmen fra transkonduktansforsterkerdelen (aktiviseringsfunksjonen) er gitt ved

$$I_{out} = I_b \tanh(\kappa \frac{V_{in+} - V_{in-}}{2V_T}) \tag{4.2}$$



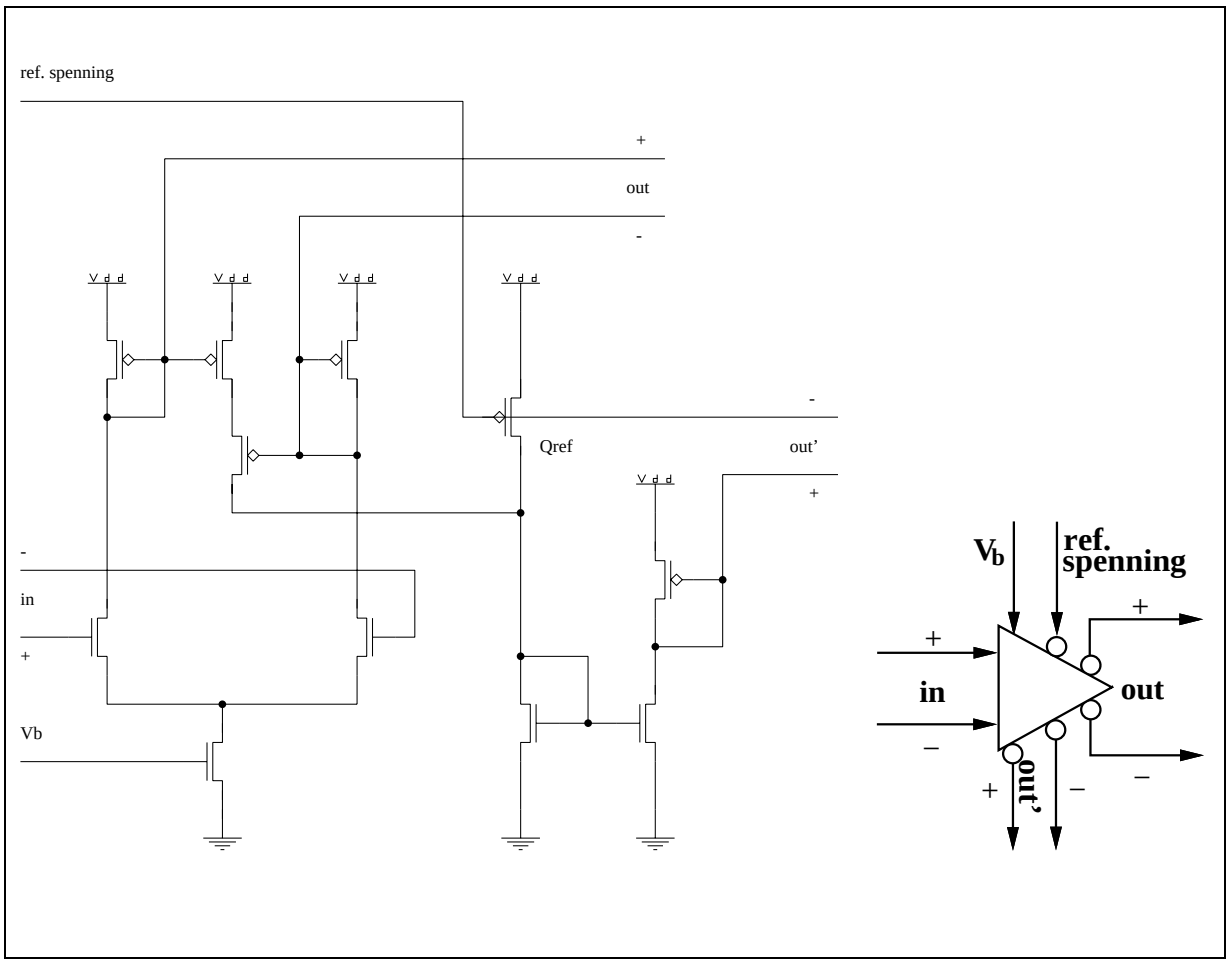
Figur 4.1: Kretsskjema og et et enkelt symbol for transkonduktansforsterkeren med diodeutganger. Merk at utsignalet out er et differensielt spenningssignal. Kretsen er egentlig kun et differensielt par med diodekoblinger på toppen.



Figur 4.2: Kretsskjema for bumpkretsen. Kretsen består av et vanlig differensielt par med en korrelator på toppen som gir utsignalet bump= I'_{out}. Bumpkretsen er en variasjon av transkonduktansforsterkeren.

og tas ut som et differensielt signal og vil da være utgangssignalet o_j fra et nevron.

Inngangssignalet til et nevron i et nett er $(net_j + \Theta_j)$ hvor net_j er summen av de veide inngangssignalene til nevronet, og θ_j er nevronets terskel. $(net_j + \Theta_j)$ vil være representert med en positiv og en negativ komponent. Siden det er en transkonduktansforsterker som gir sigmoidfunksjonen, vil inngangen V_{in+} på transkonduktansforsterker være gitt ved $(net_j + \Theta_j)_+$ etter en konvertering fra strøm til spenningssignal. Tilsvarende vil V_{in-} være gitt ved $(net_j + \Theta_j)_-$.



Figur 4.3: Kretsskjema og et enkelt symbol for transkonduktansforsterkeren med differensiell utgang for det vanlige transkonduktansforsterkersignalet out og for bumpsignalet out'. Signalet ref.spenning setter en referansestrøm slik at bumpsignalet kan konverteres til et differensielt spenningssignal out'.

Bumpsignalet er i utgangspunktet et strømsignal. For videre distribusjon av signalet har jeg imidlertid behov for å representere det som et differensielt spenningssignal. Et strømsignal kan enkelt konverteres til et spenningssignal ved hjelp av en diodekobling. I tillegg er det behov for en referanse som skal representere den negative signalkomponenten i det differensielle signalet. Bumpsignalet stabiliserer seg nær 0A når transkonduktans-

forsterkerens (bumpkretsens) to innsignaler er forskjellige. Det vil derfor by på problemer å konvertere bumpsignalet direkte til en spenning. Jeg har derfor valgt å bruke en referansespenning *ref.spenning* som i tillegg setter en referansestrøm. Referansestrømmen summerer jeg enkelt med bumpsignalet ved hjelp av Kirchhoffs strømlov. Det strømsignalet jeg får etter summeringen kan jeg konvertere til et spenningssignal uten problemer siden strømsignalet alltid vil være større enn 0A. Til slutt konverterer jeg dette spenningssignalet som blir den positive signalkomponenten slik at det blir et signal i forhold til V_{DD} siden den negative signalkomponenten (*ref.spenning*) også er i forhold til V_{DD}.

Kretsskjema og et enkelt symbol for transkonduktansforsterkeren (bumpkretsen) med differensiell utgang for det vanlige transkonduktansforsterkerensignalet *out* og bumpsignalet *out'* er vist i figur 4.3.

Det er viktig at bumpsignalet på differensiell form blir nær nok null (på positiv side) når den deriverte av utgangen fra et nevron skal være null. Når den deriverte av utgangen fra et nevron i et nett (bumpstrømmen) har stabilisert seg nær null, betyr det at nevronet har stabilisert seg som "av" eller "på". Dersom det differensielle bumpsignalet ikke blir nær nok null, betyr det at nettet sannsynligvis vil få problemer med å konvergere.

Referansesignalet bør settes eksternt. Den bør stilles slik at strømmen denne spenningen gir ikke blir stor i forhold til strømsignalet ut av korrelatordelen av bumpkretsen (bumpstrømmen).

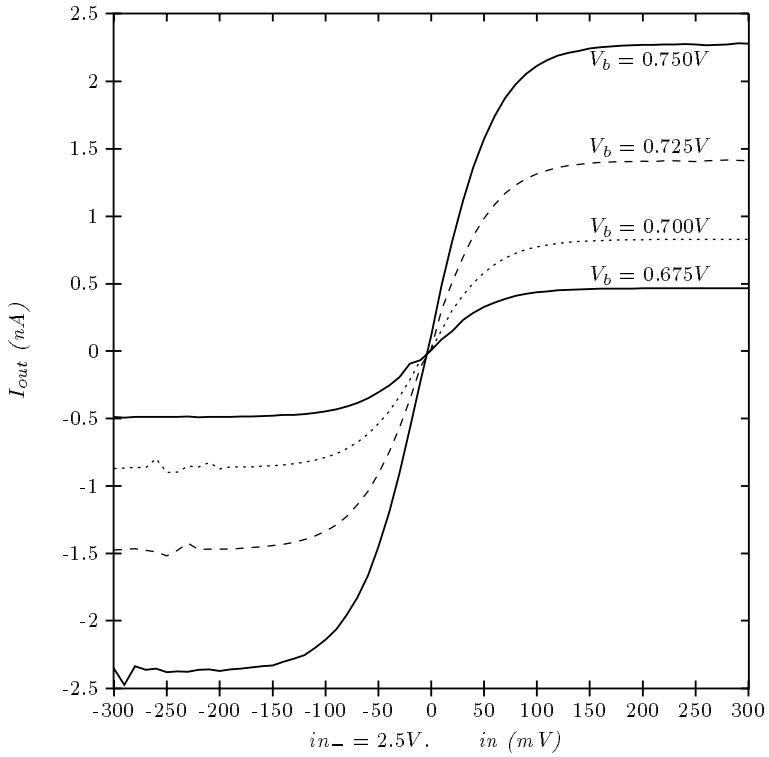
4.2.1 Målinger

Jeg har gjort målinger på en wide-range transkonduktansforsterker, både av det vanlige transkonduktansforsterkersignalet I_{out} og av bumpsignalet *bump* (I'_{out}). Kretsen i figur 4.2 er utvidet med noen strømspeil slik at den får en wide-range transkonduktansforsterker strømutgang I_{out}. Utgangstransistorene for begge signalene har jeg gjort forholdsvis store for ikke å bli plaget av stor stømlast. Under målingene lot jeg inngang in_- være fast og varierte in_+ i forhold til in_- fra -0.3V til 0.3V i steg av $10mV$ ved ulike forspenninger.

Figur 4.4 viser utsignalet I_{out} fra transkonduktansforsterkeren i *chip*#4 som funksjon av *in* ved forspenningene $V_b = 0.675V, 0.700V, 0.725V$ og $0.750V$. Figur 4.5 viser bumpsignalet som funksjon av *in* for de samme forspenningene, også disse målingene er utført på *chip*#4. Vi ser at kurvene for både I_{out} og *bump* er ønsket overføringsfunksjon. Vi kan også få et inntrykk av den eksponentielle karakteristikken for begge signaler ved å betrakte forskjellen i signalstørrelsen ved de ulike forspenningene.

Figur 4.6 viser I_{out} og *bump* for $V_b = 0.75V$. Firkanter representerer målinger, mens de heltrukkene kurvene representerer beregninger etter henholdsvis likning 4.2 og 4.1. Jeg har da valgt passende verdier av I_b og κ. Jeg valgte $\kappa = 0.77$. Jeg her derimot valgt forskjellige verdier for I_b for I_{out} og *bump* ved de teoretiske beregningene, siden størrelsen av de to signalenes utgangstransistorer ikke er den samme. Målingen viser at det lineære området for I_{out} er ca. 60mV. Bump-signalet er utsatt for litt støy på vei fra "store" verdier til små verdier. Det skyldes at amperemetret skifter måleområde fra nA til pA.

Jeg har gjort de samme målingene på fire kretser med noen forskjeller i resultatene, forskjeller som primært skyldes transistorvariasjoner og Early-effekten. Disse forskjellene kan være forholdsvis store, men selve karakteristikken for signalene er de samme. Figur

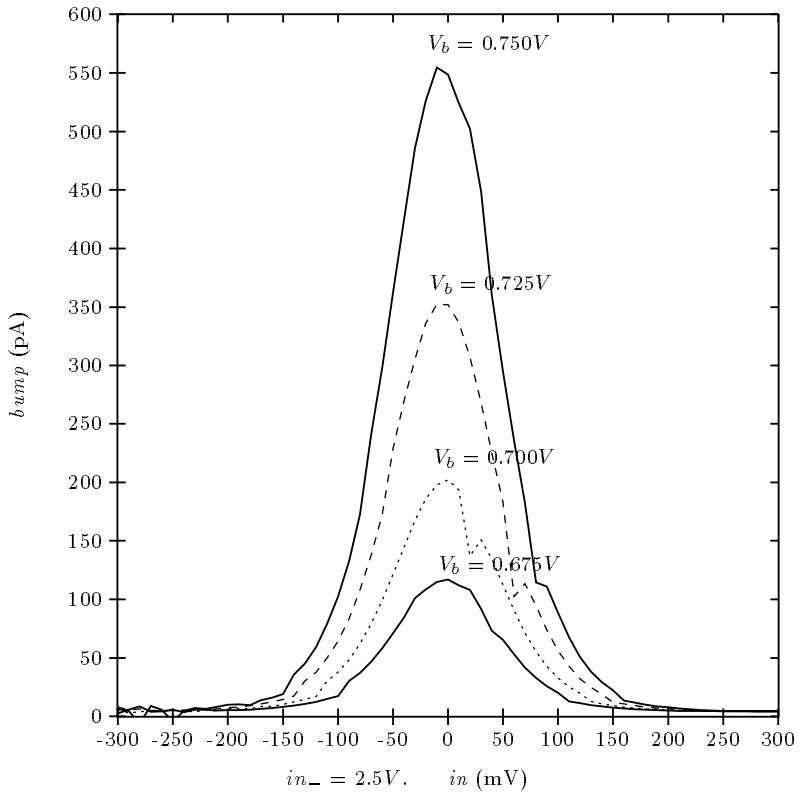


Figur 4.4: Strømutgangen I_{out} fra en wide-range transkonduktansforsterker ved 4 forspenninger

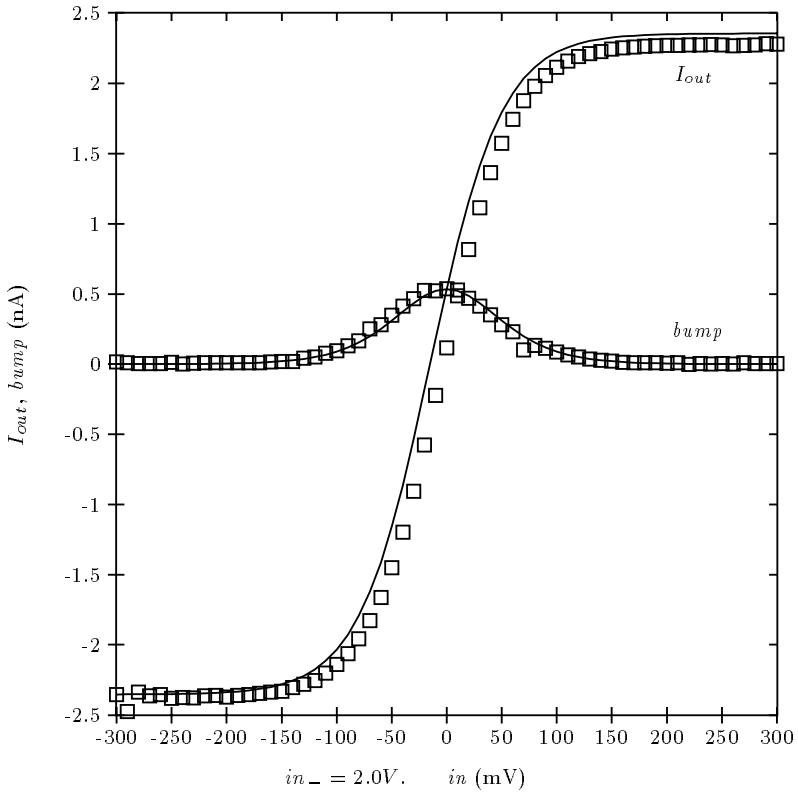
4.7 viser utsignalet I_{out} fra transkonduktansforsterkeren for tre kretser. De tre kurvene er forskjøvet et forskjellig antall mV lags x-aksen. Forskyvningen er for $chip\#1 \approx 20mV$, for $chip\#4 \approx 4mV$ og for $chip\#5 \approx 10mV$. Det vil også opptre forskyvninger av bump-signalet. Forskyvningen av bump-signalet i figur 4.5 er ca. 10mV. Forskyvningen av I_{out} og *bump* sees godt i figur 4.6. Akkurat hvor mye er derimot vanskelig å se av figuren. Forskyvningene av I_{out} og *bump*-signalene langs x-aksen skyldes transistorvariasjoner.

Avviket mellom positiv og negativ asymptote for I_{out}-signalene varierer også, fra ca. 4% for $chip\#4$ til ca. 35% for chip#1. Transistor variasjoner og Early-effekten gjør at speilene i transkonduktansforsterkeren ikke reflekterer strømsignalene 100% korrekt slik at absoluttverdien av negativ og positiv asymptote i I_{out}-signalet ikke blir identiske.

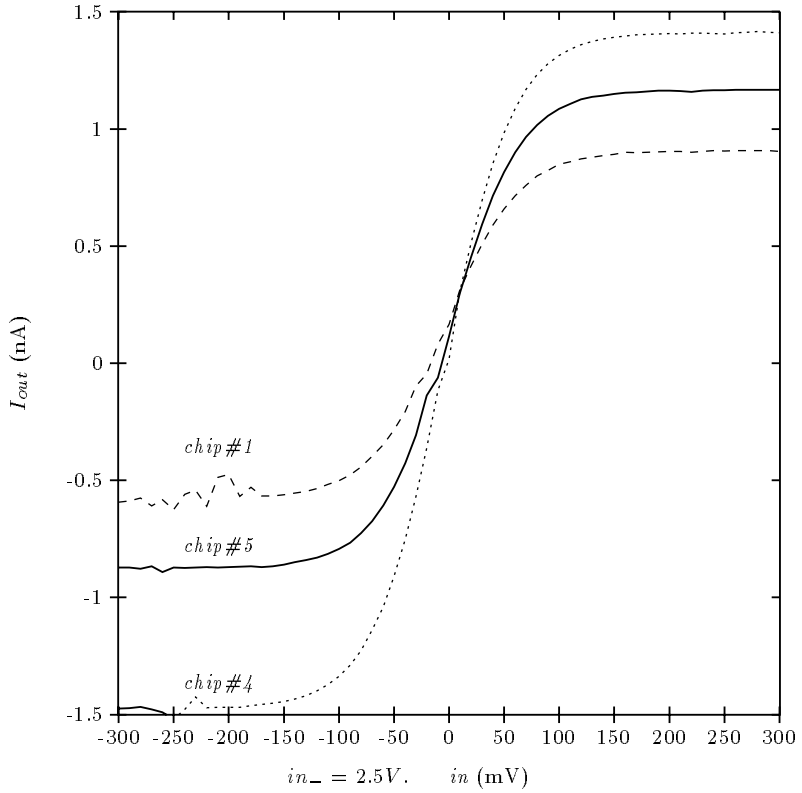
Valg av in_- skal i følge likning 4.1 og 4.2 ikke ha innvirkning verken på I_{out} eller *bump*. Målinger viser derimot at det er en liten endring av I_{out} og *bump* ved en endring av in_-. Figur 4.8 viser I_{out} når $in_- = 2.0V, 2.5V$ og $3.0V$. Likning 4.1 og 4.2 tar ikke hensyn til Early-effekten. Ved lav spenning på in_- og in_+, vil det være stor forskjell i spenningen V_{ds} over de to transistorene i hvert strømspeil, og Early-effekten vil få stor innvirkning. Av figur 3.2 ser vi at det kan være en betydelig forskjell i strøm gjennom en transistor ved varierende spenning V_{ds} for en gitt spenning V_{gs}. Det er dette som skjer ved lav spenning for in_- og in_+. Ved en høy spenning på in_- og in_+ vil spenningsforskjellen i V_{ds} for de to transistorene i strømspeilene reduseres, og Early-effekten vil få redusert innflytelse.



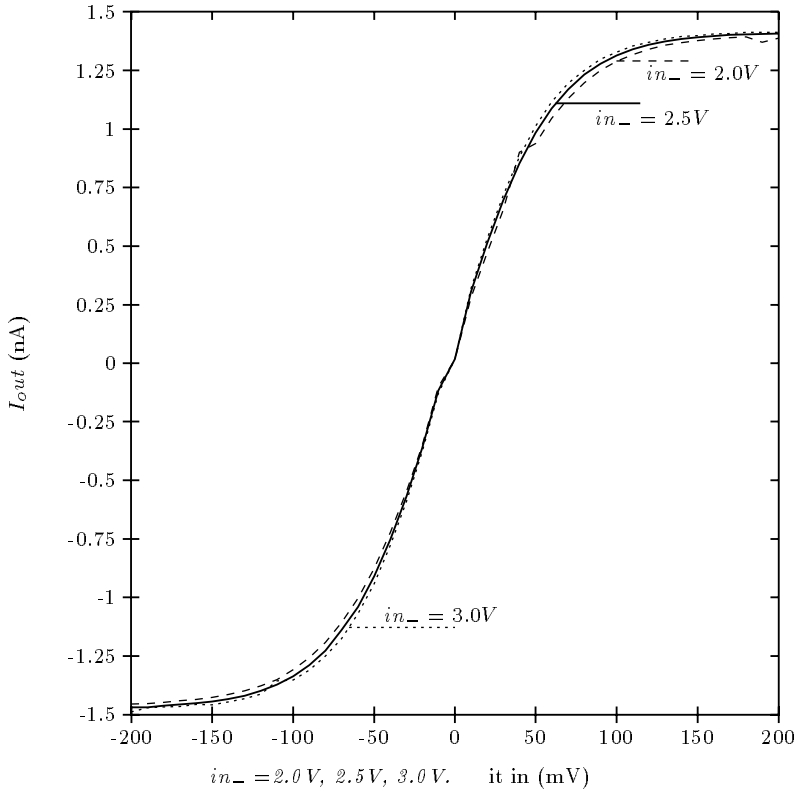
Figur 4.5: Bumputgangen fra en wide-range transkonduktansforsterker ved 4 forspenninger. Merk at bumpsignalet bump er den deriverte av sigmoidsignalet I_{out}.



Figur 4.6: Utgangene I_{out} og bump fra en wide-range transkonduktansforsterker. Firkanter representerer målinger, mens heltrukkene kurver er teoretiske beregninger. Merk avvikene mellom asymptotene for målingene av I_{out} og forskyvningene av de målte signalene langs in-aksen. Avvikene skyldes transistorvariasjoner og Early-effekten.



Figur 4.7: Strømutgangen I_{out} fra 3 wide-range transkonduktansforsterkere ved forspenning $V_b = 0.725V$. Det er forholdsvis store variasjoner mellom kretsene. Variasjonene skyldes i stor grad transistorvariasjoner.



Figur 4.8: Strømutgangen I_{out} fra en wide-range transkonduktansforsterker med forspenningen $V_b = 0.725V$, men med ulike valg av arbeidsområder for inngangssignalene. Early-effekten er årsaken til variasjonene ved ulike valg av arbeidsområder.

4.2.2 Oppsummering og konklusjon

Målingene av utsignalene fra transkonduktansforsterkeren ga overføringsfunksjoner med de samme karakteristikker som de teoretisk beregnede funksjonene. Imidlertid viser målingene en forskyvning av utsignalene i forhold til innsignal (forskyvning langs x-aksen). Dessuten er det avvik mellom tallverdien av positiv og negativ asymptote for det vanlige transkonduktansforsterker-signalet (I_{out}). Avvikene fra de teoretiske beregningene skyldes transistorvariasjoner og Early-effekten.

Variasjonene mellom I_{out}-signalet vil variere fra transkonduktansforsterker til transkonduktansforsterker både innenfor en krets og fra krets til krets. Det samme gjelder *bump*-signalet.

Målet er at transkonduktansforsterkeren fungerer tilfredstillende som beregningselement som en del av et nevralt nett til tross for de randomiserte variasjonene uansett valg av krets med det nevrale nettet.

4.3 Multiplikatoren

Svært mange av beregningene i et nett vil være multiplikasjoner. Hver eneste vekt har to multiplikasjoner i tilknytning til seg slik som vist i figur 2.3 og 2.4. Vektene w_{ji} kan være positive eller negative. Den ene multiplikasjonen er $w_{ji}o_i$ som beregner bidraget til nevronets inngangssignal net_j. Siden o_i er et sigmoidsignal fra en transkonduktansforsterker, vil o_i kunne være positiv eller negativ. Den andre multiplikasjonen er $w_{ji}\delta_j$ som beregner endringen av vekten. δ_j er et uttrykk for feilen og kan derfor være positiv eller negativ. De resterende multiplikasjonene i figur 2.3 og 2.4 er også 4 kvadrants multiplikasjoner siden disse involverer δ_j bortsett fra den i forbindelse med terskelen Θ. Det er en 2 kvadrants multiplikasjon. Terskelen Θ skal imidlertid behandles som en vanlig vekt.

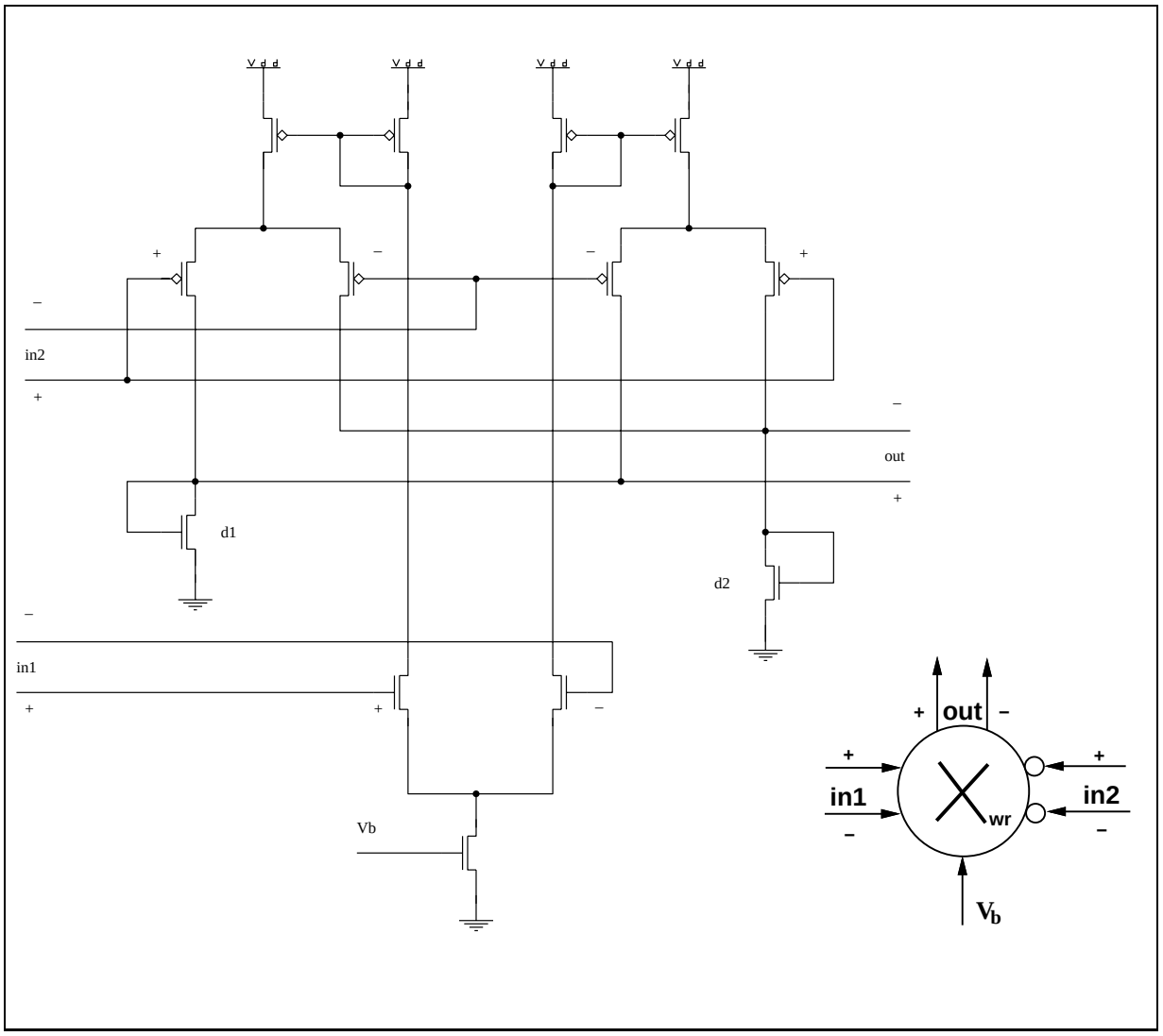
[Andreou] beskriver en translineær krets som beregner produktet av to strømsignaler. Kretsen utfører kun en 1 kvadrants multiplikasjon. Dersom jeg skulle velge å anvende den translineære multiplikatoren måtte jeg ha gjort konstruksjonen av nettet annerledes.

Jeg har derfor valgt å bruke en 4 kvadrants Gilbert multiplikator slik som vist i figur 4.9 ved all multiplikasjon. Den avviker noe fra en vanlig Gilbert multiplikator siden utsignalet er et differensielt spenningssignal i motsetning til den vanlige Gilbert multiplikatoren hvor utsignalet er et strømsignal. Figur 4.9 viser også et enkelt symbol for multiplikatoren. Boblene på symbolets inngang in2 betyr at disse inngangstransistorene er av p-type. Gilbert multiplikatoren er nært beslektet med transkonduktansforsterkeren. Inngangssignalene til begge beregningselementene er differensielle spenningssignaler.

Strømmen I_{out} ut av en Gilbert multiplikator er gitt ved

$$I_{out} = I_b \tanh(\kappa \frac{V_{in1+} - V_{in1-}}{2V_T}) \tanh(\kappa \frac{V_{in2+} - V_{in2-}}{2V_T}), \tag{4.3}$$

hvor V_T er den termiske spenningen og er omtrent lik 25mV. I_{out} er derfor lineær kun innenfor et begrenset område for de to differensielle spenningsdifferansene. Multiplikatoren er en såkalt wide range-versjon hvor de to nivåene av differensielle par er isolert fra hverandre.



Figur 4.9: Kretsskjema og et enkelt symbol for en wide-range Gilbert multiplikator med diodeutganger. Merk at utsignalet out er et differensielt spenningssignal. Det er også aktuelt å la utsignalet være et differensielt strømsignal ved å fjerne de to diodekoblingene d1 og d2 på utgangen.

Det medfører at denne versjonen ikke er begrenset av V_{min}-problemet som i enda sterkere grad begrenser en Gilbert multiplikator enn en transkonduktansforsterker [Mead]. Dersom en vanlig Gilbert multiplikator skal anvendes, må følgende kriterium være oppfylt

$$max(V_{in2+}, V_{in2-}) > min(V_{in1+}, V_{in1-}). \tag{4.4}$$

Vi kan ikke uten videre anta at kriterium 4.4 er oppfylt for multiplikatorene som skal anvendes i et nevralt nett. Arbeidsområdene for inngangsspenningene kan være såpass begrenset at kriterium 4.4 vil være oppfylt dersom arbeidsområdene til V_{in2+} og V_{in2-} plasseres et godt stykke over arbeidsområdene til V_{in1+} og V_{in1-} ved å legge inn noen ekstra diodekoblinger. Det lar seg gjøre fordi det egentlige inngangssignalet er en strøm, ikke en spenning. Strømmen blir konvertert til en spenning ved hjelp av en diodekobling eller flere diodekoblinger i kaskade for å plassere arbeidsområdet lengre fra *GND* eventuelt V_{DD}.

Nå er jeg i første omgang ikke helt sikker på at arbeidsområdene for inngangsspenningene vil være så begrenset i en realisert krets. Jeg har derfor valgt en "wide-range" Gilbert multiplikator. Ulempen med den er at den introduserer en del ekstra transistorer noe som fører til større arealforbruk per multiplikator.

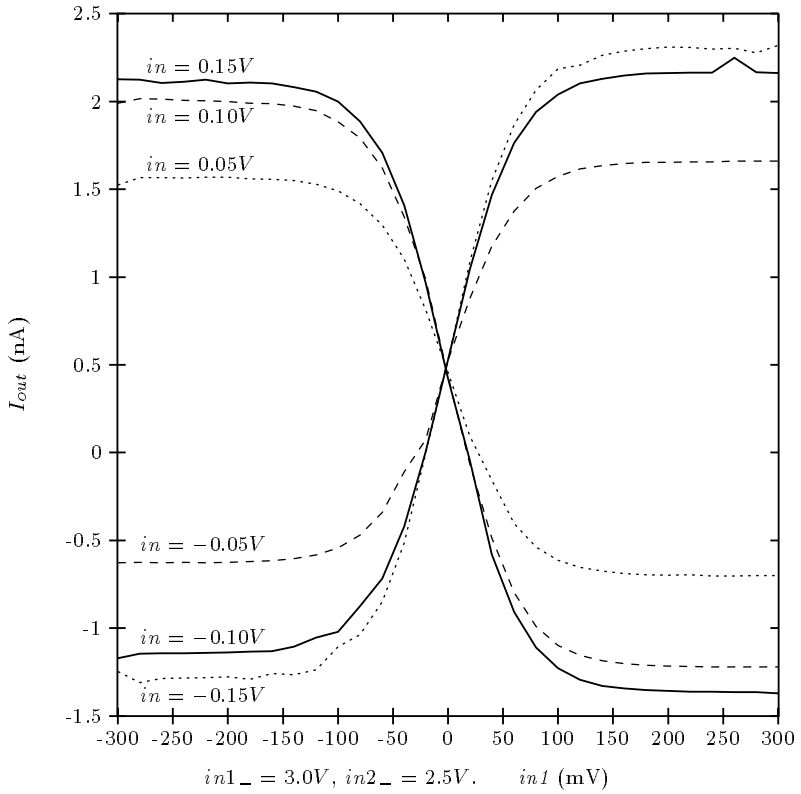
Jeg har også vurdert behovet forspenningstransistor til multiplikatoren, men har valgt å ha med forspenningstransistor i første omgang. Et realisert nett vil mest sannsynlig ha behov for en del justeringer med hensyn på signalstørrelse i de ulike elementene i nettet. Forspenningstransistor til multiplikatoren vil da være nødvendig.

Utgangsstrømmen kopieres via diodekoblingene $d1$ og $d2$ på utgangen, eller taes ut direkte ved å ikke ha diodekoblingene på utgangen. Begge alternativer er aktuelt. Dersom utgangssignalet kun skal videre til en strøminngang, er det enklest å ta ut utgangsstrømmen direkte. Ellers er diodekoblingene på utgangen mest hensiktsmessige siden det da er enkelt å lage så mange strømkopier som det er behov for. Multiplikatorens utgang er på differensiell form, altså representert som to komponenter.

I følge [Mead] er det mange fordeler med en wide-range Gilbert multiplikator frem for en vanlig, men på den andre siden er det enda flere transistorer som må balansere slik at det skulle være flere muligheter for at wide-range versjonen skulle bli ubalansert. I praksis viser det seg at wide-range versjonen ikke har flere uheldige effekter pga. transistor variasjoner enn det den vanlige versjonen har.

Gilbert multiplikatoren har en lineær karakteristikk kun over et meget begrenset område. Det er viktig at spenningsdifferansene på inngangene til multiplikatoren i størst mulig grad er av en slik størrelse at multiplikatorene opereres i det lineære området. At multiplikatorene er ulineære har også sine fordeler. Det vil ikke oppstå problemer slik som "overflow error".

Dersom det skulle passe bedre inn i sammenhengen av en større delkrets, kan vi la inngangene til in1 være av p-type og dermed også gjøre om alle andre transistorer fra n-type til p-type og omvendt. Det kan generelt gjøres for alle kretselementer der det måtte være mest hensiktsmessig.



Figur 4.10: Strømutgangen I_{out} fra en wide-range Gilbert multiplikator som funksjon av $in1$ og med konstante verdier for $in2$. Vi ser at signalene er forskjøvet langs $in1$-aksen, og at det er avvik mellom asymptotene. Avvikene skyldes transistorvariasjoner og Early-effekten spesielt i multiplikatorens mange strømspeil. Vi ser at multiplikatorens lineære område er 70-80mV.

4.3.1 Målinger

Jeg har gjort målinger av strømsignalet I_{out} fra en wide-range Gilbert multiplikator. Multiplikatoren jeg har utført målingene på er som vist i figur 4.9, men med ekstra speil på utgangen *out* slik at multiplikatoren blir en standard wide-range Gilbert multiplikator med strømutgang I_{out}. Under målingene varierte jeg $in1$ fra -0.3V til 0.3V i steg på 20mV. Figur 4.10 viser plot av målingene når in2 var -.15V, -.10V, -.05V, 0.05V, 0.10V og 0.15V ved en forspenning $V_b = 0.725V$. Målingene ble utført på *chip*#4. Vi kan se av figuren at det lineære området for signalene er ca. 70mV-80mV. Det er dette området som i størst grad bør være arbeidsområdet for multiplikatorene i et nevralt nett.

Tilsvarende som for transkonduktansforsterkeren har også utsignalet I_{out} fra multiplikatoren avvik fra teoretisk beregnet signal. Signalene er forskjøvet ca. 2mV langs x-aksen, og det er et avvik på ca. 60-70% mellom signalenes negative og positive asymptote for målingene utført på *chip*#4.

4.3.2 Forbedringer

Gilbert multiplikatorens og transkonduktansforsterkerens lineære områder er små. Det kan spesielt være et problem ved anvendelse av multiplikatoren. Det lineære området for

transkonduktansforsterkeren er ca. $60mV_{pp}$. Det eksisterer flere teknikker for å øke det lineære området for en transkonduktansforsterker slik som kapasitiv divisjon og source degenerering [Watts]. En transkonduktansforsterker med source degenerering har ekstra diodekoblede transistorer, en på hver side i det differensielle paret. Ulempen er en reduksjon i transkonduktansforsterkers operasjonsområdet og en økning i termisk støy [Watts]. Ved å legge inn en diodekobling på hver side i det differensielle paret, vil det lineære området økes til $144mV_{pp}$.

Det som vil være mest interessant, vil være å øke Gilbert multiplikatorens lineære område ved å legge inn diodekoblinger i alle de differensielle parene i multiplikatoren. Spørsmålet er om multiplikatorens operasjonsområde blir redusert for mye. Jeg har ikke implementert en multiplikator med ekstra diodekoblinger i de differensielle parene, men det kan være svært interessant å studere en slik implementasjon nærmere.

4.3.3 Oppsummering og konklusjon

Utsignalet fra Gilbert multiplikatoren har en karakteristikk slik som ønsket, men med noe avvik fra teorien slik som for transkonduktansforsterkeren. Det vesentlige er hvorvidt multiplikatorens lineære området er tilstrekkelig stort ved bruk av multiplikatoren i et nevralt nett.

4.4 Hukommelseselementet

Lagringen og endringen av vektene w_{ji} er en sentral oppgave i nettet. Til det bruker jeg analoge hukommelseselementer av floating gate-typen. De programmeres (endres) ved bestråling av UV-lys. Siden jeg benytter en differensiell signalrepresentasjon, vil det for hver vekt være behov for to hukommelseselementer, en for å lagre den negative og en for å lagre den positive komponenten av signalet.

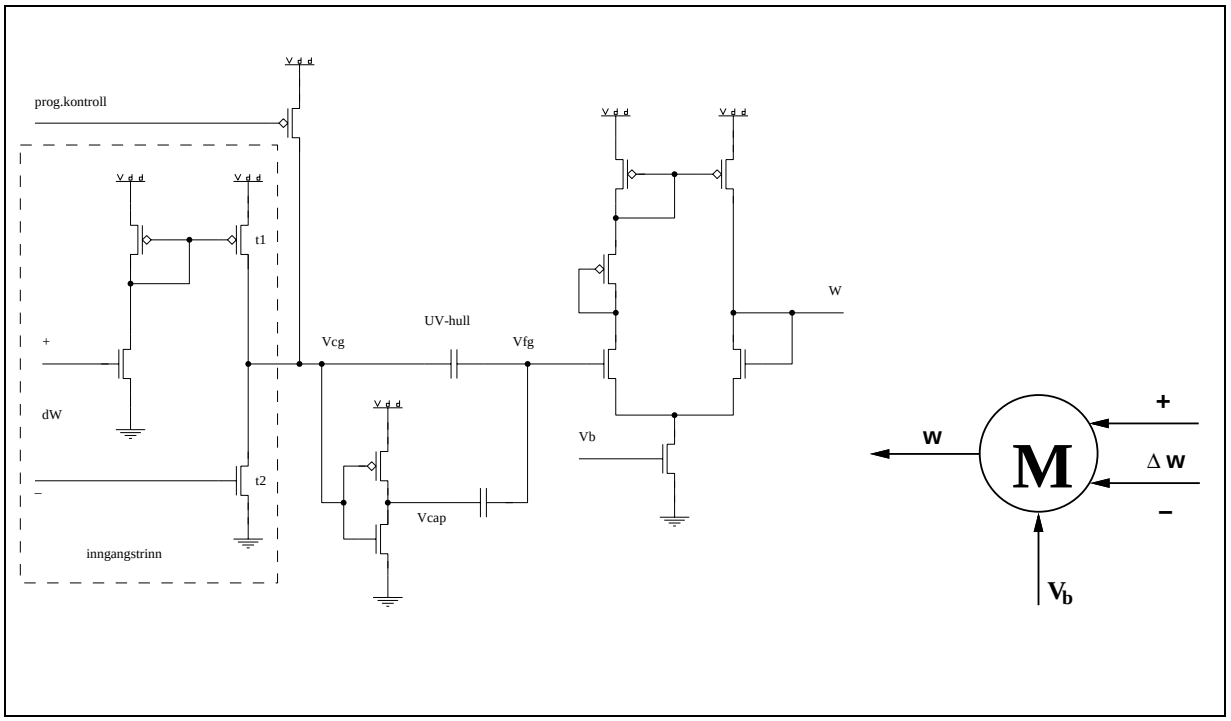
Det er endringen Δw_{ji} av en vekt w_{ji} etter presentasjon av et inngangsmønster p som GDR beregner. Endringen kan være positiv eller negativ. Endringen av en vekt w_{ji} er gitt i likning 2.4 og beregnes av en multiplikator som får δ_j og o_i som differensielle inngangssignaler. Parameteren η representeres av strømmen I_b igjennom multiplikatorens forspenningstransistor. Endringen Δw_{ji} skal justere verdien av vekten w_{ji}. Det er altså ikke endringen Δw_{ji} alene som blir ny verdi.

4.4.1 UV-strukturen

Hukommelseselementet er bygd opp rundt UV-strukturen beskrevet i avsnitt 3.2.1 og vist i figur 3.3. Det er en struktur som i stor grad er testet [Maher]. Det geniale med denne strukturen er at dersom elektroden V_{cap} påtrykkes det inverterte signalet av signalet på elektroden V_{cg}, vil den totale lastkapasistansen på floating gate noden være tilnærmet opphevet. Det er i tillegg til selve UV-strukturen derfor behov for en inverter for å invertere signalet fra floating gate noden. For at den totale lastkapasistansen i floating gate noden

skal bli tilnærmet opphevet, er det viktig at de to kondensatorene har like stor kapasistansen. Spenningen på floating gate noden V_{fg} distribueres videre med en spenningsfølger for å unngå ekstra lastkapasistans. Utgangen W fra spenningsfølgeren er utgangen fra hukommelseselementet.

UV-strukturen har ifølge [Maher] en svært rask programmeringstid siden lastkapasistansen i stor grad er redusert ved hjelp av inverteren og den ekstra kondensatoren C_{cap}.



Figur 4.11: Kretsskjema og enkelt symbol for hukommelseselementet. Den ekstra kondensatoren V_{cap} i samspill med inverteren fører til at det totalt sett blir en minimal last kapasistans på floating gate noden V_{fg}. Det fører til en økt programmeringshastighet. Det programmerte spenningsnivået på V_{fg} distribueres videre med en spenningsfølger som gir utsignalet W fra hukommelseselementet. Inngangstrinnet gir en stor forsterkning av innsignalet dW på kontroll noden V_{cg}. Signalet prog.kontroll anvendes til å gi kontroll noden et stabilt spenningsnivå etter en programmering.

Kretsskjema for hukommelseselementet er vist i figur 4.11. Inngangstrinnet til venstre i figuren er i praksis en del av utgangstrinnet fra en wide-range Gilbert multiplikator. Utgangen fra en multiplikator gir som kjent stor forsterkning av inngangssignalet. Dermed omgår jeg problemet med at floating gate noden aldri vil bli helt lik kontroll gate noden som en følge av likning 3.3 siden det med dette inngangstrinnet ikke lenger er noe mål at disse to nodene skal bli like.

Alle beregninger i det nevrale nettet vil foregå kontinuerlig. Endringer av vektene vil derfor beregnes også når nettet brukes til klassifisering, altså når kretsen ikke bestråles med UV-lys. UV-hullet er en vanlig kapasitiv kobling når det ikke belyses med UV-lys. For å stabilisere floating gate noden V_{fg} på det programmerte spenningsnivået har jeg derfor

lagt inn en ekstra p-transistor mellom V_{DD} og kontrollnoden V_{cg} slik at jeg kan bruke transistoren som en bryter og trekke V_{cg} opp mot V_{DD} når det ikke foregår en programmering. Under programmering vil den ekstra pull-up transistoren ikke være virksom.

4.4.2 Målinger uten UV-lys

Jeg har gjort målinger av hukommelseselementet i figur 4.11 uten bruk av UV-lys for å betrakte kontroll gate noden V_{cg} og dens inverterte V_{cap} ved varierende inngangsspenning dW. Dessuten har jeg undersøkt om floating gate noden V_{fg} blir påvirket av endringer på kontroll gate noden når hukommelseselementet ikke programmeres siden det er en kapasitiv kobling mellom V_{fg} og V_{cg}.

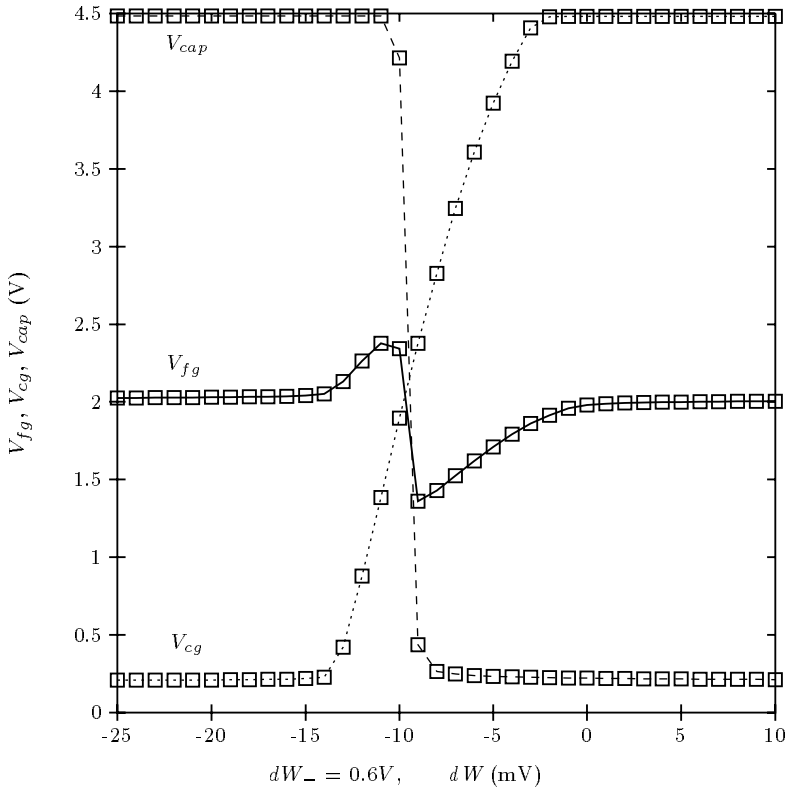
Jeg satt $prog.kontroll = 5V$ slik at transistoren som stabiliserer kontroll gate noden etter programmering var "av". Endringer av V_{cg} og V_{cap} vil da skje som følge av endringer av inngangen dW. Når hukommelseselementet brukes i en vekt, vil hukommelseselementets inngangstrinn i praksis være en del av en standard wide-range Gilbert multiplikator. Spenningene dW_- og dW_+ vil derfor variere en diode-offset fra GND (evt. V_{DD}), ca. 0.7V. Det er derfor mest interessant å la innspenningene ligge i dette området.

Jeg holdt $dW_- = 0.6V$ og varierte dW_+ mens jeg gjorde målinger uten UV-lys av V_{fg}, V_{cg} og V_{cap}. Figur 4.12 viser plot av målingene. Vi kan se at V_{cg} ligger nær GND når $dW < 0$ og nær V_{DD} når $dW > 0$. Det er ikke helt tilfelle siden transistorvariasjoner vil forskyve det relative nullpunktet langs dW-aksen noe. Vi ser at V_{cap} er den inverterte av V_{cg}. V_{cg} har en overgang fra nær GND til nær V_{DD} på ca. 15mV grunnet den store utgangsforsterkningen i inngangstrinnet. Vi ser at V_{fg} har en stabil spenning bortsett fra i området hvor V_{cg} stiger fra nær GND til nær V_{DD}. I dette avgrensede området vil V_{fg} ha et avvik fra det stabile nivået når $prog.kontroll = 5V$.

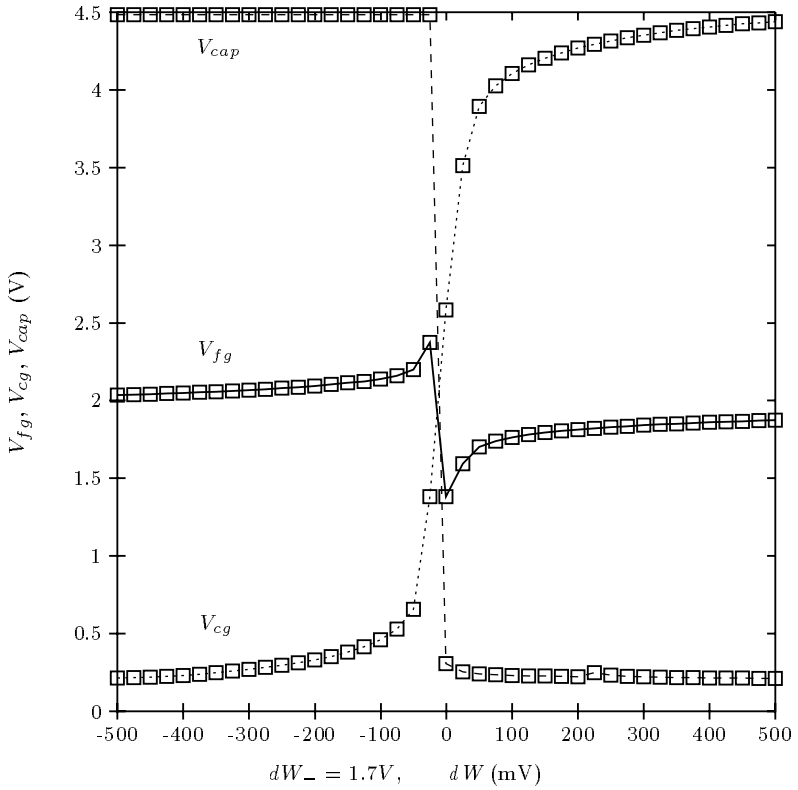
Setter vi $prog.kontroll = 0V$ og gjør samme måling på nytt, vil V_{cg} ligge stabilt nær V_{DD} og V_{cap} stabilt nær V_{DD} uansett endringer i dW. Hensikten med $prog.kontroll$ er nettopp å kunne kontrollere at V_{cg} og V_{cap} blir liggende på stabile spenningsnivåer etter programmering slik at V_{fg} ikke påvirkes gjennom den kapasitive koblingen, men blir værende helt stabil.

Figur 4.13 viser målinger av de samme nodene når $dW_- = 1.7V$. Vi ser at V_{cg} har en vesentlig mindre steil overgang fra lavt til høyt signal i motsetning til V_{cg} i figur 4.12. Legg merke til at i figur 4.12 varierer dW fra -25mV til 25mV, mens i figur 4.13 varierer dW fra -500mV til 500mV. En mindre steil overgang vil gjøre de enklere å programmere floating gate noden V_{fg} til ønsket spenningsnivå.

I både figur 4.12 og 4.13 ser vi at signalene V_{cg} og V_{cap} ikke er helt symmetriske. Spesielt er overgangen fra høyt til lavt signalnivå for V_{cap} vesentlig mer steil enn overgangen fra lavt til høyt signalnivå for V_{cg}. Det kommer av at inverteren og inngangstrinnet til hukommelseselementet ikke har samme forsterkning. Floating gate noden V_{fg} vil påvirkes av dette, og det medfører en ustabilitet i nodens signalnivå når V_{cg} og V_{cap} skifter signalnivå henholdsvis fra lavt til høyt og fra høyt til lavt eller omvendt. Det er mulig å skalere transistorene i inverteren og i inngangstrinnet slik at disse to delkretsene gir like stor forsterkning av et signal. Derimot vil transistorvariasjoner føre til at V_{cg} og V_{cap} neppe slår om fra et signalnivå til det motsatte samtidig. V_{cg} og V_{cap} blir dermed fremdeles ikke helt



Figur 4.12: Utgangene V_{fg}, V_{cg} og V_{cap} fra et hukommelseselement ved $dW = 0.6V$, $V_b = 0.725V$, prog.kontroll $= 5.0V$ og UV-lys av. Vi ser at kontroll gate signalet V_{cg} stiger fra lavt til høyt nivå i løpet av ca. 15mV, og at V_{cap} er den inverterte av V_{cg}. Floating gate noden er ustabil når prog.kontroll $= 5V$ som en følge av at V_{cg}- og V_{cap}-signalene er usymmetriske. Prog.kontroll $= 0V$ vil gi et stabilt floating gate signalnivå, og det er nettopp hensikten med dette kontrollsignalet.



Figur 4.13: Utgangene V_{fg}, V_{cg} og V_{cap} fra et hukommelseselement ved $dW = 1.7V$, $V_b = 0.725V$, prog.kontroll $= 5.0V$ og UV-lys av. Vi ser at overgangen fra lavt til høyt signalnivå for kontroll gate noden V_{cg} ikke er fullt så steil når arbeidsområdet for dW økes. Vi ser at V_{cg} ikke stabiliserer seg på samme signalnivå umiddelbart etter ustabiliteten som en følge av små og uønskete forskjeller i kapasistansen mellom nodene V_{cg} og V_{cap}.

symmetriske.

I figur 4.13 ser vi dessuten enda en effekt som påvirker floating gate noden. Vi kan av figuren se at det vil ta litt tid før V_{fg} stabiliserer seg på samme nivå etter omslaget i V_{cg} som det var før omslaget. Små og uønskete forskjeller i kapasistansen mellom nodene V_{cg} og V_{fg} og nodene V_{cap} og V_{fg} er årsaken til at V_{fg} ikke stabiliserer seg på samme spenningsnivå som før ustabiliteten umiddelbart etter ustabiliteten.

4.4.3 Programmering av hukommelseselementet - målinger med UV-lys

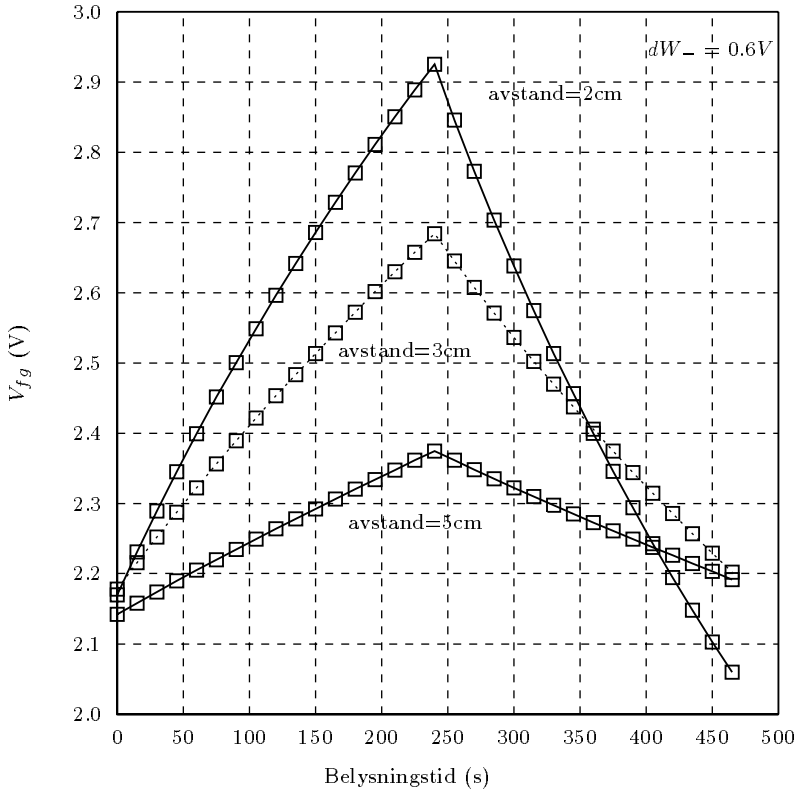
Jeg har programmert hukommelseselementet i figur 4.11 med å belyse floating gate noden med UV-lys. Lyskilden har vært UVP SpotCure. Jeg har påtrykt en differensiell spenning dW. Jeg lot så kretsen kretsen bli belyst med UV-lys et antall etterfølgende perioder med gitt lengde. Mellom hver belysningsperiode gjorde jeg så målinger av floating gate noden. Rett før hver programmeringsperiode satt jeg kontrollsignalet *prog.kontroll=5V* slik at kontroll noden ble regulert av inngangssignalet dW, mens jeg satt *prog.kontroll=0V* før jeg foretok målingen av floating gate noden slik at kontroll noden fikk et stabilt spenningsnivå nær V_{DD} så den ikke virket forstyrrende inn på floating gate noden under målingen av denne.

Et positivt signal dW vil gi en positiv endring av floating gate noden under en programmering, mens et negativt signal dW vil gi en negativ endring. Størrelsen av endringen bestemmes først og fremst av programmeringstiden og belysningseffekten (avstanden mellom lyskilde og krets). Andre faktorer har også en betydning slik som valg av arbeidsområde for dW, størrelsen av inngangssignalet dW og spenningsforskjellen mellom floating gate noden og kontroll noden.

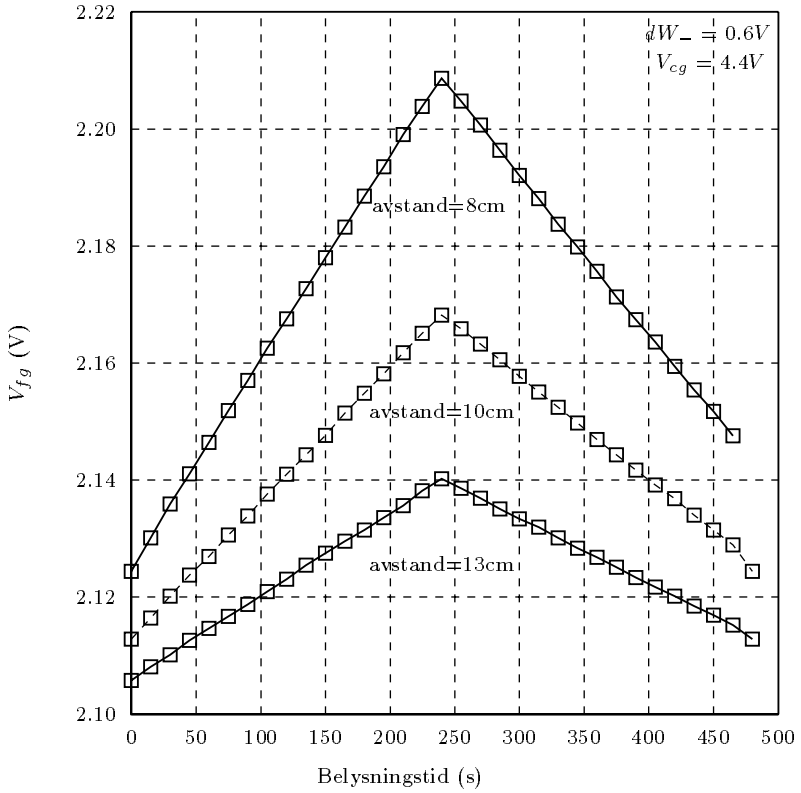
Figur 4.14 viser endringen av floating gate noden V_{fg} som en funksjon av belysningstiden. Jeg har valgt et arbeidsområde i subterskelområdet for inngangsspenningene, $dW_- = 0.6V$. De tre kurvene viser endringen av floating gate noden ved tre forskjellige avstander mellom lyskilde og krets. Langs aksen for belysningstiden har jeg fra 0s og frem til 240s påtrykt en positiv dW ($dW_+ = 0.5V$), og vi ser at endringen av floating gate noden er positiv for alle de tre kurvene. Jeg har deretter (fra 240s of frem til 480s) påtrykt en negativ dW ($dW = -0.5V$), og vi ser at endringen av floating gate noden er negativ for alle tre kurvene. Vi ser lett at endringen pr. tidsenhet øker når vi reduserer avstanden mellom kretsen og lyskilden som er et resultat av økende belysningseffekt.

Figur 4.15 viser tilsvarende målinger som i figur 4.14, men her er avstanden mellom krets og lyskilde økt betydelig. Vi ser at endringene pr. tidsenhet i figur 4.15 er vesentlig mindre enn endringene for kurvene i figur 4.14. Belysningseffekten har en eksponentiell sammenheng med avstanden mellom krets og lyskilde og vil ha innvirkning på endringens størrelse. Vi kan få et inntrykk av denne sammenhengen ved å betrakte forskjellen i endring mellom kurvene i figur 4.14 og 4.15.

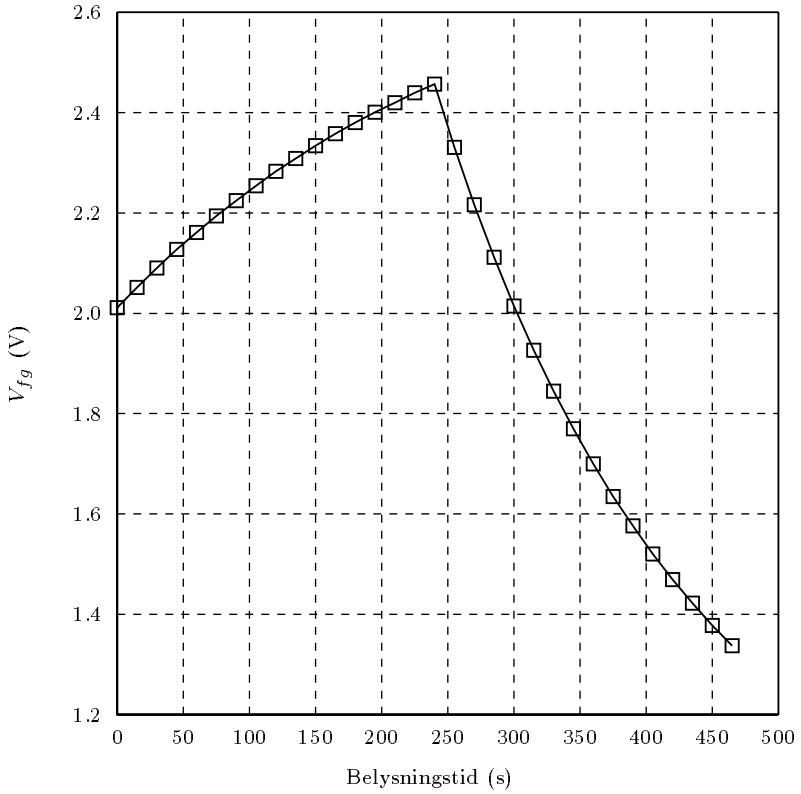
Når vi reduserer avstanden mellom krets og lyskilde oppnår vi en større belysningseffekt, som fører til en større fotoelektriske effekt [Streetman]. Fotoner som treffer kretsen vi overføre energi til elektroner slik at de frigjøres slik at enkelte isolerende lag blir ledende



Figur 4.14: Positiv og negativ endring av spenningsnivået på floating gate noden V_{fg} som en funksjon av belysningstiden ved tre forskjellige avstander mellom kretsen og lyskilden. Kretsen har et positivt inngangssignal $dW = 0.5V$ når endringen for de tre kurvene er positiv og et negativt inngangssignal $dW = -0.5V$ når endringen for de tre kurvene er negativ. Vi ser at avstanden mellom kretsen og lyskilden har stor betydning for størrelsen av endringen.



Figur 4.15: Positiv og negativ endring av spenningsnivået på floating gate noden V_{fg} som en funksjon av belysningstiden ved tre forskjellige avstander mellom kretsen og lyskilden. Kretsen har et positivt inngangssignal $dW = 0.5V$ når endringen for de tre kurvene er positiv og et negativt inngangssignal $dW = -0.5V$ når endringen for de tre kurvene er negativ. Vi ser at avstanden mellom kretsen og lyskilden har stor betydning for størrelsen av endringen.



Figur 4.16: Endringer av floating gate noden i hukommelseselementet som funksjon av belysningstiden ved stor belysningseffekt. Avstanden mellom krets og lyskilde er 1cm. Den fotoelektriske effekten gjør seg sterkt gjeldene på en uønsket måte og trekker kontroll noden V_{cg} fra 4.4V og ned til 2.85V når UV-lyset settes på. I første del av kurve er $dW = 1.1V$ som gir en positiv endring. Vi ser at endringens størrelse avtar sterkt ettersom floating gate noden nærmer seg spenningsnivået til kontroll noden som en følge av den ulinære motstanden i UV-hullet under UV-belysning. Andre del av kurven har $dW = 0.1V$ og gir en negativ endring. Vi ser her at endringen til å begynne med er svært stor som følge av en stor spenningsforskjell mellom V_{fg} og V_{cg}, men at endringen avtar etter hvert som V_{fg} nærmer seg GND.

og vi får en fotoelektrisk strøm. Den elektriske modellen for UV-strukturen i figur 3.4 viser to konduktanser, R_1 mellom V_{cg} og GND og R_2 mellom V_{cg} og V_{fg} som oppstår pga. av den fotoelektriske effekten når kretsen belyses med UV-lys. Den elektriske modellen forutsetter imidlertid ideelle driveegenskaper av kontroll noden V_{cg}. I vårt tilfelle viser det seg at driveegenskapene absolutt ikke er ideelle. Vi må derfor ta med i betraktningen av det også oppstår en konduktans R_3 parallelt med C_3 mellom V_{cg} og GND.

Reduseres avstanden tilstrekkelig, vil den fotoelektriske strømmen gjennom R_3 fra V_{cg} og til GND bli betydelig i forhold til strømmen som settes av inngangstrinnet i figur 4.11 som skal trekke kontroll noden V_{cg} mot GND for en negativ endring og mot V_{DD} ved en positiv endring. Dersom strømmen gjennom R_3 blir tilstrekkelig stor i forhold til strømmen som settes av inngangstrinnet, vil ikke strømmen fra inngangstrinnet være i stand til å trekke kontroll noden mot V_{DD}. Strømmen gjennom R_3 kan i værste fall bli så stor at kontroll noden vil trekkes nær GND selv om dW er positiv. Den store foto elektriske effekten vil da ødelegge beregningene fullstendig som en følge av svake driveegenskaper av kontroll noden.

Størrelsen av innsignalet dW bestemmer inngangstrinnets driveegenskaper av kontroll nodens spenningsnivå. En stor dW vil gi gode drive egenskaper av kontroll noden, mens en liten dW vil føre til at spenningsnivået på kontroll noden er ømfintlig for fotoelektrisk strøm.

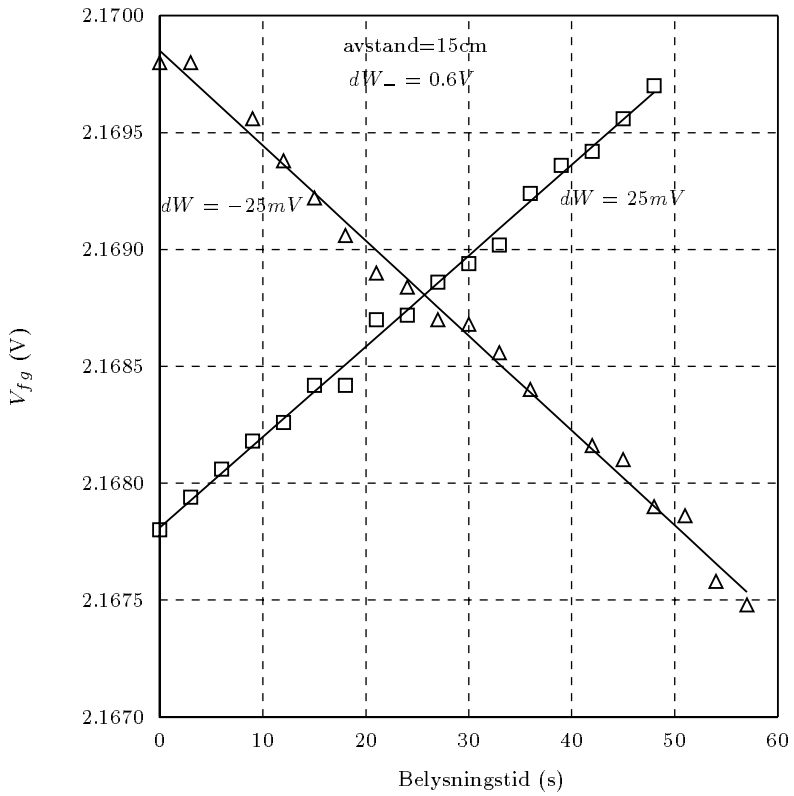
Kurvene i figur 4.14 og 4.15 har et forholdsvis stort inngangssignal dW. Når avstanden mellom krets og lyskilde blir liten, vil den fotoelektriske strømmen gjennom R_3 ha en slik størrelsesorden at inngangstrinnet ikke klarer å trekke kontroll noden nær V_{DD} (dvs. ca. 4.4V).

Når inngangsspenningen $dW = 0.5V$ og avstanden mellom krets og lyskilde er henholdsvis 5cm, 3cm og 2 cm, vil kontroll noden V_{cg} ha et spenningsnivå på henholdsvis 4.176V, 3.984V, 3.778V som en følge av en økende fotoelektrisk strøm gjennom R_3 ved økende belysningseffekt. Ved en liten dW, f.eks. 50mV, vil den fotoelektriske strømmen gjennom R_3 bli så stor at kontroll noden vil bli trukket helt ned mot GND når kretsen belyses med UV-lys ved de nevnte avstander mellom krets og lyskilde.

Figur 4.16 viseren tilsvarende kurve som kurvene i figur 4.14 og 4.15, men med en avstand mellom krets og lyskilde på bare 1cm. Ved en positiv endring (første del av kurven), vil størrelsen av den fotoelektriske strømmen gjennom R_3 ved UV-belysning føre til at kontroll noden trekkes ned til 2.85V.

I figur 4.16 ser vi også tydelig en effekt som følge av at konduktansen mellom kontroll noden og floating gate noden er ulineær og avhengig av spenningsforskjellen mellom de to nodene slik som beskrevet i avsnitt 3.2.2. Vi ser at den positive endringen i løpet av de første 240s er vesentlig mindre enn den negative endringen de etterfølgende 240s. Årsaken er at spenningsforskjellen mellom kontroll gate noden og floating gate noden er liten ved den positive endringen ($V_{cg} \approx 2.85$), mens den er stor ved den negative endringen ($V_{cg} \approx 0.5V$. Vi ser dessuten at både den positive og den negative endringen pr. tidsenhet reduseres etter hvert som spenningsnivået på floating gate noden nærmer seg spenningsnivået på kontroll gate noden.

Samme tedens kan vi også se i figur 4.14, men i figur 4.15 ser vi i svært liten grad denne tedensen. Årsaken er at den totale endringen i 4.15 er forholdsvis liten slik at



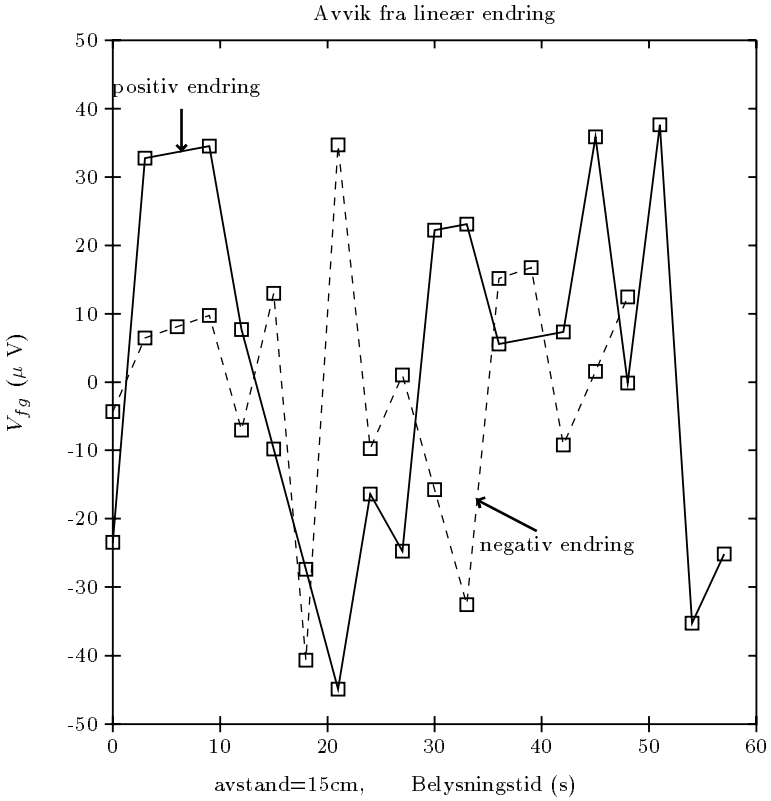
Figur 4.17: Positiv og negativ endring av floating gate noden ved små innspenninger henholdsvis $dW = 25mV$ og $dW = -25mV$. Firkanter og trekanter representerer målte verdier for henholdsvis positiv og negativ endring av floating gate noden. De heltrukkene linjene er regresjoner av de målte verdiene. Vi ser at både positiv og negativ endring er meget lineær når den totale endringen er liten og spenningsdifferansen mellom V_{fg} og V_{cg} er stor.

spenningsforskjellen mellom V_{cg} og V_{fg} reduseres i vesentlig mindre grad enn i 4.14 og 4.16.

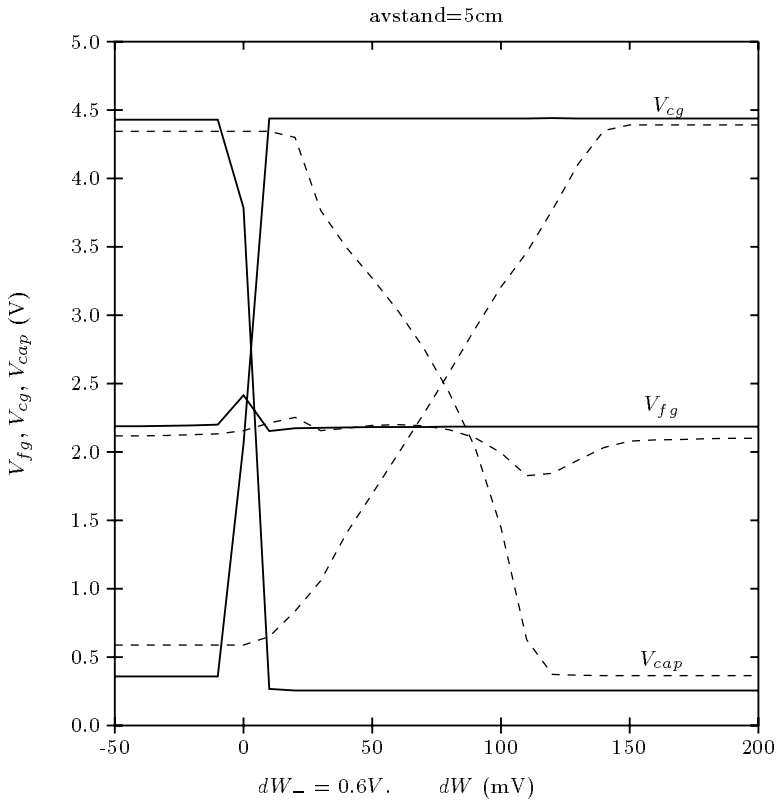
Når den totale endringen er liten kan vi som en god tilnærming anta at endringen er lineær, spesielt dersom spenningsforskjellen mellom V_{cg} og V_{fg} i tillegg er stor.

Figur 4.17 viser positiv og negativ endring av floating gate noden ved små innspenninger henholdsvis $dW = 25mV$ og $dW = -25mV$. Arbeidsområdet er fremdeles det samme, $dW_- = 0.6V$. En liten endring dW har en forholdsvis dårlig driveegenskap av kontroll noden. Under målingene i figur 4.17 brukte jeg et UV-filter for å filtrere bort det meste av det synlige lyset slik at den fotoelektriske effekten reduseres. I tillegg lot jeg avstanden mellom krets og lyskilde være stor, 15cm. Selv med nevnte tiltak ble kontroll noden trukket ned til 3.752V når kretsen ble belyst.

Selve formålet med målingene i figur 4.17 var å gi et vist inntrykk av hukommelseselementets oppløsning. Mellom hver måling har jeg programmert floating gate noden i 3 sekunder. Både den totale positive og den totale negative endringen er begge på ca. 2mV i løpet av et knapt minutt. I løpet av ca. 1mV endring har jeg utført 9 målinger. Figur 4.18 viser avvik mellom målte verdier og regresjonen for både den positive og den negative endringen i figur 4.17. Vi ser at avviket er maksimalt $\pm 40\mu V$. Endringen er for begge kurvene i figur 4.17 altså svært lineær.



Figur 4.18: Avvik mellom målte verdier og regresjonen for både den positive og den negative endringen av floating gate noden i hukommelseselementet. Avviket er lite i forhold til endringen.

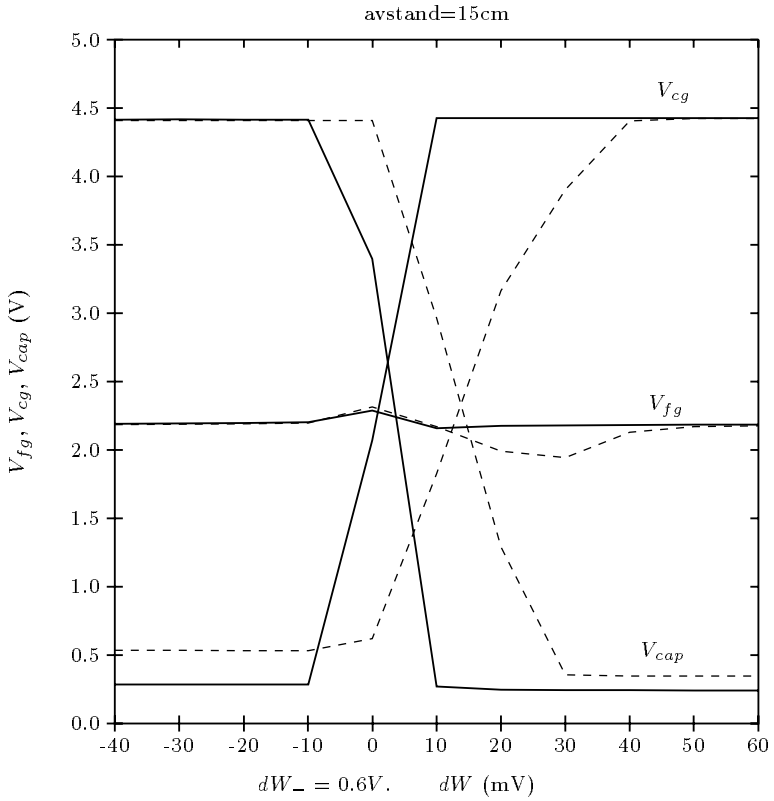


Figur 4.19: V_{fg}, V_{cg} og V_{cap} som funksjon av innsignalet dW både med og uten UV-lys og med $dW_{-} = 0.6V$ og avstanden mellom krets og lyskilde lik 5cm. Heltrukkene kurver er målinger uten UV-lys mens stiplede kurver er målinger med UV-lys. Vi ser de tre nodenes karakteristikker endrer seg dramatisk fra ingen UV-belysning til UV-belysning av kretsen som en følge at den fotoelektriske effekten.

Jeg har utført målinger for å betrakte endringen av driveegenskapene til inngangsspenningen dW av kontroll noden V_{cg} og dens inverterte V_{cap} som følge av belysningseffekten (avstanden mellom krets og lyskilde) og valg av arbeidsområde (dW_{-}).

Jeg varierte dW og målte nodene V_{fg}, V_{cg} og V_{cap} i figur 4.11. Jeg lot hele tide *prog.kontroll* være 5V slik at inngangen dW drev kontroll noden. Jeg utførte målinger både med og uten UV-lys. Når jeg belyste kretsen, dekket jeg den med et UV-filter slik at mest mulig av det synlige lyset ble filtrert bort. Figur 4.19 viser de tre nodene V_{fg}, V_{cg} og V_{cap} når $dW_{-} = 0.6V$ og avstanden mellom krets og lyskilde er 5cm. Heltrukkene kurver er målinger når kretsen ikke belyses med UV-lys, mens stiplede kurver er målinger når kretsen belyses med UV-lys. Målinger av de tre nodene er utført i steg av 10mV. Vi ser at driveegenskapene av V_{cg} og V_{cap} endrer seg dramatisk når kretsen belyses med UV-lys i forhold til driveegenskapene av de samme nodene når kretsen ikke belyses med UV-lys. Vi ser at det påtrykte innsignalet dW må være større enn 100mV for at den skal drive kontroll noden V_{cg} opp mot 5V.

Idéen med programmeringen av hukommelseselementet var at det skulle være en stor positiv spenningsdifferanse mellom V_{cg} og V_{fg} for en positiv endring av V_{fg} og en stor negativ spenningsdifferanse mellom V_{cg} og V_{fg} for en negativ endring. Dersom kriteriet



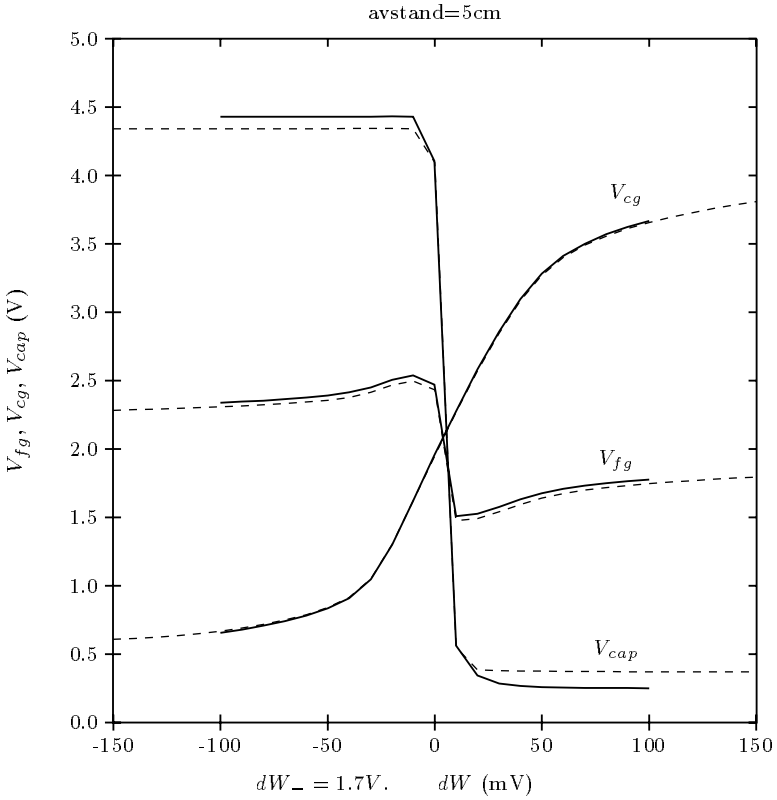
Figur 4.20: V_{fg}, V_{cg} og V_{cap} som funksjon av innsignalet dW både med og uten UV-lys, med $dW_{-} = 0.6V$ og en avstand mellom krets og lyskilde lik 15cm. Heltrukkene kurver er målinger uten UV-lys mens stiplede kurver er målinger med UV-lys. Vi ser de tre nodenes karakteristikker ikke endrer seg fult så mye ved en mindre belysningseffekt (større avstand mellom krets og lyskilde) som fører til at den fotoelektriske effekten blir mindre dominerende.

ikke oppfylles kan den faktiske endringen av floating gate noden være av motsatt retning enn hva som ønskes. Vi kan i figur 4.19 se at når $dW = 50mV$ er $V_{cg} \approx 1.5V$. Et innsignal $dW = 50mV$ betyr at floating gate noden skal ha en positiv endring, men fordi $V_{fg} \approx 2.2V$, vil det i praksis være en negativ spenningsforskjell mellom V_{cg} og V_{fg}. Endringen av V_{fg} blir derfor negativ når den faktisk var en positiv endring som var målet. Det er derfor svært viktig at V_{cg} trekkes nær V_{DD} for en positiv endring av V_{fg} og nær GND for en negativ endring.

Figur 4.20 viser tilsvarende målinger som i figur 4.19, men med en større avstand mellom krets og lyskilde, 15cm. Vi ser at endringen av V_{cg} og V_{cap} som funksjon av dW med UV-lys er forbedret betraktlig, men ikke nok i forhold til målingene i figur 4.19.

Vi ser både i figur 4.20 og 4.19 at floating gate noden er noe ustabil som en følge av at V_{cg} og V_{cap} ikke er symmetriske. Ustabiliteten er en uheldig effekt og muligens ødeleggende når hukommelseselementet anvendes i et nevralt nett.

Figur 4.21 viser tilsvarende målinger som i figur 4.19, men med et annet valg av arbeidsområde for innsignalet dW ($dW_{-} = 1.7V$). Vi ser at det er ubetydelige forskjeller for alle de tre nodene når kretsen ikke blir belyst med UV-lys i forhold til når den blir belyst. Figuren viser at utslaget i V_{fg} som følge av usymmetrien mellom V_{cg} og V_{cap} begynner å



Figur 4.21: V_{fg}, V_{cg} og V_{cap} som funksjon av innsignalet dW både med og uten UV-lys, med $dW_{-} = 1.7V$ og en avstand mellom krets og lyskilde lik 5cm. Heltrukkene kurver er målinger uten UV-lys mens stiplede kurver er målinger med UV-lys. Vi ser at den fotoelektriske effekten er ubetydelig i forhold til driveegenskapene ved valg av en arbeidsområde for innsignalet $dW_{-} = 1.7V$. Vi ser imidlertid også store uønskede utslag i V_{fg} som følge av usymmetrien mellom V_{cg} og V_{cap}.

bli svært stort.

4.4.4 Forbedringer

For å sikre gode drive egenskaper av kontroll noden V_{cg} også ved små innsignaler dW, kan vi skalere transistorene $t1$ og $t2$ i hukommelseselementets inngangstrinn (figur 4.11) slik at de to transistorene blir store og kraftige. Hukommelseselementet blir da mer robust og mindre plaget av svake driveegenskaper av kontroll noden i forbindelse med endring av floating gate noden.

Den andre ulempen med det nåværende hukommelseselementet er usymmetrien mellom kontroll noden V_{cg} og den inverterte V_{cap}. Usymmetrien fører til uønskete forstyrrelser av floating gate noden V_{fg}. Det er et problem som sannsynligvis vil være vanskelig å løse helt tilfredstillende. En mulighet er å la en transkonduktansforsterker med et differensielt innsignal dW drive kontroll noden V_{cg}, mens en annen transkonduktansforsterker med innsignal $-dW$ driver noden V_{cap}. Vi unngår da inverteren. Signalene fra de to transkonduktansforsterkerene med innsignal henholdsvis lik dW og $-dW$ vil i følge simuleringer ikke gi et fullt så usymmetrisk signalnivå mellom nodene V_{cg} og V_{cap} som i det nåværende hukommelseselementet. Men uønskete effekter i signalnivået til floating gate noden vil fremdeles være til stede. Et annet og liknende alternativ som passer bedre inn i min sammenheng vil være å bruke to like inngangstrinn slik som det i figur 4.11. Det ene inngangstrinnet med innsignal dW driver da kontroll noden V_{cg}, mens det andre inngangstrinnet med innsignal $-dW$ driver noden V_{cap} slik at vi unngår inverteren. Problemene er tilsvarende som for det første alternativet.

4.4.5 Oppsummering og konklusjon

Målinger viser at selve programmeringen av floating gate noden V_{fg} etter spenningsnivået på kontroll noden V_{cg} virker svært tilfredstillende. En stor positiv spenningsforskjell mellom V_{cg} og V_{fg} vil gi en positiv endring av V_{fg}, mens en stor negativ spenningsforskjell vil gi en negativ endring. Det er flere muligheter for å variere endringens størrelse pr. tidsenhet, bla. ved å variere avstanden mellom krets og lyskilde. Endringens oppløsning tyder i følge målinger på å være meget god.

Det nåværende hukommelseselementet har et par svakheter. Driveegenskapene av kontroll noden er ikke helt tilfredstillende, men det skal kun skalering av enkelte transistorer til for å rette opp den svakheten. Et større problem er usymmetrien i signalnivået mellom V_{cg} og V_{cap} som fører til uønskete effekter på floating gate noden, et problem som kan være vanskelig å løse helt tilfredstillende. Imidlertid er det noe uklart hvor kritisk effekten er når hukommelseselementet brukes i en større sammenheng slik som i et nevralt nett.

4.5 Subtraksjon og addisjon

Addisjon og subtraksjon av strømmer implementeres ved å anvende Kirchhoffs strømlov. Hvert nevron i et nett har en summeringer. Vi kan svært enkelt benytte Kirchhoffs strømlov

for å implementere addisjon. Det er bare å koble alle signalkomponenter som skal summeres i en felles node. Siden vi benytter en differensiell representasjon av signalet, må vi implementere en summering for de positive komponentene, og en for de negative.

Det siste grunnleggende elementet vi har behov for, er et kretselement for subtraksjon. Utgangen o_k fra nevronene i utgangslaget i et nett skal subtraheres fra ønsket utgang t_k, altså $(t_k - o_k)$. Vi kan implementere subtraksjonen ved anvendelse av Kirchhoffs strømlov

Det er behov for to delkretser, en for å beregne den positive signalkomponenten, og en for å beregne den negative. Generelt kan vi si at de signalkomponentene som skal trekkes fra, trekker i motsatt retning av de andre signalkomponentene. Utgangssignalet kopieres lett fra diodekoblingene på delkretsenes utganger. Kretsskjema og et enkelt symbol for subtraksjonskretsen er vist i figur 4.22. De to differensielle signalene $i1$ og $i2$ er inngangssignalene til subtraksjonskretsen, mens *out* er det differensielle utgangssignalet.

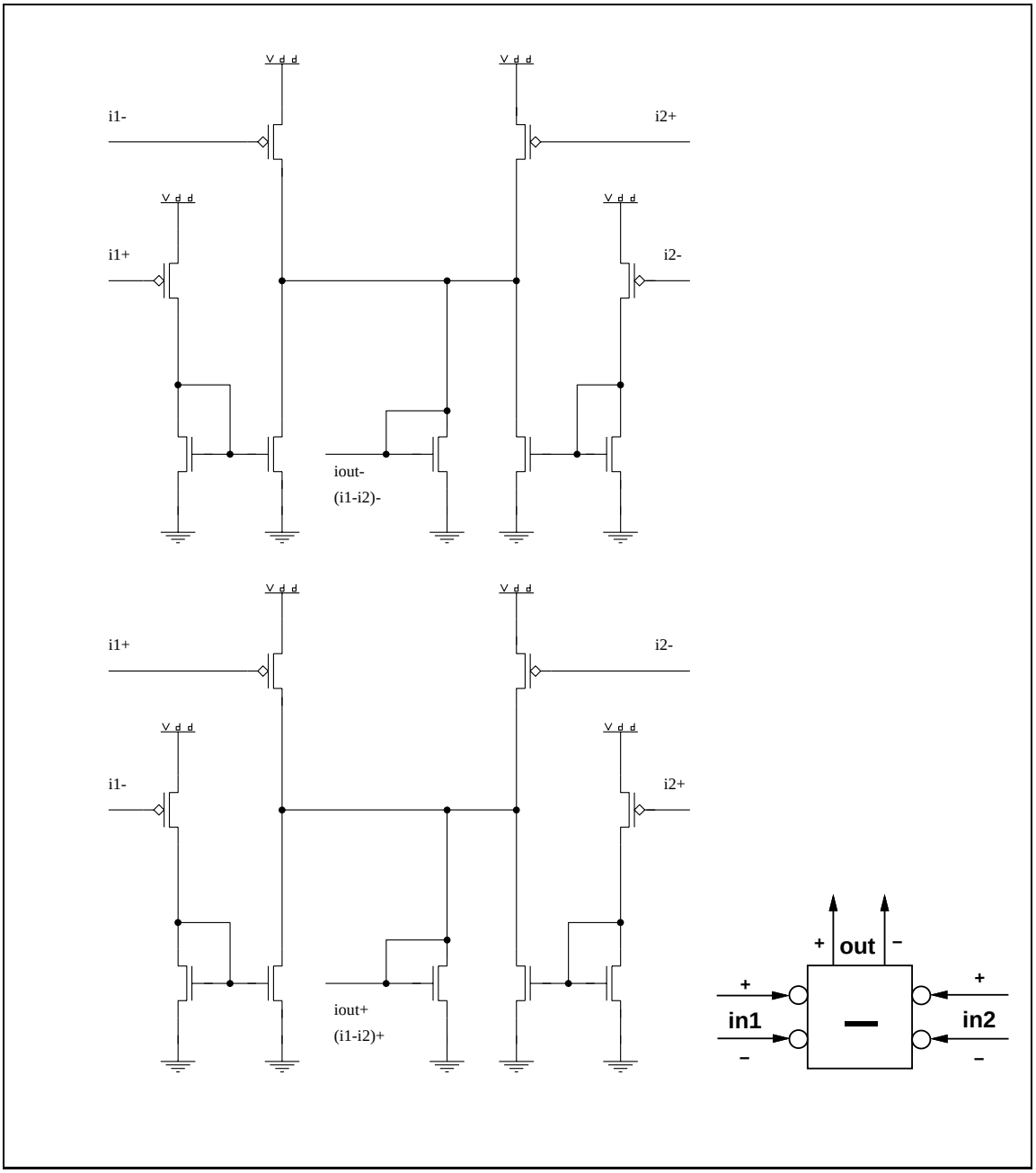
Jeg har ikke tatt med en enkeltstående subtraksjonskrets på de integrerte kretsene jeg har fått produsert slik at målinger av en separat subtraksjonskrets ikke har vært mulig.

4.5.1 Simulering

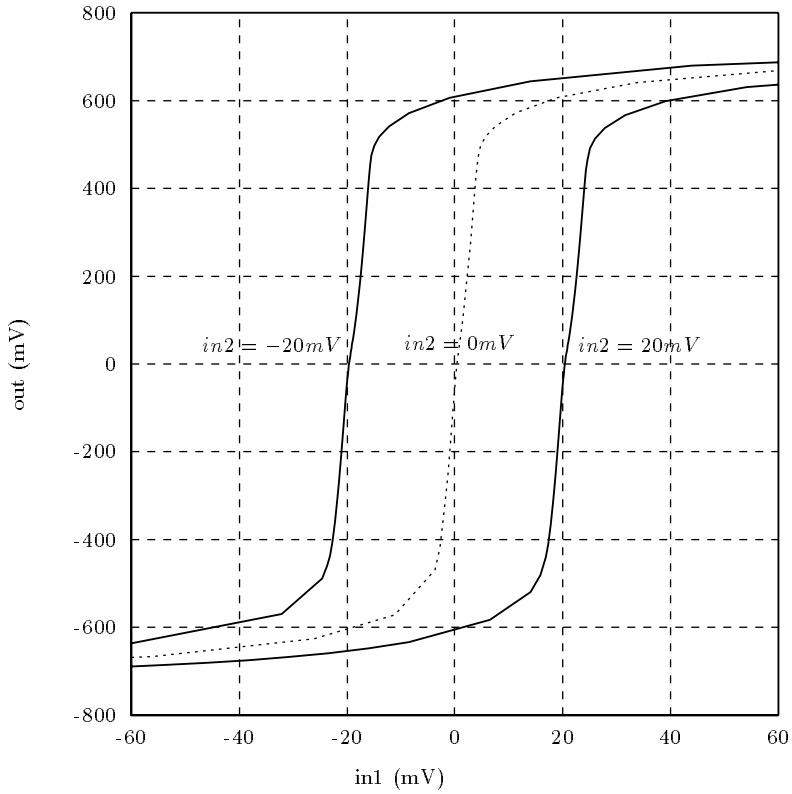
Figur 4.23 viser imidlertid en simulering i anaLOG av subtraksjonskretsen i figur 4.22. I et nevralt nett vil inngangssignalene til en subtraksjonskrets være ønsket utgangssignal t_k og faktisk utgangssignal o_k. Disse to signalene er utgangssignaler fra transkonduktansforsterkere slik som vist i figur 4.1. Ved simulering av subtraksjonskretsen har jeg latt subtraksjonskretsenes differensielle innganger $i1$ og $i2$ fått påtrykt utgangssignalet fra hver sin transkonduktansforsterker slik som vist i figur 4.1. Transkonduktansforsterkerenes differensielle inngang har jeg kalt henholdsvis $in1$ og $in2$.

Jeg har holdt inngangssignalet til transkonduktansforsterkeren med inngang $in2$ konstant og variert inngangssignalet til transkonduktansforsterkeren med inngang $in1$. Jeg har betraktet utgangsspenningen fra hver av delkretsene som utgjør subtraksjonskretsen. Jeg har deretter sett på differansen mellom utsignalene fra de to delkretsene som er det differensielle utsignalet *out* fra subtraksjonskretsen.

Figur 4.23 viser utsignalet *out* fra subtraksjonskretsen som funksjon av innsignalet $in1$. Siden innsignalene til subtraksjonskretsen kommer fra transkonduktansforsterkerere, vil innsignalet $in1$ som varieres, gi et signal med form som en sigmoid. Vi kan av figuren se at sigmoiden forflyttes langs $in1$-aksen i henhold til størrelsen av det andre innsignalet $in2$. Dessuten kan vi se at de tre sigmoidenes lineære områder fremdeles er lineære etter subtraksjonen med konstanten $in2$. I følge simuleringen fungerer altså subtraksjonskretsen tilfredstillende.



Figur 4.22: Kretsskjema og et enkelt symbol for subtraksjonskretsen. Kretsen består av en delkrets for hvert av de differensielle signalkomponentene i utsignalet.



Figur 4.23: Simulering av subtraksjonskretsen. Inngangssignalene $in1$ og $in2$ påtrykkes hver sin transkonduktansforsterker som videre gir inngangssignalene til selve subtraksjonskretsen. Det er årsaken til at kurvene har en sigmoid form. Vi kan se at kurvene forflyttes etter størrelsen av det konstante inngangssignalet $in2$.

Kapittel 5

Oppdeling av nevrale nett i moduler

Da jeg begynte å arbeide med en implementasjon av backprop nett i analog VLSI, prøvde jeg å strukturere det hele slik at jeg ikke ble bundet til en helt bestemt nettarkitektur. Jeg lagde meg to større moduler. I den første modulen implementerte jeg alt som hadde å gjøre med beregninger i forbindelse med en vekt for hele algoritmen, både for feedforward- og backprop-delen av algoritmen, samt selve vekten. Den første modulen har jeg kalt en vekt. I den andre hovedmodulen implementerte jeg alt som hadde med beregningene i et nevron å gjøre, inkludert terskelen, for hele algoritmen. Den andre modulen har jeg kalt et nevron.

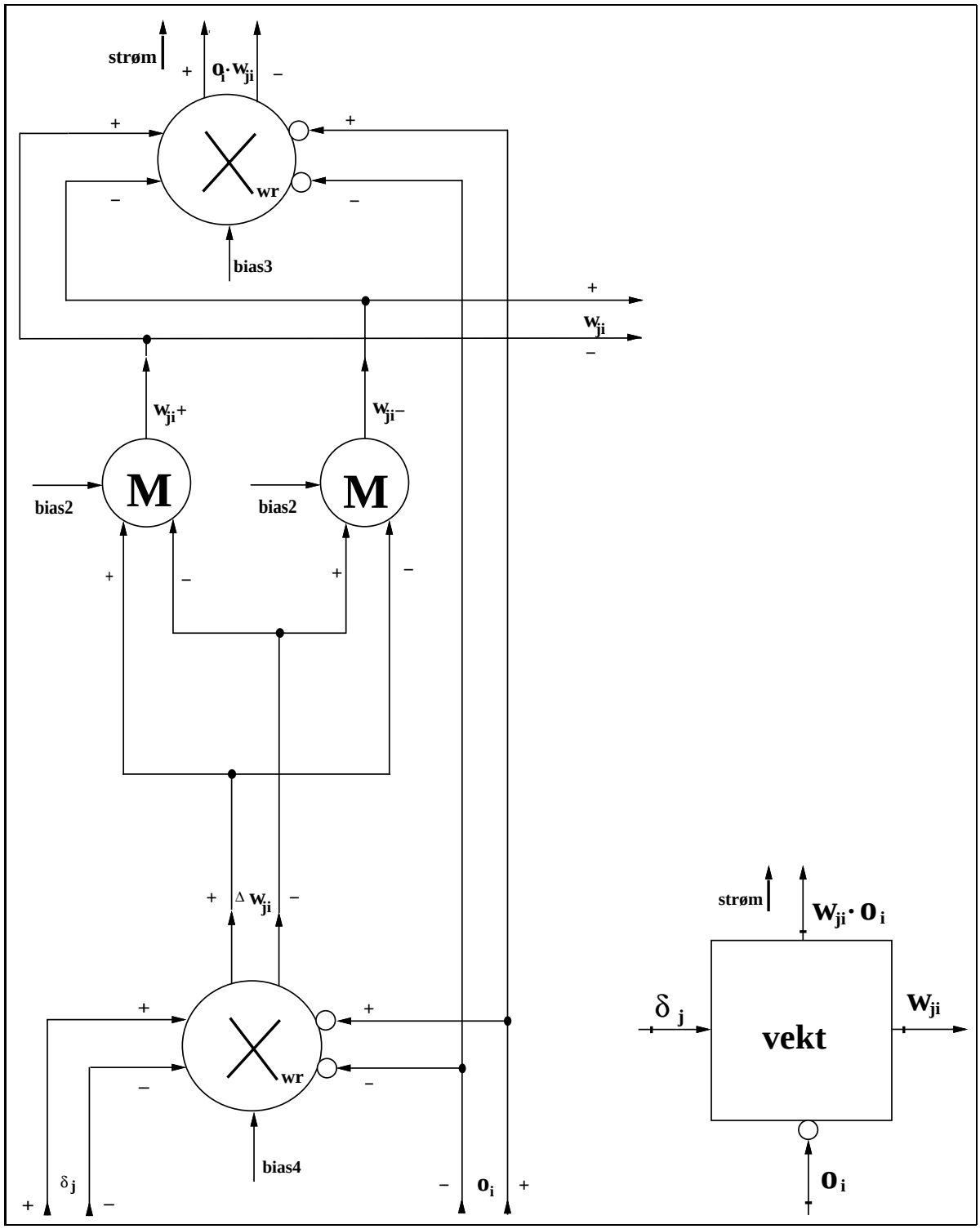
Det gir en god strukturering slik at det blir enkelt og oversiktlig å sette sammen f.eks et xor-nett slik som vist i figur 2.5. Hovedtanken med struktureringen er at det skal være enkelt å sette sammen en vilkårlig backprop nettarkitektur.

Det ikke alle beregninger som har latt seg implementere direkte etter algoritmen i analog VLSI, spesielt ikke når systemet i tillegg er kontinuerlig. Tankegangen blir derfor noe annerledes enn i et diskret system som algoritmen nok i utgangspunktet var beregnet for. Det er spesielt hvordan jeg lar vektene og tersklene endres under en opplæringsfase som adskiller seg fra algoritmen (GDR). Muligens kan det trekkes paralleller mellom f.eks. egenskaper slik som treghet ved programmering av vektene i analog VLSI og treghet ved endring av en vekt slik som det andre leddet i likning 2.7 er et uttrykk for.

Ellers har jeg stort sett implementert formlene som utgjør algoritmen direkte som kretselementer. Dog har jeg prøvd å se på bakenforliggende årsaker til formlene for om mulig se om implementasjonen kunne gjøres på en "enklere og mer naturlig måte".

5.1 Vekt

I forbindelse med en vekt vil det være behov for to multiplikatorer, en for å beregne et veid inngangssignal $o_i w_{ji}$ og en for å beregne endringen $\eta \delta_j o_i$ av vekten under en opplæringsfase. I tillegg vil det være behov for to hukommelseselementer for å representere vekten w_{ji}. Se forøvrig figur 2.3 og 2.4. Det vil derfor være naturlig å plassere de nevnte grunnleggende elementene for veiing og endring i en enkel modul. Modulen vil bestå av multiplikatoren som gir et veid inngangssignal, multiplikatoren som beregner endring



Figur 5.1: En vekt.

av vekten samt de to hukommelseselementene som utgjør selve vekten. Jeg har kalt hele modulen en vekt. Spesifikasjon av hele modulen og et enkelt symbol for modulen er vist i figur 5.1.

En vekt har to inngangssignaler. Det ene inngangssignalet er utgangssignalet o_i fra et nevron i underliggende lag. I vekten blir o_i veid med verdien av vekten w_{ji}. Produktet $o_i w_{ji}$ er et av de to utgangssignalene fra modulen. Inngangssignalet o_i anvendes dessuten sammen med det andre inngangssignalet δ_j til å beregne endringen Δw_{ji} av vekten w_{ji} under opplæring. Utgangssignalet $o_i w_{ji}$ har jeg valgt å ta ut som et strømsignal siden det vil spare en del transistorer spesielt hvis det er mange nevroner og hvert nevron har mange inngangssignaler. Utgangssignalet $o_i w_{ji}$ skal kun brukes i beregningen av det totale inngangssignalet net_j til nevronet. Det totale inngangssignalet net_j til nevronet er gitt i likning 2.1. Det andre utgangssignalet fra den vekten er verdien av vekten siden den benyttes ved beregning av δ for eventuelle underliggende nevroner.

Det vil være en fordel å forsyne den multiplikatoren som skal beregne tilhørende vekts endring med en egen forspenning som kan reguleres uavhengig av forspenningen til multiplikatorene som skal utføre andre beregninger. For en multiplikator som beregner endringen av en vekt, kan strømmen I_b i likning 4.3 som går gjennom forspenningstransistoren ses på som η i likning 2.4. Både η og I_b vil alltid være positive.

5.1.1 Avvik fra den generaliserte delta regelen

I følge den generaliserte delta regelen beregnes det en endring $\Delta_p w_{ji}$ for en vekt w_{ji} under en opplæringsfase når et mønster p presenteres nettet. I tillegg er det også vanlig praksis å presentere hele opplæringssettet med mønstre p og beregne endringen $\Delta_p w_{ji}$ for hvert mønster p for tilslutt etter at hele opplæringssettet er presentert å gjøre selve endringen Δw_{ji} av vekten w_{ji} slik som gitt i likning 2.6. Alternativt kan en vekt w_{ji} endres med $\Delta_p w_{ji}$ etterhvert som hvert mønster p blir presentert.

Siden mitt system er kontinuerlig, har det vært naturlig å velge den siste fremgangsmåten for endring av vekter. Vi kan betrakte endringen $\Delta_p w_{ji}$ som et inkrement av vekten w_{ji} hvor inkrementet kan ha en vilkårlig størrelse. I programvare og digital maskinvare er det forholdsvis enkelt å la en vekt endres med et vilkårlig inkrement.

I analog mikroelemtronikk vil det by på vesentlig større problemer å implementere en vekt som kan endres med et vilkårlig inkrement, spesielt dersom det skal foregå uten noen form for klokking av signaler.

Min løsning gir i utgangspunktet ikke en inkrementell endring av vektene. Endringer av en vekt forgår derimot kontinuerlig når vekten belyses med UV-lys. Det er flere faktorer som avgjør hvor mye en vekt endres. To av de viktigste faktorene er belysningstiden av hukommelseselementenes UV-struktur og avstanden mellom lyskilde og den integrerte kretsen med vektens to hukommelseselementer (belysningseffekten).

I utgangspunktet beregner jeg faktisk endringen av en vekt etter den generaliserte delta regelen slik som gitt i likning 2.4. Beregningen skjer kontinuerlig uansett om nettet er i en opplæringsfase eller ikke. Det er imidlertid kun i en opplæringsfase beregningen har innflytelse på vekten. Beregningen etter likning 2.4 utføres av en Gilbert multiplikator slik som vist i figur 4.9. Sammen med inngangstrinnet til hukommelseselementet slik som

vist lengst til venstre i figur 4.11 utgjør Gilbert multiplikatoren i figur 4.9 en komplett wide-range Gilbert multiplikator med strømutgang I_{out}. Det er strømsignalet I_{out} fra den komplette wide-range Gilbert multiplikatoren og inn i noden V_{cg} i hukommelseselementet som er resultatet av multiplikasjonen etter likning 2.4.

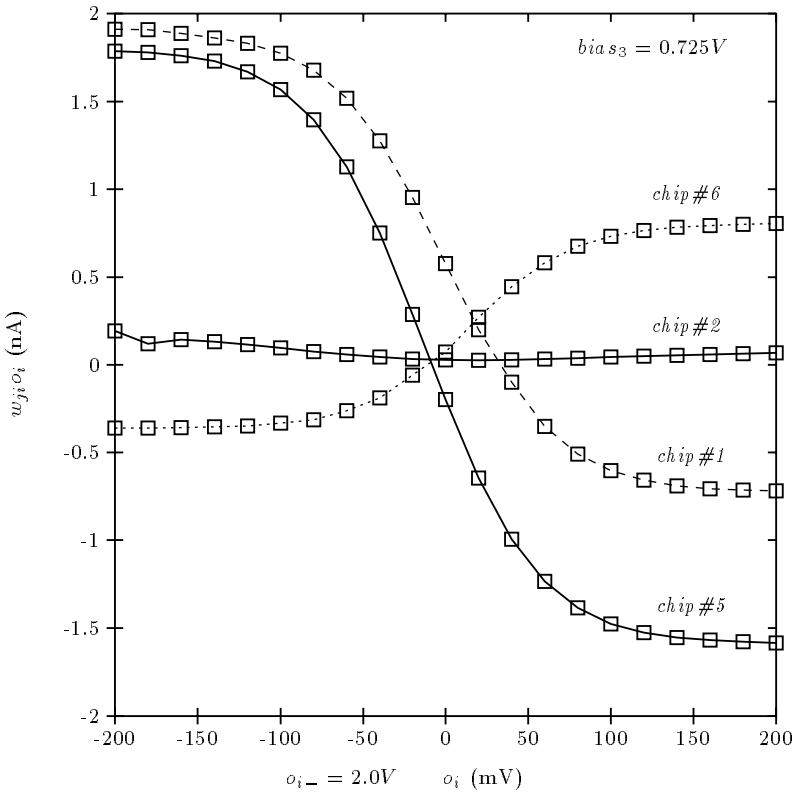
Hukommelseselementet (floating gate noden V_{fg}) programmeres imidlertid etter spenningsnivået på kontrollnoden V_{cg}. En wide-range Gilbert multiplikator har en stor utgangsforsterkning. Det betyr at multiplikatorens to differensielle inngangsspenninger ikke skal være mye forskjellig fra null før multiplikatorens utgang, dvs. kontroll noden V_{cg} i hukommelseselementet får et spenningsnivå nær *GND* når resultatet av multiplikasjonen er negativ og nær V_{DD} når resultatet av multiplikasjonen er positiv. Det er vært å merke seg at verdien av en vekt er spenningsdifferansen mellom to hukommelseselementer. Det jeg har beskrevet så langt om endringer av floating gate noden V_{fg} i et hukommelseselement som følge av inngangssignalene til multiplikatoren gjelder hukommelseselementet som utgjør den positive komponenten av vekten. For hukommelseselementet som utgjør vektens negative komponent, vil kontroll noden V_{cg} være nær V_{DD} når multiplikasjonen er negativ, mens nær *GND* når multiplikasjonen er positiv. Siden Gilbert multiplikatoren som beregner vektendringer som i praksis kun gir uttrykk for endringens fortegn, vil det være mulig å bytte den ut med en enklere krets som også utfører fortegnsberegninger.

En stor forskjell i spenningsnivå mellom V_{cg} og V_{fg} i hukommelseselementene som en følge av at V_{cg} enten er nær V_{DD} eller *GND* medfører kortere programmeringstid. Hvor mye floating gate noden V_{fg} endres mot V_{DD} eller *GND* bestemmes av bla. belysningstid og belysningseffekt. Vektene endres ikke lenger etter den generaliserte delta regelen, men vektene vil endres i riktig retning på grunnlag av endringens fortegn som sannsynligvis vil være tilstrekkelig.

For at vektene i praksis skal fungere, er det viktig at de ikke blir for store. Dersom en vekt blir stor vil det føre til at multiplikatorer som får vekten som det ene inngangssignalet ikke opererer som ønsket i sitt lineære område. Da vil vektene i praksis ha kun to verdier, positiv eller negativ. Det er absolutt ikke tilstrekkelig og det er viktig at vektene har en god oppløsning innen for et meget begrenset verdiområde, ca. $\pm 100mV$. Det er vanlig å anta at oppløsningen må være minst tilsvarende 8 bits for at vektene skal fungere tilfredstillende.

5.1.2 Ekstra multiplikator - en utvidelse av modulen vekt

Dersom inngangssignalet o_i som veies av en vekt w_{ji} er et utgangssignal fra et nevron i et skjult lag, er det dessuten behov for en multiplikator for å beregne vektens bidrag til δen (forplantet feil) til det underliggende nevronet. Figur 2.4 viser øverst til venstre to slike multiplikatorer som beregner hver sin vekts bidrag henholdsvis $\delta_{11}w_{111}$ og $\delta_{22}w_{211}$ til det underliggende nevronets forplantete feil δ_{11}. Den ekstra multiplikatoren har en nær tilknytning til en vekt og sørger egentlig for en utvidelse av vekten. Siden ikke alle vekter har behov for en ekstra multiplikator, har jeg holdt multiplikatoren utenfor modulen for en vekt.



Figur 5.2: Strømutgangen $w_{ji}o_i$ fra fire vekter som funksjon av o_i og med initielle verdier av w_{ji}. Chip#1 og chip#5 har positive verdier for w_{ji} (o_i er et signal med referanse til V_{DD}), mens chip#6 har en negativ verdi for w_{ji}. Chip#2 viser et signal for $w_{ji}o_i$ hvor $w_{ji} \approx 0V$.

5.1.3 Målinger uten UV-lys

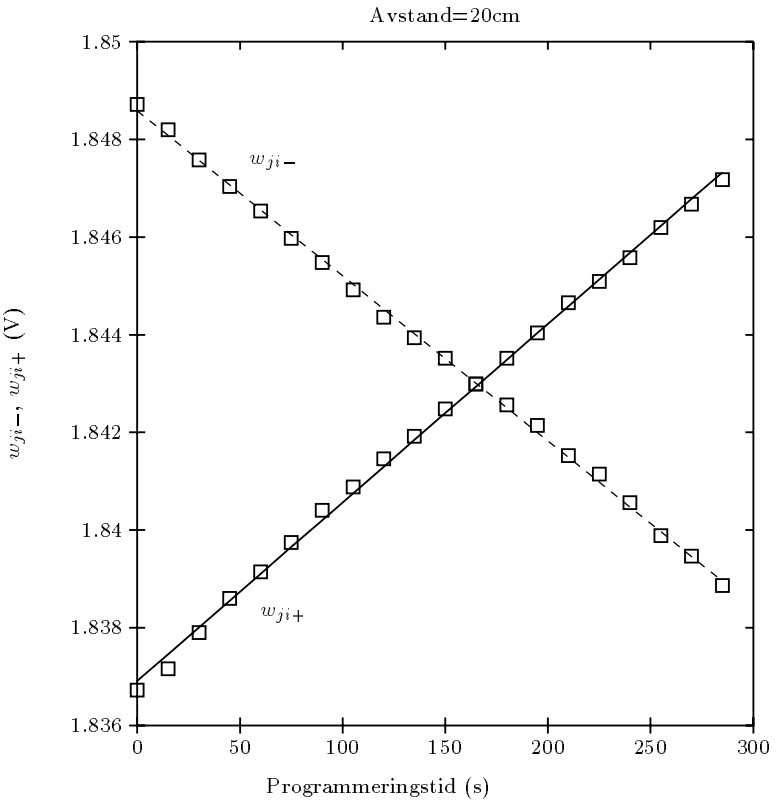
I utgangspunktet vil en tilfeldig ladning være lagret på hvert hukommelseselement på en integrert krets før den blir bestrålt med UV-lys. En vekt består som kjent av to hukommelseselementer, og det er spenningsdifferansen mellom de to hukommelseselementene som er verdien av vekten. Et av kriteriene for opplæring av et backprop nett er nettopp at verdiene av vektene er tilfeldige før en opplæring.

Jeg har gjort målinger av vekten w_{ji} i flere vekter uten bruk av UV-lys. Den initielle verdien av w_{ji} i vekten i 8 av kretsene varierer med god spredning fra -20mV og til 80mV, hvorav en faktisk har en verdi $w_{ji} \approx 0mV$.

Vekten gir det veide innsignalet $w_{ji}o_i$ ut som utsignal. Beregningen utføres av en Gilbert multiplikator. Jeg har valgt å ta ut $w_{ji}o_i$ som et strømsignal slik at signalet da er utsignal fra en vanlig wide-range Gilbert multiplikator. Jeg har variert innsignalet o_i til vekten mellom -0.3V til 0.3V i steg av 0.02V og målt utsignalet $w_{ji}o_i$ for 4 av kretsene. Figur 5.2 viser plot av målingene. Vi ser tydelig av plottet at de initielle verdiene av w_{ji} varierer.

5.1.4 Programmering av en vekt - målinger

Jeg har programmert en vekt både med positiv og negativ endring av vektens verdi w_{ji}. Jeg har dessuten målt det veide innsignalet $w_{ji}o_i$ etterhvert som jeg har programmert en



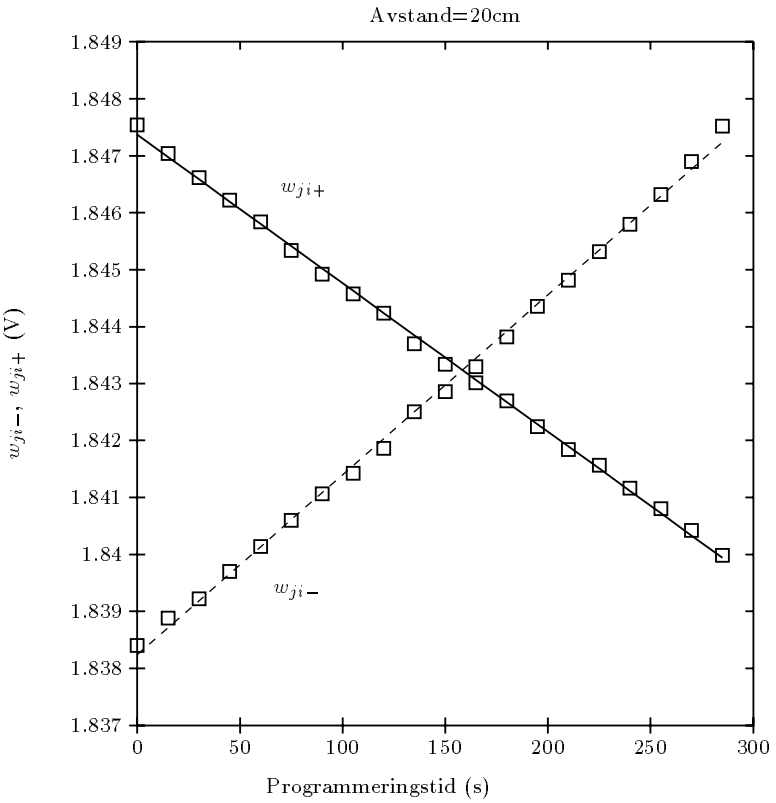
Figur 5.3: Programmering for positiv endring av vekten. Den positive komponenten w_{ji+} stiger som en følge av programmeringen, mens den negative w_{ji-} avtar. Totalt sett gir det en positiv endring av w_{ji}. Firkanter representerer målte verdier, mens linjer er lineære regresjoner. Vi ser at vi har en svært lineær karakteristikk for både w_{ji+} og w_{ji-} når den totale endring av hver av dem er liten.

endring av vekten.

Endringen av vektens retning bestemmes av innsignalene o_i og δ_j som er innsignaler til multiplikatoren som beregner endringens retning. For å sikre driveegenskapene av de to hukommelseselementenes kontroll noder best mulig som en følge av problemene beskrevet i 4.4.3, har jeg sørget for at de to innsignalene er forholdsvis store ($\pm 1.5V$). Jeg har i tillegg valgt en stor avstand mellom krets og lyskilde for å sikre best mulig overføringskarakteristikk av kontroll noden i hvert av de to hukommelseselementene. Figur 4.20 viser en slik karakteristikk hvor kontroll noden er funksjon av innsignalet til hukommelseselementet. Figuren viser en relativt god karakteristikk ved en avstand lik 15cm mellom krets og lyskilde for et enkelt hukommelseselement. Under programmeringen av en vekt har jeg valgt en enda større avstand, 20cm, for ytterligere å forbedre karakteristikken av kontroll noden.

Jeg har for alle målingene i forbindelse med programmeringen av en vekt valgt arbeidsområder for o_i og δ_j lik henholdsvis 2,5V og 3.0V. Vektens forspenninger har jeg også holdt konstante for alle målingene, hvor $bias_2 = bias_3 = bias_4 = 0.75V$.

Når en vekt w_{ji} programmeres med en positiv endring, vil hukommelseselementet som representerer den positive komponenten av vekten w_{ji+} få en positiv endring av spen-



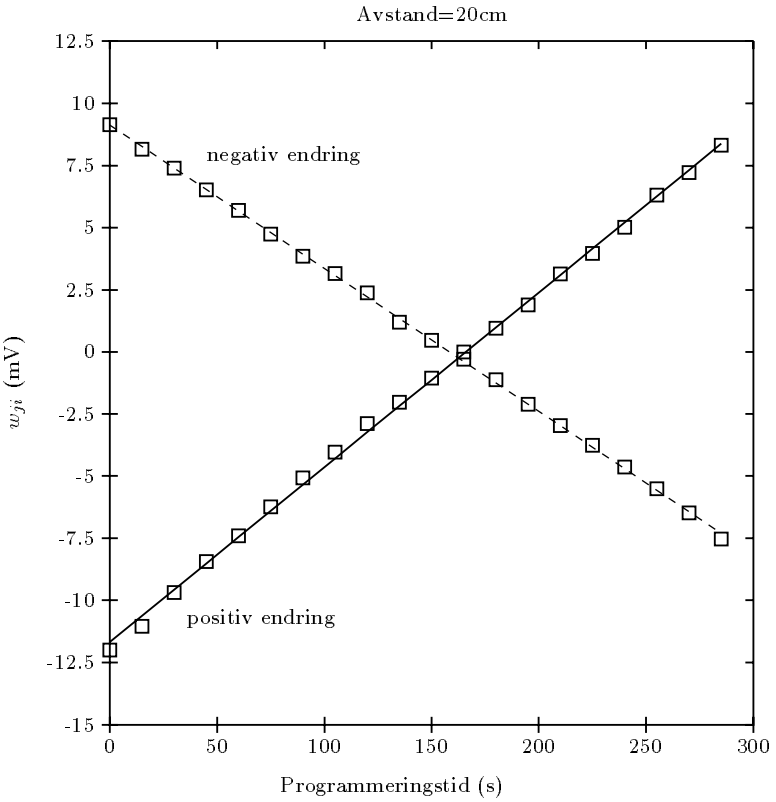
Figur 5.4: Programmering for negativ endring av vekten. Den negative komponenten w_{ji+} stiger som en følge av programmeringen, mens den positive w_{ji-} avtar. Totalt sett gir det en negativ endring av w_{ji}. Firkanter representerer målte verdier, mens linjer er lineære regresjoner. Vi ser at vi har en svært lineær karakteristikk for både w_{ji+} og w_{ji-} når den totale endring av hver av dem er liten.

ningsnivået på floating gaten noden, mens hukommelseselementet som representerer den negative komponenten w_{ji-} vil få en negativ endring. Totalt vil differansene mellom endringen av w_{ji+} og w_{ji-} gi en positiv endring av vekten w_{ji}. For en negativ endring av w_{ji} gjelder det motsatte.

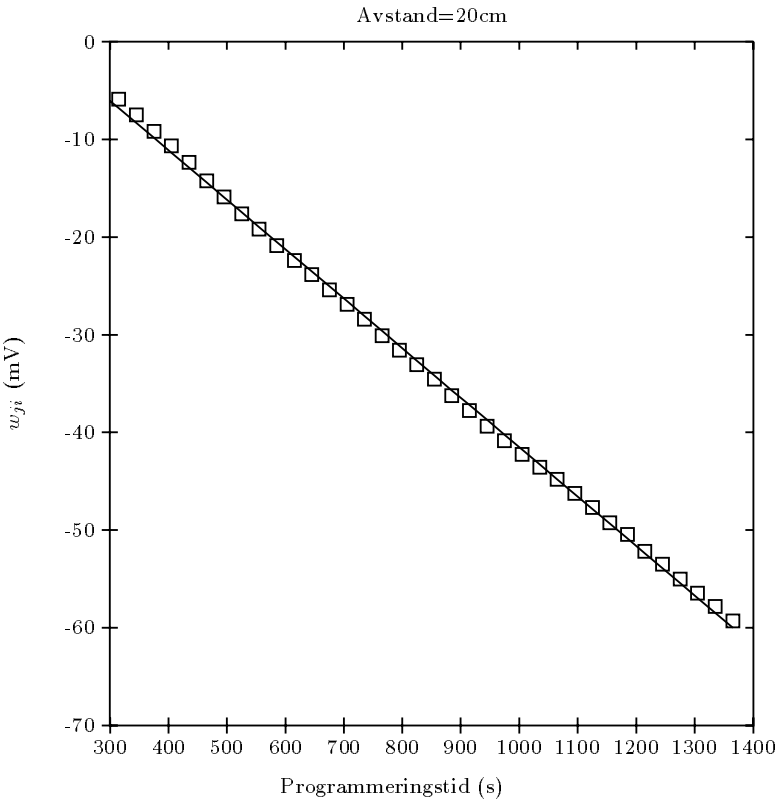
Figur 5.3 viser endringen av w_{ji+} og w_{ji-} som funksjon av programmerings eller belysningstiden. Vi ser at verdien av w_{ji+} stiger mens verdien av w_{ji-} avtar. Totalt sett gir det en positiv endring av vekten w_{ji}. Firkanter representerer målinger mens linjen er regresjoner av målepunktene for w_{ji+} og w_{ji-}. Endringen pr. tidsenhet er liten fordi avstanden mellom krets og lyskilde er stor som medfører en liten belysningseffekt. Den totale endringen av w_{ji+} og w_{ji-} er innen for en meget begrenset verdiområde, ca. 10mV, og vi ser at endringen er svært lineær.

Figur 5.4 viser en tilsvarende måling som i figur 5.3, men med en negativ endring av vekten w_{ji}. Vi ser at w_{ji-} stiger mens w_{ji+} avtar, som totalt gir en negativ endring av vekten w_{ji}.

Figur 5.5 viser differansen (w_{ji}) mellom w_{ji+} og w_{ji-} i figur 5.3 (positiv endring) og differansen (w_{ji}) mellom w_{ji+} og w_{ji-} i figur 5.4 (negativ endring). Vi ser at kurven med positiv endring endrer seg fra en negativ verdi for w_{ji} og til en positiv verdi, mens det er



Figur 5.5: Programmering for positiv og negativ endring av en vekt w_{ji}. Vekten w_{ji} er differansen mellom w_{ji+} og w_{ji-}. Firkanter representerer målte verdier, mens linjer er lineære regresjoner. Vi ser at målepunktene har en svært lineær karakteristikk når den totale endringen i en retning er relativt liten.



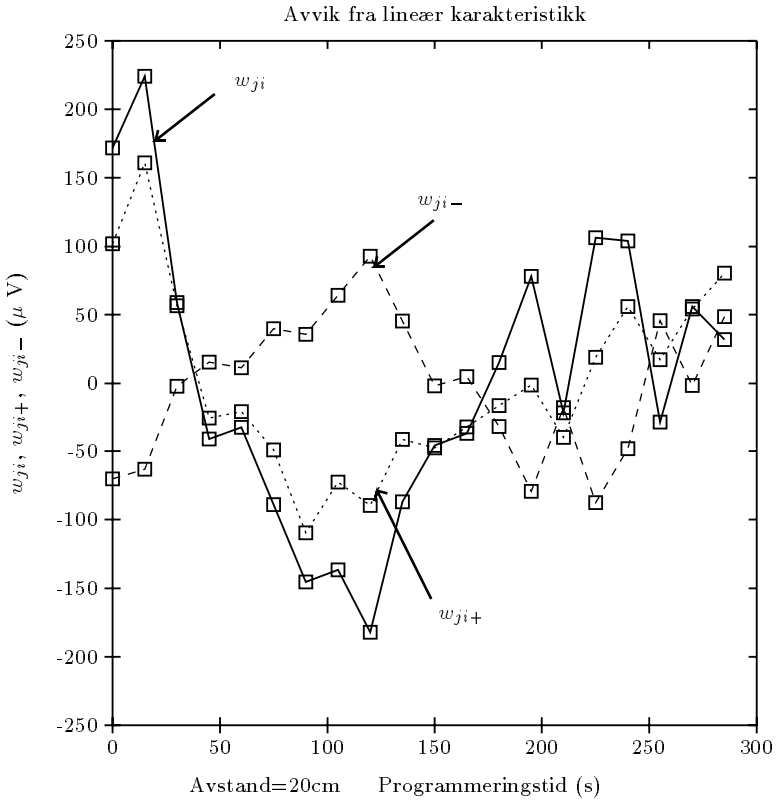
Figur 5.6: Negativ endring av en vekt w_{ji} over et noe større verdiområde. Firkanter representerer målte verdier, mens linjen er en lineær regresjon.

motsatt for kurven med negativ endring. Som en følge av at w_{ji+} f.eks. har en positiv endring og w_{ji-} dermed har en negativ endring, vil oppløsningen for vekten w_{ji} være det halve av oppløsningen for w_{ji-} og w_{ji+}. Imidlertid tyder målingene av en vekt og et hukommelseselement på at oppløsningen av en vekts verdiområde er svært god.

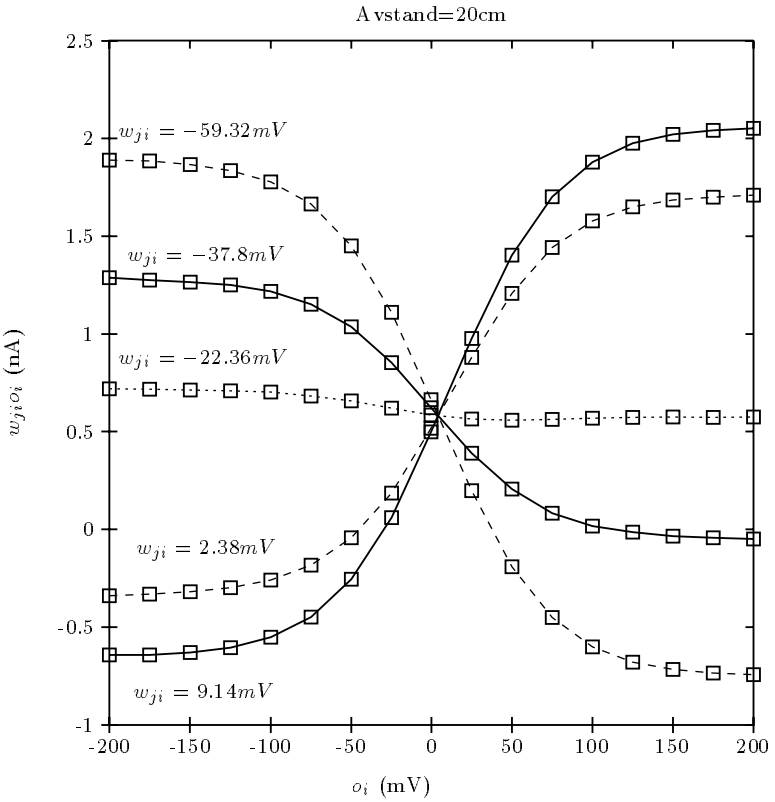
Figur 5.6 viser en negativ endring av en vekt over et større verdiområde enn i figur 5.5. Vi ser at vekten har en total endring på ca. 70mV i løpet av ca. 1000 sekunder når avstanden mellom krets og lyskilde er lik 20cm. Vi ser at den totale endringen fremdeles er svært lineær.

For å studere avviket fra lineær karakteristikk for en vekt nærmere, har jeg beregnet avviket mellom målte verdier og den lineære regresjonen for vekten w_{ji} for positiv endring i figur 5.5 og dens komponenter w_{ji-} og w_{ji+} i figur 5.3. Avvikene er vist i figur 5.7. Vi ser i figur 5.5 at den positive endringen av vekten w_{ji} i løpet av 300 sekunder er ca. 200mV, mens avviket maksimalt er $\pm 200\mu V$.

Jeg har programmert en vekt med en negativ endring slik at vekten endret seg gradvis fra positive verdier og til negative verdier og utførte målinger av vekten og produktet $w_{ji}o_i$ mellom hver programmeringsperiode. Figur 5.8 viser det veide innsignalet $w_{ji}o_i$ som funksjon av o_i for ulike programmerte verdier av w_{ji}. Vi ser tydelig at tallverdien av produktet $w_{ji}o_i$ øker med tallverdien av økende verdi av vekten w_{ji}. Vi ser også at vi har en fire kvadrants multiplikasjon og at multiplikatoren har en offset som ventet slik at karakteristikken av produktet $w_{ji}o_i$ ikke endrer seg for $w_{ji} = 0V$, men for $w_{ji} \approx 20mV$.



Figur 5.7: Avvik fra lineær karakteristikk (regresjonslinjen) for endringen av en vekt over et lite verdiområde. Figuren viser avviket for både w_{ji-}, w_{ji+} og w_{ji}. Vi ser at avviket for w_{ji} er summen av avvikene for w_{ji-} og w_{ji+}. Avviket er imidlertid lite i forholde til endringen av vekten.



Figur 5.8: Det veide inngangssignalet $w_{ji}o_i$ som funksjon av o_i med ulike programmerte verdier av w_{ji}. Vi ser at når w_{ji} endrer seg fra positive til negative verdier, endrer $w_{ji}o_i$ karakteristikk, og vi ser at multiplikatoren utfører en fire kvadrants multiplikasjon. Offset feil i multiplikatoren er årsaken til at $w_{ji} = 0V$ ikke gir $w_{ji}o_i \approx 0A$.

5.1.5 Forbedringer

Source degenerering eventuelt kapasitiv divisjon vil kunne øke multiplikatorenes lineære område slik som beskrevet i avsnitt 4.3.2. Det er spesielt interessant for multiplikatoren som beregner det veide innsignalet $w_{ji}o_i$. Dessuten vil en skalering av transistorer i hukommelseselementets inngangstrinn gi vesentlig bedre driveegenskaper av kontroll noden slik som nevnt i avsnitt 4.4.4. Det vil sikre en pålitelig endring av en vekt samtidig som belysningseffekten kan økes.

Det er mulig å legge inn ekstra diodekoblinger i tillegg til diodekoblingene $d1$ og $d2$ på utgangen av multiplikatoren (se figur 4.9) som beregner endringen, en ny diodekobling i kaskade med diodekobling $d1$ og en i kaskade med $d2$. Kontroll gate noden i de to hukommelseselementene vil da ikke gå steilt fra lavt til høyt signal slik som målingene i figur 4.12 viser siden dW_- og dW_+ vil variere i området rundt to diode-offset fra GND, ca. 1.7V slik som målingene i figur 4.13 viser. Spenningsnivået på kontroll noden vil da gi uttrykk for mer enn kun retningen av vektens endring. Den vil til en viss grad skalere størrelsen av endringen etter størrelsen av spenningsnivået på kontroll noden. Imidlertid forutsetter det svært gode driveegenskaper av kontroll noden i hukommelseselementet.

5.1.6 Oppsummering og konklusjon

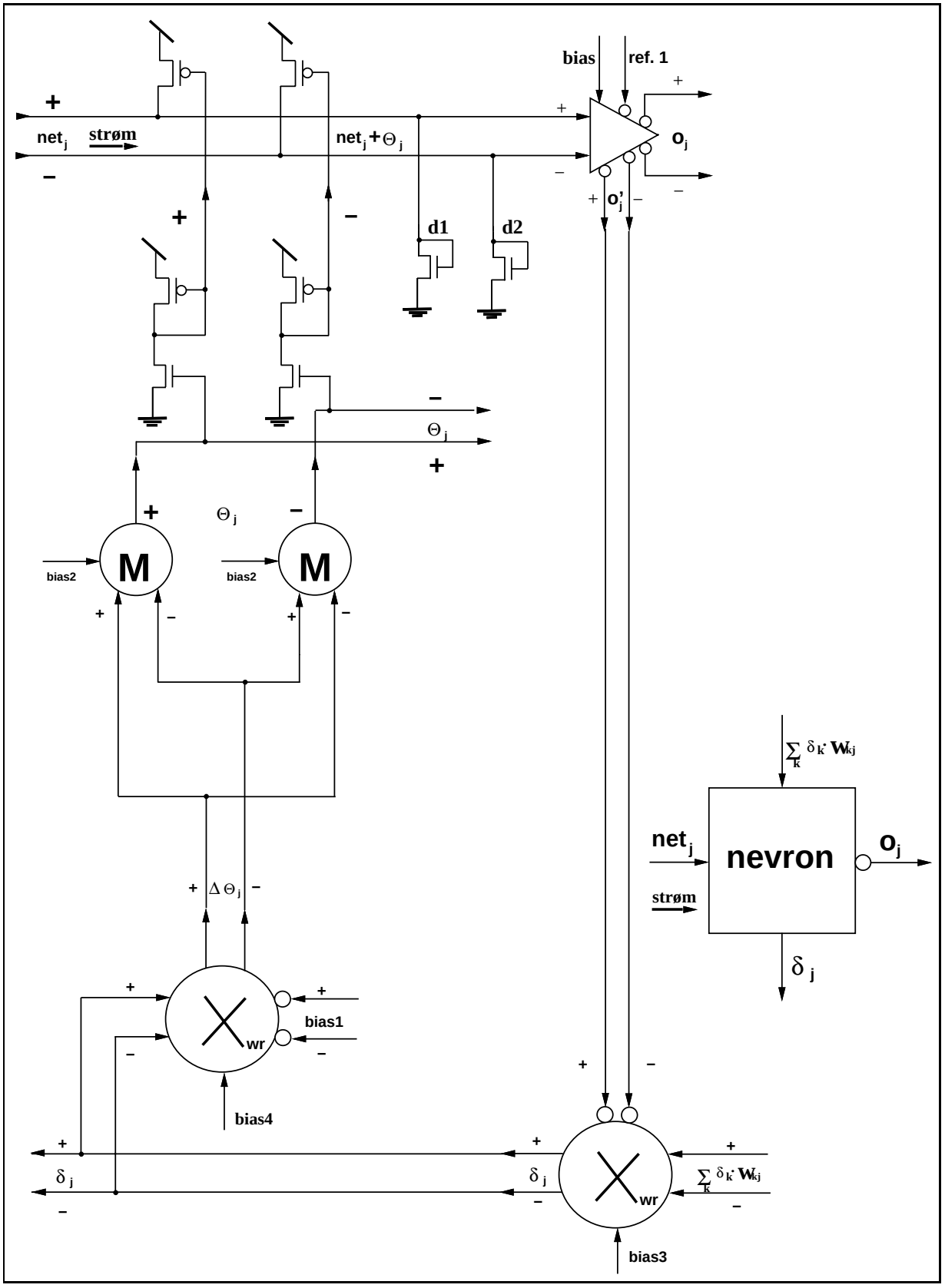
Under en klassifiseringsfase viser målinger at vekten har en stabil verdi. Målinger viser også en tilfredsstillende multiplikasjon mellom o_i og w_{ji} som gir det veide innsignalet $o_i w_{ji}$. For at multiplikasjonen skal være tilfredstillende må dessuten verdien w_{ji} av vekten befinne seg innenfor et begrenset arbeidsområde slik at multiplikatoren opererer i sitt lineære område. Det siste kravet er at vekten har en tilfredsstillende god oppløsning i arbeidsområdet, tilsvarende 8 bit.

Programmering av en vekt viser at den kan programmeres sikkert og nøyaktig med en meget god oppløsning også over et begrenset verdiområde. Som en følge av svakheter ved hukommelseselementets inngangstrinn som det er enkelt å forbedre, har jeg programmert vekten med en liten belysningseffekt. Det medfører en liten endring av vekten pr. tidsenhet, men med nevnte forbedringer av hukommelseselementet vil endringen av en vekt pr. tidsenhet kunne økes betraktlig. Vi har også sett at vekten gir et varierende utgangssignal $w_{ji}o_i$ etterhvert som vekten w_{ji} programmeres med en endring.

Målinger viser altså at modulen vekt utfører ønskete beregninger og endringer av vekten tilfredstillende. Målinger har også vist at en vekts verdi i utgangspunktet før første programmering har en randomisert verdi som er et av kravene til vektene i et nett for at nettet skal kunne læres opp. Verdien av vekten er dessuten randomisert innen for et begrenset og ønsket verdiområde.

5.2 Nevron

Det totale inngangssignalet net_j til et nevron er summen av de veide inngangssignalene til nevronet. Det kan være en eller flere inngangssignaler $w_{ji}o_i$ til et nevron.



Figur 5.9: Et nevron.

Hvert nevron består av en aktiviseringsfunksjon med den tilhørende deriverte som brukes i forbindelse med GDR. Multiplikatoren som beregner δ_j som er et uttrykk for forplantet feil i nettet frem til et bestemt nevron, tar den deriverte av utgangssignalet o_i fra nevronet som det ene av inngangssignalene. Ved å samle de nevnte beregninger i en modul, vil derivasjonssignalet bli holdt lokalt innenfor en modul. Terskelen Θ_j sørger for en forskyvning av sigmoidsignalet o_i. Terskelen Θ_j har mye til felles med en vekt, men de er ikke helt identiske. Det er derfor naturlig å la den omtalte modulen også inneholde kretselementene for lagring og endring av terskelen Θ_j.

Jeg har kalt hele modulen for et nevron. Nevronet har net_j som inngangssignal og sigmoidsignalet o_i som utgangssignal. Siden net_j er et inngangssignal i form av en strøm, og transkonduktansforsterkeren som gir sigmoidfunksjonen krever et inngangssignal i form av spenning, blir net_j etter at det er addert med terskelen Θ_j konvertert til en spenning ved hjelp av diodekoblinger. I tillegg vil det være et inngangssignal i forbindelse med beregningen av δ_j. For nevroner i utgangslaget vil inngangssignalet være $(t_k - o_k)$, mens for nevroner i skjulte lag, vil det være $\sum_k \delta_k w_{kj}$. Modulen beregner δ_j som er et uttrykk for forplantet feil fra nettets utgang og nedover i nettet frem til det aktuelle nevronet, som det andre utgangssignalet. Nevronet samt et enkelt symbol for det er vist i figur 5.9.

For at alle de grunnleggende kretselementene innen modulen skal få riktig type inngangssignal etter hvorvidt inngangstransistorene er med referanse til *GND* eller V_{DD}, må enkelte signaler konverteres slik at signalet er av den typen som inngangstransistorene krever. Jeg har prøvd å la de grunnleggende kretselementene innen modulen ha inngangstransistorer av den typen som vil gi færrest mulig konverteringer av signaler.

Jeg har utført målinger på nevronets utgangssignaler. Jeg har spesielt studert terskelens innvirkning på utgangssignalene fra nevronet, forholdet mellom terskelens innvirkning på utgangssignalene og de uønskete transistorvariasjonene samt programmering av terskelen. Jeg har også studert hvordan valg av referansespenning for den deriverte av utgangssignalet o_i virker inn på signalets kvalitet etter at det er konvertert til et differensielt signal.

5.2.1 Terskelen Θ

Terskelen Θ i nevronene i et nett læres opp på samme måte som en vanlig vekt. Vi antar at terskelen Θ er vekten av en node som alltid er "på". I vårt tilfelle hvor vi bruker utgangssignalet fra en transkonduktansforsterker som sigmoidfunksjon vil "på" bety at strømmen ut av forsterkeren er lik forsterkerens forspenningsstrøm I_b. Strømmen I_b settes av forspenningen V_b til forsterkeren.

I praksis trenger vi en spenningsdifferans som er av en slik størrelse at tilnærmet all strøm gjennom det (de) differensielle paret (parene) i multiplikatoren som spenningsdifferansen er inngangssignal til, går igjennom kun det ene av de to "beina" i det differensielle paret. Det vil da ikke nødvendigvis være akkurat en spenningsdifferans lik forspenningen til forsterkeren, men f.eks spenningsdifferansen mellom to diodekoblinger i kaskade.

5.2.2 Målinger uten UV-lys

Et enkelt nevron består av en transkonduktansforsterker med bumputgang, to multiplikatorer og to hukommelseselementer for terskelen Θ_j. I tillegg kommer et par diodekoblinger og speil. Et enkelt nevron er som tidligere nevnt vist i figur 5.9. Test-nevronet som har noen transistorer i tillegg i forbindelse med inn- og utganger, er vist i figur C.10. Utgangen o_j fra nevronet er den vanlige transkonduktansforsterkerutgangen. På det enkle nevronet har jeg tatt ut signalene o_j og o'_j (*bump*) som strømsignaler.

Når det gjelder inngangssignalet til transkonduktansforsterkeren i nevronet, kan vi ikke sette dette direkte fra inngangspadder. Inngangssignalet net_j kan vi sette fra inngangspadder. Signalet settes i form av en differensiell spenning som går inn på to gater slik at vi får et differensielt strømsignal som videre summeres med bidraget fra terskelen Θ_j, før det hele konverteres til en differensiell spenning ved hjelp av to diodekoblinger. Det er det siste signalet som er inngangssignalet til transkonduktansforsterkeren. Verdien av terskelen Θ_j er imidlertid ukjent. Det terskelen Θ_j sørger for, er forøvrig kun å forskyve nullpunktet for utgangssignalet fra transkonduktansforsterkeren langs x-aksen.

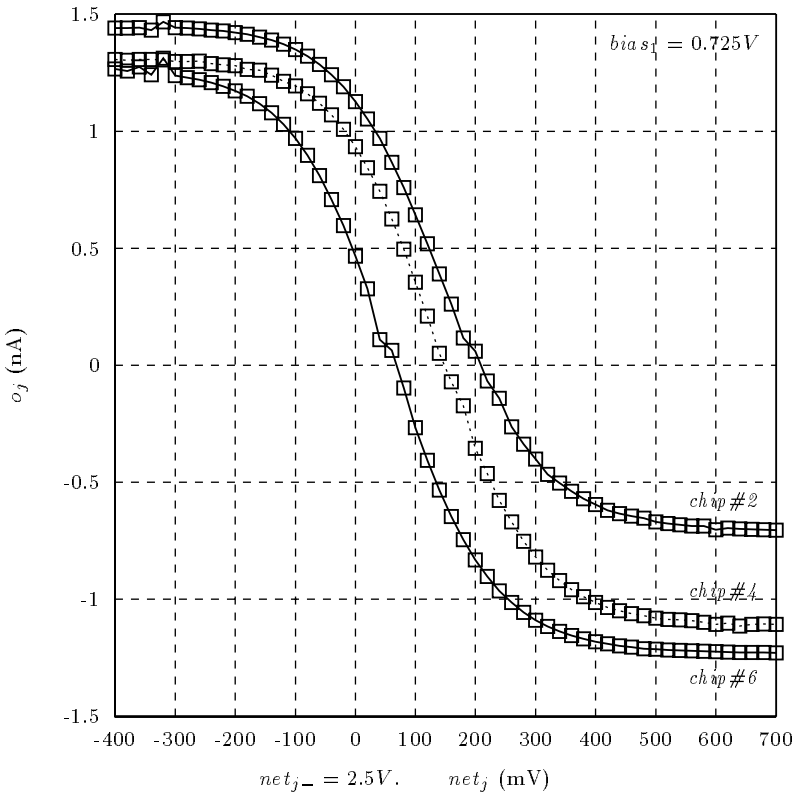
Det siste som man bør å merke seg, er at signalet net_j som settes på inngangspaddene er i forhold til V_{DD}, mens selve inngangssignalet til transkonduktansforsterkeren er i forhold til *GND*.

Figur 5.10 viser målinger på tre kretser av utsignalet o_j fra nevronet. Vi ser tydelig at de tre kurvene er forskjøvet langs x-aksen. Forskyvningen er ca. 70mV, 150mV og 210mV for henholdsvis chip #6, #4 og #2. Et typisk avvik mellom to like store transistorer av samme type er $\pm 10mV$ i gatespenning for at de skal gi samme strøm [Mead]. Den positive komponenter i utsignalet o_i fra transkonduktansforsterkeren speiles kun i et strømspeil og vil føre til et typisk avvik på $\pm 10mV$. Den negative komponenten av o_i derimot speiles i to strømspeil som totalt vil gi et typisk avvik på $\pm 20mV$. Dersom vi er mer pessimistiske kan vi regne med et avvik i transkonduktansforsterkeren på $\pm 30mV$ til $\pm 40mV$. Forskyvningen er imidlertid så stor at den ikke kan skyldes kun transistorvariasjoner. Den store forskyvningen skyldes primært terskelen Θ_j. Hensikten med Θ_j er nettopp å oppnå en slik forskyvning av signalet.

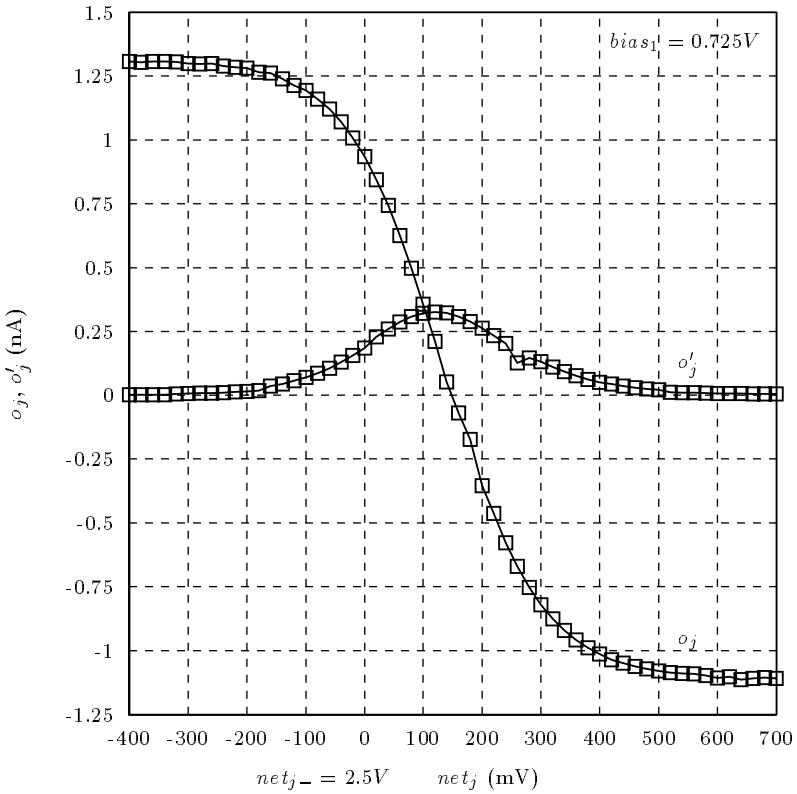
Det er verdt å merke seg at de tre signalenes lineære område er ca. 180mV som er vesentlig større enn den vanlige størrelsen av det lineære området for en vanlig transkonduktansforsterker. Signalenes lineære området i figur 5.10 er større enn normalt fordi net_j ikke er direkte innsignal til transkonduktansforsterkeren i nevronet. Innsignalet til transkonduktansforsterkeren følger net_j, men med et mindre sving som følge av diodekoblingene $d1$ og $d2$ i figur 5.9 Transkonduktansforsterkeren i nevronet har derfor ikke et så stort lineært området som figuren gir inntrykk av.

Figur 5.11 viser måling av både utsignalet o_j fra nevronet og o'_j, den deriverte av o_j. o_j er signalet fra den vanlige transkonduktansforsterkerutgangen, mens o'_j er bump-utgangen fra transkonduktansforsterkeren. Vi ser at de to signalene er forskjøvet mellom 120mV og 150mV. Det skyldes terskelen Θ_j. Imidlertid er det en forskyvning, ca. 30mV, de to signalene i mellom. Det skyldes transistorvariasjoner.

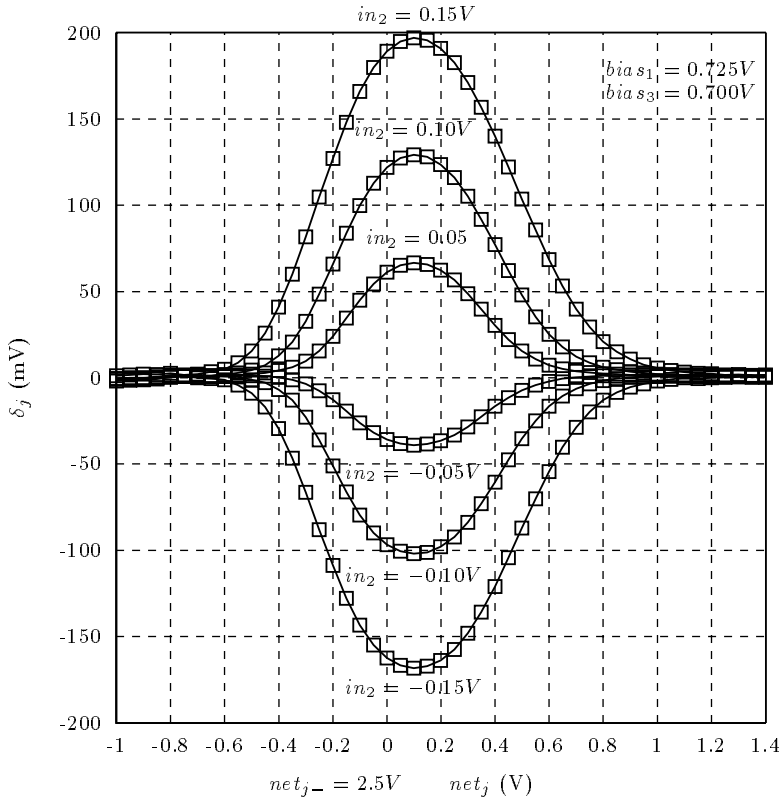
Jeg har også gjort målinger av nevronets andre utsignal δ_j, som er en multiplikasjon mellom o'_j (bump-signalet) og $\sum_k \delta_k w_{kj}$, som er et uttrykk for feilen i overliggende lag.



Figur 5.10: Strømutgangen o_j fra tre nevroner som funksjon av net_j og med initielle verdier av Θ_j. Terskelen Θ_j er årsaken til den store forskyvningen langs x-aksen. De tre kurvene har en forskyvning på henholdsvis ca. 70mV (chip#6), 150mV og 210mV.



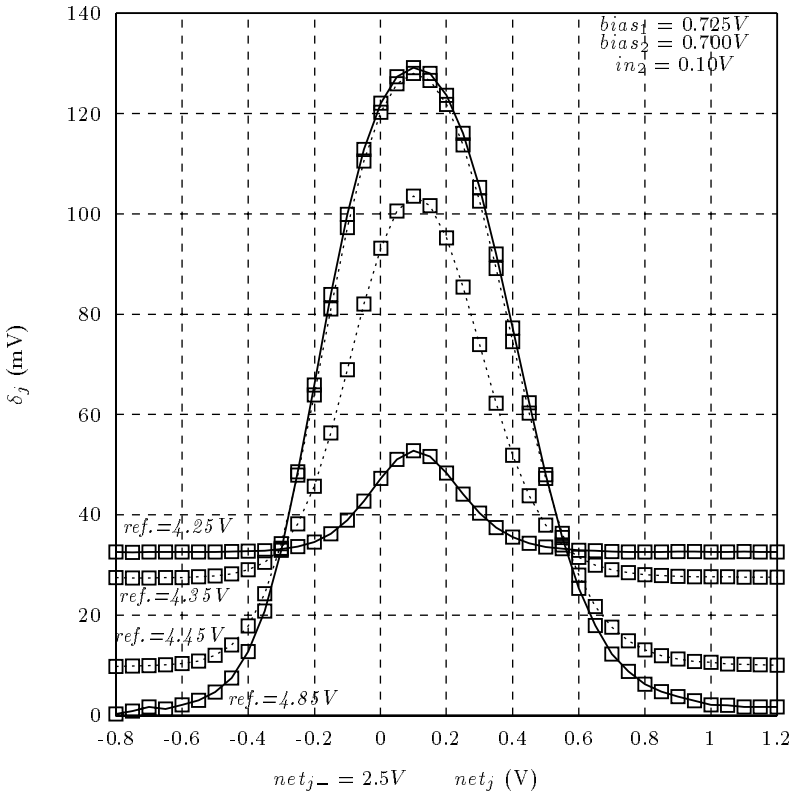
Figur 5.11: Strømutgangene o_j og o'_j fra et nevron som funksjon av net_j og med initielle verdier av Θ_j. Terskelen Θ_j er årsaken til forskyvningen langs x-aksen.



Figur 5.12: Utgangen δ_j fra et nevron som funksjon av net_j ved 6 forskjellige spenninger for $\sum_k \delta_k w_{kj}$. Terskelen Θ_j er årsaken til forskyvningen langs x-aksen. δ_j er en multiplikasjon mellom o'_j og $\sum_k \delta_k w_{kj} = in2$. Kurver med et positivt signal δ_j, har et positivt innsignal $in2$. Kurver med et negativt signal δ_j, har et negativt innsignal $in2$. En stor $in2$ gir en stor δ_j.

Utsignalet δ_j er i form av et differensielt spenningssignal. Jeg varierte innsignalet net_j og målte δ_{j-} og δ_{j+}. Målingene i figur 5.12 viser δ_j som er differansen mellom δ_{j+} og δ_{j-} som funksjon av net_j for seks konstante verdier av $\sum_k \delta_k w_{kj}$. Vi ser at signalene er tilnærmet proporsjonale og at multiplikatoren dermed virker tilfredstillende. Både målingene i figur 5.11 og 5.12 er utført på *chip*#4. Vi ser at signalene i begge figurene er forskjøvet ca. 100mV som en følge av terskelen Θ_j.

Ref.spenning setter spenningen på gaten på transistoren merket Q_{ref} i figur 4.3 som gir referansestrøm til bump-signalet. Bump-signalet kan da konverteres til en differensielt spenningssignal, men referansen må velges i den nedre delen av subterskelområdet for at konverteringen skal fungere tilfredstillende. Figur 5.13 viser δ_j ved *ref.spenning*= $4.85V, 4.35V$ og $4.25V$. Spenningene er med referanse til V_{DD}. Målinger viser at en *ref.spenning* i området 4.90V og ned til 4.50V gir et δ_j-signal som stabiliserer seg nær 0V for små og store verdier av net_j som er det ønskelige. Større verdier av *ref.spenning* vil forflytte det relative nullpunktet for δ_j slik som vi ser i figur 5.13. Dessuten vil signalet svekkes. *Ref.spenning* setter gate spenningen på transistoren i figur 4.3 som gir referansestrømmen. Det vil være en stor spenning V_{ds} over transistoren som gir referansestrømmen siden V_d for transistoren vil befinne seg en diode-offset over *GND* (ca. 0.7V) som følge av etterfølgende diodekobling med en n-kanal transistor. Den store V_{ds}-spenningen vil medføre



Figur 5.13: Utgangen δ_j fra et nevron som funksjon av net_j ved 4 forskjellige ref.spenninger. Valg av ref.spenning er viktig. Ref.spenning er i forhold til V_{DD}. Kurven som stabiliserer seg ved $\delta_j = 0V$ for store negative og positive verdier av net_j, har en ref.spenning=4.85V. Desto større ref.spenning (i forhold til V_{DD}) vil føre til at δ_j vil stabiliseres med et desto større avvik som er uønsket.

et stort utslag i Early-effekten. Ved en økende *ref.spenning*, vil referansestrømmen øke. Vi ser av figur 3.2 at ved økende V_{gs} (*ref.spenning*), så vil absoluttverdien av Early-effekten øke. Ved en stor V_{gs} vil det være en stor forskjell i strømmen gjennom referansetransistoren bestemt av størrelsen av spenningen over transistoren V_{ds}. Transistoren Q_{ref} i figur 4.3 som gir referansestrømmen har en stor V_{ds}. I tillegg kommer virkningen av Early-effekten i speil og transistorvariasjoner. Følgen er at out'_+ i figur 4.3 får et for stort avvik i forhold til out'_- (*ref.spenning*) og det differensielle bump-signalet (out') vil ikke stabilisere seg nær nok 0V for små og store innsignaler. Dette forplanter seg slik at vi ser denne effekten på utgangen δ_j av multiplikatoren. For å unngå at Early-effekten ikke skal bli for dominerende slik at referansen ikke har ønsket grad av nøyaktighet, må *ref.spenning* settes slik at referansestrømmen ikke blir stor i forhold til bump-strømmen. Referansen vil da fungere tilfredstillende.

5.2.3 Programmering av terskelen i et nevron - målinger

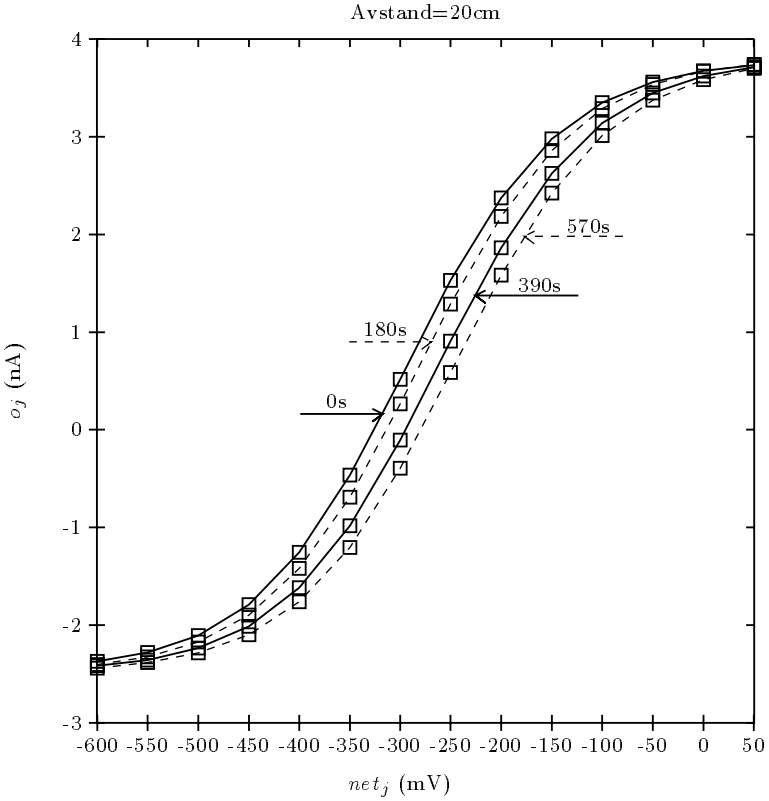
Jeg har programmert endringer av terskelen Θ_j i et nevron. Under hver programmeringsperiode påtrykte jeg innsignalene net_j og $\sum_k \delta_j w_{ji}$ spenninger slik at jeg fikk en positiv endring av terskelen. Arbeidsområdet for innsignalene valgte jeg for $net_j = 2.5V$ og for $\sum_k \delta_j w_{ji} = 3.0V$ Jeg satte *ref.spenning=4.85V* og forspenningene $bias_1 = bias_2 = bias_4 = 0.75V$. Mellom hver programmeringsperiode målte jeg utsignalet o_i fra nevronet som funksjon av net_j. Vi ser i figur 5.14 at overføringsfunksjonen forskyves mot venstre langs net_j-aksen med økende programmeringstid som et resultat av positiv endring av verdien av terskelen Θ_j.

5.2.4 Forbedringer

Bidraget som hvert av de veide innsignalene $w_{ji}o_i$ bidrar med til det totale innsignalet net_j til respektive nevron, er alle skalert med en forspenningsstrøm I_b slik som gitt i likning 4.3 som er likningen for den multiplikasjonen som Gilbert multiplikatoren utfører. Forspenningsstrømmen I_b blir av multiplikatoren som beregner $w_{ji}o_i$ bestemt av forspenningen $bias_3$.

For terskelen Θ_j i et nevron utføres det ikke en tilsvarende skalering av dens bidrag. Det er derfor sannsynlig at bidraget Θ_j fra terskelen til innsignalet til transkonduktansforsterkeren som gir utsignalet o_j fra nevronet er vesentlig større enn bidraget net_j fra de veide innsignalene $w_{ji}o_i$. Det betyr at summen $(net_j + \Theta_j)$ uansett programmeringstid aldri vil nærme seg null. Utgangssignalet o_j fra nevronet vil derfor aldri få et omslag fra "lavt" til "høyt" nivå eller omvendt. Utgangen vil da konstant befinne seg på et "lavt" eventuelt " høyt" nivå. Et nett kan da konvergere, men uten å ha lært de påtrykte inngangsmønstrene.

Problemet kan sannsynligvis løses ved å sørge for at Θ_j blir skalert på tilsvarende måte som $w_{ji}o_i$. Det kan gjøres ved å la Θ_j være innsignal til en transkonduktansforsterker slik som vist i figur 4.1. Utgangen fra transkonduktansforsterkeren summeres deretter som et strømsignal med net_j. Det kan være en fordel å la den nevnte transkonduktansforsterkeren ha en egen forspenning $bias_5$ slik at skaleringen av bidraget Θ_j kan foregå uavhengig av andre beregninger.



Figur 5.14: Endring av terskelen Θ_j i et nevron som følge av en programmering. Vi ser at terskelen Θ_j forskyver overføringsfunksjonen o_j etter hvert som den endres.

Multiplikatoren som beregner endringen av terskelen kan sannsynligvis også med fordel erstattes av en transkonduktansforsterker siden dens oppgave også kun er en skalering nærmere bestemt av δ_j.

5.2.5 Oppsummering og konklusjon

Målinger av utsignalet o_j fra nevronet viser en aktiviseringsfunksjon av ønsket kvalitet. Også multiplikasjonen som gir forplantet feil δ_j er av tilfredstillende kvalitet. Konvertering av o_i, den deriverte av utsignalet o_i, fra et strømsignal til et differensielt spenningssignal fungere tilfredsstillende ved passe valg av referansesignal. Målinger viser også at verdien av terskelen Θ_j endres i ønsket retning som en følge av en programmering slik at nevronets utsignal o_i forskyves som forventet langs net_j-aksen. Målinger viser altså at alle nevronets funksjoner isolert sett utføres tilfredstillende.

Programmering for endring av terskelen i et enkelt nevron gir oss imidlertid ikke grunnlag for å si noe om størrelsen av bidraget Θ_j i forhold til bidraget net_j fra alle de veide innsignalene $w_{ji}o_i$ til nevronet. Vi vet derfor ikke nok om hvorvidt terskelen utfører sin oppgave tilfredstillende eller ikke i en større sammenheng. Imidlertid er det mye som tyder på at terskelens bidrag blir stort i forhold til bidraget net_j slik at utgangen o_j fra nevronet blir liggende på et fast "lavt" eventuelt "høyt" nivå når nevronet benyttes i et nett.

Kapittel 6

Sammenkobling av moduler til et nevralt nett

Etter å ha definert modulene vekt og nevron, er det enkelt å benytte disse modulene til å sette sammen et nevron med vilkårlig mange vekter tilknyttet. Videre vil et vilkårlig lag i nettet bestå av en eller flere nevroner med tilhørende vekter, og til slutt vil et nett bestå av to eller flere lag. Ved å definere vekt og nevron, har jeg oppnådd en modularisering av nevrale nett i flere nivåer. Det er dermed en enkel sak å sette sammen et vilkårlig lagdelt nett. De to definerte modulene gir altså mulighet til å realisere store analoge beregningssystemer i mikroelektronikk.

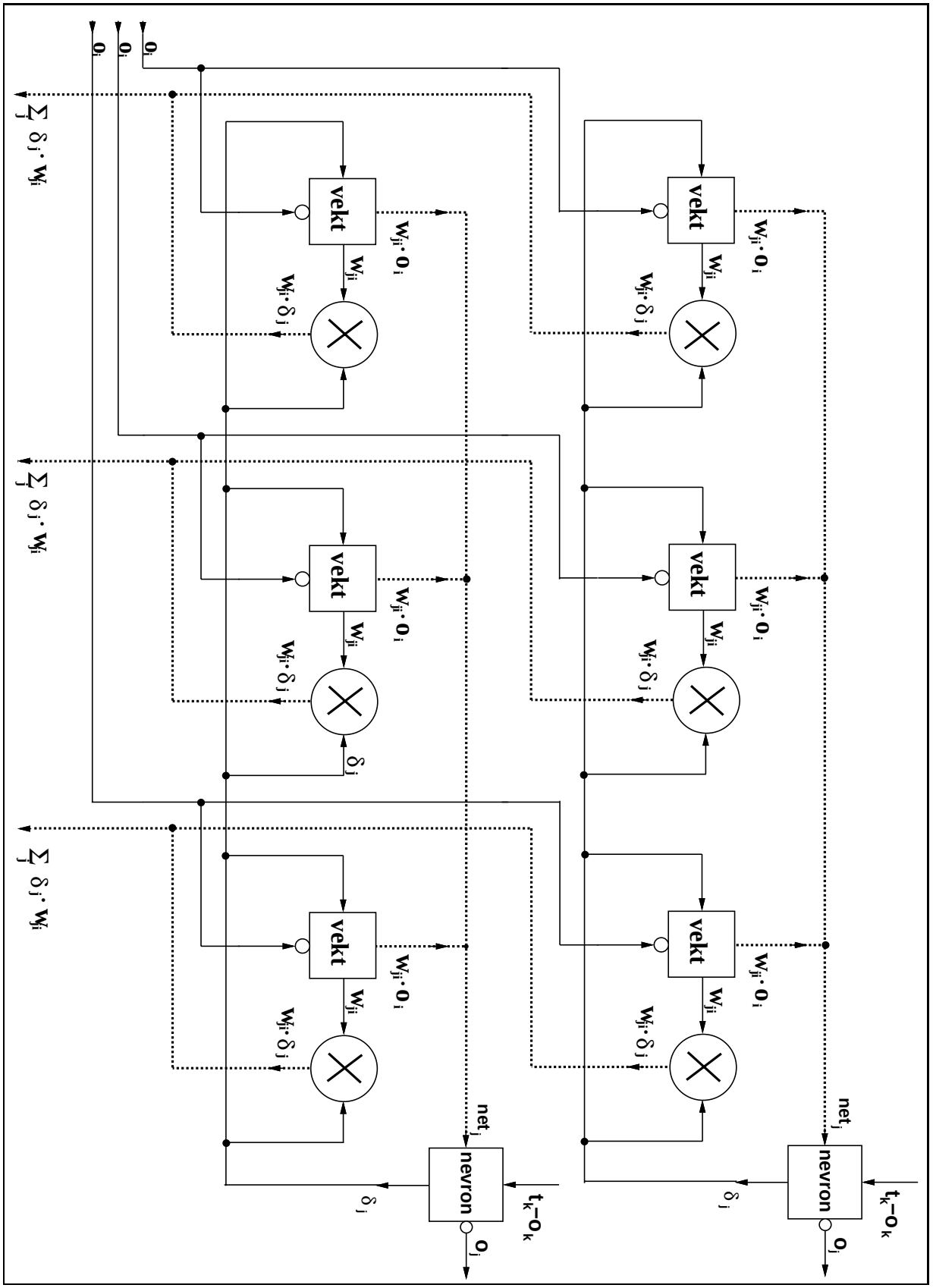
Når det gjelder programmering av et nett er det noe usikkerhet om hvorledes den bør foregå for å oppnå tilfredstillende opplæring av nettet. Det er mange faktorer og problemstillinger som bør studeres nærmere i forbindelse med en opplæring av et nett. Jeg vil derfor nøye med med å skissere noen av problemstillingene som må studeres nærmere i forbindelse med en programmering av et nett. Imidlertid har jeg realisert nettet i figur 2.5 i analog VLSI og utført noen enkle forsøk på å programmere det.

6.1 Oppbygging av et lag

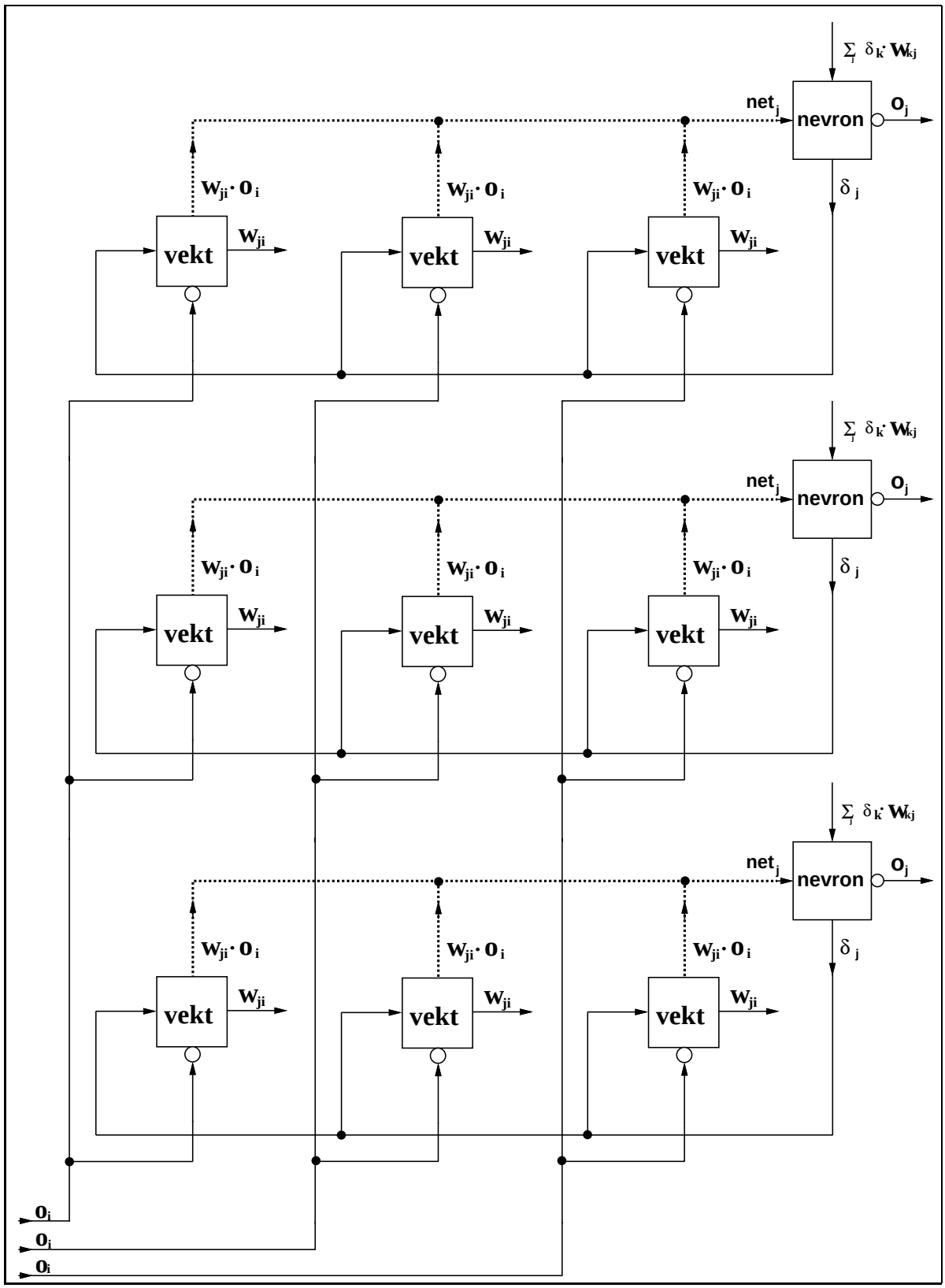
Et nevron vil ha tilknyttet et antall vekter etter hvor mange inngangssignaler det er til nevronet. I tillegg kan det være behov for en ekstra multiplikator for hver av vektene som har behov for å beregne sitt bidrag til forplantet feil δ_j ned til vektens underliggende nevron.

Et nevron i utgangslaget av et nett med f.eks tre inngangssignaler, vil ha behov for tre vekter med hver sin ekstra multiplikator for å beregne vektens bidrag til forplantet feil. Utgangssignalet $o_i w_{ji}$ fra hver av vektene vil summeres sammen til nevronets totale inngangssignal net_j. Nevronet har i tillegg til sigmoidsignalet o_i et utgangssignal δ_j som distribueres som inngangssignal til nevronets tilhørende vekter i forbindelse med endringen av vektene.

Et vilkårlig lag i et nevralt nett vil bestå av en eller flere nevroner med tilhørende vekter. Figur 6.1 viser et eksempel på et utgangslag med to nevroner som hver har tre tilhørende



Figur 6.1: Eksempel på et utgangslag med to nevroner og tre innsignaler til hvert nevron. Alle linjer representerer differensielle signaler. Heltrukkene linjene er spenningssignaler, mens stiplede linjer er strømsignaler.



Figur 6.2: Eksempel på et skjult lag med tre nevroner og tre innsignaler til hvert nevron. Alle linjer representerer differensielle signaler. Heltrukkene linjene er spenningssignaler, mens stiplede linjer er strømsignaler.

vekter. Her er begge de definerte modulene tatt i bruk i tillegg til enkle multiplikatorer. Alle linjer representerer differensielle signaler. Heltrukkene linjene er spenningssignaler, mens stiplede linjer er strømsignaler. Figur 6.2 viser et eksempel på et skjult lag med tre nevroner i et nett med et skjult lag. Laget består av tre nevroner hver med tre vekter. Det er stort sett likt som for et utgangslag bortsett fra at ingen av de tre nevronenes vekter har behov for en ekstra multiplikator. Alle linjer i figuren har samme betydning som i figur 6.1.

6.2 Forspenninger

Valg av forspenninger til de ulike kretselementene i et nett vil ha stor betydning for beregningene i nettet. Spesielt valg av forspenning til multiplikatorene som beregner endringer av en vekt vil ha stor betydning. Ved å justere denne forspenningen kan vi regulere opplæringshastigheten. Det vil være behov for en del justeringer av alle forspenningene spesielt i forbindelse med en opplæringsfase, for at globale beregninger skal utføres tilfredstillende.

Forspenningene til kretselementene vil på en måte om ikke direkte representere enkelte parametre som inngår i algoritmen. F.eks. vil justering forspenningen til en multiplikator som beregner vektendringer regulere programmeringshastigheten av vekten noe tilsvarende η i likning 2.4. Justering av forspenninger til andre kretselementer vil også ha stor innflytelse på beregningene i nettet.

Multiplikatorene anvendes i ulike forbindelser, bla. for beregning av et veid inngangssignal og beregning av en vekts endring. Multiplikatorene som beregner en vekts endring lar jeg ha en felles forspenning, mens de resterende multiplikatorene en annen felles forspenning. Alle nevroner har en transkonduktansforsterker for å gi et utgangssignal. For disse forsterkerene er det naturlig å ha en felles forspenning. I hvert hukommelseselement inngår det også en forsterker koblet som en spenningsfølger. Forspenningen til disse spenningsfølgerene lar jeg også være felles.

6.3 Opplæring

Beregningene i forbindelse med en opplæring av et nett er implementert med utgangspunkt i den generaliserte delta regelen. Imidlertid er beregningene av endringer Δw_{ji} av vektene w_{ji} i praksis nærmest redusert til kun å gi uttrykk for fortegnet til endringen slik som beskrevet i avsnitt 5.1.1.

Endringen av vektene i nettet vil skje i henhold til påtrykt inngangs- og ønsket utgangsmønster når kretsen bestråles med UV-lys. Siden beregnet endring av en vekt er redusert til kun å gi uttrykk for endringens fortegn, vil vekten trekkes oppover når endringer er positiv og nedover når endringen er negativ. Størrelsen av endringen bestemmes av tiden vekten blir utsatt for UV-lys inntil uttrykket for endringen av vekten skifter fortegn. En viktig faktor i forbindelse med programmeringshastigheten av vekten, er avstanden fra den integrerte kretsen og til lyskilden. Effekten av lyset avtar eksponentielt med avstanden til

det som belyses. Det er ikke mulig å kun belyse enkelte vekter av gangen. Det er alltid hele kretsen som belyses, dvs. alle hukommelseselementene siden aktivt område ellers på det integrerte kretsen er dekket med et skjermende metall2 lag.

6.3.1 Alternative programmeringsmetoder

I utgangspunktet så jeg for meg en svært enkel programmeringsmetode for programmering av et nett. Når inngangs- og ønsket utgangsmønster påtrykkes, var tanken å sette på UV-lyset og la det bestråle den integrerte kretsen inntil ønsket og faktisk utgangsmønster er tilstrekkelig like. Hvorvidt ønsket og faktisk utgangsmønster er tilstrekkelig like må avgjøres eksternt. Etter å ha påtrykt et par med inngangs- og ønsket utgangsmønster, kan neste par med inngangs- og ønsket utgangsmønster påtrykkes inntil alle parene i opplæringssettet er presentert.

Endringen av en vekt pr. tidsenhet i nettet bør være av minst en orden langsommere enn beregningene i nettet med hensyn på hastighet slik at signaler som er resultater av beregninger forplantes raskere enn tiden det er behov for å endre en vekt med et "minste inkrement/dekrement". Dersom ikke kravet blir oppfylt, kan resultatet bli at når det er behov for å skifte vektendringens retning, vil skifte av retning komme så sent at i mellomtiden har vekten blitt endret alt for langt i feil retning. Den nye retningen for vektendringen vil hele tiden komme på etterskudd, og vi vil få en oscillasjon. For å hindre en slik situasjon, kan belysningseffekten reduseres inntil vektendringen skjer langsomt i forhold til beregningene og forplantningen av signaler i nettet.

En alternativ opplæringsmetode vil være kun å belyse den integrerte kretsen i små perioder av bestemt varighet. For hver periode presenteres neste mønster i opplæringssettet Etter at hele opplæringssettet er presentert en gang presenterer vi opplæringssettet forfra igjen helt til nettet begynner å konvergere. Vektene vil ved en slik metode endres oppover eller nedover med en faktor som vi kan betrakte som minste inkrement av en vekt. Vi oppnår dermed en inkrementell endring av vektene ved å bruke UV-lyset som en form for klokkesignal. Et av målene var i utgangspunktet å unngå en hver form for klokking av signaler, imidlertid vil nettet ved en klassifisering fremdeles foregå uten noen form for klokking av signaler. I tillegg til lengden av en belysningsperiode, vil bla. belysningseffekten og forspenningen til multiplikatoren som beregner endringer av en vekt ha innvirkning på størrelsen av inkrementet.

Et inkrement vil ikke alltid være helt fast ved en gitt belysningstid, belysningseffekt og forspenning til multiplikatoren som beregner endringer. Årsaken er at spenningsforskjellen mellom kontroll noden og floating gate noden vil variere noe. Vi vet fra før at programmeringshastigheten reduseres ved en liten spenningsforskjell mellom de to nevnte noder. Sannsynligvis vil det i første omgang være en fordel å gjøre en forenkling og anta at størrelsen av et inkrement ikke er avhengig av spenningsforskjellen mellom kontroll noden og floating gate noden i hukommelseselementet. Når arbeidsområdet for en vekt beveger seg innenfor ca. 100mV vil forenklingen være svært god. Målinger viser at en vekt har en svært lineær endring innenfor ca. 100mV total endring.

Når kretsen belyses med UV-lys en gitt belysningstid, vil vektene endres oppover eller nedover med et gitt inkrement (dekrement). Etter at lyset er av, vil de nye signalene i nettet

som følge av vektendringen kunne forplantes videre i nettet. Ny endring med det gitte inkrementet (dekrementet) blir dermed beregnet før kretsen igjen blir belyst. Når kretsen igjen blir belyst, vil nye endinger av vektene med det gitte inkrementet (dekrementet) utføres.

Den generaliserte delta regelen endrer vektene med et vilkårlig inkrement (dekrement). Denne programmeringsmetoden derimot vil endre vektene med et fast inkrement (dekrement). Det medfører vesentlig flere iterasjoner før nettet vil konvergere. En mulighet for å redusere antall iterasjoner er å la belysningsperioden være stor i utgangspunktet for så å redusere den etter hvert. Et problem kan være at vektendringer som utføres når et mønster presenteres vil kunne bli kansellert av vektendringen for neste mønster siden vektendringen pr. tidsenhet er fast og ikke vilkårlig. Resultatet kan bli at nettet konvergerer uten å ha lært mønstrene.

En mulighet ved en slik situasjon er å presentere opplæringssettet n ganger og la programmeringstiden for de ulike mønstrene innen opplæringssettet være forskjellig, men at den totale programmeringstiden for hvert mønster etter n presentasjoner blir den samme for alle mønstrene. Sannsynligvis vil en slik programmeringsmetode hindre at vektendringen for to mønstre kansellerer hverandre, men at vi får en netto endring av vekten. Det kan også tenkes at kombinasjoner av de skisserte programmeringsmetodene vil være et interessant alternativ.

6.3.2 Avgjørende faktorer for at en opplæring skal bli vellykket

En vesentlig faktor for at et nett skal la seg læres opp er størrelsen av Gilbert multiplikatorens lineære område. I praksis må vektenes arbeidsområde være av omtrent samme størrelsesorden som multiplikatorenes lineære område. Målinger av utsignalet fra multiplikatoren beskrevet i avsnitt 4.3.1 viser et lineært område av en størrelsesorden omtrent lik 80mV. Dersom vi lar vektenes arbeidsområde være noe større enn det, f.eks. 100mV, kan vi anta at en vekts arbeidsområde bør ligge innenfor et område fra -50mV og til 50mV. En vanlig antakelse er at en vekt bør ha en oppløsning tilsvarende 8 bit (256 nivåer) for at et nett vil konvergere under en opplæringsfase. Et inkrement (minste oppløsning) bør dermed ikke være større enn 0.4mV. Målinger av hukommelseselementet tyder på at en slik oppløsning er innen rekkevidde.

Å programmere minste inkrement til kun 0.4mV vil muligens være et problem. Et større inkrement vil øke mulighetene for at nettet ikke vil konvergere under en opplæringsfase. Det er imidlertid muligheter for å øke multiplikatorens lineære område med bruk av source degenerering og kapasitiv divisjon. Dermed kan også minste inkrement økes og det vil bli lettere å programmere et slik inkrement.

Arbeidsområdet til terskelen i et nevron og hvordan terskelen endres i forhold til vektene som veier innsignalene til nevronet er svært avgjørende. Dersom verdien av terskelen i nevronet endres slik at utgangssignalet fra hvert nevron alltid vil være "høyt" eventuelt "lavt" uansett valg av inngangsmønster, så kan nettet konvergere uten å ha lært de presenterte mønstrene.

Et annet problem er hva som skjer når nettet begynner å konvergere. Innsignalet for endring av vektene vil da begynne å nærme seg null. Som følge av at signalene på

nodene V_{cg} og V_{cap} i hukommelseselementet er usymmetriske, vil et inngangssignal lik null til hukommelseselementet føre til at floating gate noden V_{fg} blir ustabil. Hva det medfører er det vanskelig å si noe bestemt om.

6.4 Xor-nettet

Figur 2.5 viser nettarkitekturen til et nett som er i stand til å læres opp til å kunne løse det klassiske xor-problemet. Med den modulariseringen jeg har oppnådd, er det enkelt å sette sammen et slikt nett ved å ta i bruk de to definerte modulene vekt og nevron for en implementasjon i analog VLSI. Målet er da å kunne programmere vektene i nettet med bruk av UV-lys slik at nettet løser det klassiske xor-problemet.

Xor-nettet i figur 2.5 består kun av et nevron i det skjulte laget og et nevron i utgangslaget i tillegg til to enkle nevroner i inngangslaget. Utgangen fra begge nevronene i inngangslaget som er enkle transkonduktansforsterkere, er inngang til både nevronet i det skjulte laget og i utgangslaget. Det medfører at det er behov for fem vekter og to nevroner som betyr at nettet utleggsmessig blir relativt lite. I første omgang vil et lite nett være hensiktsmessig å realisere i analog VLSI spesielt for å kunne holde oversikt over hva som skjer i nettet i forbindelse med en programmering. Blokkskjema av nettet er vist i figur 6.3. Alle linjer i figuren representerer differensielle signaler. Heltrukkene linjer representerer spenningssignaler, mens stiplede linjer representerer strømsignaler.

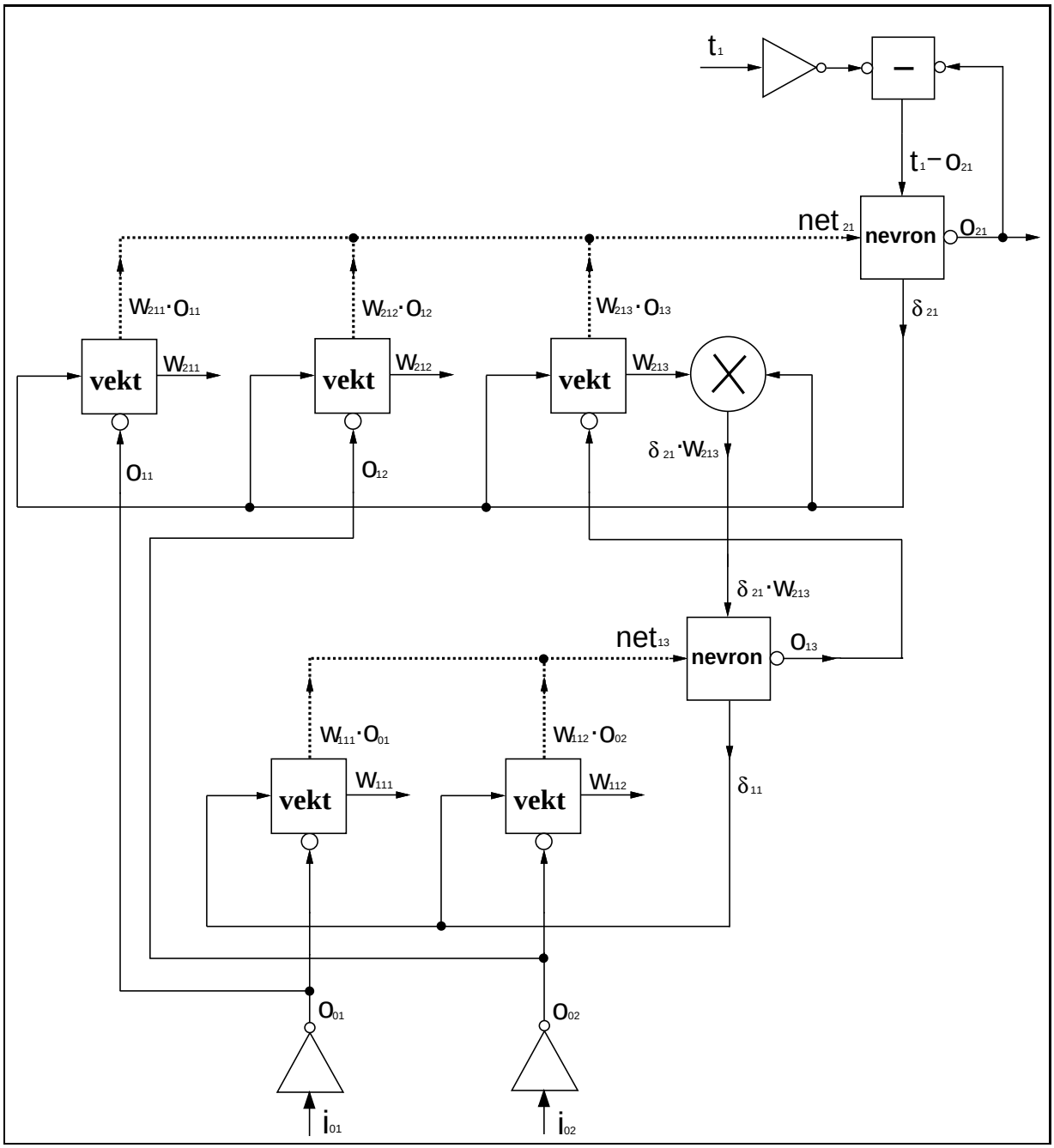
Vi ser av figuren at det er en kun vekt i utgangslaget som får sitt innsignal o_{13} fra nevronet i det skjulte laget, og ikke direkte fra inngangslaget, som har en ekstra multiplikator. Signalet δ_{21} er utsignal fra nevronet i utgangslaget og er det ene inngangssignalet til multiplikatoren.

Nettet har to differensielle spenningsinnganger som blir skalert i respektive nevroner i inngangslaget. Ønsket utgangssignal t_1 påtrykkes som et differensielt spenningssignal og skaleres av en transkonduktansforsterker på samme måte som for inngangssignalene til nettet.

Subtraksjonskretsen i figuren beregner feilen $(t_1 - o_{12})$ som er innsignal til nevronet i utgangslaget.

Multiplikatoren som beregner endringen av terskelen Θ_j i nevronene, har δ_j som det ene inngangssignalet slik som andre vekter, mens det andre inngangssignalet er spenningsdifferansen mellom to diodekoblinger i kaskade. Strømmen gjennom de to diodekoblingene blir regulert av samme forspenning som for forsterkerene som gir utgangssignal fra nevronene.

Det skjer ingen beregninger internt i nettet eller på kretsen ellers som avgjør når opplæringsfasen skal avsluttes. Opplæringsfasen pågår så lenge den integrerte kretsen belyses med UV-lys, og det vil da foregå en kontinuerlig endring av vektene. Ved simulering med programvare på en generell datamaskin er det som oftest likning 2.3 for beregning av systemfeilen E som avgjør når læringen skal avsluttes.



Figur 6.3: Blokkskjema for et nett som er ment å løse xor-problemet. Alle linjer representerer differensielle signaler. Heltrukkene linjene er spenningssignaler, mens stiplede linjer er strømsignaler. Symbolene som påtrykkes i_{01}, i_{02} og t_1 representerer transkonduktansforsterkere. Bobler på symbolene betyr at signalet er med referanse til V_{DD}.

6.4.1 Simulering av xor-nettet

Jeg har ikke hatt mulighet til å simulere hele xor-nettet. Det skyldes at det ikke har vært tilgjengelig en kraftig nok simulator til å kunne simulere en krets av en størrelsesorden som mitt xor-nett. Et annet problem er at det ikke fantes en god modell for UV-strukturer. [Maher] har forøvrig løst det siste problemet, men hennes modell for UV-strukturer er foreløpig ikke implementert i en simulator.

[Berg] har utviklet en simulator HADES for simulering av nevrale nettverksmodeller både i det nevrale domenet og i det elektroniske domenet. Simulatoren tillater noder i det nevrale domenet å kommunisere med noder i det elektroniske domenet. Dermed kan nevrale nettverksmodeller simuleres med deler av nettverket representert med modeller for elektroniske kretser slik som Gilbert multiplikatoren.

Årsaken til at jeg ikke har anvendt HADES, er at simulatoren foreløpig mangler et brukergrensesnitt. Men ikke minst må simulatoren ha implementert en modell for UV-strukturer for at jeg skal ha nytte av den. Det er nettopp hukommelseselementene (UV-strukturene) som er det mest kritiske i mitt nett, spesielt med tanke på størrelsen og oppløsningen av hukommelseselementenes arbeidsområde. Timing av signaler innen nettet er også avgjørende under en programmeringsfase.

Imidlertid har [Berg] simulert nettopp xor-nettet i såkalt integrert nevral og elektronisk simulering. Multiplikasjonen med vektene foregår med Gilbert multiplikatorer med differensielle strømutganger, og signalet ut fra nevronene beregnes av differensielle par med diodekoblinger på utgangene (tilsvarer det jeg omtaler som transkonduktansforsterkere med differensiell spenningsutgang). Opplæringen av nettet utføres i henhold til delta-regelen. [Berg] lar oppdateringen av vektene skje etter at hele settet med inngangsmønstre er påtrykt. Simuleringene ga svært lovende resultater.

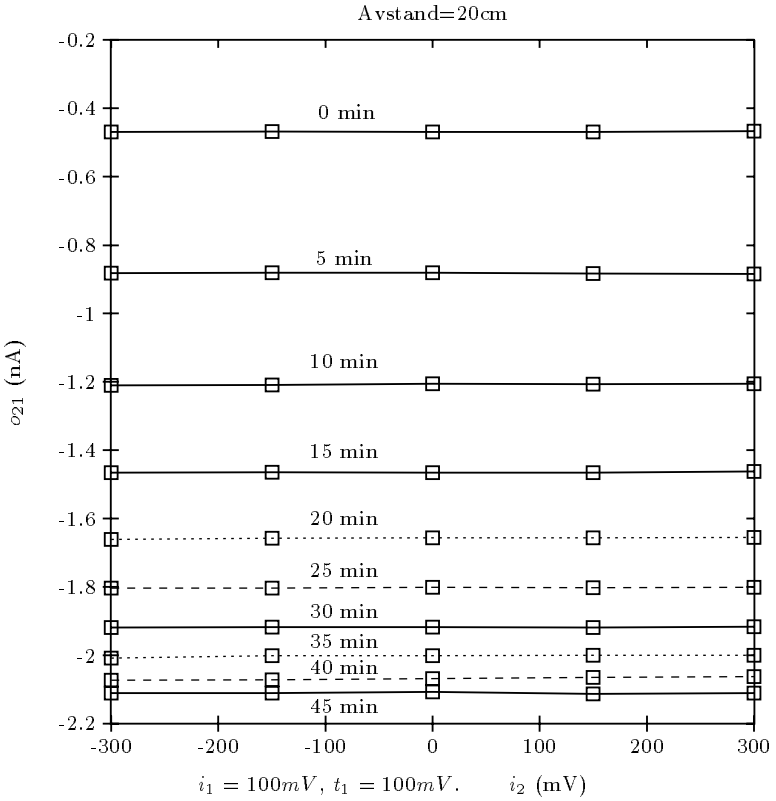
Det ville vært svært interessant å simulere elektronisk i HADES hele min implementasjon av xor-nettet. Det har dessverre ikke vært mulig til nå.

6.4.2 Forsøk på opplæring - målinger

Jeg har utført noen enkle forsøk på å lære opp xor-nettet. Xor av to inngangssignaler gir fire kombinasjoner av inngangssignaler til nettet. Når de to inngangssignalene har samme "logiske nivå" skal utgangen ha et "høyt" nivå, mens når de to inngangssignalene har forskjellig "logisk nivå" skal utgangen ha et "lavt" nivå. Jeg anvendte den første programmeringsmetoden beskrevet i avsnitt 6.3.1 og påtrykte hvert av de fire inngangsmønstrene sammen med ønsket utgangsmønster for en lengre periode mens jeg programmerte nettet.

Jeg lot et innsignal lik 100mV representere et "høyt" nivå eller logisk "1", mens et innsignal lik -100mV representere et "lavt" nivå eller logisk "0". Jeg satt forspenningene $bias_1 = bias_2 = bias_4 = 0.75V$ og $bias_3 = 0.80V$ mens jeg satt $ref.spenning = 4.85V$. Jeg lot arbeidsområdene for innspenningene være $i_{1-} = i_{2-} = t_{1-} = 2.5V$. Jeg påtrykte hvert av de 4 mønstrene en time med en avstand lik 20cm mellom krets og lyskilde og utførte målinger hvert femte minutt mens jeg varierte det ene av innsignalene i_1 og i_2 og holdt det andre fast.

Jeg kan gjøre målinger av utgangen o_{21} fra nettet både som et differensielt spenningssig-



Figur 6.4: Forsøk på programmering av et mønster med $i_1 = 100mV$, $i_2 = -100mV$ og ønsket utgangsmønster $t_1 = 100mV$. Figuren viser målinger av utgangen o_{21} som funksjon av i_2 og med $i_1 = 100mV$ mellom hver programmeringsperiode på 5 minutter. Vi ser at endringer av utsignalet o_{21} avtar ved økende programmeringstid, men at o_{21} er konstant som funksjon av i_2.

nal og som et strømsignal. Noe av poenget ved forsøket var å få et inntrykk av hvilket spenningsnivå som representerer et "lavt" signal på utgangen og hvilket spenningsnivå som representerer et "høyt" signal på utgangen. Jeg programmerte derfor nettet over en lengre periode slik at det forhåpentligvis skulle konvergere.

Jeg valgte å utføre forsøket på opplæring av nettet på en krets, *chip#12* som ikke var programmert tidligere. Jeg påtrykte først et mønster hvor $i_1 = 100mV$, $i_2 = -100mV$ og ønsket utgangsmønster $t_1 = 100mV$. Mellom hver programmeringsperiode på 5 minutter utførte jeg målinger av utgangen o_{21} (både som differensielt spenningssignal og som strømsignal), mens jeg varierte inngangen i_2 og lot i_1 vare fast. Målingene av o_{21} (strømsignalet) er vist i figur 6.4. Deretter påtrykte jeg et mønster hvor $i_1 = 100mV$, $i_2 = 100mV$ og ønsket utgangsmønster $t_1 = -100mV$ og utførte tilsvarende målinger som i figur 6.4. Målingene er vist i figur 6.5.

Vi ser i figur 6.4 at utgangen o_{21} fra nettet endrer seg i en bestemt retning og at endringens størrelse avtar ved økende programmeringstid. Endringen av det differensielle spenningssignalet o_{21} avtar noe langsommere enn strømsignalet o_{21} siden utsignalet kommer fra en transkonduktansforsterker. Imidlertid er utgangen tilnærmet konstant som funksjon av i_2

I figur 6.5 ser vi at utgangen o_{21} fortsetter å endre seg videre i samme retning som i figur 6.4 og at o_{21} fremdeles er konstant som funksjon av i_2. Størrelsen av endringen blir imidlertid stadig mindre.

De to siste mønstrene i opplæringssettet har jeg også påtrykt under en programmering og jeg har utført målinger av utsignalet o_{21} hvert femte minutt som funksjon av i_2, men nå med $i_1 = -100mV$. Nettets oppførsel er imidlertid den samme som for de to første inngangsmønstrene.

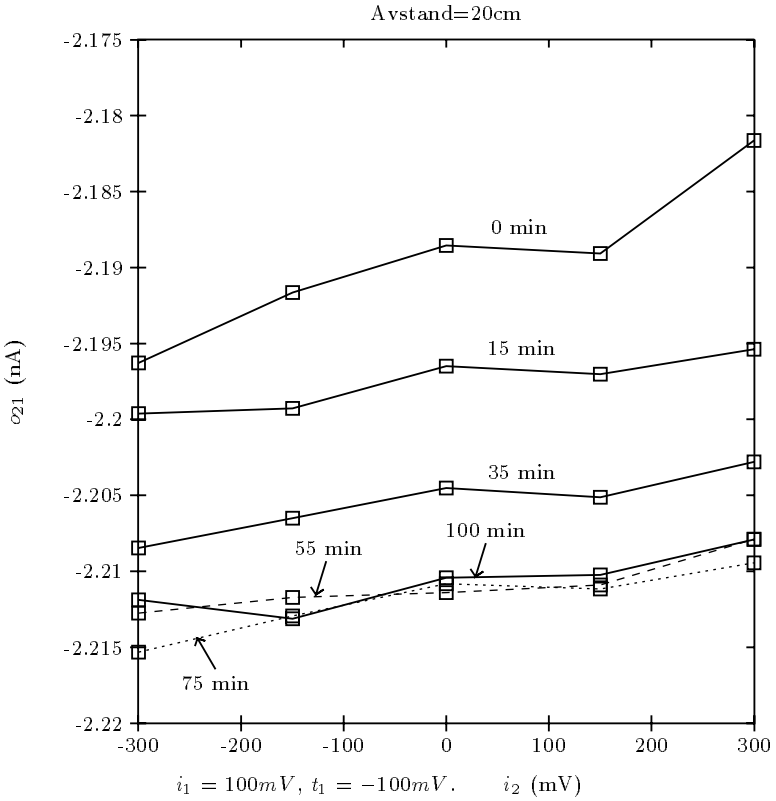
Det kan tyde på at bidraget fra terskelen i nevronet i det skjulte laget og i utgangslaget er stort i forhold til bidraget net_j fra de veide innsignalene til hver av de to nevronene. Utsignalet fra nettet blir derfor liggende konstant, i dette tilfelle med et "lavt" nivå, uansett valg av inngangsmønster. Endringene vi ser på utgangen o_{21} er mest sannsynlig som følge av endringer av terskelen i de to nevronene.

6.4.3 Forbedringer

Det tyder på at terskelen i nevronene er viktigste årsak til nettets uønskete oppførsel. Forbedringer av terskelen er beskrevet i avsnitt 5.2.4. Forbedringer av kontroll noden i hukommelseselementet slik som beskrevet i avsnitt 4.4.4 vil gjøre programmeringen av nettets vekter sikrere samtidig som nettet kan programmeres hurtigere. En forbedring av multiplikatorens lineære område slik som beskrevet i avsnitt 4.3.2 vil sikre økte muligheter for at nettet vil konvergere i tillegg til kravet om en god oppløsning av vektens verdiområde.

6.5 Oppsummering og konklusjon

En stor grad av modularisering gjør det mulig å sette sammen en vilkårlig backpropagation nettarkitektur og realisere den i analog VLSI. Som et eksempel har jeg realisert en nett-



Figur 6.5: Forsøk på programmering av et mønster med $i_1 = 100mV$, $i_2 = 100mV$ og ønsket utgangsmønster $t_1 = -100mV$. Figuren viser målinger av utgangen o_{21} som funksjon av i_2 og med $i_1 = 100mV$ mellom hver programmeringsperiode på 5 minutter. Vi ser at endringen av utgangen o_{21} fortsetter videre i samme retning som for første mønster og at o_{21} fremdeles er konstant som funksjon av i_2. Størrelsen av utgangens endring blir imidlertid stadig mindre og det tyder da på at nettet konvergerer uten å ha lært de påtrykte mønstrene. Årsaken kan være at tersklene i nevronene er alt for dominerende og at utgangen fra nevronene blir konstant liggende på et "lavt" eventuelt "høyt" nivå.

arkitektur som var tenkt å kunne læres opp til å løse det klassiske xor-problemet. Siden det realiserte nettet er kontinuerlig, fører det til en noe utradisjonell tankegang i forbindelse med opplæringsfasen av nettet.

Jeg har skissert noen mulige programmeringsmetoder og ikke minst har jeg utført forsøk på å lære opp xor-nettet. Målinger under programmeringsfasen viser at nettets utgang endrer seg i en bestemt retning ved økende programmeringstid, men at utgangssignalet er konstant uansett valg av inngangsmønster. Imidlertid blir endringen av utgangssignalet stadig mindre ved økende programmeringstid, og det tyder på at nettet konvergerer uten å ha lært de påtrykte inngangsmønstrene. Hvilket inngangsmønster som påtrykkes under programmeringen har ingen betydning for endringen av utgangen. Det kan derfor tyde på at terskelen i nevronet er helt dominerende i forhold til de veide innsignalene til nevronet, og at utgangen fra nevronet derfor har et fast "lavt" nivå eventuelt "høyt" nivå uansett programmeringstid og valg av mønster.

Kapittel 7

Oppsummering og konklusjon

Jeg har vurdert, realisert og utført målinger med generelt gode resultater av grunnleggende kretselementer for beregninger i et analog og kontinuerlig backpropagation nevralt nettverk. Spesielt programmering og måling av hukommelseselementet har gitt svært gode resultater med hensyn på nøyaktighet av en endring og hukommelseselementets oppløsning av aktuelt arbeidsområde. Hukommelseselementet lider av noen svakheter som er mulig å redusere betraktlig. Det vil føre til at det er mulig å øke programmeringshastigheten av hukommelseselementet betraktlig.

Videre har jeg delt opp et nett i to hovedtyper moduler, den første modulen er en vekt for veiing av innsignal til et nevron samt med muligheter for å gjøre styrte endringer av vektens verdi. Den andre modulen utfører beregninger i forbindelse med et nevron samt at den også inneholder nevronets terskel og muligheter for å programmere styrte endringer av terskelen. Jeg har realisert, programmert og utført målinger av de to modulene med gode resultater. Imidlertid lider terskelen i nevronet av den svakhet at det ikke blir skalert på en tilsvarende måte som et veid innsignal. Det kan føre til at terskelen blir for dominerende i forhold til de veide innsignalene til et nevron i et nett slik at nevronets utgang kan forbli fastlåst med et fast signalnivå.

For en programmering av et analogt og kontinuerlig nettverk kan det være behov for utradisjonelle metoder. Jeg har skissert noen muligheter. For å vise de definerte modulene anvendt i et nettverk, har jeg realisert et lite nettverk og utført forsøk på programmere det. Målinger viser da at nettets utgang endres i en bestemt retning uansett valg av inngangsmønster, men at endringen av utsignalet avtar med økende programmeringstid og konvergerer uten å ha lært de presenterte mønstrene. Årsaken tyder først å fremst på være påpekte svakheter ved nevronets terskel.

Totalt sett viser imidlertid de utførte målinger av alle realiserte kretselementer, større moduler og selve nettet generelt sett gode og ikke minst svært interessante resultater.

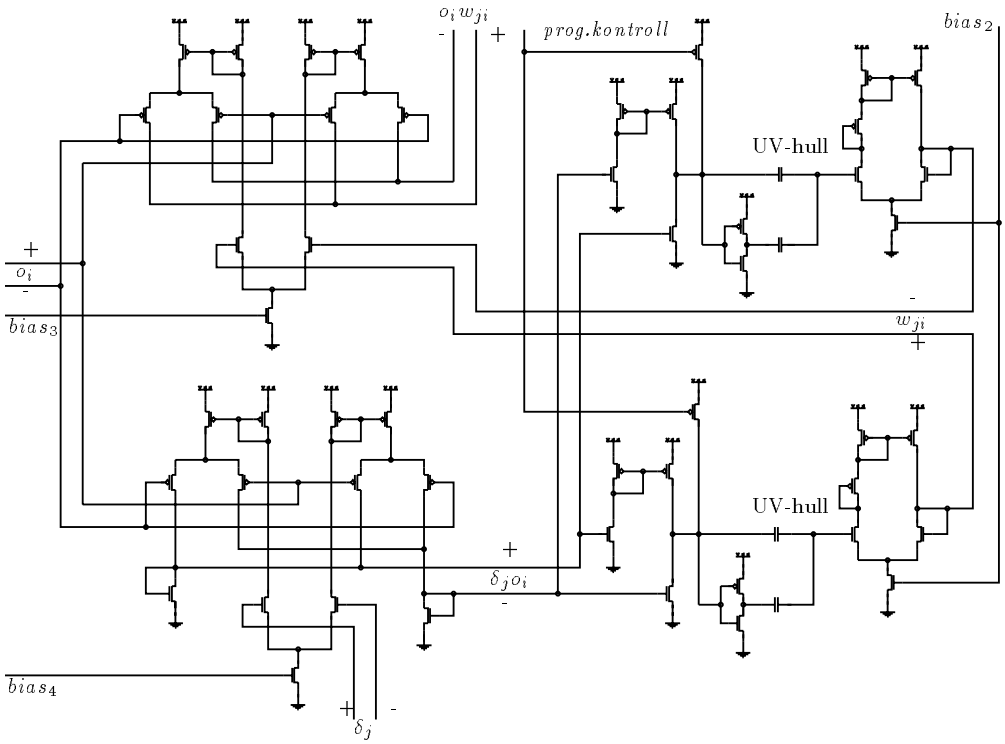
7.1 Videre arbeid

Forholdet mellom terskel i nevronet og de veide innsignalene til nevronet bør studeres nærmere. Det vil være svært interessant å realisere en ny versjon av xor-nettet med omtalte

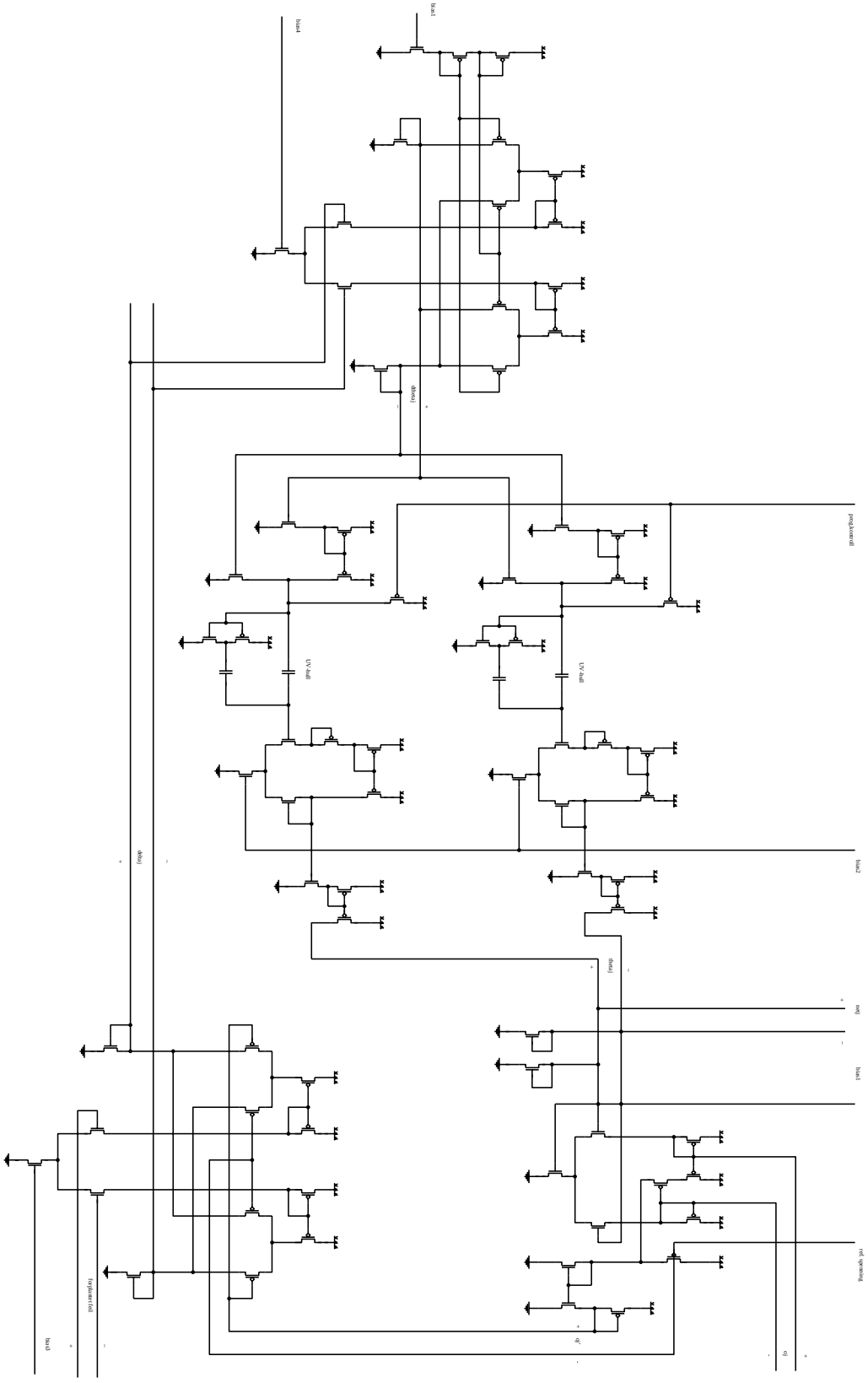
forbedringer. Ved en ny realisering av xor-nettet vil det være en fordel å ha mulighet til å kunne betrakte alle vekter og terskler i nettet.

Vedlegg A

Kretsskjema (krets2)



Figur A.1: Kretsskjema for en vekt.



Figur A.2: Kretsskjema for et nevron.

Vedlegg B

Realisering av nettet i analog VLSI

Jeg har to ganger sendt kretsutlegg til prosessering hos *Orbit*. Begge kretsutleggene har inneholdt xor-nettet i tillegg til enkle testkretser. Den viktigste forskjellen mellom de to utleggene er UV-strukturen i hukommelseselementene.

Utleggsmessig har jeg delt opp nettet i de samme blokkene slik som tidligere beskrevet ved den skjematiske beskrivelsen av nettet i figur 6.3. Et vilkårlig backprop nett består da av et antall vekter og nevroner.

B.1 Utlegg for krets1

I tillegg til xor-nettet, inneholder kretsutlegget for krets1 en enkel vekt og et enkelt nevron. Ved å la kretsutlegget også inneholde en enkel vekt og et enkelt nevron, vil det være enklere å karakterisere de enkelte modulene som en vilkårlig backprop nettarkitektur vil settes sammen av.

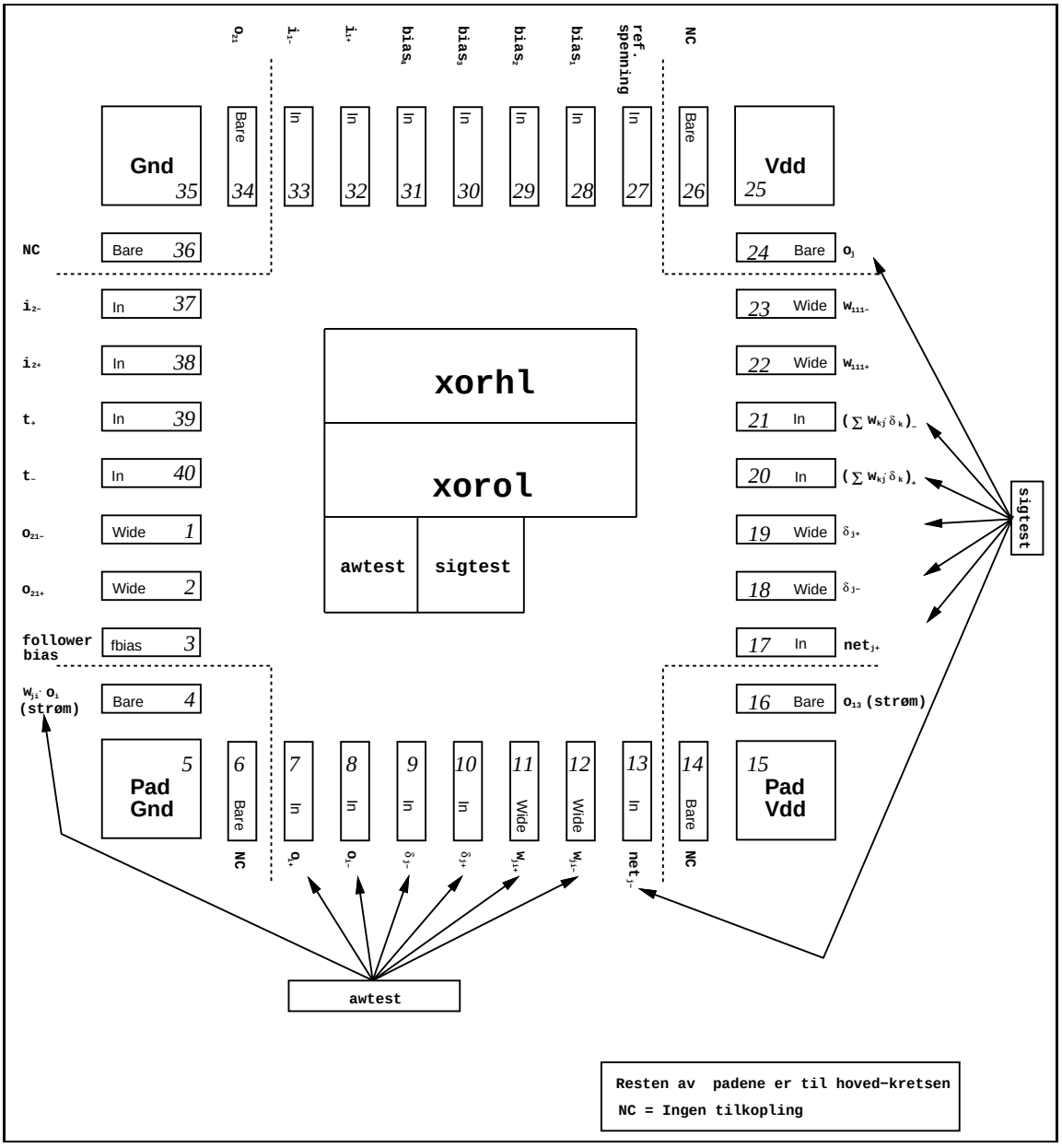
Jeg har brukt utleggseditoren *WOL* for å tegne utlegg i teknologien *scge*. Utlegget er prosessert i en $2\mu m$ n-brønn prosess med 2 poly-lag og 2 metall-lag. Metall2-laget har jeg brukt som dekklag og jord. "Pad"-rammen jeg har brukt har 40 padder, tinychip. Padrammen for krets1 er vist i figur B.1.

Både xor-nettet og de to enkle modulene har 4 forspenningssignaler og 1 ref. signal. Jeg koblet de nevnte signaler slik at de er felles for xor-nettet og de to enkle modulene. Ellers deler de 3 blokkene ingen padder.

Utleggsmessig har jeg i liten grad arbeidet med å minimalisere utlegget siden selve utlegget er nokså lite og siden målet i første omgang ikke innebærer å minimalisere utlegget. Alle cellene er ikke "selvstendige", dvs. de kan ikke opptre alene uten at design reglene brytes.

B.1.1 Transistorstørrelser

Jeg har gjort alle forspenningstransistorene smale og lange ($\frac{W}{L} = \frac{3}{10}\lambda$) for å redusere Early-effekten. Inngangstransistorene i kretselementene er generelt $\frac{W}{L} = \frac{6}{4}\lambda$. Andre transistorer spesielt i speil er gjort kvadratiske $\frac{W}{L} = \frac{4}{4}\lambda$. Alle transistorer som opptrer i par,



Figur B.1: Padramme for krets1.

dvs. differensielle inngangstransistorer er gjort like. Enkelte utgangstransistorer er gjort smale og lange slik som transistorene i forbindelse med strømutgangene som jeg tar ut på padder. Det gjelder ikke inngangstransistorene til hukommelseselementet som egentlig er utgangstransistorer fra en wide-range Gilbert multiplikator. Da jeg lagde utlegget for krets1, fryktet jeg en for stor forsterkning i inngangstrinnet i hukommelseselementet. Jeg gjorde derfor transistorene i inngangstrinnet kvadratiske. Utgangstransistorene i spenningsfølgeren i hukommelseselementet har jeg også gjort kvadratiske.

B.1.2 Ruting og busser

De 4 forspenningssignalene og det ene referansesignalet distribuerer jeg horisontalt øverst i hver av de større blokkene, mens datasignaler har jeg lagt i bunnen av hver blokk. Både den lokale rutingen og den globale rutingen til paddene er med 4λ bredde på lederne. All rutingen som hittil er nevnt er gjort i metall1. Noe av den globale ruting er gjort i metall2. Metall2 er ellers brukt som dekklag for å skjerme aktive områder og poly fra belysning av UV-lys. Dekklaget er forøvrig jordet. Det er bare "UV-hullene" som skal bestråles med UV-lys. Distribusjon av signaler mellom blokkene som ligger vertikalt kant i kant er i tillegg til metall1 også gjort i poly1.

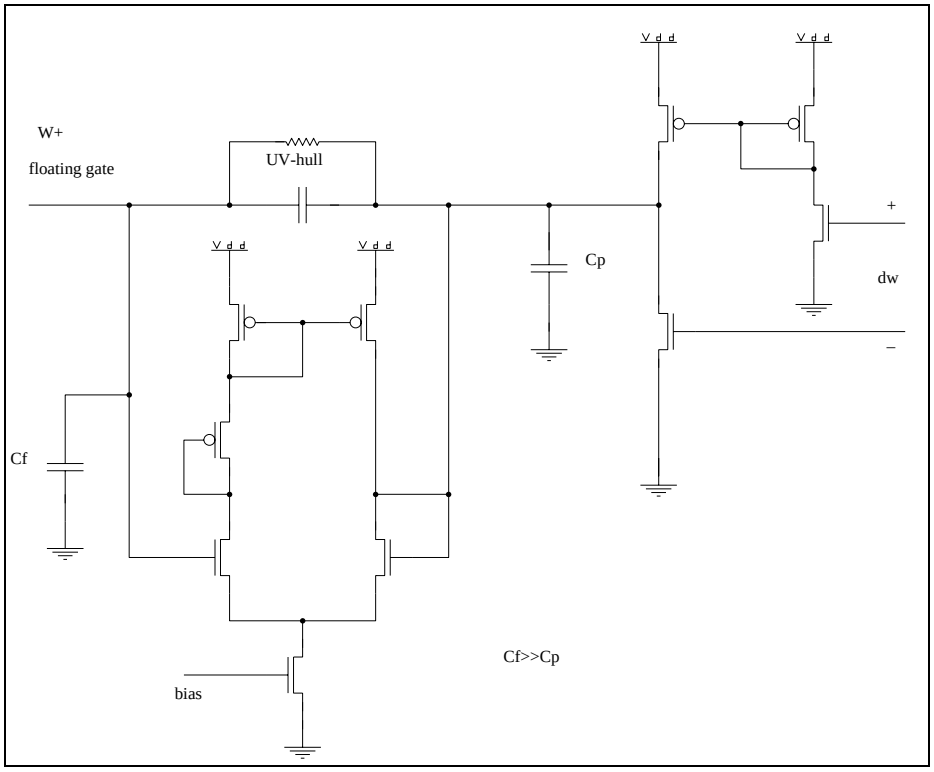
B.1.3 Det første hukommelseselementet

I det første kretsutlegget som jeg sendte til prosessering hadde hukommelseselementene en UV-struktur inspirert av [Lande]. Grunnidéen med hukommelseselementet som er vist i figur B.2 var at inngangstrinnet skulle trekke spenningen på programmeringssiden ned eller opp, mens spenningsfølgeren skulle motvirke endringer av den gamle verdien på floating gate siden.

Selve UV-hullene består som de vanlige kondensatorene av et poly1 og et poly2 lag som ligger over hverandre. Floating gate noden har jeg lagt ut i poly2, mens programmeringsnoden har jeg lagt ut i poly1.

Hvert hukommelseselement har to kondensatorer, en på hver side av UV-hullet. Kondensatoren C_p ligger på programmeringssiden av UV-hullet, mens kondensatoren C_f ligger på "floating gate"-siden. For at endringer på floating gate noden skal forsinkes noe i forhold til endringer på programmeringsnoden, bør vi ha $C_f \gg C_p$. Rent utleggsmessig har jeg gjort C_f bare litt større enn C_p. Gatekapasistansen til floating gate noden kommer i tillegg og er betydelig i forhold til C_f. Riktignok varierer gatekapasistansen med gatespenningen, men det burde være ikke være avgjørende for at en programmering av hukommelseselementet skal være mulig.

Etter å ha utført målinger på en enkel vekt, kan det tyde på at ekstra (unødvendig) last kapasistans på floating gate noden er så stor i forhold programmeringsnoden (kontrollnoden) at endringer på floating gate noden blir ubetydelige selv ved store endringer på programmeringsnoden. Jeg har bla. trukket floating gate noden rett ut på en utgangspad uten en spenningsfølger i mellom. Størrelsesordenen av den kapasitive divisjonen var nok en årsak til at hukommelseselementet ikke lot seg programmere, men neppe den eneste. Imidlertid har det ikke latt seg påvise andre grunnleggende feil for at en programmering



Figur B.2: Det første hukommelseselementet

av hukommelseselementet ikke lot seg gjøre. Dessuten har [Lande] lovende resultater fra sine hukommelseselementer.

B.1.4 Verifisering av utlegget

Jeg har gjort "design rule check" og ekstrahert hver enkelt celle og større blokker og til slutt hele utlegget (uten padder) for å eliminere design feil, kortslutninger, etc.

Jeg har også sammenlignet nettlistene fra utlegget fra *WOL* og kretsskjemaet fra *ANALOG* med *netcomp* for enkelt celler, større blokker og for hele utlegget for å være sikker på at jeg har lagt ut det jeg ønsket.

B.2 Utlegg for krets2

Jeg har gjort forholdsvis små endringer med utlegget fra krets1. Den viktigste forskjellen er den nye cellen for hukommelseselementet i tillegg til to tilhørende celler for lokal ruting til/fra hukommelseselementet. I tillegg har jeg på krets2 også tatt med enkle testceller for bumpkretsen, multiplikatoren og hukommelseselementet slik at de grunnleggende kretselementene kan testes ut hver for seg. Som i krets1 har jeg i tillegg til selve xor-nettet også i krets2 tatt med et enkelt nevron og en enkel vekt slik at disse byggeklossene kan testes ut

hver for seg. Padrammen for krets2 er vist i figur B.3.

B.2.1 Ny UV-struktur

Selve UV-strukturen i hukommelseselementet er slik som vist i figur 3.3. Floating gaten V_{fg} er i poly1. Over denne krysser de to nodene V_{cg} og V_{cap}, begge i poly2. Det er viktig at de to kondensatorene som de to nodene V_{cg} og V_{cap} lager ved å krysse V_{fg} er like store.

Den totale lastkapasistansen på noden V_{fg} vil nærmest elimineres ved hjelp av inverteren, som inverterer signalet fra noden V_{cg} slik at V_{cap} er den inverterte av V_{cg}, og den ekstra kondensatoren som oppstår mellom nodene V_{cap} og V_{fg}

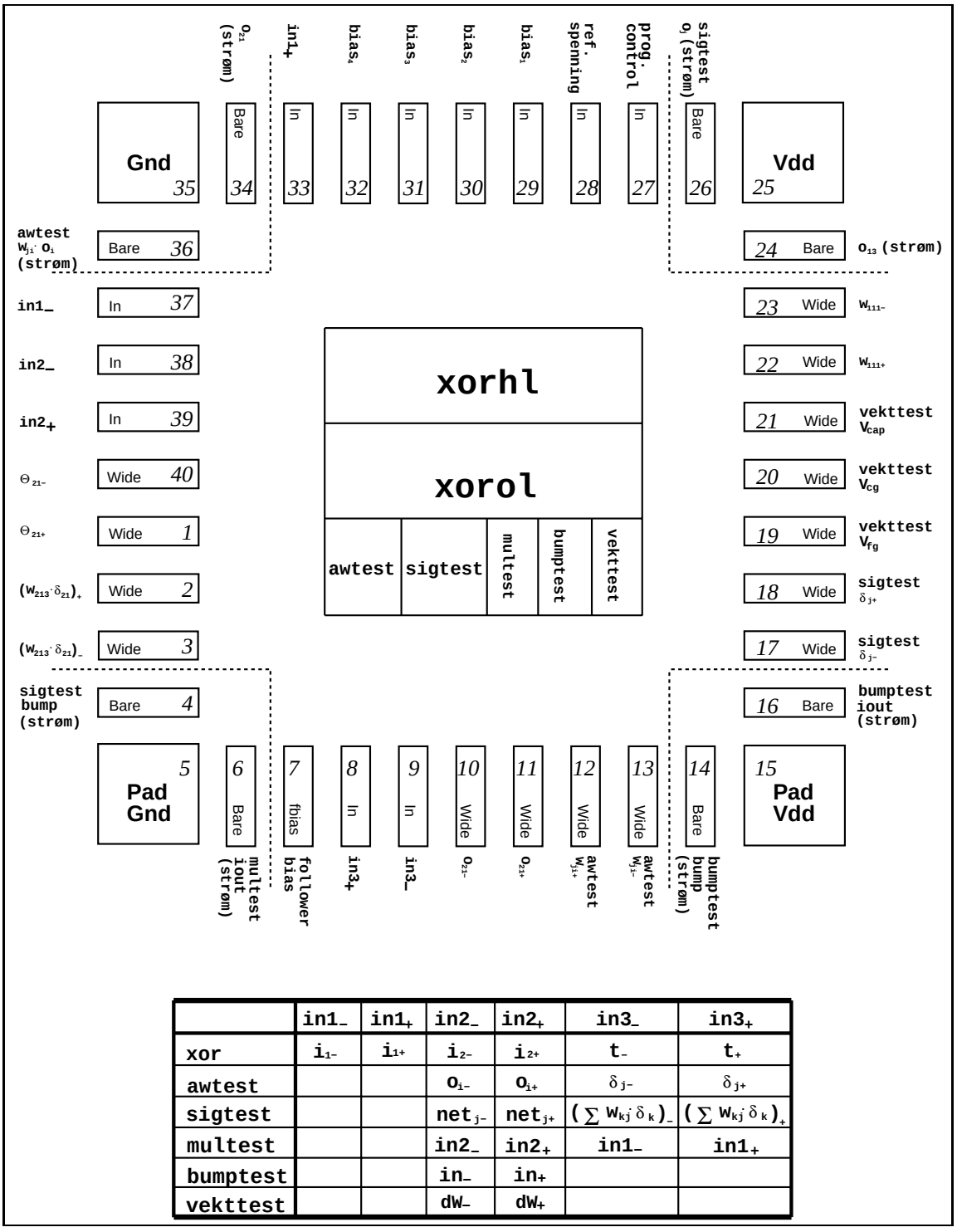
Det medfører en betydelig høyere programmeringshastighet. Alt aktivt område (på hele kretsen) dekkes med et beskyttende metall2 lag bortsett fra et lite vindu hvor noden V_{cg} som er i poly2 krysser noden V_{fg} som er i poly1. Dette er selve UV-hullet som belyses med UV-lys under programmeringen. For å redusere lasten på selve floating gate noden V_{fg}, har jeg gjort den liten og kompakt. Dessuten går signalet fra V_{fg} rett inn på inngangen til en spenningsfølger som distribuerer signalet videre. Selve UV-strukturen dekker utleggsmessig kun et areal lik $20X18\lambda$.

Det nye hukommelseselementet inneholder et nytt kontrollsignal for å kunne stabilisere kontrollnoden V_{cg} nær V_{DD} ved hjelp av en opptrekkstransistor (pull-up transistor) når programmering ikke pågår. Jeg har derfor utvidet bussen med forspennings- og kontrollsignaler fra krets1. Samtidig gjorde jeg bussene noe kraftigere.

B.2.2 Nye testkretser

Når det gjelder de nye enkle testelementene, har jeg for bumpkretsen tatt ut både det vanlige transkonduktansforsterkersignalet og bumpsignalet ut som strømsignaler. På multiplikatoren har jeg også tatt ut utgangssignalet som en strøm. Generelt har jeg for alle signaler som taes ut som et strømsignal, gjort utgangstransistorene en del kraftigere slik at jeg unngår problemer med at disse ikke driver tilfredstillende.

For det enkle hukommelseselementet har jeg tatt ut nodene V_{fg}, V_{cg} og V_{cap} slik at jeg kan betrakte de tre nodene for å bedre være i stand til å kunne oppklare feil. De tre nodene er alle tatt ut på pad via spenningsfølgere som ligger inne på selve kretsen for å ungå uønsket last på nodene. Cellen for det enkle hukommelseselementet er det samme som brukes ellers på kretsen. Hukommelseselementcellen har en ulempe når jeg skal måle på den. Ulempen er at inngangstrinnet til hukommelseselementet er slik at det skal veldig små endringer til på inngangssignalene før kontrollnoden V_{cg} endrer seg fra GND og helt opp til V_{DD} eller omvendt. Det ville ha vært en fordel å droppet dette inngangstrinnet på det enkle hukommelseselementet slik at programmeringsnoden ble bedre kontrollerbar. På den andre siden, ville denne endringen ha gitt meg to forskjellige celler, mens noe av poenget var å kontrollere at hukommelsescellene inne på kretsen ville virke før jeg begynte å måle på større moduler. Den enkle hukommelsescellen er derfor identisk med hukommelsescellene ellers på kretsen.



	in1_	in1+	in2_	in2+	in3_	in3+
xor	i_{1-}	i_{1+}	i_{2-}	i_{2+}	t_-	t_+
awtest			o_{i-}	o_{i+}	δ_{j-}	δ_{j+}
sigtest			net_{j-}	net_{j+}	$(\sum w_{kj}\delta_k)_-$	$(\sum w_{kj}\delta_k)_+$
multest			in2_	in2+	in1_	in1+
bumptest			in_	in+		
vekttest			dW_	dW+		

Figur B.3: Padramme krets2 med tabell over de forskjellige innsignalene.

B.2.3 Ruting

I motsetning til krets1, har jeg på krets2 distribuert et inngangssignal fra en innpad til flere kretselementer der hvor det har vært praktisk. Jeg har derfor fått flere tilgjengelige padder som jeg har benyttet til å ta ut flere kontrollnoder ut på pinner. Den globale rutingen har dermed blitt vesentlig mer kompleks.

Figur B.3 viser padrammen for krets2 med signalnavn og padtype. I tillegg inneholder figuren en tabell for å gi oversikt over inngangssignalene til de forskjellige kretsbitene.

Jeg har forøvring gjort vanlig design rule check for hele utlegget samt nettliste sammenlikning for hele utlegget bortsett fra paddene med nettliste fra kretsskjemaet fra *AnaLOG*.

Vedlegg C

Beskrivelse av utlegg

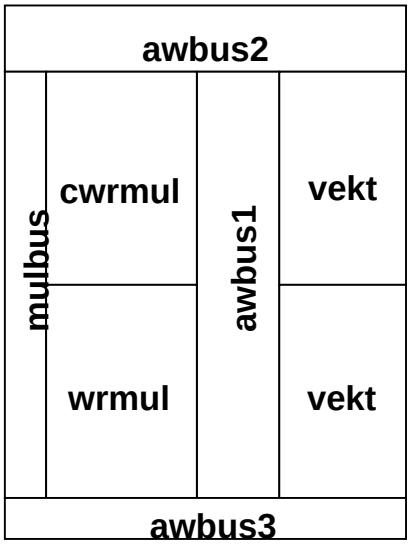
Utleggsmessig har jeg delt opp nettet i de samme blokkene slik som tidligere beskrevet ved den skjematiske beskrivelsen av nettet i figur 6.3. Et vilkårlig backprop nett består da av et antall vekter og et antall nevroner. Beskrivelsen gjelder for både krets1 og krets2. Krets2 har i tillegg testkretser for en enkel transkonduktansforsterker, en enkel multiplikator og et enkel hukommelseselement.

C.1 *Aweight* - vekt

Blokken *aweight* er en vekt slik som beskrevet i avsnitt 5.1 og består av to hukommelseselementer og to multiplikatorer. Blokkbeskrivelse av kretsutlegget for en vekt er vist i figur C.1. *Aweight* dekker utleggsmessig et areal lik $175X174\lambda$. De to multiplikatorene er litt forskjellige. Den multiplikatoren som beregner endringer av vekten har spenningsutganger. Kretsmessig er denne multiplikatoren slik som vist i figur 4.9. Cellen med denne multiplikatoren kaller jeg *wrmul*. Cellen som inneholder multiplikatoren som beregner det veide inngangssignalet kaller jeg *cwrmul*. Den har strømutganger og mangler derfor de to diodekoblingene $d1$ og $d2$ i figur 4.9 som konverterer signalet fra en strøm og til en spenning. Ellers er de to cellene identiske og like store. Multiplikatorcellen *cwrmul* har jeg speilet og plassert over wrmul slik at de får felles V_{DD}-buss.

De to hukommelseselementene er identiske. Cellen som inneholder et hukommelseselement har jeg kalt for *vekt* og er kretsmessig slik som vist i figur 4.11 for krets2. De to hukommelseselementene som inngår i en vekt har jeg plassert over hverandre på samme måte som for de to multiplikatorene.

I tillegg til de to hukommelseselementene og de to multiplikatorene, består en vekt av 4 celler med ruting/busser. Cellen *awbus2* ligger på toppen av en vekt og inneholder buss for forspennings- og referansesignaler i tillegg til det totale inngangssignalet net_j til tilhørende nevron. Cellen *mulbus* inneholder ruting av det felles inngangssignalet til de to multiplikatorene. Cellen *awbus3* inneholder bus med δ fra tilhørende nevron som er inngangssignal til multiplikatoren i cellen *wrmul* som beregner endringer av av en vekt. Cellen *awbus1* sørger for at de to hukommelseselementene og de to multiplikatorene får sine respektive forspenningssignaler som distribueres i cellen *awbus2*. Dessuten inneholder



Figur C.1: Aweight - floorplan for en vekt.

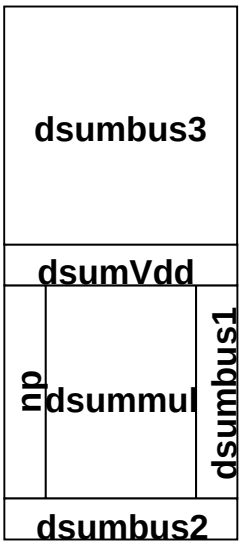
cellen ruting for kommunikasjon mellom multiplikatorene og hukommelseselementene.

C.2 *Deltasum* - ekstra multiplikator for beregning av δer i underliggende lag

Vi vet fra før at dersom nodene j er interne noder, så evaluerer vi δ_{pj} uttrykt ved hjelp av δer i overliggende lag slik som gitt i likning 2.5. Vektene som inneholder vekter w_{kj} har behov for å implementere multiplikasjonen $(\delta_{pk} w_{kj})$ fra likning 2.5. Utleggsmessig har jeg lagt multiplikasjonen innen en egen blokk *deltasum* slik som vist i figur C.2. En blokk med en vekt og en ekstra multiplikator har jeg kalt *aweightmd*. Den består av en *aweight* og en *deltasum*.

Deltasum inneholder en celle *dsummul*. Kretsmessig er *dsummul* en multiplikator slik som vist i figur 4.9. I motsetning til multiplikatoren i *wrmul* som får inngangssignalene V_{in1+} og V_{in1-} inn fra bunnen av cellen, så kommer inngangssignalene til *dsummul* inn fra venstre. Ellers er de to cellene like. V_{DD}-bussen med brønnkontakter i cellen *dsummul* er ikke komplett. *Dsum*V_{DD} sørger for å gjøre V_{DD}-bussen med brønnkontakter komplett.

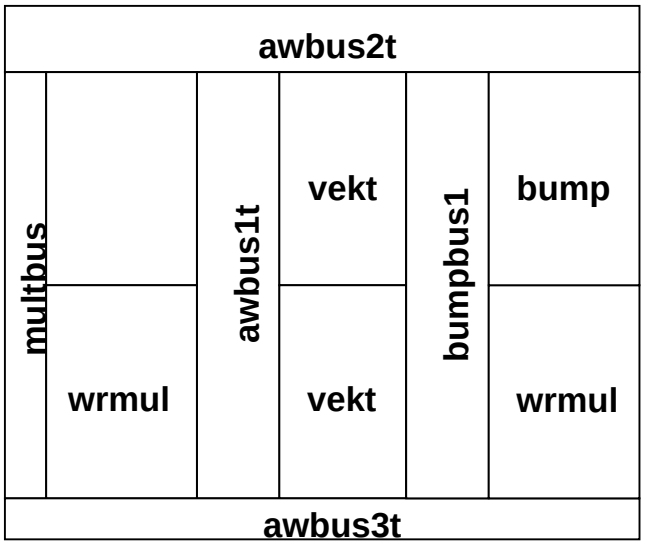
Cellen *dsumbus2* er tilsvarende *awbus3* og *awbus3t* og inneholder altså buss for δ-signalet. Dette δ-signalet benyttes bla. som inngangssignal til multiplikatoren i cellen *dsummul*. Signalet er i utgangspunktet med referanse til GND, mens cellen *dsummul* har behov for det med referanse til V_{DD}. Cellen *np* konverterer δ-signalet slik at det blir med referanse til V_{DD}. Det andre inngangssignalet til *dsummul* rutes fra vekten i tilhørende a*weight*-blokk. *Dsumbus* inneholder buss for forspennings- og referansesignaler tilsvarende *awbus2*. *Dsumbus1* inneholder ruting av utgangssignalet ut fra multiplikatoren i *dsummul*.



Figur C.2: Deltasum - floorplan for ekstra multiplikator.

C.3 *Sig* - nevron

Den blokken som jeg kaller *sig* er vist i figur C.3 og dekker utleggsmessig et areal lik $285X174\lambda$. Sig er et nevron slik som beskrevet i avsnitt 5.2. Den kan deles opp i to hoveddeler. Den ene delen har med terskelen Θ å gjøre. Denne halvdelen er svært lik en vekt bortsett fra at den kun har en multiplikator, nemlig den som beregner endringen av terskelen Θ (vekten). Utleggsmessig er halvdelen med terskelen derfor svært lik en vekt, men med et tomrom der hvor den 2. multiplikatoren skulle ha vært dersom det hadde vært en "vanlig" vekt. Jeg får da samme høyde utleggsmessig på *aweight* og *sig*-blokkene. Dvs. høyden på de 3 grunncellene *wrmul* (inkl. *cwrmul* og *wrmulmd*), *vekt* og *bump* er den samme. Den andre halvdelen av *sig*-blokken gir sigmoidsignalet og δ og består av en grunncelle av typen *bump* og en av typen *wrmul* i tillegg til en celle *bumpbus1*. Cellen *bump* inneholder bumpkretsen slik som vist i figur 4.3. Cellen *wrmul* er tidligere beskrevet og inneholder multiplikatoren i figur 4.9. Cellen *bumpbus1* inneholder diodekoblinger for å konvertere det totale inngangssignalet net_j til nevronet (*bump*-cellen) fra et strømsignal til et spenningssignal slik *bump*-cellen krever det. I tillegg sørger *bumpbus1* for å distribuere forspenningssignaler til bumpkretsen i cellen *bump* og multiplikatoren i cellen *wrmul*. Cellen *awbus2t* er en forlengelse av bussen i cellen *awbus2* i vekten. Høyden på cellen er derfor den samme som for *awbus2*. Tilsvarende for cellen *awbus3t*. *Awbus1t* inneholder ruting av forspenningssignal til multiplikator og vekt i tillegg til ruting mellom multiplikator og vekt for endring av disse. Cellen *multbus* inneholder to diodekoblinger av p-type i kaskade for å gi en passende spenningsdifferans inn på det andre settet innganger på multiplikatoren.



Figur C.3: Sig - floorplan for et nevron.

C.4 *Il* - inngangslaget

Xor-nettet består av to enkle nevroner i inngangslaget. Hvert nevron utgjøres av en celle *tkf* som er en vanlig transkonduktansforsterker, men med differensielle utganger slik som vist i figur 4.1.

I tillegg består il av 3 celler med busser/ruting for forspenningssignaler og inn- og utsignaler fra nevronene. Cellen *ilbus2* er en buss for forspennings- og referansesignaler og er tilsvarende cellene *awbus2* og *awbus2t* som er tidligere beskrevet. *Ilbus3* og *ilbus* består av ruting av henholdsvis inngangssignalene og utgangssignalene til de nevronene. Blokkskjema for kretsutlegget av inngangslaget er vist i figur C.4.

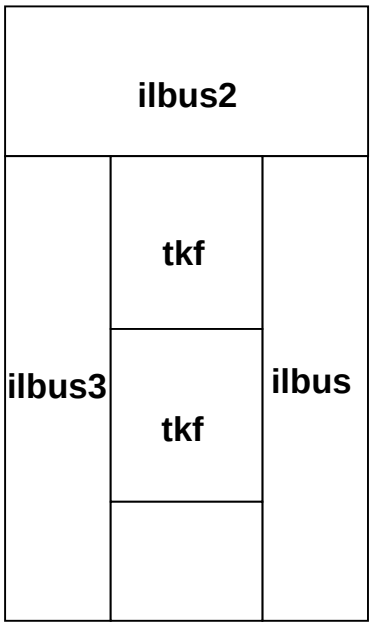
C.5 *Target* - sammenlikning av ønsket og oppnådd utgangssignal

Blokken *target* i figur C.5 gir utgangssignal fra nettet både i form av en differensiell spenning og i form av en strøm. Dessuten sammenliknes utgangssignalet fra nettet med ønsket utgangssignal (target).

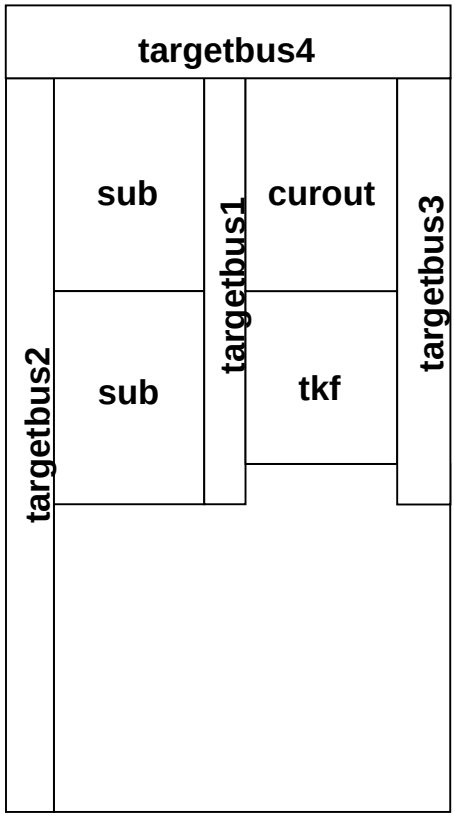
Ønsket utgangssignal skaleres i cellen *tkf* som inneholder en transkonduktansforsterker slik som vist i figur 4.1. Ønsket utgangssignal subtraheres fra utgangssignalet fra nettet ved hjelp av de to delkretsene i figur 4.22. Delkretsene i figur 4.22 implementeres med to like celler *sub*. En celle eller delkrets for hver av de to strømkomponentene.

Cellen *curout* konverterer utgangssignalet fra nettet fra et differensielt spenningssignal til et "vanlig" strømsignal.

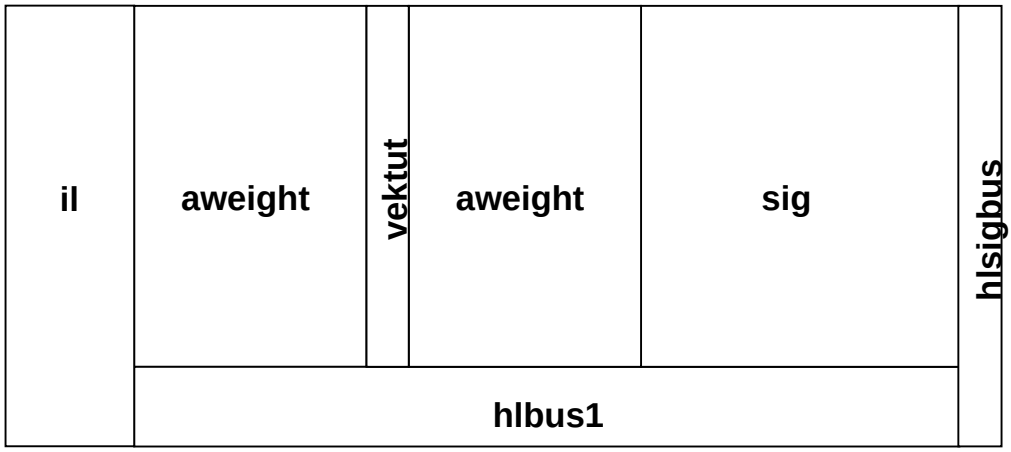
Target inneholder også 4 celler med busser/ruting. Cellen *targetbus4* er en fortsettelse av bussen for forspennings- og referansesignaler fra blokken *ol* som i tillegg inneholder ut-



Figur C.4: Il - floorplan for inngangslaget.



Figur C.5: Target - floorplan.



Figur C.6: Hl - floorplan for det skjulte laget.

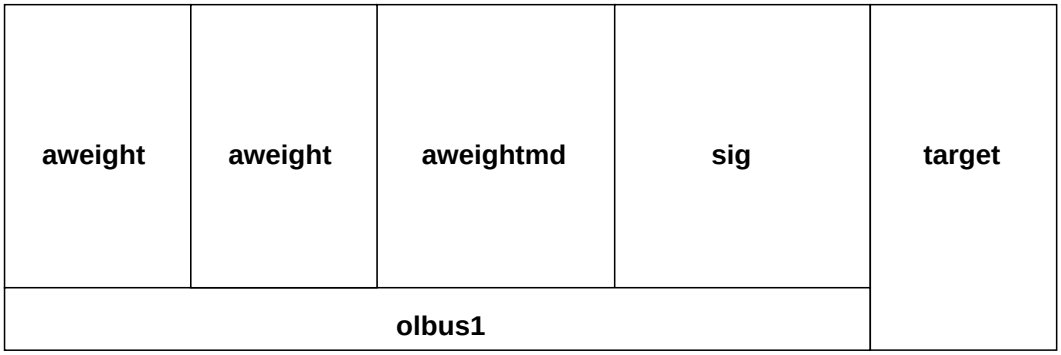
gangssignalet fra nettet i form av en spenning. Cellen *targetbus2* inneholder ruting mellom de to *sub*-cellene og *sig*-blokken. *Targetbus1* inneholder ruting av utgangssignalet fra *tkf* til inngangene på de to *sub*-cellene. *Targetbus3* inneholder ruting av forspenningssignaler og inngangssignal til transkonduktansforsterkeren i cellen *tkf* i tillegg til utgangssignal fra cellen *curout*.

C.6 *Hl* - det skjulte laget

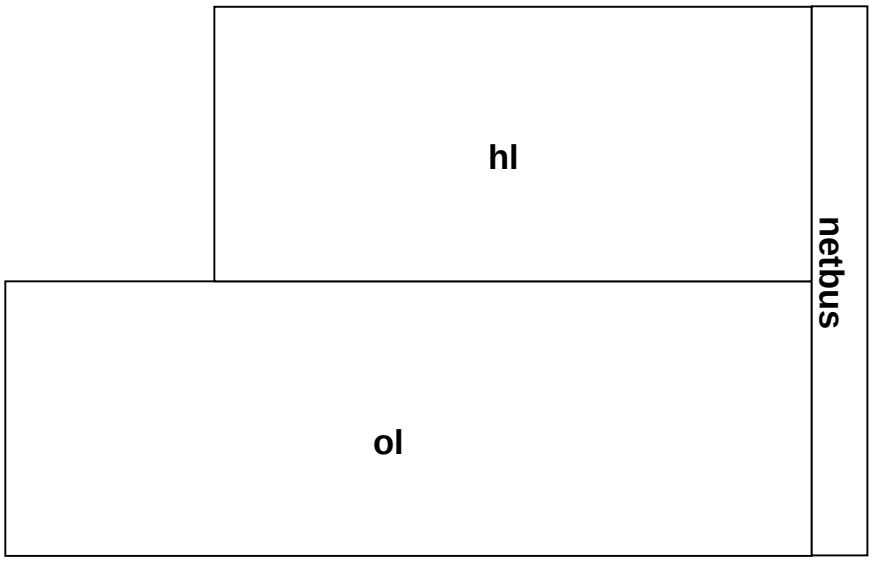
Blokken *hl* som utgjør inngangslaget og det skjulte laget i xor-nettet består i tillegg til blokkene *il*, *sig* og 2 stk. *aweight*er som er tidligere beskrevet, også av 3 grunnceller. *Hl* er vist i figur C.6. Cellen *hlbus1* inneholder buss med utgangssignalene fra inngangsnodene i *il* slik at disse signalene distribueres frem til de aktuelle vektene i blokkene *aweight*. Cellen *hlsigbus* sørger for kommunikasjon mellom bumpkretsen i cellen *bump* og multiplikatoren i cellen *dsummul* i *sig*-blokken. I tillegg distribuerer den utgangssignalet fra noden videre til neste lag. Cellen *vektut* gjør det mulig å kunne måle på en vekt. Den plasseres da kant i kant til høyre for den aktuelle *aweight*-blokken.

C.7 *Ol* - utgangslaget

Bare den ene av de 3 vektene i utgangslaget får sitt inngangssignal fra noden i det skjulte laget. De to andre vektene får sitt inngangssignal direkte fra inngangslaget. Det er derfor kun den ene av vektene som har behov for den ekstra blokken *deltasum* som bidrar til å beregne δ for underliggende interne node. Utgangslaget i xor-nettet utgjøres da av to *aweight*-blokker og en *aweightmd*-blokk i tillegg til selve nevronet, en *sig*-blokk. En grunncelle *olbus1* inneholder en buss med inngangssignalene til vektene. Til slutt i *ol* kommer en *target*-blokk siden det er snakk om utgangslaget. *Ol* er vist i figur C.7.



Figur C.7: Ol - floorplan for utgangslaget.

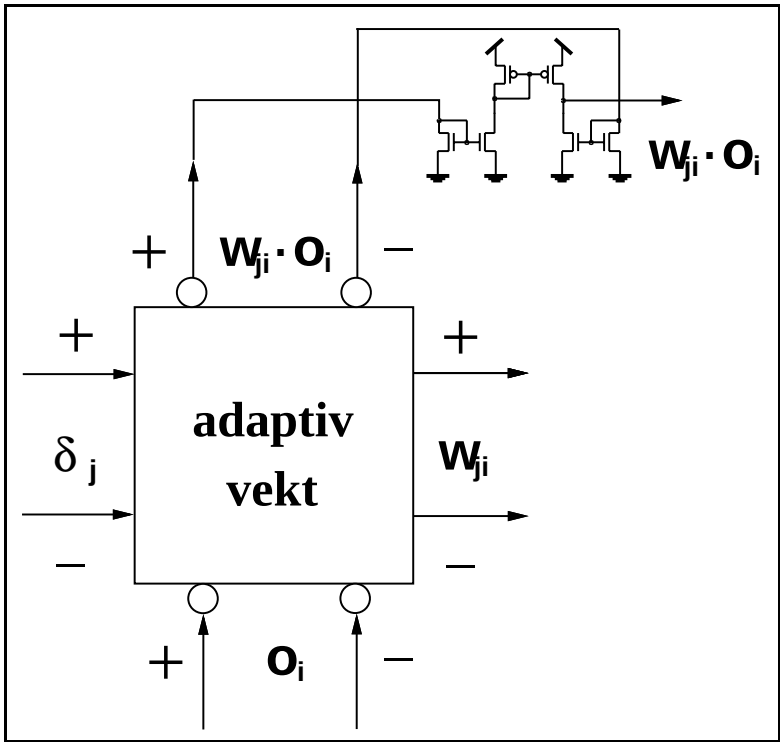


Figur C.8: Xor - floorplan for hele xor-nettet.

C.8 *Xor* - blokk som inneholder hele xor-nettet

Xor-nettet består av en *hl*-blokk og en *ol*-blokk slik som vist i figur C.8. *Ol* er lagt under *hl* og speilet siden padrammen ikke var stor nok til å legge disse to blokkene etter hverandre. Cellen *nettbus* sørger for sammenkoblingen av de to blokkene. I tillegg gir denne cellen mulighet til å ta ut utgangssignalet fra *hl* som en strøm.

Xor-nettet dekker utleggsmessig et areal lik $1204X637\lambda$. Det totale arealet innenfor en tinychip padramme er $1568X1598\lambda$. Store deler av arealet for xor-nettet benyttes til busser for globale signaler innen xor-nettet. I tillegg kommer ruting til padder. Det er i stor grad mulig å minimalisere utlegget.



Figur C.9: Blokkbeskrivelse for testutlegg av en enkel vekt - awtest.

C.9 *Awtest* - blokk for uttesting av en vekt

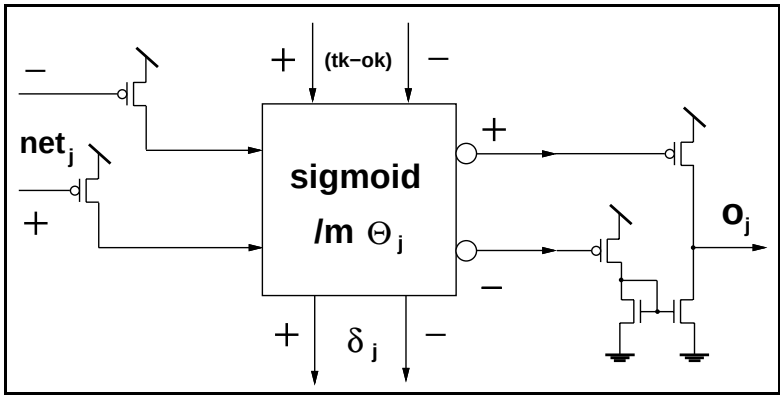
Den første av de to testblokkene kaller jeg *awtest*. Den består av en vekt hvor jeg tar ut det veide signalet som en strøm slik som vist i figur C.9.

Utleggsmessig består *awtest* av en *aweight*-blokk. I tillegg kommer 4 grunnceller. *Mulcurbus* er en forlengelse av bussen for forspenningssignalene og net_j-signalet. *Mulcurout* inneholder et strømspeil slik at de to komponentene for utgangssignalet fra vekten (bidraget til net_j) kan trekkes fra hverandre slik at utgangssignalet blir en enkel strøm. *Awoinn* sørger for tilkobling av inngangssignalet til vekten. *Awtestbus* sørger for at spenningsnivåene på de to hukommelseselementene i vekten kan taes ut på padder.

C.10 *Sigtest* - blokk for uttesting av et nevron

Den andre testblokken kaller jeg *sigtest*. Denne består av et nevron hvor jeg tar ut utgangssignalet som en strøm. Inngangssignalet net_j er i utgangspunktet et differensielt strømsignal. Jeg bruker heller et spenningssignal som inngangssignal til denne blokken slik som vist i figur C.10.

Utleggsmessig består *sigtest* av en *sig*-blokk. I tillegg kommer 4 grunnceller. *Sigtin* inneholder ruting av inngangssignalet til nevronet. Signalet er i form av en spenning. *Sigtestin* inneholder to transistorer. Inngangsspenningene påtrykkes gaten på de to transistorene slik at inngangsignalet til selve sigmoid-blokken blir i form av et strømsignal.



Figur C.10: Blokkbeskrivelse for testutlegg av et enkelt nevron - sigtest.

Se figur C.10. *Sigdsum* inneholder kontakter for tilkobling av det andre inngangssignalet $(t_k - o_k)$ til nevronet. *Sigtestout* gir utgangssignalet o_j ut i form av en strøm ved hjelp av speiling.

Referanser

[Andreou] A. G. Andreou, K. A. Boahen, P. O. Pouliquen, A. Pavasoviç, R. E. Jenkins, K. Strohbehn: *Current-Mode Subthreshold MOS Circuits for Analog VLSI Neural Systems.*
IEEE Transactions on Neural Networks, Vol. 2, No. 2, 1991

[Berg] Yngvar Berg: *A simulation and modelling framework for analog CMOS implementations of neural systems.*
Research Report Department of Informatics, University of Oslo, 1992.

[Borgstrom] T. H. Borgstrom, M. Ismail, S. B. Bibyk: *Programmable current-mode neural network for implementation in analouge MOS VLSI.*
IEEE Proceedings, Vol. 137, Pt. G, No. 2, 1990

[Burr] J. B. Burr: *Digital Neural Network Implementations.*
Standford University 1991.

[Delbrück] Tobi Delbrück: *"Bump" Circuits for Computing Similarity and Dissimilarity of Analog Voltages.*
Caltech 1991.

[Holler] M. Holler, S. Tam, H. Castro, R. Benson: *An Electrically Trainable Artificial Neural Network (ETANN) with 10240 "Floating Gate" Synapses.*

[Lande] T. S. Lande, M Sivilotti: *The Adaptive D/A-converter for Subthreshold operation.*
Ifi 1991.

[Maher] M. A. Maher:
New UV-Memory Writing Scheme (Unpublished).
1992.

[Mead] Carver Mead: *Analog VLSI and Neural Systems.*
Addison Wesley 1989.

[Pao] Yoh-Han Pao: *Adaptive Pattern Recognition.*
Addison-Wesley 1989.

[Rumelhart86] D. E. Rumelhart, J. L. McClelland: *Parallell Distributed Processing – Explorations in the Microstructure of Cognition. Vol.1 Foundations.*
MIT Press 1986.

[Streetman] Ben G. Streetman: *Solid State Electronic Devices.*
Prentice-Hall 1990.

[Watts] L. Watts, D. A. Kerns, R. F. Lyon, C. Mead: *Improved Implementation of the Silicon Cochlea.*
IEEE Journal of Solide-state Circuits, Vol. 27, No. 5, 1992.