

# Ontwerp van een RRAM geheugen

Wouter Diels  
Alexander Standaert

Thesis voorge dragen tot het behalen  
van de graad van Master of Science  
in de ingenieurswetenschappen:  
elektrotechniek, optie Elektronica en  
geïntegreerde schakelingen

**Promotor:**

Prof. dr. ir. W. Dehaene

**Assessoren:**

Prof. dr. ir. R. Lauwereins  
Prof. dr. ir. M. Verhelst

**Begeleiders:**

ir. B. Baran  
dr. ir. S. Cosemans

© Copyright KU Leuven

Zonder voorafgaande schriftelijke toestemming van zowel de promotor als de auteurs is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen tot of informatie i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, wend u tot ESAT, Kasteelpark Arenberg 10 postbus 2440, B-3001 Heverlee, +32-16-321130 of via e-mail [info@esat.kuleuven.be](mailto:info@esat.kuleuven.be).

Voorafgaande schriftelijke toestemming van de promotor is eveneens vereist voor het aanwenden van de in deze masterproef beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

# Voorwoord

Dit is mijn dankwoord om iedereen te danken die mij bezig gehouden heeft. Hierbij dank ik mijn promotor, mijn begeleider en de voltallige jury. Ook mijn familie heeft mij erg gesteund natuurlijk.

*Wouter Diels  
Alexander Standaert*

# Inhoudsopgave

<b>Voorwoord</b>	i
<b>Samenvatting</b>	iv
<b>Lijst van figuren en tabellen</b>	v
<b>Lijst van afkortingen en symbolen</b>	vii
<b>1 Inleiding</b>	1
1.1 Doel en afbakening van dit werk . . . . .	1
1.2 Structuur van de tekst . . . . .	2
<b>2 Geheugencel</b>	3
2.1 Memristor . . . . .	3
2.2 Memristortoepassingen . . . . .	6
2.3 Besluit . . . . .	6
<b>3 Geheugenarchitectuur</b>	7
3.1 Cel . . . . .	7
3.2 Branch . . . . .	7
3.3 Local Block . . . . .	8
3.4 Global Block . . . . .	9
3.5 Besluit . . . . .	10
<b>4 Lastimpedantie-analyse</b>	11
4.1 algemene last eigenschappen en specificaties . . . . .	11
4.2 evalueren van de last . . . . .	12
4.3 vergelijking van verschillende types last . . . . .	14
4.4 Besluit . . . . .	20
<b>5 Sense Amplifier analyse</b>	23
5.1 Types SA . . . . .	23
5.2 Offsetspanning . . . . .	23
5.3 Sensitiviteitsanalyse . . . . .	24
5.4 Paretosimulatie . . . . .	29
5.5 Besluit . . . . .	30
<b>6 Omringende logica</b>	31
6.1 Decoders . . . . .	31
6.2 Buffers . . . . .	34

## INHOUDSOPGAVE

---

6.3	BL- en WL-drivers . . . . .	36
6.4	Passgates . . . . .	36
6.5	Besluit . . . . .	38
<b>7</b>	<b>Timing en optimalisatie</b>	<b>41</b>
7.1	Timing . . . . .	41
7.2	Analyse verschillende geheugenconfiguraties . . . . .	47
7.3	Besluit . . . . .	49
<b>8</b>	<b>Finaal ontwerp</b>	<b>51</b>
8.1	Het finaal ontwerp . . . . .	51
8.2	Besluit . . . . .	53
<b>9</b>	<b>Besluit</b>	<b>55</b>
<b>A</b>	<b>Charge injectie met ideale spice bronnen.</b>	<b>59</b>
	<b>Bibliografie</b>	<b>61</b>

# Samenvatting

In dit **abstract** environment wordt een al dan niet uitgebreide samenvatting van het werk gegeven. De bedoeling is wel dat dit tot 1 bladzijde beperkt blijft.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

# Lijst van figuren en tabellen

## Lijst van figuren

2.1	Abrupte overgang van hoge weerstand naar lage weerstand voor NiO[1]	4
2.2	Metal-Insulator-Metal structuur[18]	4
2.3	Model van het Pt-TiO <sub>2</sub> -Pt staal[16]	5
2.4	Illustratie van forming,resetting en setting[18]	5
2.5	Een 1T1R-configuratie[18]	6
3.1	Een geheugencel en een branch	8
3.2	Een Local Block	8
3.3	Topologie om referentiesignaal te verkrijgen	9
3.4	Een Global Block	10
4.2	Test bench voor de last	13
4.3	De verschillende types last	14
4.4	Lineaire sweep van switchload	17
4.5	Lineaire sweep van biasload	17
4.6	Lineaire sweep van diodeload	18
4.7	Lineaire sweep van bulkload	18
4.8	Bitlijn voltage distributie voor een biasload	19
4.9	Lineaire sweep van switchload	19
4.10	Verschillende oplossingen voor de switchload met variabele lengtes en breetes	20
4.11	Bitlijn voltage distributie voor de finale load	21
5.1	een sense amplifier	23
5.2	Illustratie van offsetspanning	24
5.3	Door $\beta$ -mismatch is ladingsinjectie van de pass-gates niet meer gematched en gaat de SA foutief latchen	26
5.4	Simulatieopstelling voor het RC-latch-effect	26
5.5	Simulatieresultaten voor het RC-latch-effect:de 2 ingangs-uitgangsknopen zijn voorgeladen op 400mV en 380mV. Na 1,6ns wordt de SA aangezet. De SA is ideaal voor deze simulatie.	27
5.6	Vergelijking situatie met voorgeladen (eindige) capaciteit en situatie met spanningsbron (oneindige capaciteit)	27

---

## LIJST VAN FIGUREN EN TABELLEN

---

5.7	Circuit voor analyse voorwaarden RC-latch-effect . . . . .	28
5.8	De pareto-optimale sense amplifiers . . . . .	30
6.1	Opbouw voor grotere decoders . . . . .	31
6.2	basis decoders . . . . .	32
6.3	vergelijking van decoder types . . . . .	34
6.4	Energie verbruik in griddecoder . . . . .	35
6.5	Glitch in NOR-gate . . . . .	35
6.6	Gebufferde en ongebufferde signalen naar de referentie logica . . . . .	36
6.7	nMOS passgate opstelling. a: Vs > Vx, b: Vs < Vx . . . . .	37
6.8	pMOS passgate opstelling. a: Vs > Vx, b: Vs < Vx . . . . .	37
6.9	Dode zones voor verschillende types passgates . . . . .	39
7.1	Timing problemen bij de bitlijn . . . . .	42
7.2	Globalblock logica . . . . .	43
7.3	Timing globalblock . . . . .	43
7.4	Controle logica memory array . . . . .	44
7.5	Timing controle logica memory array . . . . .	44
7.6	Delay van woordlijn decoders + buffers ifv bitlijn decoders . . . . .	45
7.7	Controle logica memory array . . . . .	45
7.8	Timing controle logica memory array . . . . .	46
7.9	logica rond SA . . . . .	46
7.10	Timing logica rond SA . . . . .	47
7.11	Lees cyclus . . . . .	48
7.12	Invloed #BL en #WL op delay, energie verbruik en oppervlakte . . . . .	49
7.13	Delay, energie verbruik en oppervlakte van alle geheugenconfiguraties . . . . .	50
8.1	Resultaten speed-vdd test . . . . .	52
8.2	Bl spanningen ifv Vdd . . . . .	53
A.1	Testcircuit ladingsinjectie . . . . .	60
A.2	Stromen in circuit . . . . .	60

## Lijst van tabellen

5.1	Sensitiviteitsanalyse van de minimale SA . . . . .	25
6.1	De verschillende aantal gates in de grid decoder . . . . .	33
6.2	Lasten in de verschillende buffers . . . . .	36
8.1	Grotes van de transistoren in het finaal ontwerp . . . . .	52

# Lijst van afkortingen en symbolen

## Afkortingen

BL	Bit Line
CDF	Cumulative Distribution Function
GB	Global Block
LB	Local Block
NoBLpLB	Number of Bit Lines per Local Block
NoGB	Number of Global Blocks
NoWLpB	Number of Word Lines per Branch
PDF	Probability Distribution Function
RAM	Random Access Memory
RRAM	Resistive Random Access Memory
SA	Sense Amplifier
SL	Source Line
WL	Word Line
MTJ	Magnetic Tunnel Junction
HRS	High Resistive State
LRS	Low Resistive State



# **Hoofdstuk 1**

## **Inleiding**

Vandaag de dag is elektronica niet meer uit het leven weg te denken. Van de smartphone tot het digitaal horloge, van de bordcomputer in de moderne wagen tot de microprocessor in de vaatwasser, overal vind je wel elektronica terug. Sinds Gordon Moore ongeveer 50 jaar geleden de uitspraak deed dat het aantal transistoren op eenzelfde oppervlakte per twee jaar zou verdubbelen [10], is de industrie er over het algemeen goed in geslaagd dit te verwezenlijken. Dit leidde tot de snelle en uiterst complexe chips die we vandaag allemaal goedkoop aankopen.

Naarmate de processorkracht groter werd, steeg ook de vraag voor grotere en snellere geheugens om deze processorkracht ook effectief uit te buiten. Static Random Access Memory (SRAM) blijft een populaire keuze voor snelle ingebedde geheugens, maar heeft het nadeel vlugtig te zijn: eenmaal de voedingsspanning wordt afgeschakeld, verdwijnt de informatie. Flash-geheugens, door veel mensen gebruikt voor massa-opslag in USB-sticks of SSDs, hebben ook hun weg gevonden tot het ingebedde domein en behoren wel tot de klasse van niet-vluchige geheugens. Het blijkt echter bijzonder moeilijk om flash-geheugens verder te verkleinen [11].

Onderzoek naar nieuwe geheugens is dan ook onontbeerlijk. Zo zijn er al nieuwe nieuwe kandidaten in opmars die hoopgevende tekens geven om te concurreren met (ingebedde) flash-geheugens. MRAMs (Magnetic RAMs) en in het bijzonder STT-RAM (Spin-Transfer Torque) zullen op termijn een belangrijke rol gaan spelen.

Een andere kandidaat is Resistive RAM (RRAM of ReRAM). Daar waar SRAM-en flash-cellen de informatie bevatten via het al dan niet aanwezig zijn van lading, bevat een RRAM-cel informatie door een bepaalde elektrische weerstand aan te nemen. RRAM zou geen problemen hebben om nog even op de klassieke manier mee te schalen en is dus zonder meer een interessante piste om te onderzoeken. Bovendien zou het gefabriceerd kunnen worden met goedkopere processen dan flash-geheugens - bij flash-geheugenfabricatie zijn vaak dure extra maskers vereist.

### **1.1 Doel en afbakening van dit werk**

Dit werk beschrijft het ontwerp van een 4MByte RRAM-geheugen voor ingebedde toepassingen. De doelstelling is een pareto-optimaal (dynamische energie-snelheid-

## 1. INLEIDING

---

oppervlakte) werkend circuit te ontwerpen, gewapend tegen variabiliteit - ongecorreleerde gedragsvariaties van componenten. Het ontwerp houdt rekening met data-retentie bij het uitlezen van bits. De analyse focust op de leesbewerking, de schrijfbewerking valt buiten het bereik van dit werk. [Er worden wel mogelijke oplossingen aangereikt, maar deze werden niet uitdrukkelijk onderzocht]

Voor de leesbewerking wordt het geheugen-element gemodelleerd als een weerstand waarvan de nominale weerstandswaarde afhangt van de celtoestand. Wanneer variabiliteit wordt onderzocht, zal deze weerstandswaarde een stochastische variabele worden met een normale verdeling.

Temperatuursvariaties werden niet in rekening genomen, maar aangezien dit een globale variabele is en het systeem differentieel werkt, wordt niet verwacht dat de performantie aanzienlijk zal verminderen.

Alle data die worden getoond, komen voort uit Spectre-simulaties met 45nm PTM transistormodellen.

## 1.2 Structuur van de tekst

In hoofdstuk 2 zal de technologie van een RRAM geheugen uiteengezet worden, alsook diens toepassingen. Ook zal het elementaire principe om uit een weerstand een nuttig elektrisch signaal te vormen uitgelegd worden. In hoofdstuk 3 wordt het geheugensysteem vanuit vogelperspectief besproken. Er wordt hier ook aangehaald wat de tunebare parameters zijn van de architectuur. Voor een robuuste, snelle en laag-energetische leesoperatie uit te voeren is het belangrijk het geheugenelement te combineren met een zorgvuldig gekozen impedantie, dit wordt onderzocht in hoofdstuk 4. Uiteindelijk zal er een bitstream moeten gegenereerd worden aan de uitgang van het systeem, de sense amplifier zorgt hiervoor en wordt besproken in hoofdstuk 5. In de geheugenstructuur zijn ook bepaalde logische (digitale) operaties nodig om op basis van het opgegeven adres de juiste cel aan te spreken, de hiervoor gebruikte blokken worden beschreven en geanalyseerd in hoofdstuk 6. Ten slotte zal in hoofdstuk 7 de timing van controlesignalen onderzocht worden en hoe het systeem te optimalizeren door middel van de architectuurparameters te tunen.

# Hoofdstuk 2

## Geheugencel

Elk geheugen bestaat uit een verzameling individuele cellen die op een of andere manier informatie bevatten. In dit hoofdstuk wordt eerst dieper ingegaan op de manier waarom een R-RAM geheugencel informatie bevat en vervolgens hoe deze informatie [elektrisch] kan worden uitgelezen.

### 2.1 Memristor

Het essentiële element van een R-RAM geheugencel is ontegensprekbaar de zogenaamde memristor. De memristor wordt ook wel gezien als de 4<sup>e</sup> passieve component, naast de weerstand, spoel en condensator.

#### 2.1.1 Theoretisch principe

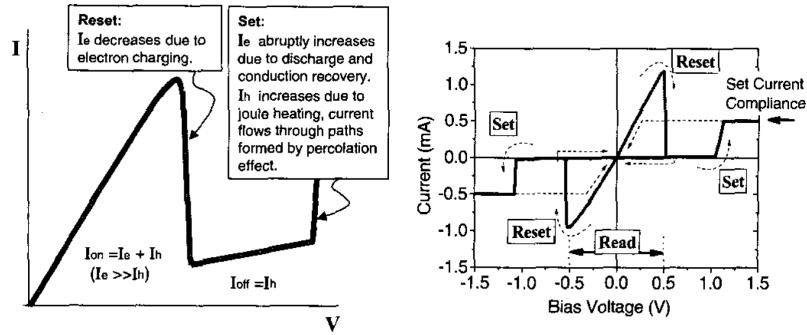
In 1971 publiceerde Leon Chua, een onderzoeker in o.a. niet-lineaire circuittheorie, een artikel waarin hij opmerkte dat er voor de 4 fundamentele circuitvariabelen (de spanning  $v$ , stroom  $i$ , lading  $q$  en flux  $\phi^1$ ) van 6 mogelijke onderlinge relaties er slechts 5 gekend waren:  $q(t) = \int_{-\infty}^t i(\tau) d\tau$ ,  $\phi(t) = \int_{-\infty}^t v(\tau) d\tau$ ,  $v(t) = R * i(t)$ ,  $q(t) = C * v(t)$  en  $\phi(t) = L * i(t)$  volgen uit de wetten van Maxwell en uit de definities van de weerstand, spoel en condensator, maar er ontbrak een relatie tussen  $\phi$  en  $q$ [3]. Hij suggereerde dat er een 4e nog niet ontdekte passieve 2-pool moest bestaan die dit verband herbergde. Hij stelde dat  $M(q) = \frac{d\phi(q)}{dq}$  met  $M$  de *memristance*. Hieruit volgt dat voor dit element  $v(t) = M(q(t))i(t)$ . Indien er een lineair verband bestaat tussen  $\phi$  en  $q$ , gedraagt dit element zich als een gewone weerstand. Enkel wanneer er een niet-lineair verband bestaat, beginnen er zich interessante karakteristieken voor te doen. Zo gedraagt het element zich ogenblikkelijk als een weerstand, maar gaat deze weerstandswaarde variëren in de tijd aan de hand van de stroom die er doorgelopen heeft. Gebaseerd op deze conclusie doopte hij deze component de memristor (een contractie van memory en resistor). Chua beëindigde zijn artikel met te erkennen dat er op dat moment nog geen fysische memristor was ontdekt, maar dat dit in de toekomst wel kon gebeuren, al dan niet zelfs per ongeluk. Hij gaf zelfs aan dat er

---

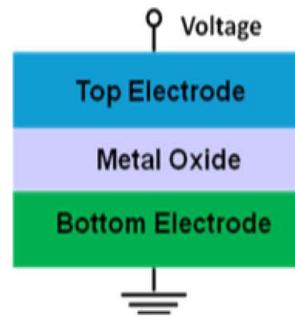
<sup>1</sup> $\phi(t) = \int_{-\infty}^t v(\tau) d\tau$ , voor een ideale inductantie is dit hetzelfde als magnetische flux

## 2. GEHEUGENCSEL

---



Figuur 2.1: Abrupte overgang van hoge weerstand naar lage weerstand voor NiO[1]



Figuur 2.2: Metal-Insulator-Metal structuur[18]

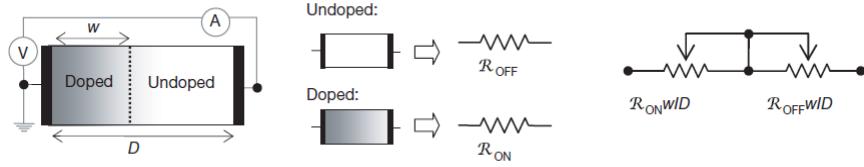
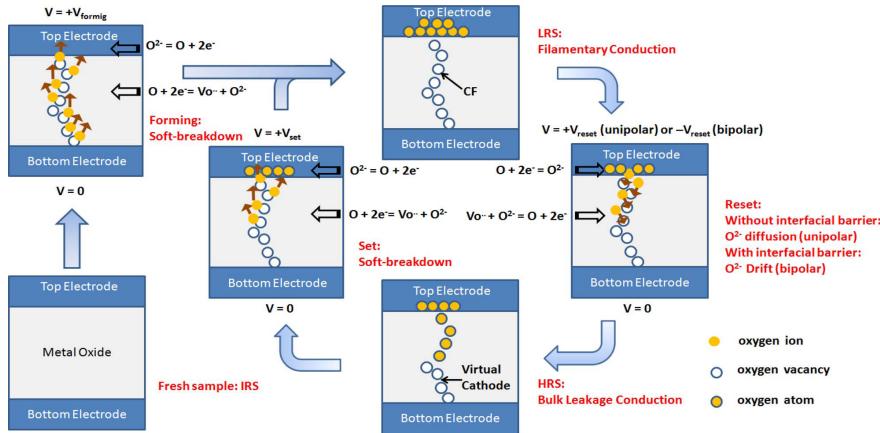
misschien al in die tijd materialen met memristorkarakteristieken gebruikt werden, maar dat men hier over keek. Hij zou gelijk krijgen.

### 2.1.2 Fysische memristors

Er werd reeds langer (zelfs al sinds de jaren 60) opgemerkt dat sommige metaaloxides, die normaal gezien als elektrisch isolator functioneren, een plotse overgang kunnen ervaren naar een veel geleidender staat (zie figuur 2.1). Dit gebeurt veelal in een configuratie waarbij het oxide wordt geplaatst tussen 2 metalen (MIM configuratie).[18] (zie ook figuur 2.2)

In 2008 publiceerde een onderzoeksgrond van Hewlett-Packard een artikel waarin ze opmerkten dat het gedrag van hun Pt-TiO<sub>2</sub>-Pt stalen een merkwaardige gelijkenis vertoonde met Chua's memristor.[16] Uit de modellering van hun stalen argumenteerden ze dat dit een ideale memristor zou zijn en dat het effect meer uitgesproken is bij kleine afmetingen: het titaniumoxide bestaat uit 2 delen: zuiver TiO<sub>2</sub>, een halfgeleider met hoge weerstand, en TiO<sub>2-x</sub> met zuurstofafwezigheid (oxygen vacancies) met een veel lagere weerstand. Door een elektrisch veld aan te leggen worden zuurstofatomen weg of in het rooster getrokken en verandert de verhouding TiO<sub>2</sub> en TiO<sub>2-x</sub> en dus ook de netto weerstand (zie figuur 2.3).

Voor deze opstelling geldt dan dat  $M(q) = R_{off}(1 - \frac{\mu_v R_{on}}{D^2} q(t))$  met D de dikte

Figuur 2.3: Model van het Pt-TiO<sub>2</sub>-Pt staal[16]

Figuur 2.4: Illustratie van forming,resetting en setting[18]

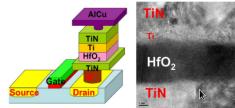
van de titaniumoxidefilm,  $\mu_v$  de mobiliteit van de zuurstofionen en  $R_{on} \leq M \leq R_{off}$ . Het dynamisch gedrag van de ogenblikkelijke weerstandswaarde is dus afhankelijk van het verloop van de stroom in de tijd en dit des te meer in het nanometerdomein.

Naast titaniumoxide zijn er nog andere materialen die schakelend weerstandsverdrag vertonen zoals nikkeloxide[1], hafniumoxide[2], aluminiumoxide[6],... Niet altijd kunnen de resultaten gemodelleerd worden volgens de originele memristortheorie, maar desalnietemin zullen deze materiaalconfiguraties bruikbaar zijn in toepassingen. Bij al deze MIM-configuraties blijft het mechanisme wel hetzelfde: na fabricatie is het oxidekristal intrinsiek zuiver, maar onder druk van een voldoende groot elektrisch veld zullen de zuurstofatomen losgerukt worden uit het rooster naar de anode. Het gebrek aan zuurstofatomen zorgt voor conductieve filamenten. Het element bereikt dan een laagresistieve staat (LRS). Deze zachte doorslag van het zuivere oxide wordt *forming* genoemd. Het proces is tot zekere hoogte omkeerbaar (*reset*), maar er zullen altijd meer defecten in het kristal zijn dan na fabricatie. Dit betekent dus ook dat nadat de memristor één keer een forming- en resetproces is ondergaan en zich terug in een hoogresistieve staat (HRS) bevindt, er hierna een minder groot elektrisch veld nodig is om terug tot een LRS te komen. Dit proces heet *setting*. Deze drie processen zijn geïllustreerd op figuur 2.5.

De verschillende MIM-structuren hebben ook verschillende eigenschappen. Zo moet er onderscheid gemaakt worden tussen unipolaire schakelen en bipolaire. Bij

## 2. GEHEUGENCSEL

---



Figuur 2.5: Een 1T1R-configuratie[18]

bipolair resistief schakelen zal forming/setting optreden wanneer de aangelegde spanning een bepaalde polariteit heeft en resetting bij de omgekeerde polariteit. Bij unipolaire schakelen is de amplitude van de spanning doorslaggevend voor welke van de 3 processen zal optreden, niet de polariteit.

## 2.2 Memristortoepassingen

Voor dit werk is het voor de hand liggend dat de memristor gebruikt kan worden als geheugenelement: de MIM-configuraties hebben op z'n minst 2 resistieve toestanden, al zijn er artikels gepubliceerd waarbij de memristor nog meer resistieve toestanden bevat[9]. Met deze MRS [multiresistive states] kan een nog hogere densiteit aan informatie gerealiseerd worden aangezien elke memristor meer dan 1 bit informatie zou bevatten. Deze 2 toestanden kunnen gebruikt worden voor geheugen- en logicatoepassingen[14][12]. In geheugentoepassingen kan men onderscheid maken tussen 1T1R-, 1R- en 1D1R-configuraties[5]. Met een 1R-configuratie kan men de grootste densiteit van geheugen bereiken, alsook een betere schaling, maar deze configuratie heeft te kampen met cellen die maar half geselecteerd worden. Dit kan opgelost worden door een selectie-element aan de configuratie toe te voegen, zoals een diode of een transistor. De 1D1R configuratie kan echter enkel geïmplementeerd worden met unipolaire memristors[19]. Daarom dat er in dit werk voor een 1T1R-configuratie geopteerd werd, de gebruikte memristor is immers een bipolaire hafniumoxide-memristor (zie figuur ??).

Naast geheugentoepassingen heeft de memristor ook potentieel in logicatoepassingen, er wordt zelfs gesproken over een volledige vervanger van de transistor[7].

## 2.3 Besluit

De memristor is een theoretische passieve component die lading en flux met elkaar verbindt. In de praktijk zijn er MIM-configuraties ontdekt die memristorkarakteristieken vertonen. Deze karakteristieken zijn bijzonder interessant voor geheugens, wat in de volgende hoofdstukken wordt toegepast.

# Hoofdstuk 3

## Geheugenarchitectuur

De afzonderlijke geheugencellen zullen samengebracht worden in een geheel. Dit hoofdstuk bespreekt de algemene structuur alsook de vrijheidsgraden die in hoofdstuk 7 onderzocht worden om tot een optimaal werkend systeem te komen. Ten slotte zullen ook nog de bouwblokken aangekaard worden die meer uitvoerig besproken worden in de volgende hoofdstukken.

### 3.1 Cel

Zoals besproken in hoofdstuk 2 is dit het bouwblok dat het vaakst terug te vinden is in het geheugensysteem. De cel bestaat uit een memristor en een transistor. De geheugencel heeft drie terminals: de gate van de transistor, die verbonden wordt met een wordline, de source van de transistor, die verbonden wordt met een sourceline en tenslotte de terminal van de memristor, die verbonden wordt met een bitline.

### 3.2 Branch

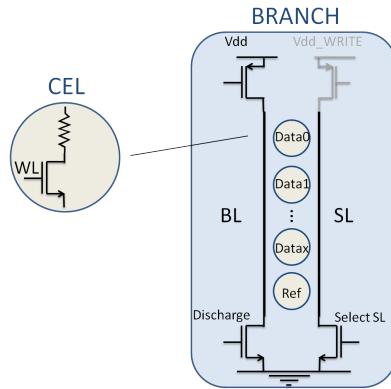
In een branch worden er een bepaald aantal datacellen verbonden aan één BL en één SL. Dit aantal wordt *Number of Word Lines per Branch* (NoWLpB) genoemd en is een van de vrijheidsgraden van de geheugenarchitectuur. Naast alle datacellen is er ook nog één referentiecel - dit zijn cellen waarvan de resistieve staat voorgeschreven is - verbonden aan de BL en SL van de branch. Elke BL wordt via een pMOS-transistor (al dan niet met nog een impedantie tussenin) gekoppeld aan de voedingsspanning Vdd en via een nMOS-transistor aan de grondspanning Vss. In dit werk is er enkel een nMOS-transistor die de SL verbindt met Vss.<sup>1</sup> De nMOS-transistoren aan BL en SL fungeren als schakelaars, de pMOS-transistor wordt gebruikt als impedantie voor een resistieve spanningsdeling (zie hoofdstuk 4). Ter illustratie wordt de samenhang tussen cel en branch getoond in figuur 3.1.

---

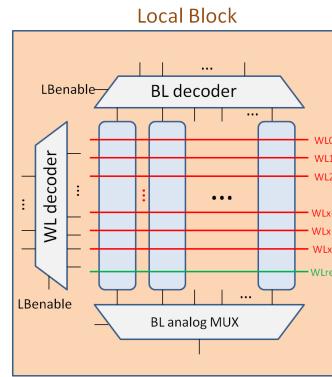
<sup>1</sup>In een volledig geheugensysteem zou de SL via een pMOS ook nog verbonden zijn met een niet onderzochte spanningsknoop Vdd\_write. De pMOS zou dan worden aangezet voor schrijfwerking.

### 3. GEHEUGENARCHITECTUUR

---



Figuur 3.1: Een geheugencel en een branch



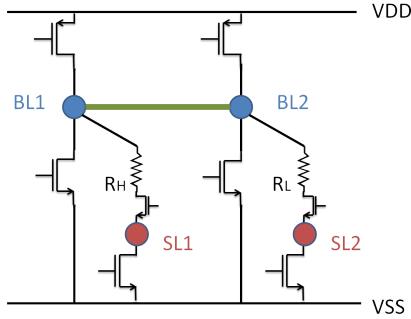
Figuur 3.2: Een Local Block

### 3.3 Local Block

Verschillende BLs en SLs worden samengebracht in een local block, waarvan de vrijheidsgraad *Number of Bit Lines per Local Block* (NoBLpLB) heet. In een LB bevinden er zich dus NoBLpLB x NoWLpB datacellen en NoBLpLB referentiecellen. Ook bevat een Local Block zowel BL- als WL-decoders. De structuur van een Local Block is geïllustreerd op figuur 3.2. De uitgangen van de WL-decoder sturen de data-WLs aan [eventueel met een buffer], de uitgangen van de BL-decoder activeren een spanningsdeling op de BLs.<sup>2</sup> De referentie-WL is via een extern signaal verbonden. Aangezien een LB zowel data- als referentiecellen bevat, gaat een LB twee werkingsmodes hebben: een mode waarbij er één datacel wordt aangesproken en een mode waarbij er een bepaald aantal referentiecellen in parallel wordt aangesproken.

---

<sup>2</sup>Indien schrijfwerking zou toegevoegd worden, zouden de uitgangen van de BL-decoder aan twee AND-poorten worden verbonden; bij leesoperatie brengt de uitgang van de ene AND-poort de resistieve deling op de BL teweeg, bij schrijfoperatie zet de uitgang van de andere AND-poort een pull-up-operatie van de BL naar Vdd\_write op.



Figuur 3.3: Topologie om referentiesignaal te verkrijgen

### 3.3.1 Data-signaal uitlezen

Het data-signaal is de spanning op de BL nadat er een resistieve deling is gebeurd, waarbij er stroom vloeit door één cel. De last die hangt aan de voedingsspanning, op figuur ?? voorgesteld als een pMOS-transistor, wordt aangeschakeld, alsook de nMOS-transistoren in de cel en aan de sourceline. Er vloeit een stroom langs dit pad:  $I=Vdd/R_{tot}$ , en de spanning op de BL is  $V=IReq$ .

### 3.3.2 Referentie-signaal uitlezen

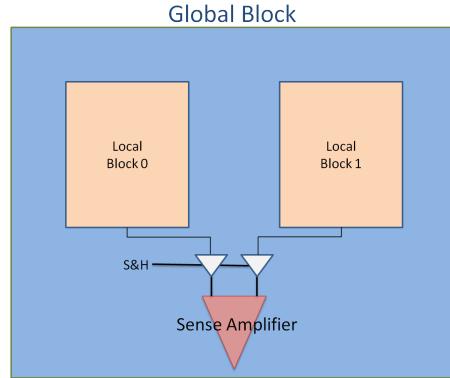
Het referentiesignaal is een spanning die tussen de spanning van een lage resistieve datacel en een hoge resistieve datacel moet liggen. Een dergelijk signaal kan verkregen worden door twee BLs kort te sluiten zoals op figuur 3.3. In dit ontwerp zal de kortsluiting gerealiseerd worden door de BLs via passgates te verbinden met een derde knooppunt. In theorie is het voldoende om 2 BLs [een aangesloten op een hoog resistief geheugenelement en een op een laag resistief geheugenelement] kort te sluiten om het referentiesignaal te verkrijgen. Er zit echter op de resistieve geheugenelementen variabiliteit: er wordt aangenomen dat  $R_H$  normaal verdeeld is met  $\mu = 32500\Omega$  en  $\sigma = 833\Omega$ .  $R_L$  is ook normaal verdeeld met  $\mu = 7500\Omega$  en  $\sigma = 833\Omega$ . Dit betekent dat ook de data-signalen en referentie-signalen stochastische variabelen zijn. Door echter steeds meer referentie-bitlijnen kort te sluiten gaat de spreiding van het referentiesignaal dalen. Bovendien kan men de verwachtingswaarde verschuiven door meer hoge (lage) resistieve referentiegeheugenelementen te gebruiken dan lage (hoge).

## 3.4 Global Block

Een global block bestaat uit twee LBs en een sense amplifier (SA). In het ene LB gaat er een datasignaal geproduceerd worden, in het andere een referentiesignaal (zie figuur 3.4). Vervolgens gaat de SA dit kleine signaalverschil versterken tot een zuivere rail-to-rail output. Aan de uitgang van het GB verschijnen dan ook de opgevraagde

### 3. GEHEUGENARCHITECTUUR

---



Figuur 3.4: Een Global Block

bits. De laatste architectuurvrijheidsgraad is de *Number of Global Blocks* (NoGB), het totale geheugen bevat dus NoGB x 2 x NoBLpLB x NoWLpB geheugencellen.

### 3.5 Besluit

De geheugenarchitectuur werd in vogelvlucht overlopen. De kleinste bouwblok is de cel, deze wordt geplaatst in een branch. Verschillende branches vormen samen een local block, dat ook decoders en multiplexers bevat. Twee local blocks en een sense amplifier met bijhorende passgates worden gegroepeerd tot een global block. Het totale geheugen bestaat tenslotte uit een verzameling global blocks.

## Hoofdstuk 4

# Lastimpedantie-analyse

Om een cel uit te lezen wordt er een spanning gegenereerd op de bitline door middel van een spanningsdeling. Het is dus belangrijk om de 2 impedanties van de spanningsdeler zodanig te kiezen voor optimale snelheid, bitline spanningsverschil en spanningsval over de memristor. Ook belangrijk is dat deze impedanties robuust zijn tegen variabiliteit.

### 4.1 algemene last eigenschappen en specificaties

In deze eerste sectie bestuderen we de combinatie van last en memristor cell als een heel simpel model namelijk twee weerstanden in serie (zie figuur 4.1a). Dit om aan te tonen dat de weerstands waarde van de last een grote invloed heeft op de het voltageverschil tussen een hoge en lage cel weerstand, bitlijn snelheid en de sensitivity van beide. Het verschil in bitlijn voltage tussen een hoge en lage cell is van belang voor de toleraties op de referentie voltage en sense amplifier mismatch. In het simple model kan het verschil in bitlijn voltage analitisch berekend worden met de volgende formule:

$$\Delta V = \frac{R_{HRS}}{R_{last} + R_{HRS}} - \frac{R_{LRS}}{R_{last} + R_{LRS}} \quad (4.1)$$

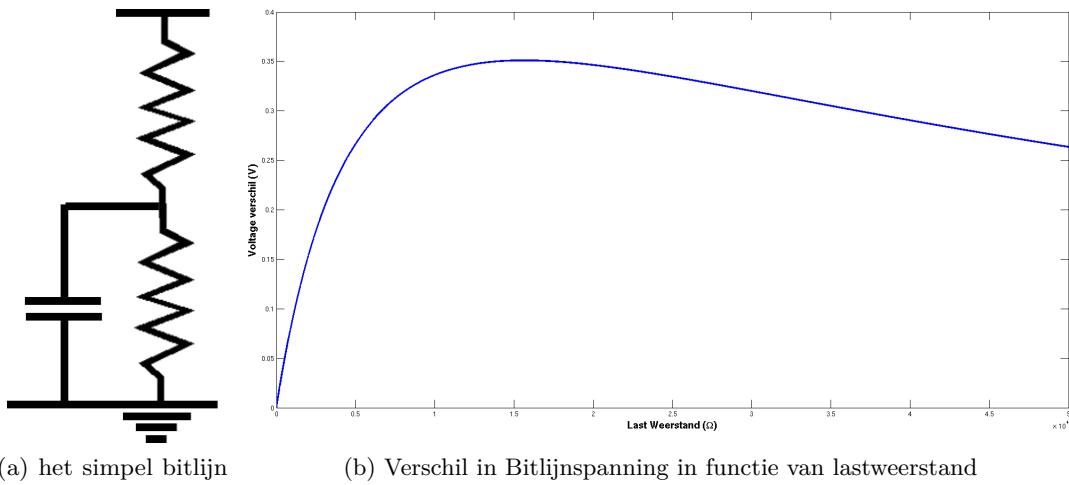
Voor constante waarden van  $R_{HRS}$  en  $R_{LRS}$  zal er een maximum zijn in  $\Delta V$  zoals duidelijk gezien kan worden in figuur 4.1b. De sensitiviteit van de last weerstand op het spanningsverschill, moet men voorzichtig interpreteren. Op figuur 4.1b kan gezien worden dat de helling voor de piek stijler is als na de piek. Het is dus beter om een iets grotere weerstand te hebben als een iets te kleine weerstand. Maar als men de weerstand naar transistor afmetingen vertaalt, kan met dit op verschillend manieren realiseren. Een grote weerstand realiseren met een transistor met minimale lengte, zal betekenen dat de breite van de transistor klein moet zijn. Dit zal dan gevoeliger zijn voor mismatch dan een grotere breite van transistor.

De snelheid van het opladen van de bitlijn kan in het simple model ook analitisch geschreven worden. De volgende vergelijking stelt de tijd voor waar de bitlijn 99% is opgeladen.

$$t = -\ln(0.01) * RC \quad (4.2)$$

$$R^{-1} = \frac{1}{R_{cell}} + \frac{1}{R_{last}} \quad (4.3)$$

Deze tijd zal kleiner worden als de R kleiner word, dit vertaalt zich dan naar een kleine last weerstand.



(a) het simpel bitlijn model

(b) Verschil in Bitlijnspanning in functie van lastweerstand

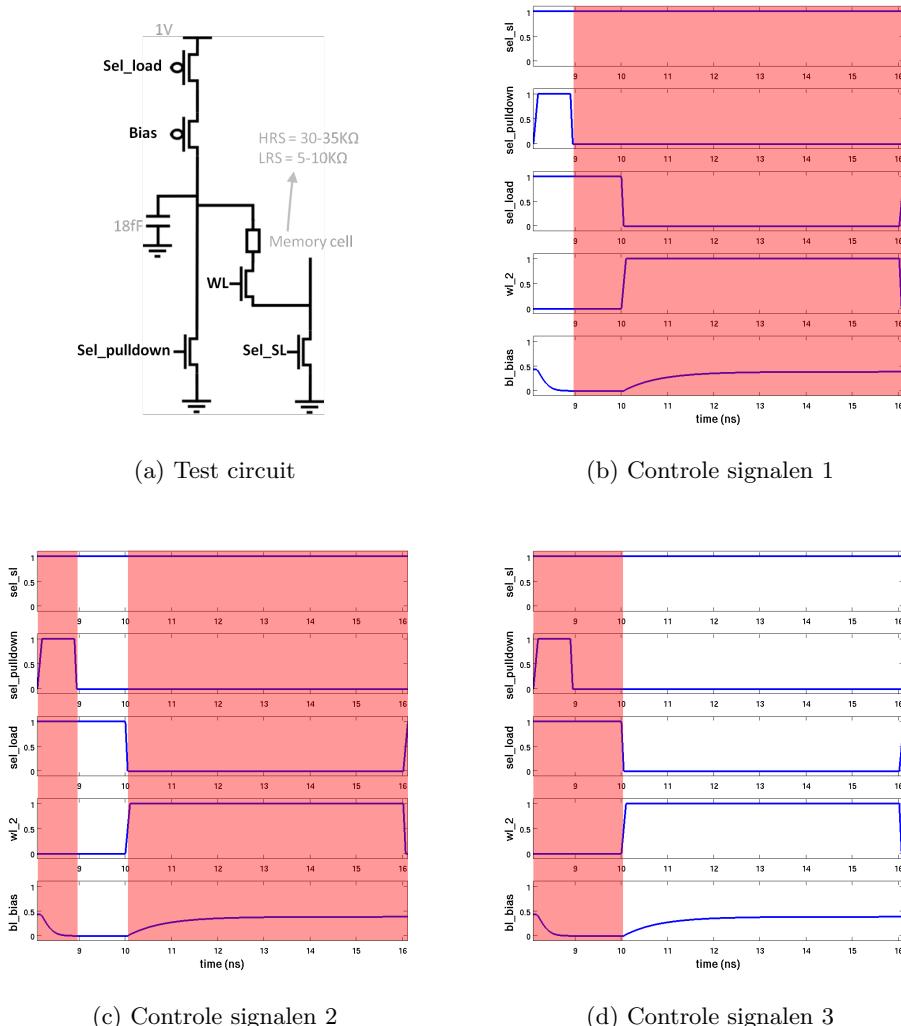
Figuur 4.1:

## 4.2 evalueren van de last

Om verschillende lasten met elkaar te kunnen vergelijken, is het belangrijk om hun eigenschappen allemaal op dezelfde manier te bekomen. Figuur 4.2 geeft de verschillende aspecten van de gebruikte simulatie setup weer. Het test circuit (Figuur 4.2a) stelt een bitlijn voor met een capaciteit van 18fF, wat ruiw weg overeenkomt met een bitlijn met 100 cellen op. Aan deze bitlijn zijn een last, een ontlaad transistor en een memristor weerstand aangesloten. De ontlaad transistor is minimaal gehouden. De memristor weerstand kan de volgende waardes hebben: tussen  $5k\Omega$  en  $10k\Omega$  voor de LRS, tussen  $30k\Omega$  en  $35k\Omega$  voor de HRS. De nominale waardes voor LRS en HRS zijn  $7.5k\Omega$  en  $32.5k\Omega$ . Tijdens monte-carlo analyses worden dan deze nominale waardes als gemiddelde van een gausische distributie genomen met  $\sigma = 0.833k\Omega$ . Aan deze memristor weerstand hangt een select transistor, die ook minimaal gehouden wordt, en de combinatie van deze wordt de geheugen cell genoemd. Aan deze geheugen cell hangt nog een selectlijn transistor met een breedte van 500nm. Deze transistor werd bewust groot gemaakt om de totale weerstand in de onderste tak voornamelijk te laten afhangen van de geheugen cell. Aan deze selectlijn werd er ook

een capaciteit van  $18\text{fF}$  aan gehangen, deze doet echter niet veel aangezien de select transistor altijd aangelaten wordt. Tenslotte wordt de voedingsspanning altijd op  $1\text{V}$  gehouden.

Figuren 4.2b tot 4.2d stellen de sequentie voor van alle controlesignalen uit tijden de simulatie. Eerst wordt de bitlijn volledig ongeladen (figuur 4.2b). Vervolgens is er een interval waar niks gebeurt (figuur 4.2c) en tenslotte wordt te last aangesloten en de bitlijn opgeladen (figuur 4.2d). De simulatie stopt als de bitlijn volledig opgeladen is.



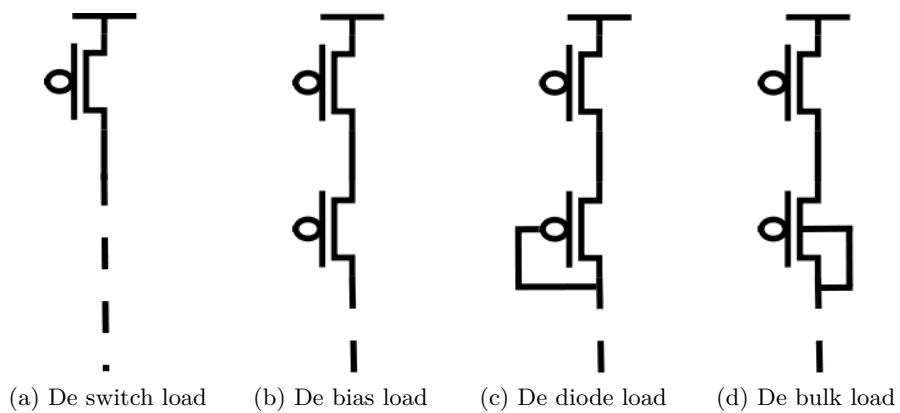
Figuur 4.2: Test bench voor de last

Eens een last gesimuleert is wordt het beoordeeld op het vlak van oppervlakte,

bitlijn oplaadsnelheid, nominaal bitlijn voltage verschil en spanningsval over de cel. Het oppervlakte wordt berekend op basis van de lengtes en breedtes van de last transistoren. De bitlijn oplaadsnelheid is de tijd dat nodig is om de bitlijn 99% op te laden. Het nominale bitlijn voltageverschil is het verschil in bitlijnvoltage tussen een cell in HRS en LRS, wanneer de bitlijn 100% opgeladen is. De bitlijn wordt veronderstelt 100% opgeladen te zijn op het einde van de simulatie en de simulatietijd wordt voldoende lang gehouden om dit te garanderen. De spanningsval over de cel is belangrijk opdat de cel in van state wisselt gedurende de leescyclus. Aangezien de cel voorgesteld wordt met een weerstand zal dit natuurlijk nooit gebeuren maar dit is wel belangrijk moest er met echte memristors gewerkt worden. De numerieke waarde van de maximale spanningsval over de cel is heel erg afhankelijk van het type memristor. In dit onderzoek wordt er gewerkt met een maximum van 0.5V over de cell [15].

### 4.3 vergelijking van verschillende types last

Voor dit onderzoek worden vier mogelijke kandidaten van last vergeleken: de switchload (figuur 4.3a), de biasload (figuur 4.3b), de diodeload (figuur 4.3c) en de bulkload (figuur 4.3d)[13]. Eerst wordt er een lineare sweep gedaan op de verschillende lasten (sectie 4.3.1), waarbij enkel de breedtes en bias spanningen worden geswept. De lengtes van de transistoren worden minimaal gehouden om er voor te zorgen dat de transistoren binnen de pitch van de bitlijn passen. Eens variabiliteit wordt toegevoegd aan de simulatie onder de vorm van monte-carlo (sectie 4.3.2), zal echter blijken dat er het verschil in bitlijnvoltage te klein is, en zal de lengte van de last transistoren ook moeten worden vergroot (sectie 4.3.3).



Figuur 4.3: De verschillende types last

#### 4.3.1 Lineaire sweep op de lasten

De switchload bestaat uit één pmos transistor die volledig wordt aan of afgesloten. Een lineare sweep met een breedte van de transistor tussen 100nm en 500nm werd gedaan en is geïllustreerd in figuur 4.4. Bij het vergroten van de breedte van de transistor zal de weerstand dalen en het verschil tussen de bitlijnen ook. Als we deze last vergelijken met het simpele model uit sectie 4.1, zit de weerstand waarde aan de linker kant van de piek uit figuur 4.1b. Bij het vergroten van de transistor breedte zal de bitlijn spanning stijgen en de spanningsval over de cell dus ook. Verder volgt de risetime ook het simple model uit sectie 4.1, waarbij de risetime daalt bij kleinere weerstandswaarden.

De biasload, is een last met twee pmos transistoren in serie. Bovenste van de twee wordt als een switch gebruikt en dus volledig aan of af gesloten. De onderste van de twee wordt op een spanning gebiased. Het voordeel van de biasload is dat men een grotere weerstand kan maken en dus de piek kan bereiken uit figuur 4.1b. Dit kan men duidelijk zien op de x-assen van figuur 4.5. Ook hier zijn de breedtes van de transistoren gesweept tussen 100nm en 500nm. De bias spanning is tussen 0V en 0.4V gesweept. Een hogere bias spanning brengt echter geen nuttige bijdrage. Door dat te kleinste weerstand dat met deze last te maken is, binnen deze sweeprange, net iets groter is als deze van de switch load, is de biasload ook iets trager. De oplossingen waarbij dit het geval is, hebben echter een onbruikbaar verschil in bitlijn voltages. De spanningsval over de cell is vergeleken met de switch load heel wat hoger maar voor de meeste oplossingen ligt het nog altijd onder de limiet van 0.5V.

De diode load bestaat ook uit twee transistoren, de bovenste wordt net als bij de biasload als een switch gebruikt. De onderste is als een diode geconnecteerde transistor gekoppeld. Uit de sweep resultaten (figuur 4.6) blijkt dat deze last heel snel is maar veel te kleinen bitlijn voltage verschillen heeft om bruikbaar te zijn.

De bulkload werd voorgestelt in de paper van Ren et al. [13] als een goede kandidaat omdat van zijn grote uitgangsimpedantie. Deze last bestaat uit een switch transistor en een bulk geconnecteerde transistor. Deze bulk geconnecteerde transistor wordt op 0V gebiaast aangezien deze de beste resultaten gaf. De breedtes van de transistoren zijn gesweept tussen 100nm en 500nm. De resultaten van deze sweep zijn geïllustreerd in figuur 4.7. In de resultaten kan gezien worden dat deze last zich vergelijkbaar gedraagt als de biaslast. Enkel op het vlak van risetime zijn er oplossingen die beter zijn.

#### 4.3.2 Het toevoegen van variabiliteit

Na een selectie te hebben gemaakt van de oplossingen uit de vorige sectie, worden met deze oplossingen nieuwe simulaties gedaan waarbij er variabiliteit is toegevoegd. Deze variabiliteit is toegevoegd op alle transistoren in het test circuit en op de weerstands waarde van de geheugen cellen. Voor de transistoren wordt er een Pelgrom constante

#### 4. LASTIMPEDANTIE-ANALYSE

---

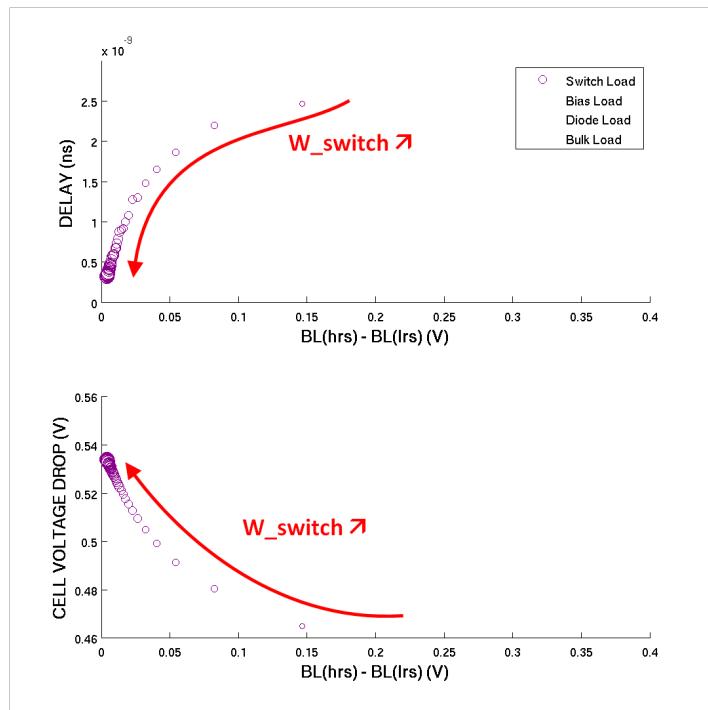
voor  $vt$  van  $2.5mV\mu m$  gebruikt en voor  $\beta$  van  $1.2\mu m$  gebruikt [8]. Voor de weerstand waarde van de memristors word er een gausische verdeling gebruikt met nominale waardes  $7.5k\Omega$  en  $32.5k\Omega$  en met  $\sigma = 0.833k\Omega$ . Er worden telkens 500 monte carlo simulaties gedaan per oplossing. Hierna worden de bitlijn voltages van cellen met een HRS en LRS gefit op een gausische distributie. De oplossing met het grootste bitlijn voltage verschil tussen de extrema van HRS en LRS is een biasload met een switch transistor breedte van 100nm, een bias transistor breedte van 180nm en een bias voltage van 0V. De bitlijn voltage distributie zijn geïllustreerd op figuur 4.8. Het voltage verschil tussen de CDF = 0.1% van HRS en CDF = 99.9% van de LRS is 65mV. Dit is niet veel aangezien de distributie van het bitlijn voltage van de referentie cell hier tussen moet passen en er daarna nog marge over moet zijn voor variabiliteit in de senseamplifier. Het aandeel to variabiliteit van beide transistoren in de last is even groot.

Figuur 4.9 stelt de distributie van het bitlijn voltage van de referentie cellen voor. Hier bij varieert het aantal referentie cellen van 2 tot 30 en er werd een even groot aantal referentie cell in HRS als LRS gehouden. Zoals gezien kan worden dat met een heel aantal cellen nodig heeft om een distributie breedte van 39mV te krijgen. Dit geeft dan een marge van ongeveer 10mV voor de senseamplifier wat helemaal niet veel is. Daarom word de constraint waarbij de transistor lengte minimaal gehouden word opgeheven in de volgende sectie.

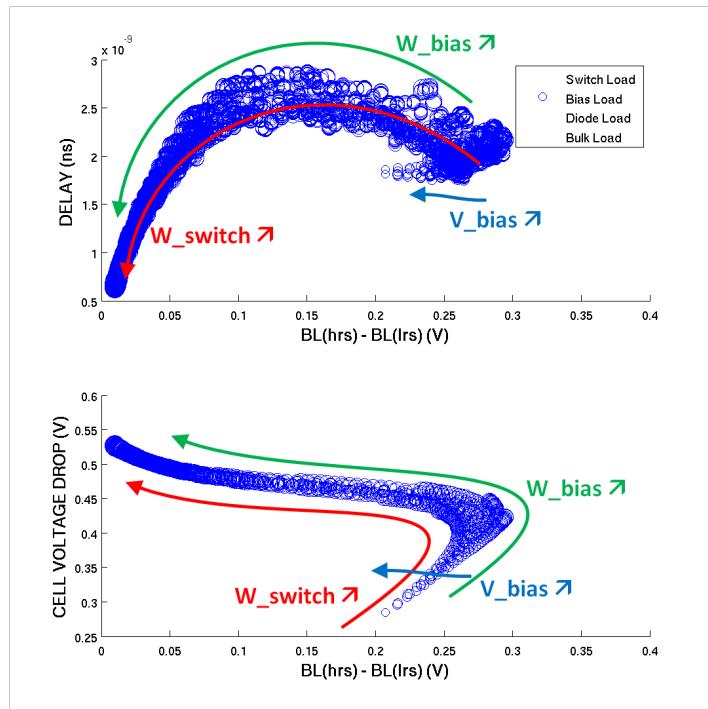
##### 4.3.3 De transistor lengte vergroten

Om de variabiliteit onder controle te houden moeten de transistoren vergroot worden. Twee opties worden hiervoor overwogen. De eerste is het toevoegen van een derde transistor in serie. Om de zelfde last impedantie te bekomen als voor 2 transistoren in serie, moeten alle drie transistoren een grote breedte hebben wat zou betekenen dat ze groter zijn en minder gevoelig voor mismatch. Een aspect waar niet mee rekening word gehouden in die redenering is de toestand waarin deze transistoren zich bevinden. Bij drie transistoren in serie zal de onderste van de drie zich in near tot sub-theashold bevinden. De stroom in het sub-threshold gebied is exponentieel met de gate-source spanning dit levered grote variatie in de stroom voor kleine vt mismatch. Dit fenomeen zien men niet bij 2 transistoren in serie, aangezien de transistoren hier in het lineare gebied zijn. Daarom word er gekozen voor een tweede optie om de mismatch onder controle te houden namelijk het vergroten van de lengte van de transistor. Als men de lengte vergroot, Stijgt de weerstand wat dan weer gecompenseert kan worden door de breedte ook wat te vergroten. Nu men deze constraint laten varen heeft, word er dan ook geopteerd om een switchload ipv een bias load te gebruiken.

Figuur 4.10 geeft de resultaten weer van een sweep van verschillende lengtes en breedtes voor een switchload. De resultaten worden voor gesteld in functie van  $W/L$  wat een indicatie is voor de weerstand van de transistor. In de bovenste figuur kan men duidelijk een maximun zien voor het verschil in bitlijn voltage zoals in sectie 4.1 werd voorspelt. Verder wordt opgemerkt dat er best één van de oplossingen aan de linkerkant van het maximum gekozen wordt aangezien de spannings val over de cell



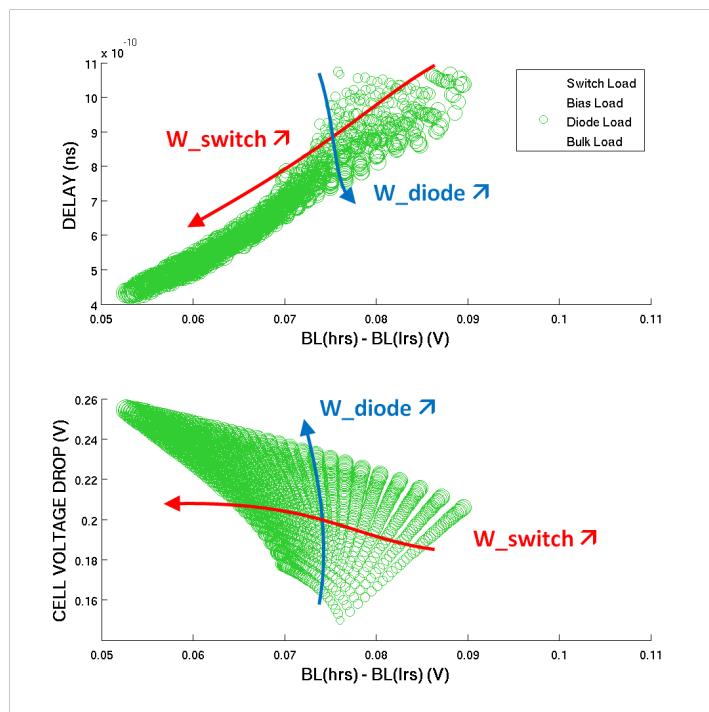
Figuur 4.4: Lineaire sweep van switchload



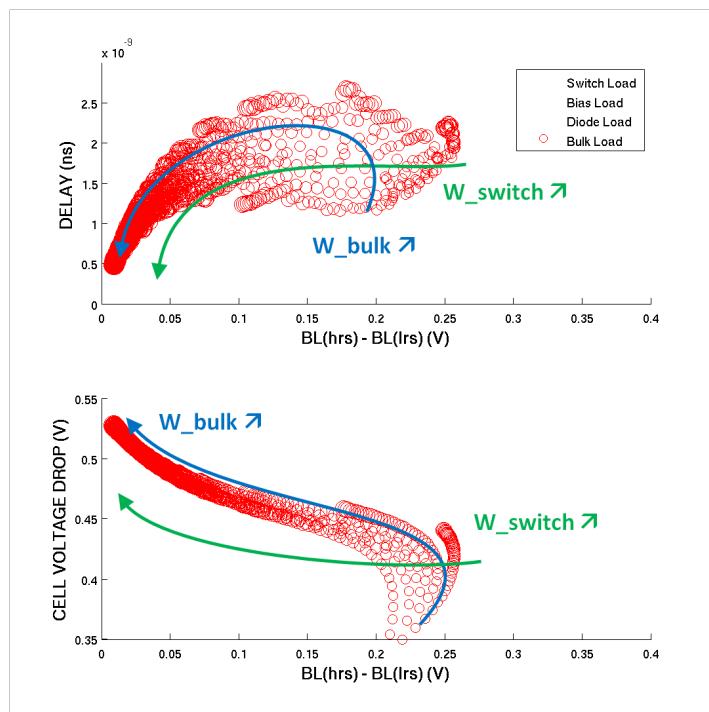
Figuur 4.5: Lineaire sweep van biasload

#### 4. LASTIMPEDANTIE-ANALYSE

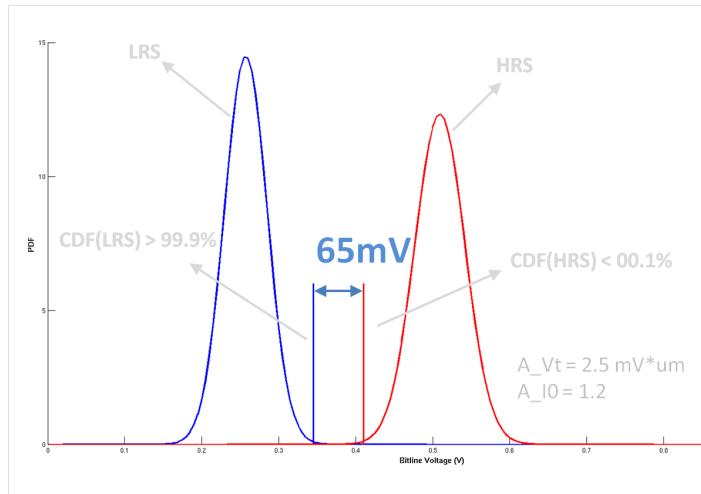
---



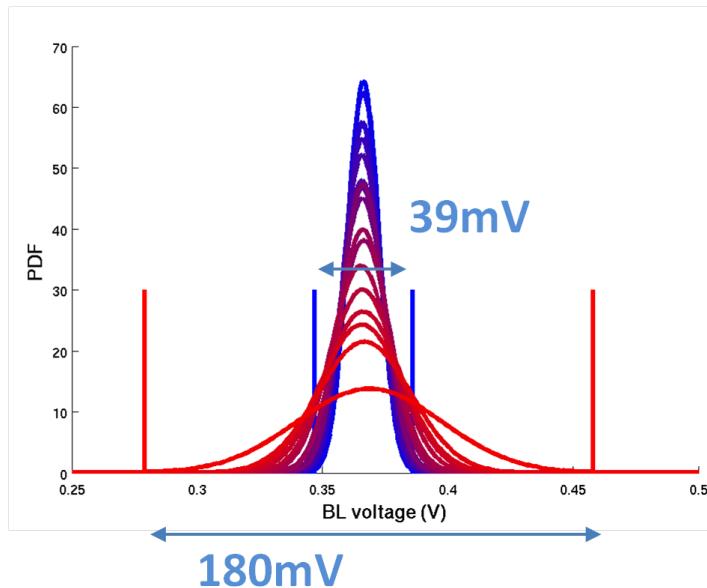
Figuur 4.6: Lineaire sweep van diodeload



Figuur 4.7: Lineaire sweep van bulkload



Figuur 4.8: Bitlijn voltage distributie voor een biasload



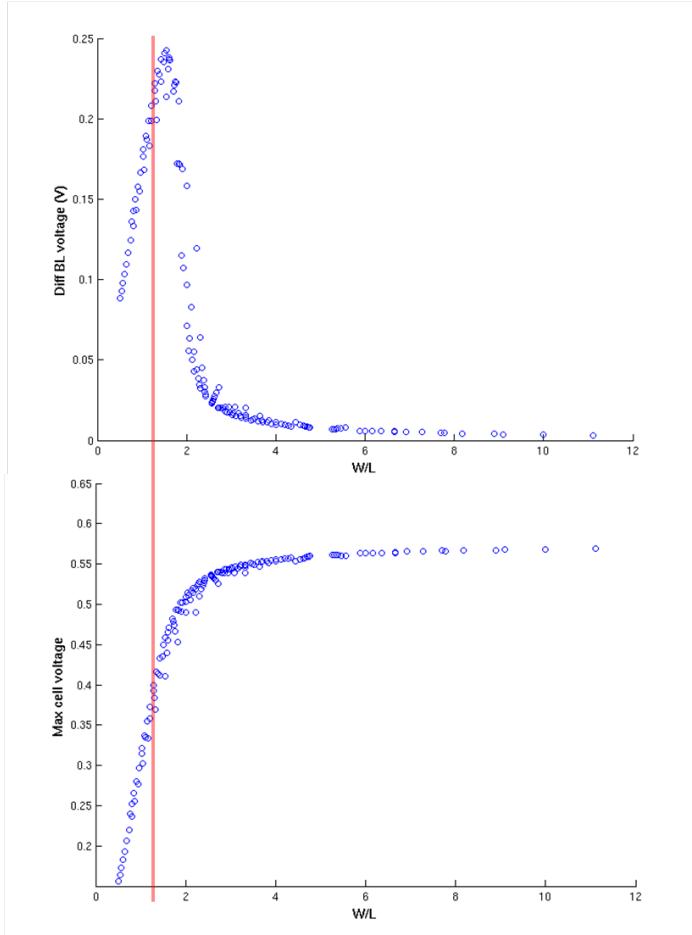
Figuur 4.9: Lineaire sweep van switchload

van de oplossingen aan de rechterkant van het maximum te hoog zijn.

Voor de finale last wordt er geopteert voor een transistor met lengte gelijk aan 198nm en breedte gelijk aan 300nm. Op figuur 4.10 wordt deze aangeduid met de rode lijn. Op figuur 4.11 wordt de bitlijn voltage distributie van deze last getoont. Het minimale verschil in bitlijn voltage is bijna 200mV. De distributie van bitlijn voltage van de referentie is ook aangegeven op deze figuur. Deze bestaat hier uit 4 referentie cellen waarvan 2 in HRS en 2 in LRS. Opvallend is dat deze referentie niet

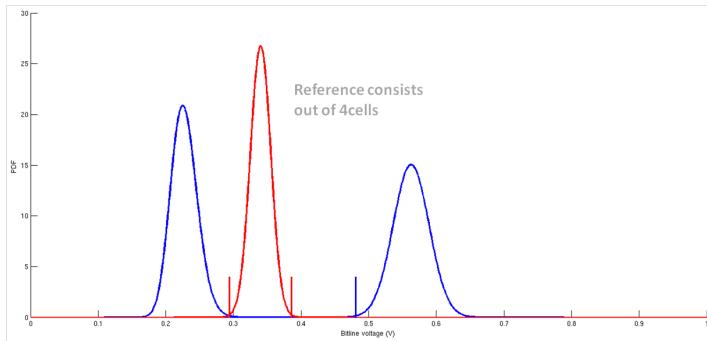
#### 4. LASTIMPEDANTIE-ANALYSE

---



Figuur 4.10: Verschillende oplossingen voor de switchload met variabele lengtes en breedtes

in het centrum zit tussen de bitlijn voltages van de cellen. Dit kan opgelost worden door een niet gelijk aantal referentie cellen in HRS en LRS te hebben. Aangezien de standaard deviatie op de bitlijn voltages heel wat beter is nu de lengte van de transistoren ook wordt gesized, kan er gerust gekozen worden voor een last met een kleiner nominaal verschil in bitlijn voltages . Dit kan 2 voordelen met zich mee brengen. Het eerste is dat de spannings val over de cell verlaagd kan worden als met voor een oplossing kiest dan meer links zit van het maximum in figuur 4.10. Het Tweede is dat men voor een oplossing kan kiezen waarbij de bitlijn voltages lager zijn wat een energie winst kan opleveren. Ondanks deze voordelen werd er toch geopteerd voor de oplossing met het grootste bitlijn voltage verschil.



Figuur 4.11: Bitlijn voltage distributie voor de finale load

#### 4.4 Besluit

Verschillende kandidaten voor lastimpedanties werden overwogen. Aanvankelijk werd er getracht een last met minimale transistorlengtes te vinden, dit bleek echter niet haalbaar wanneer variabiliteit in rekening wordt genomen. Een enkele transistor met niet-minimale afmetingen bleek optimale resultaten te leveren wat betreft BL-spanningsverschil en spanningsval over geheugenelement.



# Hoofdstuk 5

## Sense Amplifier analyse

Een sense amplifier versterkt kleine signaalverschillen tot rail-tot-rail signalen. Aangezien de uitgangsignalen hiervan ook de uitgelezen bits zijn van het geheugen, is het bovenal belangrijk dat dit op een correcte manier gebeurt, ondanks variabiliteit. Het is dus logisch om de sense amplifier wat meer te onderzoeken en zodanig te ontwerpen op een robuuste manier, terwijl er ook rekening gehouden wordt met energie en snelheid.

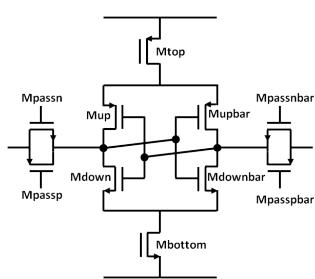
### 5.1 Types SA

...

In wat volgt zal er worden voortgewerkt met de SA van figuur 5.1.

### 5.2 Offsetspanning

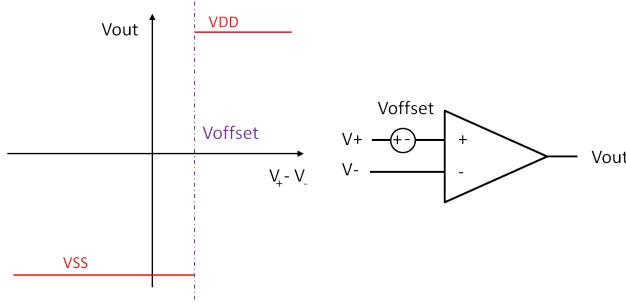
Een ideale sense amplifier zal voor elke twee ingangssignalen correct versterken, tenzij de signalen dezelfde zijn, waarna de SA in een metastabiele toestand belandt. In de praktijk is er echter wegens variabiliteit een limiet voor het ingangsspanningsverschil waarbij er correct versterkt wordt. Deze limiet heet de offsetspanning en wordt geïllustreerd in figuur 5.2. De offsetspanning van een SA is in de ontwerpfasen een stochastische variabele met gemiddelde 0V, pas nadat een chip gefabriceerd is ligt de



Figuur 5.1: een sense amplifier

## 5. SENSE AMPLIFIER ANALYSE

---



Figuur 5.2: Illustratie van offsetspanning

offsetspanning definitief vast [al kan het zijn dat deze met de tijd nog verandert]. Er zijn 2 manieren waarop men de offsetspanning van een systeem kan aanpakken: ofwel ontwerp je het systeem zodanig dat het verschil van de ingangssignalen van de SA groot genoeg is zodat ze [in 99,9% van de gevallen] niet groter is dan de offsetspanning, ofwel bouw je een mechanisme in waarbij je na fabricatie de offsetspanning meet en vervolgens compenseert. In dit werk is gekozen voor het eerste. Hiervoor is het wel belangrijk te onderzoeken wat de verdeling is van de offsetspanning, dit wordt gedaan in de volgende sectie.

### 5.3 Sensitiviteitsanalyse

De SA wordt gerealiseerd als een circuit met transistors. Elke transistor heeft 2 stochastische parameters met een normale verdeling, nl.  $\Delta V_t$  en  $\Delta \beta$ . De spreiding van deze verdelingen is gekend:  $\sigma_{\Delta V_t} = \frac{A_{V_t}}{\sqrt{WL}}$  en  $\sigma_{\Delta \beta} = \frac{A_\beta}{\sqrt{WL}}$ . Met een sensitiviteitsanalyse kan men uit deze standaardafwijkingen de standaardafwijking van de offsetspanning  $\sigma_{V_{offset}}$  berekenen. Hierbij wordt verondersteld dat de stochastische variabele  $V_{offset}$  een lineaire combinatie is van de normaal verdeelde afwijkingen  $(\Delta V_t)_i$  en  $(\frac{\Delta \beta}{\beta})_i$ :

$V_{offset} = \sum_{i=1}^N a_i (\Delta V_t)_i + b_i \left(\frac{\Delta \beta}{\beta}\right)_i$ .  $a_i$  en  $b_i$  zijn de gevoeligheden van de offset naar de variatieparameters:  $a_i = \frac{\partial V_{offset}}{\partial (\Delta V_t)_i}$  en  $b_i = \frac{\partial V_{offset}}{\partial (\frac{\Delta \beta}{\beta})_i}$ . Voor een dergelijke variabele geldt

$$\text{dan: } \sigma_{V_{offset}} = \sqrt{\sum_{i=1}^N a_i^2 (\sigma_{\Delta V_t})_i^2 + b_i^2 (\sigma_{\frac{\Delta \beta}{\beta}})_i^2}.$$

Er moet wel geverifieerd worden of de stelling dat er een lineaire afhankelijkheid is tussen  $V_{offset}$  en de variatieparameters gegronde is. Dit kan gedaan worden aan de hand van een analyse waarbij elke variatieparameter afzonderlijk geswept wordt.

#### 5.3.1 Sensitiviteitsanalyse op een minimale SA

In figuur ?? wordt het resultaat getoond voor een dergelijke analyse bij een SA met minimale afmetingen, merk op dat de richtingscoëfficient van deze curves gelijk is aan  $a_i(\Delta V_t)_i$  en  $b_i(\frac{\Delta \beta}{\beta})_i$ . In tabel 5.1 worden de resultaten en de resulterende

Transistor	Parameter	Richtingscoëfficiënt [ $\frac{mV}{\sigma}$ ]	W [nm]	L [nm]	$\sigma$
Mupbar	Vt	22.733	100	45	37.2678mV
Mup	Vt	-22.250	100	45	37.2678mV
Mupbar	$\beta$	13.583	100	45	17.8885%
Mpassn	$\beta$	13.467	100	45	29.8142%
Mpassbarn	$\beta$	-13.117	100	45	29.8142%
Mup	$\beta$	-13.033	100	45	17.8885%
Mdownbar	$\beta$	-9.383	100	45	29.8142%
Mdown	Vt	-9.267	100	45	42.0381mV
Mdownbar	Vt	9.233	100	45	42.0381mV
Mdown	$\beta$	8.217	100	45	29.8142%
Mpassp	$\beta$	-4.50	100	45	17.8885%
Mpassbarp	$\beta$	4.383	100	45	17.8885%
Mpassbarp	Vt	0.70	100	45	37.2678mV
Mpassp	Vt	-0.70	100	45	37.2678mV
Mbottom	$\beta$	0.083	100	45	29.8142%
Mbottom	Vt	-0.033	100	45	42.0381mV
Mpassbarn	Vt	0	100	45	42.0381mV
Mpassn	Vt	0	100	45	42.0381mV
Mtop	Vt	0	100	45	37.2678mV
Mtop	$\beta$	0	100	45	17.8885%
$\sigma_{V_{offset}}$ :		45.6813mV			

Tabel 5.1: Sensitiviteitsanalyse van de minimale SA

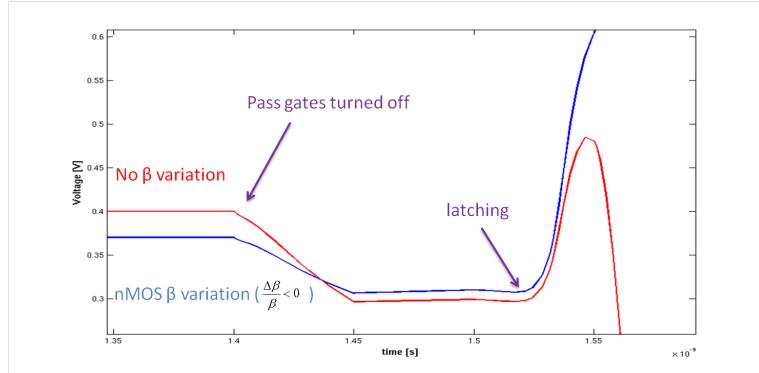
standaardvariatie van de SA getoond. Er moet opgemerkt worden dat er bij deze simulatie slechts geswept werd voor de variatieparameters van  $-4\sigma$  tot  $4\sigma$ . Dit is om de reden dat voor de minimale transistoren de standaardvariatie het grootst is. In de Spectre-simulaties zouden transistoren voor te grote negatieve  $\beta$ -mismatch stroom leveren in de omgekeerde richting. Deze situatie zal fysisch nooit optreden.

Opmerkelijk bij deze analyse is dat er een significante bijdrage is van de pass-gates door  $\beta$ -mismatch. Een nadere observatie leert dat deze bijdrage optreedt door ladingsinjectie van de pass-gates die niet meer gematched is (zie figuur 5.3). Hierbij moet wel worden opgemerkt dat voor deze simulatie er geen overlap is tussen het controlesignaal op de passgate aan te zetten en het signaal om de SA te activeren. De reden hierachter is dat als er overlap tussen deze signalen is, de SA ook de BL zou trachten op te laden. Hierbij zou er moeten ingeboet worden aan snelheid en het zou ook extra energie kosten.

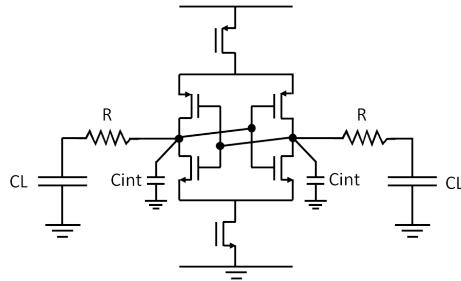
Men kan argumenteren dat er een korte overlap zou kunnen toegelaten zijn, waarna er voldoende spanningsverschil tussen de 2 ingangs-uitgangsknopen zou opgebouwd zijn opdat de ladingsinjectie geen effect meer kan hebben op het eindresultaat. Een tegenargument is dat de timing hiervoor te precies moet zijn.

## 5. SENSE AMPLIFIER ANALYSE

---



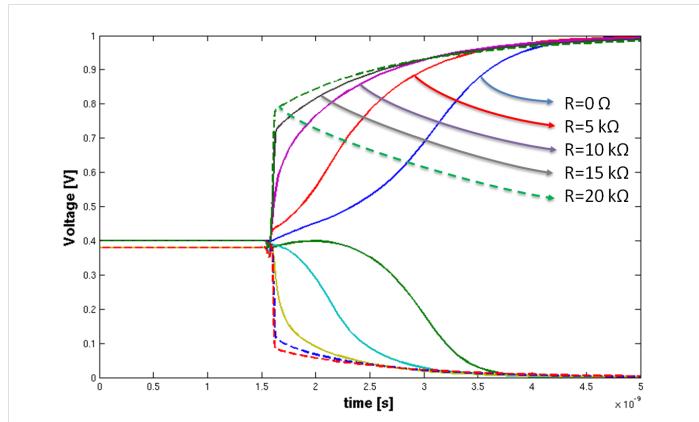
Figuur 5.3: Door  $\beta$ -mismatch is ladingsinjectie van de pass-gates niet meer gematched en gaat de SA foutief latchen



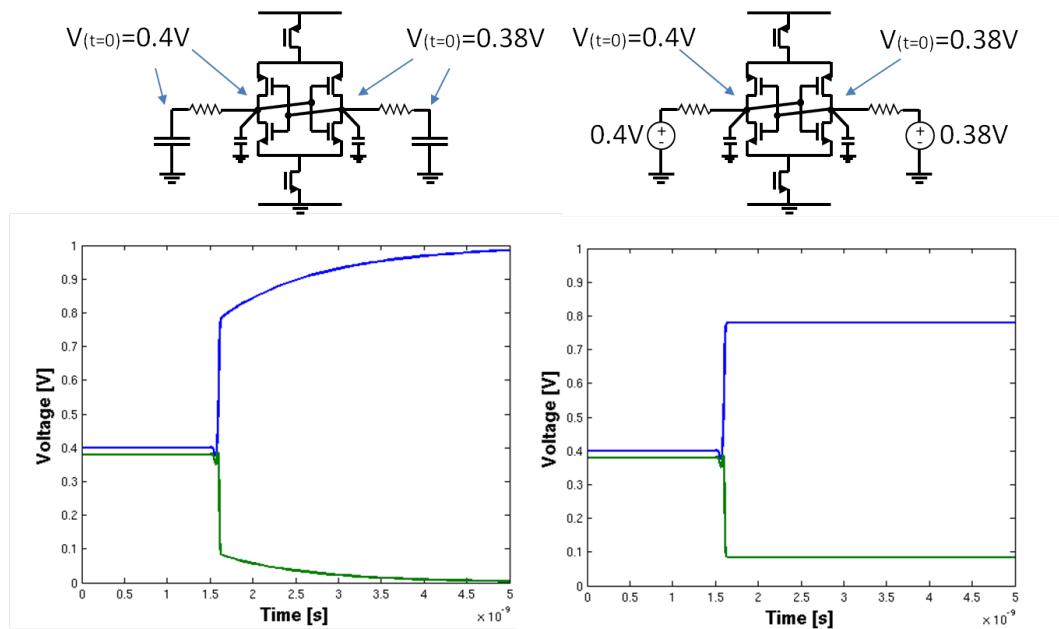
Figuur 5.4: Simulatieopstelling voor het RC-latch-effect

### 5.3.2 RC-latch-effect

De situatie waarbij er volledige overlap is tussen de controle signalen kan vereenvoudigd worden opgesteld met de situatie van figuur 5.4. De pass-gate die aanstaat wordt voorgesteld als een weerstand, de pass-gate in het local block en diens parasitaire capaciteit wordt verwaarloosd. CL bedraagt voor deze simulatie 46 fF, het equivalent voor een BL waaraan 256 cellen hangen. Cint bedraagt voor een SA met minimale transistorafmetingen 161 aF. Wanneer het dynamisch latch gedrag bekijken wordt voor verschillende waarden van R, treedt er een merkwaardig effect op (zie figuur 5.5): voor voldoende grote waarden van R lijkt het alsof de grote capaciteit ontkoppeld is van de latch tot op een zeker tijdstip, waarna een veel tragere settling optreedt. De verklaring ligt in het feit dat CL zich voor hoge frequenties als een kortsluiting gedraagt (zie figuur 5.6), een plotse stroom vloeit door de weerstand en hierdoor onstaat er een spanningsval over de weerstand. Hierna gaat er op veel lagere frequenties een spanning beginnen op te bouwen over de capaciteit waardoor de ingangs-uitgansknopen volledig kunnen laden/ontladen tot VDD en VSS. Gevolgen van wanneer dit effect optreedt is dus dat het nuttige signaal zich snel - alsof er helemaal geen last aanhangt - en lineair opbouwt en dat er geen AC-signaal is over de condensator. Een analyse van de respons van een RC-circuit



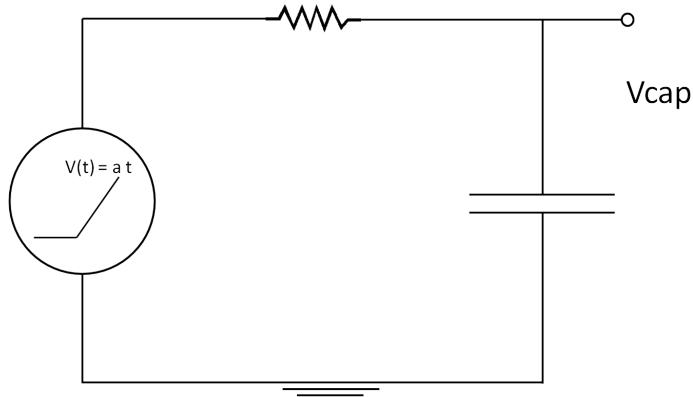
Figuur 5.5: Simulatieresultaten voor het RC-latch-effect: de 2 ingangs-uitgangsknopen zijn voorgeladen op 400mV en 380mV. Na 1,6ns wordt de SA aangezet. De SA is ideaal voor deze simulatie.



Figuur 5.6: Vergelijking situatie met voorgeladen (eindige) capaciteit en situatie met spanningsbron (oneindige capaciteit)

## 5. SENSE AMPLIFIER ANALYSE

---



Figuur 5.7: Circuit voor analyse voorwaarden RC-latch-effect

op een lineair stijgende spanningsbron geeft meer duidelijkheid voor de voorwaarden waarop het RC-latch-effect optreedt (zie figuur 5.7). De respons van de spanning over de capaciteit is  $V_{cap}(t) = at - aRC(1 - e^{-\frac{t}{CR}})$ . Uit deze uitdrukking blijkt dat het RC-latch-effect optreedt wanneer de latch zonder last snel is ( $a \ll 1$ ) en/of wanneer het RC-product hoog is ( $RC \gg 1$ ). Wanneer het effect zich voordoet zijn latching en RC-respons onafhankelijke processen. Wanneer de voorwaarden niet meer zo uitgesproken zijn, gaan deze processen met elkaar interfereren en is het moeilijk dit gecombineerde proces wiskundig te beschrijven.

Conclusie van het RC-latch effect is dat de timing helemaal niet zo kritisch is: in theorie hoeft de overlap slechts even lang te duren als de delay van de SA wanneer er geen last op is aangesloten, maar het is niet erg als de overlap wat langer duurt. De pass-gates mogen ook minimaal zijn, om hun aanweerstand te vergroten zodat het effect kan optreden. In geval verder zou gewerkt worden met een SA zonder overlap met pass-gate-enable en SA-enable, zouden de pass-gates moeten geschaald worden om de mismatch te minimaliseren. Dit zou wel betekenen dat er per schakeling van de passgates een grotere hoeveelheid lading wordt geïnjecteerd.

### 5.3.3 Sensitiviteitsanalyse voor minimale SA - vervolg

In tabel ?? worden de resultaten van een nieuwe sensitiviteitsanalyse getoond voor een minimale SA, ditmaal waarbij er dus overlap is tussen pass-gate-enable en SA-enable. In deze situatie dragen de pass-gates amper bij tot de offsetspanning. Vooral de transistors van de differentiële paren zullen opgeschaald moeten worden om de offsetspanning in te perken, de andere transistoren schalen zal invloed hebben op snelheid en energie.

### 5.3.4 Sensitiviteitsanalyse voor gebruikte SA

In tabel ?? worden de resultaten van een sensitiviteitsanalyse getoond voor de SA die gebruikt wordt in het finale geheugenontwerp. Deze is gekozen aan de hand van

de resultaten van de paretosimulatie in de volgende sectie.

## 5.4 Paretosimulatie

In het beginstadium van het ontwerp is nog niet duidelijk wat de impedantie aan de BL wordt. Het is deze impedantie die bepaalt wat het spanningsverschil is tussen het datasignaal en het referentiesignaal aan de sense amplifier. Bovendien kan het zijn dat er midden in het ontwerp besloten wordt om een andere impedantie te kiezen om alsnog te optimaliseren naar een andere variabele. Natuurlijk is het mogelijk om één SA te gebruiken die voor elke impedantie een correcte en snelle werking zou garanderen. Dit zou echter een verspilling zijn van energie. In deze sectie wordt een pareto-oppervlak opgesteld waarbij er voor elk spanningsverschil de snelste en energiezuinigste SA-ontwerpen worden gekozen.

### 5.4.1 Opstelling

Uit een verzameling van allerhande SA [dit zijn sense amplifiers waarvan de transistoren verschillend geschaald zijn - differentiële paren hebben zelfde afmetingen] worden enkel de pareto-optimale SA uitgekozen. De pareto-criteria zijn  $\Delta V$ , snelheid en dynamische energie.

Voor deze opstelling worden de pass-gates weggelaten van de SA [dit is geoorloofd zoals bleek uit de sensitiviteitsanalyse], de last aan de ingangs-uitgangsknopen is een simpele CMOS inverter. De knopen zijn voorgeladen op 2 spanningen: 0,4V en 0,4V -  $\Delta V$ . Na 0,5ns wordt de SA aangezet en wordt de tijd gemeten tot wanneer de ingangs-uitgangsknopen geladen of ontladen zijn tot 99,9% van hun finale waarde (VDD of VSS). Dit is wellicht een te strenge methode om de snelheid van de SA te bepalen aangezien de inverters al eerder zullen schakelen. Indien de snelheid van de 2 knopen verschilt, zal de traagste tijd genomen worden. De dynamische energie wordt opgemeten van het moment dat de SA wordt aangeschakeld tot dit tijdstip. Ook het statisch vermogen van de SA wordt opgemeten wanneer de ingangs-uitgangsknopen VDD en VSS bereikten. Uiteraard wordt ook geverifieerd ofdat de SA wel correct heeft gelatcht.

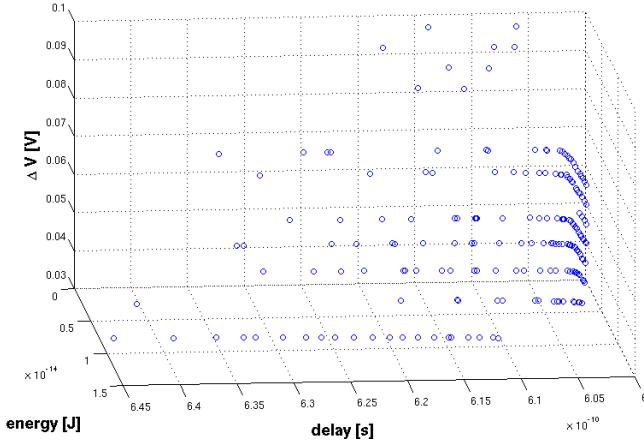
Per sense amplifier worden er 250 Monte Carlo simulaties gedaan met deze opstelling. Indien de SA niet elke keer correct functioneerde, wordt de SA verworpen. Latchte de SA wel elke keer correct, wordt het gemiddelde van de delay, dynamische energie en statisch vermogen opgeslagen.

### 5.4.2 Resultaten

Op figuur 5.8 zijn de pareto-optimale resultaten getoond van de groep sense amplifiers. Het doel van deze simulatie is veeleer om de transistorafmetingen te situeren in functie van deze optimalisatievariabelen. Voor deze simulatieopstelling kan men enkel zeggen dat de kans dat de offsetspanning lager is dan  $\Delta V$  minstens  $1 - \frac{1}{250}$  is. Dit is een veel te kleine garantie voor een sense amplifier die misschien wel miljoenen keren zal gefabriceerd worden. Voor meer informatie over de verdeling van

## 5. SENSE AMPLIFIER ANALYSE

---



Figuur 5.8: De pareto-optimale sense amplifiers

de offsetspanning te krijgen moet de standaardafwijking berekend worden met de sensitiviteitsanalyse.

### 5.5 Besluit

In dit hoofdstuk werd dieper ingegaan op de sense amplifiers, die het kleine spanningsverschil tussen datasignaal en referentiesignaal correct moet versterken tot VDD en VSS. De belangrijkste eigenschap van de SA is de offsetspanning door transistorvariaties. Deze kan voldoende klein gemaakt worden door de transistoren voldoende op te schalen. De offsetspanning kan statistisch beschreven worden met behulp van een sensitiviteitsanalyse. Tenslotte worden er ook uit een grote groep SA de pareto-optimale gekozen. De resultaten geven een idee van de grootteordes van transistorafmetingen voor een bepaalde offsetspanning, snelheid en dynamische energie.

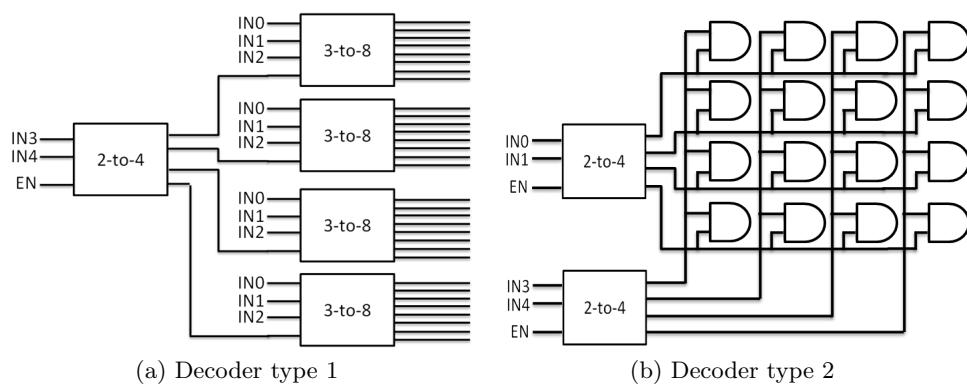
# Hoofdstuk 6

## Omringende logica

Een heleboel logische blokken, zoals decoders, drivers, pass-gates en buffers zitten verwerkt in de geheugenstructuur. In dit hoofdstuk worden deze componenten van wat dichterbij onderzocht.

### 6.1 Decoders

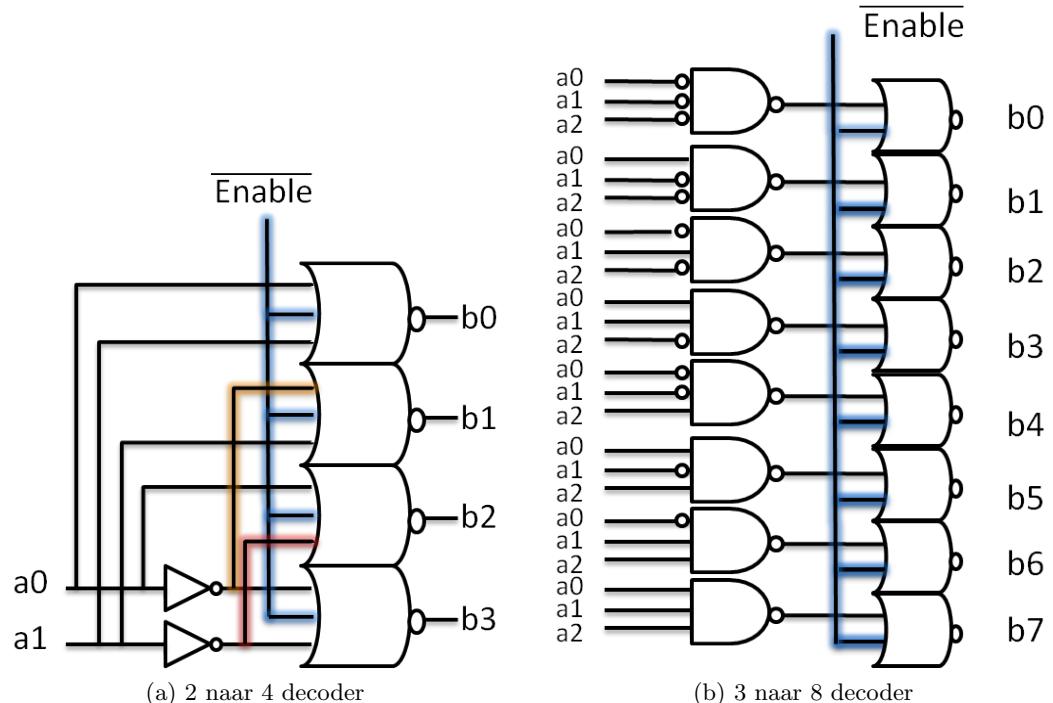
Een decoder is een logische blok dat een uitgang selecteert op basis van een geenco-deerde bus van ingangen. Aangezien het aantal globalblocks, woordlijnen en bitlijnen nog niet vastgelegd werd, werd er een gamma van decoders ontworpen gaande van een 2 naar 4 decoder tot en met een 9 naar 512 decoder. Voor het ontwerp van de grotere decoders, werd er gebruik gemaakt van de kleinere decoders als bouw blokken. Dit kan gedaan worden op 2 manieren; volgens een tree patroon ([6.1a](#)) of volgens een grid patroon ([6.1b](#)). In de volgende secties word het ontwerp van beide manieren toegelicht, en vergeleken.



Figuur 6.1: Opbouw voor grotere decoders

### 6.1.1 De tree decoder

De tree decoder is een decoder met een meerlaagse structuur dat zich uitwaaierd naar de uitgangen. De basis blokken van deze decoder zijn een 2 naar 4 decoder (figuur 6.2a) en een 3 naar 8 decoder (figuur 6.2b). In elke laag in deze structuur komen een aantal ingangen binnen en deze gaan naar de ingang van evenveel decoders als er uitgang zijn in de vorige laag. Elke uitgange van de vorige laag stuurt de enable van een van de decoders van de huidige laag aan. Dit wordt geïllustreerd in de vorm van een 5 naar 32 decoder in figuur 6.1a.



Figuur 6.2: basis decoders

### 6.1.2 De grid decoder

De grid decoder heeft een tweelaagse structuur. De eerste laag bestaat uit een aantal 2 naar 4 en/of 3 naar 8 decoders die in parallel staan. De verschillende uitgangen van deze eerste laag worden dan met AND-gates samen gevoed in een tweede laag. Om glitches te voorkomen is het belangrijk dat al de signalen gelijktijdig binnen komen in de AND-gates, vandaar dat de architectuur van 2 naar 4 decoder van figuur 6.2a veranderd werd tot een AND-OR architectuur zoals de architectuur van de 3 naar 8 decoder. Om de fanout tussen de eerste en tweede laag aan te kunnen, worden de AND-gates van de tweede laag geïmplementeerd als OR gates met inverters aan de ingangen. Deze invertors werden dan afhankelijk van de fanout als buffers gesized.

# inputs decoder	# 2naar4 decoders	# 3naar8 decoders	# AND-gates
4	2	0	16
5	1	1	32
6	0	2	64
7	2	1	128
8	1	2	256
9	0	3	512

Tabel 6.1: De verschillende aantal gates in de grid decoder

Tabel 6.1 geeft tenslotte weer hoe veel basis decoders er in de eerste laag van de grid decoder zitten en hoeveel and gates er in de tweede laag zitten, in functie van het aantal inputs

### 6.1.3 Vergelijkende studie

Eens ontworpen, kunnen de tree en grid decoders met elkaar vergelijken worden. Naast de gebruikelijke oppervlakte, energie en delay worden ook glitches, mismatch en delay verschillen tussen verschillende adressen onderzocht.

Zoals in figuur 6.3a gezien kan worden, scaalt het oppervlakte van de grid decoder veel minder als die van de tree decoder bij een groter aantal inputs. De plotse stijging in het oppervlakte van de tree decoder met 8 inputs kan verklaart worden door het gebruik van een extra laag in de boom structuur.

Het energie verbruik wordt vergeleken in figuur 6.3b. De grid decoder heeft een lichte stijging van het energie verbruik in functie van het aantal inputs, dit kan verklaart worden door het aantal gates dat geswitched wordt maar licht stijgt met het aantal inputs, daar naast gaat het meerste van de energie naar de buffers die de tweede laag aansturen (zie figuur 6.4). De tree decoder aan de andere hand heeft een sterkere stijging van energie verbruik. dit kan verklaart worden door dat alle decoders in de tree architectuur gedeeltelijk switchen. Dit zou verminderd kunnen worden door de architectuur van de basis decoders (figuur 6.1) te veranderen zodat de enable vooraan komt te staan.

De delay van de decoders kan ook afgelezen worden in figuur 6.3b. Beide decoders hebben ongeveer dezelfde delay. bij grote grid decoders kan de extra delay verklaart worden door het aansturen van een grote fanout.

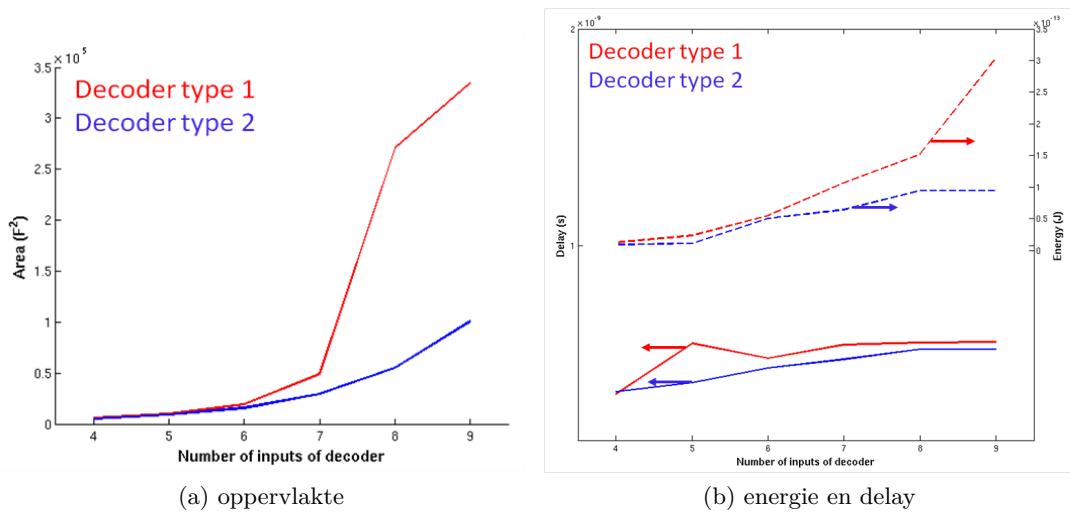
Verder werden het opduiken van glitches onderzocht. In beide decoders is er de mogelijkheid dat er glitches opduiken. Het probleem bij deze decoders is het gebruik van de NOR-gates, de glitch duikt op als de ingang veranderd van een 01 naar een 10 (zie figuur 6.5) en er is een vertraging bij een van de twee ingangen. Bij de tree decoder duiken deze glitches ingebakken in de architectuur aangezien de enable van een stage aangestuurd wordt door de vorige stage en hier altijd een zekere vertraging is. Bij de grid decoder daarentegen kan er glitch opduiken als de buffers die de tweede laag aansturen een asymmetrische delay hebben. Dit kan voor

## 6. OMRINGENDE LOGICA

komen bij bv een 5 naar 32 decoder. de uitgangen van de 2naar 4 decoder en de 3 naar 8 decoder die hier in zitten zien een andere last. Deze buffers werden met zorg ontworpen om een asymmetrische delay te voor komen.

Na een snelle mismatch analyse bleek dat de grid decoder minder variatie toon in energie verbruik en delay als de tree decoder. Tenslotte heeft de tree decoder grotere verschillen in delay, afhankelijk van welk het vorige en huidige address van de decoder is. Dit ziet men minder in de grid decoder.

Na het vergelijken van beide decoders op het vlak van oppervlakte, energie, delay, glitches, mismatch en delay verschil, Komt de grid decoder er als het beste uit en deze zal dan ook gebruikt worden in het finale ontwerp.



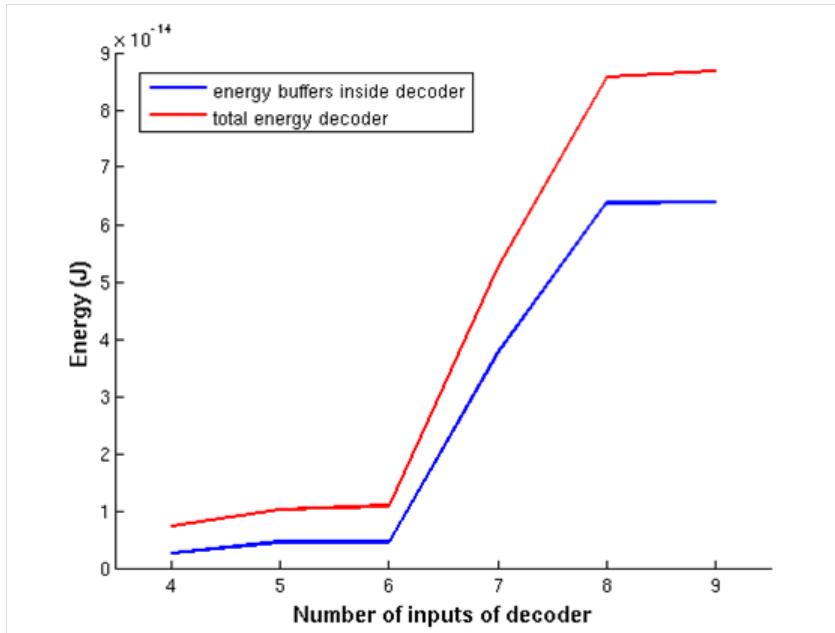
Figuur 6.3: vergelijking van decoder types

## 6.2 Buffers

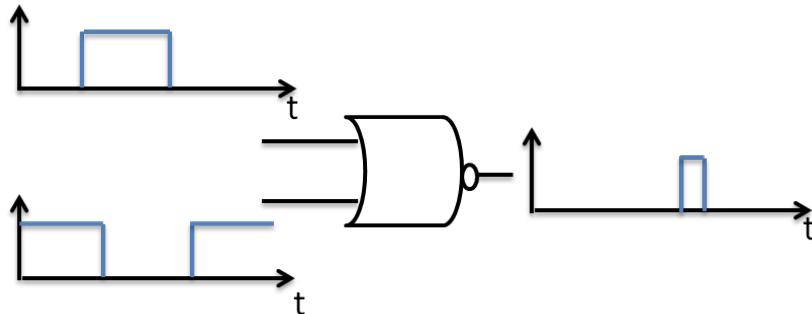
Een digitale buffer is een logische component dat de uitgang van een andere logische block meer drijf kracht geeft om een grotere last sneller aan te sturen. Buffers worden op drie plaatsen in de geheugen architectuur gebruikt. Ten eerste om de woordlijnen aan te sturen. Ten tweede om de referentie logica aan te sturen en ten slotte tussen de eerste en tweede laag in de grid decoders. Tabel 6.2 geeft een overzicht van het type last en het aantal lasten dat de verschillende buffers moeten aansturen.

De buffers werden ontworpen met logical effort waarbij het aantal stages en de sizing van elke stage werd bepaalt volgens het volgende stappen plan:

1. Bepaal de Path effort  $F = GH$  waarbij  $G = 1$  aangezien we enkel met inverters werken en  $H = \frac{C_{out}}{C_{in}}$
2. Het aantal stages wordt bepaalt door  $\hat{N} = \log_4 F$ . hierbij werd er voor een stage effort van 4 gekozen voor een optimale delay [17].  $\hat{N}$  wordt dan afgerond



Figuur 6.4: Energie verbruik in griddecoder



Figuur 6.5: Glitch in NOR-gate

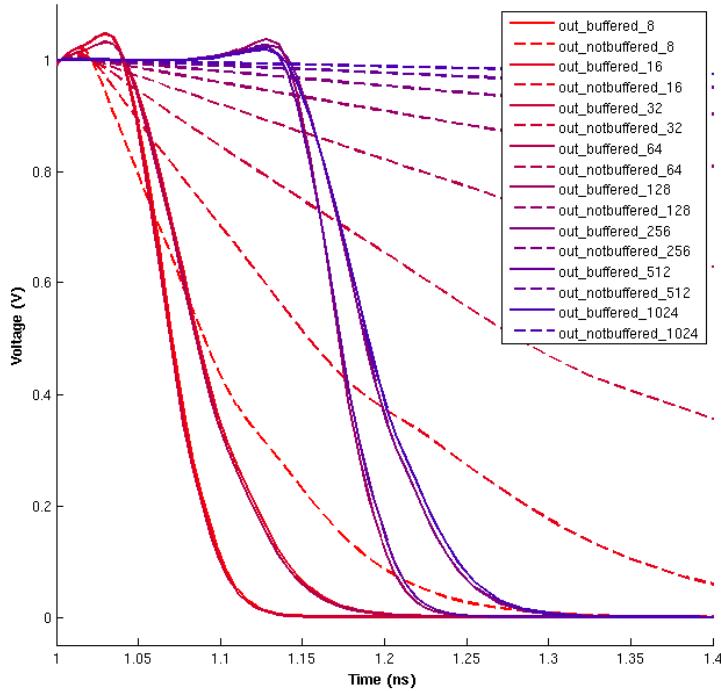
tot een even getal  $N$  voor de woordlijn en referentie buffers, en tot een oneven getal voor de decoder buffers.

3.  $N$  wordt dan gebruikt om een nieuw stage effort  $\hat{f}$  te berekenen met de formule  $\hat{f} = F^{1/N}$ .
4. Ten slotte kunnen de groottes van de verschillende invertoren in de chain berekend worden met de nieuw stage effort  $\hat{f} = gh$ .

Figuur 6.6 illustreert de ongebufferde en gebufferde signalen die naar de referentie logica gaan, voor een verschillende aantal parallele referentie logic.

	Type Last	# lasten
Woordlijn Buffer	1 Transistor	#BL
Referentie Buffer	1 Nor + 1 Inv	#BL
Decoder buffer	1 Nor	4 - 64

Tabel 6.2: Lasten in de verschillende buffers



Figuur 6.6: Gebufferde en ongebufferde signalen naar de referentie logica

### 6.3 BL- en WL-drivers

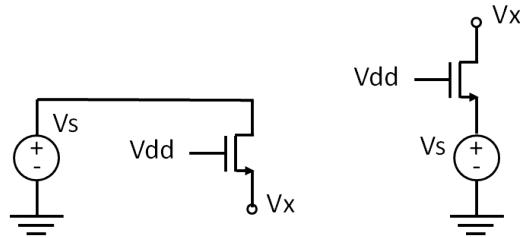
#### 6.4 Passgates

Passgates zijn schakelaars die de spanning van een laagimpedant knooppunt doorlaten naar een hoogimpedant knooppunt. Idealiter heeft de passgate geen weerstand wanneer hij aanstaat. In de praktijk zal er altijd een beetje weerstand zijn, dit heeft als gevolg dat er een kleine RC-delay vooraleer het hoogimpedante punt is geladen/ontladen tot de waarde van het laagimpedante punt. Een passgate kan gerealiseerd worden met een nMOS transistor, een pMOS of een combinatie van beiden. In wat volgt worden de verschillende scenario's besproken die kunnen

optreden wanneer de schakelaar aangezet wordt, de passgate zal immers niet altijd stroom kunnen leveren om het hoogimpedante knooppunt te (ont)laden.

#### 6.4.1 nMOS passgate

De opstelling is het circuit van figuur 6.7. Afhankelijk van de (initiële) waardes van  $V_x$  en  $V_s$ , kunnen volgende situaties optreden.

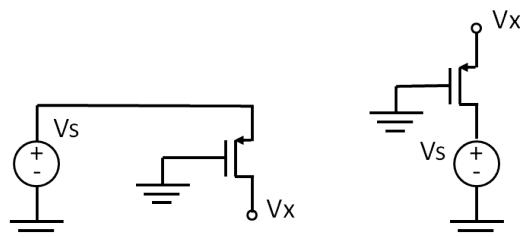


Figuur 6.7: nMOS passgate opstelling. a:  $V_s > V_x$ , b:  $V_s < V_x$

- $V_x > V_s$  en  $V_{dd} - V_s > V_{tn}$ : de transistor gaat  $V_x$  ontladen tot  $V_s$ .
- $V_x > V_s$  en  $V_{dd} - V_s < V_{tn}$ : de transistor staat af, er gebeurt niets.
- $V_x < V_s$  en  $V_s < V_{dd} - V_{tn}$ : de transistor levert stroom tot  $V_x$  is opgeladen tot  $V_s$ .
- $V_x < V_s$ ,  $V_s > V_{dd} - V_{tn}$  en  $V_x < V_{dd} - V_{tn}$ : de transistor gaat  $V_x$  opladen tot  $V_{dd} - V_{tn}$ .
- $V_x < V_s$ ,  $V_s > V_{dd} - V_{tn}$  en  $V_x > V_{dd} - V_{tn}$ : de transistor staat af, er gebeurt niets.

#### 6.4.2 pMOS passgate

De opstelling is het circuit van figuur 6.8. Er kunnen wederom verschillende situaties optreden.



Figuur 6.8: pMOS passgate opstelling. a:  $V_s > V_x$ , b:  $V_s < V_x$

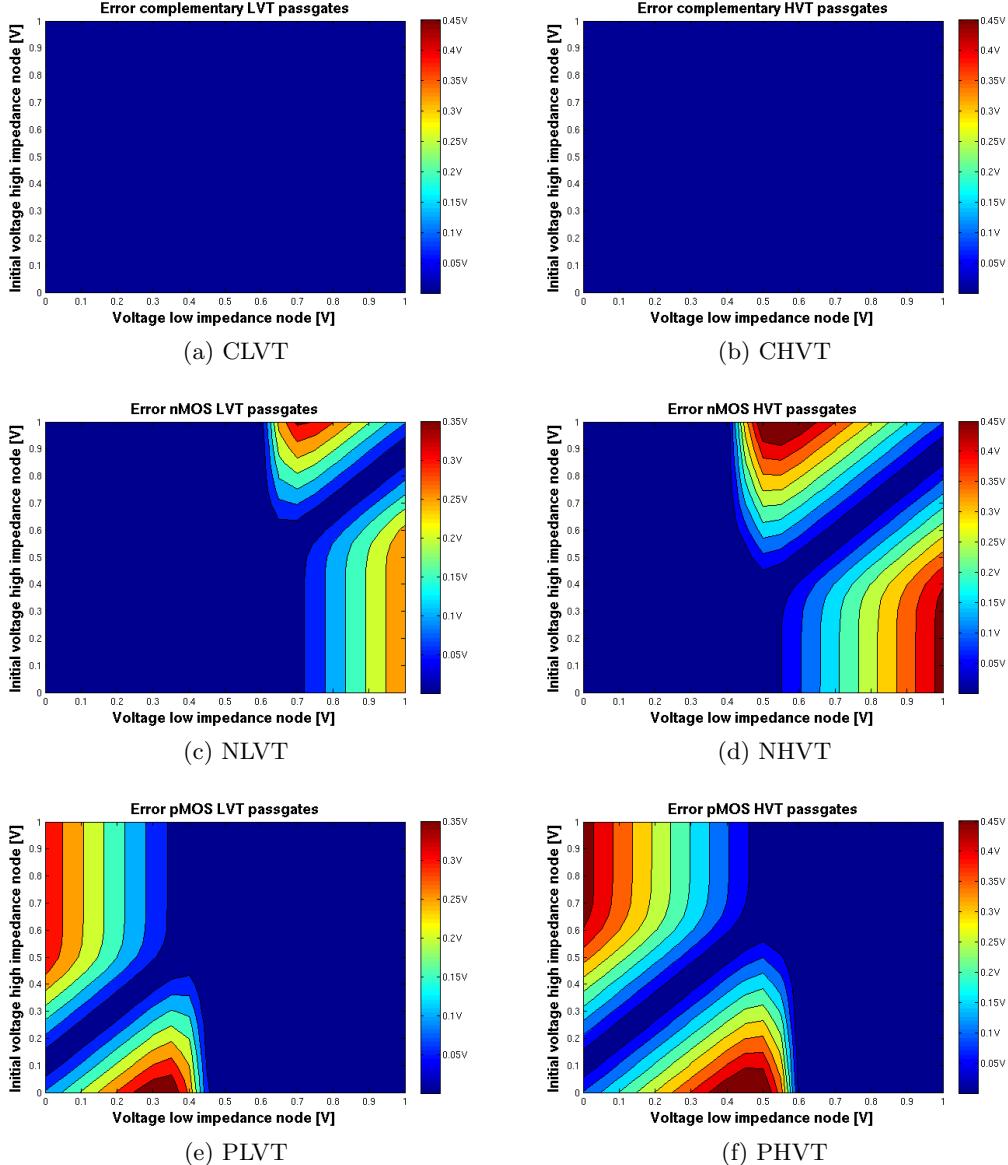
## 6. OMRINGENDE LOGICA

---

- $V_s > V_x$  en  $V_s > |V_{tp}|$ : de transistor gaat  $V_x$  opladen tot  $V_s$ .
- $V_s > V_x$  en  $V_s < |V_{tp}|$ : de transistor staat af, er gebeurt niets.
- $V_s < V_x$  en  $V_s > |V_{tp}|$ : de transistor levert stroom tot  $V_x$  is ontladen tot  $V_s$ .
- $V_s < V_x$ ,  $V_s < |V_{tn}|$  en  $V_x > |V_{tp}|$ : de transistor gaat  $V_x$  ontladen tot  $|V_{tp}|$ .
- $V_s < V_x$ ,  $V_s < |V_{tn}|$  en  $V_x < |V_{tn}|$ : de transistor staat af, er gebeurt niets.

Er zijn dus spanningszones waarvoor de passgate niet functioneert, het is belangrijk te weten wat deze zones zijn, zodat het circuit ontworpen wordt om deze zones te vermijden. Op figuur 6.9 worden deze zones in kaart gebracht voor een nMOS, een pMOS en een complementaire passgate, zowel voor LVT als HVT transistoren. Er dient opgemerkt te worden dat de passgates niet (of slechts even) aanstaan voor sommige zones, maar dat de uiteindelijke fout nog meevalt. Als  $V_s$  bijvoorbeeld 0,9V bedraagt voor een nMOS passgate en  $V_x$  initieel 0,85V (met  $V_{dd}$  1V), gaat de transistor geen stroom leveren, maar bedraagt de uiteindelijke fout 'slechts' 50mV. In deze fout zit nog geen ladingsinjectie inbegrepen, eenmaal de passgate afschakelt is de injectie onvermijdelijk. Zowel de LVT als de HVT complementaire passgates vertonen geen dode zones. Voor performantieredenen is geopteerd voor de LVT transistoren in het geheugenontwerp.

## 6.5 Besluit



Figuur 6.9: Dode zones voor verschillende types passgates



# Hoofdstuk 7

## Timing en optimalisatie

Voor een correcte werking van het geheugen, is het van belang dat de verschillende controlesignalen in een bepaalde volgorde verwerkt en doorgegeven worden. Bovendien is er ruimte voor optimalisatie door al de signalen even snel te maken als het critisch pad. In het eerste deel van dit hoofdstuk zal de invloed van architecture en sizing onderzocht worden op de timing van de signalen. De constraints en vrijheidsgraden die hier uit volgen zullen dan gebruikt worden in het tweede deel van dit hoofdstuk om een optimale architecture te vinden.

### 7.1 Timing

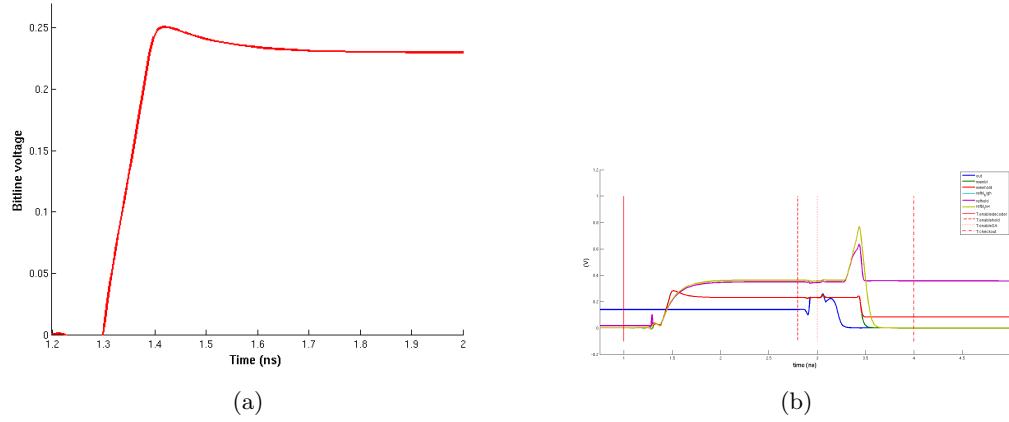
Het ontwerp van dit geheugen gaat tot het niveau van het globalblock 3.4, hierbij wordt de veronderstelling gemaakt dat alle signalen tegelijkertijd binnen komem in het globelblock. Hierna propageren de signalen door logica tot ze verschillende transistoren rond de BL aansturen. De aansturing van deze transistoren omvatten de eerste critische timing constraints. Vervolgens worden de muxen en SA aangesloten, deze zullen de tweede timing constraints bevatten.

#### 7.1.1 Critische timing voor het (de)selecteren cell

Timings problemen rond het (de)selecteren van de cell komen door het een verschil in timing voor het (de)selecteren van de load en cell. Indien de load geselecteerd wordt voor de cell zal de bitline vroegtijdig beginnen opladen naar de voedingsspanning. Wanneer de cell dan geselecteert is zal de bitline naar een betekenis volle spanning getrokken worden. Afhankelijk van het tijds verschill tussen deze twee evenementen, zal de bitlijn terug omlaag getrokken worden, wat resulteert in een energie verspilling. Dit wordt geïllustreerd in figuur 7.1a. Indien de cell gedeselecteert wordt voordat de load gedeselecteerd is, Zal de bitline ook opladen naar de voedingsspanning. Dit heeft als gevolg dat het ontladen van de bitlijn langer zal duren en de overbodige oplading resulteert ook in een energie verspilling. Door de keuze van logica (zie figure TODO) zal afhankelijk van het tijds verschill, de mux te vroeg worden afgeschakelt. Waardoor de knoop achter de mux niet volledig ontladen zal zijn. Dit heeft geen nadelige

## 7. TIMING EN OPTIMALISATIE

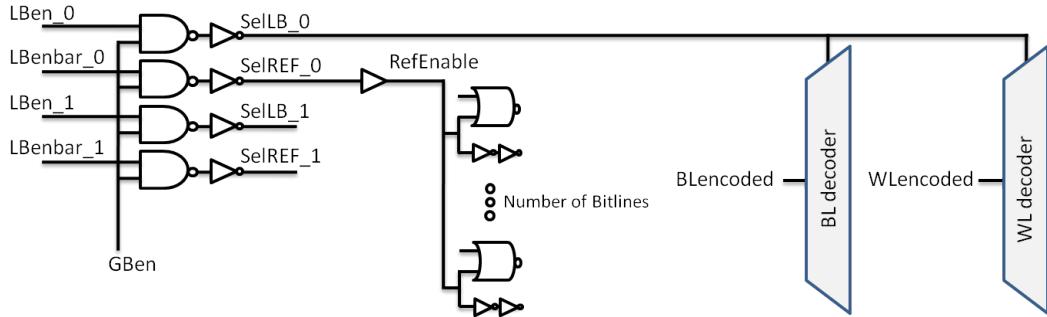
gevolgen door dat de capaciteit op dit knooppunt heel klein is en er bijgevolge een verwaarloosbare ladings injectie is in de volgende lees cyclus. Al dit word geïllustreert in figure 7.1b.



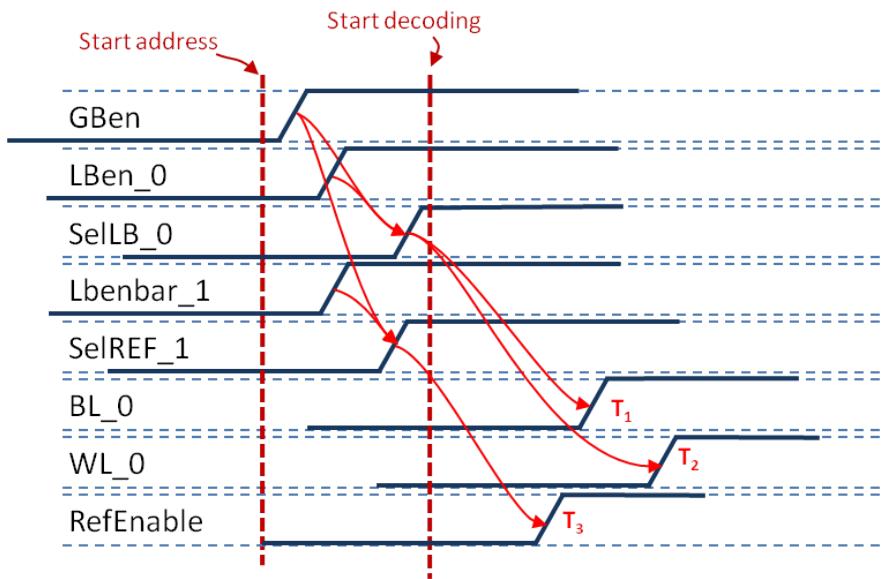
Figuur 7.1: Timing problemen bij de bitlijn

De timing begint in het globalblock. Het circuit en timings diagram word geïllustreerd in figuren 7.2 en 7.3. T1 en T2 stellen het moment voor dat de signalen uit de Bitlijn en Woordlijn decoder komen. T3 stelt het moment voor dat het signaal uit de referentie buffer komt. T1 en T3 zou op het zelfde moment moeten aankomen om een optimale timing te hebben. Indien dit niet het geval is zal de referentie bitlijnen al aanstaan vooral leer de cell bitlijn aan komt te staan. Indien er een groot aantal referentie bitlijnen zijn zal dit resulteren in een grote energie verspilling. Om deze timing te verwezelen zijn er twee opties. De eerste is het kiezen van een kleine bitlijn decoder en een grote woordlijn decoder. Dit zal voor een kleine T1 zorgen door een kleine delay in de bitlijn decoder. Dit zal een grotere T3 geven door dat de referentie buffer meer capaciteit heeft om op te laden. Een evenwicht kan zo gevonden worden om T1 en T3 op hetzelfde moment te doen verschijnen. Deze eerste optie beperkt de mogelijke architecture intensief en zal timings constraints voor T2 teniet doen zoals later zal blijken. De tweede optie voor het matchen van T1 en T3 is het vertragen van T3. Een vertraging kan gerealiseert worden door het invoeren van delay elementen of door het verslechtern van de buffer. In praktijk is ondervonden dat het invoeren van vertragings element meestal een te grote delay introduceert. Vandaar dat in de final design een buffer wordt gemaakt die niet optimaal is naar snelheid. Om het energie verbruik van de referentie bitlijnen verder te verminderen werden niet al de bitlijnen in de array gebruikt voor het genereren van het referentie signaal.

Eens de signalen uit de decoders komen worden deze gevoerd in de controle logica voor de memory array. Het circuit en timing diagram is geïllustreerd in figuren 7.4



Figuur 7.2: Globalblock logica



Figuur 7.3: Timing globalblock

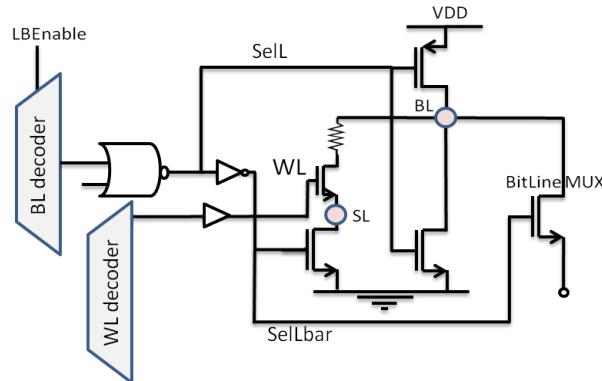
en 7.5. Bij het aanschakelen van de cell zouden de cell vroeger of tegelijk als de last moeten geschakeld worden. Op het timings diagram wordt dit geïllustreert als  $T4 = T5 = T6$ . Door de implementatie van de logica is dit niet mogelijk aangezien er altijd een inverter vertraging verschilt tussen  $T4$  en  $T5$ . Deze vertraging is minimaal en kan getollereerd worden door dat de bitlijn in elk geval moet op geladen worden tot minimum  $V_{LRS}$ . Bij lage voedingsspanningen komt dit probleem daarentegen terug boven.  $T6$  wordt bepaalt door woordlijn decoder en woordlijn buffers. Deze vertraging zou zo gemaakt moeten worden dat deze vroeger of gelijk met  $T5$  valt. Bij het afschakelijke van de cell zijn de omgekeerde voorwaarden nodig namelijk, De last zou vroeger of gelijk als de cell moeten afgeschakeld moeten worden. Deze voorwaarde is voldaan als  $T7$  voor  $T8$  en  $T9$  komt. Door de inverter is  $T7$  altijd voor  $T8$ ,  $T9$  daarentegen word bepaalt door de woordlijn decoder en buffer en zou voor

## 7. TIMING EN OPTIMALISATIE

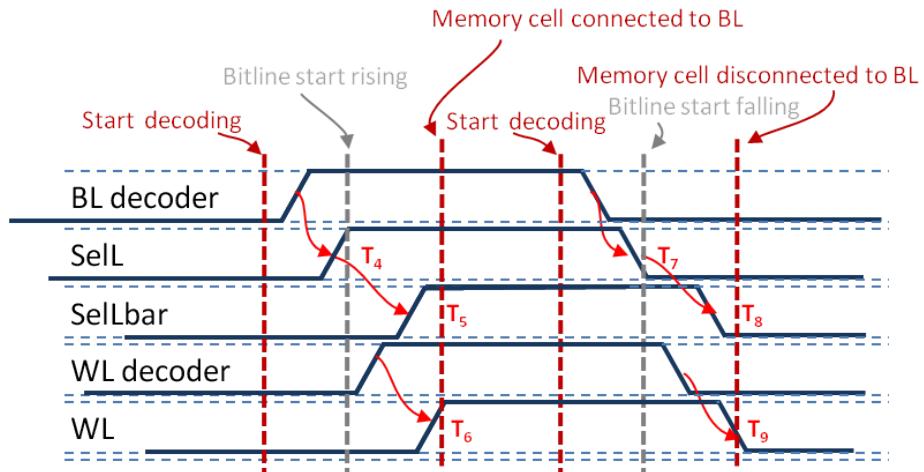
---

T7 moeten komen.

De timing van de controle logica voor de memory array staat in het circuit vast op de timing van de woordlijn na. Deze moet moet geselecteerd worden voor de sourcelijn geselecteerd is en moet gedeselecteerd worden na dat de last gedeselecteerd is. De timing van de woordlijn wordt explicet bepaald door de grote van de woordlijn decoder en impliciet door de grote van de bitlijn decoder dat de grote van de woordlijn buffer bepaalt. Figure 7.6 geeft de delay van verschillende groottes van woordlijn decoders + buffer ifv verschillende groottes van bitlijn decoders weers ... TODO

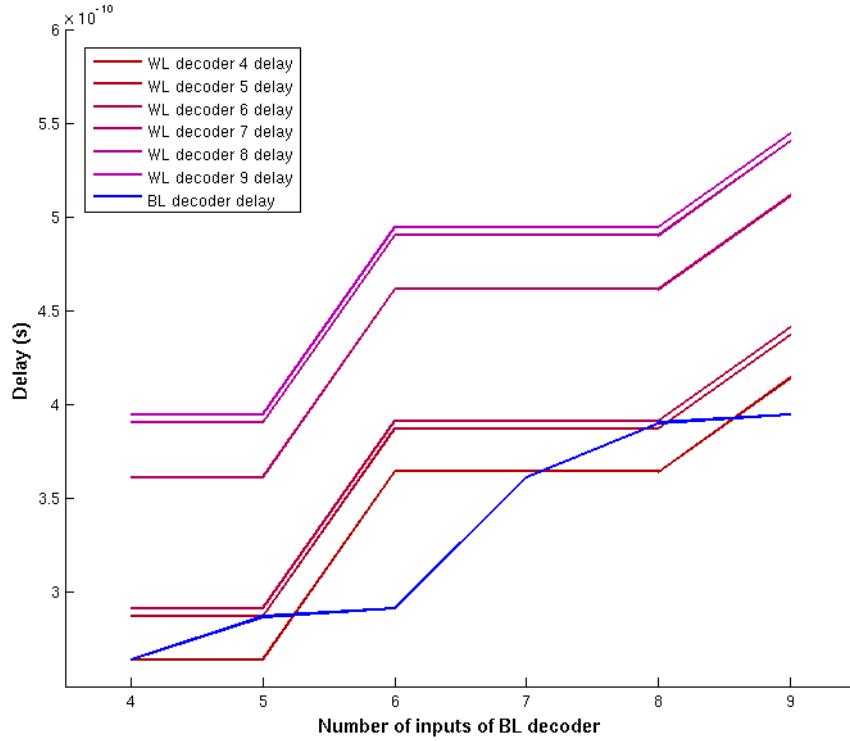


Figuur 7.4: Controle logica memory array



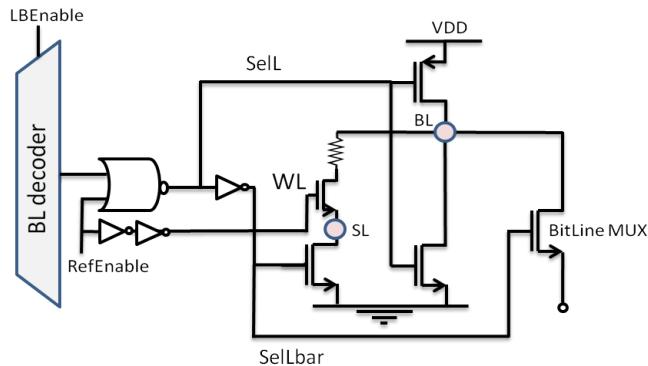
Figuur 7.5: Timing controle logica memory array

De timings voorwaarden voor het selecteren en deselecteren van de referentie cel, zijn hetzelfde als die van de memory cellen. Het circuit en timing diagram is geïllustreerd in figuren 7.7 en 7.8. Anders als bij de memory cellen worden de timings voorwaarden al in de logica zelf voldaan door dat de woordlijnen worden aangestuurd

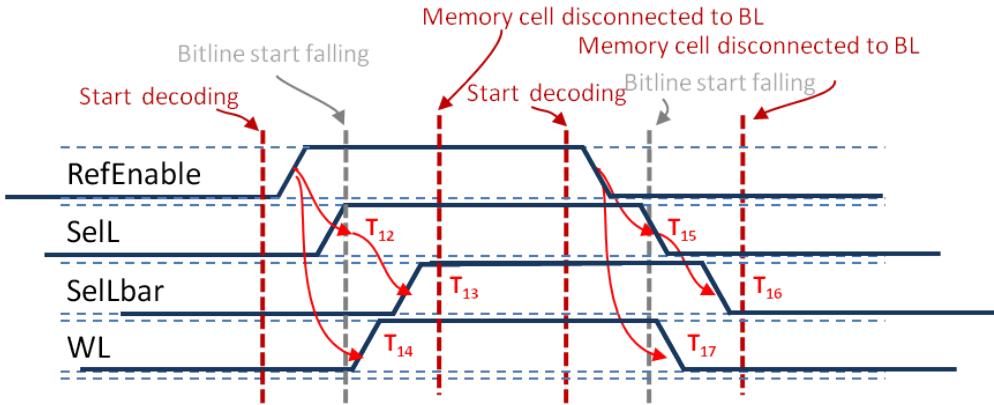


Figuur 7.6: Delay van woordlijn decoders + buffers ifv bitlijn decoders

door een signaal dat rechtstreeks van de bitlijn decoder komt. Dit signaal word dan vertraagt door twee invertoren om de juiste timing te verwezelen.



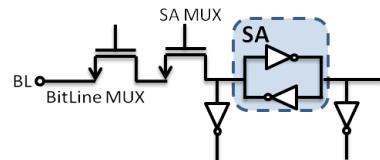
Figuur 7.7: Controle logica memory array



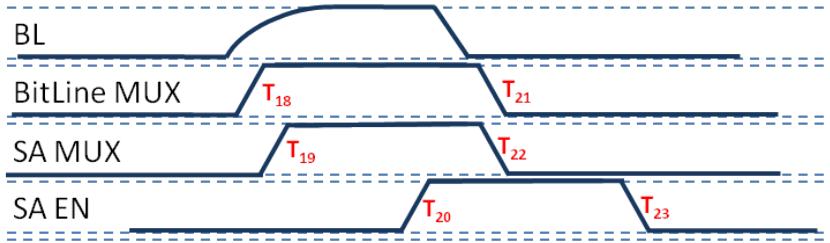
Figuur 7.8: Timing controle logica memory array

### 7.1.2 Critische timing voor het uitlezen van de cell

Eens de cel geselecteerd is wordt de bitlijn opgeladen. De volgende stap is dit signaal voeden aan de sense amplifier. Dit signaal wordt eerst door een eerst mux geleid om uit de localblock te geraken. Vervolgens wordt het signaal door een tweede mux geleid die als sample en hold dient voor de sense amplifier. Figuren 7.9 en 7.10 illustreren het circuit en timing rond de sense amplifier. Eens de bitlijn is aangesloten wordt de eerste mux automatisch aangesloten zoals uitgelegd in de vorige paragrafen. T19 stelt het tijdstip voor waar de tweede mux aangesloten moet worden. Deze timing is niet crutiaal, de mux mag zowel aangesloten worden voor als na het aansluiten van de bitlijn mux. Het tijdstip van afsluiten van deze mux (T22) is daarentegen wel belangrijk. Dit moet namelijk gebeuren voor de bitlijn mux afgesloten is (T21) anders zullen er 2 charge injections voorkomen ipv een. Om een zo snel mogelijke latching van de sense amplifiers te hebben, is het tijd stip waar de sense amplifier (T20) aangesloten word belangrijk. Wanneer de sense amplifier juist word aangesloten treed er een latching effect plaats waar de sense amplifier zich gedraagt als of er geen last aan hangt dit effect werd beschreven in sectie 5.3.2. Eens dit voorbij is zal de sense amplifier latchen aan de snelheid van de RC constante van de bitlijn. Om een snelle latching te hebben moet de tweede mux afgesloten worden voor dit latching effect voorbij is. Tenslotte kan de sense amplifier afgesloten worden eens het latchen voorbij is.



Figuur 7.9: logica rond SA



Figuur 7.10: Timing logica rond SA

## 7.2 Analyse verschillende geheugencconfiguraties

Het finale geheugen zal 4 Mbit groot zijn. Heel wat configuraties zijn mogelijk om deze grote te verwezenlijken. Om deze mogelijkheden wat in te perken worden de volgende beperking op gelegd. Het aantal Woordlijnen moet groter of gelijk zijn aan het aantal bitlijnen. Hierdoor zal het ontladen van de bitlijn sneller verlopen. Dit levert 20 mogelijke configuraties voor aantal bitlijnen,woordlijnen en globalblocks. Deze configuraties worden vergeleken op basis van hun oppervlakte, energie verbruik en leessnelheid.

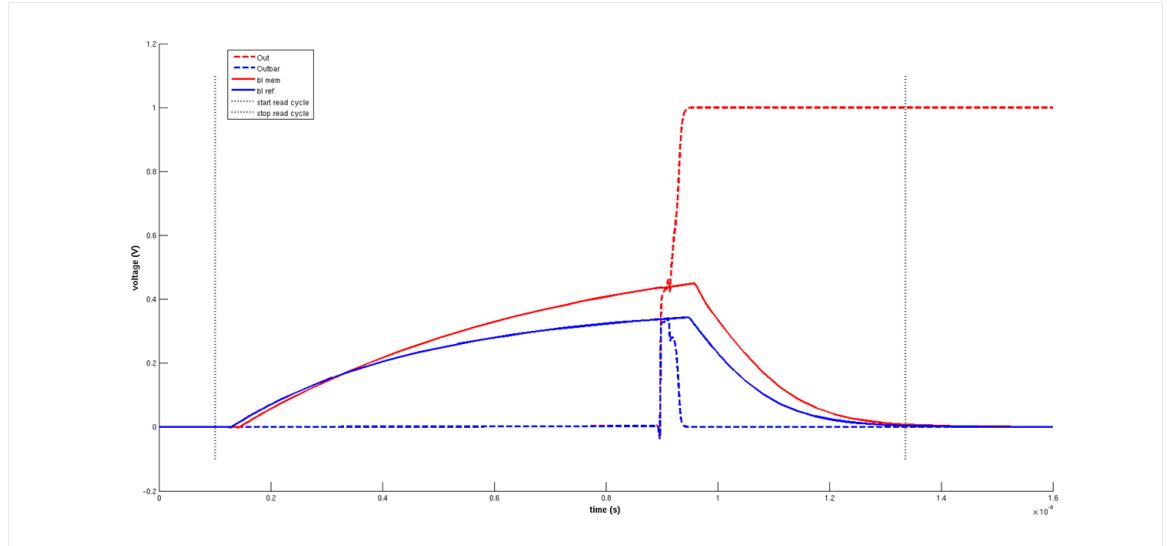
### 7.2.1 Evaluatie criteria voor de geheugencconfiguraties

Het Oppervlakte wordt berekend op basis van de lengtes en breedtes van de transistoren. Gardrings en verbindingen lijnen worden niet meegerekend in de berekeningen van het oppervlakte van de logica. De lengte van de geheugen cellen wordt  $1.5*6F$  genomen en voor de breedte wordt  $2*6F$  genomen [4]. Hoewel deze afmetingen voor een MTJ geheugen cell zijn, geven ze een goede schatting van het oppervlakte van een Memristor geheugencell. Verder is in dit oppervlakte de grote van bitlijn, woordlijn en selectlijn meegekend.

Het energie verbruik wordt berekend door de stroom van de voedingsspanning te integreren over de tijd en te vermenigvuldigen met de voedingsspanning. De signalen die binnen komen in een global block, komen van ideale spice bronnen. Dit geeft als gevolg dat er een charge injectie is naar de voedingsspanning (zie bijlage A). Dit heeft een invloed op de energie berekeningen maar er worden geen pogingen gedaan op deze te corrigeren. Verder wordt het aantal referentie cellen ook constant gehouden voor de verschillende configuraties. Dit wordt gedaan om het energie verbruik te verkleinen en omdat men maar een beperkt aantal cellen nodig heeft om een goede referentie te krijgen.

De leessnelheid is afhankelijk van de verschillende controle signalen in de lees cyclus en deze gebeurd als volgt. De lees cyclus begint wanneer de signalen binnen komen in de globalblock. De SA wordt aangesloten wanneer het verschil tussen de memory bitlijn en de referentie bitlijn 100mV bedraagt. Hierdoor wordt het tijds verschil tussen geheugens met een klein aantal woordlijnen en geheugens met een groot aantal woordlijnen verkleind, wat een betere concurrentie geeft. In het finale geheugen ontwerp zal de leessnelheid verder opgedreven worden door deze 100mV voltage

verschil te verkleinen. Verder word er ook altijd een cell met een HRS uitgelezen aangezien deze de bitlijn langer moet opladen om tot aan de 100mV verschil te komen wat een realistischere leessnelheid geeft. De lees cyclus eindigt wanneer het bitlijn voltage terug naar de grond is getrokken. Figure 7.11 illustreert de hele leescyclus.



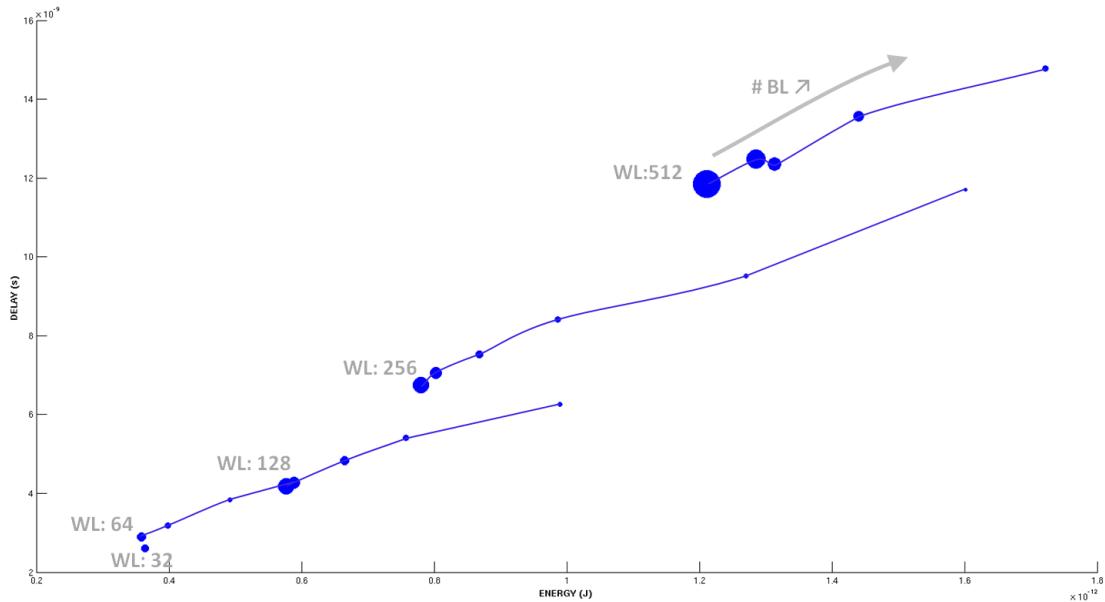
Figuur 7.11: Lees cyclus

### 7.2.2 Vergelijking van de geheugenconfiguraties

Er werden 20 mogelijke geheugenconfiguraties geselecteert als kandidaat voor het finale ontwerp, hun positie in de evaluatie ruimte wordt getoont in figuur 7.13. Hierop staat het energie verbruik op de x-as, de delay op de y-as en het oppervlakte gebruik wordt geïndiceert door de grote van de punten. Het energie verbruik en delay wordt voornamelijk bepaalt door het aantal woordlijnen en bitlijnen. De delay wordt voornamelijk bepaalt door het opladen van de bitlijnen, wat dan ook de voornaamste vorm van energie verbruik is. De snelheid van de bitlijnen wordt dan weer bepaalt door het aantal woordlijnen wat gezien kan worden in figuur 7.12. Het aantal bitlijnen beïnvloed dan weer meer het energie verbruik. Dit extra energie verbruik gaat teneerste naar de woorlijn buffers, tentweede naar de bitlijn decoders en tenslotte naar de bitlijn zelf. Deze laatste is door dat de bitlijn bij het deselecteren van de cell langer aanblijft dan bij een kleiner aantal bitlijnen. Dit is ook de reden waarom een groter aantal bitlijnen een langere delay heeft. Bij alle geheugen configuraties gaat het vermogen verbruik eerst naar de geheugecell, vervolgens naar de logica, vervolgens naar de buffers en ten slotte naar de sense amplifiers. Het oppervlakte wordt bepaalt door het aantal globalblocks en de grote van de decoders. Een groot aantal woordlijnen in combinatie met een klein aantal bitlijnen geeft de noodzaak aan een groot aantal globalblocks en dit geeft een groot oppervlak als gevolg.

Als conclusie kan gezegd worden dat de optimale geheugen configuratie bestaat uit

een klein gelijk aantal woordlijn en bitlijnen wat een optimum zal geven voor energie verbruik en delay, en een sub optimum zal geven voor oppervlakte.

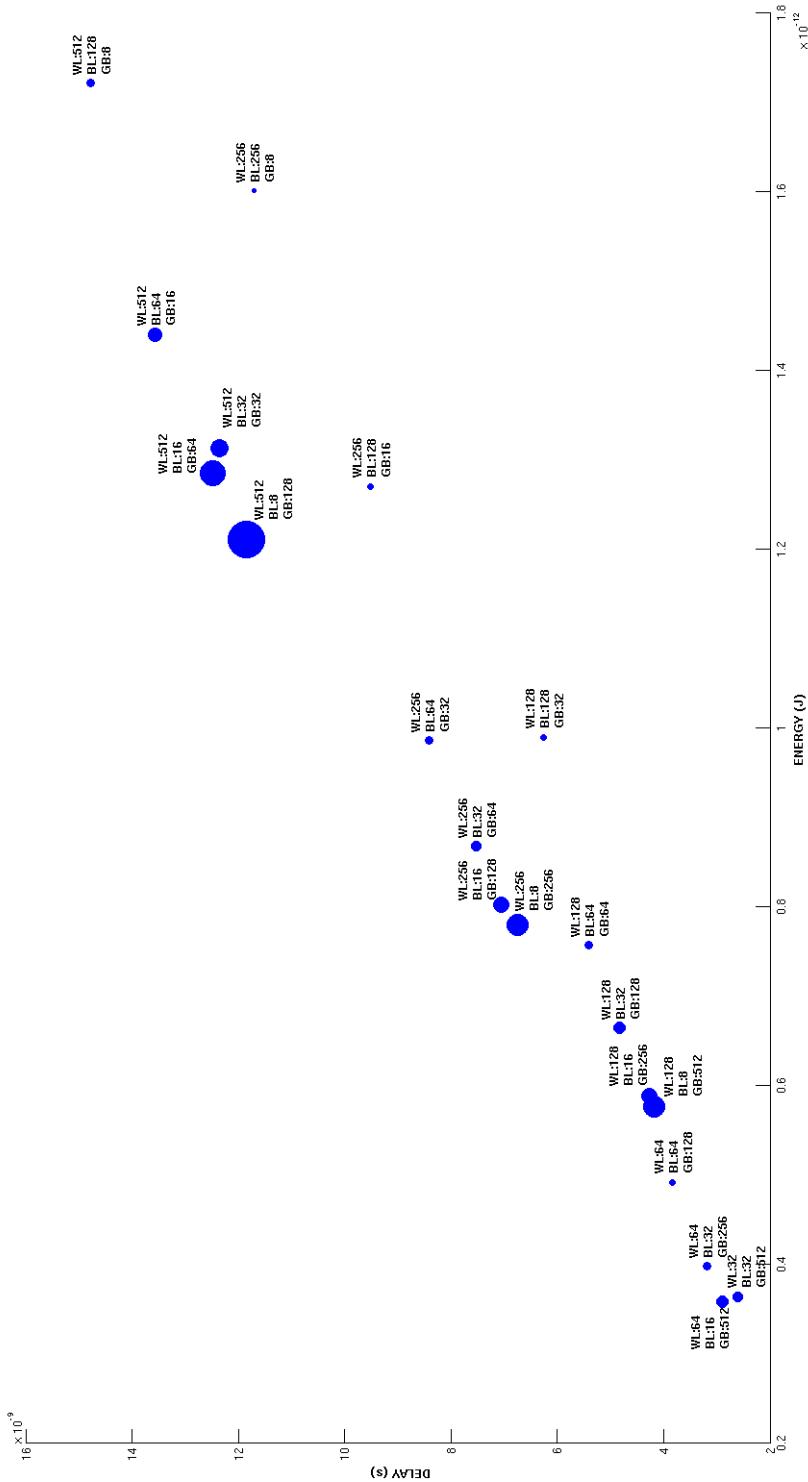


Figuur 7.12: Invloed #BL en #WL op delay, energie verbruik en oppervlakte

### 7.3 Besluit

## 7. TIMING EN OPTIMALIZATIE

---



Figuur 7.13: Delay, energie verbruik en oppervlakte van alle geheugenconfiguraties

# Hoofdstuk 8

## Finaal ontwerp

### 8.1 Het finaal ontwerp

Het finale ontwerp is een geheugen dat uit 32BL, 32WL en 512GB bestaat. Van de 32 BL worden er 16 gebruikt voor het genereren van het referentie spanning. En van deze 16 worden 6 in RHS en 12 in LRS gehouden. Dit om de referentie spanning beter te centreren tussen de bitlijn spanningen voor cellen in RHS en LRS. De afmetingen van alle transistoren staan in tabel 8.1.

Op dit geheugen wordt een speed-vdd test uitgevoerd. Dit is een test waarbij de voedings spanning wordt verlaagt en vervolgens gekeken wordt aan welke snelheid de lees cyclus kan uitgevoerd worden. In deze test werden de tijdstippen wanneer de senseamplifier aan geschakelt en wanneer de uitgang nagekeken werd, onafhankelijk van de voedingsspanning bepaalt. Dit geeft een iets optimistisere resultaten dan dat dit door een digitaal circuit zou worden aangestuurd. Figuur 8.1 toont de resultaten van deze test. Op elk punt in de figuur werden 100 montecarlo simulatie uitgevoerd. Zoals men duidelijk kan zien daalt de lees snelheid bij het verlagen van de voedingsspanning. Dit komt door een combinatie van 2 zaken. Ten eerste gaat de logica trager worden, dit heeft als gevolg dat de bitlijnen later worden aangestuurd en dat er een verschil ontstaat tussen de aansturing van de bitlijn en woordlijn. Dit verschil uit zich in het snel stijgen van de bitlijn spanning zoals in het vorig hoofdstuk word geillustreerd in figuur 7.1a. Door deze snelle stijging van de bitlijn moet met langer wachten om het bitlijn voltage van een lage cell uit te lezen. Het tweede fenomeen dat de leessnelheid doet vertragen is de sense amplifier. Bij een voedingsspanning van 1V kan de senseamplifier binnen de 0.25ns schakelen. Bij lagere voedingsspanningen kan dit afhankelijk van de mismatch tot 2ns duren vooralleer de senseamplifier volledig geschakelt is.

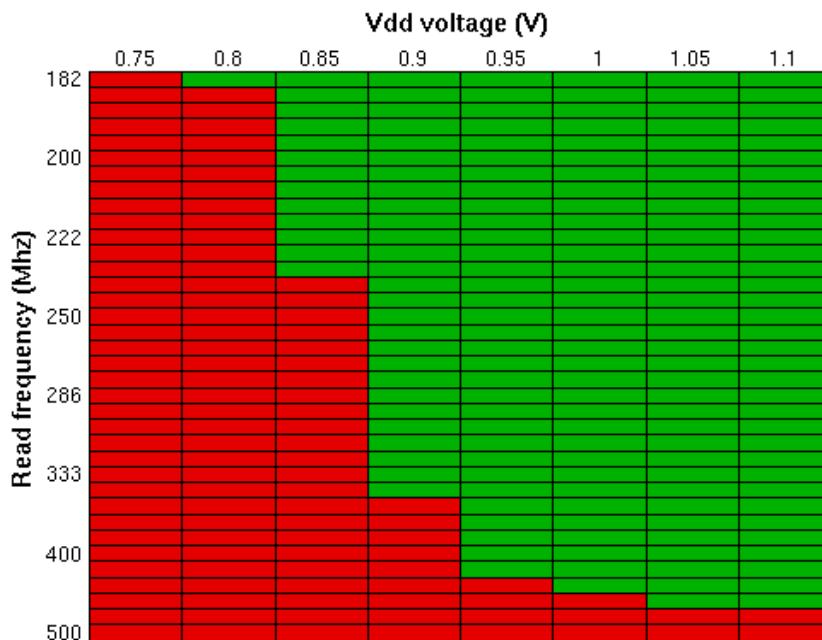
Verder kan men ook zien dat de schakeling een voedingsspanning hoger of gelijk aan 0.8V nodig heeft om correct te kunnen werken. De verklaring hiervoor kan gezien worden in figuur 8.2. Deze figuur stelt de distributie voor van de bitlijn spanningen van een cell in RHS, de referentie cellen en een cell in LRS in functie van verschillende voedings spanningen. Er kan duidelijk gezien worden dat bij het verlagen van de voedings spanningen deze distributies dichter bij elkaar komen te liggen en dat een

## 8. FINAAL ONTWERP

---

Transistor	L (nm)	W (nm)
ChargeBL	195	300
DischargeBL	45	100
DischargeSL	45	500
Sa enableP	45	900
Sa enableN	45	500
Sa P	45	1700
Sa N	45	1500
Mux LB	45	200
Mux GB	45	100

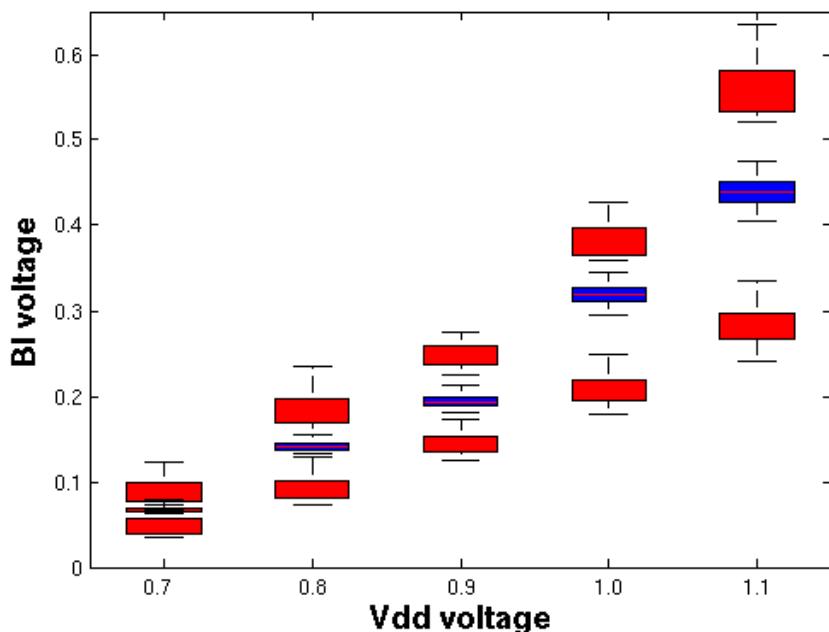
Tabel 8.1: Grootes van de transistoren in het finaal ontwerp



Figuur 8.1: Resultaten speed-vdd test

voedingsspanning van 0.8V wel degelijk een limiet is. Aangezien de extrems van de distributies bij een voedingsspanning van 0.8V zo dicht bij elkaar zitten, word er verder verwacht dat bij deze spanning de schakeling ook occasioneel zal falen door dat de senseamplifier ontworpen is voor een  $\Delta V$  van 35mV. Dit is echter niet opgedoken in de 100 montecarlo simulaties. Als men naar de bitlijn voltage distributies kijkt voor een voedingsspanning van 1V zal men opmerken dat deze niet dezelfde zijn als dezelfde distributie die getoont werden in het hoofdstuk van de last impedatie

(figuur 4.11). Dit komt door dat men in de speed-vdd test niet wacht to de bitlijn volledig is opgeladen, wat een tijds winst oplevert. Ook werd er in het hoofdstuk van de last impedatie voor gesteld dat er een energie winst zou bereikt kunnen worden door een andere last te kiezen (sectie 4.3.3). Hoewel dit mogelijk is, heeft dit wel als nadeel dat de schakeling minder tolerant zal zijn voor voedingspanning verschillen. Ten slotte moet ook vermeld worden dat de speed-vdd test uitgevoerd werd op een (spice) temperatuur van 30°C. Moest deze schakeling worden geïmplementeert in een processor is de kans groot dat dit onderhevig zal zijn aan temperaturen tussen de 30°C en 60°C wat ook een tragere leest snelheden zal opleveren. Hoe traag, werd echter niet onderzocht.



Figuur 8.2: BI spanningen ifv Vdd

Het totale energie verbruik van een leescyclus bij een voeding spanning van 1V is gemiddelt 0.51pJ. Hier bij gaat 25% van de energie naar de logic, 2% naar de senseamplifier, 65% naar de bitlijnen en 8% naar de buffers. Hier bij werden de buffers in de decoders bij de logic gerekent.

## 8.2 Besluit



## **Hoofdstuk 9**

### **Besluit**

De masterproeftekst wordt afgesloten met een hoofdstuk waarin alle besluiten nog eens samengevat worden. Dit is ook de plaats voor suggesties naar het verder gebruik van de resultaten, zowel industriële toepassingen als verder onderzoek.



# **Bijlagen**



## Bijlage A

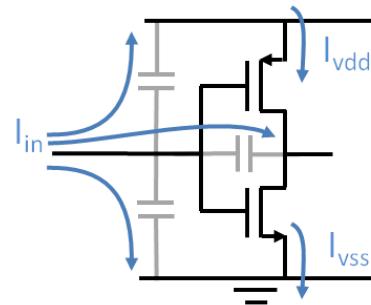
# Charge injectie met ideale spice bronnen.

Bij het berekenen van de energie consumptie van verschillende bouwblokken, valt het op dat de voeding, stroom opneemt ipv afgeeft bij het schakelen van de ideale spice bronnen. Dit komt door een charge injectie van de ideale spice bronnen. Om dit te verificeren, worden de stromen van een simpele inverterschakeling bestudeert. Figuur A.1 stelt een simple inverterschakeling voor met relevante parasite capaciteiten. Bij het plaatsen van een stap functie aan de inverter, zal er een lading door de capaciteiten vloeien. Omdat de positieve stroom die geobserveert wordt in de voeding afkomstig komt van de ingang, moet de som van de stroom door de ingang, voeding en grond moet nul zijn. De stroom door de ingang kan in Spice opgemeten worden door een weerstand met resistieve waarde gelijk aan nul, in serie met de ingang te zetten.

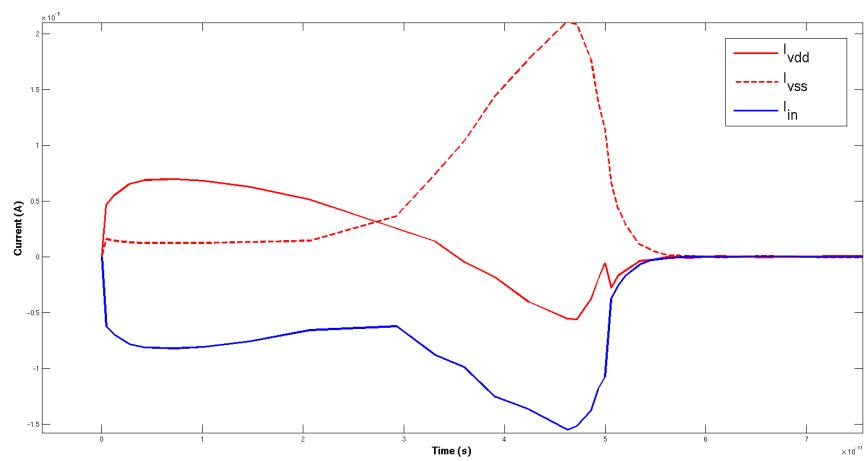
Figure A.2 toont deze drie stromen. In eerste deel van de figuur (tot tijdstip  $3 \cdot 10^{-11}$ ) is de input al aan het stijgen maar de inverter is nog niet aan schakelijk. De voeding en grond stromen komen dan puur van de ingang. Vanaf tijdstip  $3 \cdot 10^{-11}$ , is de inverter aan het schakelen en is er een aandeel van de stroom in de grond dat uit de voeding komt. De som van alle drie stromen is ten alle tijden gelijk aan nul. In conclusie is hierbij aangetoont dat er een charge injectie is van ideale spice bronnen in het circuit. Dit heeft een invloed op de stromen en daardoor ook de energie berekeningen, maar dit verwaarlozen we bij onze berekeningen.

### A. CHARGE INJECTIE MET IDEALE SPICE BRONNEN.

---



Figuur A.1: Testcircuit ladingsinjectie



Figuur A.2: Stromen in circuit

# Bibliografie

- [1] I. Baek, M. Lee, S. Seo, M.-J. Lee, D. Seo, D. S. Suh, J. Park, S. Park, T. Kim, I. Yoo, U.-i. Chung, and J. Moon. Highly scalable nonvolatile resistive memory using simple binary oxide driven by asymmetric unipolar voltage pulses. In *Electron Devices Meeting, 2004. IEDM Technical Digest. IEEE International*, pages 587–590, Dec 2004.
- [2] Y.-Y. Chen, L. Goux, L. Pantisano, J. Swerts, C. Adelmann, S. Mertens, V. Afanasiev, X. Wang, B. Govoreanu, R. Degraeve, S. Kubicek, V. Paraschiv, B. Verbrugge, N. Jossart, L. Altimime, M. Jurczak, J. Kittl, G. Groeseneken, and D. Wouters. Fully cmos beol compatible hfo<sub>2</sub> rram cell, with low program current, strong retention and high scalability, using an optimized plasma enhanced atomic layer deposition (peald) process for tin electrode. In *Interconnect Technology Conference and 2011 Materials for Advanced Metallization (IITC/MAM), 2011 IEEE International*, pages 1–3, May 2011.
- [3] L. . CHUA. Memristor-the missing circuit element. *IEEE Transactions of circuit theory*, September 1971.
- [4] S. Cosemans. Intro regarding read sensing schemes. powerpoint presentation, 2013.
- [5] Y. Deng, P. Huang, B. Chen, X. Yang, B. Gao, J. Wang, L. Zeng, G. Du, J. Kang, and X. Liu. Rram crossbar array with cell selection device: A device and circuit interaction study. *Electron Devices, IEEE Transactions on*, 60(2):719–726, Feb 2013.
- [6] K. M. Kim, B. J. Choi, B. W. Koo, S. Choi, D. S. Jeong, and C. S. Hwang. Resistive switching in pt/al<sub>2</sub>o<sub>3</sub>/tio<sub>2</sub>/ru stacked structures. *Electrochim. Solid State Lett.*, 9G343–G346, 2006.
- [7] P. J. Kuekes, D. R. Stewart, and R. S. Williams. The crossbar latch: Logic value storage, restoration, and inversion in crossbar circuits. *Journal of Applied Physics*, 97(3):–, 2005.
- [8] K. J. Kuhn. Variation in 45nm and implications for 32nm and beyond. powerpoint presentation, 2009.

- [9] L. Liu, Y. Hou, D. Yu, B. Chen, B. Gao, Y. Tian, D. Han, Y. Wang, J. Kang, and X. Zhang. Multilevel set/reset switching characteristics in al/ceox/pt rram devices. In *Electron Devices and Solid State Circuit (EDSSC), 2012 IEEE International Conference on*, pages 1–3, Dec 2012.
- [10] G. E. MOORE. Cramming more components onto integrated circuits. *Electronics*, April 1965.
- [11] K. Prall and K. Parat. 25nm 64gb mlc nand technology and scaling challenges invited paper. In *Electron Devices Meeting (IEDM), 2010 IEEE International*, pages 5.2.1–5.2.4, Dec 2010.
- [12] T. Raja and S. Mourad. Digital logic implementation in memristor-based crossbars. In *Communications, Circuits and Systems, 2009. ICCCAS 2009. International Conference on*, pages 939–943, July 2009.
- [13] F. Ren, H. Park, R. Dorrance, Y. Toriyama, C.-K. Yang, and D. Markovic. A body-voltage-sensing-based short pulse reading circuit for spin-torque transfer rams (stt-rams). In *Quality Electronic Design (ISQED), 2012 13th International Symposium on*, pages 275–282, March 2012.
- [14] G. Rose, J. Rajendran, H. Manem, R. Karri, and R. Pino. Leveraging memristive systems in the construction of digital logic circuits. *Proceedings of the IEEE*, 100(6):2033–2049, June 2012.
- [15] R. D. Stefan Cosemans. Verilog-a implementation of hourglass model. powerpoint presentation, 2012.
- [16] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams. The missing memristor found. *Nature*, 453(7191):80–83, 2008.
- [17] I. Sutherland, B. Sproull, and D. Harris. *Logical Effort: Designing Fast CMOS Circuits*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.
- [18] H. S. P. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F. Chen, and M.-J. Tsai. Metal oxide rram. *Proceedings of the IEEE*, 100(6):1951–1970, June 2012.
- [19] D. Wouters. Oxide resistive ram (oxrram) for scaled nvm application. powerpoint presentation, 2009.

## Fiche masterproef

*Studenten:* Wouter Diels  
Alexander Standaert

*Titel:* Ontwerp van een RRAM geheugen

*Engelse titel:* The best master thesis ever

*UDC:* 621.3

*Korte inhoud:*

Hier komt een heel bondig abstract van hooguit 500 woorden. L<sup>A</sup>T<sub>E</sub>X commando's mogen hier gebruikt worden. Blanco lijnen (of het commando \par) zijn wel niet toegelaten!

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Thesis voorgelegd tot het behalen van de graad van Master of Science in de ingenieurswetenschappen: elektrotechniek, optie Elektronica en geïntegreerde schakelingen

*Promotor:* Prof. dr. ir. W. Dehaene

*Assessoren:* Prof. dr. ir. R. Lauwereins  
Prof. dr. ir. M. Verhelst

*Begeleiders:* ir. B. Baran  
dr. ir. S. Cosemans